הגשת פרויקט

קבוצה מס': 7

נושא: **Personal Crypto Portfolio Tracker**

פרטי חברי הצוות:

|  |  |
| --- | --- |
| **שם: יוסרא פחאמנה** | **ת.ז: 204138879** |
| **שם: אואיס זועבי** | **ת.ז:** 209249655 |
| **שם:** נועם פורמן | **ת.ז:** 312333925 |
| **שם:** עומרי הייט | **ת.ז:** 312333925 |
| **שם:** עדן עגאינה | **ת.ז:** 322394552 |

**Website:** <https://cryptonize.vercel.app/>

Note: CoinGecko API that we used was a free trial, therefor we get limitation on the number of requests per minute. Reaching the maximum number of requests lead to getting a warning/error in the console.

**GitHub:** <https://github.com/owisezoubi/Cryptonize.git>

**תקציר של הפרויקט:**

Cryptonize הוא אתר לניהול ומעקב אחר פורטפוליו קריפטו אישי, המעניק למשתמשים פרטיים שקיפות ושליטה מלאה על נכסיהם הדיגיטליים. המערכת מאפשרת חיבור ארנק קריפטו בשתי דרכים: חיבור ישיר באמצעות WalletConnect API או באמצעות חיפוש לפי כתובת ארנק. הוא תומך בשתי רשתות מרכזיות: Ethereum ו Binance Smart Chain – ומספק מידע לגביהם, כולל ניתוחי ביצועים וסכום כולל בדולרים. בנוסף, ניתן לצפות בהיסטוריית העסקאות ולבחון את העמלות ששולמו עבורן. המערכת כוללת גם תצוגה של עשרת המטבעות עם הביצועים הטובים ביותר בארנק, עם אפשרות לסינון נתונים לפי יום, שבוע או חודש.

למשתמשים שאינם מחוברים, האתר מאפשר ניווט בין עמודי מידע, כולל עמוד "אודות", המסביר את מטרות המערכת. הם יכולים להתרשם מממשק המשתמש, אך ללא גישה לנתונים האישיים של הפורטפוליו. האתר עוצב כך שיספק חוויית משתמש נוחה ורספונסיבית, עם מידע עדכני לניהול חכם של נכסי הקריפטו.

**טכנולוגיות מרכזיות שנעשה בהם שימוש:**

1. Frontend

* :React.js + Viteלפיתוח רכיבי הממשק ולוגיקת האינטראקציה.
* JSX לבניית רכיבים בצורה דקלרטיבית בתוך .React
* React Router: לניהול ניווט בין עמודים.
* : Wagmi & RainbowKit מימוש בממשק לניהול חיבור ארנקי קריפטו.
* ECharts: להצגת גרפים ודוחות ביצועים.

1. Backend & API Integration (באמצעות APIs חיצוניים)

* Ethplorer API: לשליפת מידע על ארנקי קריפטו, טוקנים ועסקאות באמצעות wallet address.
* BNB Smart Chain & Etherscan API: לחישוב עמלות Gas של הארנק.
* :CoinGecko APIלקבלת נתוני מטבעות הקריפטו כמו מחירים עדכניים וסמלי המטבעות.

1. Styling

* Tailwind CSS: לעיצוב הממשק באופן גמיש ורספונסיבי.
* Dark/Light Mode: תמיכה במצב כהה ובהיר.

1. Version Control & Deployment

* Git & GitHub: לניהול גרסאות ולשיתוף קוד.
* Vercel: לפריסה והפעלת האתר בסביבה חי.
* **משימות חברי הצוות**

