AI-Powered Java Chatbot
Pookie Chatbot

Subho Hazra

Mail ID : subho.hazra2003@gmail.com

Github ID : https://github.com/SubhoHazra07

Contact No : +91 98315 81640

Table of Contents

Introduction

P**ookie Chatbot** is an AI-powered conversational assistant designed to provide intelligent responses to user queries. It leverages advanced **natural language processing (NLP)** techniques and integrates with **Google Gemini AI** for enhanced interaction. The chatbot is built using **Streamlit** for the frontend and utilizes **Pinecone** for efficient vector search operations.

Project Overview

The Pookie Chatbot project aims to create a **Java-based AI chatbot** that seamlessly integrates structured query processing with natural language understanding. It refines user inputs, retrieves relevant context from a vector database, and generates accurate responses using Google Gemini AI. The chatbot supports Java Queries for processing structured data-based queries, enabling it to interact with databases and execute specific commands effectively. With its advanced NLP capabilities, it maintains a dynamic conversation history to enhance user experience and response accuracy.

Tech Stacks

The project utilizes the following tech stacks to achieve its objectives

- **Python:** The primary programming language used for developing the chatbot and integrating various components.
- **NLP (Natural Language Processing):** Employed to analyze and understand user input, allowing the chatbot to provide relevant responses and process data.
- **NLTK (Natural Language Toolkit):** Used for text processing, tokenization, vectorization, and language analysis.
- **Streamlit:** Enables users to interact with the chatbot through a user-friendly web interface.
- **Langchain:** Enhances the chatbot's language capabilities, improving the quality of responses and processing large text files into vector embeddings.
- **Google Gemini API:** Integrated to provide advanced language processing capabilities and generate human-like responses.
- **Pinecone (Vector Store):** Utilized for efficient storage and retrieval of vectorized data, improving the chatbot's search and response speed.
- **GitHub:** The project is maintained in a private repository for version control and collaboration.

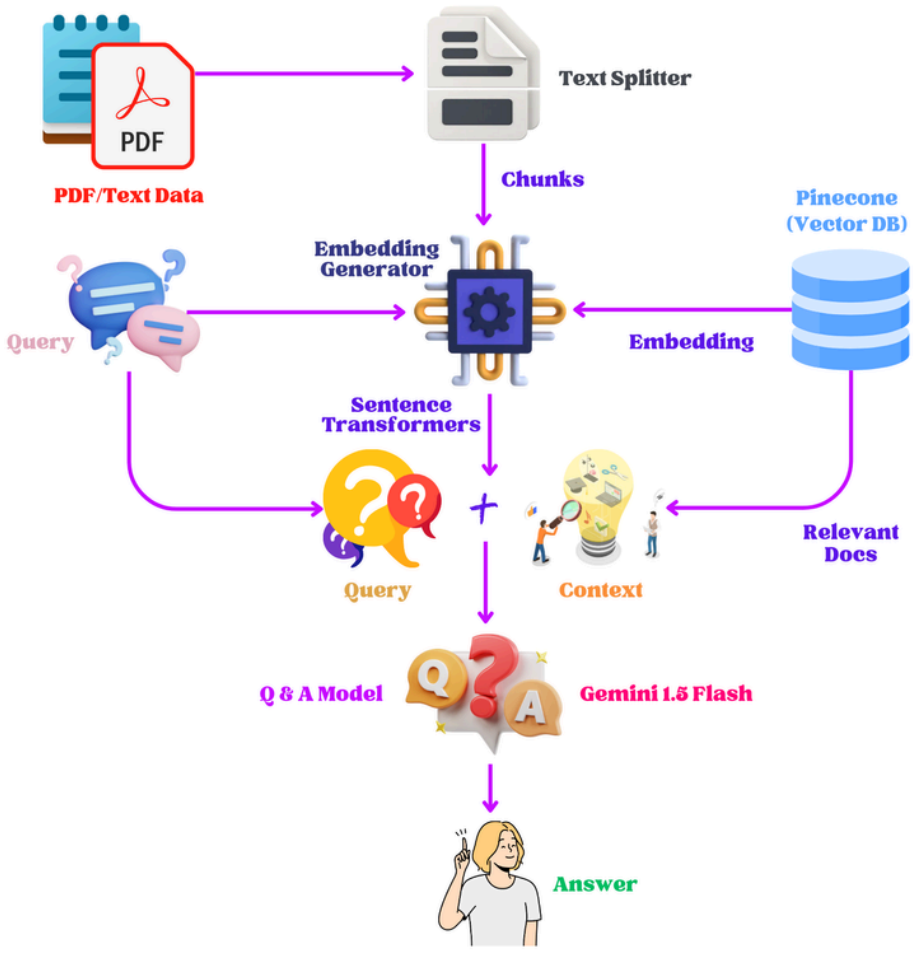
Features

The AI-Powered Java Chatbot offers the following features :

