Project Report

Team members: -

Ahmed Ali Ahmed: 42310488 (C1)

Ibrahim Saied Mohamed: 42310037 (C2)

Mohamed Abbas :42310258 (C1)

Youssef Ahmed: 42310385 (C2)

Khaled Hussein: 42320015 (C3)

INSTRUCTOR: DR. Daila Sayed

Project Report: Simple Student Management System

Introduction: -

The project is designed to implement a **Student Management System** in C++ that manages data for students, administrators, doctors, and assistants. The system uses various data structures such as linked lists, stacks, and binary trees to organize and process information efficiently. Key functionalities include managing applicants, tracking accepted and rejected students, and handling course and salary details for doctors and assistants.

Techniques Used and Their Purpose: -

1. **Linked List**
    - **Reason**: Used for dynamic and sequential storage of data such as admin IDs, student lists, and queue operations. The linked list allows flexible memory usage and ease in insertion/deletion operations.

2. **Queue**
    - **Reason**: Used to manage the sequence of applicants in the system. The queue structure ensures a first-in-first-out (FIFO) processing of student applications.

3. **Stack**
    - **Reason**: Utilized for handling salary details of doctors and assistants. The stack structure allows efficient addition and

removal of salary entries with a last-in-first-out (LIFO) approach.

4. **Binary Tree**
    - **Reason**: Implemented for organizing courses based on their attributes, such as payment rates. The binary tree enables efficient search and traversal of course data.

Functions Overview: -

1. **Admin Functions**
    - **Create_Admins_List**: Builds a linked list of admin IDs.
    - **Display_Admins_List**: Displays all admin IDs in the linked list.
    - **Search_Admins_List**: Searches for a specific admin ID in the linked list.
    - **Display_Admin**: Authenticates an admin by verifying ID and password.

2. **Applicants Functions**
    - **enqueue**: Adds a new applicant to the queue.
    - **dequeue**: Removes and processes the first applicant in the queue.
    - **displayQueue**: Displays the list of all applicants in the queue.

