

Criterion C: Development

The product that's developed is a C# product. It is an inventory management system that has been developed for the client to store various product information for his general store. The program allows the client to add, update, delete, view and search products. The client can also use the program keep track of sales. This program is protected by a login screen that requires simply the user's username and password to access product and sales information.

Libraries Used:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
```

Code 1: Libraries Used

Techniques Used:

- Microsoft Visual Studio IDE has been used.
- The program uses SQL (Structured Query Language) to store product data. SQL Server Management Studio is used to manage and operate the SQL server used for the product.
- .Net framework in C# has been used to develop the application.
- System.drawing NuGet graphic packages are used for the graphical components.
- Static variables and methods are used

Login Page:

A screenshot of a login window titled "User Log in" for "SAFEWAY GENERAL STORE". The window has a light blue background. It contains two input fields: "User Name :" and "Password :". Below the password field is a "Log In" button.

Image 1: Login Page

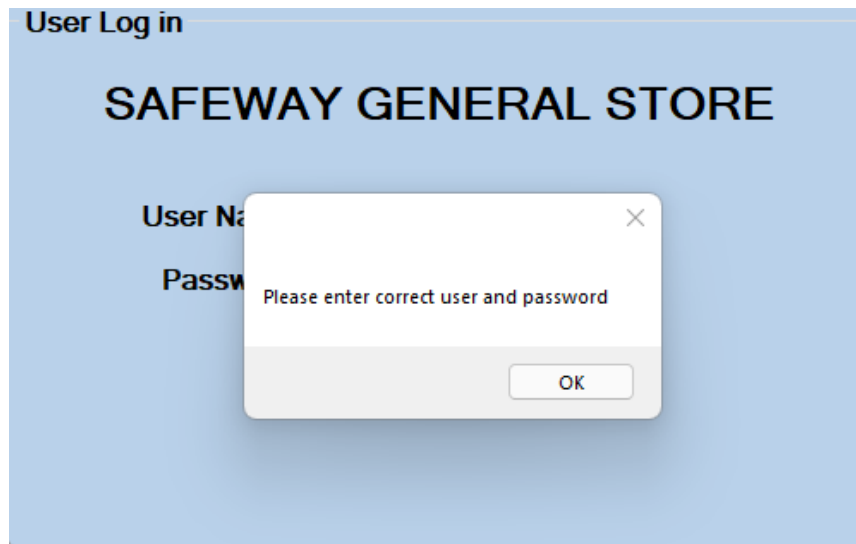


Image 2: Login Error Message

When the user runs the program, a login window will appear. This window requests the user to input their 'username' and 'password'. If the user enters the wrong information, the program will not grant access to the program. This allows the software to be protected from outside interference as any unauthorized user cannot access the program and gain confidential product information.

```

20 private void btnlogin_Click(object sender, EventArgs e)
21 {
22     if(txtpassword.Text != "" && txtuserName.Text!="")
23     {
24         SecUser user=new SecUser();
25         DBHandler repo = new DBHandler();
26
27         user.Password = txtpassword.Text;
28         user.Username = txtuserName.Text;
29         if(repo.Login(user))
30         {
31             // Main mainform=new Main();
32             MainMenu mainform=new MainMenu();
33
34             mainform.Show();
35             this.Hide();
36         }
37     }
38     else
39     {
40         MessageBox.Show("Please enter correct user and password");
41     }
42 }
43
44 else
45 {
46     MessageBox.Show("Please enter user and password");
47 }
48 }
49
50 1 reference
51 private void btnreg_Click(object sender, EventArgs e)
52 {
53     frmNewUser frmNewUserchild = new frmNewUser();
54     // Set the Parent Form of the Child window.
