Table of contents

Maps 2
Communication System 4

Maps

Introduction

Contributor: Agnirudra Sil

My project focused on creating a local navigation system tailored specifically for our college campus. The goal of this system is to help students, staff, and visitors find the quickest path to any building or location within the campus grounds. By calculating the shortest route, this system makes it easy for anyone to navigate efficiently from one point to another, saving time and reducing confusion.

This system is especially useful for new students, such as first-year students, who might not be familiar with the layout of the campus yet. Instead of struggling to find classrooms, labs, or other facilities, users can rely on this navigation tool to confidently get where they need to go. Ultimately, this project aims to make our campus more accessible and welcoming, ensuring that everyone—especially newcomers—can move around smoothly and make the most of their time here.

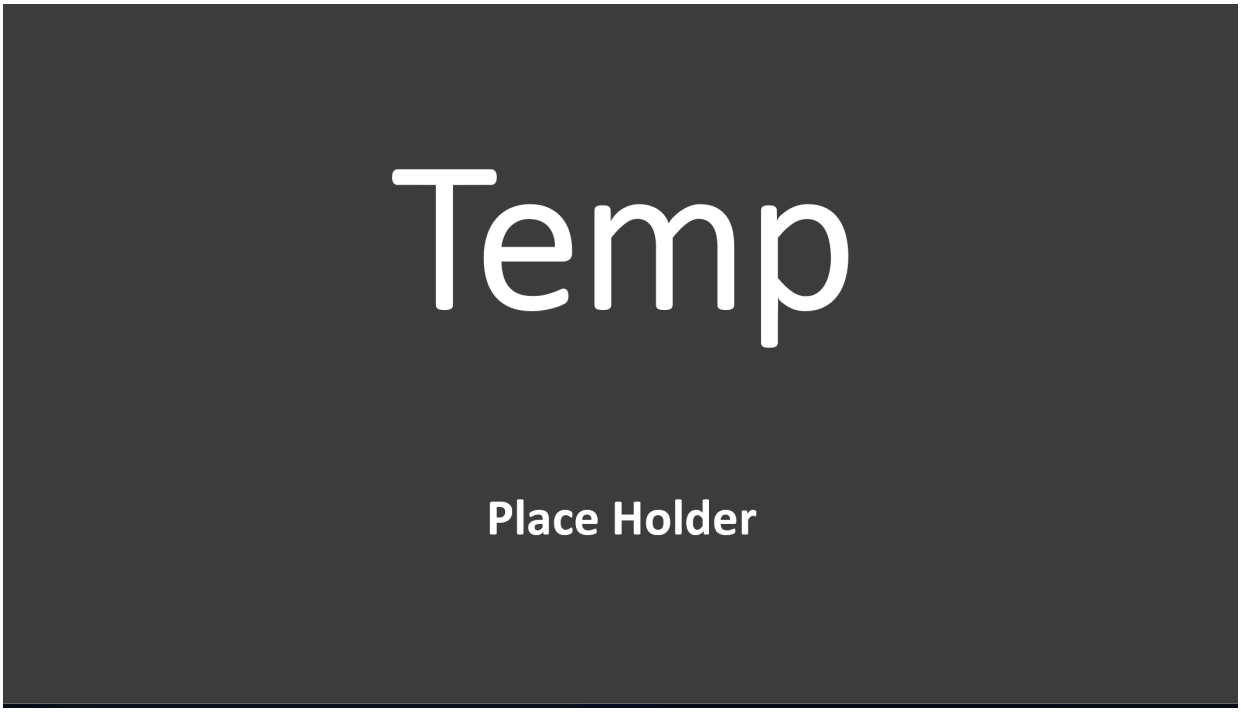
Design

Implementation

Execution

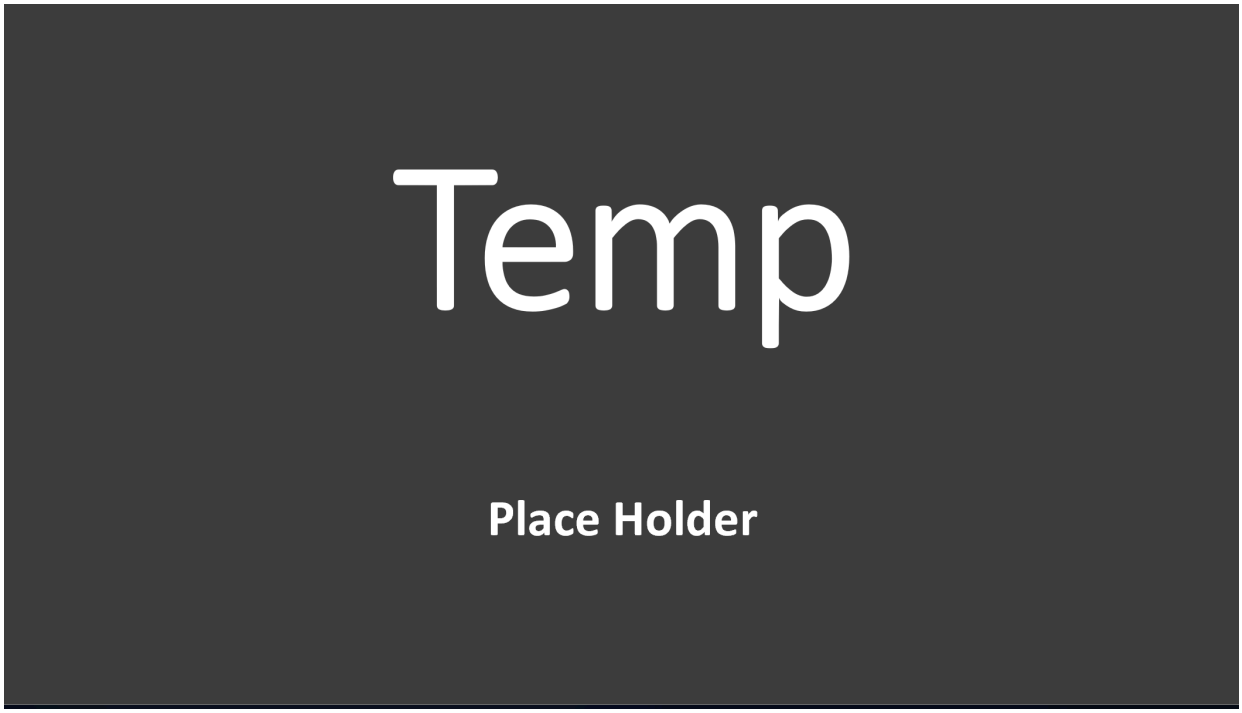
Let us take a realistic situation, Suppose I am first year student. Being new to the college, I am not very much aware of the campus. Let us say there is an event taking place at LHC-C and I wish to know the quickest path to reach there. Here is where my system would come to aid.

First, Open the app and enter your current location (In my case block 1 hostel- karavalli) then enter the location you wish to reach-destination (LHC-C).



