

**Dharmsinh Desai University, Nadiad**

Faculty of Technology

Department of Computer Engineering

B.Tech CE Semester -VI

*Subject:* Service Oriented Computing

*Project Title:* **Online Cooking Recipe Management System**

*By:*

Jinal Panchal (CE085- 18CEUBG005)

Hena Patel (CE096- 18CEUOS114)

Nirali Shah (CE122- 18CEUOS059)

*Guided By:*

Prof. Apurva A. Mehta

Prof. Prashant M. Jadav

**Table Of Contents**

Contents

[**1. Abstract 3**](#_Toc68326292)

[1.1 Purpose 3](#_Toc68326293)

[1.2 Scope 3](#_Toc68326294)

[1.3 Overview 3](#_Toc68326295)

[**2. Introduction 4**](#_Toc68326296)

[2.1 Brief Introduction (about the system) 4](#_Toc68326297)

[2.2 Technology/ Platform/ Tools Used 4](#_Toc68326298)

[**3. Software Requirements Specification for Online Cooking Recipe Management Portal 6**](#_Toc68326299)

[3.1 System User 6](#_Toc68326300)

[3.2 System features 6](#_Toc68326301)

[3.2.1 Functional Requirements 6](#_Toc68326302)

[**4. Design 9**](#_Toc68326303)

[4.1 Class Diagram 9](#_Toc68326304)

[4.2 Use Case Diagram 10](#_Toc68326305)

[4.3 E-R Diagram 11](#_Toc68326306)

[4.4 Data Dictionary 11](#_Toc68326307)

[**5. Implementation Details 13**](#_Toc68326308)

[5.1 Modules Created (Brief Description) 13](#_Toc68326309)

[5.2 Function Prototype 14](#_Toc68326310)

[**6. Testing 16**](#_Toc68326311)

[**7. Screen-shots 18**](#_Toc68326312)

[**8. Conclusions 25**](#_Toc68326313)

[**9. Limitations and Future Extensions 25**](#_Toc68326314)

[9.1 Functionalities Not Implemented 25](#_Toc68326315)

[9.2 Future Extensions 25](#_Toc68326316)

[**10. Bibliography 26**](#_Toc68326317)

# 1. Abstract

Online cooking recipe management portal enables end user to upload their recipes and learn new cooking recipes online. Using this system, users can see various types of recipes and learn their procedure. Users can also provide their feedback by means of comments and likes or dislikes.

## 1.1 Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define

the system, its requirement with respect to end users. Also, we shall predict how this system will be used in order to gain a better understanding of the project.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project’s target audience, its user interface and software requirements. It defines how our clients, team and end user see the system and its functionality.

## 1.2 Scope

The main objective of the system is that users can share their recipe ideas and learn new recipes from any place. Here, users can upload their recipes & see various kind of recipes of other users. User can get feedback about their recipes by comments and likes/dislikes provided by other users. Thus, users can manage their recipes online.

## 1.3 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this system, the functional and data requirements of the system.

# 2. Introduction

## 2.1 Brief Introduction (about the system)

Online cooking recipe management system is a web application. It allows view different kind of recipes. Anyone can start posting new recipes on the website, after doing registration successfully. Using this system, anyone on the internet can learn new recipes online. Users can see the different kind of recipes uploaded by other users by just one click & they can write some comment on that recipe. Users can also like or dislike any recipe if they wish. This system can be beneficial for a restaurant and cooking instructor to get new recipe ideas. This system makes sharing of the new recipes very easy by providing common interface for users.

## 2.2 Technology/ Platform/ Tools Used

**For WCF Service Project**

Technologies:-

* HTML5
* CSS3
* Bootstrap4
* Windows Communication Foundation Services
* ASP.NET framework
* ADO.Net
* SQL server database

Tools:-

* Visual Studio

**For Rest API Project**

Technologies:-

* HTML5
* CSS3
* Bootstrap4
* Rest API
* ASP.NET framework 4.6.1 (Web client)
* ASP.NET core 2.1 (REST API)
* SQL server database

Tools:-

* Visual Studio
* Restman/ Postman/ Fiddler

# 3. Software Requirements Specification for Online Cooking Recipe Management Portal

## 3.1 System User

* End User

## 3.2 System features

### 3.2.1 Functional Requirements

**R.1 Manage account**

User can register himself by providing necessary details and can login to the system by providing correct credentials. User can view his/her profile and update profile details anytime.

R.1.1 Register new user

Input: Enter name, email & password of user

Output: Confirmation message for successful registration, otherwise error message

R.1.2 Login to user account

Input: Email & password of user

Output: Navigate to home page if login succeeds, otherwise display error message

Processing: User credentials are validated

R.1.3 View user profile

Input: User selection to view account information

Output: Display user account details

R.1.4 Update user profile

Description: User can update his/her personal details by clicking on update profile.

R.1.4.1 Display current details of user

Input: User selection to view account information

Output: Current details of user account

R.1.4.2 Update user details

Input: Details to be updated

Output: Confirmation message for updated user details

R.1.5 Logout

Input: User selection

Output: User is redirected to login page

**R.2 Manage Recipes**

User can upload new recipe & view recipes uploaded by other users. User need to login before uploading any recipe. User can also view, update & delete his/her uploaded recipes. User can also like or dislike particular recipe uploaded by other users

R.2.1 Add new recipe

Input: Enter necessary details like title, ingredients, method, category, otherdetails & photo of recipe to be added

Output: Confirmation message, if recipe is added successfully, otherwise display error message

R.2.2 Update recipe

R.2.2.1 View recipe details

Input: User selection to view particular recipe

Output: Current details of the recipe

R.2.2.2 Update recipe details

Input: Recipe details to be updated

Output: Confirmation message regarding recipe update

R.2.3 Delete recipe

Input: User selection of particular recipe

Output: Confirmation message if delete is successful, otherwise error message

R.2.4 View recipes

Description: User can view kind of recipes. User can view all recipes including his/her own recipes & other users recipes. Additionally, user can also view particular recipe details.

Input: User selection

Output: Recipes are displayed as per user selection

R.2.5 Like or dislike recipe

Input: User selection for particular recipe

Output: Like & dislike count of selected recipe is updated

**R.3 Manage Comments**

User can add comment to particular recipe. User can also see comments added to all the recipes.

R.3.1 Add new comment

Input: Comment text about particular recipe

Output: Comment is displayed, if added successfully, otherwise error message is displayed

