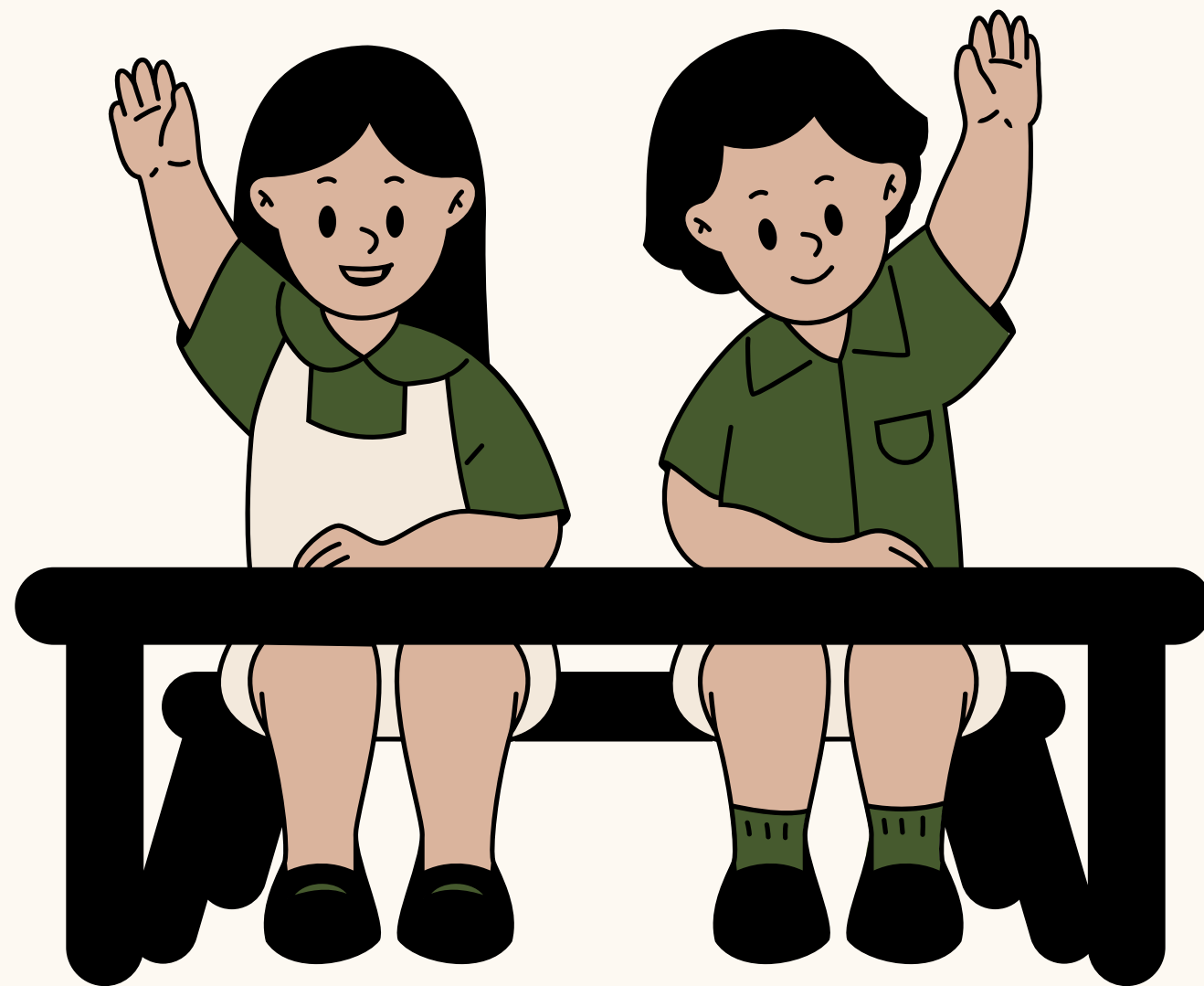


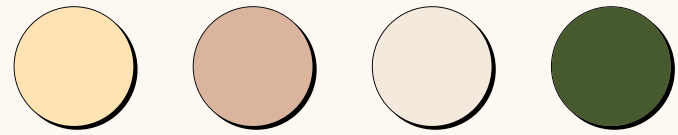
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

