**Software Requirements Specification (SRS)**

**Recipe Management App**

**1. Introduction**

**1.1 Project Overview**  
The Recipe Management App is designed to provide users with a platform to manage, browse, and share various recipes. It offers functionalities to add, edit, delete, categorize recipes, and manage user profiles and preferences.

**1.2 Objectives**

* To create an intuitive and user-friendly platform for managing and sharing recipes.
* To allow users to explore and discover new recipes through categories and search functionalities.
* To provide a comprehensive admin dashboard for managing content and users.

**1.3 Scope**  
The application will include:

* User registration, login, and profile management.
* Recipe management (add, edit, delete, view).
* Categorization and searching of recipes.
* Admin functionalities for content and user management.

**2. Functional Requirements**

* 1. **User Management**

**1)Register User:** Allows new users to create an account.

 **Acceptance Criteria:**

* A new user should be able to successfully create an account with valid credentials.
* Upon registration, the user should receive a confirmation email.

 **Test Plan:**

* Test registration process with valid and invalid inputs.
* Verify email confirmation functionality.
* Test error handling for duplicate usernames or invalid data.

**2)Login/Logout:** Users can log in and log out securely.

 **Acceptance Criteria:**

* Users should be able to securely log in using their credentials.
* Logout should successfully terminate the user session and return to the login page.

 **Test Plan:**

* Test login with correct and incorrect credentials.
* Verify session management and logout functionality.

**3)Profile Management:** Users can update their profile details.

 **Acceptance Criteria:**

* Users should be able to update their profile information (e.g., username, email, password).
* Changes to profile information should reflect accurately across the application.

 **Test Plan:**

* Test updating profile information.
* Verify data integrity and validation for profile fields.

**4)Password Management:** Users can change or reset their passwords.

 **Acceptance Criteria:**

* Users can change their password securely.
* Users can reset their password via email verification.
* Password changes and resets are immediately effective.

 **Test Plan:**

* Test changing passwords from user settings.
* Verify successful login with new credentials.
* Test password reset via email link.
* Ensure secure verification and successful login post-reset.
  1. **Recipe Management**

**1)Add Recipe:** Users can add new recipes with ingredients and instructions.

 **Acceptance Criteria:**

* Users should be able to add a new recipe with all required fields (title, ingredients, instructions).
* Newly added recipes should be visible in the user’s recipe list.

 **Test Plan:**

* Test adding recipes with various combinations of ingredients and instructions.
* Verify visibility and accessibility of added recipes.

**2)Edit Recipe:** Users can edit existing recipes.

 **Acceptance Criteria:**

* Users should be able to edit existing recipe details.
* Changes to recipe details should be saved and reflected accurately.

 **Test Plan:**

* Test editing recipe details.
* Verify data validation and persistence of changes.

**3)Delete Recipe:** Users can delete recipes they no longer want to keep.

 **Acceptance Criteria:**

* Users should be able to delete recipes from their list.
* Deleted recipes should no longer be accessible or visible in the user's recipe list.

 **Test Plan:**

* Test deleting recipes from the user interface.
* Verify that deleted recipes are no longer visible in the user's recipe list.
* Ensure proper handling of deletion confirmation and error scenarios.

**4)View Recipe:** Users can view detailed information about a recipe.

 **Acceptance Criteria:**

* Users should be able to view detailed information about a recipe.
* Recipe details (title, ingredients, instructions) should be displayed accurately.

 **Test Plan:**

* Test viewing recipes from the recipe list.
* Verify that all details (title, ingredients, instructions) are displayed correctly.
* Test navigation and accessibility of recipe details page from different parts of the application.

**5)Search Recipe:** Users can search for recipes by ingredients, category, or keywords.

 **Acceptance Criteria:**

* Users should be able to search for recipes by keyword, ingredient, or category.
* Search results should accurately display relevant recipes based on the search criteria.

 **Test Plan:**

* Test searching recipes using different keywords, ingredients, and categories.
* Verify that search results match the entered criteria and display relevant recipes.
  1. **Recipe Categorization**

**1)Add Category:** Admins can add new recipe categories.

 **Acceptance Criteria:**

* Admins should be able to add new recipe categories.
* Newly added categories should be available for recipe categorization.

 **Test Plan:**

* Test adding new categories through admin interface.
* Verify category management and association with recipes.

**2)Edit Category:** Admins can edit existing categories.

 **Acceptance Criteria:**

* Admins should be able to edit existing recipe categories.
* Changes to category details should be saved and reflected accurately.

 **Test Plan:**

* Test editing categories through the admin interface.
* Verify that changes to category names or descriptions are saved correctly.

**3)Delete Category:** Admins can delete categories.

 **Acceptance Criteria:**

* Admins should be able to delete existing recipe categories.
* Deleted categories should no longer be available for recipe categorization.

 **Test Plan:**

* Test deleting categories through the admin interface.
* Ensure that deleted categories do not appear in the category list for recipe management.
  1. **Admin Management**

**1)Manage Users:** Admins can view, edit, or delete user accounts.

 **Acceptance Criteria:**

* Admins should be able to view, edit, or delete user accounts as needed.
* Changes to user accounts should be reflected accurately.

 **Test Plan:**

* Test user management functionalities (view, edit, delete users).
* Verify permissions and data integrity for user accounts.

**2)Manage Recipes:** Admins can approve or reject user-submitted recipes.

 **Acceptance Criteria:**

* Admins should be able to manage (approve or reject) user-submitted recipes.
* Approved recipes should be visible to all users.

 **Test Plan:**

* Test managing recipes (approve or reject) through the admin interface.
* Verify that approved recipes are visible to all users and rejected recipes are appropriately handled.

