

Acknowledgement

We would like to express our deepest gratitude to all those who provided us with the opportunity to complete this coursework in a systematic order. We would like to pay our sincere acknowledgement to our module leader, Mr. Sabin Shrestha, for the valuable guidance and support throughout the coursework which shaped the present work as it shows. It was a great honour to work under his guidance. We would like to expand our gratitude to all those who have directly or indirectly helped us in completing this coursework.

This coursework could not be able to complete without the hard work and great teamwork from our group members. In addition, we would like to acknowledge our classmates who provide suggestions and valuable comment which gave us the inspiration to improve our coursework. Moreover, we would also like to extend our gratitude to RTE department for providing us the opportunity to embark on this project and all the facility that was required. Lastly, we would like to express our deepest gratitude to our family for their continued love, sacrifices, support and encouragement during hard times.

Abstract

This report contains all the detailed explanation about the related tasks that will be carried out during the development of this project. This coursework is performed in group of 4 which is mainly designed to assess student's practical problem-solving skills and critical thinking and evaluation on the design and development of database systems.

In this project, introduction, solutions of problems and other related factors are discussed and many researches have been done for developing the best system. The system, named as "Beans Stock Management", is a web-based application which is developed using ASP.NET Web Forms, SQL Server Database for back end. The system has all the features that a stock management needed which are discussed in the report below. The report part includes introduction, user manual, solution designs, algorithms, flowcharts, test cases and personal reflection of each team members. To sum up, it requires to analyse, design and implement a web-based database application to support the operations of the company, specifically for the stock management.

Table of Contents

1. Introduction.....	1
2. User Manual	3
3. Solution design of system	10
3.1 System Architecture	10
3.2 ER-diagram.....	11
3.3 Class Diagram.....	12
3.4 Use case diagram.....	12
3.5 Algorithm	14
3.6 Flowchart.....	17
4. Description of Classes Purpose and Methods	23
4.1 Purpose of classes	23
4.2 Description of methods	25
5. Testing.....	29
6. Reflection.....	51
6.1 Group Experience.....	51
6.2 Personal Experience	51

Table of Figure

Figure 1: User Manual of Login page.....	3
Figure 2: User manual of Admin Dashboard.....	4
Figure 3: User Manual of User Home Page.....	4
Figure 4: User Manual of Category Page.....	5
Figure 5: User Manual of Item Page.....	5
Figure 6: User Manual of Customer Page.....	6
Figure 7: User Manual of Stock Page.....	6
Figure 8: User Manual of Out-of-Stock Page.....	7
Figure 9: User Manual of Manage Stock page.....	7
Figure 10: User Manual of Purchase Details Page.....	8
Figure 11: User Manual of Sales Page.....	8
Figure 12: User Manual of Other Details Page.....	9
Figure 13: User Manual of Profile Page.....	9
Figure 14: System Architecture of Beans Stock Management.....	10
Figure 15: ER diagram of Beans Stock Management.....	11
Figure 16: Class Diagram of Beans Stock Management.....	12
Figure 17: Use Case Diagram of Beans Stock Management.....	13
Figure 18: Flowchart of add category.....	17
Figure 19: Flowchart of add item.....	18
Figure 20: Flowchart of add customer details.....	19
Figure 21: Flowchart of Login.....	20
Figure 22: Flowchart of Add User Details.....	21
Figure 23: Flowchart of Delete Item.....	22
Figure 24: Display Login page when user run the program.....	29
Figure 25: User provide login details.....	30
Figure 26: Display home page after successful login.....	31
Figure 27: Provide admin login details.....	32
Figure 28: Display admin dashboard after successful login.....	32
Figure 29: Provide category details.....	33
Figure 30: Display successful message.....	34
Figure 31: Diary Product category is added to the system.....	34
Figure 32: Provide item details.....	35
Figure 33: Display successful message.....	35
Figure 34: New Item Apple successfully added to the system.....	36
Figure 35: Provide customer details.....	36
Figure 36: Display successful message.....	37
Figure 37: New customer added to the system successfully.....	37
Figure 38: Display item details of particular category which are available in the stock.....	38
Figure 39: Display item details of selected item name which are available in the stock.....	39
Figure 40: Display the purchase details of the customer who had purchased by last 31 days.....	40
Figure 41: Display message to the user when the item is running out of stock at the time of login.....	41
Figure 42: Before removing the items that are stocked for a long time in the system.....	42
Figure 43: After removing the items that are stocked for a long time in the system.....	42

Figure 44: Display all the items that are currently out of stock.....	43
Figure 45: Display list of customers who never bought any item in the last 31 days	44
Figure 46: Display list of items that are not sold in the last 31 days.....	45
Figure 47: Provide customer username and password details	46
Figure 48: Password changed successfully	46
Figure 49: Before changing password	46
Figure 50: After changing password	47
Figure 51: Admin provide user details.....	47
Figure 52: User successfully added to the system.....	48
Figure 53: Before editing user details	49
Figure 54: While editing user details	49
Figure 55: After editing user details	49
Figure 56: Before deleting user.....	50
Figure 57: After user deletion.....	50

Table of Tables

Table 1: Table of purpose of classes	24
Table 2: Method description of Category class	25
Table 3: Method description of Customer class	25
Table 4: Method description of Details class.....	25
Table 5: Method description of Item class.....	26
Table 6: Method description of Login class	26
Table 7: Method description of Manage Stock class	26
Table 8: Method description of Out of Stock class	27
Table 9: Method description of Profile class	27
Table 10: Method description of Sales class.....	27
Table 11: Method description of User Profile class	28
Table 12: Method description of Purchase Details class.....	28
Table 13: Method description of Stock Page class	28
Table 14: Test Case 1	29
Table 15: Test Case 2.....	30
Table 16: Test Case 3.....	31
Table 17: Test Case 4.....	33
Table 18: Test Case 5.....	34
Table 19: Test Case 6.....	36
Table 20: Test Case 7.....	37
Table 21: Test Case 8.....	38
Table 22: Test Case 9.....	39
Table 23: Test Case 10.....	40
Table 24: Test Case 11.....	41
Table 25: Test Case 12.....	43
Table 26: Test Case 13.....	44
Table 27: Test Case 14.....	44
Table 28: Test Case 15.....	45
Table 29: Test Case 16.....	47
Table 30: Test Case 17.....	48
Table 31: Test Case 18.....	50

1. Introduction

The coursework given to us is a group coursework where we are required to form in group of 4 or 5 members in each. According to this coursework we are required to develop a web application using Visual Studio, C#, ASP.NET and Microsoft SQL. This coursework has divided into two parts: the development and the documentation. In the development part we are required to build a web-based application that should support the operations of the stock management company. As said in the question itself, the system is used to manage the detail of items, the customer's details and keep records of selling items to the customers. So, we are required to design database tables with proper attributes and we must implement 14 functionalities as given in the question.

The stock management system has features such as:

- User can add category.
- User can enter an item's details (item name, description, price, purchase date, category id) into the system.
- User can enter or select name of the category and see all items available in stocked.
- User can enter or select item name and display items detail with available quantity on stock.
- User can sale an item to the customer.
- User can view purchases detail of items within 31 days of a particular customer.
- System prompt message to the user at the time of login when the item is running out of stock by quantity 10.
- User can remove all items from stock that are stored there for a long period of time.
- User can view a list of all customers who have not bought any item in the last 31 days.
- User can also view a list of all items in the stock where no item has been sold in the last 31 days.
- User can change their own passwords while admin can change any detail of normal User and can also delete the user.

We are required to develop our web application using Visual Studio 2013 or, above. Visual Studio is an integrated development environment (IDE) from Microsoft which is used to develop various computer programs, websites, web apps, web services and also mobile applications. for the front-end part, we have to use C# programming language and for backend we need to use Microsoft SQL Management. On side by side, we must also provide appropriate report of this project. The report must contain all the necessary diagrams such as (flowcharts, use case diagrams, er diagrams) along with description. We must also include all the testing cases of the project in this report.

2. User Manual

- Login Page:

When you first run the program, it will display the login page as given in the below figure. Here if admin provide the admin details i.e., “admin” as username and “admin” as password, it will redirect admin to the admin dashboard. Whereas if it provides user details i.e., username and password, then it will redirect user to the user dashboard.

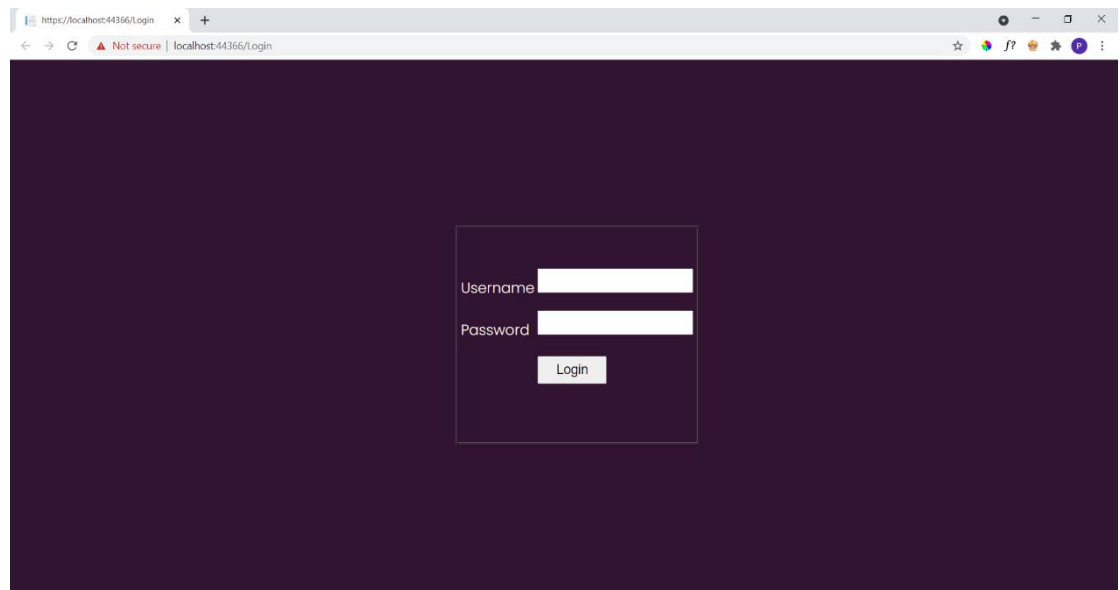


Figure 1: User Manual of Login page

- Admin dashboard:

When admin provides the correct credentials then admin dashboard is shown which is as shown in the below figure. Here admin can add new user to the system by providing username, full name, contact and password. Moreover, admin can also edit and delete the user details.

	Username	Fullname	Contact	Password
Edit Delete	sky	Sairat Gurung	980665874	123sky

Figure 2: User manual of Admin Dashboard

- User home page:

When user provides the correct credentials then they are redirected to the user dashboard which is illustrated as shown below. Here user can navigate to different web pages through navigation bar which is located in the upper part of the dashboard.

Figure 3: User Manual of User Home Page

- Category web page:

Here user can add the category details by providing category id, category name and description as shown in the figure below.

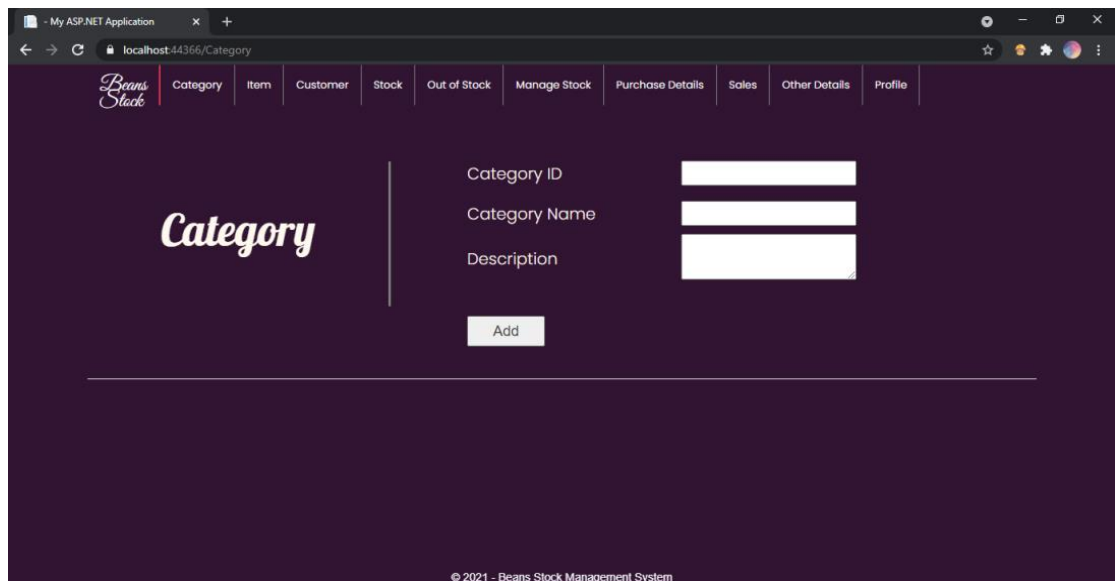


Figure 4: User Manual of Category Page

- Item web page:

Here user can add the item details by providing item id, item name, description, price of item, purchase date of item, category name, manufacture date of item and expiry date of item as shown in the figure below.

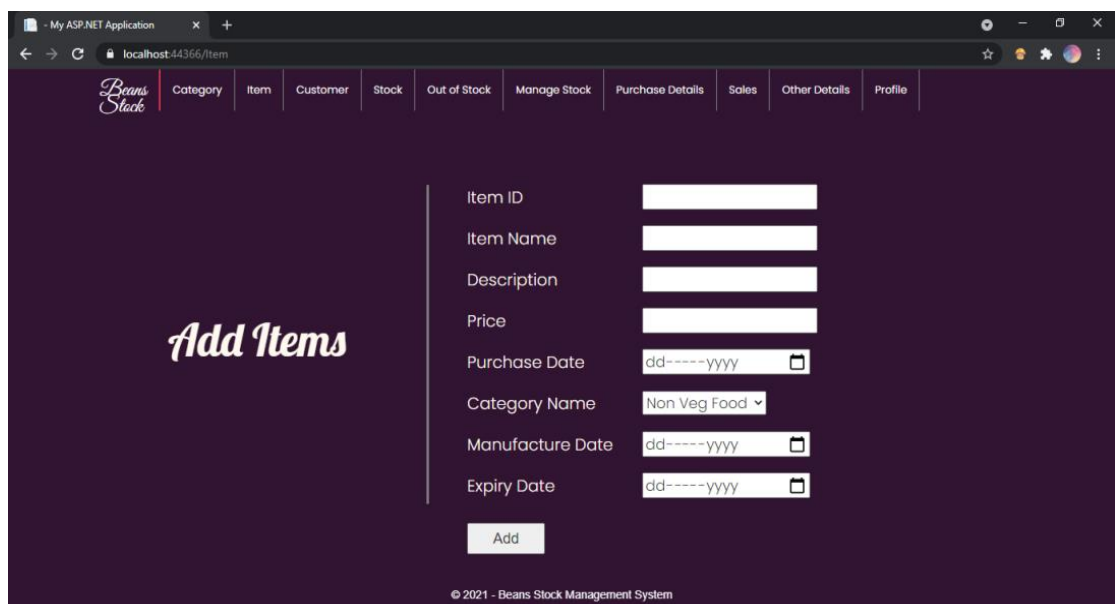


Figure 5: User Manual of Item Page

- Customer web page:

Here user can add the customer details by providing username, customer name, address, contact number, member type of customer, email of customer with its password as shown in the figure below.

