**Project Documentation**

**Cook book:your virtual kitchen assistant**

# 1. Introduction

* **Project Title:** Cook book:your virtual kitchen assistant
* **Team ID** : NM2025TMID37980
* **Team Leader:**

**Name:**SRILEKHA

**ID:**srilekajayamani@gmail.com

* **Team Members:**

Name:SHARMILA

ID: Sharmilamaha60@gmail.com

Name:ARTHI

ID:

Name:GEETHA

ID:geethas170207@gmail.com

# 2. Project Overview

* **Purpose:** Purpose: The Virtual Kitchen Assistant is designed to help users manage cooking tasks efficiently by providing personalized recipe suggestions, ingredient management, and real-time cooking guidance. It aims to simplify meal preparation, reduce food waste, and enhance the cooking experience.
* **Features:**
  + Personalized recipe recommendations based on available ingredients
  + - Voice-guided step-by-step cooking instructions
  + - Pantry and grocery management system
  + - Nutritional value and calorie tracking

# - Multi-language support for diverse user.

**3. Architecture**

* **Frontend:** React.js with Bootstrap and Material UI
* **Backend:** Node.js and Express.js for server logic and API handling
* **Database:** MongoDB to store user profiles, pantry items, recipes, and procedures

**SETUP INSTRUCTION**

* **Prerequisites:**
  + Node.js
  + MongoDB
  + Git
  + React.js
  + Express.js **–** Mongoose **–** Visual Studio Code
* **Installation Steps:**

Clone the repository

git clone <repo-link>

Install client dependencies

cd client

npm install

Install server dependencies

cd ../serve

npm install

# 5. Folder Structure

Virtual-Kitchen-Assistant

**||-client/** -This folder contains the React.js frontend code.

**| |-- components/ -**This directory holds reusable UI Components

**| |-- pages/**-This directory containts the main page components of the applications

**|-- server/ -**This folder contains the Node.js backend code.

**|-- routes/**-This directly defines the API endpoints for the application.

**|-- models/**-This directly contains the database schems

**|-- controllers/-**It contains function for creating,reading,updating and deleting data.

# |-- utils/

# 6. Running the Application

* **Frontend:**

cd client npm start

• **Backend:**

cd server npm start

* **Access:** Visit http://localhost:3000

# 7. API Documentation

* **User:**
  + User Authentication:
  + - POST /api/user/register
  + - POST /api/user/login
  + Recipes:
  + - GET /api/recipes/suggest
  + - GET /api/recipes/:id
  + Pantry Management:
  + - POST /api/pantry/add
  + - GET /api/pantry/list
  + Cooking Assistance:
  + - POST /api/cooking/guide

# - GET /api/cooking/tips

# 8. Authentication

JWT-based authentication for secure login

# Middleware protects user-specific data (pantry, preferences, saved recipes

# 9. User Interface

- Landing Page with recipe highlights

- Dashboard showing personalized meal plans

- Pantry Management Page

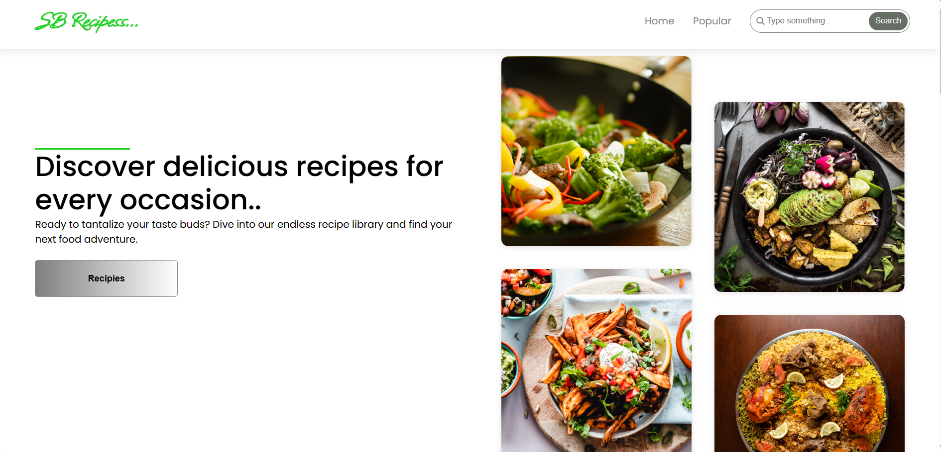
- Cooking Assistant Interface with voice guidance

# - Admin Panel for recipe and content management

# 10. Testing

* Manual testing during development milestones
* Tools: Postman, Chrome Dev Tools, Jest for unit testing

**Screenshot or Demo**

****

# 11.Future Enhancements

-AI-based meal planning with health goals

- Smart kitchen appliance integration (IoT)

-Augmented Reality (AR) guided cooking

- Community recipe sharing platform