



slington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS6004NI Application Development

Assessment Weightage Type
30% Individual Coursework

Semester
2022 Autumn

Student Name: Bhuwani Bikram Nembang

London Met ID: 22015664

College ID: NP01CP4S220164

Assignment Due Date: 20/09/2022

Assignment Submission Date: 20/09/2022

Submitted To: Mr. Samyush Maharjan

Word Count (Where required): 2795

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction	1
1.1. Authentication and User Management:.....	2
1.1.1. Order Management:	2
1.1.2. Pricing Configuration:	2
1.1.3. Customer Membership:	2
1.1.4. Reporting:.....	3
1.1.5. Database Integration:	3
1.1.6. User Interface:.....	3
1.6.7. Exception Handling and Validation:.....	3
1.6.8. Documentation:	3
2. System Requirements:	4
2.1. Program Execution	4
2.2. INSTRUCTION TO RUN PROGRAM.	5
2.2.1. Logging in as Administrator.....	5
2.2.2. Logging in as a User	10
3. Logical Solution Description	12
3.1. Flowchart of the Admin.	12
3.2. Flowchart of the User.....	13
3.3. CS Codes its property's purpose and methods.....	14
4. Implementation of Desktop Application	18
4.1. Password-protected access, with an additional layer of password security required for making changes to the prices of coffee options and add-ins.	18
4.2. A working view to order coffee(s).....	20
4.3. Ability to add multiple add-ins.	24
4.4. Application's state is saved and restored using any semi structured formats like JSON, CSV, or Excel.....	25
4.5. Membership registration with applied benefits like free coffee redemption and discounts.	26
4.6. A table to display all the purchases/sales transactions.....	27
4.7. generate file that includes daily or specific month purchases/sales.....	28
Conclusion	30

List of Figures

Figure 1 Solution explorer of the project.....	4
Figure 2: Click on Windows Machine start section or just press F5.	5
Figure 3 Logging in as Admin.....	5
Figure 4: change password of admin.	6
Figure 5 Admin panel	7
Figure 6: Coffees added by the admin.	8
Figure 7: add-ins added by the admin.	8
Figure 8: Add user in the Bislerium cafe.....	9
Figure 9: Users added into the Bislerium cafe.....	9
Figure 10:Figure of logging in as a user.	10
Figure 11: log in and order list.	11
Figure 12: flowchart of the admin.	12
Figure 13: flowchart of the user.....	13
Figure 14: admin login.....	18
Figure 15: change recent password of admin.	19
Figure 16: empty the order list before placing an order in add-to-cart.....	20
Figure 17: add-in before customer details.....	21
Figure 18: add-to cart items.	22
Figure 19: pop-up message as order is successfully placed.	23
Figure 20: Adding coffee.	24
Figure 21: Adding add-ins.	24
Figure 22: .pdf file for daily as well as monthly reports and JSON files.	25
Figure 23: discount not available for normal user or non-member.	26
Figure 24: table of purchases and sales transactions.	27
Figure 25: generate reports of the day and month.	28
Figure 26: select today date for today sale report.	28
Figure 27: select monthly date for today sale report.....	29
Figure 28: monthly sales report generated.....	29

List Of Tables

Table 1: Table of methods with descriptions.	17
---	----

Project Title: Bislerium CAFÉ

1. Introduction

In the dynamic situation of modern cafes and coffee shops, the need for an efficient and optimized point of sale (POS) system is paramount. The **Bislerium Café** POS system is a robust software solution developed in **.NET** designed to improve operational efficiency at **Bislerium Café**. This comprehensive POS system is tailored to the specific needs of cafes, providing an easy-to-use interface for staff, powerful management tools for administrators, and an improved experience for loyal customers. Some specific requirements, here's a general outline for creating a simple desktop application using C# **.NET**. I installed Visual Studio 2020 installed for C# development. **.NET MAUI** uses a single API to unify Android, iOS, macOS, and Windows APIs into a write-once run-anywhere developer experience. MAUI apps provide deep access into each native platform. **.NET 6** introduces a series of platform-specific frameworks: **.NET** for Android, **.NET** for iOS, **.NET** for macOS, and Windows UI (Win UI) Library. The **.NET 6** Base Class Library is shared among all the platforms while abstracting the individual characteristics of each platform from your code. The **.NET** runtime is used for the execution environment for MAUI applications, even though the underlying implementations of the runtime may be different, depending on the host.

Creating a Point of Sale (POS) software solution for **Bislerium Café** involves several components, including user authentication, order management, pricing configuration, customer membership, and reporting functionalities. Here's an overview of how we can structure the solution using C# **.NET**:

1.1. Authentication and User Management:

Implement a secure login system with password protection for both staff and administrators. Use role-based access control to differentiate between staff and administrators.

1.1.1. Order Management:

Create a user-friendly interface for staff to take customer orders. Each order should include:

- Coffee type (Cappuccino, Latte, Espresso, Doppio etc.).
- Add-ins (Cinnamon, Honey, Ginger, Chocolate, Ice Cream, etc.).
- Calculate the total price based on the selected items and their configured prices.

1.1.2. Pricing Configuration:

Allow administrators to configure prices for each coffee type and add-in. This functionality should be protected by a separate password to ensure only authorized personnel can adjust pricing.

1.1.3. Customer Membership:

Implement a membership system with the following features:

- Members can redeem a free coffee after every 10 purchases.
- Regular members get a 10% discount on all purchases for a month.
- Regular customer status is achieved by making daily purchases for a full month, excluding weekends.
- Membership identification through NFC-enabled cards or optional username/phone number entry.

1.1.4. Reporting:

Create a reporting system for staff to view and generate PDF reports. Reports should include:

- Daily or monthly reports.
- Top 5 most frequently purchased coffee types and add-ins.
- Quantities of each item sold.
- Total revenue.

1.1.5. Database Integration:

Use a database to store information about orders, pricing, customer details, and membership status. Entity Framework can be used for database operations.

1.1.6. User Interface:

Design an intuitive and user-friendly interface for staff and administrators. Consider using Windows Forms or WPF for desktop applications.

1.6.7. Exception Handling and Validation:

Implement error handling and validation to ensure data integrity and prevent unauthorized access.

1.6.8. Documentation:

Document the code, especially for complex logic and configurations. Include user manuals for staff and administrators.

2. System Requirements:

2.1. Program Execution

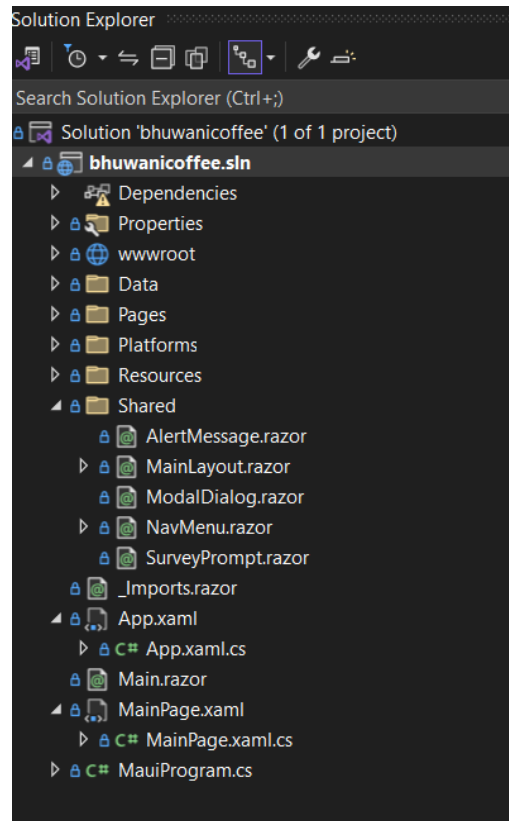


Figure 1 Solution explorer of the project.

