LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6004NI – Application Development

Assessment Weightage & Type

30% Group Coursework

Year and Semester

2020-21 Autum

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1.	**Salina Budhathoki**	**NP01CP4A180167**	**18029966**
2.	**Shreeti Mool**	**NP01CP4A180148**	**18029931**
3.	**Ashmita Serpuja**	**NP01CP4A180154**	**18029943**
4.	**Utkarsh Poudel**	**NP01CP4A180326**	**18029182**

Assignment Due Date: 30th April 2021

Assignment Submission Date: 30th April 2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Acknowledgment

We would like to take this opportunity to express gratitude towards our module leader Mr.Dhurba Sen and Tutor Mr.Sushil Sapkota for guiding us and giving their time to us throughout the coursework. Their guidance was very helpful to complete this project. This project would not have been completed by the deadline without their help.

Also, we are very thankful to my college for providing the essential resources and materials which have been very helpful throughout this project.

Table of Content

Table of Figures

Table of Tables

1 Introduction

This coursework was given to us to understand the C# programming language based on Asp.net in which we have to solve the problem based on the experience gained from the classroom and the research done. The purpose of the coursework is to deliver the study about the programming language structure and code to give the best solution to the stock management system.

The Stock Management System project is a full desktop application built with Visual Studio Software and.Net technology. The project's main goal is to create Stock Management System Model software, which will display all stock-related details. It's an intranet-based desktop application with an inventory management admin component.

This desktop program is focused on an organization's stock management. The application includes general sales information, purchasing details, customer details, and the remaining stock. There is also the option to update or edit the inventory. We have to deliver the best result to show the work that the stock shop design we are going to deliver for this course is the best and a lot of effort has been done by our group through research.

2 Instructions

1. Home Page

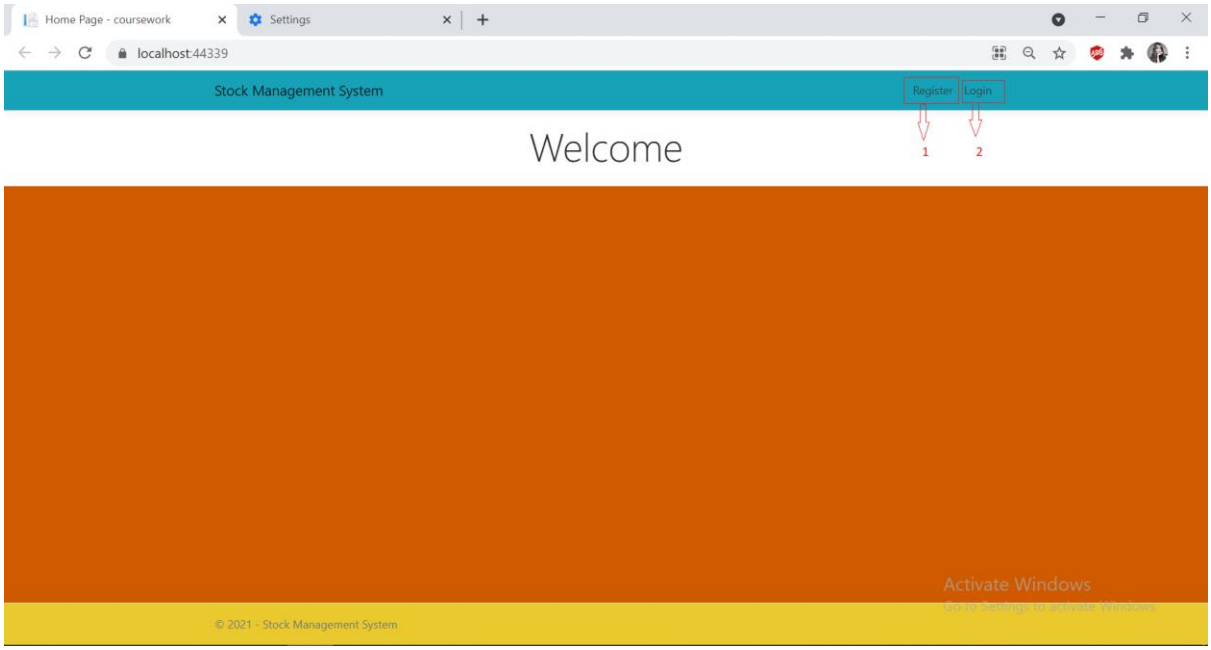

Figure 1: Manual home page

1: The user register for the first time.

2: The user log in into the system after the registration.

2. Register Page

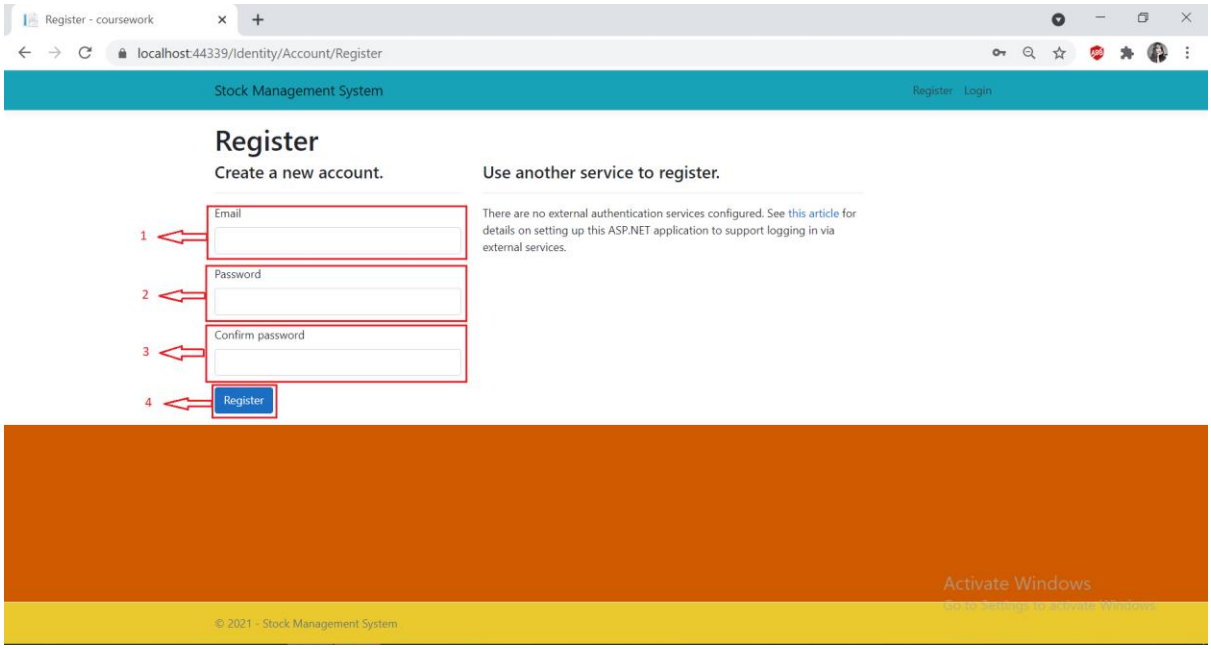

Figure 2: Manual of register page

1: Firstly, Enter your email

2: Then, Enter your password

3: Confirm your password

4: At last click, Register Button which will help you to register user.

3. Login Page

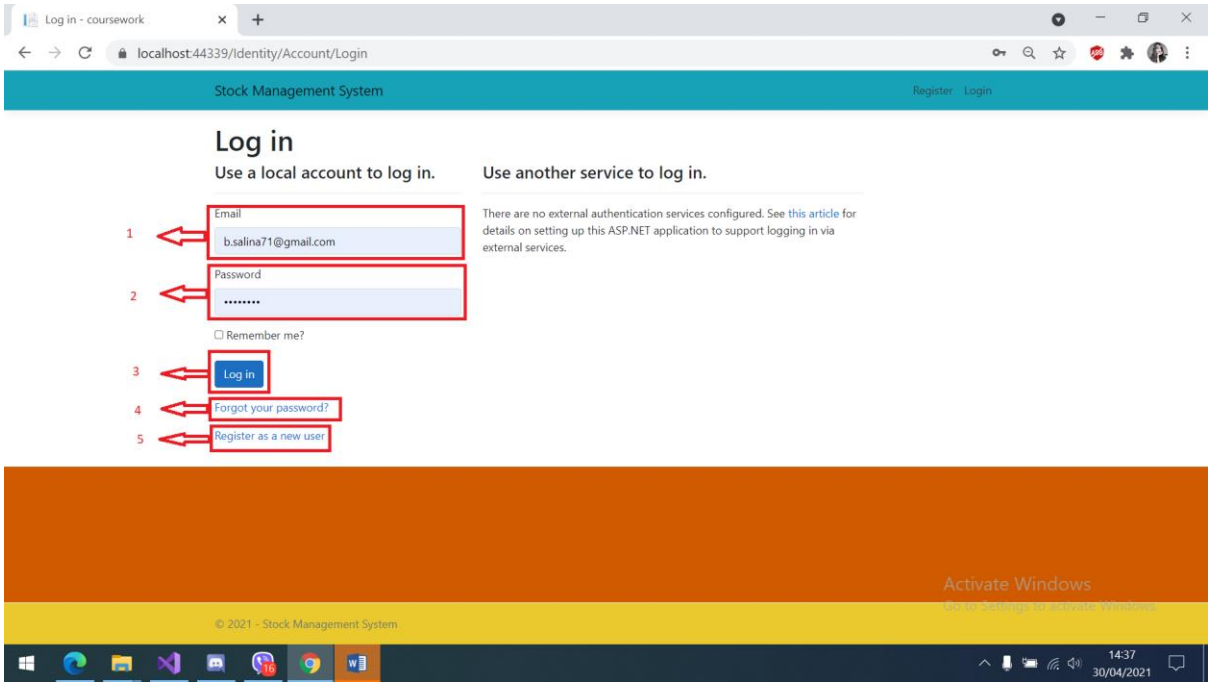

Figure 3: Manual of login page

1: Enter the email you registered.

2: Enter your password (It should match with the registered password)

3: Login Button, It will redirect you to home page.

4: In case you forgot your password you can reset your password.

5: If you're not registered user you won't get to login. It is for registering new user.

