

Mini Project Report

May 12, 2025

Name: Hitanshu Seth

Roll Number: IMT2023100

Course: EGC 301P - Operating Systems Lab

Course Management System

1. Introduction

The Course Management System is a client-server application developed in C to manage students, faculty, and courses within an educational institution. It supports role-based access for administrators, faculty, and students, providing functionalities such as course creation, enrollment, and user management, while ensuring data integrity through file locking.

2. Tech Stack and Implementation

- **Programming Language:**

Implemented in C, allowing for low-level control and efficient system resource use.

- **Socket Programming:**

Enables communication between client and server via TCP/IP.

- **File Handling:**

Data is stored in plain text files (users.txt, students.txt, etc.) and accessed using file locking to maintain consistency.

- **Multithreading:**

The server handles multiple clients concurrently using pthread.

3. Source Code Structure

3.1 Server Module (server.c)

:-Purpose: Acts as the backend, handling all business logic and file operations.

:-Key Functions:

- **Authenticate:** Authenticates user credentials and role.
- **AddStudent:** Adds a student to the system.
- **AddFaculty:** Adds a faculty member.
- **ToggleStudentStatus:** Activates or deactivates a student.
- **UpdateDetails:** Updates student or faculty names.
- **AddCourse:** Faculty creates a new course.
- **RemoveCourse:** Faculty removes a course they own.
- **ViewCourseEnrollments:** Displays enrolled students in a course.
- **EnrollCourse:** Student enrolls in a course.
- **UnenrollCourse:** Student unenrolls from a course.
- **ViewEnrolledCourses:** Student views enrolled courses.
- **ChangePassword:** Any user updates their password.

3.2 Client Module (client.c)

:-Purpose: Acts as the frontend interface that users interact with.

:-Key Functions: - Handles all server communication for student, faculty, and admin operations. - Accepts user input and displays results from the server.

3.3 Data Files

- **users.txt:** Stores login credentials and roles.
- **courses.txt:** Maintains course data.
- **enrollments.txt:** Links students to enrolled courses.

4. Features by Role

4.1 Admin

- Add new students and faculty.
- Activate/deactivate student accounts.
- Update student/faculty names.

4.2 Faculty

- Add new courses with seat limits.
- Remove existing courses.
- View enrollments for their courses.

4.3 Student

- Enroll in available courses.
 - Unenroll from courses.
 - View list of enrolled courses.
-

5. Conclusion

This Course Management System offers a simple yet powerful backend service to manage educational operations. It maintains data consistency using locking mechanisms and provides an interactive client for users across different roles. Its modular design allows for easy updates and scalability.

6. Screenshots with Examples

Below are representative screenshots showing key functionalities:

6.1 Server Setup

Initial setup showing the server awaiting client connections.

