BIS 698 Information Systems Capstone Project
Group 1

Shivani Polagouni
Srija Vadla
Amith Gundluri
Sai Chandu Anumula
Venkat Phanindra Pasunooru

CENTRAL MICHIGAN UNIVERSITY
BIS698 Information Systems Project 22454983 (Hybrid)
Prof. Wilfred Owobu

BIS 698 INFORMATION SYSTEM PROJECT

COOK MATE

Cook -Mate

Recipe Finder Application

Table of Contents

Milestone 1

Milestone 2

1.Business Description:

Name: Smart Recipe Finder (Cook Mate)

Organization: The Gourmet Hub

The Gourmet Hub is a vibrant gourmet community dedicated to making cooking enjoyable and accessible for all, ranging from novice home cooks to professional chefs. The organization offers a diverse range of services, including recipe databases, tutorials, and cooking supplies.

To enhance its online presence, The Gourmet Hub is introducing Cook Mate, a desktop application designed to simplify recipe finding by utilizing ingredients users already have. This initiative aims to foster creativity in cooking and reduce food waste while creating a community driven platform for recipe sharing and culinary learning.

Additionally, the app seeks to create a vibrant, community-driven platform where users can share their own recipes, search recipes with ingredients while learning from and inspiring one another. By combining personalized recipe recommendations with a supportive and interactive community, Cook Mate aligns with The Gourmet Hub's mission to enhance culinary experiences and make gourmet cooking an integral part of everyday life.

2.Business Problems:

Reducing Food Waste: Unused ingredients often lead to food waste in households. Cook Mate will address this issue by suggesting recipes based on available ingredients, helping users minimize waste and save money.

Simplifying Meal Planning: Busy lifestyles can result in unhealthy eating habits. Cook Mate will simplify meal planning by providing quick and efficient recipe suggestions, promoting healthier eating.

Meeting Dietary Requirements: There is an increasing demand for recipes catering to various dietary needs. Cook Mate will include features to filter recipes based on dietary constraints such as vegan and gluten-free options.

Promoting Home Cooking: Many people lack confidence in cooking due to inadequate instructions. Cook Mate will leverage its supportive community to encourage more individuals to cook at home.

Improving User Engagement: Many platforms lack interactive elements. Cook Mate will enhance user engagement by allowing recipe submissions, video uploads, and reviews, fostering a dynamic community.

Expanding Revenue Possibilities: The platform will explore new revenue streams through brand partnerships, targeted marketing, and premium content options.

3.Project Overview:

Smart Recipe Finder (Cook Mate) is a desktop application developed using Python, Tkinter for the graphical user interface, and MySQL for backend database management. The application will:

Ingredient-Based Recipe Search: Allow users to input ingredients they have and receive a customized list of recipes they can prepare.

Recipe Submission Enable users to submit their own recipes, redirect to instructional cooking videos.

Filtering Options: Offer filtering based on dietary constraints, difficulty levels, and types of cuisine.

Community Features: Create a community-driven platform where users can share their cooking experiences and interact with one another.

4.Project Feasibility

Economic Feasibility

Utilizing open-source technologies (Python, Tkinter, MySQL) will keep development costs low. Revenue will be generated through affiliate marketing, app purchases, and premium features.

Breakdown of Software Tools:

- **Python**: Free (open-source)
- **Custom Tkinter**: Free (open-source)
- **SQL Database (MySQL/PostgreSQL)**: Free (open-source)
- **PIL (Pillow for Image Processing)**: Free (open-source)
- **Design Tools**: $1,000 (for premium tools like Adobe XD, Canva Pro, or Figma)
- **Icons & Images**: $200 (for stock images and premium icon packs)

Technical Feasibility

The project employs proven technologies and tools that are well-documented and supported:

- **Programming Languages**: Python with CustomTkinter for a modern, interactive user interface.
- **Database**: SQL for reliable data management and relational queries.
- **Frameworks**: PIL for image processing and Tkinter for GUI development.
- **Technical Support**: Abundant resources and tutorials are available for troubleshooting and enhancement.

Schedule Feasibility

The project can be completed within a manageable timeline of 15 weeks, aligning with academic or operational deadlines:

- Phase 1 (Weeks 1-3): Requirement gathering, system design, and UI prototyping.
- Phase 2 (Weeks 4-8): Development of core functionalities (user authentication, recipe searches, and community features).
- Phase 3 (Weeks 9-12): Testing and debugging, implementing advanced features (real-time dashboards and data import/export).
- Phase 4 (Weeks 13-15): Deployment, user training, and final system review.

Operational Feasibility

The Cook Mate system is designed for smooth operation, minimal user training, and high user satisfaction:

- **User-Friendly Design:** Intuitive UI ensures users of all technical abilities can interact with the system effortlessly.
- **Efficient Recipe Search:** Users can search for recipes by ingredients, dietary preferences, or cooking time.
- **Admin Dashboard:** Admins can moderate content, manage users, and track community engagement through a dedicated panel.
- **Training Requirements:** Minimal training is required, as the user interface is intuitive, with step-by-step guidance.

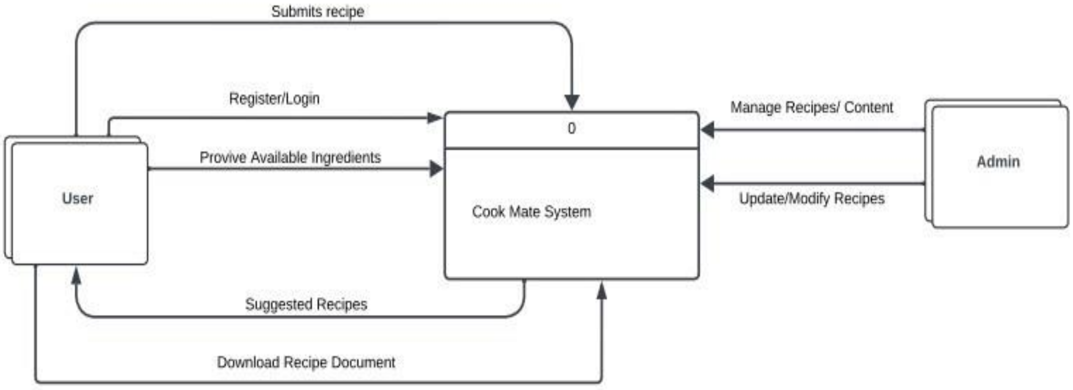
5.Functional requirements

Functional Requirements for Smart Recipe Finder (Cook Mate)

- User Authentication and Profile Management
- Users must be able to create and manage personal profiles.
- Provide secure login using username/email and password.
- Allow users to input available ingredients and receive a list of recipes that match.
- Display details for each recipe, including ingredients, cooking time, and difficulty level.
- Suggest recipe variations based on missing or alternative ingredients.
- Recipe Filtering
- Enable users to filter recipes based on: Dietary restrictions (e.g., vegan, gluten-free, dairy-free), Cuisine types (e.g., Italian, Indian, Mexican).
- Recipe Submission
- Allow registered users to submit their own recipes, including ingredients, cooking instructions, and images.
- Provide a feature to generate a shopping list based on selected recipes.
- Implement an admin panel for content moderation to review user-submitted recipes and videos.

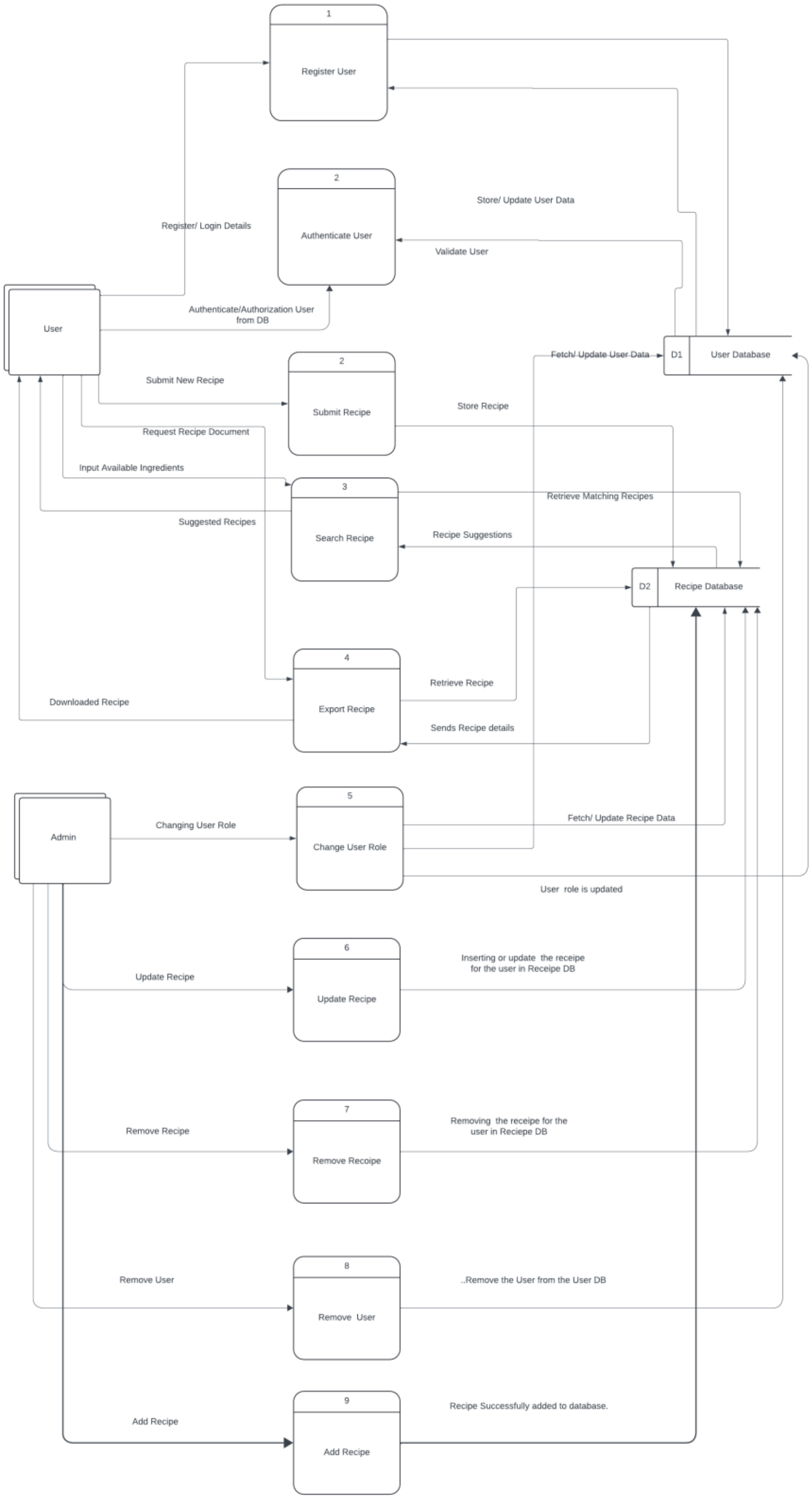
6.Data Flow Diagrams (DFD)

