



AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

**Dept. of Computer Science
Faculty of Science and Technology**

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Fall 2024-2025

Section: [H]

Group No: 01

Project Report On

Project Name [AgriMarket-Pro]

Supervised By

TASLIMUR RAHMAN

Submitted By:

Name	ID
1. SHOWRAV GHOSH	23-50666-1
2. EKRAMUL HASIB	23-51106-1
3. ABDUL ALIM	23-51105-1
4. MD. ARIFUL HASSAN COWDURY	22-49551-3

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 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.
1. Chapter: 01 (Cover Page)-----	01
2. Chapter: 02 (Table Of Content)-----	04
3. Chapter: 03 (Introduction)-----	05
4. Chapter: 04 (User Story)-----	08
5. Chapter: 05a (ER Diagram)-----	11
6. Chapter: 05b (SQL Queries)-----	12
7. Chapter: 06 (Screenshots)-----	19

Introduction:

The **AgriMarket-Pro** is a **desktop-based application** developed to assist administrators in efficiently managing **employees, farmers, users, products, and financial transactions**. The system is designed to provide a structured and **easy-to-use interface** that helps reduce manual work and errors while improving efficiency in handling records.

Many organizations, especially those dealing with **multiple employees and customers**, require an efficient **database management system** that allows them to store and retrieve information quickly. This system **eliminates the need for traditional paperwork and manual record-keeping** by providing a digital solution that ensures data is **secure, well-organized, and easily accessible**.

The system includes **user authentication**, allowing only authorized admins to access and manage data. The **dashboard** serves as a central point where admins can navigate different sections like **Employee Management, Farmer Management, User Management, Product Management, and Payment Tracking**.

One of the key features of this system is **payment management**, where admins can track **unpaid users** and manage **financial transactions**. Users can pay through multiple payment methods like **Bkash, Brac Bank, and City Bank**, and all transactions are recorded in the database. The system also has a **search and filtering option** that allows admins to **find records instantly**.

Additionally, the system ensures **data security** by allowing **deleted records to be restored** instead of being permanently erased. If an admin mistakenly removes an **employee, farmer, or user**, they can **recover the data from the deleted list**.

With a **user-friendly graphical interface (GUI)** built using **Windows Forms (WinForms) and C#**, and a **strong database foundation using Microsoft SQL Server**, this system provides an **efficient, reliable, and secure way to manage organizational records**.

Objectives:

The main goals of this **AgriMarket-Pro** are:

- Streamline Administration** – Reduce manual record-keeping by providing a **centralized digital platform** for managing employees, farmers, and users.
- Ensure Data Security** – Prevent **unauthorized access** with a **secure login system** and **store records safely** in a SQL database.
- Simplify Employee and Farmer Management** – Allow admins to **add, edit, delete, and search records quickly**.
- Enable Payment Tracking** – Keep a **record of pending payments** and allow users to **pay using multiple methods**.
- Improve Data Retrieval** – Provide **fast searching and filtering** features to find information **instantly**.
- Allow Restoration of Deleted Records** – Ensure that accidentally deleted employees, farmers, or users **can be restored easily**.
- User-Friendly Interface** – Develop a **clean and simple UI** that is **easy to navigate** for admins.

Technical Stack:

This system is built using **modern and efficient technologies** to ensure smooth operation:

- ◆ **Programming Language:** C# (.NET Framework) – Used to develop the core functionality of the system.
- ◆ **User Interface:** Windows Forms (WinForms) – A **graphical user interface (GUI)** for a smooth user experience.
- ◆ **Database:** Microsoft SQL Server (LocalDB) – Stores all records securely and allows **quick retrieval of data**.
- ◆ **Development Environment:** Visual Studio – Provides an **integrated development environment (IDE)** for writing, debugging, and testing the application.
- ◆ **Database Connectivity:** ADO.NET – Used to connect **C# with the SQL Server database**.

Features:

The AgriMarket-Pro is packed with **powerful features** that help administrators manage data **efficiently and securely**:

- 1. Admin Dashboard**
 - Provides a **centralized control panel** where admins can access all features.
 - Quick navigation to **Employee, Farmer, User, Product, and Payment sections**.
- 2. Secure Login System**
 - **Ensures only authorized admins** can access the system.
 - Prevents **unauthorized modifications or data leaks**.
- 3. Employee Management**
 - Add **new employees** to the database with details like **name, salary, phone number, and gender**.
 - **Update employee records** when needed.
 - **Delete employees** (deleted records can be restored later).
 - **Search employees** by name or phone number for quick access.
- 4. Farmer Management**
 - Register and **manage farmers** in the system.
 - Store **farmer details** including **name, birth date, address, phone number, and gender**.
 - **Search and filter farmers** based on their details.
 - **Restore deleted farmers** from the deleted list.
- 5. User Management**
 - Register and manage **users/customers** in the system.
 - Maintain records of **contact information, addresses, and transactions**.
 - Allow **searching for users by name or phone number**.
 - **Restore deleted users** if needed.
- 6. Payment Management**
 - View a **list of unpaid users** and track outstanding payments.
 - Allow users to **make payments through different methods** (Bkash, Brac Bank, City Bank).
 - Record **transaction history** in the database for future reference.
 - Generate **real-time payment reports**.

7. Product Management

- Maintain a **list of products** for business operations.
- Track **product name, quantity, and price**.
- View **all available and sold products**.

8. Search and Filtering

- Quickly find **employees, farmers, and users** using search and filtering options.
- Reduce time spent on searching records by applying specific filters.

9. Restore Deleted Records

- If an admin accidentally deletes an employee, farmer, or user, the system allows them to **restore the record**.
- Ensures **data is not permanently lost**.

10. Database Connectivity

- All data is stored securely in SQL Server, ensuring it is not lost in case of system failure.
- The system automatically retrieves and updates records in the database when changes are made.

How It Works:

1. Admin Login

Admins enter **login credentials** to access the system. Only **authorized users** can enter.

2. Managing Employees, Farmers, and Users

- Admins can **add new records**.
- They can **search, update, and delete records**.
- Deleted records are **stored separately** for recovery if needed.

3. Payment Tracking

- The system shows a **list of unpaid users**.
- Users can **make payments using different methods**.
- Transactions are **automatically recorded in the database**.

4. Product Management

- Admins can **view and manage products**.
- The system keeps track of **available and sold products**.

5. Searching and Filtering

Admins can quickly find **specific records** without having to manually scroll through lists.

6. Restoring Deleted Records

If a **record is deleted by mistake**, it can be **restored from the deleted list**.

Conclusion:

The AgriMarket-Pro System is a **powerful tool** that helps organizations manage employees, farmers, users, products, and payments with ease. It saves time, reduces errors, and ensures data security by storing information in a well-organized database. By automating record management and payment tracking, this system significantly improves efficiency and allows admins to focus on business operations instead of manual paperwork.

With features like secure login, search filters, database integration, and record restoration, this system is an ideal solution for businesses that require an organized and secure way to manage their administrative tasks.

USER STORY

Introduction:

This case study describes an entity-relationship (ER) diagram that represents a system where farmers, users, and admins interact with products, payments, and accounts. The system helps in managing transactions between different entities in an organized way.

Entities and Attributes:

Admin

- **Admin_ID** (Primary Key)
- Name
- Email
- Phone_Number
- Password
- Gender

Employee

- **Employee_ID** (Primary Key)
- Name
- Salary
- Phone_Number
- Gender

Farmer

- **Farmer_ID** (Primary Key)
- Name
- Birth_Date
- Address
- Division
- Postal_Code
- Phone_Number
- Email
- Gender
- Password

User (Customer)

- **User_ID** (Primary Key)
- Name
- Birth_Date
- Address
- Division
- Postal_Code
- Phone_Number
- Email
- Gender
- Password

Product

- **Product_ID** (Primary Key)
- Product_Name
- Product_Quantity
- Product_Price

Payment

- **Payment_ID** (Primary Key)
- User_ID (Foreign Key)
- Payment_Method (Bkash, Brac Bank, City Bank, etc.)
- Payment_Status (Paid/Unpaid)
- Amount
- Transaction_Date

Deleted_Records (for restoring data)

- **Record_ID** (Primary Key)
- Table_Name (Employee, Farmer, User)
- Deleted_Data (JSON format or separate columns for each field)
- Deletion_Date

Entities and Their Roles:

1. Farmer

- Farmers are individuals who sell products like fruits, fish, and vegetables.
- They have attributes like Name, Address, Gender, Division, Postal Code, Phone Number, and Email.
- Each farmer has an ID (Far_Id) and a Password for security.

2. User

- Users are buyers who purchase products from Admin.
- They have similar attributes to farmers: Name, Address, Gender, Division, Postal Code, Phone Number, Email, and Birthdate.
- Each user has a unique ID (U_Id) and a Password.

3. Admin

- The Admin manages the system and has control over accounts.
- Admins have an Admin ID (Ad_Id), Name, Email, Password, and Gender.
- They also have a Phone Number.

4. Employee

- Employees manage different tasks within the system.
- They have an Employee ID (Emp_Id), Name, Salary, and Gender.
- They also have a Phone Number.

5. Product

- Products are the items sold by farmers to Admin
- Each product has a unique Serial Number.
- Products include items like Hilsa, Papaya, and Apple.
- Other attributes include Product Name, Product Quantity, Product Price, and Total Price.

6. Account

- Each entity has an account for managing financial transactions.
- Admins and Employees have accounts to track salaries and payments.

7. Paid and Unpaid Transactions

- When a user purchases a product, the payment status can be either Paid or Unpaid.
- Unpaid transactions contain details like Phone Number and Total Amount.
- Paid transactions include details like Serial Number and Payment Amount.

8. Payment System

- The system allows payments through different banks and mobile banking services.

- Supported services include Brac Bank, City Bank, Rocket, Nagad, and Bkash.
- Payment details include Serial Number, Phone Number, User ID, and Account Number.

Relationships in the System:

1. Farmers can Sell Products

- Farmers can list their products for sale in the system.
- Each product has a unique Serial Number and details like Product Name, Quantity, and Price.

2. Users can Buy Products

- Users can purchase products from Admin.
- Each purchase transaction is linked to an account.

