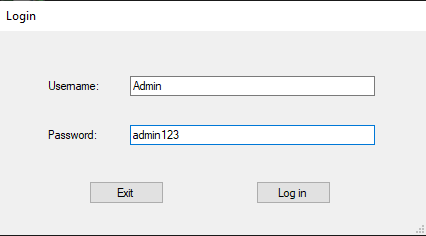
CREDIT TASK 2

***Group members:*** Lê Toàn Hiển - 521K0008

Lê Thái Gia Bảo - 521K0178

**Exercise 1:**

***frmLogin.cs:***



public partial class frmLogin : *Form*

{

public frmLogin()

{

InitializeComponent();

}

private *void* btnLogin\_Click(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*String* sSQL = "SELECT Username, Pass FROM Users WHERE " + "Username = '" + txtUsername.Text + "' AND Pass = '" + txtPassword.Text + "'";

*SqlCommand* cmd = new *SqlCommand*(sSQL, conn);

*SqlDataAdapter* da = new *SqlDataAdapter*(cmd);

*DataTable* dt = new *DataTable*();

da.Fill(dt);

if (dt.Rows.Count > 0)

{

MessageBox.Show("Login Successful!");

*this*.Close();

}

else

{

MessageBox.Show("Invalid login. Please check your username and password then try again.");

}

}

private *void* btnExit\_Click(*object* *sender*, *EventArgs* *e*)

{

Application.Exit();

}

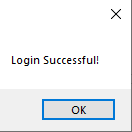
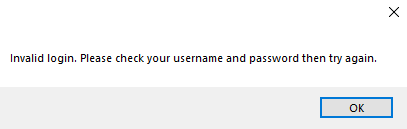
private *void* frmLogin\_Load(*object* *sender*, *EventArgs* *e*)

{

}

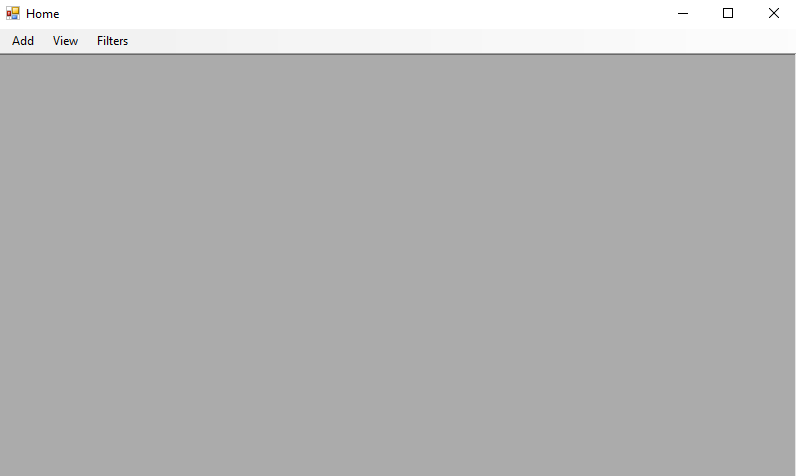
}

The user has to enter the username and password to log in into the application. After entering the information, the user must click the ‘Log in’ button. This will execute the function **btnLogin\_Click()**, the function will check the username and password if it is matched with any usernames and passwords in the database (For demo, we will use the username ‘Admin’ and ‘Admin123’ for the password). If the username and password is correct, there will be a message showing the user that their verification is successful. Otherwise, the form will ask the user to re-enter the information.

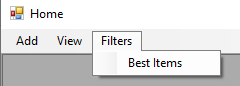
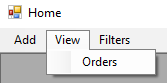
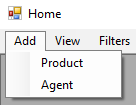
 

In addition, the ‘Exit’ button will immediately close the entire application through the **btnExit\_Click()** function.

***frmMain.cs:***



This form contains 3 menu strips ‘Add’ (Product, Agent), ‘View’ (Orders) and ‘Filters’(Best Items).



public partial class frmMain : *Form*

{

public frmMain()

{

InitializeComponent();

}

private *void* frmMain\_Load(*object* *sender*, *EventArgs* *e*)

{

*frmLogin* frmLogin = new *frmLogin*();

frmLogin.ShowDialog();

}

private *void* productToolStripMenuItem\_Click(*object* *sender*, *EventArgs* *e*)

{

*frmAddProduct* frmAddProduct = new *frmAddProduct*();

frmAddProduct.ShowDialog();

}

private *void* agentToolStripMenuItem\_Click(*object* *sender*, *EventArgs* *e*)

{

*frmAddAgent* frmAddAgent = new *frmAddAgent*();

frmAddAgent.ShowDialog();

}

private *void* ordersToolStripMenuItem\_Click(*object* *sender*, *EventArgs* *e*)

{

*frmOrder* frmOrder = new *frmOrder*();

frmOrder.ShowDialog();

}

private *void* bestItemsToolStripMenuItem\_Click(*object* *sender*, *EventArgs* *e*)

{

*frmBestItems* frmBestItems = new *frmBestItems*();

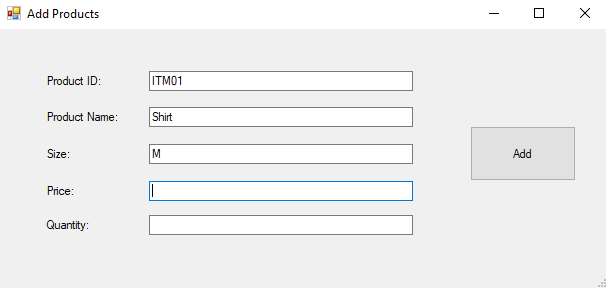
frmBestItems.ShowDialog();

}

}

The form frmMain.cs is mainly used for proceeding to other functionality forms by clicking the strip menu items.

***frmAddProduct.cs:***



The user can use this form for adding new items for sale in the shop by entering the information then clicking the ‘Add’ button.

public partial class frmAddProduct : *Form*

{

public frmAddProduct()

{

InitializeComponent();

}

private *void* btnAdd\_Click(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*String* sSQL = "INSERT INTO Item (ItemID, ItemName, Size, Price, Quantity) VALUES (@ProdID, @ProdName, @ProdSize, @ProdPrice, @ProdQuant)";

*SqlCommand* cmd = new *SqlCommand*(sSQL, conn);

cmd.Parameters.Add(new *SqlParameter*("@ProdID", txtProdID.Text));

cmd.Parameters.Add(new *SqlParameter*("@ProdName", txtProdName.Text));

cmd.Parameters.Add(new *SqlParameter*("@ProdSize", txtProdPrice.Text));

cmd.Parameters.Add(new *SqlParameter*("@ProdPrice", txtProdPrice.Text));

cmd.Parameters.Add(new *SqlParameter*("@ProdQuant", txtProdQuant.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (*Exception* ex)

{

throw new *Exception*("Error: " + ex.Message);

}

MessageBox.Show("Add product successfully!");

}

private *void* frmAddProduct\_Load(*object* *sender*, *EventArgs* *e*)

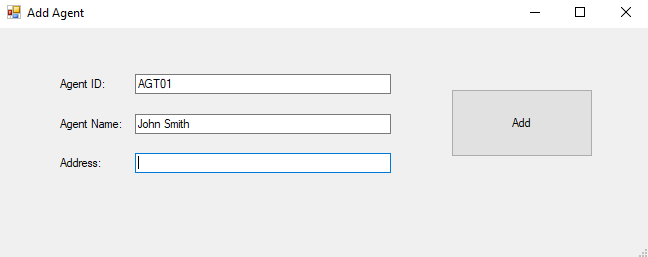
{

}

}

If the products’ information adding process is successful, a message box appears to inform the user, otherwise, it will show the error.

***frmAddAgent.cs:***



public partial class frmAddAgent : *Form*

{

public frmAddAgent()

{

InitializeComponent();

}

private *void* btnAdd\_Click(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*String* sSQL = "INSERT INTO Agent (AgentID, AgentName, Address) VALUES (@AgentID, @AgentName, @Address)";

*SqlCommand* cmd = new *SqlCommand*(sSQL, conn);

cmd.Parameters.Add(new *SqlParameter*("@AgentID", txtAgentID.Text));

cmd.Parameters.Add(new *SqlParameter*("@AgentName", txtAgentName.Text));

cmd.Parameters.Add(new *SqlParameter*("@Address", txtAddress.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (*Exception* ex)

{

throw new *Exception*("Error: " + ex.Message);

}

MessageBox.Show("Add agent successfully!");

}

private *void* frmAddAgent\_Load(*object* *sender*, *EventArgs* *e*)

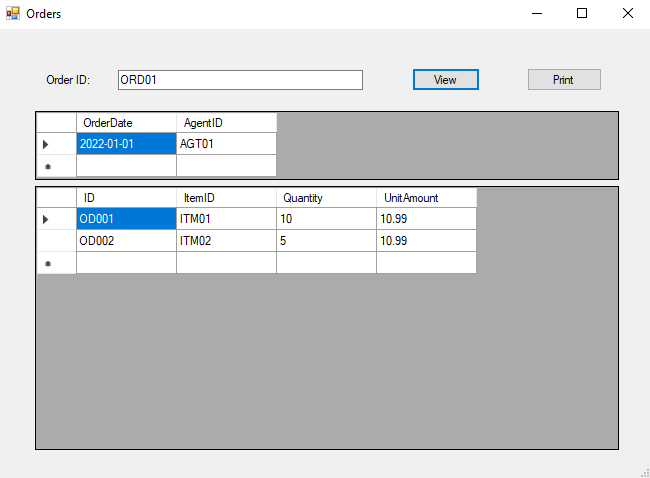
{

}

}

This form works the same as **frmAddProduct.cs**.

