



### Assignment Cover Sheet

<b>Qualification</b>	<b>Module Number and Title</b>	
HND in Computing/ HND in Software Engineering	CSE5013 Service Oriented Computing	
<b>Student Name &amp; No.</b>	<b>Assessor</b>	
M.A. Chamath Shyamal & CL/HDCSE/95/43	Upika Wijeshinge	
<b>Hand out date</b>	<b>Submission Date</b>	
2021-11-09	2022-01-23	
<b>Assessment type</b> Coursework	<b>Duration/Length of Assessment Type</b> Practical project/report	<b>Weighting of Assessment</b> <b>100%</b>

### **Learner declaration**

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

### **Marks Awarded**

First assessor		
IV marks		
Agreed grade		
Signature of the assessor		Date

## **FEEDBACK FORM**

### **INTERNATIONAL COLLEGE OF BUSINESS & TECHNOLOGY**

**Module:** Windows Application Development

**Student:**

**Assessor :**

**Assignment:**

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**

## **Learning outcomes covered**

- Understand Service oriented architecture and patterns.
- Design service application solution by applying SO concepts
- Develop service-oriented application
- Critically evaluate suitable delivery environment and deploy service application.

## **Scenario and the Task**

State Pharmaceutical Cooperation (SPC) is the responsible entity for pharmaceutical drug manufacturing and importing to the national requirement. When SPC want to buy some kind of a drug it will advertise for a supplier and the suppliers should submit a tender proposal. Before applying for the tender supplier must be registered with SPC. Also, SPC has few drugs manufacturing plants and planning to increase the facilities further in future. SPC distribute its drugs via SPC own pharmacies and also link dealer pharmacies. SPC is focus on improving its services and operation with the aid of Information technology and below are some functionalities that is identified by the SPC.

- Supplier registration can be done in any of the main branches and this feature should be added to the existing website as well.
- Manufacturing plants should be able to update stock details of the warehouse. Also the warehouse staff can update stock details based on the externally purchased quantities. Manufacturing plants are looking to add this feature with their current system. Some plants are running on web-based system and others on windows-based systems.
- SPC and registered pharmacies should be able to search for any drug and also to place an order from main warehouse. Linked pharmacy network is using some kind of a software system to manage their sales and inventory currently. This new feature should be link with their systems.

Based on the above mention requirement you need to develop a SOC based solution. Need to implement needed requirements as services and also need to build client application and consume via API.

## **Tasks**

1. Create a presentation explaining how Monolithic architecture and Service oriented architecture model can be used for this case study. Compare and contrast both architectures and justify the best architecture base on the maintainability and scalability. (20 marks)
2. Design and develop suitable application based on SOC. Need to implement the services and also must create client application to consume the services. Should be able to demonstrate and provide all source codes with suitable design diagrams. Need use proper coding standards and must focus on reusability and maintainability of the application. (60 marks)
3. Properly test the developed application should be able to demonstrate debugging process and demonstrate testing results. (10 marks)
4. Explain deployment techniques that are suitable for the developed application (Serve, Docker, Kubernetes etc..). (10 marks)

## **Acknowledgement**

Primarily I thank God for being able to complete this assignment in a successful manner. Thus, I take this opportunity to express my deep sense of gratitude and my profound respect to the lecturer who guided and inspired me in doing this assignment. I had to get the guidance of a respected and responsible person at the preparation time and while continuing the assignment. So, I would like to thank our lecturer Miss. Upika Wijeshinge whose valuable guidelines and consultations been the ones that helped me patch this assignment and make this full proof success and finalize this successfully.

And also, her instructions which were given underlying the structure has served as the major contributor towards in completing this assignment and her instructions about the programming and coding was more helpful in implementing this SOC based application for State Pharmaceutical Cooperation (SPC). Through those and by this golden opportunity, I got the full knowledge on basics and some other advance in developing Service Oriented Computing based applications as well as in C# Programming. I hope this knowledge you gave me will help me in learning more advance on Service Oriented Computing as well as my career. I really thankful for you because of the massive courage you had to teach us.

Then I would like to thank Mr. Chathura, Miss Manoda and Miss Ishani who is with us and help us in many sides since the beginning of the Bridging Program. However finally, I am really grateful because I managed to complete this assignment within the time period given by our lecturer Miss. Upika Wijeshinge. Thank you all!

## Contents

<b>TASK 1- Monolithic Architecture &amp; Service Oriented Architecture .....</b>	<b>10</b>
1.1) Monolithic Architecture .....	10
1.2) Service Oriented Architecture .....	12
1.3) How SOA suits to develop the SPC system.....	14
<b>TASK 2- Design.....</b>	<b>15</b>
2.1) Registration.....	15
2.1.1) Supplier Registration Service .....	15
2.1.2) Get Supplier ID Auto Service.....	16
2.1.3) Supplier Registration Client .....	17
2.1.4) Warehouse Staff Registration Service .....	18
2.1.5) Get Warehouse Staff ID Auto Service .....	19
2.1.6) Warehouse Staff Registration Client .....	20
2.1.7) Pharmacy Staff Registration Service .....	21
2.1.8) Get Pharmacy Staff ID Auto Service .....	22
2.1.9) Pharmacy Staff Registration Client .....	23
2.2) Login Client.....	24
2.3) Stocks.....	25
2.3.1) Add Stock Service .....	25
2.3.2) Get Stock ID Auto Service .....	26
2.3.3) Add Stock & Get Stock ID Auto Client .....	27
2.3.4) Search Stock Service .....	28
2.3.5) Search Stock Client .....	29
2.3.6) Get Supplier ID Client.....	30
2.3.7) Update Stocks Client .....	31
2.4) Place Orders.....	32
2.4.1) Place Order Service .....	32
2.4.2) Get Order ID Auto Service .....	33
2.4.3) Place Order Client .....	34
2.4.4) Search Order Service .....	35
2.4.5) Search Order Client .....	36

2.4.6) Get Pharmacy Staff ID Client.....	37
<b>TASK 3-Testing.....</b>	<b>38</b>
3.1) Test Plan .....	38
3.2) Test Cases .....	47
<b>TASK 4- Deployment Techniques .....</b>	<b>121</b>
4.1) What is Software Deployment.....	121
4.2) Local Deployment & Global Deployment.....	121
4.3) Some Deployment Techniques.....	121
4.3.1) Copying a website .....	121
4.3.2) Docker .....	122
4.3.3) Kubernetes .....	122
4.4) The most suitable Deployment Technique for the SPC System.....	124
4.4.1) Creating a setup project .....	124
<b>References.....</b>	<b>126</b>

## **Introduction**

This Assignment is regarding the Service Oriented Computing Module and this is the eighteenth assignment we got in our HD Program. Basically, this coursework covers understanding of service-oriented architecture/patterns, designing of service application solution by applying SOA concepts, developing of service-oriented application and a critical evaluation for suitable delivery environment and deploy service application. Studying about Service Oriented Computing is more important for a student who hopes to become a Software Engineer and a Web Developer as C# Programming directly explains the basics and some other advance things in building service-oriented architecture-based applications and how complex codes are made using the basic stuffs reused in order to implement a SOA based application with a low effort. In my assignment, I have ideally shown some theories used, designing as well as main coding along with the evidences.

The primary and main objective of implementing this SOA based application for State Pharmaceutical Cooperation (SPC) is to solve many problems they are facing currently. State Pharmaceutical Cooperation (SPC) is the responsible entity for pharmaceutical drug manufacturing and importing to the national requirement. When SPC want to buy some kind of a drug it will advertise for a supplier and the suppliers should submit a tender proposal. Before applying for the tender supplier must be registered with SPC. Also, SPC has few drugs manufacturing plants and planning to increase the facilities further in future. SPC distribute its drugs via SPC own pharmacies and also link dealer pharmacies. Because of the huge workload they have, they have planned to implement an application which can improve its services and operation with the aid of Information Technology.

As this Mobile Application has two different authentication levels as Manager, Warehouse Staff and Pharmacy Staff, will have the functionalities to register & login to the system, add/update/delete orders and stocks, view or search orders and stock. Further, all three user levels have different functionalities in this SOA based Web Application. In order to implement those functions, I had to use services, client applications, C# programming language which is more considerable when using to develop a SOA based Web Application in Visual Studio. Thus, this SOA based SPC Web Application will helpful for that organization to expand their businesses by minimizing the huge workload they're having currently as this Web Application is more user friendly and easier to understand when using. Thus, I hope this will be an ideal and clear assignment.

# **TASK 1- Monolithic Architecture & Service Oriented Architecture**

## **1.1) Monolithic Architecture**

Primarily, Monolithic Architecture is self-contained. Because it contains all parts of applications, packaged and deployed together. Put it simply, a Monolithic Architecture is a large container in which all of an application's software components are built, tightly packed, and delivered as a single entity. Typically, the Monolithic Architecture is often three-tiered, with the presentation layer being the first, the application layer being the second, and the data layer being the third. When it comes to the **presentation layer**, it's basically the front-end layer. Primarily, it is the user interface of a web application which is often built with web technologies such like CSS, HTML and Javascript. The presentation layer deals with application layer and data layer using API calls. **Application layer** is the layer which has all the business logics. It's often written in Java, (dotnet).Net, Python, C++, C# like programming languages. The final layer of the three-tiered architecture is **data layer** where all the data is stored. Those stored data is accessed by the application layer using API calls. Thus, it's obvious that this data layer uses database technologies such as MySQL, Oracle DB, Microsoft SQL Server and Mongo DB. As mentioned above, all the components of an application are developed together which may be interdependent on each other and deployed onto a web server as a single unit whereas all those three layers in three tiered are packaged and deployed. This is one of the main reasons why this architecture is referred to as Monolithic Architecture. (Kalske, 2022) (Haq, 2022)

Let's check some of the benefits and drawbacks of monolithic architecture. When it comes to **advantages**, the key benefit is that it is a single deployment unit. As a result, there is only one application to test and deploy. Therefore, software applications based on Monolithic Architecture are simple to develop and deploy. Monolithic Architecture is mature and old. Thus, IDs and other development tools are designed in join with monolithic architecture. Basically, those are the two most important advantages of monolithic architecture. (integrate.io, 2022)

There are more drawbacks than benefits to monolithic architecture. The first **disadvantage** is applications built with Monolithic Design are huge. If the application is relatively small, this design is ideal since it is easy to create and maintain. When an application is huge and contains many components, developers and programmers may become confused between requirements, components, and code. This will be a significant challenge in this design, as it is more difficult for a developer or a group of developers to understand the entirety of an application. As a result, understanding the application will take a long time, and time will be wasted. The next issue is that Monolithic Architecture-based applications will become locked in with their initial technology decisions. Despite the availability of improved tools, technologies, and programming languages, Monolithic Architecture-based systems must stick to the initial technologies and decisions made at the start. The next problem is that there is no one-size-fits-all approach for designing software applications. Because the Monolithic Design will restrict to use single development stack. For instance, can use either Java or .Net (dotnet). Furthermore, because there are several application components linked together in a monolithic application, frequent deployments are not possible with this architecture. Because badly, it needs the teamwork of a large number of developers and even departments in charge of these components. Also, it may require even hours or days for a proper deployment and test new features and bug fixes. Finally, scaling of monolithic applications can often be very challenging since this is deployed as a single unit.

After the introduction of new issues brought by the Internet, a **key disadvantage** occurred, namely the unpredictability of user demands on the application. That means, when the demand for an application is high, this architecture-based application decreases its performance. Simply, resources cannot be made available rapidly enough to fulfill demand. And it's possible that this will result in a crash. As a result, it's clear that monolithic architecture isn't ideal and doesn't work well with dynamically scaling resources as needed. (Konrad Gos, 2022)

## 1.2) Service Oriented Architecture

Within IT, a new systems design paradigm known as SOA has arisen during the last few decades (Service Oriented Architecture). It's a reaction to the need to create software that works in distributed and different situations, which the Internet has made more common. According to (Kathryn B. Laskey, 2022), Service oriented architecture (SOA) is a paradigm for organizing a set of capabilities, often distributed across the network and possibly under the control of different ownership domains. As a result, it's evident that SOA is an architectural strategy for distributed systems that uses linked services, standard interfaces, and protocols give efficient cross-platform connection and communication over a service bus. It's easy to combine many services using SOA to give the required functionality. Potential users can see services, they can interact with them through a various information exchange, and they can create real-world outcomes. Therefore, applications can be built with an architecture that is oriented around services.

As in (Wilkes, 2022), Service is a component which is capable of performing a task whereas a WSDL service is a collection of end points. A type of capability described using WSDL (CBDI). When it comes to a Web service, it means a software system which is designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a format that machines can process (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with XML serialization in conjunction with other Web-related standards. Thus. it ensures that the web services on a network can interact with each other seamlessly.

In Service Oriented Architecture, there are some principles to be considered. First one is, **Standardized Service Contract** which includes the descriptions about services for the better understanding of client application and the use of services. Second one is **Loose Coupling** which means that the web services and the client accessing the web service should have a low connection as necessary. As a result, if the service feature changed at any particular moment in time, the client application should not be affected. Third one is **Service Abstraction** where a service should not reveal how it performs its functionalities and encapsulation there for that. Fourth principle is

**Service Reusability** which means once a web service's code is built, it should be able to interact with a variety of application types by using the same code again and again. Fifth one is, **Service Autonomy** where the service should have full access for the code it includes since it understands everything about the functionality it provides. Sixth principle is **Service Statelessness** which the logic that services encapsulate should be under their control. This means that information should not be withheld from one state to the next. This would have to be done from either the client application or the server application. The other one is **Service Discoverability** where it is possible to locate services (usually in a service registry). Eighth one is **Service Composability** which says the service can divide into modules, each with its own set of functions. The last one is **Service Interoperability** which uses standards such as XML and HTTP communication are employed in web services to confirm that subscribers follow to the idea. (Walker, 2022)

When it comes to **advantages**, SOA allows to reuse all of the separate services. For instance, if it's needed to create a login authentication module, it's possible to utilize the Facebook/Google authentication service in SOA. Instead of that in Monolithic Architecture, that cannot be done. All modules should start from scratch. In SOA there is a manageable complexity where Monolithic Architecture has a higher complexity. So as in Monolithic Architecture, developers/ programmers won't get lost between requirements, components and code in SOA. Therefore, it won't take so long to understand the application and the time won't be wasted as in Monolithic Architecture. As there are better tools, technologies and programming languages, SOA based applications don't need to stick with the initial technologies and initial decisions which have been taken at the beginning unlike in Monolithic Architecture. Monolithic Design will restrict to use single development stack. But in SOA it's not like that. Because in SOA it's possible to use either Java or .Net (dotnet). Also in Monolithic Architecture, it may require even hours or days for a proper deployment and test new features and bug fixes. But in SOA that much of time won't require for a proper deployment or a for a testing. Scaling of monolithic applications can often be very challenging since this is deployed as a single unit whereas SOA based applications are not deployed as a single unit. After the Internet brought new challenges, Monolithic Architecture based applications reduces the performance of an application when the demand is high which might even cause a crash. But SOA is

ideal and goes well with **scaling** resources dynamically when needed. SOA make a system fault tolerant. If a single service fails, the entire system will not shut down. The most important feature of SOA is service reusability, which means that once a service is constructed, can reuse it in future projects as well. (Bloomberg, 2022)

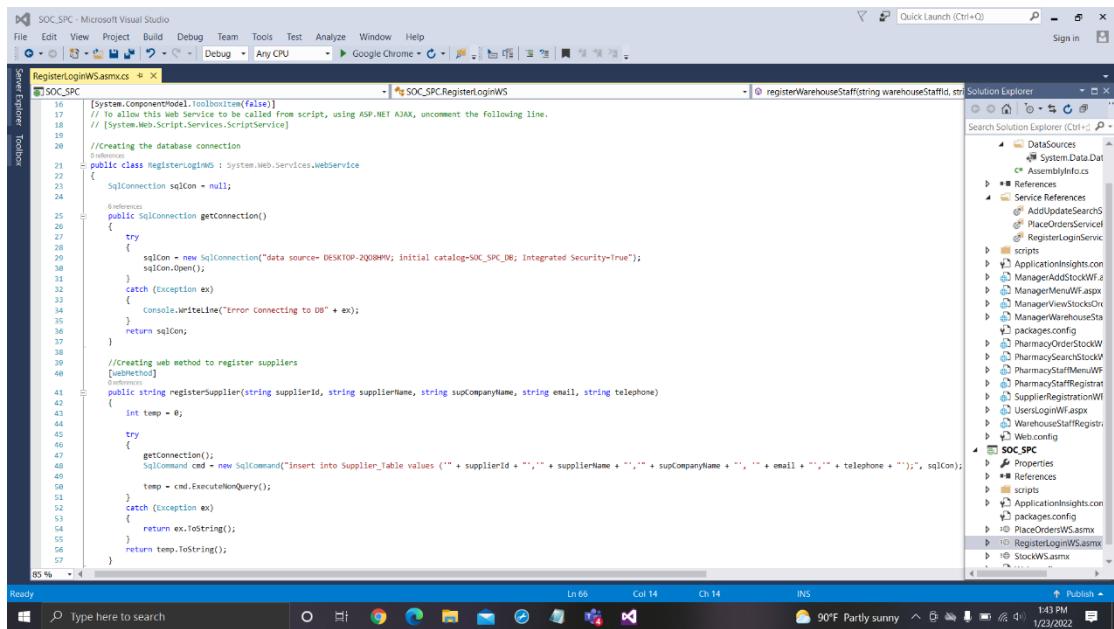
### **1.3) How SOA suits to develop the SPC system**

However, for the system of State Pharmaceutical Cooperation (SPC), Service Oriented Architecture is the best architecture because it allows to reuse all of the separate services. Login interface and some modules can be reused as accessible for user levels. And it doesn't require to start modules from scratch. Further there is a low complexity in SOA and that would be easier for the system of SPC won't get lost while coding. As this SPC system development is a short time development, its wont waste time when work based on SOA. More importantly, in SOA it's possible to use either Java or .Net and that's more reliable. From the deployment side also, SOA is the best as mentioned in the above description. Moreover, because of the reusability, it would be easier and efficient to do the modifications in the future for the SPC system. Finally, one of the main reasons for selecting SOA is if a single service fails, the entire system will not shut down.

# TASK 2- Design

## 2.1) Registration

### 2.1.1) Supplier Registration Service

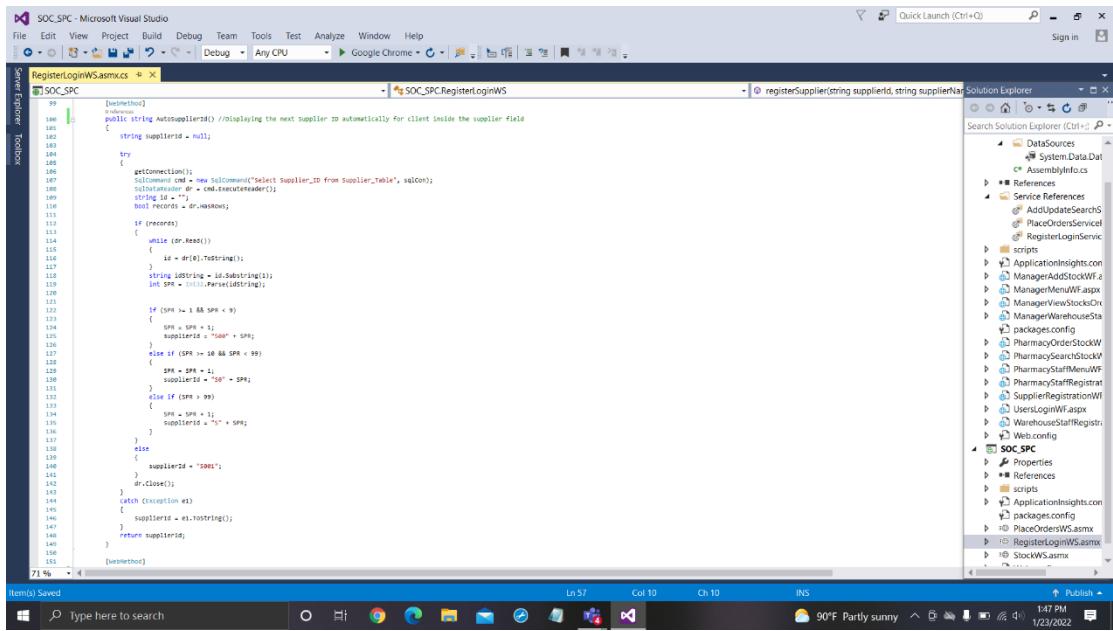


The screenshot shows the Microsoft Visual Studio interface with the project 'SOC\_SPC' open. The code editor displays the 'RegisterLoginWS.asmx.cs' file, which contains C# code for a web service. The code includes database connection logic using SQL Server and defines a web method named 'registerSupplier' that inserts supplier information into a database table. The Solution Explorer on the right shows various files and projects related to the application.

```
16 //System.ComponentModel.ToolboxItem(false)]
17 // To allow this web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
18 // [System.Web.Script.Services.ScriptService]
19
20 //Creating the database connection
21 public class RegisterLoginWS : System.Web.Services.WebService
22 {
23     SqlConnection sqlCon = null;
24
25     //Properties
26     public SqlConnection getConnection()
27     {
28         try
29         {
30             sqlCon = new SqlConnection("data source=DESKTOP-2Q08MM; initial catalog=SOC_SPC_DB; Integrated Security=True");
31             sqlCon.Open();
32         }
33         catch (Exception ex)
34         {
35             Console.WriteLine("Error Connecting to DB" + ex);
36         }
37         return sqlCon;
38     }
39
40     //Creating web method to register suppliers
41     [WebMethod]
42     [Description("")]
43     public string registerSupplier(string supplierId, string supplierName, string supCompanyName, string email, string telephone)
44     {
45         int temp = 0;
46
47         try
48         {
49             getConnection();
50             SqlCommand cmd = new SqlCommand("Insert into Supplier_Table values ('" + supplierId + "','" + supplierName + "','" + supCompanyName + "','" + email + "','" + telephone + "')", sqlCon);
51             temp = cmd.ExecuteNonQuery();
52         }
53         catch (Exception ex)
54         {
55             return ex.ToString();
56         }
57         return temp.ToString();
58     }
59 }
```

In the above screenshot, it shows the web service called “RegisterLoginWS”. All the necessary web methods for supplier, warehouse staff and pharmacy staff registration have been coded inside this web service file. All the coding has done inside the class called RegisterLoginWS. At the beginning, it shows the database connection code within try and catch blocks. After that, the web method for supplier registration is there which includes the coding for the registration of suppliers is visible in the screenshot. The “insert” query has been used to register supplier details.

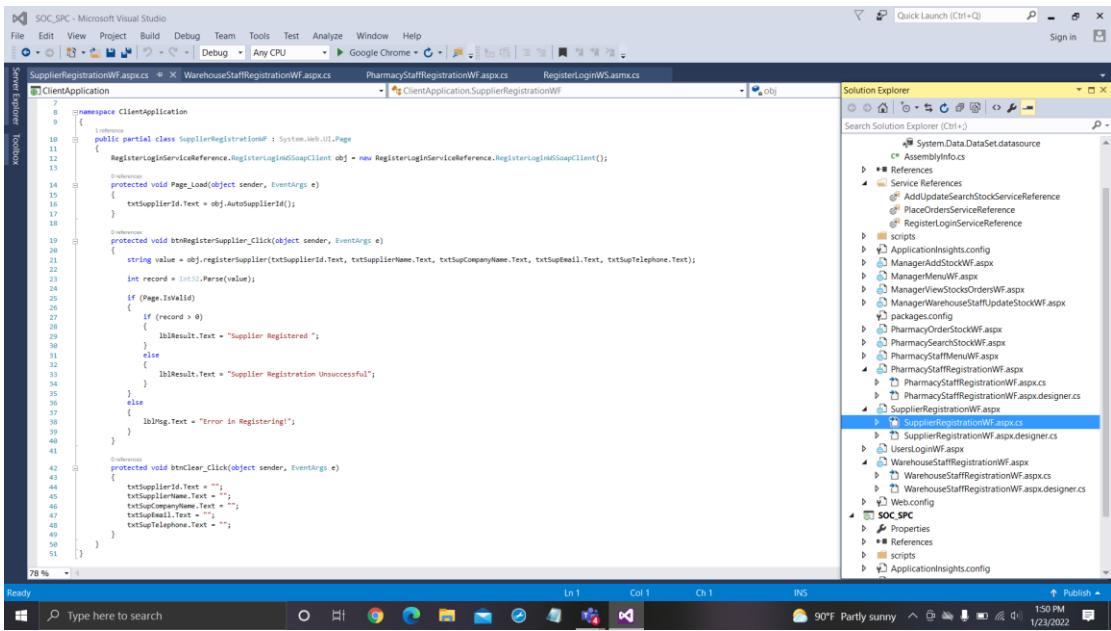
## 2.1.2) Get Supplier ID Auto Service



```
[WebService]
[WebMethod]
public string AutosupplierId() //Displaying the next supplier ID automatically for client inside the supplier field
{
    string Suppliers = null;
    try
    {
        SqlConnection sqlCon = new SqlConnection("select * from Supplier_Table", sqlCon);
        SqlCommand cmd = sqlCon.CreateCommand();
        cmd.CommandText = "cmdexecutescalar";
        string Id = "";
        bool records = dr.HasRows;
        if (records)
        {
            while (dr.Read())
            {
                Id = dr[0].ToString();
            }
            string IDString = id.Substring(1);
            int SPN = int.Parse(IDString);
            SPN++;
            if (SPN >= 1 && SPN < 9)
            {
                SPN = SPN + 1;
                supplierId = "S0" + SPN;
            }
            else if (SPN >= 10 && SPN < 99)
            {
                SPN = SPN + 1;
                supplierId = "S" + SPN;
            }
            else if (SPN >= 99)
            {
                SPN = SPN + 1;
                supplierId = "-" + SPN;
            }
        }
        else
        {
            supplierId = "S001";
        }
        dr.Close();
    }
    catch (Exception e1)
    {
        supplierId = e1.ToString();
    }
    return supplierId;
}
[WebService]
```

In the above evidence, it depicts the coding that has been done for getting the next supplier id automatically when the page is loaded. That also has been done inside a web method. For that “select” query has been used. To do that, while and if...else if...else statements has used. In that, supplier ids start from S1, S2, etc. and then S10, S11, etc. and then S100, S101, etc. And then return the correct supplier id finally.

### 2.1.3) Supplier Registration Client

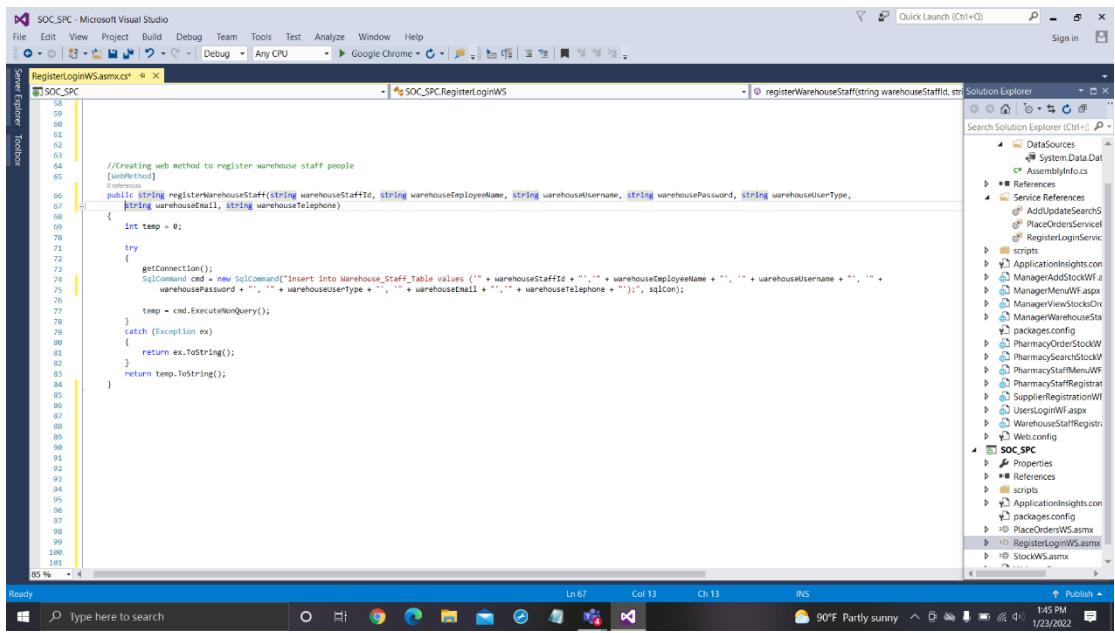


The screenshot shows the Microsoft Visual Studio interface with the project 'SOC\_SPC' open. The 'SupplierRegistrationWF.cs' file is the active code editor. The code implements a web page for supplier registration, utilizing a service reference to call a web method. The Solution Explorer on the right lists various files and references within the project.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ClientApplication;
public partial class SupplierRegistrationWF : System.Web.UI.Page
{
    RegisterLoginServiceReference.RegisterLoginGSSoapClient obj = new RegisterLoginServiceReference.RegisterLoginGSSoapClient();
    protected void Page_Load(object sender, EventArgs e)
    {
        txtSupplierId.Text = obj.AutoSupplierId();
    }
    protected void btnRegisterSupplier_Click(object sender, EventArgs e)
    {
        string value = obj.registerSupplier(txtSupplierId.Text, txtSupplierName.Text, txtSupCompanyName.Text, txtSupEmail.Text, txtSupTelephone.Text);
        int record = Int32.Parse(value);
        if (record > 0)
        {
            lblResult.Text = "Supplier Registered";
        }
        else
        {
            lblResult.Text = "Supplier Registration Unsuccessful";
        }
    }
    protected void btnClear_Click(object sender, EventArgs e)
    {
        txtSupplierId.Text = "";
        txtSupplierName.Text = "";
        txtSupCompanyName.Text = "";
        txtSupEmail.Text = "";
        txtSupTelephone.Text = "";
    }
}
```

As in the above attach screenshot, all the coding has been done inside a class called “SupplierRegistrationWF”. Then an object has created to call the web method relevant to that. Inside page load, the particular web method is called in order to get the next supplier id automatically when the page is loading. Its visible in the screenshot that the coding for supplier registration has been done by calling the method for supplier registration in the web method.

## 2.1.4) Warehouse Staff Registration Service



```
58
59
60
61
62
63
64
65
66 //Creating web method to register warehouse staff people
67 [WebMethod]
68 public string registerWarehouseStaff(string warehouseStaffId, string warehouseEmployeeName, string warehouseUsername, string warehousePassword, string warehouseUserType,
69         string warehouseEmail, string warehouseTelephone)
70 {
71     int temp = 0;
72
73     try
74     {
75         getConnection();
76         SqlCommand cmd = new SqlCommand("Insert Into Warehouse_Staff_Table values ('" + warehouseStaffId + "','" + warehouseEmployeeName + "','" + warehouseUsername + "','" + warehousePassword + "','" + warehouseUserType + "','" + warehouseEmail + "','" + warehouseTelephone + "')", saicon);
77
78         temp = cmd.ExecuteNonQuery();
79     }
80     catch (Exception ex)
81     {
82         return ex.ToString();
83     }
84     return temp.ToString();
85 }
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101 }
```

In the above screenshot, all the coding for warehouse staff registration also has been done inside the class called RegisterLoginWS. Thus, the web method for warehouse staff registration is there which includes the coding for the registration of warehouse staff is visible in the screenshot. The “insert” query has been used to register warehouse staff details.

## 2.1.5) Get Warehouse Staff ID Auto Service

```
148     }
149 
150     [webmethod]
151     public string AutowarehousestaffId() //Displaying the next warehouse staff ID automatically for client inside the warehouse Staff Id field
152     {
153         string warehousestaffId = null;
154 
155         try
156         {
157             getConnection();
158             SqlCommand cmd = new SqlCommand("Select warehouse_staff_ID from warehouse_staff_Mst", sqlcon);
159             SqlDataReader dr = cmd.ExecuteReader();
160             string idString = dr.ReadString();
161             bool records = dr.HasRows;
162 
163             if (records)
164             {
165                 while (dr.Read())
166                 {
167                     id = dr[0].ToString();
168                     idString = id.Substring(1);
169                     int WSR = int.Parse(idString);
170 
171                     if (WSR > 1 && WSR < 9)
172                     {
173                         WSR = WSR + 1;
174                         warehousestaffId = "W" + WSR;
175                     }
176                     else if (WSR >= 10 && WSR < 99)
177                     {
178                         WSR = WSR + 1;
179                         warehousestaffId = "W" + WSR;
180                     }
181                     else if (WSR > 99)
182                     {
183                         WSR = WSR + 1;
184                         warehousestaffId = "W" + WSR;
185                     }
186                 }
187             }
188             else
189             {
190                 warehousestaffId = "W001";
191             }
192             dr.Close();
193         }
194         catch (Exception e1)
195         {
196             warehousestaffId = e1.ToString();
197         }
198         return warehousestaffId;
199     }
200 }
```

In the above evidence, it depicts the coding that has been done for getting the next Warehouse Staff id automatically when the page is loaded. That also has been done inside a web method. For that “select” query has been used. To do that, while and if...else if...else statements has used. In that, Warehouse Staff ids start from W1, W2, etc. and then W10, W11, etc. and W100, W101, etc. And then return the correct Warehouse Staff id finally.

## 2.1.6) Warehouse Staff Registration Client

```

1 //namespace ClientApplication
2 {
3     public partial class WarehouseStaffRegistrationWF : System.Web.UI.Page
4     {
5         RegisterLoginServiceReference.RegisterLoginSoapClient obj = new RegisterLoginServiceReference.RegisterLoginSoapClient();
6
7         protected void Page_Load(object sender, EventArgs e)
8         {
9             txtWarehouseStaffId.Text = obj.AutoCompleteWarehouseStaffId();
10        }
11
12        protected void btnRegisterWarehouseStaff_Click(object sender, EventArgs e)
13        {
14            string value = obj.registerWarehouseStaff(txtWarehouseStaffId.Text, txtWarehouseEmployeeName.Text, txtWarehouseUsername.Text, txtWarehousePassword.Text, txtWarehouseUserType.Text,
15            txtWarehouseEmail.Text, txtWarehouseTelephone.Text);
16            int record = int.Parse(value);
17
18            if (Page.IsValid)
19            {
20                if (record > 0)
21                {
22                    lblResult.Text = "Warehouse Employee Registered";
23                }
24                else
25                {
26                    lblResult.Text = "Warehouse Employee Registration Unsuccessful";
27                }
28            }
29            else
30            {
31                lblMsg.Text = "Error in Registering!";
32            }
33        }
34
35        protected void btnClear_Click(object sender, EventArgs e)
36        {
37            txtWarehouseStaffId.Text = "";
38            txtWarehouseEmployeeName.Text = "";
39            txtWarehouseUsername.Text = "";
40            txtWarehousePassword.Text = "";
41            txtWarehouseConPassword.Text = "";
42            txtWarehouseEmail.Text = "";
43            txtWarehouseTelephone.Text = "";
44        }
45    }
46}

```

As in the above attach screenshot, all the coding has been done inside a class called “WarehouseStaffRegistrationWF”. Then an object has created to call the web method relevant to that. Inside page load, the particular web method is called in order to get the next Warehouse Staff Registration id automatically when the page is loading. Its visible in the screenshot that the coding for Warehouse Staff Registration has been done by calling the method for Warehouse Staff Registration in the web method.

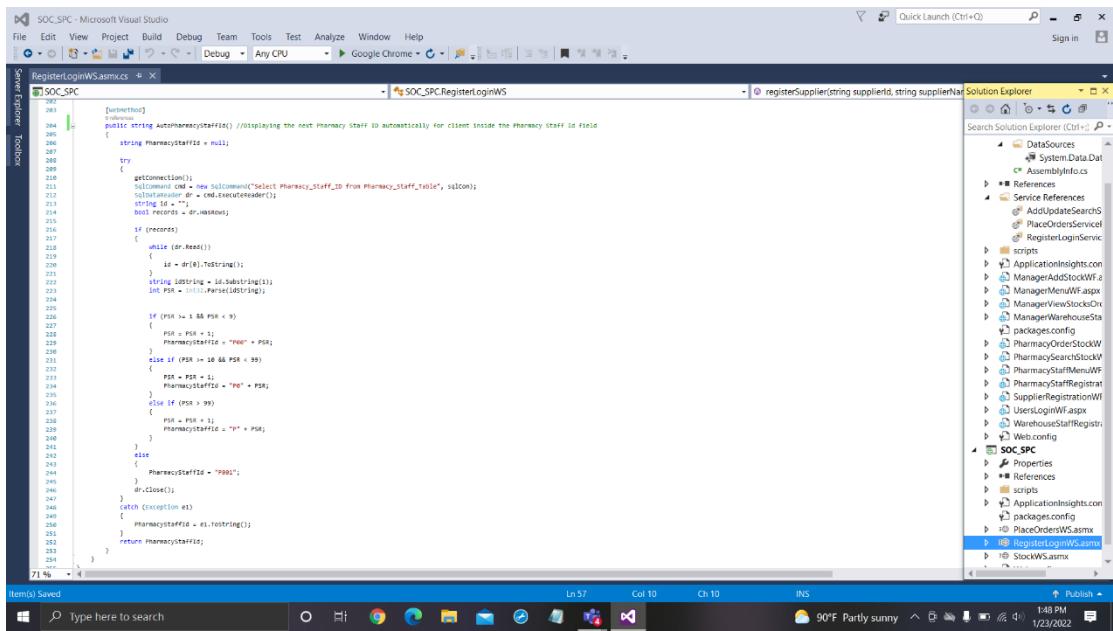
## 2.1.7) Pharmacy Staff Registration Service

The screenshot shows the Microsoft Visual Studio interface with the project 'SOC\_SPC' open. The code editor displays the 'RegisterLoginWS.asmx.cs' file, which contains C# code for a web service. The code includes a web method named 'registerPharmacyStaff' that performs an insert operation into a database table called 'Pharmacy\_Staff\_Table'. The method takes several parameters: 'pharmacyStaffId', 'pharmacyEmployeeName', 'pharmacyUsername', 'pharmacyPassword', 'pharmacyUserType', 'pharmacyName', 'pharmacyBranch', 'pharmacyEmail', and 'pharmacyTelephone'. The code uses a SqlConnection object and a SqlCommand object to execute the insert query. A try-catch block is used to handle any exceptions that may occur during the execution of the query. The Solution Explorer on the right side of the interface shows various files and projects related to the application.

