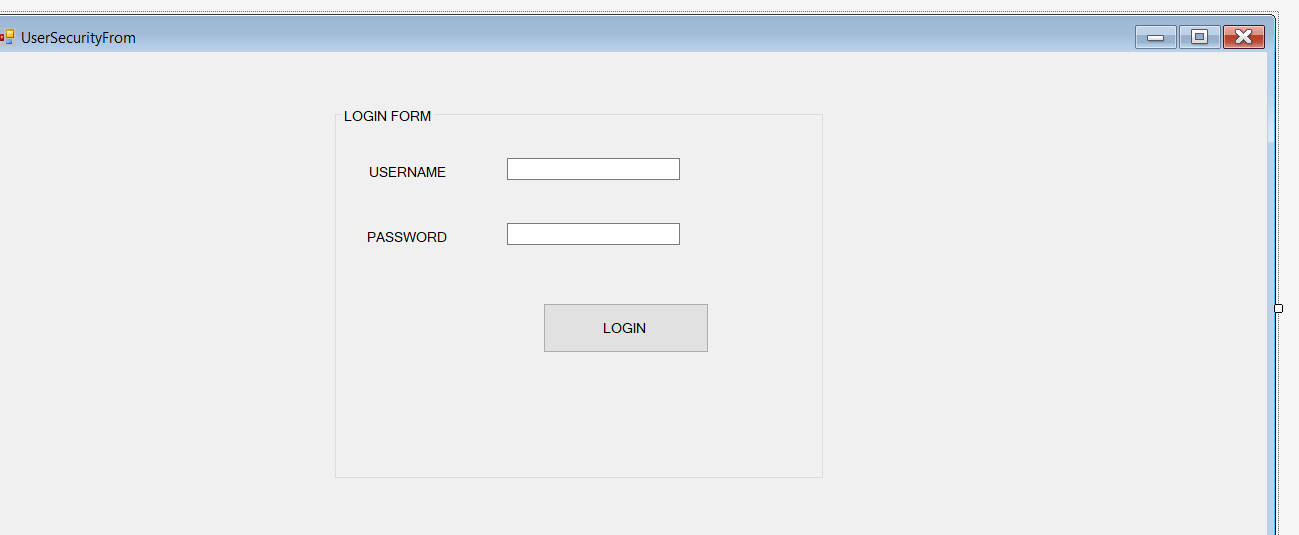
The Hi-tech Order Management Documentation

When the user starts the system the login window is presented, so user can provide the credentials to have access to the system.



using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Hi\_Tech\_Order\_Management\_System.Validation;

using Hi\_Tech\_Order\_Management\_System.BLL;

using Hi\_Tech\_Order\_Management\_System.DAL;

using System.Data.SqlClient;

namespace Hi\_Tech\_Order\_Management\_System.GUI

{

public partial class UserSecurityFrom : Form

{

SqlDataAdapter da;

DataSet dsSecurity;

DataTable dtSecurity;

SqlCommandBuilder sqlBuilder;

public UserSecurityFrom()

{

InitializeComponent();

}

private void button\_login\_Click(object sender, EventArgs e)

{

string id = textBoxLogin.Text.Trim();

string password = textBoxPassword.Text.Trim();

da = new SqlDataAdapter("SELECT \* FROM login", UtilityDB.ConnectDb());

sqlBuilder = new SqlCommandBuilder(da);

da.Fill(dtSecurity);

if (dtSecurity.Rows.Count > 0 )

{

MessageBox.Show("The login is successfull and you have entered into the application", "ok");

MainMenuForm form = new MainMenuForm();

this.Hide();

form.ShowDialog();

}

else

{

MessageBox.Show("You have entered the wrong information");

}

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void UserSecurityFrom\_Load(object sender, EventArgs e)

{

dsSecurity = new DataSet("SecurityDB");

dtSecurity = new DataTable("Security");

dsSecurity.Tables.Add(dtSecurity);

dtSecurity.Columns.Add("UserName", typeof(string));

dtSecurity.Columns.Add("Password", typeof(string));

dtSecurity.PrimaryKey = new DataColumn[] { dtSecurity.Columns["UserName"] };

}

}

}

**Form Main Menu**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Hi\_Tech\_Order\_Management\_System.GUI

{

public partial class MainMenuForm : Form

{

public MainMenuForm()

{

InitializeComponent();

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

private void employeeAndUserToolStripMenuItem\_Click(object sender, EventArgs e)

{

Sales\_Manager cusForm = new Sales\_Manager();

this.Hide();

cusForm.ShowDialog();

}

private void formsToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void booksToolStripMenuItem\_Click(object sender, EventArgs e)

{

Inventory\_Controller bookForm = new Inventory\_Controller();

this.Hide();

bookForm.ShowDialog();

}

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

{

Order\_Clerks orderForm = new Order\_Clerks();

this.Hide();

orderForm.ShowDialog();

}

private void MainMenuForm\_Load(object sender, EventArgs e)

{

}

private void employeeAndUserToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

Form1 empUser = new Form1();

this.Hide();

empUser.ShowDialog();

}

private void customerFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

Sales\_Manager cust = new Sales\_Manager();

this.Hide();

cust.ShowDialog();

}

private void booksFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

Inventory\_Controller books = new Inventory\_Controller();

this.Hide();

books.ShowDialog();

}

private void orderFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

Order\_Clerks order = new Order\_Clerks();

this.Hide();

order.ShowDialog();

}

private void mainMenuToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Really Quit?", "Exit", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

Application.Exit();