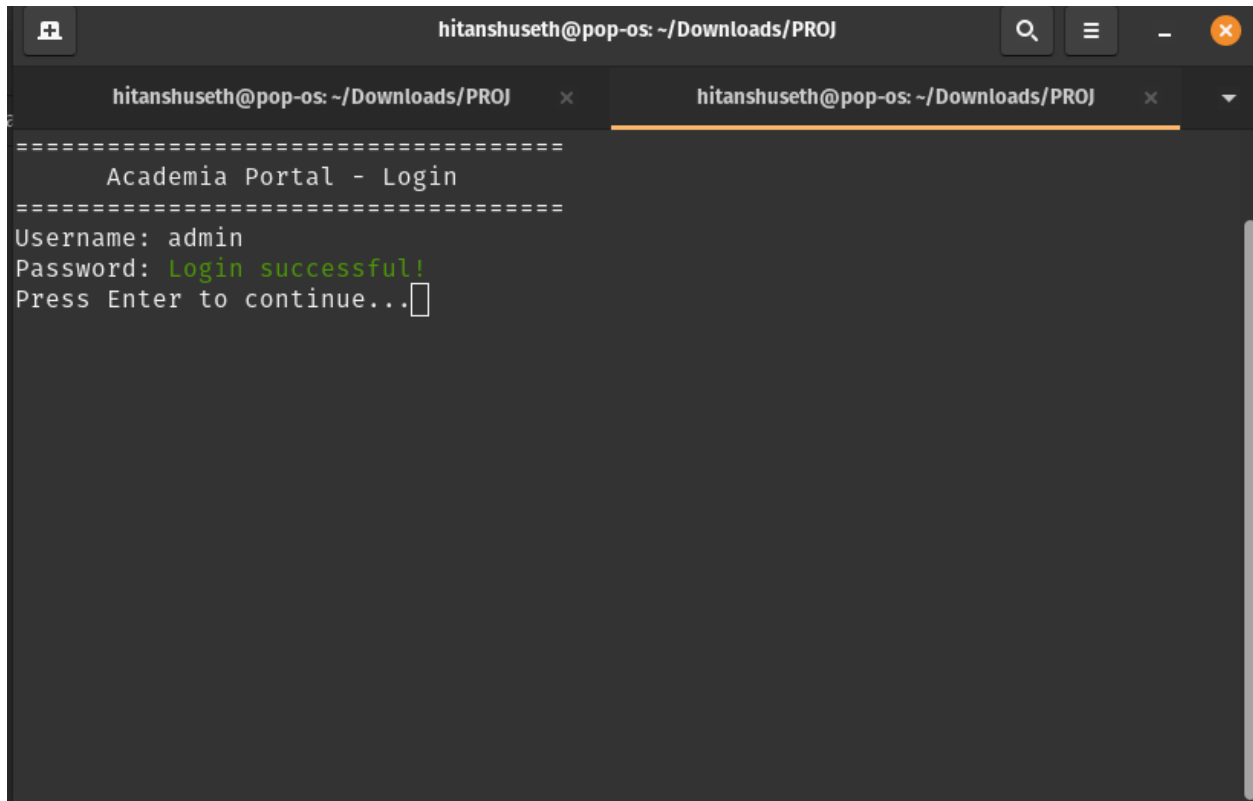
```

hitanshusheth@pop-os: ~/Downloads/PROJ
%d\n",
~~~~~
826 |             courses[i].code, courses[i].name,
~~~~~
827 |             courses[i].enrolledStudents, courses[i].totalSea
ts);
~~~~~
server.c:834:46: warning: '%s' directive writing up to 255 bytes into a region o
f size 254 [-Wformat-overflow=]
834 |             sprintf(line, "- %s (ID: %d)\n", student->us
ername, student->id);
~~~~~
server.c:834:29: note: 'sprintf' output between 12 and 277 bytes into a destinat
ion of size 256
834 |             sprintf(line, "- %s (ID: %d)\n", student->us
ername, student->id);
~~~~~
hitanshusheth@pop-os:~/Downloads/PROJ$ ./server
Academia Portal Server started on port 8080
Waiting for connections...

```

6.2 User Authentication

Login screen prompting credentials



```
hitanshuseth@pop-os: ~/Downloads/PROJ
Academia Portal - Login
Username: admin
Password: Login successful!
Press Enter to continue...
```

6.3 Client Login

Users log in and access functionalities based on their roles.

1)Admin

```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Admin Dashboard
=====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. View All Users
6. View All Courses
7. Logout
8. Exit

Enter your choice: 
```

2)Student

```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Student Dashboard
=====
1. Enroll to new Courses
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. View all available Courses
5. Change Password
6. Logout
7. Exit

