

**A PROJECT REPORT**

**on**

**“KBot for KIITians”**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfillment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**NITYA SINGH 2005317**

**SHEETAL 2005334**

**ADITYA 2005283**

**VARTIKA 2005418**

**UNDER THE GUIDANCE OF**

**Dr. Prachet Bhuyan**



**SCHOOL OF COMPUTER ENGINEERING**  
**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**  
**BHUBANESWAR, ODISHA - 751024**

**May 2023**

# IIT Deemed to be University

School of Computer Engineering  
Bhubaneswar, ODISHA 751024



## CERTIFICATE

This is certify that the project entitled

“KBot for KIITians”

submitted by

NITYA SINGH 2005317

SHEETAL 2005334

ADITYA 2005283

VARTIKA 2005418

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 01/04/2023

Dr. Prachet Bhuyan

## **Acknowledgements**

We are profoundly grateful to **Dr. Prachet Bhuyan** of **KIIT University School of Computer Science** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

NITYA SINGH 2005317

SHEETAL 2005334

ADITYA 2005283

VARTIKA 2005418

## **ABSTRACT**

KBot is an AI interactive agent that receives questions from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer.

# Contents

1	Introduction	1
2	Basic Concepts/ Literature Review	2
	2.1 Sub Section Name.....	2
3	Problem Statement / Requirement Specifications	3
	3.1 Project Planning.....	3
	3.2 Project Analysis (SRS).....	3
	3.3 System Design .....	3
	3.3.1 Design Constraints .....	3
	3.3.2 System Architecture (UML) / Block Diagram ...	3
4	Implementation	4
	4.1 Methodology / Proposal .....	4
	4.2 Testing / Verification Plan .....	4
	4.3 Result Analysis / Screenshots .....	4
	4.4 Quality Assurance .....	4
5	Standard Adopted	5
	5.1 Design Standards .....	5
	5.2 Coding Standards .....	5
	5.3 Testing Standards .....	5
6	Conclusion and Future Scope	6
	6.1 Conclusion .....	6
	6.2 Future Scope .....	6
	References	7
	Individual Contribution	8
	Plagiarism Report	9

# List of Figures

1.1 System Architecture or Block Diagram	2
--	---

# Chapter 1

## Introduction

*The KBot for KIITians is an AI interactive agent that receives queries from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, when it receives the question "What time does the gym close today?", it will give a response "The gym closes at 10pm today."*

*The main objective is creating Web API, sample web, mobile, and text messaging interfaces that demonstrate the use of the API. The goal is to provide the students and faculty of KIIT a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate the KBot for KIITians into their projects.*

*This Software is developed to act as a one-stop destination for solving all the queries of the target audience. It provides faster troubleshooting or clearing queries as the user doesn't need to search for any brochure, documentation or FAQs page. This is made possible using the state of the art Machine Learning Large Language Model (LLM) of OpenAI, GPT da-vinci 002, which simulates the learning of a document by an AI and provides response to the query based on the knowledge base.*



## Chapter 2

### Basic Concepts/ Literature Review

1. **LangChain:** *A framework used for developing applications powered by language models.*
2. **Streamlit:** *A framework used for developing quick web applications.*
3. **PyPDF2:** *An open-source pure-python PDF library capable of splitting, merging, cropping, and transforming the pages of a PDF file.*
4. **text-davinci-002:** *An InstructGPT model based on code-davinci-002.*
5. **text-davinci-003:** *An upgrade over the traditional text-davinci-002, capable of responding concisely and more accurately without the need for any examples given in the prompt.*
6. **GPT 3.5 Turbo:** *A neural network-based language model that has been trained on a massive amount of data, and is capable of performing a wide range of language tasks, including language translation, text completion, and question answering.*
7. **GPT 4:** *An upgrade over the GPT 3.5 Turbo model but can only be differentiated when the task complexity reaches a sufficient threshold. More reliable, creative, and able to handle much more nuanced instructions than GPT 3.5.*

## Chapter 3

# Problem Statement / Requirement Specifications

The goal is to provide the students and faculty of KIIT University a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate KBot into their projects.

### 3.1 Project Planning

#### Window

- **Home** - Main homepage of the software. Introduces the user to the software with graphics(images) and textual messages. This page contains the main query input content as well as the reply box.
  1. *Query Input* - Takes the query as a text input from the user and sends it to the backend code.
  2. *Response output* - Returns the processed response to the user's query.
- **About** - Introduces the user to the University. Contains infographics and information about the college.
- **Contact us** - Holds the official support channels of the college.
  1. *Link to the official website for KIITEE*
  2. *E-mail Address of the Admission Department*
  3. *Contact Number of the Admission Department*
  4. **Sidebar**

Introduces the user to the software and describes its capabilities. Contains three Hyperlink Buttons.

