

Project 1: Recording the Journey

Mikhail Nikolaenko

Theme:

Financial tracking for daily, monthly, and yearly spending with breakdowns of spending habits and patterns.

Design:

Interviews:

1a. What do you hope to learn from these interviews

I hope to learn various things from the interviews I will conduct. The main “goal” so to speak, is to discover gaps and/or features that adults believe are missing from current financial tracking applications. I know many banking institutions that have apps already have good breakdowns of spending activity, however, I know many of them are typically for specific banks and generally are varied across applications. From my personal experience having both debit and credit cards from different banks, I know that it’s a bit difficult to track everything across all of these cards. So what I hope to get from the interviews, is information what people like/dislike about current implementations, and how I can adapt these needs into a one-and-all application / website. I will also ask about designs that they like, and how I can adapt current designs into my own implementation.

1b. What questions did you ask?

1. Do you track your financial activity in any way? If so, how?
2. Do you currently use any sort of banking apps, if so which ones?
3. Are there visualizations and/or features you can think of that would be of value to you when tracking finances?
4. What challenges do you face when trying to see a full picture of your finances (across banks, cards, bills, etc.)?
5. Are there any specific moments where your current finance tools fail you or feel incomplete?
6. Would you find value in a single app that brings together multiple accounts (debit, credit, loans, investments)? Why or why not?
7. Can you describe a finance-related app or website that you found visually appealing or easy to use? What stood out about its design?
8. Have you ever stopped using a finance app because it was too confusing, slow, or overwhelming? What caused that decision?

9. How do you prefer to view your financial data: graphs, timelines, categories, summaries? Why?
10. Would you find value in a single app that brings together multiple accounts (debit, credit, loans, investments)? Why or why not?

1c. What did your interview participants tell you? What did you learn from them?

