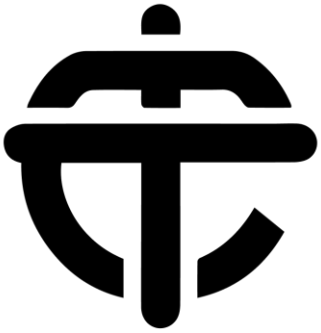
epb
escola profissional de braga

PAP - PROVA DE APTIDÃO PROFISSIONAL

Cryptrail



Discover your crypto journey.

CURSO PROFISSIONAL DE

GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS

Rodrigo Lopes Ferreira

Trabalho realizado sob a orientação de:

Prof. António Araújo - Acompanhante

Prof. João Delgado

Prof. Américo Rodrigues

Prof. Ana Marta Vilaverde

Escola Profissional de Braga

Braga, 10 de maio de 2024

PESSOAS 2030 | PORTUGAL 2030 | Cofinanciado pela União Europeia | SELO DE CONFORMIDADE EQAVET | uma escola Rumos education | Knowledge sharing

Os Fundos Europeus mais próximos de si.

Nota Introdutória

A Prova de Aptidão Profissional consiste na apresentação e defesa, perante um júri, de um projeto, consubstanciado num produto, material ou intelectual, numa intervenção ou numa atuação, bem como do respetivo relatório final de realização e apreciação crítica, demonstrativo de saberes e competências profissionais, adquiridos ao longo da formação e estruturante do futuro profissional.

A sua realização constitui-se como um dos imperativos legais para a conclusão do curso profissional de (designação do curso), que confere uma dupla certificação: qualificação profissional de nível IV e o 12º ano como certificação escolar de nível secundário.

A presente prova foi realizada na Escola Profissional de Braga no presente ano letivo 2023/2024 e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.

Os Fundos Europeus mais próximos de si.

Dedicatória

Dedico a minha prova de aptidão aos meus professores, família e amigos que sempre me apoiaram ao longo desta jornada.

Agradecimentos

Agradeço especialmente ao professor António Araújo, pois desde que começou a lecionar a nossa turma, simplificou significativamente a forma como aprendo programação. Aprecio imenso a sua maneira de ensinar.

Agradeço também em particular ao professor João Delgado, pela sua honestidade e franqueza ao dar opiniões, o que tem sido bastante benéfico para o meu desenvolvimento.

Quero também deixar o meu especial obrigado à diretora de turma, Professora Ana Marta Vilaverde, pela sua dedicação aos seus alunos, e por demonstrar tanta empatia para connosco.

Tenho a agradecer a estas e outras pessoas/professores por sempre me terem apoiado nas minhas escolhas, mesmo quando enfrentei dificuldades.

Índice

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Índice de Figuras

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Introdução

Fundamentação

A minha prova de aptidão profissional consiste na criação de uma aplicação e de um website com o nome "Cryptrail", para a facilidade na análise do mercado das moedas digitais.

Com o crescimento exponencial da utilização das moedas no mercado, a Cryptrail visa fornecer dados precisos e atualizados sobre os preços, permitindo aos utilizadores tomar decisões de investimento bem informadas neste universo em constante evolução.

A minha ideia é destacar os melhores sites para compra de criptomoedas (com as respetivas taxas indicadas) recorrendo a um conversor de moeda fiduciária para crypto (ou vice-versa), notícias em tempo real, integração de uma AI focada apenas para este nicho e uma página com as moedas e "*Exchanges*" disponíveis no mercado com algumas informações.

O website servirá apenas para dar informações ao utilizador sobre a app, termos e condições, suporte ao utilizador, políticas de privacidade, política de cookies e requisitos mínimos de sistema para utilização da aplicação vai ser o conteúdo do website.

Na minha aplicação, utilizarei como linguagem de *front-end React Native*, uma framework de *React js*, utilizada para desenvolvimento mobile. Para o *back-end*, utilizarei *Node.js*, uma linguagem em ambiente JavaScript. Para as bases de dados, vou trabalhar com PostgreSQL (pgsql), que é um sistema de gestão de bases de dados relacional baseado em SQL.

Já no website, será utilizado *HTML, CSS*, e um pouco de *JavaScript* para dinamizar a página.

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Objetivos:

- Reforçar a capacidade de superar situações inesperadas ou imprevistas nas diferentes fases do projeto;
- Desenvolver o espírito crítico e sentido de autonomia;
- Desenvolver competências de decisão, escolha e procura de informação;
- Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas;
- Melhorar a disciplina mental, capacidade de organização e metodologia de trabalho;
- Aumentar a minha capacidade de domínio sobre este tema;
- Desenvolver competências no desenvolvimento de aplicações para um sistema operativo móvel (IOS), e adquirir a sensibilidade para a publicação e gestão na loja virtual;
- Aplicar conhecimentos e competências adquiridos na área de programação em dispositivos móveis;
- Tornar-me uma pessoa proficiente em programação.

Cronograma		
Fase	**Calendarização**	**Elenco de atividades nas diversas fases**
Conceção	Setembro 2023	Conhecimento do regulamento
	Setembro 2023	Definição dos critérios da PAP pela direção
	Setembro 2023	Período de negociação / reflexão / decisão sobre os temas / problemas a explorar
	Setembro 2023	Normas para a elaboração de um trabalho científico
	Até 31 outubro 2023	Entrega do projeto de acordo com as normas do regulamento

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Desenvolvimento	Setembro 2023	Escolha do tema, baseado no interesse pela área das moedas digitais
	Setembro 2023	Escolha do nome e do slogan
	Setembro 2023	Decisão final sobre o desenvolvimento da aplicação
	Outubro 2023	Início do desenvolvimento do Website
	Outubro 2023	Iniciar estudo sobre React Native
	Novembro 2023	Início do desenvolvimento da aplicação
	Dezembro 2023	Acabar o Website
	Março 2024	Acabar Aplicação
	Abril 2024	Entrega do relatório
Autoavaliação e Relatório	Janeiro 2024	Defesa intermédia da PAP
	6 maio 2024	Conclusão do projeto e entrega do relatório
	Julho 2024	Defesa final da PAP

Recursos		
Humanos	Materiais	Financeiros
Prof. António Araújo	Computador	Domínio (cryptrail.pt)
Prof. Ana Marta Vilaverde	Visual Studio Code, Vercel, GitHub, Docker, Figma, Expo Xcode, Photoshop, Shotrr Dribble, Notion	Subscrição OpenAI API
Prof. João Delgado	PostgreSQL, React Native, HTML, CSS, JavaScript, Node.js	

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

1. Componente Teórica

No meu projeto, foram utilizados diversos recursos, cada um com finalidades específicas, desde linguagens de programação e *frameworks* até editores de texto e ferramentas de edição de imagens.

1.1 Programação da Aplicação/Website

1.1.1 Front-end

1.1.1.1 JavaScript

Para o desenvolvimento da aplicação (*front-end* e *back-end*), utilizei JavaScript. É uma linguagem *full-stack*, amplamente utilizada tanto no cliente como no servidor. Facilita a criação de código dinâmico e interativo, apesar de algumas limitações de desempenho.

Na minha aplicação, JavaScript foi utilizado e interpretado por React Native e Node.js.

Figura 1 - Logótipo JavaScript

1.1.1.2 React Native

Para a realização do *front-end*, utilizei React Native. React Native é uma biblioteca JavaScript criada pelo Facebook. É relativamente recente, com apenas 9 anos e é das melhores e mais utilizadas bibliotecas atualmente. Optei por React Native por permitir o desenvolvimento de aplicações móveis nativas com uma única base de código, pela prévia familiaridade com JavaScript e pela sua fácil aprendizagem.

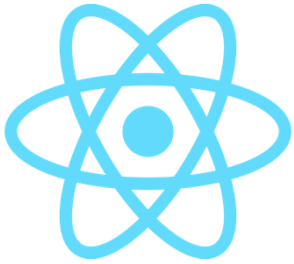


Figura 2 - Logótipo React Native

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

1.1.1.3 HTML

Para a criação do meu Website, usei HTML5. Esta linguagem de marcação serve para definir a estrutura e o conteúdo de uma página através de marcas específicas, conhecidas como "tags", que indicam diferentes tipos de conteúdo como cabeçalhos, parágrafos, ligações e imagens.

Figura 3 - Logótipo HTML5

1.1.1.4 CSS

Para a estilização do meu Website, utilizei CSS (*Cascading Style Sheets*). É uma linguagem de estilização web, que é escrita numa folha em estilo de cascata. Com CSS, é possível aplicar estilos, como cores, fontes e espaçamentos, aos elementos de uma página web. Permite também a criação de um design responsivo que se adapta a diferentes tamanhos de ecrã.

Figura 4 - Logótipo CSS

1.1.2 Back-end

1.1.2.1 Node.js

Para o desenvolvimento do *back-end*, utilizei Node.js. O Node.js é um *runtime* JavaScript criado pela Joyent. É relativamente recente, com cerca de 14 anos, e é uma das plataformas mais populares e utilizadas atualmente. Decidi utilizá-lo no desenvolvimento da minha aplicação devido à minha familiaridade prévia com JavaScript e à sua fácil aprendizagem. Além disso, o Node.js permite a criação de servidores eficientes e escaláveis, utilizando um único conjunto de ferramentas.

Figura 5 - Logótipo do Node.js

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

1.1.2.2 PostgreSQL

Para a gestão das minhas bases de dados, usei um sistema de gestão de bases de dados relacionais, o PostgresSQL. Este sistema é utilizado para armazenar, organizar e recuperar informação de forma eficiente. É bastante potente e flexível, suportando funcionalidades avançadas como transações atómicas. Foi usado devido à familiaridade, uma vez que foi previamente implementado num trabalho escolar.

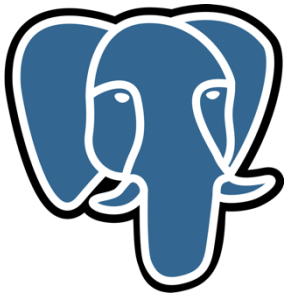


Figura 6 - Logótipo PostgreSQL

1.2 Serviços e Aplicações para o desenvolvimento

1.2.1 Design

1.2.1.1 Dribbble

Para me inspirar no design, utilizei o Dribbble, uma plataforma que reúne trabalhos de alguns dos designers mais destacados e reconhecidos a nível mundial. Esta plataforma permite que os profissionais de design partilhem amostras dos seus projetos, proporcionando uma vasta gama de estilos e abordagens criativas.

Figura 7 - Logótipo Dribble

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

1.2.1.2 Photoshop

Para transformar o meu logótipo em vetor, utilizei o Photoshop. O Photoshop é uma ferramenta de edição de imagem, que pertence ao grupo Adobe.

Figura 8 - Logótipo Photoshop

1.2.1.3 Figma

Para criar a interface da aplicação (UI), utilizei o Figma. O Figma é um editor gráfico de vetores e uma ferramenta de prototipagem de projetos. Oferece uma vasta gama de ferramentas e plugins que facilitam o processo criativo.

Figura 9 - Logótipo Figma

1.2.1.4 Shotrr

O Shotrr é uma aplicação cuja principal funcionalidade é extrair as cores presentes numa captura de ecrã, embora ofereça muitas outras funcionalidades adicionais. Utilizei esta aplicação especificamente para investigar designs que não continham a descrição do esquema de cores. O Shotrr foi essencial para identificar e analisar as paletas de cores usadas nesses designs, facilitando a minha compreensão e replicação dos estilos visuais.

Figura 10 - Logótipo Shotrr

1.2.2 Organização

1.2.2.1 Notion

Para organizar as ideias da PAP, recorri ao Notion, uma plataforma online para anotações e esquematização. O Notion oferece um conjunto abrangente de ferramentas para organização, gestão de tarefas e anotações, entre outras funcionalidades, o que foi essencial para estruturar eficientemente o meu trabalho.

Figura 11 - Logótipo Notion

1.2.2.2 GitHub

Para organizar os meus projetos de forma geral, utilizo o GitHub, um website amplamente utilizado no mundo da programação para armazenamento e arquivamento de projetos. O GitHub integra-se facilmente com outras aplicações, como editores de texto, e utiliza o sistema de controlo de versões *Git*, o que o torna uma ferramenta extremamente útil para o desenvolvimento colaborativo e a gestão eficiente do código.

Figura 12 - Logótipo GitHub

1.2.3 Hospedagem

1.2.3.1 Vercel

Para colocar o meu domínio com o site associado na internet, recorri à Vercel. A Vercel é uma plataforma de hospedagem que permite hospedar sites e aplicações web de forma eficiente, oferecendo suporte para lançamentos contínuos e integração com diversas ferramentas de desenvolvimento.

Figura 13 - Logótipo Vercel

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

1.2.3.2 Expo Go

A Expo é uma plataforma de código aberto que permite aos desenvolvedores criar aplicações universais (para todos os sistemas operativos) com JavaScript (React Native). Além disso, facilita a execução das aplicações durante o desenvolvimento, seja através de um simulador ou diretamente no telemóvel, utilizando apenas o código *QR* fornecido para cada aplicação.

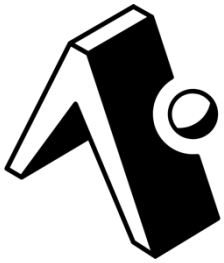


Figura 14 - Logótipo Expo Go

1.2.3.3 Docker

Para ter uma base de dados local, utilizei o Docker. O Docker é um virtualizador a nível de sistema operativo que entrega *software* ao utilizador em pacotes chamados *containers*. Estes *containers* são pacotes isolados uns dos outros, cada um contendo a sua própria configuração, bibliotecas e dependências necessárias para executar a aplicação.

Figura 15 - Logótipo Docker

1.2.4 Editor de texto

1.2.4.1 Visual Studio Code

Para programar a minha aplicação, utilizei o Visual Studio Code. Este é um editor de texto robusto que oferece uma vasta gama de ferramentas, extensões, e suporte para múltiplos protocolos, proporcionando um ambiente de desenvolvimento integrado e eficiente.

Figura 16 - Logótipo Visual Studio Code

1.2.5 Simulador

1.2.5.1 Xcode

Para simular a minha aplicação num *iPhone*, utilizei o Xcode, especificamente a funcionalidade de Simulador. O Xcode é um ambiente de desenvolvimento integrado (*IDE*) nativo da Apple, disponível apenas em computadores com macOS, e inclui um editor de texto avançado e outras ferramentas essenciais para o desenvolvimento de aplicações iOS.

Figura 17 - Logótipo Xcode

2. Projeto

2.1 Idealização

A génese da Cryptrail veio a partir de conversas, dúvidas e uma enorme paixão pelo tema.

Tudo começou quando eu tive de decidir o tema da minha PAP, que por si só, já demorou bastante tempo. Estive em dúvida sobre o que fazer; por um lado tinha aplicações mais práticas, mas que não me cativavam tanto, e por outro, o fascínio por criptomoedas. Antes de escolher definitivamente qual seria o tema, eu pensei bastante. Pensei como seria fazer uma aplicação sem ter qualquer tipo de conhecimento de uma linguagem que, efetivamente, me permitisse construir uma aplicação.

A mudança decisiva foi através de um diálogo com um amigo muito próximo. Foi algo que fez efeito como um "catalisador", se assim o posso dizer, para compreender aquilo que eu realmente devia fazer. Ele sugeriu que eu explorasse o universo das criptomoedas como tema central do meu projeto. Foi um momento tipo eureka; a ideia de me aprofundar num campo que sempre me cativou, mas que ainda estava envolto em mistério, de repente parecia não só viável, mas também possível e empolgante.

Foi assim que a Cryptrail começou a tomar forma. Queria desenvolver algo que não só saciasse a minha curiosidade e paixão pelas moedas digitais, mas que também oferecesse um valor real para outros entusiastas como eu – pessoas interessadas, investidores e curiosos em navegar com segurança por este universo. A visão era clara: uma plataforma que ajudasse na compreensão do mercado das criptomoedas, fornecendo dados precisos, análises fiáveis e um portal para as melhores oportunidades de investimento.

Cada componente da Cryptrail foi cuidadosamente pensado, repensado e aplicado, tendo dado assim o meu melhor neste projeto. Desde a escolha difícil, parecendo ou não, das tecnologias - React Native, Node.js e PostgreSQL para uma experiência intuitiva e de fácil compreensão da aplicação com uma gestão organizada da minha base de dados - até ao desenvolvimento de funcionalidades como o conversor de moeda fiduciária para cripto, integração de notícias em tempo real,

informações em tempo real. Tudo foi pensado para facilitar o acesso à informação e a tomada de decisões ao entrar neste mundo.

Olhando para trás, a criação da Cryptrail é um testemunho do meu trabalho árduo. É um lembrete de que, por vezes, as respostas para as nossas maiores questões e desafios podem surgir de onde menos esperamos – seja um interesse latente ou uma conversa com um amigo. Este projeto não é apenas a concretização de um trabalho, é também uma marca da minha jornada pessoal como programador.

2.2 Conceito

A minha aplicação, mais precisamente, consiste em ajudar as pessoas com os seus investimentos no mundo das criptomoedas.

Imaginando que um interessado por moedas digitais decide realizar a sua primeira compra, a Cryptrail terá as ferramentas necessárias para ajudar.

A primeira página da aplicação é um conversor de moedas, que permite ao utilizador realizar uma conversão de moeda fiduciária para moeda digital, ou vice-versa. Efetuada a conversão, o utilizador tem um botão "*Buy*", que quando clicado, redireciona para uma página que mostra aplicações/websites chamados de "*Exchanges*", que são sítios que permitem comprar as moedas, revelando taxas de compra e o valor total da moeda previamente selecionada. Clicando nas plataformas, o utilizador é redirecionado para o respetivo site. Mais no fundo da página, é possível ver as 6 últimas conversões feitas, tendo um botão "*See more*" que leva o utilizador para outra página onde mostra todas as conversões já feitas.

A segunda página, é o assistente "Trail", uma inteligência artificial, sendo apenas otimizado para conversar sobre assuntos de *cryptocurrency*.

A terceira página, são os mercados. Esta página consiste em mostrar ao utilizador os últimos preços do mercado, mais precisamente, moedas e *Exchanges*. Na parte das moedas, o utilizador obtém informações precisas tais como o valor do mercado, o preço e o volume nos últimos 30 dias. Como

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Exchanges, o utilizador obtém dados como o volume nas últimas 24 horas e a classificação numa escala de 0 a 10.

A quarta página é o jornal da aplicação, as notícias. Com notícias atualizadas de 24 em 24 horas, a "*News*" dá oportunidade ao utilizador de ver as últimas notícias no mundo das moedas digitais. A página apenas revela 10 notícias, sendo a primeira a notícia mais importante do dia, sendo destacada no topo da página.

Na minha aplicação, todas as páginas contêm um menu hamburger, que permite ao utilizador executar algumas funções, tais como fazer "*Logout*", ver o seu perfil com as informações colocadas no login, aceder às páginas de suporte/termos e condições do website e alguns botões que redirecionam o utilizador para as redes sociais da aplicação e para o website.

2.3 Inovação

Neste projeto, a inovação seria a integração de uma inteligência artificial focada apenas para um nicho, ajudando assim o utilizador a tomar decisões relativas aos seus investimentos.

A integração desta ferramenta em aplicações ainda é algo recente, por isso, decidi implementar como um tipo de inovação, pois pode ser realmente uma grande ajuda ter um perito na área perto de nós.

2.4 Desenvolvimento

2.4.1 Fases de desenvolvimento

O meu projeto foi dividido em 6 fases, desde o estudo do assunto, criação do conceito, realização dos esboços da aplicação, iteração da imagem da aplicação (incluindo o logótipo), escolha das linguagens para o desenvolvimento e implementação em código.

2.4.1.1 *Esboços/Ideias para a aplicação (setembro – outubro 2023)*

A partir da criação do conceito, comecei a dar vida à aplicação, iniciando com esboços tanto no papel como no computador e de seguida implementando tudo no computador.

2.4.1.1.1 Nome

Após muita pesquisa sobre o conceito da aplicação, pensei no nome "CryptoTrail", que por motivos de fluidez verbal, se tornou "Cryptrail". O nome é baseado no tema, *cryptocurrency* e na questão do caminho que as pessoas querem seguir para aprender sobre este tema, daí a origem do nome, e do slogan:

"Cryptrail, Discover your crypto journey."

2.4.1.1.2 Logótipo

A criação do logótipo, foi um processo demorado e delicado, pois está intrinsecamente ligado ao conceito de imagem que eu tenho para a aplicação.

Estes foram os primeiros resultados, feitos em papel:

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

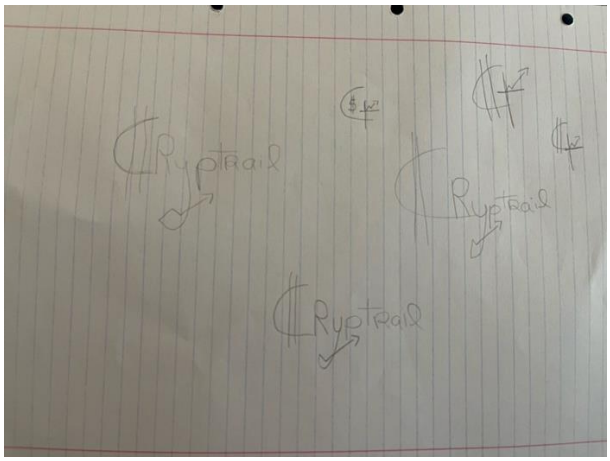


Figura 18 - Primeiros Logótipos em papel

Após a minha insatisfação com estes resultados, resolvi começar a trabalhar com o computador:

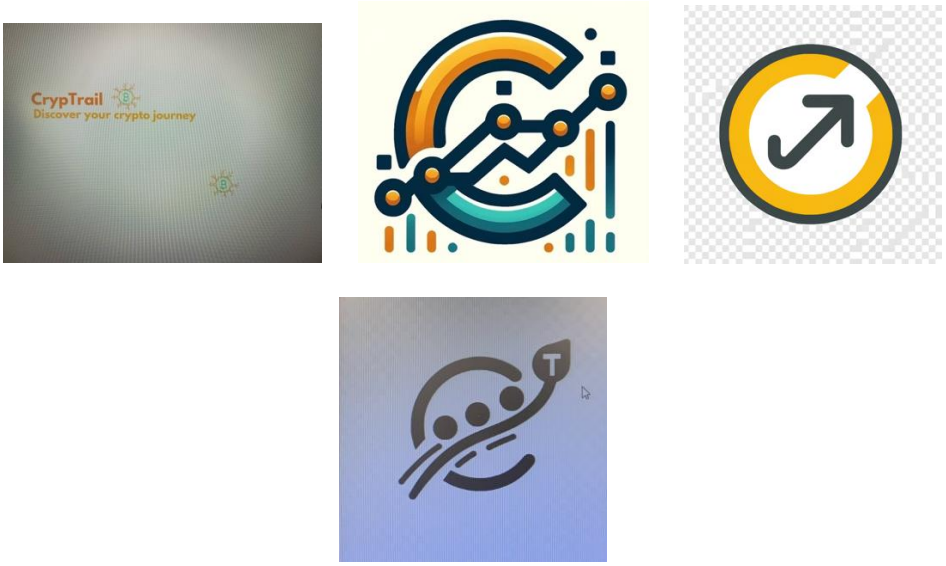


Figura 19 - Primeiros Logótipos no computador

Os Fundos Europeus mais próximos de si.

Depois de não me rever nos diferentes logótipos que fui criando, mudei de estratégia.

Assim, visto que o meu objetivo era que o logótipo representasse corretamente o nome e o slogan da aplicação optei por um logótipo minimalista, que representa a visão que eu tenho para a aplicação. Optei pela simplicidade do design para atrair os utilizadores para uma matéria tão complexa e ainda desconhecida para alguns, o mercado das moedas digitais.