4. Categories Page

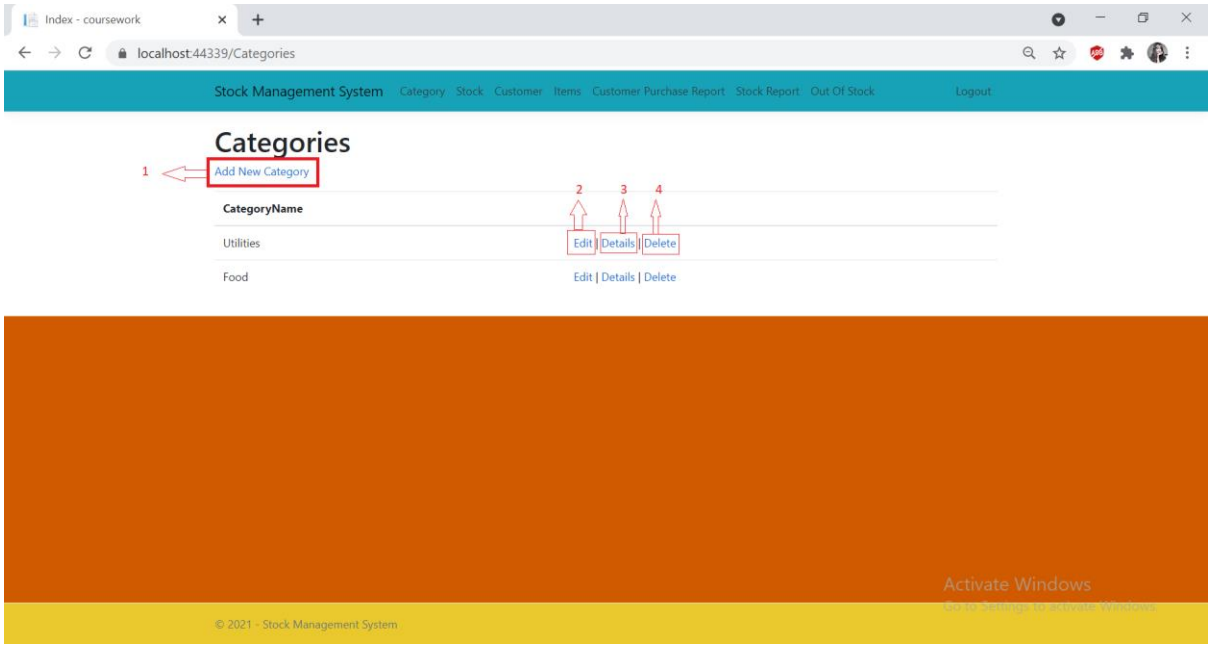

Figure 4: Manual of Categories page

1: It is for adding new categories in the list.

2. If you want to edit the data in the list, edit button is for that.

3. Similarly to show the details of categories details button is clicked.

4: To delete from the list of categories, it is used.

5. Customer Purchase Report

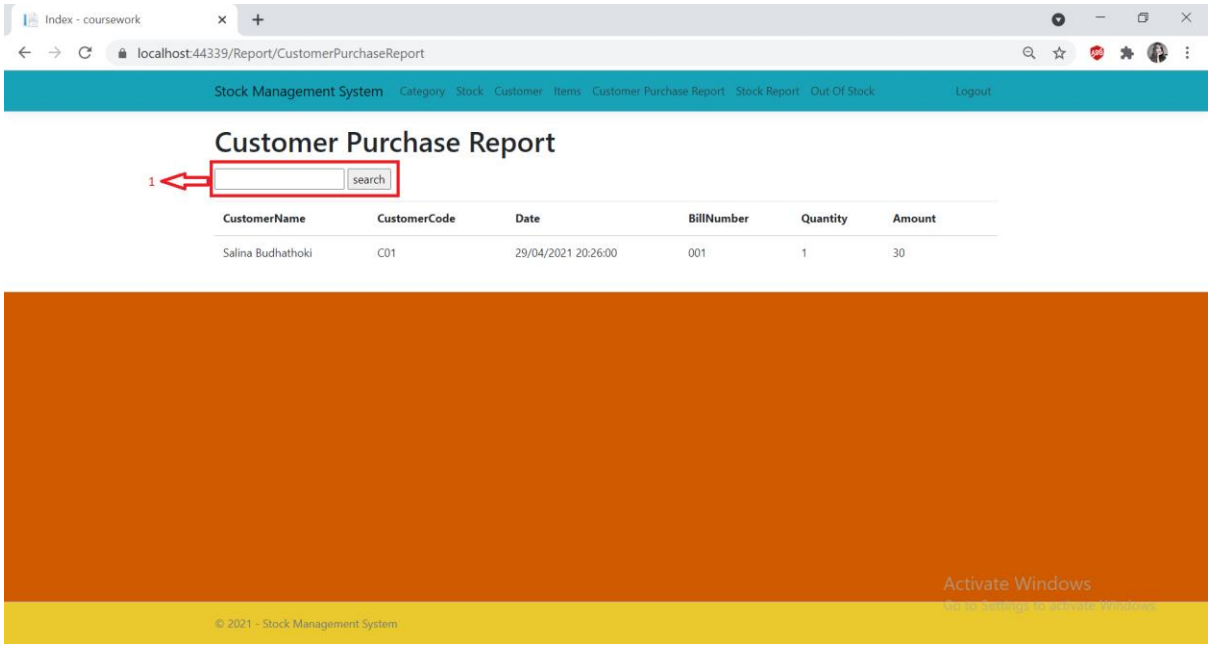

Figure 5: Manual of Customer report page

1: It is used for searching the customer report through customer name if there are more than 1000 user which will make easy to search respected person.

6. Customer Search Report

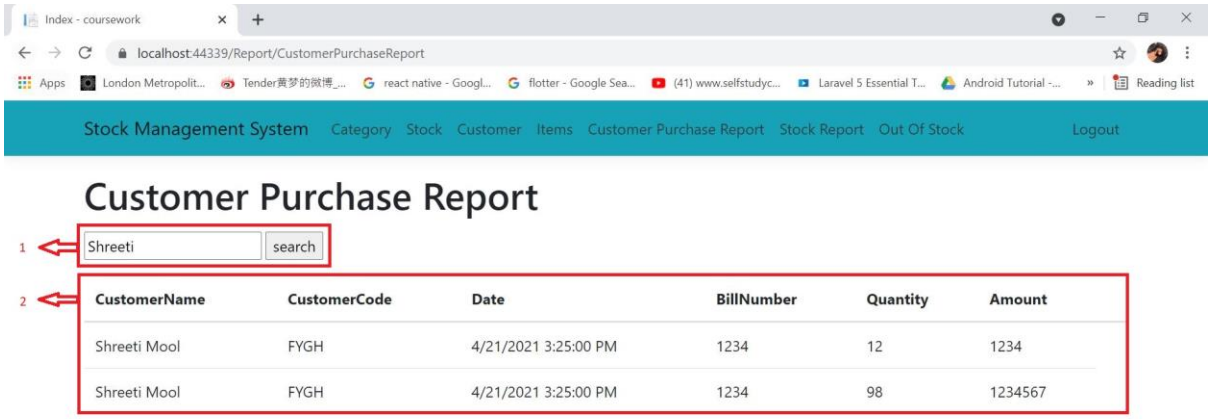

Figure 6: Customer Search Report

1: Searching the customer report adding name.

2: Showing the data of customer i.e. Shreeti Mool.