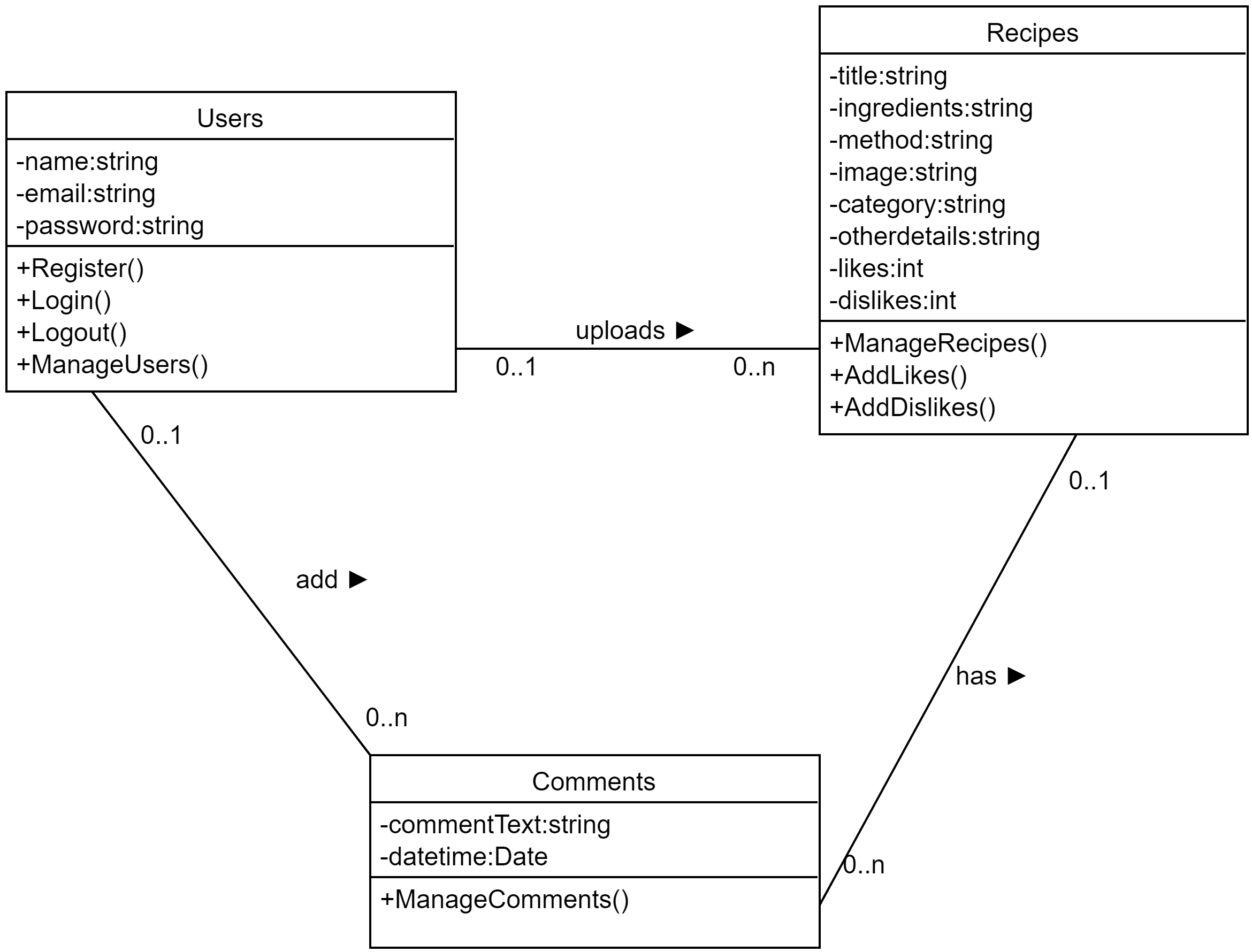
R.3.2 View comments

Input: User selection for particular recipe

Output: Comments of that recipe are displayed

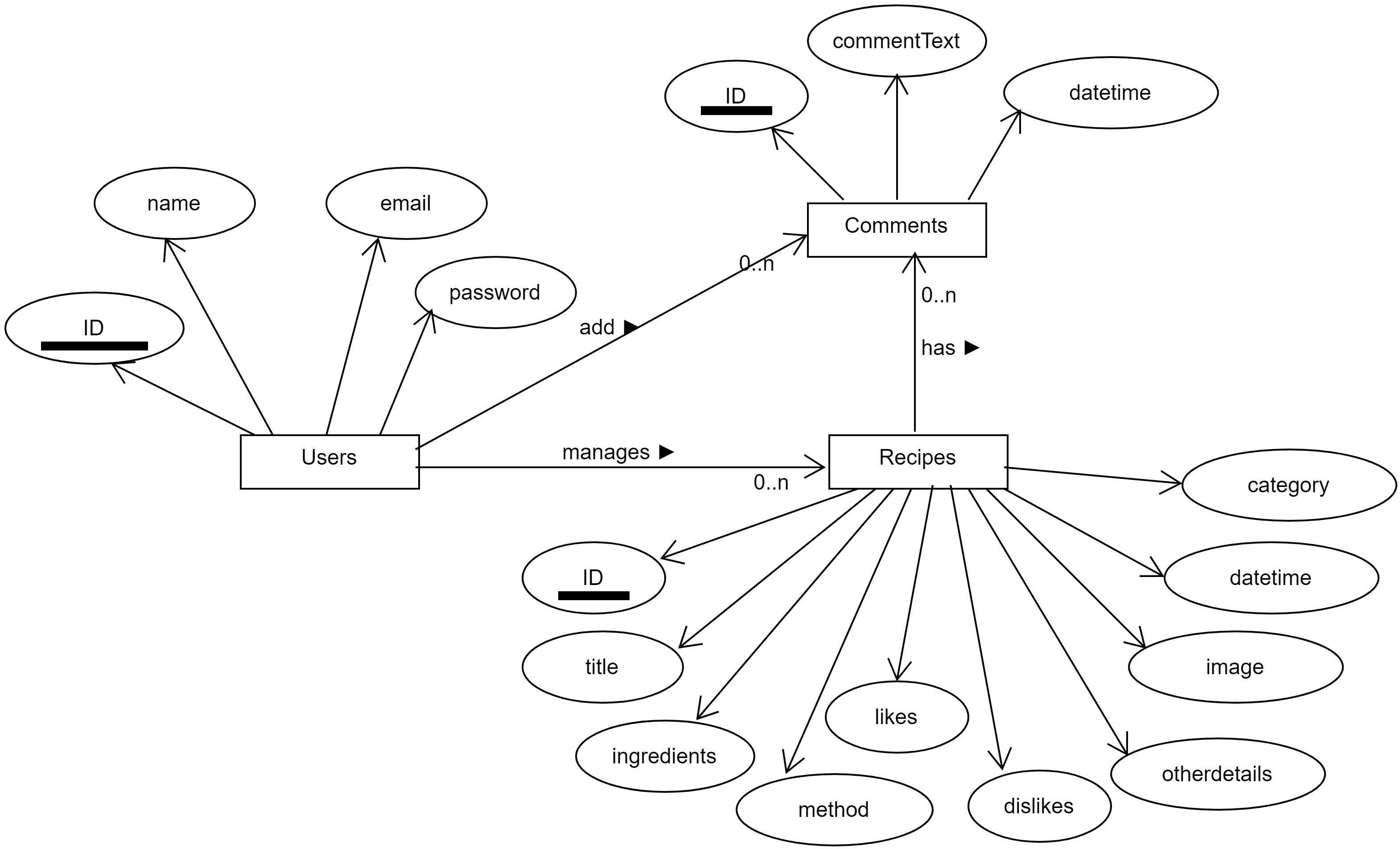
# 4. Design

## 4.1 Class Diagram

****

## 4.2 Use Case Diagram

## 4.3 E-R Diagram

****

## 4.4 Data Dictionary

**Users**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Data type** | **Required** | **Unique** | **PK/FK** | **Ref. Table** |
| 1 | Id | int | true | true | PK | - |
| 2 | name | string | true | false | - | - |
| 3 | email | string | true | true | - | - |
| 4 | password | String | true | false | - | - |

**Recipes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data type** | **Required** | **Unique** | **PK/FK** | **Ref. Table** |
| 1 | Id | int | true | true | PK | - |
| 2 | title | string | true | false | - | - |
| 3 | ingredients | string | true | false | - | - |
| 4 | method | string | true | false | - | - |
| 5 | otherdetails | string | true | false | - | - |
| 6 | image | string | false | false | - | - |
| 7 | Date | DateTime | true | false | - | - |
| 8 | Likes | int | false | false | - | - |
| 9 | Dislikes | int | false | false | - | - |
| 10 | userId | int | true | false | FK | Users |

**Comments**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data type** | **Required** | **Unique** | **PK/FK** | **Ref. Table** |
| 1 | Id | int | true | true | PK | - |
| 2 | commentText | string | true | false | - | - |
| 3 | date | datetime | true | false | - | - |
| 4 | userId | int | true | false | FK | Users |
| 5 | recipeId | int | true | false | FK | Recipes |

# 5. Implementation Details

## 5.1 Modules Created (Brief Description)

* For WCF Project

Project Structure:

**WCFServices**: WCF Service library project

**WebClient:** Web application (ASP.Net framework 4.6.1)

**ConsoleHost**: Console Application (for hosting)

Workflow: Our WCF Service library project is hosted by console application. WebClient can interact with the database using services. First Of all, client can consume services using service reference & service endpoints are configured using console host, which invokes particular service operation. Finally, service handles the provided data and performs necessary operation.

Services created**:**

* **AccountService**: It manages interaction with User. It is used to register & login user.

It also contain operations to view & update profile details.

* **RecipeService:** It is used to manage recipes. It contains operations to add, update, delete & view recipes. Using this, we can also add like or dislike to particular recipe.
* **CommentService:** It is used to add comment to particular recipe & view comments of recipe(s).
* For Rest-API project

Project Structure:

**RecipePortalWebAPI:** Web API project (ASP.Net core 2.1)

**WebClient:**  Web application (ASP.Net framework 4.6.1)

Workflow: WebClient is used to take data from the client or provide data to the client. Using HttpClient, it calls method of Web API to handle particular request. Then Web API maps particular request & uses repository pattern to perform interaction with the SQL server database. Data from the repository is returned to the API, which in-turn forwards it to the webclient.

API Controllers Created:

* **AccountController:** It controls all operation related with user. It provides methods for login, registration, view user details & update user details.
* **RecipeController:** It is used to manage the recipe. It contains method to add, update, delete & view recipes. It has method to update like & dislike count as well.It also handles add comments & view comments functionality of particular recipe.

## 5.2 Function Prototype

**For WCF Project:**

RecipeService Operations:-

* string AddRecipe(Recipe Recipe);
* Recipe EditRecipe(int Id);
* string UpdateRecipe(Recipe Recipe);
* bool DeleteRecipe(int id);
* List<Recipe> GetAllRecipes(int userId);
* Recipe GetRecipe(int id);
* bool AddLike(int id);
* bool AddDislike(int id);
* List<Recipe> GetUserSpecificRecipes(int id);

AccountService Operations:-

* string Register(Users u);
* Users Login(string email, string password);
* void UpdateUserDetails(Users u);
* Users GetUserDetail(int ID);
* DataSet GetUsers();
* void DeleteUser(int ID);
* void Logout();

CommentService Operations:-

* string AddComment(Comments comment);
* IEnumerable<Comments> ViewComments(int recipeId);

**For Rest API Project:**

AccountController Methods:-

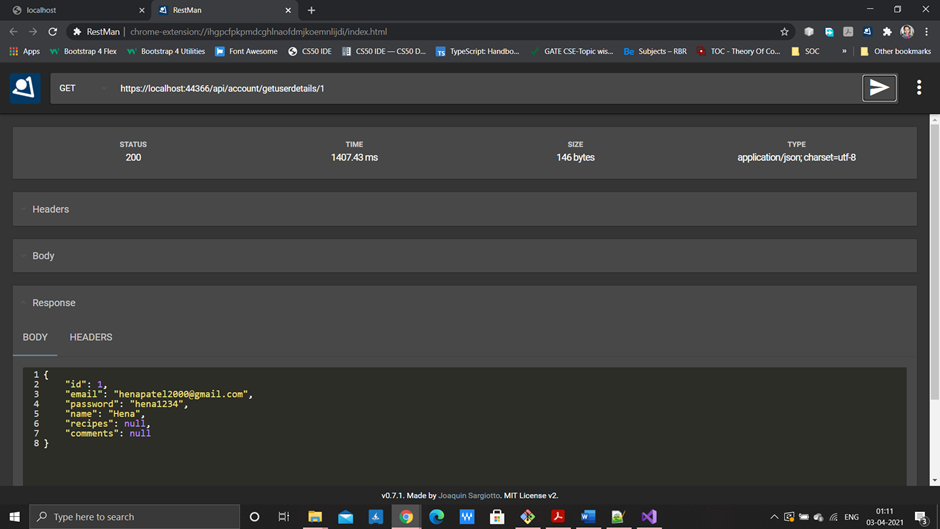
* public IActionResult Register(User user); **[**HttpPost**]**
* public IActionResult Login([FromBody] User user); **[**HttpPost**]**
* public IActionResult GetUserDetails(int id); **[**HttpGet**]**
* public IActionResult UpdateUserDetails([FromBody] User user); **[**HttpPut**]**

