AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Dept. of Computer Science
Faculty of Science and Technology

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Fall 2024-2025

Section: [*B*]

Group No: 7

Project Report On

*Project Name [**CD Shop Management System**]*

Supervised By

Md. Hasibul Hasan

Submitted By:

Name	ID
1. Sajid Hasan Mahir	**22-49485-3**
2. Md Eftikhar Mahbub	**22-47635-2**
3. Thonmoy Banik	**20-42701-1**

Obtained Marks for CO2 and CO3 (Description given in the following page)

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)

Evaluation Criteria (CO2)	Total =	Evaluation Criteria (CO3)	Total =
Requirement fulfillment		Organization of the application	
Validation		Representation and Integration of Database	
Verification		Graphical User Interface	

CO2: Display and verify the mean of a real-life Project using the concepts of C# Graphical User Interface based environment with database integration to depict a desktop-based application.					
Assessment Criteria	**Not Attended/ Incorrect (0)**	**Inadequate (1-2)**	**Average (3)**	**Good (4)**	**Excellent (5)**
Evaluation Criteria	**Evaluation Definition**				
Requirement fulfillment	Fails to demonstrate any understanding of real-life scenario-based project development or functional requirement identification. There is no attempt to depict a project or identify functional requirements accurately.	Demonstrates limited understanding of real-life scenario-based project development and functional requirement identification. The project depicted lacks coherence or relevance to real-life scenarios, and functional requirements are inaccurately identified or insufficiently described.	Presents a basic depiction of a real-life scenario-based project and identifies some functional requirements. However, the project lacks depth or complexity, and some functional requirements may be vaguely defined or missing key details.	Effectively demonstrates a realistic scenario-based project and accurately identifies most functional requirements. The project is well-developed with appropriate complexity, and functional requirements are clearly articulated with relevant details.	Exhibits an exceptional understanding of real-life scenario-based project development and accurately identifies all functional requirements. The project is meticulously developed with thorough attention to detail, reflecting a comprehensive understanding of Object-Oriented Programming project development activities.
Validation	Fails to demonstrate any understanding or implementation of validation forms in their system. There is no attempt to deal with data validation, and validation requirements are completely ignored or incorrectly applied.	Demonstrates limited understanding of validation forms and data validation techniques. While some attempt may be made to implement validation, it is incomplete or poorly executed, leading to inadequate handling of data validation.	Shows a basic understanding of validation forms and data validation techniques. They attempt to implement validation, but some aspects may be missing or incorrectly implemented, resulting in partial or inconsistent handling of data validation.	Effectively demonstrates the use of validation forms and implements data validation techniques. Validation is mostly accurate and comprehensive, ensuring the proper handling of data input and verification in the system.	Exhibits an exceptional understanding and implementation of validation forms and data validation techniques. Validation is meticulously implemented with thorough attention to detail, ensuring robust data validation procedures and contributing to the overall reliability and integrity of the system.
Verification	Fails to demonstrate any attempt to verify the system data or functional requirements. There is no evidence of understanding or implementation	Demonstrates limited understanding of verification processes and data flow in the system. Verification attempts are incomplete or	Shows a basic understanding of verification processes and attempts to verify system data. However, verification efforts may be inconsistent or	Identifies and verifies system data, ensuring proper functional requirements are met. Verification efforts are mostly accurate and thorough, with attention to	Exhibits an exceptional understanding of verification processes and meticulously verifies system data. Verification efforts are comprehensive

	of verification processes, and data flow is not considered.	inaccurate, and there is insufficient consideration given to ensuring data integrity and functionality.	lack thoroughness, and there may be gaps in ensuring proper functional requirements and data flow.	ensuring data integrity and appropriate data flow within the system.	and precise, with a keen focus on ensuring all functional requirements are met and maintaining proper data flow throughout the system.

CO3: Prepare and Explain a real life desktop based application synthesizing several component of C# along with development tools to adhere the given requirements.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	**Evaluation Definition**				
Organization of the application	Fails to identify any suitable real time application or requirements for project development activities related to OOP.	Limited understanding about the project scopes and scenarios or identification of functional requirements.	Lacks depth or relevance to OOP project development activities and may contain inaccuracies. Real-life scenarios are mentioned, but the discussion lacks depth or clarity.	Consider and integrate the idea of several core aspects of the project along with relevance to real-life scenarios. Demonstrating a solid understanding of the application presentation.	Generalize and exhibits an exceptional understanding of project preparation according to a to real-life scenarios. Also contains proper and insightful identification of the system which is comprehensive and precise.
Representation and Integration of Database	Fails to identify and present any understanding or implementation of database. Also failed to integrate the data with the project itself.	Limited understanding of the database concepts or their proper way of using in a real time project. While some attempt may be made to implement but it is incomplete or poorly executed, leading to inadequate design.	Lacks depth or relevance to database integration with the application. Shows a basic understanding but some aspects may be missing or incorrectly implemented, resulting in partial or inconsistency. May lack proper normalization.	Integrate the database with the forms properly and implements it with proper validation which is mostly accurate and comprehensive, ensuring the proper handling of data input and verification along with general normalization.	Exhibits an exceptional understanding and implementation of database ensuring attention to detail, and robust data manipulation procedures and contributing to the overall clarity.
Graphical User Interface	Fails to present or prepare GUI based application interfaces. There is no evidence of creating or integrating such things according to their usefulness.	Limited understanding of graphical user interfaces. Lack of design knowledge. Very poor attempt to make such things which are currently obsolete or can't be identified as coherent.	Shows a basic understanding of creating user interfaces. Most of them are interconnected but maybe some of them lack it. However, most of it can be described as user friendly.	Effectively identifies and meet the consider the simplicity. Design related works are mostly accurate and taken proper attention to ensuring a user-friendly coherent system.	Exhibits an exceptional work design following a high standard of simple and elegant work. Several controls and mechanism has been organized in a preferred way according to the coherent usage .

Table of Contents: Page no.

Introduction

We named our project CD Shop Management System which operates a offline CD Shop named Rockstar CDShop. Our project is a offline software solution to operate a CD selling shop digitally. This is a Desktop application build up with C#, .Net framework.

Here mainly two types of users operates this system. One is the owner which we set as Admin and another is worker which is set as Employee. There is a secured system for the entry of user to the systems. It uses only userId and password for login to the system. An admin can have higher feature than an employee. Only admin can add employee to the system. But an employee does not have that authority add anyone. He can only make bill and see bill records.

The system can generate printable bill that can be saved or given to the customer physically by printing .It also stores all bill information in the systems for record.

The systems fullfills all the basic criteria of object oriented programming.

