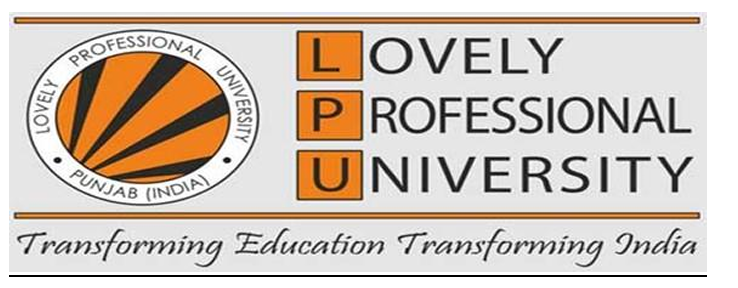
****

**Name : Maradana Vinay**

**Reg No : 12216602**

**Project : Recipe Vault**

**Submitted To : Poornima mam (Placement Trainer)**

**(Project Term June-July 2025)**

**Abstract**

RecipeVault is an interactive and user-centric web-based application designed to provide a seamless platform for culinary enthusiasts to create, manage, and explore a wide variety of recipes. Developed using Django, this system facilitates the storage, retrieval, and categorization of recipes contributed by registered users. It bridges the gap between passionate home cooks and those seeking new culinary experiences by offering a structured and visually engaging space to share food creations.

At its core, RecipeVault allows users to register, log in, and post their personal recipes with key details such as the title, a complete list of ingredients, step-by-step cooking instructions, required cooking time, and the difficulty level. Each recipe can also include an image, which is automatically resized for consistency and performance optimization. Recipes are organized into categories (e.g., Breakfast, Main Course, Dessert, Beverages), allowing users to filter and explore content more efficiently.

One of the most significant features of RecipeVault is its content moderation system. Every recipe submitted by a user is marked as “Pending” by default and requires admin approval before being visible to others. This ensures quality control and prevents the spread of incorrect or incomplete content. Additionally, the platform supports real-time status updates, recipe filtering by category or difficulty, and intuitive user interface design using Tailwind CSS and JavaScript.

From a technical perspective, RecipeVault is built on a robust backend using Django and Python, with SQLite used during development and PostgreSQL as the recommended production database. The application uses Django’s Object-Relational Mapping (ORM) to interact with the database, enhancing developer productivity and ensuring data integrity. It also integrates tools like Pillow for image processing and Git for version control.

RecipeVault stands out not only as a recipe manager but also as a scalable community platform. Its modular design allows for future enhancements like recipe reviews, ratings, video tutorials, comment sections, and mobile app integration. By focusing on simplicity, security, and scalability, RecipeVault delivers a complete solution for modern-day food lovers looking to share their culinary journey with the world.

**Major Modules**

1. User Management Module

The User Management Module is essential for handling the authentication and identity of users. It utilizes Django’s built-in user model and authentication system to provide secure and streamlined access to platform features.

Key Functionalities:

* User Registration & Login:
  + New users can register using a unique username, password, and email address.
  + Login functionality includes validation and secure session management.
* Authentication & Authorization:
  + Django handles session cookies and password hashing to ensure account security.
  + Permissions are enforced so only authenticated users can create or manage recipes.
* User-Recipe Linking:
  + Each recipe is linked to the user who created it, allowing personalized views like “My Recipes.”
* Profile Control (optional):
  + Can be extended to allow users to update their profiles or add avatars and bios.

This module ensures only valid users interact with the system, laying the groundwork for secure and personalized access.

2. Recipe Management Module

This is the heart of the RecipeVault platform. It enables users to create, view, update, and manage their recipes, along with related metadata like cooking time and difficulty.

Key Functionalities:

* Create Recipe:
  + Users can add recipes by providing a title, ingredients, instructions, cooking time (in minutes), difficulty (Easy, Medium, Hard), category, and an optional image.
* Edit/Update Recipe:
  + Users can update their own recipes, edit ingredients or steps, or change the image.
* Status Workflow:
  + Every new recipe starts with the “Pending” status.
  + Only after admin approval does the recipe become “Approved” and visible to other users.
* Timestamping:
  + Recipes are automatically stamped with created\_at and updated\_at fields for tracking.

