Academia Portal-Documentation

Operating Systems Lab: Mini Project

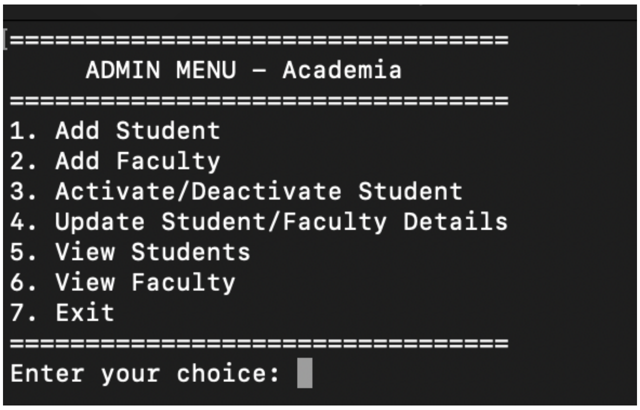
Project Overview:

- This project is about creating an easy-to-use terminal-based portal for course registration. It will help manage information related to students, teachers, and courses. The system will have three types of users: Admin, Student, and Faculty. Each user will have different features based on their role, like adding, courses, registering for courses, or managing course details.

Functionalities

Admin:

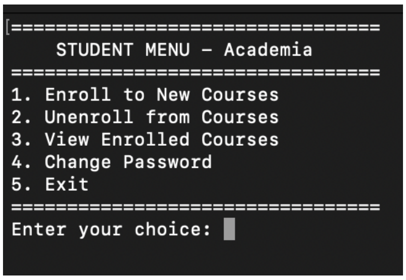
- Add a new student
- Add a new faculty member
- Activate a student account or Deactivate a student account
- Update student or faculty details
- View student or faculty information
- Exit the portal



```
[=================================
     ADMIN MENU - Academia
=================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
=================================
Enter your choice: 
```

Student:

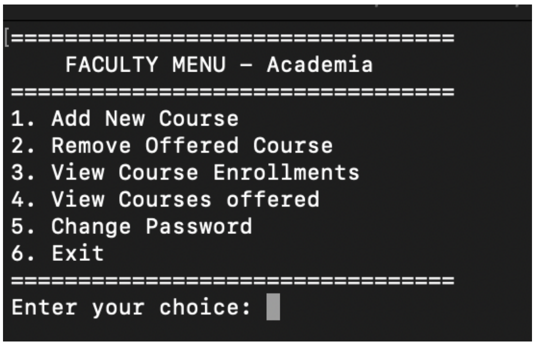
- Enroll in a new course
- Unenroll from a course
- View list of enrolled courses
- Change account password
- Exit the portal



```
[================================
    STUDENT MENU - Academia
================================
1. Enroll to New Courses
2. Unenroll from Courses
3. View Enrolled Courses
4. Change Password
5. Exit
================================
Enter your choice: 
```

Faculty:

- Add new courses to offer
- Remove courses they are offering
- View list of enrolled students
- Change account password
- View Courses offered by that faculty
- Update course information
- Logout and exit the portal



```
=================================
    FACULTY MENU - Academia
=================================
1. Add New Course
2. Remove Offered Course
3. View Course Enrollments
4. View Courses offered
5. Change Password
6. Exit
=================================
Enter your choice: 
```

Assumption:

Each course will have a limited number of seats, and students can only enroll if seats are available at the time of registration.

Mechanisms Used

- **Programming Language:** The entire application is developed using the C programming language.
- **Client-Server Architecture:** Socket programming is used to establish communication between the client and server, enabling interactions between different users and the system.
- **Persistent Storage:** Student, faculty, and course data are stored in files using file handling techniques to maintain information even after the program ends.
- **Data Safety with File Locking:** To prevent data corruption during simultaneous access, file locking is implemented. Read locks are applied when viewing course details. Write locks are used during enrollment and unenrollment.
- **Concurrent User Handling:** The server supports multiple users at the same time by handling concurrent client requests efficiently.
- **Secure Access:** Password protection is implemented to ensure that users (students, faculty, admin) can securely log in and access their features.
- And additional implementation, admin can’t add 2 same usernames for students and professors both. Also faculty can’t add two courses with same names.
- Automatically flushes for each new command
- Default pass word for every is “admin123”

Technologies Used:

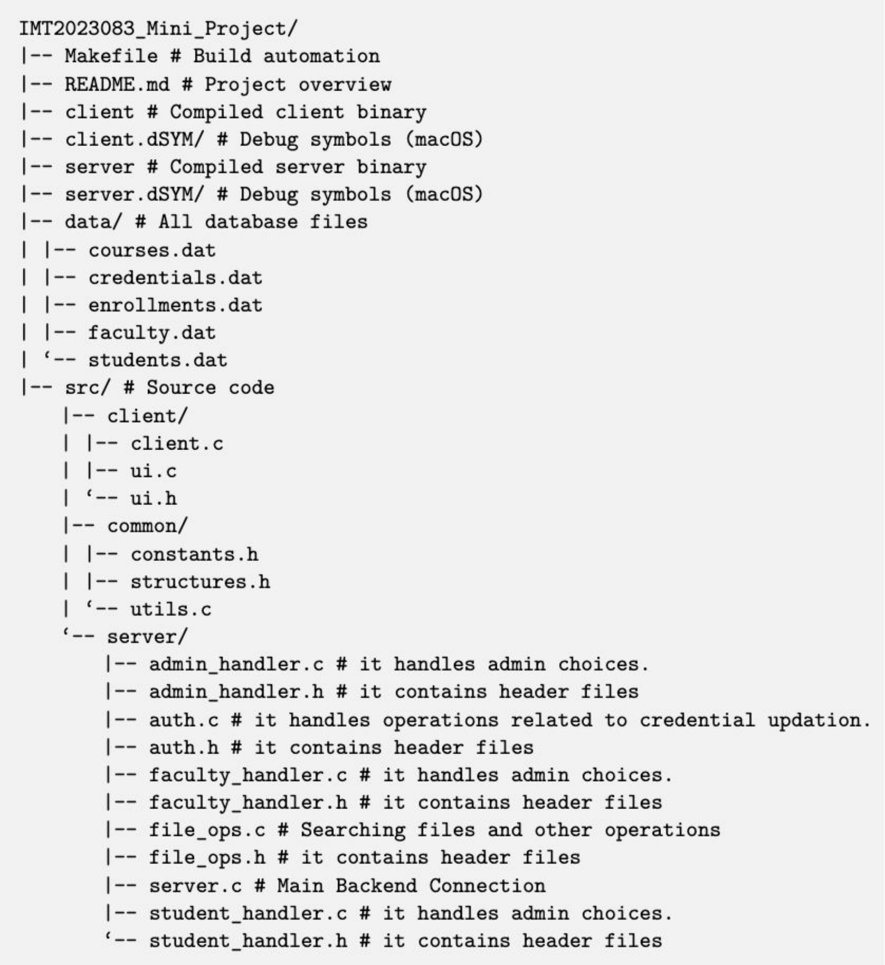
- Language: C
- Communication: TCP/IP Sockets

- Concurrency: POSIX Threads (pthreads)
- Locking Mechanism: System V file locking using flock
- System Calls: Includes read(), write(), socket(), bind(), listen(), accept()

RUN LOCALLY:

1. Execute the command “make” to build the application from the folder.
2. Run the server component by executing “`./server`”.
3. Open a new terminal window and run the client component using “`./client`”. The application will then be ready for use.

File Structure:

