

MY COOKBOOK: DEVELOPING A WEB APPLICATION FOR RECIPE MANAGEMENT USING FLASK AND MONGODB

Group members: Nguyễn Thùy Anh - 2001140003

Nguyễn Mai Phương - 2201140066

Nguyễn Thanh Thùy - 2201140092

Nguyễn Thu Trang - 2201140095

Instructor: PhD. Nguyen Xuan Thang

Abstract

My CookBook is a web application crafted to facilitate the management of cooking recipes, enabling users to browse, add, edit, delete, and organize personal recipes while managing their accounts. Developed using Flask with Python, MongoDB Atlas, HTML, CSS, JavaScript, and integrated with Google OAuth for secure authentication, the application caters to diverse users: non-registered users can explore recipes, whereas registered users and administrators wield comprehensive management tools. The application delivers a seamless user experience with a responsive interface, optimized loading times through MongoDB enhancements, and robust security via password hashing. This project offers a practical, user-centric solution that is poised for future enhancements such as recipe search and recommendation functionalities.

1. Introduction

This project addresses the need for a simple, efficient, and culturally relevant web application for recipe management in Vietnam, emphasizing Vietnamese

cuisine. My CookBook seeks to bridge this gap by leveraging Flask, MongoDB Atlas, and Google OAuth to create a platform that is both user-friendly and technically robust.

The novelty of My CookBook lies in its tailored approach. It integrates modern technologies to deliver a responsive interface compatible with desktop, tablet, and mobile devices, employs Google OAuth for secure and convenient authentication, and focuses on Vietnamese dishes to meet local needs. Additionally, it introduces an administrative role for comprehensive user and recipe oversight and implements efficient pagination to enhance usability. The methodology encompasses developing the application while collecting recipe data from public sources or manual entry, testing across multiple scenarios, and consulting technical documentation to resolve challenges. The data comprises recipes stored in MongoDB, including details like name, ingredients, and images, alongside user information such as email and hashed passwords.

2. Main Content

2.1. System Architecture

The frontend of My CookBook is built with HTML, CSS, powered by Materialize CSS, and JavaScript through the mycookbook.js file, which provides an intuitive user interface. Key templates, such as base.html for the layout foundation, all_recipes.html for recipe listings, and admin_users.html for administrative functions, facilitate seamless navigation. The backend, driven by Flask in Python, orchestrates business logic via the routes.py file, handling requests for authentication, recipe operations, and account management. MongoDB Atlas, a NoSQL database, serves as the data repository, housing two primary collections: one for users, storing email, hashed passwords, and recipe

lists, and another for recipes, capturing attributes like name, ingredients, and images. Authentication is fortified by Google OAuth, enabling secure login through Google accounts, complemented by traditional login with bcrypt-hashed passwords.

2.2. Key Features

My CookBook offers a robust suite of features tailored to diverse user needs. User authentication forms the cornerstone, allowing registration and login through conventional credentials or Google OAuth. Upon Google login, the system generates a user account using the Google email and prompts for password setup, with passwords securely hashed. An admin account, pre-configured in the database, holds privileges to oversee all users and recipes.

The Browse Recipes feature presents a comprehensive list of recipes on the `all_recipes.html` page, displayed as cards with names, images, and brief descriptions. Pagination, set at eight recipes per page, leverages Flask-PyMongo to optimize query performance. Clicking a recipe card navigates to the `single_recipe_details.html` page, which details the recipe's name, cuisine type, cooking time, servings, ingredients, instructions, and image. For authors or admins, this page includes options to edit or delete the recipe.

Registered users can add recipes via the `add_recipe.html` form, inputting mandatory fields like name and ingredients, with optional image uploads defaulting to a placeholder if omitted. Editing is streamlined through `edit_recipe.html`, where pre-filled forms simplify modifications, while deletion requires confirmation via a Materialize CSS modal to prevent errors. The My Recipes feature, accessible on `my_recipes.html`, lists a user's recipes with pagination and displays their total count, offering an "Add Recipe" button for convenience.

Account management empowers users to update usernames, passwords, or delete accounts through dedicated templates like `account_settings.html` and `delete_account.html`, with account deletion erasing associated recipes after confirmation. Administrators access a dashboard on `admin_users.html` to manage users and recipes, supported by templates like `admin_add_user.html` for adding users, with access enforced by a Flask `admin_required` decorator. Error handling is addressed with custom `404.html` and `500.html` pages for not-found and server errors, enhancing user experience.