This module ensures structured recipe storage, smooth editing experience, and a clean publishing process with admin oversight.

3. Category Management Module

Recipes are organized into categories to make browsing and filtering more effective. The Category Management Module supports scalable classification.

Key Functionalities:

* Add/Edit/Delete Categories:
  + Admins can manage the list of categories (e.g., Breakfast, Dessert, Main Course).
* Assigning Categories:
  + Each recipe is linked to a single category through a foreign key relationship.
* Category-Based Filtering:
  + On the frontend, users can browse recipes by selecting a category.

By offering categorized access, this module improves user experience and content organization.

**Minor Modules**

1. Image Processing Module

To ensure a consistent and optimized user experience, RecipeVault includes automatic image resizing. When a user uploads a recipe image, it is resized to 300x200 pixels using the Pillow (PIL) library. This helps:

* Maintain a uniform appearance across all recipes.
* Reduce page loading times.
* Save server storage space.

Invalid or corrupted images are safely handled to prevent errors during upload.

2. Status Handling Module

Each recipe in the system has a status field, which helps track its visibility:

* Pending: Submitted by the user but not yet approved.
* Approved: Visible to all users.
* Rejected: Not published, possibly due to missing or inappropriate content.

Admins have the authority to change recipe statuses. This module ensures proper moderation and content control.

3. Time & Difficulty Management Module

Every recipe includes:

* Cooking Time (in minutes): Helps users estimate preparation time.
* Difficulty Level: Labeled as Easy, Medium, or Hard.

These attributes are used to:

* Filter and sort recipes.
* Help beginners choose suitable recipes.
* Support better planning and user engagement.

**Tools Used**

RecipeVault leverages a combination of modern tools and technologies for efficient development, responsive design, database management, and image handling. Below is a breakdown of the major tools used in the project:

**1. Django (Backend Framework)**

* Used as the main web framework to handle routing, authentication, and database operations.
* Provides built-in admin panel, ORM (Object-Relational Mapping), and user authentication system.

**2. Python (Programming Language)**

* Core language used to develop backend logic, model definitions, and view functions.
* Offers simplicity, readability, and strong integration with Django.

**3. SQLite / PostgreSQL (Database)**

* **SQLite** is used during development for simplicity.
* **PostgreSQL** is the recommended production-ready relational database, known for speed, scalability, and ACID compliance.

**4. HTML & CSS**

* HTML structures the frontend content.
* CSS provides basic styling and layout to webpages.

**5. Tailwind CSS**

* A utility-first CSS framework used to build responsive and modern UIs.
* Speeds up frontend development with prebuilt utility classes.

**6. JavaScript**

* Adds interactivity to forms, navigation menus, and client-side validations.
* Ensures a dynamic and engaging user experience.

**7. Pillow (PIL)**

* Python Imaging Library used for processing and resizing uploaded recipe images.
* Ensures image dimensions and formats are consistent.

**8. Visual Studio Code (VS Code)**

* Code editor used for developing the project.
* Supports linting, debugging, Git integration, and Django extensions.

**9. Git & GitHub**

* Git handles version control for tracking code changes.
* GitHub stores the codebase remotely, allowing collaboration, backup, and deployment support.

**Database**

RecipeVault uses a **relational database** to store and manage all structured data such as user accounts, recipes, categories, and status logs. Django’s built-in **Object-Relational Mapping (ORM)** provides a high-level abstraction over raw SQL, enabling developers to interact with the database using Python code instead of SQL queries.

**Database Engine**

* **SQLite** is used during development for simplicity and quick setup.
* **PostgreSQL** is preferred for production due to its:
  + ACID compliance
  + Indexing support
  + Performance on large-scale data
  + Compatibility with Django ORM
  + Full-text search and JSON field support

**Main Tables**

**1. Users Table**

* Stores information about registered users.
* Fields: id, username, email, password, and other authentication metadata.
* Linked to recipes via a foreign key (author\_id).

