# CHAPTER 1

**INTRODUCTION**

### Database Management System

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware). As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database. With relational DBMSs (RDBMSs), this API is SQL, a standard programming.

### INTRODUCTION TO MICROSOFT VISUAL STUDIO 2019

Visual Studio is an **Integrated Development Environment (IDE)** developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

**Community:** It is a **free** version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than $1 Million (US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

### INTRODUCTION TO C#

C# is a general purpose modern and object-oriented programming language pronounced as “C#” It was developed by Microsoft led by Anders Hejlsberg and his team within the .Net initiative and was approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is among the languages for Common Language Infrastructure.

### PROBLEM STATEMENT

Coffee database is a database software where user purchases coffee in a shop, coffee’s have many category customers pays money and gets billing information where employee sales Coffee and produce bills to customer with customer details and the employee details who served it along with the coffee details and its price. This motivated us to develop an application to ease the administrative activities for these kinds of business section.

### APPLICATION AND ADVANTAGES

This software can be used to monitor and store the business data of a BlueBay Coffee store.

The main advantage of this application is that it reduces the use of papers and also reduces the time of recording the data. This is the efficient way of storing the data in the database. Advantage of using C# and Visual Studio are

* Easy to write c# code and navigate between the database as it is Inbuilt plugin.
* The BlueBay Coffee Project can be used by more than 100 devices all around the world, while the admin system is connected to non-stop internet and server.
* Visual Studio is considered has best graphic designer user interface which gives it an upper hand compared to all other software.
* It's lightweight, fast, open source and cross-platform nature along with other cool features gives it an extra edge over any other editor.

# CHAPTER 2

### 2.1 ER DAIGRAM

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. The ER Diagram of our project is shown in the figure:2.2.1

Symbols in Entity Relationship:

Attribute Relationship

Entity Derived Attribute

Multivalued Attribute Weak Entity

|  |  |  |
| --- | --- | --- |
| **RELATONSHIP NAME** | **PARTICIPATION** | **CARDINALITY RATIO** |
| Ordered | COFFEE: BILLING | M: N |
| Services | CUSTOMER: BILLING | M: N |
| Services | EMPLOYEE: BILLING | M: N |
| Receiver | EMPLOYEE: PRODUCT | N: 1 |

Weak Relationship Key Attribute

Key Attribute

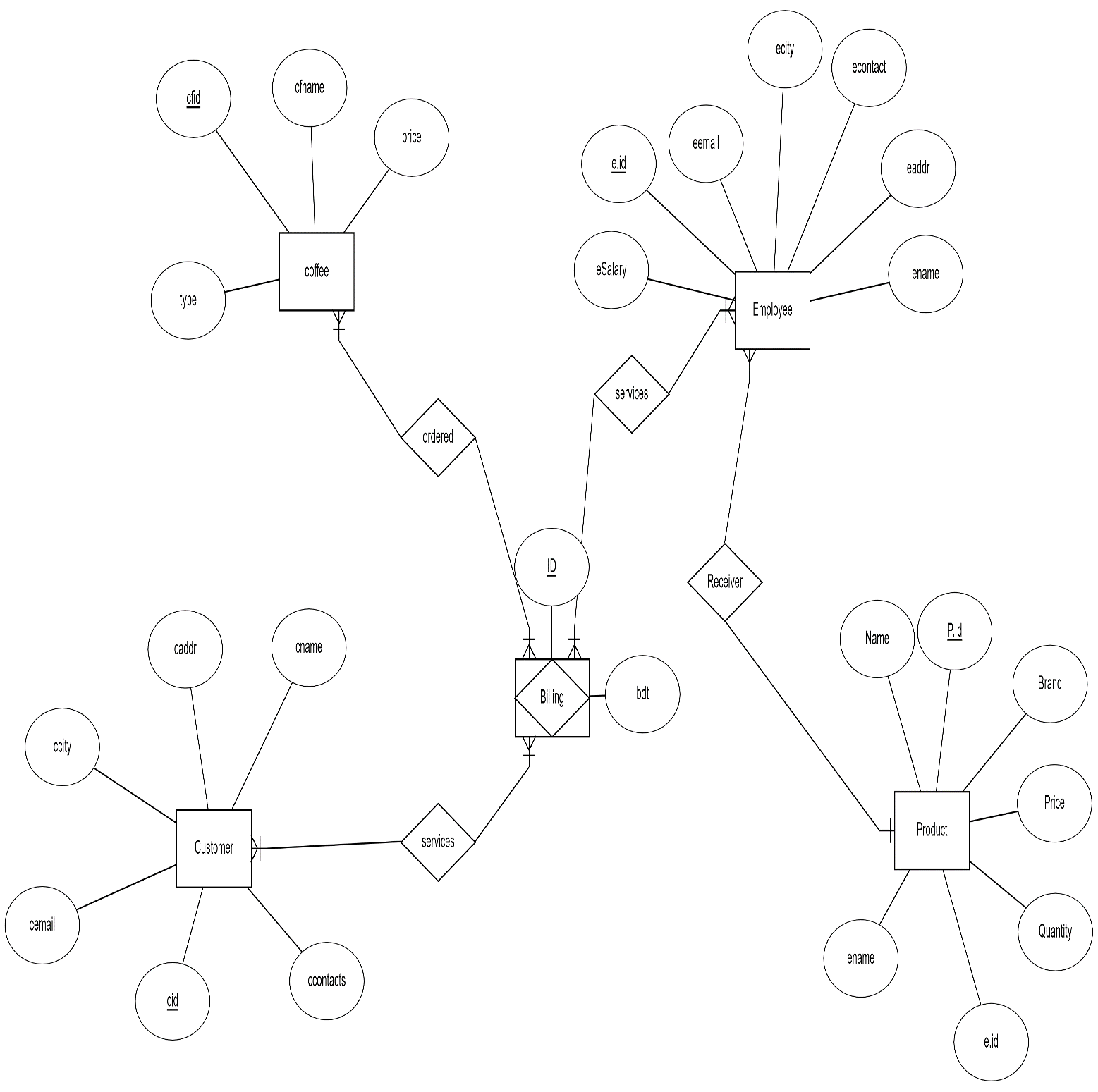
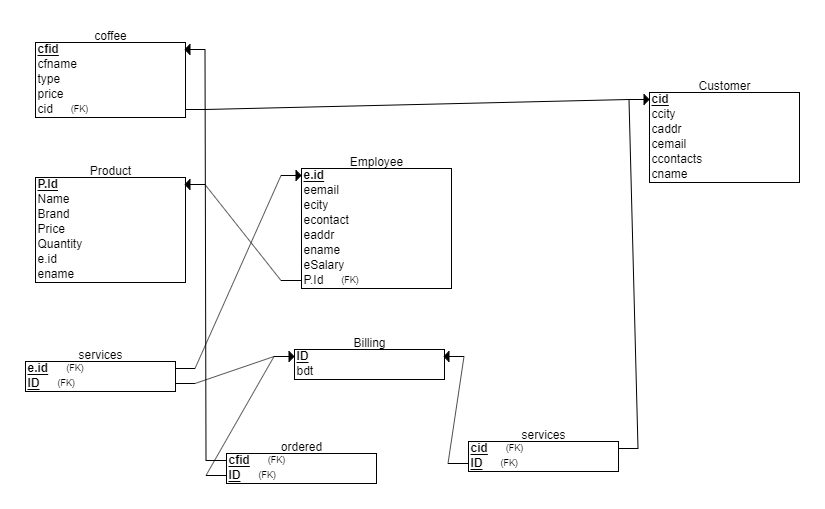


