TechEduPro: Project Report

CS51550-001, Fall 2024
PNW University
10/05/2024

Name	Email	Contributions
Pandya Shivam	**pandya23@pnw.edu**	**Database Design, JDBC Connection and Container Creation**
Mekala Ruthvik Reddy	**rmekala@pnw.edu**	**Application Code (Student Menu) and Documentation**
Nalluri Prashanth	**pnallur@pnw.edu**	**Application Code (Department and Registrar Menu)**

Table of Contents

Table of Figures

ABSTRACT

The University Registration System is a new Java application to alleviate tasks involved with university-level students, registrars, and department personnel registration processes. It uses Java to interact with a database through JDBC for course registration by students, their personal information, and their academic records while applying all sorts of academic policies and constraints. The integration of student registration, course registration, billing, and staff management components in the URS system is geared at improving efficient operations and satisfactory service to the users.

It has three categories of users, which include Registrar Staff, Department Staff, and Students with their various capabilities. Registrar Staff can load sections and grades from the files, update the section capacities, and view course schedules of classes and student transcripts. The Department Staff can approve the students to take the classes, manage the overflow for full sections, and generate class lists. It has an easy-to-use interface feature that gives students a chance to register for the courses of their choice, access academic records, and check the bills for fee payments.

Accordingly, the system checks the control of the academic policy: forbidding undergrads from taking graduate courses, managing credit hours, and avoiding course overlapping. Keeping these constraints maintained, the University Registration System builds up the integrity and develops the user experience in registration. Additionally, we utilized a Docker container for MySQL 8, enhancing the deployment process and ensuring consistent environments for database management. This project is a very important milestone toward updating the university registration system for better efficiency and adding user-friendliness to all end-users involved.

I. INTRODUCTION

In recent years, it has become apparent that there has been an evolution in the higher education which has in turn calls for more effective and user-friendly systems to manage student registration, course offerings, and records. Due to excessive time consumption on account of the traditional processes of the University registration, all parties – students, faculty as well as the administration were usually stressed out and felt exasperated. With growing student numbers and increasing acceptance of a wider range of academic programs, the need for a robust registration system has become vital.

The goal of the University Registration System is to solve these problems by way of offering a specialized product that would fit the ways of work of different user categories within a university. The system is organized into three distinct user roles: Registrar Staff, Department Staff, and Students, each bearing the autonomy to perform certain academic operations without contravening essential academic regulations. Additionally, we utilized a Docker container for MySQL8, which streamlines the deployment process and ensures a consistent environment for managing the database, further enhancing the system's reliability and performance.

The Registrar Staff handles and oversees all matters relating to the registration of students, course details as well as student records. They need a system that provides easy and fast modifications and data retrieval in order to enhance activities. Department Staff are the dealers of approving course enrollments and managing the allowances for student enrollments in the course, requiring a simple and powerful system for helping with the control and modification of students in the classes. And students should have access to facilities regarding course enrollment, class scheduling, and view their academic progress without facing hurdles and prepare for the betterment of the grades along with that students can also view their bills for particular terms.

In this regard, the University Registration System will not only ease the process of the registration but also facilitate communication and workflow among participants. Because the system is integrated with features that implement academic policies, such as enrollment restrictions by classification of students and limits of credit hours, the academic environment becomes well-organized and transparent. The different user groups are thus empowered and enabled through an educational experience for academic success with operational efficiency.

II. SYSTEM DESIGN AND IMPLEMENTATION

The work undertaken on the University Registration System started by the development of an Entity-Relationship (E.R.) diagram to clarify links between various entities such as students, courses, people, and so on. This diagram acted as a template for the later counterparts of the database and for the system itself. To enhance deployment, we utilized Docker containers to run a MySQL database, ensuring consistency across different environments. After this, we developed the application in Java with the help of JDBC which included realization of the E.R. model in the functional database schema. The coding stage was focused on creating user interfaces and business logic, designed specifically for the Registrar Staff, Staff of the Department, and Students to ensure proper integration of functionality for the smooth registration process.

1. ER Diagram

First, we created an ER Diagram for the whole system to get a better understanding of the tables in the system and to write the query easily. Since there are primary keys and foreign key we need to know the relationship of each table properly.

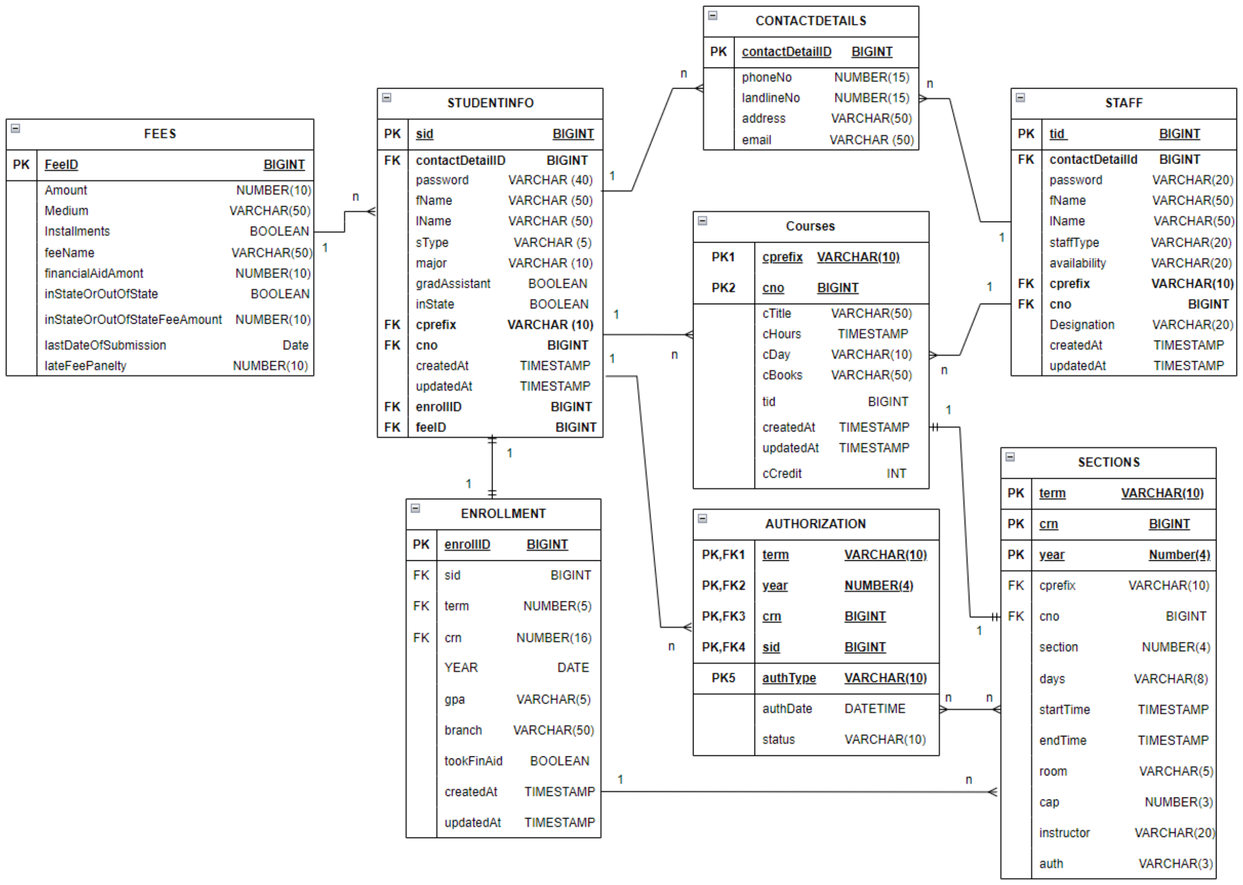
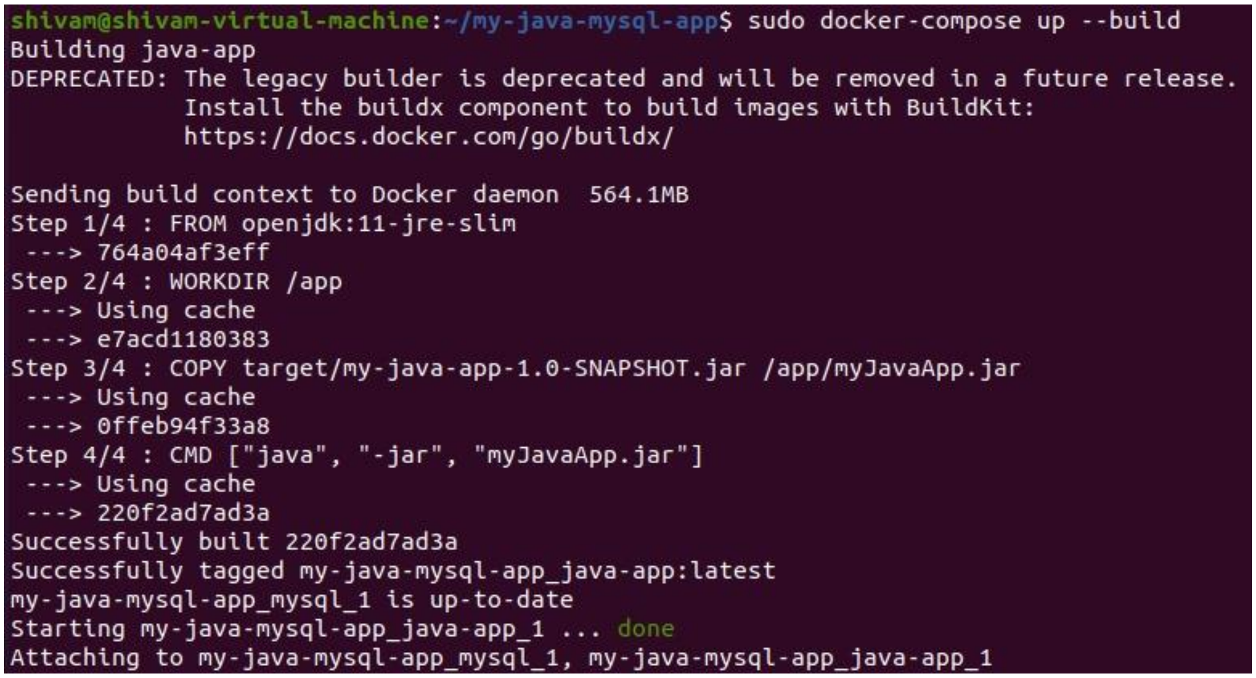
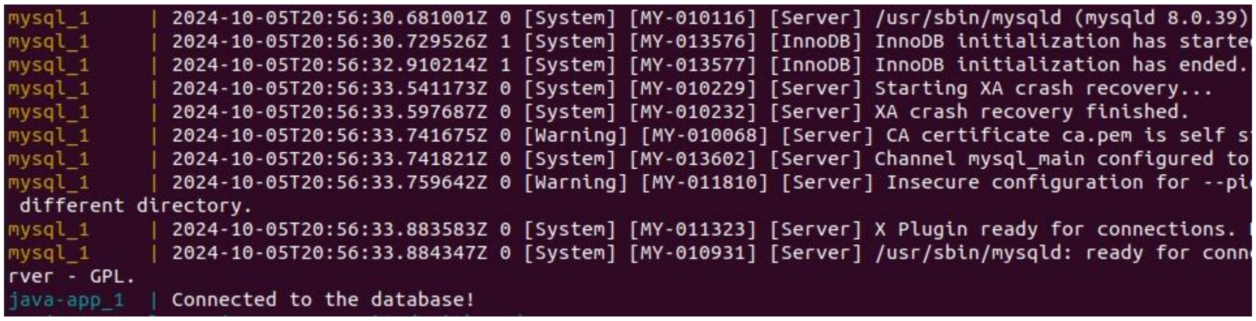


Figure 1 – ER Diagram

2. Connecting Java App to Dockerised MySQL

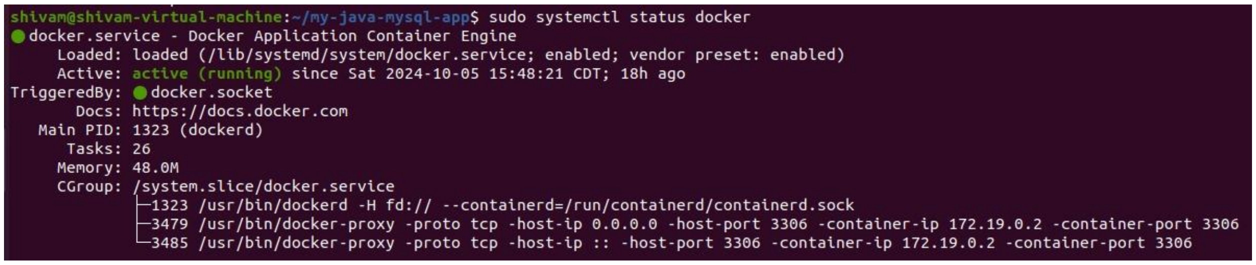


```
shivam@shivam-virtual-machine:~/my-java-mysql-app$ sudo docker-compose up --build
Building java-app
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  564.1MB
Step 1/4 : FROM openjdk:11-jre-slim
 ---> 764a04af3eff
Step 2/4 : WORKDIR /app
 ---> Using cache
 ---> e7acd1180383
Step 3/4 : COPY target/my-java-app-1.0-SNAPSHOT.jar /app/myJavaApp.jar
 ---> Using cache
 ---> 0ffeb94f33a8
Step 4/4 : CMD ["java", "-jar", "myJavaApp.jar"]
 ---> Using cache
 ---> 220f2ad7ad3a
Successfully built 220f2ad7ad3a
Successfully tagged my-java-mysql-app_java-app:latest
my-java-mysql-app_mysql_1 is up-to-date
Starting my-java-mysql-app_java-app_1 ... done
Attaching to my-java-mysql-app_mysql_1, my-java-mysql-app_java-app_1
```



```
mysql_1     | 2024-10-05T20:56:30.681001Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.39)
mysql_1     | 2024-10-05T20:56:30.729526Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has starte
mysql_1     | 2024-10-05T20:56:32.910214Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_1     | 2024-10-05T20:56:33.541173Z 0 [System] [MY-010229] [Server] Starting XA crash recovery...
mysql_1     | 2024-10-05T20:56:33.597687Z 0 [System] [MY-010232] [Server] XA crash recovery finished.
mysql_1     | 2024-10-05T20:56:33.741675Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self s
mysql_1     | 2024-10-05T20:56:33.741821Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to
mysql_1     | 2024-10-05T20:56:33.759642Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pi
 different directory.
mysql_1     | 2024-10-05T20:56:33.883583Z 0 [System] [MY-011323] [Server] X Plugin ready for connections.
mysql_1     | 2024-10-05T20:56:33.884347Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for conn
rver - GPL.
java-app_1  | Connected to the database!
```

Figure 2– Establishing the connection

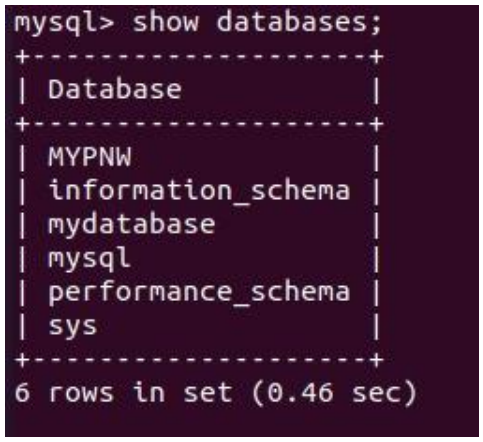


```
shivam@shivam-virtual-machine:~/my-java-mysql-app$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-10-05 15:48:21 CDT; 18h ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1323 (dockerd)
      Tasks: 26
     Memory: 48.0M
     CGroup: /system.slice/docker.service
             ├─1323 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
             ├─3479 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3306 -container-ip 172.19.0.2 -container-port 3306
             └─3485 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 3306 -container-ip 172.19.0.2 -container-port 3306
```

Figure 3– Docker Container status

3. Database Schema

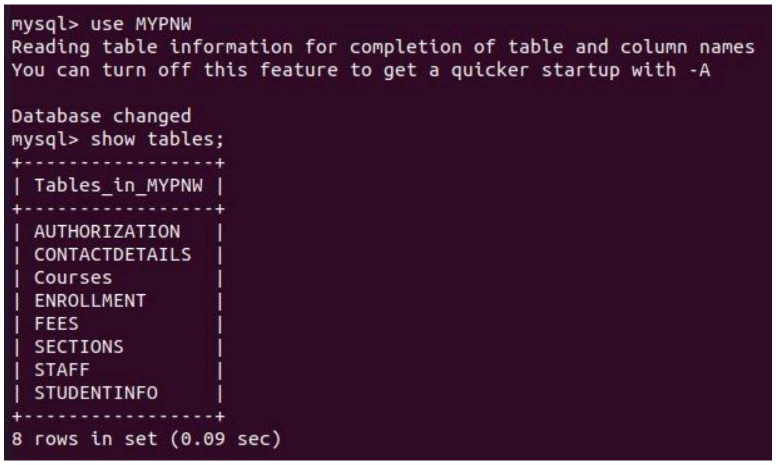
a. Then we started to create a database named MYPNW in a dockerized MySQL 8 in Linux.



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| MYPNW              |
| information_schema |
| mydatabase         |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.46 sec)
```

Figure 4- MYPNW Database

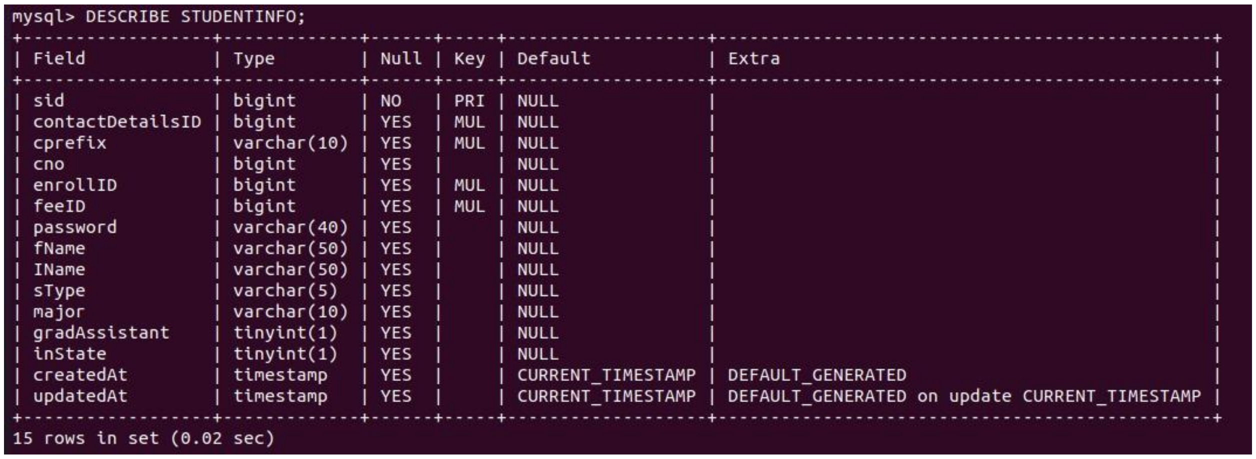
b. After that we created all the tables in the database. The complete system would be based on a database schema, containing all the following tables, connected properly: STUDENTINFO, COURSES, ENROLLMENT, CONTACT DETAILS, FEES, STAFF, SECTION, and AUTHORIZATION. These tables have been designed to accommodate some key data regarding students, courses.