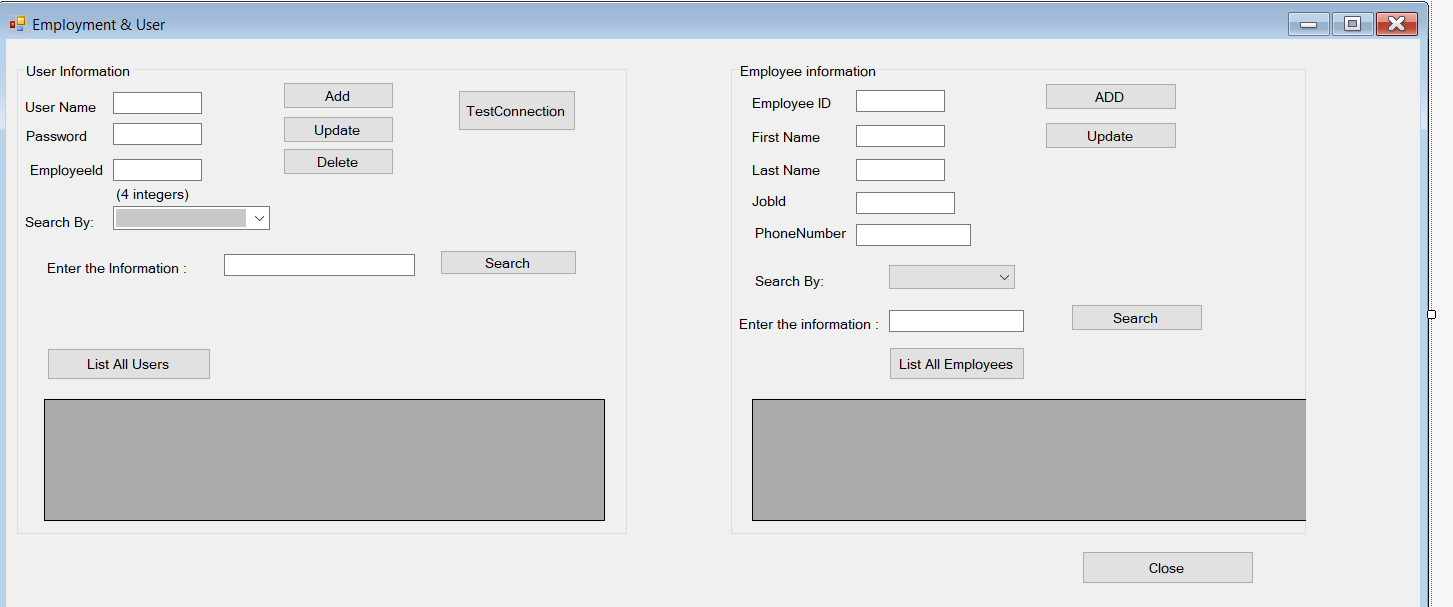
}

}

}

}

**Form MIS\_Manager**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Hi\_Tech\_Order\_Management\_System.BLL;

using Hi\_Tech\_Order\_Management\_System.DAL;

using Hi\_Tech\_Order\_Management\_System.Validation;

using System.Data.SqlClient;

namespace Hi\_Tech\_Order\_Management\_System

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void add\_click(object sender, EventArgs e)

{

string UserId = textBox2.Text.Trim();

if (!Validation.Validation.IsValidName(UserId))

{

MessageBox.Show("Invalid User Name", "Error");

textBox4.Clear();

textBox4.Focus();

return;

}

string Password = textBox1.Text.Trim();

if (!Validation.Validation.IsValidName(Password))

{

MessageBox.Show("Invalid Password", "Error");

textBox5.Clear();

textBox5.Focus();

return;

}

string EmployeeId = textBox3.Text.Trim();

if (!Validation.Validation.IsValidId(EmployeeId,4))

{

MessageBox.Show("Invalid Employee Id", "Error");

textBox4.Clear();

textBox4.Focus();

return;

}

User usr = new User();

usr.UserName= textBox2.Text.Trim();

usr.Password = textBox1.Text.Trim();

usr.EmployeeId = Convert.ToInt32(textBox3.Text.Trim());

usr.SaveUser(usr);

MessageBox.Show("The User have been saved successfully", "Conformation");

}

private void usr\_update(object sender, EventArgs e)

{

User usr = new User();

usr.EmployeeId = Convert.ToInt32(textBox3.Text);

usr.UserName = textBox2.Text;

usr.Password = textBox1.Text;

usr.Update(usr);

textBox1.Clear();

textBox1.Focus();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

MessageBox.Show("This Employee has been updated successfully!", "Confirmation");

}

private void usr\_delete(object sender, EventArgs e)

{

string userId = textBox2.Text.Trim();

if (!Validation.Validation.IsValidId(userId, 4))

{

MessageBox.Show("Invalid user id", "Error");

textBox2.Clear();

textBox2.Focus();

return;

}

User user = new User();

}

private void employee\_information(object sender, EventArgs e)

{

}

private void em\_update(object sender, EventArgs e)

{

Employee emp = new Employee();

emp.EmployeeId = Convert.ToInt32(textBox4.Text);

emp.FirstName = textBox5.Text;

emp.LastName = textBox6.Text;

emp.JobId = Convert.ToInt32(textBox7.Text);

emp.UpdateEmployee(emp);

textBox1.Clear();

textBox1.Focus();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

MessageBox.Show("This Employee has been updated successfully!", "Confirmation");

}

private void emp\_delete(object sender, EventArgs e)

{

}

private void emp\_search(object sender, EventArgs e)

{

string tempId = textBox8.Text.Trim();

if (!Validation.Validation.IsValidId(tempId, 4))

{

MessageBox.Show("Invalid Employee Id", "Error");

textBox8.Clear();

textBox8.Focus();

return;

}

// perform search operation by Employee ID

Employee emp = new Employee();

emp = emp.SearchById(Convert.ToInt32(textBox8.Text.Trim()));

if (emp != null)