Figure:2.1.1 ER-Diagram of Coffee Database Management

### 2.2 Schema diagram

The Schema diagram of our project is shown in the figure:2.3.1. Coffee details, Employee details and Customer details are automatically fetched when the required id is written in the billing. Therefore, three foreign key are connected to the Billing. Connection between employee and product is mainly due to build a data of which employee handled which product in the stock.



# CHAPTER 3

**IMPLEMENTATION**

### System Specification

Operating System: Windows XP/7/8/10/MacOS/Linux.

Memory: Minimum of 2GB of RAM, Minimum of 2GB hard disk space. Backend: Microsoft Visual Studio 2019 SQL Server Express

Frontend: Microsoft Visual Studio 2019

### +Table Structure

#### Coffee Relation

CREATE TABLE [dbo].[acoffee] (

[Id] INT NOT NULL,

[name] NVARCHAR (50) NOT NULL,

[type] NVARCHAR (50) NULL,

[price] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([Id])

);

|  |  |
| --- | --- |
| **Name** | **Type** |
| Id | INT |
| Name | NVARCHAR(50) |
| Type | NVARCHAR(50) |
| Price | NVARCHAR(50) |
|  | |

Figure:3.2.1 Coffee Table Structure

#### Customer Relation

CREATE TABLE [dbo].[cust] (

[Id]INT NOT NULL,

[name] NVARCHAR (50) NULL,

[addr] NVARCHAR (50) NULL,

[city] NVARCHAR (50) NULL,

[contact] NVARCHAR (50) NULL,

[email] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

)

|  |  |
| --- | --- |
| **Name** | **Type** |
| Name | VARCHAR(50) |
| ID | INT |
| Addr | VARCHAR(50) |
| City | VARCHAR(50) |
| contact | VARCHAR(50) |
| Email | VARCHAR(50) |
|  | |

Figure:3.2.2 Customer Table Structure

#### Employee Relation

CREATE TABLE [dbo].[emp] (

[Id] INT NOT NULL,

[name] NVARCHAR (50) NULL,

[addr] NVARCHAR (50) NULL,

[city] NVARCHAR (50) NULL,

[contact] NVARCHAR (50) NULL,

[gender] NVARCHAR (50) NULL,

[email] NVARCHAR (50) NULL,

[doj] NVARCHAR (50) NULL,

[salary] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);

|  |  |
| --- | --- |
| **Name** | **Type** |
| ID | INT |
| Name | NVARCHAR(50) |
| Addr | NVARCHAR(50) |
| City | NVARCHAR(50) |
| Contact | NVARCHAR(50) |
| Gender | NVARCHAR(50) |
| Email | NVARCHAR(50) |
| Doj | NVARCHAR(50) |
| Salary | NVARCHAR(50) |
| Figure:3.2.3 Employee Table Structure | |

