

Ministry of Education and Training
Saigon International University
SPECIALTY: ENGINEERING AND COMPUTER SCIENCE



FINAL REPORT

TOPIC: HỆ THỐNG QUẢN LÝ KHÁCH HÀNG TẠI MÔI TRƯỜNG TẬP GYM

Subject: Software Engineering

Instructor: Nguyễn Thị Bích Nguyên

Member: Võ Anh Minh

Võ Minh Khoa

Nguyễn Đức Phát

HO CHI MINH CITY, 2023

NỘI DUNG:

CHƯƠNG 1: TÓM LUỢC

CHƯƠNG 2: QUẢN LÝ KHÁCH HÀNG TẠI PHÒNG TẬP GYM

I. Đặc trưng của đồ án.....	4
II. Truy vấn SQL.....	5
1. Table users:.....	6
2. Table room:.....	6
3. Table class:.....	7
4. Table membership:.....	8
5. Table customer:.....	9
6. Table payment:.....	10
7. Table signup:.....	11
8. Table staff:.....	11
9. Table staff_class:.....	12
10. Trigger insertinuseroninsertcustomer:.....	13
11. Trigger updatepaymentoninsertcustomer:.....	13
12. Trigger updatepaymentonmembershipupgrade:.....	14
13. Trigger updatestatus (missing in sample data):.....	14
III. Một số hình ảnh về kết quả.....	15
CHAPTER 3: KẾT LUẬN	

CHƯƠNG 1. TÓM LUỢC

Ngày nay, khi cuộc sống trở nên nhanh hơn và yêu cầu về sức khỏe cũng như thể lực ngày càng tăng, các phòng tập gym đang gặp trở ngại lớn trong việc quản lý khách hàng. Đồng thời, sự đa dạng và phức tạp của thông tin cá nhân khách hàng cũng như quản lý theo dõi và giao dịch thanh toán đặt ra yêu cầu cao về hiệu suất và độ chính xác.

Trong bối cảnh đó, việc áp dụng cơ sở dữ liệu như Microsoft SQL Server (MSSQL) không chỉ trở thành xu hướng mà còn là một bước quan trọng để nâng cao khả năng quản lý của các phòng tập hiện đại. Bài viết này sẽ đi sâu vào giải pháp này, đặc biệt là trình bày cách áp dụng MSSQL để tối ưu hóa việc quản lý khách hàng.

Trước khi xem xét những ưu điểm và giải pháp mà SQL mang lại, chúng ta cần tìm hiểu thực trạng quản lý khách hàng tại các phòng gym hiện nay. Một trong những trở ngại lớn nhất là quản lý thông tin chi tiết của hàng nghìn khách hàng cùng một lúc. Từ tên, tuổi, địa chỉ,... mọi thứ đều cần được lưu trữ và truy xuất nhanh chóng.

Ngoài ra, việc theo dõi thanh toán và quản lý giao dịch cũng là một công việc khó khăn. Độ chính xác rất quan trọng để tránh nhầm lẫn và tranh chấp về thanh toán cũng như xây dựng mối quan hệ lâu dài với khách hàng. Điều này đặt ra yêu cầu về tính linh hoạt và tự động hóa trong quá trình thanh toán.

Bài viết này không chỉ nhằm mục đích giới thiệu cách sử dụng SQL trong hệ thống quản lý khách hàng mà còn bổ sung thêm NetBeans vào cơ sở quản lý này. Mục tiêu là tận dụng tính linh hoạt của NetBeans để xây dựng giao diện người dùng thân thiện và dễ sử dụng, kết hợp với sức mạnh của SQL trong xử lý và lưu trữ dữ liệu.

Chúng tôi sẽ đề cập cụ thể cách các bảng dữ liệu được thiết kế để chứa thông tin chi tiết và mối quan hệ giữa chúng. Đồng thời, chúng tôi sẽ mô tả cách NetBeans được tích hợp để tối ưu hóa trải nghiệm người dùng trong việc quản lý thông tin khách hàng, từ đăng ký đến theo dõi lịch tập luyện.

CHƯƠNG 2. QUẢN LÝ KHÁCH HÀNG TẠI PHÒNG TẬP GYM

I. Đặc trưng của đồ án:

Để làm rõ, dự án của chúng tôi nhắm đến hai đối tượng cơ bản của hệ thống: Quản lý và Khách hàng. Mục đích chính của việc nhắm đến hai đối tượng này từ đầu chủ yếu là từ các tính năng được cung cấp cho dự án. Các tính năng gồm:

- Đăng nhập và giao diện riêng biệt cho người quản lý và khách hàng.
- Xác thực và kiểm soát bảo mật để truy cập dữ liệu đăng nhập.
- Quản lý:
 - Quản lý có thể xem thông tin khách hàng hiện tại và thêm khách hàng mới vào cơ sở dữ liệu với một cấp độ thành viên cụ thể. Khách hàng sau khi được thêm vào sẽ tự động nhận được một tài khoản đăng nhập mà không cần mật khẩu, họ có thể thay đổi sau này.
 - Quản lý có thể xem thông tin về cấp độ thành viên và nâng cấp cấp độ thành viên của khách hàng. Sự thay đổi trong thanh toán do nâng cấp cấp độ thành viên sẽ được phản ánh trực tiếp vào số tiền thanh toán của khách hàng.
 - Quản lý có thể xem thông tin thanh toán và chấp nhận thanh toán từ phía khách hàng, điều này sẽ cập nhật số tiền thanh toán của khách hàng trong cơ sở dữ liệu.
 - Quản lý có thể xem lịch học hiện tại và lên lịch một buổi học mới bằng cách thêm thông tin vào cơ sở dữ liệu.
- Khách hàng:
 - Khách hàng có thể xem thông tin cá nhân của mình.
 - Khách hàng có thể thay đổi mật khẩu tài khoản của mình
 - Khách hàng có thể xem thông tin thanh toán của mình và chi tiết về thanh toán gần đây.
 - Khách hàng có thể xem thông tin về các lớp học do huấn luyện viên tổ chức và đăng ký tham gia các lớp học.

Nhưng điều đó không có nghĩa là những tính năng này cũng không gặp phải một số hạn chế. Hai hạn chế chính của hệ thống là chỉ có thể áp dụng hệ thống cho một chi nhánh của phòng tập thể dục. Và hệ thống cho phép một người quản lý xử lý việc quản lý toàn bộ phòng tập.

II. Truy vấn SQL:

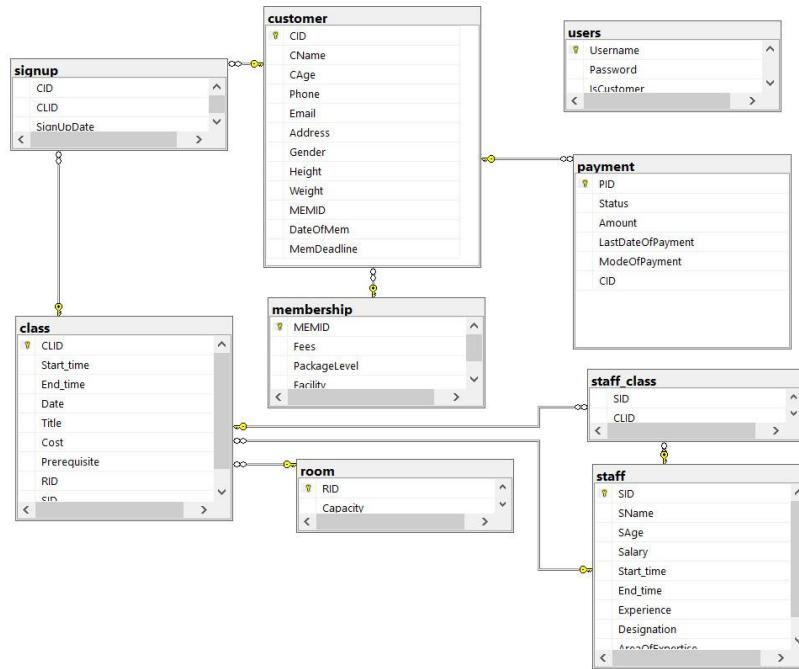


Figure 2.1. Mô hình quan hệ của hệ thống quản lý phòng gym

Để có thể tạo một cơ sở dữ liệu hoàn chỉnh, chúng tôi quyết định tạo một mô hình quan hệ (Hình 2.1) nhằm mục đích giúp việc liệt kê các thực thể và thuộc tính có thể có trong bài viết trở nên dễ dàng hơn. Tóm lại, trong phần này, chúng ta sẽ sử dụng SQL để mô tả bảng, dữ liệu bảng, truy vấn và trình kích hoạt có trong dự án này. Đối với phần này, cơ sở dữ liệu sẽ có tên là GYMMANAGEMENT.

1. Table users

```
CREATE TABLE users (
    Username varchar(255) PRIMARY KEY,
    Password varchar(255) NOT NULL,
    IsCustomer int DEFAULT '1'
);
```

Figure 2.1. Bảng SQL dành cho users

Các trường là: Users(Tên đăng nhập), Password (Mật khẩu), IsCustomer (Loại người dùng - khách hàng hoặc không).

- Username: Tên đăng nhập.
- Password: Mật khẩu.
- IsCustomer: Trường xác định xem người dùng có phải là khách hàng hay không.

2.Table room

```
CREATE TABLE room (
    RID int NOT NULL PRIMARY KEY,
    Capacity int
);
```

Figure 2.2.Bảng SQL dành cho room

Trong bảng này, bảng room chỉ mô tả thông tin về phòng trong phòng tập. Các trường cho bảng và mô tả là: RID (ID phòng), Capacity(Sức chứa).

- RID: Khóa chính của bảng, đại diện cho ID phòng.
- Capacity: Sức chứa của phòng.

3.Table class:

```
CREATE TABLE class (
    CLID int NOT NULL PRIMARY KEY IDENTITY(1,1),
    Start_time TIME,
    End_time TIME,
    Date date NOT NULL,
    Title varchar(255),
    Cost int ,
    Prerequisite varchar(255) ,
    RID int NOT NULL,
    CONSTRAINT FK_Class_Room FOREIGN KEY (RID) REFERENCES room(RID),
);
```

Figure 2.3. Bảng SQL dành cho class

Bảng class được tạo ra nhằm mục đích giới thiệu cho khách hàng lớp học để họ tham dự. Tất cả các trường trong bảng: CLID (ID lớp), ClassTimings (Thời gian lớp), Date (Ngày lớp), Title (Tên lớp), Cost (Chi phí), Prerequisite (Điều kiện tiên quyết), RID (ID phòng).

- CLID: khóa chính cho bảng này, nó đại diện cho ID của lớp.
- ClassTimings: Giờ học

- Date: Ngày học
- Title: Tiêu đề của lớp học
- Cost: Chi phí tham gia lớp học
- Prerequisite: Điều kiện tiên quyết để tham gia lớp học
- RID: Khóa ngoại để liên kết với phòng bảng.

4.Table membership

Bảng này mô tả các gói đào tạo và chi phí của chúng. Bảng này có một trường bao gồm: MEMID (ID gói tập luyện), Fees (Chi phí), PackageLevel (Loại gói), Facility (Tiện ích).

- MEMID: Khóa chính của bảng, đại diện cho ID của gói thành viên.
- Fees: Phí thành viên trọn gói.
- PackageLevel: Cấp độ của gói dịch vụ.
- Facility: Tiện ích bao gồm trong gói.

```
CREATE TABLE membership (
    MEMID int NOT NULL PRIMARY KEY IDENTITY(1,1),
    Fees int ,
    PackageLevel varchar(50),
    Facility varchar(255)
);
```

Figure 2.4. Bảng SQL dành cho membership

5.Table customer

Để lưu thông tin khách hàng. Và trường cho bảng này bao gồm: CID (ID khách hàng), CName (Tên khách hàng), CAge (Tuổi), Phone (Số điện thoại), Email (Email), Address (Địa chỉ), Gender (Giới tính), Height (Chiều cao) , Weight (Trọng lượng), MEMID (ID gói tập luyện), DateOfMem (Ngày đăng ký), MemDeadline (Hạn thanh toán).

```
CREATE TABLE customer (
    CID int NOT NULL PRIMARY KEY IDENTITY(1,1),
    CName varchar(255) NOT NULL,
    CAge int ,
    Phone varchar(255) NOT NULL,
    Email varchar(255),
    Address varchar(255),
    Gender varchar(255),
    Height int ,
    Weight int ,
    MEMID int NOT NULL,
    DateOfMem date NOT NULL,
    MemDeadline int NOT NULL,
    CONSTRAINT FK_Customer_Membership FOREIGN KEY (MEMID) REFERENCES membership(MEMID)
);
```

Figure 2.5. Bảng SQL dành cho customer

Đây là mô tả cho bảng:

- CID: Khóa chính, được gọi là ID của khách hàng
- CName: Tên khách hàng
- CAge: Tuổi của khách hàng
- Phone: Số điện thoại của khách hàng
- Email: Địa chỉ email của khách hàng
- Address: Địa chỉ nhà của khách hàng
- Gender: Giới tính của khách hàng
- Height: Chiều cao của khách hàng
- Weight: Trọng lượng khách hàng
- MEMID: Khóa ngoại cho bảng Thành viên
- DateOfMem: Ngày tham gia
- MemDeadline: Thời hạn thành viên

6.Table payment

```
CREATE TABLE payment (
    PID int NOT NULL PRIMARY KEY IDENTITY(1,1),
    Status varchar(255) ,
    Amount int DEFAULT NULL,
    LastDateOfPayment date DEFAULT NULL ,
    ModeOfPayment varchar(255) ,
    CID int NOT NULL,
    CONSTRAINT FK_Payment_Customer FOREIGN KEY(CID) REFERENCES customer(CID)
);
```

Figure 2.6.Bảng SQL dành cho payment

Đối với bảng này, dùng để ghi lại thông tin thanh toán của khách hàng. Đối với bảng này, trường dành cho bảng này sẽ là: PID (ID thanh toán), Status (Trạng thái thanh toán), Amount (Số tiền), LastDateOfPayment (Ngày thanh toán cuối cùng), ModeOfPayment (Phương thức thanh toán), CID (ID khách hàng). Và mô tả của bảng:

- PID: Khóa chính của bảng, thể hiện ID của giao dịch thanh toán.
- Status: Trạng thái thanh toán.
- Amount: Số tiền thanh toán.

