Pitch Document : Group 1

Statement of Purpose

This project aims to develop an intuitive and user-friendly expense tracker application that allows individuals to monitor, analyze, and control their daily expenses and budget. This tool aims to promote better financial habits by offering personalized insights, monthly reports, and real-time alerts for overspending, ultimately helping users make informed financial decisions.

Problem statement

Many people struggle to manage their finances effectively, often due to a lack of understanding of their spending patterns and tools to help them budget. As a result, overspending and impulse purchases can lead to financial instability. There is a need for an accessible, easy-to-use expense tracker that goes beyond simple logging and provides visual insights, monthly spending reports, and alerts to notify users when they approach or exceed their budgets. This project addresses the gap by offering a comprehensive solution for personal finance management.

Project objectives

- Expense Tracking: This feature allows users to log their expenses, categorize them, and view all transactions in one place.
- Visualized Insights: Provide visual representations, such as pie charts and bar graphs, to help users understand their spending patterns.
- Monthly Financial Reports: Generate detailed monthly and annual reports that summarize spending trends, budget adherence, and overall financial health.
- Budget Alerts: Set up notifications to alert users when they are close to or exceeding their predefined budget limits, encouraging mindful spending.
- User-Friendly Interface: Design a simple, intuitive interface that allows users of all financial backgrounds to navigate and manage their expenses easily.
- Data Security: Ensure all personal and financial data is securely stored and accessible only to the user.

Scope / Features prioritized according to implementation plan

1. Authentication, Data Security, and Backup
    a. Enable secure user registration and login, by adding a login / sign up page
    b. Implement data encryption to ensure the security of user data
    c. Regularly backup user data to prevent loss and ensure availability
2. Enhanced UI Design
    a. Continuously improve the user interface with a modern and clean design.
    b. Implement an adaptive layout for a seamless user experience across devices.
3. Expense Categorization
    a. Allow users to input expenses and assign them to custom categories (e.g., food, transport, shopping).
    b. Users can create and name their own expense categories, providing flexibility in tracking
4. Real-time Visualization of Expense Summary
    a. Display a real-time dashboard that summarizes the user's expenses, categorized by type.
    b. Offer visual insights with charts to show spending distribution and trends.
5. Budget Alerts and Notifications
    a. Provide users with the ability to set monthly or category-specific budgets.
    b. Send alerts when users approach or exceed their set budget limits, helping them stay within financial goals.
6. Monthly and Weekly Report Generation
    a. Generate downloadable reports that summarize the user's expenses, income, and budget adherence.
    b. Offer both monthly and weekly report options for users to review their financial activity and make adjustments.

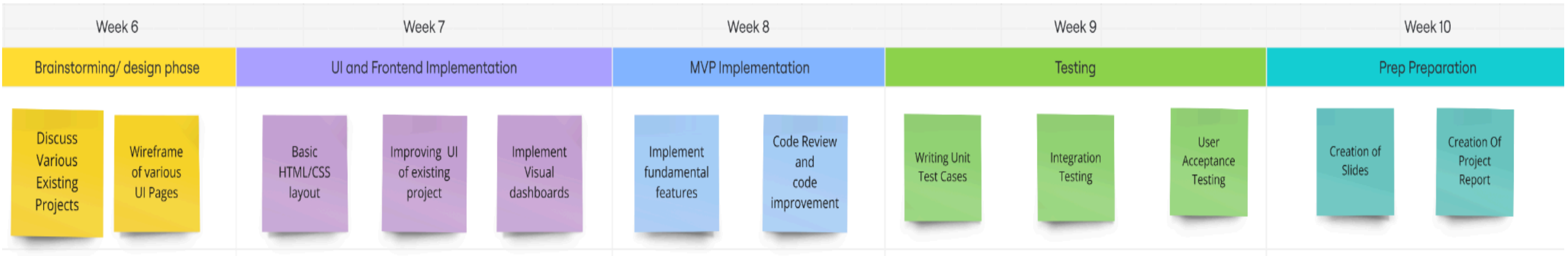
Stakeholders :

Some of the primary stakeholders in an expense tracking app Frequent travelers who need expense reporting, small business owners, freelancers requiring income and individuals who are Budget-conscious

Methodology, Technology :

The expense tracker application will be developed using a modern web stack consisting of HTML, CSS, and vanilla JavaScript, with the development process managed through Agile methodology principles to ensure efficient and iterative delivery.

