PREMIER UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Project Report On

DEVELOPMENT AND IMPLEMENTATION OF 'FOOD RECIPE WORLD': A LARAVEL-BASED BLOGGING PLATFORM FOR CULINARY ENTHUSIASTS

Course Title: Software Development
Course Code: CSE 364

Submitted To:

Tashin Hossain

Lecturer

Department of Computer Science and Engineering

Submitted By:

Mohammed Ashab Asir (2104010202244)
Safayete Yesmin Nava: (2104010202235)
Ismita Sukkur Tasin (2104010202228)

October, 2024

TABLE OF CONTENTS

1. Introduction

"Food Recipe World" is an interactive and responsive blogging website designed to provide a platform for users to explore and share food recipes. The website offers two main user roles: normal users (or chefs) and admins. Normal users can browse, view, and contribute their own recipes, while admins have additional authority to manage and approve recipes, as well as control user access. Recipes posted by users (or chefs) remain hidden from public view until they are approved by an admin, ensuring quality control. The website was developed using Laravel, a robust PHP framework for the backend, paired with Bootstrap for the frontend to achieve a visually appealing and responsive design. The database is managed using MySQL, and XAMPP is employed to provide a local development environment. This project exemplifies a well-organized, multi-user blogging platform, enabling seamless interaction between users, chefs, and the admin.

2. Project Organization

The project was a collaborative effort, involving the following team members:

- **Mohammed Ashab Asir**: Backend Developer, responsible for server-side logic and database management. He also ensured that the core functionality of the admin panel (such as user management and post approval) works smoothly.

- **Safayete Yesmin Nava**: Frontend Developer, handled the user interface design, ensuring that the website is aesthetically pleasing and easy to navigate. She also worked on ensuring the website's responsiveness across different devices.

- **Ismita Sukkur Tasin**: Frontend Developer, contributed to designing and implementing the interactive features that allow users to post recipes, view recipes in detail, and access their accounts.

The team utilized various tools and methodologies to ensure smooth collaboration and the successful completion of the project. Communication was maintained through daily stand-ups and weekly team reviews to track progress and resolve any development challenges.

2.1. Key Features

- **Account Creation and Authentication**: Both users and admins can register, login, and logout.

- **Recipe Browsing**: All users can browse through the list of recipes and view detailed instructions for each recipe.

- **Recipe Posting**: Registered users can post their own recipes, which will only be visible once approved by the admin.

- **Admin Panel**: Admins have the exclusive ability to approve or reject recipes submitted by users. They can also delete users or chefs from the system.

3. Hardware and Software Requirements

3.1. Hardware Requirements

- Processor: A dual-core 2.0 GHz or higher.

- Memory (RAM): At least 4 GB to handle development processes smoothly.

- Storage: A minimum of 10 GB free space for project files, databases, and software installation.

- Network: A stable internet connection to ensure proper communication between users and the server during development and testing phases.

3.2. Software Requirements

- **Operating System**: Any operating system that supports XAMPP (Windows, macOS, Linux).

- **Development Environment**:

    - XAMPP: A cross-platform server solution for creating and hosting web applications locally.

- Laravel Framework: PHP-based MVC framework for efficient and structured backend development.

- Bootstrap: A frontend CSS framework for creating responsive and modern-looking web pages.

- MySQL: A relational database management system used for storing user data, recipe information, and admin controls.

- Browser: Any modern browser (Google Chrome, Mozilla Firefox, etc.) to test the web application.
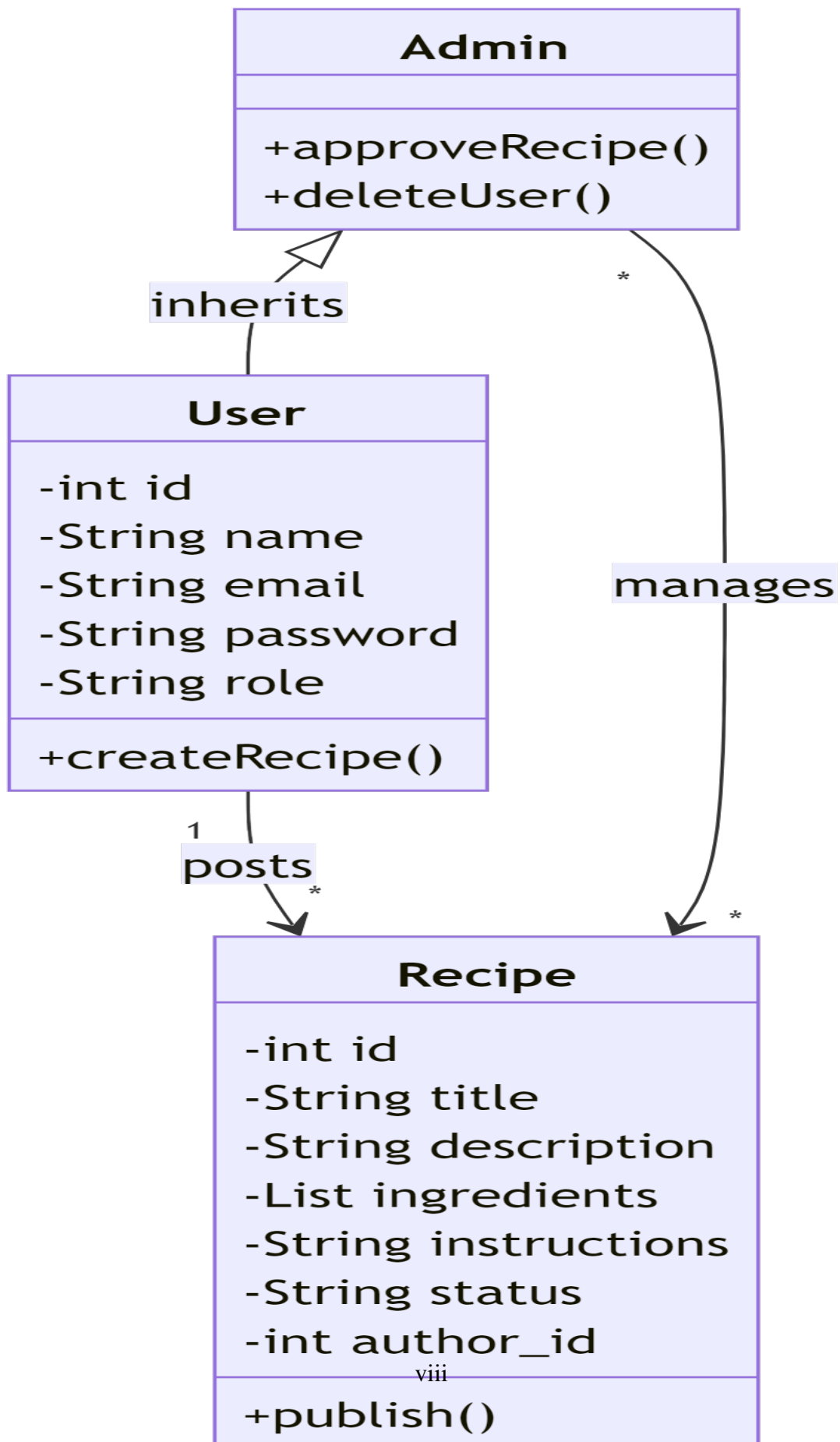
4. System Architecture

