DATABASE SYSTEM

CS-241

BSCS-20-IV

SPRING 2022

PROJECT

(BOOKSHOP MANAGAMENT SYSTEM)

SUBMITTED BY

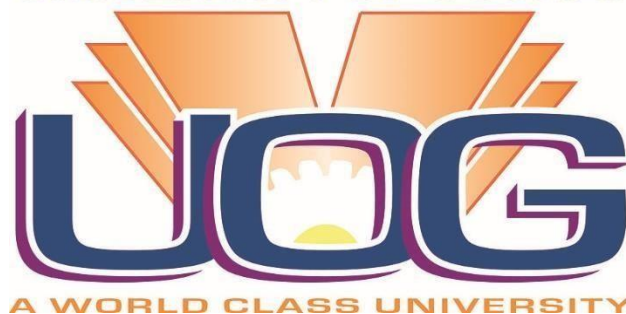| ROLL NO.	STUDENT NAME	SECTION
20021519-109	MUAHAMMAD FAHAD	A

SUBMISSION DATE

AUG 31, 2022

INSTRUCTOR

DR. NAVEED ANWER BUTT

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF GUJRAT

Table of Contents

1   PROJECT PROPOSAL

1.1   Project Description

Main object of this project is to provide easiness for bookshops to manage their records. In the Bookshop store, there are many books that are managed but sometime you cannot find the right book or sometime it is difficult for employees to remember the books name or location or maybe the price of books. So, keeping track of all the books in the store in the location is very important.

The Bookshop management system application will helps in managing the books in a well-organized way. The database should be efficient enough which will be capable of holding all the details or the information related to the bookshop.

The interface of the project will be in Windows Application Forms developed using C# language in Visual Studio 2019.

1.2   Project Objectives

The objectives of bookshop management system are related to managing (adding, deleting, changing, updating and searching) the records of Supplier, available Stock related to the books in the shop, Employees details like salaries and the orders which they are handling, Customer details, Generating bills for customer orders and managing their payments also they records of orders which have been returned by customers.

1.3   Entities of Project

1. Suppliers Table
2. Stock Table
3. Employees Table
4. Customers Table
5. Orders Table
6. Bills Table
7. Payments Table
8. Returned Orders Table