Timeline :



Week 6	Week 7	Week 8	Week 9	Week 10
Brainstorming/ design phase	UI and Frontend Implementation	MVP Implementation	Testing	Prep Preparation
Discuss Various Existing Projects Wireframe of various UI Pages	Basic HTML/CSS layout Improving UI of existing project Implement Visual dashboards	Implement fundamental features Code Review and code improvement	Writing Unit Test Cases Integration Testing User Acceptance Testing	Creation of Slides Creation Of Project Report

User personas

Users:

Ethan Walker

16 year old boy

Got his first job at Target, first debit/credit card

Avery Smith

22 year old woman

Recent college grad with first full-time job; works in marketing

Jacob Bennett

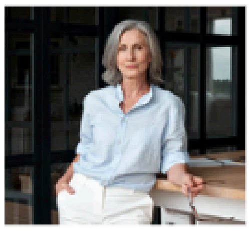
30 year old man

Preparing to save up for an engagement ring; works in software development

Sophie Daniels

33 year old woman

Wanting to organize finances in preparation for kids; works as a civil engineer



Carolyn Mitchell

60 year old woman

Wanting to manage finances to prepare for retirement; works in sales

User Stories:

1. Ethan Walker: I just started working at Target part-time so now I'll have my own income! My parents want me to be organized with my finances so I don't overspend–I want to use this software to track my spending.
2. Avery Smith: I just graduated with my Bachelor of Arts in Marketing! I'll be starting my first ever full-time position at Microsoft next year. This income tracking software will help me keep tabs of how much I'm spending on rent, utilities, food, dinners, etc. each month so I can start contributing more to my savings account.
3. Jacob Bennett: I'm ready to propose to my girlfriend of 3 years! I want to get her the perfect ring so I want to use this income tracker to contribute more money to the purchase and to create "padding" for myself so I can stay on top of my monthly payments for it.
4. Sophie Daniels: My partner and I are ready to have our first child within the next 2 yrs. On average, one child cost ~$25k per year so we need this software to ensure we have enough saved and budgeted for our new baby.
5. Carolyn Mitchell: I'm preparing to retire in the next 5 years. I've contributed to my 401k for my whole career but I want to ensure I'm not overspending and contribute a lot more to my savings the next 5 years for my retirement.

Use Cases:

1. Ethan Walker
   - To track my money earned from my job(variable pay based on hours)
   - To track my spending by category on games, food, and shoes
   - To track how much I contribute to my first savings account
2. Avery Smith
   - To track how much i spend by category on groceries & food, rent & utilities, every month
   - To track ratio of leisure expenses vs living expenses
   - To track ratio of expenditure vs savings
3. Jacob Bennett
   - To track expenses using 30% for ring payments, 30% for padding/savings, and 40% on living and other expenses
   - In the 40% living and other expenses category, to track ratio of leisure expenses vs living expenses
4. Sophie Daniels
   - To track that the couple is saving a minimum of 40k USD every year out of their salary for the next two years, and 30k USD per year after childbirth. This will cover the child costs and education.

5. Carolyn Mitchell
    - To track expenses using 40% towards retirement funds, 20% for other savings, and 40% towards living and other expenses