{

textBox4.Text = emp.EmployeeId.ToString();

textBox5.Text = emp.FirstName;

textBox6.Text = emp.LastName;

textBox7.Text = emp.JobId.ToString();

textBox9.Text = emp.PhoneNUmber;

}

else

{

MessageBox.Show("Employee not found !", "Invalid Employee ID");

textBox8.Clear();

textBox8.Focus();

}

}

private void button8\_Click(object sender, EventArgs e)

{

string tempId = textBoxInput.Text.Trim();

if (!Validation.Validation.IsValidId(tempId, 4))

{

MessageBox.Show("Invalid Employee Id", "Error");

textBoxInput.Clear();

textBoxInput.Focus();

return;

}

// perform search operation by Employee ID

User usr = new User();

usr = usr.GetUser(Convert.ToInt32(textBoxInput.Text.Trim()));

if (usr != null)

{

textBox2.Text = usr.UserName;

textBox1.Text = usr.Password;

textBox3.Text = usr.EmployeeId.ToString();

}

else

{

MessageBox.Show("Employee not found !", "Invalid Employee ID");

textBox8.Clear();

textBox8.Focus();

}

}

private void button9\_Click(object sender, EventArgs e)

{

SqlConnection connDB = UtilityDB.ConnectDb();

MessageBox.Show(connDB.State.ToString(), "Database Connection");

}

private void button10\_Click(object sender, EventArgs e)

{

Employee emp = new Employee();

List <Employee> listemp = new List<Employee>();

listemp = emp.GetEmployeeList();

dataGridView2.DataSource = listemp;

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

string EmpId = textBox4.Text.Trim();

if (!Validation.Validation.IsValidId(EmpId, 4))

{

MessageBox.Show("Invalid Employee Id", "Error");

textBox4.Clear();

textBox4.Focus();

return;

}

string empFName = textBox5.Text.Trim();

if (!Validation.Validation.IsValidName(empFName))

{

MessageBox.Show("Invalid First Name", "Error");

textBox5.Clear();

textBox5.Focus();

return;

}

string tempLName = textBox6.Text.Trim();

if (!Validation.Validation.IsValidName(tempLName))

{

MessageBox.Show("Invalid Last Name", "Error");

textBox6.Clear();

textBox6.Focus();

return;

}

string tempJob = textBox7.Text.Trim();

if (!Validation.Validation.IsValidId(tempJob,4))

{

MessageBox.Show("Job Cannot be empty", "Error");

textBox7.Clear();

textBox7.Focus();

return;

}

string tempPNumber = textBox9.Text.Trim();

//if (!Validation.Validation.IsValidId(EmpId, 11))

//{

// MessageBox.Show("Invalid Phone Number", "Error");

// textBox9.Clear();

// textBox9.Focus();

// return;

//}

Employee emp = new Employee();

emp.EmployeeId = Convert.ToInt32(textBox4.Text);

emp.FirstName = textBox5.Text.Trim();

emp.LastName = textBox6.Text.Trim();

emp.JobId = Convert.ToInt32(textBox7.Text);

emp.PhoneNUmber = textBox9.Text.Trim();

emp.SaveEmployee(emp);

MessageBox.Show("Information is saved Successfully");

}

private void button11\_Click(object sender, EventArgs e)

{

User usr = new User();

List<User> listusr = new List<User>();

listusr = usr.GetEmployeeList();

dataGridView1.DataSource = listusr;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label11\_Click(object sender, EventArgs e)

{

}

private void button12\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Really Quit?", "Exit", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

Application.Exit();

}

}

private void Form1\_Load(object sender, EventArgs e)

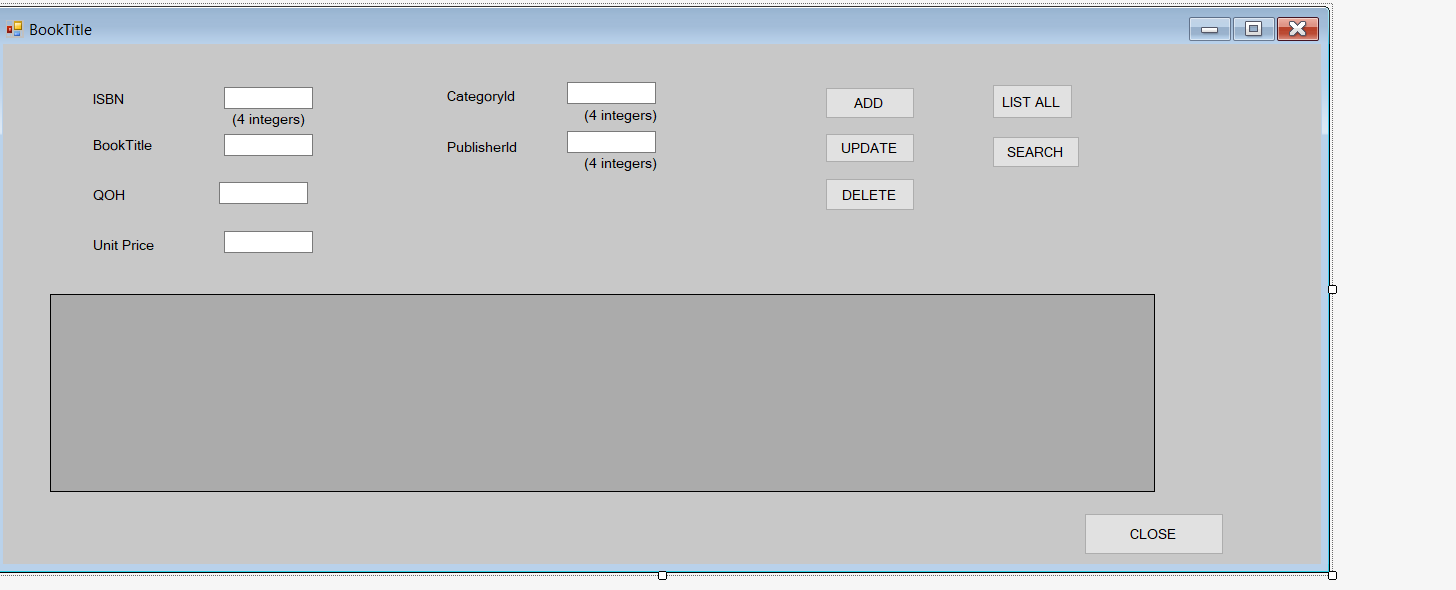
{

}

}

}

**Form Inventory\_Controller**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

using Hi\_Tech\_Order\_Management\_System.BLL;