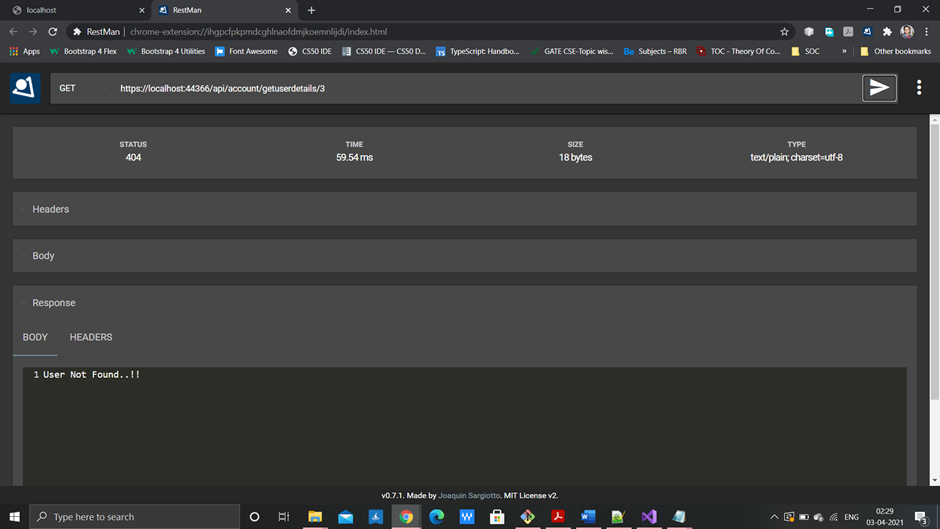
RecipeController Methods:-

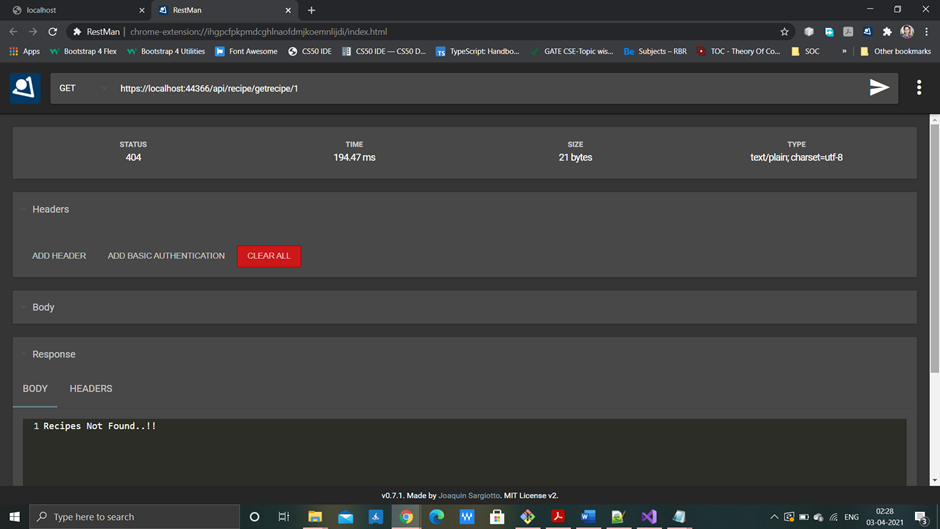
* public IActionResult GetAllRecipes(); **[**HttpGet**]**
* public IActionResult GetAllRecipes(string type, int id); **[**HttpGet**]**
* public IActionResult GetRecipeById(int id); **[**HttpGet**]**
* public IActionResult AddRecipe(Recipe newrecipe); **[**HttpPost**]**
* public IActionResult AddLike([FromBody] int id); **[**HttpPost**]**
* public IActionResult AddDislike([FromBody] int id); **[**HttpPost**]**
* public ActionResult DeleteRecipe(int id); **[**HttpDelete**]**
* public IActionResult UpdateRecipeDetails(Recipe changerecipe); **[**HttpPut**]**
* public IActionResult AddComment(Comment comment); **[**HttpPost**]**
* public IActionResult ViewComment(int id); **[**HttpGet**]**

# 6. Testing

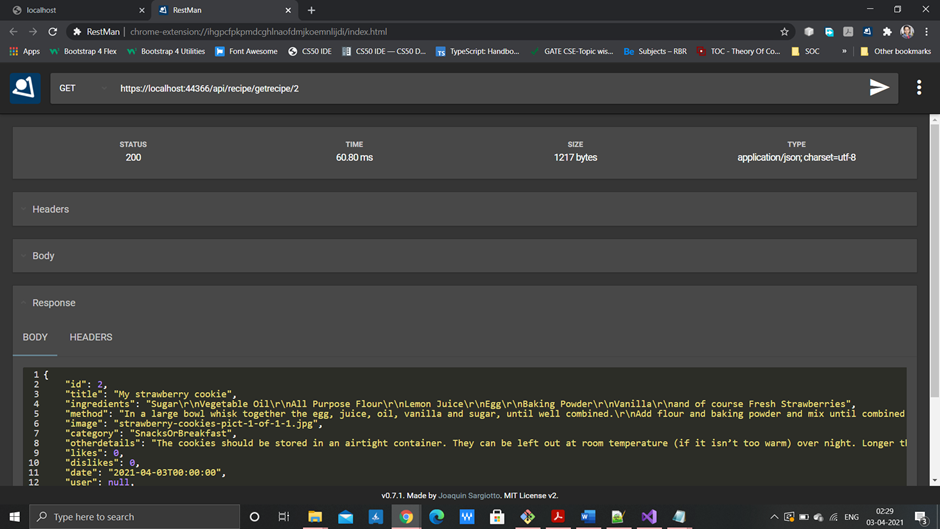
We have used Restman tool to perform testing of Web-API. Client testing is performed manually.





