

RecipeHub Pro – Project Report

Course: Web Programming

Project Type: WordPress Plugin

Project Name: RecipeHub Pro – Recipe Management System

Date: 18 November 2025

1. Introduction

RecipeHub Pro is a WordPress plugin that provides a complete recipe management system.

It allows users to register, log in, and share recipes, while administrators can manage the entire recipe database from the WordPress dashboard.

The main goals of the project were:

- Implement **register/login** with SQL database interaction
- Build a **WordPress plugin** in PHP
- Provide a **frontend interface** with HTML, CSS, JavaScript (jQuery)
- Design and use a **relational database** for recipes, favorites, and ratings
- Expose **REST API endpoints** for recipe operations

2. Features Overview

2.1 User Features

- User registration and login (based on WordPress user system)
- Submit recipes via a frontend form
- Browse recipes in a list/grid view
- View detailed recipe pages (ingredients, instructions, metadata)
- Mark recipes as favorites
- Rate recipes (1–5 stars) :contentReference[oaicite:2]{index=2}

2.2 Admin Features

- Recipe management from the WordPress admin panel
- Add, edit, delete recipes
- Manage recipe status: **published / pending / draft**
- View and moderate user submissions
- View ratings/favorites data for each recipe :contentReference[oaicite:3]{index=3}

3. Technology Stack

3.1 Backend

- **Language:** PHP 7.4+
- **Platform:** WordPress 5.0+ plugin system
- **Database:** MySQL / MariaDB (WordPress default)

Why WordPress & PHP?

- Native integration with the WordPress user system
- Built-in security tools (nonces, sanitization, capability checks)
- Simple access to the SQL database via `\\$wpdb`
- Easy deployment on standard hosting :contentReference[oaicite:4]{index=4}

3.2 Frontend

- **HTML5** for semantic structure
- **CSS3** for responsive layout
- **JavaScript (jQuery)** for AJAX and dynamic UI interactions

jQuery is used because it already ships with WordPress and is enough for this project's scope (no heavy frontend framework required).
:contentReference[oaicite:5]{index=5}

