AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Dept. of Computer Science
Faculty of Science and Technology

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Fall 2024-2025

Section: [*E*]

Group No: 01

Project Report On

Project Name [***Inventory Management System*** *]*

Supervised By

Md. Hasibul Hasan

Submitted By:

Name	ID
1. Touhid Ahmed Antu	**22-47604-2**
2. Tahmina Jerin Tinny	**22-47595-2**
3.	

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:

An Inventory Management System (IMS) is a software solution designed to streamline and automate the process of tracking, managing, and optimizing inventory levels in real time. It helps businesses keep track of stock, orders, sales, and deliveries while minimizing errors, reducing costs, and improving efficiency. Whether you're running a small retail store, a manufacturing unit, or an e-commerce business, an IMS is essential for maintaining smooth operations.

User Story: As an Admin, User.

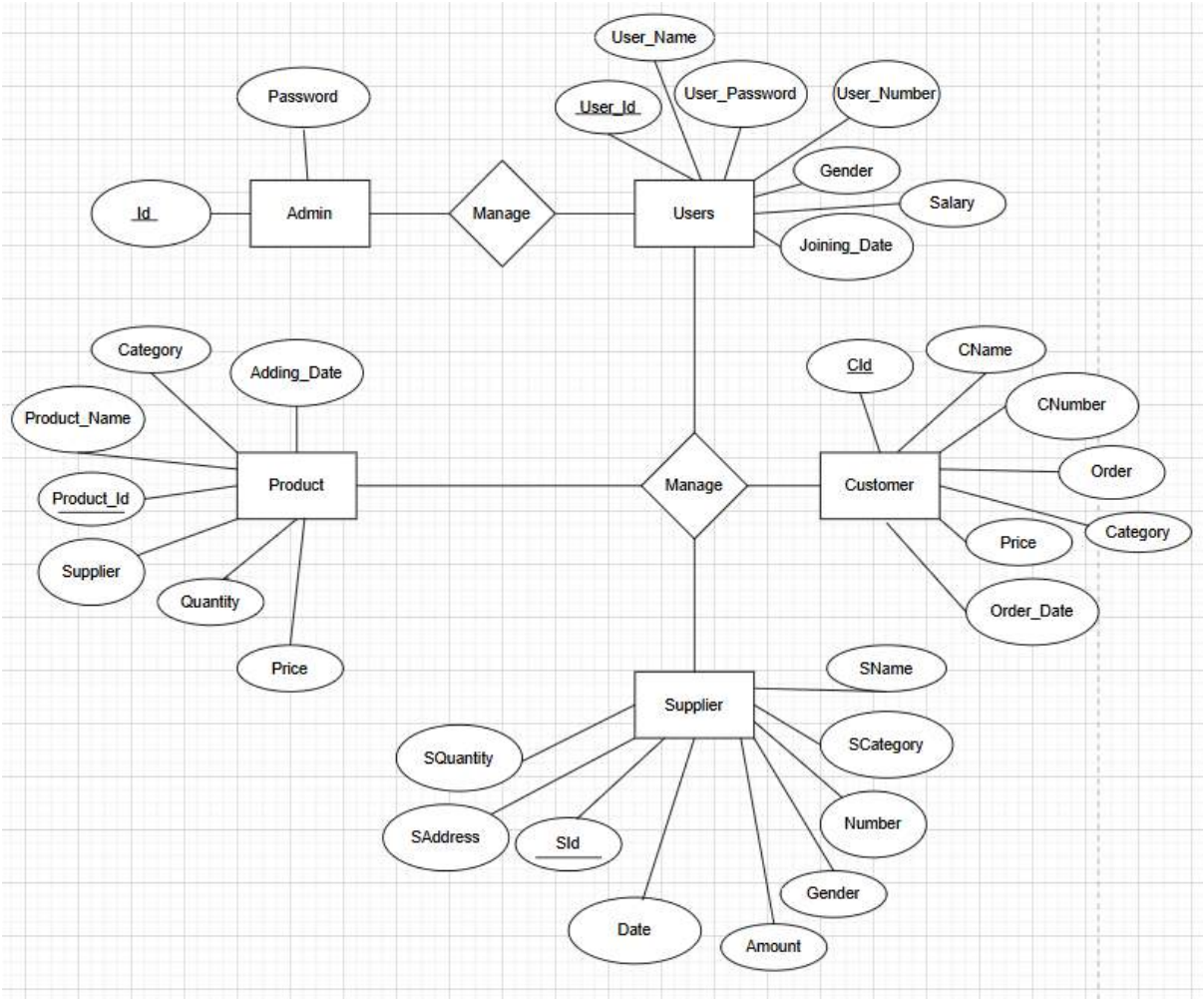
Admin Perspective:

Admin can update his profile. Admin can also update any Users profile. Admin can add or remove any Users. Admin can also see the customer, product, supplier record.

User Perspective:

Users can update their profile. Users can add or remove any product, supplier. User can also see the customer, product, supplier record.