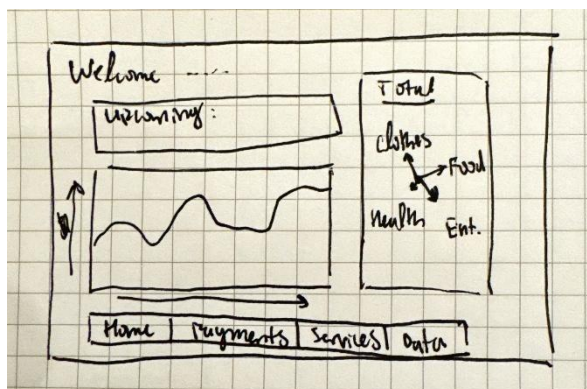
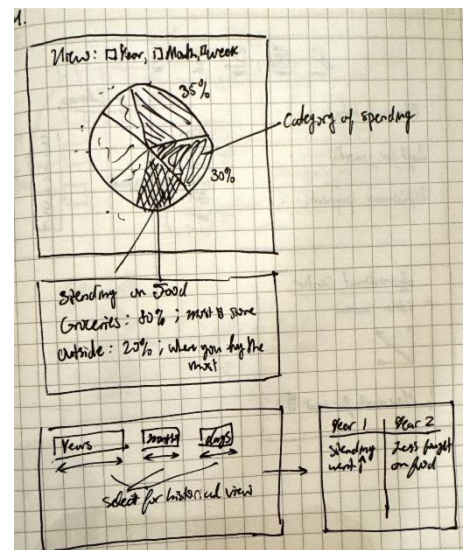
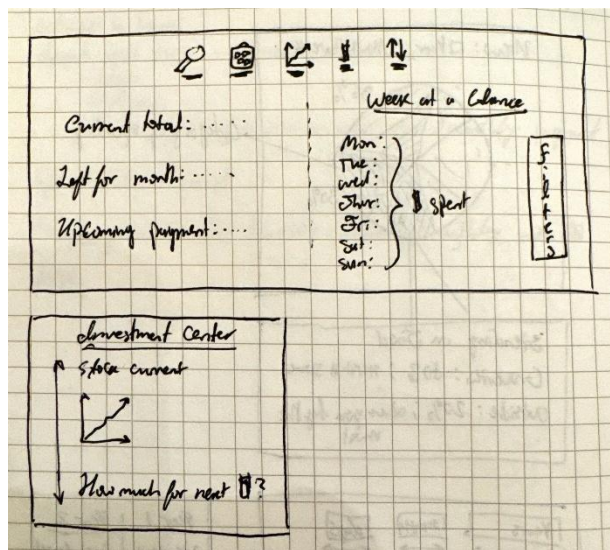
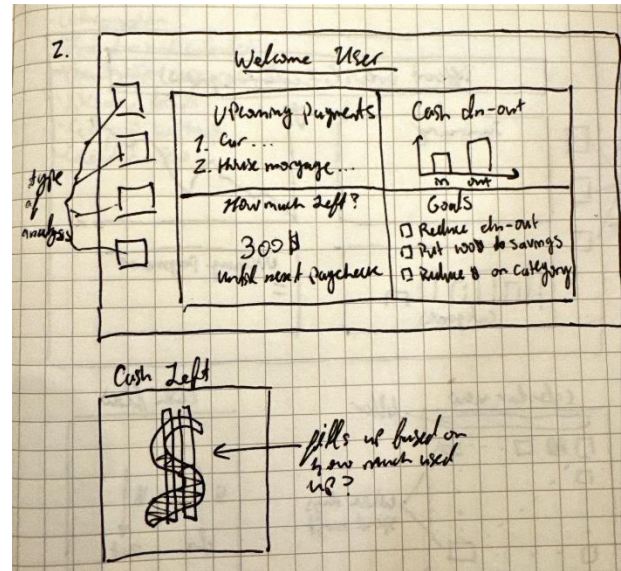
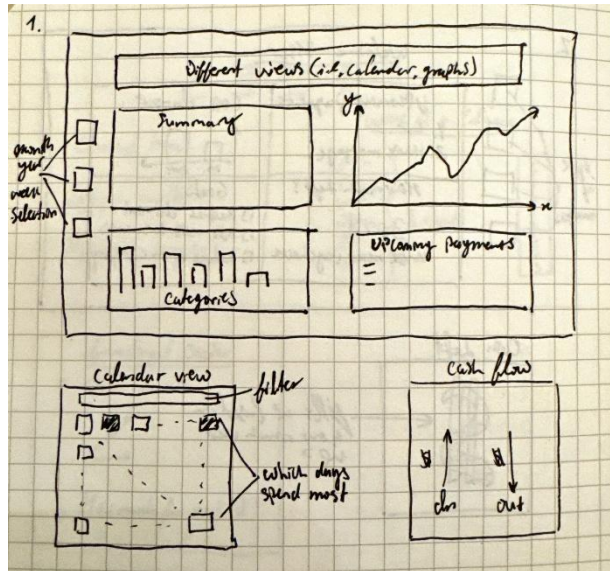
From my interviews, I learned that both participants struggle with keeping track of finances because their money is spread across different banks, credit cards, and bills. They both said current apps don't really give them a clear picture of what they can actually spend after upcoming payments. The student I interviewed wanted more detailed visualizations like cash flow graphs, spending trends, and breakdowns by category, while the older participant preferred simple summaries, pie charts, and a calendar for bills and paydays. Even though their needs were a little different, both agreed they'd find a lot of value in one app that pulls everything together in one place.

For the UI design, this means the app should focus on being clean and easy to use, while also giving options to dive deeper into the data if the user wants. A good approach would be to start with a simple dashboard that shows an overview, like total spending, upcoming bills, and "money left to spend", and then allow users to click in to see more detailed graphs and trends. Both participants mentioned liking designs from apps like Robinhood, Apple Card, and Capital One, so the style should aim to be modern, uncluttered, and visually clear.