Risks

1. Change of UI
    a. Tight coupling: The current HTML structure and JavaScript functions are closely intertwined, making significant UI changes challenging without breaking existing functionality.
    b. CSS complexity: The neomorphic styling may complicate UI updates, potentially leading to inconsistencies or visual bugs when introducing new elements.
2. Authentication
    a. Data migration: The current use of localStorage for data persistence will require a significant overhaul to support user-specific data and server-side storage.
    b. Security vulnerabilities: Implementing authentication without proper security measures could expose user financial data to potential breaches.
3. Categorization of Expense
    a. Data structure limitations: The current simple transaction structure may not easily accommodate complex categorization, potentially requiring a complete redesign of the data model.
    b. Performance issues: Adding categories might impact the efficiency of existing functions like `updateValues()` and `addTransactionDOM()`, especially with a large number of transactions.
4. Weekly/Monthly Reports
    a. Calculation accuracy: The existing `updateValues()` function may need significant modifications to support accurate periodic reporting, risking the introduction of calculation errors.
    b. Data aggregation challenges: The current data structure doesn't support easy aggregation for reports, potentially leading to inefficient or inaccurate report generation.
5. Visualization
    a. Integration complexity: Adding a visualization library to the vanilla JavaScript project may introduce compatibility issues or increase the project's complexity significantly.
    b. Performance bottlenecks: Rendering charts and graphs, especially with large datasets, could lead to performance issues in the browser.
6. Alert the User Beyond an Expense Limit
    a. Real-time monitoring challenges: The current structure doesn't support real-time expense tracking, making it difficult to implement timely alerts.
    b. User experience disruption: Implementing alerts without careful consideration could lead to an overly intrusive user experience, potentially driving users away.

General Risks:

1. Scalability issues: The current implementation using vanilla JavaScript may struggle to handle increased complexity, potentially requiring a complete rewrite using a more robust framework.
2. Code maintainability: Adding these features to the existing codebase without proper refactoring could lead to spaghetti code, making future maintenance and updates challenging.
3. Browser compatibility: New features, especially those involving advanced JavaScript or CSS, may introduce cross-browser compatibility issues.
4. Testing complexity: The lack of a formal testing structure in the current project may make it difficult to ensure reliability as new features are added.
5. Feature integration: Ensuring that new features work harmoniously with existing functionality (e.g., the screenshot and download features) could be challenging and time-consuming.

Rabbit Holes

1. Over-engineering:
   There's a risk of adding unnecessary complexity to the application by implementing advanced features that may not be essential for basic budget tracking.
2. Feature creep:
   As you start adding new features, it's easy to keep expanding the scope, potentially leading to an overly complex application that strays from its core purpose.
3. Perfecting the UI:
   Spending excessive time on refining the user interface could detract from developing core functionality.
4. Implementing advanced analytics:
   While data visualization is valuable, diving too deep into complex analytics features might be premature for an unused application.
5. Overcomplicating categorization:
   Creating an overly intricate expense categorization system could make the app difficult to use and maintain.

By being aware of these potential pitfalls and following best practices, you can more effectively build upon the existing budget tracker while avoiding unnecessary complications and ensuring the final product meets user needs.

Research about competitors

1. You Need a Budget

- Additional features:
  - Zero-based budgeting helps users assign every dollar a purpose, reducing impulse spending.
  - Real-time syncing across devices ensures users have up-to-date financial information.
  - Extensive tutorials and workshops support users in building good financial habits.
- Inspiration for Our Project:
  - Implement a zero-based budgeting approach to encourage thoughtful spending.
  - Offer simple, user-friendly guides and resources to make budgeting easier for beginners.

2. Mint

- Additional features:
  - Automatically categorize expenses, saving users time.
  - Track credit scores and provide regular updates.
  - Send bill reminders to prevent late fees.
- Inspiration for Our Project:
  - Add automatic categorization to simplify tracking for users.
  - Include bill reminder notifications to help users avoid missed payments.

3. PocketGuard

- Additional features:
  - “In My Pocket” feature shows disposable income after accounting for bills and goals, helping users manage spending.
  - Track subscriptions, suggesting cancellations for unused services to save money.
  - Allow users to set spending limits in specific categories.
- Inspiration for Our Project:
  - Include a disposable income feature to help users know how much they can safely spend.
  - Add subscription tracking to manage recurring payments effectively.

