



# AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH

## (AIUB)

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

**CSC2210: OBJECT ORIENTED PROGRAMMING 2**

**Spring 2024-2025**

**Section: R**

**Group No: 03**

**Project Report On**

*Project Name: **Scholars' Nest***

**Supervised By**

**MD. KHAIRUL ALAM MAZUMDER**

**Submitted By:**

S.N.	Name	ID
01	NAYEM SIDDIKI	23-52361-2
02	FARHAN CHOWDHURY	23-52314-2
03	MD. SAJID ARAFAT	23-53526-3
04	HASAN SHAD SHAHRIAR	23-52529-2

<b>CO2:</b> 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				Total =
Requirement fulfillment	Properly demonstrate a real-life scenario-based project with proper functional requirement identification for the Object-Oriented Programming project development activities.				
Validation	Ensuring the ability of students' proper demonstration on validation forms in their system in terms of dealing with the data.				
Verification	Identifying if the students can verify the system data along with proper functional requirements in terms of data flow.				

## Scholars' Nest

*("Where Bright Minds Find Rest")*

### TABLE OF CONTENT

<b>Chapter No</b>	<b>Title</b>	<b>Page No</b>
Chapter 01	Introduction	3
Chapter 02	User Story	4
Chapter 03	ER Diagram	5
Chapter 04	SQL Queries	8
Chapter 05	Screenshots	14

## Chapter 01: Introduction

### Purpose:

In Bangladesh, many student hostels are still operating using manual systems. For example, in Dhaka University student halls, students typically go to the dining room and manually collect tokens to buy food. This often leads to problems like food shortages because the hall management prepares meals based on rough approximations rather than actual demand. Similarly, in bachelor messes across Bangladesh, meal updates are done using traditional "**Talli Khata**" (manual logs), which creates confusion and complicates management.

The purpose of this project is to develop a **Hostel Management System** that will automate meal updates and student hostel management in an easy, efficient, and organized way. This system will solve the current problems of food wastage, poor planning, and manual complexities.

### Objective:

The main objective of this system is to:

- Provide an easy way for students to book their meals in advance.
- Help hall/mess managers accurately estimate the food needed each day.
- Simplify student admission, payment, and meal management.
- Reduce manual errors and improve the efficiency of hostel operations.

### Scope:

The system will include the following features and user roles:

- **Students:**
  - Register and log in.
  - Submit admission forms and pay admission fees.
  - Pay monthly seat rent and utility bills.
  - Deposit money into a food wallet.
  - Book meals (breakfast, lunch, dinner) for the next day.
  - View current balances, meal charges, and payment validity.
  - Access food price lists and food breakdown in daily basis.
  - Can complain
- **Admin:**
  - Only log in.
  - Manage employees (Managers) by adding, updating, deleting, searching or resetting their accounts.
  - View the list of registered students.
  - View complains and delete it.
- **Employee (Manager):**
  - Only log in.
  - Admit, remove, search, update, or reset student accounts.
  - Update the meal list and prices.

## Tools Used:

- **Programming Language:** C#
- **For Database:** Microsoft SQL Server 2021 and SQL Server Management Studio
- **IDE:** Visual Studio 2022
- **Development Approach:** The system will follow Object-Oriented Programming (OOP) principles: **Encapsulation, Inheritance, Abstraction and Polymorphism etc.**

## Chapter 02: User Story

### Purpose:

This system is designed to solve real-life problems that students, admins, and managers face in student hostels and messes in Bangladesh. It focuses on making meal booking, payments, and hostel management simple, fast, and efficient.

### User Roles:

- **Student**
- **Admin**
- **Manager (Employee)**

### User Needs/Goals:

#### 1. Student:

- **Need:** A simple way to book meals without standing in line.
- **Scenario:** A student wants to have lunch tomorrow. He books the meal today using the system to ensure food will be prepared for him.
- **Need:** Track payment and meal balances easily.
- **Scenario:** A student checks his food wallet to see if he has enough balance to book the next day's dinner.
- **Need:** Make payments for food wallet deposits in a hassle-free way.
- **Scenario:** At the beginning of the month, the student deposits money into his food wallet through the system.
- **Need:** Can complain about services.
- **Scenario:** If student faces any problem or if he finds anything wrong he can complain it to the authority.

#### 2. Admin:

- **Need:** Manage hostel employees easily.
- **Scenario:** The admin wants to add a new manager to handle student and meal operations. He can quickly create or update manager accounts through the system.
- **Need:** View all student records.
- **Scenario:** The admin checks the full list of students to verify their admission, payment, and activity status.
- **Need:** View complains.
- **Scenario:** When any of the student complains about anything the admin can check the complaint list and can solve it or ignore it.

### 3. Manager (Employee):

- **Need:** Admit new students or update their records smoothly.
- **Scenario:** A new student joins the hostel. The manager can quickly create his account and set up his details in the system.
- **Need:** Update daily meal lists and prices based on current situations.
- **Scenario:** The price of lunch increases. The manager can immediately update the price in the system, so all students see the new rate.

## Chapter 03: ER Diagram and other diagrams

**Purpose:** This chapter presents the database structure, showing the relationships between different entities (tables) using the Entity Relationship (ER) Diagram. It also includes system flow diagrams such as the Use Case Diagram, Activity Diagram, and Sequence Diagram to illustrate the user interactions and system processes clearly.

### Entity Relationship (ER) Diagram

#### 1. Student

- Attributes: name, fatherName, email (PK), phone, institution, DOB, gender, address, password

#### 2. Admin

- Attributes: adminId (PK), name, salary, password

#### 3. Employee

- Attributes: employeeId (PK), name, salary, password, role, adminId (FK)

#### 4. Meal

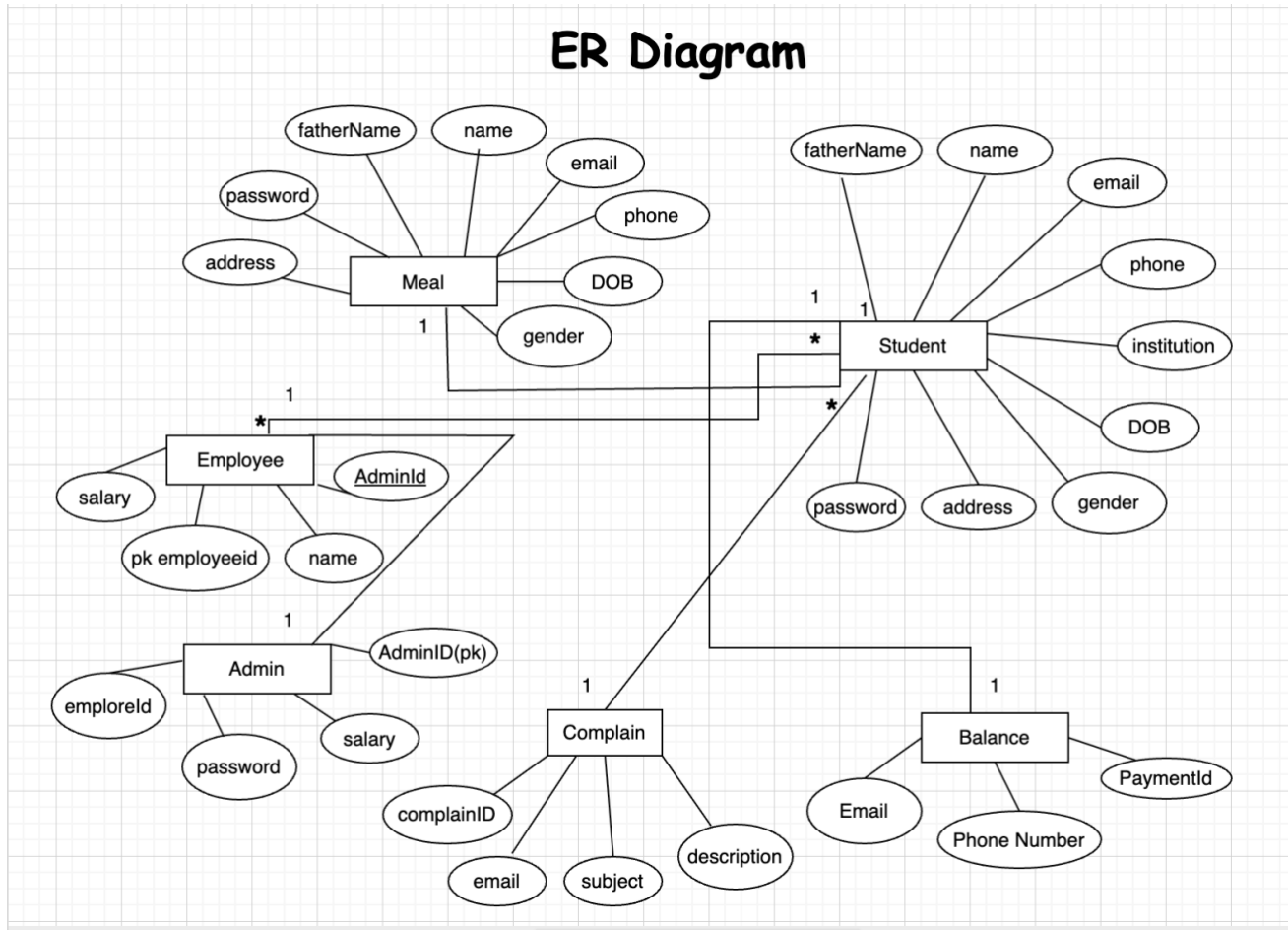
- Attributes: mealId (PK), day, breakfast, breakfast\_price, lunch, lunch\_price, dinner, dinner\_price