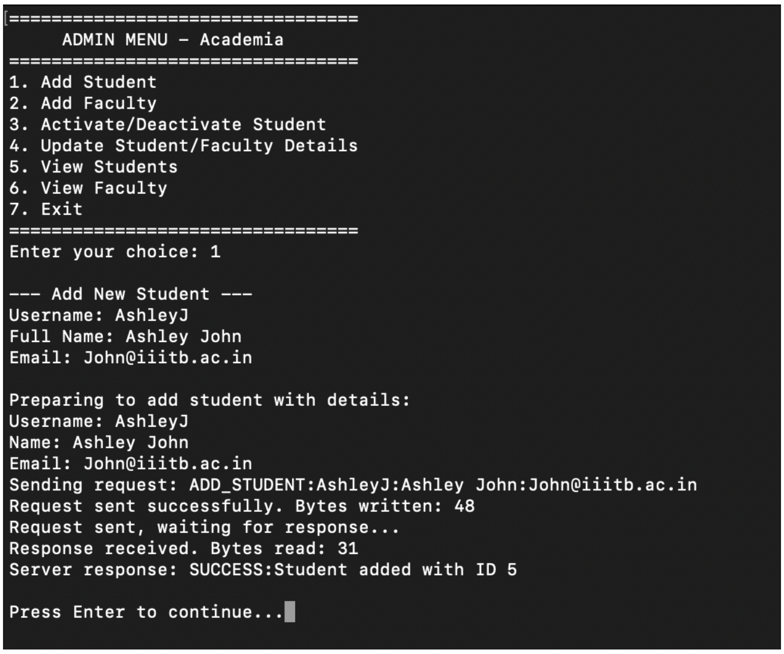


```
IMT2023083_Mini_Project/
|-- Makefile # Build automation
|-- README.md # Project overview
|-- client # Compiled client binary
|-- client.dSYM/ # Debug symbols (macOS)
|-- server # Compiled server binary
|-- server.dSYM/ # Debug symbols (macOS)
|-- data/ # All database files
| |-- courses.dat
| |-- credentials.dat
| |-- enrollments.dat
| |-- faculty.dat
| ‘-- students.dat
|-- src/ # Source code
    |-- client/
    | |-- client.c
    | |-- ui.c
    | ‘-- ui.h
    |-- common/
    | |-- constants.h
    | |-- structures.h
    | ‘-- utils.c
    ‘-- server/
        |-- admin_handler.c # it handles admin choices.
        |-- admin_handler.h # it contains header files
        |-- auth.c # it handles operations related to credential updation.
        |-- auth.h # it contains header files
        |-- faculty_handler.c # it handles admin choices.
        |-- faculty_handler.h # it contains header files
        |-- file_ops.c # Searching files and other operations
        |-- file_ops.h # it contains header files
        |-- server.c # Main Backend Connection
        |-- student_handler.c # it handles admin choices.
        ‘-- student_handler.h # it contains header files
```

Screenshots:

Admin Menu Options

1. Add student



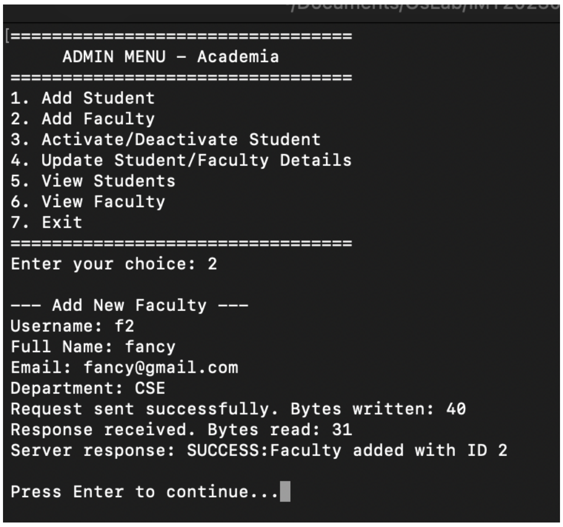
```
[================================
     ADMIN MENU - Academia
================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
================================
Enter your choice: 1

--- Add New Student ---
Username: AshleyJ
Full Name: Ashley John
Email: John@iiitb.ac.in

Preparing to add student with details:
Username: AshleyJ
Name: Ashley John
Email: John@iiitb.ac.in
Sending request: ADD_STUDENT:AshleyJ:Ashley John:John@iiitb.ac.in
Request sent successfully. Bytes written: 48
Request sent, waiting for response...
Response received. Bytes read: 31
Server response: SUCCESS:Student added with ID 5

Press Enter to continue...
```

2.Add faculty

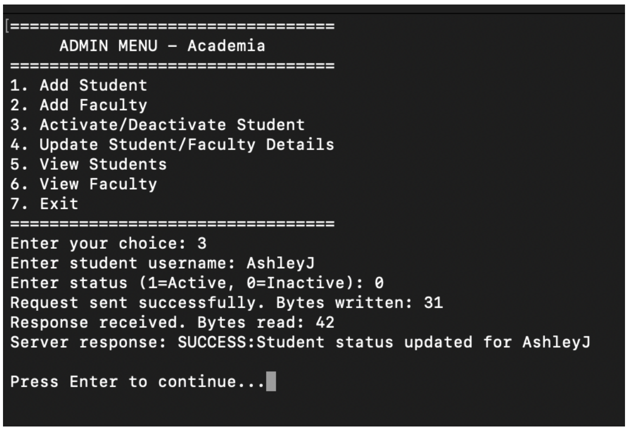


```
[================================
     ADMIN MENU - Academia
================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
================================
Enter your choice: 2

--- Add New Faculty ---
Username: f2
Full Name: fancy
Email: fancy@gmail.com
Department: CSE
Request sent successfully. Bytes written: 40
Response received. Bytes read: 31
Server response: SUCCESS:Faculty added with ID 2

Press Enter to continue...
```