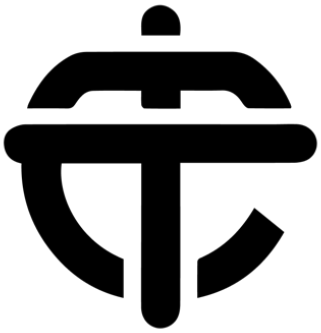


Figura 20 - Logótipo da aplicação

2.4.1.1.3 Esboços do design

Os primeiros esboços foram feitos em papel, dando assim uma primeira impressão de como seria a aplicação, chegando assim a este resultado:

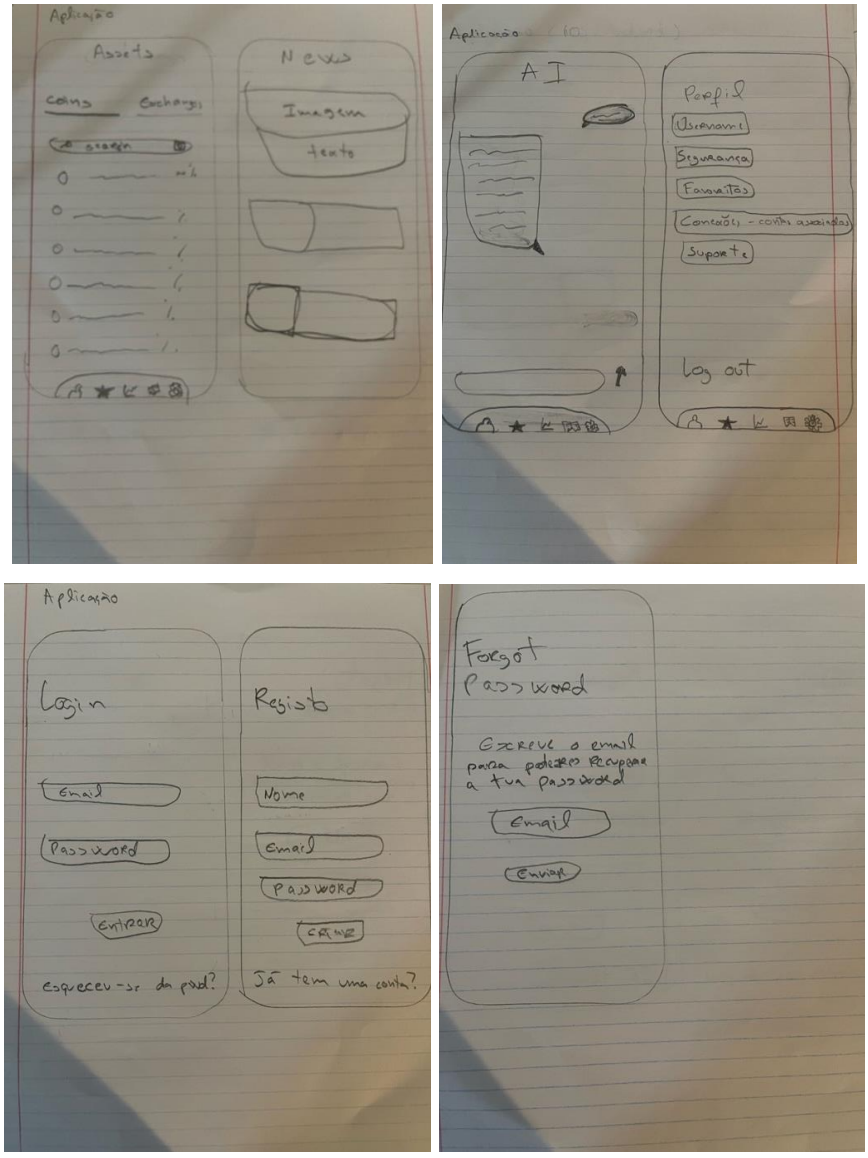


Figura 21 - Primeiros esboços do design em papel

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Tendo já feito os esboços do design em papel, decidi passá-los para o computador, focando-me inicialmente apenas no design das primeiras páginas da aplicação. Tomei esta decisão para evitar a necessidade de reformular todo o design caso não ficasse satisfeito com o resultado inicial. Assim, obtive os seguintes resultados:

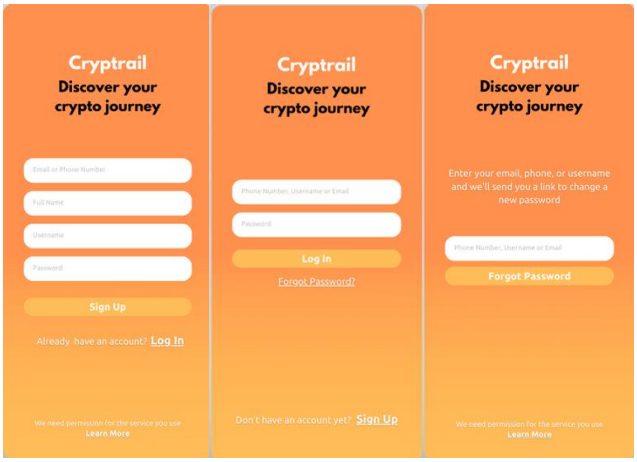


Figura 22 - Primeiros esboços do design no computador

Apesar de ter sido algo pensado e repensado várias vezes, levando sempre em consideração a simplicidade das páginas para uma melhor compreensão da aplicação, em novembro de 2023 foi finalizado o design da aplicação.

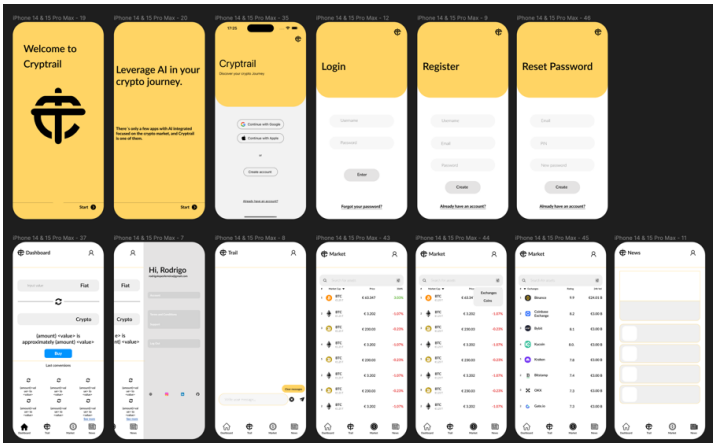
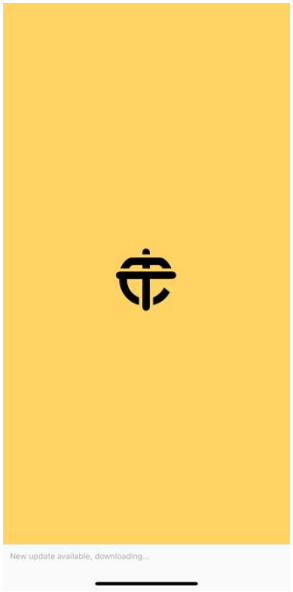


Figura 23 - Design da aplicação

2.4.1.2 *Páginas da aplicação (dezembro 2023 – maio 2024)*

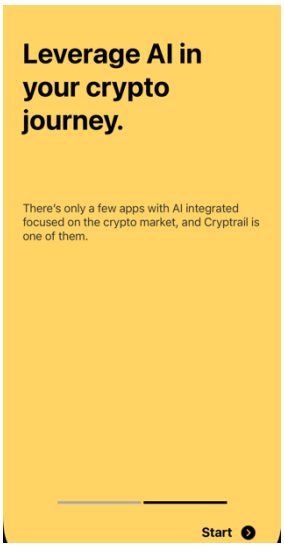


2.4.1.2.1 Ecrã de Carregamento

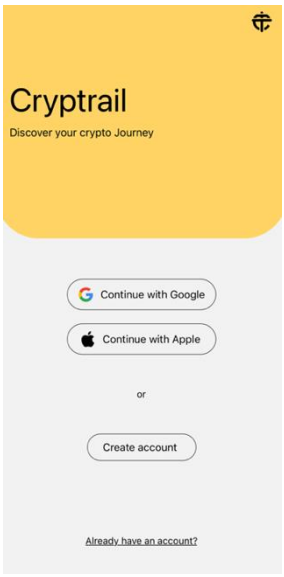
Para modernizar e dar um ar mais profissional à aplicação, decidi introduzir no ecrã de carregamento de dados, um *Splash Screen*. Consiste em mostrar o logótipo da aplicação com um fundo durante este processo.

2.4.1.2.2 Ecrã de Começo

Após o carregamento, adicionei uma página introdutória com dois slides simples para educar o utilizador. Uma breve mensagem de boas-vindas destaca a essência da aplicação. Na parte inferior direita, um botão '*Start*' leva diretamente à página de entrada após ser clicado.

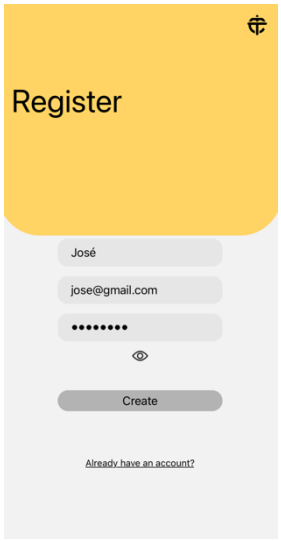


2.4.1.2.3 Página de Entrada



A página de entrada permite ao utilizador criar uma conta nova ou entrar na sua conta. Consta também com dois botões para iniciar sessão com o Google e com a Apple, que são para trabalho futuro.

2.4.1.2.4 Página de Registo

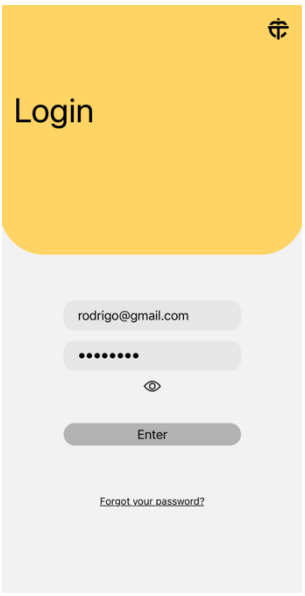


Na página de criação de conta, o utilizador encontra três campos simples para inserir o seu nome, email e palavra-passe. Um botão opcional permite alternar a visibilidade da palavra-passe por questões de privacidade. Após o registo, o utilizador é automaticamente redirecionado para a página de Login e recebe um código de confirmação. Esse código será necessário para qualquer alteração de palavra-passe no futuro. Tanto o código de confirmação quanto a palavra-passe são protegidos por criptografia na base de dados, garantindo a segurança dos dados do utilizador. No rodapé da página, encontra-se a opção "*Already have an account?*", que ao ser clicada redireciona o utilizador para a página de login.

Os Fundos Europeus mais próximos de si.

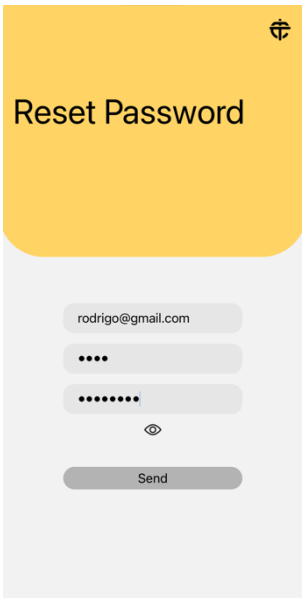
2.4.1.2.5 Página de Login

Aqui, o utilizador pode fazer login na sua conta. No rodapé da página, há a opção "*Forgot your password?*", que ao ser clicada, direciona o utilizador para a página de recuperação da palavra-passe.



2.4.1.2.6 Página de Recuperação da Palavra-passe

Uma funcionalidade fundamental em todas as aplicações é a página de recuperação de palavra-passe, e por isso, decidi incluí-la. Atualmente, o processo de recuperação envolve a submissão do pin do registo, mas numa próxima atualização, planeio enviar o pin por email por questões de segurança. Após a conclusão da recuperação, o utilizador é redirecionado para a página de login.

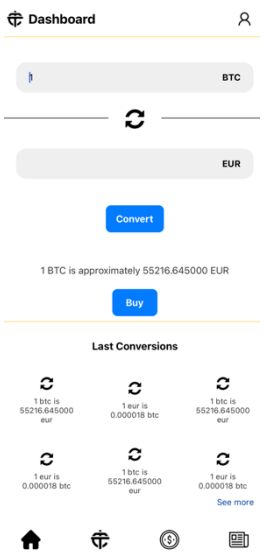


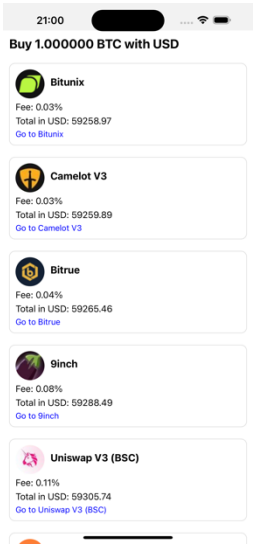
2.4.1.2.7 Página de Conversões

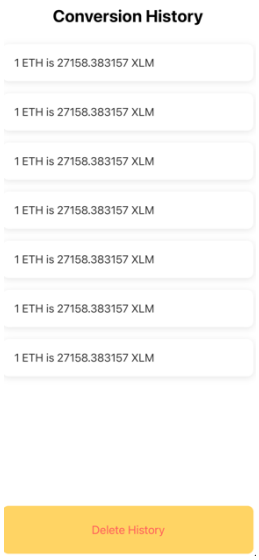
Dado como finalizado o processo de registo ou de login, o utilizador finalmente chega à página mais importante da aplicação, a "*Dashboard*". Nesta página, o utilizador pode efetuar várias conversões entre moedas fiduciárias e moedas digitais (mais de 3000), sendo todas as conversões feitas armazenadas na base de dados.

Após introduzir um valor e escolher a conversão, irão aparecer ao utilizador dois botões:

- Um deles diz "*Convert*", que quando carregado, efetua a conversão. Quando são feitas mais de 6 conversões, aparece no canto inferior direito da página uma opção "*See more*". Carregando, leva o utilizador para uma página com o histórico de todas as conversões que já fez, tendo a opção de as apagar.
- O outro botão, "*Buy*", é apenas revelado após o utilizador fazer a primeira conversão levando-o assim para uma página onde pode ver múltiplos sites de compra de moedas digitais, que revelam o preço da compra devido à taxa que cobra.







2.4.1.2.8 Página do Assistente Virtual (Trail)

Outro componente muito importante da aplicação é a página do assistente virtual (*Trail*), em que é aplicado o uso de inteligência artificial.

Trata-se de um centro de mensagens poderosíssimo onde o utilizador pode fazer perguntas sobre o mercado de moedas digitais, o que é bastante útil e completamente inovador. Foram concebidas algumas perguntas básicas, que servem de exemplos de questões a colocar.

Esta página é composta por uma caixa de texto, botões de enviar e apagar mensagens e um pequeno botão para apagar o texto já escrito de uma vez só.

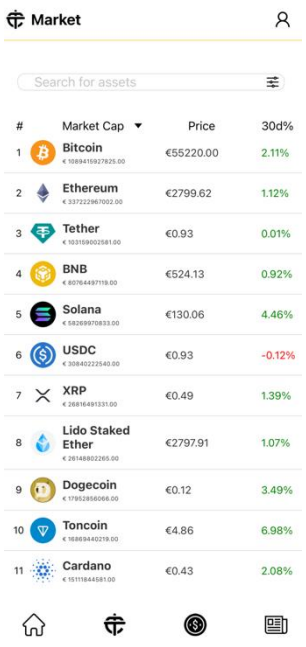
As conversas são guardadas na base de dados, tendo o utilizador sempre acesso ao histórico de conversas, tendo a opção de o eliminar.

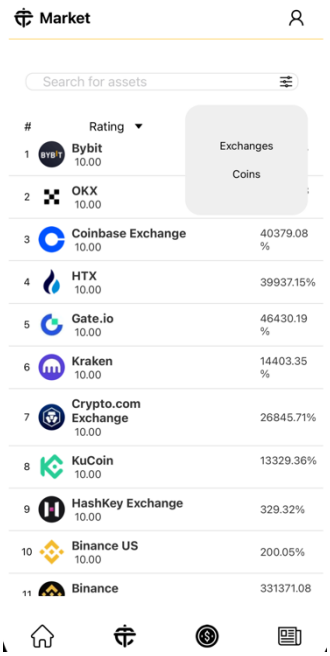
2.4.1.2.9 Página do Mercado

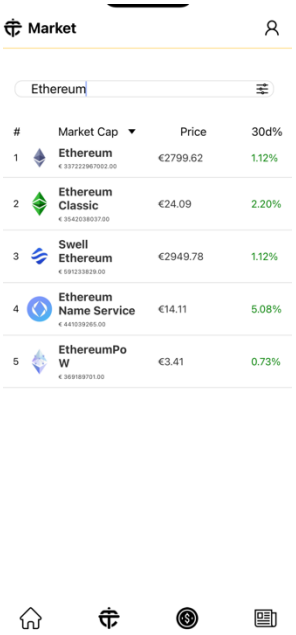
A página de mercado serve para fornecer ao utilizador, informações em tempo real sobre as diferentes componentes do mercado, tais como:

- Moedas, preço e o valor de mercado;
- *Exchanges* e avaliação correspondente;
- Volume das últimas 24 horas e variação;

Inicialmente é apresentada a lista de moedas. Para ver a lista de *Exchanges*, o utilizador clica no botão de filtrar, onde pode selecionar entre "*Coins*" e "*Exchanges*". Também é possível fazer uma pesquisa pelo nome.

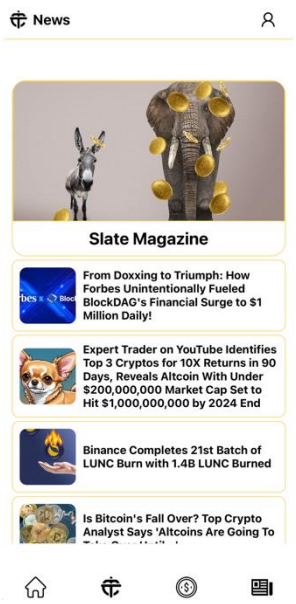






2.4.1.2.10 Página de Notícias

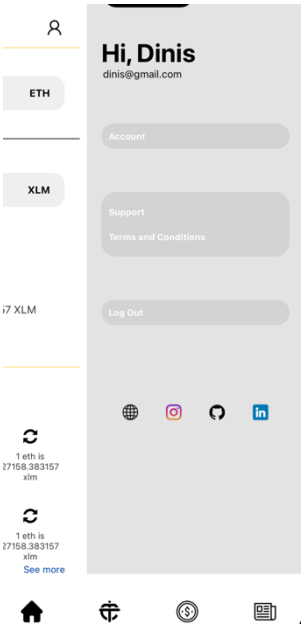
Nesta página, o utilizador tem acesso às 10 mais relevantes notícias do dia sobre *cryptocurrency*. São apresentadas com uma imagem e um título e, caso o utilizador queira ver a notícia toda, pode carregar na notícia e será redirecionado para fonte.



2.4.1.2.11 Menu do utilizador

O menu de utilizador foi criado com o intuito de facilitar o acesso à sua informação pessoal. A partir daqui consegue também aceder ao website da aplicação e às redes sociais.

Este menu encontra-se presente em todas as páginas.



2.4.1.2.12 Página de informações da conta

Nesta página, é possível para o utilizador ver as suas informações, tais como o nome e email.

No fundo da página, tem um botão que quando carregado leva para a página de recuperação da palavra-passe.

Futuramente, irei iterar sobre esta página, introduzindo funcionalidades tais como:

- Tema (claro/escuro);
- Língua;
- Moeda;
- Opções de publicidade.

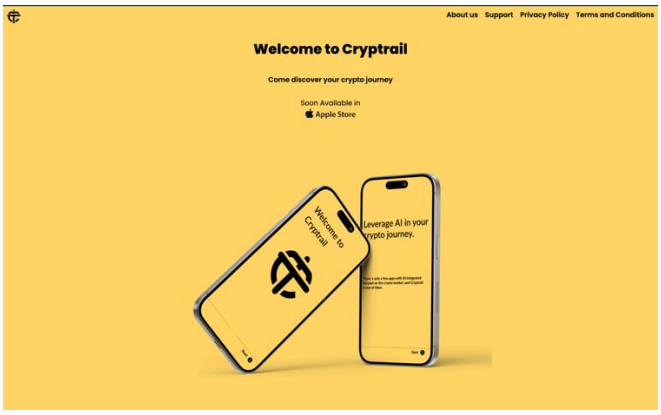
2.4.1.3 *Páginas do Website (maio 2024)*

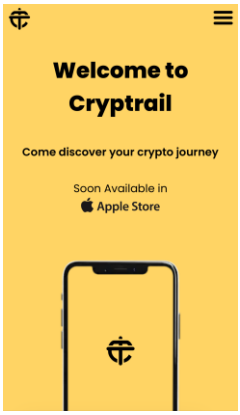
Um website institucional é um veículo de informação e um canal essencial de publicidade para qualquer aplicação.

Usando o conhecimento adquirido ao longo do curso, tornou-se trivial a execução de um website responsivo com o objetivo de publicitar a minha aplicação.

2.4.1.3.1 Página principal

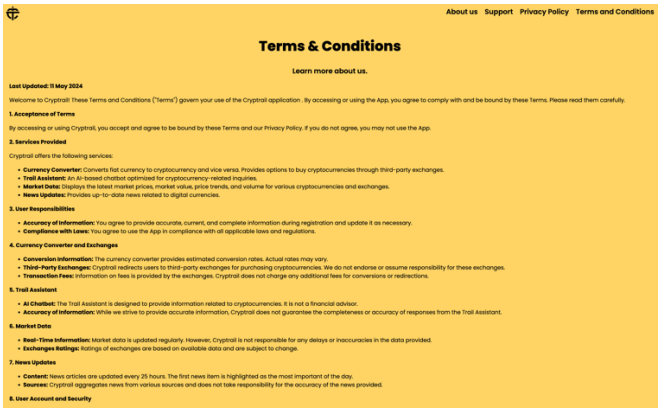
Esta página revela uma imagem alusiva à aplicação, no modelo de computador. Nos dispositivos móveis, apresenta um *mockup* de um iPhone com o logótipo centrado.

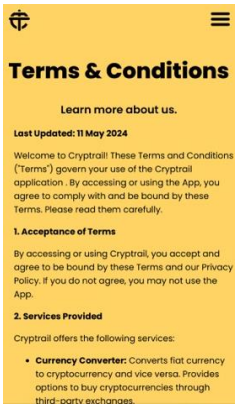




2.4.1.3.2 Página de termos e condições

Nesta página, o utilizador tem acesso aos termos e condições da aplicação.

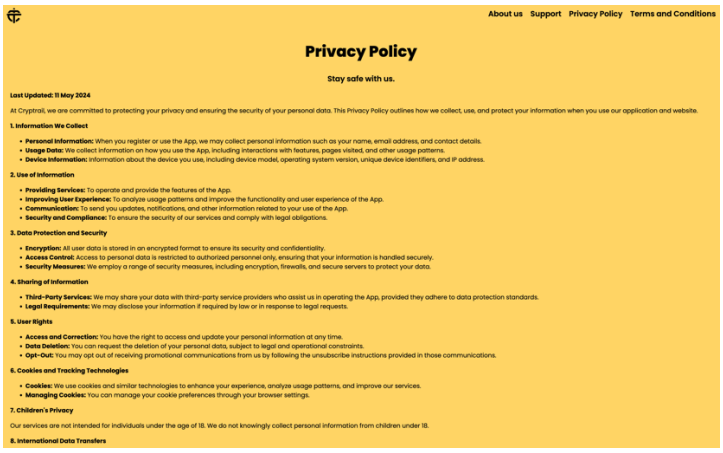




PESSOAS 2030
PORTUGAL 2030
Cofinanciado pela União Europeia
SELO DE CONFORMIDADE EQAVET
uma escola Rumos education | Knowledge sharing
Os Fundos Europeus mais próximos de si.