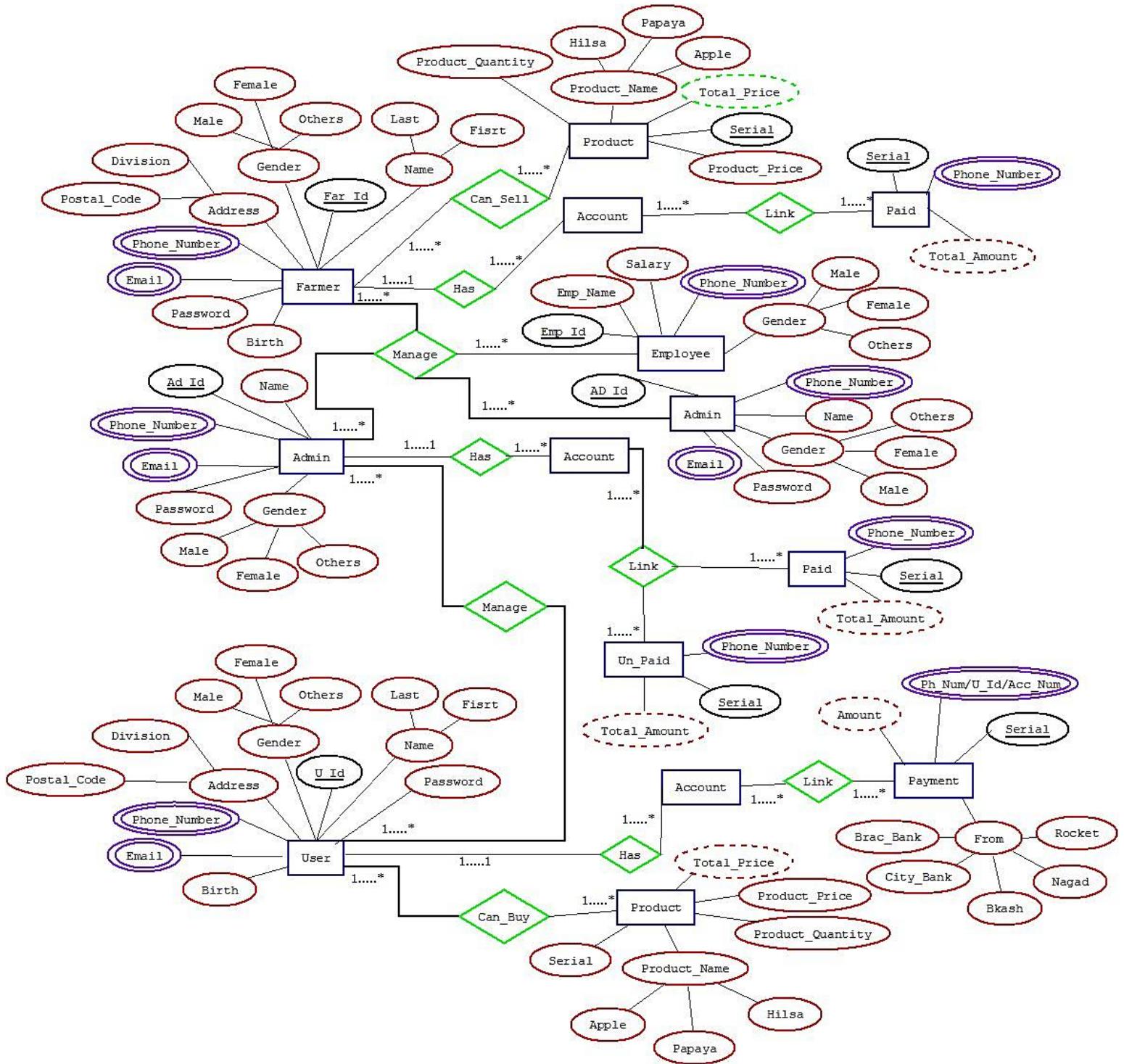
3. Admins Manage the System

- Admins oversee Farmers, Users, Employee and accounts.
- They also handle payments and ensure that transactions are recorded properly.
- Admin ensures that payments are processed correctly.
- They track Paid and Unpaid transactions.

Conclusion:

This ER diagram represents a well-structured system for managing agricultural product sales and payments. It efficiently connects farmers, users, admins, and employees, ensuring smooth transactions and account management. The system supports secure payments and helps maintain financial records effectively.

ER Diagram



SQL Queries

Admin Table:

```
CREATE TABLE [dbo].[Admin] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Email Address] VARCHAR (50) NOT NULL,
    [Password]  VARCHAR (50) NOT NULL,
    [Gender]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Name	Phone Number	Email Address	Password	Gender
▶	1	admin	01910917566	admin@gmail.com	@Ad123	Male
	6	Abir	01910917588	ab@gmail.com	@Ab123	Male
	7	Test	01234567891	test@gmail.com	!@12Ts	Male
*	8	Showrav Ghosh	01910917567	sg@gmail.com	@Sg123	Male

Delete Admin Table:

```
CREATE TABLE [dbo].[Dela] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Email Address] VARCHAR (50) NOT NULL,
    [Password]  VARCHAR (50) NULL,
    [Gender]   VARCHAR (50) NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Name	Phone Number	Email Address	Password	Gender
▶	2	Test	01234567891	test@gmail.com	!@12Ts	Male
*						

Employee Table:

```
CREATE TABLE [dbo].[Employee List] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Employee Name] VARCHAR (50) NOT NULL,
    [Salary]   VARCHAR (50) NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Gender]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Employee Name	Salary	Phone Number	Gender
▶	1	Showrav Ghosh	2000000	01910917566	Male
	5	Test1	30000	01912345678	Male
	8	Abir	890000	01910917565	Male
	10	Ayan	20004	01910917567	Male
	12	Abtahi	2000	01315477832	Male
	14	Abed	50000	01910917552	Male
*					

Delete Employee Table:

```
CREATE TABLE [dbo].[DelEmp] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Employee Name] VARCHAR (50) NOT NULL,
    [Salary]   VARCHAR (50) NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Gender]   NCHAR (10) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Employee Name	Salary	Phone Number	Gender
►	4	Abed	50000	01910917552	Male
*					

Farmer Table:

```
CREATE TABLE [dbo].[Farmer] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Birth]   VARCHAR (50) NOT NULL,
    [Address] VARCHAR (50) NOT NULL,
    [Division] VARCHAR (50) NOT NULL,
    [Postal Code] INT      NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Email]    VARCHAR (50) NOT NULL,
    [Gender]   VARCHAR (50) NOT NULL,
    [Password] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
```

Delete Farmer Table:

```
CREATE TABLE [dbo].[Delf] (
    [Id]      INT      IDENTITY (1, 2) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Birth]    VARCHAR (50) NOT NULL,
    [Address]  VARCHAR (50) NOT NULL,
    [Division] VARCHAR (50) NOT NULL,
    [Postal Code] INT      NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Email]    VARCHAR (50) NOT NULL,
    [Gender]   VARCHAR (50) NOT NULL,
    [Password] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Name	Birth	Address	Division	Postal Code	Phone Number	Email	Gender
1	Test 2		01-02-2000	Dhaka	Dhaka	1234	01234543212	tsd@gmail.com	Male

User Table:

```
CREATE TABLE [dbo].[UserList] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Birth]    VARCHAR (50) NOT NULL,
    [Address]  VARCHAR (50) NOT NULL,
    [Division] VARCHAR (50) NOT NULL,
    [Postal Code] INT      NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Email]    VARCHAR (50) NOT NULL,
    [Gender]   VARCHAR (50) NOT NULL,
    [Password] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

	Id	Name	Birth	Address	Division	Postal Code	Phone Number	Email	Gender
1	Showrav Ghosh	30-09-2002	Maheshpur	Khulna	4080	01910917566	showravghosh19....	Male	
3	Showrav Ghosh	30-09-2002	Maheshpur	Khulna	4080	01910917567	showravghosh19....	Male	
5	Razin Ahmed	12-12-2004	Dhanmundi	Dhaka	1240	01739296421	razin@gmail.com	Male	
7	Ayan Halder	27-05-2003	dhaka	Dhaka	4000	01611237016	ayan@gmail.com	Male	

Delete User Table:

```
CREATE TABLE [dbo].[Delu] (
    [Id]      INT      IDENTITY (1, 1) NOT NULL,
    [Name]    VARCHAR (50) NOT NULL,
    [Birth]    VARCHAR (50) NOT NULL,
    [Address]  VARCHAR (50) NOT NULL,
    [Division] VARCHAR (50) NOT NULL,
    [Postal Code] INT      NOT NULL,
```

```

[Phone Number] VARCHAR (50) NOT NULL,
[Email]      VARCHAR (50) NOT NULL,
[Gender]     VARCHAR (50) NOT NULL,
[Password]   VARCHAR (50) NULL,
PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

	Id	Name	Birth	Address	Division	Postal Code	Phone Number	Email	Gender
▶	2	Razin Ahmed	12-12-2004	Dhanmundi	Dhaka	1240	01739296421	razin@gmail.com	Male
*									

Fish Product For Farmer Table:

```

CREATE TABLE [dbo].[Fish] (
[Serial]      INT      IDENTITY (1, 1) NOT NULL,
[Product Name]  VARCHAR (50) NOT NULL,
[Product Quantity] VARCHAR (50) NOT NULL,
[Product Price]  VARCHAR (50) NOT NULL,
[Total Price]   VARCHAR (50) NOT NULL,
PRIMARY KEY CLUSTERED ([Serial] ASC)
);

```

	Serial	Product Name	Product Quantity		Unit Quantity	Product Price	Total Price
▶	8	Pointed Gourd	2	▶		2.00	4
*				*			

Fruit Product For Farmer Table:

```

CREATE TABLE [dbo].[Fruit] (
[Serial]      INT      IDENTITY (1, 1) NOT NULL,
[Product Name]  VARCHAR (50) NOT NULL,
[Product quantity] VARCHAR (50) NOT NULL,
[Product Price]  VARCHAR (50) NOT NULL,
[Total Price]   VARCHAR (50) NOT NULL,
PRIMARY KEY CLUSTERED ([Serial] ASC)
);

```

	Serial	Product Name	Product Quantity
▶	8	Pointed Gourd	2
*			

	Unit	Product Price	Total Price
▶		2.00	4
*			

Vegetable Product For Farmer Table:

```
CREATE TABLE [dbo].[Vegetable] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Product Name]  VARCHAR (50) NOT NULL,
    [Product Quantity] VARCHAR (50) NOT NULL,
    [Product Price]  VARCHAR (50) NOT NULL,
    [Total Price]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Product Name	Product Quantity
▶	8	Pointed Gourd	2
*			

	Unit	Product Price	Total Price
▶		2.00	4
*			

Fish Product For User Table:

```
CREATE TABLE [dbo].[UFish] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Product Name]  VARCHAR (50) NOT NULL,
    [Product Quantity] VARCHAR (50) NOT NULL,
    [Product Price]  VARCHAR (50) NOT NULL,
    [Total Price]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Product Name	Product Quantity		uct tity	Product Price	Total Price
▶	8	Pointed Gourd	2			2.00	4
●							

Fruit Product For User Table:

```
CREATE TABLE [dbo].[UFRUIT] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Product Name]  VARCHAR (50) NOT NULL,
    [Product Quantity] VARCHAR (50) NOT NULL,
    [Product Price]  VARCHAR (50) NOT NULL,
    [Total Price]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Product Name	Product Quantity		uct tity	Product Price	Total Price
▶	8	Pointed Gourd	2			2.00	4
●							

Vegetable Product For User Table:

```
CREATE TABLE [dbo].[UVegetabl] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Product Name]  VARCHAR (50) NOT NULL,
    [Product Quantity] VARCHAR (50) NOT NULL,
    [Product Price]  VARCHAR (50) NOT NULL,
    [Total Price]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Product Name	Product Quantity
▶	8	Pointed Gourd	2
*			

	Product Quantity	Product Price	Total Price
▶		2.00	4
*			

Transaction After Payment Req-Farmer Table:

```
CREATE TABLE [dbo].[Paid] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Phone Number] VARCHAR (50) NOT NULL,
    [Total Amount] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Phone Number	Total Amount
▶	1	01955188815	2.40
	2	01955188815	2.40
	3	01955188815	2.40
	4	01893371881	8.00
	5	01925354679	16.50
	6	01693783243	21.50
	7	01336456154	4.00
	8	01786360561	13.60
*			

