

Dialog Data Science Academy



Capstone Project Report

**CHATGPT BASED EDUCATION ASSISTANT FOR SRI
LANKAN EDUCATION SECTOR**

by

Pasindu Liyanage

August 2023

1 TABLE OF CONTENT

| | | |
|-----|---|---|
| 01. | Introduction | 2 |
| 1.1 | Sri Lankan Education Sector | 2 |
| 1.2 | Major Challenges In The Sri Lankan Education Sector | 2 |
| 1.3 | Scope Of The Project And Objectives..... | 3 |
| 02. | Methodology..... | 3 |
| 2.1 | Base Application Setup | 3 |
| 2.2 | Data Pre-Processing & Vector Database Creation | 3 |
| 2.3 | Llm Object Creation And Prompt Engineering | 4 |
| 2.4 | Application Integration..... | 5 |
| 2.5 | Deployment | 5 |
| 03. | Results | 5 |
| 3.1 | Llm Model Results | 6 |
| 04. | Conclusion & Wayforward..... | 7 |

2 LIST OF FIGURES

| | |
|--|---|
| Figure 1: Code Snippet used for Document Embedding | 4 |
| Figure 2: Code Snippet for Prompt Engineering | 4 |
| Figure 3: API Creation | 5 |
| Figure 4: API Call | 5 |
| Figure 5: Login Page | 5 |
| Figure 6: User Dashboard..... | 6 |
| Figure 7: LLM Model Output..... | 6 |
| Figure 8: LLM Model Multiple Question Output | 7 |

01. INTRODUCTION

1.1 Sri Lankan Education Sector

The Sri Lankan education sector has undergone significant development over the years, playing a crucial role in the country's social and economic progress. The education system follows a 13-year structure, comprising of primary, secondary, and tertiary levels. Primary and secondary education is mandatory and free for all children, reflecting the government's commitment to promoting literacy and access to education. The country boasts a relatively high literacy rate, with an emphasis on providing education in both Sinhala and Tamil languages.

At the tertiary level, Sri Lanka is home to several reputable universities and higher education institutions that offer a range of academic programs. The education sector has shown an increased focus on science, technology, engineering, and mathematics (STEM) disciplines to meet the demands of a rapidly evolving job market. However, challenges such as disparities in access to quality education between urban and rural areas, limited resources, and outdated curricula remain areas of concern. Efforts are being made to address these issues through various initiatives and reforms aimed at enhancing the overall quality and inclusivity of the education system in Sri Lanka.

1.2 Major Challenges in the Sri Lankan Education Sector

- **Geographical Disparities:** Unequal access to quality education between urban and rural areas.
- **Infrastructure and Facilities:** Lack of proper classrooms, libraries, labs, and sanitation facilities in rural schools.
- **Teacher Shortages:** Fewer qualified teachers in rural schools, leading to larger classes and reduced education quality.
- **Quality of Teaching:** Varied teacher training and development, affecting education quality in rural regions.
- **Access to Technology:** Urban-rural divide in technology and digital resources availability.
- **Transportation Challenges:** Difficulty commuting to distant rural schools, causing higher dropout rates.
- **Socioeconomic Factors:** Economic struggles and low parental education impacting rural education.

1.3 Scope of the Project and Objectives

The main scope of this project is leverage on advanced capabilities of AI and GPT framework to create a tool that could assist students anytime anywhere in providing information and answers to their questions based on the exact curriculum data and notes provided by the relevant authorities and educators.

Key Objective of the Project

Development of a web-based tool to assist students anytime anywhere in providing relevant and concise answers and information to their questions and doubts regarding the subject matter.

02. METHODOLOGY

Application development consists of several components in order to develop a functioning MVP.

1. Base Application Setup
2. Data Pre-Processing and Vector Database creation pipeline setup
3. LLM Object Creation & Prompt Engineering
4. Application Integration
5. Deployment

2.1 Base Application Setup

Base application setup was done using Flask framework where the front-end screens are created using HTML, CSS and JS. Bootstrap is used to make the application responsive so users can access it from anywhere device without issue.