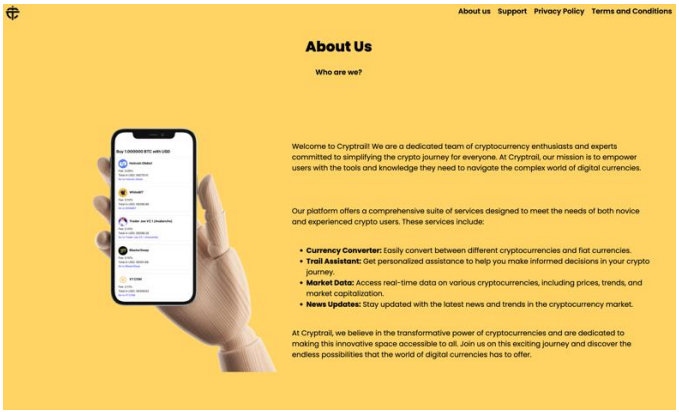
2.4.1.3.3 Página de Políticas de privacidade, proteção de dados e requisitos

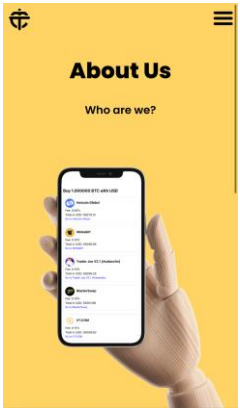
Nesta página, o utilizador tem acesso às políticas de privacidade, de proteção de dados e requisitos mínimos de utilização da aplicação.



2.4.1.3.4 Página Sobre nós

Esta é uma página que mostra uma imagem de uma página da aplicação, falando depois sobre o que faz a aplicação.

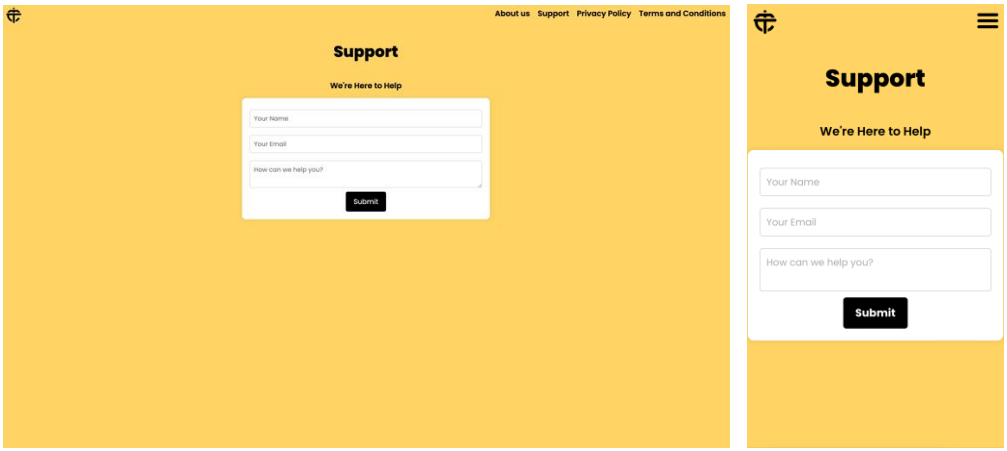




PESSOAS 2030 PROGRAMA OPERACIONAL INCLUSÃO E INCLUSÃO
PORTUGAL 2030
Cofinanciado pela União Europeia
SELO DE CONFORMIDADE EQAVET
uma escola Rumos education | Knowledge sharing
Os Fundos Europeus mais próximos de si.

2.4.1.3.5 Página de Suporte

Nesta página, o utilizador pode enviar um email para o suporte, caso tenha alguma dúvida.



2.4.2 Problemas / Dificuldades

Ao longo do desenvolvimento deste projeto, deparei-me com várias dificuldades.

A escolha do tema para a minha aplicação foi particularmente desafiante. Demorei bastante tempo a refletir, pesquisar e estudar diversas opções. No entanto, após uma análise aprofundada, todo o esforço revelou-se frutífero, transformando uma pequena ideia num projeto de grande envergadura.

A seleção das tecnologias a utilizar foi uma das decisões mais complexas. Desejava destacar-me dos meus colegas, o que me levou a optar por tecnologias recentes e inovadoras. Por exemplo, a escolha de utilizar uma base de dados local foi pensada para aproveitar os conhecimentos adquiridos nas aulas com o professor Américo Rodrigues. Além disso, a implementação de uma aplicação de virtualização de ambientes acrescentou complexidade, mas também proporcionou um ambiente de desenvolvimento mais robusto. Quero deixar o meu agradecimento aos professores da área técnica por, não me colocando entraves, me apoiarem sempre nas minhas escolhas.

Conciliar as exigências académicas da escola com a realização deste projeto foi especialmente complicado. Lidar com tarefas escolares exigentes enquanto me dedicava à minha prova de aptidão

profissional exigiu uma gestão bastante cuidadosa do meu tempo, pois ao mesmo tempo que estava a tentar fazer com que as minhas notas na escola subissem ao máximo, estava também a estudar para os exames nacionais e a concretizar este projeto.

Apesar de todas as dificuldades enfrentadas ao longo deste processo, cada obstáculo superado representou uma importante aprendizagem e contribuiu para o meu crescimento pessoal e profissional.

3. Código

3.1 Aplicação

3.1.1 Front-end

3.1.1.1 *App.js*

```
//App.js
import * as React from "react";
import { Image, Text } from "react-native";
import { UserProvider } from "./UserContext";
import { responsiveScreenWidth, responsiveScreenHeight } from 'react-native-responsive-dimensions';
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import GetStartedScreen from "./components/GetStartedScreen";
import LandingScreen from "./components/LandingScreen";
import LoginScreen from "./components/LoginScreen";
import RegisterScreen from "./components/RegisterScreen";
import ForgotPasswordScreen from "./components/ForgotPasswordScreen";
import DashboardScreen from "./components/DashboardScreen";
import TrailScreen from "./components/TrailScreen";
import MarketScreen from "./components/MarketScreen";
import NewsScreen from "./components/NewsScreen";
import BuyPlacesScreen from "./components/BuyPlacesScreen";
import FullHistory from "./components/FullHistory";
import AccountScreen from "./components/AccountScreen";
```

```
const linking = {
  prefixes: ['exp://192.168.1.70:8081'],
  config: {
    screens: {
      GetStartedScreen: 'getstarted',
      Landing: 'landing',
      Signup: 'signup',
      Login: 'login',
      Forgot: 'forgot',
      Dashboard: 'dashboard',
      Trail: 'trail',
      Market: 'market',
      News: 'news',
      BuyPlaces: 'buy places',
      FullHistory: 'full history',
      Account: 'account',
    },
  },
};
const Stack = createNativeStackNavigator();
const Tab = createBottomTabNavigator();

function ShowBottomTabs() {
  return (
    <Tab.Navigator
  screenOptions={{
    zIndex: 0,
    backgroundColor: '#fff',
    headerShown: false,
    tabBarShowLabel: false,
    tabBarStyle: {
      backgroundColor: '#fff',
      position: 'absolute',
      paddingBottom: 10,
      elevation: 0,
      borderTopWidth: 0,
      height: 100,
      alignItems: 'center',
      justifyContent: 'center',
      paddingTop: 15,
    },
    tabBarItemStyle: {
```

```
      backgroundColor: '#fff',
      width: '100%',
      justifyContent: 'center',
      alignItems: 'center',
      padding: 0,
    },
  }}
>
      <Tab.Screen
        name='Trail'
        component={DashboardScreen}
        options={{
          tabBarIcon: ({focused}) => (
            <Image
              source={focused ? require('./assets/home2.png') : require('./assets/home1.png')}
              resizeMode='contain'
              style={{
                width: responsiveScreenWidth(7),
                height: responsiveScreenHeight(4),
              }}
            />
          ),
        }}
      />
      <Tab.Screen
        name='Assistant'
        component={TrailScreen}
        options={{
          tabBarIcon: ({focused}) => (
            <Image
              source={require('./assets/cryptrail.png')}
              resizeMode='contain'
              style={{
                width: responsiveScreenWidth(8),
                height: responsiveScreenHeight(5),
              }}
            />
          ),
        }}
      />
      <Tab.Screen
        name='Market'
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
          component={MarketScreen}
          options={{
            tabBarIcon: ({focused}) => (
              <Image
                source={focused ? require('./assets/coin2.png') : require('./assets/coin1.png')}
                resizeMode='contain'
                style={{
                  width: responsiveScreenWidth(7),
                  height: responsiveScreenHeight(4),
                }}
              />
            ),
          }}
        />
        <Tab.Screen
          name='News'
          component={NewsScreen}
          options={{
            tabBarIcon: ({focused}) => (
              <Image
              source={focused ? require('./assets/news2.png') : require('./assets/news1.png')}
                resizeMode='contain'
                style={{
                  width: responsiveScreenWidth(7),
                  height: responsiveScreenHeight(4),
                }}
              />
            ),
          }}
        />
      </Tab.Navigator>
    );
}

export default function App() {
  return (
    <UserProvider>
    <NavigationContainer linking={linking} fallback={<Text>Loading...</Text>}>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
      <Stack.Screen name="GetStarted" component={GetStartedScreen} />
        <Stack.Screen name="Landing" component={LandingScreen} options={{gestureEnabled: false }}/>
        <Stack.Screen name="Signup" component={RegisterScreen} />
```

```
      <Stack.Screen name="Login" component={LoginScreen} />
      <Stack.Screen name="Forgot" component={ForgotPasswordScreen} />
      <Stack.Screen name="Dashboard0" component={ShowBottomTabs} options={{gestureEnabled: false }}
/>
      <Stack.Screen name="Trail0" component={ShowBottomTabs} options={{gestureEnabled: false }} />
      <Stack.Screen name="Market0" component={ShowBottomTabs} options={{gestureEnabled: false }} />
      <Stack.Screen name="News0" component={ShowBottomTabs} options={{gestureEnabled: false }}/>
      <Stack.Screen name="BuyPlaces" component={BuyPlacesScreen} />
      <Stack.Screen name="FullHistory" component={FullHistory} />
      <Stack.Screen name="Account" component={AccountScreen} />
    </Stack.Navigator>
   </NavigationContainer>
   </UserProvider>
  );
}
```

3.1.1.2 *UserContext.js*

```
//UserContext.js
import React, { createContext, useContext, useState } from "react";

const UserContext = createContext(null);

export const UserProvider = ({ children }) => {
  const [user, setUser] = useState(null);

  return (
    <UserContext.Provider value={{ user, setUser }}>
      {children}
    </UserContext.Provider>
  );
};

export const useUser = () => useContext(UserContext);
```

3.1.1.3 *GetStartedScreen.js*

```
//GetStartedScreen.js
import React, { useState } from "react";
import { View, Image, Text, StyleSheet, ScrollView, Dimensions, TouchableOpacity } from "react-native";
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
const { width } = Dimensions.get("window");

const PaginationIndicator = ({ totalPages, currentPage }) => {
  return (
    <View style={styles.paginationWrapper}>
      {Array.from({ length: totalPages }).map((_, index) => (
        <View
          key={index}
          style={[
            styles.paginationLine,
            currentPage === index && styles.paginationLineActive,
          ]}
        />
      ))}
    </View>
  );
};

const GetStartedScreen = ({ navigation }) => {
  const [currentPage, setCurrentPage] = useState(0);

  const handleScroll = (event) => {
    const { x } = event.nativeEvent.contentOffset;
    const indexOfNextScreen = Math.round(x / width);
    setCurrentPage(indexOfNextScreen);
  };

  return (
    <View style={styles.container}>
      <ScrollView
        horizontal={true}
        pagingEnabled={true}
        showsHorizontalScrollIndicator={false}
        onScroll={handleScroll}
        scrollEventThrottle={16}
        style={styles.scrollView}
      >
        <View style={[styles.slide, { width: width }]}>
          <View style={styles.alignCenter}>
            <Text style={styles.title}>Welcome to</Text>
            <Text style={styles.stitle}>Cryptrail</Text>
            <Image
```

```
            source={require("../assets/adaptive-icon.png")}
            style={styles.logo}
           />
          </View>
        </View>
        <View style={[styles.slide, { width: width }]}>
         <View style={styles.alignCenter}>
          <Text style={styles.title}>
           Leverage AI in your crypto journey.
          </Text>
          <Text style={styles.description}>
           There's only a few apps with AI integrated focused on the crypto
           market, and Cryptrail is one of them.
          </Text>
          <View style={{ flexDirection: 'row', paddingHorizontal: 20, marginTop: 425, alignSelf: 'flex-end' }}>
           <TouchableOpacity onPress={() => navigation.navigate("Landing")}>
            <Text style={styles.nextT}>Start</Text>
            <Image
             source={require("../assets/next.png")}
             style={styles.next}
            />
           </TouchableOpacity>
          </View>
         </View>
        </View>
       </ScrollView>
       <PaginationIndicator totalPages={2} currentPage={currentPage} />
      </View>
    );
};

const styles = StyleSheet.create({
  scrollView: {
    backgroundColor: "#FFD464",
  },
  container: {
    flex: 1,
  },
  slide: {
    flex: 1,
    padding: 20,
  },
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
title: {
  marginTop: 100,
  fontSize: 43,
  fontWeight: "bold",
  paddingHorizontal: 10,
},
stitle: {
  fontSize: 43,
  fontWeight: "bold",
  paddingHorizontal: 10,
  marginTop: 15,
},
logo: {
  marginTop: 50,
  width: 350,
  height: 350,
  marginBottom: 20,
  alignSelf: "center",
},
description: {
  fontSize: 18,
  color: "#333",
  marginTop: 100,
  paddingHorizontal: 10,
},
next: {
  width: 22,
  height: 22,
  marginTop: 2,
  alignSelf: "flex-end",
},
nextT: {
  fontSize: 20,
  fontWeight: "bold",
  alignSelf: "flex-end",
  marginRight: 36,
  marginTop: 10,
  marginBottom: -24,
},
paginationWrapper: {
  position: "absolute",
  bottom: 90,
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    left: 0,
    right: 0,
    flexDirection: "row",
    justifyContent: "center",
  },
  paginationLine: {
    width: width * 0.3,
    height: 4,
    backgroundColor: "#a9a9a9",
    marginHorizontal: 2,
  },
  paginationLineActive: {
    backgroundColor: "#000",
  },
  container: {
    flex: 1,
    backgroundColor: "#FFD464",
  },
});

export default GetStartedScreen;
```

3.1.1.4 *LandingScreen.js*

```
//LandingScreen.js
import React from 'react';
import { StyleSheet, Text, TouchableOpacity, View, Image } from 'react-native';
import { responsiveScreenHeight } from 'react-native-responsive-dimensions';


const HomeScreen = ({ navigation }) => {
  return (
    <View style={styles.container}>
      <View style={styles.yellowBackground}>
    <Image source={require('../assets/cryptrail.png')} style={styles.cryptrail}/>
    <View style={styles.begin}>
        <Text style={styles.title}>Cryptrail</Text>
        <Text style={styles.desc}>Discover your crypto Journey</Text>
    </View>
      </View>

      <View style={styles.buttons}>
```

```
      <TouchableOpacity style={styles.buttonBox}>
      <Image source={require('../assets/glogo.png')} style={styles.glogo} />
        <Text style={styles.buttonText}>Continue with Google</Text>
      </TouchableOpacity>

      <TouchableOpacity style={styles.buttonBox}>
      <Image source={require('../assets/apple.png')} style={styles.glogo} />
        <Text style={styles.buttonText}>Continue with Apple </Text>
      </TouchableOpacity>

      <Text style={styles.or}>or</Text>

      <TouchableOpacity style={styles.log} onPress={() => navigation.navigate("Signup")}>
        <Text style={styles.buttonText}>Create account</Text>
      </TouchableOpacity>

      <View>
      <TouchableOpacity onPress={() => navigation.navigate("Login")}>
        <Text style={styles.underlinedText}>Already have an account?</Text>
      </TouchableOpacity>
      </View>
    </View>
  </View>
 );
};

const styles = StyleSheet.create({
 container: {
  flex: 1,
  justifyContent: 'flex-start',
 },
 yellowBackground: {
  backgroundColor: '#FFD464',
  width: '100%',
  height: '43%',
  justifyContent: 'flex-end',
  borderBottomLeftRadius: 63,
  borderBottomRightRadius: 63,
  paddingHorizontal: 20,
  paddingBottom: 20,
 },
 cryptrail: {
```

```
  height: responsiveScreenHeight(3.5),
  width: responsiveScreenHeight(80),
  resizeMode: 'contain',
  marginBottom: 70,
},
begin: {
  marginBottom: 127,
},
title: {
  fontSize: 48,
  color: '#000',
},
desc: {
  fontSize: 16,
  color: '#000',
  marginTop: 10,
},
buttons: {
  alignItems: 'center',
  marginTop: 50,
},
buttonBox: {
  flexDirection: 'row',
  borderWidth: 0.5,
  borderRadius: 25,
  paddingVertical: 10,
  paddingHorizontal: 20,
  marginVertical: 10,
  width: 220,
  alignItems: 'center',
  justifyContent: 'center',
},
buttonText: {
  fontSize: 15.8,
  color: '#000000',
  marginLeft: 10,
},
icon: {
  fontSize: 24,
  color: '#000000',
},
glogo: {
```

```
    width: 23.66,
    height: 25,
    resizeMode: 'contain',
  },
  log: {
    flexDirection: 'row',
    borderWidth: 0.5,
    borderRadius: 25,
    paddingVertical: 10,
    paddingHorizontal: 12.5,
    marginVertical: 20,
    width: 161,
    textAlign: 'center',
    marginTop: 50,
  },
  or: {
    marginTop: 40,
  },

  underlinedText: {
    marginTop: 90,
    textDecorationLine: 'underline',
    textAlign: 'center',
  },
});

export default HomeScreen;
```

3.1.1.5 *RegisterScreen.js*

```
//RegisterScreen.js
import React, { useState, useEffect } from 'react';
import { StyleSheet, Text, TouchableOpacity, View, TextInput, TouchableWithoutFeedback, Image, Keyboard,
SafeAreaView, Alert } from 'react-native';
import axios from 'axios';
import { responsiveScreenHeight } from 'react-native-responsive-dimensions';
import AsyncStorage from '@react-native-async-storage/async-storage';
```

```
const SignupScreen = ({ navigation }) => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [errors, setErrors] = useState({});
  const [isFormValid, setIsFormValid] = useState(false);
  const [showPassword, setShowPassword] = useState(false);
  const [isClicked, setIsClicked] = useState(false);

  useEffect(() => {
    validateForm();
  }, [name, email, password]);

  const validateForm = () => {
    let errors = {};

    if (!name) {
      errors.name = 'Name is required.';
    }

    if (!email) {
      errors.email = 'Email is required.';
    } else if (!/\S+@\S+\.\S+/.test(email)) {
      errors.email = 'Email is invalid.';
    }

    if (!password) {
      errors.password = 'Password is required.';
    } else if (password.length < 8) {
      errors.password = 'Password must be at least 8 characters.';
    }

    setErrors(errors);
    setIsFormValid(Object.keys(errors).length === 0);
  };

  const sendToReg = async () => {
    try {
      const response = await axios.post('http://192.168.1.70:3000/register', {
        username: name,
        email: email,
```

```
        password: password
      });

      const { userId, pin } = response.data;
      await AsyncStorage.setItem('userId', JSON.stringify(userId));
      await AsyncStorage.setItem('username', name);
      await AsyncStorage.setItem('pin', pin);

      Alert.alert('Registration Successful', `Your PIN is ${pin}. Please keep it safe, to change your password, you need this pin.`);

      navigation.navigate("Login");
    } catch (error) {
      console.error('Error during registration:', error);
      Alert.alert('Registration Error', 'There was an issue with your registration. Please try again.');
    }
  };

  const toggleShowPassword = () => {
    setShowPassword(!showPassword);
    setIsClicked((prevState) => !prevState);
  };


  return (
    <TouchableWithoutFeedback onPress={Keyboard.dismiss}>
      <View style={styles.container}>
        <View style={styles.yellowBackground}>
          <Image source={require('../assets/cryptrail.png')} style={styles.cryptrail} />
          <Text style={styles.title}>Register</Text>
        </View>
        <View style={styles.write}>
          <SafeAreaView>
            <TextInput
              style={styles.placeholder}
              placeholder="Name"
              value={name}
              onChangeText={setName}
            />
            <TextInput
              style={styles.placeholder}
              placeholder="Email"
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    value={email}
    onChangeText={(text) => setEmail(text.toLowerCase())}
    autoCapitalize="none"
   />
   <TextInput
    secureTextEntry={!showPassword}
    value={password}
    onChangeText={setPassword}
    style={styles.placeholder}
    placeholder="Password"
    placeholderTextColor="#aaa"
   />
   <TouchableOpacity
    name={showPassword}
    size={24}
    color="#aaa"
    onPress={toggleShowPassword}
   >
    <Image
     style={styles.icon}
     source={
      isClicked
       ? require("../assets/hide.png")
       : require("../assets/view.png")
     }
    />
   </TouchableOpacity>
   <View style={styles.button}>
    <TouchableOpacity
     style={[styles.buttonBox, { opacity: isFormValid ? 1 : 0.5 }]}
     disabled={!isFormValid}
     onPress={sendToReg}
    >
     <Text style={styles.buttonText}>Create</Text>
    </TouchableOpacity>
   </View>
  </SafeAreaView>
  {Object.values(errors).map((error, index) => (
   <Text key={index} style={styles.error}>
    {error}
   </Text>
  ))}
```

```
        <TouchableOpacity onPress={() => navigation.navigate("Login")}>
          <Text style={styles.underlinedText}>Already have an account?</Text>
        </TouchableOpacity>
      </View>
    </View>
  </TouchableWithoutFeedback>
 );
};

const styles = StyleSheet.create({
 container: {
  flex: 1,
  justifyContent: 'flex-start',
 },
 yellowBackground: {
  backgroundColor: '#FFD464',
  width: '100%',
  height: '43%',
  justifyContent: 'center',
  alignItems: 'flex-start',
  borderBottomLeftRadius: 63,
  borderBottomRightRadius: 63,
  paddingHorizontal: 20,
 },
 cryptrail: {
  height: responsiveScreenHeight(3.5),
  width: responsiveScreenHeight(80),
  resizeMode: 'contain',
  marginBottom: 69,
 },
 title: {
  fontSize: 48,
  color: '#000',
  marginBottom: 109,
 },
 button: {
  alignItems: 'center',
  marginTop: 30,
 },
 buttonBox: {
  borderRadius: 25,
  paddingVertical: 5,
```

```
    paddingHorizontal: 20,
    marginVertical: 10,
    width: 250,
    alignItems: 'center',
    backgroundColor: '#B3B3B3',
  },
  icon: {
    alignSelf: "center",
    width: 24,
    height: 24,
  },
  write: {
    marginTop: 60,
    alignItems: 'center',
  },
  placeholder: {
    borderRadius: 15,
    paddingVertical: 10,
    paddingHorizontal: 20,
    marginVertical: 10,
    width: 250,
    alignItems: 'center',
    fontSize: 17,
    backgroundColor: '#E6E6E6',
    marginTop: 5,
    fontSize: 18,
    color: '#000',
  },
  user: {
    marginTop: 20,
    fontSize: 18,
    alignItems: 'flex-start',
  },
  create: {
    marginTop: 20,
    fontSize: 18,
  },
  buttonText: {
    fontSize: 18,
    textAlign: 'center',
  },
  error: {
```

```
    color: 'red',
    fontSize: 11,
  },
  underlinedText: {
    marginTop: 60,
    textAlign: 'center',
    textDecorationLine: 'underline',
  },
});

export default SignupScreen;
```