The steps to be followed to run the application are:

- Move to the location of the Blazor project and select the project file.
- After the project is open in Visual Studio, Select the desired debug and platform from the toolbar.
- Select the 'Start' button to build and run the project. (Press F5 key).

2.2. INSTRUCTION TO RUN PROGRAM.

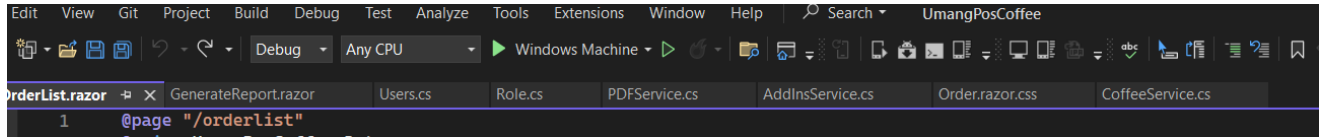


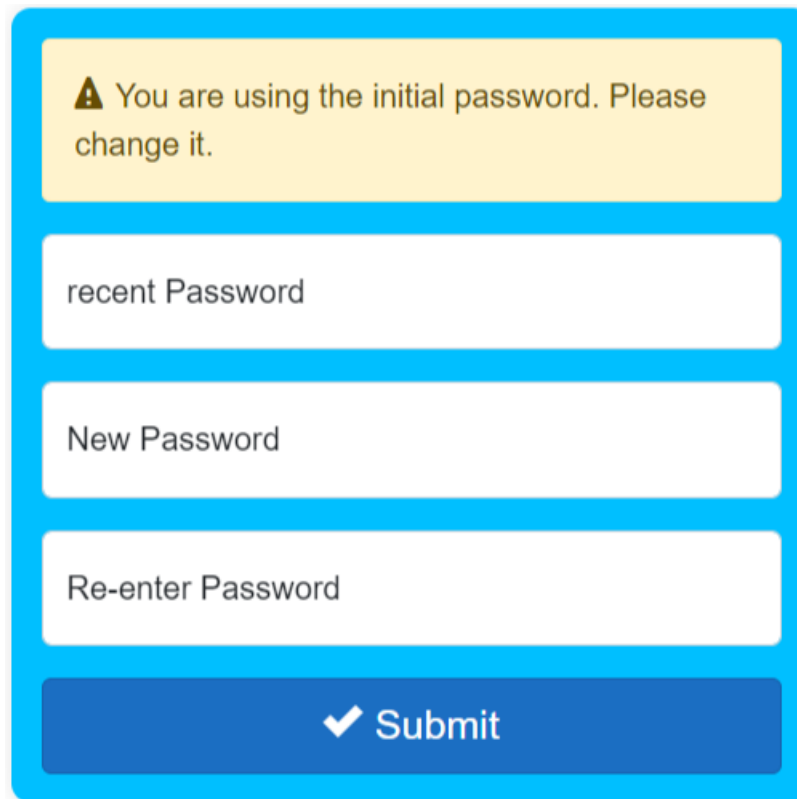
Figure 2: Click on Windows Machine start section or just press F5.

2.2.1. Logging in as Administrator.

A login form for 'Blslerium Cafe'. The background is a solid blue color. The text 'Welcome to Blslerium Cafe, Please Login' is centered in a large, black, sans-serif font. Below the text are two white input fields with rounded corners. The first field is labeled 'Username' and contains the text 'admin'. The second field is labeled 'Password' and contains the text 'admin'. To the right of the password field is a small icon of an eye with a diagonal line through it, indicating a toggle for password visibility. Below the input fields is a large black button with a white right-pointing arrow and the text 'Login' in white.

Figure 3 Logging in as Admin.

When the above form appears on the application, we added user and password of the admin to enter the admin panel.



⚠ You are using the initial password. Please change it.

recent Password

New Password

Re-enter Password

✓ Submit

Figure 4: change password of admin.

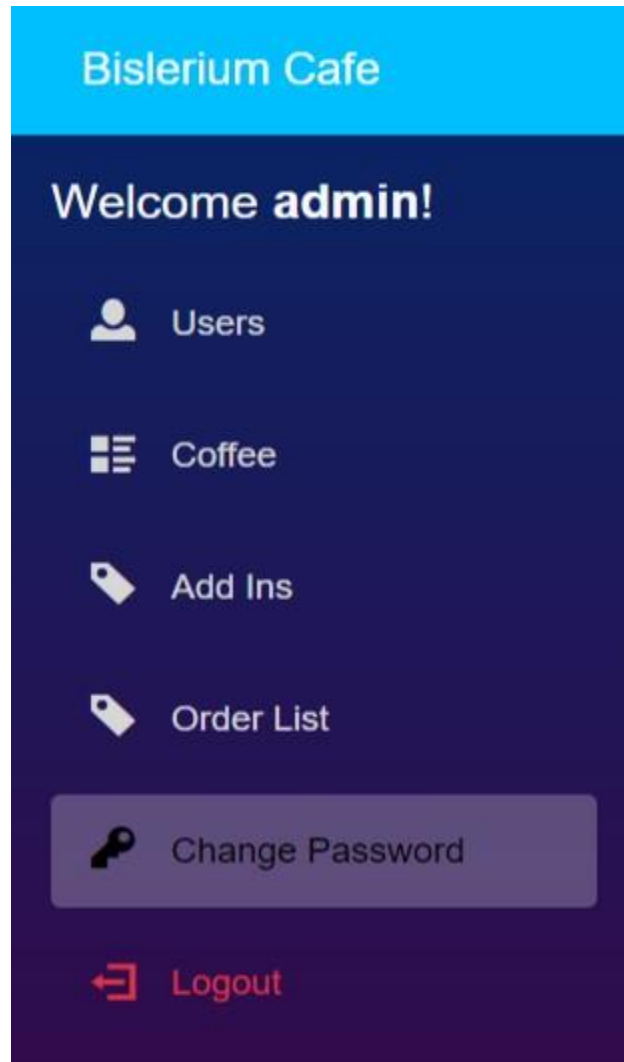


Figure 5 Admin panel

If the following screen appears then login as admin was successful.

The main responsibility of the admin is to manage the customer orders, Users. Admin will be able to change the password of the admin himself. Admin can also change the price of the coffee and add-ins with a protected password system. Admin will also be able to look through different reports generated in the app. By going through the report's admin will be able to check the functioning of the app and will be able to check the smooth running of the café.

<input type="text" value="Search"/>			
<input type="button" value="SEARCH"/>			
ID	Coffee Name	Coffee Price	Actions
1	Doppio	100	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
2	Cappuccino	150	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
3	Latte	100	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
4	Espresso	120	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
5	Mocha	200	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
<input type="button" value="+ Add"/>			

Figure 6: Coffees added by the admin.

<input type="text" value="Search"/>			
<input type="button" value="SEARCH"/>			
ID	AddIn Name	AddIn Price	Actions
1	Ice Cream	50	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
2	Cinnamon	50	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
3	Honey	80	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
4	Ginger	40	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
5	Chocolate	80	<input type="button" value="Remove"/> <input type="button" value="Edit"/>
<input type="button" value="+ Add"/>			

Figure 7: add-ins added by the admin.


Add User

Username

bhanu

Password

bhanu



☒ User

☐ Admin

Cancel

Add

Figure 8: Add user in the Bislerium cafe.