|  |  |  |
| --- | --- | --- |
| **Status** | **Tasks** | **Name** |
| Completed | - System Engineer. - Implemented Table of Currencies on the Home Screen (retrieval & display)​. - Developed the "About Us" page​ & styling. - Integrated Token Distribution Chart.​  - Implemented transaction history table​.  - Logo, Navigation bar, Footer Styling. | Yosra |
| Completed | - Implemented wallet connection using Web3 and RainbowKit​. - Handled API connections for fetching wallet assets & prices.​​ - Set up Git repository and deployment to Vercel. - Integrated network selection & switching functionality​.  - Managed state handling for wallet data.​ | Owais |
| Completed | - Designed system architecture and use case diagrams​. - Developed routing and navigation with React Router​. - Structured component-based UI layout​. - Managed data processing for best-performing tokens​.  - Created and styled video section on the homepage.  - Logo styling. | Omri |
| Completed | - Designed responsive UI and layouts using Tailwind CSS​. - Implemented theme toggling and UI components​. - Styled & structured assets and transactions tables​​. - Designed interactive filters and network selection dropdown​.  - Integrated search functionality for wallet addresses​. | Noam |
| Completed | - Documented functional and non-functional requirements. - Implemented wallet fees. - Handled table responsiveness across devices​.  - Implemented and styled Dark/Light mode. | Adan |

* חלוקת עבודה ושיתוף פעולה בצוות

במסגרת הפרויקט, החלוקה בין חברי הצוות בוצעה בהתאם לחוזקות האישיות של כל אחד, כך שכל חבר צוות לקח אחריות על תחום שבו יש לו ידע וניסיון. התהליך התחיל בדיון קבוצתי שבו כל אחד שיתף את התחומים שבהם הוא מרגיש בנוח לתרום, ובכך נבנתה חלוקה מאוזנת שהבטיחה שכל משימה תקבל מענה איכותי.

הדגש היה על שיתוף פעולה מתמשך, ולא על עבודה מבודדת של כל אחד בנפרד. חברי הצוות עבדו בשיתוף פעולה לאורך כל הדרך, והיו ביניהם אינטראקציות קבועות לצורך פתרון בעיות, שיפור הקוד, והתקדמות כללית בפרויקט.

* אמצעי התקשורת ושיתוף החומרים

כדי להבטיח תקשורת רציפה ומעקב אחר ההתקדמות, הצוות השתמש במספר פלטפורמות:

1. WhatsApp - לתקשורת מהירה, עדכונים שוטפים ותיאום בין חברי הצוות.
2. Discord - פגישות קוליות/וידאו תקופתיות לשם תיאום, דיון על אתגרים, וחלוקת משימות בהתאם להתקדמות.
3. OneDrive- שימש כמקום מרכזי לאחסון מסמכים, עיצובי מערכת, ודיאגרמות, כך שלכולם הייתה גישה נוחה ומסודרת.
4. GitHub - שימש לשמירת הגרסה האחרונה של הקוד לצורך העלאה ל- Vercel.