55     //frmCheckoutchild.MdiParent = this;
56     // Display the new form.
57     frmNewUserchild.Show();
58 }
59 //private void txtpassword_TextChanged(object sender, EventArgs e)
60 //{
61 // }
62
63 1 reference
64 private void txtpassword_TextChanged_1(object sender, EventArgs e)
65 {
66     if (txtpassword.Text != "")
67     {
68         txtpassword.UseSystemPasswordChar = true;
69     }
70     else
71     {
72         txtpassword.UseSystemPasswordChar = false;
73     }
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173 1 reference
174 public bool Login(SecUser parameter)
175 {
176     using (SqlConnection con = new SqlConnection(conString))
177     {
178         byte[] byteArray = new byte[1024];
179         using (SqlCommand cmd = new SqlCommand("spLogin", con))
180         {
181             con.Open();
182             cmd.CommandType = CommandType.StoredProcedure;
183
184             cmd.Parameters.Add("@pUserID", SqlDbType.NVarChar, 50).Value = parameter.Username;// parameter.Name;
185             cmd.Parameters.Add("@pPassword", SqlDbType.NVarChar, 50).Value = parameter.Password;//parameter.UserID;
186
187             SqlDataAdapter da = new SqlDataAdapter(cmd);
188             DataTable dt = new DataTable();
189             da.Fill(dt);
190
191             foreach (DataRow dr in dt.Rows)
192             {
193                 if (dr["UserName"] != null)
194                 {
195                     if (parameter.Username == (string)dr["UserName"] && parameter.Password == (string)dr["Password"])
196                         return true;
197                 }
198             }
199
200             return false;
201         }
202     }
203 }
204
205
206
207
208
209

```

Code 2: Login Page codes along with verification

Main Window:

After entering the correct username and password, the program will open the main window.

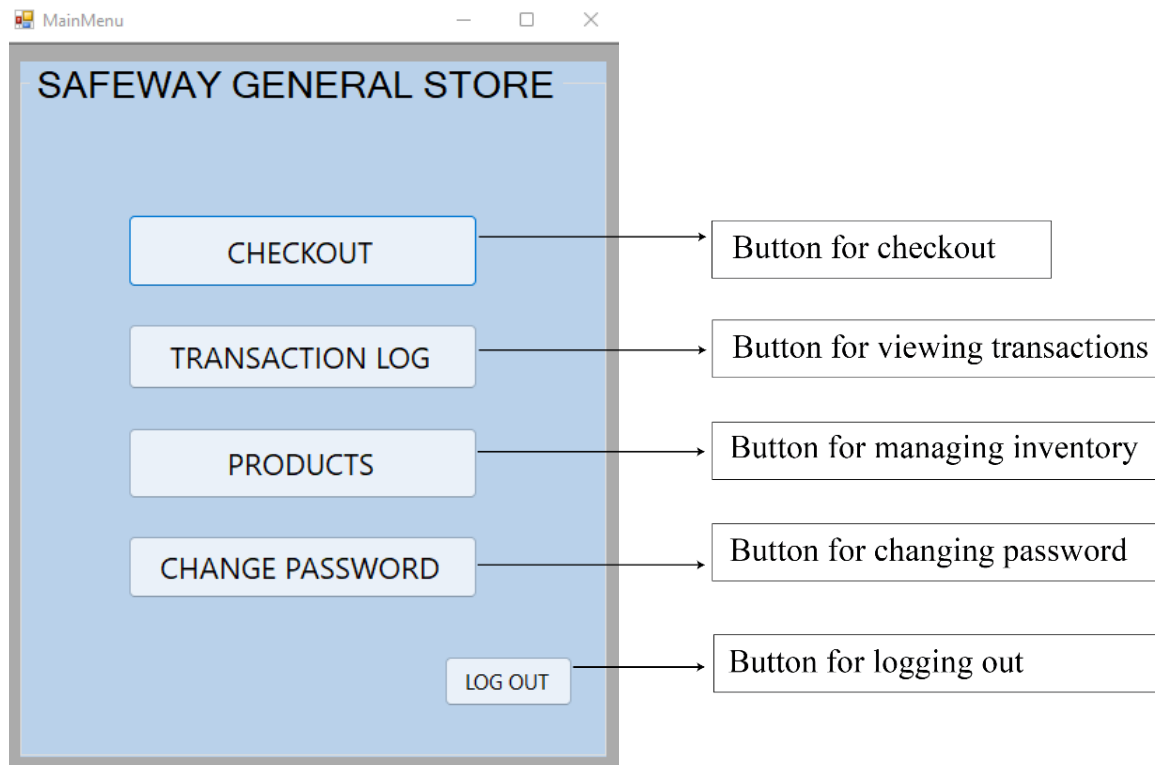


Image 3: Main Page

The main window consists of 4 main categories, Checkout, Transaction Log, Products and Change Password.

Product Window:

Upon clicking on the 'Product' button from the main window, the program directs the user to the program window.

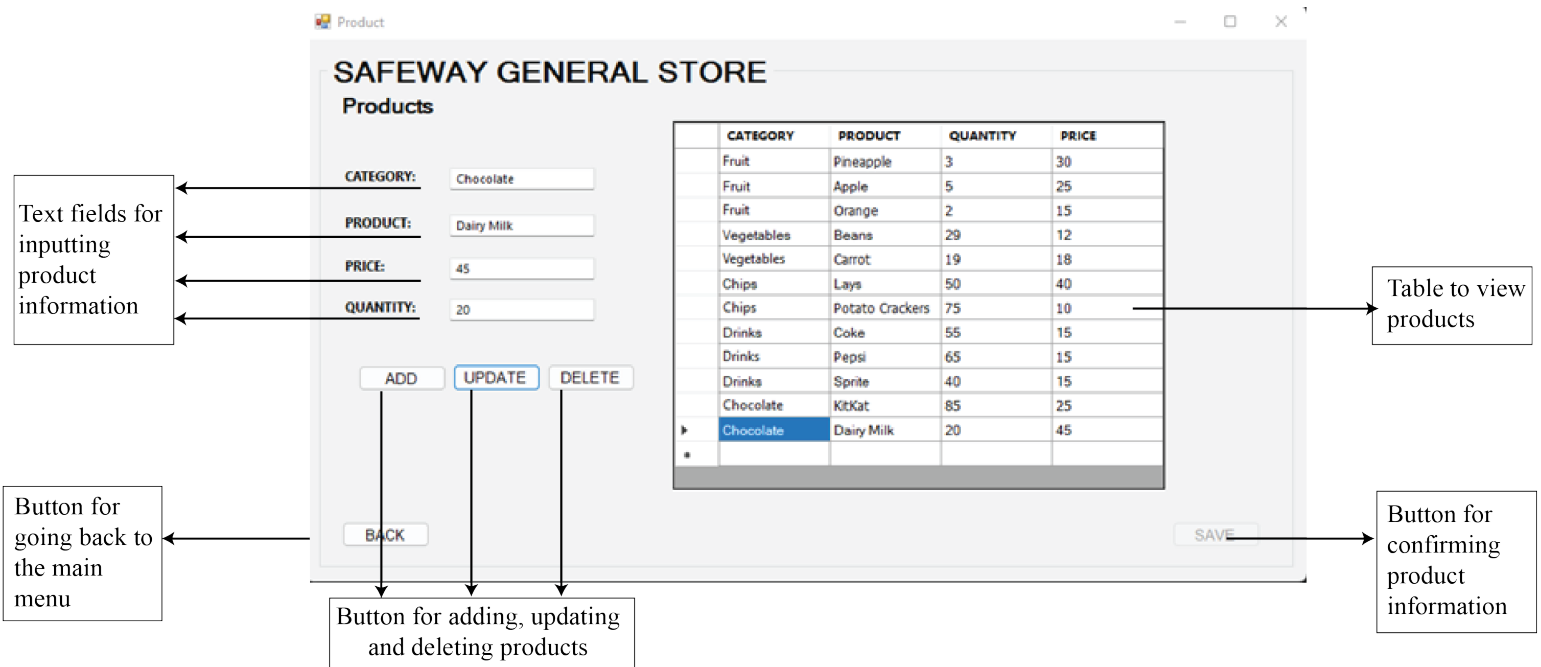


Image 4: Product Page

The product window allows the user to add, update and delete product data. This page includes text fields for entering data and buttons for adding the product into the program, making the program minimalistic and user friendly.

Adding product data into the database:

The C# code below were used to add the product information into the SQL database with the help of Microsoft SQL Server Management Studio.