3.1.1.6 *LoginScreen.js*

```
//LoginScreen.js
import React, { useState, useEffect } from "react";
import {
  Alert,
  StyleSheet,
  Text,
  TouchableOpacity,
  View,
  TextInput,
  Image,
  TouchableWithoutFeedback,
  Keyboard,
  SafeAreaView,
} from "react-native";
import axios from "axios";
import AsyncStorage from "@react-native-async-storage/async-storage";
import { MaterialCommunityIcons } from "@expo/vector-icons";
import {
  responsiveFontSize,
  responsiveHeight,
  responsiveWidth,
  responsiveScreenWidth,
  responsiveScreenHeight,
  responsiveScreenFontSize,
} from "react-native-responsive-dimensions";
import { useUser } from "../UserContext";
```

```
const SigninScreen = ({ navigation }) => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const { setUser } = useUser();
  const [isClicked, setIsClicked] = useState(false);

  const handleLogin = async () => {
    try {
      const response = await axios.post("http://192.168.1.70:3000/login", {
        email,
        password,
      });
      if (response.data.user) {
        const userData = response.data.user;


        await AsyncStorage.setItem("userId", JSON.stringify(userData.id));
        await AsyncStorage.setItem("username", userData.username);
        await AsyncStorage.setItem("email", userData.email);
        await AsyncStorage.setItem("password", password);

        setUser(userData);
        navigation.navigate("Dashboard0");
      }
    } catch (error) {
      Alert.alert(
        "Login Failed",
        "Invalid email or password. Please try again."
      );
    }
  };


  const [errors, setErrors] = useState({});
  const [isFormValid, setIsFormValid] = useState(false);

  useEffect(() => {
    validateForm();
  }, [email, password]);

  const validateForm = () => {
    let errors = {};
```

```
  if (!email) {
    errors.email = "Email is required.";
  } else if (!/\S+@\S+\.\S+/.test(email)) {
    errors.email = "Email is invalid.";
  }


  if (!password) {
    errors.password = "Password is required.";
  } else if (password.length < 8) {
    errors.password = "Password must be at least 8 characters.";
  }


  setErrors(errors);
  setIsFormValid(Object.keys(errors).length === 0);
};


const [showPassword, setShowPassword] = useState(false);

const toggleShowPassword = () => {
  setShowPassword(!showPassword);
  setIsClicked((prevState) => !prevState);
};

return (
  <TouchableWithoutFeedback onPress={Keyboard.dismiss}>
    <View style={styles.container}>
      <View style={styles.yellowBackground}>
        <Image
          source={require("../assets/cryptrail.png")}
          style={styles.cryptrail}
        />
        <Text style={styles.title}>Login</Text>
      </View>
      <View style={styles.write}>
        <SafeAreaView>
          <TextInput
            style={styles.placeholder}
            placeholder="Email"
```

```
    value={email}
    onChangeText={(text) => setEmail(text.toLowerCase())}
    autoCapitalize="none"
   />
   <TextInput
    secureTextEntry={!showPassword}
    value={password}
    onChangeText={setPassword}
    style={styles.placeholder}
    placeholder="Password"
    placeholderTextColor="#aaa"
   />
   <TouchableOpacity
    name={showPassword}
    size={24}
    color="#aaa"
    onPress={toggleShowPassword}
   >
    <Image
     style={styles.icon}
     source={
      isClicked
       ? require("../assets/hide.png")
       : require("../assets/view.png")
     }
    />
   </TouchableOpacity>
   <View style={styles.button}>
    <TouchableOpacity
     style={[styles.buttonBox, { opacity: isFormValid ? 1 : 0.5 }]}
     disabled={!isFormValid}
     onPress={handleLogin}
    >
     <Text style={styles.buttonText}>Enter</Text>
    </TouchableOpacity>
   </View>
  </SafeAreaView>
  {Object.values(errors).map((error, index) => (
   <Text key={index} style={styles.error}>
    {error}
   </Text>
  ))}
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
        <TouchableOpacity onPress={() => navigation.navigate("Forgot")}>
          <Text style={styles.underlinedText}>Forgot your password?</Text>
        </TouchableOpacity>
      </View>
    </View>
  </TouchableWithoutFeedback>
 );
};

const styles = StyleSheet.create({
 container: {
  flex: 1,
  justifyContent: "flex-start",
 },
 yellowBackground: {
  backgroundColor: "#FFD464",
  width: "100%",
  height: "43%",
  justifyContent: "center",
  alignItems: "flex-start",
  borderBottomLeftRadius: 63,
  borderBottomRightRadius: 63,
  paddingHorizontal: 20,
 },
 cryptrail: {
  height: responsiveScreenHeight(3.5),
  width: responsiveScreenHeight(80),
  resizeMode: "contain",
  marginBottom: 69,
 },
 title: {
  fontSize: 48,
  color: "#000",
  marginBottom: 109,
 },
 button: {
  alignItems: "center",
  marginTop: 30,
 },
 buttonBox: {
  borderRadius: 25,
  paddingVertical: 5,
```

```
    paddingHorizontal: 20,
    marginVertical: 10,
    width: 250,
    alignItems: "center",
    backgroundColor: "#B3B3B3",
  },
  icon: {
    alignSelf: "center",
    width: 24,
    height: 24,
  },
  write: {
    alignItems: "center",
    marginTop: 60,
  },
  placeholder: {
    borderRadius: 15,
    paddingVertical: 10,
    paddingHorizontal: 20,
    marginVertical: 10,
    width: 250,
    alignItems: "center",
    fontSize: 17,
    backgroundColor: "#E6E6E6",
    marginTop: 5,
    fontSize: 18,
    color: "#000",
  },
  create: {
    marginTop: 20,
    fontSize: 18,
  },
  buttonText: {
    fontSize: 18,
    textAlign: "center",
  },
  error: {
    color: "red",
    fontSize: 11,
  },
  underlinedText: {
    marginTop: 60,
```

```
    textAlign: "center",
    textDecorationLine: "underline",
  },
});

export default SigninScreen;
```

3.1.1.7 *ForgotPasswordScreen.js*

```
//ForgotPasswordScreen.js
import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, TouchableOpacity, Alert, StyleSheet, SafeAreaView, Image } from 'react-native';
import axios from 'axios';
import { responsiveWidth, responsiveFontSize, responsiveScreenHeight } from "react-native-responsive-dimensions";

const ForgotPasswordScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [newPassword, setNewPassword] = useState('');
  const [pin, setPin] = useState('');
  const [showPassword, setShowPassword] = useState(false);
  const [isClicked, setIsClicked] = useState(false);
  const [isFormValid, setIsFormValid] = useState(false);
  const [errors, setErrors] = useState({});

  const toggleShowPassword = () => {
    setShowPassword(!showPassword);
    setIsClicked((prevState) => !prevState);
  };

  useEffect(() => {
    validateForm();
  }, [email, newPassword, pin]);

  const validateForm = () => {
    let errors = {};

    if (!email) {
      errors.email = "Email is required.";
    } else if (!/\S+@\S+\.\S+/.test(email)) {
      errors.email = "Email is invalid.";
```

```
  }

  if (!newPassword) {
    errors.newPassword = "Password is required.";
  } else if (newPassword.length < 8) {
    errors.newPassword = "Password must be at least 8 characters.";
  }

  if (!pin) {
    errors.pin = "PIN is required.";
  } else if (pin.length !== 4) {
    errors.pin = "PIN must be 4 digits.";
  }

  setErrors(errors);
  setIsFormValid(Object.keys(errors).length === 0);
};

const handleResetPassword = async () => {
  try {
    const response = await axios.post('http://192.168.1.70:3000/resetPassword', {
      email,
      pin,
      newPassword
    });

    if (response.data.success) {
      Alert.alert('Success', 'Password has been reset successfully.');
      navigation.navigate('Login');
    } else {
      Alert.alert('Error', response.data.message);
    }
  } catch (error) {
    console.error('Error resetting password:', error);
    Alert.alert('Error', 'There was an issue resetting your password. Please try again.');
  }
};
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
return (
  <View style={styles.container}>
   <View style={styles.yellowBackground}>
    <Image
     source={require("../assets/cryptrail.png")}
     style={styles.cryptrail}
    />
    <Text style={styles.title}>Reset Password</Text>
   </View>
   <View style={styles.write}>
  <TextInput
   style={styles.placeholder}
   placeholder="Email"
   value={email}
   onChangeText={(text) => setEmail(text.toLowerCase())}
   autoCapitalize="none"
   keyboardType="email-address"
  />
  <TextInput
   style={styles.placeholder}
   placeholder="PIN"
   value={pin}
   secureTextEntry={!showPassword}
   onChangeText={setPin}
   keyboardType="numeric"
  />
  <TextInput
   style={styles.placeholder}
   secureTextEntry={!showPassword}
   placeholder="New Password"
   value={newPassword}
   onChangeText={setNewPassword}
  />
  <TouchableOpacity
        name={showPassword}
        size={24}
        color="#aaa"
        onPress={toggleShowPassword}

      >
       <Image
        style={styles.icon}
```

```
            source={
              isClicked
                ? require("../assets/hide.png")
                : require("../assets/view.png")
            }
           />
          </TouchableOpacity>
          <View style={styles.button}>
           <TouchableOpacity
            style={[styles.buttonBox, { opacity: isFormValid ? 1 : 0.5 }]}
            onPress={handleResetPassword}
           >
            <Text style={styles.buttonText}>Send</Text>
           </TouchableOpacity>
      </View>
      {Object.values(errors).map((error, index) => (
          <Text key={index} style={styles.error}>
           {error}
          </Text>
        ))}
      </View>
      </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "flex-start",
  },
  yellowBackground: {
    backgroundColor: "#FFD464",
    width: "100%",
    height: "43%",
    justifyContent: "center",
    alignItems: "flex-start",
    borderBottomLeftRadius: 63,
    borderBottomRightRadius: 63,
    paddingHorizontal: 20,
  },
  cryptrail: {
    height: responsiveScreenHeight(3.5),
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
  width: responsiveScreenHeight(80),
  resizeMode: "contain",
  marginBottom: 69,
},
title: {
  fontSize: 48,
  color: "#000",
  marginBottom: 109,
},
button: {
  alignItems: "center",
  marginTop: 30,
},
buttonBox: {
  borderRadius: 25,
  paddingVertical: 5,
  paddingHorizontal: 20,
  marginVertical: 10,
  width: 250,
  alignItems: "center",
  backgroundColor: "#B3B3B3",
},
buttonText: {
  fontSize: 18,
  textAlign: "center",
},
write: {
  alignItems: "center",
  marginTop: 60,
},
icon: {
  alignSelf: "center",
  width: 24,
  height: 24,
},
placeholder: {
  borderRadius: 15,
  paddingVertical: 10,
  paddingHorizontal: 20,
  marginVertical: 10,
  width: 250,
  alignItems: "center",
```

```
    fontSize: 17,
    backgroundColor: "#E6E6E6",
    marginTop: 5,
    fontSize: 18,
    color: "#000",
  },
  error: {
    color: "red",
    fontSize: 11,
  },
});

export default ForgotPasswordScreen;
```

3.1.1.8 *DashboardScreen.js*

```
//DashboardScreen.js
import React, { useState, useEffect, useRef } from "react";
import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  Alert,
  StyleSheet,
  Modal,
  FlatList,
  ActivityIndicator,
  Animated,
  Image,
  KeyboardAvoidingView,
  Linking,
  Platform,
} from "react-native";
import {
  responsiveWidth,
  responsiveFontSize,
  responsiveScreenHeight,
} from "react-native-responsive-dimensions";
import { useUser } from "../UserContext";

const Dashboard = ({ navigation }) => {
```

```
const [rates, setRates] = useState({});
const [isLoading, setIsLoading] = useState(false);
const [inputValue, setInputValue] = useState("");
const [baseCurrency, setBaseCurrency] = useState("FIAT");
const [targetCurrency, setTargetCurrency] = useState("CRYPTO");
const [result, setResult] = useState("");
const [modalVisible, setModalVisible] = useState(false);
const [isMenuVisible, setMenuVisible] = useState(false);
const menuAnimation = useRef(new Animated.Value(300)).current;
const contentAnimation = useRef(new Animated.Value(0)).current;
const [selectedCurrencyType, setSelectedCurrencyType] = useState("");
const [showBuyButton, setShowBuyButton] = useState(false);
const { user } = useUser();
const [conversionHistory, setConversionHistory] = useState([]);

const handleLogout = () => {
  Alert.alert(
    "Logout Confirmation",
    "Are you sure you want to logout?",
    [
      { text: "Cancel", style: "cancel" },
      { text: "Logout", onPress: () => navigation.navigate("Landing") },
    ],
    { cancelable: false }
  );
};

const toggleMenu = () => {
  Animated.timing(menuAnimation, {
    toValue: isMenuVisible ? 300 : 0,
    duration: 200,
    useNativeDriver: true,
  }).start();
  Animated.timing(contentAnimation, {
    toValue: isMenuVisible ? 0 : -300,
    duration: 200,
    useNativeDriver: true,
  }).start();
  setMenuVisible(!isMenuVisible);
};

const fetchConversionHistory = async () => {
```

```
  if (!user) return;
  try {
    const response = await fetch(`http://192.168.1.70:3000/getConversions/${user.id}`);
    const data = await response.json();
    setConversionHistory(data.reverse());
  } catch (error) {
    console.error('Error fetching conversion history:', error);
  }
};

useEffect(() => {
  const fetchInitialRates = async () => {
    setIsLoading(true);
    try {
      const response = await fetch("http://192.168.1.70:3000/rates");
      const jsonData = await response.json();
      setRates(jsonData);
    } catch (error) {
      console.error("Fetch error: ", error);
    } finally {
      setIsLoading(false);
    }
  };

  fetchInitialRates();
  fetchConversionHistory();
}, [user]);

const showCurrencyOptions = (type) => {
  setSelectedCurrencyType(type);
  setModalVisible(true);
};

const addToHistory = (base, target, baseAmount, targetAmount) => {
  const newHistory = [{ base, target, baseAmount, targetAmount }, ...conversionHistory];
  if (newHistory.length > 6) newHistory.pop();
  setConversionHistory(newHistory);
};

const selectCurrency = (currency) => {
  if (selectedCurrencyType === "base") {
    setBaseCurrency(currency);
```

```
  } else {
    setTargetCurrency(currency);
  }
  setModalVisible(false);
};

const convertCurrency = async () => {
  setShowBuyButton(false);
  setResult("");
  setIsLoading(true);
  try {
    const query = `base=${baseCurrency}&target=${targetCurrency}&amount=${inputValue}`;
    const response = await fetch(`http://192.168.1.70:3000/convert?${query}`);
    const data = await response.json();

    if (data.convertedAmount) {
      setResult(data.convertedAmount);
      addToHistory(baseCurrency, targetCurrency, inputValue, data.convertedAmount);

      await fetch('http://192.168.1.70:3000/saveConversion', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({
          userId: user.id,
          baseCurrency,
          targetCurrency,
          baseAmount: inputValue,
          targetAmount: data.convertedAmount,
        }),
      });

      setShowBuyButton(true);
      fetchConversionHistory();
    } else {
      setResult("Unavailable");
      Alert.alert("Conversion Error", "Unable to convert currency. Please try again later.");
    }
  } catch (error) {
    console.error("Conversion error: ", error);
    setResult("Unavailable");
    Alert.alert("Conversion Error", "An error occurred during currency conversion.");
  } finally {
```

```
    setIsLoading(false);
  }
};

const swapCurrencies = () => {
  setBaseCurrency((prevBase) => {
    setTargetCurrency(prevBase);
    return targetCurrency;
  });
};

const isInputValid = () => {
  return !isNaN(inputValue) && Number(inputValue) > 0;
};

const handleInputChange = (text) => {
  const numericValue = text.replace(/[^0-9.]/g, '');
  setInputValue(numericValue);
  setShowBuyButton(false);
  setResult("");
};

const renderHistoryItem = ({ item }) => (
  <View style={styles.historyItem}>
    <Image source={require("../assets/convert.png")} style={styles.convertIcon} />
    <Text style={styles.historyText}>{`${item.base_amount} ${item.base_currency} is ${item.target_amount}
${item.target_currency}`}</Text>
  </View>
);

const Separator = () => <View style={styles.separator} />;
const Seperators = () => <View style={styles.Seperators} />;

return (
  <View style={styles.container}>
    <Animated.View style={[styles.menu, { transform: [{ translateX: menuAnimation }] }]}>
      <View>
        <Text style={styles.welcome}>Hi, {user ? user.username : "Guest"}</Text>
        <Text style={{ paddingLeft: 2 }}>{user.email}</Text>
        <View style={styles.box}>
          <TouchableOpacity onPress={() => navigation.navigate('Account')}>
            <Text style={styles.text3}>Account</Text>
```

```
          </TouchableOpacity>
        </View>
        <View style={styles.box}>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/support.html')}>
            <Text style={styles.text2}>Support</Text>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/terms.html')}>
            <Text style={styles.text2}>Terms and Conditions</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.box}>
          <TouchableOpacity onPress={handleLogout}>
            <Text style={styles.text3}>Log Out</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.links}>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt')}>
            <Image source={require("../assets/site.png")} style={styles.socialM} />
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://instagram.com/rodrigo.lf6')}>
            <Image source={require("../assets/instagram.png")} style={styles.socialM} />
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://github.com/LFtech6')}>
            <Image source={require("../assets/github.png")} style={styles.socialM} />
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.linkedin.com/in/rodrigo-lopes-
ferreira-238906236/')}>
            <Image source={require("../assets/linkedin.png")} style={styles.socialM} />
          </TouchableOpacity>
        </View>
      </View>
    </Animated.View>
    <Animated.View style={[{ flex: 1 }, { transform: [{ translateX: contentAnimation }] }]}>
      <View style={{ flexDirection: "row" }}>
        <Image source={require("../assets/adaptive-icon.png")} style={styles.logo} />
        <Text style={styles.title}>Dashboard</Text>
        <TouchableOpacity style={styles.hamMenu} onPress={toggleMenu}>
          <Image style={styles.imageStyle} source={require("../assets/profile.png")} />
        </TouchableOpacity>
      </View>
      <Separator />
      <KeyboardAvoidingView style={{ flex: 1 }} behavior={Platform.OS === "ios" ? "padding" : "height"}>
```

```
      <View style={styles.view}>
        <View style={styles.convert}>
          <View style={styles.dropCont}>
            <TouchableOpacity style={styles.dropdownF} onPress={() => showCurrencyOptions("base")}>
              <View style={{ flexDirection: "row", alignItems: "center" }}>
                <TextInput
                  style={styles.input}
                  onChangeText={handleInputChange}
                  value={inputValue}
                  keyboardType="numeric"
                  placeholder="Enter amount"
                />
                <Text style={styles.dropText}>{rates[baseCurrency]?.name || "FIAT"}</Text>
              </View>
            </TouchableOpacity>
          </View>
          <View style={{ flexDirection: "row", justifyContent: "space-between" }}>
            <Seperators />
            <TouchableOpacity onPress={swapCurrencies}>
              <Image source={require("../assets/convert.png")} style={styles.change}></Image>
            </TouchableOpacity>
            <Seperators />
          </View>
          <View style={styles.dropCont}>
            <TouchableOpacity style={styles.dropdownC} onPress={() => showCurrencyOptions("target")}>
              <Text style={styles.dropText}>{rates[targetCurrency]?.name || "CRYPTO"}</Text>
            </TouchableOpacity>
          </View>
        </View>
        {isInputValid() && (
          <TouchableOpacity style={styles.convertButton} onPress={convertCurrency}>
            <Text style={styles.convertButtonText}>Convert</Text>
          </TouchableOpacity>
        )}
        {isLoading ? (
          <ActivityIndicator size="large" color="#0000ff" />
        ) : (
          <Text style={styles.resultText}>
            {`${inputValue} ${baseCurrency.toUpperCase()} is approximately ${result}
${targetCurrency.toUpperCase()}`}
          </Text>
        )}
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
        {showBuyButton && (
          <TouchableOpacity
            style={styles.button}
            onPress={() =>
              navigation.navigate("BuyPlaces", {
                cryptoCurrency: targetCurrency,
                baseCurrency: baseCurrency,
                convertedAmount: result,
                baseAmount: inputValue
              })
            }
          >
            <Text style={styles.buttonText}>Buy</Text>
          </TouchableOpacity>
        )}
      </View>
      <Modal
        animationType="slide"
        transparent={true}
        visible={modalVisible}
        onRequestClose={() => setModalVisible(!modalVisible)}
      >
        <View style={styles.centeredView}>
          <View style={styles.modalView}>
            <FlatList
              data={Object.entries(rates)}
              renderItem={({ item }) => (
                <TouchableOpacity key={item[0]} style={styles.optionItem} onPress={() =>
selectCurrency(item[0])}>
                  <Text style={styles.optionText}>{item[1].name}</Text>
                </TouchableOpacity>
              )}
              keyExtractor={(item) => item[0]}
            />
          </View>
        </View>
      </Modal>
      <Separator />
      <Text style={styles.last}>Last Conversions</Text>
      <View style={styles.historyContainer}>
        <FlatList
          data={conversionHistory.slice(0, 6)}
```

```
            renderItem={renderHistoryItem}
            numColumns={3}
            keyExtractor={(item, index) => index.toString()}
            style={{ marginBottom: 20 }}
          />
          {conversionHistory.length > 6 && (
            <TouchableOpacity
              style={styles.seeMoreButton}
              onPress={() => navigation.navigate("FullHistory", { history: conversionHistory, userId: user.id })}
            >
              <Text style={styles.seeMoreText}>See more</Text>
            </TouchableOpacity>
          )}
        </View>
      </KeyboardAvoidingView>
    </Animated.View>
  </View>
 );
};


const styles = StyleSheet.create({
 separator: {
  height: 1,
  backgroundColor: "#FFD464",
  width: "100%",
  marginVertical: 8,
 },
 container: {
  flex: 1,
  marginBottom: 80,
  padding: 10,
  backgroundColor: "white",
 },
 hamMenu: {
  position: "absolute",
  right: 0,
  top: 31,
  padding: 20,
 },
 imageStyle: {
  width: 30,
```

```
    height: 30,
  },
  menu: {
    position: 'absolute',
    backgroundColor: '#E4E3E3',
    width: 300,
    height: '100%',
    right: 0,
    padding: 20,
    zIndex: 2,
  },
  welcome: {
    fontSize: responsiveFontSize(4),
    fontWeight: "bold",
    marginTop: 70,
  },
  content: {
    marginTop: 10,
    paddingTop: 40,
  },
  box: {
    padding: 10,
    marginTop: 60,
    borderRadius: 20,
    backgroundColor: "#D3D3D3",
  },
  text1: {
    fontSize: responsiveFontSize(1.4),
    fontWeight: "bold",
    color: "white",
    paddingTop: 10,
    paddingBottom: 10,
  },
  text2: {
    fontSize: responsiveFontSize(1.4),
    fontWeight: "bold",
    color: "white",
    paddingTop: 10,
    paddingBottom: 10,
  },
  text3: {
    fontSize: responsiveFontSize(1.4),
```

```
  fontWeight: "bold",
  color: "white",
},
links: {
  flexDirection: "row",
  padding: 10,
  justifyContent: "space-around",
  marginTop: 100,
},
socialM: {
  width: responsiveWidth(5),
  height: responsiveWidth(5),
},

logo: {
  width: responsiveWidth(9),
  height: responsiveWidth(9),
  marginTop: responsiveScreenHeight(5),
},
title: {
  fontSize: responsiveFontSize(2.3),
  fontWeight: "bold",
  color: "#000",
  paddingHorizontal: responsiveWidth(0.3),
  marginTop: responsiveScreenHeight(5.8),
},
Seperators: {
  height: 1,
  backgroundColor: "#000",
  width: "40%",
  marginVertical: 8,
},
view: {
  marginTop: 60,
},
convert: {
  justifyContent: "center",
  marginBottom: 30,
},
dropCont: {
  flexDirection: "row",
  justifyContent: "center",
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
  marginBottom: 30,
},
dropdownF: {
  borderWidth: 1,
  borderColor: "white",
  backgroundColor: "#EFEFEF",
  borderRadius: 17,
  paddingVertical: 15,
  width: 370,
  alignSelf: "center",
  marginHorizontal: 10,
  marginTop: -30,
},
input: {
  flex: 1,
  borderRadius: 8,
  marginLeft: 20,
  marginRight: 5,
  fontSize: 14,
},
dropdownC: {
  borderWidth: 1,
  borderColor: "white",
  backgroundColor: "#EFEFEF",
  borderRadius: 17,
  paddingVertical: 15,
  width: 370,
  alignSelf: "center",
  marginHorizontal: 10,
  marginTop: 30,
},
dropText: {
  fontWeight: "bold",
  textAlign: "right",
  paddingHorizontal: 20,
  fontSize: 14,
},
change: {
  width: 30,
  height: 30,
  marginTop: -5,
},
```

```
  convertButton: {
    backgroundColor: "#007bff",
    borderRadius: 8,
    padding: 12,
    alignItems: "center",
    width: 90,
    alignSelf: "center",
    marginTop: -20,
    marginBottom: 40,
  },
  convertButtonText: {
    fontSize: responsiveFontSize(1.7),
    color: "#ffffff",
    fontWeight: "bold",
  },
  button: {
    backgroundColor: "#007bff",
    borderRadius: 8,
    padding: 12,
    alignItems: "center",
    width: 70,
    alignSelf: "center",
  },
  resultText: {
    fontSize: responsiveFontSize(1.7),
    color: "#333333",
    textAlign: "center",
    marginVertical: 20,
    marginTop: 10,
  },
  buttonText: {
    fontSize: responsiveFontSize(1.7),
    color: "#ffffff",
    fontWeight: "bold",
  },
  centeredView: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    marginTop: 22,
  },
  modalView: {
```

```
    margin: 20,
    backgroundColor: "white",
    borderRadius: 20,
    padding: 35,
    alignItems: "center",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 4,
    elevation: 5,
  },
  optionItem: {
    padding: 10,
    borderBottomWidth: 1,
    borderBottomColor: "#cccccc",
  },
  optionText: {
    fontSize: responsiveFontSize(2),
    color: "#333333",
  },
  last: {
    fontSize: 16,
    fontWeight: "bold",
    textAlign: "center",
    marginBottom: 10,
    marginTop: 20,
  },
  historyItem: {
    flex: 1,
    margin: 5,
    padding: 20,
    marginBottom: -15,
    alignItems: 'center',
    justifyContent: 'center',
    minHeight: 120,
  },
  convertIcon: {
    width: 20,
    height: 20,
```

```
    marginBottom: 10,
  },
  historyText: {
    fontSize: 12,
    color: "#333",
    textAlign: "center",
  },
  historyContainer: {
    flexDirection: 'row',
    flexWrap: 'wrap',
    alignItems: 'flex-end',
    justifyContent: 'space-between',
  },
  seeMoreButton: {
    padding: 10,
    alignItems: "flex-end",
    position: 'absolute',
    right: 10,
    paddingTop: 30,
  },
  seeMoreText: {
    color: "#0042A5",
    fontSize: 13,
  },
});

export default Dashboard;
```