ER Diagram:



Manages
UNF:
ID, UserID, Password, UserID, UserPassword, ProductID, Category, CustomerID, Order, SupplierID, Date, Amount
(Repeating groups for UserID, ProductID, and other entities.)
1NF:
ID, UserID, Password, ProductID, Category, CustomerID, Order, SupplierID, Date, Amount
(All attributes are atomic, but dependencies might still exist.)
2NF:
Split into multiple tables to remove partial dependencies:

1. Admin(ID, Password)
2. Users(UserID, UserPassword, UserName, Gender, Salary, JoiningDate)
3. Product(ProductID, ProductName, Category, AddingDate, Price, Quantity, SupplierID)
4. Customer(CustomerID, CName, CNumber, Order, Price, OrderDate, Category)
5. Supplier(SupplierID, SName, SCategory, SAddress, SQuantity, Date, Amount)

SQL Queries:

CustomerInfo Table

Table Creation

```
CREATE TABLE CustomerInfo (
   id INT PRIMARY KEY,
   Name VARCHAR(255),
   Number VARCHAR(15),
   Email VARCHAR(255),
   Order VARCHAR(255),
   Category VARCHAR(255),
   Quantity INT,
   Address VARCHAR(255),
   Price Float(10, 2)
);
```

Insertion

```
INSERT INTO CustomerInfo (id, Name, Number, Email, Order, Category, Quantity, Address, Price) VALUES (1, 'John Doe', '1234567890', 'johndoe@example.com', 'Order001', 'Electronics', 2, '123 Street, City', 150.50);
```

Update

```
UPDATE CustomerInfo
SET Email = 'newemail@example.com', Address = '456 New Street, City'
WHERE id = 1;
```

Delete

```
DELETE FROM CustomerInfo
```

```
WHERE id = 1;
```

Search

```
SELECT * FROM CustomerInfo
WHERE Name = 'John Doe';
```

ManageProduct Table

Table Creation

```
CREATE TABLE ManageProduct (
    Product_Id INT PRIMARY KEY,
    Product_Name VARCHAR(255),
    Category VARCHAR(255),
    Quatity INT,
    Spplier VARCHAR(255),
    Adding_Date DATE,
    Price DECIMAL(10, 2)
);
```

Insertion

```
INSERT INTO ManageProduct (Product_Id, Product_Name, Category, Quatity,
Spplier, Adding_Date, Price)
VALUES (101, 'Smartphone', 'Electronics', 50, 'TechSupplier', '2025-01-20',
300.00);
```

Update

```
UPDATE ManageProduct
SET Price = 350.00, Quatity = 45
WHERE Product_Id = 101;
```

Delete

```
DELETE FROM ManageProduct
WHERE Product_Id = 101;
```

Search

```
SELECT * FROM ManageProduct
WHERE Category = 'Electronics';
```

ManageSupplier Table

Table Creation

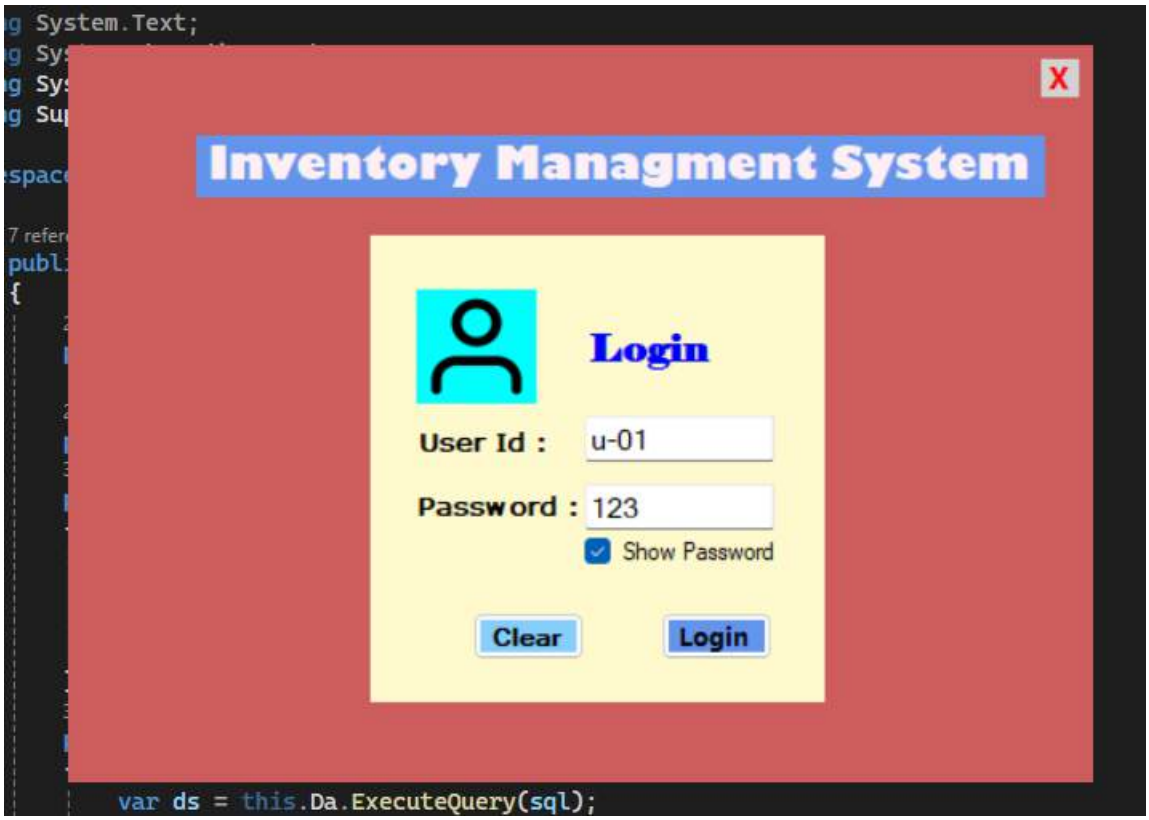
```
CREATE TABLE ManageSupplier (
    SId INT PRIMARY KEY,
    SName VARCHAR(255),
    Number VARCHAR(15),
    SCategory VARCHAR(255),
    SQuantity INT,
    SAddress VARCHAR(255),
    Gender VARCHAR(10),
```