Search

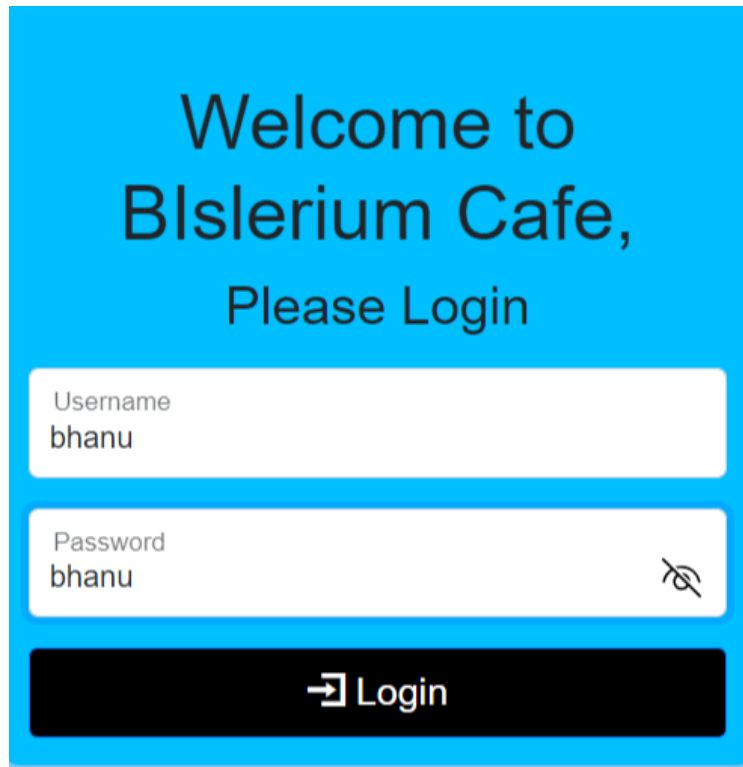
SEARCH

Username	Role	Created At	CreatedBy	Actions
admin	Admin	Jan 10, 2024 2:25 PM	Unknown	Delete
bhanu	User	Jan 10, 2024 2:25 PM	admin	Delete
ankit	User	Jan 10, 2024 2:49 PM	admin	Delete
anurodh	User	Jan 10, 2024 2:50 PM	admin	Delete
rijesh	User	Jan 10, 2024 2:51 PM	admin	Delete
umesh	User	Jan 11, 2024 12:07 AM	admin	Delete
ram	User	Jan 11, 2024 12:07 AM	admin	Delete
john	User	Jan 11, 2024 12:07 AM	admin	Delete

+ Add

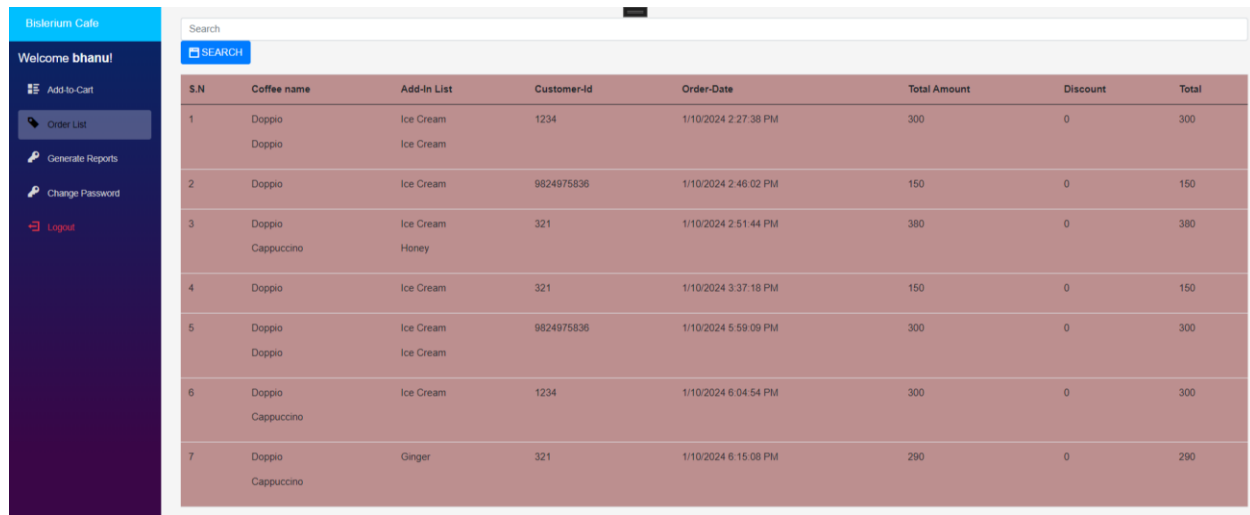
Figure 9: Users added into the Bislerium cafe.

2.2.2. Logging in as a User



A login form for 'Blslerium Cafe' with a blue background. The text 'Welcome to Blslerium Cafe, Please Login' is centered at the top. Below it are two white input fields: 'Username' with the value 'bhanu' and 'Password' with the value 'bhanu'. A toggle icon for password visibility is on the right of the password field. At the bottom is a black 'Login' button with a right-pointing arrow icon.

Figure 10:Figure of logging in as a user.



The screenshot displays the Biderium Cafe application interface. On the left is a dark sidebar with a blue header containing the text "Biderium Cafe". Below the header, it says "Welcome bhanu!". The sidebar contains several menu items: "Add to Cart", "Order List" (highlighted), "Generate Reports", "Change Password", and "Logout". The main content area has a search bar at the top with a "SEARCH" button. Below the search bar is a table with 8 columns: S.N, Coffee name, Add-In List, Customer-Id, Order-Date, Total Amount, Discount, and Total. The table contains 7 rows of order data.

S.N	Coffee name	Add-In List	Customer-Id	Order-Date	Total Amount	Discount	Total
1	Doppio	Ice Cream	1234	1/10/2024 2:27:38 PM	300	0	300
2	Doppio	Ice Cream	9824975836	1/10/2024 2:46:02 PM	150	0	150
3	Doppio	Ice Cream	321	1/10/2024 2:51:44 PM	380	0	380
4	Doppio	Ice Cream	321	1/10/2024 3:37:18 PM	150	0	150
5	Doppio	Ice Cream	9824975836	1/10/2024 5:59:09 PM	300	0	300
6	Doppio	Ice Cream	1234	1/10/2024 6:04:54 PM	300	0	300
7	Doppio	Ginger	321	1/10/2024 6:15:08 PM	290	0	290

Figure 11: log in and order list.

As a user we will have the ability to view the membership of the customers and identify the type of the customers. User will be able to See the menu and can order the coffee with different add-ins you like. User can also see the orders that they made and can make the payment in different methods they like. User will also be able to change their password according to their need.