3.1.1.9 *BuyPlacesScreen.js*

```
//BuyPlacesScreen.js
import React, { useState, useEffect } from 'react';
import { View, Text, StyleSheet, TouchableOpacity, ScrollView, ActivityIndicator, Linking } from 'react-native';
import axios from 'axios';
import { responsiveWidth, responsiveFontSize, responsiveScreenHeight } from "react-native-responsive-dimensions";

const BuyPlacesPage = ({ route }) => {
  const { cryptoCurrency, baseCurrency, convertedAmount, baseAmount } = route.params;
  const [exchanges, setExchanges] = useState([]);
  const [isLoading, setIsLoading] = useState(true);
```

```
useEffect(() => {
  const fetchExchanges = async () => {
    setIsLoading(true);
    try {
      const url = `http://192.168.1.70:3000/exchanges?currency=${cryptoCurrency}`;
      const response = await axios.get(url);
      if (response.data.length > 0) {
        const exchangesWithFees = response.data.map(exchange => {
          const feePercentage = Math.random() * 3.35;
          const feeAmount = feePercentage * baseAmount;
          return {
            ...exchange,
            feePercentage,
            feeAmount
          };
        });
        setExchanges(exchangesWithFees);
      } else {
        setExchanges([]);
      }
    } catch (error) {
      console.error('Error fetching exchanges for currency:', cryptoCurrency, error);
    }
    setIsLoading(false);
  };

  fetchExchanges();
}, [cryptoCurrency, baseAmount]);

if (isLoading) {
  return <ActivityIndicator size="large" />;
}

if (!exchanges.length) {
  return <Text>No exchanges available for {cryptoCurrency.toUpperCase()}</Text>;
}

return (
  <ScrollView style={styles.container}>
    <Text style={styles.title}>Buy {convertedAmount} {cryptoCurrency.toUpperCase()} with {baseAmount}
{baseCurrency.toUpperCase()}</Text>
    {exchanges.map((exchange, index) => (
```

```
      <View key={index} style={styles.exchangeContainer}>
        <Text style={styles.exchangeName}>{exchange.name}</Text>
        <Text style={styles.fee}>Fee: {(exchange.feePercentage).toFixed(2)}%</Text>
        <Text style={styles.total}>Total in {baseCurrency.toUpperCase()}: {((Number(baseAmount) +
exchange.feeAmount).toFixed(2))}</Text>
        <TouchableOpacity onPress={() => Linking.openURL(exchange.url)}>
          <Text style={styles.link}>Go to {exchange.name}</Text>
        </TouchableOpacity>
      </View>
    ))}
  </ScrollView>
 );
};

const styles = StyleSheet.create({
 container: {
  backgroundColor: 'white',
  flex: 1,
  padding: 10,
  marginTop: responsiveScreenHeight(5),
 },
 title: {
  fontSize: responsiveFontSize(2.5),
  fontWeight: 'bold',
  marginBottom: 20,
 },
 exchangeContainer: {
  marginBottom: 20,
  padding: 10,
  borderWidth: 1,
  borderColor: '#ddd',
  borderRadius: 10,
 },
 exchangeName: {
  fontSize: responsiveFontSize(2),
  fontWeight: 'bold',
 },
 link: {
  color: 'blue',
  marginTop: 5,
 },
 fee: {
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    fontSize: responsiveFontSize(1.8),
    marginTop: 5,
  },
  total: {
    fontSize: responsiveFontSize(1.8),
    marginTop: 5,
  },
});

export default BuyPlacesPage;
```

3.1.1.10 *FullHistory.js*

```
//FullHistory.js
import React from 'react';
import {
  View,
  Text,
  StyleSheet,
  FlatList,
  Button,
  Alert
} from 'react-native';
import { responsiveScreenHeight, responsiveFontSize } from 'react-native-responsive-dimensions';
import axios from 'axios';

const FullHistory = ({ route, navigation }) => {
  const { history, userId } = route.params;
  const [currentHistory, setCurrentHistory] = React.useState(history);

  console.log(`Received userId: ${userId}`);

  const handleDeleteHistory = async () => {
    const parsedUserId = parseInt(userId, 10);
    console.log(`Parsed userId: ${parsedUserId}`);
    if (isNaN(parsedUserId)) {
      Alert.alert("Error", "Invalid user ID format.");
      return;
    }
```

```
    try {
      const response = await axios.delete(`http://192.168.1.70:3000/conversions/${parsedUserId}`);
      if (response.status === 200) {
        Alert.alert("Success", "Conversion history deleted successfully.");
        setCurrentHistory([]);
        navigation.navigate('Dashboard0');
      } else if (response.status === 404) {
        Alert.alert("Info", "No conversion history found to delete.");
      } else {
        Alert.alert("Error", "Unexpected error occurred.");
      }
    } catch (error) {
      console.error('Delete history error:', error);
      Alert.alert("Error", `Failed to delete conversion history: ${error.response?.data?.message ||
error.message}`);
    }
  };

  const renderHistoryItem = ({ item }) => (
    <View style={styles.itemContainer}>
      <Text style={styles.itemText}>
        {`${item.base_amount ?? '0'} ${item.base_currency?.toUpperCase() ?? 'UNKNOWN'} is
${item.target_amount ?? '0'} ${item.target_currency?.toUpperCase() ?? 'UNKNOWN'}`}
      </Text>
    </View>
  );

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Conversion History</Text>
      <FlatList
        data={currentHistory}
        renderItem={renderHistoryItem}
        keyExtractor={(item, index) => index.toString()}
        contentContainerStyle={styles.listContent}
      />
      <View style={styles.buttonContainer}>
        <Button
          title="Delete History"
          onPress={handleDeleteHistory}
          color="#ff5c5c"
        />
```

```
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: responsiveScreenHeight(2),
    backgroundColor: 'white',
  },
  title: {
    marginTop: responsiveScreenHeight(4),
    fontSize: responsiveFontSize(3),
    fontWeight: 'bold',
    textAlign: 'center',
    marginBottom: responsiveScreenHeight(2),
  },
  listContent: {
    paddingHorizontal: responsiveScreenHeight(2),
  },
  itemContainer: {
    backgroundColor: '#fff',
    padding: responsiveScreenHeight(2),
    marginVertical: responsiveScreenHeight(1),
    borderRadius: responsiveScreenHeight(1),
    elevation: 3,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 5,
  },
  itemText: {
    fontSize: responsiveFontSize(2),
    color: '#333',
  },
  buttonContainer: {
    backgroundColor: '#FFD464',
    padding: responsiveScreenHeight(2),
    marginHorizontal: responsiveScreenHeight(2),
    borderRadius: responsiveScreenHeight(1),
    alignItems: 'center',
```

```
    marginVertical: responsiveScreenHeight(2),
  }
});

export default FullHistory;
```

3.1.1.11 *TrailScreen.js*

```
//TrailScreen.js
import React, { useState, useEffect, useRef } from "react";
import {
  View,
  TextInput,
  Text,
  StyleSheet,
  ScrollView,
  TouchableOpacity,
  Alert,
  Image,
  ActivityIndicator,
  TouchableWithoutFeedback,
  Keyboard,
  KeyboardAvoidingView,
  Animated,
} from "react-native";
import axios from "axios";
import { useUser } from "../UserContext";
import {
  responsiveWidth,
  responsiveScreenHeight,
  responsiveFontSize,
} from "react-native-responsive-dimensions";

const TrailScreen = ({ navigation }) => {
  const [messages, setMessages] = useState([]);
  const [inputText, setInputText] = useState("");
  const [isBotWriting, setIsBotWriting] = useState(false);
  const [isMenuVisible, setMenuVisible] = useState(false);
  const menuAnimation = useRef(new Animated.Value(300)).current;
  const contentAnimation = useRef(new Animated.Value(0)).current;
  const [hasSentMessage, setHasSentMessage] = useState(false);
  const { user, setUser } = useUser();
```

```
useEffect(() => {
  if (user && user.id) {
    fetchMessages(user.id);
  }
}, [user]);

const fetchMessages = async (userId) => {
  try {
    const response = await axios.get(
      `http://192.168.1.70:3000/conversations/${userId}`
    );
    setMessages(response.data);
  } catch (error) {
    console.error("Failed to fetch messages:", error);
  }
};

const saveConversation = async (userId, conversationContent) => {
  try {
    const response = await axios.post(
      "http://192.168.1.70:3000/saveConversation",
      {
        userId,
        content: conversationContent,
      }
    );
    console.log("Conversation saved. ID:", response.data.conversationId);
  } catch (error) {
    console.error("Failed to save conversation:", error);
  }
};

const sendMessage = async () => {
  if (!inputText.trim()) return;
  const userMessage = { role: "user", content: inputText };
  setIsBotWriting(true);
  try {
    const botResponse = await axios.post(
      "https://api.openai.com/v1/chat/completions",
      {
        model: "gpt-3.5-turbo",
```

```
        messages: [{ ...userMessage }],
        temperature: 0.5,
      },
      {
        headers: {
          "Content-Type": "application/json",
          Authorization:
            "Bearer sk-proj-aVKnxkecomwLBFdWmrWgT3BlbkFJjfllx1tX46NeXj31Tx6t",
        },
      }
    );

    const botMessage = {
      role: "bot",
      content: botResponse.data.choices[0].message.content,
    };
    const updatedMessages = [...messages, userMessage, botMessage];

    setMessages(updatedMessages);
    setIsBotWriting(false);
    setHasSentMessage(true);



    await saveConversation(user.id, updatedMessages);


    setInputText("");
  } catch (error) {
    console.error("Error sending or saving message:", error);
    setIsBotWriting(false);
  }
};

const clearMessages = async () => {
  Alert.alert(
    "Confirm Deletion",
    "Are you sure you want to delete all messages?",
    [
      {
        text: "Cancel",
        onPress: () => console.log("Deletion cancelled."),
```

```
        style: "cancel",
      },
      {
        text: "OK",
        onPress: async () => {
          try {
            await axios.delete(
              `http://192.168.1.70:3000/conversations/${user.id}`
            );
            setMessages([]);
            setHasSentMessage(false);
          } catch (error) {
            console.error("Error clearing messages:", error);
          }
        },
      },
    ],
    { cancelable: true }
  );
};



const toggleMenu = () => {
  Animated.timing(menuAnimation, {
    toValue: isMenuVisible ? 300 : 0,
    duration: 200,
    useNativeDriver: true,
  }).start();

  Animated.timing(contentAnimation, {
    toValue: isMenuVisible ? 0 : -300,
    duration: 200,
    useNativeDriver: true,
  }).start();

  setMenuVisible(!isMenuVisible);
};


const handleLogout = () => {
  Alert.alert(
```

```
    "Logout Confirmation",
    "Are you sure you want to logout?",
    [
      {
        text: "Cancel",
        style: "cancel",
      },
      {
        text: "Logout",
        onPress: () => {
          navigation.navigate("Landing");
        },
      },
    ],
    { cancelable: false }
  );
};


const defaultQuestions = [
  "How to start investing on crypto?",
  "What is blockchain?",
  "Who was the first person ever buying bitcoin?",
  "Who is Satoshi Nakamoto?",

];

const validateMessageContent = (content) => {
  if (typeof content === "string") {
    return content;
  }

  console.warn("Invalid content type:", typeof content, content);
  return "";
};

const clearInput = () => setInputText("");

const Separator = () => <View style={styles.separator} />;

return (
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    <View style={styles.container}>
      <Animated.View
      style={[
        styles.menu,
        {
          transform: [{ translateX: menuAnimation }],
        },
      ]}
    >
        <View style={styles.content1}>
        <Text style={styles.welcome}>Hi, {user ? user.username : "Guest"}</Text>
        <Text style={{ paddingLeft: 2,}}>{user.email}</Text>
        <View style={styles.box}>
        <TouchableOpacity onPress={() => navigation.navigate('Account')}>
            <Text style={styles.text3}>Account</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.box}>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/support.html')}>
            <Text style={styles.text2}>Support</Text>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/terms.html')}>
            <Text style={styles.text2}>Terms and Conditions</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.box}>
          <TouchableOpacity onPress={handleLogout}>
            <Text style={styles.text3}>Log Out</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.links}>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt')}>
            <Image source={require("../assets/site.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://instagram.com/rodrigo.lf6')}>
            <Image source={require("../assets/instagram.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://github.com/LFtech6')}>
            <Image source={require("../assets/github.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.linkedin.com/in/rodrigo-lopes-
ferreira-238906236/')}>
```

```
        <Image source={require("../assets/linkedin.png")} style={styles.socialM}/>
       </TouchableOpacity>
      </View>
      </View>
    </Animated.View>
    <Animated.View
    style={[
     { flex: 1 },
     {
      transform: [{ translateX: contentAnimation }],
     },
    ]}
  >
     <View style={{ flexDirection: "row" }}>
      <Image
       source={require("../assets/adaptive-icon.png")}
       style={styles.logo}
      />
      <Text style={styles.title}>Trail</Text>
      <TouchableOpacity style={styles.hamMenu} onPress={toggleMenu}>
        <Image
         style={styles.imageStyle}
         source={require("../assets/profile.png")}
        />
       </TouchableOpacity>
    </View>
    <Separator />
    <ScrollView style={styles.messageContainer}>
     {messages.map((message, index) => {
      return (
       <View
        key={index}
        style={[
         styles.messageBubble,
         message.role === "user"
          ? styles.userMessage
          : styles.botMessage,
        ]}
       >
        <Text style={styles.messageText}>{message.content}</Text>
       </View>
      );
```

```
      ))}

      {isBotWriting && (
        <View style={styles.loadingContainer}>
          <ActivityIndicator style={styles.loading} />
          <Text style={styles.loadingText}>Writing you a message...</Text>
        </View>
      )}
    </ScrollView>
    {!hasSentMessage && (
  <ScrollView horizontal style={styles.defaultQuestionsContainer}>
    {defaultQuestions.map((question, index) => (
      <TouchableOpacity
        key={index}
        style={styles.defaultQuestion}
        onPress={() => {
          setInputText(question);
          setHasSentMessage(true);
        }}
      >
        <View style={styles.question}>
          <Text style={styles.textQ}>{question}</Text>
        </View>
      </TouchableOpacity>
    ))}
  </ScrollView>
)}
<View style={{marginBottom: 3}}>
    <TouchableOpacity
      onPress={clearMessages}
      style={styles.clearMessagesButton}
    >
      <Text style={styles.text}>Clear Messages</Text>
    </TouchableOpacity>
    <KeyboardAvoidingView behavior="padding">
      <View style={styles.inputContainer}>
        <TextInput
          style={styles.input}
          onChangeText={setInputText}
          value={inputText}
          placeholder="Write your message..."
        />
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
          <TouchableOpacity onPress={clearInput} style={styles.clearButton}>
            <Image source={require('../assets/times-circle.png')} style={styles.Icon}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={sendMessage} style={styles.sendButton}>
            <Image source={require('../assets/paper-plane.png')} style={styles.Icon}/>
          </TouchableOpacity>
        </View>
      </KeyboardAvoidingView>
      </View>
      </Animated.View>
    </View>
  );
};

const styles = StyleSheet.create({
  separator: {
    height: 1,
    backgroundColor: "#FFD464",
    width: "100%",
    marginVertical: 8,
  },
  container: {
    flex: 1,
    marginBottom: 80,
    padding: 10,
    backgroundColor: "#fff",
  },
  hamMenu: {
    position: "absolute",
    right: 0,
    top: 31,
    padding: 20,
  },
  imageStyle: {
    width: 30,
    height: 30,
  },
  menu: {
    position: 'absolute',
    backgroundColor: '#E4E3E3',
    width: 300,
    height: '100%',
```

```
  right: 0,
  padding: 20,
  zIndex: 2,
},
welcome: {
  fontSize: responsiveFontSize(4),
  fontWeight: "bold",
  marginTop: 70,
},
content: {
  marginTop: 10,
  paddingTop: 40,
},
box: {
  padding: 10,
  marginTop: 60,
  borderRadius: 20,
  backgroundColor: "#D3D3D3",
},
text1: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text2: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text3: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
},
links: {
  flexDirection: "row",
  padding: 10,
  justifyContent: "space-around",
```

```
  marginTop: 100,
},
socialM: {
  width: responsiveWidth(5),
  height: responsiveWidth(5),
},

logo: {
  width: responsiveWidth(9),
  height: responsiveWidth(9),
  marginTop: responsiveScreenHeight(5),
},
title: {
  fontSize: responsiveFontSize(2.3),
  fontWeight: "bold",
  color: "#000",
  paddingHorizontal: responsiveWidth(0.3),
  marginTop: responsiveScreenHeight(5.8),
},
messageBubble: {
  paddingVertical: 8,
  paddingHorizontal: 14,
  borderRadius: 20,
  marginVertical: 5,
},
userMessage: {
  backgroundColor: "#1a73e8",
  alignSelf: "flex-end",
},
botMessage: {
  backgroundColor: "#e0f0e0",
  alignSelf: "flex-start",
},
messageText: {
  color: "#000",
},
inputContainer: {
  flexDirection: "row",
  alignItems: "center",
  margin: 10,
},
input: {
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    flex: 1,
    paddingVertical: 10,
    paddingHorizontal: 15,
    backgroundColor: "#f9f9f9",
    borderRadius: 50,
    marginRight: 10,
    borderWidth: 1,
    borderColor: "#FFD464",
  },
  clearButton: {
    padding: 10,
  },
  clearMessagesButton: {
    padding: 10,
    alignItems: "center",
    backgroundColor: "#FFD464",
    borderRadius: 50,
    width: "28%",
    alignSelf: "flex-end",
    marginBottom: -10,
  },
  text: {
    fontSize: 10,
    fontWeight: "bold",
  },
  sendButton: {
    padding: 10,
  },
  Icon: {
    width: 20,
    height: 20,
  },
  loadingContainer: {
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "center",
  },
  loading: {
    marginRight: 10,
  },
  loadingText: {
    fontSize: 16,
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
  },
  defaultQuestionsContainer: {
    flexDirection: 'row',
    paddingTop: 540,
  },
  defaultQuestion: {
    marginRight: 20,
    borderRadius: 5,
  },
  question: {
    padding: 10,
    alignItems: "center",
    backgroundColor: "#FFD464",
    borderRadius: 50,
  },
  textQ: {
    fontSize: 13,
    fontWeight: "bold",
  },
});

export default TrailScreen;
```