- Homepage Button - Redirects the user to the official KIIT homepage.
  - Admissions Button - Redirects the user to the website of the Admission Department of KIIT.
  - Academics button - Redirects the user to the Academics Page of KIIT University.
- School of Computer Engineering, KIIT, BBSR*

## 3.2 Project Analysis

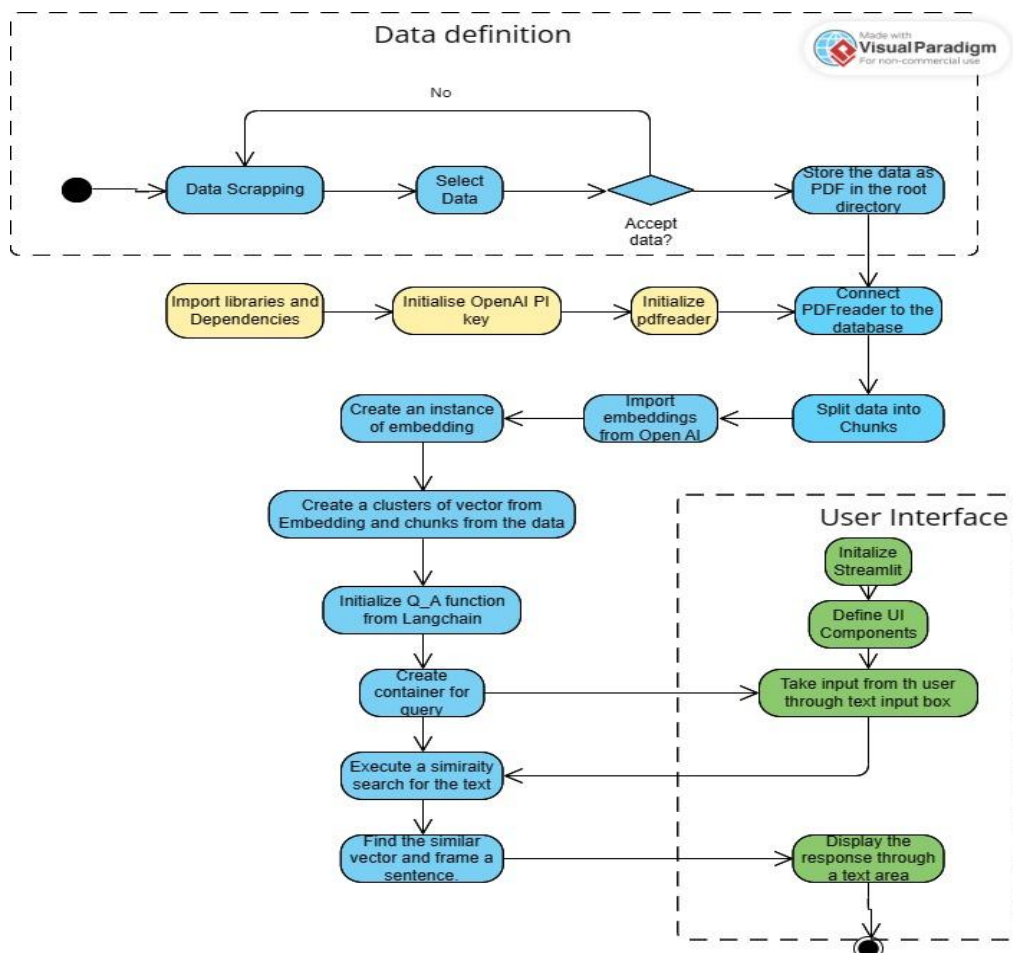
- Limitation due to the available document
- Language Inconsistency

## 3.3 System Design

### 3.3.1 Design Constraints

- The Software, the Frontend User Interface, and even the Backend have been implemented using Python and its libraries or solely functional on python environments. Hence, running on the local machine requires the user to fulfill the version and dependencies of python and the used libraries.
- **Limited Question Scope-** *Creating a chatbot able to answer every single question about KIIT University isn't practically possible to implement with current technology and resources, and within the duration of the project, so the system will be able to answer questions about limited topics.*
- **Language-** *The system will only support questions in standard English.*

### 3.3.2 System Architecture OR Block Diagram



# Chapter 4

## Implementation

### 4.1 Methodology OR Proposal

The implementation is divided into the three parts

#### 1. Data

The data to be used for the query is searched and extracted. A preliminary research is conducted to check the feasibility of the data extracted. On satisfying the essential requirements, The data is preprocessed to remove ambiguous and unnecessary information. Finally, It is passed to the code through an instance of PyPDF2.