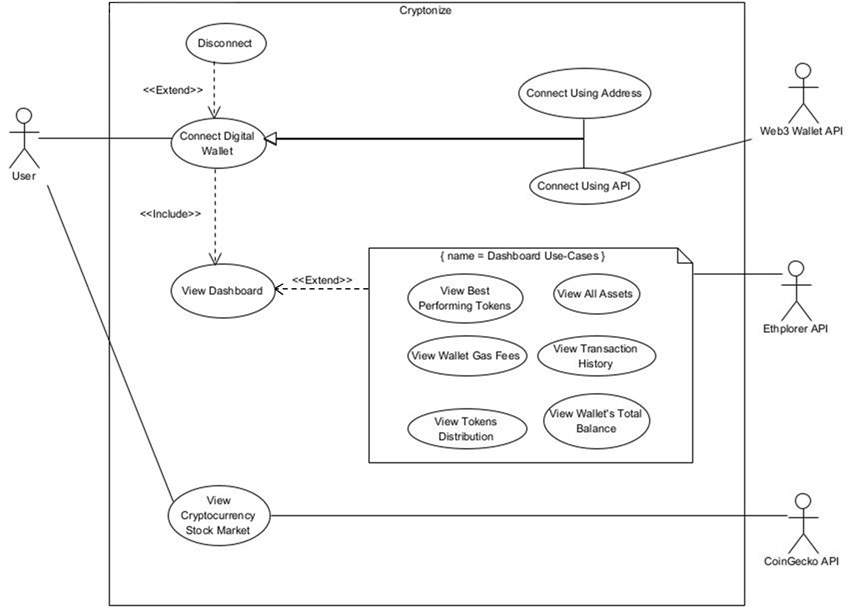
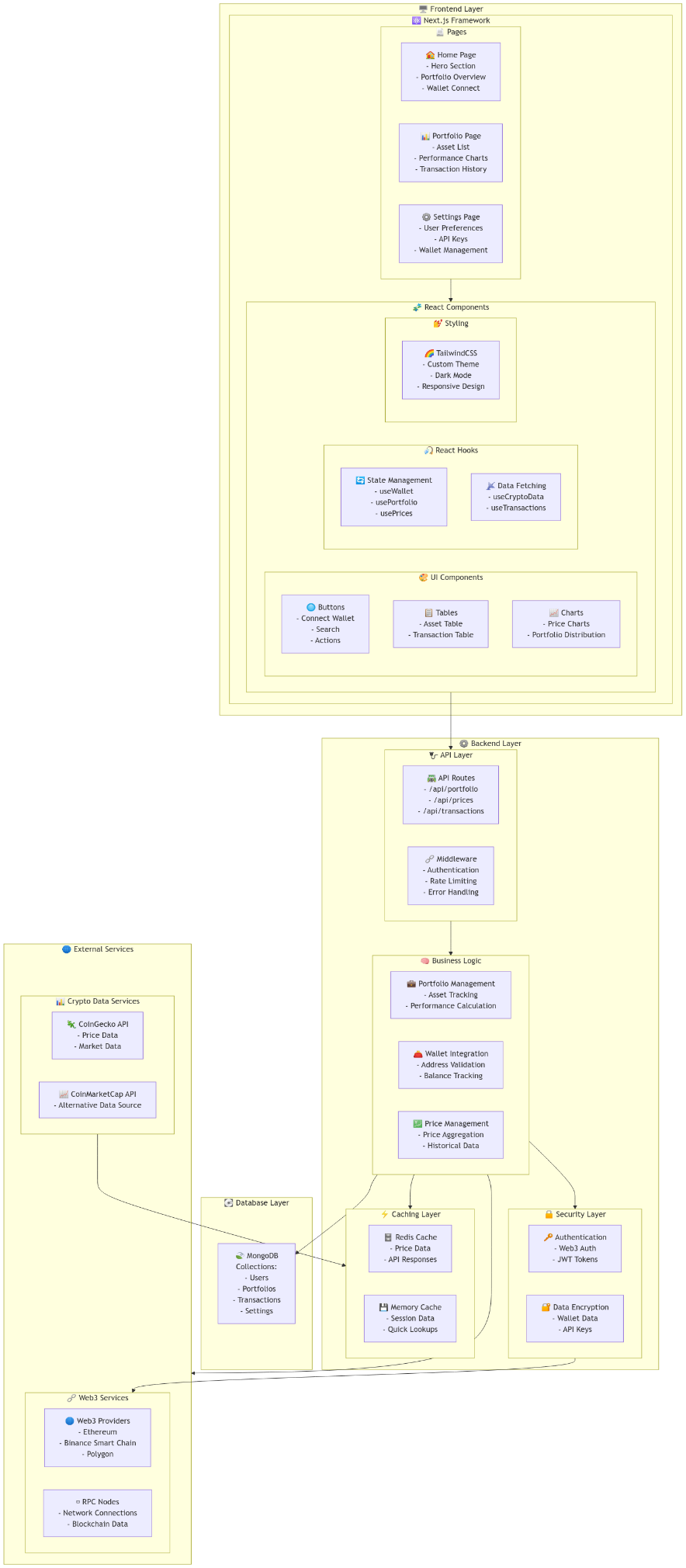
* **דרישות המערכת**

|  |  |
| --- | --- |
| **Req. Number** | **Functional Requirement** |
| 1 | The system allows users to create and customize their cryptocurrency portfolios by adding, editing, and removing holdings. |
| 2 | The system allows users to view real-time price updates, fetched from a reliable external API, within 1-2 seconds. |
| 3 | The system allows users to view historical price trends over customizable timeframes. |
| 4 | The system allows users to identify top-performing assets in their portfolio. |
| 5 | The system allows users to view daily, weekly, and monthly profit/loss summaries. |
| 6 | The system allows users to manage assets across various block chain networks. |