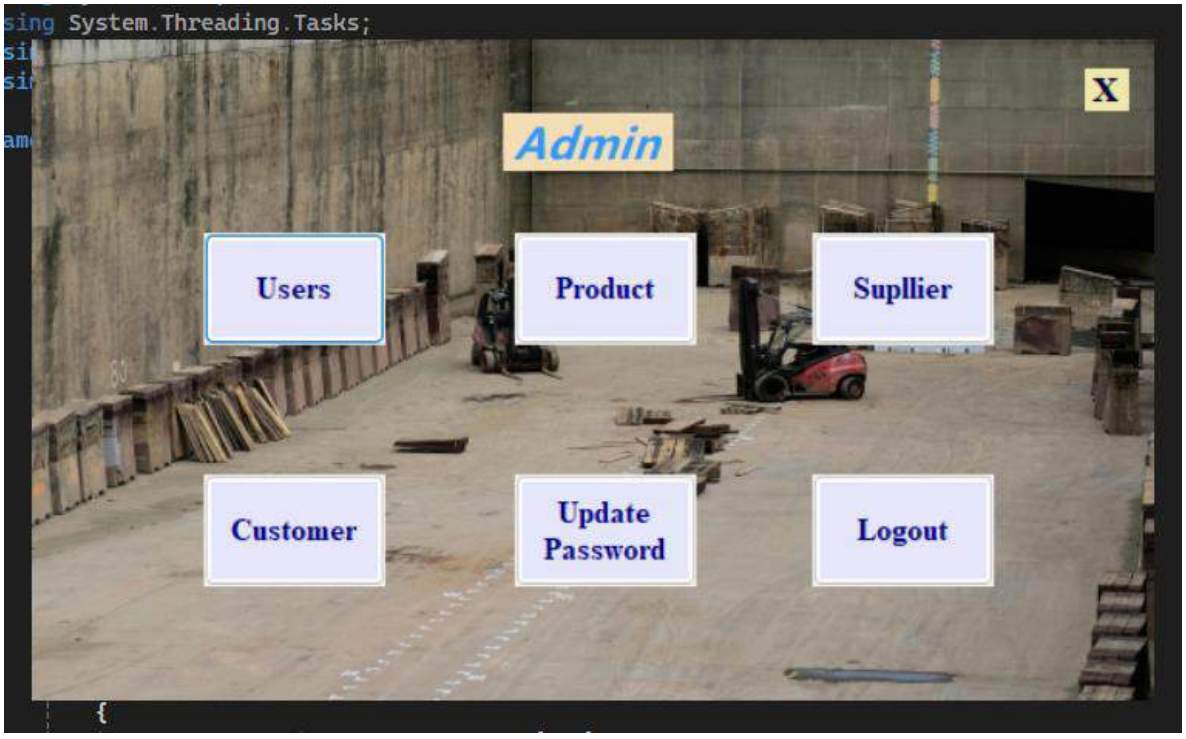
```
    Date DATE,
    Amount DECIMAL(10, 2)
);
```
Insertion
```
INSERT INTO ManageSupplier (SId, SName, Number, SCategory, SQuantity,
SAddress, Gender, Date, Amount)
VALUES (1, 'Supplier A', '9876543210', 'Electronics', 100, '789 Supplier Lane,
City', 'Male', '2025-01-15', 1000.00);
```
Update
```
UPDATE ManageSupplier
SET SQuantity = 120, Amount = 1100.00
WHERE SId = 1;
```
Delete
```
DELETE FROM ManageSupplier
WHERE SId = 1;
```
Search
```
SELECT * FROM ManageSupplier
WHERE SName = 'Supplier A';
```

