School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

MINI PROJECT REPORT

"AI-Based Virtual Campus Guide"

submitted to,
School of Engineering and Technology, CMR University in partial fulfilment

of the requirement for the award of the degree of

Bachelor of Technology, in COMPUTER SCIENCE AND ENGINEERING(AI&ML)

by

Nandini A S(USN:24BBTCA077)
Niverthy Harshitha(USN:24BBTCA079)
Neysa Mary Pramod(24BBTCA078)
Om Chavhan(24BBTCA080)

Under the guidance of, Dr. Kedar Prasad Agrawal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-(AI&ML)
CMR UNIVERSITY

2024-25

School of Engineering and Technology, CMR

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-(AI&ML)

CERTIFICATE

Certified that the mini project titled carried out by Mr./Ms. Nandini AS (USN:24BBTCA077), Niverthy Harshitha(USN:24BBTCA079) Neysa Mary Pramod (USN: 24BBTCA078) ,Om Chavhan (USN: 24BBTCA080) in partial fulfilment for the award of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING-(AI&ML) of CMR University, during the year 2024-25. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Dr. Kedar Prasad Agrawal

Signature of the Guide

Examiners

Name of the examiners	Signature with date
1	
2	

DECLARATION

We ***Nandini AS,Niverthy Harshitha,Neysa Mary Pramod,Om Chavhan,*** *students of School of Engineering and Technology, CMR university, hereby declare that the dissertation titled* ***"AI-Based Virtual Campus Guide"*** *embodies the report of my mini project carried out independently by us during third semester of* ***Bachelor of Technology in Computer Science and Engineering(AI&Ml),*** *under the supervision of* **Dr. Kedar Prasad Agrawal,** *Department of Computer Science and Engineering and this work has been submitted in partial fulfilment for the award of the* ***Bachelor of Technology*** *degree.*

We have not submitted the project for the award of any other degree of any other university or institution.

Date :

Place :

Nandini AS(USN: 24BBTCA077)
SIGNATURE OF STUDENT-1

Niverthy Harshitha (USN: 24BBTCA079)
SIGNATURE OF STUDENT-2

Neysa Mary Pramod (USN: 24BBTCA078)
SIGNATURE OF STUDENT-3

Om Chavhan (USN: 24BBTCA080)
SIGNATURE OF STUDENT-4

1. Abstract

Project Title: AI-Based Virtual Campus Guide.
Problem:A rule-based chatbot or menu-driven assistant.
Objective: Create a simple agent that provides campus related responses(deparments,labs,timings).

Methodology:

1. **Define the Agent and Environment**
   - **Agent**: Rule-based chatbot or menu-driven assistant
   - **Environment**: Campus-related queries (departments, lab timings, locations)

2. **Knowledge Representation**
   - Use a **Python dictionary** as a simple knowledge base (KB) to store:
   - Department names and descriptions
   - Lab timings and locations
   - Faculty contacts or office hours (optional)

3. **Agent Behavior Design**

   - **Reflex Agent**: Uses if-else rules to respond directly to known queries.

   - **Model-Based Agent**: Maintains internal state (e.g., last query context).

   - **Goal-Based Agent** (optional): Can prioritize user goals like finding open labs now.

4. **Input Processing**

   - Accept user input via:

   - **Command line** (basic)

   - **Tkinter GUI** (optional for visual interface)

5. **Response Generation**

   - Match input to KB using keyword detection or menu selection.

   - Return predefined answers.

Key Outcomes:

OUTCOME	DESCRIPTION
Understanding AI Foundations	Students grasp basic agent types and rule-based logic
Practical Agent Design	Build a working chatbot using Python
Knowledge Representation	Use dictionaries to model campus data
User Interaction	Implement input/output handling via CLI or GUI
Modularity	Code structured for easy expansion (e.g., adding FAQs, maps)

2. Introduction

Artificial Intelligence (AI) is transforming how we interact with digital systems, enabling machines to simulate human-like decision-making and responsiveness. In this project, students will explore the foundational concepts of AI by building a **simple intelligent agent**—a virtual campus guide that responds to student queries about departments, lab timings, and other campus-related information.