**2. Categories Table**

* Maintains a list of recipe categories.
* Fields: id, name (unique).
* One category can have many recipes.

**3. Recipes Table**

* Core table containing all recipe-related data.
* Fields include:
  + title
  + ingredients
  + instructions
  + cooking\_time
  + difficulty (easy, medium, hard)
  + image (optional)
  + status (pending, approved, rejected)
  + author (foreign key to Users)
  + category (foreign key to Categories)
  + created\_at, updated\_at (timestamps)

**Relationships**

* **User → Recipes**: One-to-Many (each user can create many recipes).
* **Category → Recipes**: One-to-Many (each category includes many recipes).

This well-structured schema allows efficient data storage, filtering, and retrieval throughout the application.

**ER Diagram**

ER Diagram (Explained)

The Entity-Relationship (ER) Model for RecipeVault consists of three primary entities: User, Recipe, and Category. These entities are connected through well-defined relationships that support the system’s core functionalities like recipe submission, categorization, and user ownership.

1. User Entity

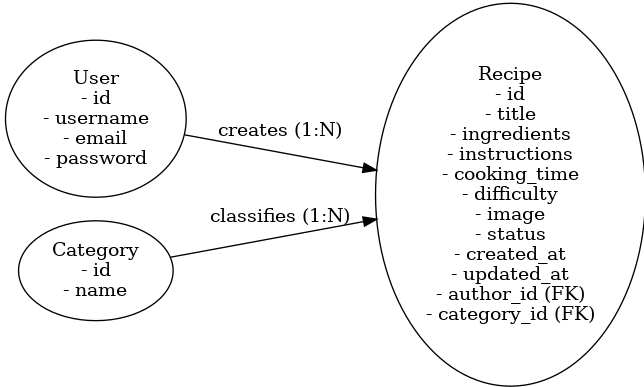
* Represents registered users of the platform.
* Contains fields like id, username, email, password, etc.
* Responsible for authentication and ownership of recipes.

2. Category Entity

* Represents types or classifications of recipes (e.g., Breakfast, Dessert).
* Contains id and name (unique for each category).
* Helps group and filter recipes based on their type.

3. Recipe Entity

* Represents the main content in the platform.
* Contains all necessary fields: title, ingredients, instructions, cooking\_time, difficulty, image, status, created\_at, and updated\_at.
* Foreign Keys:
  + author → linked to User (who created the recipe).
  + category → linked to Category (type of the recipe).



**Description :**

RecipeVault is a web-based application designed to enable users to share, discover, and manage food recipes in a simple and structured manner. Built using the Django framework, the platform provides authenticated users with the ability to post recipes, categorize them, and view those shared by others. The application caters to both casual users looking for quick meal ideas and passionate cooks wanting to showcase their culinary creations.

The core functionality revolves around a recipe submission workflow. Registered users can create a new recipe by filling out a form that includes the recipe title, ingredients, step-by-step cooking instructions, estimated cooking time, difficulty level, and an optional image. Once submitted, the recipe enters a moderationphase, where it is marked as “Pending” until reviewed by an admin. Admin users can either approve the recipe, making it public, or reject it if the content is inappropriate or incomplete.