#### 2. Preprocessing for the model

The data is further preprocessed by

- Splitting data into chunks.
- lowercasing all the words
- removing special characters
- eliminating stop words.
- removing the headers and Titles from each page

#### 3. OpenAI

The Large language model used for the implementation of transformers is done through the API of GPT 3.5 turbo. After creating an account with OpenAI. An API key is requested. The API key and the library of OpenAI is integrated in the program. Embeddings from OpenAI are imported and its instance is created.

#### 4. LangChain

Using the embeddings from Open AI and clusters from the data, we create vectors of statements. The langchain's Q\_A function is initialized which holds the capability to find the similarity in the vectors from the collection and the user's query and return a well formed response.

#### 5. User Interface

For creating a UI, we have selected Streamlit. It is a python based library which provides templates, themes as well as Javascript elements. A website with three pages and a side bar is created. For accepting the response from the user a text box is placed on the homepage. The user enters the query into the box and a textual response is provided in a text area below.

#### 6. Deployment

For deployment. The project is uploaded to GitHub. The project is initialized on the Streamlit community using the github repository. The environment variables are added(API keys).

## 4.2 Testing OR Verification Plan

Some sample queries are :

### 1. Query : when is the exam?

Response :

The entrance examination will take place from 14-04-2023 to 20-04-2023 for the first phase and from 14-05-2023 to 20-05-2023 for the second phase. These dates are tentative and subject to change.

### 2. Query : what are the subject weightage?

Response :

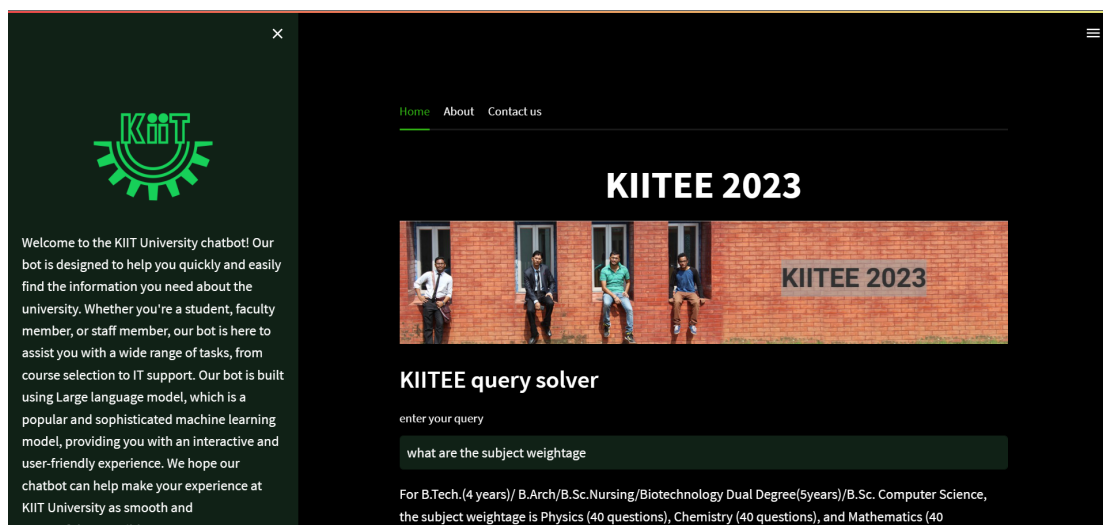
For B.Tech (4 years)/B.Arch/B.Sc Nursing/Biotechnology Dual Degree (5 years)/B.Sc Computer Science, the subject weightage is 40 questions each for Physics, Chemistry, and Mathematics. For B.Tech (Lateral Entry) (3 years) the subject weightage is 40 questions for Mathematics. For MCA, the subject weightage is 30 questions for Computer Awareness, then Mathematics, and then age (preference to older candidate). For M.Tech, preference is given to older candidates. For M.Sc (Applied Microbiology/Biotechnology), the subject weightage is 40 questions for Biology, then Chemistry, then Mathematics, and then age (preference to older candidates). For LLM, preference is given to older candidates. For M.Com/M.A in Economics/M.A in English/M.A in Sociology/M.Sc in Computer Science/Master of Mass Communication, preference is given to older candidates. For Msc (2 years) (Physics/Chemistry/Mathematics and Data Sciences), preference is given to older candidates. For M.A (Yoga Therapy) (2 years), preference is given to older candidates. For M.Sc.

### 3. Query : how to obtain the admit card?

Response :

The Admit Card will be hosted in the website and also send through email. Candidates must preserve the Admit Card till the admission process is over.