6.1 Context Level Diagram



The above Context level diagram shows "Cook Mate" system's workflow where users can register, submit recipes, provide ingredients, and receive suggested recipes, while admins manage and update recipe content.

6.2 Level 0 Diagram



User Processes:

1. **Login User:** ○ Users enter their login credentials to access the system.

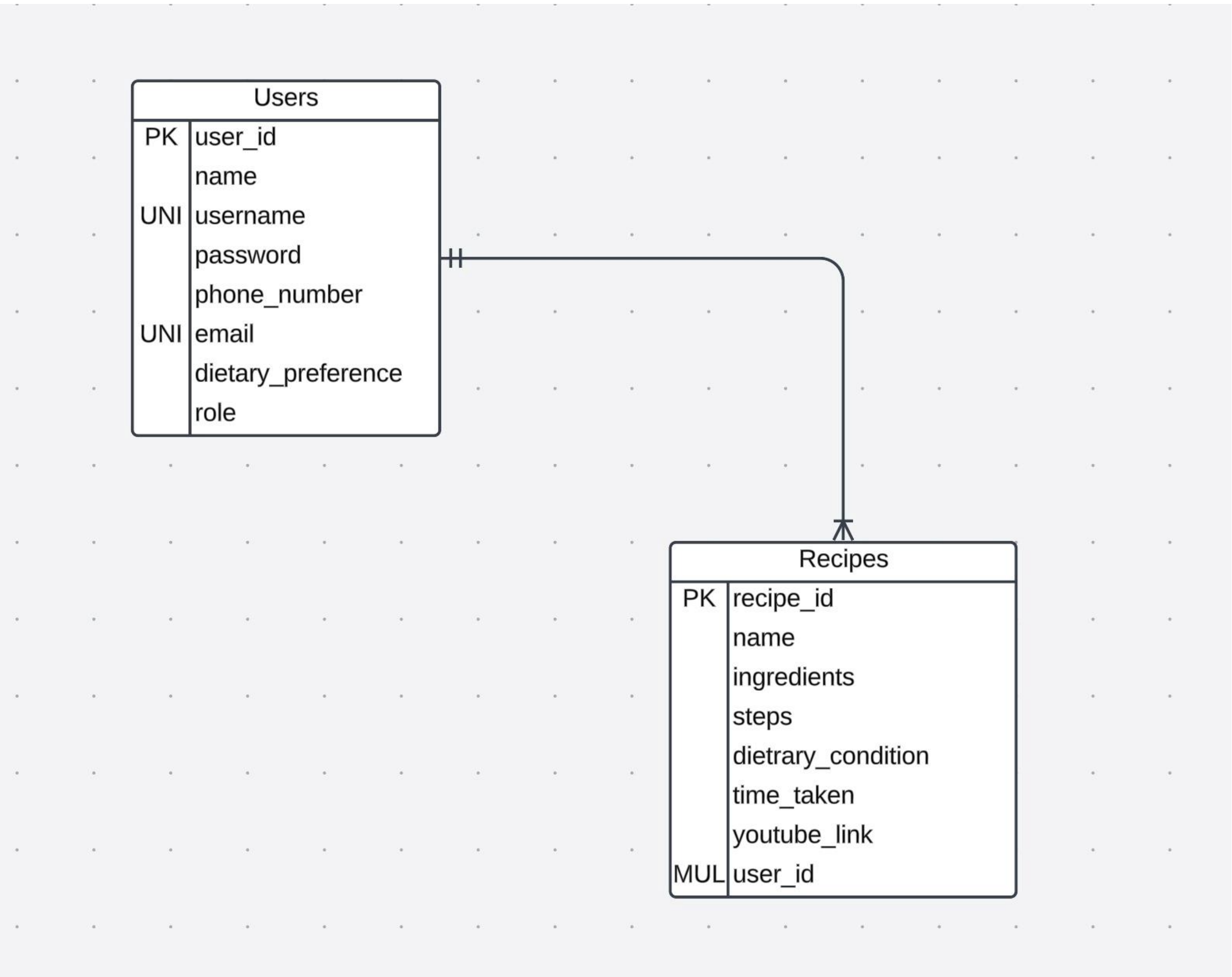
- The system forwards the details for authentication.

2. **Authenticate User:**
    - Validates user credentials against stored records.
    - Grants or denies access based on the authentication result.
3. **Submit Recipe:** o Users submit recipes with details like ingredients and instructions.
    - The system validates the input and forwards it for review.
4. **Search Recipe:**
    - Users search for recipes based on keywords or ingredients.
    - The system retrieves matching results from the database.
5. **Export Recipe:**
    - Users can export or download a recipe in a chosen format. **o** The system formats and provides the recipe for download**.**

Admin Processes:

6. **Add** User: o Admins create new user accounts by entering user details.
    - The system stores the new user profile in the database.
7. **Add Recipe:**
    - Admins manually add recipes to the system for immediate storage.
    - The recipe is validated and then added to the recipe database.
8. **Remove User:** o Admins can remove a user profile from the system.
    - The system deletes the user's account and related data**.**
9. **Remove Recipe:** o Admins delete recipes that are outdated or inappropriate.
    - The system removes the recipe entry from the database.

7.ERD Diagram



1. Admin

- Attributes:
  - Admin_id (Primary Key) ○ Admin_name ○ Email
- **Function:** The "Admin" entity represents system administrators who manage the recipes submitted by users. Admins can oversee and approve or reject recipes in the system.
  - **Relationship:** Admins manage the Recipe entity, helping to maintain the quality and content of submitted recipes.

2.User

- Attributes: ○ User_id (Primary Key) ○ Username ○ Email ○ Password
- **Function:** The "User" entity represents individuals who register in the system. They are responsible for submitting recipes and ingredients.

- **Relationship:**
  - Submits: Users submit both Recipe and Ingredient entities. A user can submit multiple recipes and ingredients.
  - Linked to: Recipes and ingredients created by a specific user are linked back to them through the foreign key User_id.

3.Recipe

- Attributes:
  - Recipe_id (Primary Key) o Recipe_name o Instructions o Dietary_preference
- **Function:** The "Recipe" entity stores information about the recipes submitted by users, including instructions, name, and any dietary preferences.
- **Relationship:**
  - Contains Ingredients: The Recipe entity has a foreign key Ingredient_id that indicates which ingredients are used in the recipe.
  - Submitted by: Recipes are submitted by users, linked via User_id.
  - Managed by: Recipes are managed by admins, linked via Admin_id.

4.Ingredient

- Attributes:
  - Ingredient_id (Primary Key) o Ingredient_name
- **Function:** The "Ingredient" entity stores individual ingredients, which can be used in recipes.
- **Relationship:**
  - Used in Recipes: Ingredients are linked to the Recipe entity through the Ingredient_id foreign key. This shows which recipes use a particular ingredient.
  - Submitted by Users: Ingredients are submitted by users and are linked through the foreign key User_id to track who added them.

Entity Relationships:

- **Admin - Recipe:** Admins manage recipes (one-to-many relationship), ensuring control over the content of the recipe submissions.
- **User - Recipe:** Users submit recipes (one-to-many relationship), where a user can submit many recipes.
- **User - Ingredient:** Users submit ingredients (one-to-many relationship), where a user can submit many ingredients.

- **Recipe - Ingredient:** Recipes are made of one or more ingredients (many-to-many relationship), and this is modeled using foreign keys to show which ingredients are used in a specific recipe.

8.Use Cases

Use Case Name: User Registration

ID: UC-1
Priority: High
Actor: User
Description: The user creates new account by registers as new user.
Trigger: User wants to access the application’s features.
Type: External
Preconditions: 1. The user is not logged in. 2. The system is online and accessible.
Normal Course: User Registration and Login 1. User navigates to the login/registration page. 2. User selects either **Signup** or **Login**. 3. If S**ignup** is selected: a. In a new window User inputs necessary registration details (username, email, password, phone number, email etc.). b. The system validates the registration information. c. If validation is successful, the system creates a new user entry in the **User Database**. d. The user is notified of successful registration and redirected to the login page.
Postconditions: 1. User is registered in successfully. 2. User gains access to relevant features based on their role (user/admin).

Use Case Name : Authenticate User - Login

ID: UC-2

Priority: High

Actor: User

Description: The user or logs in with existing credentials to access the system features.

