MAY 21, 2024

ACADEMIAEASE: A COMPLETE STUDENT MANAGEMENT SYSTEM

UNDER THE SUPERVISION OF
DR HADEER AHMED
DR MANAL
TA AHMED SAED

PREPARED BY:
USAMA MOHAMMED MOHAMMED ABDULGWAD
AHMED SHAABAN RAGAB
KHALED TARKE SAYED
HOSSAM ALDIN OMAR MOHAMMED

TABLE OF CONTENTS

INTRODUCTION

AcademiaEase is an innovative management system designed to streamline the administration of students, courses, and professors within academic institutions. Developed to address the complexities and inefficiencies inherent in traditional educational management systems, AcademiaEase provides a comprehensive, user-friendly solution tailored to meet the needs of modern educational environments.

AcademiaEase represents a significant step forward in academic management systems, offering a blend of functionality, reliability, and ease of use.

PURPOSE

The primary goal of AcademiaEase is to facilitate seamless interaction and management across various academic activities. By leveraging advanced database management techniques, the system ensures efficient handling of student records, course registrations, and professor, thereby enhancing the overall productivity and operational effectiveness of educational institutions.

KEY FEATURES:

- Student Management: Comprehensive management of student information, including enrollment, personal details, academic performance, and attendance records.
- Course Management: Efficient handling of course details, schedules, and prerequisites, enabling easy course creation, modification, and deletion.
- Professor Management: Detailed management of professor profiles, including their teaching schedules, courses taught, and performance evaluations.
- Payments Management: Secure and efficient processing of tuition fees, scholarships, and other financial transactions, ensuring accurate tracking and reporting of payments.
- User-Friendly Interface: An intuitive interface designed to ensure ease of use for administrators, professors, and students alike.
- Robust Database: A reliable and scalable database that ensures data integrity, security, and quick access to information.

MEMBERS CONTRIBUTION:

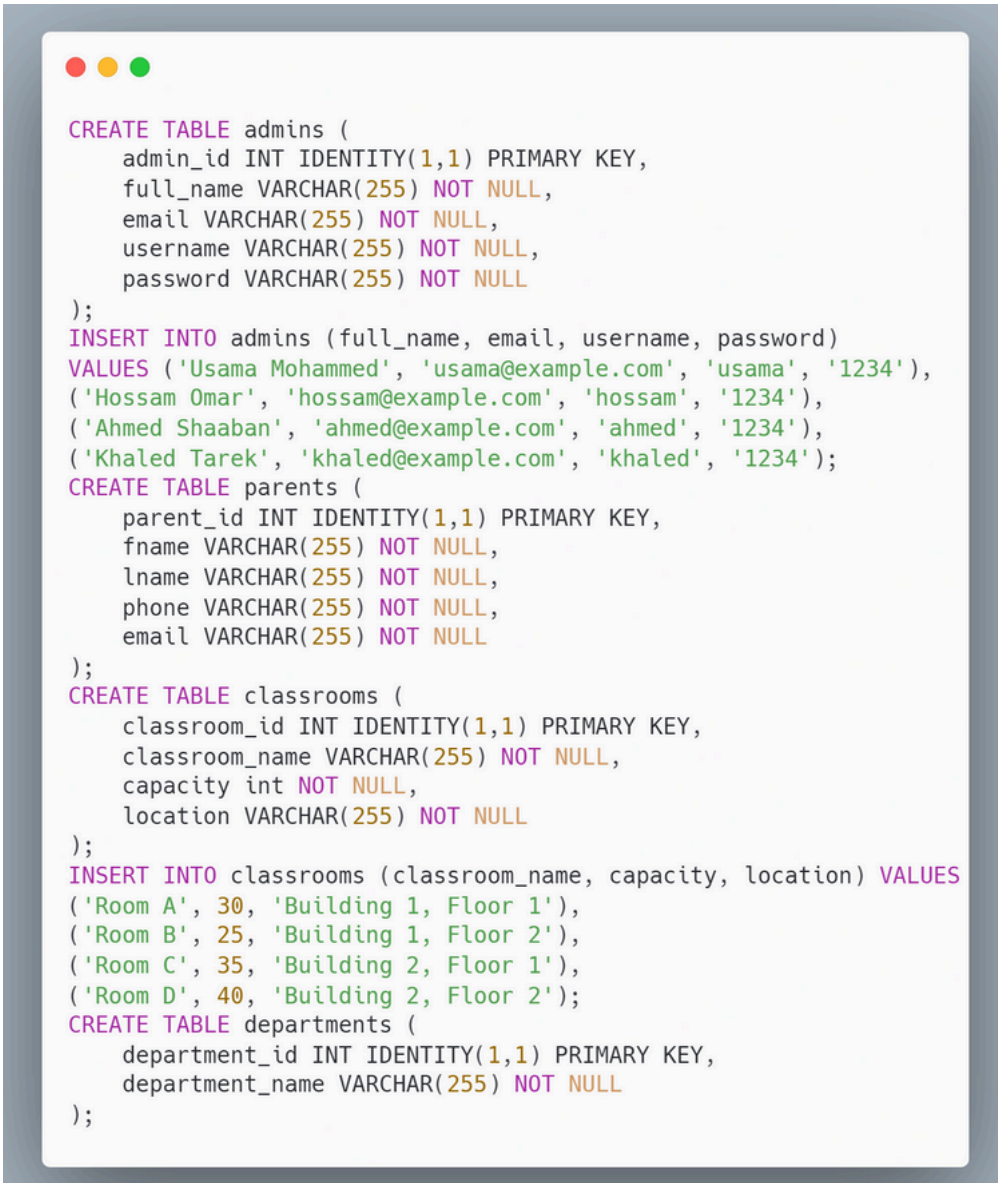
-Usama Mohamed Mohamed Abdelgawad:
was responsible for crucial functionalities of the project: managing and adding new students to the database and updating their information in addition to booking courses for the students and relating their id's to their parents to allow the parent to monitor his children's progress.

-Hossam Eldin Omar Mohamed:
Played a critical role in creating and manipulating the courses which the students will enroll in and assigning the professors with their courses, and specify a classroom for each group of students. That was done comprehensively by joining the students' and professors' tables with the courses' & classrooms' table.

-Khaled Tarek Sayed Othman:
Was responsible for creating the admin's table which is used to only allow admins to login to the system. And was responsible for creating the ER Diagram Model and the relational data model and their relations.