```
85
86
87
88
89
90
91
92 [WebMethod]
93 [RequestWrapper("")]
94 public string registerPharmacyStaff(string pharmacyStaffId, string pharmacyEmployeeName, string pharmacyUsername, string pharmacyPassword, string pharmacyUserType, string pharmacyName,
95 string pharmacyBranch, string pharmacyEmail, string pharmacyTelephone)
96 {
97     int temp = 0;
98
99     try
100     {
101         getConnection();
102         SqlCommand cmd = new SqlCommand("Insert Into Pharmacy_Staff_Table values ('" + pharmacyStaffId + "','" + pharmacyEmployeeName + "','" + pharmacyUsername + "','" + pharmacyPassword +
103             "','" + pharmacyUserType + "','" + pharmacyName + "','" + pharmacyBranch + "','" + pharmacyEmail + "','" + pharmacyTelephone + "')", sqlCon);
104
105         temp = cmd.ExecuteNonQuery();
106     }
107     catch (Exception ex)
108     {
109         return ex.ToString();
110     }
111     return temp.ToString();
112 }
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128 }
```

In the above screenshot, all the coding for pharmacy staff registration also has been done inside the class called RegisterLoginWS. Thus, the web method for pharmacy staff registration is there which includes the coding for the registration of pharmacy staff is visible in the screenshot. The “insert” query has been used to register pharmacy staff details.

## 2.1.8) Get Pharmacy Staff ID Auto Service



```
203     [WebMethod]
204     [RequestFilter("registerSupplier")]
205     public string AutoPharmacyStaffId() //displaying the next Pharmacy Staff ID automatically for client inside the Pharmacy Staff Id field
206     {
207         string PharmacyStaffId = null;
208
209         try
210         {
211             SqlCommand cmd = new SqlCommand("select Pharmacy_Staff_ID from Pharmacy_Staff_Table", scon);
212             SqlDataReader dr = cmd.ExecuteReader();
213             string listing = dr.ReadString();
214             bool records = dr.HasRows;
215
216             if (records)
217             {
218                 while (dr.Read())
219                 {
220                     id = dr[0].ToString();
221                 }
222                 string listing1 = id.Substring(1);
223                 int PSR = int.Parse(listing1);
224
225                 if (PSR >= 1 && PSR < 9)
226                 {
227                     PSR = PSR + 1;
228                     PharmacyStaffId = "P00" + PSR;
229                 }
230                 else if (PSR >= 10 && PSR < 99)
231                 {
232                     PSR = PSR + 1;
233                     PharmacyStaffId = "P0" + PSR;
234                 }
235                 else if (PSR > 99)
236                 {
237                     PSR = PSR + 1;
238                     PharmacyStaffId = "P" + PSR;
239                 }
240             }
241             else
242             {
243                 PharmacyStaffId = "P000";
244             }
245         }
246         dr.Close();
247         catch (Exception e1)
248         {
249             PharmacyStaffId = e1.ToString();
250         }
251         return PharmacyStaffId;
252     }
253 }
```

In the above evidence, it depicts the coding that has been done for getting the next Pharmacy Staff id automatically when the page is loaded. That also has been done inside a web method. For that “select” query has been used. To do that, while and if...else if...else statements has used. In that, Pharmacy Staff ids start from P1, P2, etc. and then P10, P11, etc. and P100, P101, etc. And then return the correct Pharmacy Staff id finally.

## 2.1.9) Pharmacy Staff Registration Client

```

namespace ClientApplication
{
    public partial class PharmacyStaffRegistrationWF : System.Web.UI.Page
    {
        RegisterLoginServiceReference.RegisterInSoapClient obj = new RegisterLoginServiceReference.RegisterInSoapClient();

        protected void Page_Load(object sender, EventArgs e)
        {
            txtPharmacyStaffId.Text = obj.AutoPharmacyStaffID();
        }

        protected void btnRegisterPharmacyStaff_Click(object sender, EventArgs e)
        {
            string value = obj.registerPharmacyStaff(txtPharmacyStaffId.Text, txtPharmacyEmployeeName.Text, txtPharmacyUsername.Text, txtPharmacyPassword.Text, dlPharmacyUserType.Text,
                txtPharmacyName.Text, txtBranch.Text, txtPharmacyEmail.Text, txtPharmacyTelephone.Text);
            int record = Int32.Parse(value);

            if (Page.IsValid)
            {
                if (record > 0)
                {
                    lblResult.Text = "Pharmacy Employee Registered";
                }
                else
                {
                    lblMsg.Text = "Error in Registering!";
                }
            }
        }

        protected void btnClear_Click(object sender, EventArgs e)
        {
            txtPharmacyStaffId.Text = "";
            txtPharmacyEmployeeName.Text = "";
            txtPharmacyUsername.Text = "";
            txtPharmacyPassword.Text = "";
            txtPharmacyUserType.Text = "";
            txtPharmacyName.Text = "";
            txtBranch.Text = "";
            txtPharmacyEmail.Text = "";
            txtPharmacyTelephone.Text = "";
        }
    }
}

```

As in the above attach screenshot, all the coding has been done inside a class called “PharmacyStaffRegistrationWF”. Then an object has created to call the web method relevant to that. Inside page load, the particular web method is called in order to get the next Pharmacy Staff id automatically when the page is loading. Its visible in the screenshot that the coding for Pharmacy Staff Registration has been done by calling the method for Pharmacy Staff Registration in the web method.

## 2.2) Login Client

```

public partial class UserLoginWF : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            SqlConnection sqlCon = new SqlConnection("Data Source=DESKTOP-20204WV;Initial Catalog=SOC_SP_C;Integrated Security=True");
            sqlCon.Open();
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error connecting to DB: " + ex);
        }
    }

    protected void btnLogin_Click(object sender, EventArgs e)
    {
        if (drUserType.SelectedValue == "Manager")
        {
            if (txtUsername.Text == "Manager" & txtPassword.Text == "Manager123")
            {
                Response.Redirect("ManagerHomeWF.aspx");
            }
            else
            {
                lblMsg.Text = "Invalid login credentials! Try Again";
            }
        }
        else if (drUserType.SelectedValue == "Warehouse Staff")
        {
            string warehouseName = txtUsername.Text;
            string warehousePass = txtPassword.Text;
            string query = "Select * from Warehouse_Staff_Table where Username=" + warehouseName + " and Password=" + warehousePass + "";
            SqlCommand cmd = new SqlCommand(query, sqlCon);
            SqlDataReader sdr = cmd.ExecuteReader();
            if (sdr.Read())
            {
                Response.Redirect("WarehouseStaffMenuWF.aspx");
            }
            else
            {
                lblMsg.Text = "Invalid login credentials! Try Again";
            }
            sdr.Close();
        }
        else if (drUserType.SelectedValue == "Pharmacy Staff")
        {
            string pharmacyName = txtUsername.Text;
            string pharmacyPass = txtPassword.Text;
            string query = "Select * from Pharmacy_Staff_Table where Username=" + pharmacyName + " and Password=" + pharmacyPass + "";
            SqlCommand cmd = new SqlCommand(query, sqlCon);
            SqlDataReader sdr = cmd.ExecuteReader();
            if (sdr.Read())
            {
                Response.Redirect("PharmacyStaffMenuWF.aspx");
            }
            else
            {
                lblMsg.Text = "Invalid login credentials! Try Again";
            }
            sdr.Close();
        }
        else
        {
            lblMsg.Text = "Error in login!";
        }
    }
}

```

For all three users, there's only one login interface which is based on the user type. Thus, there's no any web service or web method used for login. However as in the above screenshot, database connection for login has been done inside the page load. Then inside the on click of the login button, coding needed for users to login based on user type has written inside if...if...else...else blocks. As the above screenshot depicts, manager's login has been hardcoded as there's only one manager given in the scenario. Then for the warehouse staff login coding is there. For that “select” query has been used.

```

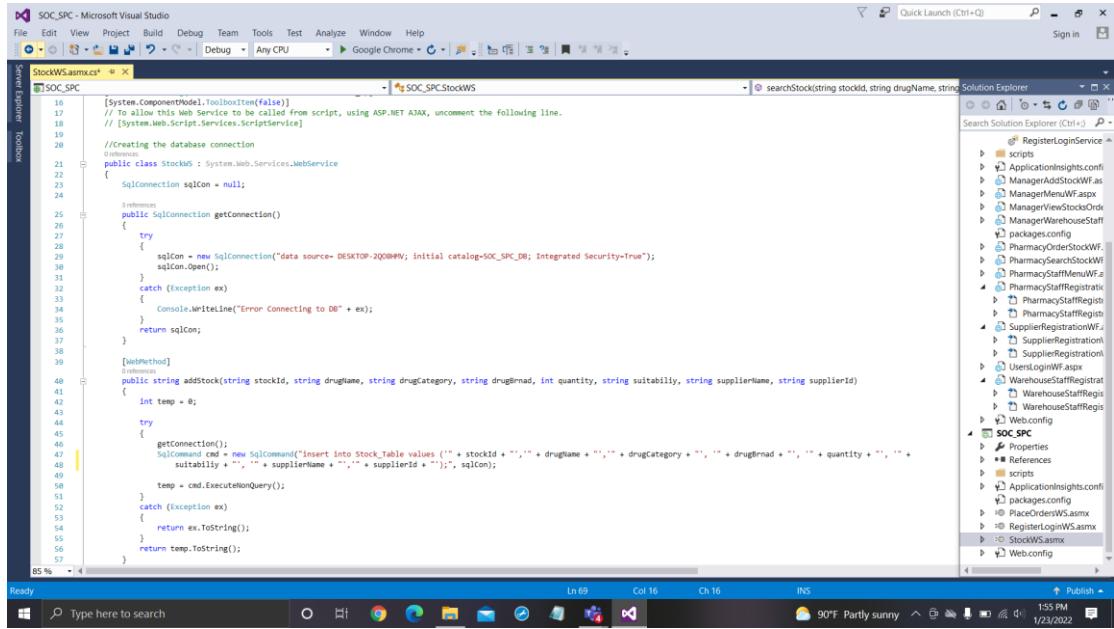
else if (drUserType.SelectedValue == "Pharmacy Staff")
{
    string pharmacyName = txtUsername.Text;
    string pharmacyPass = txtPassword.Text;
    string query = "Select * from Pharmacy_Staff_Table where Username=" + pharmacyName + " and Password=" + pharmacyPass + "";
    SqlCommand cmd = new SqlCommand(query, sqlCon);
    SqlDataReader sdr = cmd.ExecuteReader();
    if (sdr.Read())
    {
        Response.Redirect("PharmacyStaffMenuWF.aspx");
    }
    else
    {
        lblMsg.Text = "Invalid login credentials! Try Again";
    }
    sdr.Close();
}
else
{
    lblMsg.Text = "Error in login!";
}

```

As the above screenshot depicts, for the pharmacy staff login coding is there. For that also “select” query has been used.

## 2.3) Stocks

### 2.3.1) Add Stock Service



The screenshot shows the Microsoft Visual Studio interface with the StockWS.asmx.cs file open in the editor. The code implements a web service class named StockWS that connects to a database and provides methods for adding stocks and searching for them.

```
[System.ComponentModel.ToolboxItem(false)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
// [System.Web.Script.Services.ScriptService]

public class StockWS : System.Web.Services.WebService
{
    SqlConnection sqlCon = null;

    public SqlConnection getConnection()
    {
        try
        {
            sqlCon = new SqlConnection("data source= DESKTOP-JQ0M9MV; initial catalog=SOC_SPC_DB; Integrated Security=True");
            sqlCon.Open();
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error Connecting to DB" + ex);
        }
        return sqlCon;
    }

    [WebMethod]
    public string addStock(string stockId, string drugName, string drugCategory, string drugBrand, int quantity, string suitability, string supplierName, string supplierId)
    {
        int temp = 0;
        try
        {
            getConnection();
            SqlCommand cmd = new SqlCommand("Insert into Stock_Table values ('" + stockId + "','" + drugName + "','" + drugCategory + "','" + drugBrand + "','" + quantity + "','" + suitability + "','" + supplierName + "','" + supplierId + "')", sqlCon);
            temp = cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            return ex.ToString();
        }
        return temp.ToString();
    }
}
```

In the above screenshot, it shows the web service called “StockWS”. All the necessary web methods for add stocks, get supplier id auto and search stock have been coded inside this web service file. All the coding has done inside the class called StockWS. At the beginning, it shows the database connection code withing try and catch blocks. After that, the web method for adding stocks is there which includes the coding for the adding of stocks is visible in the screenshot. The “insert” query has been used to add new stock details.

### 2.3.2) Get Stock ID Auto Service

The screenshot shows the Microsoft Visual Studio interface with the project 'SOC\_SPC' open. The code editor displays the 'StockWS.asmx.cs' file, which contains a single method named 'AutoStockId'. This method uses a SQL query to select the next available Stock\_ID from the 'Stock\_Table' database. The code includes logic to handle the result set and generate a unique stock ID based on the value. The Solution Explorer on the right shows various files and projects related to the application.

```
79     [WebMethod]
80     public string AutoStockId() //Displaying the next category ID automatically for client inside the category ID field
81     {
82         string stockid = null;
83
84         try
85         {
86             getconnection();
87             SqlCommand cmd = new SqlCommand("select Stock_ID from Stock_Table", sqlCom);
88             SqlDataReader dr = cmd.ExecuteReader();
89             string id = "";
90             bool records = dr.HasRows;
91
92             if (records)
93             {
94                 while (dr.Read())
95                 {
96                     id = dr[0].ToString();
97
98                     string laststring = id.Substring(2);
99                     int STK = int.Parse(laststring);
100
101                     if (STK >= 1 && STK < 9)
102                     {
103                         STK = STK + 1;
104                         stockid = "ST0" + STK;
105                     }
106                     else if (STK >= 10 && STK < 99)
107                     {
108                         STK = STK + 1;
109                         stockid = "ST" + STK;
110                     }
111                     else if (STK > 99)
112                     {
113                         STK = STK + 1;
114                         stockid = "S" + STK;
115                     }
116                 }
117             }
118             else
119                 stockid = "ST001";
120
121             dr.Close();
122         }
123         catch (Exception e2)
124         {
125             stockid = e2.ToString();
126         }
127         return stockid;
128     }
129 }
```

In the above evidence, it depicts the coding that has been done for getting the next Stock id automatically when the page is loaded. That also has been done inside a web method. For that “select” query has been used. To do that, while and if...else if...else statements has used. In that, Stock ids start from ST1, ST2, etc. and then ST10, ST11, etc. and then ST100, ST101, etc. And then return the correct Stock id finally.

### 2.3.3) Add Stock & Get Stock ID Auto Client

SOC\_SP\_C - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

ManagerAddStockF.aspx.cs StockWS.asmx.cs

ClientApplication

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data.SqlClient;
4  using System.Linq;
5  using System.Web;
6  using System.Web.UI;
7  using System.Web.UI.WebControls;
8
9  namespace ClientApplication
10 {
11     public partial class ManagerAddUpdateStockWF : System.Web.UI.Page
12     {
13         protected void Page_Load(object sender, EventArgs e)
14         {
15             txtStockId.Text = obj.AutoStockId();
16         }
17     }
18 }
19
20 protected void btnAddStock_Click(object sender, EventArgs e)
21 {
22     string value = obj.addStock(txtStockId.Text, txtDrugName.Text, txtDrugCategory.Text, txtDrugBrand.Text, Convert.ToInt32(txtQuantity.Text), txtSuitability.Text,
23     txtSupplierName.Text, txtSupplierID.Text);
24     int record = int.Parse(value);
25
26     if (Page.IsValid)
27     {
28         if (record > 0)
29         {
30             lblResult.Text = "Stock Added";
31         }
32         else
33         {
34             lblResult.Text = "Stock Adding Unsuccessful";
35         }
36     }
37     else
38     {
39         lblMsg.Text = "Error in Adding!";
40     }
41 }
42 }
```

Quick Launch (Ctrl+Q) Sign in

ManagerAddStockF.aspx.cs StockWS.asmx.cs

ClientApplication

btnClear\_Click(object sender, EventArgs e)

Search Solution Explorer (Ctrl+F)

DataSource

- System.Data.DataSet.datasource
- AssemblyInfo.cs

References

- Service References
  - AddUpdateStockServiceReference
  - PlaceOrdersServiceReference
  - RegisterLoginServiceReference
- scripts
- ManagerAddStockWs.config
- ManagerAddStockWs.aspx
- ManagerAddStockWs.aspx.cs
- ManagerAddStockWs.aspx.designer.cs
- ManagerMenuWs.aspx
- ManagerViewStockOrderWF.aspx
- ManagerWarehouseStaffUpdateStockWF.aspx
- packages.config
- PharmacyOrderStockWF.aspx
- PharmacySearchStockWF.aspx
- PharmacyStaffMenuWF.aspx
- PharmacyStaffRegistrationWF.aspx
- SupplierRegistrationWF.aspx
- UsersLoginWF.aspx
- WarehouseStaffRegistrationWF.aspx
- Web.config

SOC\_SP\_C

- Properties
- References
- scripts
- ApplicationInsights.config
- packages.config
- PlaceOrdersWS.asmx

Ready

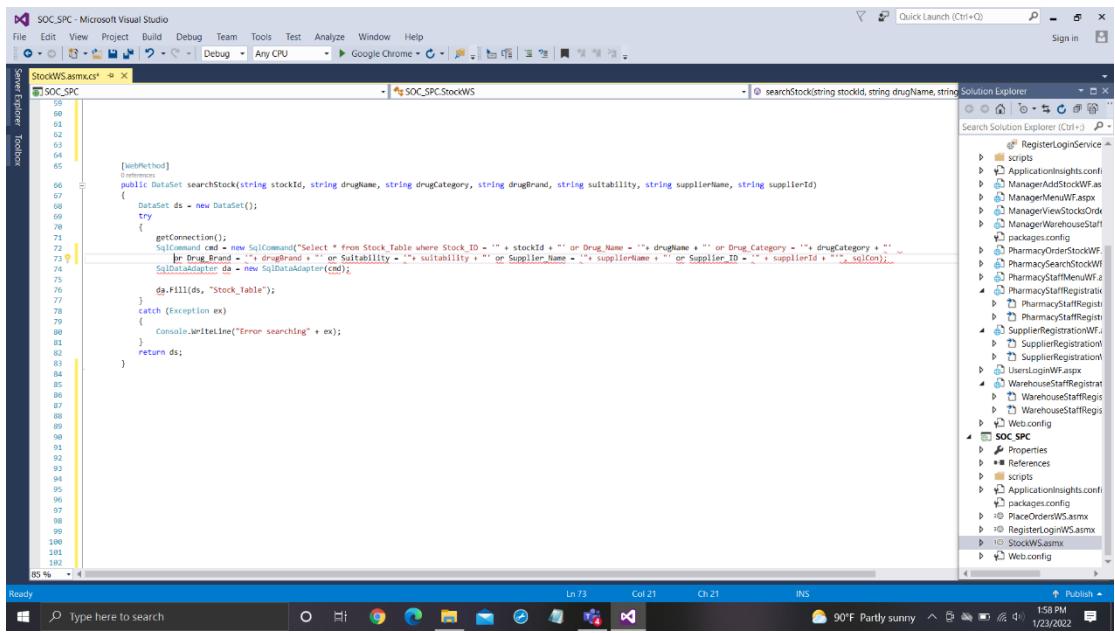
85 %

In 48 Col 36 Ch 36 INS

90°F Mostly cloudy 10:23 PM 1/23/2022

As in the above attach screenshot, all the coding has been done inside a class called “ManagerAddUpdateStockWF”. Then an object has created to call the web method relevant to that. Inside page load, the particular web method is called in order to get the next Stock id automatically when the page is loading. Its visible in the screenshot that the coding for Adding Stock has been done by calling the method for Add Stock in the web method.

### 2.3.4) Search Stock Service

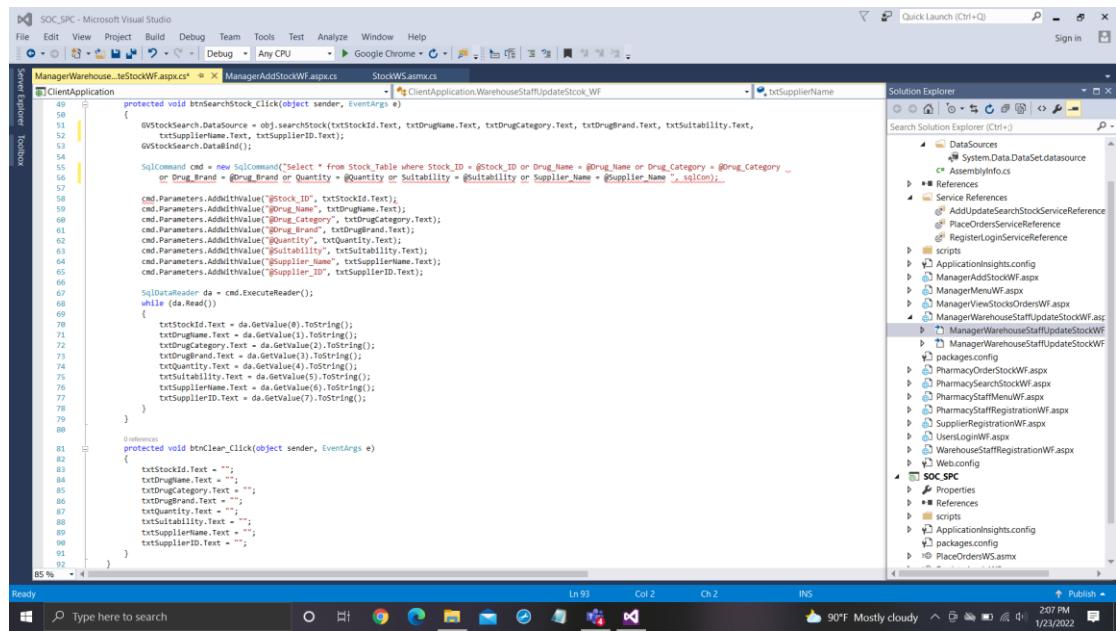


The screenshot shows the Microsoft Visual Studio interface with the project "SOC\_SPC" open. The code editor displays the file "StockWS.asmx.cs". The "StockWS" class contains a single method, "searchStock", which takes five parameters: string stockId, string drugName, string drugCategory, string drugBrand, and string suitability. The method uses a try-catch block to execute a SQL query against a "Stock Table" to find records matching the provided criteria. The catch block handles any exceptions and outputs an error message to the console. The Solution Explorer on the right shows various files and projects related to the application.

```
[WebService]
[WebMethod]
public DataSet searchStock(string stockId, string drugName, string drugCategory, string drugBrand, string suitability)
{
    DataSet ds = new DataSet();
    try
    {
        getconnection();
        SqlCommand cmd = new SqlCommand("Select * from Stock Table where Stock_ID = '" + stockId + "' or Drug_Name = '" + drugName + "' or Drug_Category = '" + drugCategory + "' or Drug_Brand = '" + drugBrand + "' or Suitability = '" + suitability + "' or Supplier_Name = '" + supplierName + "' or Supplier_ID = '" + supplierId + "'", con);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(ds, "Stock_Table");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error searching" + ex);
    }
    return ds;
}
```

This screenshot shows the web method which has been created for searching stock. This also has written inside the web service called “StcokWS”. In here, the “select” query has been used in order to search from the stock database table.

### 2.3.5) Search Stock Client



```

protected void btnSearchStock_Click(object sender, EventArgs e)
{
    GvStockSearch.DataSource = obj.searchStock(txtStockId.Text, txtDrugName.Text, txtDrugCategory.Text, txtDrugBrand.Text, txtSuitability.Text,
        txtSupplierName.Text, txtSupplierID.Text);
    GvStockSearch.DataBind();
}

SqlCommand cmd = new SqlCommand("Select * from Stock_Table where Stock_ID = @Stock_ID or Drug_Name = @Drug_Name or Drug_Category = @Drug_Category or
    Drug_Brand = @Drug_Brand or Quantity = @Quantity or Suitability = @Suitability or Supplier_Name = @Supplier_Name", sqlCom);

cmd.Parameters.AddWithValue("@Stock_ID", txtStockId.Text);
cmd.Parameters.AddWithValue("@Drug_Name", txtDrugName.Text);
cmd.Parameters.AddWithValue("@Drug_Category", txtDrugCategory.Text);
cmd.Parameters.AddWithValue("@Drug_Brand", txtDrugBrand.Text);
cmd.Parameters.AddWithValue("@Quantity", txtQuantity.Text);
cmd.Parameters.AddWithValue("@Suitability", txtSuitability.Text);
cmd.Parameters.AddWithValue("@Supplier_Name", txtSupplierName.Text);
cmd.Parameters.AddWithValue("@Supplier_ID", txtSupplierID.Text);

SqlDataReader da = cmd.ExecuteReader();
while (da.Read())
{
    txtStockId.Text = da.GetValue(0).ToString();
    txtDrugName.Text = da.GetValue(1).ToString();
    txtDrugCategory.Text = da.GetValue(2).ToString();
    txtDrugBrand.Text = da.GetValue(3).ToString();
    txtQuantity.Text = da.GetValue(4).ToString();
    txtSuitability.Text = da.GetValue(5).ToString();
    txtSupplierName.Text = da.GetValue(6).ToString();
    txtSupplierID.Text = da.GetValue(7).ToString();
}
}

protected void btnClear_Click(object sender, EventArgs e)
{
    txtStockId.Text = "";
    txtDrugName.Text = "";
    txtDrugCategory.Text = "";
    txtDrugBrand.Text = "";
    txtQuantity.Text = "";
    txtSuitability.Text = "";
    txtSupplierName.Text = "";
    txtSupplierID.Text = "";
}

```

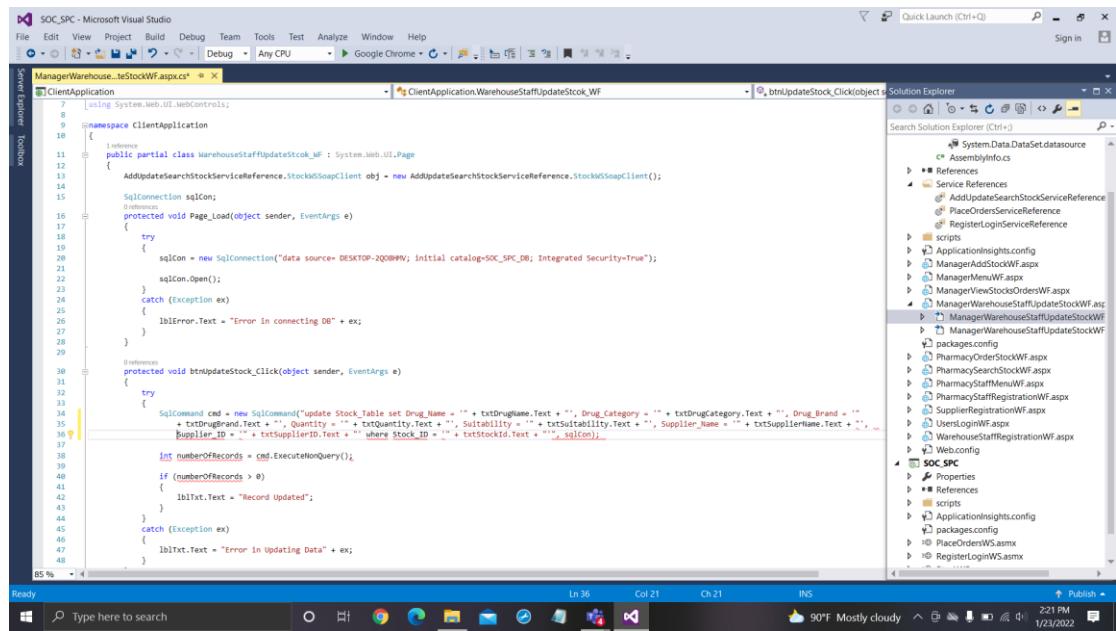
As displayed in the attached screenshot, the web method is called to an object in order to search. In this stock search, there are two searching ways are used as one is advantages for updating. One way is when user fill any field and press on search button, the particular record available for the search value will be displayed in the text boxes. The other way is, at the same time when user click on search button, a table will be appeared showing the results for the searched value. Both these tasks happen parallelly at once. For obtaining the table, the web method is called whereas to get filled all the textboxes a web method is not used. To get values auto filled in textboxes, the database connection has been written inside the page load and the necessary coding inside the search button's on click.

### 2.3.6) Get Supplier ID Client

```
41     }
42 
43     protected void btnClear_Click(object sender, EventArgs e)
44     {
45         txtStockId.Text = "";
46         txtSupplierName.Text = "";
47         txtDrugCategory.Text = "";
48         txtUnitPrice.Text = "";
49         txtQuantity.Text = "";
50         txtSuitability.Text = "";
51         txtSupplierName.Text = "";
52         txtSupplierID.Text = "";
53     }
54 
55     protected void btnGetSupplierId_Click(object sender, EventArgs e)
56     {
57         SqlConnection sqlCon = new SqlConnection("data source=DESKTOP-2Q0B4MV; initial catalog=SOC_SPC_DB; integrated security=True");
58         sqlCon.Open();
59 
60         SqlCommand cmd = new SqlCommand("Select Supplier_ID from Supplier_Table where Supplier_Name =@Supplier_Name", sqlCon);
61 
62         cmd.Parameters.AddWithValue("@Supplier_Name", txtSupplierName.Text);
63 
64         SqlDataReader da = cmd.ExecuteReader();
65         while (da.Read())
66         {
67             txtSupplierID.Text = da.GetValue(0).ToString();
68         }
69     }
70 }
```

As it's hard to remember each supplier's IDs for the Manager, he should be eligible to type supplier name and get the id automatically. In order to fulfill that requirement, I have implemented the database connection as well as coding and the "select" query inside the on click of the button get supplier id.

### 2.3.7) Update Stocks Client



```
using System.Web.UI.WebControls;
namespace ClientApplication
{
    public partial class WarehouseStaffUpdateStock_WF : System.Web.UI.Page
    {
        AddUpdateSearchStockServiceReference.StockWSSoapClient obj = new AddUpdateSearchStockServiceReference.StockWSSoapClient();
        SqlConnection sqlCon;
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                sqlCon = new SqlConnection("data source= DESKTOP-2Q0B9PN; initial catalog=SOC_SPC_DB; Integrated Security=True");
                sqlCon.Open();
            }
            catch (Exception ex)
            {
                lblError.Text = "Error in connecting DB" + ex;
            }
        }
        protected void btnUpdateStock_Click(object sender, EventArgs e)
        {
            try
            {
                SqlCommand cmd = new SqlCommand("update Stock_Table set Drug_Name = '" + txtDrugName.Text + "', Drug_Category = '" + txtDrugCategory.Text + "', Drug_Brand = '" + txtDrugBrand.Text + "', Quantity = '" + txtQuantity.Text + "', Suitability = '" + txtSuitability.Text + "', Supplier_Name = '" + txtSupplierName.Text + "' , Supplier_ID = '" + txtSupplierID.Text + "' where Stock_ID = '" + txtStockId.Text + "'", sqlCon);
                int numberofRecords = cmd.ExecuteNonQuery();
                if (numberofRecords > 0)
                {
                    lblTxt.Text = "Record Updated";
                }
            }
            catch (Exception ex)
            {
                lblTxt.Text = "Error in Updating Data" + ex;
            }
        }
    }
}
```

When it comes to update of stocks, there is no web method is used for that as well. Thus, database connection has been implemented inside the page load and the coding for updating stocks has been written inside the on click of the update button. In that, “update” query has been used in order to update data inside the stock table. If a mistake happened when adding stock details, that can be modified using this update function. Also, for an instance if manager or a warehouse staff need to modify the quantity, they are pleased with this function.

## 2.4) Place Orders

### 2.4.1) Place Order Service

The screenshot shows the Microsoft Visual Studio interface. The main window displays the code for `PlaceOrdersWS.asmx.cs`. The code defines a class `PlaceOrdersWS` that implements a web service. It includes a database connection method `getConnection()` and a web method `placeOrders()` which performs an `insert` operation into a database table. The Solution Explorer on the right shows various project files and references.

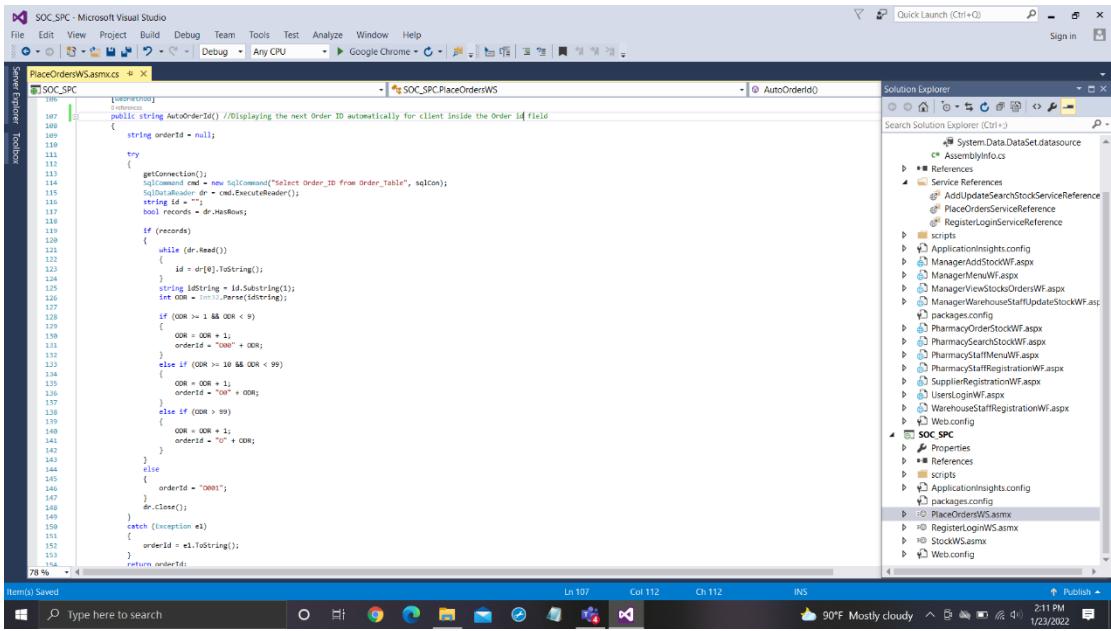
```
// [System.Web.Script.Services.ScriptService]
//Creating the database connection
public class PlaceOrdersWS : System.Web.Services.WebService
{
    SqlConnection sqlCon = null;

    [WebMethod]
    public SqlConnection getConnection()
    {
        try
        {
            sqlCon = new SqlConnection("data source= DESKTOP-QQ8H4N; initial catalog=SOC_SPC_DB; Integrated Security=True");
            sqlCon.Open();
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error Connecting to DB" + ex);
        }
        return sqlCon;
    }

    [WebMethod]
    public string placeOrders(string orderId, string drugName, string drugBread, int quantity, string pharmacyName, string branch, string pharmacyStaffId)
    {
        int temp = 0;
        try
        {
            getConnection();
            SqlCommand cmd = new SqlCommand("insert into Order_Table values ('" + orderId + "','" + drugName + "','" + drugBread + "','" + quantity + "','" + branch + "','" + pharmacyName + "','" + pharmacyStaffId + "')", sqlCon);
            temp = cmd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            return ex.ToString();
        }
        return temp.ToString();
    }
}
```

In the above screenshot, it shows the web service called “PlaceOrdersWS”. All the necessary web methods for place orders, get order id auto and search orders have been coded inside this web service file. All the coding has done inside the class called PlaceOrdersWS. At the beginning, it shows the database connection code withing try and catch blocks. After that, the web method for placing orders is there which includes the coding for the placing of orders is visible in the screenshot. The “insert” query has been used to place new order details.

## 2.4.2) Get Order ID Auto Service



The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** SOC\_SPC - Microsoft Visual Studio
- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help
- Toolbar:** Standard toolbar icons.
- Code Editor:** The file `PlaceOrdersWS.asmx.cs` is open, showing C# code for generating an auto-order ID. The code uses a `SqlConnection` and `SqlCommand` to query a database table named `Order_Table` to find the next available order ID. It handles various conditions for generating the ID, including handling exceptions and returning the generated ID.
- Solution Explorer:** Shows the project structure for `SOC_SPC`, including files like `PlaceOrdersWS.asmx`, `Web.config`, and several ASPX pages such as `ManageOrderHistoryWF.aspx` and `ManageViewStocksOrdersWF.aspx`.
- Toolbox:** Standard .NET toolbox items.
- Status Bar:** Shows the current file is `PlaceOrdersWS.asmx.cs`, line 107, column 112, character 112. It also shows the date and time as 1/23/2022 2:11 PM.

In the above evidence, it depicts the coding that has been done for getting the next Order id automatically when the page is loaded. That also has been done inside a web method. For that “select” query has been used. To do that, while and if...else if...else statements has used. In that, Order ids start from O1, O2, etc. and then O10, O11, etc. and then O100, O101, etc. And then return the correct Order id finally.

### 2.4.3) Place Order Client

SOC\_SPC - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

PharmacySearchStockWF.aspx.cs PharmacyOrderStockWF.aspx.cs PlaceOrdersWSasmx.cs ClientApplication

```
7  using System.Web.UI.WebControls;
8
9  namespace ClientApplication
10 {
11     public partial class PharmacyOrderStock : System.Web.UI.Page
12     {
13         PlaceOrdersServiceReference.PlaceOrdersWSSoapClient obj = new PlaceOrdersServiceReference.PlaceOrdersWSSoapClient();
14         SqlConnection sqlConn;
15
16         protected void Page_Load(object sender, EventArgs e)
17         {
18             txtOrderId.Text = obj.AutoOrderID();
19         }
20
21         protected void btnPlaceOrder_Click(object sender, EventArgs e)
22         {
23             string value = obj.placeOrders(txtOrderID.Text, txtDrugName.Text, txtDrugBrand.Text, Convert.ToInt32(txtQuantity.Text), txtPharmacyName.Text,
24             txtBranch.Text, txtPharmacyStaffId.Text);
25             int record = int.Parse(value);
26
27             if (Page.IsValid)
28             {
29                 if (record > 0)
30                 {
31                     lblResult.Text = "Order Placed";
32                 }
33                 else
34                 {
35                     lblResult.Text = "Order Placing Unsuccessful";
36                 }
37             }
38             else
39             {
40                 lblMsg.Text = "Error In Placing!";
41             }
42         }
43
44     }
45 }
```

Quick Launch (Ctrl+Q) Sign in

ClientApplication

PlaceOrdersWSasmx.cs

btnPlaceOrder\_Click(object sender, EventArgs)

Search Solution Explorer (Ctrl+F)

- System.Data.DataSet.datasource
- AssemblyInfo.cs
- References
- Service References
  - AddUpdateSearchStockServiceReference
  - PlaceOrdersServiceReference
  - RegisterLoginServiceReference
- scripts
- ApplicationInsights.config
- ManagerAddStockWF.aspx
- ManagerMenuWF.aspx
- ManagerViewStocksOrdersWF.aspx
- ManagerWarehouseStaffUpdateStockWF.aspx
- packages.config
- PharmacyAddStockWF.aspx
- PharmacyOrderStockWF.aspx
- PharmacyOrderStockWF.aspx.cs
- PharmacyOrderStockWF.aspx.designer.cs
- PharmacySearchStockWF.aspx
- PharmacySearchStockWF.aspx.cs
- PharmacySearchStockWF.aspx.designer.cs
- PharmacyStaffMenuWF.aspx
- PharmacyStaffRegistrationWF.aspx
- SupplierRegistrationWF.aspx
- UsersLoginWF.aspx
- WarehouseStaffRegistrationWF.aspx
- Web.config
- SOC\_SPC
- Properties
- References
- scripts
- ApplicationInsights.config
- packages.config

94% 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

Ready Ln 24 Col 17 Ch 17 INS

90°F Mostly cloudy 12:14 PM 1/23/2023

As in the above attach screenshot, all the coding has been done inside a class called "PharmacyOrderStock". Then an object has created to call the web method relevant to that. Inside page load, the particular web method is called in order to get the next Order id automatically when the page is loading. Its visible in the screenshot that the coding for Placing Order has been done by calling the method for Place Orders in the web method.

## 2.4.4) Search Order Service

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** SOC\_SPC - Microsoft Visual Studio
- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help
- Toolbar:** Standard toolbar icons.
- Code Editor:** The file `PlaceOrdersWS.asmx.cs` is open, showing C# code for a web service. The code defines a `[WebMethod]` named `searchOrders` which takes parameters `orderId`, `drugName`, `drugBrand`, `pharmacyName`, `branch`, and `pharmacyStaffId`. It uses a `DataSet` and `SqlDataAdapter` to execute a `SqlCommand` against the `Order_Table`.
- Solution Explorer:** Shows the project structure for `SOC_SPC` including files like `System.Data.Dataset.datasource`, `AssemblyInfo.cs`, `References`, `Service References`, `scripts`, `packages`, and `Web.config`. Specific files listed include `AddUpdateSearchStockServiceReference.cs`, `PlaceOrdersServiceReference.cs`, `RegisterLoginServiceReference.cs`, `ApplicationInsights.config`, `packages.config`, and `Web.config`.
- Status Bar:** Shows the date and time as 1/23/2022 2:10 PM.

This screenshot shows the web method which has been created for searching orders. This also has written inside the web service called “PlaceOrdersWS”. In here, the “select” query has been used in order to search from the orders database table.

