INTERNET AND WEB SYSTEMS-1 PROJECT PROPOSAL

SAI SRITEJ PALACHARLA

02112185

1. Introduction

Recipe Kavala Babu was founded to bridge the gap in internet platforms dedicated to authentic Indian cuisine. During the COVID-19 pandemic, more people started cooking at home, but they struggled to locate reliable, well-organized recipes. This platform was created to offer a curated collection of both everyday and traditional Indian dishes, allowing users to easily explore, share, and bookmark recipes. Furthermore, it provides premium content such as regional specialties and festival recipes, catering to both novice and professional cooks. It aspires to foster a community of foodies while providing a combination of free and premium material.

2. What?

Recipe Kavala Babu is an easy-to-use web application with a diverse collection of recipes, including vegetarian, non-vegetarian, vegan, gluten-free, and dessert options. Users can simply filter recipes based on cooking time, ingredients, skill level, and dietary restrictions, making it useful for both new and seasoned home cooks. The platform provides personalized recommendations based on user preferences and browsing history, which improves the discovery experience. Paid services include access to premium recipes, including unique chef-curated dishes and regional delicacies, as well as interactive tools such as meal planners, shopping lists, and nutritional breakdowns.

3. Why?

In today's fast-paced world, individuals are increasingly turning to the internet to discover new and simple recipes. Recipe Kavala Babu addresses this demand by providing a curated collection of premium and free recipes that are simple to follow, available from anywhere, and designed for home cooks who like experimenting with new cuisines. Whether customers want quick weekday dinners, traditional Indian feasts, or healthy selections that meet dietary requirements, the site provides a smooth experience.

Recipe Kavala Babu allows users to express their culinary creativity through features such as step-by-step directions, ingredient filters, and the ability to scale recipes based on serving size. The platform also includes time-saving features such as tailored recommendations, shopping lists, and meal planners, making cooking pleasurable and efficient for busy individuals and families. Furthermore, the inclusion of premium material enables users to access unique recipes from prominent chefs, boosting their home-cooking experience to restaurant-quality cuisine.

4. List of Supported Features

- **User Registration & Login**: Users can create an account and log in to access personalized features.
- **Recipe Browsing & Filtering**: Users can browse recipes by categories such as **Indian Curries**, **Desserts**, and **Biryani**. They can also filter by dietary preferences (e.g., vegetarian, vegan).
- **Save to Favorites**: Users can add their favorite recipes to their profiles for easy access.
- **Recipe Submission**: Users can submit their own recipes, complete with images and ingredient lists.
- **Premium Recipes**: Access to exclusive recipes through payment integration (Stripe/PayPal).
- **Push Notifications**: Notify users of trending recipes, offers, and new submissions.

5. List of Not Supported Features

- **Real-Time Recipe Collaboration**: Users cannot edit or collaborate on recipes simultaneously in real-time.
- **Video Tutorials**: The platform does not support video-based cooking tutorials at this stage.
- **In-App Messaging**: No support for direct messaging between users.

6. List of Future Planned Features

. Recipe Rating & Reviews

- **Feature**: Users will be able to leave reviews and rate recipes on a scale of 1 to 5 stars. This will help others discover the most popular and well-reviewed recipes.
- **Benefit**: This creates a community-driven feedback system, encouraging engagement.

2. Social Media Sharing

- **Feature**: Allow users to share their favorite recipes directly to social media platforms like Facebook, Instagram, and Twitter.
- **Benefit**: Increases user interaction and drives more traffic to the platform through social media visibility.

3. Shopping List Generation

- **Feature**: Users can generate a shopping list from the ingredients in selected recipes. They can also export this list as a PDF or sync it with grocery apps.
- **Benefit**: Makes it convenient for users to gather ingredients and shop more efficiently.

