Project Documentation

Overview

Project Name	recipe management system
Project Manager	by - AMAN KUMAR TOMAR,HANS RAJ, DEVANSH SHARMA,SAUMYA VATS
Project Dates	Start Date: Jan 10, 2025 End Date: Jan 10, 2025
Background	The Recipe Management System is designed to simplify and organize the management of culinary recipes for users. It provides a platform for adding, updating, and deleting recipes while offering features to search and filter recipes based on various criteria. This project aims to enhance user convenience by enabling them to maintain a structured repository of recipes that is accessible and easy to manage.
Objectives	1. Develop a web-based application using Maven. 2. Implement a robust backend using Servlets and JSP. 3. Ensure user-friendly interfaces with Bootstrap styling. 4. Provide core features such as recipe management, user authentication, and category-based filtering. 5. Validate inputs and implement error-handling mechanisms. 6. Enable easy deployment and scalability of the system for future enhancements. •
Target Audience	ALL THE PEOPLES WHO WAS INTRESTED IN PROGRAMING

Review1 - Project Initialization and Database Setup

Deliverables:

- **Title**: Recipe Management System
- **Description**: A web application for users to manage their recipe collections effectively.

Details:

- **Database Setup**:
  - Use MySQL to store user and recipe data.
  - Tables: Users, Recipes, Categories, and Comments.
  - Relationships: Users can add multiple recipes, and recipes can have multiple categories and comments.
- **Technology Setup**:
  - Use Maven for project dependency management.
  - Set up a multi-module project structure for better organization.

Review 2 - HTML Templates and CSS/Bootstrap Styling

Deliverables:

1. **HTML Templates**:
   - User Login Page
   - Recipe Listing Page
   - Recipe Detail Page
   - Recipe Creation/Update Page
   - User Profile Management Page
2. **CSS/Bootstrap Styling**:
   - Responsive design with Bootstrap grid system.
   - Professionally styled forms and tables.
   - Interactive buttons, navigation menus, and modals.
3. **JavaScript Validation and Interactivity**:
   - Form validation for fields like recipe name, ingredients, and instructions.
   - Dynamic error messages for invalid inputs.
   - Real-time feedback for password strength and field completion.
4. **GitHub Repository**:
   - A well-structured repository with the following hierarchy:
     - /src: Source code
     - /resources: CSS, JS, and configuration files
     - /webapp: HTML and JSP files
     - README.md: Instructions to set up and run the project.

Marking Rubric:

- HTML Templates for User Management: 4 marks
- CSS and Bootstrap Styling: 2 marks
- JavaScript for Form Validation: 2 marks

Review 3 - Implementation of Core Features

Deliverables:

1. **Servlet Implementation**:
   - Configured Servlets to handle doGet and doPost requests.
   - Recipe CRUD operations implemented via Servlets.
2. **Integration of JSP and Servlets**:
   - Dynamic data display on JSP pages using JSTL and EL.
   - Error messages and validation feedback displayed seamlessly.
3. **Core Features**:
   - User Authentication
   - Recipe Filtering by Category
   - Pagination for Recipe Lists
   - Search functionality for recipe names and ingredients.
4. **GitHub Repository**:
   - Updated code with unit tests for the service and DAO layers.
   - Comprehensive README file.

Marking Rubric:

- Servlet Configuration: 3 marks
- doGet and doPost Implementation: 3 marks
- Recipe CRUD Features: 4 marks
- JSP Integration: 4 marks
- JSTL/EL Usage: 2 marks

Review 4 - Final Submission

Deliverables:

1. **Testing and Validation**:
   - Unit tests for service and DAO layers using JUnit.
   - Integration tests to ensure seamless operation.
   - End-to-end testing for core workflows.
2. **Annotations**:
   - Use of @WebServlet, @WebFilter, and other annotations for configuration.
   - Incorporate @RequestMapping, @Controller, and similar annotations for Spring Boot integration.
3. **Documentation**:
   - Detailed project documentation including:
     - Project Overview
     - Features and Functionalities
     - Setup Instructions
     - Technology Stack
     - Testing and Validation Details
     - Future Enhancements and Roadmap
4. **Deployment**:
   - Package the project as a WAR file and deploy on Apache Tomcat.

- Provide Docker configurations for containerized deployment.

Marking Rubric:

- Unit Tests: 3 marks
- Annotations: 5 marks
- Project Documentation: 3 marks
- Deployment and Usability: 4 marks

Technology Stack

- **Backend**: Java Servlets and JSP
- **Frontend**: HTML, CSS, JavaScript, Bootstrap
- **Database**: MySQL
- **Build Tool**: Maven
- **Testing**: JUnit
- **Deployment**: Apache Tomcat, Docker

Instructions to Run the Project

1. Clone the repository from GitHub.
2. Import the project into an IDE like IntelliJ or Eclipse.
3. Set up the database schema using the provided SQL script.
4. Configure database connection in persistence.xml.
5. Build the project using Maven.
6. Deploy the application on a Tomcat server.
7. Access the application via http://localhost:8080/RecipeManagementSystem.
8. Optional: Use the provided Dockerfile to run the project in a containerized environment.

Future Enhancements

1. Add API endpoints for integration with external applications.
2. Implement user roles and permissions (e.g., admin, regular user).
3. Introduce image upload functionality for recipes.
4. Add analytics and reporting features for user activity.
5. Enable multi-language support for broader accessibility.

Conclusion

Project Outcomes	The Recipe Management System is a comprehensive platform designed to streamline recipe management for users. With a focus on functionality, aesthetics, and user experience, the project serves as an excellent demonstration of the team's ability to deliver high-quality software solutions. This system has the potential to evolve into a scalable and feature-rich application, catering to a wide audience of culinary enthusiasts.