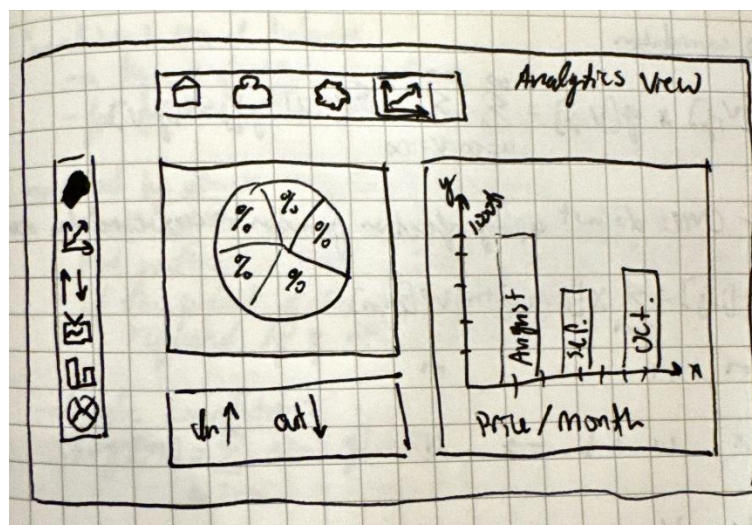
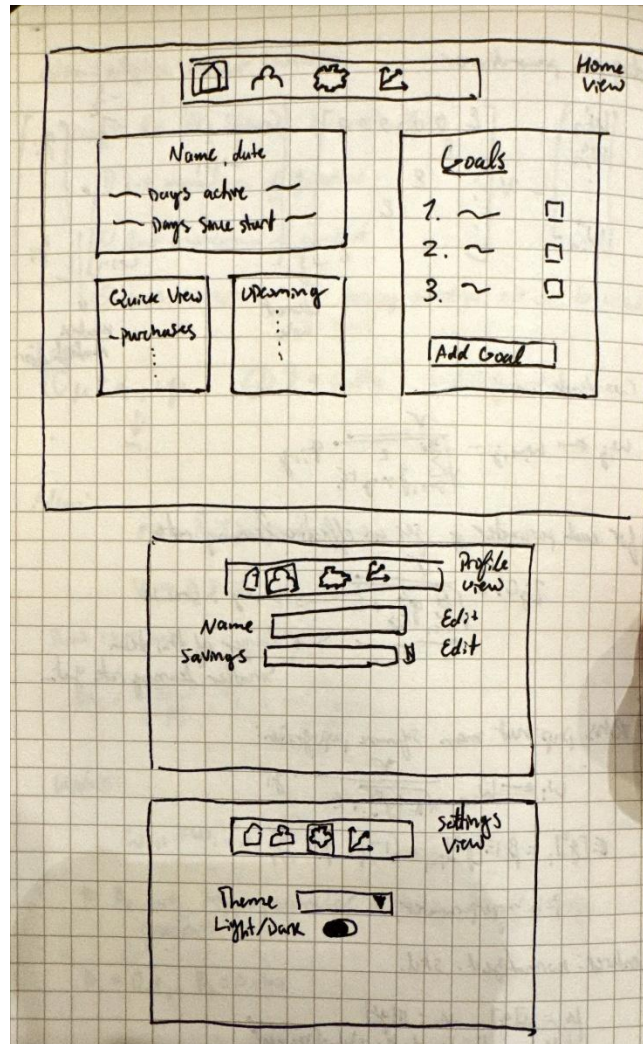
Design Requirements:

1. A way to navigate between main sections (home, profile, settings, data view).
2. Basic forms for adding and editing purchases, recurring payments, and paychecks.
3. A place to view financial data visually (pie chart, trend line, in-out cash, heatmap, plots, etc.).
4. Ability to set and track goals which motivate user.
5. Support for light/dark mode and basic theme options.
6. Save user data so it stays after refreshing the page.

Sketching:



Prototype Sketch:



Feedback on prototype sketch:

- Clear overall layout with navigation separated from content.
- Card-style looks interesting and easy to navigate, keep for consistency and organization.
- Spacing could be improved, make things more spaced out / separated.
- Little clickable icons would be nice with images representing the page.
- Data visualization would be huge plus, good to have on separate tab away from everything else for when you want to “dig in”. Might be overkill so include just main ones like category split.
- Overall, the design feels good and functional.
- Don’t go overboard on main screen, keep it simple yet informative.

