



UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

“Recipe Book Management System”

Submitted by:

Meghana N	<PES1UG21CS334>
Mullapudi Jahnavi	<PES1UG21CS346>
N Shreelekha	<PES1UG21CS353>
Mrunal Kulkarni	<PES2UG21CS304>

6th Semester F Section

Prof. Bhargavi Mokashi

Assistant Professor

January - May 2024

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

- Problem statement (synopsis):

1. Overview:

Ever found yourself scrolling through endless recipe websites, trying to decide what to cook for dinner? It's a hassle, right? Well, that's where our Recipe Book Management System swoops in to save the day! Whether you're a cooking enthusiast or just trying to spice up your meals, our platform is like your culinary sidekick, making it easy-peasy to find and organize recipes.

2. Problem Description:

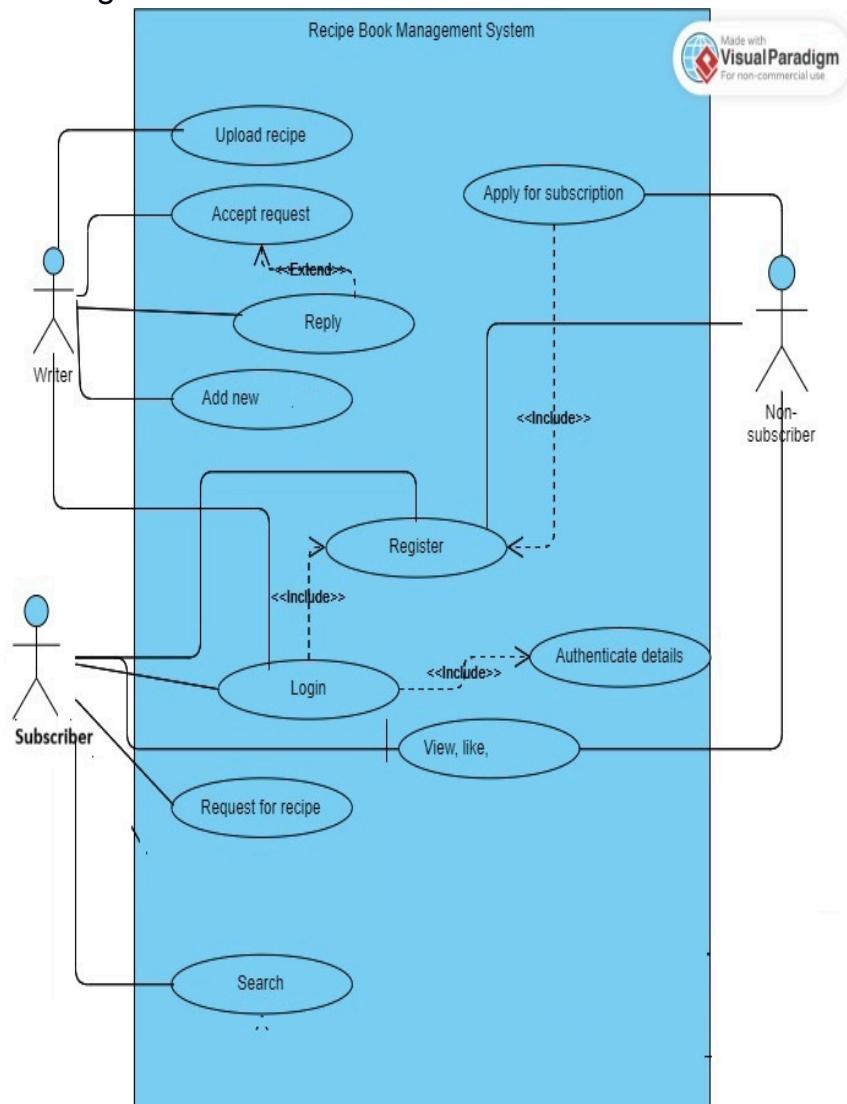
Here's the scoop: flipping through cookbooks or searching online for recipes can be a real headache. But fear not! Our system is your digital recipe hub, keeping all your favorite dishes neatly organized in one place. Plus, if there's a specific recipe you're craving, you can shoot a request to our resident writer and voila! Your culinary dreams come true. It's like having your own personal chef on speed dial!

3. Significance:

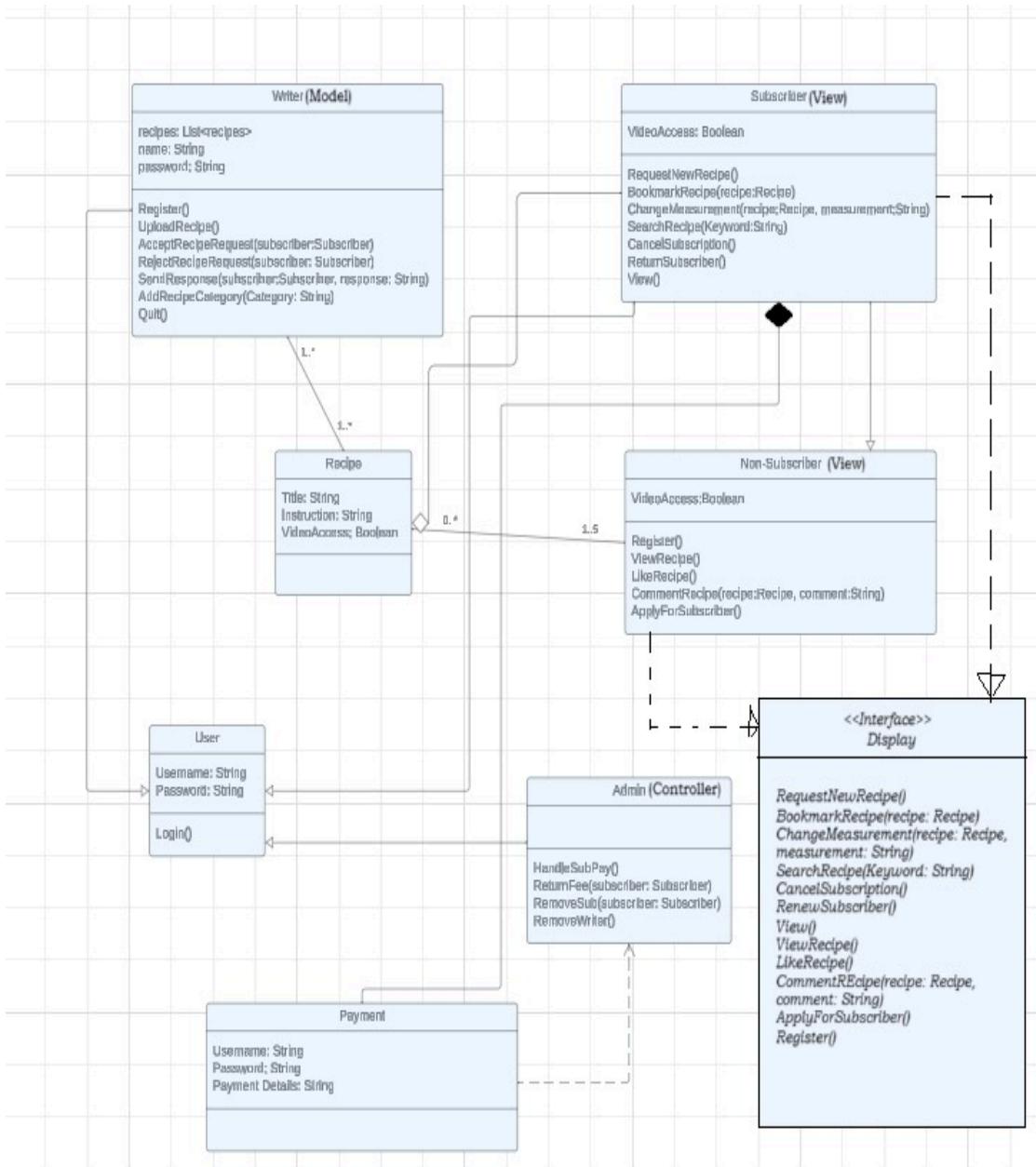
So, why should you care? Well, imagine waving goodbye to kitchen chaos and hello to stress-free cooking. Our Recipe Book Management System isn't just about finding recipes - it's about making your life easier and your meals tastier. With everything at your fingertips, you'll spend less time searching for recipes and more time enjoying delicious dishes. It's a win-win!

- Models (Use Case, Class Models, State & Activity):

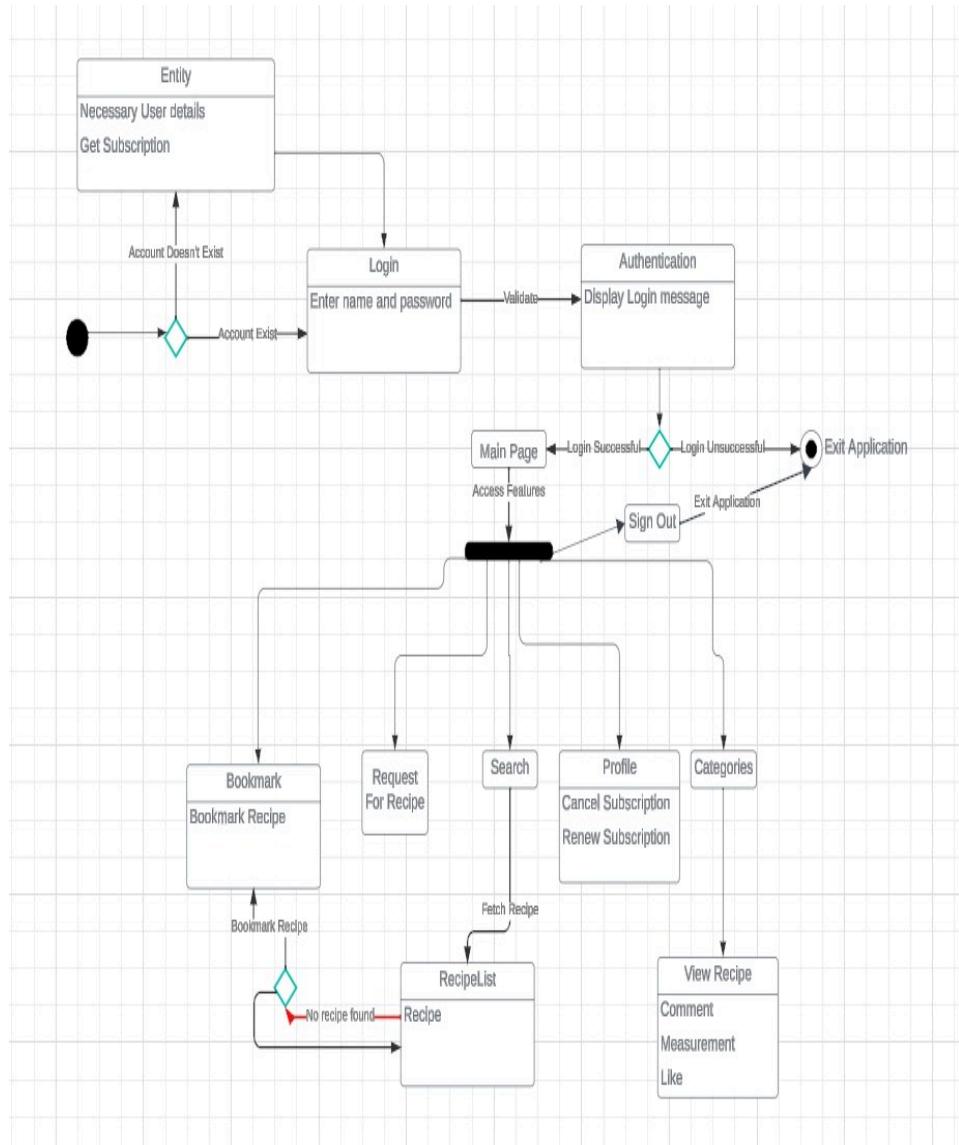
1. Use case diagram:



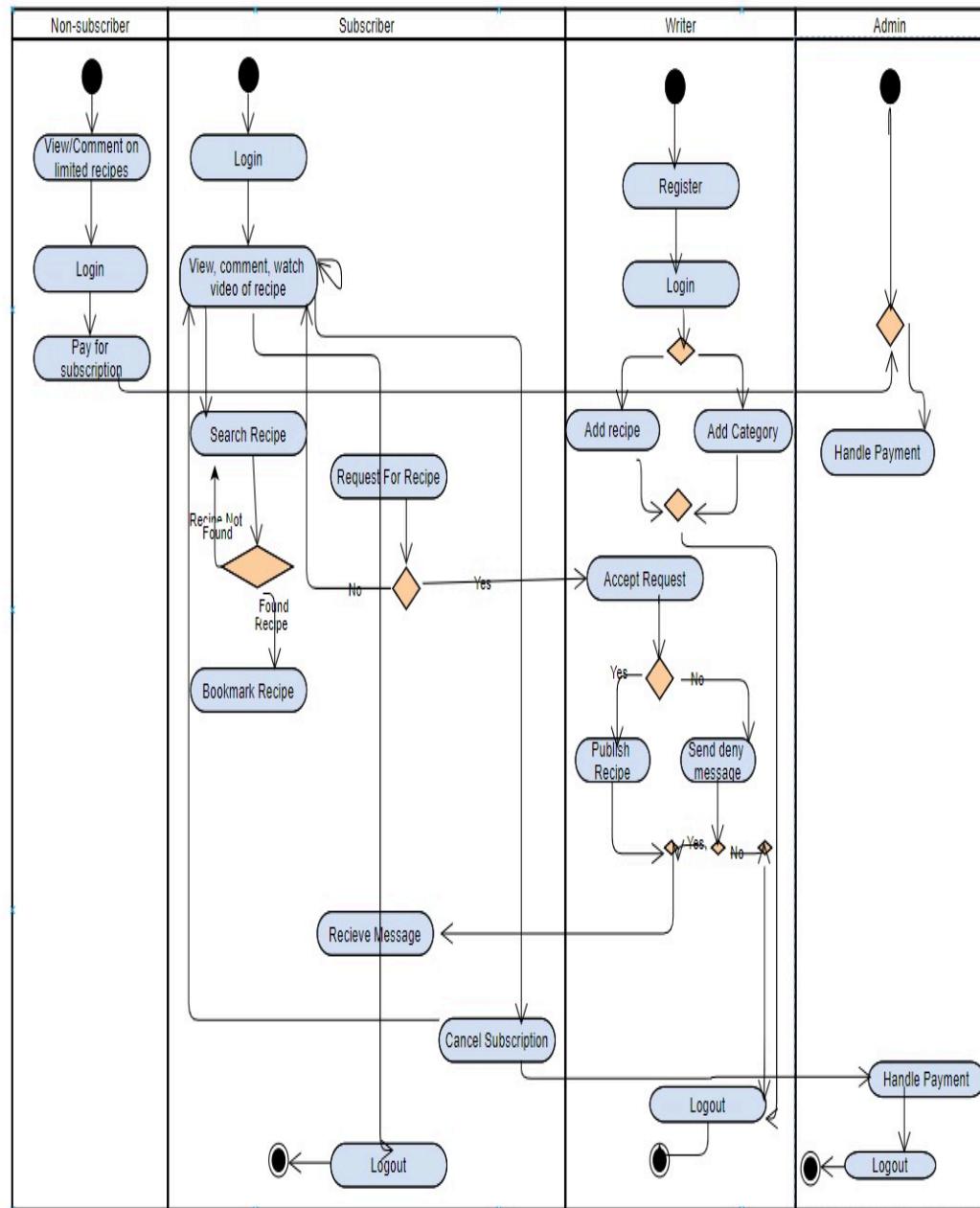
2. Class Diagram:



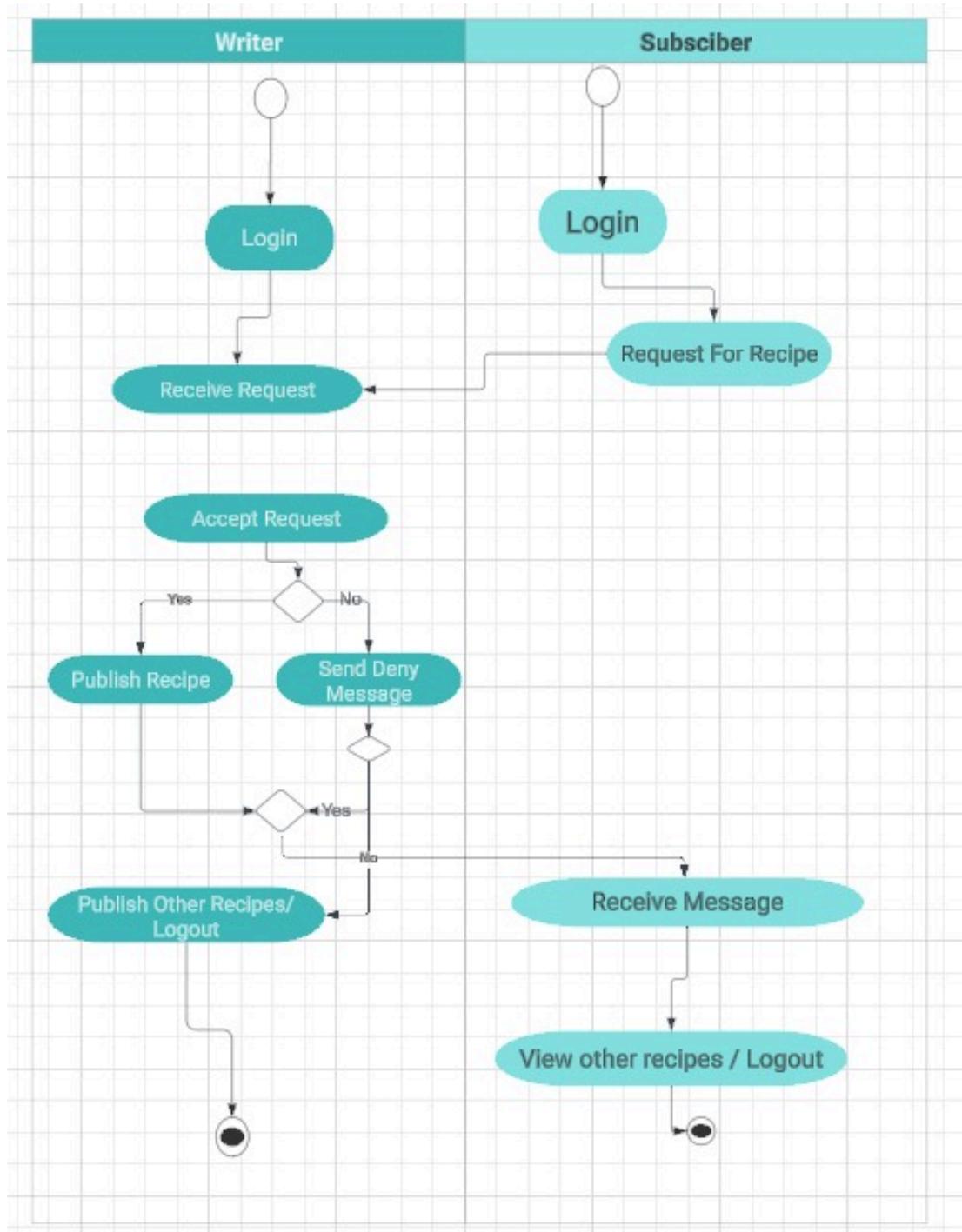
3. State Diagram:



4. Activity Diagram:



5. Activity diagram for request recipe:



- Architecture Patterns, Design Principles, and Design Patterns:

- 1) Architecture pattern:

- a) MVC Architecture:

- i) Model: The model represents the domain-specific data and business logic of the application. In our case, entities like User, Recipe, Indian, Continental, etc., along with their associated data and behavior, serve as the model components.
- ii) View: The view is responsible for presenting the data to the user and handling user interface interactions. In our architecture, the presentation layer mentioned earlier fulfills the role of the view. It encompasses the user interfaces designed to display information retrieved from the model and interact with users.
- iii) Controller: The controller acts as an intermediary between the model and the view, handling user input and updating the model accordingly. In our application, the service layer, including classes like UserController and WriterController, can be viewed as fulfilling the role of controllers. They receive requests from the presentation layer (view), process them by invoking appropriate methods on the model (data access layer), and update the view with the results.

- b) Layered Architecture:

In our project, we've structured the code following a Layered Architecture approach. This means we've divided the application into different layers, each with its own role, making it easier to manage and maintain.

- i) Entities Layer: Here, we define classes like User, Recipe, indian, and continental. These classes hold the data and behavior specific to each entity in our application.
- ii) Data Access Layer: This layer deals with interacting with the database. Classes like UserDao and WriterDao handle tasks such as saving, retrieving, updating, and deleting data from the database tables corresponding to our entity classes.

- iii) **Service Layer**: The service layer contains classes like UserService, WriterService, and UserProxy. These classes implement the business logic of our application. They perform operations related to users, recipes, and other entities, acting as an intermediary between the presentation layer and the data access layer.
- iv) **Presentation layer**: The presentation layer is an integral part of our application, responsible for managing user interfaces, interactions, and overall user experience.

2) Design Principles:

- a) **Dependency Injection:**
 - i) **Intent**: Dependency Injection (DI) is a design pattern in which the dependencies of a class are provided externally rather than created within the class itself. This allows for better decoupling, flexibility, and testability of the code.
 - ii) **Usage in the Code**: Throughout the codebase, dependencies such as UserDao and WriterDao are injected into the service classes (UserService, WriterService, UserProxy) via constructor injection. For example, the UserService class accepts a UserDao instance in its constructor, allowing it to interact with the data access layer without directly instantiating the Data Access Object (DAO). By injecting dependencies, the classes become more modular and easier to test, as mock or stub implementations of dependencies can be provided during testing.
- b) **Separation of Concerns:**
 - i) **Intent**: Separation of Concerns (SoC) is a design principle that advocates for breaking a system into distinct, independent modules, each responsible for a specific aspect of functionality. This enhances maintainability, reusability, and comprehensibility of the codebase.
 - ii) **Usage in the Code**: The codebase follows SoC by separating different concerns into separate classes or layers, such as entities (e.g., User, Recipe), data access objects (e.g., UserDao, WriterDao), and service classes (e.g., UserService, WriterService). Each class or layer has a clearly defined responsibility, such as data persistence, business logic, or presentation. For example, the UserService focuses on user-related operations, while the WriterService handles functionality related to recipe

management. By adhering to SoC, the codebase becomes more modular, easier to maintain, and less prone to unintended side effects or coupling between different components.

c) Single Responsibility Principle (SRP):

- i) Intent: The Single Responsibility Principle (SRP) states that a class should have only one reason to change, meaning it should have only one responsibility or job. This promotes high cohesion and reduces the risk of unintended side effects when modifying the code.
- ii) Usage in the Code: Each class in the codebase adheres to SRP by having a single responsibility or focusing on a specific aspect of functionality. For example, the UserService class is responsible for user-related operations such as login, registration, and profile management, while the WriterService class handles recipe-related operations. Additionally, the UserDao and WriterDao classes are responsible for data access and persistence operations related to their respective entities.

3) Design Patterns:

a) Facade

- i) Intent: The Facade pattern provides a unified interface to a set of interfaces in a subsystem. It simplifies the interactions between the client and the subsystem by providing a higher-level interface that hides the complexities of the subsystem's components.
- ii) Usage in the Code: The UserProxy class acts as a facade to the UserDao, providing a simplified interface for client code to interact with user-related functionalities. Client code interacts with methods like login, registerUser, and deleteUser through the UserProxy class without needing to directly access the UserDao. The UserProxy class encapsulates the logic for coordinating multiple operations within the UserDao, such as validating user input, logging method invocations, and applying security checks. This abstraction shields client code from the complexities of the UserDao, promoting simplicity and maintainability.

b) Singleton Pattern:

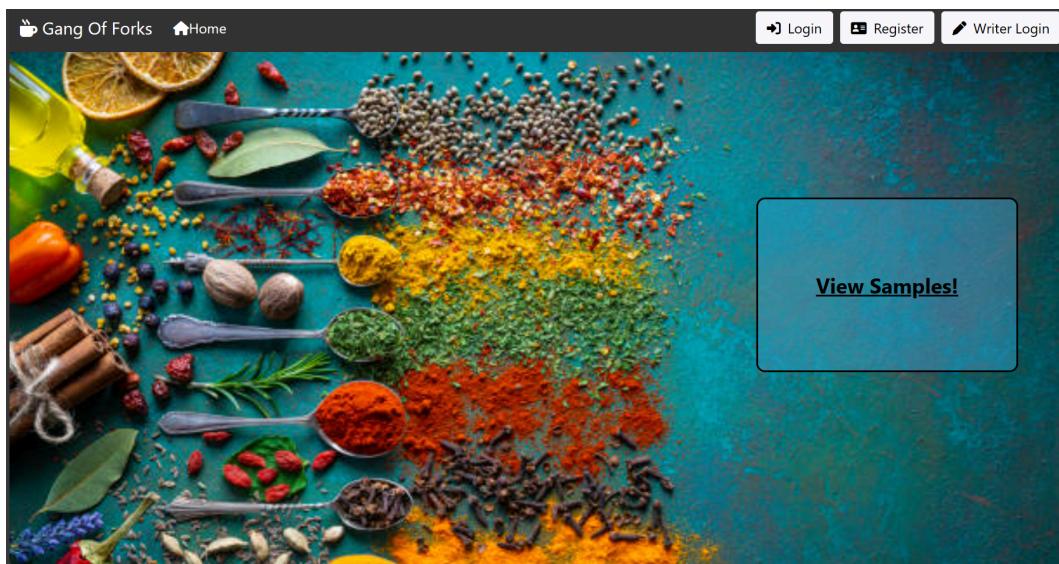
- i) Intent: The Singleton pattern ensures that a class has only one instance and provides a global point of access to that instance.
 - ii) Usage in the Code: The UserService class employs a form of the Singleton pattern by providing a static method getInstance() to obtain the single instance of UserService. This ensures that throughout the application, there's only one instance of UserService available, preventing multiple instances from being created inadvertently. By having a single instance of UserService, it facilitates centralised management of user-related operations and data, promoting consistency and avoiding duplication of resources.
- c) Factory Pattern:
- i) Intent: The Factory pattern provides an interface for creating objects without specifying the concrete classes of objects to be created. It delegates the responsibility of object instantiation to subclasses or a separate factory class.
 - ii) Usage in the Code: The DaoFactory is responsible for creating instances of data access objects (UserDao, WriterDao), abstracting away the concrete instantiation logic from the client code (UserService). It exposes methods like getUserDao() and getWriterDao() to create instances of the respective DAOs based on some configuration or criteria, providing a flexible way to obtain DAO instances without coupling the client code to specific implementations.
- Github link to the Codebase:
https://github.com/Shreelekha-142/Recipe_Book
 - Individual contributions of the team members.
 - Meghana: Design Pattern:
 Singleton Pattern
 - Design Principle:
 Separation of Concern
 - Jahnavi: Design Pattern:
 Singleton Pattern
 - Design Principle:
 Single Responsibility

Shreelekha: Design Pattern:
 Facade Pattern
Design Principle:
 Dependency injection

Mrunal: Design Pattern:
 Factory Pattern
 Design Principle:
 Dependency injection

- Screenshots with input values populated and output shown (Use white background screens).