|  |  |  |
| --- | --- | --- |
| **Req. Number** | **Non-Functional Requirement** | **Classification\*** |
| 1 | Ensure the system updates real-time prices within 1-2 seconds. | Performance |
| 2 | Use industry-standard encryption protocols for data in transit and at rest. | Security |
| 3 | Provide multilingual support for a global audience. | Accessibility |
| 4 | Design for scalability to handle many concurrent users. | Scalability |
| 5 | Maintain 99.9% uptime for critical features such as portfolio access  and real-time updates. | Reliability |

\* לפי <https://en.wikipedia.org/wiki/Non-functional_requirement>

* **מבנה ואינטראקציה**

1. דיאגרמת use case
2. ארכיטקטורה של האתר

* **מבנה סופי של האתר**

1. מבנה והסבר

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

חלק מ- src

**public/ – Public Assets**

This folder contains static media files that are publicly accessible by the application without needing imports, needed to be created to pass the Vercel deployment compiler.

* Includes background videos and logos.

**src/ – Source Code**

This is the main folder containing the entire React application, including components, styles, utilities, and configurations.

1. assets/ – Static Resources

Stores static files such as images, logos, and icons used in the project.

1. components/ – UI Components

Contains React components responsible for building the user interface.  
Each file in this folder represents a specific UI element, such as tables, buttons, charts, and navigation elements.

1. config/ – Application Configuration

Stores configuration files related to state management and data flow.

1. icon/ – Icons & Team Avatars

Contains icons and images used throughout the application, including team member avatars, blockchain logos, and mode toggle icons.

1. styles/ – Component Styling

Includes Tailwind CSS and additional JavaScript styling files that define the look and feel of individual components.

1. utils/ – Utility Functions

Holds helper functions that handle data processing, API requests, and formatting.

1. Main Files in Root (src/)
   * App.jsx – The main React component structuring the application.
   * main.jsx – The entry point for rendering the app.
   * index.css – The global CSS file for styling.
2. פריטי המידע

In our project, data is retrieved from an external API and displayed to users in a structured and accessible way. Below is a breakdown of the key data items used in the system.

* Before Wallet Connection

**General Cryptocurrency Market Data**

Before a user connects their wallet, the system displays real-time market data:

* Rank – The ranking of the cryptocurrency in the market.
* Token Name & Symbol – The name and trading symbol of the cryptocurrency (e.g., Bitcoin (BTC), Ethereum (ETH)).
* Token Price – The current price of the cryptocurrency in USD.
* Price Change (1h, 24h, 7d) – Percentage price fluctuations over different timeframes.
* After Wallet Connection

**Wallet Data**

Once the user connects their wallet, additional personalized data is displayed:

* Wallet Address – The address of the connected wallet.
* Wallet Balance – The total value of all assets in the wallet (in USD).
* Gas Fees – Displays transaction fees for Ethereum and Binance Smart Chain networks.
* Token Distribution – A graphical breakdown of the proportion of different assets in the wallet, showing the percentage allocation of each token.

**Network & Filtering Data**

General cryptocurrency market data used for filtering and selection:

* Available Networks – Supported blockchain networks (Ethereum, Binance Smart Chain).
* Current Market Prices – Live price updates for various cryptocurrencies.
* Price Change Percentage – Displays price fluctuation over different periods.
* Total Market Value – The total value of all cryptocurrencies held in the wallet.

**Crypto Assets & Tokens (Personal Holdings)**

Details of the specific tokens the user holds in their wallet:

* Token Name & Symbol – The official name and symbol of the cryptocurrency.
* Amount – The quantity of each token in the wallet.
* 24h Change (%) – The percentage change in the token's price over the last 24 hours.
* Price (USD) – The current market price of the token in USD.
* Total (USD) – The total value of the user's holdings for this token (Amount × Price).

