IT5412 - ADVANCED DATA STRUCTURES LABORATORY

MINI-PROJECT REPORT

# PROJECT REPORT ON
# "AN ONLINE LEARNING PORTAL MANAGEMENT SYSTEM"

Submitted By:

| | |
|---|---|
| 2022506014 | Haritha K |
| 2022506037 | Monica P |
| 2022506040 | Aanisha S |
| 2022506059 | Vedha R |
| 2022506061 | Amsapriya K |
| 2022506111 | Alamelu KR |
| 2022506301 | Suganya T |

B.Tech IT (4/8)

DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY, CHENNAI – 600 004.

This project report has been submitted as a requirement for completion of the subject Advanced Data Structures with the subject code IT5412.

# TABLE OF CONTENTS

## PREFACE

- In the digital age, online learning has become essential, driving the need for versatile and comprehensive learning platforms. This report details our project to develop an Online Learning Portal Management System designed to meet the diverse needs of educators and learners, using terminal-based outputs. Implemented with C++ and CSV files for data management, the system aims to provide essential tools for course management, user administration, and engagement.
- The project addresses critical challenges in current online learning platforms, such as the lack of effective management functionalities and user engagement tools. By leveraging C++ for its power and efficiency and CSV files for their simplicity, we aim to deliver a robust and accessible solution.
- This report covers the problem statement, proposed solution, technologies used, and the system's functionalities. It includes diagrams of the system architecture and modules and concludes with the project's novelty, societal impact, outcomes, and potential future enhancements. It offers valuable insights into the development and capabilities of our Online Learning Portal Management System, reflecting our commitment to enhancing online education through innovative solutions.

## PROBLEM STATEMENT & PROPOSED SOLUTION

- The surge in the demand for online education and the sheer growth in the number of online learning platforms have brought forth the need for a versatile Online Learning Portal Management System.
- Current platforms may lack certain functionalities crucial for effective management and user engagement.
- Therefore, our project seeks to develop an effective Online Learning Portal Management System.

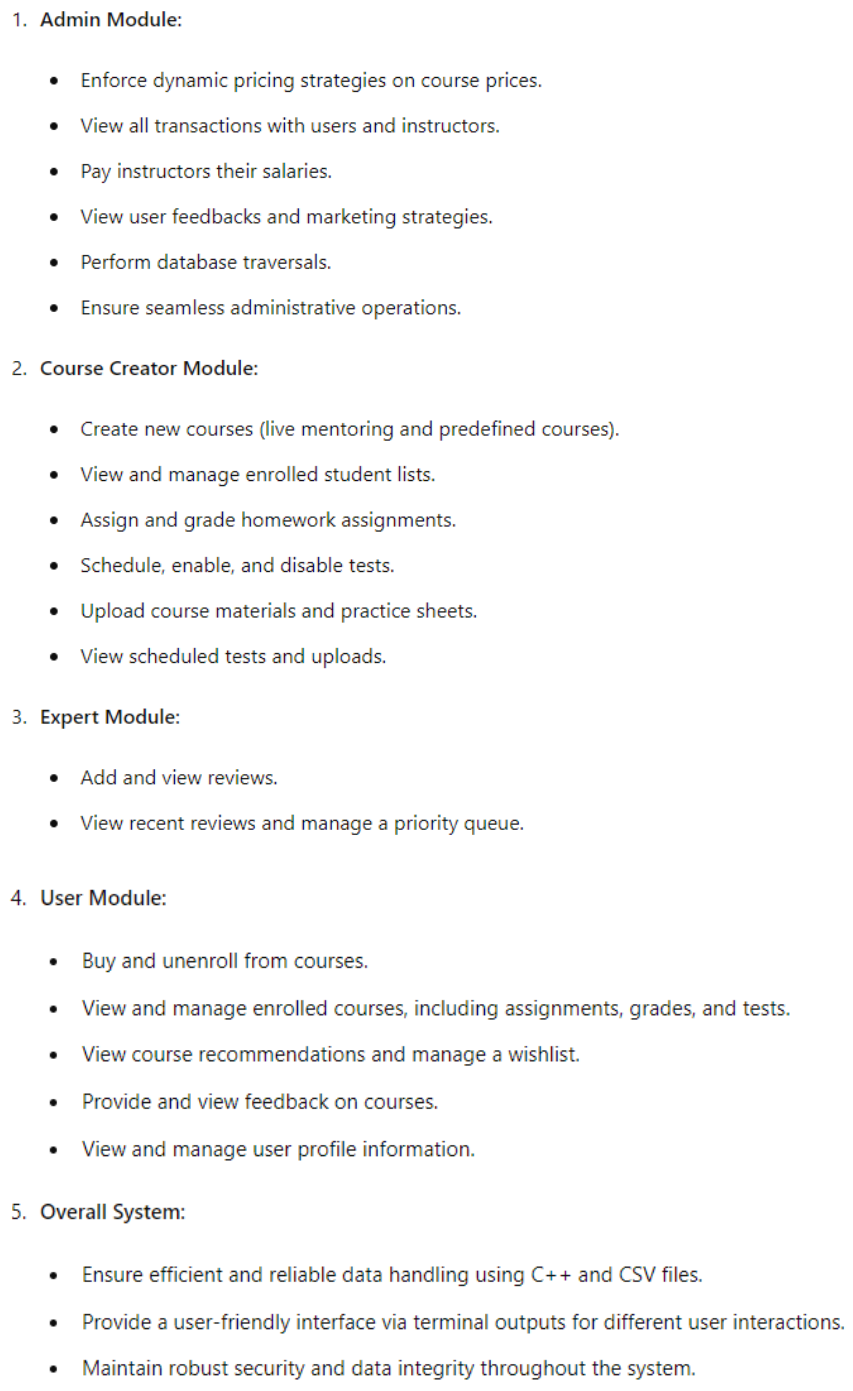
## OBJECTIVE

- The primary objective of our project is to develop an advanced Online Learning Portal Management System that caters to the needs of both educators and learners.
- It tries to foster a conducive environment for effective online education.
- This includes providing a user-friendly interface, robust course management tools, and seamless administrative functionalities.

## PROJECT SCOPE AND DELIVERABLES

- The scope of our Online Learning Portal Management System project encompasses the development and implementation of a comprehensive terminal-based system that caters to the needs of different users, including admins, course creators, experts, and general users.
- By delivering the following functionalities, the project aims to enhance the online learning experience for all users, ensuring effective management, seamless user interaction, and comprehensive educational support through a terminal-based interface.

### Deliverables:



1. **Admin Module:**
   - Enforce dynamic pricing strategies on course prices.
   - View all transactions with users and instructors.
   - Pay instructors their salaries.
   - View user feedbacks and marketing strategies.
   - Perform database traversals.
   - Ensure seamless administrative operations.
2. **Course Creator Module:**
   - Create new courses (live mentoring and predefined courses).
   - View and manage enrolled student lists.
   - Assign and grade homework assignments.
   - Schedule, enable, and disable tests.
   - Upload course materials and practice sheets.
   - View scheduled tests and uploads.
3. **Expert Module:**
   - Add and view reviews.
   - View recent reviews and manage a priority queue.
4. **User Module:**
   - Buy and unenroll from courses.
   - View and manage enrolled courses, including assignments, grades, and tests.
   - View course recommendations and manage a wishlist.
   - Provide and view feedback on courses.
   - View and manage user profile information.
5. **Overall System:**
   - Ensure efficient and reliable data handling using C++ and CSV files.
   - Provide a user-friendly interface via terminal outputs for different user interactions.
   - Maintain robust security and data integrity throughout the system.

## TECH STACK AND PROJECT APPROACH

Tech Stack:

- Programming Language: C++ was used for developing the core functionality of the system due to its efficiency, performance, and object-oriented features.
- Data Storage: CSV Files were employed for data management and storage, allowing easy read/write operations and facilitating straightforward data manipulation.
- Development Environment: Terminal and C++ compatible IDE code editor, Visual Studio were used.
- Compiler: GCC (GNU Compiler Collection) compatible with the development environment was used.

Project Approach:

- Requirements Gathering:
  - The needs of different user roles (admin, course creator, expert, user) and their specific functionalities were identified.
- Design Phase:
  - System Architecture Design: The overall system structure and interaction between different modules were outlined.
  - Class Design: Classes and their relationships based on system functionalities were defined.
- Implementation Phase:
  - The Admin Module was developed, implementing functionalities for pricing strategy, transaction management, and feedback viewing.
  - The Course Creator Module was built, including tools for course creation, student management, and material uploads.
  - The Expert Module was enabled for review management and priority queue handling.
  - The User Module and Transaction Module was implemented for course enrollment, assignment management, and feedback mechanisms.
  - CSV File Handling was integrated to ensure efficient data storage and retrieval.
- Testing Phase:
  - Unit Testing: Individual components were tested for functionality and performance.
  - Integration Testing: Different modules were tested to ensure seamless interaction.
  - User Acceptance Testing: The system was validated with potential users to ensure it met their requirements.

- Deployment Phase:
  - Deployment scripts and documentation were prepared.
  - The system was made ready for use in the target environment (terminal-based execution).

**A STRUCTURED LAYOUT OF THE FUNCTIONALITIES OF THE SYSTEM ACHIEVED AND THEIR RESPECTIVE CLASSES**

1. Admin
    1. Enforce dynamic pricing strategy on the prices of the courses offered - transaction
    2. View all transactions with the users - transaction
    3. Pay an instructor his/her salary - transaction
    4. View all transactions with the instructors - transaction
    5. View user feedbacks - feedback
    6. View Marketing Strategies – admin
    7. Database traversals - admin
    8. Exit
2. Course Creator – instructor + test
    1. New course
        1. Live mentoring
        2. Predefined course
        3. Back
    2. View my courses
        1. View enrolled student list
        2. Assign homework
        3. Grade assignments
        4. Schedule a test
        5. Enable a test
        6. Disable a test
        7. Upload material
        8. Upload practice sheets
        9. View Scheduled tests
        10. View uploads
        11. Back
    3. Exit

3. Expert – instructor + feedback
    1. Add Review
    2. View Recent Reviews
    3. View Priority Queue
    4. Exit
4. User
    1. Buy a Course - transaction
    2. View Enrolled Courses - user
        1. Select
            1. View assignments
            2. View grades
            3. Submit assignment
            4. Take Tests - test
            5. Back
        2. Back
    3. Unenroll from a Course - transaction
    4. View Course Recommendations - user
    5. Manage Wishlist - courses
        1. Add
        2. Remove
        3. View
    6. Give Feedback on a Course - feedback
    7. View given feedbacks - feedback
    8. View Profile - admin
    9. Exit
5. Exit

## CLASSES USED IN THE IMPLEMENTATION OF THE SYSTEM

- Admin
- User
- Instructor (Course Creator, Expert)
- Courses
- Transaction
- Test
- Feedback

In the implementation of all the above classes, object-oriented programming concepts like object creation, inheritance, virtual functions, pure virtual functions, abstract classes, etc. have been used.

## DATA STRUCTURES AND ALGORITHMS USED IN THE IMPLEMENTATION OF THE SYSTEM

Data structures used:

- AVL Tree
- Heap Tree
- Splay Tree
- Graph

Algorithms used:

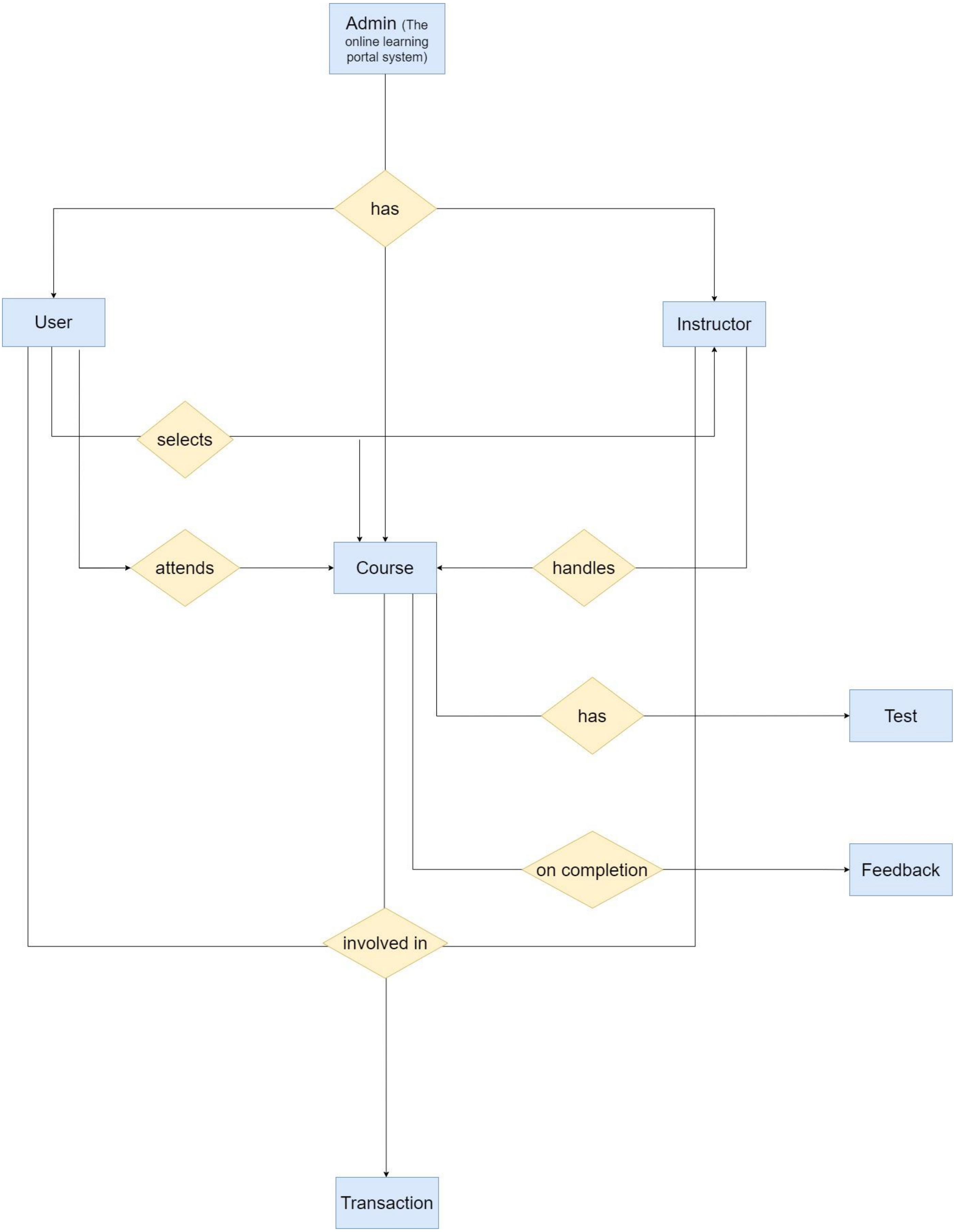
- Depth First Traversal
- Prim’s Algorithm
- Topological Sorting
- Dijkstra’s Algorithm
- Bellman-Ford Algorithm
- Floyd Warshall Algorithm

Brief explanation as to what DSA concept was chosen to implement which functionality and why so:

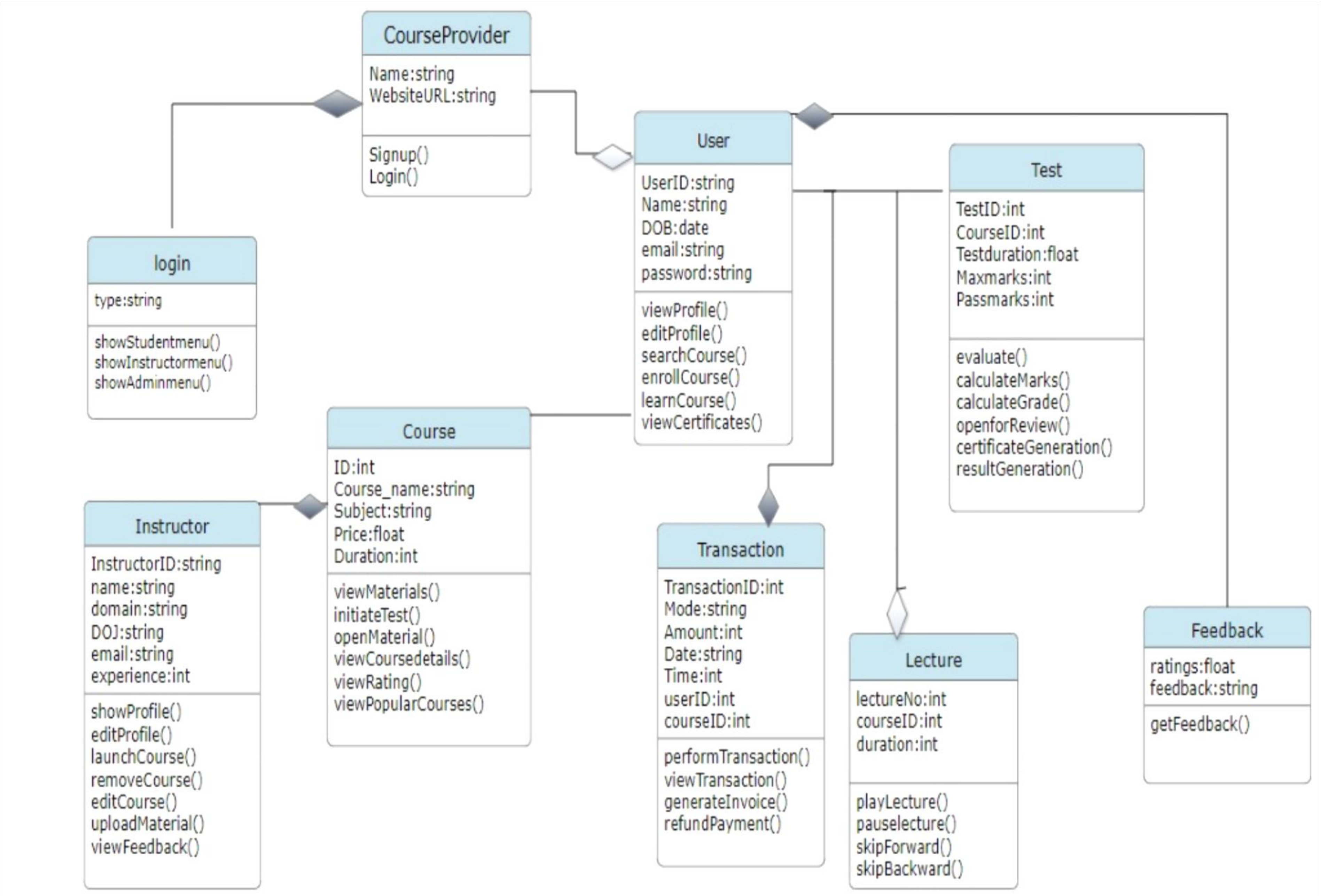
- **AVL Tree**: Utilized for managing the user's wishlist. An AVL tree was chosen for its self-balancing property, ensuring that the wishlist operations such as insertion, deletion, and search can be performed efficiently with a balanced tree structure. This ensures optimal performance even with a large number of wishlist items. (This functionality is associated with the Course Module)
- **Splay Tree**: Used for splaying the reviews entered by experts. Splay trees were chosen for their self-adjusting property, where recently accessed nodes are brought to the root, which optimizes access times for frequently accessed reviews and improves overall performance. (This functionality is associated with the Instructor Module)
- **Heap Tree**: Employed to prioritize enrollment requests of premium users over non-premium users, as well as to prioritize the subjects to be reviewed by experts. A heap tree allows for efficient insertion and removal of elements while maintaining the desired priority order, ensuring that premium users and high-priority subjects are processed promptly. (These functionalities are associated with the Transaction, Instructor and Feedback Module)