- LastDateOfPayment: Ngày thanh toán cuối cùng.
- ModeOfPayment: Phương thức thanh toán.
- CID: Khóa ngoại gắn với bảng khách hàng.

7.Table signup

```
CREATE TABLE signup (
  CID int NOT NULL,
  CLID int NOT NULL,
  SignUpDate date ,
  CONSTRAINT FK_Signup_Customer FOREIGN KEY(CID) REFERENCES customer(CID),
  CONSTRAINT FK_Signup_Class FOREIGN KEY(CLID) REFERENCES class(CLID)
);
```

Figure 2.7.Bảng SQL dành cho signup

Bảng tiếp theo, bảng signup dùng để liên kết thông tin đăng ký khách hàng với lớp. Mô tả và các trường cho bảng này là: CID (ID khách hàng), CLID (ID lớp), SignUpDate (Ngày đăng ký).

- CID: Khóa ngoại gắn với bảng khách hàng.
- CLID: Khóa ngoại liên kết với bảng lớp.
- SignUpDate: Ngày đăng ký.

8.Table staff

```

CREATE TABLE staff (
    SID int NOT NULL PRIMARY KEY IDENTITY(1,1),
    SName varchar(255) NOT NULL,
    SAge int ,
    Salary int ,
    Start_time TIME,
    End_time TIME,
    Experience varchar(255),
    Designation varchar(255),
    AreaOfExpertise varchar(255)
);

```

Figure 2.8. Bảng SQL dành cho staff

Bảng staff là để hiển thị thông tin của nhân viên. Trường và mô tả: SID (ID nhân viên), SName (Tên nhân viên), SAge (Tuổi), Salary (Mức lương), Timings (Thời gian làm việc), Experience (Kinh nghiệm), Designation (Vị trí), AreaOfExpertise (Chuyên môn). Và mô tả:

- SID: Khóa chính của bảng, đại diện cho ID của nhân viên.
- SName: Tên nhân viên.
- SAge: Tuổi của nhân viên.
- Salary: Lương của nhân viên.
- Timings: Thời gian làm việc của nhân viên.
- Experience: Kinh nghiệm của nhân viên.
- Designation: Chức vụ của nhân viên.
- AreaOfExpertise: Lĩnh vực chuyên môn của nhân viên.

9.Table staff class

```

CREATE TABLE staff_class (
    SID int NOT NULL,
    CLID int NOT NULL,
    constraint fk_Staff_class_Class foreign key(CLID) references class(CLID),
    constraint fk_Staff_class_Staff foreign key(SID) references staff(SID)
);

```

Figure 2.9.Bảng SQL dành cho staff_class

Trong bảng staff_class dùng để Liên kết nhân viên với lớp họ dạy. Các trường đó là: SID (ID nhân viên), CLID (ID lớp học). Mô tả cho bảng:

- SID: Khóa ngoại liên kết với bảng nhân viên.
- CLID: Khóa ngoại liên kết với bảng lớp.

10.Trigger insertinuseroninsertcustomer:

Mục đích: Trigger được kích hoạt sau khi dữ liệu được chèn vào bảng khách hàng. Nó tự động thêm người dùng mới vào bảng người dùng với thông tin tương ứng từ bảng khách hàng.

Hành động: Khi một hàng mới được chèn vào khách hàng, trình kích hoạt sẽ thêm người dùng mới vào bảng người dùng, sử dụng tên khách hàng từ bảng khách hàng.

```

CREATE TRIGGER insertinuseroninsertcustomer
ON customer
AFTER INSERT
AS
BEGIN
    INSERT INTO users (Username, Password, IsCustomer)
    SELECT Name, '', 1
    FROM inserted;
END;
GO

```

Figure 3.1.Tạo Trigger cho insertinuseroninsertcustomer

11. Trigger updatepaymentoninsertcustomer:

Mục đích: Khi một khách hàng mới được thêm vào bảng khách hàng, trình kích hoạt này sẽ tạo một khoản thanh toán mới trong bảng thanh toán dựa trên thông tin gói thành viên và thời hạn đã chọn.

Hành động: Sau khi chèn một hàng mới vào khách hàng, trình kích hoạt sẽ tạo một giao dịch thanh toán mới trong bảng thanh toán. Thông tin bao gồm trạng thái 'Đang chờ xử lý', số tiền được tính dựa trên gói thành viên và thời hạn được chọn từ bảng thành viên.

```
CREATE TRIGGER updatepaymentoninsertcustomer
ON customer
AFTER INSERT
AS
BEGIN
    INSERT INTO payment (Status, Amount, CID)
    SELECT 'Pending', (SELECT Fees FROM membership WHERE MEMID = inserted.MEMID) * inserted.MemDeadline, inserted.CID
    FROM inserted;
END;
GO
```

Figure 3.2.Tạo Trigger cho updatepaymentoninsertcustomer

12. Trigger updatepaymentonmembershipupgrade:

Mục đích: Khi khách hàng được cập nhật gói thành viên mới, trình kích hoạt này sẽ cập nhật giao dịch thanh toán trong bảng thanh toán.

Hành động: Khi thông tin khách hàng được cập nhật với tư cách thành viên mới, trigger sẽ cập nhật số tiền giao dịch thanh toán trong bảng thanh toán. Cập nhật dựa trên những thay đổi về chi phí gói thành viên.

```
CREATE TRIGGER updatepaymentonmembershipupgrade
ON customer
AFTER UPDATE
AS
BEGIN
    UPDATE payment
    SET Amount = Amount + (SELECT Fees FROM membership WHERE MEMID = inserted.MEMID) - (SELECT Fees FROM membership WHERE MEMID = deleted.MEMID)
    FROM payment
    INNER JOIN inserted ON inserted.CID = payment.CID
    INNER JOIN deleted ON inserted.CID = deleted.CID;
END;
GO
```

Figure 3.3.Tạo Trigger cho updatepaymentonmembershipupgrade

13. Trigger updatestatus (missing in sample data):

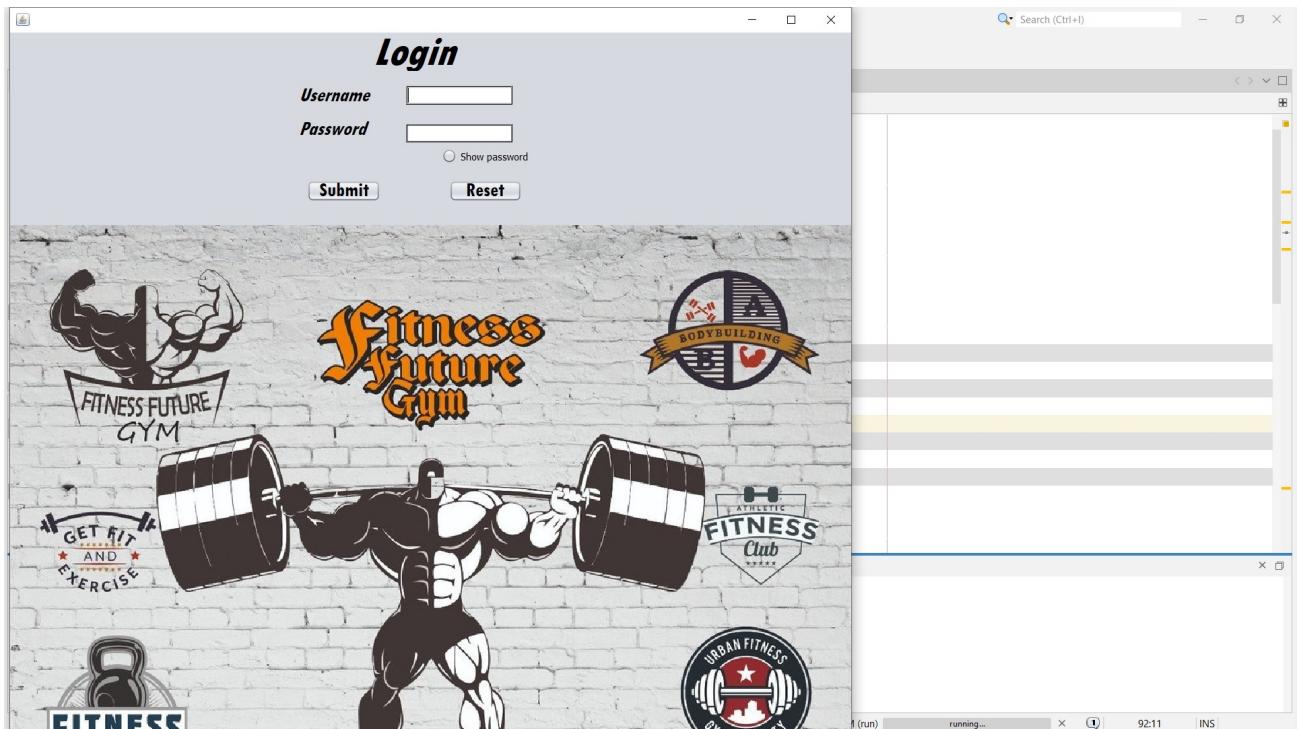
Mục đích: Trigger dùng để cập nhật trạng thái giao dịch thanh toán trong bảng thanh toán trước khi có thay đổi.

Hành động: Nếu số tiền trong giao dịch thanh toán là 0, trình kích hoạt này sẽ tự động cập nhật trạng thái thanh toán thành 'Xóa' trước khi thực hiện cập nhật dữ liệu.