3 Description of Solution

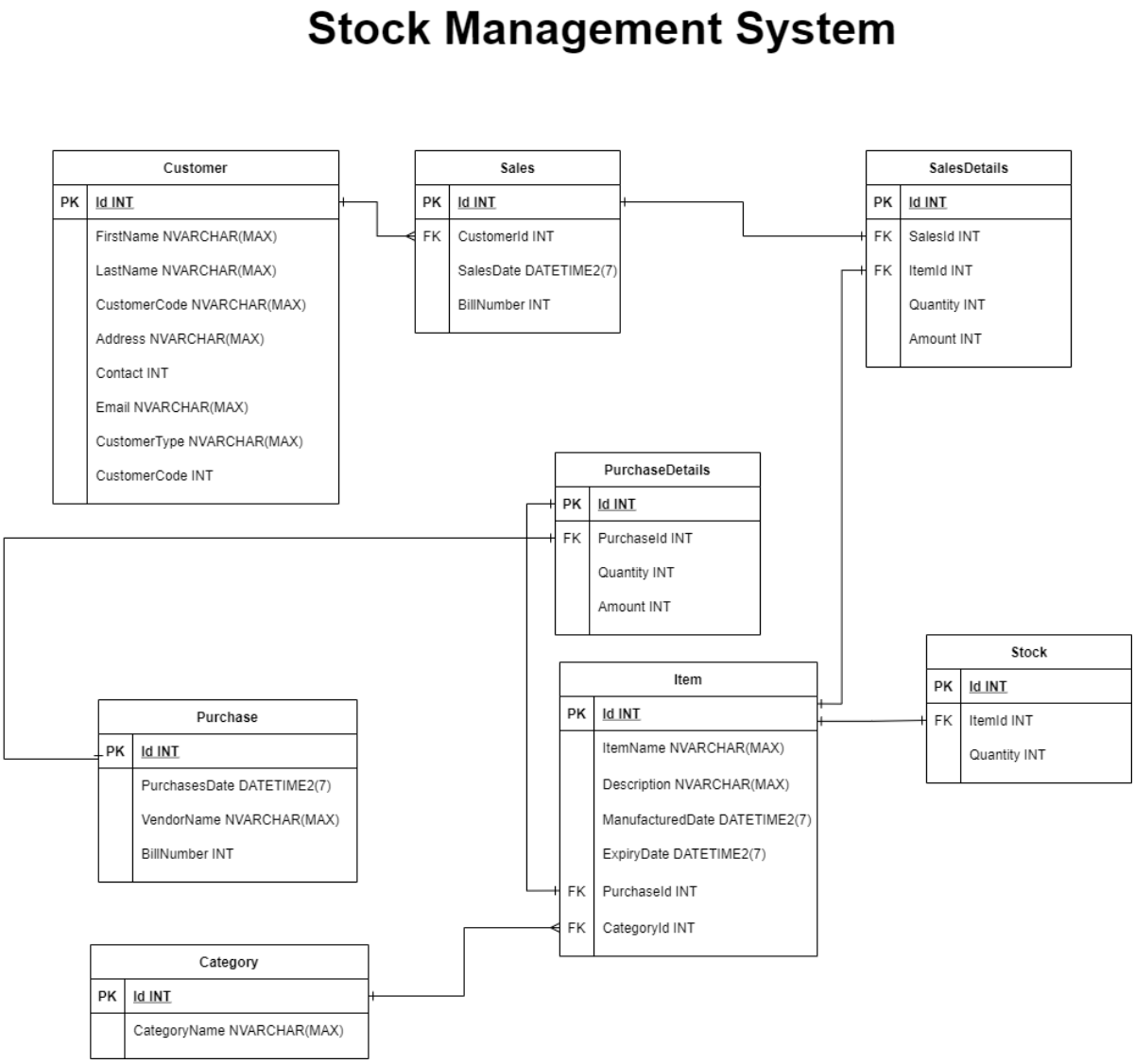

Figure 7: ERD

Home Page

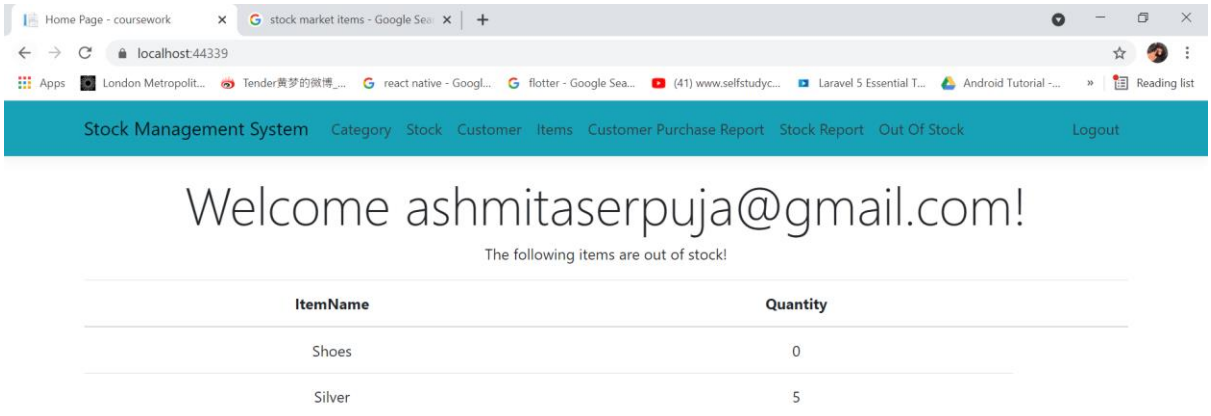

Figure 8 - Home Page

While opening the home page, the index method is called which displays the item list where the stock is unavailable.

Customer Page (Index)

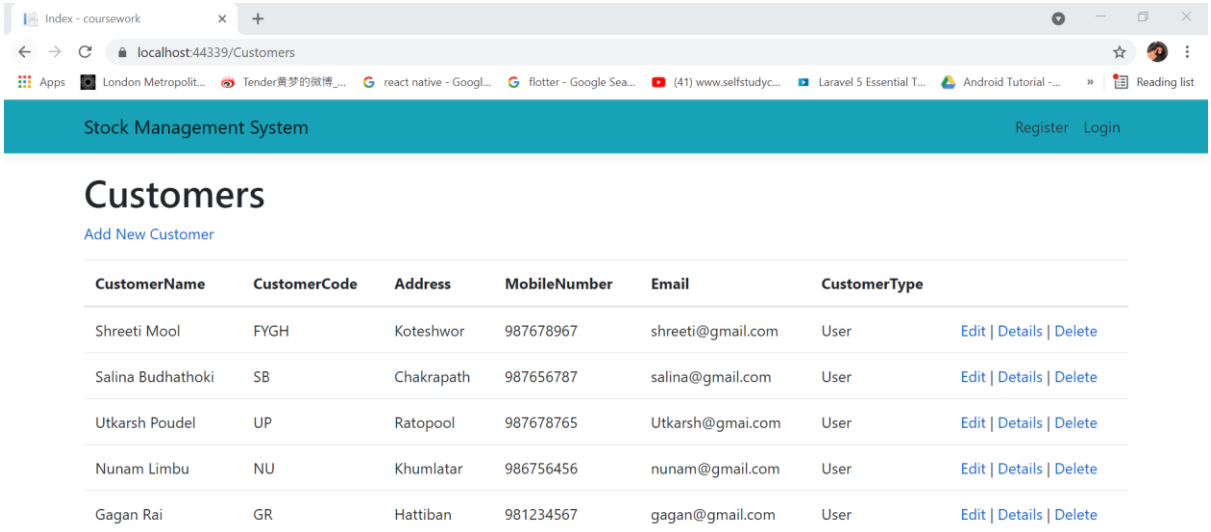

Figure 9 - Customer Index Page

While opening the customer page, the index method is called which displays the list of customers

Likewise, the rest of the pages also have the same functionality.

Customer Page (Create)

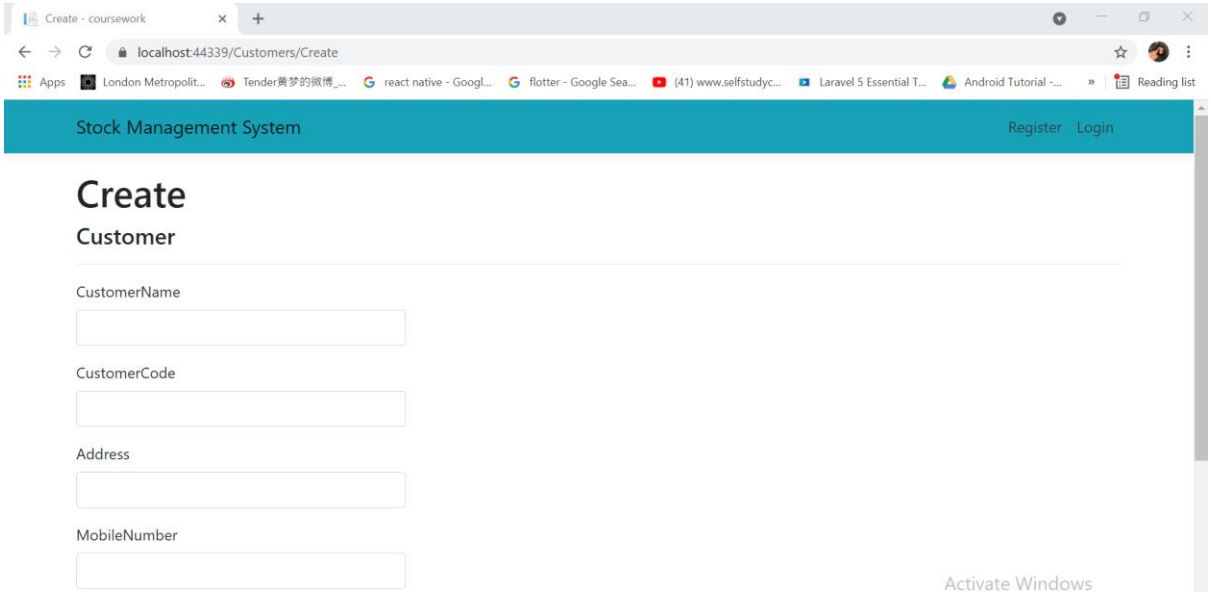

Figure 10 - Customer Create page

While adding new customer details, create method is called which allows the customer to add their credentials and those data are send to customer modules and adds to the database.

Likewise, the rest of the create pages also have the same functionality.

Report Page

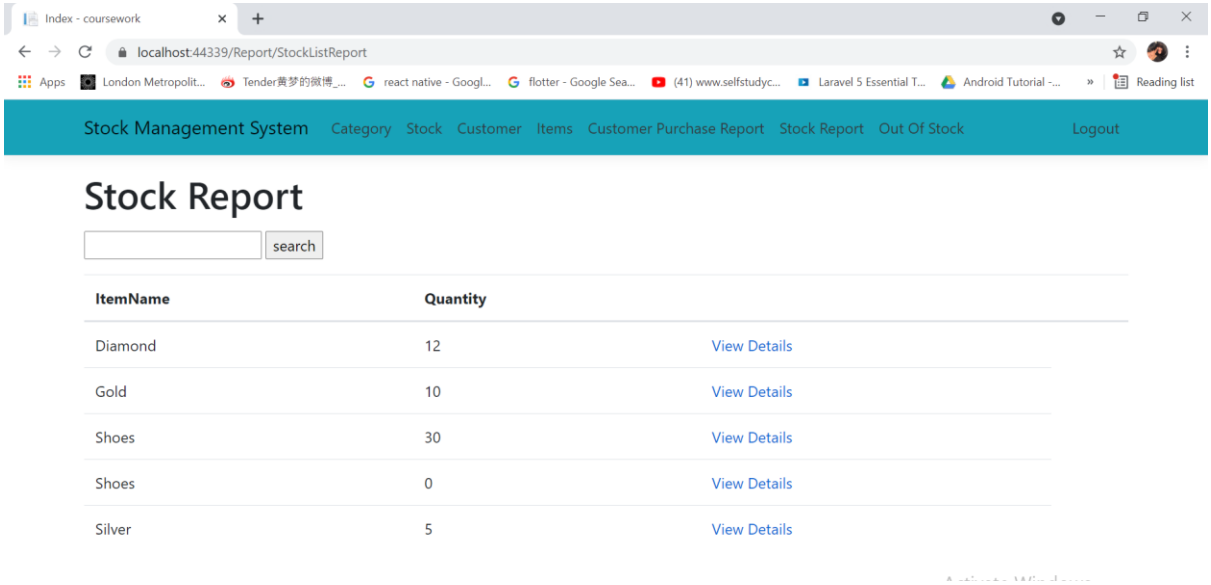

Figure 11 - Stock Report Page

On this page, stocklistreport method is called which displays the product list along with the quantity with the help of SQL query. It also allows to search data according to the product name.