```
23 private void GridLayout1_Click(object sender, EventArgs e)
24 {
25     //add the product
26     ProductObject prod = new ProductObject();
27
28     if(checkproduct())
29     {
30         prod.ProductCategory = txtcat.Text;
31         prod.ProductName = txtprod.Text;
32         prod.ProductPrice = float.Parse(txtprice.Text); // if show the price the data type will be float
33         prod.ProductQuantity = int.Parse(txtquantity.Text);
34
35         DataGridViewRow newRow = new DataGridViewRow();
36         newRow.CreateCells(datagrdProduct);
37
38         newRow.Cells[0].Value = prod.ProductCategory;
39         newRow.Cells[1].Value = prod.ProductName;
40         newRow.Cells[2].Value = prod.ProductQuantity;
41         newRow.Cells[3].Value = prod.ProductPrice;
42
43         datagrdProduct.Rows.Add(newRow);
44     }
45
46     btnsave.Enabled = true;
47 }
48
49 }
```

Code 2: Button for adding data into database

```
52 public bool AddProduct(ProductObject prodparameter)
53 {
54     try
55     {
56         //SqlCommand sqlCom = new SqlCommand("select * from dms_DeviceLog where ID = (select MAX(ID) from dms_DeviceLog)");
57         //conn.Open();
58         //query = "spInsertTemplate";
59
60         using (SqlConnection con = new SqlConnection(conString))
61         {
62             using (SqlCommand cmd = new SqlCommand("spAddProduct", con))
63             {
64                 con.Open();
65                 cmd.CommandType = CommandType.StoredProcedure;
66
67                 cmd.Parameters.Add("@pCategory", SqlDbType.NVarChar, 50).Value = prodparameter.ProductCategory;
68                 cmd.Parameters.Add("@pName", SqlDbType.NVarChar, 50).Value = prodparameter.ProductName;
69                 cmd.Parameters.Add("@pPrice", SqlDbType.Float).Value = prodparameter.ProductPrice;
70                 cmd.Parameters.Add("@pQuantity", SqlDbType.Int).Value = prodparameter.ProductQuantity;
71
72                 cmd.ExecuteNonQuery();
73
74                 return true;
75             }
76         }
77     }
78     catch (Exception ex)
79     {
80         return false;
81     }
82 }
83
84 }
```

Code 3: Adding data into database ¹

¹ "How to insert data to a database in C#," Stack Overflow, February 2022, <https://stackoverflow.com/questions/57448296/how-to-insert-data-to-a-database-in-c-sharp>

User can use the text fields on the side to input product information. If the data is validated, the user can click on the 'add button' to add the product to the database. Upon successful addition of product into the program, the product will be added to the table on the right-hand side.

GENERAL STORE

DELETE

	CATEGORY	PRODUCT	QUANTITY	PRICE
	Fruit	Pineapple	3	30
	Fruit	Apple	5	25
	Fruit	Orange	2	15
	Vegetables	Beans	29	12
	Vegetables	Carrot	19	18
	Chips	Lays	50	40
	Chips	Potato Crackers	75	10
	Drinks	Coke	55	15
	Drinks	Pepsi	65	15
	Drinks	Sprite	40	15
	Chocolate	KitKat	85	25
	Chocolate	Dairy Milk	20	45
▶	Chocolate	Toblerone	12	75

Image 5: Product Table View

However, in order to prevent incorrect data type to be inputted into the corresponding product information fields, there are various validations used in this part of the program. If the wrong input is given, error messages will pop up.

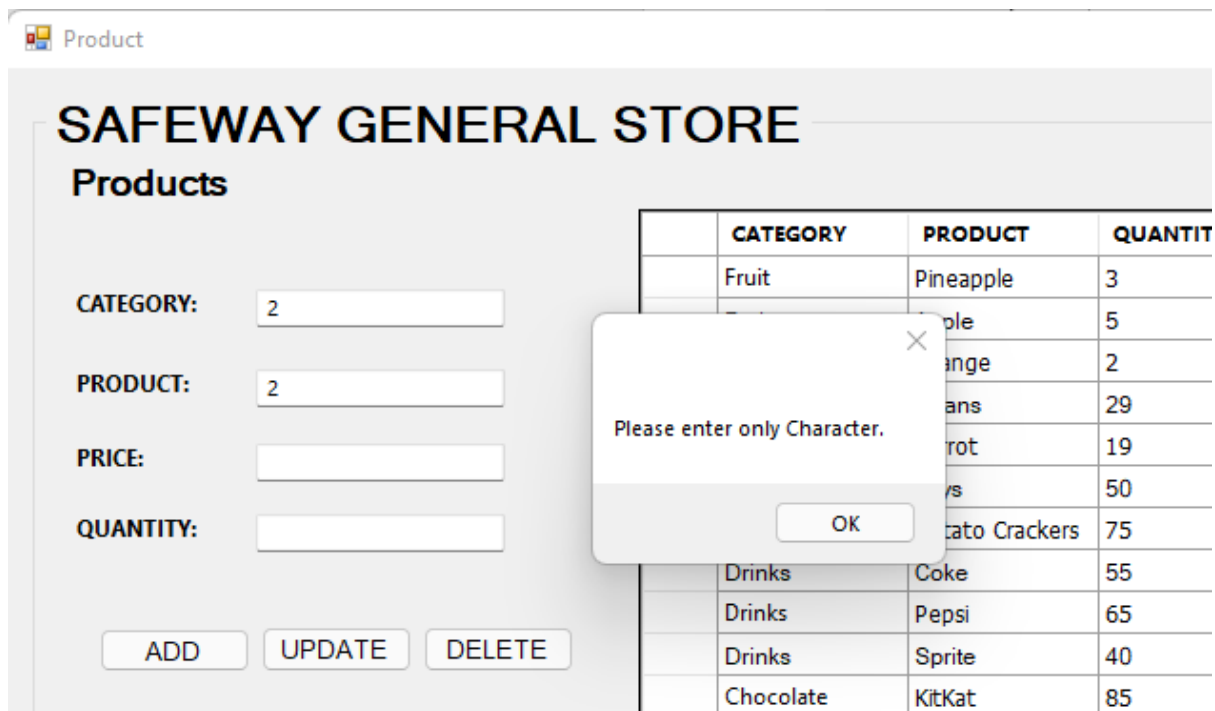


Image 6: Error message for wrong input

```

232 | 1 reference
233 | private void txtcat_TextChanged(object sender, EventArgs e)
234 | {
235 |     if (!System.Text.RegularExpressions.Regex.IsMatch(txtcat.Text, "^[a-zA-Z ]*$"))
236 |     {
237 |         MessageBox.Show("Please enter only Character.");
238 |         txtcat.Text = txtcat.Text.Remove(txtcat.Text.Length - 1);
239 |     }
240 | }
241 |
242 | 1 reference
243 | private void txtprod_TextChanged(object sender, EventArgs e)
244 | {
245 |     if (!System.Text.RegularExpressions.Regex.IsMatch(txtprod.Text, "^[a-zA-Z ]*$"))
246 |     {
247 |         MessageBox.Show("Please enter only Character.");
248 |         txtprod.Text = txtprod.Text.Remove(txtcat.Text.Length - 1);
249 |     }

```

Code 4: Validation ²

The image above is the code used to validate the ‘Category’ and ‘Product’ text field. This validation will only allow letters, white space and backspace to be entered, if any other data type such as integers are inputted, the error message will pop up and the product will not be added. If any text fields are empty spaces, an error would appear as well.

²“Validation of numbers and letters using C# for a beginner’,” Stack Overflow, February 2022, <https://stackoverflow.com/questions/8867097/validation-of-numbers-and-letters-using-c-sharp-for-a-beginner>


```

142 2 references
143 private bool checkproduct()
144 {
145     if (txtcat.Text == "")
146     {
147         if (txtprod.Text == "")
148         {
149             MessageBox.Show("Please Insert the Product Name ");
150         }
151     }
152
153     if (txtprice.Text == "")
154     {
155         MessageBox.Show("Please Insert the price");
156         return false;
157     }
158
159     if (txtquantity.Text == "")
160     {
161         MessageBox.Show("Please Insert the Quantity");
162         return false;
163     }
164
165     return true;
166 }
167
168
169
170
171
172
173

```

Code 5: Validation code for empty spaces in text fields

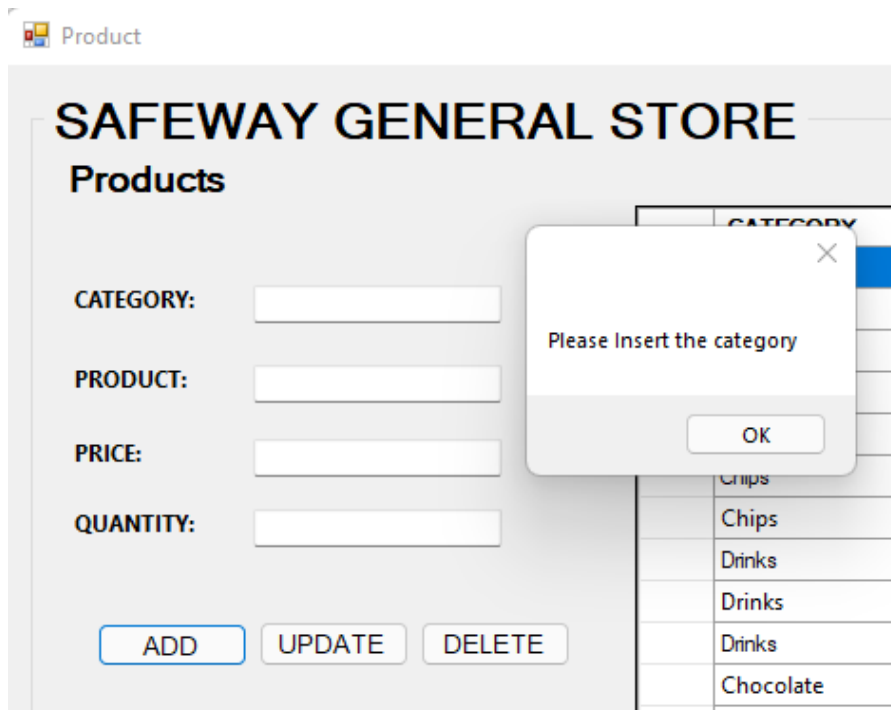


Image 7: Error message for blank text fields

Updating existing product data into the database:

If the user wants to edit existing product data, he simply needs to click on the product from the table that is displayed. The text fields will automatically be filled with the existing product information. The user can simply edit the necessary product information and click on the ‘update’ button. Once the data input is validated and saved into the database, a ‘saved successfully’ message will pop up.

