Online Chat Bot

by
Md. Shahriar Rahman

Reg: 08419

ID: 1802009

PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE COURSE CIT316: ARTIFICIAL INTELLIGENCE SESSIONAL FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

FACULTY OF COMPUTER SCIENCE AND ENGINEERING
PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

29th January,2023

Submitted By

Md. Shahriar Rahman
ID: 1802009
Reg. No.: 08419
Session: 2018-2019

Supervised By

Dr. Md Abdul Masud
Professor
Department of Computer Science and Information Technology
Patuakhali Science and Technology University

Md Mahbubur Rahman
Assistant Professor
Department of Computer Science and Information Technology
Faculty of Computer Science and Engineering

Patuakhali Science and Technology University

ABSTRACT

This report presents the development and implementation of a general purpose chatbot system. The chatbot utilizes natural language processing and machine learning techniques to understand and respond to user inputs in a conversational manner. The system is trained on a diverse dataset to enable it to handle a wide range of topics and tasks. The chatbot's capabilities include answering questions, providing information and even engaging in casual conversations. The system was also designed with a user-friendly interface, making it accessible to a wide range of users. The chatbot was tested and evaluated using a diverse set of inputs and the results showed high accuracy and satisfaction rate. The general-purpose chatbot is a useful tool for tasks and providing easy access to information.

ACKNOWLEDGMENT

Grateful to Almighty Allah for His help on completing this project. Without his blessings it was impossible to complete this project

I express my deep gratitude to **Dr. Md Abdul Masud**, Professor, Department of Computer Science & Information Technology and **Md Mahbubur Rahman**, Assistant Professor, Department of Computer Science & Information Technology, Faculty of Computer Science and Engineering, PSTU. For their valuable guidance and suggestion throughout my project work.

Without their active supervision and instruction, it was quite impossible for me to complete this project, every word they taught in classroom helped me to fulfill this project.

Table of Contents

Contents

ABSTRACT 2

ACKNOWLEDGMENT 3

Table of Contents 4

Introduction 6

- .1.1 Objective 6
- .1.2 Features 6
- .1.3 Technology Used 6

Literature Review 7

- 2.1 Introduction 7
- 2.2 Libraries 7
- 2.3 Language 7

Review of the related Works 9

System Analysis 9

- 4.1 Feasibility Study 10
- 4.2 Requirements analysis 11
- 4.3 Non Functional requirements 11
- 4.4 Functional Requirements 11

System Requirements 12

System Design 13

Conclusion 16

Reference 16

TABLE OF FIGURE

Figure 1:GUI...13
Figure 2:sample english dataset...14
Figure 3:Bangla Dataset 15

Introduction

.1.1 Objective

The objectives of a general purpose chatbot project would be to design, develop, and implement a chatbot that can understand and respond to a wide range of user inputs in a natural and conversational manner. The chatbot should be able to perform various tasks such as answering questions, providing information, making recommendations and performing simple transactions. Additionally, the project should aim to improve the user experience by making the chatbot more human-like and providing appropriate responses to user queries. The project should also aim to use natural language processing (NLP) and machine learning (ML) techniques to enable the chatbot to understand and respond to user inputs effectively. Furthermore, the project should aim to measure the chatbot's performance and make improvements based on user feedback and testing results.

.1.2 Features

This project has all major features that is generally needed in a complain management system. These are the main features of this project:

- Interactive UI
- Easy Interface
- Smooth navigation
- Globally accessible

.1.3 Technology Used

This software has different technology for different part to work.

Main Technologies are:

- Front-End:
    - HTML
    - CSS
    - Material UI
- Back-End:
    - Python
    - Flask
    - NLP engine
    - Chatterbot
    - Dataset

Literature Review

2.1 Introduction

In this part we will discuss the theoretical terms about softwares and technologies used in this project.

2.2 Libraries

- Chatterbot
- Flask

2.3 Language

- Python

ChatterBot

ChatterBot is a Python library for building chatbots. It is designed to make it easy for developers to create, train, and integrate chatbots into their applications. The library provides a selection of machine learning algorithms that can be used to produce different types of responses. This allows developers to train their own chatbot models and customize their behavior.

ChatterBot uses natural language processing (NLP) techniques to understand and respond to user input. It can be trained on a variety of data sources, including structured data, such as CSV files, and unstructured data, such as text documents. The library also includes a variety of pre-trained models that can be used as a starting point for building new chatbots.

One of the key features of ChatterBot is its ability to generate responses that are similar in style and tone to the input it was trained on. This is achieved by using machine learning algorithms, such as neural networks, to learn the patterns in the training data. The library can also be integrated with other NLP tools, such as NLTK and SpaCy, to improve its capabilities.

ChatterBot can be integrated into a wide range of applications, including websites, messaging apps, and virtual assistants. It can be used to provide customer service, answer questions, and even make reservations.