The feedback on my prototype sketches was mostly positive, with some good ideas for tweaks. People liked the overall layout, especially having navigation separate from the content, and the card style came across as clean and easy to use. One thing that stood out was spacing, the suggestion was to spread things out more so it doesn’t feel too packed. Another idea was to add little clickable icons with images to make the pages easier to recognize. For data visualization, the advice was to stick with the basics, like a category split chart, and keep them on a separate tab so the main screen doesn’t get too busy. Overall, the design came across as solid and functional, with the reminder to keep the main screen simple but still useful.

Mock User Profile:

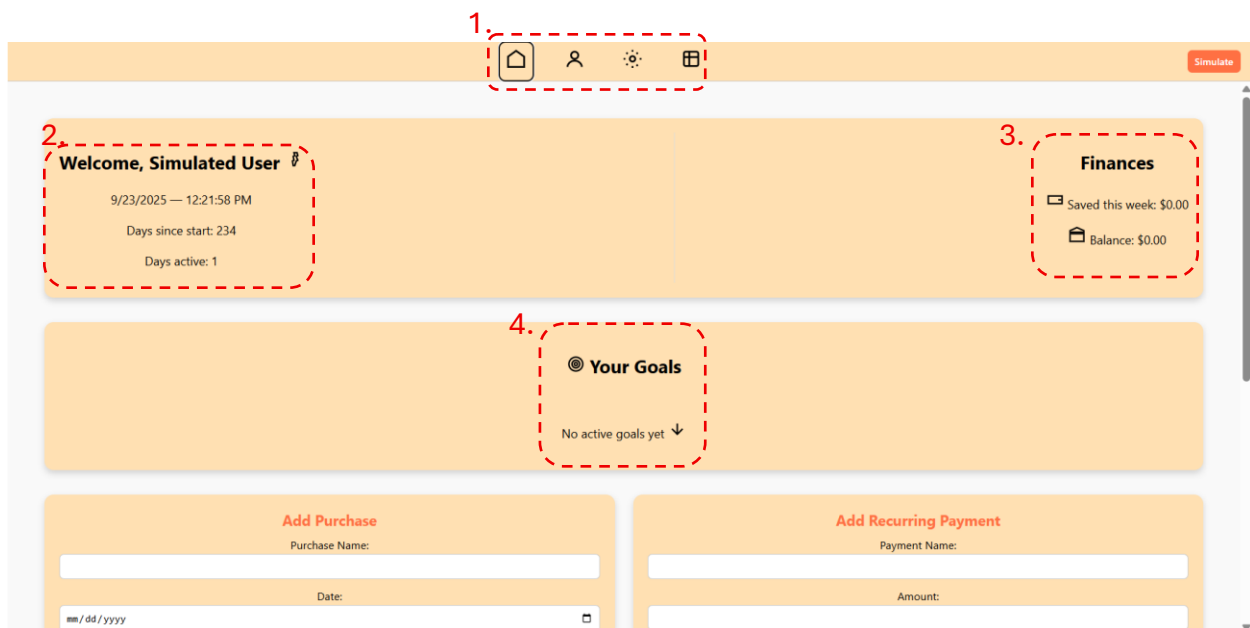
- Age: 21
- Status: College student with a part-time job
- Income: Bi-weekly paychecks
- Expenses: Regular bills (rent, phone plan, subscriptions) and one-time purchases (food, clothes, entertainment)
- Goals: Keep weekly food costs under budget and save for larger purchases

This user would use the application to record purchases, recurring payments, and paychecks. They would rely on the heatmap and category breakdowns to quickly see spending patterns and use the goals feature to track progress toward specific limits or savings targets. The simple layout makes it easier for them to stay organized without needing a more complex budgeting tool.

Interface Explanation:

Main features:

1. **Tab switch** – top tab that allows users to navigate between different pages of the UI. This includes the home page, profile page, settings, and data center page.
2. **General information** – located on left side of hero card, displays name of user, current time & date, as well as days since start using app and days active.
3. **Finances quick view** – displays money saved (modified via finishing goals) as well as current balance (modified in profile page).
4. **Goal view** – displays current goals, these can be set in the lower portion of the home page.