## 2.4.5) Search Order Client

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7
8  namespace ClientApplication
9  {
10     public partial class PharmacySearchOrderStock : System.Web.UI.Page
11     {
12         AddUpdateSearchStockServiceReference.StockWSSoapClient obj = new AddUpdateSearchStockServiceReference.StockWSSoapClient();
13
14         protected void Page_Load(object sender, EventArgs e)
15         {
16         }
17
18         protected void btnPharmacySearchStock_Click(object sender, EventArgs e)
19         {
20             GVPharmacySearchStock.DataSource = obj.searchStock(txtStockId.Text, txtDrugName.Text, txtDrugCategory.Text, txtDrugBrand.Text, txtSuitability.Text,
21             txtSupplierName.Text, txtSupplierID.Text);
22             GVPharmacySearchStock.DataBind();
23         }
24
25         protected void btnClear_Click(object sender, EventArgs e)
26         {
27             txtStockId.Text = "";
28             txtDrugName.Text = "";
29             txtDrugCategory.Text = "";
30             txtDrugBrand.Text = "";
31             txtSuitability.Text = "";
32             txtSupplierName.Text = "";
33             txtSupplierID.Text = "";
34         }
35     }
36 }

```

As displayed in the attached screenshot, the web method is called to an object in order to search. In this Order search, there is only one searching way is used. That way is, when user clicks on search button, a table will be appeared showing the results for the searched value. For obtaining the table, the web method is called. The necessary coding for that has been written inside the search button's on click.

## 2.4.6) Get Pharmacy Staff ID Client

```
using System;
using System.Data.SqlClient;
using System.Configuration;
using System.Web.UI.WebControls;
using System.Web.UI;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web;

public partial class PharmacySearchStockWF : System.Web.UI.Page
{
    protected void btnClear_Click(object sender, EventArgs e)
    {
        txtOrderId.Text = "";
        txtDrugName.Text = "";
        txtDrugBrand.Text = "";
        txtQuantity.Text = "";
        txtPharmacyName.Text = "";
        txtBranch.Text = "";
        txtPharmacyStaffId.Text = "";
    }

    protected void btnGetPharmacyStaffId_Click(object sender, EventArgs e)
    {
        SqlConnection sqlCon = new SqlConnection("data source=DESKTOP-2Q8H#W; initial catalog=SOC_SPC_DB; Integrated Security=True");
        sqlCon.Open();

        SqlCommand cmd = new SqlCommand("Select Branch , Pharmacy_Staff_ID from Pharmacy_Staff_Table where Pharmacy_Name =@Pharmacy_Name", sqlCon);
        cmd.Parameters.AddWithValue("@Pharmacy_Name", txtPharmacyName.Text);

