



PROJECT PROPOSAL

DATA STRUCTURES AND ALGORITHMS



GROUP MEMBERS

- | | | |
|----|-----------------------|---------------|
| 1. | MUHAMMAD ANAS | 02-134242-068 |
| 2. | MUHAMMAD ADIL SHABBIR | 02-134242-033 |
| 3. | AMJAD ELLAHI | 02-134242-092 |

BS(CS) 3-B

SUBMITTED TO: MISS RABIA AMJAD

COURSE REGISTRATION SYSTEM

PROJECT SCOPE:

A course registration system has broad application across universities, colleges, training programs, and corporate learning platforms. It streamlines course selection, enrollment, schedule management, and administrative operations through efficient data structures and algorithms.

PROJECT ABSTRACT:

Objective

Streamline course registration process within educational institutions.

Scope

- User registration
- Course browsing
- Enrollment management
- Administration
- Scheduling

Methodology

- Requirements gathering
- System design
- Implementation using **C++ and Data Structures with Algorithms**
- Documentation

Key Features

- User-friendly interfaces
- Real-time seat availability update

Expected Outcomes

- Enhanced efficiency and accessibility
- Improved stakeholder experience
- Scalability and reliability

PROJECT FUNCTIONALITIES (with Algorithms)

User Registration and Authentication

- **Hashing Algorithms:** For fast username/password lookup using **Hash Tables**.
- **String Matching (KMP/Rabin-Karp):** For validating credentials securely.

Course Browsing

- **Binary Search Algorithm:** To quickly search courses in a sorted list by course code/name.
- **Sorting Algorithms (Merge Sort/Quick Sort):** To sort courses alphabetically, by credit hours, or by availability.

Course Enrollment

- **Queue Implementation:** Waitlist handling for full courses.
- **AVL Trees / BSTs:** Efficient course storage and seat availability updates in logarithmic time.

Course Management (Admin side)

- **Insertion/Deletion in Trees:** Adding/removing courses efficiently.
- **Hash Maps:** For mapping course codes / course details.

User Profile Management

- **Linked List Traversal:** For storing and updating enrolled course history dynamically.

Payment Processing

- **Hash Maps:** Track transaction IDs and payment status.
- **Search Algorithms:** For verifying payment records.

Prerequisite Checking

- **Graph Traversal (DFS/BFS):** To check whether all prerequisite courses have been completed before enrollment.

Gradebook Integration

- **Balanced Trees / Hash Tables:** For quick retrieval and updates of student grades.
 - **Sorting Algorithms:** Rank students by grades (Quick Sort/Heap Sort).
-

DATA STRUCTURE & ALGORITHM IMPLEMENTATION

Arrays: Store list of available courses.

- **Algorithm:** Linear Search for small, unsorted lists.

Linked Lists: Store enrolled courses per student dynamically.

- **Algorithm:** Traversal and Insertion for updating course history.

Hash Tables: For fast lookups (students, courses, payments).

- **Algorithm:** Hashing (Chaining/Linear Probing).

Trees (BST, AVL Trees): Efficient searching, insertion, and deletion of courses.

- **Algorithm:** Binary Search Tree Operations + AVL Balancing.

Graphs: Model prerequisites and scheduling conflicts.

- **Algorithm:** Topological Sort, BFS/DFS Traversals.

Stacks & Queues:

- **Queue:** Manage course waitlists.
 - **Stack:** Undo/rollback last enrollment operation.
-

MODULE DISTRIBUTION (with Algorithm Mapping)

User Management Module

- **User Registration:** Hashing Algorithm

- **Authentication:** String Matching
- **Profile Management:** Linked List Traversal

Course Management Module

- **Course Browsing:** Binary Search + Sorting
- **Course Enrollment:** AVL Trees + Queues
- **Prerequisite Checking:** Graph Traversal (DFS/BFS)

Administration Module

- **Course Management (Add/Delete):** Tree Insertion/Deletion
- **Enrollment Management:** Queue Handling
- **Payment Processing:** Hash Table Lookup

Integration Module

- **Grade Book:** Sorting (Heap Sort) + Hash Map storage

Payment Processing Module

- **Payment Gateway Integration:** Hash Maps for Transactions
- **Payment Status Management:** Searching Algorithms

Documentation Module

- User Manual
- Technical Documentation

Teacher Remarks: _____