```
191 private void btnupdate_Click(object sender, EventArgs e) // if you want to update the product
192 // then select the item from the product list the update it from the txt box
193 {
194     ProductObject prod = new ProductObject();
195     if (checkproduct())
196     {
197         prod.ProductCategory = txtcat.Text;
198         prod.ProductName = txtprod.Text;
199         prod.ProductPrice = float.Parse(txtprice.Text);
200         prod.ProductQuantity = int.Parse(txtquantity.Text);
201
202         if(repo.UpdateProductDetails(prod))
203         {
204             MessageBox.Show(prod.ProductName + " Update Successful");
205             DisplayData();
206             textClear();
207         }
208     }
209 }
210
211
212
213
214
215
216
```

Code 6: Button for updating data into database

```
361 public bool UpdateProductDetails(ProductObject prodparameter)
362 {
363     try
364     {
365         using (SqlConnection con = new SqlConnection(conString))
366         {
367             using (SqlCommand cmd = new SqlCommand("sp_UpdateProduct", con))
368             {
369                 con.Open();
370                 cmd.CommandType = CommandType.StoredProcedure;
371
372                 cmd.Parameters.Add("@pCategory", SqlDbType.NVarChar, 50).Value = prodparameter.ProductCategory;
373                 cmd.Parameters.Add("@pName", SqlDbType.NVarChar, 50).Value = prodparameter.ProductName;
374                 cmd.Parameters.Add("@pPrice", SqlDbType.Float).Value = prodparameter.ProductPrice;
375                 cmd.Parameters.Add("@pQuantity", SqlDbType.Int).Value = prodparameter.ProductQuantity;
376
377                 cmd.ExecuteNonQuery();
378
379                 return true;
380             }
381         }
382     }
383     catch (Exception ex)
384     {
385         return false;
386     }
387 }
388
389
390
391
392
393
```

Code 7: Updating data into database ³

³ Ehtesham Mehmood, "Insert, Update, Delete, Display Data in Mysql Using C#." C# Corner, June 3, 2020. <https://www.c-sharpcorner.com/UploadFile/9582c9/insert-update-delete-display-data-in-mysql-using-C-Sharp/>.

SAFEWAY GENERAL STORE

Products

CATEGORY:

PRODUCT:

PRICE:

QUANTITY:

	CATEGORY	PRODUCT
	Fruit	Pineapple
	Fruit	Apple
	Fruit	Orange
	Vegetables	Beans
		Carrot
		Peas
		Potato Crackers
		Coke
		psi
	Drinks	Sprite
	Chocolate	KitKat
	Chocolate	Dairy Milk

Toblerone Update Successful

Image 8: Success message for updating product

Deleting a product data from the database:

If the user wants to edit delete a certain product, he simply needs to select on the product from the table and click on the delete button. If the product is deleted successfully from the database, a 'deleted successfully' message will pop up.

```

102     1 reference
103     private void btnDelete_Click(object sender, EventArgs e) // select the item and delete it
104     {
105         ProductObject prodObj = new ProductObject();
106         prodObj.ProductCategory = txtcat.Text;
107         prodObj.ProductPrice = float.Parse(txtprice.Text);
108         prodObj.ProductName = txtprod.Text;
109         prodObj.ProductQuantity = int.Parse(txtquantity.Text);
110
111
112         if (repo.deleteProduct(prodObj))
113         {
114             MessageBox.Show("Deleted successfully");
115             DisplayData();
116             textClear();
117         }
118         else
119         {
120             MessageBox.Show("Not successfully deleted");
121         }
122     }
123

```

Code 8: Button for deleting data into database

```

394 1 reference
395 public bool deleteProduct(ProductObject prodObj)
396 {
397     try
398     {
399         //SqlCommand sqlCom = new SqlCommand("select * from dms_DeviceLog where ID = (select MAX(ID) from dms_DeviceLog)");
400         //conn.Open();
401         //query = "spInsertTemplate";
402
403         using (SqlConnection con = new SqlConnection(conString))
404         {
405             using (SqlCommand cmd = new SqlCommand("sp_deleteProduct", con))
406             {
407                 con.Open();
408                 cmd.CommandType = CommandType.StoredProcedure;
409
410                 cmd.Parameters.Add("@pProductName", SqlDbType.NVarChar, 50).Value = prodObj.ProductName;
411
412                 cmd.ExecuteNonQuery();
413                 return true;
414             }
415         }
416     }
417     catch (Exception ex)
418     {
419         return false;
420     }
421 }
422
423
424
425
426
427
428

```

Code 9: Deleting data from database

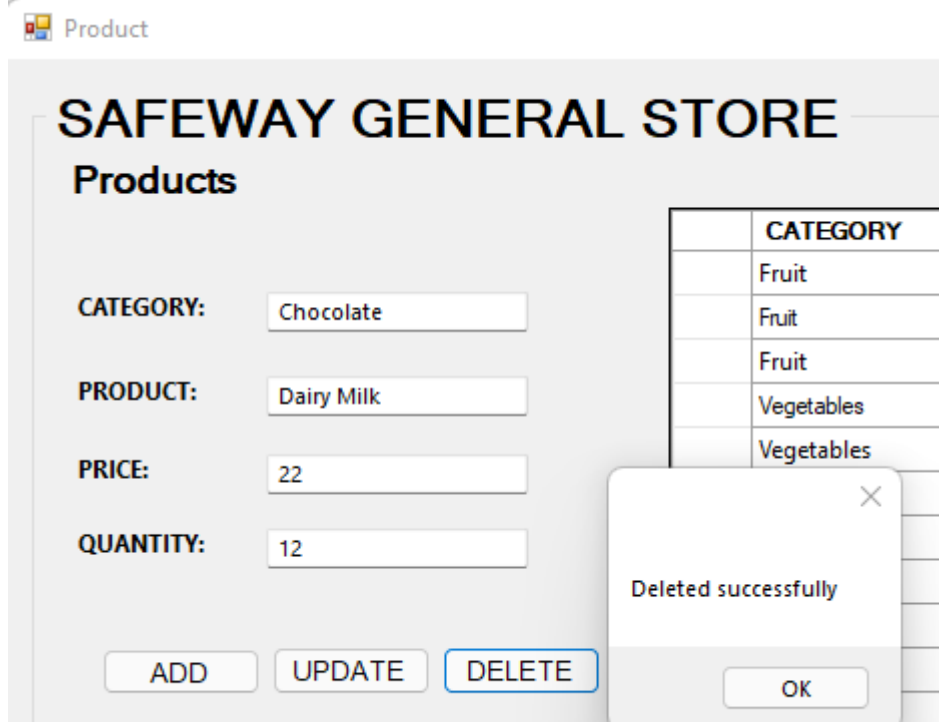


Image 9: Product deleted success message ⁴

Once the user made the necessary changes to the products, the user has to click on the ‘save’ button on the bottom right corner. The user then can click on the back button to return to the main window.

⁴ Ehtesham Mehmood, “Insert, Update, Delete, Display Data in Mysql Using C#.” C# Corner, June 3, 2020. <https://www.c-sharpcorner.com/UploadFile/9582c9/insert-update-delete-display-data-in-mysql-using-C-Sharp/>.

Checkout Window:

The user can use the checkout window in order to record a sale of a product. Upon clicking the checkout button, the window below will be opened.

The screenshot shows the 'CHECKOUT' window for 'SAFEGWAY GENERAL STORE'. It includes a 'NO:' field, 'PRODUCT DETAILS' with a 'CATEGORY' dropdown and a 'Search' button, an 'ITEMS' table with columns 'NAME', 'PRICE', and 'Select', and 'ADD'/'DELETE' buttons. The 'CUSTOMER DETAILS' section has 'NAME' and 'MOBILE' text fields. The 'CART' section features a table with columns 'ITEMS', 'QUANTITY', 'PRICE', and 'TOTAL'. A 'PROCEED' button is at the bottom right. Annotations point to various elements: 'Dropdown menu to search products by category' points to the 'CATEGORY' dropdown; 'Search fields to directly search products' points to the 'Search' button; 'Button for going back to the main menu' points to the 'BACK' button; 'Text fields to enter customer information' points to the 'NAME' and 'MOBILE' fields; 'Table to view products in cart' points to the 'CART' table; and 'Button for adding and deleting products from cart' points to the 'ADD' and 'DELETE' buttons.

Image 10: Checkout Window