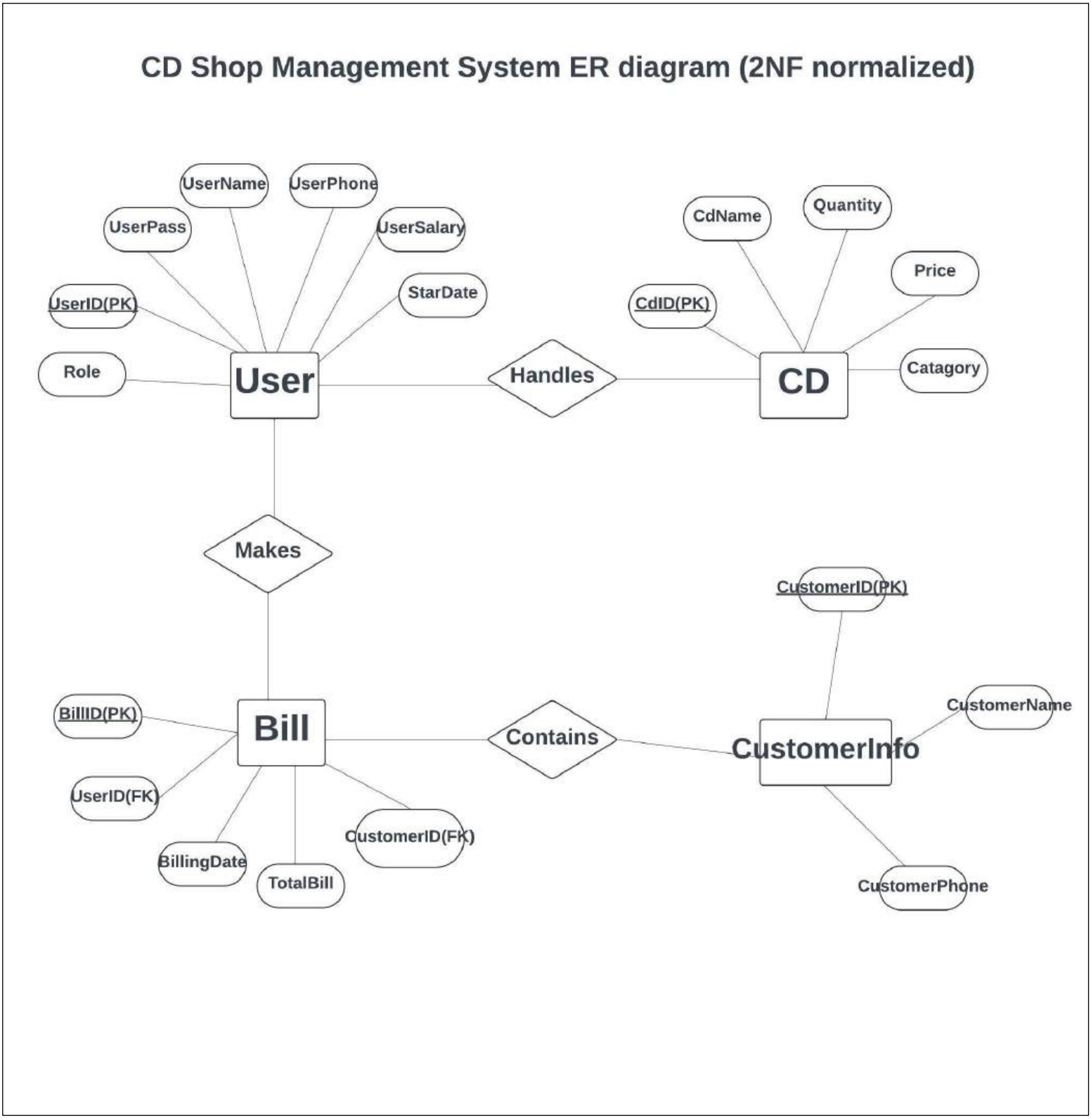
User Story

User Admin:

- ✓ Can add an employee user to the system
- ✓ Can add an another admin user to the system
- ✓ Can remove an employee from the system
- ✓ Can remove other admin from the system
- ✓ Can update an employee information
- ✓ Can update another admin information
- ✓ Can see all user details or information
- ✓ Can add new CD to the system
- ✓ Can remove a CD from system
- ✓ Can update a CD information
- ✓ Can search a specific CD from the system
- ✓ Can search a specific user from the system
- ✓ Can make bill and print bill
- ✓ Can add customer information to the bill
- ✓ Can see all bill information
- ✓ Can search a specific customer
- ✓ Can see total sales

User Employee:

- ✓ Can make bills and print bill
- ✓ Can add customer to the bill
- ✓ Can see all bill details
- ✓ Can search a specific customer
- ✓ Can search a specific CD
- ✓ Can see total sales


CD Shop Management System ER diagram (2NF normalized)
UserName
UserPhone
UserPass
UserSalary
StarDate
UserID(PK)
Role
User
Handles
CdName
Quantity
Price
CdID(PK)
CD
Catagory
Makes
CustomerID(PK)
BillID(PK)
Bill
Contains
CustomerInfo
CustomerName
UserID(FK)
CustomerID(FK)
BillingDate
TotalBill
CustomerPhone

SQL Queries

```
"SELECT * FROM userInfo;

select userInfo.UserID from userInfo order by userInfo.UserID desc;

"SELECT * FROM userInfo where userInfo.UserName like '" +
this.txtSearch.Text + "%';"

"select * from userInfo where UserID = '" + this.txtId.Text + "';";

@"INSERT INTO userInfo(UserID, UserPass, UserName,
UserPhone, UserSalary, StartDate, Role)
                        VALUES('" + this.txtId.Text + @"',
                        '" + this.txtPassword.Text + @"',
                        '" + this.txtName.Text + @"',
                        '" + this.txtPhone.Text + @"',
                        '" + this.txtSalary.Text + @"',
                        '" + this.dtpStartDate.Text + @"',
                        '" + this.cmbType.Text + "');";

"select * from userInfo where UserID = '" + this.txtId.Text + "';";

@"UPDATE userInfo
                    SET UserPass = '" + this.txtPassword.Text + @"',
                    UserName = '" + this.txtName.Text + @"',
                    UserPhone = '" + this.txtPhone.Text + @"',
                    UserSalary = '" + this.txtSalary.Text + @"',
                    StartDate = '" + this.dtpStartDate.Text + @"',
                    Role = '" + this.cmbType.Text + @"'
                    WHERE UserID = '" + this.txtId.Text + "'; ";

"delete from userInfo where UserID = '" + id + "';";

"SELECT
cdInfo.CdID,cdInfo.CdName,cdInfo.Price,cdInfo.Quantity,cdInfo.Cat
agory FROM cdInfo"
```

```
"select cdInfo.CdID from cdInfo order by cdInfo.CdID desc;";

"SELECT * FROM cdInfo where cdInfo.CdID = '" + this.txtId.Text +
"';";

@"INSERT INTO cdInfo(CdName, Quantity, Price, Catagory)
                        VALUES('" + this.txtName.Text + @"',
                        '" + this.txtQuantity.Text + @"',
                        '" + this.txtPrice.Text + @"',
                        '" + this.cmbCategory.Text + "');";

"SELECT * FROM cdInfo where cdInfo.CdID = '" + this.txtId.Text +
"';";

@"UPDATE cdInfo
                    SET CdName = '" + this.txtName.Text + @"',
                    Quantity = '" + this.txtQuantity.Text + @"',
                    price = '" + this.txtPrice.Text + @"',
                    Catagory = '" + this.cmbCategory.Text + @"'
                    WHERE CdID = '" + this.txtId.Text + "'; ";


"delete from cdInfo where CdID = '" + id + "';";

"SELECT * FROM cdInfo where cdInfo.CdName like '" +
this.txtSearch.Text + "%';";

"select CustomerID from customerInfo order by CustomerID desc;";

"select * from cdInfo;")

"SELECT * FROM cdInfo where cdInfo.CdName like '" +
this.txtSearch.Text + "%';";

@"UPDATE cdInfo
                SET Quantity = '" + newQty + @"'
```

```
WHERE CdID = '" + CID + "'; ";
```

```
"select BillID from billInfo order by BillID desc;";
```

```
$"UPDATE cdInfo SET Quantity = Quantity + '{quantity}' WHERE
CdName = '{productName}'";
```

```
"select * from customerInfo where CustomerID = '" +
this.txtCID.Text + "';";
```

```
@"INSERT INTO customerInfo(CustomerName, CustomerPhone)
                    VALUES('" + this.txtCusName.Text + @"',
                    '" + this.txtCusPhone.Text + "');";
```

```
"select * from billInfo where BillID = '" + this.lblBillId.Text + "';";
```

```
"select * from customerInfo where CustomerID = '" +
this.txtCID.Text + "';";
```

```
@"INSERT INTO
billInfo(UserID,BillingDate,TotalBill,CustomerID)
                        VALUES('" + ID + @"',
                        '" + formattedDate + @"',
                        '" + GrdTotal + @"',
                        '" + this.txtCID.Text + "');";
```

```
"SELECT
billInfo.BillID,userInfo.UserID,userInfo.UserName,billInfo.TotalBill,
billInfo.BillingDate,customerInfo.CustomerID,customerInfo.Custome
rName,customerInfo.CustomerPhone FROM billInfo INNER JOIN
customerInfo ON billInfo.CustomerID=customerInfo.CustomerID
INNER JOIN userInfo ON userInfo.UserID=billInfo.UserID;")
```

```
@"
        SELECT
           billInfo.BillID,
           userInfo.UserID,
           userInfo.UserName,
           billInfo.TotalBill,
           billInfo.BillingDate,
           customerInfo.CustomerID,
           customerInfo.CustomerName,
           customerInfo.CustomerPhone
        FROM
           billInfo
        INNER JOIN
           customerInfo
           ON billInfo.CustomerID = customerInfo.CustomerID
        INNER JOIN
           userInfo
           ON userInfo.UserID = billInfo.UserID
        WHERE
           customerInfo.CustomerName LIKE '" + this.txtSearch.Text +
@"%';";


"SELECT * FROM userInfo;")

"SELECT * FROM userInfo where userInfo.UserName like '" +
this.txtSearch.Text + "%';";
```