***frmOrder.cs:***



This form is used for viewing the information of an order in the database by entering the order ID then selecting the button ‘View’ and printing the order if the user wants to.

public partial class frmOrder : *Form*

{

public frmOrder()

{

InitializeComponent();

}

private *void* frmOrder\_Load(*object* *sender*, *EventArgs* *e*)

{

}

private *void* btnView\_Click(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*String* sSQL1 = "SELECT OrderDate, AgentID FROM Orders WHERE OrderID = '" + txtOrderID.Text + "'";

*String* sSQL2 = "SELECT ID, ItemID, Quantity, UnitAmount FROM OrderDetail WHERE OrderID = '" + txtOrderID.Text + "'";

*SqlCommand* cmd1= new *SqlCommand*(sSQL1, conn);

*SqlDataAdapter* da1 = new *SqlDataAdapter*(cmd1);

*DataTable* dt1 = new *DataTable*();

da1.Fill(dt1);

if(dt1.Rows.Count > 0 )

{

dataGridView2.DataSource = dt1;

}

else

{

MessageBox.Show("Invalid order");

}

*SqlCommand* cmd2 = new *SqlCommand*(sSQL2, conn);

*SqlDataAdapter* da2 = new *SqlDataAdapter*(cmd2);

*DataTable* dt2 = new *DataTable*();

da2.Fill(dt2);

if(dt2.Rows.Count > 0 )

{

dataGridView1.DataSource = dt2;

}

else

{

MessageBox.Show("No data recorded");

}

}

private *void* btnPrint\_Click(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*frmPrint* frmPrint = new *frmPrint*();

frmPrint.Show();

*String* sSQL = "SELECT \* FROM OrderDetail WHERE OrderID = '" + txtOrderID.Text + "'";

*SqlCommand* cmd = new *SqlCommand*(sSQL, conn);

*SqlDataAdapter* da = new *SqlDataAdapter*(cmd);

*DataSet* ds = new *DataSet*();

da.Fill(ds, "OrderDetail");

*rptOrder* rpt = new *rptOrder*();

rpt.SetDataSource(ds);

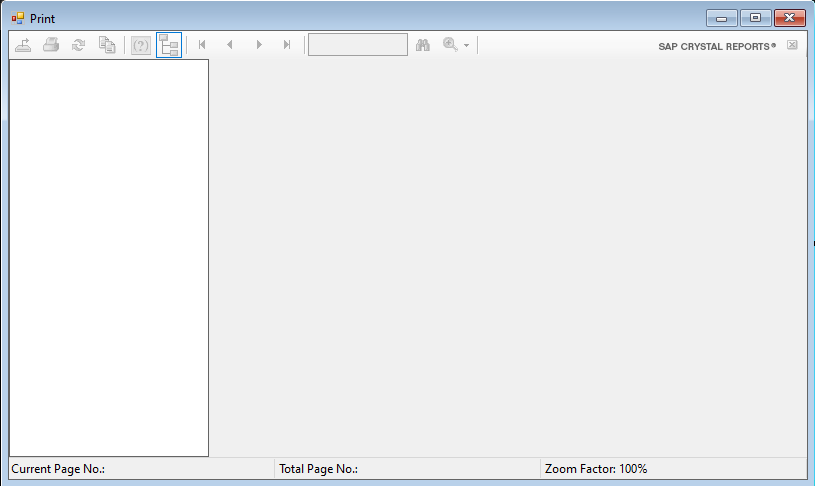
frmPrint.crystalReportViewer1.ReportSource = rpt;

frmPrint.crystalReportViewer1.Refresh();

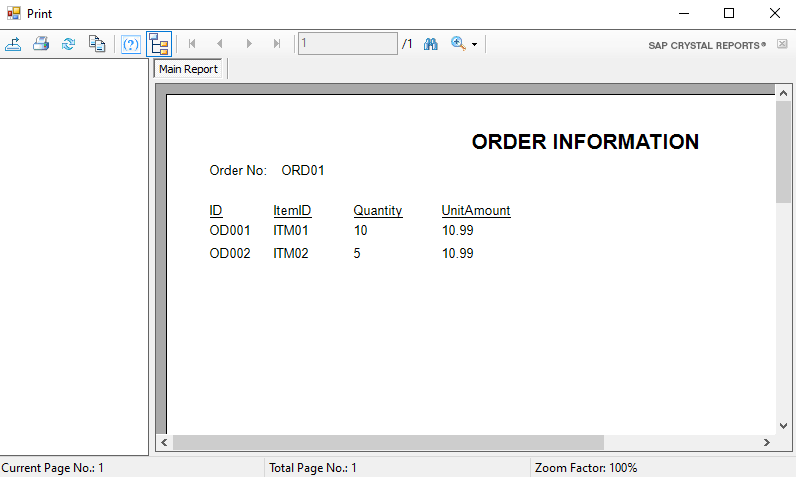
}

}

The user can insert the order’s ID for which order he/she wants to view its information. If the user clicks the ‘Print’ button, the **btnPrint\_Click()** is triggered to be executed.

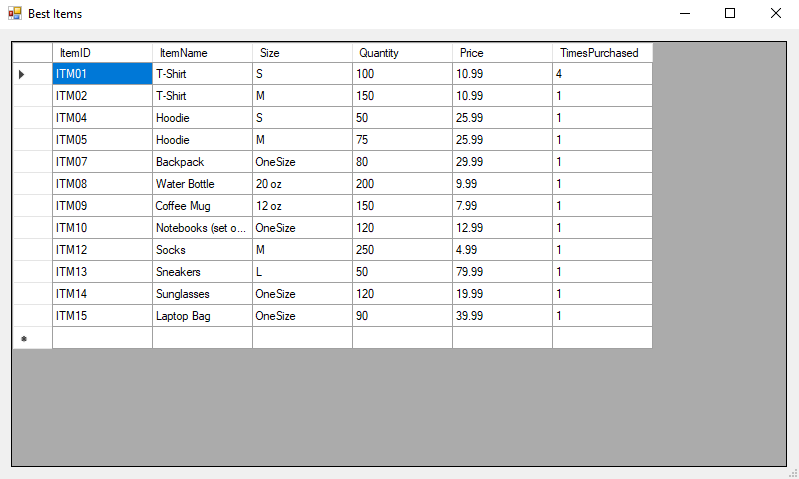


The **frmPrint.cs** form contains the CrystalReportViewer tool provide by **SAP Crystal Report** for viewing the report printed in digital form, such as:



***frmBestItems.cs:***

This form contains a data grid view showing the information of the most popular items purchased by the agents and the times that an item is purchased (ordered by times purchase in decreasing).



public partial class frmBestItems : *Form*

{

public frmBestItems()

{

InitializeComponent();

}

private *void* frmBestItems\_Load(*object* *sender*, *EventArgs* *e*)

{

*String* strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

*SqlConnection* conn = new *SqlConnection*(strConn);

conn.Open();

*String* sSQL = "SELECT I.ItemID, I.ItemName, I.Size, I.Quantity, I.Price, COUNT(O.ItemID) as TimesPurchased FROM Item I JOIN OrderDetail O ON I.ItemID = O.ItemID GROUP BY I.ItemID, I.ItemName, I.Size, I.Quantity, I.Price ORDER BY COUNT(O.ItemID) DESC";

*SqlCommand* cmd = new *SqlCommand*(sSQL, conn);

*SqlDataAdapter* da = new *SqlDataAdapter*(cmd);

*DataTable* dt = new *DataTable*();

da.Fill(dt);

if(dt.Rows.Count > 0)

{

dataGridView1.DataSource = dt;

}

else

{

MessageBox.Show("Error");

