PROJECT REPORT
Spring 2025

Artificial Intelligence Lab

Bahria University
Discovering Knowledge

(AIL-201)

Project Title:

BU Chatbot (Smart Handbook Chatbot)

BS (AI)-4A

Group Members

Name	Enrollment
AFFAN HYDER	02-136232-013
JUNAID AHMED	02-136232-008
MUNEEB-UR-REHMAN	02-136232-034

SUBMITTED TO:

Ms. AMNA IFTIKHAR

BAHRIA UNIVERSITY KARACHI CAMPUS

Department of Computer Science

Table of Contents

Table of Contents 2

Abstract 3

1. Introduction 4

2. Methodology 5

    Framework & Libraries 5

    Data Preparation 5

    Retrieval Chain & Prompt Design 5

    Streamlit Interface 5

    Session Management 5

3. Results and Evaluation 6

4. Discussion 7

5. Conclusion 8

6. Future Work 9

7. Code 10

8. Output 17

Abstract

Reading lengthy handbooks to find one small rule or policy can be frustrating for students. Our project aims to solve this problem by converting the Bahria University Student Handbook into an interactive chatbot. Instead of going through pages of rules, students can now simply ask a question and get instant answers from the handbook. This smart assistant saves time, improves accessibility, and gives students the clarity they need in seconds. The chatbot brings more ease to student life and helps them stay better informed about university policies.

1. Introduction

We all know how overwhelming university handbooks can be. They’re full of useful info, but let’s be honest, no one really wants to dig through them unless they absolutely have to. That’s why we created the BU Chatbot. The idea is simple: instead of flipping through a PDF or searching keywords manually, students can just ask the chatbot. This project makes that search instant and stress-free.

This system aims to reduce the time students waste trying to find specific rules. It makes things easier, smarter, and definitely more student-friendly.

2. Methodology

We followed a well-structured approach to turn a static handbook into an AI chatbot. We used a mix of technologies, data processing, and a neat interface to ensure a smooth experience. Here's how we did it.

Framework & Libraries

- We used Streamlit to build the chatbot's interface because it's lightweight and super easy to deploy.
- The core logic is handled by LangChain, which helps with document processing, text splitting, and retrieval.
- For storing and searching through handbook content, we used FAISS (a vector database).
- Google Generative AI Embeddings generate rich embeddings of the handbook text for semantic similarity.
- ChatGroq is used as the Large Language Model to produce fast and relevant responses.

Data Preparation

- The handbook PDF is loaded using PyPDFLoader, which breaks it down into readable chunks.
- We used a RecursiveCharacterTextSplitter to divide the content into overlapping 1,000-character pieces so that context isn't lost.
- These chunks are then embedded using Google's AI embeddings and stored in a FAISS vector index.

Retrieval Chain & Prompt Design

- When a user asks a question, the chatbot uses the FAISS index to find the most relevant handbook chunks.
- It pulls only the top results that best match the question, not the entire document.
- These results, along with the user's query, are fed into LangChain's document chain and then passed to the LLM.
- The model then forms an answer based strictly on the retrieved content to stay accurate.

Streamlit Interface

- Users input their Groq API Key and Google API Key in the sidebar.
- We styled the interface with a dark theme and bubble-styled chat for a clean, modern look.
- An expandable section shows the actual handbook excerpts used in each answer for full transparency.

Session Management

- We used st.session_state to save chat history and the vector store.
- This avoids the need to reprocess the handbook on every reload, making the experience faster.
- It helps keep conversations consistent and smooth.

3. Results and Evaluation

The chatbot works well in giving precise answers from the handbook. We tested it with many types of queries like policies on attendance, grades, and scholarships. The responses were accurate and based on real content. Students who tried the chatbot found it simple, quick, and much easier than reading through long documents. It really enhances the way students access important academic information.

4. Discussion

Like every good project, we ran into a few bumps along the way. One of the challenges was helping users understand how to input their API keys. We also noticed that the initial embedding process takes some time, although it only happens once. Occasionally, the answers may sound a bit generic if the right content isn’t found in the handbook.

Despite this, the overall experience is smooth, and it really does solve the problem of handbook overload. The feedback we received has been encouraging and motivates us to keep improving.