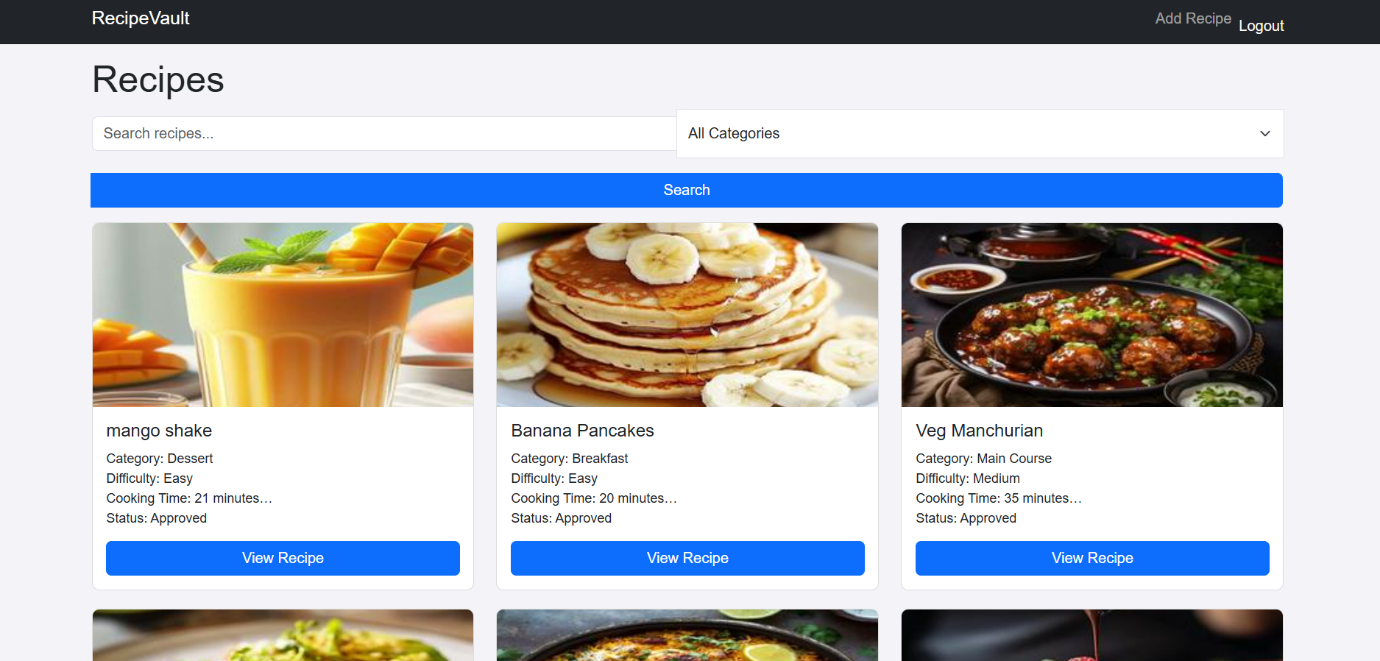
Each recipe is linked to a category such as Breakfast, Dessert, or Main Course, enabling efficient filtering and browsing. Users can also filter recipes based on cooking time and difficulty level, making it easier to find suitable meals according to time and skill constraints.