#### Product Relation

CREATE TABLE [dbo].[product] (

[Id] INT IDENTITY (1, 1) NOT NULL,

[p\_name] NVARCHAR (50) NULL,

[brand] NVARCHAR (50) NULL,

[price] NVARCHAR (50) NULL,

[quantity] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);

|  |  |
| --- | --- |
| **Name** | **Type** |
| ID | INT |
| P\_name | NVARCHAR(50) |
| Brand | NVARCHAR(50) |
| Price | NVARCHAR(50) |
| Quantity | NVARCHAR(50) |

Figure :3.2.4 Product Table Structure

#### Order Relation

CREATE TABLE [dbo].[p\_order] (

[Id] INT NOT NULL,

[cust\_no] NVARCHAR (50) NULL,

[name] NVARCHAR (50) NULL,

[addr] NVARCHAR (50) NULL,

[city] NVARCHAR (50) NULL,

[c\_name] NVARCHAR (50) NULL,

[price] NVARCHAR (50) NULL,

[date] NVARCHAR (50) NULL,

[EId] INT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

|  |  |
| --- | --- |
| **Name** | **Type** |
| ID | NUMBER(3) |
| Cust\_no | NVARCHAR(50) |
| Name | NVARCHAR(50) |
| Addr | NVARCHAR(50) |
| City | NVARCHAR(50) |
| C\_name | NVARCHAR(50) |
| Price | NVARCHAR(50) |
| Date | NVARCHAR(50) |
| Eid | INT |

Figure:3.2.5 Order Table Structure

### Functionalities

#### 3.3.1 Connecting to Database

The “BlueBay Coffee Database System” has been developed C#. It is used by Microsoft SQL Server Express database for storing the data. It is connected by the following syntax:

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sudhvina A.S\Downloads\CoffeeBluebay\CoffeShopManegementSystemCSharp\CoffeShopManegementSystemCSharp\coffee.mdf;Integrated Security=True");

#### 3.3.1 Add

Add operation is used to add an Customer details, Employee detail, Coffee detail, Product detail ,Place Order detail to the coffee database.

J

private void button1\_Click(object sender, EventArgs e)

{

try

{

new System.Net.Mail.MailAddress(this.textBox6.Text);

// return;

}

catch (ArgumentException e1)

{

MessageBox.Show("empty");

return;

}

catch (FormatException e2)

{

MessageBox.Show("invalid email");

return;

}

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sudhvina A.S\Downloads\CoffeeBluebay\CoffeShopManegementSystemCSharp\CoffeShopManegementSystemCSharp\coffee.mdf;Integrated Security=True");

con.Open();

try

{

string str = " INSERT INTO cust(Id,name,addr,city,contact,email) VALUES(" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "'); ";

SqlCommand cmd = new SqlCommand(str, con);

cmd.ExecuteNonQuery();

//-------------------------------------------//

string str1 = "select max(Id) from cust;";

SqlCommand cmd1 = new SqlCommand(str1, con);

SqlDataReader dr = cmd1.ExecuteReader();

if (dr.Read())

{

MessageBox.Show("Customers Information Registered Successfully..");

}

this.Close();

}

catch (SqlException ee)

{

MessageBox.Show(ee.Message);

}

con.Close();

}

#### 3.3.2 Delete

Delete operation is used to delete the entry made to the Customer details, Employee detail, Coffee detail, Product detail ,Place Order by pressing delete key from the firework database.

private void button3\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sudhvina A.S\Downloads\CoffeeBluebay\CoffeShopManegementSystemCSharp\CoffeShopManegementSystemCSharp\coffee.mdf;Integrated Security=True");

con.Open();

string str = "DELETE FROM cust WHERE Id = '" + textBox1.Text + "'";

SqlCommand cmd = new SqlCommand(str, con);

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show(" Customer Information is Removed Succefully");

SqlCommand cmd5 = new SqlCommand(str, con);

SqlDataAdapter da = new SqlDataAdapter(cmd5);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = new BindingSource(dt, null);

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

MessageBox.Show("Please Enter Customer Id..");

}

}

#### 3.3.3 View

View button is used to view the values entered into the table from coffee database

private void textBox1\_TextChanged(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sudhvina A.S\Downloads\CoffeeBluebay\CoffeShopManegementSystemCSharp\CoffeShopManegementSystemCSharp\coffee.mdf;Integrated Security=True");

con.Open();

if (textBox1.Text != "")

{

try

{

string getCust = "select name,addr,city,contact,email from cust where Id=" + textBox1.Text + " ;";

SqlCommand cmd = new SqlCommand(getCust, con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

if (dr.Read())

{

textBox2.Text = dr.GetValue(0).ToString();

textBox3.Text = dr.GetValue(1).ToString();

textBox4.Text = dr.GetValue(2).ToString();

textBox5.Text = dr.GetValue(3).ToString();

textBox6.Text = dr.GetValue(4).ToString();

}

else

{

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

}

}

catch (SqlException excep)

{

MessageBox.Show(excep.Message);

}

con.Close();

}

}

#### 3.3.4 Reset

Reset button is used to reset the values inserted in every attributes of each table

private void button5\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sudhvina A.S\Downloads\CoffeeBluebay\CoffeShopManegementSystemCSharp\CoffeShopManegementSystemCSharp\coffee.mdf;Integrated Security=True");

con.Open();

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

}

#### 3.3.5 Trigger

Trigger operation is used raise an error message if the entered amount is negative in coffee table.

