`

` Cookbook Virtual Kitchen Assistant

1. Introduction:

Project Title: Cookbook Virtual Kitchen Assistant

Team members:

Fronted Developer-Srisathya R

UI/UX Designer-Vithya D, Poornima T

React Developer- Uma A.R

Project Manager-Priya V

2. Project Overview:

Purpose:

The Virtual Kitchen Assistant is an interactive web application designed to help users discover, manage, and cook recipes efficiently. It includes features like voice assistance, personalized recipe recommendations, and smart pantry management.

Features:

Recipe Search And Filtering

Step-By-Step Cooking Instuction

Virtual Pantry With Ingredient Tracking

Meal Planning And Scheduling

Voice Command Support For Hands-Free Cooking

3.Architecture:

Component Structure:

App: Root Component

Navbar: Navigation And Links

RecipeList: Displays a list of recipe cards

RecipeDetail: Detailed view of a selected recipe

Pantry: Shows Available ingredients

Planner: Meal planning interface

Voice Assistant:Voice Command Interface

State Management:

Using Context API for global state (e.g., user session, pantry items) and useState for local component states.

Routing:

Handled with React Router v6,routes include:

/:Home

/recipes:All recipes

/recipes/:id:Single recipe detail

/pantry :pantry manager

/planner:meal planner

4. Setup Instructions:

Prerequisites:

Node.jsv16+

Npm or yarn

git

Installation:

1. Clone the repo:

git clone https://github.com/yourusername/cookbook-virtual-kitchen.git

2. Navigate to client directory:

cd cookbook-virtual-kitchen/client

3. Install dependencies:

npm install

4. Create .env file for environment variables:

REACT_APP_API_URL=https://api.yourbackend.com

5. Folder Structure

cookbook-app/

├— public/

├— src/

├— components/

```
        ├── RecipeCard.js
        ├── SearchBar.js
        │   └── Favorites.js
        ├── pages/
        ├── Home.js
        ├── Search.js
        │   └── Profile.js
        ├── state/
        │   └── context.js
        └── main.css
        │   ├── utils/
        │   └── api.js
        │   └── App.js
    ├── package.json
        └── README.md
```

Utilities:

hooks/useVoiceCommands.js: Handles voice input

utils/api.js: API service wrapper

helpers/formatTime.js: Helper for time formatting

6. Running the Application:

Frontend:

cd client:

npm start

This starts the app on http://localhost:3000.

7. Component Documentation:

Key Components:

RecipeCard:

Displays summary info (props: title, image, prepTime)

VoiceAssistant:

Manages voice input (props: none, uses context)

Reusable Components:

Button:

Custom styled button (props: label, onClick, variant)

Modal:

Pantry items:

Here's a classic Chicken Biryani recipe that balances rich flavor, aroma, and spice. You can adapt it for mutton, beef, or vegetables as well.

🍗 Chicken Biryani Recipe

🕒 Prep Time: 30 mins

🕒 Cook Time: 45 mins

🍽️ Serves: 4–5

🧂 Ingredients

For Marination:

500g chicken (bone-in, cut into medium pieces)

1 cup yogurt

1 tbsp ginger-garlic paste

1 tsp red chili powder

½ tsp turmeric powder

1 tsp garam masala

Salt to taste

Juice of ½ lemon

Fresh chopped coriander & mint leaves (a handful each)

For Rice:

2 cups basmati rice (soaked 30 mins)

1 bay leaf

4–5 cloves

2–3 cardamom pods

1-inch cinnamon stick

Salt to taste

For Biryani:

2 large onions (thinly sliced)

2 tomatoes (chopped)

1 tsp cumin seeds

2 green chilies (slit)

1 tsp biryani masala (optional)

¼ cup oil or ghee

Saffron strands soaked in 2 tbsp warm milk (or use a pinch of food color)

Fried onions (for garnish)

Mint and coriander leaves

🔪 Instructions:

1. Marinate the Chicken:

Mix all marination ingredients in a bowl.

Add chicken and coat well.

Cover and marinate for at least 1 hour (overnight is best).

2. Cook the Rice:

Boil water in a large pot.

Add whole spices and salt.

Add soaked rice and cook until 70% cooked (grains should still be firm).

Drain and set aside.

3. Prepare the Biryani Masala:

Heat oil/ghee in a deep pan.

Fry onions until golden brown. Remove half for garnish.

In the same pan, add cumin seeds, green chilies, and chopped tomatoes.

Cook until tomatoes are soft.

Add marinated chicken. Cook covered on medium heat for 10–15 mins until the chicken is cooked and oil separates.

4. Layer the Biryani:

Lower the heat.

Layer half of the partially cooked rice over the chicken.