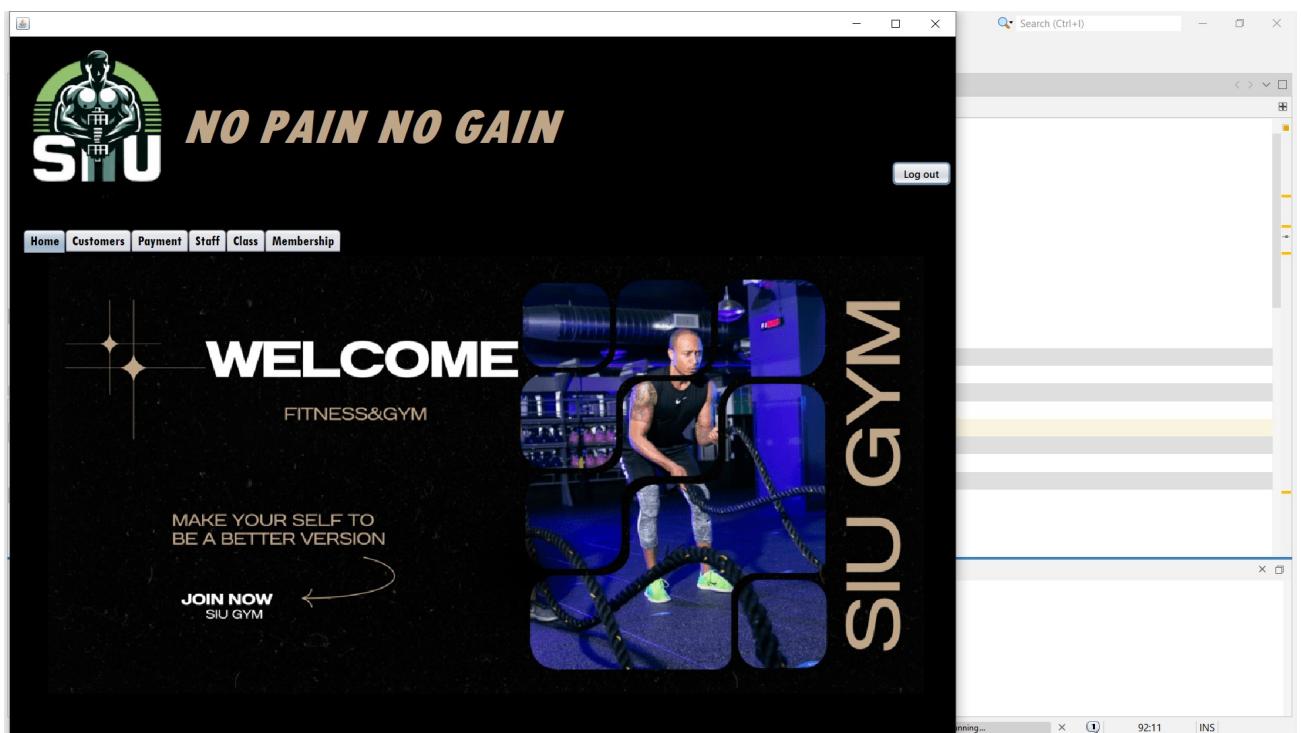
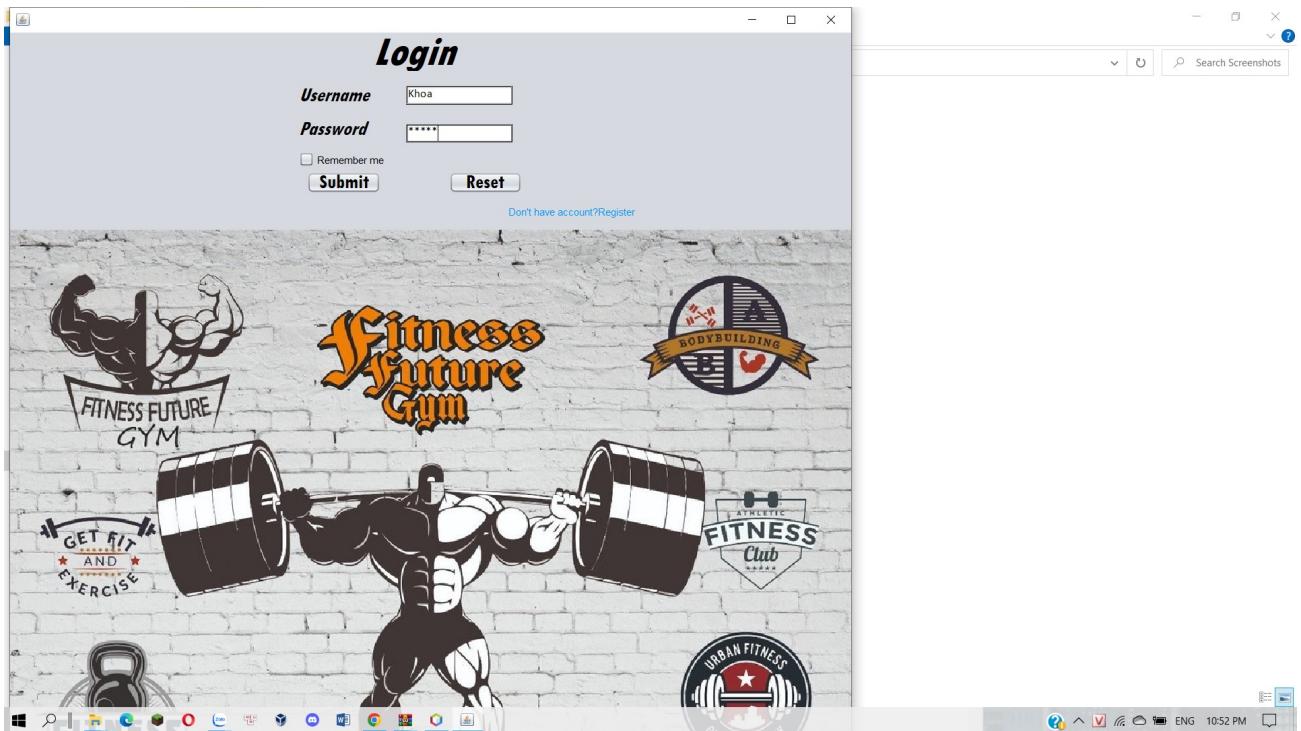
```
CREATE TRIGGER updatestatus
ON payment
AFTER UPDATE
AS
BEGIN
    UPDATE payment
    SET Status = 'Clear'
    WHERE Amount = 0;
END;
GO
```

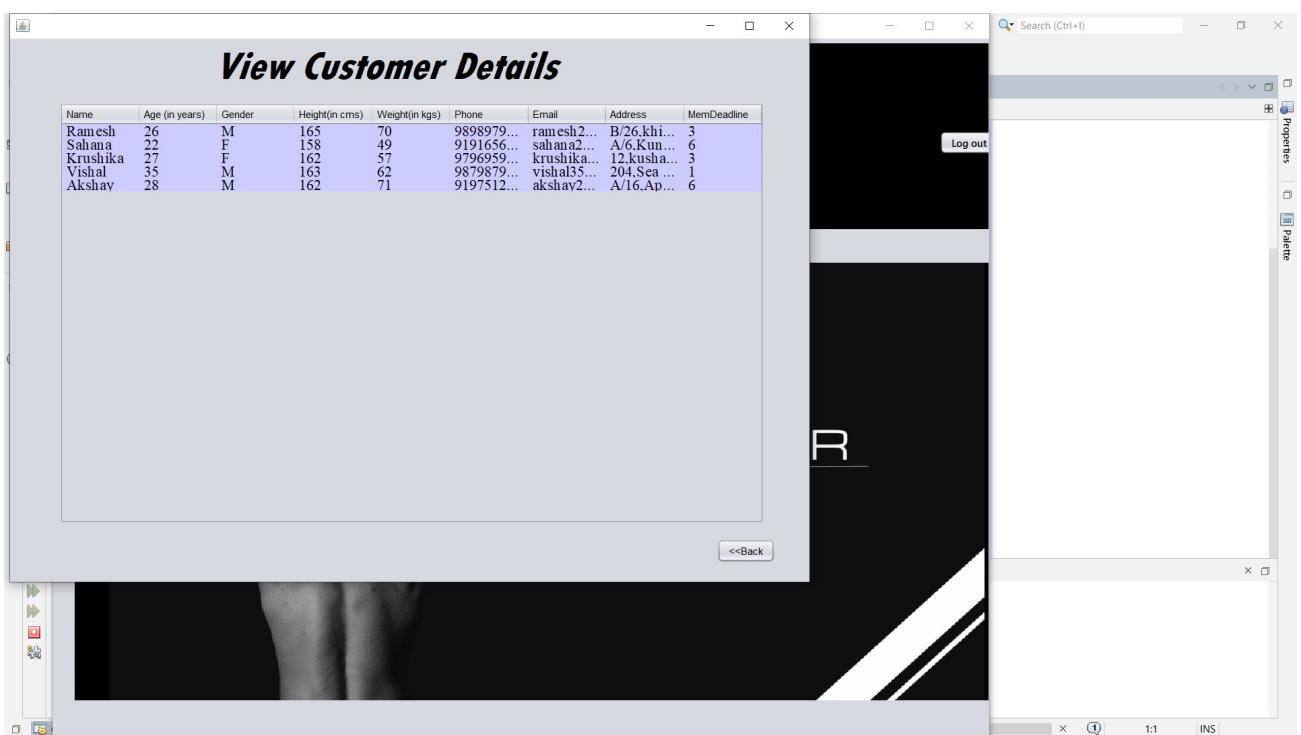
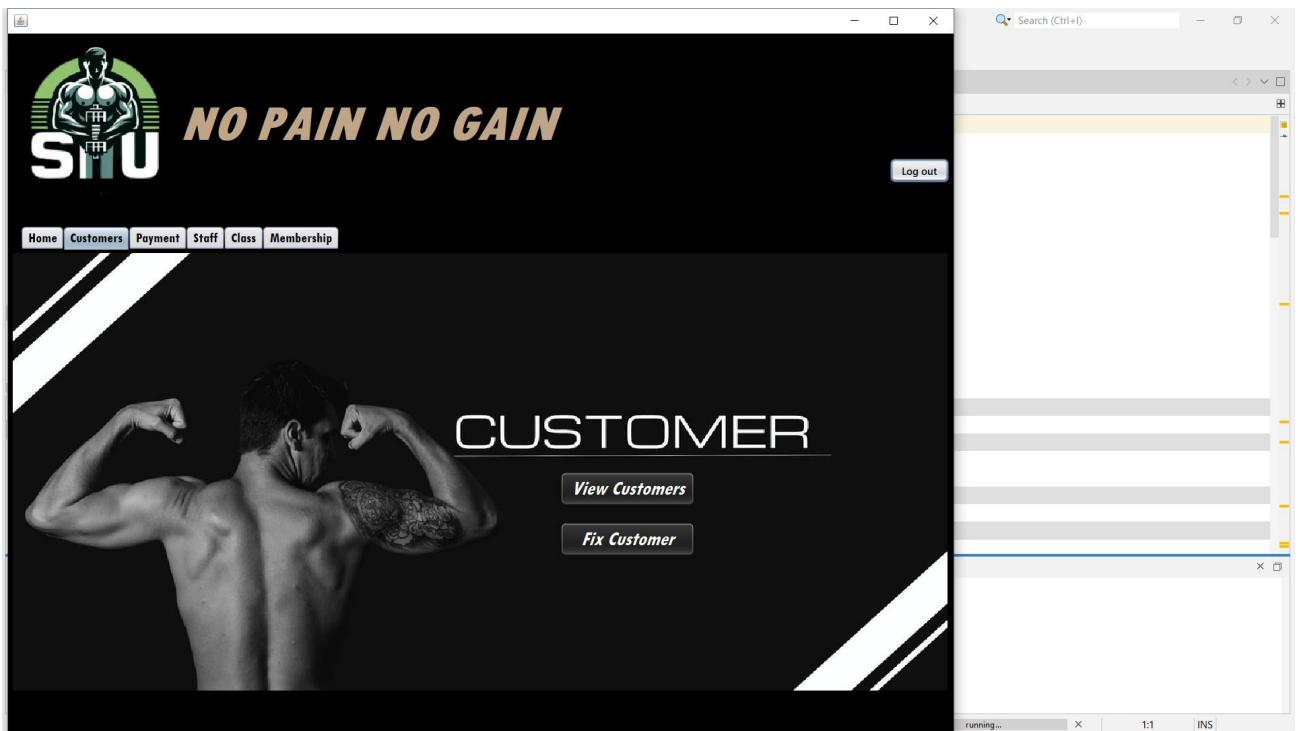
Figure 3.4. Tạo Trigger cho updatestatus