**Transaction History**

A record of past transactions associated with the connected wallet:

* Transaction Hash – A unique identifier for each transaction.
* Date & Time – When the transaction was processed.
* From Address – The sender’s wallet address.
* To Address – The recipient’s wallet address.
* Amount – The quantity of cryptocurrency transferred.
* Type – Specifies whether the transaction is incoming (received) or outgoing (transfer).
* Token – The cryptocurrency used in the transaction (e.g., ETH, BNB).

**Best Performing Tokens**

Displays the top 10 performing tokens in the user's wallet based on recent price trends:

* Token Name & Symbol – The cryptocurrency’s name and symbol.
* Price (USD) – The current market price of the token in USD.
* Balance – The quantity of each token held in the wallet.
* Balance (USD) – The total value of the token holdings in USD (Balance × Price).
* 24H / 7D / 30D Performance – Percentage change in price over different timeframes. (24 hours, 7 days, and 30 days).
* **תיק למתכנת**

1. תיעוד פונקציות מרכזיות בקוד

**Main Application Structure**

* App.jsx

Function: App()

The main component that structures the application.

Manages the overall layout and renders the navigation bar, footer, and main sections.

**Core Functionality Components**

* WalletConnectComponent.jsx

Function: handleWalletConnect()

Connects the user’s crypto wallet via WalletConnect.

Stores the wallet address and network in the application's state.

Function: handleDisconnect()

Disconnects the connected wallet and clears session data.

* ChooseNetwork.jsx

Function: selectNetwork(network)

Allows users to switch between Ethereum and Binance Smart Chain (BSC) networks.

Updates the selected network state for subsequent API requests.

* CryptoTable.jsx

Function: fetchMarketData()

Fetches real-time cryptocurrency market data from CoinGecko API.

Updates the table with rank, token name, price, price change %, and market cap.

* AssetsTable.jsx

Function: fetchWalletAssets(walletAddress)

Retrieves all tokens in the user’s wallet from Ethplorer API.

Displays the token name, balance, USD value, and price changes.

* DistributionChart.jsx

Function: generateChartData(assetsList)

Processes wallet assets and calculates token distribution for visualization.

Converts token holdings into a percentage-based chart dataset.

* BestPerformingTable.jsx

Function: getTopPerformingAssets()

Filters and displays the top 10 best-performing assets in the user's wallet.

Uses 24H / 7D / 30D performance metrics to rank assets.

* TransactionTable.jsx

Function: fetchTransactions(walletAddress)

Retrieves transaction history from Ethplorer API.

Displays transaction hash, date, sender/receiver, token, type, and amount.

* WalletFees.jsx

Function: fetchGasFees(network)

Retrieves real-time gas fees for the selected blockchain network.

Supports Ethereum and Binance Smart Chain (BSC).

**Utility Functions (utils/ Directory)**

* walletUtils.js

Function: validateWalletAddress(address)

Checks if the wallet address entered by the user is valid.

* coinGecko.js

Function: fetchTokenPrice(tokenId)

Retrieves the latest price for a given cryptocurrency from CoinGecko API.

* transactionUtils.js

Function: formatTransactionData(transactionList)

Formats the transaction history before displaying it in TransactionTable.jsx.

* feesUtils.js

Function: calculateGasCost(transactionData)

Extracts and calculates gas fee details from a transaction.

**Configuration (config/ Directory)**

* CryptoDataContext.jsx

Function: CryptoDataProvider()

Uses React Context API to provide global state management for cryptocurrency data.

1. API שנעשה בהם שימוש וקטעי קוד מיוחדים

**APIs Used in the Project**

The project does not have a traditional backend. Instead, it retrieves data from multiple external APIs. Below is a breakdown of the APIs used, their purpose, and what data they. provide:

* CoinGecko API

Purpose: Retrieves real-time cryptocurrency market data.

Used In: CryptoTable.jsx, coinGecko.js

Data Provided:

* Token price (USD)
* 24h, 7d, 30d price changes (%)
* Market capitalization
* Cryptocurrency rank
* Token icons
* Ethplorer API

Purpose: Fetches wallet data, assets, and transaction history.