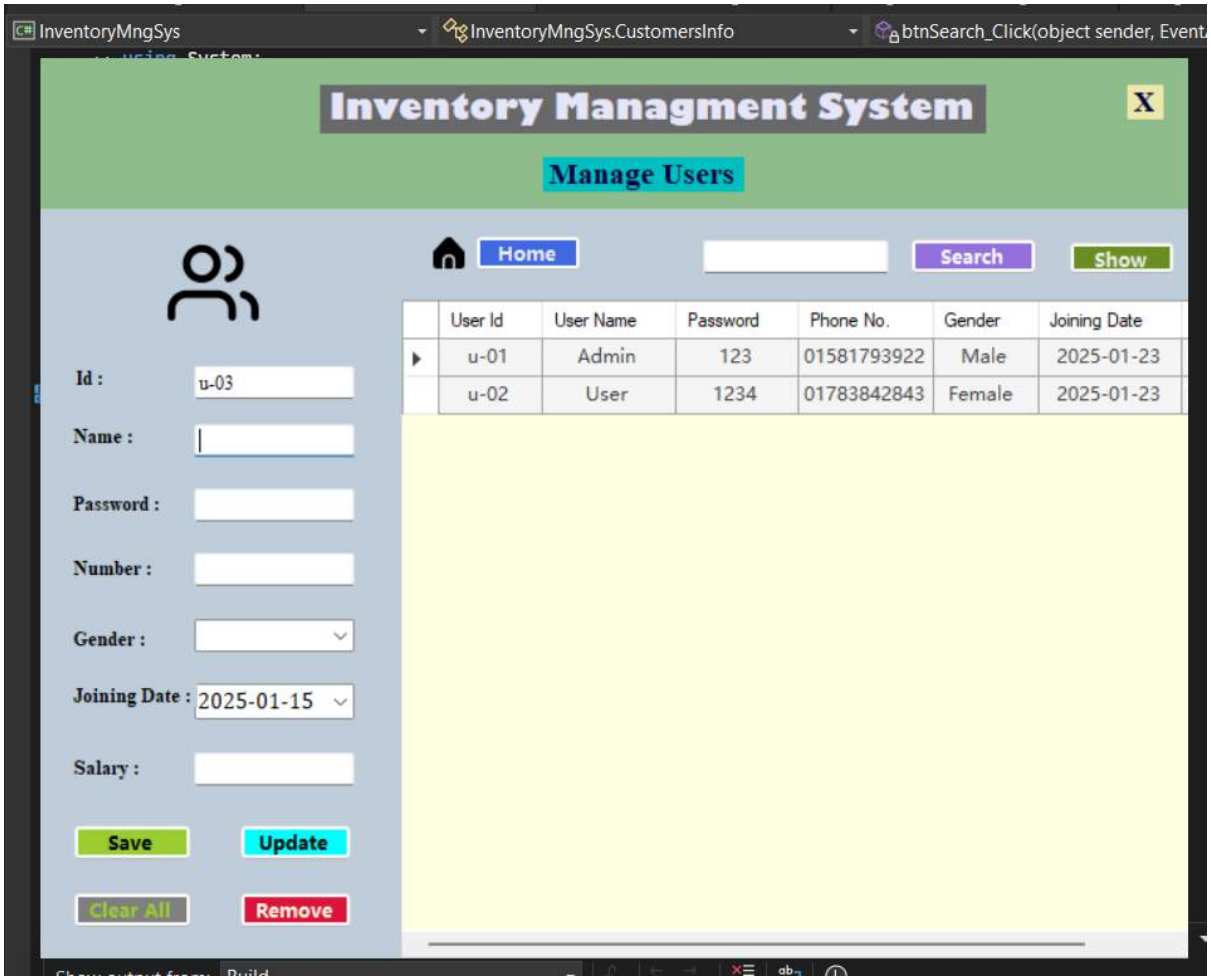
ManageUsers Table
Table Creation
```
CREATE TABLE ManageUsers (
    User_Id INT PRIMARY KEY,
    User_Name VARCHAR(255),
    User_Password VARCHAR(255),
    User_Number VARCHAR(15),
    Gender VARCHAR(10),
    Joining_Date DATE,
    Salary DECIMAL(10, 2)
););
```
Insertion
```
INSERT INTO ManageUsers (User_Id, User_Name, User_Password,
User_Number, Gender, Joining_Date, Salary)
VALUES (1, 'admin', 'admin123', '1231231234', 'Male', '2025-01-01',
50000.00);
```
Update
```
UPDATE ManageUsers
SET User_Password = 'newpassword123', Salary = 55000.00
WHERE User_Id = 1;
```
Delete
```
DELETE FROM ManageUsers
WHERE User_Id = 1;
```