III. Một số hình ảnh về kết quả



1. Đầu tiên ta đăng nhập bằng quyền quản lý



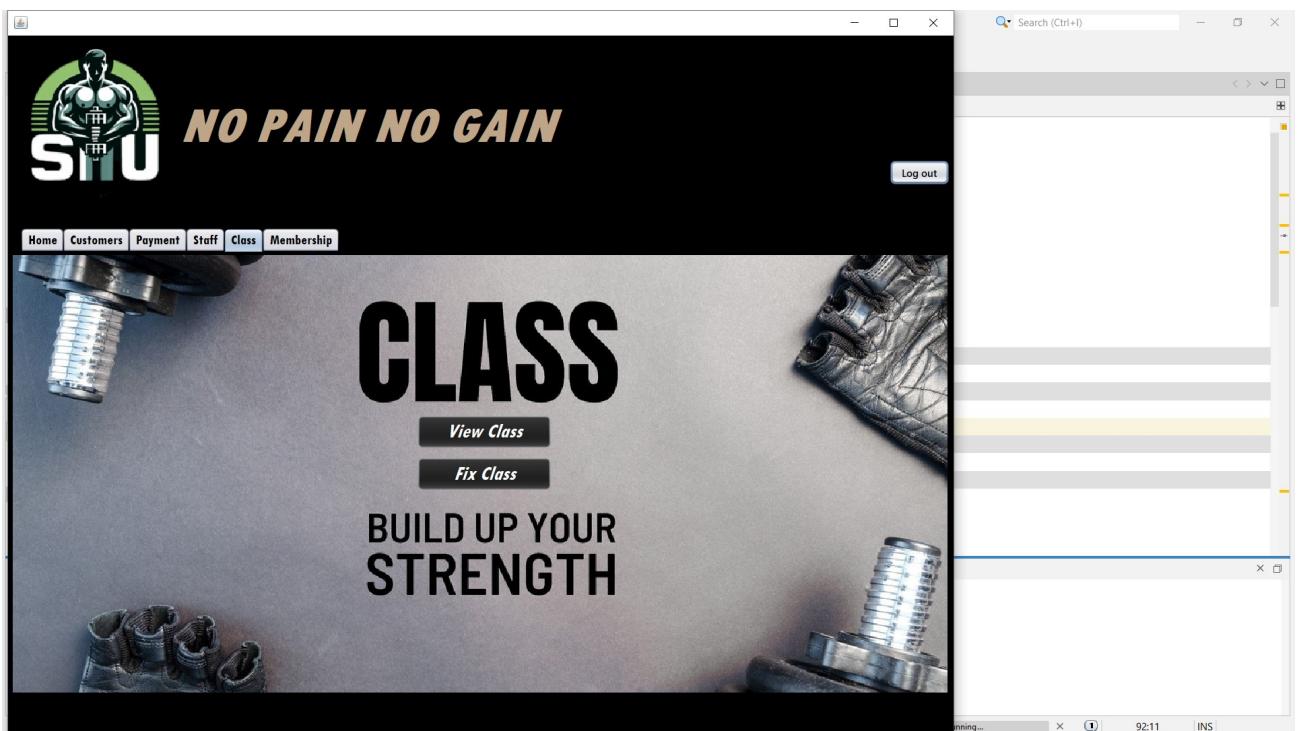
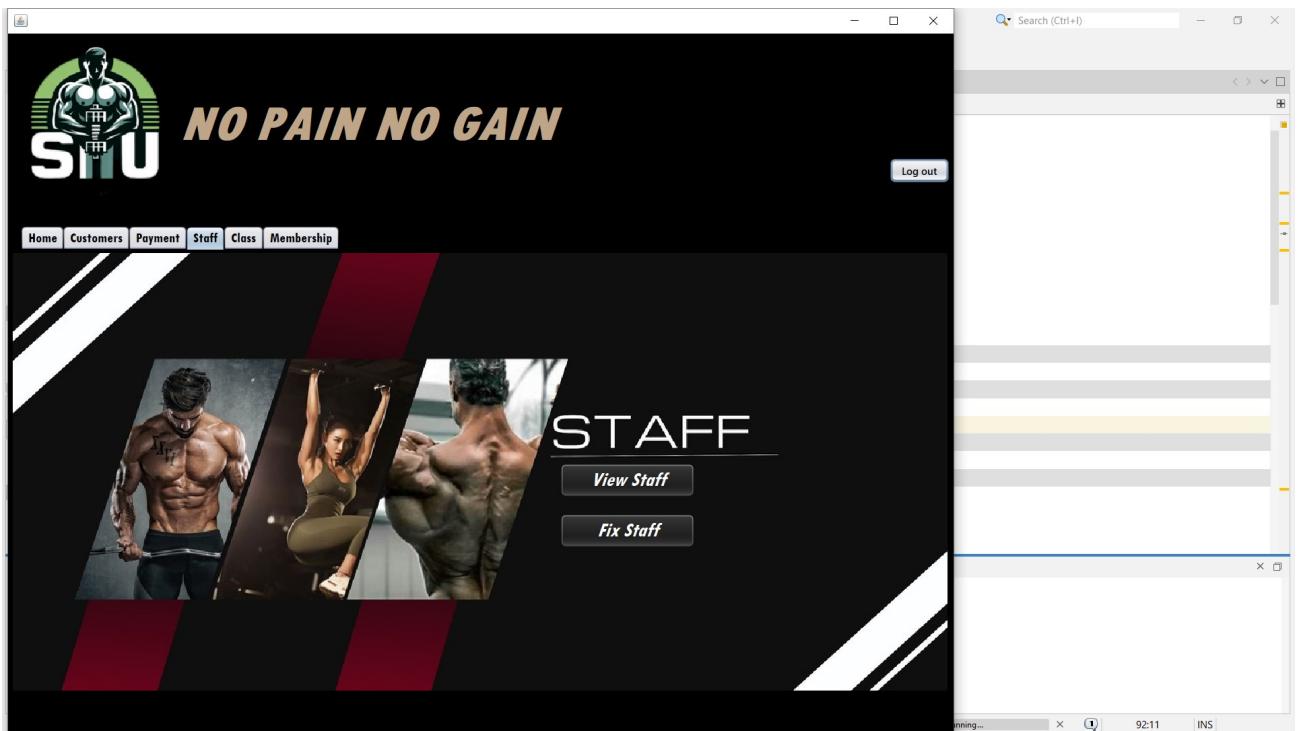


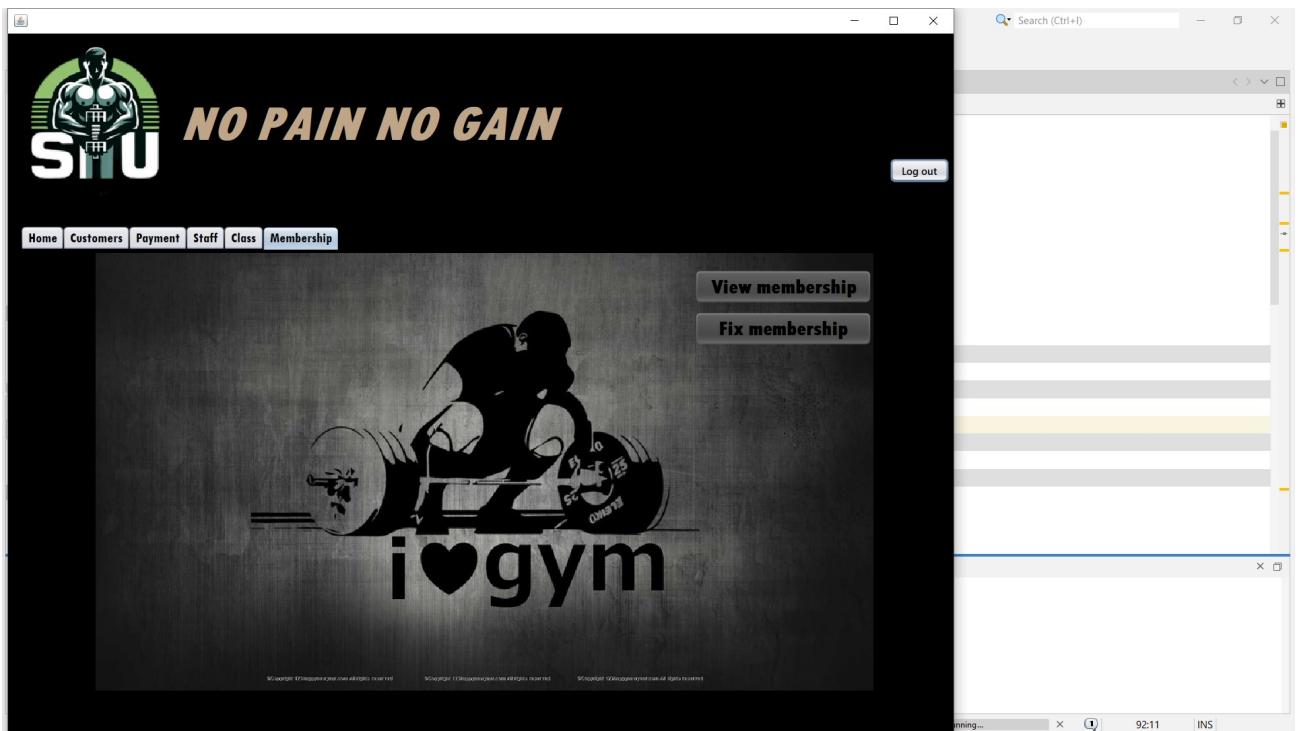
Add Customer Details

Name	<input type="text" value="Akshay"/>
Age	<input type="text" value="28"/>
Address	<input type="text" value="A/16,Aparna apartment,kurla(w)"/>
Gender	<input type="text" value="M"/>
Email Id	<input type="text" value="akshay28@rediffmail.com"/>
Contact No	<input type="text" value="9197512365"/>
Height	<input type="text" value="162"/>
Weight	<input type="text" value="71"/>
Package Level	<input type="text" value="Bronze"/>
Membership Validity	<input type="text" value="6"/>