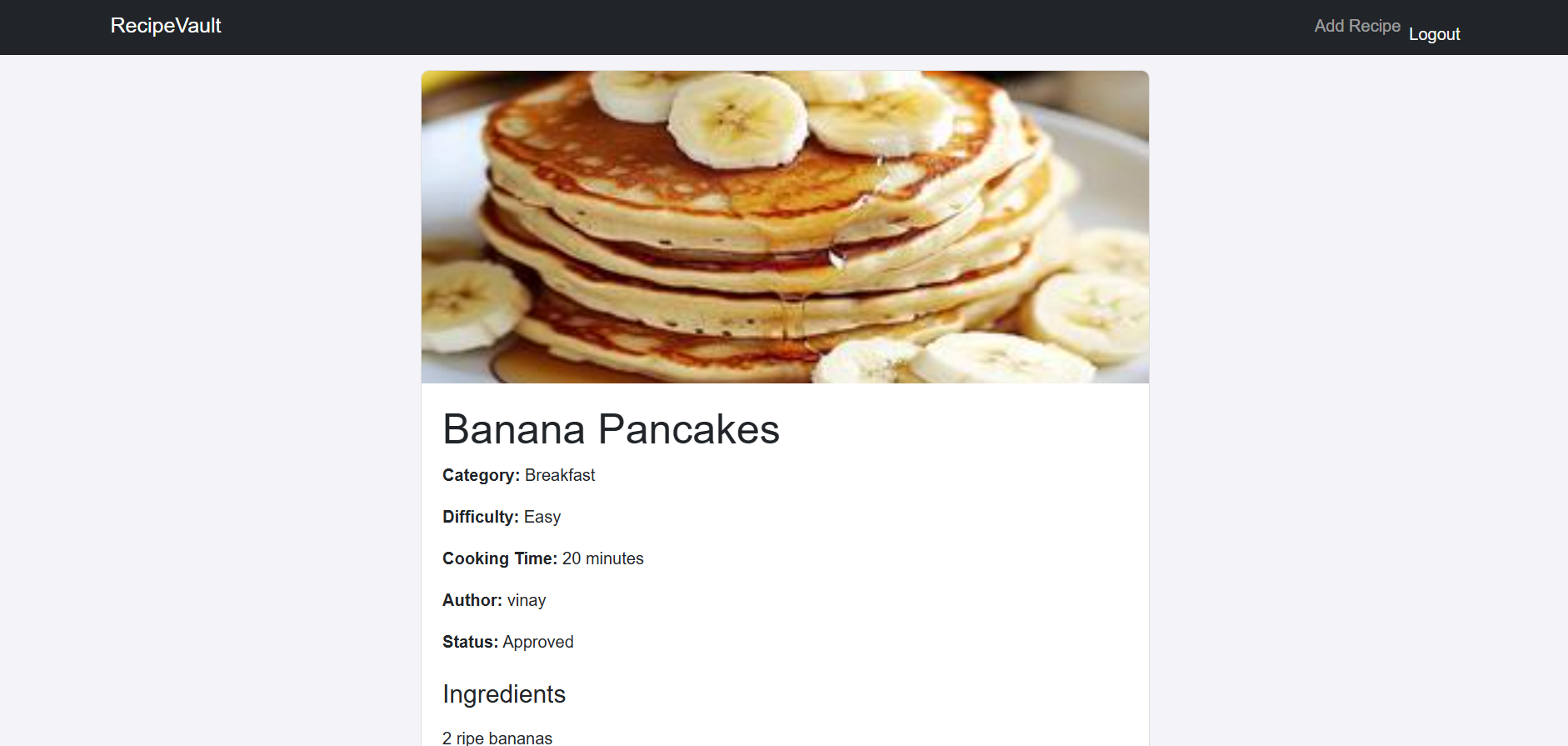
The frontend is styled using Tailwind CSS to ensure a modern, responsive layout across devices. The backend uses Django’s ORM to handle data modeling and database interactions, providing secure, scalable performance. Uploaded images