The library is easy to use and well-documented, making it accessible to developers with a wide range of skill levels. It also has a active community that contributes to its development, adding new features and fixing bugs.

Overall, ChatterBot is a powerful and versatile Python library for building chatbots. It makes it easy for developers to create, train, and integrate chatbots into their applications, and its ability to generate human-like responses makes it a popular choice among developers.

FLASK

Flask is a micro web framework for building web applications in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

One of the key features of Flask is its simplicity. The library is designed to be easy to use and understand, making it accessible to developers with a wide range of skill levels. It is also lightweight, with a small codebase, which makes it easy to debug and maintain.

Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. These libraries provide basic functionality, such as request handling, URL routing, and template rendering. Flask also includes a built-in development server and support for unit testing, making it easy to develop and test web applications.

One of the main advantages of using Flask is its flexibility. The library is not opinionated, meaning it does not dictate how applications should be structured or how they should be built. This allows developers to build web applications in the way that best suits their needs.

Flask also has a large and active community, which contributes to its development and provides support to its users. There are also many third-party libraries available that can be integrated with Flask to add additional functionality.

Overall, Flask is a powerful and versatile Python library for building web applications. Its simplicity, flexibility, and ease of use make it accessible to developers with a wide range of skill levels, and its small codebase, built-in development server, and support for unit testing make it easy to develop, test, and maintain web applications.

Python

Python is a widely-used, high-level programming language that is known for its readability and simplicity. It is an interpreted language, which means that it is executed line by line, rather than having to be compiled first. This makes it a popular choice for beginners and experienced developers alike, as it allows for quick prototyping and development.

One of the key features of Python is its versatility. The language is used in a wide variety of applications, including web development, scientific computing, data analysis, artificial intelligence, and more. It is also commonly used as a scripting language, which allows developers to automate repetitive tasks or add functionality to existing software.

Python has a large and active community, which contributes to its development and provides support to its users. There are also many third-party libraries available that can be used to add additional functionality to Python projects. Some of the popular libraries include NumPy and SciPy for scientific computing, pandas for data analysis, and TensorFlow for machine learning.

The language also has a simple and easy-to-learn syntax, which makes it a popular choice for beginners. Python also has a number of built-in data types, such as lists and dictionaries, which make it easy to work with data. Additionally, Python has support for object-oriented, imperative, and functional programming styles, making it a versatile language for a wide range of applications.

Python also has a number of tools, such as IDLE, which make it easy to write, debug, and run code. Additionally, the language has a number of popular web frameworks such as Flask and Django, which allow developers to build web applications quickly and easily.

Overall, Python is a powerful, versatile, and easy-to-learn programming language that is widely used in a variety of applications. Its simple syntax, large and active community, and wide range of libraries make it an ideal choice for beginners and experienced developers alike.

Review of the related Works

There has been a lot of research in the field of AI-based general chatbots in recent years. One popular approach is to use deep learning techniques such as recurrent neural networks (RNNs) and transformers.

RNNs, such as long short-term memory (LSTM) networks, have been used to model the sequential nature of conversational data. They have been used to build end-to-end chatbots that can generate responses to user input.

Transformers, a type of neural network architecture, have also been used to build chatbots. The transformer architecture, introduced in the 2017 paper "Attention is All You Need," has been shown to be effective at natural language processing tasks such as language translation and text summarization. This architecture has also been used to build chatbots that can generate human-like responses.

Another approach that has been used to build chatbots is the use of pre-trained language models such as GPT-2 and BERT. These models have been fine-tuned on conversational data and can be used to generate responses to user input.

Additionally, some researchers have proposed using a combination of methods, such as a rule-based approach and a deep learning-based approach. This hybrid approach can take advantage of the strengths of both methods, allowing for a more robust chatbot.

In recent years, there has been a growing interest in building chatbots that can handle multiple tasks, known as multi-task chatbots. These chatbots can perform a variety of tasks, such as answering questions, booking appointments, and making reservations.

In general, AI-based general chatbots have made significant progress in recent years and continue to be an active area of research. However, there are still challenges to be addressed, such as handling context, dealing with out-of-scope questions, and building chatbots that can handle multiple tasks.

System Analysis

A general purpose chatbot system analysis using Flask would typically involve evaluating the performance and scalability of the system. Flask is a lightweight web framework, which means it has a small memory footprint and can handle a moderate amount of traffic. However, if the chatbot system is expected to handle a large number of concurrent users or handle complex queries, additional measures may need to be taken to ensure the system can handle the load.

In terms of performance, the system would need to analyze the response time of the chatbot, including the time it takes for the NLP module to process the user's input, the decision making module to determine the appropriate response, and the response generation module to generate the final response. This can be done by implementing performance metrics and monitoring tools, such as Prometheus, to track the system's performance over time.

