

# Client-Server Based Course Registration System (Academia portal)

## Problem Statement

To build a client-server based academic course registration portal, where users can act as Admin, Faculty, or Student. The server handles authentication, user management, course management, and enrollment operations, while clients interact over a network to perform various tasks based on their roles.

## Implementation Details

### Sockets Usage

Used to establish connections between the client and server. The server initializes a socket, binds to a port (8080), listens and accepts multiple client connections. Each client is handled in a separate thread after connection. Client communicates by sending and receiving messages using this socket connection.

### Locks (pthread\_mutex\_t)

Usage Three mutexes are used: - Student Mutex: To ensure mutual exclusion while accessing or modifying students.dat. - Faculty Mutex: To protect faculty.dat during read/write operations. - Course Mutex: Used to synchronize access when enrolling, unenrolling, or updating course data. These locks ensure data consistency during concurrent client access.

### Semaphores

Usage Included through but not actively used in the current implementation. Possibly reserved for future use (e.g., access control or managing concurrent limits).

### Threading

Each client is assigned a new thread upon connection. Threads are detached so they clean up automatically when done. This allows handling multiple concurrent clients efficiently.

### File Operations

Data is stored and manipulated using:

- admin.dat — Admin credentials.

- students.dat — Student information.
- faculty.dat — Faculty and offered course info.

Files are accessed using system calls (open, read, write, lseek). File-level locking is simulated using mutexes for consistency.

### Signal Handling

A signal handler is set up to clean up mutex resources on server shutdown (SIGINT). Ensures graceful termination and avoids resource leakage.

### User Roles

- Admin: Add/Update Students and Faculty, Activate/Deactivate students.
- Faculty: Add/Remove offered courses, View enrollments.
- Student: Enroll or unenroll in courses, View enrolled courses, Change password.

### Role-Based Functionalities

#### Admin:

- Add New Student
- Add Faculty
- Activate Student
- Deactivate Student
- Update student / Faculty details
- Exit

#### Student:

- Enroll New Course
- Unenroll Courses
- View Enrolled Courses
- Password Change
- Exit

#### Faculty:

- Add New Courses
- Remove Offered Courses
- View Enrollments
- Password Change
- Exit

Structures Used Three main structures are used in the system:

### 1. Student:

- ID
- Username
- Password
- Active Status
- List of Enrolled Courses
- Course Count

### 2. Faculty:

- ID
- Username
- Password
- List of Offered Courses
- Seats per Course
- Course Count

### 3. Admin:

- Username
- Password

## Server Code Functionality

The server initializes by setting up a socket, binds to a port, and listens for incoming client connections. When a client connects, a new thread is created to handle the client independently.

Each client goes through a login interface to select their role (Admin, Student, or Faculty), provide credentials, and get authenticated. Post-authentication, each role is presented with a unique menu and capabilities.

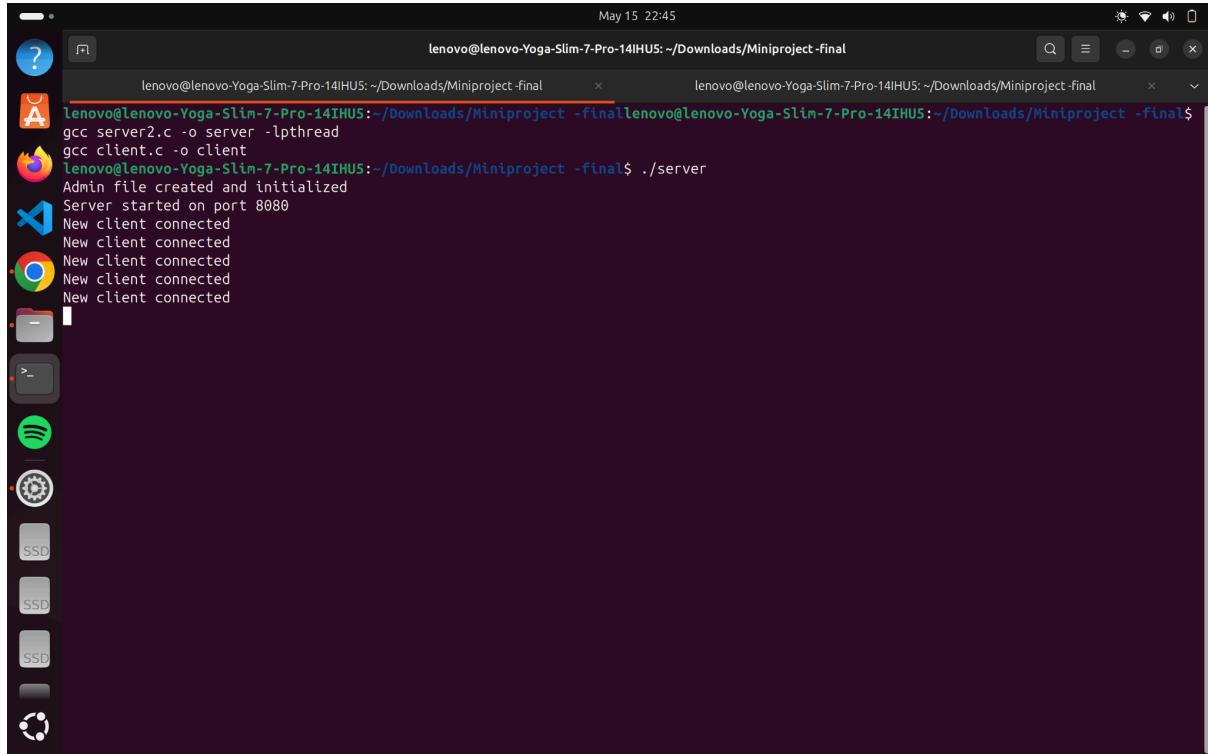
The server handles all core logic like reading/writing to files, validating credentials, updating student/faculty details, enrolling and unenrolling in courses, and changing passwords. Mutex locks ensure that no two threads corrupt shared data.