```
mysql> use MYPNW
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------+
| Tables_in_MYPNW |
+----------------+
| AUTHORIZATION  |
| CONTACTDETAILS |
| Courses        |
| ENROLLMENT     |
| FEES           |
| SECTIONS       |
| STAFF          |
| STUDENTINFO    |
+----------------+
8 rows in set (0.09 sec)
```

Figure 5- Tables in the Database

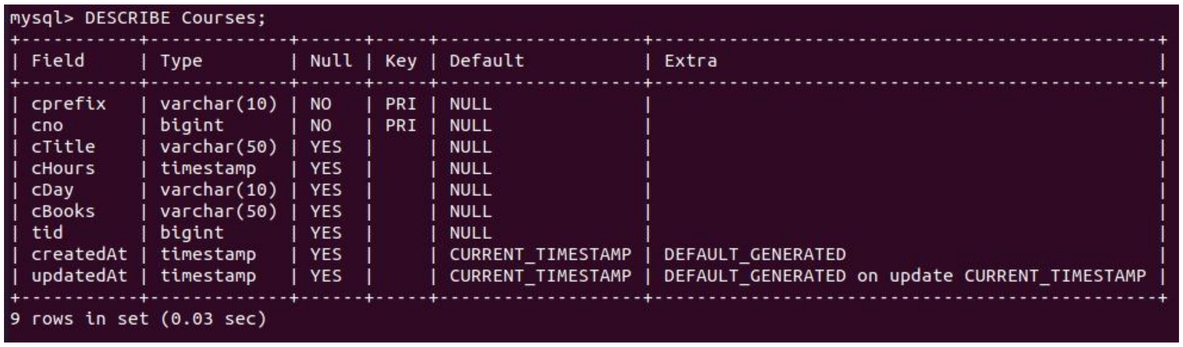
c. **STUDENTINFO:** This table contains the essential information about a student's record at the institution, such as SID, personal information, passwords, major and financial aid awarded.



```
mysql> DESCRIBE STUDENTINFO;
+------------------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field            | Type        | Null | Key | Default           | Extra                                         |
+------------------+-------------+------+-----+-------------------+-----------------------------------------------+
| sid              | bigint      | NO   | PRI | NULL              |                                               |
| contactDetailsID | bigint      | YES  | MUL | NULL              |                                               |
| cprefix          | varchar(10) | YES  | MUL | NULL              |                                               |
| cno              | bigint      | YES  |     | NULL              |                                               |
| enrollID         | bigint      | YES  | MUL | NULL              |                                               |
| feeID            | bigint      | YES  | MUL | NULL              |                                               |
| password         | varchar(40) | YES  |     | NULL              |                                               |
| fName            | varchar(50) | YES  |     | NULL              |                                               |
| IName            | varchar(50) | YES  |     | NULL              |                                               |
| sType            | varchar(5)  | YES  |     | NULL              |                                               |
| major            | varchar(10) | YES  |     | NULL              |                                               |
| gradAssistant    | tinyint(1)  | YES  |     | NULL              |                                               |
| inState          | tinyint(1)  | YES  |     | NULL              |                                               |
| createdAt        | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED                             |
| updatedAt        | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+------------------+-------------+------+-----+-------------------+-----------------------------------------------+
15 rows in set (0.02 sec)
```

Figure 6- STUDENTINFO Table

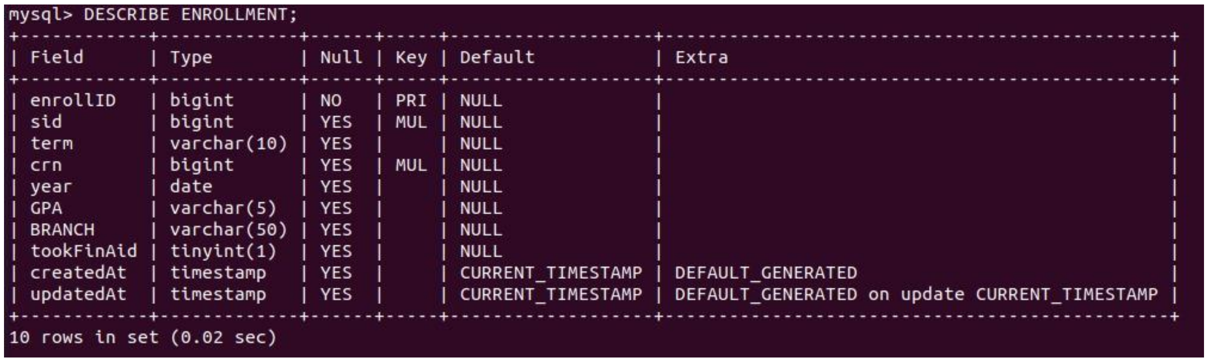
d. **COURSES:** This table includes Course Title, including course prefix, course number - CNO, credit hours, and availability.



```
mysql> DESCRIBE Courses;
+-----------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field     | Type        | Null | Key | Default           | Extra                                         |
+-----------+-------------+------+-----+-------------------+-----------------------------------------------+
| cprefix   | varchar(10) | NO   | PRI | NULL              |                                               |
| cno       | bigint      | NO   | PRI | NULL              |                                               |
| cTitle    | varchar(50) | YES  |     | NULL              |                                               |
| cHours    | timestamp   | YES  |     | NULL              |                                               |
| cDay      | varchar(10) | YES  |     | NULL              |                                               |
| cBooks    | varchar(50) | YES  |     | NULL              |                                               |
| tid       | bigint      | YES  |     | NULL              |                                               |
| createdAt | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED                             |
| updatedAt | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----------+-------------+------+-----+-------------------+-----------------------------------------------+
9 rows in set (0.03 sec)
```

Figure 7- Courses Table

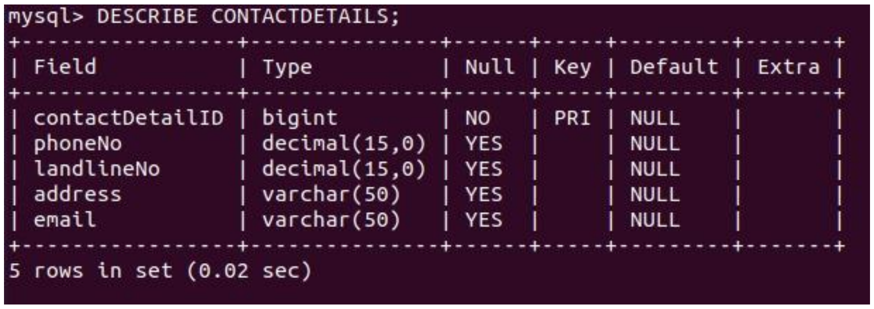
e. **ENROLLMENT:** This table consists of student enrollments in particular courses and the relevant data of their semester and grades.



```
mysql> DESCRIBE ENROLLMENT;
+------------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field      | Type        | Null | Key | Default           | Extra                                         |
+------------+-------------+------+-----+-------------------+-----------------------------------------------+
| enrollID   | bigint      | NO   | PRI | NULL              |                                               |
| sid        | bigint      | YES  | MUL | NULL              |                                               |
| term       | varchar(10) | YES  |     | NULL              |                                               |
| crn        | bigint      | YES  | MUL | NULL              |                                               |
| year       | date        | YES  |     | NULL              |                                               |
| GPA        | varchar(5)  | YES  |     | NULL              |                                               |
| BRANCH     | varchar(50) | YES  |     | NULL              |                                               |
| tookFinAid | tinyint(1)  | YES  |     | NULL              |                                               |
| createdAt  | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED                             |
| updatedAt  | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+------------+-------------+------+-----+-------------------+-----------------------------------------------+
10 rows in set (0.02 sec)
```

Figure 8- ENROLLMENT Table

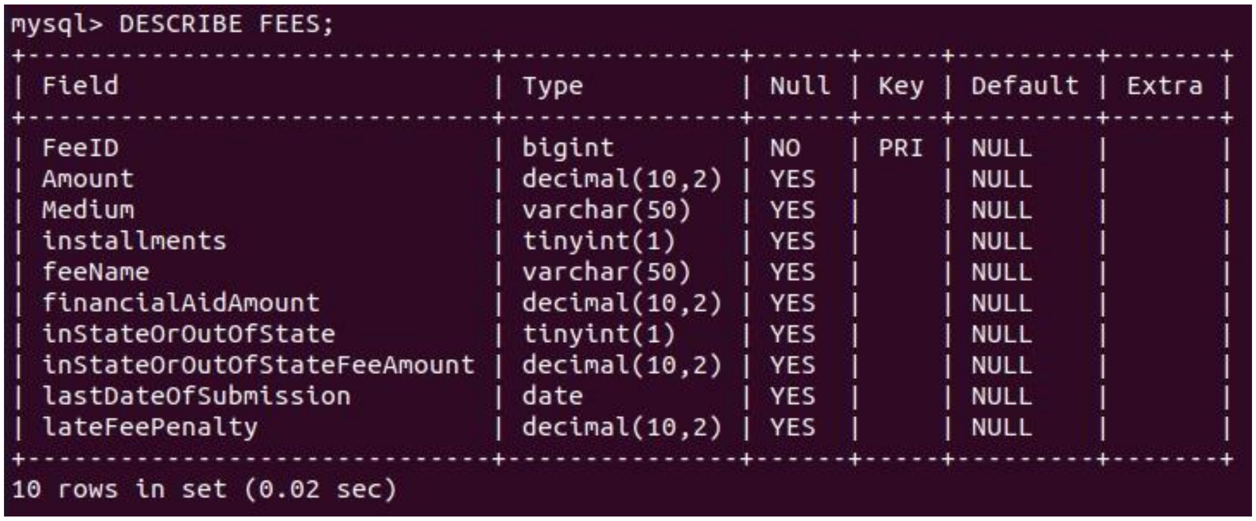
f. **CONTACTDETAILS:** This table has the contact information of both students and staff along with their house address and email address



```
mysql> DESCRIBE CONTACTDETAILS;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| contactDetailID | bigint        | NO   | PRI | NULL    |       |
| phoneNo         | decimal(15,0) | YES  |     | NULL    |       |
| landlineNo      | decimal(15,0) | YES  |     | NULL    |       |
| address         | varchar(50)   | YES  |     | NULL    |       |
| email           | varchar(50)   | YES  |     | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```

Figure 9- CONTACTDETAILS Table

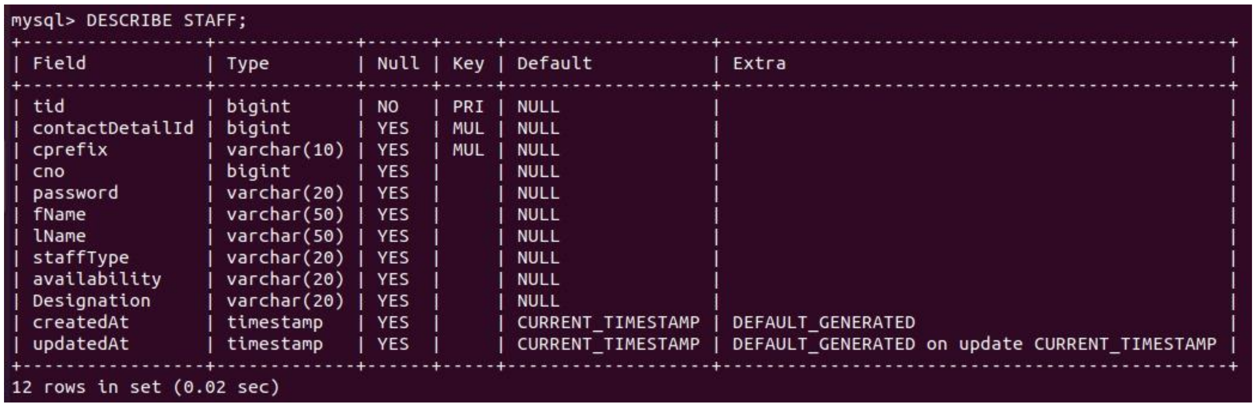
g. **FEES:** This table manages the financial aids of the students, tuition fees, installment payments,etc.



```
mysql> DESCRIBE FEES;
+-----------------------------+---------------+------+-----+---------+-------+
| Field                       | Type          | Null | Key | Default | Extra |
+-----------------------------+---------------+------+-----+---------+-------+
| FeeID                       | bigint        | NO   | PRI | NULL    |       |
| Amount                      | decimal(10,2) | YES  |     | NULL    |       |
| Medium                      | varchar(50)   | YES  |     | NULL    |       |
| installments                | tinyint(1)    | YES  |     | NULL    |       |
| feeName                     | varchar(50)   | YES  |     | NULL    |       |
| financialAidAmount          | decimal(10,2) | YES  |     | NULL    |       |
| inStateOrOutOfState         | tinyint(1)    | YES  |     | NULL    |       |
| inStateOrOutOfStateFeeAmount | decimal(10,2) | YES  |     | NULL    |       |
| lastDateOfSubmission        | date          | YES  |     | NULL    |       |
| lateFeePenalty              | decimal(10,2) | YES  |     | NULL    |       |
+-----------------------------+---------------+------+-----+---------+-------+
10 rows in set (0.02 sec)
```

Figure 10- FEES Table

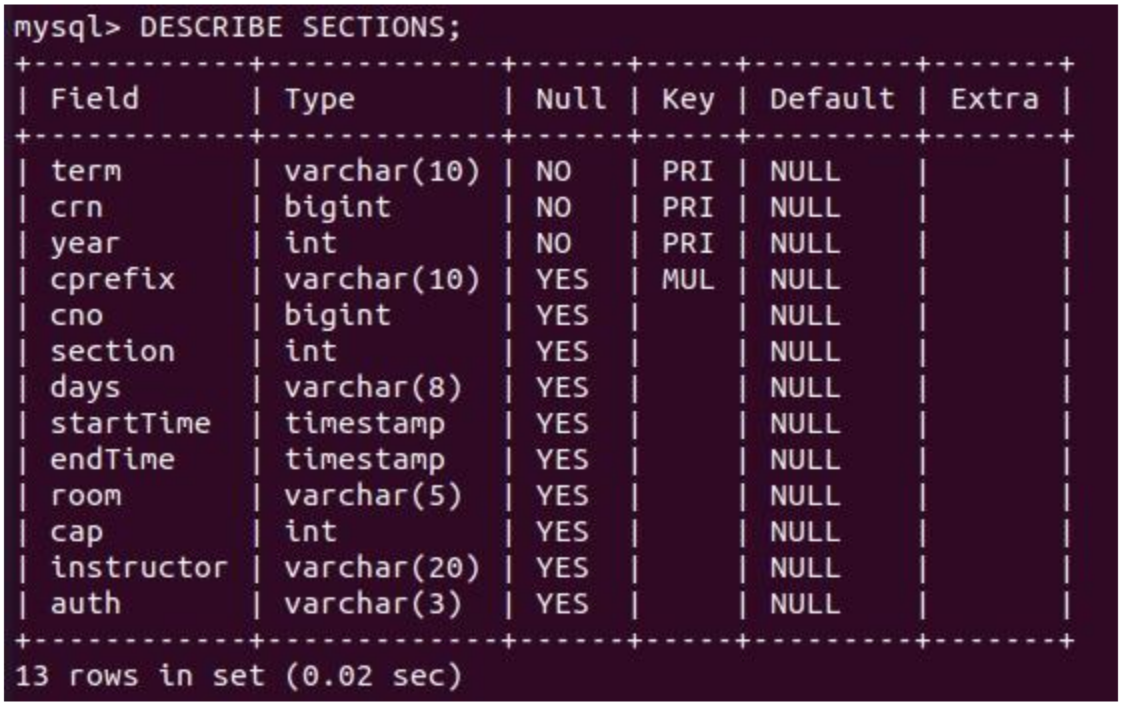
h. **STAFF:** This table manages the information of the staff members along with their identification number, passwords and their designation (Department, Registrar).



```
mysql> DESCRIBE STAFF;
+-----------------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field           | Type        | Null | Key | Default           | Extra                                         |
+-----------------+-------------+------+-----+-------------------+-----------------------------------------------+
| tid             | bigint      | NO   | PRI | NULL              |                                               |
| contactDetailId | bigint      | YES  | MUL | NULL              |                                               |
| cprefix         | varchar(10) | YES  | MUL | NULL              |                                               |
| cno             | bigint      | YES  |     | NULL              |                                               |
| password        | varchar(20) | YES  |     | NULL              |                                               |
| fName           | varchar(50) | YES  |     | NULL              |                                               |
| lName           | varchar(50) | YES  |     | NULL              |                                               |
| staffType       | varchar(20) | YES  |     | NULL              |                                               |
| availability    | varchar(20) | YES  |     | NULL              |                                               |
| Designation     | varchar(20) | YES  |     | NULL              |                                               |
| createdAt       | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED                             |
| updatedAt       | timestamp   | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
+-----------------+-------------+------+-----+-------------------+-----------------------------------------------+
12 rows in set (0.02 sec)
```

