**WALMART SUPERCENTER**

**(DESKTOP APPLICATION)**



Session: 2021 – 2025

**Submitted by:**

Muhammad Danish 2021-CS-167

**Supervised by:**

Miss Maida Shahid

Department of Computer Science

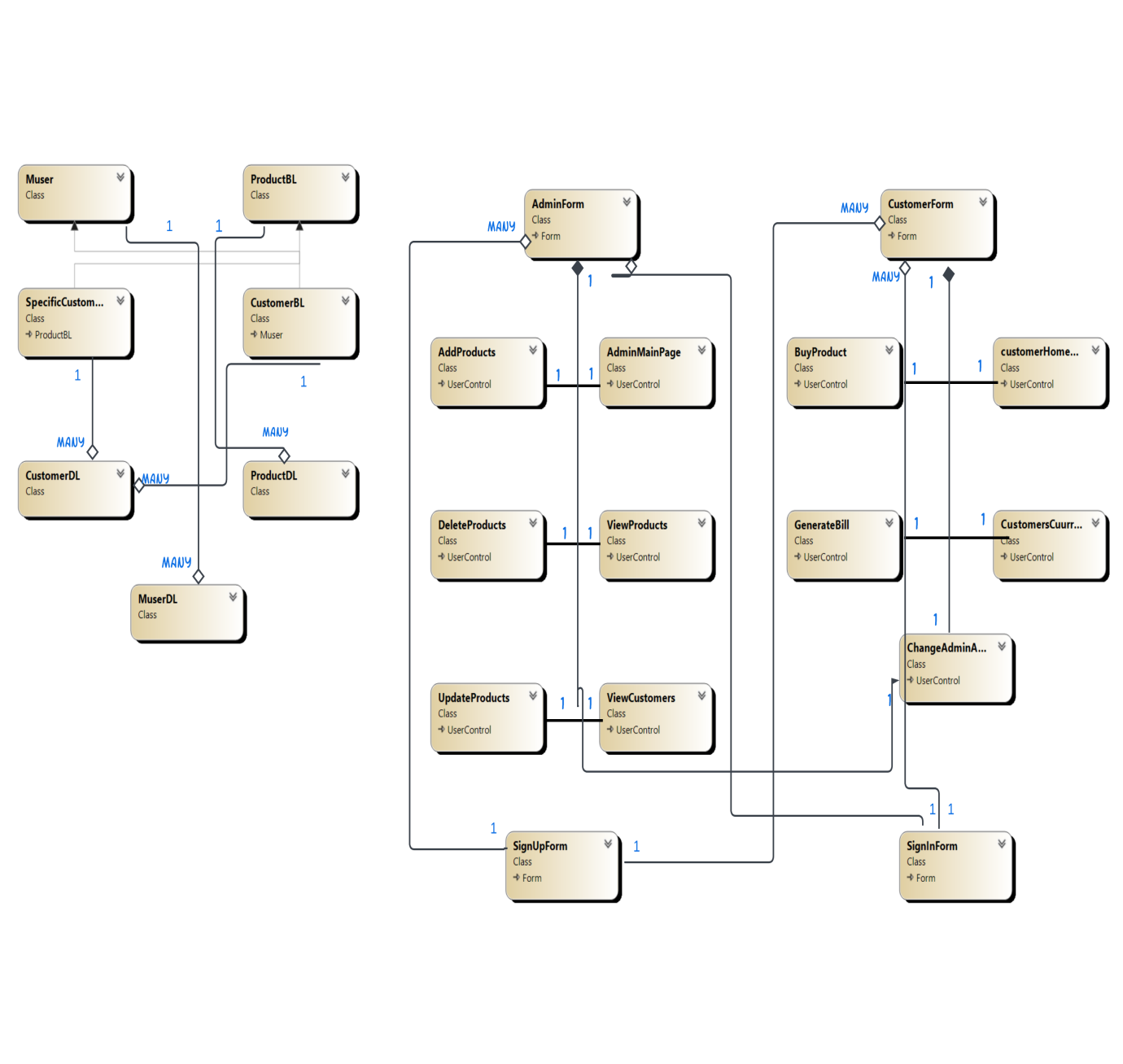
**University of Engineering and Technology**

**Lahore Pakistan**

**Contents**

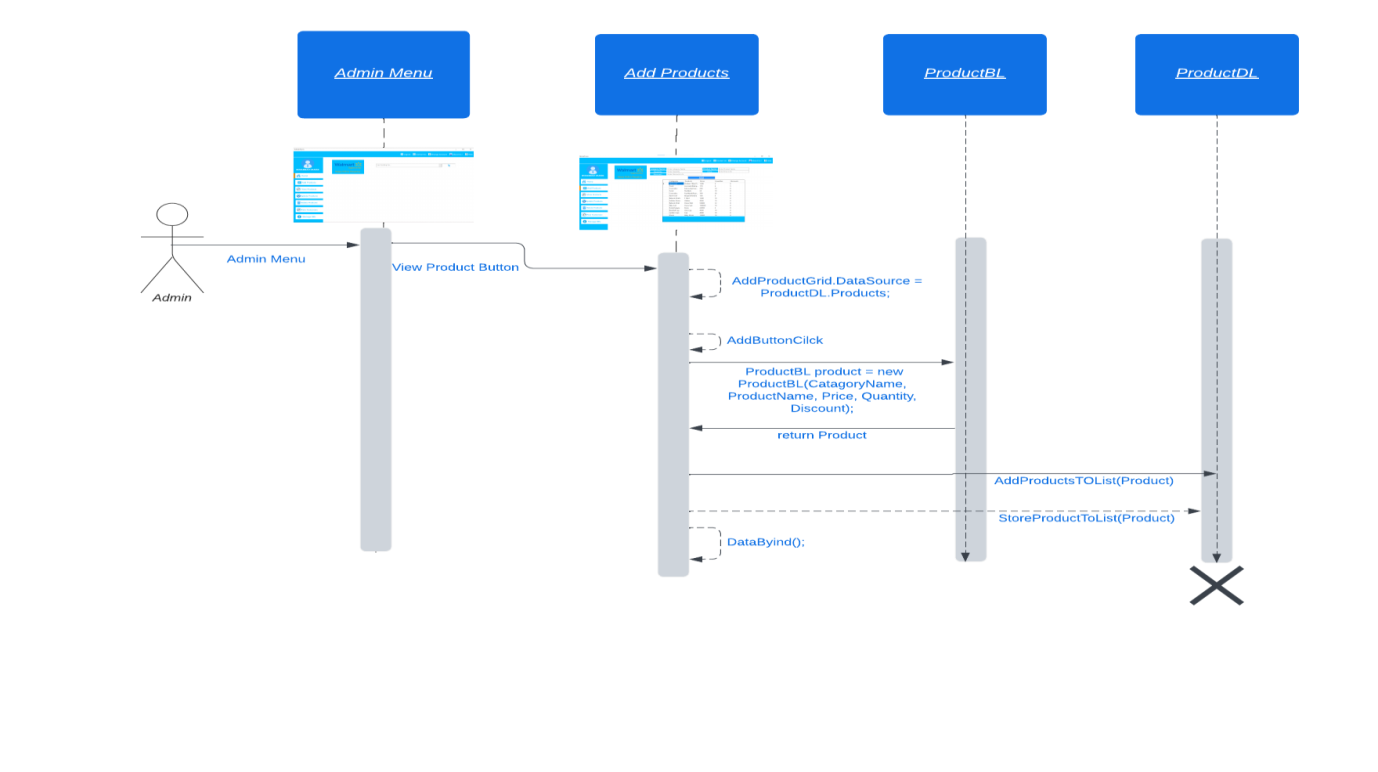
* [**Class Diagram** 3](#_Toc104986801)
* [**Sequence Diagram** 4](#_Toc104986802)
* [**Wire Frames** 7](#_Toc104986804)
* [**Full Code** 15](#_Toc104986805)