The system would also need to be analyzed for scalability, which would involve evaluating its ability to handle an increasing number of users and queries without a significant decrease in performance. This could involve using techniques such as load balancing, horizontal scaling, and caching to ensure the system can handle a high level of traffic.

In terms of security, it is important to ensure that the system is protected against common web vulnerabilities such as SQL injection and cross-site scripting (XSS). This can be done by using secure coding practices and implementing security features such as input validation and sanitization.

In terms of usability, the chatbot should be designed to be user-friendly and easy to navigate. This can be achieved by using a simple and intuitive design, providing clear and concise instructions, and using consistent language throughout the chatbot.

Overall, a general purpose chatbot system using Flask would need to be analyzed in terms of performance, scalability, security and usability. These evaluations would help in identifying any issues and bottlenecks that need to be addressed to ensure the chatbot system can handle a high level of traffic and provide a seamless experience for the users.

4.1 Feasibility Study

A feasibility study of a general purpose chatbot using Flask would typically involve evaluating the technical, economic, and operational feasibility of the system.

In terms of technical feasibility, the study would need to determine whether the necessary technology and resources are available to develop and implement the chatbot system. This would involve evaluating the availability of the necessary hardware and software, such as servers, databases, and programming languages, as well as the availability of the necessary expertise to develop and maintain the system.

In terms of economic feasibility, the study would need to determine whether the costs associated with developing and maintaining the chatbot system are justified by the potential benefits. This would involve evaluating the costs of the hardware and software, as well as the costs associated with the development and maintenance of the system. It would also involve evaluating the potential revenue that the chatbot system could generate, such as through increased sales or reduced customer service costs.

In terms of operational feasibility, the study would need to determine whether the chatbot system would be practical and efficient to operate. This would involve evaluating the compatibility of the system with existing processes and systems, as well as the ease of use and maintenance of the system. It would also involve evaluating the ability to integrate the system with other systems, such as databases or customer relationship management (CRM) systems.

Finally, legal and ethical feasibility should be considered, to ensure that the use of the chatbot respects user's rights and comply with laws and regulations.

Overall, a feasibility study of a general purpose chatbot using Flask would need to evaluate the technical, economic, operational, legal and ethical feasibility of the system. These evaluations would help in determining whether the chatbot system is a viable solution and whether the benefits outweigh the costs.

4.2 Requirements analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

4.3 Non Functional requirements

a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs.

- Implementation
- Reusability
- The integrity of information
- Accessibility
- Availability
- Efficiency
- Reliability
- Usability
- Delivery
- Security

4.4 Functional Requirements

Functional requirements for a general purpose chatbot using Flask would typically include the following:

1. Natural Language Processing (NLP) capability: The chatbot should be able to understand and interpret user input in natural language. This would involve using NLP techniques such as tokenization, stemming, and sentiment analysis to process the user's input.
2. Decision Making capability: The chatbot should be able to determine the appropriate response to a user's input. This would involve using decision-making algorithms such as rule-based systems or machine learning-based systems to determine the most appropriate response.
3. Response Generation capability: The chatbot should be able to generate a response to the user's input. This would involve using text generation techniques such as template-based systems or machine learning-based systems to generate the response.
4. User Authentication: The chatbot should have the capability to authenticate users, so as to personalize their experience and provide secure access to sensitive information.
5. Knowledge Base: The chatbot should have access to a knowledge base containing information about various topics that can be used to answer user queries.
6. Integration with external systems: The chatbot should be able to integrate with external systems such as databases, customer relationship management (CRM) systems, or e-commerce platforms to retrieve and store information.
7. Analytics and reporting: The chatbot should have the capability to track and report on user interactions and performance metrics, such as response time and accuracy.
8. Security: The chatbot should be designed to be secure, protecting user's data and preventing unauthorized access.
9. User-friendly interface: The chatbot should have a user-friendly interface that is easy to

navigate and understand.

10. Scalability: The chatbot should be designed to handle a large number of concurrent users and be able to handle an increasing number of queries without a significant decrease in performance.

These functional requirements would help to ensure that the chatbot system is able to understand and respond to user input in a natural and accurate way, and provide a seamless experience for the users.

