## 

## KGiSL INSTITUTE OF TECHNOLOGY

## 

**ChatBot Name**: EduBot

**Problem Statement Title:** Chatbot-based helpdesk for an educational institution using Python

**Category**: Software

**Domain Bucket:** Smart Automation

**PROBLEM DEFINITION**

In today's fast-paced educational environment, educational institutions are challenged with effectively handling student inquiries, providing support, and disseminating information. Traditional methods, such as phone calls and emails, are often slow, resource-intensive, and insufficient to meet the growing demand for immediate assistance. This results in frustrated students, extended response times, and increased operational costs.

There is a pressing need for an efficient and effective support solution that can provide timely responses, handle routine inquiries, and seamlessly escalate complex issues to human agents when necessary. To address this challenge, we are implementing a state-of-the-art educational support chatbot named EduBot. EduBot leverages natural language processing and artificial intelligence technologies to enhance student experiences, reduce response times, and optimize support operations.

The specific need for an educational institution includes:

- Handling student inquiries and providing academic information.

- Assisting with course registration and schedule inquiries.

- Answering frequently asked questions about campus services.

- Guiding students to appropriate resources and support services.

- Enhancing the overall educational experience and satisfaction of students.

Our goal is to create a chatbot that can understand and respond to a wide range of student queries, including academic and non-academic inquiries, ensuring that students have access to timely support and information.

By developing and implementing EduBot, we aim to:

- Improve student satisfaction and academic support.

- Increase operational efficiency.

- Reduce the overall cost of providing high-quality support services.

- Empower our educational institution to meet the evolving needs of our students, enhancing our reputation as a student-centric institution.

**DESIGN THINKING**

1. *Architecture*:

- Client Interface: Students can interact with EduBot through the institution's website, mobile app, or messaging platforms (e.g., Slack).

- Server: The chatbot server processes incoming student messages and generates responses.

2. *Natural Language Processing (NLP):*

- Intent Recognition: Implementing an NLP model (e.g., a pre-trained transformer model like BERT or GPT) to understand the intent of student messages. This helps in categorizing queries and deciding how to respond.

- Entity Recognition: Extracting relevant entities (e.g., course names, dates, or locations) from student queries to provide context-aware responses.

- Sentiment Analysis: Analyzing student sentiment to gauge their emotional state and respond appropriately.

3. *Knowledge Base:*

- FAQs and Documentation: Storing a knowledge base of frequently asked academic questions, course information, and campus services details.

- Content Management: Implementing a content management system (CMS) to easily update and maintain the knowledge base.

4. *Dialogue Management:*

- State Management: Tracking the conversation state, including context, student preferences, and previous interactions, to maintain coherent and context-aware conversations.

- Response Generation: Generating responses based on recognized intents, extracted entities, and the current dialogue context.

- Fallback Mechanism: Implementing a fallback mechanism for handling queries EduBot cannot confidently respond to. This may involve escalating the query to a human agent.

5. *Integration:*

- API Integration: Connecting EduBot with various backend systems and databases to fetch academic information, course schedules, or student details.

- Third-Party Services: Integrating with third-party services for functionalities like language translation, weather updates, or campus event information.

6. *Security and Privacy:*

- Data Encryption: Implementing encryption to ensure the security and privacy of student data.

- User Authentication: Authenticating students when accessing sensitive information.

- Data Retention Policy: Defining a data retention policy to manage the storage and handling of student interactions and data.

7*. Scalability and Load Balancing:*

- Deploying EduBot on a scalable infrastructure to handle varying levels of student traffic.

- Implementing load balancing to distribute incoming requests evenly among multiple servers.

8. *Monitoring and Analytics:*

- Setting up monitoring tools to track EduBot performance, including response times, student satisfaction, and error rates.

- Using analytics to gather insights into student behavior and preferences, enabling continuous improvement.

9. *Testing and Quality Assurance:*

- Implementing automated testing for NLP model performance, regression testing, and dialogue flows.

- Conducting user acceptance testing (UAT) to ensure EduBot meets student expectations.

10. *Deployment:*

- Deploying EduBot on cloud infrastructure (e.g., AWS, Azure, or Google Cloud) for scalability and accessibility.

11. *Continuous Improvement:*

- Implementing a feedback loop to gather student feedback and using it to continuously improve EduBot's performance and capabilities.

12*. Documentation and Training:*

- Creating documentation for developers, support staff, and students to understand how to interact with and maintain EduBot.

**DATA SET**

CSV FILE

**Tech Stacks**

- Programming Languages: Python

- Machine Learning and NLP Libraries: TensorFlow, PyTorch, spaCy, NLTK

- Web Development: HTML, CSS, JavaScript

- Cloud Infrastructure: AWS, Azure, Google Cloud

- Database: SQL, NoSQL

- APIs: Third-party service APIs (e.g., Slack, Google Maps)

**Mentor:** UDHAYAKUMAR S

**Team Members:**

- SENTHURAN L K

- MONA B

-SHWETHA D

- SURYA THANGARAJ B

-SRIKANTH C