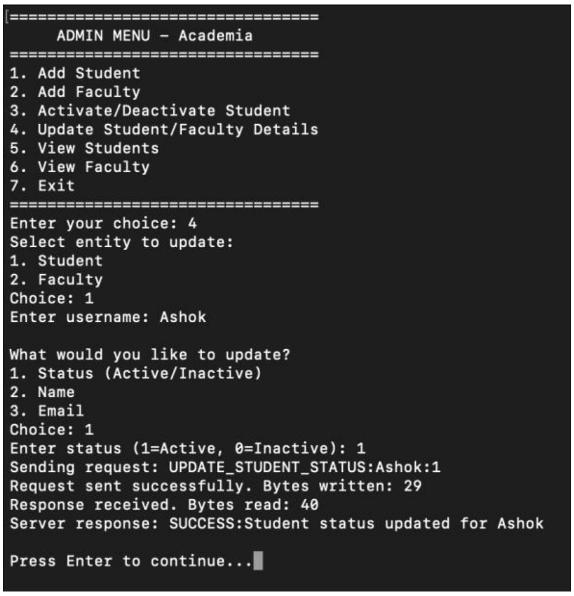
3. Activate/Deactivate Student



```
[================================
     ADMIN MENU - Academia
================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
================================
Enter your choice: 3
Enter student username: AshleyJ
Enter status (1=Active, 0=Inactive): 0
Request sent successfully. Bytes written: 31
Response received. Bytes read: 42
Server response: SUCCESS:Student status updated for AshleyJ

Press Enter to continue...
```

4. Modify Student/Faculty Details

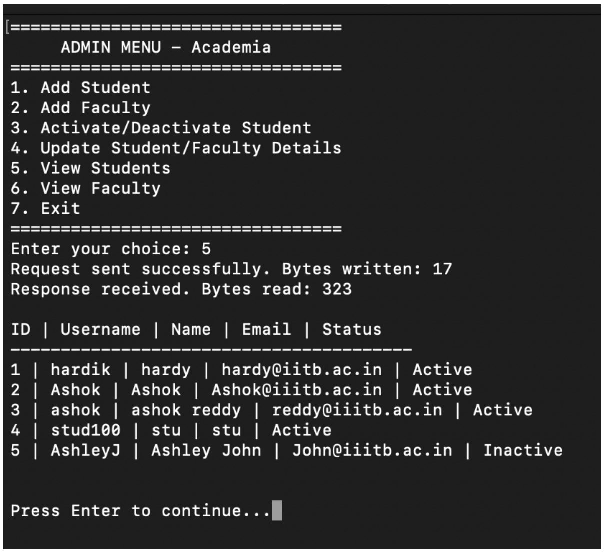


```
================================
     ADMIN MENU - Academia
================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
================================
Enter your choice: 4
Select entity to update:
1. Student
2. Faculty
Choice: 1
Enter username: Ashok

What would you like to update?
1. Status (Active/Inactive)
2. Name
3. Email
Choice: 1
Enter status (1=Active, 0=Inactive): 1
Sending request: UPDATE_STUDENT_STATUS:Ashok:1
Request sent successfully. Bytes written: 29
Response received. Bytes read: 40
Server response: SUCCESS:Student status updated for Ashok

Press Enter to continue...
```

5. View Student Details



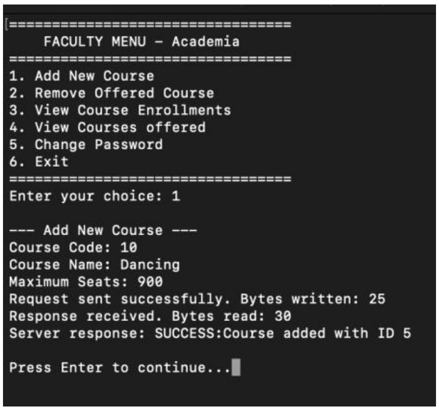
```
=================================
     ADMIN MENU - Academia
=================================
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty Details
5. View Students
6. View Faculty
7. Exit
=================================
Enter your choice: 5
Request sent successfully. Bytes written: 17
Response received. Bytes read: 323

ID | Username | Name | Email | Status
----------------------------------------
1 | hardik | hardy | hardy@iitb.ac.in | Active
2 | Ashok | Ashok | Ashok@iiitb.ac.in | Active
3 | ashok | ashok reddy | reddy@iiitb.ac.in | Active
4 | stud100 | stu | stu | Active
5 | AshleyJ | Ashley John | John@iiitb.ac.in | Inactive


Press Enter to continue...
```

