Team 5 Project Proposal

Project Overview

Our expense tracking and budgeting app helps users monitor their spending habits and stay on track with their financial goals. With an intuitive interface, users can manually log purchases or upload receipts for automatic expense tracking.

Many budgeting apps are overly complex, making it difficult for students and those new to budgeting to manage their finances. Our app simplifies the process by focusing on expense tracking without the complications of taxes and investments. By providing accessible and flexible budgeting tools, it empowers users to build better financial habits and take control of their spending.

Project Scope

- MVP (core functionalities/features)
    - User authentication utilizing JSON web tokens
    - Receipt reading OCR through Azure
    - Display budget statistics based on the user's goals
        - Allow user to sort by type, date, quantity, categories
    - Show past history of spending habits through a convenient dashboard
    - Allow users to easily input purchases manually
- Stretch Goals
    - Analyze spending habits and recommend points of improvement
    - Warn user if spending habits show cause for concern
    - Allow users to customize which categories they like
    - Quickly add purchase history or quick notes with a voice memo
    - Integrate with the user's bank account to automatically log purchases

Project Objectives

1. Create a web app using React.
2. Create an easy to use, intuitive, and visually appealing UI.
3. Allow users to log in and store user data securely.

4. Have a dashboard page that allows users to take in their finances at a glance, including recent purchases, a pie chart of expenses, a trendline showing how on track they are to achieving their financial goals, etc.
5. Allow users to log their purchases and store the date, amount, and type into a cloud-based database using MongoDB Atlas.
6. Use an OCR API to allow users to scan receipts to log purchases instead of having to manually input a purchase.
7. Give users a variety of options to view their stats, like limiting to a date, focusing on a certain category, seeing overall statistics, etc.

Specifications

This application will be available on the web, as we have judged our target demographic to be more likely to work on their finances with their other workloads on a computer. While mobile functionality provides its own unique conveniences, we view college students as more comfortable doing expense tracking and budget planning on a laptop or desktop computer, which they use daily for schoolwork and other important tasks.

User Interface (UI) Design

- User Sign Up page: allows a user to create an account with a username and password which they must confirm
- User Login page: allows users to log into their account with their username and password
- Dashboard page: displays a user's financial overview including…
    - a pie chart of expenses categorized by type
    - a chart showing spending trends over time which can be filtered by expense type
    - recent transactions
- Receipt upload / expense tracking page: allows users to upload receipts for automatic expense logging, supports manual expense entry, and features Quick Add – enables users to add a voice memo or quick note to remind them to enter an expense later
- Profile page: displays the user's information and allows the user to make changes to their account

User Interaction Elements

- Buttons for logging in/signing up, uploading receipts, and manually adding expenses
- Interactive charts to help a user better see their financial trends (pie charts, line graphs, etc.)

- Filters to customize data views based on the user's needs

Backend & APIs

- MongoDB database stores user profiles, budgeting plans, and transaction data
- Azure Document Intelligence Receipt API for handling automatic receipt processing
- API for calculating goals, percentages, and trends based on the data stored within the database
- JSON Web Tokens (JWTs) for user authentication and protection of user data (protects pages and API calls from unauthorized users)

Tech Stack

- **Frontend:** React, Figma
- **Backend:** Next.js
- **Database:** MongoDB
- **Cloud & Hosting:** Azure Document Intelligence API for receipts

Hardware Requirements

As our target demographic includes people who are entering financial budgeting, our goal is to make our application as accessible as possible with no need for any high-end, uncommon hardware or technology. All that the user would need is:

- A computer with access to a Wi-Fi network and an installed browser
    - Must have decent processing capabilities in order to use or manage our application
- Proper input devices for the computer, such as a mouse and keyboard
- A screen or monitor to view the application as the user or developer interacts with it

Software Requirements

- User software requirements
    - Any browser that supports Chrome extensions
- Developer software requirements
    - Any IDE that supports React and Node.js development
    - A Figma account with any desired plugins

- An installed version of MongoDB Compass
- Any browser that supports web development

Project Timeline

Phase	Duration	Tasks		
		Front end	**Back end**	**General**
Phase 1	[Feb 10] - [Feb 17]	Finalize which pages need to be designed	Research database, API, & authentication	Project proposal & environment setup
Phase 2	[Feb 17] - [Mar 3]	Finalize Figma design for all pages	Database & API setup	
Phase 3	[Mar 3] - [April 14] *Break Mar 17-21	Implement pages	User authentication, database & API integration	Development and initial implementation
Phase 4	[April 14] - [April 28]	Finalize pages	Near-complete integration with data elements	Testing and integration
Phase 5	[April 28] - [May 2]	Finalize pages	Polished integration with data elements	Final testing, deployment, and presentation

Team Leader Rotation

February 3 - February 21	Gabrielle Kuruvilla
February 24 - March 14	Harper Wood
March 24 - April 11	Viet Vu
April 14 - May 2	Kacie Yee

* March 17-21 is Spring Break, no team leader assigned

Project Team

Role	Team Member	Responsibilities
Frontend Developer	Gabrielle Kuruvilla	UI design and development

Frontend Developer	Kacie Yee	UI design and development
Backend Developer	Viet Vu	Database
Backend Developer	Harper Wood	API & authentication

Links

- **GitHub Repository**
- **Agile Board**
- **Figma**