        SqlDataReader da = cmd.ExecuteReader();
        while (da.Read())
        {
            txtBranch.Text = da.GetValue(0).ToString();
            txtPharmacyStaffId.Text = da.GetValue(1).ToString();
        }
    }
}
```

As it's hard to remember each Pharmacy Staff members IDs for them, they should be eligible to type pharmacy name they are assigned and get the branch name and staff pharmacy staff id automatically. In order to fulfill that requirement, I have implemented the database connection as well as coding and the “select” query inside the on click of the button get pharmacy staff id.

## TASK 3-Testing

### 3.1) Test Plan

TEST CASE ID	TEST CASE NAME	DESCRIPTION	EXPECTED RESULT
TC1	Check Login empty fields	Have to press on login button without filling any field	Should display an Error Message on username field as username can't be blank and on password fields as password can't be blank
TC2	Manager Login		
TC2.1	Check with empty username field	Have to enter only a password and press Login as button	Should display an Error Message on username field as username can't be blank
TC2.2	Check with empty password field	Have to enter only hardcoded username and press Login as button	Should display an Error Message on password field as password can't be blank
TC2.3	Check with incorrect manager username	Have to enter incorrect username with correct hardcoded manager password and manager user type	Should display an Error Message as Invalid login credentials! Try again
TC2.4	Check with incorrect manager password	Have to enter incorrect password with correct hardcoded manager username and manager user type	Should display an Error Message as Invalid login credentials! Try again
TC2.5	Check with incorrect user type	Have to enter hardcoded correct manager username and password without Manager User Type	Should display an Error Message as Invalid login credentials! Try again
TC2.6	Check Login working without any empty field and with correct login credentials	Have to enter hardcoded correct Manager username, password and Manager User Type	Should display the Manager Menu
TC3	Warehouse Staff Registration		

TC3.1	Check Register empty fields	Have to press on register button without filling any fields	Should display Error Messages on each field except warehouse staff id and user type fields as (field name) can't be blank
TC3.2	Check other empty fields by filling only employee name field	Have to press on register button by filling only employee name field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.3	Check other empty fields by filling only username field	Have to press on register button by filling only username field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.4	Check other empty fields by filling only password field	Have to press on register button by filling only password field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.5	Check other empty fields by filling only confirm password field	Have to press on register button by filling only confirm password field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.6	Check other empty fields by filling only email field	Have to press on register button by filling only email field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.7	Check other empty fields by filling only telephone field	Have to press on register button by filling only telephone field	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
TC3.8	Check matching password and confirm password	Have to enter a password and a different confirm password	Should display an Error Message on Confirm Password field as password and confirm password should match
TC3.9	Check email format	Have to enter an email with an incorrect email format	Should display an Error Message on email field as Invalid Email
TC3.10	Check Registration working without any empty fields	Have to fill all fields and press register button	Should display a message as Warehouse Staff Registered Successfully

TC4	Warehouse Staff Login		
TC4.1	Check with empty username field	Have to enter only a password and press Login as button	Should display an Error Message on username field as username can't be blank
TC4.2	Check with empty password field	Have to enter only a username and press Login as button	Should display an Error Message on password field as password can't be blank
TC4.3	Check with incorrect warehouse staff username	Have to enter incorrect username with correct warehouse staff password and warehouse staff user type	Should display an Error Message as Invalid login credentials! Try again
TC4.4	Check with incorrect warehouse staff password	Have to enter incorrect password with correct warehouse staff username and warehouse staff user type	Should display an Error Message as Invalid login credentials! Try again
TC4.5	Check with incorrect user type	Have to enter correct warehouse staff username and password without warehouse staff User Type	Should display an Error Message as Invalid login credentials! Try again
TC4.6	Check Login working without any empty field and with correct login credentials	Have to enter correct Manager warehouse staff, password and warehouse staff User Type	Should display the Stock Update interface
TC5	Pharmacy Staff Registration		
TC5.1	Check Register empty fields	Have to press on register button without filling any fields	Should display Error Messages on each field except Pharmacy Staff id and user type fields as (field name) can't be blank
TC5.2	Check other empty fields by filling only employee name field	Have to press on register button by filling only employee name field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields

TC5.3	Check other empty fields by filling only username field	Have to press on register button by filling only username field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.4	Check other empty fields by filling only password field	Have to press on register button by filling only password field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.5	Check other empty fields by filling only confirm password field	Have to press on register button by filling only confirm password field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.6	Check other empty fields by filling only pharmacy name field	Have to press on register button by filling only pharmacy name field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.7	Check other empty fields by filling only branch field	Have to press on register button by filling only branch field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.8	Check other empty fields by filling only email field	Have to press on register button by filling only email field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.9	Check other empty fields by filling only telephone field	Have to press on register button by filling only telephone field	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
TC5.10	Check matching password and confirm password	Have to enter a password and a different confirm password	Should display an Error Message on Confirm Password field as password and confirm password should match
TC5.11	Check email format	Have to enter an email with an incorrect email format	Should display an Error Message on email field as Invalid Email
TC5.12	Check Registration working without any empty fields	Have to fill all fields and press register button	Should display a message as Pharmacy Staff Registered Successfully

TC6	Pharmacy Staff Login		
TC6.1	Check with empty username field	Have to enter only a password and press Login as button	Should display an Error Message on username field as username can't be blank
TC6.2	Check with empty password field	Have to enter only a username and press Login as button	Should display an Error Message on password field as password can't be blank
TC6.3	Check with incorrect Pharmacy Staff username	Have to enter incorrect username with correct Pharmacy Staff password and Pharmacy Staff user type	Should display an Error Message as Invalid login credentials! Try again
TC6.4	Check with incorrect Pharmacy Staff password	Have to enter incorrect password with correct Pharmacy Staff username and Pharmacy Staff user type	Should display an Error Message as Invalid login credentials! Try again
TC6.5	Check with incorrect user type	Have to enter correct Pharmacy Staff username and password without Pharmacy Staff User Type	Should display an Error Message as Invalid login credentials! Try again
TC6.6	Check Login working without any empty field and with correct login credentials	Have to enter correct Manager Pharmacy Staff, password and Pharmacy Staff User Type	Should display the Pharmacy Menu
TC7	Supplier Registration		
TC7.1	Check Register empty fields	Have to press on register button without filling any fields	Should display Error Messages on each field except Supplier id as (field name) can't be blank
TC7.2	Check other empty fields by filling only supplier name field	Have to press on register button by filling only supplier name field	Should display Error Messages on all other fields with field name
TC7.3	Check other empty fields by filling only company name field	Have to press on register button by filling only company name field	Should display Error Messages on all other fields with field name

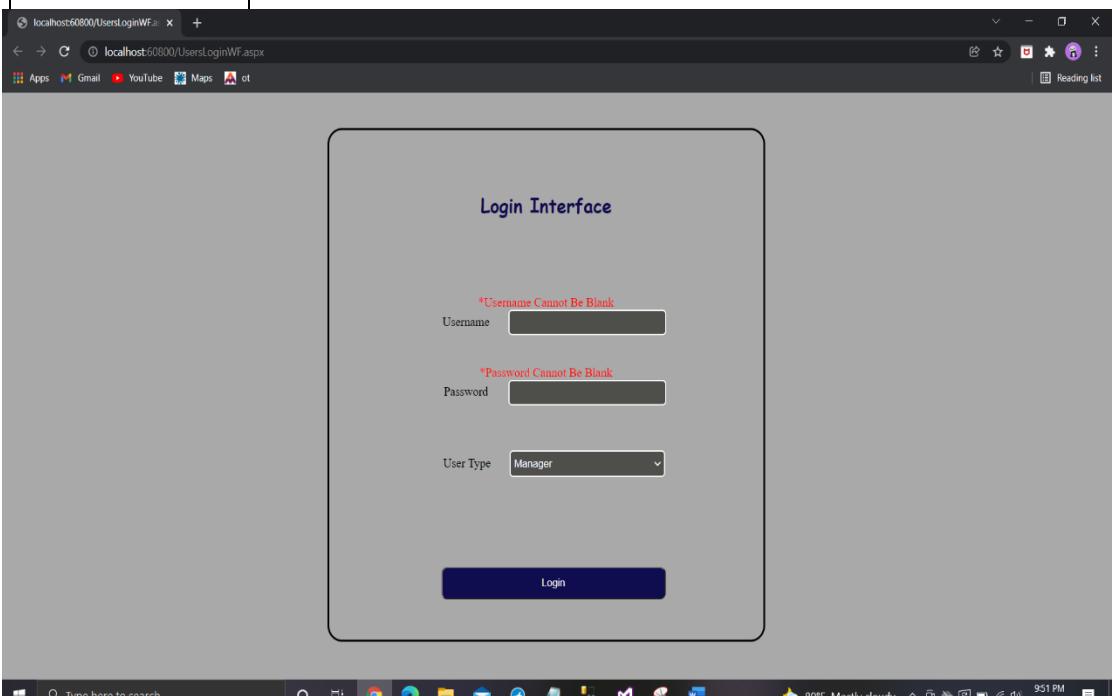
TC7.4	Check other empty fields by filling only email field	Have to press on register button by filling only email field	Should display Error Messages on all other fields with field name
TC7.5	Check other empty fields by filling only telephone field	Have to press on register button by filling only telephone field	Should display Error Messages on all other fields with field name
TC7.6	Check email format	Have to enter an email with an incorrect email format	Should display an Error Message on email field as Invalid Email
TC7.7	Check Registration working without any empty field and with proper data	Have to fill all fields and press register button	Should display a message as Supplier Registration Successful
TC8	Add Stock		
TC8.1	Check gets supplier id button empty fields	Have to press on Get Supplier ID button when the interface loaded	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.2	Check working of get supplier button	Enter a supplier name and enter Get Supplier Id button	Should supplier id get filled automatically
TC8.3	Check Add stock empty fields	Have to press on add stock button without filling any fields	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.4	Check other empty fields by filling only drug name field	Have to press on add stock button by filling only drug name field	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.5	Check other empty fields by filling only drug category field	Have to press on add stock button by filling only drug category field	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.6	Check other empty fields by filling only drug brand field	Have to press on add stock button by filling only drug brand field	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank

TC8.7	Check other empty fields by filling only quantity field	Have to press on add stock button by filling only quantity field	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.9	Check other empty fields by filling only name of the illness field	Have to press on add stock button by filling only name of the illness field	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
TC8.10	Check Add Stock working without any empty fields	Have to fill all fields and press add stock button	Should display a message as Stock Added Successfully
TC9	Update Stock		
TC9.1	Check search button with empty fields in update stock interface	Press search stock button without filling any field	A table should appear and show an Error Message as No Results Found
TC9.2	Check search button in update stock interface	Have to enter any field expect quantity field and press on search stock button	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row
TC9.3	Check update button	Have to search by entering any field expect quantity field and then update necessary fields and press on update button	A message will be appeared as Stock Updated
TC10	View added stocks and placed orders		
TC10.1	Check search button with empty fields in view added stocks and placed orders interface	Press search stock button without filling any field	A table should appear and show an Error Message as No Results Found
TC10.2	Check search button in view added stocks and placed orders interface	Have to enter any field and press on search orders button	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row

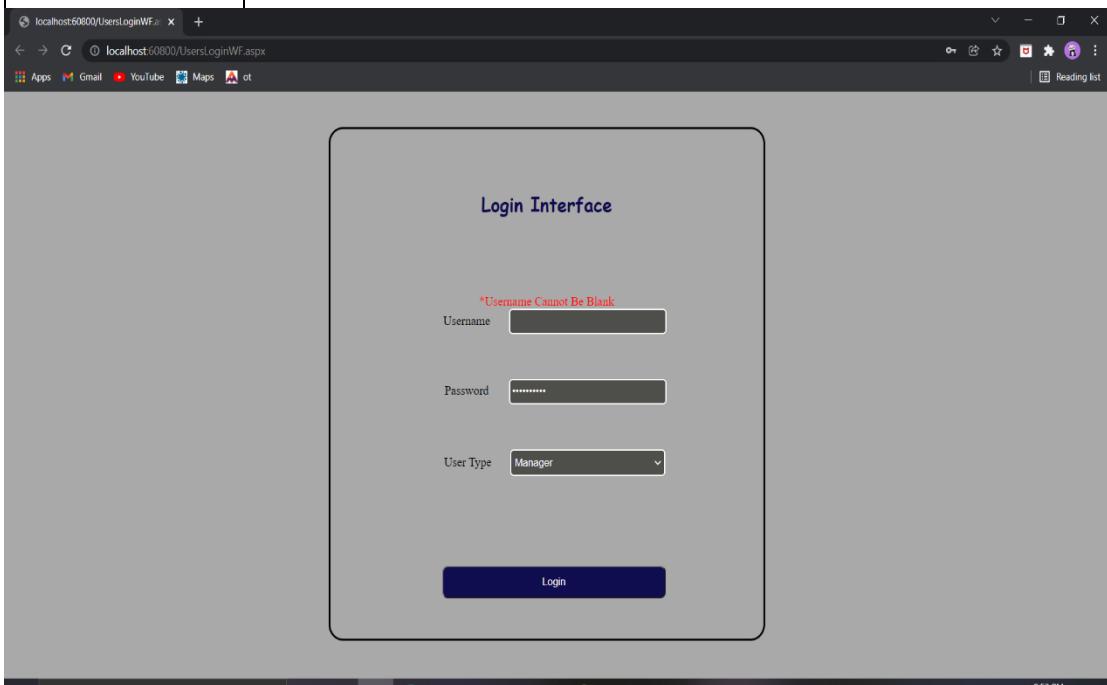
TC10.3	Check View Stock button	Press on View Stock button when the interface is loaded	A table should be appeared with the values in the particular database table row
TC10.4	Check View Orders button	Press on View Orders button when the interface is loaded	A table should be appeared with the values in the particular database table row
TC11	Order Drugs		
TC11.1	Check Get Branch & Staff ID button empty fields	Have to press on Get Branch & Staff ID button when the interface loaded	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
TC11.2	Check working Get Branch & Staff ID button	Enter a pharmacy name and enter Get Branch & Staff ID button	Should branch and pharmacy staff id fields get filled automatically
TC11.3	Check Order Drugs empty fields	Have to press on Place an Order button without filling any fields	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
TC11.4	Check other empty fields by filling only drug name field	Have to press on Place an Order button by filling only drug name field	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
TC11.5	Check other empty fields by filling only drug brand field	Have to press on Place an Order button by filling only drug brand field	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
TC11.6	Check other empty fields by filling only quantity field	Have to press on Place an Order button by filling only quantity field	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
TC11.7	Check other empty fields by filling only pharmacy name field	Have to press on Place an Order button by filling only pharmacy name field	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank

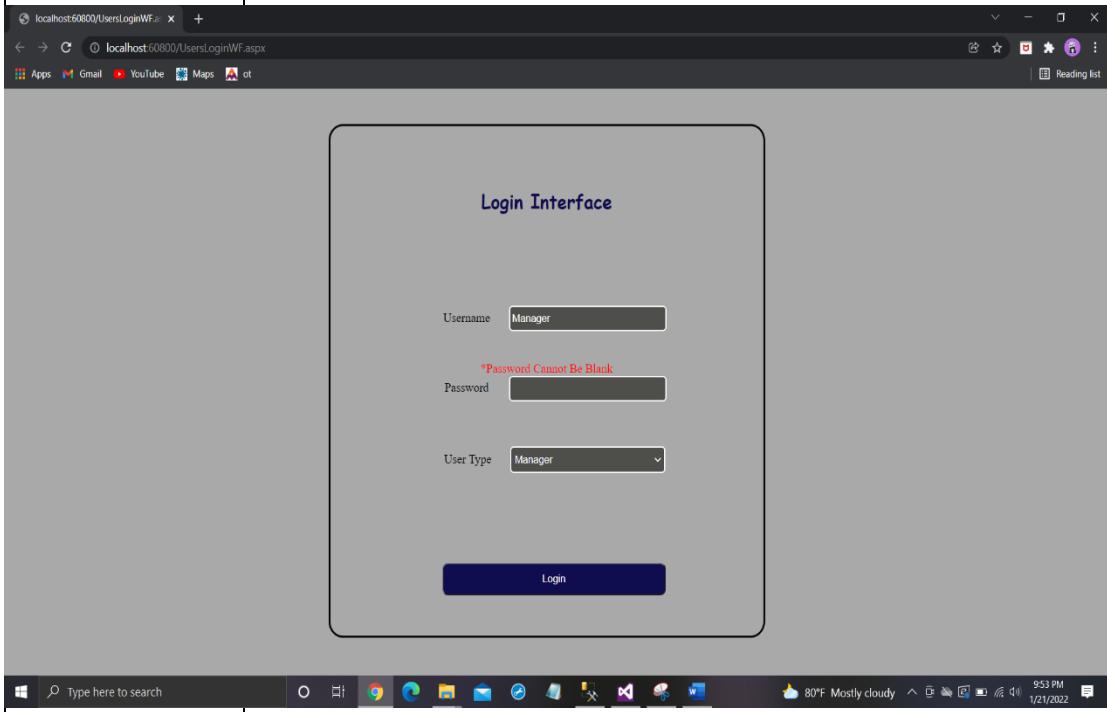
TC11.8	Check Place an Order working without any empty fields	Have to fill all fields and press Place an Order button	Should display a message as Order Placed Successfully
TC12	Pharmacy Search Drugs		
TC12.1	Check search drugs button with empty fields	Press search drugs button without filling any field	A table should appear and show an Error Message as No Results Found
TC12.2	Check search drugs button	Have to enter any field except quantity field and press on search drugs button	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row

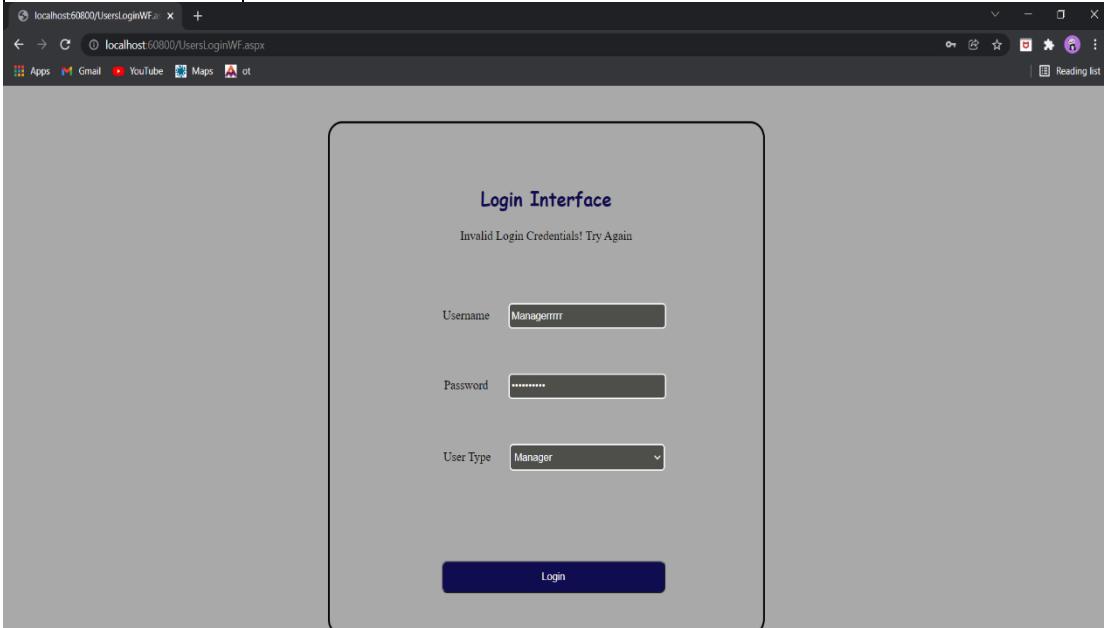
### 3.2) Test Cases

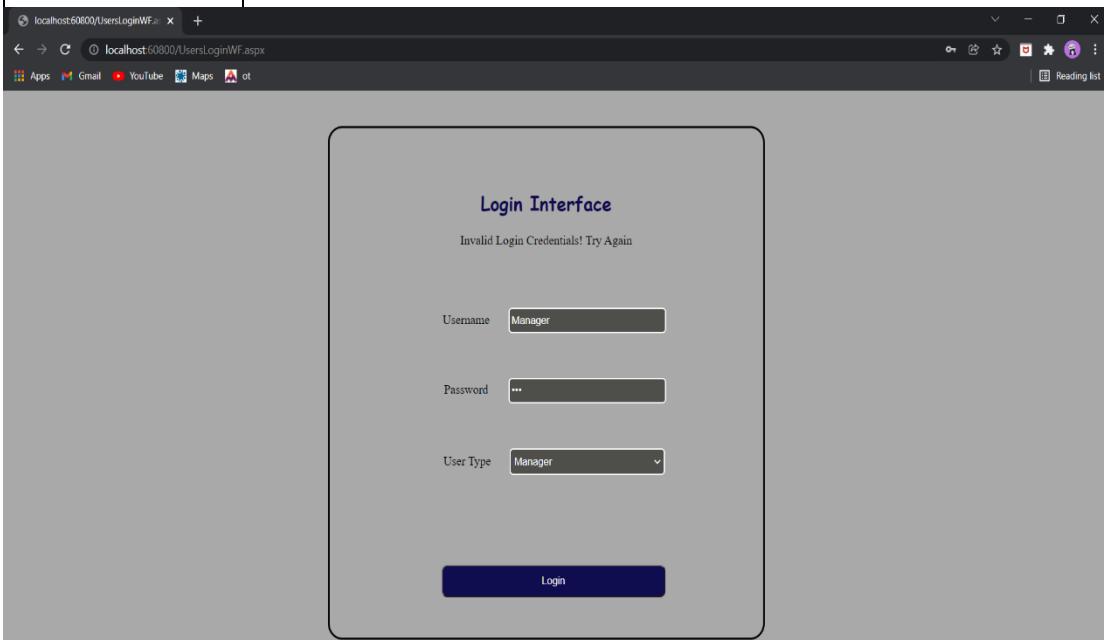
TEST CASE ID	TC1
TEST CASE NAME	Check Login empty fields
DESCRIPTION	Have to press on login button without filling any field
TEST STEPS	Just press on login button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display an Error Message on username field as username can't be blank and on password fields as password can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages displayed on the text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

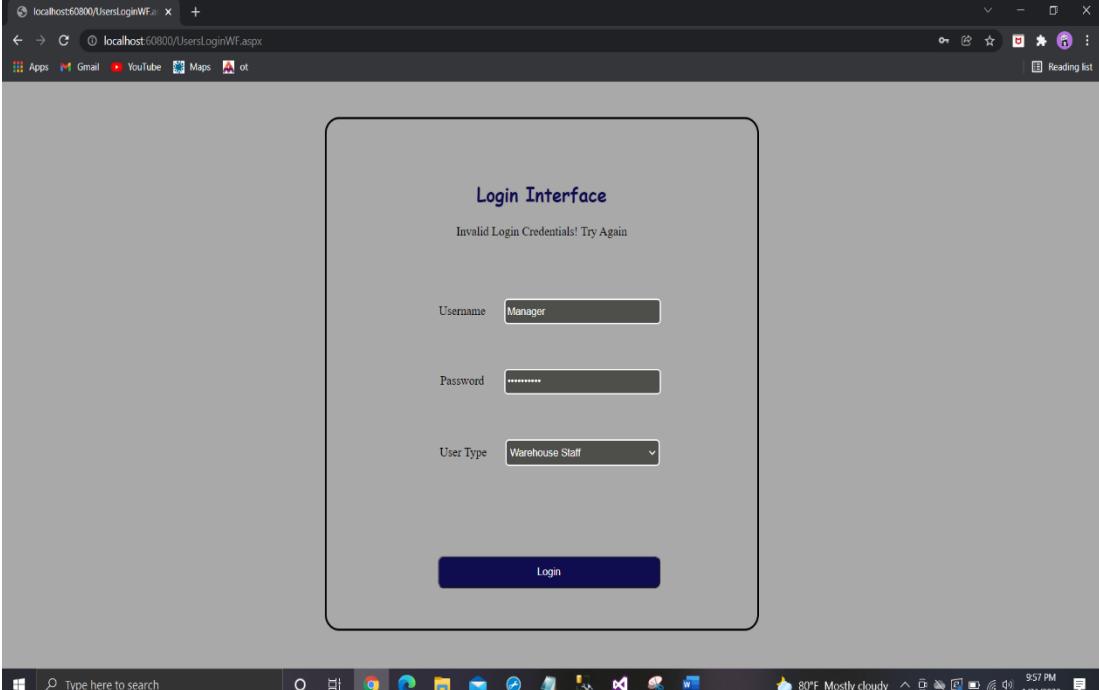
## TC2 - Manager Login

TEST CASE ID	TC2.1
TEST CASE NAME	Check with empty username field
DESCRIPTION	Have to enter only a password and press Login as button
TEST STEPS	1.Fill only the password field when the interface is loaded 2.Press on login button
TEST DATA	Password: Manager123
EXPECTED RESULT	Should display an Error Message on password field as password can't be blank
ACTUAL RESULT	 <p>The screenshot shows a Windows desktop environment. A browser window titled 'localhost:60800/UsersLoginWF.aspx' is open, displaying a 'Login Interface'. The interface includes fields for 'Username' (empty), 'Password' (filled with dots), and 'User Type' (set to 'Manager'). An error message 'Username Cannot Be Blank' is displayed above the empty 'Username' field. A large black rectangular box covers the bottom half of the screen, obscuring the keyboard and taskbar.</p>
CONCLUSION	The expected an error message displayed on the username text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

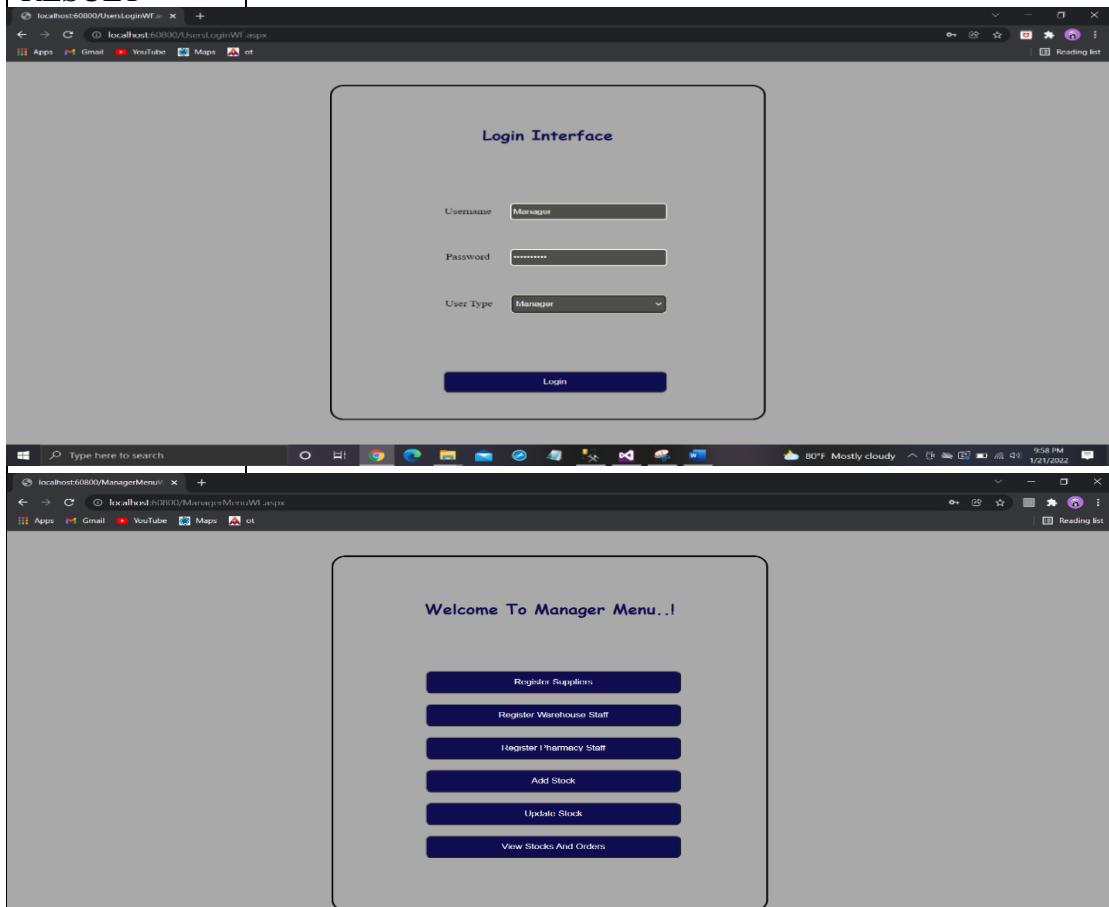
TEST CASE ID	TC2.2
TEST CASE NAME	Check with empty password field
DESCRIPTION	Have to enter only hardcoded username and press Login as button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only the username field when the interface is loaded</li> <li>2.Press on login button</li> </ol>
TEST DATA	Username: Manager
EXPECTED RESULT	Should display an Error Message on password field as password can't be blank
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed on the password text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.3
TEST CASE NAME	Check with incorrect manager username
DESCRIPTION	Have to enter incorrect username with correct hardcoded manager password and manager user type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with wrong username</li> <li>2.Fill password field with a correct hardcoded password</li> <li>3.Select user type as Manager</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Managerrrrr
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/UsersLoginWF.aspx". The page has a header "Login Interface" and a message "Invalid Login Credentials! Try Again". It contains three input fields: "Username" with the value "Managerrrrr", "Password" with masked input, and "User Type" set to "Manager". A "Login" button is at the bottom.
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC2.4
TEST CASE NAME	Check with incorrect manager password
