*ASSIGNMENT NO:-9*

# **Student Admission System**

*submitted in partial fulfillment of the requirements for completion of SE LAB of*

TY COMP *in*

Computer Engineering *by*

Onkar Bhagwat : 612203017

*Department of Computer Engineering*

A logo of a university

Description automatically generated

Department of Computer Engineering,

COEP Technological University (COEP Tech)

(A Unitary Public University of Govt. of Maharashtra)

Shivajinagar, Pune-411005,Maharashtra, INDIA

October, 2024

**Abstract:**

The **Student Admission and Branch Allocation System** is designed to streamline the admission process by automatically allocating seats to students based on their CET percentile and category. The system uses a priority queue to manage students' rankings, ensuring that those with higher CET scores get prioritized for seat allocation. Additionally, the system enforces seat quotas for different categories, ensuring fairness in the distribution of seats. The project integrates functionality for CSV-based data input, making it easier to handle large numbers of student applications.

**Features:**

1. **Priority Queue for Admission**:
   * The system employs a max-heap-based priority queue where students are inserted based on their CET percentile. This ensures that the student with the highest percentile is at the top of the queue, making the seat allocation process fair and efficient.
   * As each student is removed from the queue, they are considered for branch allocation based on their preferences and category.
2. **Branch Allocation with Seat Quotas**:
   * Each branch has a limited number of seats, which are further divided based on categories (Open, OBC, NT, SC, ST). The system ensures that students are allocated seats according to their category quota, maintaining fairness.
   * Students can specify their first and second branch preferences. If the first preference is not available due to seat limitations, the system attempts to allocate them their second preference.
   * The system checks for seat availability in each category (Open, OBC, NT, SC, ST) before finalizing the allocation.
3. **CSV Input for Bulk Student Data**:
   * The system supports reading student data from a CSV file, simplifying the input process for large numbers of applications.
   * Each line of the CSV file contains information about the student’s name, CET percentile, category, and branch preferences. This data is automatically inserted into the priority queue.
4. **Dynamic Memory Management**:
   * The system dynamically manages memory for student data and branch allocations. It ensures efficient allocation and deallocation of memory as students are added to or removed from the system.
5. **Error Handling and Feedback**:
   * The system provides clear feedback to the user, such as notifications when the priority queue is full or when a student cannot be allocated a seat due to a lack of available seats in both their preferred branches.
   * It also checks for invalid inputs and handles errors gracefully, ensuring robustness.
6. **Interactive Console-Based Interface**:
   * Users can interact with the system through a simple console interface. The system allows users to:
     + Submit admission forms.
     + Allocate seats for all students in the queue.
     + Load student data from a CSV file.
     + Display the list of students allocated to each branch.

**System Requirements:**

* **Hardware**:
  + Processor: Intel i3 or higher.
  + RAM: 4 GB or more.
  + Hard Disk: 500 GB minimum.
  + Operating System: Windows, Linux, or macOS.
* **Software**:
  + Programming Language: C++ (with data structures like Priority Queue and Arrays).
  + IDE: Any C++ compatible IDE such as Visual Studio Code, JetBrains CLion, or Code::Blocks.
  + Tools: CSV reader for bulk student data input, file handling for input/output operations.

**Design:**

1. **Student Data Structure**:
   * The Student structure stores essential information such as the student’s name, CET percentile, category, and their branch preferences (first and second priority).
2. **Priority Queue**:
   * The PriorityQueue structure implements a max-heap, where students are inserted and sorted based on their CET percentile. The heap property is maintained during every insertion and removal, ensuring that the student with the highest percentile is always at the root.
3. **Branch Data Structure**:
   * The branch structure stores information about the available seats in each branch. It tracks the total number of seats and the number of seats allocated to each category (Open, OBC, NT, SC, ST).
   * The branch allocation function checks seat availability and assigns students based on their category. If seats are available in the open category, they are filled first; otherwise, category-specific seats are allocated.
4. **Heapify Operations**:
   * The system uses heapifyUp and heapifyDown functions to maintain the heap property in the priority queue during insertions and deletions. These operations ensure that students are always correctly prioritized based on their CET percentile.
5. **CSV Input Handling**:
   * The system reads student data from a CSV file where each line contains the student's details. These details are parsed and inserted into the priority queue, enabling bulk student processing without manual input for each student.

**Project Flow:**

1. **Admission Form Submission**:
   * Students can submit their admission forms either manually through the console or by reading from a CSV file. The form collects details like name, CET percentile, category, and branch preferences.
   * The system inserts each student into the priority queue based on their CET percentile, ensuring that high-scoring students get priority in seat allocation.
2. **Branch Allocation Process**:
   * The system removes students from the priority queue in descending order of their CET percentile.
   * For each student, the system attempts to allocate a seat in their first preferred branch. If no seat is available, the system tries to allocate the second preferred branch.
   * The allocation considers category quotas and ensures that the seat allocation adheres to the predefined rules for different categories.
3. **Display of Allocated Students**:
   * Once all students are processed, the system displays the list of students allocated to each branch, along with their name, CET percentile, and category.
4. **CSV File Input**:
   * The system can read student data directly from a CSV file, simplifying the process for handling multiple applications. This feature is useful when dealing with a large number of students.

**Future Enhancements:**

1. **Web-Based Front-End**:
   * Integrating a web-based user interface to allow students to submit forms online. This would make the system more accessible and user-friendly.
2. **Waiting List**:
   * Introducing a waiting list for students who cannot be allocated their preferred branch due to seat limitations. This would give them a chance to be allocated a seat if any seats become available later.
3. **Multiple College Handling**:
   * Expanding the system to handle admissions for multiple colleges, with a centralized system managing seat allocation across different institutions.
4. **Improved Error Handling**:
   * Enhancing error handling to manage edge cases such as duplicate entries, invalid categories, or corrupted CSV files.