-**Ahmed Shaban Ragab:**
Was responsible for managing the payments and transactions for the each student using comprehensive database queries by joining the students' and payments table.

SQL COMMANDS



```sql
CREATE TABLE admins (
    admin_id INT IDENTITY(1,1) PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
);
INSERT INTO admins (full_name, email, username, password)
VALUES ('Usama Mohammed', 'usama@example.com', 'usama', '1234'),
('Hossam Omar', 'hossam@example.com', 'hossam', '1234'),
('Ahmed Shaaban', 'ahmed@example.com', 'ahmed', '1234'),
('Khaled Tarek', 'khaled@example.com', 'khaled', '1234');
CREATE TABLE parents (
    parent_id INT IDENTITY(1,1) PRIMARY KEY,
    fname VARCHAR(255) NOT NULL,
    lname VARCHAR(255) NOT NULL,
    phone VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
);
CREATE TABLE classrooms (
    classroom_id INT IDENTITY(1,1) PRIMARY KEY,
    classroom_name VARCHAR(255) NOT NULL,
    capacity int NOT NULL,
    location VARCHAR(255) NOT NULL
);
INSERT INTO classrooms (classroom_name, capacity, location) VALUES
('Room A', 30, 'Building 1, Floor 1'),
('Room B', 25, 'Building 1, Floor 2'),
('Room C', 35, 'Building 2, Floor 1'),
('Room D', 40, 'Building 2, Floor 2');
CREATE TABLE departments (
    department_id INT IDENTITY(1,1) PRIMARY KEY,
    department_name VARCHAR(255) NOT NULL
);
```

```
INSERT INTO departments (department_name)
VALUES
    ('Chemical Engineering'),
    ('Computer Engineering'),
    ('Electronics & Communication Engineering'),
    ('Power Machines Engineering'),
    ('Mechanical Engineering'),
    ('Civil Engineering');

CREATE TABLE payments (
    payment_id INT IDENTITY(1,1) PRIMARY KEY,
    payment_amount VARCHAR(255) NOT NULL,
    payment_date VARCHAR(255) NOT NULL,
    student_firstname VARCHAR(255) NOT NULL,
    student_lastname VARCHAR(255) NOT NULL,
    student_id int
);

CREATE TABLE courses (
    course_id INT IDENTITY(1,1) PRIMARY KEY,
    course_name VARCHAR(255) NOT NULL,
    description VARCHAR(255) NOT NULL,
    department_id int NOT NULL,
    classroom_id int NOT NULL,
    CONSTRAINT classroom_id
    FOREIGN KEY (classroom_id)
    REFERENCES classrooms(classroom_id),

    CONSTRAINT department_id
    FOREIGN KEY (department_id)
    REFERENCES departments(department_id)
);
INSERT INTO courses (course_name, description, department_id, classroom_id) VALUES
('Introduction to Programming', 'Basic programming concepts and techniques.', 1, 1),
('Database Management', 'Managing and querying databases using SQL.', 2, 2),
('Computer Networks', 'Understanding network architectures and protocols.', 3, 3),
('Software Engineering', 'Software development methodologies and practices.', 1, 4);
```

```sql
CREATE TABLE professors (
    professors_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name varchar(20),
    last_name varchar(20),
    department_id int NOT NULL,
    course_id int NOT NULL,

    FOREIGN KEY (course_id)
    REFERENCES courses(course_id),

    FOREIGN KEY (department_id)
    REFERENCES departments(department_id)
);
INSERT INTO professors (first_name, last_name, department_id, course_id)
VALUES ('John', 'Doe', 1, 5),
       ('Jane', 'Smith', 2, 3),
       ('Alice', 'Johnson', 2, 5),
       ('Bob', 'Williams', 2, 3),
       ('Charlie', 'Brown', 2, 5),
       ('David', 'Lee', 2, 3);
CREATE TABLE students (
    student_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name VARCHAR(20) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    department_id INT NOT NULL,
    course_id INT NOT NULL,
    parent_id INT NOT NULL,
    birthdate DATE,
    gender VARCHAR(10),

    FOREIGN KEY (course_id) REFERENCES courses(course_id),
    FOREIGN KEY (department_id) REFERENCES departments(department_id),
    FOREIGN KEY (parent_id) REFERENCES parents(parent_id)
);
INSERT INTO students (first_name, last_name, department_id, course_id, parent_id, birthdate, gender)
VALUES
    ('John', 'Doe', 1, 3, 1, '2000-01-01', 'M'),
    ('Jane', 'Smith', 2, 3, 2, '2001-02-15', 'F'),
    ('Alice', 'Johnson', 2, 3, 2, '1999-05-20', 'F'),
    ('Bob', 'Brown', 1, 3, 2, '2002-03-10', 'M'),
    ('Eve', 'White', 2, 3, 2, '1998-06-25', 'F'),
    ('Charlie', 'Green', 3,3, 1, '2003-09-05', 'M');
INSERT INTO parents (fname, lname, phone, email, relation)
VALUES
    ('John', 'Doe', '123-456-7890', 'john.doe@example.com'),
    ('Jane', 'Smith', '456-789-0123', 'jane.smith@example.com'),
    ('Alice', 'Johnson', '789-012-3456', 'alice.johnson@example.com'),
    ('Bob', 'Brown', '012-345-6789', 'bob.brown@example.com',),
    ('Eve', 'White', '345-678-9012', 'eve.white@example.com',),
    ('Charlie', 'Green', '678-901-2345', 'charlie.green@example.com');
```

