**REPORT**

**RECIPE IDEAS – A Full-Stack Web Application**

**1. Introduction**

Project Title: Recipe Ideas Web Application Using Full Stack Technologies  
Project Objective:  
The primary objective of this project is to develop an interactive and user-centric web application that allows users to discover, share, and manage cooking recipes. The system facilitates user authentication, recipe categorization, and personalized features such as marking favourite dishes. The platform aims to provide a practical example of full-stack development using Python, Flask, SQLite, HTML, CSS, and related technologies.

**2. Purpose of the System**

The Recipe Ideas Web Application is designed to address the need for an organized, user-friendly platform where users can access a wide variety of recipes, contribute their own, and maintain a curated list of favourites. This system demonstrates essential CRUD operations, user management, and category filtering in a professional web development context.

**3. System Architecture Overview**

**Frontend Layer:**

* Languages and Tools: HTML5, Tailwind CSS, Jinja2 Templates
* Role: Responsible for rendering dynamic content, collecting user inputs through forms, and presenting data retrieved from the backend.

**Backend Layer:**

* Framework: Python Flask
* Libraries: Flask-Login, Werkzeug Security
* Role: Manages business logic, handles user sessions, processes form, interacts with the database, and controls application flow.

**Database Layer:**

* Technology: SQLite3
* Role: Stores persistent information including user credentials and recipe data.

**4. Functional Requirements**

* User Authentication: Secure registration, login, and logout functionalities.
* Recipe Browsing: View all available recipes with options to filter by category or search by ingredients.
* Recipe Management: Users can add new recipes through a structured form.
* Favourites Management: Users can mark recipes as favourites and view them separately.

**5. Technology Stack in Detail**

| Component | Technology | Purpose |
| --- | --- | --- |
| Frontend | HTML, CSS (Tailwind) | UI/UX Design and Layout |
| Backend | Python Flask | Web Framework for Request Handling |
| User Session | Flask-Login | User Authentication and Session Management |
| Password Security | Werkzeug | Password Hashing and Verification |
| Database | SQLite3 | Storing Users and Recipes Data |
|  |  |  |

**6. System Setup Instructions**

Prerequisites

* Python 3.9 or higher
* pip (Python package manager)
* SQLite3 installed on the system

Installation Procedure

1. Clone the Repository:

>git clone https://github.com/yourusername/recipe-ideas-app.git

>cd recipe-ideas-app

1. Create and Activate Virtual Environment:

>python -m venv venv

source venv/bin/activate # For Windows: venv\Scripts\activate

1. Install Required Packages:

>pip install -r requirements.txt

1. Database Initialization:

>python db\_setup.py

1. Run the Application:

>python app.py

1. Accessing the Application:  
    >Navigate to http://127.0.0.1:5000 in a web browser.

**7. Database Schema Explanation**

Users Table

* id: Integer, Primary Key
* username: Text, Unique, Not Null
* password\_hash: Text, Not Null

Recipes Table

* id: Integer, Primary Key
* title: Text, Not Null
* ingredients: Text, Not Null
* instructions: Text, Not Null
* category: Text, Not Null

**8. Folder Structure Overview**

recipe-ideas-app/

│-- app.py

│-- db\_setup.py

│-- templates/

├ home.html

├ recipes.html

├ add\_recipe.html

├ register.html

├ login.html

├ favourites.html

│-- README.md

**9. Usage Instructions**

User Interaction Flow

* Users visit the home page.
* Registration is required for new users.
* After login, users can browse, search, and filter recipes.
* Authenticated users can add new recipes.
* Favourite management is available for logged-in users.