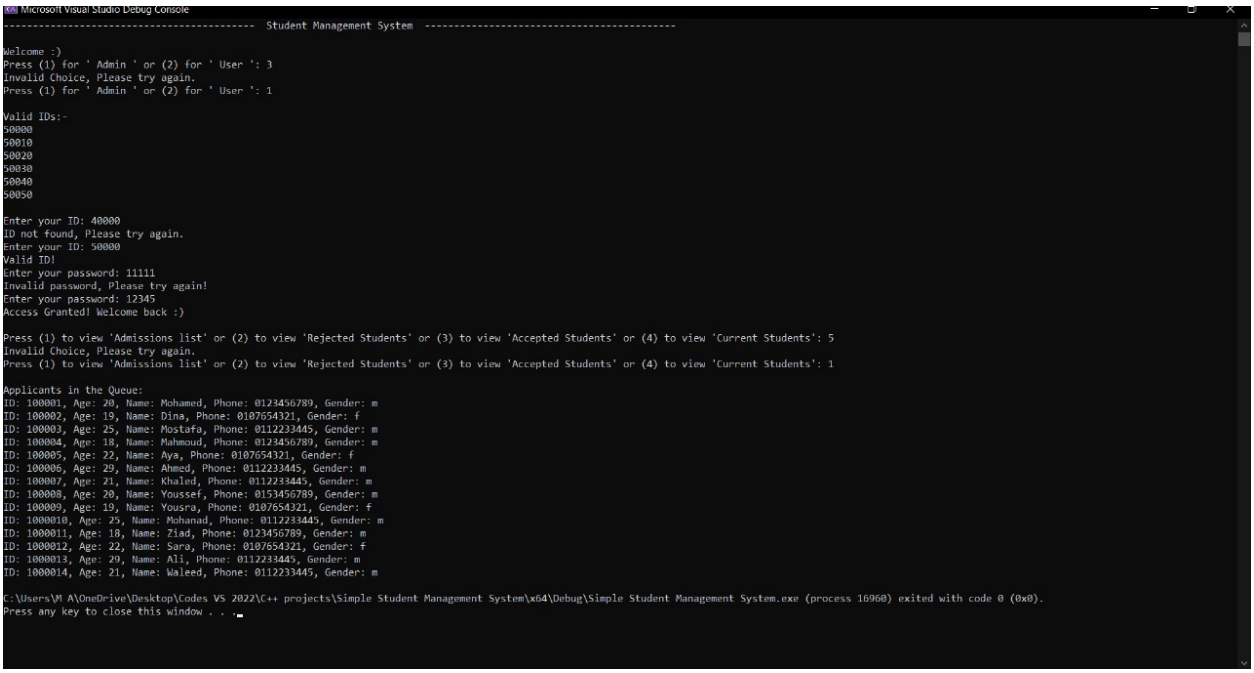
3. **Applicants Status Functions**

- **insert**: Adds a student to the accepted or rejected list (linked list).
- **display**: Displays the details of students in a specific list.

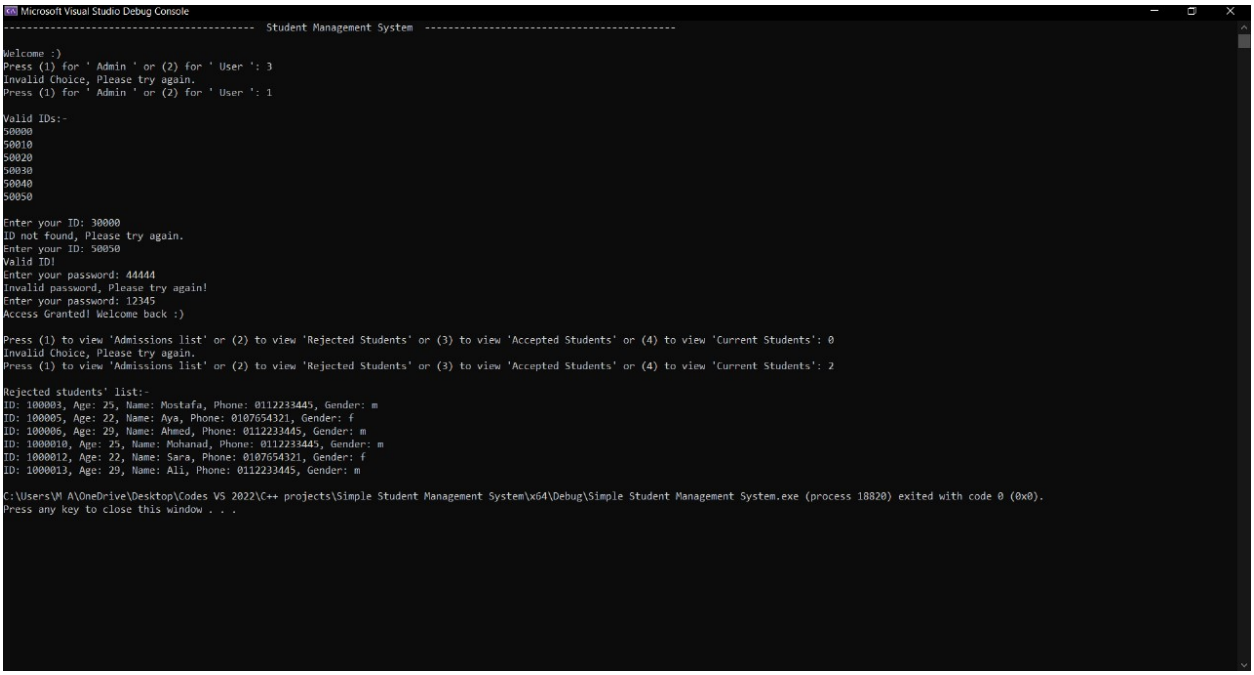
4. **Doctor and Assistant Functions**
    - **addCourses_Doctors / addCourses_Assistants**: Adds courses to a binary tree.
    - **InorderDisplayCourses_Doctors / InorderDisplayCourses_Assistants**: Displays all courses in an in-order traversal.
    - **pushSalary_Doctors / pushSalary_Assistants**: Adds salary details to a stack.
    - **popSalary_Doctors / popSalary_Assistants**: Removes the latest salary entry from the stack.
    - **calculateTotalSalary_Doctors / calculateTotalSalary_Assistants**: Computes the total salary based on hours and rate.
    - **displaySalaryDetails_Doctors / displaySalaryDetails_Assistants**: Displays the total salary details.