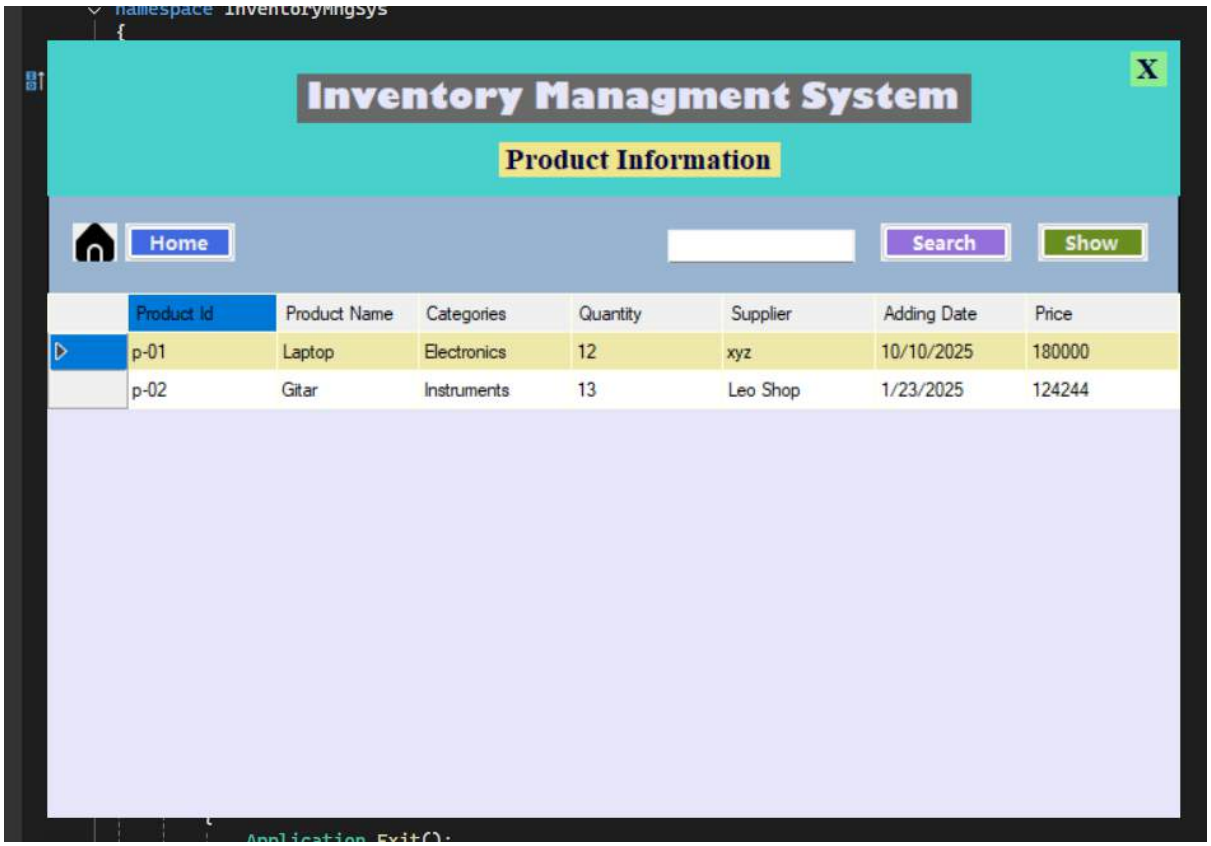
Search
SELECT * FROM ManageUsers
WHERE User_Name = 'admin';

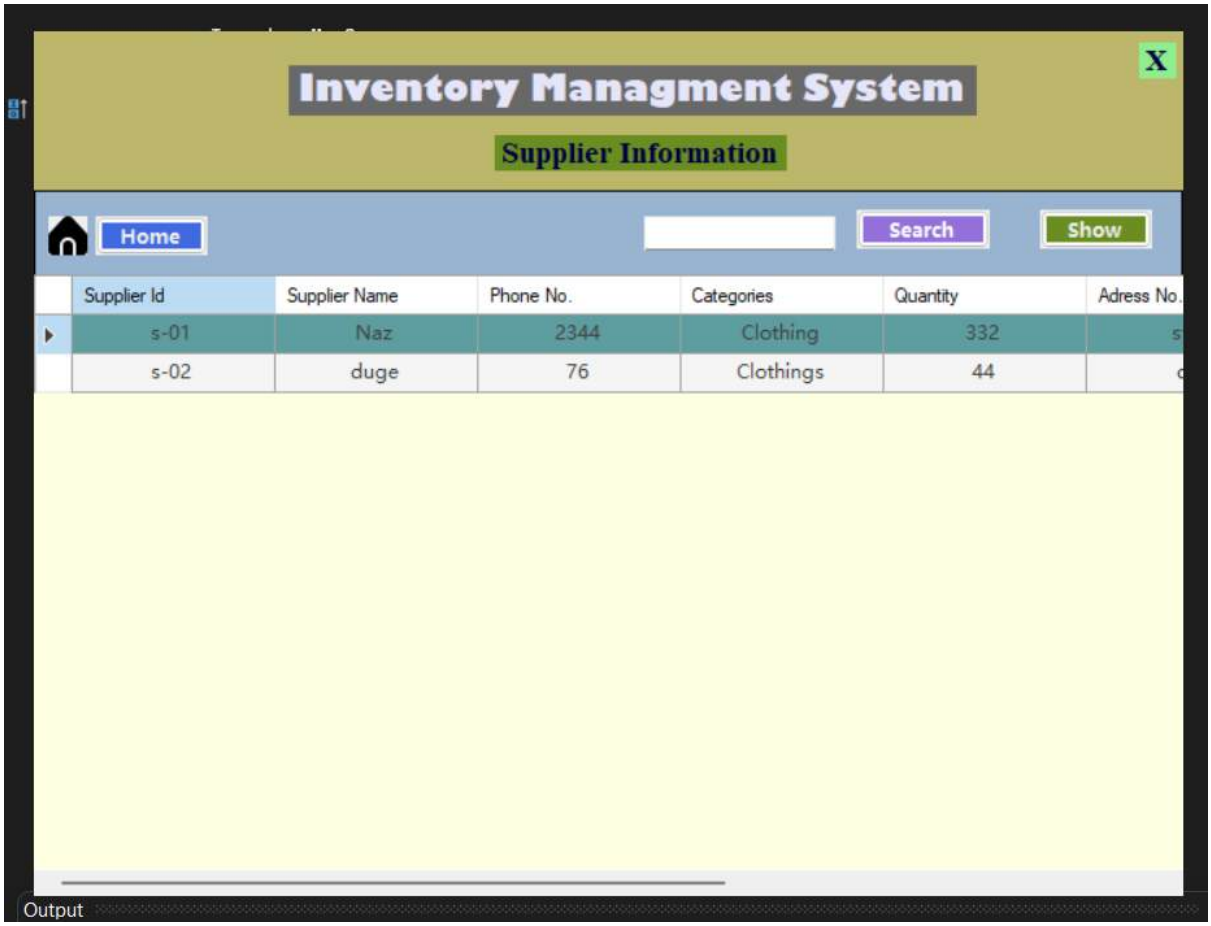
Screenshots:

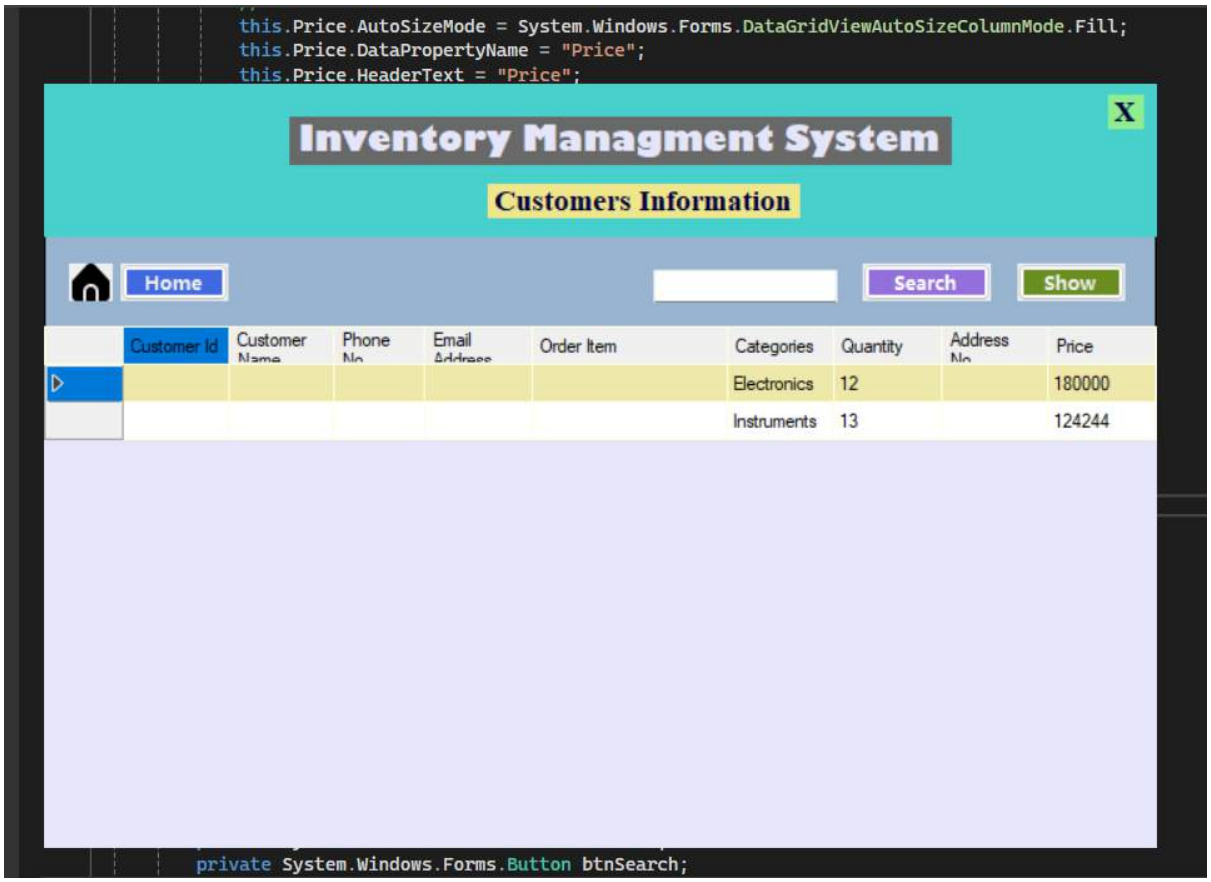




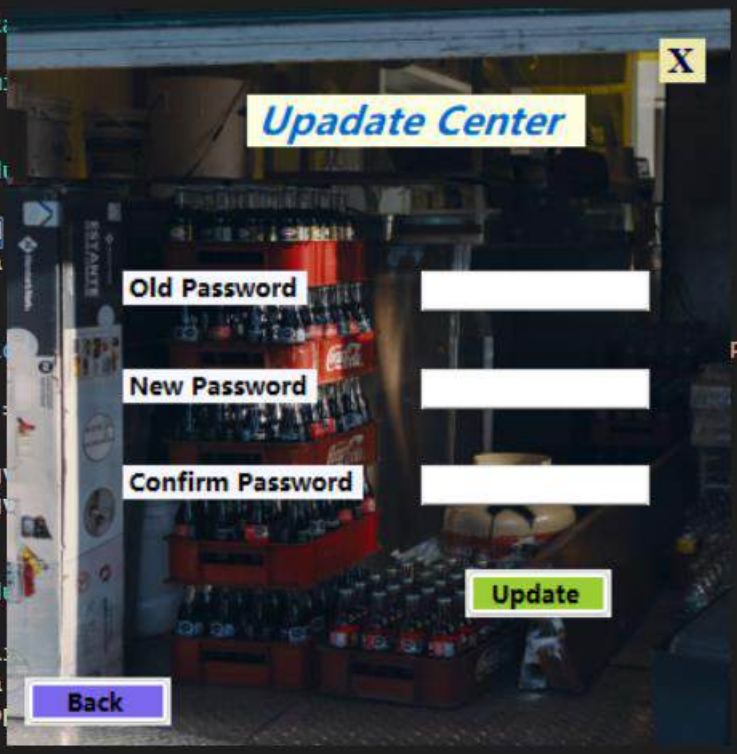

InventoryMngSys
InventoryMngSys.CustomersInfo
btnSearch_Click(object sender, Event
Inventory Managment System
X
Manage Users
Home
Search
Show
User Id
User Name
Password
Phone No.
Gender
Joining Date
u-01
Admin
123
01581793922
Male
2025-01-23
u-02
User
1234
01783842843
Female
2025-01-23
Id :
u-03
Name :
Password :
Number :
Gender :
Joining Date :
2025-01-15
Salary :
Save
Update
Clear All