Transaction After Payment Done For Farmer Payment Request Table:

```
CREATE TABLE [dbo].[Unpaid] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Phone Number] VARCHAR (50) NULL,
    [Total Amount] VARCHAR (50) NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

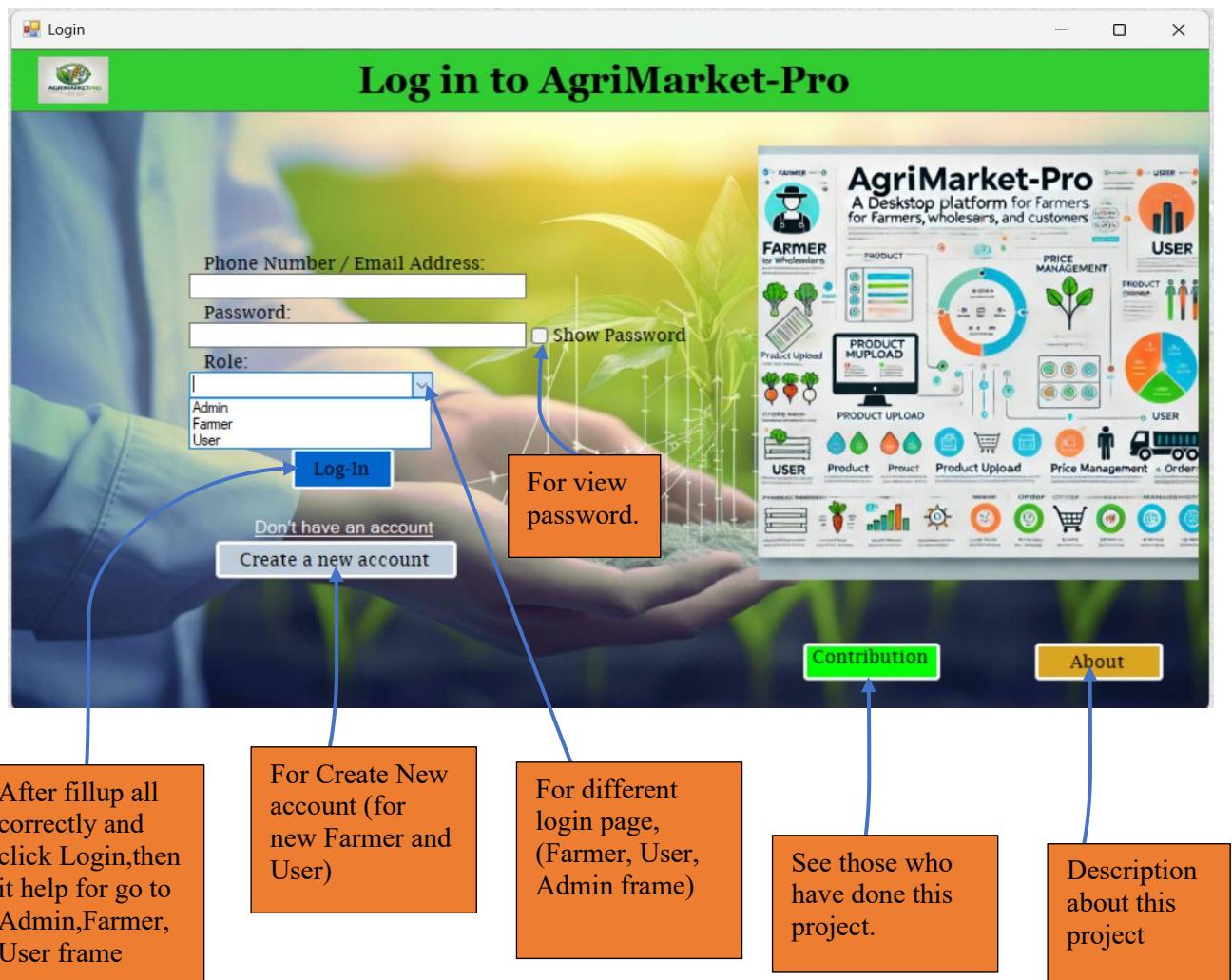
	Serial	Phone Number	Total Amount
▶	1	1234567890	6.40
	2	1234567890	9.60
	3	1234567890	2.40
	15	01474997607	4.00
*			

Transaction After Payment Done User Table:

```
CREATE TABLE [dbo].[Upaid] (
    [Serial]      INT      IDENTITY (1, 1) NOT NULL,
    [Ph_Num/U_ID/Acc_Num] VARCHAR (50) NULL,
    [Amount]      VARCHAR (50) NULL,
    [From]        NCHAR (10) NULL,
    PRIMARY KEY CLUSTERED ([Serial] ASC)
);
```

	Serial	Ph_Num/U_ID/Acc	Amount	From
▶	1	01910917566	3.00	Bkash
	2	453456	4.40	BracBank
	3	01910915566	30.00	Bkash
	4	01987564544	30.00	Nagad
	5	01646745342	30.00	Rocket
	6	234562345678	30.00	CityBank
	7	453278	30.00	BracBank
	8	01783645431	23.00	Bkash
*	9	775434	4.50	BracBank

Screenshots:



Registration

The registration form includes fields for First Name, Last Name, Birth (D/M/YY), Address, Division, Postal Code, Phone Number, Email Address, Gender (Male, Female, Other), Password, Re-enter Password, and Role. It features a 'Register' button, a 'Reset' button, and a 'View' link for passwords.

Annotations:

- Back Previous Frame.**: Points to the back/previous frame button in the top right corner.
- For select DOB from this ComboBox.**: Points to a calendar control showing February 2025, with the date 05-02-2025 selected.
- For select Division from this comboBox**: Points to a dropdown menu listing districts: Disha, Chittagong, Khulna, Rangpur, Comilla, Mymensingh, Pabna, Rajshahi, Barisal, and Sylhet, with 'Rangpur' selected.
- Select Role (Farmer or User) From this ComboBox.**: Points to a dropdown menu listing 'Farmer' and 'User', with 'Farmer' selected.
- After type all data and then click Register for new account**: Points to the 'Register' button.
- For Remove all Text and selected field.**: Points to the 'Reset' button.

