# **Report – Week 1**

**Project Title:** A Student Campus Graph Modeling for Navigation

**Prepared by:** K Praneeth Achari

**Week:** 1

**1. Introduction**

In the first week, we worked on setting up a basic map of the campus for our project. We did this by thinking of the campus as a graph, where the important spots like landmarks or waypoints are the points on the map, and the paths connecting them are the lines. This gives us a simple but solid base to build on, and next we’ll be using it to figure out routes and coverage for autonomous navigation.

## **2. Methodology**

1. **Collecting Data**  
   We started by working with a KML file of the campus map and pulled out all the geographic coordinates. From this, we picked out key landmarks like academic blocks, gates, hostels, sports facilities, and utility points. These became the main nodes in our graph.
2. **Breaking Down the Paths**  
   The KML file also gave us path information in the form of polylines. We broke these into smaller segments so that each path could be represented more accurately. We also kept the intermediate points along each pathway so the routes match the real campus layout.
3. **Building the Graph**  
   After that, we assigned latitude and longitude to every node. Then, we linked these nodes together based on the actual paths in the map. To make navigation realistic, we calculated the distances between connected nodes using the geodesic formula, which ensures that the edge weights represent true distances on the ground.

## 

## **3. PROBLEM STATEMENT**

Key landmarks modeled as graph nodes include:

* **Academic Infrastructure:** Block A, Block B
* **Accommodation & Services:** Hostel, Food Court, Rest Area
* **Entry & Security Points:** In Main Gate, Out Gate, Check Post, Flag Post
* **Sports Facilities:** Basketball Ground, Cricket Ground, Football Ground, Volleyball Ground, Tennis Court

## **4. Objectives**

## **Build a graph-based model of Chanakya University campus, representing nodes (landmarks) and edges (walkable/drivable paths).**

## **Implement robust pathfinding algorithms for efficient route suggestions across campus.**

## **Develop a chatbot interface (using Rasa or Dialogflow) to handle navigational queries and information retrieval.**

## **Integrate mapping tools and geolocation APIs for accuracy.**

## **Design a user-friendly interface accessible via mobile or web for ease of use**

## 

## 

**5.Scope**

The system will focus on making campus life easier for students, staff, and visitors by:

* Helping users find their way between important places such as academic blocks, hostels, sports complexes, knowledge centres, dining areas, and more.
* Offering quick access to useful information like department roles, frequently asked questions, office timings, and available amenities.
* Suggesting the shortest or most convenient routes within the campus to save time and effort.

## 

## **6. Week 1 Deliverables**

* Extracted coordinates of all major landmarks and path points
* Constructed campus path network with distances
* Built weighted undirected graph representation
* Prepared adjacency structure for algorithmic use
* Draft report compiled for documentation

**Appendex 1 :**

****

**Functional Requirements**

**1) Back-End**

* **Represent the campus as a graph with at least 12 main landmarks.**
* **Use algorithms (BFS, DFS, UCS, A\* ) to find routes.**
* **Show path, distance, and walking time for chosen routes.**

**2) User Interface (UI)**

* **Dropdowns to pick start and end locations.**
* **Option to choose which algorithm to run.**
* **A “Find Route” button to generate the path.**
* **Show a simple campus map, highlight the route, and display distance + time.**
* **Clicking on a building gives extra details (like purpose, office hours, etc.).**

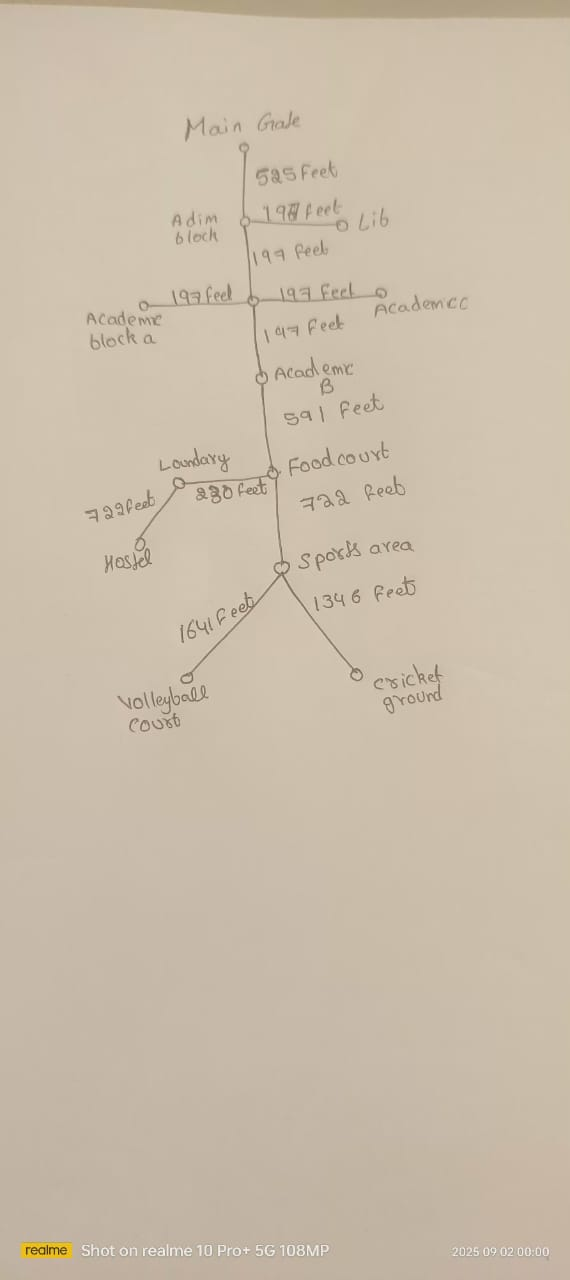
**3) Algorithm Comparison**

* **Show how many nodes each algorithm explored, to compare their efficiency.**

**Data Requirements**

* **Graph Data:**
  + **Nodes = 12+ landmarks (Main Gate, Admin Block, Library, Hostels, etc.)**
  + **Edges = Paths between them with distances.**
  + **Coordinates = X-Y values to help A\* algorithm.**
* **Location Info:**
  + **Simple dataset with names + details of each building.**
* **Tools & Technologies:**
  + **Python for coding algorithms and UI.**
  + **Google Maps API or OpenStreetMap for the campus layout.**
  + **MS Word & PowerPoint for documentation.**

**WEEK 2**

****