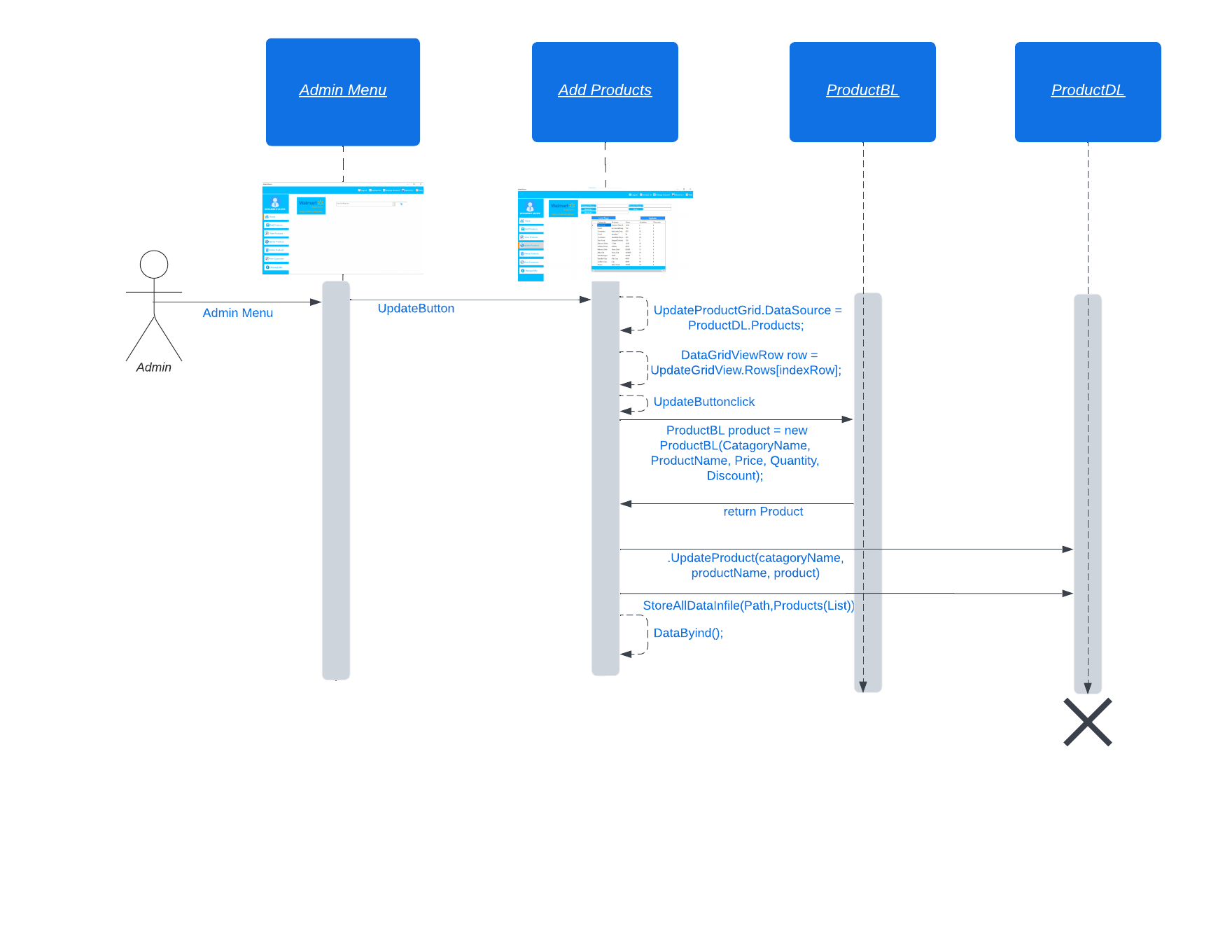
# **Class Diagram**



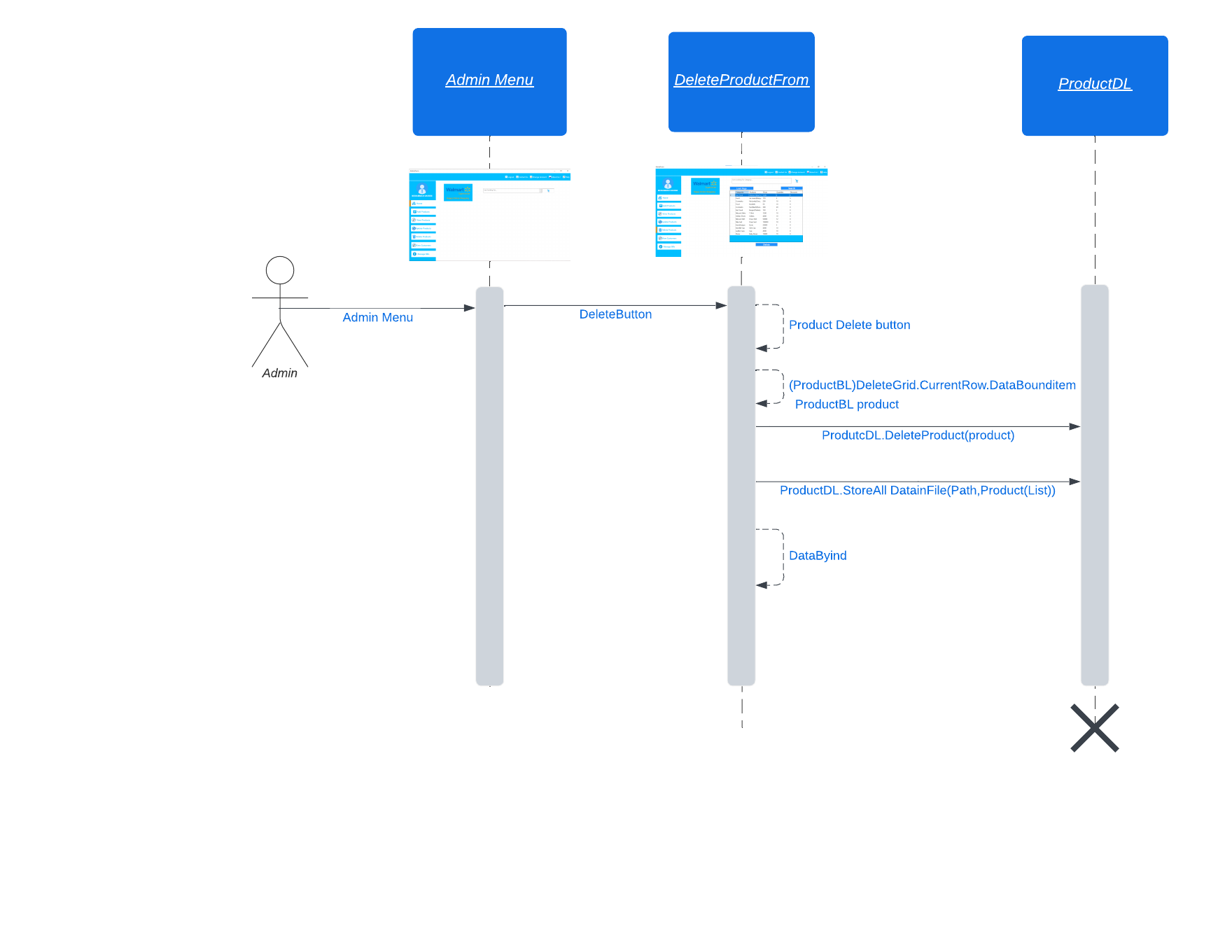
# **Sequence Diagram**

# **Add Products**

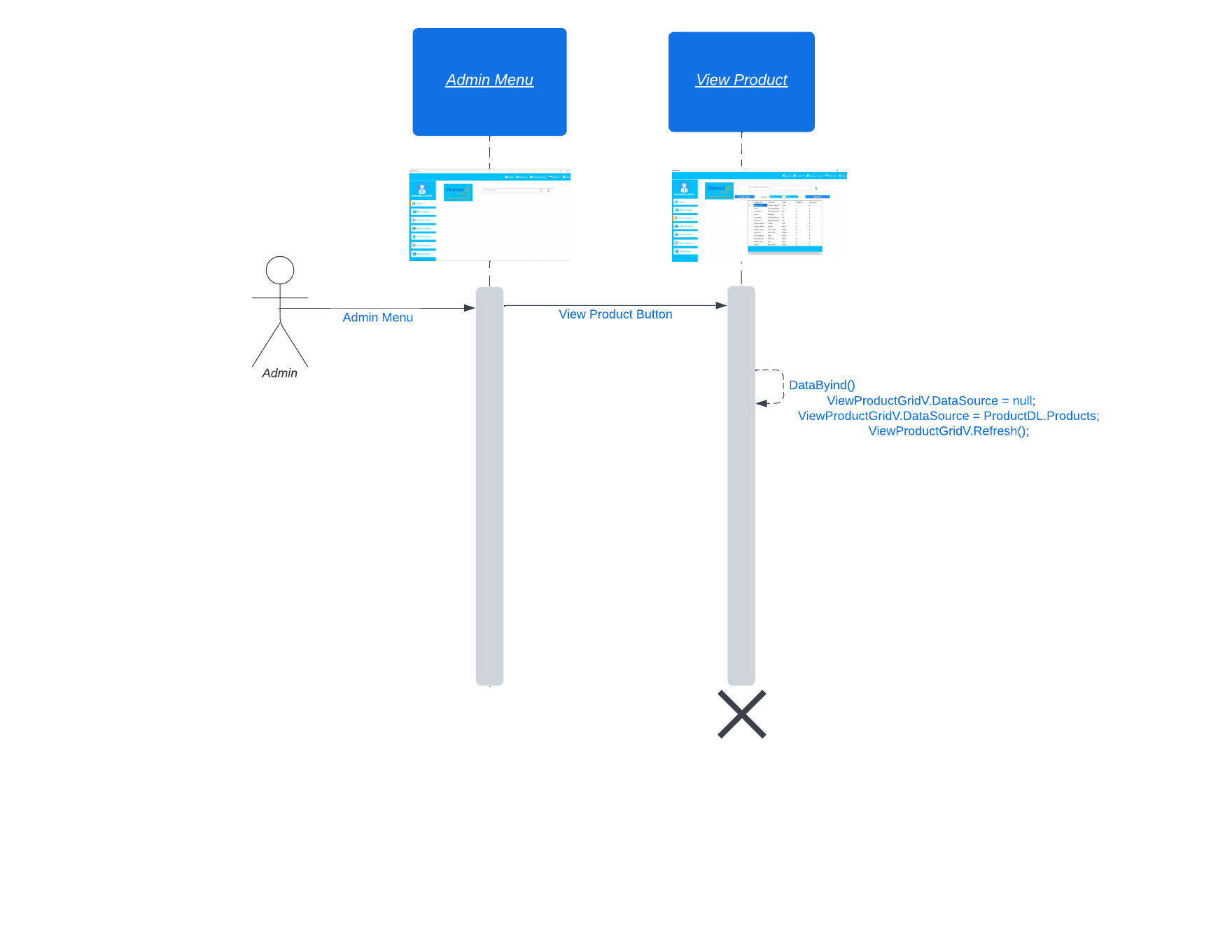
**Update Products**

****

**Delete Product**

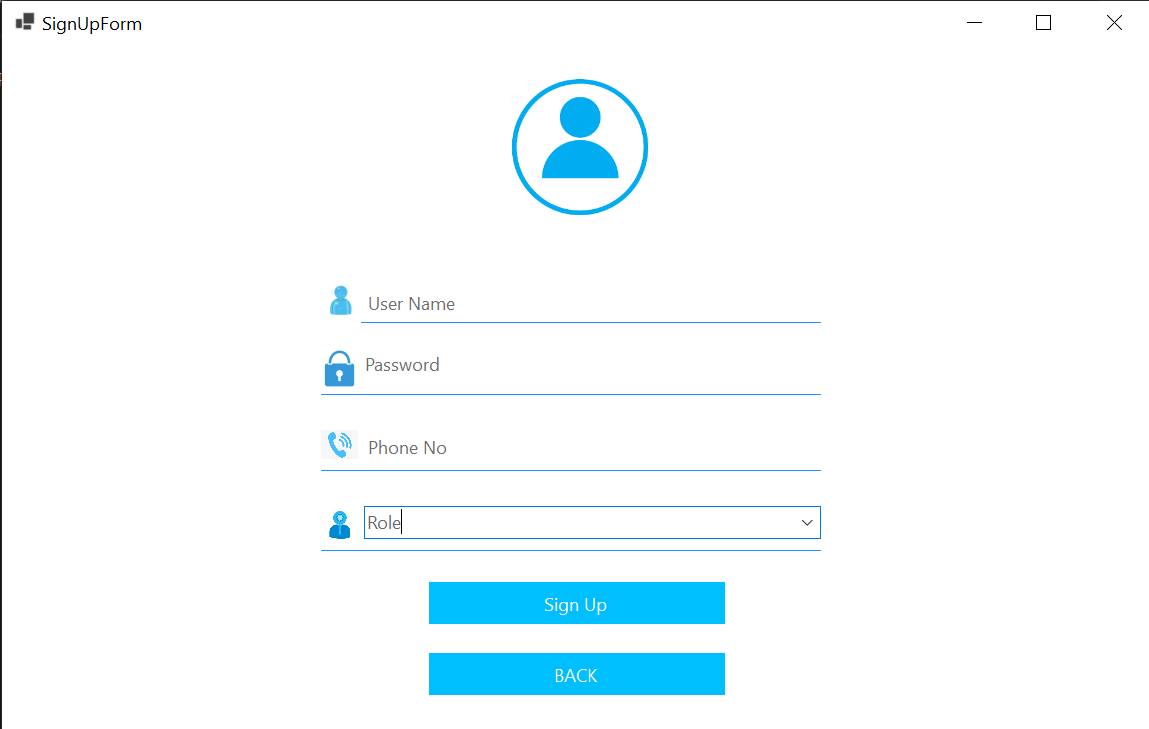
****

**Retrieve Products**

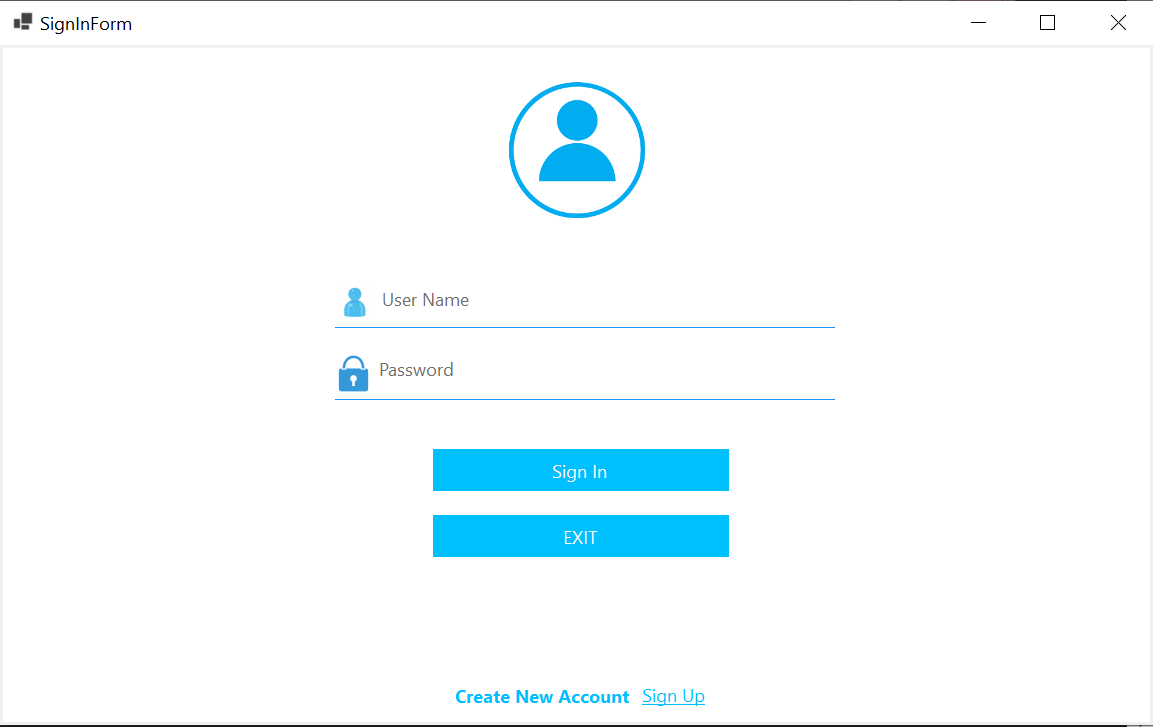
****

# **Wire Frames**

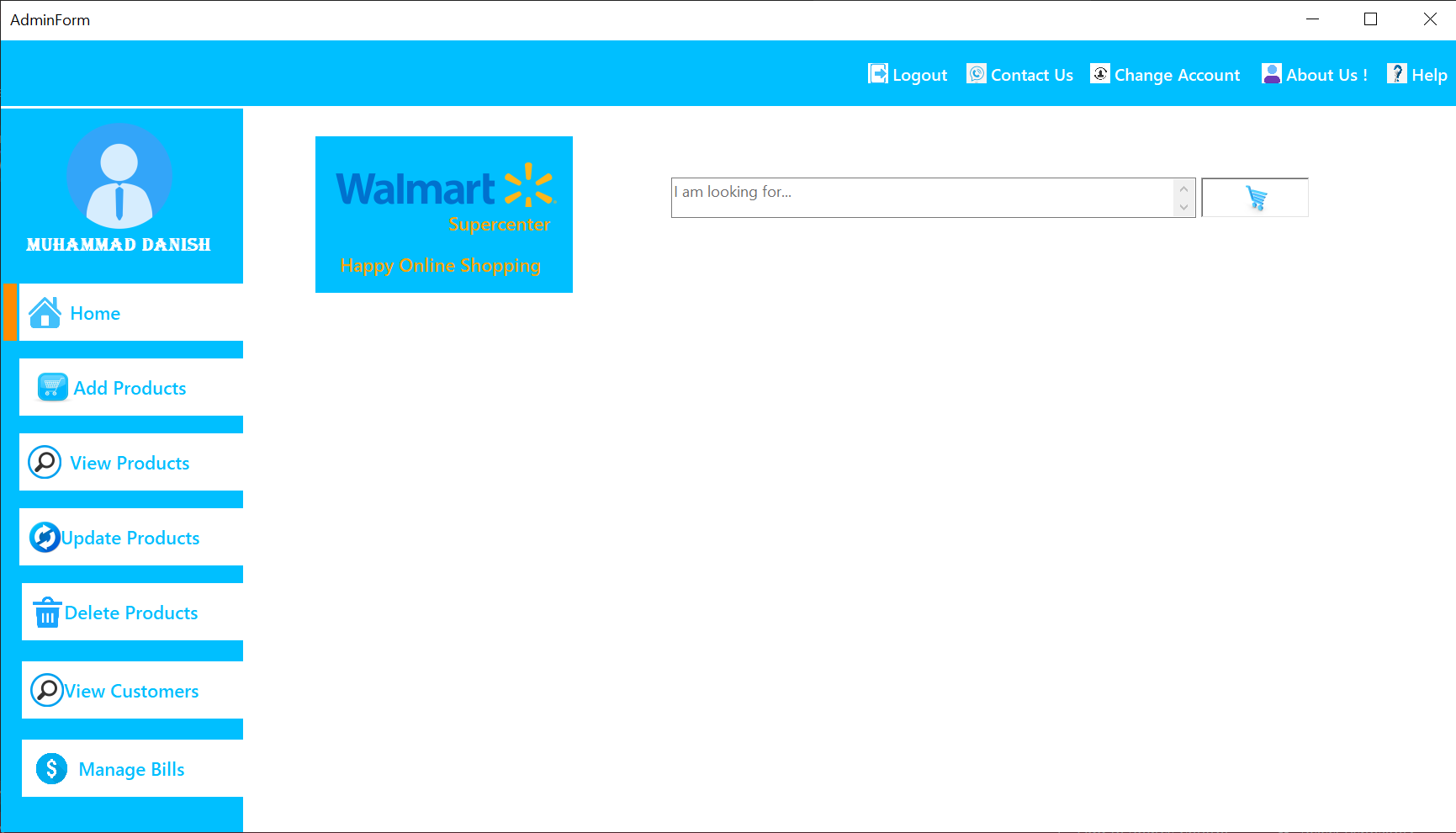
**Sign Up**

****

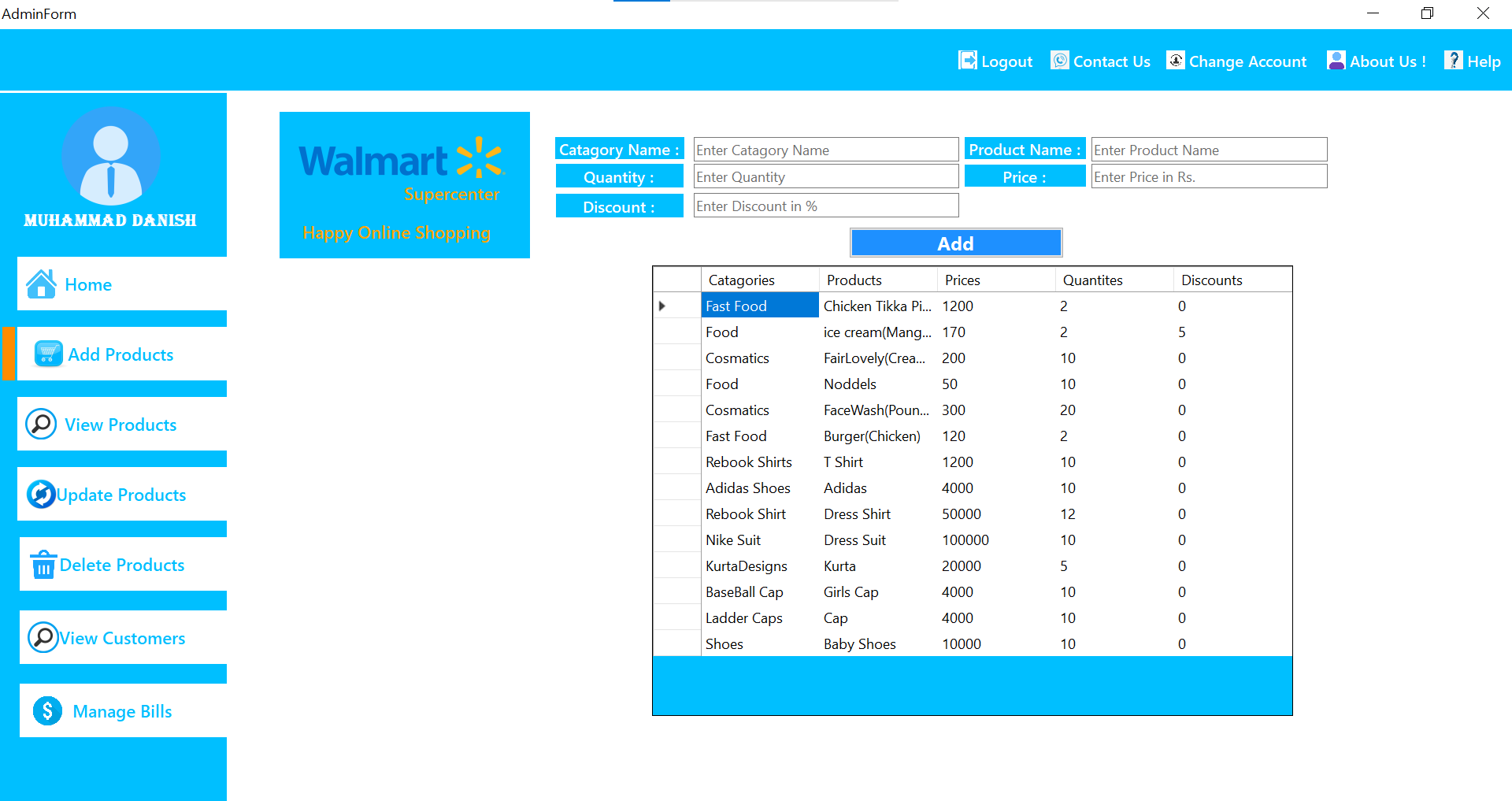
**Sign In**

****

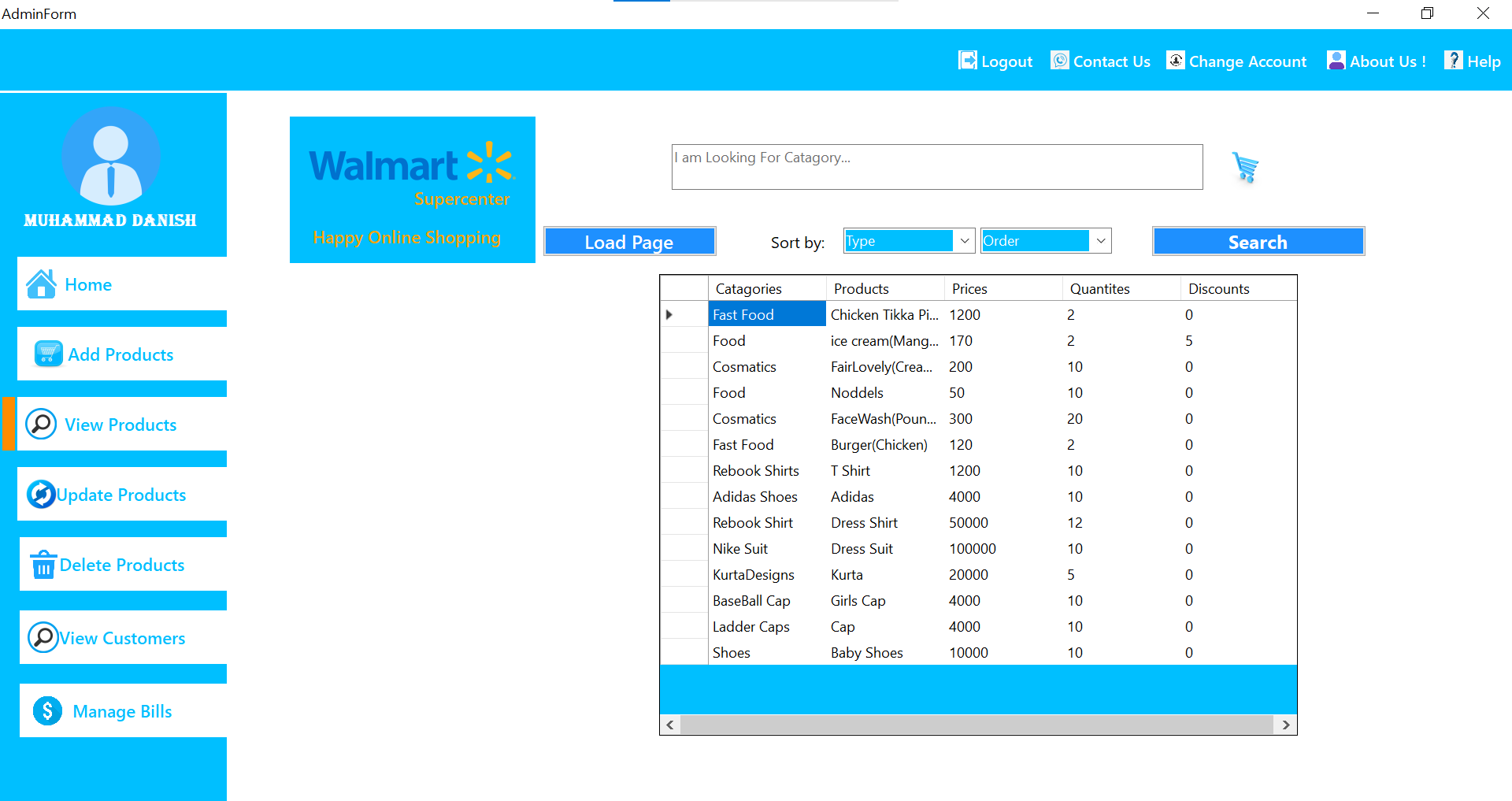
**Admin Menu**

****

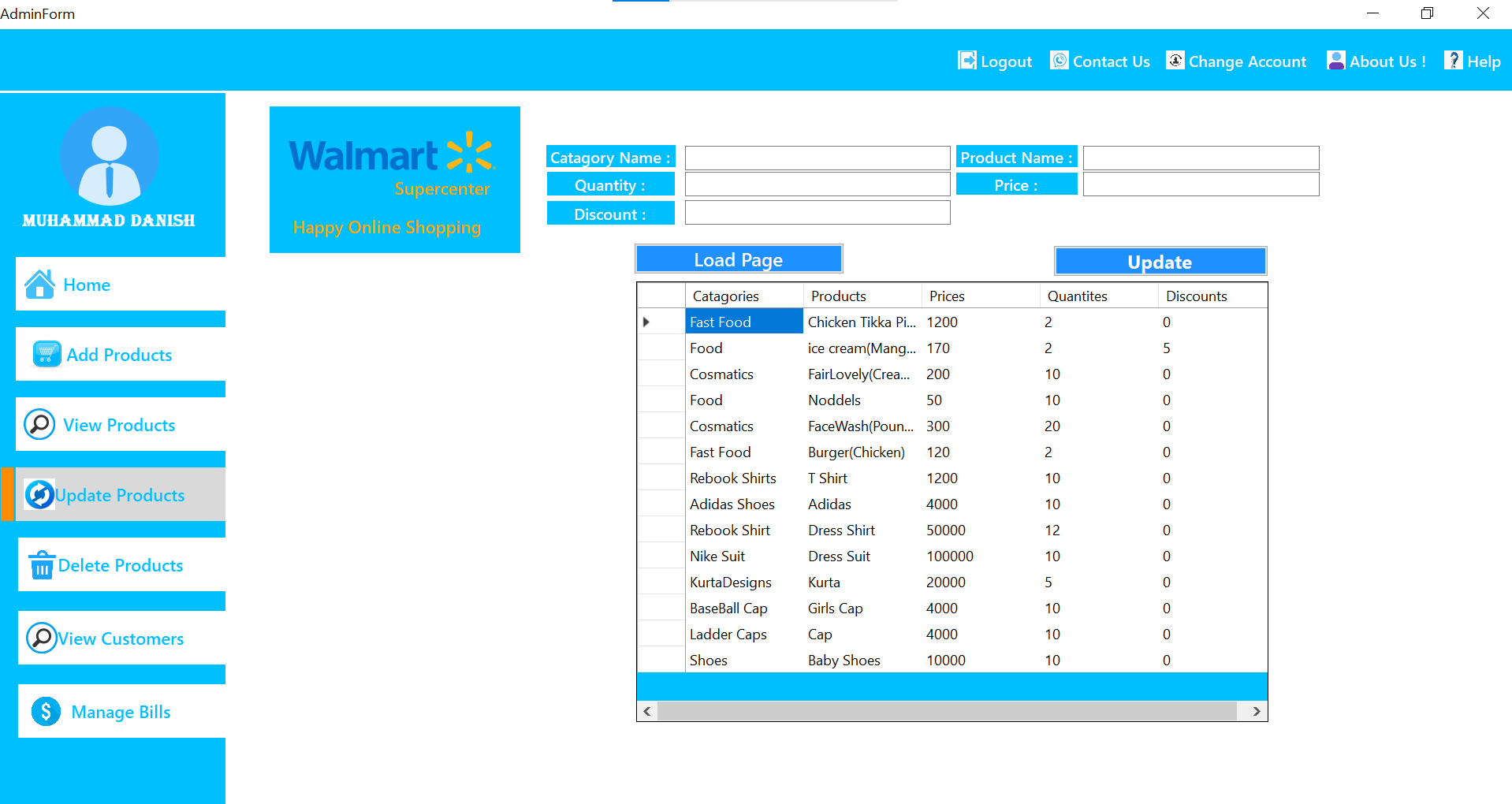
* **Add Products**

****

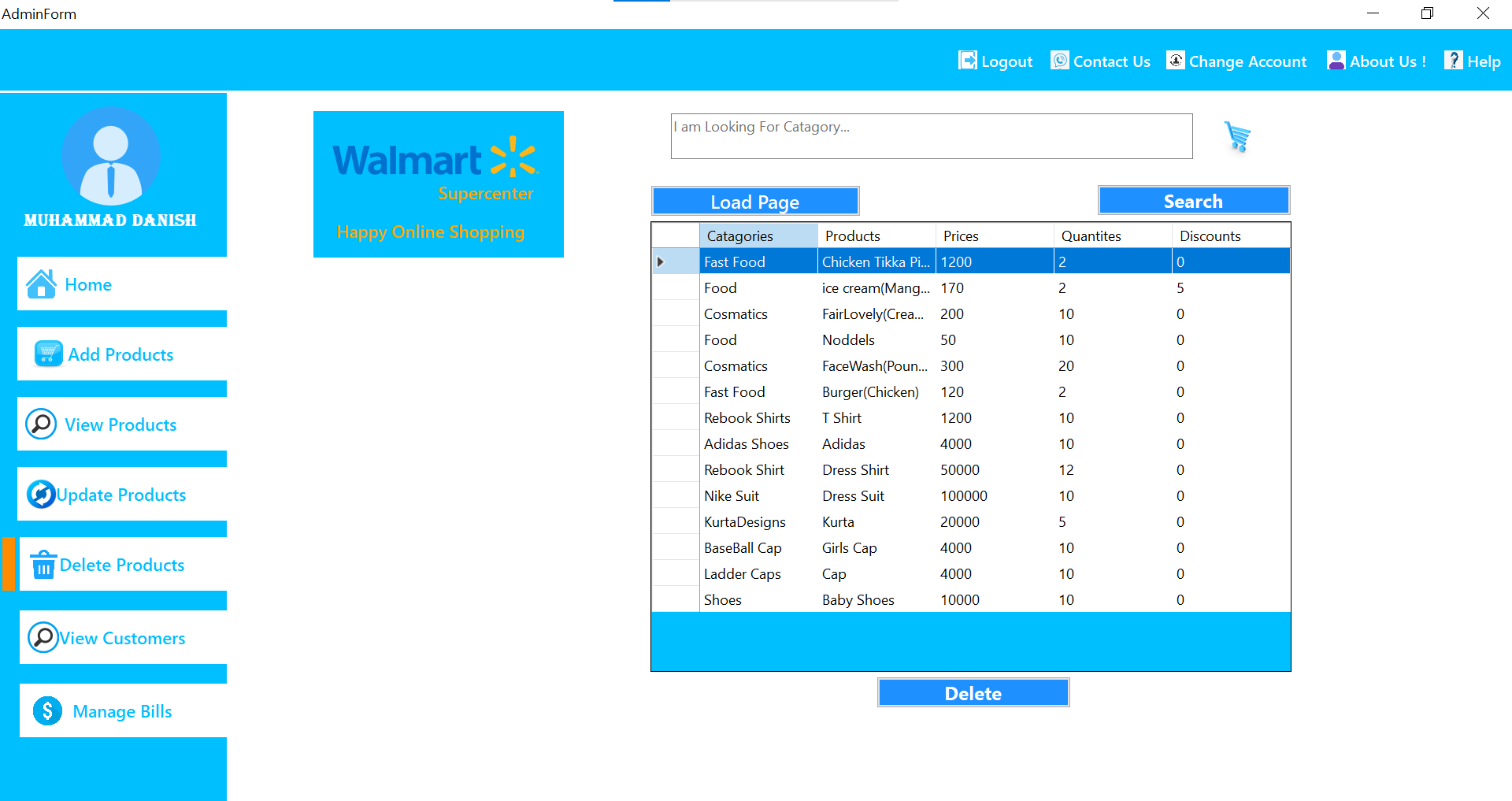
* **View Products**

****

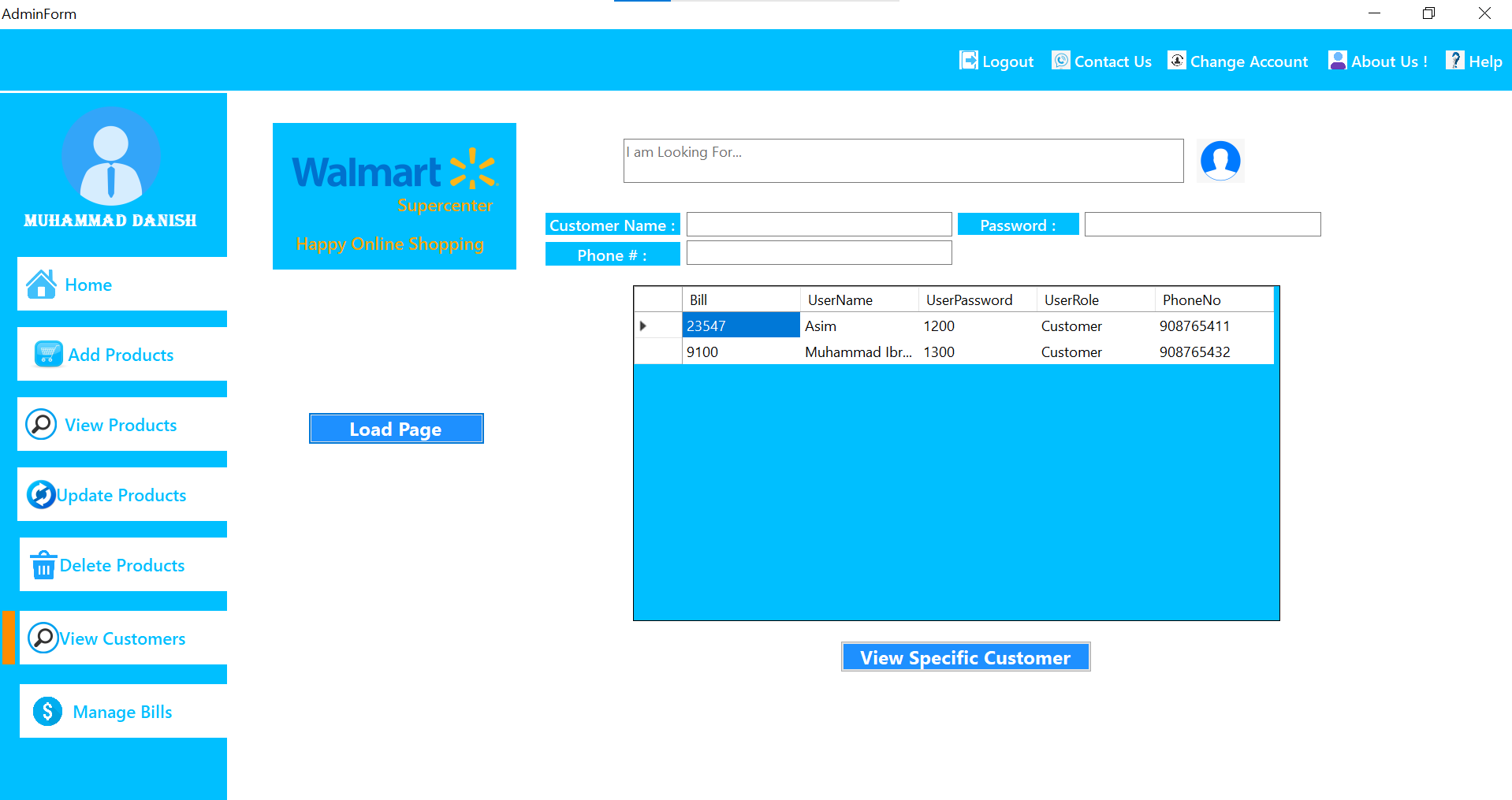
* **Update Products**

****

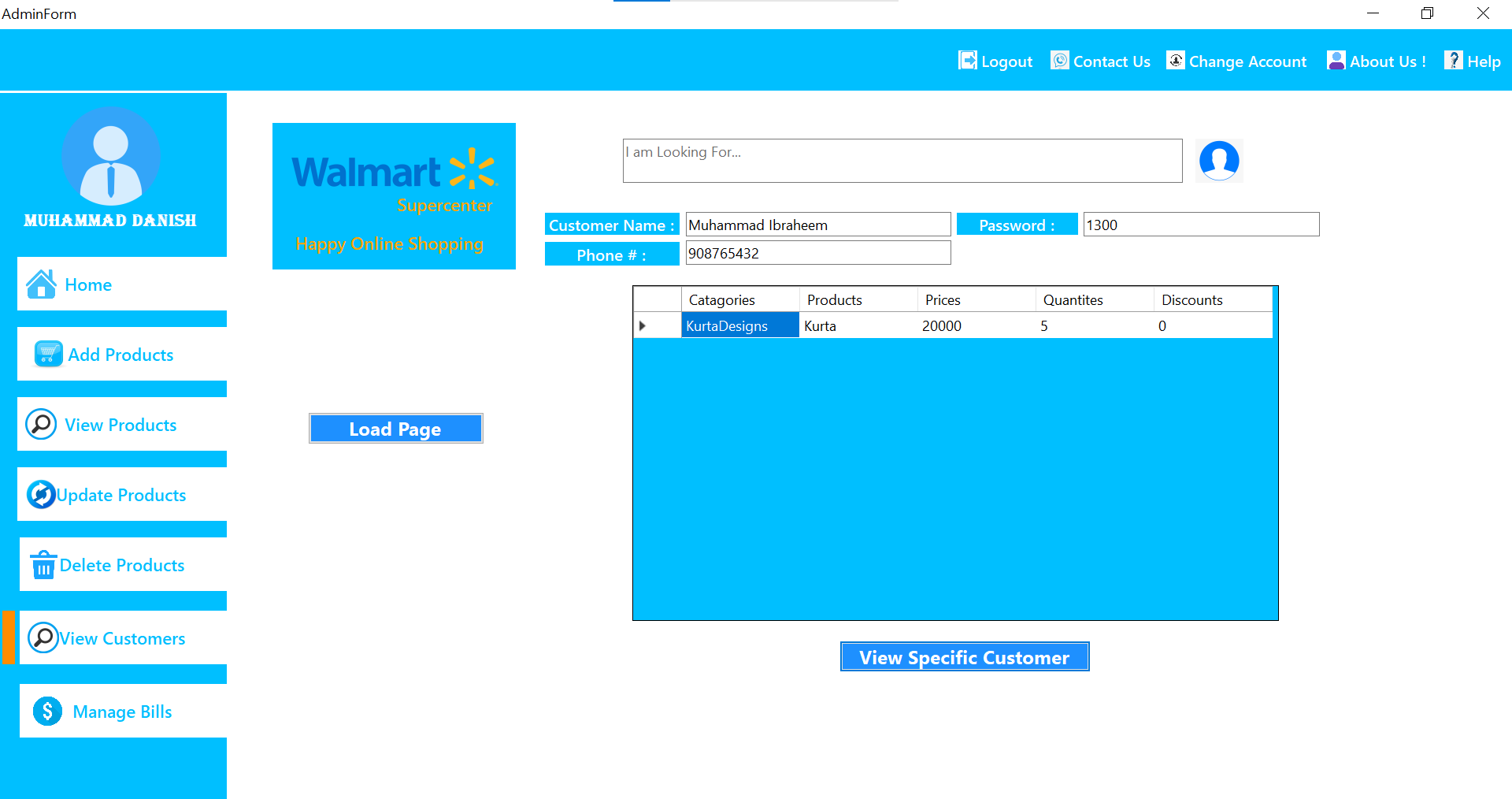
* + **Delete Products**

****

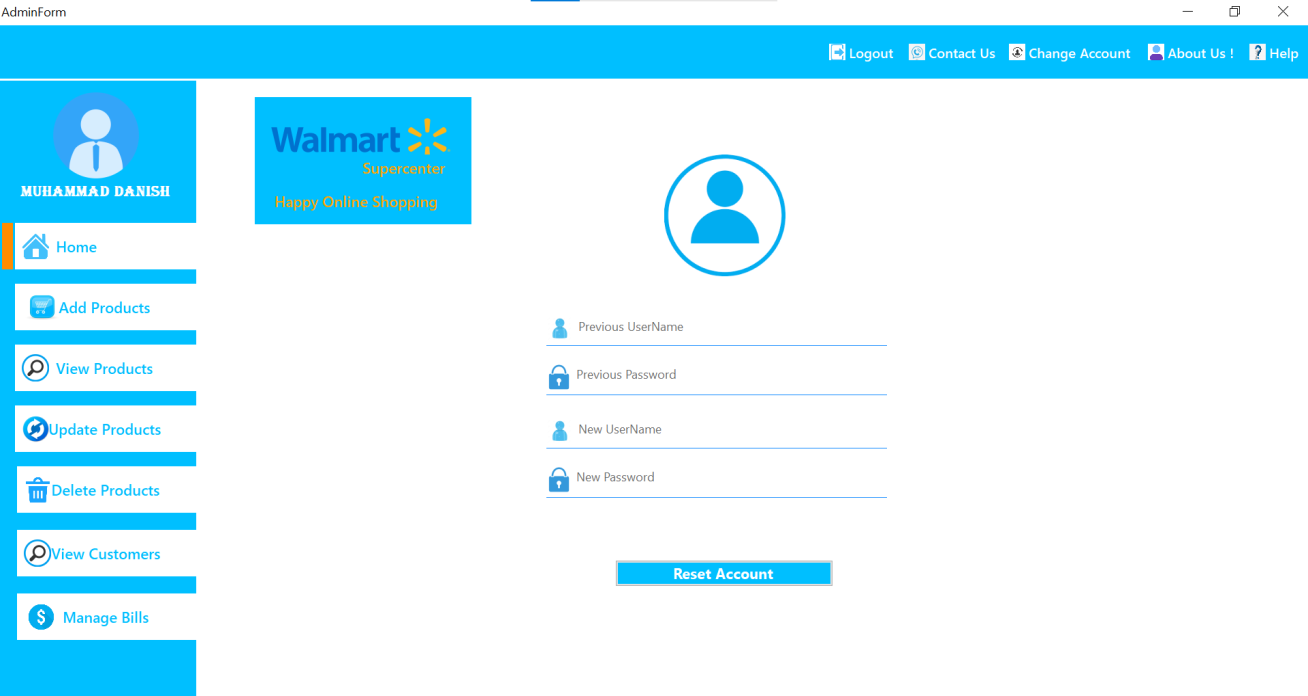
* + **View Customers**

****

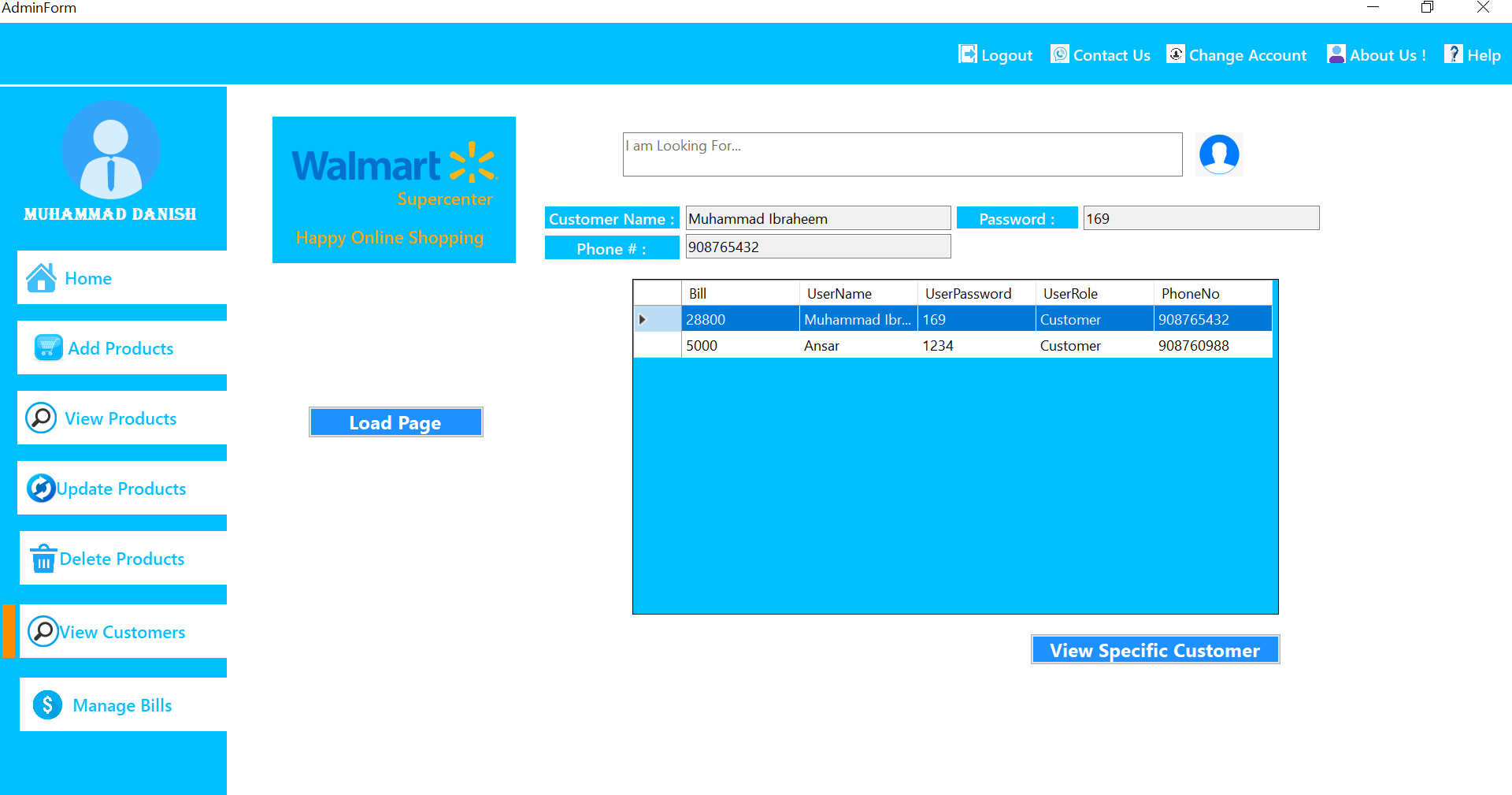
* + **View Specific Customer Button**

****

* **Change Account**

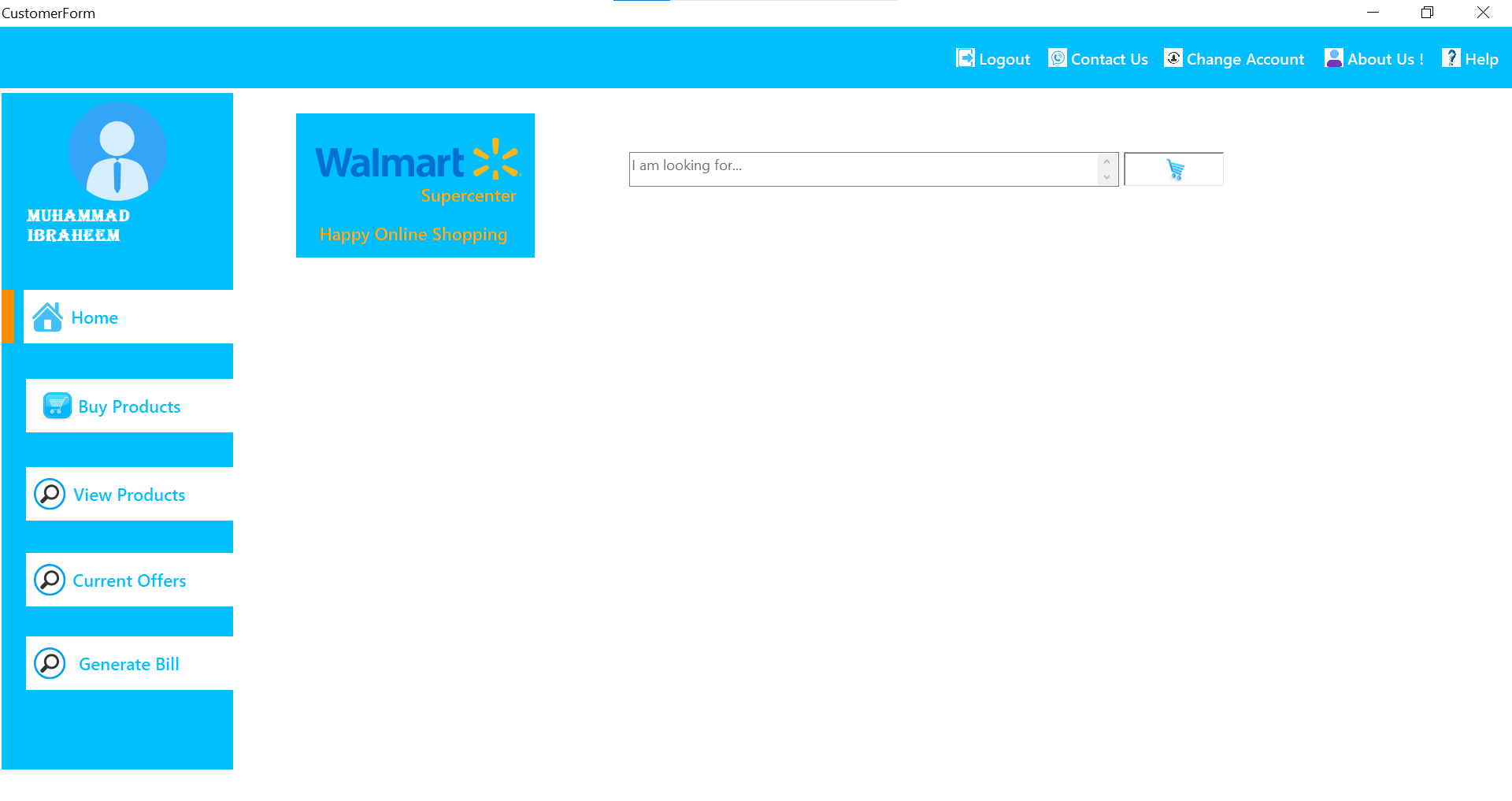
****

* **View Customer**

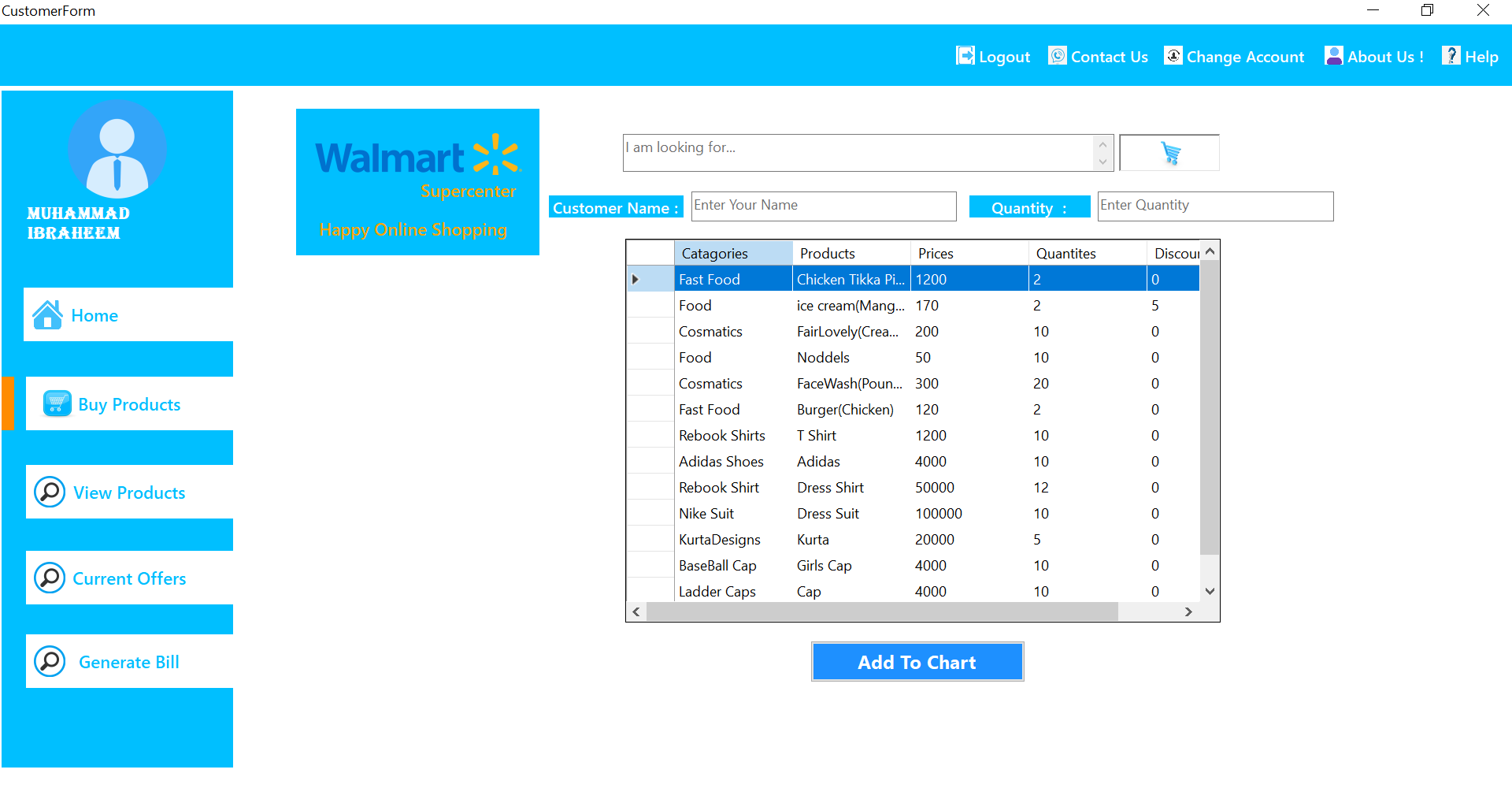
****

**Customer Section**

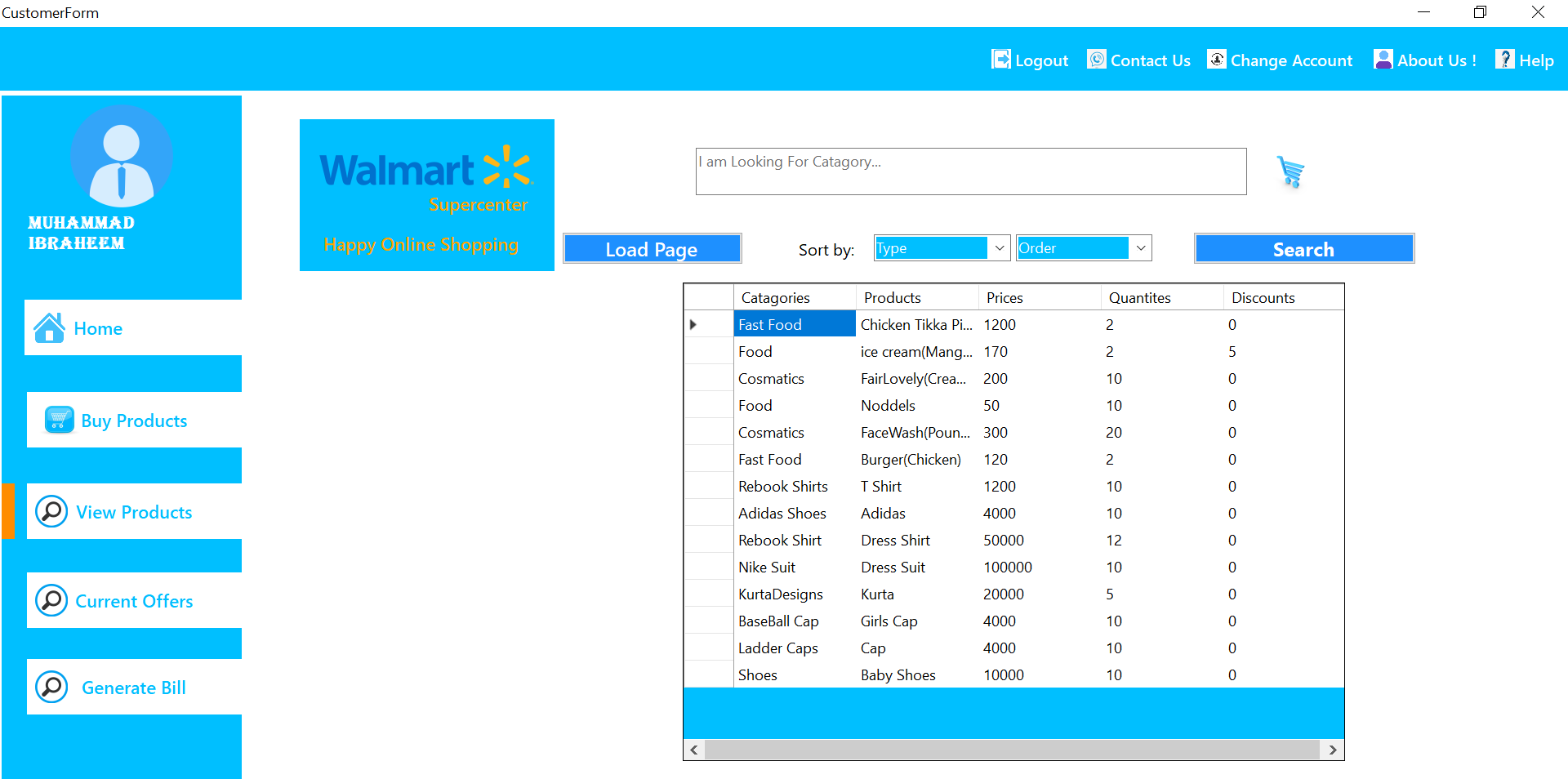
* **Customer Menu**

****

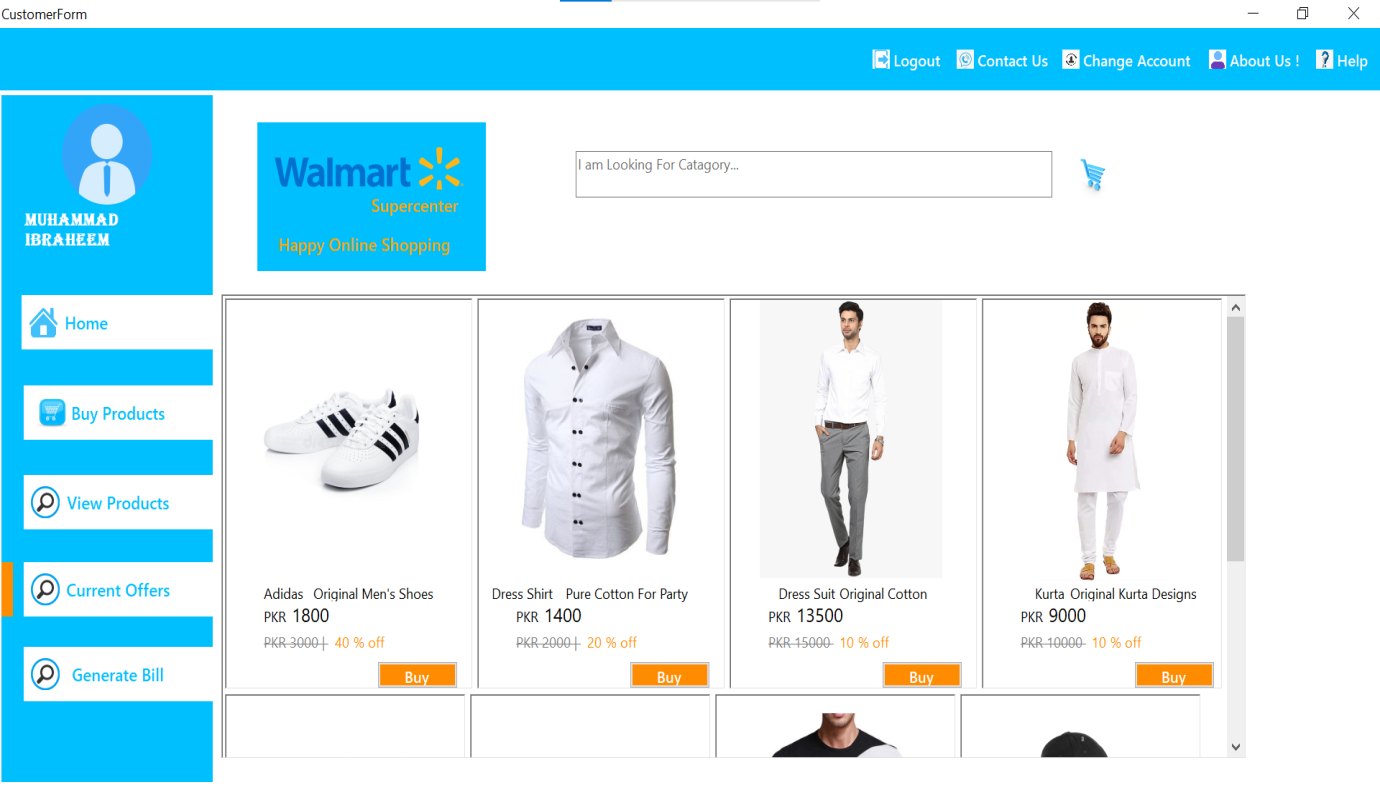
* **Buy Products**

****

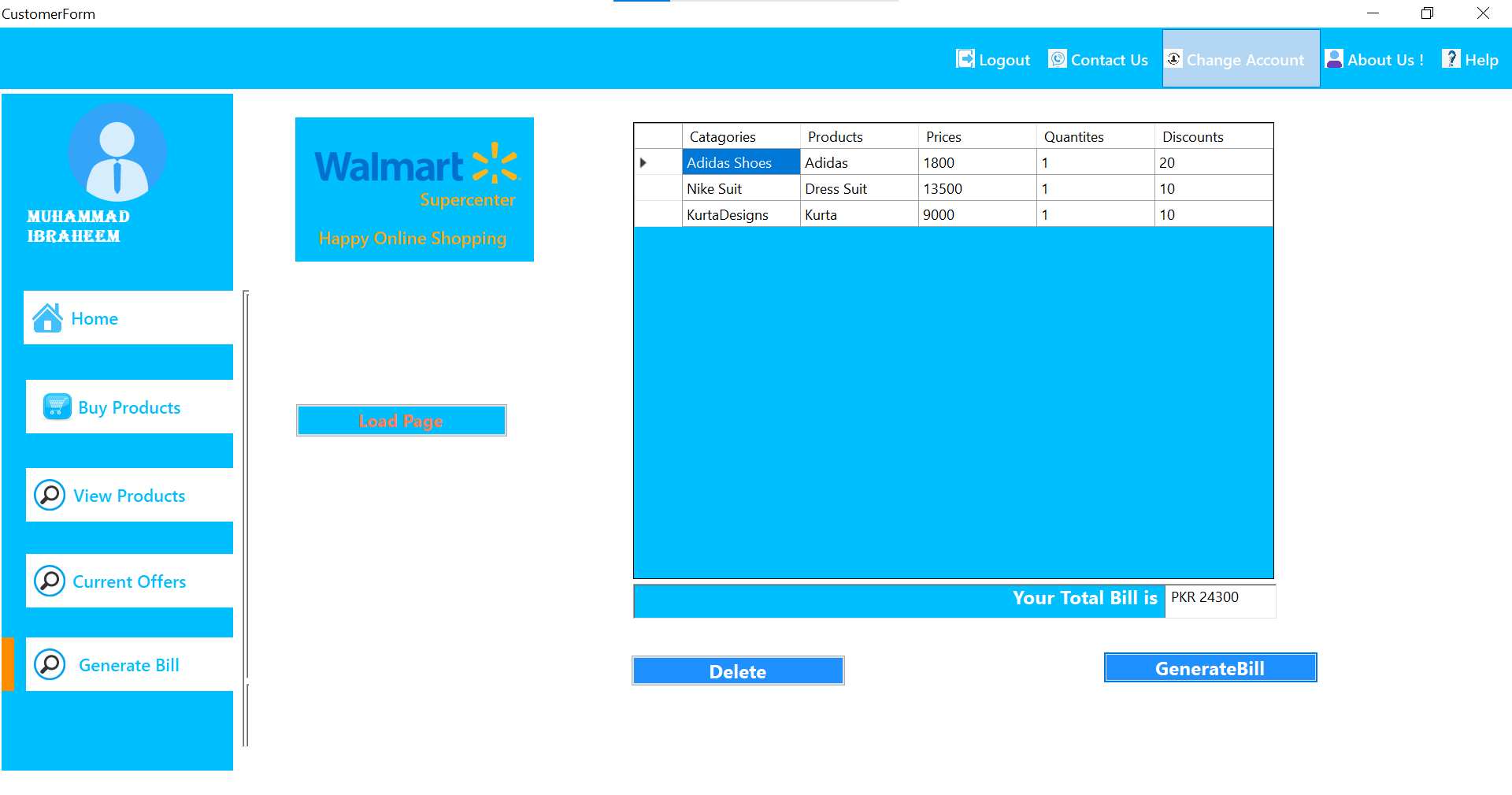
* **View Product**

****

* **Current Offers**

****

* **Generate Bill**



# **Full Code**

**MuserBL:**

public class Muser

{

private string userName;

private string userPassword;

private string userRole;

private string phoneNo;

public Muser(string UserName, string UserPassword)

{

this.UserName = UserName;

this.UserPassword = UserPassword;

}

public Muser(string userName, string userPassword, string userRole,string UserPhoneNo) : this(userName, userPassword)

{

PhoneNo = UserPhoneNo;

UserRole = userRole;

}

public string UserName

{

get => userName; set

{

if (userName != "")

{

userName = value;

}

}

}

public string UserPassword

{

get => userPassword; set

{

if (userPassword != "")

{

userPassword = value;

}

}

}

public string UserRole

{

get => userRole; set

{

if (userRole != "")

{

userRole = value;

