Pull Requests

Gabriel Shows integration between front and backend
Ben Shows clear separation of frontend and backend
Kiarash Shows accessibility implementations and reusable components
Jennifer Shows Liskov Substitution and Dependency Inversion
Yoyo Shows Single Responsibility and Interface Segregation
Chong Shows Open-Closed Principle and Dependency Inversion

Test Coverage Video

- Backend: See supplementals/backend-test-running
- Frontend: See supplementals/Screen Recording 2024-12-03 at 10.56.55 PM.mov

Deployed Website

- Full Website: https://rattm.vercel.app/
- Backend: https://rattm-f300025e7172.herokuapp.com/[ENDPOINT]

Major Design Decisions / Group Abnormality

Front End

Architecture & Tech Stack

1. We chose to build a website over a mobile app because it is simpler for realizing a Minimum Viable Product (MVP). Websites require a single codebase and are accessible across all devices, whereas mobile apps require separate codebases for Android and iOS, increasing complexity and development time.
2. Next.js was chosen as the frontend framework for its robust features, including server-side rendering (SSR) and static site generation (SSG), which improve SEO and performance.
    a. We opted for TypeScript over JavaScript to ensure type safety, making our code more predictable and reducing runtime errors. TypeScript integrates seamlessly with Next.js, enhancing our development experience.
3. Tailwind CSS was used for styling because it promotes a utility-first approach, eliminating the need for separate CSS files and enabling faster, more consistent UI

development directly in HTML. Also, we based several of our React components on this Next.js dashboard template.

4. Our frontend is hosted on Vercel, which provides seamless integration with GitHub, enabling an easy-to-implement CI/CD pipeline and automatic deployments for rapid iteration.

UX/UI

We decided to first map out our design in figma to help us both visualize the website. We then chose a dashboard to be the first thing shown when entering the website because we wanted to mimic our app to the likes of a credit score visualizer. This is to allow the users to quickly see their environmental impact over a period of time, and give them helpful insights in one place. If they wish to explore more details about their shopping habits, they can explore the "transaction" and "map" pages.

Back End

Architecture

1. We have ensured that each file and class we have created only handle one functionality. For example, the URL.py classes will just handle the URL calls that the frontend is making to the backend.
2. We decided to follow the clean architecture principles in our project, implementing classes in a way that introduces no dependencies between higher level classes and lower level classes (Dependency Inversion).
    a. We have created abstract classes for the interactions between the external drivers layer and the business enterprise layer. Specifically, we have created the "abstract_data_access" class that allows us to create new classes in python easily if we wanted to switch out our current database for a new database.
    b. We have created abstract classes (in our case URL endpoints) for the front end to be able to call the backend business calculations.
3. We have used Heroku to host our backend because it integrates seamlessly with git and has a very easy to implement CI/CD pipeline.

Tech Stack Decision

1. We have decided to use Firebase as our database because it can simplify the user authentication process and it is also a real time database, which can mimic the inflow of data in a real world context, such as when Cash App is running our program with their database that is constantly changing.

2. We have decided to use Python because that is a language everyone can work with, and we chose Django because Python works well with it.

Workflow - Agile

We chose to have one repository for both frontend and backend to have easy integration with both Vercel and Heroku. We have also decided to use the Kanban board on github to split up the tasks and keep track of our progress.

Group Abnormality

- We had a separate backend repository for a period of time: https://github.com/jnnchi/rattm-backend