#### 5. Balance

- Attributes: paymentID (PK), email (FK), phone, balance

#### 6. Complain

- Attributes: complainId (PK), email (FK), subject, description



## Use Case Diagram:

Purpose:

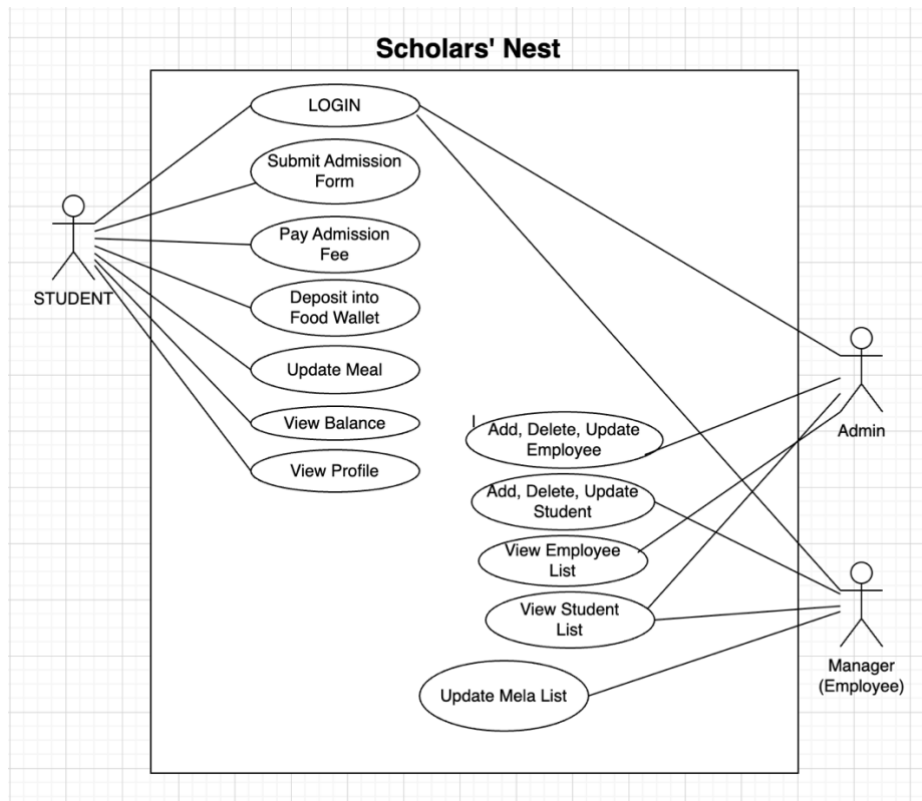
Shows the interactions between users (Actors) and the system functionalities.

Key Actors:

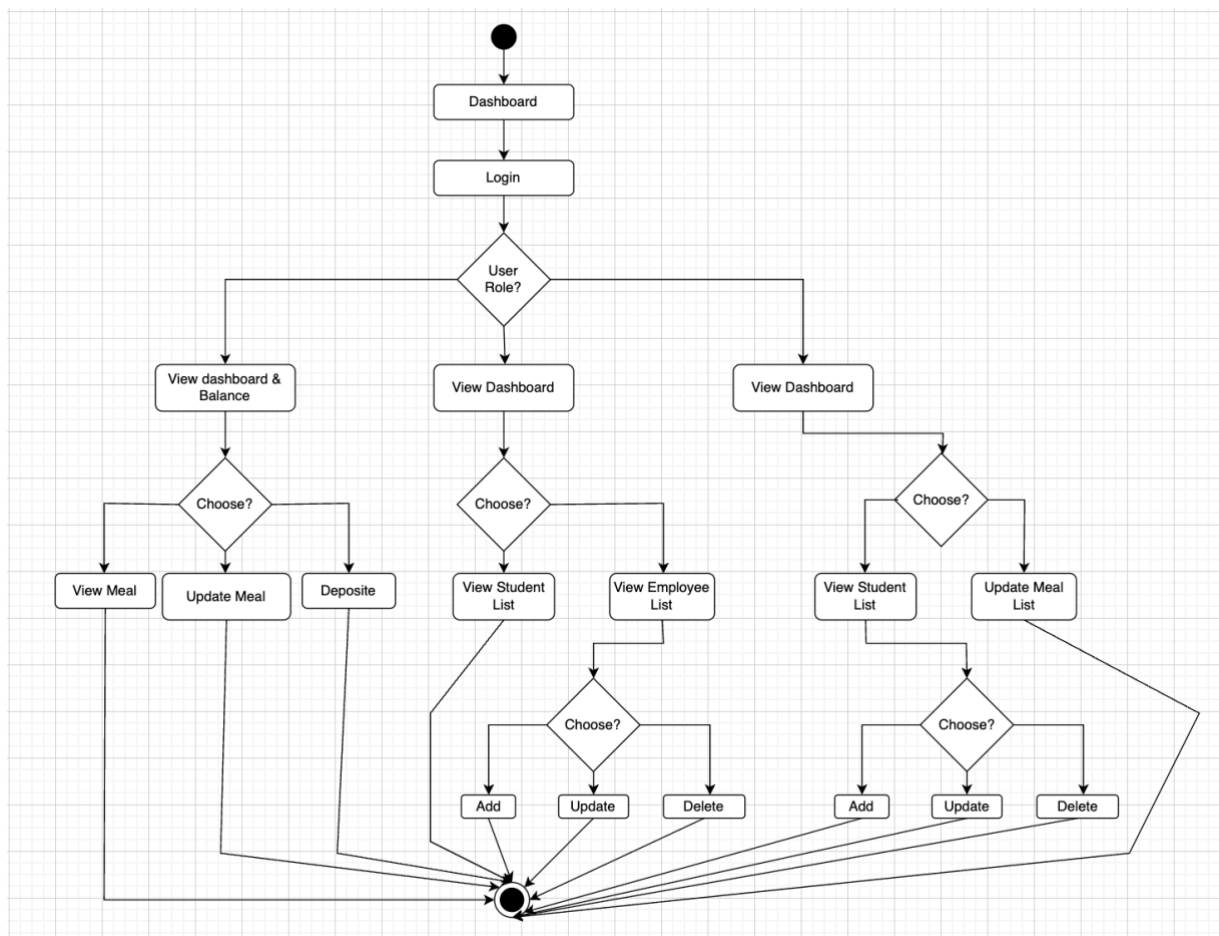
- **Student**
- **Admin**
- **Manager (Employee)**

Main Use Cases:

- Student: Login, Admission Form Fill Up, Update Balance, Book Meal, Check Balance.
- Admin: Manage Employees, View Student Info.
- Manager: View and Manage Student info, Update Meal List and Price.

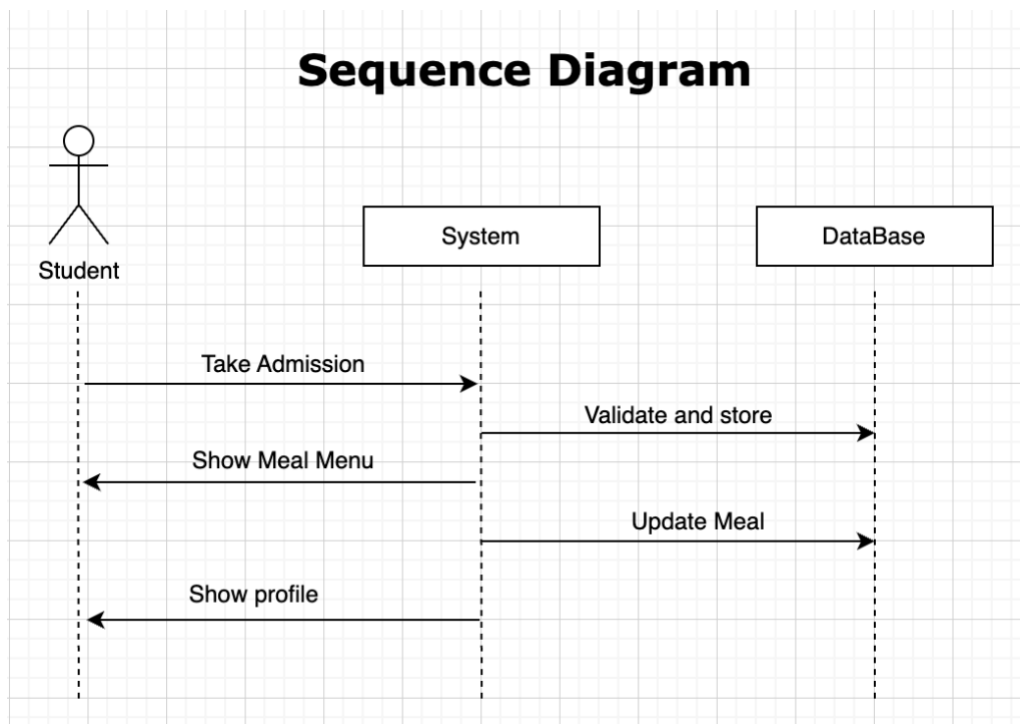


## Activity Diagram:





## Sequence Diagram:



## Chapter 04: SQL Queries

### Purpose:

This chapter demonstrates how the **ScholarsNest** system interacts with the database using **SQL queries**. It includes:

- Table Creation Queries
- Insert Queries
- Select Queries
- Update/Delete Queries
- Search Queries

### Database:

```
USE ScholarsNest;
```

### Table Creation Queries

#### 1. Student Table

```
CREATE TABLE Student (
    name VARCHAR(100),
    fatherName VARCHAR(100),
    email VARCHAR(100) PRIMARY KEY,
    phone VARCHAR(20),
    institution VARCHAR(100),
```

```

        DOB VARCHAR(20),
        gender VARCHAR(10),
        address VARCHAR(255),
        password VARCHAR(100)
    );

```

## 2. Admin Table

```

CREATE TABLE Admin (
    adminId VARCHAR(10) PRIMARY KEY,
    name VARCHAR(100),
    salary DECIMAL(10, 2),
    password VARCHAR(100)
);

```

## 3. Employee Table

```

CREATE TABLE Employee (
    employeeId VARCHAR(10) PRIMARY KEY,
    name VARCHAR(100),
    salary VARCHAR(10),
    password VARCHAR(100),
    role VARCHAR(50)
);

```

## 4. Meal Table

```

CREATE TABLE Meal (
    mealId VARCHAR(10) PRIMARY KEY,
    day VARCHAR(20),
    breakfast VARCHAR(100),
    breakfast_price VARCHAR(10),
    lunch VARCHAR(100),
    lunch_price VARCHAR(10),
    dinner VARCHAR(100),
    dinner_price VARCHAR(10)
);

```

## 5. Complaint Table

```

CREATE TABLE Complaint (
    complainId INT IDENTITY(200,1) PRIMARY KEY,
    studentemail VARCHAR(100),
    subject VARCHAR(255),
    description VARCHAR(500),
    FOREIGN KEY (studentemail) REFERENCES Student(email)
);

```

## 6. AcceptedStudent Table

```
CREATE TABLE AcceptedStudent (
    serial INT IDENTITY(1,1) PRIMARY KEY,
    id AS ('23-' + RIGHT('0000' + CAST(serial AS VARCHAR(4)), 4))
PERSISTED,
    name VARCHAR(100),
    fatherName VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(20),
    institution VARCHAR(100),
    DOB VARCHAR(20),
    gender VARCHAR(10),
    address VARCHAR(255),
    password VARCHAR(100)
);
```

## 7. RequestedStudent Table

```
CREATE TABLE RequestedStudent (
    name VARCHAR(100),
    fatherName VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    institution VARCHAR(100),
    DOB VARCHAR(20),
    gender VARCHAR(10),
    address VARCHAR(255),
    password VARCHAR(100)
);
```

## 8. StudentMeal Table

```
CREATE TABLE StudentMeal (
    studentEmail VARCHAR(100),
    mealDate VARCHAR(50),
    bprice VARCHAR(10),
    lprice VARCHAR(10),
    nprice VARCHAR(10),
    total VARCHAR(10),
    CONSTRAINT FK_StudentEmail FOREIGN KEY (studentEmail)
        REFERENCES Student(email)
);
```

## 9. Payment Table

```
CREATE TABLE Payment (
    serial INT IDENTITY(1,1) PRIMARY KEY,
    paymentId AS ('AX-' + RIGHT('00' + CAST(serial AS VARCHAR(2)),
2)) PERSISTED,
    studentEmail VARCHAR(100),
    paymentDate VARCHAR(30),
    amount VARCHAR(30),
```

```

    phone VARCHAR(30),
    paymentMethod VARCHAR(50),
    CONSTRAINT FK_Payment_StudentEmail FOREIGN KEY (studentEmail)
        REFERENCES Student(email)
);

```

## Insert Queries

### 1. Insert Admin Data

```

INSERT INTO Admin (adminId, name, salary, password)
VALUES
('F-1', 'Farhan', 75000.00, 'farhan@123'),
('S-1', 'Sajid', 72000.00, 'sajid@123'),
('N-1', 'Nayeem', 73000.00, 'nayeem@123');

```

### 2. Insert Employee Data

```

INSERT INTO Employee (employeeId, name, salary, password, role)
VALUES
('E-101', 'Alice Karim', '35000.00', 'alice123', 'Manager'),
('E-102', 'Tanvir Rahman', '28000.00', 'tanvir123', 'Staff'),
('E-201', 'Nusrat Jahan', '32000.00', 'nusrat321', 'HR'),
('E-301', 'Sajid Khan', '30000.00', 'sajid789', 'Assistant'),
('E-302', 'Rumi Das', '26000.00', 'rumi456', 'Staff');

```

### 3. Insert Student Data (Dynamic)

```

INSERT INTO Student VALUES (@name, @fatherName, @email, @phone,
@institution, @dob, @gender, @address, @password);

```

### 4. Insert Meal Data

```

INSERT INTO Meal (mealId, day, breakfast, breakfast_price, lunch,
lunch_price, dinner, dinner_price)
VALUES
('M-1', 'Monday', 'Paratha & Egg', '25.00', 'Rice & Chicken Curry',
'60.00', 'Roti & Daal', '30.00'),
('M-2', 'Tuesday', 'Bread & Jam', '20.00', 'Rice & Fish Curry',
'55.00', 'Khichuri & Egg', '35.00'),
('M-3', 'Wednesday', 'Suji & Banana', '18.00', 'Polao & Chicken
Roast', '70.00', 'Roti & Mixed Veg', '28.00'),
('M-4', 'Thursday', 'Chotpoti', '22.00', 'Plain Rice & Egg Curry',
'50.00', 'Noodles & Sausage', '40.00'),
('M-5', 'Friday', 'Halwa & Puri', '30.00', 'Beef Tehari', '80.00',
'Roti & Lentils', '25.00'),
('M-6', 'Saturday', 'Panta & Fried Hilsha', '35.00', 'Vegetable
Biriyani', '65.00', 'Roti & Egg Curry', '30.00'),
('M-7', 'Sunday', 'Milk Bread & Banana', '20.00', 'Chicken
Biriyani', '75.00', 'Khichuri & Chicken', '40.00');

```

## Select Queries

### 1. View All Students

```
SELECT * FROM Student;
```

### 2. View All Admins

```
SELECT * FROM Admin;
```

### 3. View All Employees

```
SELECT * FROM Employee;
```

### 4. View All Meals

```
SELECT * FROM Meal;
```

### 5. View All Complaints

```
SELECT * FROM Complaint;
```

### 6. View All Accepted Students

```
SELECT * FROM AcceptedStudent;
```

### 7. View All Requested Students

```
SELECT * FROM RequestedStudent;
```

### 8. View All Student Meals

```
SELECT * FROM StudentMeal;
```

### 9. View All Payments

```
SELECT * FROM Payment;
```

## Search Queries

### 1. Search Student by Institution

```
SELECT name, email, password FROM Student WHERE institution =  
'aiub';
```

### 2. Search Monday's Breakfast Menu

```
SELECT breakfast, breakfast_price FROM Meal WHERE day = 'Monday';
```

### 3. Search Accepted Student by Email

```
SELECT id, password FROM AcceptedStudent WHERE email =  
'taimoor.aslam@example.com';
```

## Update Queries

### 1. Update Full Student Record

```
UPDATE Student  
SET name = @name, fatherName = @fatherName, phone = @phone,  
institution = @institution, DOB = @dob, gender = @gender, address =  
@address, password = @password  
WHERE email = @email;
```