using Hi\_Tech\_Order\_Management\_System.DAL;

namespace Hi\_Tech\_Order\_Management\_System.GUI

{

public partial class Inventory\_Controller : Form

{

SqlDataAdapter da;

DataSet dsbooks;

DataTable dtbooks;

SqlCommandBuilder sqlBuilder;

Books bs = new Books();

public Inventory\_Controller()

{

InitializeComponent();

}

private void list\_Click(object sender, EventArgs e)

{

//Books books = new Books();

//List<Books> listbooks = new List<Books>();

//listbooks = books.GetList();

//dataGridViewBooks.DataSource = listbooks;

da.Fill(dsbooks.Tables["Books"]);

dataGridViewBooks.DataSource = dsbooks.Tables["Books"];

}

private void search\_click(object sender, EventArgs e)

{

int bookId = Convert.ToInt32(textISBN.Text.Trim());

DataRow dr = dtbooks.Rows.Find(bookId);

if (dr != null)

{

textISBN.Text = dr["ISBN"].ToString();

textBookTitle.Text = dr["BookTitle"].ToString();

textBoxQOH.Text = dr["QOH"].ToString();

textBoxUnitPrice.Text = dr["UnitPrice"].ToString();

textBoxCategory.Text = dr["CategoryId"].ToString();

textBoxPublisher.Text = dr["PublisherId"].ToString();

}

else

{

MessageBox.Show("Book not found!", "Invalid ISBN ID", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void add\_click(object sender, EventArgs e)

{

DataRow dr = dtbooks.NewRow();

dr["ISBN"] = textISBN.Text.Trim();

dr["BookTitle"] = textBookTitle.Text.Trim();

dr["QOH"] = textBoxQOH.Text.Trim();

dr["UnitPrice"] = textBoxUnitPrice.Text.Trim();

dr["CategoryId"] = textBoxCategory.Text.Trim();

dr["PublisherId"] = textBoxPublisher.Text.Trim();

dtbooks.Rows.Add(dr);

MessageBox.Show(dr.RowState.ToString());

textISBN.Text = "";

textBookTitle.Text = "";

textBoxQOH.Text = "";

textBoxUnitPrice.Text = "";

textBoxCategory.Text = "";

textBoxPublisher.Text = "";

}

private void Update\_click(object sender, EventArgs e)

{

DataRow dr = dtbooks.Rows.Find(textISBN.Text.Trim());

dr["BookTitle"] = textBookTitle.Text.Trim();

dr["QOH"] = textBoxQOH.Text.Trim();

dr["UnitPrice"] = textBoxUnitPrice.Text.Trim();

dr["CategoryId"] = textBoxCategory.Text.Trim();

dr["PublisherId"] = textBoxPublisher.Text.Trim();

MessageBox.Show(dr.RowState.ToString());

textISBN.Text = "";

textBookTitle.Text = "";

textBoxQOH.Text = "";

textBoxUnitPrice.Text = "";

textBoxCategory.Text = "";

textBoxPublisher.Text = "";

}

private void delete\_click(object sender, EventArgs e)

{

int isbn = Convert.ToInt32(textISBN.Text.Trim());

DataRow dr = dtbooks.Rows.Find(isbn);

dr.Delete();

MessageBox.Show(dr.RowState.ToString());

textISBN.Text = "";

textBookTitle.Text = "";

textBoxQOH.Text = "";

textBoxUnitPrice.Text = "";

textBoxCategory.Text = "";

textBoxPublisher.Text = "";

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textISBN\_TextChanged(object sender, EventArgs e)

{

}

private void button\_exit\_Click(object sender, EventArgs e)

{

MainMenuForm bks = new MainMenuForm();

bks.Hide();

bks.ShowDialog();

}

private void BooksForm\_Load(object sender, EventArgs e)

{

dsbooks = new DataSet("BooksDS");

dtbooks = new DataTable("Books");

dsbooks.Tables.Add(dtbooks);

dtbooks.Columns.Add("ISBN", typeof(Int32));

dtbooks.Columns.Add("BookTitle", typeof(string));

dtbooks.Columns.Add("QOH", typeof(int));

dtbooks.Columns.Add("UnitPrice", typeof(int));

dtbooks.Columns.Add("CategoryId", typeof(int));

dtbooks.Columns.Add("PublisherId", typeof(int));

dtbooks.PrimaryKey = new DataColumn[] { dtbooks.Columns["ISBN"] };

da = new SqlDataAdapter("SELECT \* FROM Books", UtilityDB.ConnectDb());