- **Graph along with Depth First Traversal (DFS)**: Used for viewing the instructors and the courses undertaken by them. DFS traversal algorithms allow for exploring the instructor-course relationships efficiently, providing insights into the distribution of courses among instructors and facilitating effective course management. (This functionality is associated with the Admin Module)
- **Graph along with Dijkstra's Algorithm and Prim's Algorithm**: Utilized for course recommendations to users. By representing the relationships between courses as a graph, Dijkstra's algorithm was employed to find the shortest path from a user's enrolled courses to potential recommended courses. Additionally, Minimum Spanning Tree (Prim's Algorithm) was used to identify a subset of courses that maximizes the diversity of recommendations while maintaining relevance. (This functionality is associated with the User Module)
- **Graph along with Topological Sorting**: Implemented for sorting marketing strategies and displaying to the admin the various possible topological sorts of the orders of the marketing strategies. Topological sorting ensures that the marketing strategies are executed in the correct sequence, maximizing their effectiveness and ensuring smooth execution of marketing campaigns. (This functionality is associated with the Admin Module)
- **Graph along with Bellman-Ford Algorithm and Floyd Warshall Algor**ithm: Implemented to handle two different kinds of dynamic pricing strategies. A graph data structure was chosen to represent the relationships between courses and their pricing components. The Bellman-Ford algorithm was used to find the shortest path for one type of dynamic pricing, while the Floyd Warshall algorithm was utilized for another type, providing flexibility in implementing various pricing strategies efficiently. (This functionality is associated with the Transaction Module)
- By leveraging these carefully chosen data structures and algorithms, our Online Learning Portal Management System achieves optimal performance, scalability, and efficiency in managing various functionalities and processing user interactions.

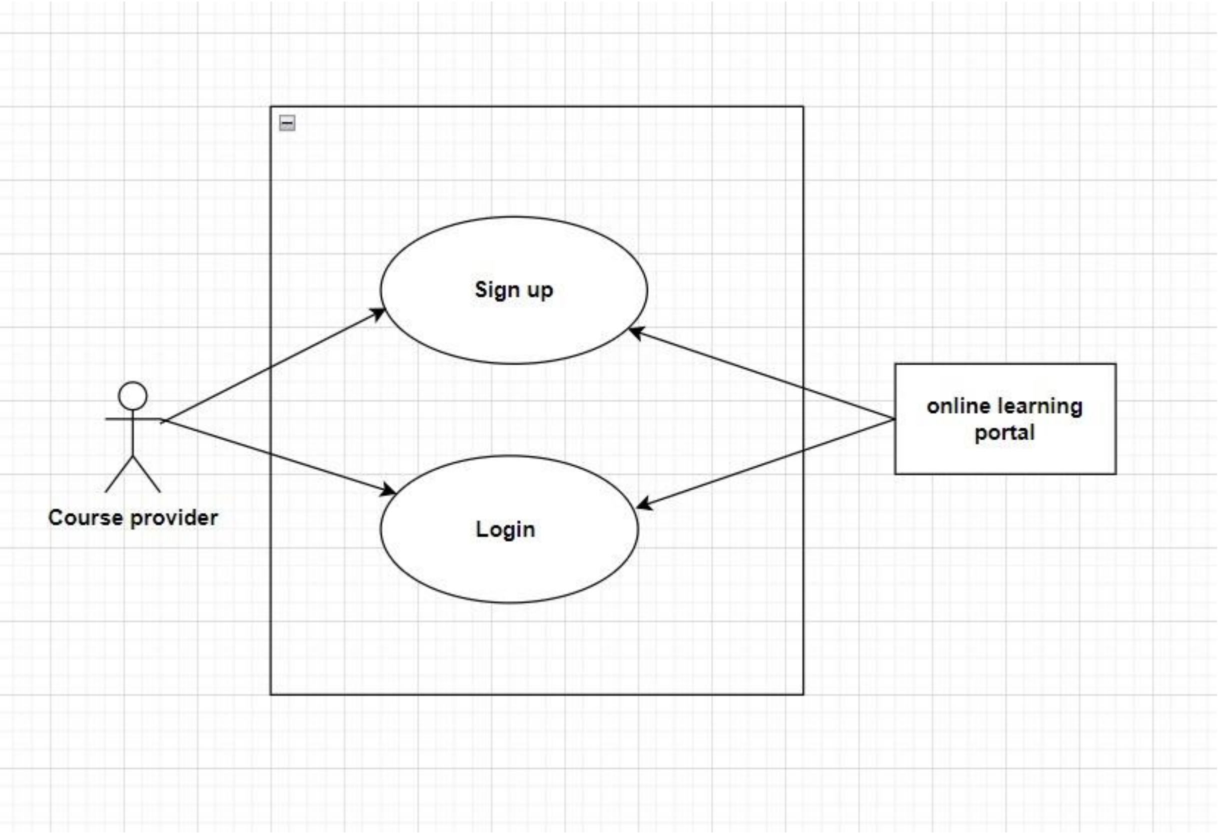
## OVERALL BLOCK DIAGRAM OF THE SYSTEM



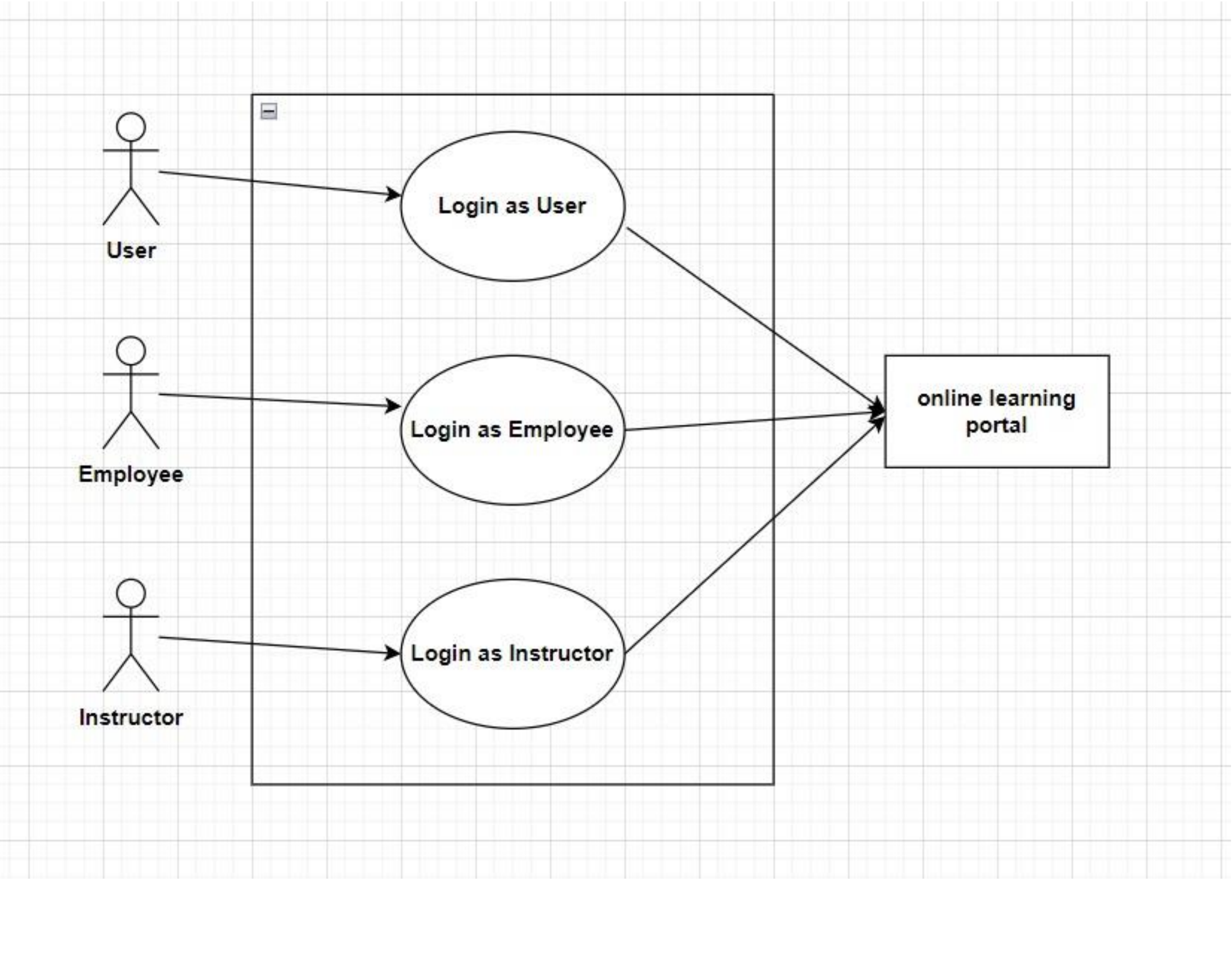
## GENERIC CLASS DIAGRAM OF THE SYSTEM



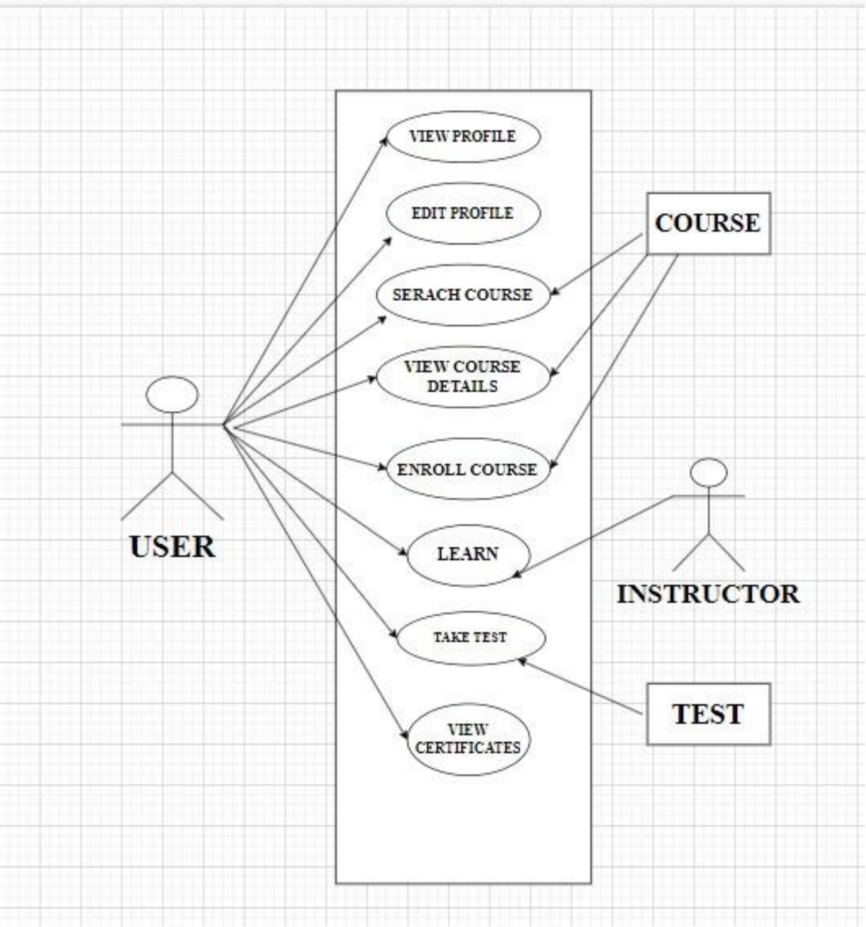
## GENERIC USE CASE DIAGRAM OF COURSE PROVIDER MODULE



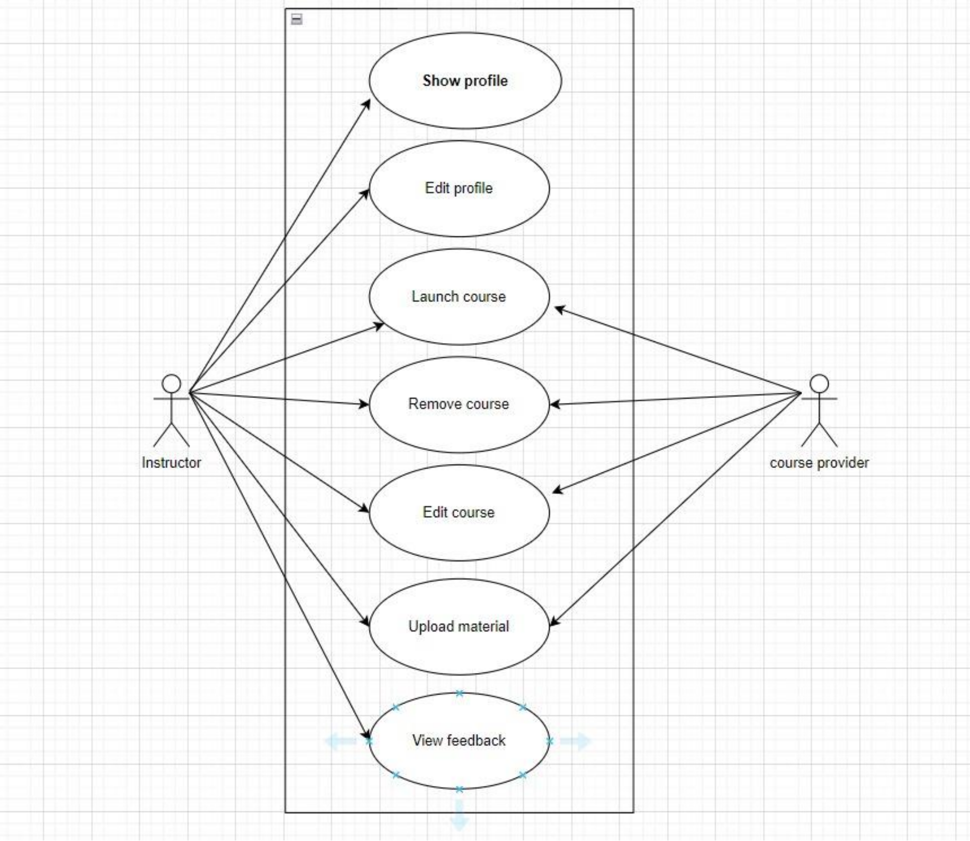
## GENERIC USE CASE DIAGRAM OF LOGIN MODULE



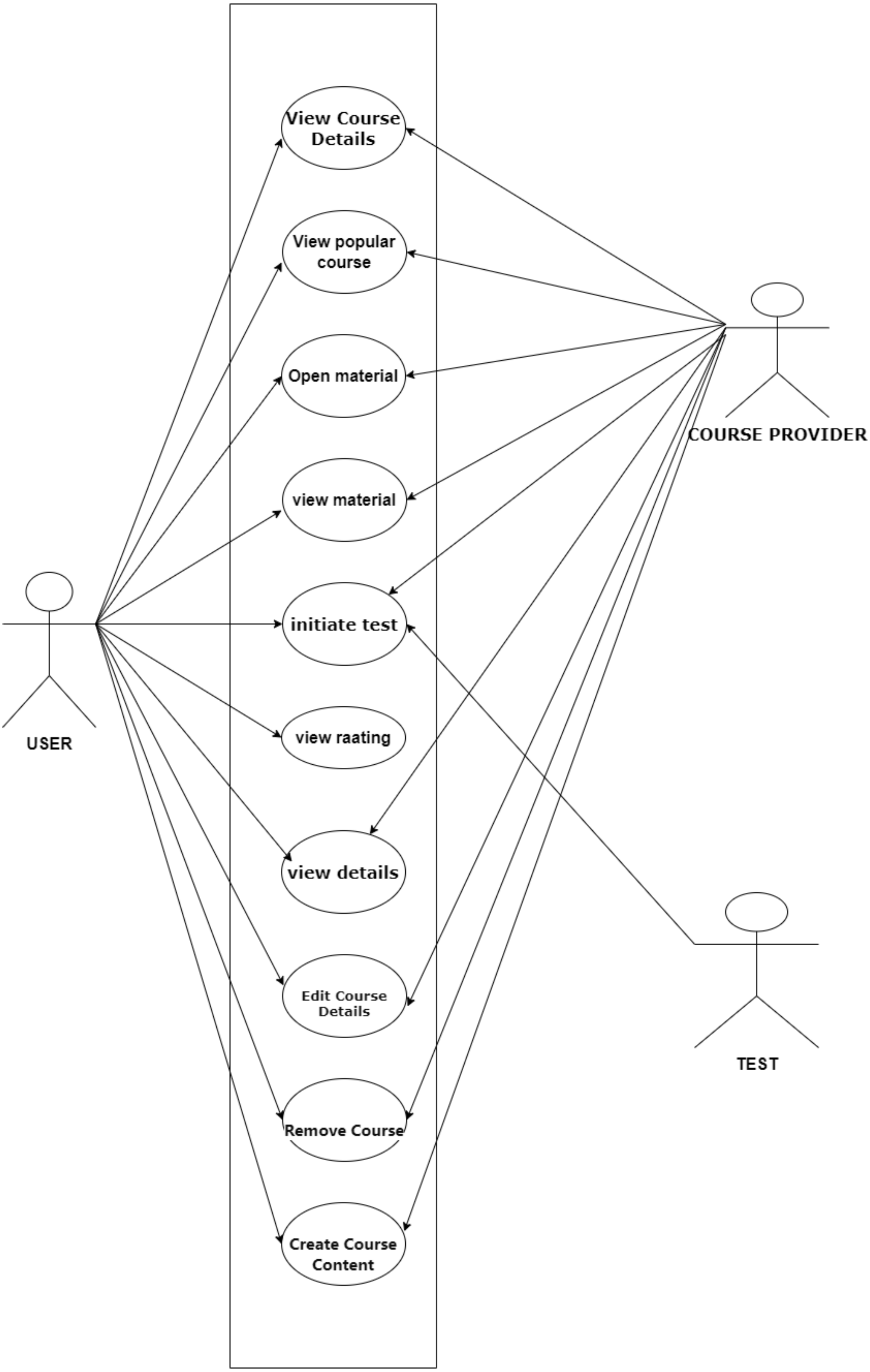
## GENERIC USE CASE DIAGRAM OF USER MODULE