3.1.1.12 *MarketScreen.js*

```
//MarketScreen.js
import React, { useState, useEffect, useRef } from "react";
import {
  Image,
  StyleSheet,
  Text,
  TouchableOpacity,
  View,
  FlatList,
  Linking,
  TextInput,
  Modal,
  TouchableWithoutFeedback,
  Alert,
  Keyboard,
  KeyboardAvoidingView,
  Animated,
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
} from "react-native";
import {
  responsiveWidth,
  responsiveScreenHeight,
  responsiveFontSize,
} from "react-native-responsive-dimensions";
import { useUser } from "../UserContext";

const MarketScreen = ({ navigation }) => {
  const [selectedTab, setSelectedTab] = useState("Coins");
  const [data, setData] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState(null);
  const [sortDescending, setSortDescending] = useState(true);
  const [showModal, setShowModal] = useState(false);
  const [searchTerm, setSearchTerm] = useState("");
  const [items, setItems] = useState([]);
  const [filteredItems, setFilteredItems] = useState([]);
  const [isMenuVisible, setMenuVisible] = useState(false);
  const menuAnimation = useRef(new Animated.Value(300)).current;
  const contentAnimation = useRef(new Animated.Value(0)).current;
  const { user, setUser } = useUser();

  const handleSearch = (text) => {
    setSearchTerm(text);
    if (text) {
      const formattedQuery = text.toLowerCase();
      const newItems = items.filter((item) =>
        item.name.toLowerCase().includes(formattedQuery)
      );
      setFilteredItems(newItems);
    } else {
      setFilteredItems(items);
    }
  };

  const toggleMenu = () => {
    Animated.timing(menuAnimation, {
      toValue: isMenuVisible ? 300 : 0,
      duration: 200,
      useNativeDriver: true,
    }).start();
```

```
  Animated.timing(contentAnimation, {
    toValue: isMenuVisible ? 0 : -300,
    duration: 200,
    useNativeDriver: true,
  }).start();

  setMenuVisible(!isMenuVisible);
};

useEffect(() => {
  fetchDataFromApi(selectedTab);
}, [selectedTab]);

const fetchDataFromApi = async (endpoint) => {
  setIsLoading(true);
  setError(null);
  try {
    const response = await fetch(`http://192.168.1.70:3000/${endpoint}`);
    let json = await response.json();
    if (endpoint === "Exchanges") {
      json = json.sort((a, b) => b.trust_score - a.trust_score);
    } else if (endpoint === "Coins") {
      json = sortData(json);
    }
    setData(json);
    setItems(json);
    setFilteredItems(json);
  } catch (error) {
    setError(`Failed to fetch ${endpoint}. Please try again later.`);
    console.error(error);
  } finally {
    setIsLoading(false);
  }
};

const renderCoinItem = ({ item, index }) => {
  const priceChange = parseFloat(item.price_change_percentage_24h);

  return (
    <View style={styles.listItem}>
      <Text style={styles.coinNumber}>{index + 1}</Text>
```

```
      <Image source={{ uri: item.image_url }} style={styles.coinImage} />
      <View style={{ flex: 1 }}>
        <Text style={styles.coinName}>{item.name}</Text>
        <Text style={styles.coinCap}>€ {item.market_cap}</Text>
      </View>
      <View style={{ flex: 1 }}>
        <Text style={styles.coinPrice}>€{item.current_price}</Text>
      </View>
      <View style={{ flex: 1 }}>
        <Text
          style={[
            styles.variation,
            { color: priceChange < 0 ? "red" : "green" },
          ]}
        >
          {isNaN(priceChange) ? "N/A" : priceChange.toFixed(2)}%
        </Text>
      </View>
    </View>
  );
};

const handleLogout = () => {
  Alert.alert(
    "Logout Confirmation",
    "Are you sure you want to logout?",
    [
      {
        text: "Cancel",
        style: "cancel",
      },
      {
        text: "Logout",
        onPress: () => {
          navigation.navigate("Landing");
        },
      },
    ],
    { cancelable: false }
  );
};
```

```
const sortData = (coins) => {
  return coins.sort((a, b) => {
    if (sortDescending) {
      return b.market_cap - a.market_cap;
    } else {
      return a.market_cap - b.market_cap;
    }
  });
};

const toggleSortOrder = () => {
  setSortDescending(!sortDescending);
};

const toggleModal = () => {
  setShowModal(!showModal);
};

const renderExchangeItem = ({ item, index }) => {
  const priceChange = parseFloat(item.trade_volume_24h_btc);

  return (
    <View style={styles.listItem}>
      <Text style={styles.coinNumber}>{index + 1}</Text>
      <Image source={{ uri: item.image_url }} style={styles.exchangeImage} />
      <View style={{ flex: 1 }}>
        <TouchableOpacity onPress={() => Linking.openURL(item.url)}>
          <Text style={styles.exchangeName}>{item.name}</Text>
        </TouchableOpacity>
        <Text style={styles.exchangeTrade}>{item.trust_score}</Text>
      </View>
      <View style={{ flex: 1 }}>
        <Text style={[styles.variationE]}>
          {isNaN(priceChange) ? "N/A" : priceChange.toFixed(2)}%
        </Text>
      </View>
    </View>
  );
};

const renderContent = () => {
  if (error) return <Text style={styles.errorText}>{error}</Text>;
```

```
  return (
    <FlatList
      data={filteredItems}
      renderItem={
        selectedTab === "Coins" ? renderCoinItem : renderExchangeItem
      }
      keyExtractor={(item) => item.id.toString()}
    />
  );
};
const Separator = () => <View style={styles.separator} />;

return (
  <View style={styles.container}>
    <Animated.View
      style={[
        styles.menu,
        {
          transform: [{ translateX: menuAnimation }],
        },
      ]}
    >
      <View style={styles.content1}>
      <Text style={styles.welcome}>Hi, {user ? user.username : "Guest"}</Text>
      <Text style={{ paddingLeft: 2,}}>{user.email}</Text>
      <View style={styles.box}>
      <TouchableOpacity onPress={() => navigation.navigate('Account')}>
          <Text style={styles.text3}>Account</Text>
        </TouchableOpacity>
      </View>
      <View style={styles.box}>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/support.html')}>
          <Text style={styles.text2}>Support</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/terms.html')}>
          <Text style={styles.text2}>Terms and Conditions</Text>
        </TouchableOpacity>
      </View>
      <View style={styles.box}>
        <TouchableOpacity onPress={handleLogout}>
          <Text style={styles.text3}>Log Out</Text>
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
          </TouchableOpacity>
        </View>
        <View style={styles.links}>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt')}>
          <Image source={require("../assets/site.png")} style={styles.socialM}/>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => Linking.openURL('https://instagram.com/rodrigo.lf6')}>
          <Image source={require("../assets/instagram.png")} style={styles.socialM}/>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => Linking.openURL('https://github.com/LFtech6')}>
          <Image source={require("../assets/github.png")} style={styles.socialM}/>
        </TouchableOpacity>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.linkedin.com/in/rodrigo-lopes-
ferreira-238906236/')}>
          <Image source={require("../assets/linkedin.png")} style={styles.socialM}/>
        </TouchableOpacity>
        </View>
        </View>
      </Animated.View>
      <Animated.View
      style={[
        { flex: 1 },
        {
          transform: [{ translateX: contentAnimation }],
        },
      ]}
    >
    <View style={{ flexDirection: "row" }}>
      <Image
        source={require("../assets/adaptive-icon.png")}
        style={styles.logo}
      />
      <Text style={styles.title}>Market</Text>
      <TouchableOpacity style={styles.hamMenu} onPress={toggleMenu}>
          <Image
            style={styles.imageStyle}
            source={require("../assets/profile.png")}
          />
        </TouchableOpacity>
    </View>
    <Separator />
    <KeyboardAvoidingView>
```

```
<View style={styles.searchBarContainer}>
 <View style={styles.searchArea}>
  <TextInput
   style={styles.searchBar}
   placeholder="Search for assets"
   value={searchTerm}
   onChangeText={handleSearch}
  />
  <TouchableOpacity style={styles.toggleButton} onPress={toggleModal}>
   <Image
    source={require("../assets/Slider.png")}
    style={styles.headerIcon}
   />
  </TouchableOpacity>
 </View>

 <Modal
  transparent={true}
  visible={showModal}
  onRequestClose={() => {
   setShowModal(!showModal);
  }}
 >
  <View style={styles.modalView}>
   <TouchableOpacity
    style={styles.modalButton}
    onPress={() => {
     setSelectedTab("Exchanges");
     toggleModal();
    }}
   >
    <Text style={styles.modalText}>Exchanges</Text>
   </TouchableOpacity>
   <TouchableOpacity
    style={styles.modalButton}
    onPress={() => {
     setSelectedTab("Coins");
     toggleModal();
    }}
   >
    <Text style={styles.modalText}>Coins</Text>
   </TouchableOpacity>
```

```
        </View>
      </Modal>
    </View>
  </KeyboardAvoidingView>
  <View
    style={{
      flexDirection: "row",
      justifyContent: "space-between",
      paddingHorizontal: 20,
      marginTop: 20,
    }}
  >
    <Text style={styles.headerNumber}>#</Text>
    {selectedTab === "Coins" ? (
      <>
        <TouchableOpacity
          style={styles.headerButton}
          onPress={toggleSortOrder}
        >
          <Text style={styles.headerTitle}>Market Cap</Text>
          <Image
            source={require("../assets/setabaixo.png")}
            style={styles.organizer}
          />
        </TouchableOpacity>
        <Text style={styles.headerTitle}>Price</Text>
        <Text style={styles.headerPercentage}>30d%</Text>
      </>
    ) : (
      <>
        <TouchableOpacity
          style={styles.headerButton}
          onPress={toggleSortOrder}
        >
          <Text style={styles.headerTitle}> Rating</Text>
          <Image
            source={require("../assets/setabaixo.png")}
            style={styles.organizer}
          />
        </TouchableOpacity>
        <Text style={styles.headerTitle}></Text>
        <Text style={styles.headerVolumeE}>24h Vol</Text>
```

```
        </>
       )}
      </View>
      {renderContent()}
      </Animated.View>
    </View>
  );
};

const styles = StyleSheet.create({
  separator: {
    height: 1,
    backgroundColor: "#FFD464",
    width: "100%",
    marginVertical: 8,
  },
  container: {
    flex: 1,
    marginBottom: 80,
    padding: 10,
    backgroundColor: "#fff",
  },
  hamMenu: {
    position: "absolute",
    right: 0,
    top: 31,
    padding: 20,
  },
  imageStyle: {
    width: 30,
    height: 30,
  },
  menu: {
    position: 'absolute',
    backgroundColor: '#E4E3E3',
    width: 300,
    height: '100%',
    right: 0,
    padding: 20,
    zIndex: 2,
  },
  welcome: {
```

```
  fontSize: responsiveFontSize(4),
  fontWeight: "bold",
  marginTop: 70,
},
content: {
  marginTop: 10,
  paddingTop: 40,
},
box: {
  padding: 10,
  marginTop: 60,
  borderRadius: 20,
  backgroundColor: "#D3D3D3",
},
text1: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text2: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text3: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
},
links: {
  flexDirection: "row",
  padding: 10,
  justifyContent: "space-around",
  marginTop: 100,
},
socialM: {
  width: responsiveWidth(5),
  height: responsiveWidth(5),
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
},
logo: {
  width: responsiveWidth(9),
  height: responsiveWidth(9),
  marginTop: responsiveScreenHeight(5),
},
title: {
  fontSize: responsiveFontSize(2.3),
  fontWeight: "bold",
  color: "#000",
  paddingHorizontal: responsiveWidth(0.3),
  marginTop: responsiveScreenHeight(5.8),
},
searchBarContainer: {
  flexDirection: "row",
  justifyContent: "space-between",
  alignItems: "center",
  paddingHorizontal: responsiveWidth(5),
  paddingVertical: responsiveScreenHeight(1),
  marginTop: responsiveScreenHeight(2),
},
searchArea: {
  flexDirection: "row",
  alignItems: "center",
  justifyContent: "flex-start",
  flex: 1,
},
searchBar: {
  flex: 1,
  borderWidth: 1,
  borderColor: "#e0e0e0",
  borderRadius: 10,
  paddingLeft: responsiveWidth(5),
  fontSize: responsiveFontSize(2),
},
toggleButton: {
  paddingHorizontal: responsiveWidth(4),
  paddingVertical: responsiveScreenHeight(1),
  borderRadius: 10,
  marginLeft: -responsiveWidth(10),
},
toggleButtonText: {
```

```
    fontSize: responsiveFontSize(2),
    color: "#333",
  },
  headerIcon: {
    width: 20,
    height: 20,
  },
  modalView: {
    margin: 20,
    backgroundColor: "#EFEFEF",
    borderRadius: 20,
    padding: 35,
    alignItems: "center",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    borderColor: "#e0e0e0",
    elevation: 5,
    position: "absolute",
    top: 170,
    right: 10,
  },
  modalButton: {
    padding: 10,

  },
  modalText: {
    textAlign: "center",

  },
  content: {
    flex: 1,
    alignItems: "center",
  },
  listItem: {
    flexDirection: "row",
    alignItems: "center",
    paddingVertical: responsiveScreenHeight(1),
    paddingHorizontal: responsiveWidth(2),
    borderBottomWidth: 1,
```

```
  borderBottomColor: "#e0e0e0",
},
coinNumber: {
  fontSize: responsiveFontSize(1.5),
  width: responsiveWidth(7),
  textAlign: "center",
},
coinImage: {
  width: responsiveWidth(8),
  height: responsiveWidth(8),
  borderRadius: responsiveWidth(4),
  marginRight: responsiveWidth(2),
},
coinName: {
  fontSize: responsiveFontSize(1.8),
  fontWeight: "bold",
  color: "#333",
},
coinCap: {
  fontSize: responsiveFontSize(0.9),
  color: "#333",
  marginTop: responsiveScreenHeight(0.5),
},
coinPrice: {
  fontSize: responsiveFontSize(1.6),
  color: "#333",
  marginLeft: responsiveWidth(7),
},
variation: {
  fontSize: responsiveFontSize(1.6),
  color: "#333",
  marginLeft: responsiveWidth(12),
},
exchangeImage: {
  width: responsiveWidth(8),
  height: responsiveWidth(8),
  borderRadius: responsiveWidth(4),
  marginRight: responsiveWidth(2),
},
exchangeName: {
  fontSize: responsiveFontSize(1.8),
  fontWeight: "bold",
```

```
    color: "#333",
  },
  exchangeTrade: {
    fontSize: responsiveFontSize(1.6),
    color: "#333",
    marginLeft: responsiveWidth(1),
  },
  variationE: {
    fontSize: responsiveFontSize(1.6),
    color: "#333",
    marginLeft: responsiveWidth(20),
  },
  headerNumber: {
    fontSize: responsiveFontSize(1.8),
    width: responsiveWidth(2),
    textAlign: "center",
  },
  headerTitle: {
    fontSize: responsiveFontSize(1.8),
    marginRight: responsiveWidth(2),
  },
  headerPercentage: {
    fontSize: responsiveFontSize(1.8),
  },
  headerVolume: {
    fontSize: responsiveFontSize(1.8),
  },
  headerVolumeE: {
    fontSize: responsiveFontSize(1.8),
  },
  headerButton: {
    flexDirection: "row",
    alignItems: "center",
    marginRight: responsiveWidth(2),
  },
  organizer: {
    width: 10,
    height: 10,
    marginLeft: 5,
  },
});
```

```
export default MarketScreen;
```

3.1.1.13 *NewsScreen.js*

```
//NewsScreen.js
import React, { useState, useEffect, useRef } from "react";
import {
  View,
  Text,
  Image,
  StyleSheet,
  Alert,
  ScrollView,
  RefreshControl,
  TouchableOpacity,
  Linking,
  SafeAreaView,
  Animated,
  ActivityIndicator,
} from "react-native";
import {
  responsiveWidth,
  responsiveFontSize,
  responsiveScreenHeight,
} from "react-native-responsive-dimensions";
import axios from "axios";
import { useUser } from "../UserContext";

const NewsScreen = ({ navigation }) => {
  const [newsArticles, setNewsArticles] = useState([]);
  const [refreshing, setRefreshing] = useState(false);
  const [loading, setLoading] = useState(true);
  const [isMenuVisible, setMenuVisible] = useState(false);
  const menuAnimation = useRef(new Animated.Value(300)).current;
  const contentAnimation = useRef(new Animated.Value(0)).current;
  const { user, setUser } = useUser();

  const handleLogout = () => {
    Alert.alert(
      "Logout Confirmation",
      "Are you sure you want to logout?",
      [
```

```
      {
        text: "Cancel",
        style: "cancel",
      },
      {
        text: "Logout",
        onPress: () => {
          navigation.navigate("Landing");
        },
      },
    ],
    { cancelable: false }
  );
};

const fetchNews = async () => {
  try {
    setLoading(true);
    const response = await axios.get("http://192.168.1.70:3000/news");
    setNewsArticles(response.data || []);
  } catch (error) {
    console.error("Error fetching news:", error);
  } finally {
    setLoading(false);
    setRefreshing(false);
  }
};

const toggleMenu = () => {
  Animated.timing(menuAnimation, {
    toValue: isMenuVisible ? 300 : 0,
    duration: 200,
    useNativeDriver: true,
  }).start();

  Animated.timing(contentAnimation, {
    toValue: isMenuVisible ? 0 : -300,
    duration: 200,
    useNativeDriver: true,
  }).start();

  setMenuVisible(!isMenuVisible);
```

```
};

const openLink = async (url) => {
  if (await Linking.canOpenURL(url)) {
    await Linking.openURL(url);
  } else {
    console.error(`Don't know how to open this URL: ${url}`);
  }
};

useEffect(() => {
  fetchNews();
}, []);

if (loading) {
  return (
    <View style={styles.container}>
      <ActivityIndicator size="large" color="#000000" />
    </View>
  );
}

const Seperator = () => <View style={styles.Seperator} />;

return (
  <View style={styles.container}>
    <Animated.View
      style={[
        styles.menu,
        {
          transform: [{ translateX: menuAnimation }],
        },
      ]}
    >
        <View style={styles.content1}>
        <Text style={styles.welcome}>Hi, {user ? user.username : "Guest"}</Text>
        <Text style={{ paddingLeft: 2,}}>{user.email}</Text>
        <View style={styles.box}>
        <TouchableOpacity onPress={() => navigation.navigate('Account')}>
            <Text style={styles.text3}>Account</Text>
          </TouchableOpacity>
        </View>
```

```
        <View style={styles.box}>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/support.html')}>
            <Text style={styles.text2}>Support</Text>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt/terms.html')}>
            <Text style={styles.text2}>Terms and Conditions</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.box}>
          <TouchableOpacity onPress={handleLogout}>
            <Text style={styles.text3}>Log Out</Text>
          </TouchableOpacity>
        </View>
        <View style={styles.links}>
        <TouchableOpacity onPress={() => Linking.openURL('https://www.cryptrail.pt')}>
            <Image source={require("../assets/site.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://instagram.com/rodrigo.lf6')}>
            <Image source={require("../assets/instagram.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://github.com/LFtech6')}>
            <Image source={require("../assets/github.png")} style={styles.socialM}/>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => Linking.openURL('https://www.linkedin.com/in/rodrigo-lopes-
ferreira-238906236/')}>
            <Image source={require("../assets/linkedin.png")} style={styles.socialM}/>
          </TouchableOpacity>
        </View>
        </View>
      </Animated.View>
      <Animated.View
      style={[
        { flex: 1 },
        {
          transform: [{ translateX: contentAnimation }],
        },
      ]}
    >
    <View style={{ flexDirection: "row" }}>
      <Image
        source={require("../assets/adaptive-icon.png")}
        style={styles.logo}
```

```
      />
      <Text style={styles.title}>News</Text>
      <TouchableOpacity style={styles.hamMenu} onPress={toggleMenu}>
          <Image
            style={styles.imageStyle}
            source={require("../assets/profile.png")}
          />
        </TouchableOpacity>
    </View>
    <Seperator />
    <ScrollView
      style={styles.newsScroll}
      contentContainerStyle={styles.contentContainer}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={() => {}} />
      }
    >
      {newsArticles.map((article, index) => (
        <TouchableOpacity
          key={index}
          onPress={() => openLink(article.link)}
          style={index === 0 ? styles.featuredArticle : styles.article}
        >
          <Image
            source={{
              uri: article.image_url || "https://via.placeholder.com/120",
            }}
            style={index === 0 ? styles.featuredImage : styles.articleImage}
          />
          <Text
            style={index === 0 ? styles.featuredTitle : styles.articleTitle}
          >
            {article.title}
          </Text>
        </TouchableOpacity>
      ))}
    </ScrollView>
    </Animated.View>
  </View>
  );
};
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    marginBottom: 40,
    padding: 10,
    backgroundColor: "#fff",
  },
  hamMenu: {
    position: "absolute",
    right: 0,
    top: 31,
    padding: 20,
  },
  imageStyle: {
    width: 30,
    height: 30,
  },
  menu: {
    position: 'absolute',
    backgroundColor: '#E4E3E3',
    width: 300,
    height: '100%',
    right: 0,
    padding: 20,
    zIndex: 2,
  },
  welcome: {
    fontSize: responsiveFontSize(4),
    fontWeight: "bold",
    marginTop: 70,
  },
  content: {
    marginTop: 10,
    paddingTop: 40,
  },
  box: {
    padding: 10,
    marginTop: 60,
    borderRadius: 20,
    backgroundColor: "#D3D3D3",
  },
  text1: {
```

```
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text2: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
  paddingTop: 10,
  paddingBottom: 10,
},
text3: {
  fontSize: responsiveFontSize(1.4),
  fontWeight: "bold",
  color: "white",
},
links: {
  flexDirection: "row",
  padding: 10,
  justifyContent: "space-around",
  marginTop: 100,
},
socialM: {
  width: responsiveWidth(5),
  height: responsiveWidth(5),
},
logo: {
  width: responsiveWidth(9),
  height: responsiveWidth(9),
  marginTop: responsiveScreenHeight(5),
},
title: {
  fontSize: responsiveFontSize(2.3),
  fontWeight: "bold",
  color: "#000",
  paddingHorizontal: responsiveWidth(0.3),
  marginTop: responsiveScreenHeight(5.8),
},
Seperator: {
  height: 1,
```

```
    backgroundColor: "#FFD464",
    width: "100%",
    marginVertical: 8,
  },
  newsScroll: {
    paddingTop: 20,
    marginTop: 20,
    marginBottom: 60,
  },
  contentContainer: {
    paddingBottom: 30,
  },
  featuredArticle: {
    marginHorizontal: 10,
    marginTop: 10,
    borderRadius: 20,
    overflow: "hidden",
    backgroundColor: "white",
    elevation: 5,
    shadowColor: "#000",
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 20,
    borderColor: "#FFD464",
    borderWidth: 1,
  },
  article: {
    borderColor: "#FFD464",
    borderWidth: 1,
    flexDirection: "row",
    alignItems: "center",
    backgroundColor: "white",
    padding: 10,
    marginHorizontal: 10,
    marginTop: 5,
    borderRadius: 10,
    elevation: 3,
    shadowColor: "#000",
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 10,
    overflow: "hidden",
```

```
  },
  featuredImage: {
    borderWidth: 1,
    borderColor: "#FFD464",
    width: "100%",
    height: 200,
    resizeMode: "cover",
  },
  articleImage: {
    width: 80,
    height: 80,
    borderRadius: 10,
    marginRight: 10,
  },
  featuredTitle: {
    fontWeight: "bold",
    fontSize: 24,
    padding: 10,
    textAlign: "center",
  },
  articleTitle: {
    fontSize: 16,
    fontWeight: "bold",
    flex: 1,
  },
});

export default NewsScreen;
```

3.1.1.14 *AccountScreen.js*

```
//AccountScreen.js
import React, { useState, useEffect } from "react";
import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  Alert,
  StyleSheet,
  SafeAreaView,
} from "react-native";
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
import AsyncStorage from "@react-native-async-storage/async-storage";
import { useUser } from "../UserContext";
import {
  responsiveWidth,
  responsiveFontSize,
  responsiveScreenHeight,
} from "react-native-responsive-dimensions";

const AccountScreen = ({ navigation }) => {
  const { user } = useUser();
  const [showPassword, setShowPassword] = useState(false);
  const [enteredPin, setEnteredPin] = useState("");
  const [storedPin, setStoredPin] = useState("");
  const [password, setPassword] = useState("");

  useEffect(() => {
    const fetchPinAndPassword = async () => {
      try {
        const pin = await AsyncStorage.getItem("pin");
        const storedPassword = await AsyncStorage.getItem("password");
        setStoredPin(pin);
        setPassword(storedPassword);
      } catch (error) {
        console.error("Error fetching PIN and password:", error);
      }
    };

    fetchPinAndPassword();
  }, []);

  const handleRevealPassword = () => {
    if (enteredPin === storedPin) {
      setShowPassword(true);
    } else {
      Alert.alert("Incorrect PIN", "The PIN you entered is incorrect.");
    }
  };

  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.title}>Account Information</Text>
      <View style={styles.info}>
```

```
      <View style={styles.infoContainer}>
        <Text style={styles.label}>Username:</Text>
        <Text style={styles.value}>{user.username}</Text>
      </View>
      <View style={styles.infoContainer}>
        <Text style={styles.label}>Email:</Text>
        <Text style={styles.value}>{user.email}</Text>
      </View>
      <View style={styles.infoContainer}>
        <Text style={styles.label}>Password:</Text>
        {showPassword ? (
          <Text style={styles.value}>{password}</Text>
        ) : (
          <Text style={styles.value}>******</Text>
        )}
      </View>
    </View>
    <View style={{ flex: 1.8 }}>
      {!showPassword && (
        <View style={styles.pinContainer}>
          <TextInput
            style={styles.input}
            placeholder="Enter PIN to reveal password"
            value={enteredPin}
            onChangeText={setEnteredPin}
            keyboardType="numeric"
            maxLength={4}
          />
          <TouchableOpacity
            style={styles.button}
            onPress={handleRevealPassword}
          >
            <Text style={styles.buttonText}>Reveal Password</Text>
          </TouchableOpacity>
        </View>
      )}
    </View>
    <TouchableOpacity onPress={() => navigation.navigate("Forgot")}>
      <Text style={styles.forgotPasswordText}>Forgot your password?</Text>
    </TouchableOpacity>
  </SafeAreaView>
);
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: responsiveWidth(5),
    backgroundColor: "white",
  },
  title: {
    fontSize: responsiveFontSize(3),
    fontWeight: "bold",
    marginBottom: responsiveScreenHeight(10),
    paddingHorizontal: responsiveWidth(3),
    marginTop: responsiveScreenHeight(5),
  },
  info: {
    flex: 1,
    marginBottom: responsiveScreenHeight(1),
  },
  infoContainer: {
    marginTop: responsiveScreenHeight(2),
    flexDirection: "row",
    marginBottom: responsiveScreenHeight(2),
    paddingHorizontal: responsiveWidth(3),
  },
  label: {
    fontWeight: "bold",
    flex: 1,
    fontSize: responsiveFontSize(2),
  },
  value: {
    flex: 2,
    fontSize: responsiveFontSize(2),
  },
  pinContainer: {
    flexDirection: "row",
    alignItems: "center",
  },
  input: {
    borderWidth: 1,
    borderColor: "#ccc",
    borderRadius: 5,
```

```
    padding: responsiveWidth(3),
    marginBottom: responsiveScreenHeight(2),
    width: responsiveWidth(60),
    alignSelf: "flex-start",
    marginLeft: responsiveWidth(4),
  },
  button: {
    backgroundColor: "#007bff",
    padding: responsiveWidth(3),
    borderRadius: 5,
    alignItems: "center",
    width: responsiveWidth(34),
    marginBottom: responsiveScreenHeight(2),
    alignSelf: "center",
  },
  buttonText: {
    color: "#fff",
    fontWeight: "bold",
    fontSize: responsiveFontSize(1.6),
  },
  forgotPasswordText: {
    marginTop: responsiveScreenHeight(2),
    color: "#007bff",
    textDecorationLine: "none",
    textAlign: "center",
    fontSize: responsiveFontSize(2),
  },
});

export default AccountScreen;
```

3.1.2 Back-end

3.1.2.1 *schema.sql*

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 16.2 (Debian 16.2-1.pgdg120+2)
```