4 Software Architecture

The definition and structuring of a solution that satisfies technological and operational requirements are known as software architecture. Security, efficiency, and manageability are just a few of the attributes that software architecture optimizes. These choices affect the application's efficiency, upkeep, results, and overall success. (Techopedia, 2021)

The Model-View-Controller (MVC) architectural pattern divides an application into three logical components: model, view, and controller. Each of these components is designed to handle unique aspects of application development. MVC is a common industry-standard web development platform for developing scalable and extensible projects. (tutorialspoint, 2021)

More Explanation for MVC pattern is the appendix section Software Architecture

5 Description of Classes

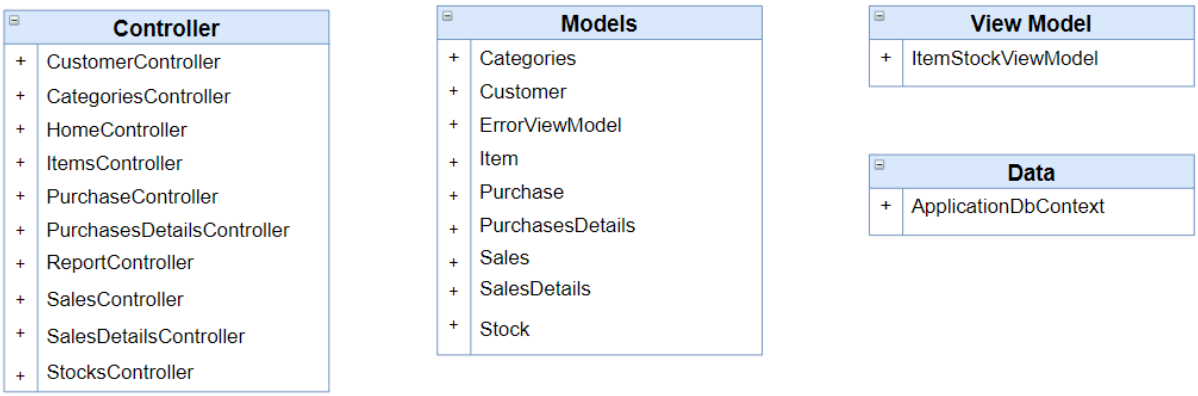

Figure 12: Class Diagram

6 Description of Methods

Since there are a lot of classes in this system, it was a huge time-taking and error-prone program but with help from our tutors and colleagues, we completed this system. There are a lot of controllers and models used in this system as it was a really big system. The controllers and the views here have individual components for every feature.

Some of the methods are described below:

CATEGORY TABLE

S. No	Methods	Description
1	Index()	Listing the category data. It is a written type
2	Details(int? id)	Shows the details of a particular selected category and also checks whether the id is null or not. It is a written type
3	Create()	Displays category form.
4	Create([Bind("Id,CategoryName")] Category category)	Checks the data whether it is valid or not and if it is valid, the data is stored in the database and after storing the data it is returned to the index page
5	Edit(int? id)	Displays the edited form along with selected category data
6	Edit(int id, [Bind("Id,CategoryName")] Category category) {	Allows the user to edit the data of available selected category.
7	Delete(int? id)	Allows the users to delete the selected category. Also allows whether it is null or not and if it is null then the data can be deleted
8	CategoryExists(int id)	Checks whether the particular Id exists or not.

9	CategoriesController(ApplicationDbContext context)	Initialize the data members of new object generally.

Table 1: Category controller table

Likewise in the table of items controller, customer controller, purchase controller, purchase detail controller, sales controller, sales detail controller, and stock controller contains the same methods with different parameters and database table

HOME CONTROLLER

S. No	Methods	Description
1	Index()	Database connected with the help of SQL query. Get details of the products which are out of stocks
2	Error()	Displays the error details if occur

Table 2: Home controller table

REPORT CONTROLLER

S. No	Methods	Description
1	OutOfStockReport()	Database connected with the help of SQL query. Get details of the products which are out of stocks
2	StockListReport(string search)	Shows the available details of item and quantity while connecting with the database. Also, it searches according to the product name

Table 3: Report controller table

7 Description of Properties

S.NO	Properties Name	Data Type	Description
1	Id { get; set; }	Int	ID shows the primary key of Category
2	CategoryName { get; set; }	String	It shows the description of the category
3	FirstName { get; set; }	String	It shows first name of the customer
4	LastName { get; set; }	String	It shows last name of the customer
5	Contact { get; set; }	String	It shows the contact number of customer
6	Email { get; set; }	String	It shows email of the customer
7	Address { get; set; }	String	It shows address of the customer
8	CustomerCode { get; set; }	Int	It shows code of the customer
9	CustomerType { get; set; }	String	It shows type of the customer
10	RequestId { get; set; }	String	It is the primary key of the request
11	PurchaseId { get; set; }	Int	It is the primary key of the Purchase
12	PurchasesDetails { get; set; }	Virtual	It is the foreign key of the table purchase
13	CategoryId { get; set; }	Int	It is the primary key of the Category
14	Category { get; set; }	Virtual	It is the foreign key of the table purchase
15	ItemName { get; set; }	String	It shows the name of the items
16	Description { get; set; }	String	It describes about the items
17	ManufacturedDate { get; set; }	DateTime	It shows the manufacture date of the item

18	ExpiryDate { get; set; }	DateTime	It shows the manufacture date of the expiry date
19	PurchasesDate { get; set; }	DateTime	It shows the manufacture date of the purchased item
20	BillNumber { get; set; }	Int	It shows bill number of the purchased item
21	VendorName { get; set; }	String	It shows the vendor name
22	Purchase { get; set;}	Virtual	It is the foreign key of the purchase table
23	SalesDate { get; set; }	DateTime	It shows the sales date of the item
24	CustomerId { get; set; }	Int	It is the primary key of the customer table
25	Customer { get; set; }	Virtual	It is the foreign key of the customer table
26	BillNumber { get; set; }	Int	It shows the bill number
27	SalesId { get; set; }	Int	It is the primary key of sales table
28	Sales { get; set; }	Virtual	It is the foreign key of the sales table
29	Amount { get; set; }	Int	It shows the amount of the item
30	Quantity { get; set; }	Int	It shows the quantity of the item
31	ItemId { get; set; }	Int	It is the primary key of the item table
32	Item { get; set; }	Virtual	It is the foreign key of the item table
33	ProductId { get; set; }	Int	It is the primary key of the Product
34	ProductName { get; set; }	String	It shows the name of the products

Table 4: Properties Table 1

S.No	Properties Name	Description
1	DbSet<Coursework.Models.Category> Category { get; set; }	Its brings the data from the form/Controller and sets it to the Category Model
2	DbSet<Coursework.Models.Item> Item { get; set; }	Its brings the data from the form/Controller and sets it to the Item Model
3	DbSet<Coursework.Models.Customer> Customer { get; set; }	Its brings the data from the form/Controller and sets it to the Customer Model
4	DbSet<Coursework.Models.Purchase> Purchase { get; set; }	Its brings the data from the form/Controller and sets it to the Purchase Model
5	DbSet<Coursework.Models.PurchasesDetails> PurchasesDetails { get; set; }	Its brings the data from the form/Controller and sets it to the PurchasesDetails Model
6	DbSet<Coursework.Models.Sales> Sales { get; set; }	Its brings the data from the form/Controller and sets it to the Sales Model
7	DbSet<Coursework.Models.SalesDetails> SalesDetails { get; set; }	Its brings the data from the form/Controller and sets it to the SalesDetails Model
8	DbSet<Coursework.Models.Stock> Stock { get; set; }	Its brings the data from the form/Controller and sets it to the Stock Model

Table 5: Properties Table 2

8 Testing

Test Case	1
Goal	To test whether the customer gets logged in
Required Output	The customer should be able to log in to the system
Actual Output	Logged in to the system without any errors
Test Result	Successful