- **Question Answering:** Provides precise and context-aware answers to Java-related queries, including syntax, best practices, and coding techniques.
- **Explanation Generation:** Generates detailed explanations for complex Java concepts, such as object-oriented programming, data structures, and multithreading.
- **Interactive Conversations:** Engages users with a natural, dynamic, and multi-turn conversation flow, making Java learning more intuitive.
- **Personalization:** Adapts to user preferences and maintains context for improved response accuracy, tailoring explanations based on user expertise.
- **Multi-Modal Interface:** Supports text-based interactions with potential expansion to code snippet generation and debugging assistance.
- **Advanced Language Capabilities:** Handles complex Java-related queries, multi-turn conversations, contextual reasoning, and sentiment-based responses for a more human-like interaction.

Architecture

The architecture of the AI-Powered Java Chatbot consists of the following components :



- **User Input:** Users ask Java-related questions through the Streamlit-based web interface.
- **Query Processing:** The chatbot understands the user's question using NLP and breaks it down into meaningful components.
- **Text Chunking:** Relevant Java-related reference materials (PDFs, documentation, etc.) are split into smaller text chunks for better retrieval.
- **Embedding Generation:** Each chunk is converted into vector embeddings using Sentence Transformers.
- **Vector Storage & Retrieval:** The embeddings are stored in Pinecone, a vector database, allowing for fast and accurate retrieval of relevant documents.
- **Contextual Search:** When a user asks a question, the chatbot retrieves the most relevant text chunks from Pinecone to provide accurate answers.
- **Response Generation:** The chatbot sends the retrieved context to Google Gemini AI, which formulates a structured and contextually relevant response.
- **Answer Presentation:** The chatbot presents the answer in an easy-to-understand format, including text explanations, code snippets, or step-by-step solutions.
- **Session Management:** The chatbot maintains a conversation history, allowing users to ask follow-up questions and get coherent responses.
- **Continuous Improvement:** The chatbot refines its responses over time by analyzing user interactions and feedback.

Installation and Setup

To set up the AI-Powered Java Chatbot on your local machine follow these steps :

- Clone the repository from Github :
  https://github.com/SubhoHazra07/Java-AI-Chatbot.git
- Install the required Python dependencies using :
  pip install -r requirements.txt
- Set up Pinecone and create a vectore store for Java concepts.
- Obtain API keys for Pinecone and Gemini API.
- Replace the API keys in the appropiate configuration file (.env) :
  GOOGLE_API_KEY=your_google_api_key
  PINECONE_API_KEY=your_pinecone_api_key
- Now run the chatbot using :
  streamlit run main.py

Usage

Once the setup is complete, follow these steps to use the Chatbot :

Streamlit App : **https://pookie-chatbot.streamlit.app/**

Github Repository : **https://github.com/SubhoHazra07/Java-AI-Chatbot.git**

Using Streamlit :

- Open the chatbot directly in your browser using the following link : **Pookie-Chatbot**
- Access the Chatbot interface via your web browser.
- Enter your Java-related questions and topics.
- Engage in interactive conversation with the chatbot.
- Explore answers, explanations, and personalized learning experiences.

Using Github :

- Clone the Github Repository : **Java-AI-Chatbot**
- Go to the terminal.
- Run the file using : **streamlit run main.py**
- Access the Chatbot interface via your web browser.
- Enter your Java-related questions and topics.
- Engage in interactive conversation with the chatbot.
- Explore answers, explanations, and personalized learning experiences.

Deployment

For an easy and hassle-free deployment, you can host the chatbot on Streamlit Community Cloud by following these steps:

- **Push Code to GitHub:** Ensure that your chatbot code is available in a public GitHub repository.
- Do not upload your **.env file** containing sensitive **API keys or credentials to GitHub.**
- **Set Up Secret Keys (Instead of Uploading .env) :** In **Advanced Settings > Secrets,** add your API keys
- **Sign in to Streamlit Cloud:** Go to Streamlit Community Cloud and log in with your GitHub account.

- **Deploy the Application:**
  a. Click on **"New App."**
  b. Select the **GitHub repository** where your chatbot is stored.
  c. Choose the main script file **(e.g., main.py)**.
  d. Click **"Deploy."**

- **Access the Deployed Chatbot:** Once deployed, you'll get a unique Streamlit link that allows users to access your chatbot instantly.