Sample Outputs with all possible options: -

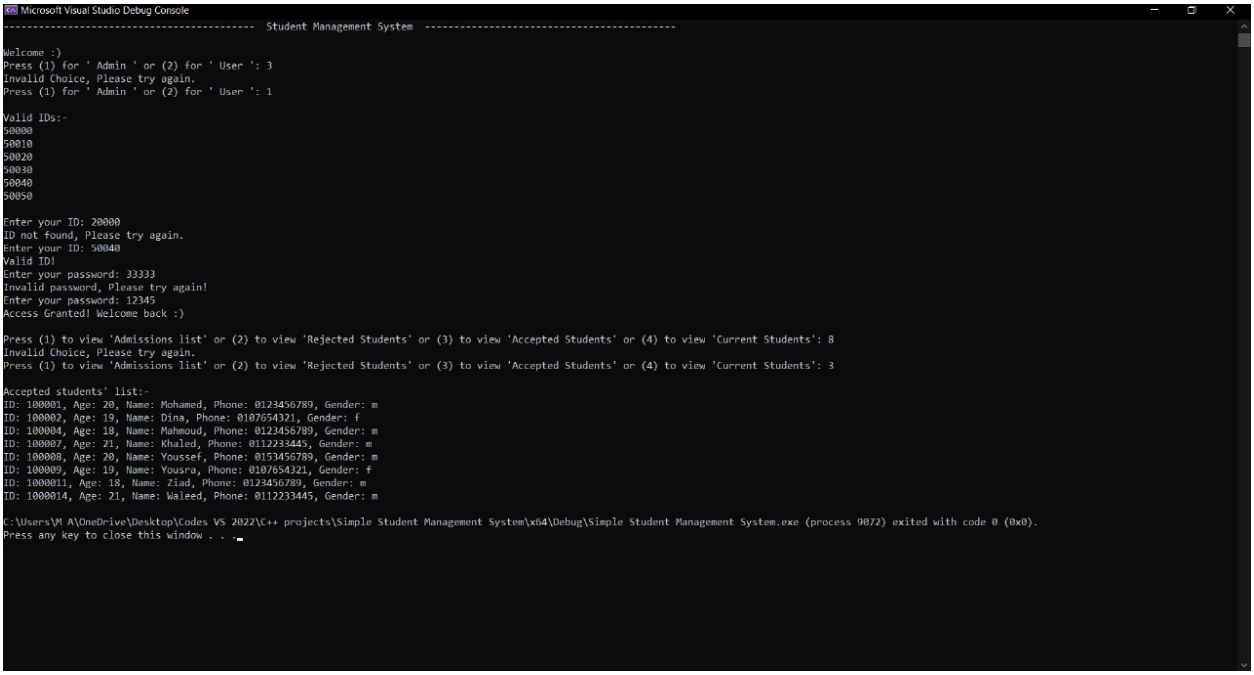
Admin: first possibility



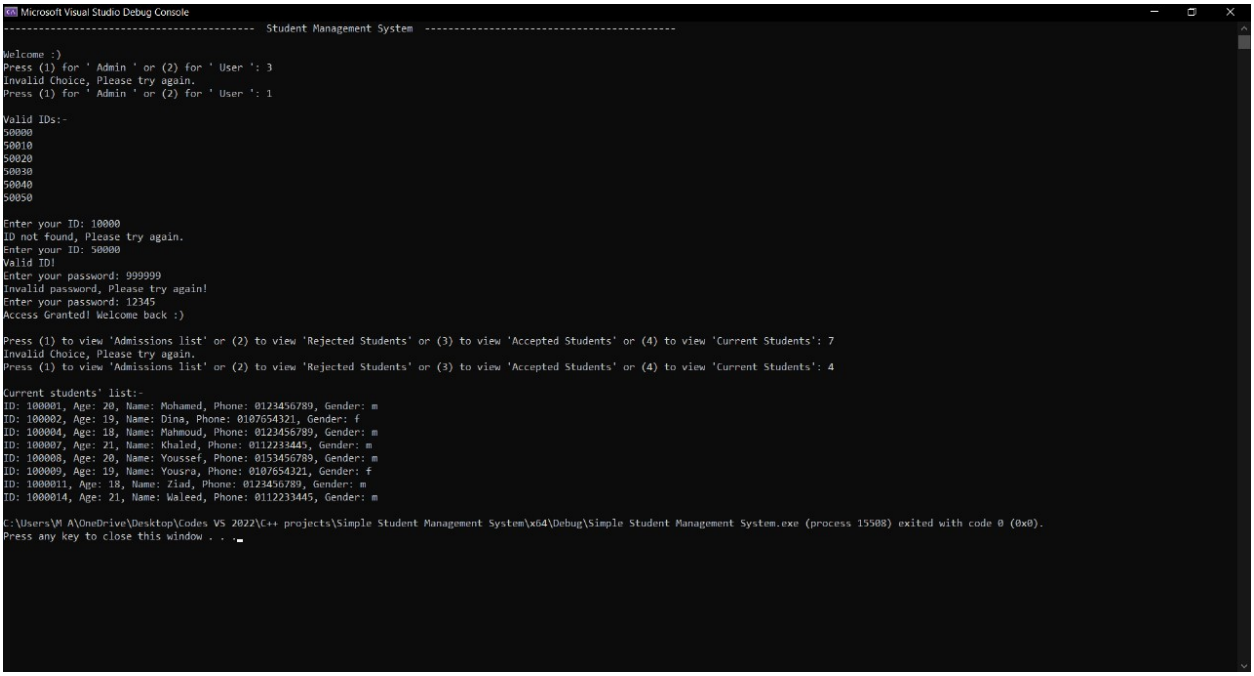