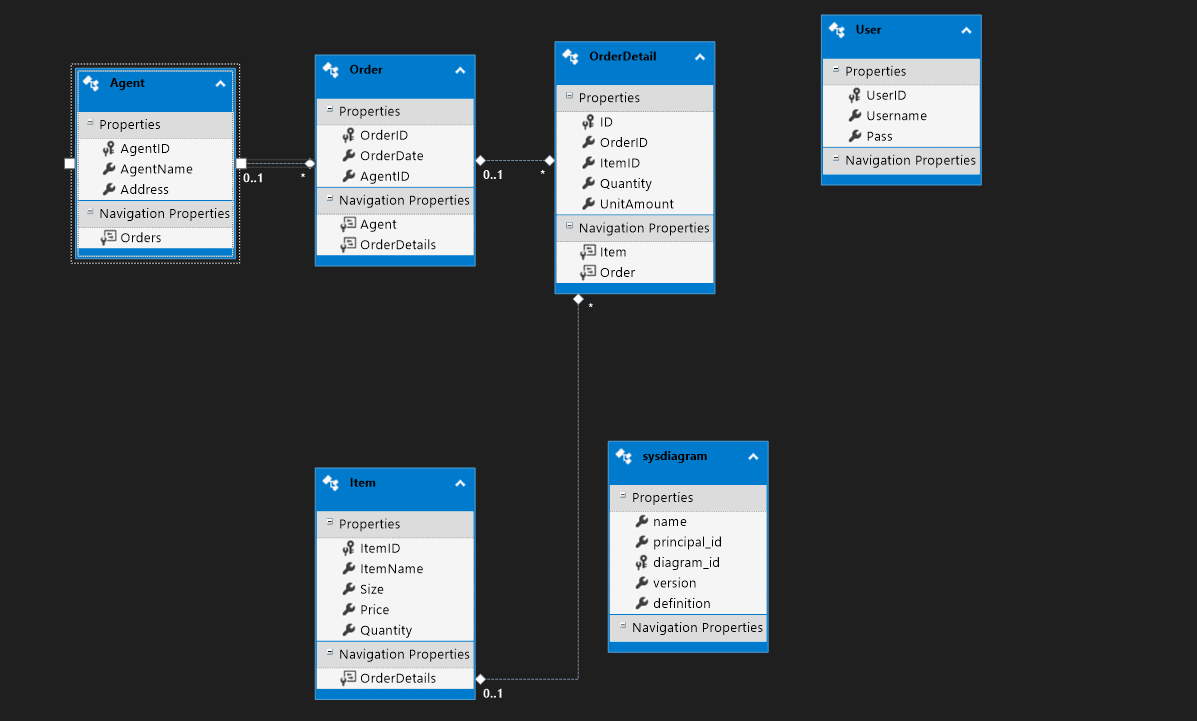
}

}

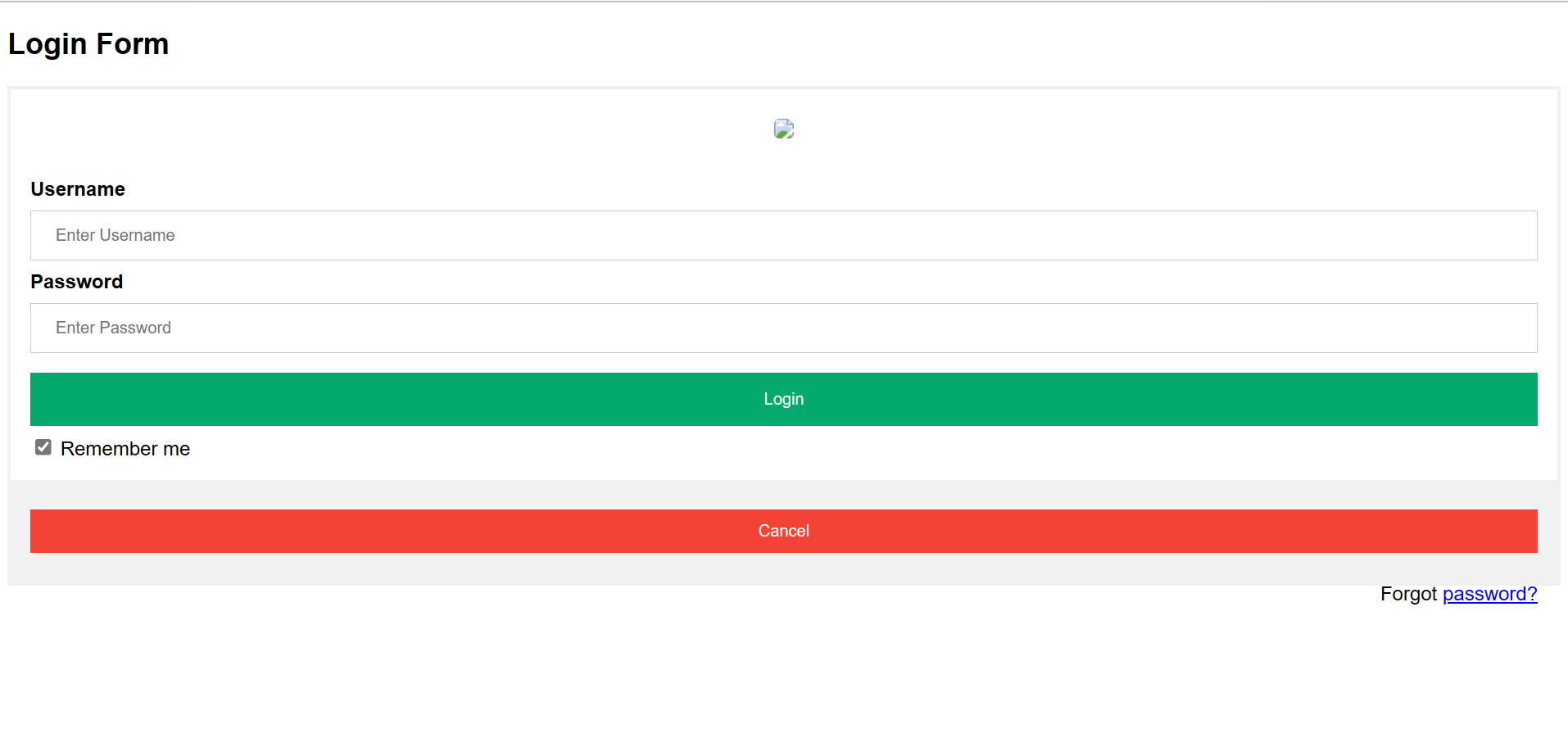
}

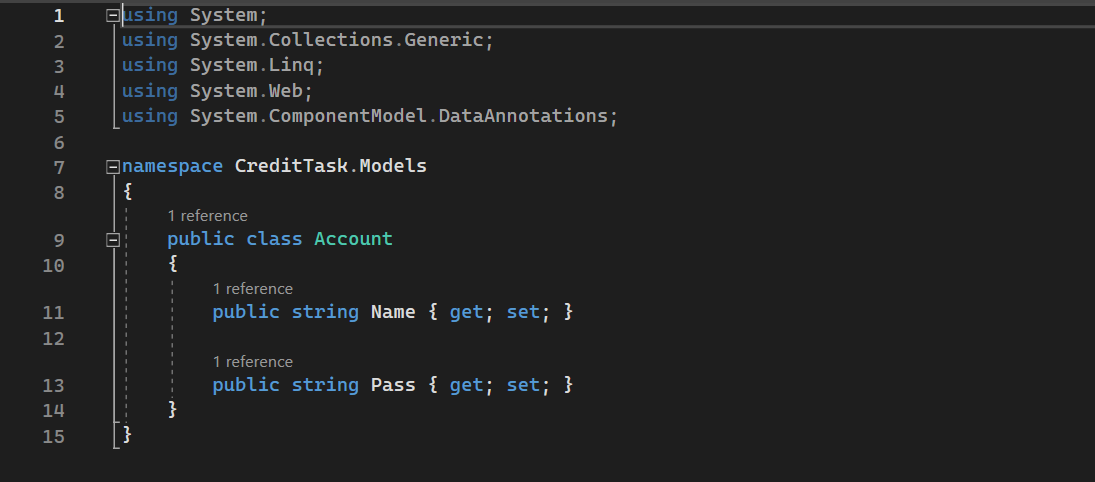
**Exercise 2:**

***Database Model***

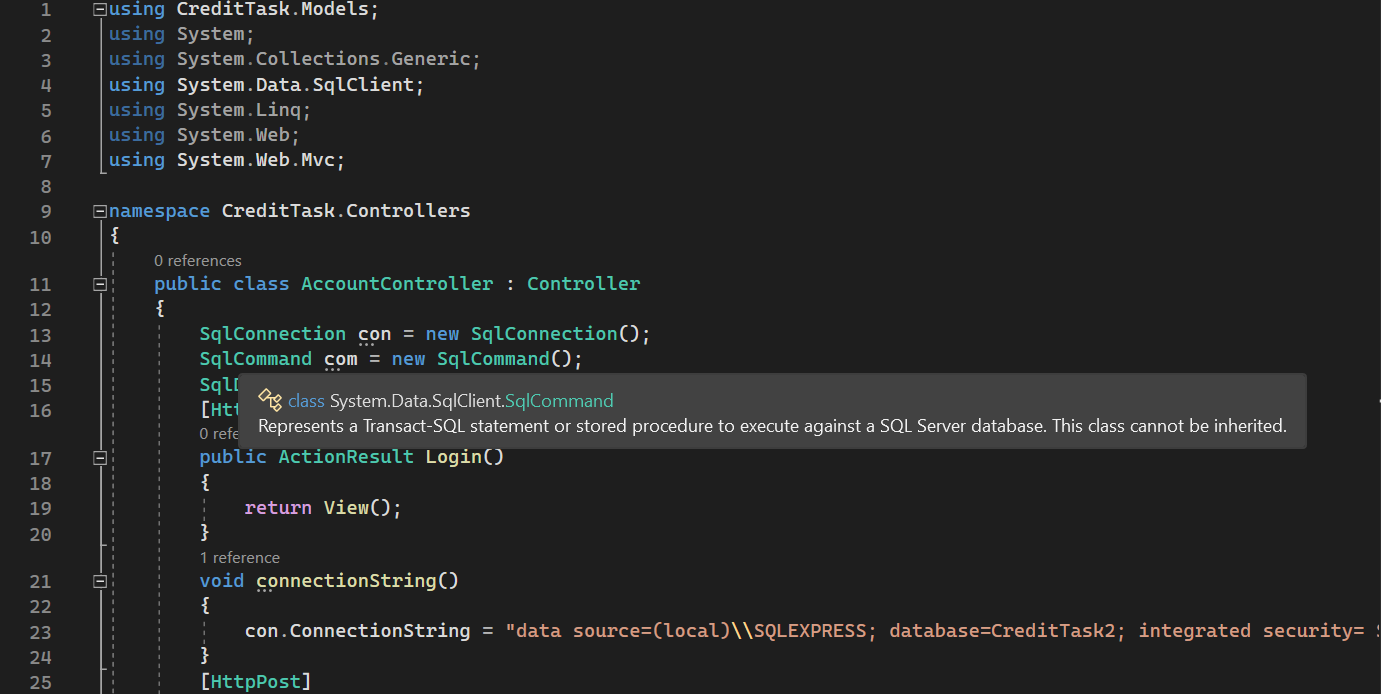