}

}

}

public string PhoneNo { get => phoneNo; set => phoneNo = value; }

public bool IsAdmin()

{

if (UserRole == "Admin")

{

return true;

}

return false;

}

**ProductBL:**

public class ProductBL

{

private string catagories;

private string products;

private int prices;

private int quantites;

private int discounts;

public ProductBL(string catagoryName, string productName, int productPrice, int productQuantity, int productDiscount)

{

Catagories = catagoryName;

Products = productName;

Prices = productPrice;

Quantites = productQuantity;

Discounts = productDiscount;

}

public ProductBL(string productName,int productQuantity,int productPrice)

{

Products = productName;

Quantites = productQuantity;

Prices = productPrice;

}

public string Catagories { get => catagories; set => catagories = value; }

public string Products { get => products; set => products = value; }

public int Prices

{

get => prices;

set

{

if (value > 0)

{

prices = value;

}

}

}

public int Quantites { get => quantites; set

{

if (value > 0)

{

quantites = value;

}

}

}

public int Discounts { get => discounts; set => discounts = value; }

}

} if (quantity >= 0) quantity = value;

**CutomerBL :**

public class CustomerBL : Muser

{

private int bill;

private List<ProductBL> buyProduct;

public CustomerBL(string customerName, string Passsword, string Role,string PhoneNo,int bill, List<ProductBL> buyProduct):base(customerName,Passsword,Role,PhoneNo)

{

MessageBox.Show("Constructer Created");

Bill = bill;

BuyProduct = buyProduct;

}

public int Bill { get => bill; set => bill = value; }

public List<ProductBL> BuyProduct { get => buyProduct; set => buyProduct = value; }

}

}

**SpecificCustomerBL:**

//

// I made this Class For Mapping Because I dont think there is no Another way of Mapping Without this Solution According To My situation

//

public class SpecificCustomerBL : ProductBL

{

private string Name;

public SpecificCustomerBL(string Name, string Catagory, string Product, int Price, int Quantity, int Discount) : base(Catagory,Product,Price,Quantity,Discount)

{

Name1 = Name;

}

public string Name1 { get => Name; set => Name = value; }

