# CRYPTOCUREENCY PORTFOLIO TRACKER

A PROJECT REPORT

#### Submitted by

SHREYAS CHOWDHURY [RA2211027010028]

MUFEEDA O [RA2211027010035]

SACHIN RAJESH PAL [RA2211027010061]

#### Under the Guidance of

## Dr. P. VARUN

Assistant Professor, Department of Data Science and Business System

### in partial fulfillment of the requirements for the degree of

## BACHELOR OF TECHNOLOGY

## in

## COMPUTER SCIENCE ENGINEERING

## with specialization in BIG DATA ANALYTICS



## DEPARTMENT OF

## DATA SCIENCE AND BUSINESS SYSTEM

## COLLEGE OF ENGINEERING ANDTECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## KATTANKULATHUR- 603 203

## MAY 2025

Department of Data Science and Business System

##### SRM Institute of Science & Technology

##### Own Work Declaration Form

**Degree/ Course : Bachelor of Technology in Computer Science with Specialization in**

**Big Data Analytics**

**Student Name : Shreyas Chowdhury, Mufeeda O, Sachin Rajesh Pal**

##### Registration Number : RA2211027010028 ,RA2211027010035,RA2211027010061

**Title of Work : Cryptocurrency Portfolio Tracker**

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated and that we have met the following conditions:

* Clearly referenced all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that we have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| We are aware of and understand the University’s policy on Academic misconduct and plagiarism and we certify that this assessment is our own work, except where indicated by referring and that we have followed the good academic practices noted above.  RA2211027010028  RA2211027010035  RA2211027010061 |
| 05/05/2025 05/05/2025 |

# Logo

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

## BONAFIDE CERTIFICATE

Certified that 21CSS301T - Project report titled **“CRYPTOCURRENCY PORTFOLIO TRACKER ” is the Bonafide work of “SHREYAS CHOWDHURY [RA2211027010028], MUFEEDA O [RA2211027010035], SACHIN RAJESH PAL[RA2211027010061]”** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

|  |  |  |
| --- | --- | --- |
| **Dr. Varun P**  **Assistant Professor Department of Data**  **Science and Business System** |  | **Dr. V Kavitha**  **Professor and Head of**  **Department of Data**  **Science and Business Systems** |

**ACKNOWLEDGEMENTS**

We are deeply thankful to **Dr. C. Muthamizhchelvan**, Vice-Chancellor of SRM Institute of Science and Technology, for providing the necessary facilities and ongoing support for this project.

Our heartfelt appreciation goes to **Dr. Leenus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable encouragement.

We are grateful to **Dr. Revathi Venkataraman**, Professor and Chairperson of the School of Computing, SRM Institute of Science and Technology, for her guidance throughout this endeavour.

We also thank **Dr. M. Pushpalatha**, Professor and Associate Chairperson – CS and **Dr. C.Lakshmi**, Professor and Associate Chairperson – AI, School of Computing, SRM Institute of Science and Technology, for their insightful contributions.

We extend our gratitude to **Dr. V Kavitha**, Head of the Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her advice and motivation at every stage of this project.

Our thanks go to the Project Coordinators, Panel Head and Panel Members of the Department of Data Science and Business System, SRM Institute of Science and Technology, for their feedback during project reviews.

We are immensely appreciative of our Faculty Advisor, **Dr. Panimalar K**, from the Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her guidance in navigating the course requirements.

We owe a special debt of gratitude to our project guide, **Dr. P. Varun**, from the Department of Data Science and Business Systems, SRM Institute of Science and Technology, for giving us the opportunity to work under his mentorship. His encouragement to explore diverse research topics and his dedication to impactful problem-solving have been truly motivating.

We also acknowledge the support of all faculty members in the Department of Data Science and Business Systems, SRM Institute of Science and Technology, for their assistance throughout this project. Lastly, we are grateful to our parents, family and friends for their unwavering love, support and encouragement.

### **ABSTRACT**

This project, titled Kuberdhan, is a modern web-based trading and financial dashboard built using the Next.js framework, TypeScript and Tailwind CSS. It serves as a unified platform for users interested in stock market activities, financial news, portfolio tracking and simulated paper trading. The project follows the App Router architecture introduced in recent versions of Next.js, providing a modular, scalable and server-friendly structure that enhances both performance and maintainability.

Key features of Kuberdhan include user authentication (login/signup), a personalized dashboard, real-time market data views, a news feed, alerts for significant financial events and a paper trading interface for simulating investments without risk. The inclusion of a profile section and an achievements page suggests a focus on user engagement and education, making the application suitable for both novice traders and experienced market enthusiasts. The use of Tailwind CSS ensures responsive design and modern UI consistency across components.

From a technical standpoint, the project showcases a clean separation of concerns, leveraging Next.js's file-based routing and the flexibility of TypeScript to build a robust, type-safe frontend application. Configuration files such as tailwind .config.ts, postcss.config.mjs and tsconfig.json demonstrate a developer-friendly environment optimized for collaboration and performance. This project lays a strong foundation for future enhancements such as real-time socket integration, AI-driven trading suggestions, or integration with brokerage APIs for live trading.

Its technical and functional strengths, Kuberdhan emphasizes user-centric design and modular development, making it adaptable for a wide range of financial applications. Its architecture supports future integration of advanced tools such as machine learning-based market prediction, personalized financial insights or blockchain-backed asset tracking. The clean and extensible codebase, along with the use of modern development practices, positions Kuberdhan as both a practical solution for individual investors and a valuable foundation for educational or enterprise-level financial platforms.

**TABLE OF CONTENTS**

**ABSTRACT 3**

**TABLE OF CONTENTS** **4**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| **1** | **INTRODUCTION** | **5** |
| 1.1 | Introduction to Project | 5 |
| 1.2 | Problem Statement | 6 |
| 1.3 | Objectives of the Project | 6 |
| 1.4 | Scope of the application | 7 |
| **2** | **TECHNOLOGY STACK** | **8** |
| 2.1 | Frontend | 8 |
| 2.2 | Backend | 8 |
| 2.3 | Database | 8 |
| 2.4 | Tools Used | 8 |
| **3** | **SYSTEM DESIGN** | **9** |
| 3.1 | Architecture Diagram | 9 |
| 3.2 | Database Schema | 10 |
| **4** | **MODULE DESCRIPTION** | **12** |
| 4.1 | Authentication Module | 12 |
| 4.2 | Landing Page Module | 13 |
| 4.3 | Dashboard Module | 13 |
| 4.4 | Market/Product Management | 14 |
| 4.5 | Paper Trading Module | 14 |
| 4.6 | News Module | 15 |
| 4.7 | Alerts Module | 15 |
| 4.8 | Achievements Module | 16 |
| 4.9 | Profile Management Module | 16 |
| 4.10 | UI/UX & Theming Module | 17 |
| **5** | **IMPLEMENTATION DETAILS** | **18** |
| 5.1 | Overview of Features Developed | 18 |
| 5.2 | Code Structure | 21 |
| **6** | **DEPLOYMENT DETAILS** | **23** |
| 6.1 | GitHub Repository Link | 23 |
| 6.2 | Vercel (Frontend) Link | 23 |
| **7** | **PROJECT IMPLEMENTATION STATUS TABLE** | **24** |
| **8** | **REFERENCES** | **26** |
| 8.1 | Tutorials & Learning Resources | 26 |
| 8.2 | Tools & Platforms | 26 |
| **9** | **APPENDIX** | **27** |
| 9.1 | Screenshots | 27 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction to Project:**

In today’s fast-paced financial environment, access to real-time market data, personalized insights and intuitive trading tools is essential for both new and experienced investors. However, many platforms are either too complex for beginners or lack the flexibility and customization that advanced users require. To address this gap, Kuberdhan was developed as a modern, web-based financial dashboard that consolidates essential trading and investment features into a single, user-friendly platform.