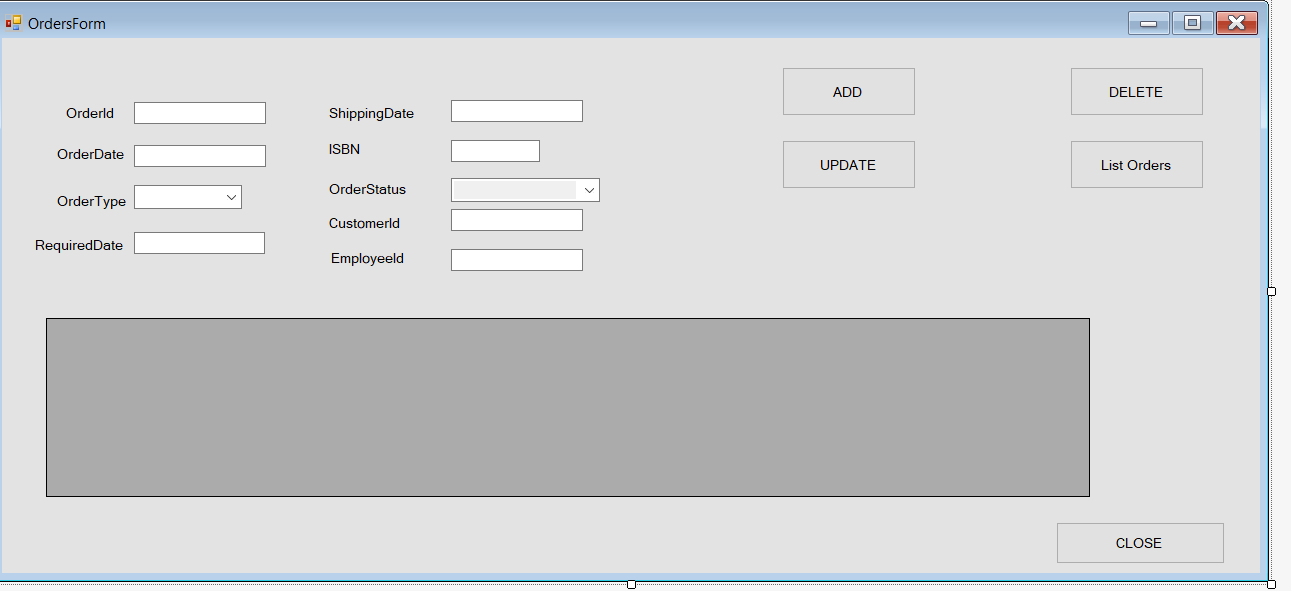
sqlBuilder = new SqlCommandBuilder(da);

}

}

}

**Form Order\_Clerks**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Hi\_Tech\_Order\_Management\_System.BLL;

using Hi\_Tech\_Order\_Management\_System.DAL;

namespace Hi\_Tech\_Order\_Management\_System.GUI

{

public partial class Order\_Clerks : Form

{

FinalProjectMultitieringEntities7 dbEntities = new FinalProjectMultitieringEntities7();

public Order\_Clerks()

{

InitializeComponent();

}

private void buttonAdd\_Click(object sender, EventArgs e)

{

Order odr = new Order();

odr.OrderId = Convert.ToInt32(textBoxOrderId.Text.Trim());

odr.OrderStatus = comboBox1.Text.Trim();

odr.OrderType = comboBox2.Text.Trim();

odr.RequiredDate = textBox1.Text.Trim();

odr.ShippingDate = textBoxShippingDate.Text.Trim();

odr.OrderDate = textBoxOrderDate.Text.Trim();

Employee emp = new Employee();

emp.EmployeeId = Convert.ToInt32(textBoxEmployeeId.Text.Trim());

Customer cst = new Customer();

cst.CustomerId = Convert.ToInt32(textBoxCustomerId.Text.Trim());

dbEntities.Orders.Add(odr);

dbEntities.SaveChanges();

MessageBox.Show("Order have been saved successfully");

textBoxOrderId.Clear();

comboBox1.Text = "";

comboBox2.Text = "";

textBox1.Text = "";

textBoxShippingDate.Text = "";

textBoxOrderDate.Text = "";

// odr.CustomerId = Convert.ToInt32(textBoxCustomerId.Text.Trim());

//odr.

}

private void buttonDelete\_Click(object sender, EventArgs e)

{

Order odr = new Order();

odr = dbEntities.Orders.Find(textBoxOrderId.Text.Trim());

if (odr != null)

{

dbEntities.Orders.Remove(odr);

dbEntities.SaveChanges();

MessageBox.Show("Order Has been deleted successfully", "Confirmation");

}

else

{

MessageBox.Show("OrderId not found!", "Confirmation");

return;

}

}

private void button1\_Click(object sender, EventArgs e)

{

var listOdr = (from odr in dbEntities.Orders

select odr).ToList<Order>();

dataGridView1.DataSource = listOdr;

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

Orders ord = new Orders();

}

private void buttonUpdate\_Click(object sender, EventArgs e)

{

Order odr = new Order();

odr = dbEntities.Orders.Find(Convert.ToInt32(textBoxOrderId.Text.Trim()));

if (odr != null)

{

odr.OrderStatus = comboBox1.Text.Trim();

odr.OrderType = comboBox2.Text.Trim();

odr.RequiredDate = textBox1.Text.Trim();

odr.ShippingDate = textBoxShippingDate.Text.Trim();

odr.OrderStatus = comboBox1.Text.Trim();

}

else

{

MessageBox.Show("Order does not found");

return;

}

}

private void button2\_Click(object sender, EventArgs e)

{

MainMenuForm odr = new MainMenuForm();

