CONTENTS

1.0 INTRODUCTION

The project "AI-Assisted IT Placement Finder: Streamlining Student Industrial Training in Abuja" aims to streamline the process of finding suitable industrial training placements for students in Abuja, Nigeria. By harnessing the capabilities of OpenAI's language models, this project seeks to provide personalized recommendations tailored to students' courses, locations, and other preferences.

1.1 Aim of the Project

The purpose of this project is to address the challenges faced by students in locating appropriate companies for their industrial training placements. By leveraging advanced natural language processing techniques provided by OpenAI, the project aims to simplify and expedite the search process, ultimately enabling students to find placements that align closely with their academic and geographical preferences.

1.2 Objective of the Project

The key objectives of the project include:

I. Facilitating an efficient and user-friendly platform for students to explore and discover potential IT placement opportunities in Abuja.
II. Leveraging OpenAI's language models to generate personalized recommendations based on students' courses of study, preferred locations, and other relevant criteria.
III. Enhancing the overall experience of students seeking industrial training placements by providing accurate and relevant information about available opportunities.
IV. Empowering students to make informed decisions regarding their industrial training placements, thereby maximizing the value and relevance of their learning experiences.

2.0 PROJECT OVERVIEW

The process of finding suitable industrial training placements can often be daunting and time-consuming for students. The motivation behind the project "AI-Assisted IT Placement Finder: Streamlining Student Industrial Training in Abuja" stems from the desire to alleviate these challenges and provide students with a more efficient and effective means of securing valuable industrial training experiences. The obstacles often faced include a lack of centralized information about available opportunities, difficulty in matching students' academic backgrounds with relevant industry sectors, and logistical challenges associated with identifying placements that are geographically convenient.

The significance of helping students find IT placements in Abuja cannot be overstated. Industrial training plays a crucial role in the educational journey of students, offering them the opportunity to apply theoretical knowledge in real-world settings, gain practical skills, and build valuable professional networks. By facilitating access to quality industrial training placements, this project aims to empower students to maximize their learning experiences and enhance their employability upon graduation.

2.1 Key Features and Functionality

I. Personalized Recommendations: The project utilizes OpenAI's language models to generate personalized recommendations for students based on their courses of study, preferred locations within Abuja, and other relevant criteria.

II. Course-Industry Mapping: A robust mapping system is implemented to match students' academic backgrounds with relevant industry sectors in Abuja. This ensures that recommendations are tailored to each student's field of study and career aspirations.

III. User-Friendly Interface: The project features an intuitive and user-friendly interface that allows students to easily input their preferences, explore available IT placement opportunities, and engage with the recommendation system.

3.0 PROJECT METHODOLOGY

The methodology of the "AI-Assisted IT Placement Finder: Streamlining Student Industrial Training in Abuja" project follows a systematic approach to leverage OpenAI's capabilities and deliver personalized recommendations to students seeking industrial training placements in Abuja. The methodology involves several key steps:

3.1 Importing Libraries:

The project begins by importing necessary libraries and dependencies, including OpenAI's libraries for accessing language models and performing natural language processing tasks.

```
pip install openai
```

```
# imports the built-in Python module os, which provides a way to interact with the operating system
import os

# imports the OpenAI Python package, allowing you to interact with OpenAI's APIs and access their language models.
import openai
```

Figure 3.1: Importing Libraries Code

3.2 Model and Client Definition:

Next, the project defines the model and client necessary for interacting with OpenAI's API. This involves selecting the appropriate language model (e.g., ChatGPT) and initializing the client to make API calls for generating responses based on user inputs.

```
# Set your OpenAI API key
openai.api_key = "[redacted]"

# Define the language model engine to use
model = "gpt-3.5-turbo"

# Initialize the OpenAI API client
client = openai.Client(api_key=openai.api_key)
```

Figure 3.2: Model and Client Definition Code

3.3 Define the Conversation:

The project establishes a structured conversation format, comprising messages from the user and responses generated by the AI model. Each message contains information about the role (e.g., user or assistant) and the content of the message (e.g., user input or AI response).

```
conversation = [
    {"role":"system",
     "content":"You are a recommender to provide companies where student can have their Industrial training based on their co
    {"role":"user",
     "content":"As an Electrical and Electronics Engineering Student, where can I do my Industrial Training in Abuja?"},
    {"role":"assistant",
     "content":"The following are the companies I can recommend for you to have your Industrial training..."}
]
```