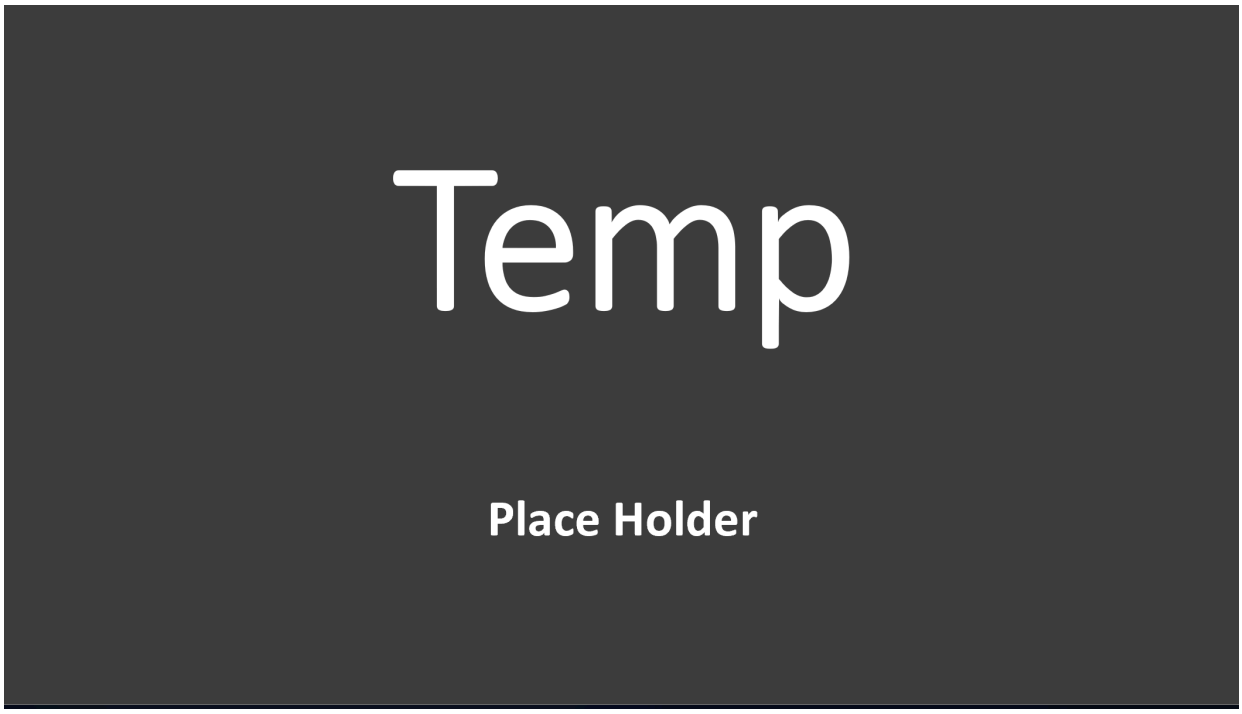
place holder.png

The app would immediately give me the shortest path to reach LHC-C. In this case, it would take me on the sky-walk and give me a rough estimate of the time it would take to reach there.



place holder.png

Also, the app allows me to know if the destination location is open or closed



place holder.png

Communication System

Contributor: Jaicharan Yeluri

My work was to build an effective communication system for our college. However, Implementing the entire communication system from scratch would have been a lengthy process, making it difficult to complete the project within a feasible timeframe. To address this, I used an existing project of my team member, Agnirudra. His project, a Discord messenger clone, closely resembles the communication system we envisioned for our college. By building upon this foundation, we were able to streamline development and bring our idea closer to realization.

The aim of this project is to establish a single communication system for the entire college, allowing for streamlined, effective communication with all users, with minimal effort.

Design