ScreenShots

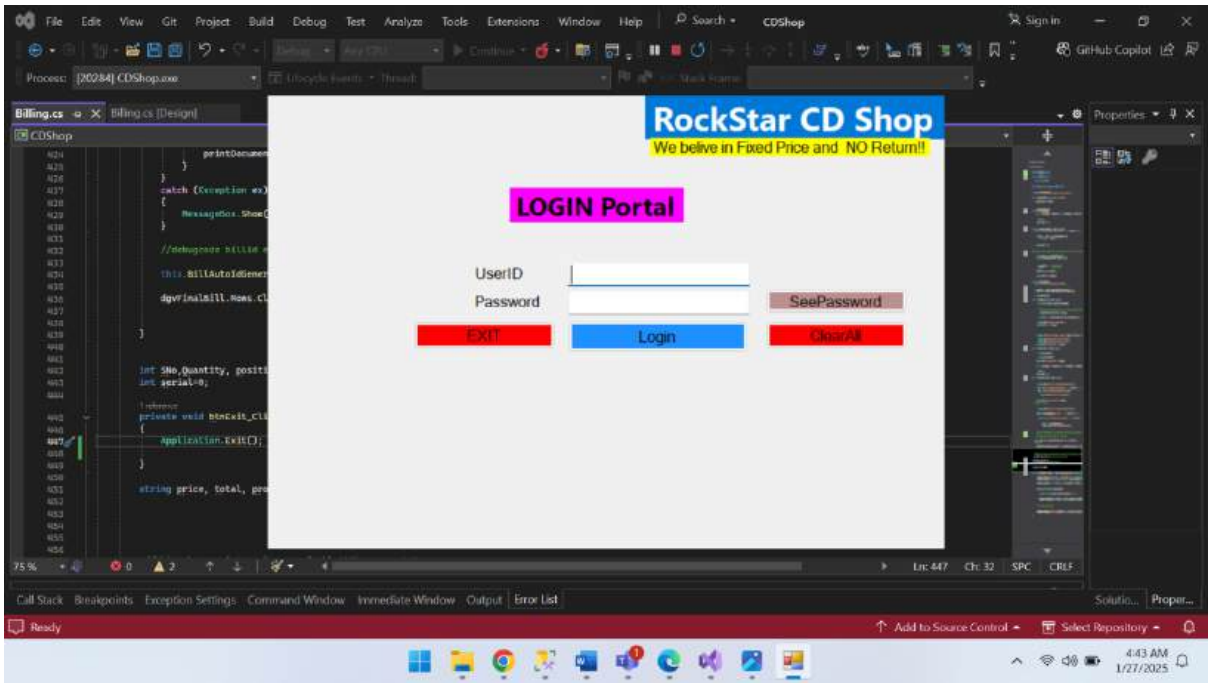


Figure: Login as admin.

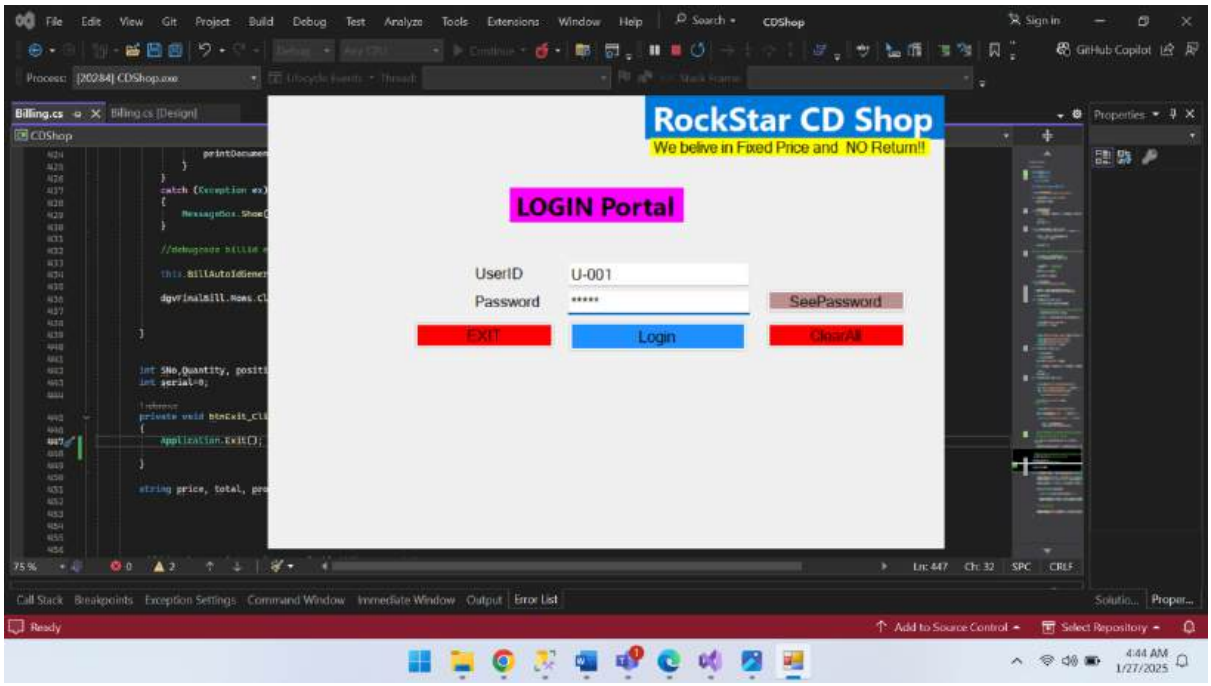


Figure: Login as admin.

