Group DynamicsDevs
Members: Andrew Jung, Williams Chen, Gowrisankar Palanickal, Nayib Mall
ITWS 4500 - Web Science Development

Project Proposal

Here were some of the initial ideas from brainstorming ideas for our group project:

- Stock and News Viewer: Find stocks that were severely impacted and look at the news articles for that day
- Personal Budgeting & Expense Tracker
    - could share purchasing history with other users

Summary

Our application aims to be a collaborative financial tracker that allows users to share and keep track of their finances with others. This platform will allow users to set up shared budgets, track spending, and engage with others, including friends, family, or even broader groups toward financial goals. The app will offer real-time spending insights, interactive charts, and personalized financial tips, helping users make better financial decisions. By sharing progress, offering encouragement, and setting challenges, the app will promote a sense of community around financial success. It will provide an intuitive interface with tools for goal tracking, budget alerts, and even group financial challenges, allowing users to gain better control over their finances through mutual encouragement and collaboration.

Users/Stakeholders

The main users of this app will be people who need help with budgeting, such as college students and recent graduates. College students are a good fit because the app can help them track their income and expenses, figure out where to cut back on spending, and manage their limited money better. Since they are young and often new to managing money, the app can teach them responsible financial habits. Recent graduates are also a great audience because many of them are earning good money for the first time. The app can help them create budgets, manage their expenses, and learn to save and spend wisely as they start their careers. Key stakeholders for this app include young adults, like students and recent graduates, as well as parents who want to help their kids learn about money and manage it responsibly. Financial institutions may also be stakeholders, as they could benefit from encouraging better money habits through tools like this.

Technologies

We plan to mainly use TypeScript along with the MERN stack (MongoDB, Express, React, and Node.js)– MongoDB as the database for storing user accounts, budgets, and transaction data; Express and Node.js as the backend framework for managing APIs, exporting transaction histories in PDF/CSV formats, and integrating with legacy bank APIs; React for building an interactive and dynamic user interface, allowing users to view interactive charts of spending habits, set financial goals, and track monthly budgets. Additionally, we aim to integrate external APIs like the Google Charts API for data visualization.

We will most likely be extending these technologies with additional packages in order to make developing easier, make our app more useful, and make our app even more interactive.

Functional and Non-Functional Requirements

One of the main functional requirements of our app will be user account management– users should be able to create accounts that save their personalized information and that they can log in and out of. Another functional requirement is budget and expense tracking. Users should be able to create, edit, and delete income, expenses, and budgets, with the app displaying and categorizing transaction summaries. Users should also be able to create and join shared budgets with other users; this collaborative budgeting should give users the ability to assign expenses to specific members and track contributions. The app will have functionality to set individual or group financial goals as well, including deadlines and progress tracking. There will also be interactive charts and graphs to make the financial data easier for the users to visualize and interpret.

Important non-functional requirements include scalability and performance– the platform should be able to handle a large number of users at the same time without a noticeable decrease in performance. It must grow to accommodate growing user bases and group collaborations. The application must also be secure, encrypting all sensitive user data for authentication and other functionality. Another non-functional requirement is to be intuitive and accessible; it should be able to be used by a wide range of users with ease and should be mobile-friendly, with a responsive design for handheld devices.

Preliminary Project Schedule

1. Planning and Setup (Weeks 1–3)
    - Define project requirements, finalize the scope, and set up the MERN stack with TypeScript. Design the MongoDB database schema to handle users, budgets, transactions, and shared groups. Research and select

external APIs, such as Plaid for bank integration or Google Charts for visualizations, and configure the development environment, version control, and hosting platform.

