|  |
| --- |
| STUDENT PORTFOLIO |
| **Name: ATHISH J M**  **Register Number: RA2311056010004**  **Section: AL1**  **Mail ID:** aj0094@srmist.edu.in  **Department: DATA SCIENCE AND BUSINESS SYSTEMS**  **Semester: 3** |
| Subject Title: Data Structures and Algorithm (21CSC201J) Handled By: Dr Rajkumar K. |
| **LinkedIn**: https://www.linkedin.com/in/athish-j-m-275374219?utm\_source=share&utm\_campaign=share\_via&utm\_content=profile&utm\_medium=ios\_app  **GitHub:**  https://github.com/AthishJM  **E-Lab Completion Status** |

|  |
| --- |
| **Lab Record Completion Status**    **All the 15 experiments were solved, run and executed successfully and took the completed sign from Faculty.** |

**Real-World Application in DSA: Ticket reservation using queue**

**Overview**

The Ticket Reservation System employs queue data structures to manage reservations efficiently, handling customer requests in a fair and orderly manner. This structure ensures optimal resource utilization and seamless processing of requests in real time.

**Queue Data Structures in Ticket Reservation**

1. Queue Representation

* Elements: Each element in the queue represents a customer.
* Attributes: Includes customer name, ticket type, and status (e.g., pending, reserved).

2. Operations:

* Enqueue: Adds a customer to the reservation queue.
* Dequeue: Processes the customer at the front of the queue, issuing their ticket.
* Peek: Views the next customer in line without removing them.

3. Algorithm Implementation:

* First Come, First Served: Ensures fairness by processing requests in the order received.
* Priority Queue: For scenarios with different priority levels (e.g., VIP customers).
* Dynamic Queue Management: Handles cancellations and real-time updates to ensure accurate processing.

**Key Components**

1. Array/Linked List-Based Queues

* Array: Fixed size, simple implementation, suitable for small-scale systems.
* Linked List: Flexible and ideal for handling an unknown or variable number of reservations.

2. Circular Queue

* Prevents memory wastage in static arrays by reusing available space when the queue wraps around.

3. Priority Queue

* Used to prioritize urgent or high-value ticket requests.

4. Dynamic Updates

* Incorporates real-time changes such as cancellations, new bookings, or modifications in customer details.

**Skills Demonstrated**

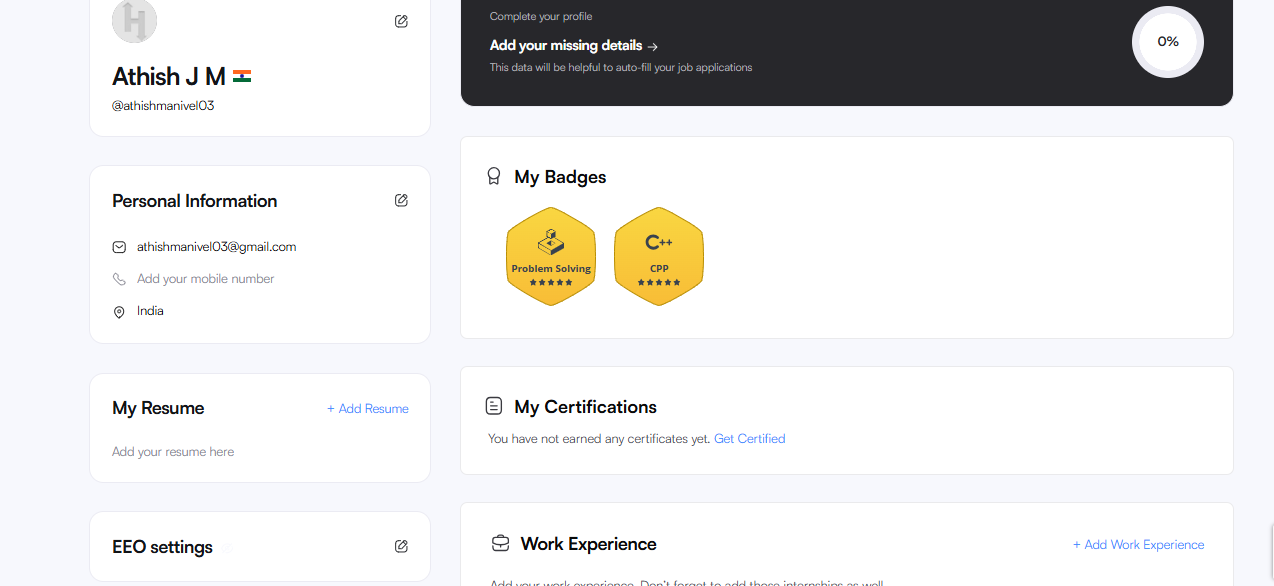
* Algorithm Design: Implementing tailored queue operations for efficient reservation handling.
* Data Structure Analysis: Choosing the right queue type for memory and processing efficiency.
* Programming Proficiency: Advanced queue manipulation for dynamic ticket allocation.
* User Interface: Creating intuitive input-output interfaces for users and operators.