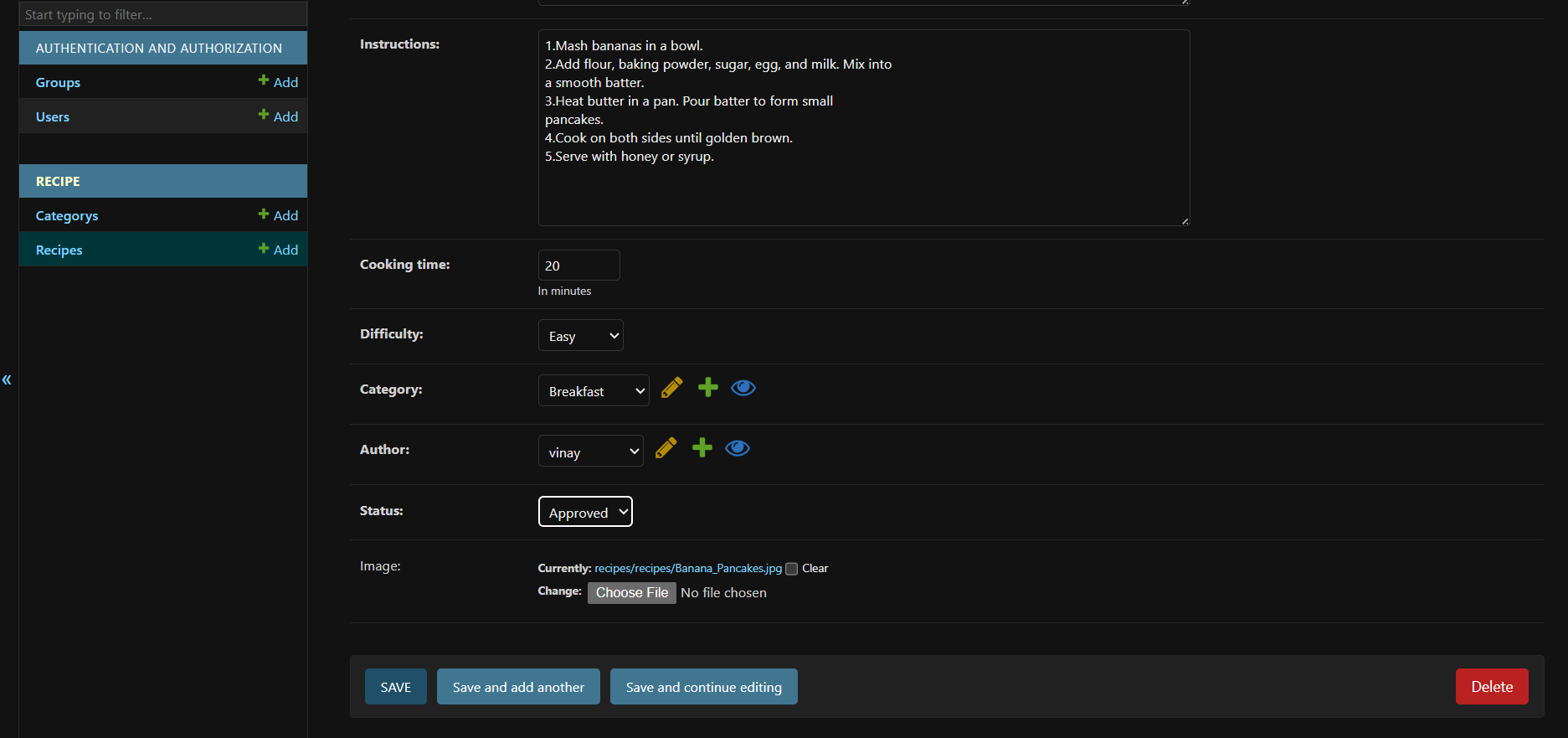
are automatically resized using Pillow (PIL) to maintain uniformity and reduce load times.