3. Logical Solution Description

3.1. Flowchart of the Admin.

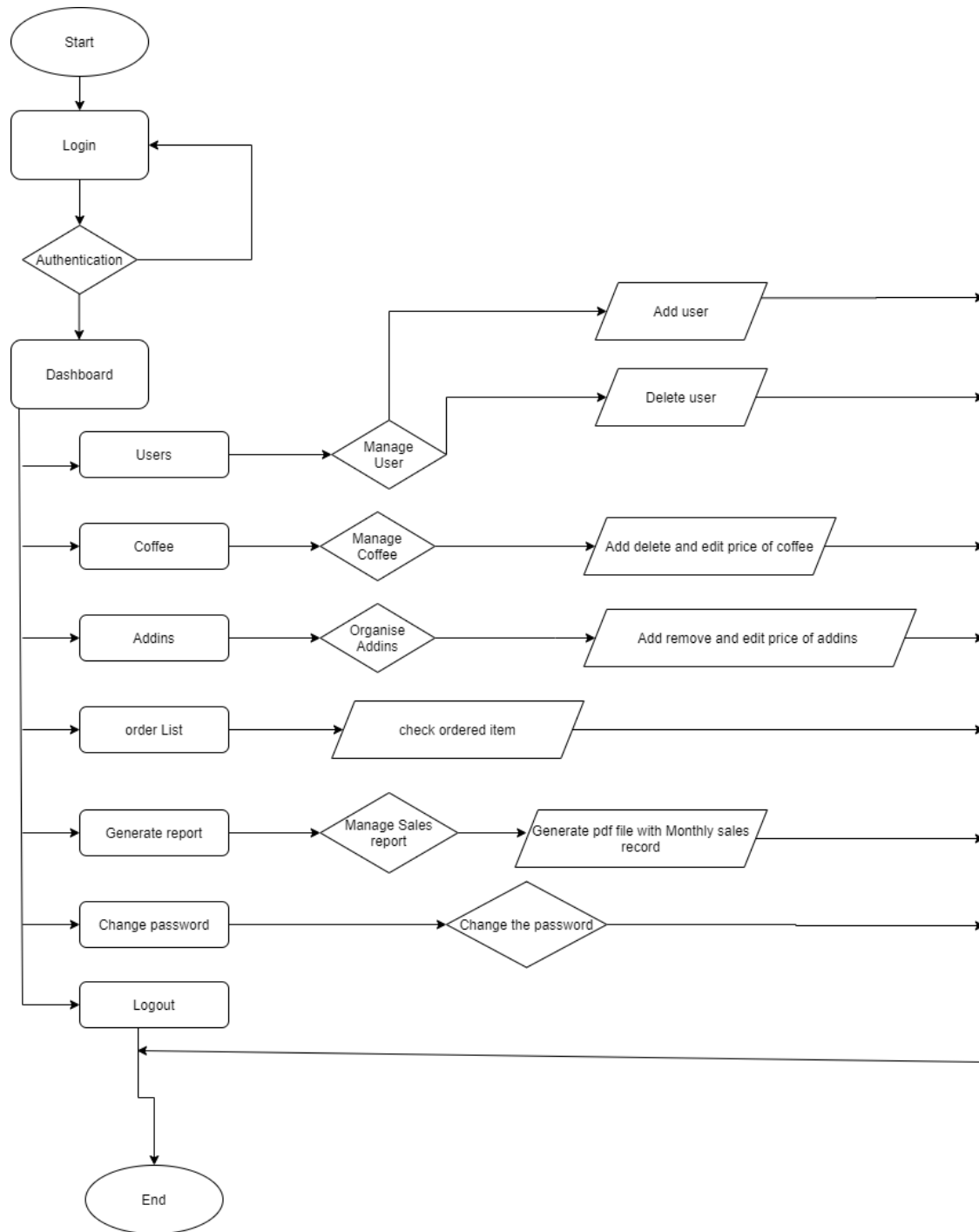


Figure 12: flowchart of the admin.

3.2. Flowchart of the User.

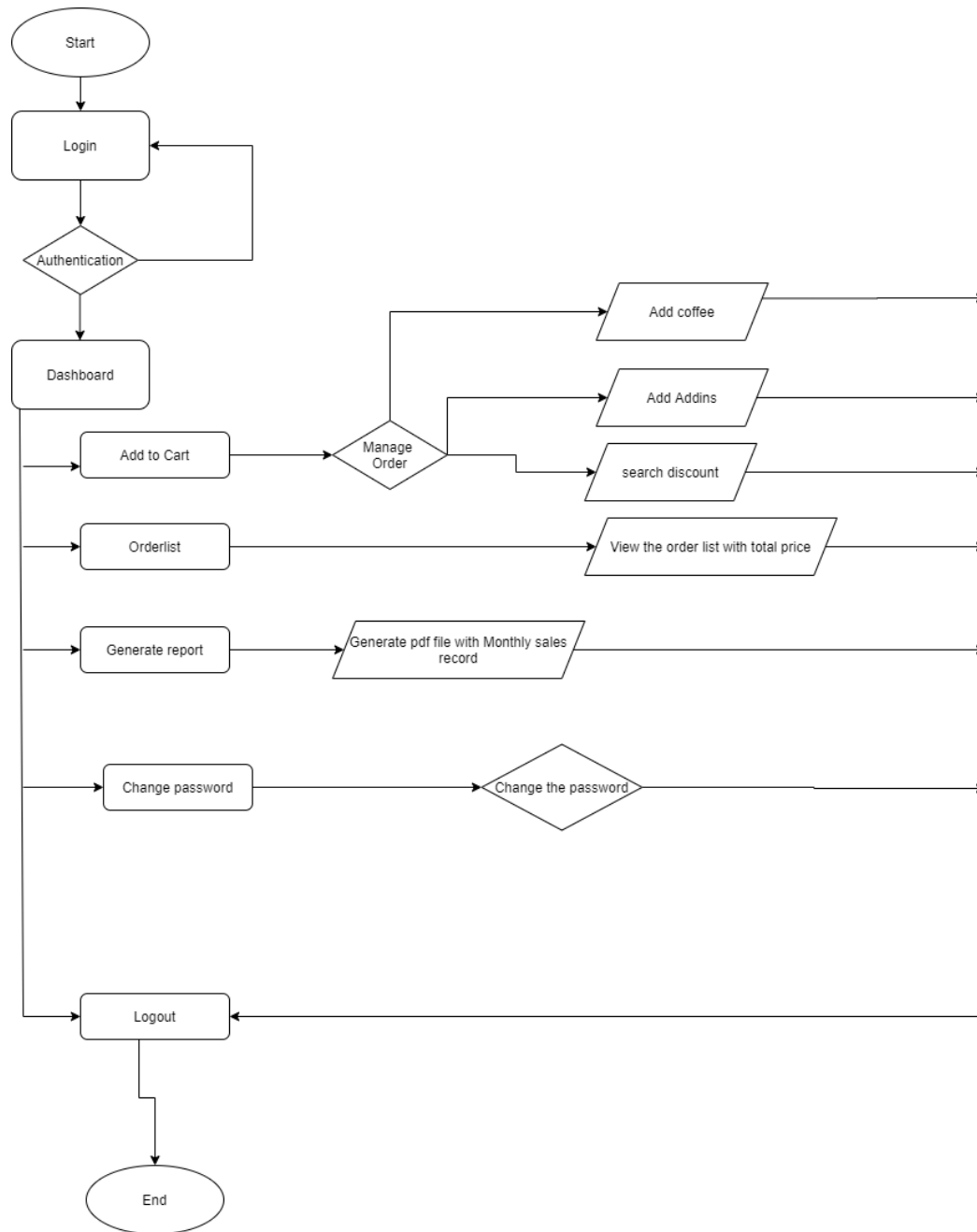


Figure 13: flowchart of the user.