Used In: AssetsTable.jsx, TransactionTable.jsx, walletUtils.js

Data Provided:

* Wallet balance
* List of assets held in a wallet
* Token balances & USD value
* Transaction history (hash, sender, receiver, amount)
* BSCScan & Etherscan API

Purpose: Retrieves gas fees for transactions.

Used In: WalletFees.jsx, feesUtils.js

Data Provided:

* Gas fees for Ethereum and Binance Smart Chain
* Estimated transaction costs
* WalletConnect (RainbowKit + Wagmi)

Purpose: Enables secure wallet connection to the site.

Used In: WalletConnectComponent.jsx

Data Provided:

* Wallet address
* Network information
* Connection status

**Referenced Code & Sources**

In this project, most of the code was written by the development team, except for some referenced implementations used in the configuration file.

* CryptoDataContext.jsx (React Context API Implementation)

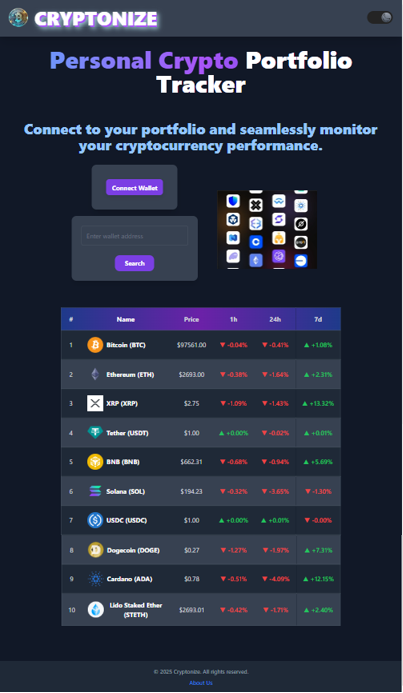
Purpose: Manages global cryptocurrency data across the application.

Why Referenced: The basic structure of the React Context API was adapted from official documentation.

Original Source: React Official Docs - Context API

<https://react.dev/reference/react/createContext>

* **תיק משתמש**

1. **מסך ראשי **

About Us – מציג מידע על חברי הצוות שהיו שותפים לפיתוח האתר

טבלה שמציגה מידע לגבי 10 סוגים שונים של crypto : מחיר עדכני בדולרים, שינוי (עלייה/ ירידה) במהלך שעה, יום ושבוע

בחירת theme לאתר

((dark / light

אפשרויות התחברות לארנק

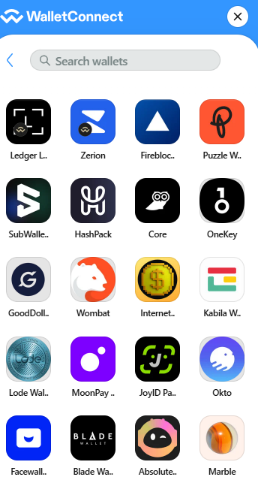
1. **אפשרויות התחברות לארנק**

* דרך אפליקציה

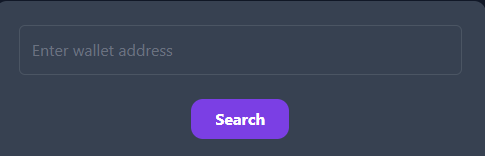
A purple rectangular button with white text

AI-generated content may be incorrect.Screens screenshot of a computer screen

AI-generated content may be incorrect.

הערה: בבחירת WalletConnect מקבלים מבחר רב של אפליקציות מעבר למוצג למעלה

.....

