**Steps to Develop App**

Step 1: **Project Setup and Environment**

* Set up your development environment: Make sure you have Python, Flask, and any necessary packages installed.
* Create a new directory for your project: This will keep your files organized.

Step 2: **Plan Your Database**

* Decide on the data you want to store: Recipe details, user information, saved recipes, etc.
* Create a database schema: Design the structure of your database tables and their relationships.

Step 3: **Set Up Your Flask App**

* Create a virtual environment: Isolate your project's dependencies.
* Install Flask and other required packages.
* Create Flask app structure: Organize your app using blueprints for different components (recipes, user, etc.).
* Set up your app's routes: Define routes for different pages and functionalities.

Step 4: **Integrate the Spoonacular API**

* Sign up for the Spoonacular API: Get your API key.
* Set up API integration: Write functions to interact with the API using the requests library.
* Fetch recipe data: Retrieve recipe information from the API.

Step 5: **Design the User Interface**

* Create HTML templates: Design templates for different pages (recipe search, recipe detail, user profile, etc.).
* Use Bootstrap: Enhance your UI using Bootstrap for styling and responsiveness.

Step 6: **User Authentication and User Interaction**

* Implement user registration and login: Use Flask's Flask-Login extension for user authentication.
* Create user profile: Allow users to view and manage their saved recipes, preferences, etc.

Step 7: **Implement Recipe Sharing on Facebook**

* Follow the steps you've outlined in your proposal for Facebook sharing.
* Add the Facebook sharing button to the recipe detail page.
* Test the sharing functionality on your local environment.

Step 8: **User-Generated Recipes and Moderation**

* Implement user-generated recipe submission form.

Set up a moderation process: Create a dashboard for moderators to review and approve/reject user-submitted recipes.

Step 9: Additional Features

Implement recipe filtering and sorting.

Create a meal planning feature.

Set up recipe recommendation system based on user preferences.

Add comments and interaction features for users.

Step 10: Testing and Deployment

Thoroughly test your app on your local environment.

Deploy your app: Choose a hosting platform (like Heroku) and deploy your app online.

Remember, starting with a clear plan, breaking tasks into smaller steps, and tackling one step at a time will help you make steady progress on your Capstone 1 project. Good luck!

**Database Tables:**

**Users Table**:

Columns: user\_id (Primary Key), username, email, password (hashed), profile\_image, registration\_date

**Recipes Table**:

Columns: recipe\_id (Primary Key), title, image\_url, cooking\_instructions, preparation\_time, total\_time, servings, user\_id (Foreign Key), creation\_date

**Ingredients Table**:

Columns: ingredient\_id (Primary Key), name, category

**RecipeIngredients Table** (Many-to-Many Relationship):

Columns: recipe\_id (Foreign Key), ingredient\_id (Foreign Key), quantity, unit

**UserSavedRecipes Table** (Many-to-Many Relationship):

Columns: user\_id (Foreign Key), recipe\_id (Foreign Key), save\_date

**UserGeneratedRecipes Table**:

Columns: user\_recipe\_id (Primary Key), user\_id (Foreign Key), title, image\_url, cooking\_instructions, creation\_date, status (pending/approved/rejected)

**Comments Table**:

Columns: comment\_id (Primary Key), user\_id (Foreign Key), recipe\_id (Foreign Key), text, comment\_date

**Database Relationships**:

Users can have multiple saved recipes (UserSavedRecipes).

Users can create and submit multiple user-generated recipes (UserGeneratedRecipes).

Recipes can have multiple ingredients (RecipeIngredients).

Recipes can have multiple comments (Comments).

UserSavedRecipes connects Users and Recipes in a many-to-many relationship.

UserGeneratedRecipes connects Users and UserGeneratedRecipes in a one-to-many relationship.

RecipeIngredients connects Recipes and Ingredients in a many-to-many relationship.

Comments connect Users and Recipes in a one-to-many relationship.

Users:

- user\_id (PK)

- username

- email

- password

- profile\_img

- reg\_date

Recipes:

- recipe\_id (PK)

- title

- image\_url

- cooking\_instr

- prep\_time

- total\_time

- servings

Ingredients:

- ingredient\_id (PK)

- name

- category

RecipeIngredients (Many-to-Many):

- recipe\_id (FK) (References Recipes.recipe\_id)

- ingredient\_id (FK) (References Ingredients.ingredient\_id)

- quantity

- unit

UserSavedRecipes (Many-to-Many):

- user\_id (FK) (References Users.user\_id)

- recipe\_id (FK) (References Recipes.recipe\_id)

- save\_date

UserGeneratedRecipes:

- user\_recipe\_id (PK)

- user\_id (FK) (References Users.user\_id)

- title

- image\_url

- cooking\_instr

- creation\_date

- status

Comments:

- comment\_id (PK)

- user\_id (FK) (References Users.user\_id)

- recipe\_id (FK) (References Recipes.recipe\_id)

- text

- comment\_date

**Routes to implement;**

**User Authentication**:

/register: User registration form and handling.

/login: User login form and handling.

/logout: Logout the current user.

**User Profile**:

/profile/<int:user\_id>: Display user profile with saved recipes, user-generated recipes, and user information.

**Recipe Search and Display**:

/: Homepage displaying a list of recipes.

/recipe/<int:recipe\_id>: Display detailed information about a specific recipe.

**User Interaction**:

/recipe/<int:recipe\_id>/save: Allow users to save a recipe to their profile.

/recipe/<int:recipe\_id>/unsave: Allow users to remove a saved recipe from their profile.

/recipe/<int:recipe\_id>/comment: Allow users to post comments on a recipe.

**User-Generated Recipes**:

/recipe/submit: Form to submit a user-generated recipe.

/recipe/pending: Display user-generated recipes pending moderation.

/recipe/approve/<int:user\_recipe\_id>: Approve a user-generated recipe.

/recipe/reject/<int:user\_recipe\_id>: Reject a user-generated recipe.

**Recipe Filtering and Sorting**:

/recipes/filter: Page to filter recipes by cuisine, dietary preferences, etc.

/recipes/sort: Page to sort recipes by rating, preparation time, etc.

**Meal Planning**:

/meal/plan: Page to plan meals by selecting recipes and generating a shopping list.

**Social Sharing**:

/recipe/<int:recipe\_id>/share/facebook: Share a recipe on Facebook.

**Recipe Recommendations**:

/recommendations: Display personalized recipe recommendations based on user preferences.