5. Conclusion

This chatbot brings a smart twist to how students access university rules. It's fast, helpful, and way less stressful than reading through pages. We believe this project has real potential to make a difference in student life, and we’re proud of how far it’s come. From better information access to a more modern student experience, BU Chatbot has shown great promise.

6. Future Work

We’ve got a lot of ideas for making this chatbot even better. Here's what we plan to do next:

- Add voice input support for hands-free queries.
- Allow uploading and reading from more than one document.
- Introduce user login and save chat history for future use.
- Make the chatbot more responsive and user-friendly on mobile devices.

These improvements will help us take the BU Chatbot to the next level and make it even more useful for students.

7. Code

```
import streamlit as st
import os
import tempfile
from langchain_groq import ChatGroq
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.chains.combine_documents import create_stuff_documents_chain
from langchain_core.prompts import ChatPromptTemplate
from langchain.chains import create_retrieval_chain
from langchain_community.vectorstores import FAISS
from langchain_community.document_loaders import PyPDFLoader
from langchain_google_genai import GoogleGenerativeAIEmbeddings

# --------------------------- App Configuration ---------------------------
st.set_page_config(
    page_title="BU Chatbot - Smart Handbook Assistant",
    page_icon="⍰",
    layout="wide"
)

# --------------------------- Custom CSS Styling ---------------------------
st.markdown("""
    <style>
    body, .main {
        background-color: #1e1e1e;
        color: #ffffff;
```

```
}

.stChatMessage {
    background-color: #2b2b2b;
    color: #f0f0f0;
    padding: 15px;
    border-radius: 12px;
    border: 1px solid #444;
    margin-bottom: 10px;
}

.stChatMessage.user {
    background-color: #3a3a3a;
    color: #ffffff;
}

.stChatMessage.assistant {
    background-color: #1f3b4d;
    color: #ffffff;
}

.stTextInput>div>div>input {
    background-color: #333;
    color: #fff;
    border: 1px solid #555;
    }


    .stButton button {
        background-color: #0066cc;
        color: #ffffff;
        font-weight: bold;
        border-radius: 10px;
        padding: 10px 20px;
    }


    .stMarkdown h1, .stMarkdown h2, .stMarkdown h3 {
        color: #ffffff;
    }


    .stExpanderHeader {
        color: #ffffff;
    }


    .sidebar .sidebar-content {
        background-color: #202020;
        color: white;
    }


    </style>
""", unsafe_allow_html=True)
```

```
# -------------------------- Sidebar --------------------------
with st.sidebar:
    st.image("img/download.png", width=200)
    st.markdown("## ⍰ BU Chatbot")
    st.write(
        "An intelligent assistant to help you explore **Bahria University Handbook** policies & rules instantly."
    )

    st.markdown("### ⍰ Settings")
    groq_api_key = st.text_input("⍰ Enter Groq API Key", type="password")
    google_api_key = st.text_input("⍰ Enter Google API Key", type="password")

    st.markdown("---")
    st.info("⍰ Tip: Ask anything about attendance policy, grading, scholarships, etc.")

# -------------------------- App Core --------------------------
st.title("BU Chatbot ⍰")
st.subheader("Your AI Assistant for Bahria University Rules & Policies")

# Load document from static folder
handbook_path = "data/handbook.pdf"

if groq_api_key and google_api_key:
    os.environ["GOOGLE_API_KEY"] = google_api_key
```

```
llm = ChatGroq(groq_api_key=groq_api_key, model_name="gemma2-9b-it")

with st.spinner("⍰ Processing... Please wait."):

    if "vectors" not in st.session_state:
        embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")

        loader = PyPDFLoader(handbook_path)
        raw_docs = loader.load()

        text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
        documents = text_splitter.split_documents(raw_docs)

        st.session_state.vectors = FAISS.from_documents(documents, embeddings)

    prompt = ChatPromptTemplate.from_template("""
    Answer the questions based on the provided context only.
    Please provide the most accurate response.
    <context>
    {context}
    <context>
    Question: {input}
    """)

    document_chain = create_stuff_documents_chain(llm, prompt)
```

```
retriever = st.session_state.vectors.as_retriever()
retrieval_chain = create_retrieval_chain(retriever, document_chain)

if "messages" not in st.session_state:
    st.session_state.messages = []

for msg in st.session_state.messages:
    with st.chat_message(msg["role"]):
        st.markdown(msg["content"])

if user_input := st.chat_input("Ask me..."):
    st.session_state.messages.append({"role": "user", "content": user_input})
    with st.chat_message("user"):
        st.markdown(user_input)

    with st.chat_message("assistant"):
        result = retrieval_chain.invoke({"input": user_input})
        answer = result["answer"]
        st.markdown(answer)

        st.session_state.messages.append({"role": "assistant", "content": answer})

        with st.expander(" Source Context"):
            for i, doc in enumerate(result["context"]):
                st.markdown(f"**Page {i+1}**")
```