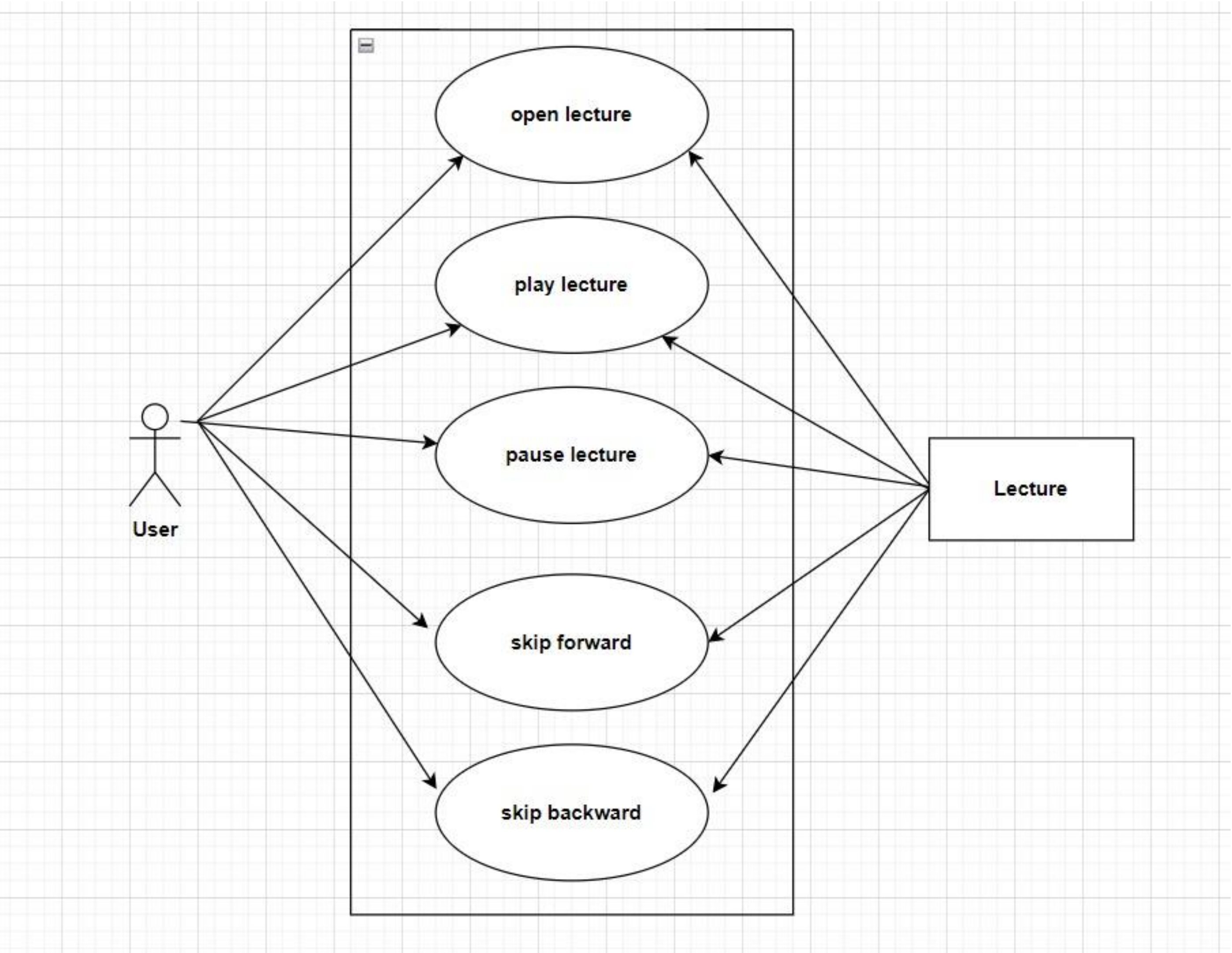
## GENERIC USE CASE DIAGRAM OF INSTRUCTOR MODULE



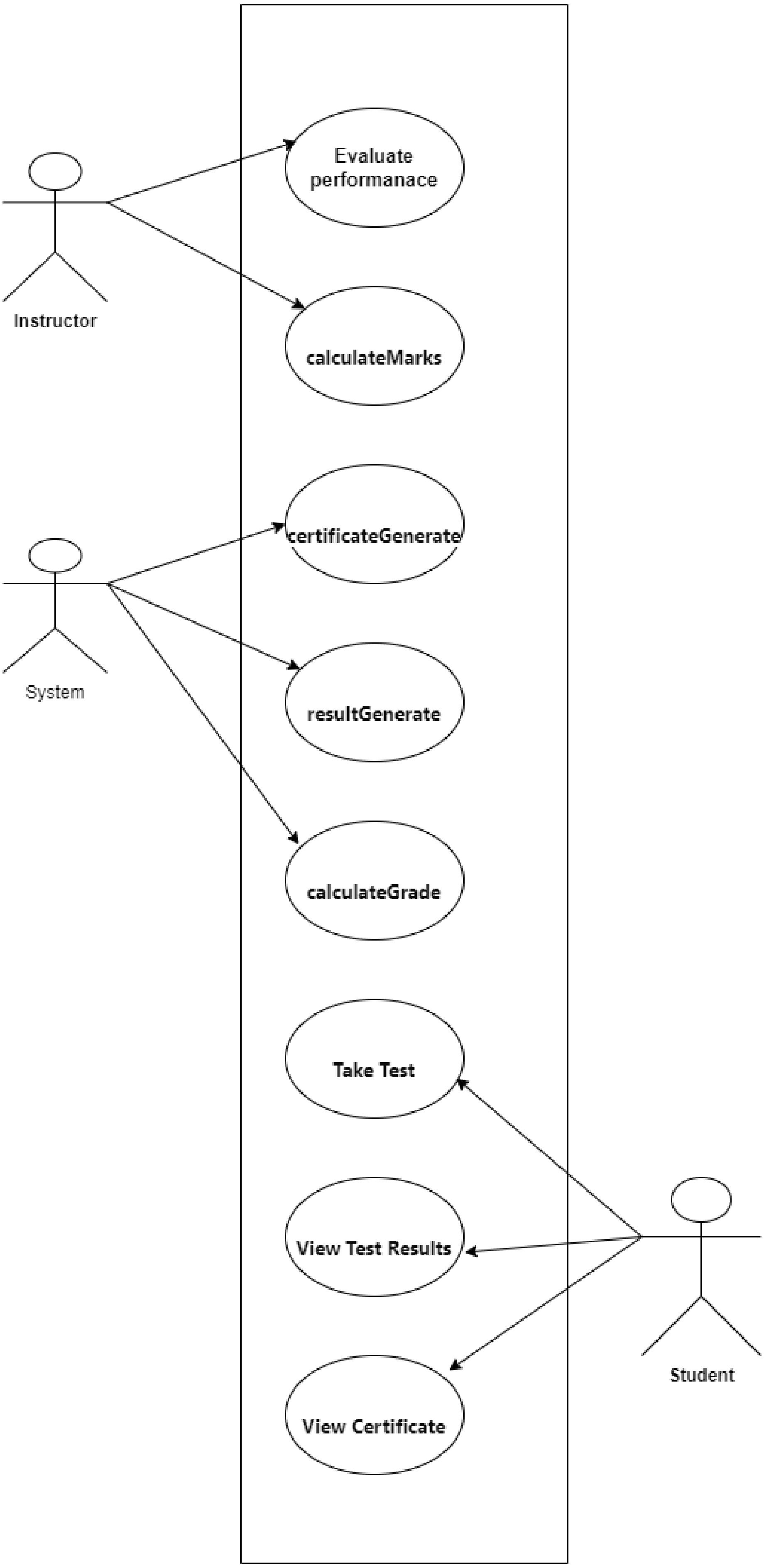
## GENERIC USE CASE DIAGRAM OF COURSE MODULE



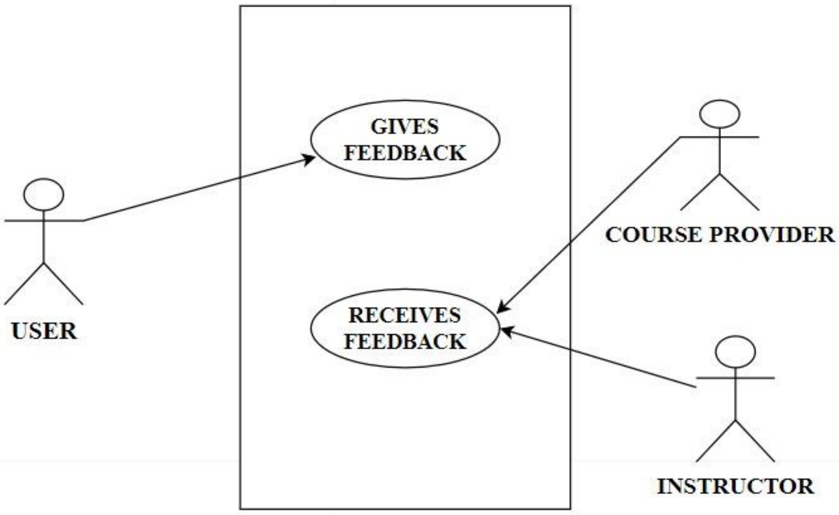
## GENERIC USE CASE DIAGRAM OF LECTURE MODULE



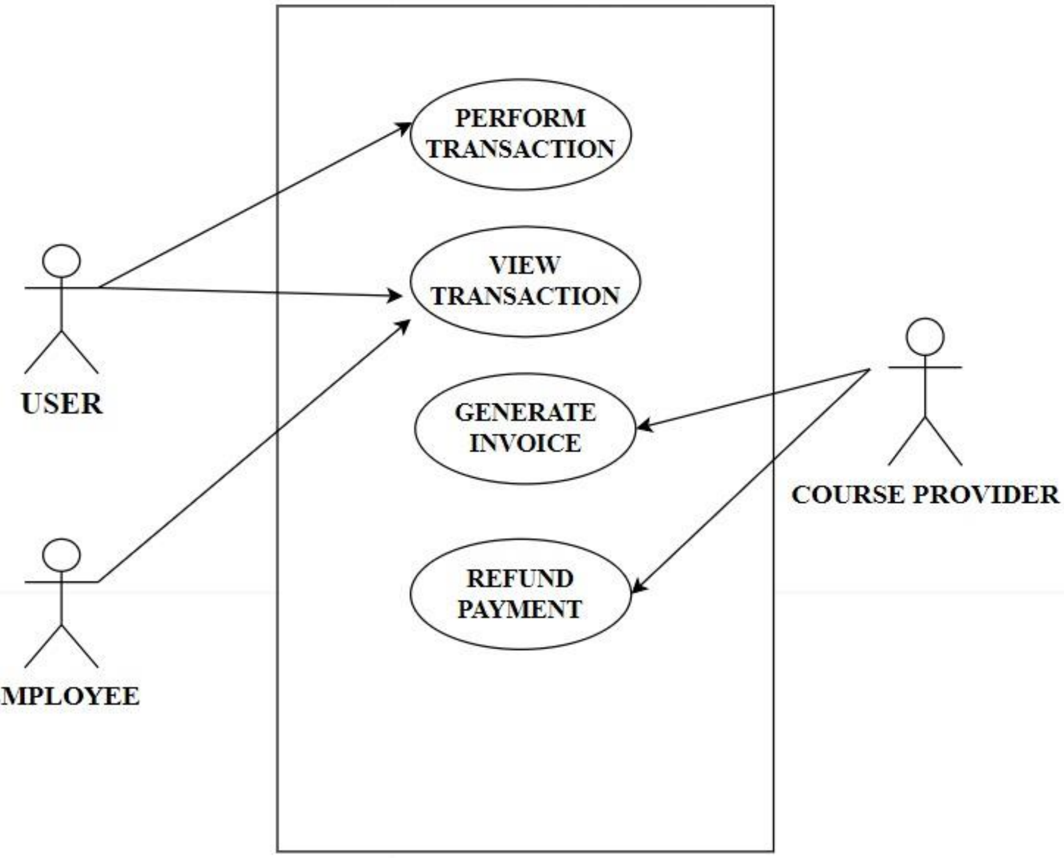
## GENERIC USE CASE DIAGRAM OF TEST MODULE



## GENERIC USE CASE DIAGRAM OF FEEDBACK MODULE

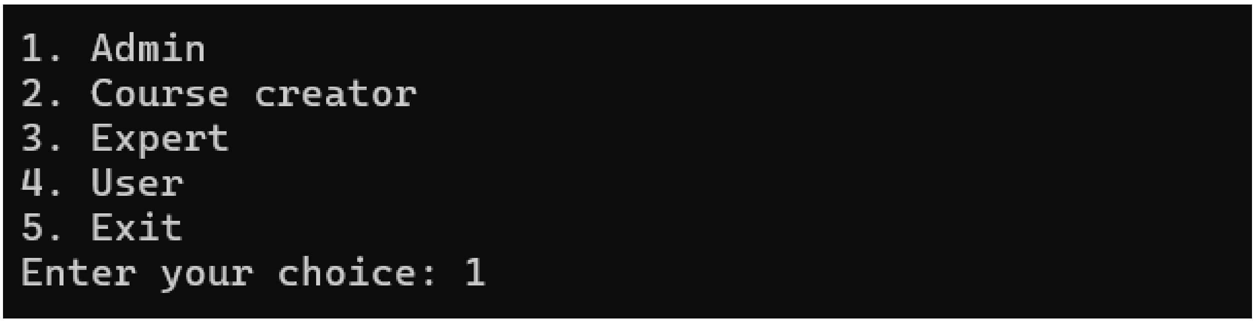


## GENERIC USE CASE DIAGRAM OF TRANSACTION MODULE



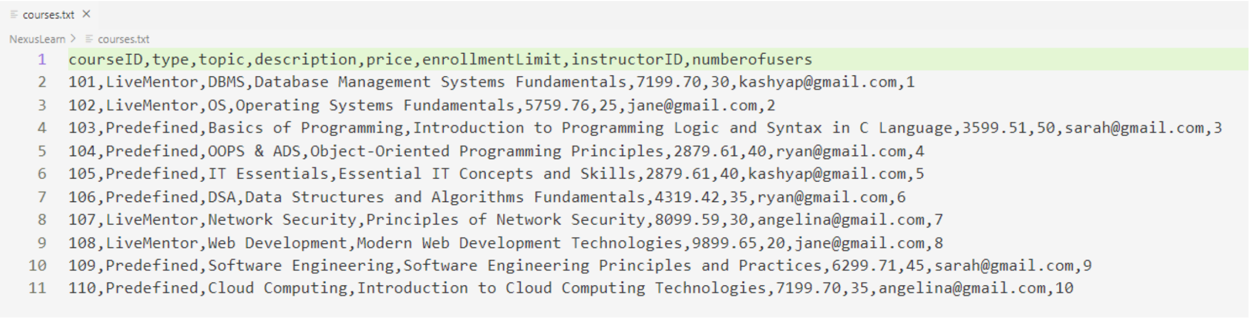
## SCREENSHOTS OF THE SYSTEM

### Admin's functionalities once they login:

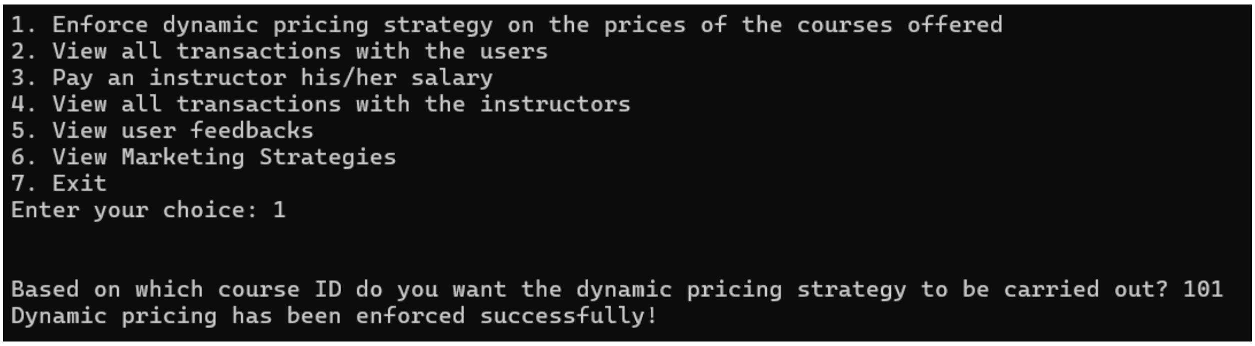


```
1. Admin
2. Course creator
3. Expert
4. User
5. Exit
Enter your choice: 1
```