```

181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
208
209
210
211
212
213
214
215
216
217

1 reference
private bool checkproduct()
{
    if (txtname.Text == "")
    {
        MessageBox.Show("Please Insert the Customer Name ");
    }

    if (txtmobile.Text == "")
    {
        MessageBox.Show("Please Insert the Mobile No ");
        return false;
    }

    return true;
}

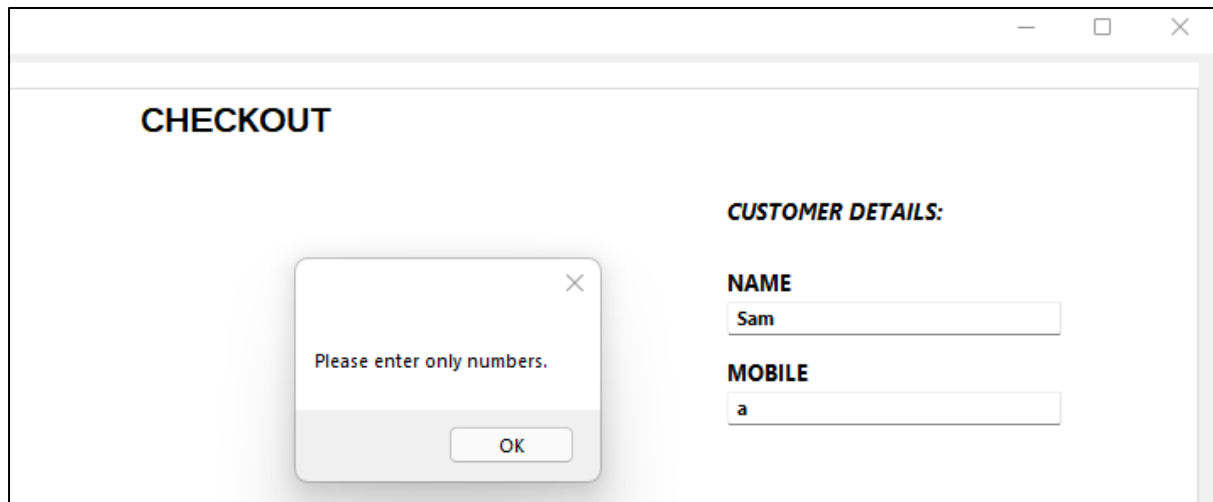
1 reference
private void btnback_Click(object sender, EventArgs e)

1 reference
private void txtmobile_TextChanged(object sender, EventArgs e)
{
    if (System.Text.RegularExpressions.Regex.IsMatch(txtmobile.Text, "[^0-9]"))
    {
        MessageBox.Show("Please enter only numbers.");
        txtmobile.Text = txtmobile.Text.Remove(txtmobile.Text.Length - 1);
    }
}

```

Code 10: Validation for customer name and phone number

Firstly, the user needs to enter the name and mobile number of the customer. A validation check for the name and mobile number is in place as well. If wrong character input or empty space is given, an error message would popup.



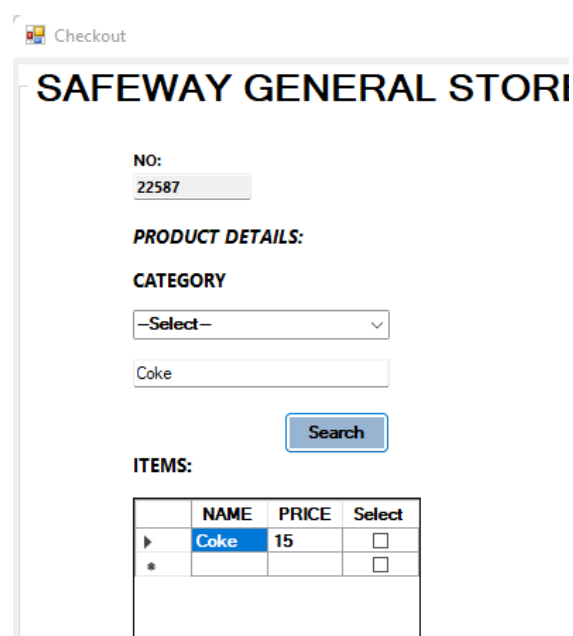
The screenshot shows a web browser window titled "CHECKOUT". On the right side, under the heading "CUSTOMER DETAILS:", there are two input fields. The "NAME" field contains the text "Sam", and the "MOBILE" field contains the text "a". A modal dialog box is centered on the screen, displaying the message "Please enter only numbers." with an "OK" button at the bottom. The dialog box has a close button (X) in the top right corner.

Image 8: Error message for wrong input

Afterwards, the user can select the category from the drop the menu or search the product specifically. The items will then be viewed on the item table and then the user can use the add and delete button to put the products into cart.

Searching a product into the database:

User can also use the search function in the checkout page to search for a product. User simply have to enter the product name in the search text field and click on the 'search' button.



The screenshot shows a web browser window titled "Checkout" with the heading "SAFEMART GENERAL STORE". Below the heading, there is a "NO:" label followed by a text input field containing "22587". Under the heading "PRODUCT DETAILS:", there is a "CATEGORY" label followed by a dropdown menu showing "--Select--" and a text input field containing "Coke". Below these fields is a blue "Search" button. Under the heading "ITEMS:", there is a table with the following data:

	NAME	PRICE	Select
▶	Coke	15	<input type="checkbox"/>
*			<input type="checkbox"/>

Image 9a: Search field

```

218 1 reference
219 private void btnSearch_Click(object sender, EventArgs e)
220 {
221     String SearchItems = txtSearch.Text;
222
223
224     dataGridViewItem.Rows.Clear();
225
226
227
228
229     DataTable list = repo.GetProductItemList(SearchItems);
230
231
232
233
234     foreach (DataRow dr in list.Rows)
235     {
236         string[] listda = { dr["Name"].ToString(), dr["Price"].ToString() };
237         dataGridViewItem.Rows.Add(listda);
238
239
240
241     }
242 }
243

```

Code 11: Search button code

```

308 1 reference
309 public DataTable GetProductItemList(string ItemName)
310 {
311
312
313
314     using (SqlConnection con = new SqlConnection(conString))
315     {
316
317
318
319         using (SqlCommand cmd = new SqlCommand("spGetItemList", con))
320         {
321             con.Open();
322             cmd.CommandType = CommandType.StoredProcedure;
323
324
325
326             cmd.Parameters.Add("@pItem", SqlDbType.NVarChar, 50).Value = ItemName; // parameter.Name;
327
328
329
330             SqlDataAdapter da = new SqlDataAdapter(cmd);
331             DataTable dt = new DataTable();
332             da.Fill(dt);
333

```

Code 12: Searching 'Products' from database ⁵

⁵ "Searching for and displaying sql database data C#," Stack Overflow, February 2022, <https://stackoverflow.com/questions/26678009/searching-for-and-displaying-sql-database-data-c-sharp>

Checkout

SAFEGWAY GENERAL S

NO:
22587

PRODUCT DETAILS:

CATEGORY

--Select--

- Chips
- Chocolate
- Drinks
- Fruit
- Vegetables

Search

ITEMS:

	NAME	PRICE	Select
▶	Coke	15	<input type="checkbox"/>
*			<input type="checkbox"/>

Checkout

SAFEGWAY GENERAL ST

NO:
22587

PRODUCT DETAILS:

CATEGORY

Drinks

Search

ITEMS:

	NAME	PRICE	Select
▶	Coke	15	<input type="checkbox"/>
	Pepsi	15	<input type="checkbox"/>
	Sprite	15	<input type="checkbox"/>
*			<input type="checkbox"/>

Image 9b: Viewing products by category

ITEMS:

	NAME	PRICE	Select
*			<input type="checkbox"/>

ADD

DELETE

BACK

Image 10: Buttons to add items to the cart or delete items from the cart