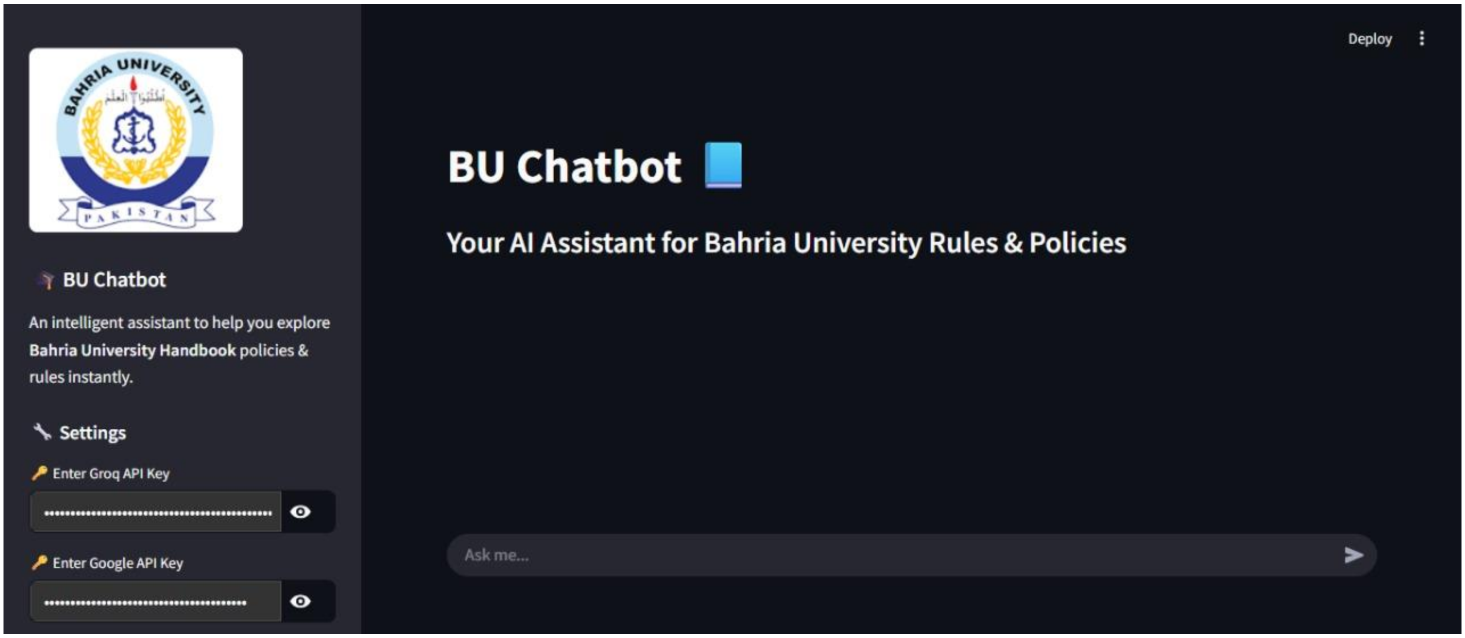
```
                    st.write(doc.page_content)

                    st.markdown("---")

else:

    st.warning("Please enter your API keys in the sidebar to begin.")
```

8. Output



BU Chatbot

An intelligent assistant to help you explore **Bahria University Handbook** policies & rules instantly.