2 Project ERD
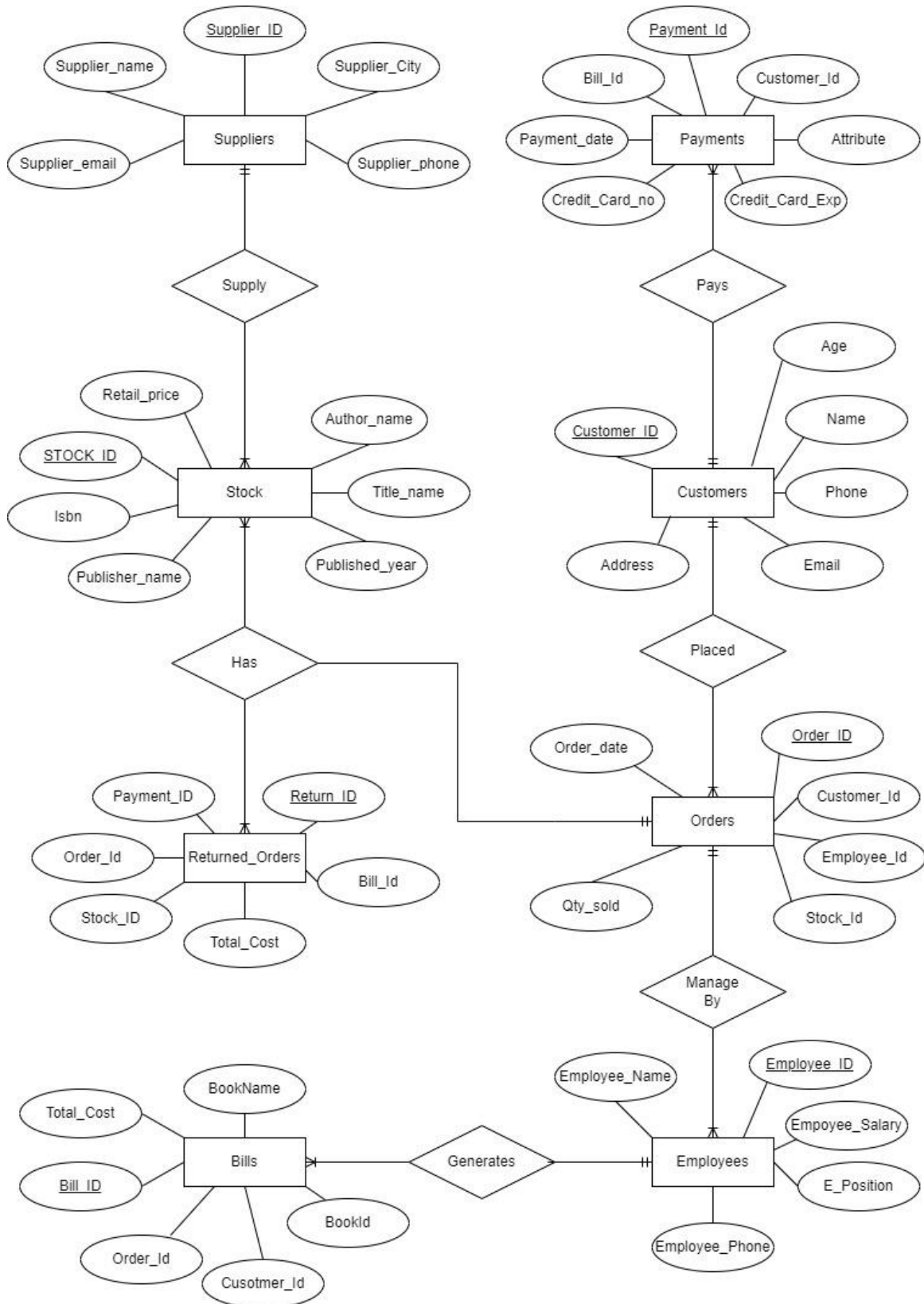
3   Database Code and Queries

```sql
--BOOKSHOP MANAGEMENT SYSTEM

CREATE TABLE SuppliersTb(
       Supplier_ID INT NOT NULL,
       Supplier_name VARCHAR(55) NOT NULL,
    Supplier_City VARCHAR(55),
       Supplier_email VARCHAR(55),
       Supplier_phone VARCHAR(55) NOT NULL,
    PRIMARY KEY (Supplier_ID)
    );

insert into SuppliersTb values(1,'SN1','City1','SN1@gmail.com','0321-6020450')
insert into SuppliersTb values(2,'SN2','City2','SN2@gmail.com','0322-6020450')
insert into SuppliersTb values(3,'SN3','City3','SN3@gmail.com','0323-6020450')
insert into SuppliersTb values(4,'SN4','City4','SN4@gmail.com','0324-6020450')
insert into SuppliersTb values(5,'SN5','City5','SN5@gmail.com','0325-6020450')
insert into SuppliersTb values(6,'SN6','City6','SN6@gmail.com','0326-6020450')
insert into SuppliersTb values(7,'SN7','City7','SN7@gmail.com','0327-6020450')
insert into SuppliersTb values(8,'SN8','City8','SN8@gmail.com','0328-6020450')
insert into SuppliersTb values(9,'SN9','City9','SN9@gmail.com','0329-6020450')
insert into SuppliersTb values(10,'SN0','City10','SN10@gmail.com','0330-6020450')
select * from SuppliersTb

drop table SuppliersTb

create table StockTb(
       StockID INT primary key,
       SupplierID INT foreign key references SuppliersTb(Supplier_ID),
       BookID varchar(50) unique not null,
       AuthorName varchar(50) not null,
       BookName varchar(50) not null,
       PublisherName varchar(50) not null,
       PublishedYear varchar(50) not null,
       Price int not null
);

drop table StockTb

INSERT INTO StockTb VALUES(1,1,'BOOK1','J.R.R Tolken','The Hobbit','Allen &
Unwin','1937', 1100);
INSERT INTO StockTb VALUES(2,1,'BOOK2','Tanith Lee','The Castle of
Dark','Macmilla','1930', 1200);
INSERT INTO StockTb VALUES(3,2,'BOOK3','Tanith Lee','The Winter
Players','Macmilla','1977', 1300);
INSERT INTO StockTb VALUES(4,2,'BOOK4','Anne Rice','Tale of the Thief','Penguin','2016',
1400);
INSERT INTO StockTb VALUES(5,3,'BOOK5','J.R.R Tolken','The Lord of the Rings : Fellowship
of the ring','Allen & Unwin','1937', 1500);
INSERT INTO StockTb VALUES(6,3,'BOOK6','Mark Stevenson','Prince and the Pauper','American
Pushlishing Co','2011', 1600);
INSERT INTO StockTb VALUES(7,4,'BOOK7','Ribbly Scott','Alien','Morpheus','1996', 1700);
INSERT INTO StockTb VALUES(8,4,'BOOK8','James Clavell','Gone Girl','Paramount','2015',
1800);
INSERT INTO StockTb VALUES(9,5,'BOOK9','Megan Miranda','All the Missing Girls','H &
R','2016', 1900);
```

```sql
INSERT INTO StockTb VALUES(10,6,'BOOK10','Sarah Mass','Empire of Storms','Pearson
Plc','2006', 2000);

select * from StockTb

select AuthorName, COUNT(PublishedYear)
from StockTb
group by AuthorName

CREATE TABLE Employees(
      Employee_ID INT PRIMARY KEY,
      Employee_Name VARCHAR(50),
      Employee_ContactNumber VARCHAR(50),
      Employee_Position VARCHAR(50),
      Employee_Salary MONEY CHECK(Employee_Salary>20000) NOT NULL,
      );

insert into Employees values (1,'Rizwan','012-4289087','Manager',30000);
insert into Employees values (2,'Aasd','013-4289087','Accountant',35000);
insert into Employees values (3,'Umar','014-4289087','Clerak',22000);
drop table Employees

CREATE TABLE Customer(
      Customer_ID INT NOT NULL,
      Name VARCHAR(30) NOT NULL,
      CustomerAddress VARCHAR(max) NOT NULL,
      Phone VARCHAR(50) NOT NULL,
      Email VARCHAR(50) NOT NULL,
      Age INT NOT NULL CHECK(Age>10),
      PRIMARY KEY (Customer_ID)
);

      INSERT INTO Customer VALUES(1,'Nouman','Sialkot','0313-
12345433','Customer1@gmail.com',18)
      INSERT INTO Customer VALUES(2,'Asad','Sialkot','0314-
12145436','Customer2@gmail.com',17)
      INSERT INTO Customer VALUES(3,'Sarfaraz','Sialkot','0315-