CREATE TRIGGER [dbo].[PINS]

ON [dbo].[acoffee]

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Id int, @name nvarchar(50),@type nvarchar(50), @price int

DECLARE @MESSAGE VARCHAR(100)

SELECT @Id = INSERTED.Id,

@name = INSERTED.name,

@type = INSERTED.type,

@price = INSERTED.price

FROM INSERTED

IF(@price<0)

BEGIN

SET @MESSAGE = 'INVALID PRICE'

RAISERROR(@MESSAGE,16,1)

END

ELSE

BEGIN

INSERT INTO acoffee VALUES(@Id,@name,@type,@price)

END

END

#### 3.3.6 Stored Procedure

A procedure is created in our project to update the contents of table coffee.

CREATE PROCEDURE [dbo].UPDATECOFFEE

@Id int,

@name nvarchar (50),

@type nvarchar (50),

@price nvarchar (50)

AS

UPDATE acoffee SET name=@name, type=@type, price=@price WHERE Id=@Id

RETURN 0

# CHAPTER 4 RESULTS

BlueBay Coffee is a database software where user purchases coffee in shop. Our Coffee have many categories for customers and placing order is much easier as it stores entire value of coffee, customer key details and employee key details in the billing of the system

## Snapshots

### Login Page

It is a multiple user login system, where one ca

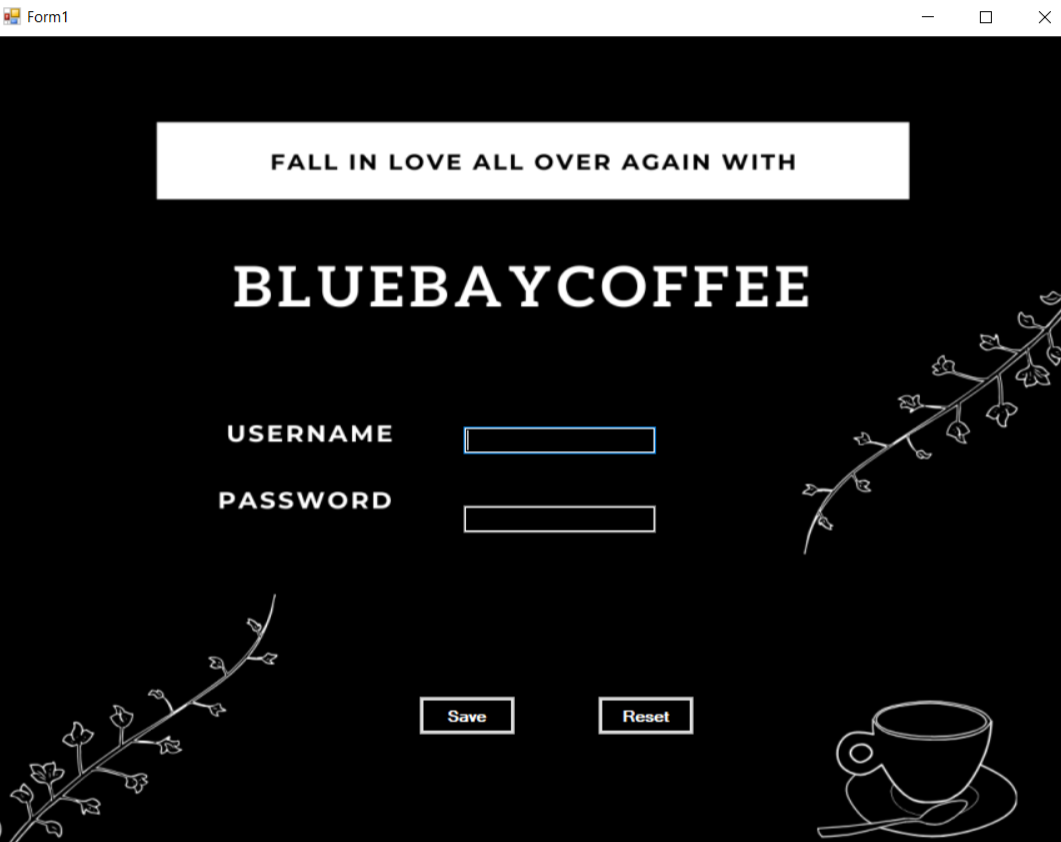


Figure: 4.1.1 Login Page

### Home Page

In home page the tables in the database can be selected to insert, delete, display, etc.

