**INTRODUCTION**

**Project Tittle:COOK BOOK**

**Team ID** : NM2025TMID46876

**Team Leader:**ABINAYA A

**Mail id:**abinayaa082007@gmail. Com

**Team Members**: 5

|  |  |
| --- | --- |
| Team members | Mail id |
| Abinaya A | [abinayamanjula2007@gmail.com](mailto:abinayamanjula2007@gmail.com) |
| Aruna M | [appaaruna25092007@gmail.com](mailto:appaaruna25092007@gmail.com) |
| Elakkiya D | [elakkiya112006@gmail.com](mailto:elakkiya112006@gmail.com) |
| Dheersa L | [ldheersa55@gmail.com](mailto:ldheersa55@gmail.com) |

1. **Project Overview:**

# Purpose:

The project aims to provide users with a virtual cooking assistant that helps them discover, organize, and follow recipes easily. It will make meal planning more convenient and interactive.

# Features:

* Recipe search and filtering
* Step-by-step cooking instructions
* Voice assistance for hands-free cooking
* Personalized recipe recommendations
* Save and manage favorite recipes

## Architecture

* Frontend: React.js with Bootstrap and Material UI
* Backend: Node.js and Express.js managing server logic and API endpoints
* Database: MongoDB stores user data, project information, applications, and chat messages

# Component Structure:

* Header (Navigation)
* Recipe List (Displays recipes)
* Recipe Details (Shows full recipe with steps C ingredients)
* Search/Filter Bar
* Favorites Section
* Voice Assistant Component

# State Management:

Redux or Context API for managing recipes, favorites, and user preferences.

# Routing:

* React Router for navigating between Home, Recipes, Favorites, and About pages.

## Setup Instructions Prerequisites:

* Node.js
* MongoDB
* Git

-React.js

-Express.js – Mongoose – Visual Studio Code

# Installation:

* Git clone [repository link]
* Cd cookbook-virtual-assistant
* Npm install
* Npm start

## Folder Structure

SB-Works/

|-- client/ # React frontend

| components/ L pages/

| server/ # Node.js backend

| routes/

| models/

| controllers/

# Client:

* /components → Header, RecipeCard, SearchBar, etc.
* /pages → Home.js, RecipeDetails.js, Favorites.js
* /assets → Images, icons
* /hooks → Custom hooks

# Utilities:

* Helper functions for API calls, formatting ingredients, etc.

1. **Running the Application**

# Frontend:

* Npm start
* Runs the app in development mode at <http://localhost:3000/>

1. **API Documentation**

* User:
  + /api/user/register
  + /api/user/login
* Projects:
  + /api/projects/create
  + /api/projects/:id • Applications: /api/apply
* Chats:
  + /api/chat/send
  + /api/chat/:userId

1. **Component Documentation**

# Key Components:

* RecipeCard: Displays recipe preview
* RecipeDetails: Shows full recipe steps
* SearchBar: Filters recipes by name/ingredient

# Reusable Components:

* Button, Modal, Loader, Card

1. **State Management**

# Global State:

Recipes, favorites, and user preferences stored in Redux/Context API.

# Local State:

Input fields, modal visibility, and step-by-step instructions handling.

## G. User Interface

* Screenshots or GIFs of:
* Home page with search bar
* Recipe list grid
* Detailed recipe view
* Favorites page

## Styling

* + CSS Frameworks/Libraries: Tailwind CSS or Styled- Components.
  + Theming: Light/Dark mode toggle for better UX.

## Testing

* Testing SStrateg
* Unit testing with Jest
* Component testing with React Testing Library

# Code Coverage:

Ensure all recipe, search, and favorite functionalities are tested.

## Screenshots or Demo

* Add screenshots of the UI or provide a demo link (e.g., deployed on Netlify/Vercel).

## Known Issues

* Some recipes may lack nutritional info.
* Voice assistant may not support all languages initially.

## Future Enhancements

* Add grocery list generator
* Support multiple languages and regional cuisines
* AI-based meal planner
* Offline recipe access.

1. **screenshots**