Figure 6: User Manual of Customer Page

- Stock web page:

Here user can view all the item details available in the stock by providing category name at the first table as shown in the below figure whereas user can also view item details available in the stock by providing the item name.

ItemName	description	price	purchaseDate	manufactureDate	expiryDate	quantity	stockDate
Frozen MoMo	Made of Chicken	100	2021/04/29	2021/04/29	2021/05/02	50	2021/04/13
Apple	Marpha Apple	50	2021/05/01	2021/04/25	2021/05/20	100	2021/05/01

ItemName	description	price	purchaseDate	manufactureDate	expiryDate	quantity
Frozen MoMo	Made of Chicken	100	2021/04/29	29-Apr-21 12:00:00 AM	2021/05/02	50

Figure 7: User Manual of Stock Page

- Out of Stock web page:

Here user can view the details of items that are out of stocks especially items less than 10 in quantity as shown in the below figure. Item details are sorted by item name in ascending order and then displayed. Moreover, item details are also displayed by sorting the data by item stock quantity in high to low order and also by item stocked date in descending order.

Sorted by item name

Item ID	Item Name	Description	Price	Purchase Date	Quantity	Stock Date
3	Biscuit	Chocolate Biscuit	20	2021/04/30	5	2021/03/15
4	Noodles	Veg Noodles	15	2021/04/25	9	2021/04/25

Sorted by item stock quantity

Item ID	Item Name	Description	Price	Purchase Date	Quantity	Stock Date
4	Noodles	Veg Noodles	15	2021/04/25	9	2021/04/25
3	Biscuit	Chocolate Biscuit	20	2021/04/30	5	2021/03/15

Sorted by item stocked date

Item ID	Item Name	Description	Price	Purchase Date	Quantity	Stock Date
3	Biscuit	Chocolate Biscuit	20	2021/04/30	5	2021/03/15
4	Noodles	Veg Noodles	15	2021/04/25	9	2021/04/25

© 2021 - Beans Stock Management System

Figure 8: User Manual of Out-of-Stock Page

- Manage Stock web page:

Here the items that are on stock for a long time is displayed and user can delete the items from the stock as shown in the given figure.

Current Date: 2021-05-02

	Stock ID	Item ID	Item Name	Description	Price	Purchase Date	Category ID	Stock Date
Delete	3	3	Biscuit	Chocolate Biscuit	20	2021/04/30	3	2021/03/15

© 2021 - Beans Stock Management System

Figure 9: User Manual of Manage Stock page

- Purchase Details web page:

Here list of purchase details made within a month by the particular customer is displayed by providing the customer username.

Customer Name	Item Name	Description	Price	Purchase Date
Priyanka Gurung	Biscuit	Chocolate Biscuit	20	2021/04/15

© 2021 - Beans Stock Management System

Figure 10: User Manual of Purchase Details Page

- Sales web page:

Here user can make sales of items to the customer by providing the item code and quantity. And when the user enter the code, system displayed the item details i.e., item name, description, category name, purchase date, manufacture date, expiry date price and total price as shown in the figure below.

Item Name	Description	Category Name	Purchase Date	Manufacture Date	Expiry Date	Price	Total Price
Frozen MoMo	Made of Chicken	Non Veg Food	2021/04/29	2021/04/29	2021/05/02	100	100

© 2021 - Beans Stock Management System

Figure 11: User Manual of Sales Page

- Other Details web page:

Here list of the customer who have not bought any items from the last 31 days are displayed along with the list of items are not sold in the last 31 days as well.

Current Date: 2021-05-02

List of all customers who have not bought any items in the last 31 days

Username	Customer Name	Address	Contact No	Email	Member Type	Purchase Date
rosna	Roshna Gurung	Lali Gurans Tole	9827116908	grg.rosna@gmail.com	New	2021/03/15
divya	Divya Gurung	Rambazar	9846012345	divya@gmail.com	Regular	2021/03/20

List of all items that hasn't been sold in the last 31 days

Item Name	Description	Price	Purchase Date	Manufacture Date	Expiry Date
Frozen MoMo	Made of Chicken	100	2021/04/29	2021/04/29	2021/05/02
Bread	White Bread	50	2021/04/29	2021/04/29	2021/05/01

© 2021 - Beans Stock Management System

Figure 12: User Manual of Other Details Page

- Profile web page:

Here user can edit their own details by providing the username they want to change and can change the existing password to new password.

Username:

Password:

New Password:

© 2021 - Beans Stock Management System

Figure 13: User Manual of Profile Page

3. Solution design of system

3.1 System Architecture

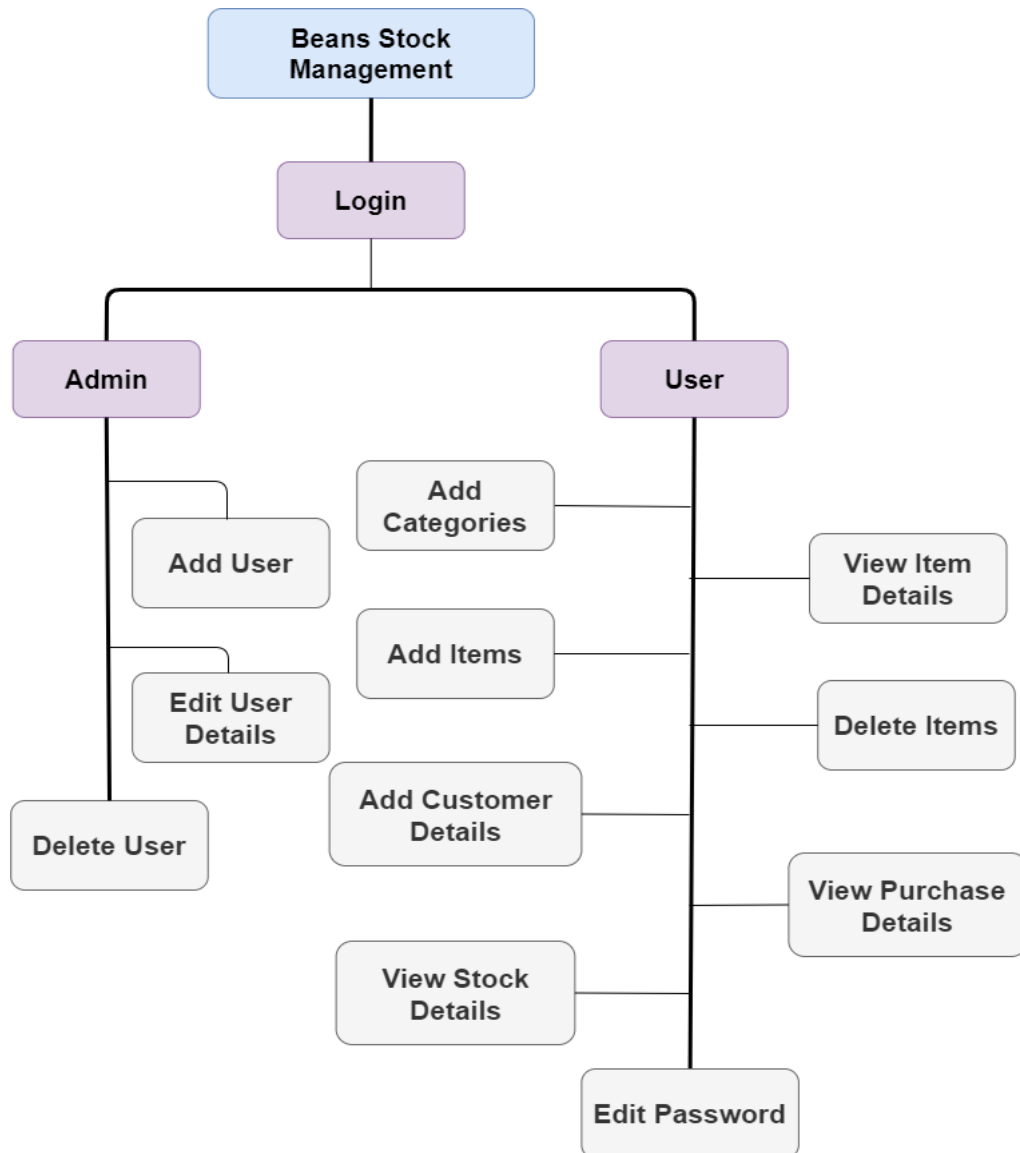


Figure 14: System Architecture of Beans Stock Management

In the above diagram, system architecture of “Stock Management System” is depicted. The system is divided into two actors i.e., admin and user. Admin and user both need to login first to get access to the system. Once admin is successfully logged in admin can add, edit and delete the user details. Whereas user can add categories, view item details, add items available in the stock, delete items that are out of stock, add customer details, view purchase details and stock details. Moreover, user can also edit their own password.

3.2 ER-diagram

An entity relationship diagram is a graphical representation of entities and their relations with each other. Entities are the objects that exists in the real-world whereas attributes are the properties of entity. The following figure is the ER diagram of our project “Stock Management System”.

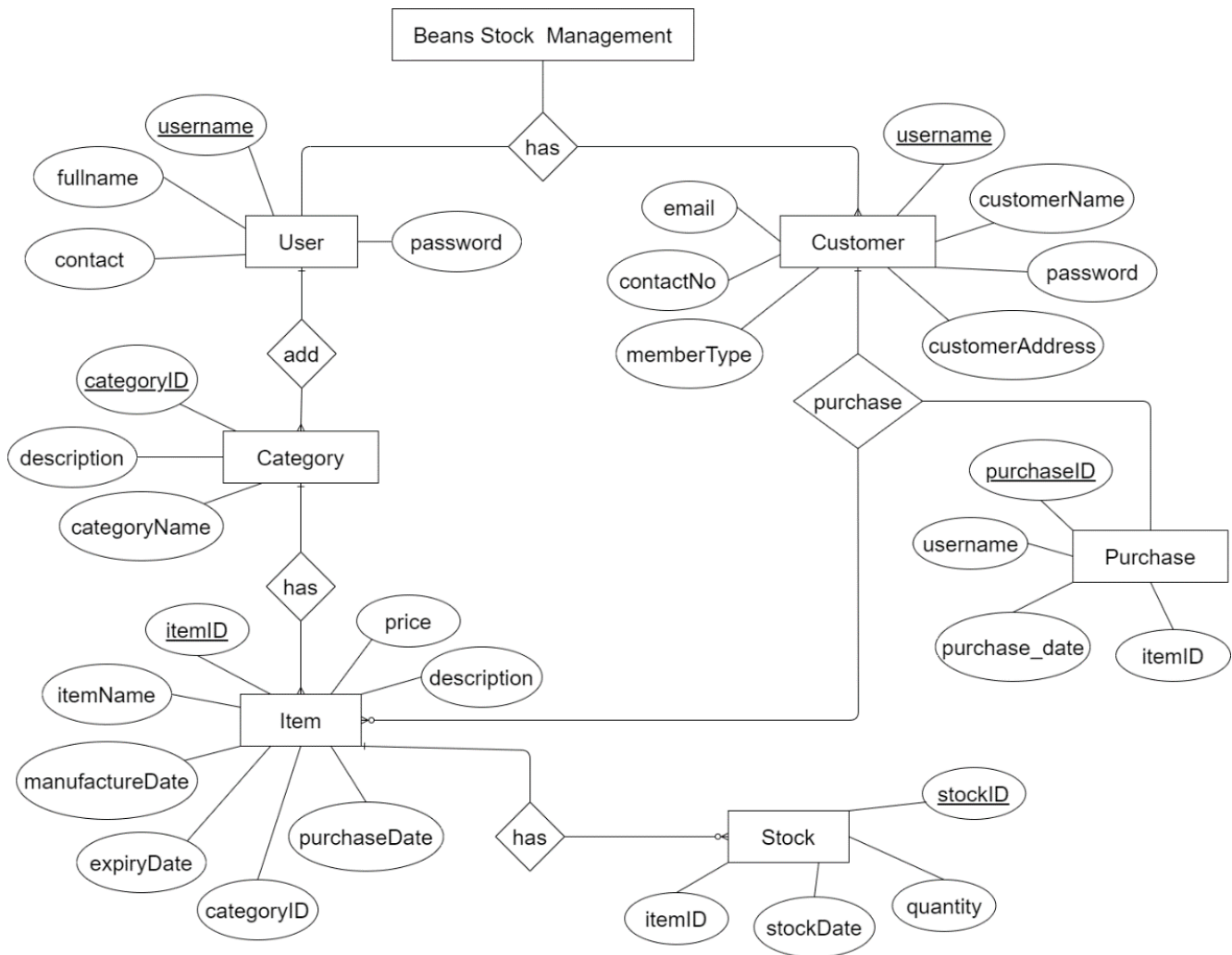


Figure 15: ER diagram of Beans Stock Management

3.3 Class Diagram

Class diagram is a static diagram that illustrates the attributes, operations and relationships between or, among classes of the system. It depicts the static view of a system. The following figure is the class diagram of “Stock Management System”:

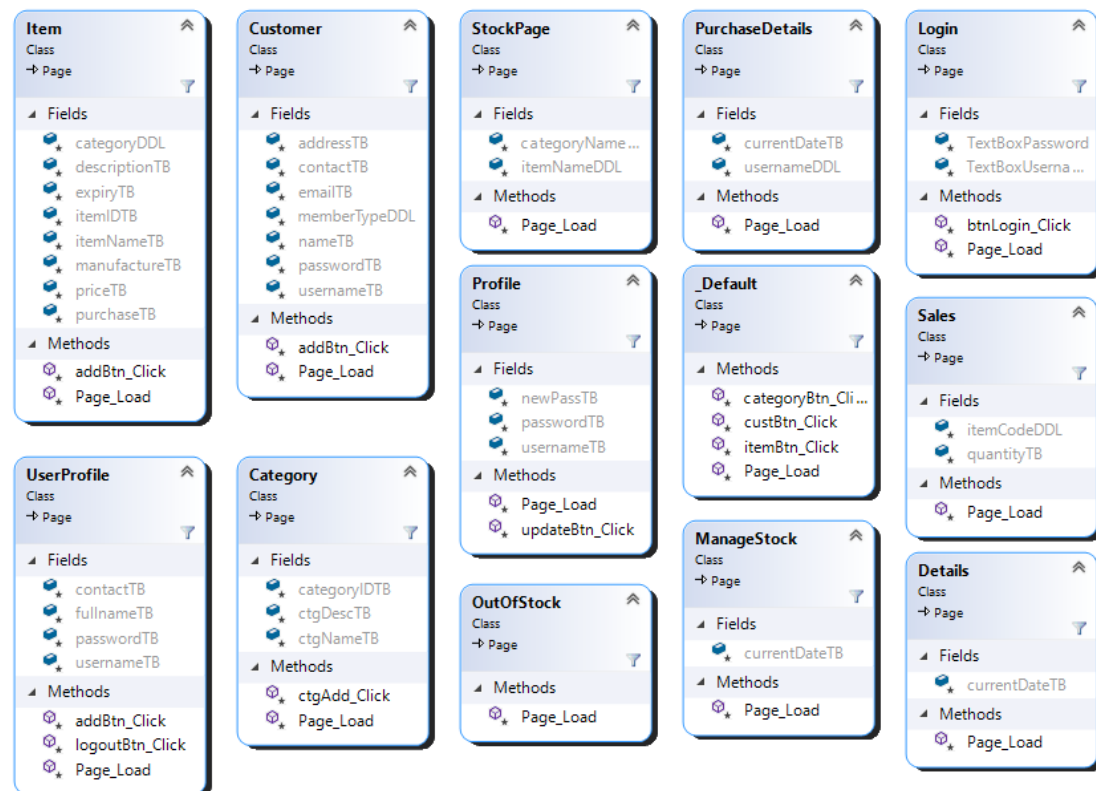


Figure 16: Class Diagram of Beans Stock Management

3.4 Use case diagram

Use case diagram represents the functionality of the system from the user's point of view. The use case diagram basically contains the following elements:

- Actor: An actor interacts with the system and is represented by a stick like figure with their name written underneath.
- Use case or, processes: It is characterized by an oval shape that contains the name of the process or task. It is placed within the boundary of the system.