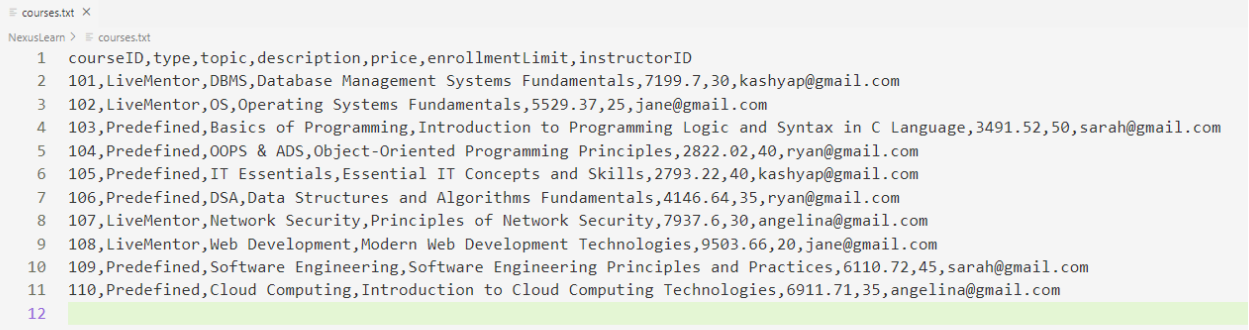
Enforcing dynamic pricing on the prices of the courses in the database:



```
courseID,type,topic,description,price,enrollmentLimit,instructorID,numberofusers
101,LiveMentor,DBMS,Database Management Systems Fundamentals,7199.70,30,kashyap@gmail.com,1
102,LiveMentor,OS,Operating Systems Fundamentals,5759.76,25,jane@gmail.com,2
103,Predefined,Basics of Programming,Introduction to Programming Logic and Syntax in C Language,3599.51,50,sarah@gmail.com,3
104,Predefined,OOPS & ADS,Object-Oriented Programming Principles,2879.61,40,ryan@gmail.com,4
105,Predefined,IT Essentials,Essential IT Concepts and Skills,2879.61,40,kashyap@gmail.com,5
106,Predefined,DSA,Data Structures and Algorithms Fundamentals,4319.42,35,ryan@gmail.com,6
107,LiveMentor,Network Security,Principles of Network Security,8099.59,30,angelina@gmail.com,7
108,LiveMentor,Web Development,Modern Web Development Technologies,9899.65,20,jane@gmail.com,8
109,Predefined,Software Engineering,Software Engineering Principles and Practices,6299.71,45,sarah@gmail.com,9
110,Predefined,Cloud Computing,Introduction to Cloud Computing Technologies,7199.70,35,angelina@gmail.com,10
```

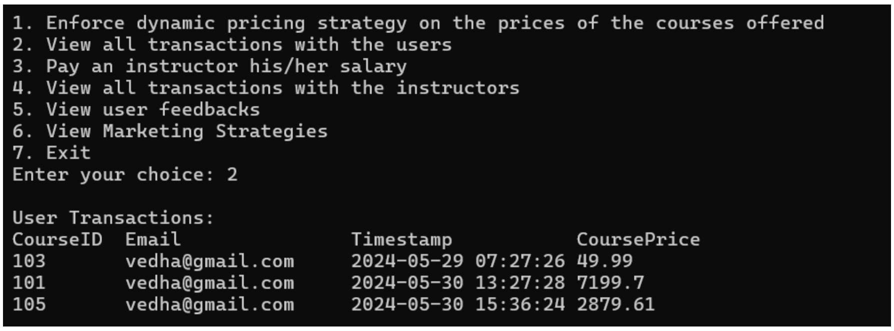


```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 1


Based on which course ID do you want the dynamic pricing strategy to be carried out? 101
Dynamic pricing has been enforced successfully!
```



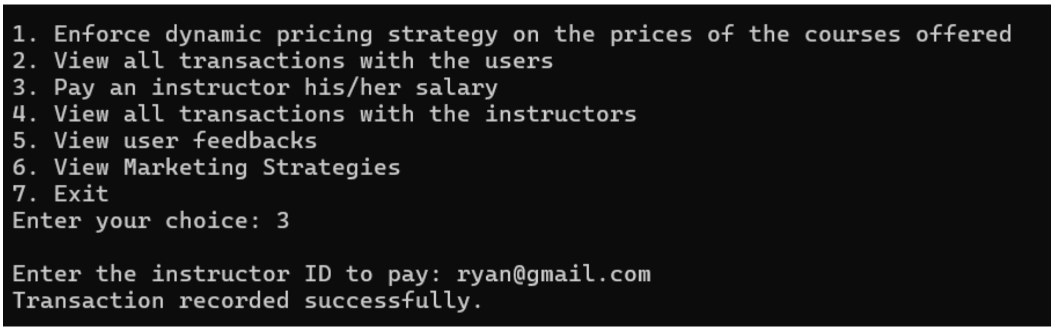
```
courseID,type,topic,description,price,enrollmentLimit,instructorID
101,LiveMentor,DBMS,Database Management Systems Fundamentals,7199.7,30,kashyap@gmail.com
102,LiveMentor,OS,Operating Systems Fundamentals,5529.37,25,jane@gmail.com
103,Predefined,Basics of Programming,Introduction to Programming Logic and Syntax in C Language,3491.52,50,sarah@gmail.com
104,Predefined,OOPS & ADS,Object-Oriented Programming Principles,2822.02,40,ryan@gmail.com
105,Predefined,IT Essentials,Essential IT Concepts and Skills,2793.22,40,kashyap@gmail.com
106,Predefined,DSA,Data Structures and Algorithms Fundamentals,4146.64,35,ryan@gmail.com
107,LiveMentor,Network Security,Principles of Network Security,7937.6,30,angelina@gmail.com
108,LiveMentor,Web Development,Modern Web Development Technologies,9503.66,20,jane@gmail.com
109,Predefined,Software Engineering,Software Engineering Principles and Practices,6110.72,45,sarah@gmail.com
110,Predefined,Cloud Computing,Introduction to Cloud Computing Technologies,6911.71,35,angelina@gmail.com

```

Viewing all transactions between the users and the system:



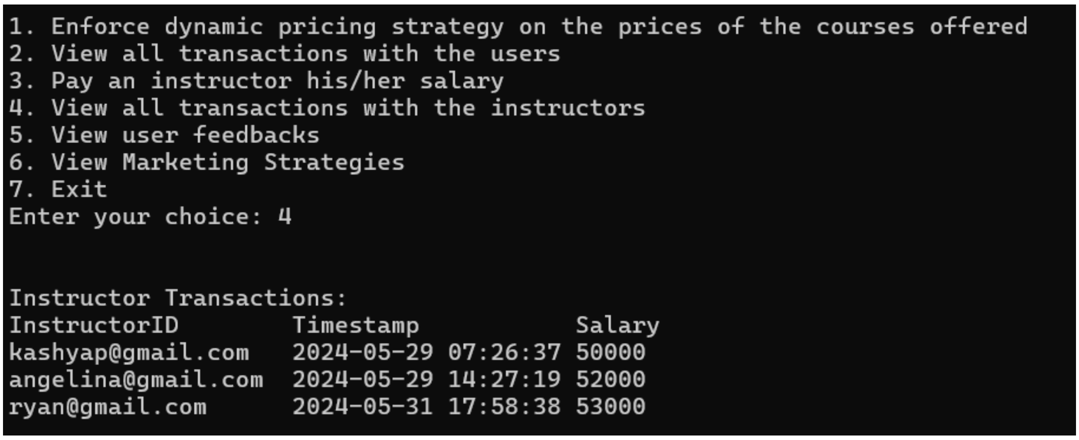
```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 2

User Transactions:
CourseID  Email               Timestamp           CoursePrice
103       vedha@gmail.com     2024-05-29 07:27:26 49.99
101       vedha@gmail.com     2024-05-30 13:27:28 7199.7
105       vedha@gmail.com     2024-05-30 15:36:24 2879.61
```

Paying an instructor:



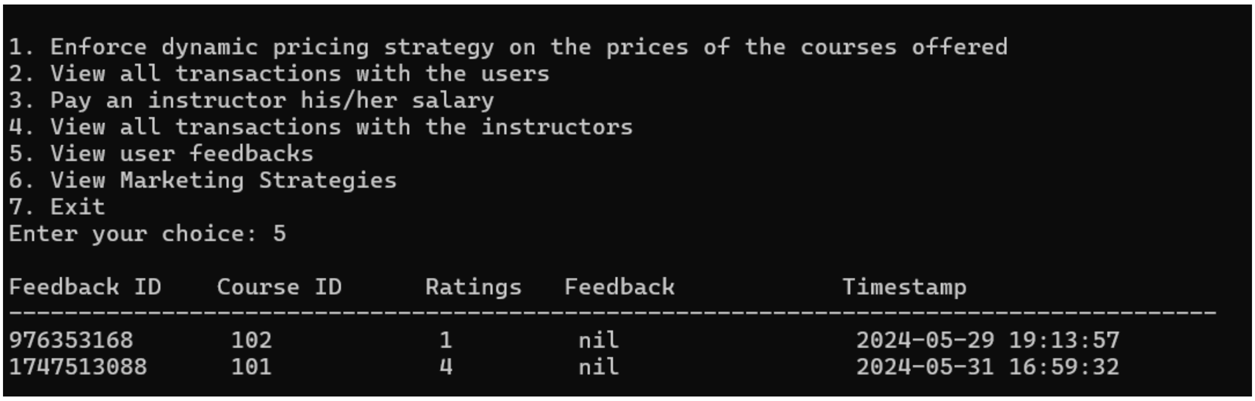
```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 3

Enter the instructor ID to pay: ryan@gmail.com
Transaction recorded successfully.
```

Viewing all transactions between the system and the instructors:



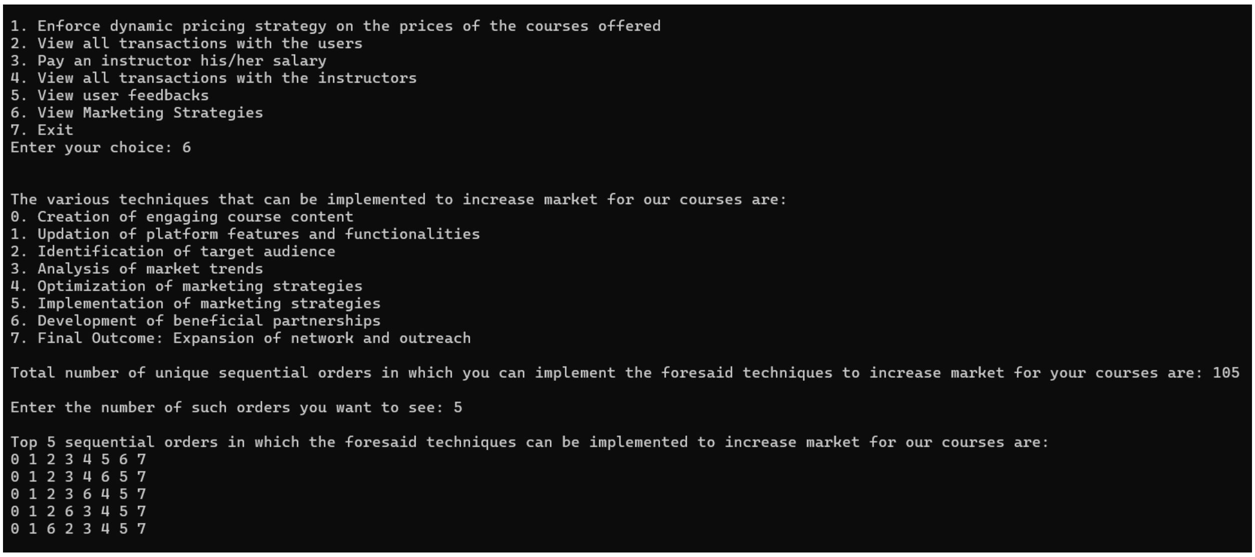
```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 4


Instructor Transactions:
InstructorID        Timestamp           Salary
kashyap@gmail.com   2024-05-29 07:26:37 50000
angelina@gmail.com  2024-05-29 14:27:19 52000
ryan@gmail.com      2024-05-31 17:58:38 53000
```

Viewing user feedbacks:



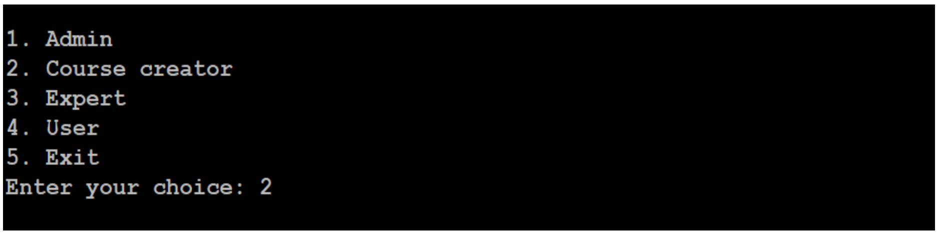
```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 5

Feedback ID    Course ID      Ratings   Feedback           Timestamp
-------------------------------------------------------------------------------------
976353168       102            1         nil                2024-05-29 19:13:57
1747513088      101            4         nil                2024-05-31 16:59:32
```

Viewing marketing strategies:



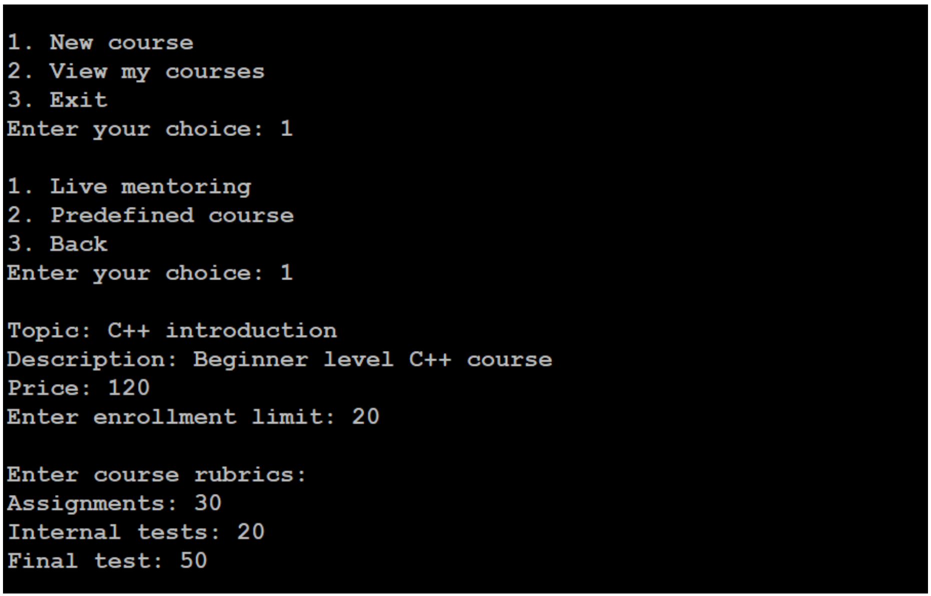
```
1. Enforce dynamic pricing strategy on the prices of the courses offered
2. View all transactions with the users
3. Pay an instructor his/her salary
4. View all transactions with the instructors
5. View user feedbacks
6. View Marketing Strategies
7. Exit
Enter your choice: 6


The various techniques that can be implemented to increase market for our courses are:
0. Creation of engaging course content
1. Updation of platform features and functionalities
2. Identification of target audience
3. Analysis of market trends
4. Optimization of marketing strategies
5. Implementation of marketing strategies
6. Development of beneficial partnerships
7. Final Outcome: Expansion of network and outreach

Total number of unique sequential orders in which you can implement the foresaid techniques to increase market for your courses are: 105

Enter the number of such orders you want to see: 5

Top 5 sequential orders in which the foresaid techniques can be implemented to increase market for our courses are:
0 1 2 3 4 5 6 7
0 1 2 3 4 6 5 7
0 1 2 3 6 4 5 7
0 1 2 6 3 4 5 7
0 1 6 2 3 4 5 7
```

**Course creator’s functionalities once they login:**

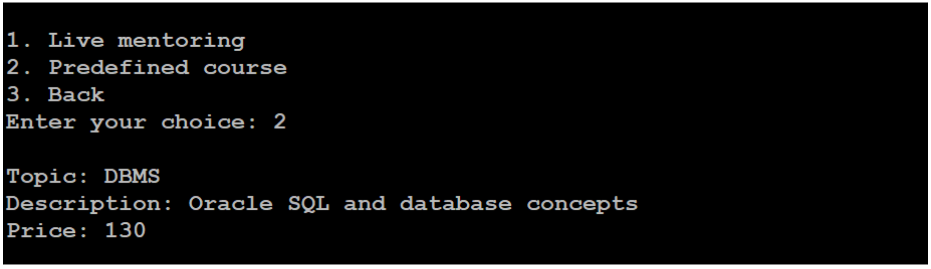


```
1. Admin
2. Course creator
3. Expert
4. User
5. Exit
Enter your choice: 2
```

Adding new courses:

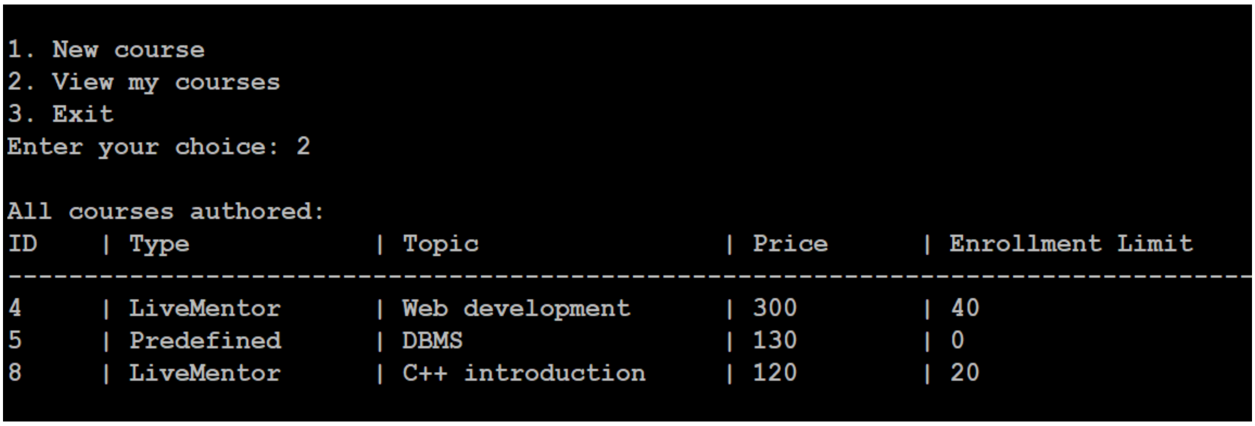


```
1. New course
2. View my courses
3. Exit
Enter your choice: 1

1. Live mentoring
2. Predefined course
3. Back
Enter your choice: 1

Topic: C++ introduction
Description: Beginner level C++ course
Price: 120
Enter enrollment limit: 20

Enter course rubrics:
Assignments: 30
Internal tests: 20
Final test: 50
```