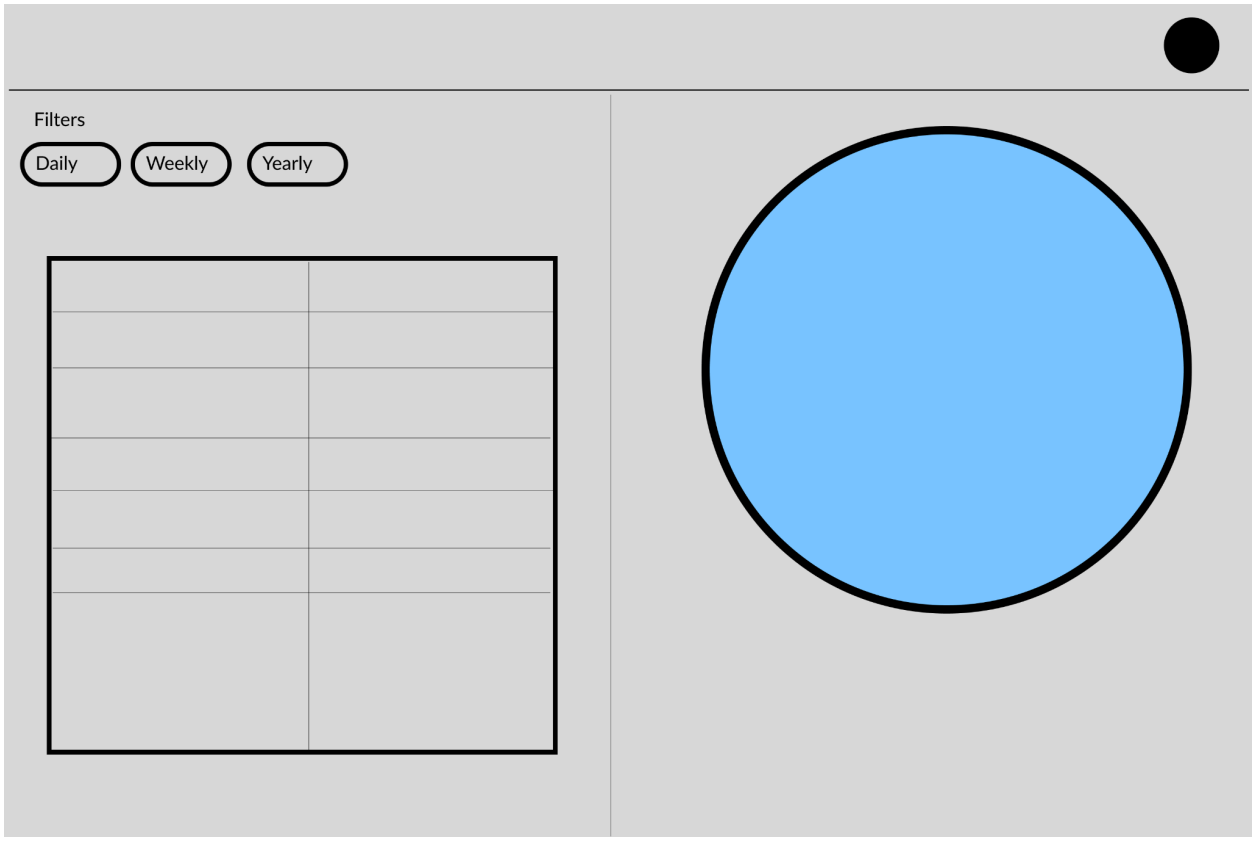
2. Core Development (Weeks 4–9)
    - Develop the backend using Node.js and Express to implement user management, budgeting, transaction tracking, and shared group features. Build APIs for exporting transaction data in PDF/CSV formats. Create the frontend using React to develop a user-friendly interface for login, dashboard, and budget tracking. Integrate Google Charts API to display dynamic charts and begin implementing collaborative features like shared budgets and group goal tracking.
3. Testing, Optimization, and Deployment (Weeks 10–15)
    - Conduct testing to ensure functionality and fix any bugs. Optimize performance for handling large datasets and user groups, and ensure scalability and security by encrypting sensitive data. Polish the UI for responsiveness across devices, add finishing touches to features, and prepare the application for deployment. Deploy the application on a hosting platform. Prepare a project demo, showcase key features, and gather feedback for future improvements.