## Client Code Functionality

The client connects to the server via sockets and interacts through text-based prompts. It displays menus for each role and takes user input which it sends to the server. It also displays messages and feedback received from the server such as successful operations, errors, or prompts.

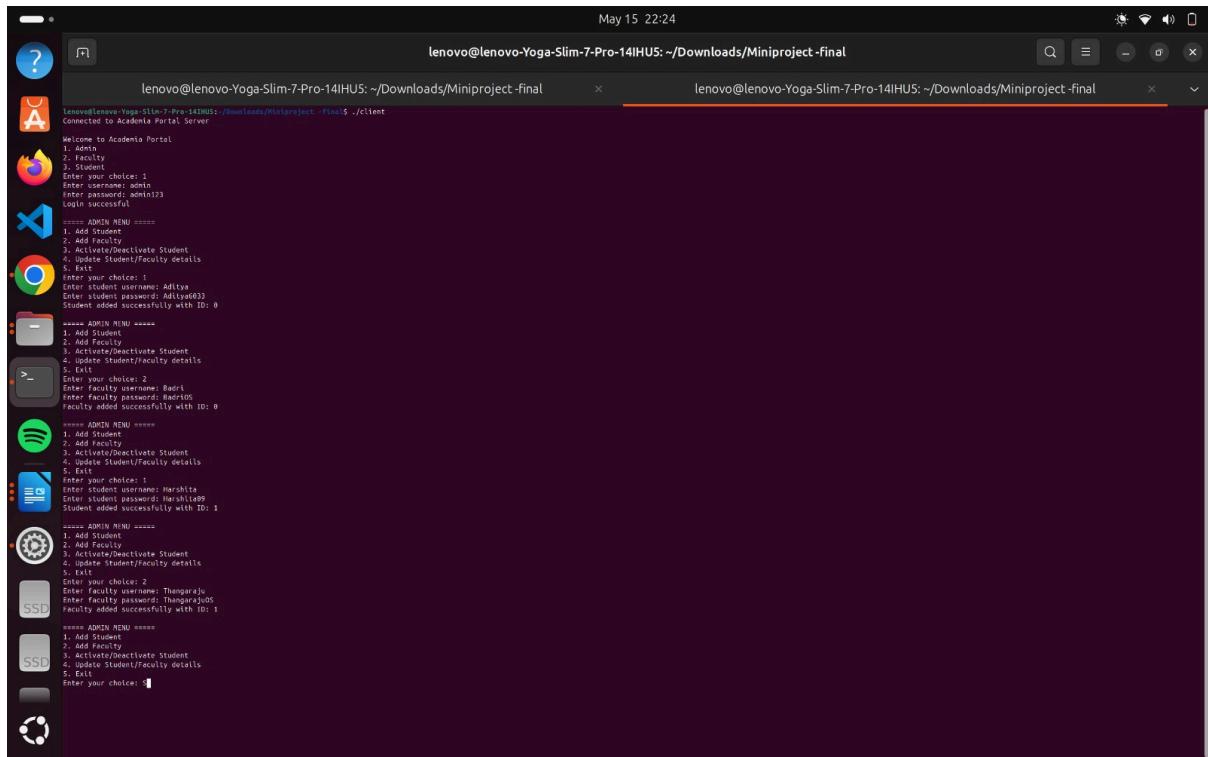
## Output screenshots

### 1.. Server terminal showing connections.



```
lenovo@lenovo-Yoga-Slim-7-Pro-14IHUS:~/Downloads/Miniproject-final$ gcc server2.c -o server -lpthread
lenovo@lenovo-Yoga-Slim-7-Pro-14IHUS:~/Downloads/Miniproject-final$ ./server
Admin file created and initialized
Server started on port 8080
New client connected
```

### 2.Admin menu interface.



```
lenovo@lenovo-Yoga-Slim-7-Pro-14IHUS:~/Downloads/Miniproject-final$ ./client
Connected to Academia Portal Server
Welcome to Academia Portal
1. Admin
2. Faculty
3. Student
Enter your choice: 3
Enter username: admin
Enter password: admin123
Login successful
===== ADMIN MENU =====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 1
Enter student username: Aditya
Enter student password: Aditya@03
Student added successfully with ID: 0
===== ADMIN MENU =====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 2
Enter faculty username: Baari
Enter faculty password: Baari@03
Faculty added successfully with ID: 0
===== ADMIN MENU =====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 3
Enter student username: Harshita
Enter student password: Harshita@03
Student added successfully with ID: 1
===== ADMIN MENU =====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 2
Enter faculty username: Thengaraju
Enter faculty password: Thengaraju@03
Faculty added successfully with ID: 1
===== ADMIN MENU =====
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 5
```

### 3.Faculty adding a course.

```
May 15 22:25
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5: ~/Downloads/Miniproject -final
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5:~/Downloads/Miniproject -final$ ./client
Connected to Academia Portal Server

Welcome to Academia Portal
1. Admin
2. Faculty
3. Student
Enter your choice: 2
Enter username: Badri
Enter password: BadriOS
Login successful

===== FACULTY MENU =====
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice: 1
Enter course name: OS
Enter number of seats: 50
Course added successfully

===== FACULTY MENU =====
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice: 1
```