- **Update & Maintain:** Any updates pushed to the GitHub repository will automatically reflect in the deployed chatbot (after re-running the app).

Conclusion

Pookie Chatbot is an AI-powered assistant that helps users with **Java-related queries** using **NLP, Google Gemini AI,** and **Pinecone** for fast, accurate responses. With an easy-to-use **Streamlit interface**, it enables interactive conversations, making Java learning simpler and more efficient. The chatbot improves over time and has potential for future upgrades like better code analysis and multi-language support, making it a useful tool for developers and learners.

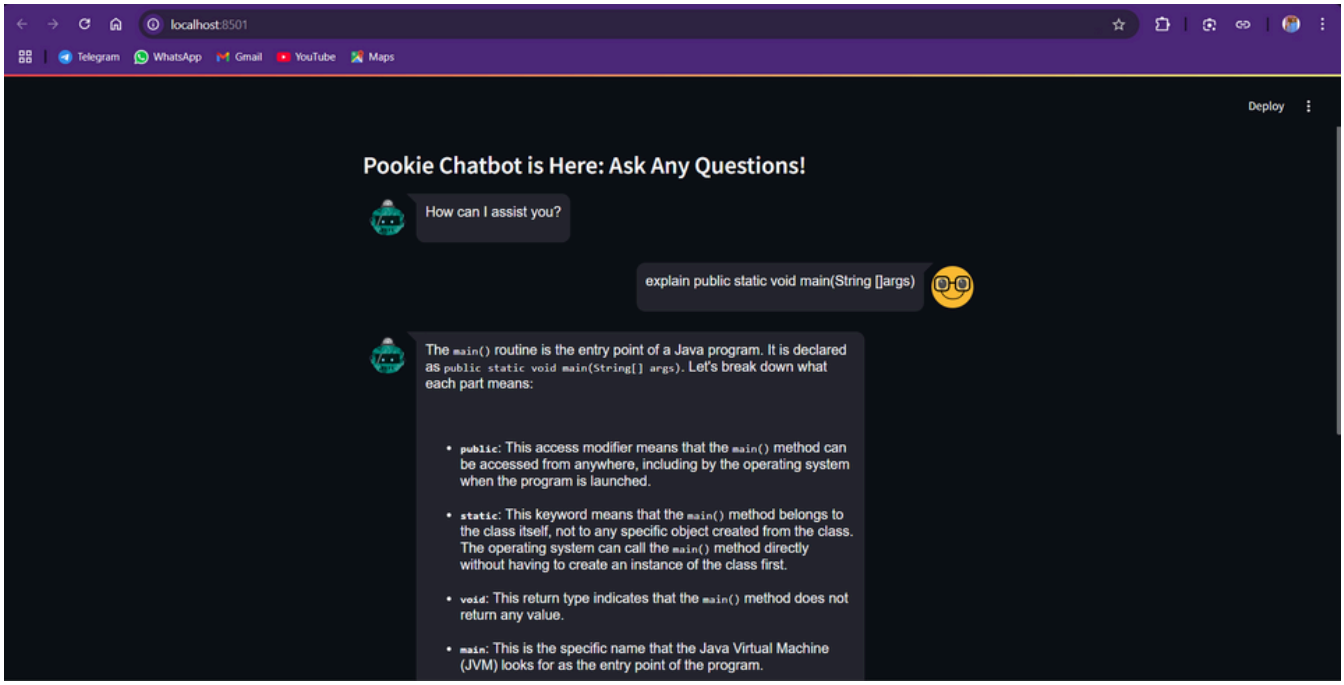
Acknowledgments

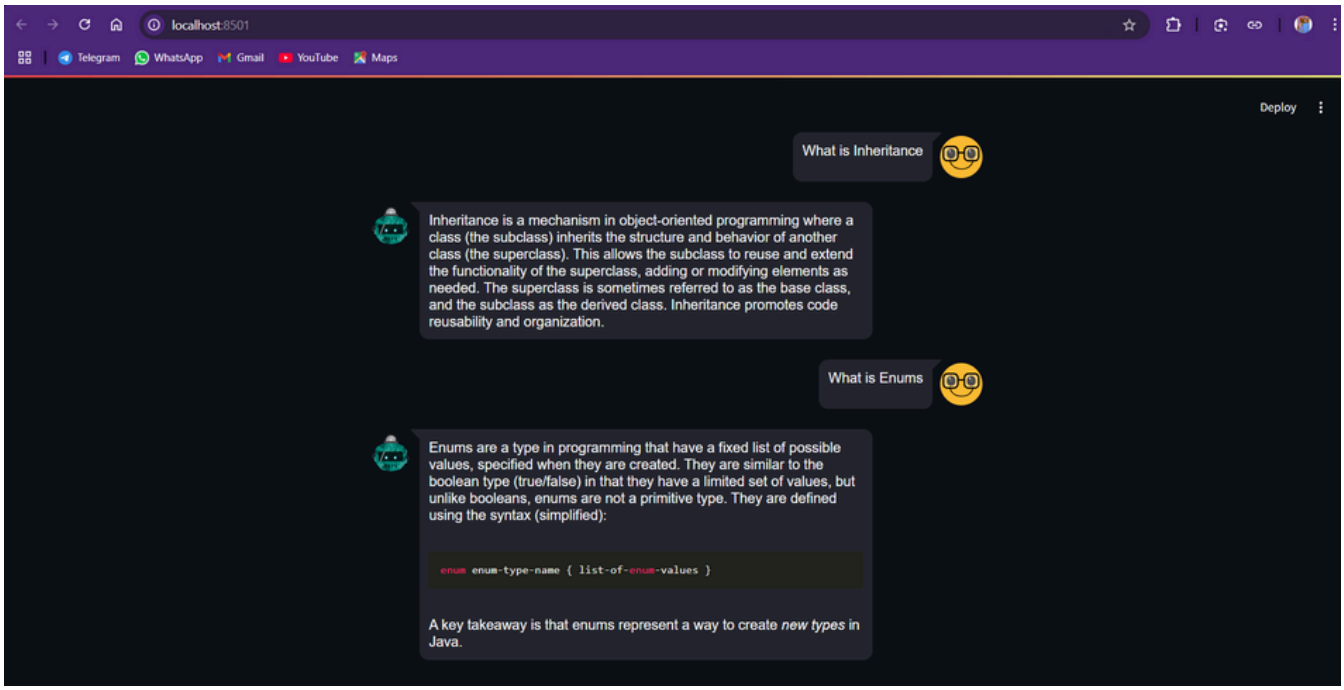
