REHNUMA

By Naveen Zaidi, Mannan, Sarah Faisal, Sabahat Zehra, Aina Shakeel

Software Engineering | Final Demo

WHAT IS REHNUMA?

- "Rehnuma" means guide — and that’s exactly what our web app does for grocery shoppers.
- It’s a smart platform that revolutionizes grocery shopping by:
  - Minimizing total grocery expenses
  - Creating intelligent shopping routes across multiple supermarkets
  - Factoring in both product prices and transportation costs

OBJECTIVES OF REHNUMA

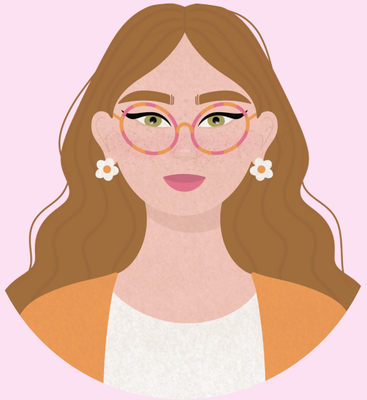
- Provide cost-efficient grocery shopping by analyzing price variations across stores
- Optimize purchases with transportation/fuel costs in mind
- Simple and user-friendly UI for inputting grocery lists
- Regular updates on supermarket inventories and prices

Mannan

- Home page front end
- History page front end
- Optimization algorithm
- history backend.
- Geolocation tracking

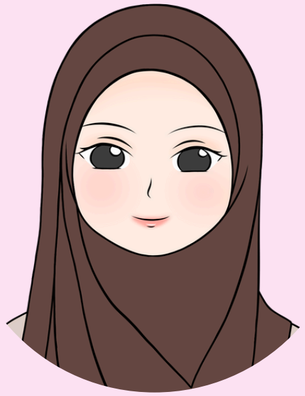
Aina

- Output page front end and backend
- Handled admin functionalities by implementing Python data scrapper
- created super markets schemas



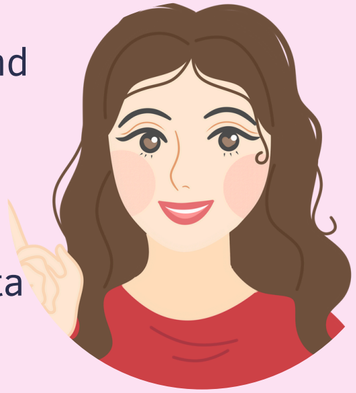
Naveen

- Signup page front end and backend
- Database connection with signup page
- Api calls on user information



Sabahat

- User profile front end and backend
- GeneratePlan front end
- Reset and Forgot password implementaiton using otp



Sarah

- Login page front end and backend
- Database setup and connection
- User schemas design and implementation
- Api calls on user information

WORK DONE TILL NOW

- Completed front end (sprint 1)
- Completed backend (sprint 2)
- Completed database population (sprint 2)
- Completed integration of all components (sprint 2)
- End user testing is still in progress. (sprint 2)

WORKING RUN OF OUR APP

Video Link

CHALLENGES

JWT Authentication

Implementing login/signup using JWT was tough since it was our first time working with it.

User Input Handling

Dealing with typos or missing products in the database made input matching a major challenge

Storing Multiple Plans

Designing a schema to save multiple optimized plans per user was complex.

Displaying Optimized Output

Showing the results from the optimization algorithm clearly was difficult due to many variables.

Geolocation Issues

Geolocation didn’t work smoothly on localhost due to security restrictions; caused major delays.

Map API Limitations

Google Maps API was paid, so we had to switch to Geoapify after exploring multiple free options.

Frontend Disruptions

Backend updates caused major frontend breaks, requiring significant rework and coordination.