### 2. Update Student Password Only

```
UPDATE Student  
SET password = @password  
WHERE email = @email;
```

## Delete Queries

### 1. Delete a Student by Email

```
DELETE FROM Student  
WHERE email = 'clara.lee@example.com';
```

### 2. Delete Accepted Student by Name

```
DELETE FROM AcceptedStudent  
WHERE name = 'test';
```

## Chapter 05: Screenshots

**Purpose:** Demonstrate your working application using screenshots.

**Screenshots:**

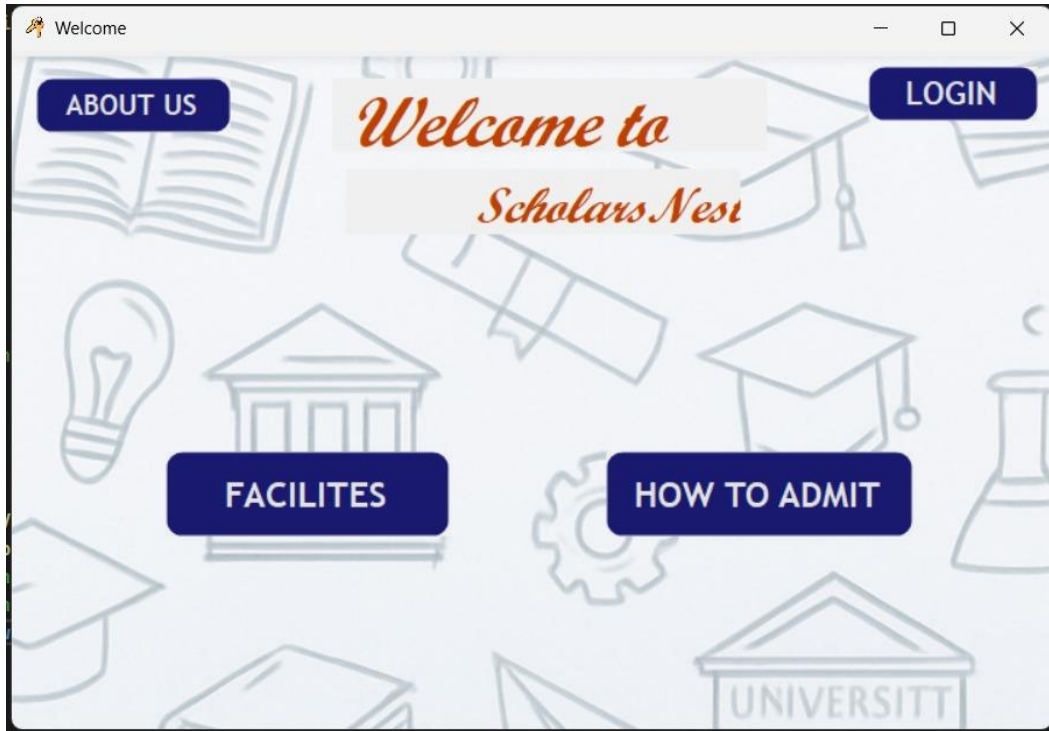


Figure 01: Welcome Page

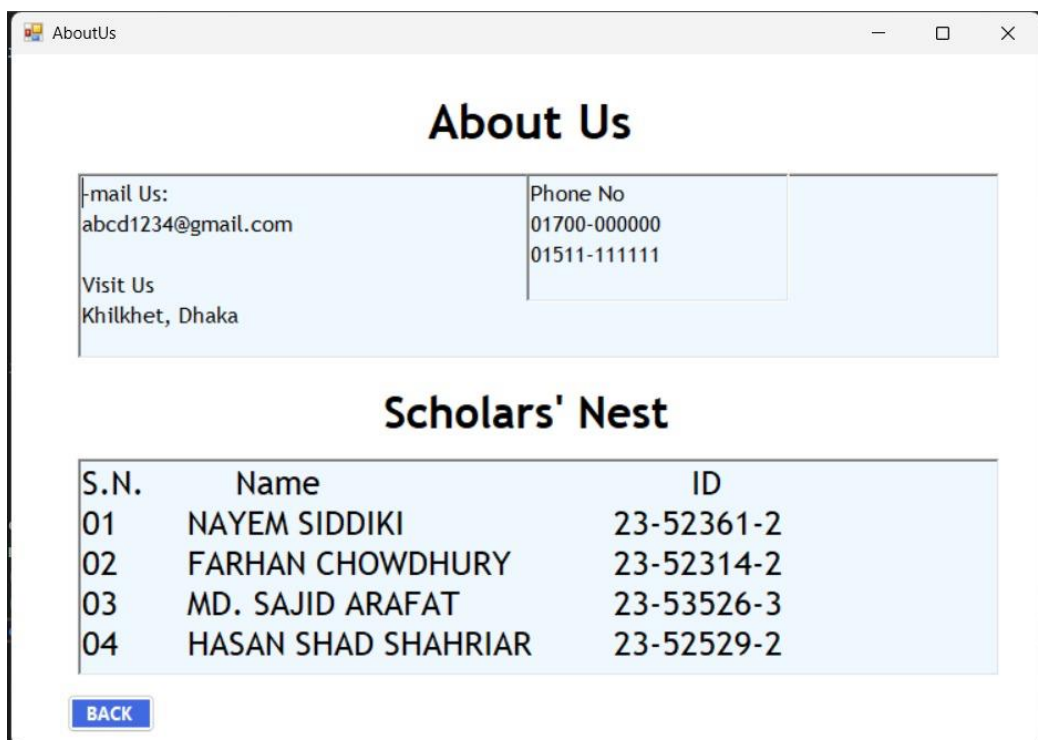


Figure 02: About Us Page

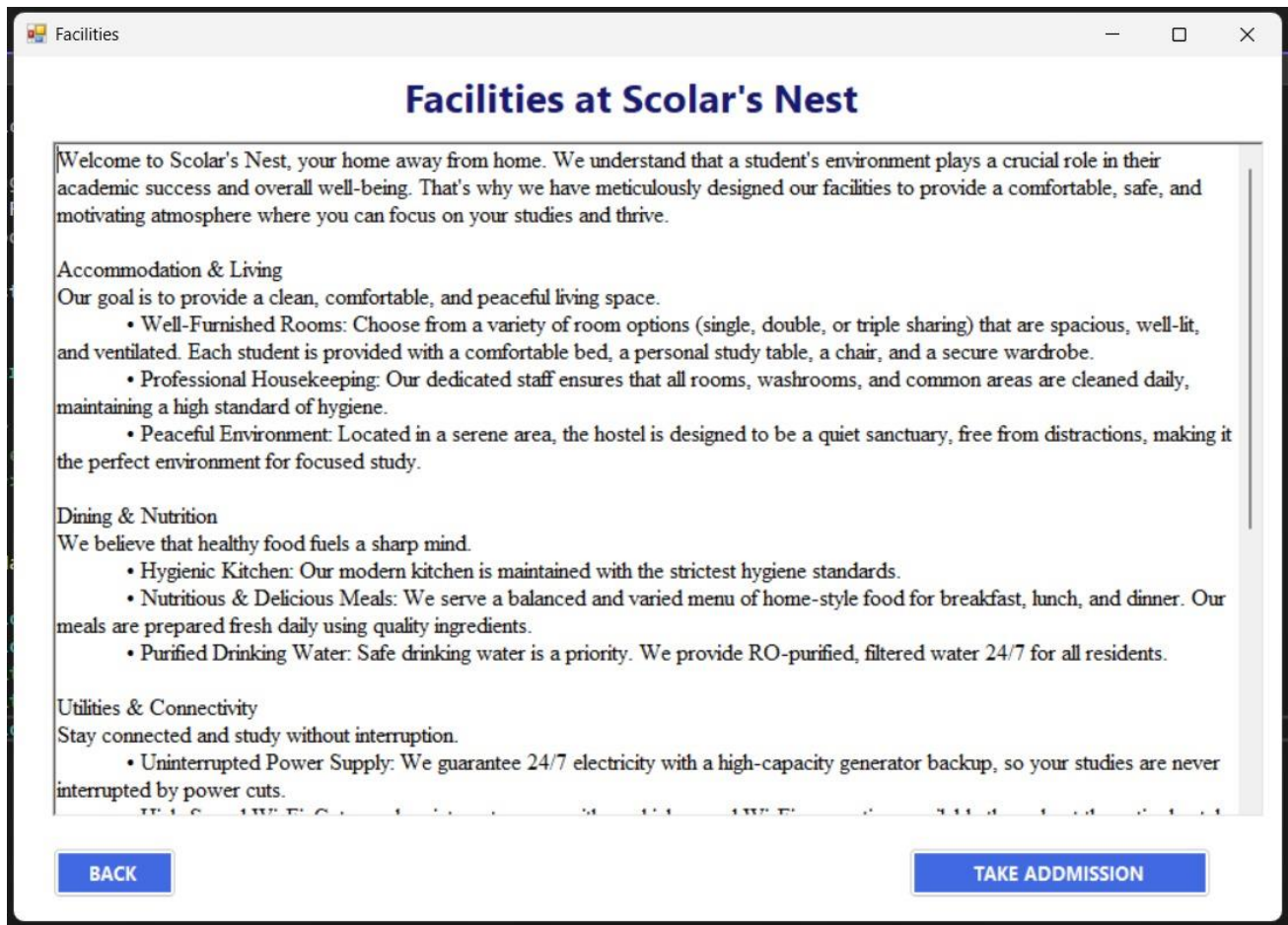


Figure 03: Facilities Page

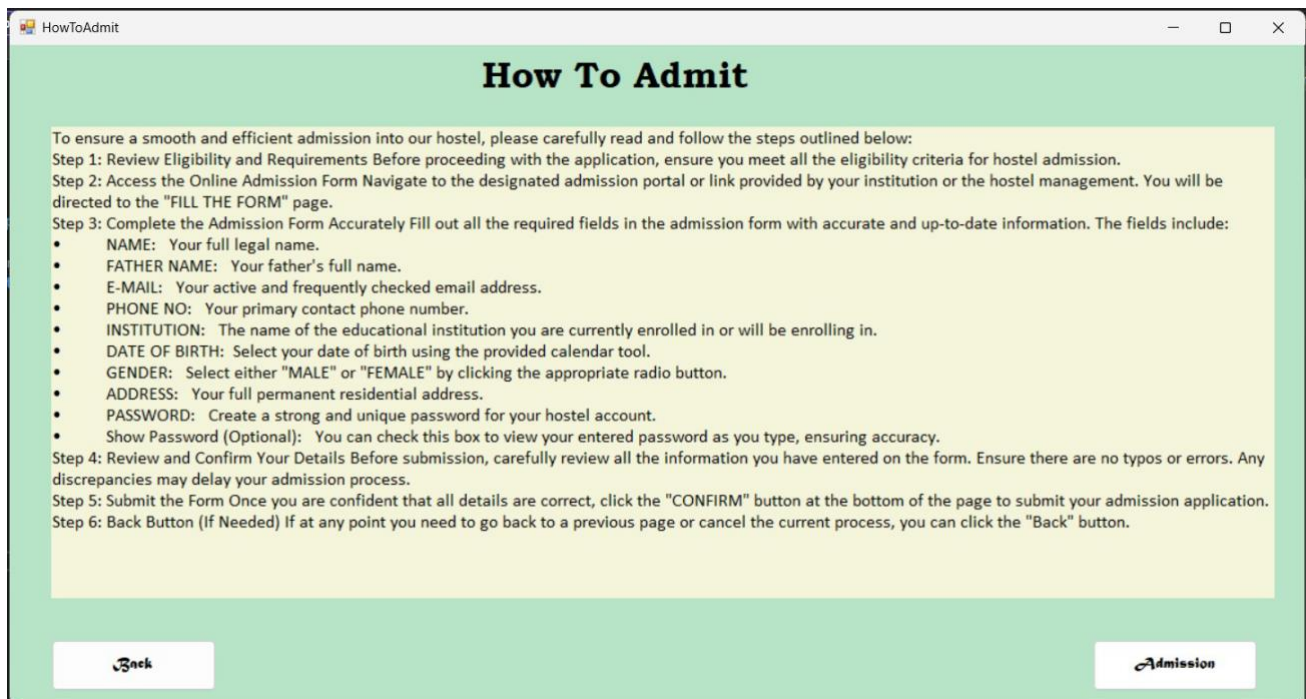