12145436','Customer3@gmail.com',22)
      INSERT INTO Customer VALUES(4,'Ali','Sialkot','0316-
12145436','Customer4@gmail.com',23)
      INSERT INTO Customer VALUES(5,'Usman','Sialkot','0317-
12145436','Customer5@gmail.com',24)
      INSERT INTO Customer VALUES(6,'Salman','Sialkot','0318-
12145436','Customer6@gmail.com',25)
      INSERT INTO Customer VALUES(7,'Hassan','Sialkot','0319-
12145436','Customer7@gmail.com',26)
      INSERT INTO Customer VALUES(8,'Daud','Sialkot','0320-
12145436','Customer8@gmail.com',28)
      INSERT INTO Customer VALUES(9,'Nauman','Sialkot','0321-
12145436','Customer9@gmail.com',30)
      INSERT INTO Customer VALUES(10,'Yaseen','Sialkot','0322-
12145436','Customer10@gmail.com',33)
      drop table Customer

      CREATE TABLE Orders(
      Order_ID INT NOT NULL,
      Customer_ID INT FOREIGN KEY REFERENCES Customer(Customer_ID),
      Customer_Name VARCHAR(50),
```

```sql
    Employee_ID INT FOREIGN KEY REFERENCES Employees(Employee_ID),
    StockID      INT FOREIGN KEY REFERENCES StockTb(StockID),
    Qty_sold INT,
    Order_Date VARCHAR(55),
PRIMARY KEY (Order_ID),
);

    INSERT INTO Orders VALUES(1,1,'Nauman'  ,1,1,1,'2-2-2022')
    INSERT INTO Orders VALUES(2,2,'Asad'    ,1,2,2,'3-2-2022')
    INSERT INTO Orders VALUES(3,1,'Nauman'  ,1,1,1,'4-2-2022')
    INSERT INTO Orders VALUES(4,3,'Sarfaraz',1,3,2,'5-2-2022')
    INSERT INTO Orders VALUES(5,3,'Sarfaraz',1,4,1,'6-2-2022')

    Select * From Orders

    Select Customer_ID, Qty_sold From Orders
    group by Customer_ID, Qty_sold

    drop table Orders

    CREATE TABLE Bill_Generate(
    Bill_ID INT primary key,
    Order_ID INT FOREIGN KEY REFERENCES Orders(Order_ID),
    Customer_ID INT FOREIGN KEY REFERENCES Customer(Customer_ID),
    StockID      INT FOREIGN KEY REFERENCES StockTb(StockID),
    Bill_Date date,
    Total_Cost INT CHECK(Total_Cost>0) NOT NULL,
    );

    INSERT INTO Bill_Generate VALUES(1,1,1,1,'2-2-2022',1100)
    INSERT INTO Bill_Generate VALUES(2,2,2,2,'3-2-2022',1200)
    INSERT INTO Bill_Generate VALUES(3,3,1,1,'4-2-2022',1100)
    INSERT INTO Bill_Generate VALUES(4,4,3,3,'5-2-2022',1300)
    INSERT INTO Bill_Generate VALUES(5,5,3,4,'6-2-2022',1300)


    DROP TABLE Bill_Generate

    CREATE TABLE PAYMENTS(
    Payment_ID INT PRIMARY KEY,
    Bill_ID INT FOREIGN KEY REFERENCES Bill_Generate(Bill_ID),
    Customer_ID INT FOREIGN KEY REFERENCES Customer(Customer_ID),
    Credit_card_numb VARCHAR(55) NOT NULL,
    Credit_card_expiry varchar(55) NOT NULL,
    PaymentPaid INT CHECK(PaymentPaid>0) NOT NULL,
    );

    INSERT INTO PAYMENTS VALUES(1,1,1,'1717-22-233','11-17-2024',1100)
    INSERT INTO PAYMENTS VALUES(2,2,2,'1718-23-233','01-17-2025',1200)
    INSERT INTO PAYMENTS VALUES(3,3,1,'1719-24-233','03-17-2023',1100)
    INSERT INTO PAYMENTS VALUES(4,4,3,'1720-25-233','05-17-2030',1300)
    INSERT INTO PAYMENTS VALUES(5,5,3,'1721-26-233','07-17-2045',1300)

    drop table PAYMENTS

    CREATE TABLE Returns(
    Return_ID INT PRIMARY KEY,
    Bill_ID INT FOREIGN KEY REFERENCES Bill_Generate(Bill_ID),
```

```
    Payment_ID INT FOREIGN KEY REFERENCES PAYMENTS(Payment_ID),
    Order_ID INT FOREIGN KEY REFERENCES Orders(Order_ID),
PayReturned int,
)

INSERT INTO Returns VALUES(1,1,1,1,1100)
    INSERT INTO Returns VALUES(2,5,5,5,1300)

    select * from Returns
    drop table Returns

    SELECT * FROM Orders
    WHERE Order_Date BETWEEN '2-2-2022'AND'5-2-2022';
    Select * From StockTb
    SELECT * from StockTb where PublishedYear = 2016
    --total sales group by customer id from payments
    select Sum(Total_Cost) from Payments
    select Sum(Total_Cost),Customer_ID
    from Payments
    Group by Customer_ID
    having Max(Total_Cost) > 2000
    select Sum(Total_Cost) As 'Total Sale' From PAYMENTS;
```