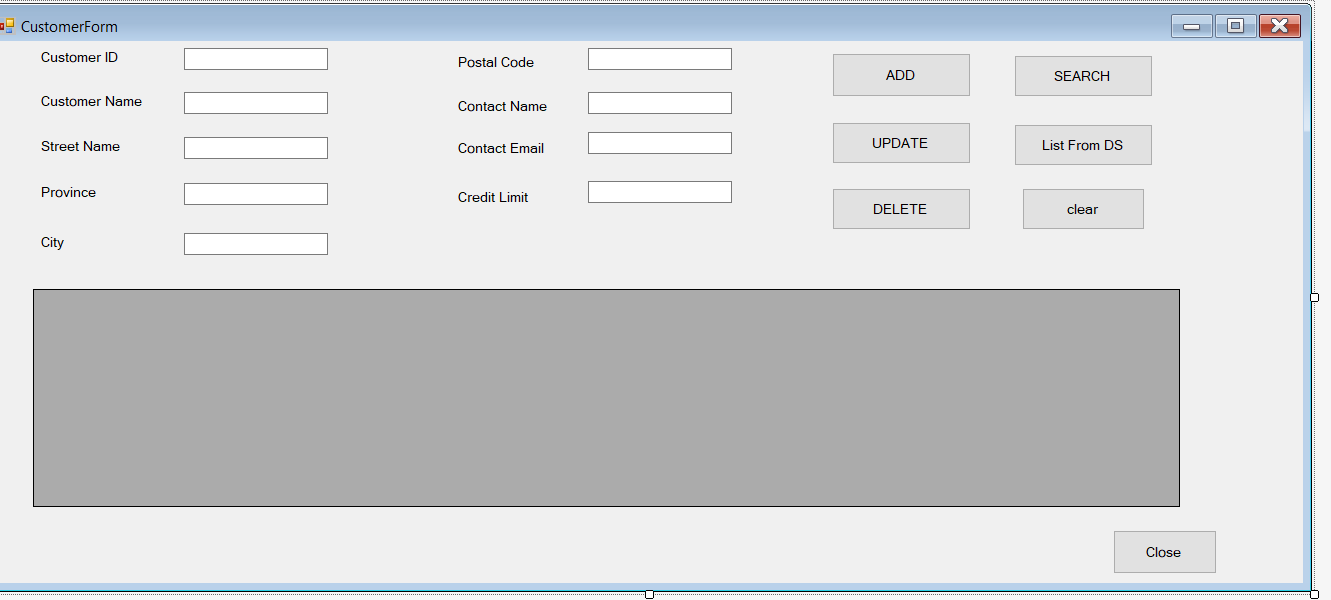
odr.Hide();

odr.ShowDialog();

}

}

**Form Sales Manager**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Hi\_Tech\_Order\_Management\_System.DAL;

using Hi\_Tech\_Order\_Management\_System.BLL;

using Hi\_Tech\_Order\_Management\_System.Validation;

using System.Data.SqlClient;

using System.Xml;

//using Hi\_Tech\_Order\_Management\_System.Validation;

namespace Hi\_Tech\_Order\_Management\_System.GUI

{

public partial class Sales\_Manager : Form

{

SqlDataAdapter da;

DataSet dscustomers;

DataTable dtcustomers;

SqlCommandBuilder sqlBuilder;

Customer cs = new Customer();

public Sales\_Manager()

{

InitializeComponent();

}

private void list\_Click(object sender, EventArgs e)

{

Customer customer = new Customer();

List<Customer> listcst = new List<Customer>();

//listcst = customer.GetEmployeeList();

dataGridViewDataset.DataSource = listcst;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void CustomerForm\_Load(object sender, EventArgs e)

{

dscustomers = new DataSet("CustomerDB");

dtcustomers = new DataTable("Customers");

dscustomers.Tables.Add(dtcustomers);

dtcustomers.Columns.Add("CustomerId", typeof(Int32));

dtcustomers.Columns.Add("CustomerName", typeof(string));

dtcustomers.Columns.Add("StreetName", typeof (string));

dtcustomers.Columns.Add("Province" , typeof(string));

dtcustomers.Columns.Add("City" , typeof(string));

dtcustomers.Columns.Add("PostalCode", typeof(string));

dtcustomers.Columns.Add("ContactName", typeof(string));

dtcustomers.Columns.Add("ContactEmail", typeof(string));

dtcustomers.Columns.Add("CreditLimit", typeof(string));

dtcustomers.PrimaryKey = new DataColumn[] { dtcustomers.Columns["CustomerId"] };

da = new SqlDataAdapter("SELECT \* FROM CUSTOMERS", UtilityDB.ConnectDb());

sqlBuilder = new SqlCommandBuilder(da);

}

private void add\_Click(object sender, EventArgs e)

{

DataRow dr = dtcustomers.NewRow();

dr["CustomerId"] = textBoxcstId.Text.Trim();

dr["CustomerName"] = textBoxcstName.Text.Trim();

dr["StreetName"] = textBoxstrName.Text.Trim();

dr["Province"] = textBoxProvName.Text.Trim();

dr["City"] = textBoxCityName.Text.Trim();

dr["PostalCode"] = textBoxPostalCode.Text.Trim();

dr["ContactName"] = textBoxCtName.Text.Trim();

dr["ContactEmail"] = textBoxEmail.Text.Trim();

dr["CreditLimit"] = textBoxCreditLim.Text.Trim();

dtcustomers.Rows.Add(dr);

MessageBox.Show(dr.RowState.ToString());

textBoxcstId.Text = "";

textBoxcstName.Text = "";

textBoxstrName.Text = "";

textBoxProvName.Text = "";

textBoxCityName.Text = "";

textBoxPostalCode.Text = "";

textBoxCtName.Text = "";

textBoxEmail.Text = "";

textBoxCreditLim.Text = "";

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button6\_Click(object sender, EventArgs e)

{

//da.Fill(dsCollegeDB.Tables["Students"]);

//dataGridViewStudentFromDS.DataSource = dsCollegeDB.Tables["Students"];

da.Fill(dscustomers.Tables["Customers"]);

dataGridViewDataset.DataSource = dscustomers.Tables["Customers"];

}

private void button2\_Click(object sender, EventArgs e)

{

int customerId = Convert.ToInt32(textBoxcstId.Text.Trim());