Add information / goals:

1. **Add purchase** – allow users to input a recent purchase they made. They can set the name, the date of purchase, price, and select from a series of categories.
2. **Add recurring payment** – allow users to input recurring payments they have. They can input payment name, amount, category, and billing date.
3. **Add paycheck** – allow user to input their paycheck. They can input the name of the paycheck, the amount, and the day on which they receive the paycheck.

4. **Set a goal** – allow users to set a goal they may have for their finances. They can select from a few options, the amount associated with the option, and a timeframe in which they want to achieve the goal.

1.

Add Purchase

Purchase Name:

Date:

mm / dd / yyyy

Price:

Category:

Food

Save

2.

Add Recurring Payment

Payment Name:

Amount:

Category:

Subscription

Billing Date:

mm / dd / yyyy

Save

3.

Add Paycheck

Paycheck Name:

Amount:

Day:

mm / dd / yyyy

Save

4.

Set a Goal

Goal Type:

Limit Spending

Amount (\$):

Timeframe:

This Week

Save Goal

1.

Food

Clothing

Transportation

Entertainment

Utilities

Healthcare

Education

Gifts

Electronics

Travel

Other

2.

Subscription

Car Loan

Mortgage

Insurance

Utilities

Phone Plan

Other

4.

Limit Spending

Save At Least

Limit Category

This Week

This Month

© Your Goals

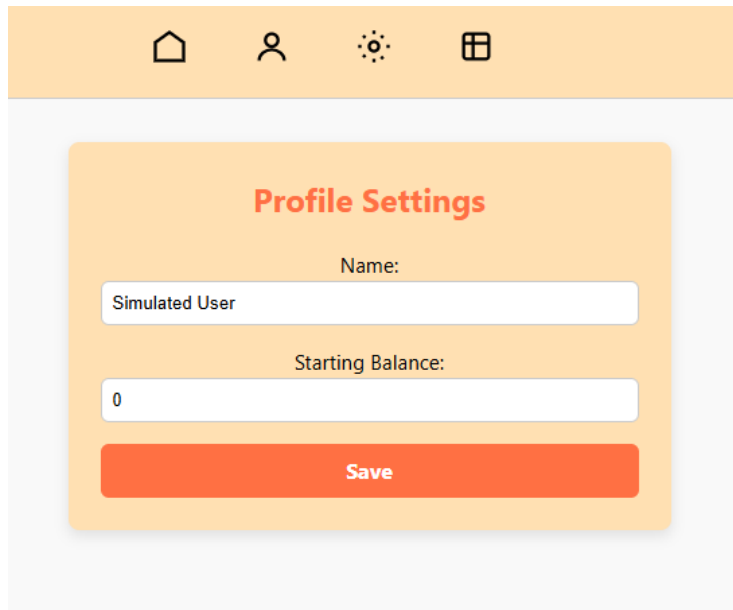
Save at least \$100 this week

4.

✓ Success

Profile page:

1. **Profile interface** – allow users to change their profile name and starting balance.



The image shows a mobile application interface for a 'Profile Settings' page. At the top, there is an orange navigation bar with four icons: a house (home), a person (profile), a gear (settings), and a grid (tabs). Below the navigation bar, the main content area is light gray. In the center, there is an orange rounded rectangle containing the title 'Profile Settings' in bold red text. Below the title, there are two input fields. The first is labeled 'Name:' and contains the text 'Simulated User'. The second is labeled 'Starting Balance:' and contains the number '0'. At the bottom of the orange rectangle is a red 'Save' button.