In the project, I worked on how to process multiple messages when several users are posting messages in the same group. So my work was to orderly collect all the messages and present them in the group is a structured manner. I implemented this feature using a linear queue data structure with a singly linked list. While an array could have been used, it would make dequeuing a given node much more complex and time consuming. If the head element needs to be removed in an array, it requires shifting the indices of all subsequent elements, which would impact performance. By using a linked list, however, this challenge is minimized, as enqueuing and dequeuing a node is straightforward and efficient."

Implementation

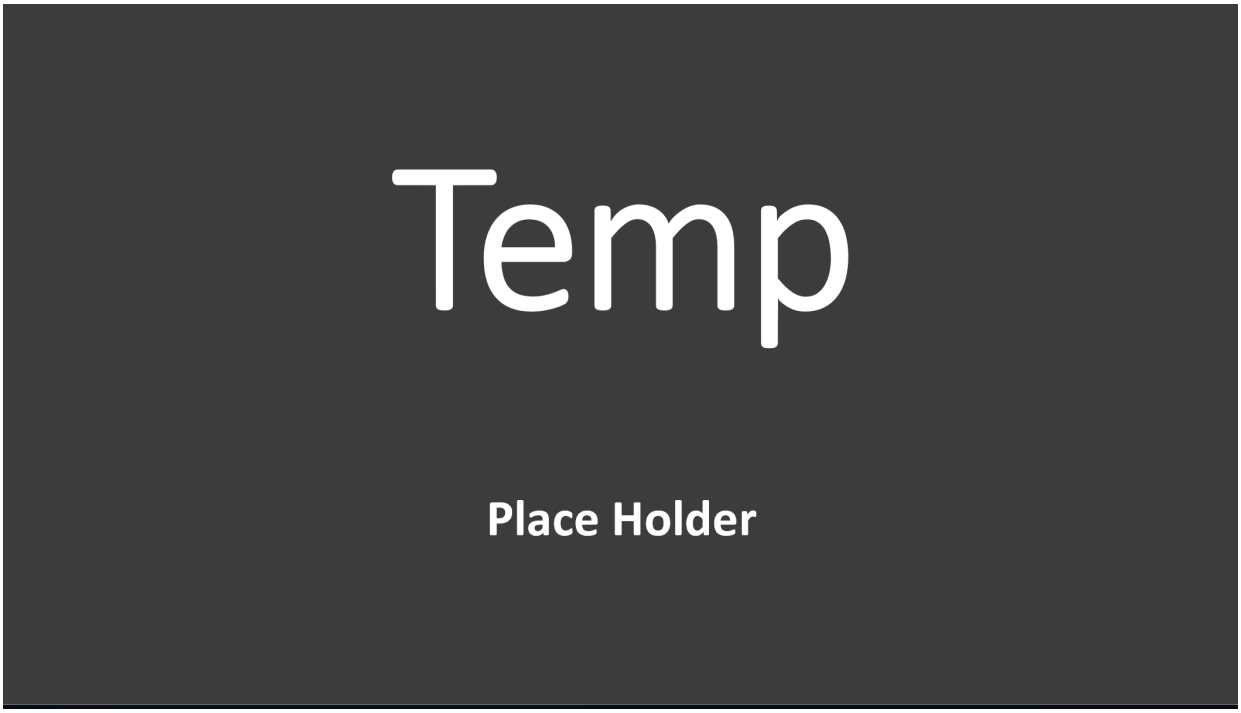
I have implemented linked list and the queue data structure for searching in python in the "linked_list.py" and "queue.py" files respectively. One of the reason why I chose python over other languages to implement the data structures is because python is very direct, almost like writing in english. This way I can focus more on the logic than the syntax of the code. I have also documented my code effectively. Links to the code are provided below

