

Project Documentation

CookBook: Your Virtual Kitchen Assistant

Project Title: CookBook: Your Virtual Kitchen Assistant

Team Leader: M.Ramzan Begum
ramzanbegum2122@gmail.com

Team Members:

1. A.Dhevayani
adhevayani1122@gmail.com

2. A.Shifana
shifanaa305@gmail.com

3. N.Faiza Naaz
nasarfaiza5@gmail.com

1. Introduction:

• Project Title: CookBook: Your Virtual Kitchen Assistant • Team ID: •
Team Leader: Name & Mail id • Team Members: – Name & Mail id – Name & Mail id

2. Project Overview :

• The primary goal of CookBook is to provide a user-friendly platform that caters to individuals passionate about cooking, baking, and exploring new culinary horizons. Our objectives include: • User-Friendly Experience: Create an interface that is easy to navigate, ensuring users can effortlessly discover, save, and share their favourite recipes. • Comprehensive Recipe Management: Offer robust features for organizing and managing recipes, including advanced search options. • Technology Stack: Leverage modern web development technologies, including React.js, to ensure an efficient, and enjoyable user experience. • Purpose: CookBook is a virtual kitchen assistant designed to help users set up and execute a cooking project by guiding them with prerequisites, setup, and execution steps. • Features: – Guided setup for Node.js installation – Step-by-step instructions to run the project – Local server hosting for accessing the application

3. Architecture:

• Frontend: React.js • Backend: Node.js and Express.js • Database: Not specified

4. Setup Instructions:

- To get the Application project from drive: • Download project code: https://drive.google.com/drive/folders/1u8PnV_mE0mwKkH_CvuNpliZtRLJZMqrO?usp=sharing • Navigate into the cloned repository directory and install libraries: `cd fitness-app-react npm install` • Start the development server: `npm start` • Access the app at `http://localhost:3000`

5. Folder Structure:

- Folder Structure `CookBook/` |-- `components/` # Reusable UI components |-- `pages/` # Application pages mapped to routes |-- `styles/` # CSS styling files

6. Running the Application:

- Start with: `npm start` • Access: Visit <http://localhost:3000>

7. API Documentation:

- Recipes: `GET /api/recipes` → Fetch all recipes – `GET /api/recipes/:id` → Fetch recipe by ID – `POST /api/recipes` → Add a new recipe – `PUT /api/recipes/:id` → Update existing recipe – `DELETE /api/recipes/:id` → Delete a recipe

- Users: `POST /api/users/register` → Register a new user – `POST /api/users/login` → Login user – `GET /api/users/profile` → Get user profile (protected)

- Favorites: `POST /api/favorites/add` → Add recipe to favorites – `GET /api/favorites` → Get user's favorite recipes

8. Authentication:

- JWT-based authentication for secure login and user sessions. • Middleware protects private routes such as user profiles and favorites.

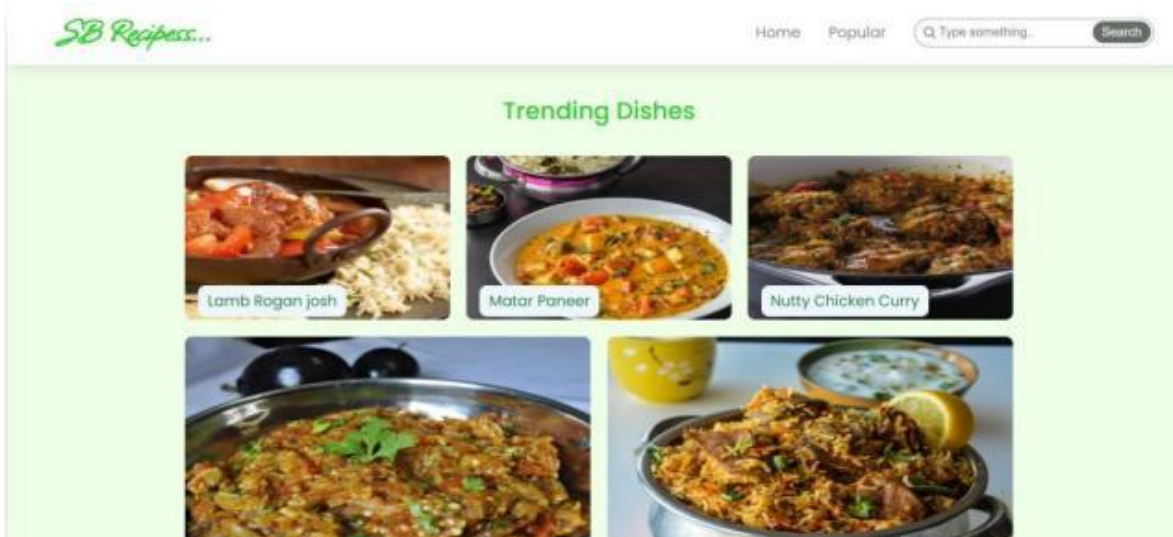
9. User Interface:

- Application launches in browser on localhost • Organized into Pages, Components, and Styles

10. Testing:

- Manual testing while running the application on localhost • Tools: Postman for API testing, Chrome Dev Tools

11. Screenshots or Demo:



12. Known Issues:

- Not documented

13. Future Enhancements:

- More advanced API endpoints (recipe categories, search filters)
- Database integration (MongoDB or MySQL)
- Enhanced authentication (OAuth, social login)
- Mobile responsive design