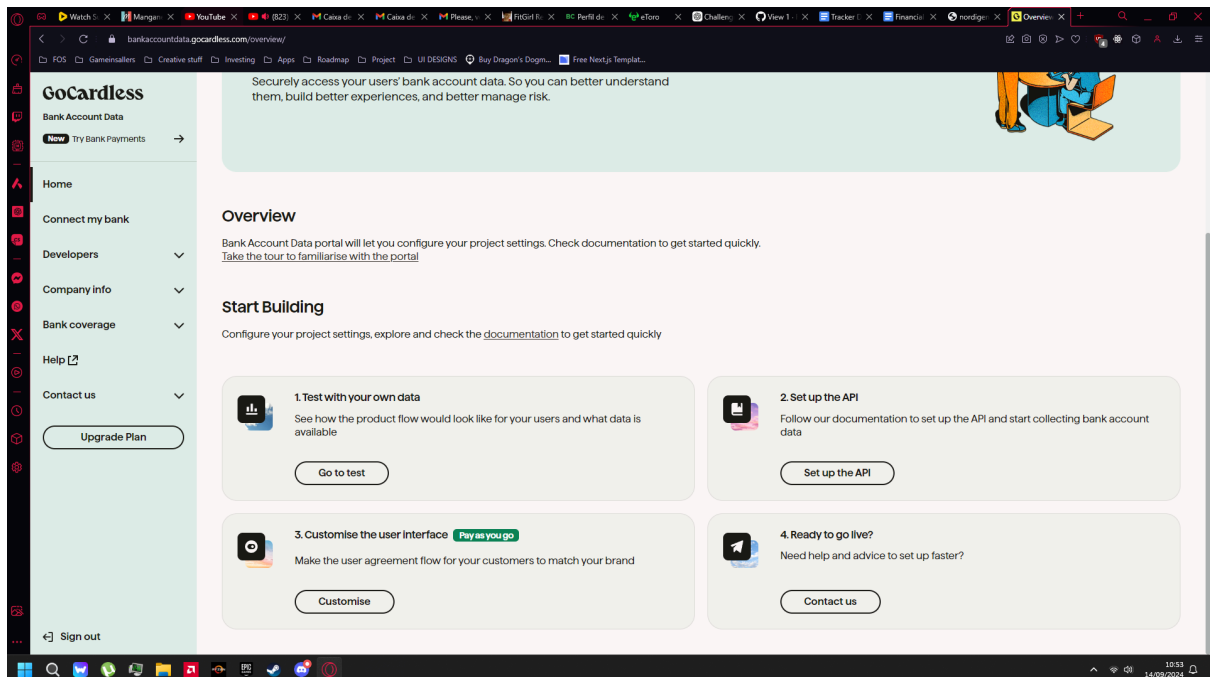


Financial Tracker - Documento de Pesquisa

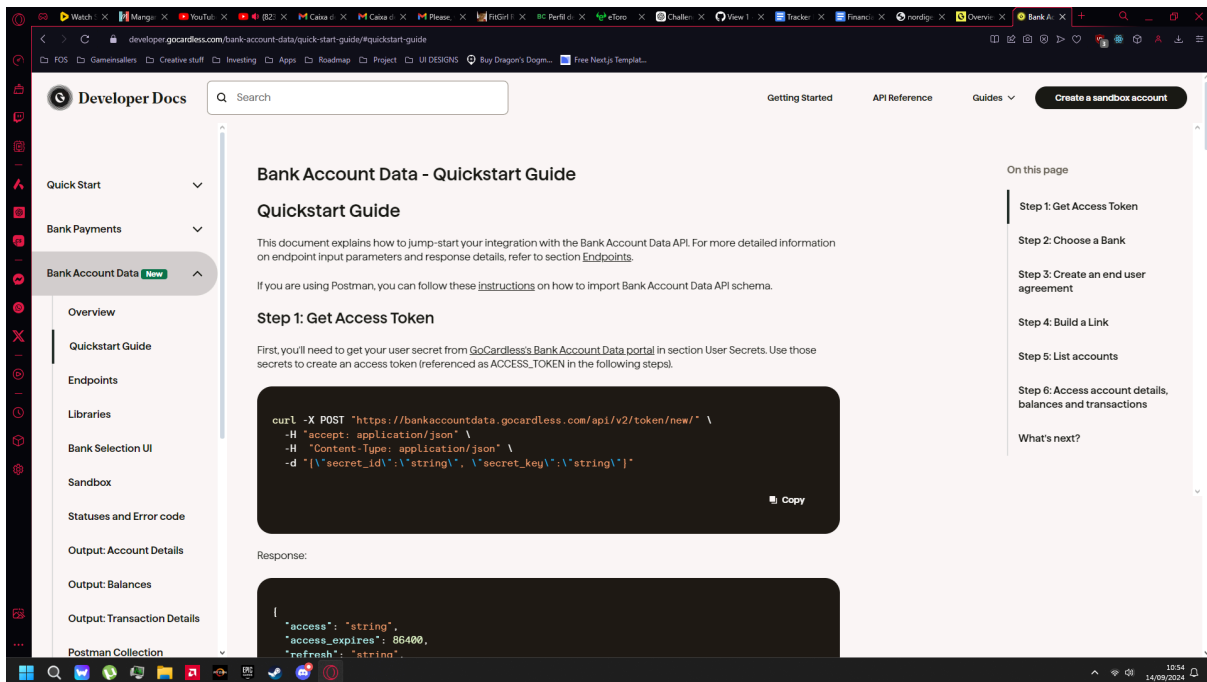
1. Pesquisa de APIs

- Esta tarefa envolve a exploração detalhada e integração das principais APIs utilizadas no projeto: [Nordigen](#) para dados bancários, **Alpha Vantage** para informações sobre ações e o mercado financeiro, e **CoinMarketCap** para dados de criptomoedas. A pesquisa incluirá uma avaliação completa da documentação de cada API, a fim de garantir a segurança dos dados e o desempenho ideal do sistema.
- **Responsável:** Dmytro
- **Estado:** In Progress
- **Última Atualização:** 14 de setembro de 2024

Nordigen “[Website da API](#)”



Quick Start



2. Design do Frontend

- Focado na pesquisa de boas práticas de **UX/UI** e na aplicação de frameworks modernos, como **Tailwind CSS** para estilização e **Framer Motion** para animações. O objetivo é garantir uma interface de utilizador fluida, responsiva e visualmente apelativa para facilitar a navegação entre secções financeiras e relatórios detalhados.
- **Responsável:** Dmytro
- **Estado:** In Progress
- **Última Atualização:** 14 de setembro de 2024

Figma “Template”

Tema e Estilo:

- **Dark Mode:** Fundo escuro com verde fluorescente para contraste.
- **Fonte:** Moderna e sem serifa (ex.: **Roboto**, **Inter**).
- **Tailwind:** Utiliza para cores (**bg-black**, **text-green-500**) e responsividade (**sm:**, **md:**, **lg:**).

Componentes:

- **Navbar:** Fixa no topo, fácil navegação.
- **Botões:** Verde com efeito hover (**hover:bg-green-500**).
- **Cards:** Usar **shadow-lg**, **p-4**, **rounded-lg** para informações.

Animações:

- **Framer Motion:** Animação nos cards ao rolar ou ao fazer hover.
- **Hover Effects:** Para gráficos e botões (animar **scale**).

Gráficos:

- **Chart.js** ou **Recharts** para visualizações de dados interativas.
- Gráficos de barra e pizza nas secções de finanças.

Formulários:

- **Login/Signup:** Design simples, usar **React Hook Form** para validação.
- **Validação:** Verificação em tempo real no input.

Responsividade:

- **Tailwind** classes para breakpoints (**sm**, **md**, **lg**).
- Ajustar grids e texto conforme o tamanho da tela.

