

# Report

## University Information Management System: New Business Model and Architectural Design

### Abstract

This report aims to provide an improved data model for university information management systems. Compared to traditional university information systems, the new model offers a more unified and efficient approach to handling large amounts of data and relationships by centralizing storage and management.

---

### Table of Contents

1. Current State Analysis
  2. Proposed Solution and Advantages
  3. Architectural Diagram
  4. Class Diagram and New Capabilities
  5. Complete Running Application
  6. Overall Assessment and Future Implications
  7. Conclusion
- 

### 1. Current State Analysis

Traditional university data management systems are often fragmented, possibly managed by different departments or systems. For example, student information, professor information, and course information may be stored in separate databases or tables, leading to data redundancy and increased complexity in management.

### Challenges

- Fragmented data storage
  - Data redundancy
  - Lack of real-time data updates
-

## 2. Proposed Solution and Advantages

### Proposed Solutions

1. **Data Encryption and Security:** Implement stronger encryption measures for the `NewPassword` field, such as using modern, secure hashing algorithms like bcrypt or Argon2.
2. **Data Validation and Integrity:** Enforce data validation rules for all input fields (e.g., `Username`, `Language`, `Rate`, etc.) to ensure data accuracy and integrity.
3. **Activity Monitoring and Auto-updating:** Automatically update the `LastActive` and `LastUpdate` fields through either application logic or database triggers.
4. **Student Evaluation System:** Introduce a student evaluation system that allows students to rate the teaching effectiveness of professors, affecting their `Rate` field.
5. **Teacher Resource Sharing:** Provide a platform for professors to upload and share teaching resources like lecture slides, papers, etc.
6. **Business Collaboration:** Collaborate with corporations and research institutions to provide internship and research opportunities, categorized via the `Region` and `Language` fields.
7. **Rating Algorithm Optimization:** Use a more complex algorithm (such as a weighted average) for calculating the `Rate` to achieve a more accurate and fair rating.
8. **Multi-Language Support:** Allow the `Language` field to store multiple languages to reflect the professor's multilingual capabilities or teaching requirements.
9. **Regional Categorization and Search:** Standardize and categorize the `Region` field for easier future data analysis and search functionalities.

### Advantages

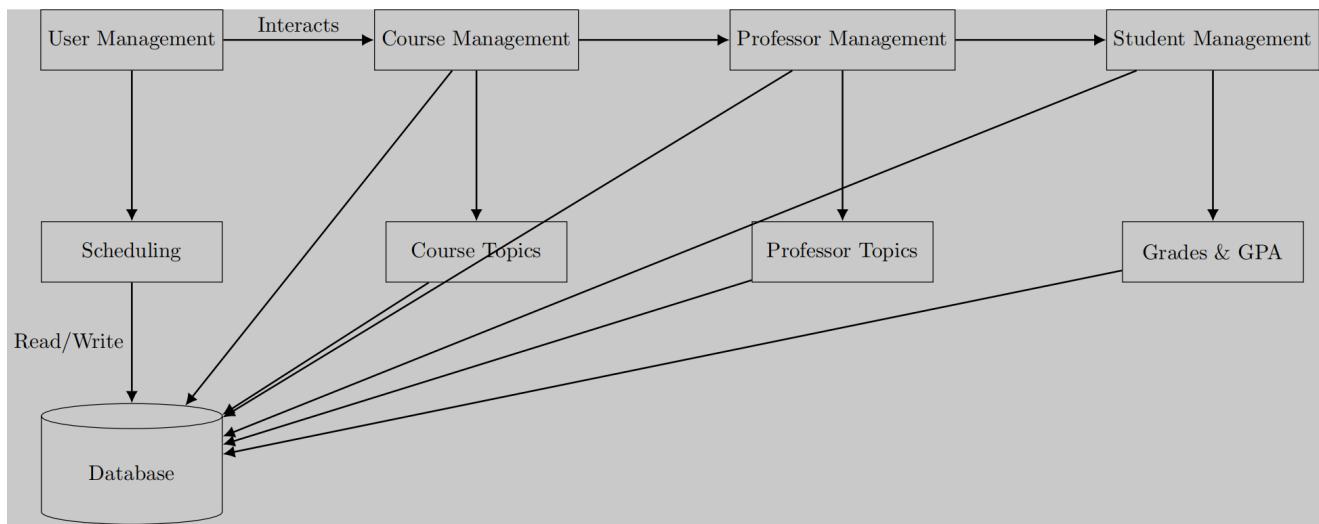
1. **Enhanced Security:** Strong hashing algorithms and data encryption will greatly improve data security.
2. **Data Accuracy:** Data validation and integrity checks will reduce errors and inconsistencies, enhancing the quality of the data.
3. **Real-time Updates:** Automatic updating of activity and update timestamps ensures that the data is always up-to-date.
4. **Accurate and Fair Ratings:** An optimized rating algorithm will more accurately reflect the professor's performance and evaluation.
5. **Diversity and Inclusivity:** Multi-language support and regional categorization not only provide more information but also cater to more diverse query requirements.
6. **Ease of Analysis and Search:** A standardized and categorized `Region` field will simplify the data analysis process and provide more accurate search results.
7. **Student Engagement:** The student evaluation system will increase student engagement and provide more feedback to both professors and educational institutions.

8. **Teacher Development:** The resource sharing platform will aid in the professional development of teachers and improve the quality of teaching.
9. **Business Opportunities:** Collaborations with corporations and feedback from employees will offer more business and research opportunities for schools and professors.

By employing these solutions and advantages, we can manage and utilize the `Professor`, `Student`, `Employee` table more effectively, thereby enhancing the overall quality and usability of the system.

---

### 3. Architectural Diagram and Design



In this model, all people (whether employees, professors, or students) are stored in a centralized `Person` table and differentiated by a `Role` field. This is unlike traditional models, which may scatter these roles across multiple different tables or databases.

---

### 4. Class Diagram and New Capabilities

#### Class Diagram



In the new model, almost all tables are connected to other tables through foreign keys to ensure data referential integrity. This ensures that any updates to a table at any time will correctly reflect in other tables associated with it.

## Database Architecture

<b>Course</b>	<b>Professor</b>	<b>Rate</b>	<b>Student</b>
<ul style="list-style-type: none"> <li>■ <b>id</b> varchar(255)</li> <li>■ <b>semesterid</b> varchar(255)</li> <li>■ <b>name</b> varchar(255)</li> <li>■ <b>introduction</b> varchar(255)</li> <li>■ <b>point</b> int</li> <li>■ <b>location</b> varchar(255)</li> <li>■ <b>statue</b> varchar(255)</li> <li>■ <b>studentlimited</b> int</li> <li>■ <b>studentcount</b> int</li> <li>■ <b>begintime</b> timestamp</li> <li>■ <b>endtime</b> timestamp</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>id</b> varchar(255)</li> <li>■ <b>username</b> varchar(255)</li> <li>■ <b>nowpassword</b> varchar(255)</li> <li>■ <b>language</b> varchar(255)</li> <li>■ <b>enabled</b> varchar(255)</li> <li>■ <b>rate</b> double</li> <li>■ <b>region</b> varchar(255)</li> <li>■ <b>lastactive</b> timestamp</li> <li>■ <b>lastupdate</b> timestamp</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>professor_id</b> varchar(255)</li> <li>■ <b>student_id</b> varchar(255)</li> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>scorepart1</b> varchar(255)</li> <li>■ <b>scorepart2</b> varchar(255)</li> <li>■ <b>scorepart3</b> varchar(255)</li> <li>■ <b>scorepart4</b> varchar(255)</li> <li>■ <b>comment</b> varchar(255)</li> <li>■ <b>score</b> varchar(255)</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>id</b> varchar(255)</li> <li>■ <b>username</b> varchar(255)</li> <li>■ <b>nowpassword</b> varchar(255)</li> <li>■ <b>enabled</b> varchar(255)</li> <li>■ <b>gpa</b> double</li> <li>■ <b>lastactive</b> timestamp</li> <li>■ <b>lastupdate</b> timestamp</li> </ul>
<b>Employee</b>	<b>PersonSchedule</b>	<b>Semester</b>	<b>CourseSchedule</b>
<ul style="list-style-type: none"> <li>■ <b>id</b> varchar(255)</li> <li>■ <b>username</b> varchar(255)</li> <li>■ <b>nowpassword</b> varchar(255)</li> <li>■ <b>enabled</b> varchar(255)</li> <li>■ <b>lastactive</b> timestamp</li> <li>■ <b>lastupdate</b> timestamp</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>semesterid</b> varchar(255)</li> <li>■ <b>starttime</b> timestamp</li> <li>■ <b>endtime</b> timestamp</li> <li>■ <b>personid</b> varchar(255)</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>id</b> varchar(255)</li> <li>■ <b>semstername</b> varchar(255)</li> <li>■ <b>semesterstart</b> timestamp</li> <li>■ <b>semesterend</b> timestamp</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>weekday</b> varchar(255)</li> <li>■ <b>starttime</b> timestamp</li> <li>■ <b>endtime</b> timestamp</li> </ul>
<b>Person</b>	<b>CourseStudent</b>	<b>ProfessorTopic</b>	<b>CourseProfessor</b>
<ul style="list-style-type: none"> <li>■ <b>PersonName</b> varchar(255)</li> <li>■ <b>PersonID</b> varchar(255)</li> <li>■ <b>role</b> varchar(255)</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>student_id</b> varchar(255)</li> <li>■ <b>score</b> double</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>professor_id</b> varchar(255)</li> <li>■ <b>topic</b> varchar(255)</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>professor_id</b> varchar(255)</li> </ul>
<b>CourseTopic</b>	<b>Historypwd</b>		
<ul style="list-style-type: none"> <li>■ <b>course_id</b> varchar(255)</li> <li>■ <b>topic</b> varchar(255)</li> </ul>	<ul style="list-style-type: none"> <li>■ <b>personid</b> varchar(255)</li> <li>■ <b>password</b> varchar(255)</li> </ul>		

The database uses a Relational Database Management System (RDBMS) for organizing and managing the data. MySQL is the primary database system used here.

## 1. Person Class

This table stores basic information related to all persons on the platform.

Field Name	Data Type	Description	Constraint
PersonID	varchar(255)	Unique Identifier	Primary Key
PersonName	varchar(255)	Name of the person	
Role	varchar(255)	Role (e.g., Student, Professor, Admin)	

## Overview

The `Person` class is a Java class designed to represent individuals. It contains basic attributes such as name, ID, and role, and provides methods for saving, updating, loading, and deleting person objects in a database.

## Attributes

- `personName` (static String) : The name of the person.
- `personID` (String) : The unique identifier for the person.
- `role` (String) : The role or position of the person.

## Constructor

```
public Person(String personName, String personID, String role)
```

The constructor is used to create a `Person` object and initialize the person's name, ID, and role.

## Methods

### Getting and Setting Attribute Values

- `getPersonName(): String` : Get the person's name.
- `setPersonName(String personName)` : Set the person's name.
- `getPersonID(): String` : Get the person's unique ID.
- `setPersonID(String personID)` : Set the person's unique ID.
- `getRole(): String` : Get the person's role.
- `setRole(String role)` : Set the person's role.

### Database Interaction

- `saveToDatabase(Connection connection)` : Save the `Person` object to the database. Inserts a new record with the person's name, ID, and role into the database.
- `updateInDatabase(Connection connection)` : Update the information of the `Person` object in the database. Modifies the name and role based on the person's ID.
- `loadFromDatabase(Connection connection, String personID)` : Load a `Person` object from the database using the person's ID. Returns a `Person` object with the retrieved information.
- `deleteFromDatabase(Connection connection, String personID)` : Delete a specific `Person` object from the database using the person's ID.

## Usage Examples

Here are some examples of using the `Person` class:

```
// Create a new person object
Person person = new Person("John Doe", "12345", "Employee");

// Save the person to the database
```

```

Connection dbConnection = // Obtain a database connection
person.saveToDatabase(dbConnection);

// Load a person from the database
Person loadedPerson = Person.loadFromDatabase(dbConnection, "12345");

```

## Notes

- Properly handle opening and closing database connections when working with a database.
  - This class is a simplified example and can be extended to include more attributes and methods as needed.
- 

## 2. Employee Class

This table stores information related to employees, such as username and password.

Field Name	Data Type	Description	Constraint
ID	varchar(255)	Unique Identifier for Employee	Foreign Key (Person)
Username	varchar(255)	Username	
NowPassword	varchar(255)	Current Hashed Password	
Enabled	varchar(255)	Is Enabled (e.g., 'Yes', 'No')	
LastActive	timestamp	Last Active Timestamp	
LastUpdate	timestamp	Last Update Timestamp	

## Overview

The `Employee` class is a Java class for representing employees, extending from the `Person` class. In addition to the basic attributes such as name, ID, and role from `Person`, it also includes username, hashed password, enabled status, last active timestamp, and last updated timestamp. The class provides methods for interacting with the database.

## Attributes

- `ID (String)` : Unique identifier for the employee, a foreign key pointing to `personID` in `Person`.
- `username (String)` : Username of the employee.
- `hashedPassword (String)` : Hashed password of the employee.
- `enabled (String)` : Indicates whether the employee is enabled.
- `lastActive (Timestamp)` : Timestamp of the last activity.
- `lastUpdate (Timestamp)` : Timestamp of the last update.

## Constructors

```
public Employee(String ID, String username, String hashedPassword, String enabled, Timestamp lastActive, Timestamp lastUpdate)
```

Constructor to create an `Employee` object and initialize all its attributes.

## Methods

### Getters and Setters

- `getID(): String` : Gets the unique ID of the employee.
- `setID(String ID)` : Sets the unique ID of the employee.
- `getUsername(): String` : Gets the username of the employee.
- `setUsername(String username)` : Sets the username of the employee.
- `getHashedPassword(): String` : Gets the hashed password of the employee.
- `setHashedPassword(String hashedPassword)` : Sets the hashed password of the employee.
- `getEnabled(): String` : Gets whether the employee is enabled.
- `setEnabled(String enabled)` : Sets whether the employee is enabled.
- `getLastActive(): Timestamp` : Gets the last active timestamp of the employee.
- `setLastActive(Timestamp lastActive)` : Sets the last active timestamp of the employee.
- `getLastUpdate(): Timestamp` : Gets the last updated timestamp of the employee.
- `setLastUpdate(Timestamp lastUpdate)` : Sets the last updated timestamp of the employee.

### Database Interaction

- `saveToDatabase(Connection connection)` : Saves the `Employee` object to the database. Inserts all attributes as a new record into the database.
- `updateInDatabase(Connection connection)` : Updates the `Employee` object information in the database. Modifies other attributes based on the employee's ID.
- `loadFromDatabase(Connection connection, String ID)` : Loads the `Employee` object using the employee's ID from the database. Returns an `Employee` object with the retrieved information.
- `deleteFromDatabase(Connection connection, String ID)` : Deletes a specific `Employee` object using the employee's ID from the database.

## Usage Example

```
// Create a new employee object
Employee employee = new Employee("12345", "johndoe", "hashedPassword123",
```

```

"enabled", new Timestamp(System.currentTimeMillis()), new
Timestamp(System.currentTimeMillis()));

// Save the employee to the database
Connection dbConnection = // Get database connection
employee.saveToDatabase(dbConnection);

// Load the employee from the database
Employee loadedEmployee = Employee.loadFromDatabase(dbConnection,
"12345");

```

## Note

- Properly handle opening and closing of the database connection when interacting with the database.
  - This class is a simplified example and can be extended to include more attributes and methods as needed.
- 

## 3. Historypwd Table

This table stores password history information related to all persons on the platform.

Field Name	Data Type	Description	Constraint
PersonID	varchar(255)	Unique Identifier for Person	Foreign Key (Person)
Password	varchar(255)	Old Hashed Password	

## Overview

The `Historypwd` table is a database table used for recording password history. It is linked with the `Person` table and stores the password history associated with specific individuals. This can be useful for implementing security policies like password complexity or password change history.

## Fields

- `PersonID` (`varchar(255)`) : The unique identifier for the individual related to the password history, a foreign key pointing to `personID` in the `Person` table.
- `Password` (`varchar(255)`) : The hashed password stored in the history.

## Primary Operations

### Data Insertion

- `INSERT INTO Historypwd (PersonID, Password) VALUES (?, ?)` : Inserts a new password history record into the `Historypwd` table.

## Data Retrieval

- `SELECT * FROM Historypwd WHERE PersonID = ?` : Queries the password history related to a specified `PersonID`.

## Data Deletion

- `DELETE FROM Historypwd WHERE PersonID = ? AND Password = ?` : Deletes a specific password history record based on the specified `PersonID` and `Password`.

## Usage Example

```
-- Insert a new password history record
INSERT INTO Historypwd (PersonID, Password) VALUES ('12345',
'hashedPassword123');

-- Query the password history for a specific individual
SELECT * FROM Historypwd WHERE PersonID = '12345';
```

## Note

- Properly handle opening and closing of the database connection when interacting with the database.
- This table is a simplified example and can be extended to include more fields and operations as needed.

## 4. Professor Table

This table stores information related to professors.

Field Name	Data Type	Description	Constraint
ID	varchar(255)	Unique Identifier for Professor	Foreign Key (Person)
Username	varchar(255)	Username	
NowPassword	varchar(255)	Current Hashed Password	
Language	varchar(255)	Language of Teaching	
Enabled	varchar(255)	Is Enabled	
Rate	double	Average Rating of Professor	
Region	varchar(255)	Geographical Location or Office	

Field Name	Data Type	Description	Constraint
LastActive	timestamp	Last Active Timestamp	
LastUpdate	timestamp	Last Update Timestamp	

## Overview

The `Professor` table is a database table used for storing information about professors. It is linked with the `Person` table through a foreign key and primarily records the username, password, language, enablement status, rating, region, and the last active and updated timestamps for the professor.

## Fields

- `ID` (`varchar(255)`) : The unique identifier for the professor, a foreign key pointing to `personID` in the `Person` table.
- `Username` (`varchar(255)`) : The username for the professor.
- `NowPassword` (`varchar(255)`) : The hashed password for the professor.
- `Language` (`varchar(255)`) : The language associated with the professor.
- `Enabled` (`varchar(255)`) : Indicates whether the professor is enabled or not.
- `Rate` (`double`) : The rating for the professor.
- `Region` (`varchar(255)`) : The region where the professor is based.
- `LastActive` (`timestamp`) : The last active timestamp.
- `LastUpdate` (`timestamp`) : The last updated timestamp.

## Primary Operations

### Data Insertion

- `INSERT INTO Professor (ID, Username, NowPassword, Language, Enabled, Rate, Region, LastActive, LastUpdate) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)` : Inserts a new professor record into the `Professor` table.

### Data Retrieval

- `SELECT * FROM Professor WHERE ID = ?` : Retrieves information about a professor based on the specified `ID`.

### Data Update

- `UPDATE Professor SET Field = ? WHERE ID = ?` : Updates a certain field for a professor record based on the specified `ID`.

### Data Deletion

- `DELETE FROM Professor WHERE ID = ?` : Deletes a specific professor record based on the specified `ID`.

## Usage Example

```
-- Insert a new professor record
INSERT INTO Professor (ID, Username, NowPassword, Language, Enabled, Rate,
Region, LastActive, LastUpdate) VALUES ('12345', 'ProfJohn',
'hashedPassword123', 'English', 'Enabled', 4.5, 'USA', CURRENT_TIMESTAMP,
CURRENT_TIMESTAMP);

-- Retrieve information for a specific professor
SELECT * FROM Professor WHERE ID = '12345';
```

## Note

- Properly handle opening and closing of the database connection when interacting with the database.
- This table is a simplified example and can be extended to include more fields and operations as needed.

## 5. Student Table

This table stores information related to students.

Field Name	Data Type	Description	Constraint
ID	varchar(255)	Unique Identifier for Student	Foreign Key (Person)
Username	varchar(255)	Username	
NowPassword	varchar(255)	Current Hashed Password	
Enabled	varchar(255)	Is Enabled	
GPA	double	Grade Point Average	
LastActive	timestamp	Last Active Timestamp	
LastUpdate	timestamp	Last Update Timestamp	

## Overview

The `Student` table is designed to store detailed information about students. It is related to the `Person` table through a foreign key and primarily holds records of the student's username, hashed password, enabled status, Grade Point Average (GPA), and timestamps for last activity and last update.

## Fields

- `ID (varchar(255))` : A unique identifier for the student, which is a foreign key pointing to the `Person` table.
- `Username (varchar(255))` : The username of the student.
- `NowPassword (varchar(255))` : The hashed password for the student.
- `Enabled (varchar(255))` : Indicates whether the student's account is enabled.
- `GPA (double)` : The Grade Point Average of the student.
- `LastActive (timestamp)` : The timestamp for the student's last activity.
- `LastUpdate (timestamp)` : The timestamp for the last update of the student's information.

## Primary Operations

### Data Insertion

- `INSERT INTO Student (ID, Username, NowPassword, Enabled, GPA, LastActive, LastUpdate) VALUES (?, ?, ?, ?, ?, ?, ?)` : Inserts a new record into the `Student` table.

### Data Query

- `SELECT * FROM Student WHERE ID = ?` : Retrieves information related to a student based on the unique identifier (ID).

### Data Update

- `UPDATE Student SET Field = ? WHERE ID = ?` : Updates a specific field for a student record based on the unique identifier (ID).

### Data Deletion

- `DELETE FROM Student WHERE ID = ?` : Deletes a specific student record based on the unique identifier (ID).