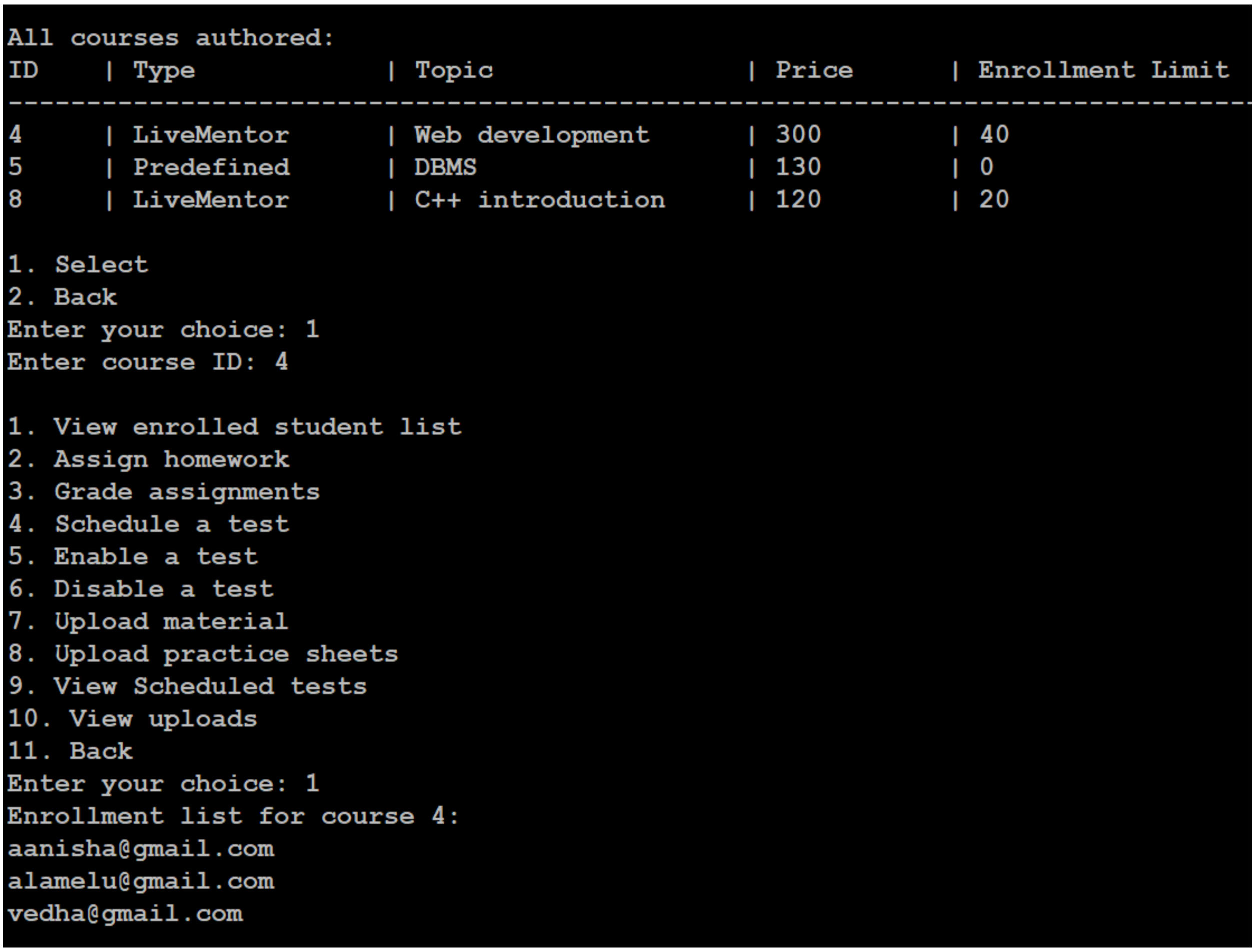
```
1. Live mentoring
2. Predefined course
3. Back
Enter your choice: 2

Topic: DBMS
Description: Oracle SQL and database concepts
Price: 130
```

Viewing added courses:



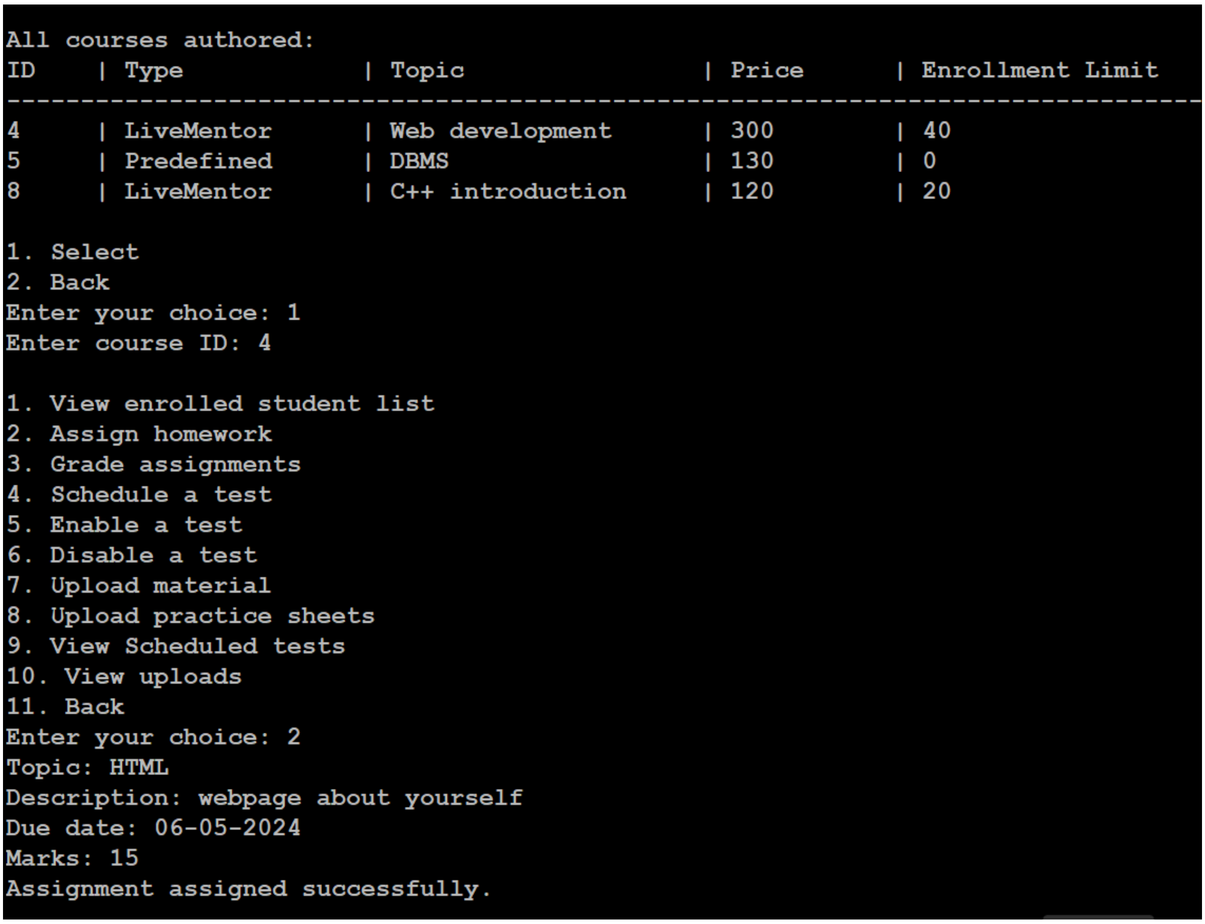
```
1. New course
2. View my courses
3. Exit
Enter your choice: 2

All courses authored:
ID     | Type              | Topic                  | Price       | Enrollment Limit
--------------------------------------------------------------------------------------
4      | LiveMentor        | Web development        | 300         | 40
5      | Predefined        | DBMS                   | 130         | 0
8      | LiveMentor        | C++ introduction       | 120         | 20
```

Seeing the student enrollment list for a particular course:



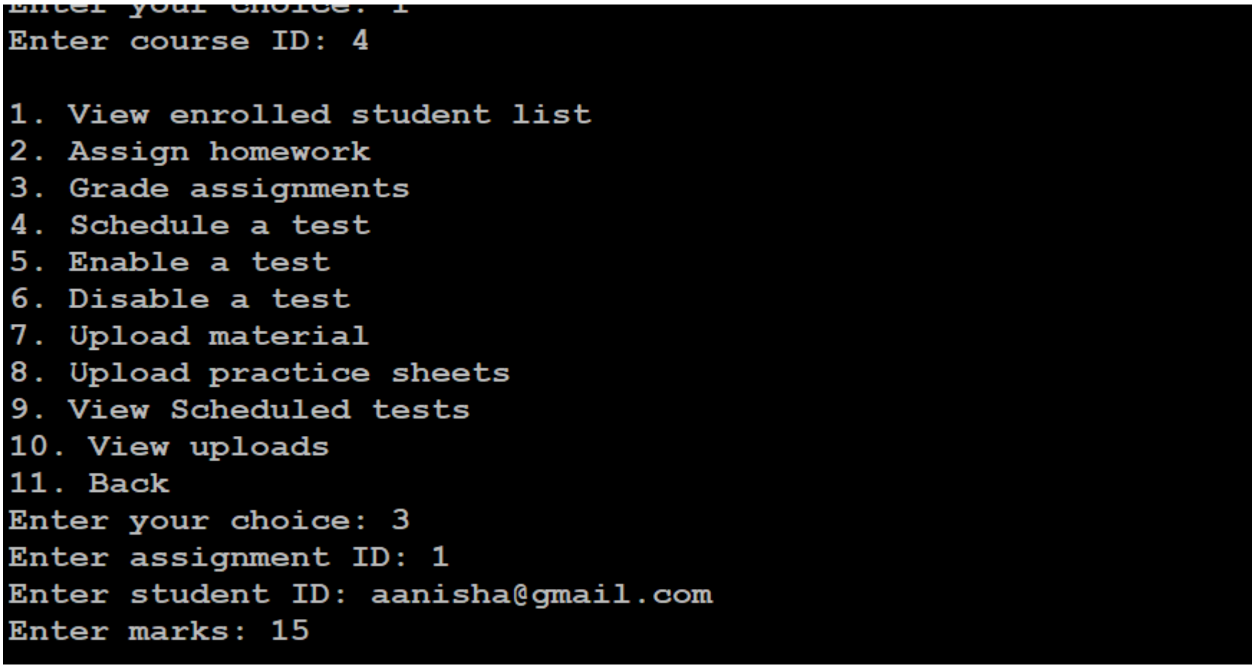
```
All courses authored:
ID     | Type              | Topic                  | Price       | Enrollment Limit
--------------------------------------------------------------------------------------
4      | LiveMentor        | Web development        | 300         | 40
5      | Predefined        | DBMS                   | 130         | 0
8      | LiveMentor        | C++ introduction       | 120         | 20

1. Select
2. Back
Enter your choice: 1
Enter course ID: 4

1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 1
Enrollment list for course 4:
aanisha@gmail.com
alamelu@gmail.com
vedha@gmail.com
```

Posting assignments:



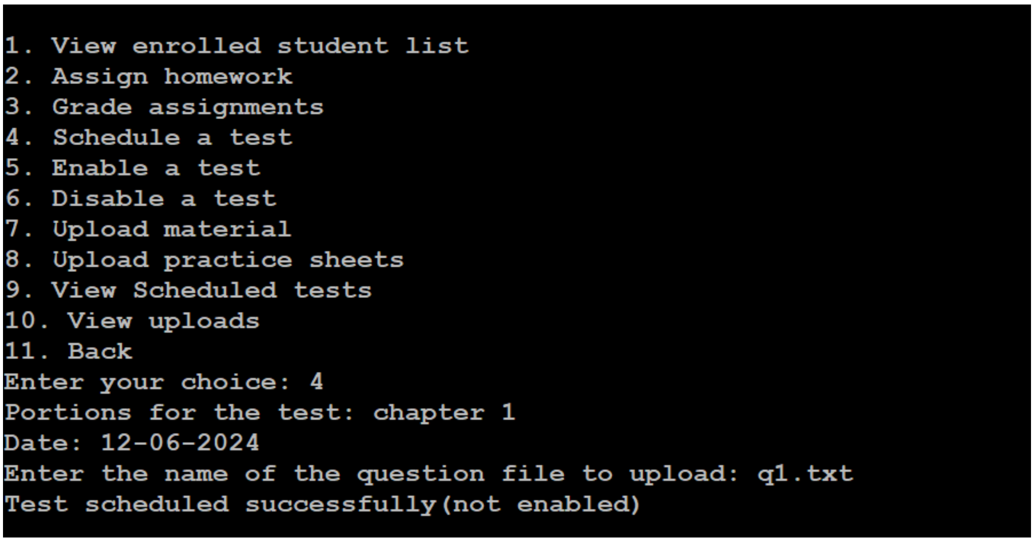
```
All courses authored:
ID     | Type            | Topic                 | Price       | Enrollment Limit
-------------------------------------------------------------------------------------
4      | LiveMentor      | Web development       | 300         | 40
5      | Predefined      | DBMS                  | 130         | 0
8      | LiveMentor      | C++ introduction      | 120         | 20

1. Select
2. Back
Enter your choice: 1
Enter course ID: 4

1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 2
Topic: HTML
Description: webpage about yourself
Due date: 06-05-2024
Marks: 15
Assignment assigned successfully.
```

Grading a student’s assignment:



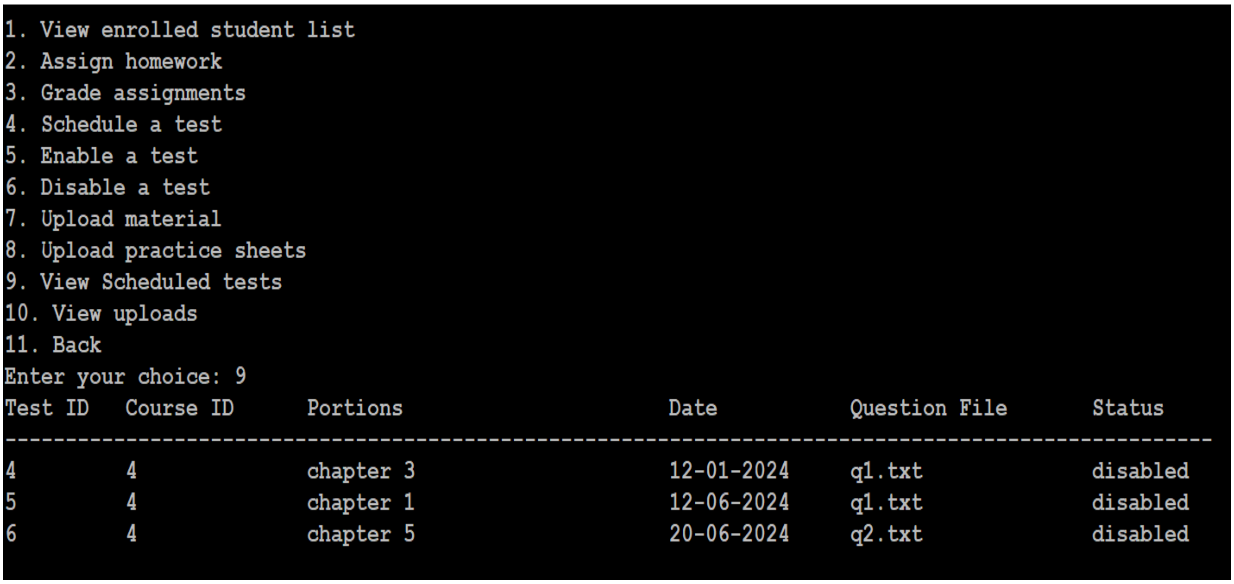
```
Enter your choice: 1
Enter course ID: 4

1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 3
Enter assignment ID: 1
Enter student ID: aanisha@gmail.com
Enter marks: 15
```

Scheduling tests:

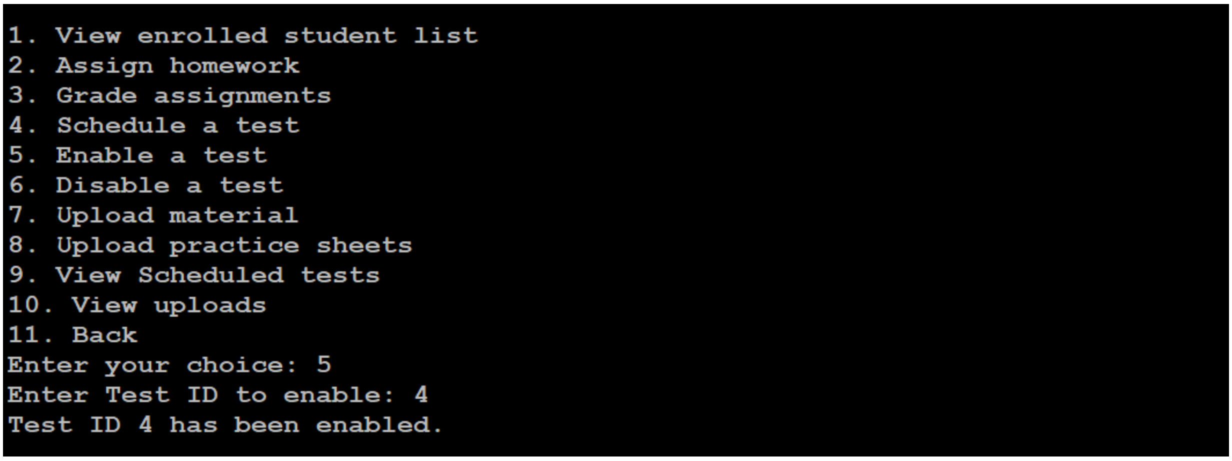


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 4
Portions for the test: chapter 1
Date: 12-06-2024
Enter the name of the question file to upload: q1.txt
Test scheduled successfully(not enabled)
```

Enabling scheduled tests:

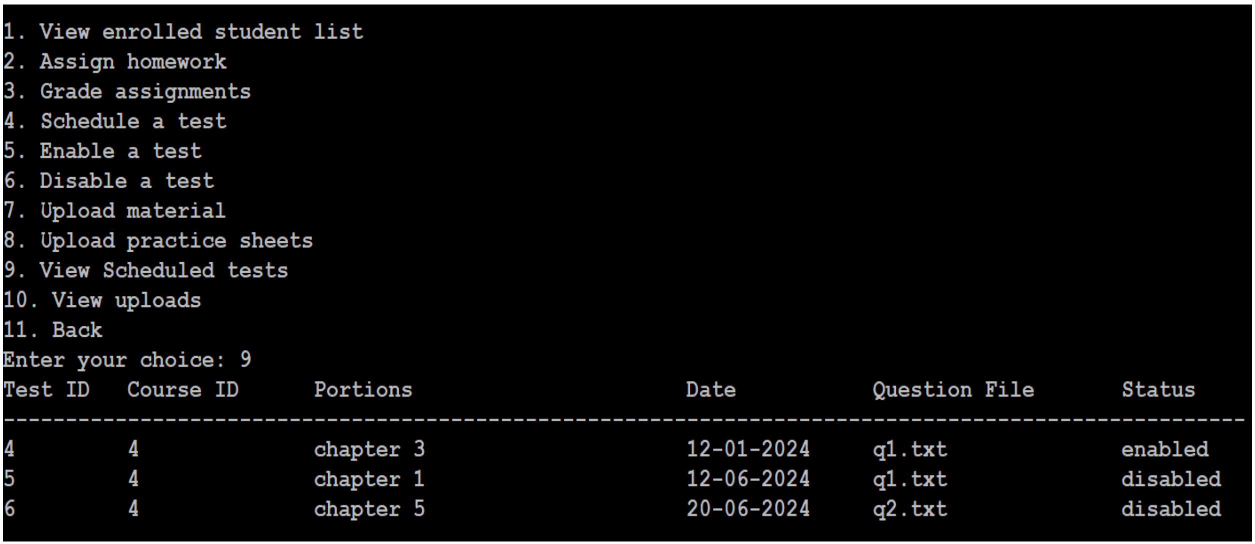


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 9
Test ID   Course ID      Portions                    Date          Question File       Status
-----------------------------------------------------------------------------------------------
4         4              chapter 3                   12-01-2024    q1.txt              disabled
5         4              chapter 1                   12-06-2024    q1.txt              disabled
6         4              chapter 5                   20-06-2024    q2.txt              disabled
```

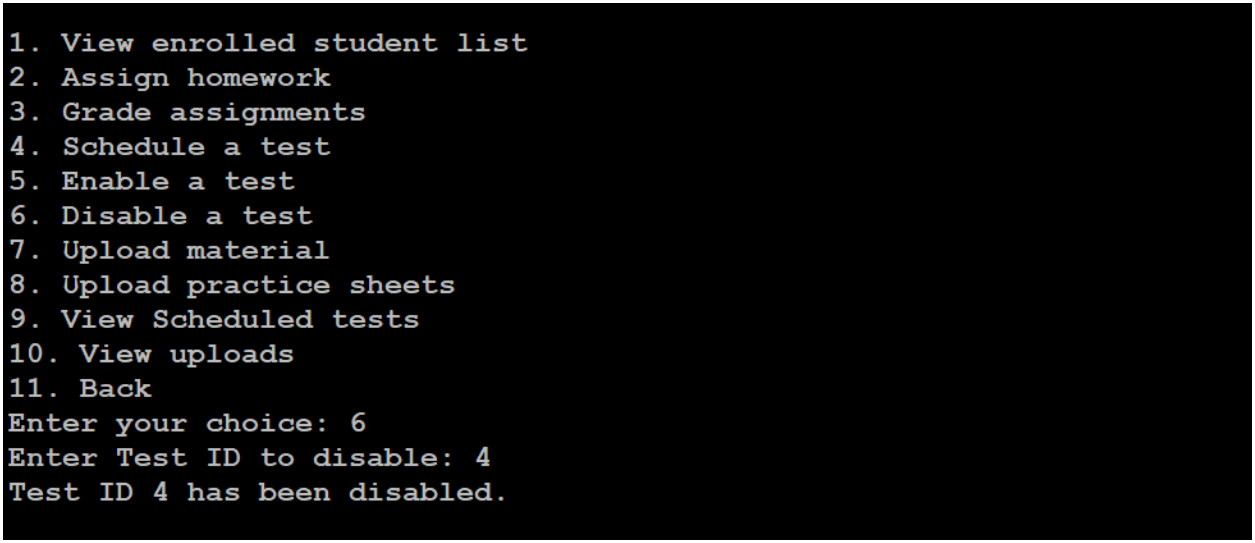


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 5
Enter Test ID to enable: 4
Test ID 4 has been enabled.
```

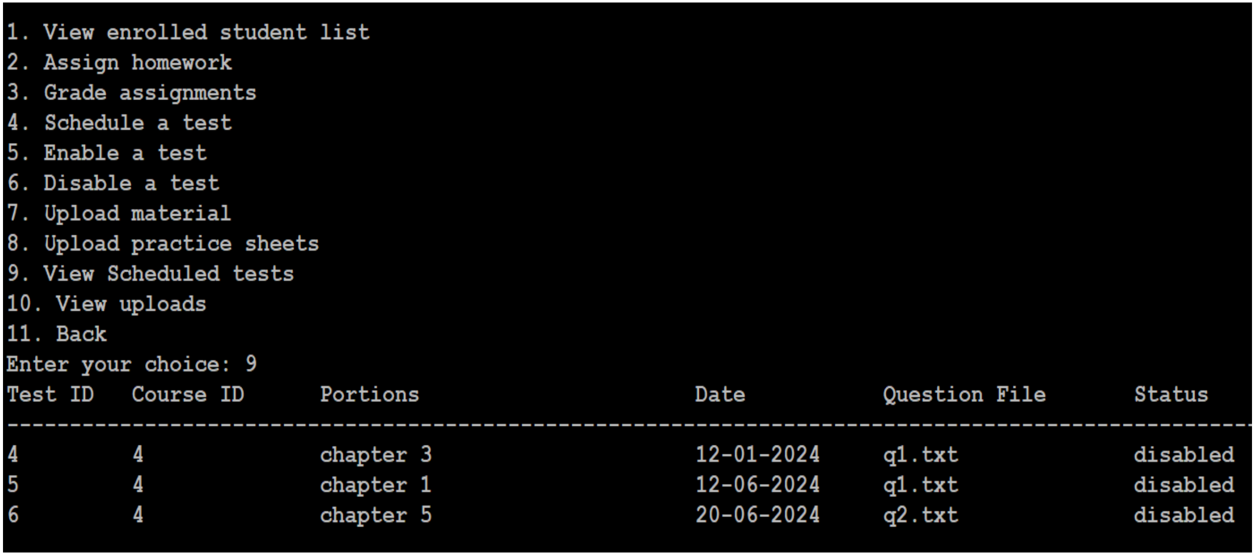
Disabling the tests (students cannot attempt them anymore):



```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 9
Test ID   Course ID      Portions                      Date          Question File       Status
----------------------------------------------------------------------------------------------------
4         4              chapter 3                     12-01-2024    q1.txt              enabled
5         4              chapter 1                     12-06-2024    q1.txt              disabled
6         4              chapter 5                     20-06-2024    q2.txt              disabled
```

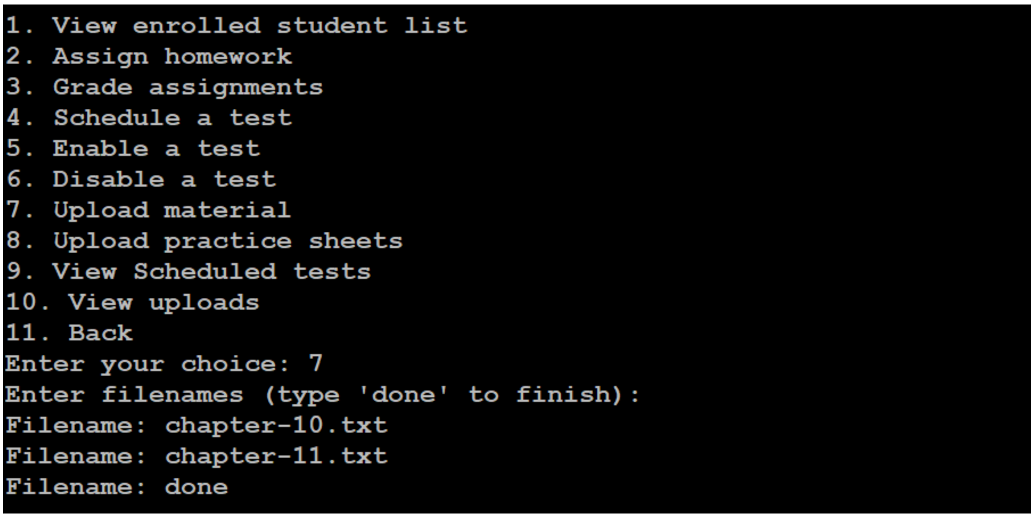


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 6
Enter Test ID to disable: 4
Test ID 4 has been disabled.
```

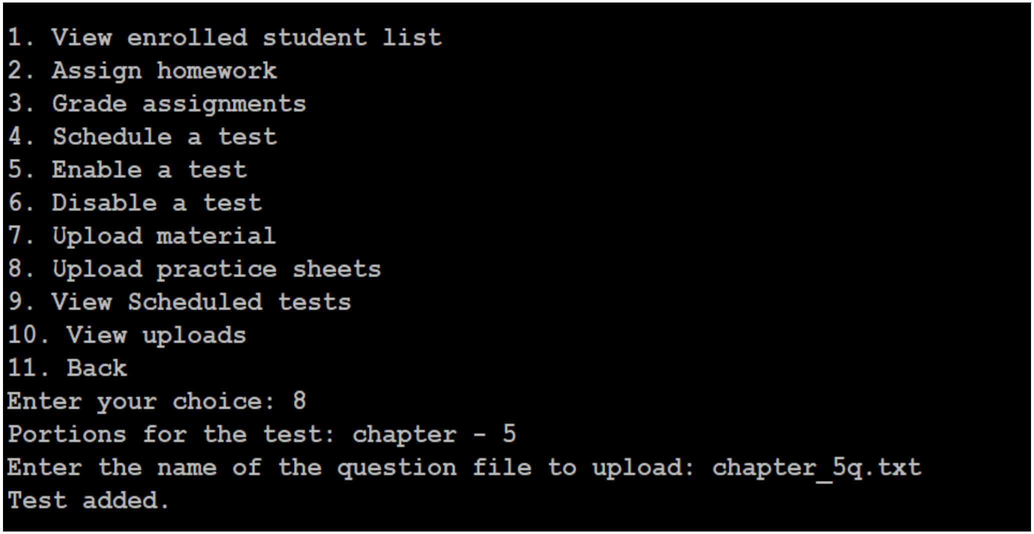


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 9
Test ID   Course ID      Portions                      Date          Question File       Status
----------------------------------------------------------------------------------------------------
4         4              chapter 3                     12-01-2024    q1.txt              disabled
5         4              chapter 1                     12-06-2024    q1.txt              disabled
6         4              chapter 5                     20-06-2024    q2.txt              disabled
```

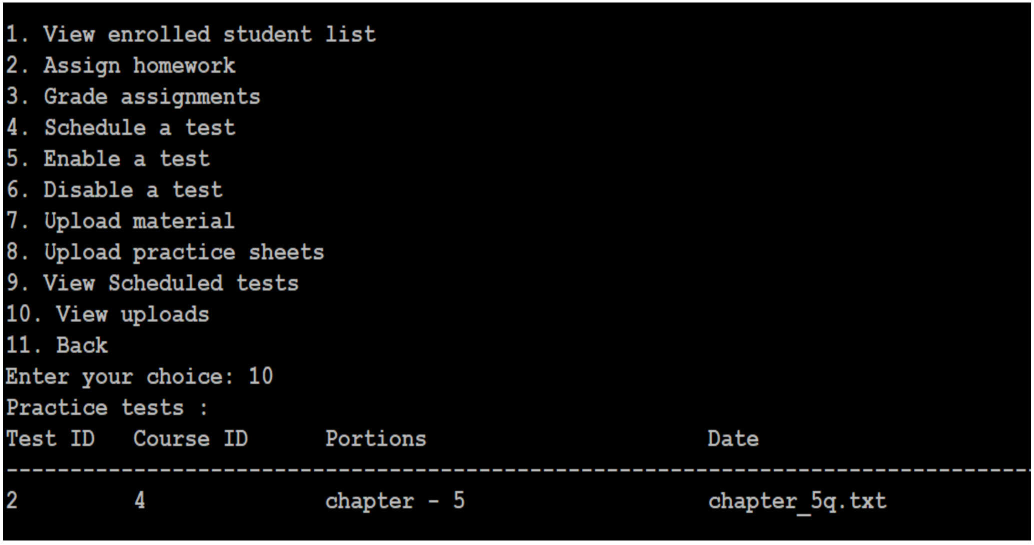
Uploading materials and practice tests, viewing the uploaded practice tests:



```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 7
Enter filenames (type 'done' to finish):
Filename: chapter-10.txt
Filename: chapter-11.txt
Filename: done
```

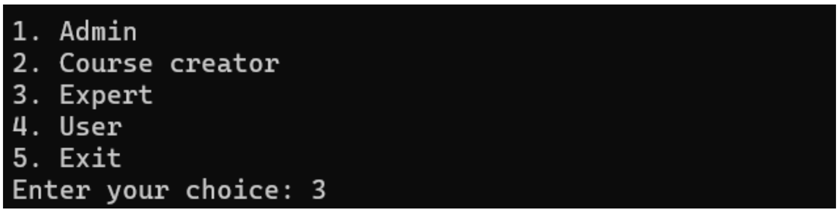


```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 8
Portions for the test: chapter - 5
Enter the name of the question file to upload: chapter_5q.txt
Test added.
```



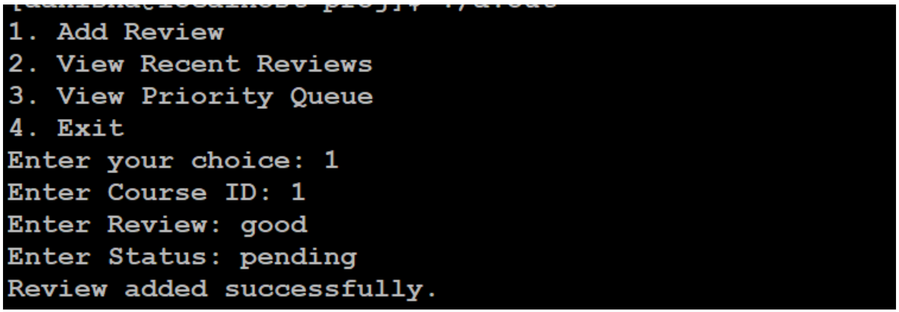
```
1. View enrolled student list
2. Assign homework
3. Grade assignments
4. Schedule a test
5. Enable a test
6. Disable a test
7. Upload material
8. Upload practice sheets
9. View Scheduled tests
10. View uploads
11. Back
Enter your choice: 10
Practice tests :
Test ID   Course ID      Portions                      Date
--------------------------------------------------------------------------------
2         4              chapter - 5                   chapter_5q.txt
```

Expert’s functionalities once they login:

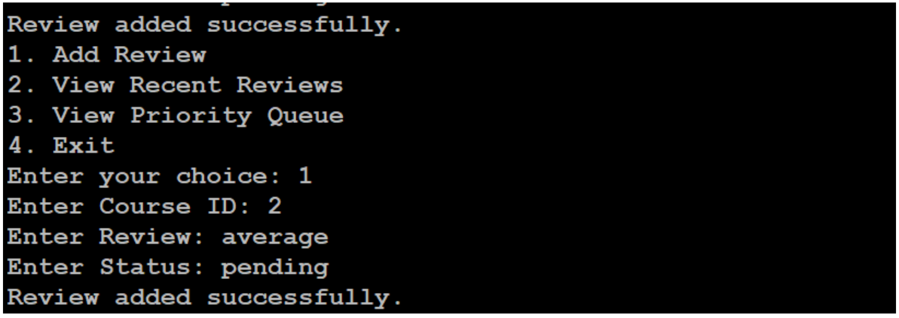


```
1. Admin
2. Course creator
3. Expert
4. User
5. Exit
Enter your choice: 3
```

Adding review for a particular course:

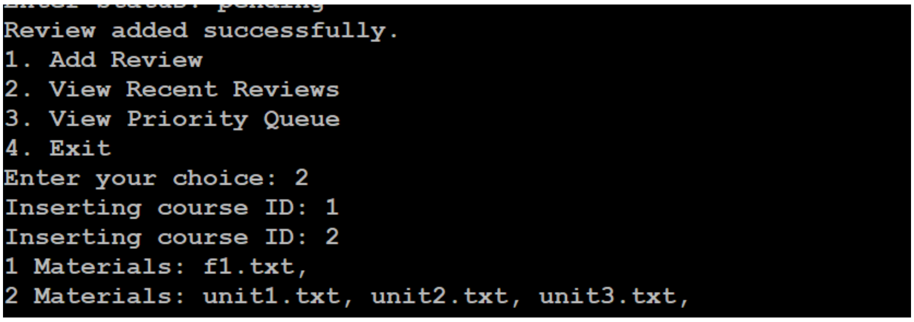