Figure 04: How To Admit Page



**Admission**

### FILL THE FORM

**NAME**

**FATHER NAME**

**E-MAIL**

**PHONE NO**

**INSTITUTION**

**ADDRESS**

**PASSWORD**

☐ Show Password Minimum 8 characters

**DATE OF BIRTH**

**GENDER** ☐ MALE ☐ FEMALE

**BACK** **CONFIRM**

Figure 05: Admission Form Page

**Login**

**Scholars' Nest**

*"Where Bright Minds Find Rest"*

### Log In

**E-mail/User ID**

**Password**

[Forgot Password?](#) ☐ Show Password

**HOME** **LOG IN**

Find Us

Figure 06: Log In Page

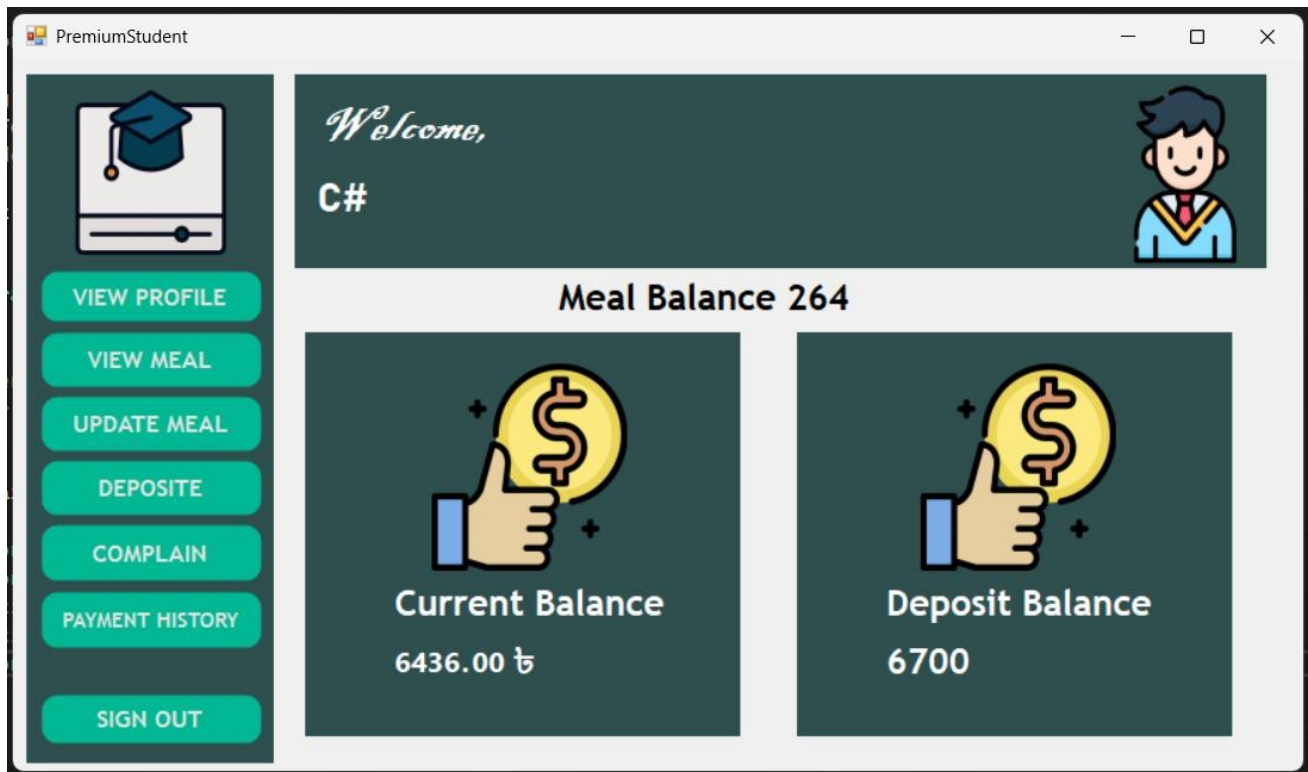


Figure 07: Student Dashboard

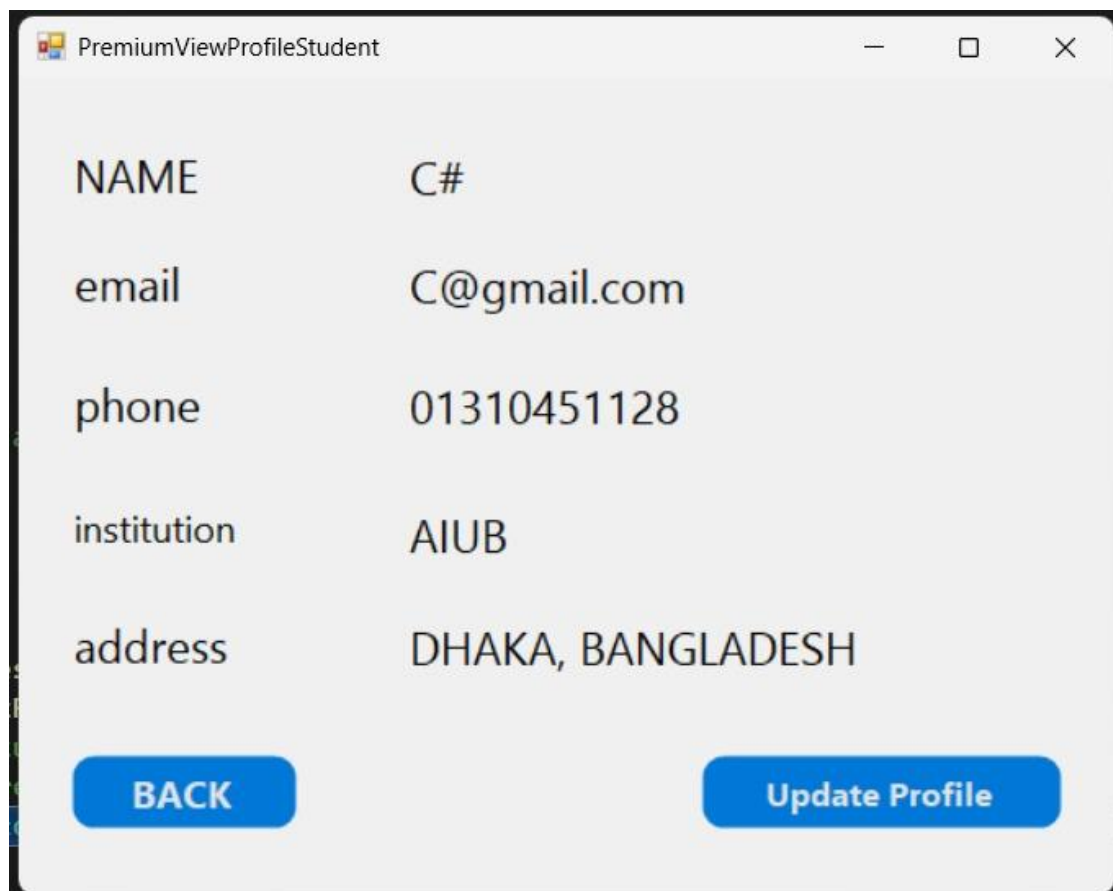
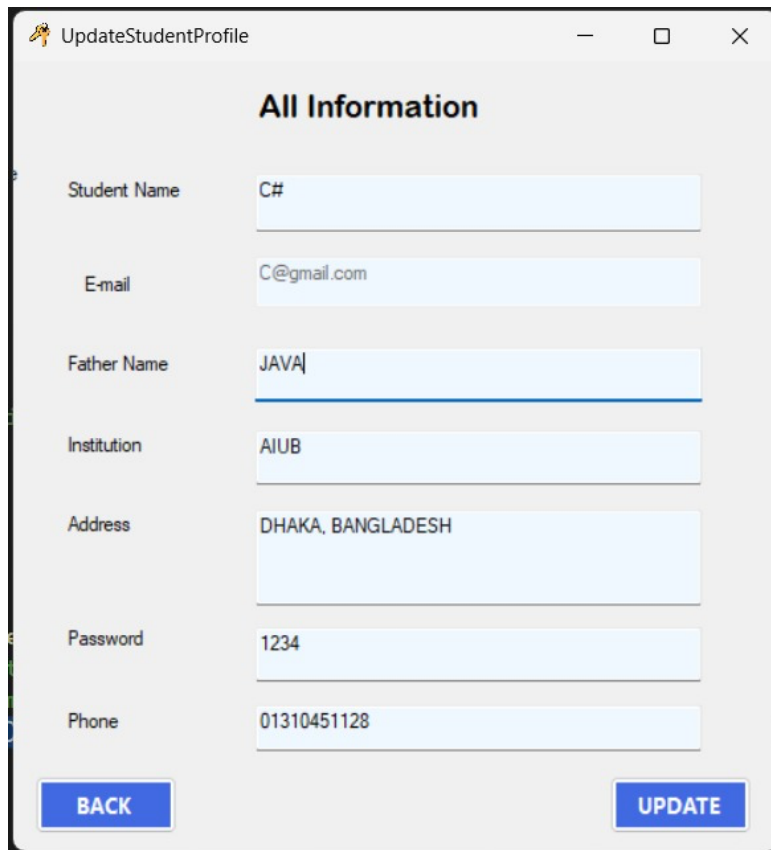