```

1 reference
71 private void btnadd_Click(object sender, EventArgs e)
72 {
73     DataTable dt = new DataTable();
74     dt.Columns.Add("NAME");
75     dt.Columns.Add("PRICE");
76     foreach (DataGridViewRow row in dataGridViewItem.Rows)
77     {
78         bool isSelected = Convert.ToBoolean(row.Cells["checkbox"].Value);
79         if (isSelected)
80         {
81             if (checkProdArrival(row.Cells[0].Value.ToString()))
82             {
83                 if (AddItem(row.Cells[0].Value.ToString(), row.Cells[1].Value.ToString()))
84                     DisplayData();
85             }
86             else
87             {
88                 MessageBox.Show("PRODUCT Is not available");
89             }
90         }
91     }
92 }
93
94
95
96
97
98

```

Code 13: Add product to cart button

```

1 reference
648 public bool AddItemBill(string itemName, string itemPrice)
649 {
650
651     try
652     {
653         //SqlCommand sqlCom = new SqlCommand("select * from dms_DeviceLog where ID = (select MAX(ID) from dms_DeviceLog)");
654         //conn.Open();
655         //query = "spInsertTemplate";
656
657         using (SqlConnection con = new SqlConnection(conString))
658         {
659             using (SqlCommand cmd = new SqlCommand("sp_InsertItemBill", con))
660             {
661                 con.Open();
662                 cmd.CommandType = CommandType.StoredProcedure;
663
664                 cmd.Parameters.Add("@Item", SqlDbType.NVarChar, 50).Value = itemName;
665                 cmd.Parameters.Add("@Price", SqlDbType.Float).Value = float.Parse(itemPrice);
666
667                 cmd.ExecuteNonQuery();
668
669                 return true;
670             }
671         }
672     }
673     catch
674     {
675     }
676 }
677
678

```

Code 14: Adding product to cart

```

1 reference
112 private bool DeleteItem(string itemname, string price)
113 {
114
115     return repo.DeleteItemBill(itemname, price);
116 }
117

```

Code 15: Delete product to cart button

```

768 1 reference
769 public bool DeleteItemBill(string itemName, string itemPrice)
770 {
771     try
772     {
773         //SqlCommand sqlCom = new SqlCommand("select * from dms_DeviceLog where ID = (select MAX(ID) from dms_DeviceLog)");
774
775         //conn.Open();
776         //query = "spInsertTemplate";
777
778         using (SqlConnection con = new SqlConnection(conString))
779         {
780
781             using (SqlCommand cmd = new SqlCommand("sp_DeleteItemBill", con))
782             {
783
784                 con.Open();
785                 cmd.CommandType = CommandType.StoredProcedure;
786
787                 cmd.Parameters.Add("@Item", SqlDbType.NVarChar, 50).Value = itemName;
788                 cmd.Parameters.Add("@Price", SqlDbType.Float).Value = float.Parse(itemPrice);
789                 cmd.ExecuteNonQuery();
790
791                 return true;
792             }
793         }
794     }
795     catch (Exception errorException)
796     {
797         return false;
798     }
799 }
800
801

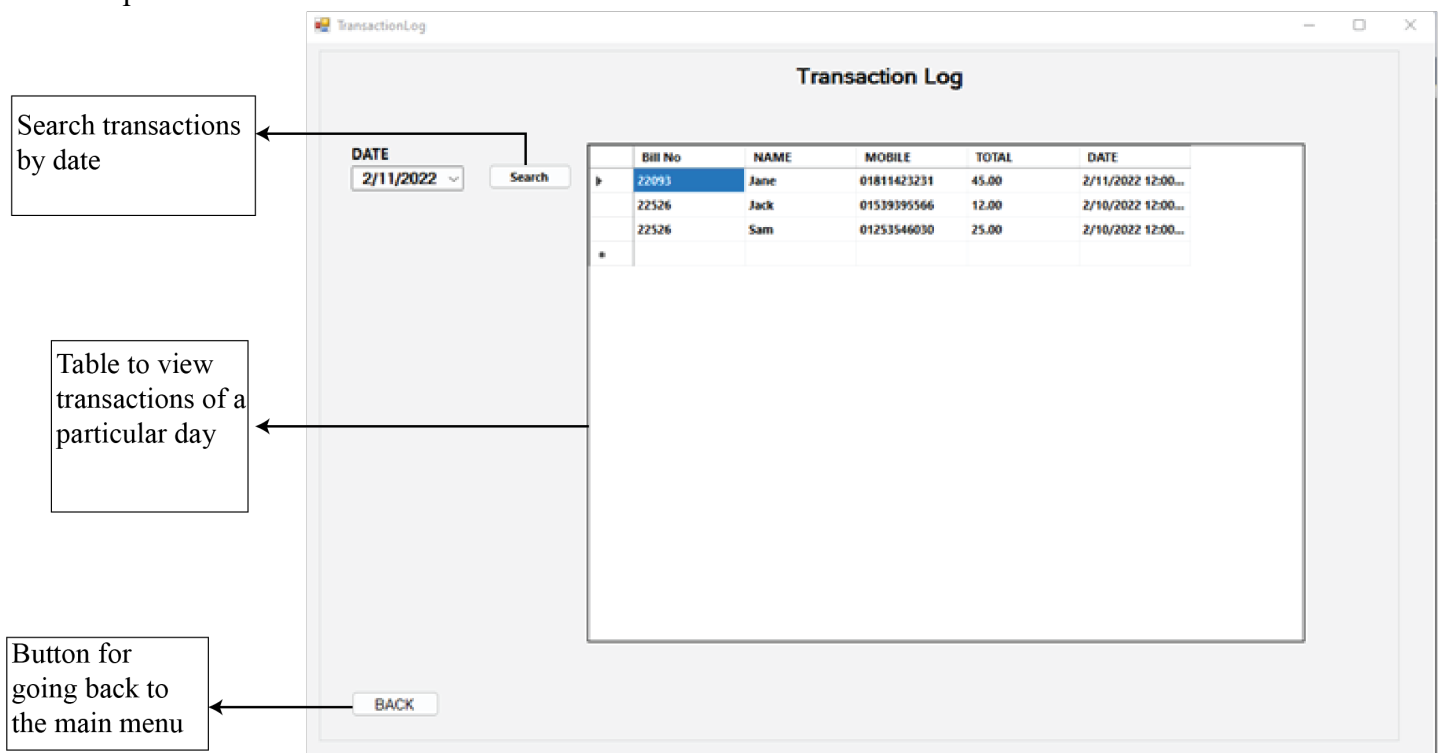
```

Code 16: Deleting product from cart

After the products are added into the cart, the user can then click on ‘proceed’ button for checkout.

Transaction Log Window:

The user can click on the transaction log button to view past transaction of sales that took place in his store.



The user can select a particular date and click on the search button to view the list of transactions that took on that particular day. Important information such as Bill No., Name, mobile and the total amount sold is displayed on the table.

