

Marvelous Meal Planner

Weekly Meal Planning



Meet Our Team!



Angelo

Frontend, Backend,
Database



Baopu

Frontend, Backend,
Project Manager



Daniel

Frontend, Backend



Lakshmi

Frontend, Backend,
Testing



Agalya

Frontend, Testing



Lynn

Frontend, UI/UX

Motivations



Many find cooking mundane, this affects how quickly they use up their food in cooking.

50% of households in New Zealand report food waste due to expired ingredients

Prices of common groceries, like eggs, have risen by over 50% from 2023 to 2024

Existing Solutions:



Paprika

- Planner + inventory management
- Cluttered UI
- Recipes from online sources
- Cannot browse grocery products

Mealime

- Only for mobile
- Clean, modern UI
- Recipes written by staff
- Also cannot browse grocery products

Grocer

- Dedicated grocery product browsing app
- No cooking assistance capabilities

Market Gaps

- Lacks generative AI
- Fragmented solutions
- Some of them have lacklustre UI

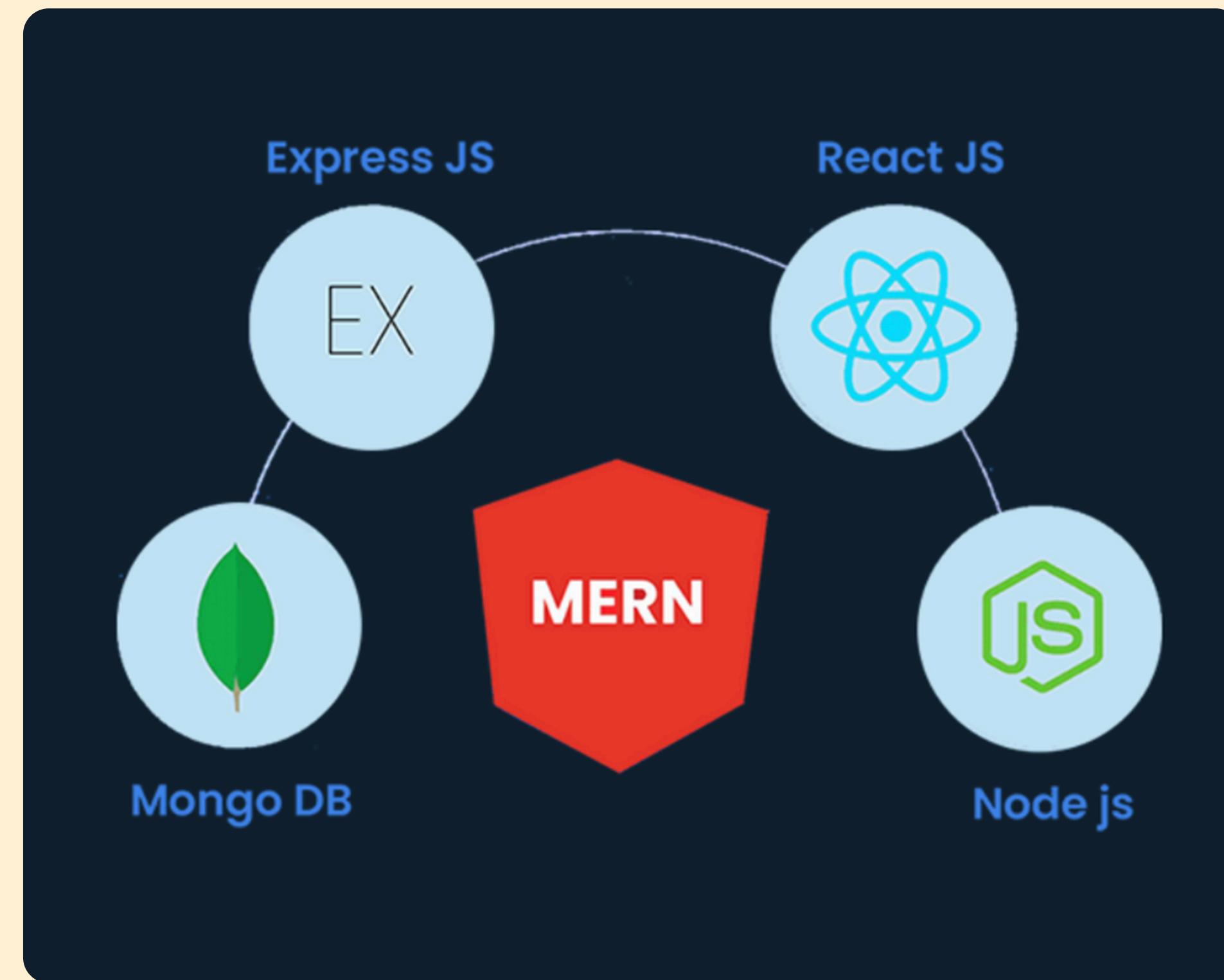


Goals

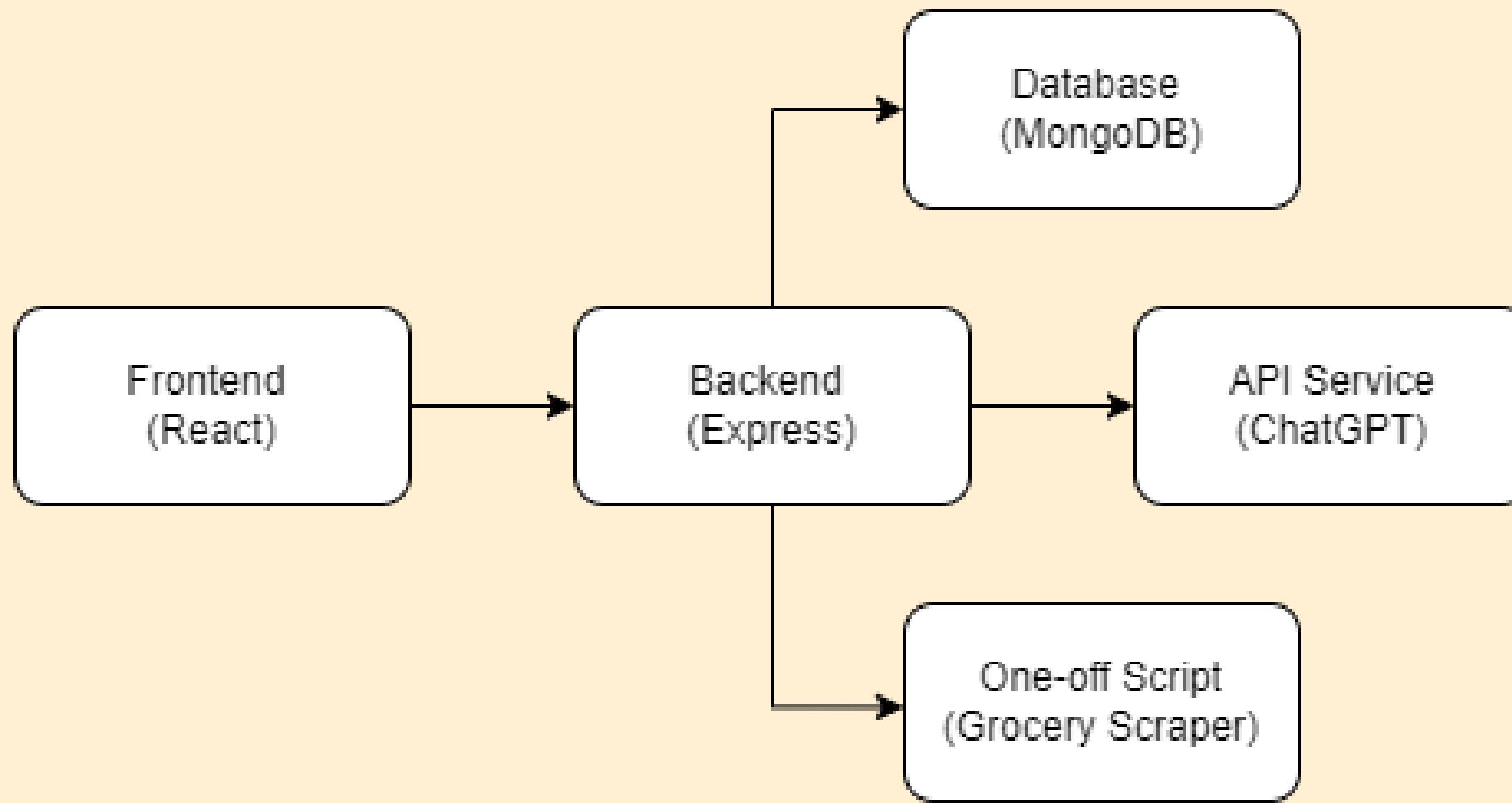
- 1. Diverse Recipe Suggestions**
- 2. Straightforward Meal Planning**
- 3. Pantry Management**
- 4. Grocery Budgeting**
- 5. User-Friendly UI**