Built using Next.js, TypeScript and Tailwind CSS, Kuberdhan leverages cutting-edge web technologies to deliver a responsive, secure and scalable application. It includes modules for user authentication, market tracking, financial news, trading alerts, simulated (paper) trading and user achievements—making it an all-in-one solution for learning, experimenting and staying informed about the financial world. The use of the App Router in Next.js ensures efficient routing and layout management, while the modular design promotes maintainability and future expansion.

Kuberdhan not only serves as a practical tool for retail investors and finance students but also provides a foundation for integrating more advanced features like AI-driven market predictions, portfolio optimization algorithms and real-time brokerage connections. This introduction lays the groundwork for understanding the project's goals, technical components and its potential as a learning and trading companion in the digital finance space.

**1.2 Problem Statement**

Many existing financial and trading platforms are either too complex for beginners or lack flexibility for advanced users. There is a need for a unified, user-friendly solution that offers essential features like market tracking, financial news, simulated (paper) trading and alerts within a modern interface. Educational tools often fail to reflect real market dynamics, limiting their effectiveness. Kuberdhan addresses this gap by providing an intuitive, modular web application built with Next.js and Tailwind CSS. It caters to both novice and experienced users, supporting learning, experimentation and portfolio monitoring, while being scalable for future enhancements like AI-based insights and real-time trading APIs.

**1.3 Objectives of the Project**

The primary objective of the Kuberdhan project is to design and develop a comprehensive, web-based financial dashboard that enables users to monitor market trends, access financial news, simulate trading strategies and receive customized alerts. The project aims to:

* Provide an intuitive and responsive user interface for seamless navigation and interaction.
* Offer paper trading features to help users practice trading without financial risk.
* Integrate modules for user authentication, profile management and performance tracking.
* Aggregate financial news and market data in real time to support informed decision-making.
* Create a scalable and modular architecture to support future features like AI-based recommendations and live trading integration.

**1.4 Scope of the Application**

The *Kuberdhan* application is designed to serve as a comprehensive financial dashboard that caters to both beginner and intermediate-level investors. Its scope includes:

* **User Management**: Supporting user registration, login and profile handling.
* **Dashboard**: Providing an overview of market activity, user progress and quick access to key modules.
* **Market Tracking**: Displaying real-time or regularly updated stock market data and trends.
* **Paper Trading**: Allowing users to simulate trading strategies without using real money, promoting safe learning.
* **News Integration**: Aggregating financial news to keep users informed of market-moving events.
* **Alerts System**: Enabling users to set custom alerts for market changes or asset price movements.
* **Achievements & Learning**: Tracking user milestones and encouraging engagement through gamification.
* **Modular Design**: Ensuring that new features such as AI-based trading assistants or brokerage API integration can be easily added.

**CHAPTER 2**

**TECHNOLOGY STACK**

**2.1 Frontend**  
The frontend is developed using React via the Next.js framework, which supports server-side rendering and dynamic routing. It is styled with Tailwind CSS for rapid and responsive UI development.

**2.2 Backend**  
The backend logic is intended to run on Node.js with support from Express.js for handling server-side APIs and routing. While this project currently focuses on frontend routing, it is designed to accommodate backend integration as needed.

**2.3 Database**  
MongoDB is used as the primary database for storing user data, trading history, alerts and other relevant information. It offers flexibility in handling structured and unstructured data.

**2.4 Tools Used**

GitHub for version control and project collaboration.

Netlify or Vercel (commonly used with Next.js) for continuous deployment and hosting.

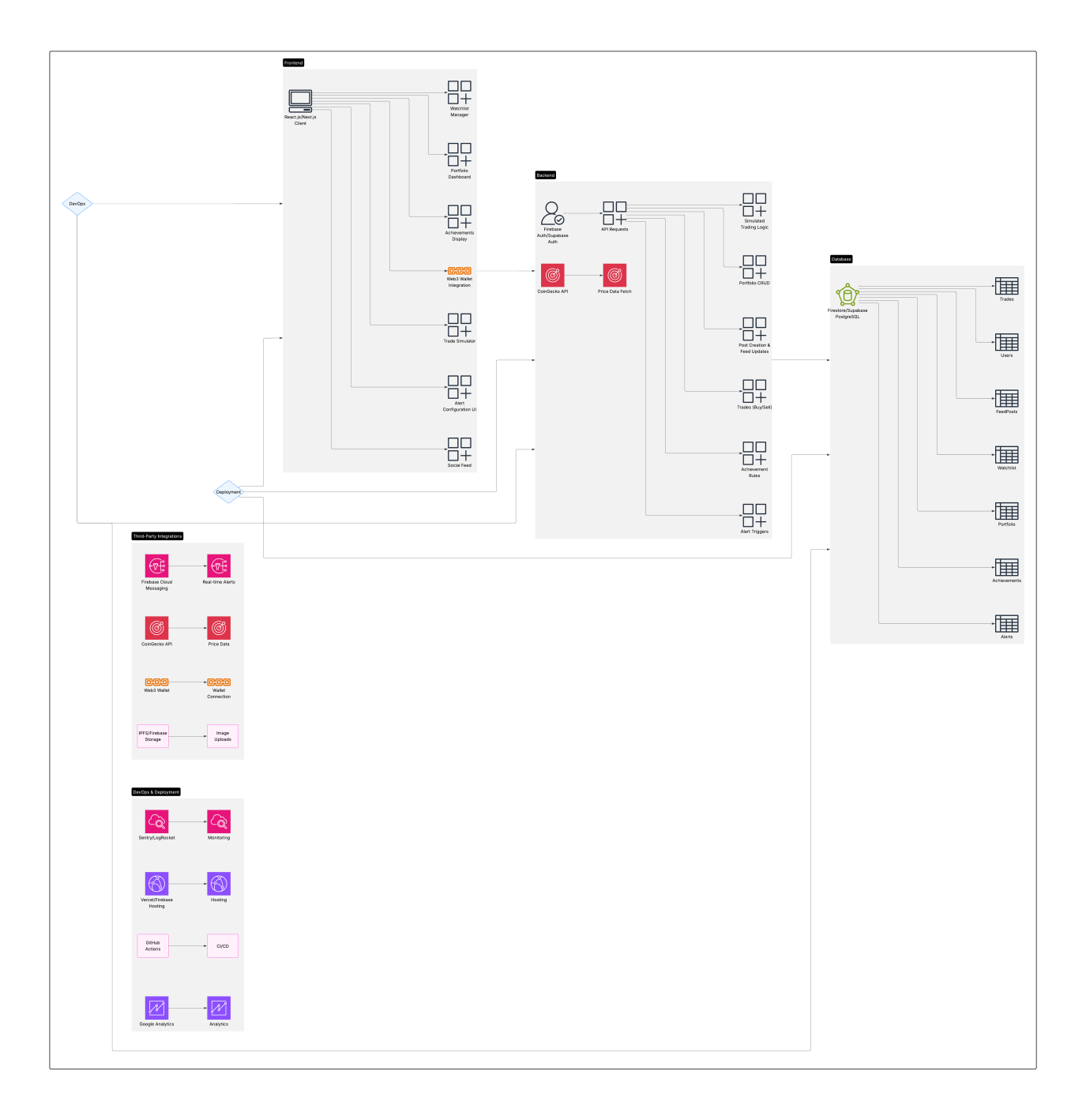
PostCSS and Tailwind CLI for advanced CSS processing.

pnpm as the package manager for faster and more efficient dependency handling.