DataRow dr = dtcustomers.Rows.Find(customerId);

//dr["CustomerName"] = textBoxcstName.Text.Trim();

dr["StreetName"] = textBoxstrName.Text.Trim();

dr["Province"] = textBoxProvName.Text.Trim();

dr["City"] = textBoxCityName.Text.Trim();

dr["PostalCode"] = textBoxPostalCode.Text.Trim();

dr["ContactName"] = textBoxCtName.Text.Trim();

dr["ContactEmail"] = textBoxEmail.Text.Trim();

dr["CreditLimit"] = textBoxCreditLim.Text.Trim();

MessageBox.Show(dr.RowState.ToString());

}

private void button4\_Click(object sender, EventArgs e)

{

int searchId = Convert.ToInt32(textBoxcstId.Text.Trim());

DataRow dr = dtcustomers.Rows.Find(searchId);

if (dr != null)

{

textBoxcstName.Text = dr["CustomerName"].ToString();

textBoxstrName.Text = dr["StreetName"].ToString();

textBoxProvName.Text = dr["Province"].ToString();

textBoxCityName.Text = dr["City"].ToString();

textBoxPostalCode.Text = dr["PostalCode"].ToString();

textBoxCtName.Text = dr["ContactName"].ToString();

textBoxEmail.Text = dr["ContactEmail"].ToString();

textBoxCreditLim.Text = dr["CreditLimit"].ToString();

}

else

{

MessageBox.Show("Student not found!", "Invalid Student ID", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button7\_Click(object sender, EventArgs e)

{

MainMenuForm cus = new MainMenuForm();

this.Hide();

cus.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

int cusId = Convert.ToInt32(textBoxcstId.Text.Trim());

DataRow dr = dtcustomers.Rows.Find(cusId);

dr.Delete();

MessageBox.Show(dr.RowState.ToString());

}

private void button5\_Click(object sender, EventArgs e)

{

textBoxcstId.Text = "";

textBoxcstName.Text = "";

textBoxstrName.Text = "";

textBoxProvName.Text = "";

textBoxCityName.Text = "";

textBoxPostalCode.Text = "";

textBoxCtName.Text = "";

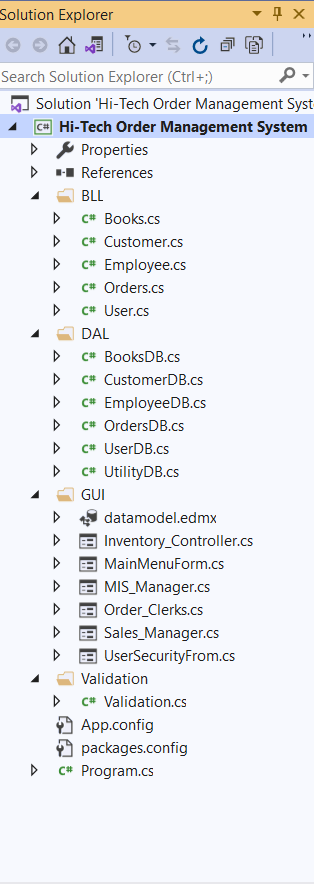
textBoxEmail.Text = "";

textBoxCreditLim.Text = "";

}

}

}



**Script File**

USE [master]

GO

/\*\*\*\*\*\* Object: Database [HiTechDB] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

CREATE DATABASE [HiTechDB]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'HiTechDB', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER2017\MSSQL\DATA\HiTechDB.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'HiTechDB\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER2017\MSSQL\DATA\HiTechDB\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

ALTER DATABASE [HiTechDB] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [HiTechDB].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [HiTechDB] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [HiTechDB] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [HiTechDB] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [HiTechDB] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [HiTechDB] SET ARITHABORT OFF

GO

ALTER DATABASE [HiTechDB] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [HiTechDB] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [HiTechDB] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [HiTechDB] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [HiTechDB] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [HiTechDB] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [HiTechDB] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [HiTechDB] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [HiTechDB] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [HiTechDB] SET DISABLE\_BROKER

GO

ALTER DATABASE [HiTechDB] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [HiTechDB] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [HiTechDB] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [HiTechDB] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [HiTechDB] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [HiTechDB] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [HiTechDB] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [HiTechDB] SET RECOVERY SIMPLE

GO

ALTER DATABASE [HiTechDB] SET MULTI\_USER

GO

ALTER DATABASE [HiTechDB] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [HiTechDB] SET DB\_CHAINING OFF

GO

ALTER DATABASE [HiTechDB] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [HiTechDB] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [HiTechDB] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [HiTechDB] SET QUERY\_STORE = OFF

GO

USE [HiTechDB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Author] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Author](

[AuthorId] [int] NOT NULL,

[FirstName] [nvarchar](max) NOT NULL,

[LastName] [nvarchar](max) NOT NULL,

[Email] [nvarchar](max) NOT NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[AuthorBooks] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AuthorBooks](

[AuthorId] [int] NOT NULL,

[ISBN] [nvarchar](max) NOT NULL,

[YearPublished] [nvarchar](max) NOT NULL,

[Edition] [nchar](10) NOT NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Books] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Books](

[ISBN] [int] IDENTITY(1111,1) NOT NULL,

[BookTitle] [nchar](10) NOT NULL,

[QOH] [int] NOT NULL,

[UnitPrice] [int] NOT NULL,

[CategoryId] [int] NOT NULL,

[PublisherId] [int] NOT NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Categories] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Categories](

[CategoryId] [int] NOT NULL,

[CategoryName] [nvarchar](max) NOT NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Customers] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Customers](