## Example Usage

```
-- Insert a new student record
INSERT INTO Student (ID, Username, NowPassword, Enabled, GPA, LastActive, LastUpdate) VALUES ('54321', 'StudentJane', 'hashedPassword321', 'Enabled', 3.5, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP);

-- Query all information of a specific student
SELECT * FROM Student WHERE ID = '54321';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
  - This table serves as a simplified example and can be extended to include more fields and features as needed.
- 

## 6. Semester Table

This table stores information related to semesters.

Field Name	Data Type	Description	Constraint
ID	varchar(255)	Unique Identifier for Semester	Primary Key
SemesterName	varchar(255)	Name of the Semester (e.g., 'Fall 2021')	
SemesterStart	timestamp	Start Date of Semester	
SemesterEnd	timestamp	End Date of Semester	

## Overview

The `Semester` table is designed to store information related to academic semesters. It primarily holds records of the unique identifier for the semester, its name, start date, and end date.

## Fields

- `ID` (`varchar(255)`) : A unique identifier for the semester.
- `SemesterName` (`varchar(255)`) : The name of the semester.
- `SemesterStart` (`timestamp`) : The start date for the semester.
- `SemesterEnd` (`timestamp`) : The end date for the semester.

## Primary Operations

### Data Insertion

- `INSERT INTO Semester (ID, SemesterName, SemesterStart, SemesterEnd) VALUES (?, ?, ?, ?)` : Inserts a new record into the `Semester` table.

### Data Query

- `SELECT * FROM Semester WHERE ID = ?` : Retrieves information related to a semester based on its unique identifier (ID).

## Data Update

- UPDATE Semester SET Field = ? WHERE ID = ?: Updates a specific field for a semester record based on its unique identifier (ID).

## Data Deletion

- DELETE FROM Semester WHERE ID = ?: Deletes a specific semester record based on its unique identifier (ID).

## Example Usage

```
-- Insert a new semester record
INSERT INTO Semester (ID, SemesterName, SemesterStart, SemesterEnd) VALUES
('Fall2023', 'Fall 2023', '2023-09-01 00:00:00', '2023-12-31 23:59:59');

-- Query all information of a specific semester
SELECT * FROM Semester WHERE ID = 'Fall2023';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- This table serves as a simplified example and can be extended to include more fields and features as needed.

---

## 7. Course Table

This table stores information related to courses.

Field Name	Data Type	Description	Constraint
ID	varchar(255)	Unique Identifier for Course	Primary Key
SemesterID	varchar(255)	Unique Identifier for Semester	Foreign Key (Semester)
Name	varchar(255)	Name of the Course	
Introduction	varchar(255)	Course Description	
Point	integer	Credit Points	
Location	varchar(255)	Location for the Course	
Status	varchar(255)	Status of the Course (e.g., 'Open', 'Closed')	
StudentLimited	integer	Maximum Student Capacity	

Field Name	Data Type	Description	Constraint
StudentCount	integer	Current Number of Students	
BeginTime	timestamp	Course Start Time	
EndTime	timestamp	Course End Time	

## Overview

The `Course` table is designed to store information related to courses. It includes fields for the unique identifier of the course, associated semester, name, introduction, credits, location, status, student limit, current student count, start time, and end time.

## Fields

- `ID` (`varchar(255)`) : The unique identifier for the course.
- `SemesterID` (`varchar(255)`) : The unique identifier for the semester to which the course belongs, foreign key pointing to the `Semester` table.
- `Name` (`varchar(255)`) : The name of the course.
- `Introduction` (`varchar(255)`) : A brief introduction or description of the course.
- `Point` (`integer`) : The number of credits the course is worth.
- `Location` (`varchar(255)`) : The location or venue of the course.
- `Status` (`varchar(255)`) : The current status of the course.
- `StudentLimited` (`integer`) : The maximum number of students allowed in the course.
- `StudentCount` (`integer`) : The current number of enrolled students in the course.
- `BeginTime` (`timestamp`) : The start time of the course.
- `EndTime` (`timestamp`) : The end time of the course.

## Primary Operations

### Data Insertion

- `INSERT INTO Course (....) VALUES (....)` : Inserts a new record into the `Course` table.

### Data Query

- `SELECT * FROM Course WHERE ID = ?` : Retrieves information related to a course based on its unique identifier (ID).

### Data Update

- `UPDATE Course SET Field = ? WHERE ID = ?` : Updates a specific field for a course record based on its unique identifier (ID).

### Data Deletion

- `DELETE FROM Course WHERE ID = ?` : Deletes a specific course record based on its unique identifier (ID).

## Example Usage

```
-- Insert a new course record
INSERT INTO Course (ID, SemesterID, Name, Introduction, Point, Location,
Status, StudentLimited, StudentCount, BeginTime, EndTime) VALUES ('CS101',
'Fall2023', 'Computer Science 101', 'Introduction to Computer Science', 3,
'Room 201', 'Open', 50, 0, '2023-09-01 09:00:00', '2023-12-20 09:00:00');

-- Query all information of a specific course
SELECT * FROM Course WHERE ID = 'CS101';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- This table serves as a simplified example and can be extended to include more fields and features as needed.

## 8. CourseSchedule Table

This table stores information related to the course timetable.

Field Name	Data Type	Description	Constraint
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)
Weekday	varchar(255)	Day of the Week (e.g., 'Monday', 'Tuesday')	
StartTime	timestamp	Class Start Time	
EndTime	timestamp	Class End Time	

## Overview

The `CourseSchedule` table is designed to store information related to the timing and scheduling of courses. This table records the unique identifier for the course (foreign key pointing to the `Course` table), the weekday on which the course is held (e.g., Monday, Tuesday, etc.), the start time, and the end time for each session.

## Fields

- Course\_ID (varchar(255)) : The unique identifier for the course, foreign key pointing to the Course table.
- Weekday (varchar(255)) : The weekday on which the course is held (e.g., Monday, Tuesday, etc.).
- StartTime (timestamp) : The start time for the course session.
- EndTime (timestamp) : The end time for the course session.

## Primary Operations

### Data Insertion

- `INSERT INTO CourseSchedule (...) VALUES (...)` : Inserts a new scheduling record into the CourseSchedule table.

### Data Query

- `SELECT * FROM CourseSchedule WHERE Course_ID = ?` : Retrieves the scheduling information related to a specific course based on its unique identifier (Course\_ID).

### Data Update

- `UPDATE CourseSchedule SET Field = ? WHERE Course_ID = ? AND Weekday = ?` : Updates specific fields for a course schedule record based on its unique identifier (Course\_ID) and the weekday.

### Data Deletion

- `DELETE FROM CourseSchedule WHERE Course_ID = ? AND Weekday = ?` : Deletes a specific scheduling record based on its unique identifier (Course\_ID) and the weekday.

## Example Usage

```
-- Insert a new scheduling record
INSERT INTO CourseSchedule (Course_ID, Weekday, StartTime, EndTime) VALUES
('CS101', 'Monday', '2023-09-01 09:00:00', '2023-09-01 10:00:00');

-- Query all scheduling information of a specific course
SELECT * FROM CourseSchedule WHERE Course_ID = 'CS101';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- This table serves as a simplified example and can be extended to include more fields and features as needed.

## 9. PersonSchedule Table

This table stores information related to personal class timetables.

Field Name	Data Type	Description	Constraint
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)
SemesterID	varchar(255)	Unique Identifier for Semester	Foreign Key (Semester)
StartTime	timestamp	Class Start Time	
EndTime	timestamp	Class End Time	
PersonID	varchar(255)	Unique Identifier for Person	Foreign Key (Person)

## Overview

The `PersonSchedule` table is designed to store information related to individual course schedules. This table records the unique identifier for the course (foreign key pointing to the `Course` table), the unique identifier related to the semester (foreign key pointing to the `Semester` table), the start and end times for each session, and the unique identifier related to the person (foreign key pointing to the `Person` table).

## Fields

- `Course_ID` (`varchar(255)`) : The unique identifier for the course, foreign key pointing to the `Course` table.
- `SemesterID` (`varchar(255)`) : The unique identifier related to the semester, foreign key pointing to the `Semester` table.
- `StartTime` (`timestamp`) : The start time for the course session.
- `EndTime` (`timestamp`) : The end time for the course session.
- `PersonID` (`varchar(255)`) : The unique identifier related to the person, foreign key pointing to the `Person` table.

## Primary Operations

### Data Insertion

- `INSERT INTO PersonSchedule (....) VALUES (....)` : Inserts a new schedule record into the `PersonSchedule` table.

### Data Query

- `SELECT * FROM PersonSchedule WHERE PersonID = ?` : Retrieves the schedule information related to a specific person based on their unique identifier (PersonID).

### Data Update

- `UPDATE PersonSchedule SET Field = ? WHERE PersonID = ? AND Course_ID = ?;`  
Updates specific fields for a schedule record based on its unique identifier (PersonID) and the course ID.

## Data Deletion

- `DELETE FROM PersonSchedule WHERE PersonID = ? AND Course_ID = ?;` Deletes a specific schedule record based on its unique identifier (PersonID) and the course ID.

## Example Usage

```
-- Insert a new schedule record
INSERT INTO PersonSchedule (Course_ID, SemesterID, StartTime, EndTime,
PersonID) VALUES ('CS101', '2023Fall', '2023-09-01 09:00:00', '2023-09-01
10:00:00', '12345');

-- Query all schedule information of a specific person
SELECT * FROM PersonSchedule WHERE PersonID = '12345';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- This table serves as a simplified example and can be extended to include more fields and features as needed.

## 10. CourseTopic Table

This table stores information related to topics associated with courses.

Field Name	Data Type	Description	Constraint
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)
Topic	varchar(255)	Topic associated with the Course (e.g., 'Mathematics', 'Computer Science')	

## Overview

The `CourseTopic` table is designed to store information related to topics that are associated with courses. This table includes the unique identifier for the course (foreign key pointing to the `Course` table) and the topic related to the course.

## Fields

- Course\_ID (varchar(255)) : The unique identifier for the course, foreign key pointing to the Course table.
- Topic (varchar(255)) : The topic related to the course.

## Primary Operations

### Data Insertion

- `INSERT INTO CourseTopic (Course_ID, Topic) VALUES (?, ?)` : Inserts a new record into the CourseTopic table.

### Data Query

- `SELECT * FROM CourseTopic WHERE Course_ID = ?` : Retrieves the topic(s) related to a specific course based on its unique identifier.

### Data Deletion

- `DELETE FROM CourseTopic WHERE Course_ID = ? AND Topic = ?` : Deletes a specific record based on its course unique identifier and topic.

## Example Usage

```
-- Insert a new course-topic record
INSERT INTO CourseTopic (Course_ID, Topic) VALUES ('CS101', 'Data
Structures');

-- Query all topics related to a specific course
SELECT * FROM CourseTopic WHERE Course_ID = 'CS101';
```

## 11. ProfessorTopic Table

This table stores information related to topics that professors specialize in or are associated with.

Field Name	Data Type	Description	Constraint
Professor_ID	varchar(255)	Unique Identifier for Professor	Foreign Key (Professor)
Topic	varchar(255)	Topic associated with the Professor (e.g., 'Mathematics', 'Computer Science')	

# Overview

The `ProfessorTopic` table is used for storing topics related to specific professors. Each record is associated with the `Professor` table via a foreign key.

## Fields

- `Professor_ID` (`varchar(255)`) : The unique identifier for the professor, a foreign key pointing to the `Professor` table.
- `Topic` (`varchar(255)`) : The topic related to the professor.

## Primary Operations

### Data Insertion

- `INSERT INTO ProfessorTopic (Professor_ID, Topic) VALUES (?, ?)` : Inserts a new topic related to a specific professor into the table.

### Data Retrieval

- `SELECT * FROM ProfessorTopic WHERE Professor_ID = ?` : Retrieves all topics related to a professor based on their unique identifier.

### Data Deletion

- `DELETE FROM ProfessorTopic WHERE Professor_ID = ? AND Topic = ?` : Deletes a record related to a specific professor and topic.

## Usage Example

```
-- Insert a new record
INSERT INTO ProfessorTopic (Professor_ID, Topic) VALUES ('12345',
'Computer Science');

-- Retrieve all topics for a specific professor
SELECT * FROM ProfessorTopic WHERE Professor_ID = '12345';
```

## Note

- This table primarily serves to record the areas of expertise or teaching topics for professors.
- Properly handle opening and closing of the database connection when interacting with the database.

## 12. CourseProfessor Table

This table stores information related to the association between courses and professors.

Field Name	Data Type	Description	Constraint
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)
Professor_ID	varchar(255)	Unique Identifier for Professor	Foreign Key (Professor)

### Overview

The `CourseProfessor` table is designed to store information related to the relationship between courses and professors. This table includes the unique identifier for the course (foreign key pointing to the `Course` table) and the unique identifier for the professor (foreign key pointing to the `Professor` table).

### Fields

- `Course_ID` (`varchar(255)`) : The unique identifier for the course, foreign key pointing to the `Course` table.
- `Professor_ID` (`varchar(255)`) : The unique identifier for the professor, foreign key pointing to the `Professor` table.

### Primary Operations

#### Data Insertion

- `INSERT INTO CourseProfessor (Course_ID, Professor_ID) VALUES (?, ?)` : Inserts a new record into the `CourseProfessor` table.

#### Data Query

- `SELECT * FROM CourseProfessor WHERE Course_ID = ? OR Professor_ID = ?` : Retrieves records related to a specific course or professor based on their unique identifiers.

#### Data Deletion

- `DELETE FROM CourseProfessor WHERE Course_ID = ? AND Professor_ID = ?` : Deletes a specific record based on its course and professor unique identifiers.

### Example Usage

```
-- Insert a new course-professor relationship record
INSERT INTO CourseProfessor (Course_ID, Professor_ID) VALUES ('CS101',
'P123');
```

```
-- Query all professors related to a specific course  
SELECT * FROM CourseProfessor WHERE Course_ID = 'CS101';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
  - These tables serve as simplified examples and can be extended to include more fields and features as needed.
  -
- 

## 13.CourseStudent Table

This table stores information related to the association between courses and students.

Field Name	Data Type	Description	Constraint
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)
Student_ID	varchar(255)	Unique Identifier for Student	Foreign Key (Student)
Score	double	Score of the Student in the Course	

## Overview

The `CourseStudent` table is designed to store information related to the relationship between courses and students. This table includes the unique identifier for the course (foreign key pointing to the `Course` table), the unique identifier for the student (foreign key pointing to the `Student` table), and the score the student has achieved in the course.

## Fields

- `Course_ID` (`varchar(255)`) : The unique identifier for the course, foreign key pointing to the `Course` table.
- `Student_ID` (`varchar(255)`) : The unique identifier for the student, foreign key pointing to the `Student` table.
- `Score` (`double`) : The score achieved by the student in the course.

## Primary Operations

### Data Insertion

- `INSERT INTO CourseStudent (Course_ID, Student_ID, Score) VALUES (?, ?, ?)` : Inserts a new record into the `CourseStudent` table.

## Data Query

- `SELECT * FROM CourseStudent WHERE Course_ID = ? OR Student_ID = ?:`  
Retrieves records related to a specific course or student based on their unique identifiers.

## Data Update

- `UPDATE CourseStudent SET Score = ? WHERE Course_ID = ? AND Student_ID = ?:` Updates the score for a specific student in a specific course.

## Data Deletion

- `DELETE FROM CourseStudent WHERE Course_ID = ? AND Student_ID = ?:` Deletes a specific record based on its course and student unique identifiers.

## Example Usage

```
-- Insert a new course-student relationship record with the score
INSERT INTO CourseStudent (Course_ID, Student_ID, Score) VALUES ('CS101',
'S123', 95);

-- Query all students and their scores related to a specific course
SELECT * FROM CourseStudent WHERE Course_ID = 'CS101';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- These tables serve as simplified examples and can be extended to include more fields and features as needed.

---

## 14. Rate Table

This table stores information related to ratings and reviews for professors.

Field Name	Data Type	Description	Constraint
Professor_ID	varchar(255)	Unique Identifier for Professor	Foreign Key (Professor)
Student_ID	varchar(255)	Unique Identifier for Student	Foreign Key (Student)
Course_ID	varchar(255)	Unique Identifier for Course	Foreign Key (Course)

Field Name	Data Type	Description	Constraint
ScorePart1	varchar(255)	Rating Part 1 (e.g., 'Teaching Quality')	
ScorePart2	varchar(255)	Rating Part 2 (e.g., 'Responsiveness')	
ScorePart3	varchar(255)	Rating Part 3 (e.g., 'Clarity')	
ScorePart4	varchar(255)	Rating Part 4 (e.g., 'Engagement')	
Comment	varchar(255)	Review or Comment	
Score	varchar(255)	Total Rating	

## Overview

The `Rate` table is designed to store ratings and reviews that students give to professors. This table includes the unique identifiers for the professor (foreign key pointing to `Professor` table), the student (foreign key pointing to `Student` table), and the course (foreign key pointing to `Course` table), along with various scores and comments.

## Fields

- `Professor_ID` (varchar(255)) : The unique identifier for the professor, foreign key pointing to `Professor` table.
- `Student_ID` (varchar(255)) : The unique identifier for the student, foreign key pointing to `Student` table.
- `Course_ID` (varchar(255)) : The unique identifier for the course, foreign key pointing to `Course` table.
- `ScorePart1` (varchar(255)) : Score part 1.
- `ScorePart2` (varchar(255)) : Score part 2.
- `ScorePart3` (varchar(255)) : Score part 3.
- `ScorePart4` (varchar(255)) : Score part 4.
- `Comment` (varchar(255)) : Comments on the rating.
- `Score` (varchar(255)) : Overall score.

## Primary Operations

### Data Insertion

- `INSERT INTO Rate (Professor_ID, Student_ID, Course_ID, ScorePart1, ScorePart2, ScorePart3, ScorePart4, Comment, Score) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)` : Inserts a new rating record into the `Rate` table.

### Data Query

- `SELECT * FROM Rate WHERE Professor_ID = ? OR Student_ID = ? OR Course_ID = ?`: Retrieves rating records related to a specific professor, student, or course based on their unique identifiers.

## Data Update

- `UPDATE Rate SET Comment = ?, Score = ? WHERE Professor_ID = ? AND Student_ID = ? AND Course_ID = ?`: Updates the comments and overall score of a specific rating.

## Data Deletion

- `DELETE FROM Rate WHERE Professor_ID = ? AND Student_ID = ? AND Course_ID = ?`: Deletes a specific rating record.

## Example Usage

```
-- Insert a new rating record
INSERT INTO Rate (Professor_ID, Student_ID, Course_ID, ScorePart1,
ScorePart2, ScorePart3, ScorePart4, Comment, Score) VALUES ('P123',
'S123', 'CS101', '5', '4', '5', '5', 'Excellent', '19');