Figure 08: View Profile Page



UpdateStudentProfile

### All Information

Student Name: C#

E-mail: C@gmail.com

Father Name: JAVA

Institution: AIUB

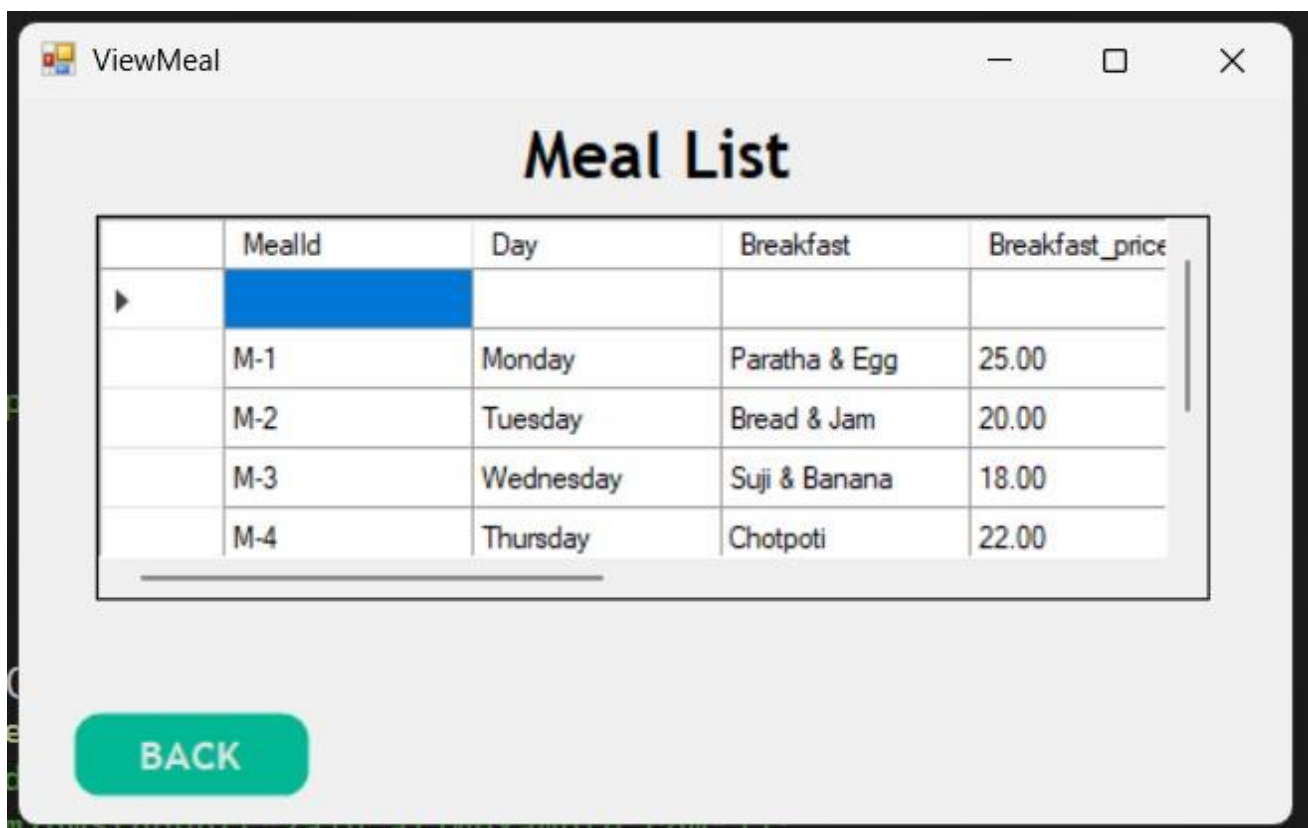
Address: DHAKA, BANGLADESH

Password: 1234

Phone: 01310451128

BACK UPDATE

Figure 09: Update Student Profile Page



ViewMeal

### Meal List

	MealId	Day	Breakfast	Breakfast_price
▶				
	M-1	Monday	Paratha & Egg	25.00
	M-2	Tuesday	Bread & Jam	20.00
	M-3	Wednesday	Suji & Banana	18.00
	M-4	Thursday	Chotpoti	22.00

BACK

Figure 10: View Meal Page

**UpdateMeal**

**Meal Price**

**Update Today's Meal**

Date: Monday , June 30, 2025

Balance

*Breakfast*    B-Food    B-Price \$    Count :

*Lunch*    L-Food    L-Price \$    Count :

*Dinner*    D-Food    D-Price \$    Count :

**Back**    **Confirm**

Figure 11: Update Meal Page

**Deposit**

Date: Monday , June 30, 2025

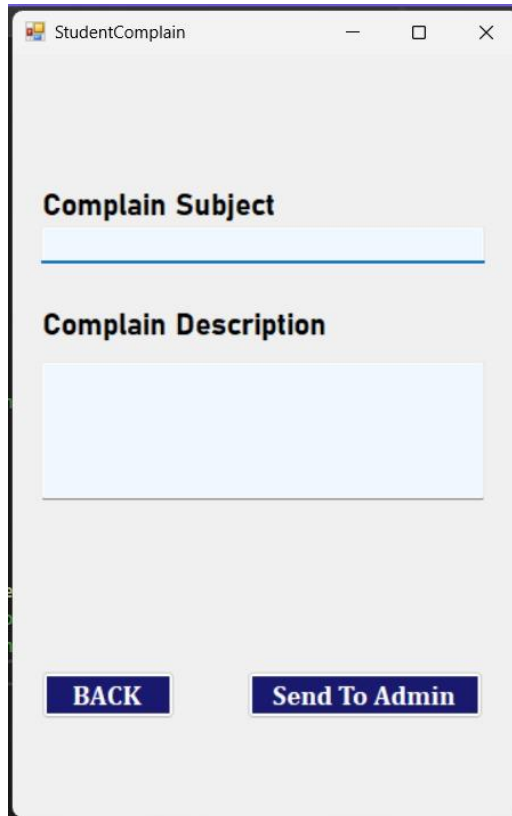
Phone No:

Enter Amount:

Payment Method: ☐ Bkash ☐ Nogod ☐ Roket ☐ Upay

**BACK**    **CONFIRM**

Figure 12: Deposit Page



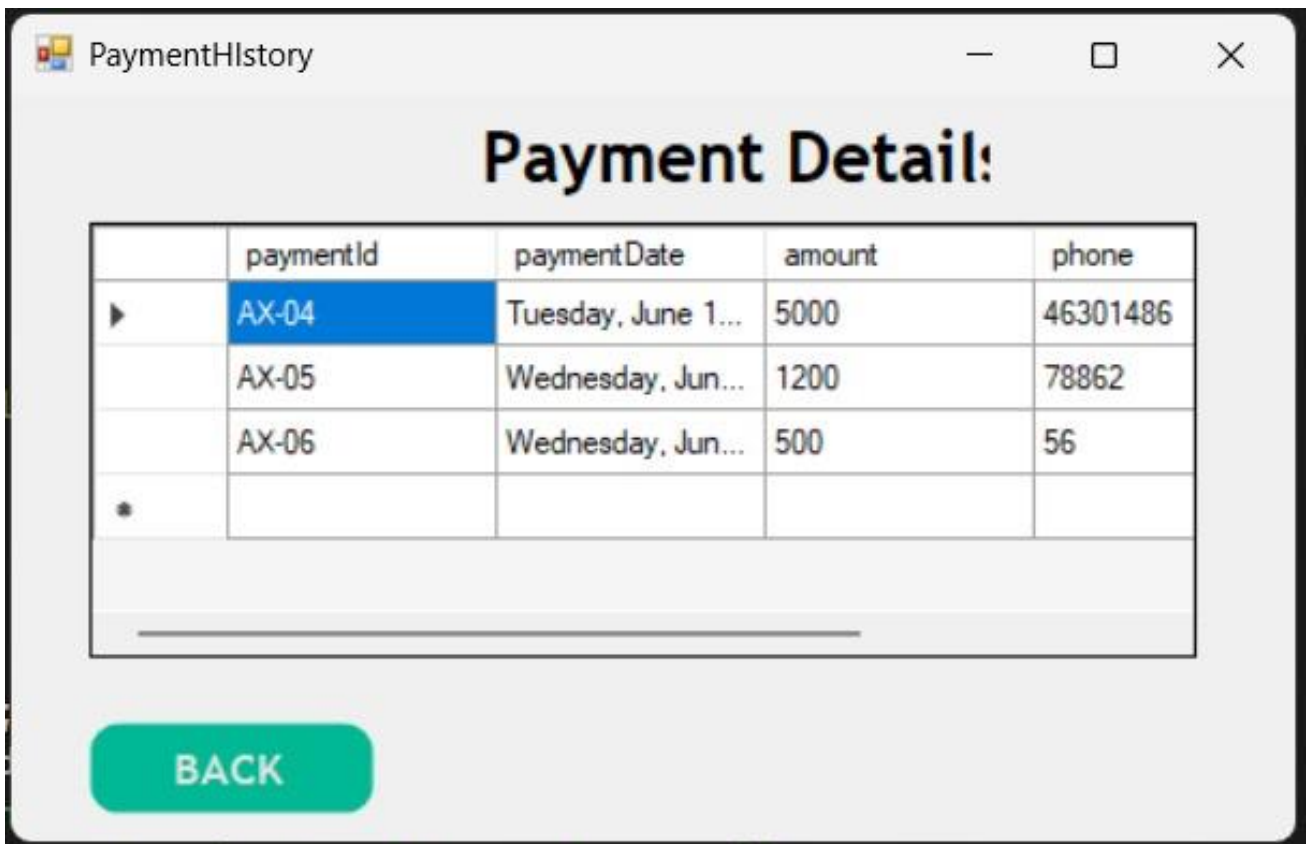
StudentComplain

**Complain Subject**

**Complain Description**

**BACK** **Send To Admin**

Figure 13: Student Complain Page



PaymentHistory

**Payment Detail:**

	paymentId	paymentDate	amount	phone
▶	AX-04	Tuesday, June 1...	5000	46301486
	AX-05	Wednesday, Jun...	1200	78862
	AX-06	Wednesday, Jun...	500	56
•				

**BACK**

Figure 14: Payment History Page

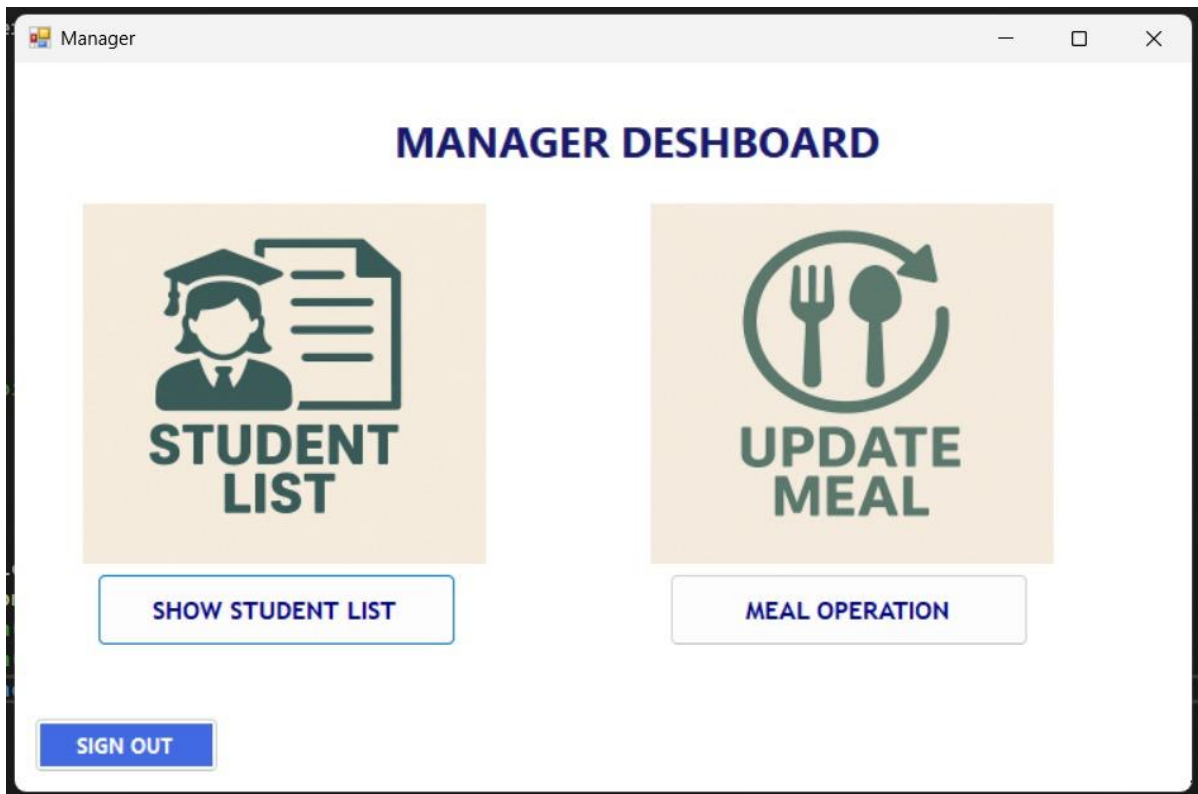


Figure 15: Manager Dashboard

The screenshot shows a web application window titled 'StudentListbyM'. The main heading is 'LIST OF ALL STUDENT'. Below the heading is a table with the following data:

	Name	FatherName	Email	Phone	Institution	Dob	Gender
▶	Bushra Tariq	Tariq Jamil	bushra.tariq@exa...	03171234567	NED	1999-07-07	Female
	C#	JAVA	C@gmail.com	01310451128	AIUB	6/18/2025	
	FARHAN	HELLO	farhan@gmail.com	12456	AIUB	6/10/2025	MALE
	Farhan Chowdhury	coming	farhanchowdhury...	12346789212	AIUB	7/16/2009	MALE
	Fatima Rizvi	Imran Rizvi	fatima.rizvi@exa...	030512	IBA	6/10/2025	MALE

Below the table are several buttons: BACK, DELETE, ADD, RESET, UPDATE, and Search. Below these buttons is a form with the following fields:

- Student Name:
- Father Name:
- Institution:
- Address:
- Password:
- Phone:
- E-mail:
- DOB:
- GENDER: ☐ MALE ☐ FEMALE

Figure 16: Student List by Manager (For Add, Delete, Update Information)

**UpdateMealManager**

### MEAL LIST

MealId	Day	Breakfast	Breakfast_price	Lunch
M-1	Monday	Paratha & Egg	25.00	Rice & Chicken C...
M-2	Tuesday	Bread & Jam	20.00	Rice & Fish Curry
M-3	Wednesday	Suji & Banana	18.00	Polao & Chicken ...
M-4	Thursday	Chapati	22.00	Plain Rice & Egg

Meal-Id  day   
 Breakfast  Price   
 Lunch  Price   
 Dinner  Price

Figure 17: View and Update Meal Menu by Manager

**AdminDashboard**

### ADMIN DASHBOARD

**STUDENT LIST**

**EMPLOYEE LIST**

**EMPLOYEE LIST**

Figure 18: Admin Dashboard Page



The screenshot shows a window titled "ShowStudent" with a title bar containing standard Windows window controls. The main content area is titled "LIST OF ALL STUDENT" in blue text. Below the title is a table with the following columns: Name, FatherName, Email, Phone, Institution, and Dob. The table contains 12 rows of student data. Below the table, there are three buttons: "BACK", "Search", and a text input field labeled "enter email".