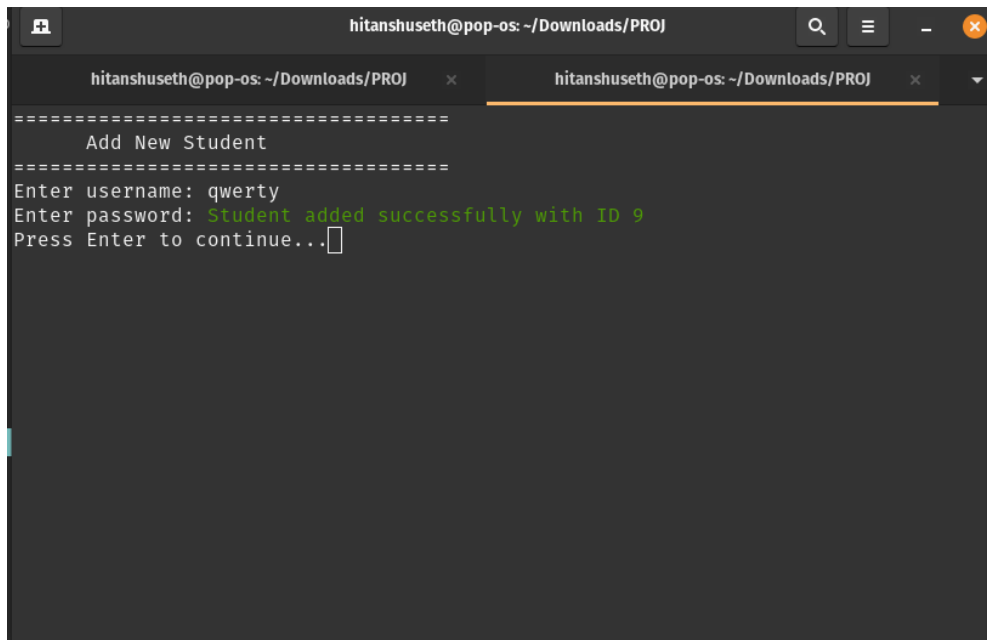
Enter your choice: 
```

3)Faculty

```
hitanshuseth@pop-os: ~/Downloads/PROJ
Faculty Dashboard
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. View your Courses
5. Change Password
6. Logout
7. Exit
Enter your choice: 
```

6.4 Admin - Add Student

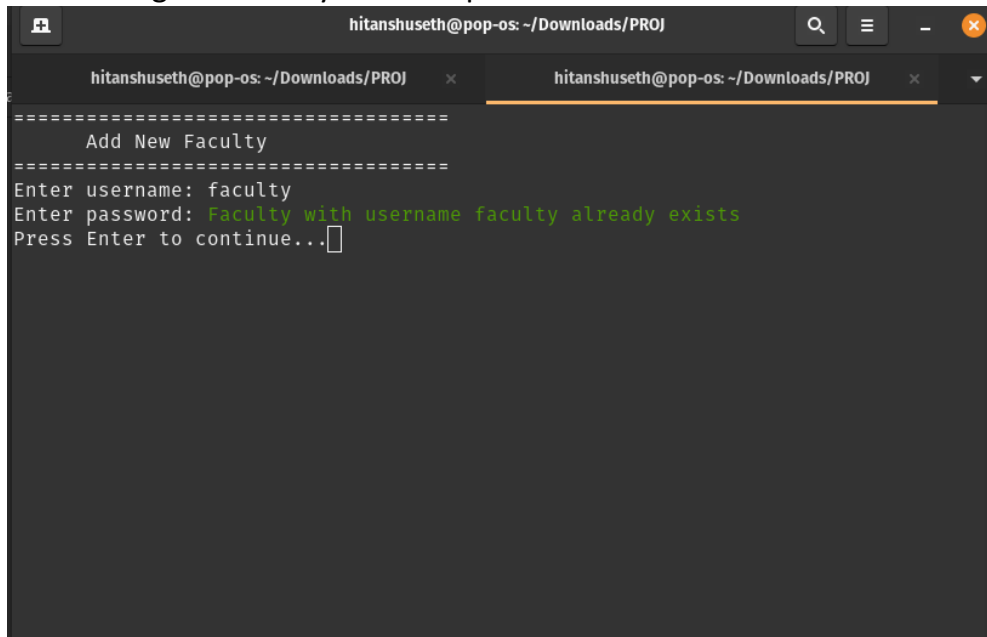
Admin enters student details for registration.



A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a program running with the following text: '=====
Add New Student
=====
Enter username: qwerty
Enter password: Student added successfully with ID 9
