# **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**להגשה עד 14.4.24 בשעה 23:59**

**Group Members:**

Khalid Sweid: 324264530

Safa Kheir: 322829102

Tameer Amer: 322836388

Yoda Amer: 211853502

Nabi Meri: 316197490

GitHub Link: <https://github.com/Safa-Khier/b5>.

Web Link: <https://b5-theta.vercel.app>.

MTW Link: <https://www.morethanwallet.com/app/690>.

1. עליכם להמשיך את בניית האתר לפי האלמנטים המתקדמים שלמדתם
2. יש למנות מהנדס מערכת בכל צוות, אשר יהיה אחראי על הגדרת והקצאת המשימות בתרגיל זה.  
   נא לרשום את שם הסטודנט בתרגיל זה. על מהנדס המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

|  |  |  |
| --- | --- | --- |
| **Completed Tasks** | **Assigned Tasks** | **Member Name** |
| All Done | * Setup Firebase project * Implement Firebase connection * Develop backend logic for processing data * Adding API data to the DB | Khalid Sweid |
| All Done | * Develop interactive and responsive user interface * Implement features and functions based on UI/UX designs * Design and implement API endpoints * Integrate external APIs for fetching cryptocurrency news data * Implements compare cryptocurrencies | Safa Kheir |
| All Done | * Gather requirements for UseCase diagram * Update UseCase diagram | Yoda Amer |
| All Done | * Gather requirements for UseCase diagram * Update UseCase diagram * Translating the page to multiple languages | Tameer Amer |
| All Done | * Implement responsive and user-friendly components * Develop UI and Frontend * Update UseCase diagram | Nabi Meri |
| All Done | * Design and implement a real-time price widget on the homepage to display current prices of selected cryptocurrencies. * Ensure user-friendly displays * Develop Frontend | The whole team |

System Engineer: Khalid Sweid

As the system engineer of the team, I led the team through the second phase of the project, which primarily focused on connecting our website to the database we’ve chosen and it’s Firebase, illustrating the architecture of the website, creating a UseCase diagram, and advancing the development of the API, UI, Backend, Frontend of the website.

2. הציגו רשימת דרישות פונקציונליות ולא פונקציונליות (בנפרד, יש לסווג דרישות לא פונקציונליות לפי wikipedia NFR).

1. According to Wikipedia: functional requirements are usually in the form of "system shall do <requirement>", while non-functional requirements are in the form of "system shall be <requirement>".

FR:

* The system shall provide information about blocks.
* The system shall allow users to access details about transactions.
* The system shall enable users to explore details about specific cryptocurrency accounts.
* The system shall display account details.
* The system shall integrate real-time cryptocurrency news.
* The system shall display the latest news about blockchain and cryptocurrency.
* The system shall support mobile responsiveness.
* The system shall support theme modes (Light/Dark).

NFR:

Classification

Performance:

The system shall be optimized to have low-response times for retrieving blockchain data.

The system shall be capable of handling many requests.

Security:

The system shall be designed to have secure communication protocols (e.g., HTTPS).

Compatibility and Responsiveness:

The system shall be optimized for various devices, ensuring a responsive design that works well on both desktop and mobile platforms.

User Interface and Experience:

The system shall be designed to display account details in a clear and concise manner for users.

The system shall support real-time updates of account information to ensure users have access to the latest data.

The system shall be optimized for real-time data synchronization, ensuring that the integrated news feed is updated with the latest information.

The system shall provide users with the option to switch between Light and Dark modes to accommodate individual preferences.

The system shall apply instant theme mode changes without requiring a page refresh, ensuring a smooth and visually pleasing experience.

Usability:

The system shall provide intuitive navigation and user-friendly interfaces for ease of use.

The system shall offer clear instructions and guidance to users for performing tasks efficiently.

Accessibility:

The system shall adhere to web accessibility standards to ensure it is usable by individuals with disabilities.

The system shall provide options for users to adjust text size, contrast, and other accessibility features.

Reliability and Availability:

The system shall maintain high availability to ensure users can access it at any time.

The system shall be designed with redundancy and failover mechanisms to minimize downtime.

Scalability:

The system architecture shall be scalable to accommodate increased user demand and data volume over time.