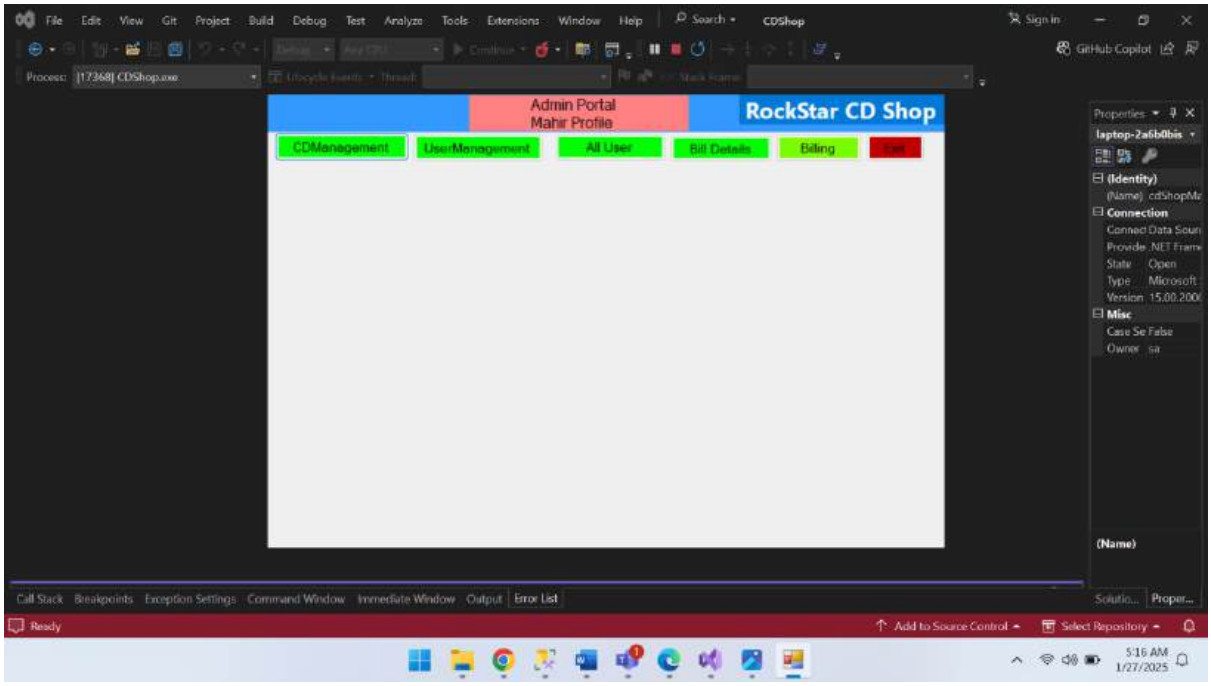


Figure: Admin Portal

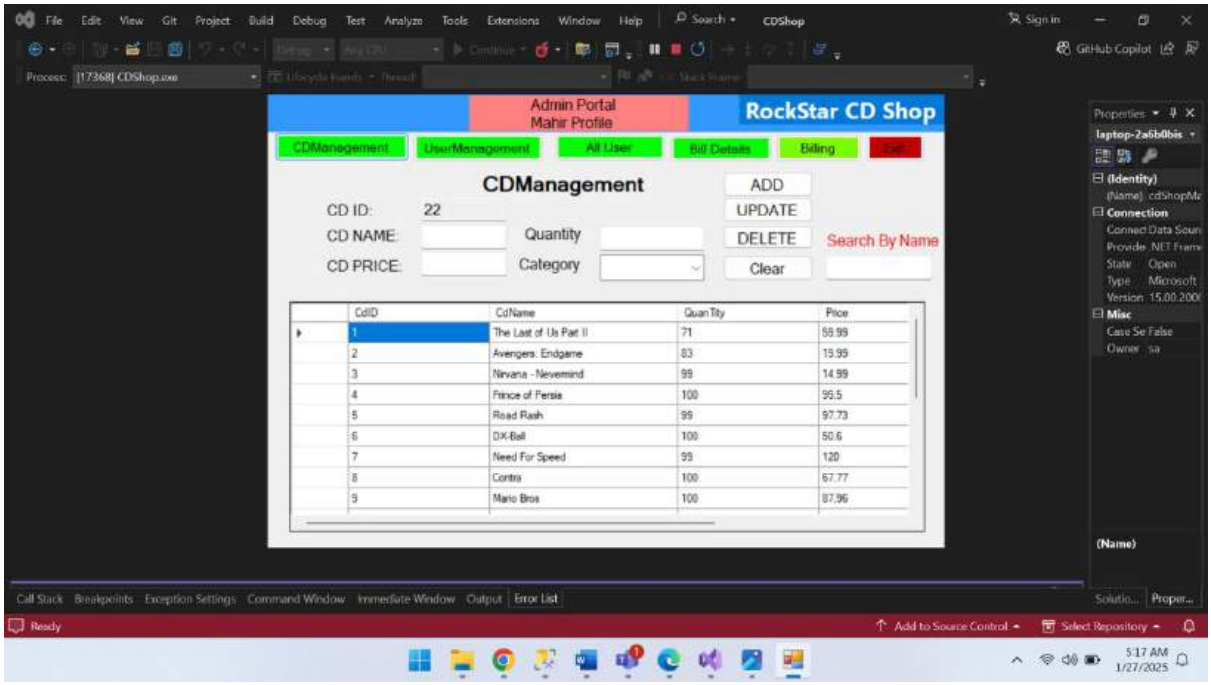


Figure: CdManagement User control

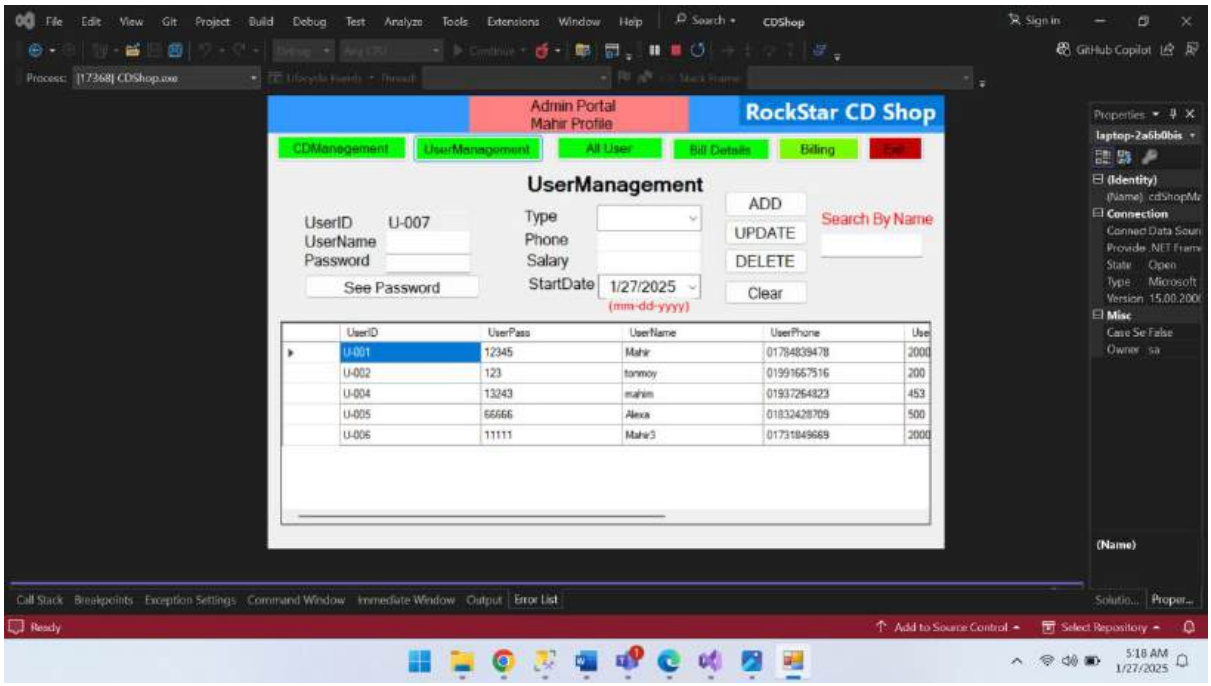


Figure: UserManagement User control.