3.3. CS Codes its property's purpose and methods.

CS Codes	Property	Purpose	Methods
Add-Ins	<ul style="list-style-type: none"> • Int Add-ins String Name* • Int Add-ins Price 	To represent and contain information about add-ins in the setting of a coffee related system.	Addins() Save all addins() Add addins() Getalladdins() Update addins() Delete alladdins()
Coffee	<ul style="list-style-type: none"> • String Coffee Name • Int Coffee price • Int ID 	Main purpose is to evaluate and manage information regarding coffee. Instance of the class can be made to represent individual coffee options.	Coffees() Saveallcoffee() Addcoffee() Getall coffee() Update coffee() Delete coffee()
Coffee Service	No properties	The class covers all the methods for managing different coffees objects, i.e. Saving, adding, retrieving. It helps to perform CRUD operation.	Get all() Saveallcoffee() Addcoffee() Updatecoffee() Deletecoffee
Global State	<ul style="list-style-type: none"> • Current User 	This class is used to maintain the single property for current user. It provides access to global app state, referring current user.	No methods

Orders	<ul style="list-style-type: none"> • List coffee • List Add-ins • Int total amount • Int discount • String Customer phone • Datetime OrderDate • Int Grandtotal 	This class is representation of order in bislerium app. It intakes information about list of coffee and add-ins, total amount and soon	Add()
Order Service	<ul style="list-style-type: none"> • Orders orderInstance 	The class serves to manage order in the app.it contain methods to retrieve,count and analyze orderdetermining the ability for discount.It allows persistent storage of information of order.	Order service() Get all orders() Get order count () Getuniqueorderforcustomer() Checkdiscountavailability() Getcoffeeewithhighprice() Saveallorders()
PDF Service	No Property	The class serves to generate pdf reports, especially daily report, and monthly sales. It uses QuestPDF library to generate the pdf.	Generate daily report ()

Users	<ul style="list-style-type: none"> • GuidId • String Username • String Password hash • Role Role • Bool Hasinitialpassword • Datetime Createdat • Guid Createdby 	The class represent the user entity.It covers the properties related to user information which includes unique identifier,username,password hash,role, initial password and soon.	Newguid()
User Service	<ul style="list-style-type: none"> • String SeedUser name • String SeedPassword 	This class provides the functionality to the user management in system.This class also contains the user related logic, such as password hashing, verification allowing other part of the system to engage with userdata through these methods.	Saveall() Getall() Create() Seed user() Getbyid() Delete() Login() Changepassword()
Utils	<ul style="list-style-type: none"> • Char SegmentDelimeter 	This class provide different helper methods for the common task in bislerium cafes app like Encryption methods and file path methods.	Hashsecret() Verifyhash() Getdirectorypath() Getappusersfilepath() Getappcoffeefilepath() Getappaddinsfilepath()

			Getappordersdirectory path()
--	--	--	---------------------------------

Table 1: Table of methods with descriptions.

4. Implementation of Desktop Application

4.1. Password-protected access, with an additional layer of password security required for making changes to the prices of coffee options and add-ins.

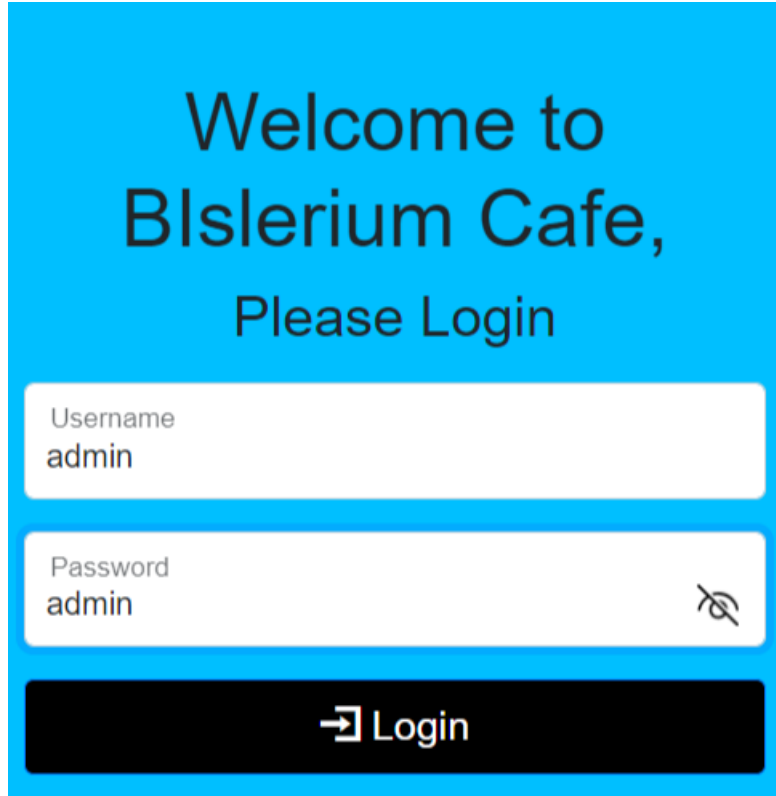
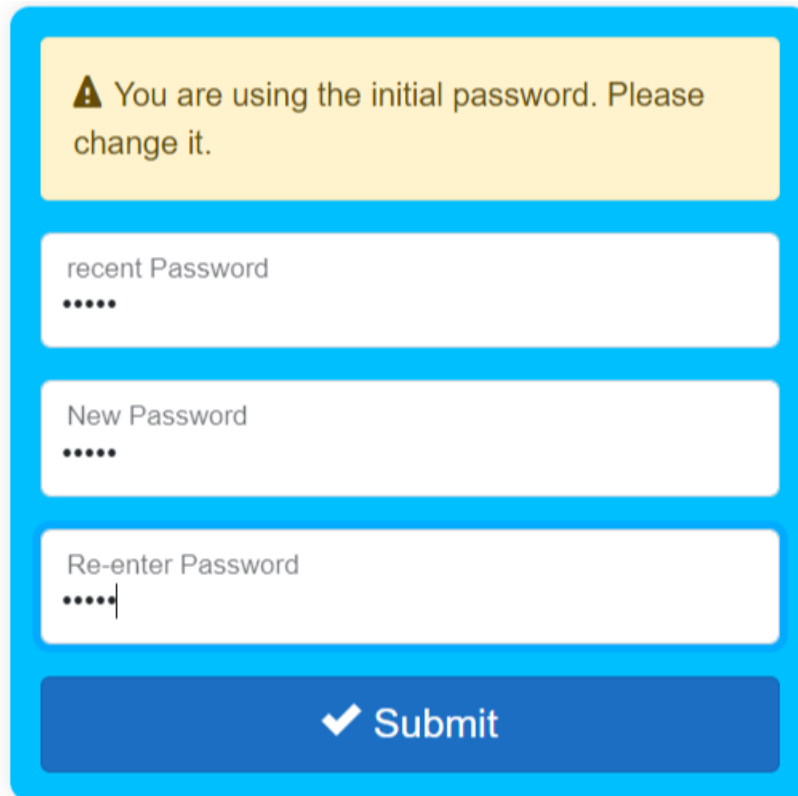
The image shows a login interface for 'Blslerium Cafe'. It has a bright blue background. At the top, the text 'Welcome to Blslerium Cafe, Please Login' is centered in a large, black, sans-serif font. Below this, there are two white input fields with rounded corners. The first field is labeled 'Username' and contains the text 'admin'. The second field is labeled 'Password' and contains the text 'admin'. To the right of the password field is a small icon of an eye with a diagonal line through it, indicating a toggle for password visibility. At the bottom of the form is a large black button with rounded corners. On the button, there is a white icon of a right-pointing arrow inside a square, followed by the word 'Login' in white text.

Figure 14: admin login.



⚠ You are using the initial password. Please change it.

recent Password
.....

New Password
.....

Re-enter Password
.....|

✓ Submit

Figure 15: change recent password of admin.