About

AgriMarket-Pro: A Desktop Platform for Farmers, Wholesalers, and Customers

AgriMarket-Pro is a user-friendly desktop application designed to connect farmers, wholesalers, and customers in one platform. It helps farmers upload their seasonal products, allows admins to manage product prices and set minimum order quantities, and enables users to browse and order products easily.

Key Features:

- Farmers can add product details like name, quantity, and season and manage their listings.
- Admins control pricing, set order limits, and monitor the system for smooth operation.
- Users (Wholesalers or Customers) can search for products, view details like price and minimum order quantity, and place orders that meet the set criteria.

The platform includes dynamic pricing updated daily, special categories like seasonal and best-selling products, and reporting tools for admins. Built using WinForms, C#, and SQL Server, AgriMarket-Pro ensures efficient management of agricultural trade with a focus on simplicity and accessibility. Future upgrades may include payment gateways, notifications, and a mobile app.

The application interface shows a dashboard with sections for FARMER, WHOLESALE, and USER. Key features include Product Upload, Price Management, and Order tracking. Icons represent various functions like Product Upload, Price Management, and Order.

Annotations:

- Back Previous Frame.**: Points to the back/previous frame button in the top right corner.

Contribution

Contribution Team

The screenshot shows a window titled "Contribution" with a header "Contribution Team". It displays four team members in a grid:

- Name: SHOWRAV GHOSH**
ID: 23-50666-1
Department: BSc in CSE
- Name: EKRAMUL HASIB**
ID: 23-51106-1
Department: BSc in CSE
- Name: ABDUL ALIM**
ID: 23-51105-1
Department: BSc in CSE
- Name: MD. ARIFUL HASSAN**
ID: 22-49551-3
Department: BSc in CSE

A blue arrow points from the "Back Previous Frame" button in the top right corner to the left edge of the window.

FarmerHome

Welcome To AgriMarket-Pro

The screenshot shows a window titled "FarmerHome" with a header "Welcome To AgriMarket-Pro". It features a central callout box with the text "Please Select Your Product" and three categories:

- Vegetable** (with an image of various fruits and vegetables)
- Fish** (with an image of fish in a basket)
- Fruits** (with an image of various fruits)

Below each category is an orange box with descriptive text:

- Foe Sell Vegetable items.**
- For Sell Fish items.**
- For Sell Fruits items.**

A blue arrow points from the "Go for Log-in Frame" button in the top right corner to the left edge of the window.

FarmerVegetable

SEE PRODUCT LIST

Serial	Product Name	Product Quantity
*		

Total Amount: 0.00

Payment Request

Mushroom \$2.5/KG SELL
Papaya \$0.75/KG SELL
Pointed gourd \$0.75/KG SELL
Cabbage \$0.5/KG SELL
Bottle gourd \$0.65/KG SELL
Cucumber \$0.90/KG SELL
Radish \$0.5/KG SELL
Sweet Gourd \$0.75/KG SELL
Capsicum \$2.5KG SELL

Foe increases and decrease product quantity

Add product inside the table for sell.

Request send to Admin for payment.

Go for Log-in Frame

Go for Previous frame.