2.3. Development Process

The development of My CookBook unfolded from April 14 to May 7, 2025, across four distinct phases, as documented in the project tracking sheet. The initial phase, spanning April 14 to 20, focused on establishing the Flask project, defining the directory structure, and connecting to MongoDB Atlas. User authentication, encompassing registration, login, and Google OAuth integration, was implemented, alongside basic templates like `base.html` and `home.html`. CSS files, including `style.css` and `background_forms.css`, and JavaScript via `mycookbook.js` laid the groundwork for the interface.

The second phase, from April 21 to 27, concentrated on core recipe functionalities. The Browse Recipes feature was developed with pagination on `all_recipes.html`, followed by Single Recipe Details on `single_recipe_details.html`, and Add/Edit Recipe capabilities via `add_recipe.html` and `edit_recipe.html`. Forms were created in `forms.py`, and authentication and browsing features underwent initial testing to ensure reliability.

The third phase, from April 28 to May 4, advanced administrative capabilities and interface refinement. The admin role was implemented with templates like `admin_users.html`, and the Delete Recipe function was deployed. Responsive

design was optimized for desktop, tablet, and mobile devices, addressing earlier issues with content overflow. Error pages, 404.html and 500.html, were introduced, and comprehensive testing covered authentication, recipe management, and account operations.

The final phase, from May 5 to 7, completed the My Recipes feature on my_recipes.html with pagination, followed by rigorous testing across devices to validate functionality. Technical documentation and a user manual were finalized, culminating in the preparation of this project report.

2.4. Results and Discussion

The My CookBook application achieved its objectives, delivering a fully functional platform with all intended features: authentication, recipe management, account management, and administrative oversight. The interface, optimized with Materialize CSS and media queries, performs seamlessly across desktop, tablet, and mobile devices. Performance enhancements, achieved by indexing MongoDB fields like author and _id and optimizing queries with projection, resulted in significantly faster page loading times. Security measures, including bcrypt password hashing, Google OAuth, and admin access controls, ensure a robust and safe user experience.

Development was not without challenges. Configuring Google OAuth proved difficult due to ambiguous documentation on redirect URIs, necessitating consultation of Authlib resources and verification on the Google Cloud Console. Responsive design initially faltered, with recipe cards overflowing on mobile devices and slow loading for large recipe lists, which was resolved by applying overflow-x: auto and refining CSS with Materialize. Admin privilege management encountered issues, such as residual data after user deletion, addressed by revising the delete_many logic and adding session termination.

Comprehensive testing across devices was time-intensive, particularly with large datasets, but strategic time allocation and prioritization of key features mitigated this.

Compared to platforms like Yummly or Allrecipes, My CookBook excels in its focus on Vietnamese cuisine, lightweight design, and secure authentication via Google OAuth. However, it lacks advanced features like keyword-based search or social media sharing, which are prevalent in international counterparts. These gaps highlight opportunities for future enhancements.

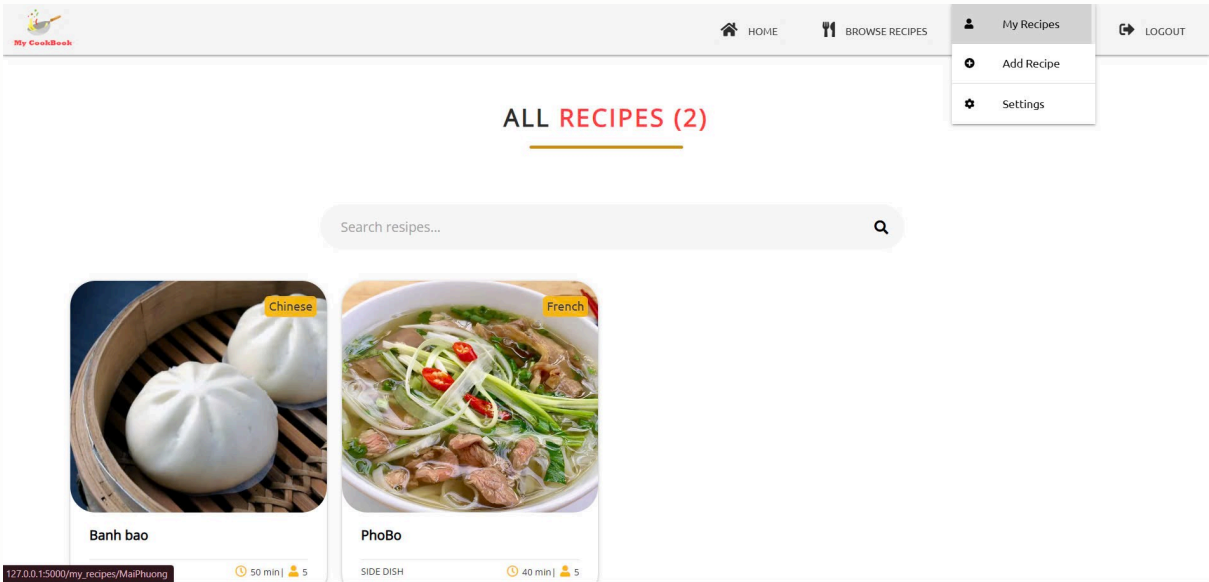


Figure 2: Browse Recipes interface (all_recipes.html)