Press Enter to continue...'. The text 'Student added successfully with ID 9' is highlighted in green.

6.5 Admin - Add Faculty

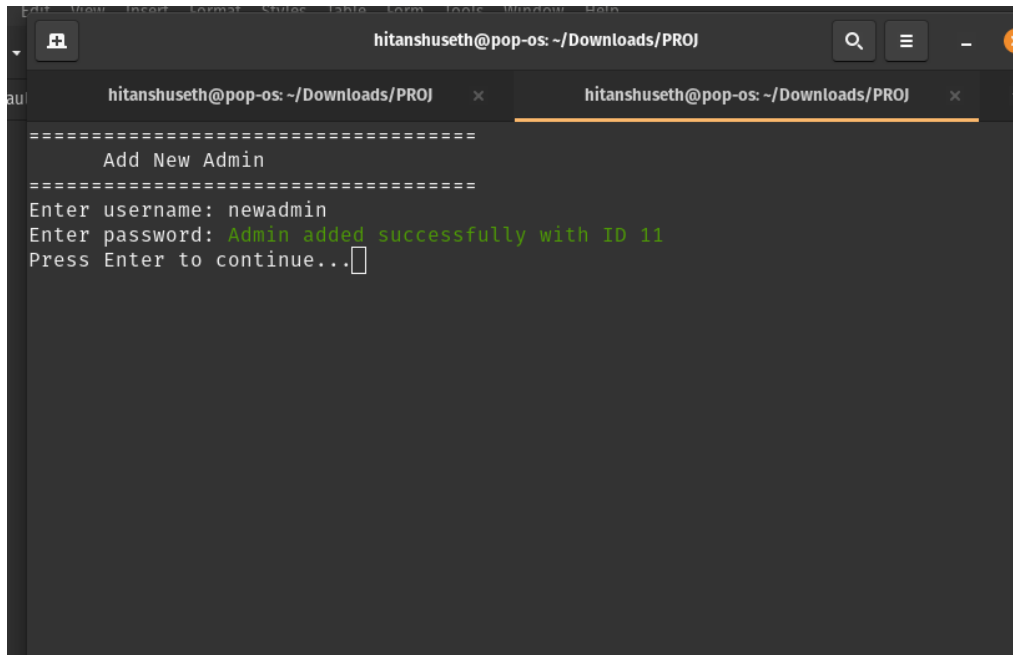
Admins register faculty with unique IDs.



A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a program running with the following text: '=====
Add New Faculty
=====
Enter username: faculty
Enter password: Faculty with username faculty already exists
Press Enter to continue...'. The text 'Faculty with username faculty already exists' is highlighted in green.

6.6 Admin - Add Admin

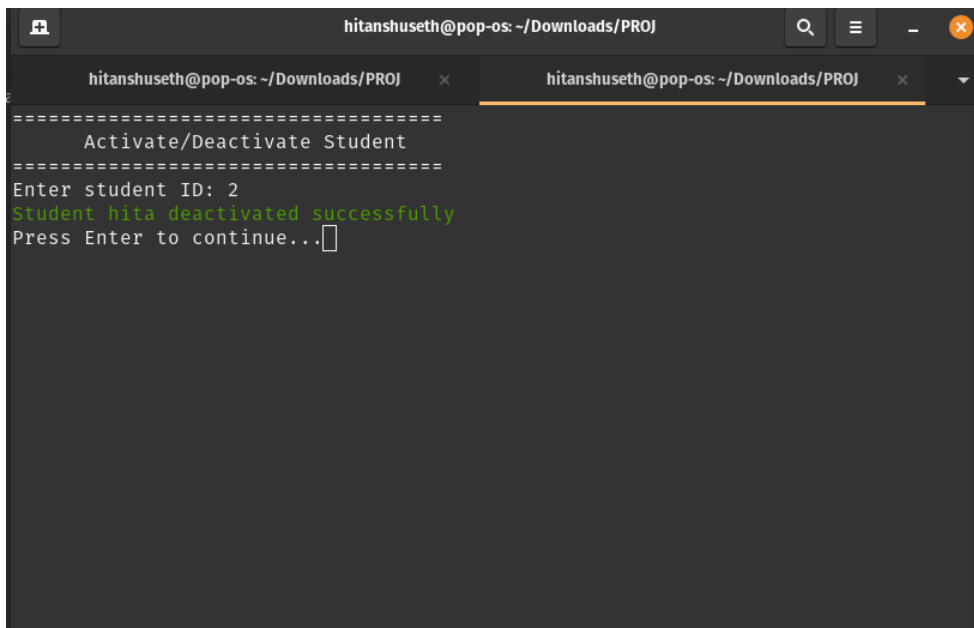
Admins register admin with unique IDs.