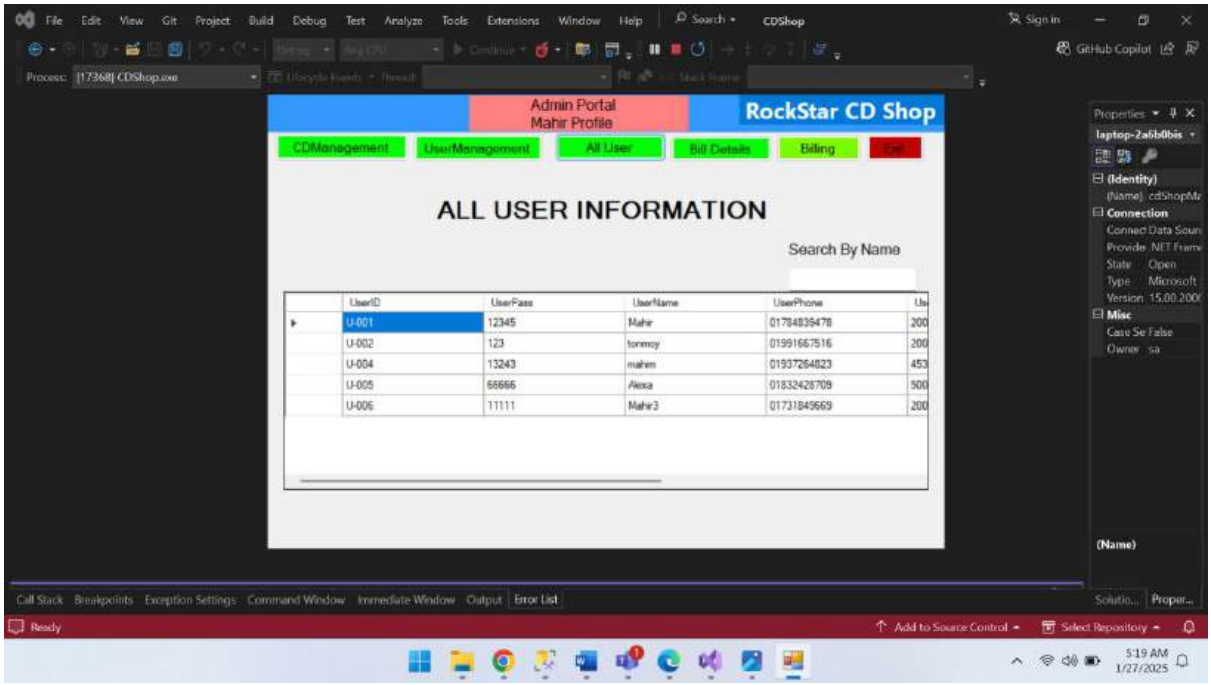


Figure: All User information User Control

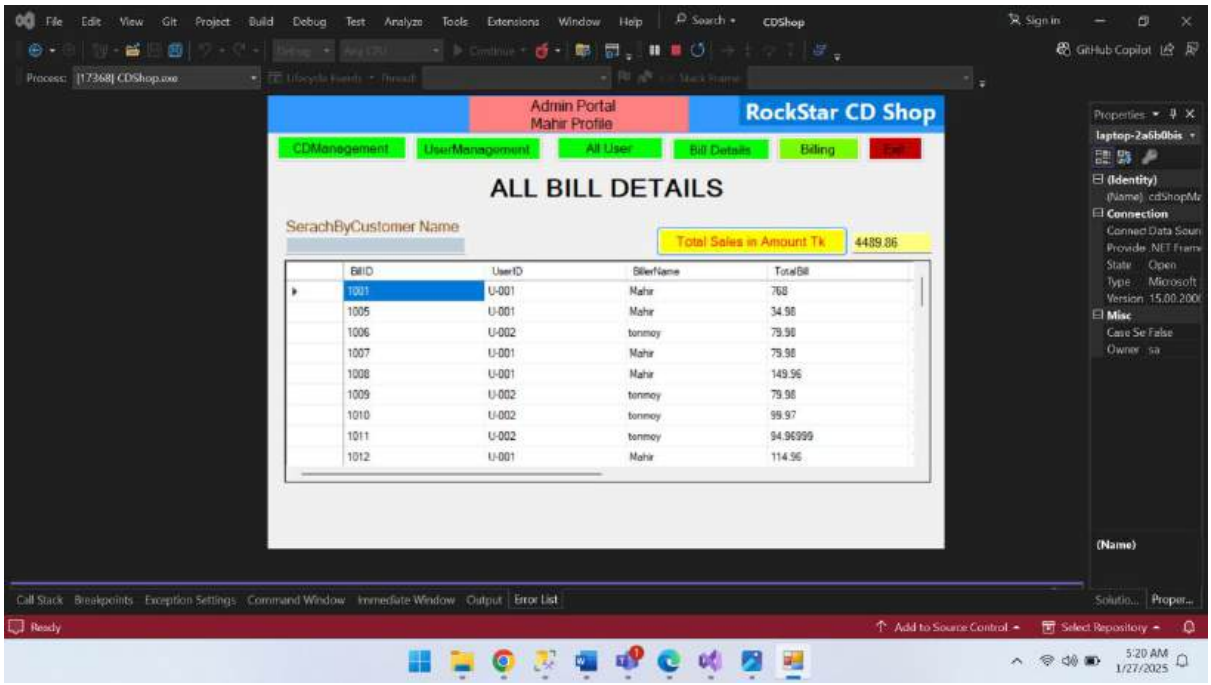


Figure: Bill details User Control.

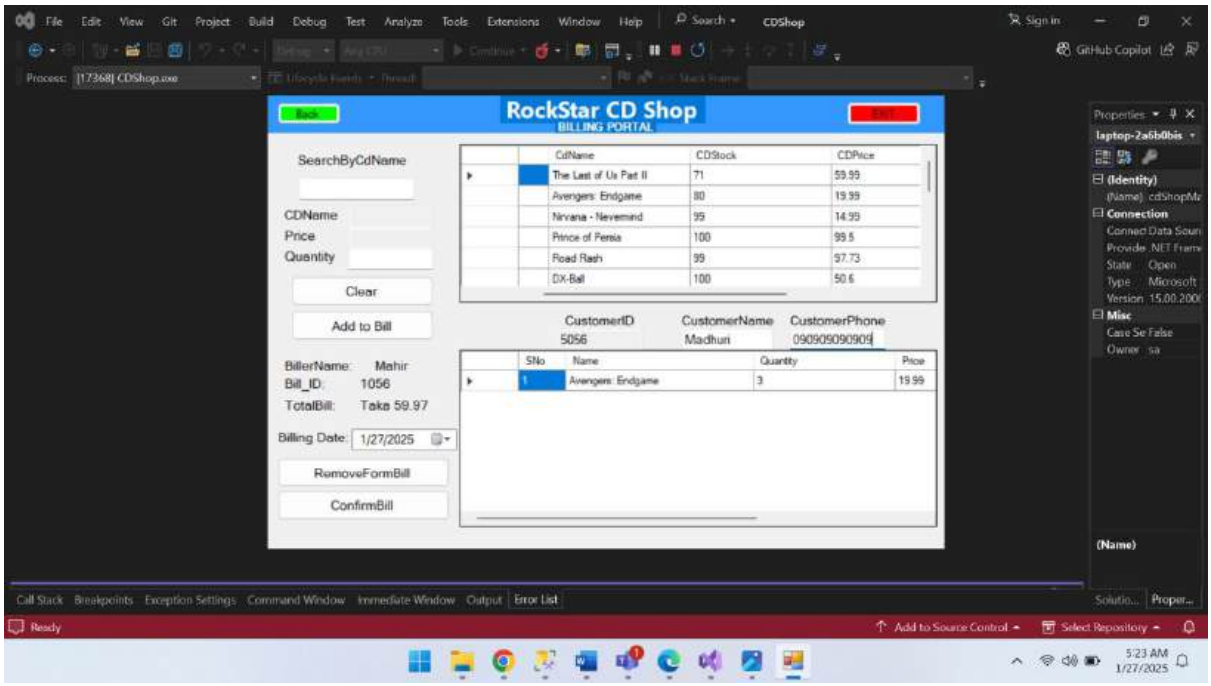


Figure: Billing as an Admin user.