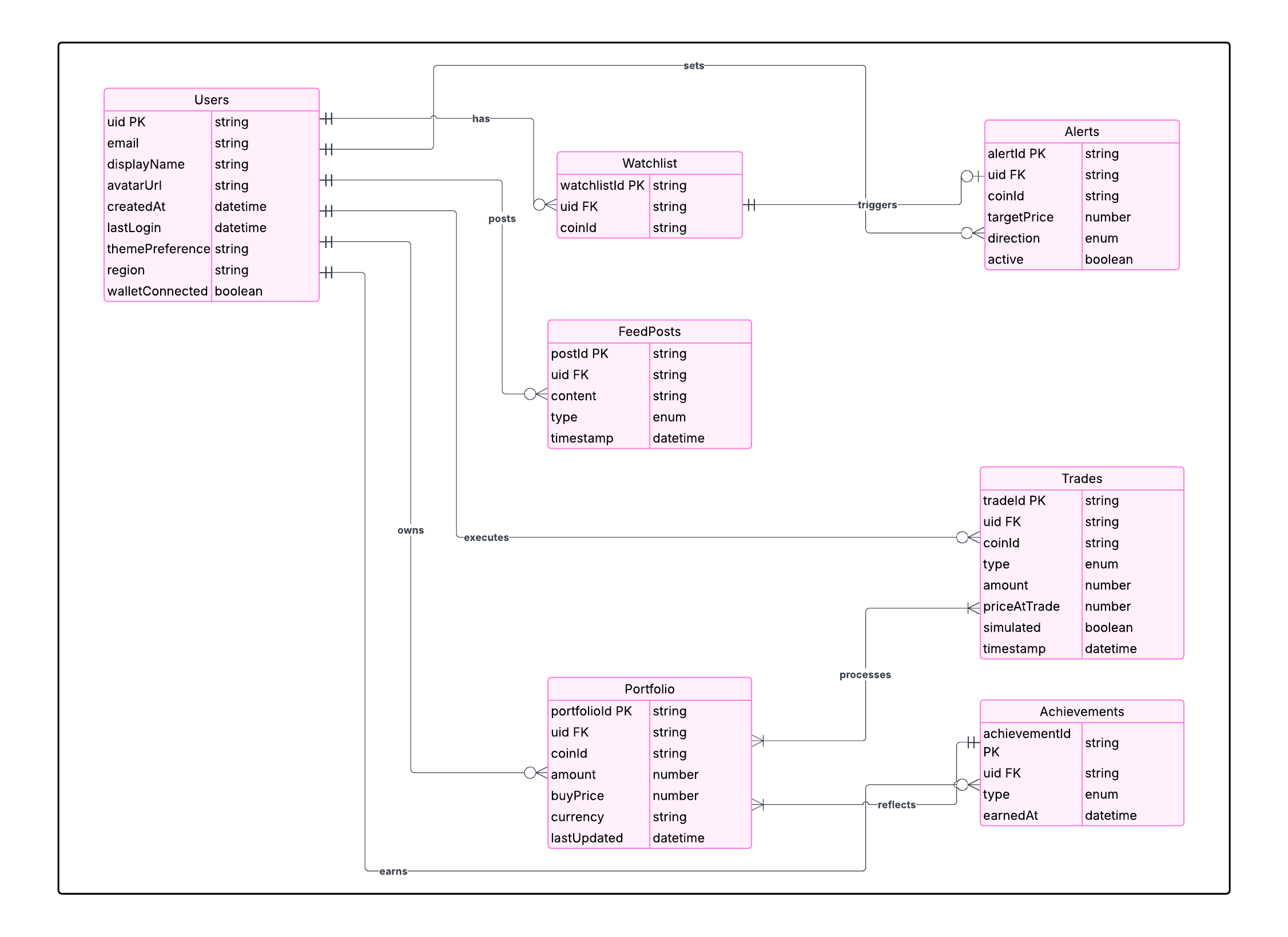
**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Architecture Diagram**



**3.2 Database Schema**



**CHAPTER 4**

**MODULE DESCRIPTION**

**4.1 Authentication Module**

Files Involved:

* login-page.tsx
* signup-page.tsx
* auth-provider.tsx

Description:  
This module manages the secure login and registration functionalities. The login-page.tsx provides a form where users can enter credentials to access the application, while the signup-page.tsx enables new users to register an account. The auth-provider.tsx is responsible for handling authentication logic—like session persistence, user context management and route guarding. This could be using third-party services such as Firebase, Supabase, or a custom backend.

Key Features:

* Secure login & logout
* Password handling and validation
* Persistent user sessions using context or cookies
* Possibly supports OAuth or email-based verification

**4.2 Landing Page Module**

File:

* landing-page.tsx

Description:  
The Landing Page is the user's first impression of the application. This component likely contains promotional content, app features overview, navigation to login/signup pages and visual highlights of the app's functionality. It may also contain calls to action for user onboarding.

Key Features:

* Introductory banners or hero section
* Feature highlights or how-it-works
* Navigation to other modules like login or market

**4.3 Dashboard Module**

File:

* dashboard-page.tsx

Description:  
The Dashboard serves as the central hub for users once logged in. It aggregates essential user-specific data like portfolio value, watchlist, market summaries and trading performance. This module offers users a snapshot of their activities and quick links to other key areas like alerts, trading, or profile.

Key Features:

* Interactive charts and stats
* Overview of market data or user trades
* Widgets for alerts, recent activity, achievement.

**4.4 Market / Product Management Module**

File:

* market-page.tsx

Description:  
This module displays real-time or simulated market data—such as cryptocurrency prices, stock values, or asset listings. It likely allows users to browse different products, add them to their watchlist and perhaps trigger trades or alerts. This is the core of any trading or investment-oriented platform.

Key Features:

* List and detail view of market instruments
* Price tracking and charting
* Integration with alerts or trading modules

**4.5 Paper Trading Module**

File:

* paper-trading-page.tsx

Description:  
Paper trading allows users to simulate trades with virtual currency. This module is crucial for learning and testing strategies without financial risk. It mimics real-time trading environments, allowing users to practice and analyze trade performance.

Key Features:

* Buy/sell simulations
* Virtual wallet balance
* Trade history and performance metrics

**4.6 News Module**

File:

* news-page.tsx

Description:  
This module provides users with financial news, market insights, or educational articles. The content may be static, pulled from APIs, or customized based on user interests or portfolio. It helps users stay informed about market trends and events that impact trading decisions.

Key Features:

* News feeds categorized by market or topic
* External news API integration
* Article previews and full content views

**4.7 Alerts Module**

File:

* alerts-page.tsx

Description:  
Users can set alerts based on market triggers (e.g., price threshold, volume change). This module allows managing these alerts—setting conditions, toggling on/off, or viewing triggered alerts. It ensures users never miss significant market movements.

Key Features:

* Create, edit and delete alerts
* Real-time or scheduled notifications
* Integration with market or trading module

**4.8 Achievements Module**

File:

* achievements-page.tsx

Description:  
This gamification feature motivates users by rewarding achievements—such as completing tutorials, hitting trade milestones, or using advanced features. It encourages user engagement and progress tracking.