Trigger: User wants to access the application’s features.

Type: External

Preconditions:

3. The user is not logged in.
4. The system is online and accessible.

Normal Course: Authenticate User -Login

4. User navigates to the login/registration page.
5. User selects **Login**.
6. If **Login** is selected:
   a. In Login Window User enters username and password which are already registered before.
   b. The system checks the credentials against stored data in the **User Database**.
   c. If credentials are valid, the user is authenticated and logged in.
   d. Upon successful login user is notifies a message window as “Welcome, user name!”
   e. The system displays the user dashboard.

Postconditions:

3. User is logged in successfully.
4. User gains access to relevant features based on their role (user/admin).

Use Case Name: Ingredient/Recipe name -Based Recipe Search

ID: UC-3

Priority: High

Actor: User

Description: The user searches for recipes using a list of available ingredients or any already existing recipe.

Trigger: User wants to find recipes based on available ingredients/ recipe name.

Type: External

Preconditions:

1. The user is authenticated and logged in.
2. The user has a list of available ingredients/ recipe name to be searched.

Normal Course: 1.0 Ingredient-Based Recipe Search

1. Click Search Recipe in the dashboard.
2. User inputs a recipe name or list of ingredients they currently have.
3. User also selects **Dietary concerns** such as vegan, vegetarian etc and **Time Taken** such less than 15 mins, 30mins etc.
4. On clicking **Search Button** the system verifies the input for valid ingredient/recipe name entries.
5. The system searches the **Recipe Database** for recipes that match the input ingredients.
6. Matching recipes are displayed to the user, sorted by relevance.
7. User can view the recipe.

Postconditions:

1. A list of matching recipes is displayed to the user.
2. User can choose to save or download the recipe.

Use Case Name: Add recipe

ID: UC-4

Priority: Medium

Actor: User

Description: The user submits a new recipe to the system, including details such as ingredients, instructions, and dietary preferences.

Trigger: User wants to contribute a new recipe to the community.

Type: External

Preconditions:

1. The user is logged in.
2. The user has the necessary recipe details ready to input.

Normal Course: Recipe Submission 1. On Clicking **Add Recipe** user navigates to the recipe submission page. 2. User enters the recipe name, ingredients, cooking instructions, and any relevant dietary information , time taken for the recipe and clicks on Add Recipe. 3. The system validates the recipe details (e.g., required fields are filled). 4. If the validation is successful, the recipe is stored in the **Recipe Database**. 5. A confirmation message is displayed, and the recipe is made available for future searches.
Postconditions: 1. Recipe is saved in the **Recipe Database**. 2. Recipe becomes searchable and visible to other users.

Use Case Name: View My Recipe

ID: UC-5
Priority: Medium
Actor: User
Description: The user views the recipes they have submitted.
Trigger: User wants to see a list of their added recipes.
Type: External
Preconditions: User is logged in
Normal Course: View My Recipes 1. On clicking **View Recipes** user navigates to the "My Recipes" section. 2. The system retrieves the user's recipes from the database. 3. The system displays the list of recipes to the user.
Postconditions: 1.User can view their submitted recipes.

Use Case Name: Download/Export Recipe

ID: UC-6

Priority: Medium
Actor: User
Description: The user downloads or exports selected recipes in a pdf format.
Trigger: User wants to save/download a recipe locally.
Type: External
Preconditions: 1.User is logged in. 2.Recipes are available in the system.
Normal Course: Export Recipes 1. In **View Recipes** user navigates to the recipe they want to download/export. 2. User selects the download/export option. 3. The system generates the file and makes it available for download. 4. User downloads the file.
Postconditions: 1. Recipe is successfully downloaded/exported in the selected format.

Use Case Name: Admin Authentication- Login

ID: UC-7
Priority: High
Actor: Admin
Description: The admin logs in with their credentials to manage system features.
Trigger: Admin wants to access the system's administrative features
Type: External
Preconditions: 1. Admin account exists in the system. 2. The system is logged in and accessible..

Normal Course:

1. Admin navigates to the login page.

2. Admin inputs username and password.

3.The system validates the credentials against stored data in the Admin Database.

4.If credentials are valid, the admin is authenticated and logged in.

5.The system displays the admin dashboard.

Postconditions:

1. Admin is logged in successfully.
2. Admin gains access to administrative features.

Use Case Name: Admin Add Recipe

ID: UC-8

Priority: High

Actor: User

Description: Admin adds a new recipe to the system for all users to access.

Trigger: Admin wants to publish a new recipe.

Type: External

Preconditions:

1. Admin is logged in.

Normal Course: Add Recipe

1. Admin navigates to the "Add Recipe" section.

2. Admin fills in the recipe details (e.g., name, ingredients, procedure, dietary conditions, time taken).

3. Admin submits the recipe.

4. The system validates the data and saves the recipe in the database.

5. Admin receives a confirmation that the recipe has been added successfully.

Postconditions: 1. Recipe is stored in the system and is visible to users.

Use Case Name: Admin Edit Recipe

ID: UC-9
Priority: High
Actor: User
Description: Admin edits an existing recipe's details
Trigger: Admin wants to update a recipe's information.
Type: External
Preconditions: 1.Admin is logged in. 2. Recipe exists in the system.
Normal Course: 1. Admin navigates to the recipe to be edited. 2. Admin clicks the "Edit" button. 3. Admin modifies the desired details (e.g., name, procedure, ingredients, dietary conditions, time taken). 4. Admin clicks the "Save Changes" button. 5. The system updates the recipe details in the database
Postconditions: 1. Recipe details are updated successfully.

Use Case Name: Admin Delete Recipe

ID: UC-10
Priority: Medium
Actor: User
Description: Admin deletes a recipe from the system.
Trigger: Admin wants to remove an existing recipe.
Type: External
Preconditions: 1.Admin is logged in.

2. Recipe exists in the system.

Normal Course: Delete Recipe

1. Admin navigates to the recipe to be deleted.
2. Admin clicks the "Delete" button.
3. A confirmation box appears asking, "Are you sure?" with "Yes" and "No" options.
4. Admin clicks "Yes."
5. The system removes the recipe from the database.

Postconditions:

1. Recipe is deleted from the system.

Use Case Name: Admin Edit User Role

ID: UC-11

Priority: Medium

Actor: User

Description: Admin updates a user's role.

Trigger: Admin wants to modify user access

Type: External

Preconditions:
1. Admin is logged in.
2. User exists in the system.

Normal Course: Edit User Role

1. Admin navigates to the "User Management" section.
2. Admin selects a user to modify.
3. Admin changes the user role to "Admin" or "User."
4. Admin clicks the "Save Role" button.
5. The system updates the user's role in the database.

Postconditions: 1. User role is updated successfully.

Use Case Name: Admin Delete User Role

ID: UC-12
Priority: Medium
Actor: User
Description: Admin removes a user from the system.
Trigger: Admin wants to delete a user account.
Type: External
Preconditions: 1. Admin is logged in. 2. User exists in the system.
Normal Course: Delete User Role 1. Admin navigates to the "User Management" section. 2. Admin selects a user to delete. 3. Admin clicks the "Delete" button. 4. A confirmation box appears asking, "Are you sure?" with "Yes" and "No" options. 5. Admin clicks "Yes." 6. The system removes the user account from the database.
Postconditions: 1. User account is deleted from the system.

Use Case Name: Admin View Insights

ID: UC-13
Priority: Medium

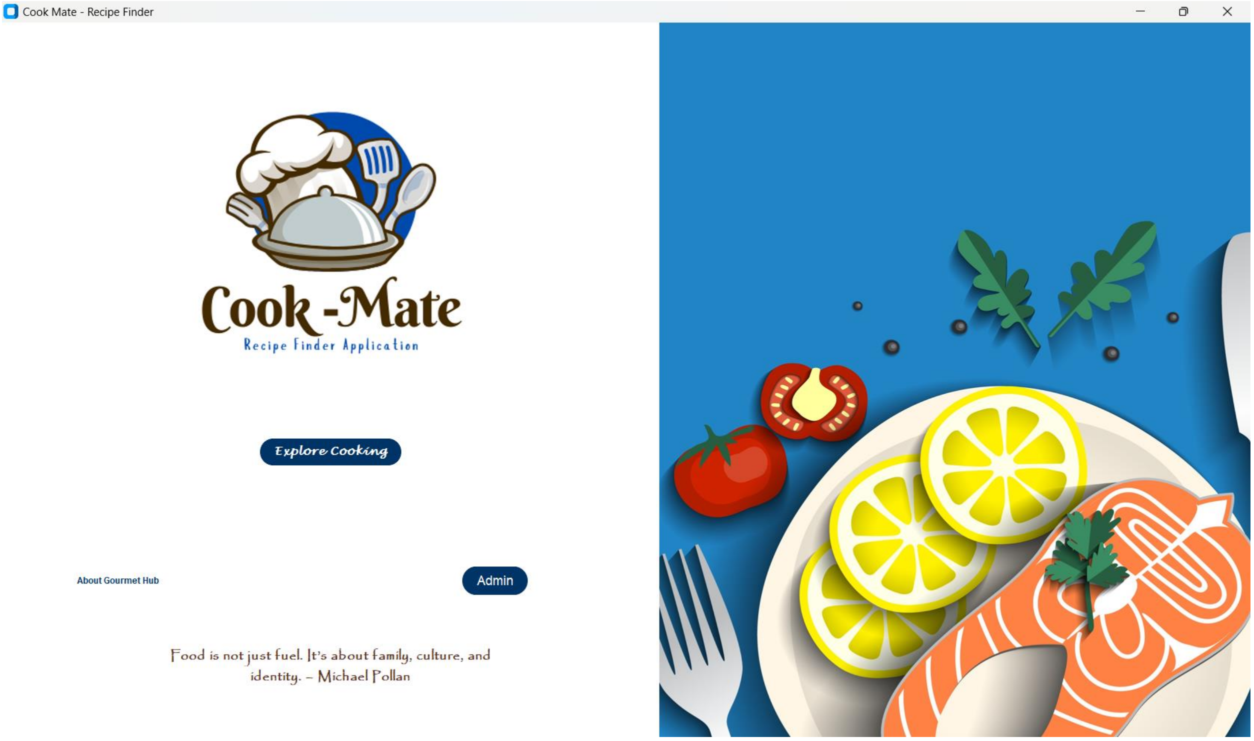
Actor: User
Description: Admin views insights and analytics related to recipes and system usage
Trigger Admin wants to analyze system data.
Type: External
Preconditions: 1. Admin is logged in.
Normal Course: View Insights 1. Admin navigates to the "Insights" section. 2. The system displays data such as: a. Recipes by dietary conditions. b. Recipes by time taken. 3. Admin clicks the "Download Insights" button to save insights as images.
Postconditions: 1. Insights are displayed and optionally downloaded as images.