Table for show selected items.

FarmerFish

SEE PRODUCT LIST

Serial	Product Name	Product Quantity
*		

Total Amount: 0.00

Payment Request

Kural \$10/KG SELL
Prawn \$4.0/KG SELL
Pangasius \$1.9/KG SELL
Wallago \$4.5/KG SELL
Hilsa \$15/KG SELL
Mrigal carp \$4.50/KG SELL
St. Peter's \$3.5/KG SELL
Rui \$5.5/KG SELL
Tuna \$9KG SELL

Foe increases and decrease product quantity

Add product inside the table for sell.

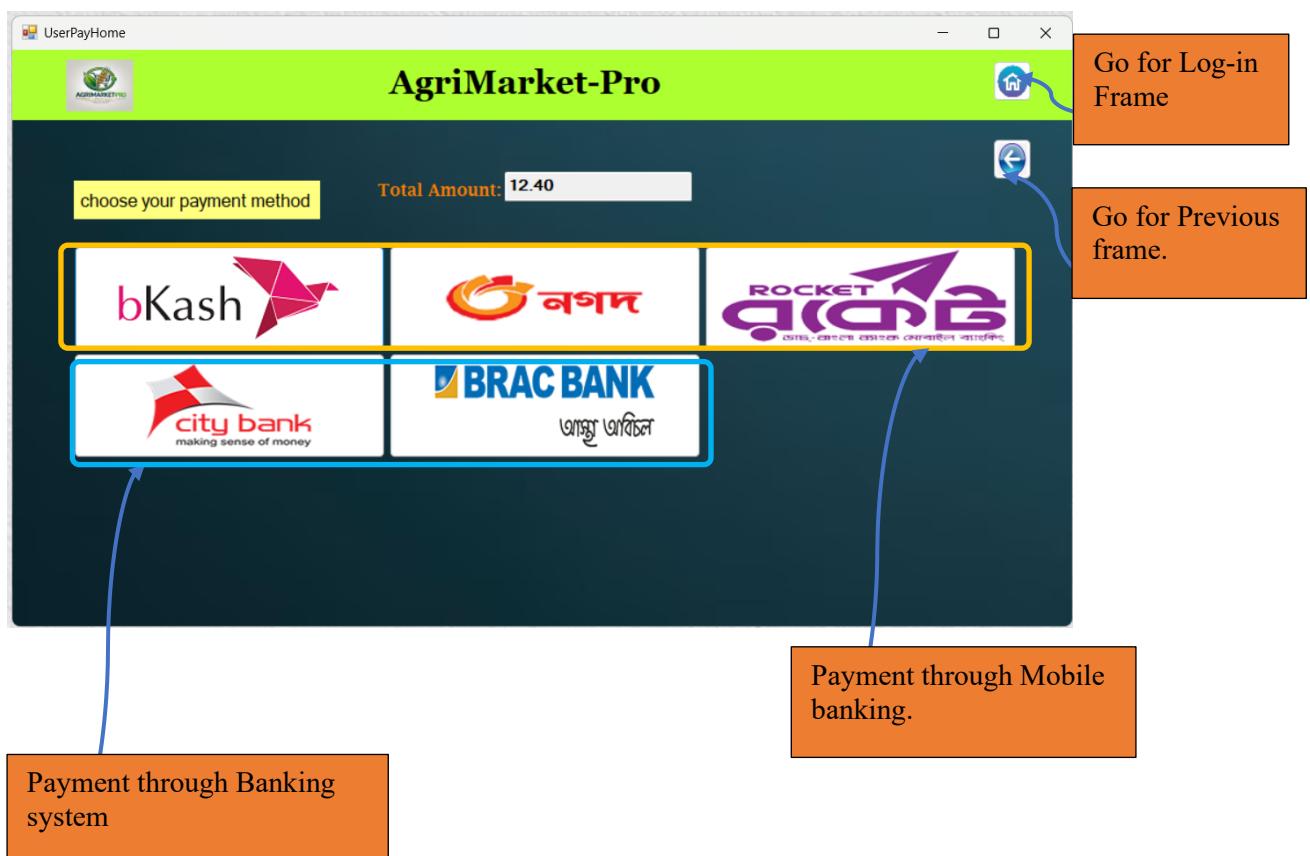
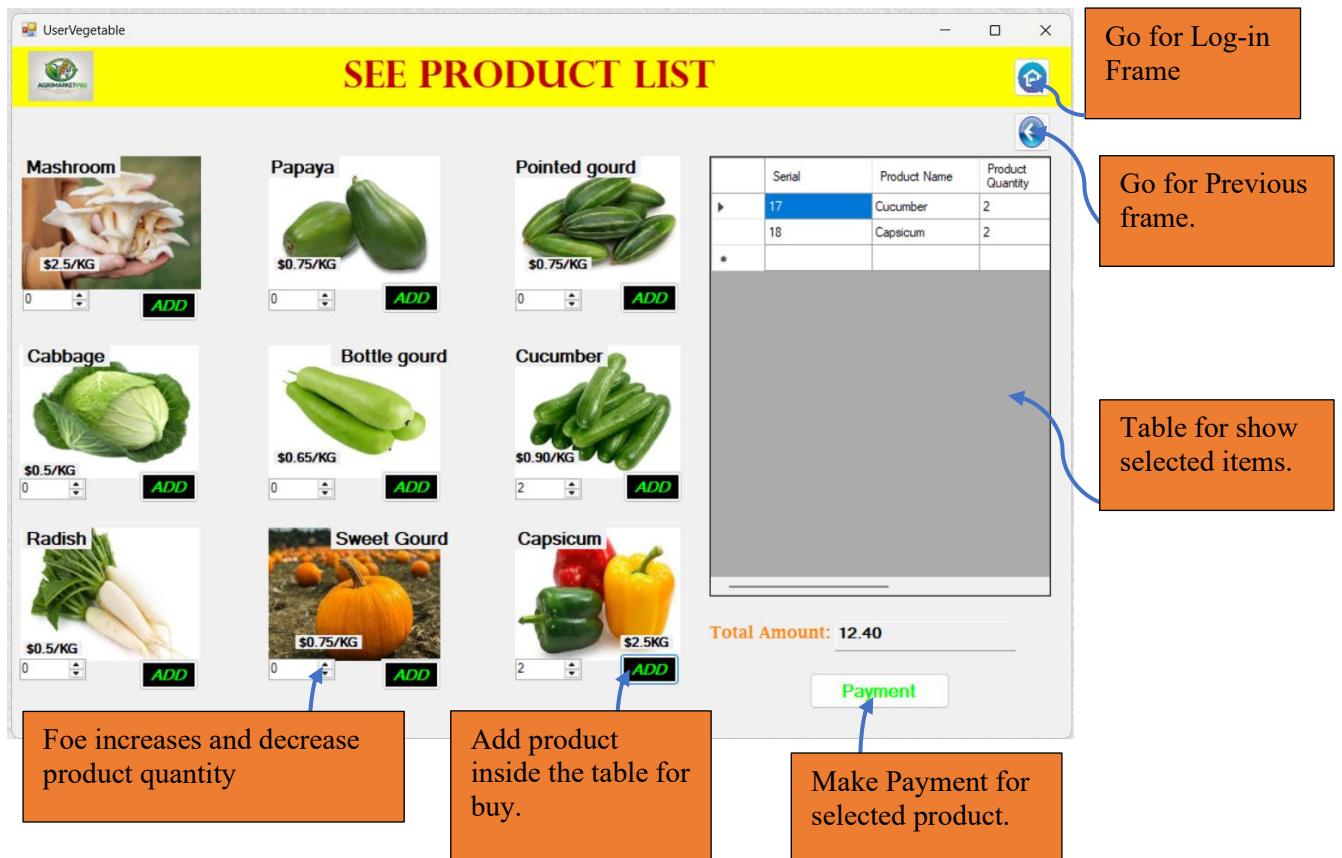
Request send to Admin for payment.

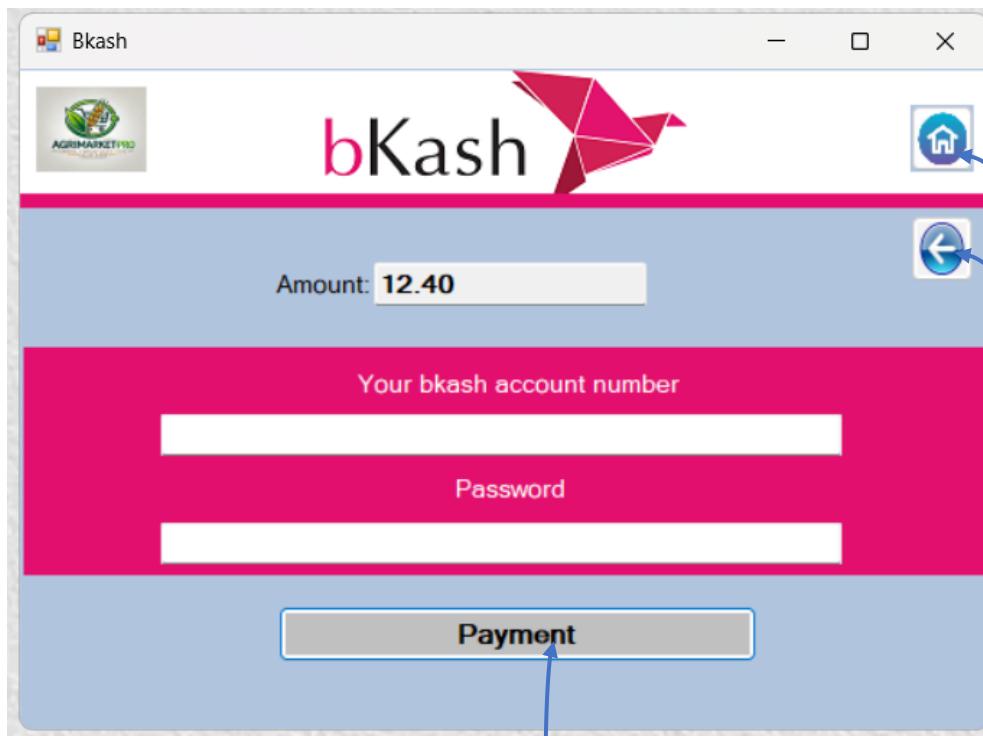
Go for Log-in Frame

Go for Previous frame.

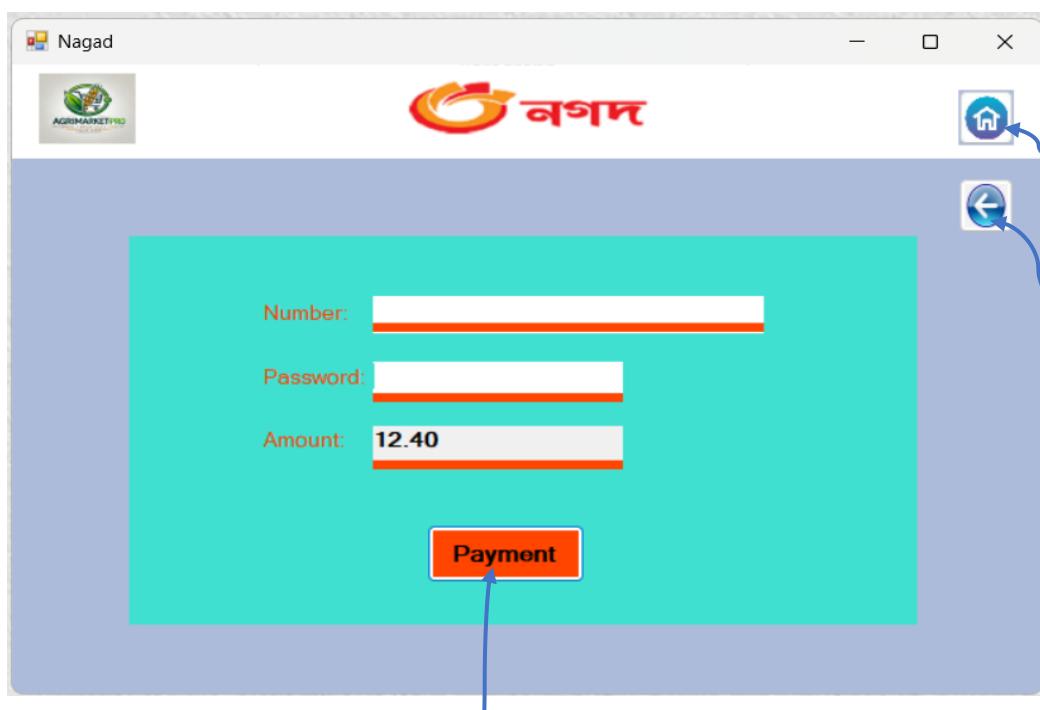
Table for show selected items.



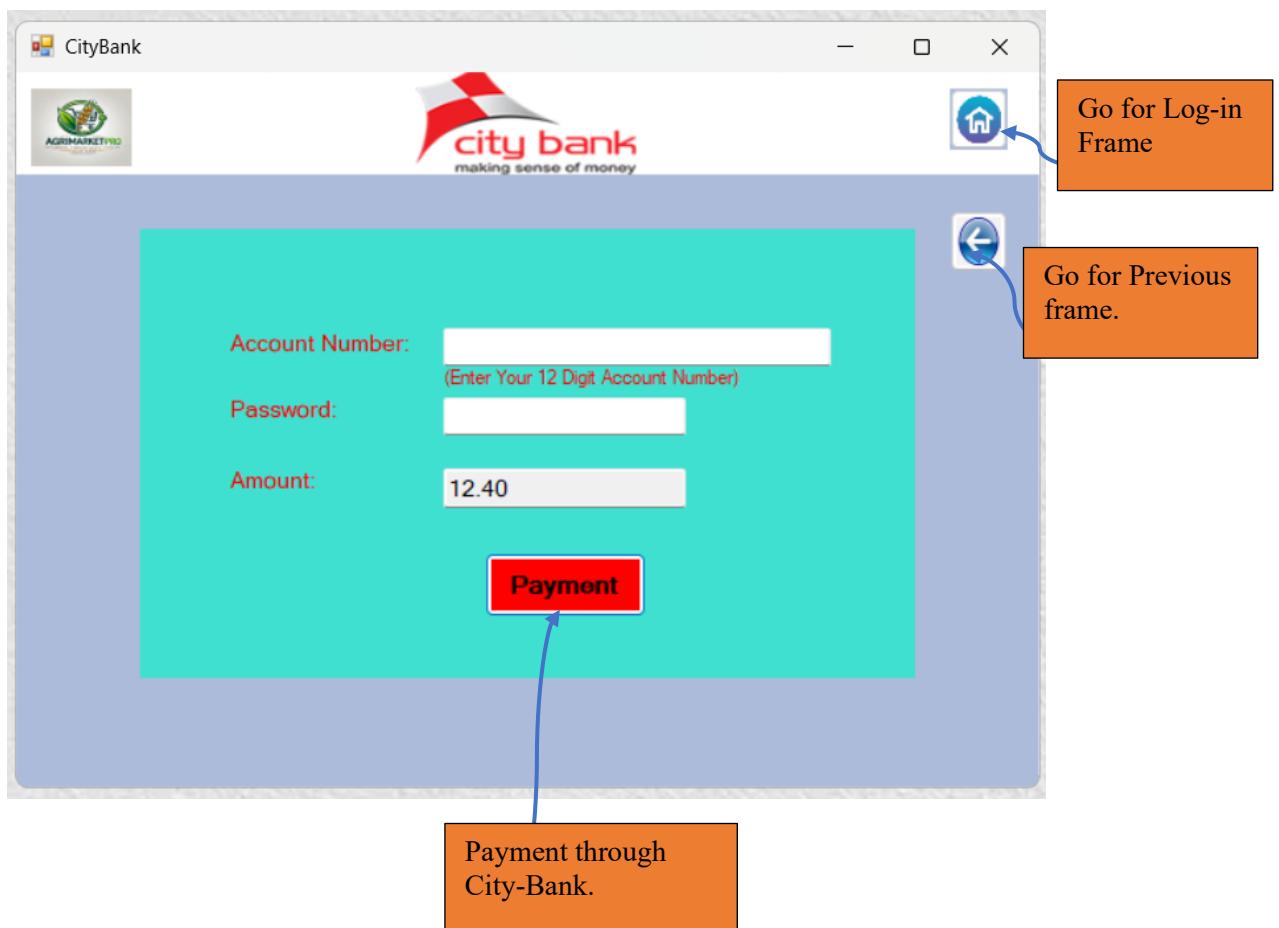
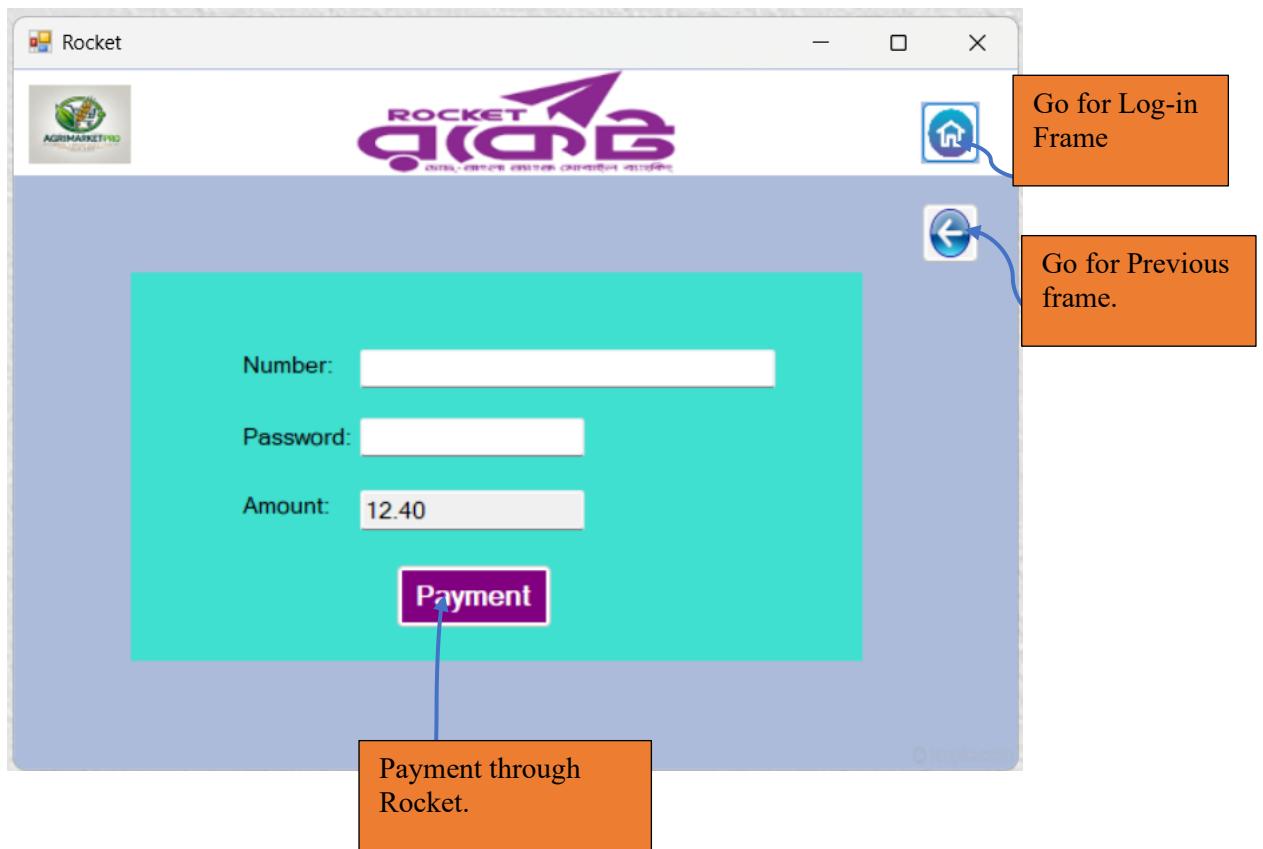


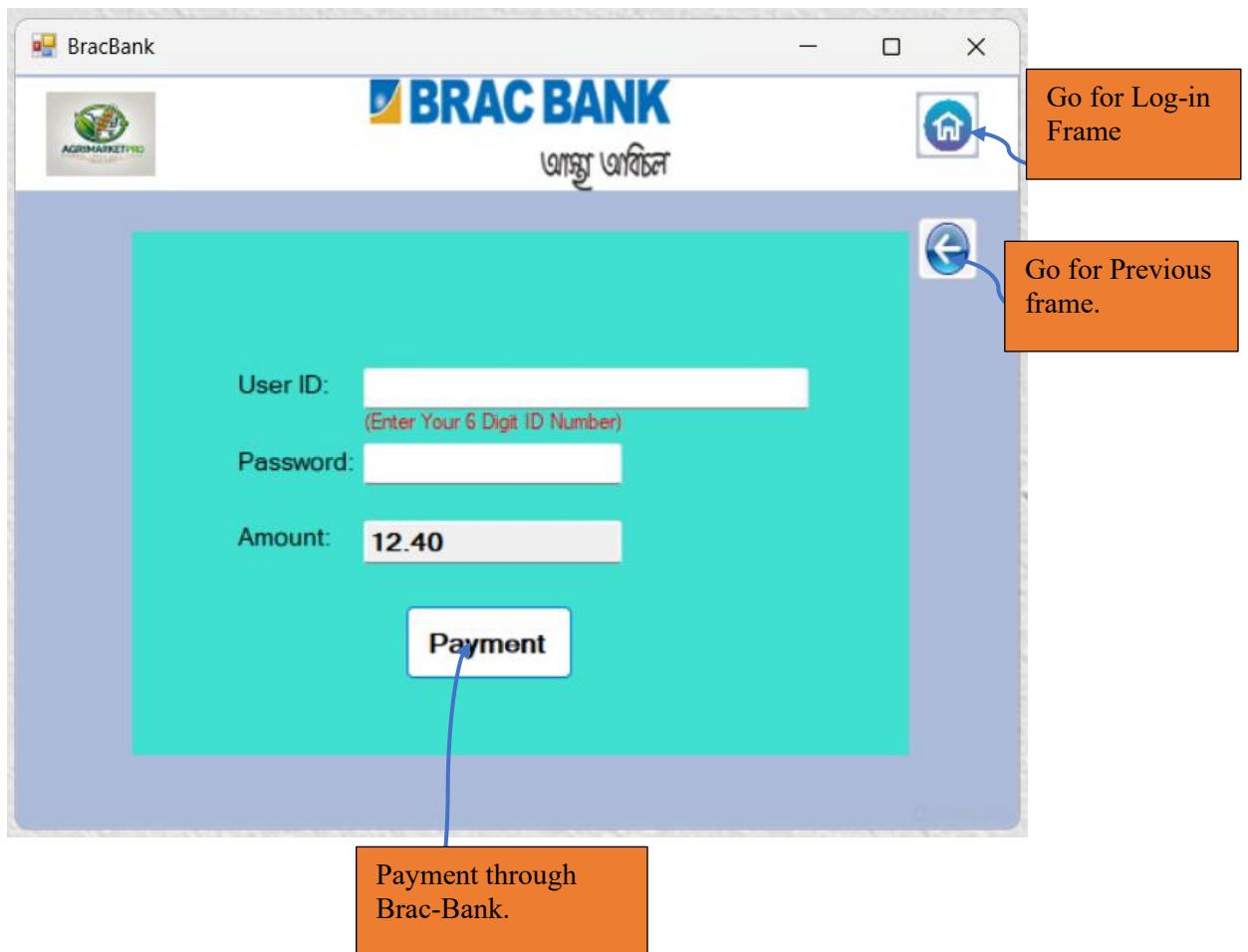


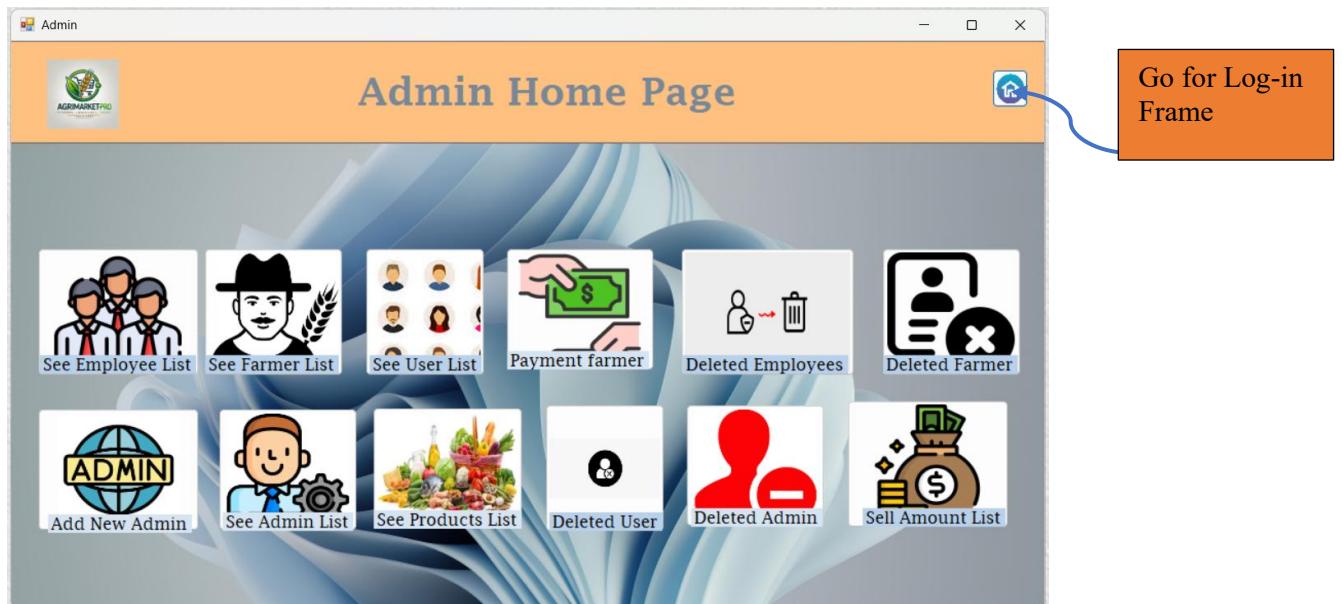
Payment through
Bkash.



Payment through
Nagad.







EmployeeList

Employee List

Employee Name:	
Salary:	
Phone Number:	
Gender:	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Add	Update
Delete	Reset

Search

Id	Employee Name	Salary	Phone Number	Gender
1	Showrav Ghosh	2000000	01910917566	Male
5	Test1	30000	01912345678	Male
8	Abir	890000	01910917565	Male
10	Ayan	20004	01910917567	Male
12	Abtahi	2000	01315477832	Male
14	Abed	50000	01910917552	Male
*				

Search employee by their name or phone number.

Update data from inside the table.

Go for Log-in Frame

Go previous page.

For add new Employee

For delete admin from list.

Remove all selected things.

DeletedEmployee

Deleted Employee List

