**Frontend Development with React.js**

**Project Documentation format**

1. **Introduction**
   * **Project Title**: [Cookbook: Your virtual kitchen Assistant]

**Team Members**:

* + **Gokul Krishnan { Team leader}**
  + **Dhina {video make}**
  + **Neelakandan {code editing}**
  + **Anadhanraman {github posting}**
  + **Poovarasan {checking program}**

1. **Project Overview**
   * **Purpose**:

**The purpose of the project is to make easy work decisions for cooking its very simple and man was important role is food its making is must important to human life there is easy to making desires and its teach step by step information to all decisions that is only goals.**

* + **Features**:

1. Easy design make simple and order linear style.
2. There are image section is appear in home.
3. Simple UI designs visions shout be location.
4. All of the type ingredient is mention step by step.
5. **Architecture**
   * **Component Structure**:

***App.js: Root component managing routes.***

***Naval: Navigation across pages.***

***Recipe List: Displays a list of recipes.***

***Recipe Details: Shows ingredients, steps, and instructions.***

***Favourites: Stores user’s saved recipes.***

***Search Bar: Allows search by keyword or ingredient***

***User Profile: User’s personalized settings and preferences***

* + **State Management**:

Global State: Managed with Redux for recipes, user data, and favorites.

Local State: Managed using React hooks (useState/useReducer) for form inputs and component-specific states.

* + **Routing**:

**Implemented using React Router DOM.**

**Routes include:**

**/ → Home**

**/recipes → Recipe list**

**/recipes/:id → Recipe details**

**/favorites → Saved recipes**

**/profile → User profile**

1. **Setup Instructions**
   * **Prerequisites**:

Node.js (v16 or later)

npm or yarn

Git

* + **Installation**:

**git clone https://github.com/your-repo/cookbook.git**

**cd cookbook**

**npm install**

**npm start**

.

1. **Folder Structure**
   * **Client**:

**/src**

**/components**

**Navbar.js**

**RecipeList.js**

**RecipeDetails.js**

**Favorites.js**

**SearchBar.js**

**UserProfile.js**

**/pages**

**Home.js**

**Recipes.js**

**Profile.js**

**/redux**

**store.js**

**recipesSlice.js**

**favoritesSlice.js**

**/assets**

**images/**

**styles/**

**App.js**

**index.js**

* + **Utilities**:

api.js → Handles API calls (e.g., fetching recipes).

helpers.js → Utility functions like unit conversions, string formatting.