- Associations: Associations are represented by solid straight lines that illustrates the interactions between the actors and processes.

The use case diagram of “Stock Management System” is given below:

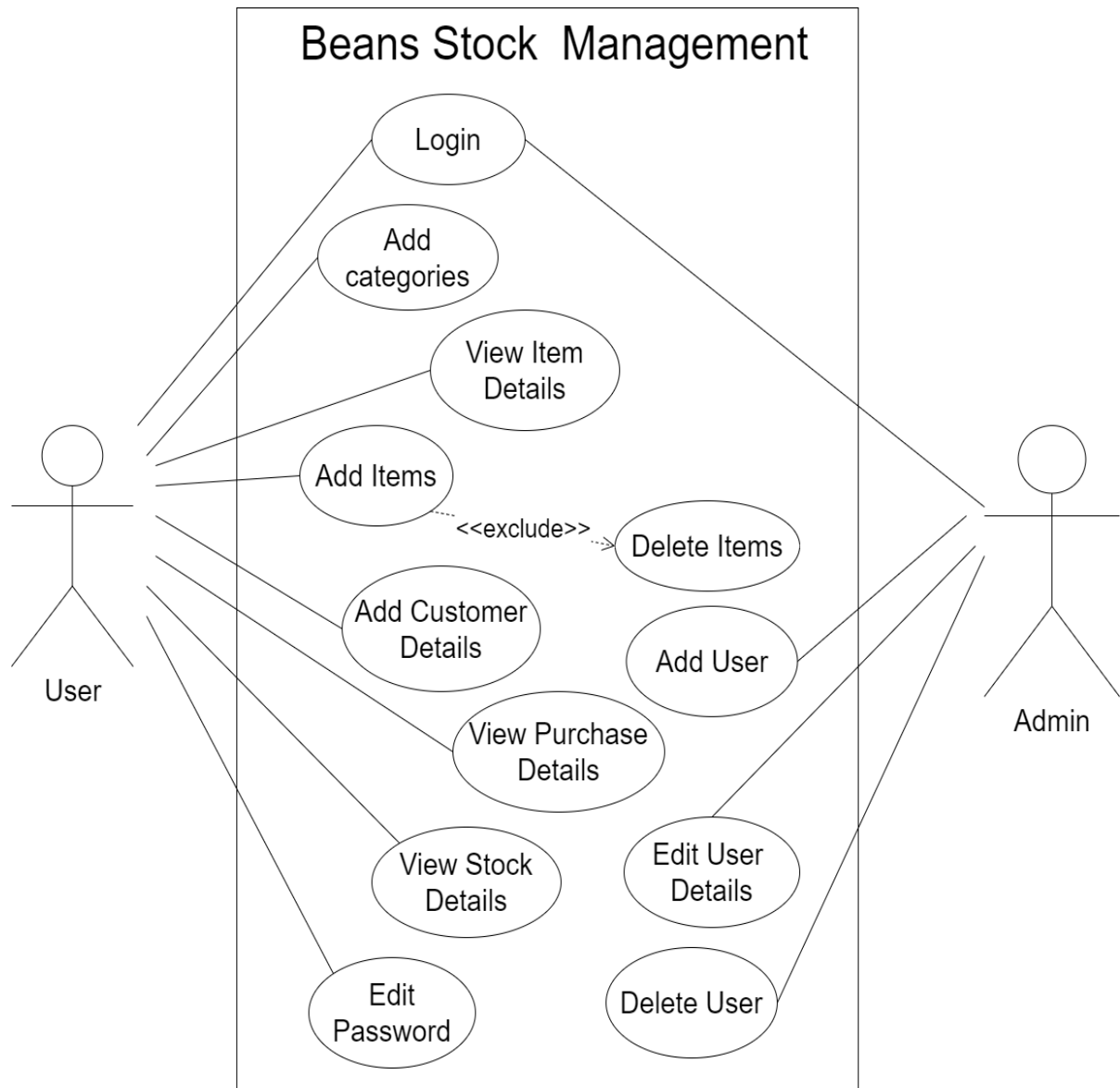


Figure 17: Use Case Diagram of Beans Stock Management

3.5 Algorithm

Algorithm for add category:

- Step 1: Start
- Step 2: Display Category Page
- Step 3: Enter Category details
- Step 4: Does Category id already exists?
 - If yes go to step 3
 - If no go to step 4
- Step 5: Add category detail
- Step 6: Display message
- Step 7: End

Algorithm for add item:

- Step 1: Start
- Step 2: Display item page
- Step 3: Enter item details
- Step 4: does item id already exists?
 - If yes go to step 3
 - If no go to step 5
- Step 5: Add item detail
- Step 6: Display message
- Step 7: End

Algorithm for add customer details:

Step 1: Start

Step 2: Display Customer Page

Step 3: Enter customer details

Step 4: does customer id already exists?

 If yes go to step 3

 If no go to step 5

Step 5: Add customer detail

Step 6: Display Message

Step 7: End

Algorithm for login:

Step 1: Start

Step 2: Display Login page

Step 3: Enter username and password

Step 4: Is username and password valid?

 If yes go to step 3

 If no go to step 5

Step 5: Display Home page

Step 6: End

Algorithm for add user:

Step 1: Start

Step 2: Display login page

Step 3: Enter admin's username password

Step 4: Is username and password valid?

 If no go to step 3

 If yes go to step 5

Step 5: Display admin's home page

Step 6: Enter user's detail

Step 7: Is user's detail valid?

 If yes go to step 8

 If no go to step 6

Step 8: Add user's detail

Step 9: End

Algorithm for delete item:

Step 1: Start

Step 2: Display ManageStock page

Step 3: Display items which are on stock for long time

Step 4: Click delete button

Step 5: End

3.6 Flowchart

Flowchart is a diagrammatical representation of a process of a system or, computer algorithm. It is widely used to understand the complex processes in clear and easy-to-understand diagram. Different shapes are used to display the flow of the program which that their own conventional meanings. The flowchart used to build the program is given below:

- Flowchart of add category:

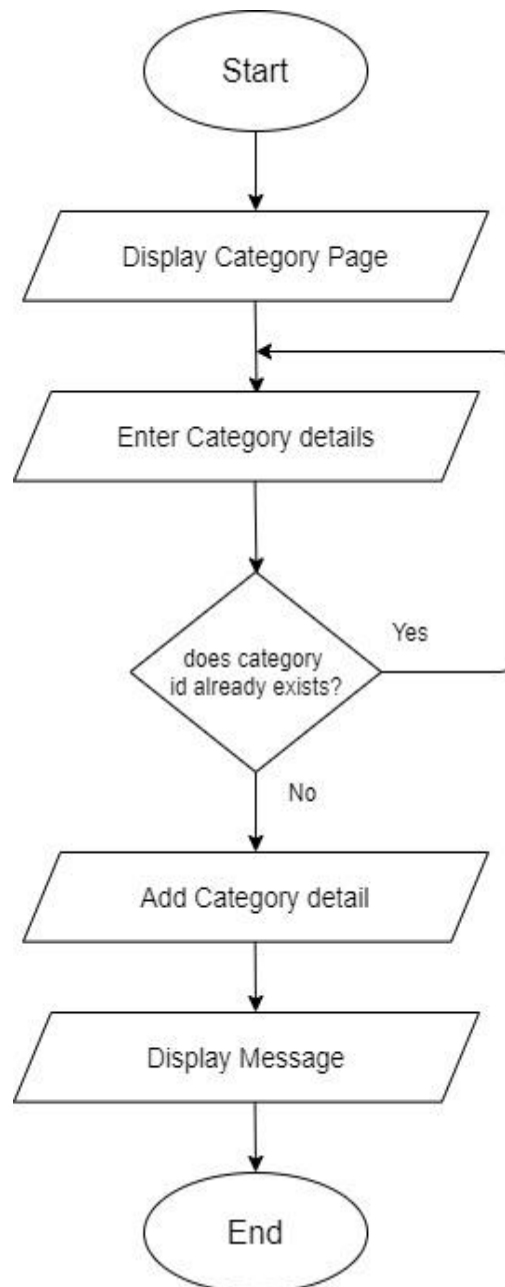


Figure 18: Flowchart of add category

- Flowchart of add item:

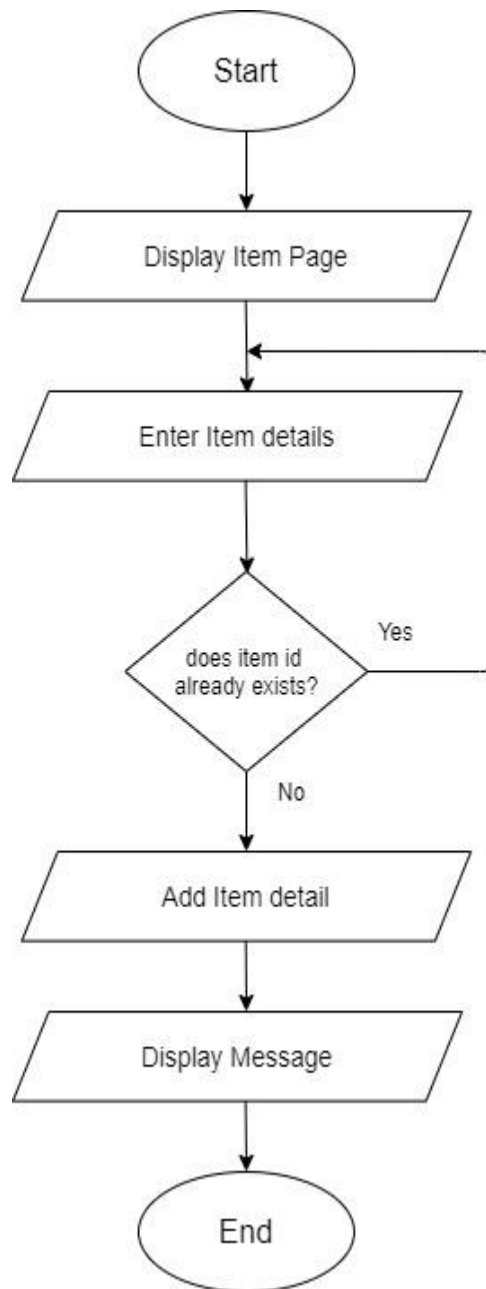


Figure 19: Flowchart of add item

- Flowchart of add customer details:

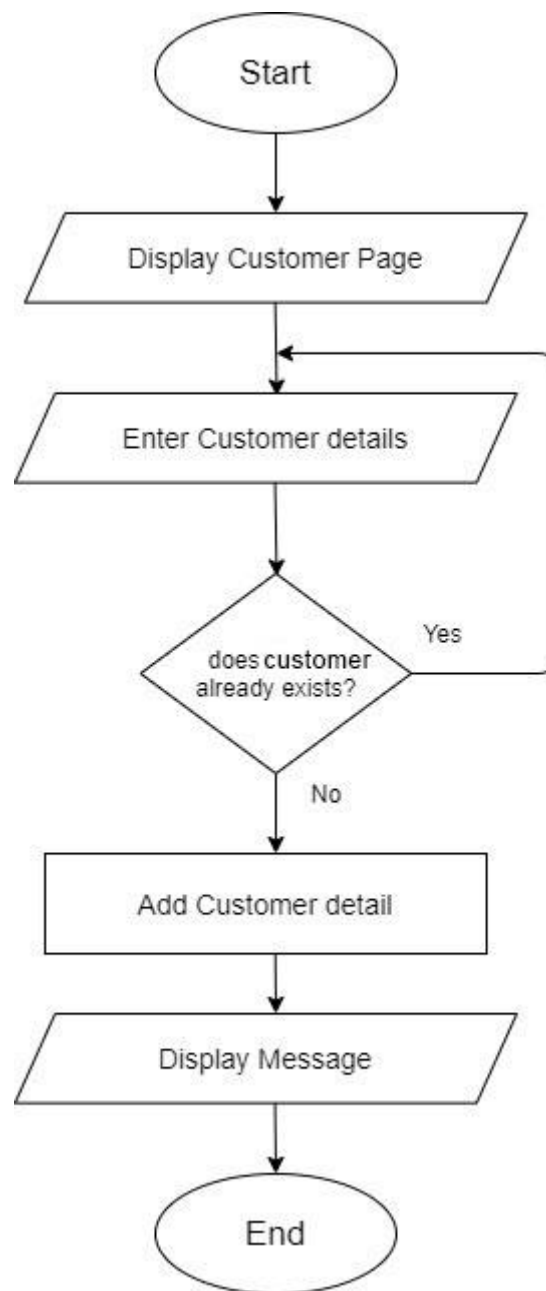


Figure 20: Flowchart of add customer details

- Flowchart of login:

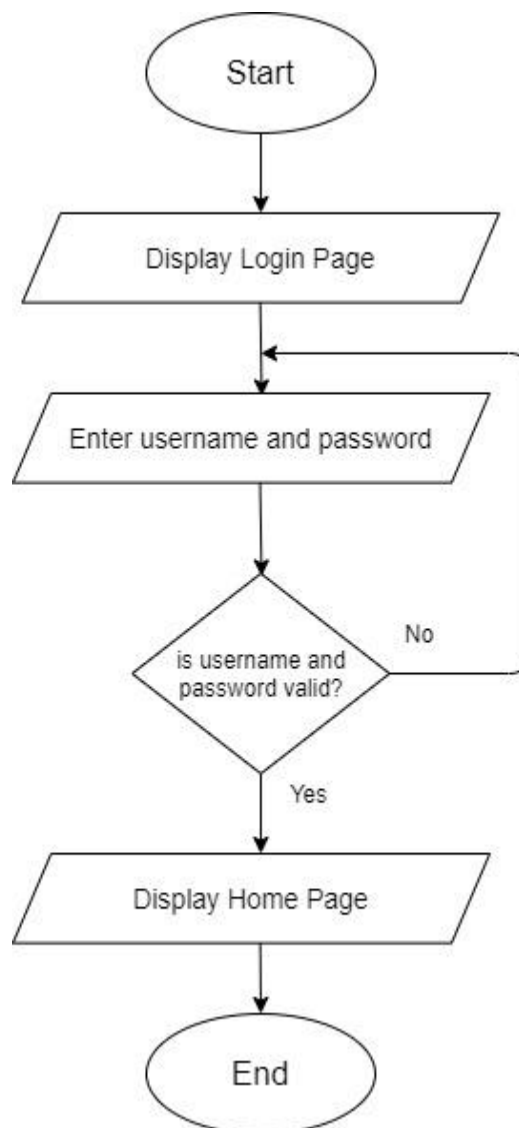


Figure 21: Flowchart of Login

- Flowchart of add user details:

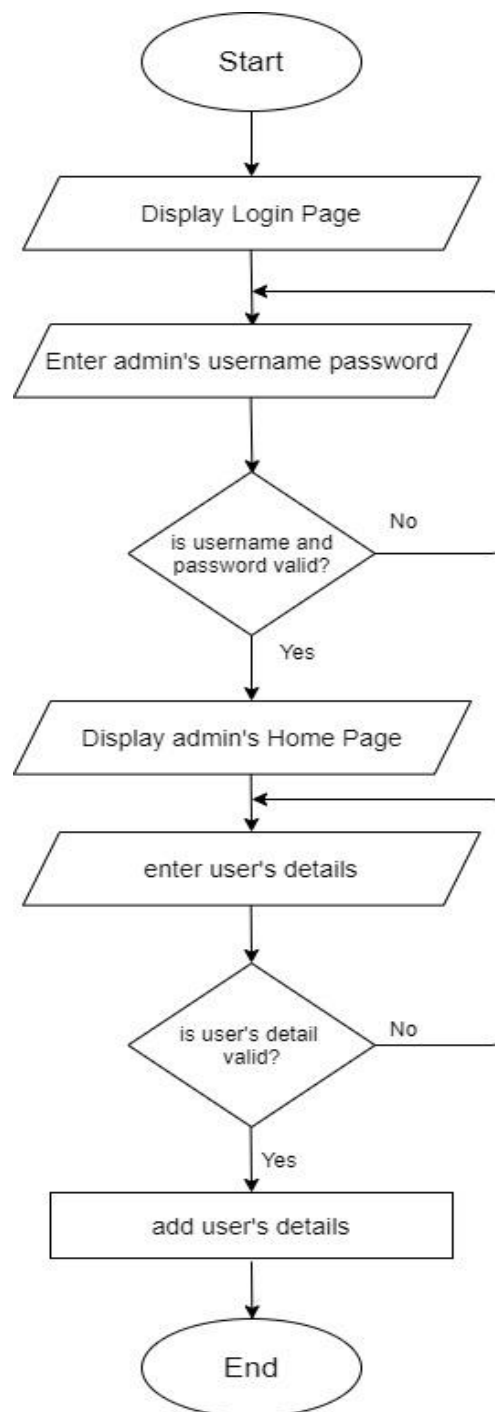


Figure 22: Flowchart of Add User Details

- Flowchart of delete item:

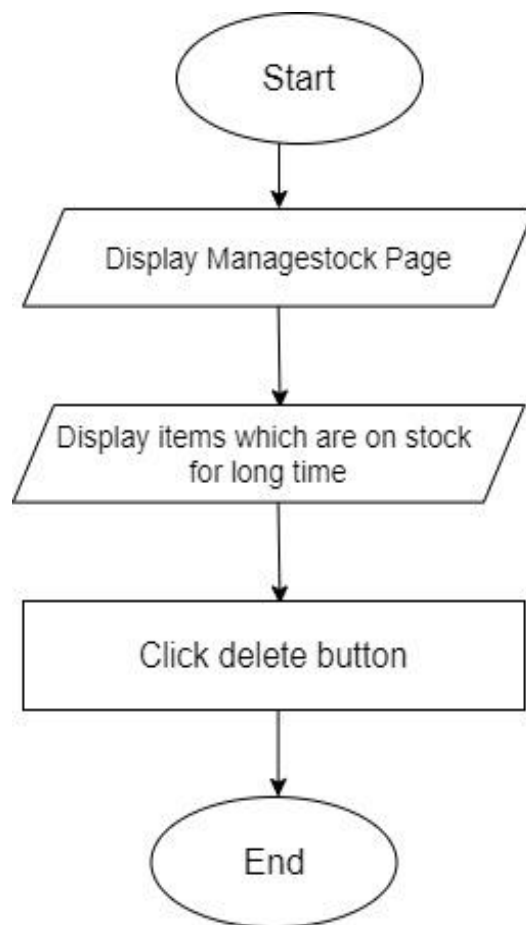


Figure 23: Flowchart of Delete Item

4. Description of Classes Purpose and Methods

4.1 Purpose of classes

Class	Purpose
Default	This class is the dashboard implementation of the system. Here user can see the home page of user after successful login.
Category	This class consists of a category form where user can add new category details to the system.
Customer	This class consists of customer details form where user can add the new customer details to the system.
Details	This class displays the list of all the customers who have not bought any items in the last 31 days and also the list of items that has not been sold in the last 31 days.
Item	This class consists of a form that ask user to add item details in the system.
Login	This class consists of a login form where the system ask user to provide login details i.e., username and password. Here system also check whether the user is admin or, not. If the credentials are correct then it redirects user to the respective home page.
Manage Stock	This class consists of manage stock form where the items that are on stock for a long time is displayed and user can delete the items from the stock.
Out of Stock	This class consist of out-of-stock form where list of item sorted by its name in ascending order is displayed along with the item sorted by stock quantity in high to low and item sorted by stocked date are displayed.
Profile	This class consists of form where user can edit their details i.e., they can change existing password to new password by providing username.

Purchase Details	This class consists of form where list of purchase details made within a month by the particular customer is displayed by providing the customer username.
Sales	This class consists of sales form where user can make sales of items to the customer by providing the item code and quantity. And when the user enters the code, system displayed the item details i.e., item name, description, category name, purchase date, manufacture date, expiry date price and total price.
Stock Page	This class consists of stock details form where user can view all the item details available in the stock by providing category name and can also view item details available in the stock by providing the item name.
User Profile	This class is the admin dashboard where admin can add the new user to the system by providing username, full name of user, contact and password. Admin can also edit and delete the existing user from the system.

Table 1: Table of purpose of classes

4.2 Description of methods

Category:

Class	Methods	Description
Category	Page_Load()	This method is automatically be called every time the page is loaded.
	ctgAdd_Click()	This method adds category_id, category_name and category_description in category table.

Table 2: Method description of Category class

Customer:

Class	Methods	Description
Customer	Page_Load()	This method is automatically be called every time the page is loaded.
	addBtn_Click()	This method adds customer details in customer table.

Table 3: Method description of Customer class

Details:

Class	Methods	Description
Details	Page_Load()	<p>This method imports the current date and time and set a textbox name "currentDateTB" in a datetime format.</p> <p>This method displays list of all customers who have not bought any items in the last 31 days.</p> <p>This method displays list of all items that hasn't been sold in the last 31 days</p>

Table 4: Method description of Details class

Item:

Class	Methods	Description
Item	Page_Load()	This method is automatically be called every time the page is loaded.
	addBtn_Click()	This method adds all the items details in item table.

Table 5: Method description of Item class

Login:

Class	Methods	Description
Login	Page_Load()	This method is automatically be called every time the page is loaded.
	btnLogin_Click()	This method redirects to the "USerProfile.aspx" page if the admin's username and password, and user's username and password are correct.

Table 6: Method description of Login class

Manage Stock:

Class	Methods	Description
Manage Stock	Page_Load()	This method imports the current date and time and set a textbox name "currentDateTB" in a datetime format. This method displays a list of all items which are on stock for long time and delete the items.

Table 7: Method description of Manage Stock class

Out of Stock:

Class	Methods	Description
Out of Stock	Page_Load()	This method displays three item tables which are sorted by name in ascending order, by stock quantity in high to low and by item stocked date in descending order.

Table 8: Method description of Out of Stock class

Profile:

Class	Methods	Description
Profile	Page_Load()	This method is automatically be called every time the page is loaded.
	updateBtn_Click()	This method allows users to update their username and password.

Table 9: Method description of Profile class

Sales:

Class	Methods	Description
Sales	Page_Load()	This method takes item id and quantity from user to retrieve information like item name, category, price etc. for billing purpose.

Table 10: Method description of Sales class

User Profile:

Class	Methods	Description
User Profile	Page_Load()	This method is automatically be called every time the page is loaded.
	logoutBtn_Click	This method logouts user and redirect them to them to the login page.

	addBtn_Click()	This method adds a new user in a user table.
--	----------------	--

Table 11: Method description of User Profile class

Purchase Details:

Class	Methods	Description
Purchase Details	Page_Load()	This method takes customer name and date, and view purchases detail that they have been purchased by the last 31 Days.

Table 12: Method description of Purchase Details class

Stock Page

Class	Methods	Description
Stock Page	Page_Load()	This method takes category name from user and displays all items available in stocked. And takes item name from user and displays items detail with available quantity on stock.

Table 13: Method description of Stock Page class

5. Testing

Test case 1:

Objective	To test whether program run successfully without any error.
Action	Run the program.
Expected output	The ASP.net program should execute successfully in the chrome.
Actual output	The ASP.net program executes successfully in the chrome.
Result	Test successful.

Table 14: Test Case 1

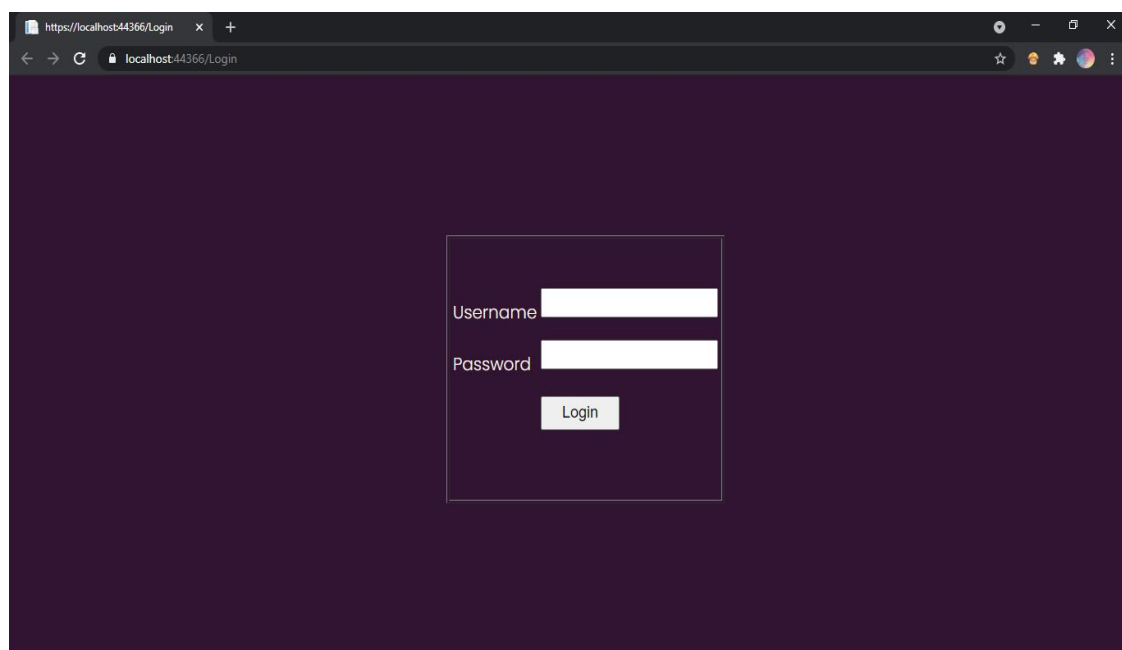


Figure 24: Display Login page when user run the program

Test case 2:

Objective	To test whether user can login to the system successfully.
Action	Provide user's detail i.e., username and password.
Expected output	User should login successfully and user home page should display.
Actual output	User logs in successfully and user home page displays on the screen.
Result	Test successful.

Table 15: Test Case 2

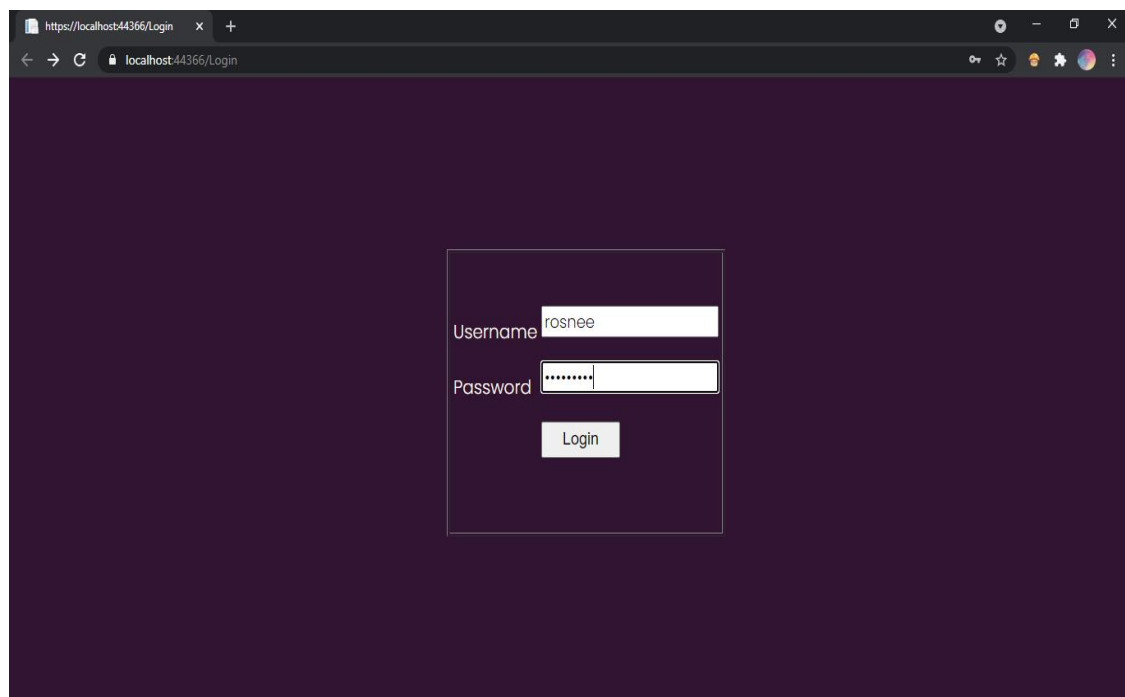


Figure 25: User provide login details

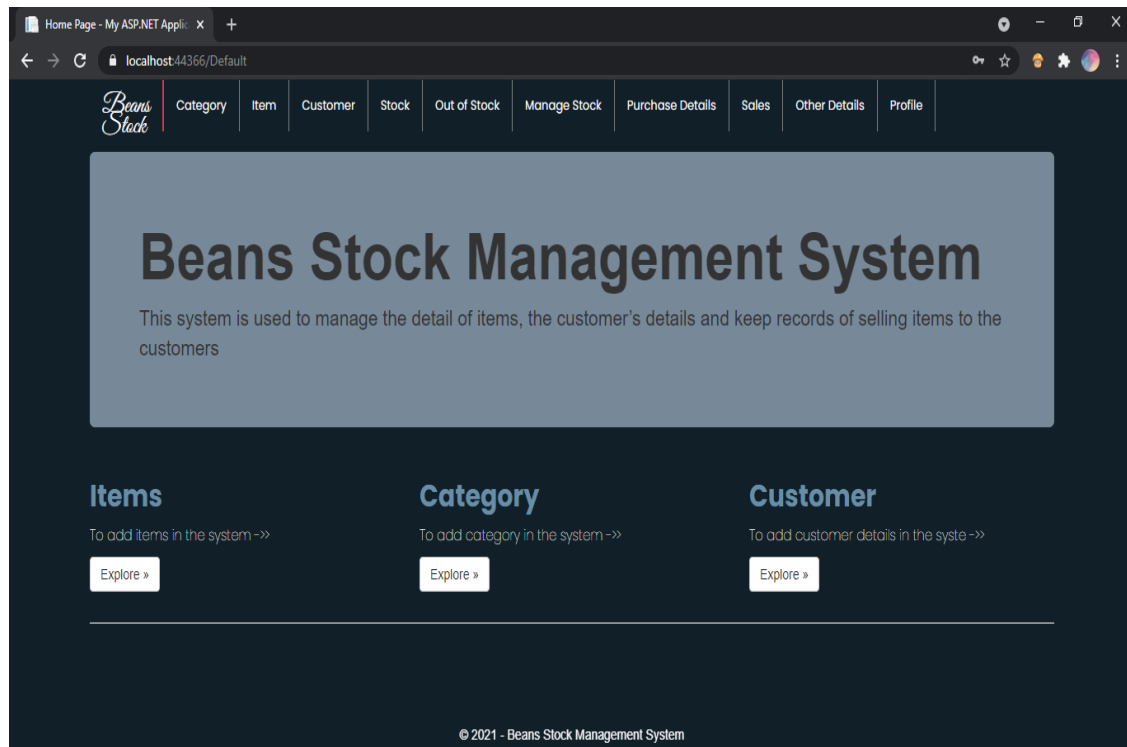


Figure 26: Display home page after successful login

Test case 3:

Objective	To test whether admin can login to the system successfully or not.
Action	Provide admin's detail i.e., name and password.
Expected output	Admin should login successfully and admin panel should display.
Actual output	Admin logs in successfully and admin panel displays on the screen.
Result	Test successful.

Table 16: Test Case 3

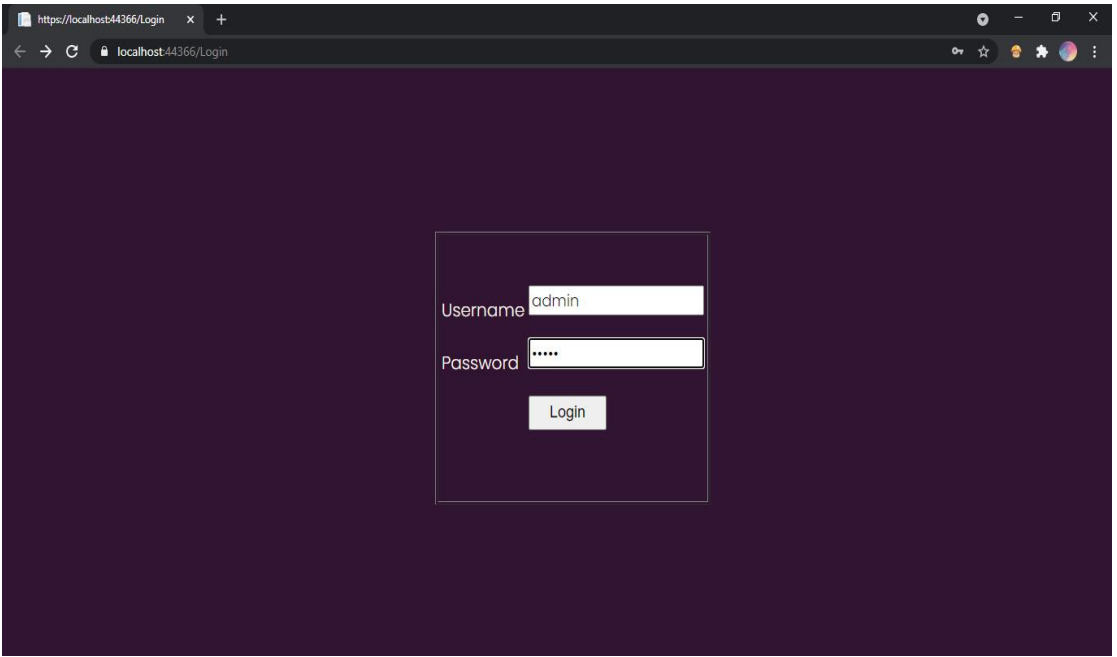


Figure 27: Provide admin login details

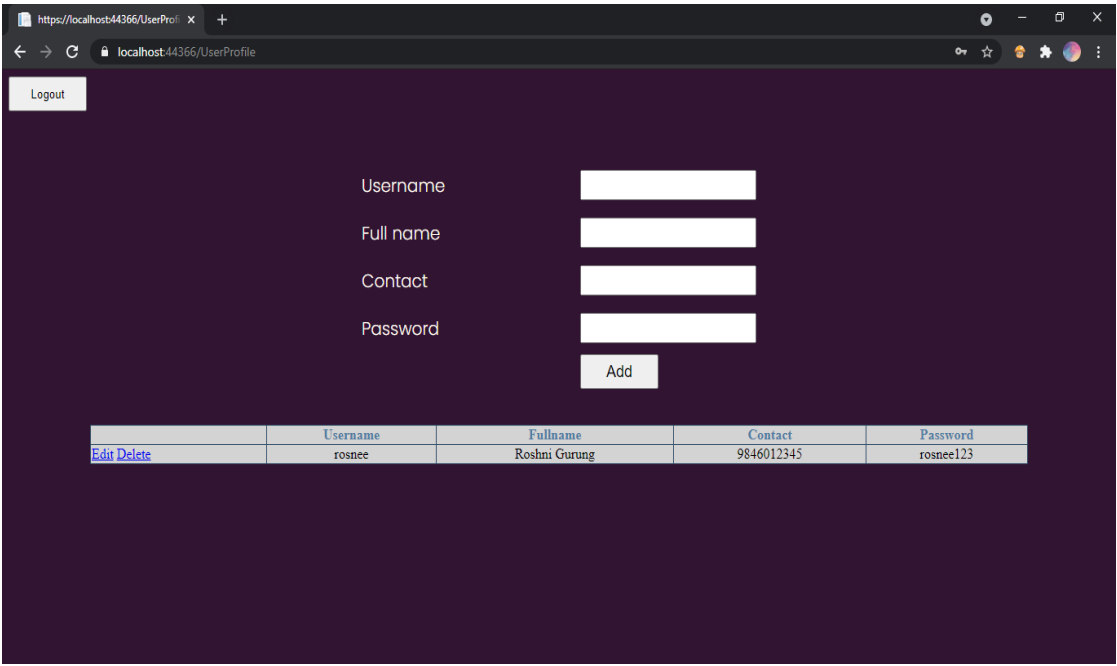


Figure 28: Display admin dashboard after successful login

Test case 4:

Objective	To test whether user can add category successfully.
Action	Enter category details.
Expected output	Category should be added in the database.
Actual output	Category is added in the database successfully.
Result	Test successful.

Table 17: Test Case 4

The screenshot shows a web browser window with the title 'My ASP.NET Application'. The address bar shows 'localhost:44366/Category'. The application has a dark purple theme. A navigation bar at the top contains the 'Beans Stock' logo and several menu items: Category, Item, Customer, Stock, Out of Stock, Manage Stock, Purchase Details, Sales, Other Details, and Profile. The 'Category' menu item is highlighted. The main content area features a large 'Category' title on the left. To the right, there is a form with three fields: 'Category ID' (a dropdown menu showing '4'), 'Category Name' (a text input field containing 'Dairy Products'), and 'Description' (a text area containing 'Made from milk, including butter, cheese, ice cream'). Below these fields is an 'Add' button. At the bottom of the page, there is a copyright notice: '© 2021 - Beans Stock Management System'.

Figure 29: Provide category details

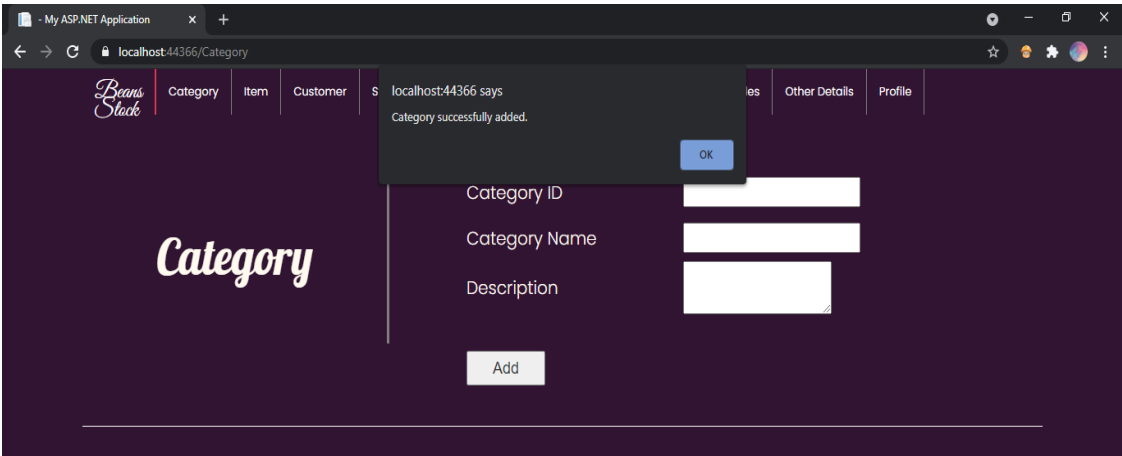


Figure 30: Display successful message

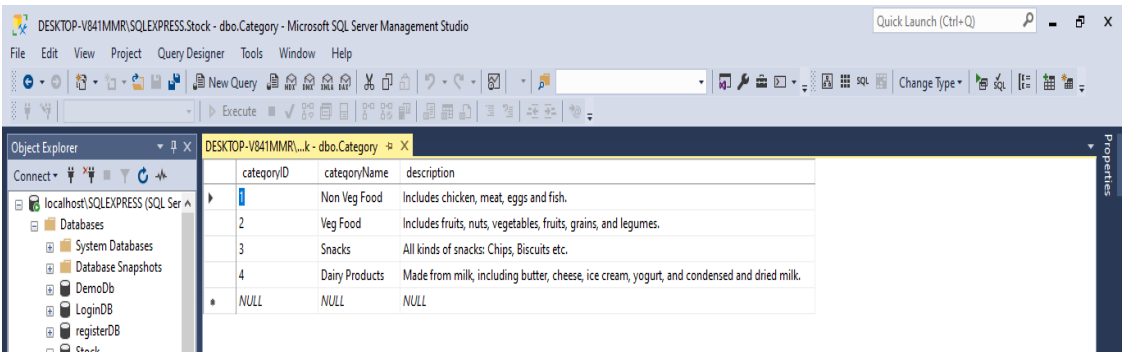


Figure 31: Dairy Product category is added to the system

Test case 5:

Objective	To test whether user can add items successfully.
Action	Provide items details.
Expected output	Item should be added in the system.
Actual output	Item is added to the system successfully.
Result	Test successful.

Table 18: Test Case 5

My ASP.NET Application

localhost:44366/Item

Beans Stock

Category Item Customer Stock Out of Stock Manage Stock Purchase Details Sales Other Details Profile

Add Items

Item ID: 6

Item Name: Apple

Description: Marpha Apple

Price: 50

Purchase Date: 01-May-2021

Category Name: Veg Food

Manufacture Date: 25-Apr-2021

Expiry Date: 20-May-2021

Add

© 2021 - Beans Stock Management System

Figure 32: Provide item details

My ASP.NET Application

localhost:44366/Item

Beans Stock

Category Item Customer Stock Out of Stock Manage Stock Purchase Details Sales Other Details Profile

localhost:44366 says
Successfully added

OK

Add Items

Item ID:

Item Name:

Description:

Price:

Purchase Date: 01-May-2021

Category Name: Non Veg Food

Manufacture Date: 25-Apr-2021

Expiry Date: 20-May-2021

Add

© 2021 - Beans Stock Management System

Figure 33: Display successful message

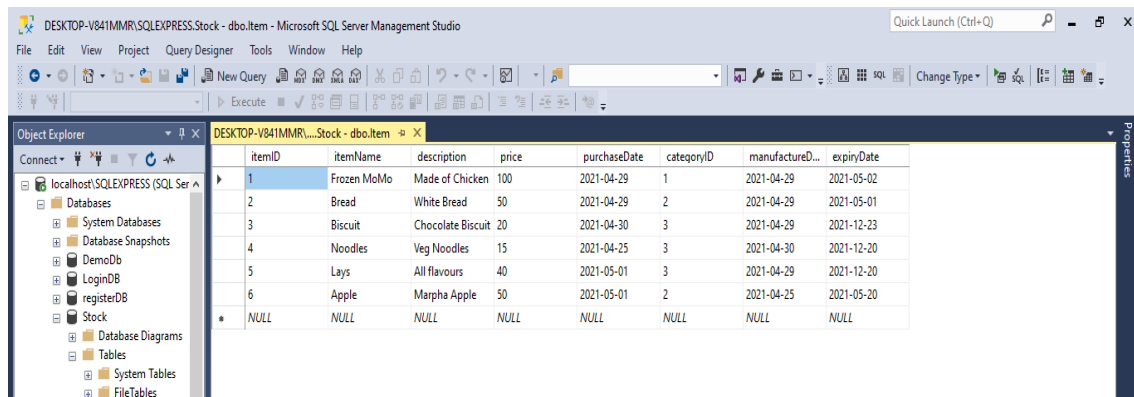


Figure 34: New Item Apple successfully added to the system

Test case 6:

Objective	To test whether user can add customer details in the system successfully.
Action	Provide customer details.
Expected output	Customer details should be stored in the system.
Actual output	Customer details are stored successfully in the system.
Result	Test successful.