```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Add New Admin
=====
Enter username: newadmin
Enter password: Admin added successfully with ID 11
Press Enter to continue...
=====
```

6.7 Admin - Toggle Student Status

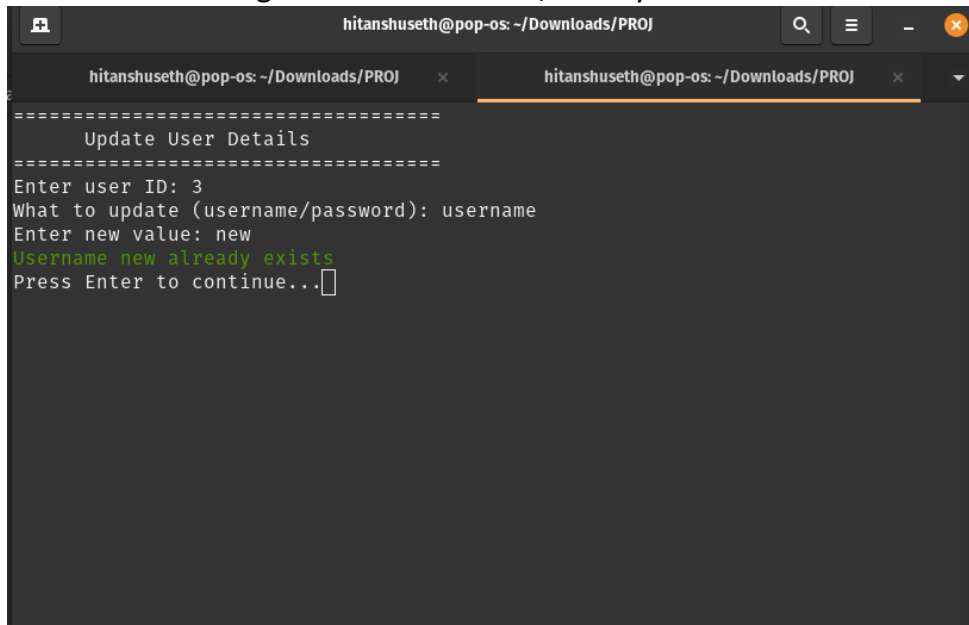
Admins activate or deactivate student access.