4.2. A working view to order coffee(s).

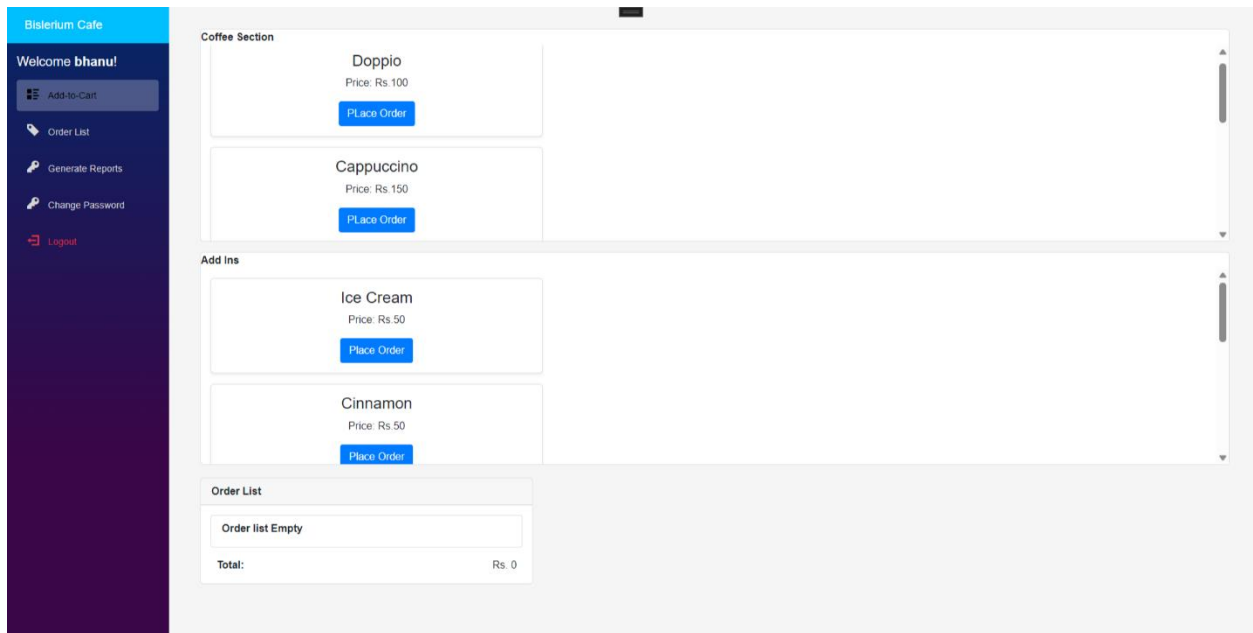




Figure 16: empty the order list before placing an order in add-to-cart.

Order List

Doppio	Price: Rs. 100	
Ice Cream	Price: Rs. 50	

Total: Rs. 150

Customer Phone




 Search for customer for discounts


Figure 17: add-in before customer details.

Order List

Doppio	Price: Rs. 100	
Ice Cream	Price: Rs. 50	


Total: Rs. 150

Customer Phone

 No Free Coffee

Free Coffee?: No

Discount: Rs. 0

 No discount available

Grand Total: Rs. 150

Figure 18: add-to cart items.

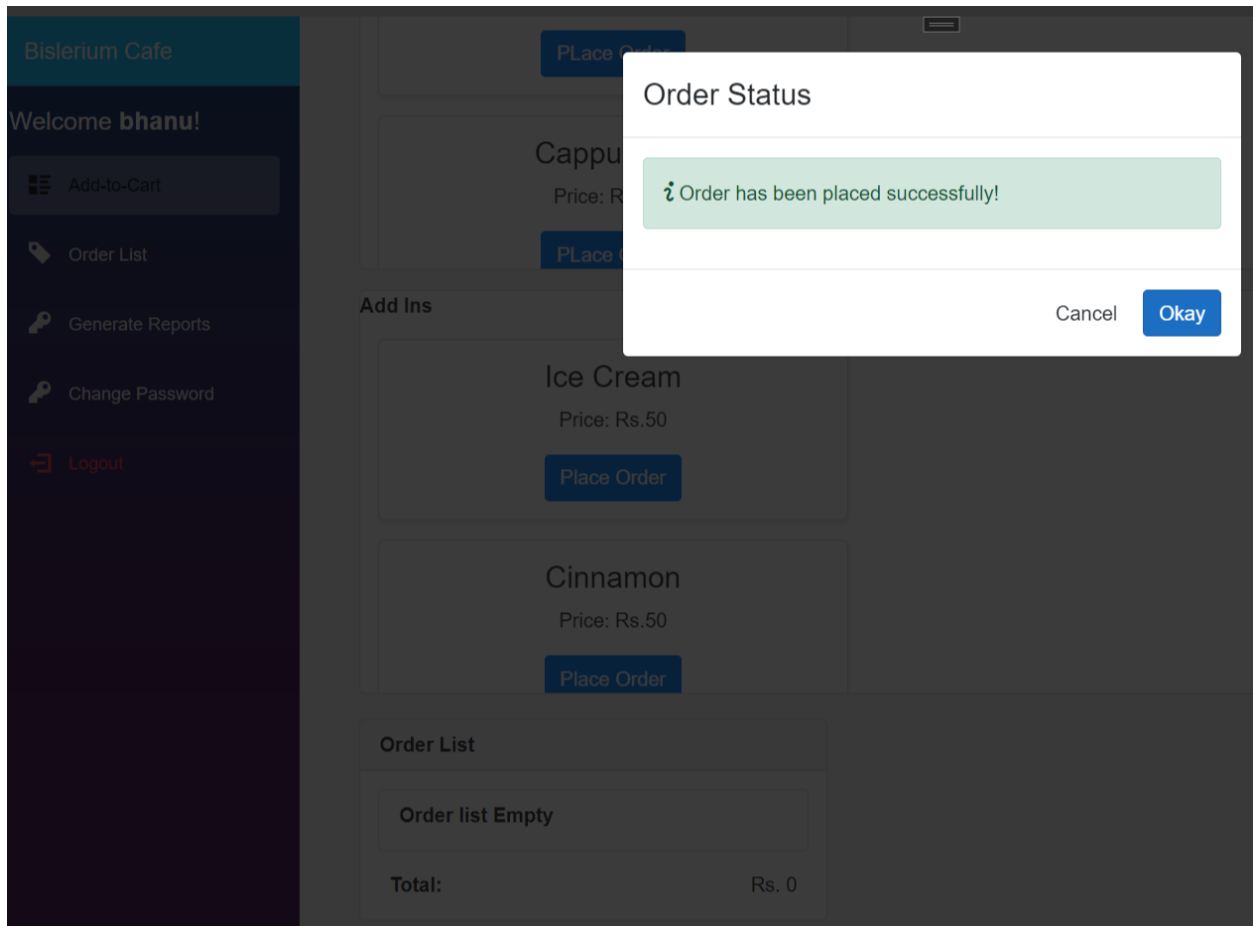


Figure 19: pop-up message as order is successfully placed.

4.3. Ability to add multiple add-ins.

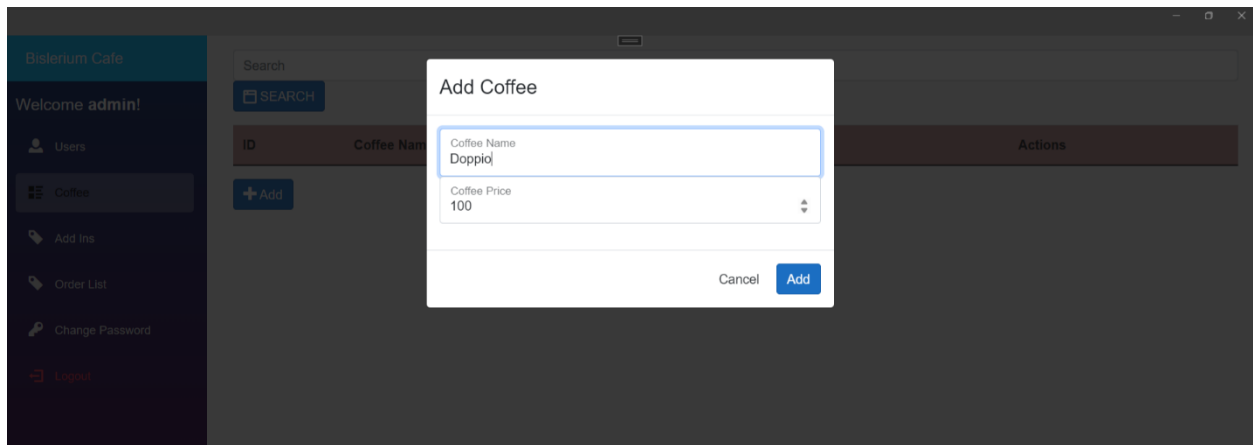


Figure 20: Adding coffee.