Remove


Inventory Managment System
X
Product Information
Home
Search
Show
Product Id
Product Name
Categories
Quantity
Supplier
Adding Date
Price
p-01
Laptop
Electronics
12
xyz
10/10/2025
180000
p-02
Gitar
Instruments
13
Leo Shop
1/23/2025
124244

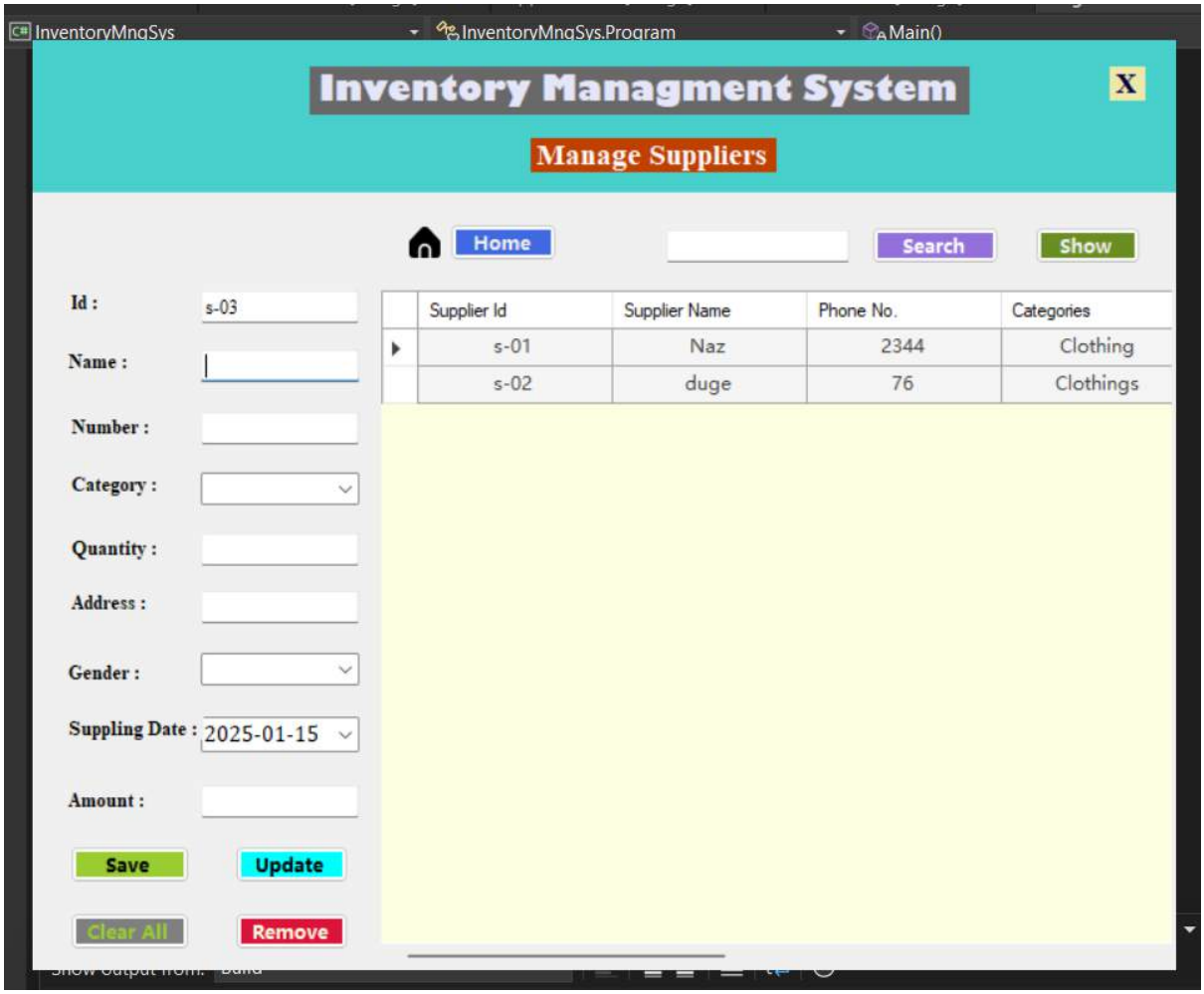

Inventory Managment System
Supplier Information
Home
Search
Show
Supplier Id
Supplier Name
Phone No.
Categories
Quantity
Adress No.
s-01
Naz
2344
Clothing
332
s-02
duge
76
Clothings
44
Output