```
-- Dumped by pg_dump version 16.2 (Debian 16.2-1.pgdg120+2)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: coins; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.coins (
    id integer NOT NULL,
    name character varying(255) NOT NULL,
    symbol character varying(10) NOT NULL,
    market_cap numeric(18,2),
    current_price numeric(18,2),
    last_updated timestamp with time zone,
    image_url character varying(255),
    price_change_percentage_24h character varying(255)
);


--
-- Name: coins_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.coins_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
```

```
    NO MAXVALUE
    CACHE 1;


--
-- Name: coins_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.coins_id_seq OWNED BY public.coins.id;


--
-- Name: conversations; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.conversations (
    id integer NOT NULL,
    user_id integer NOT NULL,
    content jsonb NOT NULL,
    created_at timestamp without time zone DEFAULT now()
);


--
-- Name: conversations_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.conversations_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: conversations_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.conversations_id_seq OWNED BY public.conversations.id;
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
--
-- Name: conversion_history; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.conversion_history (
    id integer NOT NULL,
    user_id integer NOT NULL,
    base_currency character varying(255),
    target_currency character varying(255),
    base_amount numeric,
    target_amount numeric,
    created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);


--
-- Name: conversion_history_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.conversion_history_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: conversion_history_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.conversion_history_id_seq OWNED BY public.conversion_history.id;


--
-- Name: conversion_rates; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.conversion_rates (
    id integer NOT NULL,
    base_currency character varying(10) NOT NULL,
```

```
    target_currency character varying(10) NOT NULL,
    rate numeric(18,8),
    last_updated timestamp with time zone
);


--
-- Name: conversion_rates_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.conversion_rates_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: conversion_rates_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.conversion_rates_id_seq OWNED BY public.conversion_rates.id;


--
-- Name: exchanges; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.exchanges (
    id integer NOT NULL,
    name character varying(255) NOT NULL,
    url character varying(255) NOT NULL,
    trust_score numeric(18,2),
    image_url character varying(255),
    trade_volume_24h_btc numeric
);


--
-- Name: exchanges_id_seq; Type: SEQUENCE; Schema: public; Owner: -
```

```
--

CREATE SEQUENCE public.exchanges_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: exchanges_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.exchanges_id_seq OWNED BY public.exchanges.id;


--
-- Name: news; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.news (
    id integer NOT NULL,
    title character varying(255) NOT NULL,
    link character varying(255) NOT NULL,
    summary text,
    published_at timestamp with time zone,
    image_url character varying(255)
);


--
-- Name: news_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.news_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1;


--
-- Name: news_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.news_id_seq OWNED BY public.news.id;


--
-- Name: pins; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.pins (
    id integer NOT NULL,
    user_id integer,
    pin character varying(6) NOT NULL,
    created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP
);


--
-- Name: pins_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.pins_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: pins_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.pins_id_seq OWNED BY public.pins.id;
```

```
--
-- Name: users; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.users (
    id integer NOT NULL,
    username character varying(255) NOT NULL,
    password character varying(255) NOT NULL,
    email character varying(255) NOT NULL,
    created_at timestamp with time zone DEFAULT CURRENT_TIMESTAMP,
    last_access timestamp with time zone
);


--
-- Name: users_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.users_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: users_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.users_id_seq OWNED BY public.users.id;


--
-- Name: coins id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.coins ALTER COLUMN id SET DEFAULT
nextval('public.coins_id_seq'::regclass);
```

```
--
-- Name: conversations id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversations ALTER COLUMN id SET DEFAULT
nextval('public.conversations_id_seq'::regclass);


--
-- Name: conversion_history id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversion_history ALTER COLUMN id SET DEFAULT
nextval('public.conversion_history_id_seq'::regclass);


--
-- Name: conversion_rates id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversion_rates ALTER COLUMN id SET DEFAULT
nextval('public.conversion_rates_id_seq'::regclass);


--
-- Name: exchanges id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.exchanges ALTER COLUMN id SET DEFAULT
nextval('public.exchanges_id_seq'::regclass);


--
-- Name: news id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.news ALTER COLUMN id SET DEFAULT
nextval('public.news_id_seq'::regclass);


--
-- Name: pins id; Type: DEFAULT; Schema: public; Owner: -
```

```
--

ALTER TABLE ONLY public.pins ALTER COLUMN id SET DEFAULT nextval('public.pins_id_seq'::regclass);


--
-- Name: users id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.users ALTER COLUMN id SET DEFAULT
nextval('public.users_id_seq'::regclass);


--
-- Name: coins coins_name_unique; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.coins
    ADD CONSTRAINT coins_name_unique UNIQUE (name);


--
-- Name: coins coins_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.coins
    ADD CONSTRAINT coins_pkey PRIMARY KEY (id);


--
-- Name: conversations conversations_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversations
    ADD CONSTRAINT conversations_pkey PRIMARY KEY (id);


--
-- Name: conversion_history conversion_history_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversion_history
```

```
    ADD CONSTRAINT conversion_history_pkey PRIMARY KEY (id);


--
-- Name: conversion_rates conversion_rates_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversion_rates
    ADD CONSTRAINT conversion_rates_pkey PRIMARY KEY (id);


--
-- Name: exchanges exchanges_name_unique; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.exchanges
    ADD CONSTRAINT exchanges_name_unique UNIQUE (name);


--
-- Name: exchanges exchanges_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.exchanges
    ADD CONSTRAINT exchanges_pkey PRIMARY KEY (id);


--
-- Name: news news_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.news
    ADD CONSTRAINT news_pkey PRIMARY KEY (id);


--
-- Name: news news_title_unique; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.news
    ADD CONSTRAINT news_title_unique UNIQUE (title);
```

```
--
-- Name: pins pins_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.pins
    ADD CONSTRAINT pins_pkey PRIMARY KEY (id);


--
-- Name: conversion_rates unique_base_target_constraint; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversion_rates
    ADD CONSTRAINT unique_base_target_constraint UNIQUE (base_currency, target_currency);


--
-- Name: pins user_id_unique; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.pins
    ADD CONSTRAINT user_id_unique UNIQUE (user_id);


--
-- Name: users users_email_key; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_email_key UNIQUE (email);


--
-- Name: users users_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_pkey PRIMARY KEY (id);


--
-- Name: users users_username_key; Type: CONSTRAINT; Schema: public; Owner: -
```

```
--

ALTER TABLE ONLY public.users
    ADD CONSTRAINT users_username_key UNIQUE (username);


--
-- Name: conversations conversations_user_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.conversations
    ADD CONSTRAINT conversations_user_id_fkey FOREIGN KEY (user_id) REFERENCES public.users(id);


--
-- Name: conversion_history conversion_history_user_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: -
--

ALTER TABLE ONLY public.conversion_history
    ADD CONSTRAINT conversion_history_user_id_fkey FOREIGN KEY (user_id) REFERENCES
public.users(id);


--
-- Name: pins pins_user_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.pins
    ADD CONSTRAINT pins_user_id_fkey FOREIGN KEY (user_id) REFERENCES public.users(id) ON
DELETE CASCADE;


--
-- PostgreSQL database dump complete
--
```

3.1.2.2 *api.js*

```
// api.js
const express = require('express');
const cors = require('cors');
```

```
const { Pool } = require('pg');
const bcrypt = require('bcryptjs');
const axios = require('axios');

const app = express();
app.use(cors());
app.use(express.json());

const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'cryptrail',
  password: 'admin123',
  port: 5432,
});


app.post('/register', async (req, res) => {
  const { username, password, email } = req.body;

  if (!username || !password || !email) {
    return res.status(400).json({ error: 'Missing required fields' });
  }

  try {
    const hashedPassword = await bcrypt.hash(password, 10);
    const randomPin = Math.floor(1000 + Math.random() * 9000).toString();

    await pool.query('BEGIN');

    const userResult = await pool.query(
      'INSERT INTO users (username, password, email) VALUES ($1, $2, $3) RETURNING id',
      [username, hashedPassword, email]
    );
    const userId = userResult.rows[0].id;

    await pool.query(
      'INSERT INTO pins (user_id, pin) VALUES ($1, $2)',
      [userId, randomPin]
    );

    await pool.query('COMMIT');
```

```
    res.status(201).json({ userId: userId, pin: randomPin });
  } catch (err) {
    await pool.query('ROLLBACK');
    console.error(err);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});

  app.post('/login', async (req, res) => {
    const { email, password } = req.body;
    try {
        const result = await pool.query('SELECT * FROM users WHERE email = $1', [email]);
        if (result.rows.length === 0) {
            return res.status(401).json({ error: 'Invalid email or password' });
        }

        const user = result.rows[0];
        const isValid = await bcrypt.compare(password, user.password);
        if (!isValid) {
            return res.status(401).json({ error: 'Invalid email or password' });
        }

        await pool.query('UPDATE users SET last_access = NOW() WHERE id = $1', [user.id]);

        const { password: _, ...userData } = user;
        res.status(200).json({ message: 'Login successful', user: userData });
    } catch (err) {
        console.error(err);
        res.status(500).json({ error: 'Internal Server Error' });
    }
});

app.post('/resetPassword', async (req, res) => {
  const { email, pin, newPassword } = req.body;

  if (!email || !pin || !newPassword) {
    return res.status(400).json({ message: 'Missing required parameters' });
  }

  try {
    const userResult = await pool.query('SELECT id FROM users WHERE email = $1', [email]);
```

```
    if (userResult.rows.length === 0) {
      return res.status(404).json({ message: 'User not found' });
    }

    const userId = userResult.rows[0].id;

    const pinResult = await pool.query('SELECT pin FROM pins WHERE user_id = $1', [userId]);
    if (pinResult.rows.length === 0 || pinResult.rows[0].pin !== pin) {
      return res.status(401).json({ message: 'Invalid PIN' });
    }
    const hashedPassword = await bcrypt.hash(newPassword, 10);

    await pool.query('UPDATE users SET password = $1 WHERE id = $2', [hashedPassword, userId]);

    res.status(200).json({ success: true, message: 'Password reset successfully' });
  } catch (error) {
    console.error('Error resetting password:', error);
    res.status(500).json({ message: 'Failed to reset password', error: error.message });
  }
});


app.delete('/conversations/:userId', async (req, res) => {
  const userId = parseInt(req.params.userId);
  if (isNaN(userId)) {
    return res.status(400).json({ message: "Invalid user ID format" });
  }

  try {
    await pool.query('DELETE FROM conversations WHERE user_id = $1', [userId]);
    res.status(200).json({ message: 'Conversations deleted successfully' });
  } catch (error) {
    console.error('Database or server error:', error);
    res.status(500).json({ message: 'Failed to delete conversations', error: error.message });
  }
});

app.get('/conversations/:userId', async (req, res) => {
  const userId = parseInt(req.params.userId);
  if (isNaN(userId)) {
    return res.status(400).json({ message: "Invalid user ID format" });
  }
```

```
  try {
    const result = await pool.query('SELECT * FROM conversations WHERE user_id = $1', [userId]);
    if (result.rows.length > 0) {
      console.log(result.rows[0]);
      const conversations = result.rows[0].content.map(row => {
        return {
          id: result.rows[0].id,
          role: row.role,
          content: row.content,
          createdAt: result.rows[0].created_at
        };
      });
      res.json(conversations);
    } else {
      res.status(200).json([]);
    }
  } catch (error) {
    console.error('Database or server error:', error);
    res.status(500).json({ message: 'Failed to fetch conversations', error: error.message });
  }
});


app.post('/saveConversation', async (req, res) => {
  const { userId, content } = req.body;
  if (!userId) {
    return res.status(400).json({ message: "User ID is required" });
  }
  try {
    const result = await pool.query('INSERT INTO conversations (id, user_id, content) VALUES ($1, $2, $3) ON
CONFLICT(id) DO UPDATE SET content = EXCLUDED.content RETURNING id;', [userId, userId,
JSON.stringify(content)]);
    res.json({ conversationId: result.rows[0].id });
  } catch (error) {
    console.error('Error saving conversation:', error);
    res.status(500).json({ message: 'Failed to save conversation', error: error.message });
  }
});

app.post('/saveConversion', async (req, res) => {
  const { userId, baseCurrency, targetCurrency, baseAmount, targetAmount } = req.body;
```

```
  if (!userId || !baseCurrency || !targetCurrency || !baseAmount || !targetAmount) {
    return res.status(400).json({ error: 'Missing required parameters' });
  }

  try {
    const result = await pool.query(
      'INSERT INTO conversion_history (user_id, base_currency, target_currency, base_amount,
target_amount) VALUES ($1, $2, $3, $4, $5)',
      [userId, baseCurrency, targetCurrency, baseAmount, targetAmount]
    );
    res.status(201).json({ message: 'Conversion history saved successfully' });
  } catch (error) {
    console.error('Error saving conversion history:', error);
    res.status(500).json({ error: 'Failed to save conversion history', message: error.message });
  }
});


app.get('/getConversions/:userId', async (req, res) => {
  const userId = parseInt(req.params.userId);
  if (!userId) {
    return res.status(400).json({ message: "Invalid user ID format" });
  }

  try {
    const result = await pool.query('SELECT * FROM conversion_history WHERE user_id = $1', [userId]);
    res.json(result.rows);
  } catch (error) {
    console.error('Database error:', error);
    res.status(500).json({ message: 'Failed to fetch conversion history', error: error.message });
  }
});

app.delete('/conversions/:userId', async (req, res) => {
  const userId = parseInt(req.params.userId, 10);
  console.log(`Received userId: ${req.params.userId}, Parsed userId: ${userId}`);

  if (isNaN(userId)) {
    console.error(`Invalid user ID format: ${req.params.userId}`);
    return res.status(400).json({ message: "Invalid user ID format" });
  }
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
  try {
    const deleteResult = await pool.query('DELETE FROM conversion_history WHERE user_id = $1', [userId]);
    if (deleteResult.rowCount > 0) {
      res.status(200).json({ message: 'Conversion history deleted successfully' });
    } else {
      res.status(404).json({ message: 'No conversion history found for the user' });
    }
  } catch (error) {
    console.error('Database or server error:', error);
    res.status(500).json({ message: 'Failed to delete conversion history', error: error.message });
  }
});

app.get('/news', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM news ORDER BY published_at DESC');
    console.log(result.rows);
    res.json(result.rows);
  } catch (error) {
    console.error('Error fetching news from the database:', error);
    res.status(500).json({ message: 'Failed to fetch news from the database' });
  }
});

app.get('/coins', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM coins ORDER BY market_cap DESC LIMIT 5000');
    res.json(result.rows);
  } catch (error) {
    console.error('Error fetching coins data from the database:', error);
    res.status(500).json({ message: 'Failed to fetch coins data from the database' });
  }
});

app.get('/exchanges', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM exchanges LIMIT 5000');
    res.json(result.rows);
  } catch (error) {
    console.error('Error fetching exchanges data from the database:', error);
    res.status(500).json({ message: 'Failed to fetch exchanges data from the database' });
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
  }
 });

 app.get('/rates', async (req, res) => {
  try {
   const result = await pool.query('SELECT * FROM conversion_rates');
   const formattedRates = result.rows.reduce((acc, curr) => {
    acc[curr.target_currency] = { ...acc[curr.target_currency], rate: curr.rate, name:
curr.target_currency.toUpperCase() };
    return acc;
   }, {});
   res.json(formattedRates);
  } catch (error) {
   console.error('Error fetching rates from the database:', error);
   res.status(500).json({ message: 'Failed to fetch rates from the database' });
  }
 });

 app.get('/convert', async (req, res) => {
  const { base, target, amount } = req.query;

  if (!base || !target || !amount) {
   return res.status(400).json({ message: 'Missing required query parameters' });
  }
 console.log('Converting:', base, target, amount)
  try {
   let baseToBTCRate = 1;
   if (base.toLowerCase() !== 'btc') {
    const baseToBTCResult = await pool.query('SELECT rate FROM conversion_rates WHERE
target_currency = $1', [base.toLowerCase()]);
    if (baseToBTCResult.rows.length === 0) {
     return res.status(404).json({ message: `Rate not found for base currency: ${base}` });
    }
    baseToBTCRate = 1 / parseFloat(baseToBTCResult.rows[0].rate);
   }

   let btcToTargetRate = 1;
   if (target.toLowerCase() !== 'btc') {
    const btcToTargetResult = await pool.query('SELECT rate FROM conversion_rates WHERE
target_currency = $1', [target.toLowerCase()]);
    if (btcToTargetResult.rows.length === 0) {
     return res.status(404).json({ message: `Rate not found for target currency: ${target}` });
```

```
      }
      btcToTargetRate = parseFloat(btcToTargetResult.rows[0].rate);
    }

    const convertedAmount = (parseFloat(amount) * baseToBTCRate * btcToTargetRate).toFixed(6);

    res.json({ convertedAmount });
  } catch (error) {
    console.error('Error during conversion:', error);
    res.status(500).json({ message: 'Failed to perform conversion', error: error.message });
  }
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

3.1.2.3 *docker-compose.yml*

```
version: '3.8'
services:
  db:
    image: postgres
    container_name: cryptrail
    environment:
      POSTGRES_PASSWORD: admin123
    volumes:
      - ./init:/docker-entrypoint-initdb.d
      - postgres_data:/var/lib/postgresql/data
    ports:
      - "5432:5432"

volumes:
  postgres_data:
```

3.1.2.4 *compose-down.sh*

```
#!/bin/bash
```

```
set -e

CONTAINER_NAME="cryptrail"
DB_USER="postgres"
DB_NAME="cryptrail"
SCHEMA_FILE="./init/schema.sql"

docker exec $CONTAINER_NAME pg_dump -U $DB_USER --schema-only --no-owner $DB_NAME >
$SCHEMA_FILE

echo "Schema exported to $SCHEMA_FILE"

docker-compose down
```

3.1.2.5 *Dockerfile*

```
FROM postgres:latest
ENV POSTGRES_DB=cryptrail
ENV POSTGRES_USER=postgres
ENV POSTGRES_PASSWORD=admin123
```

3.1.2.6 *Import-schema.sql*

```
#!/bin/bash
set -e

CONTAINER_NAME="cryptrail"
DB_USER="postgres"
DB_NAME="cryptrail"
SCHEMA_FILE="./init/schema.sql"

until docker exec $CONTAINER_NAME pg_isready -U $DB_USER; do
  echo "Waiting for PostgreSQL to become ready..."
  sleep 1
done

echo "Importing schema to $DB_NAME"
cat $SCHEMA_FILE | docker exec -i $CONTAINER_NAME psql -U $DB_USER -d $DB_NAME
```

3.1.2.7 *Info-fetcher.js*

```
// info-fetcher.js
const { Pool } = require('pg');
const axios = require('axios');

const NEWSDATA_API_URL = 'https://newsdata.io/api/1/news';
const NEWSDATA_API_KEY='pub_obfuscated;
const OPENAI_API_KEY='sk-obfuscated;

const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'cryptrail',
  password: 'admin123',
  port: 5432,
});

// Endpoint para as notícias
async function fetchNews() {
  console.log('Fetching news...');
  try {
    const response = await axios.get(NEWSDATA_API_URL, {
      params: {
        apikey: NEWSDATA_API_KEY,
        q: 'cryptocurrency',
        country: 'us',
        language: 'en'
      }
    });
    if (response.data.results) {
      const articles = response.data.results;
      console.log('Fetched news:', articles.length);
      for (let article of articles) {
        console.log('Inserting article:', article);
        await pool.query(`INSERT INTO news (title, image_url, link, summary, published_at)
        VALUES ($1, $2, $3, $4, $5)
        ON CONFLICT (title) DO UPDATE SET
        image_url = EXCLUDED.image_url,
        link = EXCLUDED.link,
        summary = EXCLUDED.summary,
        published_at = EXCLUDED.published_at`,
        [article.title, article.image_url, article.link, article.summary, article.published_at]
        );
```

```
      }
    }
  } catch (error) {
    console.error('Error fetching news:', error);
  }
};

async function fetchCoins() {
  console.log('Fetching coins...');
  try {
    const response = await axios.get('https://api.coingecko.com/api/v3/coins/markets?vs_currency=eur', {
    params: {
      order: 'market_cap_desc',
      per_page: 5000,
      page: 1,
      sparkline: false,
      vs_currency: 'eur',
    }
  });
  if (response.data) {
    const assets = response.data;
    console.log('Fetched coins:', assets.length);
    for (let asset of assets) {
      console.log('Inserting coins:', asset);
  if (!asset.name || !asset.symbol || !asset.market_cap || !asset.current_price || !asset.last_updated ||
!asset.image || asset.price_change_percentage_24h === null) {
    console.error('Invalid asset data:', asset);
    continue;
  }
      await pool.query(`INSERT INTO coins (name, symbol, market_cap, current_price, last_updated,
image_url, price_change_percentage_24h)
      VALUES ($1, $2, $3, $4, $5, $6, $7)
      ON CONFLICT (name) DO UPDATE SET
      symbol = EXCLUDED.symbol,
      market_cap = EXCLUDED.market_cap,
      current_price = EXCLUDED.current_price,
      last_updated = EXCLUDED.last_updated,
      image_url = EXCLUDED.image_url`,
      [asset.name, asset.symbol, asset.market_cap, asset.current_price, asset.last_updated, asset.image,
asset.price_change_percentage_24h]
      );
    }
```

```
  }
} catch (error) {
  console.error('Error fetching coins data:', error);
}
};

async function fetchExchanges() {
  console.log('Fetching exchanges...');
  try {
    const response = await axios.get('https://api.coingecko.com/api/v3/exchanges', {
      params: {
        vs_currency: 'eur',
        per_page: 5000,
        page: 1,
      },
    });
    if (response.data) {
      const exchanges = response.data;
      console.log('Fetched exchanges:', exchanges.length);
      for (let exchange of exchanges) {
        if (exchange.url) {
          console.log('Inserting exchange:', exchange.name);
          await pool.query(`INSERT INTO exchanges (name, url, trust_score, image_url, trade_volume_24h_btc)
          VALUES ($1, $2, $3, $4, $5)
          ON CONFLICT (name) DO UPDATE SET
          url = EXCLUDED.url,
          trust_score = EXCLUDED.trust_score,
          image_url = EXCLUDED.image_url,
          trade_volume_24h_btc = EXCLUDED.trade_volume_24h_btc`,
          [exchange.name, exchange.url, exchange.trust_score, exchange.image,
exchange.trade_volume_24h_btc]
          );
        } else {
          console.log('Skipping exchange with null url:', exchange.name);
        }
      }
    }
  } catch (error) {
    console.error('Error fetching exchanges data:', error);
  }
};
```

```
async function fetchAndStoreRates() {
  console.log('Fetching exchange rates...');
  try {
    const response = await axios.get('https://api.coingecko.com/api/v3/exchange_rates');
    const rates = response.data.rates;

    for (const [currency, data] of Object.entries(rates)) {
      console.log(`Inserting rate for ${currency}:`, data.value);
      await pool.query(`
        INSERT INTO conversion_rates (base_currency, target_currency, rate, last_updated)
        VALUES ($1, $2, $3, NOW())
        ON CONFLICT (base_currency, target_currency) DO UPDATE SET
        rate = EXCLUDED.rate, last_updated = NOW()`,
        ['BTC', currency, data.value]
      );
    }
  } catch (error) {
    console.error('Error fetching and storing rates:', error);
  }
};



(async () => {
  try {
    //await fetchNews();
    //await fetchCoins();
    //await fetchExchanges();
    //await fetchAndStoreRates();
  } catch (err) {
    console.error(err);
  }
})();

setInterval(() => {
  //fetchNews().catch(err => console.error(err));
  //fetchCoins().catch(err => console.error(err));
  //fetchExchanges().catch(err => console.error(err));
  //fetchAndStoreRates().catch(err => console.error(err));
}, 24 * 60 * 60 * 1000);
```

3.2 Website

3.2.1 HTML

3.2.1.1 *index.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Cryptrail</title>
</head>
<body>

    <div style="flex-direction: row">
        <a href="index.html"><img src="./adaptive-icon.png" alt="Cryptrail" class="logo"></a>
        <a class="links" href="terms.html">Terms and Conditions</a>
        <a class="links" href="politics.html">Privacy Policy</a>
        <a class="links" href="support.html">Support</a>
        <a class="links" href="about.html">About us</a>
        <img src="./hamenu.png" alt="hamburguer" class="ham" onclick="toggleMenu()">
        <div class="dropdown" id="dropdownMenu">
            <a href="terms.html">Terms and Conditions</a>
            <a href="politics.html">Privacy Policy</a>
            <a href="support.html">Support</a>
            <a href="about.html">About us</a>
        </div>
    </div>