Figure 11- STAFF Table

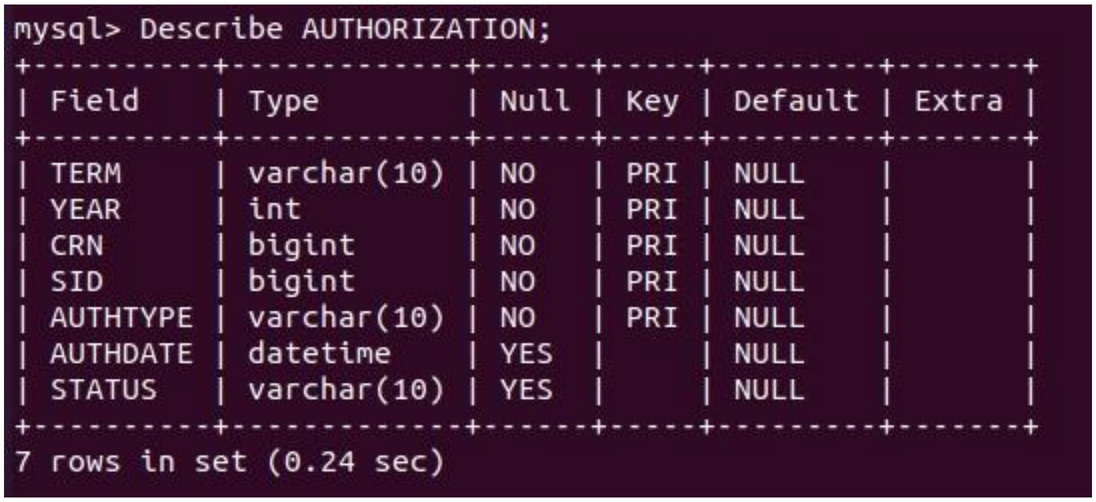
i. **SECTIONS:** Provides the section information for each class about the capacity, current enrollment, meeting times, and instructor.



```
mysql> DESCRIBE SECTIONS;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| term       | varchar(10) | NO   | PRI | NULL    |       |
| crn        | bigint      | NO   | PRI | NULL    |       |
| year       | int         | NO   | PRI | NULL    |       |
| cprefix    | varchar(10) | YES  | MUL | NULL    |       |
| cno        | bigint      | YES  |     | NULL    |       |
| section    | int         | YES  |     | NULL    |       |
| days       | varchar(8)  | YES  |     | NULL    |       |
| startTime  | timestamp   | YES  |     | NULL    |       |
| endTime    | timestamp   | YES  |     | NULL    |       |
| room       | varchar(5)  | YES  |     | NULL    |       |
| cap        | int         | YES  |     | NULL    |       |
| instructor | varchar(20) | YES  |     | NULL    |       |
| auth       | varchar(3)  | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
13 rows in set (0.02 sec)
```

Figure 12- SECTIONS Table

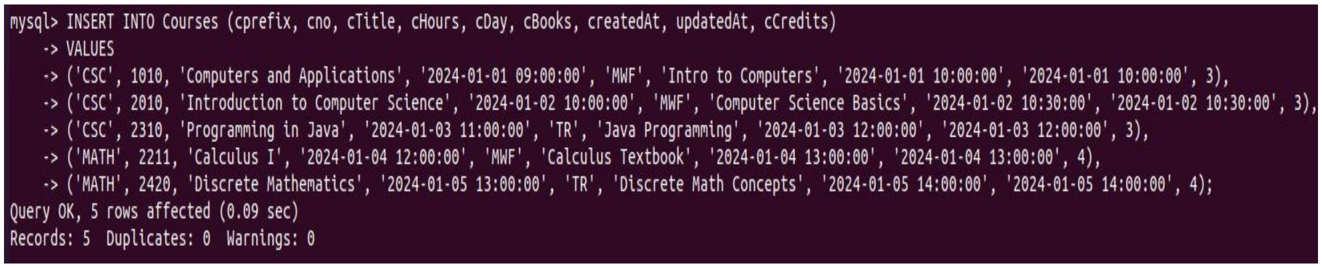
j. **AUTHORIZATION:** This table shows the authorization required for courses linking it to students.



```
mysql> Describe AUTHORIZATION;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| TERM     | varchar(10) | NO   | PRI | NULL    |       |
| YEAR     | int         | NO   | PRI | NULL    |       |
| CRN      | bigint      | NO   | PRI | NULL    |       |
| SID      | bigint      | NO   | PRI | NULL    |       |
| AUTHTYPE | varchar(10) | NO   | PRI | NULL    |       |
| AUTHDATE | datetime    | YES  |     | NULL    |       |
| STATUS   | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
7 rows in set (0.24 sec)
```

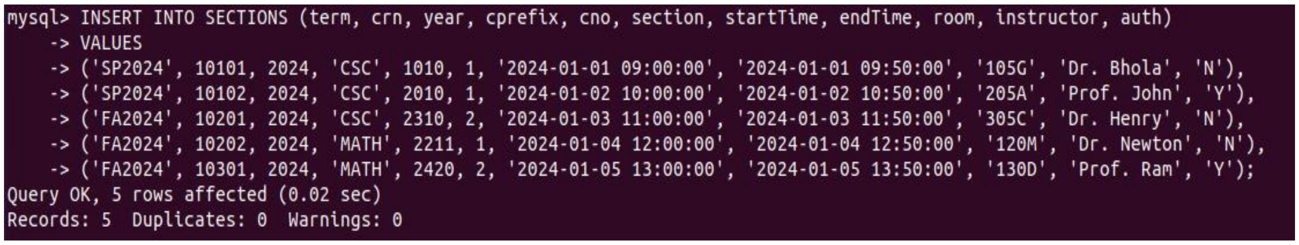
Figure 13- AUTHORIZATION Table

k. Inserting Values into the tables.



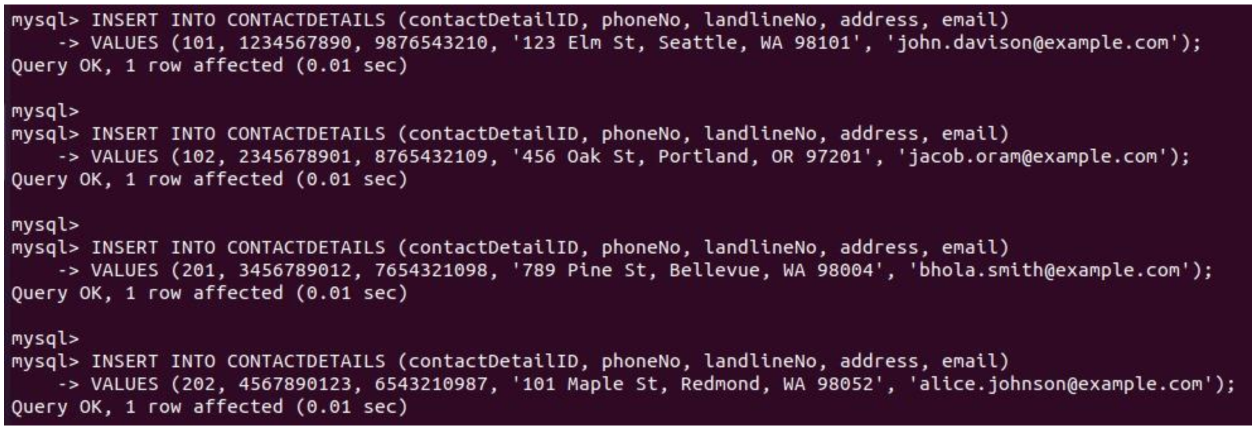
```
mysql> INSERT INTO Courses (cprefix, cno, cTitle, cHours, cDay, cBooks, createdAt, updatedAt, cCredits)
    -> VALUES
    -> ('CSC', 1010, 'Computers and Applications', '2024-01-01 09:00:00', 'MWF', 'Intro to Computers', '2024-01-01 10:00:00', '2024-01-01 10:00:00', 3),
    -> ('CSC', 2010, 'Introduction to Computer Science', '2024-01-02 10:00:00', 'MWF', 'Computer Science Basics', '2024-01-02 10:30:00', '2024-01-02 10:30:00', 3),
    -> ('CSC', 2310, 'Programming in Java', '2024-01-03 11:00:00', 'TR', 'Java Programming', '2024-01-03 12:00:00', '2024-01-03 12:00:00', 3),
    -> ('MATH', 2211, 'Calculus I', '2024-01-04 12:00:00', 'MWF', 'Calculus Textbook', '2024-01-04 13:00:00', '2024-01-04 13:00:00', 4),
    -> ('MATH', 2420, 'Discrete Mathematics', '2024-01-05 13:00:00', 'TR', 'Discrete Math Concepts', '2024-01-05 14:00:00', '2024-01-05 14:00:00', 4);
Query OK, 5 rows affected (0.09 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Figure 14- Inserting Values into Courses



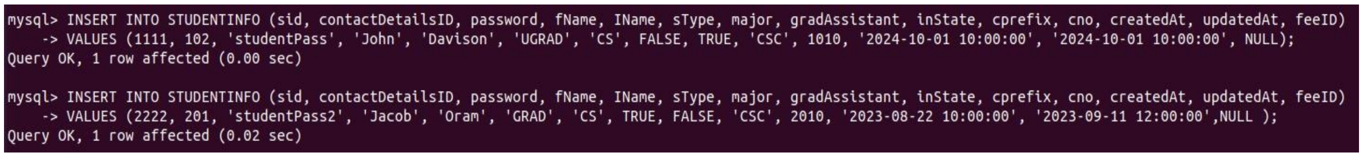
```
mysql> INSERT INTO SECTIONS (term, crn, year, cprefix, cno, section, startTime, endTime, room, instructor, auth)
    -> VALUES
    -> ('SP2024', 10101, 2024, 'CSC', 1010, 1, '2024-01-01 09:00:00', '2024-01-01 09:50:00', '105G', 'Dr. Bhola', 'N'),
    -> ('SP2024', 10102, 2024, 'CSC', 2010, 1, '2024-01-02 10:00:00', '2024-01-02 10:50:00', '205A', 'Prof. John', 'Y'),
    -> ('FA2024', 10201, 2024, 'CSC', 2310, 2, '2024-01-03 11:00:00', '2024-01-03 11:50:00', '305C', 'Dr. Henry', 'N'),
    -> ('FA2024', 10202, 2024, 'MATH', 2211, 1, '2024-01-04 12:00:00', '2024-01-04 12:50:00', '120M', 'Dr. Newton', 'N'),
    -> ('FA2024', 10301, 2024, 'MATH', 2420, 2, '2024-01-05 13:00:00', '2024-01-05 13:50:00', '130D', 'Prof. Ram', 'Y');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Figure 15- Inserting Values into SECTIONS



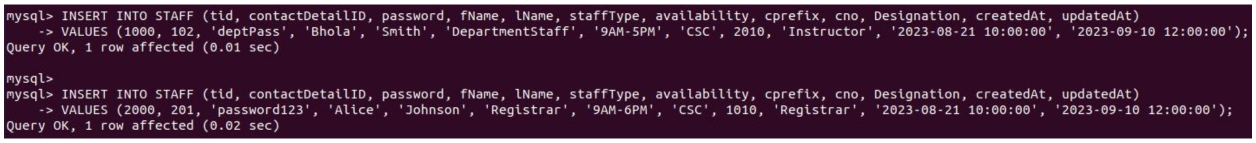
```
mysql> INSERT INTO CONTACTDETAILS (contactDetailID, phoneNo, landlineNo, address, email)
    -> VALUES (101, 1234567890, 9876543210, '123 Elm St, Seattle, WA 98101', 'john.davison@example.com');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO CONTACTDETAILS (contactDetailID, phoneNo, landlineNo, address, email)
    -> VALUES (102, 2345678901, 8765432109, '456 Oak St, Portland, OR 97201', 'jacob.oram@example.com');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO CONTACTDETAILS (contactDetailID, phoneNo, landlineNo, address, email)
    -> VALUES (201, 3456789012, 7654321098, '789 Pine St, Bellevue, WA 98004', 'bhola.smith@example.com');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO CONTACTDETAILS (contactDetailID, phoneNo, landlineNo, address, email)
    -> VALUES (202, 4567890123, 6543210987, '101 Maple St, Redmond, WA 98052', 'alice.johnson@example.com');
Query OK, 1 row affected (0.01 sec)
```

Figure 16- Inserting Values into CONTACTDETAILS



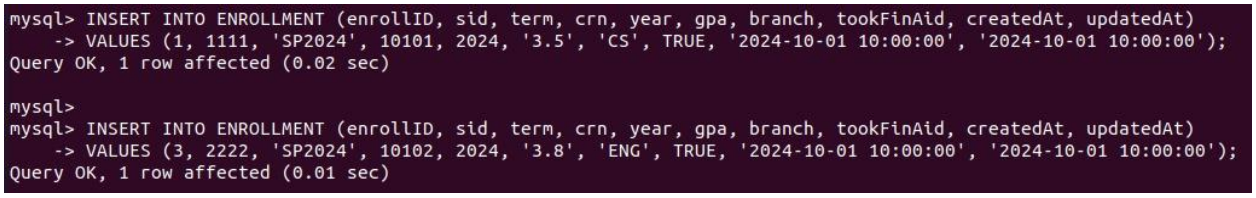
```
mysql> INSERT INTO STUDENTINFO (sid, contactDetailsID, password, fName, IName, sType, major, gradAssistant, inState, cprefix, cno, createdAt, updatedAt, feeID)
    -> VALUES (1111, 102, 'studentPass', 'John', 'Davison', 'UGRAD', 'CS', FALSE, TRUE, 'CSC', 1010, '2024-10-01 10:00:00', '2024-10-01 10:00:00', NULL);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENTINFO (sid, contactDetailsID, password, fName, IName, sType, major, gradAssistant, inState, cprefix, cno, createdAt, updatedAt, feeID)
    -> VALUES (2222, 201, 'studentPass2', 'Jacob', 'Oram', 'GRAD', 'CS', TRUE, FALSE, 'CSC', 2010, '2023-08-22 10:00:00', '2023-09-11 12:00:00',NULL );
Query OK, 1 row affected (0.02 sec)
```

Figure 17- Inserting Values into STUDENTINFO



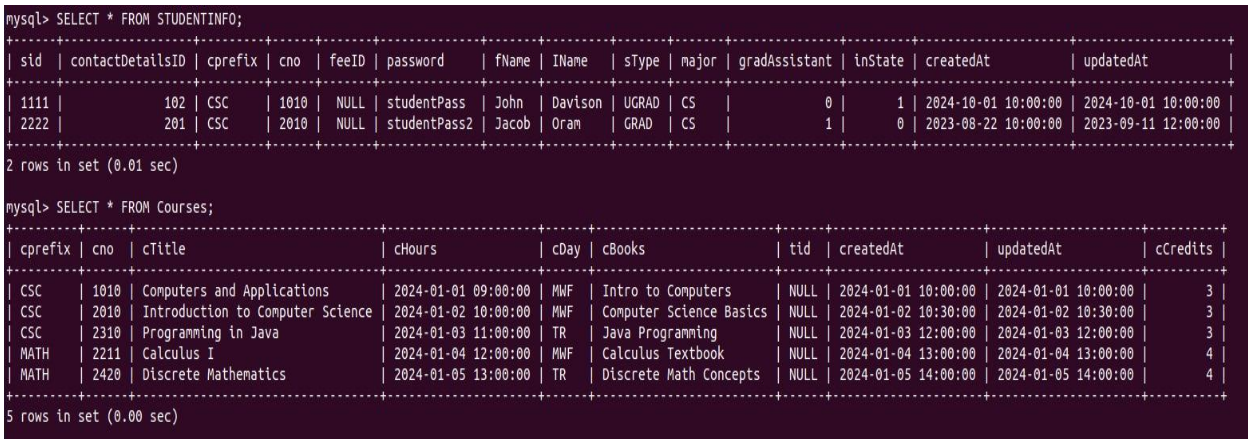
```
mysql> INSERT INTO STAFF (tid, contactDetailID, password, fName, lName, staffType, availability, cprefix, cno, Designation, createdAt, updatedAt)
    -> VALUES (1000, 102, 'deptPass', 'Bhola', 'Smith', 'DepartmentStaff', '9AM-5PM', 'CSC', 2010, 'Instructor', '2023-08-21 10:00:00', '2023-09-10 12:00:00');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO STAFF (tid, contactDetailID, password, fName, lName, staffType, availability, cprefix, cno, Designation, createdAt, updatedAt)
    -> VALUES (2000, 201, 'password123', 'Alice', 'Johnson', 'Registrar', '9AM-6PM', 'CSC', 1010, 'Registrar', '2023-08-21 10:00:00', '2023-09-10 12:00:00');
Query OK, 1 row affected (0.02 sec)
```

Figure 18- Inserting Values into STAFF



```
mysql> INSERT INTO ENROLLMENT (enrollID, sid, term, crn, year, gpa, branch, tookFinAid, createdAt, updatedAt)
    -> VALUES (1, 1111, 'SP2024', 10101, 2024, '3.5', 'CS', TRUE, '2024-10-01 10:00:00', '2024-10-01 10:00:00');
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> INSERT INTO ENROLLMENT (enrollID, sid, term, crn, year, gpa, branch, tookFinAid, createdAt, updatedAt)
    -> VALUES (3, 2222, 'SP2024', 10102, 2024, '3.8', 'ENG', TRUE, '2024-10-01 10:00:00', '2024-10-01 10:00:00');
Query OK, 1 row affected (0.01 sec)
```

Figure 19- Inserting Values into ENROLLMENT



```
mysql> SELECT * FROM STUDENTINFO;
+------+------------------+---------+------+-------+--------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| sid  | contactDetailsID | cprefix | cno  | feeID | password     | fName | lName   | sType | major | gradAssistant | inState | createdAt           | updatedAt           |
+------+------------------+---------+------+-------+--------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| 1111 |              102 | CSC     | 1010 |  NULL | studentPass  | John  | Davison | UGRAD | CS    |             0 |       1 | 2024-10-01 10:00:00 | 2024-10-01 10:00:00 |
| 2222 |              201 | CSC     | 2010 |  NULL | studentPass2 | Jacob | Oram    | GRAD  | CS    |             1 |       0 | 2023-08-22 10:00:00 | 2023-09-11 12:00:00 |
+------+------------------+---------+------+-------+--------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
2 rows in set (0.01 sec)

mysql> SELECT * FROM Courses;
+---------+------+----------------------------------+---------------------+------+-------------------------+------+---------------------+---------------------+----------+
| cprefix | cno  | cTitle                           | cHours              | cDay | cBooks                  | tid  | createdAt           | updatedAt           | cCredits |
+---------+------+----------------------------------+---------------------+------+-------------------------+------+---------------------+---------------------+----------+
| CSC     | 1010 | Computers and Applications       | 2024-01-01 09:00:00 | MWF  | Intro to Computers      | NULL | 2024-01-01 10:00:00 | 2024-01-01 10:00:00 |        3 |
| CSC     | 2010 | Introduction to Computer Science | 2024-01-02 10:00:00 | MWF  | Computer Science Basics | NULL | 2024-01-02 10:30:00 | 2024-01-02 10:30:00 |        3 |
| CSC     | 2310 | Programming in Java              | 2024-01-03 11:00:00 | TR   | Java Programming        | NULL | 2024-01-03 12:00:00 | 2024-01-03 12:00:00 |        3 |
| MATH    | 2211 | Calculus I                       | 2024-01-04 12:00:00 | MWF  | Calculus Textbook       | NULL | 2024-01-04 13:00:00 | 2024-01-04 13:00:00 |        4 |
| MATH    | 2420 | Discrete Mathematics             | 2024-01-05 13:00:00 | TR   | Discrete Math Concepts  | NULL | 2024-01-05 14:00:00 | 2024-01-05 14:00:00 |        4 |
+---------+------+----------------------------------+---------------------+------+-------------------------+------+---------------------+---------------------+----------+
5 rows in set (0.00 sec)
```

Figure 20- Tables After inserting Values

4. Relationships

Following is the relation of these tables with one another:

STUDENINFO to ENROLLMENT: One to many, since one student gets enrolled with multiple course codes.