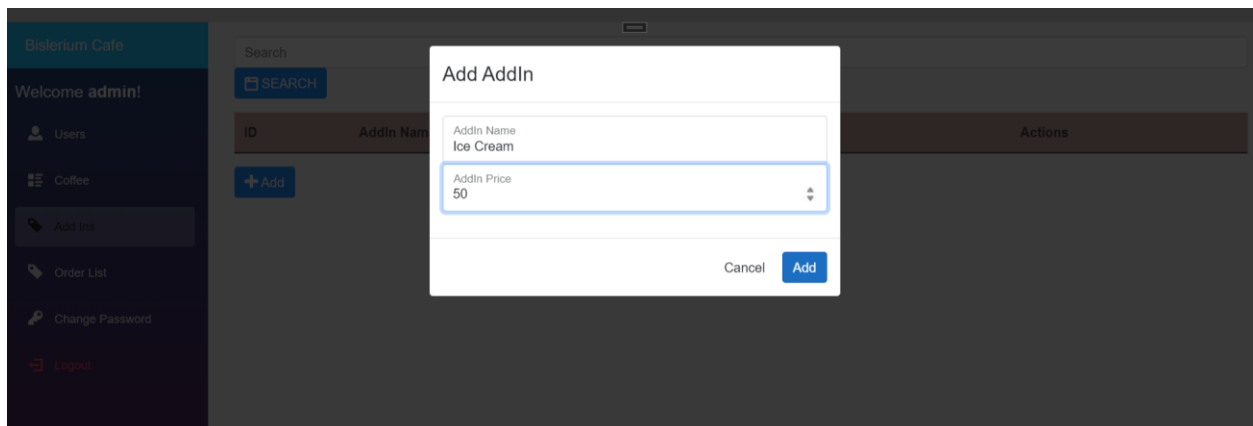
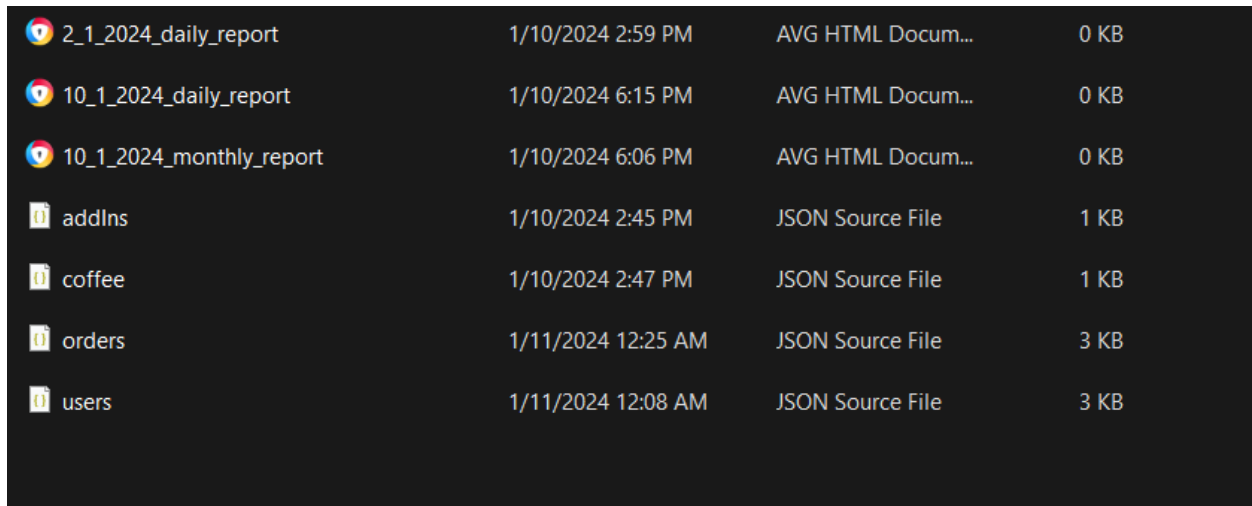


Figure 21: Adding add-ins.

4.4. Application's state is saved and restored using any semi structured formats like JSON, CSV, or Excel.










 2_1_2024_daily_report	1/10/2024 2:59 PM	AVG HTML Docum...	0 KB
 10_1_2024_daily_report	1/10/2024 6:15 PM	AVG HTML Docum...	0 KB
 10_1_2024_monthly_report	1/10/2024 6:06 PM	AVG HTML Docum...	0 KB
 addIns	1/10/2024 2:45 PM	JSON Source File	1 KB
 coffee	1/10/2024 2:47 PM	JSON Source File	1 KB
 orders	1/11/2024 12:25 AM	JSON Source File	3 KB
 users	1/11/2024 12:08 AM	JSON Source File	3 KB

Figure 22: .pdf file for daily as well as monthly reports and JSON files.

4.5. Membership registration with applied benefits like free coffee redemption and discounts.

Customer Phone

9824975836

Search

🔔 No Free Coffee

Free Coffee?: No

Discount: Rs. 0

⚠️ No discount available

Grand Total: Rs. 150

Check Out

Figure 23: discount not available for normal user or non-member.

4.6. A table to display all the purchases/sales transactions.


<input type="text" value="Search"/>							
 SEARCH							
S.N	Coffee name	Add-In List	Customer-Id	Order-Date	Total Amount	Discount	Total
1	Doppio	Ice Cream	1234	1/10/2024 2:27:38 PM	300	0	300
	Doppio	Ice Cream					
2	Doppio	Ice Cream	9824975836	1/10/2024 2:46:02 PM	150	0	150
3	Doppio	Ice Cream	321	1/10/2024 2:51:44 PM	380	0	380
	Cappuccino	Honey					
4	Doppio	Ice Cream	321	1/10/2024 3:37:18 PM	150	0	150
5	Doppio	Ice Cream	9824975836	1/10/2024 5:59:09 PM	300	0	300
	Doppio	Ice Cream					
6	Doppio	Ice Cream	1234	1/10/2024 6:04:54 PM	300	0	300
	Cappuccino						
7	Doppio	Ginger	321	1/10/2024 6:15:08 PM	290	0	290
	Cappuccino						

Figure 24: table of purchases and sales transactions.

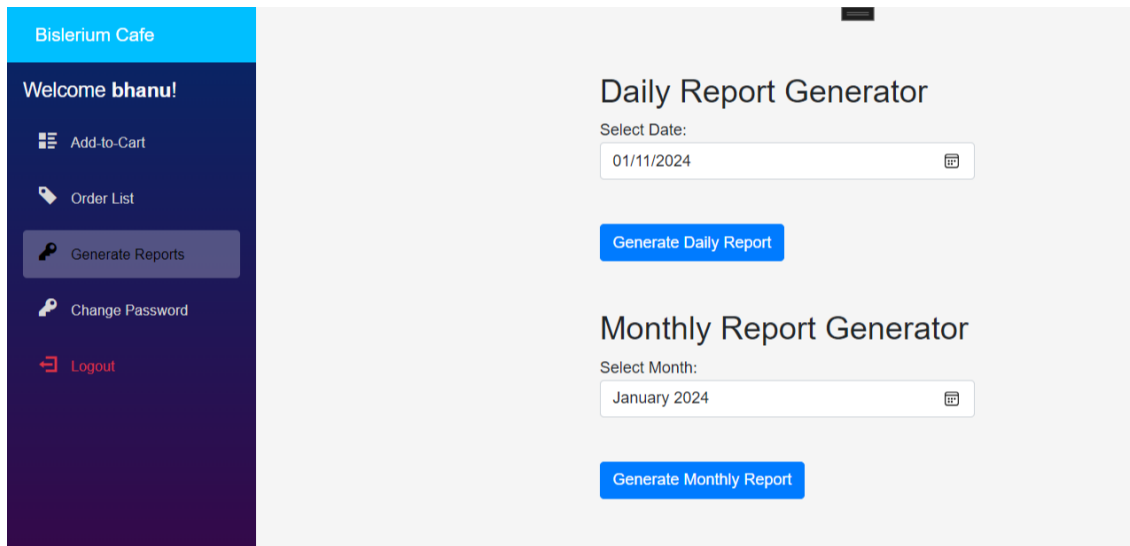
4.7. generate file that includes daily or specific month purchases/sales.