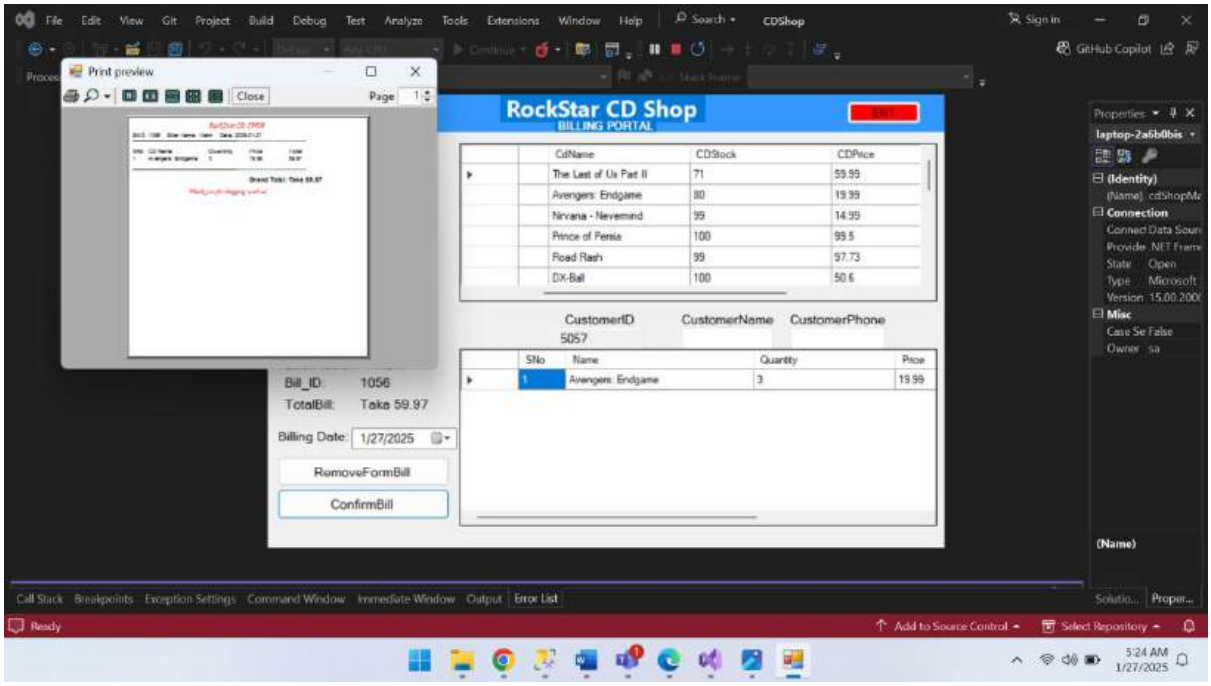


Figure: Bill Printing.

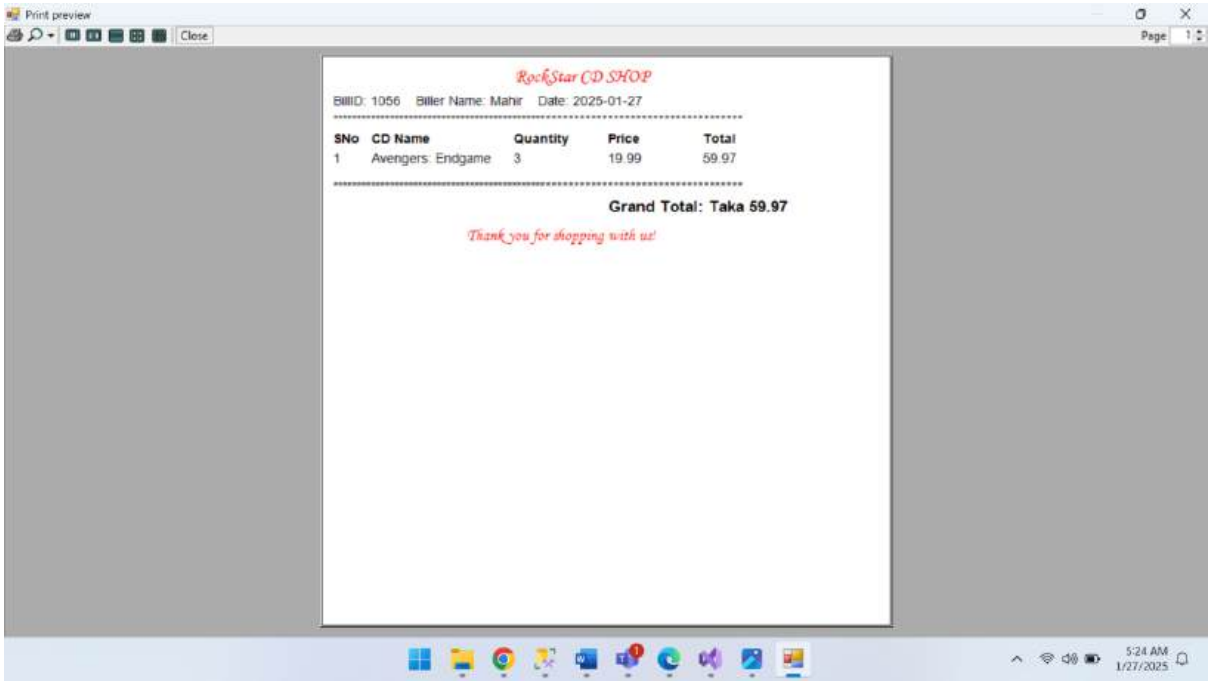


Figure: Bill Copy preview.