4. System Design & Architecture

4.1 Plugin Structure

```
```text
recipe-collector/
 ├── recipe-collector.php # Main plugin file
 ├── includes/
 │ ├── class-rc-database.php # Database operations
 │ ├── class-rc-api.php # REST API endpoints
 │ ├── class-rc-admin.php # Admin interface
 │ ├── class-rc-frontend.php # Frontend shortcodes & views
 │ └── class-rc-auth.php # Authentication (register/login)
 └── assets/
 ├── css/
 │ ├── style.css # Frontend styles
 │ └── admin.css # Admin styles
 └── js/
 ├── script.js # Frontend JS
 └── admin.js # Admin JS

```

### Main responsibilities:

- **RC\_Database**: create custom tables, run CRUD queries
- **RC\_API**: define REST routes and handle requests
- **RC\_Admin**: render admin pages and forms
- **RC\_Frontend**: register shortcodes and output HTML on the site
- **RC\_Auth**: manage user registration and login flow

## 5. Database Design

RecipeHub Pro uses **custom tables** instead of WordPress posts to keep the data model clean and optimized for recipe queries.

### 5.1 Tables

*rc\_recipes*

| Field               | Type         | Description                        |
|---------------------|--------------|------------------------------------|
| <b>id</b>           | bigint       | Primary key                        |
| <b>title</b>        | varchar(255) | Recipe title                       |
| <b>description</b>  | text         | Short description                  |
| <b>ingredients</b>  | text         | Ingredients list                   |
| <b>instructions</b> | text         | Step-by-step instructions          |
| <b>prep_time</b>    | int          | Preparation time (minutes)         |
| <b>cook_time</b>    | int          | Cooking time (minutes)             |
| <b>servings</b>     | int          | Number of servings                 |
| <b>difficulty</b>   | varchar(50)  | easy / medium / hard               |
| <b>category</b>     | varchar(100) | Category (e.g. Dessert, Main Dish) |
| <b>image_url</b>    | varchar(500) | Image URL                          |
| <b>created_by</b>   | bigint       | WordPress user ID                  |
| <b>status</b>       | varchar(20)  | published / pending / draft        |

*rcFavorites*

- id – Primary key
- user\_id – WordPress user
- recipe\_id – Linked recipe

*rc\_ratings*

- id – Primary key
- user\_id – WordPress user
- recipe\_id – Linked recipe
- rating – Value 1–5

## 6. API Endpoints

Base URL:

/wp-json/recipe-collector/v1/

## 6.1 Recipes

| Method | Endpoint      | Description       | Auth   |
|--------|---------------|-------------------|--------|
| GET    | /recipes      | List recipes      | Public |
| GET    | /recipes/{id} | Get single recipe | Public |
| POST   | /recipes      | Create recipe     | Admin  |
| PUT    | /recipes/{id} | Update recipe     | Admin  |
| DELETE | /recipes/{id} | Delete recipe     | Admin  |

## 6.2 User Actions

| Method | Endpoint               | Description         | Auth |
|--------|------------------------|---------------------|------|
| POST   | /recipes/{id}/favorite | Add/remove favorite | User |
| POST   | /recipes/{id}/rating   | Submit rating       | User |
| GET    | /favorites             | Get user favorites  | User |
| GET    | /my-recipes            | Get recipes by user | User |

All write operations use nonces, capability checks, and prepared statements for security.

## 7. Security Considerations

- **Nonce verification** on all AJAX/API requests
- **Sanitization/escaping** of user input and output
- **Capability checks** for admin actions (e.g. manage\_options)
- **Prepared statements** to prevent SQL injection
- Protection against CSRF via WordPress nonce system

## 8. Screenshots

### 8.1 Admin Interface

1. **Admin Dashboard – Recipe List**
  - Shows all recipes with status, search, and edit/delete actions.
2. **Admin – Add/Edit Recipe Form**
  - Full recipe form with title, description, ingredients, instructions, times, difficulty, and status.

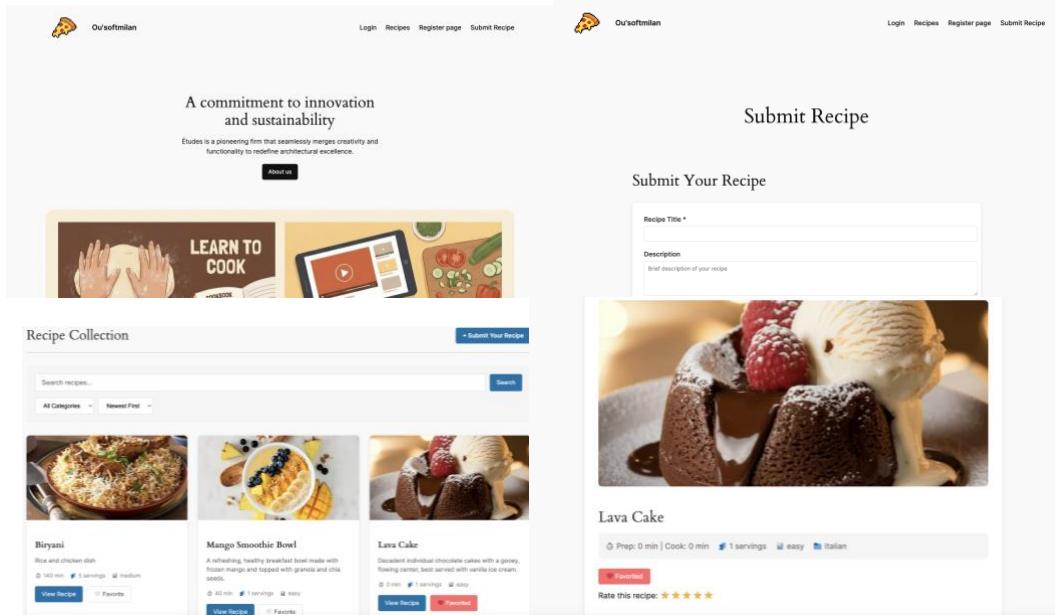
The screenshot shows the WordPress admin interface for the RecipeHub Pro plugin. The left sidebar has a dark theme with various menu items like Dashboard, Posts, Media, Pages, Comments, RecipeHub Pro (which is active and highlighted in blue), All Recipes, Add New, Appearance, Plugins, Users, Tools, and Settings. The main content area is titled "RecipeHub Pro - All Recipes". It features a grid of four recipe cards. Each card includes a thumbnail image, the recipe name, a "PUBLISHED" status badge, the category, difficulty, and creation date. There are "Edit" and "Delete" buttons for each card. At the top right, there's a search bar with a "Search" button. The footer contains links for Login, Recipes, Register page, and Submit Recipe.

## Login

The screenshot shows a simple login form titled "Login". It has fields for "Username" and "Password", a "Remember me" checkbox, and a "Login" button at the bottom. The background is white with a light gray border around the form itself.

## 8.2 Frontend Interface

3. **Recipes Listing Page**
  - o Grid layout of recipe cards with search/filter options.
4. **Recipe Detail Page**
  - o Full recipe view with ingredients, instructions, prep/cook time, servings, difficulty, and rating.
5. **Submit Recipe Form**
  - o Frontend form used by logged-in users to submit recipes.
6. **Registration/Login Pages** Simple forms for user registration and login before interacting with recipes.



## 9. Conclusion

RecipeHub Pro meets all the required points of the course project:

- Register and login with SQL database interaction
- PHP backend as a WordPress plugin
- Frontend built with HTML, CSS, JS/jQuery
- Admin interface for recipe management
- User interface for browsing and submitting recipes
- REST API endpoints for recipe operations
- Project report (this document)

### Key outcomes:

- Solid understanding of **WordPress plugin development**
- Practical experience with **OOP PHP** and `$wpdb`
- Hands-on design of a **relational database** for a real use case
- Implementation of **REST APIs** and **AJAX** in a WordPress context
- Application of **web security best practices** in a small but complete system

### End of Report

*This report demonstrates the complete implementation of RecipeHub Pro, a WordPress plugin that fulfills all project requirements for the course examination.*