Project Report on

Recipe Management System

IEM

श्रद्धावान लभते ज्ञानम्

INSTITUTE OF ENGINEERING & MANAGEMENT

Good Education, Good Jobs

Submitted by

SWASTIK KHAN

AKASH KUMAR MISHRA

Bachelors of Technology

(CSE-AIML)

Batch 2022-2026

ABSTRACT

This project presents a recipe management system developed using SQLite3 for database management and Streamlit for the user interface. The system allows users to input recipe details, including the name, ingredients, and method, which are then stored in a SQLite3 database. The Streamlit frontend provides a user-friendly interface for adding, viewing, and potentially searching recipes.

This application will be designed with the help of SQLite3 as back-end and Html, CSS & Streamlit as front-end, and the expected outcome for this project is that the user can store food recipes and access it whenever necessary.

ACKNOWLEDGEMENT

We would like to express my special thanks to “Dr. Deepshubhra Guha Roy”, for his able guidance and support in completing our project.

Swastik Khan
(12022002016057)
B.Tech
V-Semester

Akash Kumar Mishra
(12022002016008)
B.Tech
V-Semester

CHAPTER: 1

INTRODUCTION

Problem Definition:

The project aims to create a recipe management system that enables users to store, retrieve, update, and delete recipes. The system should be able to handle recipe details, including the name, instructions, and a list of ingredients with their respective quantities.

Report Overview:

This report will delve into the design and implementation of a recipe management system. It will cover the following aspects:

1. **Database Design:**
    - Explanation of the database schema, including the recipes and ingredients tables.
    - Justification for the chosen data structures and relationships.

2. **System Functionality:**
    - Detailed description of the core functionalities:
        - Adding new recipes
        - Retrieving specific recipes
        - Listing all recipes
        - Updating existing recipes
        - Deleting recipes

3. **Code Implementation:**
    - Analysis of the provided Python code, focusing on:
        - Database connection and query execution
        - Data insertion, retrieval, and modification
        - Error handling and exception management

4. **User Interface :**
    - Overview of the user interface design
    - Explanation of how the user interacts with the system

CHAPTER 2:

SYSTEM DESCRIPTION

Systems are designed keeping in mind an issue that is to be solved. Every system is designed in its unique way, keeping in mind the requirement of the problem or the issue. Our system solves the problem of searching for the good that the customer's needs.
System design involves the design of overall architecture, based on which we design components, modules and interfaces. The beginning of any system architecture is by decomposing it into smaller fragments. Decomposition and binding of components makes the architecture easy to understand and makes it easier to understand.
Our system uses an algorithm for collecting data which will collect the data of the user and we have a data analyzing algorithm which will analyze and highlight the needs of the user.

This Python code, leveraging SQLite3 and Streamlit, implements a functional recipe management system. The system allows users to:

1. **Add Recipes:**

    ○ Input recipe name, instructions, and a list of ingredients.
    ○ Store the recipe and its ingredients in the SQLite database.

2. **View Recipes:**

    ○ Retrieve a list of all saved recipes.
    ○ Select a specific recipe to view its details, including name, instructions, and ingredients.

3. **Update Recipes:**

    - Choose a recipe to modify.
    - Edit the recipe's name, instructions, and ingredients.
    - Update the changes in the database.

4. **Delete Recipes:**

    - Select a recipe to remove.
    - Delete the recipe and its associated ingredients from the database.

Key Components:

- **SQLite3 Database:**
    - Stores recipe information (name, instructions) and ingredient details (recipe ID, name, quantity).

- **Streamlit Frontend:**
    - Provides a user-friendly web interface for:
        - Adding, viewing, updating, and deleting recipes.
        - Displaying recipe details and sharing options.

- **Python Backend:**
    - Handles database interactions (CRUD operations).
    - Implements business logic for recipe management.
    - Generates shareable links for recipes.

This system offers a convenient way to organize and share recipes, making it a valuable tool for home cooks and food enthusiasts.