Faculty Menu Options:

1. Add course

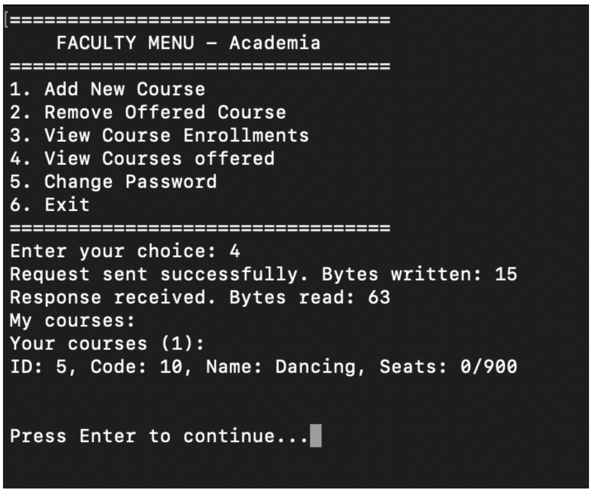


```
================================
    FACULTY MENU - Academia
================================
1. Add New Course
2. Remove Offered Course
3. View Course Enrollments
4. View Courses offered
5. Change Password
6. Exit
================================
Enter your choice: 1

--- Add New Course ---
Course Code: 10
Course Name: Dancing
Maximum Seats: 900
Request sent successfully. Bytes written: 25
Response received. Bytes read: 30
Server response: SUCCESS:Course added with ID 5

Press Enter to continue...
```

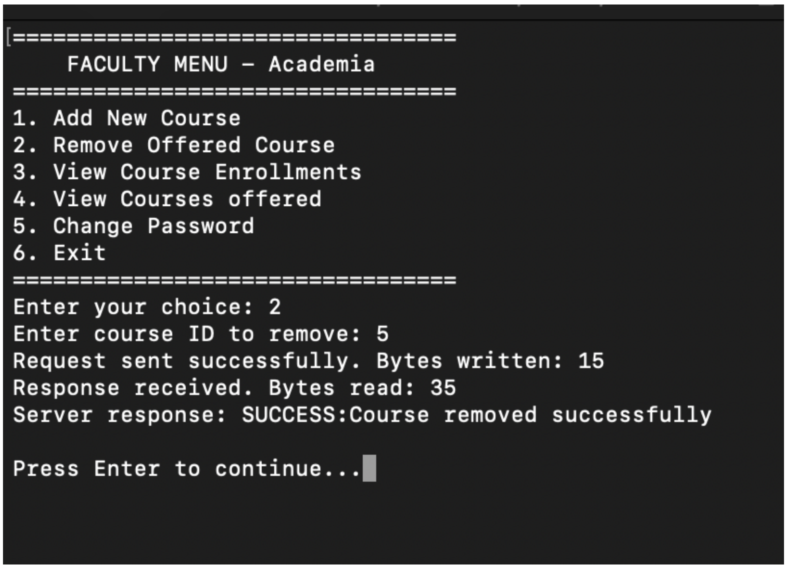
2. View Course



```
[================================
    FACULTY MENU - Academia
================================
1. Add New Course
2. Remove Offered Course
3. View Course Enrollments
4. View Courses offered
5. Change Password
6. Exit
================================
Enter your choice: 4
Request sent successfully. Bytes written: 15
Response received. Bytes read: 63
My courses:
Your courses (1):
ID: 5, Code: 10, Name: Dancing, Seats: 0/900


Press Enter to continue...
```

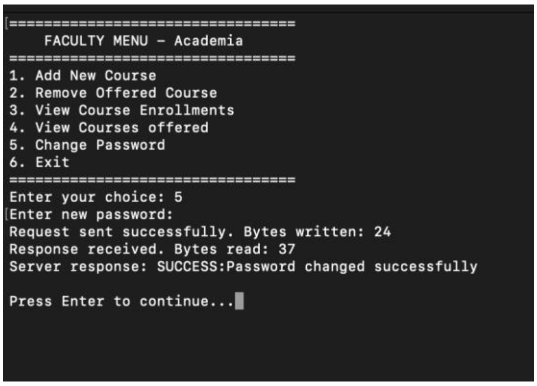
3. Remove course



```
[=================================
    FACULTY MENU - Academia
=================================
1. Add New Course
2. Remove Offered Course
3. View Course Enrollments
4. View Courses offered
5. Change Password
6. Exit
=================================
Enter your choice: 2
Enter course ID to remove: 5
Request sent successfully. Bytes written: 15
Response received. Bytes read: 35
Server response: SUCCESS:Course removed successfully

Press Enter to continue...
```

