Project Title: Cookbook: Your Virtual Kitchen Assistant

* Team ID: NM2025TMID39842
* Team Leader: POONGAVANAM P & ID: ppoongavanam025@gmail.com
* Team Members:
* KEERTHIKA P & ID: kkeerthi70803@gmail.com
* BHUVANESHWARI R & ID: bhuvaneshwariraji32@gmail.com
* YAMUNA G & ID: yamunasathiya2006@gmail.com

COOK BOOK

1. Introduction

* Food has always been more than just sustenance—it's an experience, a celebration, and a way to connect. In this cookbook, I invite you to embark on a culinary journey that will inspire creativity, nurture your love for flavours, and remind you of the simple joy of cooking.
* Whether you're a seasoned chef or a beginner in the kitchen, these recipes are designed to bring out the best in every cook and every meal.
* Each dish in this book is a reflection of my passion for [type of cuisine or cooking style], with a focus on [fresh, seasonal ingredients, comfort food, quick meals, etc.].
* Along with easy-to-follow instructions and a few pro tips, my hope is that these pages become a go-to resource in your kitchen, no matter the occasion.

1. Project overview

* Purpose: To provide readers with the collection of recipes and instructions for food preparation, acting as a guide for culinary techniques, ingredient information, and cultural and contexts of dishes.
* Features: Well-written recipes with clear instructions, ingredient lists in order of use, and step-by-step processes, often accompanied by food photography and serving suggestions.

1. Architecture:

* Frontend:
* **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.
* Backend:
* **Node.js**: Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.
* Database:
* MongoDB is a NoSQL database that stores data in flexible, JSON- like documents, allowing for dynamic schema changes.

1. Setup Instructions:

* Prerequisites: **Node.js and npm**:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

* Download: <https://nodejs.org/en/download/>
* Installation instructions: <https://nodejs.org/en/download/package-manager/>
* **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

cd my-react-app

* **Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

 • Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

* **Installation steps**
* Navigate into the cloned repository directory and install libraries:

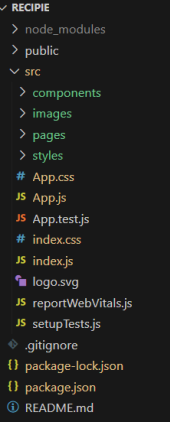
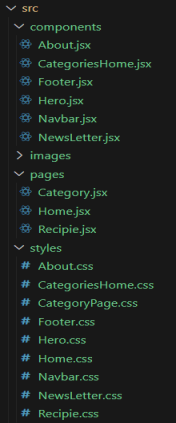
cd fitness-app-react

npm install

* **Start the Development Server**:
* To start the development server, execute the following command:

npm start

1. Folder structure

1. Running the Application

* Running the React App:

With the React app created, you can now start the development server and see your React application in action.

* Start the development server:

npm start

This command launches the development server, and you can access your React app at [http://localhost:3000](about:blank) in your web browser.

1. API Documentation

* USERS
  + POST/users
  + GET/users/{id}
  + PUT/users/{id}
* PROJECTS
  + POST/projects
  + GET/users/{id}/projects
  + PUT/projects/{id}
* APPLICATION
  + POST/applications
  + GET/projects /{id}/ applications
  + PUT/users/{id}
* CHATS
  + POSTS/chats
  + GET/projects/{id}/chats

1. Authentication

* JWT-based authentication for secure login.
* Middleware to protect private routes (like adding/editing recipes.)

