

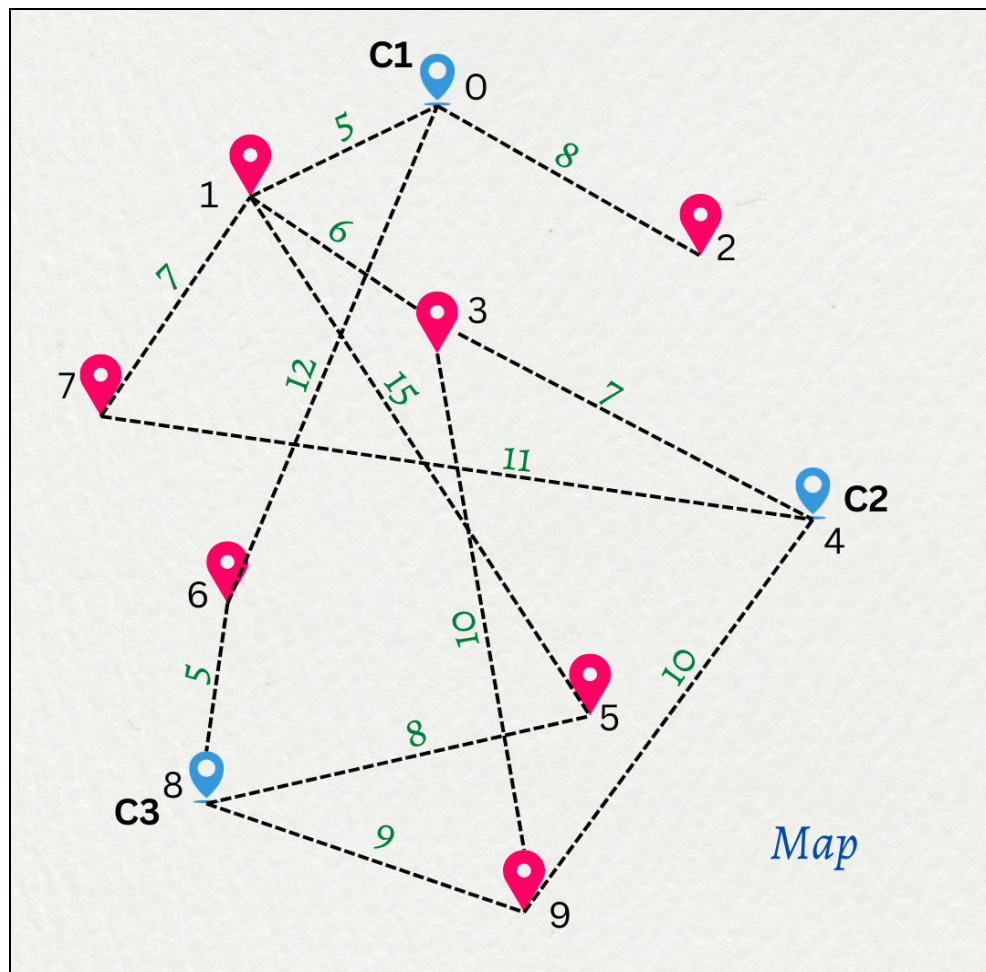
MINI-PROJECT - 'PAWSITIVITY'

DOGS RESCUE AND ADOPTION CENTERS



Group Details -

1. Anushri Patil (UCE2021450)
2. Isha Shinde (UCE2021465)
3. Mrunal Vibhute (UCE2021470)
4. Aditi Wagh (UCE2021472)



PROJECT DETAILS

Our project **“Pawsitivity”** focuses on reducing the damage that we humans have caused to dogs' lives by rescuing the abandoned dogs and taking good care of them. Many times people wish to genuinely offer a helping hand for these innocent animals but cannot contribute after an extent. So we realized that we need a centralized system that would connect the potential helpers and the needy dogs. Its an initiative where user can :

1. **Rescue a dog and hand it over to our trusted NGO rescue centers :** If a user finds a dog who needs help the user just needs to enter the pin code of his current location and he will be suggested nearest rescue center.To perform this we have used graph data structure.
2. **Adopt a dog in need , and bestow it with happiness :** For adoption we have used array data structure. We used an array as an object of a class that has various attributes. This array served as a database for our NGO.
3. **Donate money , funding our NGO that might help a dog in need.**

Our platform is not just user friendly but also admin friendly. Which makes the process of rescue , adoption and donation easier.At the admin side we keep record of various details and analyze the data for further expansion of business.