Demo

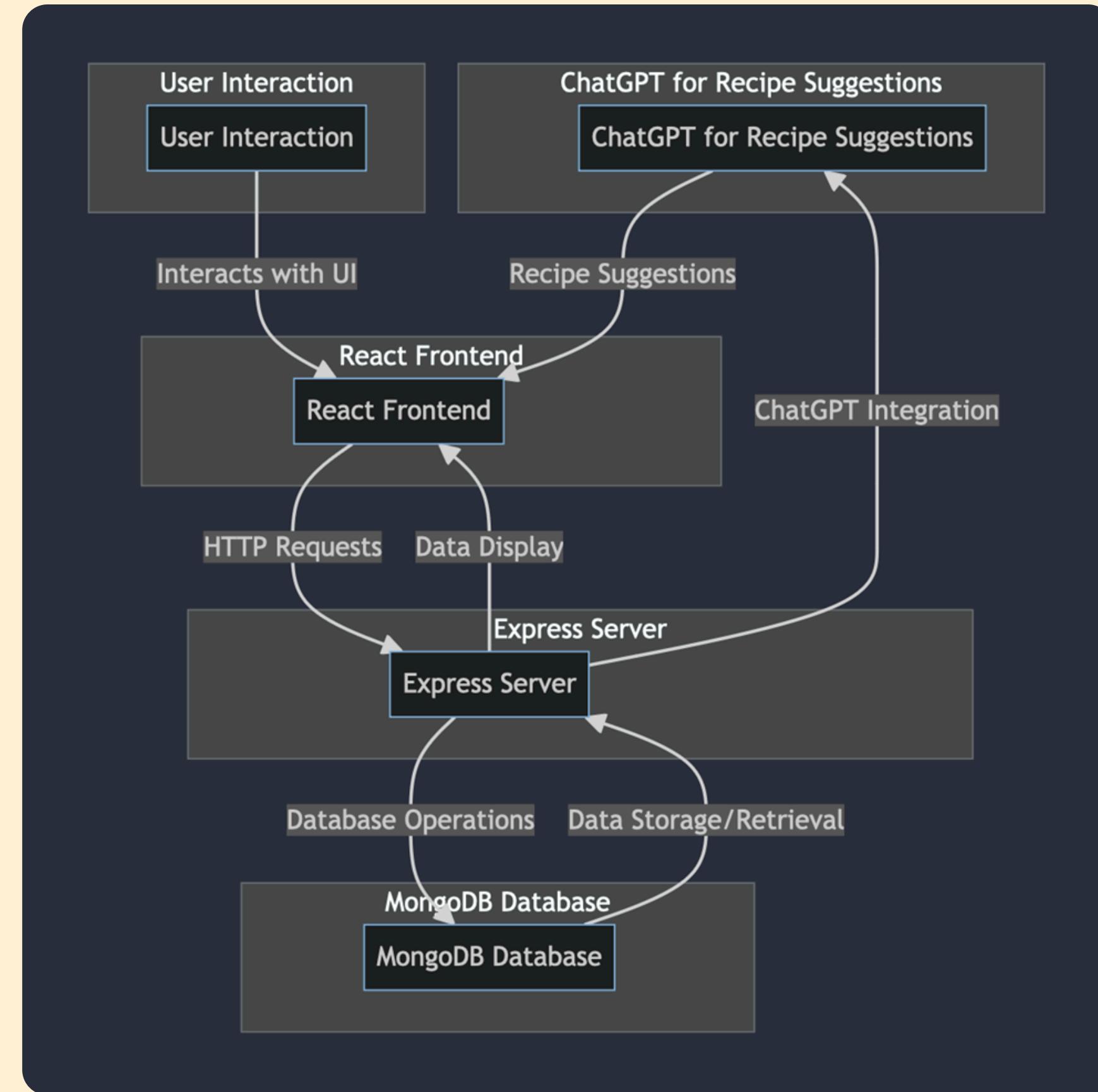
Tech Stack - MERN



Overall Architecture



User Flow



Frontend Architecture

1) Pages:

- Top-level components representing different routes/features in the application

2) Shared Components:

- Re-usable components shared across different pages

3) Hooks

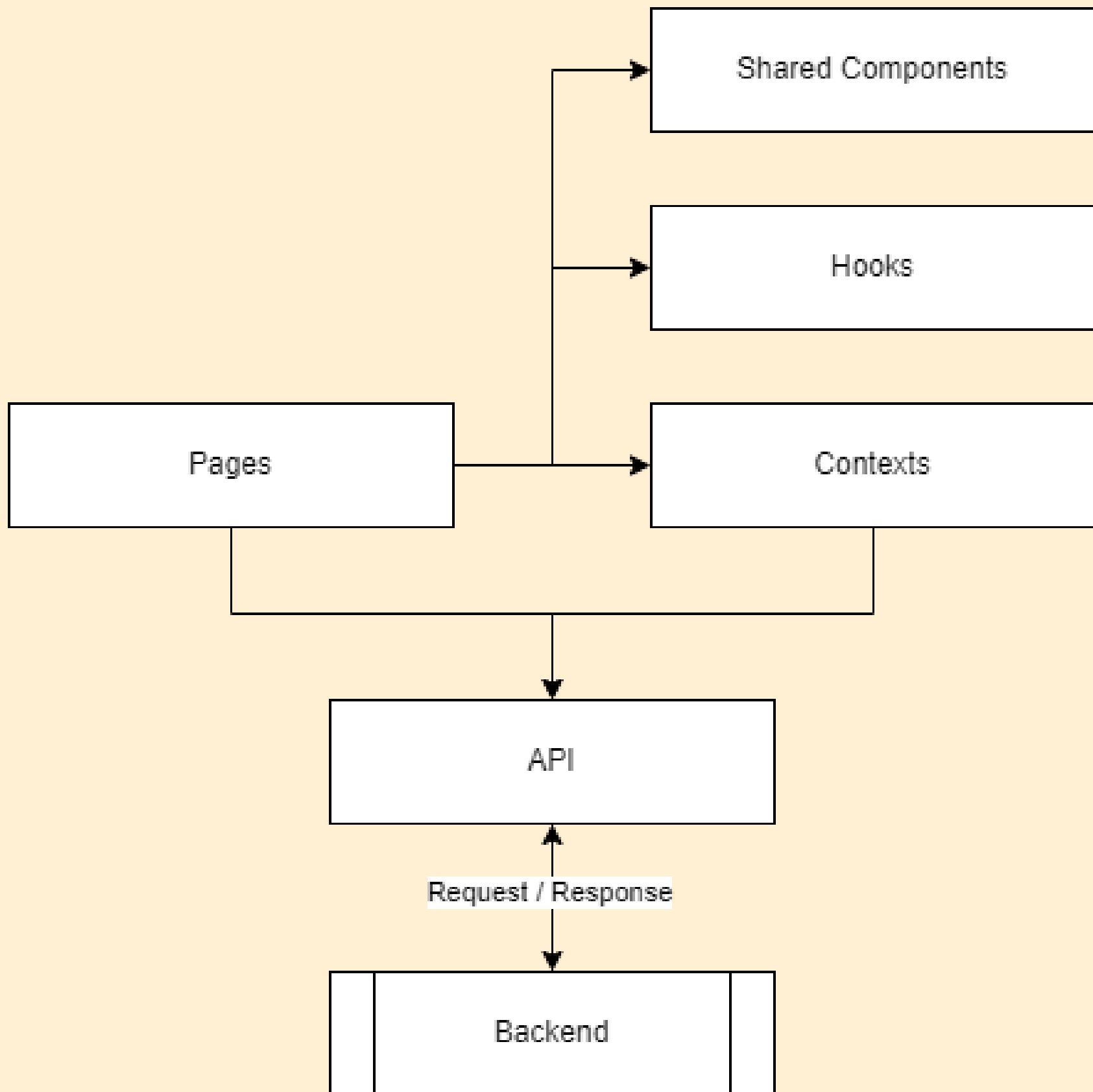
- Custom hooks that encapsulate logic and can be re-used across different components