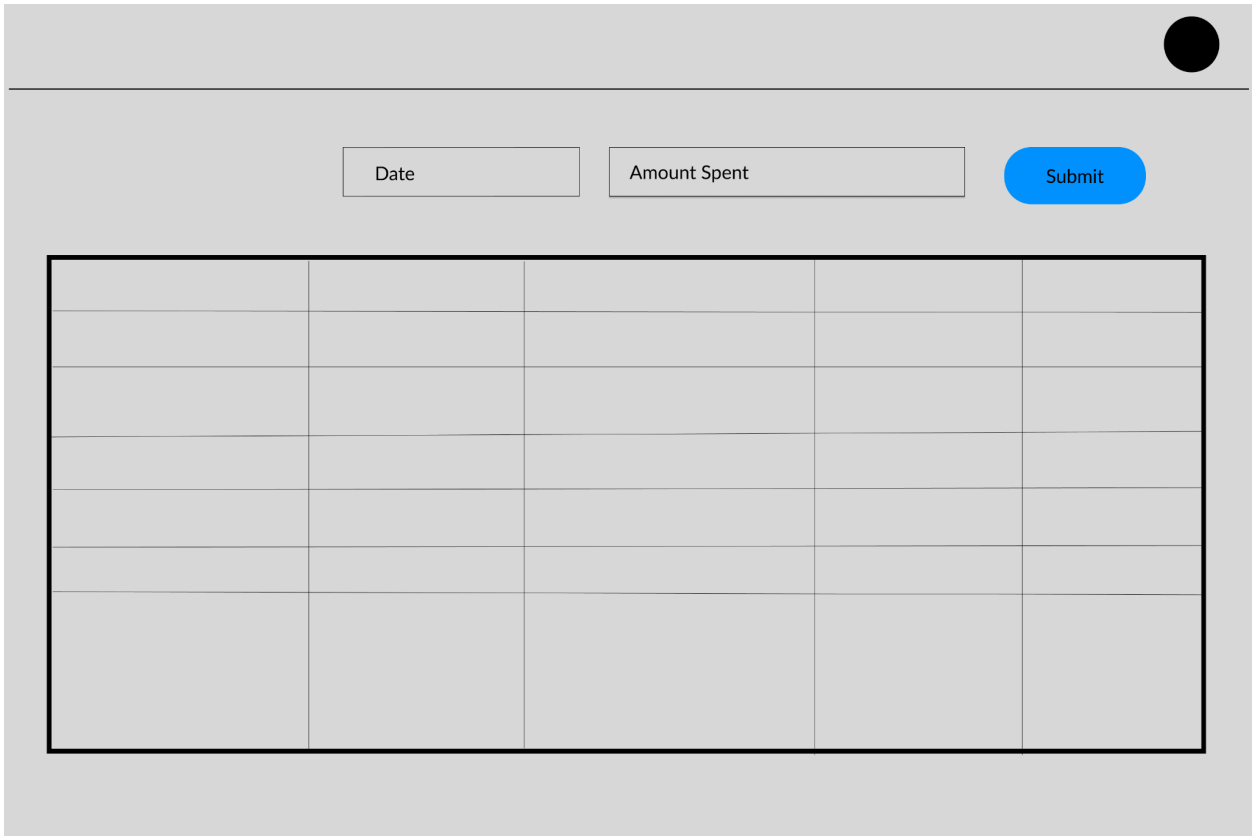
Sitemap

```
├── README.md
├── backend
│   ├── api.js
│   └── server.js
└── frontend
    ├── README.md
    ├── index.html
    ├── package-lock.json
    ├── package.json
    ├── public
    ├── src
    │   ├── App.tsx
    │   ├── components
    │   │   ├── Footer.tsx
    │   │   ├── Graph.tsx
    │   │   ├── Header.tsx
    │   │   ├── SubmitForm.tsx
```

```
│   │   └── Table.tsx
│   └── main.tsx
├── tsconfig.app.json
├── tsconfig.json
└── tsconfig.node.json
```

Wireframes




Date
Amount Spent
Submit