```

557 1 reference
558 public DataTable GetTransactionLog()
559 {
560
561     using (SqlConnection con = new SqlConnection(conString))
562     {
563
564         using (SqlCommand cmd = new SqlCommand("sp_GetTransactionLog", con))
565         {
566             con.Open();
567             cmd.CommandType = CommandType.StoredProcedure;
568
569             SqlDataAdapter da = new SqlDataAdapter(cmd);
570             DataTable dt = new DataTable();
571             da.Fill(dt);
572
573             //foreach (DataRow dr in dt.Rows)
574             //{
575                 // if (dr["UserName"] != null)
576                 //{
577                     // if (parameter.Username == (string)dr["UserName"] && parameter.Password == (string)dr["Password"])
578                     //     return true;
579                 // }
580             // }
581             // }
582
583             return dt;
584         }
585     }
586 }
587
61 1 reference
62 private void btnSearch_Click(object sender, EventArgs e)
63 {
64     DateTime theDate = dateTimeTransactionLog.Value;
65     DisplayData(theDate);
66
67 }
68
69

```

Code 14: Code for searching transaction records by date

Change Password Window:

If the user wishes to change his password, he can go to the change password window to change it accordingly.

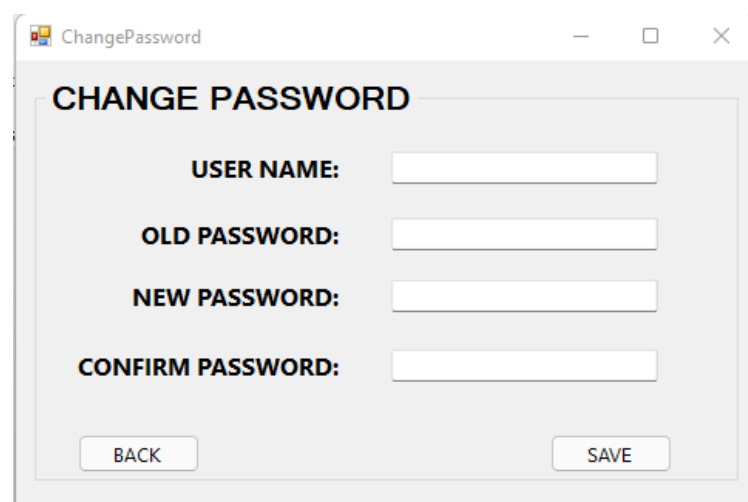


Image 12: Change password window

A verification is taken initially, as the program will be asking the user to input his existing username and password. Once the new password is inputted, the user will again be prompted to re-input the new password. A series of validations will take place such as, if the username old password does not match with the data in the database, and error message will appear. Similarly, if the new password entered doesn't match with the input that is in confirm password, an error message will pop up as well.

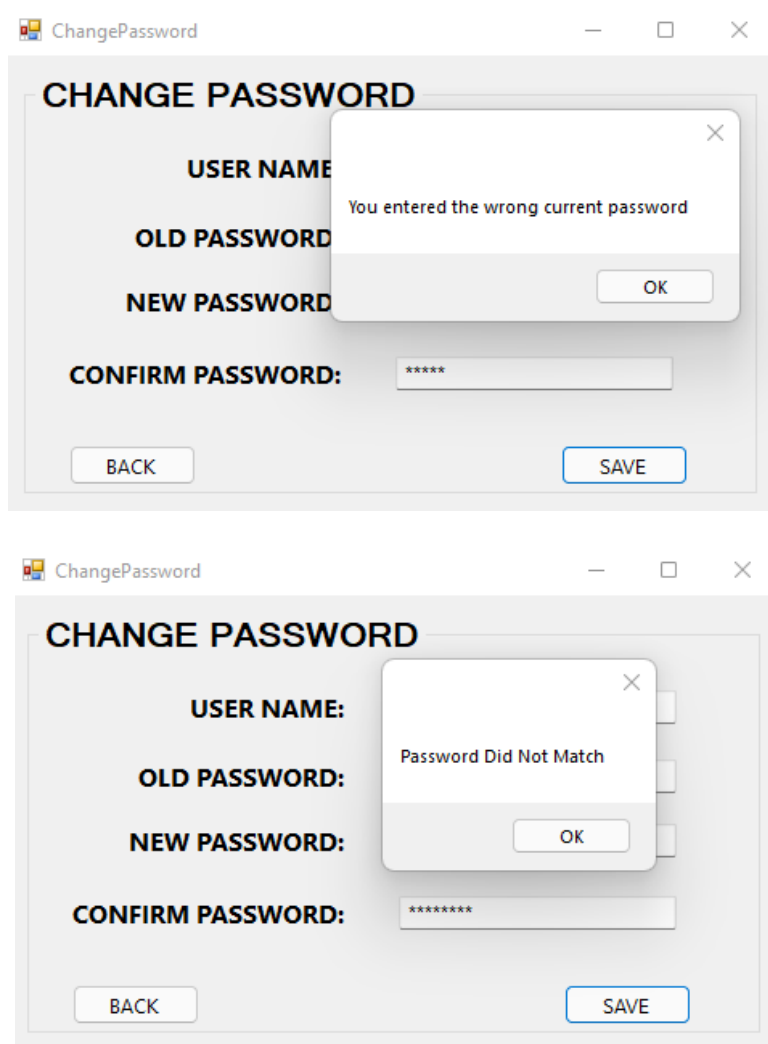


Image 13: Password change error messages

If the validations are successful, a 'password successfully changed' message will popup, informing the user that the password change was successful.

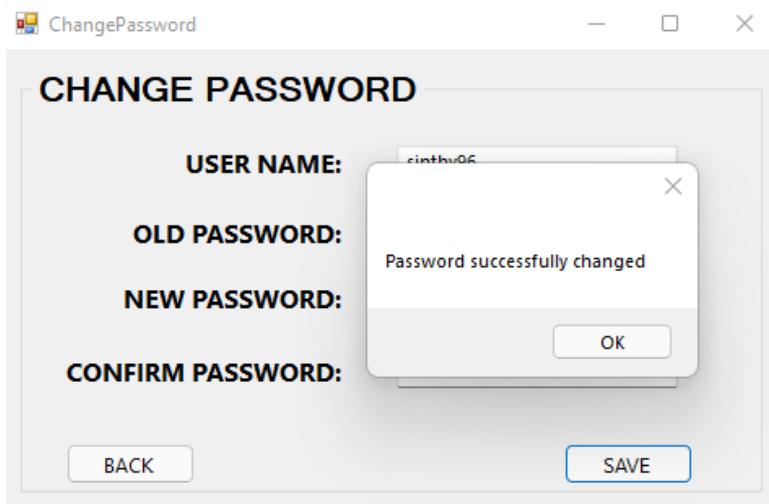


Image 14: Password change success message

```

21 1 reference
22 private void btnsave_Click(object sender, EventArgs e)
23 {
24     SecUser ChangeUser = new SecUser();
25
26     if (checkpass())
27     {
28         ChangeUser.Username = txtUserName.Text;
29         ChangeUser.OldPassword = txtOldPassword.Text;
30         ChangeUser.NewPassword = txtNewPassword.Text;
31         ChangeUser.Password = txtConfirmPassword.Text;
32
33
34         if (ChangeUser.NewPassword == ChangeUser.Password)
35         {
36             DBHandler chPass = new DBHandler();
37
38             if (chPass.CheckExistingPassword(ChangeUser))
39             {
40                 if (chPass.ChangePassword(ChangeUser))
41                 {
42                     MessageBox.Show("Password successfully changed");
43                     ClearText();
44                 }
45             }
46             else
47             {
48                 MessageBox.Show("Password Did not Match");
49             }
50         }
51         else
52         {
53             MessageBox.Show("You entered the wrong current password");
54         }
55     }
56     else
57     {
58         MessageBox.Show("Password Did Not Match");
59     }
60 }
61
62
63
64

```

Code 14: Password validation ⁶

⁶“How to change password to user account, by c# code?” Stack Overflow, February 2022, <https://stackoverflow.com/questions/4253893/how-to-change-password-to-user-account-by-c-sharp-code>