* דרך כתובת הארנק

דוגמאות שניתן לעשות בהם שימוש:

0xA69babEF1cA67A37Ffaf7a485DfFF3382056e78C

0x00000000219ab540356cBB839Cbe05303d7705Fa

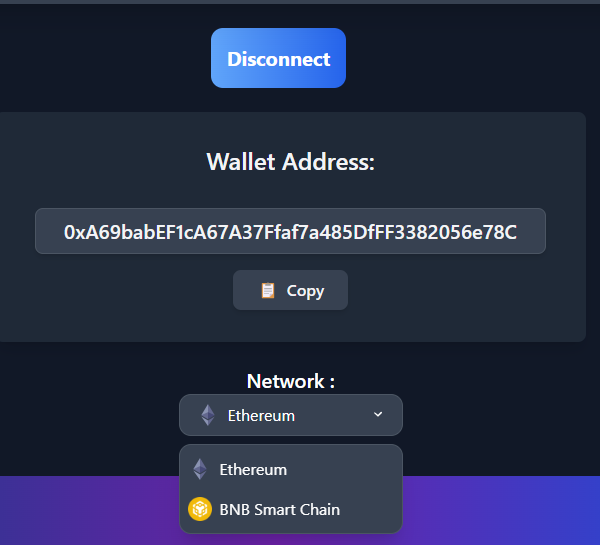
1. About Us

A screenshot of a website

AI-generated content may be incorrect.

כפתור חזרה לעמוד שהיה לפני

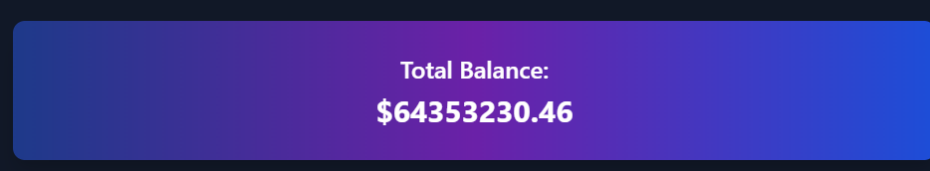
1. אחרי ההתחברות לארנק



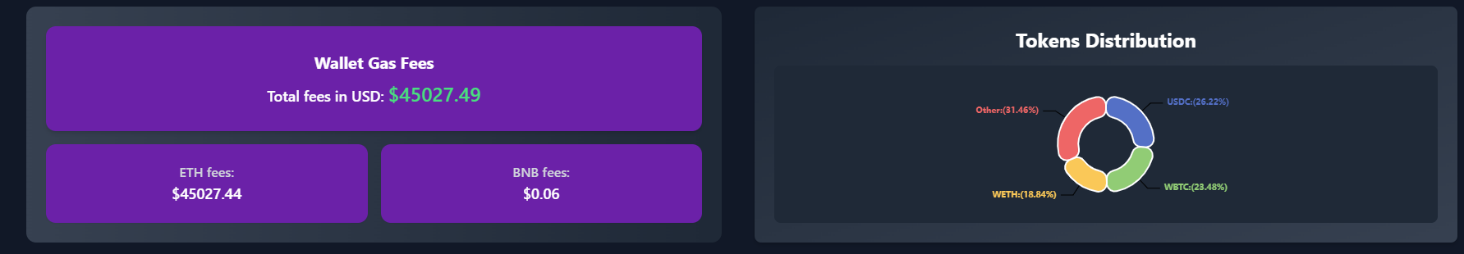
כפתור שמנתק את המשתמש מהארנק ומחזיר אותו למסך ראשי