Github links to the documented codes:

- Linked lists : https://github.com/JCode16/avault/blob/ff7407cd5df86d551fe315a41cd330943d97bdd7/backend/app/api/core/linked_list.py
- Queue : https://github.com/JCode16/avault/blob/092398f92ebcbae0a097a244a56606b39508bbf4/backend/app/api/core/queue.py

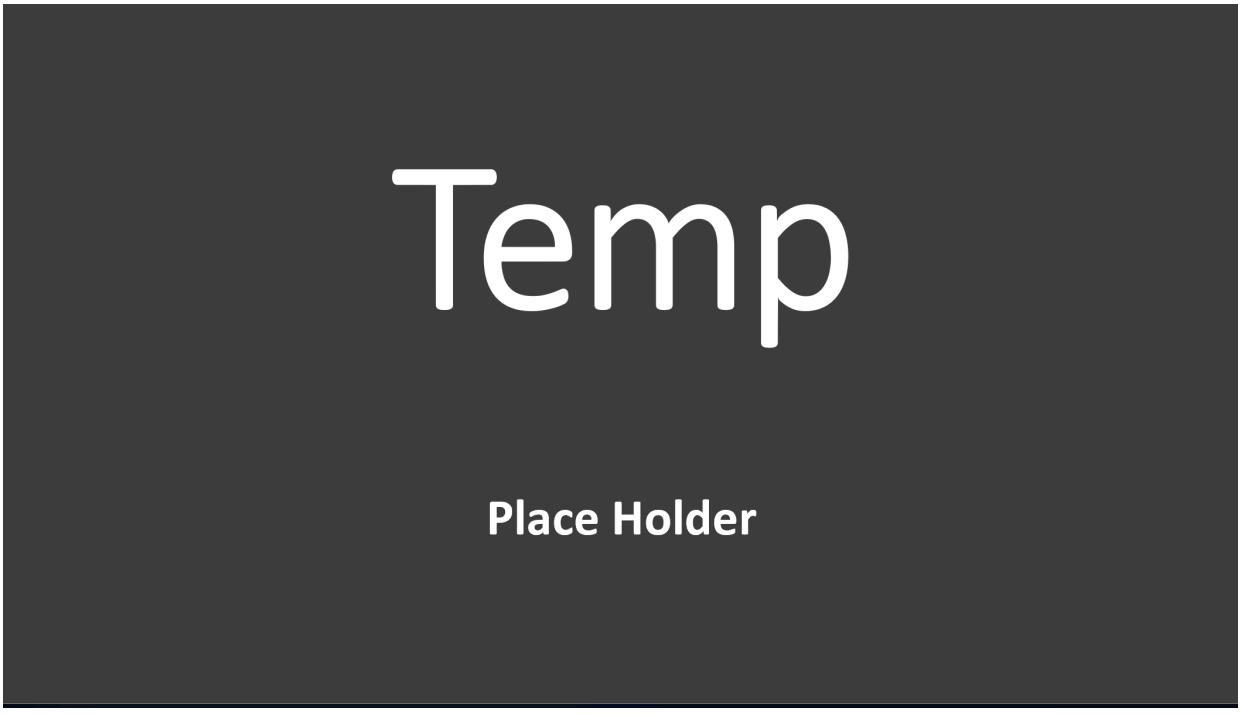
Execution

Let us take a realistic situation, Suppose I have a group chat with my team members.



place holder.png

we would be sending different messages at different time intervals. The feature that I implemented now would take all this messages based on a FIFO(first in first out) principle and deque them in the chat group



place holder.png