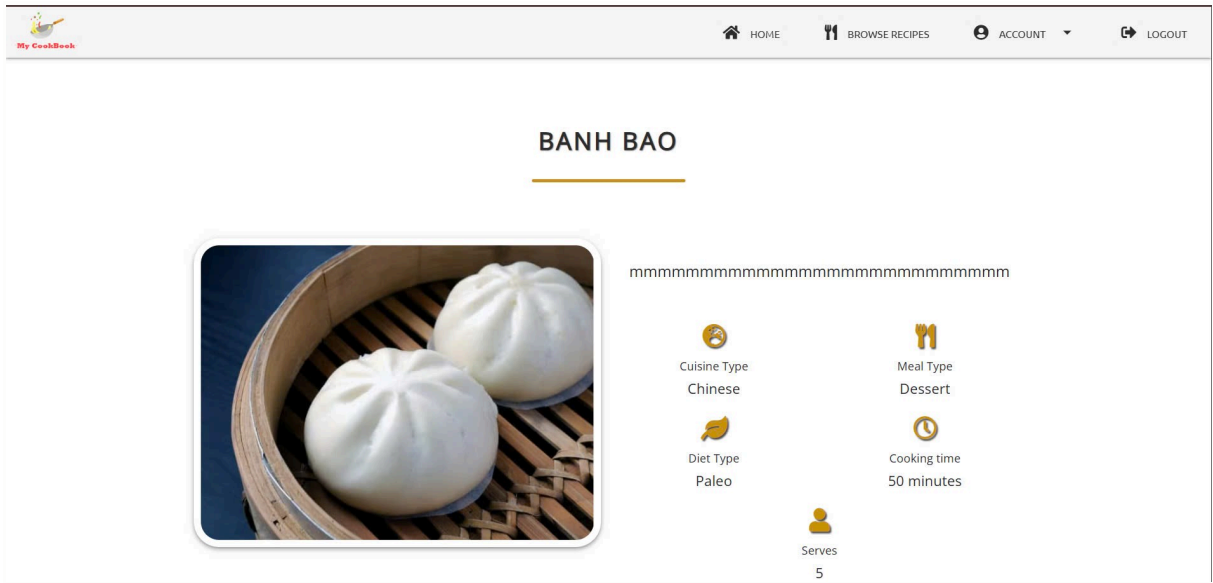


Figure 3: Single Recipe Details interface (single_recipe_details.html)

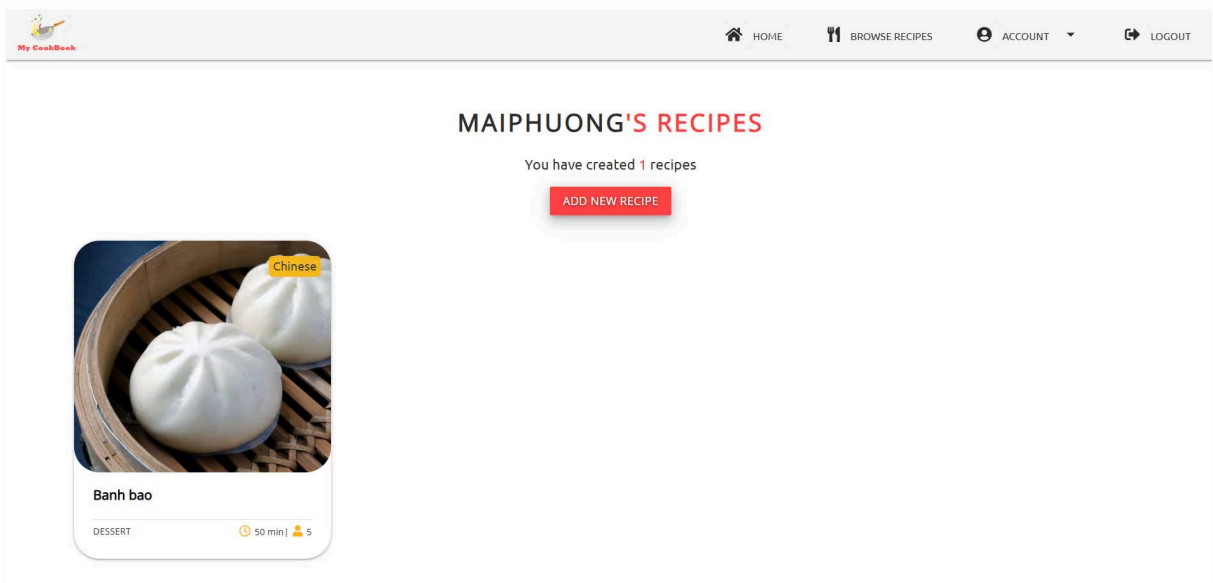


Figure 4: My Recipes interface (my_recipes.html)