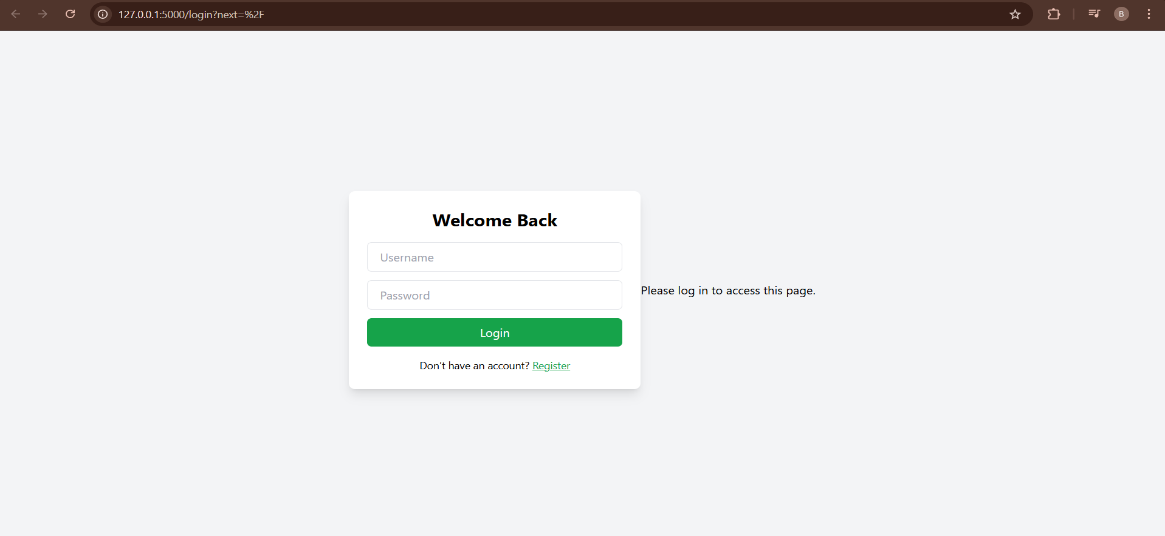
****

Fig.1 Login page

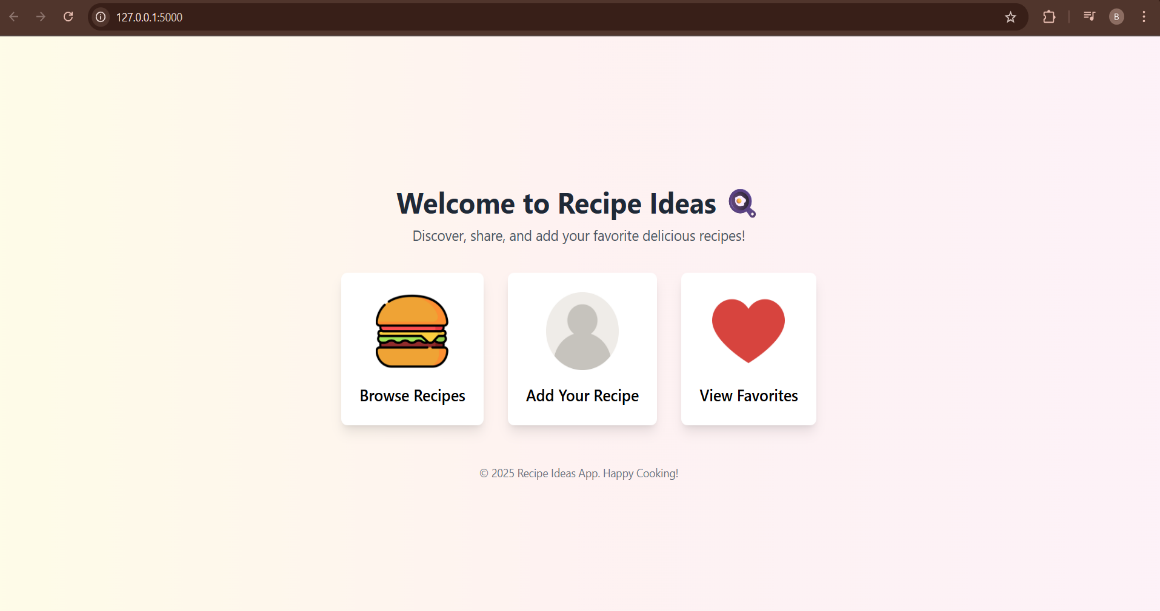
****

Fig.2 Home page

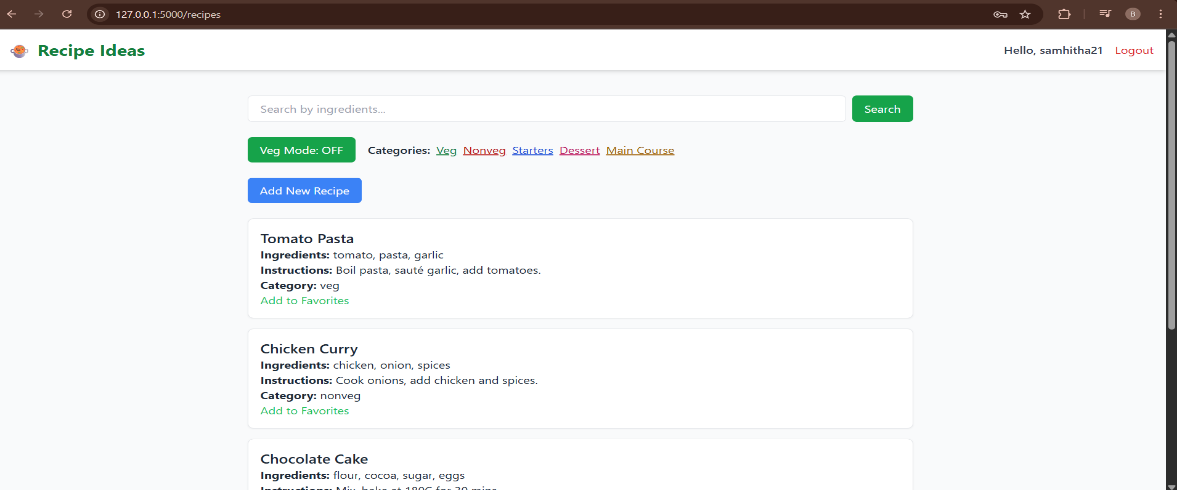
****

Fig.3 List of recipes

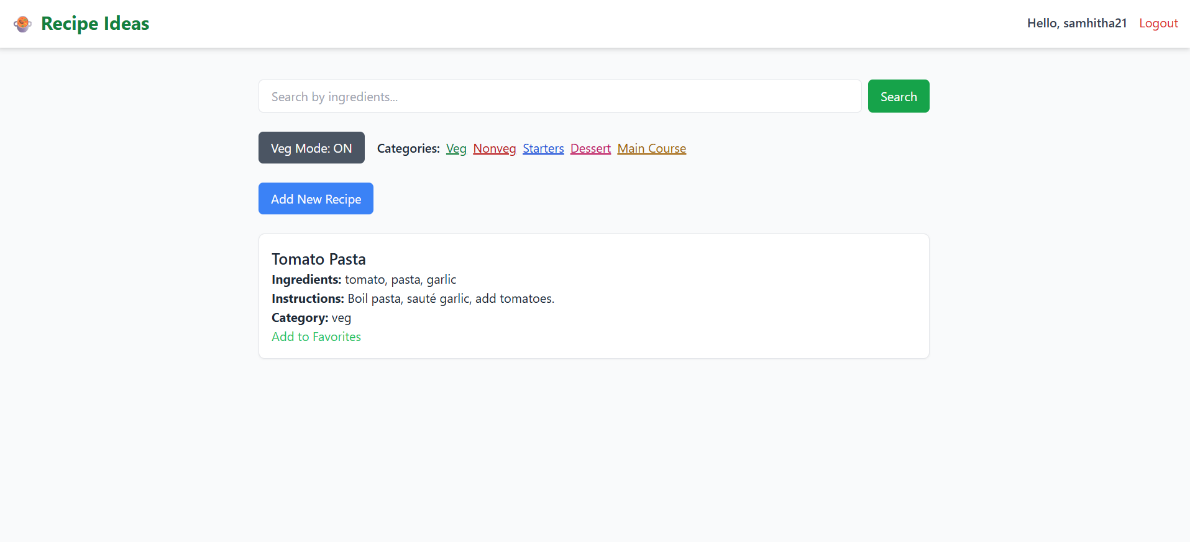
****

Fig.4 Veg mode ON

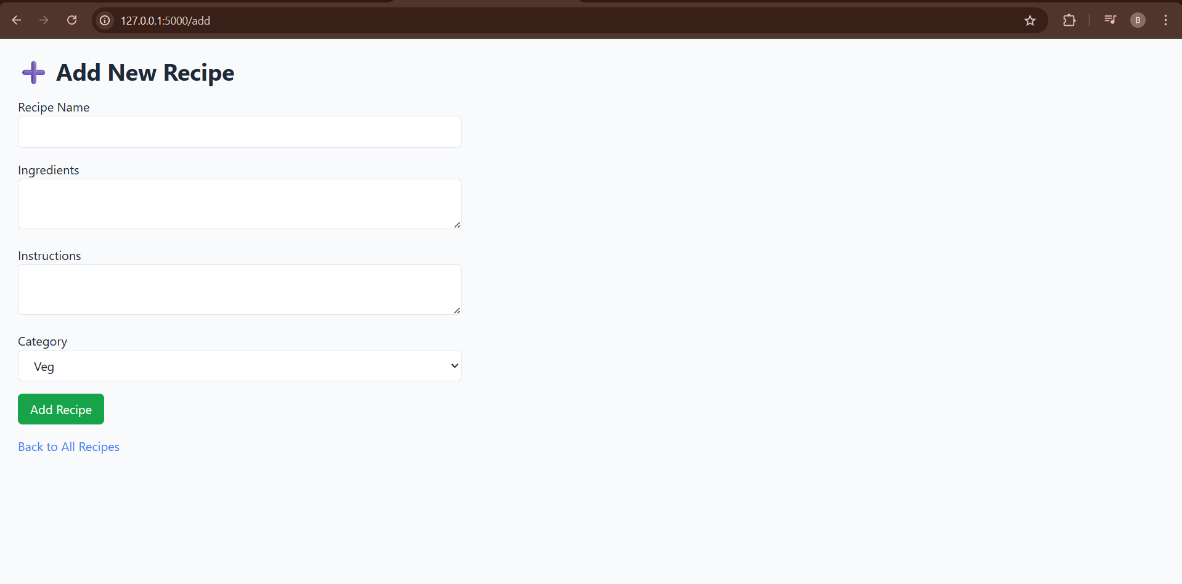
****

Fig.5 Add new recipe page

**10. Limitations and Future Work**

* Image Upload Feature: Currently not implemented but can be added using Flask extensions.
* Cloud Deployment: Transition from local SQLite3 to cloud-based databases for scalability.
* User Profile Enhancements: Expanding user management with profile editing and history tracking.

**11. Conclusion**

This Recipe Ideas Web Application is a comprehensive full-stack project aimed at demonstrating practical web development skills. It is suitable for educational and small-scale production use cases, providing a foundational structure that can be expanded with additional features.