4   Application Interface and Code

4.1   Application Interface

Supplier Records

Supplier ID: [ ]   Email Address: [ ]

Name: [ ]   Phone: [ ]

Address: [ ]

[Insert]   [Update]   [Delete]   [Reset]

	Supplier_ID	Supplier_name	Supplier_City	Supplier_email	Supplier_phone
▶	1	SN1	City1	SN1@gmail.com	0321-6020450
	2	SN2	City2	SN2@gmail.com	0322-6020450
	3	SN3	City3	SN3@gmail.com	0323-6020450
	4	SN4	City4	SN4@gmail.com	0324-6020450
	5	SN5	City5	SN5@gmail.com	0325-6020450

Stock Records

Stock ID: [1]   Book Name: [The Hobbit]

Supplier ID: [1]   Publisher Name: [Allen & Unwin]

Book ID: [BOOK1]   Publisher Year: [1937]

Aurthor Name: [J.R.R Tolken]   Book Price: [1100]

[Insert]   [Update]   [Delete]   [Reset]

	StockID	SupplierID	BookID	AuthorName	BookName	PublisherName	PublishedYear	Price
▶	1	1	BOOK1	J.R.R Tolken	The Hobbit	Allen & Unwin	1937	1100
	2	1	BOOK2	Tanith Lee	The Castle of Dark	Macmilla	1930	1200
	3	2	BOOK3	Tanith Lee	The Winter Players	Macmilla	1977	1300
	4	2	BOOK4	Anne Rice	Tale of the Thief	Penguin	2016	1400
	5	3	BOOK5	J.R.R Tolken	The Lord of the ...	Allen & Unwin	1937	1500

Customers Record

Customer ID: [ ]   Phone: [ ]

Name: [ ]   Email: [ ]

Address: [ ]   Age: [ ]

[Insert]   [Update]   [Delete]   [Reset]

Customer_ID	Name	CustomerAddress	Phone	Email	Age
1	Nouman	Sialkot	0313-12345433	Customer1@gmai...	18
2	Asad	Sialkot	0314-12145436	Customer2@gmai...	17
3	Sarfaraz	Sialkot	0315-12145436	Customer3@gmai...	22
4	Ali Amjad	Sialkot	0316-12145436	Customer4@gmai...	23
5	Usman	Sialkot	0317-12145436	Customer5@gmai...	24

Orders

Order ID: [ ]   Stock ID: [ ]

Customer ID: [ ]   Quantity Sold: [ ]

Customer Name: [ ]   Order Date: [ ]

Employee ID: [ ]

[Insert]   [Update]   [Delete]   [Reset]

Order_ID	Customer_ID	Customer_Name	Employee_ID	StockID	Qty_sold	Order_Date
1	1	Nauman	1	1	1	2-2-2022
2	2	Asad	1	2	2	3-2-2022
3	1	Nauman	1	1	1	4-2-2022
4	3	Sarfaraz	1	3	2	5-2-2022
5	3	Sarfaraz	2	4	1	6-2-2022

Generate Bills

Bill ID: 6

Stock ID: 4

Order ID: 5

Bill Generate Date: 6/2/2022 12:00:0

Customer ID: 3

Total Bill: 1300

[Insert] [Update] [Delete] [Reset]

Bill_ID	Order_ID	Customer_ID	StockID	Bill_Date	Total_Cost
2					1200
3					1100
4					1300
5					1300
*					

Saved

New Bill Generated Sucessfully Saved In The Database

[OK]

Payments Details

Payment ID: 5

Credit Card No: 1721-26-233

Bill ID: 5

Card Expiry Date: 07-17-2045

Customer ID: 3

Payment Paid: 1300

[Insert] [Update] [Delete] [Reset]

Payment_ID	Bill_ID	Customer_ID	Credit_card_numb	Credit_card_expiry	Payment Paid
1	1	1	1717-22-233	11-17-2024	1100
2	2	2	1718-23-233	01-17-2025	1200
3	3	1	1719-24-233	03-17-2023	1100
4	4	3	1720-25-233	05-17-2030	1300
5	5	3	1721-26-233	07-17-2045	1300

4.2   Application Code

4.2.1   Main Menu Form Code

```
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class Menu : Form
    {
        public Menu()
        {
            InitializeComponent();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            Supplier s1 = new Supplier();
            s1.Show();
        }

        private void Button2_Click(object sender, EventArgs e)
```

```csharp
        {
            Stock f1 = new Stock();
            f1.Show();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            Employees e1 = new Employees();
            e1.Show();
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            Customers_Details c1 = new Customers_Details();
            c1.Show();
        }

        private void Button5_Click(object sender, EventArgs e)
        {
            Orders o1 = new Orders();
            o1.Show();
        }

        private void Button6_Click(object sender, EventArgs e)
        {
            Bills b1 = new Bills();
            b1.Show();
        }

        private void Menu_Load(object sender, EventArgs e)
        {

        }

        private void Button7_Click(object sender, EventArgs e)
        {
            Payments p1 = new Payments();
            p1.Show();
        }

        private void Button8_Click(object sender, EventArgs e)
        {
            ReturnedOrders r1 = new ReturnedOrders();
            r1.Show();
        }
    }
}
```

4.2.2   Supplier Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
```

```csharp
    public partial class Supplier : Form
    {
        public Supplier()
        {
            InitializeComponent();
        }
        public int SupplierID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Supplier_Load(object sender, EventArgs e)
        {
            GetSupplierRecords();
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            SupplierID = 0;
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox6.Clear();
            textBox8.Clear();
            textBox2.Focus();
        }

        private void GetSupplierRecords()
        {

            SqlCommand cmd = new SqlCommand("Select * from SuppliersTb", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            SuppliersRecordsDataGridView.DataSource = dt;
        }
        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void Button1_Click(object sender, EventArgs e)
        {
            if(IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO SuppliersTb VALUES
(@SupplierID, @Supplier_name, @Supplier_City, @Supplier_email, @Supplier_phone)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@SupplierID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Supplier_name", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Supplier_City", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Supplier_email", textBox6.Text);
                    cmd.Parameters.AddWithValue("@Supplier_phone", textBox8.Text);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
```

```csharp
                MessageBox.Show("New Supplier Record Sucessfully Saved In The
Database", "Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);
                GetSupplierRecords();
            }
            catch(Exception Ex)
            {
                MessageBox.Show(Ex.Message);
            }