Key Features:

* Badges and milestone indicators
* User progression logic
* Visual leaderboard or timeline

**4.9 Profile Management Module**

File:

* profile-page.tsx

Description:  
This module allows users to view and update their personal information, including email, password, avatar and preferences. It may also offer account deletion or security settings.

Key Features:

* Editable user info forms
* Avatar and theme preferences
* Privacy and account controls

**4.10 UI/UX & Theming Module**

Files:

* theme-provider.tsx, theme-toggle.tsx,
* All ui/\* components (e.g., buttons, dialogs, cards, charts, etc.)

Description:  
This module provides the reusable UI components and theme infrastructure used across the application. It supports light/dark mode, consistent design systems using TailwindCSS and accessibility-friendly components. These files ensure a smooth and unified user experience.

Key Features:

* Dark/light mode switcher
* Component library (buttons, forms, charts)
* Responsive design using TailwindCSS

**CHAPTER 5**

**IMPLEMENTATION DETAILS**

**5.1 Overview of Features Developed**

User Authentication System

* Secure Login and Signup pages for user access.
* Uses an auth-provider.tsx to manage session state, user context and protect private routes.
* Provides a foundation for user-specific interactions across the platform.

Landing Page Experience

* Acts as the homepage for new visitors.
* Showcases application features and serves as the entry point to login or signup.
* Designed to be visually engaging and informative.

User Dashboard

* Displays a centralized overview of the user’s data and interactions.
* May include widgets like trading history, news highlights, or account stats.
* Built using reusable UI components for responsiveness.

Market View / Product Management

* A dedicated Market Page where users can explore various products (stocks, crypto, etc.).
* Allows viewing of current prices, historical trends and possibly watchlists.
* Sets the groundwork for future trading or investment actions.

Paper Trading Feature

* Simulates real-time trading with virtual funds.
* Users can practice buying/selling assets without risking actual money.
* Tracks performance and educates users on trading strategies.

Real-Time Alerts System

* Users can configure price alerts or market movement triggers.
* Alerts are managed and viewed via the Alerts Page.
* Enhances user awareness and trading responsiveness.

Financial News Feed

* Presents up-to-date market and financial news.
* Keeps users informed on trends that affect investments.

Achievements / Gamification

* Motivates users by offering badges or levels for activities like trading or learning.
* Tracks progress and shows achievement history.
* Encourages continued use through a reward-based system.

User Profile Management

* Users can update their personal information, settings and preferences.
* Includes features like changing avatars, updating emails, or managing themes.

Dynamic Theme Support

* Integrated dark/light mode switching.
* Custom theming handled by theme-provider.tsx and theme-toggle.tsx.
* Enhances user experience and accessibility.

Componentized UI System

* A large set of reusable UI components under components/ui/ (e.g., forms, buttons, cards, modals).
* Built with Tailwind CSS for consistency and responsiveness.
* Enables rapid development and cleaner code organization.

**5.2 Code Structure**

kuberdhan/

├── app/

│ ├── globals.css # Global CSS styles (Tailwind base setup)

│ ├── layout.tsx # Root layout component for all pages

│ ├── page.tsx # Landing page entry (renders <LandingPage />)

│ └── (other optional routes) # Additional page-level routes can be added

│

├── components/

│ ├── landing-page.tsx # Landing page UI logic

│ ├── login-page.tsx # Login UI

│ ├── signup-page.tsx # Signup UI

│ ├── dashboard-page.tsx # Dashboard view for logged-in users

│ ├── market-page.tsx # Displays financial market/product data

│ ├── paper-trading-page.tsx # Paper trading simulation module

│ ├── news-page.tsx # News listing component

│ ├── alerts-page.tsx # User alert configurations

│ ├── achievements-page.tsx # Badge and milestone display

│ ├── profile-page.tsx # User profile & account settings

│ ├── auth-provider.tsx # Authentication context provider

│ ├── theme-provider.tsx # Light/Dark theme manager

│ ├── theme-toggle.tsx # UI button to toggle themes

│ └── ui/ # Folder for reusable UI components

│ ├── button.tsx # Standard button component

│ ├── input.tsx # Form input field

│ ├── dialog.tsx # Modal dialogs

│ └── ... # More utility components (badges, cards, etc.)

│

├── public/ # Static files like logos, icons, assets

│

├── styles/ # Custom CSS (if separated)

│

├── utils/ # Utility functions (if present)

│

├── package.json # Project metadata and dependencies

├── tsconfig.json # TypeScript configuration

├── tailwind.config.ts # Tailwind CSS setup

├── postcss.config.mjs # PostCSS config (used by Tailwind)

├── next.config.mjs # Next.js configuration

└── .gitignore # Git tracking rules

**CHAPTER 6**

**DEPLOYMENT DETAILS**

**6.1 GitHub Repository Link**

**6.2 Vercel Link**

*kuberdhan.vercel.app*

**CHAPTER 7**

**PROJECT IMPLEMENTATION STATUS TABLE**

| **Module** | **Description** | **Status** | **Notes** |
| --- | --- | --- | --- |
| Authentication | Login, Signup and session handling | Completed | Auth provider implemented; login/signup pages functional |
| Landing Page | Introductory page with app overview | Completed | Landing UI is in place and renders correctly |
| Dashboard | User activity overview and stats | Completed | Dashboard component is present, but may need real data integration |
| Market/Product View | Displays market or asset information | Completed | Functional UI; may require dynamic API integration |
| Paper Trading | Simulated trading with virtual funds | Completed | UI and core logic implemented; can be tested further |
| Alerts | Create and manage market alerts | Completed | Alert UI exists; backend logic or real-time notifications may be pending |
| News Feed | Display market/financial news | Completed | UI is set up; may need integration with news API |
| Achievements | Badge system and user progress tracking | Completed | Static structure exists; dynamic tracking logic might be pending |
| Profile Management | Manage user info and settings | Completed | Basic profile page functional; may need settings enhancements |
| Theme Support | Light/Dark mode toggling | Completed | Theme provider and toggle work as expected |
| UI Components | Reusable buttons, forms, dialogs, etc. | Completed | Wide set of components already in place using Tailwind |

**CHAPTER 8**

**REFERENCES**

**8.1 Tutorials & Learning Resources**

1. Next.js Documentation – https://nextjs.org/docs  
   Used for understanding the app directory structure, routing and layout systems.
2. Tailwind CSS Documentation – <https://tailwindcss.com/docs>Referred for responsive UI design and utility-first styling approach**.**

