Frontend Development with React.js

Project Documentation format

Cookbook: Your Virtual Kitchen Assistant

Introduction

Project Title:

Cookbook: Your Virtual Kitchen Assistant

Team Leader: SHOBANA R
Team Member: VIJI P H
Team Member: SHIVAANI P V
Team Member: LOGESHWARI B
Team Member: UMA S

Project Overview

Purpose:

Cookbook is a virtual kitchen assistant designed to help users discover, organize, and manage recipes efficiently. The goal is to provide a seamless user experience for meal planning, grocery list management, and step-by-step cooking guidance.

Features:

Search and browse a vast collection of recipes.

Save favorite recipes to personal collections.

Generate shopping lists from selected recipes.

Step-by-step cooking instructions with a guided mode.

User account management with preferences and history.

Architecture

Component Structure:

App.js - Root component.

NavigationBar.js - Handles app navigation.

RecipeList.js - Displays a list of recipes.

RecipeDetail.js - Shows detailed recipe information.

ShoppingList.js - Generates grocery lists.

UserProfile.js - Manages user preferences and saved recipes.

State Management:

Context API is used for global state management to share user preferences, saved recipes, and shopping lists across the app.

useState and useReducer for local component state.

Routing:

Implemented using react-router-dom.

Routes:

/ - Home Page

/recipes - Recipe Listing

/recipe/:id - Recipe Details

/shopping-list - Shopping List Page

/profile - User Profile

Setup Instructions

Prerequisites:

Node.js (>= 14.x)

npm or yarn

Installation:

Navigate to the client directory:

```
cd cookbook/client
```

Install dependencies:

```
npm install
```

Start the development server:

```
npm start
```

Folder Structure

Client:

components/ - Reusable UI components.

pages/ - Main application pages.

assets/ - Images, icons, and styles.

hooks/ - Custom React hooks.

context/ - Context API providers.

services/ - API calls and data fetching logic.

Utilities:

helpers.js - Helper functions for data formatting.

auth.js - Authentication utilities.

api.js - Axios setup for API requests.

Running the Application

To start the frontend server locally, use:

```
npm start
```

Component Documentation

Key Components:

RecipeList - Displays all available recipes.

RecipeDetail - Shows full details of a selected recipe.

ShoppingList - Helps users generate grocery lists.

Reusable Components:

Button.js - Customizable buttons.

InputField.js - Styled input fields.

Modal.js - Pop-up modal component.

State Management

Global State:

Managed with Context API for storing user data, saved recipes, and shopping lists.

Local State:

Managed with useState and useReducer for form inputs and interactive elements.

User Interface

Screenshots showcasing:

Recipe Listing Page

Recipe Details Page

Shopping List Feature

User Profile Page

Styling

CSS Frameworks/Libraries:

Tailwind CSS for styling.

Styled-Components for component-level styling.

Theming:

Dark mode and light mode options.

Custom design system implemented for a consistent UI.

Screenshot or Demo:

https://drive.google.com/file/d/1B0Df0ZqlLs6d-JZ6DWwZrEaGLRIyai0x/view?usp=drivesdk