        }
    }
    private bool IsValid()
    {
        if (textBox2.Text == string.Empty)
        {
            MessageBox.Show("Supplier ID Is Required", "Failed",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
        return true;
    }
    private void SuppliersRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        SupplierID =
Convert.ToInt32(SuppliersRecordsDataGridView.SelectedRows[0].Cells[0].Value);
        textBox2.Text =
SuppliersRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
        textBox3.Text =
SuppliersRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
        textBox4.Text =
SuppliersRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
        textBox6.Text =
SuppliersRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
        textBox8.Text =
SuppliersRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
    }

    private void Button3_Click(object sender, EventArgs e)
    {
        if (SupplierID > 0)
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM SuppliersTb WHERE
Supplier_ID = @ID", con);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@ID", this.SupplierID);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Supplier Record Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            GetSupplierRecords();
            ResetFormControls();
        }
        else
        {
            MessageBox.Show("Please Select A Supplier Records To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```csharp
                }
            }
            private void Button2_Click(object sender, EventArgs e)
            {
                if (SupplierID > 0)
                {
                    try
                    {
                        SqlCommand cmd = new SqlCommand("UPDATE SuppliersTb SET Supplier_ID =
@SupplierID, Supplier_name = @SupplierName , Supplier_city = @SupplierCity,
Supplier_email = @SupplierEmail, Supplier_phone = @SupplierPhone WHERE Supplier_ID =
@ID", con);
                        cmd.CommandType = CommandType.Text;
                        cmd.Parameters.AddWithValue("@SupplierID", textBox2.Text);
                        cmd.Parameters.AddWithValue("@SupplierName", textBox3.Text);
                        cmd.Parameters.AddWithValue("@SupplierCity", textBox4.Text);
                        cmd.Parameters.AddWithValue("@SupplierEmail", textBox6.Text);
                        cmd.Parameters.AddWithValue("@SupplierPhone", textBox8.Text);
                        cmd.Parameters.AddWithValue("@ID", this.SupplierID);
                        con.Open();
                        cmd.ExecuteNonQuery();
                        con.Close();
                        MessageBox.Show("Stock Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        GetSupplierRecords();
                        ResetFormControls();
                    }
                    catch (Exception Ex)
                    {
                        MessageBox.Show(Ex.Message);
                    }
                }
                else
                {
                    MessageBox.Show("Please Select A Stock To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
}
```

4.2.3 Stock Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class Stock : Form
    {
        public Stock()
        {
            InitializeComponent();
        }
        public int StockID;
```

```csharp
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Button1_Click(object sender, EventArgs e)
        {
            if(IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO StockTb VALUES
(@StockID, @SupplierID, @BookID, @AuthorName, @BookName, @PublisherName, @PublishedYear,
@Price)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@StockID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@SupplierID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@BookID", textBox3.Text);
                    cmd.Parameters.AddWithValue("@AuthorName", textBox4.Text);
                    cmd.Parameters.AddWithValue("@BookName", textBox5.Text);
                    cmd.Parameters.AddWithValue("@PublisherName", textBox6.Text);
                    cmd.Parameters.AddWithValue("@PublishedYear", textBox7.Text);
                    cmd.Parameters.AddWithValue("@Price", textBox8.Text);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("New Stock Sucessfully Saved In The Database",
"Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetStockRecords();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }

            }
        }
        private bool IsValid()
        {
            if(textBox2.Text == string.Empty)
            {
                MessageBox.Show("Stock ID Is Required", "Failed", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return false;
            }
            return true;
        }
        private void Button2_Click(object sender, EventArgs e)
        {
            if(StockID>0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE StockTb SET StockID =
@StockId, SupplierID = @SupplierID , BookID = @BookID, AuthorName = @AuthorName,
@BookName = @BookName, PublisherName = @PublisherName, PublishedYear = @PublishedYear,
Price = @Price WHERE StockID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@StockID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@SupplierID", textBox2.Text);
```

```csharp
                cmd.Parameters.AddWithValue("@BookID", textBox3.Text);
                cmd.Parameters.AddWithValue("@AuthorName", textBox4.Text);
                cmd.Parameters.AddWithValue("@BookName", textBox5.Text);
                cmd.Parameters.AddWithValue("@PublisherName", textBox6.Text);
                cmd.Parameters.AddWithValue("@PublishedYear", textBox7.Text);
                cmd.Parameters.AddWithValue("@Price", textBox8.Text);
                cmd.Parameters.AddWithValue("@ID", this.StockID);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Stock Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetStockRecords();
                ResetFormControls();
            }
            catch (Exception Ex)
            {
                MessageBox.Show(Ex.Message);
            }
        }
        else
        {
            MessageBox.Show("Please Select A Stock To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        GetStockRecords();
        ResetFormControls();
    }

    private void GetStockRecords()
    {

        SqlCommand cmd = new SqlCommand("Select * from StockTb", con);
        DataTable dt = new DataTable();
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        dt.Load(sdr);
        con.Close();
        StockRecordsDataGridView.DataSource = dt;
    }

    private void Button4_Click(object sender, EventArgs e)
    {
        ResetFormControls();

    }
    private void ResetFormControls()
    {
        StockID = 0;
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
```

```csharp
            textBox5.Clear();
            textBox6.Clear();
            textBox7.Clear();
            textBox8.Clear();
            textBox1.Focus();
        }

        private void StockRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            StockID =
Convert.ToInt32(StockRecordsDataGridView.SelectedRows[0].Cells[0].Value);
            textBox1.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
            textBox2.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
            textBox3.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
            textBox4.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
            textBox5.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
            textBox6.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[5].Value.ToString();
            textBox7.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[6].Value.ToString();
            textBox8.Text =
StockRecordsDataGridView.SelectedRows[0].Cells[7].Value.ToString();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            if(StockID>0)
            {
                SqlCommand cmd = new SqlCommand("DELETE FROM StockTb WHERE StockID =
@ID", con);
                cmd.CommandType = CommandType.Text;

                cmd.Parameters.AddWithValue("@ID", this.StockID);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Stock Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetStockRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Stock To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

4.2.4   Employees Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class Employees : Form
    {
        public Employees()
        {
            InitializeComponent();
        }
        public int EmployeeID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Employees_Load(object sender, EventArgs e)
        {
            GetEmployeesRecords();
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            EmployeeID = 0;
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox6.Clear();
            textBox8.Clear();
            textBox2.Focus();
        }
        private void GetEmployeesRecords()
        {
            SqlCommand cmd = new SqlCommand("Select * from Employees", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            EmployeesRecordsDataGridView.DataSource = dt;
        }
        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void Button1_Click(object sender, EventArgs e)
        {
            if(IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO Employees VALUES
(@Employee_ID, @Employee_Name, @Employee_ContactNumber, @Employee_Position,
@Employee_Salary)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Employee_ID", textBox2.Text);
```

```csharp
                cmd.Parameters.AddWithValue("@Employee_Name", textBox3.Text);
                cmd.Parameters.AddWithValue("@Employee_ContactNumber",
textBox4.Text);
                cmd.Parameters.AddWithValue("@Employee_Position", textBox6.Text);
                cmd.Parameters.AddWithValue("@Employee_Salary", textBox8.Text);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("New Employee Record Sucessfully Saved In The
Database", "Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);
                GetEmployeesRecords();
            }
            catch (Exception Ex)
            {
                MessageBox.Show(Ex.Message);
            }
        }
    }
    private bool IsValid()
    {
        if (textBox2.Text == string.Empty)
        {
            MessageBox.Show("Supplier ID Is Required", "Failed",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
        return true;
    }

    private void EmployeesRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        EmployeeID =
Convert.ToInt32(EmployeesRecordsDataGridView.SelectedRows[0].Cells[0].Value);
        textBox2.Text =
EmployeesRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
        textBox3.Text =
EmployeesRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
        textBox4.Text =
EmployeesRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
        textBox6.Text =
EmployeesRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
        textBox8.Text =
EmployeesRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
    }

    private void Button3_Click(object sender, EventArgs e)
    {
        if (EmployeeID > 0)
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM Employees WHERE Employee_ID
= @ID", con);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@ID", this.EmployeeID);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
```

```csharp
                MessageBox.Show("Employee Record Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                GetEmployeesRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Employee Records To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        private void Button2_Click(object sender, EventArgs e)
        {
            if (EmployeeID > 0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE Employees SET Employee_ID =
@Employee_ID, Employee_Name = @EmployeeName , Employee_ContactNumber =
@Employee_ContactNumber, Employee_Position = @Employee_Position, Employee_Salary =
@Employee_Salary WHERE Employee_ID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Employee_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@EmployeeName", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Employee_ContactNumber",
textBox4.Text);
                    cmd.Parameters.AddWithValue("@Employee_Position", textBox6.Text);
                    cmd.Parameters.AddWithValue("@Employee_Salary", textBox8.Text);
                    cmd.Parameters.AddWithValue("@ID", this.EmployeeID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Stock Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetEmployeesRecords();
                    ResetFormControls();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }

            }
            else
            {
                MessageBox.Show("Please Select A Stock To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}