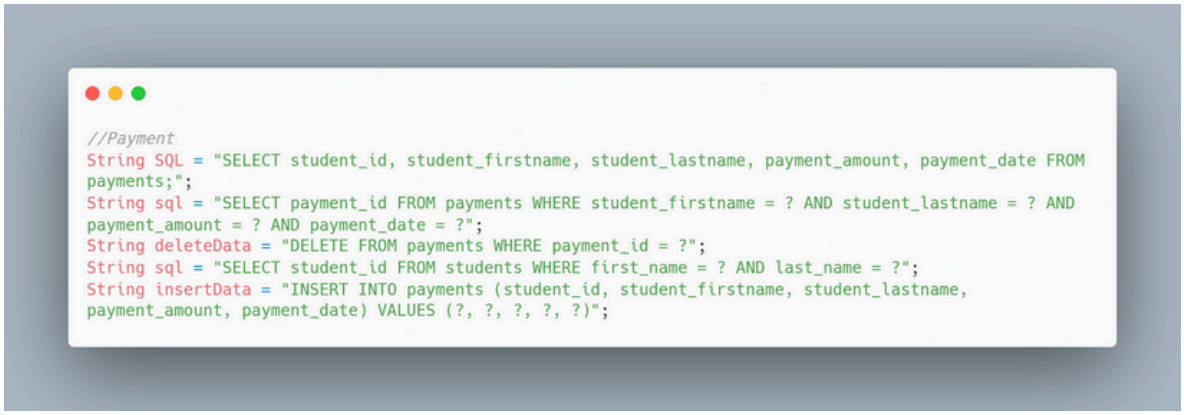
```java
//Login

String SQLQuery = "SELECT * FROM admins WHERE username = ? AND password = ?";
String insertData = "INSERT INTO admins "
        + "(full_name,email,username,password) "
        + "VALUES(?,?,?,?)";
String checkData = "SELECT email FROM admins WHERE email = '"
```

```
//Home
String TotalSutdentsSQL = "SELECT COUNT(*) FROM students;";
String TotalMaleSutdentsSQL = "SELECT COUNT(*) FROM students where gender = 'male';";
String TotalFemaleSutdentsSQL = "SELECT COUNT(*) FROM students where gender = 'female';";
String TotalProfessorssSQL = "SELECT COUNT(*) FROM professors;";
String TotaldepartmentsSQL = "SELECT COUNT(*) FROM departments;";
String TotalcoursesSQL = "SELECT COUNT(*) FROM courses;";
Student Affairs
String SQL = "SELECT department_name from departments;";
String sql = "SELECT * FROM parents WHERE fname = ? AND lname = ? AND phone = ?";
String sql = "INSERT INTO parents (fname, lname, phone, email, relation) OUTPUT INSERTED.parent_id
VALUES (?, ?, ?, ?, ?)";
String insertDataintoStudents = "INSERT INTO students (first_name, last_name, department_id, course_id,
parent_id, birthdate, gender) VALUES (?, ?, ?, ?, ?, ?, ?)";
String sql = "SELECT parent_id FROM parents WHERE fname = ? AND lname = ? AND phone = ?";
String departmentQuery = "SELECT department_id FROM departments WHERE department_name = ?";
String courseQuery = "SELECT course_id FROM courses WHERE course_name = ?";
String insertDataintoStudents = "INSERT INTO students (first_name, last_name, department_id, course_id,
parent_id, birthdate, gender) VALUES (?, ?, ?, ?, ?, ?, ?)";
String departmentQuery = "SELECT department_id FROM departments WHERE department_name = ?";
String courseQuery = "SELECT course_id FROM courses WHERE course_name = ?";
String SQL = "SELECT parent_id from students WHERE first_name = ? AND last_name = ?;";
String parentSQL = "SELECT * FROM parents WHERE parent_id = ?";
String SQL = "SELECT s.first_name, s.last_name, d.department_name, c.course_name, birthdate, gender
FROM students s JOIN courses c ON s.course_id = c.course_id JOIN departments d ON s.department_id =
d.department_id;";
String sql = "SELECT COUNT(*) AS count FROM students WHERE parent_id = ?";
String sql = "SELECT student_id FROM students WHERE first_name = ? AND last_name = ? AND department_id
= ? AND course_id = ? AND parent_id = ? AND birthdate = ? AND gender = ?";
String deleteQuery = "DELETE FROM students WHERE student_id = ?";
String deleteQuery = "DELETE FROM parents WHERE parent_id = ?";
```

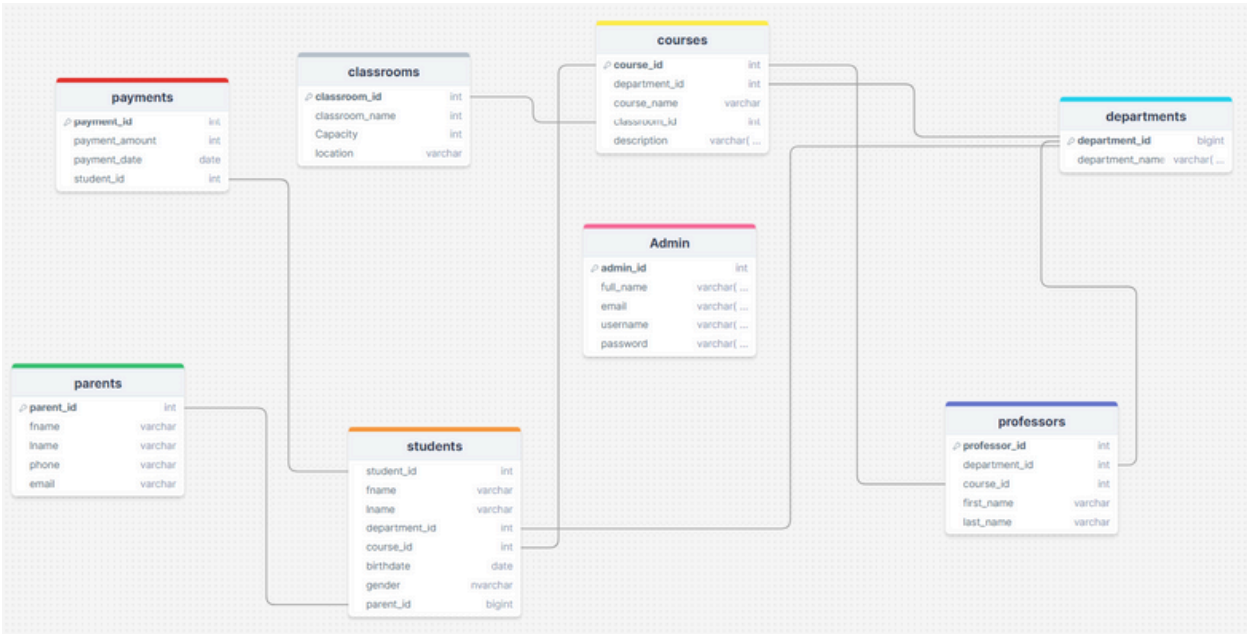
```
//Professors Affairs
String SQL = "SELECT p.first_name, p.last_name, d.department_name, c.course_name FROM professors p JOIN
courses c ON p.course_id = c.course_id JOIN departments d ON p.department_id = d.department_id;";
String SQL = "SELECT department_name from departments;";
String SQL = "SELECT course_name FROM courses;";
String insertDataintoProfessors = "INSERT INTO professors (first_name, last_name, department_id,
course_id) VALUES (?, ?, ?, ?)";
String sql = "SELECT professors_id FROM professors WHERE first_name = ? AND last_name = ? AND
department_id = ? AND course_id = ?";
String deleteData = "DELETE FROM professors WHERE professors_id = ?;";
```