2.2 Data Pre-Processing & Vector Database Creation

This application leverages on utilizing School Textbook documents to extract the relevant content and provide answers to students accordingly. Hence for the textbook documents will be broken down into segments and they will be embedded using OpenAI Embedding to create the vector store which would be saved in a ChromaDB locally. By storing the ChromaDB in a persist directory, makes it easier to retrieve the embedded data without performing the embedding function every time the application is executed.

In order to develop the application, Data was taken from NIE website with regards to Grade 10 History subject – Lesson 01

```

embeddings = OpenAIEmbeddings(openai_api_key=openai_api_key)
new *
def documentEmbedding(path):
    loader = PyPDFLoader(path)
    pages = loader.load_and_split()

    text_splitter = RecursiveCharacterTextSplitter(chunk_size=500,
                                                    chunk_overlap=0,
                                                    length_function=len,
                                                    separators=['\n\n', '\n', '.'],
                                                    add_start_index=True,
                                                    )

    text_chunks = text_splitter.split_documents(pages)
    retriever = Chroma.from_documents(text_chunks, embeddings, persist_directory="chroma_db")
    return retriever

```

Figure 1: Code Snipped used for Document Embedding

2.3 LLM Object Creation and Prompt Engineering

Prompt Engineering plays a crucial role in the success and the accuracy of the application. Hence the definition of the prompt was done in a way that it would provide clear and concise instructions to LLM how the data should be looked, and the answer should be generated.

```

def promptEngineering():

    prompt_template = """Use the following pieces of CONTEXT to answer the question at the end and give the explanation \
as you are explaining to a High School Student. \
Do not answer anything outside the CONTEXT given. If you don't know the answer, just say that you don't know, don't \
If answer contain a list, output as a bulleted or numbered list. If answer contain a table return as tab delimited. \
try to make up an answer.\

CONTEXT: {context}

Question: {question}
Answer: """

    PROMPT = PromptTemplate(
        template=prompt_template, input_variables=["context", "question"]
    )

    chain_type_kwargs = {"prompt": PROMPT}

    MODEL = "gpt-3.5-turbo"

    chat_llm = ChatOpenAI(model=MODEL,
                           temperature=0,
                           max_tokens=200,
                           openai_api_key=openai_api_key,
                           verbose=True)

    retriever = getEmbeddings()

    qa = RetrievalQA.from_chain_type(llm=chat_llm,
                                     chain_type="stuff",
                                     retriever=retriever.as_retriever(),
                                     chain_type_kwargs=chain_type_kwargs,
                                     return_source_documents=True)

    return qa

```

Figure 2: Code Snippet for Prompt Engineering

Langchain Library was used in defining the prompt template and for the development of the LLM Chain.

2.4 Application Integration

Application Integration with the front-end web app was done by creating a Flask API which can be called from the front-end which would in turn run all the necessary commands to generate the prompt and call the OpenAI model and return the answer.

```
@blueprint.route('/chat', methods=['POST'])
def chat():
    user_input = request.form['user_input']
    print("USER INPUT from routespy:", user_input)

    response = generateResponse(user_input)
    print(response)
    return jsonify(response=response)
```

Figure 3: API Creation

```
const response = await fetch("/chat", {
  method: "POST",
  headers: {
    "Content-Type": "application/x-www-form-urlencoded",
  },
  body: `user_input=${encodeURIComponent(userMessage)}`,
});
```

Figure 4: API Call

2.5 Deployment

Application was deployed in AWS EC2 server to allow access for users to test out the model.

03. RESULTS

Application consisted of multiple front-end pages for the basic functionalities including Registration, Login, Dashboard and Chat Interface.