```

4.2.5 Customer Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
```

```csharp
using System.Data;
namespace Bookshop
{
    public partial class Customers_Details : Form
    {
        public Customers_Details()
        {
            InitializeComponent();
        }
        public int CustomerID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Customers_Details_Load(object sender, EventArgs e)
        {
            GetCustomersRecords();
        }
        private void GetCustomersRecords()
        {

            SqlCommand cmd = new SqlCommand("Select * from Customer", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            CustomersRecordsDataGridView.DataSource = dt;
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            CustomerID = 0;
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
            textBox1.Focus();
        }
        private void Button1_Click(object sender, EventArgs e)
        {
            if(IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO Customer VALUES
(@Customer_ID, @Name, @CustomerAddress, @Phone, @Email, @Age)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Name", textBox2.Text);
                    cmd.Parameters.AddWithValue("@CustomerAddress", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Phone", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Email", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Age", textBox6.Text);
```

```csharp
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("New Stock Sucessfully Saved In The Database",
"Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetCustomersRecords();
            }
            catch (Exception Ex)
            {
                MessageBox.Show(Ex.Message);
            }

        }
    }
    private bool IsValid()
    {
        if (textBox1.Text == string.Empty)
        {
            MessageBox.Show("Customer ID Is Required", "Failed",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
        return true;
    }

    private void CustomersRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        CustomerID =
Convert.ToInt32(CustomersRecordsDataGridView.SelectedRows[0].Cells[0].Value);
        textBox1.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
        textBox2.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
        textBox3.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
        textBox4.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
        textBox5.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
        textBox6.Text =
CustomersRecordsDataGridView.SelectedRows[0].Cells[5].Value.ToString();
    }
    private void Button3_Click(object sender, EventArgs e)
    {
        if (CustomerID > 0)
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM Customer WHERE Customer_ID =
@ID", con);
            cmd.CommandType = CommandType.Text;

            cmd.Parameters.AddWithValue("@ID", this.CustomerID);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Customer Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```csharp
                GetCustomersRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Customer To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        private void Button2_Click(object sender, EventArgs e)
        {
            if (CustomerID > 0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE Customer SET Customer_ID =
@Customer_ID, Name = @Name, CustomerAddress = @CustomerAddress, Phone = @Phone, Email =
@Email, Age = @Age WHERE Customer_ID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Name", textBox2.Text);
                    cmd.Parameters.AddWithValue("@CustomerAddress", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Phone", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Email", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Age", textBox6.Text);
                    cmd.Parameters.AddWithValue("@ID", this.CustomerID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Stock Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetCustomersRecords();
                    ResetFormControls();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
            else
            {
                MessageBox.Show("Please Select A Customer To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