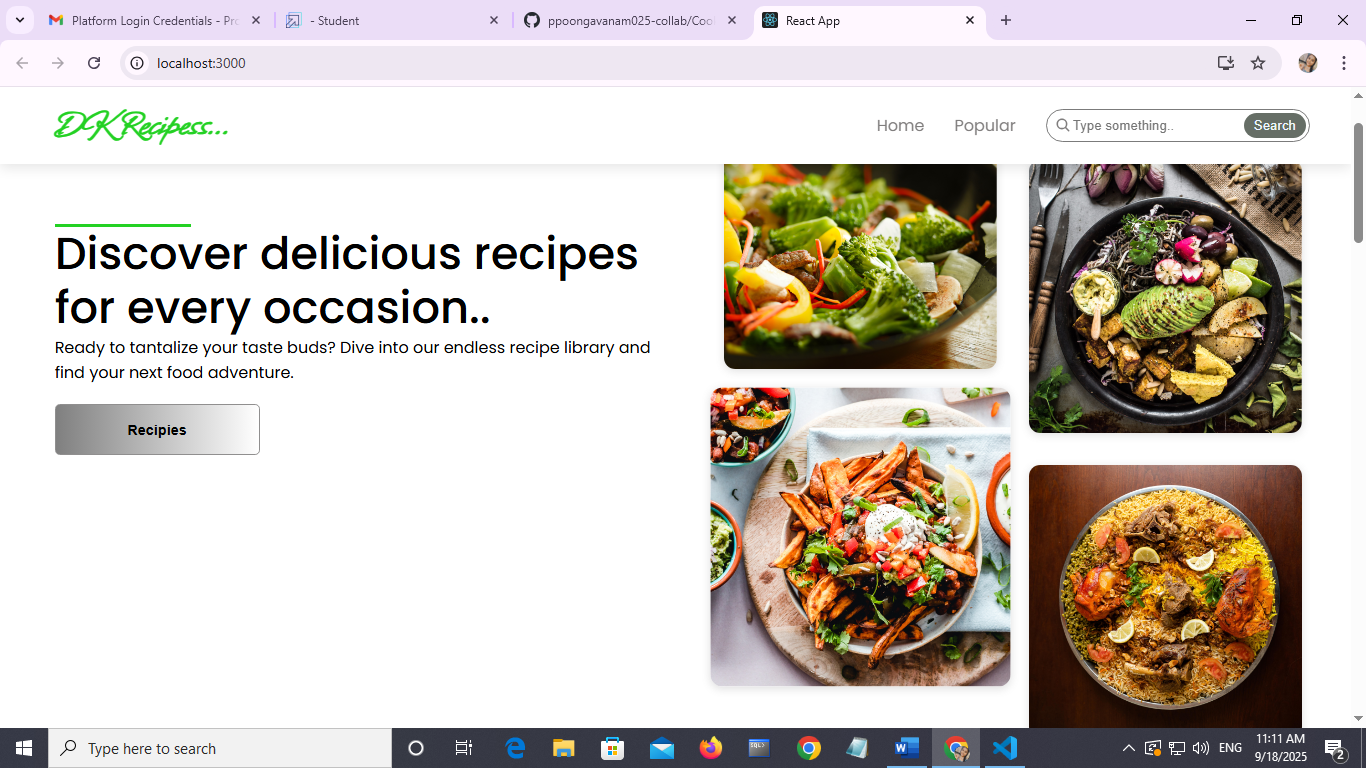
1. User Interface

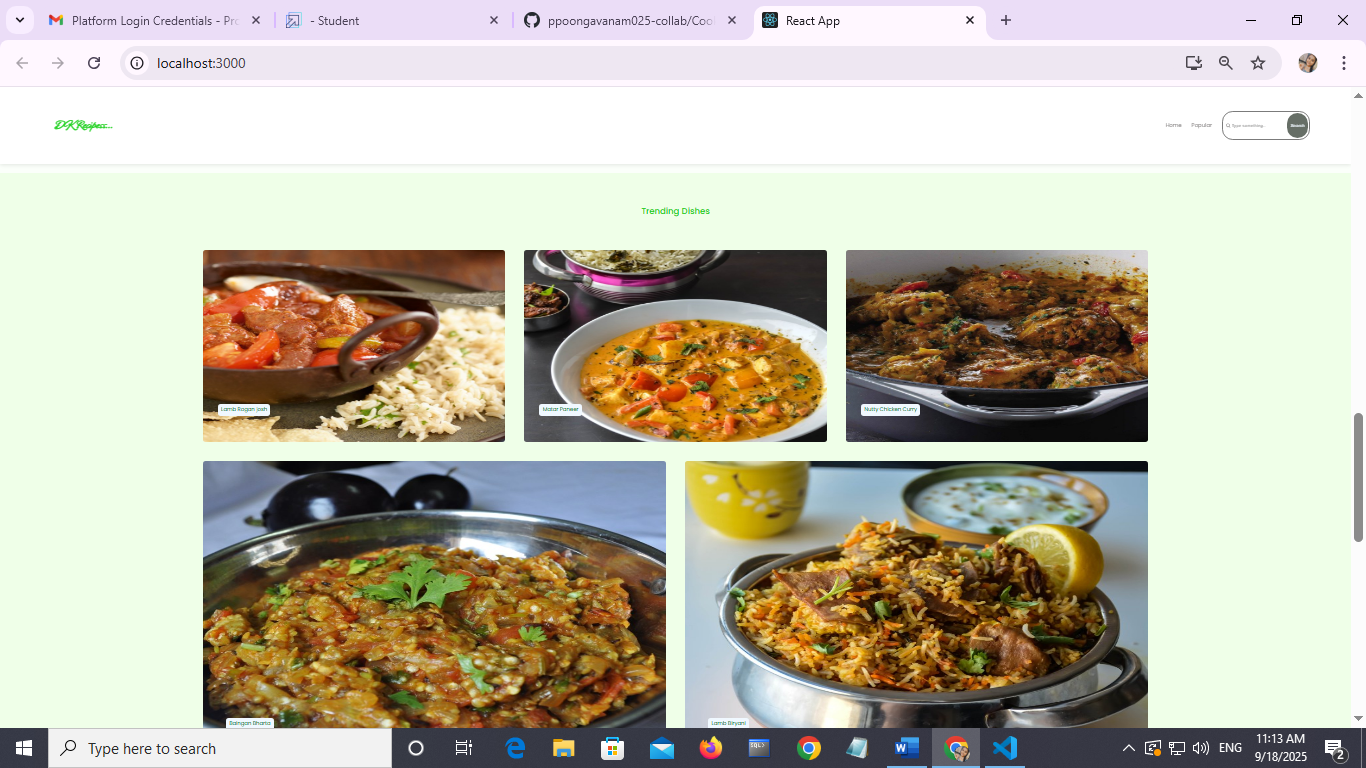
* Landing page (popular recipes, categories)
* User Dashboard (saved recipes, contributions)
* Amin Panel (manage users/recipes)
* Recipes Detail page (ingredients, steps, ratings, comments)