```
1. Add Review
2. View Recent Reviews
3. View Priority Queue
4. Exit
Enter your choice: 1
Enter Course ID: 1
Enter Review: good
Enter Status: pending
Review added successfully.
```



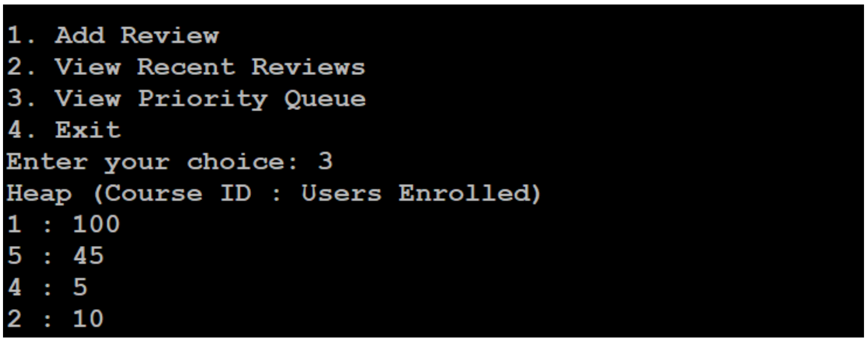
```
Review added successfully.
1. Add Review
2. View Recent Reviews
3. View Priority Queue
4. Exit
Enter your choice: 1
Enter Course ID: 2
Enter Review: average
Enter Status: pending
Review added successfully.
```

Viewing recently reviewed courses that which are still in pending for further review(s):



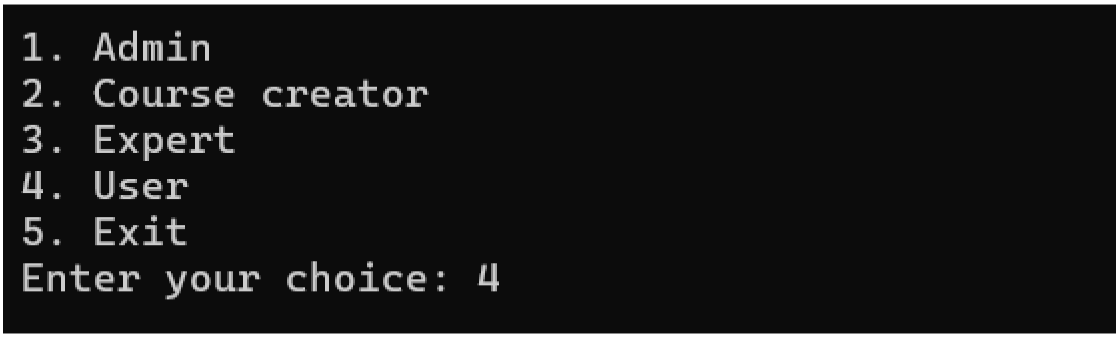
```
Review added successfully.
1. Add Review
2. View Recent Reviews
3. View Priority Queue
4. Exit
Enter your choice: 2
Inserting course ID: 1
Inserting course ID: 2
1 Materials: f1.txt,
2 Materials: unit1.txt, unit2.txt, unit3.txt,
```

Seeing the order of priority for reviewing the courses:



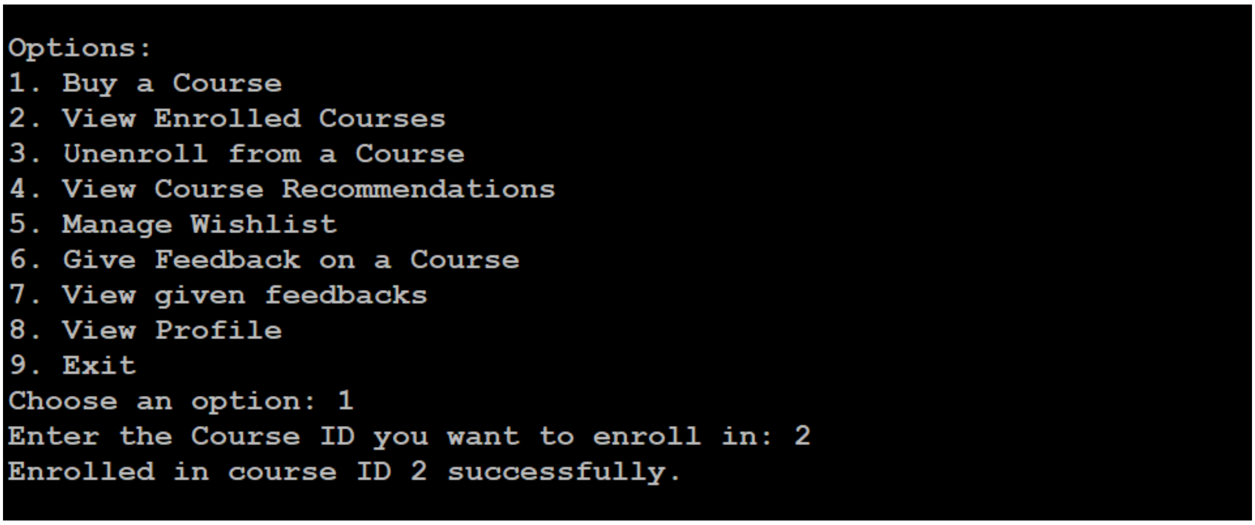
```
1. Add Review
2. View Recent Reviews
3. View Priority Queue
4. Exit
Enter your choice: 3
Heap (Course ID : Users Enrolled)
1 : 100
5 : 45
4 : 5
2 : 10
```

**User’s functionalities once they login:**



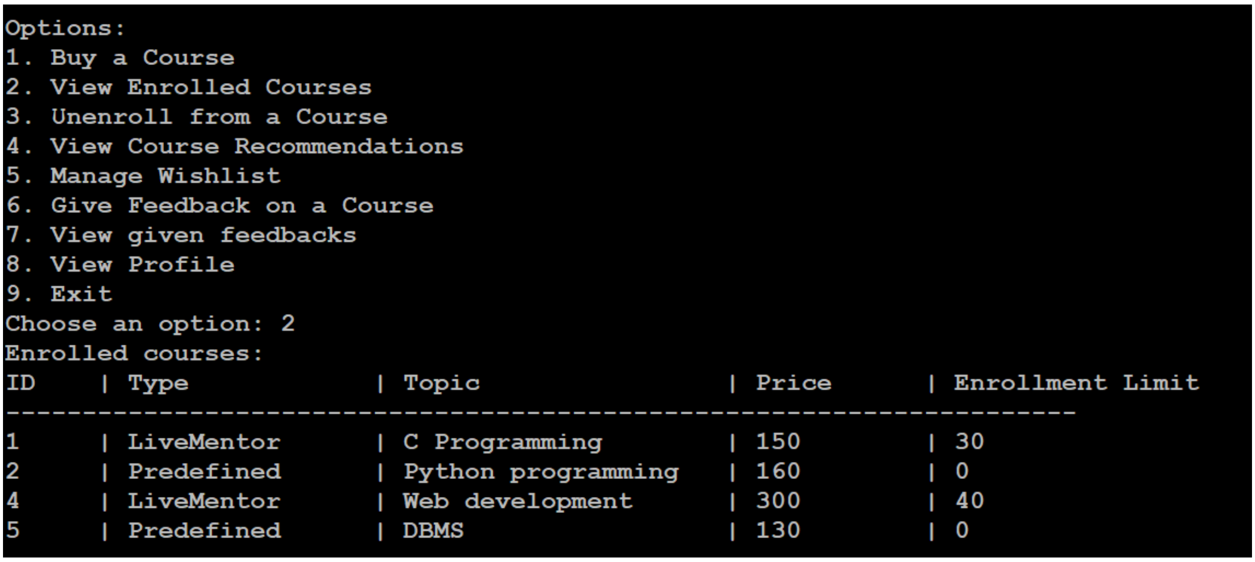
```
1. Admin
2. Course creator
3. Expert
4. User
5. Exit
Enter your choice: 4
```

Enrolling in a course:



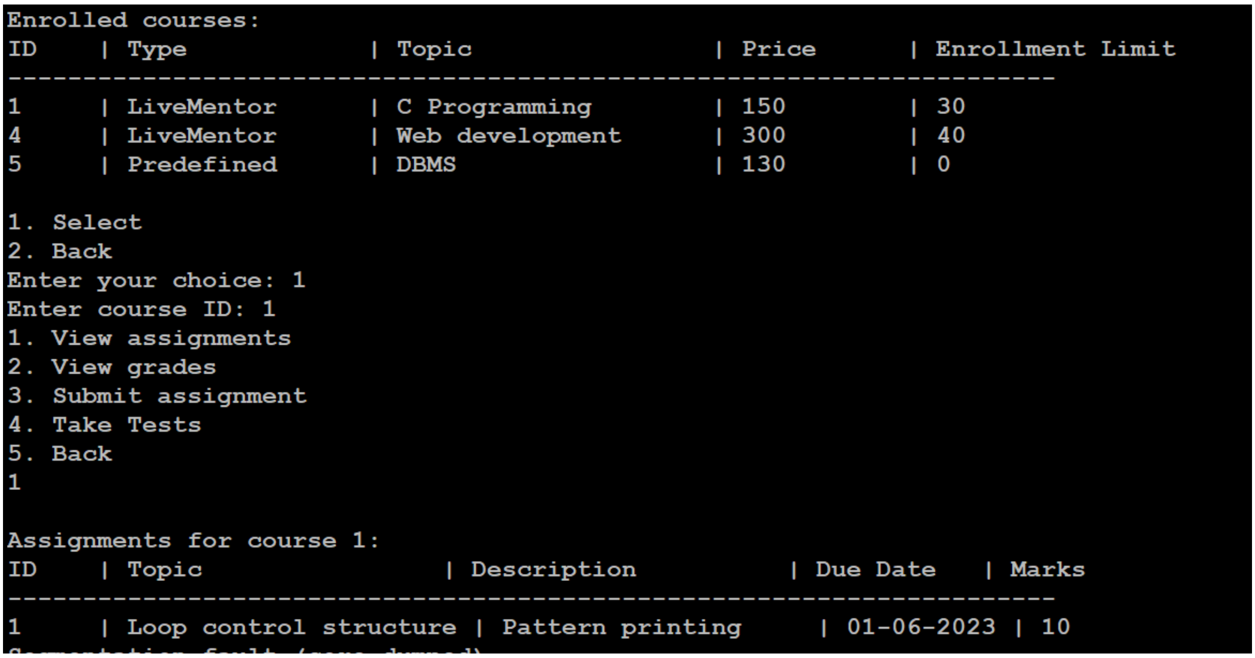
```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 1
Enter the Course ID you want to enroll in: 2
Enrolled in course ID 2 successfully.
```

Viewing the enrolled courses:



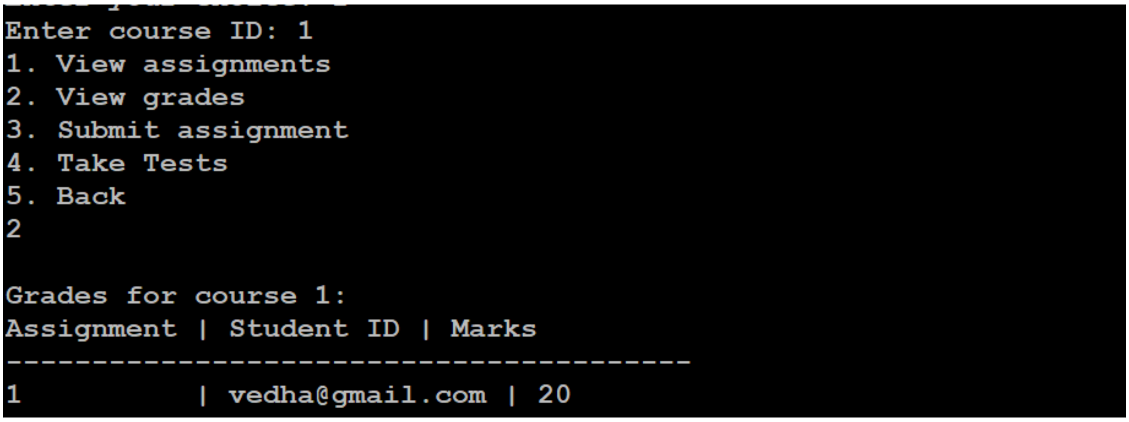
```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 2
Enrolled courses:
ID     | Type             | Topic                 | Price        | Enrollment Limit
----------------------------------------------------------------------------
1      | LiveMentor       | C Programming         | 150          | 30
2      | Predefined       | Python programming    | 160          | 0
4      | LiveMentor       | Web development       | 300          | 40
5      | Predefined       | DBMS                  | 130          | 0
```

Viewing assignments posted for a particular course:



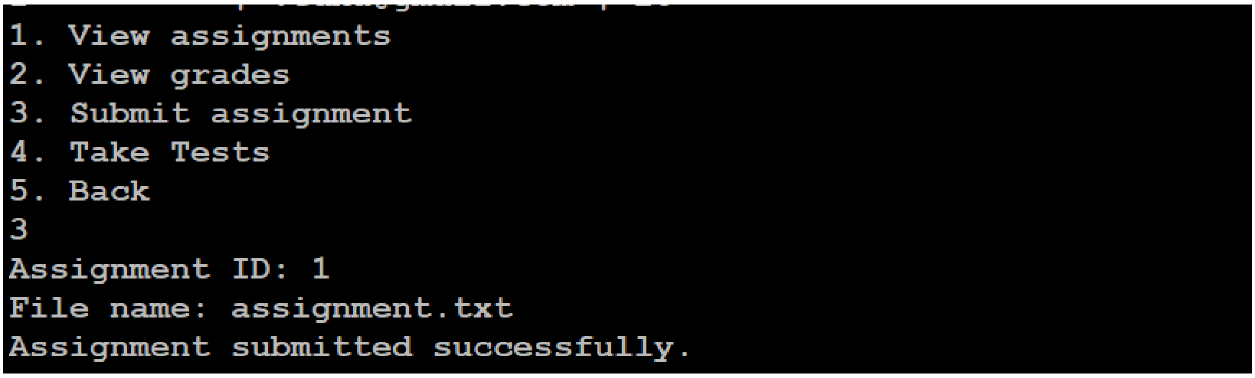
```
Enrolled courses:
ID     | Type             | Topic                 | Price        | Enrollment Limit
----------------------------------------------------------------------
1      | LiveMentor       | C Programming         | 150          | 30
4      | LiveMentor       | Web development       | 300          | 40
5      | Predefined       | DBMS                  | 130          | 0

1. Select
2. Back
Enter your choice: 1
Enter course ID: 1
1. View assignments
2. View grades
3. Submit assignment
4. Take Tests
5. Back
1

Assignments for course 1:
ID     | Topic                  | Description          | Due Date   | Marks
----------------------------------------------------------------------
1      | Loop control structure | Pattern printing     | 01-06-2023 | 10
Segmentation fault (core dumped)
```

Viewing grades



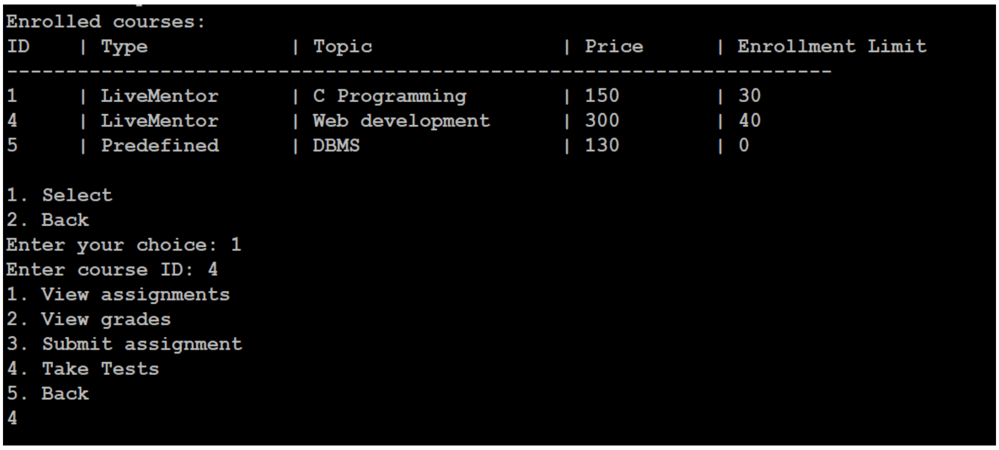
```
Enter course ID: 1
1. View assignments
2. View grades
3. Submit assignment
4. Take Tests
5. Back
2

