



Vivekanand Education Society's Institute Of Technology  
Department Of Information Technology  
DSA miniProject  
A.Y. 2025-26

Title: **IPO subscription simulator**

**Domain:** Data Structures and Algorithms  
(DSA)

**Member:** Om Sawlani

Mentor Name: Kajal Joseph

**1** NO  
POVERTY



**2** ZERO  
HUNGER



**3** GOOD HEALTH  
AND WELL-BEING



**4** QUALITY  
EDUCATION



**5** GENDER  
EQUALITY



**6** CLEAN WATER  
AND SANITATION



**7** AFFORDABLE AND  
CLEAN ENERGY



**8** DECENT WORK AND  
ECONOMIC GROWTH



**9** INDUSTRY, INNOVATION  
AND INFRASTRUCTURE



**10** REDUCED  
INEQUALITIES



**11** SUSTAINABLE CITIES  
AND COMMUNITIES



# THE GLOBAL GOALS

For Sustainable Development

**12** RESPONSIBLE  
CONSUMPTION  
AND PRODUCTION



**13** CLIMATE  
ACTION



**14** LIFE BELOW  
WATER



**15** LIFE  
ON LAND



**16** PEACE AND JUSTICE  
STRONG INSTITUTIONS



**17** PARTNERSHIPS  
FOR THE GOALS





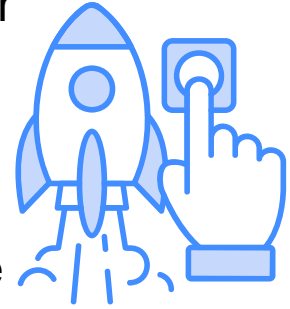
# Content

1. Introduction to the Project
2. Problem Statement
3. Objectives of the Project
4. Scope of the Project
5. Requirements of the System (Hardware, Software)
6. ER Diagram of the Proposed System
7. Data Structure & Concepts Used
8. Algorithm Explanation
9. Time and Space Complexity
10. Front End
11. Implementation
12. Gantt Chart
13. Test Cases
14. Challenges and Solutions
15. Future Scope
16. Code
17. Output Screenshots
18. Conclusion
19. References (in IEEE Format)



# Introduction to Project

In the financial world, **Initial Public Offerings (IPOs)** are a popular method for companies to raise capital by issuing shares to the public. Managing IPO applications involves **handling large amounts of investor data, validating it, and allotting shares fairly**. To simulate this process in a computer program, we can use **Data Structures such as Queue** (for application entry), **Stack** (for rejected applications), and **Linked List** (for final allotments). This project implements an **IPO Subscription Simulator** in **C**, providing a simplified yet effective demonstration of how applications can be processed





# Problem Statement

Traditional IPO allotment processes involve **manual record-keeping or complex online portals**, which may not be feasible for academic purposes. There is a need for a **DSA-based mini-project** that demonstrates how applications can be efficiently **stored, processed, and managed using fundamental data structures**.

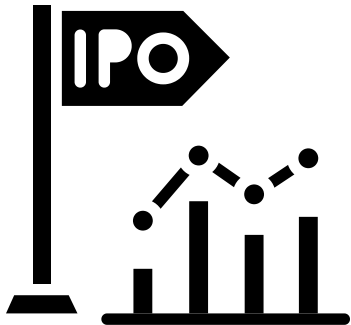
The problem is to **design and implement a simulator that accepts IPO applications, verifies them, allots shares to valid applicants, and maintains lists of successful and rejected investors**.





# Objectives of the project

- To simulate an IPO subscription system using stack, queue, and linked list.
- To process investor applications in FIFO order using queues.
- To record and manage IPO data dynamically using linked lists.
- To enable undo actions (cancellations) using stacks.
- To calculate and display oversubscription ratios efficiently.





# Scope

- **Academic Scope:** Demonstrates real-world application of DSA.
- **Practical Scope:** Can be extended to simulate multiple IPOs, investor categories, and advanced allocation strategies.
- **Limitations:** Does not integrate real-time databases or actual stock exchange APIs.





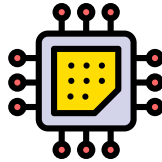
# Requirements of the system (Hardware, software)