21CSC201J – Data Structures and Algorithm



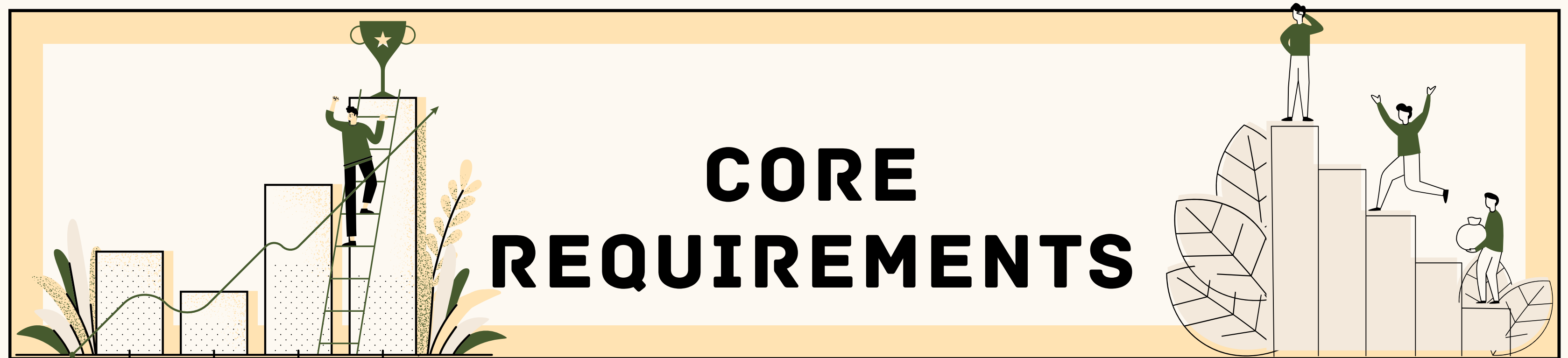
STUDENT ENROLLMENT SYSTEM

Sai Tulasi Asrithra Sri
(RA2311027010110)



PROBLEM **UNDERSTANDING**

- In an academic institution, efficiently managing student enrollments is crucial for maintaining organized records and supporting real-time access to student information.
- The system should allow seamless additions of new students, quick search capabilities by student ID, and ensure records are sorted for easy report generation.



OUR FIRST GOAL

Allow insertion of new students in real-time as they enroll.

OUR SECOND GOAL

Enable easy and quick access to student records, particularly by student ID.

OUR THIRD GOAL

Display the students enrolled in a specific course and the courses a specific student is enrolled in.