Table 19: Test Case 6

Customer

Username:

Customer Name:

Address:

Contact No.:

Member Type:

Email:

Password:

© 2021 - Beans Stock Management System

Figure 35: Provide customer details

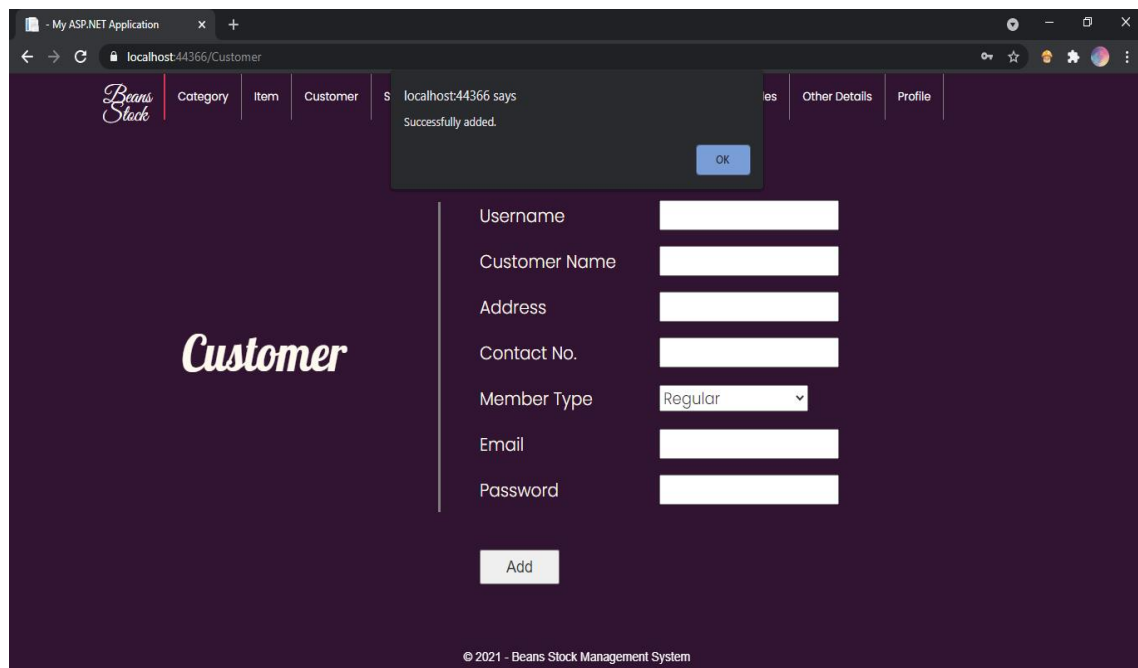


Figure 36: Display successful message

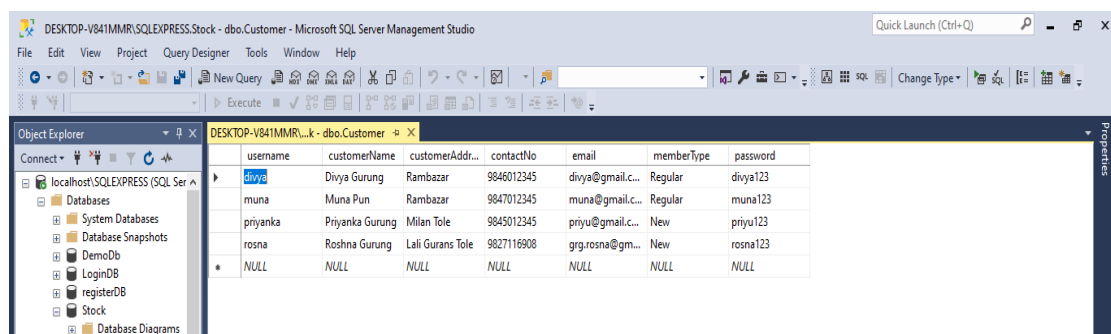


Figure 37: New customer added to the system successfully

Test case 7:

Objective	To test whether all available items of certain category in the stock displays.
Action	Select the category name
Expected output	All available items of given category should display.
Actual output	All available items of given category display on the screen.
Result	Test successful.

Table 20: Test Case 7

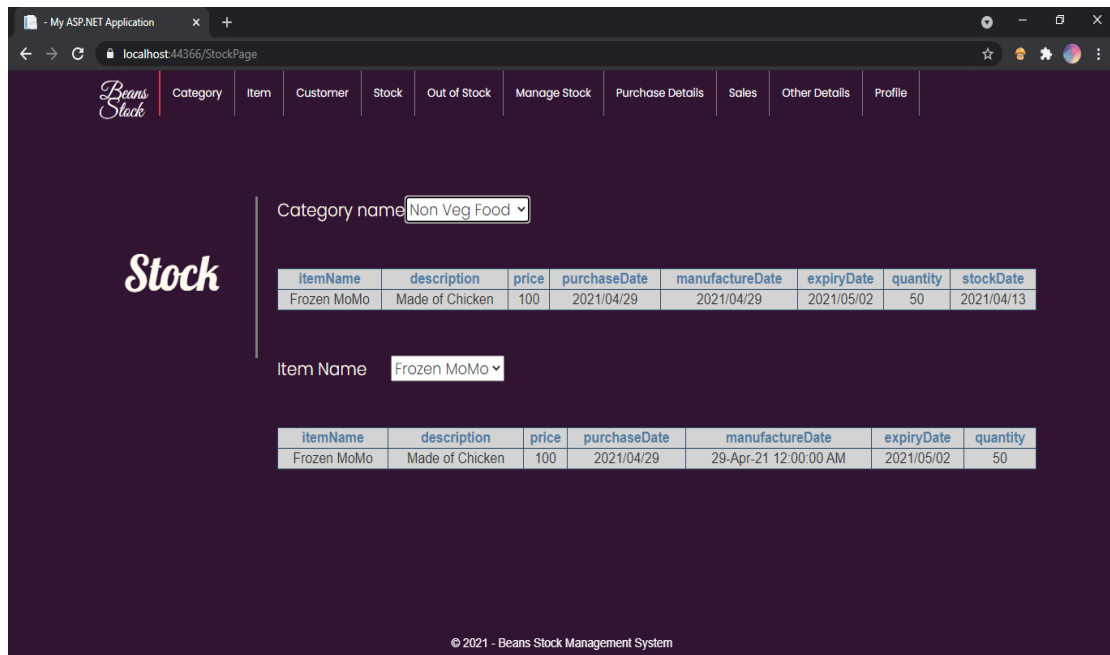


Figure 38: Display item details of particular category which are available in the stock

Test case 8:

Objective	To test whether details of selected item displays.
Action	Select the item name.
Expected output	Details of selected item should display.
Actual output	Details of selected item displays on the screen.
Result	Test successful.

Table 21: Test Case 8

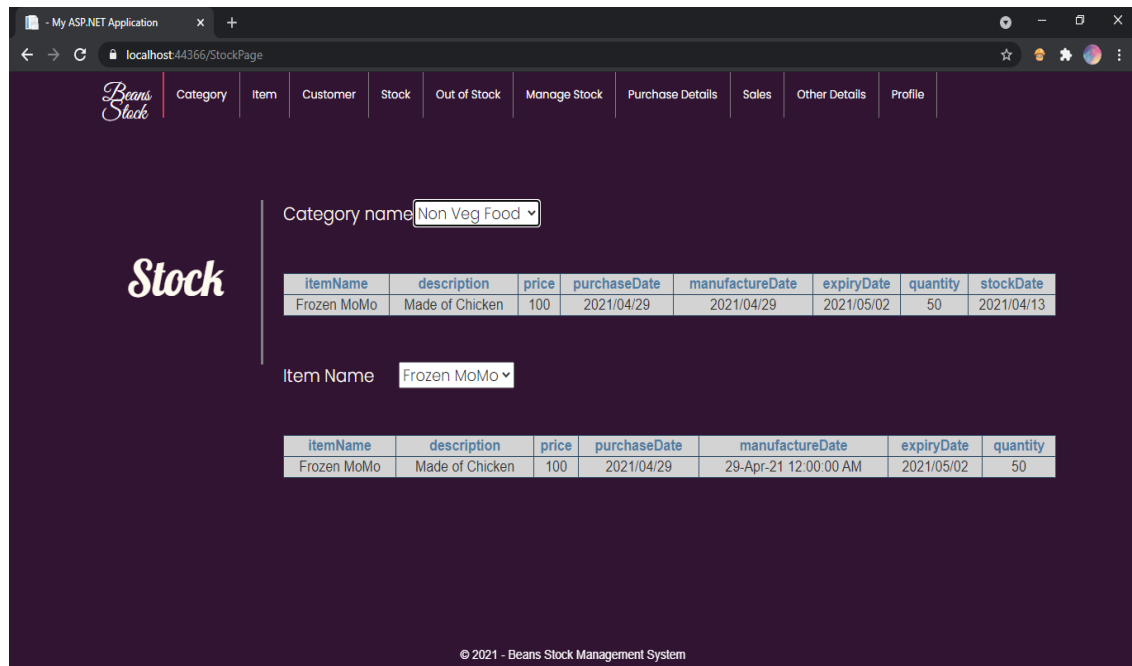


Figure 39: Display item details of selected item name which are available in the stock

Test case 9:

Objective	To test whether the program displays the purchase details of the customer who had purchased by last 31 days.
Action	Select customer username
Expected output	Purchase details by last 31 days of particular customer should display.
Actual output	Purchase details by last 31 days of particular customer displays.
Result	Test successful.

Table 22: Test Case 9

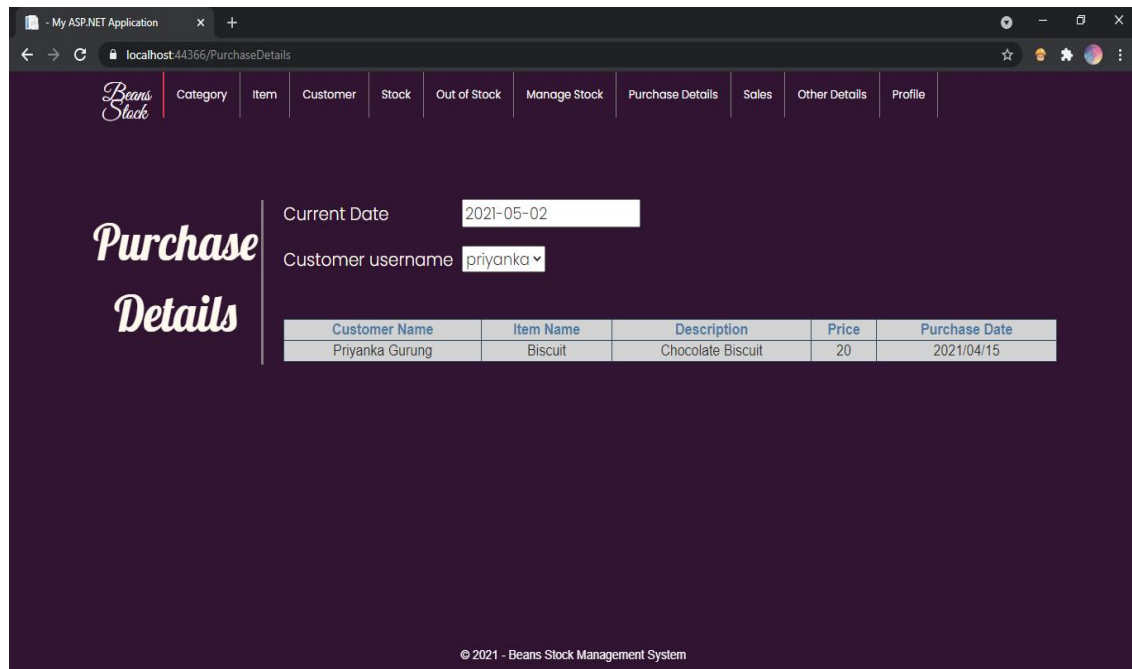


Figure 40: Display the purchase details of the customer who had purchased by last 31 days

Test case 10:

Objective	To test whether the program displays message to the user when the item is running out of stock at the time of login.
Action	User login to the system.
Expected output	Message should display when the item is less than 10 in the stock.
Actual output	Message displays when the item is less than 10 in the stock.
Result	Test successful.

Table 23: Test Case 10

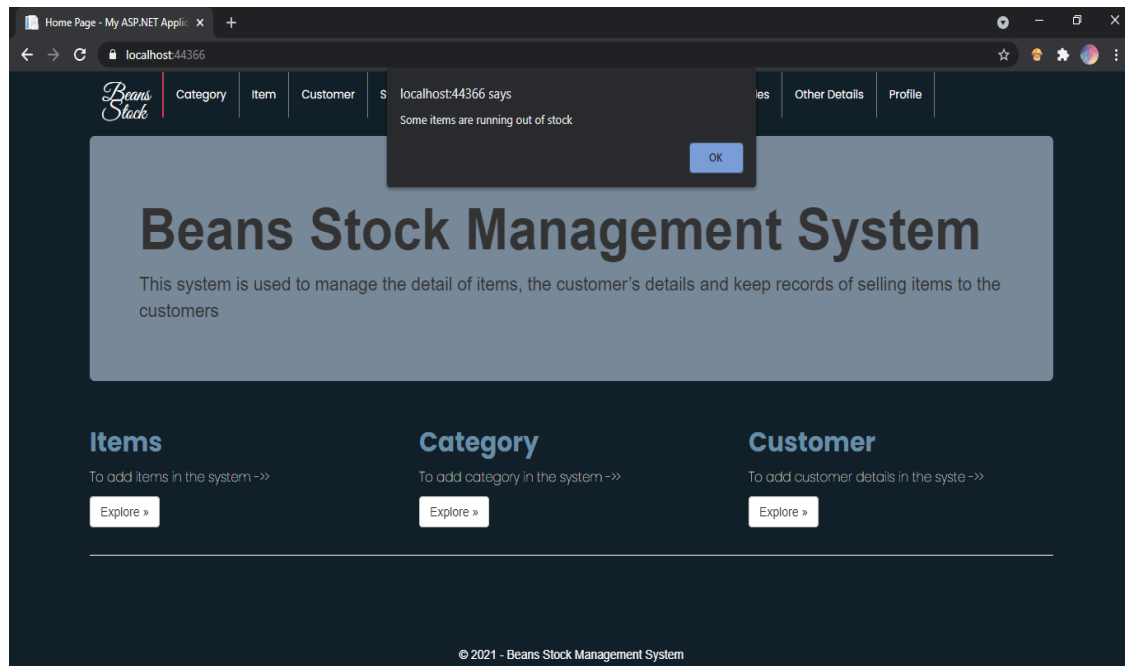


Figure 41: Display message to the user when the item is running out of stock at the time of login

Test case 11:

Objective	To test whether the system allow user to remove all items that are on stock for long time.
Action	Click the delete button.
Expected output	Items which are on stock for long time should remove from the system.
Actual output	Items which are on stock for long time are removed from the system.
Result	Test successful.