ENROLLMENT to COURSES: A many-to-one relationship, registrations are available in many courses and all the registrations belong to one course.

COURSES to SECTIONS: One-to-many, since a course can be offered into different sections in different times of the day or week.

STUDENTINFO TO CONTACTDETAILS: one to one, wherein one student will be having only one contact detail.

STAFF to CONTACTDETAILS: One contact detail comes from staff; this is one to one in relationship.

STUDENTINFO to FEES: One to many; each student can have more than one record for fees.

ENROLLMENT to AUTHORIZATION: Many-to-one, since some enrollment records would be related to authorization needs of courses.

SECTIONS to AUTHORIZATION: One to many, meaning each section is potentially authorized for more than one student.

These tables will use foreign key constraints that will give them referential integrity, rightly associating each student with his enrollment records and their respective contact and course sections. This can then make possible the stalwart structure of the university's registration system and the effectiveness in managing and retrieving information.

5. User Authentication

The system utilizes a strong method of authentication to facilitate secure access through both staff and students. When logging in for the first time, the username and a password are the first details that are required from users. These usernames and passwords are authenticated either from the STUDENTINFO or STAFF tables according to the user type. Only students are verified against the STUDENTINFO table that stores the passwords, personal information and student ID SIDs. However, through the STAFF table that consists of passwords and staff designation, type like registrar or department staff, the system also authenticates users who oversee staff members.

Once authentication is complete, a user gets access to functions corresponding to the user's role. For students, all of which can perform se1 and fee details, the students can register courses, have schedule view, and transcripts for review. Privileges that relate to higher roles are for instance managing student and staff authorizations, creating class lists and adding assistantships. A registrar staff can load subsections, change grades as well as the load balance of courses.

Due to security reasons, the system is implemented using a role-based access control system. Staff members cannot execute actions based on a student level and students cannot do staff level actions too which helps in securing and protecting the database. With regards to enhancing security, all sensitive data such as passwords are made inconspicuous and secure.

III. USER INTERFACE

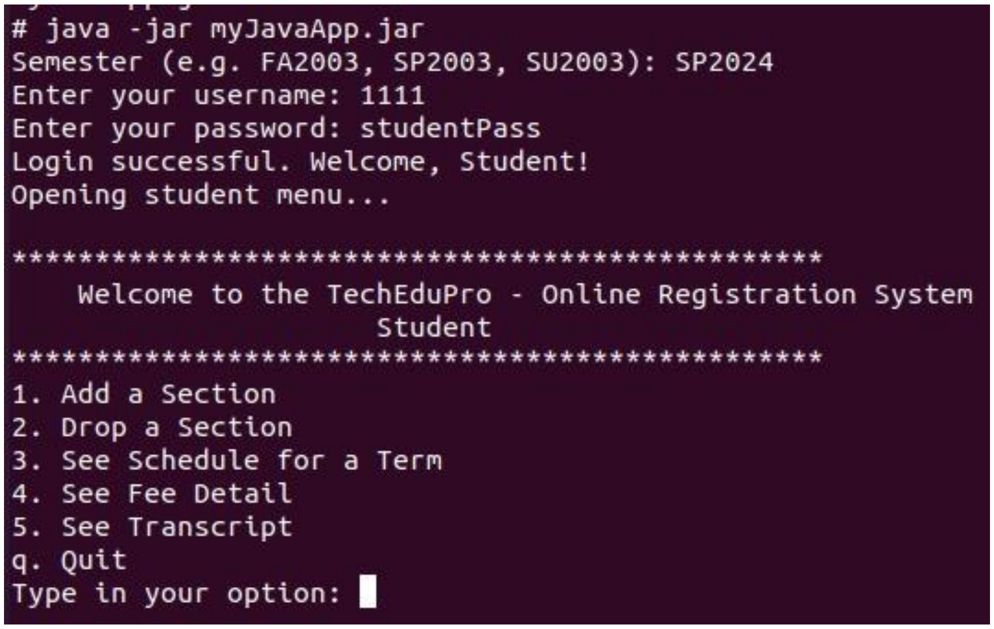
The System has different user interfaces for students, department staff, and registrar staff because the roles are also different. Students can manage course registration and academic details; department staff can handle student authorizations and assistantships. Course capacities are managed by the registrar staff, along with loading the data.

GitHub Link: https://github.com/shivampandya67/Student-Management-System

Demo Link: https://www.youtube.com/watch?v=Gn7lZ2NRSQU

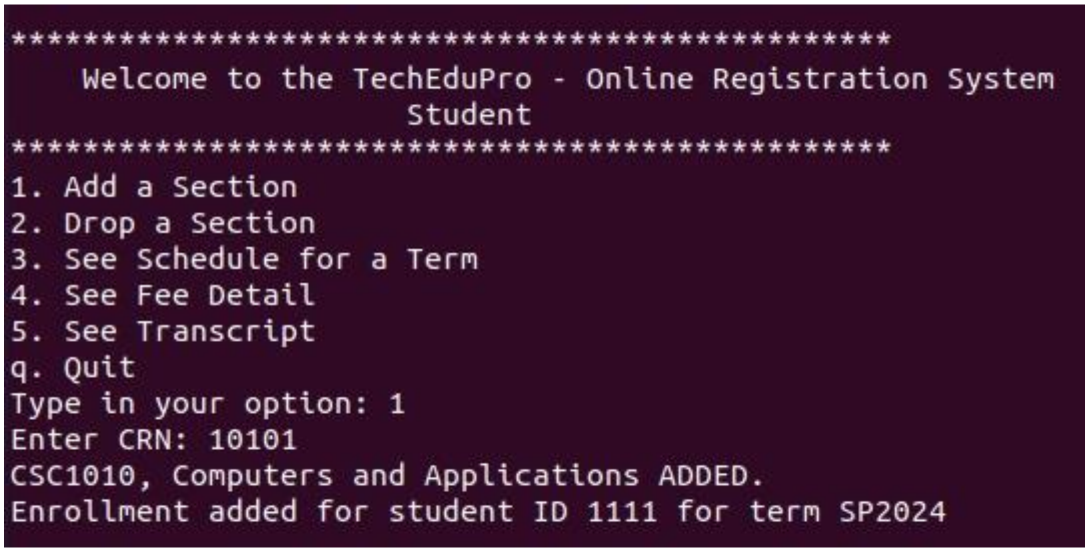
1. Student Menu

The student interface has a number of components that assist students manage their academic experience. Once students log into the system they can enroll in or drop sections, see their academic schedules for a particular term, examine fee details and avail official transcripts. The system enforces constraints like course capacity, time conflicts, credit hour limitations in order to ensure that students can only register for valid courses.



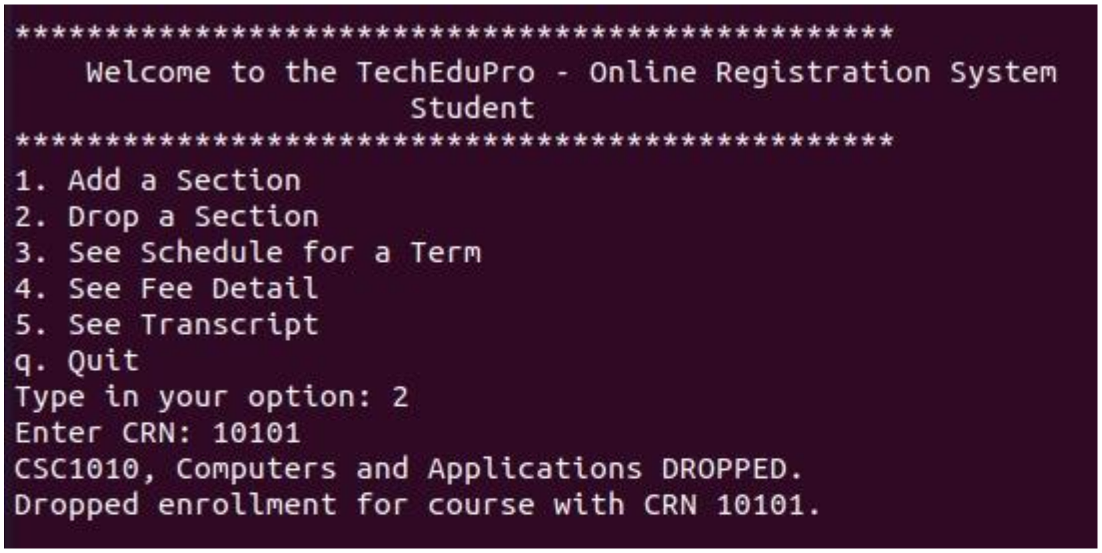
```
# java -jar myJavaApp.jar
Semester (e.g. FA2003, SP2003, SU2003): SP2024
Enter your username: 1111
Enter your password: studentPass
Login successful. Welcome, Student!
Opening student menu...

*************************************************
    Welcome to the TechEduPro - Online Registration System
                        Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option:
```

Figure 21- Student Menu



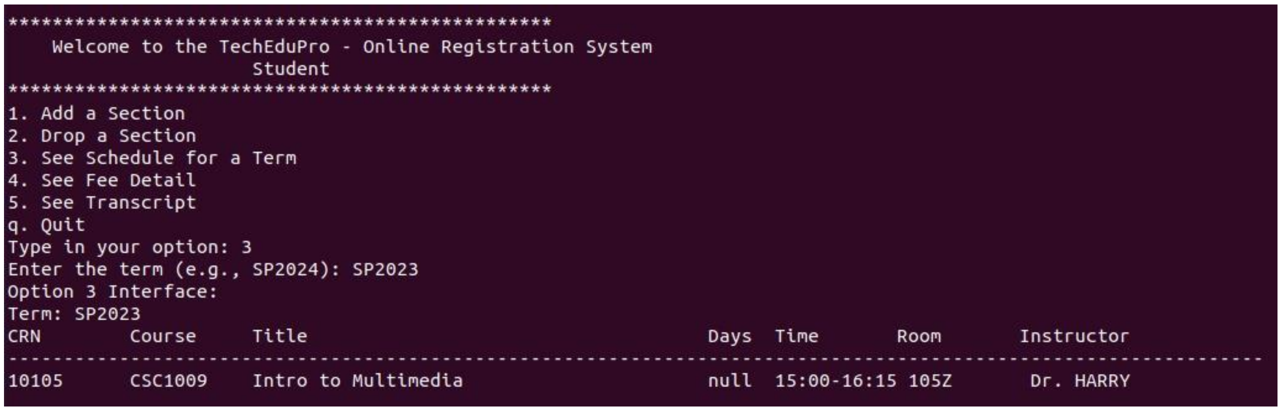
```
*************************************************
    Welcome to the TechEduPro - Online Registration System
                          Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option: 1
Enter CRN: 10101
CSC1010, Computers and Applications ADDED.
Enrollment added for student ID 1111 for term SP2024
```

Figure 22- Add section Interface



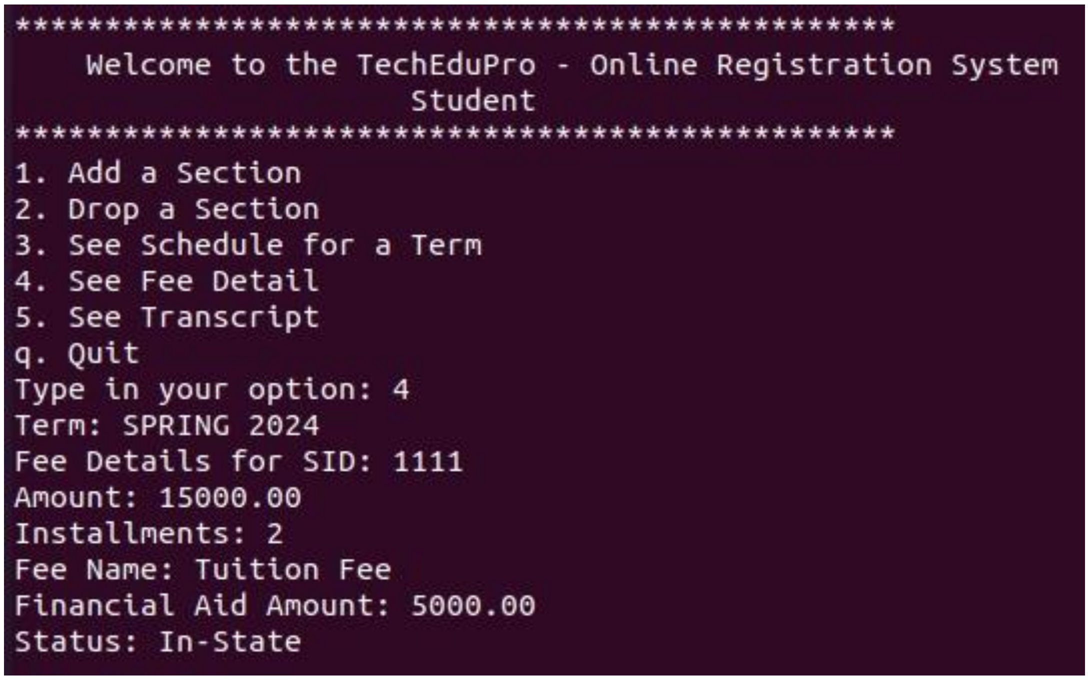
```
*************************************************
    Welcome to the TechEduPro - Online Registration System
                          Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option: 2
Enter CRN: 10101
CSC1010, Computers and Applications DROPPED.
Dropped enrollment for course with CRN 10101.
```

Figure 23- Drop section Interface



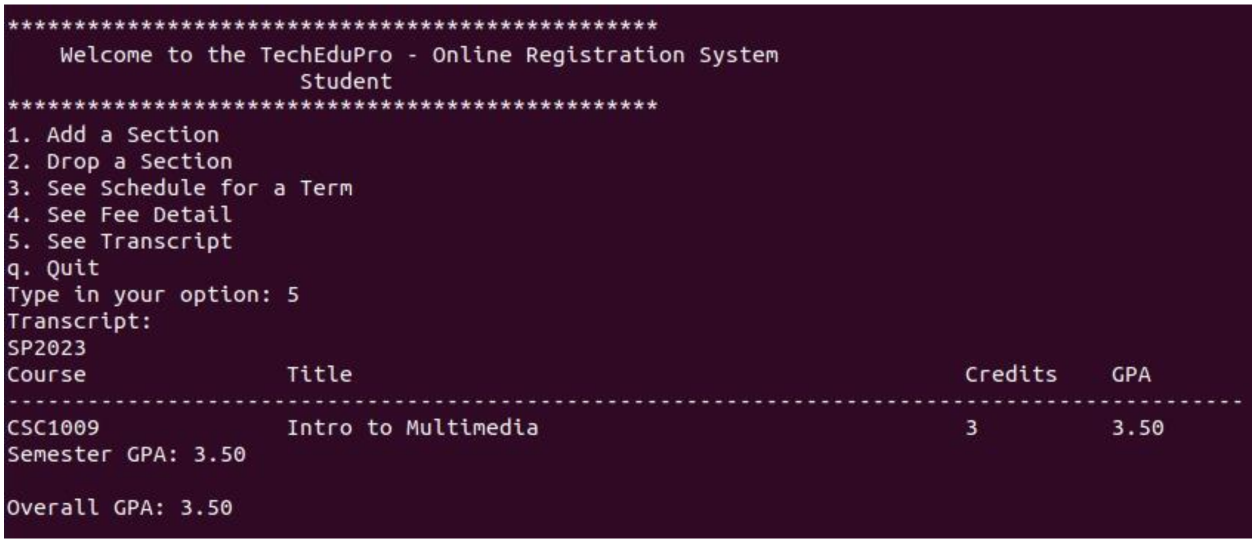
```
*************************************************
    Welcome to the TechEduPro - Online Registration System
                      Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option: 3
Enter the term (e.g., SP2024): SP2023
Option 3 Interface:
Term: SP2023
CRN        Course     Title                                     Days  Time        Room       Instructor
------------------------------------------------------------------------------------------------------------------
10105      CSC1009    Intro to Multimedia                       null  15:00-16:15 105Z        Dr. HARRY
```

Figure 24- See Schedule for a Term Interface



```
*************************************************
    Welcome to the TechEduPro - Online Registration System
                      Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option: 4
Term: SPRING 2024
Fee Details for SID: 1111
Amount: 15000.00
Installments: 2
Fee Name: Tuition Fee
Financial Aid Amount: 5000.00
Status: In-State
```