DATA STRUCTURE SELECTION



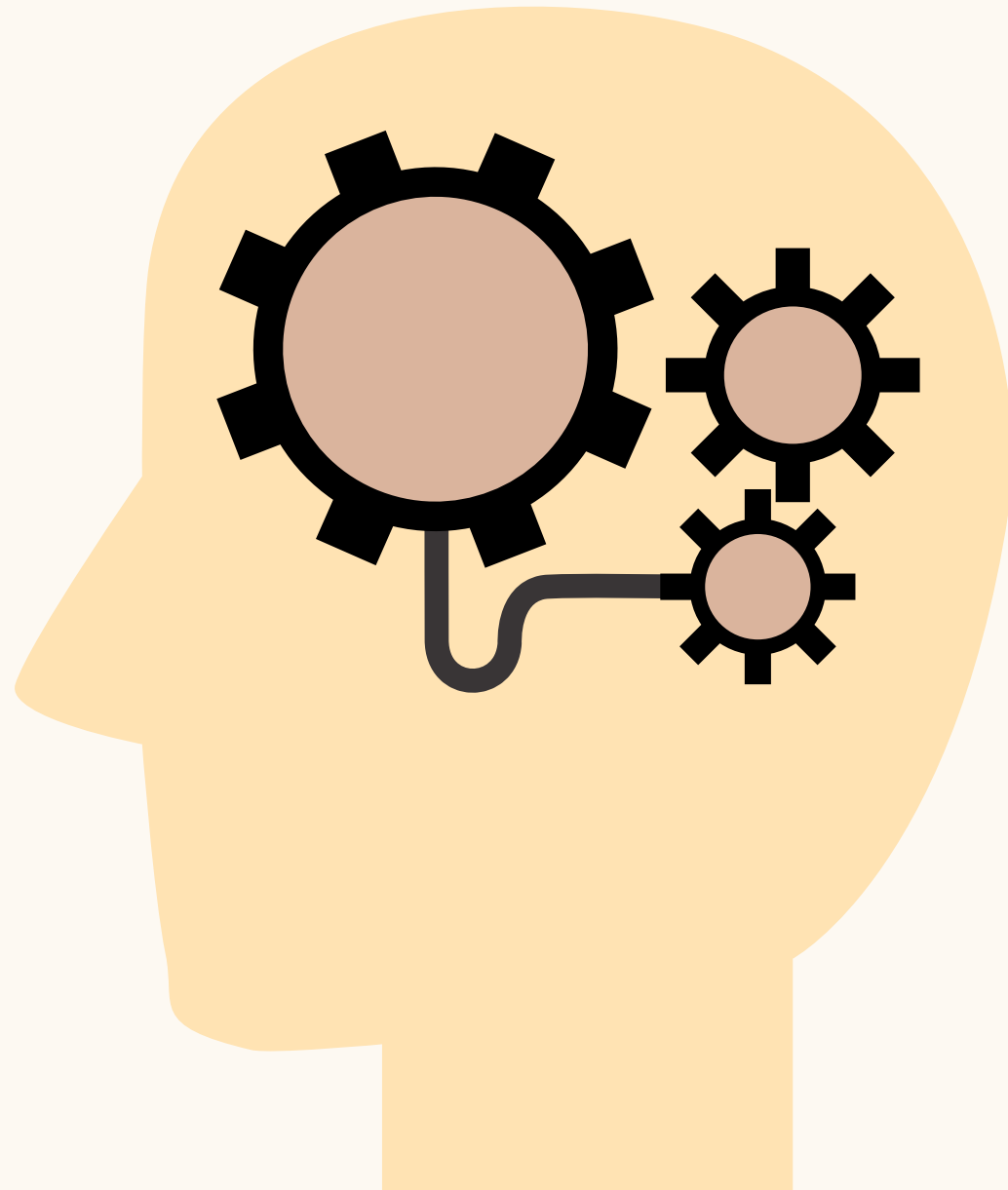
CHOOSEN DATA STRUCTURE

Tree (Binary Search Tree).

REASON

- A BST allows efficient organization of student records based on a key (e.g., student ID).
- The BST structure enables fast lookups, insertions, and deletions in an organized manner.
- It maintains sorted order, making easy to retrieve students data.

OPERATIONS ON DATA STRUCTURE



ADD A STUDENT/COURSE

Insert in sorted order or insert and then sort.

FIND STUDENT/COURSE

Use binary search to locate the student or course by ID.

ENROLL STUDENT IN COURSE

First, locate both the student and the course using binary search, then add the relationship between them.

DELETE STUDENT OR COURSE

Remove a specific student or course record by ID from the BST.

EFFICIENCY AND JUSTIFICATION

TIME COMPLEXITY

Add Student (Insert Operation):
 $O(\log n)$

Find Student (Search Operation):
 $O(\log n)$

Remove Student (Delete Operation):
 $O(\log n)$



SPACE COMPLEXITY

$O(n)$, where n is the total number of students. The BST uses memory space linearly with the number of enrolled students.



JUSTIFICATION



DYNAMIC DATA HANDLING

The BST structure can efficiently accommodate the frequent addition and removal of student or course records, allowing the system to adapt to changing enrollment data without performance degradation.

UNIQUE KEY MANAGEMENT

A BST structure ensures that each student or course ID is unique, preventing duplicate entries, which is vital for data integrity in an enrollment system.

SIMPLICITY

Implementing a BST is relatively straightforward in programming, making it a practical choice for developing a Student Enrollment System



THANK YOU