We would like to express our gratitude to the contributors, open-source communities, and developers who have contributed to the technologies and frameworks used in this project.

References

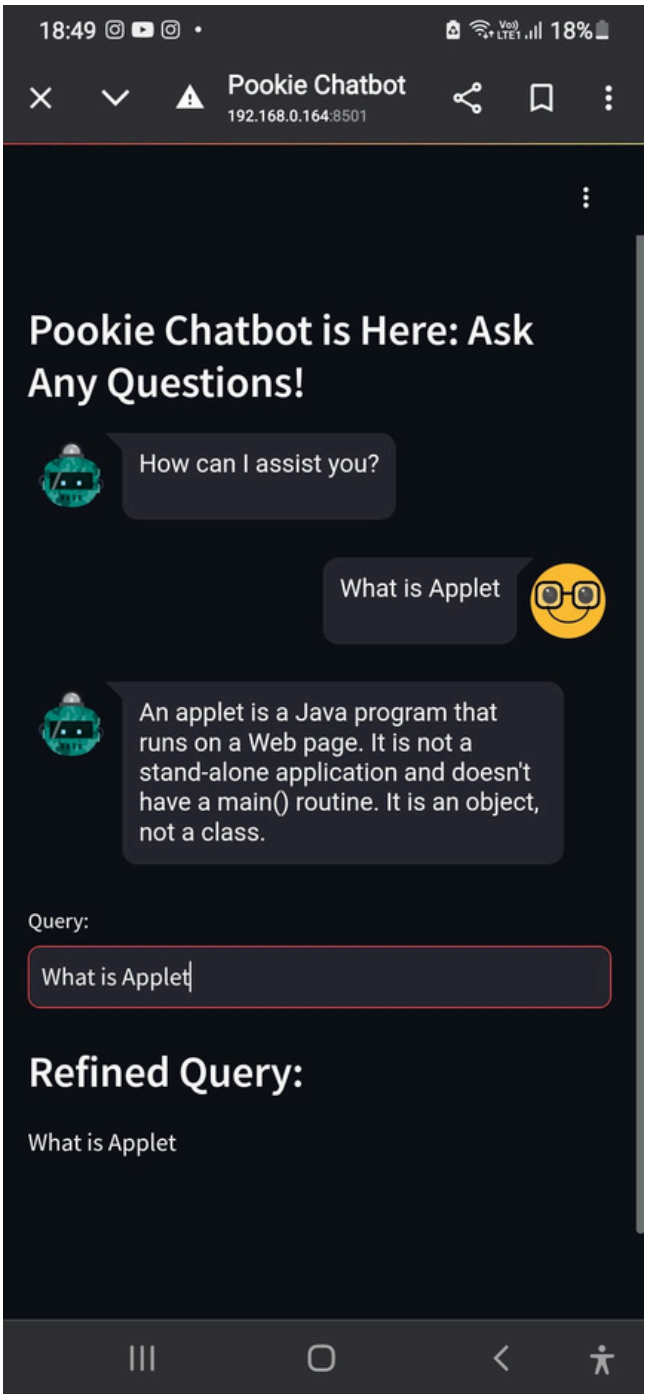
- NLTK Documentation: **https://www.nltk.org/**
- Streamlit Documentation: **https://docs.streamlit.io/**
- Langchain Documentation: **https://langchain.ai/docs/**
- Google Gemini API Documentation: **https://ai.google.dev/gemini-api/docs**
- Pinecone Documentation: **https://www.pinecone.io/docs/**
- Streamlit Cloud Deployment: **https://docs.streamlit.io/deploy/streamlit-community-cloud**
- Data 1 :
  https://iasyc.in/download/book/Programming_With_Java_A_primer_3e_by_balagurusamy.pdf
- Data 2 :
  https://www.iitk.ac.in/esc101/share/downloads/javanotes5.pdf