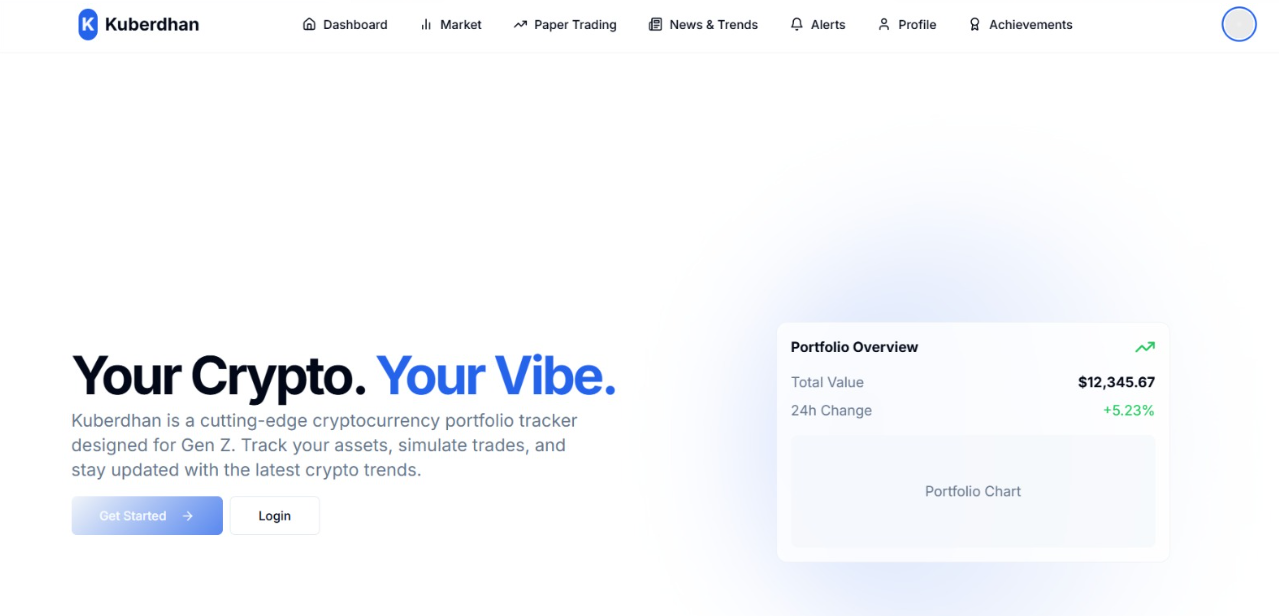
**8.2 Tools & Platforms**

* Visual Studio Code – Primary code editor.
* Vercel – Deployment platform for Next.js (if deployed).
* Postman – For API testing and debugging.