The system shall be able to scale horozontally and vertically to hande users and resources effectively.

הציגו ארכיטקטורה מעודכנת של האתר (תרשים הכולל את האלמנטים המרכזיים).

A diagram of a computer server

Description automatically generated

.3הציגו דיאגרמת use case המתארת את השימוש באתר.

A diagram of a company

Description automatically generated

.4 יש להציג מבנה סופי של האתר שלכם:   
א. האתר ימומש ב -react/preact, וכן שימוש ב Tailwind - נא להציג דיאגרמה המתארת את התיקיות והקבצים השונים.

b5

├─ 📁build

│  ├─ 📁assets

│  │  ├─ 📄china-BJC3NXIF.png

│  │  ├─ 📄germany-RcDfzgS0.webp

│  │  ├─ 📄index-3Nz7iKXt.js

│  │  ├─ 📄index-bQ9Zgb\_f.css

│  │  ├─ 📄italy-k4diam-A.webp

│  │  ├─ 📄logo-DRWUd7yX.png

│  │  ├─ 📄logoB-BNS7NqN4.png

│  │  ├─ 📄russia-BFOhkMUR.png

│  │  └─ 📄usa-BeFRcQv-.png

│  ├─ 📁locales

│  │  ├─ 📁de

│  │  │  └─ 📄translation.json

│  │  ├─ 📁en

│  │  │  └─ 📄translation.json

│  │  ├─ 📁it

│  │  │  └─ 📄translation.json

│  │  ├─ 📁ru

│  │  │  └─ 📄translation.json

│  │  └─ 📁zh

│  │     └─ 📄translation.json

│  ├─ 📄favicon.ico

│  ├─ 📄index.html

│  ├─ 📄mockData.jsx

│  ├─ 📄publicFunctions.jsx

│  └─ 📄vite.svg

├─ 📁functions

│  ├─ 📄.gitignore

│  ├─ 📄index.js

│  ├─ 📄package-lock.json

│  └─ 📄package.json

├─ 📁public

│  ├─ 📁locales

│  │  ├─ 📁de

│  │  │  └─ 📄translation.json

│  │  ├─ 📁en

│  │  │  └─ 📄translation.json

│  │  ├─ 📁it

│  │  │  └─ 📄translation.json

│  │  ├─ 📁ru

│  │  │  └─ 📄translation.json

│  │  └─ 📁zh

│  │     └─ 📄translation.json

│  ├─ 📄favicon.ico

│  ├─ 📄mockData.jsx

│  ├─ 📄publicFunctions.jsx

│  └─ 📄vite.svg

├─ 📁src

│  ├─ 📁assets

│  │  ├─ 📁icons

│  │  │  ├─ 📄china.png

│  │  │  ├─ 📄germany.webp

│  │  │  ├─ 📄homeScreenLogo.png

│  │  │  ├─ 📄italy.webp

│  │  │  ├─ 📄logo.png

│  │  │  ├─ 📄logoB.png

│  │  │  ├─ 📄russia.png

│  │  │  ├─ 📄usa.png

│  │  │  └─ 📄userIcon.webp

│  │  └─ 📄react.svg

│  ├─ 📁atoms

│  │  ├─ 📄cryptoData.js

│  │  ├─ 📄cryptoNews.js

│  │  ├─ 📄userData.js

│  │  └─ 📄webSettings.js

│  ├─ 📁components

│  │  ├─ 📁alert

│  │  │  ├─ 📄alert.css

│  │  │  └─ 📄alert.jsx

│  │  ├─ 📁creditCard

│  │  │  ├─ 📄creditCardForm.css

│  │  │  └─ 📄creditCardForm.jsx

│  │  ├─ 📁navigationBar

│  │  │  ├─ 📄halving.jsx

│  │  │  ├─ 📄navigationBar.css

│  │  │  └─ 📄navigationBar.jsx

│  │  ├─ 📁screens

│  │  │  ├─ 📁home

│  │  │  │  ├─ 📁dashboard

│  │  │  │  │  ├─ 📁cashIn

│  │  │  │  │  │  ├─ 📄buy.currencey.screen.jsx

│  │  │  │  │  │  ├─ 📄cashIn.jsx

│  │  │  │  │  │  └─ 📄trade.currencey.screen.jsx

│  │  │  │  │  ├─ 📁withdraw

│  │  │  │  │  │  └─ 📄withdraw.jsx

│  │  │  │  │  ├─ 📄dashboard.css

│  │  │  │  │  └─ 📄dashboard.jsx

│  │  │  │  ├─ 📄coins.screen.jsx