-- Query all ratings related to a specific professor
SELECT * FROM Rate WHERE Professor_ID = 'P123';
```

## Note

- Properly manage the opening and closing of database connections when performing database operations.
- These tables serve as simplified examples and can be extended to include more fields and features as needed.

## Foreign Key Relationships

To maintain data referential integrity, multiple foreign key relationships exist between the tables. These relationships have been outlined above in the table designs.

## 5.Complete Running Application

### Login Area:

## 1. Login Page:

- **UserName:** Allows users to input their username.
- **Password:** Allows users to input their password.
- **User Type Selection:** Users can choose their role, e.g., Admin, Faculty, or Student.
- **Login Button:** Click to log in.
- **Register Button:** Click to register if the user does not have an account.

The screenshot shows a window titled "Login in" with a light gray background. At the top left, there is a small message "Please log in". Below it, there are two text input fields: "User Name:" and "Password:", each preceded by a label and followed by a blue-bordered input box. Underneath these fields is a row of three radio buttons labeled "Admin", "Faculty", and "Student". At the bottom of the window are two buttons: "Login" on the left and "Register" on the right. The window has a standard title bar with three colored dots (red, yellow, green) at the top left corner.

## 2. Validation Page:

- **Person Id:** Input for the user's ID.
- **Person Name:** Input for the user's name.
- **User Type Selection:** Similar to the login page, users can choose their role.
- **Register Button:** Click to complete registration.

- **Back Button:** Click to return to the previous page.

The screenshot shows a window titled "Validation". On the left side, there is a vertical panel with the text "Please log in". The main panel contains the following fields:

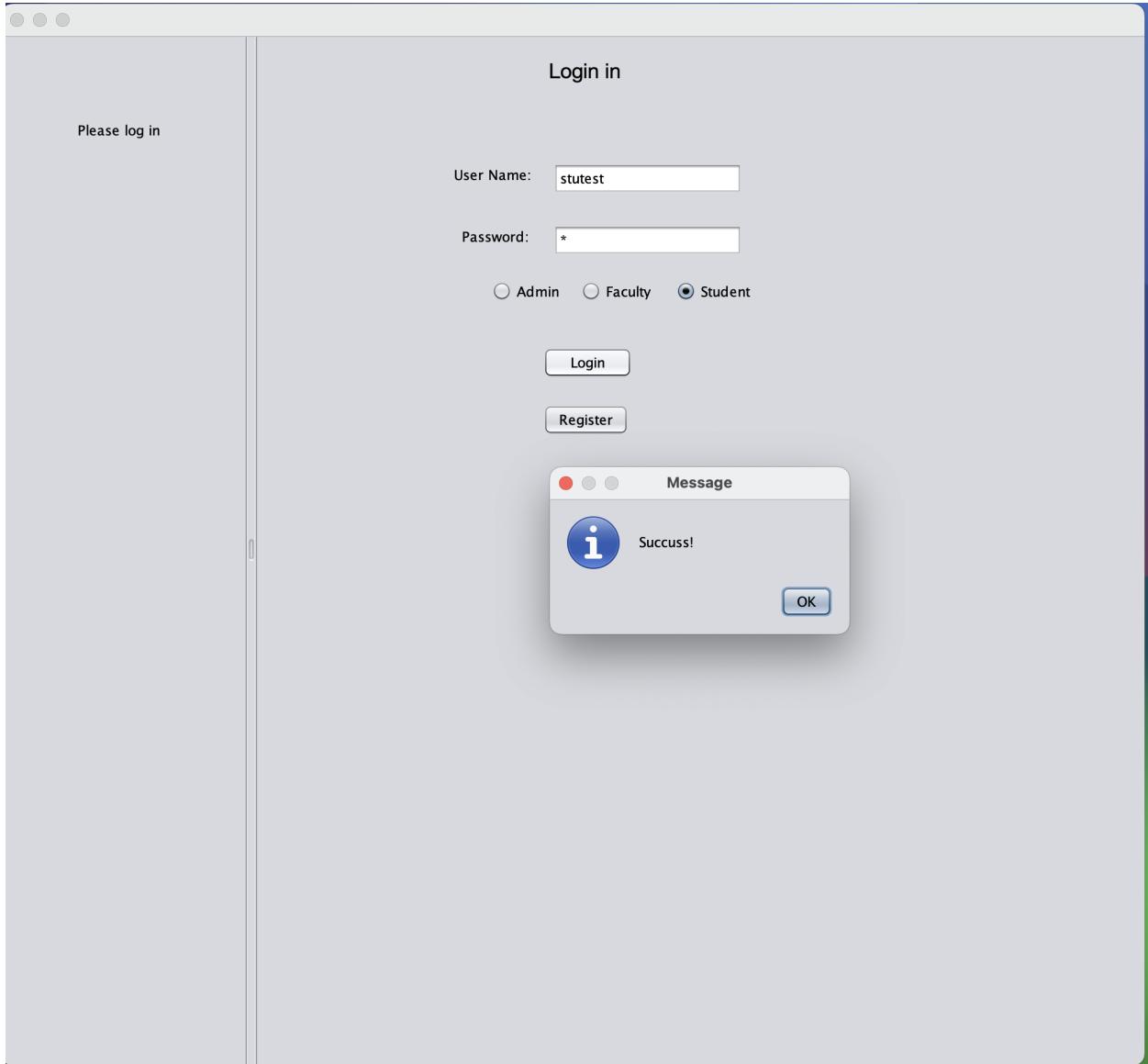
- "Person Id:" input field containing "10"
- "Person Name:" input field containing "teststu"
- Three radio buttons labeled "Admin", "Faculty", and "Student". The "Student" option is selected.

At the bottom of the main panel are two buttons: "Register" and "Back".

### 3.Create Account :

- **Username:** Input for the username.
- **Password:** Input for the user's password.

- **Create:** Click to create user's account

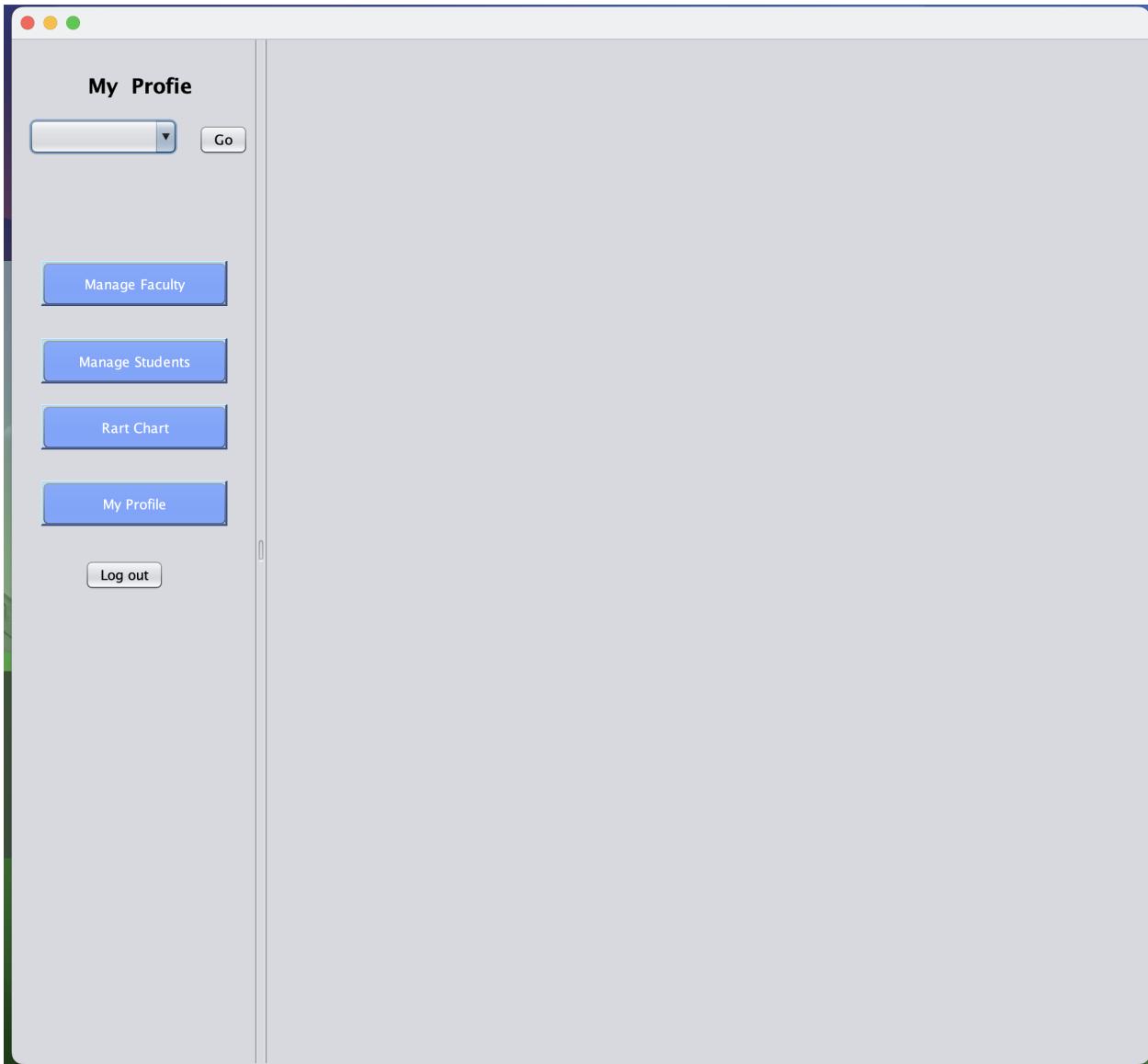


## Business WorkArea:

### 1. Left Navigation Panel:

- **Universal ComboxArea:** Click to go Where you want to visit and operate.
- **Manage Courses:** Enter the course management page to view, create, and edit courses.
- **Manage Student:** Enter the student management page to view, create, and edit courses.
- **Rate Chart:** View and edit your personal information.

- **Log out:** Click to log out of the system.



## 2. Universal ComboBoxArea:

### **CourseComboBoxArea JPanel:**

Click to view details and manage your courses.

#### 1. Course Overview:

##### **Table:**

- **Course ID:** Displays the identification numbers of the various courses.
- **Course Name:** Lists the names of the courses available.
- **Semester:** Shows the academic semester during which the course is offered.
- **Professor:** Indicates the instructor or professor responsible for the course.
- **Course Status:** Provides the current status of the course, likely indicating available slots or the course's full capacity.

#### 2. Course Search and Actions:

##### **Search Bar:**

- Allows users to input specific keywords or course IDs to find a particular course quickly.

#### **Buttons:**

- **Search:** Enables the user to execute a search based on the inputted keywords or course IDs.
- **Add:** Provides an option for the user to add a new course to the system.
- **View:** Allows the user to view detailed information about a specific course.
- **Delete:** Offers an option to remove a selected course from the system.
- **Update:** Enables the user to make changes to an existing course's details.

#### 3. Course Detail Input:

##### **Input Fields:**

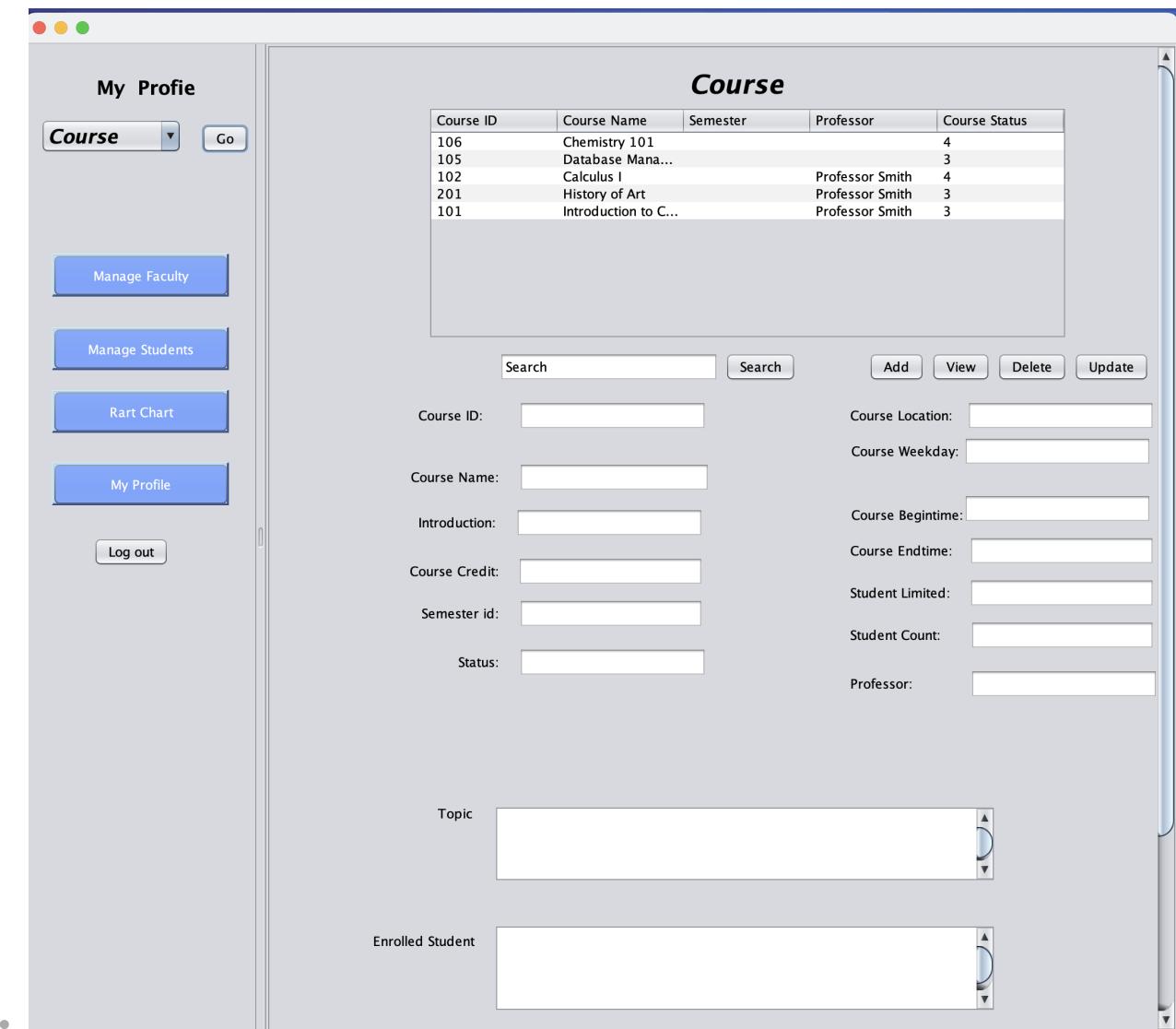
- **Course ID:** Where users can input or modify the identification number of a course.
- **Course Name:** A field to enter or change the name of a course.
- **Introduction:** Allows users to provide a brief description or introduction about the course.
- **Course Credit:** Lets users specify the credit value of the course.
- **Semester ID:** A field to indicate the specific semester code or identification.
- **Status:** Denotes the current status of the course.
- **Course Location:** Specifies the venue or location where the course will be conducted.
- **Course Weekday:** Indicates the day(s) on which the course takes place.
- **Course Begintime:** Sets the starting time of the course.
- **Course Endtime:** Sets the ending time of the course.
- **Student Limited:** Specifies the maximum number of students that can enroll in the course.
- **Student Count:** Indicates the current number of students enrolled in the course.
- **Professor:** Allows the user to assign or modify the professor in charge of the course.

##### **Text Area:**

- **Topic:** A space where specific topics or subjects covered in the course can be detailed.

##### **List Display:**

- **Enrolled Student:** Shows a list of students currently enrolled in the particular course.



## CourseScheduleComboBoxAreaJPanel:

### 1. Schedule Overview:

#### Table:

- Course ID:** Lists the identification numbers of the courses that are scheduled.
- Weekday:** Specifies the day of the week when the course is held.
- Start Time:** Indicates the beginning time for each course.
- End Time:** Indicates the ending time for each course.

### 2. Course Schedule Input:

#### Input Fields:

- Course ID:** A field where users can enter or edit the identification number of the course.
- Weekday:** A dropdown or input field to specify or edit the day of the week for the course.
- Start Time:** A field to set or modify the starting time of the course.
- End Time:** A field to set or modify the ending time of the course.

### 3. Actions:

#### Buttons:

- **View:** Allows the user to view detailed information or specifics about the course schedule.
- **Create:** Enables the user to add a new course schedule using the input fields provided.
- **Update:** Allows the user to modify or update an existing course schedule.
- **Delete:** Provides an option to remove a selected course schedule from the list.

The screenshot shows a Java Swing application window titled "Course Schedule". On the left, there is a sidebar with a "My Profile" section containing a dropdown menu ("Course S...") and a "Go" button, followed by several blue rectangular buttons labeled "Manage Faculty", "Manage Students", "Rart Chart", "My Profile", and "Log out". The main area is titled "Course Schedule" and contains a table with columns: Course ID, Weekday, Start Time, and End Time. The table data is as follows:

Course ID	Weekday	Start Time	End Time
101	Monday	2023-08-21 09:00:00	2023-08-21 10:00:00
101	Wednesday	2023-08-23 09:00:00	2023-08-23 10:00:00
102	Tuesday	2023-08-22 11:00:00	2023-08-22 12:00:00
201	Thursday	2024-01-11 14:00:00	2024-01-11 16:00:00
105	Monday	2024-08-21 13:00:00	2024-08-21 15:00:00
105	Wednesday	2024-08-23 13:00:00	2024-08-23 15:00:00
106	Tuesday	2025-01-11 08:00:00	2025-01-11 10:00:00

Below the table are four buttons: "View", "Create", "Update", and "Delete". At the bottom, there are four input fields with placeholder values: "Course ID" (101), "Start Time" (13-08-23 09:00:00), "Weekday" (Wednesday), and "End Time" (13-08-23 10:30:00).

### EmployeeComboBoxArea JPanel:

Navigate to the employee management page to oversee, add, or modify employee data.

#### 1. Employee Overview:

#### Table:

- **ID:** Displays the identification numbers of the various employees.
- **Name:** Shows the names of the employees.
- **isEnabled:** Indicates whether the employee account is active or not.
- **Password:** Provides a masked view of the employee's password.

## 2. Employee Search and Actions:

### Search Bar:

- **Search ID:** Allows users to input a specific employee ID to quickly locate an employee's details.

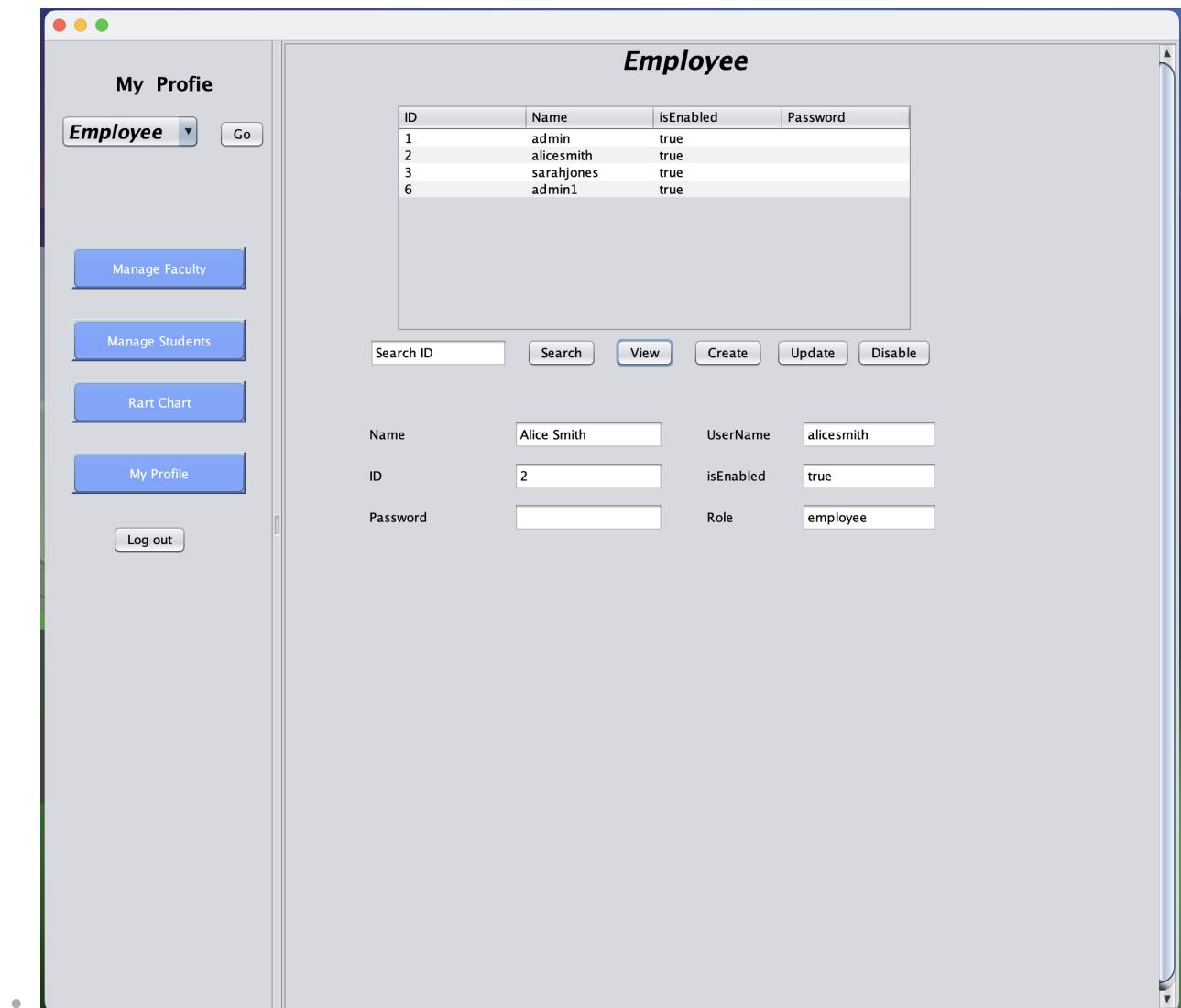
### Buttons:

- **Search:** Enables the user to execute a search based on the inputted employee ID.
- **View:** Allows the user to view detailed information about a specific employee.
- **Create:** Provides an option for the user to add a new employee to the system.
- **Update:** Enables the user to make changes to an existing employee's details.
- **Disable:** Offers an option to deactivate an employee's account.

## 3. Employee Detail Input:

### Input Fields:

- **Name:** A field where users can enter or modify an employee's name.
- **ID:** Where users can input or change the identification number of an employee.
- **UserName:** Allows users to provide or change the username used by the employee for login purposes.
- **Password:** Lets users input or change the password for the employee's account.
- **isEnabled:** A checkbox or toggle to indicate if the employee account is active.
- **Role:** Specifies the role or designation of the employee within the organization.



## FacultyComboBoxAreaJPanel:

Engage with the faculty management area to network, share, or gather faculty-specific resources.

### 1. Professor Overview:

#### Table:

- **ID:** Displays the identification numbers of the various professors.
- **Name:** Shows the names of the professors.
- **isEnabled:** Indicates whether the professor's account is active or not.
- **Rating:** Provides a numeric rating for the professors, indicating their performance or feedback score.
- **Region:** Specifies the region where the professor is based or assigned.

### 2. Professor Search and Actions:

#### Search Bar:

- **Search ID:** Allows users to input a specific professor ID to quickly locate a professor's details.