DESCRIPTION	Have to enter incorrect password with correct hardcoded manager username and manager user type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill password field with wrong password</li> <li>2.Fill username field with a correct hardcoded username</li> <li>3.Select user type as Manager</li> <li>4.Press on Login button</li> </ol>
TEST DATA	password: 123
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

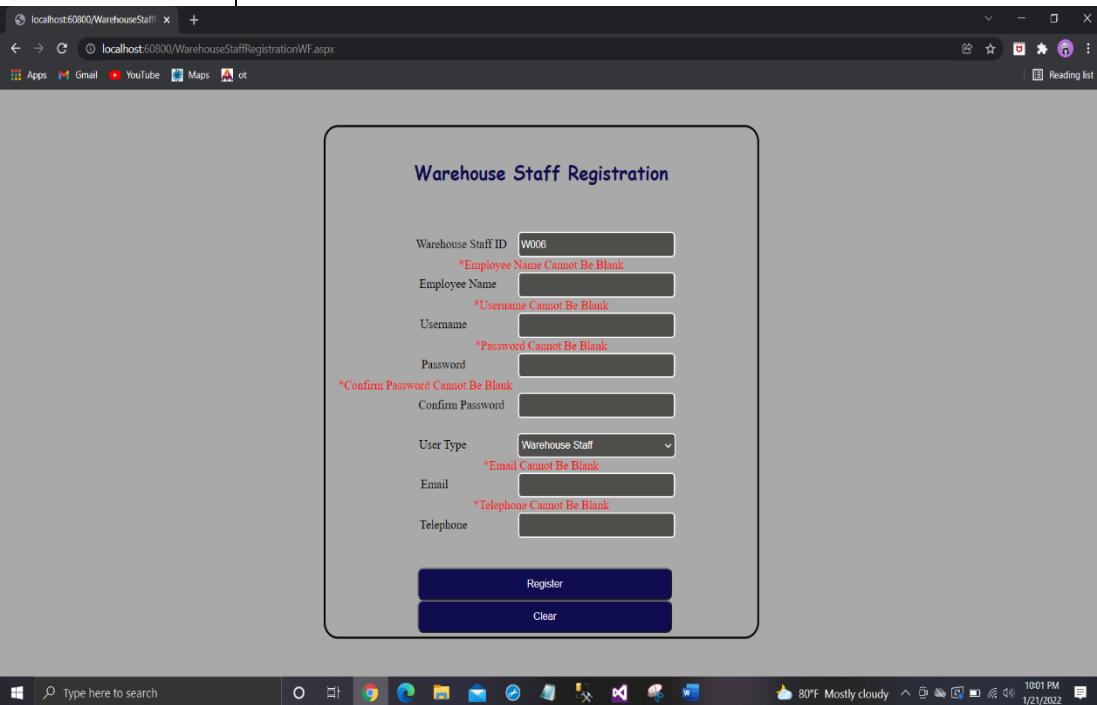
TEST CASE ID	TC2.5
TEST CASE NAME	Check with incorrect user type
DESCRIPTION	Have to enter hardcoded correct manager username and password without Manager User Type
TEST STEPS	<ol style="list-style-type: none"> <li>Fill username field with correct hardcoded username</li> <li>Fill password field with a correct hardcoded password</li> <li>Select user type as Warehouse Staff</li> <li>Press on Login button</li> </ol>
TEST DATA	Username: Manager Password: Manager123 User Type: Warehouse Staff
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	

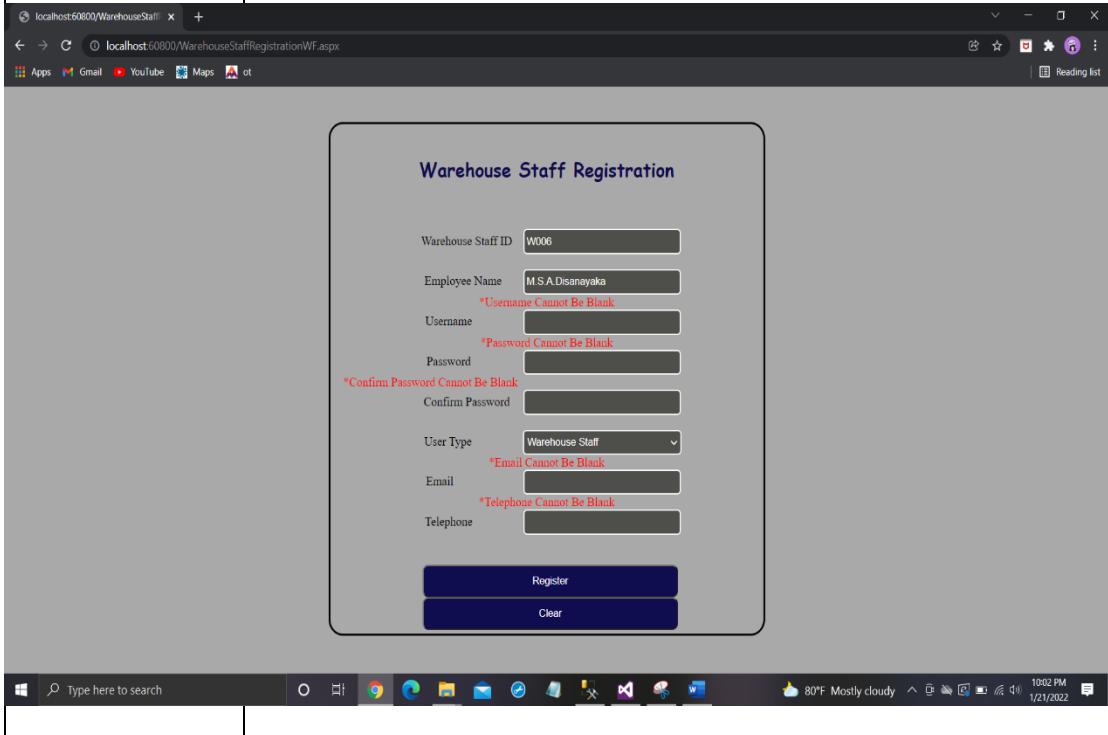
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass
TEST CASE ID	TC2.6
TEST CASE NAME	Check Login working without any empty field and with correct login credentials
DESCRIPTION	Have to enter hardcoded correct Manager username, password and Manager User Type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with correct hardcoded username</li> <li>2.Fill password field with a correct hardcoded password</li> <li>3.Select user type as Manager</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Manager Password: Manager123 User Type: Manager
EXPECTED RESULT	Should display the Manager Menu
ACTUAL RESULT	

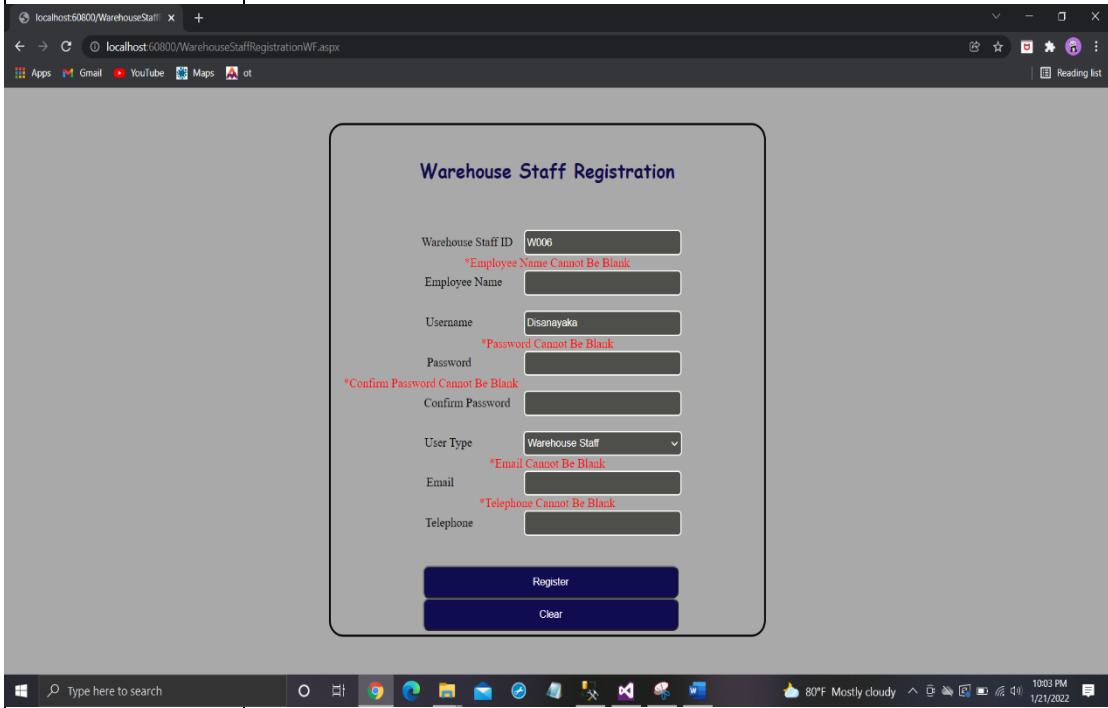


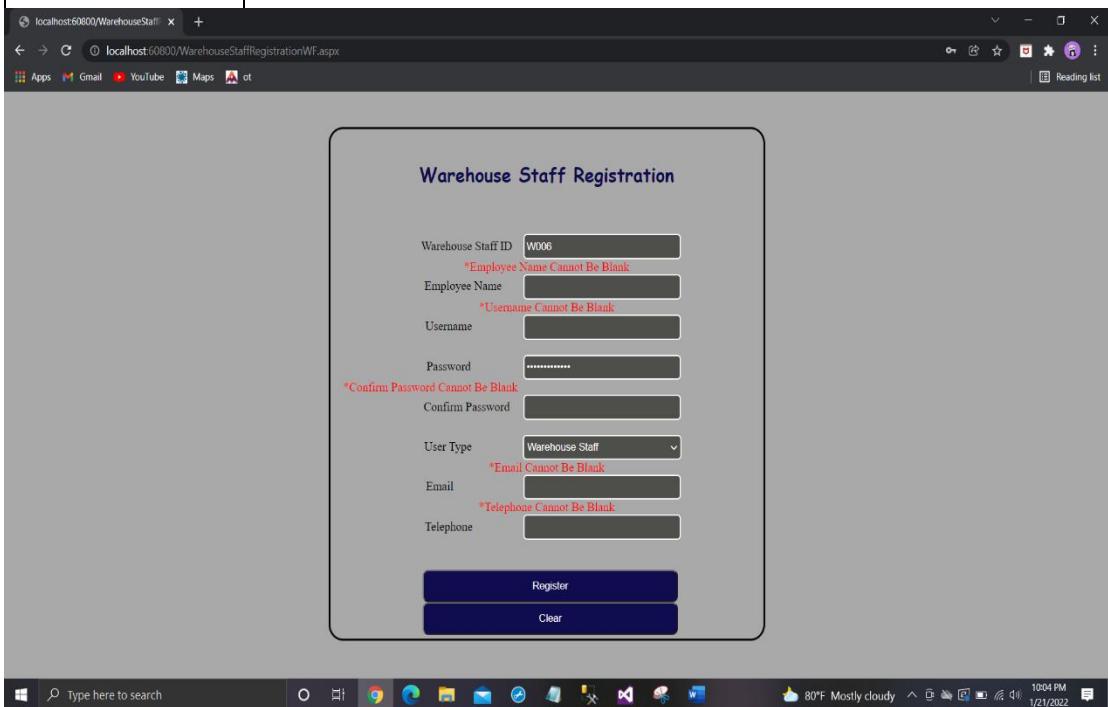
CONCLUSION	The expected an interface is displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

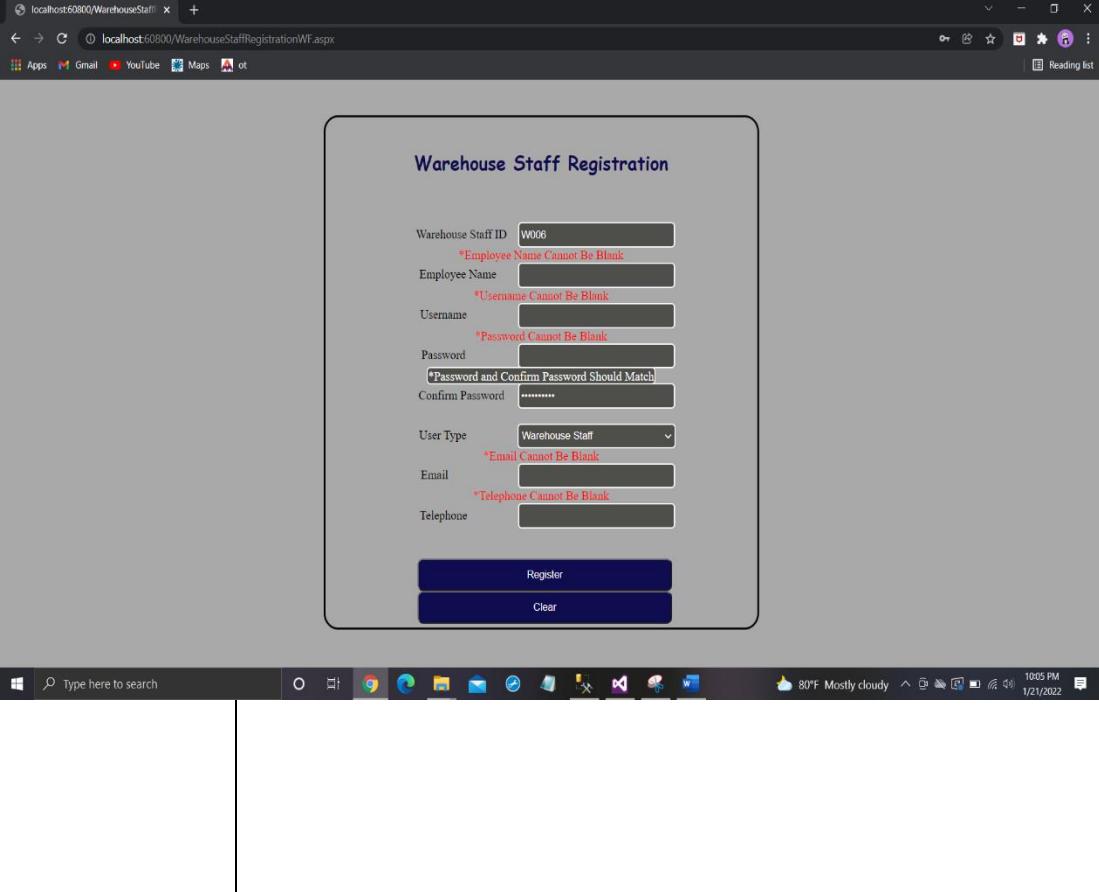
### TC3 - Warehouse Staff Registration

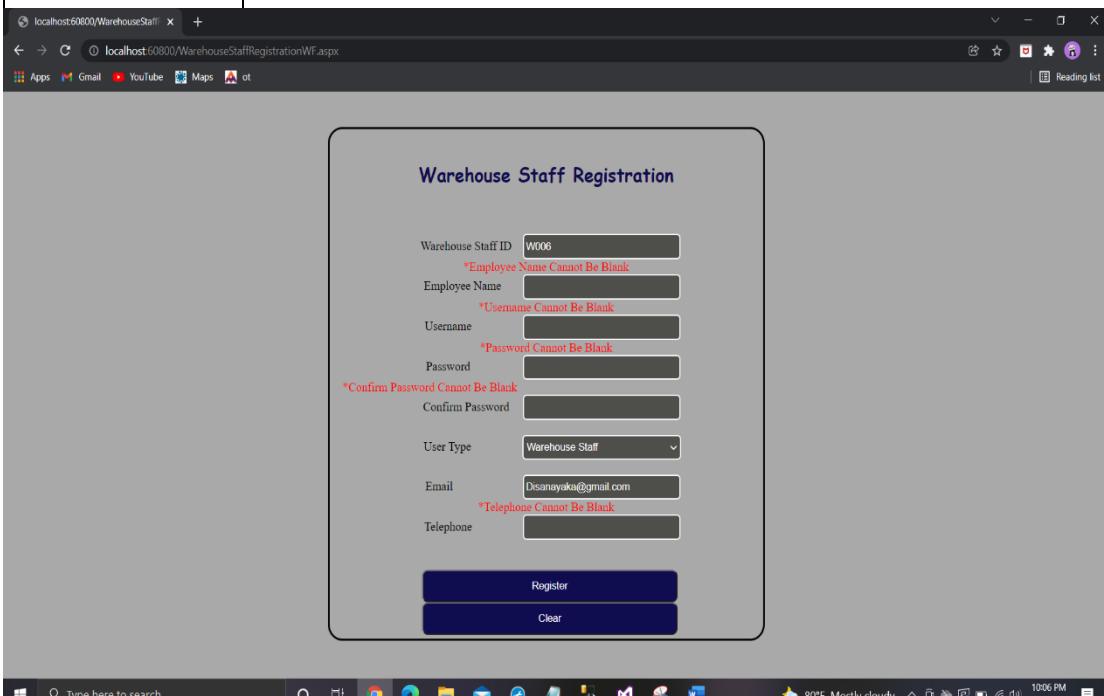
TEST CASE ID	TC3.1
TEST CASE NAME	Check Register empty fields
DESCRIPTION	Have to press on register button without filling any fields
TEST STEPS	Just press on register button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except warehouse staff id and user type fields as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

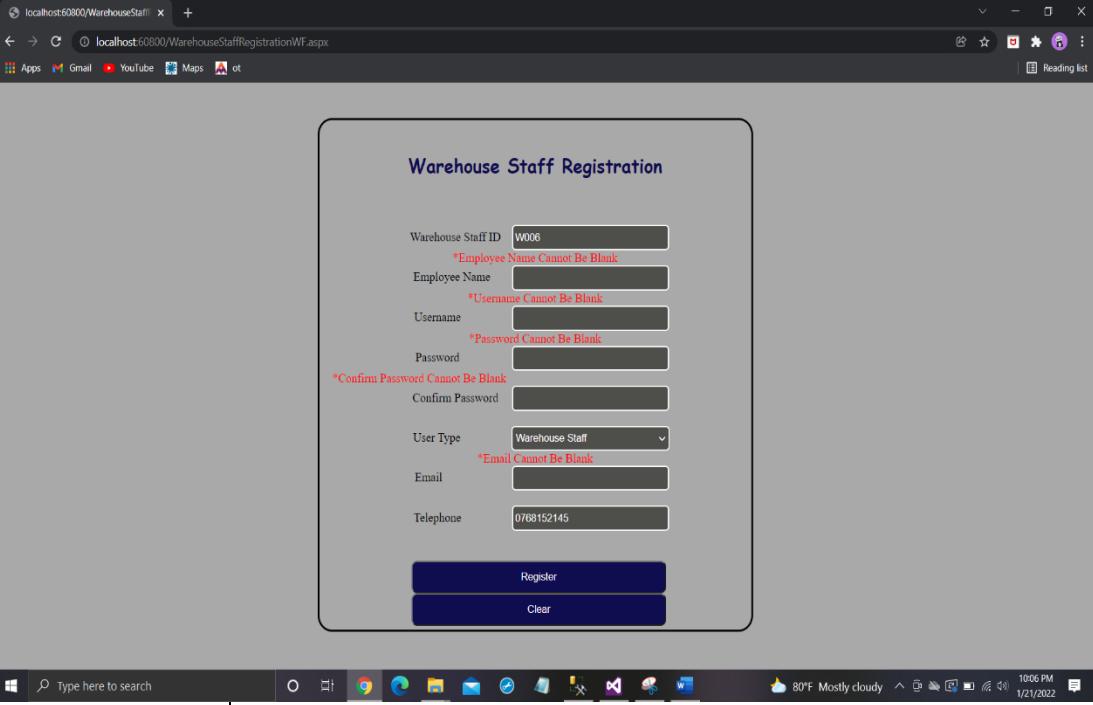
TEST CASE ID	TC3.2
TEST CASE NAME	Check other empty fields by filling only employee name field
DESCRIPTION	Have to press on register button by filling only employee name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only Employee Name field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except warehouse staff id and user type fields as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

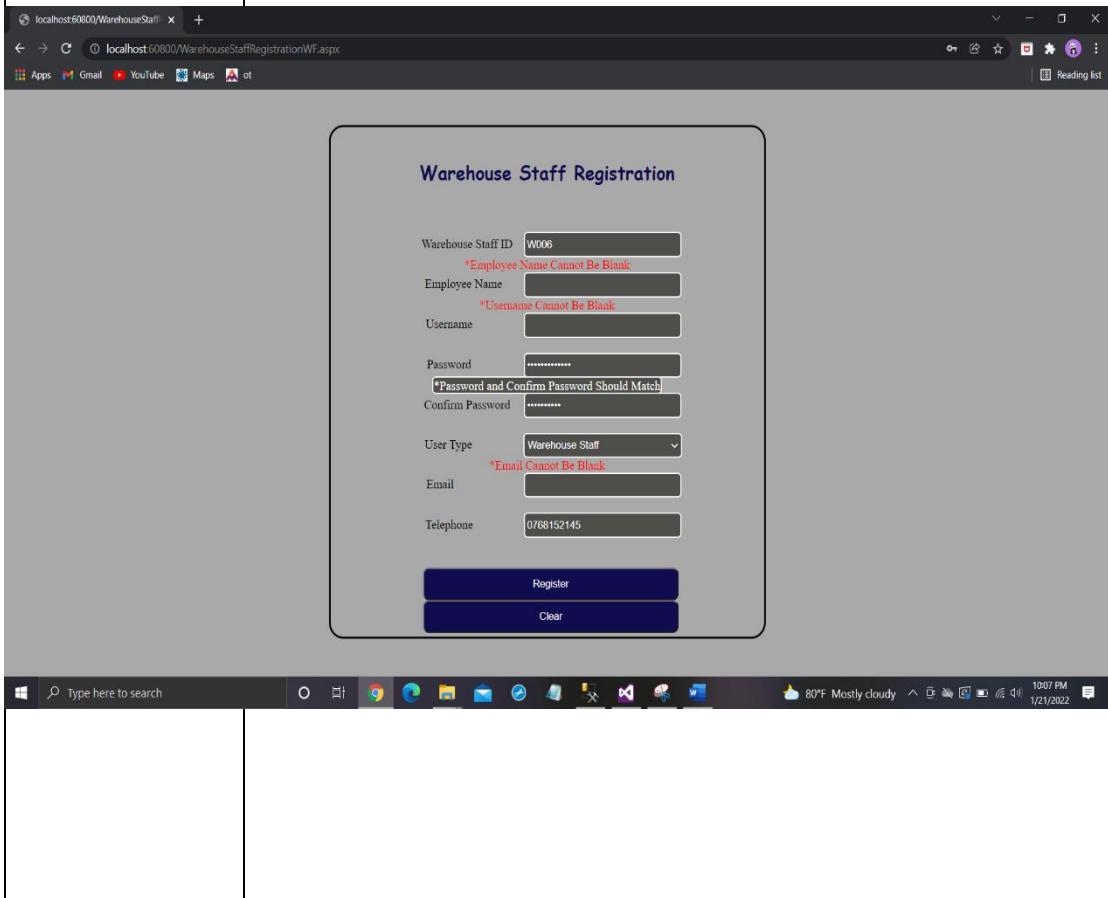
TEST CASE ID	TC3.3
TEST CASE NAME	Check other empty fields by filling only username field
DESCRIPTION	Have to press on register button by filling only username field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only username field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' page. The form fields and their corresponding error messages are:</p> <ul style="list-style-type: none"> <li>Warehouse Staff ID: W006 (No error message)</li> <li>Employee Name: (Error message: *Employee Name Cannot Be Blank)</li> <li>Username: Disanayaka (Error message: *Password Cannot Be Blank)</li> <li>Password: (Error message: *Confirm Password Cannot Be Blank)</li> <li>Confirm Password: (Error message: *Confirm Password Cannot Be Blank)</li> <li>User Type: Warehouse Staff (No error message)</li> <li>Email: (Error message: *Email Cannot Be Blank)</li> <li>Telephone: (Error message: *Telephone Cannot Be Blank)</li> </ul> <p>At the bottom, there are 'Register' and 'Clear' buttons.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

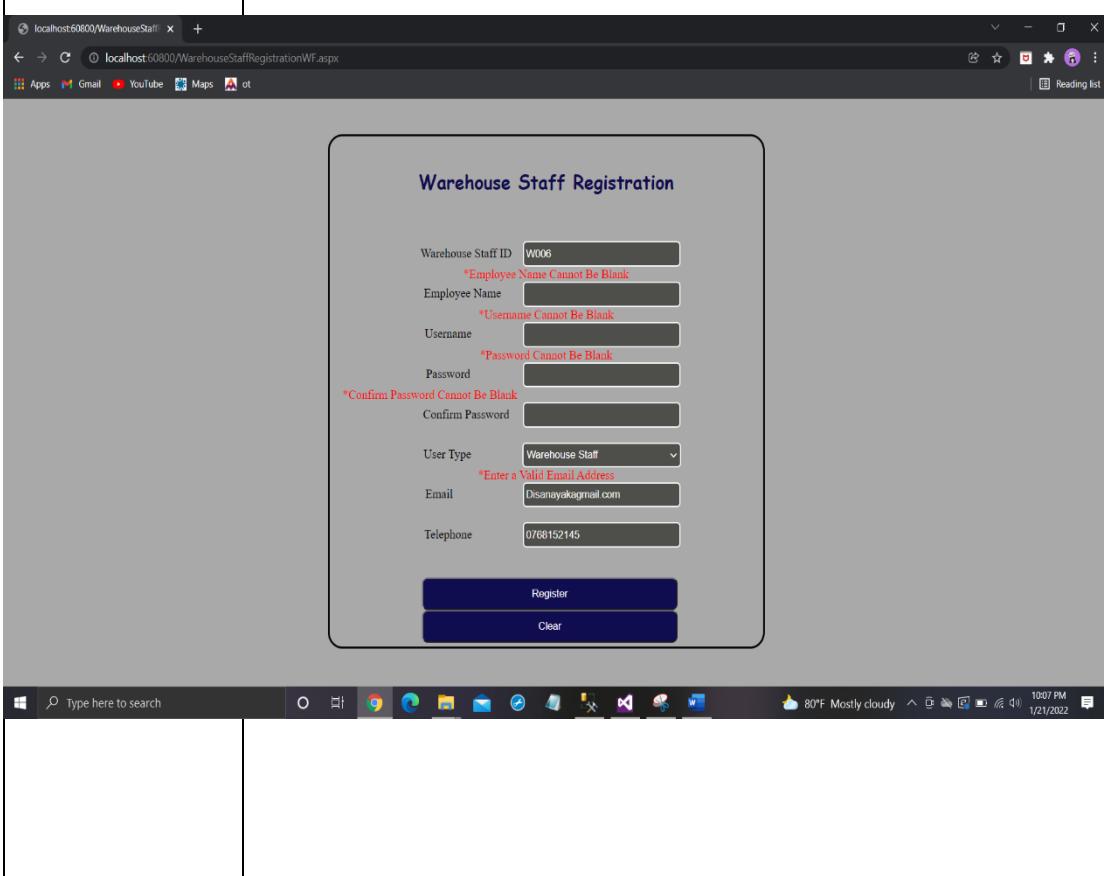
TEST CASE ID	TC3.4
TEST CASE NAME	Check other empty fields by filling only password field
DESCRIPTION	Have to press on register button by filling only password field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only password field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' page. The form fields are as follows:</p> <ul style="list-style-type: none"> <li>Warehouse Staff ID: W006 (no error message)</li> <li>Employee Name: (highlighted in red) "Employee Name Cannot Be Blank"</li> <li>Username: (highlighted in red) "Username Cannot Be Blank"</li> <li>Password: (highlighted in red) "Confirm Password Cannot Be Blank"</li> <li>Confirm Password: (highlighted in red) "Confirm Password Cannot Be Blank"</li> <li>User Type: Warehouse Staff (no error message)</li> <li>Email: (highlighted in red) "Email Cannot Be Blank"</li> <li>Telephone: (highlighted in red) "Telephone Cannot Be Blank"</li> </ul> <p>At the bottom are two buttons: 'Register' and 'Clear'.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

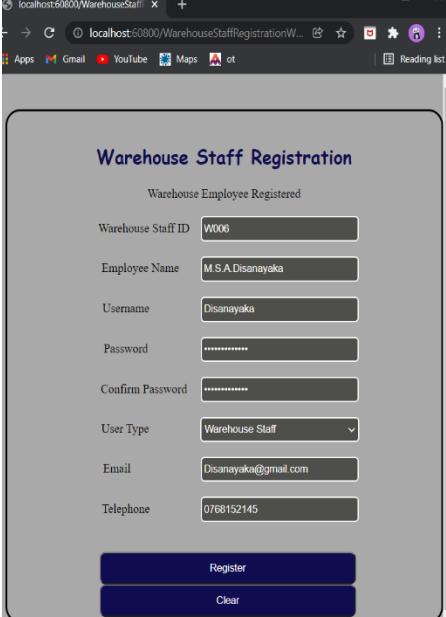
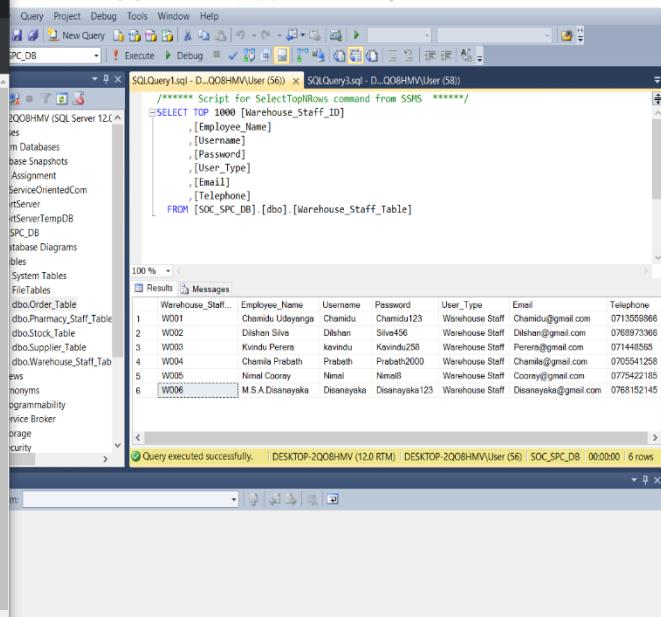
TEST CASE ID	TC3.5
TEST CASE NAME	Check other empty fields by filling only confirm password field
DESCRIPTION	Have to press on register button by filling only confirm password field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only confirm password field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' page. The form fields are as follows:</p> <ul style="list-style-type: none"> <li>Warehouse Staff ID: W006 (no error message)</li> <li>Employee Name: (highlighted in red, error message: *Employee Name Cannot Be Blank)</li> <li>Username: (highlighted in red, error message: *Username Cannot Be Blank)</li> <li>Password: (highlighted in red, error message: *Password Cannot Be Blank)</li> <li>Confirm Password: (highlighted in red, error message: *Password and Confirm Password Should Match)</li> <li>User Type: Warehouse Staff (no error message)</li> <li>Email: (highlighted in red, error message: *Email Cannot Be Blank)</li> <li>Telephone: (highlighted in red, error message: *Telephone Cannot Be Blank)</li> </ul> <p>At the bottom, there are two buttons: 'Register' (dark blue) and 'Clear' (light blue).</p>
CONCLUSION	The expected error messages were displayed on each text boxes and on confirm password an error message displayed as Confirm Password and Password should match
STATUS (PASS/FAIL)	Fail

TEST CASE ID	TC3.6
TEST CASE NAME	Check other empty fields by filling only email field
DESCRIPTION	Have to press on register button by filling only email field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only email field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' page. The 'Employee Name' field has an error message: *Employee Name Cannot Be Blank. The 'Username' field has an error message: *Username Cannot Be Blank. The 'Password' field has an error message: *Password Cannot Be Blank. The 'Confirm Password' field has an error message: *Confirm Password Cannot Be Blank. All other fields (Warehouse Staff ID, Email, Telephone, User Type) are filled correctly. The registration button is at the bottom.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

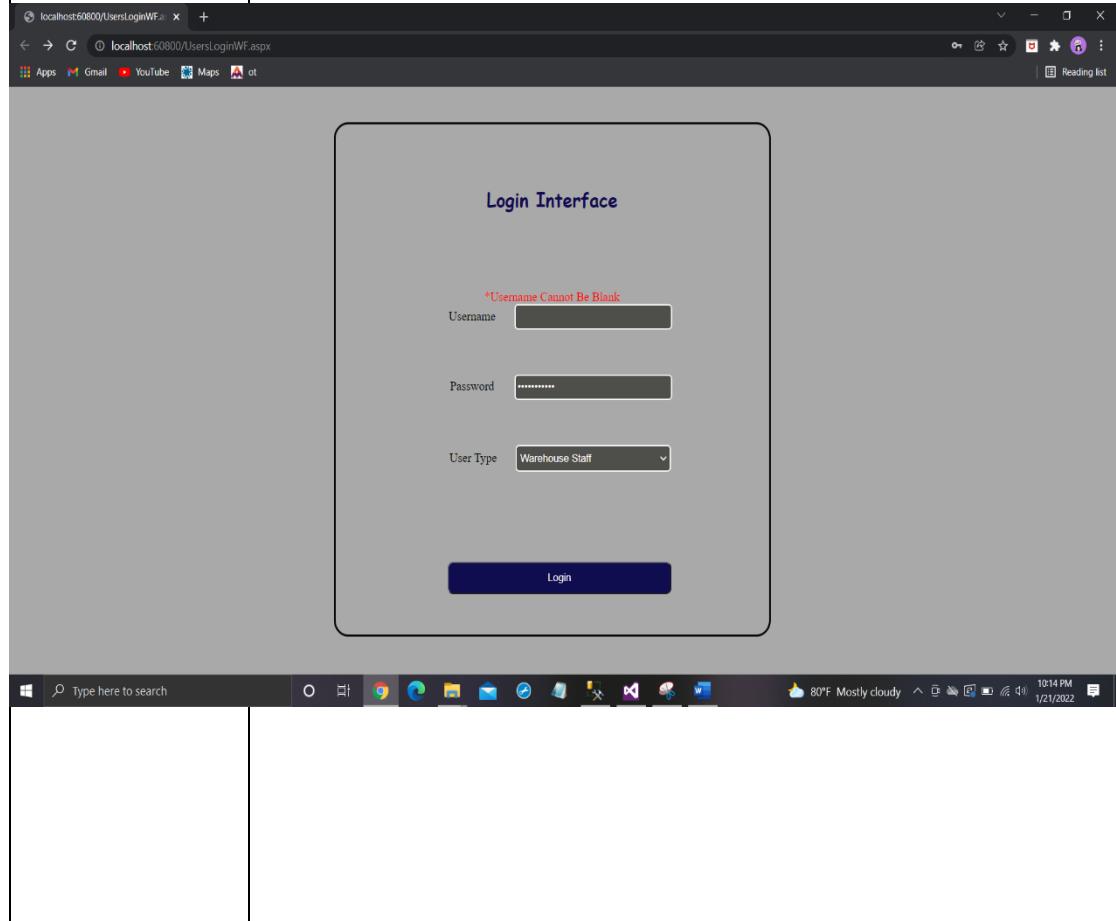
TEST CASE ID	TC3.7
TEST CASE NAME	Check other empty fields by filling only telephone field
DESCRIPTION	Have to press on register button by filling only telephone field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only telephone field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect warehouse staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' form. The 'Employee Name' field contains 'W006' and has a red validation message: '*Employee Name Cannot Be Blank'. The 'Username' field has a red validation message: '*Username Cannot Be Blank'. The 'Password' field has a red validation message: '*Password Cannot Be Blank'. The 'Email' field has a red validation message: '*Email Cannot Be Blank'. The 'Telephone' field contains '0768152145'. Below the form are two buttons: 'Register' and 'Clear'.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

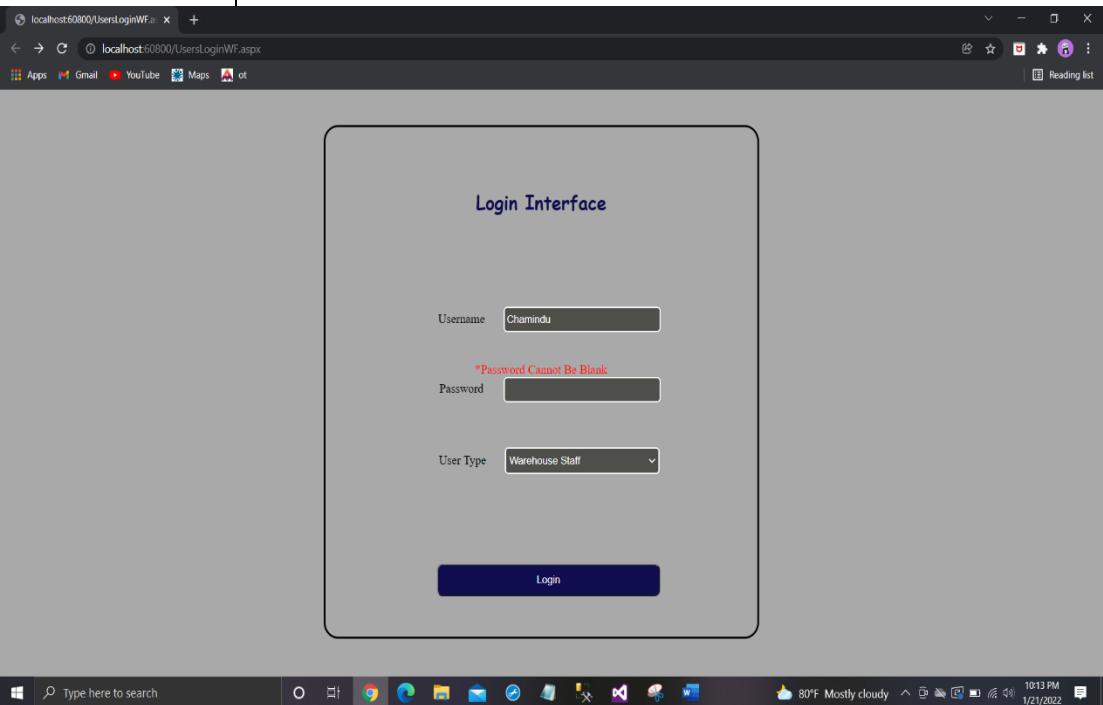
TEST CASE ID	TC3.8
TEST CASE NAME	Check matching password and confirm password
DESCRIPTION	Have to enter a password and a different confirm password
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only telephone field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Password: Disanayaka123 Confirm password: Disanayaka
EXPECTED RESULT	Should display an Error Message on Confirm Password field as password and confirm password should match
ACTUAL RESULT	 <p>The screenshot shows the 'Warehouse Staff Registration' form. The fields and their corresponding error messages are:</p> <ul style="list-style-type: none"> <li>Employee Name: *Employee Name Cannot Be Blank</li> <li>Username: *Username Cannot Be Blank</li> <li>Email: *Email Cannot Be Blank</li> </ul>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

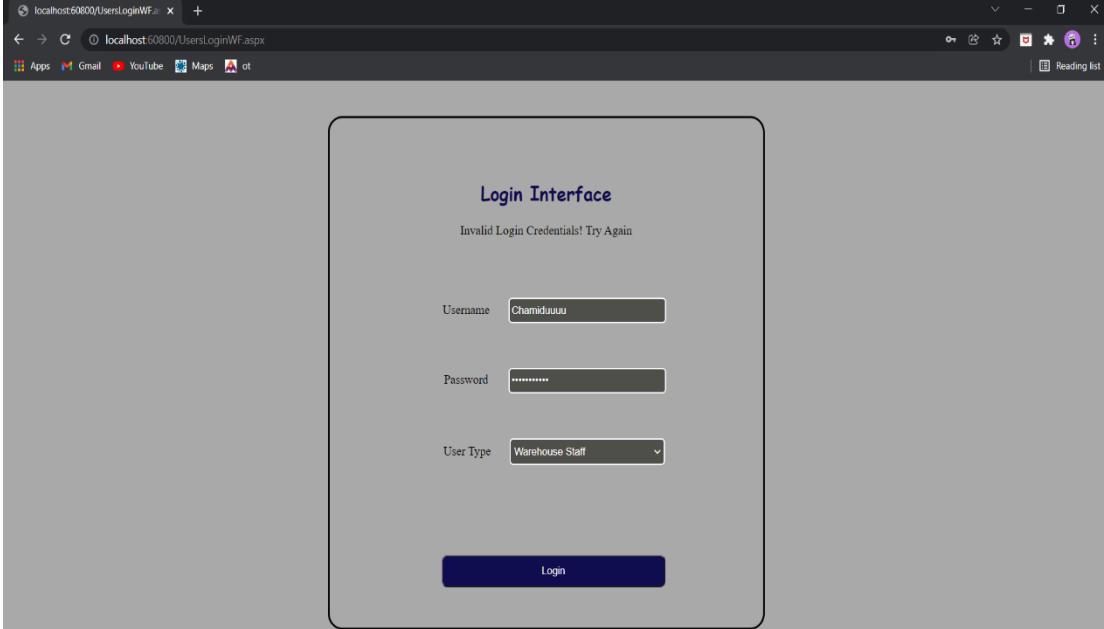
TEST CASE ID	TC3.9
TEST CASE NAME	Check email format
DESCRIPTION	Have to enter an email with an incorrect email format
TEST STEPS	1.Fill only email field (without @ symbol) 2.Press on register button
TEST DATA	Email: Disanayakagmail.com
EXPECTED RESULT	Should display an Error Message on email field as Invalid Email
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

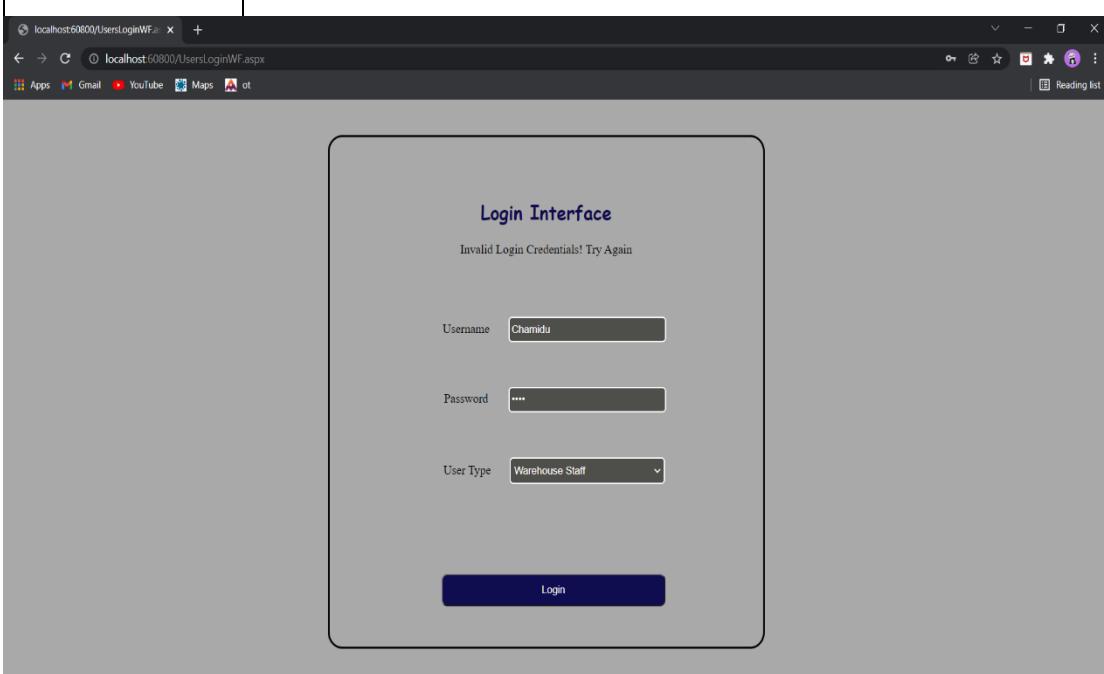
TEST CASE ID	TC3.10																																																								
TEST CASE NAME	Check Registration working without any empty fields																																																								
DESCRIPTION	Have to fill all fields and press register button																																																								
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all fields</li> <li>2.Press on register button</li> </ol>																																																								
TEST DATA	Employee Name: M.S.A Disanayaka Username: Disanayaka Password: Disanayaka123 Confirm password: Disanayaka 123 User Type: Warehouse Staff Email: Disanayaka@gmail.com Telephone: 0768152145																																																								
EXPECTED RESULT	Should display a message as Warehouse Staff Registered Successfully																																																								
ACTUAL RESULT	  <pre>***** Script for SelectTopNRows command from SSMS ***** SELECT TOP 1000 [Warehouse_Staff_ID] ,[Employee_Name] ,[Username] ,[Password] ,[User_Type] ,[Email] ,[Telephone] FROM [SOC_SPC_DB].[dbo].[Warehouse_Staff_Table]</pre> <table border="1"> <thead> <tr> <th></th> <th>Warehouse_Staff_ID</th> <th>Employee_Name</th> <th>Username</th> <th>Password</th> <th>User_Type</th> <th>Email</th> <th>Telephone</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>W001</td> <td>Chamidu Udayanga</td> <td>Chamidu</td> <td>Chamidu123</td> <td>Warehouse Staff</td> <td>Chamidu@gmail.com</td> <td>0713559866</td> </tr> <tr> <td>2</td> <td>W002</td> <td>Dilshan Silva</td> <td>Dilshan</td> <td>Silva456</td> <td>Warehouse Staff</td> <td>Dilshan@gmail.com</td> <td>0789973366</td> </tr> <tr> <td>3</td> <td>W003</td> <td>Kvindu Perera</td> <td>kvindu</td> <td>Kvindu258</td> <td>Warehouse Staff</td> <td>Perera@gmail.com</td> <td>0714458565</td> </tr> <tr> <td>4</td> <td>W004</td> <td>Chamila Prabath</td> <td>Prabath</td> <td>Prabath2000</td> <td>Warehouse Staff</td> <td>Chamila@gmail.com</td> <td>0705512568</td> </tr> <tr> <td>5</td> <td>W005</td> <td>Nimal Cooray</td> <td>Nimal</td> <td>Nimal8</td> <td>Warehouse Staff</td> <td>Cooray@gmail.com</td> <td>0775422185</td> </tr> <tr> <td>6</td> <td>W006</td> <td>M S A Disanayaka</td> <td>Disanayaka</td> <td>Disanayaka123</td> <td>Warehouse Staff</td> <td>Disanayaka@gmail.com</td> <td>0768152145</td> </tr> </tbody> </table>		Warehouse_Staff_ID	Employee_Name	Username	Password	User_Type	Email	Telephone	1	W001	Chamidu Udayanga	Chamidu	Chamidu123	Warehouse Staff	Chamidu@gmail.com	0713559866	2	W002	Dilshan Silva	Dilshan	Silva456	Warehouse Staff	Dilshan@gmail.com	0789973366	3	W003	Kvindu Perera	kvindu	Kvindu258	Warehouse Staff	Perera@gmail.com	0714458565	4	W004	Chamila Prabath	Prabath	Prabath2000	Warehouse Staff	Chamila@gmail.com	0705512568	5	W005	Nimal Cooray	Nimal	Nimal8	Warehouse Staff	Cooray@gmail.com	0775422185	6	W006	M S A Disanayaka	Disanayaka	Disanayaka123	Warehouse Staff	Disanayaka@gmail.com	0768152145
	Warehouse_Staff_ID	Employee_Name	Username	Password	User_Type	Email	Telephone																																																		
1	W001	Chamidu Udayanga	Chamidu	Chamidu123	Warehouse Staff	Chamidu@gmail.com	0713559866																																																		
2	W002	Dilshan Silva	Dilshan	Silva456	Warehouse Staff	Dilshan@gmail.com	0789973366																																																		
3	W003	Kvindu Perera	kvindu	Kvindu258	Warehouse Staff	Perera@gmail.com	0714458565																																																		
4	W004	Chamila Prabath	Prabath	Prabath2000	Warehouse Staff	Chamila@gmail.com	0705512568																																																		
5	W005	Nimal Cooray	Nimal	Nimal8	Warehouse Staff	Cooray@gmail.com	0775422185																																																		
6	W006	M S A Disanayaka	Disanayaka	Disanayaka123	Warehouse Staff	Disanayaka@gmail.com	0768152145																																																		
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear																																																								
STATUS (PASS/FAIL)	Pass																																																								

## TC4 - Warehouse Staff Login

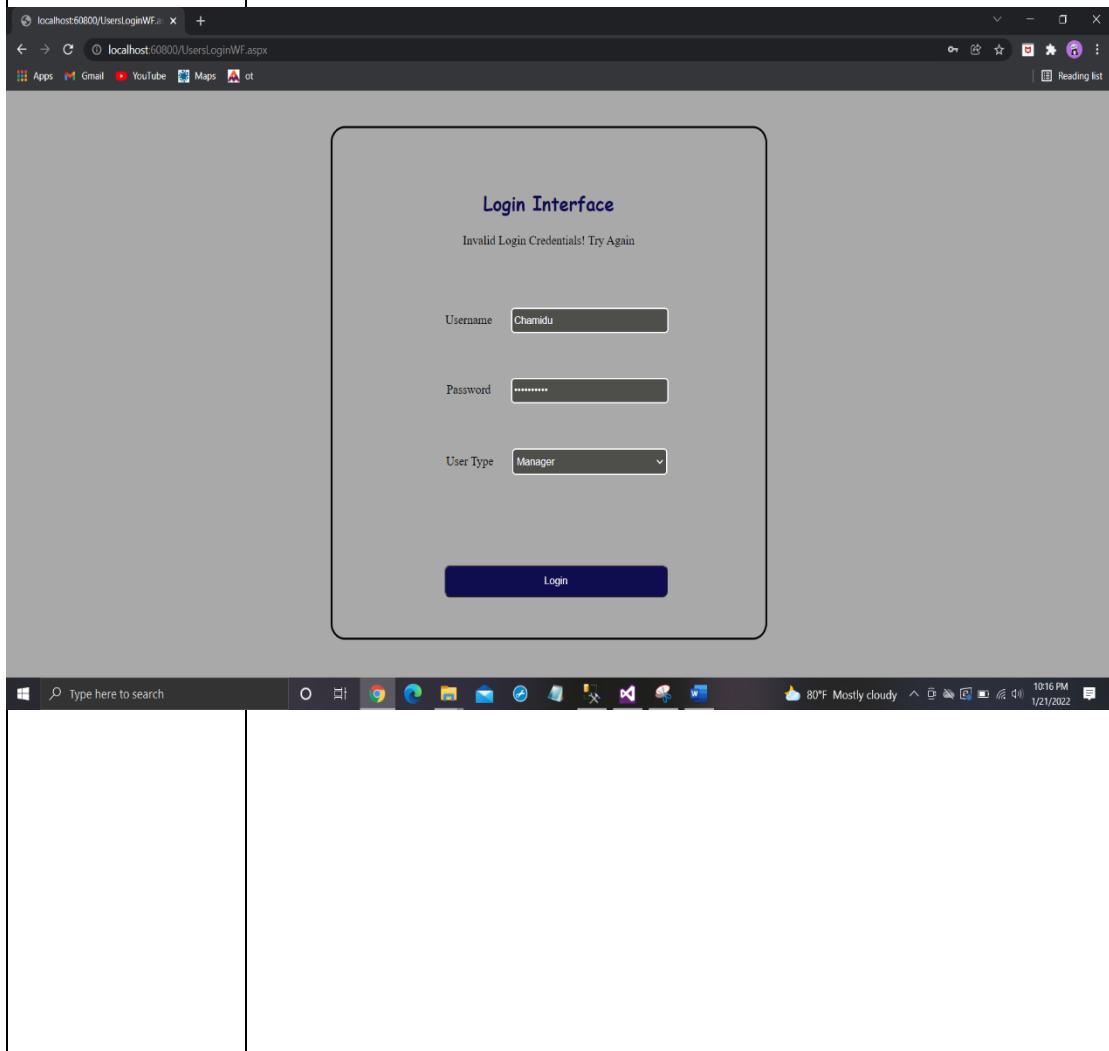
TEST CASE ID	TC4.1
TEST CASE NAME	Check with empty username field
DESCRIPTION	Have to enter only a password and press Login as button
TEST STEPS	1.Fill only the password field when the interface is loaded 2.Press on login button
TEST DATA	Password: Chamidu123
EXPECTED RESULT	Should display an Error Message on username field as username can't be blank
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/UsersLoginWF.aspx". The page is titled "Login Interface". It contains three input fields: "Username" (empty), "Password" (filled with dots), and "User Type" (set to "Warehouse Staff"). A red error message "Username Cannot Be Blank" is displayed above the empty "Username" field. A "Login" button is at the bottom.
CONCLUSION	The expected an error message displayed on the username text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC4.2
TEST CASE NAME	Check with empty password field
DESCRIPTION	Have to enter only a username and press Login as button
TEST STEPS	<p>1.Fill only the username field when the interface is loaded</p> <p>2.Press on login button</p>
TEST DATA	Username: Chamidu
EXPECTED RESULT	Should display an Error Message on password field as password can't be blank
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/UsersLoginWF.aspx". The page is titled "Login Interface". It has three input fields: "Username" with value "Chamidu", "Password" which is empty and highlighted with a red border, and "User Type" with value "Warehouse Staff". Below the fields is a large blue "Login" button. A red error message "Password Cannot Be Blank" is displayed above the empty password field.
CONCLUSION	The expected an error message displayed on the password text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

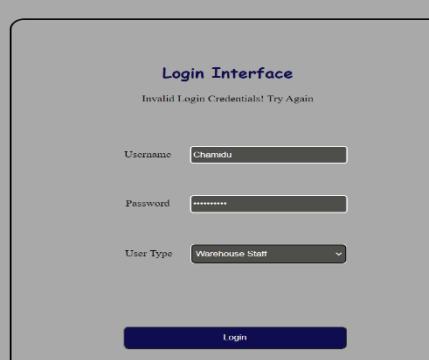
TEST CASE ID	TC4.3
TEST CASE NAME	Check with incorrect warehouse staff username
DESCRIPTION	Have to enter incorrect username with correct warehouse staff password and warehouse staff user type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with wrong username</li> <li>2.Fill password field with a correct password</li> <li>3.Select user type as warehouse staff</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Chamiduuu Password: Chamidu123
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/UsersLoginWF.aspx". The page is titled "Login Interface" and displays the message "Invalid Login Credentials! Try Again". It contains three input fields: "Username" with the value "Chamiduuu", "Password" with a masked value, and "User Type" set to "Warehouse Staff". A blue "Login" button is at the bottom.
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC4.4
TEST CASE NAME	Check with incorrect warehouse staff password
DESCRIPTION	Have to enter incorrect password with correct warehouse staff username and warehouse staff user type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill password field with wrong password</li> <li>2.Fill username field with a correct username</li> <li>3.Select user type as warehouse staff</li> <li>4.Press on Login button</li> </ol>
TEST DATA	password: 123
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC4.5
TEST CASE NAME	Check with incorrect user type
DESCRIPTION	Have to enter correct warehouse staff username and password without warehouse staff User Type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with correct username</li> <li>2.Fill password field with a correct password</li> <li>3.Select user type as Manager</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Chamidu Password: Chamidu123 User Type: Manager
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	

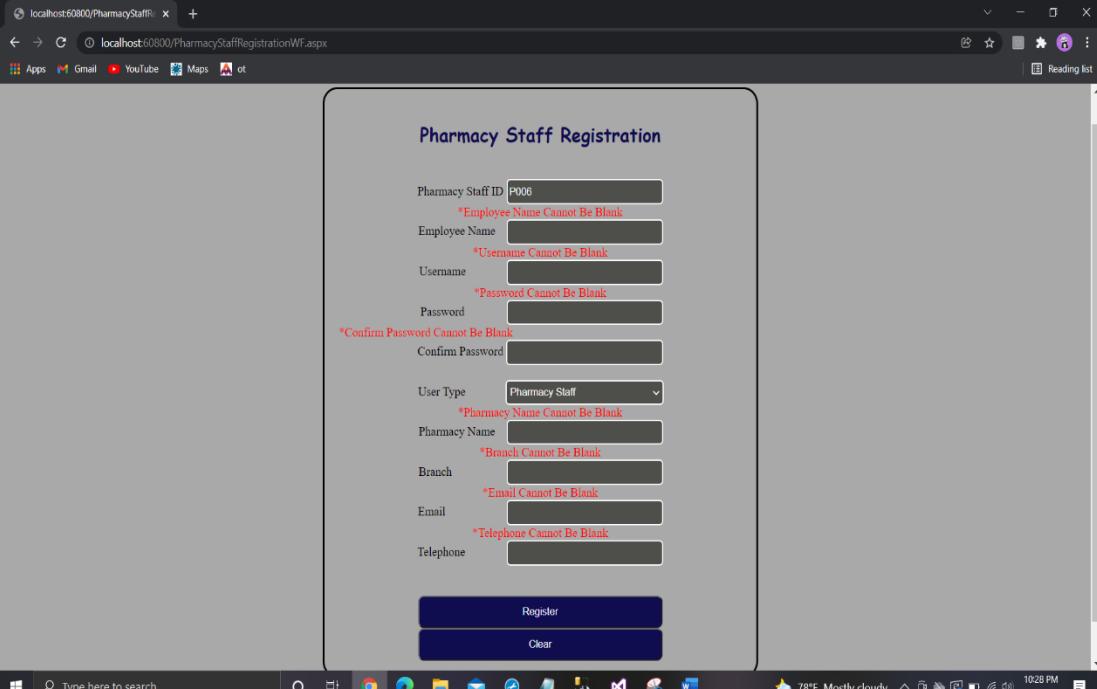


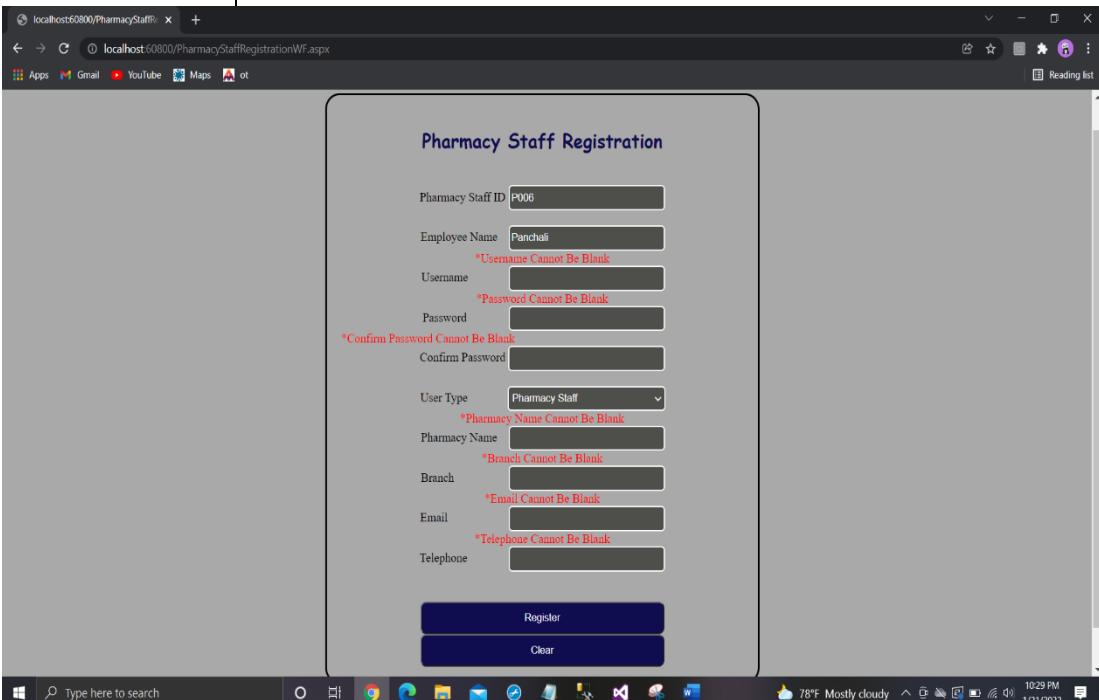
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass
TEST CASE ID	TC4.6
TEST CASE NAME	Check Login working without any empty field and with correct login credentials
DESCRIPTION	Have to enter correct Manager warehouse staff, password and warehouse staff User Type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with correct username</li> <li>2.Fill password field with a correct password</li> <li>3.Select user type as warehouse staff</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Chamidu Password: Chamidu123 User Type: warehouse staff
EXPECTED RESULT	Should display the Stock Update interface
ACTUAL RESULT	

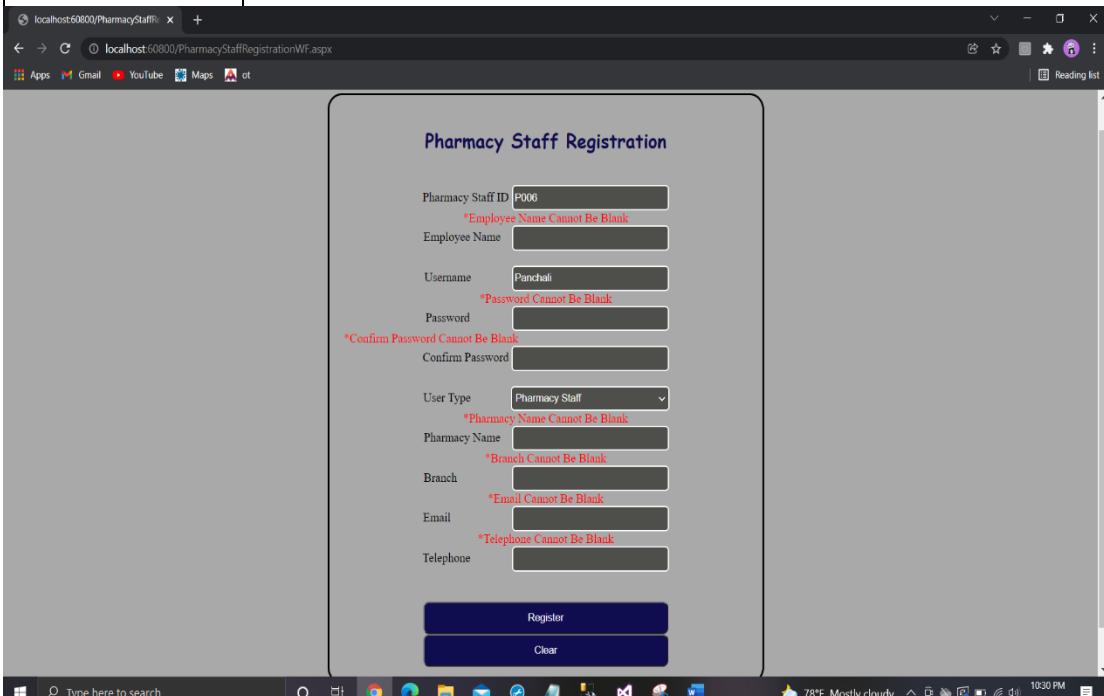


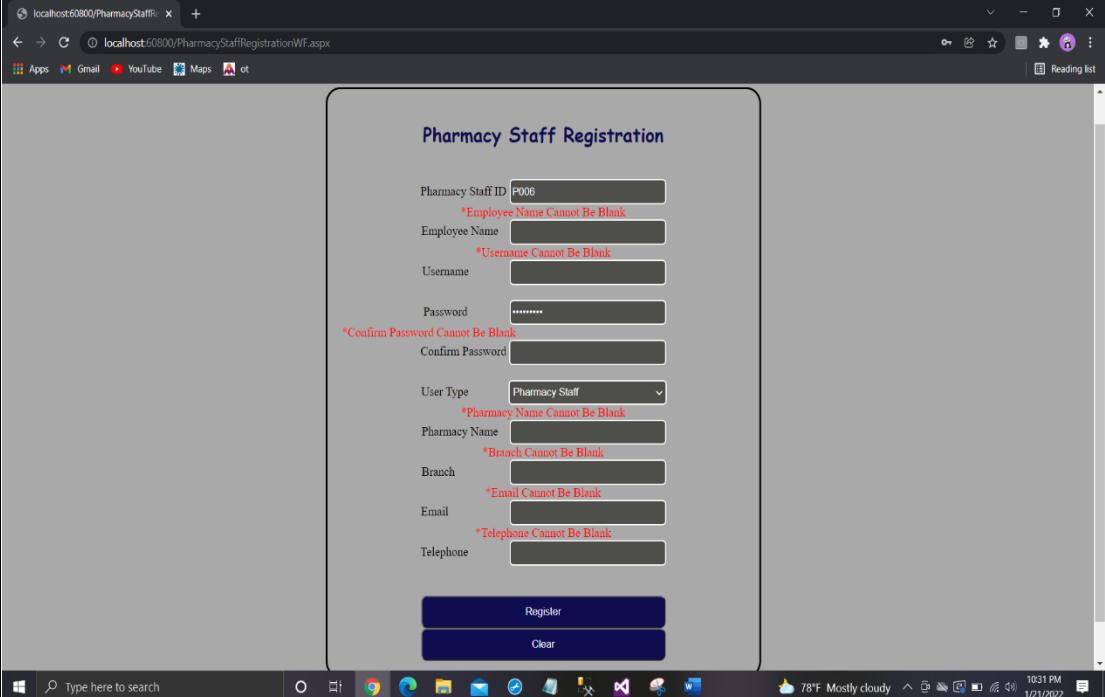

CONCLUSION	The expected an interface is displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

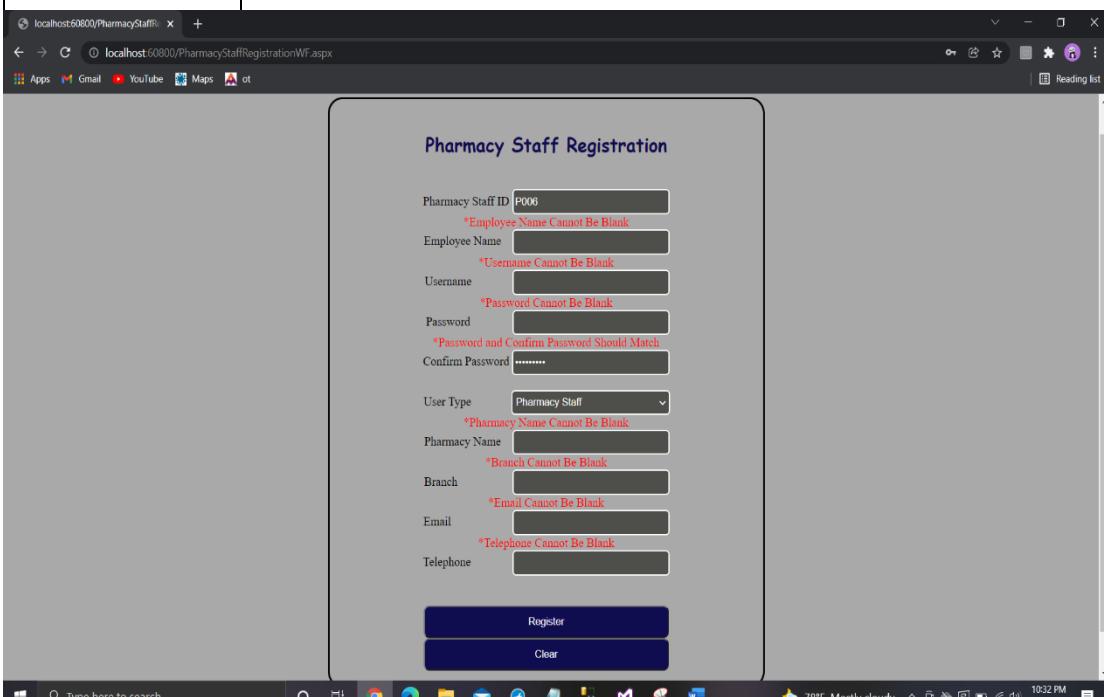