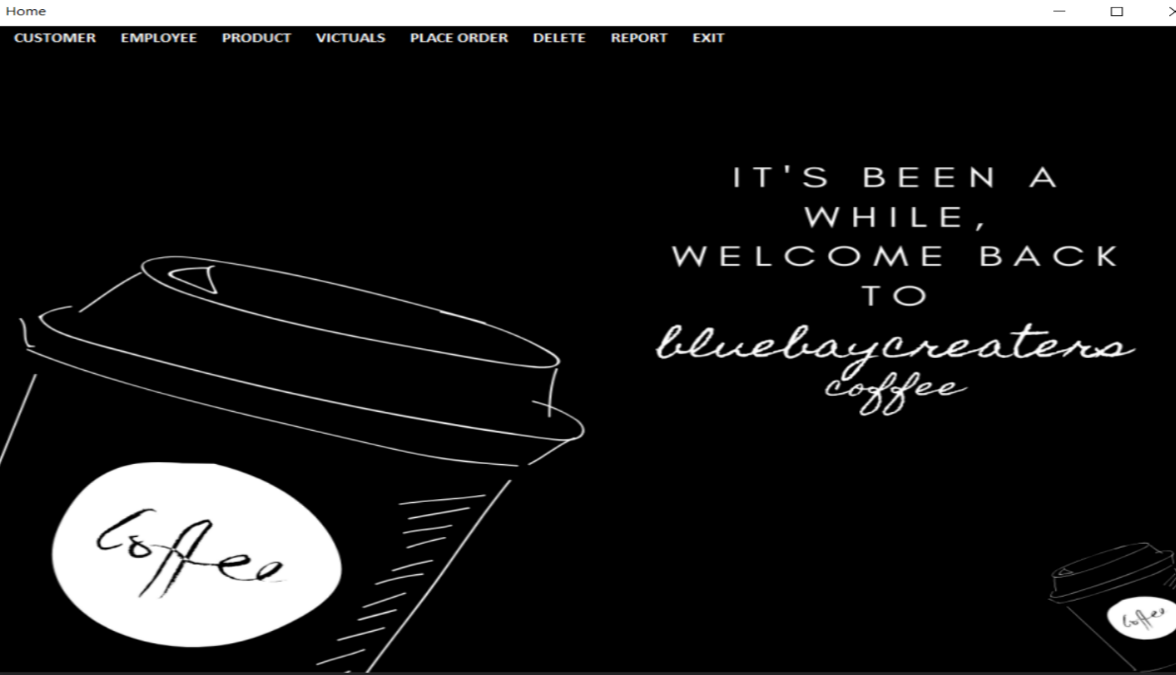


Figure: 4.1.2 Home Page

### Coffee details

In Coffee page we can insert, delete, modify and display the details of Coffee and Snacks

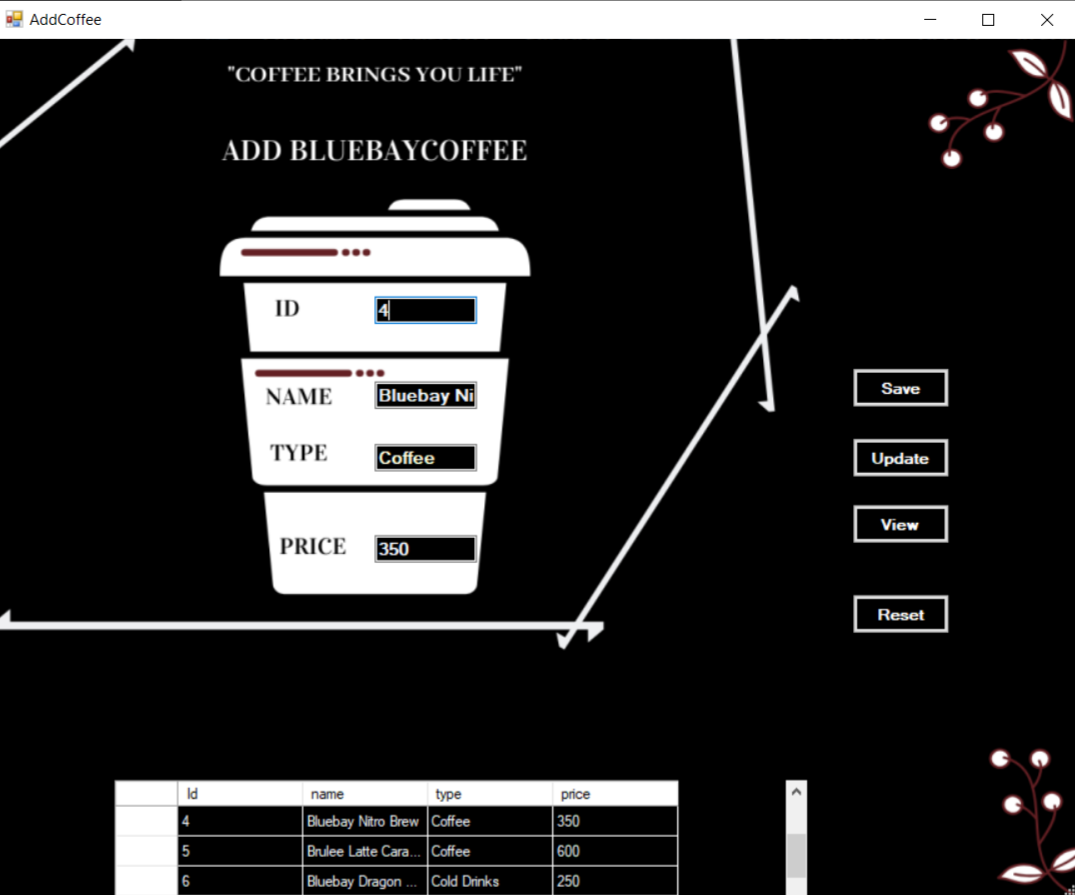


Figure: 4.1.3 Coffee details

### Customer details

In customer page we can insert, delete, modify and display the details of customer.

