Cook Book: Your Visual Kitchen Assistant

1. Introduction

Project Title: Cook Book – Your Visual Kitchen Assistant

Team leader: Sanmathi Kumar

Team Members: Arifa Zohra, Ainul Liyana PM, Vaishali. S

**2. Project Overview**

Purpose:

The Cook Book project is a virtual kitchen assistant application that helps users discover recipes and explore food categories with ease. It simplifies cooking by providing detailed steps and ingredient lists.

Features:

Browse recipes by categories such as chicken, beef, pasta, and more.

Search recipes using dish name, ingredients, or category.

Add and remove recipes from the favourites list.

View step-by-step instructions with ingredient details.

**3. Architecture**

Component Structure:

Navbar Component – Navigation bar for easy access to categories and favourites.

Recipe List Component – Displays recipes from different categories.

Recipe Detail Component – Shows instructions and ingredients for a selected recipe.

Favourites Component – Manages the user’s saved recipes.

Search Component – Handles search queries by dish name or ingredient.

State Management:

Local state is managed using React’s useState and useEffect hooks. JSON Server is used for handling data persistence like recipes and favourites.

Routing:

Implemented using React Router, enabling navigation between home, categories, recipe details, and favourites.

**4. Setup Instructions**

Prerequisites:

Node.js

Visual Studio Code

Installation:

1. Create a folder named Cook Book and open it in VS Code.

2. Download and extract the project source code into this folder.

3. Open a terminal in VS Code and run:

npm install

npm run dev

4. For database setup:

cd db

npm install -g json-server

json-server --watch db.json --port 3000

**5. Folder Structure**

Client:

src/components – Navbar, RecipeList, RecipeDetail, Favourites, Search

src/pages – Home, Categories, RecipePage

src/assets – Images, icons, and styles

Utilities:

Helper functions for API calls

Custom hooks for managing recipes and favourites

**6. Running the Application**

Frontend:

Run the command:

npm run dev – Starts the frontend locally

Backend (JSON Server):

json-server --watch db.json --port 3000

**7. Component Documentation**

Key Components:

Navbar – Provides navigation options

RecipeList – Displays recipe cards by category

RecipeDetail – Shows instructions and ingredients

Favourites – Manages saved recipes

Search – Filters recipes based on input

Reusable Components:

RecipeCard – Reused in multiple lists

Button & Input Components – For user actions

**8. State Management**

Global State:

Recipes and favourites data flow across components using props drilling and React hooks.

Local State:

Search queries, category filters, and favourites list are handled using useState.

**9. User Interface**

Simple, clean, and user-friendly design.

Recipe cards with images, titles, and categories.

Interactive search bar.

Favourites section for quick access.

**10. Styling**

CSS Frameworks/Libraries:

Plain CSS3 with flexbox and grid for layout.

Theming:

Simple responsive theme with consistent colors for categories and recipes.

**11. Testing**

Testing Strategy:

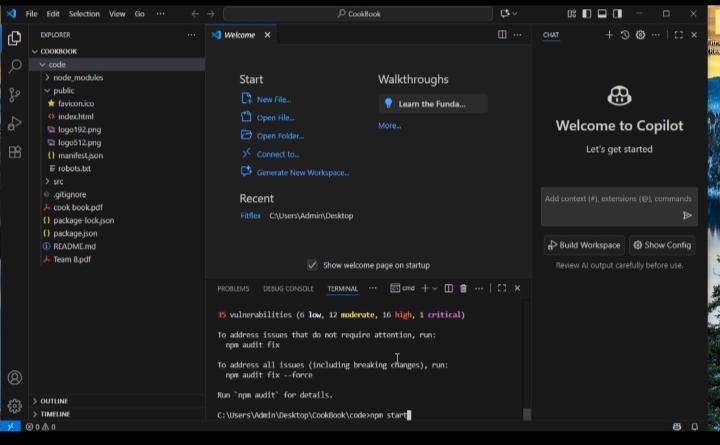
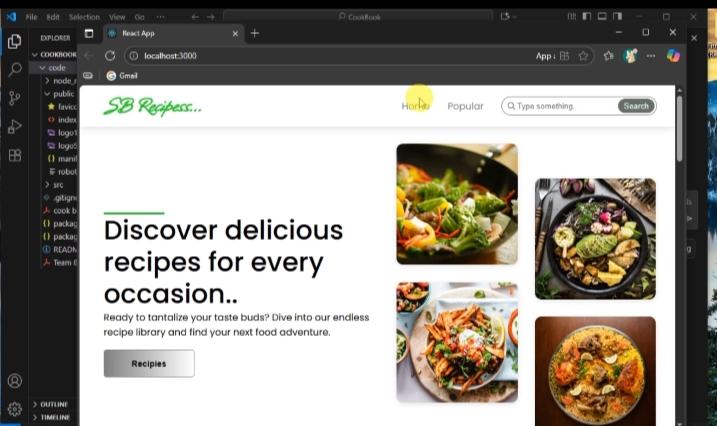
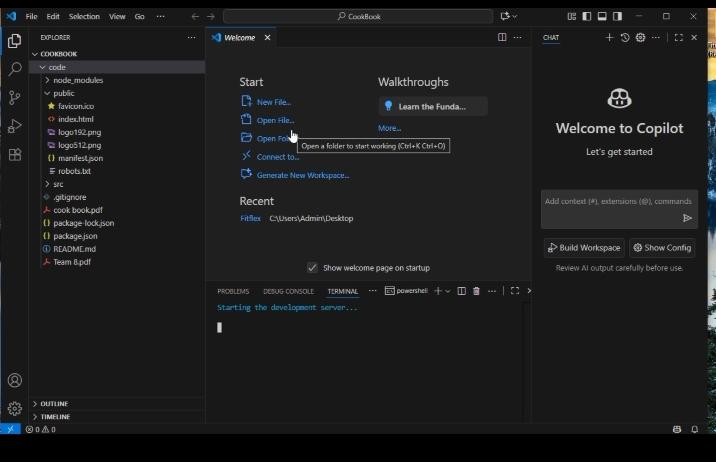
The project was manually tested to ensure that all major features, such as recipe browsing, search functionality, adding/removing favourites, and database integration with JSON Server, work as expected. Each component was tested for proper rendering and responsiveness.

(Future plan: integrate automated testing using Jest and React Testing Library for unit and component testing.)

Code Coverage:

At this stage, testing was mainly functional and exploratory. Formal test coverage tools were not implemented, but all core features were validated through multiple usage scenarios to ensure reliability.

**12. Screenshots or Demo**



**Demo video link:**

<https://drive.google.com/file/d/1PjlOegBFsvWMoKNEVx2tZZJW1WkPwkc9/view?usp=drivesdk>

**13. Known Issues**

Limited recipe dataset as of now (only sample categories included).

No authentication system (users cannot create accounts yet).

Favourites are stored locally and reset if the JSON server is restarted.

Application UI is responsive but may need more optimization for smaller devices.

**14. Future Enhancements**

Implement user authentication (login/signup) to save favourites permanently.

Add more categories and an expanded recipe dataset.

Integrate API-based recipes for real-time content.

Improve UI with animations and advanced theming.

Add shopping list feature – allowing users to generate a list of ingredients from selected recipes.

Enable voice search for recipes.