4. Step-by-Step Cooking Videos

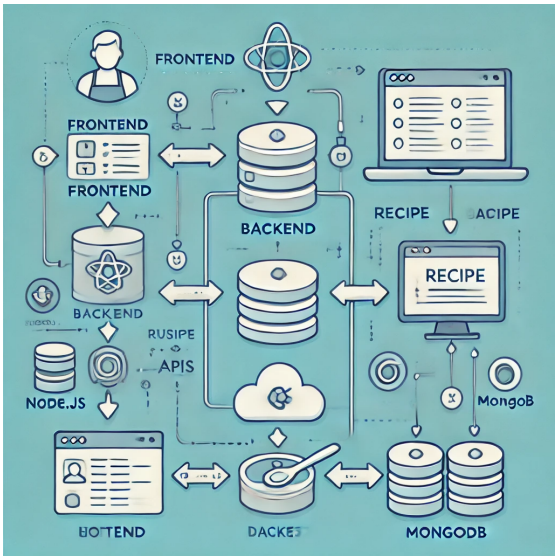
- **Feature**: Provide video tutorials alongside text-based recipes, with step-by-step cooking instructions.
- **Benefit**: Helps users follow along visually, making it easier for beginner cooks to replicate complex recipes.

7. How?

1. High-Level Diagram

The system architecture includes:

- **Frontend (React.js)**: The client-side part where users interact with the application.
- **Backend (Node.js + Express)**: Manages API requests and handles business logic.
- **Database (MongoDB)**: Stores user data, recipes, and payment transaction data.



2. List of Components/Modules

- **Frontend**: User interface (React.js), interactive components for browsing recipes and managing user profiles.
- **Backend**: REST API using Express.js to handle user authentication, recipe management, and payment processing.
- **Database**: MongoDB to store all user data, recipes, and transactional information.

3. Languages to be Used for Each Module

- **Frontend**: HTML, CSS, JavaScript (React.js)
- **Backend**: JavaScript (Node.js, Express.js)
- **Database**: MongoDB (NoSQL)

4. List of 3rd-Party/Open-Source Modules

- **Mongoose**: For MongoDB interaction in Node.js.
- **Stripe/PayPal SDK**: For payment integration.
- **Express.js**: Backend framework for REST API.
- **React Router**: For client-side routing.

1. Table of Licenses

- **React.js**: MIT License
- **Express.js**: MIT License
- **Mongoose**: MIT License
- **Stripe/PayPal SDK**: Commercial License

5. List of Any 3rd-Party Services/APIs

- **Stripe** (Paid): Used for handling payment processing for premium recipe access.
- **MongoDB Atlas** (Free/Paid): Cloud-hosted database service for managing the application's data.
- **Google Analytics** (Free): To track user behavior and improve user experience.

6. REST API Endpoints with Payloads

Here are some key API endpoints:

- **POST /api/users/signup**
  - Payload: `{ "name": "John Doe", "email": "john@example.com", "password": "password123" }`
- **POST /api/users/login**
  - Payload: `{ "email": "john@example.com", "password": "password123" }`
- **GET /api/recipes**
  - Payload: N/A (Fetches a list of recipes).
- **POST /api/recipes**
  - Payload: `{ "title": "Butter Chicken", "ingredients": ["Chicken", "Butter", "Tomatoes"], "instructions": "Grill chicken, make sauce..." }`
- **POST /api/payments**
  - Payload: `{ "userId": "12345", "amount": 9.99, "paymentMethod": "Stripe" }`

Build Steps

Frontend (React.js) Build for Netlify

1. **Navigate to the Frontend Directory**:

cd frontend

npm install

npm run build

Deploy to Netlify:

- Go to Netlify and log in to your account.
- Select "**New site from Git**" and link your GitHub repository.
- Choose the branch that contains the latest version of your React app (e.g., `week3`).
- Set the **Build Command** as `npm run build` and the **Publish Directory** as `build/`.
- Click on **Deploy Site**.

Install Steps

Frontend (React.js) Install Steps for Netlify

1. **Clone the Repository**: If you haven't already cloned your project from GitHub, run the following:

Clone the Repository: If you haven't already cloned your project from GitHub, run the following:

```
git clone https://github.com/SAISRITEJ-PALACHARLA/
comp5130.git
```

Navigate to the Frontend Directory:

cd frontend

Install Dependencies: Run the following command to install all React dependencies:

npm install

Build the Project: Create a production build of your frontend with the following:

```
npm run build
```

GitHub Information

- **Repository Name**: `comp5130`
- **Branching Model: Develop in the week3 branch, then merge into develop once the Week 3 deliverables are ready.**

9. References

- **React Documentation**: https://reactjs.org/docs/getting-started.html
- **Express.js Documentation**: https://expressjs.com/
- **MongoDB Documentation**: https://docs.mongodb.com/
- **Stripe API Documentation**: https://stripe.com/docs/api