│  │  │  │  ├─ 📄home.jsx

│  │  │  │  ├─ 📄news.screen.jsx

│  │  │  │  ├─ 📄settings.jsx

│  │  │  │  └─ 📄transactionsHistory.jsx

│  │  │  └─ 📁welcome

│  │  │     ├─ 📄login.jsx

│  │  │     ├─ 📄signup.jsx

│  │  │     └─ 📄welcome.jsx

│  │  ├─ 📁searchBar

│  │  │  ├─ 📄searchBar.css

│  │  │  └─ 📄searchBar.jsx

│  │  ├─ 📁table

│  │  │  ├─ 📁currenciesTable

│  │  │  │  ├─ 📄currenciesRow.jsx

│  │  │  │  ├─ 📄currenciesTable.jsx

│  │  │  │  └─ 📄dataSparkline.jsx

│  │  │  ├─ 📁holdingCoinsTable

│  │  │  │  ├─ 📄holdingCoinsRow.jsx

│  │  │  │  └─ 📄holdingCoinsTable.jsx

│  │  │  ├─ 📁newsTable

│  │  │  │  ├─ 📄newsTab.jsx

│  │  │  │  └─ 📄newsTable.jsx

│  │  │  ├─ 📁transactionsTable

│  │  │  │  ├─ 📁buyTable

│  │  │  │  │  ├─ 📄transactionsBuyRow.jsx

│  │  │  │  │  └─ 📄transactionsBuyTable.jsx

│  │  │  │  ├─ 📁tradeTable

│  │  │  │  │  ├─ 📄transactionsTradeRow.jsx

│  │  │  │  │  └─ 📄transactionsTradeTable.jsx

│  │  │  │  └─ 📁withdrawTable

│  │  │  │     ├─ 📄transactionsWithdrawRow.jsx

│  │  │  │     └─ 📄transactionsWithdrawTable.jsx

│  │  │  ├─ 📄loading.data.screen.jsx

│  │  │  └─ 📄paging.jsx

│  │  ├─ 📁userMenu

│  │  │  ├─ 📄userMenu.css

│  │  │  └─ 📄userMenu.jsx

│  │  ├─ 📄footer.jsx

│  │  ├─ 📄loading.screen.jsx

│  │  └─ 📄particlesBackground.jsx

│  ├─ 📄App.jsx

│  ├─ 📄AuthContext.js

│  ├─ 📄firebase.js

│  ├─ 📄i18n.js

│  ├─ 📄index.css

│  ├─ 📄main.jsx

│  └─ 📄protectedRoute.jsx

├─ 📄.eslintrc.cjs

├─ 📄.firebaserc

├─ 📄.gitignore

├─ 📄.prettierignore

├─ 📄.prettierrc

├─ 📄firebase.json

├─ 📄index.html

├─ 📄package-lock.json

├─ 📄package.json

├─ 📄postcss.config.cjs

├─ 📄README.md

├─ 📄tailwind.config.cjs

└─ 📄vite.config.js

ב. יש לפרט את פריטי המידע - יש להשתמש במידע אמיתי ורלוונטי לפרויקט שלכם (בשליפה ממסד נתונים חיצוני או מ - API). יש להראות דיאגרמת מבנה DB.

Firestore

├─ cryptocurrencies

│  ├─ ath: Number

│  ├─ ath\_change\_percentage: Number

│  ├─ ath\_date: String

│  ├─ atl: Number

│  ├─ atl\_change\_percentage: Number

│  ├─ atl\_date: String

│  ├─ circulating\_supply: Number

│  ├─ current\_price: Number

│  ├─ fully\_diluted\_valuation: Number

│  ├─ high\_24h: Number

│  ├─ id: String

│  ├─ image: String

│  ├─ last\_updated: Number

│  ├─ low\_24h: Number

│  ├─ market\_cap: Number

│  ├─ market\_cap\_change\_24h: Number

│  ├─ market\_cap\_change\_percentage\_24h: Number

│  ├─ market\_cap\_rank: Number

│  ├─ max\_supply: Number

│  ├─ name: String

│  ├─ price\_change\_24h: Number

│  ├─ price\_change\_percentage\_1h\_in\_currency: Number

│  ├─ price\_change\_percentage\_24h: Number

│  ├─ price\_change\_percentage\_24h\_in\_currency: Number

│  ├─ roi: null

│  ├─ sparkline\_in\_7d

│  │  └─ price: array[168]