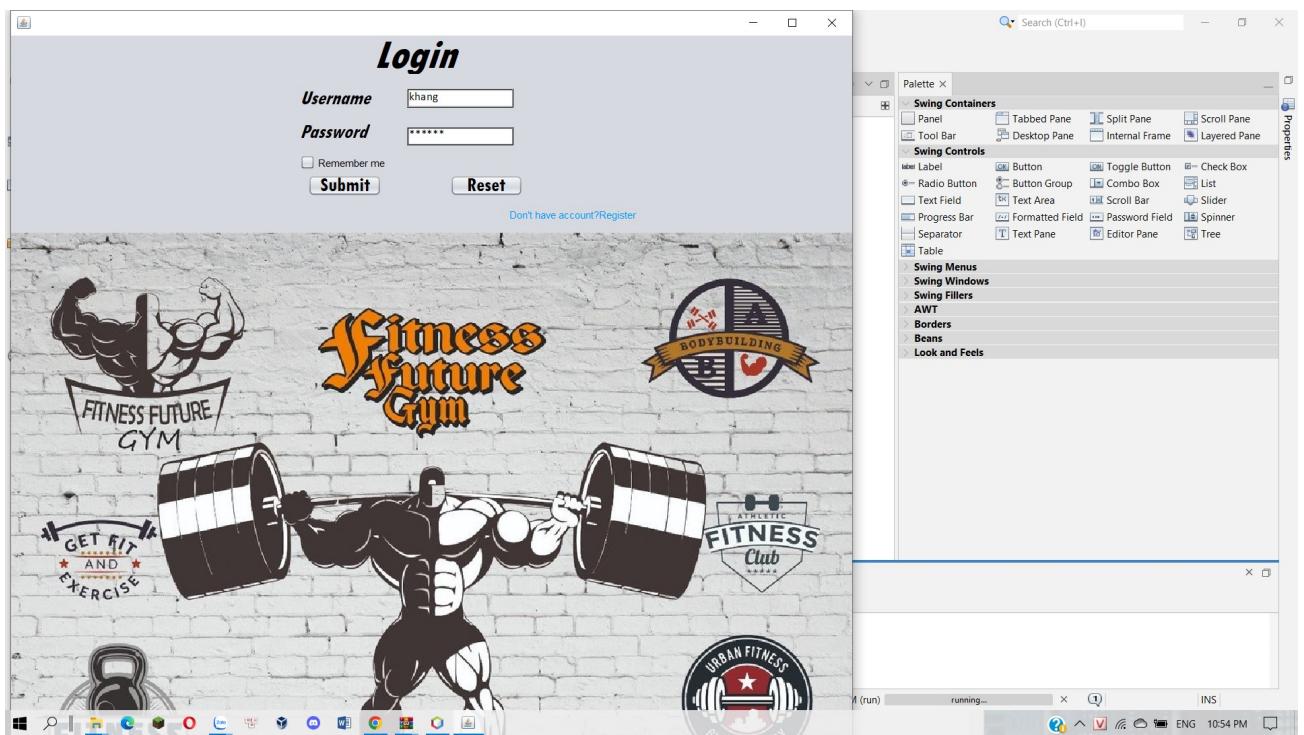
SMU NO PAIN NO GAIN

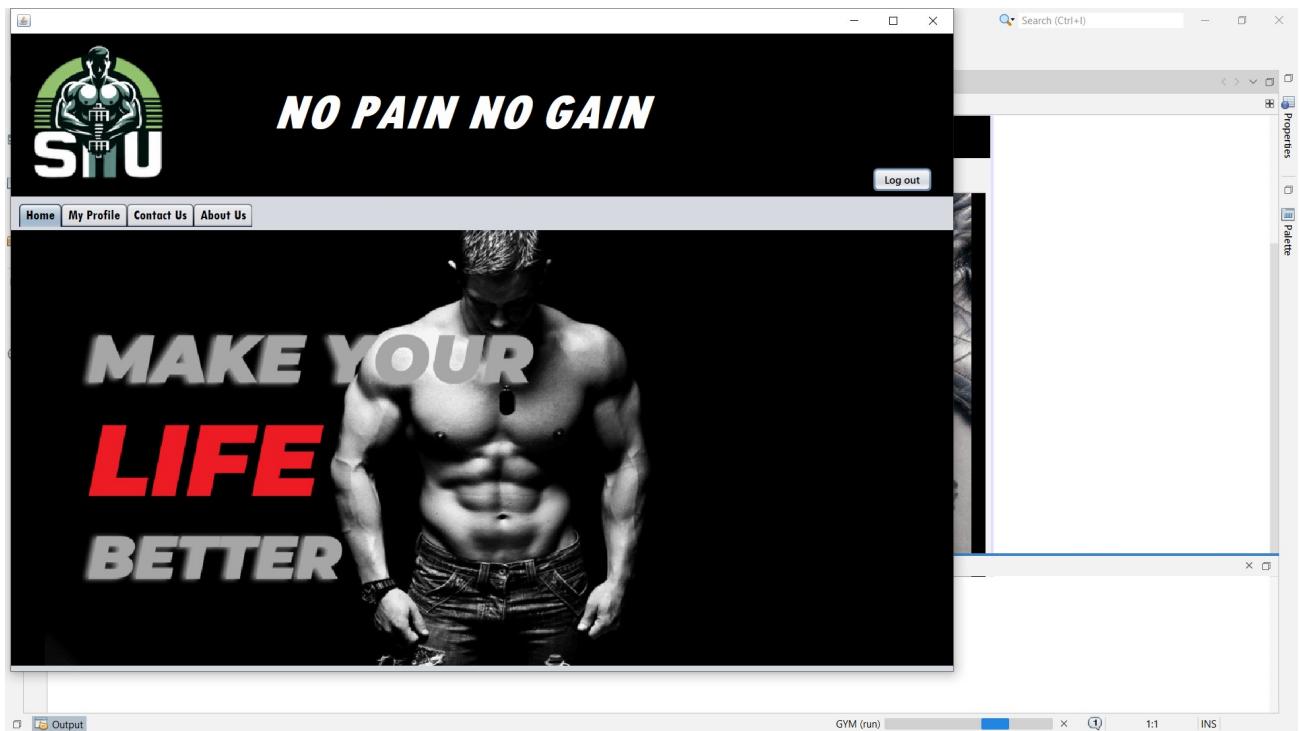
PAYMENT



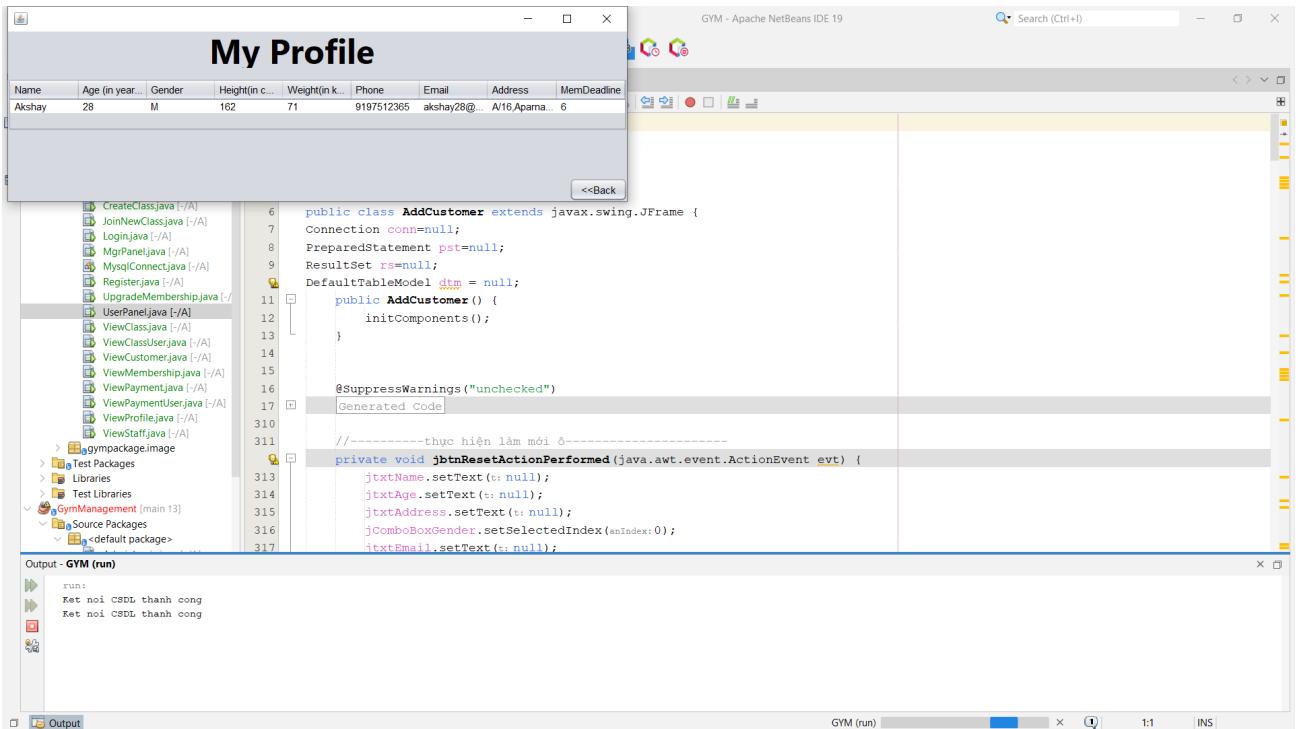


2. Đăng nhập bằng quyền user

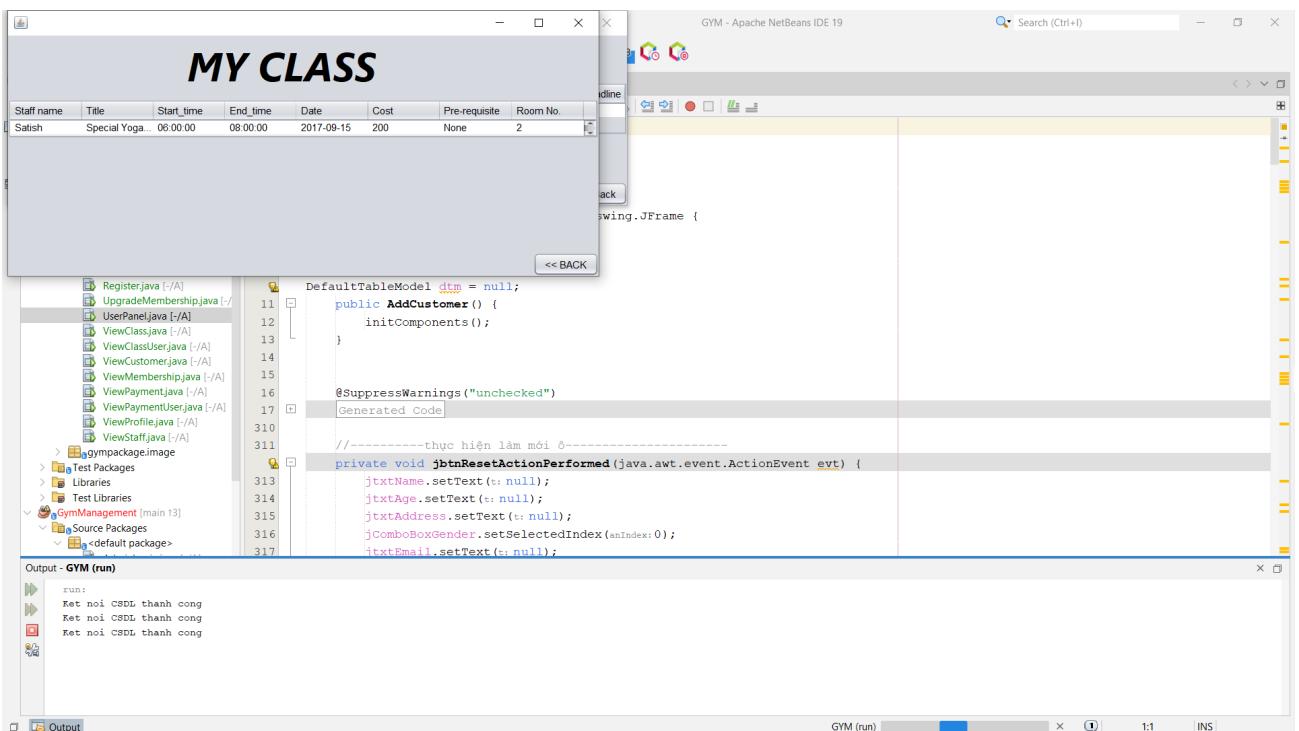




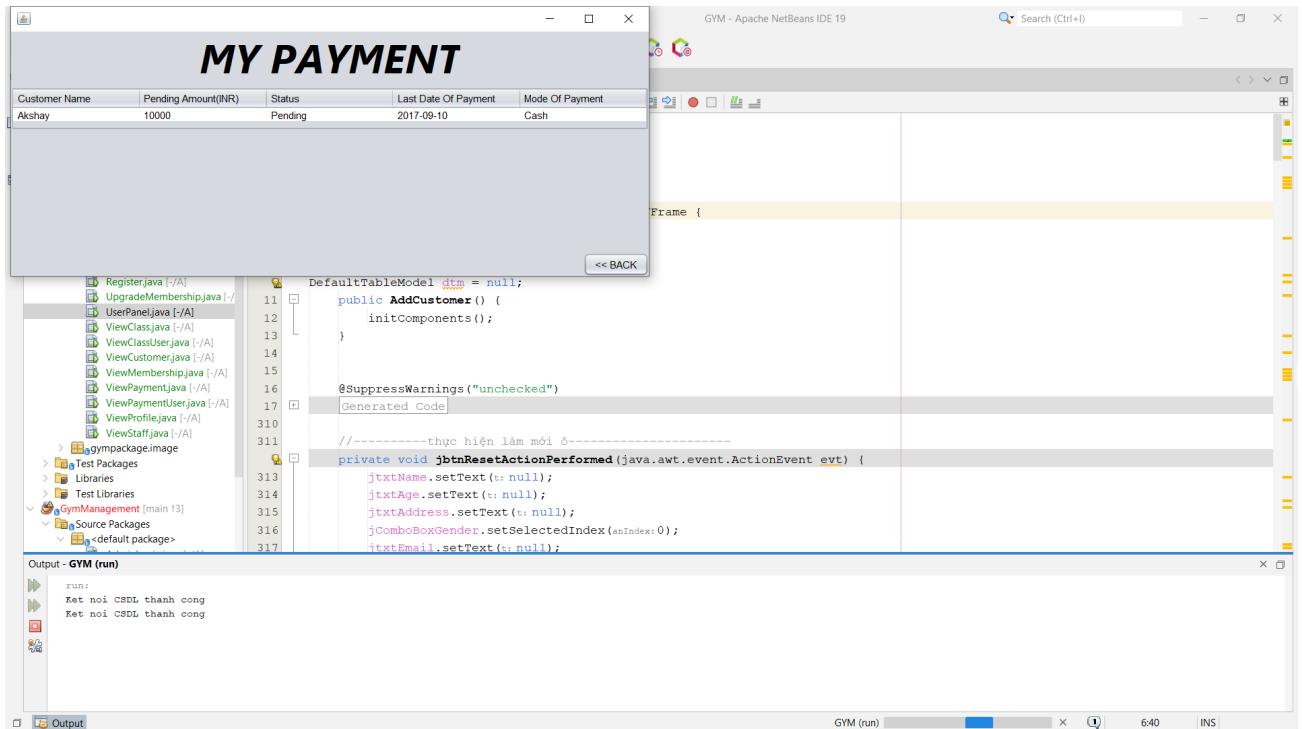