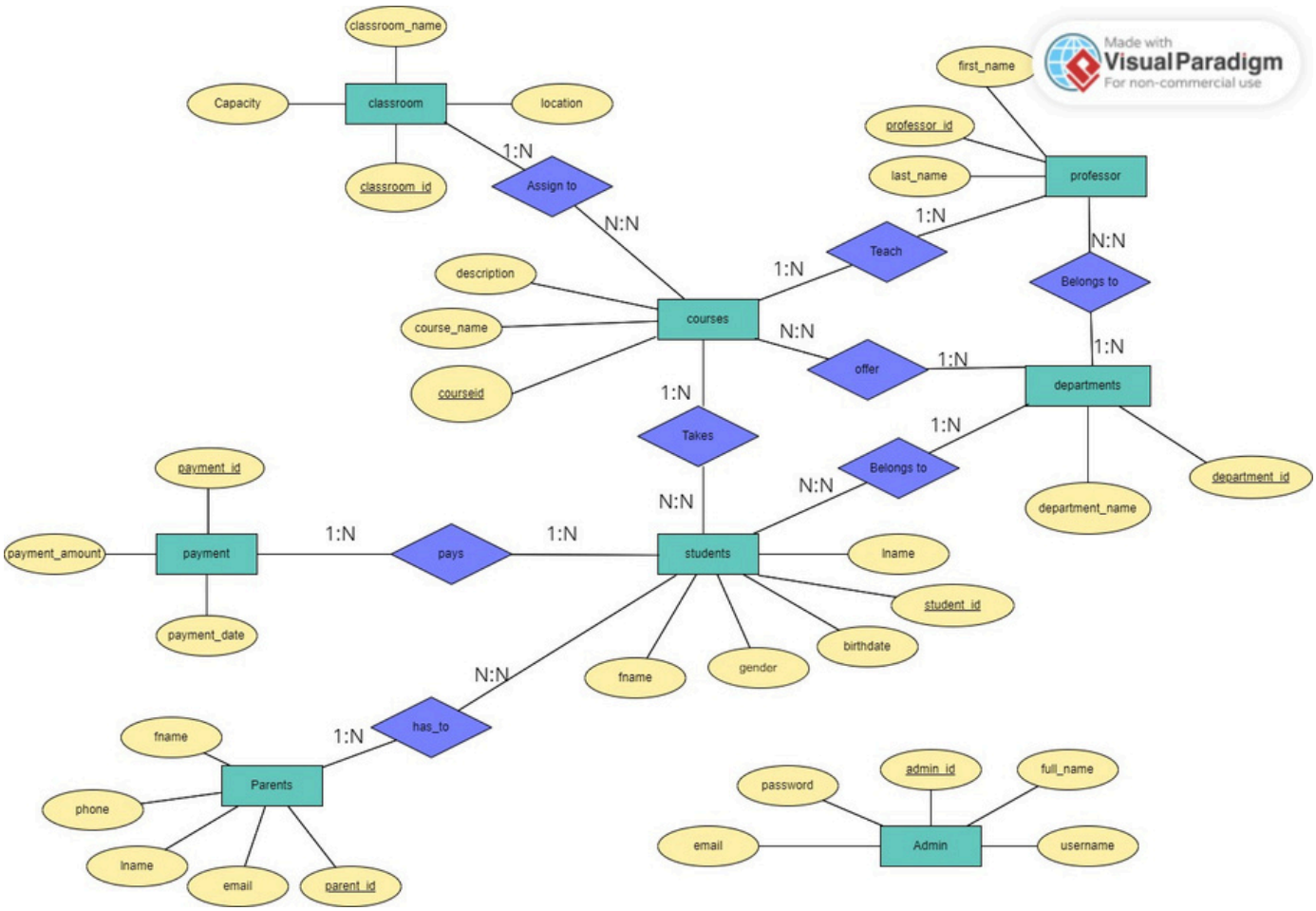
```
//Courses & Classrooms
String SQL = "SELECT classroom_name from classrooms;";
String SQL = "SELECT department_name from departments;";
String SQL = "SELECT c.course_name, c.description, d.department_name, cl.classroom_name FROM courses c
JOIN classrooms cl ON c.classroom_id = cl.classroom_id JOIN departments d ON c.department_id =
d.department_id;";
String departmentQuery = "SELECT department_id FROM departments WHERE department_name = ?";
String classroomQuery = "SELECT classroom_id FROM classrooms WHERE classroom_name = ?";
String insertData = "INSERT INTO courses (course_name, department_id, classroom_id, description) VALUES
(?, ?, ?, ?)";
String deleteData = "DELETE FROM courses WHERE course_name = '"
String SQL = "SELECT classroom_name, capacity, location from classrooms;";
String insertData = "INSERT INTO classrooms (classroom_name, capacity, location) VALUES (?, ?, ?)";
String deleteData = "DELETE FROM classrooms WHERE classroom_name = '"
```