}

**ProductDL:**

public class ProductDL

{

public static List<ProductBL> Products = new List<ProductBL>();

public static List<ProductBL> SearchList = new List<ProductBL>();

//

// SearchProduct

//

public static List<ProductBL> GetSearchProductList(string Catagory)

{

SearchList.Clear();

foreach (ProductBL product in ProductDL.Products)

{

if (product.Catagories == Catagory)

{

ProductDL.SearchList.Add(product);

}

}

return ProductDL.SearchList;

}

// Products Load From ProductFile //

public static bool Loadproducts(string path)

{

Products.Clear();

if (File.Exists(path))

{

StreamReader file = new StreamReader(path);

string record;

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string CatagoryName = splittedRecord[0];

string ProductName = splittedRecord[1];

int ProductPrize = int.Parse(splittedRecord[2]);

int ProductQuantity = int.Parse(splittedRecord[3]);

int ProductDiscount = int.Parse(splittedRecord[4]);

ProductBL product = new ProductBL(CatagoryName, ProductName, ProductPrize, ProductQuantity, ProductDiscount);

AddProductToList(product);

}

file.Close();

return true;

}

else

{

return false;

}

}

//

// Add Products To ProductList

//

public static void AddProductToList(ProductBL product)

{

Products.Add(product);

}

// Store Products in ProductFile//

public static void StoreProductsInFile(ProductBL product)

{

StreamWriter file = new StreamWriter(ProductsPath.ProductPath, true);

file.WriteLine(product.Catagories + "," + product.Products + "," + product.Prices + "," + product.Quantites + "," + product.Discounts);

file.Flush();

file.Close();

}

public static void StoreAllDataInfile(string Path,List<ProductBL> Products)

{

StreamWriter file = new StreamWriter(Path);

foreach (ProductBL product in Products)

{

file.WriteLine(product.Catagories + "," + product.Products + "," + product.Prices + "," + product.Quantites + "," + product.Discounts);

}

file.Flush();

file.Close();

}

//

//UpdateProduct

//

public static void UpdateProduct(string CatagoryName, string ProductName, ProductBL NewProduct)

{

for (int i = 0; i < Products.Count; i++)

{