│  ├─ symbol: String

│  ├─ total\_supply: Number

│  └─ total\_volume: Number

├─ cryptonews

│  ├─ body: String

│  ├─ categories: String

│  ├─ downvotes: String

│  ├─ guid: String

│  ├─ id: String

│  ├─ imageurl: String

│  ├─ lang: String

│  ├─ published\_on: Number

│  ├─ source: String

│  ├─ source\_info

│  │  ├─ img: String

│  │  ├─ lang: String

│  │  └─ name: String

│  ├─ tags: String

│  ├─ title: String

│  ├─ upvotes: String

│  └─ url: String

└─ users

   ├─ createdAt: Timestamp

   ├─ displayName: String

   ├─ email: String

   ├─ phone: String

   ├─ photoURL: String

   ├─ uid: String

   ├─ wallet

   │  ├─ amount: Number

   │  └─ id: String

   └─ transactions

      ├─ accountBalance: Number

      ├─ amount: Number

      ├─ creditCardDetails // only for buy transactions type

      │  ├─ cardName: String

      │  ├─ cardNumber: String

      │  ├─ ccv: String

      │  └─ expDate: String

      ├─ currencyId: String

      ├─ price: Number

      ├─ timestamp: Timestamp

      ├─ transactionType: String // "buy" or "trade" or " withdraw"

      ├─ boughtCurrency // only for trade transactions type

      │  ├─ amount: Number

      │  └─ id: Number

      ├─ soldCurrency // only for trade transactions type

      │  ├─ amount: Number

      │  └─ id: Number

      ├─ bankAccountDetails // only for withdraw transactions type

      │  ├─ accountNumber: String

      │  ├─ bankNumber: String

      │  └─ branchNumber: String

      └─ price: Number

* Cryptocurrencies Collection

General Attributes: Includes a comprehensive set of attributes for each cryptocurrency, such as ath (all-time high), atl (all-time low), market cap, supply details, and price changes. These attributes enable detailed tracking of each cryptocurrency's performance.

Sparkline Data: The sparkline\_in\_7d.price array holds 168 values, presumably representing hourly price points over the past 7 days. This is a great feature for visualizing price trends.

Data Freshness: The last\_updated attribute is crucial for ensuring users receive the most current data. However, consider storing this as a Timestamp instead of a Number for consistency and ease of use with time-based queries.

* CryptoNews Collection

Rich Content Structure: Each news item includes content (body), categorization (categories, tags), voting mechanism (upvotes, downvotes), and source information (source, source\_info). This structure supports a dynamic and interactive news feed.

Localization: The lang attribute in both news items and source info suggests support for multiple languages, enhancing user experience across different regions.

Engagement Tracking: Using upvotes and downvotes as Strings is unusual. Typically, these would be Numbers to easily calculate total votes. If the intention is to track individual user votes, consider a separate collection or subcollection for scalability.

* Users Collection

Profile Information: Contains standard user profile information (displayName, email, photoURL) along with wallet and transactions for financial interactions.

Wallet: Simple structure with an amount and an id. It's unclear whether id links to a currency or is a wallet identifier. Clarification or renaming could improve understanding.

Transactions: Supports various types (buy, trade, withdraw) with specific attributes for each. This flexible structure accommodates different transaction scenarios effectively.

Security Concern: Storing creditCardDetails in plain text poses a significant security risk. Consider implementing a more secure method of handling sensitive financial information, such as tokenization or using secure payment gateways.

Trade Details: For trades, the inclusion of both boughtCurrency and soldCurrency is good practice. However, the use of id as a Number for boughtCurrency and potentially also for soldCurrency (it's not specified but implied) might be an inconsistency if currencyId elsewhere is a String.