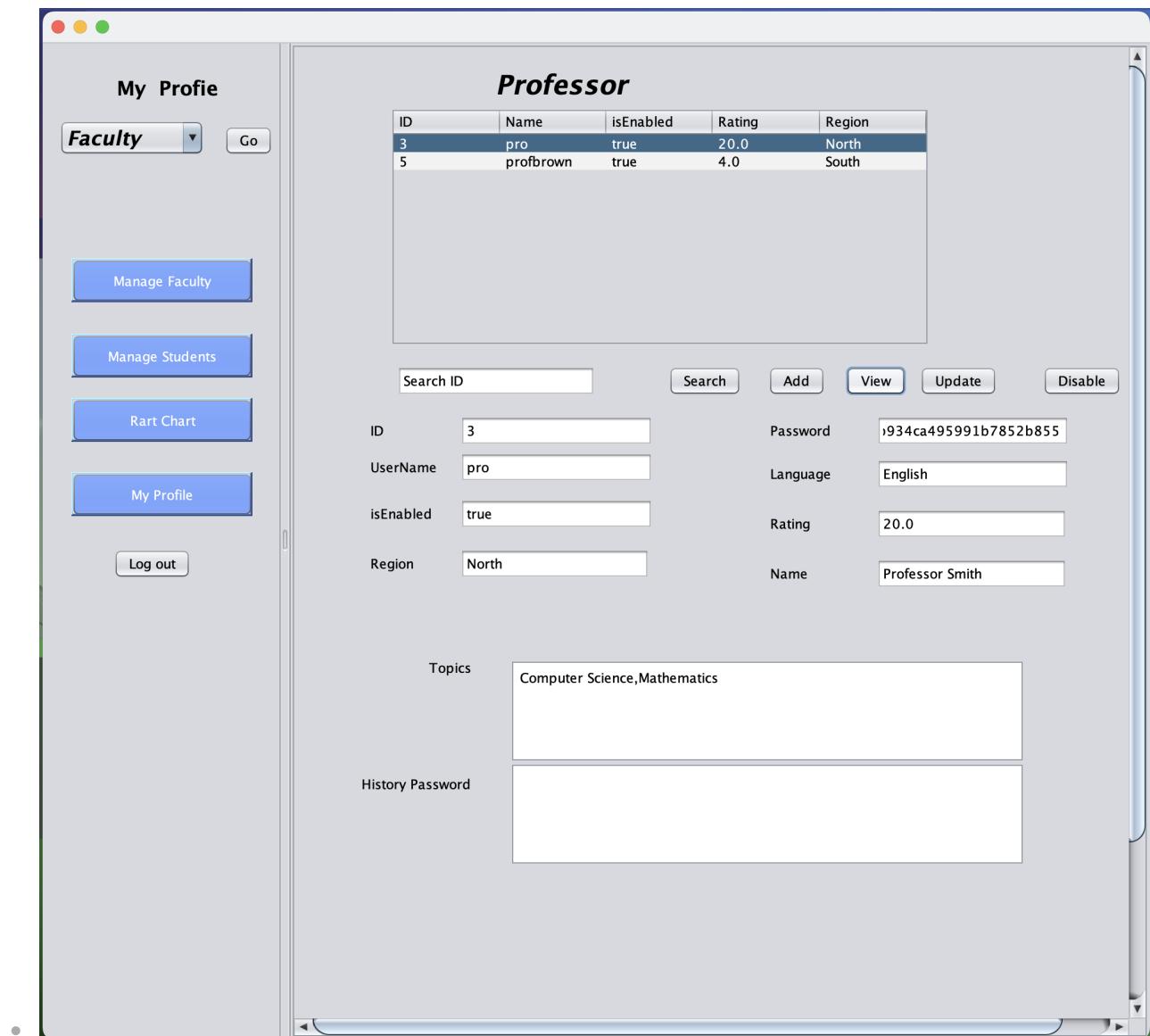
#### Buttons:

- **Search:** Enables the user to execute a search based on the inputted professor ID.
- **Add:** Provides an option for the user to add a new professor to the system.
- **View:** Allows the user to view detailed information about a specific professor.
- **Update:** Enables the user to make changes to an existing professor's details.
- **Disable:** Offers an option to deactivate a professor's account.

#### 3. Professor Detail Input:

#### Input Fields:

- **ID:** Where users can input or change the identification number of a professor.
- **Name:** A field where users can enter or modify a professor's name.
- **isEnabled:** A checkbox or toggle to indicate if the professor account is active.
- **Password:** Lets users input or change the password for the professor's account.
- **Language:** Allows users to specify the languages spoken or taught by the professor.
- **Rating:** A field where users can input or modify the rating score of a professor.
- **Region:** Specifies the region or location where the professor operates.
- **UserName:** Allows users to provide or change the username used by the professor for login purposes.
- **Topics:** A larger text field where users can enter or modify the subjects or areas of expertise of the professor.
- **History Password:** A text area for users to view or input the password history associated with a professor's account.



## PersonComboBoxAreaJPanel:

Access and edit individual information or connect with others on the platform.

### 1. Person Overview:

#### Table:

- **ID:** Displays the identification numbers of various individuals.
- **Name:** Shows the names of the individuals.
- **Role:** Specifies the roles of the individuals, such as Employee, Professor, or Student.

### 2. Person Search and Actions:

#### Search Bar:

- **Search ID:** Allows users to input a specific person ID to quickly locate an individual's details.

#### Buttons:

- **Search:** Enables the user to execute a search based on the inputted person ID.
- **View:** Allows the user to view detailed information about a specific individual.
- **Create:** Provides an option for the user to add a new individual to the system.
- **Update:** Enables the user to make changes to an existing individual's details.
- **Delete:** Offers an option to remove an individual's information from the system.

### 3. Person Detail Input:

#### Input Fields:

- **Name:** A field where users can enter or modify an individual's name.
- **Role:** Allows users to specify the role of the individual, such as Employee, Professor, or Student.
- **ID:** Where users can input or change the identification number of an individual.

The screenshot shows a Java Swing application window titled "My Profie". On the left, there is a sidebar with buttons for "Manage Faculty", "Manage Students", "Rart Chart", "My Profile", and "Log out". The main area is titled "Person" and contains a table with columns "ID", "Name", and "Role". The table data is as follows:

ID	Name	Role
1	John Doe	employee
2	Alice Smith	Employee
3	Professor Smith	Professor
4	Student Johnson	Student
5	Professor Brown	Professor
6	p	Employee
9	student 2	Student
10	teststu	Student
11	testpro	Professor
10	teststu	Student
10	teststu	Student

Below the table are buttons for "Search ID", "Search", "View", "Create", "Update", and "Delete". At the bottom, there are fields for "Name" (containing "Professor Smith"), "Role" (containing "Professor"), and "ID" (containing "3").

#### PersonScheduleComboBoxAreaJPanel:

View and make changes to your personal schedule or appointments.

### 1. Schedule Overview:

**Table:**

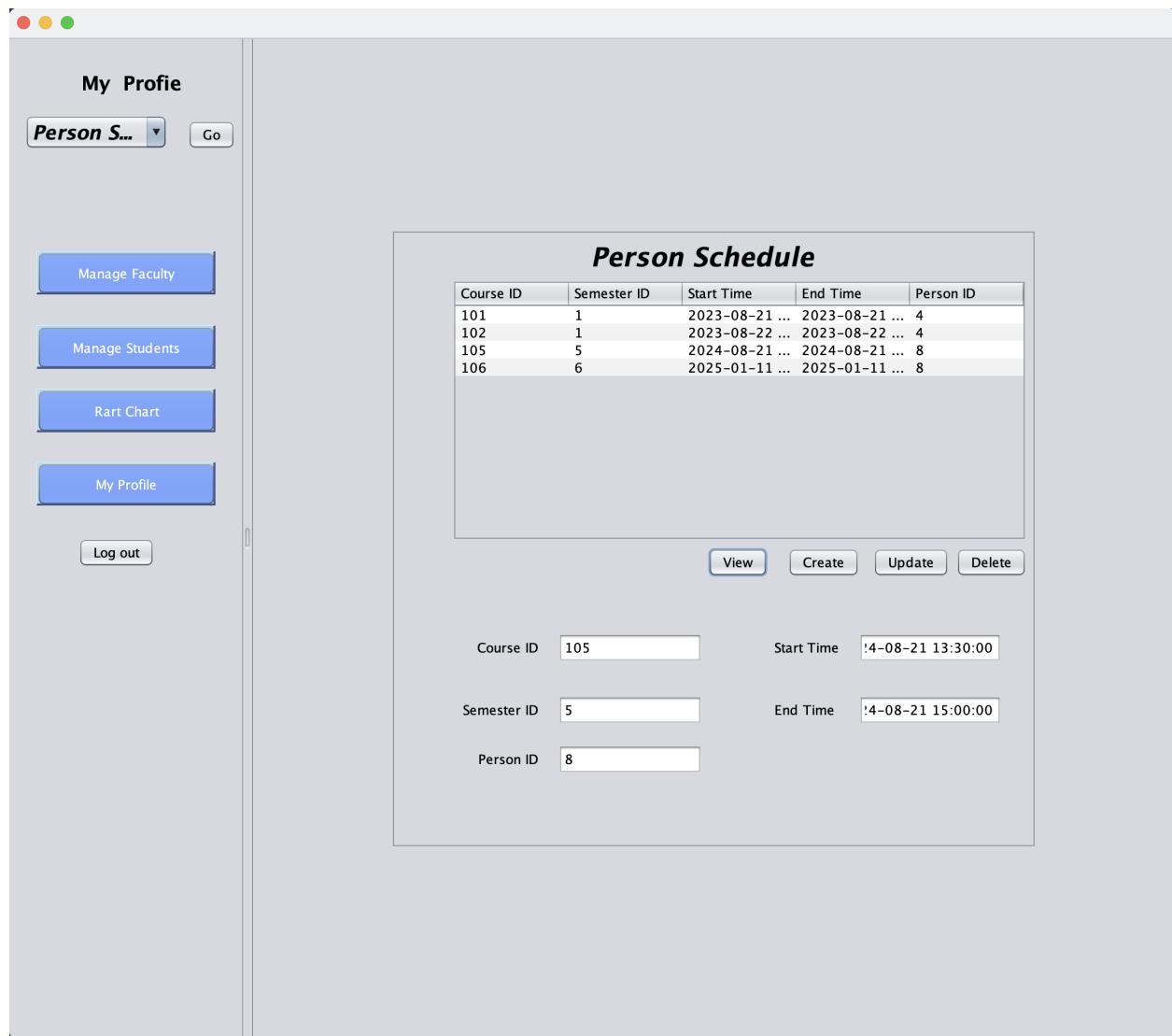
- **Course ID:** Displays the identification numbers of various courses.
- **Semester ID:** Indicates the semester in which the course is taking place.
- **Start Time:** Specifies the commencement time of the course.
- **End Time:** Indicates the conclusion time of the course.
- **Person ID:** Displays the identification numbers of the individuals associated with the courses.

**2. Course Detail Actions:****Buttons:**

- **View:** Enables the user to view comprehensive information about a specific course.
- **Create:** Provides an option for the user to add a new course schedule to the system.
- **Update:** Allows the user to make modifications to the details of an existing course schedule.
- **Delete:** Offers the capability to remove a course schedule from the system.

**3. Course Detail Input:****Input Fields:**

- **Course ID:** A field where users can input or alter the identification number of a course.
- **Semester ID:** Allows users to specify or modify the semester of the course.
- **Start Time:** Provides space for users to input or adjust the commencement time of the course.
- **End Time:** A field where users can set or change the conclusion time of the course.
- **Person ID:** Enables users to enter or modify the identification number of the individual associated with the course.



## RateComboBoxAreaJPanel:

Navigate to view or modify rates, perhaps related to courses or services.

### 1. Rating Overview:

#### Table:

- **Course ID:** Displays the identification numbers of the courses being rated.
- **Professor ID:** Indicates the ID of the professor associated with the course.
- **Student ID:** Specifies the ID of the student giving the rating.
- **Score:** Shows the aggregate score or rating given to the course.

### 2. Course Rating Actions:

#### Buttons:

- **View:** Allows the user to view detailed ratings and comments for a specific course.
- **Create:** Provides the user with the option to input a new rating for a course.
- **Update:** Lets the user modify an existing rating or comment.

- **Delete:** Gives the option to remove a particular rating or comment.

### 3. Course Rating Input:

#### Input Fields:

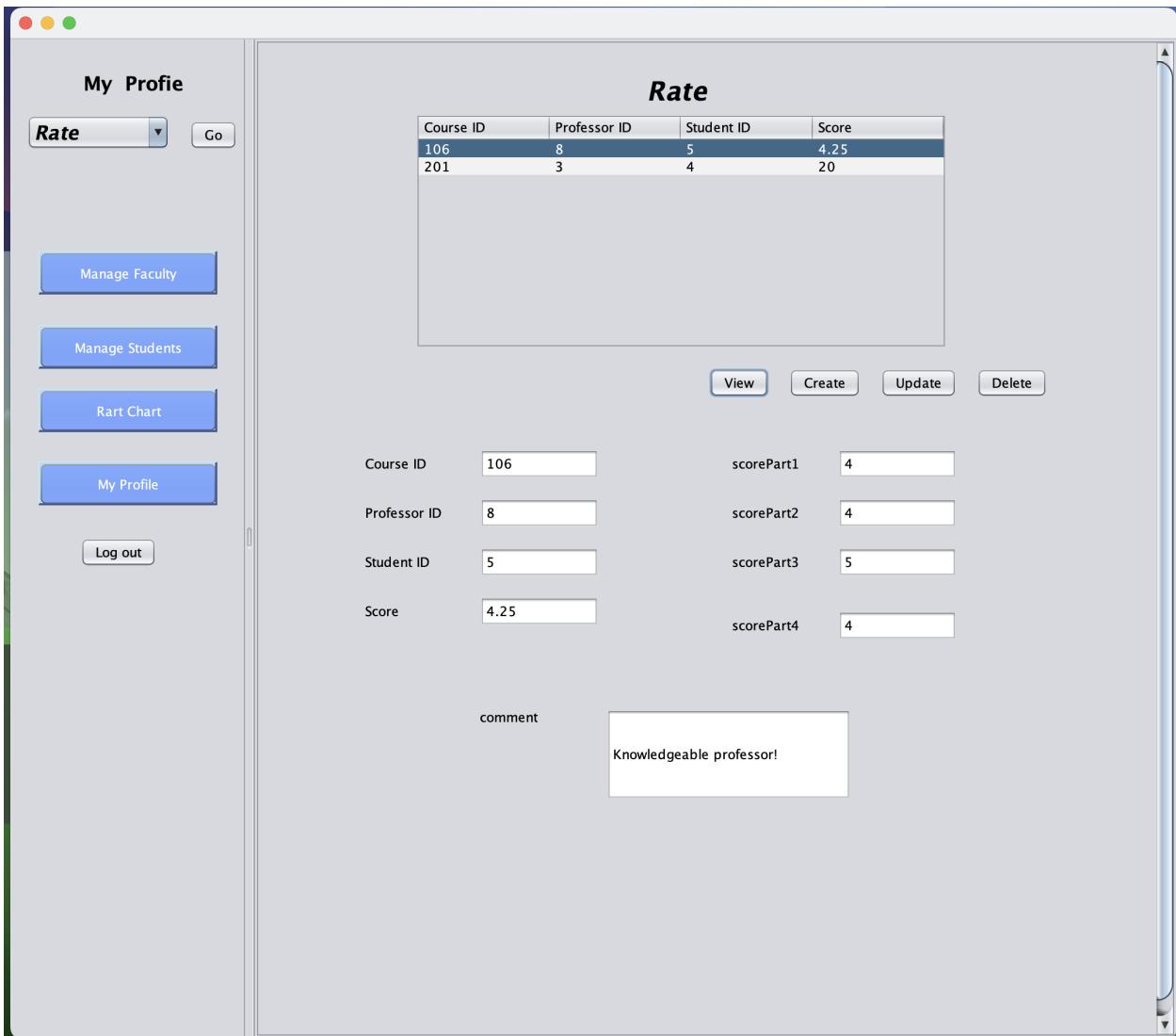
- **Course ID:** A space where users can input the identification number of the course they wish to rate.
- **Professor ID:** Allows users to specify the ID of the associated professor.
- **Student ID:** A field where users can mention their student identification.
- **Score:** Enables users to input an aggregate score for the course based on various rating parts.

#### Sub-scores:

- **scorePart1:** Represents a rating component where users can provide a score.
- **scorePart2:** Another component for users to rate.
- **scorePart3:** A third rating section.
- **scorePart4:** A fourth segment for course rating.

#### Comment Box:

- **comment:** Provides a space for users to leave feedback or comments about the course, complementing their score.



## SemesterComboBoxAreaJPanel:

Access to view, plan, or modify academic semester details.

### 1. Semester Overview:

#### Table:

- **ID:** Displays the unique identification numbers assigned to each semester.
- **Name:** Lists the specific names or titles of the semesters, such as "Fall 2023" or "Spring 2024".
- **Start Time:** Indicates the beginning date and time of the semester.
- **End Time:** Specifies the ending date and time of the semester.

### 2. Semester Actions:

#### Input Field:

- **Search ID:** Allows the user to input the ID of a specific semester they wish to search for.

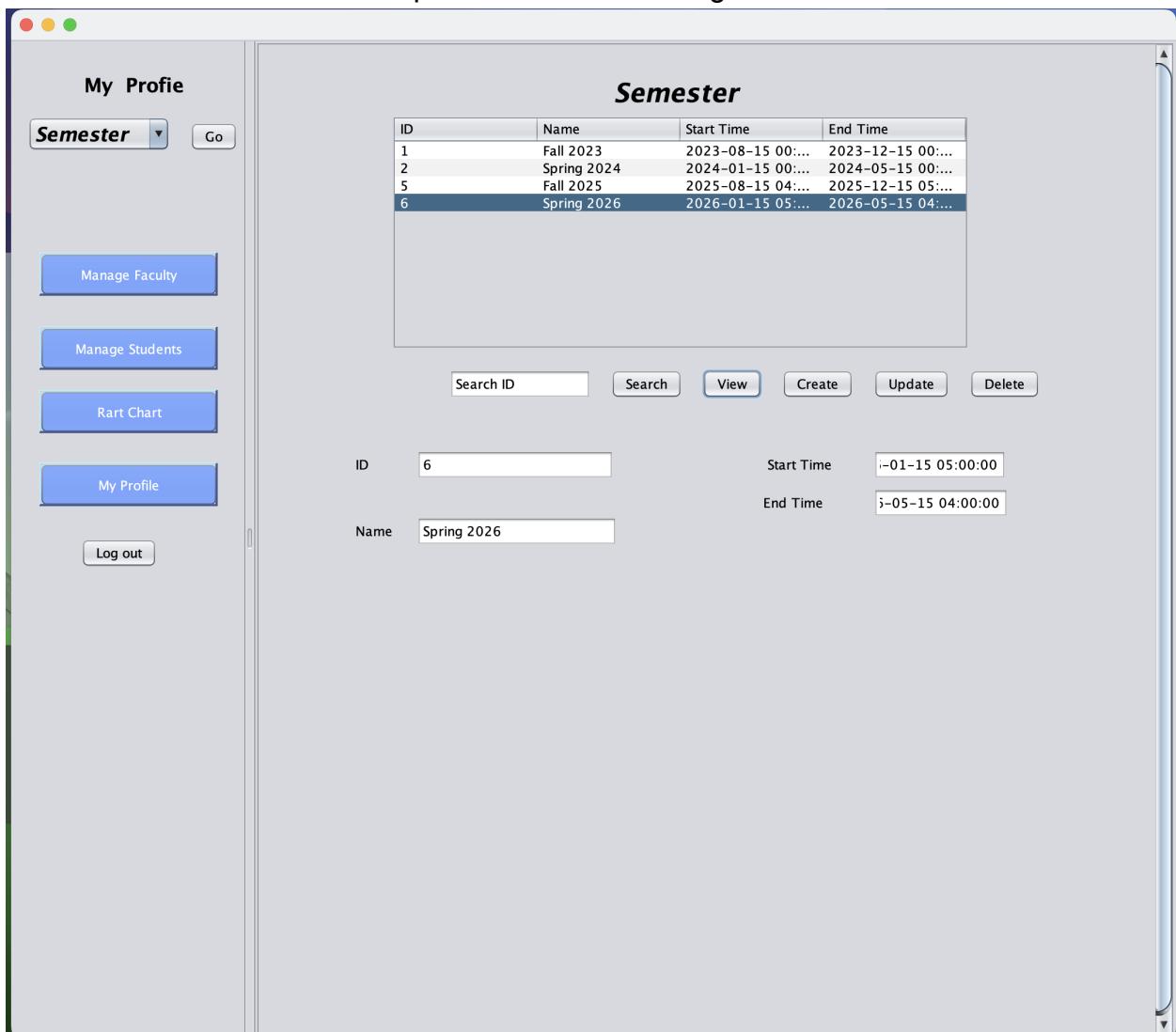
#### Buttons:

- **Search:** Lets the user find the details of a semester by its ID.
- **View:** Provides the user with the option to see the complete details of a selected semester.
- **Create:** Enables users to input details for a new semester.
- **Update:** Allows the user to modify the details of an existing semester.
- **Delete:** Gives the option to remove the record of a particular semester from the system.

### 3. Semester Details Input:

#### Input Fields:

- **ID:** A space where users can input or view the identification number of the semester.
- **Name:** Allows users to specify or view the name of the semester.
- **Start Time:** A field where users can input or see the starting date and time of the semester.
- **End Time:** Enables users to input or check the ending date and time of the semester.



#### StudentComboBoxArea JPanel:

Enter the student section to oversee, communicate, or modify student data.

## 1. Student Overview:

### Table:

- **ID**: Displays the unique identification numbers assigned to each student.
- **Name**: Lists the specific names of the students.
- **isEnabled**: Indicates whether the student's profile is active (true) or not (false).
- **GPA**: Represents the Grade Point Average of the student.

## 2. Student Actions:

### Input Field:

- **Search ID**: Allows the user to input the ID of a specific student they wish to search for.

### Buttons:

- **Search**: Enables the user to find the details of a student by their ID.
- **Add**: Gives the option to add a new student's details to the system.
- **View**: Provides the user with the option to see the complete details of a selected student.
- **Update**: Allows the user to modify the details of an existing student.
- **Disable**: Lets the user deactivate a student's profile.

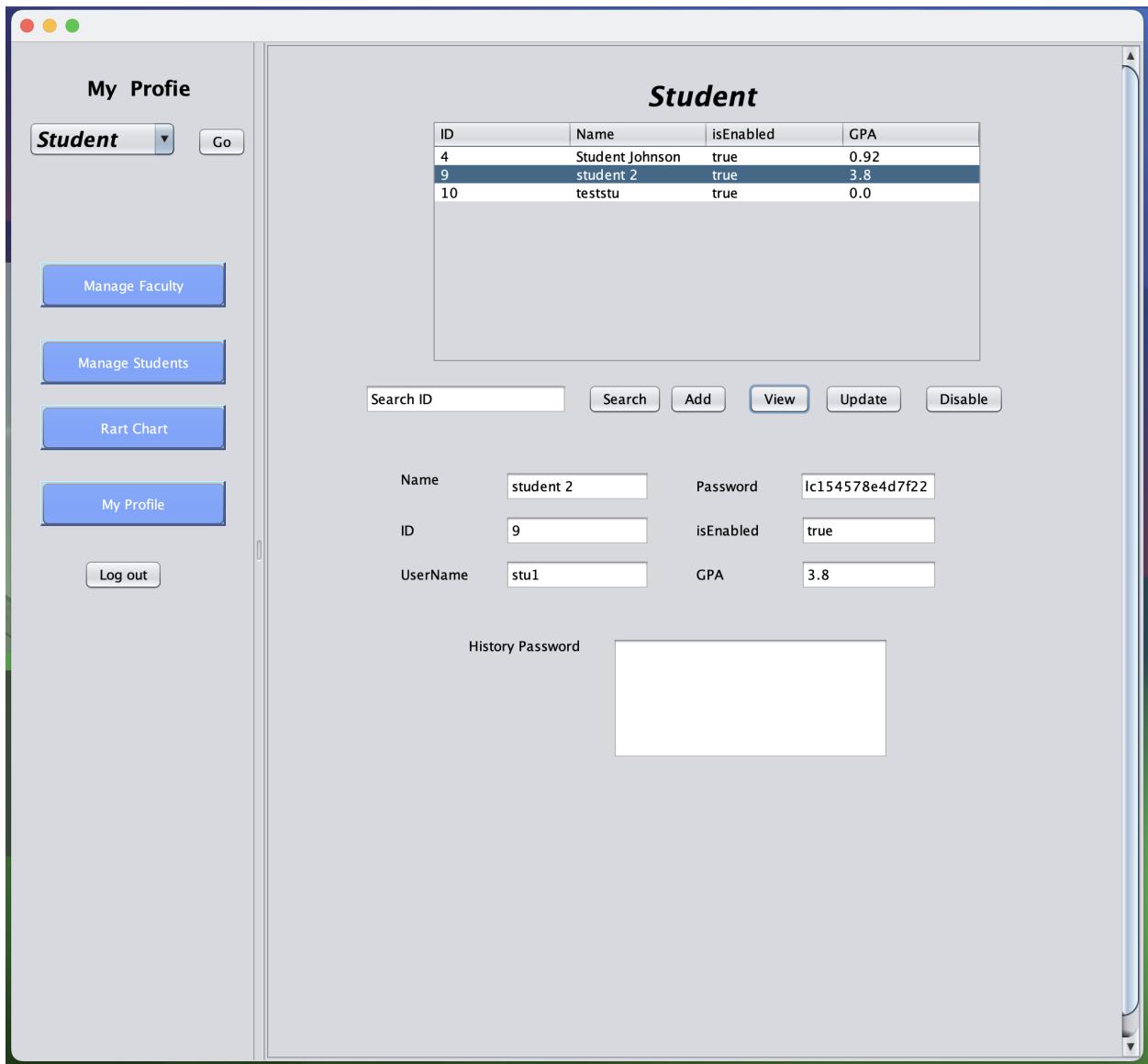
## 3. Student Details Input:

### Input Fields:

- **Name**: A space where users can input or view the name of the student.
- **ID**: Allows users to specify or view the unique identification number of the student.
- **UserName**: Enables users to input or check the username associated with the student's profile.
- **Password**: Field for users to input or view the password associated with the student's account.
- **isEnabled**: Checkbox or input that lets users know or set whether the student's profile is active or not.
- **GPA**: A field where users can input or view the student's Grade Point Average.

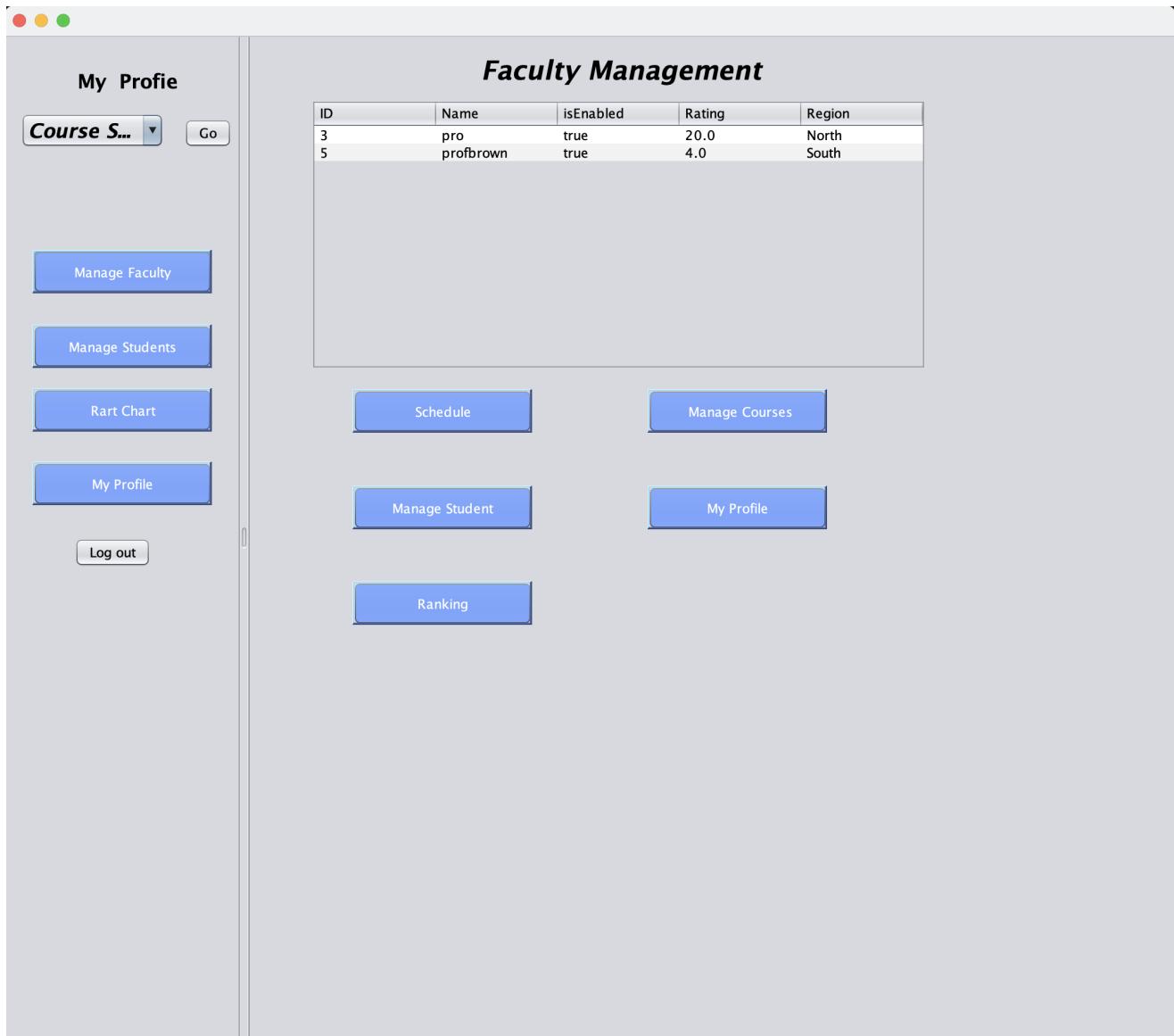
### Text Area:

- **History Password**: Provides a space for users to view the password history associated with the student's profile.