Grades for course 1:
Assignment | Student ID | Marks
----------------------------------------
1          | vedha@gmail.com | 20
```

Assignment submission:



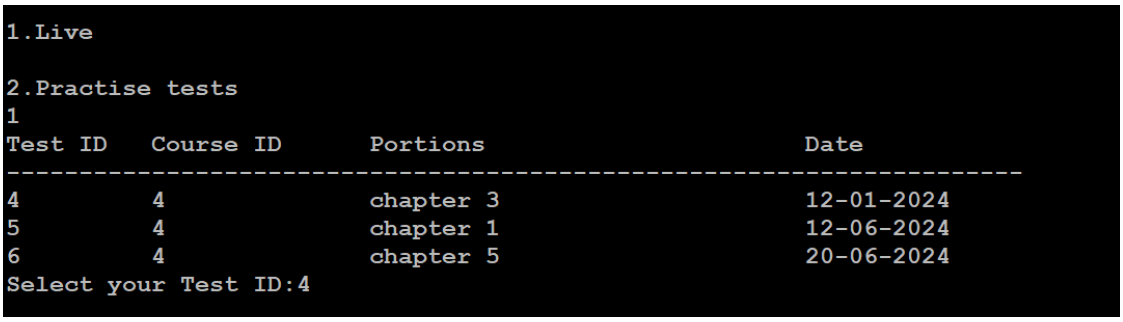
```
1. View assignments
2. View grades
3. Submit assignment
4. Take Tests
5. Back
3
Assignment ID: 1
File name: assignment.txt
Assignment submitted successfully.
```

Attempting tests:



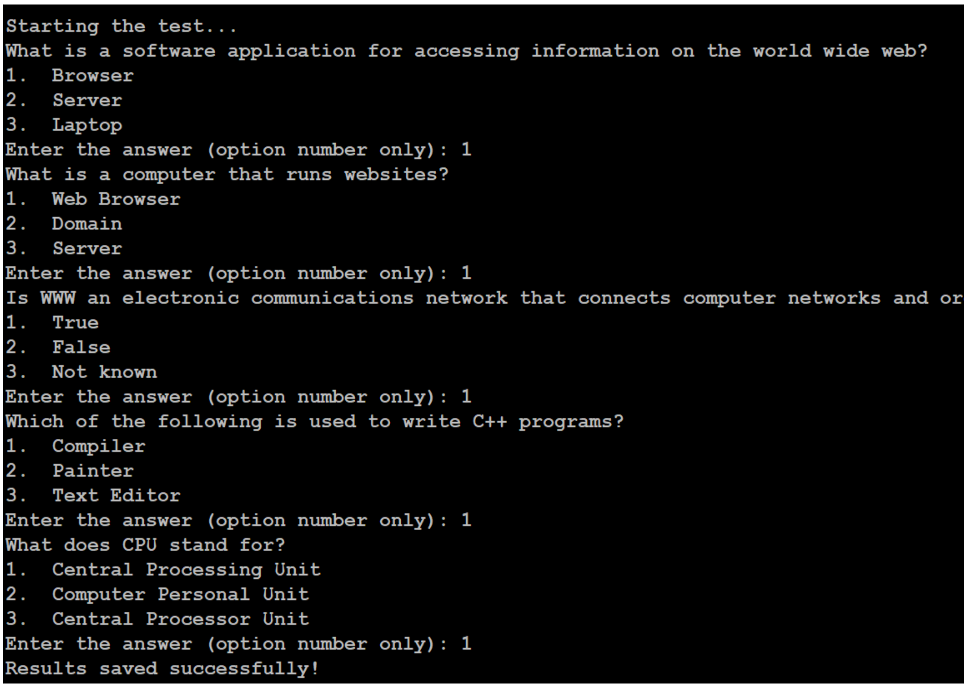
```
Enrolled courses:
ID     | Type            | Topic                 | Price       | Enrollment Limit
----------------------------------------------------------------------
1      | LiveMentor      | C Programming         | 150         | 30
4      | LiveMentor      | Web development       | 300         | 40
5      | Predefined      | DBMS                  | 130         | 0

1. Select
2. Back
Enter your choice: 1
Enter course ID: 4
1. View assignments
2. View grades
3. Submit assignment
4. Take Tests
5. Back
4
```

Viewing all live tests:



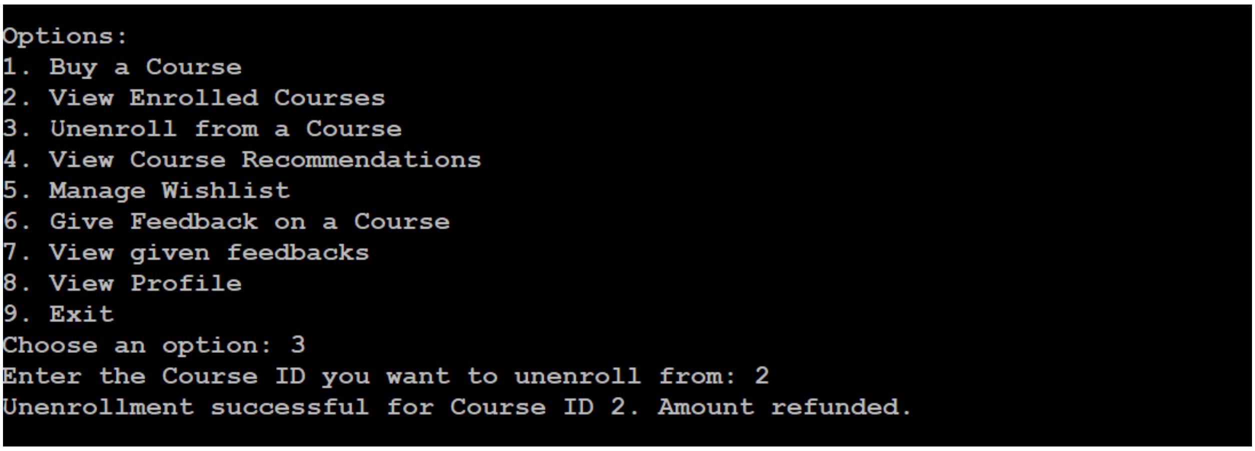
```
1.Live

2.Practise tests
1
Test ID   Course ID      Portions                     Date
---------------------------------------------------------------------
4         4              chapter 3                    12-01-2024
5         4              chapter 1                    12-06-2024
6         4              chapter 5                    20-06-2024
Select your Test ID:4
```

Taking a live test:



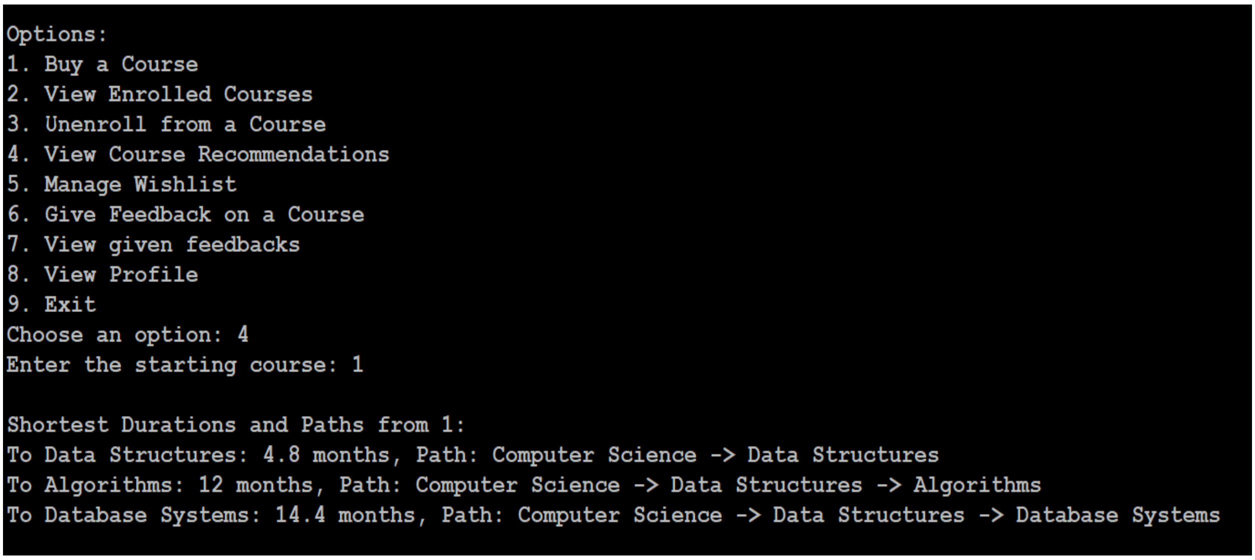
```
Starting the test...
What is a software application for accessing information on the world wide web?
1.  Browser
2.  Server
3.  Laptop
Enter the answer (option number only): 1
What is a computer that runs websites?
1.  Web Browser
2.  Domain
3.  Server
Enter the answer (option number only): 1
Is WWW an electronic communications network that connects computer networks and or
1.  True
2.  False
3.  Not known
Enter the answer (option number only): 1
Which of the following is used to write C++ programs?
1.  Compiler
2.  Painter
3.  Text Editor
Enter the answer (option number only): 1
What does CPU stand for?
1.  Central Processing Unit
2.  Computer Personal Unit
3.  Central Processor Unit
Enter the answer (option number only): 1
Results saved successfully!
```

Unenrolling from a course:



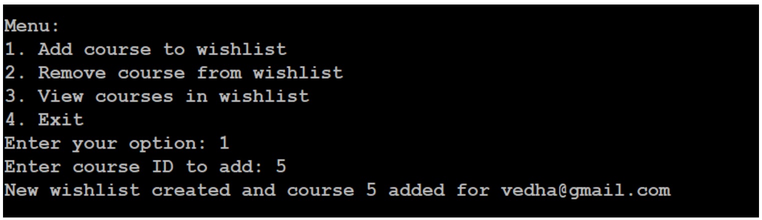
```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 3
Enter the Course ID you want to unenroll from: 2
Unenrollment successful for Course ID 2. Amount refunded.
```

Viewing course recommendations:

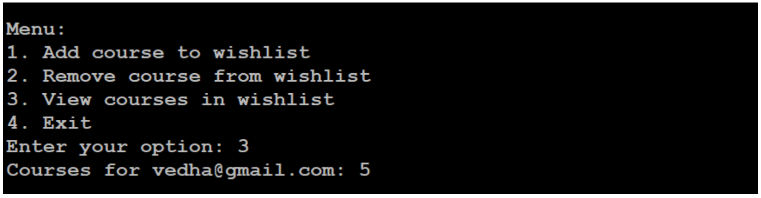


```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 4
Enter the starting course: 1

Shortest Durations and Paths from 1:
To Data Structures: 4.8 months, Path: Computer Science -> Data Structures
To Algorithms: 12 months, Path: Computer Science -> Data Structures -> Algorithms
To Database Systems: 14.4 months, Path: Computer Science -> Data Structures -> Database Systems
```

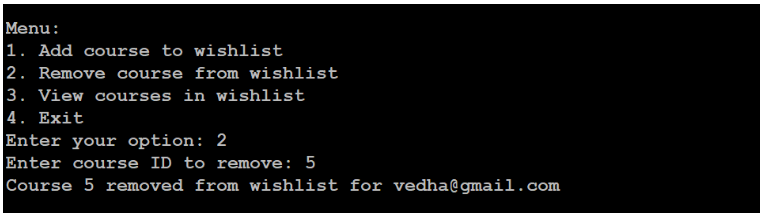
Wishlist management:



```
Menu:
1. Add course to wishlist
2. Remove course from wishlist
3. View courses in wishlist
4. Exit
Enter your option: 1
Enter course ID to add: 5
New wishlist created and course 5 added for vedha@gmail.com
```

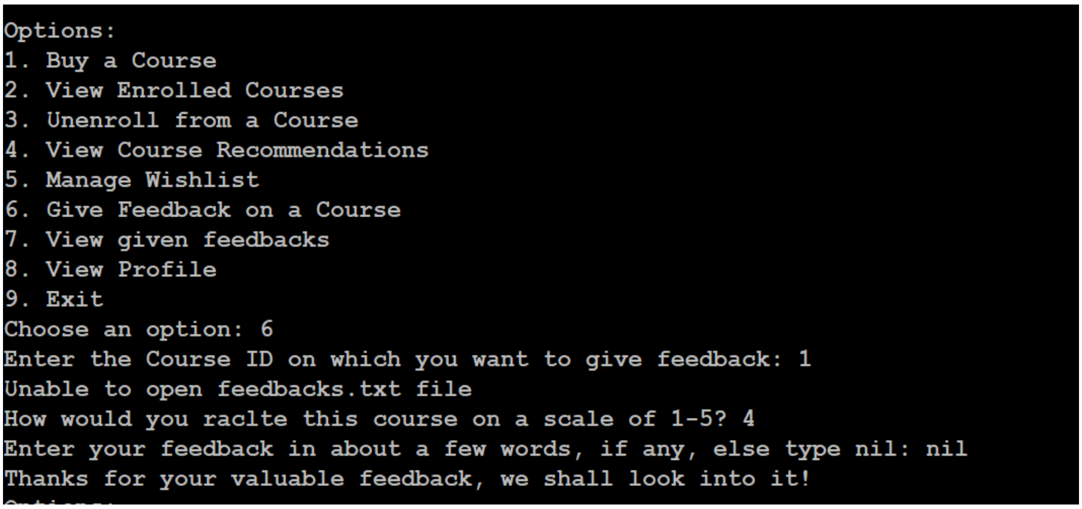


```
Menu:
1. Add course to wishlist
2. Remove course from wishlist
3. View courses in wishlist
4. Exit
Enter your option: 3
Courses for vedha@gmail.com: 5
```



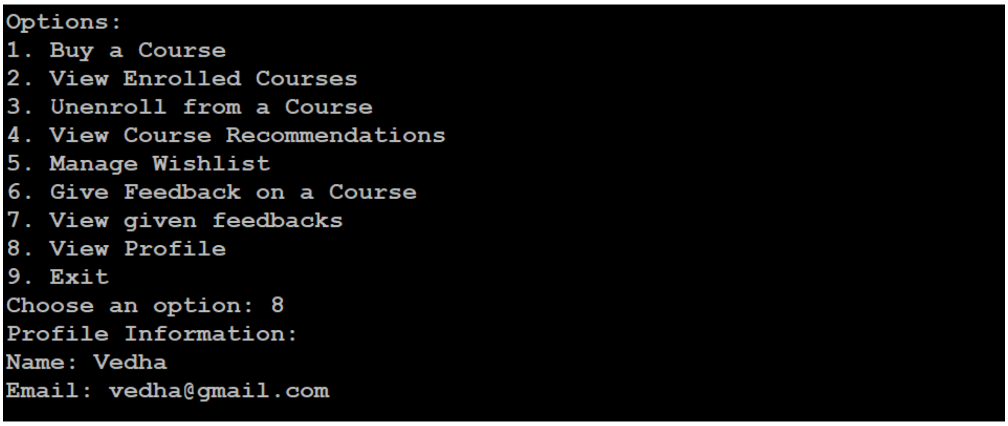
```
Menu:
1. Add course to wishlist
2. Remove course from wishlist
3. View courses in wishlist
4. Exit
Enter your option: 2
Enter course ID to remove: 5
Course 5 removed from wishlist for vedha@gmail.com
```

Giving feedback:



```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 6
Enter the Course ID on which you want to give feedback: 1
Unable to open feedbacks.txt file
How would you raclte this course on a scale of 1-5? 4
Enter your feedback in about a few words, if any, else type nil: nil
Thanks for your valuable feedback, we shall look into it!
```

Viewing user profile:



```
Options:
1. Buy a Course
2. View Enrolled Courses
3. Unenroll from a Course
4. View Course Recommendations
5. Manage Wishlist
6. Give Feedback on a Course
7. View given feedbacks
8. View Profile
9. Exit
Choose an option: 8
Profile Information:
Name: Vedha
Email: vedha@gmail.com
```

PS: The above attached output screenshots cover the resultant outcome of almost all the functionalities that are implemented through the system, and are almost in the same order as the order in which the menus have been nested.

## NOVELTY OF THE PROJECT

- The Online Learning Portal Management System developed by our team offers several novel features and approaches that distinguish it from existing online learning platforms as follows:
- **Terminal-Based Interaction**: Unlike traditional online learning platforms with graphical user interfaces (GUI), our system operates entirely through terminal-based interactions. This minimalist approach simplifies usage, reduces system resource requirements, and enhances accessibility, particularly for users with limited bandwidth or older hardware.

- **Dynamic Pricing Strategies**: The integration of graph-based dynamic pricing strategies using algorithms such as Bellman-Ford and Floyd Warshall sets our system apart. This feature allows for adaptive pricing based on various factors, including course popularity, user demand, and market trends, providing a more flexible and responsive pricing model.
- **Integrated Marketing Strategies**: Our system incorporates topological sorting of marketing strategies, enabling administrators to organize and visualize various marketing campaigns based on their effectiveness and interdependencies. This feature enhances strategic planning and decision-making, optimizing the impact of marketing efforts.
- **Personalized Course Recommendations**: Leveraging graph algorithms like Dijkstra's and Minimum Spanning Tree (Prim's Algorithm), our system delivers personalized course recommendations to users based on their enrolled courses and preferences. This personalized approach enhances user engagement and satisfaction by offering relevant and tailored learning opportunities.

## SOCIETAL USAGE OF THE PROJECT

- The societal impact of our Online Learning Portal Management System extends beyond its technical capabilities as follows:
- **Increased Accessibility**: By providing a terminal-based interface and optimizing system performance, our platform extends access to online education to users in resource-constrained environments, remote areas, and regions with limited internet connectivity.
- **Affordable Learning Opportunities**: Through dynamic pricing strategies and efficient course management tools, our system enables educational institutions and content creators to offer affordable and accessible learning opportunities, thereby democratizing access to quality education.
- **Skill Development and Empowerment**: By facilitating online learning and skill development, our platform empowers individuals to pursue lifelong learning, upskill or reskill, and enhance their employability and socioeconomic opportunities.

## FINAL OUTCOMES OF THE PROJECT

- The successful development and deployment of our Online Learning Portal Management System have resulted in several key outcomes as follows:
- **Functional System**: A fully functional and feature-rich online learning platform capable of managing courses, users, transactions, and feedback efficiently through terminal-based interactions.

- **Scalable Architecture**: A modular and scalable system architecture that can accommodate future growth, additional features, and evolving user needs.
- **Enhanced Learning Experience**: Improved access to quality education, personalized learning opportunities, and enhanced user engagement, resulting in a more effective and enjoyable learning experience for users.

## POTENTIAL FUTURE ENHANCEMENTS

- While our Online Learning Portal Management System has achieved significant milestones, there are several avenues for future enhancements and development as follows:
- **Enhanced User Interface**: Exploration of options to improve the user interface while maintaining compatibility with terminal-based interactions, potentially incorporating text-based graphics or enhanced formatting.
- **Advanced Analytics**: Integration of data analytics and machine learning techniques to provide deeper insights into user behavior, learning patterns, and course effectiveness, enabling data-driven decision-making and personalized learning experiences.
- **Expanded Content Offerings**: Collaboration with content creators and educational institutions to expand the range of available courses, subjects, and learning materials, catering to diverse interests and learning objectives.
- **Mobile Compatibility**: Development of a mobile-friendly version or companion app to extend accessibility and usability to mobile device users, enabling learning on-the-go.
- **Social Integration**: Integration with social media platforms or online communities to facilitate collaboration, discussion, and knowledge sharing among users, fostering a more interactive and engaging learning environment.
- By pursuing these potential enhancements and continuing to innovate, our Online Learning Portal Management System can further solidify its position as a leading platform for online education, benefiting learners, educators, and institutions alike.

## CONCLUSION

- In conclusion, our proposed Online Learning Portal Management System aims to revolutionize the e-learning experience by fostering collaboration, embracing advanced technologies, and remaining agile in the face of evolving educational paradigms.