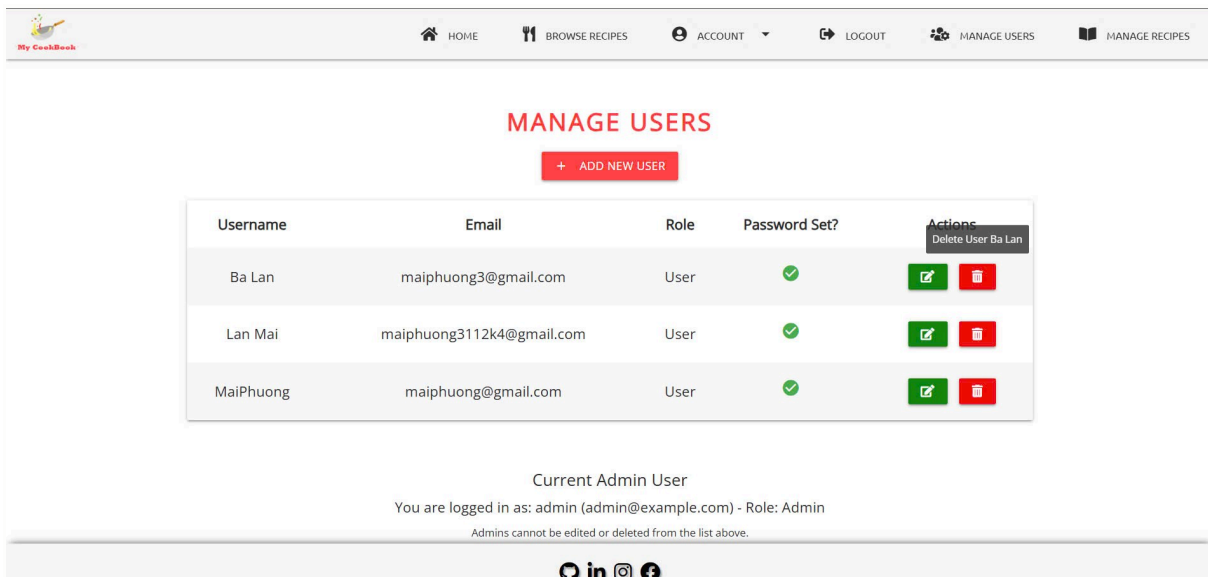


Figure 5: Admin Dashboard interface (admin_users.html)

2.5. Comparative Analysis

My CookBook distinguishes itself from global platforms by prioritizing Vietnamese culinary needs, offering a streamlined interface, and ensuring mobile optimization. Its integration of Google OAuth provides a security edge over many local alternatives. Nevertheless, the absence of search and recommendation features limits its competitiveness against established platforms, underscoring areas for future development.

3. Conclusion

The My CookBook project successfully delivered a web application for recipe management, tailored to Vietnamese users and cuisine. By harnessing Flask, MongoDB Atlas, and Google OAuth, it offers a user-friendly, high-performance platform with robust security. All features, from authentication to admin management, were implemented and rigorously tested, meeting the objectives of the Specialized Subject 2 course and demonstrating practical development skills.

The project's contributions include providing an accessible tool for recipe organization, enhancing user experience through responsive design, and addressing local culinary needs. However, limitations persist, such as the lack of search and recommendation functionalities and a modest initial recipe dataset. Future iterations could incorporate keyword-based search, ingredient-driven recipe suggestions, and social media integration, alongside expanding the recipe database with diverse Vietnamese dishes from credible sources.

REFERENCES

Using OAuth 2.0 to Access Google APIs. (n.d.). Google for Developers.

<https://developers.google.com/identity/protocols/oauth2>

Documentation - materialize. (n.d.). <https://materializecss.com/>

What is MongoDB Atlas? - Atlas - MongoDB Docs. (n.d.).

<https://www.mongodb.com/docs/atlas/>

Acsany, P. (2024, March 22). *Build a scalable Flask web project from scratch.*

<https://realpython.com/flask-project/>

W3Schools.com. (n.d.). <https://www.w3schools.com/html/>