At the admin side we have :

1. **Keeping record of all dogs present in the NGO at the moment.**
2. **Analysis of the number of dogs in each rescue center.**
3. **Analysis of number of dogs received from different pin codes.**
4. **Record analysis of adopted dogs from our NGO.**
5. **Check the funds available.**

We have made use of many conditional statements like if else , switch case and loops that systematically bind our applications and maintain the flow of our

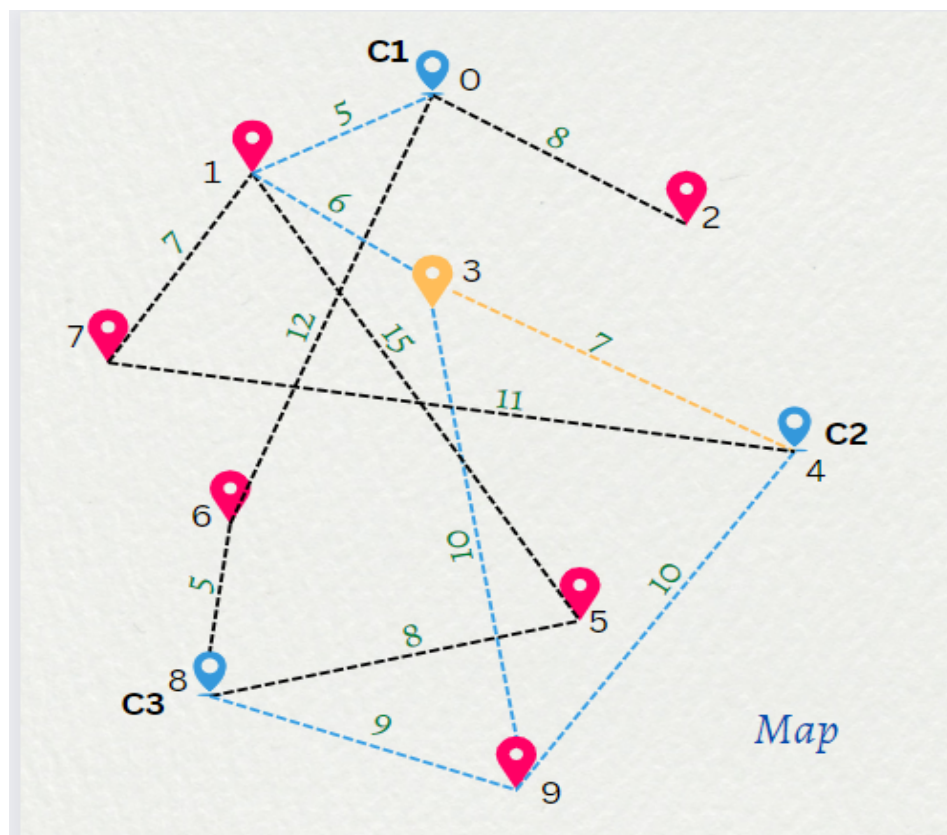
program. We also made a point to include exception handling to handle the input mismatch exception.

DATA STRUCTURES USED

The Various Data Structures we have used are Graph, Array and Linked list.

Graph : A graph is a data structure that stores connected data. In other words, a graph G (or g) is defined as a set of vertices (V) and edges (E) that connect vertices. *We used a graph for creating our city that consists of areas divided as per pin codes.* If a user finds a dog who needs help the user just needs to enter the pin code of his current location and he will be *suggested nearest rescue center*. To perform this we have used graph data structure.

The graph is implemented with the help of a weighted adjacency Matrix, where the weights are the distance between 2 pincodes (regions). The Dijkstra's Algorithm fulfills the requirement.



Ex. If the dogs to be rescued is in the region with pincode 3

Array : An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed. For adoption we have used array data structure. We used an array as an object of a class that has various attributes. *This array serves as a database for our NGO.* One of the reasons we used arrays is that the length is fixed and can be easily accessible which is a crucial point when we are using it as a database. We have performed searching operations many times and in case of arrays the searching operation has the time complexity of $O(1)$.

Linked list : A linked list is a common data structure that is made of a chain of nodes. Each node contains a value and a pointer to the next node in the chain. The head pointer points to the first node, and the last element of the list points to null. We have used a linked list to *store adopted dogs data with adopter details, store information of donors to keep track of fund amounts donated.* The reason we used linked list is because linked list can be easily updated as well as it's a dynamic data structure and makes efficient use of memory.