CHAPTER 3:

THEORETICAL ANALYSIS

Core Functionalities

The system primarily revolves around four core functionalities:

1. **Adding Recipes:**

   - **Data Input:** Users provide recipe name, instructions, and a list of ingredients.
   - **Database Insertion:** The system inserts the recipe and its ingredients into the SQLite database.

2. **Viewing Recipes:**

   - **Database Query:** The system fetches all recipes from the database.
   - **User Interface:** The retrieved recipes are displayed in a user-friendly format, allowing users to select a specific recipe.
   - **Detailed View:** Once a recipe is selected, the system retrieves its detailed information (name, instructions, and ingredients) and presents it to the user.

3. **Updating Recipes:**

   - **Recipe Selection:** The user chooses a recipe to modify.
   - **Data Retrieval:** The system fetches the selected recipe's details from the database.
   - **Data Modification:** The user can update the recipe's name, instructions, and ingredients.

- **Database Update:** The modified information is updated in the database.

4. **Deleting Recipes:**
    - **Recipe Selection:** The user chooses a recipe to delete.
    - **Database Deletion:** The system removes the selected recipe and its associated ingredients from the database.

Technical Implementation

Database Design:

- **Recipes Table:** Stores recipe ID, name, and instructions.
- **Ingredients Table:** Stores ingredient ID, recipe ID, name, and quantity.

Python Backend:

- **Database Interactions:** Uses SQLite3 to connect to the database and execute SQL queries for CRUD operations.
- **Data Processing:** Handles input validation, data formatting, and error handling.
- **Business Logic:** Implements the core functionalities of adding, viewing, updating, and deleting recipes.

Streamlit Frontend:

- **User Interface:** Provides a web-based interface for user interaction.
- **Form Handling:** Collects user input for adding and updating recipes.
- **Data Display:** Presents recipes in a clear and concise manner.
- **State Management:** Manages user interactions and data persistence.

Security Considerations

- **Data Privacy:** While the system primarily handles personal recipes, it's essential to consider data privacy and security measures, especially if the system is deployed publicly.
- **Input Validation:** Implementing input validation can help prevent malicious input and potential security vulnerabilities.
- **Database Security:** Ensuring the database is secured with appropriate access controls and encryption can protect sensitive recipe data.

Potential Enhancements

- **User Authentication:** Allow users to create accounts and log in to personalize their recipe collection.
- **Recipe Sharing:** Enable users to share recipes with others via social media or email.
- **Search Functionality:** Implement a search feature to quickly find recipes based on keywords or ingredients.
- **Ingredient Substitution Suggestions:** Provide suggestions for alternative ingredients based on user preferences or dietary restrictions.
- **Nutritional Information:** Integrate with a nutrition database to provide nutritional information for recipes.
- **Mobile Optimization:** Make the system accessible on mobile devices.

By addressing these aspects, the recipe management system can evolve into a robust and user-friendly tool for food enthusiasts.

Chapter 4:

CONCLUSION

The recipe management system, implemented using Python, SQLite3, and Streamlit, provides a user-friendly and efficient solution for organizing and managing personal recipes. The system offers a range of functionalities, including adding, viewing, updating, and deleting recipes.

By leveraging the power of SQLite3 for database operations and Streamlit for the web interface, the system delivers a seamless user experience. The system's modular design and clear code structure make it easy to maintain and extend.

While the current system offers a solid foundation, there are several potential areas for future enhancements, such as implementing user authentication, advanced search capabilities, and integration with external services for nutritional information.

Overall, this recipe management system demonstrates the effectiveness of using Python and web technologies to create practical and useful applications.

Chapter 5:

REFERENCE

- Khan, P., & Tabassum, A. (2010). Service quality and customer satisfaction of the beauty-care service industry in Dhaka: A study on high-end women’s parlors. *Journal of Business in Developing Nations*, 12, 33.
- Sultana, S., & Das, T. I. S. (2016). Measuring customer satisfaction through SERVQUAL model: A study on beauty parlors in Chittagong. *European Journal of Business and Management*, 8(35), 97-108.
- Jayasathya, R., & Priya, P. (n.d.). A study on customer preference towards natural’s unisex salon and spa (with special reference to Coimbatore city).