Settings

Enter Groq API Key

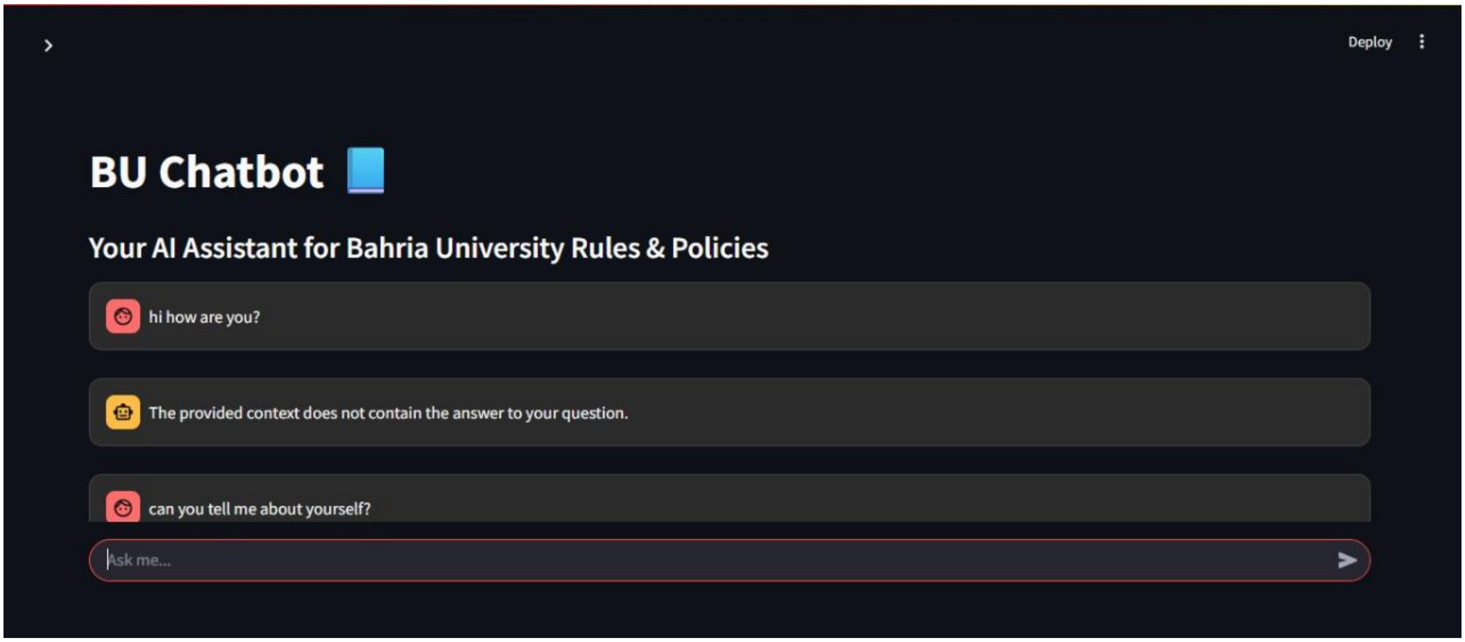
••

Enter Google API Key

•••

BU Chatbot 📘

Your AI Assistant for Bahria University Rules & Policies



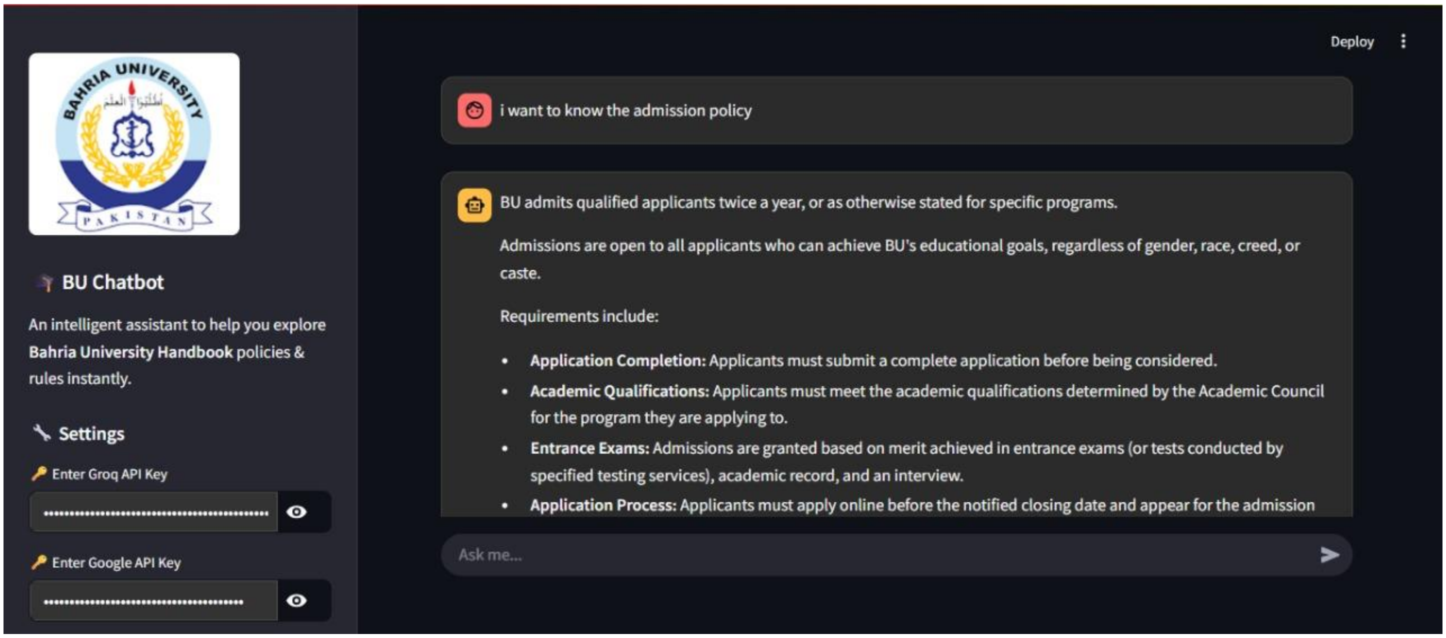
BU Chatbot 📘

Your AI Assistant for Bahria University Rules & Policies

hi how are you?

The provided context does not contain the answer to your question.

can you tell me about yourself?



BU Chatbot

An intelligent assistant to help you explore **Bahria University Handbook** policies & rules instantly.