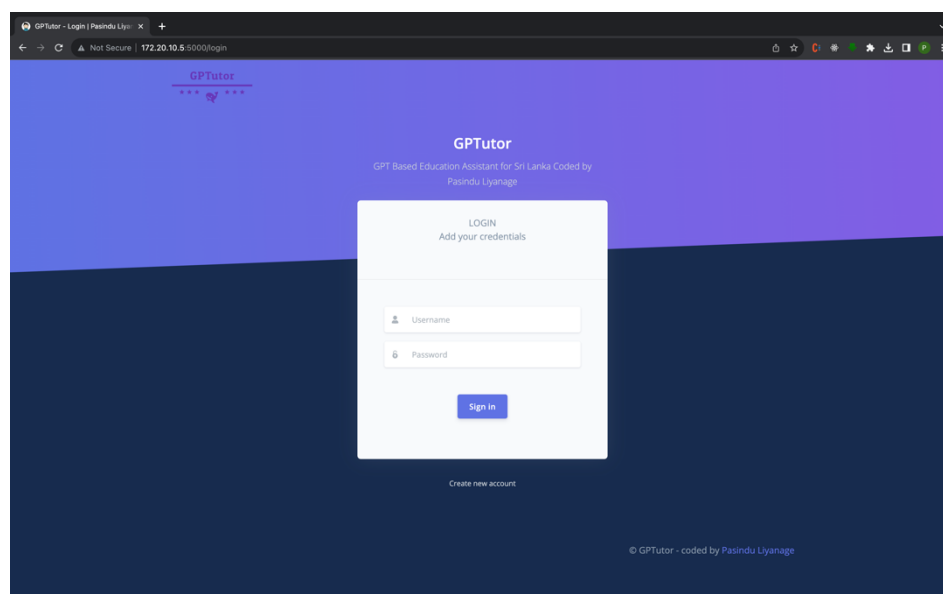


Figure 5: Login Page

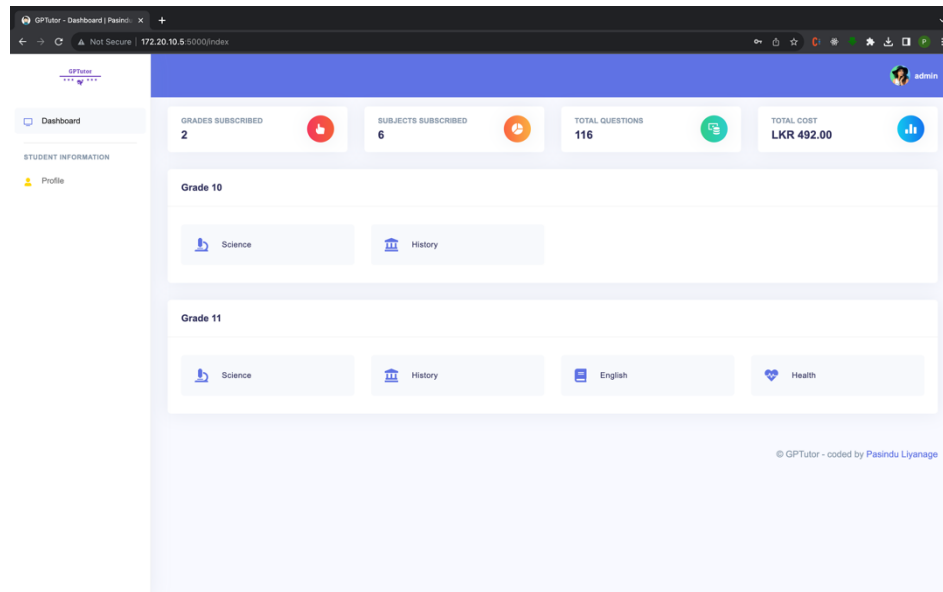


Figure 6: User Dashboard

The dashboard screen will give show all the Subjects, the student has subscribed to and some important information about the service usage. By clicking on any given subject, student can enter the specific bot created for the subject.