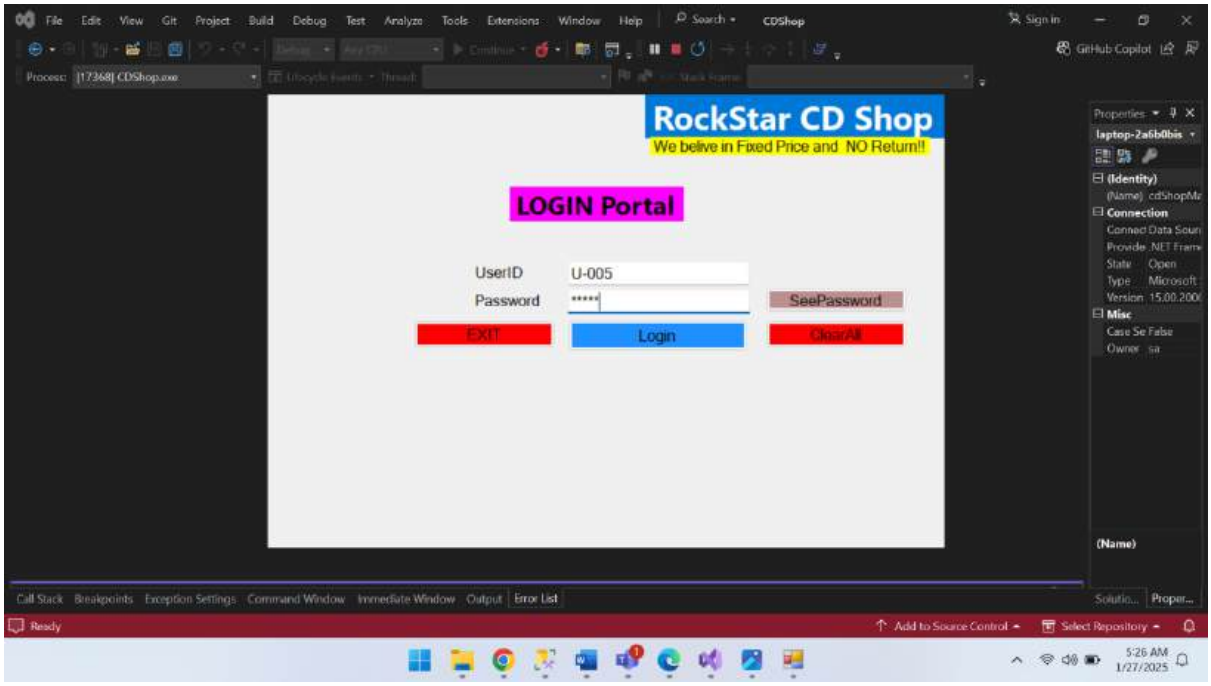


Figure: Login to the system as an Employee user.

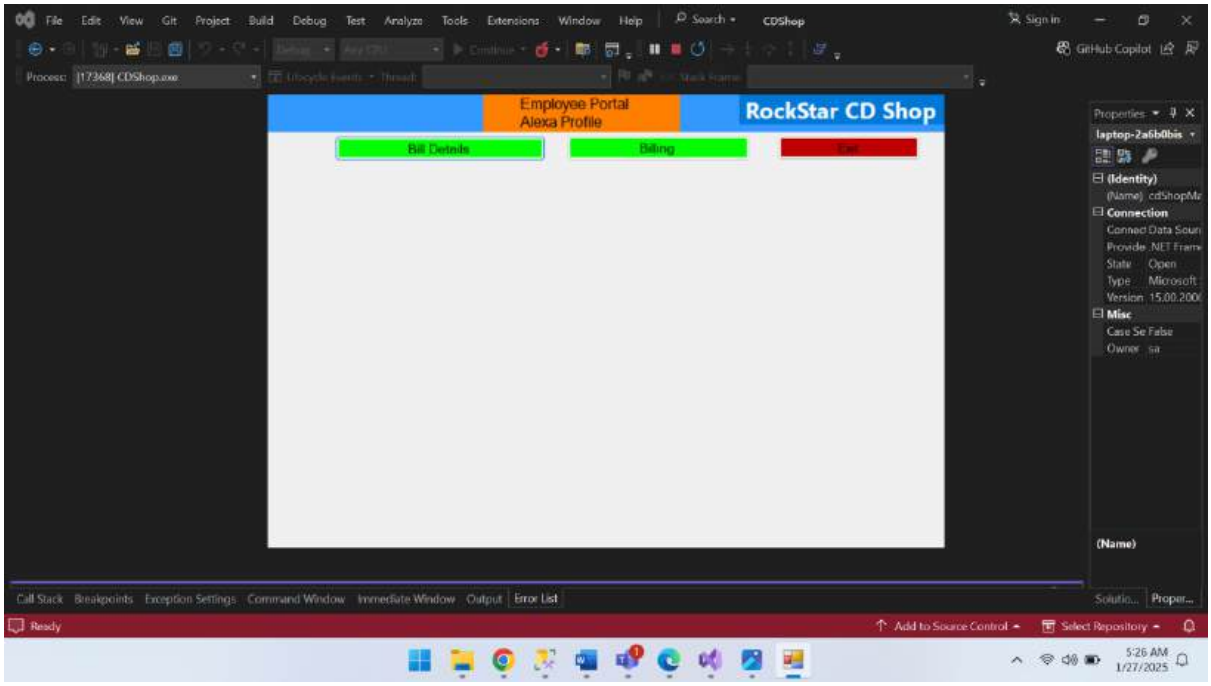


Figure: Employee Portal.

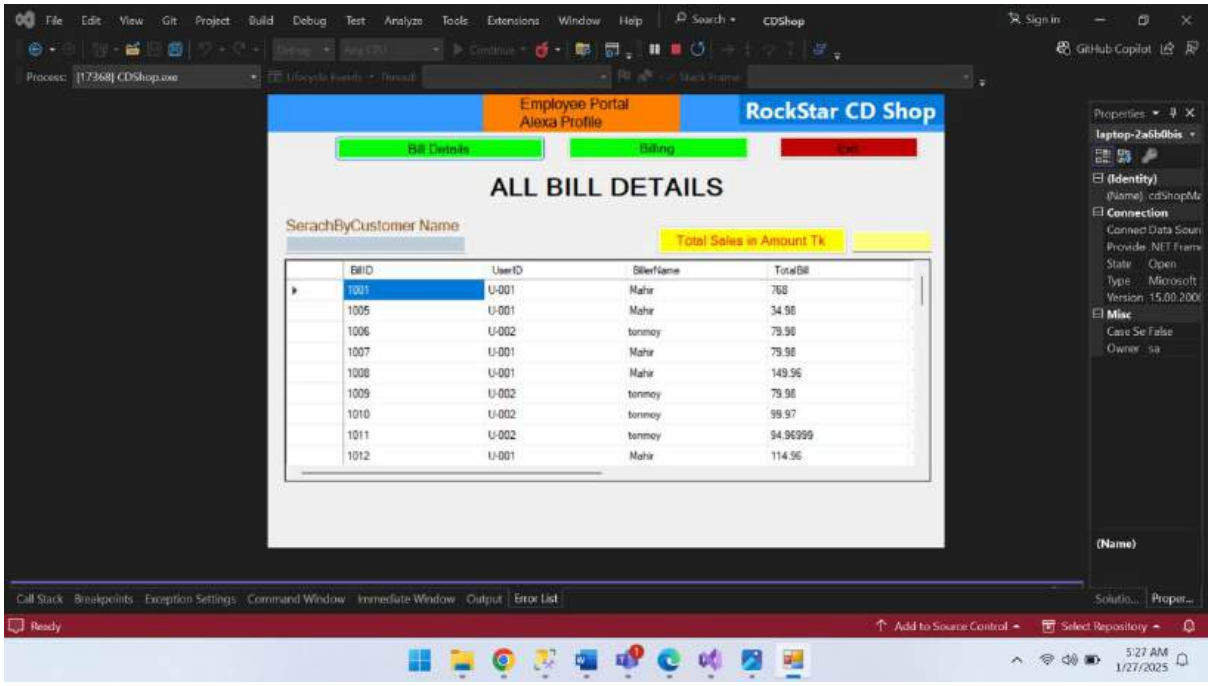


Figure: Bill details User Control.