Home:



Non subscriber (sample recipes):

Samples!

Recipe 1: Pasta Primavera

A light and flavorful pasta dish with seasonal vegetables.

Ingredients:

- 1 pound pasta (e.g., penne, farfalle)
- 1 tablespoon olive oil
- 1 onion, chopped
- 2 cloves garlic, minced
- 2 cups assorted vegetables (e.g., broccoli, peas, carrots), chopped
- 1 cup cherry tomatoes, halved
- 1/2 cup crumbled feta cheese (optional)
- 1/4 cup fresh basil leaves, chopped
- Salt and pepper to taste

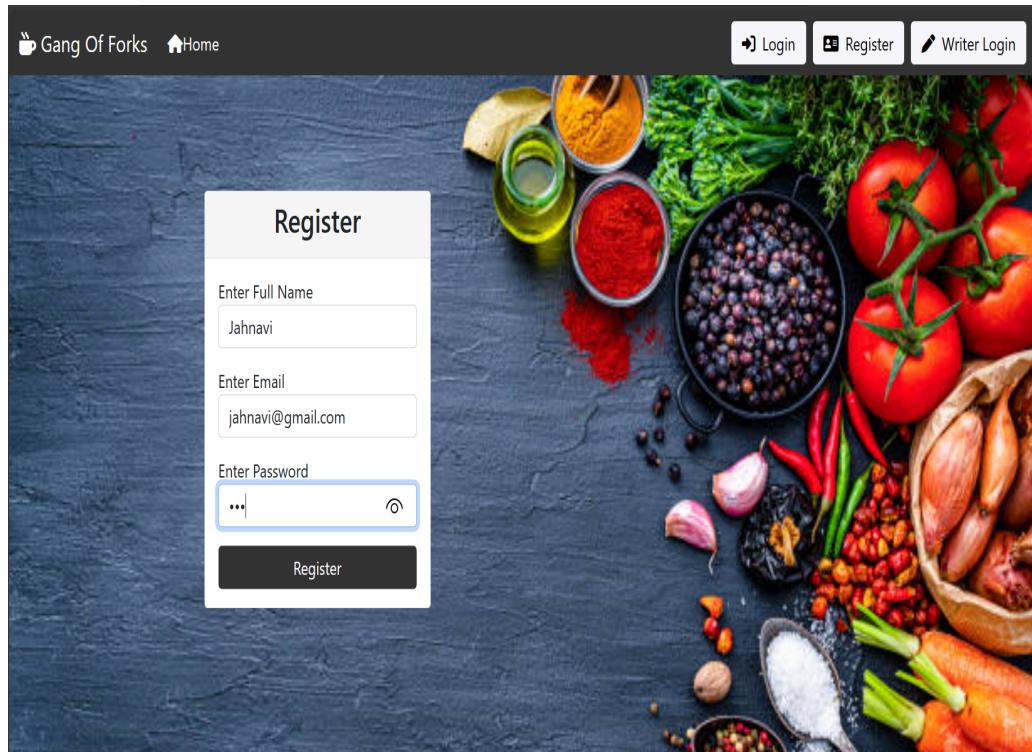
Recipe 5: Vegetable Stir-Fry

Quick and easy stir-fry loaded with colorful vegetables and tossed in a savory sauce, perfect for a healthy weeknight meal.

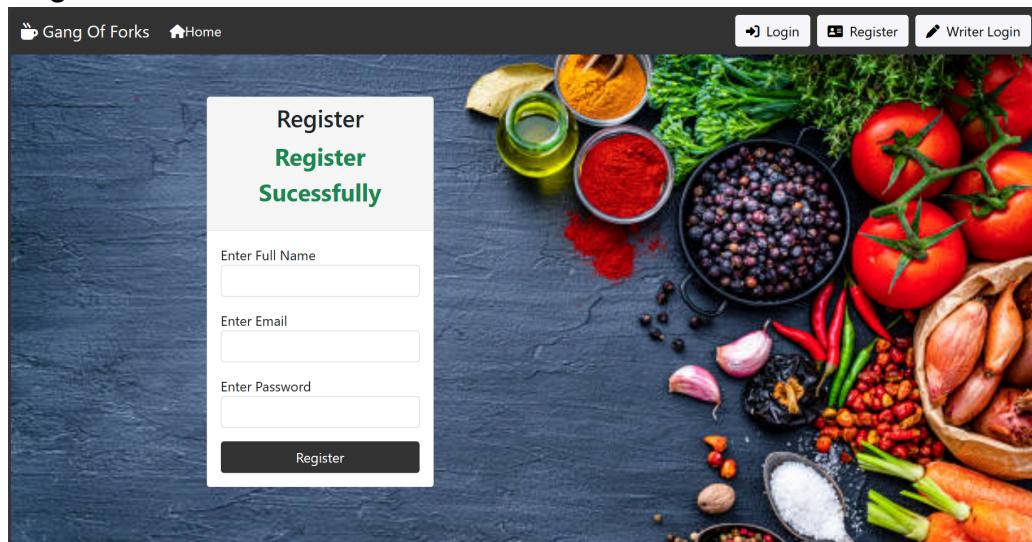
Ingredients:

- 2 tablespoons vegetable oil
- 1 bell pepper, thinly sliced
- 1 cup broccoli florets
- 1 cup sliced carrots
- 1 cup snow peas
- 1 cup sliced mushrooms
- 3 cloves garlic, minced
- 1 tablespoon grated ginger
- 1/4 cup soy sauce
- 2 tablespoons hoisin sauce
- 1 tablespoon rice vinegar
- 1 teaspoon sesame oil
- Optional garnish: chopped green onions, sesame seeds

Register page:

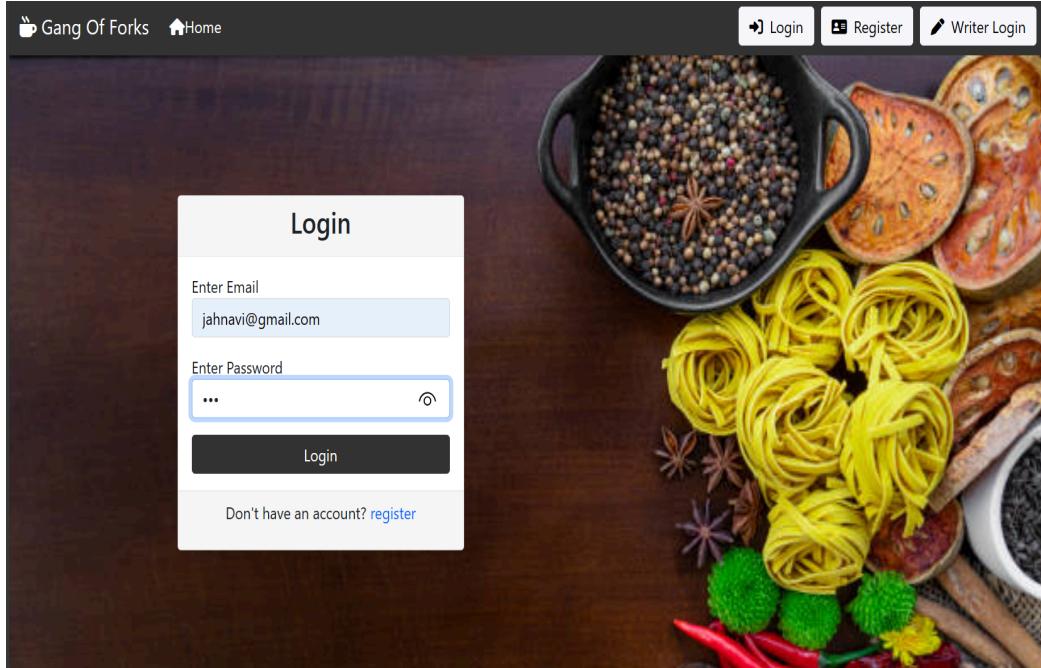


Registration successful:

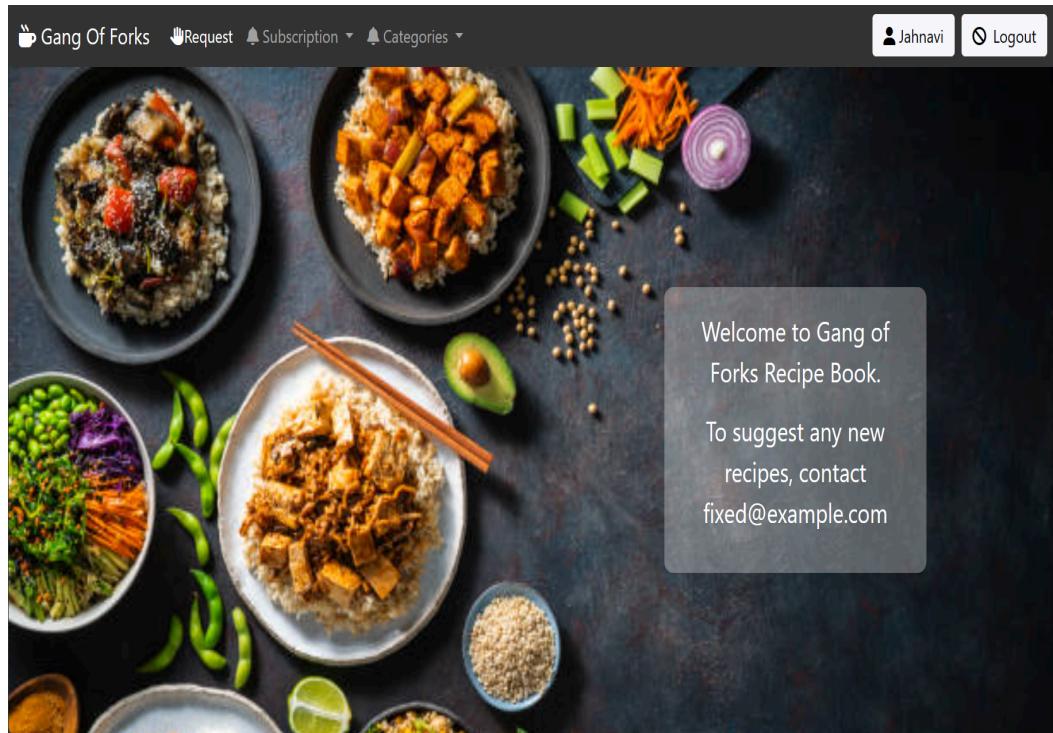


Subscriber:

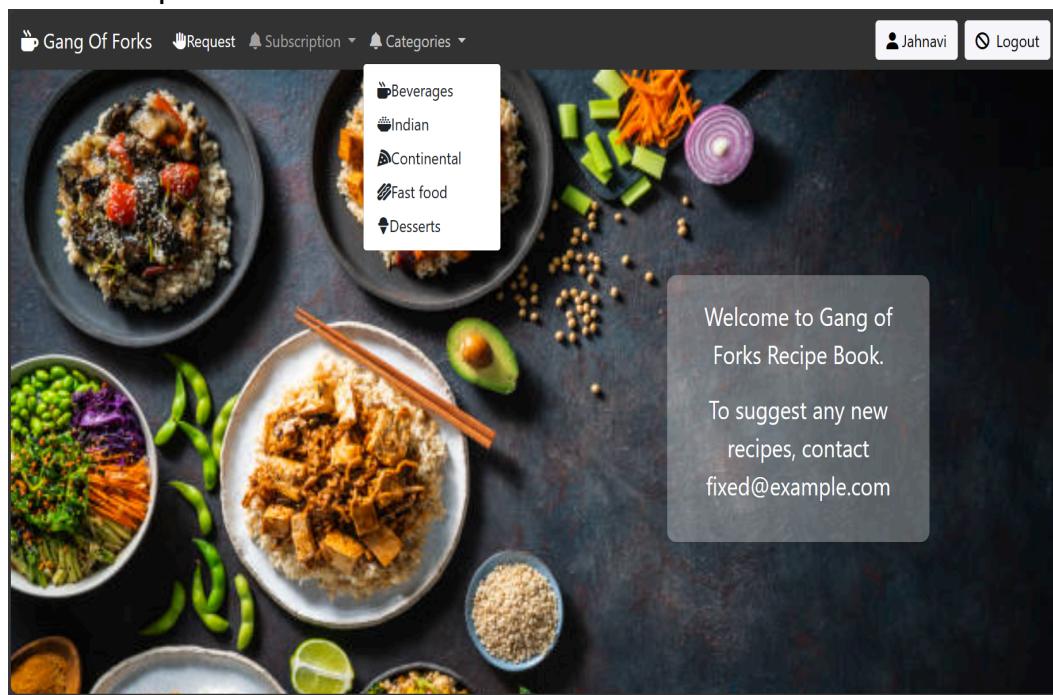
Login Page:



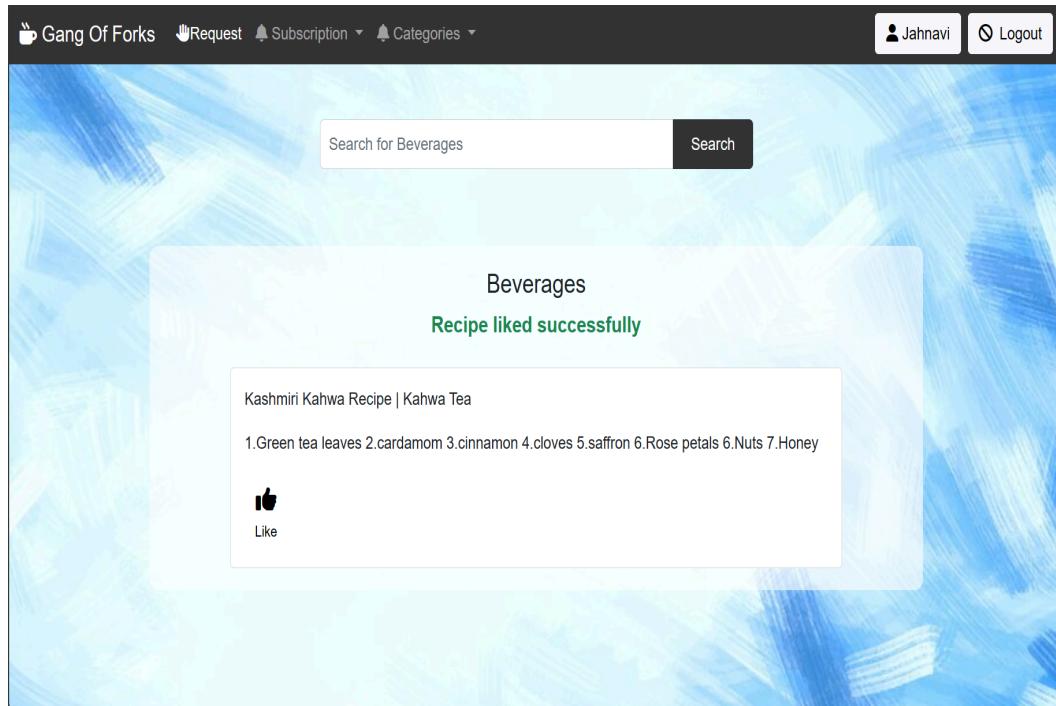
Login successful:



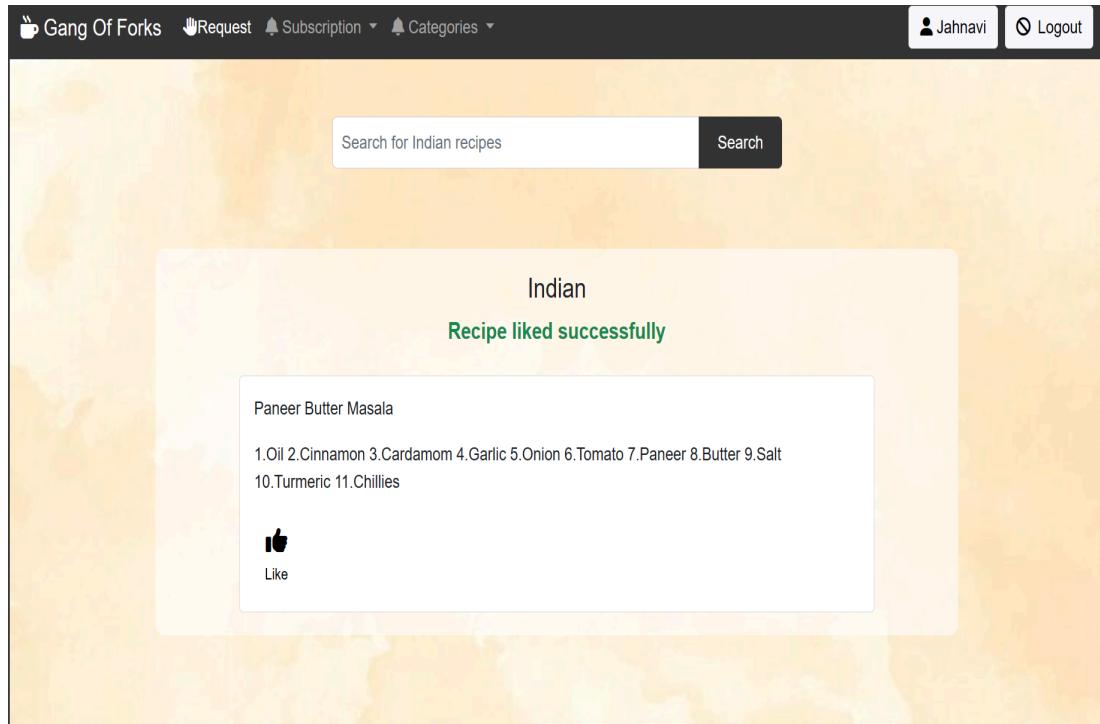
View Recipes:



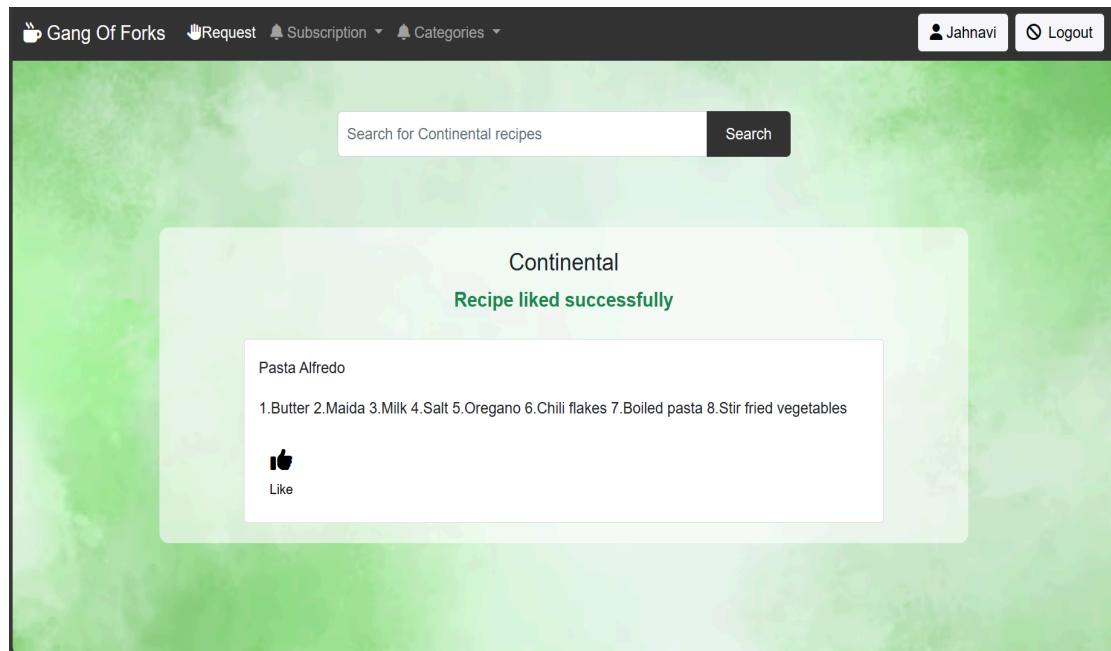
View and like Beverages:



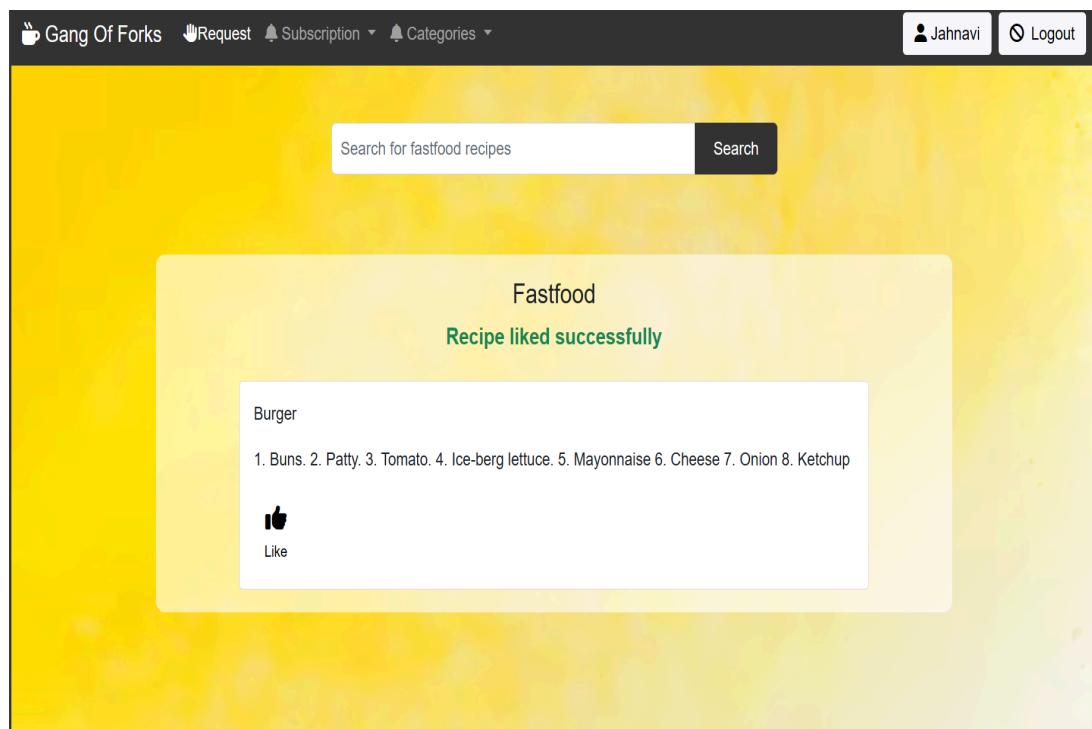
View and like Indian:



View and like continental:



View and like fast food:



View and like desserts:

The screenshot shows a user interface for a food app. At the top, there's a dark navigation bar with icons for 'Gang Of Forks', 'Request', 'Subscription', 'Categories', a user profile for 'Jahnavi', and a 'Logout' button. Below the bar is a search bar with the placeholder 'Search for desserts' and a 'Search' button. A large, semi-transparent overlay window is centered on the page, featuring a pink and yellow floral background. The overlay has a white header with the word 'Desserts' and a green success message 'Recipe liked successfully'. Inside the overlay, there's a section for a recipe titled 'Gajar Ka Halwa' with a list of ingredients: '1.Grated Carrots 2.Ghee 3.Milk 4.Sugar 5.Cardamom 6. Fried Cashews 7. Khoya'. At the bottom of this section is a 'Like' button with a thumbs-up icon.

Search for Beverages:

The screenshot shows a user interface for a food app, similar to the previous one but with a blue and white brushstroke background. At the top, there's a dark navigation bar with icons for 'Gang Of Forks', 'Request', 'Subscription', 'Categories', a user profile for 'Jahnavi', and a 'Logout' button. Below the bar is a search bar with the placeholder 'Search' and a 'Search' button. A large, semi-transparent overlay window is centered on the page, featuring a blue and white brushstroke background. The overlay has a white header with the word 'Beverages' and a green success message 'Kashmiri Kahwa Recipe | Kahwa Tea'. Inside the overlay, there's a section for a recipe titled 'Kashmiri Kahwa Recipe | Kahwa Tea' with a list of ingredients: '1.Green tea leaves 2.cardamom 3.cinnamon 4.cloves 5.saffron 6.Rose petals 6.Nuts 7.Honey'. At the bottom of this section is a 'Like' button with a thumbs-up icon.