Table 24: Test Case 11

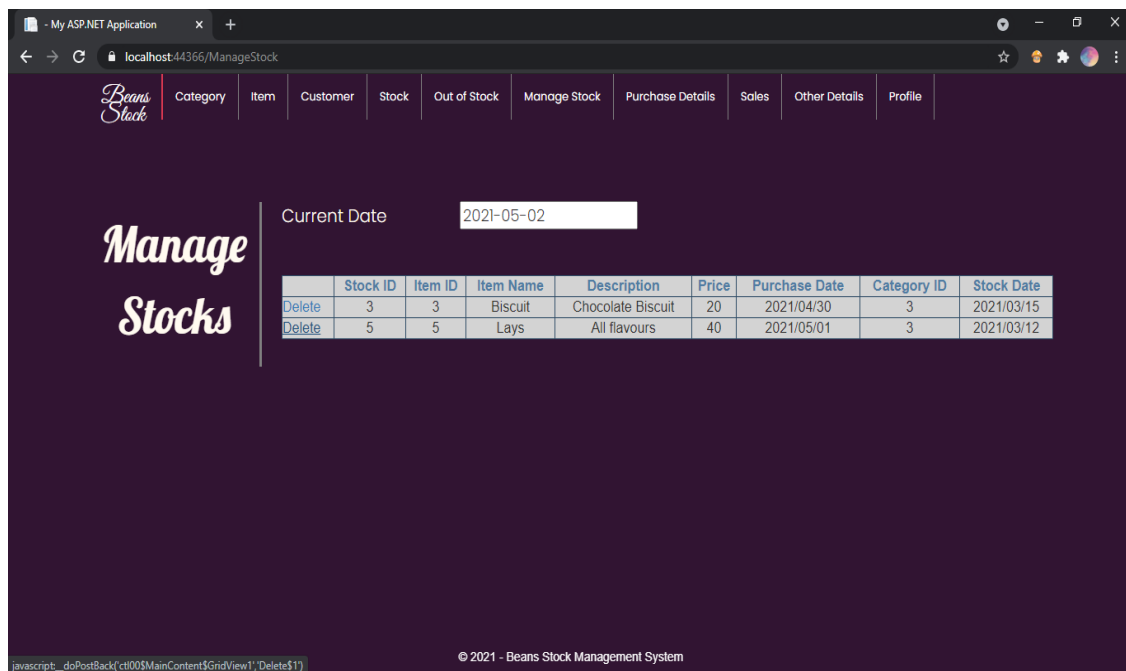


Figure 42: Before removing the items that are stocked for a long time in the system

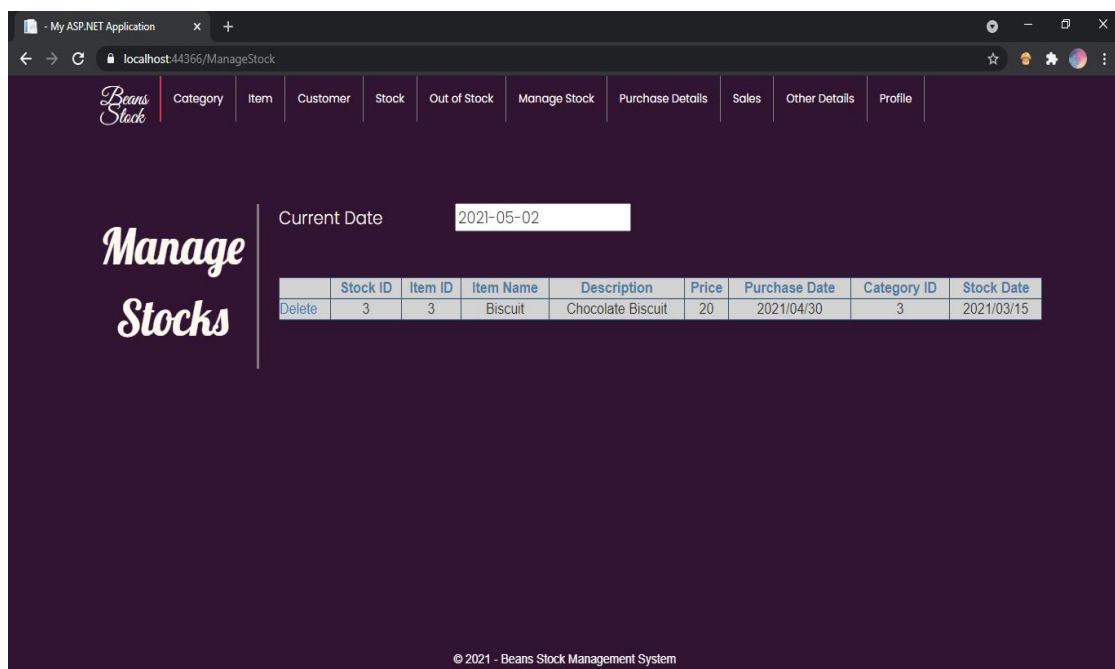


Figure 43: After removing the items that are stocked for a long time in the system

Test case 12:

Objective	To test whether the system displays all the items that are currently out of stock.
Action	Run the “Out of Stock” page.
Expected output	List of items that are currently out of stock should display.
Actual output	Displays list of items that are currently out of stock.
Result	Test successful.

Table 25: Test Case 12

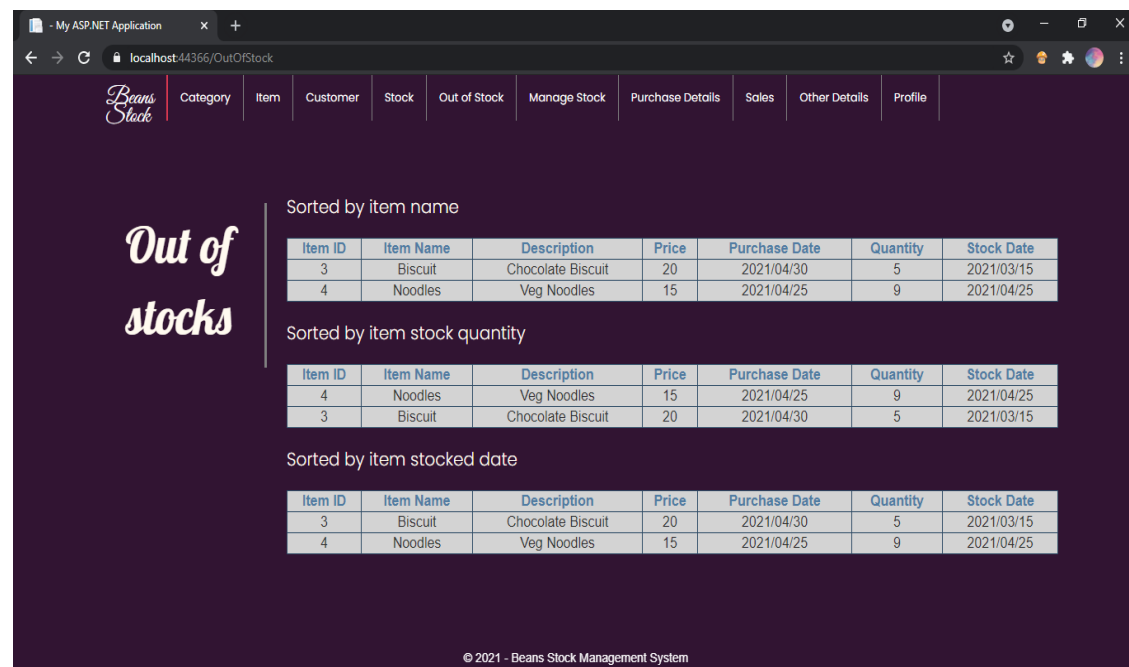


Figure 44: Display all the items that are currently out of stock.

Test case 13:

Objective	To test whether the system displays all the customers who never bought any item in the last 31 days.
Action	Run “Other Details” page.
Expected output	Display list of customers who never bought any item in the last 31 days.

Actual output	Displays list of customers who never bought any item in the last 31 days.
Result	Test successful.

Table 26: Test Case 13

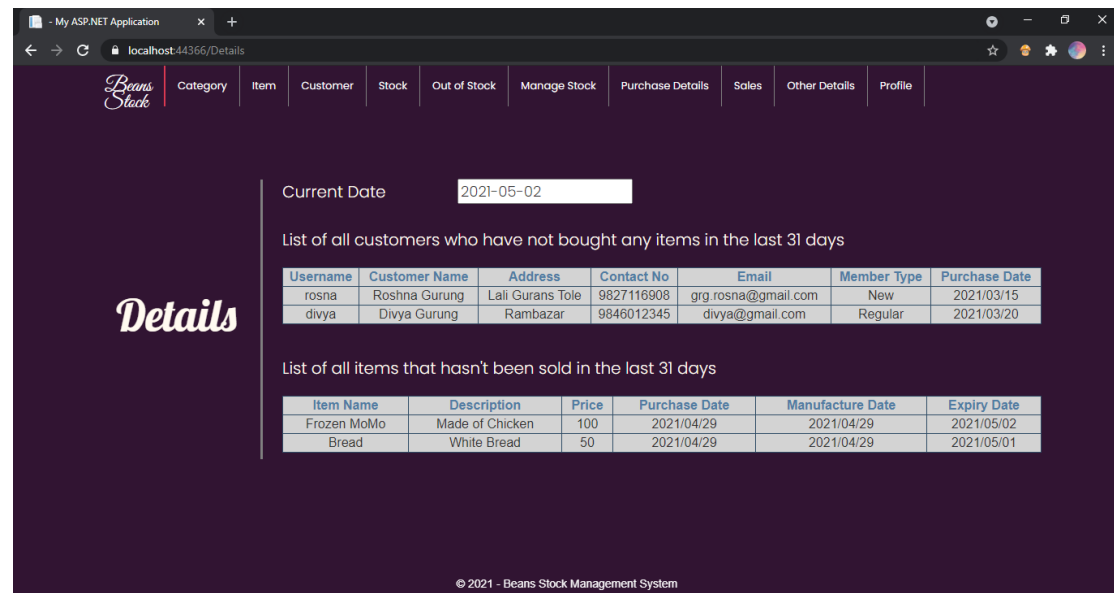


Figure 45: Display list of customers who never bought any item in the last 31 days

Test case 14:

Objective	To test whether the system displays all the items that are not sold in the last 31 days.
Action	Run "Other Details" page.
Expected output	List of items that are not sold in the last 31 days should display.
Actual output	Displays list of items that are not sold in the last 31 days.
Result	Test successful.

Table 27: Test Case 14

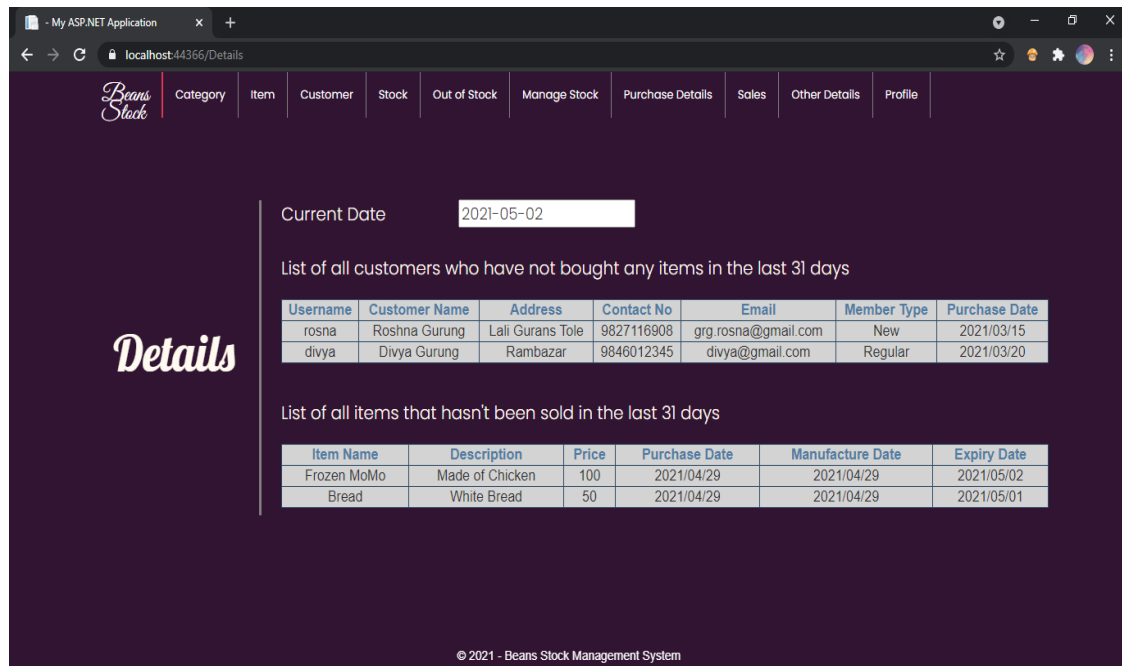


Figure 46: Display list of items that are not sold in the last 31 days

Test case 15:

Objective	To test whether the user can change their own passwords.
Action	Provide the user details.
Expected output	Password should change and successful message should display.
Actual output	Change password and displays successful message.
Result	Test successful.

Table 28: Test Case 15

Beans Stock

Category | Item | Customer | Stock | Out of Stock | Manage Stock | Purchase Details | Sales | Other Details | Profile

Profile

Username:

Password:

New Password:

© 2021 - Beans Stock Management System

Figure 47: Provide customer username and password details

Beans Stock

Category | Item | Customer | Stock | Out of Stock | Manage Stock | Purchase Details | Sales | Other Details | Profile

Profile

Username:

Password:

New Password:

localhost:44366 says
Successfully updated.

Figure 48: Password changed successfully

DESKTOP-V841MMR\SQLEXPRESS.Stock - dbo.UserTable - Microsoft SQL Server Management Studio

File Edit View Project Query Designer Tools Window Help

Object Explorer

Connect

localhost\SQLEXPRESS (SQL Ser)

Databases

System Databases

username	fullname	contact	password
rosnee	Roshni Gurung	9846012345	rosnee
NULL	NULL	NULL	NULL

Figure 49: Before changing password

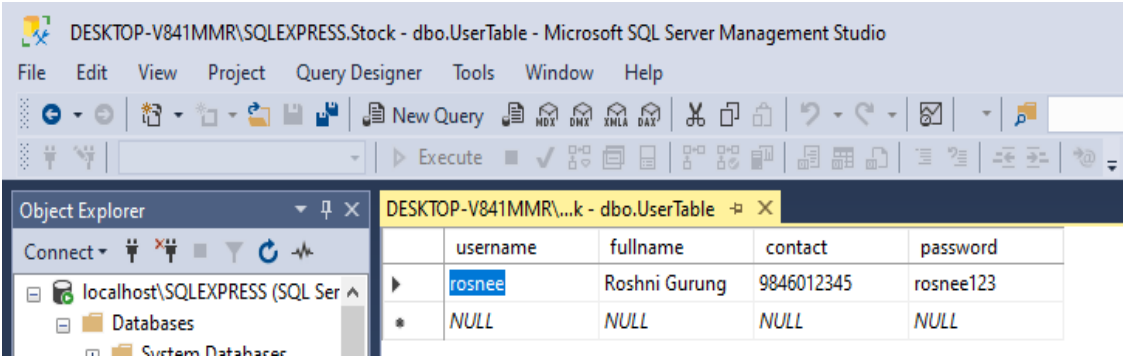


Figure 50: After changing password

Test case 16:

Objective	To test whether the admin can add users or not.
Action	Enter user’s details.
Expected output	User should be added in the database.
Actual output	User is added in the database.
Result	Test successful.

