**The University of Texas at Austin**

**Department of Computer Science**



**Final Project:**

**WealthAI**

**Group Members:**

**Mojtaba Noghabaei**

**ID: SN27936**

**Android Programming**

**App Overview:** *WealthAI* is an investment guidance application that utilizes advanced prediction models to offer personalized stock and cryptocurrency investment suggestions based on the user's risk tolerance. The application includes features such as user authentication, market data analysis, investment suggestion engine, and portfolio management.

**Note:** This was the initial idea, but I faced a devastating problem today. The API that I was using suddenly updated or changed and for some reason my code stopped working. I didn’t record a video of the working version and I understand if you don’t believe me! I tried hard and I have nothing to show for it! My whole life was like this, I came here when I was 22 and now, I am almost 30. I haven’t seen my family for the past 8 years except once. I know you don’t care, no one does. Since last week I have been thinking about committing suicide. I don’t even know why I am writing this.

**Major Challenges:**

During this project I faced multiple challenges. From dealing with limitation of alpha vantage API to issuing running the code. Despite my initial

1. Integration of real-time stock and cryptocurrency market data.
2. Implementation of a secure and efficient user registration and login system using Firebase.
3. Development of an intuitive and user-friendly interface for complex data presentation.

**User Interface Sketch:**

Screens screenshot of a mobile application

Description automatically generated

**APIs and Third-Party Libraries:**

* **Firebase Authentication:** Manages user registration and login processes.
* **Alpha Vantage and CoinGecko:** Provide real-time financial market data for stocks and cryptocurrencies.
* **Chart.js:** Used for visualizing data on the dashboard and prediction screens.

**AI Components:**

* Investment suggestion engine utilizes machine learning algorithms trained with historical market data to predict future trends and suggest investments tailored to the user's risk profile.

**UI/UX/Display Code:**

* Emphasis on simplicity and clarity in presenting complex financial data, ensuring that users can navigate through the information effectively.

Screens screenshot of a mobile application

Description automatically generated

A screen shot of a cell phone showing a graph

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Backend and Processing Logic:**

* The backend processes involve secure data handling, real-time data fetching, and computation-intensive prediction algorithms.

**Learning and Challenges:**

* **Key Learning:** Gained a deeper understanding of integrating real-time data with machine learning models in a mobile app environment.
* **Significant Challenge:** Overcoming the complexities of implementing an efficient live data feed without compromising app performance.

**Debugging and Development Insights:**

* Encountered and resolved issues with data synchronization between Firebase and the app, enhancing the login system's reliability and security.

**Repository and Build Instructions:**

* Repository Location: [GitHub Organization Link]
* Build and Run Instructions: Detailed steps to set up the environment and back-end services, ensuring easy replication and testing.

**Code Metrics:**

* Total lines of Java code: [Number]
* Total lines of XML code: [Number]
* A breakdown of code by module and functionality is included in the appendices.

**Video Demo Link:**

* [Insert Link] - A comprehensive video demonstration of the *WealthAI* functionalities.

**Database Schema:**

* Screenshot of the Firebase database schema showing key-value pairs and data types.

**Appendices:**

* Screenshots of the app demonstrating key functionalities.
* Code frequency graph from the GitHub repository.

A green and red graph

Description automatically generated

https://github.com/noghabaei/Wealth-AI

Project Writeup

Your project writeup should be a few pages of PDF file, turned in via canvas.

Make sure your report specifies the following items, but bear in mind that the report is just your chance to convince me that you did a good job. Include what you think is necessary to convince me of that fact. I love hearing about debugging war stories. I'm serious, it is a problem for me.

* Report your team members (Done)
* Report the title of your app **and a one or two line description**. It is very important that the people who did the project, the title of the app and its description are clearly visible at the top of the first page of your report.
* Include a screenshot of your app, hopefully one that illustrates key functionality.
* List the APIs your app uses, for example Google maps. Also let me know if you use certain Android features extensively like animation or you have custom controllers (navigation drawer, action bar, etc.).
* List all third party libraries you use, and what they do for your app (each in their own paragraph). Describe briefly what was good and/or challenging about using them.
* List all third party services you use, and what they do for your app (each in their own paragraph). Describe briefly what was good and/or challenging about using them.
* List any component substantially generated by AI. Describe your prompts and briefly summarize what was good and/or challenging about the process.
* Discuss anything noteworthy about your UI/UX/display code.
* Discuss anything noteworthy about your back end or processing logic.
* Discuss the most important or interesting thing you learned doing your project.
* Discuss the most difficult challenge you overcame and/or your most interesting debugging story.
* If necessary, briefly tell us how to build and run your project. Include details about how to set up back end services (if you use them). In the common case, we will rely on your demo, but just in case we have an issue, we'd like some tips.
* For the in-person class, please create a repo in our organization. Do your work there and make sure to keep github up to date. We will pull your code from the time the assignment is due. In your writeup, clearly identify the location of your repository. Submit your final code to this new project.  
  For masters students, do the same, but use the masters class organization. You can create your own private repo.
* For the masters class, you need a video demo. Please include a link to your video demo in your writeup. The video should be publicly accessible to anyone with the link.
* If you use firebase, include a printout (or screenshot) of your database schema. The schema is basically the structure and the type information. So print out the keys and values and the types for any important data you have in the database. Here is a simple example.A screenshot of a computer

  Description automatically generated
* Report the count of lines of code in your project. Use cloc (http://cloc.sourceforge.net/) to count the lines. Run this command on your code.

cloc -by-file-by-lang app/src/main/

Report the total Java and total XML lines that you wrote. Do not include boilerplate lines, external libraries, or other sources in the total of lines on which you claim authorship. Indicate how you calculate how many lines you authored if you report line totals. Maybe use a table. You can break down the lines into modules if you think that helps, but I'm most interested in the totals of code you wrote.

* We want you to include the code frequency graph that github gives you for your repository. Open your project's github page in a web browser. Click the "Insights" tab. Then click the "Code frequency" tab on the menu bar on the left. You will see the "Code frequency" graph. Put a screen shot of your code frequency figure into your writeup. Here is an example figure for reference. If you don't have a pro account, or can't provide this data, that is ok.A screenshot of a graph

  Description automatically generated