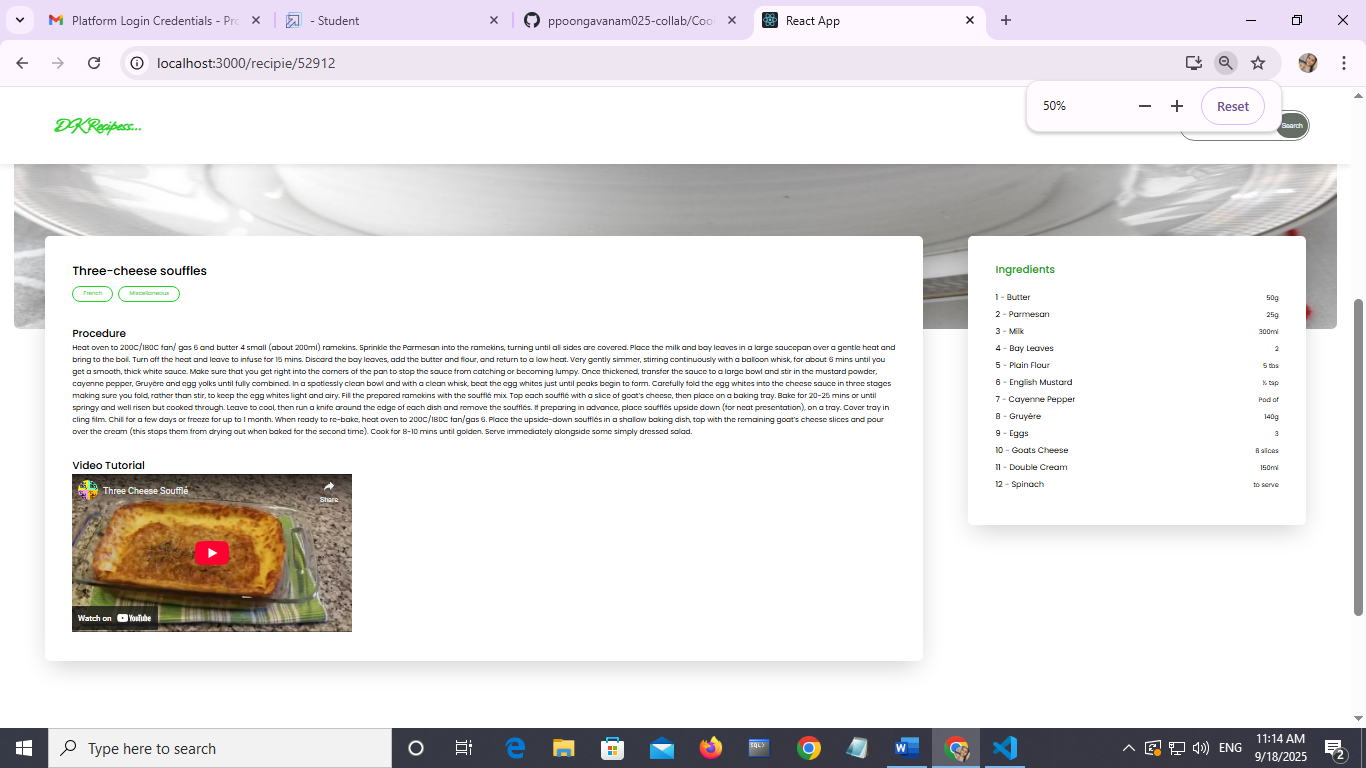
10. Testing

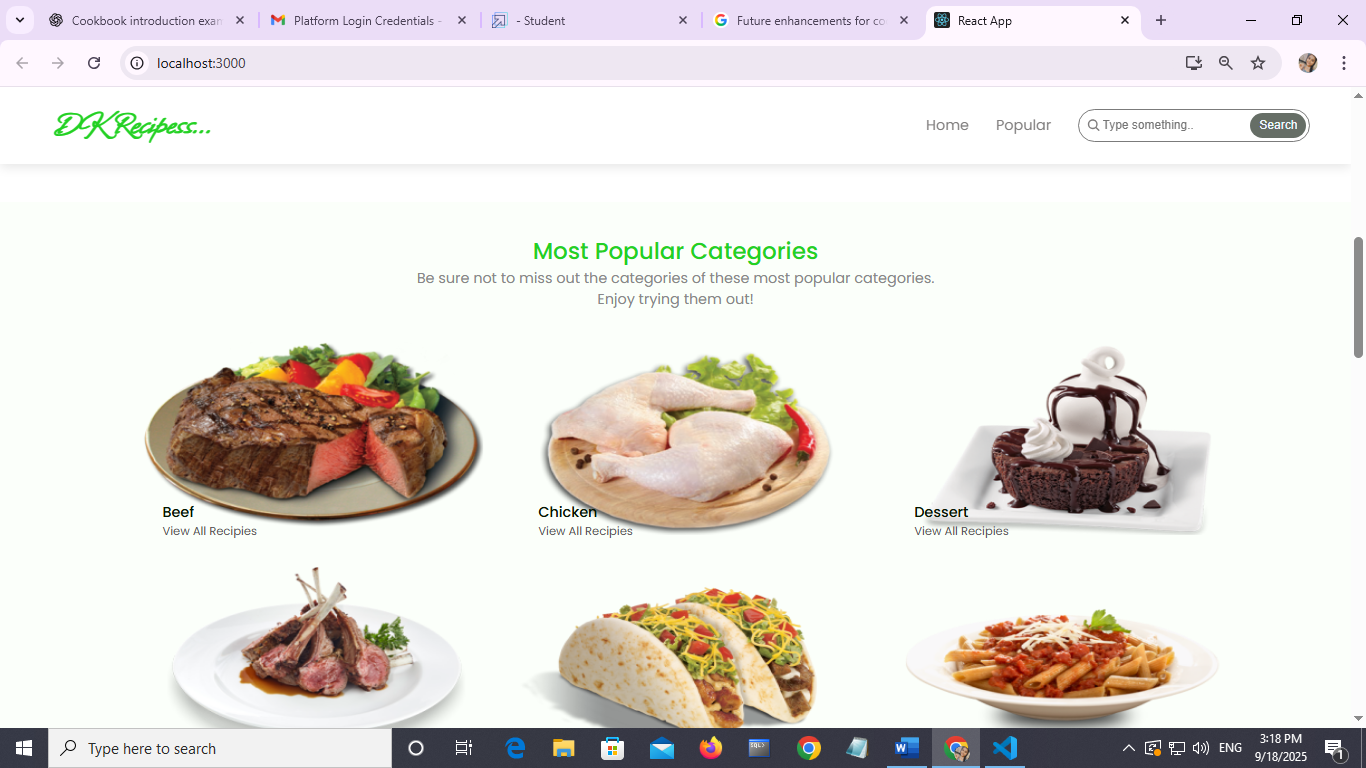
* Manual Testing
  + Tested API endpoints during each milestone.
  + Verified secure login/logout flow with JWT authentication.
  + Checked user, project, application, and chat functionalities.
  + Ensured middleware correctly products private routes.
* Tools Used
  + Postman 🡪 For API request/response validation.
  + Chrome Dev Tools 🡪For debugging frontend UI and API integration.

11. Screenshots or Demo









12. Known Issues

* Unreliable or poorly written recipes, such as
* missing ingredients,
* confusing methods, or incorrect measurements,
* as well as poor photography that doesn't match the actual results,
* difficulty finding specific recipes due to inadequate indexing or organization,
* A lack of helpful context like glossaries or explanations for unfamiliar techniques or ingredients.

13. Future Enhancements

* Interactive elements and high-quality visuals, will likely attract a wider audience.
* Integration of sustainable and recyclable materials in production aligns with growing environmental awareness, potentially expanding the market reach