```

1 reference
93 private bool checkpass()
94 {
95     if (txtUserName.Text == "")
96     {
97         MessageBox.Show("Please Insert the Username");
98         return false;
99     }
100
101     if (txtOldPassword.Text == "")
102     {
103         MessageBox.Show("Please Insert your old password ");
104     }
105
106
107     if (txtNewPassword.Text == "")
108     {
109         MessageBox.Show("Please Insert the new password");
110         return false;
111     }
112
113     if (txtConfirmPassword.Text == "")
114     {
115         MessageBox.Show("Please confirm your password");
116         return false;
117     }
118
119     return true;
120 }

```

Once the user has completed his work, he can simply logout from the program using the logout button from the main window.

Methods and classes used:

Classes:

No.	Class	Description
1	public class DBHandler	Main class
2	public partial class Login	Login Window Class
3	public partial class MainMenu	Main Menu Window Class
4	public partial class Checkout	Checkout Window Class
5	public partial class frmTransaction	Transaction Log Window Class
6	public partial class frmproduct	Products Window Class
7	public partial class frmChangePassword	Password Change WindowClass

Public Main Methods:

No.	Method	Description
1	<code>public bool AddProduct(ProductObject prodparameter)</code>	Method to push the product data into the database
2	<code>internal bool AddProduct(object prodObj)</code>	Method for adding product into the database
3	<code>public bool AddTransactionLog(ProductObject prodparameter)</code>	Method for storing
4	<code>public bool UserCheck(string UserName, string UserPassword, int strUserType)</code>	for the validations of user id from database
5	<code>public bool LogIn(SecUser parameter)</code>	Method for getting user from database
6	<code>public DataTable deleteProduct()</code>	Method for deleting product from database
7	<code>public DataTable Getcatogry()</code>	Method for getting product category from database
8	<code>public DataTable GetProductCategoryList(string categoryName)</code>	Method for getting product category list for the dropdown menu
9	<code>public DataTable GetProductItemList(string ItemName)</code>	Getting product item list for the search
10	<code>public bool UpdateProductDetails(ProductObject prodparameter)</code>	Method for updating product details in database and table
11	<code>public bool deleteProduct(ProductObject prodOBJ)</code>	Method for deleting product from the database
12	<code>public bool ChangePassword (SecUser ChangePass)</code>	Method for changing user password
13	<code>public bool CheckExistingPassword(SecUser ChangePass)</code>	Method for validating existing username and password from database

14	<code>public bool checkprod (string prodname)</code>	Method for checking whether product is available in stock
15	<code>public bool productCalculation(string prodName)</code>	Method for stock deduction
16	<code>public DataTable GetProductbillList()</code>	Method for getting product cart list
17	<code>public DataTable GetTransactionLog()</code>	Method for pulling data for the transaction log from the database
18	<code>public DataTable GetTransactionLog(DateTime adddate)</code>	searching transaction log by date
19	<code>public bool ClearTempBillTable()</code>	Method for clearing checkout list
20	<code>public bool AddItemBill(string itemName, string itemPrice)</code>	Method for adding item to checkout from database
20	<code>public bool DeleteItemBill(string itemName, string itemPrice)</code>	Method for deleting item to checkout from database

Private methods used in login page:

No.	Method	Description
1	<code>private void btnlogin_Click(object sender, EventArgs e)</code>	Method for logging user to main menu
2	<code>private void txtpassword_TextChanged_1(object sender, EventArgs e)</code>	Verification for logging user into the program

Private methods used in products page:

No.	Method	Description
1	<code>private void btnadd_Click(object sender, EventArgs e)</code>	Add button
2	<code>private void btnupdate_Click(object sender, EventArgs e)</code>	Update button
3	<code>private void btnsave_Click(object sender, EventArgs e)</code>	Save button
4	<code>private void btndelete_Click(object sender, EventArgs e)</code>	Delete Button
5	<code>private void DisplayData()</code>	Displays products in the table
6	<code>private void datagrdProduct_CellClick(object sender, DataGridViewCellEventArgs e)</code>	Method for selecting product from its cell
7	<code>private void textClear()</code>	Clear text from text field
8	<code>private void btnback_Click(object sender, EventArgs e)</code>	Back button to go back to main menu
9	<code>private void txtcat_TextChanged(object sender, EventArgs e)</code>	Method for Category Field validation
10	<code>private void txtprod_TextChanged</code>	Method for Product Field validation

Private methods used in checkout page:

No.	Method	Description
1	<code>private void label1_Click(object sender, EventArgs e)</code>	Method for clicking label
2	<code>private void loadCombobox()</code>	Method for category list
3	<code>private void cmbcat_SelectedIndexChanged(object sender, EventArgs e)</code>	Dropdown category list
4	<code>private void dataGridViewItem_CellContentClick(object sender, DataGridViewCellEventArgs e)</code>	Showing item on grid

5	<code>private void btnadd_Click(object sender, EventArgs e)</code>	Adding items to cart button
6	<code>private void btndelete_Click(object sender, EventArgs e)</code>	Deleting items from cart button
7	<code>private bool checkProdArrival(string prodName)</code>	Check for stock
8	<code>private bool AddItem(string itemname , string price)</code>	Method call for add item action
9	<code>private bool DeleteItem(string itemname, string price)</code>	Method call for delete item action
10	<code>private void DisplayData()</code>	Displaying item on table
11	<code>private void btnSave_Click(object sender, EventArgs e)</code>	Proceed Button
12	<code>private bool checkproduct()</code>	Validation for customer name
13	<code>private void txtmobile_TextChanged(object sender, EventArgs e)</code>	Validation for customer phone number
14	<code>private void btnback_Click(object sender, EventArgs e)</code>	Back button to go back to main menu
15	<code>private void btnSearch_Click(object sender, EventArgs e)</code>	Search Button

Private methods used in transaction log page:

No.	Method	Description
1	<code>private void DisplayData(DateTime adddate)</code>	Display table by date
2	<code>private void btnSearch_Click(object sender, EventArgs e)</code>	Search Button
3	<code>private void btnback_Click(object sender, EventArgs e)</code>	Back button to go back to main menu

Word Count – 999 Words (Excluding captions and headings)

Bibliography

- “How to insert data to a database in C#,” Stack Overflow, February 2022, <https://stackoverflow.com/questions/57448296/how-to-insert-data-to-a-database-in-c-sharp>
- Mehmood, Ehtesham. “Insert, Update, Delete, Display Data in Mysql Using C#.” C# Corner, June 3, 2020. <https://www.c-sharpcorner.com/UploadFile/9582c9/insert-update-delete-display-data-in-mysql-using-C-Sharp/>.
- “Searching for and displaying sql database data C#,” Stack Overflow, February 2022, <https://stackoverflow.com/questions/26678009/searching-for-and-displaying-sql-database-data-c-sharp>
- “Validation of numbers and letters using C# for a beginner’,” Stack Overflow, February 2022, <https://stackoverflow.com/questions/8867097/validation-of-numbers-and-letters-using-c-sharp-for-a-beginner>