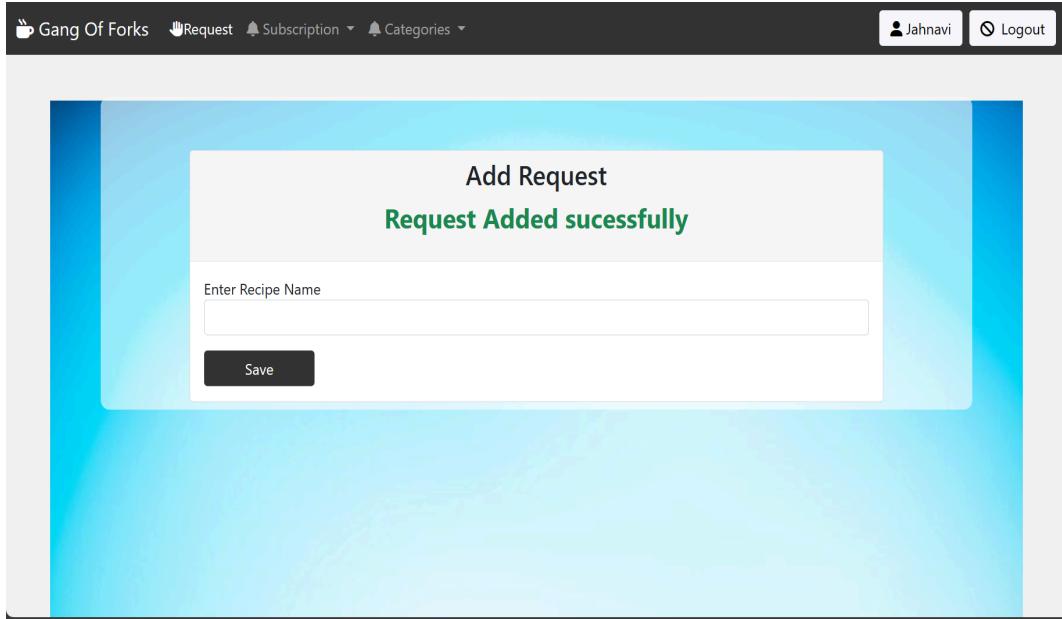
Search for fast food:

The screenshot shows a web interface for a food-related application. At the top, there is a navigation bar with icons for 'Gang Of Forks', 'Request', 'Subscription', and 'Categories'. On the right side of the bar are user profile information ('Jahnavi') and a 'Logout' button. Below the navigation bar, a search bar contains the text 'Burger'. To the right of the search bar is a 'Search' button. A large yellow rectangular area serves as a background for the results. Inside this area, a white card-like box is titled 'Fastfood'. It displays the search term 'Burger' and a numbered list of ingredients: '1. Buns. 2. Patty. 3. Tomato. 4. Ice-berg lettuce. 5. Mayonnaise 6. Cheese 7. Onion 8. Ketchup'. Below the list is a 'Like' button with a thumbs-up icon.

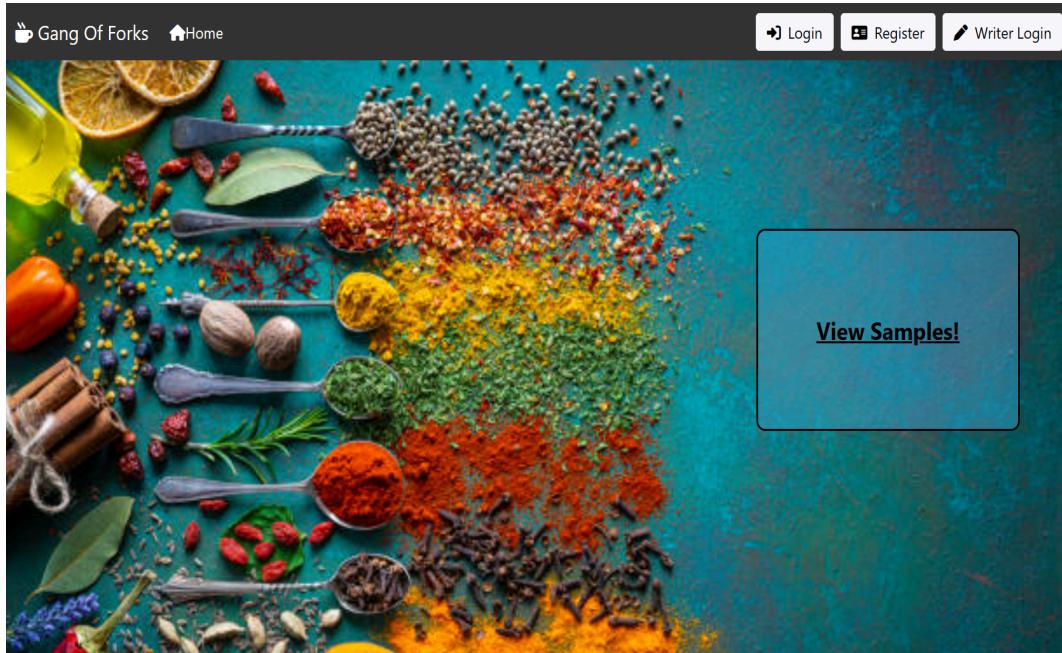
Request for Recipe:

The screenshot shows a web interface for adding a recipe request. At the top, there is a navigation bar with icons for 'Gang Of Forks', 'Request', 'Subscription', and 'Categories'. On the right side of the bar are user profile information ('Jahnavi') and a 'Logout' button. Below the navigation bar, a modal window titled 'Add Request' is open. Inside the modal, there is a text input field labeled 'Enter Recipe Name' containing the text 'puliyogare'. At the bottom of the modal is a dark blue 'Save' button.

Request Successful:

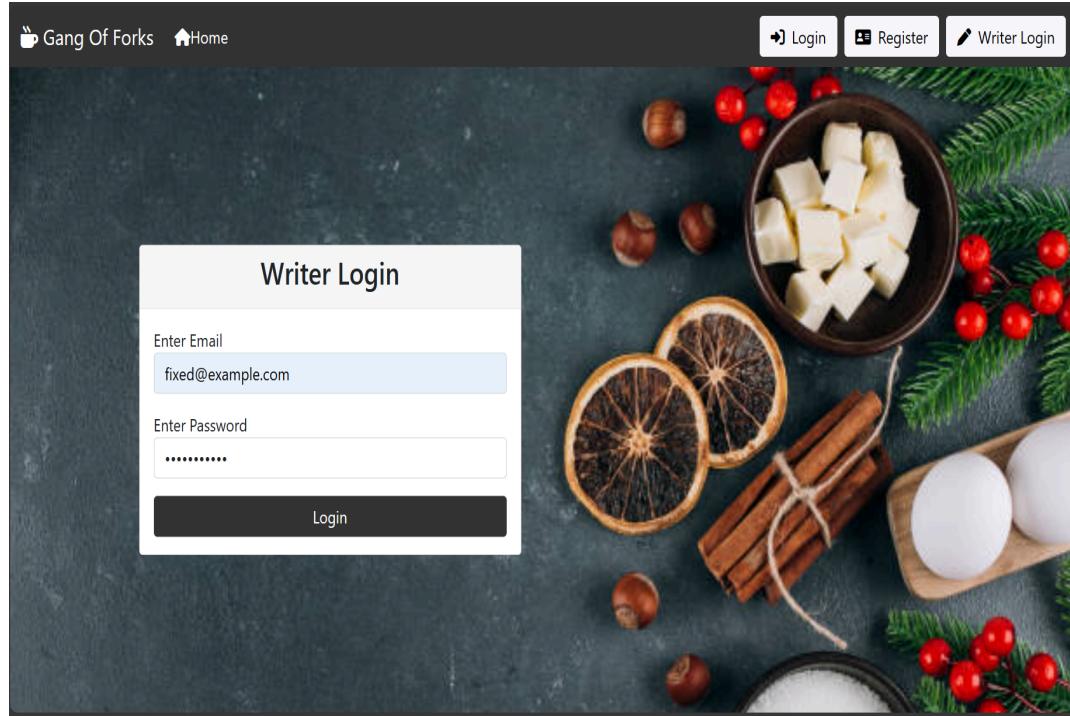


On logout:

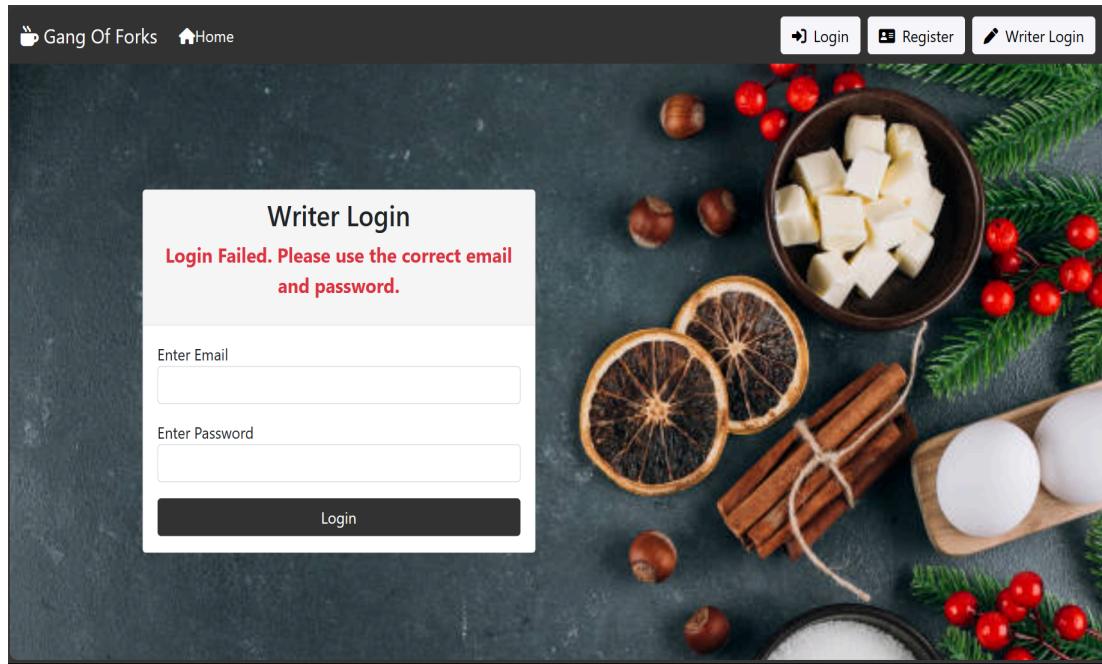


Writer:

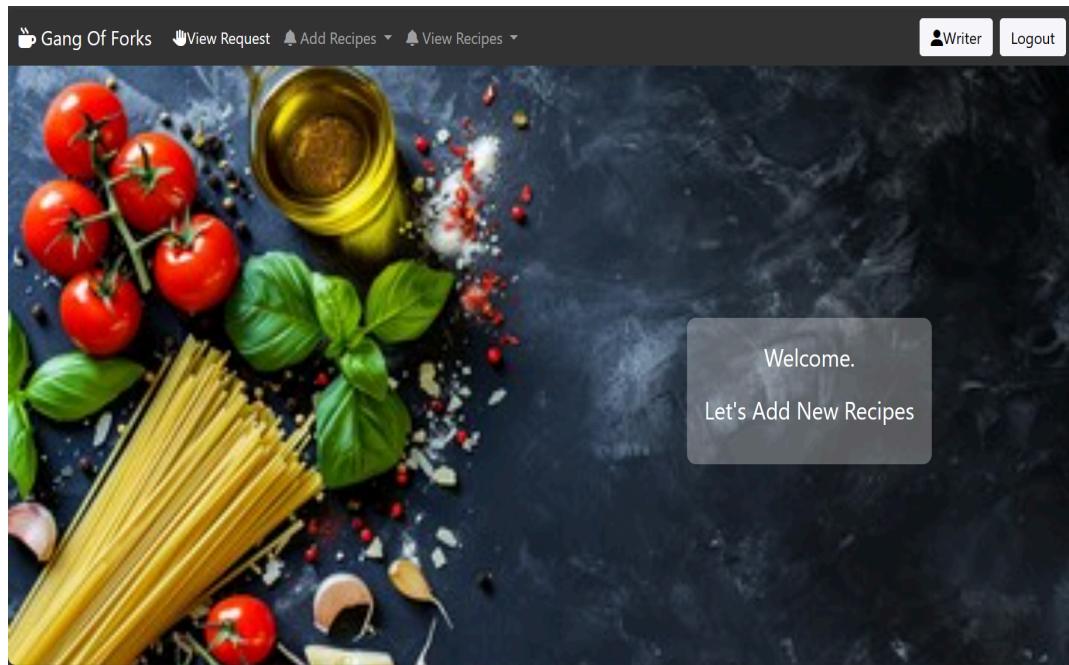
Login:



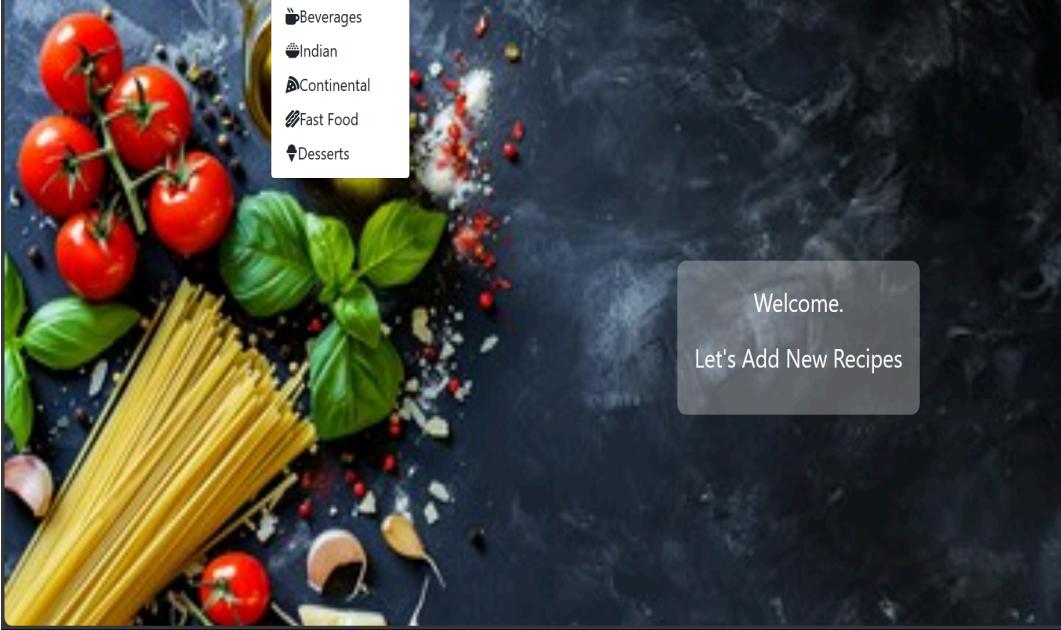
Login Unsuccessful:



Home:



Add Recipes:



Gang Of Forks View Request Add Recipes View Recipes

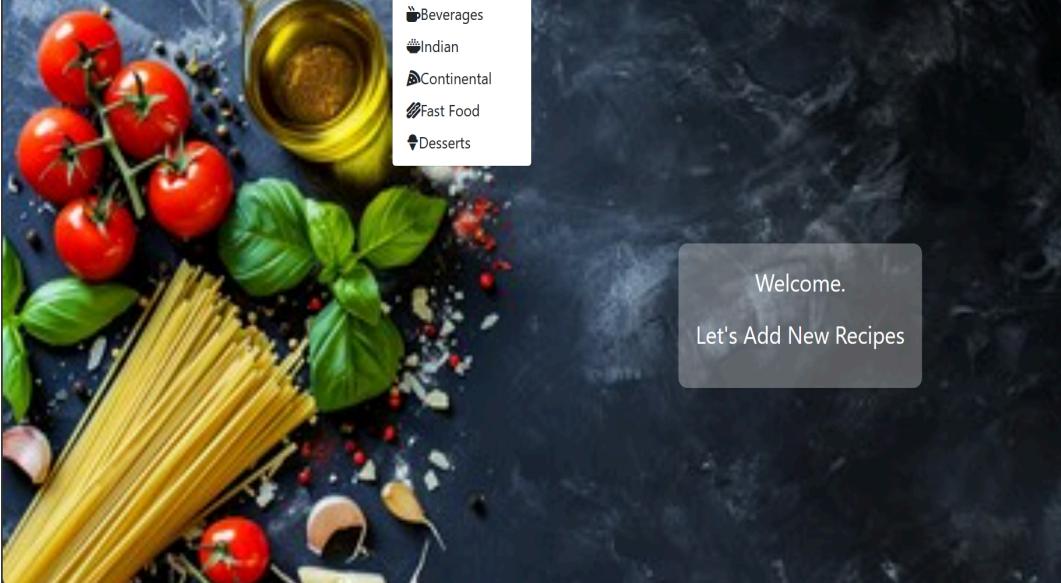
Welcome.

Let's Add New Recipes

- Beverages
- Indian
- Continental
- Fast Food
- Desserts

Writer Logout

View Recipes:



Gang Of Forks View Request Add Recipes View Recipes

Welcome.

Let's Add New Recipes

- Beverages
- Indian
- Continental
- Fast Food
- Desserts

Writer Logout

Beverages:

Add recipe:

Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Beverages

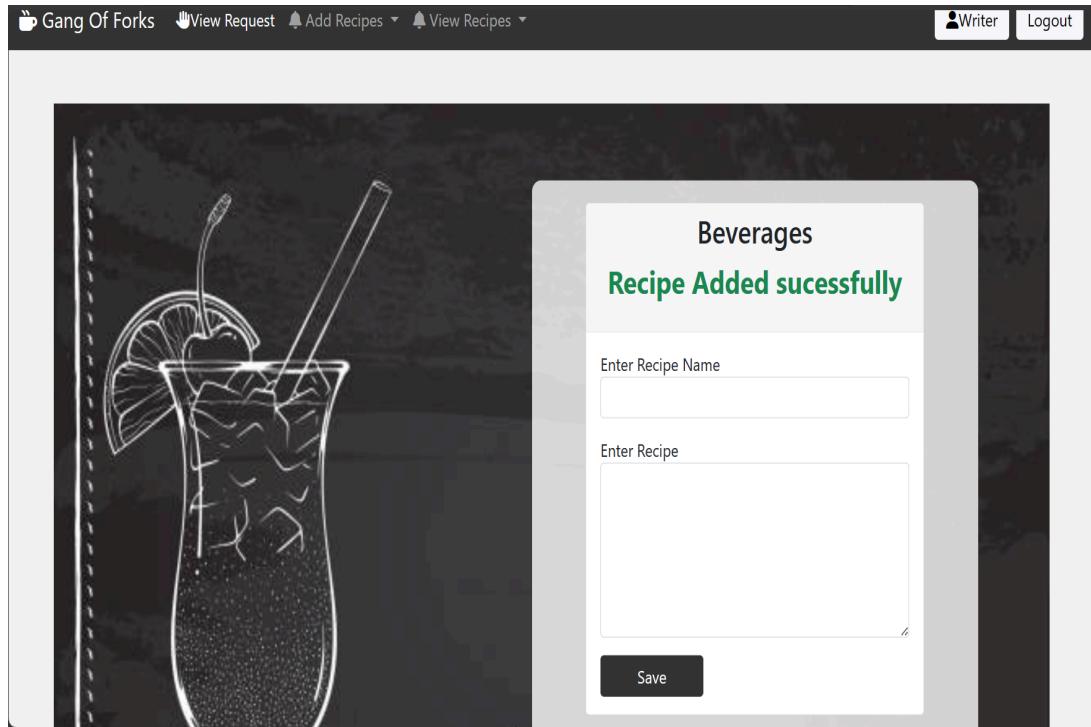
Enter Recipe Name
Kashmiri Kahwa Recipe | Kahwa Tea

Enter Recipe

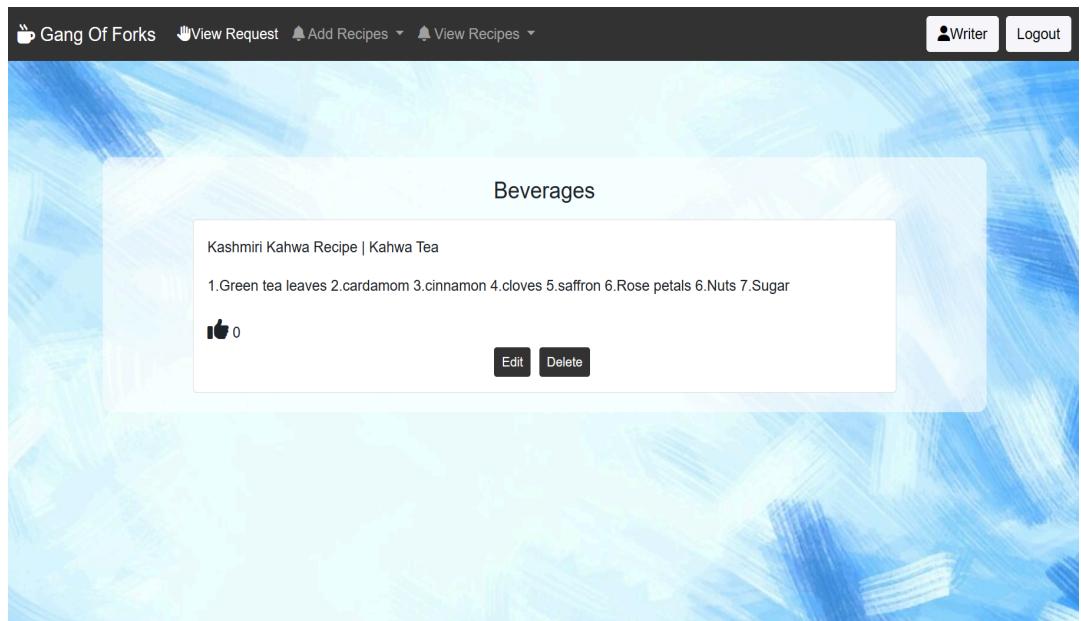
- Green tea leaves
- cardamom
- cinnamon
- cloves
- saffron
- Rose petals

Save

Added successfully:



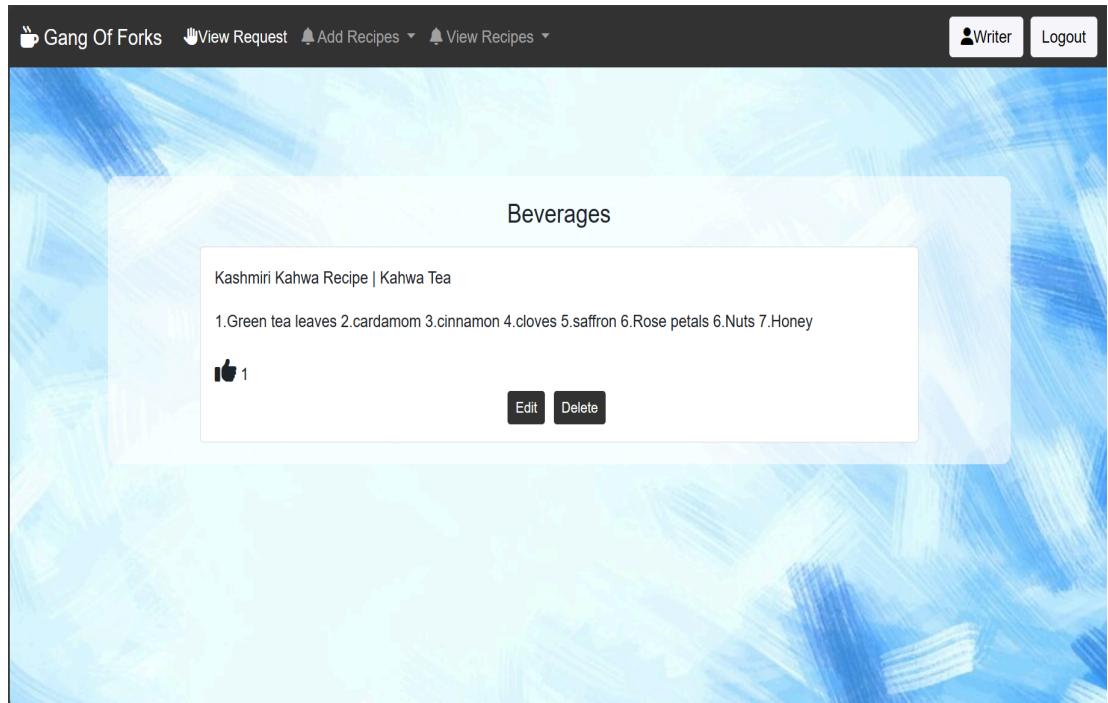
View:



Edit:

The screenshot shows a web application interface for editing a recipe. At the top, there's a navigation bar with icons for 'Gang Of Forks', 'View Request', 'Add Recipes', 'View Recipes', 'Writer', and 'Logout'. The main area is titled 'Edit your Recipes'. It has two input fields: 'Enter Title' containing 'Kashmiri Kahwa Recipe | Kahwa Tea' and 'Enter Description' containing a list of ingredients: '3.cinnamon', '4.cloves', '5.saffron', '6.Rose petals', '6.Nuts', and '7.Honey'. A blue box highlights the '7.Honey' entry. Below the description field is a dark blue 'Update' button.