```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Activate/Deactivate Student
=====
Enter student ID: 2
Student hita deactivated successfully
Press Enter to continue...
=====
```

6.8 Admin - Update User Details

Used for correcting names of students/faculty.



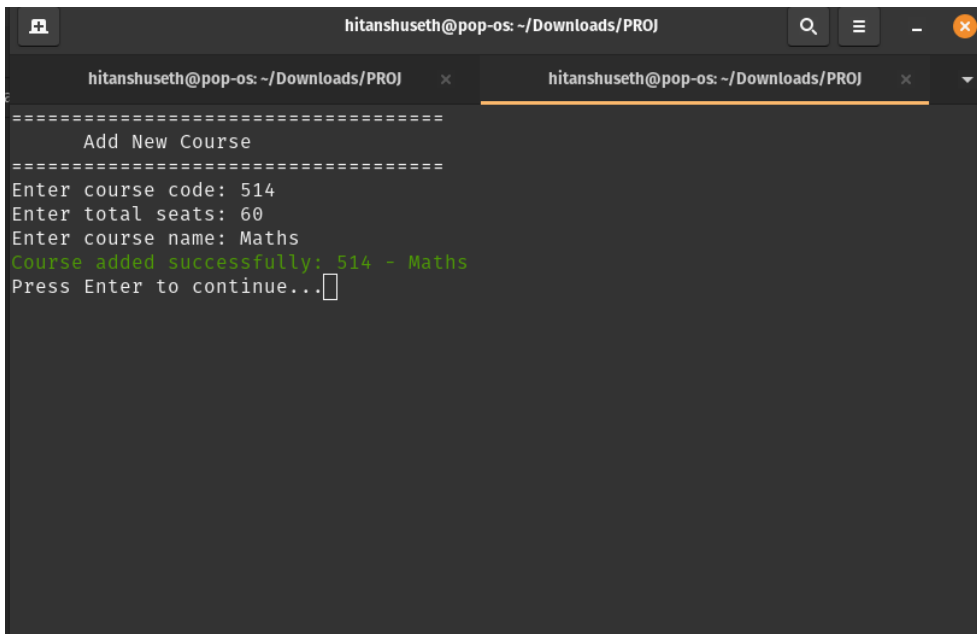
A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a menu with 'Update User Details' selected. The user enters ID 3 and chooses to update the username to 'new'. A green message 'Username new already exists' is displayed, followed by a prompt to press Enter to continue.

```
hitanshuseeth@pop-os: ~/Downloads/PROJ
=====
Update User Details
=====
Enter user ID: 3
What to update (username/password): username
Enter new value: new
Username new already exists
Press Enter to continue...

```

6.9 Faculty - Add Course

Faculty enters new course details including seat count.



A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a menu with 'Add New Course' selected. The user enters course code 514, total seats 60, and course name 'Maths'. A green message 'Course added successfully: 514 - Maths' is displayed, followed by a prompt to press Enter to continue.

```
hitanshuseeth@pop-os: ~/Downloads/PROJ
=====
Add New Course
=====
Enter course code: 514
Enter total seats: 60
Enter course name: Maths
Course added successfully: 514 - Maths
Press Enter to continue...

```

6.10 Faculty - Remove Course

Faculty deletes a course they created.

```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Remove Course
=====
Your courses:
Code: 514, Name: Maths, Enrollment: 0/60