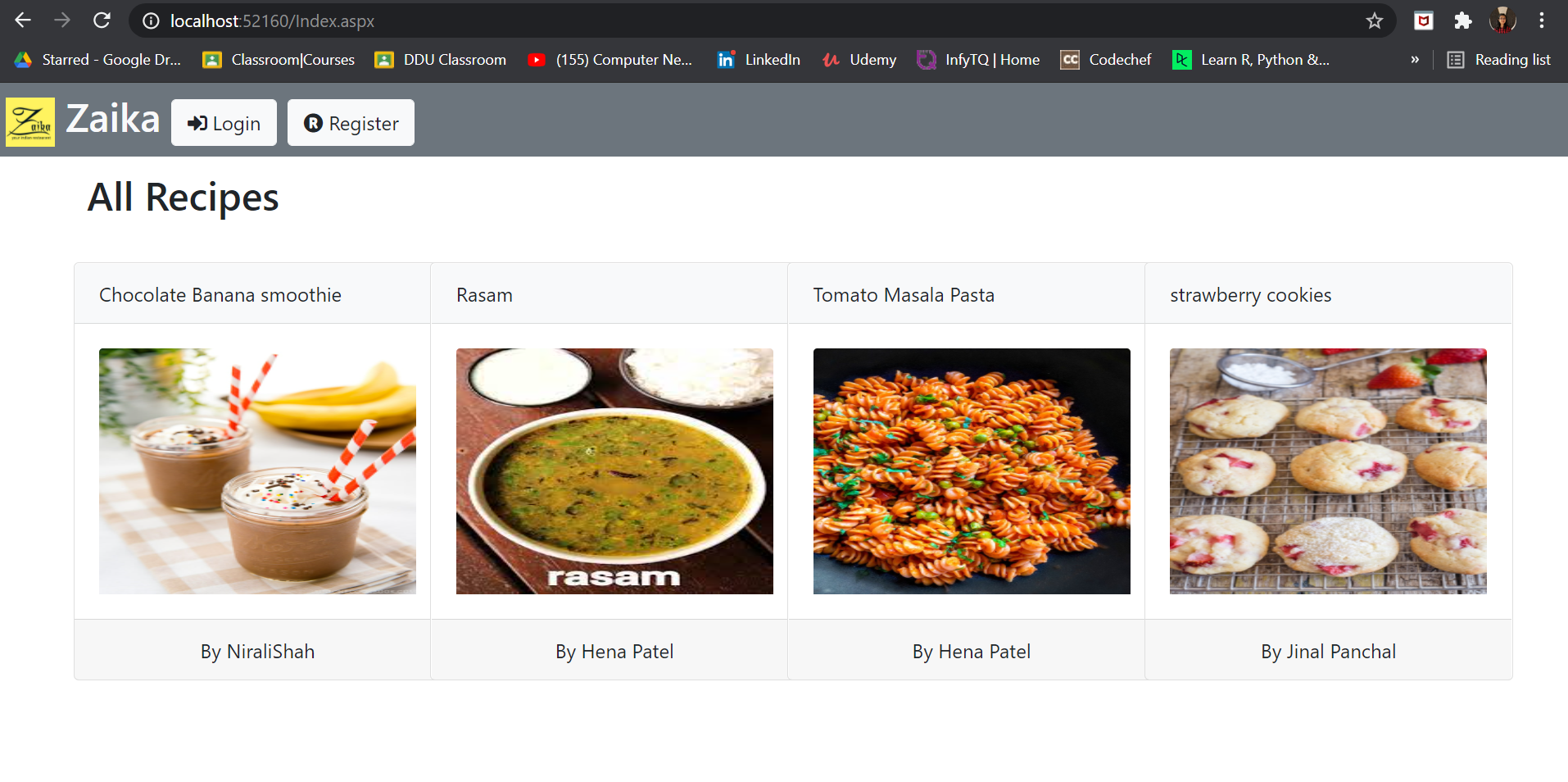


# 

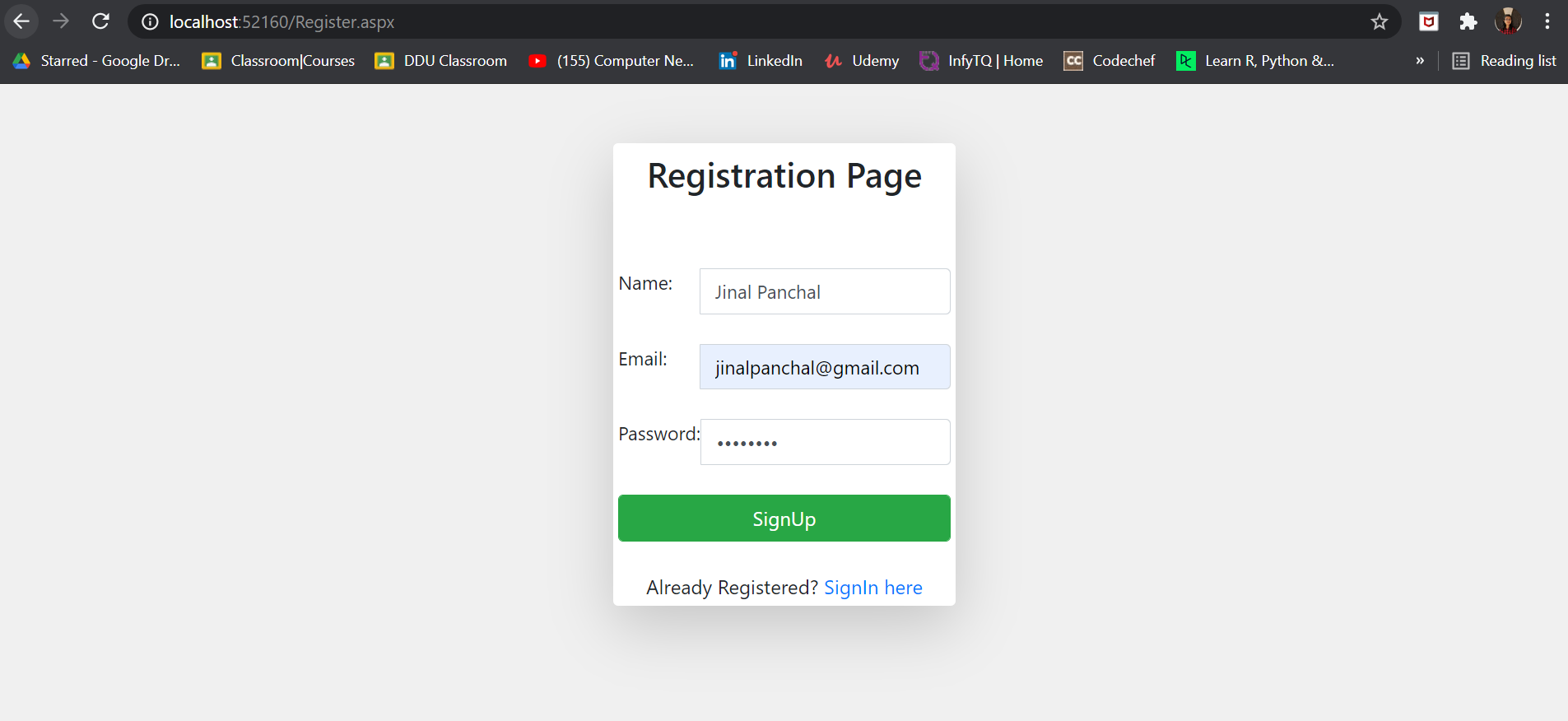


# 7. Screen-shots

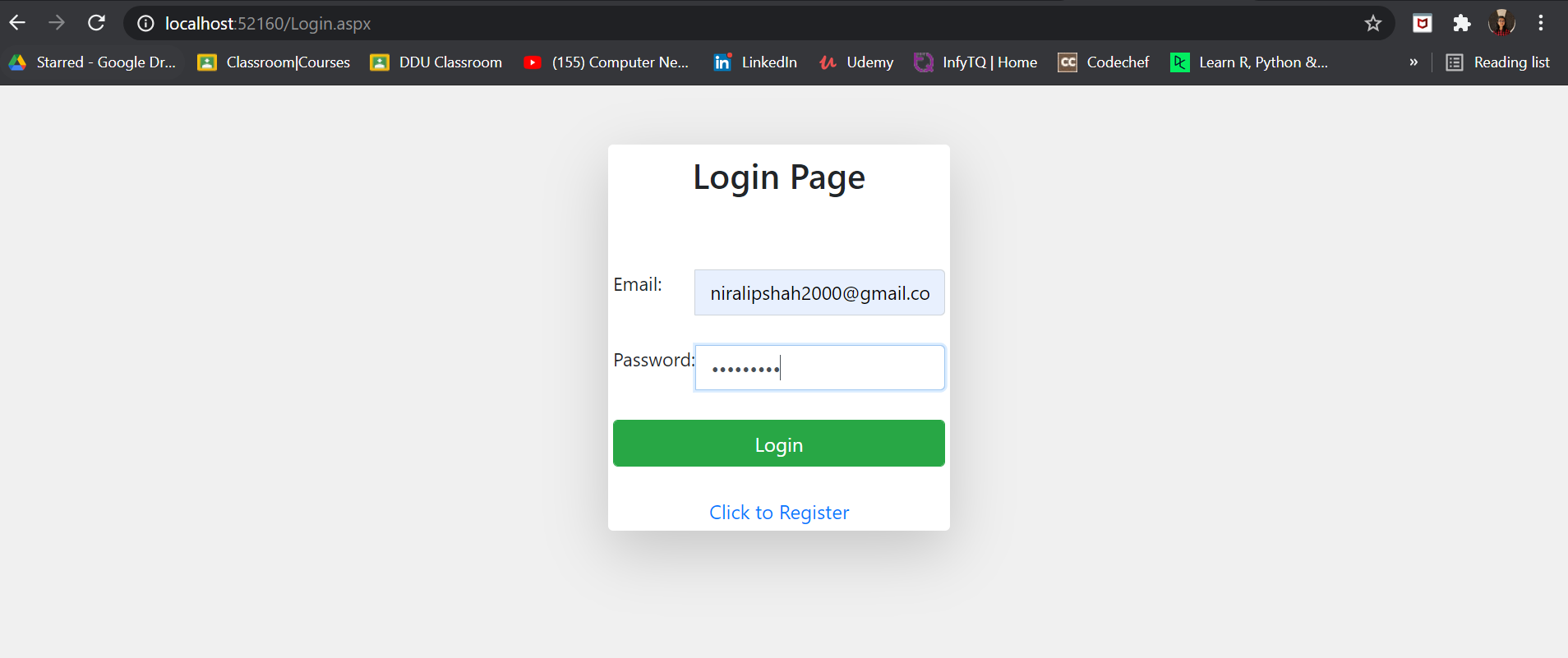
**7.1 Index Page**



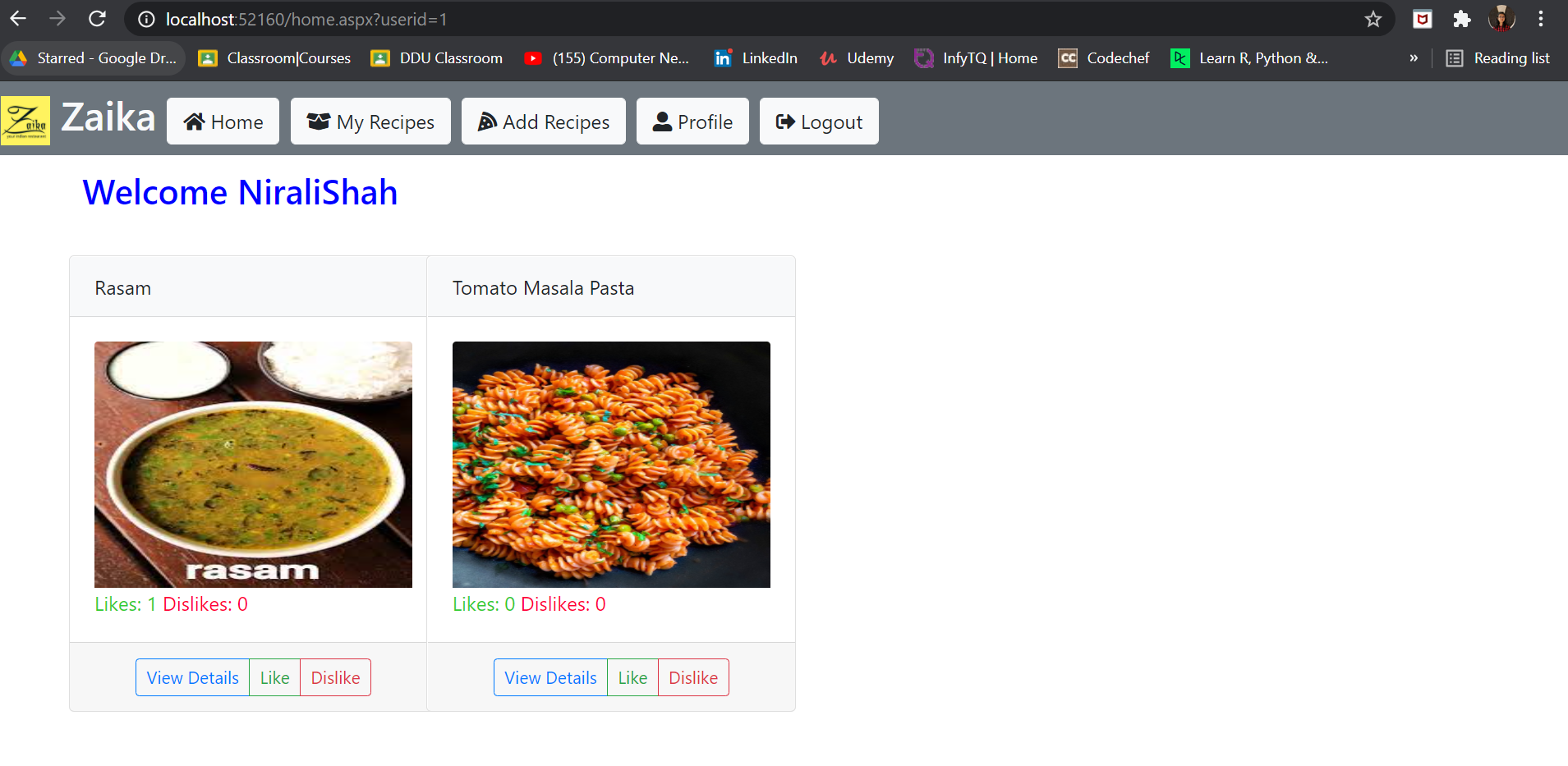