Table 6 - To test whether the customer gets logged in

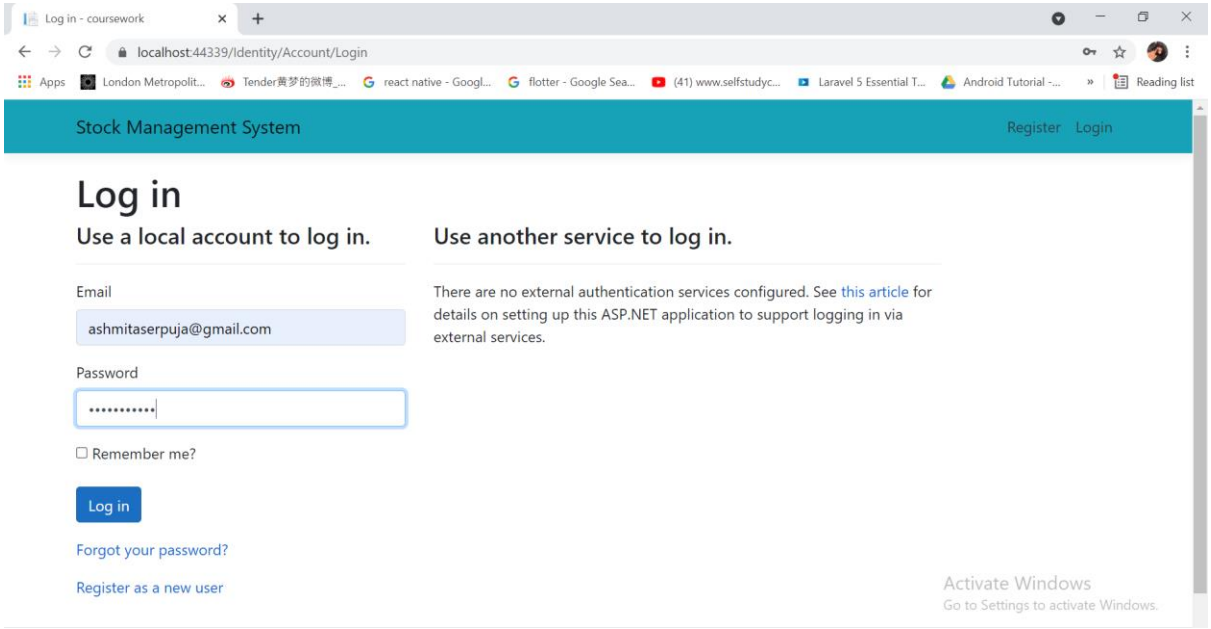

Figure 13 - Login page

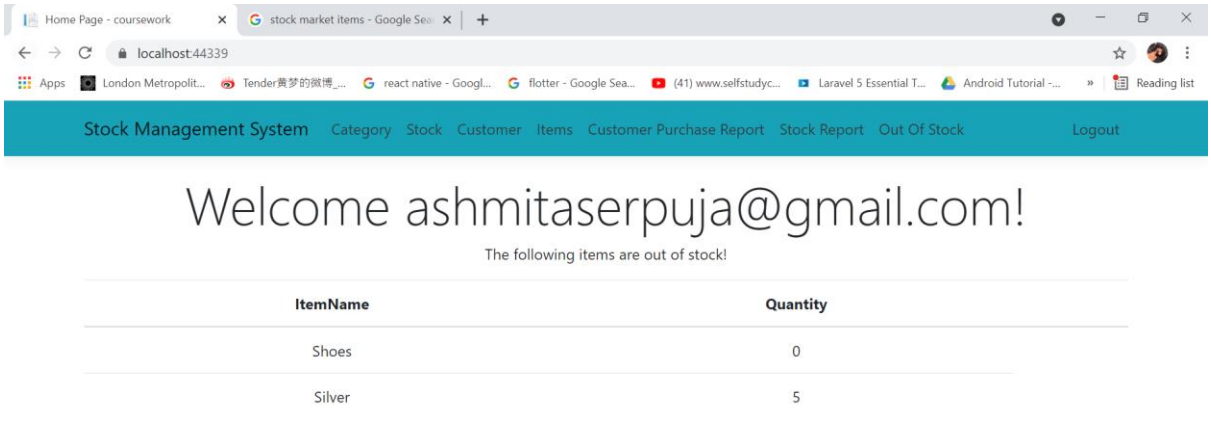

Figure 14 - Login Successful

Test Case	2
Goal	To test whether the system registers new customers
Required Output	The system should register new customers
Actual Output	New Customer can register to the system
Test Result	Successful

Table 7 – To test whether the system registers new customers

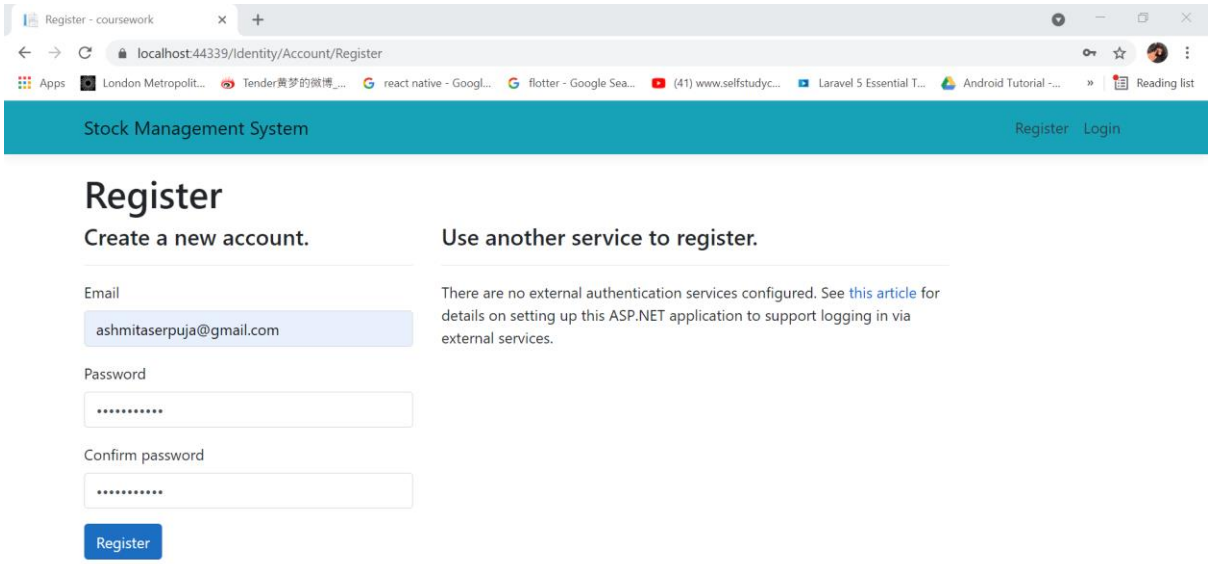

Figure 15 - Registration

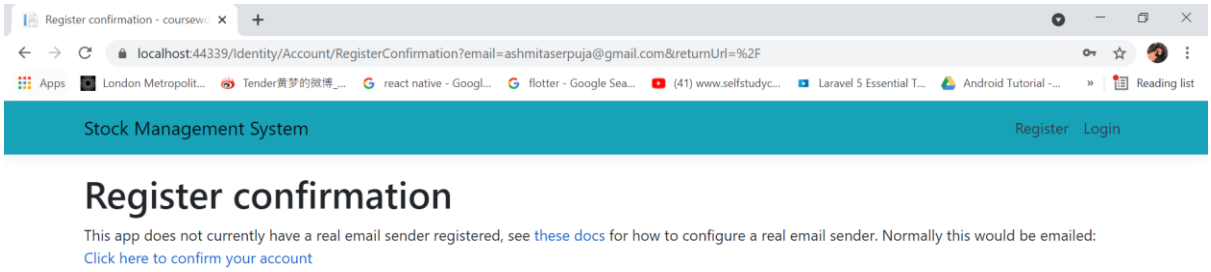

Figure 16 - Registration Conformation

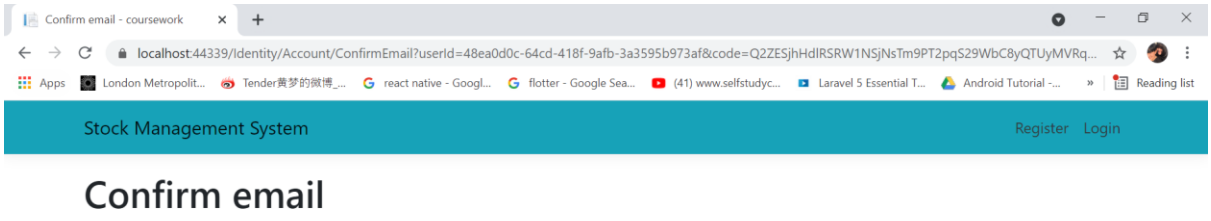

Figure 17 - Confirm email

Test Case	3
Goal	To test whether the customer can search
Required Output	The search function should be functional
Actual Output	Customers can search for the product according to its name.
Test Result	Successful

Table 8 - To test whether the customer can search

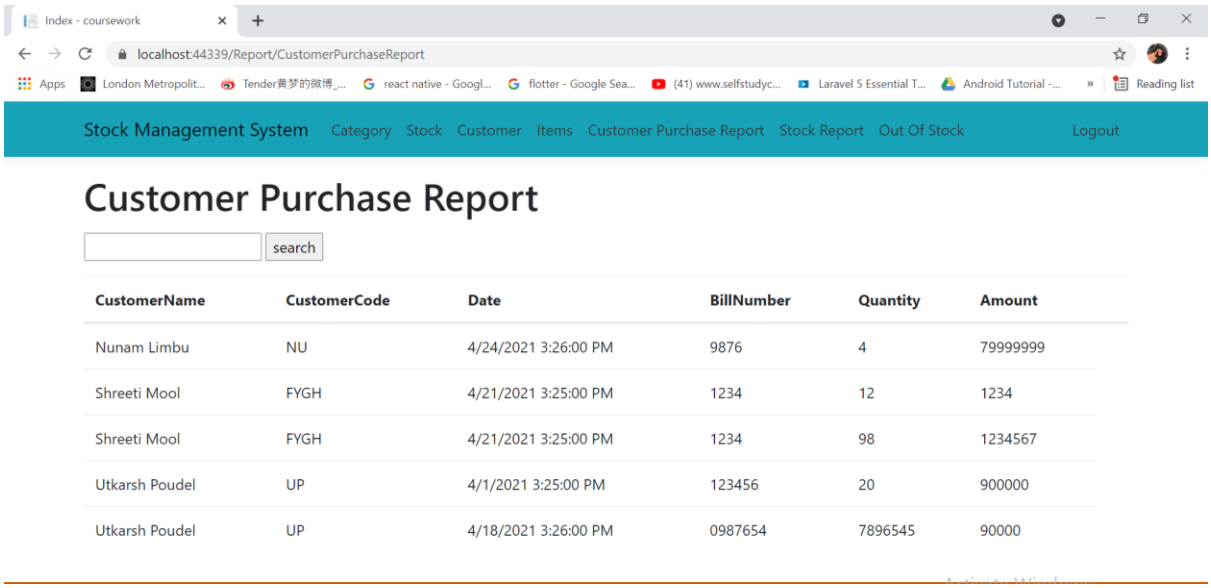

Figure 18 - Customer Purchases Report

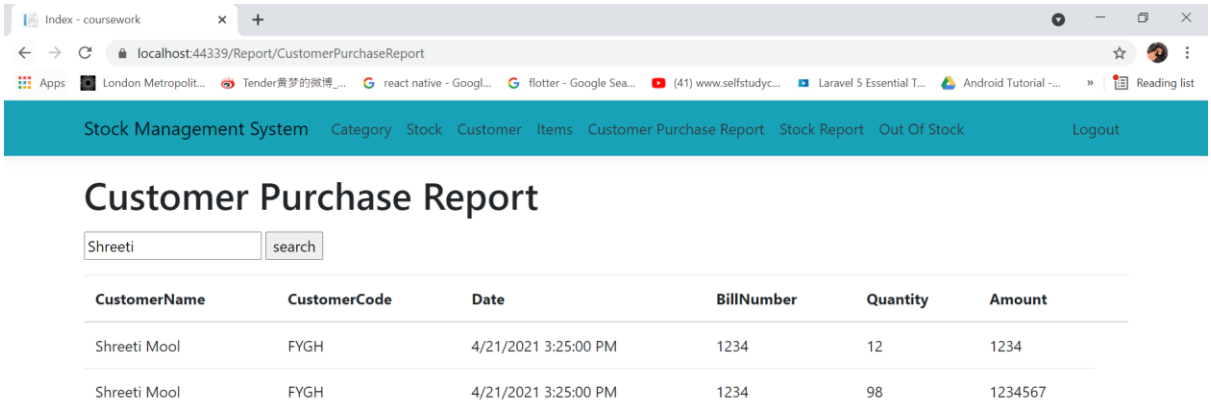

Figure 19 - Search Result

Test Case	4
Goal	To show the list of available stocks
Required Output	The system should be able to show the list of the available stocks
Actual Output	The list of available items are shown
Test Result	Successful