The architecture of "Food Recipe World" is designed using the Model-View-Controller (MVC) paradigm to separate business logic from the presentation layer, making the system more scalable and maintainable.

- **Model**: Represents the application data. For instance, the Recipe model contains information about each recipe, such as its title, description, ingredients, instructions, and approval status. The User model represents users and admins, with attributes like name, email, and role.

- **View**: Handles the user interface, built using Bootstrap. This is where the web pages are rendered for users and admins, providing a responsive and clean design. Views display dynamic content such as the list of recipes, detailed recipe descriptions, and user account information.

- **Controller**: Acts as an intermediary between the model and view. For example, when a user submits a new recipe, the RecipeController handles the incoming request, validates the data, and stores the recipe in the database. Admins interact with the AdminController to approve or delete posts and manage users.

4.1. Flow of the System

- **User Panel**: Allows normal users and chefs to explore recipes, view details, and post new recipes. When a user submits a recipe, it enters a "pending" state.

- **Admin Panel**: Admins can view all recipe submissions. They can approve recipes to make them publicly visible or reject them. Admins can also manage users, removing inactive or abusive accounts from the platform.

5. Class Diagram of the System

Admin
+approveRecipe()
+deleteUser()

inherits

User
-int id
-String name
-String email
-String password
-String role
+createRecipe()

1
posts
*

manages

*

Recipe
-int id
-String title
-String description
-List ingredients
-String instructions
-String status
-int author_id

+publish()

6. Project Demonstration

During the demonstration of the "Food Recipe World" project, the following key workflows will be highlighted:

- **User Registration and Login**:

    - Demonstrating how users can create accounts and log in.

    - Showing different views for normal users/chefs and admins after login.

- **Recipe Browsing and Viewing**: Users can browse the list of recipes, filter them by categories, and view detailed instructions for each recipe.

- **Recipe Posting**: Logged-in users can post new recipes, filling out a form with the recipe title, ingredients, and instructions. Once submitted, the recipe enters a pending state, awaiting admin approval.

- **Admin Privileges**:

    - Admins can view a list of pending recipes and either approve or reject them.

    - Admins also have access to a list of users and can delete any user if necessary.

7.   Testing

Testing was conducted at various stages to ensure the reliability and functionality of the website.

7.1.   Unit Testing

Laravel's built-in testing framework, PHPUnit, was used to test individual components of the system. For example, tests were written to verify that the recipe submission form works correctly and that the admin can approve or reject posts.

7.2.   Functional Testing

Manual testing was carried out to ensure that users can register, log in, post recipes, and that admins can perform their duties without errors. Testing scenarios included different user actions like submitting a recipe, managing users, and viewing approved recipes.

7.3.   Responsive Testing

The frontend was tested across different devices and screen resolutions to ensure that the website is fully responsive and functional, whether accessed on a desktop, tablet, or mobile phone. Bootstrap's grid system was used extensively to handle screen size adjustments.

8. Limitations

Despite the functionality and responsiveness of the website, there are some limitations:

- **Admin Approval Delay**: Since every recipe post must be approved by an admin, there could be delays in content publishing if the admin does not frequently monitor the pending posts.

- **Limited User Interaction**: Currently, user interaction is limited to posting recipes. Features like commenting on or rating recipes could be introduced to make the platform more interactive and engaging.

- **Lack of Notifications**: Users do not receive notifications when their recipes are approved or rejected, which could reduce engagement. Adding a notification system would improve the user experience.

- **Scalability**: While the project works well for a small user base, scaling the platform to support a larger audience would require optimizations in the database and server infrastructure.

9. Project Link

- **Github Link**: https://github.com/Ashab-Asir/SD-Project