Enter course code to remove: 514
Course 514 removed successfully
Press Enter to continue...
```

6.11 Faculty - View Enrollments

Displays all students enrolled in the selected course.

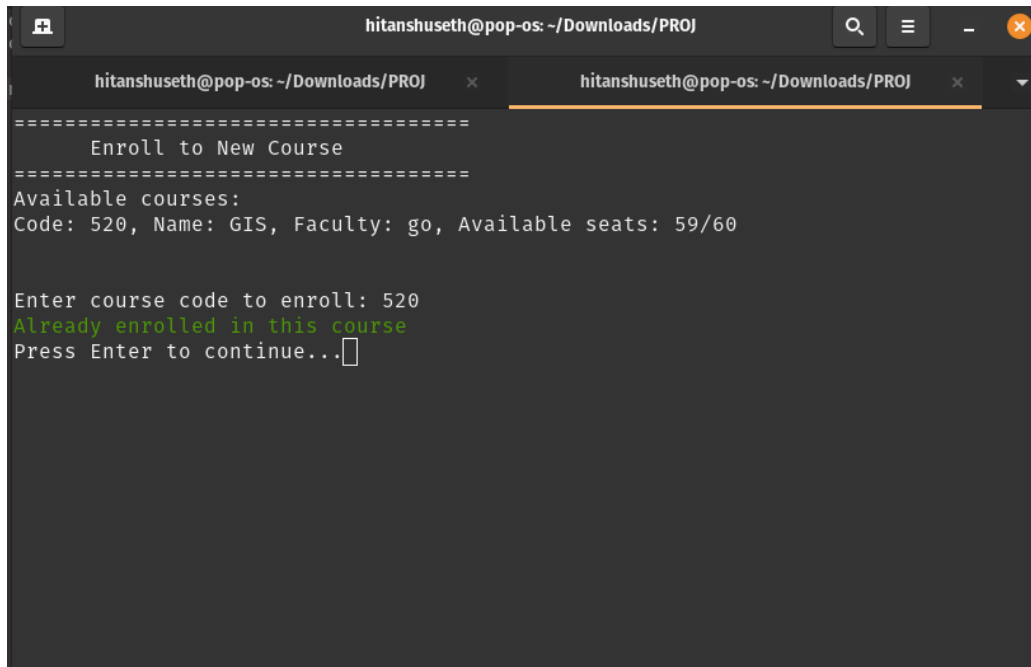
```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Course Enrollments
=====
Course enrollments:

Course: 520 - GIS
Enrolled students: 1/60
- qwerty (ID: 9)

Press Enter to continue...
```

6.12 Student - Enroll in Course

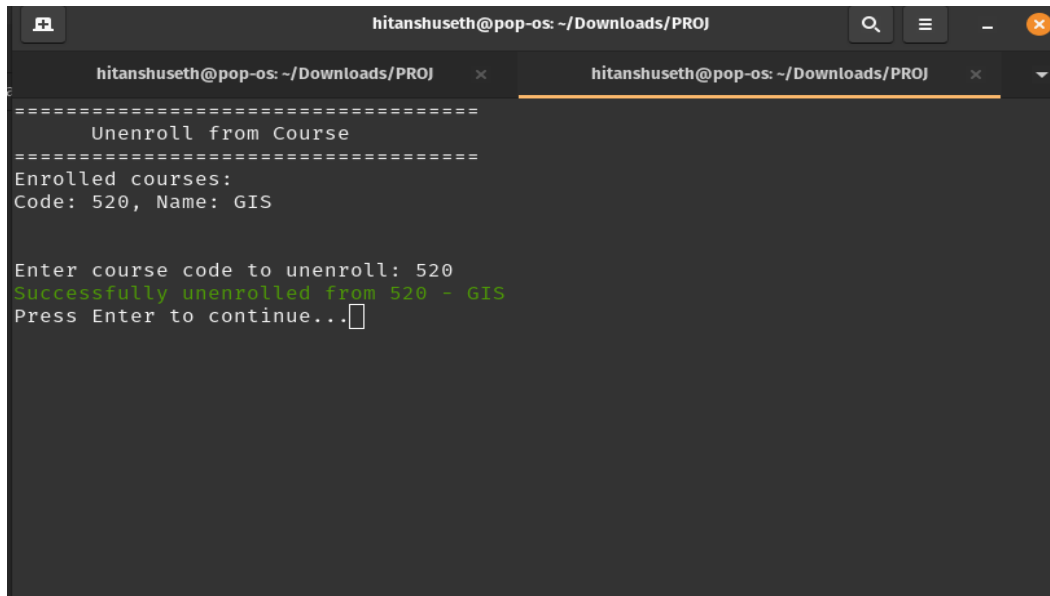
Students view and enroll in available courses.

A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a menu with 'Enroll to New Course' selected. Below the menu, it lists 'Available courses:' with details 'Code: 520, Name: GIS, Faculty: go, Available seats: 59/60'. The user enters '520' for the course code, and the terminal displays 'Already enrolled in this course' in green text. The prompt 'Press Enter to continue...' is shown with a cursor.

