

Recipe Manager Project Documentation

Project Overview

Objective: To create a Recipe Manager application that allows users to manage recipes, including adding, editing, viewing, and deleting recipes. The application will be built using the Laravel framework and will follow MVC architecture.

Requirement Analysis

Functional Requirements

1. User Authentication:

- User registration and login.
- Password reset functionality.

2. Recipe Management:

- Add new recipes with details (title, ingredients, steps, category, cooking time, etc.).
- Edit existing recipes.
- View a list of all recipes.
- View detailed information for a specific recipe.
- Delete recipes.

3. Category Management:

- Add, edit, and delete recipe categories.

4. Search and Filter:

- Search recipes by title or ingredients.
- Filter recipes by category or cooking time.

5. User Profile:

- View and edit user profile details.
- View a list of user-specific recipes.

6. Ratings and Reviews:

- Allow users to rate recipes.
- Allow users to leave reviews and comments on recipes.
- Display average rating and recent reviews on the recipe detail page.

7. Favorite Recipes:

- Allow users to mark recipes as favorites.
- Provide a view for users to see their favorite recipes.

8. Recipe Sharing:

- Enable users to share recipes via social media (Facebook, Twitter, etc.).
- Provide a shareable link for each recipe.

9. Recipe Collections:

- Allow users to create collections (e.g., “Weeknight Dinners”, “Holiday Baking”).
- Users can add recipes to their collections.

10. Meal Planning:

- Allow users to plan their meals for the week.
- Provide a calendar view where users can drag and drop recipes to plan their meals.

11. Shopping List:

- Enable users to generate a shopping list from selected recipes.
- Allow users to manually add items to the shopping list.
- Provide a printable view of the shopping list.

12. Recipe Import/Export:

- Allow users to import recipes from popular cooking websites.
- Provide an option to export recipes in various formats (PDF, Word).

13. Nutritional Information:

- Include fields for nutritional information (calories, fat, protein, etc.) in recipes.
- Display nutritional information on the recipe detail page.

14. Advanced Search and Filter:

- Add filters for dietary preferences (vegetarian, vegan, gluten-free, etc.).
- Enable advanced search by multiple criteria (ingredients, cooking time, rating).

15. Recipe Recommendations:

- Suggest recipes to users based on their preferences and previous interactions.
- Implement a “You might also like” section on the recipe detail page.

16. Admin Panel:

- Create an admin dashboard to manage users, recipes, and categories.
- Provide tools for moderating reviews and comments.

17. Notifications:

- Send email notifications for recipe updates, new reviews, and comments.
- Provide in-app notifications for user activities.

18. User Roles and Permissions:

- Implement different user roles (admin, user, guest).
- Define permissions for each role (e.g., only admin can delete recipes).

Non-Functional Requirements

1. Usability:

- User-friendly interface.
- Responsive design to support various devices.

2. Performance:

- Efficient database queries to ensure fast load times.

3. Security:

- Secure user authentication and authorization.
- Input validation to prevent SQL injection and XSS attacks.

Features

1. User Registration and Authentication:

- Registration form with validation.
- Login form with validation.
- Password recovery using email.

2. Recipe CRUD Operations:

- Form to add new recipes with title, ingredients, steps, category, cooking time, etc.
- List all recipes with options to view, edit, or delete each recipe.
- Detailed view of a single recipe.

3. Category Management:

- Add new categories.
- Edit existing categories.
- Delete categories.

4. Search and Filter:

- Search bar to find recipes by keywords.
- Filters to narrow down recipes by category or cooking time.

5. User Profile:

- User can view and update their profile information.
- List of recipes created by the user.

6. Ratings and Reviews:

- Allow users to rate recipes.
- Allow users to leave reviews and comments on recipes.
- Display average rating and recent reviews on the recipe detail page.

7. Favorite Recipes:

- Allow users to mark recipes as favorites.
- Provide a view for users to see their favorite recipes.

8. Recipe Sharing:

- Enable users to share recipes via social media (Facebook, Twitter, etc.).
- Provide a shareable link for each recipe.

9. Recipe Collections:

- Allow users to create collections (e.g., “Weeknight Dinners”, “Holiday Baking”).
- Users can add recipes to their collections.

10. Meal Planning:

- Allow users to plan their meals for the week.
- Provide a calendar view where users can drag and drop recipes to plan their meals.

11. Shopping List:

- Enable users to generate a shopping list from selected recipes.
- Allow users to manually add items to the shopping list.
- Provide a printable view of the shopping list.