Use Case Name: Generating Reports

ID: UC-14
Priority: Medium
Actor: User
Description: Admin generates a detailed analytics report for the system.
Trigger : Admin wants a comprehensive report on system usage.
Type: External
Preconditions: 1. Admin is logged in.
Normal Course: Generate Reports 1. Admin navigates to the "Reports" section. 2. Admin clicks the "Generate Report" button. 3. A message box appears stating, "CookMate Analytic Report generated successfully." 4. The system generates a Word document with: a. Executive summary.

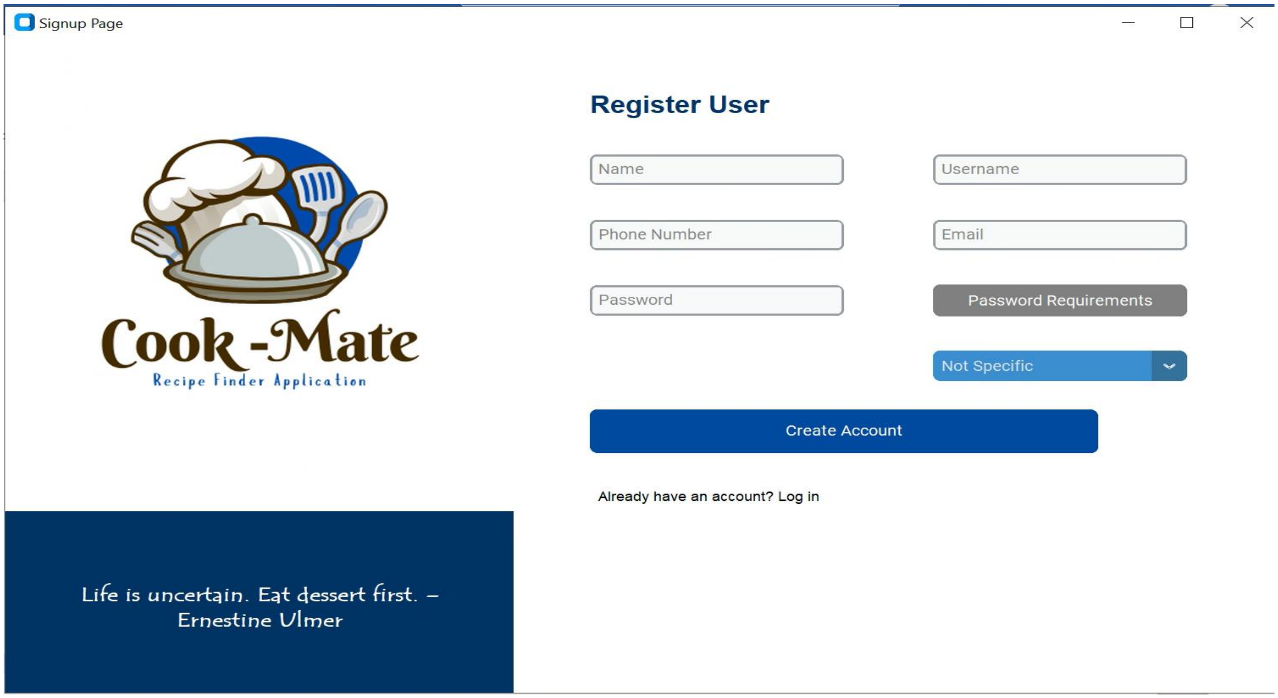
b. Recipe analytics. c. User demographics analysis. d. Top contributors. 5. Admin downloads the report.
Postconditions: 1. Report is generated and downloaded successfully.

9. APPLICATION SCREENS



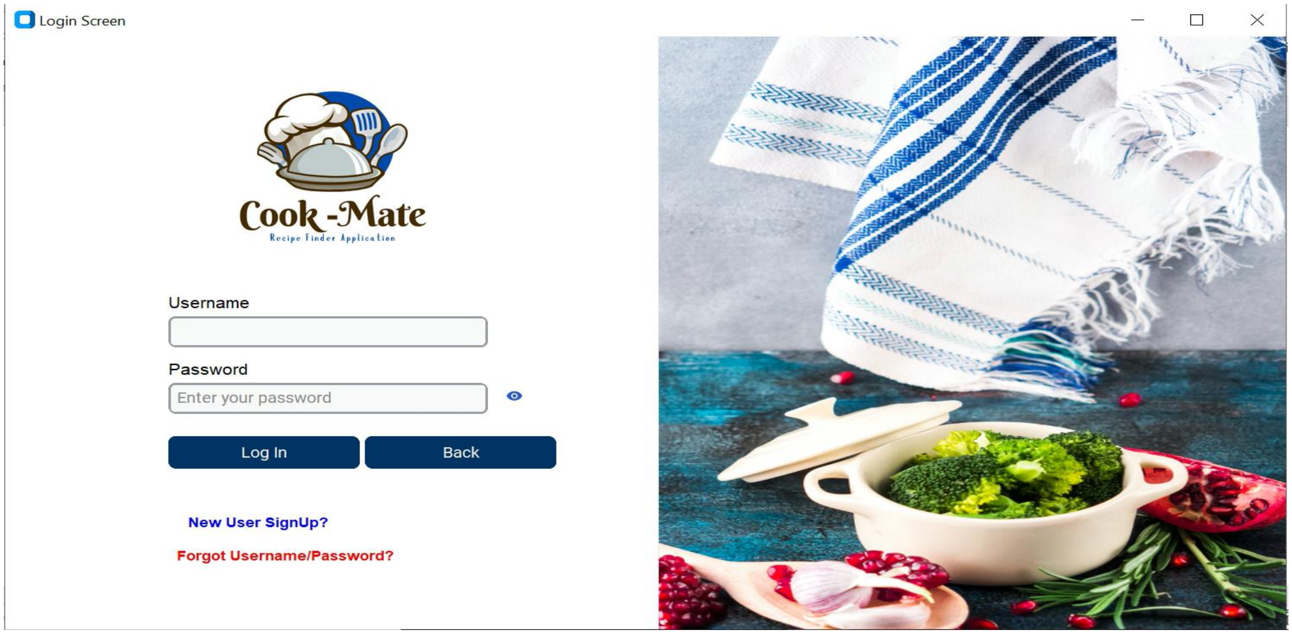
Food is not just fuel. It's about family, culture, and identity. – Michael Pollan