מציג את כתובת הארנק שניתן גם להעתיק

בחירת מתוך שתי Networks להצגת מידע

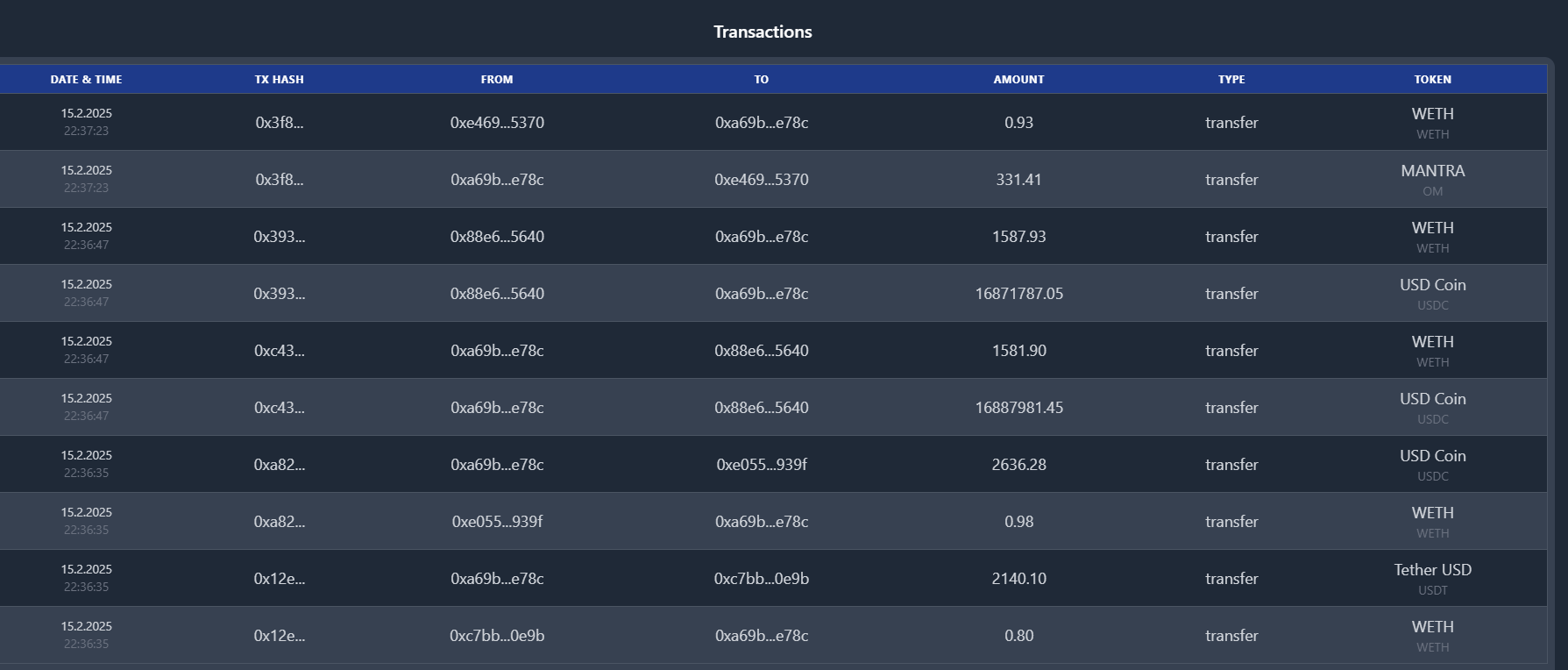


יתרה כוללת של אותו Network



מציג את חלוקת האחזקות בארנק לפי סוגי טוקנים והשווי שלהם

מציג עמלות שונות שקשורות לארנק

* טבלת Assets: הטבלה מציגה סקירה מסודרת של כל המטבעות שבארנק המחובר. מיוצג שם המטבע, הכמות שיש, שינוי האחוזים במחיר ב-24 השעות האחרונות, המחיר הנוכחי בדולרים, והסכום הכולל שהשקעת במטבע זה. מידע זה מאפשר לעקוב אחרי ביצועי ההשקעות ולבצע ניתוחים פיננסיים בצורה נוחה (בטבלה גודלה יש אפשרות מעבר על ידי slider).
* טבלת Transactions: הטבלה מציגה רשימה של transactions, עם פרטים כמו תאריך ושעה, קוד ההזמנה ,(Tx Hash) כתובת השולח והמקבל, הכמות של המטבעות המעורבים, סוג transaction (בטבלה גדולה המעבר הוא מדף לדף)
* טבלת Best Performing Tokens: הטבלה מציגה את המטבעות בעלות הביצועים הטובים ביותר בהתאם .כל שורה בטבלה מייצגת מטבע ומציגה את המדדים הבאים: דירוג, שם המטבע, המחיר הנוכחי בדולרים, אחוז השינוי במחיר לפי קריטריון הזמן הנבחר (24 שעות, שבוע, או חודש), הכמות של המטבעות שברשותך, והסכום הכולל.