Sprinkle some mint, coriander, fried onions, and saffron milk.

Repeat with remaining rice and toppings.

5. Dum (Steam) Cooking:

Cover tightly with a lid or seal with dough.

Cook on low heat for 20–25 minutes.

Alternatively, place a heavy object on the lid and cook on a griddle (tawa) to prevent burning.

🍽 Serve:

Gently mix before serving.

Serve hot with raita, salad, or boiled eggs.

8. State Management:

Global State:

Context API Handle:

Auth state:

Pantry items:

Current voice command:

Local State:

Handled via useState for form inputs, search filters, modal visibility

9. User Interface:

Screenshots (replace with your actual images or include links):

Home page with top recipes

Detailed recipe view

Pantry management interface

Meal planner calendar

Voice command interface

(Add screenshots or GIFs here)

10. Styling:

CSS Frameworks/Libraries:

TailwindCSS for utility-first styling

Emotion for component-level styles

Theming:

Supports light and dark themes

Custom themes settings stored in theme.js

11. Testing:

Testing Strategy:

Unit tests: Jest + React Testing Library

Integration:

Testing interactions like adding a recipe to planner

End-to-End:

Cypress (planned for future)

Code Coverage:

Jest configured with coverage reporting:

npm test -- --coverage

12. Screenshots or Demo:

🔗 Live Demohttps://virtualkitchenassistant.vercel.app

Screenshots:

Home Page

Recipe Detail:

Pantry Management:

Meal Planner:

(Insert images or link to external demo/gallery)

13. Known Issues:

Voice commands may not be fully accurate in noisy environments

Limited browser support for speech recognition (only Chrome fully supported)

Pantry items not syncing in real-time

14. Future Enhancements:

AI-based recipe suggestions based on pantry contents

Integration w

State Management:

Using Context API for global state (e.g., user session, pantry items) and useState for local component states.

Routing:

Handled with React Router v6, routes include:

/: Home

/recipes: All recipes

/recipes/:id: Single recipe detail

/pantry: Pantry manager

/planner: Meal planner

---

4. Setup Instructions

Prerequisites:

Node.js v16+

npm or yarn

Git

Installation:

1. Clone the repo:

git clone https://github.com/yourusername/cookbook-virtual-kitchen.git

2. Navigate to client directory:

cd cookbook-virtual-kitchen/client

3. Install dependencies:

npm install

4. Create .env file for environment variables:

REACT_APP_API_URL=https://api.yourbackend.com

---

5. Folder Structure

📁 client/

```
├── assets/         # Images, icons, etc.
├── components/     # Reusable UI components
├── pages/          # Page-level components
├── hooks/          # Custom React hooks
├── context/        # Global state providers
├── styles/         # Global styles and themes
├── App.js
├── index.js
```

Utilities:

hooks/useVoiceCommands.js: Handles voice input

utils/api.js: API service wrapper

helpers/formatTime.js: Helper for time formatting

---

6. Running the Application

Frontend:

cd client

npm start

This starts the app on http://localhost:3000.

---

7. Component Documentation

Key Components:

RecipeCard: Displays summary info (props: title, image, prepTime)

VoiceAssistant: Manages voice input (props: none, uses context)

Reusable Components:

Button: Custom styled button (props: label, onClick, variant)

Modal: Popup for recipe steps or pantry items

---

8. State Management

Global State:

Context API handles:

Auth state

Pantry items

Current voice command

Local State:

Handled via useState for form inputs, search filters, modal visibility

---

9. User Interface

Screenshots (replace with your actual images or include links):

Home page with top recipes

Detailed recipe view

Pantry management interface

Meal planner calendar

Voice command interface

(Add screenshots or GIFs here)

---

10. Styling

CSS Frameworks/Libraries:

TailwindCSS for utility-first styling

Emotion for component-level styles

Theming:

Supports light and dark themes

Custom theme settings stored in theme.js

---

11. Testing

Testing Strategy:

Unit tests: Jest + React Testing Library

Integration: Testing interactions like adding a recipe to planner

End-to-End: Cypress (planned for future)

Code Coverage:

Jest configured with coverage reporting:

npm test -- --coverage

---

12. Screenshots or Demo

🔗 Live Demo: https://virtualkitchenassistant.vercel.app

Screenshots:

Home Page

Recipe Detail

Pantry Management

Meal Planner

(Insert images or link to external demo/gallery)

---

13. Known Issues

Voice commands may not be fully accurate in noisy environments

Limited browser support for speech recognition (only Chrome fully supported)

Pantry items not syncing in real-time

---

14. Future Enhancements

AI-based recipe suggestions based on pantry contents

Integration w