Table 9 - To show the list of available stocks

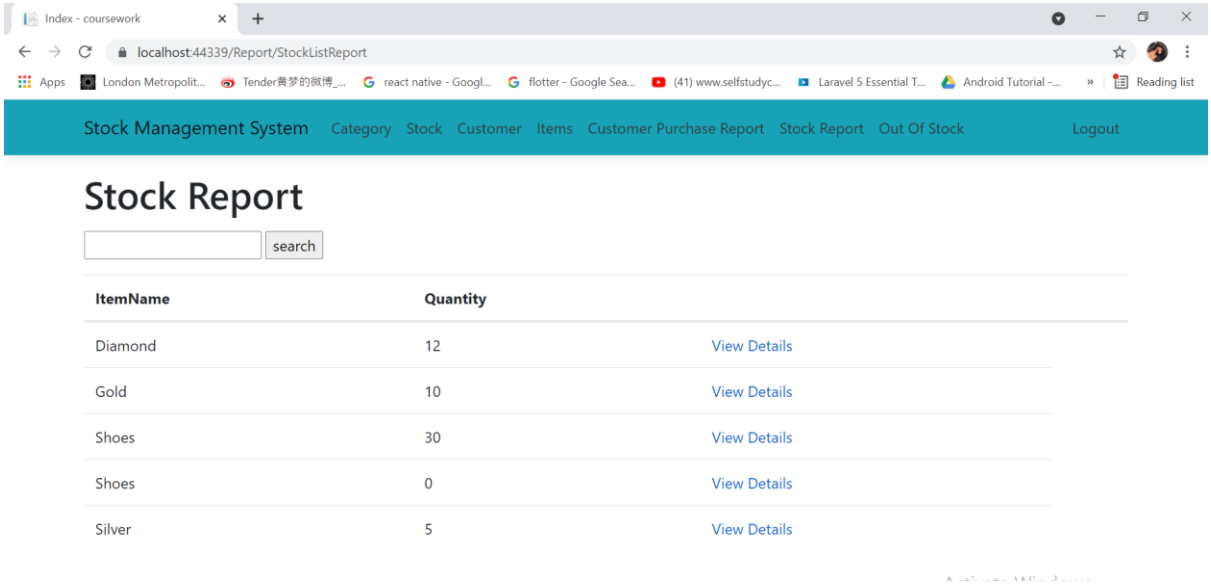

Figure 20 - Stock Report

Test Case	5
Goal	To test whether the customer can add, delete, edit, create data
Required Output	The system should be able to add, delete, edit, create data
Actual Output	Data is added and edit. The details of the data is shown and it is also deleted
Test Result	Sucessful