Profile Settings

Name:

Simulated User

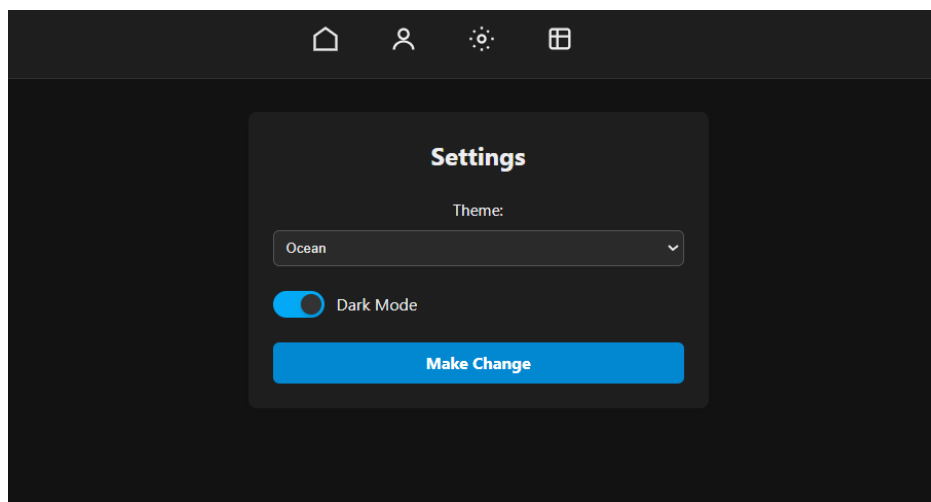
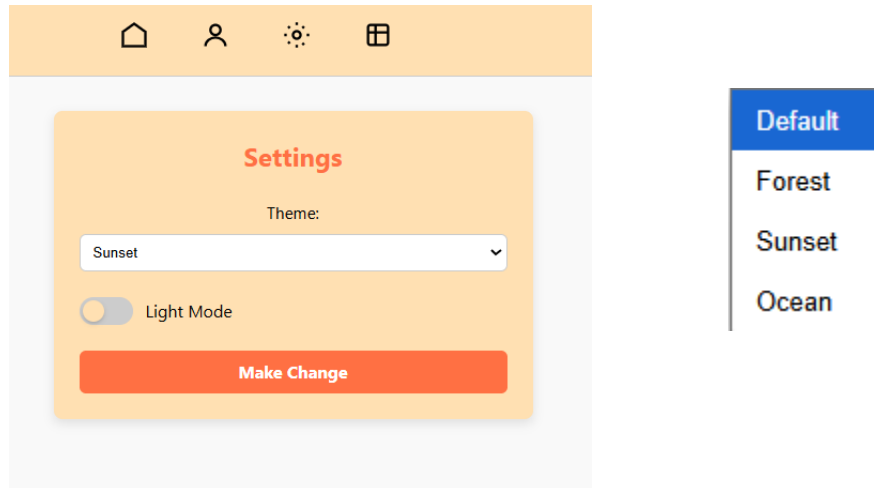
Starting Balance:

0

Save

Settings page:

1. **Settings interface** – allow users to select various themes from a dropdown menu and switch between light and dark mode.



Dark + different theme example

Data center page:

Main view:

1. **Spending heatmap** – heatmap of the year, divided into 365 days, scaled based on lowest and highest purchases/payments made over the course of the year.
2. **Recent purchases view** – shows most recent entries for purchases, with name, date, and amount. Only shows 10 most recent entries but can be expanded to see all via “See All” button.
3. **Switch tab (left side)** – allows users to switch between data center views, those being Main, and Edit Entries.



Transactions					
Item 88	entertainment	9/21/2025	one-time	\$13.69	
Item 91	food	9/20/2025	one-time	\$53.01	
Item 96	entertainment	9/20/2025	one-time	\$40.84	
Item 6	transportation	9/20/2025	one-time	\$14.38	
Item 64	healthcare	9/19/2025	one-time	\$40.77	
Item 64	clothing	9/19/2025	one-time	\$48.78	

2. [See All](#)

Edit Entries view:

1. **Expenses view** – main section which shows users all their entries, including purchases, recurring payments, and paychecks.
2. **Edit / Delete buttons** – allows users to delete an entry or optionally edit any parameter of a given entry.