4.2.6   Orders Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
```

```csharp
{
    public partial class Orders : Form
    {
        public Orders()
        {
            InitializeComponent();
        }
        public int OrderID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Orders_Load(object sender, EventArgs e)
        {
            GetOrdersRecords();
        }
        private void GetOrdersRecords()
        {
            SqlCommand cmd = new SqlCommand("Select * from Orders", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            OrdersRecordsDataGridView.DataSource = dt;
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            OrderID = 0;
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
            textBox7.Clear();
            textBox1.Focus();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            if(IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO Orders VALUES
(@Order_ID, @Customer_ID, @Customer_Name, @Employee_ID, @StockID, @Qty_sold,
@Order_Date)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Order_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_Name", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Employee_ID", textBox4.Text);
                    cmd.Parameters.AddWithValue("@StockID", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Qty_sold", textBox6.Text);
```

```csharp
                cmd.Parameters.AddWithValue("@Order_Date", textBox7.Text);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("New Stock Sucessfully Saved In The Database",
"Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetOrdersRecords();
            }
            catch(Exception Ex)
            {
                MessageBox.Show(Ex.Message);
            }
        }
    }
    private bool IsValid()
    {
        if (textBox1.Text == string.Empty)
        {
            MessageBox.Show("Order ID Is Required", "Failed", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            return false;
        }
        return true;
    }

    private void OrdersRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        OrderID =
Convert.ToInt32(OrdersRecordsDataGridView.SelectedRows[0].Cells[0].Value);
        textBox1.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
        textBox2.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
        textBox3.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
        textBox4.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
        textBox5.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
        textBox6.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[5].Value.ToString();
        textBox7.Text =
OrdersRecordsDataGridView.SelectedRows[0].Cells[6].Value.ToString();
    }

    private void Button3_Click(object sender, EventArgs e)
    {
        if (OrderID > 0)
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM Orders WHERE Order_ID =
@ID", con);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@ID", this.OrderID);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
```

```csharp
                MessageBox.Show("Stock Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetOrdersRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Order To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            if (OrderID > 0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE Orders SET Order_ID =
@Order_ID, Customer_ID = @Customer_ID , Customer_Name = @Customer_Name, Employee_ID =
@Employee_ID, StockID = @StockID, Qty_sold = @Qty_sold, Order_Date = @Order_Date WHERE
Order_ID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Order_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_Name", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Employee_ID", textBox4.Text);
                    cmd.Parameters.AddWithValue("@StockID", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Qty_sold", textBox6.Text);
                    cmd.Parameters.AddWithValue("@Order_Date", textBox7.Text);
                    cmd.Parameters.AddWithValue("@ID", this.OrderID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Order Record Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    GetOrdersRecords();
                    ResetFormControls();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
            else
            {
                MessageBox.Show("Please Select A Order To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

4.2.7   Generate Bills Form Code

```csharp
using System;
using System.Windows.Forms;
```

```csharp
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class Bills : Form
    {
        public Bills()
        {
            InitializeComponent();
        }
        public int BillID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Bills_Load(object sender, EventArgs e)
        {
            GetBillsRecords();
        }
        private void GetBillsRecords()
        {

            SqlCommand cmd = new SqlCommand("Select * from Bill_Generate", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            BillsRecordsDataGridView.DataSource = dt;
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            BillID = 0;
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
            textBox1.Focus();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            if (IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO Bill_Generate VALUES
(@Bill_ID , @Order_ID, @Customer_ID, @StockID, @Bill_Date , @Total_Cost)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Bill_ID ", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Order_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox3.Text);
```

28

```csharp
                    cmd.Parameters.AddWithValue("@StockID", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Bill_Date ", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Total_Cost", textBox6.Text);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("New Bill Generated Sucessfully Saved In The
Database", "Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetBillsRecords();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
        }
        private bool IsValid()
        {
            if (textBox1.Text == string.Empty)
            {
                MessageBox.Show("Bill ID Is Required", "Failed", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return false;
            }
            return true;
        }

        private void BillsRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            BillID =
Convert.ToInt32(BillsRecordsDataGridView.SelectedRows[0].Cells[0].Value);
            textBox1.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
            textBox2.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
            textBox3.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
            textBox4.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
            textBox5.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
            textBox6.Text =
BillsRecordsDataGridView.SelectedRows[0].Cells[5].Value.ToString();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            if (BillID > 0)
            {
                SqlCommand cmd = new SqlCommand("DELETE FROM Bill_Generate WHERE Bill_ID
= @ID", con);
                cmd.CommandType = CommandType.Text;

                cmd.Parameters.AddWithValue("@ID", this.BillID);
                con.Open();
                cmd.ExecuteNonQuery();
```

```
                con.Close();
                MessageBox.Show("Bill Record Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                GetBillsRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Bill To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            if (BillID > 0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE Bill_Generate SET Bill_ID  =
@Bill_ID,  Order_ID = @Order_ID, Customer_ID = @Customer_ID, StockID= @StockID, Bill_Date
= @Bill_Date, Total_Cost = @Total_Cost WHERE Bill_ID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Bill_ID ", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Order_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox3.Text);
                    cmd.Parameters.AddWithValue("@StockID", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Bill_Date ", textBox5.Text);
                    cmd.Parameters.AddWithValue("@Total_Cost", textBox6.Text);
                    cmd.Parameters.AddWithValue("@ID", this.BillID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Bill Record Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetBillsRecords();
                    ResetFormControls();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
            else
            {
                MessageBox.Show("Please Select A Bill To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

4.2.8   Payments Form Code

```
using System;
using System.Windows.Forms;
```

```csharp
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class Payments : Form
    {
        public Payments()
        {
            InitializeComponent();
        }
        public int PaymentID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void Payments_Load(object sender, EventArgs e)
        {
            GetPaymentRecords();
        }
        private void GetPaymentRecords()
        {

            SqlCommand cmd = new SqlCommand("Select * from PAYMENTS", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            PaymentsRecordsDataGridView.DataSource = dt;
        }
        private void ResetFormControls()
        {
            PaymentID = 0;
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
            textBox1.Focus();
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }

        private void PaymentsRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            PaymentID =
Convert.ToInt32(PaymentsRecordsDataGridView.SelectedRows[0].Cells[0].Value);
            textBox1.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
            textBox2.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
            textBox3.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
```