****

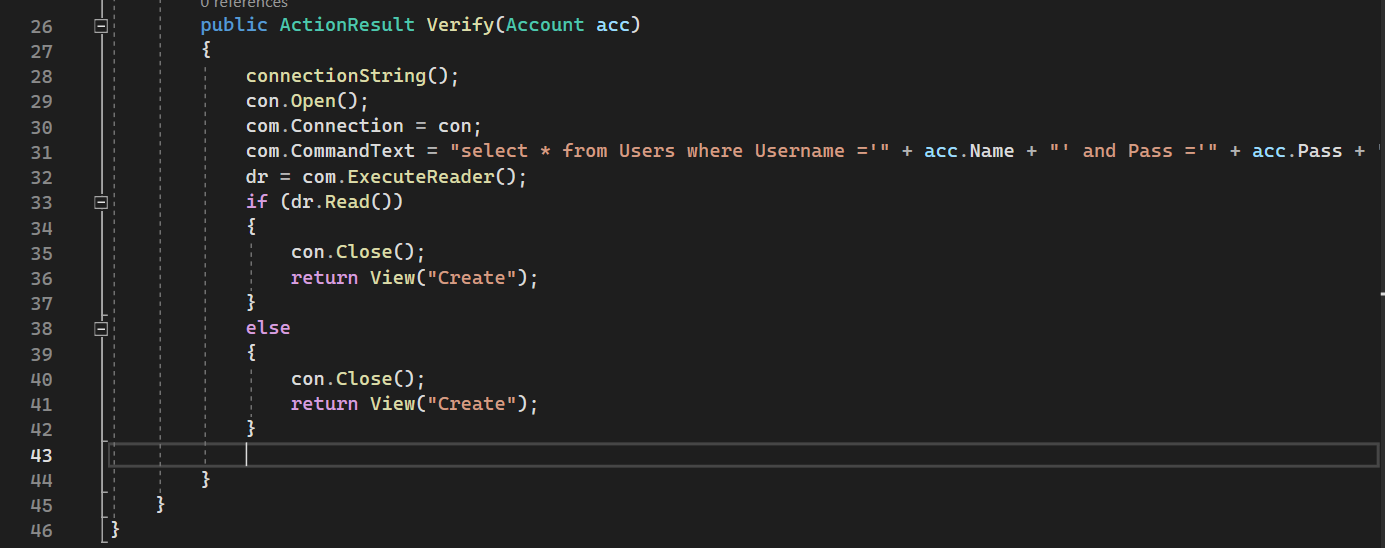
***Login***



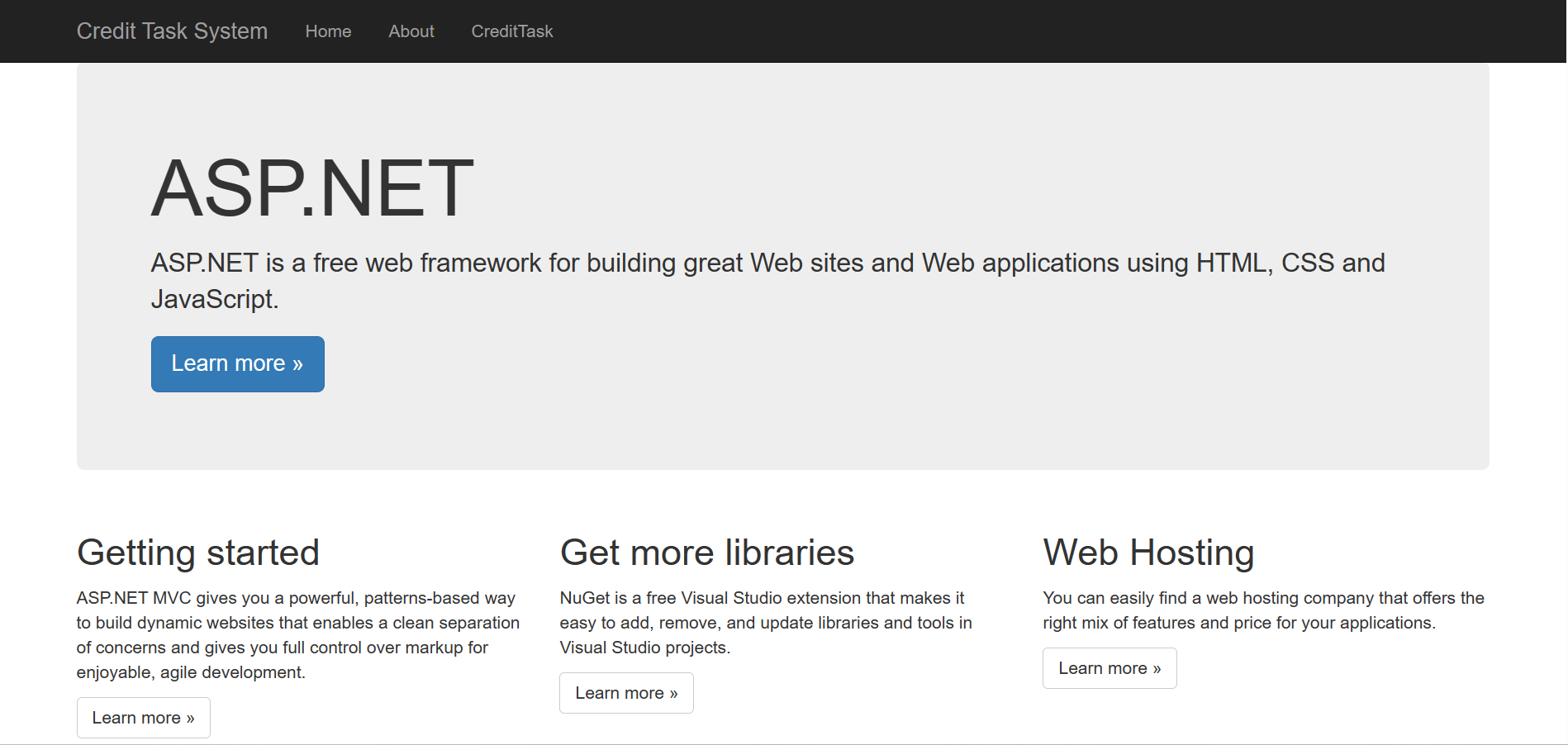


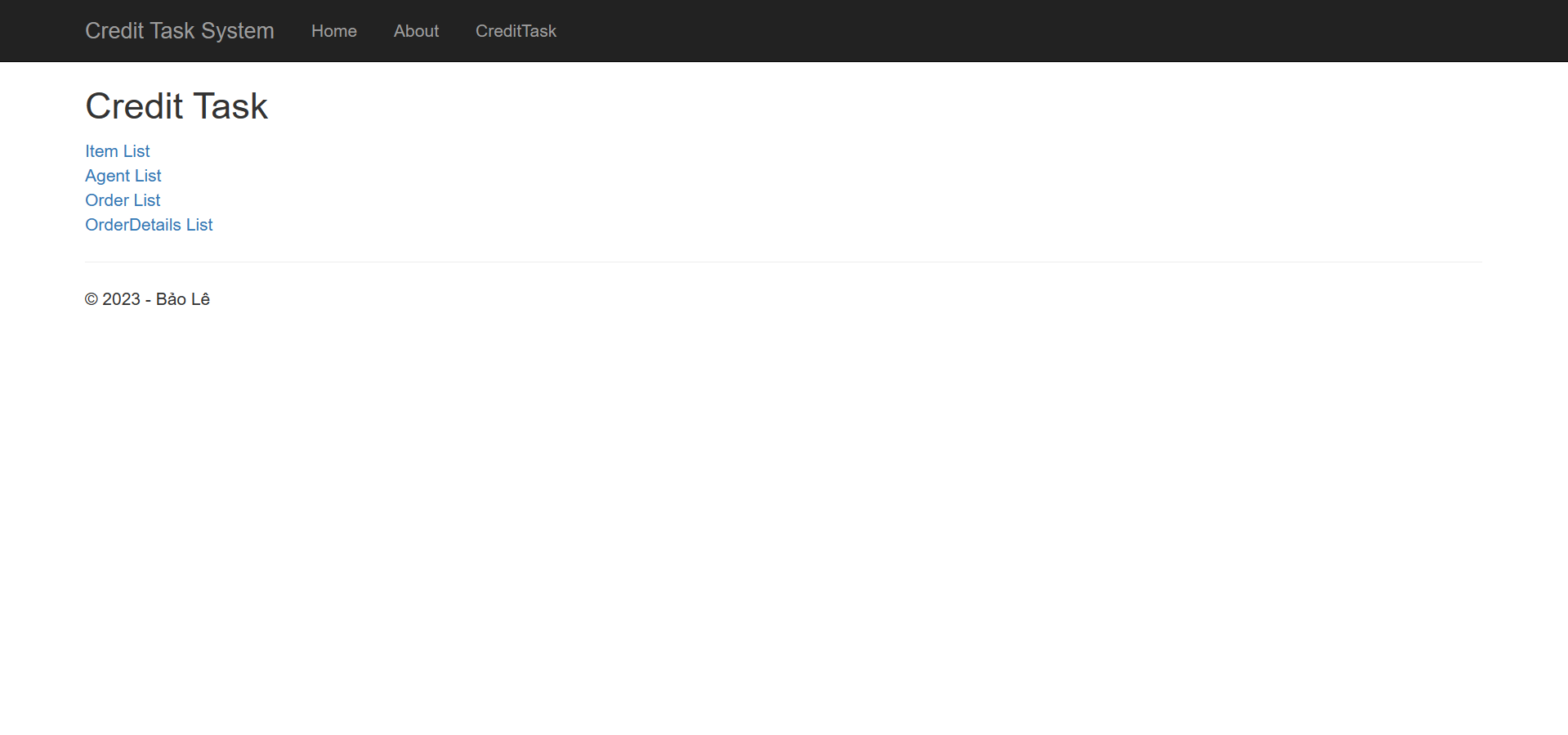
After connecting to the database server, this is code for getting and setting User’s