Id	Employee Name	Salary	Phone Number	Gender
**				

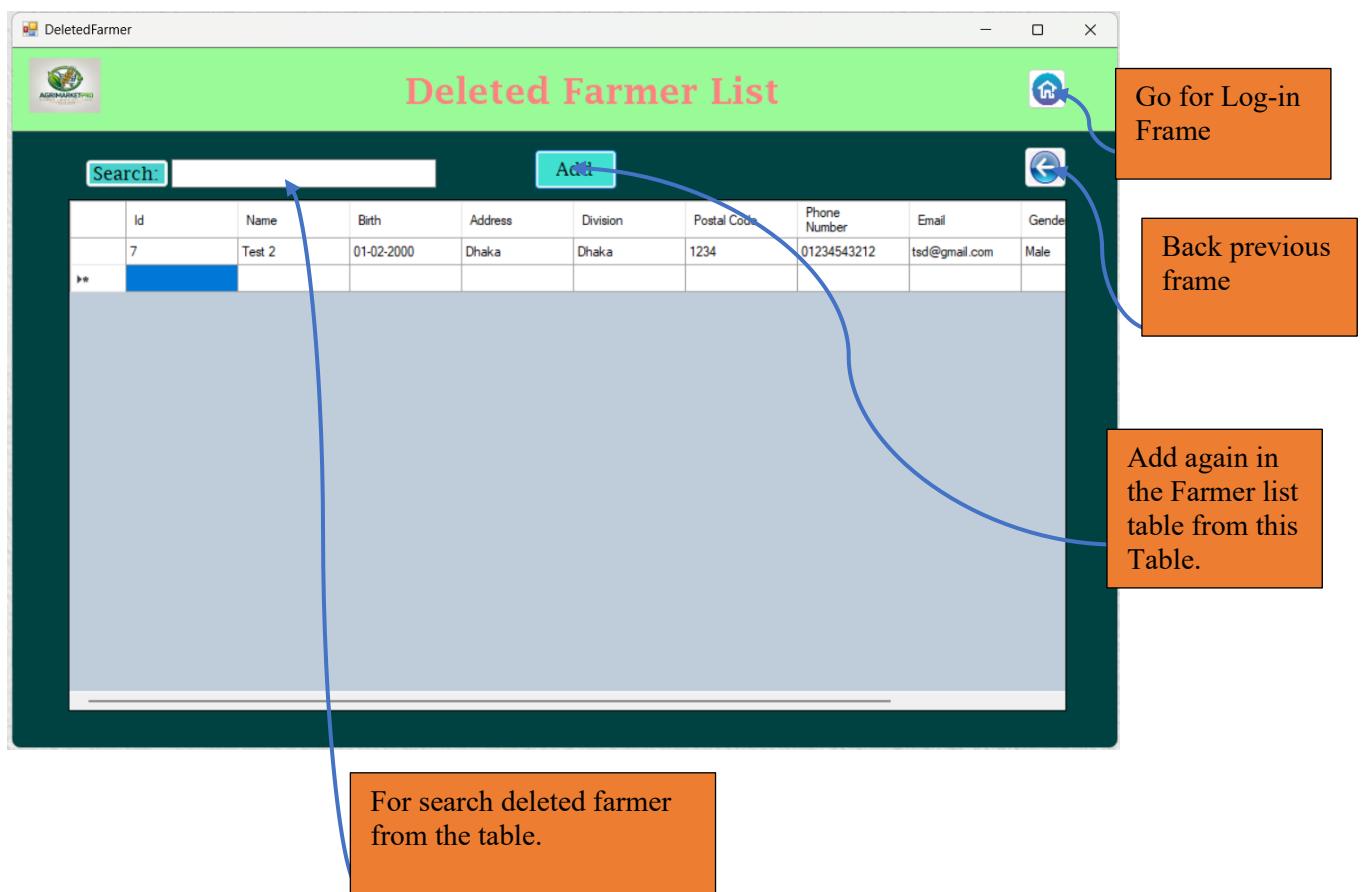
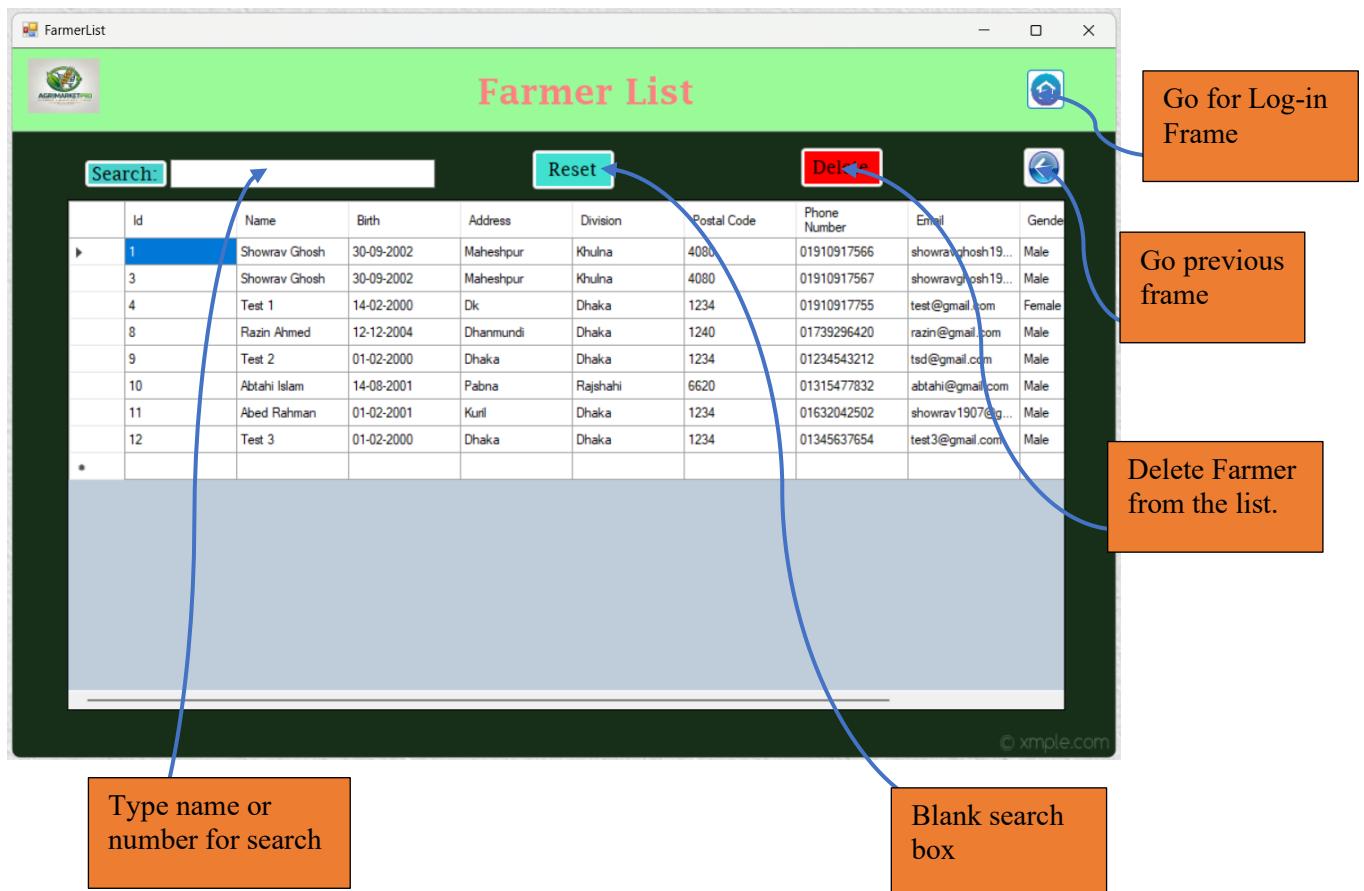
Search

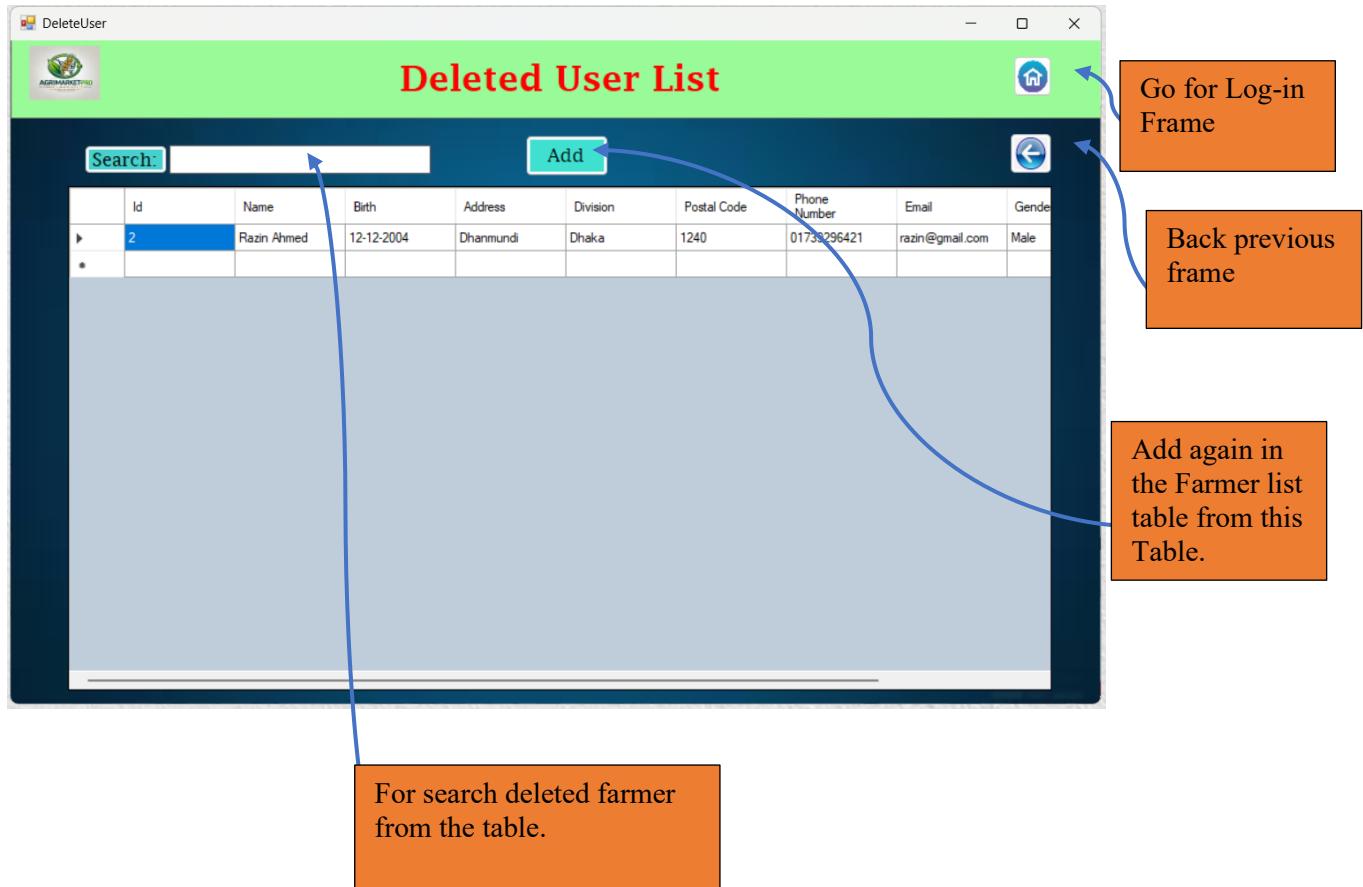
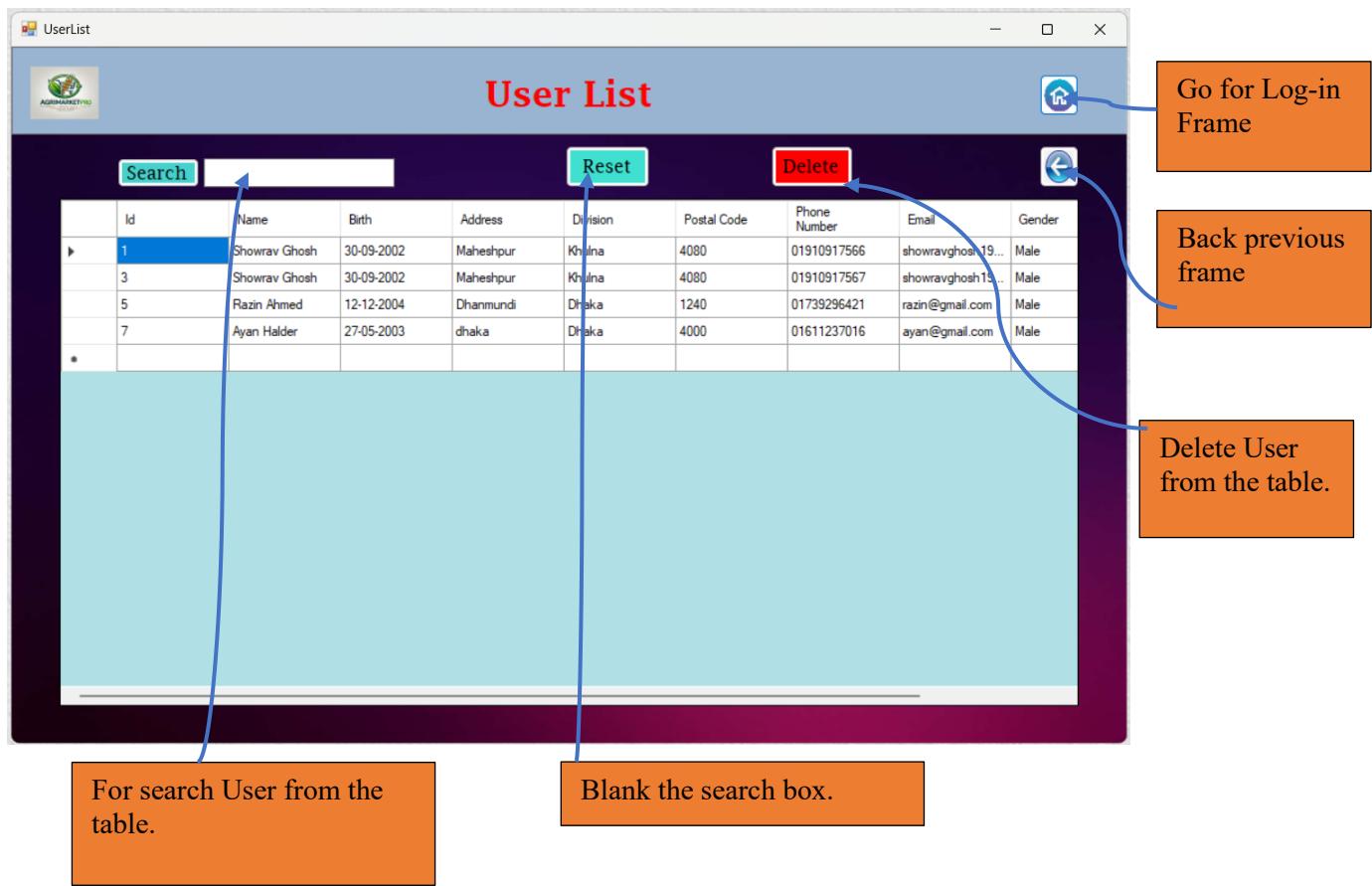
Go for Log-in Frame

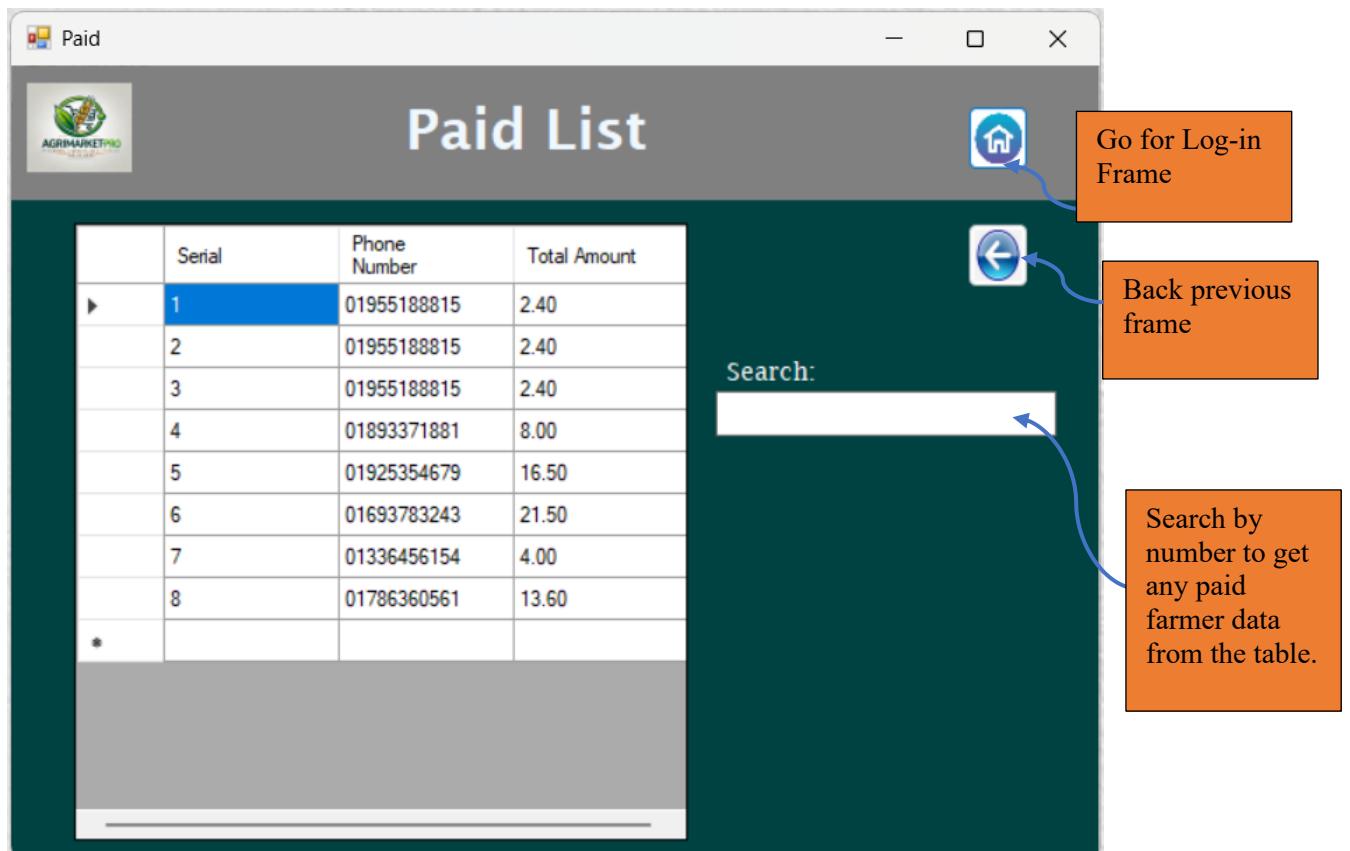
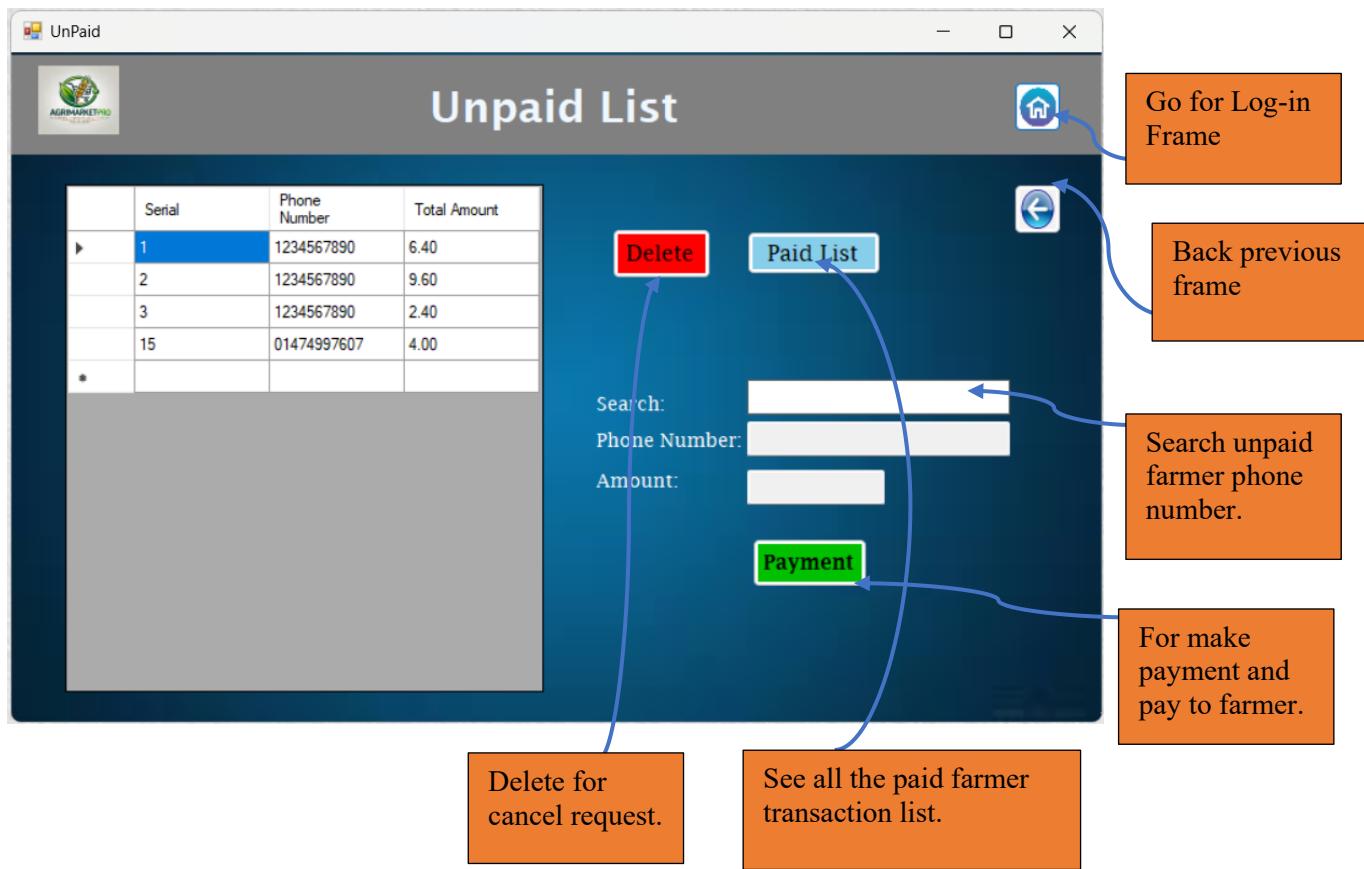
Go previous frame.

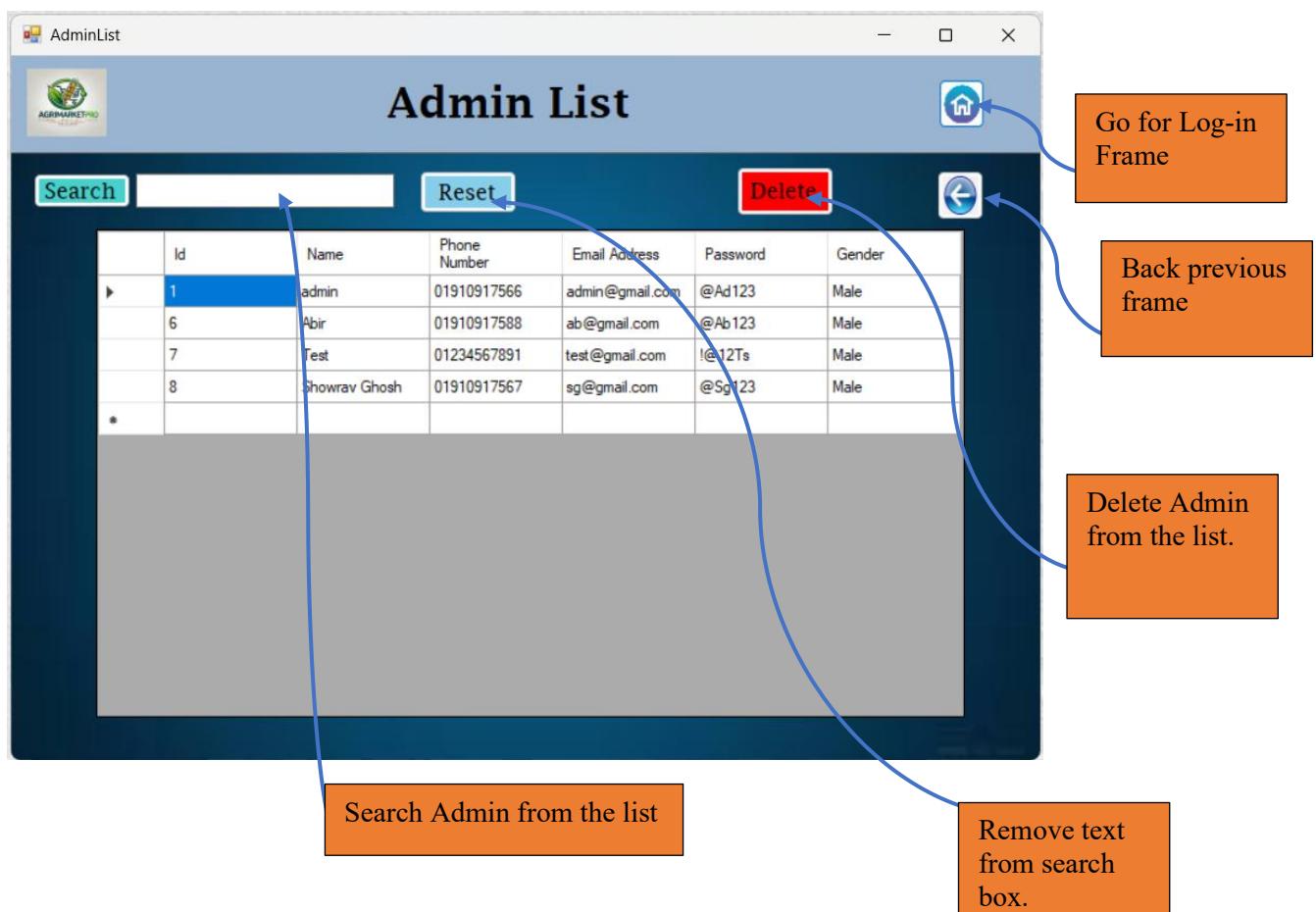
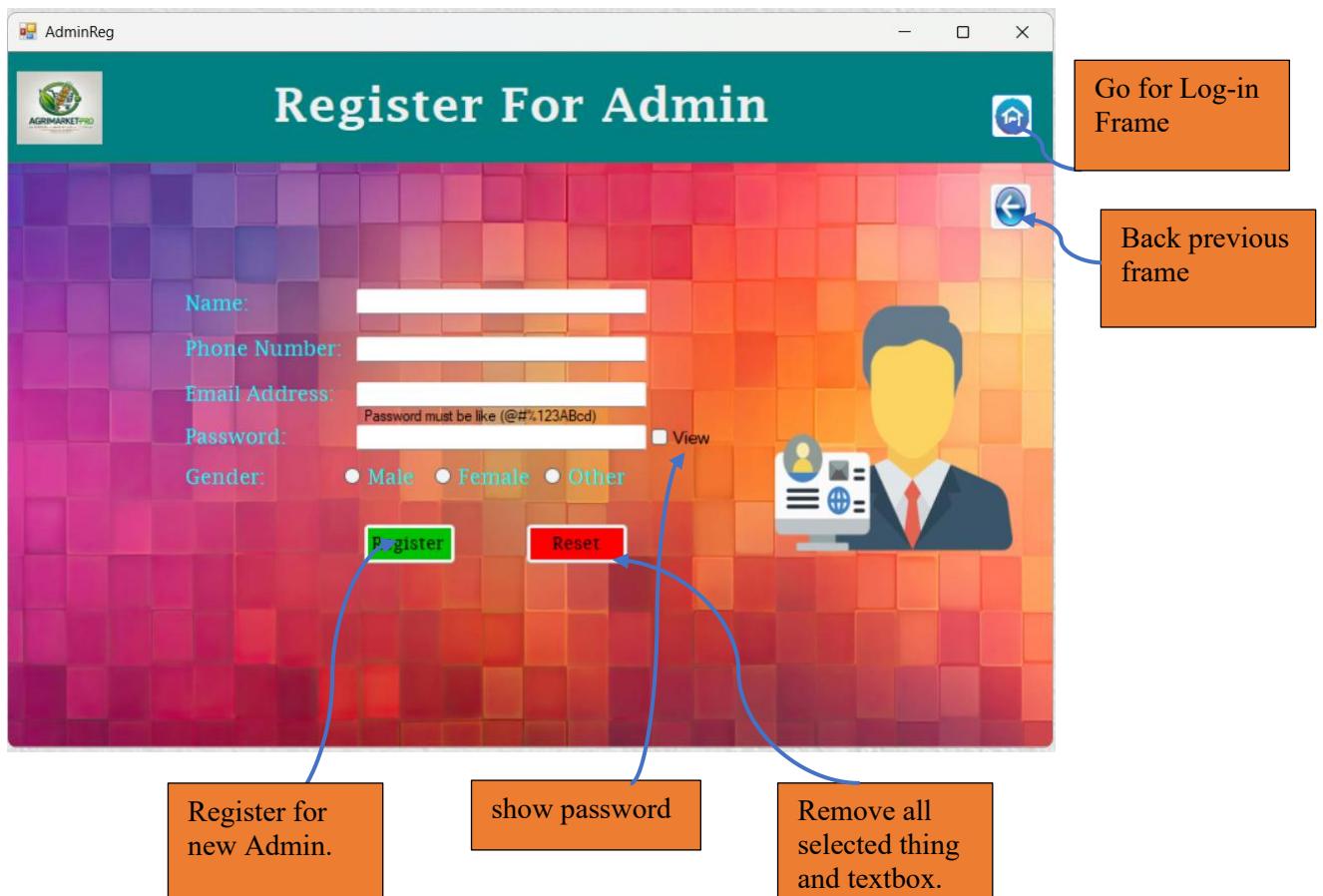
Search deleted employee from the table.

For Add again employee in the employee list table.









Deleted Admin List

	Id	Name	Phone Number	Email Address	Password	Gender
▶	2	Test	01234567891	test@gmail.com	!@12T	Male
◀						

For search deleted Admin from the table.

Go for Log-in Frame

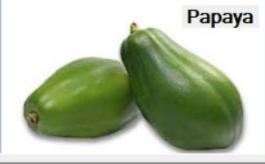
Back previous frame

Add again in the Admin list table from this Table.

Search

Add

ALL PRODUCTS LIST

Vegetables Item	Fruits Item	Fish Item
 Capsicum \$2.5/KG	 Banana \$.5/KG	 Pangasius \$1.9/KG
 Mashroom \$2.5/KG	 Strawberry \$4.0/KG	 Mrigal carp \$4.50/KG
 Papaya	 Watermelon	 Tuna

Go for Log-in Frame

Back previous frame

Up down for see more product from this list.

UserPaid

Sell Product Amount

Search:

	Serial	Ph_Num/U_ID/Acc	Amount	From
▶	1	01910917566	3.00	Bkash
	2	453456	4.40	BracBank
	3	01910915566	30.00	Bkash
	4	01987564544	30.00	Nagad
	5	01646745342	30.00	Rocket
	6	234562345678	30.00	CityBank
	7	453278	30.00	BracBank
	8	01783645431	23.00	Bkash
	9	775434	4.50	BracBank
*				

Go for Log-in Frame

Back previous frame

Search by number for see all the transaction that made by user from the table.