HOMEPAGE

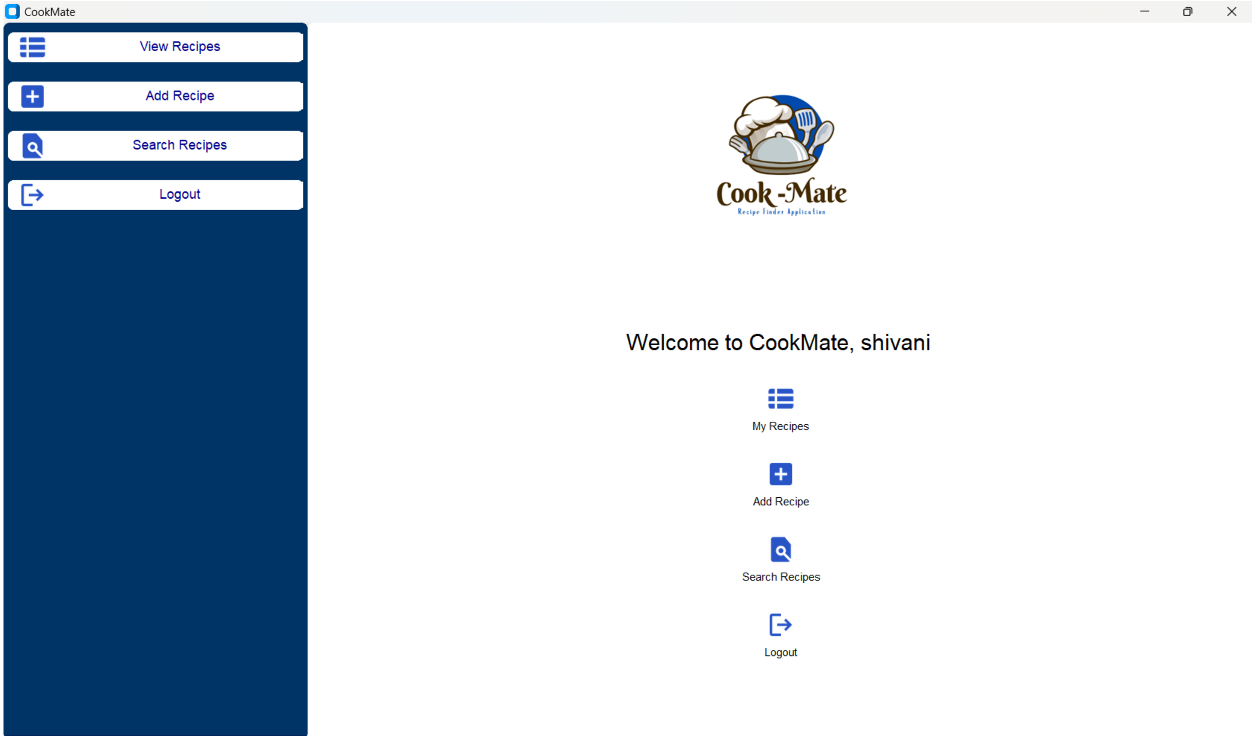


Life is uncertain. Eat dessert first. – Ernestine Ulmer

SIGN UP SCREEN

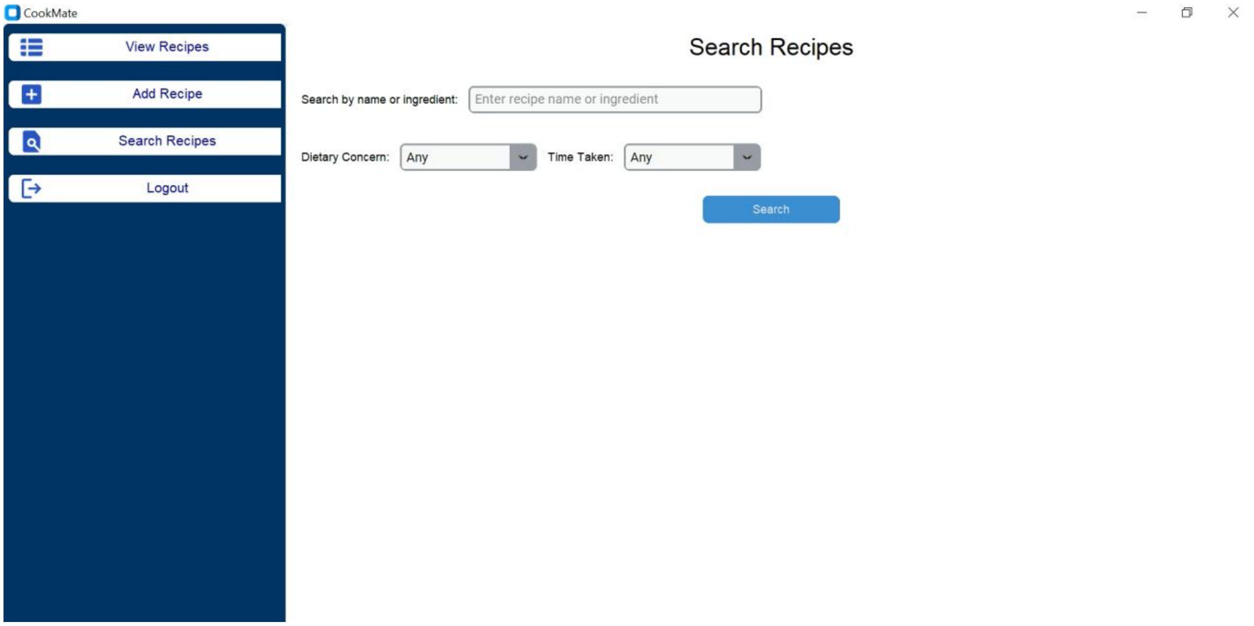


LOGIN SCREEN



Welcome to CookMate, shivani

USER DASHBOARD

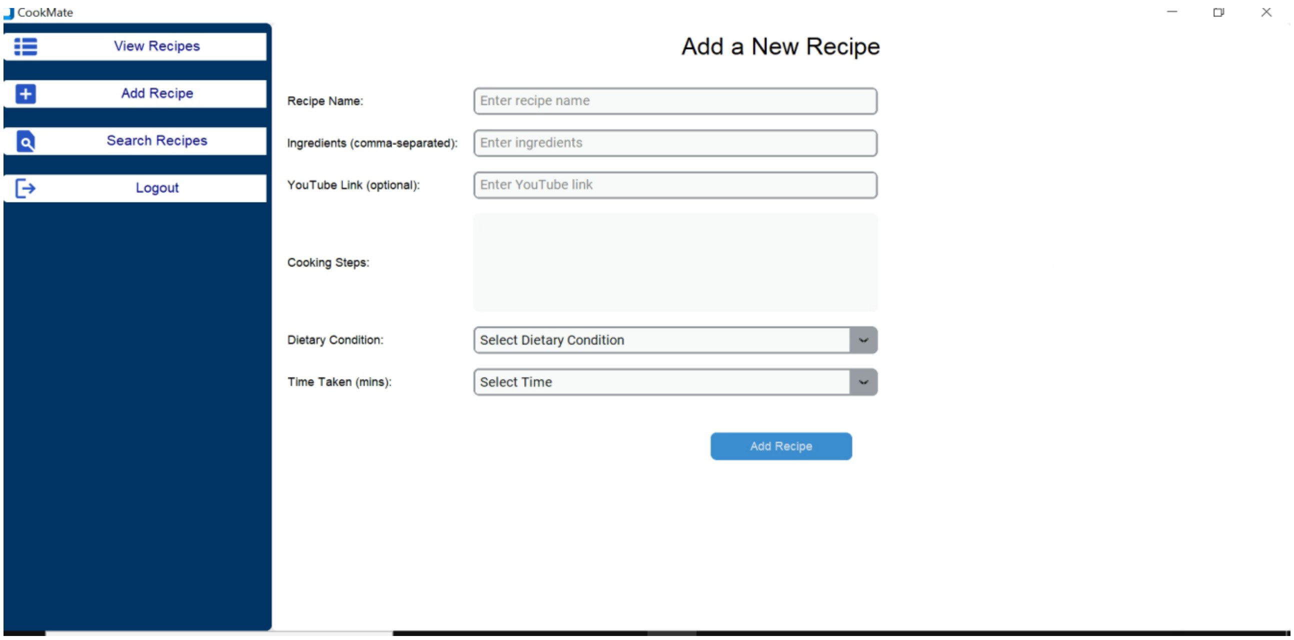


Search Recipes

Search by name or ingredient: Enter recipe name or ingredient

Dietary Concern: Any Time Taken: Any

SEARCH RECIPES



Add a New Recipe

Recipe Name: Enter recipe name

Ingredients (comma-separated): Enter ingredients

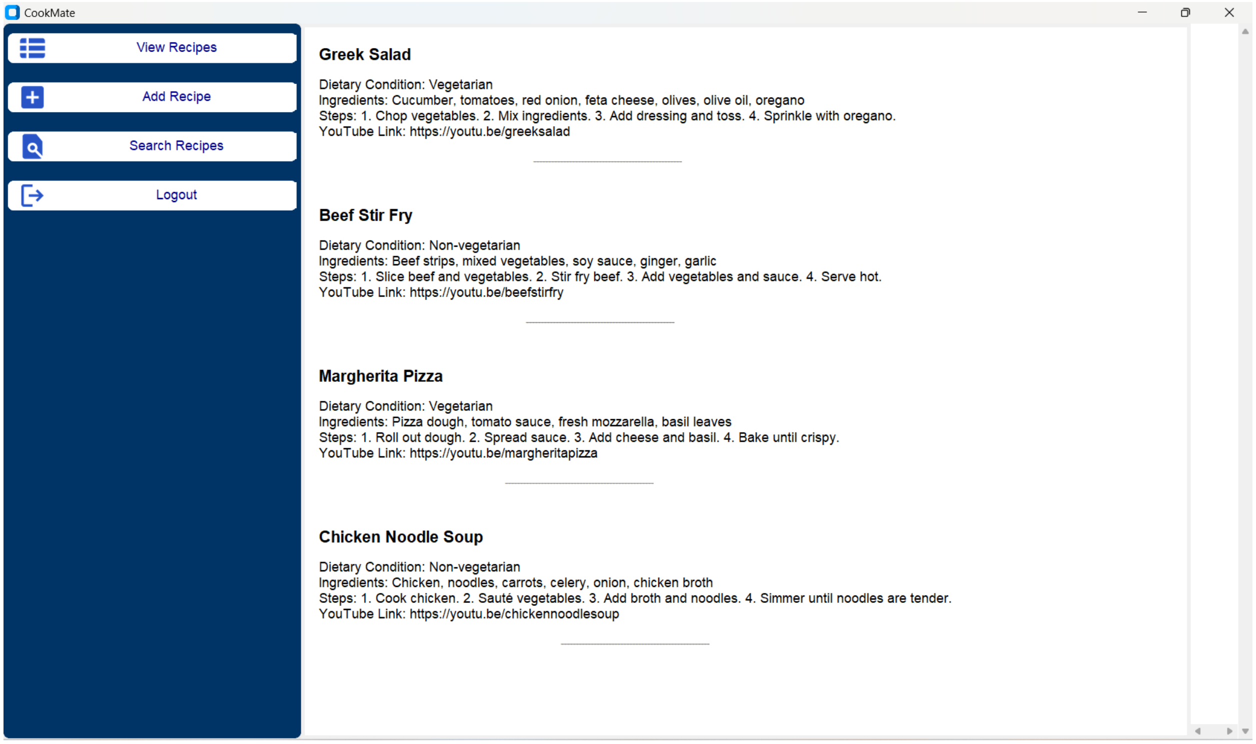
YouTube Link (optional): Enter YouTube link