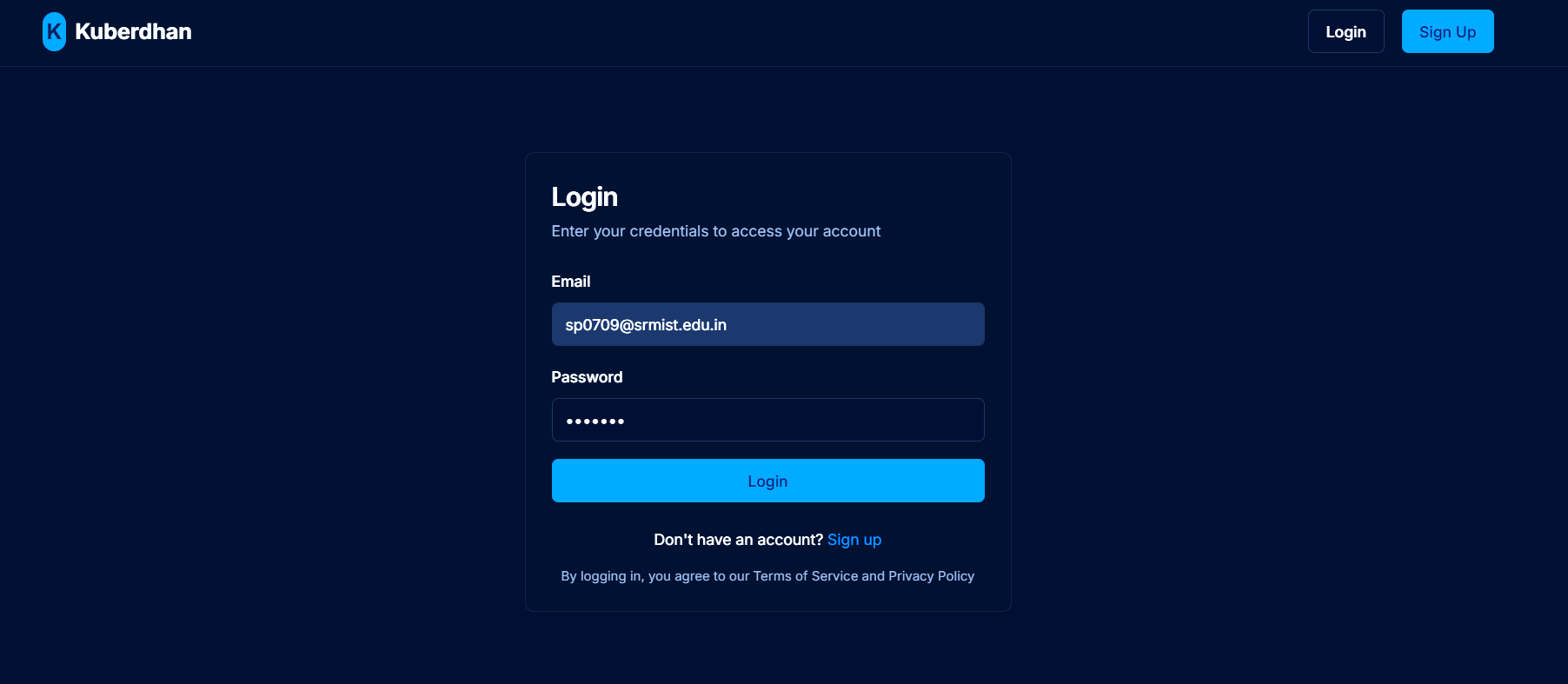
**CHAPTER 9**

**APPENDIX**

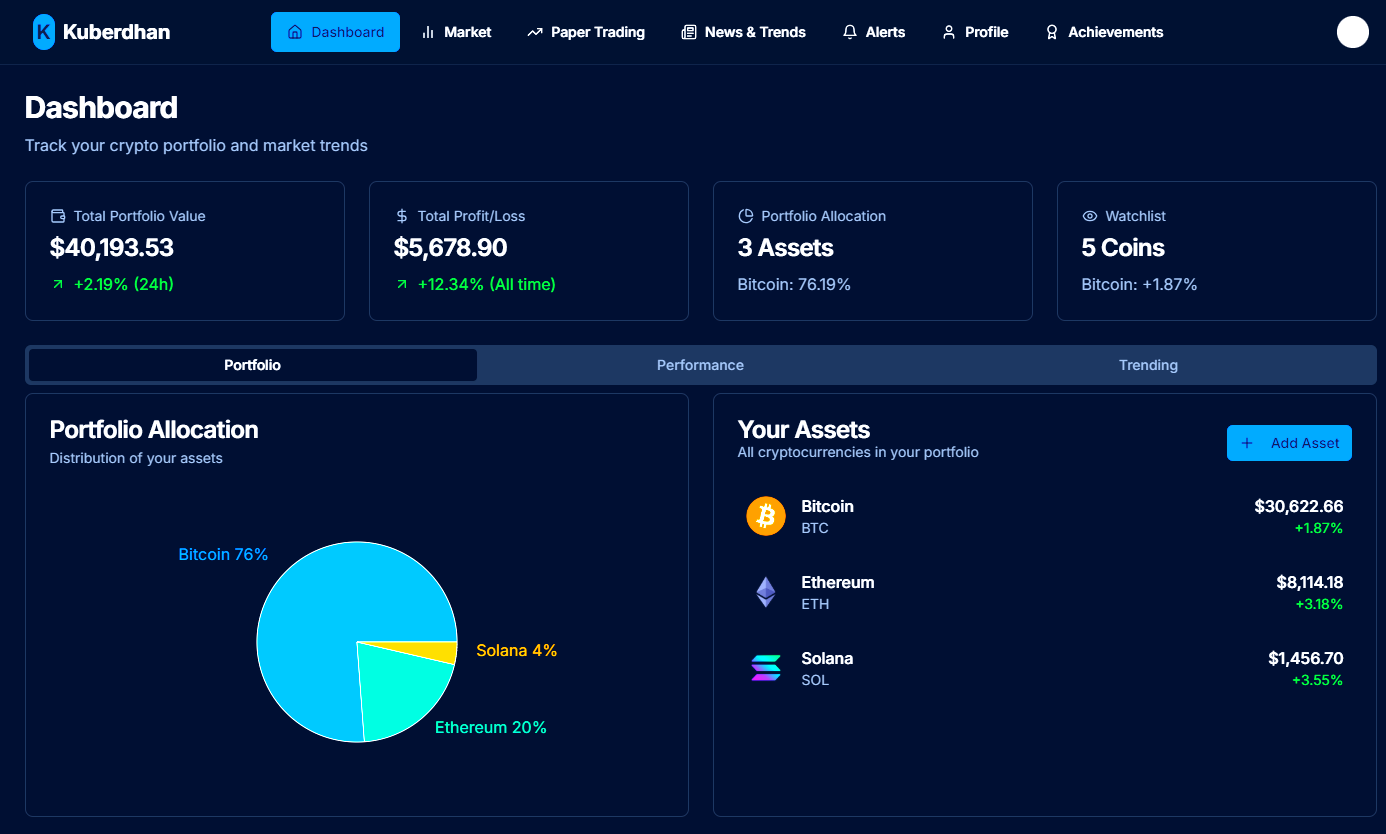
**9.1 Screenshots**

****

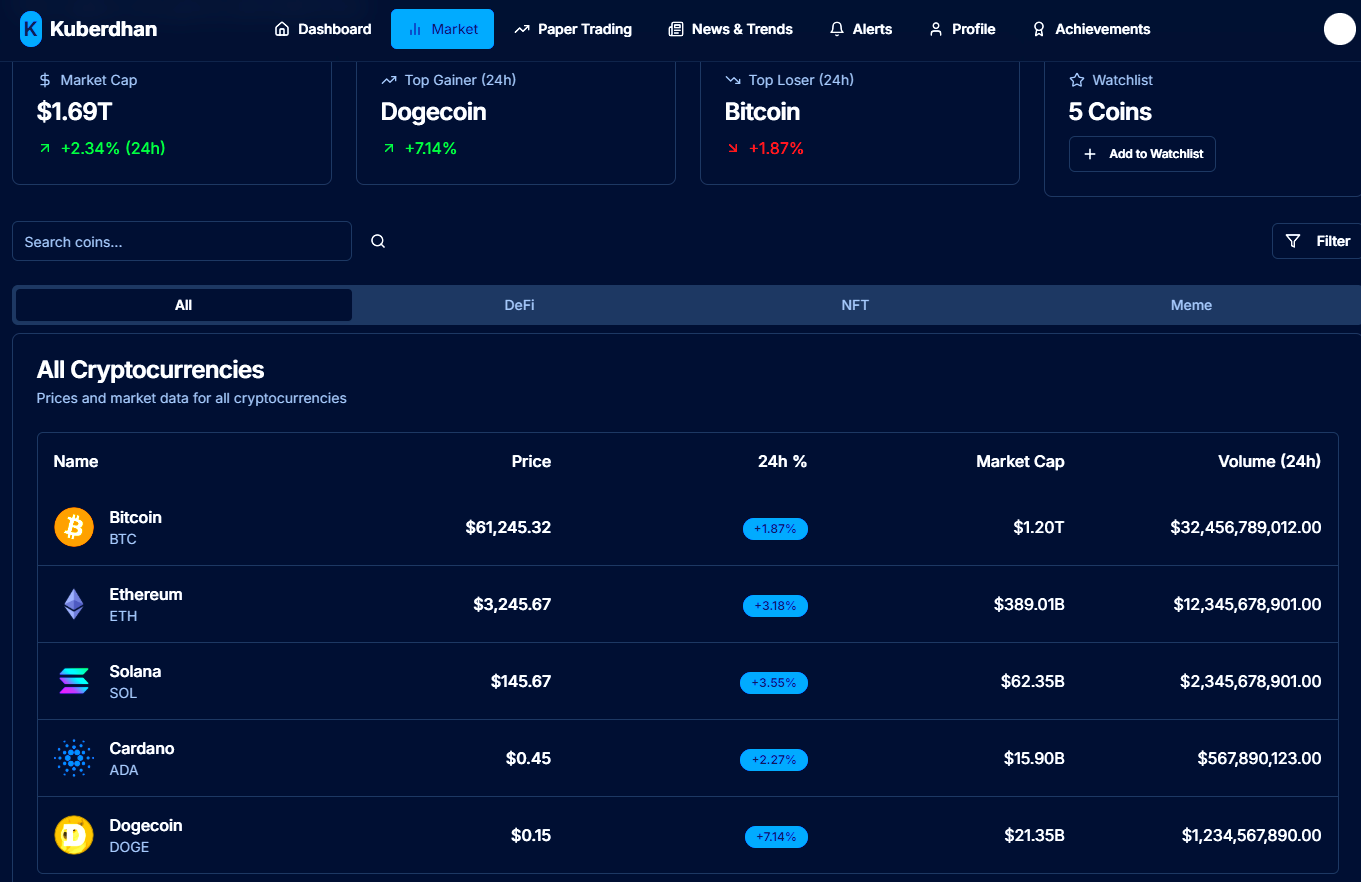
Home Page