## TC5 - Pharmacy Staff Registration

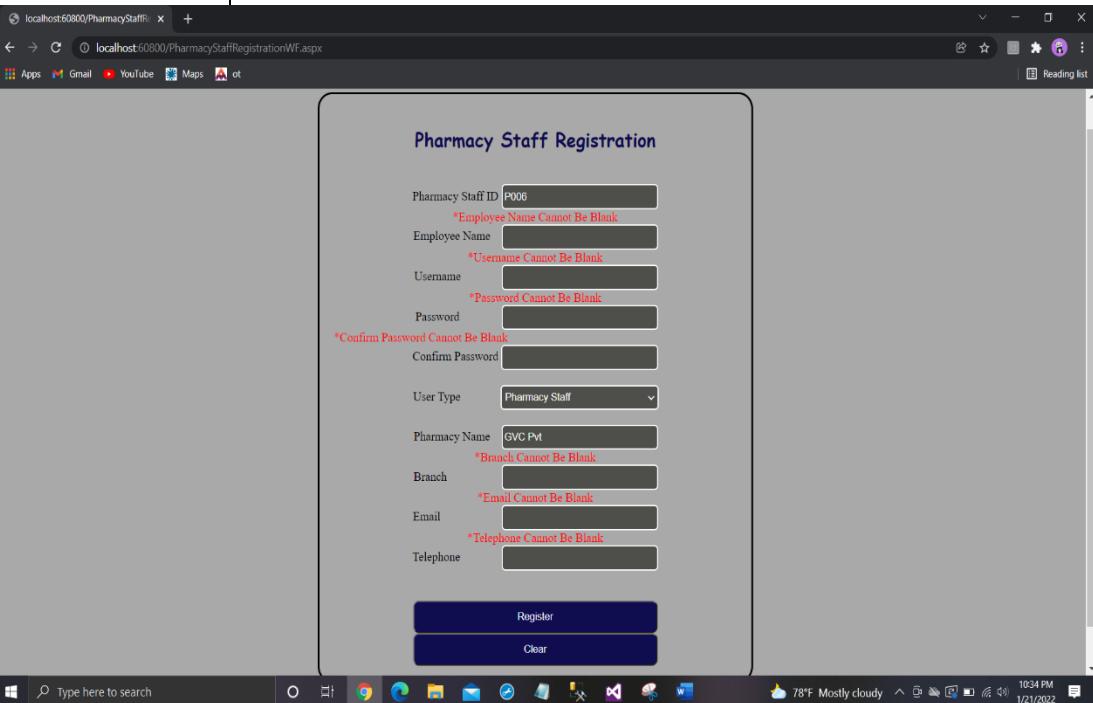
TEST CASE ID	TC5.1
TEST CASE NAME	Check Register empty fields
DESCRIPTION	Have to press on register button without filling any fields
TEST STEPS	Just press on register button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Pharmacy Staff id and user type fields as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

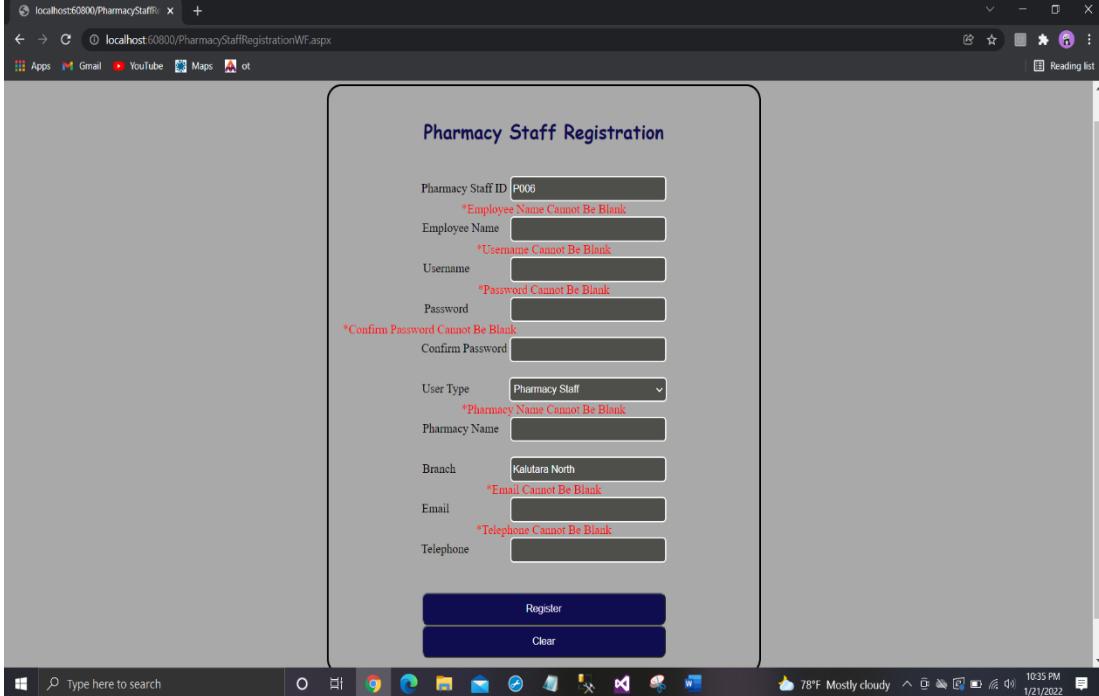
TEST CASE ID	TC5.2
TEST CASE NAME	Check other empty fields by filling only employee name field
DESCRIPTION	Have to press on register button by filling only employee name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only Employee Name field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Pharmacy Staff Registration' page. The 'Employee Name' field contains 'Panchali'. The 'Username' field has an error message: '*Username Cannot Be Blank'. The 'Password' and 'Confirm Password' fields both have error messages: '*Password Cannot Be Blank' and '*Confirm Password Cannot Be Blank'. The 'User Type' dropdown is set to 'Pharmacy Staff'. The 'Pharmacy Name' field has an error message: '*Pharmacy Name Cannot Be Blank'. The 'Branch' field has an error message: '*Branch Cannot Be Blank'. The 'Email' field has an error message: '*Email Cannot Be Blank'. The 'Telephone' field has an error message: '*Telephone Cannot Be Blank'. The 'Register' button is at the bottom.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

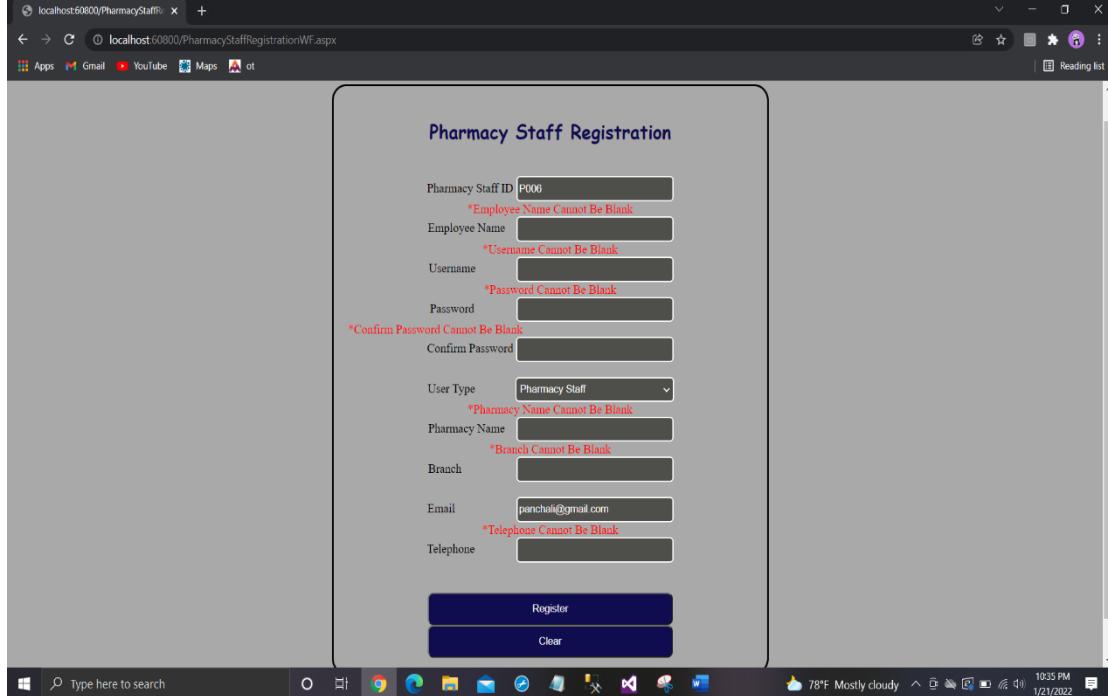
TEST CASE ID	TC5.3
TEST CASE NAME	Check other empty fields by filling only username field
DESCRIPTION	Have to press on register button by filling only username field
TEST STEPS	1.Fill only username field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

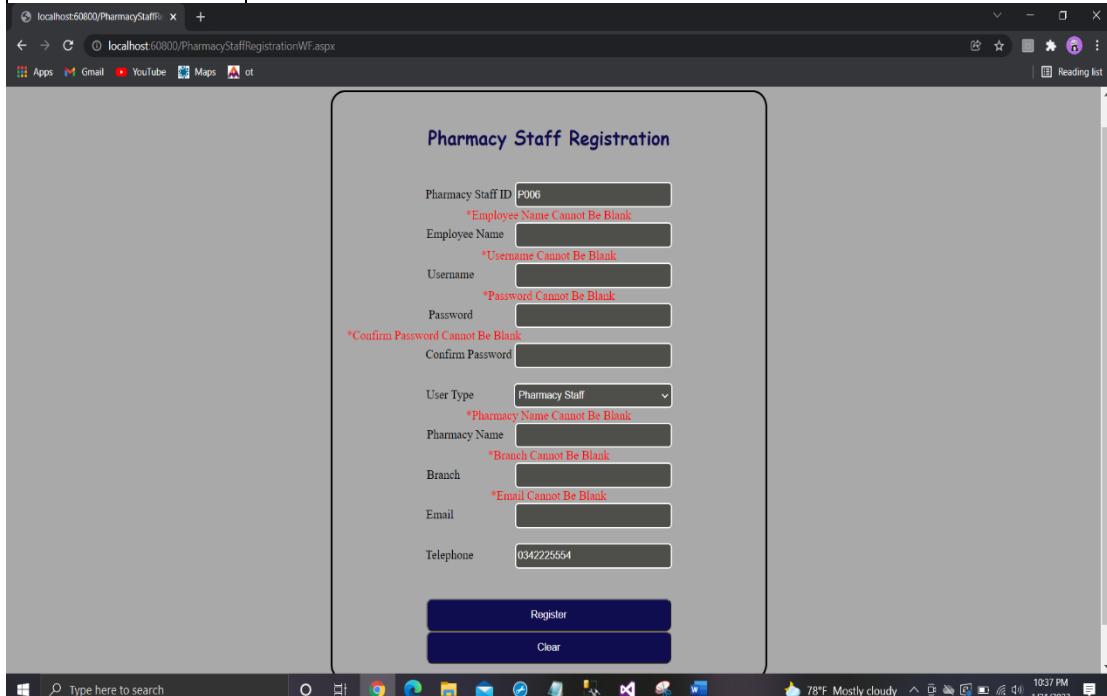
TEST CASE ID	TC5.4
TEST CASE NAME	Check other empty fields by filling only password field
DESCRIPTION	Have to press on register button by filling only password field
TEST STEPS	1.Fill only password field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

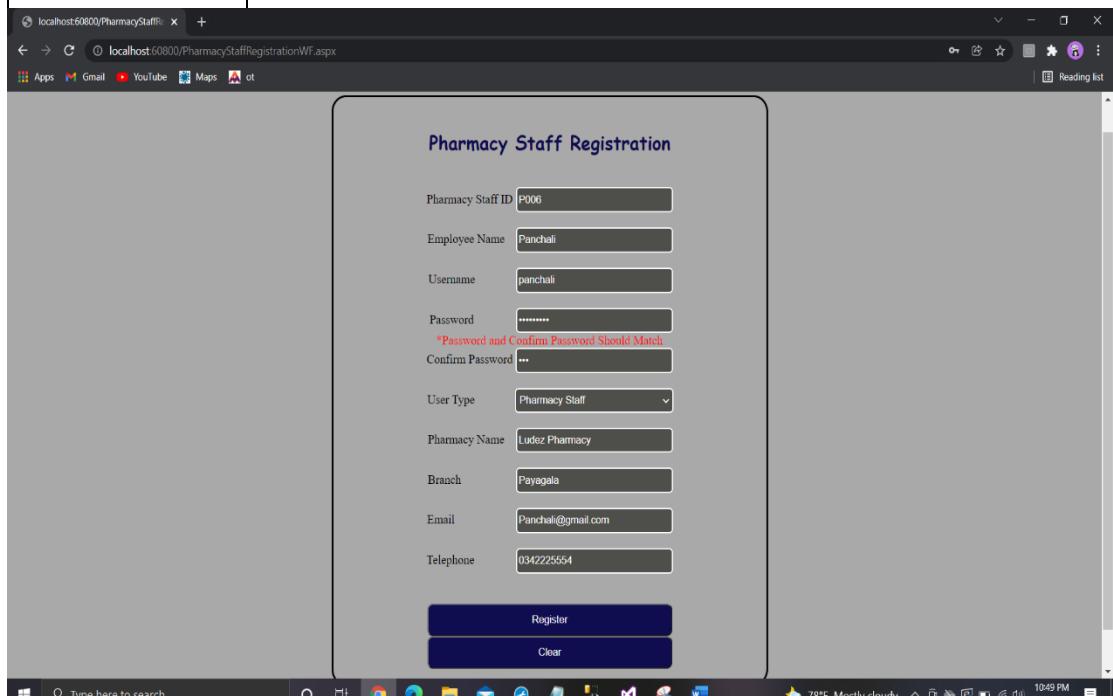
TEST CASE ID	TC5.5
TEST CASE NAME	Check other empty fields by filling only confirm password field
DESCRIPTION	Have to press on register button by filling only confirm password field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only confirm password field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

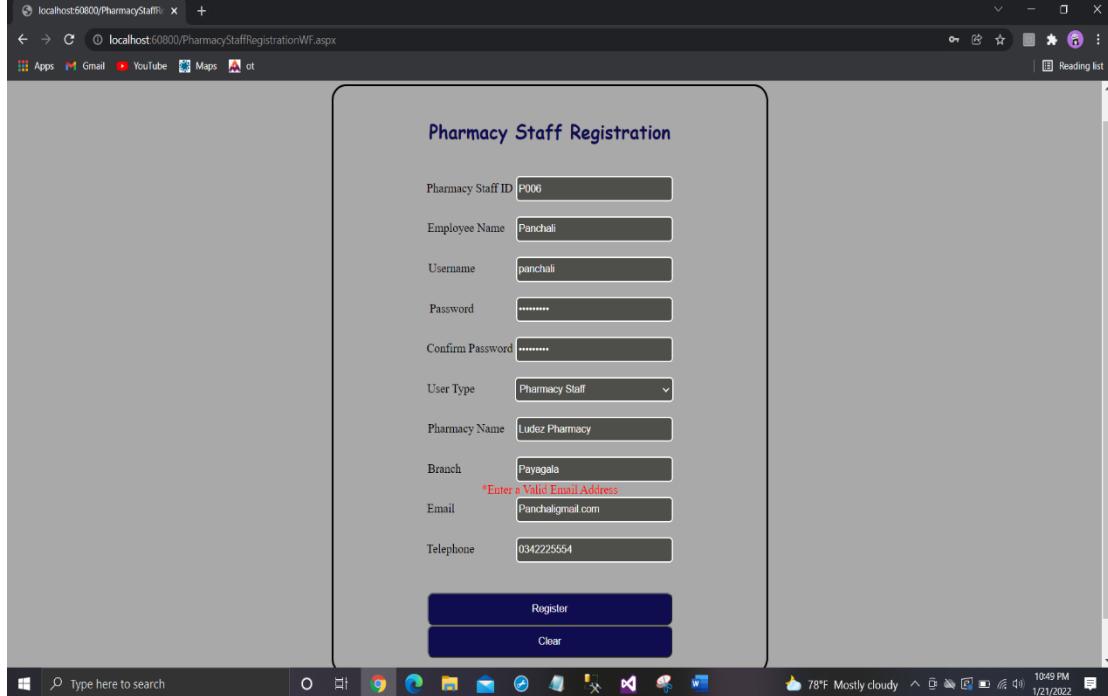
TEST CASE ID	TC5.6
TEST CASE NAME	Check other empty fields by filling only pharmacy name field
DESCRIPTION	Have to press on register button by filling only pharmacy name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only pharmacy name field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

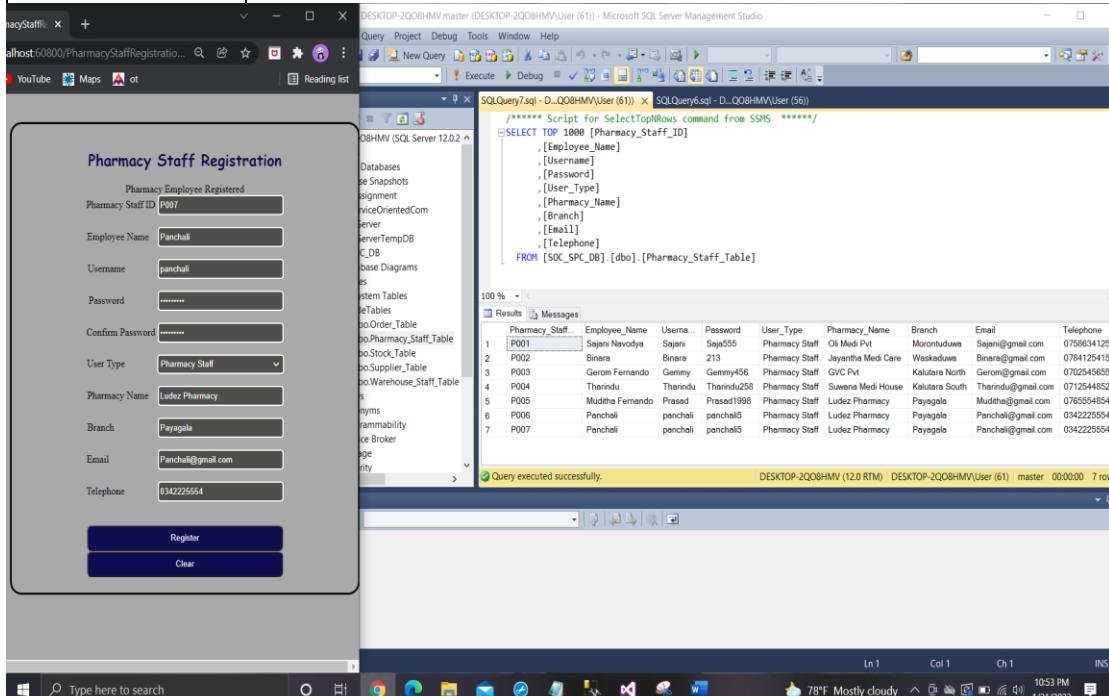
TEST CASE ID	TC5.7
TEST CASE NAME	Check other empty fields by filling only branch field
DESCRIPTION	Have to press on register button by filling only branch field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only branch field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	 <p>The screenshot shows the 'Pharmacy Staff Registration' form. The 'Branch' field contains 'Kalutara North'. All other fields (Employee Name, Username, Password, Confirm Password, Pharmacy Name, Email, Telephone) are empty. Red validation error messages are displayed above each of these empty fields: 'Employee Name Cannot Be Blank', 'Username Cannot Be Blank', 'Password Cannot Be Blank', 'Confirm Password Cannot Be Blank', 'Pharmacy Name Cannot Be Blank', 'Email Cannot Be Blank', and 'Telephone Cannot Be Blank'. The 'User Type' dropdown is set to 'Pharmacy Staff'. The 'Register' button is at the bottom.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC5.8
TEST CASE NAME	Check other empty fields by filling only email field
DESCRIPTION	Have to press on register button by filling only email field
TEST STEPS	1.Fill only email field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

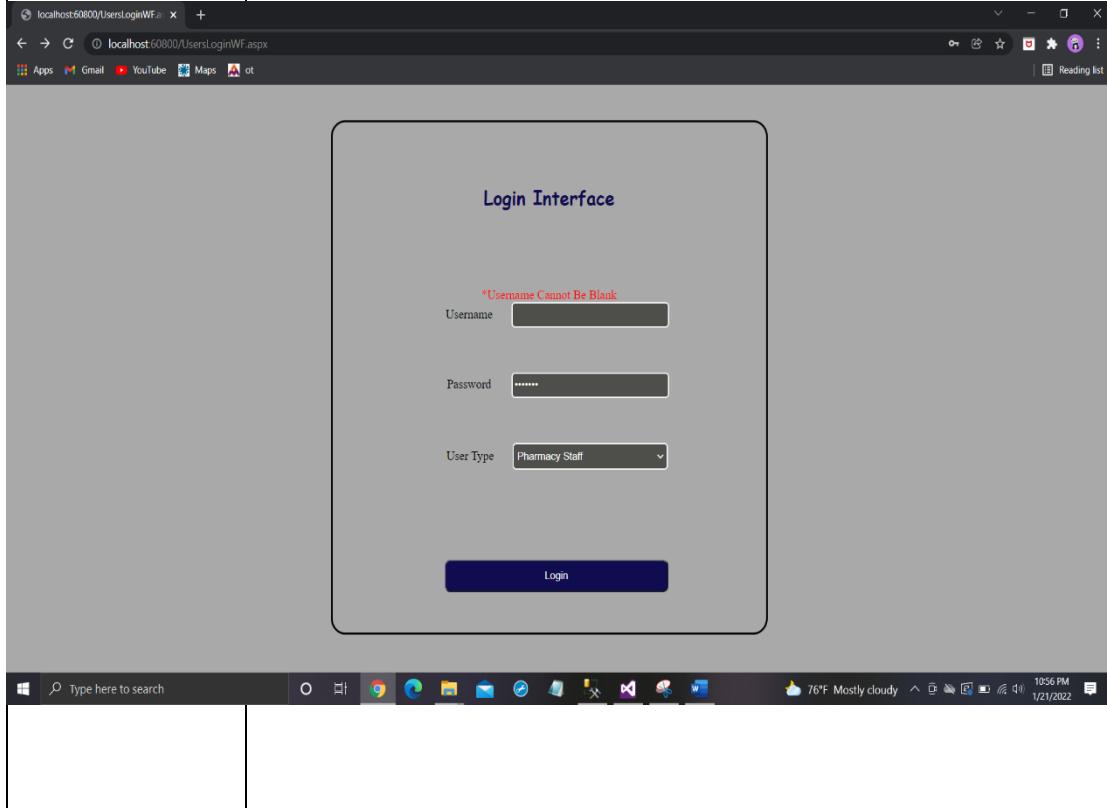
TEST CASE ID	TC5.9
TEST CASE NAME	Check other empty fields by filling only telephone field
DESCRIPTION	Have to press on register button by filling only telephone field
TEST STEPS	1.Fill only telephone field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name expect Pharmacy Staff id and user type fields
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

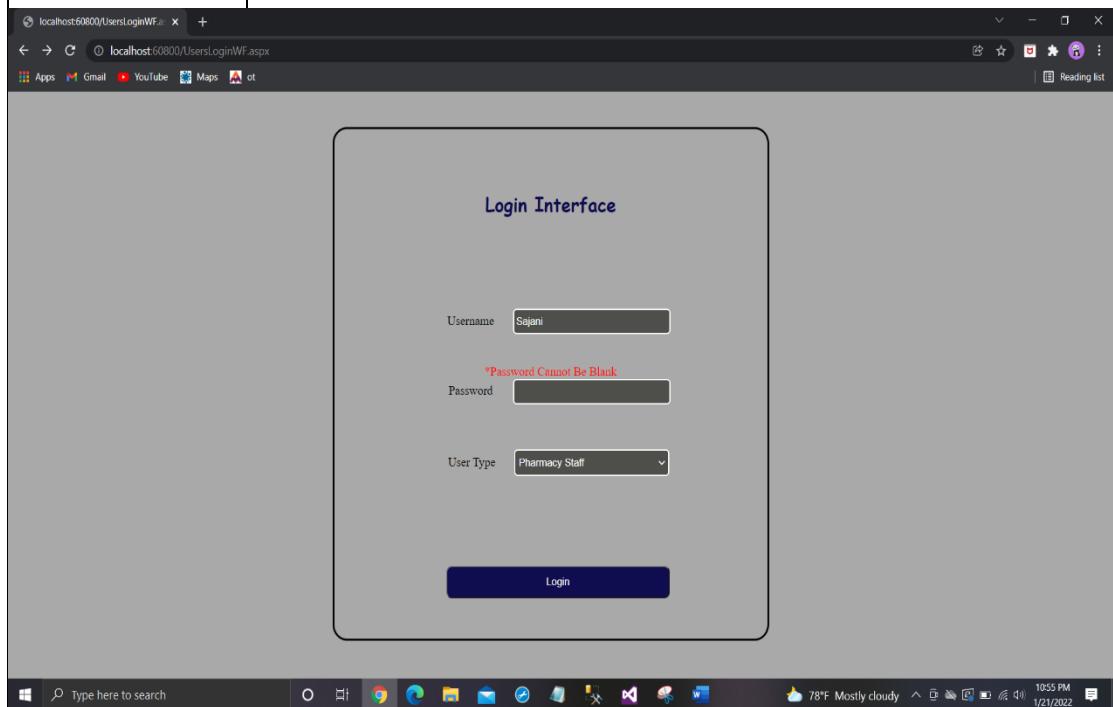
TEST CASE ID	TC5.10
TEST CASE NAME	Check matching password and confirm password
DESCRIPTION	Have to enter a password and a different confirm password
TEST STEPS	1.Fill password and a different confirm password field 2.Press on register button
TEST DATA	Password: panchali5 Confirm password: 123
EXPECTED RESULT	Should display an Error Message on Confirm Password field as password and confirm password should match
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/PharmacyStaffReg...". The main title bar says "localhost:60800/PharmacyStaffRegistrationWF.aspx". Below the title bar is a toolbar with icons for Apps, Gmail, YouTube, Maps, and other browser controls. The main content area is a form titled "Pharmacy Staff Registration". The form fields are: Pharmacy Staff ID (P006), Employee Name (Panchali), Username (panchali), Password (redacted), Confirm Password (redacted) - which has a red error message "*Password and Confirm Password Should Match", User Type (Pharmacy Staff dropdown), Pharmacy Name (Ludez Pharmacy), Branch (Payagala), Email (Panchali@gmail.com), and Telephone (0342225554). At the bottom are two buttons: "Register" and "Clear".
CONCLUSION	The expected error message was displayed on confirm password text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

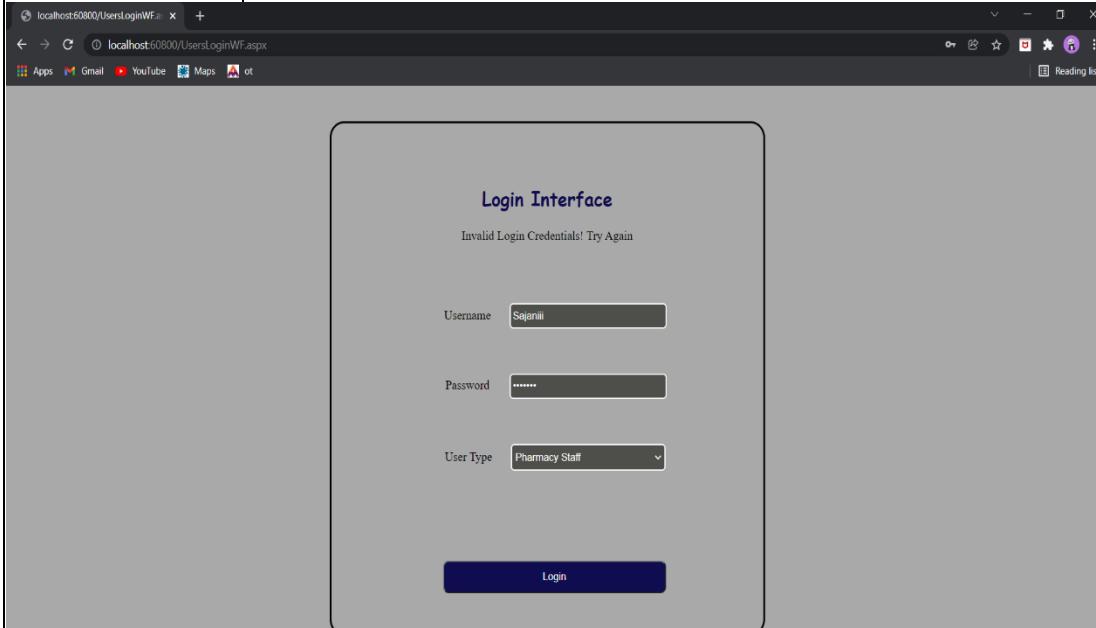
TEST CASE ID	TC5.11
TEST CASE NAME	Check email format
DESCRIPTION	Have to enter an email with an incorrect email format
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only email field (without @ symbol)</li> <li>2.Press on register button</li> </ol>
TEST DATA	Email: Panchal@gmail.com
EXPECTED RESULT	Should display an Error Message on email field as Invalid Email
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

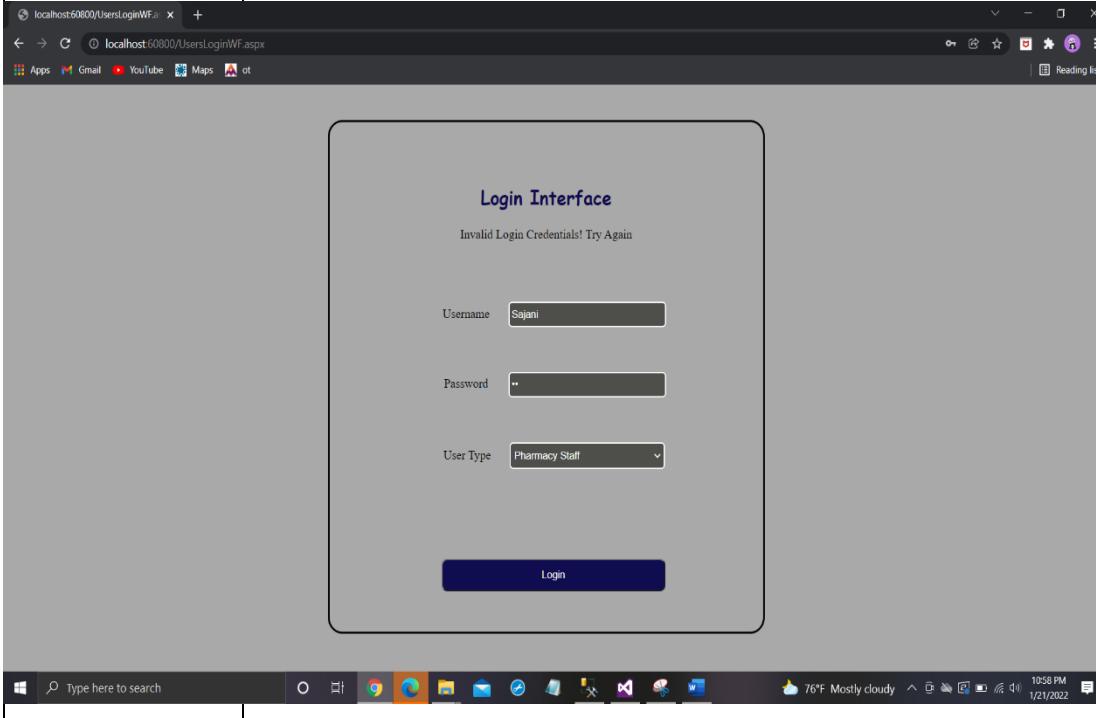
TEST CASE ID	TC3.10
TEST CASE NAME	Check Registration working without any empty fields
DESCRIPTION	Have to fill all fields and press register button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all fields</li> <li>2.Press on register button</li> </ol>
TEST DATA	<p>Employee Name: M.S.A Disanayaka      Username: Disanayaka      Password: Disanayaka123      Confirm password: Disanayaka 123      User Type: Warehouse Staff      Email: Disanayaka@gmail.com      Telephone: 0768152145</p>
EXPECTED RESULT	Should display a message as Pharmacy Staff Registered Successfully
ACTUAL RESULT	
 <p>The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, there is a registration form titled 'Pharmacy Staff Registration' with fields for Employee Name, Username, Password, Confirm Password, User Type, Pharmacy Name, Branch, Email, and Telephone. The 'User Type' dropdown is set to 'Pharmacy Staff'. On the right, a query window displays a result set from a SELECT statement. The results show 7 rows of data for various staff members, including their Employee_ID, Employee_Name, Username, Password, User_Type, Pharmacy_Name, Branch, Email, and Telephone. Below the results, a message says 'Query executed successfully.'</p>	
CONCLUSION	The expected successfully registered message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

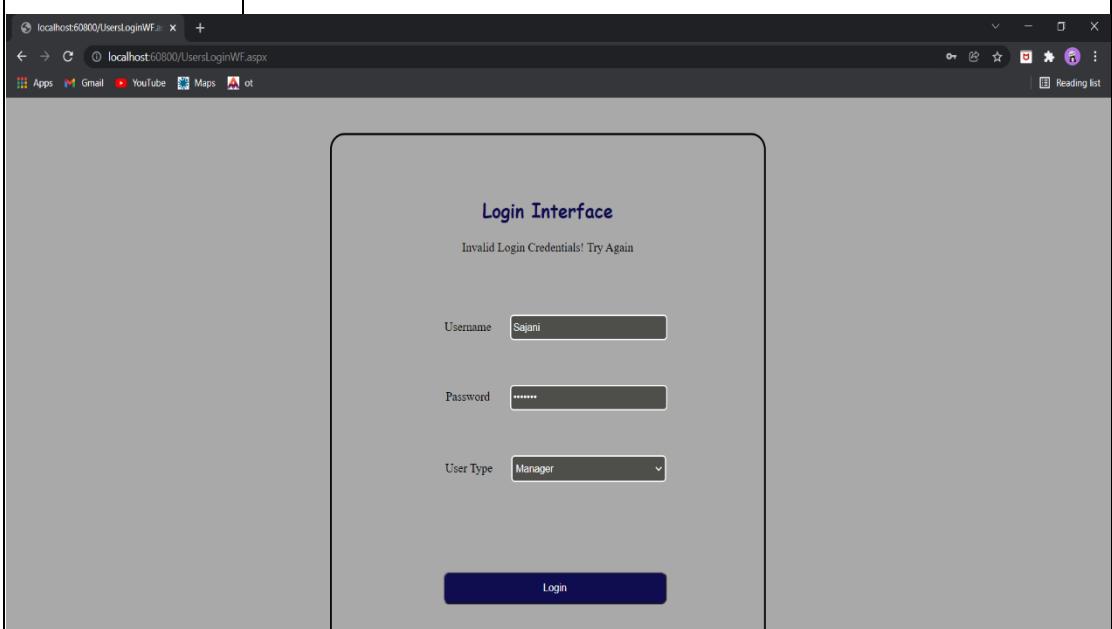
## TC6 - Pharmacy Staff Login

TEST CASE ID	TC6.1
TEST CASE NAME	Check with empty username field
DESCRIPTION	Have to enter only a password and press Login as button
TEST STEPS	1.Fill only the password field when the interface is loaded 2.Press on login button
TEST DATA	Password: Saja555
EXPECTED RESULT	Should display an Error Message on username field as username can't be blank
ACTUAL RESULT	 <p>The screenshot shows a browser window titled "localhost:60800/UsersLoginWF.aspx". Inside, a modal dialog box is titled "Login Interface". It contains three input fields: "Username" (empty), "Password" (containing dots), and "User Type" (set to "Pharmacy Staff"). A red error message "Username Cannot Be Blank" is displayed above the empty "Username" field. A "Login" button is at the bottom of the dialog.</p>
CONCLUSION	The expected an error message displayed on the username text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

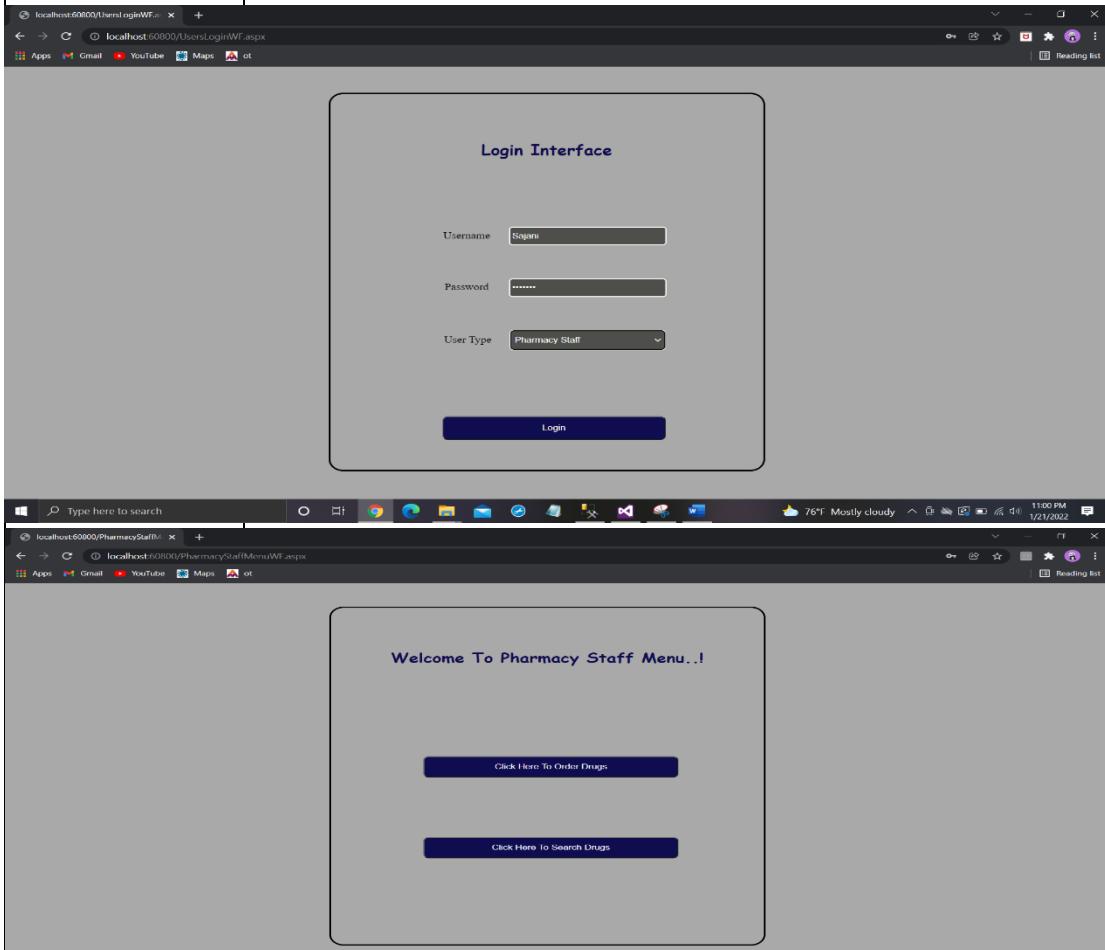
TEST CASE ID	TC6.2
TEST CASE NAME	Have to enter only a username and press Login as button
DESCRIPTION	Have to enter only a username and press Login as button
TEST STEPS	1.Fill only the username field when the interface is loaded 2.Press on login button
TEST DATA	Username: Sajani
EXPECTED RESULT	Should display an Error Message on password field as password can't be blank
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed on the password text box and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC6.3
TEST CASE NAME	Check with incorrect Pharmacy Staff username
DESCRIPTION	Have to enter incorrect username with correct Pharmacy Staff password and Pharmacy Staff user type
TEST STEPS	<ol style="list-style-type: none"> <li>Fill username field with wrong username</li> <li>Fill password field with a correct password</li> <li>Select user type as Pharmacy Staff</li> <li>Press on Login button</li> </ol>
TEST DATA	Username: Sajaniii Password: Saja555
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

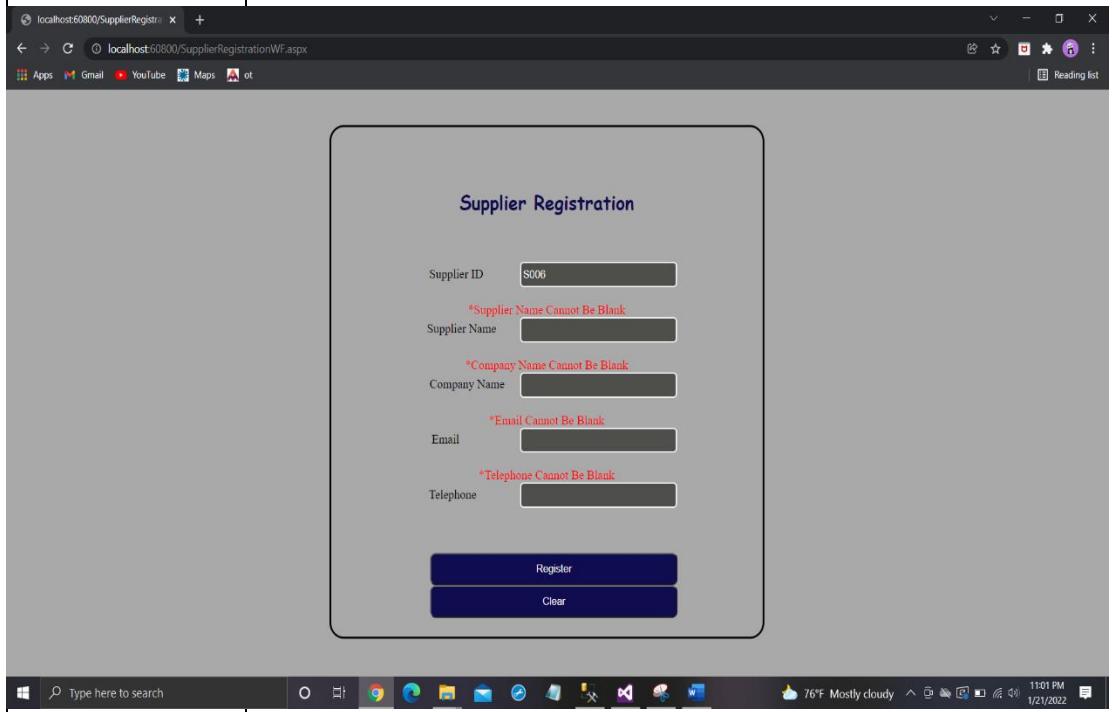
TEST CASE ID	TC6.4
TEST CASE NAME	Check with incorrect Pharmacy Staff password
DESCRIPTION	Have to enter incorrect password with correct Pharmacy Staff username and Pharmacy Staff user type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill password field with wrong password</li> <li>2.Fill username field with a correct username</li> <li>3.Select user type as Pharmacy Staff</li> <li>4.Press on Login button</li> </ol>
TEST DATA	password: 12
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	
CONCLUSION	The expected error message was displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

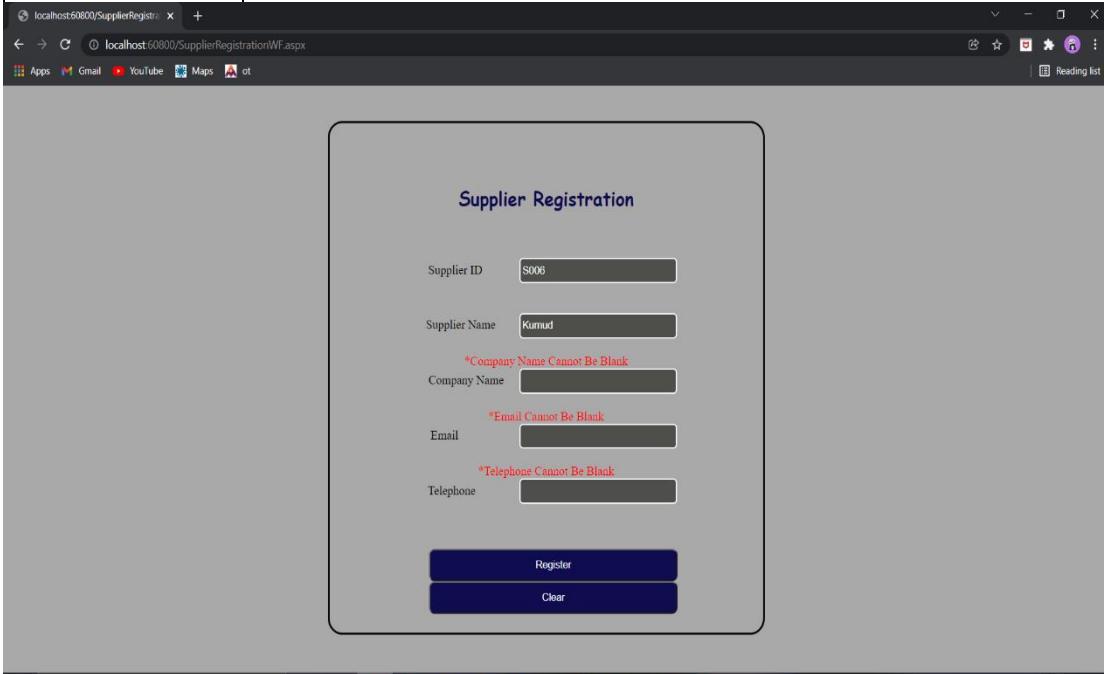
TEST CASE ID	TC6.5
TEST CASE NAME	Check with incorrect user type
DESCRIPTION	Have to enter correct Pharmacy Staff username and password without Pharmacy Staff User Type
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill username field with correct username</li> <li>2.Fill password field with a correct password</li> <li>3.Select user type as Manager</li> <li>4.Press on Login button</li> </ol>
TEST DATA	Username: Sajani Password: Saja555 User Type: Manager
EXPECTED RESULT	Should display an Error Message as Invalid login credentials! Try again
ACTUAL RESULT	
CONCLUSION	The expected an error message displayed and outcomes are very clear

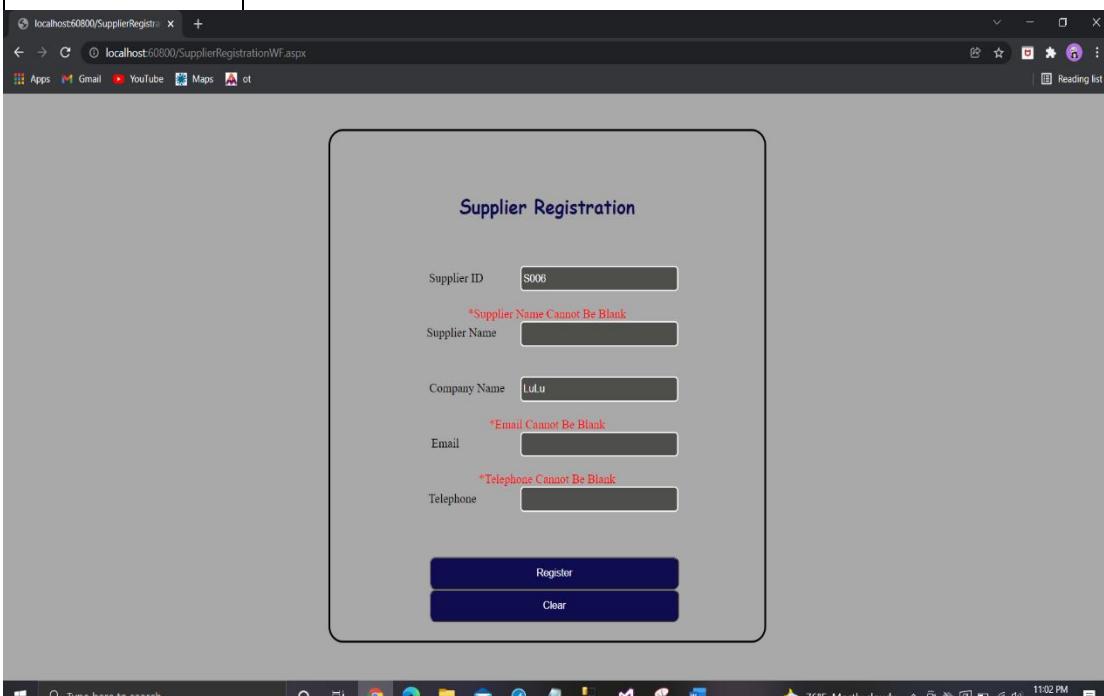
<b>STATUS (PASS/FAIL)</b>	Pass
<b>TEST CASE ID</b>	TC4.6
<b>TEST CASE NAME</b>	Check Login working without any empty field and with correct login credentials
<b>DESCRIPTION</b>	Have to enter correct Manager Pharmacy Staff, password and Pharmacy Staff User Type
<b>TEST STEPS</b>	<ol style="list-style-type: none"> <li>1.Fill username field with correct username</li> <li>2.Fill password field with a correct password</li> <li>3.Select user type as Pharmacy Staff</li> <li>4.Press on Login button</li> </ol>
<b>TEST DATA</b>	Username: Sajani Password: Saja555 User Type: Pharmacy Staff
<b>EXPECTED RESULT</b>	Should display the Pharmacy Menu
<b>ACTUAL RESULT</b>	

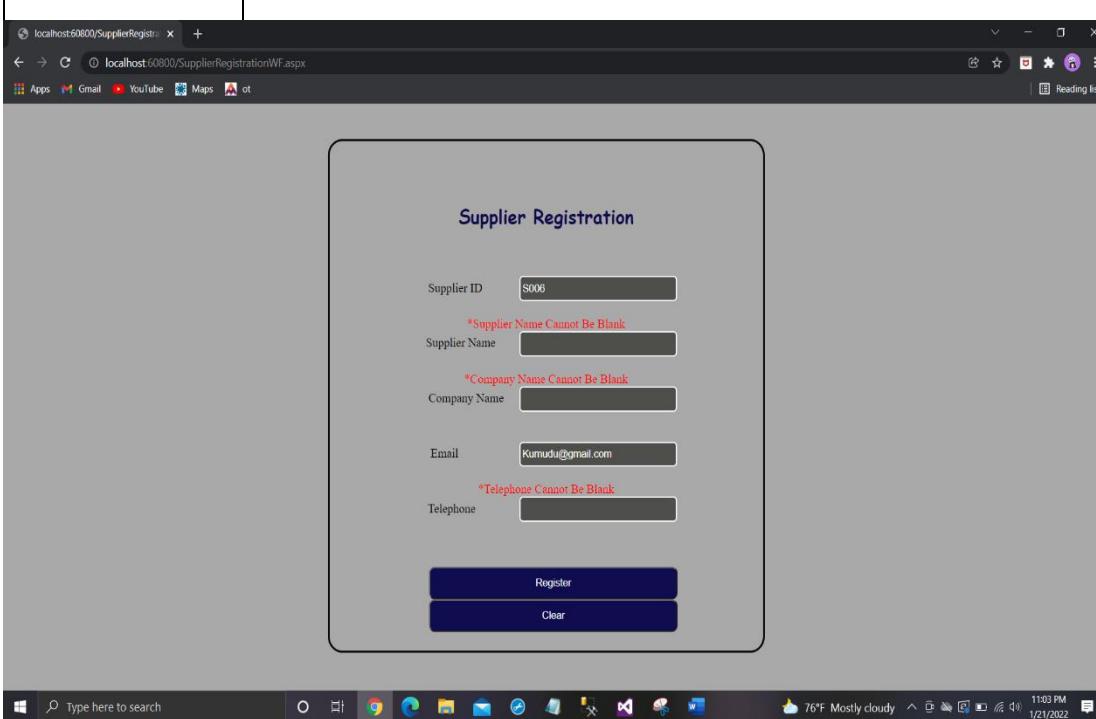


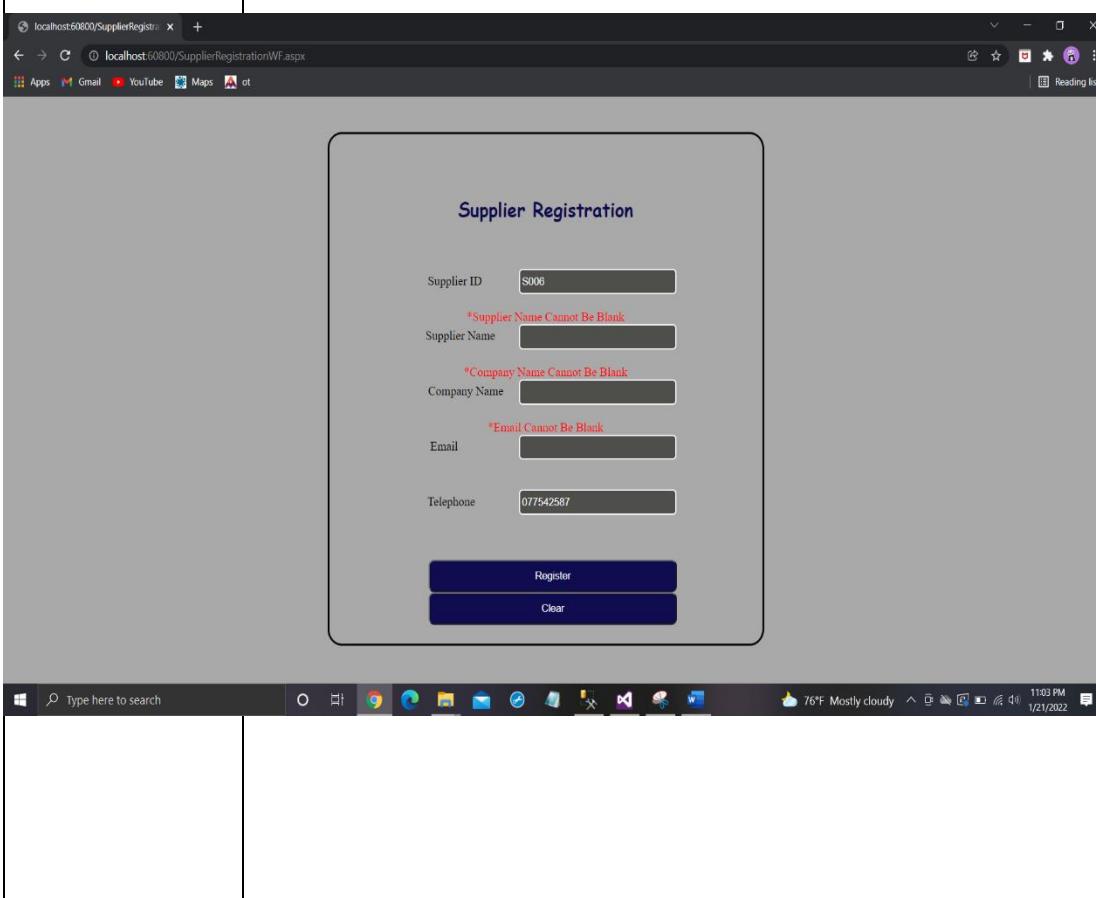
## TC7 - Supplier Registration

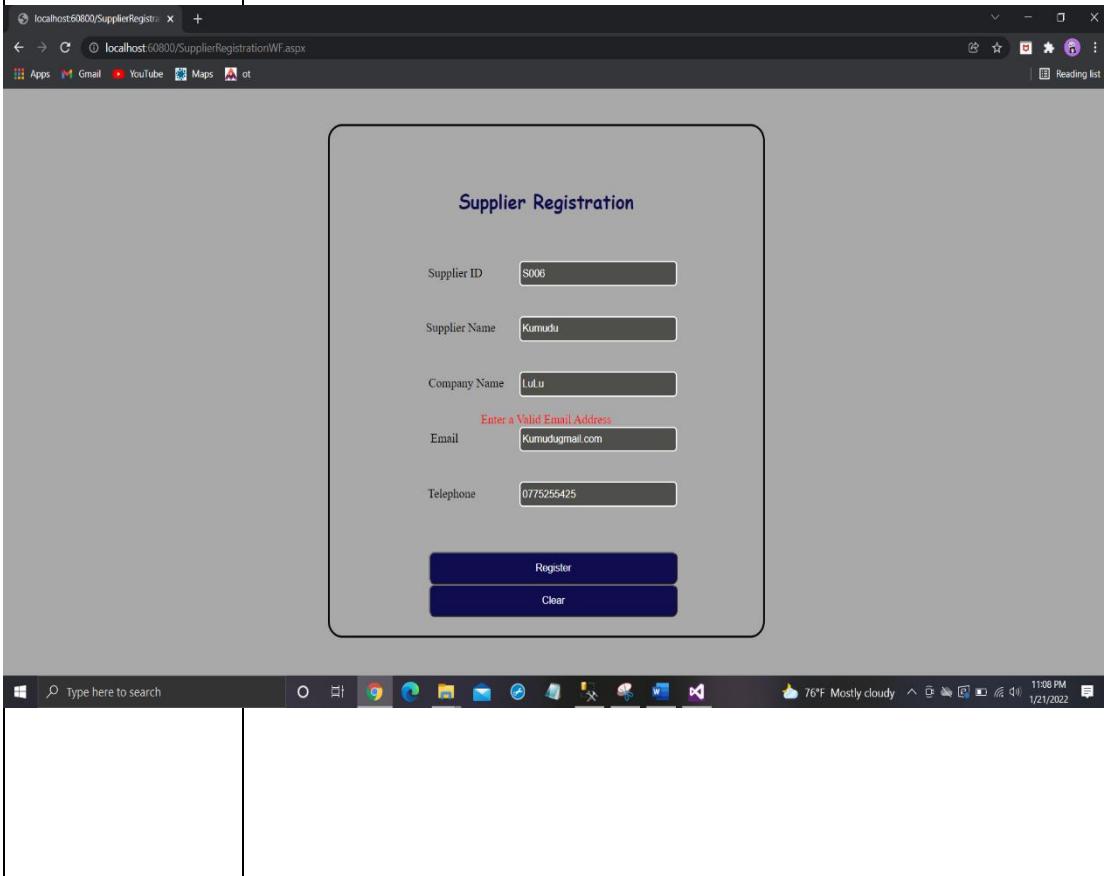
TEST CASE ID	TC7.1
TEST CASE NAME	Check Register empty fields
DESCRIPTION	Have to press on register button without filling any fields
TEST STEPS	Just press on register button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

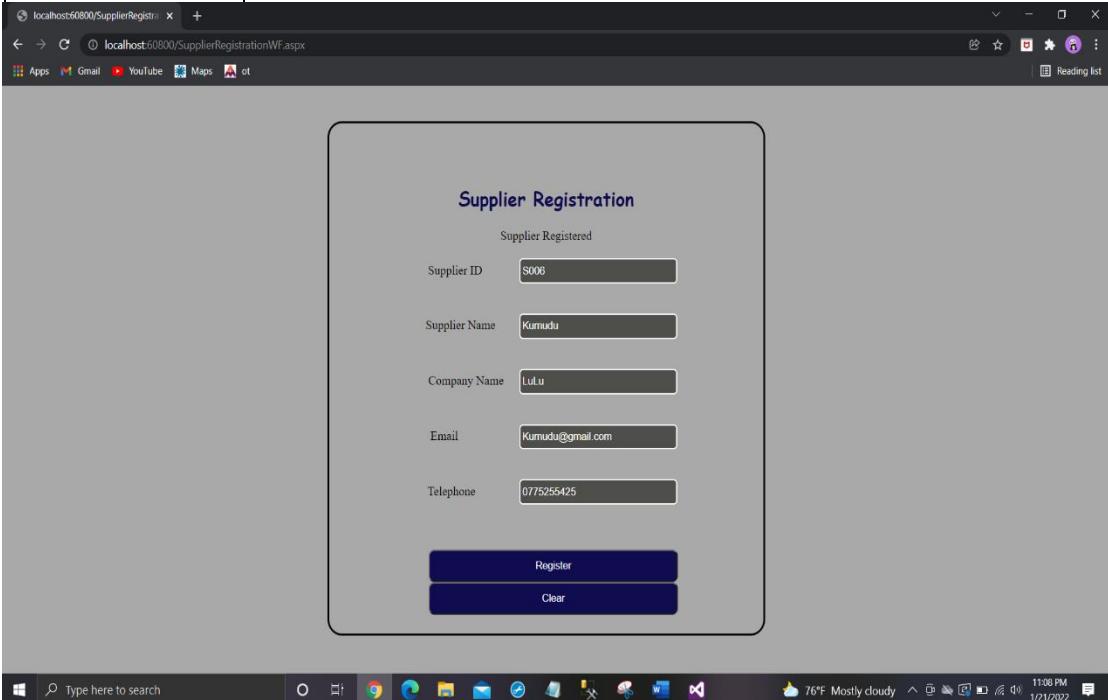
TEST CASE ID	TC7.2
TEST CASE NAME	Check other empty fields by filling only supplier name field
DESCRIPTION	Have to press on register button by filling only supplier name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only supplier Name field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC7.3
TEST CASE NAME	Check other empty fields by filling only company name field
DESCRIPTION	Have to press on register button by filling only company name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only company name field</li> <li>2.Press on register button</li> </ol>
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

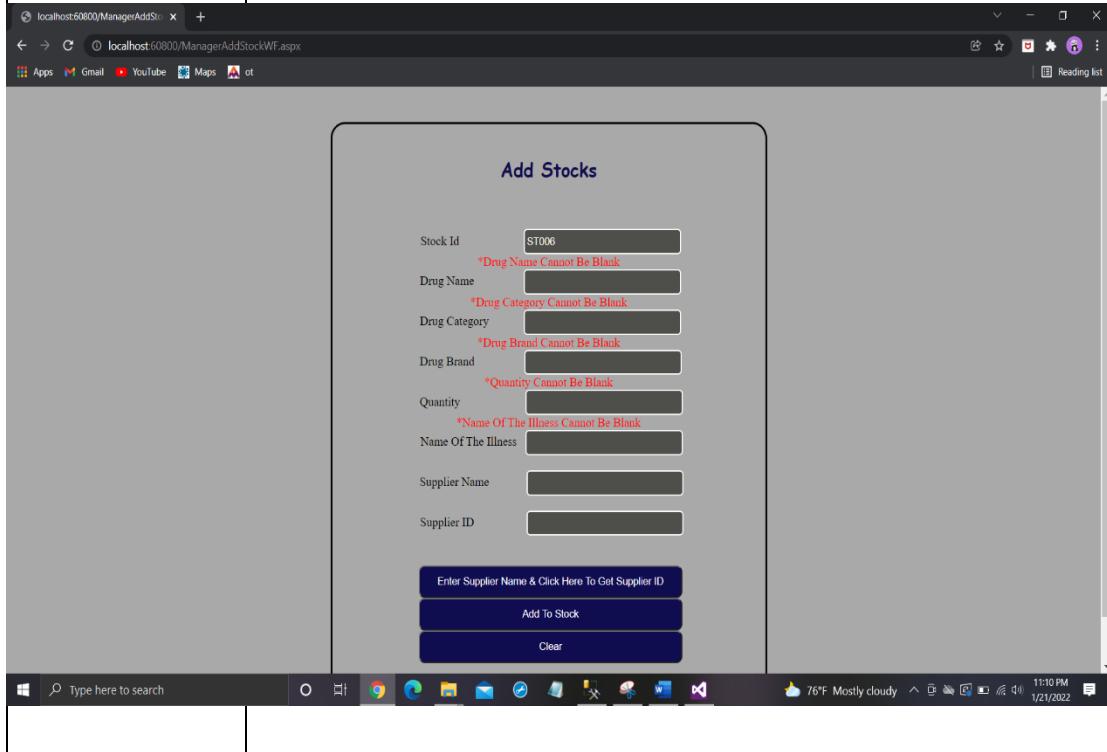
TEST CASE ID	TC7.4
TEST CASE NAME	Check other empty fields by filling only email field
DESCRIPTION	Have to press on register button by filling only email field
TEST STEPS	1.Fill only email field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

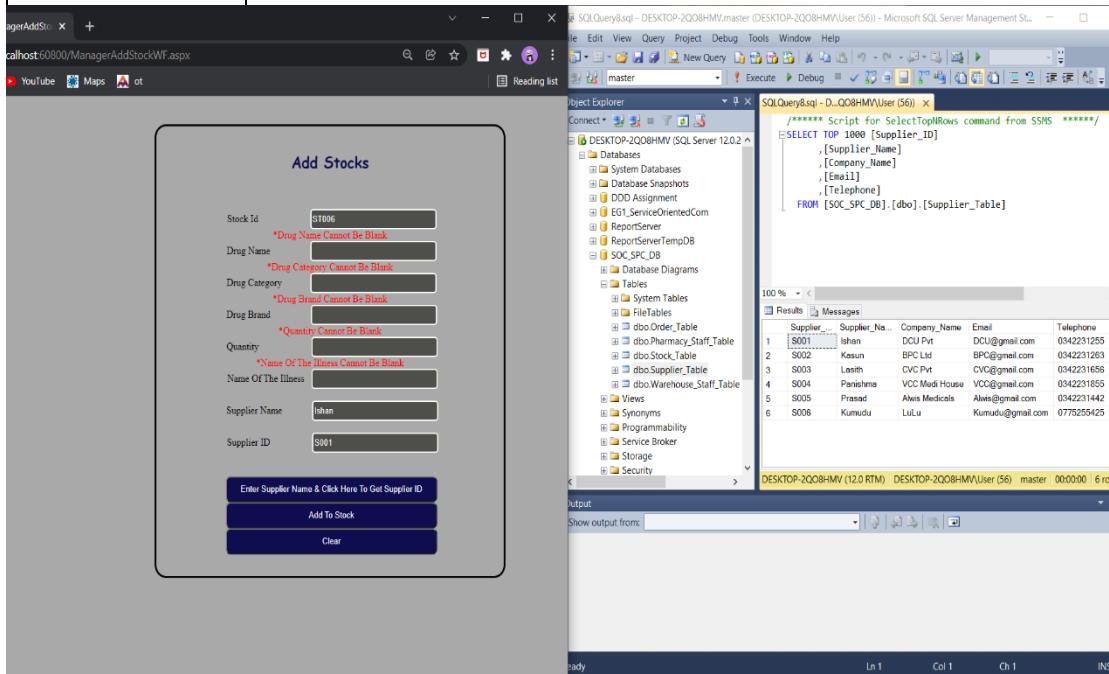
TEST CASE ID	TC7.5
TEST CASE NAME	Check other empty fields by filling only telephone field
DESCRIPTION	Have to press on register button by filling only telephone field
TEST STEPS	1.Fill only telephone field 2.Press on register button
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on all other fields with field name
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

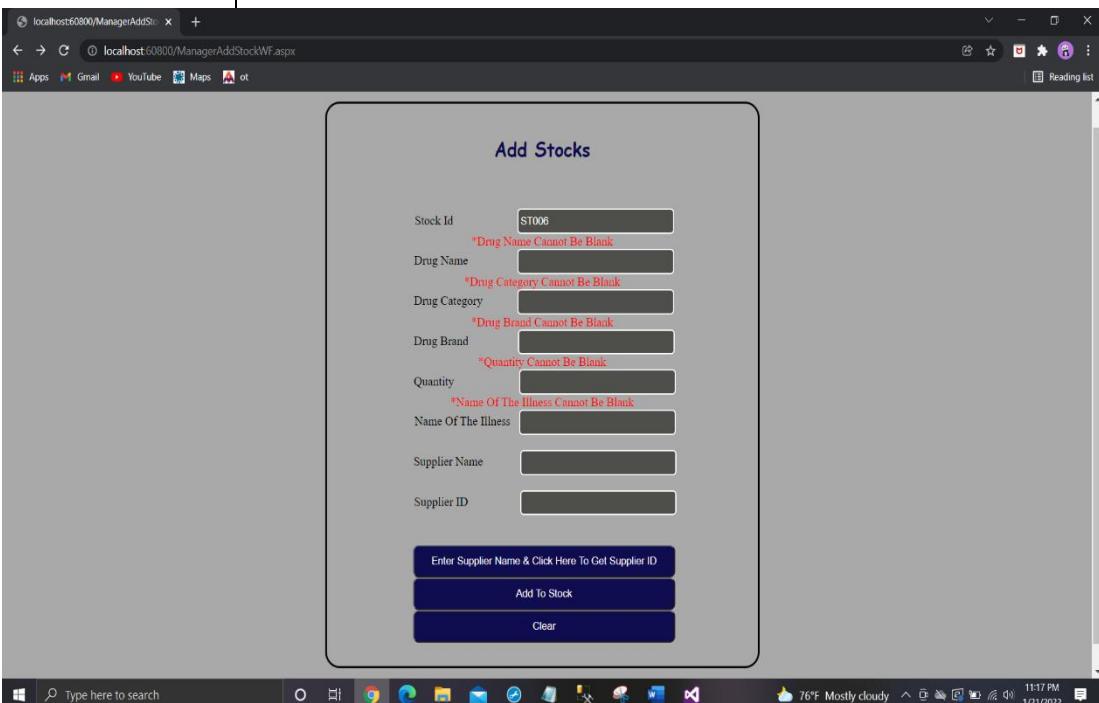
TEST CASE ID	TC7.6
TEST CASE NAME	Check email format
DESCRIPTION	Have to enter an email with an incorrect email format
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only email field (without @ symbol)</li> <li>2.Press on register button</li> </ol>
TEST DATA	Email: Kumudugmail.com
EXPECTED RESULT	Should display an Error Message on email field as Invalid Email
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

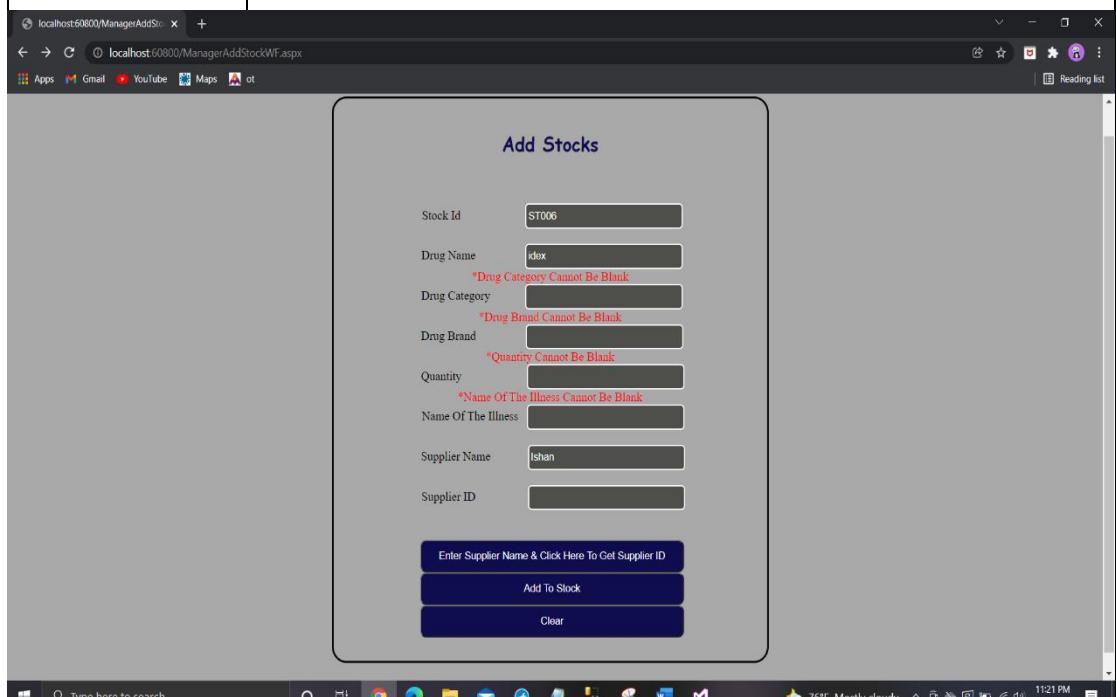
TEST CASE ID	TC7.7
TEST CASE NAME	Check Registration working without any empty field and with proper data
DESCRIPTION	Have to fill all fields and press register button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all fields</li> <li>2.Press on register button</li> </ol>
TEST DATA	<p>Supplier Name: Kumudu          Company Name: LuLu          Email: Kumudu@gmail.com          Telephone: 0775255425</p>
EXPECTED RESULT	Should display a message as Supplier Registration Successful
ACTUAL RESULT	
CONCLUSION	The expected successful message was displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

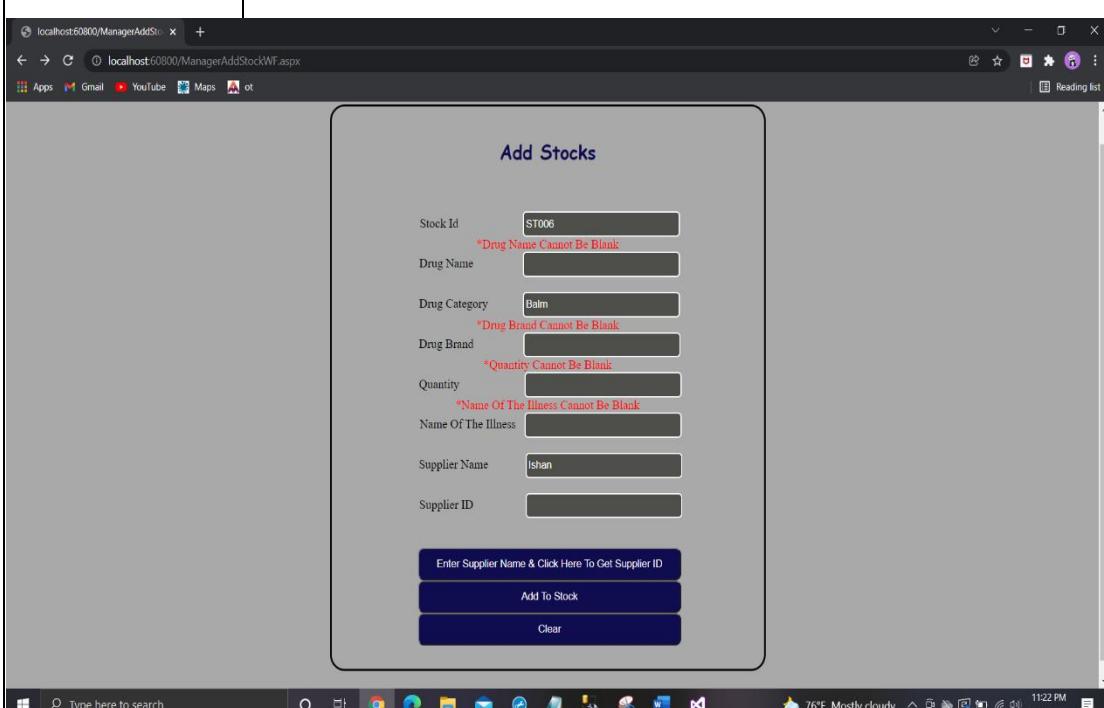
## TC8 - Add Stock

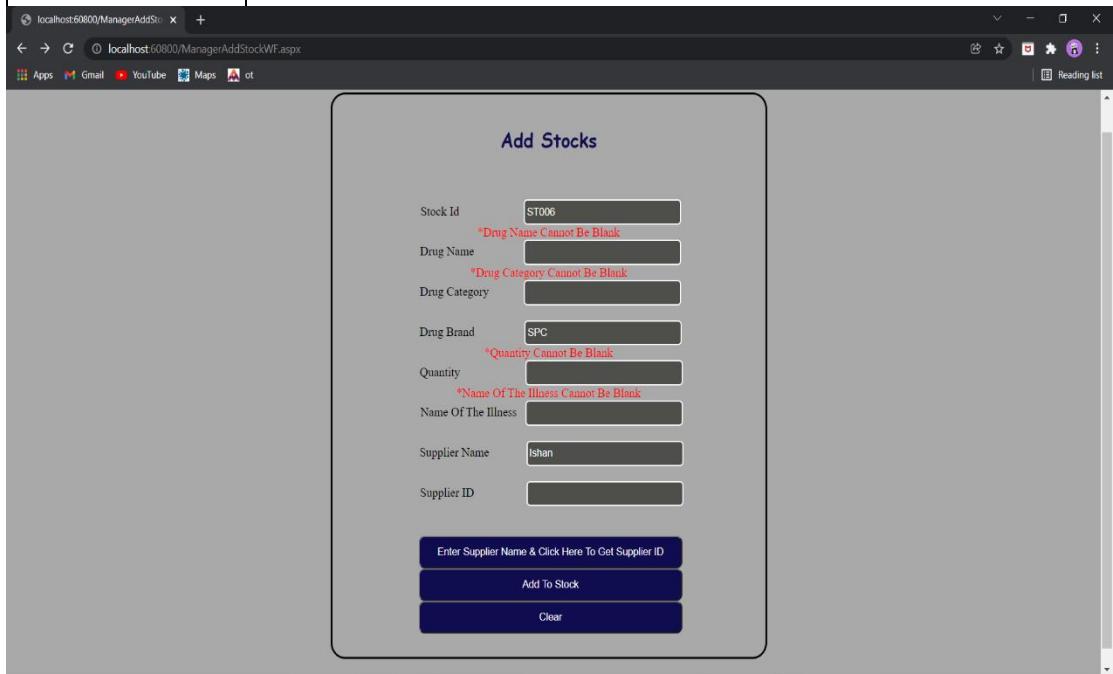
TEST CASE ID	TC8.1
TEST CASE NAME	Check gets supplier id button empty fields
DESCRIPTION	Have to press on Get Supplier ID button when the interface loaded
TEST STEPS	Press on Get Supplier ID button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

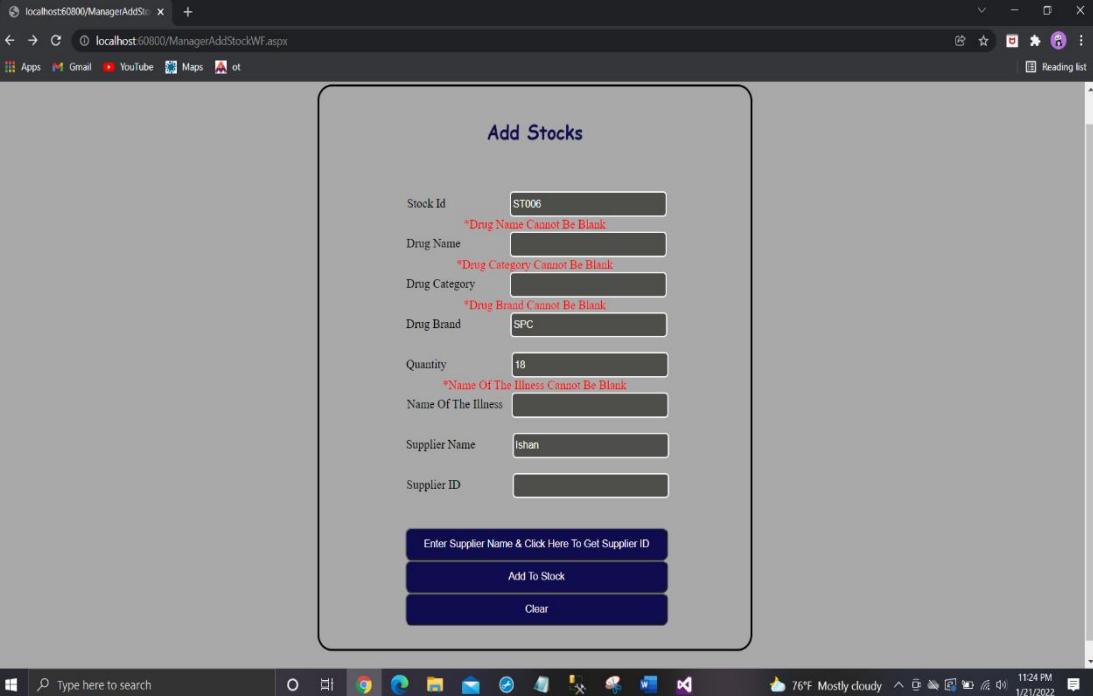
TEST CASE ID	TC8.2
TEST CASE NAME	Check working of get supplier button
DESCRIPTION	Enter a supplier name and enter Get Supplier Id button
TEST STEPS	Press on Get Supplier ID button when the interface is loaded
TEST DATA	Get Supplier Id button correctness
EXPECTED RESULT	Should supplier id get filled automatically
ACTUAL RESULT	
CONCLUSION	As expected, the supplier id appeared in the supplier's name field after entering the button Get Supplier ID and outcomes are very clear
STATUS (PASS/FAIL)	Pass

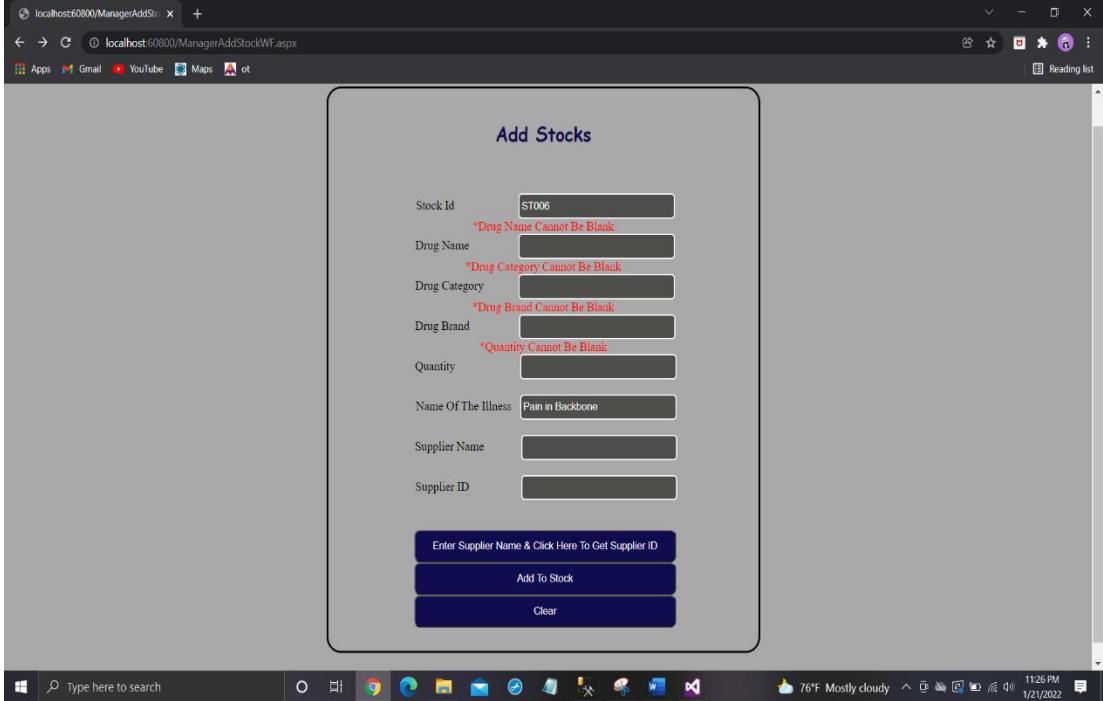
TEST CASE ID	TC8.3
TEST CASE NAME	Check add stock empty fields
DESCRIPTION	Have to press on add stock button without filling any fields
TEST STEPS	Press on add stock button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

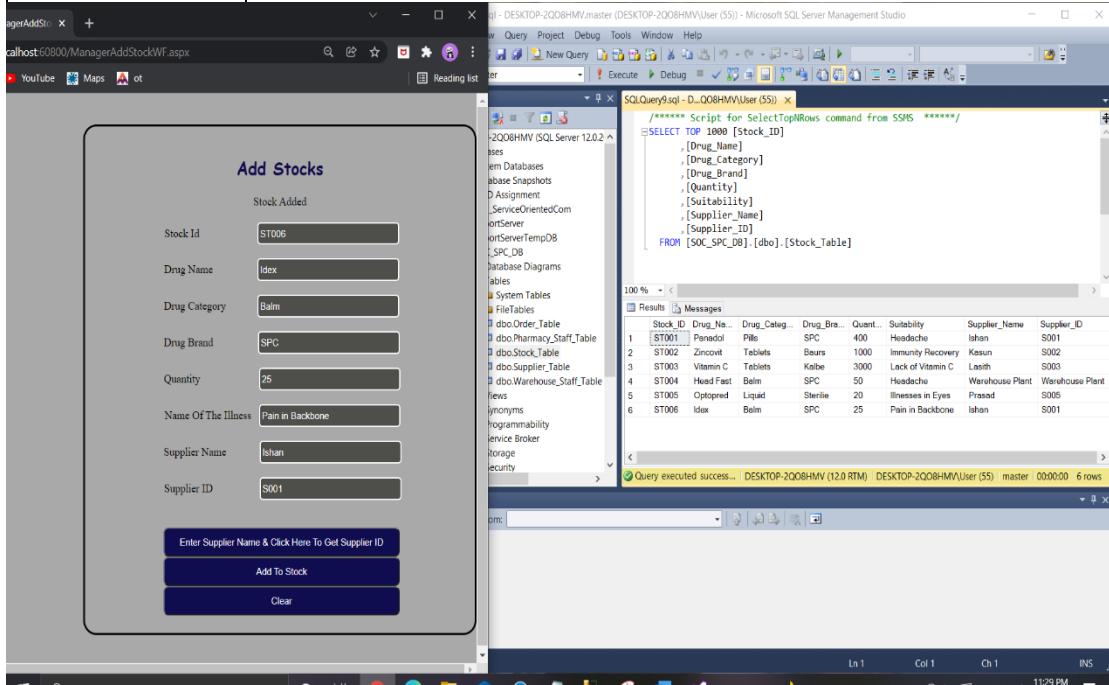
TEST CASE ID	TC8.4
TEST CASE NAME	Check other empty fields by filling only drug name field
DESCRIPTION	Have to press on add stock button by filling only drug name field
TEST STEPS	1.Fill only Drug Name field 2.Press on Add Stocks button
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC8.5
TEST CASE NAME	Check other empty fields by filling only drug category field
DESCRIPTION	Have to press on add stock button by filling only drug category field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only Drug category field</li> <li>2.Press on Add Stocks button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	 <p>The screenshot shows a Windows desktop environment with a browser window open to 'localhost:60800/ManagerAddStockWF.aspx'. The page title is 'Add Stocks'. The form has several text input fields with associated error messages in red:</p> <ul style="list-style-type: none"> <li>Stock Id: ST006 (no message)</li> <li>Drug Name: *Drug Name Cannot Be Blank (highlighted in red)</li> <li>Drug Category: Balm (no message)</li> <li>Drug Brand: *Drug Brand Cannot Be Blank (highlighted in red)</li> <li>Quantity: *Quantity Cannot Be Blank (highlighted in red)</li> <li>Name Of The Illness: *Name Of The Illness Cannot Be Blank (highlighted in red)</li> <li>Supplier Name: Ishan (no message)</li> <li>Supplier ID: (no message)</li> </ul> <p>Below the form are three buttons: 'Enter Supplier Name &amp; Click Here To Get Supplier ID' (disabled), 'Add To Stock' (disabled), and 'Clear'.</p>
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

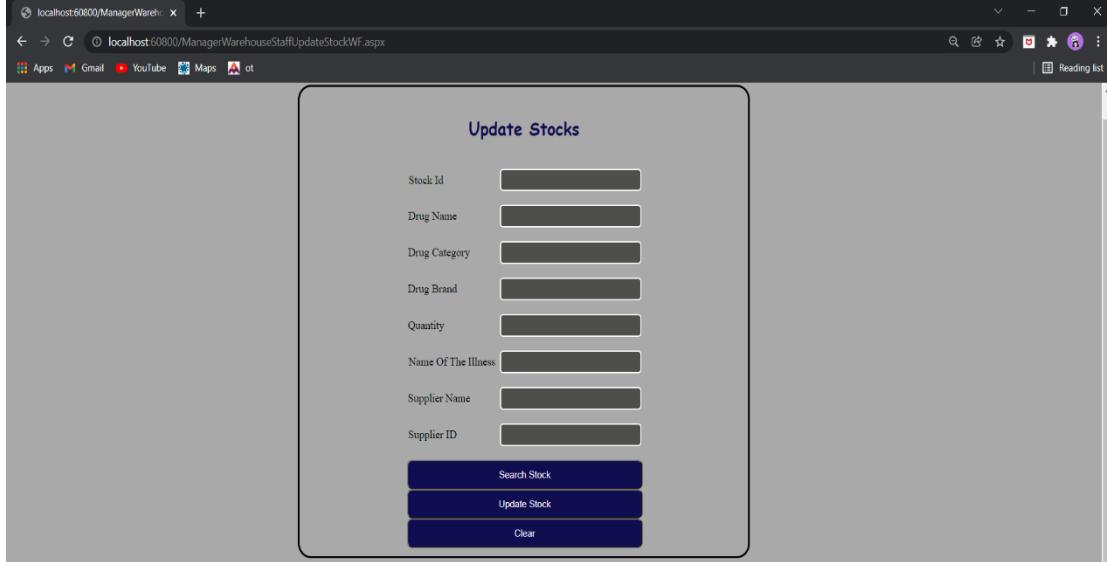
TEST CASE ID	TC8.6
TEST CASE NAME	Check other empty fields by filling only drug brand field
DESCRIPTION	Have to press on add stock button by filling only drug brand field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only Drug brand field</li> <li>2.Press on Add Stocks button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

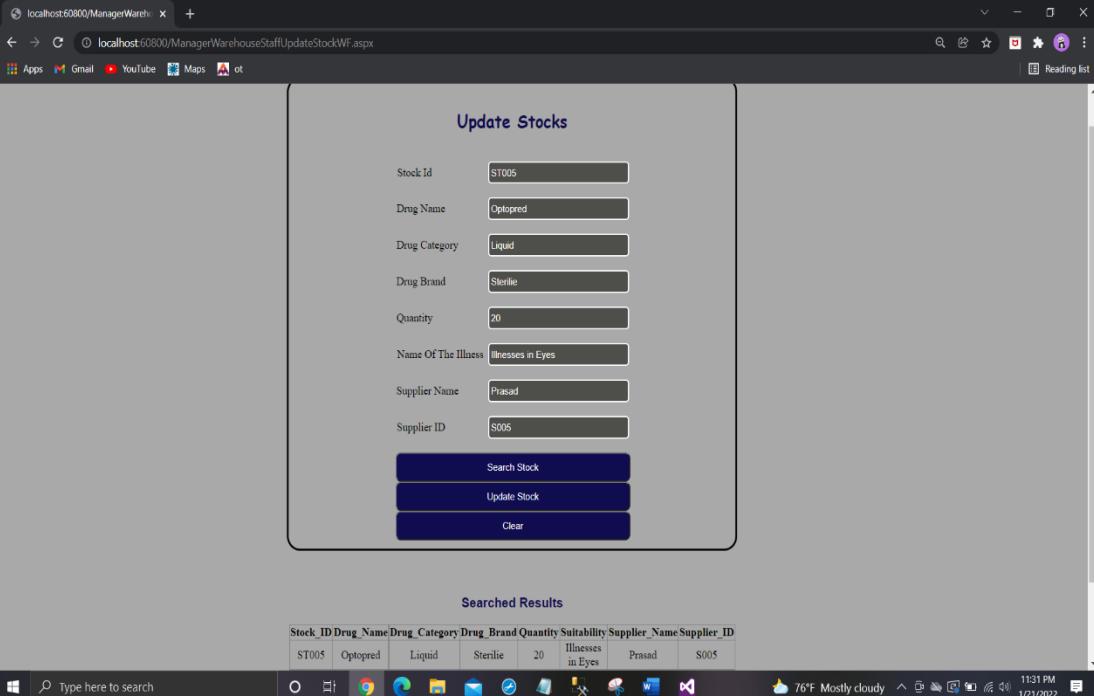
TEST CASE ID	TC8.7
TEST CASE NAME	Check other empty fields by filling only quantity field
DESCRIPTION	Have to press on add stock button by filling only quantity field
TEST STEPS	1.Fill only quantity field 2.Press on Add Stocks button
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