Admin: second possibility



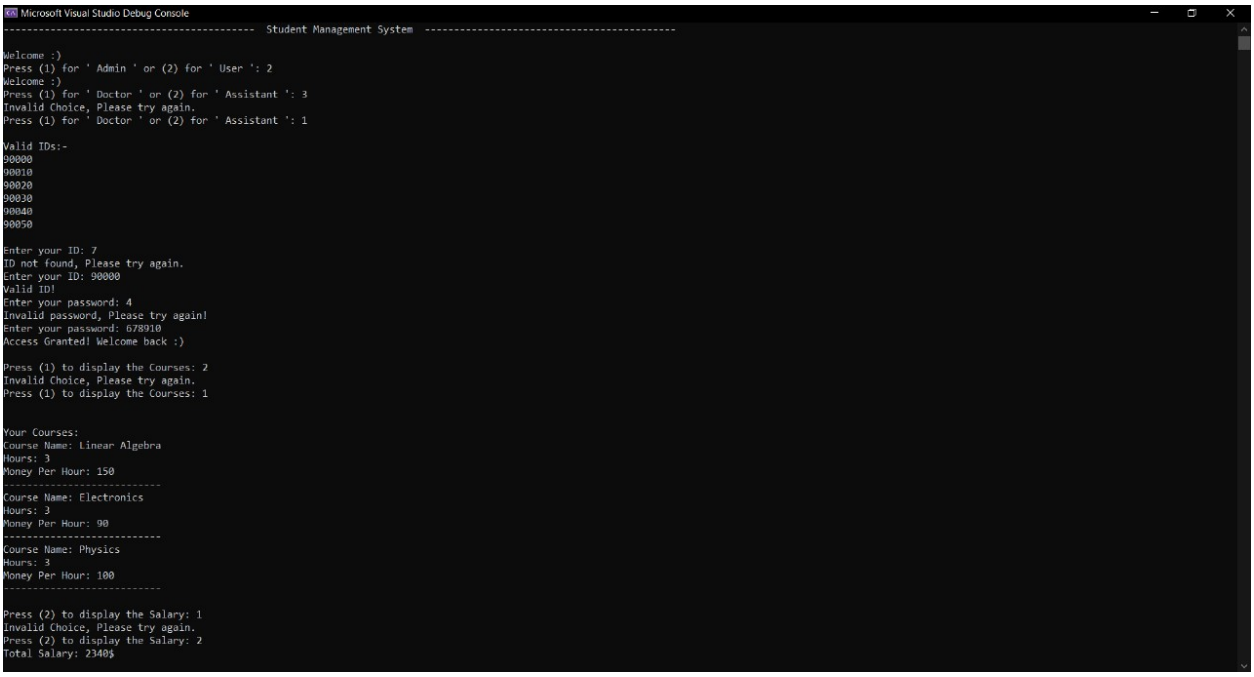
Admin: third possibility



Admin: fourth possibility



User: Doctor



User: Assistant