### 3. FacultyManagementJPanel:

Engage with the faculty management panel to oversee, collaborate, and manage faculty-related activities and resources.



## 1. Faculty Navigation Panel:

- **Schedule:** Click to view your course timetable.
- **Manage Courses:** Enter the course management page to view, create, and edit courses.
- **Manage Student:** Click to manage student information related to you.
- **My Profile:** View and edit your personal information.
- **Ranking:** View your teaching ranking or evaluations.
- **Log out:** Click to log out of the system.

## 2. MyScheduleJPanel:

**Dropdown Menu:** Allows the user to select the desired semester.

**Information Label:** Displays the semester's time range.

**Table:**

- **Course Name:** Lists the names of the courses for the selected semester.
- **Location:** Provides the room or venue where the course will take place.
- **Course Time:** Specifies the days and times the courses will be held.
- **Semester:** Indicates the academic semester for the listed courses.

#### **Button:**

- **Add To Calendar:** Allows the user to add the displayed courses to their personal calendar.

### **3. ManageCoursesJPanel:**

#### **Tabs:**

- **Current Course:** Displays the current courses for a selected semester.
- **Create Course:** Allows users to create a new course.

#### **Dropdown Menu:**

- **Select Semester:** Enables the user to select the desired semester to view the courses.

#### **Table (Current Course Tab):**

- **Course ID:** Lists the ID of the courses.
- **Course Name:** Specifies the name of each course.
- **Course Topic:** Describes the main subject or topic of each course.
- **Time:** Provides the timing details for each course.

#### **Search Bar:**

- **Course Name/Course Topic:** Allows users to search for courses based on name or topic.

#### **Form Fields (Both Tabs):**

- **Course ID:** Input for the course's identification number.
- **Course Location:** Input for the location or room where the course will be conducted.
- **Course Topic:** Input for the main subject or topic of the course.
- **Course Name:** Input for the name of the course.
- **Course point:** Input for the credit or point value of the course.
- **Course Begintime:** Input for the start time of the course.
- **Course Endtime:** Input for the end time of the course.
- **Student Limited:** Input for the maximum number of students allowed in the course.
- **Student Count:** Input to display the current number of enrolled students.
- **Semester:** Input for the academic semester.

- **Course Introduction:** Text area to provide a detailed introduction or description of the course.
- **Professor:** Input for the name of the course instructor.
- **Professor Rating in this class:** Input to display or update the professor's rating for this particular course.

#### **Student List (Current Course Tab):**

- Columns for **Student Name**, **Student NUID**, and **Student Score**. Allows for viewing and editing student details for a specific course.

#### **Buttons:**

- **Search:** Initiates the search based on the input in the search bar.
  - **Save:** Saves any changes made to course details.
  - **Create** (in Create Course Tab): Creates a new course based on the inputted details.
  - **View** (in Create Course Tab): Views the created or updated course details.
- 

## **4. ManageStudent JPanel**

#### **Search Section:**

- **Search Bar:** Allows users to input search criteria.
- **Search Button:** Initiates the search based on the criteria entered in the search bar.

#### **Student Listing:**

- **Student ID:** Displays the unique identification number of the student.
- **Student Name:** Shows the full name of the student.
- **If Choose my course:** Checkbox indicating if the student has chosen the user's course.

#### **Student Details Section:**

- **Student Name:** Input field for the student's full name.
- **Student ID:** Input field for the student's unique identification number.
- **Student GPA:** Input field for the student's Grade Point Average.
- **View Button:** Allows users to view detailed information about the student.
- **Information Label:** A label that informs users they can only view details about their own students.

## **5. MyProfile JPanel**

#### **Personal Information:**

- **Professor ID:** Displays the unique identification number of the professor.
- **Professor Name:** Shows the full name of the professor.
- **Professor Language:** Indicates the language proficiency or preferred language of the professor.

#### Courses Taught:

- **Course Name:** Lists the name of the courses taught by the professor.
- **Course Time:** Shows the scheduled time for each course.
- **Course Rating:** Provides the average rating or feedback score given to the course by students.

#### Rating and Region:

- **Professor Rate:** Displays the overall rating or feedback score given to the professor by students.
- **Professor Region:** Indicates the geographical region or campus location where the professor is based.

#### Login Credentials:

- **Username:** Shows the username used by the professor to log into the system.
- **Professor Password:** Input field for the professor's password.

#### Action Button:

- **Save:** Button to save any changes or updates made to the profile.
- 

## 6. Ranking JPanel

#### Professor Rankings:

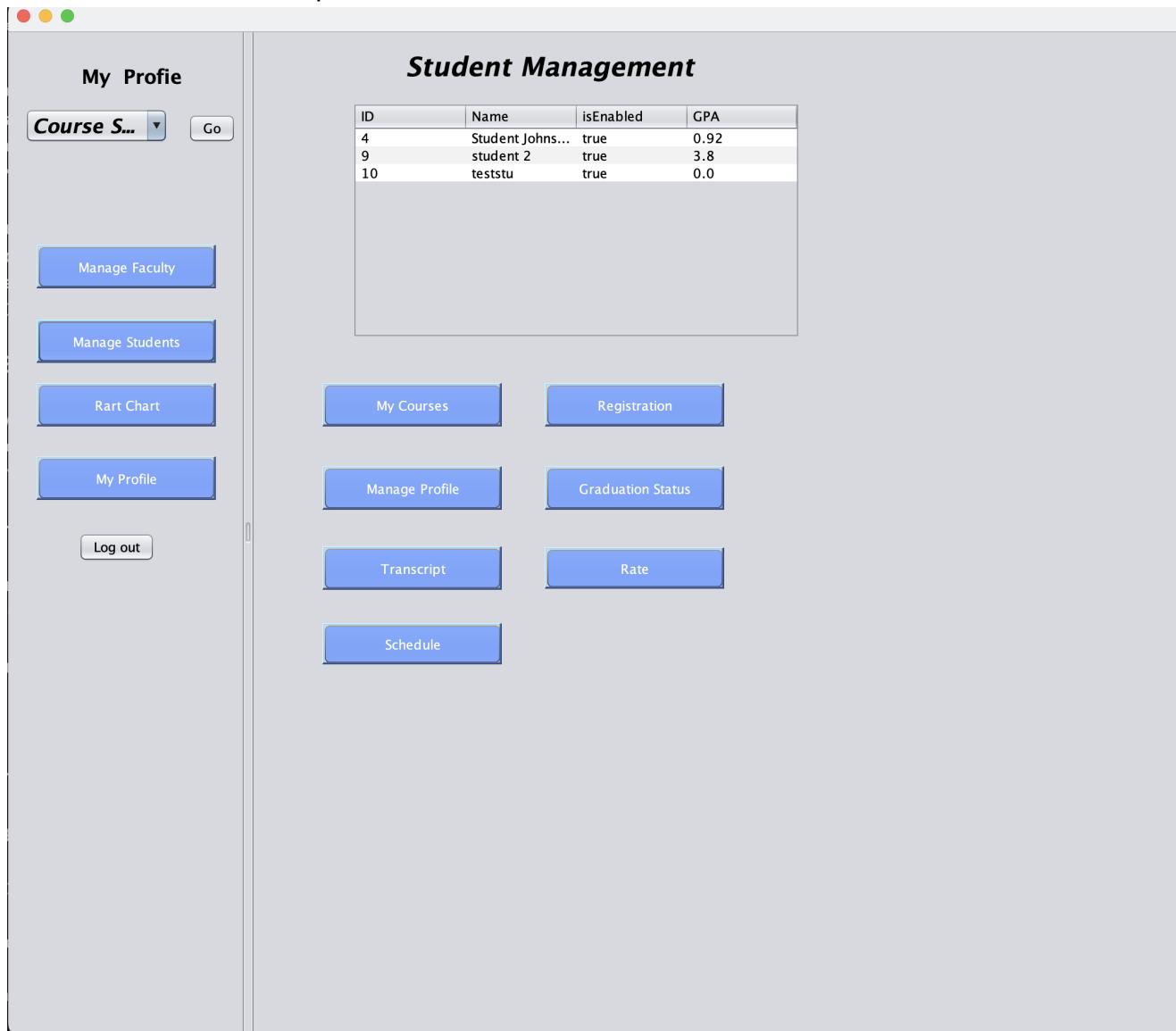
- **Professor Ranking:** Lists the numerical rank of the professor based on score.
- **Professor Name:** Specifies the name of the professor being ranked.
- **Professor Score:** Provides the score or feedback given to the professor, usually by students or peers.

#### Course Rankings:

- **Course Ranking:** Displays the numerical rank of courses.
- **Course Name:** Lists the name of the courses in ranked order.
- **Professor Name (for Course):** Shows the name of the professor teaching the ranked course.
- **Course Score:** Provides the score or feedback given to the course by students.

## 4. StudentManagementJPanel:

Access the student management area to oversee, communicate with, or make necessary modifications to student profiles, data, or academic status.



### 1. Student Navigation Panel:

- **Schedule:** Click to view your personal course timetable.
- **My Courses:** Access your current course list for the semester.
- **Registration:** Navigate to the course registration page to enroll in new courses.
- **Manage Profile:** View and edit your personal and academic details.
- **Graduation Status:** Check your academic progress and any pending requirements for graduation.
- **Transcript:** View your academic transcript with grades and credits.

- **Rate:** Provide feedback and ratings for your courses and professors.

- **Log out:** Click to securely exit the system.

## 2. MyScheduleJPanel:

**Dropdown Menu:** Allows you to select the relevant academic semester.

**Information Label:** Displays the date range for the chosen semester.

**Table:**

- **Course Name:** Lists the names of enrolled courses for the selected semester.
- **Location:** Shows the designated room or venue for each course session.
- **Course Time:** Indicates the specific days and time slots when the courses are scheduled.
- **Semester:** Highlights the academic term for the mentioned courses.

**Button:**

- **Add To Calendar:** Enables students to synchronize the listed schedule with their personal calendar for better time management.

## 3. My Course JPanel

**Current Course Selection:**

**Dropdown Menu:** Offers students the ability to choose the relevant academic semester.

**Search Bar:** Allows students to search for specific courses by entering the Professor's Name, Language, Region, Course Name, or Course Topic.

**Table:**

- **Course ID:** Displays the unique identification number assigned to each course.
- **Course Name:** Enumerates the titles of the courses available for the selected semester.
- **Professor:** Names the academic in charge of the respective course.
- **Course Status:** Informs whether the course is open for enrollment or otherwise.
- **Score:** Indicates the student's academic performance in the given course (if applicable).
- **Semester:** Denotes the academic term for the depicted courses.

**Detailed Course Information:**

**Fields:**

- **Course ID:** Enter or view the identification number for the course.
- **Course Topic:** Input or observe the main topic or theme of the course.

- **Course Name:** Type in or see the official title of the course.
- **Course Credit:** Specify or check the number of credits awarded upon successful completion of the course.
- **Semester:** Enter or review the semester when the course is offered.
- **Course Location:** Fill in or view the designated room or venue for the course sessions.
- **Course Begintime:** Indicate or look up the starting time of the course sessions.
- **Course Endtime:** Enter or view the ending time of the course sessions.
- **Student Limited:** Define or see the maximum number of students allowed to enroll in the course.
- **Student Count:** Check the current number of students enrolled in the course.
- **Course Introduction:** Provides a brief overview or summary of what the course entails.
- **Professor:** Enter or view the name of the professor instructing the course.
- **Professor Region:** Specify or check the region or origin of the professor.
- **Professor Language:** Enter or view the primary language used by the professor.

#### **Button:**

- **Drop:** Allows students to unenroll from the selected course.

## **4. Registration Page:**

#### **Course Selection:**

**Search Bar:** Enables students to search for specific courses using parameters such as Course ID, Course Name, Professor Name, Professor Topic, Professor Language, and Course Region.

#### **Table:**

- **Course Id:** Displays the unique identification number for each course.
- **Course Name:** Lists the names of the available courses for registration.
- **Professor Name:** Names the instructor responsible for each course.