Table 29: Test Case 16

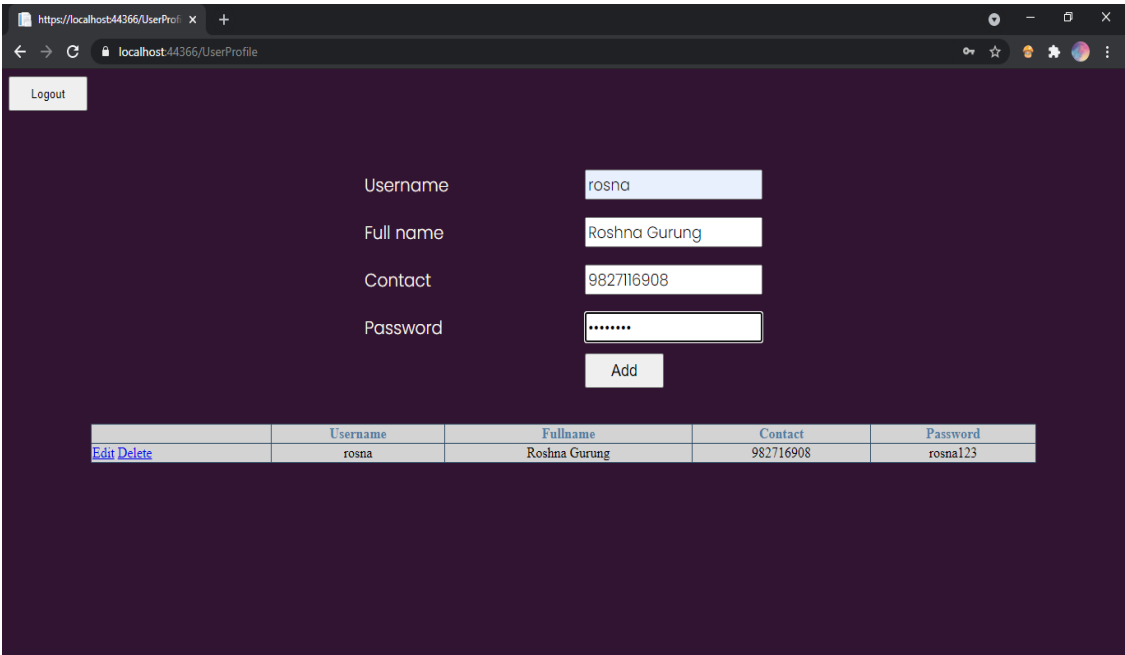


Figure 51: Admin provide user details

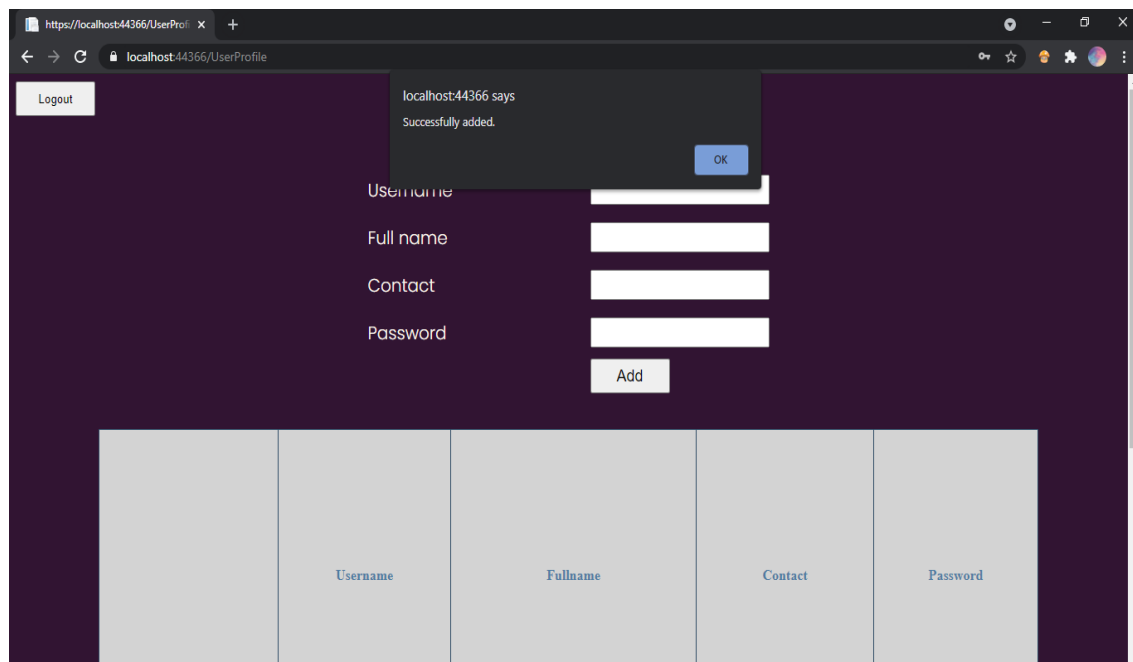


Figure 52: User successfully added to the system

Test case 17:

Objective	To test whether the admin can change user's details.
Action	Edit user's detail.
Expected output	User's detail should change and be updated in the database.
Actual output	User's detail changes and updates in the database.
Result	Test successful.

Table 30: Test Case 17

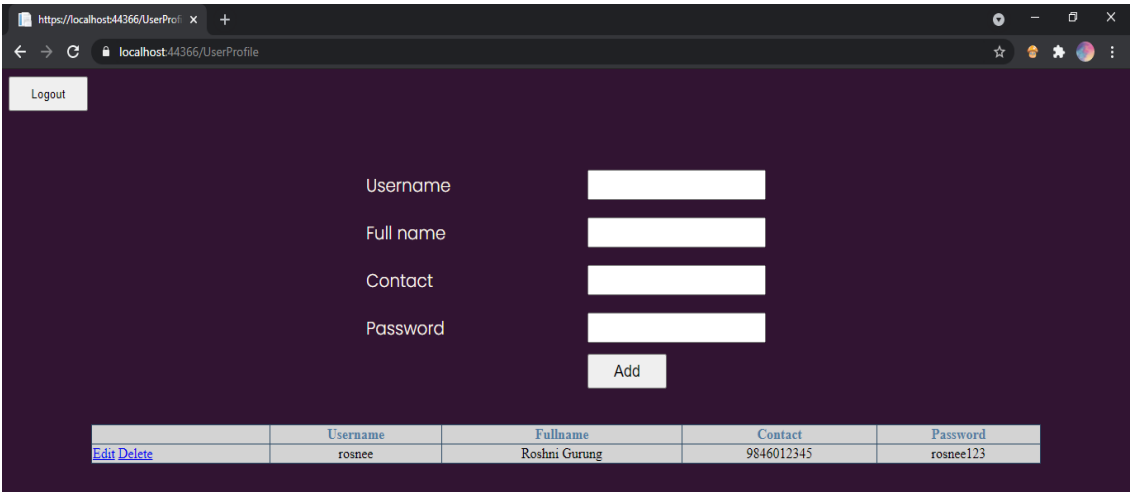


Figure 53: Before editing user details

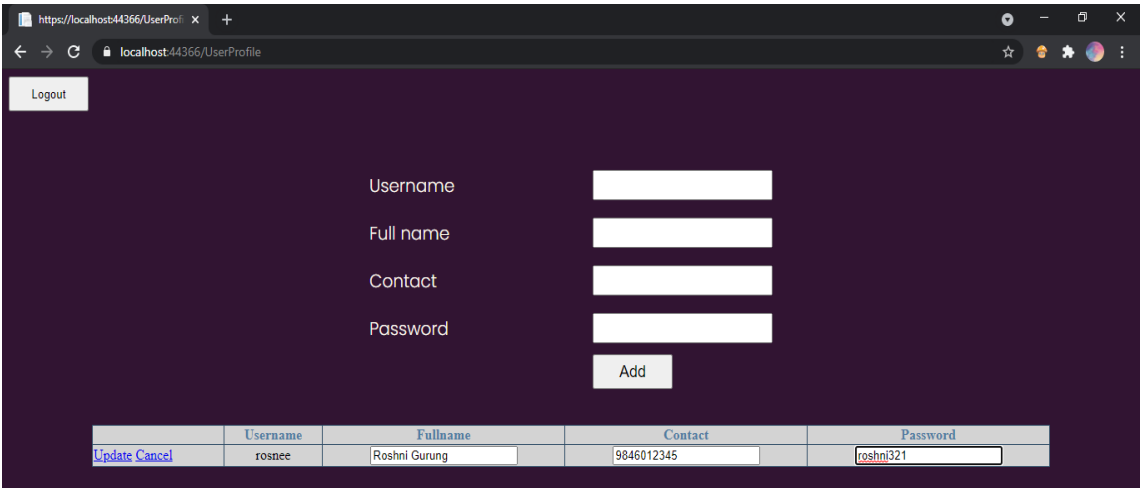


Figure 54: While editing user details

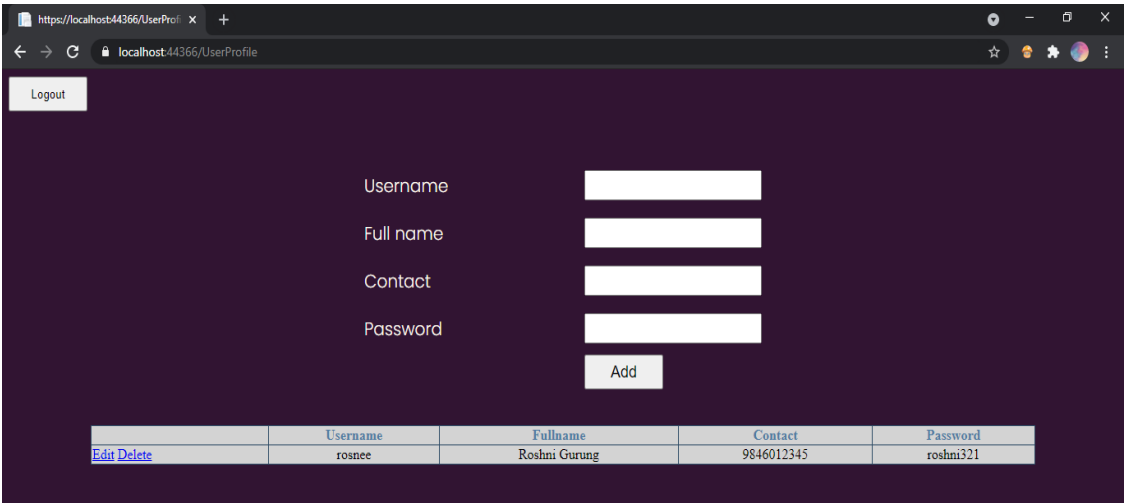


Figure 55: After editing user details

Test case 18:

Objective	To test whether the admin can delete user's account.
Action	Delete user's account.
Expected output	User's account should delete.
Actual output	User's account is deleted.
Result	Test successful.

Table 31: Test Case 18

Logout

Username

Full name

Contact

Password

Add

	Username	Fullname	Contact	Password
Edit Delete	rosna	Roshna Gurung	982716908	rosna123
Edit Delete	rosnee	Roshni Gurung	9846012345	roshni321

Figure 56: Before deleting user

Logout

Username

Full name

Contact

Password

Add

	Username	Fullname	Contact	Password
Edit Delete	rosna	Roshna Gurung	982716908	rosna123

Figure 57: After user deletion

6. Reflection

6.1 Group Experience

This was the group coursework where group of 4 members were involved to complete the given tasks for the module Application development. This was our second coursework of this module. This coursework wasn't easy, there were different obstacles to face various problem like, connecting the database to the server and creating the layout and managing the database. After the completion of the coursework, all the members of the group were able to build up self-confidence and gain various skills to implement functionalities of the program as well as this project has also helped us to sharpen our database skills since the DBMS software such as MS SQL server was entirely new to us. Well, we must admit that we are comfortable with concepts of stored procedures, MS SQL Server, SQL Data modeler and Visual Studio, C#, ASP.NET amongst other technologies and concepts. It is also worth observing that this project has immensely helped each of us to further sharpen our programming and debugging skills. While solving the given questions, various bugs were aroused during the development phase and going through the debugging phase has enhanced our problem-solving skills. Overall, this coursework helped us to work in a group with a co-operative manner.

6.2 Personal Experience

Divya Gurung:

The group coursework of module Application Development is about the development of a web application "Stock Management System" using C# programming language, asp.net as a developer platform, visual studio an integrated development environment and MS SQL as a backend. This coursework is my second project I have ever developed in C# programming language and visual studio but using MS SQL as a backend is first time using it which has made me more familiar with so much tools techniques related with web application development. This coursework has polished my skill on using database queries and connecting databases.

Since, the second coursework of Application Development is a group course, we had four members in our group we had divided our work where I had contribution on both development and document part. I have created flowchart and wrote all the methods' description of the application and worked on login and registered backend part. Although, visual studio, C# and MS SQL were new chapter for us but we have accomplished our work on developing a stock management system after a hard research and group work.

As a Java Developer, I am barely a C# developer but the things I like about C# language is that it supports both reference-type (class) and value-type (struct) user-defined types. Although, I have faced many problems working on MS SQL and C# but with the help of group members the web application has been successfully developed fulfilling all the requirement of this coursework.

Muna Tilija Pun:

The given coursework is about the development of web application for the company, specifically for the stock management. This is our second coursework of the Application Development. Though I wasn't familiar with Microsoft SQL, with the help of module leader and teammates now, I can clearly understand and comfortable to use SQL server. This coursework helped me to increase my knowledge in C#, asp.net framework also helped me to work and learn in a group and gave me more deep knowledge about SQL queries, ASP.NET, C#, visual studio, and other related research skills. Lastly, I would like to thank our module leader Mr. Sabin Shrestha for always being there for guidance, I am very thankful for him to make us understand all the things that we didn't know and I am also thankful to my group member for making this group coursework happen.

Priyanka Gurung:

All the development part of the coursework was done in Visual Studio whereas documentation was done in Word. Visual Studio has everything that any programmer expects from any code editor with a lot of exciting, useful and cool features. It provides numerous development tools to develop different software for any platform like web, desktop. Moreover, it provides rich selection of development languages like C#, COBOL, Visual Basic, JavaScript, C++. Using visual studio was not a new experience to me but it still feels refreshing while using it. It helps me to code more efficiently as it has all the built-in features and libraries inside.

According to the scenario provided in the coursework, the project named as “Beans Stock Management” was built using C# language for the front end whereas SQL for the backend. The overall project helped me to increase the knowledge in ASP.net framework. The main challenging part was to connect the server to the webpage as it is quite new to me. And also, the logics and algorithms for adding the details in the database. However, accepting these challenges and solving them helped me to increase my skill. With the help of my group members and depth research I achieved my goal in the project. Overall, it was an exciting experience to work as a group to develop the project. At last, from connecting the front end to the back-end, handling the errors and building the logics to overcome different difficulties, this coursework helped me to work as a group in a co-operative manner.

Roshna Gurung:

The development part of our coursework was done in Visual Studio. It was easy to use, write code and navigate between classes. With the help of Visual Studio, development of software for web, mobile, desktop and server are easy. It also provides languages like C#, PHP, JavaScript and many more. One of the reasons why I really liked Visual studio is that it supports IntelliSense feature. While writing object name for classes, it gives meaningful suggestions i.e., quite similar to the class name. In our

coursework, coding was done using C# programming language and for backend, SQL was used. C# supports the concepts of inheritance, encapsulation and polymorphism.

Since I knew how to use Visual Studio and had some basic knowledge of C# language, it wasn't hard to start the project. But some of the tasks assigned in the coursework was quite hard for me. But after doing lots of research related to this topic, I cleared my doubts. Also, with the help of my teachers and friends, the problems were tackled. As a whole, it was very interesting and fun to do the project. I learned more about C# and SQL. I hope I was able to meet that objectives and lived up to the standards.