**7.2 Sign Up**

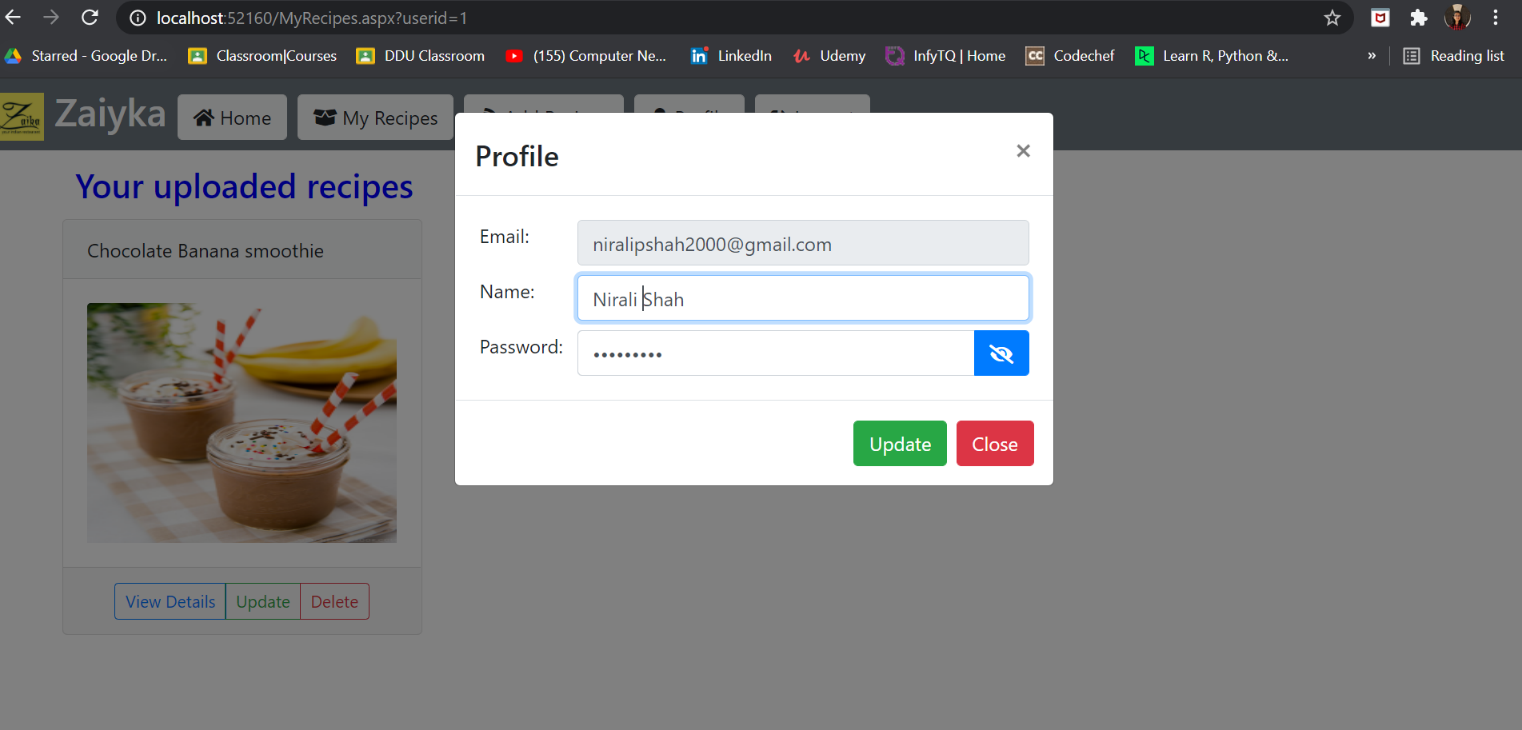


**7.3 Login**

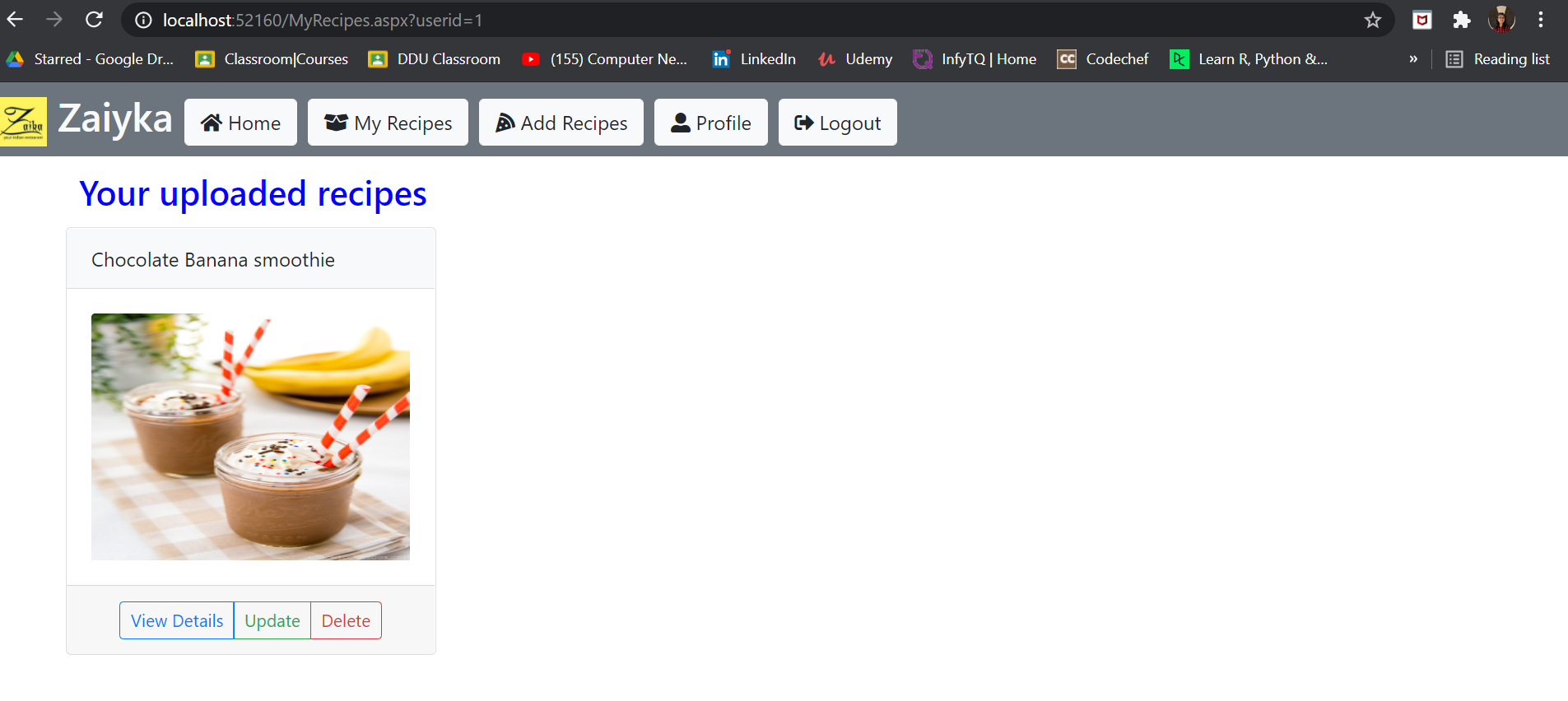


**7.4 User Homepage(Other users recipes are displayed)**

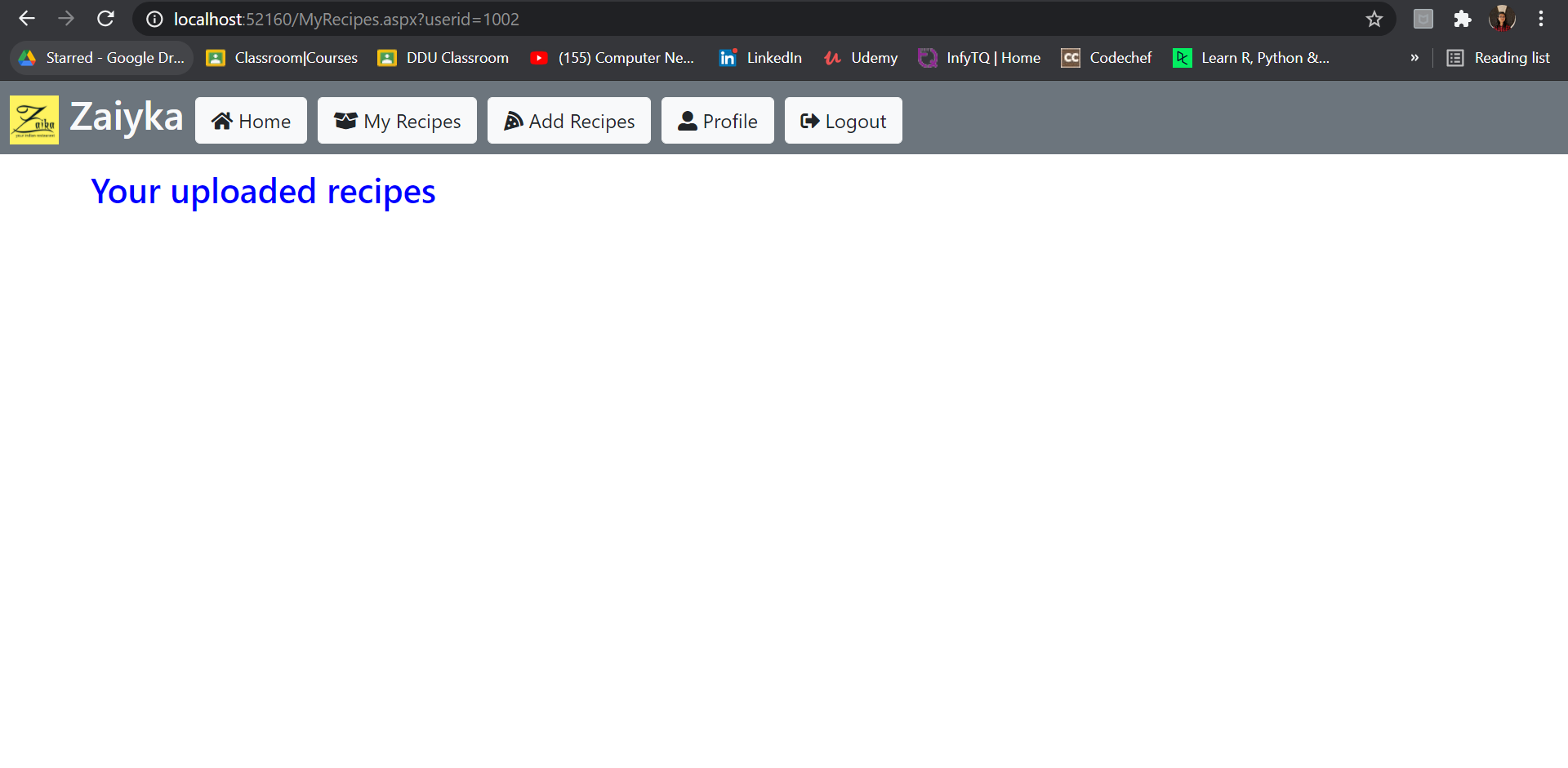


**7.5 Update Profile**

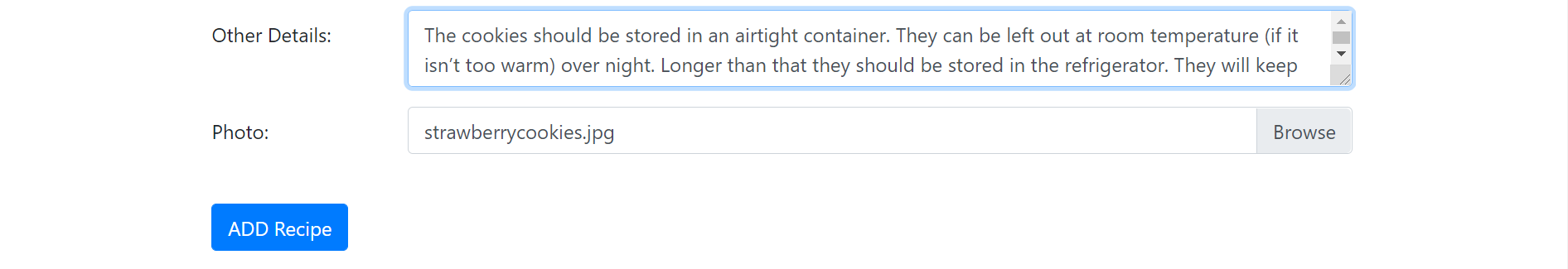