4. change password

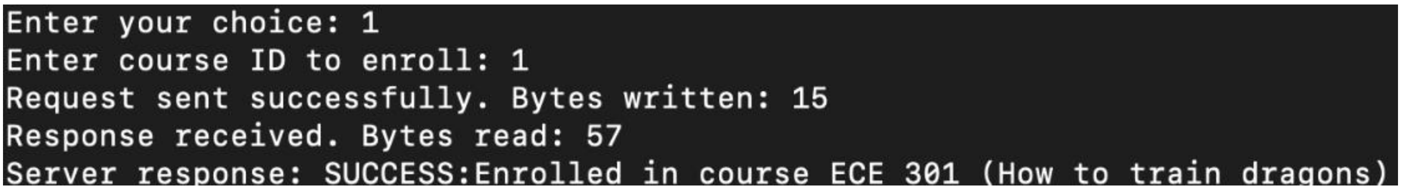


```
[================================
    FACULTY MENU - Academia
================================
1. Add New Course
2. Remove Offered Course
3. View Course Enrollments
4. View Courses offered
5. Change Password
6. Exit
================================
Enter your choice: 5
[Enter new password:
Request sent successfully. Bytes written: 24
Response received. Bytes read: 37
Server response: SUCCESS:Password changed successfully

Press Enter to continue...
```

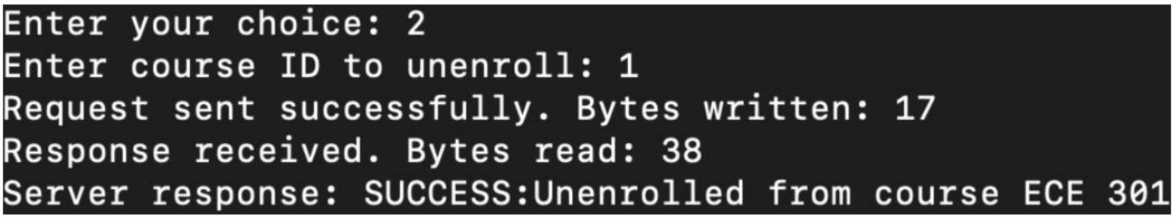
Student menu options:

1. Enroll new courses



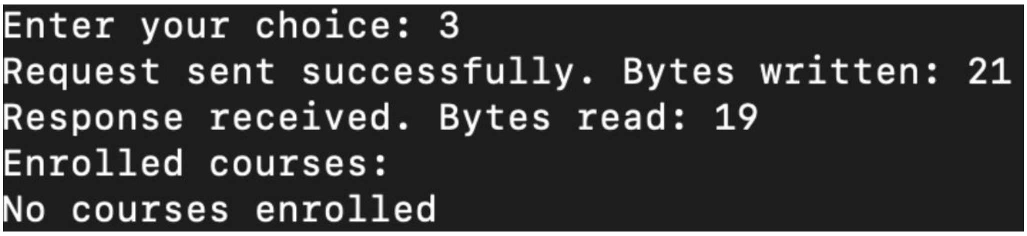
```
Enter your choice: 1
Enter course ID to enroll: 1
Request sent successfully. Bytes written: 15
Response received. Bytes read: 57
Server response: SUCCESS:Enrolled in course ECE 301 (How to train dragons)
```

2. unenroll course



```
Enter your choice: 2
Enter course ID to unenroll: 1
Request sent successfully. Bytes written: 17
Response received. Bytes read: 38
Server response: SUCCESS:Unenrolled from course ECE 301
```

3. view enrolled course



```
Enter your choice: 3
Request sent successfully. Bytes written: 21
Response received. Bytes read: 19
Enrolled courses:
No courses enrolled
```

Troubleshooting:

4.1 Common Issues

- Connection Failed: Ensure server is running and on correct port.

- Invalid Credentials: Check username/password; ensure account is active.

- Account Inactive: Contact admin to activate.

- Menu Unresponsive: Press Enter or restart client.

- Data Issues: Delete corrupted data files and restart server.