Cooking Steps:

Dietary Condition: Select Dietary Condition

Time Taken (mins): Select Time

ADD NEW RECIPE



Greek Salad

Dietary Condition: Vegetarian
Ingredients: Cucumber, tomatoes, red onion, feta cheese, olives, olive oil, oregano
Steps: 1. Chop vegetables. 2. Mix ingredients. 3. Add dressing and toss. 4. Sprinkle with oregano.
YouTube Link: https://youtu.be/greeksalad

Beef Stir Fry

Dietary Condition: Non-vegetarian
Ingredients: Beef strips, mixed vegetables, soy sauce, ginger, garlic
Steps: 1. Slice beef and vegetables. 2. Stir fry beef. 3. Add vegetables and sauce. 4. Serve hot.
YouTube Link: https://youtu.be/beefstirfry

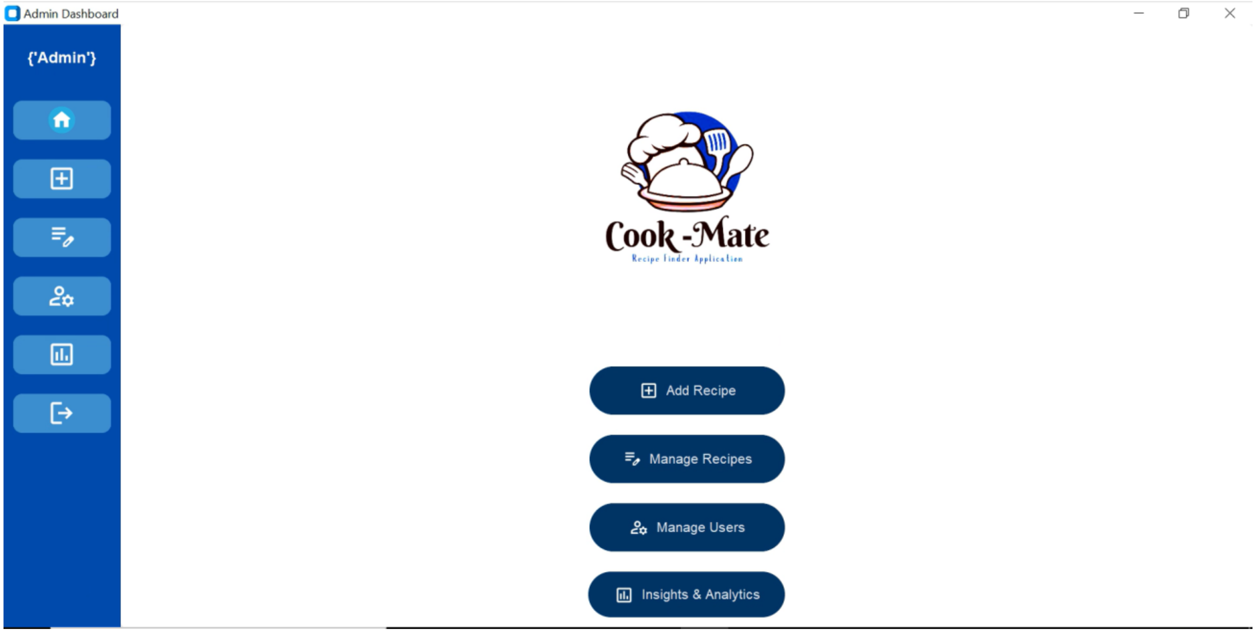
Margherita Pizza

Dietary Condition: Vegetarian
Ingredients: Pizza dough, tomato sauce, fresh mozzarella, basil leaves
Steps: 1. Roll out dough. 2. Spread sauce. 3. Add cheese and basil. 4. Bake until crispy.
YouTube Link: https://youtu.be/margheritapizza

Chicken Noodle Soup

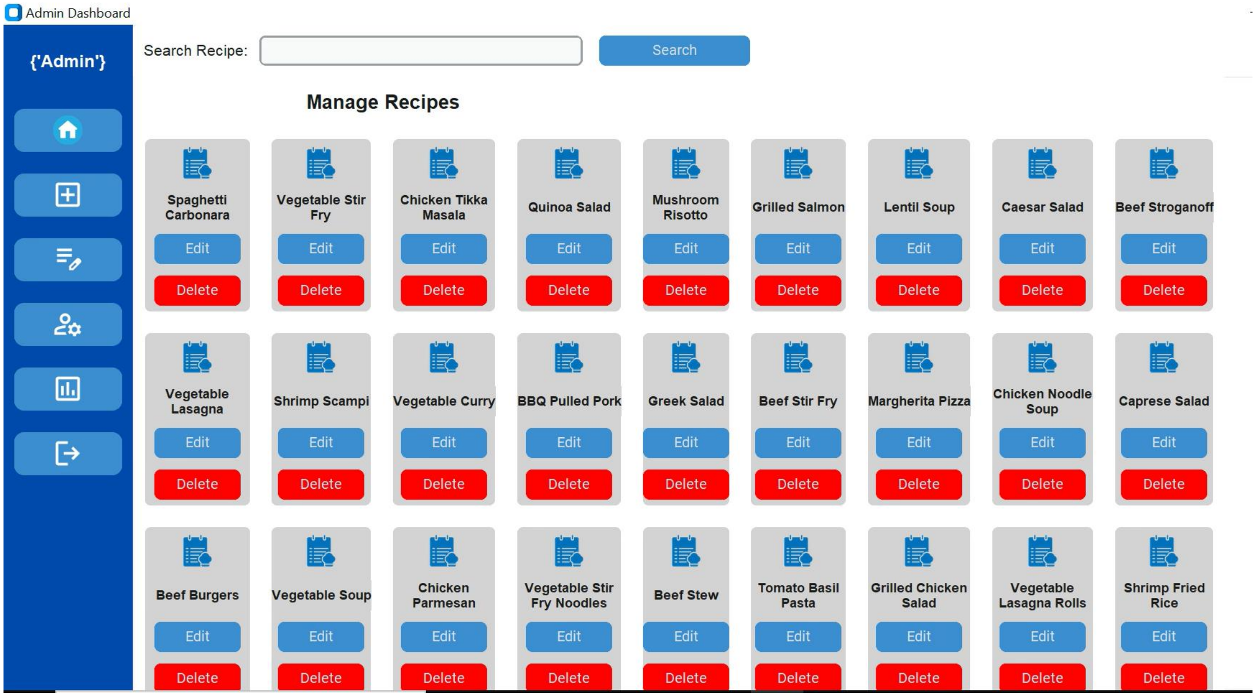
Dietary Condition: Non-vegetarian
Ingredients: Chicken, noodles, carrots, celery, onion, chicken broth
Steps: 1. Cook chicken. 2. Sauté vegetables. 3. Add broth and noodles. 4. Simmer until noodles are tender.
YouTube Link: https://youtu.be/chickennoodlesoup

VIEW MY RECIPE



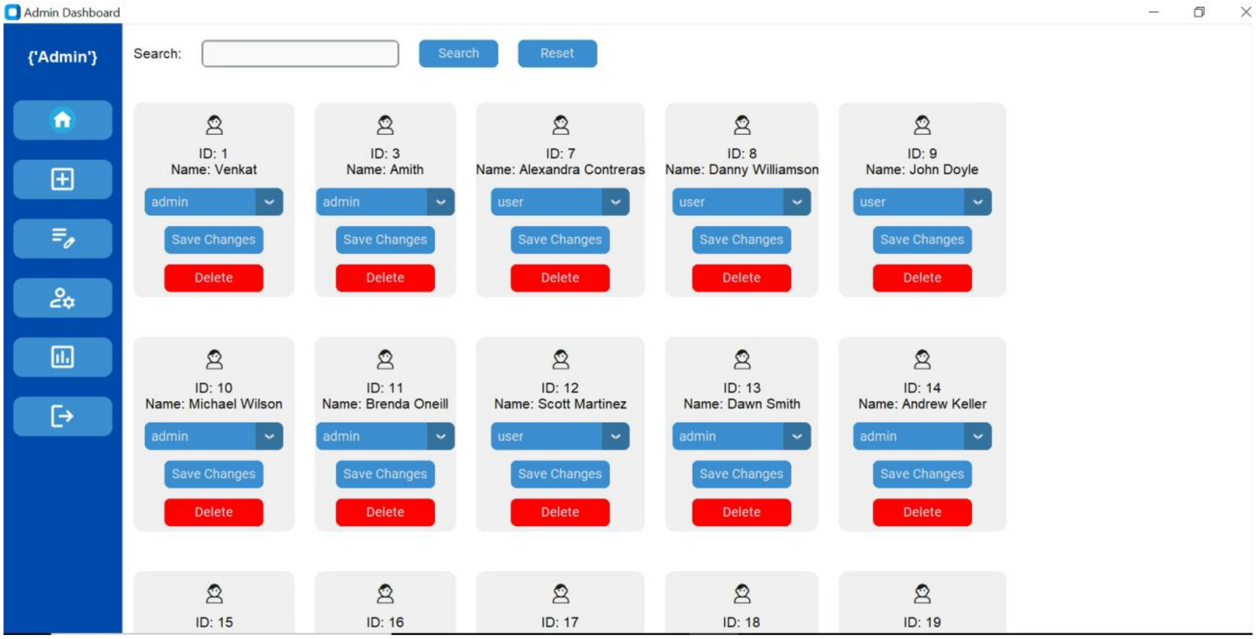
ADMIN LOGIN

We have hardcoded the values for admin to login
Username: Admin/admin
Password: Admin@123