Name and Password through **Account.cs**

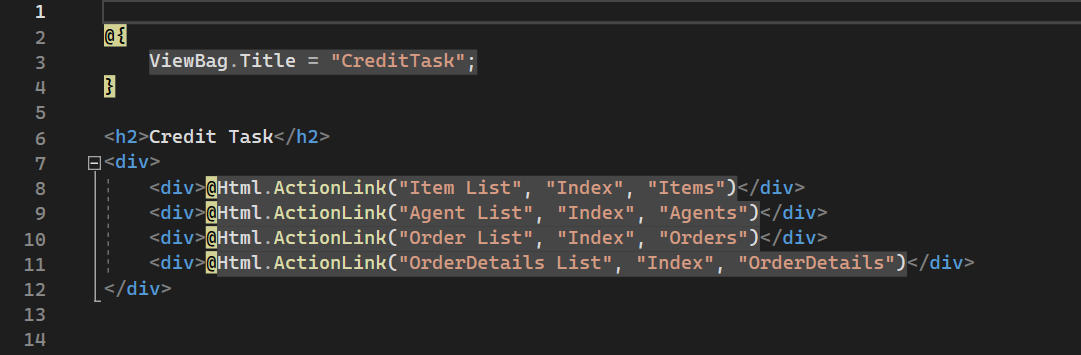
****

****

Then read the user’s information which is matched with available data in the database, and go on connecting to the home page.