## 4.3 Result Analysis OR Screenshots





## About KIIT

KIIT is a private academic institution dedicated to providing educational opportunities for the intellectual, social, and professional development of a diverse student population. To achieve this purpose, the

## Contact us



### Website :

<https://www.kiitee.com/>

### Email :

[admission@kiit.ac.in](mailto:admission@kiit.ac.in)

### Phone :

+918080735735

## Chapter 6

# Conclusion and Future Scope

### 6.1 Conclusion

In conclusion, the development of an AI virtual assistant for a university can provide significant benefits to both students and staff. With the ability to quickly and easily access information and perform tasks, the KBot can improve the overall efficiency and effectiveness of university operations. By addressing key user needs and providing a range of helpful features, the chatbot can also improve the student experience and help promote student success.

The document outlined here provides a comprehensive guide to the development of a chatbot for a university. By addressing key considerations such as purpose, users, features, platform, integration, development, testing and launch, and maintenance, the document provides a solid foundation for the successful implementation of the chatbot.

Overall, the chatbot represents a valuable tool for universities seeking to improve their operations and support their students. By leveraging the power of artificial intelligence and natural language processing, the chatbot can provide fast, accurate, and personalized assistance to students and staff alike. With the SRS document as a guide, universities can move forward confidently in the development and implementation of this powerful technology.

### 6.2 Future Scope

The future scope of a chatbot made for a university is quite promising. As technology advances, chatbots will become increasingly sophisticated and capable of handling a wider range of tasks and user needs. Here are a few potential areas for future development and expansion of a university chatbot:

- **Personalization:** As chatbots become more advanced, they will be better able to understand and respond to individual user needs and preferences. By leveraging data and analytics, a university chatbot can provide personalized recommendations and assistance to students and staff.
- **Integration:** As universities continue to adopt new technologies and systems, chatbots can serve as a bridge between these disparate systems. By integrating with other university tools and platforms, a chatbot can provide a seamless and unified experience for users.
- **Multilingual support:** As universities become more diverse and global, chatbots will need to support a wider range of languages and cultural contexts. By incorporating multilingual support, a university chatbot can serve a broader range of students and staff.
- **Voice-enabled interfaces:** As voice recognition technology improves, chatbots will become more versatile and convenient to use. A voice-enabled chatbot could allow students and staff to access information and perform tasks hands-free, making it even easier to get things done on the go.
- **AI-powered analytics:** As chatbots become more integrated into university operations, they can generate valuable data and insights about user needs and behaviors. By leveraging AI-powered analytics, a university chatbot can help administrators make more informed decisions about resource allocation and student support.

In summary, the future scope of a university chatbot is vast and varied. By embracing new technologies and adapting to changing user needs, a university chatbot can continue to provide value and support to students and staff for years to come.



## ***References***

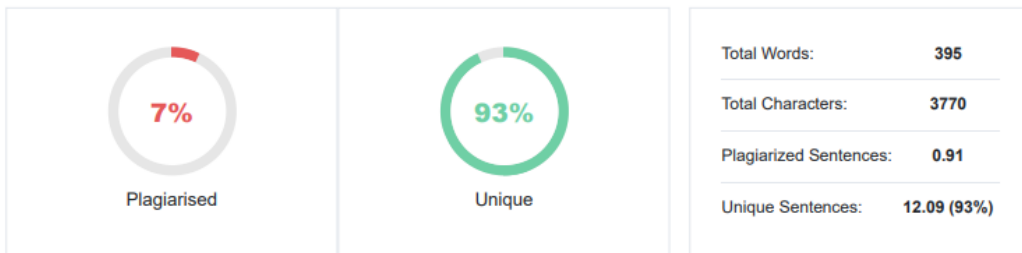
1. Open Ai <https://platform.openai.com/docs/introduction>
2. LangChain <https://docs.langchain.com/docs/>
3. PyPDF <https://pypdf2.readthedocs.io/en/3.0.0/>
4. Streamlit <https://docs.streamlit.io/>
5. KIIT <https://kiit.ac.in/>
6. KIITEEE <https://kiitee.kiit.ac.in/>

# TURNITIN PLAGIARISM REPORT

(This report is mandatory for all the projects and plagiarism must be below 25%)

## Plagiarism Scan Report

Report Generated on: May 01,2023



### Content Checked for Plagiarism

```
# The code is a Python script that creates a chatbot for KIIT University using Streamlit and various
# natural language processing tools such as OpenAI and FAISS. The chatbot is designed to help users
# find information about the university, including admissions, academics, and general queries related
# to KIITEE exam. The script also includes information about the university and contact details for
# further inquiries.
# The line 'from langchain.chains.question_answering import load_qa_chain' is importing the
# 'load_qa_chain' function from the 'question_answering' module of the 'langchain.chains' package.
# This function is used to load a pre-trained question answering model that can be used to answer
# questions based on a given input document or set of documents.
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI
from PyPDF2 import PdfReader
from langchain.embeddings.openai import OpenAIEmbeddings
from lanachain.text splitter import CharacterTextSplitter
```