This agent acts as a **rule-based chatbot or menu-driven assistant**, demonstrating how AI systems can operate within defined environments using structured knowledge and decision logic. By leveraging **Python** and basic GUI tools like **Tkinter**, students will gain hands-on experience in designing agents that process inputs and deliver meaningful outputs.

3. Problem Statement

As educational institutions grow in size and complexity, students often face challenges in accessing timely and accurate information about campus facilities, departments, and lab schedules. Traditional methods such as notice boards, printed brochures, or manual inquiries are inefficient and may not provide real-time or consistent responses.To address this, we aim to develop an **AI-Based Virtual Campus Guide**—a simple, intelligent agent that can interact with students and provide predefined answers to frequently asked campus-related questions. This project introduces students to the foundational concepts of Artificial Intelligence, particularly the design and implementation of **rule-based agents**.The virtual guide will function as a **reflex or model-based agent**, using **if-else logic** or a **dictionary-based knowledge base** to respond to user queries. It will simulate intelligent behavior by mapping

specific inputs (e.g., “Where is the AI Lab?”) to corresponding outputs (e.g., “The AI Lab is located in Block C and is open from 9 AM to 5 PM.”). An optional **Tkinter-based GUI** can be implemented to enhance user interaction.

Goals

- Introduce students to **AI agents and environments**
- Demonstrate **reflex, model-based, and goal-based agent behavior**
- Apply **rule-based logic** to simulate intelligent responses
- Build a functional **chatbot or menu-driven assistant** using Python

4.Implementation

Here's a complete Python-based implementation of the AI-Based Virtual Campus Guide, aligned with the concepts of reflex agents and rule-based systems using if-else logic and a dictionary knowledge base. An optional Tkinter GUI is also included.

Tools & Technologies

- **Python**: Core programming language
- **If-Else Rules / Dictionary KB**: For decision-making and knowledge storage
- **Tkinter (Optional)**: For building a graphical user interface.

This project serves as a practical entry point into AI, helping students bridge theoretical knowledge with real-world application in a familiar campus setting. Let me know if you'd like help drafting the system design or code structure next.

1. **Programming Language Python 3.13**
   Used for implementing the rule-based system, query processing, fuzzy matching, and the GUI application.Provides simple syntax and large library support for AI and automation.

2. **Libraries & Frameworks**

   a. Streamlit

Used to build the user interface (GUI).

Provides fast, web-based application development with minimal code.

Used for: Inputfields,Buttons,Displaying responses,Layout management

b. RapidFuzz

A high-performance fuzzy string-matching library.

Used to match user queries with stored campus information even when:

Spelling mistakes exist,

Different wording is used (“ai lab”, “ai ml laboratory”, “ml Partial matches are provided.

c.JSON (Knowledge Base Storage)

The rule-based knowledge base is stored in a .json file (e.g., kb.json).Stores all campus department information.

3. Core Concepts & AI Techniques

a. Rule-Based System

Decision-making is based on if-else logic and a structured knowledge base.

Each defined rule maps user inputs to predefined responses.