```
hitanshuseeth@pop-os: ~/Downloads/PROJ
=====
Enroll to New Course
=====
Available courses:
Code: 520, Name: GIS, Faculty: go, Available seats: 59/60

Enter course code to enroll: 520
Already enrolled in this course
Press Enter to continue... 
```

6.13 Student - Unenroll from Course

A terminal window titled 'hitanshuseeth@pop-os: ~/Downloads/PROJ' with two tabs. The active tab shows a menu with 'Unenroll from Course' selected. Below the menu, it lists 'Enrolled courses:' with details 'Code: 520, Name: GIS'. The user enters '520' for the course code, and the terminal displays 'Successfully unenrolled from 520 - GIS' in green text. The prompt 'Press Enter to continue...' is shown with a cursor.

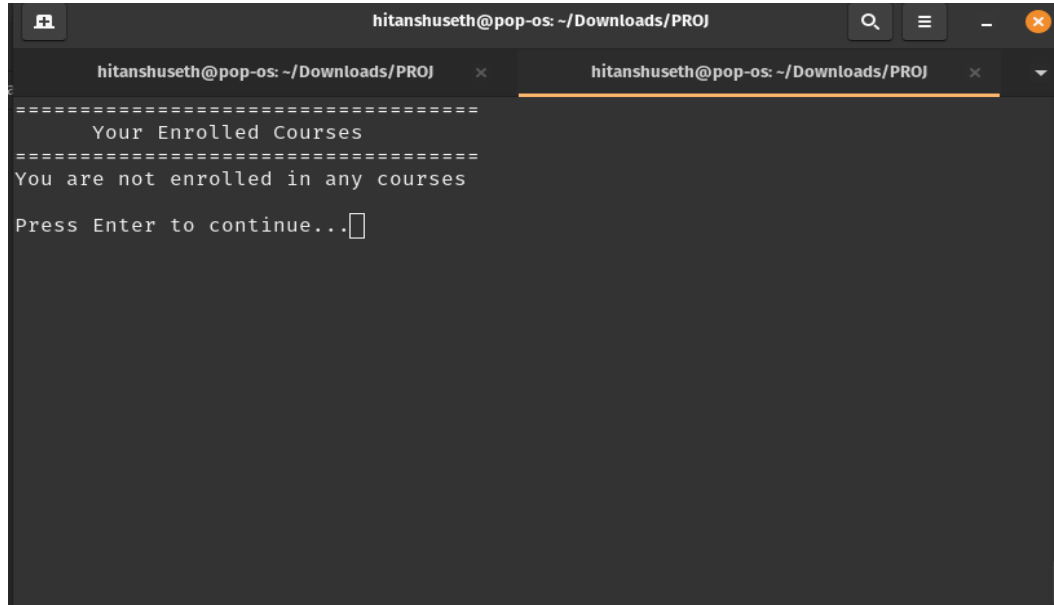
```
hitanshuseeth@pop-os: ~/Downloads/PROJ
=====
Unenroll from Course
=====
Enrolled courses:
Code: 520, Name: GIS

Enter course code to unenroll: 520
Successfully unenrolled from 520 - GIS
Press Enter to continue... 
```

Students can drop a course they no longer wish to take.

6.14 Student - View Enrolled Courses

Shows all courses the student is currently enrolled in.



```
hitanshuseth@pop-os: ~/Downloads/PROJ
=====
Your Enrolled Courses
=====
You are not enrolled in any courses
Press Enter to continue...
```