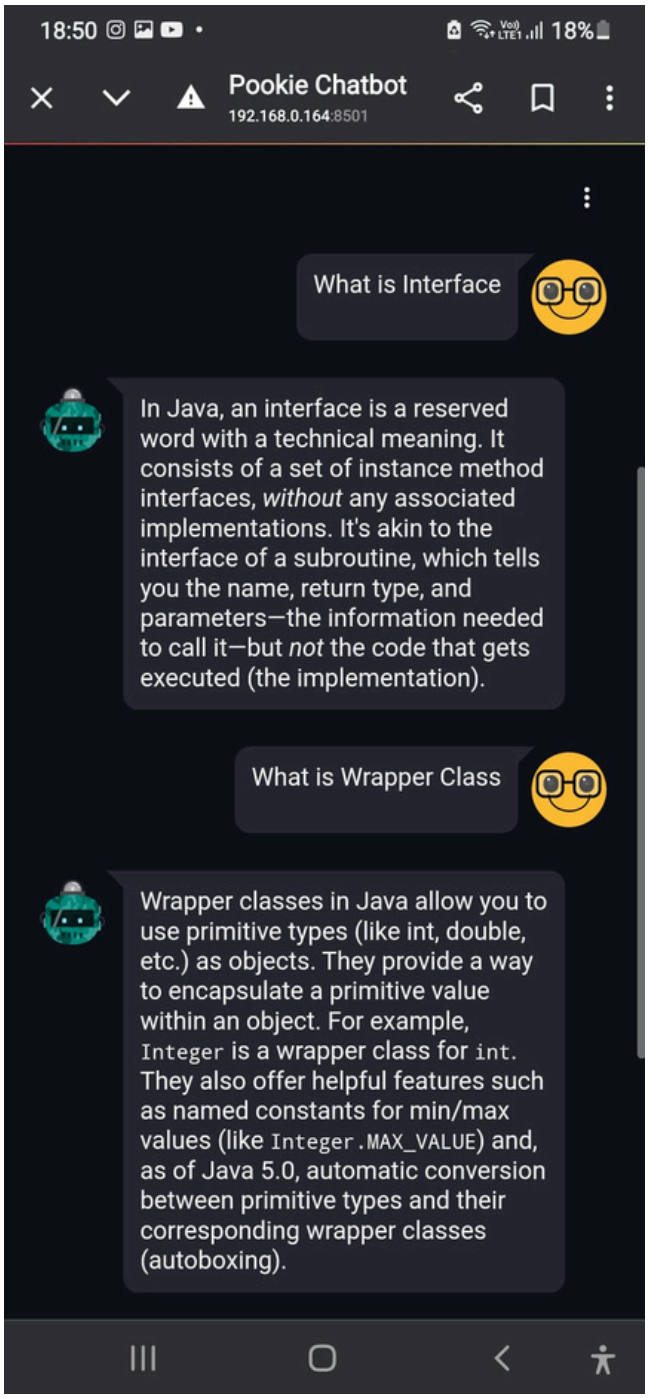
Demo Chat with Pookie🎀





Demo Chat with Pookie🎀





Demo Chat with Pookie🎀