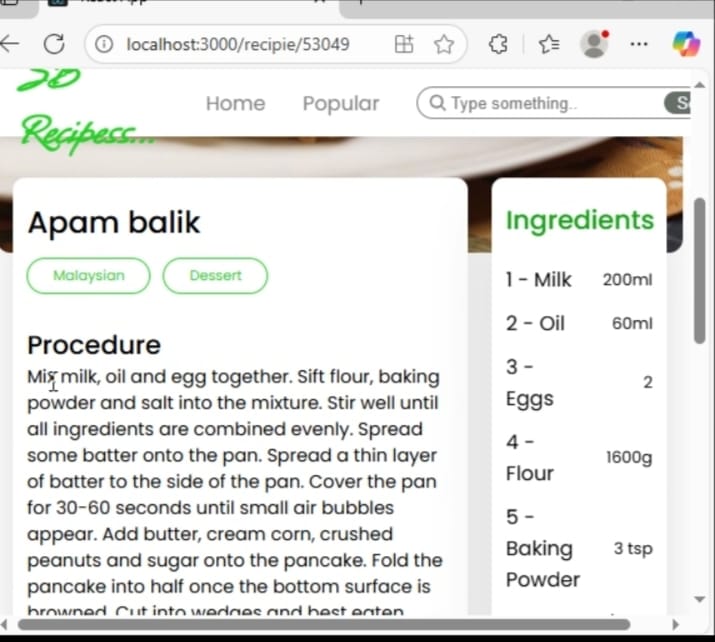
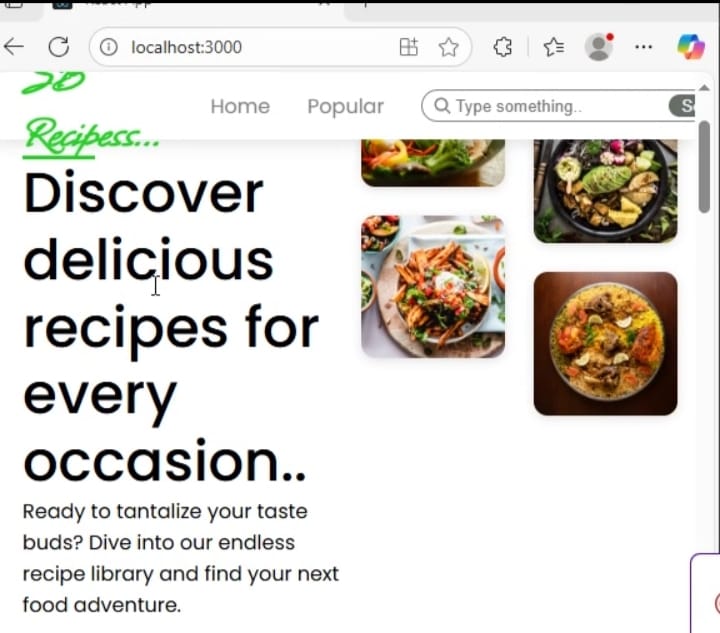
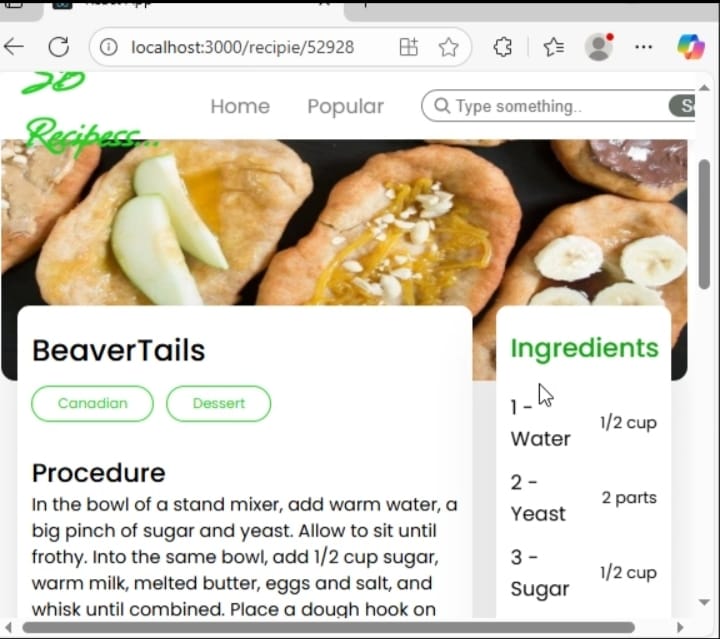
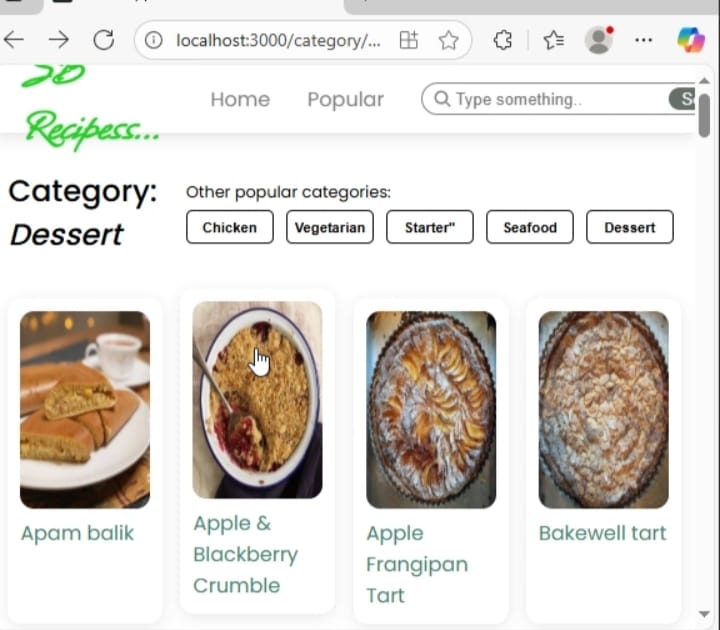
1. **Running the Application**
   * Provide commands to start the frontend server locally.
     + **Frontend**: npm start in the client directory.
     + Runs the app in development mode at http://localhost:3000.
2. **Component Documentation**
   * RecipeList: Fetches and displays recipes based on search/filter.
   * RecipeDetails: Displays recipe instructions, images, and cooking steps.
   * Favorites: Allows users to save and manage favorite recipes.
   * Reusable Components
   * SearchBar: Used across multiple pages.
   * Button, Card, Modal: Reusable UI components
3. **State Management**
   * **Global State**: Recipes, user authentication, and favorites stored in Redux.
   * **Local State**: Input handling in search bar, form data in profile settings.
4. **User Interface**
   * Provide screenshots or GIFs showcasing different UI features, such as pages, forms, or interactions.
5. **Styling**

* **CSS Frameworks/Libraries**: CSS Frameworks/Libraries:
* Tailwind CSS for utility-first styling.
* Styled-components for component-level customization.
* **Theming**: Light and Dark mode implemented.

1. **Testing**

* **Testing Strategy**: Unit testing with Jest.
* Component testing with React Testing Library.
* End-to-end testing with Cypress (optional).
* **Code Coverage**: Jest and Istanbul (nyc) for ensuring >80% coverage.

1. **Screenshots or Demo**

* 
* 

1. **Known Issues**

* Limited recipe API free-tier may restrict daily requests.
* Search performance may slow down with large datasets

1. **Future Enhancements**

* Voice-assisted recipe navigation (hands-free cooking).
* Meal planner & grocery list generator.
* Integration with IoT smart kitchen devices
* Multi-language recipe support.