**Frontend Development with React.js**

**Project Documentation for Recipe Application**

1. **Introduction**

• Project Title: Cook Book

• Team Members:

Sandhiya R [Email id:sandhiyarraja7@gmail.com]

Indhumathi R [Email id:indhumathiramesh2004@gmail.com]

Shalini S [Email id:sharmass1246@gmail.com]

Thilothamma T [Email id:abithilo777gmail.com]

1. **Project Overview**

**Purpose:**

The primary purpose of the Recipe Application is to make cooking accessible and enjoyable by offering an extensive collection of recipes with user-friendly features.

* + **Recipe Discovery**: Help users find new and diverse recipes based on their preferences and dietary restrictions.
  + **Meal Planning**: Enable users to plan meals in advance with easy-to-follow recipes.
  + **Sharing and Collaboration**: Allow users to share their own recipes and explore those created by others.

**Features**:

* + - **User Accounts**: Allow users to create accounts and save favorite recipes, meal plans, and grocery lists.
    - **Recipe Search & Filtering**: Provide a search feature with filters such as cuisine, meal type (breakfast, lunch, dinner), dietary needs (vegan, gluten-free, etc.), and preparation time.
    - **Recipe Details**: Each recipe will include detailed instructions, a list of ingredients, nutritional information, and the number of servings.
    - **Personalized Suggestions**: Recommend recipes based on previous activity, taste preferences, and dietary restrictions.

**3.Architecture:**

* Frontend (Client-Side)
* Backend (Server-Side)
* Database Layer
* External APIs (Optional)
* Cloud Storage (Optional)
* Authentication & Authorization

**1.Client-Side (Frontend)**

**Technologies**: React.js / Angular / Vue.js, HTML, CSS, JavaScript, Axios (for API requests)

The frontend is the user interface of the application, which interacts directly with users. It should be highly interactive and visually appealing to enhance the user experience.

**2.Backend (Server-Side)**

**Technologies:** Node.js with Express.js / Python with Django / Flask, RESTful APIs, JSON Web Tokens (JWT) for authentication, B crypt (for password hashing)

The backend handles all the business logic, data processing, and communication with the database. It processes requests from the frontend and ensures that data is correctly stored, retrieved, and manipulated.

**3.Database Layer**

**Technologies**: MongoDB (NoSQL) / PostgreSQL (SQL), Redis (for caching)

The database stores all the application's data, including user information, recipes, ratings, meal plans, and user-generated content.

**4. Setup Instructions**

Prerequisites:

Node.js (v16 or higher)

npm (v8 or higher)

Git

Installation:

1. Clone the repository:

2. Navigate to the client directory:

3. Install dependencies: npm install

4. Configure environment variables: Create a .env file in the client directory and

add the necessary variables (e.g., API keys).

1. Start the development server: npm start.
2. **Folder Structure**

**Client:**

* /assets: Contains all static assets like images, fonts, and CSS styles.
* /images: Recipe photos or illustrations.
* /fonts: Custom fonts for the design.
* /styles: Stylesheets for the layout and design of the cookbook  **/pages**: Pages for the application (e.g., homepage, individual recipe pages)
* **/public:** Public-facing files like index.html, favicon, etc.

**Utilities**:

* + - * /dataprocessing: Scripts for handling raw recipe data, cleaning data, or importing recipes.
      * /scripts: Backend scripts that may automate tasks like generating a table of contents, exporting the cookbook to PDF, etc.
      * /services: Services for handling business logic, like managing recipe data or communicating with an API.
      * /metadata: Stores metadata like recipe tags or cookbook information.

**6. Running the Application:**

* **To start the frontend server, run the following command in the client directory:**
* **npm start**
* **npm install**
* **npx json-server ./db/db.json**
* **npm run dev**
* **The application will be available at** [**http://localhost:3000**](http://localhost:3000)

**7.Component Documentation:**

**Key Components:**

1. **Recipes Management –** Organizes recipes with clear, standardized content (ingredients, instructions).
2. **User Interface Components** – The visual design and interaction elements of the cookbook (e.g., homepage, recipe pages).
3. **Recipe Database (Backend)** – Stores all recipes, user data, and cookbook metadata in an organized structure.
4. **Content Management System (CMS) –** A tool for managing content and recipe creation/editing.
5. **Cookbook Layout/Design** – Defines the aesthetic layout and structure of the cookbook, both digital and print.
6. Top of Form

**Reusable Components:**

**Recipe Card:**

Displays a preview of a recipe, typically with its name, image, brief description, and a link to view the full recipe.

**8.State Management:**

* Informational Statements: Giving general information about recipes, categories, etc.
* Error Statements: Helping users understand issues they may encounter.
* Success Statements: Celebrating completed actions (e.g., saving a recipe).
* Confirmation Statements: Validating user actions like deleting or editing recipes.
* Instructional Statements: Guiding users through the usage of features.
* Warning Statements: Alerting users to potential risks or issues.
* Loading/Progress Statements: Informing users when the system is processing actions.
* Feedback Statements: Giving users feedback on their ratings, reviews, or actions.

**10. Styling:**

• CSS Frameworks/Libraries: The application uses Styled-Components for styling. This allows for modular and scoped CSS within components.

• Theming: A custom theme is implemented using Styled-Components, with support for light and dark modes.

**11. Testing:**

Testing Strategy: o Unit Testing: Using Jest and React Testing Library.

Integration Testing: Is performed to ensure that components work together as expected.

End-to-End Testing: Cypress is used for end-to-end testing of user flows.

• Code Coverage: o Code coverage is monitored using Jest’s built in coverage tool. The current coverage is 85%.

**12. Screenshots or Demo :**

Demo link: <https://drive.google.com/file/d/19fgpR6h81lKn5A35DQBPE5GYfDTivTW/view?usp=drivesdk>

**13.Future Enhancement:**

* User Personalization
* Recipe Creation & Sharing
* Advanced Search & Filters
* Enhanced User Engagement

This documentation provides a comprehensive overview of the Cook Book project, including its architecture, setup instructions, and future plans.