Table 10 - To test whether the customer can add, delete, edit, create data

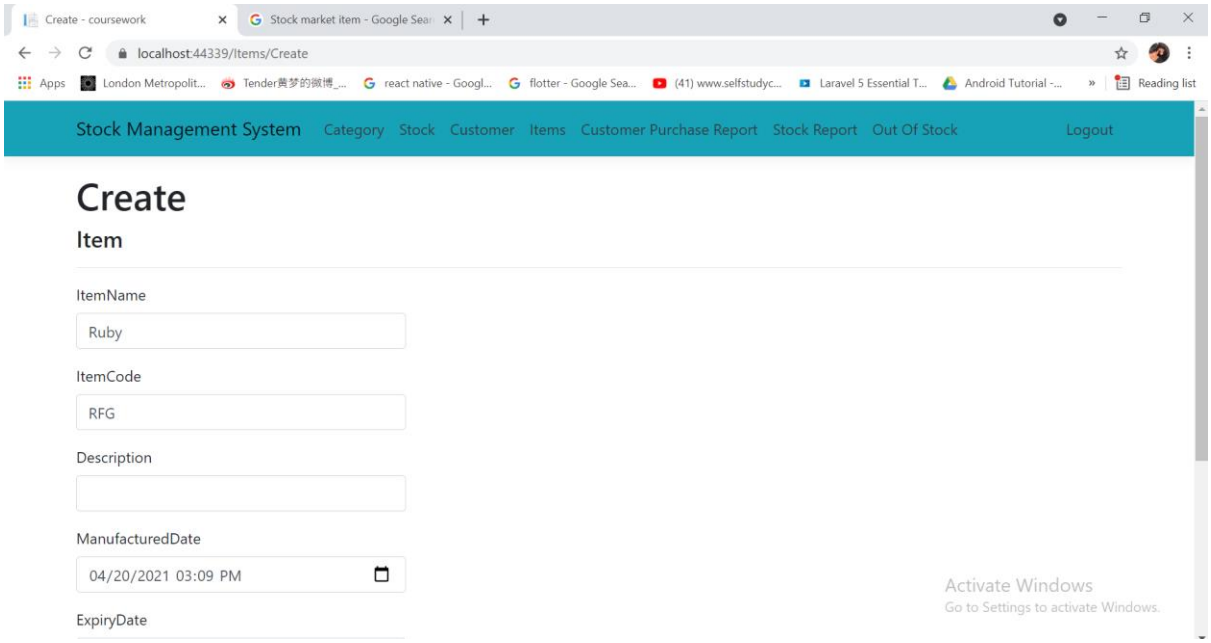

Figure 21 - Create

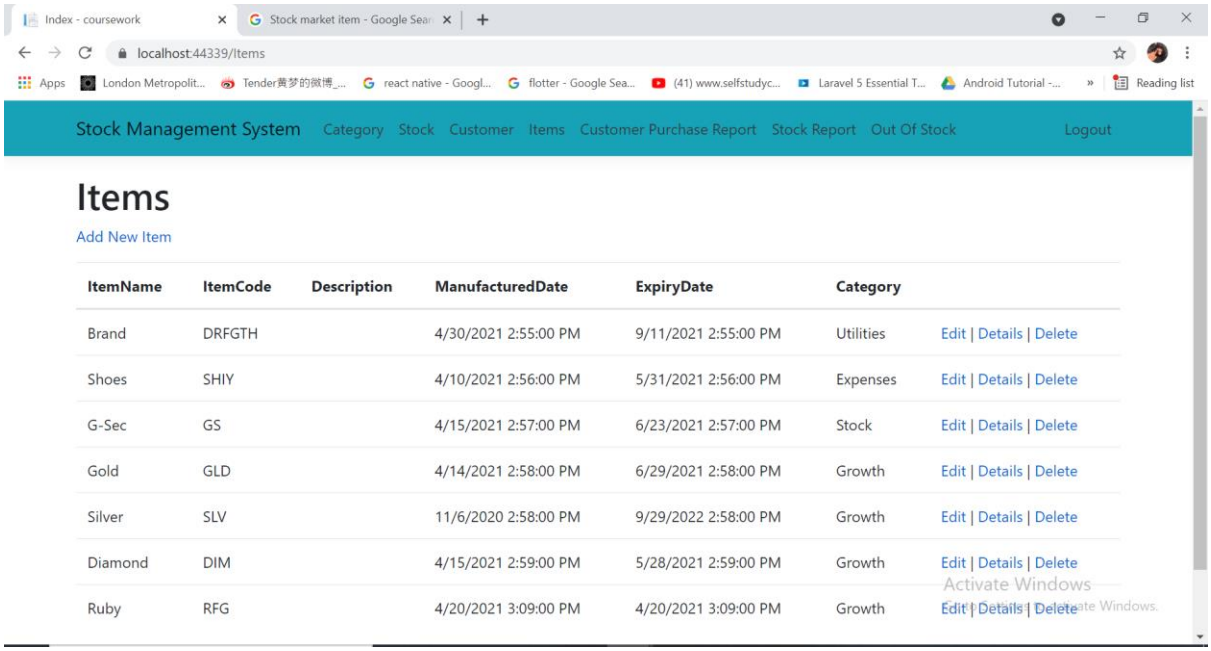

Figure 22 – Item List

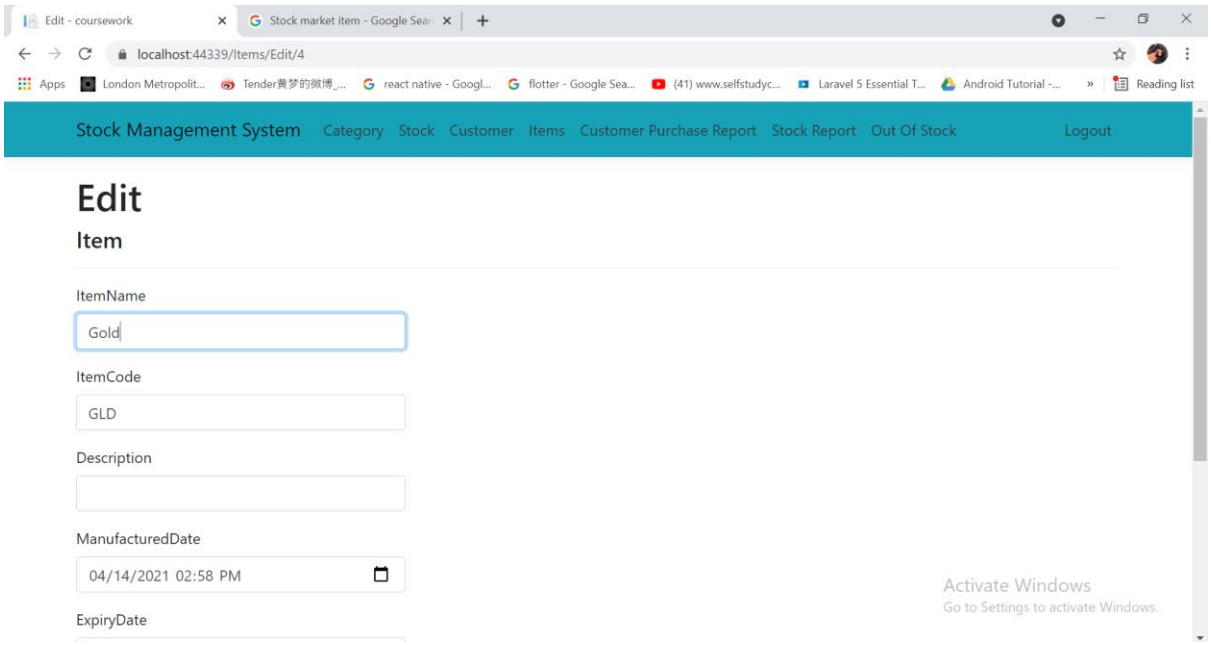

Figure 23 - Item Edit

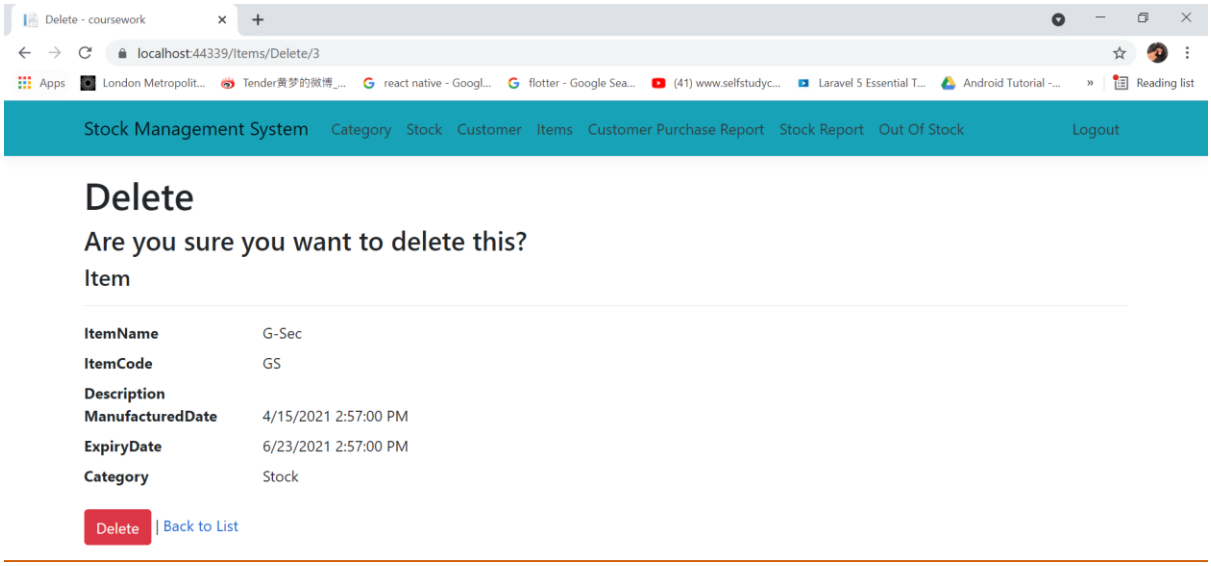

Figure 24 - Item Delete

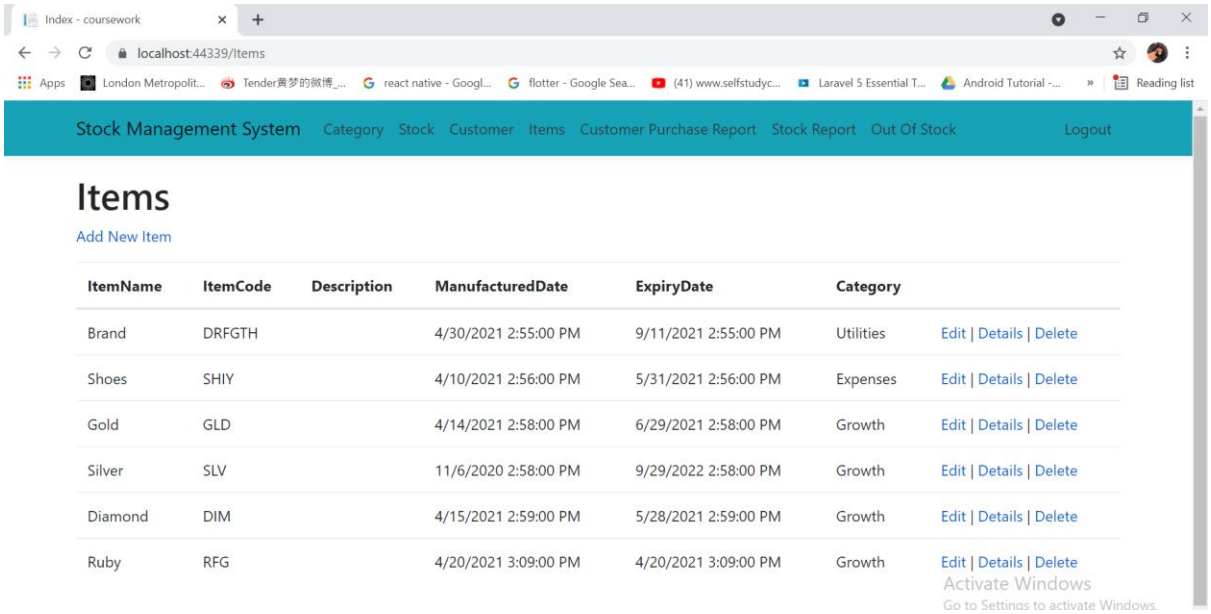

Figure 25 - Sucessful Message

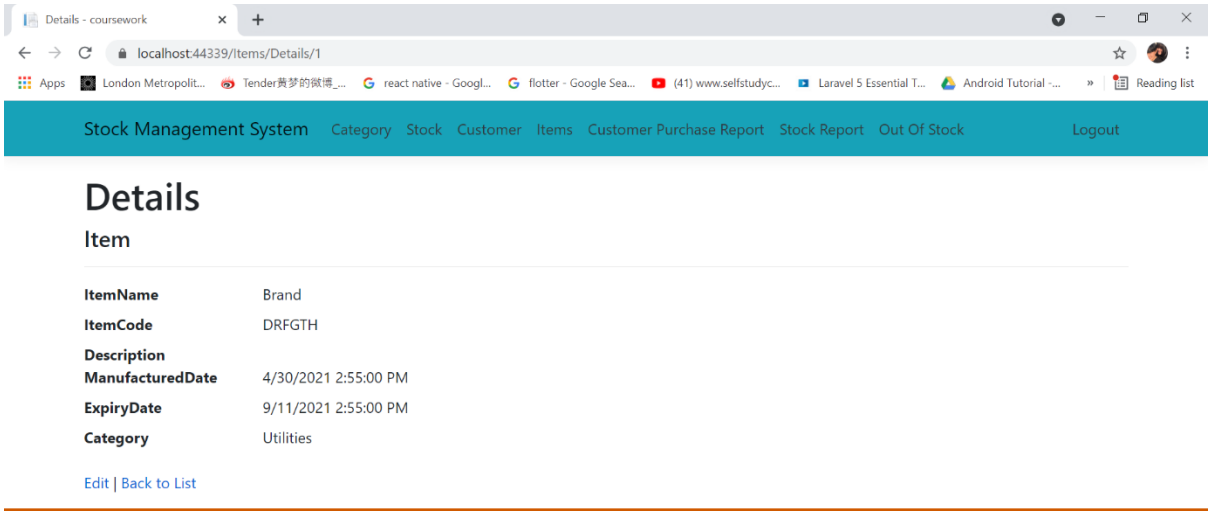

Figure 26 - Items Details

Test Case	6
Goal	To test whether the value is sorted by item name
Required Output	The system should sort the listed value by name
Actual Output	Value is sorted by item name and displayed in list
Test Result	Successful

Table 11 - To test whether the value is sorted by item name

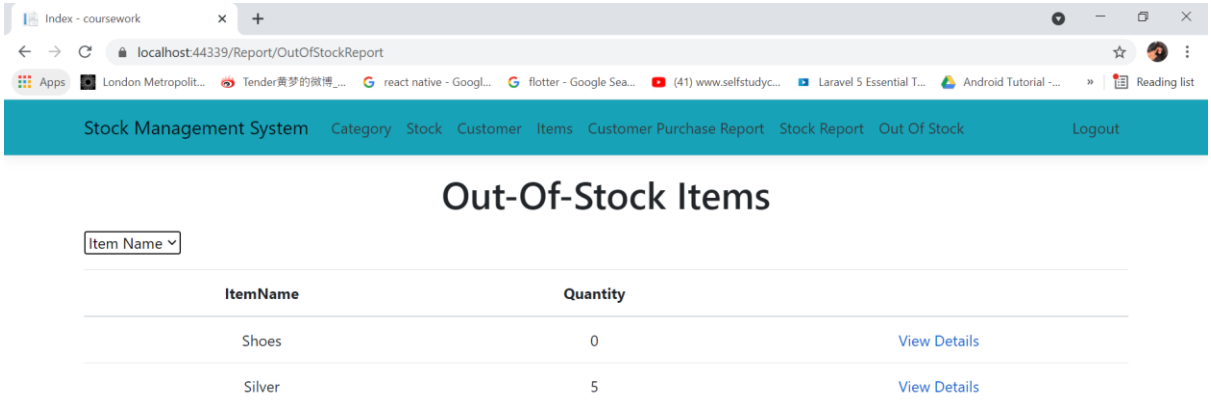

Figure 27 - Out of Stock(Sorting)

Test Case	7
Goal	To test whether it is wrong credentials let you log in
Required Output	The system send error message
Actual Output	Error message is displayed
Test Result	Successful

