**1. Introduction:**

**COOK BOOK: YOUR VIRTUAL KITCHEN ASSISTANT**

Project Title: Cook Book: Your Virtual Kitchen Assistant

Team Leader:

VIJAYAKUMAR M (Vijayakumar463600@gmail.com)

Team Members:

THIRUPATHI S (Thirupathis968@gmail.com

THARUN T (Tharunthagavel26@gmail.com

SATHASIVAM N (Sathansivam394@gamil.com

DEEPAK K (Deepakkaruppannan@gamil.com

The Cook Book project is designed as a virtual kitchen assistant that helps users search, explore, and organize recipes easily. It simplifies cooking by providing step-by-step instructions, ingredient lists, and personalization features.

**2. Project Overview:**

Purpose:

To provide users with a digital cooking companion that makes meal planning and cooking more convenient, interactive, and enjoyable.

Features:

* Recipe search and filtering by ingredients.
* Step-by-step cooking instructions.
* Favorites and saved recipes.
* User-friendly interface with responsive design.
* Easy navigation for beginners and advanced users.

**3. Architecture:**

Frontend: React.js (with component-based architecture).

Backend: Node.js / Express (or Firebase if applicable).

Database: MongoDB / Firebase Firestore.

API Integration: External recipe APIs (e.g., Spoonacular, Edamam).

Component Structure:

* Search Bar → allows recipe search.
* Recipe Card → displays recipe details.
* Favorites → manages user’s saved recipes.
* Instructions → guides users with cooking steps.

Routing: Implemented with react-router-dom (e.g., /home, /search, /favorites).

**4. Setup Instructions:**

Prerequisites

* Visual studio code
* npm or yarn
* Git

Installation:

git clone <repo-url>

cd cookbook-app

npm install

npm start

**5. Folder Structure:**

cookbook-app/

├── public/

├── src/

│ ├── components/

│ │ ├── RecipeCard.js

│ │ ├── SearchBar.js

│ │ └── Favorites.js

│ ├── pages/

│ │ ├── Home.js

│ │ ├── Search.js

│ │ └── Profile.js

│ ├── state/

│ │ └── context.js

│ ├── styles/

│ │ └── main.css

│ ├── utils/

│ │ └── api.js

│ └── App.js

├── package.json

└── README.md

**6.Running the Application:**

* cd code
* npm install
* npm start

**7. Component Documentation:**

* RecipeCard: Displays recipe image, title, and link to instructions.
* SearchBar: Input field with filters for recipe queries.
* Favorites: Shows saved recipes.
* Instructions: Guides users with cooking steps.
* Navbar: Provides navigation between pages.

**8. State Management:**

* Global State: Managed using Context API (or Redux if chosen).
* Local State: Controlled within components for inputs, toggles, etc.

**9. User Interface:**

* + Responsive and mobile-friendly layout.
  + Main navigation: Home, Search, Favorites, Profile.
  + Intuitive card-based recipe display.

**10. Styling:**

* CSS / TailwindCSS / Material UI.
* Warm food-themed color palette.
* Consistent typography and spacing.

**11. Testing:**

* Unit Tests: Jest + React Testing Library.
* Integration Tests: Test major flows like search → view recipe → save recipe.
* Coverage Tools: Istanbul / built-in Jest coverage.

**12. Screenshot or Demo:**

**13. Known Issues:**

* API rate limits may cause search failures.
* Limited offline support.
* Some UI responsiveness issues on smaller screens.

**14. Future Enhancements:**

* AI-powered personalized meal planning.
* Voice assistant for hands-free cooking.
* Multilingual recipe support.
* Grocery list generation + integration with delivery service.