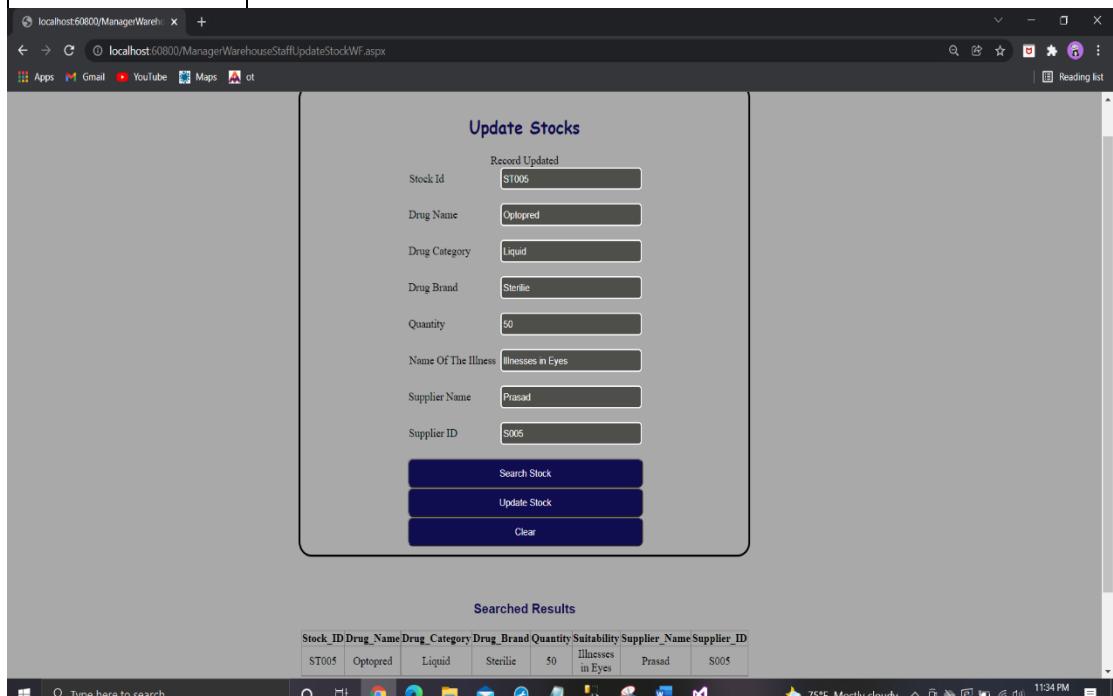
TEST CASE ID	TC8.9
TEST CASE NAME	Check other empty fields by filling only name of the illness field
DESCRIPTION	Have to press on add stock button by filling only name of the illness field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only name of the illness field</li> <li>2.Press on Add Stocks button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Stock id, supplier name and supplier id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC8.10																																																							
TEST CASE NAME	Check Add Stock working without any empty fields																																																							
DESCRIPTION	Have to fill all fields and press add stock button																																																							
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all the fields</li> <li>2.Press on Add Stocks button</li> </ol>																																																							
TEST DATA	<p>Drug name: Idex          Drug Category: Balm          Drug Brand: SPC          Quantity: 25          Name of the illness: Pain in Backbone          Supplier name: Ishan          Supplier id: S001</p>																																																							
EXPECTED RESULT	Should display a message as Stock Added Successfully																																																							
ACTUAL RESULT																																																								
 <p>The screenshot displays two windows side-by-side. On the left is a web browser window titled 'ManagerAddStockWF.aspx' showing the 'Add Stocks' form. The form has fields for Stock Id (S006), Drug Name (Idex), Drug Category (Balm), Drug Brand (SPC), Quantity (25), Name Of The Illness (Pain in Backbone), Supplier Name (Ishan), and Supplier ID (S001). Below the form are buttons for 'Enter Supplier Name &amp; Click Here To Get Supplier ID', 'Add To Stock', and 'Clear'. On the right is Microsoft SQL Server Management Studio (SSMS) showing a query window with the following SQL script and results:</p> <pre> SELECT TOP 1000 [Stock_ID]       ,[Drug_Name]       ,[Drug_Category]       ,[Drug_Brand]       ,[Quantity]       ,[Suitability]       ,[Supplier_Name]       ,[Supplier_ID]    FROM [SOC_SPC_DB].[dbo].[Stock_Table]   </pre> <p>The results grid shows six rows of data from the Stock_Table:</p> <table border="1"> <thead> <tr> <th>Stock_ID</th> <th>Drug_Nam...</th> <th>Drug_Categ...</th> <th>Drug_Bra...</th> <th>Quant...</th> <th>Suitability</th> <th>Supplier_Nam...</th> <th>Supplier_ID</th> </tr> </thead> <tbody> <tr> <td>S001</td> <td>Paracet</td> <td>Pills</td> <td>SPC</td> <td>400</td> <td>Headache</td> <td>Ishan</td> <td>S001</td> </tr> <tr> <td>S002</td> <td>Zincov</td> <td>Tablets</td> <td>Beurs</td> <td>1000</td> <td>Immunity Recovery</td> <td>Kesun</td> <td>S002</td> </tr> <tr> <td>S003</td> <td>Vitamin C</td> <td>Tablets</td> <td>Kelbe</td> <td>3000</td> <td>Lack of Vitamin C</td> <td>Lashif</td> <td>S003</td> </tr> <tr> <td>S004</td> <td>Head Fast</td> <td>Balm</td> <td>SPC</td> <td>50</td> <td>Headache</td> <td>Warehouse Plant</td> <td>Warehouse Plant</td> </tr> <tr> <td>S005</td> <td>Optopred</td> <td>Liquid</td> <td>Sterile</td> <td>20</td> <td>Illnesses in Eyes</td> <td>Prasad</td> <td>S005</td> </tr> <tr> <td>S006</td> <td>Idex</td> <td>Balm</td> <td>SPC</td> <td>25</td> <td>Pain in Backbone</td> <td>Ishan</td> <td>S001</td> </tr> </tbody> </table> <p>Below the results grid, the status bar indicates: Query executed successfully. DESKTOP-2Q08HMV (12.0 RTM) DESKTOP-2Q08HMV\User (55) master 00:00:00 6 rows.</p>	Stock_ID	Drug_Nam...	Drug_Categ...	Drug_Bra...	Quant...	Suitability	Supplier_Nam...	Supplier_ID	S001	Paracet	Pills	SPC	400	Headache	Ishan	S001	S002	Zincov	Tablets	Beurs	1000	Immunity Recovery	Kesun	S002	S003	Vitamin C	Tablets	Kelbe	3000	Lack of Vitamin C	Lashif	S003	S004	Head Fast	Balm	SPC	50	Headache	Warehouse Plant	Warehouse Plant	S005	Optopred	Liquid	Sterile	20	Illnesses in Eyes	Prasad	S005	S006	Idex	Balm	SPC	25	Pain in Backbone	Ishan	S001
Stock_ID	Drug_Nam...	Drug_Categ...	Drug_Bra...	Quant...	Suitability	Supplier_Nam...	Supplier_ID																																																	
S001	Paracet	Pills	SPC	400	Headache	Ishan	S001																																																	
S002	Zincov	Tablets	Beurs	1000	Immunity Recovery	Kesun	S002																																																	
S003	Vitamin C	Tablets	Kelbe	3000	Lack of Vitamin C	Lashif	S003																																																	
S004	Head Fast	Balm	SPC	50	Headache	Warehouse Plant	Warehouse Plant																																																	
S005	Optopred	Liquid	Sterile	20	Illnesses in Eyes	Prasad	S005																																																	
S006	Idex	Balm	SPC	25	Pain in Backbone	Ishan	S001																																																	
CONCLUSION	The expected successful message was displayed and outcomes are very clear																																																							
STATUS (PASS/FAIL)	Pass																																																							

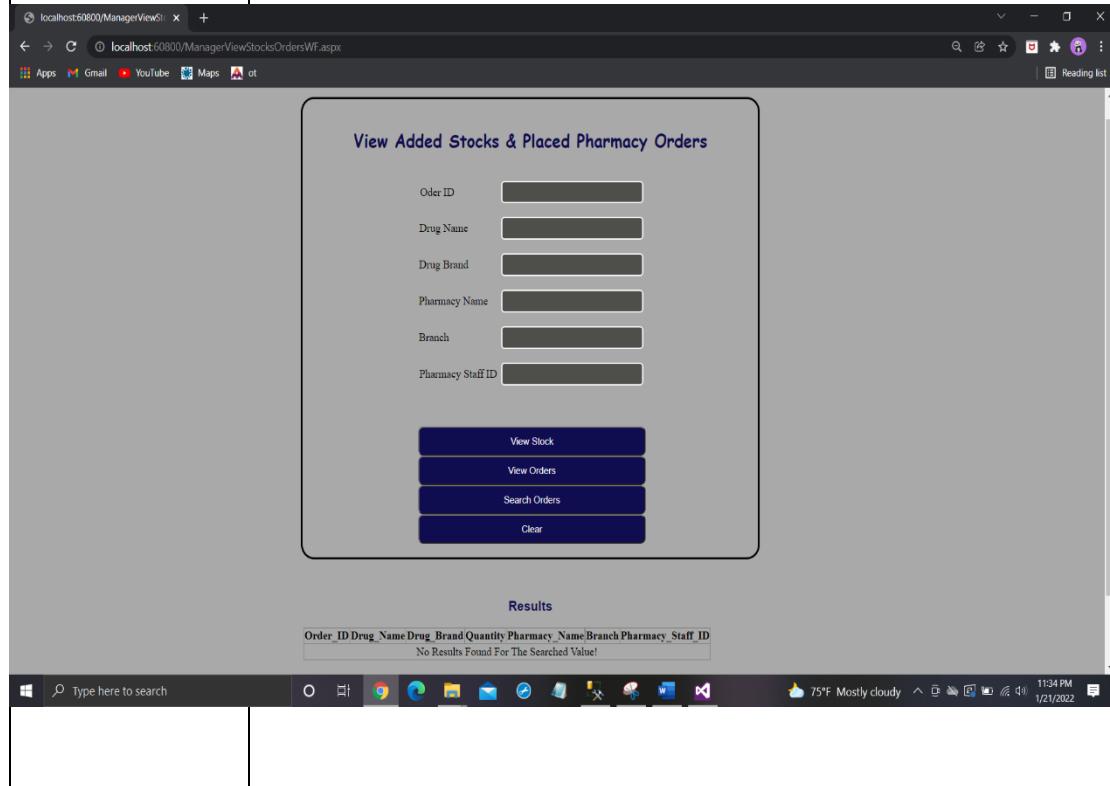
## TC9 - Update Stock

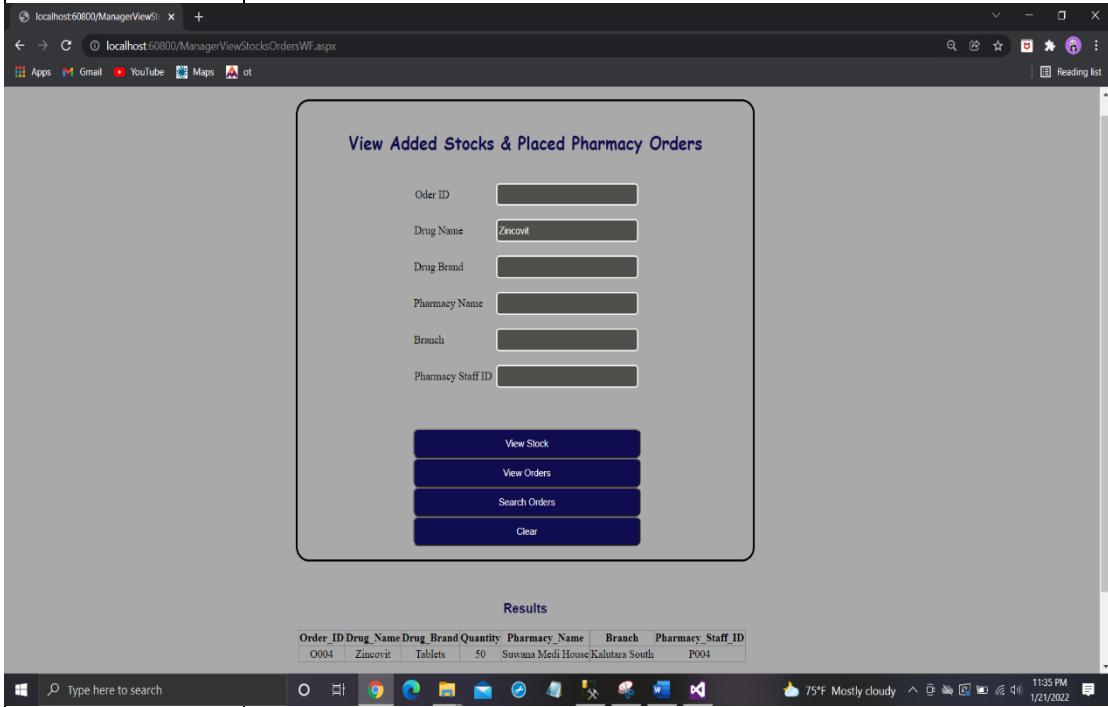
TEST CASE ID	TC9.1
TEST CASE NAME	Check search button with empty fields in update stock interface
DESCRIPTION	Press search stock button without filling any field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all the fields</li> <li>2.Press on Add Stocks button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	A table should appear and show an Error Message as No Results Found
ACTUAL RESULT	
CONCLUSION	The expected table appeared the error message was also displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC9.2																
TEST CASE NAME	Check search button in update stock interface																
DESCRIPTION	Have to enter any field expect quantity field and press on search stock button																
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill any field name except quantity</li> <li>2.Press on search stock button</li> </ol>																
TEST DATA	Drug Category: Liquid																
EXPECTED RESULT	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row																
ACTUAL RESULT	 <p>Searched Results</p> <table border="1"> <thead> <tr> <th>Stock_ID</th> <th>Drug_Name</th> <th>Drug_Category</th> <th>Drug_Brand</th> <th>Quantity</th> <th>Suitability</th> <th>Supplier_Name</th> <th>Supplier_ID</th> </tr> </thead> <tbody> <tr> <td>ST005</td> <td>Optopred</td> <td>Liquid</td> <td>Sterile</td> <td>20</td> <td>Illnesses in Eyes</td> <td>Prasad</td> <td>S005</td> </tr> </tbody> </table>	Stock_ID	Drug_Name	Drug_Category	Drug_Brand	Quantity	Suitability	Supplier_Name	Supplier_ID	ST005	Optopred	Liquid	Sterile	20	Illnesses in Eyes	Prasad	S005
Stock_ID	Drug_Name	Drug_Category	Drug_Brand	Quantity	Suitability	Supplier_Name	Supplier_ID										
ST005	Optopred	Liquid	Sterile	20	Illnesses in Eyes	Prasad	S005										
CONCLUSION	The expected table appeared the record was also displayed and outcomes are very clear																
STATUS (PASS/FAIL)	Pass																

TEST CASE ID	TC9.3
TEST CASE NAME	Check update button
DESCRIPTION	Have to search by entering any field expect quantity field and then update necessary fields and press on update button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Change the needed field name(s)</li> <li>2.Press on update stock button</li> </ol>
TEST DATA	Quantity: 50
EXPECTED RESULT	A message will be appeared as Stock Updated
ACTUAL RESULT	
CONCLUSION	Expected successful message appeared and outcomes are very clear
STATUS (PASS/FAIL)	Pass

## TC10 - View added stocks and placed orders

TEST CASE ID	TC10.1
TEST CASE NAME	Check search button with empty fields in view added stocks and placed orders interface
DESCRIPTION	Press search stock button without filling any field
TEST STEPS	Just press on search stock button without filling any field
TEST DATA	Empty Fields
EXPECTED RESULT	A table should appear and show an Error Message as No Results Found
ACTUAL RESULT	
CONCLUSION	The expected table appeared the error message was also displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

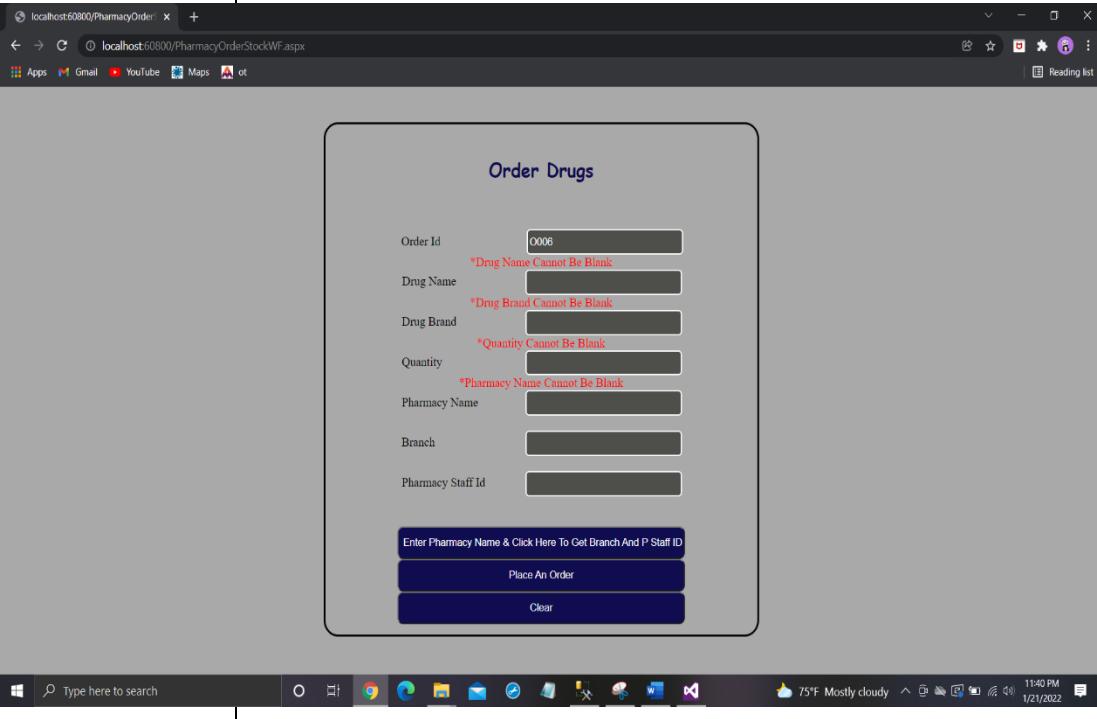
TEST CASE ID	TC10.2
TEST CASE NAME	Check search button in view added stocks and placed orders interface
DESCRIPTION	Have to enter any field and press on search stock button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill any field</li> <li>2.Press on search orders button</li> </ol>
TEST DATA	Search Orders button correctness
EXPECTED RESULT	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row
ACTUAL RESULT	
CONCLUSION	The expected table appeared the record was also displayed and outcomes are very clear
STATUS (PASS/FAIL)	Pass

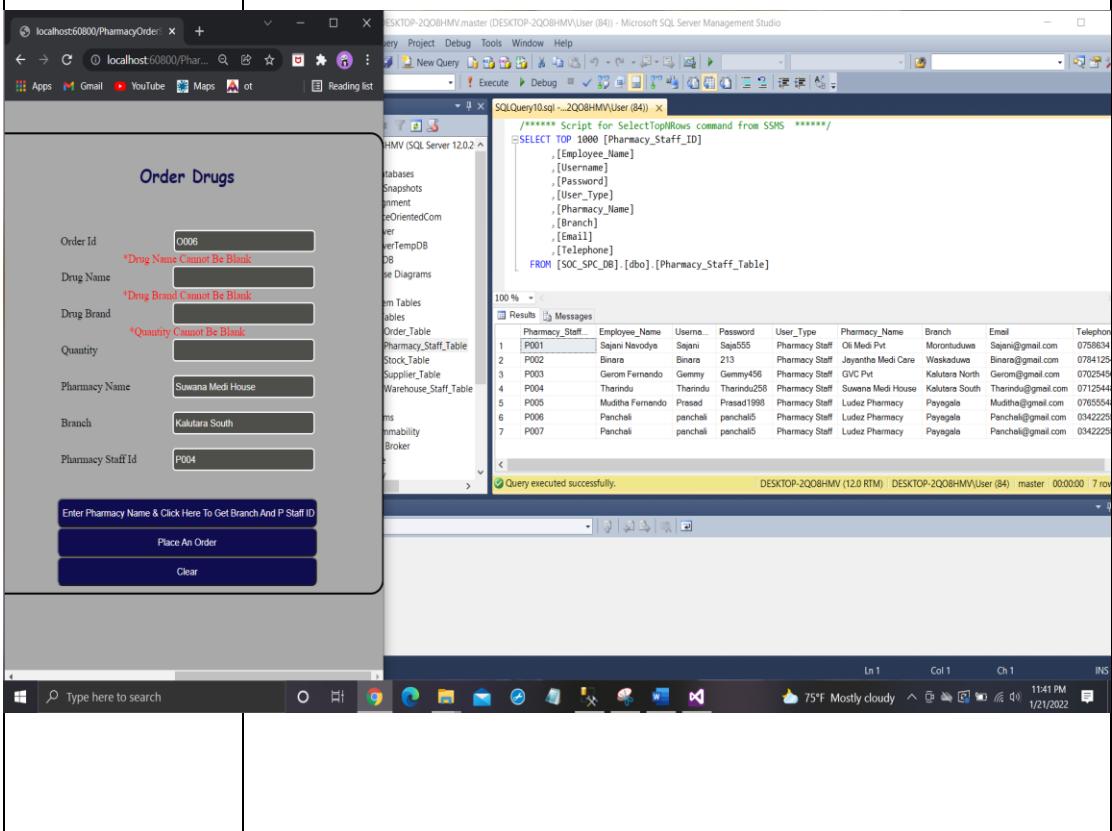
TEST CASE ID	TC10.3																																																								
TEST CASE NAME	Check View Stock button																																																								
DESCRIPTION	Press on View Stock button when the interface is loaded																																																								
TEST STEPS	Press on View Stock button																																																								
TEST DATA	View Stock button correctness																																																								
EXPECTED RESULT	A table should be appeared with the values in the particular database table row																																																								
ACTUAL RESULT	<table border="1"> <thead> <tr> <th>Stock_ID</th> <th>Drug_Name</th> <th>Drug_Category</th> <th>Drug_Brand</th> <th>Quantity</th> <th>Suitability</th> <th>Supplier_Name</th> <th>Supplier_ID</th> </tr> </thead> <tbody> <tr> <td>ST001</td> <td>Panadol</td> <td>Pills</td> <td>SPC</td> <td>400</td> <td>Headache</td> <td>Ishan</td> <td>9001</td> </tr> <tr> <td>ST002</td> <td>Zanacit</td> <td>Tablets</td> <td>Baurs</td> <td>1000</td> <td>Immunity Recovery</td> <td>Karen</td> <td>9002</td> </tr> <tr> <td>ST003</td> <td>Vitamin C</td> <td>Tablets</td> <td>Kalle</td> <td>3000</td> <td>Lack of Vitamin C</td> <td>Louth</td> <td>9003</td> </tr> <tr> <td>ST004</td> <td>Head Fast</td> <td>Bals</td> <td>SPC</td> <td>50</td> <td>Headache</td> <td>Warehouse Plant</td> <td>Warehouse Plant</td> </tr> <tr> <td>ST005</td> <td>Optopred</td> <td>Liquid</td> <td>Stenber</td> <td>50</td> <td>Illnesses in Eyes</td> <td>Prasad</td> <td>9005</td> </tr> <tr> <td>ST006</td> <td>Ibas</td> <td>Bals</td> <td>SPC</td> <td>25</td> <td>Pain in Backbone</td> <td>Ishan</td> <td>9001</td> </tr> </tbody> </table>	Stock_ID	Drug_Name	Drug_Category	Drug_Brand	Quantity	Suitability	Supplier_Name	Supplier_ID	ST001	Panadol	Pills	SPC	400	Headache	Ishan	9001	ST002	Zanacit	Tablets	Baurs	1000	Immunity Recovery	Karen	9002	ST003	Vitamin C	Tablets	Kalle	3000	Lack of Vitamin C	Louth	9003	ST004	Head Fast	Bals	SPC	50	Headache	Warehouse Plant	Warehouse Plant	ST005	Optopred	Liquid	Stenber	50	Illnesses in Eyes	Prasad	9005	ST006	Ibas	Bals	SPC	25	Pain in Backbone	Ishan	9001
Stock_ID	Drug_Name	Drug_Category	Drug_Brand	Quantity	Suitability	Supplier_Name	Supplier_ID																																																		
ST001	Panadol	Pills	SPC	400	Headache	Ishan	9001																																																		
ST002	Zanacit	Tablets	Baurs	1000	Immunity Recovery	Karen	9002																																																		
ST003	Vitamin C	Tablets	Kalle	3000	Lack of Vitamin C	Louth	9003																																																		
ST004	Head Fast	Bals	SPC	50	Headache	Warehouse Plant	Warehouse Plant																																																		
ST005	Optopred	Liquid	Stenber	50	Illnesses in Eyes	Prasad	9005																																																		
ST006	Ibas	Bals	SPC	25	Pain in Backbone	Ishan	9001																																																		
CONCLUSION	The expected table appeared with all the records and outcomes are very clear																																																								
STATUS (PASS/FAIL)	Pass																																																								

TEST CASE ID	TC10.4
TEST CASE NAME	Check View Orders button
DESCRIPTION	Press on View Orders button when the interface is loaded
TEST STEPS	Press on View Orders button
TEST DATA	View Orders button correctness
EXPECTED RESULT	A table should be appeared with the values in the particular database table row
ACTUAL RESULT	

	<p>View Added Stocks &amp; Placed Pharmacy Orders</p> <table border="1"> <thead> <tr> <th>Order_ID</th> <th>Drug_Name</th> <th>Drug_Brand</th> <th>Quantity</th> <th>Pharmacy_Name</th> <th>Branch</th> <th>Pharmacy_Staff_ID</th> </tr> </thead> <tbody> <tr> <td>O001</td> <td>Panadol</td> <td>SPC</td> <td>10</td> <td>Oil Medi Pet</td> <td>Morontaduwa</td> <td>P001</td> </tr> <tr> <td>O002</td> <td>Venomax C</td> <td>Tablets</td> <td>100</td> <td>Jayantha Medi Care</td> <td>Wankadawwa</td> <td>P002</td> </tr> <tr> <td>O003</td> <td>Panadol</td> <td>SPC</td> <td>80</td> <td>GVC Pet</td> <td>Kalutara North</td> <td>P003</td> </tr> <tr> <td>O004</td> <td>Zincovit</td> <td>Tablets</td> <td>50</td> <td>Suwana Medi House</td> <td>Kalutara South</td> <td>P004</td> </tr> <tr> <td>O005</td> <td>Optoperif</td> <td>Liquid</td> <td>15</td> <td>Ludez Pharmacy</td> <td>Pavagala</td> <td>P005</td> </tr> </tbody> </table>	Order_ID	Drug_Name	Drug_Brand	Quantity	Pharmacy_Name	Branch	Pharmacy_Staff_ID	O001	Panadol	SPC	10	Oil Medi Pet	Morontaduwa	P001	O002	Venomax C	Tablets	100	Jayantha Medi Care	Wankadawwa	P002	O003	Panadol	SPC	80	GVC Pet	Kalutara North	P003	O004	Zincovit	Tablets	50	Suwana Medi House	Kalutara South	P004	O005	Optoperif	Liquid	15	Ludez Pharmacy	Pavagala	P005
Order_ID	Drug_Name	Drug_Brand	Quantity	Pharmacy_Name	Branch	Pharmacy_Staff_ID																																					
O001	Panadol	SPC	10	Oil Medi Pet	Morontaduwa	P001																																					
O002	Venomax C	Tablets	100	Jayantha Medi Care	Wankadawwa	P002																																					
O003	Panadol	SPC	80	GVC Pet	Kalutara North	P003																																					
O004	Zincovit	Tablets	50	Suwana Medi House	Kalutara South	P004																																					
O005	Optoperif	Liquid	15	Ludez Pharmacy	Pavagala	P005																																					
CONCLUSION	The expected table appeared with all the records and outcomes are very clear																																										
STATUS (PASS/FAIL)	Pass																																										

## TC11 - Order Drugs

TEST CASE ID	TC11.1
TEST CASE NAME	Check Get Branch & Staff ID button empty fields
DESCRIPTION	Have to press on Get Branch & Staff ID button when the interface loaded
TEST STEPS	Press on Get Branch & Staff ID button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC11.2
TEST CASE NAME	Check working Get Branch & Staff ID button
DESCRIPTION	Enter a pharmacy name and enter Get Branch & Staff ID button
TEST STEPS	Press on Get Branch & Staff ID button when the interface is loaded
TEST DATA	Get Branch & Staff ID button correctness
EXPECTED RESULT	Should branch and pharmacy staff id fields get filled automatically
ACTUAL RESULT	
CONCLUSION	As expected, the branch name and pharmacy staff id appeared in the branch and pharmacy staff id fields after entering the button Get Branch & Staff ID button and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC11.3
TEST CASE NAME	Check Order Drugs empty fields
DESCRIPTION	Have to press on Place an Order button without filling any fields
TEST STEPS	Press on Place an Order button when the interface is loaded
TEST DATA	Empty fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	

localhost:60800/PharmacyOrderStockWF.aspx

Order Drugs

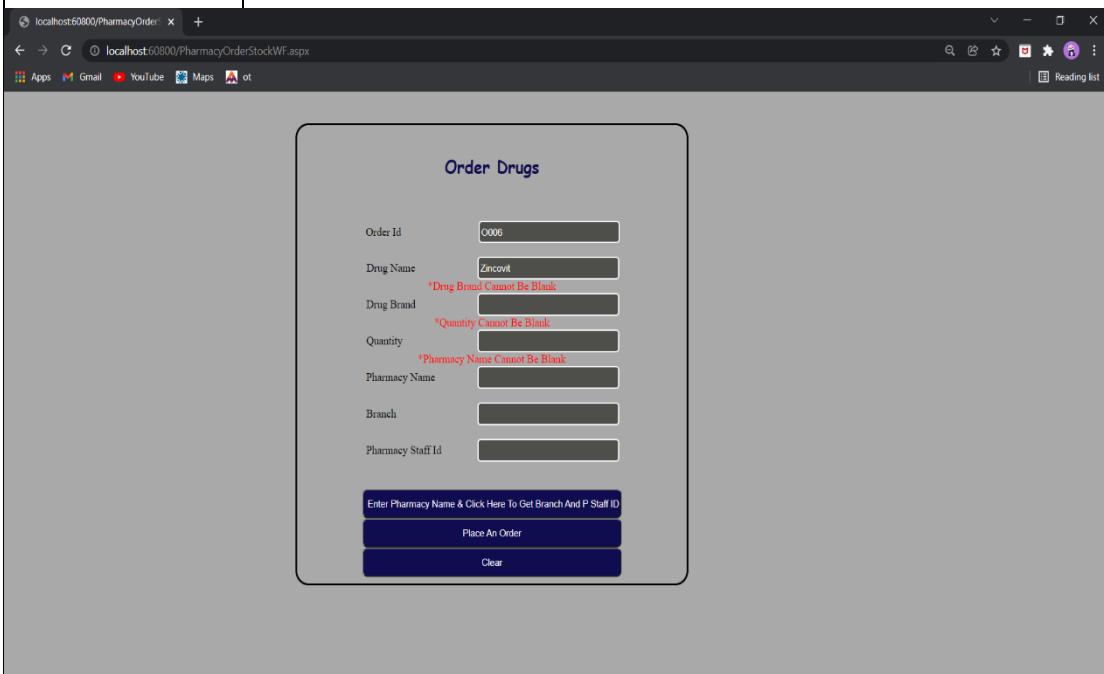
Order Id	0006
Drug Name	*Drug Name Cannot Be Blank
Drug Brand	*Drug Brand Cannot Be Blank
Quantity	*Quantity Cannot Be Blank
Pharmacy Name	*Pharmacy Name Cannot Be Blank
Branch	
Pharmacy Staff Id	

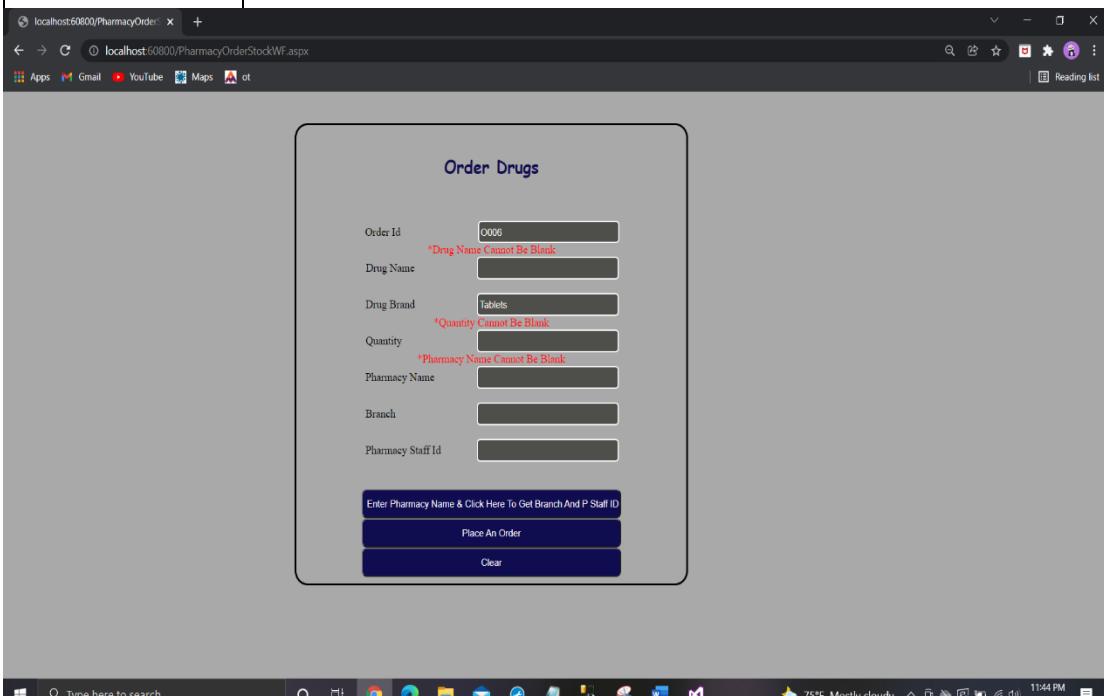
Enter Pharmacy Name & Click Here To Get Branch And P Staff ID

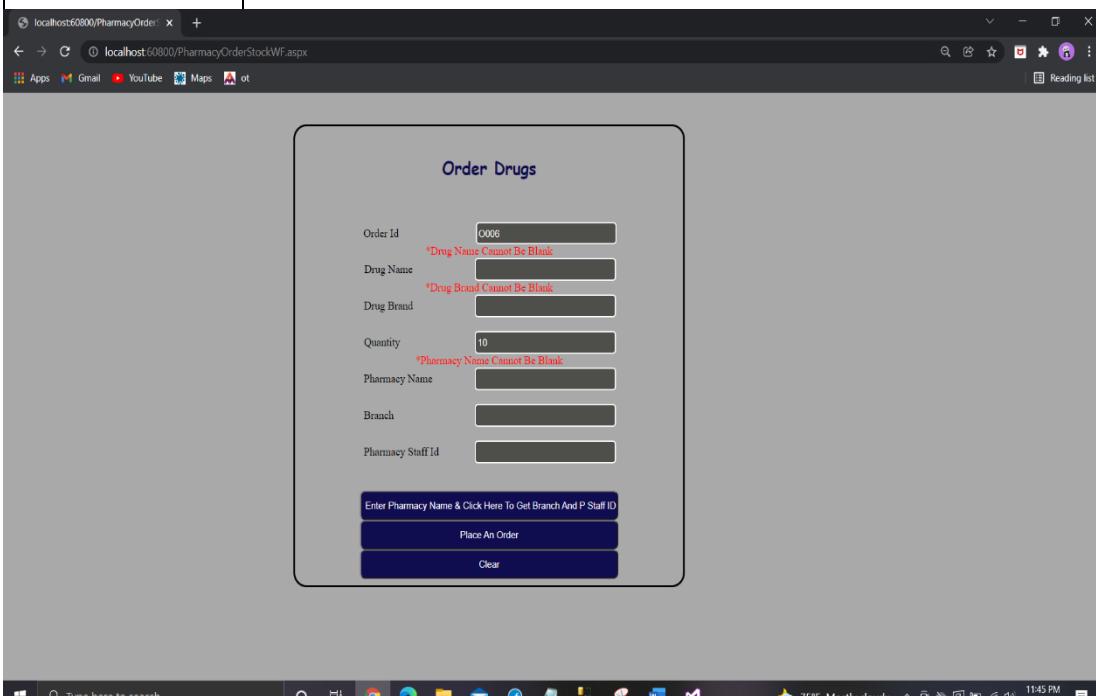
Place An Order

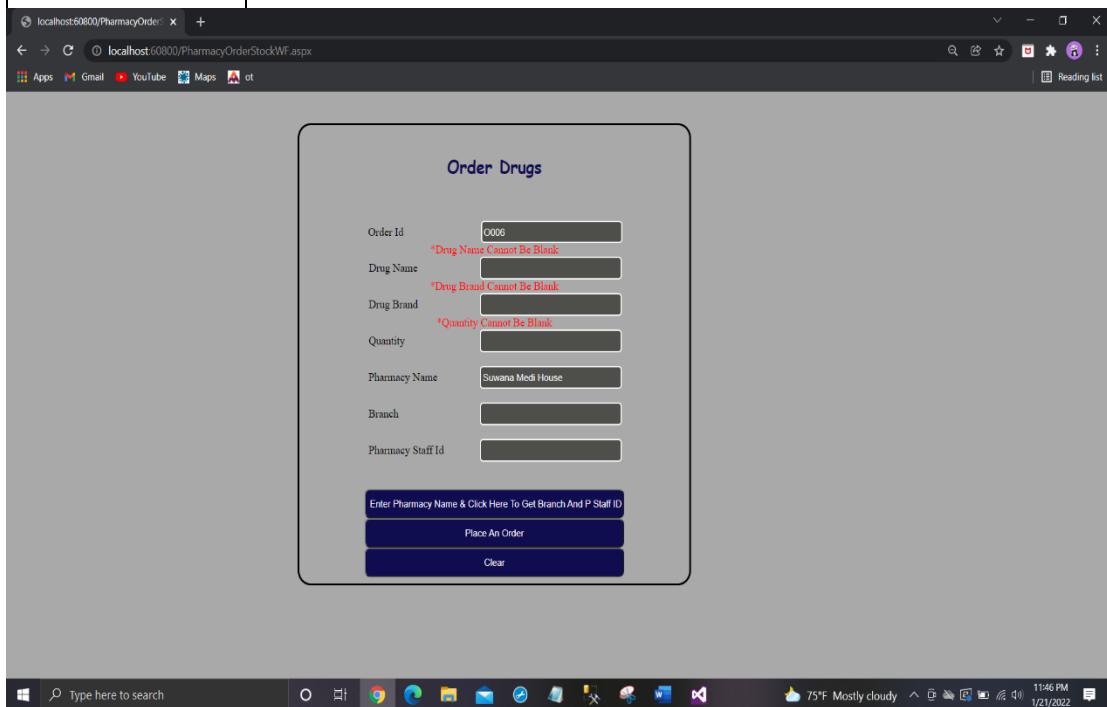
Clear

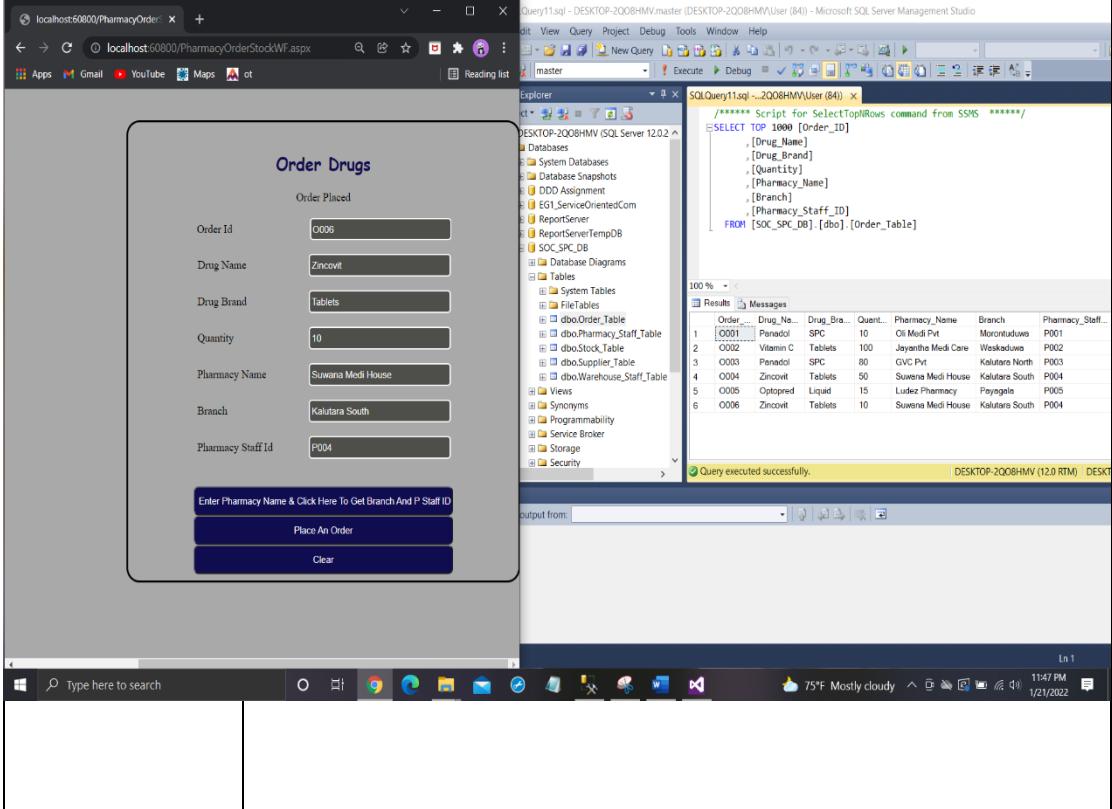
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC11.4
TEST CASE NAME	Check other empty fields by filling only drug name field
DESCRIPTION	Have to press on Place an Order button by filling only drug name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only Drug Name field</li> <li>2.Press on Place an Order button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

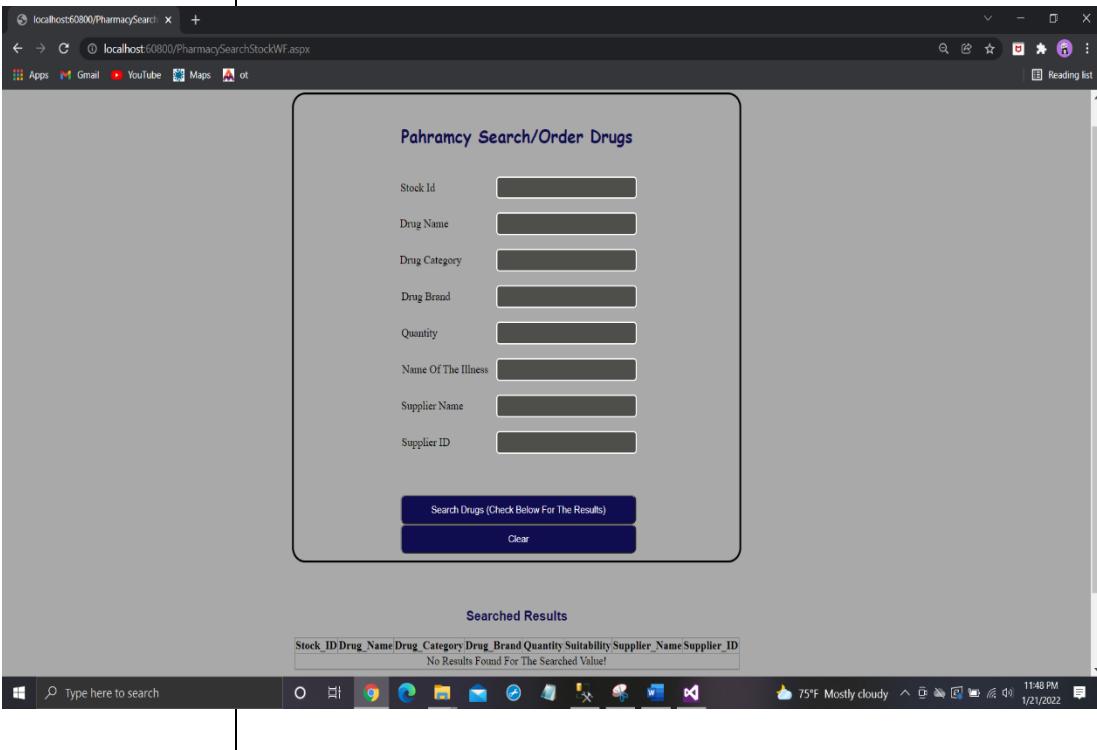
TEST CASE ID	TC11.5
TEST CASE NAME	Check other empty fields by filling only drug brand field
DESCRIPTION	Have to press on Place an Order button by filling only drug brand field
TEST STEPS	1.Fill only Drug brand field 2.Press on Place an Order button
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

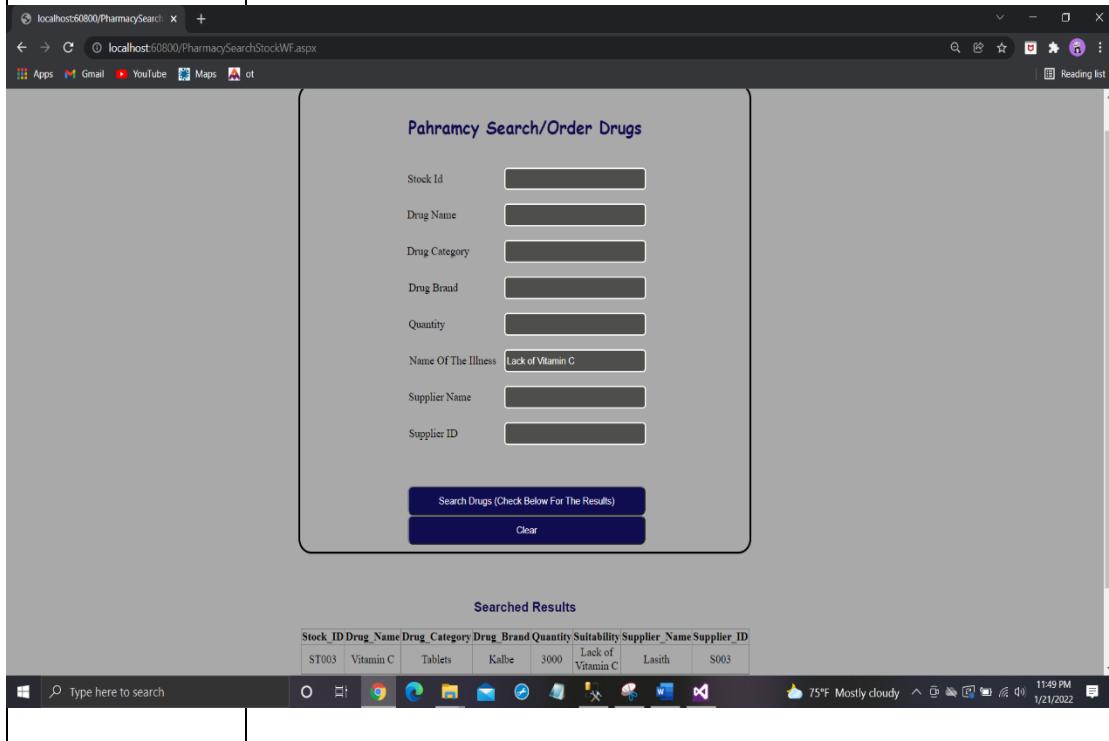
TEST CASE ID	TC11.6
TEST CASE NAME	Check other empty fields by filling only quantity field
DESCRIPTION	Have to press on Place an Order button by filling only quantity field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only quantity field</li> <li>2.Press on Place an Order button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC11.7
TEST CASE NAME	Check other empty fields by filling only pharmacy name field
DESCRIPTION	Have to press on Place an Order button by filling only pharmacy name field
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill only pharmacy name field</li> <li>2.Press on Place an Order button</li> </ol>
TEST DATA	Empty Fields
EXPECTED RESULT	Should display Error Messages on each field except Order id, Branch and pharmacy staff id as (field name) can't be blank