4) Contexts:

- Used for managing shared state(s) between the different pages

5) API:

- Used for making API calls to the various resources/endpoints of the backend



Backend Architecture

1) Routes:

- Defines the backend's API endpoints
- Entry point for HTTP requests that are sent from the frontend

2) Middlewares:

- Used for request authentication / authorization and validating other input data

3) Controllers:

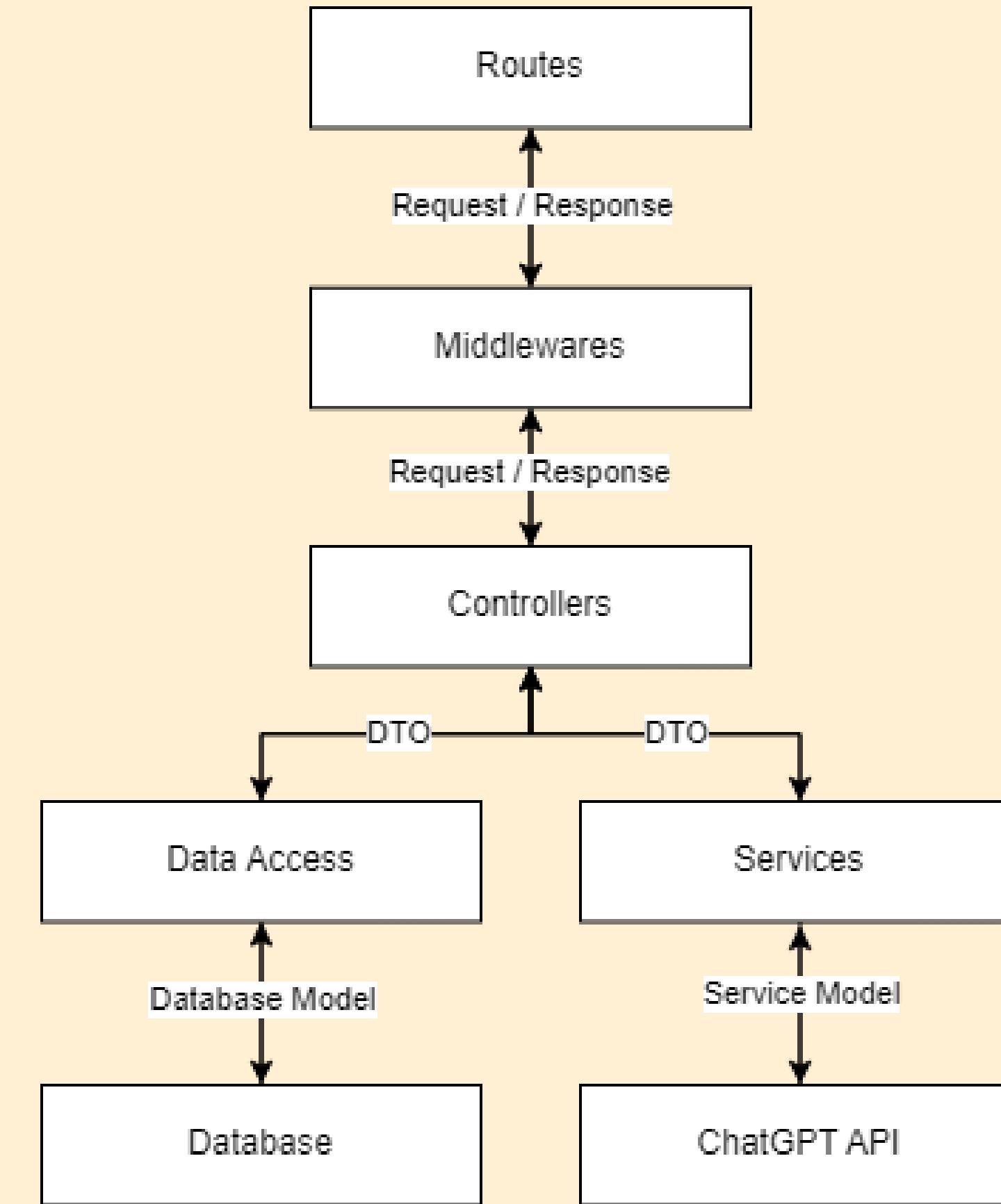
- Used for handling incoming requests, processing them to create and send responses
- Orchestrates main logic by interacting with both the data access and services layers

4) Data Access:

- Handles interactions with the database, abstracting queries and operations

5) Services:

- Handles interactions with external services



Web Scraping

1. Process Overview:

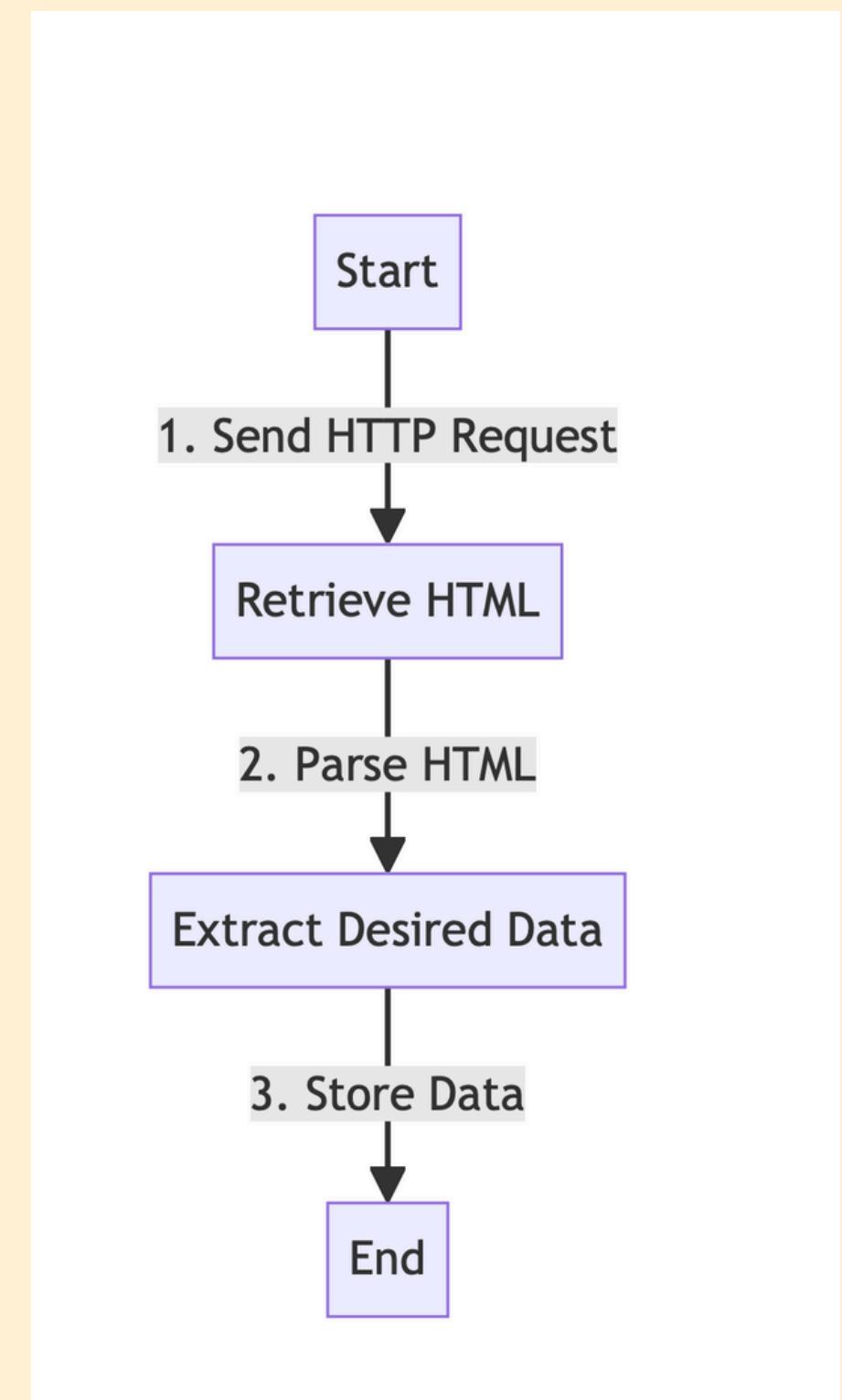
- a. Web scraping extracts data from websites.
- b. It involves sending an HTTP request to retrieve HTML content.
- c. HTML content is parsed to extract desired data.

2. Tools and Libraries:

- a. Requests: Python's Requests library for sending HTTP requests and retrieving web page content.
- b. BeautifulSoup: A Python library for parsing HTML and XML documents, facilitating data extraction.
- c. Scrapy: A Python framework for web scraping that provides a high-level interface for crawling and extracting data.

3. Challenges:

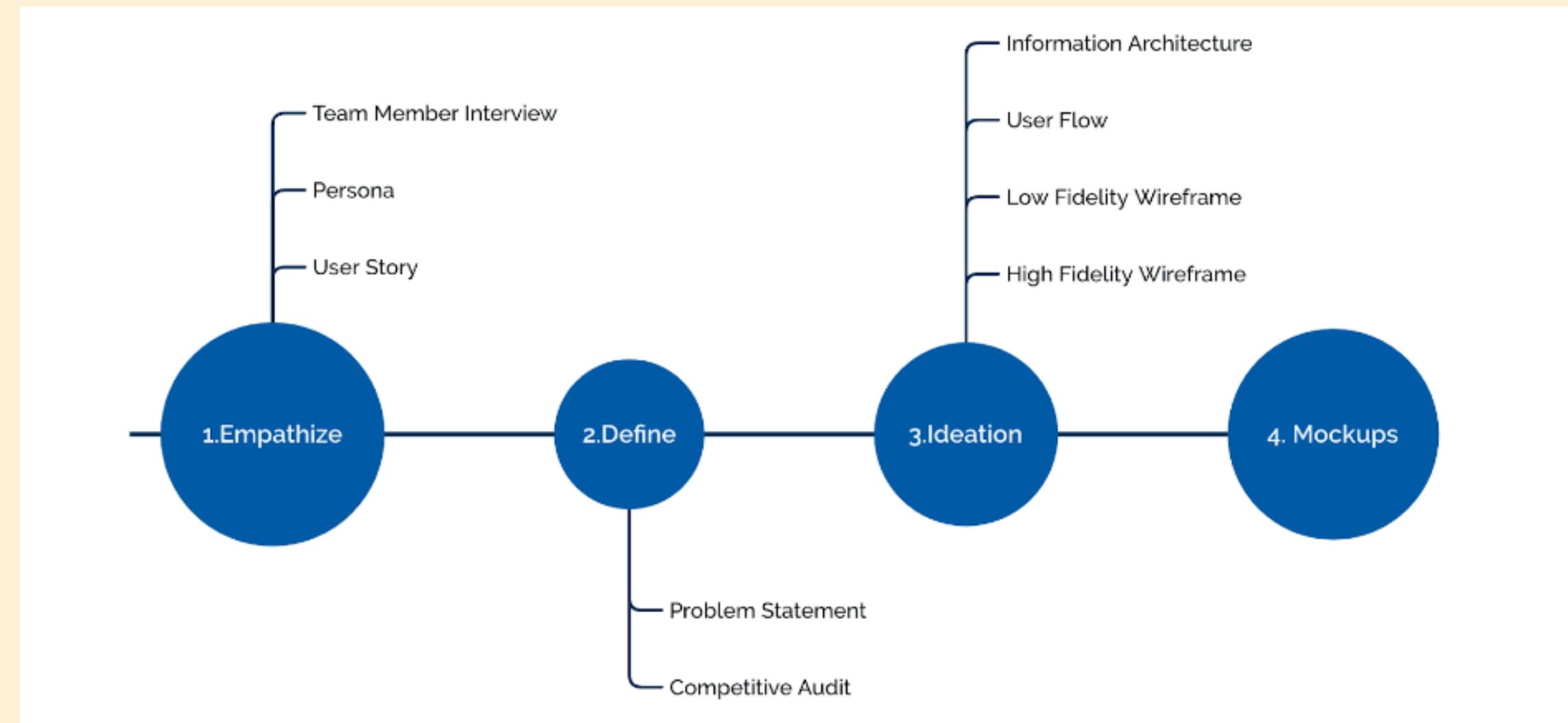
- a. Anti-Scraping Measures: CAPTCHA or IP blocking may be implemented to prevent scraping.
- b. Data Quality: Ensuring accurate and reliable data extraction, especially with inconsistencies or poorly structured HTML.