Figure 25- See Fee Details Interface

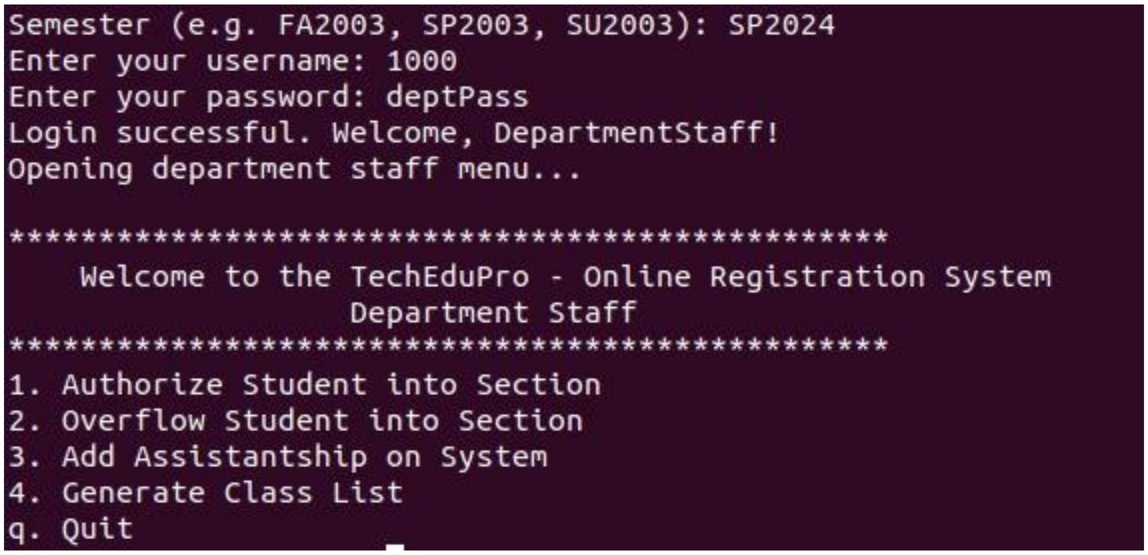


```
*************************************************
    Welcome to the TechEduPro - Online Registration System
                      Student
*************************************************
1. Add a Section
2. Drop a Section
3. See Schedule for a Term
4. See Fee Detail
5. See Transcript
q. Quit
Type in your option: 5
Transcript:
SP2023
Course               Title                                               Credits    GPA
---------------------------------------------------------------------------------------------
CSC1009              Intro to Multimedia                                 3          3.50
Semester GPA: 3.50

Overall GPA: 3.50
```

Figure 26- See Transcript Interface

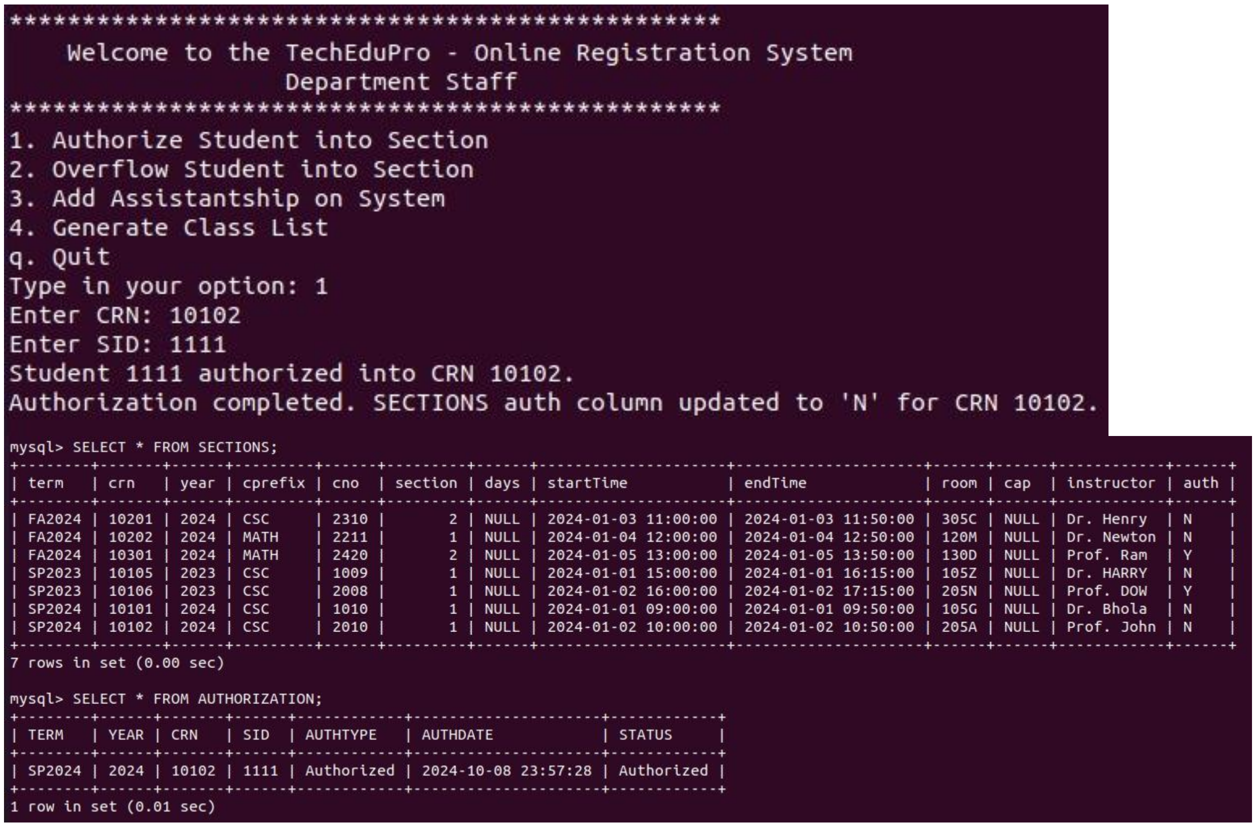
2. Department Staff Menu

The department staff menu is intended towards giving department-level administrators the ability to control student enrollments at prescribed periods called as term. The department staff can invoke students into special permission sections, whose occupancy has already been met. However, they also administrate student assistantships to graduate assistants or prepare class lists for any offered course during a particular term. This menu enables the department staff to better manage course enrollment of the department than especially those courses that are of special attention for example over enrolled or restricted ones.



```
Semester (e.g. FA2003, SP2003, SU2003): SP2024
Enter your username: 1000
Enter your password: deptPass
Login successful. Welcome, DepartmentStaff!
Opening department staff menu...

*************************************************
    Welcome to the TechEduPro - Online Registration System
                   Department Staff
*************************************************
1. Authorize Student into Section
2. Overflow Student into Section
3. Add Assistantship on System
4. Generate Class List
q. Quit
```

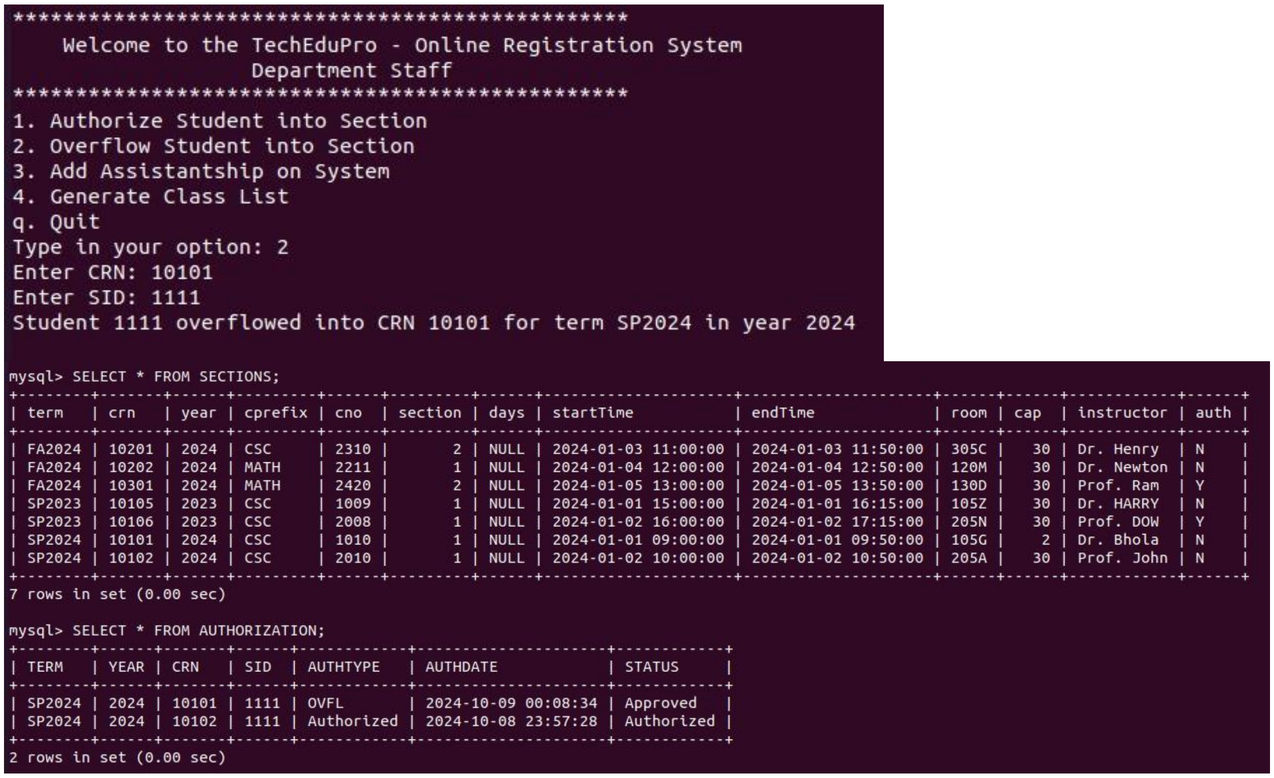
Figure 27- Department Staff Menu



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                   Department Staff
**************************************************
1. Authorize Student into Section
2. Overflow Student into Section
3. Add Assistantship on System
4. Generate Class List
q. Quit
Type in your option: 1
Enter CRN: 10102
Enter SID: 1111
Student 1111 authorized into CRN 10102.
Authorization completed. SECTIONS auth column updated to 'N' for CRN 10102.
```

```
mysql> SELECT * FROM SECTIONS;
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| term   | crn   | year | cprefix | cno  | section | days | startTime           | endTime             | room | cap  | instructor | auth |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| FA2024 | 10201 | 2024 | CSC     | 2310 |       2 | NULL | 2024-01-03 11:00:00 | 2024-01-03 11:50:00 | 305C | NULL | Dr. Henry  | N    |
| FA2024 | 10202 | 2024 | MATH    | 2211 |       1 | NULL | 2024-01-04 12:00:00 | 2024-01-04 12:50:00 | 120M | NULL | Dr. Newton | N    |
| FA2024 | 10301 | 2024 | MATH    | 2420 |       2 | NULL | 2024-01-05 13:00:00 | 2024-01-05 13:50:00 | 130D | NULL | Prof. Ram  | Y    |
| SP2023 | 10105 | 2023 | CSC     | 1009 |       1 | NULL | 2024-01-01 15:00:00 | 2024-01-01 16:15:00 | 105Z | NULL | Dr. HARRY  | N    |
| SP2023 | 10106 | 2023 | CSC     | 2008 |       1 | NULL | 2024-01-02 16:00:00 | 2024-01-02 17:15:00 | 205N | NULL | Prof. DOW  | Y    |
| SP2024 | 10101 | 2024 | CSC     | 1010 |       1 | NULL | 2024-01-01 09:00:00 | 2024-01-01 09:50:00 | 105G | NULL | Dr. Bhola  | N    |
| SP2024 | 10102 | 2024 | CSC     | 2010 |       1 | NULL | 2024-01-02 10:00:00 | 2024-01-02 10:50:00 | 205A | NULL | Prof. John | N    |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM AUTHORIZATION;
+--------+------+-------+------+------------+---------------------+------------+
| TERM   | YEAR | CRN   | SID  | AUTHTYPE   | AUTHDATE            | STATUS     |
+--------+------+-------+------+------------+---------------------+------------+
| SP2024 | 2024 | 10102 | 1111 | Authorized | 2024-10-08 23:57:28 | Authorized |
+--------+------+-------+------+------------+---------------------+------------+
1 row in set (0.01 sec)
```

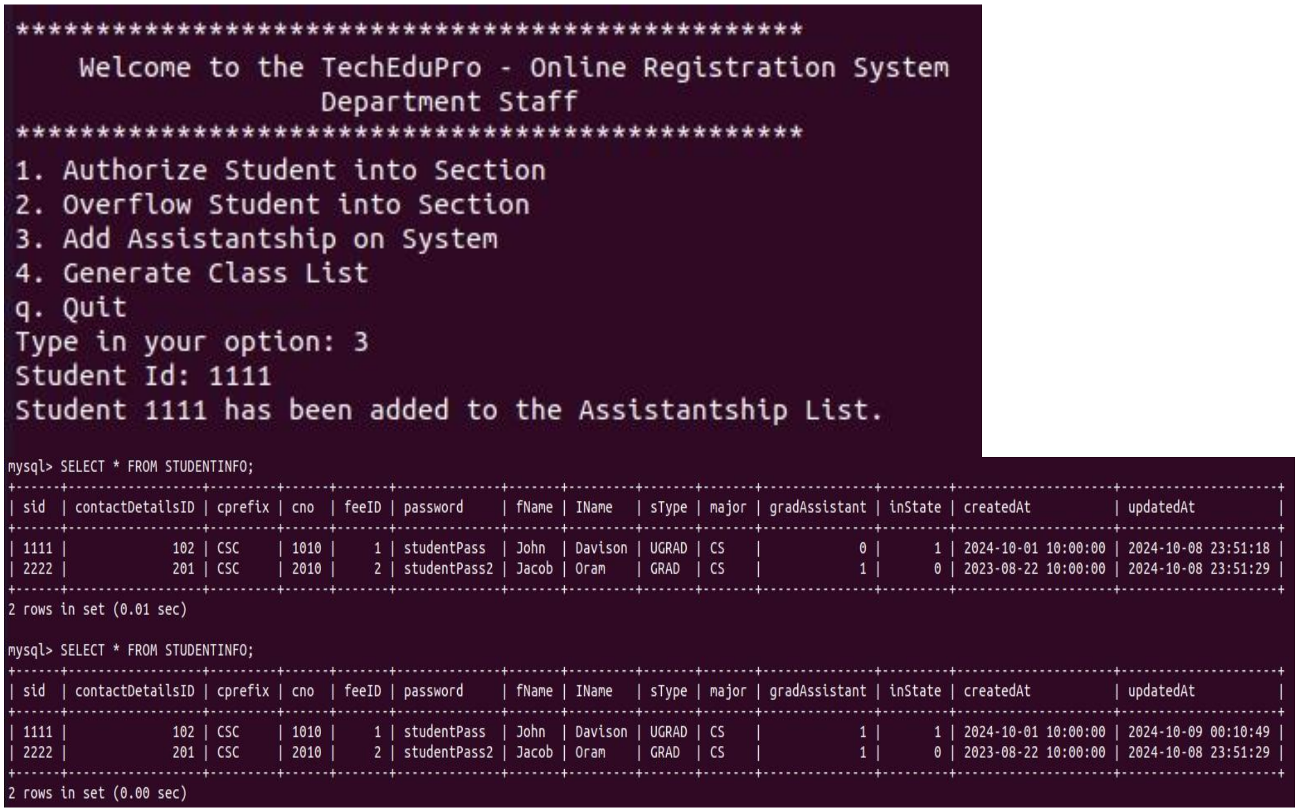
Figure 28- Authorize Student Interface



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                   Department Staff
**************************************************
1. Authorize Student into Section
2. Overflow Student into Section
3. Add Assistantship on System
4. Generate Class List
q. Quit
Type in your option: 2
Enter CRN: 10101
Enter SID: 1111
Student 1111 overflowed into CRN 10101 for term SP2024 in year 2024
```

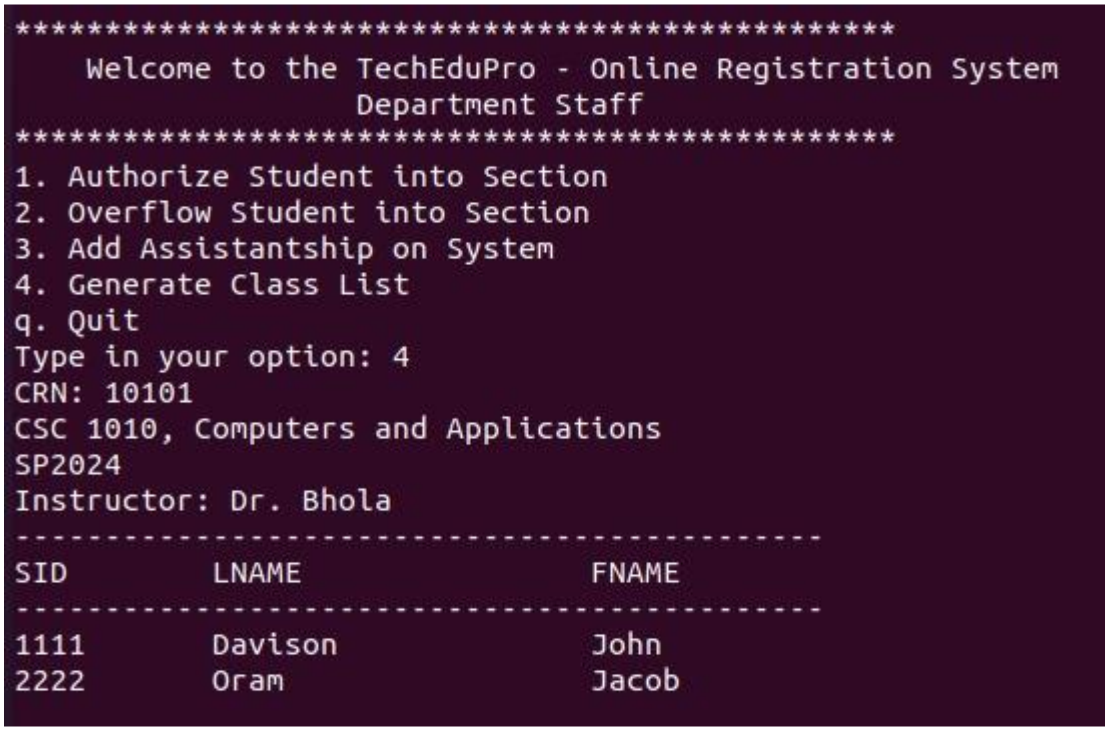
```
mysql> SELECT * FROM SECTIONS;
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| term   | crn   | year | cprefix | cno  | section | days | startTime           | endTime             | room | cap  | instructor | auth |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| FA2024 | 10201 | 2024 | CSC     | 2310 |       2 | NULL | 2024-01-03 11:00:00 | 2024-01-03 11:50:00 | 305C |   30 | Dr. Henry  | N    |
| FA2024 | 10202 | 2024 | MATH    | 2211 |       1 | NULL | 2024-01-04 12:00:00 | 2024-01-04 12:50:00 | 120M |   30 | Dr. Newton | N    |
| FA2024 | 10301 | 2024 | MATH    | 2420 |       2 | NULL | 2024-01-05 13:00:00 | 2024-01-05 13:50:00 | 130D |   30 | Prof. Ram  | Y    |
| SP2023 | 10105 | 2023 | CSC     | 1009 |       1 | NULL | 2024-01-01 15:00:00 | 2024-01-01 16:15:00 | 105Z |   30 | Dr. HARRY  | N    |
| SP2023 | 10106 | 2023 | CSC     | 2008 |       1 | NULL | 2024-01-02 16:00:00 | 2024-01-02 17:15:00 | 205N |   30 | Prof. DOW  | Y    |
| SP2024 | 10101 | 2024 | CSC     | 1010 |       1 | NULL | 2024-01-01 09:00:00 | 2024-01-01 09:50:00 | 105G |    2 | Dr. Bhola  | N    |
| SP2024 | 10102 | 2024 | CSC     | 2010 |       1 | NULL | 2024-01-02 10:00:00 | 2024-01-02 10:50:00 | 205A |   30 | Prof. John | N    |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM AUTHORIZATION;
+--------+------+-------+------+------------+---------------------+------------+
| TERM   | YEAR | CRN   | SID  | AUTHTYPE   | AUTHDATE            | STATUS     |
+--------+------+-------+------+------------+---------------------+------------+
| SP2024 | 2024 | 10101 | 1111 | OVFL       | 2024-10-09 00:08:34 | Approved   |
| SP2024 | 2024 | 10102 | 1111 | Authorized | 2024-10-08 23:57:28 | Authorized |
+--------+------+-------+------+------------+---------------------+------------+
2 rows in set (0.00 sec)
```