3. Arquitetura do Backend

- A pesquisa aqui será sobre a organização e estruturação do backend, que será construído em **Node.js/Express**. Serão definidos os diretórios, rotas, e a lógica de comunicação com as APIs externas. Além disso, será necessário definir a interação entre o backend e os serviços de **machine learning** em **Python**.
- **Responsável:** Dmytro
- **Estado:** In Progress
- **Última Atualização:** 14 de setembro de 2024

BankAccounts Storage size: 20.48 kB Documents: 1 Avg. document size: 205.00 B Indexes: 1 Total index size: 20.48 kB	CryptoAccounts Storage size: 20.48 kB Documents: 1 Avg. document size: 186.00 B Indexes: 1 Total index size: 20.48 kB	CryptoTransactions Storage size: 20.48 kB Documents: 1 Avg. document size: 168.00 B Indexes: 1 Total index size: 20.48 kB	InvestmentPlatforms Storage size: 20.48 kB Documents: 1 Avg. document size: 136.00 B Indexes: 1 Total index size: 20.48 kB
Investments Storage size: 20.48 kB Documents: 1 Avg. document size: 140.00 B Indexes: 1 Total index size: 20.48 kB	Transactions Storage size: 20.48 kB Documents: 1 Avg. document size: 146.00 B Indexes: 1 Total index size: 20.48 kB	Users Storage size: 20.48 kB Documents: 1 Avg. document size: 385.00 B Indexes: 1 Total index size: 20.48 kB	

```
  _id: ObjectId('66e56d21102f86f8eec73bfb')
  name: "John Doe"
  email: "johndoe@example.com"
  passwordHash: "hashed_password_here"
  userId: "user123"
  createdAt: 2024-09-14T11:01:53.924+00:00
  ▶ bankAccounts: Array (1)
  ▶ cryptoAccounts: Array (1)
  ▶ investmentPlatforms: Array (1)
```

```
▼ {
  ▶ "_id": {},
    "transactionId": "tx001",
  ▶ "accountId": {},
  ▶ "transactionDate": {},
    "amount": 100,
    "type": "deposit",
    "category": "salary"
}
```

```
▼ {
  ▶ "_id": {},
    "investmentId": "inv001",
  ▶ "platformId": {},
    "assetType": "stock",
    "quantity": 10,
    "currentValue": 1000,
    "profitLoss": 200
}
```

```
▼ {
  ▶ "_id": {},
    "platformId": "platform001",
  ▶ "userId": {},
    "platformName": "Robinhood",
    "accountBalance": 10000,
    "investments": []
}
```

```
▼ {
  ▶ "_id": {},
    "transactionId": "crypto_tx001",
    "walletAddress": "1A2b3C4d5E",
  ▶ "transactionDate": {},
    "amount": 0.1,
    "currency": "BTC",
    "transactionType": "buy"
}
```

```
▼ {
  ▶ "_id": {},
    "transactionId": "crypto_tx001",
    "walletAddress": "1A2b3C4d5E",
  ▶ "transactionDate": {},
    "amount": 0.1,
    "currency": "BTC",
    "transactionType": "buy"
}
```

```
▼ {
  ▶ "_id": {},
    "accountId": "acc001",
  ▶ "userId": {},
    "bankName": "Bank of America",
    "balance": 5000,
    "accountType": "savings",
  ▶ "createdAt": {},
  ▶ "transactions": []
}
```

4. Planejamento da Base de Dados

- Esta secção envolverá a pesquisa e definição da estrutura das coleções no **MongoDB**. Será necessário definir o modelo de dados, incluindo as coleções de **utilizadores**, **transações**, e **previsões financeiras**. A pesquisa incluirá estratégias para garantir a escalabilidade e a segurança dos dados sensíveis.
- **Responsável:** Dmytro
- **Estado:** Todo

5. Ferramentas de Desenvolvimento

- A tarefa será pesquisar e escolher as melhores ferramentas e plataformas para o desenvolvimento contínuo do projeto. Isso inclui explorar a hospedagem do frontend em **Vercel**, o deploy do backend em serviços como **Heroku**, e a implementação de automação e CI/CD através de **GitHub Actions**.
- **Responsável:** Dmytro
- **Estado:** Todo