Settings

Enter Groq API Key

Enter Google API Key

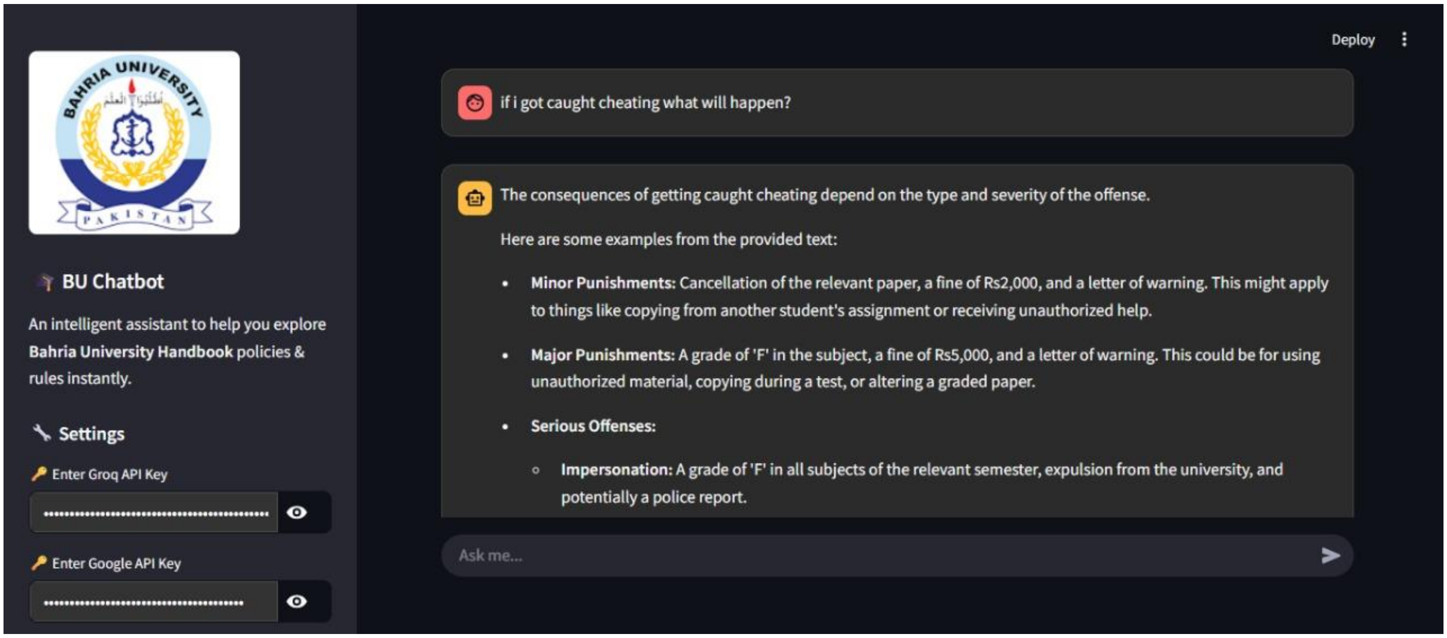
i want to know the admission policy

BU admits qualified applicants twice a year, or as otherwise stated for specific programs.

Admissions are open to all applicants who can achieve BU's educational goals, regardless of gender, race, creed, or caste.

Requirements include:

- **Application Completion:** Applicants must submit a complete application before being considered.
- **Academic Qualifications:** Applicants must meet the academic qualifications determined by the Academic Council for the program they are applying to.