Figure 25: generate reports of the day and month.

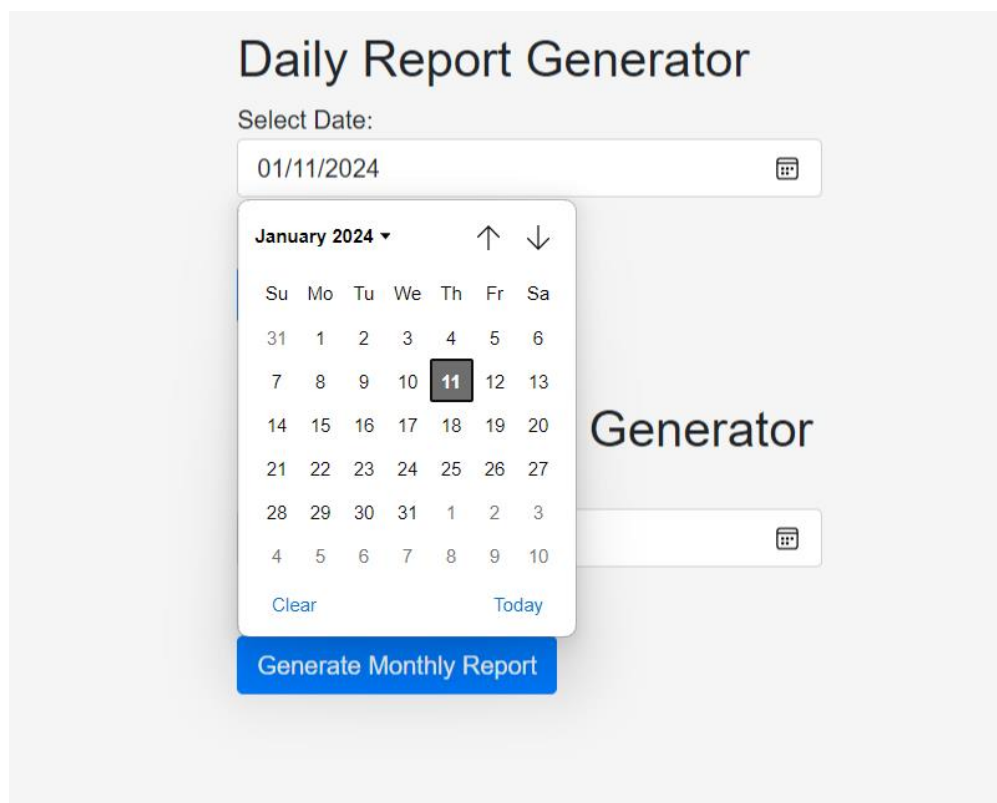


Figure 26: select today date for today sale report.

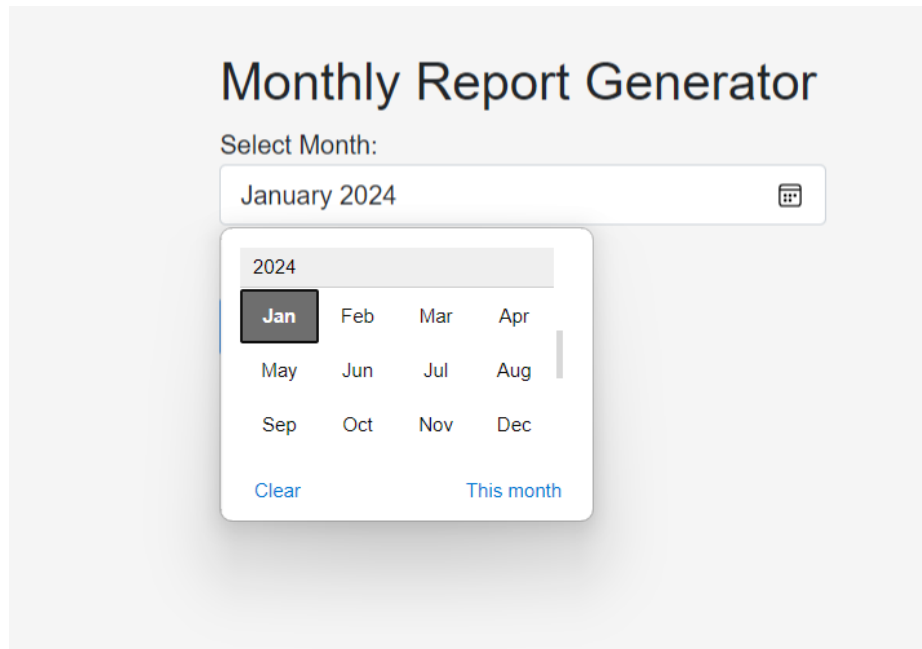


Figure 27: select monthly date for today sale report.








 2_1_2024_daily_report	1/10/2024 2:59 PM	AVG HTML Docum...
 10_1_2024_daily_report	1/10/2024 6:15 PM	AVG HTML Docum...
 10_1_2024_monthly_report	1/10/2024 6:06 PM	AVG HTML Docum...
 addIns	1/10/2024 2:45 PM	JSON Source File
 coffee	1/10/2024 2:47 PM	JSON Source File
 orders	1/11/2024 12:25 AM	JSON Source File
 users	1/11/2024 12:08 AM	JSON Source File

Figure 28: monthly sales report generated.

Conclusion

The development of the **Bislerium Café** Point of Sale (POS) computer program utilizing C# and Visual Studio has demonstrated to be an exceedingly viable and fulfilling endeavours, tending to the cafe's complex necessities with artfulness. C#'s object-oriented worldview encouraged the creation of an adaptable and viable codebase, whereas Visual Studio's coordinates suite of improvement instruments, counting investigating and code route highlights, essentially assisted the improvement preparation. Despite experiencing challenges, such as the complexity of the participation rationale and time limitations related to NFC integration, the advancement group adeptly explored these obstacles. The arrangement permitting clients to supply elective identifiers, such as usernames or phone numbers, showcased a down-to-business approach to overcoming usage imperatives inside the given time outline. All through the improvement handle, thorough testing, counting boundary and edge case scenarios, was vital in guaranteeing the unwavering quality of the computer program. Visual Studio's investigating instruments played a significant part in recognizing and settling issues expeditiously, contributing to the general strength of the application.

In conclusion, the combination of C# and Visual Studio has demonstrated to be a great choice for creating the **Bislerium Café** POS program. The venture has effectively conveyed a feature-rich, secure, and user-friendly application that adjusts consistently with the cafe's prerequisites. The collaboration of these devices empowered the advancement group to form an arrangement that not as it were meets the quick needs of the cafe but moreover gives a strong establishment for future upgrades and adaptability.