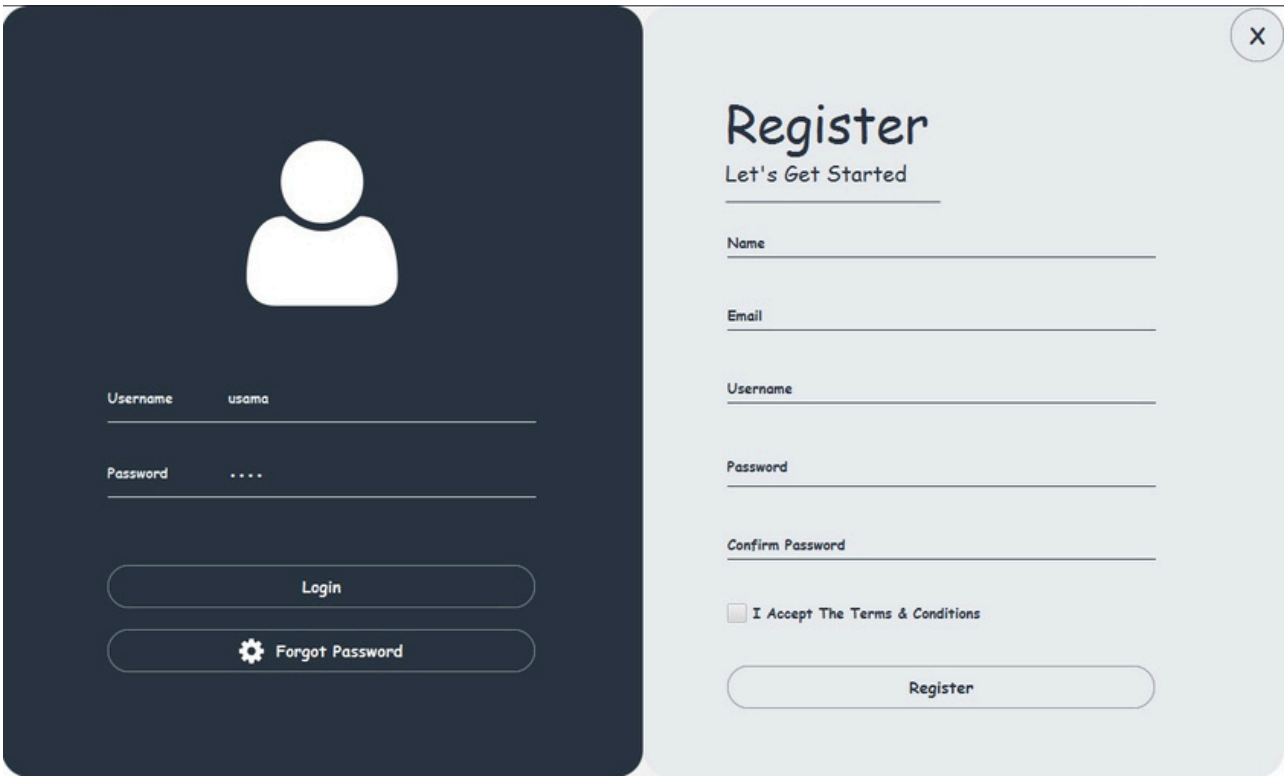
```
//Payment
String SQL = "SELECT student_id, student_firstname, student_lastname, payment_amount, payment_date FROM
payments;";
String sql = "SELECT payment_id FROM payments WHERE student_firstname = ? AND student_lastname = ? AND
payment_amount = ? AND payment_date = ?";
String deleteData = "DELETE FROM payments WHERE payment_id = ?";
String sql = "SELECT student_id FROM students WHERE first_name = ? AND last_name = ?";
String insertData = "INSERT INTO payments (student_id, student_firstname, student_lastname,
payment_amount, payment_date) VALUES (?, ?, ?, ?, ?)";
```