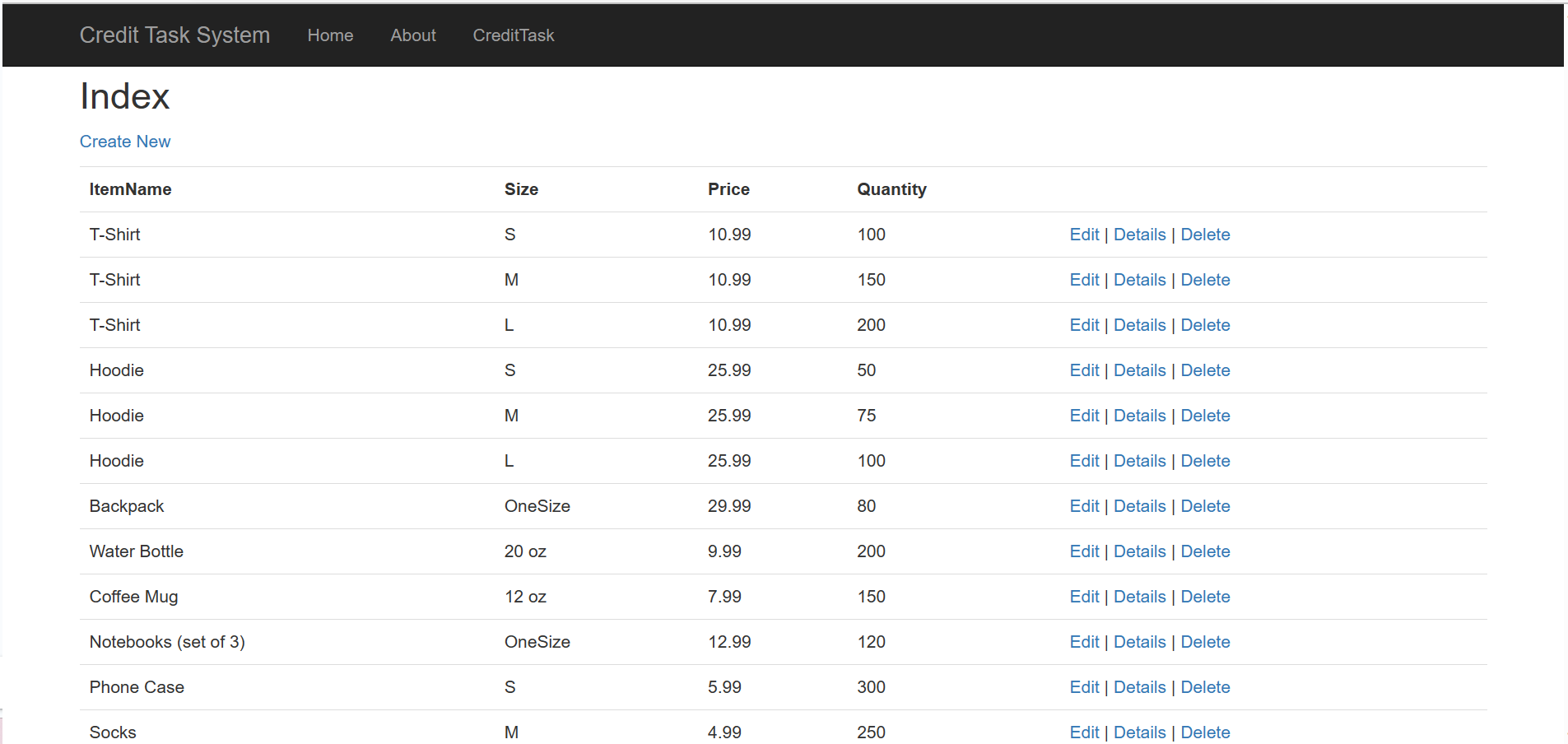


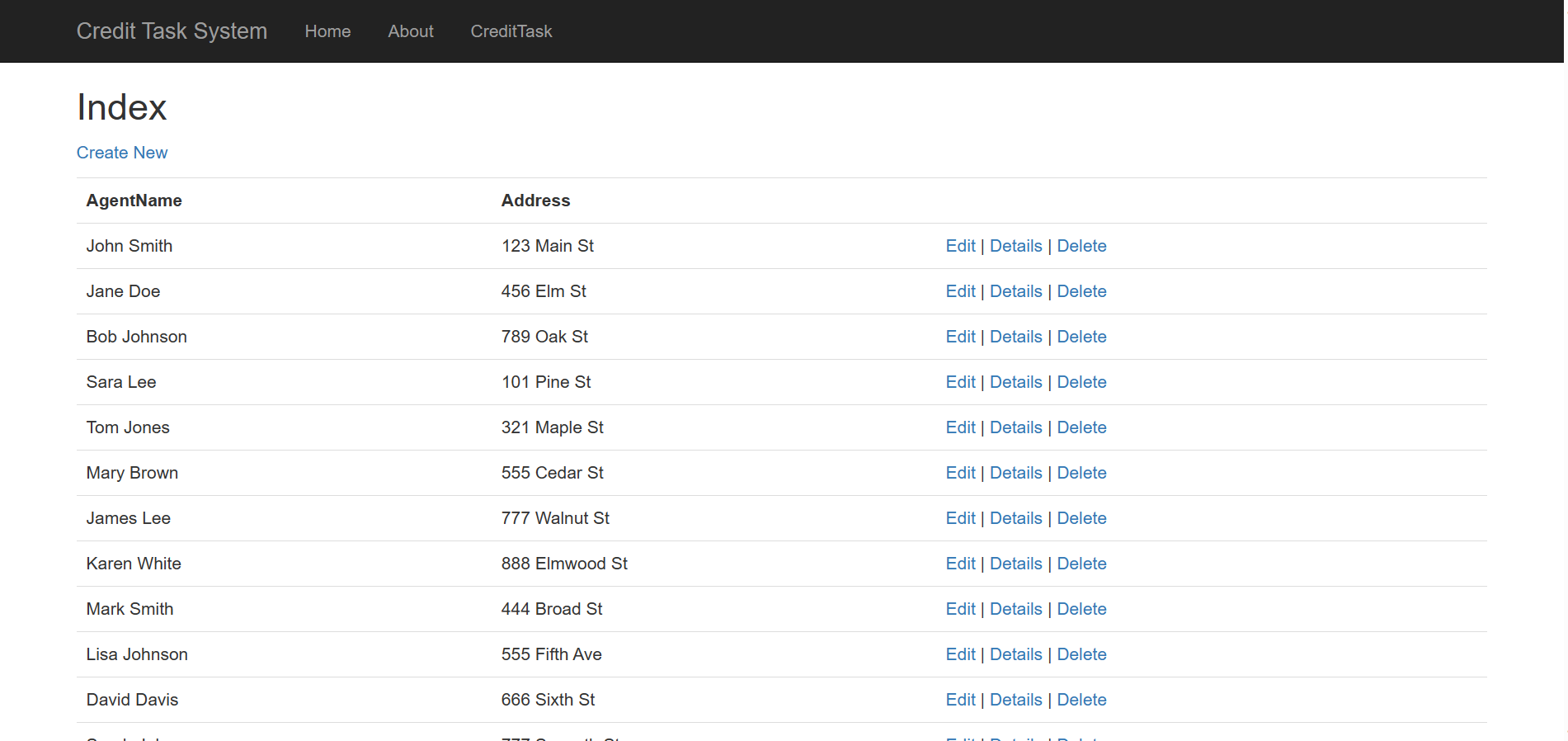


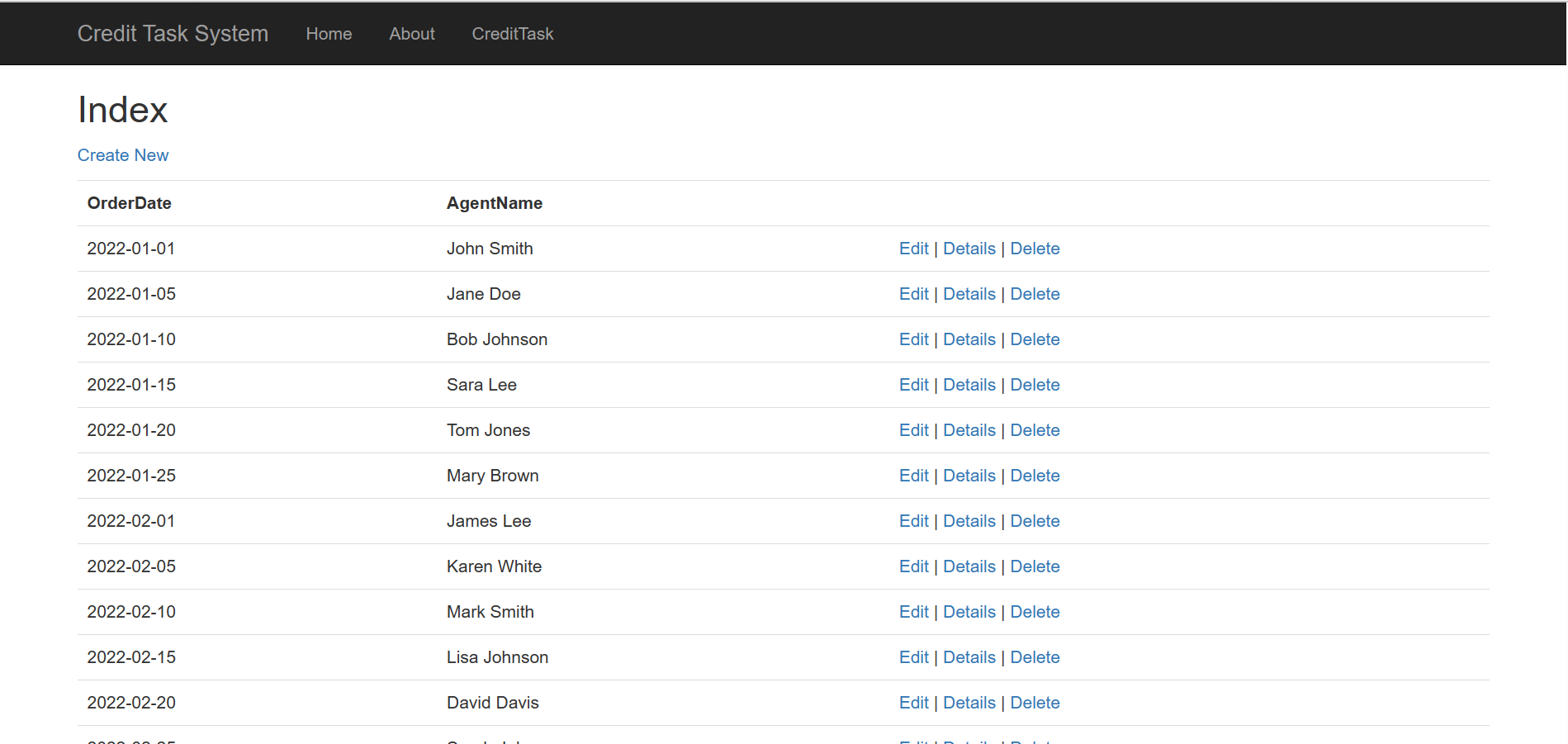


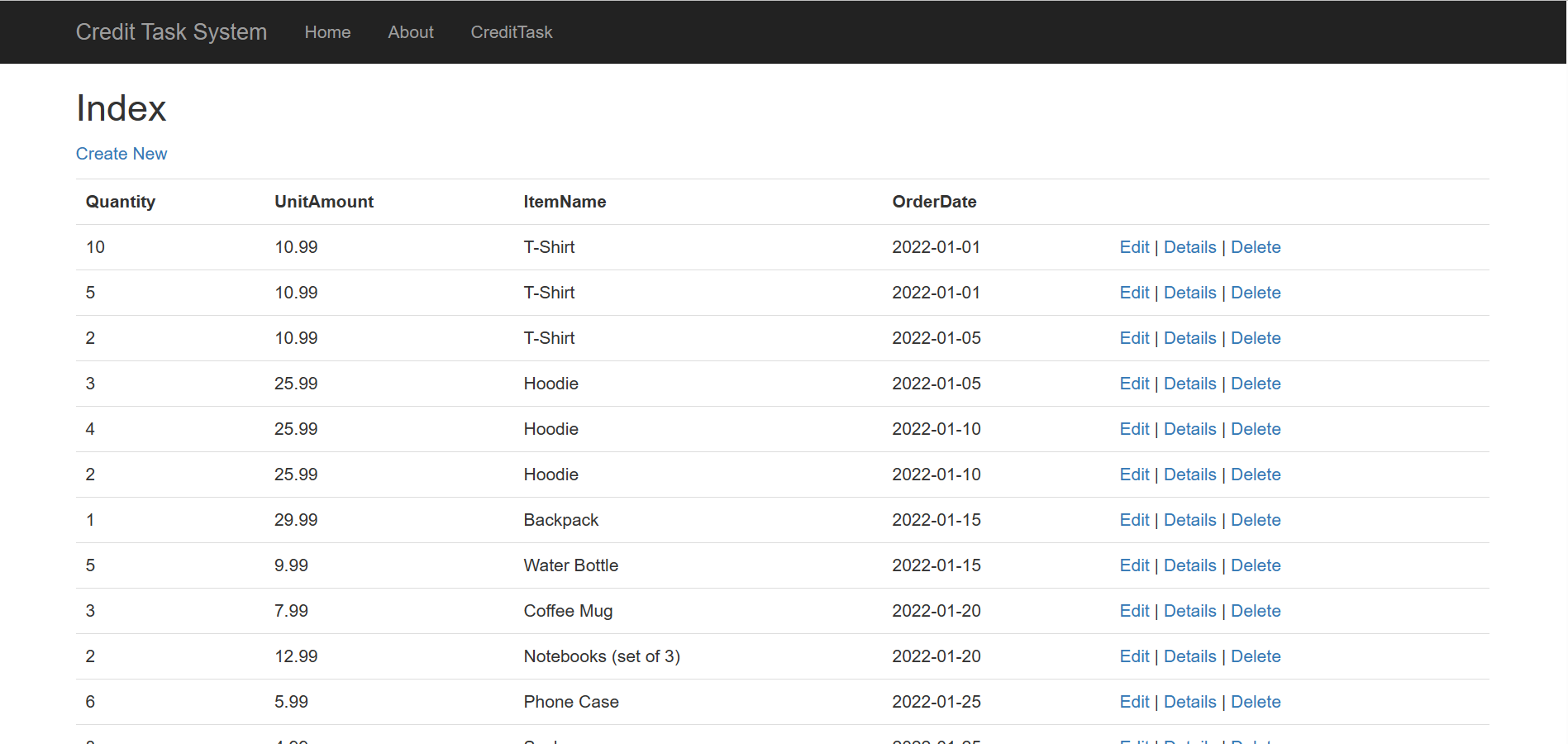
This is our home page, specifically the Credit Task System.

After clicking on the Credit Task on the taskbar, there are 4 pieces of data such as Item List, Agent List, Order List and OrderDetail List.

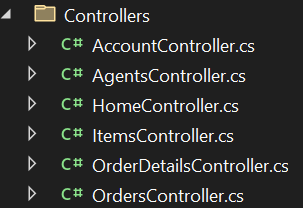