4. Goodbudget

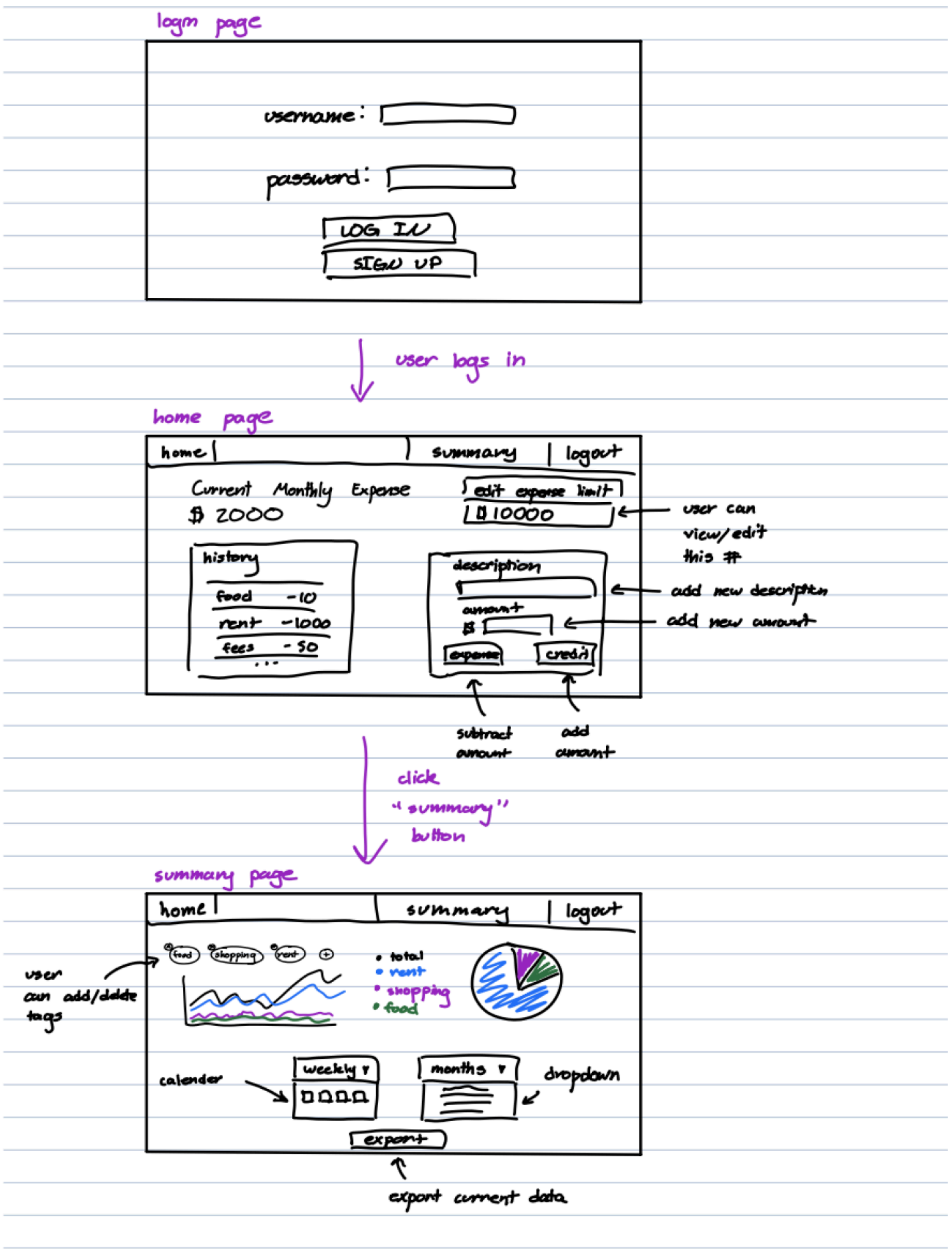
- Additional features:
  - Use an envelope budgeting system to allocate funds to specific categories.
  - Support shared budgeting, making it easier for families or partners to budget together.
  - Offer debt tracking to help users manage and reduce their debt.
- Inspiration for Our Project:
  - Implement an envelope system to encourage proactive money allocation.
  - Add shared budgeting features for families or groups to manage finances collaboratively.