- **Professor Topic:** Describes the main academic focus or area of specialization of the professor for the course.
- **Professor Language:** Specifies the primary teaching language used by the professor.
- **Course Region:** Indicates the geographical or institutional region where the course is offered.

#### **Detailed Course Information:**

#### **Fields:**

- **Course ID:** Allows entry or displays the unique identifier for the selected course.
- **Course Name:** Facilitates entry or displays the title of the chosen course.

- **Semester:** Offers a field to input or view the academic term when the course is offered.
- **Professor:** Provides space to enter or view the name of the instructor responsible for the course.
- **Professor Rating:** Lets students view the average rating or feedback given to the professor by former students.
- **Course Location:** Gives a field to input or view the designated venue or room for the course.
- **Course Introduction:** Contains a textbox for a brief overview or description of the course content and objectives.
- **Student Limited:** Displays the maximum number of students allowed to enroll in the course.
- **Student Count:** Shows the current number of students registered for the course.

#### **Button:**

- **Enroll:** Permits students to register for the selected course from the list.
- 

## **5.Student Profile Panel:**

### **Personal Information:**

#### **Fields:**

- **Student ID:** Displays the unique identification number of the student.
- **Student Name:** Shows the full name of the student.

### **Academic Record:**

#### **Table:**

- **Course Name:** Lists the names of the courses the student is enrolled in or has completed.
- **Professor Name:** Provides the name of the professor who taught or is teaching the course.
- **Student Score:** Displays the score or grade the student received for each course.
- **Semester:** Indicates the academic term in which the student took or is taking the course.

### **Performance Metric:**

#### **Fields:**

- **GPA:** Shows the student's current Grade Point Average on a scale, typically from 0 to 4 or 5.

## **Account Settings:**

### **Fields:**

- **Username:** Displays the student's username for accessing the academic platform or system.
- **Password:** Provides a masked field to protect the student's password. Only dots or asterisks are visible, ensuring privacy.

### **Buttons:**

- **Save:** Allows the student to save any changes they've made to their profile.
- **Change:** Enables the student to modify or update their password.

## **6. Graduation Status:**

**Information Label:** States the requirement for graduation, specifically mentioning the number of courses a student must complete before they can graduate.

### **Fields:**

- **Courses Completed:** Displays the number of courses the student has completed so far.

### **Graduation Actions:**

#### **Button:**

- **Checking Status:** Allows the user to verify their current academic status and see how many more courses they need to complete for graduation.
- **Apply for Graduation:** Enables the student to officially apply for graduation once they've met the required criteria. This button may be active or inactive depending on the student's academic status.

## **##### Student Transcript Panel:**

## **7. Transcript Overview:**

**Title:** Clearly labels the section as "Transcript".

### **Table:**

- **Semester:** Shows the academic semester in which the course was taken.
- **Course:** Lists the names of the courses the student has taken during the specified semester.
- **Professor:** Specifies the instructor or professor who taught the course.
- **Score:** Displays the grade or score the student received for each course.

### **Information Label:**

- **GPA:** Represents the Grade Point Average of the student based on the scores from the courses listed. Displays the cumulative GPA value.

### **Actions:**

#### **Button:**

- **Print:** Allows the user to print a physical copy or save a digital copy of their transcript.

## **8.Student Rate Panel:**

### **Course & Professor Rating Table:**

- **Course Name:** Lists the names of the courses that the student is eligible to rate.
- **Professor Name:** Specifies the instructor or professor of the respective course.
- **Rate:** Indicates the current rating status of the course. If not rated, will display "Need Rate".
- **Score:** Displays the student's score or grade for the respective course. If not graded, will display "NO Score Yet".

**\*\*Rating Criteria**

#### **Rating Questions\*\*:**

- **Course Materials:** Asks if the professor provided course materials, such as lecture notes, presentations, and reading materials, on time. Rate ranges from 0 to 5.
- **Feedback and Guidance:** Inquires if appropriate feedback and guidance were given to students. Rate ranges from 0 to 5.
- **Grading Promptness:** Checks if the professor provided grades and feedback promptly. Rate ranges from 0 to 5.
- **Relevance and Application:** Asks if the professor possesses relevant academic or industry experience and effectively integrates theory with real-world applications. Rate ranges from 0 to 5.

**\*\*Feedback Section:**

**Comment Box\*\*:** Allows the student to provide additional feedback or comments regarding the course or professor.

**\*\*Actions**

#### **Button\*\*:**

- **Submit:** Enables the student to submit their ratings and feedback for the courses and professors listed.

- **View:** (If applicable) Allows the student to view detailed ratings and comments from other students or view their own previous ratings.
- 

## 5. RateChartJPanel:

Navigate through this panel to view, analyze, or adjust the rate charts, possibly related to course fees, faculty compensation, or other services.

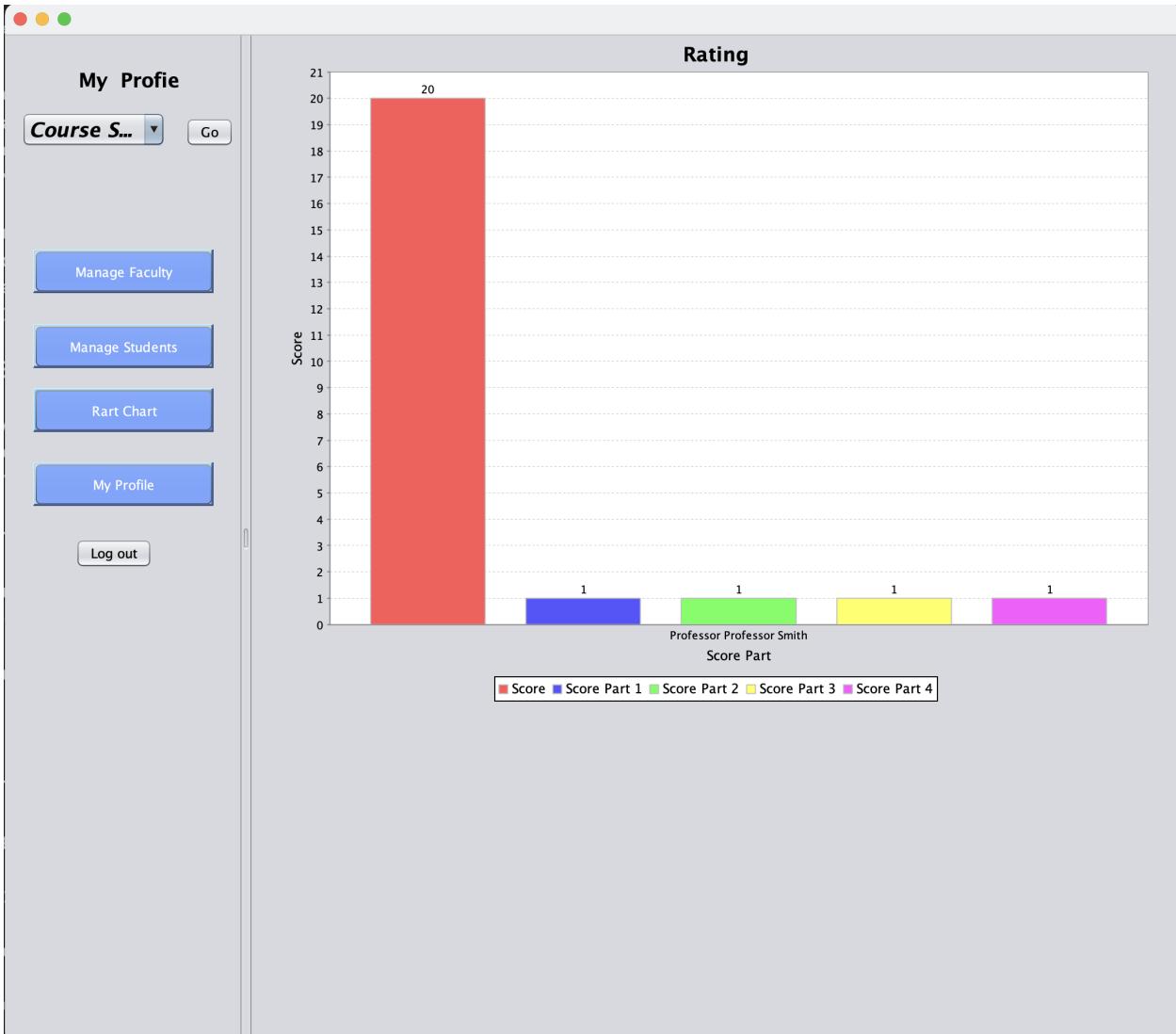
### Bar Chart:

- **Score Axis:** Represents the score given, ranging from 0 to 21.
- **Professor Names:** Displays the names of the professors, e.g., "Professor Professor Smith".
- **Score Parts:** Differentiates the scores based on four categories: Score, Score Part 1, Score Part 2, Score Part 3, and Score Part 4.

### Color Legend:

- **Score:** Red color represents the overall score for the professor.
- **Score Part 1:** Blue color indicates the score for the first part.
- **Score Part 2:** Purple color signifies the score for the second part.
- **Score Part 3:** Yellow color depicts the score for the third part.

- **Score Part 4:** Green color marks the score for the fourth part.



## 6. MyProfileJPanel:

Navigate through this panel to view, analyze, or adjust the rate charts, possibly related to course fees, faculty compensation, or other services.

### Labels and Input Fields:

- **Employee ID:** Displays the identification number of the admin or employee.
- **Employee Name:** Allows the user to view or edit their name.
- **Username:** Shows the current username used for login.
- **Password:** An input field to view or change the current password.

### Button:

- **Save:** Enables the user to save any changes made to their profile.

The screenshot shows a web application window titled "My Profile". On the left, there is a sidebar with the following buttons:

- Course S... (dropdown menu)
- Go
- Manage Faculty
- Manage Students
- Rank Chart
- My Profile** (highlighted in blue)
- Log out

The main content area is titled "My Profile" and contains the following form fields:

Employee I...	<input type="text" value="1"/>
Employee Name:	<input type="text" value="John Doe"/>
Username:	<input type="text" value="admin"/>
Password:	<input type="password"/>

A "Save" button is located at the bottom right of the form area.

## Faculty WorkArea:

### 1. Left Navigation Panel:

- **Left Navigation Panel:**
  - **Schedule:** Click to view your course timetable.
  - **Manage Courses:** Enter the course management page to view, create, and edit courses.
  - **Manage Student:** Click to manage student information related to you.
  - **My Profile:** View and edit your personal information.
  - **Ranking:** View your teaching ranking or evaluations.

- **Log out:** Click to log out of the system.

# **My Faculty Profie**

Schedule

Manage Courses

Manage Student

My Profile

Ranking

Log out

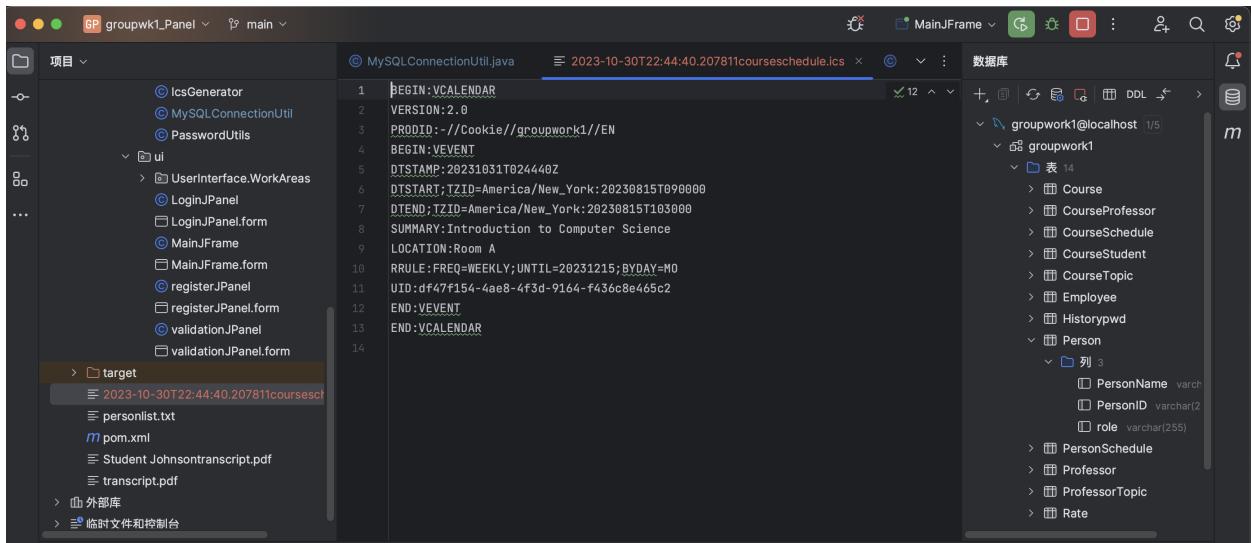
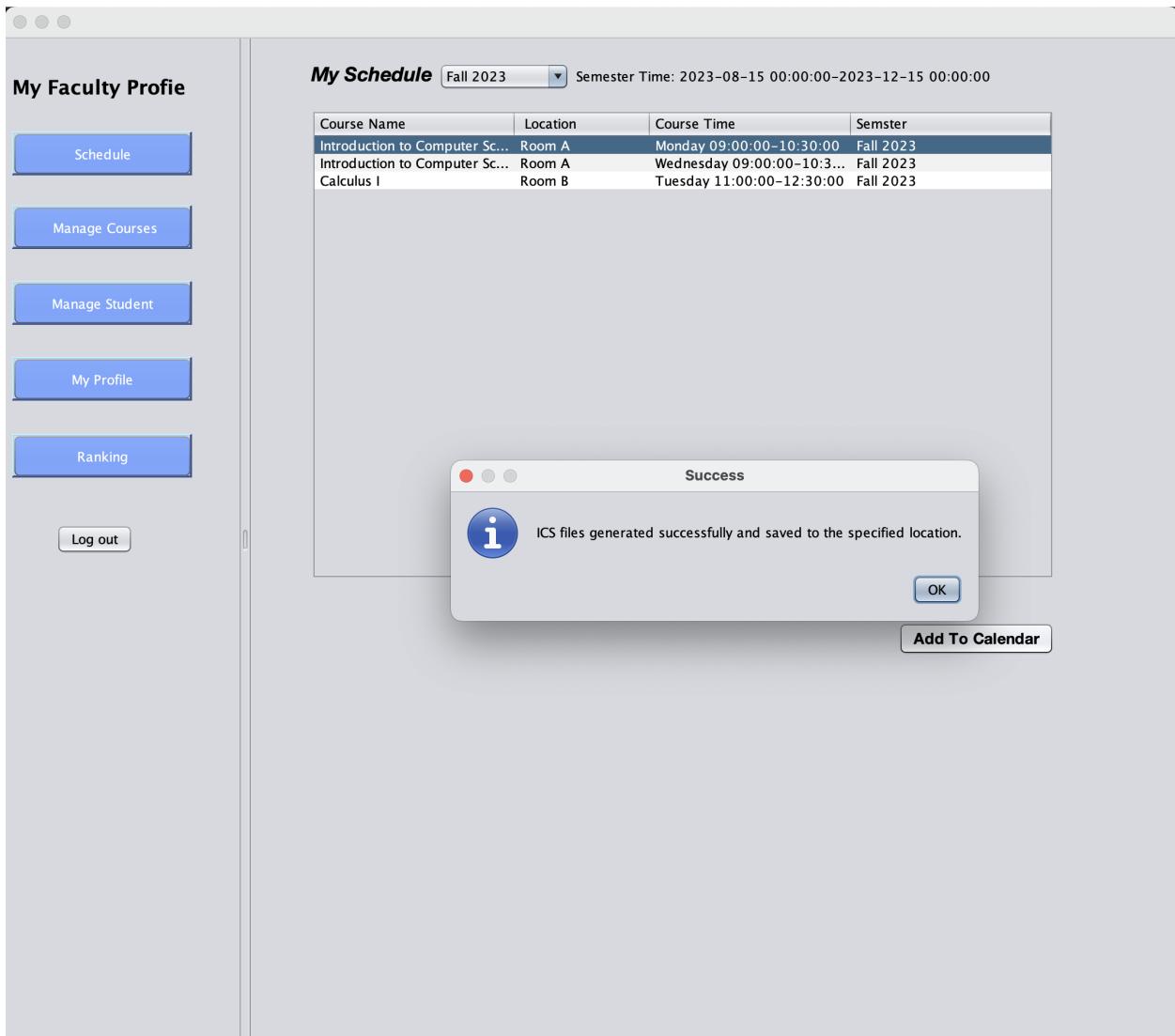
## **2. MyScheduleJPanel:**

**Dropdown Menu:** Allows the user to select the desired semester.

**Information Label:** Displays the semester's time range.

### **Table:**

- **Course Name:** Lists the names of the courses for the selected semester.
- **Location:** Provides the room or venue where the course will take place.
- **Course Time:** Specifies the days and times the courses will be held.
- **Semester:** Indicates the academic semester for the listed courses.
- **Button:**
- **Add To Calendar:** Allows the user to add the displayed courses to their personal calendar.



### 3. ManageCoursesJPanel:

#### Tabs:

- **Current Course:** Displays the current courses for a selected semester.
- **Create Course:** Allows users to create a new course.

#### Dropdown Menu:

- **Select Semester:** Enables the user to select the desired semester to view the courses.

### Table (Current Course Tab):

- **Course ID:** Lists the ID of the courses.
- **Course Name:** Specifies the name of each course.
- **Course Topic:** Describes the main subject or topic of each course.
- **Time:** Provides the timing details for each course.

### Search Bar:

- **Course Name/Course Topic:** Allows users to search for courses based on name or topic.
- **Form Fields (Both Tabs):**
  - **Course ID:** Input for the course's identification number.
  - **Course Location:** Input for the location or room where the course will be conducted.
  - **Course Topic:** Input for the main subject or topic of the course.
  - **Course Name:** Input for the name of the course.
  - **Course point:** Input for the credit or point value of the course.
  - **Course Begintime:** Input for the start time of the course.