if (Products[i].Catagories == CatagoryName && Products[i].Products == ProductName)

{

Products[i] = NewProduct;

}

}

}

//

//Valid Catagory

//

public static bool ValidCatagory(string Catagory)

{

foreach (ProductBL product in ProductDL.Products)

{

if (product.Catagories == Catagory)

{

return true;

}

}

return false;

}

//

//DeleteProduct

//

public static void DeleteProduct(ProductBL prod)

{

for(int i=0; i<Products.Count; i++)

{

if (Products[i].Catagories == prod.Catagories && Products[i].Products == prod.Products)

{

Products.RemoveAt(i);

}

}

}

//

//CheckOfferProductValid

//

public static ProductBL CheckPRoductValid(string ProductName)

{

foreach(ProductBL product in Products)

{

if(product.Products == ProductName && product.Quantites > 0)

{

return product;

}

}

return null;

}

//

//CalculatePricewithDiscount

//

public static int CalculatePrice(ProductBL ordersBill)

{

return (ordersBill.Prices - (ordersBill.Prices \* ordersBill.Discounts / 100)) \* ordersBill.Quantites;

}

public static ProductBL isProductExist(string Product)

{

foreach(ProductBL prod in Products)

{

if(prod.Products == Product)

{

return prod;

}

}

return null;

}

}

}

**Customer DL :**

public class CustomerDL

{

//

// List Of Only One Customer Record

//

private static List<ProductBL> CustomerOrders = new List<ProductBL>();

//

// List Of All Customer Record

//

private static List<CustomerBL> Customers = new List<CustomerBL>();

//

//List Of Orders With Customer Name

//

private static List<SpecificCustomerBL> Customer = new List<SpecificCustomerBL>();

//

//Builtin Gettter Setter Function

internal static List<ProductBL> CustomerOrders1 { get => CustomerOrders; set => CustomerOrders = value; }

internal static List<CustomerBL> Customers1 { get => Customers; set => Customers = value; }

public static List<SpecificCustomerBL> Customer1 { get => Customer; set => Customer = value; }

//

//Add Orders To List

//

public static void AddOrderToList(ProductBL orders)

{

CustomerOrders1.Add(orders);

}

//

// For Add ALL Customer With Specific Orders

public static void AddAllCustomerwithOrders(SpecificCustomerBL customer)