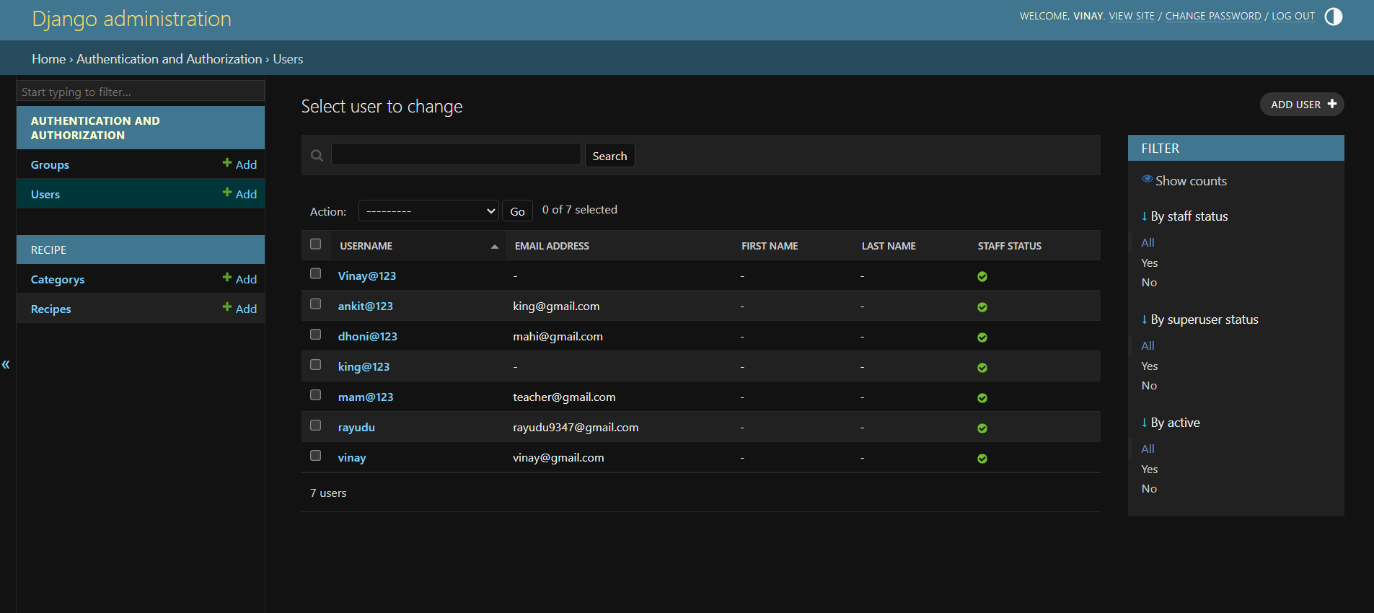
RecipeVault is built with modularity and future scalability in mind. The system supports extension for features like user ratings, comments, and video tutorials. It serves as a practical and elegant solution for digital recipe management in today’s connected world.