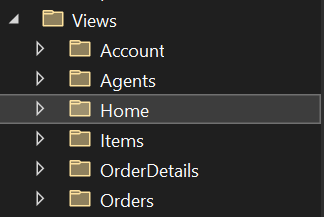




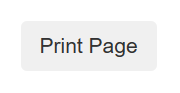


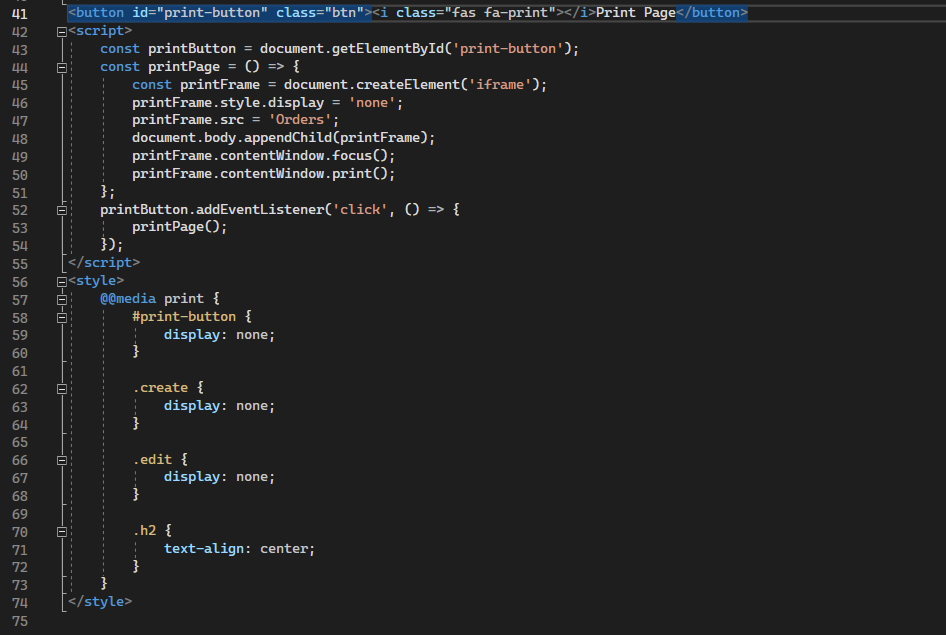
There are four specific data that show on the database.





Controllers and Views for creating these 4 tables of data.

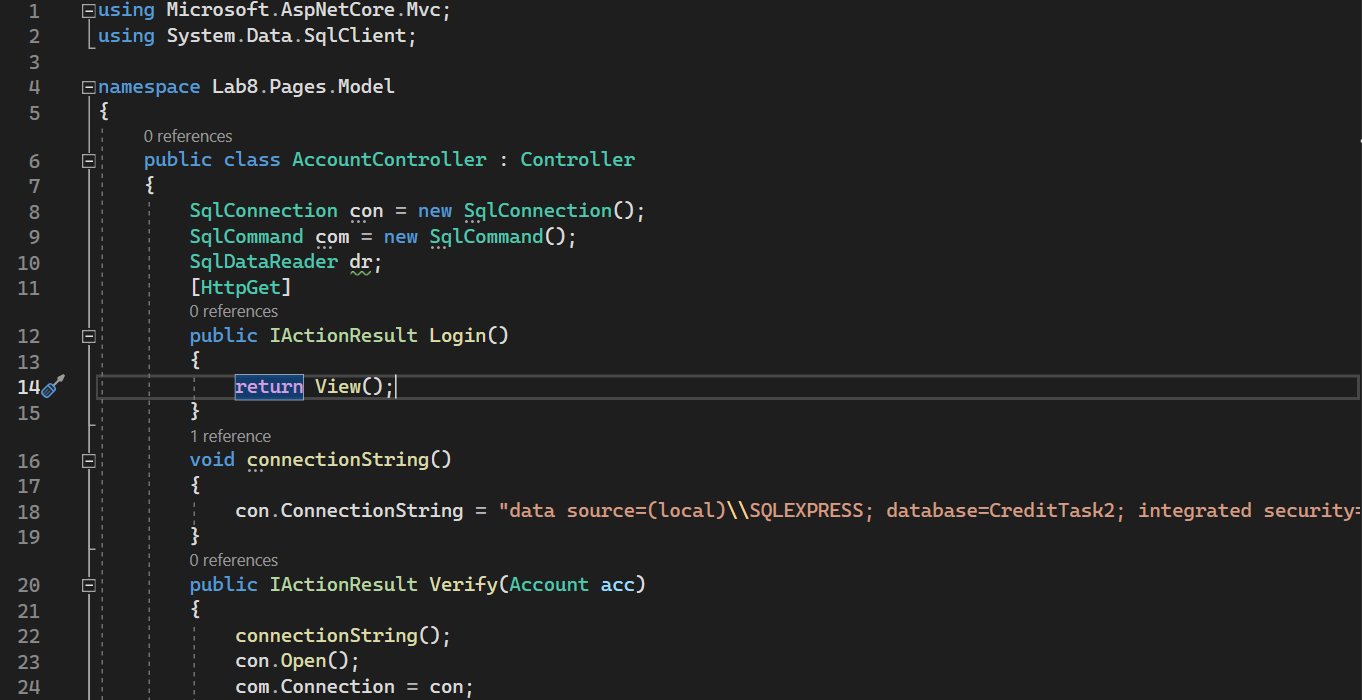




Importantly, in the corner of the Order List page, we create a button for printing information of order.