  - **Course Endtime:** Input for the end time of the course.
  - **Student Limited:** Input for the maximum number of students allowed in the course.
  - **Student Count:** Input to display the current number of enrolled students.
  - **Semester:** Input for the academic semester.
  - **Course Introduction:** Text area to provide a detailed introduction or description of the course.
  - **Professor:** Input for the name of the course instructor.
  - **Professor Rating in this class:** Input to display or update the professor's rating for this particular course.

### Student List (Current Course Tab):

- Columns for **Student Name**, **Student NUID**, and **Student Score**. Allows for viewing and editing student details for a specific course.

### Buttons:

- **Search:** Initiates the search based on the input in the search bar.
- **Save:** Saves any changes made to course details.
- **Create** (in Create Course Tab): Creates a new course based on the inputted details.

- **View** (in Create Course Tab): Views the created or updated course details.

**My Faculty Profie**

Schedule

Manage Courses

Manage Student

My Profile

Ranking

Log out

**Manage Courses**

**Current Course** **Create Course**

Select Semester: Fall 2023 Semester Time: 2023-08-15 00:00:00~2023-12-15 00:00:00

Course ID	Couse Name	Course Topic	Time
101	Introduction to Computer S...	[Programming Fundamentals]	Monday 09:00:00-10:30:00
101	Introduction to Computer S...	[Programming Fundamentals]	Wednesday 09:00:00-10:30:00
102	Calculus I	[Limits and Derivatives]	Tuesday 11:00:00-12:30:00

Course Name/Course Topic  Search

Course ID: 101 Course Location: Room A

Course Topic: [Programming Fundamentals] Course Time: 09:00:00-10:30:00

Course Name: Introduction to Computer Science Course Weekday: Monday

Course point: 3 Student Limited: 50

Semester: Fall 2023 Student Count: 1

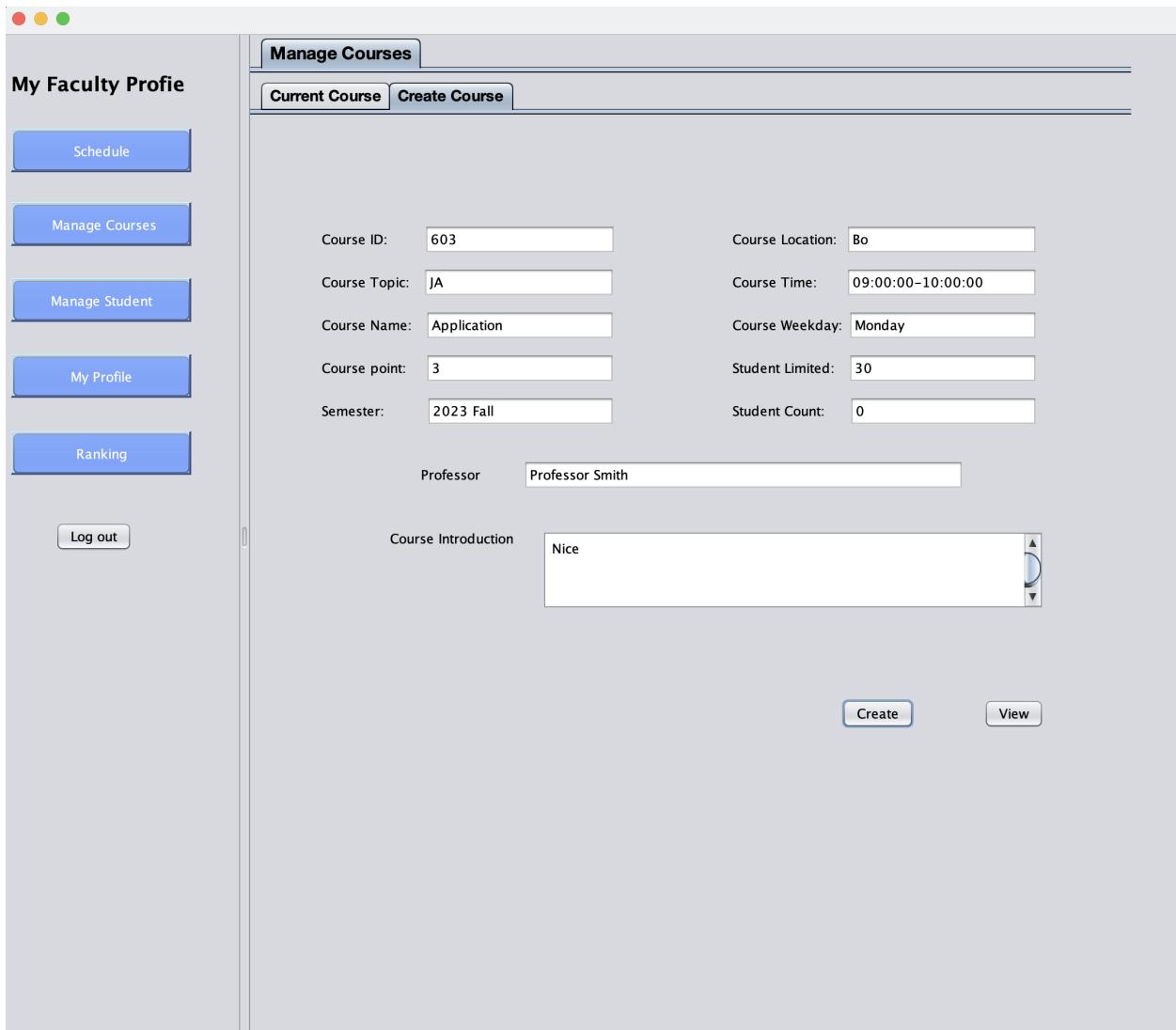
Course Introduction: Introductory course for CS majors

Professor: Professor Smith Professor Rating in this class:

Student List:

Student Name	Student NUID	Student Score
Student Johnson	4	44.0

**Save**



## 4. ManageStudent JPanel

### Search Section:

- **Search Bar:** Allows users to input search criteria.
- **Search Button:** Initiates the search based on the criteria entered in the search bar.

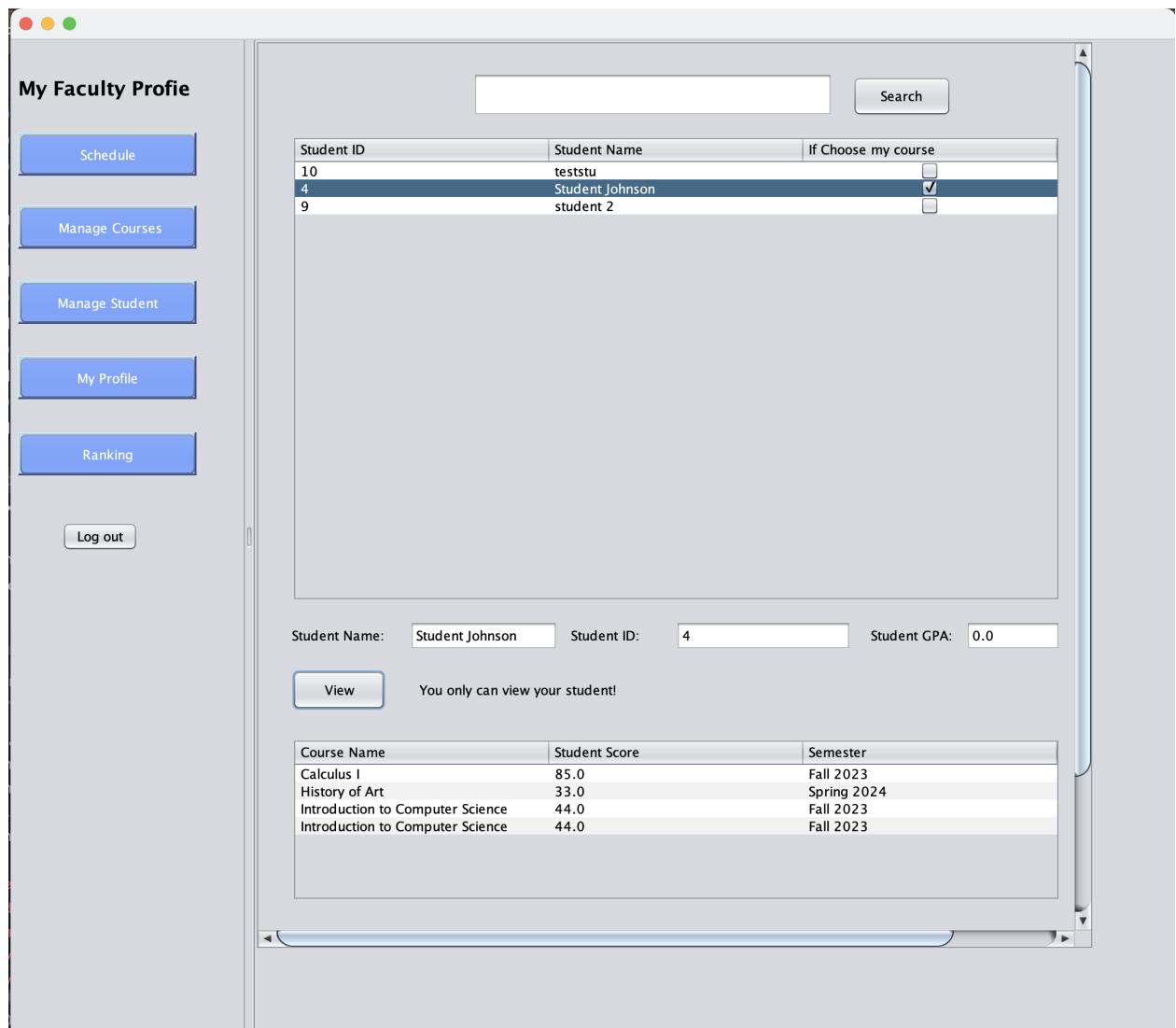
### Student Listing:

- **Student ID:** Displays the unique identification number of the student.
- **Student Name:** Shows the full name of the student.
- **If Choose my course:** Checkbox indicating if the student has chosen the user's course.

### Student Details Section:

- **Student Name:** Input field for the student's full name.
- **Student ID:** Input field for the student's unique identification number.
- **Student GPA:** Input field for the student's Grade Point Average.

- **View Button:** Allows users to view detailed information about the student.
- **Information Label:** A label that informs users they can only view details about their own students.



## 5. MyProfileJPanel

### Personal Information:

- **Professor ID:** Displays the unique identification number of the professor.
- **Professor Name:** Shows the full name of the professor.
- **Professor Language:** Indicates the language proficiency or preferred language of the professor.
- **Courses Taught:**
- **Course Name:** Lists the name of the courses taught by the professor.
- **Course Time:** Shows the scheduled time for each course.
- **Course Rating:** Provides the average rating or feedback score given to the course by students.
- **Rating and Region:**
- **Professor Rate:** Displays the overall rating or feedback score given to the professor by students.

- **Professor Region:** Indicates the geographical region or campus location where the professor is based.

### Login Credentials:

- **Username:** Shows the username used by the professor to log into the system.
- **Professor Password:** Input field for the professor's password.

### Action Button:

- **Save:** Button to save any changes or updates made to the profile.

The screenshot shows a Java Swing application window titled "My Faculty Profie". On the left, there is a vertical sidebar with several buttons: "Schedule", "Manage Courses", "Manage Student", "My Profile" (which is highlighted in blue), and "Ranking". Below these buttons is a "Log out" button. The main content area is titled "My Profile" and contains the following fields:

Professor ID:	<input type="text" value="3"/>															
Professor Name:	<input type="text" value="Professor Smith"/>															
Professor Language:	<input type="text" value="English"/>															
Courses	<table border="1"> <thead> <tr> <th>Course Name</th> <th>Course Time</th> <th>Course Rating</th> </tr> </thead> <tbody> <tr> <td>Calculus I</td> <td>Tuesday 11:00:00-12:30...</td> <td>0.0</td> </tr> <tr> <td>History of Art</td> <td>Thursday 14:00:00-16:0...</td> <td>20.0</td> </tr> <tr> <td>Introduction to Computer S...</td> <td>Monday 09:00:00-10:30...</td> <td>0.0</td> </tr> <tr> <td>Introduction to Computer S...</td> <td>Wednesday 09:00:00-10....</td> <td>0.0</td> </tr> </tbody> </table>	Course Name	Course Time	Course Rating	Calculus I	Tuesday 11:00:00-12:30...	0.0	History of Art	Thursday 14:00:00-16:0...	20.0	Introduction to Computer S...	Monday 09:00:00-10:30...	0.0	Introduction to Computer S...	Wednesday 09:00:00-10....	0.0
Course Name	Course Time	Course Rating														
Calculus I	Tuesday 11:00:00-12:30...	0.0														
History of Art	Thursday 14:00:00-16:0...	20.0														
Introduction to Computer S...	Monday 09:00:00-10:30...	0.0														
Introduction to Computer S...	Wednesday 09:00:00-10....	0.0														
Professor Rate:	<input type="text" value="20.0"/>															
Professor Region:	<input type="text" value="North"/>															
Username:	<input type="text" value="pro"/>															
Professor Password:	<input type="password"/>															

At the bottom right of the main content area is a "Save" button.

## 6.Ranking JPanel

### Professor Rankings:

- **Professor Ranking:** Lists the numerical rank of the professor based on score.
- **Professor Name:** Specifies the name of the professor being ranked.
- **Professor Score:** Provides the score or feedback given to the professor, usually by students or peers.

## Course Rankings:

- **Course Ranking:** Displays the numerical rank of courses.
- **Course Name:** Lists the name of the courses in ranked order.
- **Professor Name (for Course):** Shows the name of the professor teaching the ranked course.
- **Course Score:** Provides the score or feedback given to the course by students.

The screenshot shows a user interface for a faculty member. On the left, there's a sidebar titled "My Faculty Profie" with buttons for "Schedule", "Manage Courses", "Manage Student", "My Profile", and "Ranking". The "Ranking" button is highlighted. At the bottom of the sidebar is a "Log out" button. The main content area is titled "Ranking" and contains two tables. The top table, titled "Professor Ranking", lists two rows: 1 Professor Smith (Score 20.0) and 1 Professor Brown (Score 4.0). The bottom table, titled "Course Ranking", lists five rows: History of Art (Score 20.0), Chemistry 101 (Score 4.25), Database Management (Score 0.0), Calculus I (Score 0.0), and Introduction to Computer S... (Score 0.0).

Professor Ranking	Professor name	Professor Score
1	Professor Smith	20.0
1	Professor Brown	4.0

Course Ranking	Course Name	Professor name	Course Score
1	History of Art	Professor Smith	20.0
1	Chemistry 101		4.25
1	Database Management		0.0
1	Calculus I	Professor Smith	0.0
1	Introduction to Computer S...	Professor Smith	0.0

## Student WorkArea:

### 1. Left Navigation Panel:

- **Left Navigation Panel:**
  - **Schedule:** Click to view your personal course timetable.
  - **My Courses:** Access your current course list for the semester.
  - **Registration:** Navigate to the course registration page to enroll in new courses.
  - **Manage Profile:** View and edit your personal and academic details.

- **Graduation Status:** Check your academic progress and any pending requirements for graduation.
- **Transcript:** View your academic transcript with grades and credits.
- **Rate:** Provide feedback and ratings for your courses and professors.
- **Log out:** Click to securely exit the system.



# **My Student Profie**

Schedule

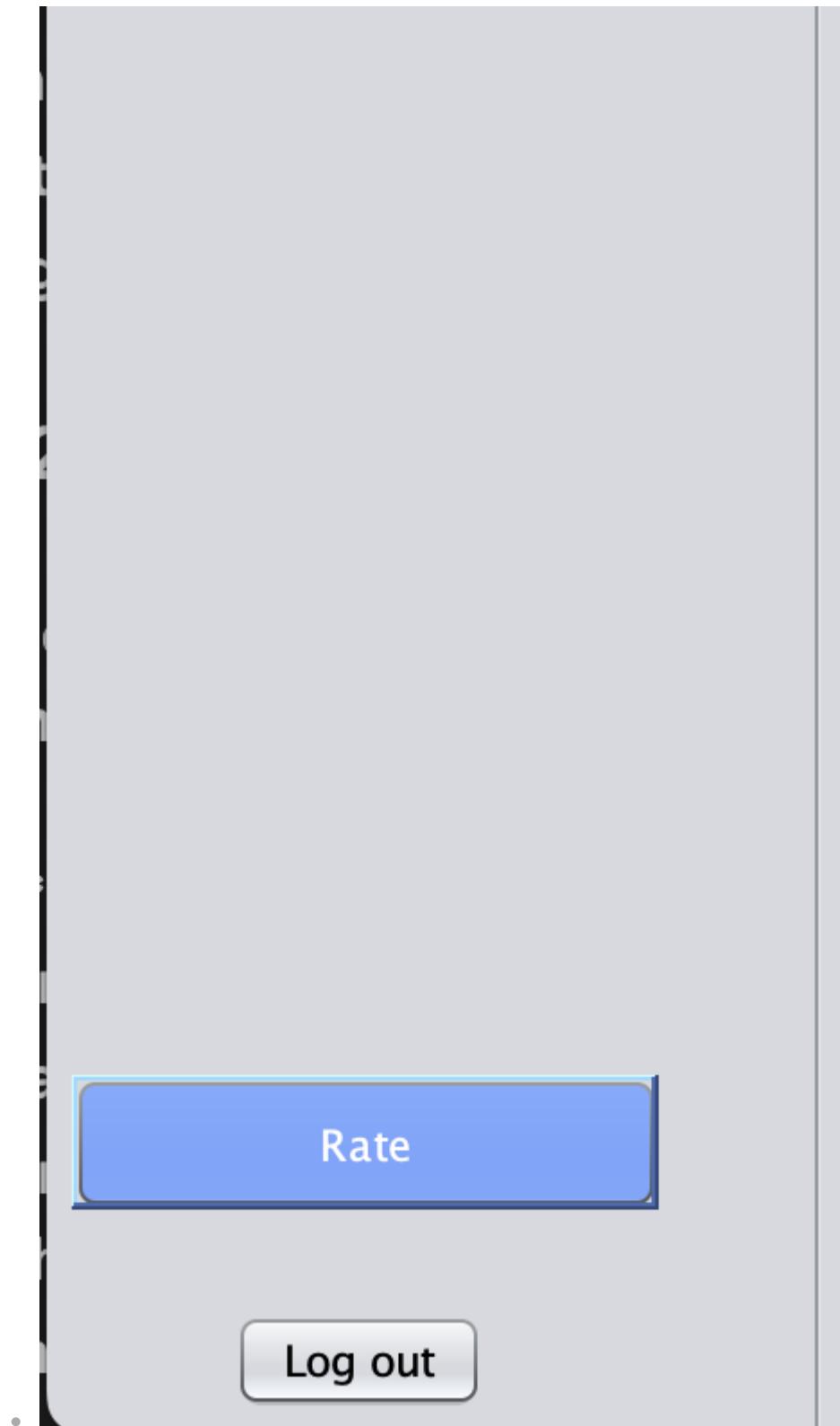
My Courses

Registration

Manage Profile

Graduation Status

Transcript



## 2. MySchedule JPanel:

**Dropdown Menu:** Allows you to select the relevant academic semester.

**Information Label:** Displays the date range for the chosen semester.

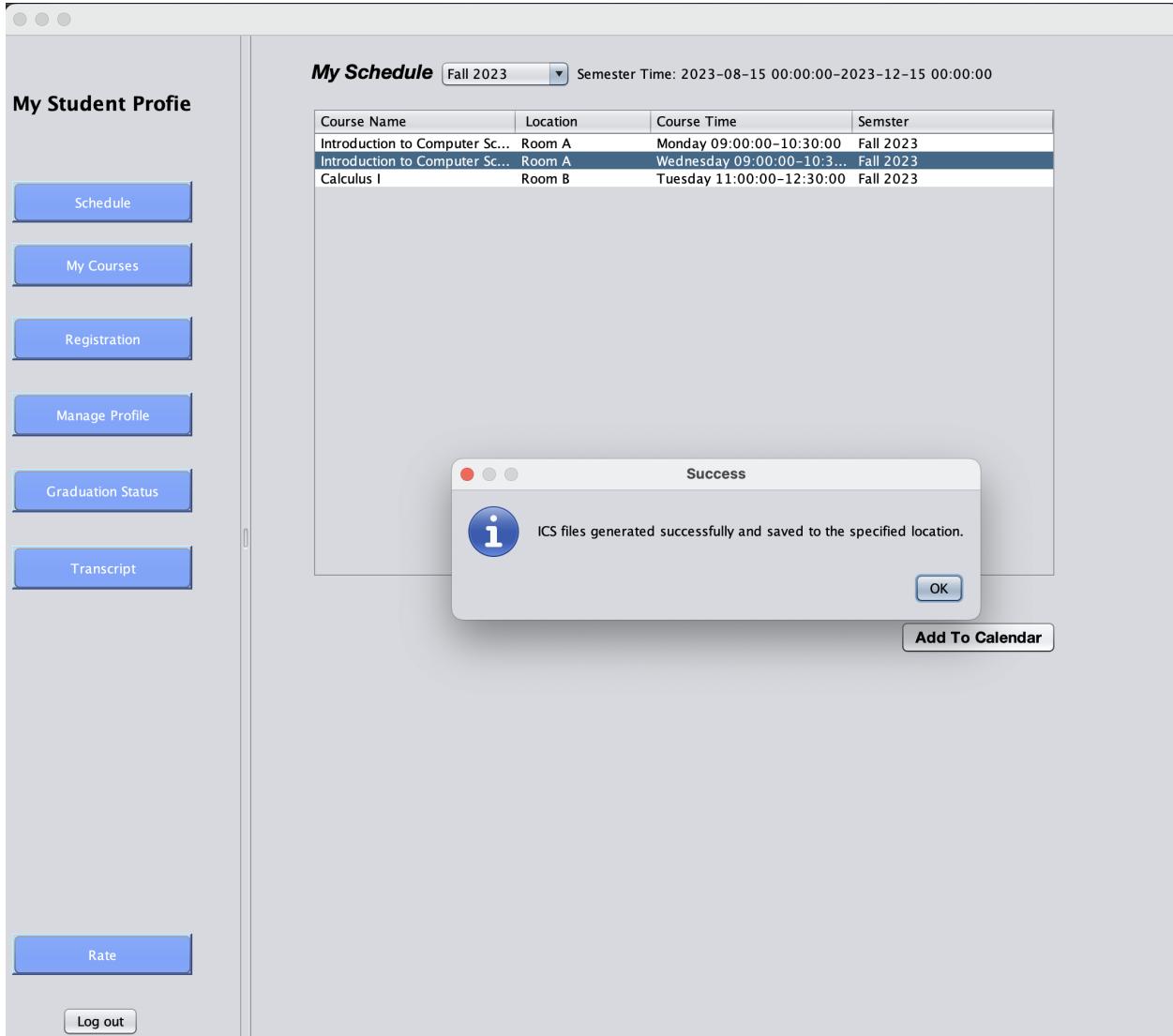
### Table:

- **Course Name:** Lists the names of enrolled courses for the selected semester.
- **Location:** Shows the designated room or venue for each course session.

- **Course Time:** Indicates the specific days and time slots when the courses are scheduled.
- **Semester:** Highlights the academic term for the mentioned courses.

### Button:

- **Add To Calendar:** Enables students to synchronize the listed schedule with their personal calendar for better time management.



The screenshot shows a database interface with a project tree on the left and a database browser on the right. The project tree includes 'IcsGenerator', 'MySQLConnectionUtil', 'PasswordUtils', 'ui' (containing 'UserInterface.WorkAreas', 'LoginJPanel', 'MainJFrame', 'registerJPanel', 'registerJPanel.form', 'validationJPanel', 'validationJPanel.form'), and 'target' (containing '2023-10-30T22:44:40.207811courseschedule.ics' and '2023-10-30T22:53:16.756038courseschedule.ics'). The database browser shows a schema for 'groupwork1' with tables like 'Course', 'CourseProfessor', 'CourseSchedule', 'CourseStudent', 'CourseTopic', 'Employee', 'Historypwd', 'Person', and 'PersonSchedule'. The '2023-10-30T22:44:40.207811courseschedule.ics' file content is displayed in the center:

```

1 BEGIN:VCALENDAR
2 VERSION:2.0
3 PRODID:-//Cookie//groupwork1//EN
4 BEGIN:VEVENT
5 DTSTART;TZID=America/New_York:20230815T090000
6 DTEND;TZID=America/New_York:20230815T103000
7 SUMMARY:Introduction to Computer Science
8 LOCATION:Room A
9 RRULE:FREQ=WEEKLY;UNTIL=20231215;BYDAY=WE
10 UID:c5fea583-a931-40b4-aa9b-cb10a1b49ad6
11 END:VEVENT
12 END:VCALENDAR
13
14

```

### 3.My Course JPanel

#### Current Course Selection:

**Dropdown Menu:** Offers students the ability to choose the relevant academic semester.

**Search Bar:** Allows students to search for specific courses by entering the Professor's Name, Language, Region, Course Name, or Course Topic.

#### Table:

- **Course ID:** Displays the unique identification number assigned to each course.
- **Course Name:** Enumerates the titles of the courses available for the selected semester.
- **Professor:** Names the academic in charge of the respective course.
- **Course Status:** Informs whether the course is open for enrollment or otherwise.
- **Score:** Indicates the student's academic performance in the given course (if applicable).
- **Semester:** Denotes the academic term for the depicted courses.

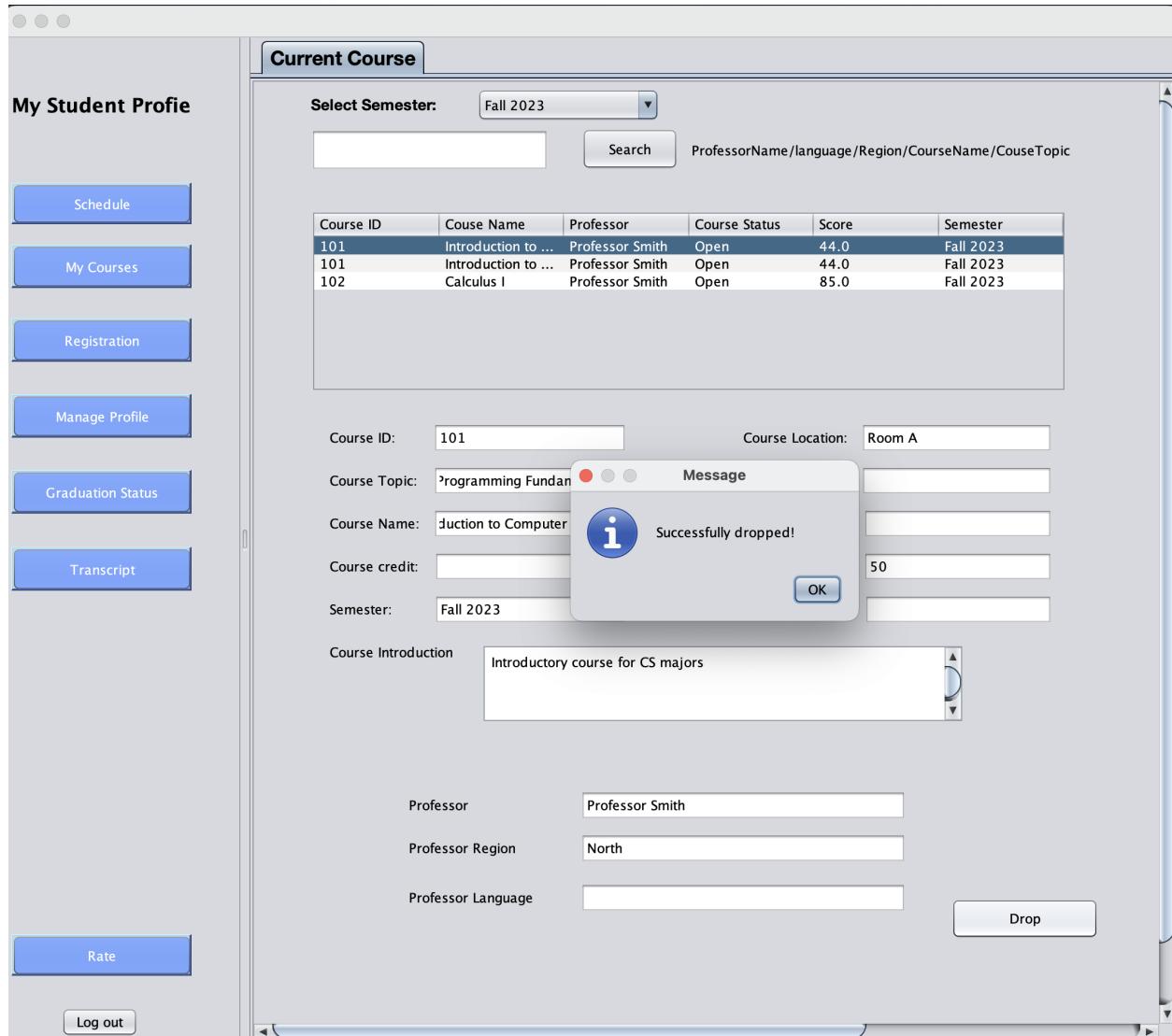
#### Detailed Course Information:

#### Fields:

- **Course ID:** Enter or view the identification number for the course.
- **Course Topic:** Input or observe the main topic or theme of the course.
- **Course Name:** Type in or see the official title of the course.
- **Course Credit:** Specify or check the number of credits awarded upon successful completion of the course.
- **Semester:** Enter or review the semester when the course is offered.
- **Course Location:** Fill in or view the designated room or venue for the course sessions.
- **Course Begintime:** Indicate or look up the starting time of the course sessions.
- **Course Endtime:** Enter or view the ending time of the course sessions.
- **Student Limited:** Define or see the maximum number of students allowed to enroll in the course.
- **Student Count:** Check the current number of students enrolled in the course.
- **Course Introduction:** Provides a brief overview or summary of what the course entails.
- **Professor:** Enter or view the name of the professor instructing the course.
- **Professor Region:** Specify or check the region or origin of the professor.
- **Professor Language:** Enter or view the primary language used by the professor.

#### Button:

- **Drop:** Allows students to unenroll from the selected course.



## 4. Registration Page:

### Course Selection:

**Search Bar:** Enables students to search for specific courses using parameters such as Course ID, Course Name, Professor Name, Professor Topic, Professor Language, and Course Region.

### Table:

- **Course Id:** Displays the unique identification number for each course.
- **Course Name:** Lists the names of the available courses for registration.
- **Professor Name:** Names the instructor responsible for each course.
- **Professor Topic:** Describes the main academic focus or area of specialization of the professor for the course.
- **Professor Language:** Specifies the primary teaching language used by the professor.
- **Course Region:** Indicates the geographical or institutional region where the course is offered.

### Detailed Course Information:

## Fields:

- **Course ID:** Allows entry or displays the unique identifier for the selected course.
- **Course Name:** Facilitates entry or displays the title of the chosen course.
- **Semester:** Offers a field to input or view the academic term when the course is offered.
- **Professor:** Provides space to enter or view the name of the instructor responsible for the course.
- **Professor Rating:** Lets students view the average rating or feedback given to the professor by former students.
- **Course Location:** Gives a field to input or view the designated venue or room for the course.
- **Course Introduction:** Contains a textbox for a brief overview or description of the course content and objectives.
- **Student Limited:** Displays the maximum number of students allowed to enroll in the course.
- **Student Count:** Shows the current number of students registered for the course.

## Button:

- **Enroll:** Permits students to register for the selected course from the list.

The screenshot shows a user interface for a student profile management system. On the left, a sidebar titled "My Student Profile" lists several menu items: Schedule, My Courses, Registration, Manage Profile, Graduation Status, Transcript, Rate, and Log out. The "Registration" item is highlighted. The main content area is titled "Registration". At the top, there is a search bar with the value "101" and a "Search" button. Below the search bar is a table showing course listings:

Course Id	Course Name	Professor Name	Professor Topic	Professor Language	Course Region
106	Chemistry 101	[Chemical Reactions]	Chemistry	West	
105	Database Manage...	[Relational Databas...	English	East	
101	Introduction to Co...	Professor Smith	[Programming Fun...	English	North

Below the table, there are several input fields for course registration:

- Course ID: 101
- Course Name: Introduction to Computer Science
- Semester: Fall 2023
- Professor: Professor Smith
- Professor Rating: (empty)
- Course Location: Room A
- Course Introduction: Introductory course for CS majors
- Student Limited: 50
- Student Count: (empty)

At the bottom right of the form is a large "Enroll" button.

---

## 5. Student Profile Panel:

### Personal Information:

#### Fields:

- **Student ID:** Displays the unique identification number of the student.
- **Student Name:** Shows the full name of the student.

### Academic Record:

#### Table:

- **Course Name:** Lists the names of the courses the student is enrolled in or has completed.
- **Professor Name:** Provides the name of the professor who taught or is teaching the course.
- **Student Score:** Displays the score or grade the student received for each course.
- **Semester:** Indicates the academic term in which the student took or is taking the course.

### Performance Metric:

#### Fields:

- **GPA:** Shows the student's current Grade Point Average on a scale, typically from 0 to 4 or 5.

### Account Settings:

#### Fields:

- **Username:** Displays the student's username for accessing the academic platform or system.
- **Password:** Provides a masked field to protect the student's password. Only dots or asterisks are visible, ensuring privacy.

### Buttons:

- **Save:** Allows the student to save any changes they've made to their profile.

- **Change:** Enables the student to modify or update their password.

The screenshot shows a student profile interface with a sidebar on the left containing links for Schedule, My Courses, Registration, Manage Profile, Graduation Status, and Transcript. The main area displays student information: Student ID (4), Student Name (Student Johnson), Courses (Calculus I, Professor Smith, 85.0, Fall 2023; History of Art, Professor Smith, 33.0, Spring 2024), GPA (1.71), Username (stu), and Password. There are Save and Change buttons for the password. At the bottom are Rate and Log out buttons.

Course Name	Professor name	Student Score	Semester
Calculus I	Professor Smith	85.0	Fall 2023
History of Art	Professor Smith	33.0	Spring 2024

## 6. Graduation Status:

**Information Label:** States the requirement for graduation, specifically mentioning the number of courses a student must complete before they can graduate.

**Fields:**

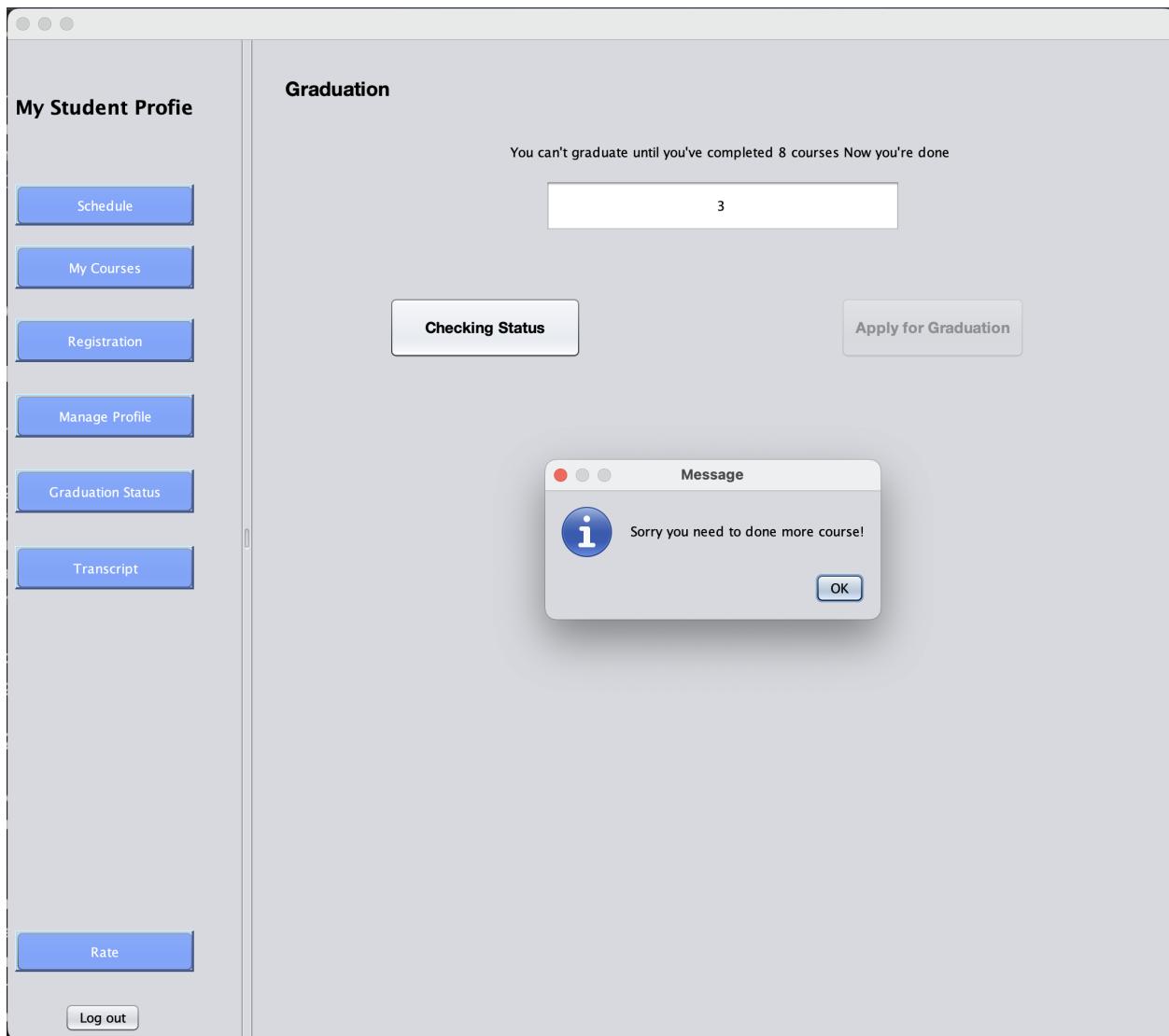
- **Courses Completed:** Displays the number of courses the student has completed so far.

**Graduation Actions:**

**Button:**

- **Checking Status:** Allows the user to verify their current academic status and see how many more courses they need to complete for graduation.
- **Apply for Graduation:** Enables the student to officially apply for graduation once they've met the required criteria. This button may be active or inactive depending on the

student's academic status.



## 7. Transcript Overview:

**Title:** Clearly labels the section as "Transcript".

**Table:**

- **Semester:** Shows the academic semester in which the course was taken.
- **Course:** Lists the names of the courses the student has taken during the specified semester.
- **Professor:** Specifies the instructor or professor who taught the course.
- **Score:** Displays the grade or score the student received for each course.

**Information Label:**

- **GPA:** Represents the Grade Point Average of the student based on the scores from the courses listed. Displays the cumulative GPA value.

## Actions:

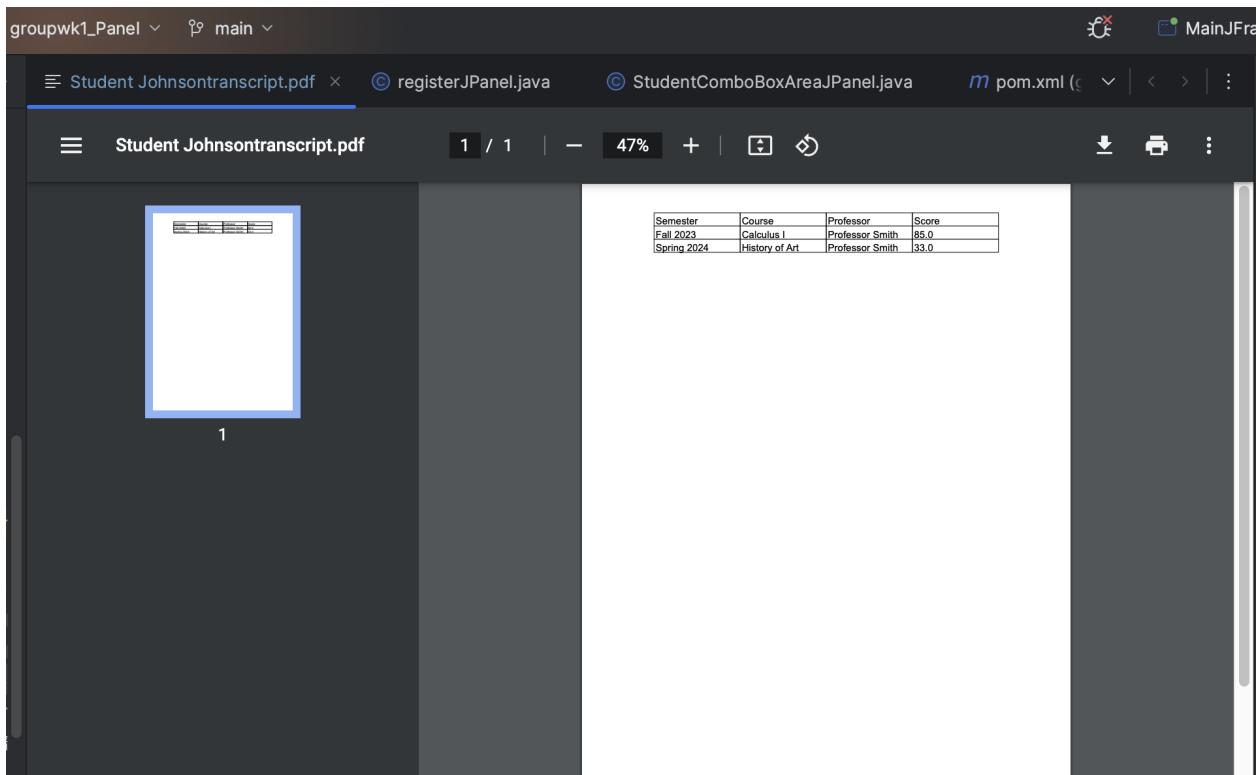
### Button:

- **Print:** Allows the user to print a physical copy or save a digital copy of their transcript.

The screenshot shows a student profile interface. On the left, a sidebar titled "My Student Profile" contains several buttons: Schedule, My Courses, Registration, Manage Profile, Graduation Status, Transcript, Rate, and Log out. The "Transcript" button is highlighted with a blue background. The main area is titled "Transcript" and displays a table with four columns: Semester, Course, Professor, and Score. The table has two rows: Fall 2023 (Calculus I, Professor Smith, 85.0) and Spring 2024 (History of Art, Professor Smith, 33.0). A "Print" button is located in the top right corner of the transcript area. At the bottom of the page, there is a text input field labeled "GPA:" containing the value "1.71".

Semester	Course	Professor	Score
Fall 2023	Calculus I	Professor Smith	85.0
Spring 2024	History of Art	Professor Smith	33.0

GPA:



## 8. Student Rate Panel:

### Course & Professor Rating Table:

- **Course Name:** Lists the names of the courses that the student is eligible to rate.
- **Professor Name:** Specifies the instructor or professor of the respective course.
- **Rate:** Indicates the current rating status of the course. If not rated, will display "Need Rate".
- **Score:** Displays the student's score or grade for the respective course. If not graded, will display "NO Score Yet".

### Rating Criteria

#### Rating Questions:

- **Course Materials:** Asks if the professor provided course materials, such as lecture notes, presentations, and reading materials, on time. Rate ranges from 0 to 5.
- **Feedback and Guidance:** Inquires if appropriate feedback and guidance were given to students. Rate ranges from 0 to 5.
- **Grading Promptness:** Checks if the professor provided grades and feedback promptly. Rate ranges from 0 to 5.
- **Relevance and Application:** Asks if the professor possesses relevant academic or industry experience and effectively integrates theory with real-world applications. Rate ranges from 0 to 5.

#### Feedback Section:

**Comment Box:** Allows the student to provide additional feedback or comments regarding the course or professor.

## Actions

Button:

- **Submit:** Enables the student to submit their ratings and feedback for the courses and professors listed.
- **View:** (If applicable) Allows the student to view detailed ratings and comments from other students or view their own previous ratings.

The screenshot shows a user interface for a digital educational platform. On the left, there's a sidebar with a "My Student Profile" section containing buttons for "Schedule", "My Courses", "Registration", "Manage Profile", "Graduation Status", and "Transcript". Below this is a "Rate" button and a "Log out" button. The main content area has a title "Rate" and a table showing course ratings. The table has columns for Course Name, Professor Name, Rate, and Score. It lists "Calculus I" with Professor Smith and a rate of 0, and "History of Art" with Professor Smith and a rate of 20. A "View" button is at the top right of the table. Below the table are four rating questions with input fields:

- Did the professor provide course materials, like lecture notes, presentations, and reading materials, on time? Rate from 0 to 5 (0 for No, 5 for Yes). Input: 1
- Were appropriate feedback and guidance provided to help students improve their academic performance? Rate from 0 to 5 (0 for No, 5 for Yes). Input: 1
- Did the professor provide grades and feedback promptly to help students understand their performance? Rate from 0 to 5 (0 for No, 5 for Yes). Input: 1
- Did the professor possess relevant academic or industry experience and effectively integrate theory with real-world applications? Rate from 0 to 5 (0 for No, 5 for Yes). Input: 1

A "Comment:" field contains the text "1". At the bottom is a "Submit" button.

## 6. Overall Assessment of the Digital Educational Platform

1. **User Interface (UI) Design:** The design is simplistic and straightforward. The use of tabs and sections makes it easy for students, professors, employee to navigate through their profiles and perform necessary tasks.
2. **Functionality:** The platform offer essential functions required by students and professors:
  - **Manage Courses:** Professor can create, publish, edit and delete courses, courses details.

- **My Schedule:** Choose your semester and directly show all information you need, even add to your Calendar.
- **Manage Student:** Through entering keywords to search for students, display student informations.
- **Course Registration:** Students can view available courses, course details, and enroll in them.
- **Profile Management:** Personal information, courses taken, and other relevant data can be viewed and edited.
- **Graduation Status:** This gives students an idea of their progress towards degree completion.
- **Transcript:** It provides an organized view of grades from different semesters.
- **Rate:** This section would showcase professor rankings based on metrics like course feedback, possibly student feedback, or other KPIs.

3. **Language and Region Customization:** The platform appears to consider language and regional preferences for both courses and professors, which can be beneficial for international students or those seeking courses in specific languages.

### **Opinion on Future Digital Educational Platforms:**

If designed and implemented effectively, digital educational platforms can have a significant positive impact on making education more accessible and affordable.

#### **1. Accessibility:**

- **Geographic Barriers:** Digital platforms remove geographic barriers, allowing students from remote areas to access quality education.
- **Flexibility:** They cater to different learning paces and schedules, benefiting working students or those with other commitments.
- **Diverse Offerings:** Online platforms can host a plethora of courses from various fields, giving students a wider range of options.

#### **2. Affordability:**

- **Reduced Costs:** There are fewer overhead costs associated with digital platforms compared to traditional educational settings, potentially leading to reduced tuition fees.
- **Resource Sharing:** E-books, online journals, and shared resources can reduce the cost of textbooks and materials.

#### **3. Inclusivity for the Less Fortunate:**

- **Scholarships & Discounts:** Online platforms can provide scholarships or discounted courses for students from less privileged backgrounds.
- **Free Courses:** Many platforms offer free courses, ensuring that finances are not a barrier to learning.
- **Micro-Credits:** Some platforms allow students to take small, modular courses and accumulate credits over time, making education more modular and manageable.

However, challenges exist:

1. **Digital Divide:** Not everyone has access to stable internet and digital devices. Efforts must be made to bridge this gap.
2. **Quality Control:** Ensuring the quality of online education and preventing dilution of standards is crucial.
3. **Personal Interaction:** Physical classroom interactions have their benefits in terms of networking, hands-on experiences, and soft skill development. Hybrid models can be a solution.

In conclusion, while digital educational platforms show great promise in democratizing access to education, careful design, and thoughtful implementation are crucial. With strategic efforts, they can indeed make education more accessible and affordable to the less fortunate.

---

## 7. Conclusion

In the rapidly evolving landscape of education, the envisioned digital educational platform stands as a beacon of innovation and decentralization, aiming to revolutionize the traditional paradigms of learning. By placing the professor at the core of the educational experience, this system not only celebrates the individual talents and expertise of educators but also democratizes access to quality education on a global scale.

This platform's decentralized nature ensures that knowledge is not confined within the walls of institutions but flows freely, harnessed and directed by the professors themselves. They are empowered to curate their curriculum, ensuring that students receive content that is both up-to-date and aligned with the educator's passion and expertise.

The reputation index serves as a vital tool, ensuring transparency and quality assurance. It empowers students to make informed decisions, fostering a competitive environment that rewards excellence in teaching. Moreover, by allowing professors to operate remotely, the platform ensures that geographic boundaries no longer limit education. Students, regardless of their location, can access world-class expertise and diverse perspectives.

Furthermore, the platform's compatibility with various devices makes education truly omnipresent, ensuring that learning can occur anytime, anywhere, breaking down traditional barriers and creating a more inclusive environment.

By necessitating that students engage with multiple professors worldwide to earn their degree, the system promotes a holistic and diversified learning experience, enriching students with varied perspectives and expertise.

In amalgamation with the earlier discussed operating system, this digital platform has the potential to usher in a new era of education – one that is more equitable, accessible, and

focused on lifelong learning. Through its visionary approach, it underscores the belief that improving global education standards is not just the responsibility of institutions but also of individuals. By harnessing the power of technology and the internet, we are on the brink of creating an ecosystem where quality education is not a privilege but a right accessible to all.

---