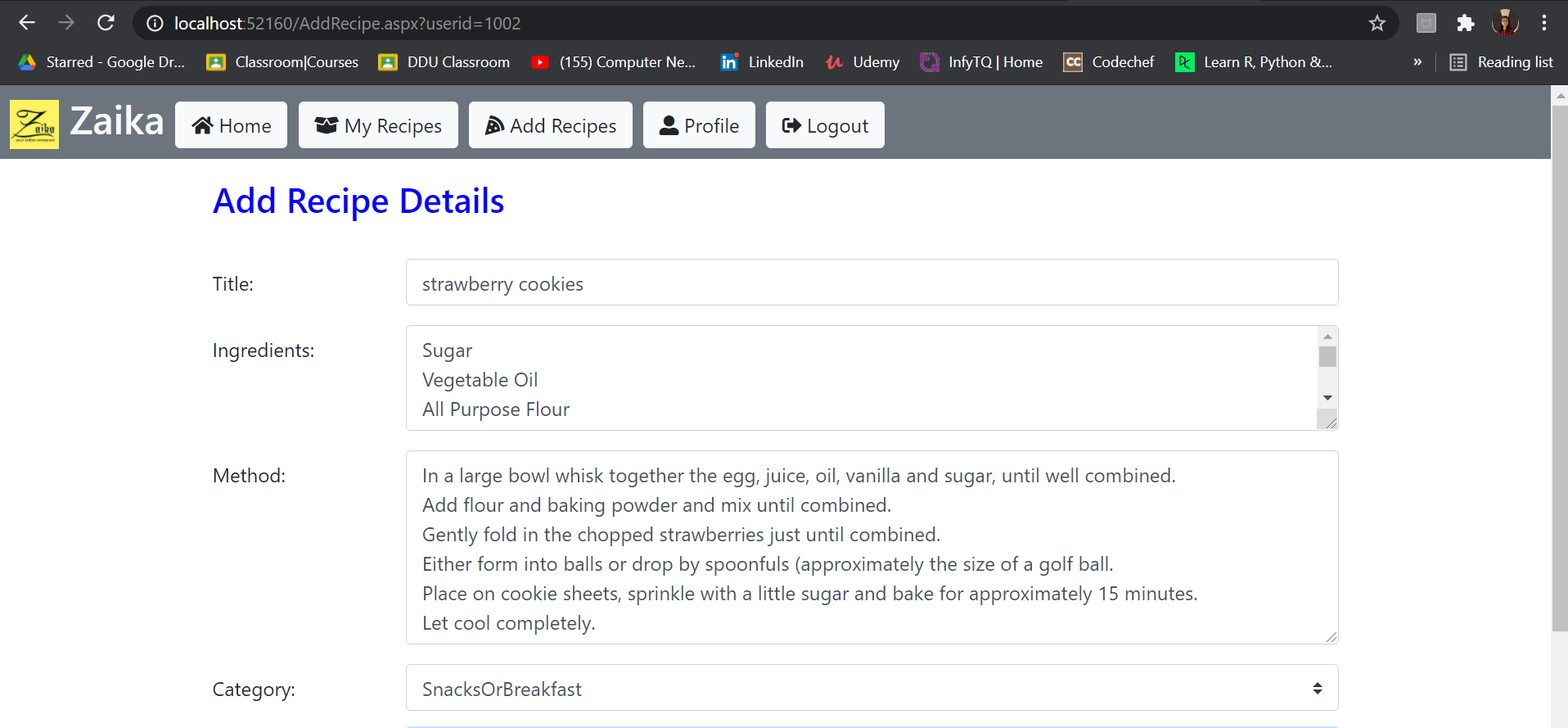
**7.6 My Recipes(users own recipes)**



🡪when you have not uploaded any recipe



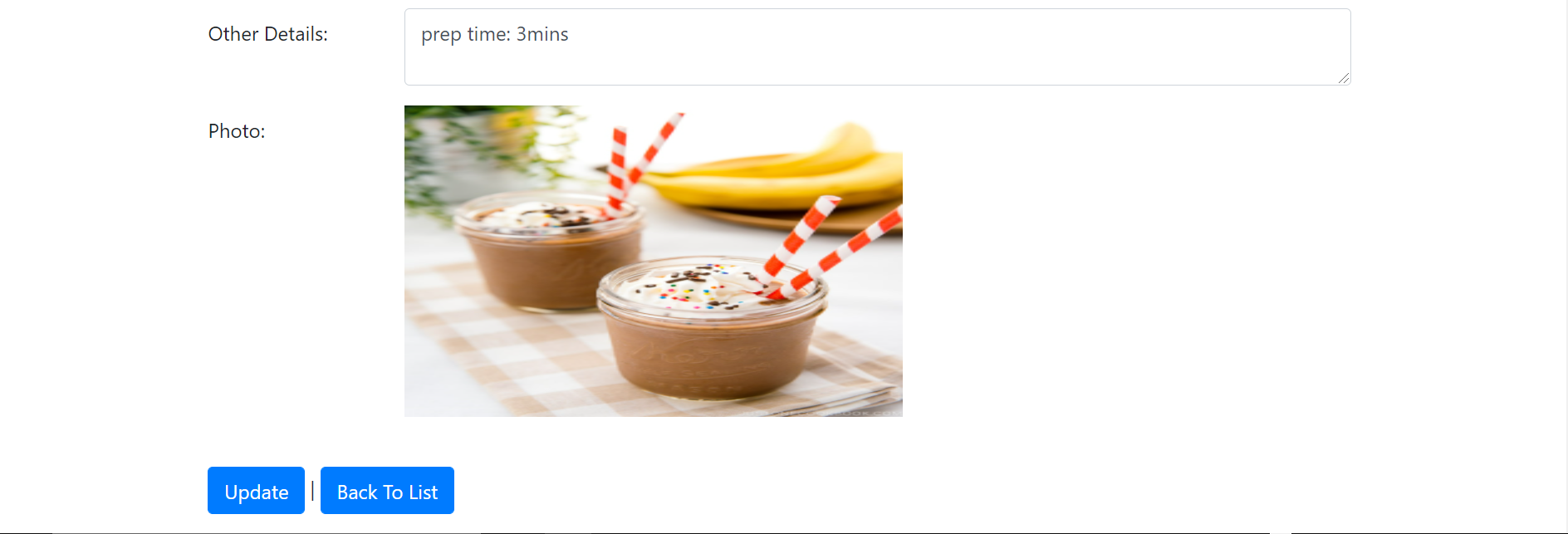
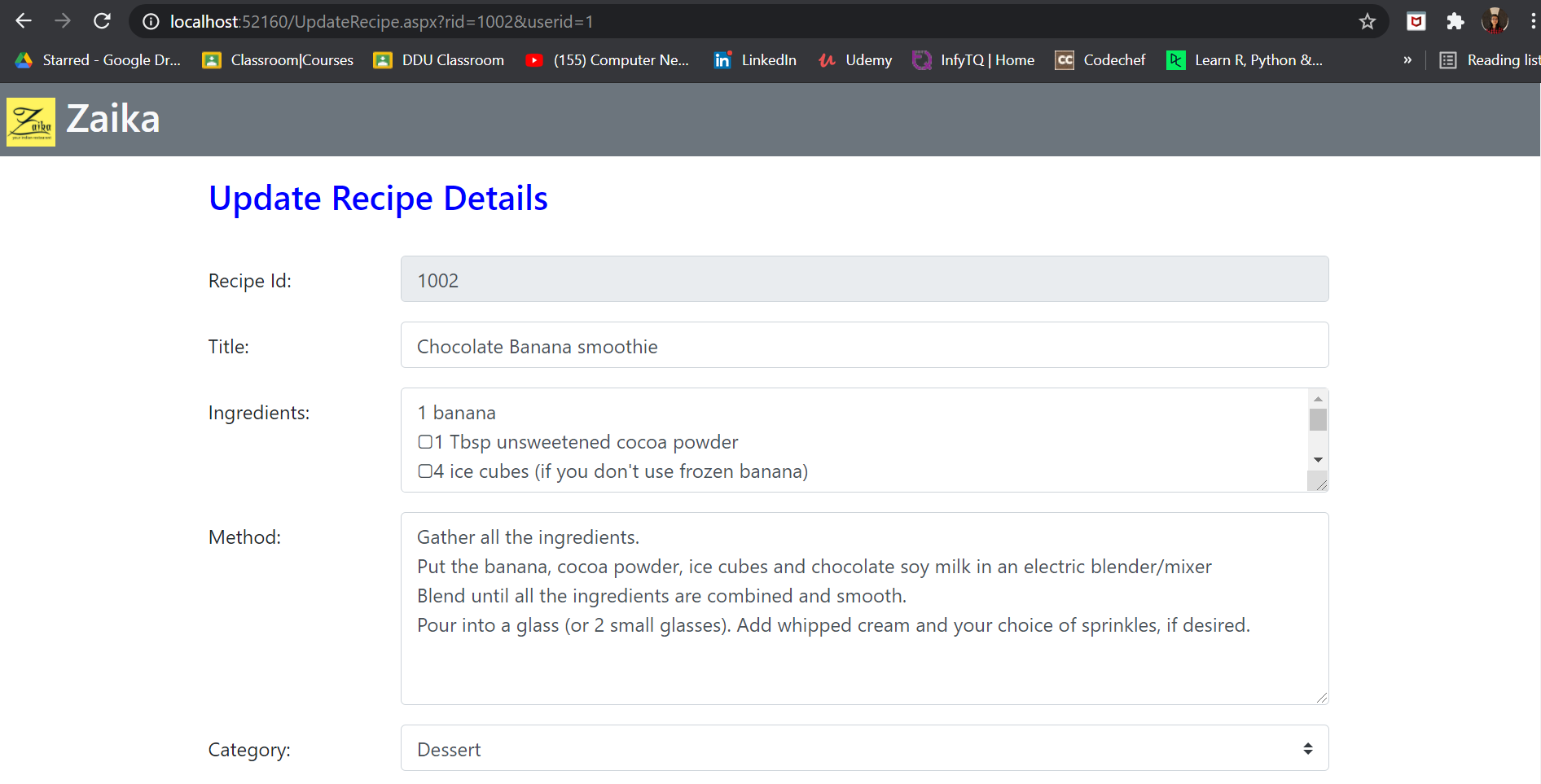
**7.7 Add Recipe**