Xem thông tin cá nhân



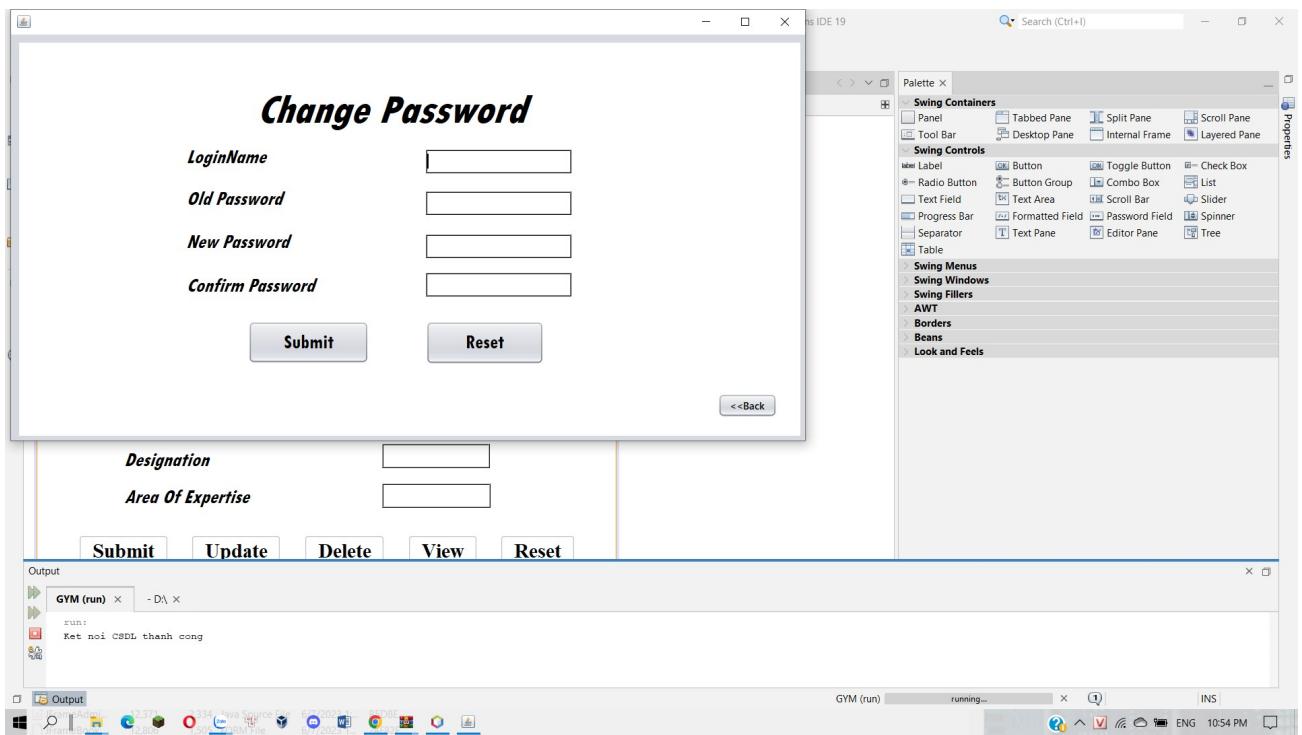
Xem thông tin lớp đang tham gia

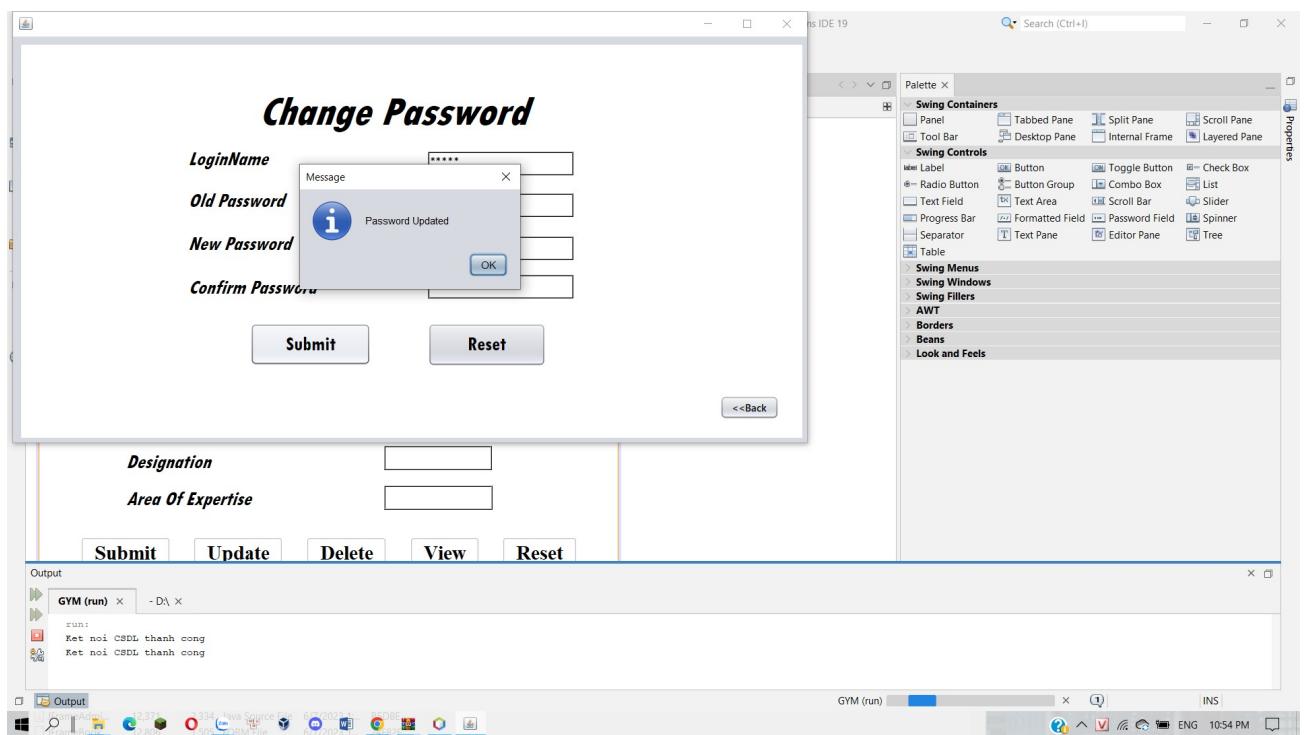
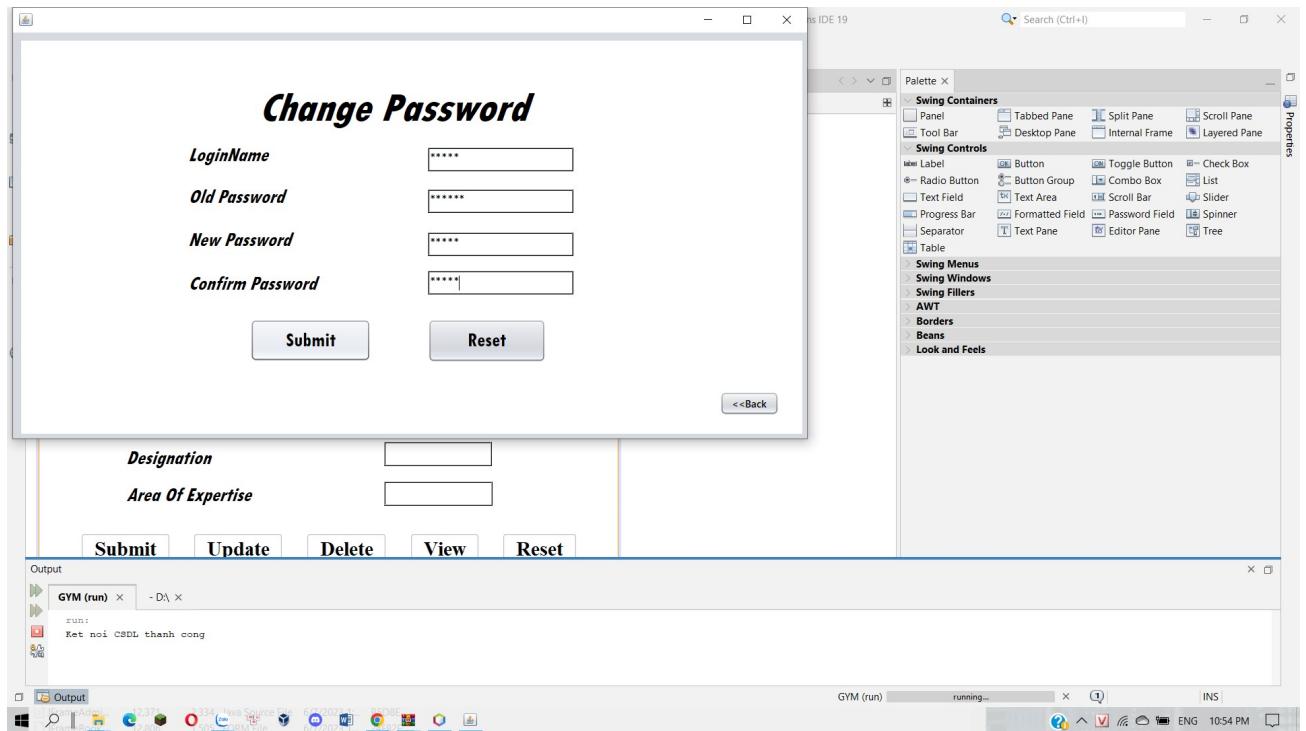


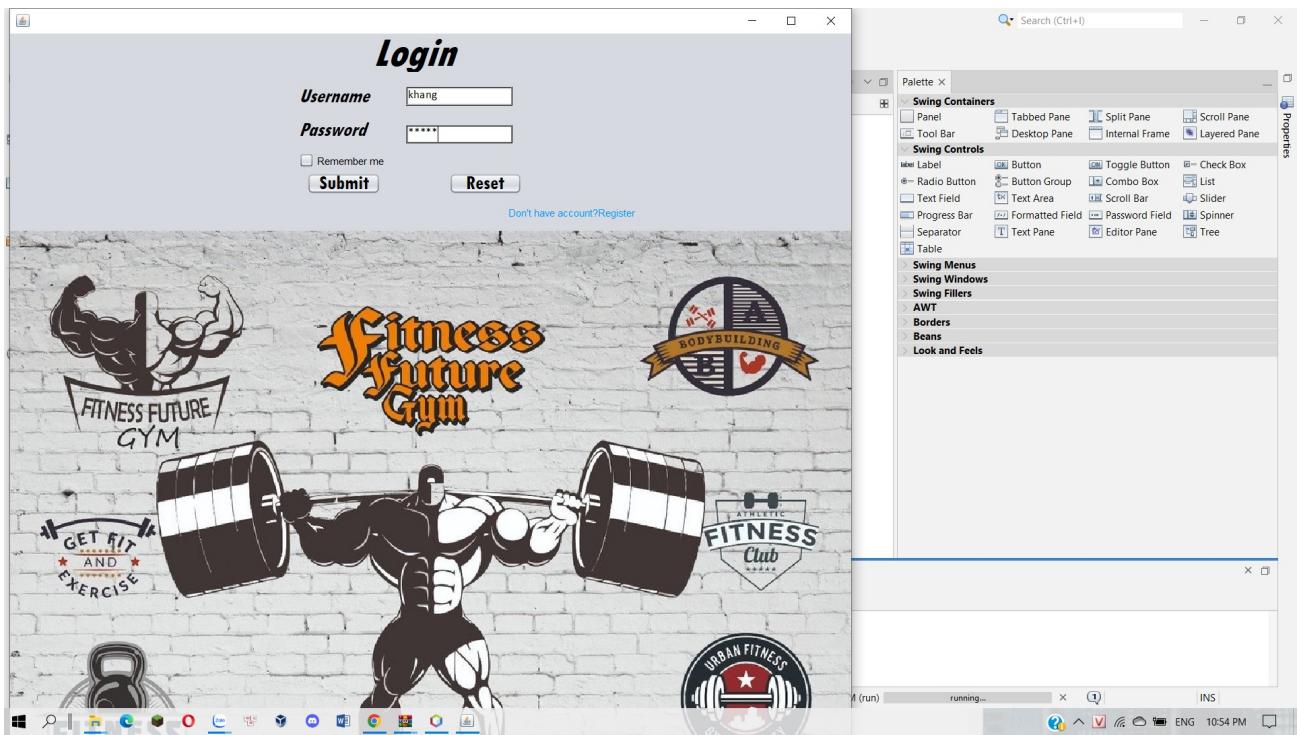
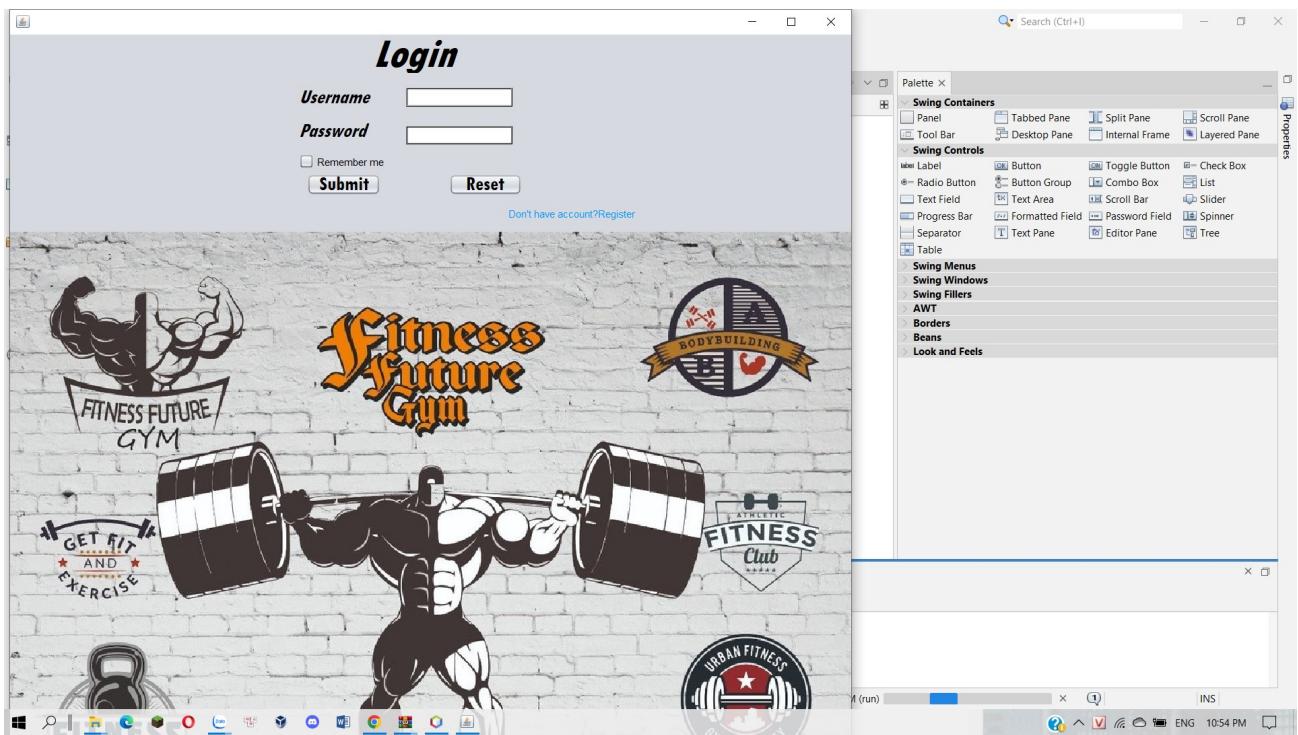
Xem thông tin thanh toán