Edit Entries

One-Time Purchases

Item	Category	Date	Type	Amount	Edit	Delete
Item 74	transportation	2/1/2025	one-time	\$30.07		
Item 92	transportation	2/1/2025	one-time	\$54.71		
Item 23	healthcare	2/1/2025	one-time	\$32.66		
Item 3	transportation	2/2/2025	one-time	\$12.13		
Item 92	entertainment	2/3/2025	one-time	\$32.75		
Item 36	entertainment	2/3/2025	one-time	\$8.93		
Item 13	food	2/4/2025	one-time	\$22.88		
Item 22	healthcare	2/5/2025	one-time	\$33.00		
Item 32	entertainment	2/5/2025	one-time	\$48.52		
Item 27	clothing	2/6/2025	one-time	\$54.24		

Recurring Payments

Item	Category	Date	Type	Amount	Edit	Delete
Netflix	subscription	2/4/2025	recurring	\$15.99		
Car Loan	car-loan	2/9/2025	recurring	\$350		

Paychecks

Item	Date	Type	Amount	Edit	Delete
Monthly Salary	2/14/2025	paycheck	\$3000		

Edit purchase

Name:

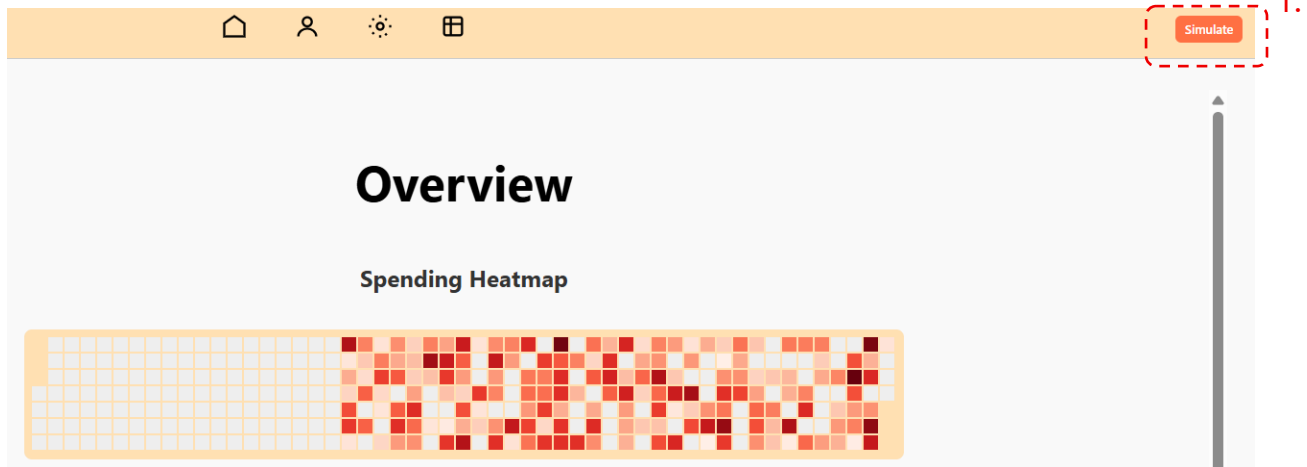
Category:

Amount:

Date:

For testing purposes:

1. **Simulate button** – when pressed, entries random entries over the span of 5-6 months including purchases, recurring payments, and paychecks.



Implementation:

The application is organized so that each feature has its own section of code. Home, profile, settings, editing entries, and data views are handled in **separate** components, which makes it easier to update one part of the app without breaking another. A **central store** is used to keep track of the user's information such as purchases, recurring payments, paychecks, and goals. This store also saves everything to *localStorage*, so the data stays even after refreshing or closing the browser. **Utility functions** are used for tasks like adding, updating, or deleting entries. By putting this logic in one place, the code stays more **consistent** and **reusable** across different parts of the app.

For visualization, I included a **spending heatmap** that shows daily totals over time. This part uses the *D3 library* to draw a **calendar-style** chart with color intensity representing spending levels. It helps turn raw numbers into something easier to understand at a glance. The rest of the interface uses smaller reusable building blocks, like **cards for layout** and **icons for buttons**, which keeps the design simple but consistent.