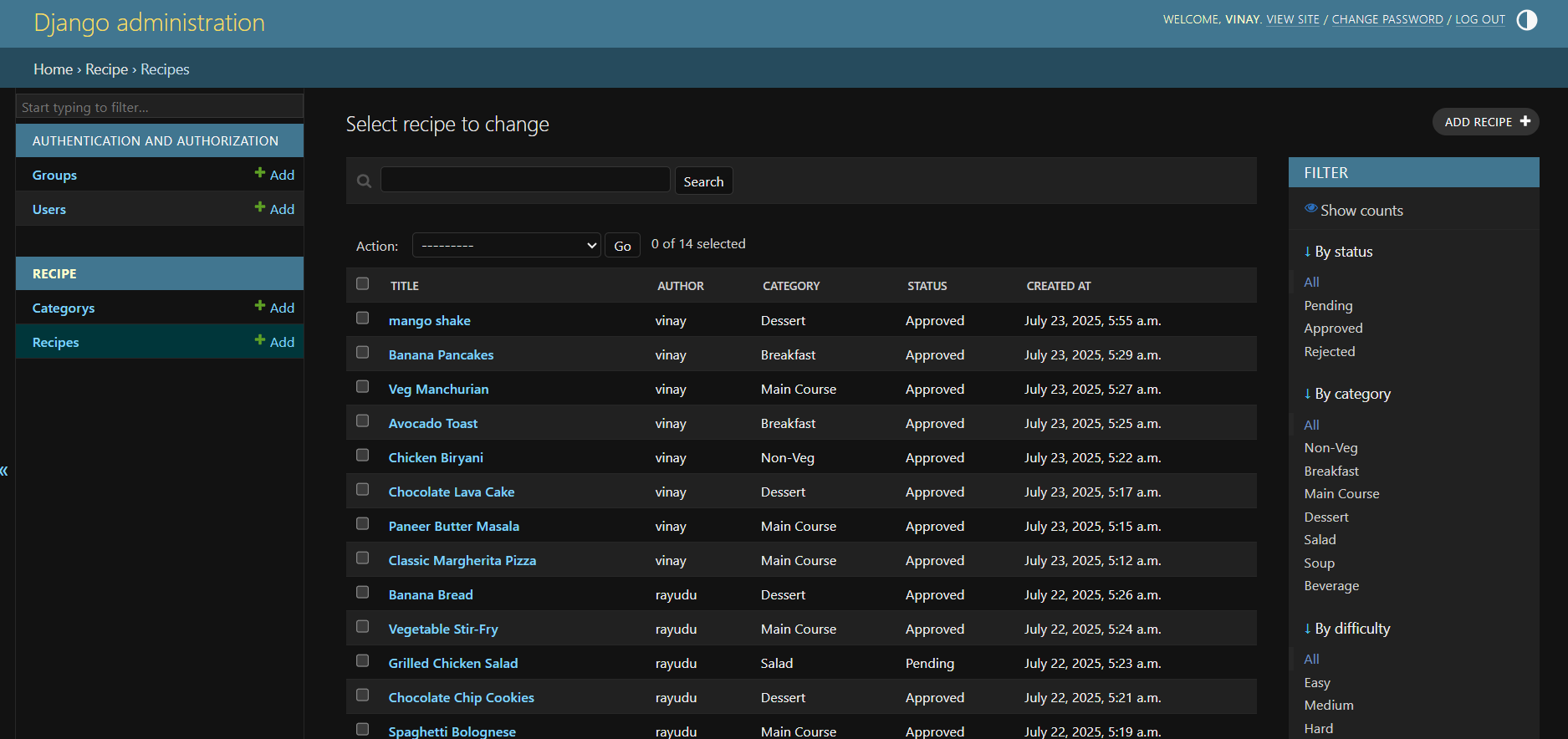
**Screenshot :**

****

****

****

****

****

**Feature Enhancement :**

To improve functionality, user engagement, and scalability, the following feature enhancements are proposed for future versions of RecipeVault:

 1. Advanced Filtering and Search

Introduce powerful filtering options to allow users to search for recipes by:

* Ingredients
* Cooking time range
* Category (e.g., Breakfast, Main Course)
* Difficulty (Easy, Medium, Hard)

This will make it easier for users to find recipes that match their needs and preferences.

 2. Recipe Ratings and Reviews

Allow users to:

* Rate recipes using a star-based system (1 to 5 stars)
* Leave reviews or feedback about their cooking experience  
  This will promote the best recipes and build trust among users.

 3. Commenting System

Add a commenting feature to allow interaction under each recipe. Users can:

* Ask questions
* Give cooking tips or alternatives
* Discuss their results

 4. Mobile App API Integration

Develop RESTful APIs to support mobile applications. This will:

* Enable mobile access to recipes
* Allow notifications for updates or approvals
* Expand user reach

 5. Video Support for Recipes

Allow users to upload or embed cooking videos to:

* Demonstrate step-by-step preparation
* Increase clarity and appeal
* Enhance engagement for complex recipes

 6. Multi-language Support

Implement multilingual options to allow users to view content in:

* Regional languages (Telugu, Hindi, Tamil)
* International languages (English, Spanish)  
  This will improve accessibility and expand the platform's audience.

 7. Enhanced Security Features

Future security enhancements could include:

* Two-Factor Authentication (2FA)
* Login alerts and password recovery workflows

**Conclusion**

The RecipeVault project successfully demonstrates the implementation of a full-featured, scalable, and user-friendly recipe management system. Built using Django, the platform enables users to register, submit, and explore recipes through a clean, categorized interface while ensuring quality control through an admin-based approval system.

Key features such as image optimization, cooking time and difficulty tracking, and category-based organization enhance usability and functionality. The system’s modular design, use of Django ORM, and integration with frontend technologies like Tailwind CSS make it efficient, maintainable, and extensible.

By leveraging modern development tools and best practices, RecipeVault delivers a secure and interactive user experience that supports culinary creativity and collaboration. It serves not only as a personal recipe book but as a social platform for sharing food knowledge across diverse audiences.

With future enhancements like mobile integration, user ratings, and multilingual support, RecipeVault holds significant potential to grow into a comprehensive digital hub for cooking enthusiasts. Overall, it is a well-structured and impactful project with real-world applicability and long-term scalability.