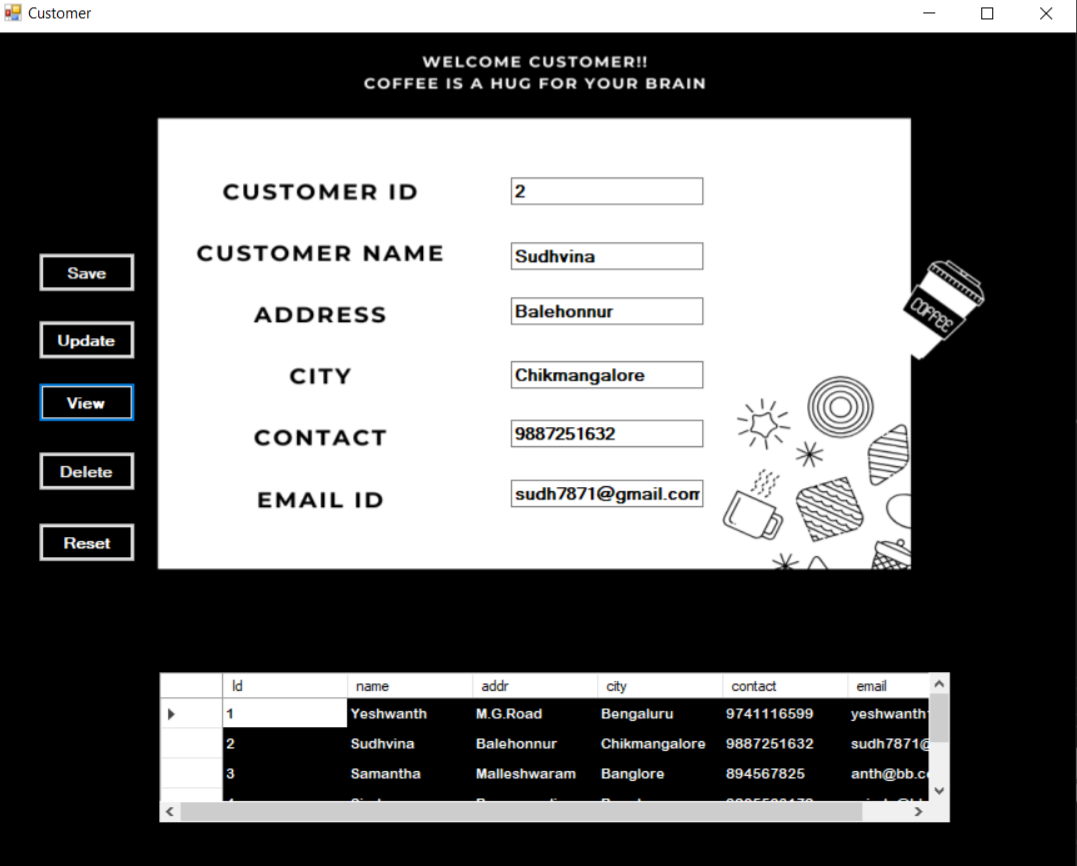


Figure: 4.1.4 Customer details

### Place Order details

In billing page, we can insert, delete, modify and display the details of billing details



Figure: 4.1.5 Place Order details

### Employee details

In Employee page we can insert, delete, modify and display the details of Employee.