3.1 LLM Model Results

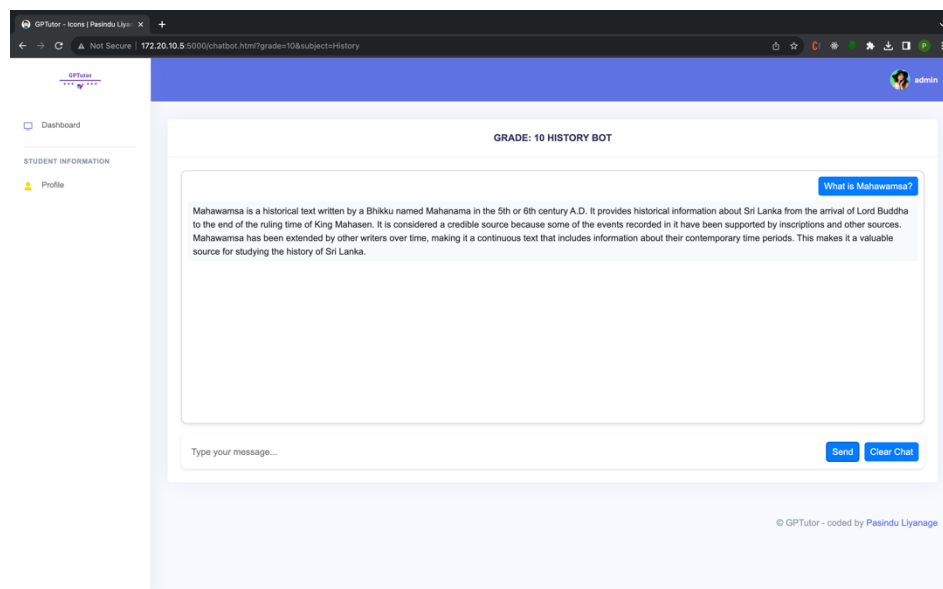


Figure 7: LLM Model Output

LLM model successfully responds to questions asked on the Sri Lankan History very accurately with a great formation of an answer.

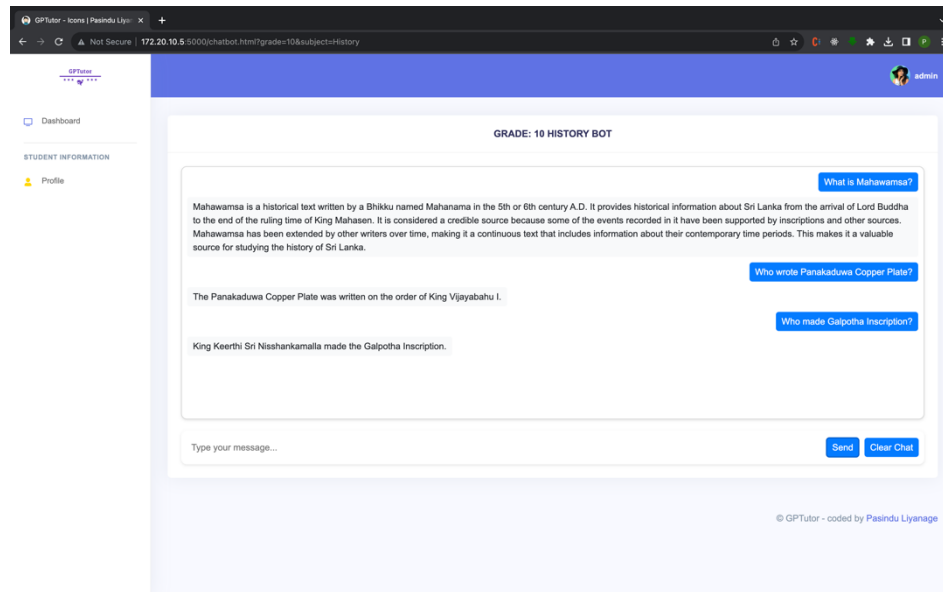


Figure 8: LLM Model Multiple Question Output

Multiple Questions can be asked from the LLM which provides very accurate and precise answers from the document.

04. CONCLUSION & WAYFORWARD

Prototype Development was concluded with highly accurate and successful results. Accordingly the application will be enhanced to support other Subjects as well and Native Language option would also be included to increase the adoption of the tool.