

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
 - Receipt Reading OCR
 - Display budget statistics
 - Allow user to sort by type, date, quantity
 - Show past history of spending habits
 - Allow users to log 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 with voice memo
 - Integrate with your bank account to automatically log purchases

Project Objectives

1. Create a web app using React
2. 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 database.
6. Use an OCR API to allow user to scan receipts to log purchases instead
7. Give users a variety of options to view their stats, like limiting to a date, focusing on a certain category, etc

Specifications

This application will be available on the web.

User Interface (UI) Design

- User Sign-In Page: allows users to create an account and log in.
- Dashboard: displays a user's financial overview, shows a pie chart of expenses categorized by type, and lists the most recent expenses and income.
- Receipt Upload Page: allows users to scan 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.
- Trends Page: displays a line graph showing month-to-month spending, includes a budget comparison line to track financial goals, allows filtering by categories.

User Interaction Elements

- Buttons for logging in, uploading receipts, and manually adding expenses.
- Interactive charts.
- Filters to customize data views.

Backend & APIs

- MongoDB database will store user profiles, budgeting categories, transaction data through receipts and manual inputs, etc.
- API for handling OCR processing.
- API for calculating goals, percentages, and trends accordingly.

Tech Stack

- **Frontend:** React, Figma

- **Backend:** Node.js
- **Database:** MongoDB
- **Cloud & Hosting:** Azure Document Intelligence API for receipts

Hardware Requirements

- A computer with access to a Wi-Fi network
 - 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
 - Any browser that supports Chrome extensions

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]			Testing and integration
Phase 5	[April 28] - [May 2]			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](#)