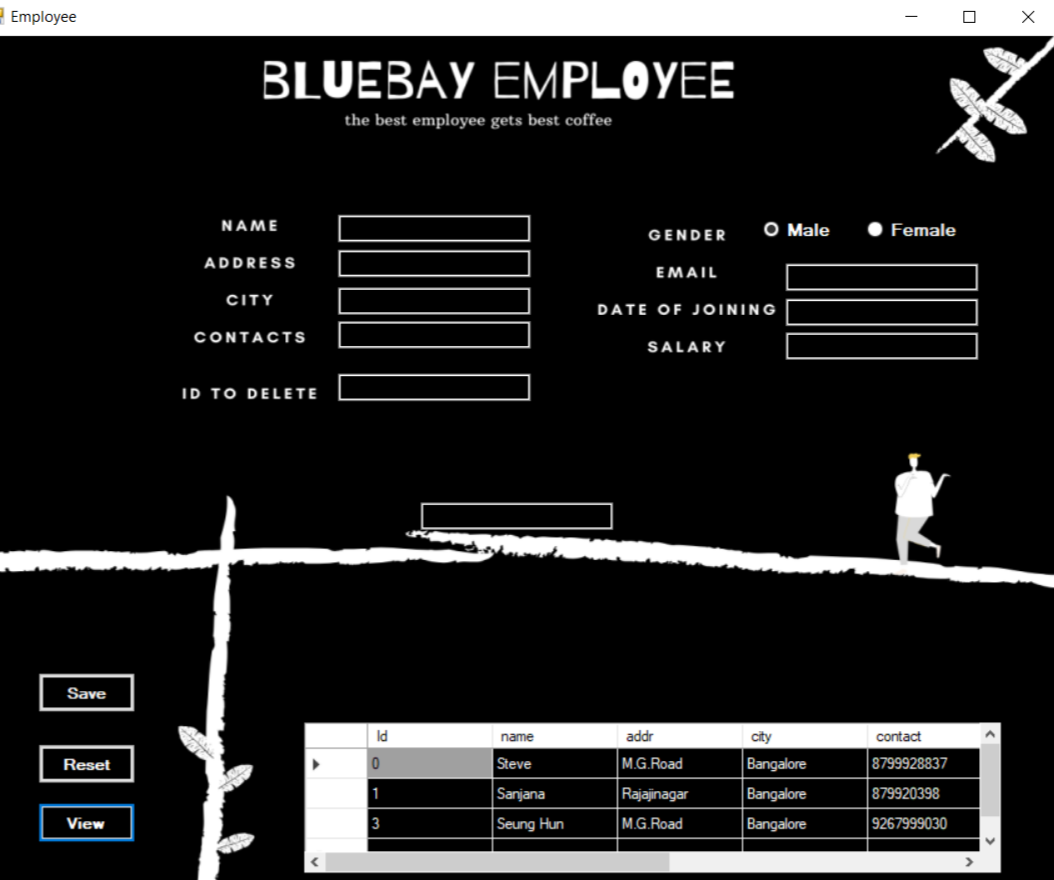


Figure: 4.1.6 Employee details

### Product details

In Product page we can insert, delete, modify and display the details of ingredients required for making coffee or snacks

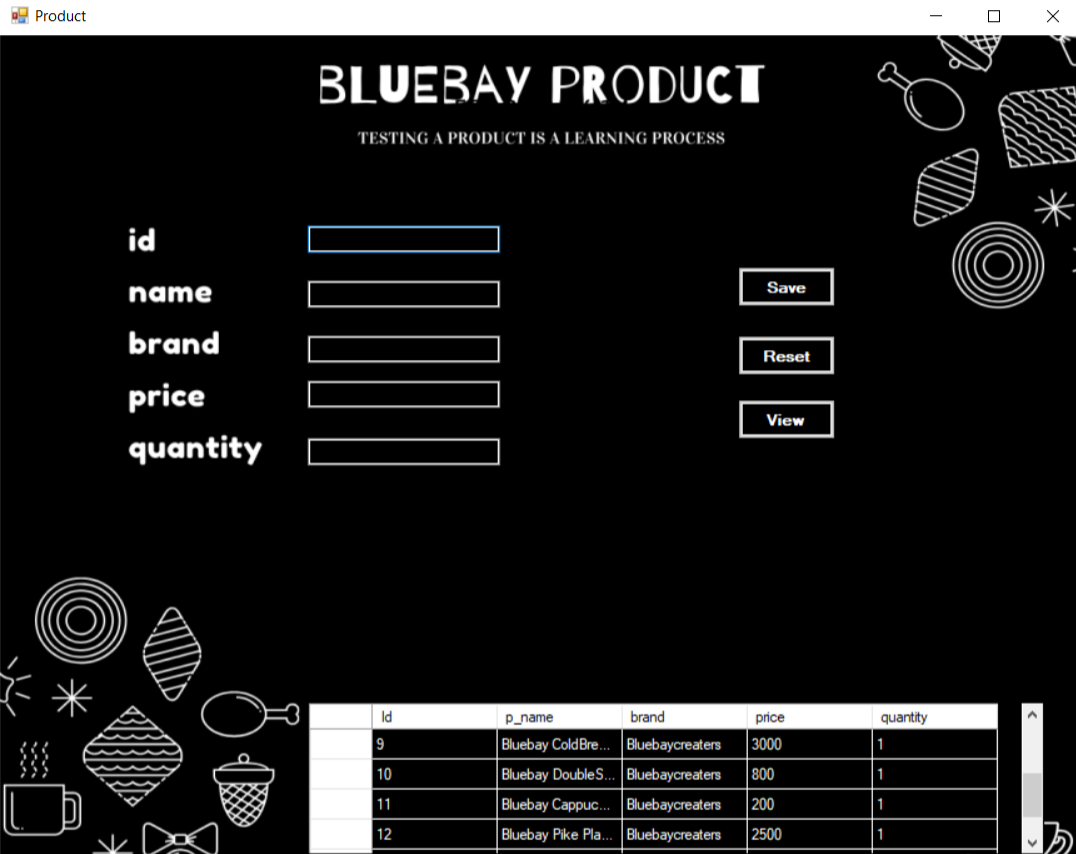
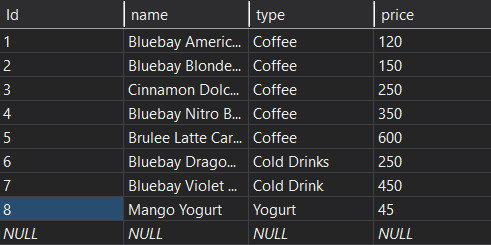
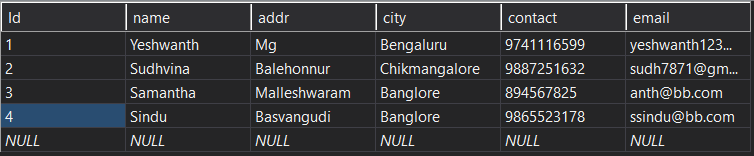