Edit successful:



Indian food: Add recipe:

Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Indian

Enter Recipe Name
Paneer Butter Masala

Enter Recipe
Tomato
Paneer
Butter
Salt
Turmeric
Chili Powder

Save

Added successfully:

Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Indian

Recipe Added sucessfully

Enter Recipe Name

Enter Recipe

Save

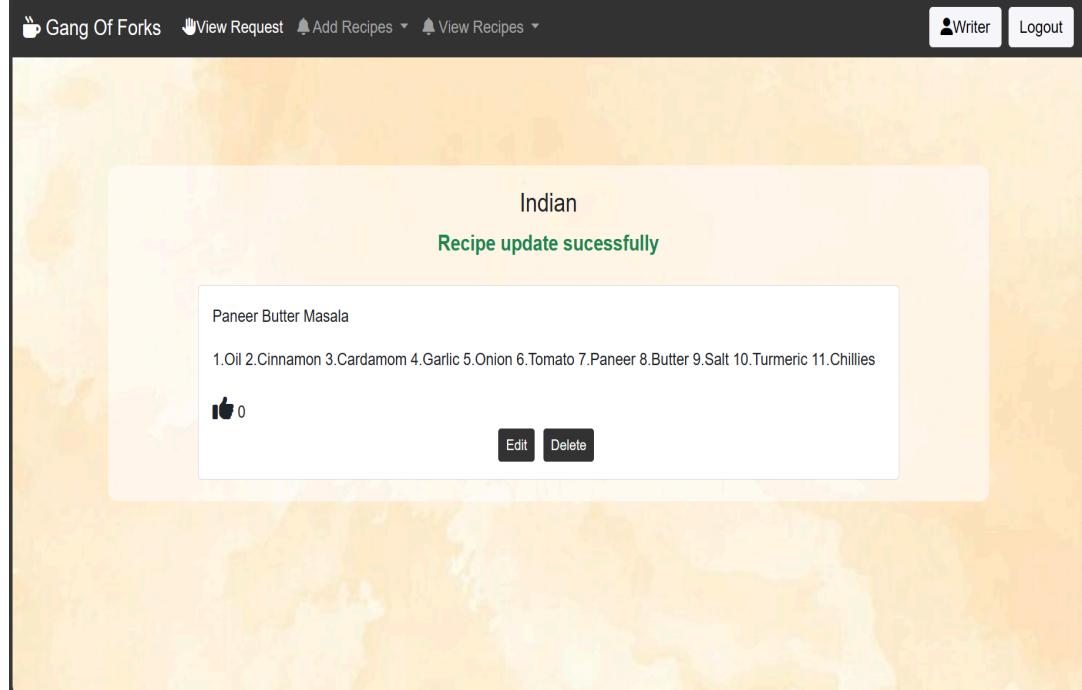
View:

The screenshot shows a recipe card on a website. At the top, there's a navigation bar with icons for a coffee cup, a fork, and a knife, followed by the text "Gang Of Forks". To the right of the navigation are links for "View Request", "Add Recipes", "View Recipes", a user icon labeled "Writer", and a "Logout" button. The main content area has a light orange background. A white rectangular card is centered, titled "Indian". Inside the card, the recipe name "Paneer Butter Masala" is displayed, along with a list of ingredients: "1.Oil 2.Cinnamon 3.Cardamom 4.Garlic 5.Onion 6.Tomato 7.Paneer 8.Butter 9.Salt 10.Turmeric 11.Chili powder". Below the ingredients is a "like" icon with the number "1". At the bottom of the card are two buttons: "Edit" and "Delete".

Edit:

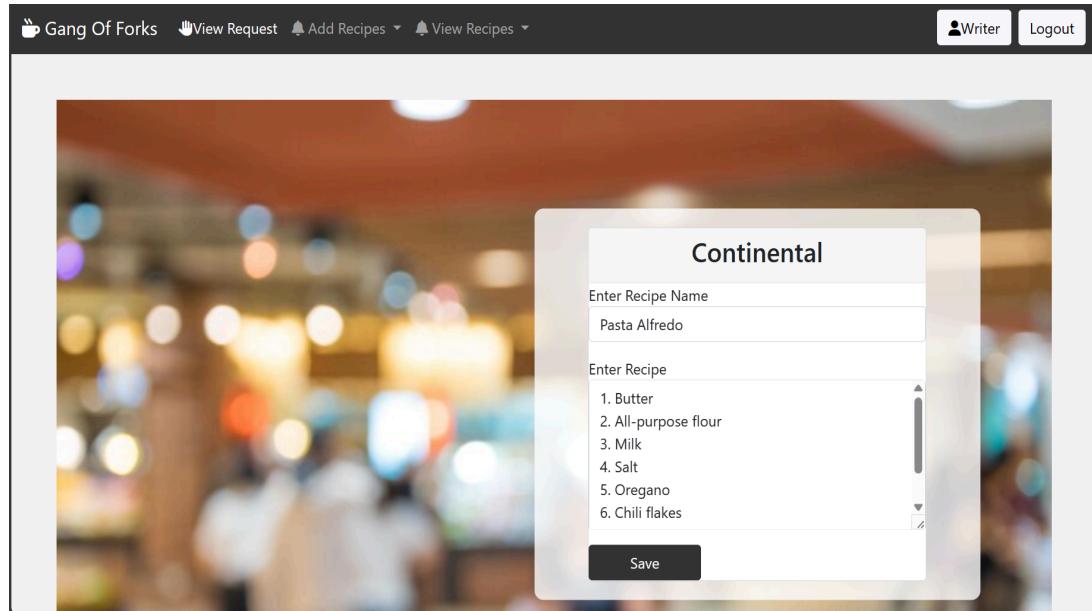
The screenshot shows an "Edit your Recipes" form. At the top, there's a header bar with the "Gang Of Forks" logo and navigation links. The main form area has a light gray background. It features a title field labeled "Enter Title" containing "Paneer Butter Masala". Below it is a large text area labeled "Enter Description" containing a list of ingredients: "6.Tomato", "7.Paneer", "8.Butter", "9.Salt", "10.Turmeric", and "11.Chillies". The "11.Chillies" entry is highlighted with a blue selection bar. At the bottom of the form is a single "Update" button.

Edit successful:

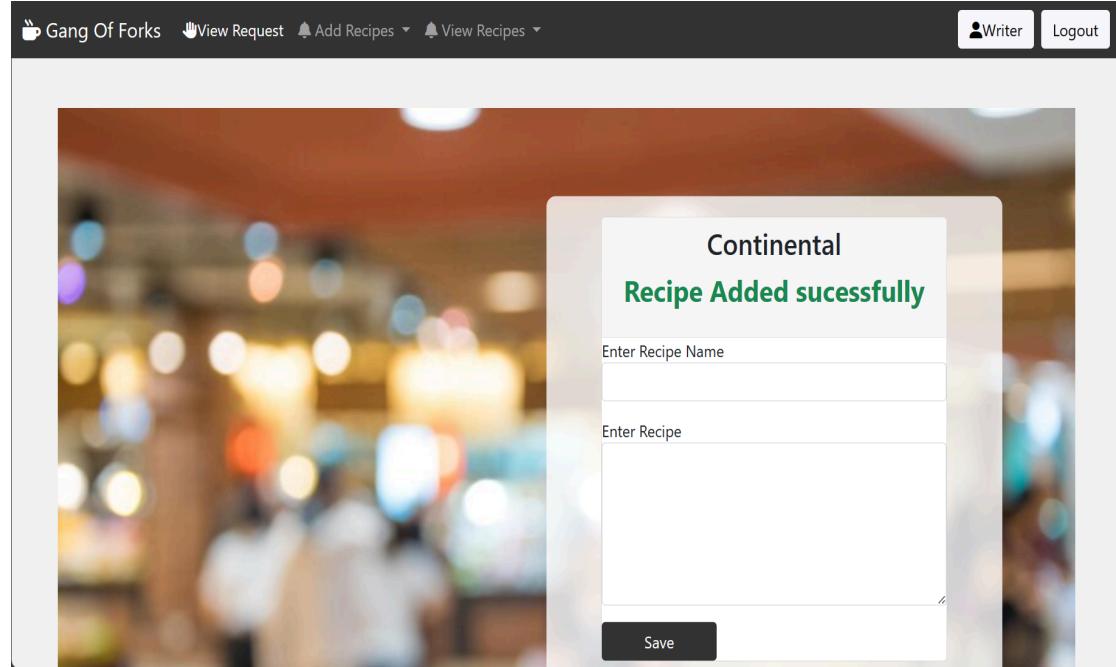


Continental food:

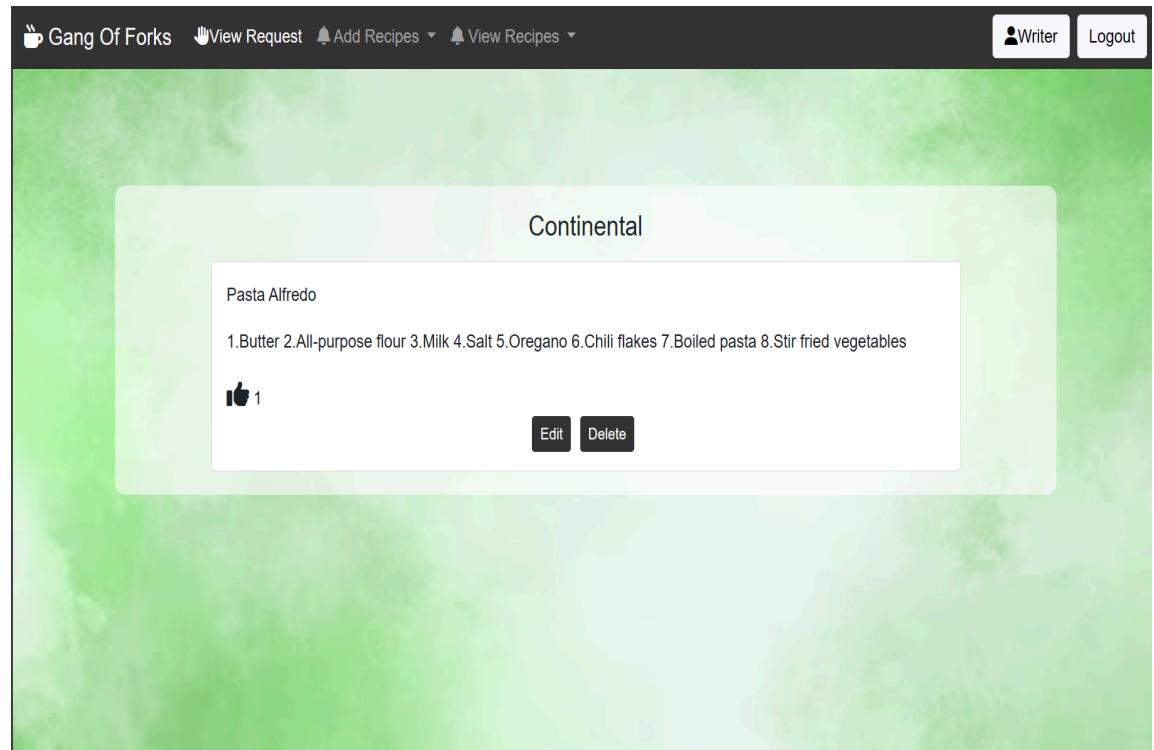
Add recipe:



Added successfully:



View:



Edit:

The screenshot shows a web application interface for editing a recipe. At the top, there is a dark header bar with the logo 'Gang Of Forks', navigation links for 'View Request', 'Add Recipes', 'View Recipes', and user account options ('Writer' and 'Logout'). Below the header is a light gray form area with a title 'Edit your Recipes'. The form contains fields for 'Enter Title' (containing 'Pasta Alfredo') and 'Enter Description' (containing a list of ingredients: 1. Butter, 2. Maida, 3. Milk, 4. Salt, 5. Oregano, 6. Chili flakes). A scroll bar is visible on the right side of the description input field. At the bottom of the form is a dark 'Update' button.