12. Recipe Import/Export:

- Allow users to import recipes from popular cooking websites.
- Provide an option to export recipes in various formats (PDF, Word).

13. Nutritional Information:

- Include fields for nutritional information (calories, fat, protein, etc.) in recipes.
- Display nutritional information on the recipe detail page.

14. Advanced Search and Filter:

- Add filters for dietary preferences (vegetarian, vegan, gluten-free, etc.).
- Enable advanced search by multiple criteria (ingredients, cooking time, rating).

15. Recipe Recommendations:

- Suggest recipes to users based on their preferences and previous interactions.
- Implement a “You might also like” section on the recipe detail page.

16. Admin Panel:

- Create an admin dashboard to manage users, recipes, and categories.
- Provide tools for moderating reviews and comments.

17. Notifications:

- Send email notifications for recipe updates, new reviews, and comments.
- Provide in-app notifications for user activities.

18. User Roles and Permissions:

- Implement different user roles (admin, user, guest).
- Define permissions for each role (e.g., only admin can delete recipes).

Database Design

Tables:

1. Users:

- `id` (primary key)
- `name`
- `email`
- `password`
- `created_at`
- `updated_at`

2. Recipes:

- `id` (primary key)
- `user_id` (foreign key to Users)
- `title`
- `ingredients` (text)
- `steps` (text)
- `category_id` (foreign key to Categories)
- `cooking_time` (integer, in minutes)
- `calories` (integer)
- `fat` (integer)
- `protein` (integer)
- `carbs` (integer)
- `created_at`
- `updated_at`

3. Categories:

- `id` (primary key)
- `name`
- `created_at`
- `updated_at`

4. Reviews:

- id (primary key)
- recipe_id (foreign key to Recipes)
- user_id (foreign key to Users)
- rating (integer)
- comment (text)
- created_at
- updated_at

5. Favorites:

- id (primary key)
- user_id (foreign key to Users)
- recipe_id (foreign key to Recipes)
- created_at
- updated_at

6. Collections:

- id (primary key)
- user_id (foreign key to Users)
- name
- created_at
- updated_at

7. Collection_Recipe:

- id (primary key)
- collection_id (foreign key to Collections)
- recipe_id (foreign key to Recipes)
- created_at
- updated_at

8. Meal_Plans:

- id (primary key)
- user_id (foreign key to Users)
- date (date)
- recipe_id (foreign key to Recipes)
- created_at
- updated_at

9. Shopping_Lists:

- id (primary key)
- user_id (foreign key to Users)
- items (text)
- created_at
- updated_at

Project Timeline (15 Days)

Day 1-2:

- Set up Laravel project.
- Configure database and environment settings.
- Create migrations for users, recipes, categories, reviews, favorites, collections, collection_recipe, meal_plans, and shopping_lists tables.

Day 3-4:

- Implement user registration and authentication.
- Create authentication views (register, login, password reset).

Day 5-6:

- Design and implement CRUD operations for recipes.
- Create views for adding, editing, listing, and viewing recipes.

Day 7-8:

- Design and implement category management.
- Create views for adding, editing, and listing categories.

Day 9:

- Implement search and filter functionality for recipes.
- Add search bar and filter options in recipe list view.

Day 10:

- Implement ratings and reviews for recipes.
- Create views for submitting and displaying reviews.

Day 11:

- Implement favorite recipes feature.
- Create views for marking and listing favorite recipes.

Day 12:

- Implement recipe sharing and collections.
- Create views for sharing recipes and managing collections.

Day 13:

- Implement meal planning and shopping lists.
- Create views for meal planning calendar and shopping list generation.

Day 14:

- Work on user interface and responsiveness.
- Use Laravel Mix to compile assets (CSS, JS).
- Implement user roles and permissions.

Day 15:

- Final testing and deployment preparation.
- Write documentation for the project (setup guide, feature explanation).

Technologies and Tools

- ❖ **Backend:** Laravel Framework
- ❖ **Frontend:** Blade templating engine, Bootstrap (or any CSS framework)
- ❖ **Database:** MySQL or PostgreSQL
- ❖ **Version Control:** Git (GitHub or GitLab)
- ❖ **Testing:** PHPUnit, Laravel Dusk for browser testing

Conclusion

By following this project overview and timeline, we are able to develop a comprehensive Recipe Manager application using Laravel within 15 days. Finally, we will ensure a robust and user-friendly application.