5. Simplifi by Quicken

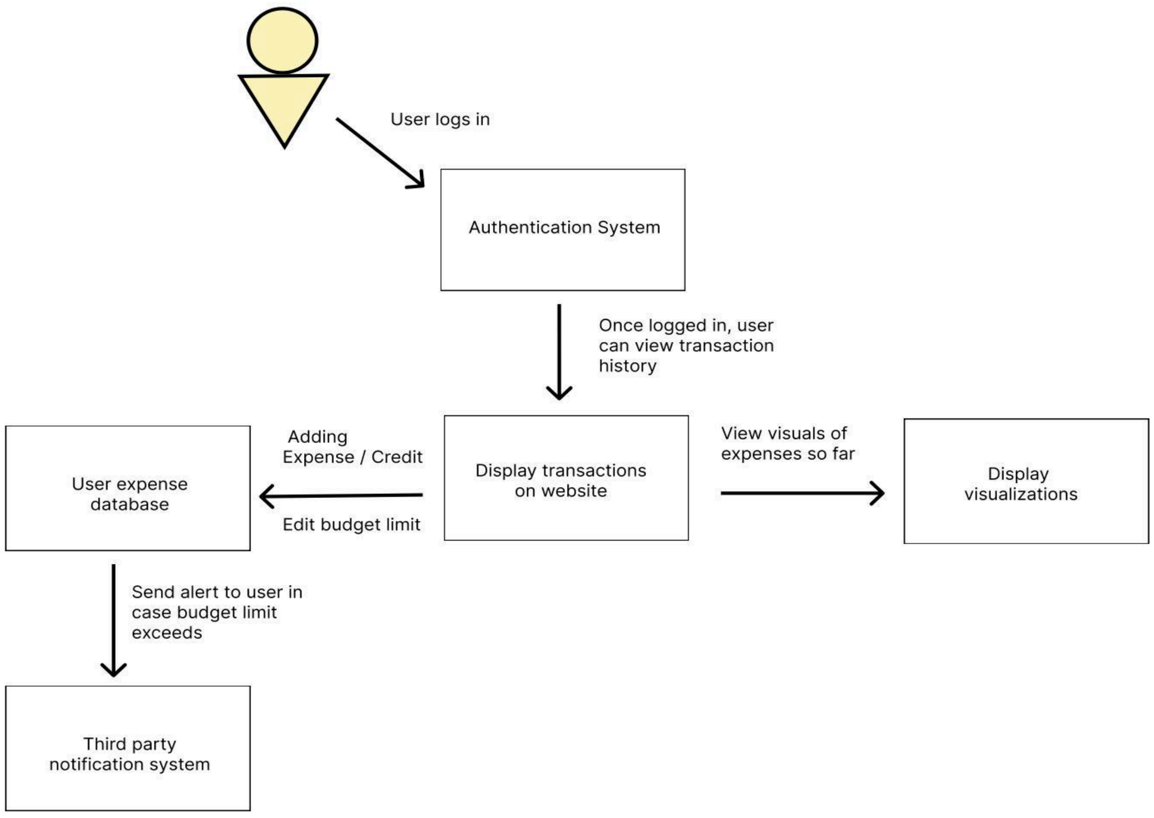
- Additional features:
  - Customizable watchlists allow users to monitor specific spending habits.
  - Projected cash flow helps users plan for future expenses.
  - Investment tracking provides a holistic view of financial health.
- Inspiration for Our Project:
  - Introduce custom watchlists so users can focus on specific spending areas.
  - Develop a cash flow forecast tool to help users anticipate upcoming financial needs.