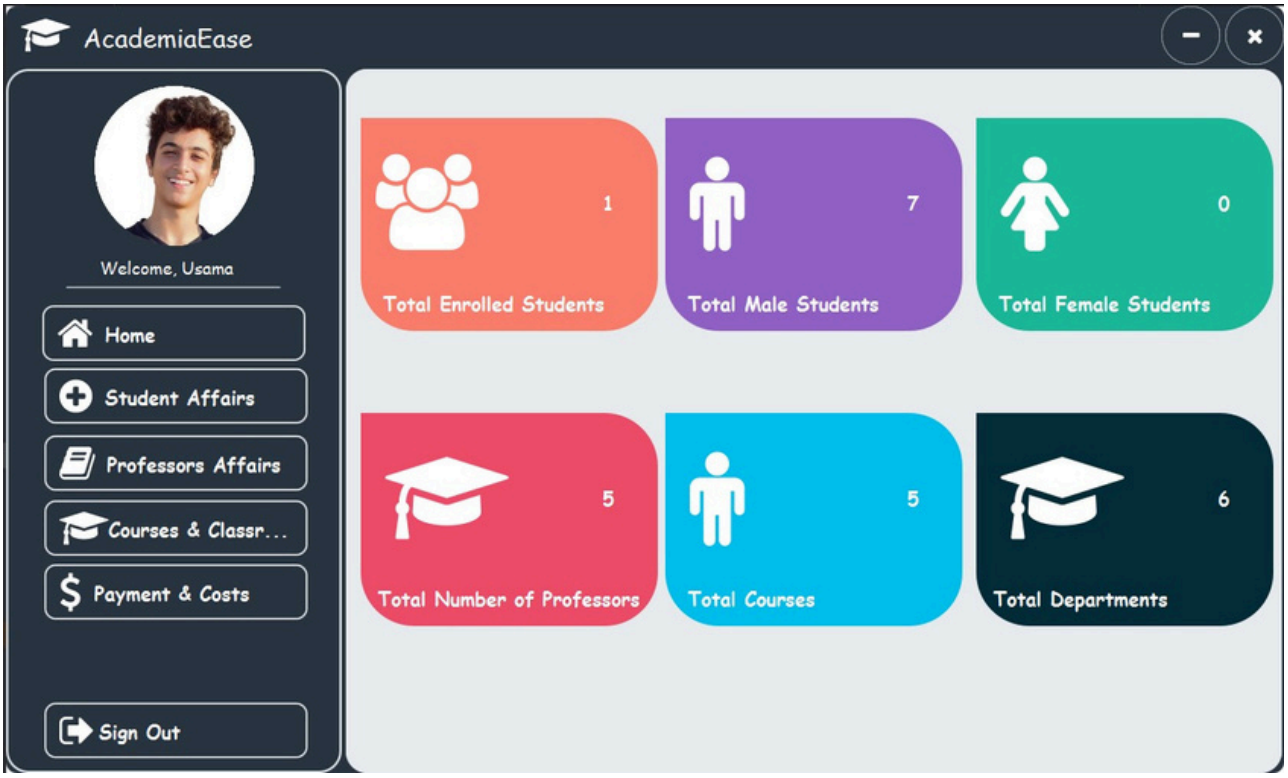
ER DIAGRAM & SCHEMA DIAGRAM

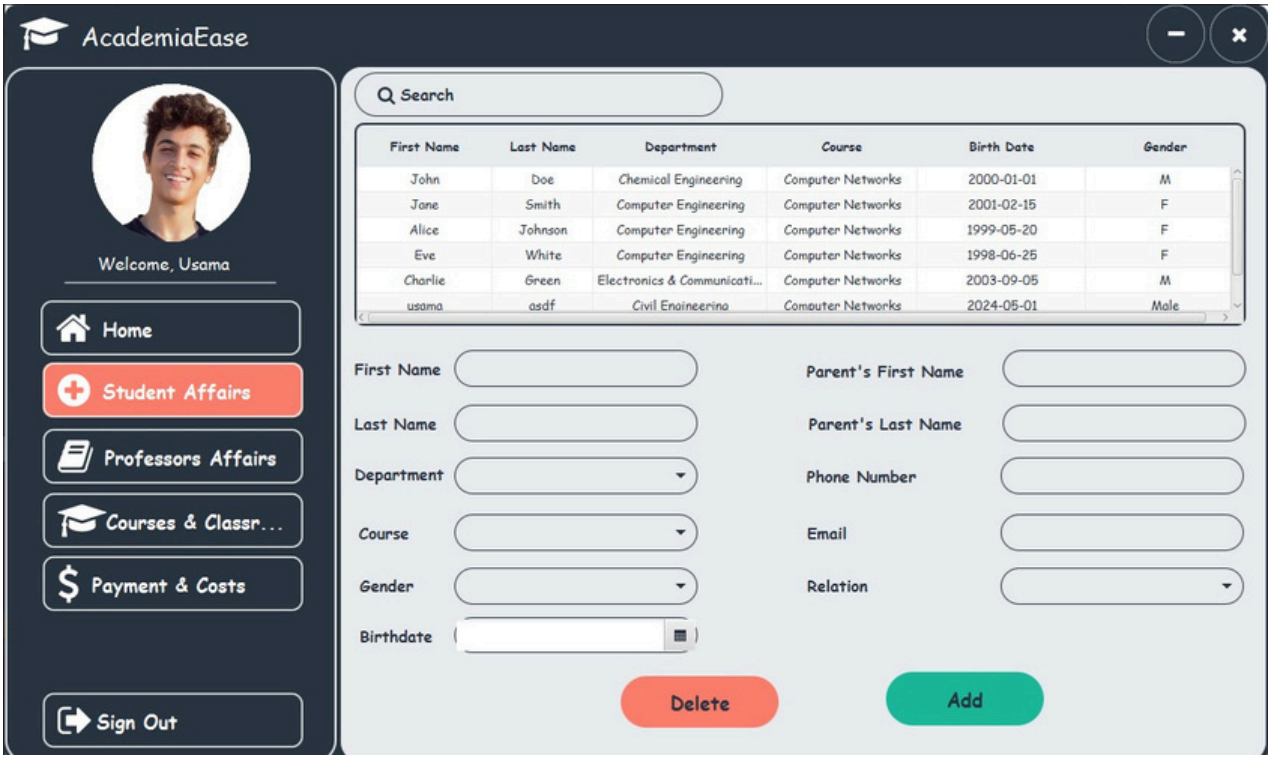


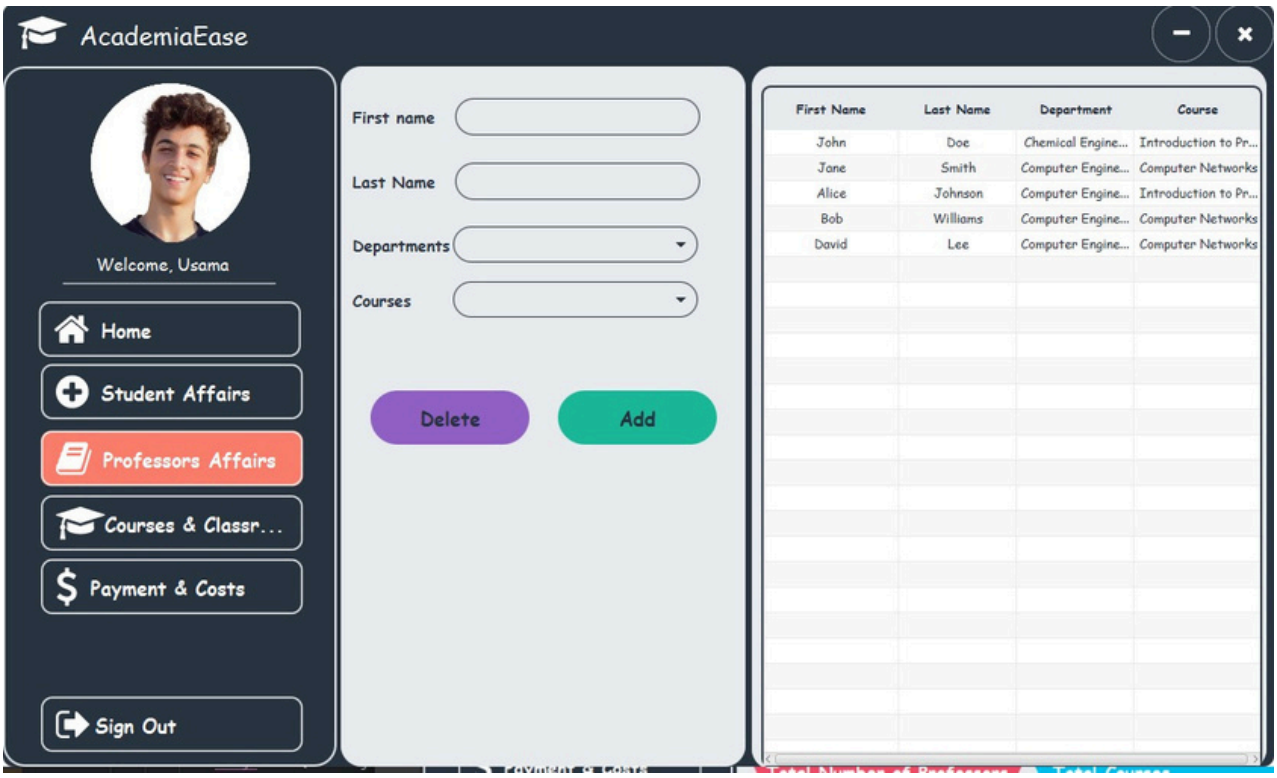


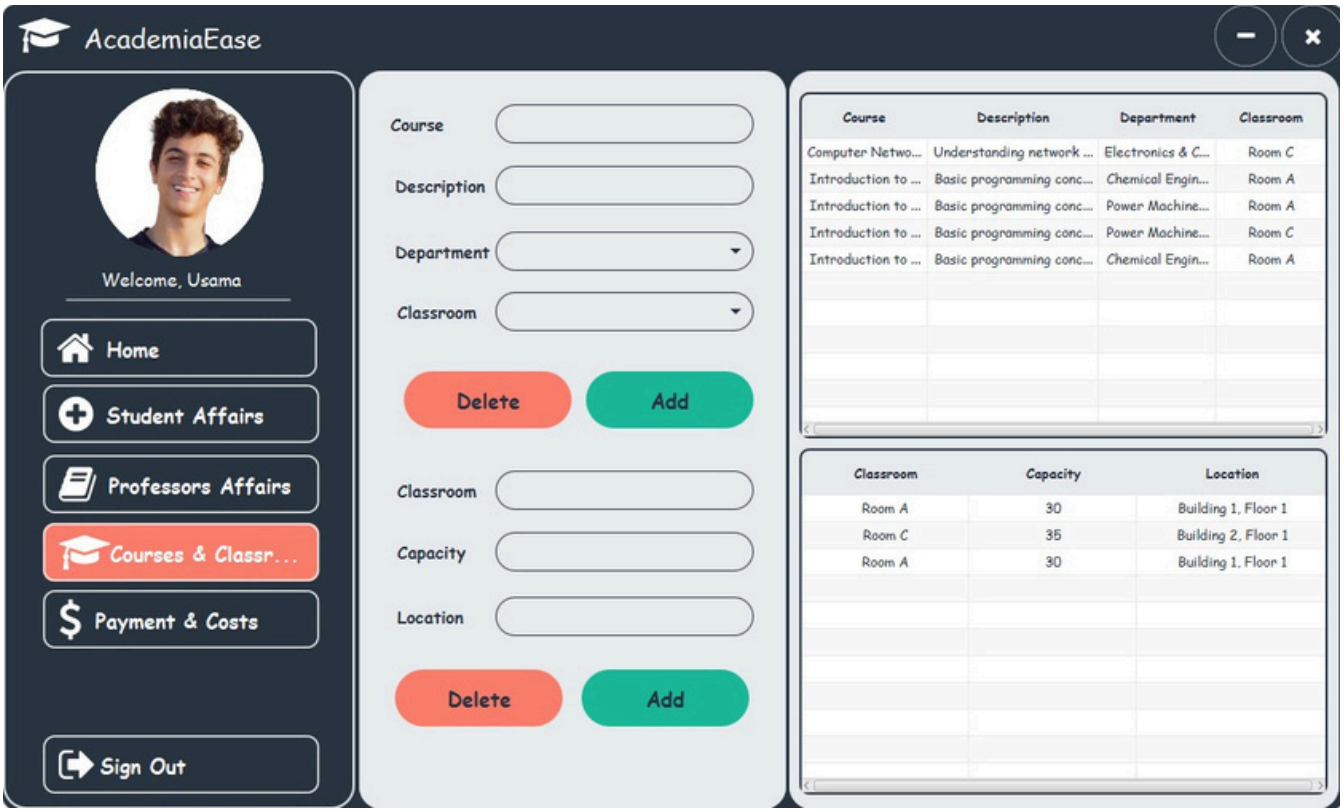
APPLICATION SNAPSHOTS

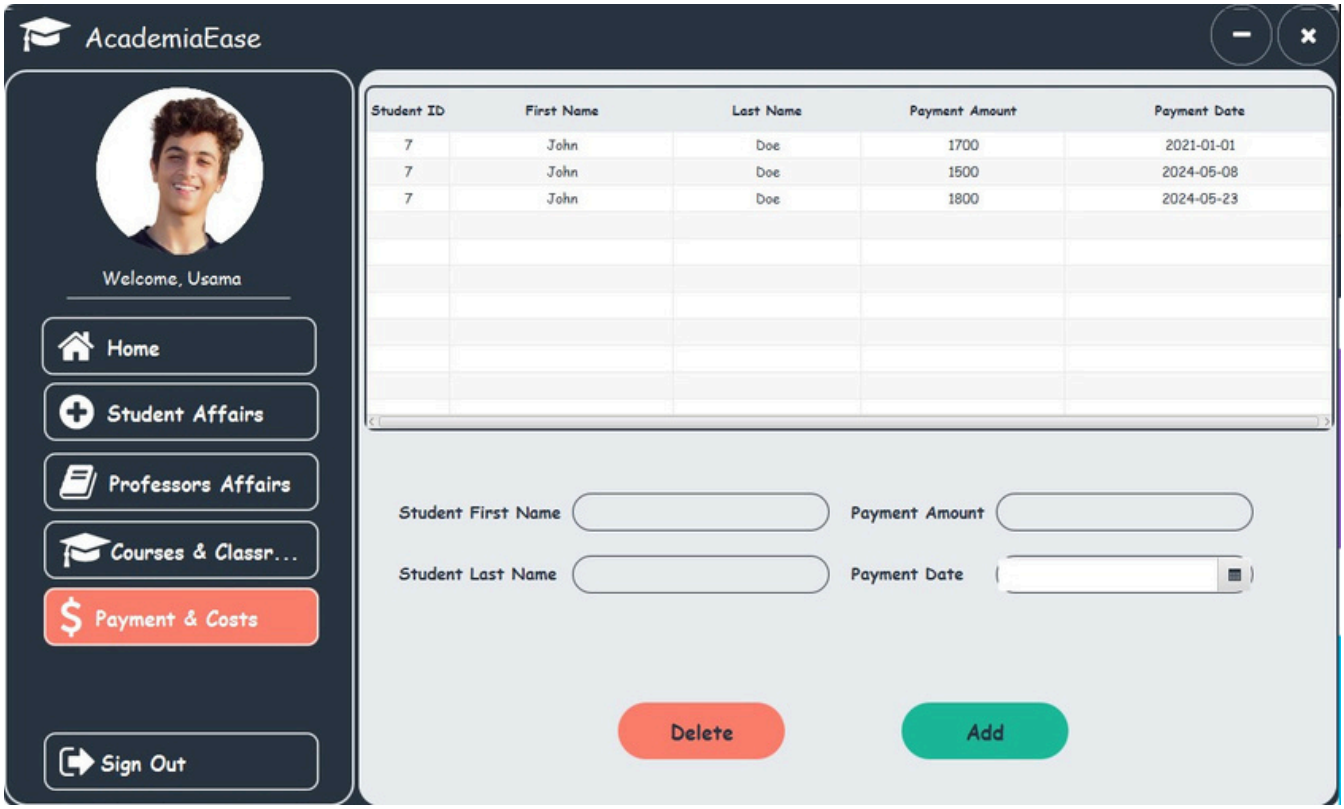


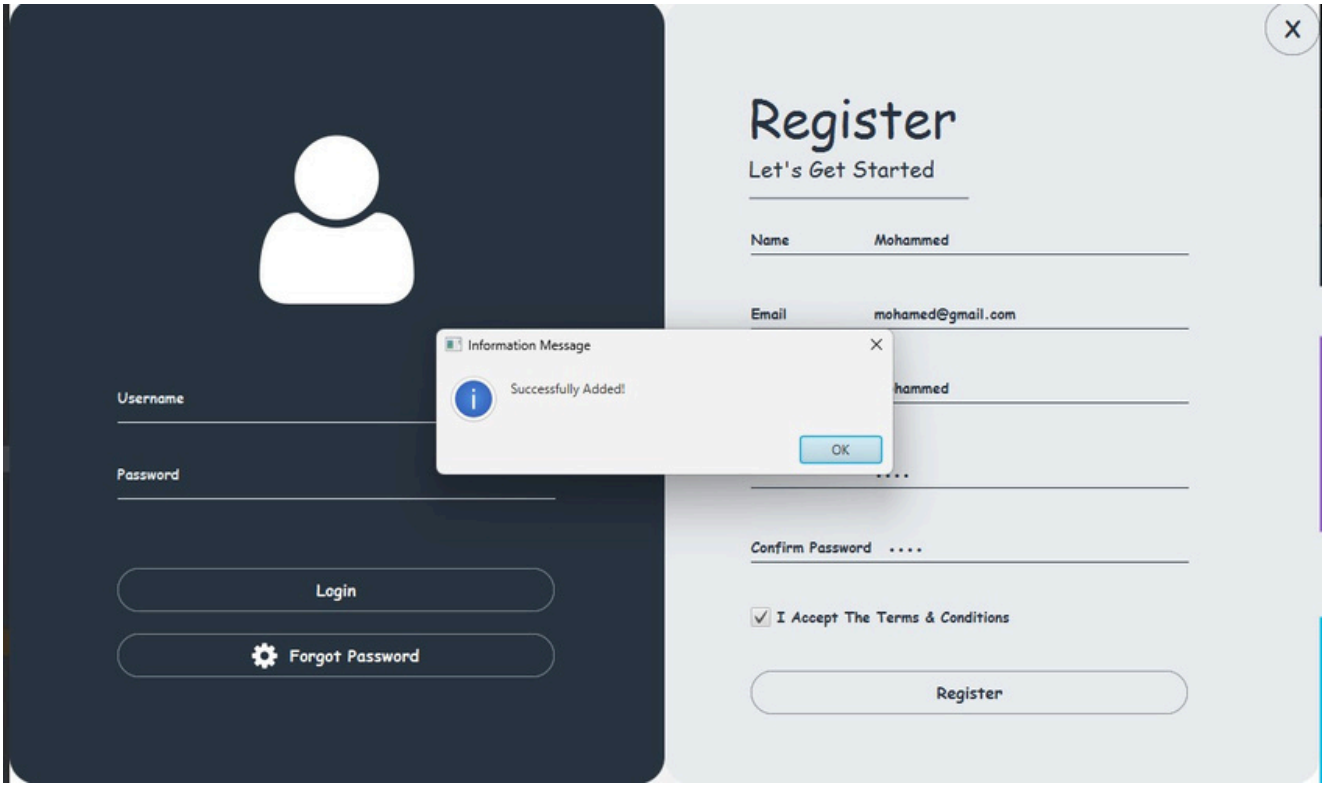



AcademiaEase
Welcome, Usama
Home
Student Affairs
Professors Affairs
Courses & Classr...
Payment & Costs
Sign Out
Search
First Name
Last Name
Department
Course
Birth Date
Gender
John
Doe
Chemical Engineering
Computer Networks
2000-01-01
M
Jane
Smith
Computer Engineering
Computer Networks
2001-02-15
F
Alice
Johnson
Computer Engineering
Computer Networks
1999-05-20
F
Eve
White
Computer Engineering
Computer Networks
1998-06-25
F
Charlie
Green
Electronics & Communicati...
Computer Networks
2003-09-05
M
usama
asdf
Civil Engineering
Computer Networks
2024-05-01
Male
First Name
Last Name
Department
Course
Gender
Birthdate