Figure 29- Overflow Student Interface



```
************************************************
    Welcome to the TechEduPro - Online Registration System
                   Department Staff
************************************************
1. Authorize Student into Section
2. Overflow Student into Section
3. Add Assistantship on System
4. Generate Class List
q. Quit
Type in your option: 3
Student Id: 1111
Student 1111 has been added to the Assistantship List.

mysql> SELECT * FROM STUDENTINFO;
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| sid  | contactDetailsID | cprefix | cno  | feeID | password    | fName | lName   | sType | major | gradAssistant | inState | createdAt           | updatedAt           |
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| 1111 |              102 | CSC     | 1010 |     1 | studentPass | John  | Davison | UGRAD | CS    |             0 |       1 | 2024-10-01 10:00:00 | 2024-10-08 23:51:18 |
| 2222 |              201 | CSC     | 2010 |     2 | studentPass2 | Jacob | Oram    | GRAD  | CS    |             1 |       0 | 2023-08-22 10:00:00 | 2024-10-08 23:51:29 |
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
2 rows in set (0.01 sec)

mysql> SELECT * FROM STUDENTINFO;
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| sid  | contactDetailsID | cprefix | cno  | feeID | password    | fName | lName   | sType | major | gradAssistant | inState | createdAt           | updatedAt           |
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
| 1111 |              102 | CSC     | 1010 |     1 | studentPass | John  | Davison | UGRAD | CS    |             1 |       1 | 2024-10-01 10:00:00 | 2024-10-09 00:10:49 |
| 2222 |              201 | CSC     | 2010 |     2 | studentPass2 | Jacob | Oram    | GRAD  | CS    |             1 |       0 | 2023-08-22 10:00:00 | 2024-10-08 23:51:29 |
+------+------------------+---------+------+-------+-------------+-------+---------+-------+-------+---------------+---------+---------------------+---------------------+
2 rows in set (0.00 sec)
```

Figure 30- Add Assistantship on System Interface

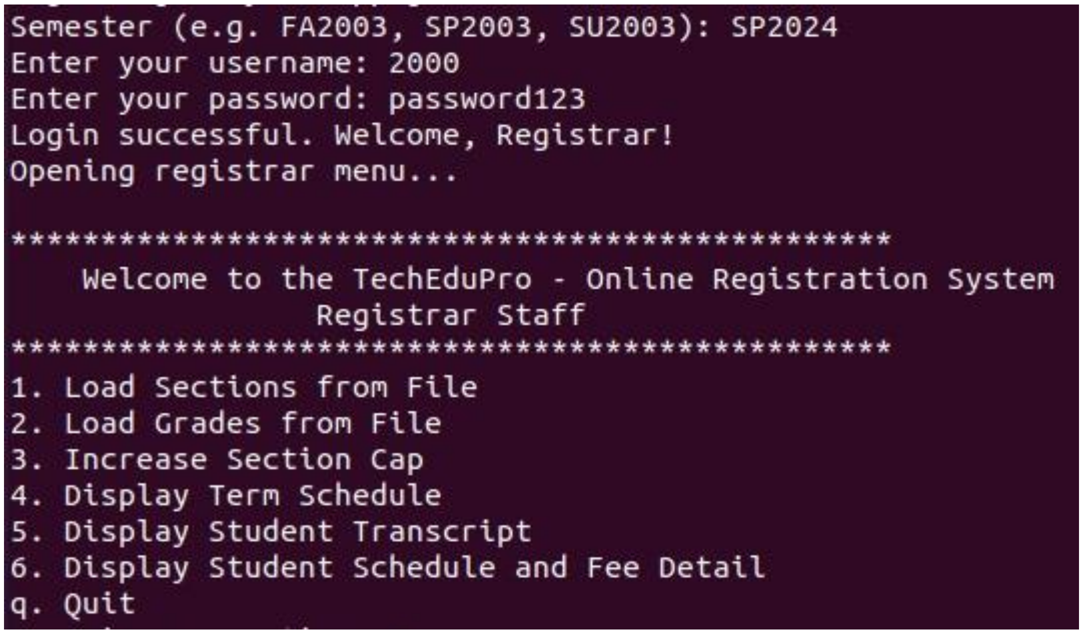


```
************************************************
    Welcome to the TechEduPro - Online Registration System
                   Department Staff
************************************************
1. Authorize Student into Section
2. Overflow Student into Section
3. Add Assistantship on System
4. Generate Class List
q. Quit
Type in your option: 4
CRN: 10101
CSC 1010, Computers and Applications
SP2024
Instructor: Dr. Bhola
---------------------------------------------
SID        LNAME                    FNAME
---------------------------------------------
1111       Davison                  John
2222       Oram                     Jacob
```

Figure 31- Generate Class List Interface

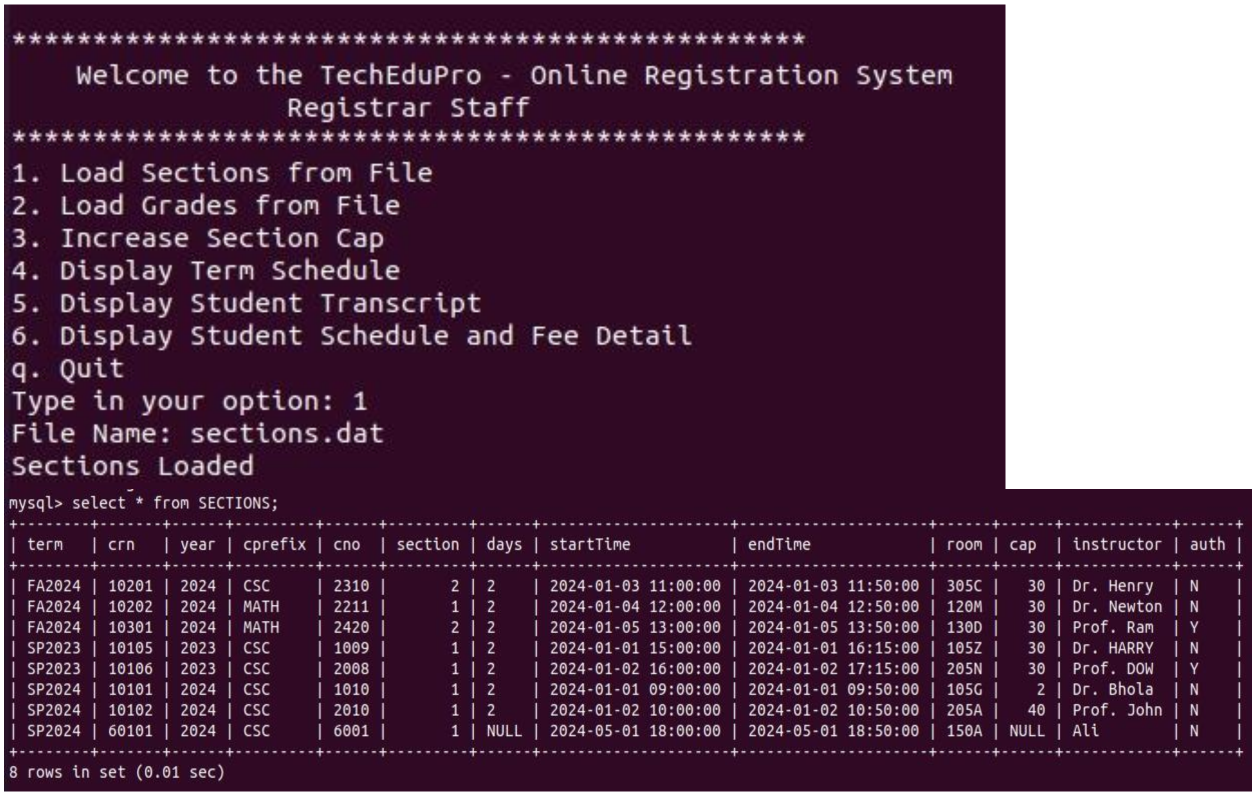
3. Registrar Staff Menu

In the registrar staff interface, the tools are more extended with the possibility to control the complete process of registration. Registrar staff can load the sections and grades in the system from external files; manage the course capacities by increasing the section caps; display the term schedule of all the courses. It is possible to generate some reports like student's transcript, fee details for particular terms. This administrative interface will ensure that registrar's staff will maintain integrity while carrying out the registration process, hence handling a wide range of activities efficiently with regard to course and student management.



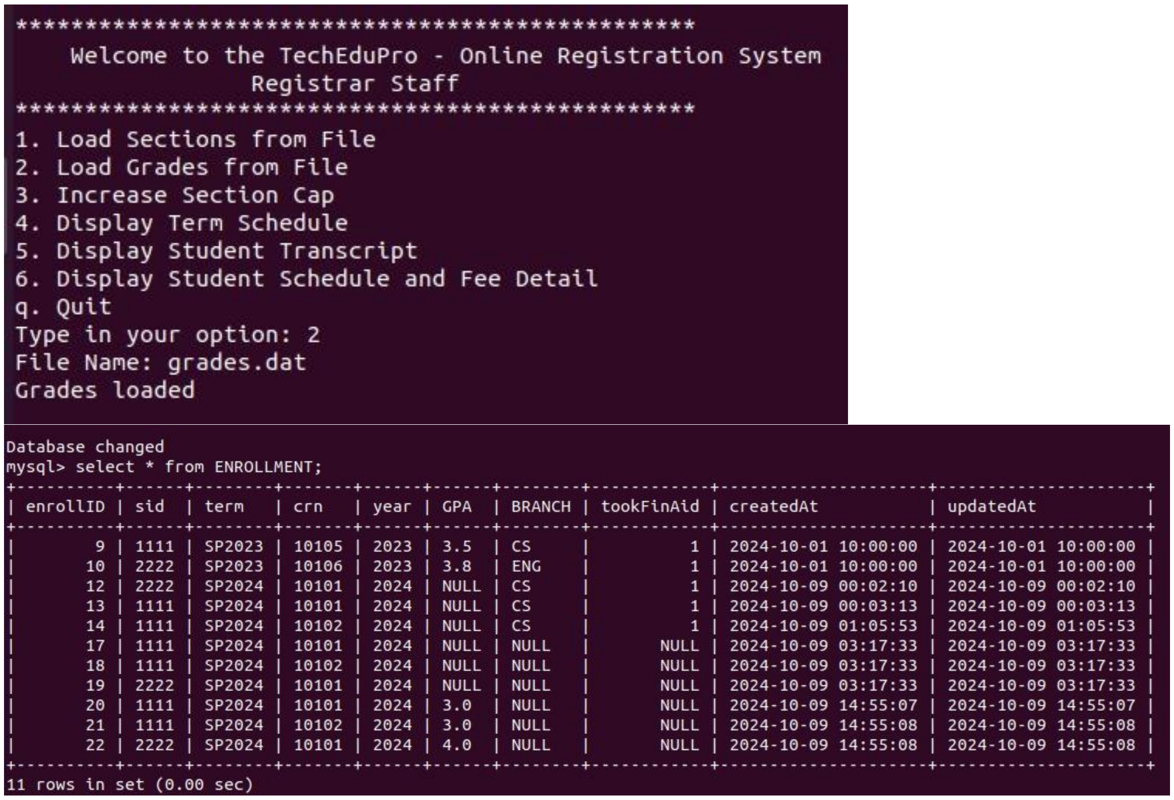
```
Semester (e.g. FA2003, SP2003, SU2003): SP2024
Enter your username: 2000
Enter your password: password123
Login successful. Welcome, Registrar!
Opening registrar menu...

*************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
*************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
```

Figure 32- Registrar Staff Menu



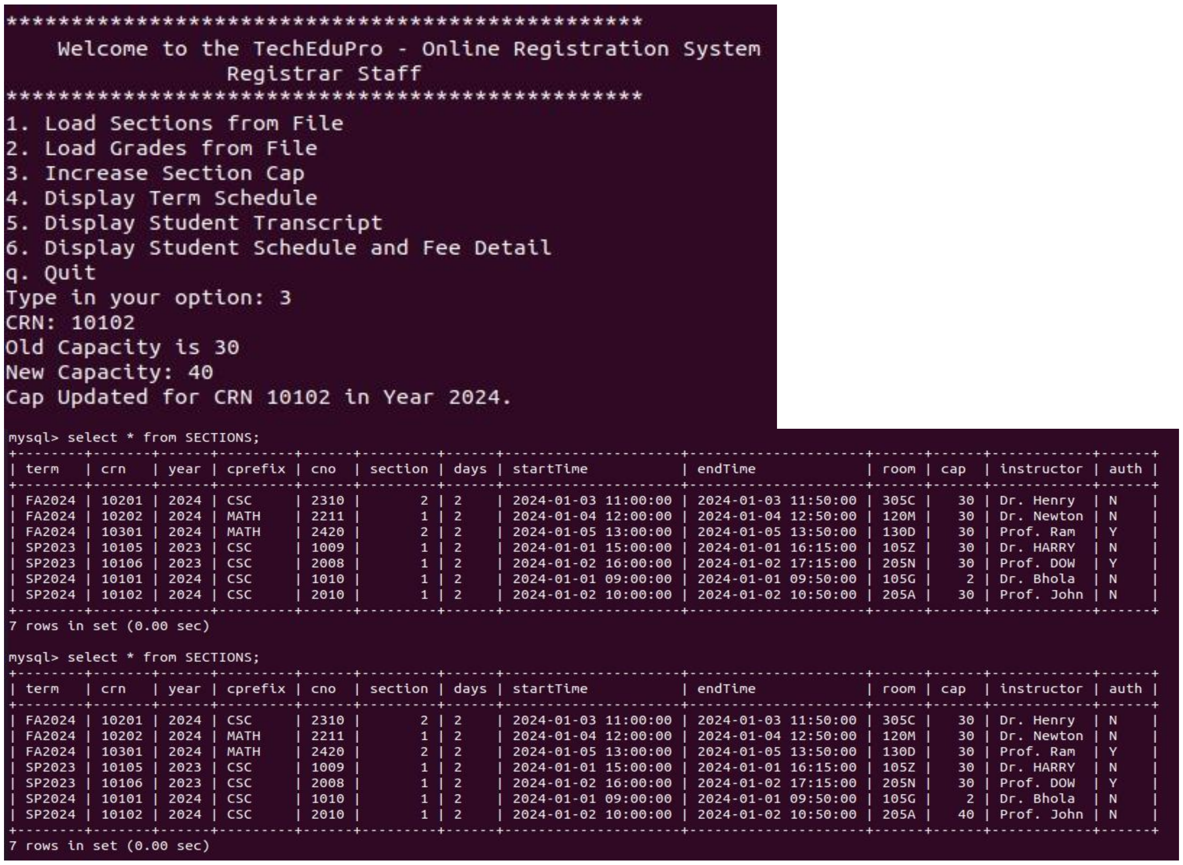
```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 1
File Name: sections.dat
Sections Loaded
mysql> select * from SECTIONS;
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| term   | crn   | year | cprefix | cno  | section | days | startTime           | endTime             | room | cap  | instructor | auth |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| FA2024 | 10201 | 2024 | CSC     | 2310 |       2 | 2    | 2024-01-03 11:00:00 | 2024-01-03 11:50:00 | 305C |   30 | Dr. Henry  | N    |
| FA2024 | 10202 | 2024 | MATH    | 2211 |       1 | 2    | 2024-01-04 12:00:00 | 2024-01-04 12:50:00 | 120M |   30 | Dr. Newton | N    |
| FA2024 | 10301 | 2024 | MATH    | 2420 |       2 | 2    | 2024-01-05 13:00:00 | 2024-01-05 13:50:00 | 130D |   30 | Prof. Ram  | Y    |
| SP2023 | 10105 | 2023 | CSC     | 1009 |       1 | 2    | 2024-01-01 15:00:00 | 2024-01-01 16:15:00 | 105Z |   30 | Dr. HARRY  | N    |
| SP2023 | 10106 | 2023 | CSC     | 2008 |       1 | 2    | 2024-01-02 16:00:00 | 2024-01-02 17:15:00 | 205N |   30 | Prof. DOW  | Y    |
| SP2024 | 10101 | 2024 | CSC     | 1010 |       1 | 2    | 2024-01-01 09:00:00 | 2024-01-01 09:50:00 | 105G |    2 | Dr. Bhola  | N    |
| SP2024 | 10102 | 2024 | CSC     | 2010 |       1 | 2    | 2024-01-02 10:00:00 | 2024-01-02 10:50:00 | 205A |   40 | Prof. John | N    |
| SP2024 | 60101 | 2024 | CSC     | 6001 |       1 | NULL | 2024-05-01 18:00:00 | 2024-05-01 18:50:00 | 150A | NULL | Ali        | N    |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
8 rows in set (0.01 sec)
```