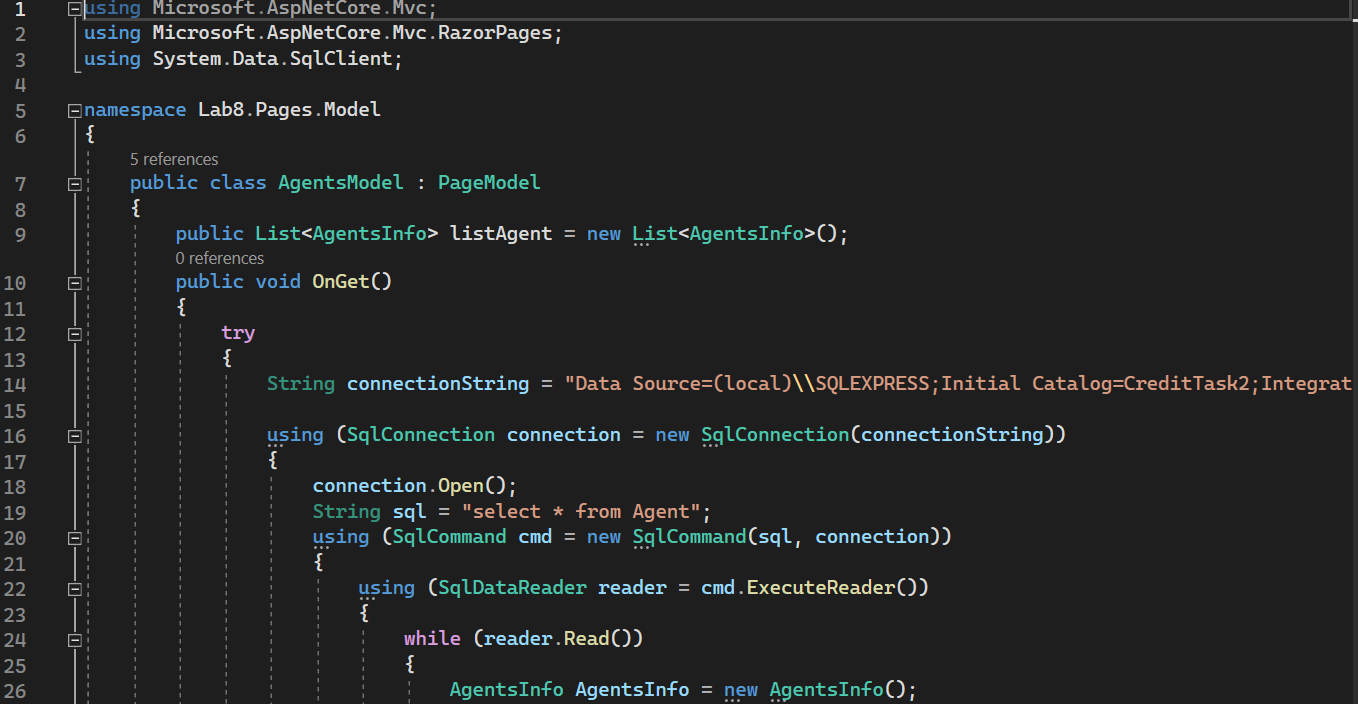
**Exercise 3:**

As the nearly same appearance of exercise 2, we will show our code in ASP.Net Core Web App.



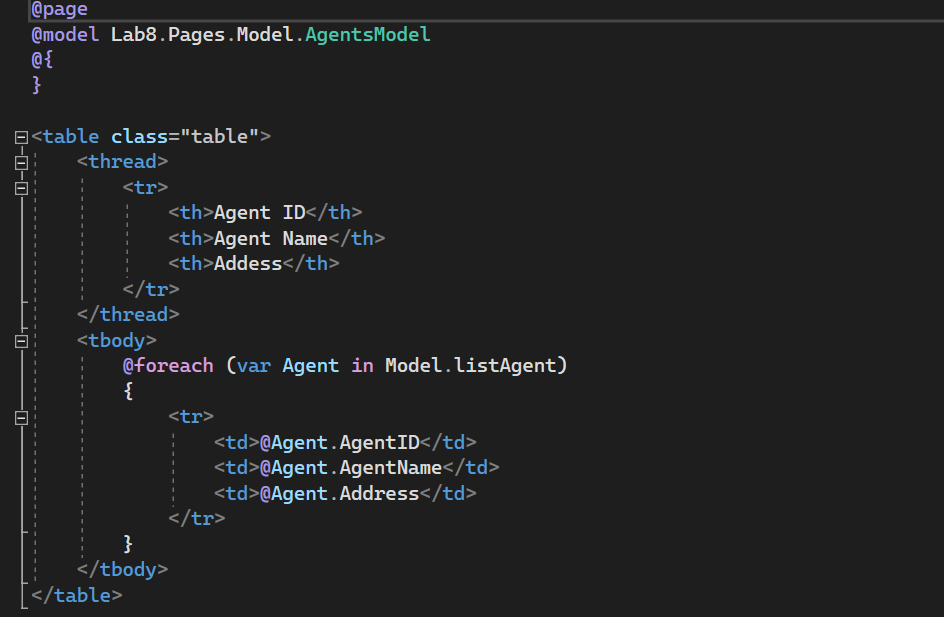


This is code for verifying the login page.

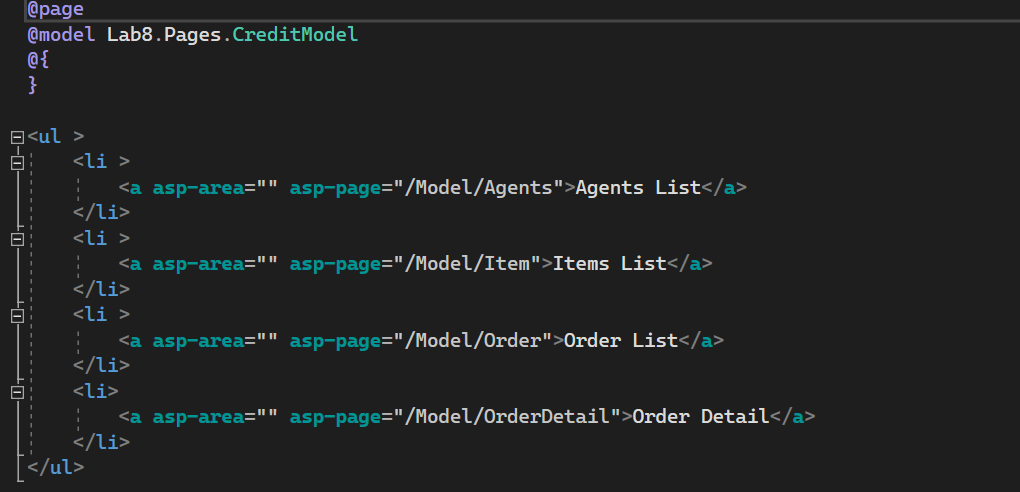




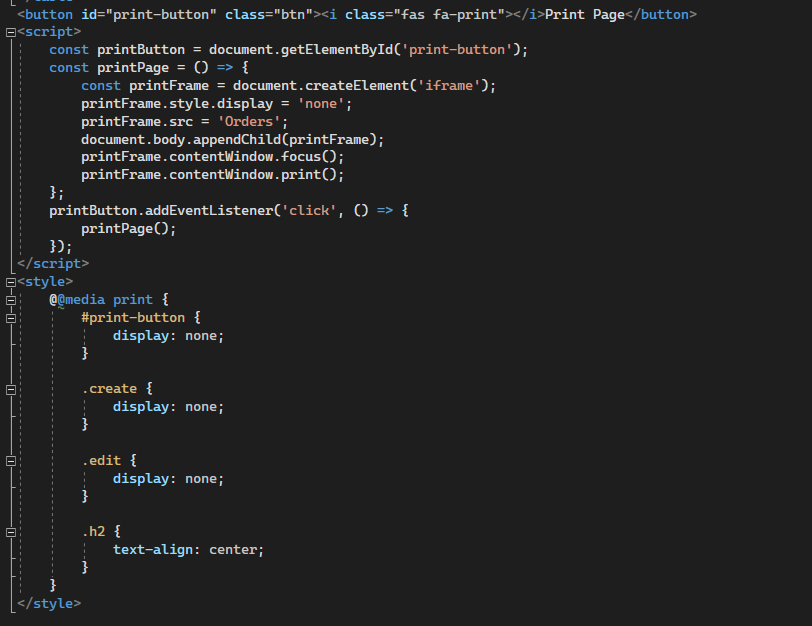
This is code for getting data from Agents List from the database. And it is the same with Item List, Order List and Order Detail List.



This is code for showing the data table of Agent List on the screen. And it is the same with others.



This is code for showing the Credit page to the screen.



Finally, this is the code of the button that prints the data of the Order.