- **Entrance Exams:** Admissions are granted based on merit achieved in entrance exams (or tests conducted by specified testing services), academic record, and an interview.
- **Application Process:** Applicants must apply online before the notified closing date and appear for the admission



BU Chatbot

An intelligent assistant to help you explore **Bahria University Handbook** policies & rules instantly.

Settings

Enter Groq API Key

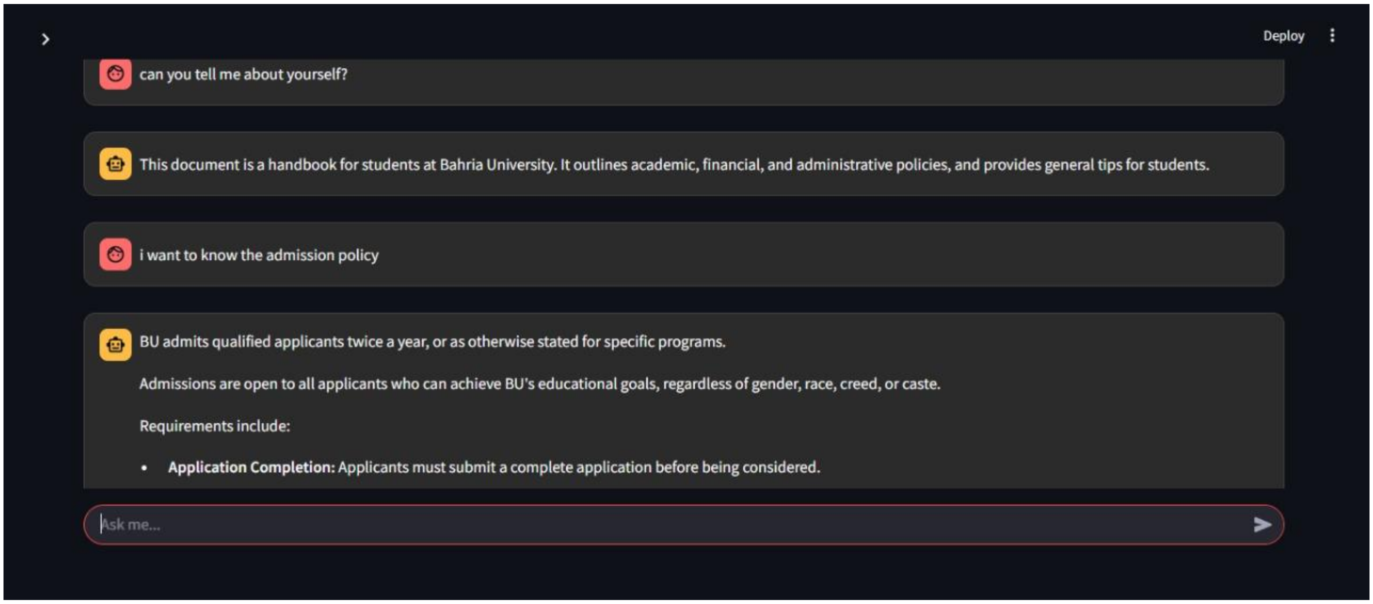
Enter Google API Key

if i got caught cheating what will happen?