Parent's First Name
Parent's Last Name
Phone Number
Email
Relation
Delete
Add


AcademiaEase
Welcome, Usama
Home
Student Affairs
Professors Affairs
Courses & Classr...
Payment & Costs
Sign Out
First name
Last Name
Departments
Courses
Delete
Add
First Name
Last Name
Department
Course
John
Doe
Chemical Engine...
Introduction to Pr...
Jane
Smith
Computer Engine...
Computer Networks
Alice
Johnson
Computer Engine...
Introduction to Pr...
Bob
Williams
Computer Engine...
Computer Networks
David
Lee
Computer Engine...
Computer Networks


AcademiaEase
Welcome, Usama
Home
Student Affairs
Professors Affairs
Courses & Classr...
Payment & Costs
Sign Out
Course
Description
Department
Classroom
Delete
Add
Classroom
Capacity
Location
Delete
Add
Course | Description | Department | Classroom
Computer Netwo... | Understanding network ... | Electronics & C... | Room C
Introduction to ... | Basic programming conc... | Chemical Engin... | Room A
Introduction to ... | Basic programming conc... | Power Machine... | Room A
Introduction to ... | Basic programming conc... | Power Machine... | Room C
Introduction to ... | Basic programming conc... | Chemical Engin... | Room A
Classroom | Capacity | Location
Room A | 30 | Building 1, Floor 1
Room C | 35 | Building 2, Floor 1
Room A | 30 | Building 1, Floor 1


AcademiaEase
Welcome, Usama
Home
Student Affairs
Professors Affairs
Courses & Classr...
Payment & Costs
Sign Out
Student ID | First Name | Last Name | Payment Amount | Payment Date
7 | John | Doe | 1700 | 2021-01-01
7 | John | Doe | 1500 | 2024-05-08
7 | John | Doe | 1800 | 2024-05-23
Student First Name
Student Last Name
Payment Amount
Payment Date
Delete
Add


Register
Let's Get Started
Name Mohammed
Email mohamed@gmail.com
Confirm Password ....
I Accept The Terms & Conditions
Register
Information Message
Successfully Added!
OK
Username
Password
Login
Forgot Password