    <header>
        <h1>Welcome to Cryptrail</h1>
        <h2>Come discover your crypto journey</h2>
    </header>
```

```
    <main>
        <p class="store">Soon Available in <br> <img src="./appstore.png" class="app" alt="Apple Store"></p>
        <img src="./mockup.png" alt="Phone Image" class="mockupr">
        <img src="./rmockup.png" alt="Phone Image" class="mockup">
        <img src="./adaptive-icon.png" alt="Cryptrail" class="logo2">
    </main>


    <script src="script.js"></script>
</body>
</html>
```

3.2.1.2 *terms.html*

```
<!DOCTYPE html>
<html lang en="en">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cryptrail</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div style="flex-direction: row">
        <a href="index.html"><img src="./adaptive-icon.png" alt="Cryptrail" class="logo"></a>
        <a class="links" href="terms.html">Terms and Conditions</a>
        <a class="links" href="politics.html">Privacy Policy</a>
        <a class="links" href="support.html">Support</a>
        <a class="links" href="about.html">About us</a>
        <img src="./hamenu.png" alt="hamburguer" class="ham" onclick="toggleMenu()">
        <div class="dropdown" id="dropdownMenu">
            <a href="terms.html">Terms and Conditions</a>
            <a href="politics.html">Privacy Policy</a>
            <a href="support.html">Support</a>
            <a href="about.html">About us</a>
        </div>
```

```
  </div>


<div>
  <h1>Terms & Conditions</h1>
  <h2>Learn more about us.</h2>

  <div class="tc">
    <p><strong>Last Updated: 11 May 2024</strong></p>

    <p>Welcome to Cryptrail! These Terms and Conditions ("Terms") govern your use of the Cryptrail application  . By accessing or using the App, you agree to comply with and be bound by these Terms. Please read them carefully.</p>

    <p><strong>1. Acceptance of Terms</strong></p>
    <p>By accessing or using Cryptrail, you accept and agree to be bound by these Terms and our Privacy Policy. If you do not agree, you may not use the App.</p>

    <p><strong>2. Services Provided</strong></p>
    <p>Cryptrail offers the following services:</p>
    <ul>
      <li><strong>Currency Converter:</strong> Converts fiat currency to cryptocurrency and vice versa. Provides options to buy cryptocurrencies through third-party exchanges.</li>
      <li><strong>Trail Assistant:</strong> An AI-based chatbot optimized for cryptocurrency-related inquiries.</li>
      <li><strong>Market Data:</strong> Displays the latest market prices, market value, price trends, and volume for various cryptocurrencies and exchanges.</li>
      <li><strong>News Updates:</strong> Provides up-to-date news related to digital currencies.</li>
    </ul>

    <p><strong>3. User Responsibilities</strong></p>
    <ul>
      <li><strong>Accuracy of Information:</strong> You agree to provide accurate, current, and complete information during registration and update it as necessary.</li>
      <li><strong>Compliance with Laws:</strong> You agree to use the App in compliance with all applicable laws and regulations.</li>
    </ul>

    <p><strong>4. Currency Converter and Exchanges</strong></p>
    <ul>
      <li><strong>Conversion Information:</strong> The currency converter provides estimated conversion rates. Actual rates may vary.</li>
```

Third-Party Exchanges: Cryptrail redirects users to third-party exchanges for purchasing cryptocurrencies. We do not endorse or assume responsibility for these exchanges.
Transaction Fees: Information on fees is provided by the exchanges. Cryptrail does not charge any additional fees for conversions or redirections.

5. Trail Assistant

AI Chatbot: The Trail Assistant is designed to provide information related to cryptocurrencies. It is not a financial advisor.
Accuracy of Information: While we strive to provide accurate information, Cryptrail does not guarantee the completeness or accuracy of responses from the Trail Assistant.

6. Market Data

Real-Time Information: Market data is updated regularly. However, Cryptrail is not responsible for any delays or inaccuracies in the data provided.
Exchanges Ratings: Ratings of exchanges are based on available data and are subject to change.

7. News Updates

Content: News articles are updated every 25 hours. The first news item is highlighted as the most important of the day.
Sources: Cryptrail aggregates news from various sources and does not take responsibility for the accuracy of the news provided.

8. User Account and Security

Account Creation: You may be required to create an account to use certain features of the App.
Security: You are responsible for maintaining the confidentiality of your account information and for all activities that occur under your account.

9. Intellectual Property

Ownership: All content, trademarks, and data on Cryptrail are the property of Cryptrail or its licensors and are protected by intellectual property laws.

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Usage: You may not reproduce, distribute, or create derivative works from the content without express permission.

10. Limitation of Liability

No Warranty: Cryptrail is provided on an "as-is" basis without any warranties of any kind.
No Liability: Cryptrail is not liable for any damages arising from the use or inability to use the App, including but not limited to direct, indirect, incidental, or consequential damages.

11. Indemnification
You agree to indemnify and hold Cryptrail, its affiliates, and employees harmless from any claims, damages, or expenses arising from your use of the App or violation of these Terms.

12. Changes to Terms
Cryptrail reserves the right to modify these Terms at any time. Changes will be effective immediately upon posting. Continued use of the App signifies acceptance of the revised Terms.

13. Governing Law
These Terms are governed by and construed in accordance with the laws of internet data privacy, without regard to its conflict of law principles.

14. Contact Information
For any questions or concerns regarding these Terms, please contact us at:

Email: cryptrailpt@gmail.com

15. Miscellaneous

Entire Agreement: These Terms constitute the entire agreement between you and Cryptrail regarding the use of the App.
Severability: If any provision of these Terms is found to be invalid, the remaining provisions will continue in full force and effect.
Waiver: No waiver of any term shall be deemed a further or continuing waiver of such term or any other term.

Thank you for using Cryptrail! We hope you enjoy our services.

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    </div>
</div>

<script src="script.js"></script>
</body>
</html>
```

3.2.1.3 *politics.html*

```
<!DOCTYPE html>
<html lang en="en">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cryptrail</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div style="flex-direction: row">
        <a href="index.html"><img src="./adaptive-icon.png" alt="Cryptrail" class="logo"></a>
        <a class="links" href="terms.html">Terms and Conditions</a>
        <a class="links" href="politics.html">Privacy Policy</a>
        <a class="links" href="support.html">Support</a>
        <a class="links" href="about.html">About us</a>
        <img src="./hamenu.png" alt="hamburguer" class="ham" onclick="toggleMenu()">
        <div class="dropdown" id="dropdownMenu">
            <a href="terms.html">Terms and Conditions</a>
            <a href="politics.html">Privacy Policy</a>
            <a href="support.html">Support</a>
            <a href="about.html">About us</a>
        </div>
    </div>


<div>
```

```
<h1>Privacy Policy</h1>
<h2>Stay safe with us.</h2>

<div class="tc">
    <p><strong>Last Updated: 11 May 2024</strong></p>

    <p>At Cryptrail, we are committed to protecting your privacy and ensuring the security of your personal data. This Privacy Policy outlines how we collect, use, and protect your information when you use our application  and website.</p>

    <p><strong>1. Information We Collect</strong></p>
    <ul>
        <li><strong>Personal Information:</strong> When you register or use the App, we may collect personal information such as your name, email address, and contact details.</li>
        <li><strong>Usage Data:</strong> We collect information on how you use the App, including interactions with features, pages visited, and other usage patterns.</li>
        <li><strong>Device Information:</strong> Information about the device you use, including device model, operating system version, unique device identifiers, and IP address.</li>
    </ul>

    <p><strong>2. Use of Information</strong></p>
    <ul>
        <li><strong>Providing Services:</strong> To operate and provide the features of the App.</li>
        <li><strong>Improving User Experience:</strong> To analyze usage patterns and improve the functionality and user experience of the App.</li>
        <li><strong>Communication:</strong> To send you updates, notifications, and other information related to your use of the App.</li>
        <li><strong>Security and Compliance:</strong> To ensure the security of our services and comply with legal obligations.</li>
    </ul>

    <p><strong>3. Data Protection and Security</strong></p>
    <ul>
        <li><strong>Encryption:</strong> All user data is stored in an encrypted format to ensure its security and confidentiality.</li>
        <li><strong>Access Control:</strong> Access to personal data is restricted to authorized personnel only, ensuring that your information is handled securely.</li>
        <li><strong>Security Measures:</strong> We employ a range of security measures, including encryption, firewalls, and secure servers to protect your data.</li>
    </ul>

    <p><strong>4. Sharing of Information</strong></p>
```


Third-Party Services: We may share your data with third-party service providers who assist us in operating the App, provided they adhere to data protection standards.
Legal Requirements: We may disclose your information if required by law or in response to legal requests.

5. User Rights

Access and Correction: You have the right to access and update your personal information at any time.
Data Deletion: You can request the deletion of your personal data, subject to legal and operational constraints.
Opt-Out: You may opt out of receiving promotional communications from us by following the unsubscribe instructions provided in those communications.

6. Cookies and Tracking Technologies

Cookies: We use cookies and similar technologies to enhance your experience, analyze usage patterns, and improve our services.
Managing Cookies: You can manage your cookie preferences through your browser settings.

7. Children's Privacy
Our services are not intended for individuals under the age of 18. We do not knowingly collect personal information from children under 18.

8. International Data Transfers
Your information may be transferred to and maintained on servers located outside your state, province, or country, where data protection laws may differ. We take steps to ensure that your data is protected in accordance with this Privacy Policy.

9. Changes to This Privacy Policy
We may update this Privacy Policy from time to time. Any changes will be posted on this page, and we will notify you of significant changes via email or through the App.

10. Contact Information
If you have any questions or concerns regarding this Privacy Policy, please contact us at:


```
      <li><strong>Email:</strong> <a style="color: black"
href="mailto:cryptrailpt@gmail.com">cryptrailpt@gmail.com</a></li>
    </ul>

    <p><strong>11. Minimum System Requirements</strong></p>
    <ul>
      <li><strong>Operating System:</strong> iOS 17.4 or later</li>
      <li><strong>Device Compatibility:</strong> Available for iOS devices only</li>
    </ul>

    <p><strong>12. Consent</strong></p>
    <p>By using the Cryptrail App, you consent to the collection, use, and sharing of your information as
described in this Privacy Policy.</p>
    </div>
</div>

<script src="script.js"></script>

</body>
</html>
```

3.2.1.4 *about.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Cryptrail</title>
</head>
<body>

    <div style="flex-direction: row">
      <a href="index.html"><img src="./adaptive-icon.png" alt="Cryptrail" class="logo"></a>
      <a class="links" href="terms.html">Terms and Conditions</a>
```

```
    <a class="links" href="politics.html">Privacy Policy</a>
    <a class="links" href="support.html">Support</a>
    <a class="links" href="about.html">About us</a>
    <img src="./hamenu.png" alt="hamburger" class="ham" onclick="toggleMenu()">
    <div class="dropdown" id="dropdownMenu">
      <a href="terms.html">Terms and Conditions</a>
      <a href="politics.html">Privacy Policy</a>
      <a href="support.html">Support</a>
      <a href="about.html">About us</a>
    </div>
  </div>

  <h1>About Us</h1>
  <h2>Who are we?</h2>

  <div style="flex-direction: row; padding-left: 10px; padding-right: 10px; margin-top: -50px">
    <img src="./bla.png" class="hand" alt="Illustration">
    <div class="about">
      Welcome to Cryptrail! We are a dedicated team of cryptocurrency enthusiasts and experts committed to simplifying the crypto journey for everyone.
      At Cryptrail, our mission is to empower users with the tools and knowledge they need to navigate the complex world of digital currencies.
      <br><br>
      <br><br>
      Our platform offers a comprehensive suite of services designed to meet the needs of both novice and experienced crypto users. These services include:
      <br><br>

      <ul>
        <li><strong>Currency Converter:</strong> Easily convert between different cryptocurrencies and fiat currencies.</li>
        <li><strong>Trail Assistant:</strong> Get personalized assistance to help you make informed decisions in your crypto journey.</li>
        <li><strong>Market Data:</strong> Access real-time data on various cryptocurrencies, including prices, trends, and market capitalization.</li>
        <li><strong>News Updates:</strong> Stay updated with the latest news and trends in the cryptocurrency market.</li>
      </ul>
      <br>
      At Cryptrail, we believe in the transformative power of cryptocurrencies and are dedicated to making this innovative space accessible to all.
```

Join us on this exciting journey and discover the endless possibilities that the world of digital currencies has to offer.

```
        </div>
    </div>

    <script src="script.js"></script>
</body>
</html>
```

3.2.1.5 *support.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900&display=swap" rel="stylesheet">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Cryptrail - Support</title>
</head>
<body>

    <div style="flex-direction: row">
        <a href="index.html"><img src="./adaptive-icon.png" alt="Cryptrail" class="logo"></a>
        <a class="links" href="terms.html">Terms and Conditions</a>
        <a class="links" href="politics.html">Privacy Policy</a>
        <a class="links" href="support.html">Support</a>
        <a class="links" href="about.html">About us</a>
        <img src="./hamenu.png" alt="hamburger" class="ham" onclick="toggleMenu()">
        <div class="dropdown" id="dropdownMenu">
            <a href="terms.html">Terms and Conditions</a>
            <a href="politics.html">Privacy Policy</a>
            <a href="support.html">Support</a>
            <a href="about.html">About us</a>
        </div>
    </div>
```

```
    <header>
        <h1>Support</h1>
        <h2>We're Here to Help</h2>
    </header>

    <main>
        <form class="contact-form" action="mailto:cryptrailpt@example.com" method="post"
enctype="text/plain">
            <input type="text" class="form-input" name="name" placeholder="Your Name" required>
            <input type="email" class="form-input" name="email" placeholder="Your Email" required>
            <textarea class="form-input" name="message" placeholder="How can we help you?"
required></textarea>
            <button type="submit" class="submit-btn">Submit</button>
        </form>
    </main>


    <script src="script.js"></script>
</body>
</html>
```

3.2.2 CSS

3.2.2.1 *style.css*

```
body {
            margin: 0;
            padding: 0;
            font-family: 'Poppins', sans-serif;
            background-color: #FFD464;
            height: auto;
        }
        .logo {
            width: 60px;
            height: 60px;
        }

        .ham {
            display: none;
            cursor: pointer;
```

```
}

.links {
    text-decoration: none;
    color: #000;
    font-family: 'Poppins', sans-serif;
    font-weight: 600;
    font-size: 20px;
    float: right;
    padding: 10px;
}

header, main, footer {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}

header {
    margin-bottom: -20px;
}

h1 {
    text-align: center;
    font-family: 'Poppins', sans-serif;
    font-weight: 800;
    font-size: 40px;
    border: #fff;
}

h2 {
    text-align: center;
    font-family: 'Poppins', sans-serif;
    font-weight: 600;
    font-size: 20px;
    color: #000;
}

.store {
    text-align: center;
    font-size: 20px;
```

```
    margin-top: 40px;
}

.app {
    width: 150px;
}

.mockup {
    display: none;
    width: 50%;
    height: auto;
}

.mockupr {
    width: 50%;
    height: auto;
}

.logo2 {
    display: none;
}

.dropdown {
    display: none;
}

.tc {
    margin-left: 20px;
}

.about {
    margin-top: 210px;
    float: right;
    width: 1000px;
    margin-right: 60px;
    font-size: 20px;
}

.hand {
    width: 600px;
    height: auto;
    float: left;
```


PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education | Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    margin-top: 20px;
    margin-left: 160px;
}

.contact-form {
    display: flex;
    flex-direction: column;
    align-items: center;
    background: #fff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 90%;
    max-width: 600px;
    margin: 20px 0;
}

.form-input {
    font-family: 'Poppins', sans-serif;
    font-size: 16px;
    padding: 10px;
    margin: 10px 0;
    width: 100%;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
}

.form-input textarea {
    resize: none;
    height: 150px;
}

.submit-btn {
    font-family: 'Poppins', sans-serif;
    font-size: 18px;
    padding: 12px 20px;
    color: white;
    background-color: #000;
    border: none;
    border-radius: 5px;
    cursor: pointer;
```

```
    transition: background-color 0.3s ease;
}

.submit-btn:hover {
    background-color: #333;
}
@media (max-width: 768px) {
    body {
        background-color: #FFD464;
    }
    .mockup {
        display: block;
        width: 100%;
        height: 100%;
    }

    .mockupr {
        display: none;
    }

    .logo2 {
        display: block;
        width: 80px;
        height: 80px;
        margin-top: -150px
    }

    .ham {
        display: block;
        width: 40px;
        margin-top: 8px;
        height: 40px;
        margin-right: 7px;
        float: right;
    }

    .links {
        display: none;
    }

    .dropdown {
        display: none;
```

```
            position: absolute;
            top: 60px;
            right: 10px;
            background-color: #FFD464;
            box-shadow: 0px 8px 16px rgba(0,0,0,0.2);
            z-index: 1;
        }

        .dropdown a {
            color: black;
            padding: 12px 16px;
            text-decoration: none;
            display: block;
        }

        .dropdown a:hover {
            background-color: #ddd;
        }

        .about {
            margin-top: 100px;
            margin-left: 20px;
            float: right;
            width: 100%;
            margin-right: 0px;
            font-size: 15px;
        }

        .hand {
            width: 400px;
            height: auto;
            float: left;
            margin-top: 20px;
            margin-left: 0px;
        }
    }
```

3.2.3 JavaScript

3.2.3.1 *script.js*

```
function toggleMenu() {
    var menu = document.getElementById('dropdownMenu');
```

```
    if (menu.style.display === 'block') {
        menu.style.display = 'none';
    } else {
        menu.style.display = 'block';
    }
}
```

Conclusão

Foi na Escola Profissional de Braga que cimentei a minha paixão pela tecnologia e programação, e é com grande entusiasmo que confirmo que foi graças ao meu percurso na EPB me percebi que esta é a área profissional que desejo seguir.

Agradeço profundamente à EPB e ao corpo docente que me acompanhou nos últimos três anos por me terem proporcionado um ambiente de aprendizagem tão estimulante e por terem desempenhado um papel crucial na minha escolha profissional.

Quanto à prova de aptidão profissional em si, foi sem dúvida o projeto mais significativo da minha vida até agora.

Todo o trabalho desenvolvido na PAP permitiu-me fortalecer a minha determinação e habilidades, preparando-me para enfrentar desafios futuros com confiança e resiliência, com a noção que tudo está ao nosso alcance desde que estejamos dispostos a realizar o trabalho necessário com perseverança e consistência.

No demais, a motivação que fui ganhando, fez-me mais empenhado na constante melhoraria das minhas notas, com o objetivo de melhorar a minha média para garantir o meu ingresso no ensino superior e compreendi verdadeiramente que, sem esforço e dedicação, não se alcança nada de significativo na vida.

Acrescento que todo o tempo investido neste projeto foi extremamente compensador, porque me fez crescer, tanto a nível pessoal como enquanto programador e que estou ansioso por continuar a

desenvolvê-lo e a melhorar tudo o que for possível nela, transformando-o na minha primeira obra-prima.

Por fim, embora este ano tenha sido especialmente desafiador, revelou-se o mais gratificante da minha vida e fez-me perceber que quero continuar a aprofundar os meus conhecimentos em áreas que tanto me fascinam, como a tecnologia e a programação, no ensino superior, frequentando uma licenciatura em Engenharia Informática.

Hoje tenho a certeza, que devido ao meu percurso nos últimos 3 anos, me quero tornar Engenheiro Informático.

Termino agradecendo à minha família e amigos e mais uma vez aos meus professores por todos os ensinamentos, que enriqueceram o meu conhecimento técnico e as minhas *soft skills*.

Bibliografia

Webgrafia:

Componentes React Native - https://callstack.github.io/react-native-paper/

Componentes Node.js – https://www.w3schools.com/nodejs/default.asp

Integração da inteligência artificial - https://medium.com/@vibinreji123/integrating-chatgpt-3-5-into-your-react-native-application-a5a6f691b061

Ícones - https://www.flaticon.com/

Transformar um logótipo em vetores - https://www.adobe.com/products/photoshop/vectorize-image.html

Ligar *front-end* ao *back-end*- https://groovetechnology.com/blog/how-to-connect-react-native-with-node-js-a-simple-guide/

Comandos PostgreSQL - https://www.w3schools.com/postgresql/index.php

Alojamento - https://vercel.com/docs/deployments/overview

Ambiente de desenvolvimento - https://expo.dev/

API notícias - https://newsdata.io/

API moedas, *exchanges*, taxas - https://coinmarketcap.com/

API OpenAI - https://openai.com/index/openai-api/

Comparação de preços da API da OpenAI - https://openai.com/pricing

Virtualizador de *software* - https://www.docker.com/

PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Anexos

Anexo 1 –

Rodrigo Lopes Ferreira

Nationality: Portuguese **Date of birth:** 26/11/2006 **Gender:** Male

Phone number: (+351) 914537371

Email address: rodrigolopesferreirawork@gmail.com

WhatsApp Messenger: 914537371

Instagram: https://www.instagram.com/rodrigo.lf6/ **Website:** ftech.pt

Home: Braga (Portugal)

ABOUT ME

I am an extroverted and well humorated person. I like to work in groups. I have a friendly personality, and I am always available to help other people.

EDUCATION AND TRAINING

12th Grade Student - Technical Course in Management and Programming of Computer Systems
Escola Profissional de Braga [16/09/2021 – Current]

Address: R. Augusto Veloso 140, 4705-082 Braga (Portugal)
Website: epb.pt

• DST Innovation Challenge;
• Video promoting human rights;

LANGUAGE SKILLS

Mother tongue(s): **Portuguese**

Other language(s):

English	**Spanish**
LISTENING C2 **READING** C2 **WRITING** C2	**LISTENING** B1 **READING** A1 **WRITING** A1
SPOKEN PRODUCTION C2 **SPOKEN INTERACTION** C2	**SPOKEN PRODUCTION** A2 **SPOKEN INTERACTION** A1

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

DIGITAL SKILLS

CMD / Microsoft Teams / Microsoft Word / Microsoft Excel / Microsoft PowerPoint / Google Maps / Google Docs / Google Earth / Google Drive / Google Photos / Gmail / Outlook / Instagram / Twitter / FL Studio / Canva / WhatsApp / LinkedIn / Youtube / User Knowledge / Avast AntiVirus / Internet user / Google Meet / Google Translate / GitHub / Vercel

PROGRAMMING SKILLS

Languages:

- JavaScript
- HTML and CSS
- React Native (learning)
- Basic PHP
- Basic MySQL
- Basic PostgreSQL
- Basic NodeJs

1 / 3

PROJECTS

Portfolio theme

My own portfolio can be used as theme for your personal portfolio.

Link: lftech.pt

Cryptrail

[10/05/2024]

My final course project, Cryptrail, a application for IOS wich allows the user to follow the digital cryptocurrency market, know the coins prices, the best sites to buy or exchange, updated graphics, coin exchange tax, and ultimate crypto news.

Link: cryptrail.pt

HOBBIES AND INTERESTS

Guitar Player | Watch/Play Football | Play Videogames | Horror Movies | Listen To Music

COMMUNICATION AND INTERPERSONAL SKILLS

1. Respectful;
2. Assiduous;
3. Great Communicator;
4. Nice leadership;
5. Empathy;
6. Criativity;
7. Teamwork;
8. Proactivity;
9. Work Ethic;
10. Critical thoughts
11. Positive Attitude;

VOLUNTEERING

EMACI2022 - Bragaades

[Altice Arena Fórum, Braga, Portugal , 19/02/2022 – 27/02/2022]

I served as a translator;

Help people to reach their destiny;

Work at the athlete courses, organizing competition equipment.

Senior Formator

[04/11/2022 – Current]

•Senior formator in MS Office, E-mail, Web browsing, Cybersecurity, Computer History;

CERTIFICATES

Microsoft Office Specialist - Associate

[29/03/2023 – Current]

1. Word specialist, certified in Word 2019 Associate;
2. Excel specialist, certified in Excel 2019 Associate;
3. PowerPoint specialist, certified in Powerpoint 2019 Associate;