{

Customer1.Add(customer);

}

//

//Add All Customers With Bills In List

//

public static void AddCustomersToList(CustomerBL customer)

{

Customers1.Add(customer);

}

//

//Load Orders of Current Customer

//

public static bool LoadOrders(string Path,string Name)

{

CustomerOrders1.Clear();

StreamReader file = new StreamReader(Path);

string record;

if (File.Exists(Path))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string CustomerName = splittedRecord[0];

string CatagoryName = splittedRecord[1];

string ProductName = splittedRecord[2];

int ProductPrize = int.Parse(splittedRecord[3]);

int Quantity = int.Parse(splittedRecord[4]);

int Discount = int.Parse(splittedRecord[5]);

if (CustomerName == Name)

{

ProductBL product = new ProductBL(CatagoryName, ProductName, ProductPrize, Quantity, Discount);

AddOrderToList(product);

}

}

file.Close();

return true;

}

else

{

return false;

}

}

//

//Check Orders Exist in Current Customer List

//

public static ProductBL isOrderExist(string Product)

{

foreach (ProductBL prod in CustomerOrders1)

{

if(prod.Products == Product)

{

return prod;

}

}

return null;

}

//

//Load All customers With Bills And Orders

//

public static bool LoadCustomers(string path)

{

StreamReader f = new StreamReader(path);

string record;

if (File.Exists(path))

{

while ((record = f.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string CustomerName = splittedRecord[0];

string Password = splittedRecord[1];

string Role = splittedRecord[2];

string PhoneNo = splittedRecord[3];

int Bill = int.Parse(splittedRecord[4]);

string[] splittedRecordForPreference = splittedRecord[5].Split(';');

LoadOrders(ProductsPath.OrdersPath, CustomerName);

List<ProductBL> products = new List<ProductBL>();

for (int x = 0; x < splittedRecordForPreference.Length; x++)

{

ProductBL prod = ProductDL.isProductExist(splittedRecordForPreference[x]);

if (prod != null)

{

MessageBox.Show(CustomerName + prod.Products);

prod = isOrderExist(splittedRecordForPreference[x]);

if (prod != null)

{

// if (!(products.Contains(prod)))

products.Add(prod);

}

}

}

CustomerBL customer = new CustomerBL(CustomerName, Password,Role,PhoneNo, Bill, products);

AddCustomersToList(customer);

}

f.Close();

return true;

}

else

return false;

}

//

//Store All Customers

//

public static void StoreCustomers()

{

string OrdersProduct = "";

StreamWriter f = new StreamWriter(CustomerPath.CustomersPath);

for (int i = 0; i < Customers.Count; i++)

{

for (int x = 0; x < Customers[i].BuyProduct.Count - 1; x++)

{

OrdersProduct = OrdersProduct + Customers[i].BuyProduct[x].Products + ";";

}

OrdersProduct = OrdersProduct + Customers[i].BuyProduct[Customers[i].BuyProduct.Count - 1].Products;

f.WriteLine(Customers[i].UserName + "," + Customers[i].UserPassword + "," + Customers[i].UserRole + "," + Customers[i].PhoneNo + "," + Customers[i].Bill + ","+ OrdersProduct);

OrdersProduct = "";

}

f.Flush();

f.Close();

}

//

// Store Orders of Current Customer

//

public static void storeOrders(ProductBL product,string Name)

{

StreamWriter file = new StreamWriter(ProductsPath.OrdersPath, true);

file.WriteLine(Name + "," + product.Catagories + "," + product.Products + "," + product.Prices + "," + product.Quantites + "," + product.Discounts);

file.Flush();

file.Close();

}

//

// Store Customer Orders in File for Mapping

//

public static void StoreAllCustomersOrders()

{

StreamWriter file = new StreamWriter(ProductsPath.OrdersPath, false);

foreach (SpecificCustomerBL customer in Customer1)

{

file.WriteLine(customer.Name1 + "," + customer.Catagories + "," + customer.Products + "," + customer.Prices + "," + customer.Quantites + "," + customer.Discounts);

}

file.Flush();

file.Close();

}

//

//Delete Orders before Generate Bill

//

public static void DeleteOrders(ProductBL product)

{

for(int i=0; i< CustomerOrders1.Count; i++)

{

if (CustomerOrders1[i].Catagories == product.Catagories && CustomerOrders1[i].Products == product.Products)

{

CustomerOrders1.RemoveAt(i);

}

}

}

//

// For Assign Customer If Exist otherwise return for Add

//

public static CustomerBL AssignListToCustomer(List<ProductBL> prod, Muser Customer)

{

CustomerBL cust = new CustomerBL(Customer.UserName, Customer.UserPassword, Customer.UserRole, Customer.PhoneNo, GenerateBill.Bills, prod);

int count = 0;

for (int i=0; i<Customers1.Count; i++)

{

if (Customers1[i].UserName == Customer.UserName)

{

Customers1[i].UserName = cust.UserName;

Customers1[i].UserPassword = cust.UserPassword;

Customers1[i].UserRole = cust.UserRole;

Customers1[i].PhoneNo = cust.PhoneNo;

Customers1[i].Bill = cust.Bill;

Customers[i].BuyProduct = cust.BuyProduct;

count++;

}

}

if(count == 0)

{

return cust;

}

return null;

}

//

// Get Customer Object

//

public static CustomerBL GetCustomerObject(string Name)

{

foreach(CustomerBL customer in Customers)

{

if(customer.UserName == Name)

{

return customer;

}

}

return null;

}

//

// Admin Full Fill Orders Form Current Customer List

//

public static void FullFillOrders(SpecificCustomerBL customer)

{

for(int i=0; i<Customer1.Count; i++ )

{

if (Customer1[i].Name1 == customer.Name1 && Customer1[i].Products == customer.Products)

{

Customer1.RemoveAt(i);

}

}

}

//

// Delete Customer From His Current List After Full Fill Orders

//

public static void DeleteCustomer(List<ProductBL> Product,ProductBL product,CustomerBL cust)

{

for(int i=0; i<Product.Count; i++)

{

foreach (ProductBL prod in ProductDL.Products)

{

if (Product[i].Products == product.Products && prod.Products == product.Products)

{

cust.Bill = cust.Bill - (prod.Prices \* product.Quantites);

Product.RemoveAt(i);

}

}

}

}

//

// Load All Customer With Name For Mapping

//

public static bool LoadallCutomerOrders(string Path)

{

Customer1.Clear();

StreamReader file = new StreamReader(Path);

string record;

if (File.Exists(Path))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string CustomerName = splittedRecord[0];

string CatagoryName = splittedRecord[1];

string ProductName = splittedRecord[2];

int ProductPrize = int.Parse(splittedRecord[3]);

int Quantity = int.Parse(splittedRecord[4]);

int Discount = int.Parse(splittedRecord[5]);

SpecificCustomerBL orders = new SpecificCustomerBL(CustomerName,CatagoryName, ProductName, ProductPrize, Quantity, Discount);

AddAllCustomerwithOrders(orders);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**MuserDL:**

public class MuserDL

{

//

//List Of Users

//

private static List<Muser> user = new List<Muser>();

//

//gettter Settter

//

internal static List<Muser> User { get => user; set => user = value; }

//

//Check While Signin

//

public static Muser Signin(Muser user)

{

foreach (Muser SUser in User)

{

if (SUser.UserName == user.UserName && SUser.UserPassword == user.UserPassword)

{

return SUser;

}

}

return null;

}

//

// Load Users Form file

//

public static bool ReadData(string path)

{

StreamReader file = new StreamReader(path);

string record;

if (File.Exists(path))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string UserName = splittedRecord[0];

string UserPassword = splittedRecord[1];

string UserRole = splittedRecord[2];

string UserPhoneNo = splittedRecord[3];

Muser user = new Muser(UserName, UserPassword, UserRole,UserPhoneNo);

AddUserTOList(user);

}

file.Close();

return true;

}

else

return false;

}

//

// Store User While Sign Up

//

public static void StoreinFile(Muser user)

{

StreamWriter file = new StreamWriter(UserId.UserIDPath, true);

file.WriteLine(user.UserName + "," + user.UserPassword + "," + user.UserRole + "," + user.PhoneNo);

file.Flush();

file.Close();

}

//

// After Update Store All Users

//

public static void StoreAllUserinFile()

{

StreamWriter file = new StreamWriter(UserId.UserIDPath, false);

foreach (Muser user in User)

{

file.WriteLine(user.UserName + "," + user.UserPassword + "," + user.UserRole + "," + user.PhoneNo);

}

file.Flush();

file.Close();

}

//

// Add User To List

//

public static void AddUserTOList(Muser user)

{

User.Add(user);

}

//

// For Get Customer Detail

//

public static Muser GetCustomerDetail(string Name)

{

foreach(Muser user in User)

{

if(user.UserName == Name)

{

return user;

}

}

return null;

}

//

// For Change Account Check Validation

//

public static bool ChangeUserAccount(Muser Previous, Muser New)

{

int count = 0;

foreach(Muser user in User)

{

if(user.UserName == Previous.UserName && user.UserPassword == Previous.UserPassword)

{

user.UserName = New.UserName;

user.UserPassword = New.UserPassword;

count++;

}

}

if(count == 1)

{

return true;

}

return false;

}

//

// Update Account of Both Admin And Customer

//

public static void UpdateAdmininList(string Name,string Role)

{

foreach(Muser user in User)

{

if(user.UserName == Name)

{

if(user.UserRole == "Admin")

{

}

}

}

}

} **GUI**

**AdminForm:**

public AdminForm(string Name)

{

InitializeComponent();

ProductDL.Loadproducts(ProductsPath.ProductPath);

SlidePanel.Height = HomeButton.Height;

SlidePanel.Top = HomeButton.Top;

adminMainPage1.BringToFront();

AdminName = Name;

}

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

{

AdminNametext.Text = AdminName;

ProductDL.Loadproducts(ProductsPath.ProductPath);

}

private void guna2Button5\_Click(object sender, EventArgs e)

{

SlidePanel.Height = HomeButton.Height;

SlidePanel.Top = HomeButton.Top;

adminMainPage1.BringToFront();

}

private void guna2Button9\_Click(object sender, EventArgs e)

{

SlidePanel.Height = DeleteButton.Height;

SlidePanel.Top = DeleteButton.Top;

deleteProducts1.BringToFront();

}

private void SlidePanel\_Paint\_1(object sender, PaintEventArgs e)

{

}

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

{

SlidePanel.Height = AddProductButton.Height;

SlidePanel.Top = AddProductButton.Top;

addProducts1.BringToFront();

}

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

{

SlidePanel.Height = ViewProductButton.Height;

SlidePanel.Top = ViewProductButton.Top;

viewProducts1.BringToFront();

}

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

{

SlidePanel.Height = UpdateButton.Height;

SlidePanel.Top = UpdateButton.Top;

updateProducts1.BringToFront();

}

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

{

SlidePanel.Height = CustomerViewButton.Height;

SlidePanel.Top = CustomerViewButton.Top;

viewCustomers1.BringToFront();

}

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

{

Process.Start("https://www.mozilla.org/en-US/firefox/");

}

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

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Logout!", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if(dialogResult == DialogResult.Yes)

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

}

else if(dialogResult == DialogResult.No)

{

this.Activate();

}

}

private void AdminForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Logout!", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

this.Activate();

}

}

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

{

changeAdminAccount1.BringToFront();

}

**AddProductForm:**

public partial class AddProducts : UserControl

{

public AddProducts()

{

InitializeComponent();

}

public void ByindProductList()

{

AddProductGrid.DataSource = null;

AddProductGrid.DataSource = ProductDL.Products;

AddProductGrid.Refresh();

}

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

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

AddProductGrid.DataSource = ProductDL.Products;

}

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

{

try

{

string CatagoryName = CatagoryText.Text;

string ProductName = ProductText.Text;

int Quantity = int.Parse(QuantityText.Text);

int Price = int.Parse(PriceText.Text);

int Discount = int.Parse(discountText.Text);

ProductBL product = new ProductBL(CatagoryName, ProductName, Price, Quantity, Discount);

ProductDL.AddProductToList(product);

ProductDL.StoreProductsInFile(product);

ByindProductList();

CatagoryText.Clear();

ProductText.Clear();

QuantityText.Clear();

PriceText.Clear();

discountText.Clear();

}

catch(Exception exp)

{

MessageBox.Show(exp.Message);

}

}

**ViewProductForm:**

public partial class ViewProducts : UserControl

{

public ViewProducts()

{

InitializeComponent();

}

public void DataByind()

{

ViewProductGridV.DataSource = null;

ViewProductGridV.DataSource = ProductDL.Products;

ViewProductGridV.Refresh();

}

public void SortProduct()

{

if (TypeCombobox.Text == "Price")

{

if (SortOrderBox.Text == "Ascending")

{

ProductDL.Products.OrderBy(o => o.Prices).ToList();

}

else if (SortOrderBox.Text == "Descending")

{

ProductDL.Products.OrderByDescending(o => o.Prices).ToList();

}

}

else if (TypeCombobox.Text == "Discount")

{

if (SortOrderBox.Text == "Ascending")

{

ProductDL.Products.OrderBy(o => o.Prices).ToList();

}

else if (SortOrderBox.Text == "Descending")

{

ProductDL.Products.OrderByDescending(o => o.Prices).ToList();

}

}

DataByind();

}

private void guna2Button2\_Click(object sender, EventArgs e)

{

}

public void LoadForm()

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

DataByind();

}

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

{

LoadForm();

}

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

{

string Catagory = ViewSeachBoxtext.Text;

if (ProductDL.ValidCatagory(Catagory))

{

List<ProductBL> product = ProductDL.GetSearchProductList(Catagory);

ViewProductGridV.DataSource = null;

ViewProductGridV.DataSource = product;

ViewProductGridV.Refresh();

}

else

{

MessageBox.Show("You Enter Wrong Catagory Check Again!");

}

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

SortProduct();

}

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

{

LoadForm();

}

}

**DeleteProductForm:**

public partial class DeleteProducts : UserControl

{

public DeleteProducts()

{

InitializeComponent();

}

public void DataByind()

{

DeleteGrid.DataSource = null;

DeleteGrid.DataSource = ProductDL.Products;

DeleteGrid.Refresh();

}

public void LoadForm()

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

DataByind();

}

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

{

string Catagory = SearchDelBox.Text;

if (ProductDL.ValidCatagory(Catagory))

{

List<ProductBL> product = ProductDL.GetSearchProductList(Catagory);

DeleteGrid.DataSource = null;

DeleteGrid.DataSource = product;

DeleteGrid.Refresh();

}

else

{

MessageBox.Show("You Enter Wrong Catagory Check Again!");

}

}

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

{

LoadForm();

}

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

{

LoadForm();

}

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

{

ProductBL product = (ProductBL)DeleteGrid.CurrentRow.DataBoundItem;

ProductDL.DeleteProduct(product);

ProductDL.StoreAllDataInfile(ProductsPath.ProductPath, ProductDL.Products);

DataByind();

UpdateProducts Update = new UpdateProducts();

Update.DataByind();

ViewProducts View = new ViewProducts();

View.DataByind();

AddProducts AddProd = new AddProducts();

AddProd.ByindProductList();

}

}

**UpdateProductForm:**

public partial class UpdateProducts : UserControl

{

public UpdateProducts()

{

InitializeComponent();

}

public void DataByind()

{

UpdateGridView.DataSource = null;

UpdateGridView.DataSource = ProductDL.Products;

UpdateGridView.Refresh();

}

/\* I Make This Attributes Because I Make Only Form For Update Products.