ACTUAL RESULT	 A screenshot of a web browser window titled "localhost:60800/PharmacyOrder". The URL bar shows "localhost:60800/PharmacyOrderStockWF.aspx". The main content area has a title "Order Drugs". There are seven input fields: "Order Id" (value: 0006), "Drug Name" (error message: *Drug Name Cannot Be Blank), "Drug Brand" (error message: *Drug Brand Cannot Be Blank), "Quantity" (error message: *Quantity Cannot Be Blank), "Pharmacy Name" (value: Suwana Medi House), "Branch" (empty), and "Pharmacy Staff Id" (empty). Below the fields is a button "Enter Pharmacy Name & Click Here To Get Branch And P Staff ID". At the bottom are buttons "Place An Order" and "Clear". The browser's taskbar at the bottom shows various pinned icons.
CONCLUSION	The expected error messages were displayed on each text boxes and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC11.8																																																	
TEST CASE NAME	Check Place an Order working without any empty fields																																																	
DESCRIPTION	Have to fill all fields and press Place an Order button																																																	
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill all fields</li> <li>2.Press on Place an Order button</li> </ol>																																																	
TEST DATA	<p>Drug name: Zincovit          Drug Brand: Tablets          Quantity: 10          Pharmacy Name: Suwana Medi House          Branch: Kalutara South          Pharmacy Staff Id: P004</p>																																																	
EXPECTED RESULT	Should display a message as Order Placed Successfully																																																	
ACTUAL RESULT																																																		
 <table border="1"> <thead> <tr> <th>Order_Id</th> <th>Drug_Name</th> <th>Drug_Brand</th> <th>Quantity</th> <th>Pharmacy_Name</th> <th>Branch</th> <th>Pharmacy_Staff_ID</th> </tr> </thead> <tbody> <tr> <td>0001</td> <td>Panadol</td> <td>SPC</td> <td>10</td> <td>Oli Medi Pvt</td> <td>Moratuwae</td> <td>P001</td> </tr> <tr> <td>0002</td> <td>Vitamin C</td> <td>Tablets</td> <td>100</td> <td>Jayathme Medi Care</td> <td>Weskaduwa</td> <td>P002</td> </tr> <tr> <td>0003</td> <td>Panadol</td> <td>SPC</td> <td>80</td> <td>GVC Pvt</td> <td>Kalutara North</td> <td>P003</td> </tr> <tr> <td>0004</td> <td>Zincovit</td> <td>Tablets</td> <td>50</td> <td>Suwana Medi House</td> <td>Kalutara South</td> <td>P004</td> </tr> <tr> <td>0005</td> <td>Optopred</td> <td>Liquid</td> <td>15</td> <td>Ludez Pharmacy</td> <td>Peyagala</td> <td>P005</td> </tr> <tr> <td>0006</td> <td>Zincovit</td> <td>Tablets</td> <td>10</td> <td>Suwana Medi House</td> <td>Kalutara South</td> <td>P004</td> </tr> </tbody> </table>		Order_Id	Drug_Name	Drug_Brand	Quantity	Pharmacy_Name	Branch	Pharmacy_Staff_ID	0001	Panadol	SPC	10	Oli Medi Pvt	Moratuwae	P001	0002	Vitamin C	Tablets	100	Jayathme Medi Care	Weskaduwa	P002	0003	Panadol	SPC	80	GVC Pvt	Kalutara North	P003	0004	Zincovit	Tablets	50	Suwana Medi House	Kalutara South	P004	0005	Optopred	Liquid	15	Ludez Pharmacy	Peyagala	P005	0006	Zincovit	Tablets	10	Suwana Medi House	Kalutara South	P004
Order_Id	Drug_Name	Drug_Brand	Quantity	Pharmacy_Name	Branch	Pharmacy_Staff_ID																																												
0001	Panadol	SPC	10	Oli Medi Pvt	Moratuwae	P001																																												
0002	Vitamin C	Tablets	100	Jayathme Medi Care	Weskaduwa	P002																																												
0003	Panadol	SPC	80	GVC Pvt	Kalutara North	P003																																												
0004	Zincovit	Tablets	50	Suwana Medi House	Kalutara South	P004																																												
0005	Optopred	Liquid	15	Ludez Pharmacy	Peyagala	P005																																												
0006	Zincovit	Tablets	10	Suwana Medi House	Kalutara South	P004																																												
CONCLUSION	The expected successful message was displayed and outcomes are very clear																																																	
STATUS (PASS/FAIL)	Pass																																																	

## TC12 - Pharmacy Search Drugs

TEST CASE ID	TC12.1
TEST CASE NAME	Check search drugs button with empty fields
DESCRIPTION	Press search drugs button without filling any field
TEST STEPS	Just press on Search drugs button when the interface is loaded
TEST DATA	Empty Fields
EXPECTED RESULT	A table should appear and show an Error Message as No Results Found
ACTUAL RESULT	
CONCLUSION	The expected table appeared with the error message and outcomes are very clear
STATUS (PASS/FAIL)	Pass

TEST CASE ID	TC12.2
TEST CASE NAME	Check search drugs button
DESCRIPTION	Have to enter any field except quantity field and press on search drugs button
TEST STEPS	<ol style="list-style-type: none"> <li>1.Fill any field</li> <li>2.Press on Search drugs button</li> </ol>
TEST DATA	Name of the illness: Lack of Vitamin C
EXPECTED RESULT	All fields should be filled with the relevant values and a table should be appeared with the values in the particular database table row
ACTUAL RESULT	
CONCLUSION	The expected table appeared with the searched results and outcomes are very clear
STATUS (PASS/FAIL)	Pass

## **TASK 4- Deployment Techniques**

### **4.1) What is Software Deployment**

Software deployment refers to many methods and procedures that are used to make system software available to users. One of the most important aspects of every Software Development Life Cycle and development project is the software deployment procedure. When the development work on a program is finished, the task isn't finished. The program will not be ready to provide its intended purpose until it is correctly deployed. That is why software deployment is so critical to the overall system. Further, it is not recommended that software be deployed unless it has been extensively tested. This is done to make sure there are no bugs or drawbacks. (REYNOLDS, 2022)

### **4.2) Local Deployment & Global Deployment**

In Software Deployment, there are two categories as Local Deployment and Global Deployment. In Local Deployment, the complete program is placed within a virtual location which contains and makes available all of the components and modules whereas components are available to all server-side applications in Global Deployment. (tutorialspoint, 2022)

### **4.3) Some Deployment Techniques**

#### **4.3.1) Copying a website**

When it comes to the deployment technique of copying a website, it is a Visual Studio tool which allows to copy the contents of a currently existing website to another place where it's usable. It's best described as an integrated FTP tool. This tool can be used to connect to the chosen copy type at the targeted destination. As mentioned above, the Visual Studio environment can be used to cooperate with the deployment process. Unfortunately, there are some reasons not to select this technique as the deployment technique of SPC system. They are because the source is being shared, the techniques for pre-compilation and data validation are ignored and the opening page will take so long to load. (D., 2022)

#### 4.3.2) Docker

As (Stephen J. Bigelow, 2022)says, Docker is an open source software platform to create, deploy and manage virtualized application containers on a common operating system (OS), with an ecosystem of allied tools. Initially Docker was designed to run only on Linux. By now, it has been expanded to be used for some other operating systems except Linux such as Microsoft Windows and Apple OS X. Moreover, there are Docker editions for Microsoft Azure as well as for Amazon Web Services. Importantly, to run several containers on the same OS, Docker is using resource isolation in the OS kernel. Basically, a container contains all of the libraries, configuration files, dependencies, other components and required limits to run an application service or a function. (Konrad Gos, 2022)

Even though Docker's flexibility lets users to sign up and share containers across many servers in both personal and open domains, faster deployment, good security and lower management cost in this Docker Deployment Technique, there are some issues with this technique which led to not to select this deployment technique for SPC system. Because there are numerous functionalities that are lacking, like container self-registration and file transfer from the host to the container. Further, when a container fails, it requires a backup and recovery. Even though, it has numerous solutions for this, they are not automated. When compared to virtual computers, the Docker doesn't offer a lower expenditure. Furthermore, it's challenging to keep track of various moving parts in a large-scale, dynamic Docker system. (DataFlair, 2022)

#### 4.3.3) Kubernetes

A Kubernetes deployment is a service type in Kubernetes that allows applications to collect continuous modifications. A deployment help to specify details about an application's life cycle, such as which images to apply and how they should be modified. A Kubernetes deployment makes the process automated and repeatable. The Kubernetes backend handles all deployments, and the entire update activity is handled on the server without any client involvement. (Hat, 2022)

Despite the fact that Kubernetes enables developers to deploy workloads across the whole cluster, track multiple parallelly running containers, and run any application that can run in a container, there are a few reasons why this technique should not be chosen as the best matching deployment technique for the SPC system. Those reasons are, Kubernetes might be a little too much at times, especially when considering local improvements and basic programs that can impact on performance. In addition, the adjustment to Kubernetes may be time-consuming, complex, and difficult to operate. Furthermore, the growing number of connections and new technologies may also be stressful for several deployment procedures, as it spends a significant amount of time, resources and energy. (Taylor, 2022)

## **4.4) The most suitable Deployment Technique for the SPC System**

### **4.4.1) Creating a setup project**

As the deployment technique for the system of the SPC project, Creating A Setup Project has been selected. Thus, the below paragraphs highlight the reasons and enough justifications for selecting that deployment technique. Windows Installer is used in this technique to package a web application for deployment on the production server. Through Visual Studio IDE, the building of deployment packages can be done easily with few steps. As an instance for this method, a data binding project can be pointed out. (MacDonald, 2022)

Below two points shows the two files created after successfully done the build process;

- Setup.exe
- Setup-databinding.msi

The above pointed two files must be moved to the server. Further, a double-click on the setup file can be used for the installation of the .msi file's information on some kind of selected local machine. ASP.NET development services are used this step to deploy a particular application. When it comes to ASP.Net framework, it has become a popular choice among web developers by now. It is one of the mostly used programming environments. Also, it's much easier and simpler to create user friendly and dynamic websites as it provides a user-friendly, dynamic website.

When it comes to the AS.Net Project Deployment process, there are certain task to be done. At first, the IIS server should be installed. After installing IIS server, have to run IIS Manager in order to ensure that the .NET Framework version is assigned to the default application pool. Finally, we have everything we need to publish a web project to IIS. (D., 2022) (tutorialspoint, 2022)

Complex programs may require a more flexible environment for a number of applications. As a result, developing setup projects enables the creation of absolute and easily customizable deployment methods. It also enables for the creation of files on the target machines, registry changes while installation and the deployment of custom operations while installing, etc. Furthermore, as it enables to publish the

program and its supporting files to a location; for instance, a web site, then users can simply visit there just to download. Also, it's possible to publish an application to a shared hosting provider and then users will be able to install it directly. One of the most significant advantages is the simplicity with which updates may be published. All that is required is for the new version to be published to the previous location. On the next startup of the program, all users will receive the updates. Thus, those were the main reasons to choose Creating A Setup Project deployment technique which is available in the visual studio as the deployment technique of the SPC Web Application.

(Prasad, 2022)

## References

- Bloomberg, J. (2022, January 06). *How to Avoid SOA Chaos*. Retrieved from ZapThink: <https://doveltech.com/wp-content/uploads/2017/09/AvoidSOAChaos-Systinet-102004-ZTP-0161-1.pdf>
- D., S. (2022, January 08). *What Is The ASP.NET Project Deployment And Maintenance Strategy?* Retrieved from CIS: <https://www.cisin.com/coffee-break/enterprise/what-is-the-asp-net-project-deployment-and-maintenance-strategy.html>
- DataFlair. (2022, January 10). *Advantages and Disadvantages of Docker – Learn Docker*. Retrieved from DataFlair: <https://data-flair.training/blogs/advantages-and-disadvantages-of-docker/>
- Haq, S. u. (2022, January 02). *Introduction to Monolithic Architecture and MicroServices Architecture*. Retrieved from KODELABS: <https://medium.com/koderlabs/introduction-to-monolithic-architecture-and-microservices-architecture-b211a5955c63>
- Hat, R. (2022, January 11). *What is a Kubernetes deployment*. Retrieved from Red Hat: <https://www.redhat.com/en/topics/containers/what-is-kubernetes-deployment>
- integrate.io. (2022, January 03). *What is Monolithic Architecture?* Retrieved from integrate.io: <https://www.integrate.io/glossary/what-is-monolithic-architecture/>
- Kalske, M. (2022, January 02). *HELDA*. Retrieved from Transforming monolithic architecture towards microservice architecture: <https://helda.helsinki.fi/handle/10138/234239>
- Kathryn B. Laskey, K. L. (2022, January 05). *Service oriented architecture*. Retrieved from WIREs: <https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wics.8>
- Konrad Gos, W. Z. (2022, January 05). *The Comparison of Microservice and Monolithic*. Retrieved from IEEE: [https://www.researchgate.net/profile/Wojciech-Zabierowski/publication/341956559\\_The\\_Comparison\\_of\\_Microservice\\_and\\_Monolithic\\_Architecture/links/5edf80fe299bf1d20bdb24e2/The-Comparison-of-Microservice-and-Monolithic-Architecture.pdf](https://www.researchgate.net/profile/Wojciech-Zabierowski/publication/341956559_The_Comparison_of_Microservice_and_Monolithic_Architecture/links/5edf80fe299bf1d20bdb24e2/The-Comparison-of-Microservice-and-Monolithic-Architecture.pdf)
- MacDonald, M. (2022, January 13). *ClickOnce Deployment*. Retrieved from SpringerLink: [https://link.springer.com/chapter/10.1007/978-1-4302-4684-8\\_33](https://link.springer.com/chapter/10.1007/978-1-4302-4684-8_33)
- Prasad, V. (2022, January 13). *Creating Setup and Deployment Projects in VS.NET*. Retrieved from C#Corner: <https://www.c-sharpcorner.com/article/creating-setup-and-deployment-projects-in-vs-net/>
- REYNOLDS, A. J. (2022, January 07). *Everything You Need to Know About the Software Deployment Process*. Retrieved from Zibtek: <https://www.zibtek.com/blog/everything-you-need-to-know-about-software-deployment/>
- Stephen J. Bigelow. (2022, January 10). *Docker*. Retrieved from TechTarget: <https://searchitoperations.techtarget.com/definition/Docker>
- Taylor, K. (2022, January 12). *Pros and Cons of Kubernetes Explained*. Retrieved from HighTechNectar: <https://www.hitechnectar.com/blogs/pros-cons-kubernetes/>

- tutorialspoint. (2022, January 07). *ASP.NET - Deployment*. Retrieved from tutorialspoint:  
[https://www.tutorialspoint.com/asp.net/asp.net\\_deployment.htm#:~:text=Loca](https://www.tutorialspoint.com/asp.net/asp.net_deployment.htm#:~:text=Loca)  
1%20deployment%20%3A%20In%20this%20case,application%20running%2  
0on%20the%20server
- Walker, A. (2022, January 06). *What is SOA? Service-Oriented Architecture Principles*. Retrieved from Guru99: <https://www.guru99.com/soa-principles.html>
- Wilkes, D. S. (2022, January 05). *Understanding Service-Oriented*. Retrieved from Understanding Service-Oriented:  
<file:///C:/Users/User/Downloads/Understanding%20Service-Oriented%20Architecture.pdf>

## **Conclusion**

However, I could get many advantages from this course work such as how to solve doubts and issues, how to gather requirements needed, how to do the analyzation properly, how to manage time effectively, how to overcome from errors which are arising when writing the code and so on. Finally...with all above mentioned details and me proper time management, I could prepare a successful document along with a SOA based Web Application to complete this assignment. Thus, I hope this assignment been a great help for me to get learnt about Service Oriented Computing and to prove that I have successfully completed my eighteenth assignment in my HD Program. Moreover, during this Service Oriented Computing assessment, I learnt about many more things which are essential in designing and about coding while learning C# programming language more and many more other things. Developing of this SOA based Web Application for SPC gave me the chance to try my new skills in practice. Specially this recalled my knowledge for analyzing, C# coding and OOP. So, I have built the application in a user-friendly manner including pharmacy staff registration, warehouse staff registration with a common login which based on User Type, Manager menu, add stock/ orders, view stock, order, etc. While doing this project I also gained deep understanding on programming/coding, services and how it can be implemented in real life situations as now I have the experience in creating a SOA based Web Application for SPC. Thus, I believe that this was a great chance for me to improve my service-oriented computing and computer programming skills.