**This approach highlights how queue-based systems provide scalable and reliable solutions for modern ticketing challenges.**

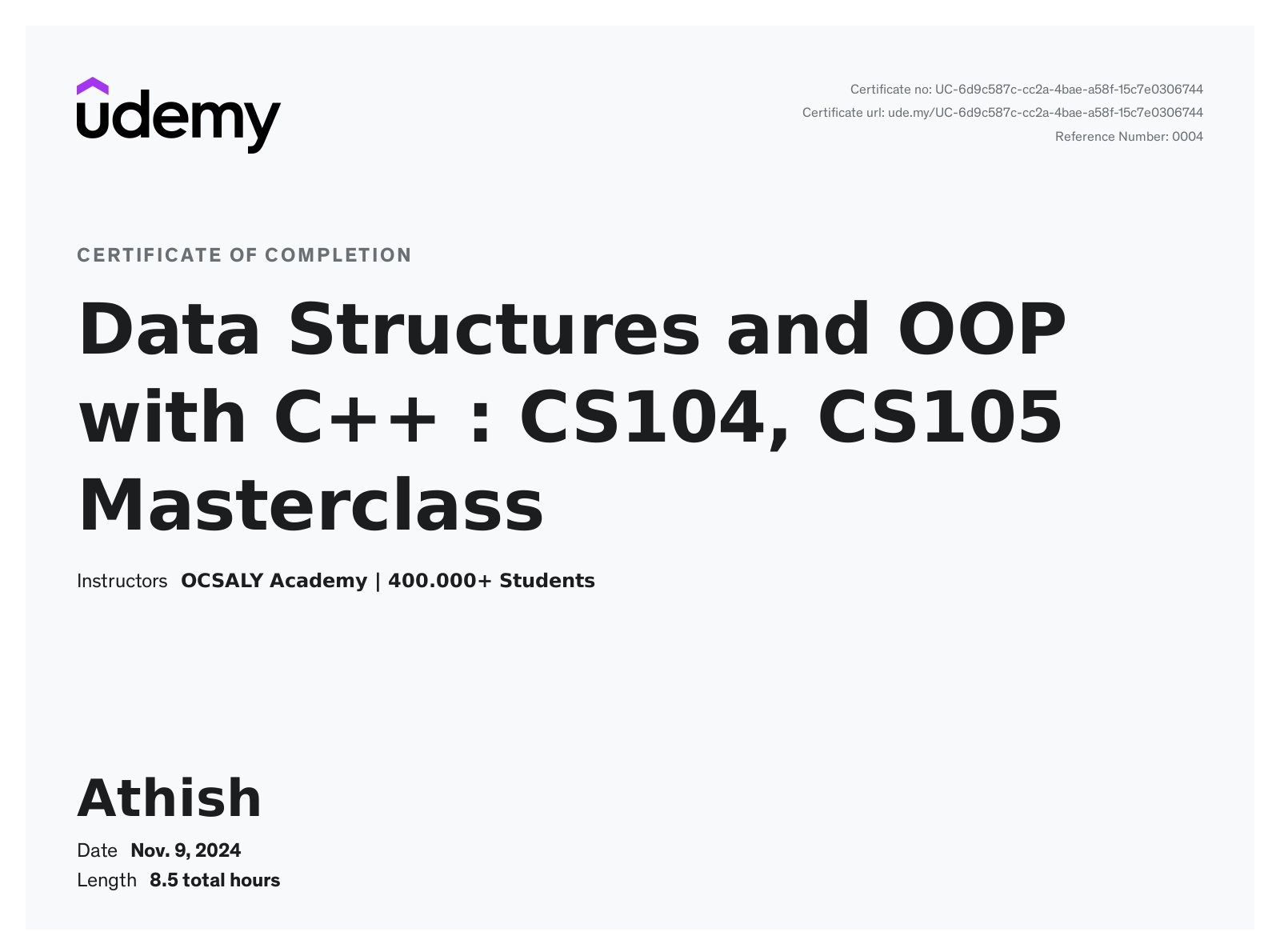
**GATE Questions Solution**

https://docs.google.com/document/d/1IwI4QiuQIIYTIoarLIERH9BRG3DcMwer/edit?usp=drive\_link&ouid=114553263009749379325&rtpof=true&sd=tr

**HackerRank ,Codechef and other certificates**

****



****

SIGNATURE OF THE STUDENT