Diagrams:

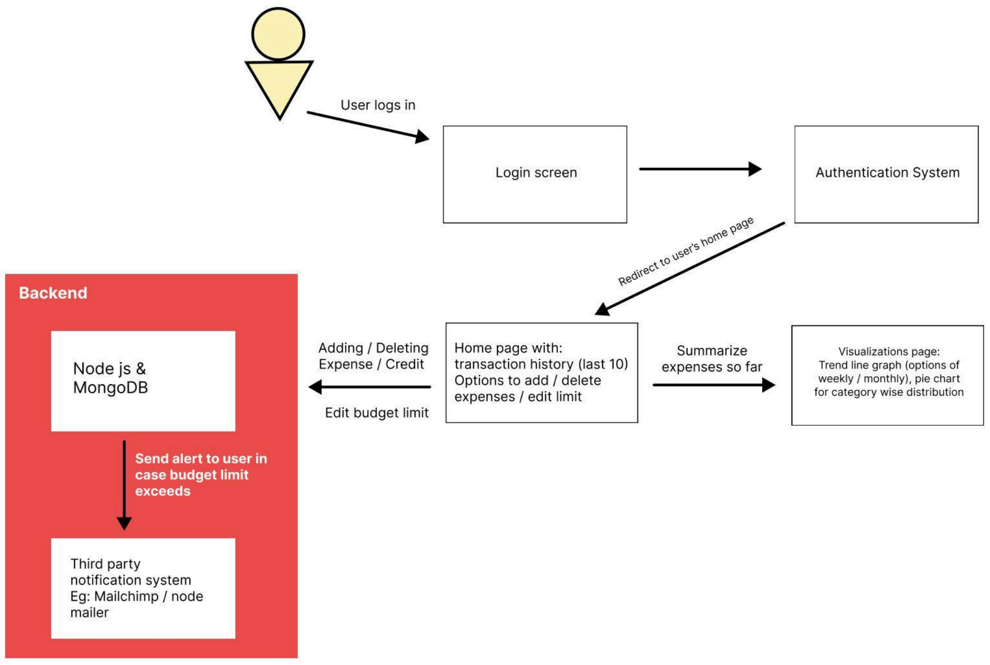
1. Fat marker sketch:



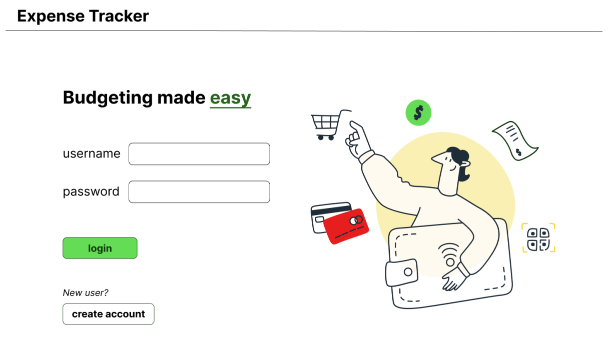
2. System component Diagram


User logs in
Authentication System
Once logged in, user can view transaction history
Display transactions on website
Adding Expense / Credit
Edit budget limit
User expense database
View visuals of expenses so far
Display visualizations
Send alert to user in case budget limit exceeds
Third party notification system

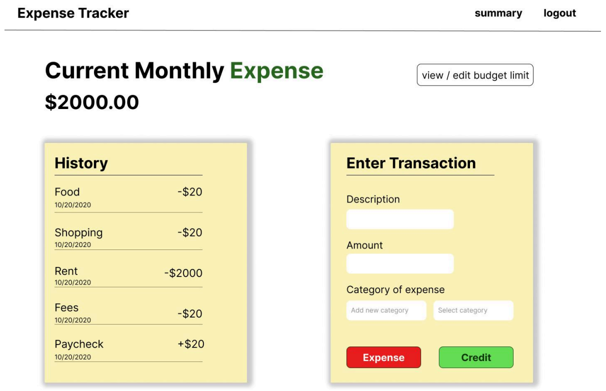
3. Container Diagram


User logs in
Login screen
Authentication System
Redirect to user's home page
Home page with: transaction history (last 10) Options to add / delete expenses / edit limit
Summarize expenses so far
Visualizations page: Trend line graph (options of weekly / monthly), pie chart for category wise distribution
Adding / Deleting Expense / Credit
Edit budget limit
Backend
Node js & MongoDB
Send alert to user in case budget limit exceeds
Third party notification system Eg: Mailchimp / node mailer