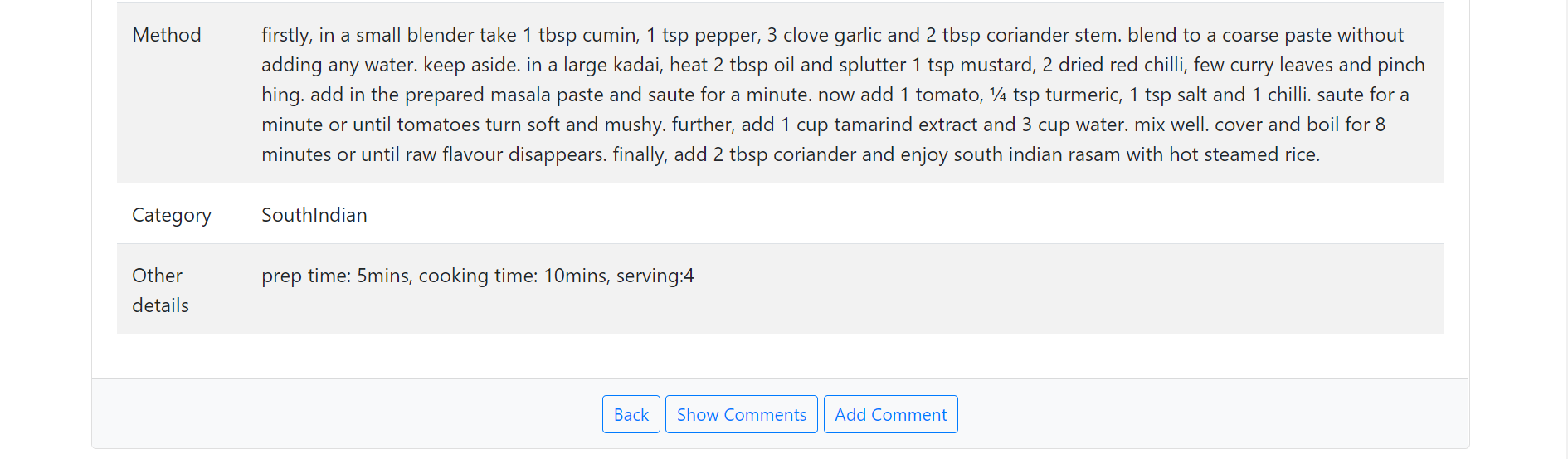
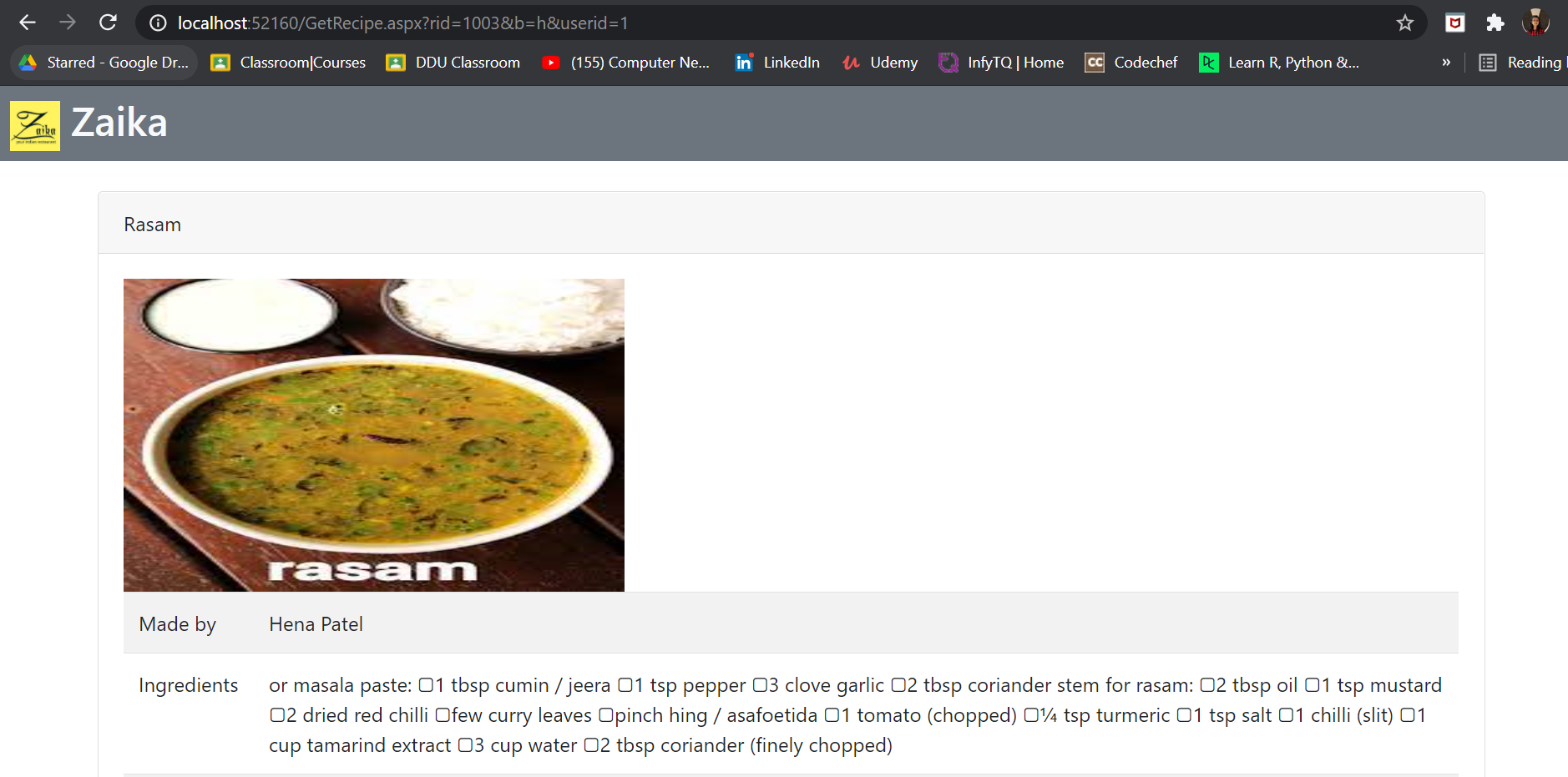
**7.8 Delete Recipe**