Table 12 - To test whether it is wrong credentials let you log in

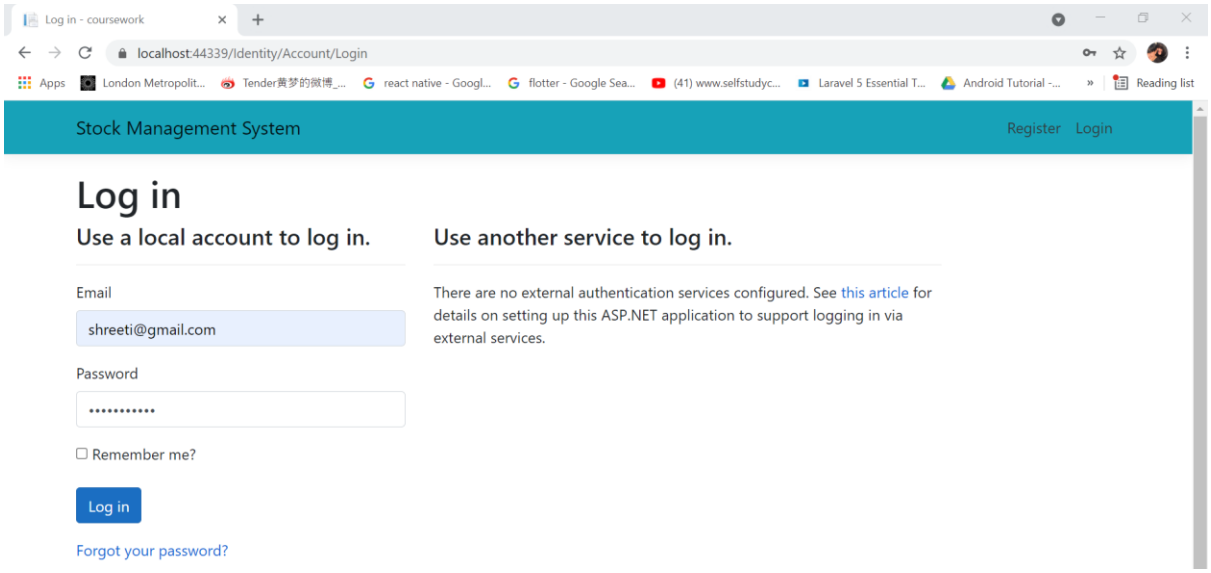

Figure 28 - Invalid Data Login

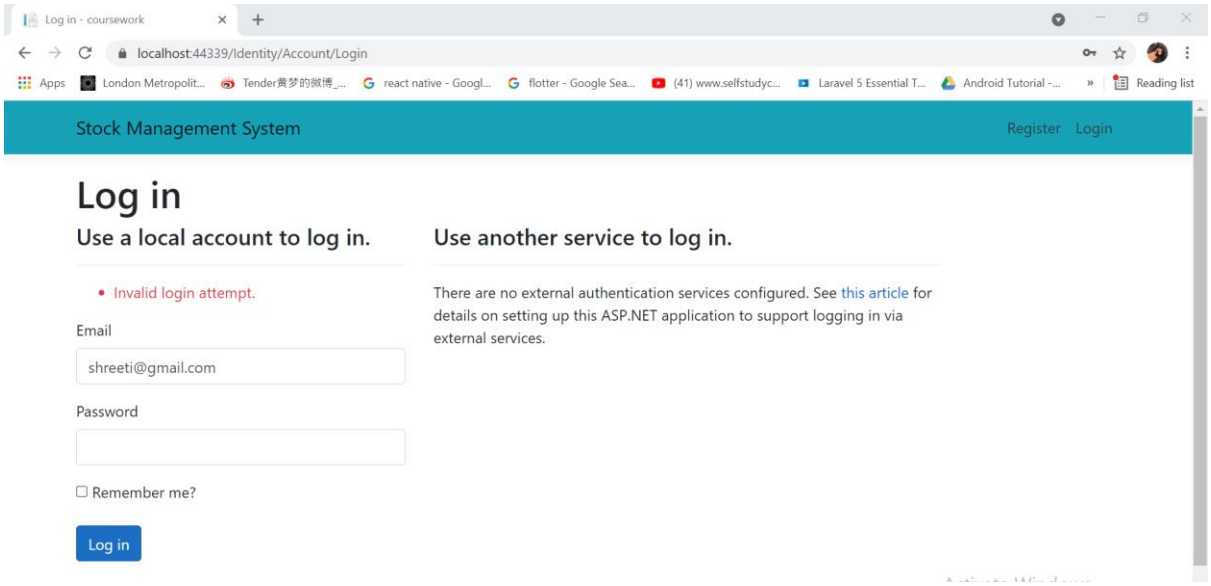

Figure 29 - Error Message

9 Group Members Reflection

9.1 Salina Budhathoki

Our main objective was to work on the platform that was specified in the coursework, and our main goal was to stick to that idea and use the logic provided to complete the project correctly. We had done a project with C# and ASP.net before, but this time it had to be linked with an MSSQL server to a much larger database, which was different from the previous time. It was not an easy task to design about the customer, purchase, sales order, and monitor sales history, but with a lot of shared knowledge, we were able to complete the work on time. The rest of the reflection is in Group Reflection

9.2. Shreeti Mool

This coursework was much more difficult than I thought it was. The main purpose of this coursework was to make an application on Stock Management System. My problem was that I needed to think about how the code would work because the work was challenging for me. The problem was how to improve the efficiency of the stock management store, which includes a sale and purchasing system.

The feature that I liked most about this program is that users can search according to the name of the product which can save the time user. Implementing this feature was a bit difficult and time-consuming. It was a difficult situation while writing the SQL query and making it functional, I overcame this situation by discussing the errors with my tutor and by searching on the internet. The rest of the reflection is in Group Reflection

9.3. Ashmita Serpuja

The main objective of this work was to develop our skills in C# and get familiar with Visual Studio Code, along with working in a group. We all were familiar with C# and Visual studio as our previous project was also created in it. But this time we had to use the MVC framework along with linking it to the MSSQL server. The coursework was harder than expected in the beginning. The rest of the reflection is in Group Reflection

9.4. Utkarsh Poudel

The ultimate goal of this workshop is to create a stock management system. The project will be completed in a group using C# and the.NET framework as the primary programming languages. The NET Framework is a C#-based framework that aids in concept comprehension. In addition, the system was built using the MVC (Model View Controller) system architecture, with the system focusing primarily on the three-component models, view, and controller. With this, I can confidently state that I used this energizing stage to create a work area application, overcoming all of the challenges and roadblocks along the way. To successfully carry out the program, I conducted numerous research and studies.

10. Conclusion

To conclude, Stock Management System is a simple desktop-based application. It has every basic item which is used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a manual to know the daily sales and purchase details. We acquired knowledge of C sharp language, Visual Studio, and the NET. Framework after the completion of this project will help us a lot in the future.

11. References

Techopedia, 2021. *What is Software Architecture? - Definition from Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/24596/software-architecture
[Accessed 29 April 2021].

tutorialspoint, 2021. *MVC Framework - Introduction - Tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm#:~:text=MVC%20Framework%20-%20Introduction.%20The%20Model-View-Controller%20%28MVC%29%20is,to%20handle%20specific%20development%20aspects%20of%20an%20application.
[Accessed 30 April 2021].

12. Appendix

12.2. Software Architecture

Model: The data-related logic that the user works with is represented by the Model component. This may be the data being transferred between the View and Controller components or some other data relevant to business logic. A Customer object, for example, can extract customer information from a database, manipulate it, and then update or make the data back to the database. (tutorialspoint, 2021)

View: All of the application's UI logic is handled by the View component. The Customer view, for example, would contain all of the UI components that the final user interacts with, such as text boxes, dropdowns, and so on. (tutorialspoint, 2021)

Controller: Controllers act as an interface between the Model and View components, processing all business logic and incoming requests, manipulating data with the Model, and interacting with Views to make the final output. The Customer controller, for example, will manage all interactions and inputs from the Customer View and use the Customer Model to update the database. The Customer Data will be viewed using the same controller. (tutorialspoint, 2021)

12.3. Group Reflection

Salina Budhathoki

C# with asp.net was not easy to use, and I had a lot of errors and trials to overcome the fact that it would not be completed on time. But I was able to do it with the help provided by my colleagues or friends and teachers it was a bit easy and was able to solve the problem. In this program, I and my team members put in a lot of effort to achieve the best results possible, and we were able to complete it on time.

In the current situation, we were given a huge task to complete in a limited amount of time, but we did our best to complete it, despite some issues. I appreciate our teacher and module leaders recognizing the seriousness of the situation so that we could give you the best project possible.

Shreeti Mool

I was able to understand the situation a lot better after a lot of hard work and searching through various resources available on the internet, but the code was difficult to deal with and bear with because it was a huge opportunity for me and my group members to display our skills and demonstrate them. The thing that I had a lot of trouble with was doing the work that the coursework required, so I looked up various options on the internet, such as journals, blogs, and articles about it. We came to a certain concept through a lot of debate with the team members, which we have shown in the coursework.

Running the previous problem of the coursework was easy, but this time it was large and had to be completed in the time given by the coursework deadline. The main objective was to complete the documentation and coding with the assistance of the teammates. We were able to solve the problem on time with a lot of hard work and commitment.

Ashmita Serpuja

While developing the system, many errors occurred which took lots of research to be solved. Our tutorial teacher Sushil sir's video, fortunately, helps a lot in creating this system. My teammates also helped a lot in creating this system. We all have given our best effort in meeting the requirement of the coursework. Due to this pandemic increasing in our country and the lockdown situation going on we were unable to meet

for group coursework but thanks to the internet we were able to virtually meet and complete the system.

At a point, everyone including me lost the hope of us being able to complete but in the end, we motivated and pushed each other into completing this system. Thank our teacher even at the last moment he was ready to help us and clear our confusion regarding the coursework. Without my teammates and Sushil sirs help we were able to complete this coursework on time and I am thankful for that.

Utkarsh Poudel

I studied several C # tutorials. Additionally, the teachers' lectures, tutorial slides, and videos aided in the completion of the task. The group members and teacher were unable to meet physically due to the second variant of covid-19 and lockdown and had to rely on a virtual meeting. Despite the difficulties, our instructor was able to convey all of the necessary information to complete the coursework. Without the tutors' constant encouragement and guidance, we would not have been able to complete the coursework. Overall, this coursework aided in the development of not only the language C# ASP.Net platform but also the social skills required for working in a community. When working in a group, many issues can arise, such as time management among team members and member opposition. Open communication among members and during the development process helped to overcome such obstacles. The method was both challenging and instructive.