```csharp
            textBox4.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
            textBox5.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
            textBox6.Text =
PaymentsRecordsDataGridView.SelectedRows[0].Cells[5].Value.ToString();
        }
        private void Button1_Click(object sender, EventArgs e)
        {
            if (IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO PAYMENTS VALUES
(@Payment_ID, @Bill_ID, @Customer_ID, @Credit_card_numb, @Credit_card_expiry,
@PaymentPaid)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Payment_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Bill_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Credit_card_numb", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Credit_card_expiry", textBox5.Text);
                    cmd.Parameters.AddWithValue("@PaymentPaid", textBox6.Text);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Payment Made Sucessfully Saved In The Database",
"Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetPaymentRecords();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
        }
        private bool IsValid()
        {
            if (textBox1.Text == string.Empty)
            {
                MessageBox.Show("Payment ID Is Required", "Failed", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return false;
            }
            return true;
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            if (PaymentID > 0)
            {
                SqlCommand cmd = new SqlCommand("DELETE FROM PAYMENTS WHERE Payment_ID =
@ID", con);
                cmd.CommandType = CommandType.Text;

                cmd.Parameters.AddWithValue("@ID", this.PaymentID);
                con.Open();
```

```csharp
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show("Payment Record Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                GetPaymentRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Customer To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            if (PaymentID > 0)
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("UPDATE PAYMENTS SET Payment_ID =
@Payment_ID, Bill_ID = @Bill_ID, Customer_ID = @Customer_ID, Credit_card_numb=
@Credit_card_numb, Credit_card_expiry = @Credit_card_expiry, PaymentPaid = @PaymentPaid
WHERE Payment_ID = @ID", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Payment_ID", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Bill_ID", textBox2.Text);
                    cmd.Parameters.AddWithValue("@Customer_ID", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Credit_card_numb", textBox4.Text);
                    cmd.Parameters.AddWithValue("@Credit_card_expiry", textBox5.Text);
                    cmd.Parameters.AddWithValue("@PaymentPaid", textBox6.Text);
                    cmd.Parameters.AddWithValue("@ID", this.PaymentID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Payment Details Updated Sucessfully", "Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetPaymentRecords();
                    ResetFormControls();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
            else
            {
                MessageBox.Show("Please Select A Customer To Update Its Information.",
"Select?", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

4.2.9   Returned Orders Form Code

```csharp
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;

namespace Bookshop
{
    public partial class ReturnedOrders : Form
    {
        public ReturnedOrders()
        {
            InitializeComponent();
        }
        public int ReturnedID;
        SqlConnection con = new SqlConnection("Data
Source=FAHADMUGHAL\\SQLEXPRESS;Initial Catalog=CSharp;Integrated Security=True");
        private void ReturnedOrders_Load(object sender, EventArgs e)
        {
            GetReturnedRecords();
        }
        private void GetReturnedRecords()
        {

            SqlCommand cmd = new SqlCommand("Select * from Returns", con);
            DataTable dt = new DataTable();
            con.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            dt.Load(sdr);
            con.Close();
            ReturnedRecordsDataGridView.DataSource = dt;
        }

        private void Button4_Click(object sender, EventArgs e)
        {
            ResetFormControls();
        }
        private void ResetFormControls()
        {
            ReturnedID = 0;
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox1.Focus();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            if (IsValid())
            {
                try
                {
                    SqlCommand cmd = new SqlCommand("INSERT INTO Returns VALUES
(@Return_ID , @Bill_ID, @Payment_ID, @Order_ID, @PayReturned)", con);
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.AddWithValue("@Return_ID ", textBox1.Text);
                    cmd.Parameters.AddWithValue("@Bill_ID", textBox2.Text);
```

```csharp
                    cmd.Parameters.AddWithValue("@Payment_ID", textBox3.Text);
                    cmd.Parameters.AddWithValue("@Order_ID", textBox4.Text);
                    cmd.Parameters.AddWithValue("@PayReturned ", textBox5.Text);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Returned Order Sucessfully Saved In The Database",
"Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    GetReturnedRecords();
                }
                catch (Exception Ex)
                {
                    MessageBox.Show(Ex.Message);
                }
            }
        }
        private bool IsValid()
        {
            if (textBox1.Text == string.Empty)
            {
                MessageBox.Show("Return Order ID Is Required", "Failed",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
            return true;
        }

        private void ReturnedRecordsDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            ReturnedID =
Convert.ToInt32(ReturnedRecordsDataGridView.SelectedRows[0].Cells[0].Value);
            textBox1.Text =
ReturnedRecordsDataGridView.SelectedRows[0].Cells[0].Value.ToString();
            textBox2.Text =
ReturnedRecordsDataGridView.SelectedRows[0].Cells[1].Value.ToString();
            textBox3.Text =
ReturnedRecordsDataGridView.SelectedRows[0].Cells[2].Value.ToString();
            textBox4.Text =
ReturnedRecordsDataGridView.SelectedRows[0].Cells[3].Value.ToString();
            textBox5.Text =
ReturnedRecordsDataGridView.SelectedRows[0].Cells[4].Value.ToString();
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            if (ReturnedID > 0)
            {
                SqlCommand cmd = new SqlCommand("DELETE FROM Returns WHERE Return_ID =
@ID", con);
                cmd.CommandType = CommandType.Text;

                cmd.Parameters.AddWithValue("@ID", this.ReturnedID);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
```

```csharp
                MessageBox.Show("Returned Record Deleted Sucessfully", "Deleted",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                GetReturnedRecords();
                ResetFormControls();
            }
            else
            {
                MessageBox.Show("Please Select A Customer To Delete.", "Select?",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```