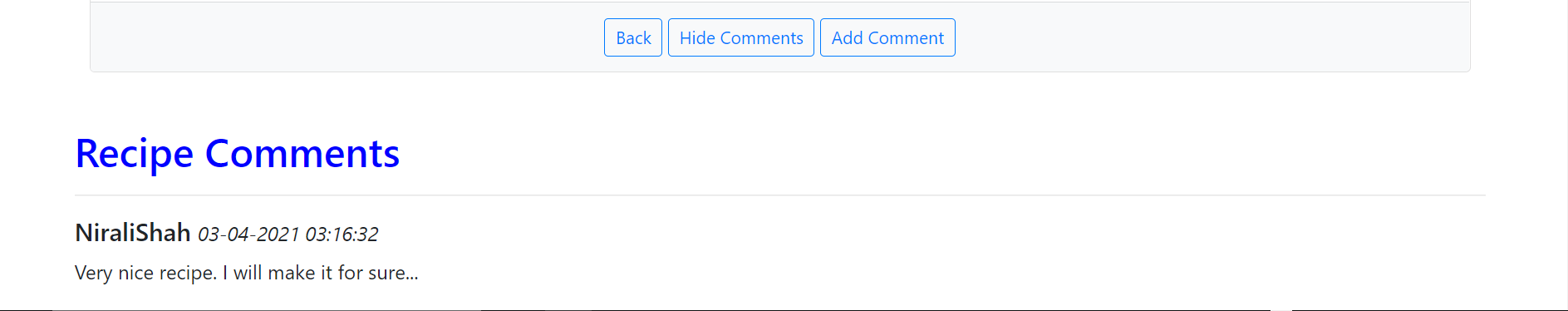
**7.9 Update Recipe**



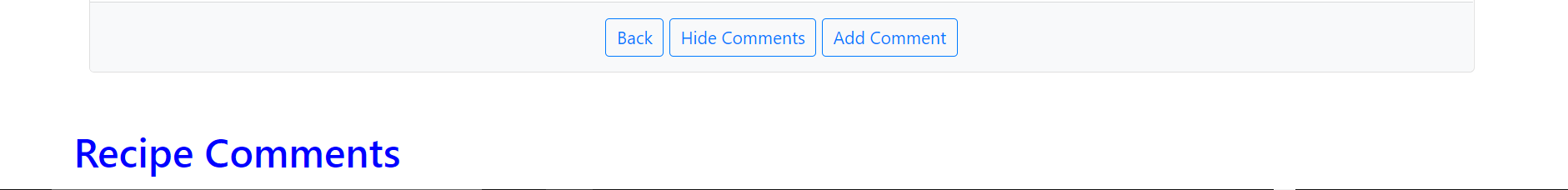
**7.10 View RecipeDetails**



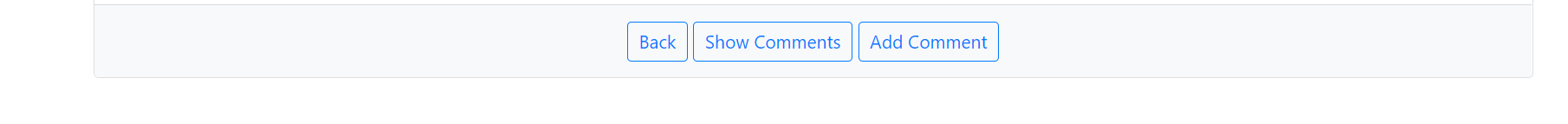
**7.11 ShowComments**



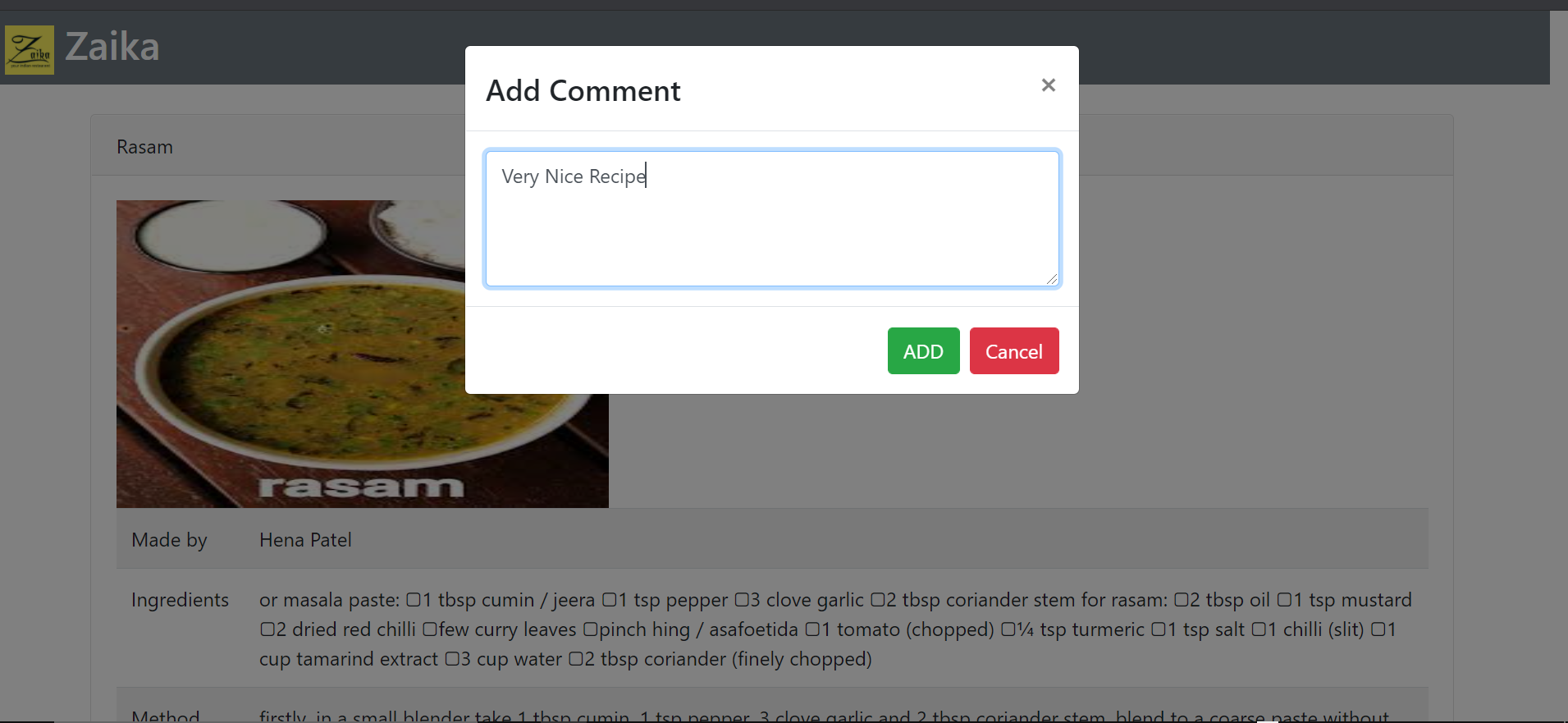
🡪when no comments are available.



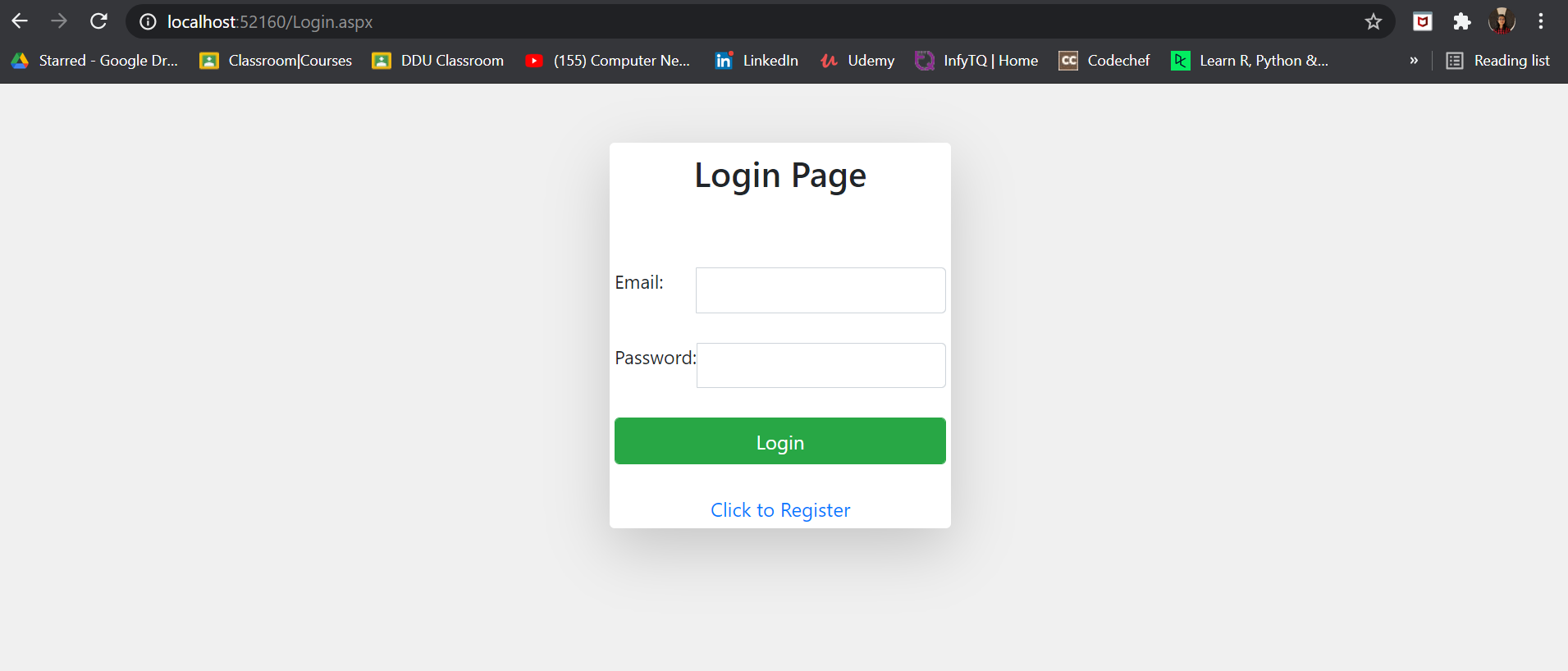
**7.12 Hide Comments**



**7.13 Add Comments**



**7.14 After Logout**



# 8. Conclusions

The functionality implemented in the system was done after understanding all the

system modules according to the requirements.

Functionalities successfully implemented in the system:-

* User registration
* Login
* Logout
* Add,update,delete recipe
* View recipes
* Add comment
* View comments
* Add like or dislike to recipeId

After implementation of the system, comprehensive testing was performed to check for possible errors or bugs in the system.

# 9. Limitations and Future Extensions

## 9.1 Functionalities Not Implemented

* Forgot password & reset password
* Sharing of recipes

## 9.2 Future Extensions

* Rating of recipes
* Add reply to comments
* Like or dislike any comment

# 10. Bibliography

For solving errors & exceptions:-

* https://stackoverflow.com/questions/18465334/getting-thread-abort-exception-while-using-response-redirect
* https://docs.microsoft.com/en-us/dotnet/api/system.threading.threadabortexception?view=net-5.0
* https://forums.asp.net/t/2163298.aspx?How+can+I+resolve+System+Net+Sockets+SocketException+An+existing+connection+was+forcibly+closed+by+the+remote+host

For cooking recipe data:-

* https://www.mycakeschool.com/recipes/cherry-cake-a-scratch-recipe/
* https://hebbarskitchen.com/

For model validation purpose:-

* [https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-3.1#:~:text=expects%20an%20integer).-,Model%20validation%20occurs%20after%20model%20binding%20and%20reports%20errors%20where,rating%20between%201%20and%205).&text=%2FIndex%22)%3B%20%7D-,Web%20API%20controllers%20don%27t%20have%20to%20check%20ModelState.,have%20the%20%5BApiController%5D%20attribute](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-3.1#:~:text=expects an integer).-,Model validation occurs after model binding and reports errors where,rating between 1 and 5).&text=%2FIndex)