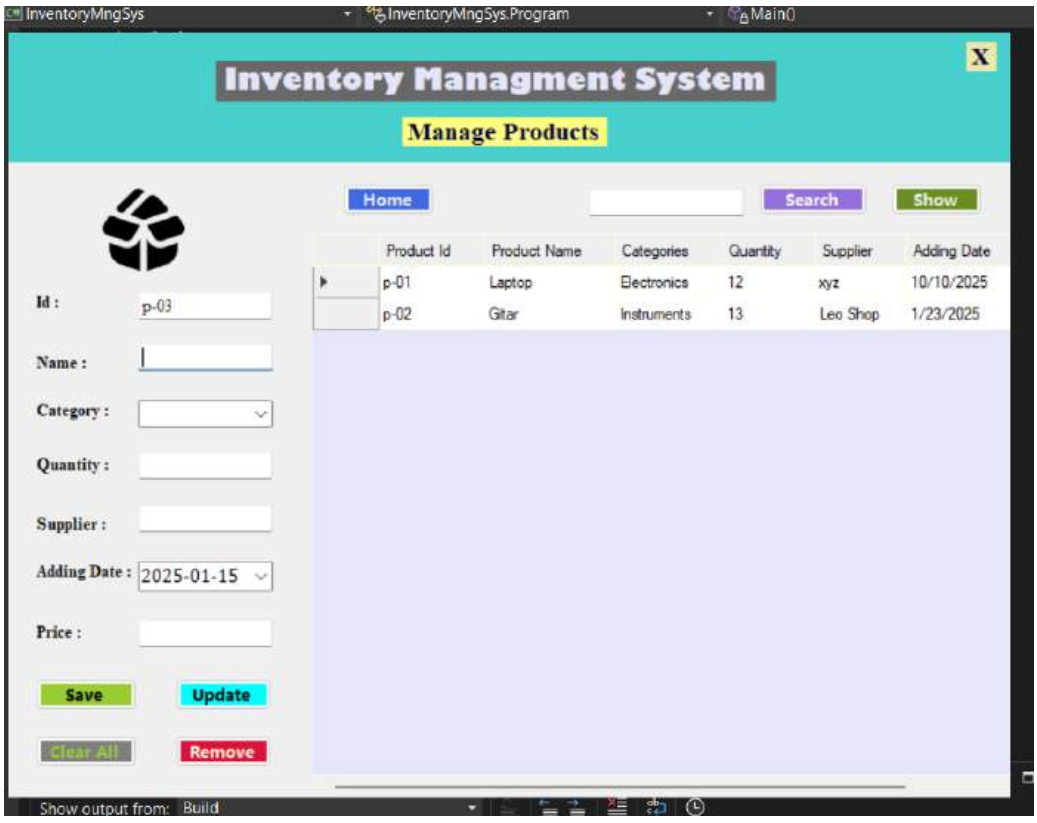

this.Price.AutoSizeMode = System.Windows.Forms.DataGridViewAutoSizeColumnMode.Fill;
this.Price.DataPropertyName = "Price";
this.Price.HeaderText = "Price";
Inventory Managment System
Customers Information
Home
Search
Show
Customer Id
Customer Name
Phone No
Email Address
Order Item
Categories
Quantity
Address No
Price
Electronics
12
180000
Instruments
13
124244
private System.Windows.Forms.Button btnSearch;

```
6 references
public partial class ProductInfo : Form
{
    3 references
    private Dat
    2 references
    private Adm

    0 references
    public Produ
    {
        this.Ad
        this.Da
    }
    1 reference
    private voi
    {
        var ds =

        this.dgv
        this.dgv
    }
    2 references
    public Prod
    {
        Initial
        this.Da
        this.Po
    }

    1 reference
```




```
namespace InventoryMngSys
{

}
```


InventoryMngSys
InventoryMngSys.Program
Main()
Inventory Managment System
X
Manage Suppliers
Home
Search
Show
Id :
s-03
Name :
Number :
Category :
Quantity :
Address :
Gender :
Suppling Date :
2025-01-15
Amount :
Save
Update
Clear All
Remove
Supplier Id
Supplier Name
Phone No.
Categories
s-01
Naz
2344
Clothing
s-02
duge
76
Clothings


InventoryMngSys
InventoryMngSys.Program
Main()
Inventory Managment System
X
Manage Products
Home
Search
Show
Id :
p-03
Name :
Category :
Quantity :
Supplier :
Adding Date :
2025-01-15
Price :
Save
Update
Clear All
Remove
Product Id
Product Name
Categories
Quantity
Supplier
Adding Date
p-01
Laptop
Electronics
12
xyz
10/10/2025
p-02
Gitar
Instruments
13
Leo Shop
1/23/2025
Show output from: Build