Project Planning & Understanding: BotBrain Campus Navigator (Week 1)

*This document outlines the initial project plan, scope, requirements, and technology stack for the "**BotBrain**" intelligent campus navigation agent for Chanakya University*.

1. Project Synopsis

Project Title

Intelligent Campus Navigation Agent (ICN)

➢ Abstract

BotBrain is an intelligent agent designed to help students, faculty, and visitors navigate the Chanakya University campus efficiently. The project focuses on implementing and comparing four fundamental search algorithms (BFS, DFS, UCS, A*) to find optimal paths between various campus locations. Through a user-friendly interface, users can select navigation algorithms, visualize search processes, and receive detailed route information. The system serves as both a practical navigation tool and an educational platform for understanding artificial intelligence search methodologies in real-world applications.

➢ Introduction

Navigating a university campus, especially for new students and visitors, presents significant challenges in finding efficient routes between buildings and facilities. The Chanakya University campus, with its diverse academic blocks, residential areas, and recreational facilities, requires an intelligent solution to address wayfinding difficulties. The "BotBrain" project addresses this challenge by developing a goal-based intelligent agent that applies classical AI search algorithms to solve path-finding problems. Beyond providing navigation assistance, this project demonstrates the practical application of artificial intelligence concepts in solving real-world spatial reasoning problems.

➢ Problem Statement

Students and visitors at Chanakya University face multiple navigation challenges:

- New users struggle to locate buildings and understand campus layout
- Manual route selection often leads to longer walking distances and time wastage
- Difficulty in obtaining building-specific information and services during navigation

Objectives

The primary objectives of the BotBrain project are:

- Model the Chanakya University campus as a weighted graph with buildings as nodes and walkways as weighted edges
- Implement four core search algorithms (BFS, DFS, UCS, A*) with proper data structures and optimization techniques
- Develop a goal-based agent following PEAS (Performance, Environment, Actuators, Sensors) architecture principles
- Conduct comprehensive algorithm comparison analyzing path optimality, node exploration, and computational efficiency
- Provide contextual building information including services and departmental details

Scope

In-Scope:

- Campus coverage includes all 12+ mandatory buildings specified in assignment requirements
- Implementation of all four required search algorithms with complete functionality
- Text-based user interface supporting query processing and algorithm selection
- Basic building information database with services and operational details
- Algorithm performance comparison with quantitative metrics
- PEAS analysis and intelligent agent architecture documentation

- One selected system enhancement (Visual Output recommended)

Out-of-Scope (Future Enhancements):

- Real-time GPS integration and mobile application development
- Dynamic obstacle detection and traffic-aware routing
- Multi-modal transportation options (vehicle, bicycle routing)
- Integration with university information systems and databases
- Advanced heuristics and machine learning-based route optimization
- Voice-based interaction and accessibility features

➢ 2. Requirement Document

Functional Requirements

Core Navigation Functional requirements:

- **Campus Graph Representation:** The system shall model the campus as an undirected weighted graph with buildings as nodes and walkways as edges
- **Algorithm Selection:** Users shall be able to specify which search algorithm to use for pathfinding
- **Route Visualization:** The system shall display calculated paths with step-by-step directions and distance information
- **Performance Monitoring:** The system shall track and display algorithm performance metrics including nodes explored and execution time
- **Building Information Display:** The system shall provide basic information about destination buildings including services and hours
- **Text-Based Interface:** The system shall provide a command-line interface for user interaction
- **Input Validation:** The system shall validate user inputs and provide helpful error messages for invalid queries
- **Menu Navigation:** The system shall offer clear menu options for algorithm selection and feature access

- **Output Formatting:** The system shall present results in clear, readable format with proper alignment and spacing

Data Requirements

- **Building Registry:** Complete list of campus buildings with unique identifiers, coordinates, and classifications
  - **Academic Facilities:** Academic Block A (Engineering/CS), Academic Block B (Management), Academic Block C (Arts/Humanities)
  - **Support Services:** Dr. Sita Ram Jindal Administrative Block, Vidyaranya Library, Medical Center
  - **Residential:** Main Hostel, Faculty Housing (if applicable)
  - **Recreation/Dining:** Sri OP Jindal Food Court, Sports Complex, Student Center
  - **Infrastructure:** Main Gate, Preksha Auditorium

- **Path Network:** Bidirectional walkway connections between buildings with accurate distance measurements in meters
- **Path Attributes:** Walking surface types, accessibility features, and any special constraints (one-way paths, construction zones)
- **Service Catalogs:** Comprehensive lists of services available in each building
- **Operational Information:** Building hours, access restrictions, contact information
- **Departmental Mapping:** Office locations within buildings, faculty directories, facility-specific information
- **Navigation Aids:** Building entrance locations, parking availability, accessibility features
- **Execution Metrics:** Storage for algorithm performance data including execution time, memory usage, and nodes explored

3. Tools & Technology Stack

Core Development Technologies

- **Primary backend Language:** Python
- **Search Algorithms:**
  - **Breadth-First Search**
  - **Depth-First Search**
  - **Uniform Cost Search**
  - *A Search:** Heuristic

Data Management

- **Data Storage:** JSON files for campus data persistence and easy modification
- **Configuration Management:** YAML/JSON configuration files for system parameters and building information
- **Logging System:** Python `logging` module for debugging and performance analysis

Potential Future Technologies:

- Google Maps API