Figure 33- Load Sections from File Interface



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 2
File Name: grades.dat
Grades loaded

Database changed
mysql> select * from ENROLLMENT;
+----------+------+--------+-------+------+------+--------+------------+---------------------+---------------------+
| enrollID | sid  | term   | crn   | year | GPA  | BRANCH | tookFinAid | createdAt           | updatedAt           |
+----------+------+--------+-------+------+------+--------+------------+---------------------+---------------------+
|        9 | 1111 | SP2023 | 10105 | 2023 | 3.5  | CS     |          1 | 2024-10-01 10:00:00 | 2024-10-01 10:00:00 |
|       10 | 2222 | SP2023 | 10106 | 2023 | 3.8  | ENG    |          1 | 2024-10-01 10:00:00 | 2024-10-01 10:00:00 |
|       12 | 2222 | SP2024 | 10101 | 2024 | NULL | CS     |          1 | 2024-10-09 00:02:10 | 2024-10-09 00:02:10 |
|       13 | 1111 | SP2024 | 10101 | 2024 | NULL | CS     |          1 | 2024-10-09 00:03:13 | 2024-10-09 00:03:13 |
|       14 | 1111 | SP2024 | 10102 | 2024 | NULL | CS     |          1 | 2024-10-09 01:05:53 | 2024-10-09 01:05:53 |
|       17 | 1111 | SP2024 | 10101 | 2024 | NULL | NULL   |       NULL | 2024-10-09 03:17:33 | 2024-10-09 03:17:33 |
|       18 | 1111 | SP2024 | 10102 | 2024 | NULL | NULL   |       NULL | 2024-10-09 03:17:33 | 2024-10-09 03:17:33 |
|       19 | 2222 | SP2024 | 10101 | 2024 | NULL | NULL   |       NULL | 2024-10-09 03:17:33 | 2024-10-09 03:17:33 |
|       20 | 1111 | SP2024 | 10101 | 2024 | 3.0  | NULL   |       NULL | 2024-10-09 14:55:07 | 2024-10-09 14:55:07 |
|       21 | 1111 | SP2024 | 10102 | 2024 | 3.0  | NULL   |       NULL | 2024-10-09 14:55:08 | 2024-10-09 14:55:08 |
|       22 | 2222 | SP2024 | 10101 | 2024 | 4.0  | NULL   |       NULL | 2024-10-09 14:55:08 | 2024-10-09 14:55:08 |
+----------+------+--------+-------+------+------+--------+------------+---------------------+---------------------+
11 rows in set (0.00 sec)
```

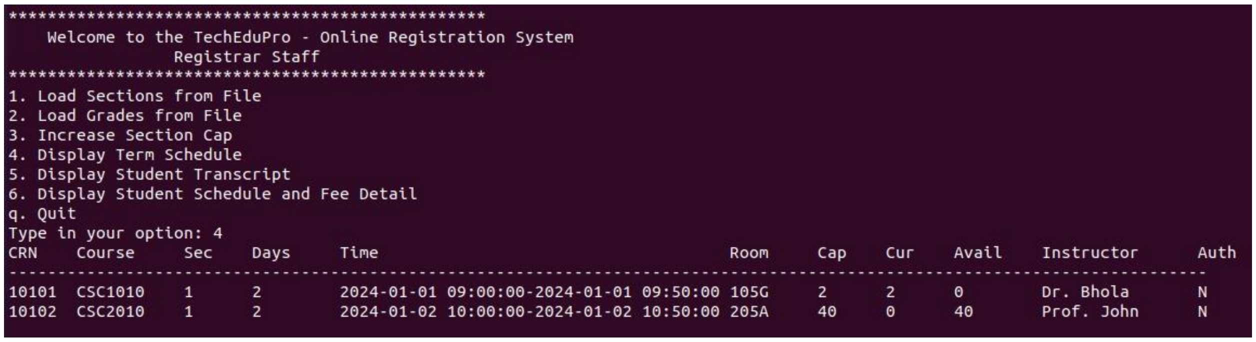
Figure 34- Load Grades from File Interface



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 3
CRN: 10102
Old Capacity is 30
New Capacity: 40
Cap Updated for CRN 10102 in Year 2024.
```

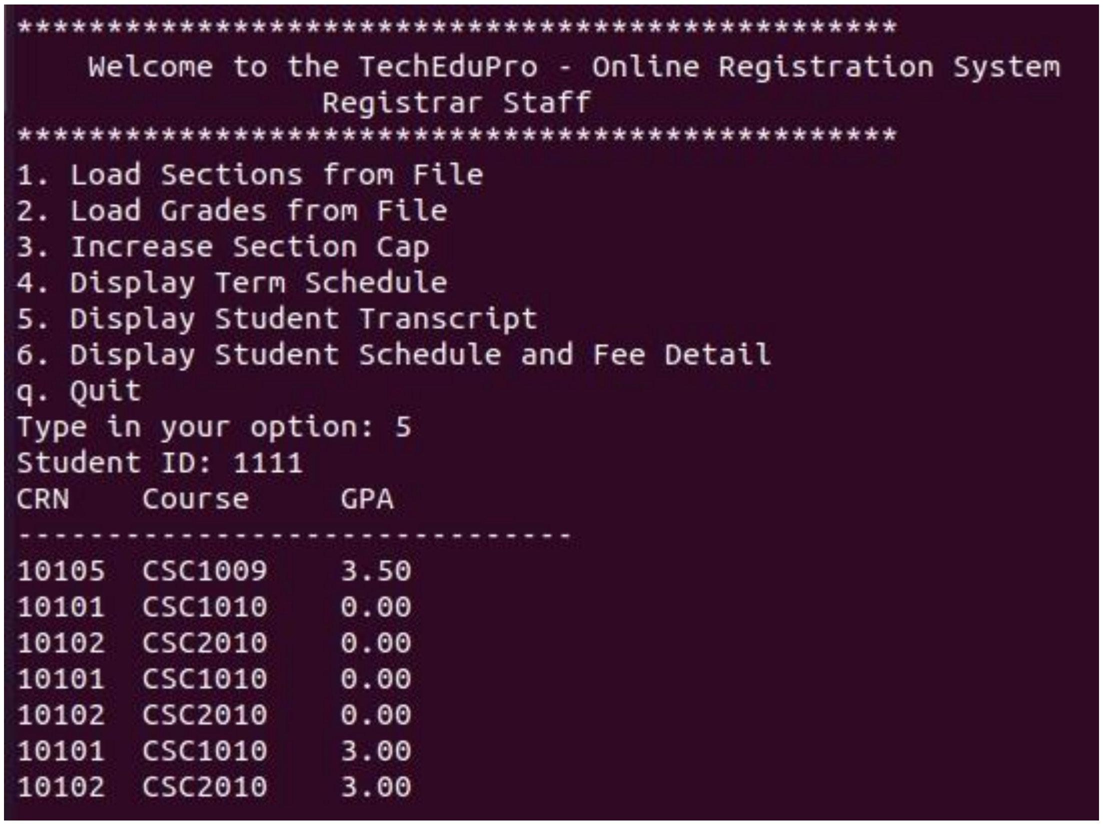
```
mysql> select * from SECTIONS;
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| term   | crn   | year | cprefix | cno  | section | days | startTime           | endTime             | room | cap  | instructor | auth |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| FA2024 | 10201 | 2024 | CSC     | 2310 |       2 | 2    | 2024-01-03 11:00:00 | 2024-01-03 11:50:00 | 305C |   30 | Dr. Henry  | N    |
| FA2024 | 10202 | 2024 | MATH    | 2211 |       1 | 2    | 2024-01-04 12:00:00 | 2024-01-04 12:50:00 | 120M |   30 | Dr. Newton | N    |
| FA2024 | 10301 | 2024 | MATH    | 2420 |       2 | 2    | 2024-01-05 13:00:00 | 2024-01-05 13:50:00 | 130D |   30 | Prof. Ram  | Y    |
| SP2023 | 10105 | 2023 | CSC     | 1009 |       1 | 2    | 2024-01-01 15:00:00 | 2024-01-01 16:15:00 | 105Z |   30 | Dr. HARRY  | N    |
| SP2023 | 10106 | 2023 | CSC     | 2008 |       1 | 2    | 2024-01-02 16:00:00 | 2024-01-02 17:15:00 | 205N |   30 | Prof. DOW  | Y    |
| SP2024 | 10101 | 2024 | CSC     | 1010 |       1 | 2    | 2024-01-01 09:00:00 | 2024-01-01 09:50:00 | 105G |    2 | Dr. Bhola  | N    |
| SP2024 | 10102 | 2024 | CSC     | 2010 |       1 | 2    | 2024-01-02 10:00:00 | 2024-01-02 10:50:00 | 205A |   30 | Prof. John | N    |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
7 rows in set (0.00 sec)

mysql> select * from SECTIONS;
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| term   | crn   | year | cprefix | cno  | section | days | startTime           | endTime             | room | cap  | instructor | auth |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
| FA2024 | 10201 | 2024 | CSC     | 2310 |       2 | 2    | 2024-01-03 11:00:00 | 2024-01-03 11:50:00 | 305C |   30 | Dr. Henry  | N    |
| FA2024 | 10202 | 2024 | MATH    | 2211 |       1 | 2    | 2024-01-04 12:00:00 | 2024-01-04 12:50:00 | 120M |   30 | Dr. Newton | N    |
| FA2024 | 10301 | 2024 | MATH    | 2420 |       2 | 2    | 2024-01-05 13:00:00 | 2024-01-05 13:50:00 | 130D |   30 | Prof. Ram  | Y    |
| SP2023 | 10105 | 2023 | CSC     | 1009 |       1 | 2    | 2024-01-01 15:00:00 | 2024-01-01 16:15:00 | 105Z |   30 | Dr. HARRY  | N    |
| SP2023 | 10106 | 2023 | CSC     | 2008 |       1 | 2    | 2024-01-02 16:00:00 | 2024-01-02 17:15:00 | 205N |   30 | Prof. DOW  | Y    |
| SP2024 | 10101 | 2024 | CSC     | 1010 |       1 | 2    | 2024-01-01 09:00:00 | 2024-01-01 09:50:00 | 105G |    2 | Dr. Bhola  | N    |
| SP2024 | 10102 | 2024 | CSC     | 2010 |       1 | 2    | 2024-01-02 10:00:00 | 2024-01-02 10:50:00 | 205A |   40 | Prof. John | N    |
+--------+-------+------+---------+------+---------+------+---------------------+---------------------+------+------+------------+------+
7 rows in set (0.00 sec)
```

Figure 35- Increase Section Cap Interface



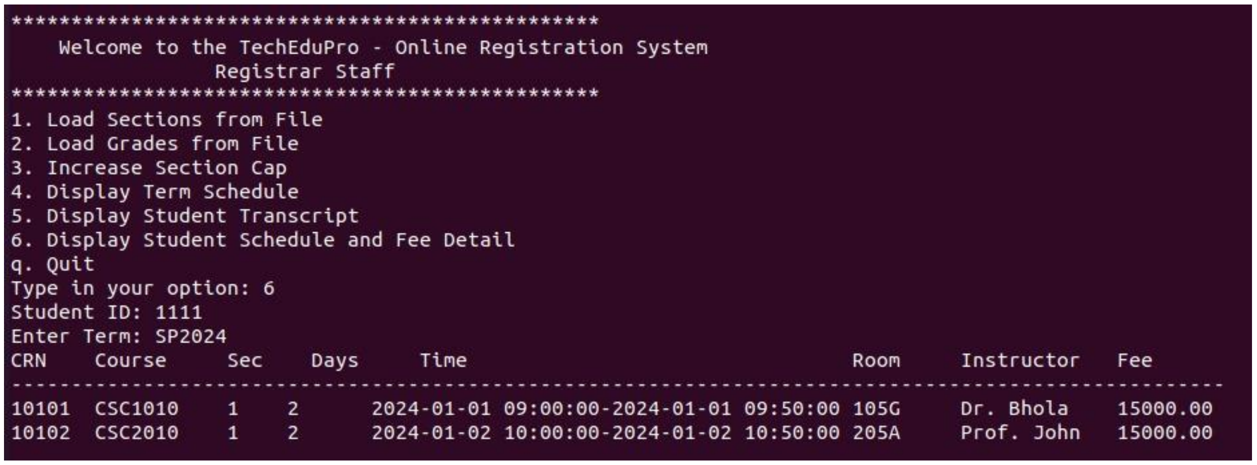
```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 4
CRN    Course     Sec    Days     Time                                    Room     Cap     Cur     Avail    Instructor      Auth
------------------------------------------------------------------------------------------------------------------------
10101  CSC1010    1      2        2024-01-01 09:00:00-2024-01-01 09:50:00 105G     2       2       0        Dr. Bhola       N
10102  CSC2010    1      2        2024-01-02 10:00:00-2024-01-02 10:50:00 205A     40      0       40       Prof. John      N
```

Figure 36- Display Term Schedule Interface



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 5
Student ID: 1111
CRN    Course     GPA
-------------------------------
10105  CSC1009    3.50
10101  CSC1010    0.00
10102  CSC2010    0.00
10101  CSC1010    0.00
10102  CSC2010    0.00
10101  CSC1010    3.00
10102  CSC2010    3.00
```

Figure 37- Display Student Transcript Interface



```
**************************************************
    Welcome to the TechEduPro - Online Registration System
                 Registrar Staff
**************************************************
1. Load Sections from File
2. Load Grades from File
3. Increase Section Cap
4. Display Term Schedule
5. Display Student Transcript
6. Display Student Schedule and Fee Detail
q. Quit
Type in your option: 6
Student ID: 1111
Enter Term: SP2024
CRN    Course     Sec    Days     Time                                Room     Instructor    Fee
--------------------------------------------------------------------------------------------
10101  CSC1010    1      2        2024-01-01 09:00:00-2024-01-01 09:50:00 105G     Dr. Bhola     15000.00
10102  CSC2010    1      2        2024-01-02 10:00:00-2024-01-02 10:50:00 205A     Prof. John    15000.00
```

Figure 38- Display Student Schedule and FeeeDetail Interface

IV. KEY FUNCTIONALITIES

The built system has several core features, mainly course enrollment, fee management, and transcript generation. These facilities will make the entire experience very easy for students in the areas of course registration, computation of fees based on various variables, and easy access to academic records. The system would ease workload for both the students and administrative personnel.

1. Course Enrollment

The purpose of the system's course enrollment process is to assist students in signing up for courses in conformity with the numerous academic requirements. The students are given selection of courses which are on offer in a specific semester complete with some relevant information like numbers attached to the courses, course titles, time for meeting and names of the course's instructors. Depending on what has been specified by the school, the system ensures that the required prerequisites are fulfilled before a student is allowed to register in particular courses. More importantly, the system keeps track of the number of students in a particular course such that over enrollment is only possible under the management approval. The students do not enrol into classes that collide in their schedule and are advised not to exceed the maximum limits for hours: 20 hours for undergraduate students and 15 hours for graduate students. This ensures that every student has not been overloaded with courses taking into account the policies of the university. In case a particular class or course has to be offered but it requires some prerequisites or approval or is already booked up then students must get allocated space with strict permission by the departmental staff.

2. Fee Management

The system of fee administration in the system makes certain that each student is billed the relevant fees, depending on enrollment in the various courses as well as the individual's economic situations. Different criteria are used to determine the tuition, for instance, the total number of credit hours that have been registered, the place of residence of the particular student whether in state or out of state and whether the student has been awarded any finance like scholarships or student loans. Other costs, such as technology, health, activity and transportation fees, are also charged and included in the summary of the charges for the term to the students as the system also implements additional fees automatically. For those who still owe money and there is a possibility of making the payment in installments, students are informed of the installments and late payments will carry penalties for late payments. The system also captures the students' financial help where after the financial help is given, the student's balance is computed. Fee management, as part of the whole registration process, is quite crucial in the depiction of the students' accountabilities towards education as well as in the management of fee collection for the university administration.

3. Transcript Generation

The system allows for the retrieval of academic transcripts with ease as these are very essential documents that have records of academic activities undertaken by students as well as performance indicators. Each transcript includes a listing of all objectives fulfilled by the student over the years at the university, arranged year after year within academic calendars together with the corresponding grades. The system further provides both the term GPA as well the cumulative GPA which presents the status of a student academically. More than basic course and grades, the transcript goes further to include most of the courses by their CRN, course title and the credit hours for each course. This capability makes it easy for students to grasp the non-obvious metrics of their academic performance and degree completion. The process of generation of the transcripts in her case is a real time process whereby each time new grades are put in after the end of the term, the transcripts are instantly generated. This is useful for students who want to monitor their goals and deadlines in terms of academics and apply to internships or send clerical work to employers or graduate schools. It also makes it easier for academic advisors as well as university staff in carrying out their non-teaching functions.

V. DISCUSSION

Team Project 1 involved working with a range of technologies critical to building the University Registration System. The project required hands-on implementation and integration of various tools, including (1) setting up MySQL using Docker containers, (2) establishing Java Database Connectivity (JDBC) for interacting with the MySQL database, (3) developing Java applications to handle user interaction and system logic, and (4) utilizing Maven for managing dependencies and building the Java project. Each task provided the team with a comprehensive understanding of system design, implementation, and database integration. The project presented both successes and challenges, fostering collaborative problem-solving within the team.

1. Challenges Faced

During the University registration system development process, some difficulties were faced, mainly in managing the system database and its overall integrity. For example, one of the major challenges was implementing proper foreign key relationships to create proper inter table connections, which included linking of the students with their respective courses, enrollment and fee records. Rather, it needed great focus to make sure that the system did not compromise data such as when a student incurred more than one fee charged on their course enrollment while factoring in the course name. In solving the above issues, the appropriate primary and foreign key designs were put in place, and quite a number of error control strategies were adopted, cont. For instance, preventing students from enrolling in time conflicting courses or going beyond the credit hours prescribed because of course fears, or practice required nerves at both database and application side.

Aside from handling foreign keys, managing concurrent transactions was also challenging, more so when the system was being accessed by numerous users at the same time. It was important to introduce adequate transaction isolation levels in order to ensure there were no race conditions or deadlocks and that the database was always in a consistent state. Additionally, while recognizing the necessity of authorization for course registration, it was difficult to accommodate all the requirements, such as prerequisite checks and overflow control.

2. Performance Considerations

Performance was a central concern during the design and development of the university registration system, and even more, during its user testing and deployment. Managing database query performance was also important in the first place due to the system expecting concurrent users. In order to favor response time when carrying out recurrent processes like course registration and transcript printing, indexing was done on certain fields namely SID, CNO, and CRN. In addition, minimizing nested level query clauses to a reasonable level, using the joins effectively and other query enhancing approaches were also utilized to improve data retrieval from the large volumes of data stored in the database. Constant instances of performance profiling were done to pinpoint the performance hindrances and actions were taken to cut down the time taken to execute the queries and the way the database connections were handled. Out of efficiency, responses were optimized resource wise and optimal caching was also done whenever deemed necessary to enhance response time in stressful situation of many requests on the system. Such improvement efforts were significant in ensuring smooth use of the system especially during the registration peak's when the system is the most burdened.

VI. CONCLUSION

In this project, we had practical work in creating an efficient registration system for a university with the use of Java and MySQL. We were engaged in things that included database modification, fee computation and its activities like user registrations and course enrolment. As a group, these tasks were divided according to the strength of each group member and it ensured that none of the activities was overlooked.

In order to execute the project, we made sure that each individual had the sufficient materials and software such as Docker for MySQL containerization and Java IDE for codes. We were able to do this in a more tactical way, and that is why we sought to make some adjustments a s far as things like database queries and UI were concerned.

Without teamwork, this project would not have been successfully completed. The participation of each individual was very active for instance, contributions during discussions, testing and even debugging which made the final product better than if done individually. Difficult issues surrounding database-related interaction and the legitimacy of the data was distributed efficiently among the group indicating that such tasks are best performed as a group. In conclusion, this project indeed enhanced our technical capacities but also improved our skills on working with other people and meeting new requirements.

Acknowledge

I acknowledge the invaluable guidance and support of my professor, whose expertise and encouragement were instrumental in completing this assignment. Special thanks to my institution for providing the necessary resources and environment to carry out this work successfully.

REFERENCES

1. 'Brien, J. A., & Marakas, G. M. (2011). *Management Information Systems* (10th ed.). McGraw-Hill.
2. Chen, W., & Zhao, H. (2019). A study on the improvement of student registration system. *International Journal of Emerging Technologies in Learning (iJET)*, 14(4), 95-106. https://doi.org/10.3991/ijet.v14i04.9324
3. Sabri, N. K., & Ali, H. (2020). Designing a web-based student registration system for universities: A case study of Yarmouk University. *Journal of Computer Science and Technology*, 35(1), 115-123. https://doi.org/10.1007/s11390-020-00121-0
4. Tham, J., & Ramasamy, P. (2018). E-registration systems in Malaysian universities: Issues and challenges. *International Journal of Information Systems and Change Management*, 11(1), 1-15. https://doi.org/10.1504/IJISCM.2018.091106
5. Aksakallı, S. (2019). A cloud-based student registration system: Development and implementation. *Journal of Computer Applications in Technology*, 64(1), 44-52. https://doi.org/10.1002/cat.2353
6. Obad, M. A., & Sulaiman, A. (2020). Implementing a student registration system using a mobile application: A case study. *Journal of Software Engineering and Applications*, 13(3), 96-109. https://doi.org/10.4236/jsea.2020.133008
7. Al-Sharhan, S., & Al-Khaldi, M. (2019). Development of an online university student registration system: A case study of Kuwait University. *International Journal of Computer Applications*, 975, 12-18. https://doi.org/10.5120/ijca2019918453
8. Mabe, A., & Asher, D. (2016). Student information systems: A critical review of the literature. *Journal of Computing in Higher Education*, 28(1), 77-99. https://doi.org/10.1007/s12528-015-9108-1
9. Wozniak, M., & Skawronski, P. (2021). The use of mobile applications in student registration systems. *Mobile Networks and Applications*, 26(3), 1220-1229. https://doi.org/10.1007/s11036-020-01654-1
10. Ranjan, J., & Arora, N. (2020). An automated student registration system using cloud computing. *International Journal of Cloud Computing and Services Science*, 9(1), 55-63. https://doi.org/10.11591/ijccs.v9i1.4960
11. Nithya, R., & Vidhya, S. (2017). A study on online student registration system for educational institutions. *International Journal of Research in Engineering and Technology*, 6(2), 36-40. https://doi.org/10.15623/ijret.2017.0602005

12. Kharbanda, P., & Singla, S. (2019). Implementation of a user-friendly student registration system using PHP and MySQL. *International Journal of Engineering and Advanced Technology*, 8(6), 3416-3421. https://doi.org/10.35940/ijeat.F9405.088619
13. Herath, T., & Hettiarachchi, S. (2018). Student registration system: A case study of Sabaragamuwa University of Sri Lanka. *International Journal of Scientific and Research Publications*, 8(3), 216-219. https://doi.org/10.29322/IJSRP.8.3.2018.p7599
14. Kumar, S., & Kumari, S. (2020). Web-based student registration system: A review. *International Journal of Computer Applications*, 975, 20-25. https://doi.org/10.5120/ijca2020919581
15. Tarek, A., & Shaikh, A. (2019). Enhancing university student registration system: An approach using web technologies. *Journal of Software Engineering and Applications*, 12(3), 138-145. https://doi.org/10.4236/jsea.2020.123008

Appendix

1. Creating tables for the Entities