So Thats Why I Can Make Object of ProductBL that Passes To

Another Form.\*/

public string catagoryName;

public string productName;

public void LoadForm()

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

DataByind();

}

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

{

LoadForm();

}

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

{

LoadForm();

}

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

{

string CatagoryName = UpdateCatagorytext.Text;

string ProductName = UpdateProductText.Text;

int Quantity = int.Parse(UpdateQuantityText.Text);

int Price = int.Parse(UpdatePriceText.Text);

int discount = int.Parse(UpdateDiscountText.Text);

ProductBL product = new ProductBL(CatagoryName, ProductName, Price, Quantity, discount);

ProductDL.UpdateProduct(catagoryName, productName, product);

ProductDL.StoreAllDataInfile(ProductsPath.ProductPath,ProductDL.Products);

DataByind();

UpdateCatagorytext.Clear();

UpdateProductText.Clear();

UpdateQuantityText.Clear();

UpdatePriceText.Clear();

UpdateDiscountText.Clear();

}

private void UpdateGridView\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

int indexRow = e.RowIndex;

DataGridViewRow row = UpdateGridView.Rows[indexRow];

UpdateCatagorytext.Text = row.Cells[0].Value.ToString();

UpdateProductText.Text = row.Cells[1].Value.ToString();

UpdatePriceText.Text = row.Cells[2].Value.ToString();

UpdateQuantityText.Text = row.Cells[3].Value.ToString();

UpdateDiscountText.Text = row.Cells[4].Value.ToString();

catagoryName = UpdateCatagorytext.Text;

productName = UpdateProductText.Text;

}

}

**ViewCustomers:**

public partial class ViewCustomers : UserControl

{

public ViewCustomers()

{

InitializeComponent();

}

public static string CustomerName;

public static int count;

public void DataByind()

{

ViewcustomerGrid.DataSource = null;

ViewcustomerGrid.DataSource = CustomerDL.Customers1;

ViewcustomerGrid.Refresh();

}

public void LoadForm()

{

CustomerDL.LoadCustomers(CustomerPath.CustomersPath);

}

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

{

FullFillOrderButton.Visible = false;

CustomerDL.LoadCustomers(CustomerPath.CustomersPath);

ViewcustomerGrid.DataSource = CustomerDL.Customers1;

}

private void ViewCustomers\_Load\_1(object sender, EventArgs e)

{

}

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

{

count = 0;

if (CustomerDL.Customers1.Count == 0)

{

CustomerDL.LoadCustomers(CustomerPath.CustomersPath);

}

DataByind();

}

public void OrdersByind(CustomerBL customer)

{

MessageBox.Show(customer.BuyProduct.Count.ToString());

ViewcustomerGrid.DataSource = null;

ViewcustomerGrid.DataSource = customer.BuyProduct;

ViewcustomerGrid.Refresh();

}

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

{

try

{

CustomerDL.LoadOrders(ProductsPath.OrdersPath, CustomerName);

CustomerBL customer = CustomerDL.GetCustomerObject(CustomerName);

OrdersByind(customer);

FullFillOrderButton.Visible = true;

}

catch

{

MessageBox.Show("Please Select Any Row");

}

count = 1;

}

private void ViewcustomerGrid\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

int indexRow = e.RowIndex;

DataGridViewRow row = ViewcustomerGrid.Rows[indexRow];

if (count == 0)

{

Nametxt.Text = row.Cells[1].Value.ToString();

Passwordtxt.Text = row.Cells[2].Value.ToString();

Phonetxt.Text = row.Cells[4].Value.ToString();

CustomerName = Nametxt.Text;

}

}

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

{

CustomerDL.LoadallCutomerOrders(ProductsPath.OrdersPath);

try

{

ProductBL product = (ProductBL)ViewcustomerGrid.CurrentRow.DataBoundItem;

SpecificCustomerBL customer = new SpecificCustomerBL(CustomerName, product.Catagories, product.Products, product.Prices, product.Quantites, product.Discounts);

CustomerDL.FullFillOrders(customer);

CustomerDL.StoreAllCustomersOrders();

MessageBox.Show(CustomerName);

CustomerBL Customer = CustomerDL.GetCustomerObject(CustomerName);

CustomerDL.DeleteCustomer(Customer.BuyProduct, product,Customer);

CustomerDL.StoreCustomers();

MessageBox.Show(Customer.BuyProduct.Count.ToString());

OrdersByind(Customer);

}

catch (Exception exp)

{

MessageBox.Show(exp.Message);

}

}

**CustomerMenu:**

public partial class CustomerForm : Form

{

public static string CustomerName;

public CustomerForm(string Name)

{

InitializeComponent();

SlidePanel2.Height = cutomerHomeButton.Height;

SlidePanel2.Top = cutomerHomeButton.Top;

customerHomeForm1.BringToFront();

CustomerName = Name;

}

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

{

SlidePanel2.Height = CurrentOfferButton.Height;

SlidePanel2.Top = CurrentOfferButton.Top;

customersCuurrentOffers1.BringToFront();

}

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

{

SlidePanel2.Height = cutomerHomeButton.Height;

SlidePanel2.Top = cutomerHomeButton.Top;

customerHomeForm1.BringToFront();

}

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

{

SlidePanel2.Height = customersViewProductButton.Height;

SlidePanel2.Top = customersViewProductButton.Top;

viewProducts1.BringToFront();

}

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

{

SlidePanel2.Height = GenerateBillButton.Height;

SlidePanel2.Top = GenerateBillButton.Top;

generateBill1.BringToFront();

}

private void buyProduct1\_Load(object sender, EventArgs e)

{

}

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

{

SlidePanel2.Height = BuyButton.Height;

SlidePanel2.Top = BuyButton.Top;

buyProduct1.BringToFront();

}

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

{

CustomerNameText.Text = CustomerName;

if (CustomerDL.Customers1.Count == 0)

{

CustomerDL.LoadCustomers(CustomerPath.CustomersPath);

}

}

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

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Logout!", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

Muser Customer = MuserDL.GetCustomerDetail(CustomerForm.CustomerName);

CustomerBL cust = CustomerDL.AssignListToCustomer(CustomerDL.CustomerOrders1,Customer);

if (cust != null)

{

CustomerDL.AddCustomersToList(cust);

}

if(CustomerDL.Customers1.Count != 0)

{

MessageBox.Show(CustomerDL.Customers1.Count.ToString());

}

CustomerDL.StoreCustomers();

}

else if (dialogResult == DialogResult.No)

{

this.Activate();

}

}

private void CustomerForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Logout!", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

this.Activate();

}

}

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

{

MessageBox.Show("Conatct With us At Email: dnmughal54321@gmail.com", "Contact us", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

}

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

{

Process.Start("https://www.google.com/");

}

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

{

changeAdminAccount1.BringToFront();

}

}

}

**ViewProduct :**

public partial class ViewProducts : UserControl

{

public ViewProducts()

{

InitializeComponent();

}

public void DataByind()

{

ViewProductGridV.DataSource = null;

ViewProductGridV.DataSource = ProductDL.Products;

ViewProductGridV.Refresh();

}

public void SortProduct()

{

if (TypeCombobox.Text == "Price")

{

if (SortOrderBox.Text == "Ascending")

{

ProductDL.Products.OrderBy(o => o.Prices).ToList();

}

else if (SortOrderBox.Text == "Descending")

{

ProductDL.Products.OrderByDescending(o => o.Prices).ToList();

}

}

else if (TypeCombobox.Text == "Discount")

{

if (SortOrderBox.Text == "Ascending")

{

ProductDL.Products.OrderBy(o => o.Prices).ToList();

}

else if (SortOrderBox.Text == "Descending")

{

ProductDL.Products.OrderByDescending(o => o.Prices).ToList();

}

}

DataByind();

}

private void guna2Button2\_Click(object sender, EventArgs e)

{

}

public void LoadForm()

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

DataByind();

}

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

{

LoadForm();

}

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

{

string Catagory = ViewSeachBoxtext.Text;

if (ProductDL.ValidCatagory(Catagory))

{

List<ProductBL> product = ProductDL.GetSearchProductList(Catagory);

ViewProductGridV.DataSource = null;

ViewProductGridV.DataSource = product;

ViewProductGridV.Refresh();

}

else

{

MessageBox.Show("You Enter Wrong Catagory Check Again!");

}

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

SortProduct();

}

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

{

LoadForm();

}

}

**BuyProduct:**

public partial class BuyProduct : UserControl

{

public BuyProduct()

{

InitializeComponent();

}

private void tableLayoutPanel3\_Paint(object sender, PaintEventArgs e)

{

}

private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

ProductDL.Loadproducts(ProductsPath.ProductPath);

BuyGridView.DataSource = ProductDL.Products;

}

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

{

try

{

int Quantity = int.Parse(customerQuantitytxt.Text);

ProductBL orders = (ProductBL)BuyGridView.CurrentRow.DataBoundItem;

if (orders.Quantites > Quantity)

{

orders.Quantites = orders.Quantites - Quantity;

ProductBL ordersBill = new ProductBL(orders.Catagories, orders.Products, orders.Prices, orders.Quantites, orders.Discounts);

ordersBill.Quantites = Quantity;

int Price = ProductDL.CalculatePrice(ordersBill);

ordersBill.Prices = Price;

CustomerDL.AddOrderToList(ordersBill);

CustomerDL.storeOrders(ordersBill,CustomerForm.CustomerName);

MessageBox.Show("Your Orders SuccessFully Add To Your Orders List Please Visit Generate Bill Option!");

}

customerQuantitytxt.Clear();

}

catch

{

MessagePanel.Visible = true;

}

}

**CurrentOffer:**

public partial class CustomersCuurrentOffers : UserControl

{

public CustomersCuurrentOffers()

{

InitializeComponent();

}

public void CheckOfferProductValid(string productName,int Price,int Discount)

{

ProductBL product = ProductDL.CheckPRoductValid(productName);

if (product != null)

{

product.Quantites = product.Quantites - 1;

product.Products = productName;

int Quantites = 1;

ProductBL ordersProduct = new ProductBL(product.Catagories, product.Products, Price, Quantites, Discount);

CustomerDL.AddOrderToList(ordersProduct);

CustomerDL.storeOrders(ordersProduct, CustomerForm.CustomerName);

MessageBox.Show("Your Orders SuccessFully Add To Your Orders List Please Visit Generate Bill Option!");

}

else

{

MessageBox.Show("Sorry Now Product is Not Available!");

}

}

private void Offer1BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer1ProductName.Text;

int Price = int.Parse(Offer1Price.Text);

int Discount = int.Parse(Offer2Discount.Text);

CheckOfferProductValid(ProductName,Price,Discount);

}

private void Offer2BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer2ProductName.Text;

int Price = int.Parse(Offer2Price.Text);

int Discount = int.Parse(Offer2Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer3BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer3ProductName.Text;

int Price = int.Parse(Offer3Price.Text);

int Discount = int.Parse(Offer3Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer4BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer4ProductName.Text;

int Price = int.Parse(Offer4Price.Text);

int Discount = int.Parse(Offer4Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer8BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer8ProductName.Text;

int Price = int.Parse(Offer8Price.Text);

int Discount = int.Parse(Offer8Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer6Button\_Click(object sender, EventArgs e)

{

string ProductName = Offer6ProductName.Text;

int Price = int.Parse(Offer6Price.Text);

int Discount = int.Parse(Offer6Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer5BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer5ProductName.Text;

int Price = int.Parse(Offer5Price.Text);

int Discount = int.Parse(Offer5Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

private void Offer7BuyButton\_Click(object sender, EventArgs e)

{

string ProductName = Offer7ProductName.Text;

int Price = int.Parse(Offer7Price.Text);

int Discount = int.Parse(Offer7Discount.Text);

CheckOfferProductValid(ProductName, Price, Discount);

}

}

}

**GenerateBillForm:**

public partial class GenerateBill : UserControl

{

public GenerateBill()

{

InitializeComponent();

}

private static int sum;

public static int Bills;

public void OrdersByind()

{

GenerateBillGrid.DataSource = null;

GenerateBillGrid.DataSource = CustomerDL.CustomerOrders1;

GenerateBillGrid.Refresh();

}

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

{

CustomerDL.LoadOrders(ProductsPath.OrdersPath,CustomerForm.CustomerName);

GenerateBillGrid.DataSource = CustomerDL.CustomerOrders1;

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

public void LoadForm()

{

CustomerDL.LoadOrders(ProductsPath.OrdersPath, CustomerForm.CustomerName);

OrdersByind();

}

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

{

try

{

ProductBL product = (ProductBL)GenerateBillGrid.CurrentRow.DataBoundItem;

CustomerDL.DeleteOrders(product);

ProductDL.StoreAllDataInfile(ProductsPath.OrdersPath, CustomerDL.CustomerOrders1);

OrdersByind();

}

catch(Exception exp)

{

MessageBox.Show(exp.Message);

}

}

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

{

sum = 0;

foreach(ProductBL Product in CustomerDL.CustomerOrders1)

{

sum = sum + Product.Prices;

}

Bills = sum;

GenerateBillTXT.Text = "PKR " + sum;

}

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

{

}

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

{

LoadForm();

}

}

**SigninForm:**

public partial class SignInForm : Form

{

public SignInForm()

{

InitializeComponent();

}

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

{

MuserDL.ReadData(UserId.UserIDPath);

MessageBox.Show("Successfuly Data Read");

}

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

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Exit!", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else if (dialogResult == DialogResult.No)

{

this.Activate();

}

}

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

{

if (UserNameTex1.Text != "User Name" && UserPasswordTex1.Text != "Password")

{

string UserName = UserNameTex1.Text;

string Password = UserPasswordTex1.Text;

Muser user = new Muser(UserName, Password);

user = MuserDL.Signin(user);

MessageText.Clear();

MessageText.Visible = true;

if (user != null)

{

if (user.IsAdmin())

{

AdminForm Admin = new AdminForm(user.UserName);

Admin.Show();

this.Hide();

}

else

{

CustomerForm Customer = new CustomerForm(user.UserName);

Customer.Show();

this.Hide();

}

}

else

{

MessageText.ForeColor = Color.Red;

MessageText.Text = "Invalid UserName And Password";

}

UserNameTex1.Clear();

UserPasswordTex1.Clear();

UserNameTex1.Focus();

}

else

{

MessageBox.Show("Please Enter Valid Input Format");

}

}

private void lblSignUP\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

SignUpForm form = new SignUpForm();

form.Show();

this.Hide();

}

**SignUp Form:**

public partial class SignUpForm : Form

{

public SignUpForm()

{

InitializeComponent();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

bool flag = false;

string UserName = UserNameTex2.Text;

string Password = UserPasswordtex2.Text;

string Role = RolecomboBox2.Text;

string PhoneNo = UserPhoneno.Text;

Muser user = new Muser(UserName, Password, Role,PhoneNo);

foreach (Muser s in MuserDL.User)

{

if (s.UserName == UserName && s.UserPassword == Password && s.UserRole == Role)

{

flag = true;

break;

}

}

if (flag == false)

{

if (UserNameTex2.Text == "User Name" || UserPasswordtex2.Text == "Password")

{

MessageBox.Show("Please Fill The Required Field");

}

else

{

MuserDL.AddUserTOList(user);

MuserDL.StoreinFile(user);

MessageBox.Show("Your are Successfully Registered Press OK to Continue!");

}

}

else

{

MessageBox.Show("Your are Not Eligible Try Another User Name Press OK to Continue!");

}

UserNameTex2.Clear();

UserPasswordtex2.Clear();

RolecomboBox2.Text = "Role";

UserNameTex2.Focus();

}

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

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

}

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

{

}

private void SignUpForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure? You Want To Close Form!", "Quit SignUp", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

SignInForm form = new SignInForm();

form.Show();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

this.Activate();

}

}

**Complete**

CustomerToList(customer);

}

else

{

UsersList.Add(user);

}

}

public static void deleteUserFromList(UserBL user)

{

for (int index = 0; index < usersList.Count; index++)

{

if (usersList[index].Name == user.Name && usersList[index].Pin == user.Pin)

{

usersList.RemoveAt(index);

}

}

}

public static List<UserBL> UsersList { get => usersList; set => usersList = value; }

#region file handling

public static void storeAllDataIntoFile(string path)

{

StreamWriter file = new StreamWriter(path);

foreach (UserBL storedUser in UsersList)

{

file.WriteLine(storedUser.Name + "," + storedUser.Pin + "," + storedUser.Role);

}

file.Flush();

file.Close();

}

public static bool readDataFromFile(string path)

{

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] splitRecord = record.Split(',');

string userName = splitRecord[0];

string userPassword = splitRecord[1];

string userRole = splitRecord[2];

UserBL user = new UserBL(userName, userPassword, userRole);

addUsertoList(user);

}

fileVariable.Close();

return true;

}

else

return false;

}

public static void storeUserIntoFile(UserBL user, string path)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(user.Name + "," + user.Pin + "," + user.Role);

file.Flush();

file.Close();

}

#endregion

}

class CustomerDL

{

private static List<CustomerBL> customerList = new List<CustomerBL>();

private static List<StockBL> orders = new List<StockBL>();

private static List<StockBL> productsBrought = new List<StockBL>();

public static void AddOrderInList(StockBL order)

{

Orders.Add(order);

}

public static bool checkCustomer(CustomerBL customer)

{

foreach (CustomerBL customer1 in CustomerList)

{

if (customer1.Name == customer.Name && customer1.Pin == customer.Pin)

{

return true;

}

}

return false;

}

public static void storeOrderInFile(StockBL order, string path)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(order.ItemName + "," + order.Quantity + "," + order.Price);

file.Flush();

file.Close();

}

public static bool readDataFromFile(string path)

{

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] splitRecord = record.Split(',');

string Name = splitRecord[0];

int quantity = int.Parse(splitRecord[1]);

double price = double.Parse(splitRecord[2]);

StockBL stock = new StockBL(Name, quantity, price);

StockDL.AddStock\_To\_List(stock);

}

fileVariable.Close();

return true;

}

else

return false;

}

public static List<CustomerBL> CustomerList { get => customerList; set => customerList = value; }

public static List<StockBL> Orders { get => orders; set => orders = value; }

public static List<StockBL> ProductsBrought { get => productsBrought; set => productsBrought = value; }

public static void AddCustomerToList(CustomerBL customer)