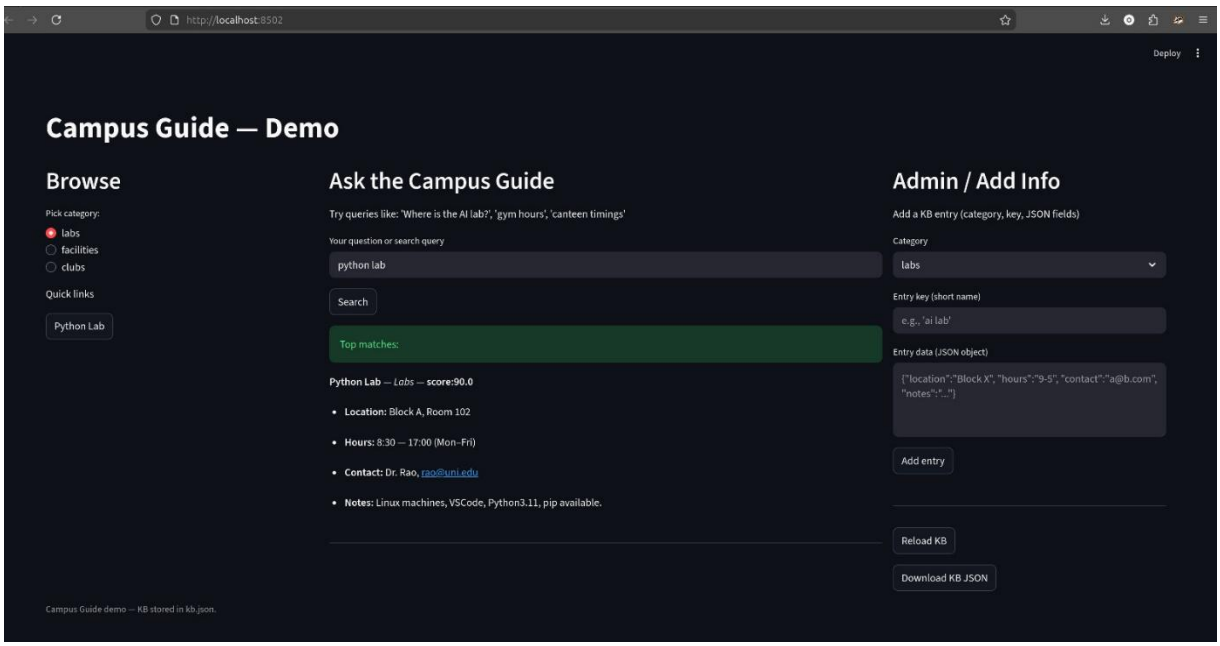
b. Knowledge Representation

A dictionary-based knowledge base is used for structured storage of campus data.

c. Fuzzy Matching (NLP technique)

Allows flexible query interpretation.Ensures correct responses even with incomplete or misspelled queries.

5. Results



6. Conclusion and Future Work

The AI-Based Virtual Campus Guide serves as a practical introduction to the foundational concepts of Artificial Intelligence, particularly the design and implementation of intelligent agents. By building a rule-based chatbot using Python, students gain hands-on experience with reflex agent behavior, knowledge representation, and input-output mapping.

This project successfully demonstrates how simple agents can operate within a defined environment—responding to student queries about departments and lab timings using predefined rules and a structured knowledge base. The optional GUI built with Tkinter enhances user interaction and showcases how AI systems can be made more accessible and user-friendly.Through this exercise, students not only reinforce their understanding of AI principles but also develop essential programming and problem-solving skills applicable to real-world scenarios.

Future work

1. Model-Based Agent Behavior

- Maintain internal state (e.g., track previous queries or user preferences)
- Enable context-aware responses

2. Goal-Based Agent Extension

- Allow users to specify goals (e.g., “Find labs open now”)
- Implement decision logic to fulfill user-defined objectives

3. Natural Language Processing (NLP)

- Integrate basic NLP libraries (e.g., NLTK, spaCy) for better query understanding
- Support synonyms, sentence variations, and fuzzy matching

4. Dynamic Knowledge Base

- Load campus data from external files (CSV, JSON) or databases
- Enable easy updates without modifying code

References

1. Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3), 171–176.
2. Turban, E., & Aronson, J. (2001). Decision Support Systems and Intelligent Systems. Prentice Hall.
3. Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson Education. (Chapter on Rule-Based Agents) .
4. Streamlit Inc. (2023). Streamlit Documentation. Retrieved from https://docs.streamlit.io/
5. Python Software Foundation. (2023). Python Programming Language – Official Documentation. Retrieved from https://www.python.org/doc/
6. RapidFuzz Team. (2024). RapidFuzz: Rapid fuzzy string matching in Python. Retrieved from https://maxbachmann.github.io/RapidFuzz/
7. Richardson, L. (2023). Beautiful Soup Documentation. https://www.crummy.com/software/BeautifulSoup/