[CustomerId] [int] NOT NULL,

[CustomerName] [nvarchar](50) NOT NULL,

[StreetName] [nchar](50) NOT NULL,

[Province] [nchar](50) NOT NULL,

[City] [nchar](50) NOT NULL,

[PostalCode] [nchar](50) NOT NULL,

[ContactName] [nchar](50) NOT NULL,

[ContactEmail] [nchar](50) NOT NULL,

[CreditLimit] [nchar](50) NOT NULL,

CONSTRAINT [PK\_Customers] PRIMARY KEY CLUSTERED

(

[CustomerId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Employees] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Employees](

[EmployeeId] [int] NOT NULL,

[FirstName] [nchar](10) NOT NULL,

[LastName] [nchar](10) NOT NULL,

[PhoneNumber] [int] NOT NULL,

[JobId] [int] NOT NULL,

CONSTRAINT [PK\_Employees] PRIMARY KEY CLUSTERED

(

[EmployeeId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Jobs] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Jobs](

[JobId] [int] NOT NULL,

[Jobtitle] [nvarchar](max) NOT NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[login] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[login](

[UserName] [nchar](10) NOT NULL,

[Password] [nchar](10) NOT NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[OrderLines] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[OrderLines](

[OrderId] [int] NOT NULL,

[ISBN] [int] NOT NULL,

[QuantityOrdered] [int] NOT NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Orders] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Orders](

[OrderId] [int] NOT NULL,

[OrderDate] [nchar](50) NOT NULL,

[OrderType] [nchar](50) NOT NULL,

[RequiredDate] [nchar](50) NOT NULL,

[ShippingDate] [nchar](50) NOT NULL,

[OrderStatus] [nchar](50) NOT NULL,

CONSTRAINT [PK\_Orders] PRIMARY KEY CLUSTERED

(

[OrderId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[UserAccount] Script Date: 2020-12-15 5:55:56 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[UserAccount](

[UserName] [nchar](10) NULL,

[Password] [nchar](10) NULL,

[EmployeeId] [int] NULL

) ON [PRIMARY]

GO

SET IDENTITY\_INSERT [dbo].[Books] ON

INSERT [dbo].[Books] ([ISBN], [BookTitle], [QOH], [UnitPrice], [CategoryId], [PublisherId]) VALUES (1111, N'PaniPat ', 2, 23, 12, 21)

INSERT [dbo].[Books] ([ISBN], [BookTitle], [QOH], [UnitPrice], [CategoryId], [PublisherId]) VALUES (1212, N'Ishwae ', 1221, 1212, 1212, 1212)

INSERT [dbo].[Books] ([ISBN], [BookTitle], [QOH], [UnitPrice], [CategoryId], [PublisherId]) VALUES (1213, N'jiming ', 22, 2000, 2345, 2345)

SET IDENTITY\_INSERT [dbo].[Books] OFF

GO

INSERT [dbo].[Customers] ([CustomerId], [CustomerName], [StreetName], [Province], [City], [PostalCode], [ContactName], [ContactEmail], [CreditLimit]) VALUES (1111, N'Puran', N'Lincoln ', N'Quebec ', N'Montreal ', N'H3H1H9 ', N'Ishu ', N'Ishu@123 ', N'null ')

GO

INSERT [dbo].[Employees] ([EmployeeId], [FirstName], [LastName], [PhoneNumber], [JobId]) VALUES (0, N'Puneet ', N'Kaur ', 222222, 1122)

INSERT [dbo].[Employees] ([EmployeeId], [FirstName], [LastName], [PhoneNumber], [JobId]) VALUES (1111, N'Ishwarjot ', N'Singh ', 2212121, 1212)

INSERT [dbo].[Employees] ([EmployeeId], [FirstName], [LastName], [PhoneNumber], [JobId]) VALUES (1212, N'Ishwarjot ', N'Singh ', 123456789, 1234)

INSERT [dbo].[Employees] ([EmployeeId], [FirstName], [LastName], [PhoneNumber], [JobId]) VALUES (1234, N'Ishwae ', N'Singh ', 12333333, 1111)

GO

INSERT [dbo].[login] ([UserName], [Password]) VALUES (N'Ishwarjot ', N'singh ')

GO

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (1212, N'24/9/2020 ', N'Light ', N'22/6/2020 ', N'12/6/2018 ', N'Delivering ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (1234, N'23/nov/2020 ', N'Light ', N'25/nov/2020 ', N'27/nov/2020 ', N'Pending ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (2222, N'20/10/2020 ', N'ni ', N'20/10/2020 ', N'20/10/2020 ', N'nu ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (5555, N'30/6/2020 ', N'heavy ', N'25/10/2020 ', N'28/10/2020 ', N'shipped ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (6666, N'20/6/2020 ', N'light ', N'22/10/2020 ', N'21/10/2020 ', N'packed ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (7777, N'12/5/2020 ', N'medium ', N'25/6/2020 ', N'26/6/2020 ', N'delivired delivered ')

INSERT [dbo].[Orders] ([OrderId], [OrderDate], [OrderType], [RequiredDate], [ShippingDate], [OrderStatus]) VALUES (8888, N'23/1/2020 ', N'light ', N'25/6/2020 ', N'27/6/2020 ', N' delivered ')

GO

INSERT [dbo].[UserAccount] ([UserName], [Password], [EmployeeId]) VALUES (N'Ishwa ', N'paoa ', 1111)

INSERT [dbo].[UserAccount] ([UserName], [Password], [EmployeeId]) VALUES (N'Ishawww ', N'ususu ', 1111)

INSERT [dbo].[UserAccount] ([UserName], [Password], [EmployeeId]) VALUES (N'shsh ', N'shsh ', 2222)

GO

USE [master]

GO

ALTER DATABASE [HiTechDB] SET READ\_WRITE

GO

**Conclusion**

The project helped me to go through advance C sharp coding which is really in scope now a days. This was fun doing the project and understanding the whole concept about connected mode with C# , disconnected mode and EntityFrameWork.