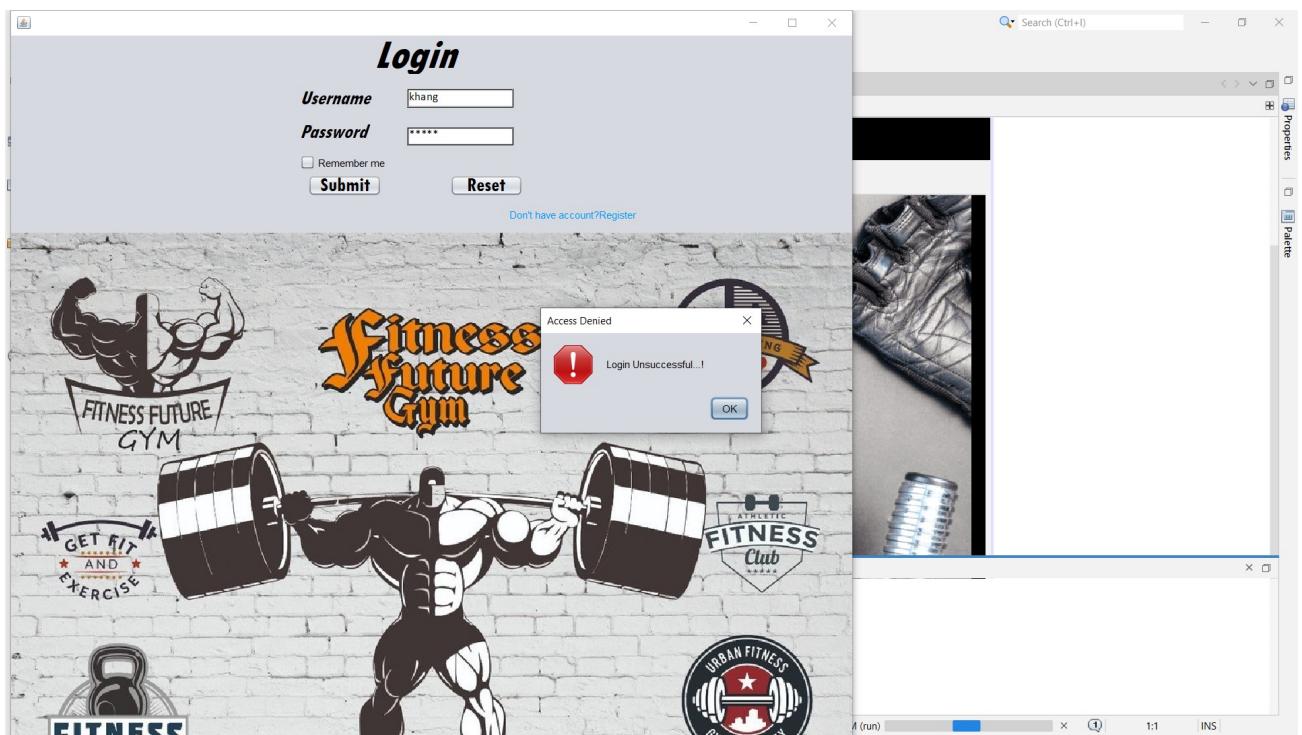
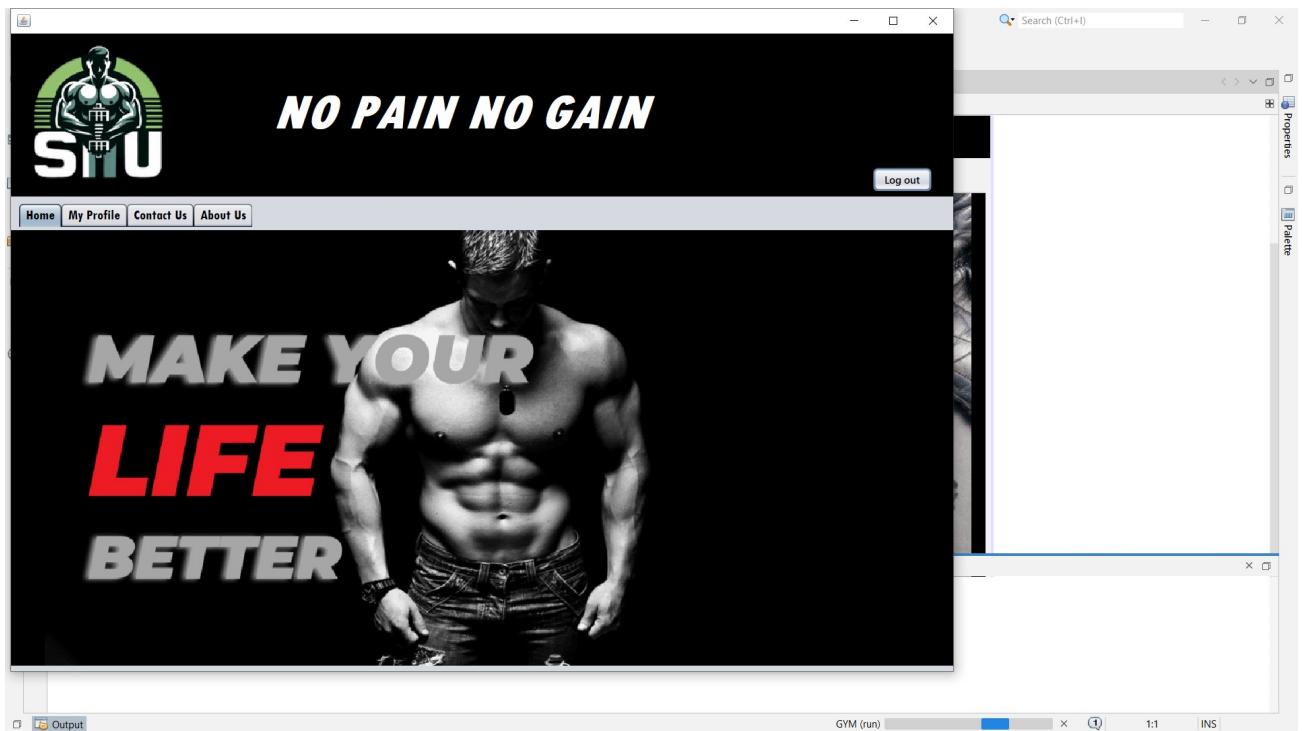


Thay đổi password cho tài khoản của khách hàng

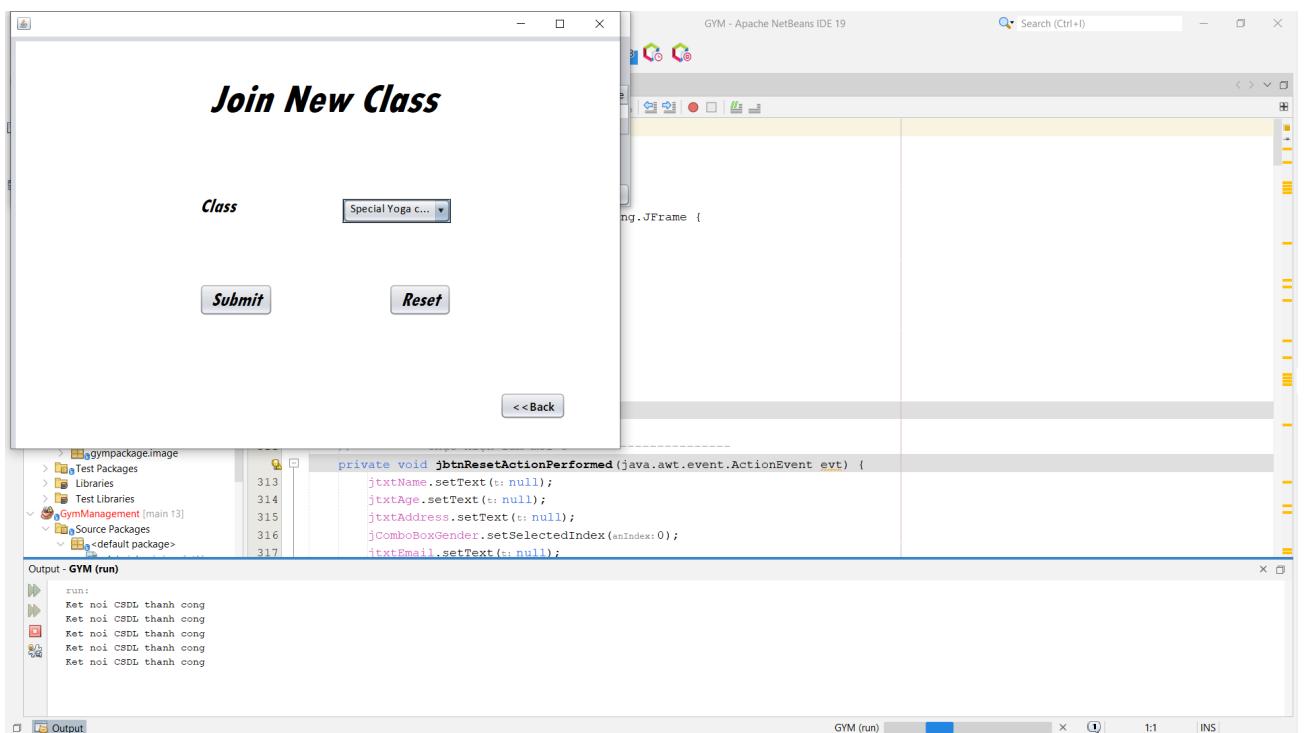
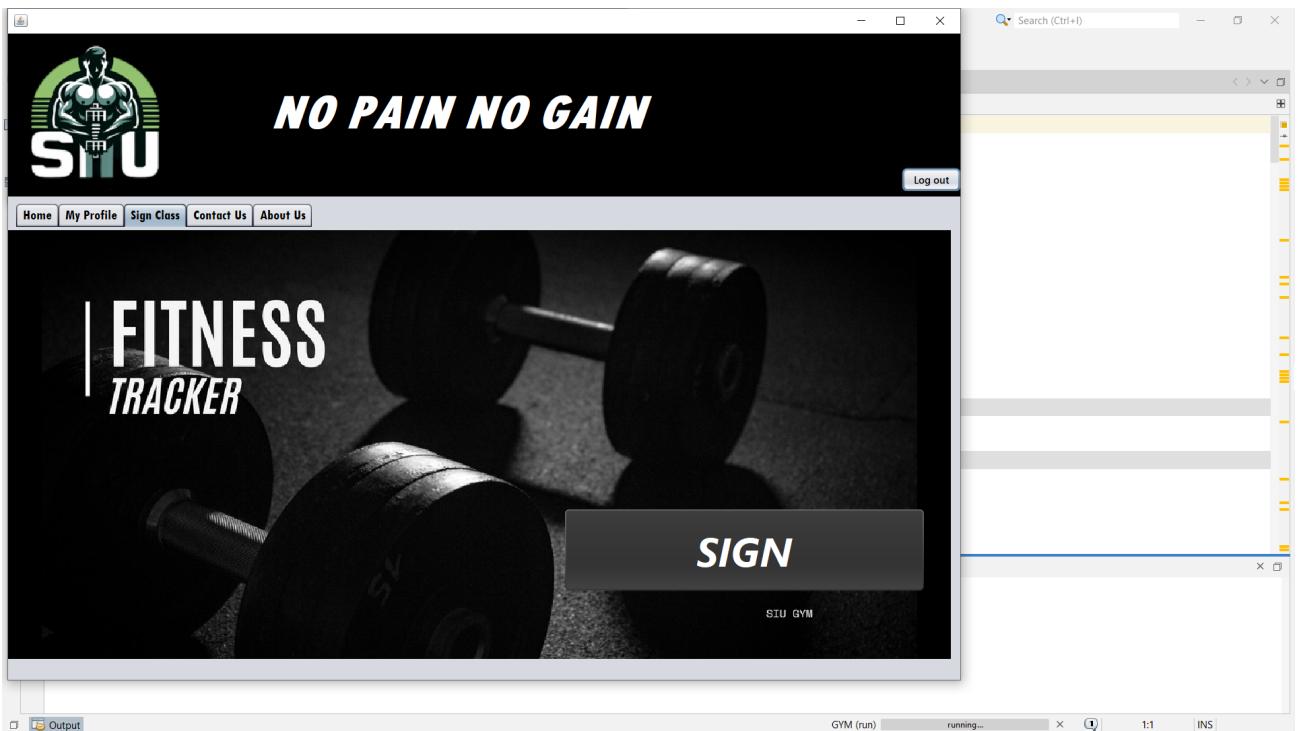