Spaghetti Carbonara	Vegetable Stir Fry	Chicken Tikka Masala	Quinoa Salad	Mushroom Risotto	Grilled Salmon	Lentil Soup	Caesar Salad	Beef Stroganoff
Vegetable Lasagna	Shrimp Scampi	Vegetable Curry	BBQ Pulled Pork	Greek Salad	Beef Stir Fry	Margherita Pizza	Chicken Noodle Soup	Caprese Salad
Beef Burgers	Vegetable Soup	Chicken Parmesan	Vegetable Stir Fry Noodles	Beef Stew	Tomato Basil Pasta	Grilled Chicken Salad	Vegetable Lasagna Rolls	Shrimp Fried Rice

MANAGE RECIPES



ID: 1
Name: Venkat
admin

ID: 3
Name: Amith
admin

ID: 7
Name: Alexandra Contreras
user

ID: 8
Name: Danny Williamson
user

ID: 9
Name: John Doyle
user

ID: 10
Name: Michael Wilson
admin

ID: 11
Name: Brenda Oneill
admin

ID: 12
Name: Scott Martinez
user

ID: 13
Name: Dawn Smith
admin

ID: 14
Name: Andrew Keller
admin

ID: 15

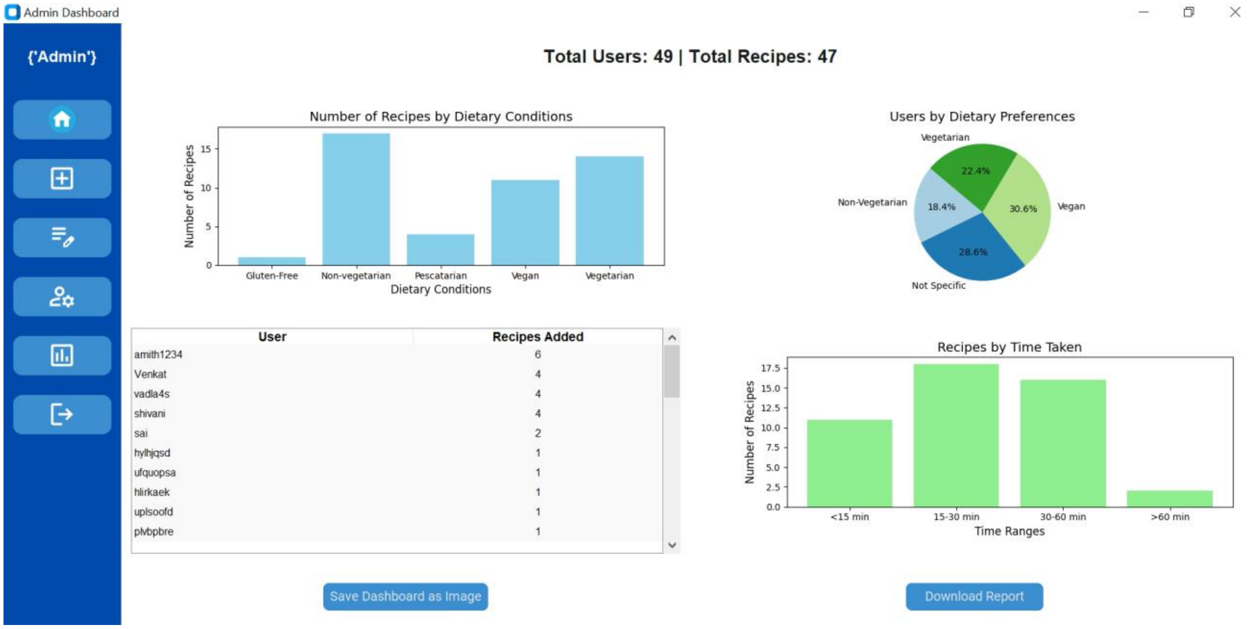
ID: 16

ID: 17

ID: 18

ID: 19

MANAGE USERS



Total Users: 49 | Total Recipes: 47

User	Recipes Added
amith1234	6
Venkat	4
vadla4s	4
shivani	4
sai	2
hylhjqsd	1
ufquopsa	1
hlirkaek	1
uplsoofd	1
plvbpbre	1

INSIGHTS AND ANALYTICS

10.Task List

Phase 1: Project Proposal (10 days)

Task 1	Initiate Project Proposal	1 Day
Task 2	Define System Requirements	2 Days
Task 3	Draft Project Proposal Document	3 Days
Task 4	Project Proposal Review and Approval	2 Days

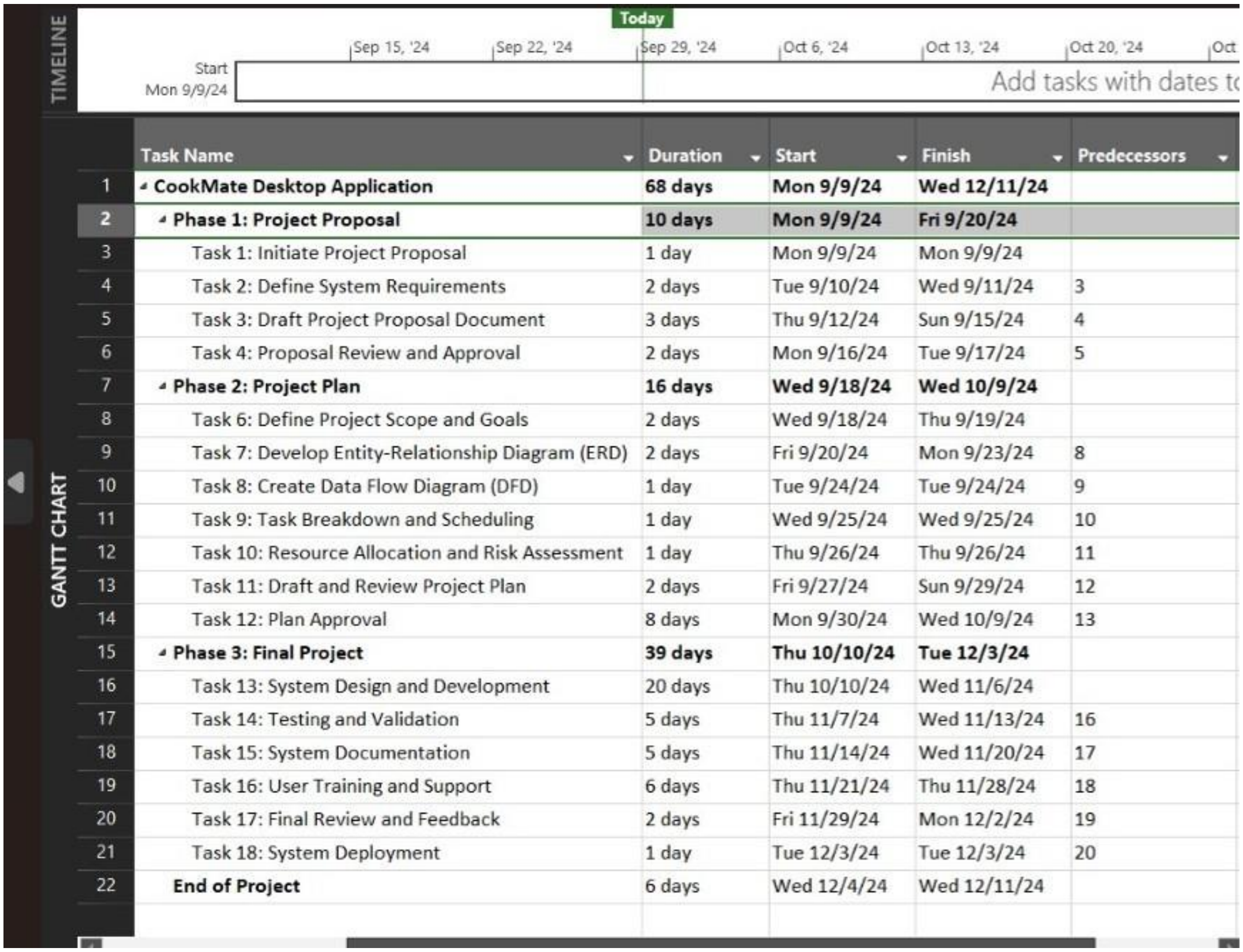
Phase 2: Project Plan (16 days)

Task 6	Define Project Scope and Goals	2 Days
Task 7	Develop Entity-Relationship Diagram (ERD)	2 Days
Task 8	Create Data Flow Diagram (DFD)	1 Day
Task 9	Task Breakdown and Scheduling	1 Day
Task 10	Resource Allocation and Risk Assessment	2 Days
Task 11	Draft and Review Project Plan	1 Day
Task12	Project Plan Approved	8 Days

Phase 3: Final Project (39 days)

Task 13	System Design and Development	20 Days
Task 14	Testing and Validation	5 Days
Task 15	System Documentation	5 Days
Task 16	User Training and Support	6 Days
Task 17	Final Review and Feedback	2 Days
Task 18	System Deployment Delivered	1 Day