****

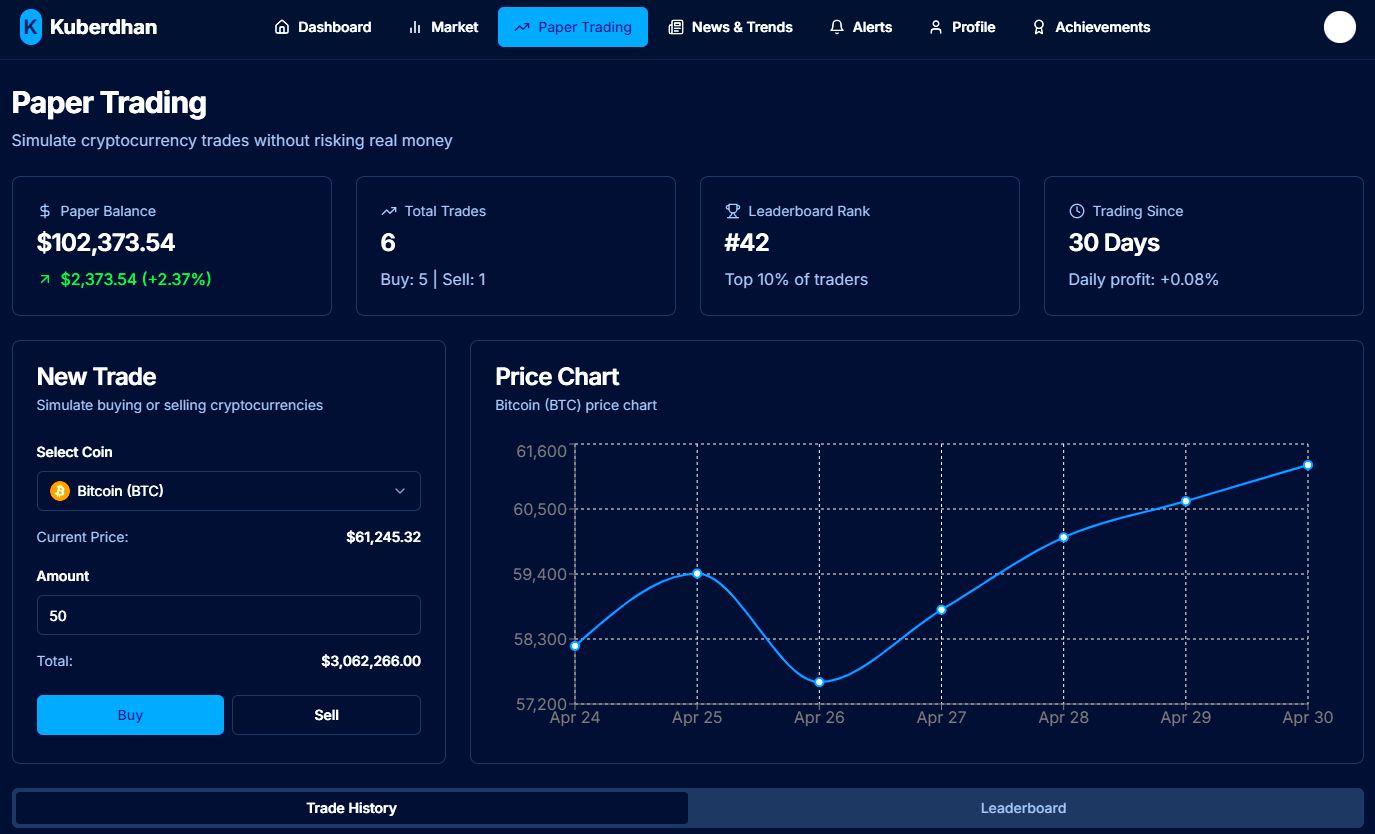
Login Page

****

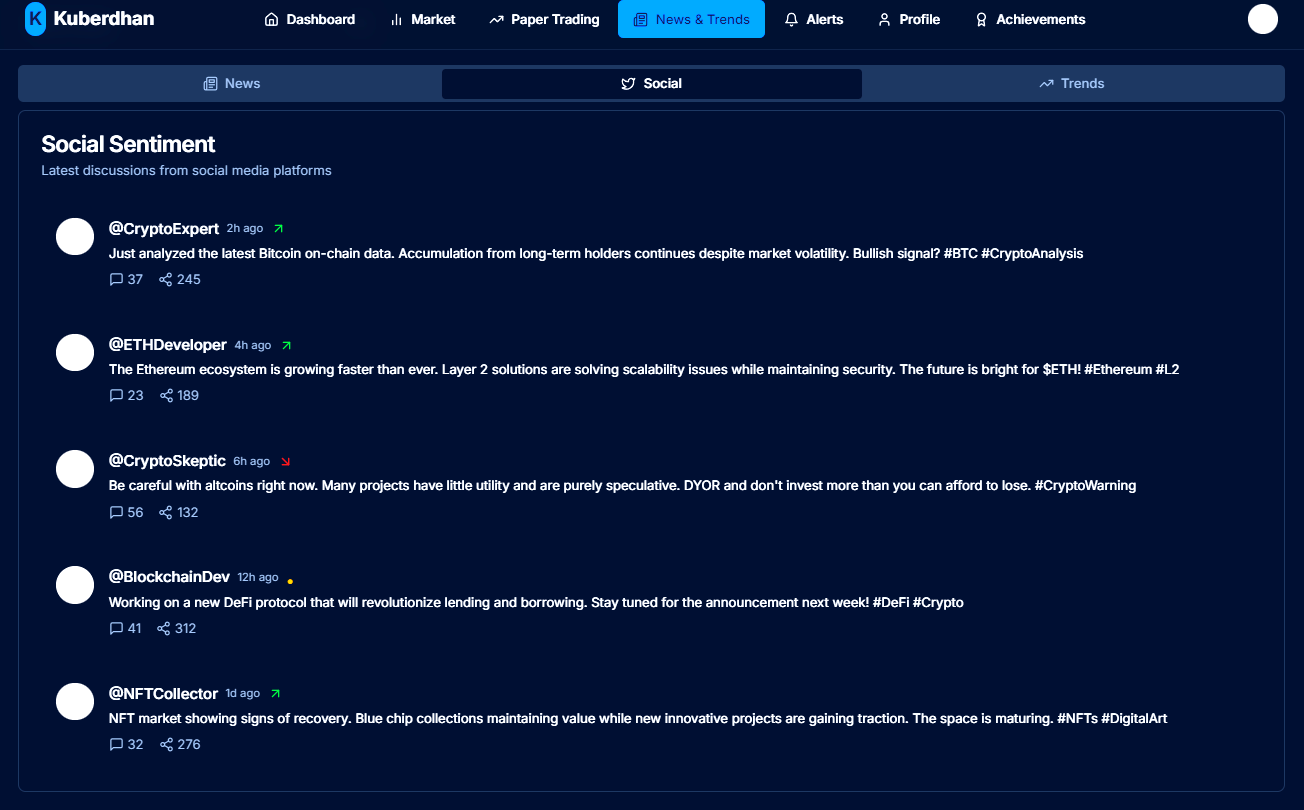
Dashboard

****

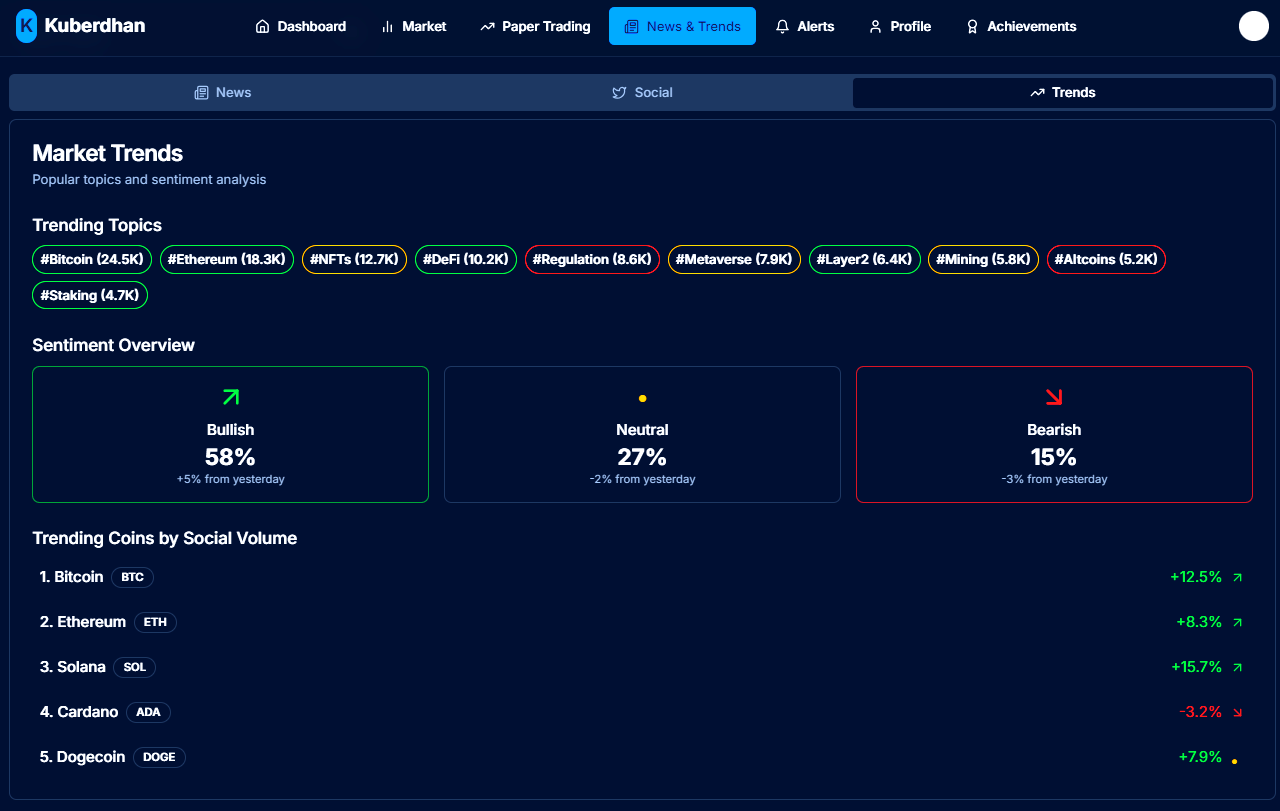
Market Information Page