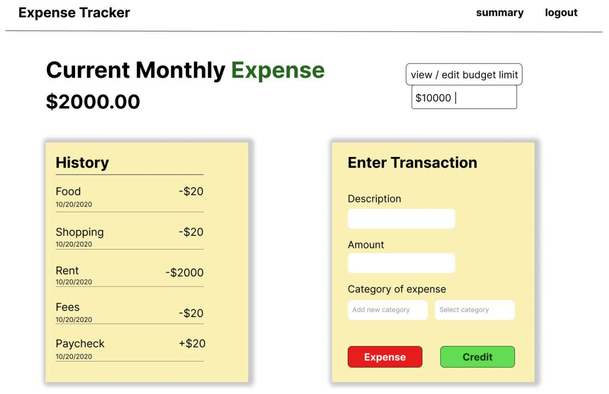
4. Wireframes:



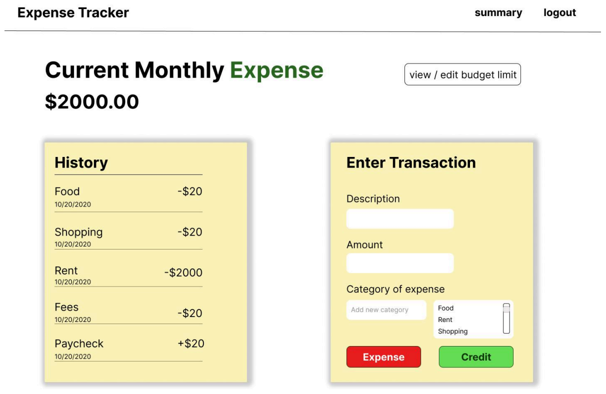
Frame 1: Login Page



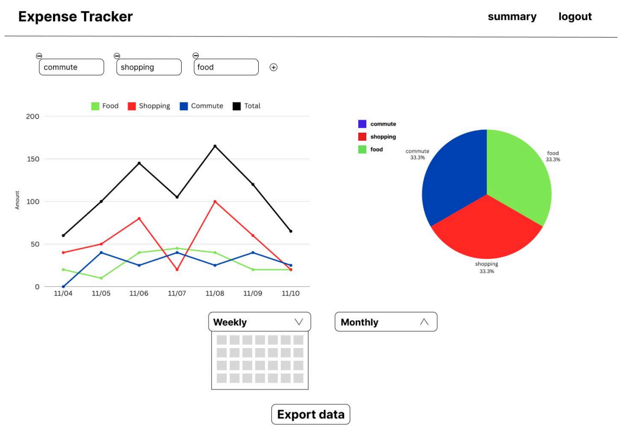
Frame 2: Home Page



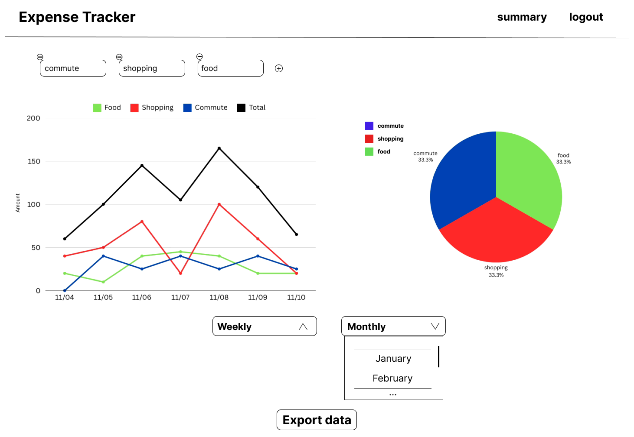
Frame 2.1: Home Page - Edit budget limit



Frame 2.2: Home page - Search category



Frame 3: Visualizations Page



Frame 3.1: Visualizations Page - Monthly option

Existing Project link: http://expense_tracker_midha.surge.sh/
Existing codebase: https://github.com/MidhaTahir/Expense_Tracker_VanillaJS/tree/master