#### 4. Student enrolling in a course

```
May 15 22:26
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5: ~/Downloads/Miniproject -final
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5:~/Downloads/Miniproject -final$ ./client
Connected to Academia Portal Server

4. Password Change
5. Exit
Enter your choice: ^Z
[2]+  Stopped                  ./client
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5:~/Downloads/Miniproject -final$ ./client

Welcome to Academia Portal
1. Admin
2. Faculty
3. Student
Enter your choice: 3
Enter username: Aditya
Enter password: Aditya6033
Login successful

===== STUDENT MENU =====
1. Enroll to new Courses
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. Password Change
5. Exit
Enter your choice: 1
Available Courses:
- OS (Available seats: 50)
Enter course name to enroll: OS
Successfully enrolled in course

===== STUDENT MENU =====
1. Enroll to new Courses
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. Password Change
5. Exit
Enter your choice:
```

#### 5. Student viewing enrolled courses.

```
May 15 22:27
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5: ~/Downloads/Miniproject -final
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5:~/Downloads/Miniproject -final$ ./client
Connected to Academia Portal Server

Welcome to Academia Portal
1. Admin
2. Faculty
3. Student
Enter your choice: 3
Enter username: Aditya
Enter password: Aditya6033
Login successful

===== STUDENT MENU =====
1. Enroll to new Courses
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. Password Change
5. Exit
Enter your choice: 3

==== Your Enrolled Courses ====
Total courses enrolled: 1

1. OS

===== STUDENT MENU =====
1. Enroll to new Courses
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. Password Change
5. Exit
Enter your choice:
```

## 6. Faculty viewing course enrollments.

```
May 15 22:27
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5: ~/Downloads/Miniproject -final
lenovo@lenovo-Yoga-Slim-7-Pro-14IHU5:~/Downloads/Miniproject -final$ ./client
Connected to Academia Portal Server

Welcome to Academia Portal
1. Admin
2. Faculty
3. Student
Enter your choice: 2
Enter username: Badri
Enter password: BadriOS
Login successful

===== FACULTY MENU =====
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice: 3

==== Course Enrollments ====
Course: OS
Enrolled Students: 1/50
Students enrolled:
- Aditya (ID: 0)
-----
===== FACULTY MENU =====
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice: 1
```