The consequences of getting caught cheating depend on the type and severity of the offense.

Here are some examples from the provided text:

- **Minor Punishments:** Cancellation of the relevant paper, a fine of Rs2,000, and a letter of warning. This might apply to things like copying from another student's assignment or receiving unauthorized help.
- **Major Punishments:** A grade of 'F' in the subject, a fine of Rs5,000, and a letter of warning. This could be for using unauthorized material, copying during a test, or altering a graded paper.
- **Serious Offenses:**
  - **Impersonation:** A grade of 'F' in all subjects of the relevant semester, expulsion from the university, and potentially a police report.



can you tell me about yourself?

This document is a handbook for students at Bahria University. It outlines academic, financial, and administrative policies, and provides general tips for students.

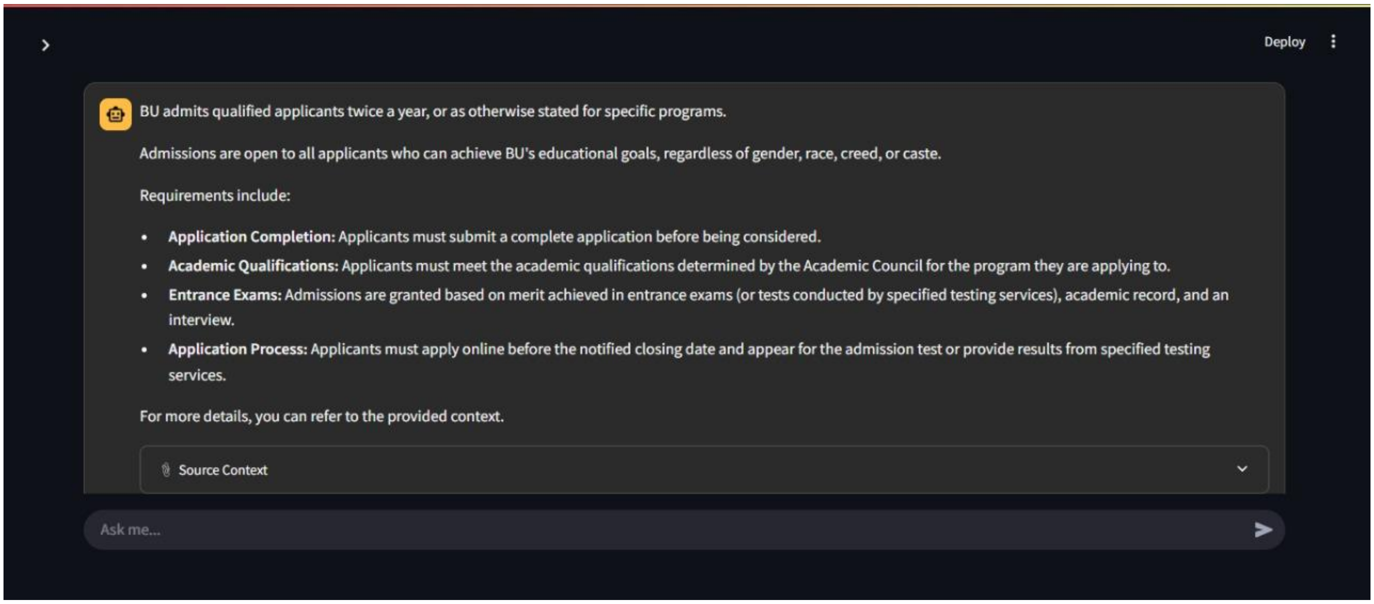
i want to know the admission policy

BU admits qualified applicants twice a year, or as otherwise stated for specific programs.

Admissions are open to all applicants who can achieve BU's educational goals, regardless of gender, race, creed, or caste.

Requirements include:

- **Application Completion:** Applicants must submit a complete application before being considered.