## Hardware Requirements:

Processor: Intel i3 or higher

RAM: 4 GB minimum

Storage: 1 GB free space



## Software Requirements:

OS: Windows / Linux

IDE:- VS Code

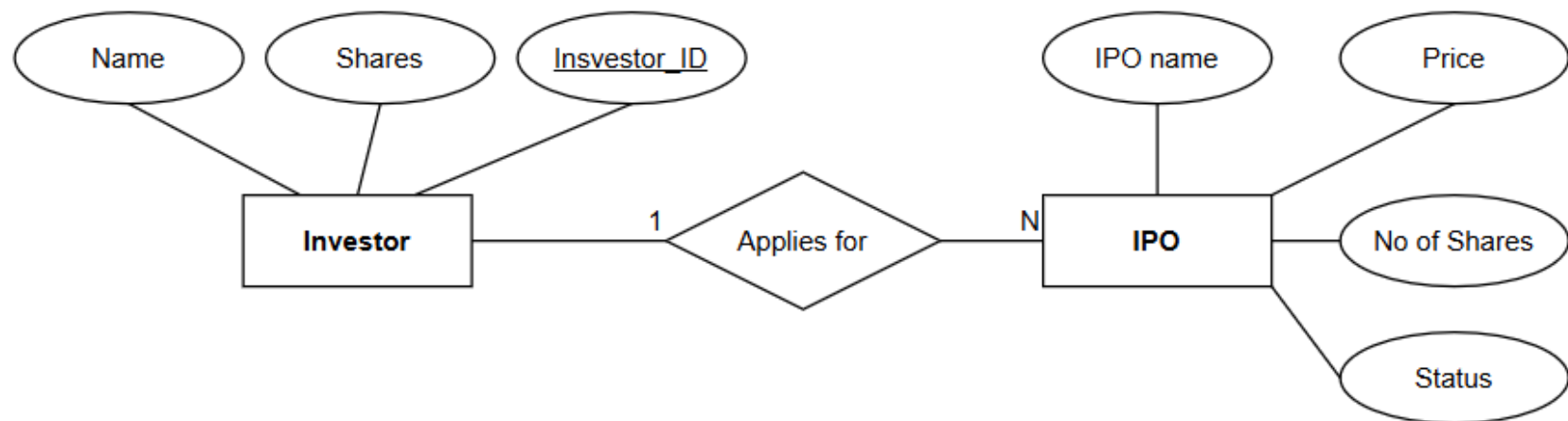
Programming Language: Python

Github (for version controls and submission)





# ER diagram of the proposed system





# Front End

- It provides users with a clear and interactive interface, where each panel corresponds to a specific stage of the IPO subscription process.
- A visual, component-based interface for:
- IPO Applications → Submit, Process, Display Status
- Application Queue (FIFO) → Enqueue new applications, Dequeue for processing, Visualize
- Allotted Investors (Linked List) → Add approved applications, Display list
- Rejected Applications (Stack) → Push invalid applications, Display list
- The dashboard provides a real-time summary of pending, allotted, and rejected applications.

The screenshot shows a web application titled "IPO Subscription Simulator". It features a dark blue header with the title in yellow. Below the header, there are input fields for "Investor ID:", "Name:", "Select IPO:" (with a dropdown menu showing "TechNova Ltd"), and "Shares:". Below these fields are four yellow buttons: "Apply for IPO", "Process All", "Reset All", and "Export CSV". Below the buttons is a table with columns for "ID", "Name", "IPO", and "Shares". The table is currently empty. At the bottom, there is a section titled "IPO Info Panel" with the text "Select an IPO to see details."



# Data Structures & Concepts Used

**Queue (FIFO):** Applications enter in order.

**Stack (LIFO):** Rejected applications stored.

**Linked List:** Stores final list of successful investors.

Searching & Traversal Algorithms.





# Algorithm Explanation

**ApplyIPO:** Enqueue application in Queue.

**ProcessApplications:** Dequeue applications, validate.

- If valid → Insert into Linked List.
- If invalid → Push into Stack.

**SearchInvestor:** Traverse Linked List.

**DisplayAllottees:** Print Linked List.

**DisplayRejected:** Print Stack.



# Future Scope

- Multiple IPO handling.
- Integration with database.
- GUI-based frontend.
- Real-world brokerage simulation





# Gantt Chart

Task	Duration	Start	End
Rquirement Analysis	2	Day 1	Day 2
Design and ER model	3	Day 3	Day 5
Data Structures & Concept	5	Day 6	Day 10
Algorithn Explanation	2	Day 11	Day 12
Implementation	4	Day 13	Day 16
testing and debuggomg	3	Day 17	Day 19
Future Scope	1	Day 20	Day 20
Conclusion	2	Day 21	Day 22



# Conclusion

This project demonstrates how Queue, Stack, and Linked List can be applied in a real-world scenario like IPO Subscription. Applications are managed in a queue, valid investors are allotted shares through a linked list, and invalid ones are pushed into a stack. The simulator helped us understand the working of data structures practically. It shows how simple concepts can solve complex tasks effectively. This project also improved our coding, problem-solving, and analytical skills.



# References

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, Introduction to Algorithms, MIT Press, 2009.
- GeeksforGeeks, “Data Structures in C,” 2024.
- Investopedia, “Initial Public Offering (IPO),” 2024.