BOTBRAIN: AN INTELLIGENT STUDENT CAMPUS NAVIGATOR

Abstract:

This project aims to design and implement an intelligent agent called Bot-Brain that assists students in navigating the Chanakya University campus. The system models the campus as a graph and implements AI-based search algorithms (BFS, DFS, UCS, and A\*) on it to provide chatbot-like navigation services between any two points. Additionally, it will provide information about the building, such as facilities, timings, and FAQs. The project utilizes AI and graph theory concepts in agent design to promote campus accessibility and enhance the student experience.

Introduction:

Navigating a large campus like Chanakya University can be challenging, especially for new students and visitors. An intelligent navigation assistant could provide directions and building details. Bot-Brain is a student-friendly AI-powered campus assistant.

Problem Statement

Students struggle to find necessary university buildings, facilities, and administrative offices. Currently, no centralized AI-based navigation system exists. This inevitably leads to wasted time, confusion, and inefficiency. There is a need to develop a digital solution that supports navigation, information query services, and intelligent queries.

Objectives

* To create a graph-based campus model with all major locations.
* To implement BFS, DFS, UCS, and A\* search algorithms for navigation.
* To develop a chatbot-style interface for users’ queries.
* To provide information about the building such as times, facilities, and services.
* To compare the efficiency of various search algorithms.

Scope

The scope is restricted to navigation and building information retrieval within Chanakya University campus. Users can input queries like “Find path from Hostel to Library. “The system provides text-based outputs.” Advanced features such as GPS tracking and voice assistance are beyond the scope of this project, but they could be incorporated later.

Functional Requirements:

1. Back-End:

* Visualize the campus as a graph with 12 nodes and weighted links.
* Implement BFS, DFS, UCS, and A\* pathfinding methods.
* Provide calculations for path, distance and timings of walking.

1. User Interface (UI):

* Dropdown menus to select start and end locations.
* Ability to select the algorithm.
* A “Find Route” button to process the query.
* Display a simplified campus map, highlight the shortest path on it and show the distance and walking time.
* Provide location details upon clicking.

1. Algorithm Comparison:

* Indicate for analysis how many nodes each algorithm explored.

Data Requirements:  
  
i)Graph Data:  
  
Nodes: At least 12 campus landmarks (Main Gate, Admin Block, Library, etc.)  
  
Edges: Paths between landmarks with distances (meters).  
  
Coordinates: X-Y values to assist A\* heuristic implementations.  
  
ii) Location Info: This is the key-value pair dataset used for descriptions of each building.  
  
iii) Tools & Technologies:  
  
Programming: Python for algorithms, UI, and data processing.  
  
Libraries: Tkinter, PySimpleGUI (or optionally PyQt/PySide) for the interface.  
  
Maps: Google Maps API or OpenStreetMap for the realistic layout simulation.  
  
Documentation: MS Word and PowerPoint for reports and presentations.

WEEK 2