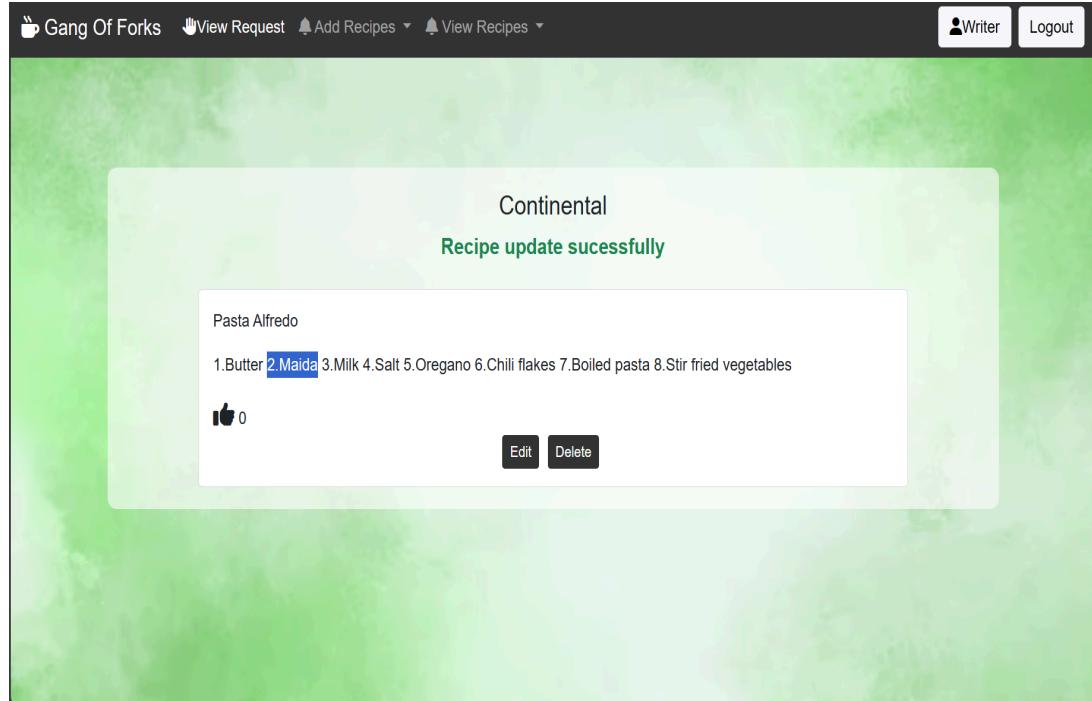
Enter Title
Pasta Alfredo

Enter Description

1. Butter
2. Maida
3. Milk
4. Salt
5. Oregano
6. Chili flakes

Update

Edit successful:



Fast food:

Add recipe:

The screenshot shows a user interface for adding a new recipe. On the left, there is a background image of a meal consisting of a burger, fries, and ketchup on a plate with cutlery. On the right, a form is displayed with the title 'Fast Food'. The first field, 'Enter Recipe Name', contains the value 'Burger'. The second field, 'Enter Recipe', contains a list of ingredients: '1. Buns.
2. Patty.
3. Tomato.
4. Lettuce.
5. Mayonnaise
6. Cheese'. A vertical scroll bar is visible on the right side of this input field. At the bottom of the form is a 'Save' button.

Added successfully:



Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Fast Food
Recipe Added sucessfully

Enter Recipe Name

Enter Recipe

Save

This screenshot shows a successful addition of a recipe. On the left, there's a photograph of a meal consisting of a burger, fries, and a side of ketchup. To the right, a modal window titled 'Fast Food' displays the message 'Recipe Added sucessfully'. It contains fields for 'Enter Recipe Name' and 'Enter Recipe', and a 'Save' button at the bottom.

View:

Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Fastfood

Burger

1. Buns. 2. Patty. 3. Tomato. 4. Lettuce. 5. Mayonnaise 6. Cheese 7. Onion 8. Ketchup

1

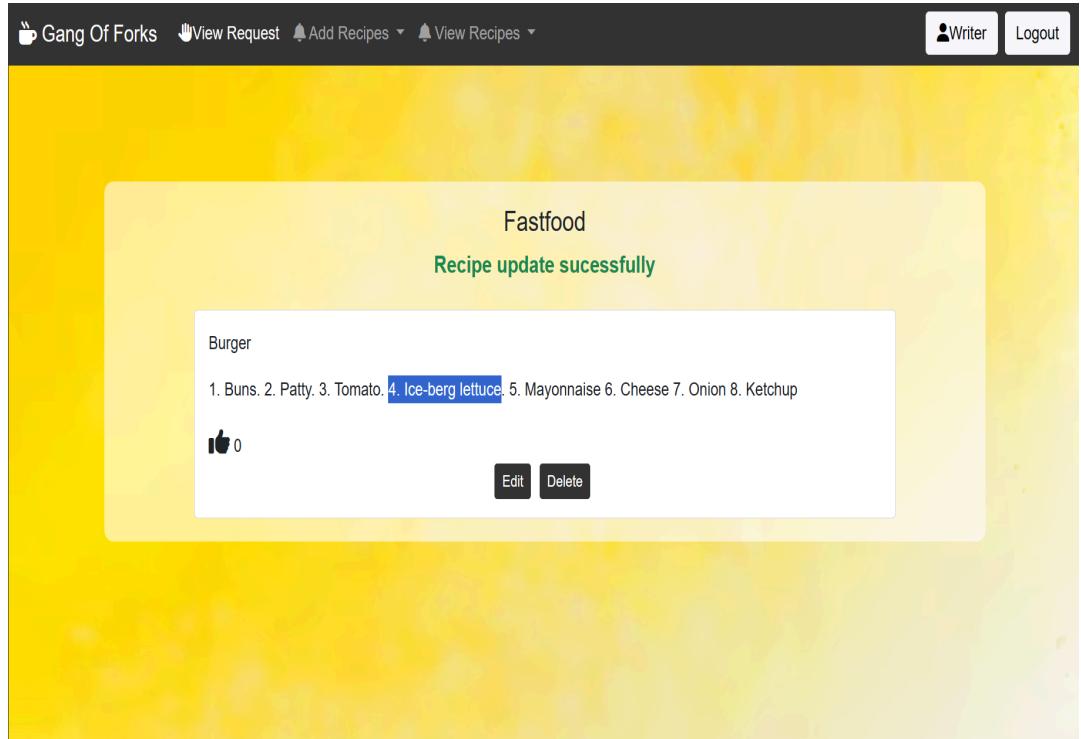
Edit Delete

This screenshot shows the view of a saved recipe. The title is 'Fastfood' and the specific item is 'Burger'. The ingredients listed are: 1. Buns. 2. Patty. 3. Tomato. 4. Lettuce. 5. Mayonnaise 6. Cheese 7. Onion 8. Ketchup. Below the list is a thumbs-up icon with the number '1'. At the bottom of the card are 'Edit' and 'Delete' buttons.

Edit:

The screenshot shows a web application interface for editing a recipe. At the top, there is a navigation bar with icons for 'Gang Of Forks', 'View Request', 'Add Recipes', 'View Recipes', 'Writer', and 'Logout'. The main area has a title 'Edit your Recipes'. Below it, there is a 'Enter Title' field containing 'Burger'. Under 'Enter Description', there is a list of ingredients: 1. Buns, 2. Patty, 3. Tomato, 4. Ice-berg lettuce, 5. Mayonnaise, 6. Cheese, and 7. Onion. The fourth item, '4. Ice-berg lettuce', is highlighted with a blue border. A large 'Update' button is at the bottom of the form.

Edit successful:



Desserts:
Add recipe:



Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Desserts

Enter Recipe Name
Gajar Ka Halwa

Enter Recipe

- 1.Grated Carrots
- 2.Ghee
- 3.Milk
- 4.Sugar
- 5.Cardamom
- 6. Fried Cashews

Save

Added successfully:



Gang Of Forks View Request Add Recipes View Recipes

Writer Logout

Desserts

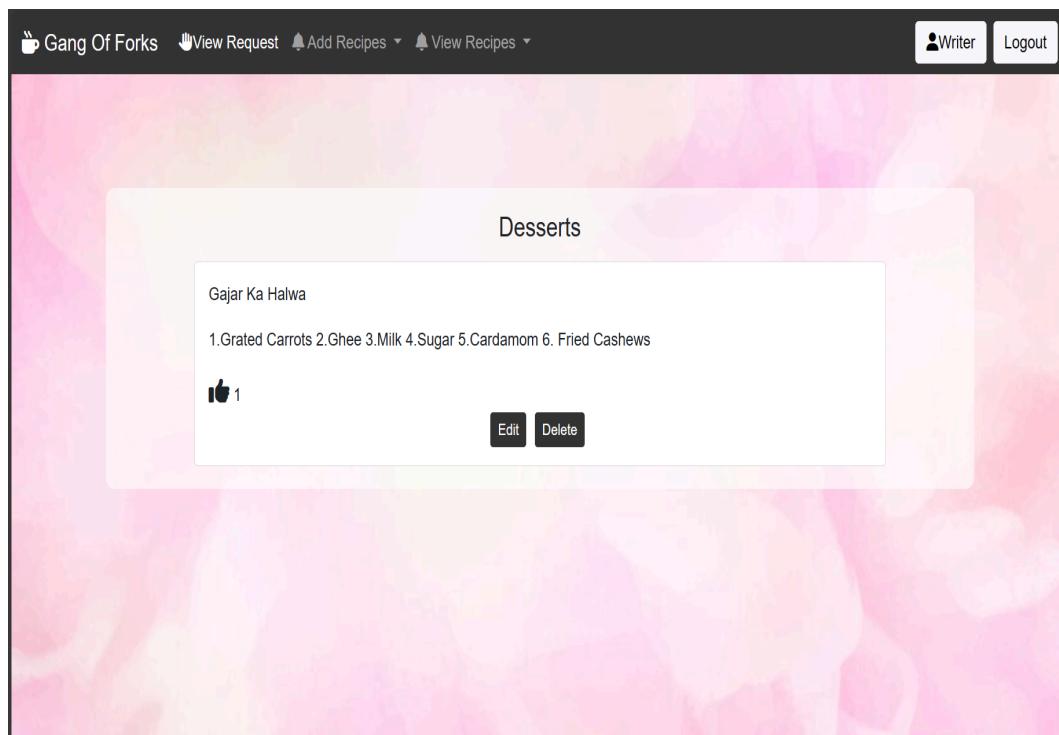
Recipe Added sucessfully

Enter Recipe Name

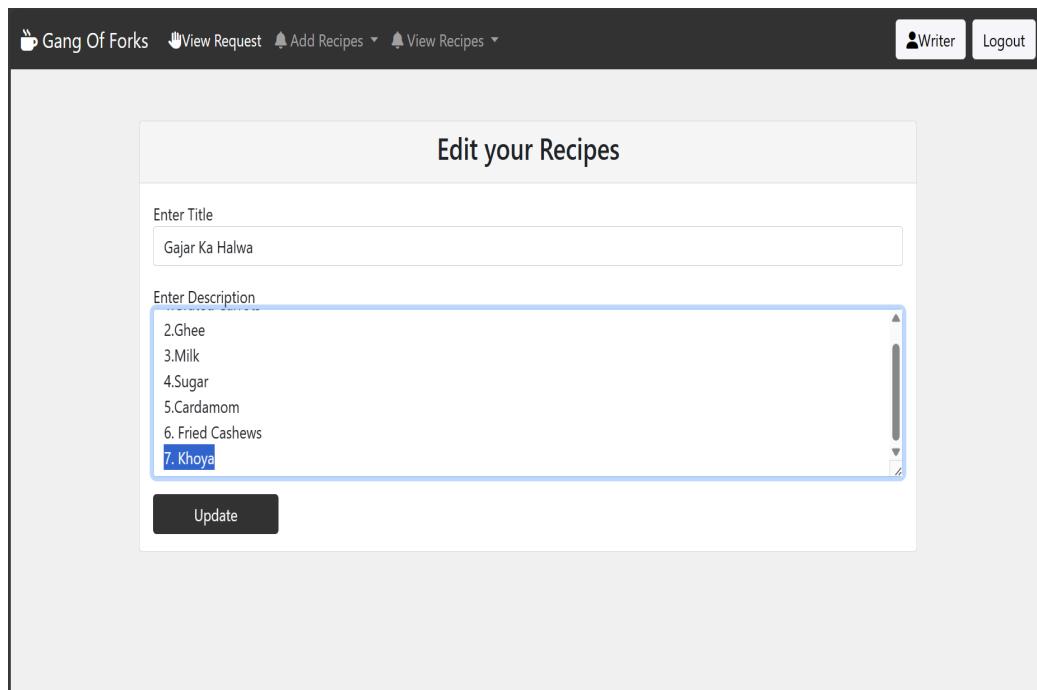
Enter Recipe

Save

View:



Edit:



Edit successful:

The screenshot shows a web application interface. At the top, there is a navigation bar with the logo "Gang Of Forks", menu items "View Request", "Add Recipes", "View Recipes", and user options "Writer" and "Logout". The main content area has a pink gradient background. A central box displays the title "Desserts" and a success message "Recipe update sucessfully". Inside this box, there is a card for "Gajar Ka Halwa" with the following details:

- Ingredients: 1.Grated Carrots 2.Ghee 3.Milk 4.Sugar 5.Cardamom 6. Fried Cashews 7. Khoya
- Thumbs up icon with a count of 0
- Buttons for "Edit" and "Delete"

View Request:

The screenshot shows a web application interface. At the top, there is a navigation bar with the logo "Gang Of Forks", menu items "View Request", "Add Recipes", "View Recipes", and user options "Writer" and "Logout". The main content area has a blue gradient background. A central box displays the title "View Requests" and a list of two items:

- "lemon rice" with a "Delete" button
- "puliyogare" with a "Delete" button

Delete Request:

The screenshot shows a web application interface. At the top, there is a dark header bar with the following items from left to right: a coffee cup icon followed by the text "Gang Of Forks", a "View Request" button, a "Add Recipes" dropdown menu, a "View Recipes" dropdown menu, a user profile icon labeled "Writer", and a "Logout" button.

The main content area has a blue gradient background. In the center, there is a white rectangular box containing the following text:

View Requests
Request Deleted Sucessfully

Below this message, there is a smaller white box with the text "puliyogare" and a "Delete" button at the bottom right.