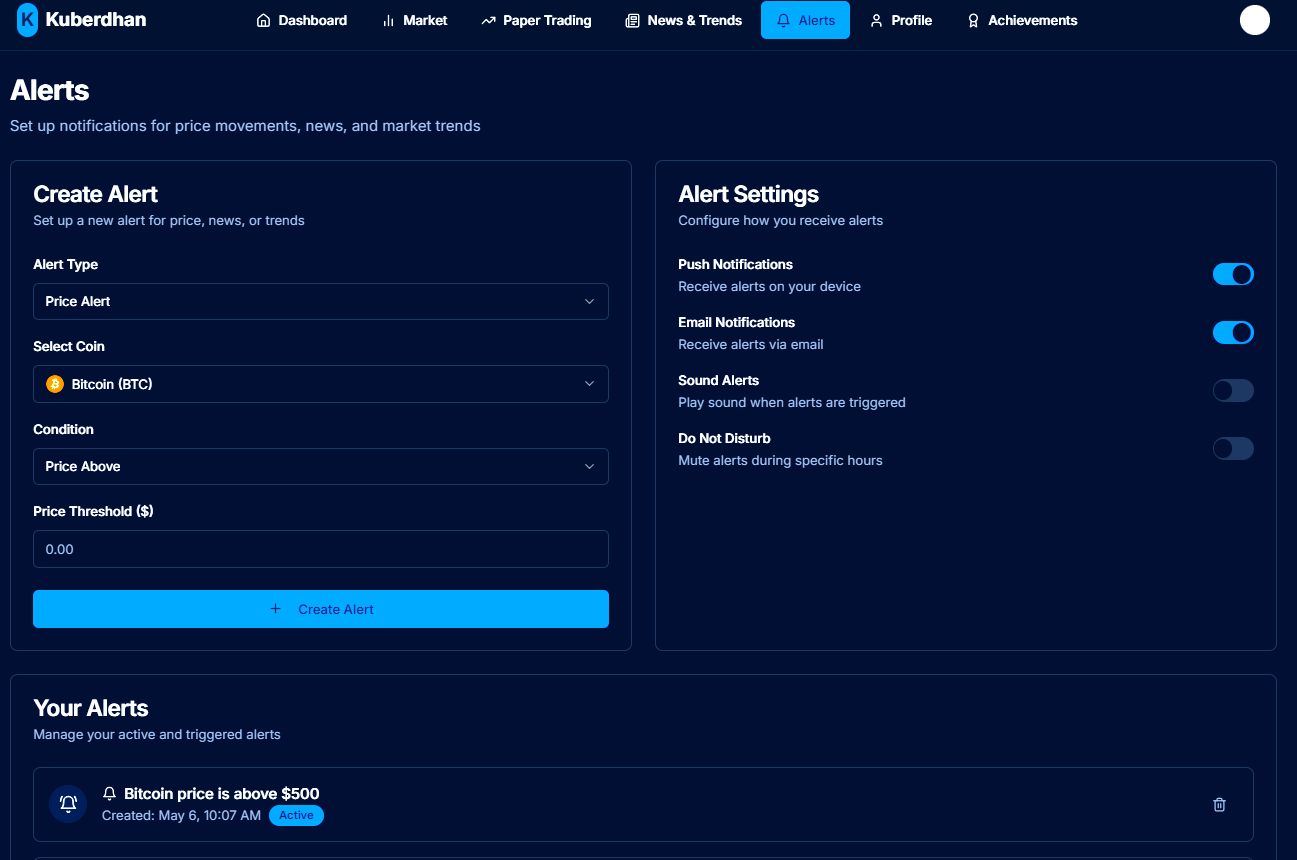
Paper Trading Page



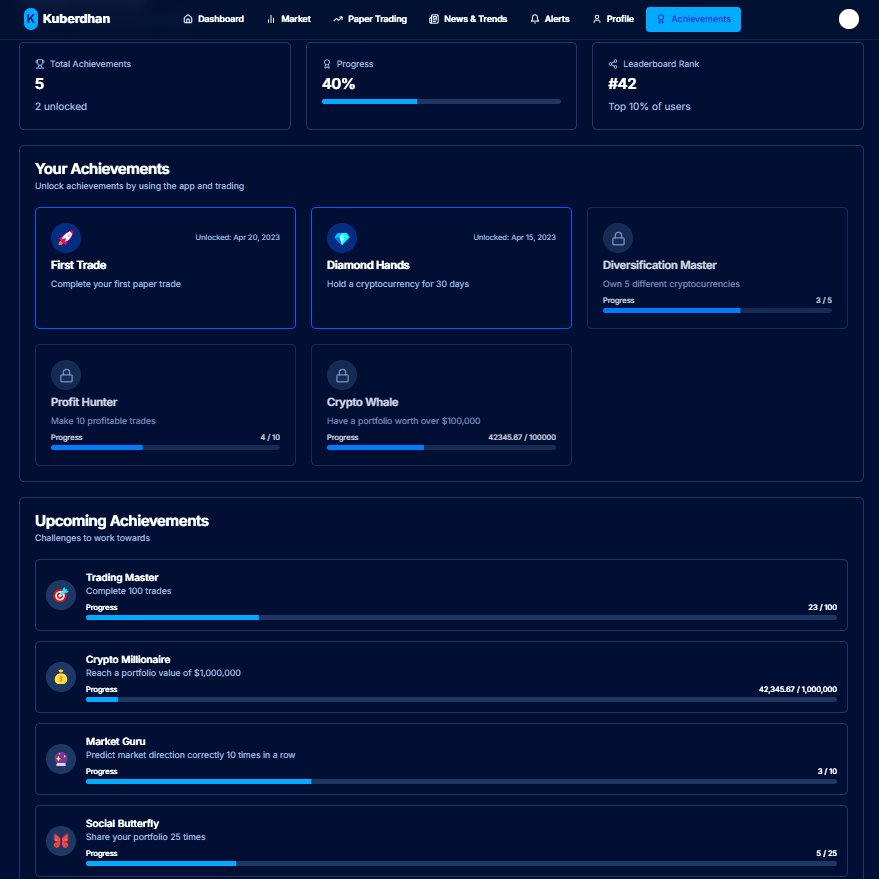
Social News Page



Market Trends Page



Alert Page



Achievements Page