```
--Authorization
CREATE TABLE AUTHORIZATION (
  TERM varchar(10) NOT NULL,
  YEAR int NOT NULL,
  CRN bigint NOT NULL,
  SID bigint NOT NULL,
  AUTHTYPE varchar(10) NOT NULL,
  AUTHDATE datetime DEFAULT NULL,
  STATUS varchar(10) DEFAULT NULL,
  PRIMARY KEY (TERM,YEAR,CRN,SID,AUTHTYPE),
  KEY FK_AUTHORIZATION_SID (SID),
  KEY FK_AUTHORIZATION_SECTIONS (TERM,CRN,YEAR),
  CONSTRAINT FK_AUTHORIZATION_SECTIONS FOREIGN KEY (TERM, CRN, YEAR)
REFERENCES SECTIONS (term, crn, year) ON DELETE CASCADE,
  CONSTRAINT FK_AUTHORIZATION_SID FOREIGN KEY (SID) REFERENCES
STUDENTINFO (sid) ON DELETE CASCADE
)

--ContactDetails
CREATE TABLE CONTACTDETAILS (
  contactDetailID bigint NOT NULL,
  phoneNo decimal(15,0) DEFAULT NULL,
  landlineNo decimal(15,0) DEFAULT NULL,
  address varchar(50) DEFAULT NULL,
  email varchar(50) DEFAULT NULL,
  PRIMARY KEY (contactDetailID)
)

--Courses
CREATE TABLE Courses (
  cprefix varchar(10) NOT NULL,
  cno bigint NOT NULL,
  cTitle varchar(50) DEFAULT NULL,
  cHours timestamp NULL DEFAULT NULL,
  cDay varchar(10) DEFAULT NULL,
  cBooks varchar(50) DEFAULT NULL,
  tid bigint DEFAULT NULL,
  createdAt timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  cCredits int DEFAULT NULL,
  PRIMARY KEY (cprefix,cno)
)

--Enrollment
CREATE TABLE ENROLLMENT (
  enrollID bigint NOT NULL,
  sid bigint DEFAULT NULL,
  term varchar(10) DEFAULT NULL,
  crn bigint DEFAULT NULL,
  year date DEFAULT NULL,
  GPA varchar(5) DEFAULT NULL,
  BRANCH varchar(50) DEFAULT NULL,
  tookFinAid tinyint(1) DEFAULT NULL,
  createdAt timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (enrollID),
  KEY FK_ENROLLMENT_SID (sid),
  KEY FK_ENROLLMENT_SECTIONS (crn,term),
  CONSTRAINT FK_ENROLLMENT_SECTIONS FOREIGN KEY (crn, term) REFERENCES
SECTIONS (crn, term) ON DELETE CASCADE,
  CONSTRAINT FK_ENROLLMENT_SID FOREIGN KEY (sid) REFERENCES
STUDENTINFO (sid) ON DELETE CASCADE
)

--Fees
CREATE TABLE FEES (
  FeeID bigint NOT NULL,
  Amount decimal(10,2) DEFAULT NULL,
  Medium varchar(50) DEFAULT NULL,
  installments tinyint(1) DEFAULT NULL,
  feeName varchar(50) DEFAULT NULL,
  financialAidAmount decimal(10,2) DEFAULT NULL,
  inStateOrOutOfState tinyint(1) DEFAULT NULL,
  inStateOrOutOfStateFeeAmount decimal(10,2) DEFAULT NULL,
  lastDateOfSubmission date DEFAULT NULL,
  lateFeePenalty decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (FeeID)
)

--Sections
CREATE TABLE SECTIONS (
  term varchar(10) NOT NULL,
  crn bigint NOT NULL,
  year int NOT NULL,
  cprefix varchar(10) DEFAULT NULL,
  cno bigint DEFAULT NULL,
  section int DEFAULT NULL,
  days varchar(8) DEFAULT NULL,
  startTime timestamp NULL DEFAULT NULL,
  endTime timestamp NULL DEFAULT NULL,
  room varchar(5) DEFAULT NULL,
  cap int DEFAULT NULL,
  instructor varchar(20) DEFAULT NULL,
  auth varchar(3) DEFAULT NULL,
  PRIMARY KEY (term,crn,year),
  KEY fk_sections_course (cprefix,cno),
  KEY idx_sections_crn_term (crn,term),
  CONSTRAINT fk_sections_course FOREIGN KEY (cprefix, cno) REFERENCES Courses
(cprefix, cno) ON DELETE CASCADE ON UPDATE CASCADE
)

--Staff
CREATE TABLE STAFF (
  tid bigint NOT NULL,
  contactDetailId bigint DEFAULT NULL,
  cprefix varchar(10) DEFAULT NULL,
  cno bigint DEFAULT NULL,
  password varchar(20) DEFAULT NULL,
  fName varchar(50) DEFAULT NULL,
  lName varchar(50) DEFAULT NULL,
  staffType varchar(20) DEFAULT NULL,
  availability varchar(20) DEFAULT NULL,
  Designation varchar(20) DEFAULT NULL,
  createdAt timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (tid),
  KEY fk_staff_contactDetails (contactDetailId),
  KEY fk_staff_course (cprefix,cno),
  CONSTRAINT fk_staff_contactDetails FOREIGN KEY (contactDetailId) REFERENCES
CONTACTDETAILS (contactDetailID) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT fk_staff_course FOREIGN KEY (cprefix, cno) REFERENCES Courses
(cprefix, cno) ON DELETE CASCADE ON UPDATE CASCADE
)
```

2. Login.java

```
import java.io.*;
import java.sql.*;
import java.util.*;

public class MyJavaApp {
    // Create a map to store users and their roles (in a real scenario, you'd get this from a DB)

    public static void main(String[] args) {

        try (Scanner scanner = new Scanner(System.in)) {

            System.out.print("Semester (e.g. FA2003, SP2003, SU2003): ");
            String semester = scanner.nextLine();

            System.out.print("Enter your username: ");
            String username = scanner.nextLine();

            System.out.print("Enter your password: ");
            String password = scanner.nextLine();

            // Authenticate user from the database
            String role = authenticateUser(username, password);

            if (role != null) {
                System.out.println("Login successful. Welcome, " + role + "!");
                // Open the respective menu based on the role
                openMenuForRole(role,username,semester);
            } else {
                System.out.println("Login failed. Invalid credentials.");
            }
        }

    }
    // Authentication method
    private static String authenticateUser(String username, String password) {
        String role = null;
        try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3333/myPNW", "root", "root")) {
            // Check if the user is a student
            String studentQuery = "SELECT password, 'Student' as role FROM STUDENTINFO
WHERE sid = ?";
            try (PreparedStatement stmt = conn.prepareStatement(studentQuery)) {
                stmt.setString(1, username);
                try (ResultSet rs = stmt.executeQuery()) {
                    if (rs.next() && rs.getString("password").equals(password)) {
                        return rs.getString("role");
                    }
                }
            }

            // If not a student, check if the user is a staff member
            String staffQuery = "SELECT password, staffType as role FROM STAFF WHERE tid =
?";
            try (PreparedStatement stmt = conn.prepareStatement(staffQuery)) {
                stmt.setString(1, username);
                try (ResultSet rs = stmt.executeQuery()) {
                    if (rs.next() && rs.getString("password").equals(password)) {
                        return rs.getString("role");  // role could be Registrar, DepartmentStaff, etc.
                    }
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Database connection error.");
        }
        return role; // Return null if authentication fails
    }


    private static void openMenuForRole(String role,String username,String semester) {
        Scanner scanner = new Scanner(System.in);
        switch (role) {
            case "Student":
                // Open student menu
                System.out.println("Opening student menu...");
                studentMenu(scanner,username,semester);
                break;
            case "Registrar":
                // Open registrar menu
                System.out.println("Opening registrar menu...");
                registrarMenu(scanner,semester);
                break;
            case "DepartmentStaff":
                // Open department staff menu

                System.out.println("Opening department staff menu...");
                departmentMenu(scanner,semester);
                break;
            default:
                System.out.println("Role not recognized.");
        }
    }


    // Registrar menu
    private static void registrarMenu(Scanner scanner, String semester) {
        String option;
        do {
            System.out.println("\n**************************************************");
            System.out.println("    Welcome to the TechEduPro - Online Registration System");
            System.out.println("                    Registrar Staff                    ");
            System.out.println("**************************************************");
            System.out.println("1. Load Sections from File");
            System.out.println("2. Load Grades from File");
            System.out.println("3. Increase Section Cap");
            System.out.println("4. Display Term Schedule");
            System.out.println("5. Display Student Transcript");
            System.out.println("6. Display Student Schedule and Fee Detail");
            System.out.println("q. Quit");

            System.out.print("Type in your option: ");
            option = scanner.next();
            switch (option) {
                case "1":
                    loadSectionsFromFile();
                    break;
                case "2":
                    loadGradesFromFile();
                    break;
                case "3":
                    increaseSectionCap(semester);
                    break;
                case "4":
```

```
                displayTermSchedule(semester);
                break;
            case "5":
                displayStudentTranscript();
                break;
            case "6":
                displayStudentScheduleAndFee();
                break;
            case "q":
                System.out.println("Returning to the main menu...");
                break;
            default:
                System.out.println("Invalid option.");
        }
    }while (!option.equals("q"));
}

// Department staff menu
private static void departmentMenu(Scanner scanner,String semester) {
    String option;
    do {
        System.out.println("\n************************************************");
        System.out.println("    Welcome to the TechEduPro - Online Registration System");
        System.out.println("                    Department Staff                    ");
        System.out.println("************************************************");
        System.out.println("1. Authorize Student into Section");
        System.out.println("2. Overflow Student into Section");
        System.out.println("3. Add Assistantship on System");
        System.out.println("4. Generate Class List");
        System.out.println("q. Quit");

        System.out.print("Type in your option: ");
        option = scanner.next();
        switch (option) {
            case "1":
                authorizeStudent(scanner);
                break;
            case "2":
                overflowStudent(scanner,semester);
                break;
            case "3":
                addAssistantship(scanner);
                break;
            case "4":
                generateClassList(scanner);
                break;
            case "q":
                System.out.println("Returning to the main menu...");
                break;
            default:
                System.out.println("Invalid option.");
        }
    }while (!option.equals("q"));
}

// Student menu

private static void studentMenu(Scanner scanner,String username,String semester) {
    String option;
    do {
        System.out.println("\n**************************************************");
        System.out.println("    Welcome to the TechEduPro - Online Registration System");
        System.out.println("                        Student                        ");
        System.out.println("**************************************************");
        System.out.println("1. Add a Section");
        System.out.println("2. Drop a Section");
        System.out.println("3. See Schedule for a Term");
        System.out.println("4. See Fee Detail");
        System.out.println("5. See Transcript");
        System.out.println("q. Quit");

        System.out.print("Type in your option: ");
        option = scanner.next();
        switch (option) {
            case "1":
                addSection(scanner, username,semester);
                break;
            case "2":
                dropSection(scanner, username);
                break;
            case "3":
                seeSchedule(scanner, username);
                break;
            case "4":
                seeFeeDetail(scanner, username,semester);
                break;
            case "5":
                seeTranscript(scanner, username);
                break;
            case "q":
                System.out.println("Returning to the main menu...");
                break;
                default:
                    System.out.println("Invalid option.");
            }
        } while (!option.equals("q"));
    }
}
```

3. Dependency-reduced-pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>my-java-app</artifactId>
  <version>1.0-SNAPSHOT</version>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.2.0</version>
        <configuration>
          <archive>
            <manifest>
              <addClasspath>true</addClasspath>
              <mainClass>com.example.myJavaApp</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>
      <plugin>
        <artifactId>maven-shade-plugin</artifactId>
        <version>3.2.4</version>
        <executions>
          <execution>
            <phase>package</phase>
            <goals>
              <goal>shade</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
  <properties>
    <maven.compiler.target>11</maven.compiler.target>
    <maven.compiler.source>11</maven.compiler.source>
  </properties>
</project>
```

4. pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example</groupId>
    <artifactId>my-java-app</artifactId>
    <version>1.0-SNAPSHOT</version>
    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <version>8.0.33</version>
        </dependency>
        <!-- Other dependencies can go here -->
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jar-plugin</artifactId>
                <version>3.2.0</version>
                <configuration>
                    <archive>
                        <manifest>
                            <addClasspath>true</addClasspath>
                            <mainClass>com.example.myJavaApp</mainClass>
                        </manifest>
                    </archive>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-shade-plugin</artifactId>
                <version>3.2.4</version>
                <executions>
                    <execution>
                        <phase>package</phase>
                        <goals>
                            <goal>shade</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</project>
```

5. docker-compose.yml

```
 version: '3.8'

services:
  mysql:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: user
      MYSQL_PASSWORD: userpassword
    ports:
      - "3306:3306"

  java-app:
    build:
      context: .
      dockerfile: Dockerfile
    environment:
      DB_URL: jdbc:mysql://mysql:3306/mydatabase
      DB_USER: user
      DB_PASSWORD: userpassword
    depends_on:
      - mysql
```

6. DockerFile

```
# Dockerfile
FROM openjdk:11-jre-slim

WORKDIR /app

# Copy the JAR file into the container
COPY target/my-java-app-1.0-SNAPSHOT.jar /app/myJavaApp.jar

# Command to run the JAR file
CMD ["java", "-jar", "myJavaApp.jar"]
```