Design Process

we used design thinking approach, its briefly a non-linear, iterative process that teams use to understand users, challenge assumptions, rederine problems and create innovative solutions to prototype and test.

paragraph text



Personas

We created personas to help design the UI/UX experience:

- Fictional characters
- Represents larger groups of users with certain characteristics and backgrounds

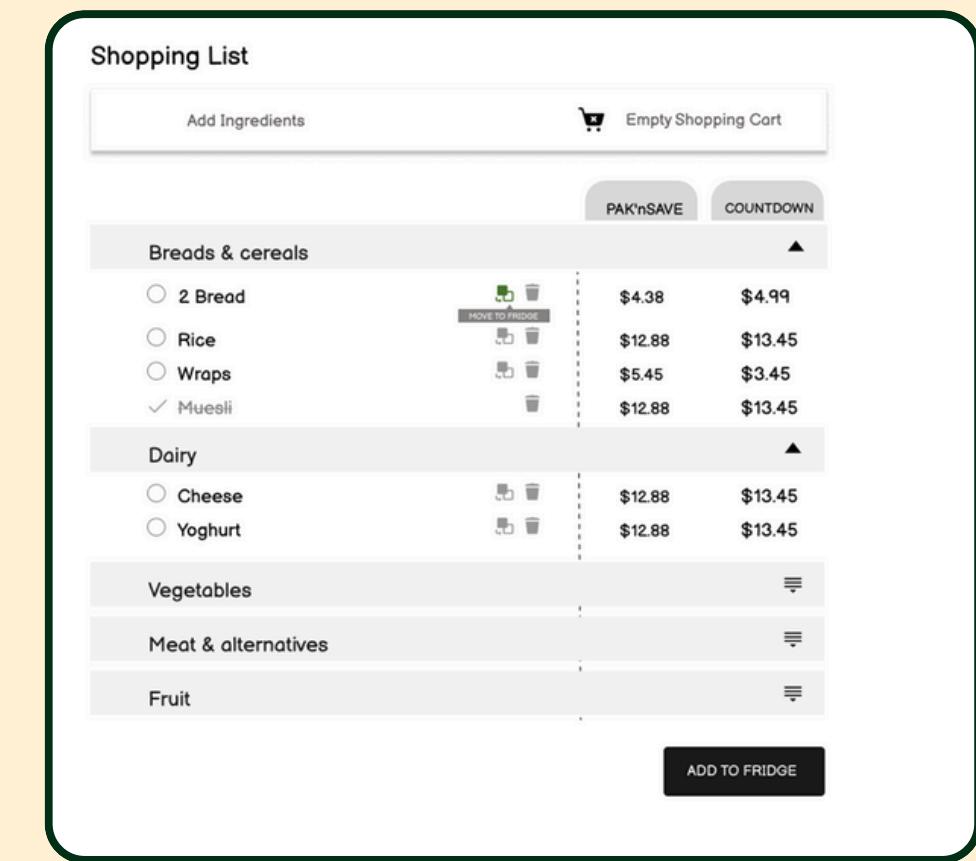
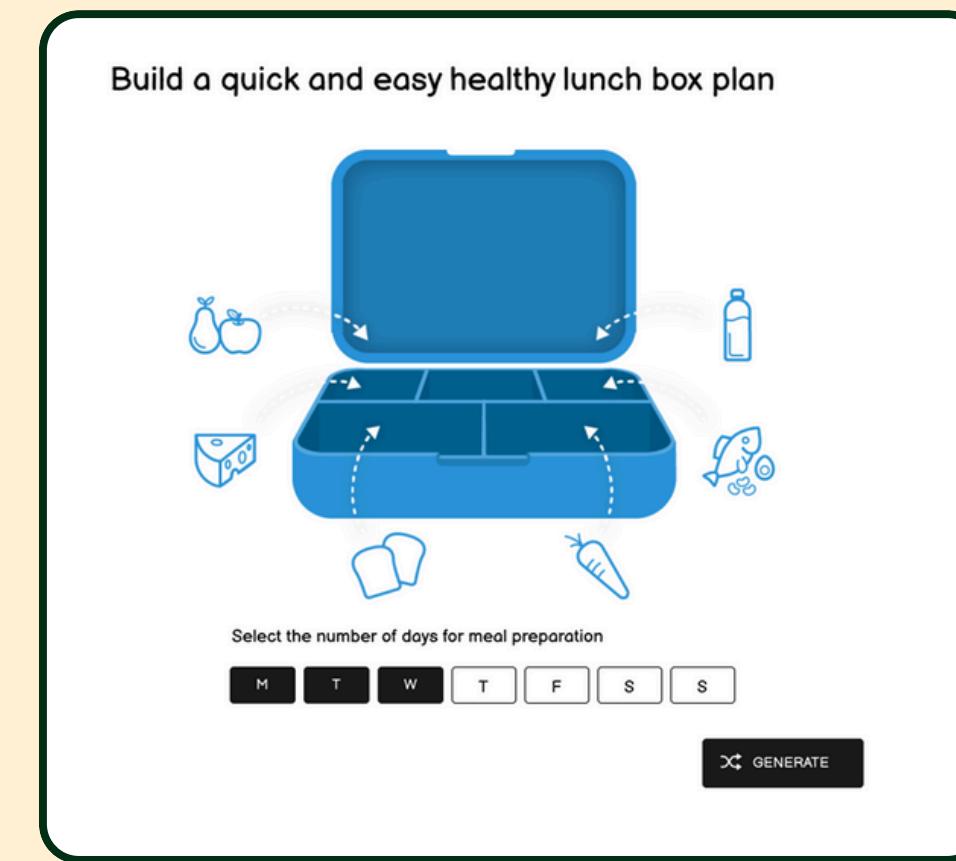
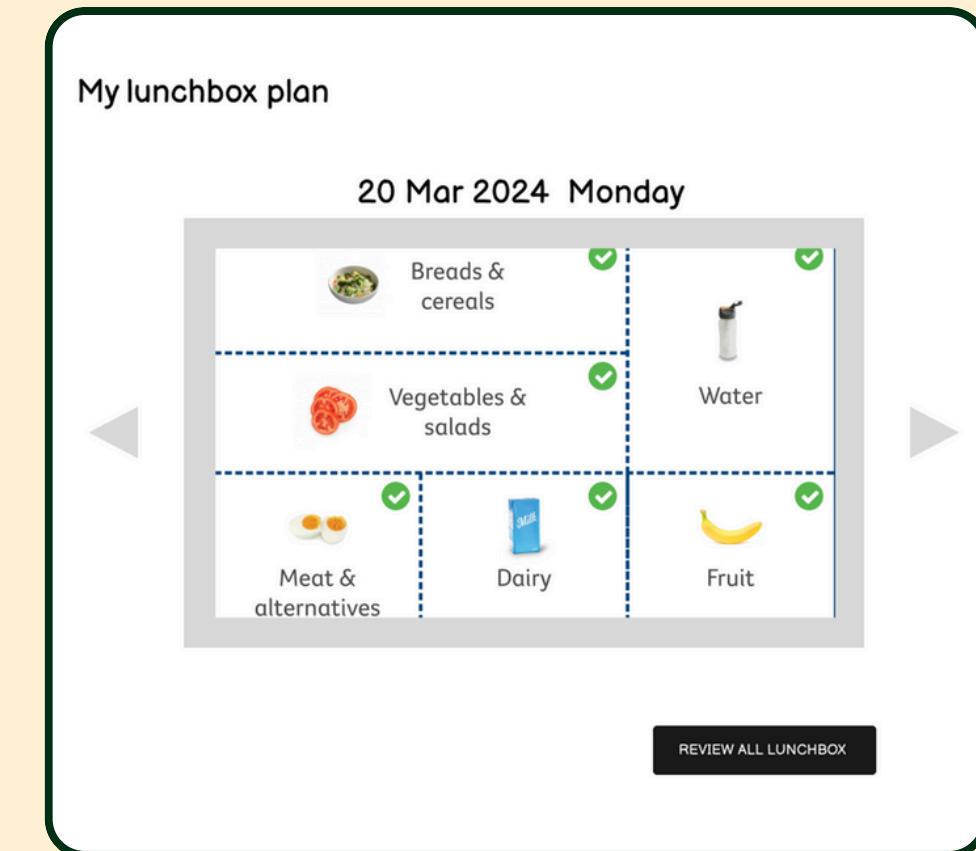
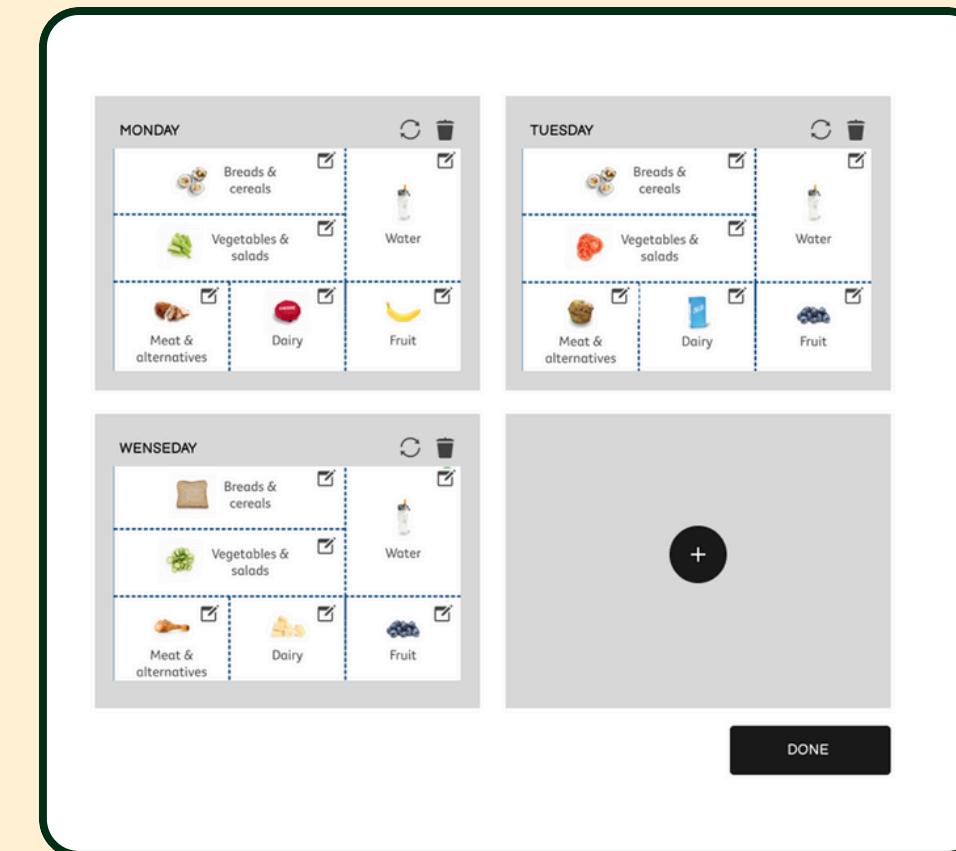
Helen