Overall, the structure **separates** the interface, the data management, and the visualization parts. This makes it easier to maintain, since each piece has a clear role. It also leaves room for **future improvements**, such as adding more types of charts or expanding the kinds of financial data that can be tracked.

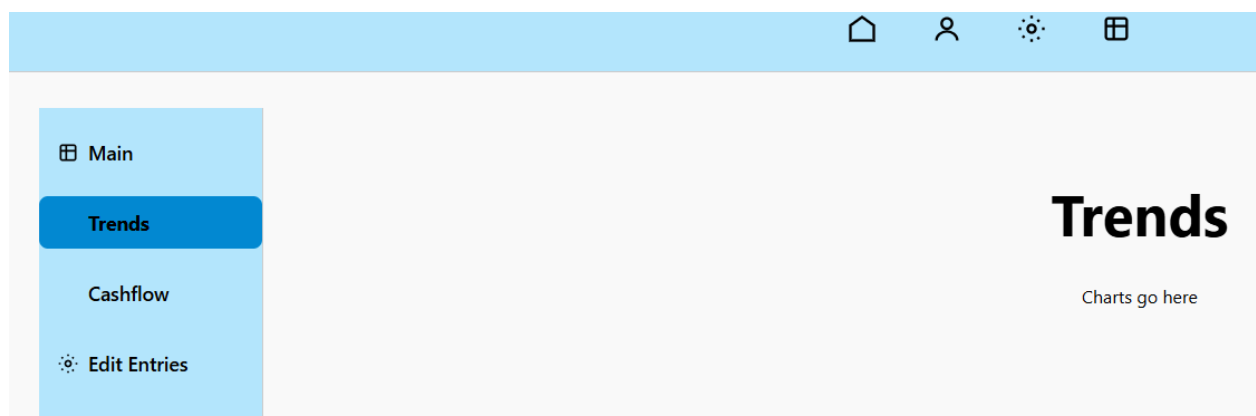
AI Usage:

I used AI mainly to help with smaller but time-consuming parts of the project. For example, it was useful in generating lists of categories and options (like purchase categories or recurring payment types), which saved me from having to think them all up manually. I also used it to help design the simulation logic that creates random purchases and paychecks. This made testing easier, since I could quickly generate a large amount of realistic data without having to enter everything by hand.

The strength of using AI in this way is that it sped up development and let me focus on the main features of the app instead of setup work. It was also helpful for brainstorming ideas or double-checking that my approach to handling data was reasonable. The limitation is that AI-generated code or suggestions still needed to be reviewed and adjusted to actually fit my project. It wasn't something I could just drop in and use without testing or rewriting, so the responsibility for making it work correctly still fell on me.

Future:

For future work, I would like to add more features and visualizations to the application. My original idea included an in/out cashflow visualization, where income and expenses would be shown together over time. At the moment, the only visualization is the spending heatmap, which is helpful but not enough to fully capture financial activity. Adding bar charts, line graphs, or cashflow diagrams would make the data easier to interpret. During development, I had already incorporated the code to handle both trends and cashflow pages as seen in the screenshot; however, due to timeline constraints, I never incorporated the actual diagrams for these pages, so I had to omit these pages in the end during submission. However, I would like to reuse the D3 library to incorporate various statistic visualizations in the future.



Another area I would improve is the goals feature. Right now, goals exist separately from purchases and recurring payments. In the future, I want goals to integrate directly with those entries. For example, a spending limit goal could automatically update as new purchases are

added, or a savings goal could track progress against paychecks. This would make goals more meaningful and interactive, instead of being just a list.

Overall, there are many directions to expand the project: more kinds of charts, better integration between features, and possibly adding alerts or notifications when spending patterns cross certain thresholds. These improvements would make the application completer and more useful for everyday financial tracking.

Video Demonstration:

<https://youtu.be/igYdxig5E6Y>

Link to GitHub source code:

<https://github.com/MikhailNikolaenko/MoneyTrackUI>

Link to public hosted website:

<https://mikhailnikolaenko.github.io/MoneyTrackUI-GithubPages/>