Figure 3.3: The Conversation List

3.4 Creating a Loop for Question and Response:

A loop is implemented to facilitate the interaction between the user and the AI model. Within this loop, the project prompts the user to input questions or preferences related to their industrial training placement. The user's input is then formatted and added to the conversation, and the AI model generates a response based on the input provided. This process continues iteratively until the user's needs are adequately addressed or until a specified stopping condition is met.

```
# Initialize an empty list to store user questions
questions = []

# Prompt the user to input their questions
while True:
    question = input("Ask your question (enter 'done' when finished): ")
    if question.lower() == 'done':
        break
    questions.append(question)

# Loop through each question to generate responses
for question in questions:
    # Format the user input into dictionary form
    input_dict = {"role": "user", "content": question}

    # Add the user input dictionary to the conversation
    conversation.append(input_dict)

    # Make the API call to generate responses for all questions
    response = client.chat.completions.create(
        model=model,
        messages=conversation,
        temperature=0.0,
        max_tokens=100
    )

    # Print the response from the model
    resp = response.choices[0].message.content
    print(resp)

    # Convert the response into the dictionary
    resp_dict = {"role":"assistant",
                 "content": resp}

    # Append the response to the conversation
    conversation.append(resp_dict)
```

Figure 3.4: Conversation Loop Code

4.0 RESULT AND DISCUSSION

4.1 Result

The provided code snippet demonstrates an interactive system designed to assist users in querying and receiving personalized recommendations based on their input. Upon execution, the user is prompted to input questions, with the option to terminate the interaction by entering "done". The system processes each question sequentially, utilizing OpenAI's language model to generate responses tailored to the user's queries.

```
Ask your question (enter 'done' when finished): I live in Garki, Abuja. List the companies available in this area
Ask your question (enter 'done' when finished): Out of this companies, recommend for me as an anatomy student, Where can I h
ave my IT?
Ask your question (enter 'done' when finished): done
1. Nigerian Communications Satellite Ltd (NIGCOMSAT) - Garki
2. National Agency for Science and Engineering Infrastructure (NASENI) - Garki
3. Abuja Electricity Distribution Company (AEDC) - Garki
4. Nigerian Electricity Regulatory Commission (NERC) - Garki
5. Galaxy Backbone - Garki
I recommend National Agency for Science and Engineering Infrastructure (NASENI) in Garki, Abuja for your Industrial Training
as an Anatomy student.
```

Figure 4.1: The Result Snippet

4.2 Discussion

This approach facilitates a conversational interface where users can engage with the system naturally, posing questions in plain language and receiving relevant responses in return. By employing a loop to handle user input, the system accommodates multiple queries within a single session, enhancing user flexibility and convenience.

In the provided example, the user first inquiries about companies available in the Garki area of Abuja, to which the system promptly responds with a list of relevant companies. Subsequently, the user seeks a recommendation for an industrial training (IT) placement as an anatomy student, prompting the system to suggest a suitable option based on the user's academic background and location preferences.

Overall, this code snippet exemplifies the versatility and utility of leveraging natural language processing (NLP) models to create interactive systems capable of understanding and responding to user queries in real-time. Such systems hold significant potential in various domains, including customer service, educational assistance, and personalized recommendation services.

5.0 RECOMMENDATION

The interactive system showcased in the provided in the methodology represents a promising approach for assisting students in finding suitable industrial training placements in Abuja. Building upon this foundation, the project "AI-Assisted IT Placement Finder: Streamlining Student Industrial Training in Abuja" can benefit from several enhancements to further optimize its functionality and user experience:

I. Real-Time Updates:

Integrating real-time updates into the project would ensure that students have access to the latest information about available IT placement opportunities in Abuja. This could involve regularly syncing the system with external databases or sources of information to retrieve up-to-date data on companies offering industrial training placements. Implementing real-time updates would enhance the accuracy and relevance of the recommendations provided to students, enabling them to make informed decisions based on the most current information available.

II. Communication Channel Integration:

Integrating communication channels into the project would facilitate seamless interaction between students and the recommended companies. This could include features such as direct messaging, email integration, or contact forms embedded within the platform. By enabling students to communicate directly with companies from within the platform, the project would streamline the process of initiating contact, inquiring about internship opportunities, and facilitating further engagement. Additionally, integrating communication channels would enhance the user experience by providing a convenient and centralized means of interaction, ultimately increasing the likelihood of successful placements for students.

REFERENCES

https://app.datacamp.com/workspace/w/bcd59ae9-767f-40e6-827e-f8c434a59818/edit