	Name	FatherName	Email	Phone	Institution	Dob
▶	Bushra ...	Tariq Ja...	bushra....	031712...	NED	1999-0...
	C#	JAVA	C@qma...	013104...	AIUB	6/18/20...
	FARHAN	HELLO	farhan...	12456	AIUB	6/10/20...
	Farhan ...	coming	farhanc...	123467...	AIUB	7/16/20...
	Fatima ...	Imran R...	fatima.r...	030512	IBA	6/10/20...
	Hafsa Ri...	Riaz Ud...	hafsa.ri...	032512...	IBA	2003-0...
	Hamza ...	Faisal S...	hamza.s...	030412...	COMSA...	1997-1...
	hello	hello	hello@...	017929...	hello	6/16/20...
	Hira Qu...	Salman ...	hira.qur...	030912...	QAU	2000-0...
	Imran ...	Haider ...	imran.h...	031812...	PIEAS	1998-1...
	Moshif B...	Reza Ak	moshif...	031012	AIUB	1998-0...

Buttons: BACK, Search, enter email

Figure 19: Student List (only) View by Admin

The screenshot shows a window titled "ShowEmployeeOp" with a title bar containing standard Windows window controls. The main content area is titled "LIST OF ALL EMPLOYEE" in blue text. Below the title is a table with the following columns: EmployeeId, Name, Salary, Password, and Role. The table contains 2 rows of employee data. Below the table, there are five buttons: "BACK", "ADD", "DELETE", "RESET", and "UPDATE". Below the buttons, there is a "Role" dropdown menu. At the bottom, there are three text input fields labeled "Employee ID", "Name", and "Salary".

EmployeeId	Name	Salary	Password	Role
E-101	Alice Kari...	350.00	alice123	Others
E-102	Tanvir Ra...	28000.00	tanvir123	Staff

Buttons: BACK, ADD, DELETE, RESET, UPDATE

Role:

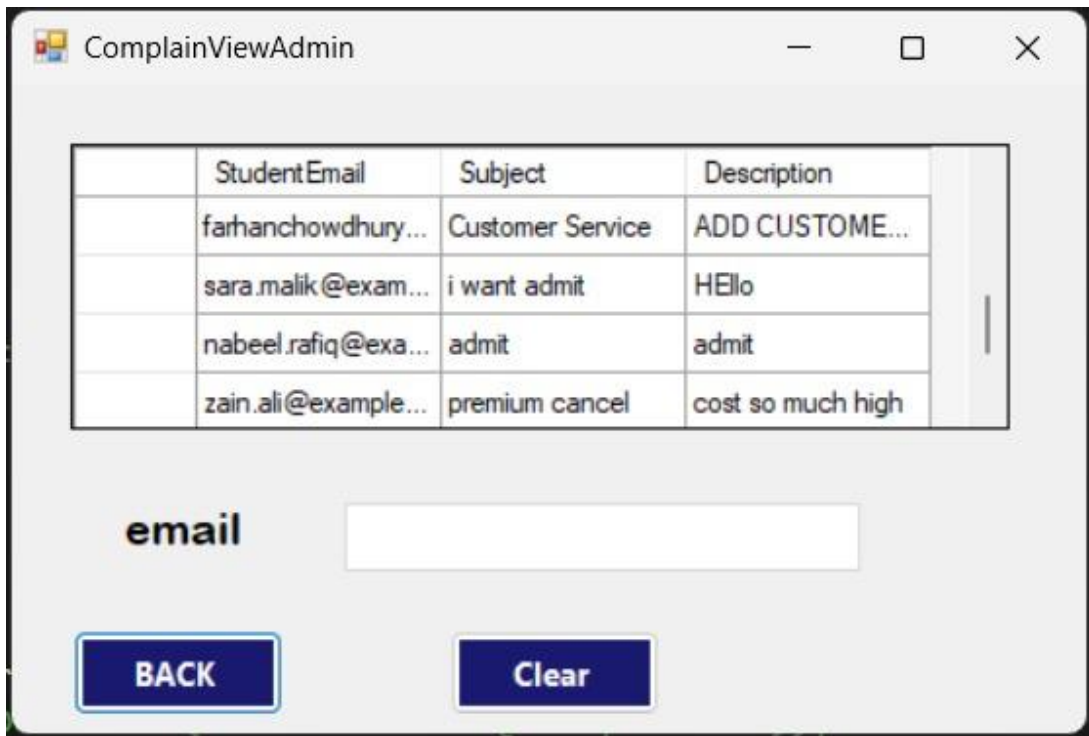
Employee ID:

Name:

Salary:

Figure 20: Password Update after Verifying E-mail



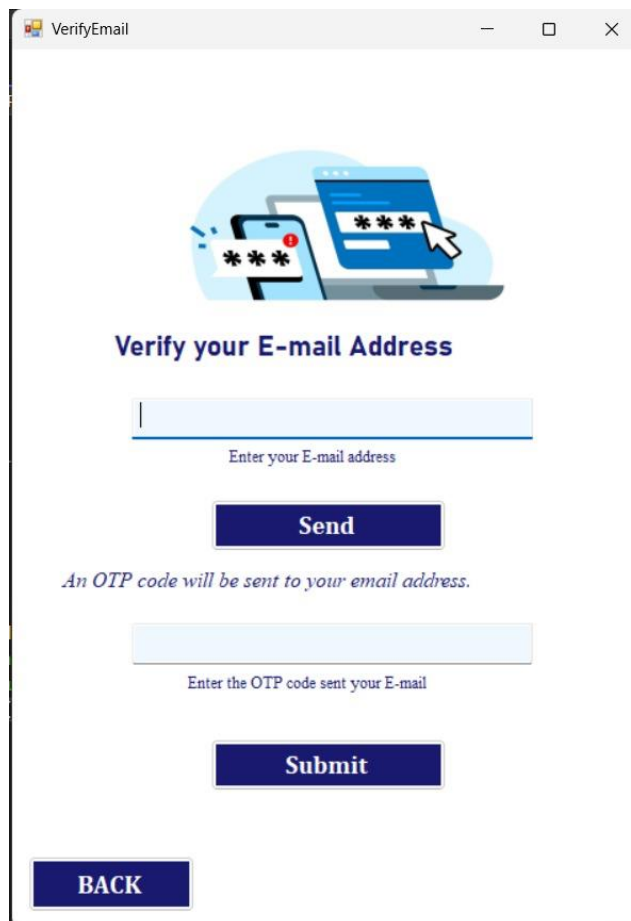


The screenshot shows a window titled "ComplainViewAdmin". Inside, there is a table with the following data:

	StudentEmail	Subject	Description
	farhanchowdhury...	Customer Service	ADD CUSTOME...
	sara.malik@exam...	i want admit	HEllo
	nabeel.rafiq@exa...	admit	admit
	zain.ali@example...	premium cancel	cost so much high

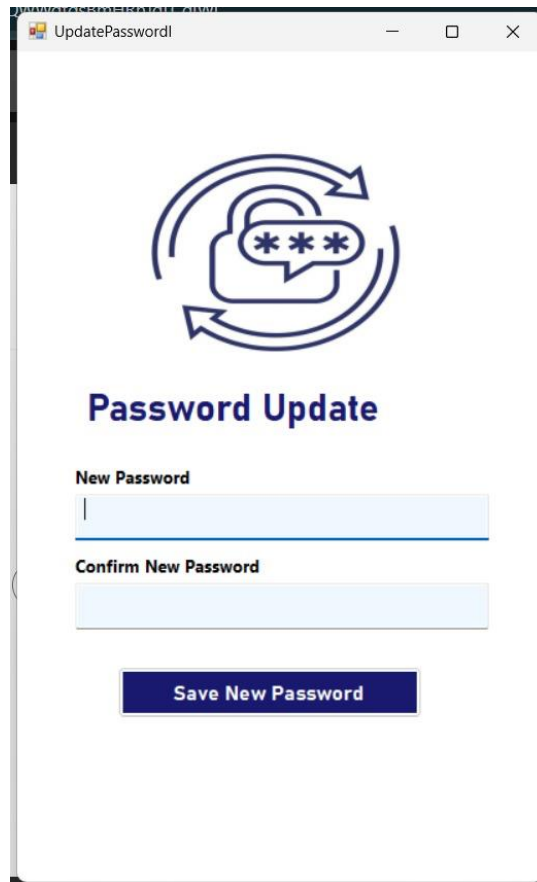
Below the table, there is a label "email" followed by a text input field. At the bottom, there are two buttons: "BACK" and "Clear".

Figure 21: View Complain Page



The screenshot shows a window titled "VerifyEmail". It contains an illustration of a smartphone and a laptop displaying a login screen. Below the illustration, the text "Verify your E-mail Address" is displayed. There is a text input field for the email address, followed by the label "Enter your E-mail address". Below this is a "Send" button. A message states "An OTP code will be sent to your email address." Below this is another text input field for the OTP code, followed by the label "Enter the OTP code sent your E-mail". Below this is a "Submit" button. At the bottom left, there is a "BACK" button.

Figure 22: Verify E-mail for Forget Password Click



The screenshot shows a web browser window titled "UpdatePasswordI". The page features a large circular icon with a padlock and three asterisks, surrounded by curved arrows, indicating a password update process. Below the icon, the heading "Password Update" is displayed in a bold, dark blue font. The form consists of two text input fields: the first is labeled "New Password" and the second is labeled "Confirm New Password". Both fields are currently empty. At the bottom of the form is a dark blue button with the text "Save New Password" in white.

Figure 23: Password Update after Verifying E-mail

THE END