**3. Non-functional Requirements**

**3.1 Performance**

 **Acceptance Criteria:**

* The application should load and display recipes within 2 seconds.
* It should support at least 100 concurrent users without performance degradation.

 **Test Plan:**

* Perform load testing to measure response times under various user loads.
* Validate scalability by simulating concurrent users.

**3.2 Usability**

 **Acceptance Criteria:**

* The interface should be intuitive and easy to navigate.
* The app should be responsive and compatible with mobile devices.

**Test Plan:**

* Conduct usability testing with representative users.
* Evaluate accessibility and responsiveness across different devices.

**3.3 Security**

 **Acceptance Criteria:**

* User data should be encrypted and stored securely.
* Role-based access control (RBAC) should be implemented to manage feature access.

 **Test Plan:**

* Conduct security testing to verify data encryption during transmission and storage.
* Test RBAC implementation by validating user access permissions across different features.

**3.4 Reliability**

 **Acceptance Criteria:**

* The application should be available 99.9% of the time (high availability).
* Data backup and recovery mechanisms should be in place to prevent data loss.

 **Test Plan:**

* Test application uptime and availability under normal and peak loads.
* Test data backup procedures to ensure data integrity and completeness.

**3.5 Maintainability**

 **Acceptance Criteria:**

* Code should be modular and well-documented to facilitate future maintenance and feature additions.
* Version control systems (e.g., Git) should be used to manage code changes and version history effectively.

 **Test Plan:**

* Conduct code reviews to assess code modularity and adherence to coding standards.
* Test version control workflows to ensure seamless code collaboration and change management.

**4. Technical Requirements**

**4.1 Technology Stack**

* **Backend**: Python with Flask for server-side logic.
* **Frontend**: HTML for structure, CSS for styling, and JavaScript for interactivity.
* **Database**: MySQL.
* **Deployment**: AWS for hosting.

**4.2 Development Tools**

* **IDE**: PyCharm or Visual Studio Code for development.
* **Version Control**: Git for version control and GitHub for repository management.
* **Testing Tools**:
  + **Manual Testing**: Jira for test management and issue tracking.
  + **Test Case Management**: TestRail for managing test cases and execution.

**4.3 Integration**

* Integration with email services (e.g., SMTP) for notifications.
* API endpoints for mobile app integration, allowing third-party services to interact with the app.

**5. Business Requirements**

**5.1 Goals**

* Enhance user engagement by providing a rich repository of recipes.
* Facilitate a community-driven platform for sharing and discovering recipes.
* Provide administrative tools for effective content and user management.

**5.2 Constraints**

* **Budget Limitations**: Constraints on hosting and maintenance costs.
* **Timeline**: Project must be completed within the stipulated time frame.

**5.3 Compliance Considerations**

* **Data Protection**: Ensure compliance with data protection regulations like GDPR.
  + Implement user consent mechanisms for data collection.
  + Provide users with options to view and delete their data.
* **Accessibility**: Adhere to accessibility standards (e.g., WCAG) to ensure the app is usable by people with disabilities.
* **Privacy Policy**: Clearly state how user data will be used and protected.
* **Terms of Service**: Define acceptable use policies for users and administrators.

**6. Use Cases**

**6.1 Add New Recipe**

* **Description:** A user adds a new recipe to the application.
* **Preconditions:** User is logged in.
* **Postconditions:** The recipe is added to the user's recipe list and can be viewed or edited later.
* **Acceptance Criteria:** A user can successfully add a new recipe to the application.
* **Test Plan:** Execute test cases for adding recipes and verify outcomes against acceptance criteria.

**6.2 Edit Recipe**

* **Description:** A user edits the details of an existing recipe.
* **Preconditions:** Recipe exists in the user's recipe list.
* **Postconditions:** The recipe details are updated in the system.
* **Acceptance Criteria:** Users can successfully edit existing recipe details.
* **Test Plan:** Execute test cases for editing recipes through the user interface.

**6.3 Delete Recipe**

* **Description:** A user deletes a recipe from their list.
* **Preconditions:** Recipe exists in the user's recipe list.
* **Postconditions:** The recipe is removed from the user's recipe list.
* **Acceptance Criteria:** Users can successfully delete recipes from their list.
* **Test Plan:** Test deleting recipes from the user interface.

**6.4 Search Recipe**

* **Description:** A user searches for recipes by keyword, ingredient or category.
* **Preconditions:** User is on the search page.
* **Postconditions:** Search results are displayed, and the user can view recipe details.
* **Acceptance Criteria:** Users should be able to search for recipes by keyword, ingredient, or category.
* **Test Plan:** Test searching recipes using different keywords, ingredients, and categories.

**6.5 Admin Approve Recipe**

* **Description:** An admin reviews and approves a user-submitted recipe.
* **Preconditions:** The recipe is pending approval.
* **Postconditions:** The recipe is marked as approved and is visible to all users.

**Acceptance Criteria:**

* Admins should be able to approve or reject user-submitted recipes.
* Approved recipes should be visible to all users.
* Rejected recipes should not be accessible to users.

 **Test Plan:** Test approving and rejecting recipes through the admin interface.

**7. Data Requirements**

**7.1 User Data**

* User ID, Username, Email, Password (hashed), Profile Information.

**7.2 Recipe Data**

* Recipe ID, Title, Ingredients, Instructions, Category, Image, User ID (creator).

**7.3 Category Data**

* Category ID, Name, Description.

**8. Assumptions and Constraints**

**8.1 Assumptions**

* Users have a basic understanding of using web applications.
* Admins have knowledge of recipe management and user administration.

**8.2 Constraints**

* The application must be compatible with the latest versions of major web browsers.
* The app should be scalable to accommodate future growth in user base and recipe data.