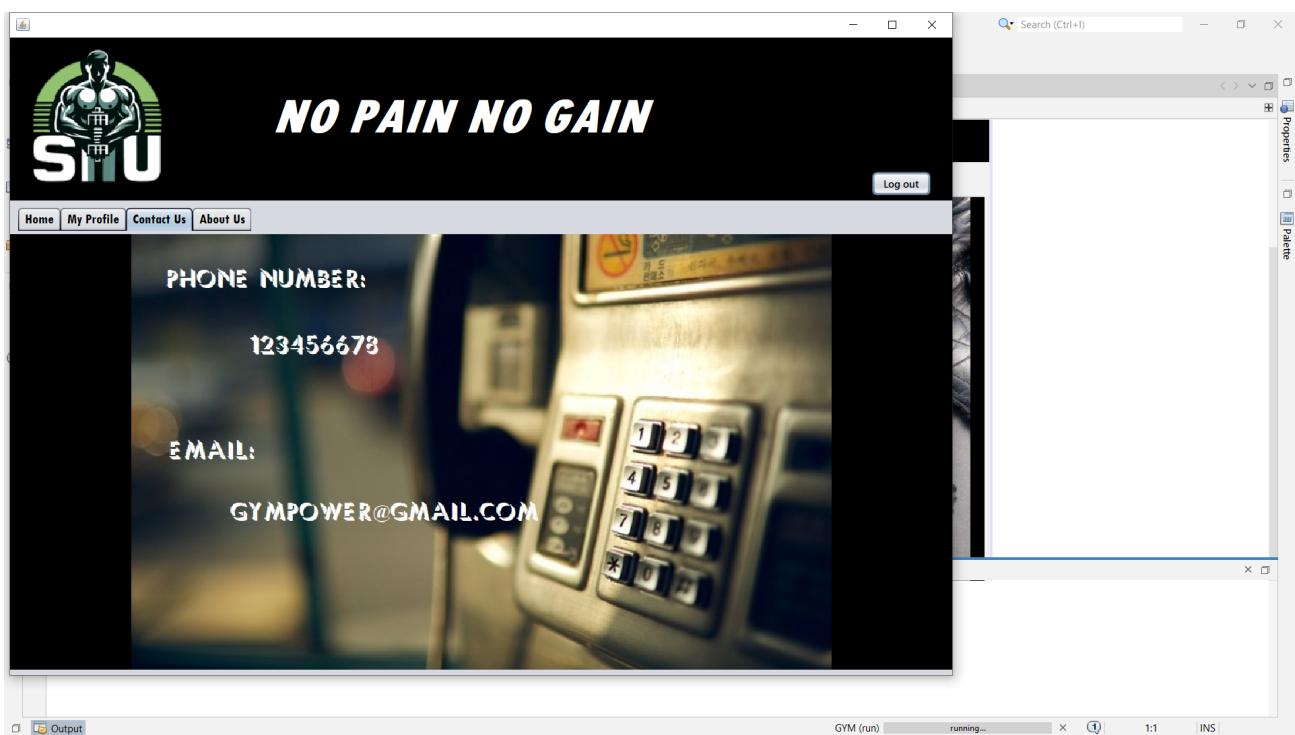
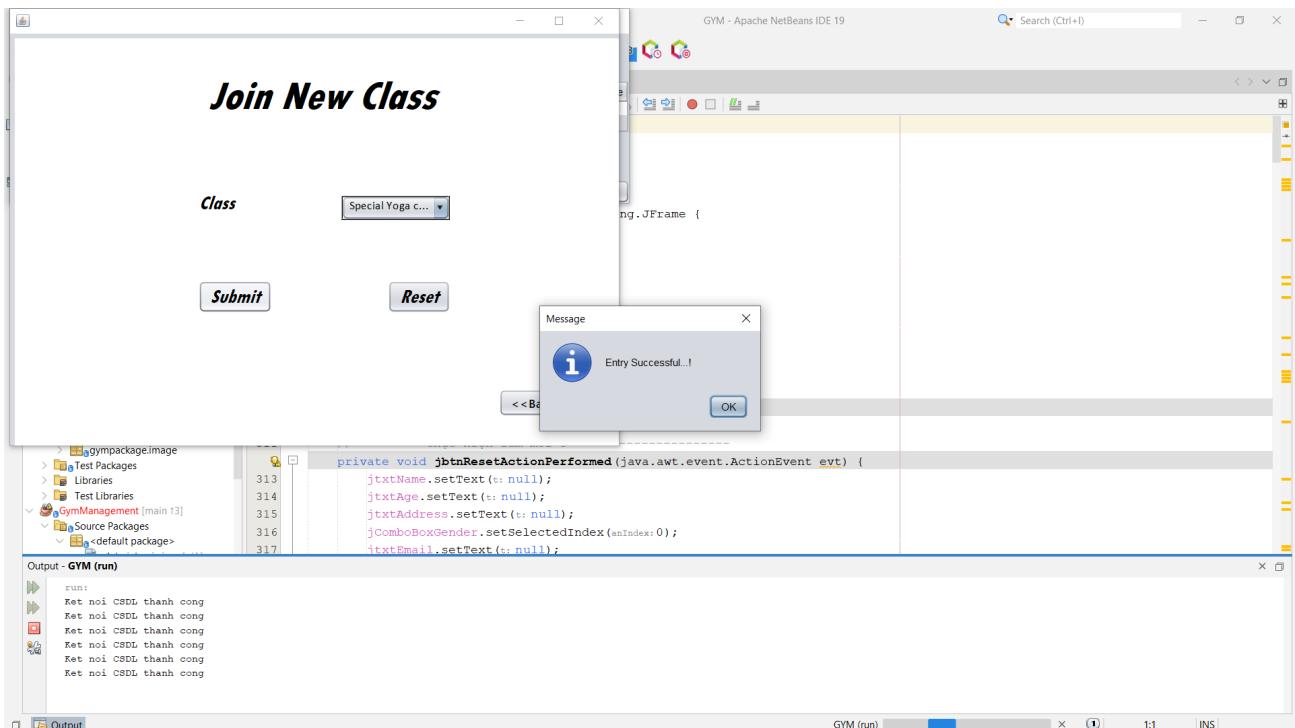


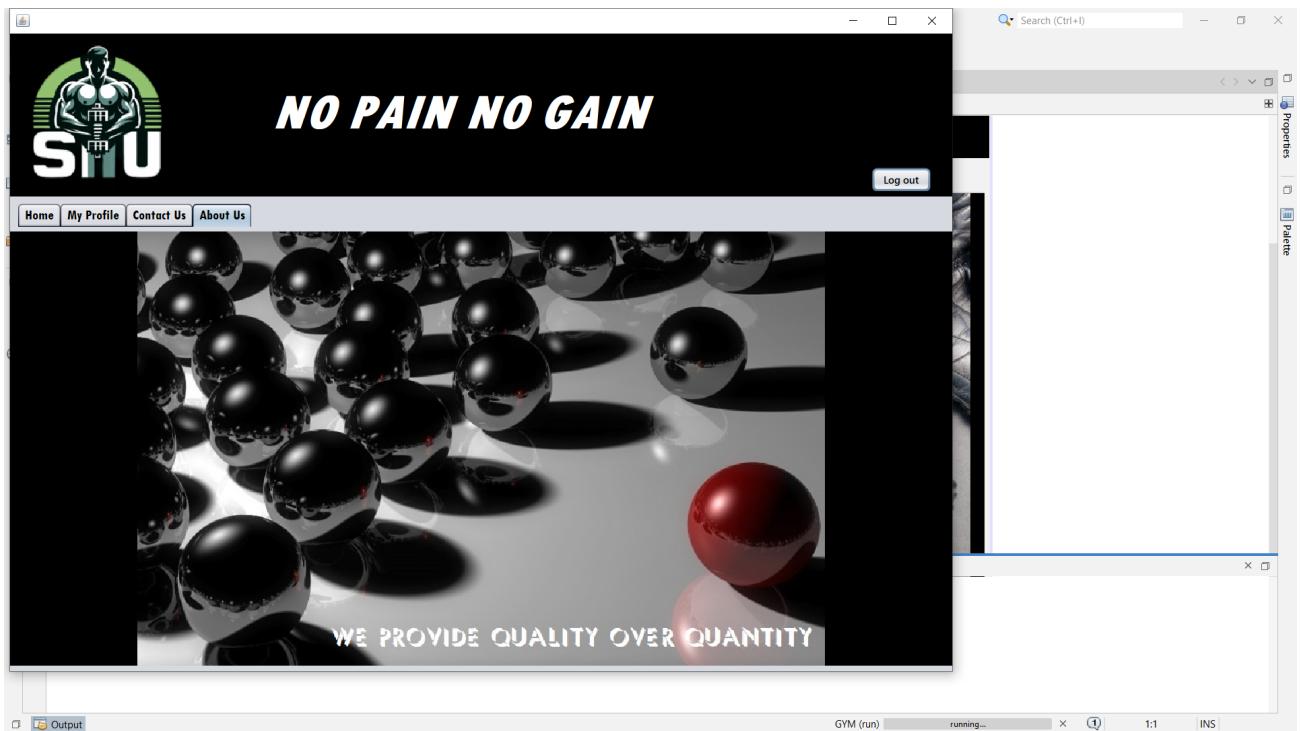




Thực hiện đăng ký lớp học







CHAPTER 3. KẾT LUẬN

MSSQL, một hệ thống quản lý cơ sở dữ liệu phổ biến, đã cho phép bạn lưu trữ và truy xuất dữ liệu khách hàng một cách hiệu quả. Bạn đã tạo các bảng cơ sở dữ liệu như "Khách hàng", "Gói đào tạo", "Lịch đào tạo" và mối quan hệ giữa chúng để lưu trữ thông tin chi tiết về khách hàng, các gói đào tạo và lịch trình đào tạo của họ. Điều này giúp bạn dễ dàng tìm kiếm, truy xuất và cập nhật dữ liệu khi cần thiết.

NetBeans, một môi trường phát triển tích hợp (IDE) mạnh mẽ, hỗ trợ bạn phát triển các ứng dụng quản lý khách hàng. Bằng cách sử dụng NetBeans, bạn có thể thiết kế giao diện thân thiện với người dùng và tạo các chức năng quản lý khách hàng như thêm, xem, sửa đổi và xóa thông tin khách hàng. Giao diện người dùng được thiết kế dễ sử dụng và trực quan, giúp người dùng dễ dàng tương tác với hệ thống.

Thông qua dự án này, bạn đã thể hiện khả năng sử dụng SQL và NetBeans để xây dựng một hệ thống quản lý khách hàng phòng tập hoàn chỉnh. Hệ thống của bạn không chỉ giúp quản

lý thông tin khách hàng hiệu quả mà còn mang lại trải nghiệm tốt cho người dùng. Bạn đã thể hiện khả năng thiết kế, phát triển và triển khai ứng dụng thành công.

Tuy nhiên, để nâng cao dự án của bạn, có thể bạn có thể xem xét một số cải tiến tiềm năng. Ví dụ: bạn có thể thêm tính năng báo cáo và thống kê để phân tích dữ liệu khách hàng và hiệu suất phòng tập. Bạn cũng có thể cân nhắc việc tích hợp tính năng bảo mật để bảo vệ thông tin khách hàng. Ngoài ra, tối ưu hóa hiệu suất và mở rộng khả năng mở rộng của hệ thống cũng là những điểm cần cân nhắc để đảm bảo có thể đáp ứng nhu cầu ngày càng tăng của phòng tập.

Tóm lại, dự án quản lý khách hàng tập gym bằng SQL và NetBeans của bạn đã tạo ra một hệ thống quản lý mạnh mẽ và linh hoạt. Bạn đã thể hiện khả năng của mình trong việc thiết kế và phát triển ứng dụng và dự án này cung cấp nền tảng tốt cho việc nâng cao và mở rộng trong tương lai.