11.Project Schedule



	Task Name	Duration	Start	Finish	Predecessors
1	**CookMate Desktop Application**	**68 days**	**Mon 9/9/24**	**Wed 12/11/24**	
2	**Phase 1: Project Proposal**	**10 days**	**Mon 9/9/24**	**Fri 9/20/24**	
3	Task 1: Initiate Project Proposal	1 day	Mon 9/9/24	Mon 9/9/24	
4	Task 2: Define System Requirements	2 days	Tue 9/10/24	Wed 9/11/24	3
5	Task 3: Draft Project Proposal Document	3 days	Thu 9/12/24	Sun 9/15/24	4
6	Task 4: Proposal Review and Approval	2 days	Mon 9/16/24	Tue 9/17/24	5
7	**Phase 2: Project Plan**	**16 days**	**Wed 9/18/24**	**Wed 10/9/24**	
8	Task 6: Define Project Scope and Goals	2 days	Wed 9/18/24	Thu 9/19/24	
9	Task 7: Develop Entity-Relationship Diagram (ERD)	2 days	Fri 9/20/24	Mon 9/23/24	8
10	Task 8: Create Data Flow Diagram (DFD)	1 day	Tue 9/24/24	Tue 9/24/24	9
11	Task 9: Task Breakdown and Scheduling	1 day	Wed 9/25/24	Wed 9/25/24	10
12	Task 10: Resource Allocation and Risk Assessment	1 day	Thu 9/26/24	Thu 9/26/24	11
13	Task 11: Draft and Review Project Plan	2 days	Fri 9/27/24	Sun 9/29/24	12
14	Task 12: Plan Approval	8 days	Mon 9/30/24	Wed 10/9/24	13
15	**Phase 3: Final Project**	**39 days**	**Thu 10/10/24**	**Tue 12/3/24**	
16	Task 13: System Design and Development	20 days	Thu 10/10/24	Wed 11/6/24	
17	Task 14: Testing and Validation	5 days	Thu 11/7/24	Wed 11/13/24	16
18	Task 15: System Documentation	5 days	Thu 11/14/24	Wed 11/20/24	17
19	Task 16: User Training and Support	6 days	Thu 11/21/24	Thu 11/28/24	18
20	Task 17: Final Review and Feedback	2 days	Fri 11/29/24	Mon 12/2/24	19
21	Task 18: System Deployment	1 day	Tue 12/3/24	Tue 12/3/24	20
22	**End of Project**	6 days	Wed 12/4/24	Wed 12/11/24	

Phase 1: Project Proposal (10 days)

- **Task 1: Initiate Project Proposal (1 day):** o Meet with stakeholders to discuss project goals and objectives. o Identify project scope and deliverables.
  - Assign roles and responsibilities to team members.
- **Task 2: Define System Requirements (2 days):**
  - Gather requirements from stakeholders and end-users.
  - Create a detailed list of functional and non-functional requirements.
- **Task 3: Draft Project Proposal Document (3 days):**

- Write a comprehensive project proposal document outlining project scope, objectives, deliverables, timeline, and budget.
- Include sections on project management methodology, risk assessment, and communication plan.

- **Task 4: Project Proposal Review and Approval (2 days):**
  - Present the draft project proposal to stakeholders for review and feedback. o Incorporate feedback and finalize the project proposal.
  - Obtain approval from key stakeholders.

Milestone 1 – Project Proposal Approved

Phase 2: Project Plan (16 days)

- **Task 6: Define Project Scope and Goals (2 days):**
  - Refine project scope and goals based on feedback from stakeholders.
  - Create a detailed project scope statement.
- **Task 7: Develop Entity-Relationship Diagram (ERD) (2 days):**
  - Design the database structure for the Cook Mate application.
  - Create an ERD to visualize the relationships between entities.
- **Task 8: Create Data Flow Diagram (DFD) (1 day):**
  - Model the data flow within the application.
  - Identify data inputs, processes, and outputs.

Task 9: Task Breakdown and Scheduling (1 day):

- Break down the project into smaller, manageable tasks.
- Create a detailed project schedule using a Gantt chart or other suitable tool.

- **Task 10: Resource Allocation and Risk Assessment (2 days):**
  - Allocate necessary resources (human, financial, and technical) to each task.
  - Identify potential risks and develop mitigation strategies.
- **Task 11: Draft and Review Project Plan (1 day):**
  - Write a comprehensive project plan document.
  - Review the plan with stakeholders to ensure alignment with project objectives.

- **Task 12: Project Plan Approved (8 days):**
    - Obtain approval for the project plan from key stakeholders.
    - Make necessary revisions based on feedback.

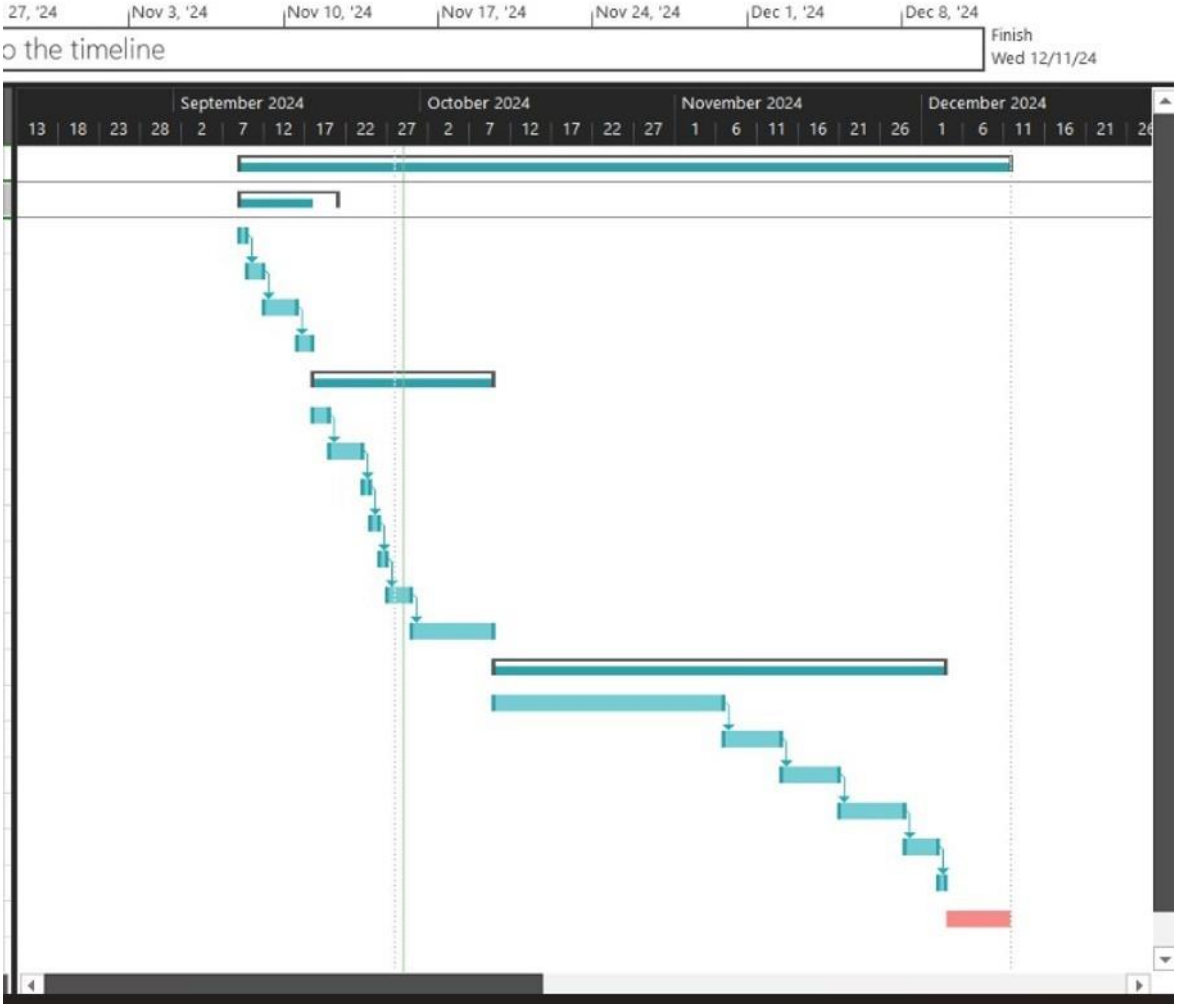
Milestone 2 – Project Plan Approved

Phase 3: Final Project (39 days)

- **Task 13: System Design and Development (20 days):** o Develop the application's architecture and design. o Implement the system using Python and Tkinter.
    - Integrate the database (MySQL) with the application.
- **Task 14: Testing and Validation (5 days):**
    - Conduct unit testing, integration testing, and system testing.
    - Identify and fix defects.
- **Task 15: System Documentation (5 days):** o Create user manuals and technical documentation for the application.
- **Task 16: User Training and Support (6 days):** o Develop training materials for users. o Provide training sessions for users.
    - Establish a support system for users.
- **Task 17: Final Review and Feedback (2 days):**
    - Conduct a final review of the application with stakeholders.
    - Gather feedback and address any remaining issues.
- **Task 18: System Deployment Delivered (1 day):**
    - Deploy the Cook Mate application to the production environment.

Milestone 3 – Final Project Delivered

12.Critical Path



The tasks included in the critical path are the ones that take more time to complete, which determines how long the project will take in total. The key path in the above Gantt chart consists of tasks related to project planning, system development, testing, and deployment. The project completion date will be directly impacted by any delays in these tasks. To guarantee that these tasks are finished on time, it is important that they are well managed and evaluated.

13. Conclusion

The Smart Recipe Finder (Cook Mate) project is set to significantly advance The Gourmet Hub's mission to make cooking accessible and enjoyable for all. By offering a desktop application that tailors recipe suggestions based on available ingredients, Cook Mate will simplify meal planning, reduce food waste, and inspire creative cooking. The app’s community-driven features will enhance user engagement through recipe sharing, instructional videos, and reviews, fostering a supportive and interactive environment. Additionally, Cook Mate’s filtering options for dietary needs will ensure inclusivity for diverse nutritional requirements. This project not only promises to enrich the culinary experience for users but also supports The Gourmet Hub’s growth by driving engagement and exploring new revenue streams. Overall, Cook Mate represents a strategic initiative to blend personalized cooking experiences with community collaboration, enhancing both user satisfaction and organizational impact.