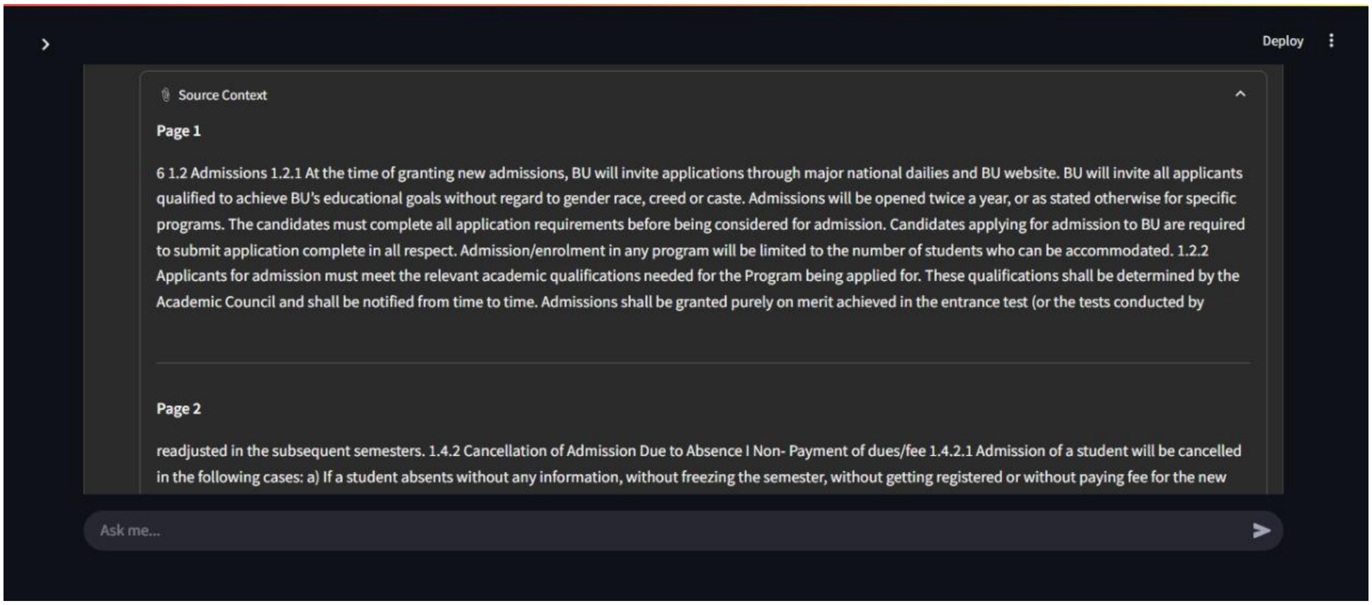
BU admits qualified applicants twice a year, or as otherwise stated for specific programs.

Admissions are open to all applicants who can achieve BU's educational goals, regardless of gender, race, creed, or caste.

Requirements include:

- **Application Completion:** Applicants must submit a complete application before being considered.
- **Academic Qualifications:** Applicants must meet the academic qualifications determined by the Academic Council for the program they are applying to.
- **Entrance Exams:** Admissions are granted based on merit achieved in entrance exams (or tests conducted by specified testing services), academic record, and an interview.
- **Application Process:** Applicants must apply online before the notified closing date and appear for the admission test or provide results from specified testing services.

For more details, you can refer to the provided context.



Page 1

6 1.2 Admissions 1.2.1 At the time of granting new admissions, BU will invite applications through major national dailies and BU website. BU will invite all applicants qualified to achieve BU's educational goals without regard to gender race, creed or caste. Admissions will be opened twice a year, or as stated otherwise for specific programs. The candidates must complete all application requirements before being considered for admission. Candidates applying for admission to BU are required to submit application complete in all respect. Admission/enrolment in any program will be limited to the number of students who can be accommodated. 1.2.2 Applicants for admission must meet the relevant academic qualifications needed for the Program being applied for. These qualifications shall be determined by the Academic Council and shall be notified from time to time. Admissions shall be granted purely on merit achieved in the entrance test (or the tests conducted by

---

Page 2

readjusted in the subsequent semesters. 1.4.2 Cancellation of Admission Due to Absence I Non- Payment of dues/fee 1.4.2.1 Admission of a student will be cancelled in the following cases: a) If a student absents without any information, without freezing the semester, without getting registered or without paying fee for the new

9. Project link

https://github.com/affan1311/BuChatbot