- Studies in UOA of master degree,
- Works part-time.
- Has a busy life
- Doesn't want to spend time going to market and grocery shopping.
- Spends her weekends with her friends and sometimes by attending workshops.

Alex

- Middle-aged
- Has a happy family life with her husband and daughter.
- Enjoys preparing lunch meal for her child.
- Prefers cheapest options when shopping

Low Fidelity Wireframe



High Fidelity Wireframe

The wireframe displays the following screens:

- Welcome Screen:** Features a large green cartoon monster, a "Welcome, let's get yummy!" message, and "Sign Up" and "Forgot Password" buttons.
- Create Account Screen:** Shows a form for entering Name, Email, Password, and Address, with a "Sign up >" button and a "Login" link.
- Reset Password Screen:** Shows a form for entering Name, Email, and Password, with "Back" and "Reset" buttons.
- My Fridge Screen:** Displays a grid of ingredients (Eggs, Watermelon, Apples, Cabbage, Rice) with quantity and serving options, and buttons for "Create Meal Plan" and "Go Groceries".
- New Meal Plan Screen:** Allows selecting the number of days for meal preparation (1 to 8) and a "CREATE" button.
- Building LunchBox Screen:** Shows a grid for Monday and Tuesday with categories like Bread & cereals, Vegetables & fruits, Meat & alternatives, Dairy, and Fruits.
- Shopping List Screen:** Lets users add ingredients (Bread, Rice, Mashed) to a list, with a "Done" button and an "ADD TO FRIDGE" button.
- My Weekly Meal Plan Screen:** Shows a weekly meal plan for Monday through Thursday, with a "Select a day for view meal plan" button and an "Add Meal" button.
- My Favourite Recipes Screen:** Lists "Classic Bruschetta" recipes with details like 11 ingredients, 45 minutes, and 310 Calories, with a "SEE ALL" button.

Project Management

- **Agile approach:**
 - 6 development sprints
 - Story points workload estimation
 - User stories
- **Ticket Management / Allocation**
 - Trello Kanban board
- **Version Control:**
 - GitHub
- **Communication:**
 - Discord
 - In-person meetings



Trello Boards

The image shows a Trello board titled "Marvelous Moose Softeng Project". The board is divided into six main sections:

- User Stories**: Contains cards for managing fridge inventories, grocery browsing on a budget, shopping lists, user profiles, making lunchboxes, and generating recipes.
- Product Backlog**: Contains cards for creating an API for frontend usage, integrating a lunchbox with fridge inventory tracking, ChatGPT meal plan generation, ChatGPT recipe suggestions based on user fridge content, and integrating recipe generation with fridge inventory quantity tracking.
- Sprint Backlog**: Contains cards for UI matching project standards, extending the Grocer app, adding a "Confirm purchase" button, and integrating ChatGPT recipe generation into the fridge inventory UI.
- Doing**: Contains cards for a search bar in the fridge inventory management page and a login/signup page.
- Waiting for Peer Code Review/Integration**: Contains cards for lunchbox preparation day selection and user authentication with MongoDB.
- Current Sprint Done**: Contains cards for creating a login/signup page and lunchbox content selection.
- Project Done**: Contains cards for implementing a menu for navigation, routing configuration, UX discovery (analyzing competitor websites/apps), and UX discovery (creating Lofi prototypes).

Each card includes a description, a due date, and a set of labels (e.g., AC, BZ, AT, DS, L) indicating its priority or dependencies. The "Waiting for Peer Code Review/Integration" and "Current Sprint Done" columns are highlighted with red borders, while the "Project Done" column has a yellow background. The overall background of the board is a photograph of a wooden surface with a lemon and lime.

Future Work

- Users can set a food budget while creating their shopping list
- Users can access details like preparation time through the UI
- Users can tell ChatGPT what they want to cook first and receive a response
- The app can save favorite recipes and ingredients for easier access
- Users can take a photo of their open fridge, and the app will automatically identify the food items



Thank You!

We are now open to any
questions