{

CustomerList.Add(customer);

}

public static double CalculateBill()

{

double sum = 0;

foreach (StockBL stock in ProductsBrought)

{

sum = sum + (stock.Price \* stock.Quantity);

}

return sum;

}

public static void UpdatePin(CustomerBL user, string newPin)

{

foreach (CustomerBL customer in CustomerList)

{

if (customer.Name == user.Name && customer.Pin == user.Pin)

{

customer.Pin = newPin;

}

}

foreach (UserBL userBL in UserDL.UsersList)

{

if (userBL.Name == user.Name && userBL.Pin == user.Pin)

{

userBL.Pin = newPin;

}

}

}

public static void UpdateBill(CustomerBL user)

{

foreach (CustomerBL customer in CustomerList)

{

if (customer.Name == user.Name && customer.Pin == user.Pin)

{

customer.Bill = user.Bill;

}

}

}

}

class S\_BillDL

{

private static List<S\_BillBL> shopBill\_List = new List<S\_BillBL>();

public static List<S\_BillBL> ShopBill\_List { get => shopBill\_List; set => shopBill\_List = value; }

public static void AddShopBill\_To\_List(S\_BillBL bill)

{

ShopBill\_List.Add(bill);

}

#region FILE HANDLING

public static void storeAllDataIntoFile(string path)

{

StreamWriter file = new StreamWriter(path, false);

foreach (S\_BillBL bill in ShopBill\_List)

{

file.WriteLine(bill.BillName + "," + bill.BillAmount);

}

file.Flush();

file.Close();

}

public static bool readDataFromFile(string path)

{

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] splitRecord = record.Split(',');

string Name = splitRecord[0];

double Amount = double.Parse(splitRecord[1]);

S\_BillBL bill = new S\_BillBL(Name, Amount);

AddShopBill\_To\_List(bill);

}

fileVariable.Close();

return true;

}

else

return false;

}

public static void storeBillIntoFile(S\_BillBL bill, string path)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(bill.BillName + "," + bill.BillAmount);

file.Flush();

file.Close();

}

#endregion

public static void EditBillFromList(S\_BillBL previous, S\_BillBL updated)

{

foreach (S\_BillBL bill in ShopBill\_List)

{

if (bill.BillName == previous.BillName && bill.BillAmount == previous.BillAmount)

{

previous.BillName = updated.BillName;

previous.BillAmount = updated.BillAmount;

}

}

}

}

class StockDL

{

public static List<StockBL> stocks\_List = new List<StockBL>();

public static List<StockBL> Stocks\_List { get => stocks\_List; set => stocks\_List = value; }

public static void AddStock\_To\_List(StockBL item)

{

Stocks\_List.Add(item);

}

public static void EditStockFromList(StockBL previous, StockBL updated)

{

foreach (StockBL stock in Stocks\_List)

{

if (stock.ItemName == previous.ItemName && stock.Quantity == previous.Quantity)

{

previous.ItemName = updated.ItemName;

previous.Quantity = updated.Quantity;

}

}

}

#region file handling

public static void storeAllDataIntoFile(string path)

{

StreamWriter file = new StreamWriter(path, false);

foreach (StockBL Item in Stocks\_List)

{

file.WriteLine(Item.ItemName + "," + Item.Quantity + "," + Item.Price);

}

file.Flush();

file.Close();

}

public static void storeStockIntoFile(StockBL stock, string path)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(stock.ItemName + "," + stock.Quantity + "," + stock.Price);

file.Flush();

file.Close();

}

public static bool readDataFromFile(string path)

{

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] splitRecord = record.Split(',');

string Name = splitRecord[0];

int quantity = int.Parse(splitRecord[1]);

double price = double.Parse(splitRecord[2]);

StockBL stock = new StockBL(Name, quantity, price);

AddStock\_To\_List(stock);

}

fileVariable.Close();

return true;

}

else

return false;

}

#endregion

}

public partial class MainForm : Form

{

//load in next form

public string pathUser = "Data.txt";

public MainForm()

{

InitializeComponent();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

Form form = new SignIn();

form.Visible = true;

this.Visible = false;

}

private void MainForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

private void clearButtons()

{

userName\_txt.Clear();

Pasword\_txt.Clear();

roleCombox.Text = "";

}

private void SignUp\_btn\_Click(object sender, EventArgs e)

{

if ((userName\_txt.Text != string.Empty || userName\_txt.Text != " ") && (Pasword\_txt.Text != string.Empty) && (roleCombox.SelectedValue != null))

{

string userName = userName\_txt.Text;

string pin = Pasword\_txt.Text;

string role = roleCombox.Text;

UserBL user = new UserBL(userName, pin, role);

UserDL.addUsertoList(user);

UserDL.storeUserIntoFile(user, pathUser);

MessageBox.Show("User added succesfully", "Messege", MessageBoxButtons.OK, MessageBoxIcon.Information);

clearButtons();

}

else

{

MessageBox.Show("Information is missing Complete the Information", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

Pasword\_txt.UseSystemPasswordChar = true;

}

else

{

Pasword\_txt.UseSystemPasswordChar = false;

}

}

}

public partial class AdminMenu : Form

{

public AdminMenu()

{

InitializeComponent();

}

#region(side buttons)

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

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Form form = new SignIn();

form.Visible = true;

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

if (!Visible)

{

Visible = true;

}

}

}

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

{

Process.Start("http://www.google.com");

}

#endregion

#region closing

private void AdminMenu\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Environment.Exit(1);

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

this.Activate();

}

}

#endregion

private void customerSection\_btn\_Click(object sender, EventArgs e)

{

Form form = new Customer\_Section();

form.ShowDialog();

}

private void stockSection\_btn\_Click(object sender, EventArgs e)

{

Form form = new StockSection();

form.ShowDialog();

}

private void ShopBill\_btn\_Click(object sender, EventArgs e)

{

Form form = new ShopBill();

form.ShowDialog();

}

private void CustomerBill\_btn\_Click(object sender, EventArgs e)

{

Form form = new CustomersBills();

form.ShowDialog();

}

private void Order\_btn\_Click(object sender, EventArgs e)

{

Form form = new Orders();

form.ShowDialog();

}

}

public partial class CustomerMenu : Form

{

private UserBL user;

private string path = "Data.txt";

public CustomerMenu(UserBL user)

{

InitializeComponent();

this.user = user;

}

private void help\_btn\_Click(object sender, EventArgs e)

{

Process.Start("http://www.google.com");

}

private void CustomerMenu\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

CustomerDL.ProductsBrought.Clear();

Environment.Exit(1);

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

this.Activate();

}

}

private void Exit\_btn\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

CustomerDL.ProductsBrought.Clear();

Form form = new SignIn();

form.Visible = true;

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

if (!Visible)

{

Visible = true;

}

}

}

private void ShopProduct\_btn\_Click(object sender, EventArgs e)

{

Form form = new ShopProducts();

form.ShowDialog();

}

private void viewCart\_btn\_Click(object sender, EventArgs e)

{

Form form = new ViewCart(user);

form.ShowDialog();

}

private void ChangePasword\_Btn\_Click(object sender, EventArgs e)

{

Form form = new ChangePasword(user);

form.ShowDialog();

UserDL.storeAllDataIntoFile(path);

}

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

{

Form form = new CustomerOrder();

form.ShowDialog();

}

}

public partial class Customer\_Section : Form

{

public string path = "Data.txt";

public Customer\_Section()

{

InitializeComponent();

}

public void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.CustomerList.Select(c => new { c.Name, c.Pin }).ToList();

dataGridView1.Refresh();

}

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

{

if (CustomerDL.CustomerList.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

UserBL user = (UserBL)dataGridView1.CurrentRow.DataBoundItem;

DialogResult dialogResult = MessageBox.Show("Are You Sure You Want to Delete ?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

CustomerDL.CustomerList.RemoveAt(rowIndex);

UserDL.deleteUserFromList(user);

UserDL.storeAllDataIntoFile(path);

dataBind();

}

else if (dialogResult == DialogResult.No)

{

CancelEventArgs cancelEvent = new CancelEventArgs();

cancelEvent.Cancel = true;

Activate();

}

}

else

{

DialogResult dialogResult = MessageBox.Show("No data to delete", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void Customer\_Section\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

private void Customer\_Section\_Load(object sender, EventArgs e)

{

if (dataGridView1.DataSource == null)

{

dataBind();

}

}

}

public partial class StockSection : Form

{

public string path = "Stock.txt";

public StockSection()

{

InitializeComponent();

}

public void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = StockDL.Stocks\_List;

dataGridView1.Refresh();

}

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

{

dataBind();

}

#region buttons

private void Edit\_btn\_Click(object sender, EventArgs e)

{

if (StockDL.Stocks\_List.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

Form form = new EditStock(StockDL.Stocks\_List[rowIndex]);

form.ShowDialog();

StockDL.storeAllDataIntoFile(path);

dataBind();

}

else

{

MessageBox.Show("No data to Edit", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void Delete\_btn\_Click(object sender, EventArgs e)

{

if (StockDL.Stocks\_List.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

DialogResult dialogResult = MessageBox.Show("Are You Sure You Want to Delete ?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

StockDL.Stocks\_List.RemoveAt(rowIndex);

StockDL.storeAllDataIntoFile(path);

dataBind();

}

else if (dialogResult == DialogResult.No)

{

CancelEventArgs cancelEvent = new CancelEventArgs();

cancelEvent.Cancel = true;

Activate();

}

}

else

{

MessageBox.Show("No data to delete", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void New\_btn\_Click(object sender, EventArgs e)

{

Form form = new NewStock();

form.ShowDialog();

StockDL.storeAllDataIntoFile(path);

dataBind();

}

#endregion

private void StockSection\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class ShopBill : Form

{

private string path = "ShopBill.txt";

public ShopBill()

{

InitializeComponent();

}

public void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = S\_BillDL.ShopBill\_List;

dataGridView1.Refresh();

}

#region BUTTONS

private void New\_btn\_Click(object sender, EventArgs e)

{

Form form = new NewBill();

form.ShowDialog();

S\_BillDL.storeAllDataIntoFile(path);

dataBind();

}

private void Edit\_btn\_Click(object sender, EventArgs e)

{

if (S\_BillDL.ShopBill\_List.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

Form form = new EditBill(S\_BillDL.ShopBill\_List[rowIndex]);

form.ShowDialog();

S\_BillDL.storeAllDataIntoFile(path);

dataBind();

}

else

{

MessageBox.Show("No data to Edit", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void Delete\_btn\_Click(object sender, EventArgs e)

{

if (S\_BillDL.ShopBill\_List.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

DialogResult dialogResult = MessageBox.Show("Are You Sure You Want to Delete ?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

S\_BillDL.ShopBill\_List.RemoveAt(rowIndex);

S\_BillDL.storeAllDataIntoFile(path);

dataBind();

}

else if (dialogResult == DialogResult.No)

{

CancelEventArgs cancelEvent = new CancelEventArgs();

cancelEvent.Cancel = true;

Activate();

}

}

else

{

MessageBox.Show("No data to delete", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

#endregion

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

{

dataBind();

}

private void ShopBill\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class CustomersBills : Form

{

public CustomersBills()

{

InitializeComponent();

}

private void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.CustomerList.Select(c => new { c.Name, c.Bill }).ToList();

dataGridView1.Refresh();

}

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

{

dataBind();

}

private void CustomersBills\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class Orders : Form

{

public Orders()

{

InitializeComponent();

}

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

{

dataBind();

}

private void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.Orders.Select(c => new { c.ItemName, c.Quantity }).ToList();

dataGridView1.Refresh();

}

private void Orders\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class ShopProducts : Form

{

public ShopProducts()

{

InitializeComponent();

dataBind();

}

private void dataBind()

{

shopGrid.DataSource = null;

shopGrid.DataSource = StockDL.stocks\_List;

shopGrid.Refresh();

}

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

{

int rowIndex = shopGrid.CurrentCell.RowIndex;

CustomerDL.ProductsBrought.Add(StockDL.Stocks\_List[rowIndex]);

MessageBox.Show("Added to Cart", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void ShopProducts\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class ViewCart : Form

{

private UserBL user;

public ViewCart(UserBL user)

{

InitializeComponent();

this.user = user;

}

private void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.ProductsBrought;

dataGridView1.Refresh();

}

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

{

dataBind();

}

private void CalculateBill\_btn\_Click(object sender, EventArgs e)

{

if (CustomerDL.ProductsBrought.Count != 0)

{

MessageBox.Show("Your Total Bill :" + CustomerDL.CalculateBill(), "Bill", MessageBoxButtons.OK, MessageBoxIcon.Information);

CustomerBL customer = new CustomerBL(user.Name, user.Pin, user.Role);

customer.Bill = CustomerDL.CalculateBill();

CustomerDL.UpdateBill(customer);

}

else

{

MessageBox.Show("There is no item In Cart", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Delete\_Btn\_Click(object sender, EventArgs e)

{

if (CustomerDL.ProductsBrought.Count != 0)

{

int rowIndex = dataGridView1.CurrentCell.RowIndex;

CustomerDL.ProductsBrought.RemoveAt(rowIndex);

dataBind();

}

else

{

MessageBox.Show("There is no item In Cart To delete", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ViewCart\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class ChangePasword : Form

{

private UserBL User;

public ChangePasword(UserBL User)

{

InitializeComponent();

this.User = User;

}

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

{

Name\_txt.Text = User.Name;

}

private void Confirm\_btn\_Click(object sender, EventArgs e)

{

if (Name\_txt.Text != string.Empty || oldpin\_txt.Text != string.Empty || newPin\_txt.Text != string.Empty)

{

CustomerBL customer = new CustomerBL(User.Name, User.Pin, User.Role);

bool isTrue = CustomerDL.checkCustomer(customer);

if (isTrue)

{

CustomerDL.UpdatePin(customer, newPin\_txt.Text);

MessageBox.Show("Password Updated Successfully!", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Invalid user ", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("Information is missing Or Invalid", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void ChangePasword\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class CustomerOrder : Form

{

public CustomerOrder()

{

InitializeComponent();

}

private void clearForm()

{

textBox1.Clear();

numericUpDown1.Value = 1;

}

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

{

int quantity = (int)numericUpDown1.Value;

StockBL order = new StockBL(textBox1.Text, quantity);

CustomerDL.AddOrderInList(order);

DialogResult dialogResult = MessageBox.Show("Order is being Placed.Do you want to place more Orders?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

clearForm();

}

else if (dialogResult == DialogResult.No)

{

Visible = false;

}

}

}

public partial class EditStock : Form

{

private StockBL previous;

public EditStock(StockBL previous)

{

InitializeComponent();

this.previous = previous;

}

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

{

textBox1.Text = previous.ItemName;

numericUpDown1.Value = previous.Quantity;

numericUpDown2.Value = (decimal)previous.Price;

}

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

{

if (textBox1.Text != string.Empty || textBox1.Text != " ")

{

int quantity = (int)numericUpDown1.Value;

double price = (double)numericUpDown2.Value;

StockBL updated = new StockBL(textBox1.Text, quantity, price);

StockDL.EditStockFromList(previous, updated);

Close();

}

else

{

MessageBox.Show("Information is missing Or Invalid", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

public partial class NewStock : Form

{

public NewStock()

{

InitializeComponent();

}

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

{

if (textBox1.Text != string.Empty || textBox1.Text != " ")

{

int quantity = (int)numericUpDown1.Value;

double price = (double)numericUpDown2.Value;

StockBL stock = new StockBL(textBox1.Text, quantity, price);

StockDL.AddStock\_To\_List(stock);

MessageBox.Show("Stock added successfully!", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

Close();

}

else

{

MessageBox.Show("Information is missing ", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

public partial class EditBill : Form

{

private S\_BillBL previous;

public EditBill(S\_BillBL previous)

{

InitializeComponent();

this.previous = previous;

}

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

{

textBox1.Text = previous.BillName;

numericUpDown1.Value = (decimal)previous.BillAmount;

}

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

{

if (textBox1.Text != string.Empty || textBox1.Text != " ")

{

int quantity = (int)numericUpDown1.Value;

S\_BillBL updated = new S\_BillBL(textBox1.Text, quantity);

S\_BillDL.EditBillFromList(previous, updated);

Close();

}

else

{

MessageBox.Show("Information is missing or Invalid", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void EditBill\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Are you Sure?", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Visible = false;

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

Activate();

}

}

}

public partial class NewBill : Form

{

public NewBill()

{

InitializeComponent();

}

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

{

if (textBox1.Text != string.Empty || textBox1.Text != " ")

{

int amount = (int)numericUpDown1.Value;

S\_BillBL bill = new S\_BillBL(textBox1.Text, amount);

S\_BillDL.AddShopBill\_To\_List(bill);

MessageBox.Show("Bill added successfully!", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

Close();

}

else

{

MessageBox.Show("Information is missing Or Invalid", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}