Figure: 4.1.7 Product details

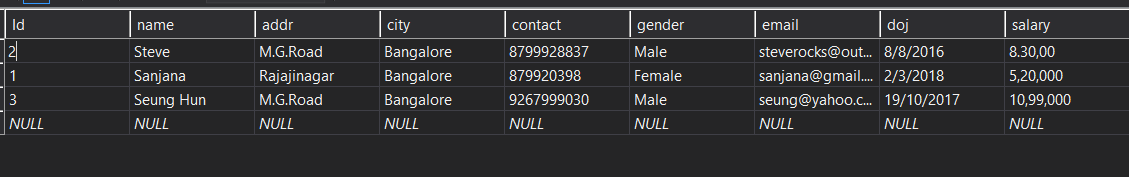
* + 1. **Select \* from Coffee**



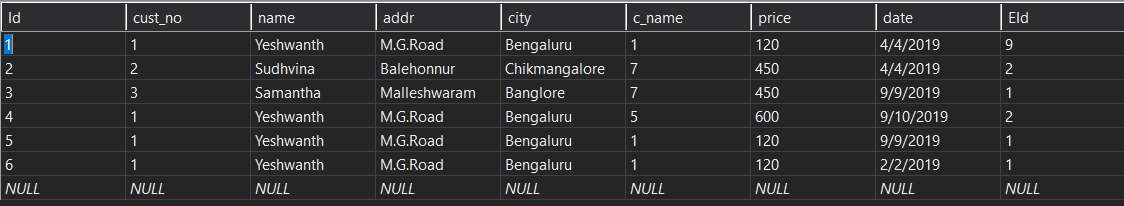
* + 1. **Select \* from Customer**



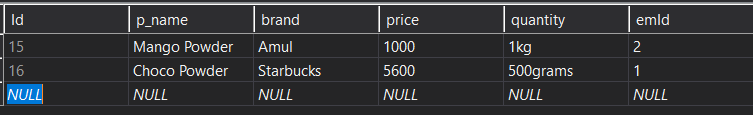
**4.1.10 Select \* from Employee**



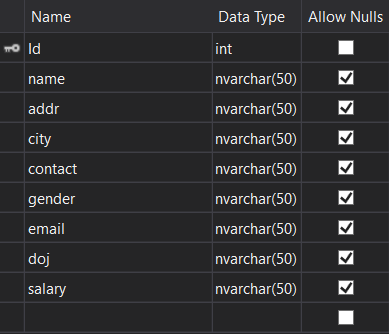
* + 1. **Select \* from Place Order**



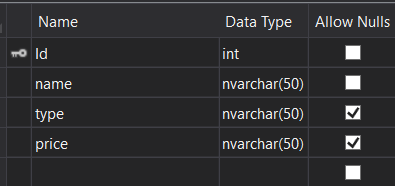
* + 1. **Select \* from Product**



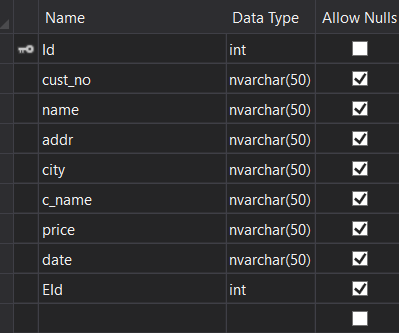
**4.1.13 Desc Employee**



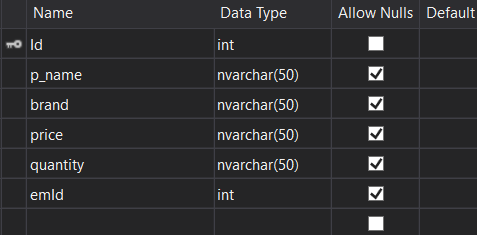
* + 1. **Desc Coffee**



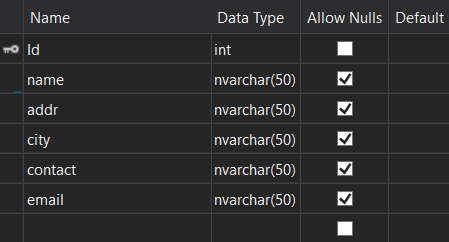
* + 1. **Desc Order**



* + 1. **Desc Product**



* + 1. **Desc Customer**



## CONCLUSION

It has been a matter of immense pleasure, honor and challenge to have this opportunity to take up this project and complete it successfully. Our project maintains the entire database of an BlueBay Coffee Management. The software can be used in any Coffee shop even any snacks shop by just changing the attribute names, to maintain its coffee, products , employee, customer and Place Order details. While developing this project we have learnt a lot about Coffee database, we have also learnt how to make it user friendly (easy to use and handle) by hiding the complicated parts of its from users. During the development process we studied more about developing a software, how to implement the backend stored database in the real time system. We have tried to implement the project making it as user friendly and error free as possible. In future this project can be improved by recording the day-to-day activities taking place in a coffee database. So our project proves that, without investing money and by providing the required efforts we can make many people life move easier.

**REFERENCES**

1. C# 7.0 Pocket Reference: Instant help for Beginners by Ben Albhari and Joseph Albhari.
2. Raghu Ramakrishnan, and Gehrke, “Database management systems”, McGraw Hill, 3rd Edition, 2014,
3. C# for Dummies by John Paul and Bill Sempf 1st Edition, Kindle Edition.