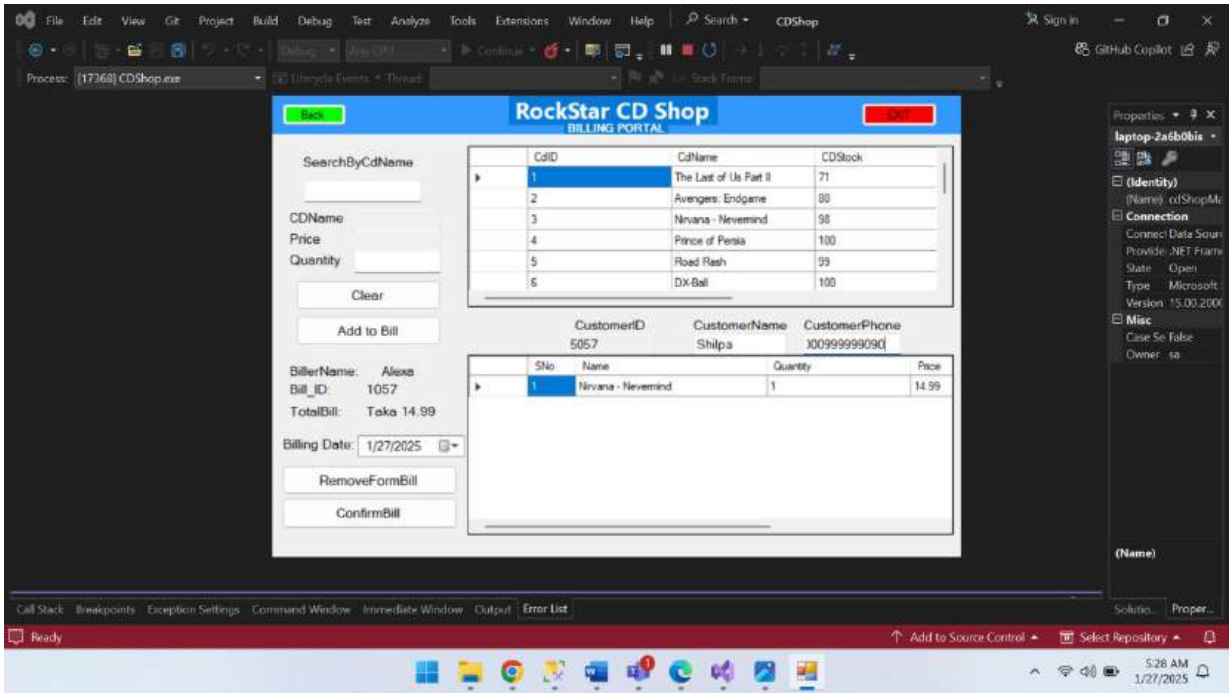


Figure: Billing as an Employee user.

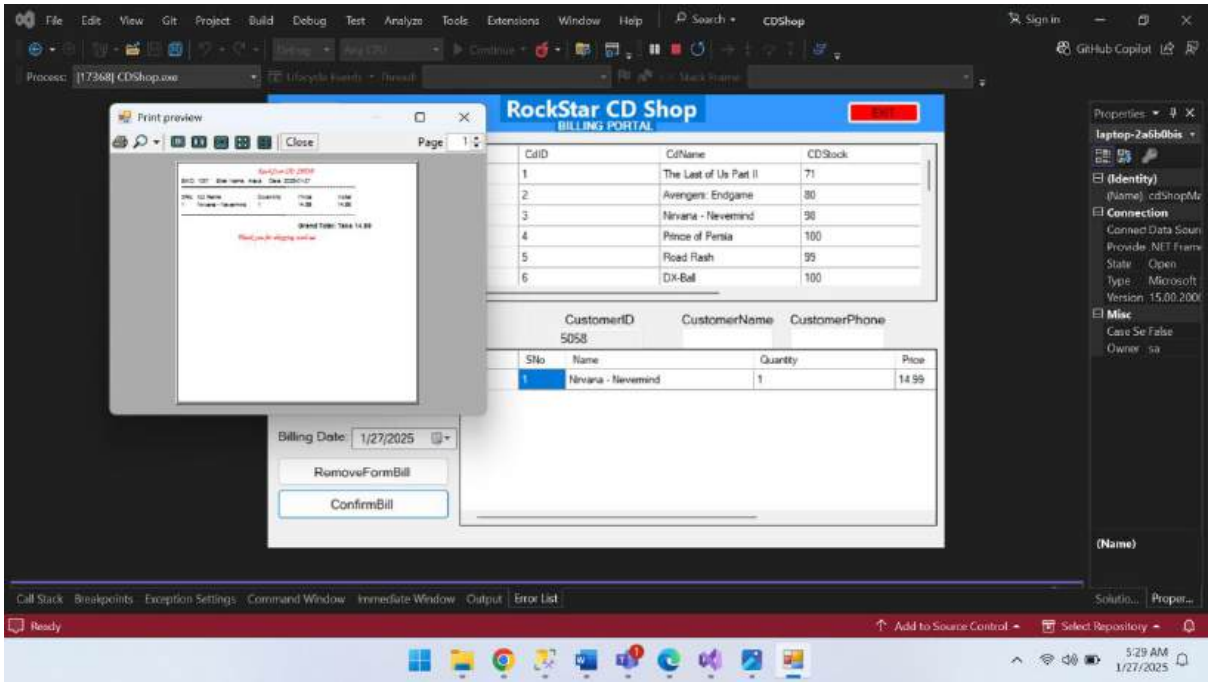


Figure: Bill print.

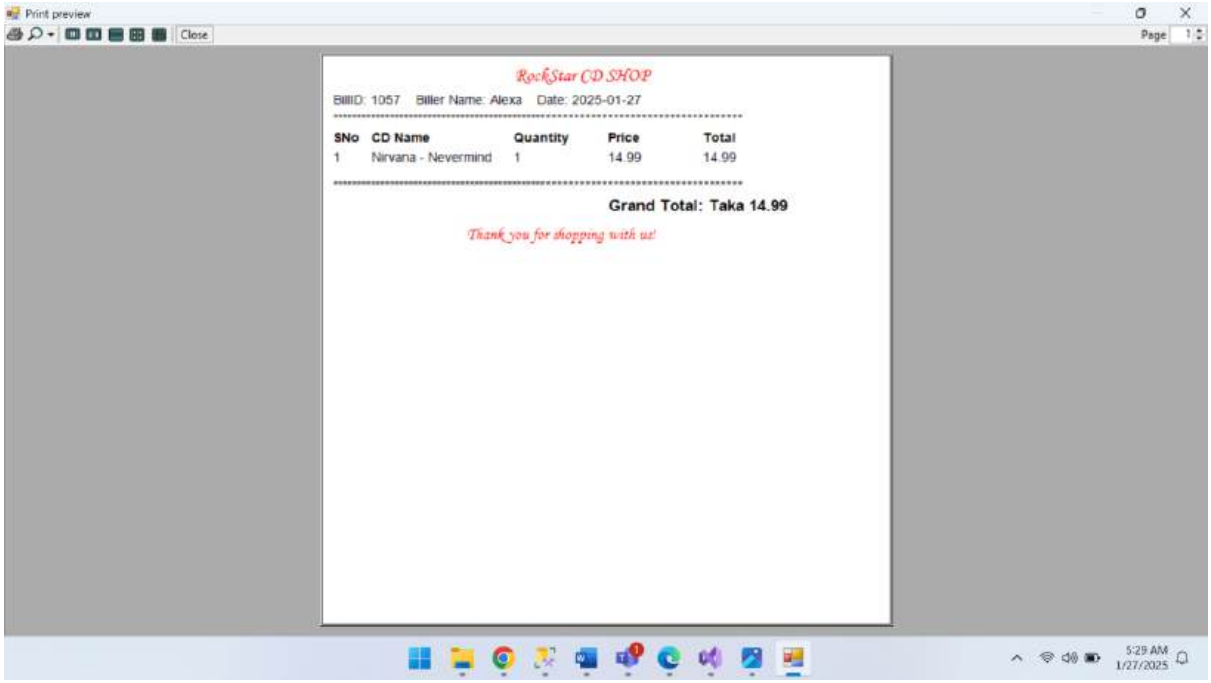


Figure: Print Preview

References:

1. https://youtu.be/5DI-KDxhjYI?si=PUFyFPB-vvXoOho4
2. https://youtu.be/ILGUQBFjZUk?si=3KZKbXpbSpi-kl7Z
3. Md Hasibul Hasan sir’s lectures & codes.