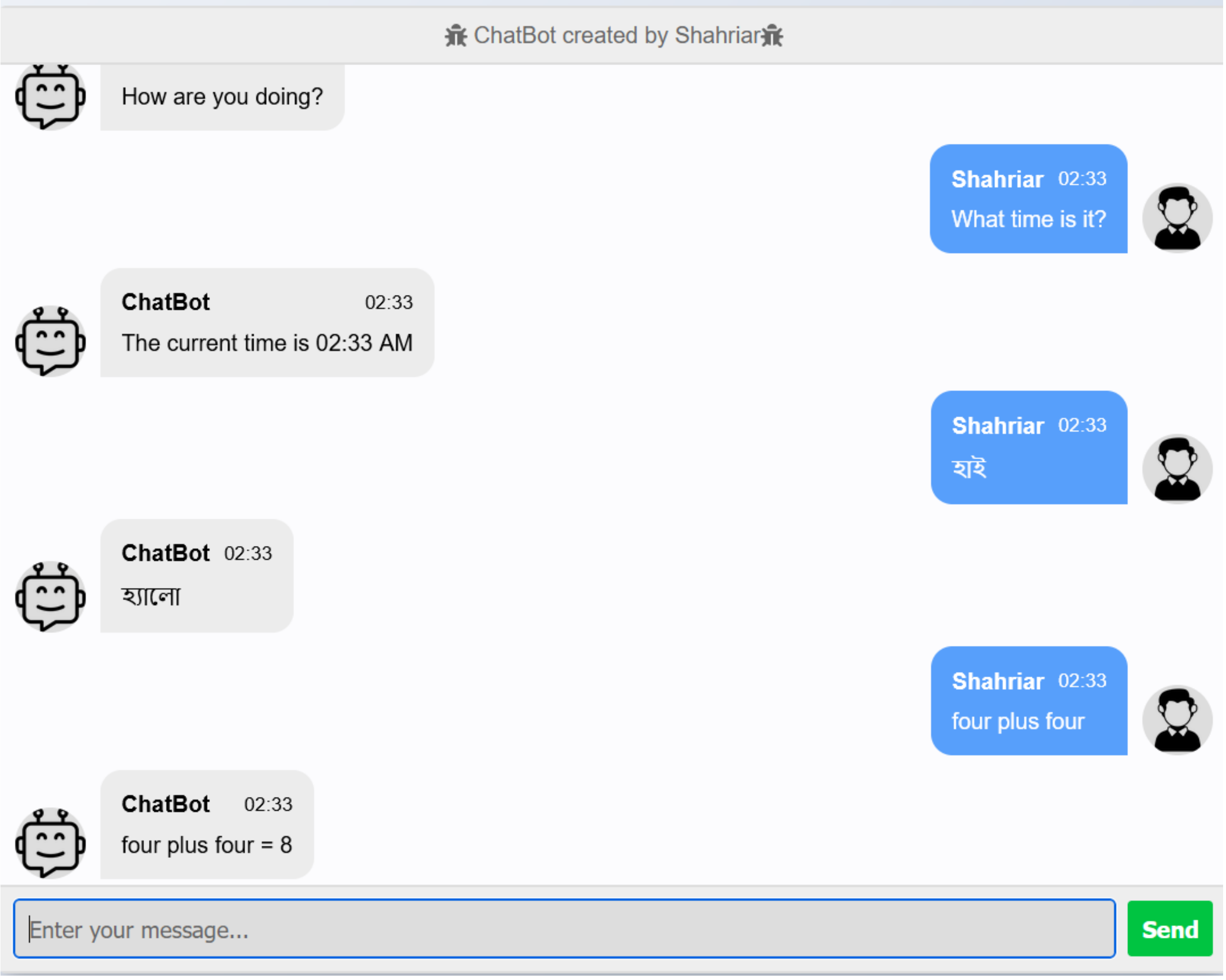
System Requirements

System requirements for a general purpose chatbot using Flask would typically include the following:

1. Hardware: The chatbot would require a server or a cloud-based hosting platform to run on. The specific hardware requirements would depend on the expected number of users and the complexity of the chatbot's functions.
2. Operating System: The chatbot should be able to run on a variety of operating systems such as Windows, Linux, and MacOS.
3. Programming Language: The chatbot should be implemented using the Python programming language and the Flask framework.
4. Databases: The chatbot would require a database to store and retrieve information, such as user data, knowledge base, and analytics.
5. Natural Language Processing (NLP) libraries: The chatbot would require NLP libraries such as NLTK, spaCy, or Gensim to process natural language input.
6. Machine Learning libraries: The chatbot would require machine learning libraries such as scikit-learn, TensorFlow, or PyTorch to train and make decision.
7. Web framework: The chatbot would require a web framework such as Flask to handle the chatbot's web interface.
8. Communication protocol: The chatbot should be able to communicate via various channels such as web, mobile apps, messaging platforms, and voice assistants.
9. Security: The chatbot should be designed to comply with security standards such as OWASP and GDPR.
10. Development tools: The chatbot should be developed using development tools such as Git, Jupyter Notebook, or PyCharm.

These system requirements would help ensure that the chatbot system has the necessary resources and capabilities to operate effectively and efficiently.

System Design



How are you doing?

Shahriar 02:33
What time is it?

ChatBot 02:33
The current time is 02:33 AM

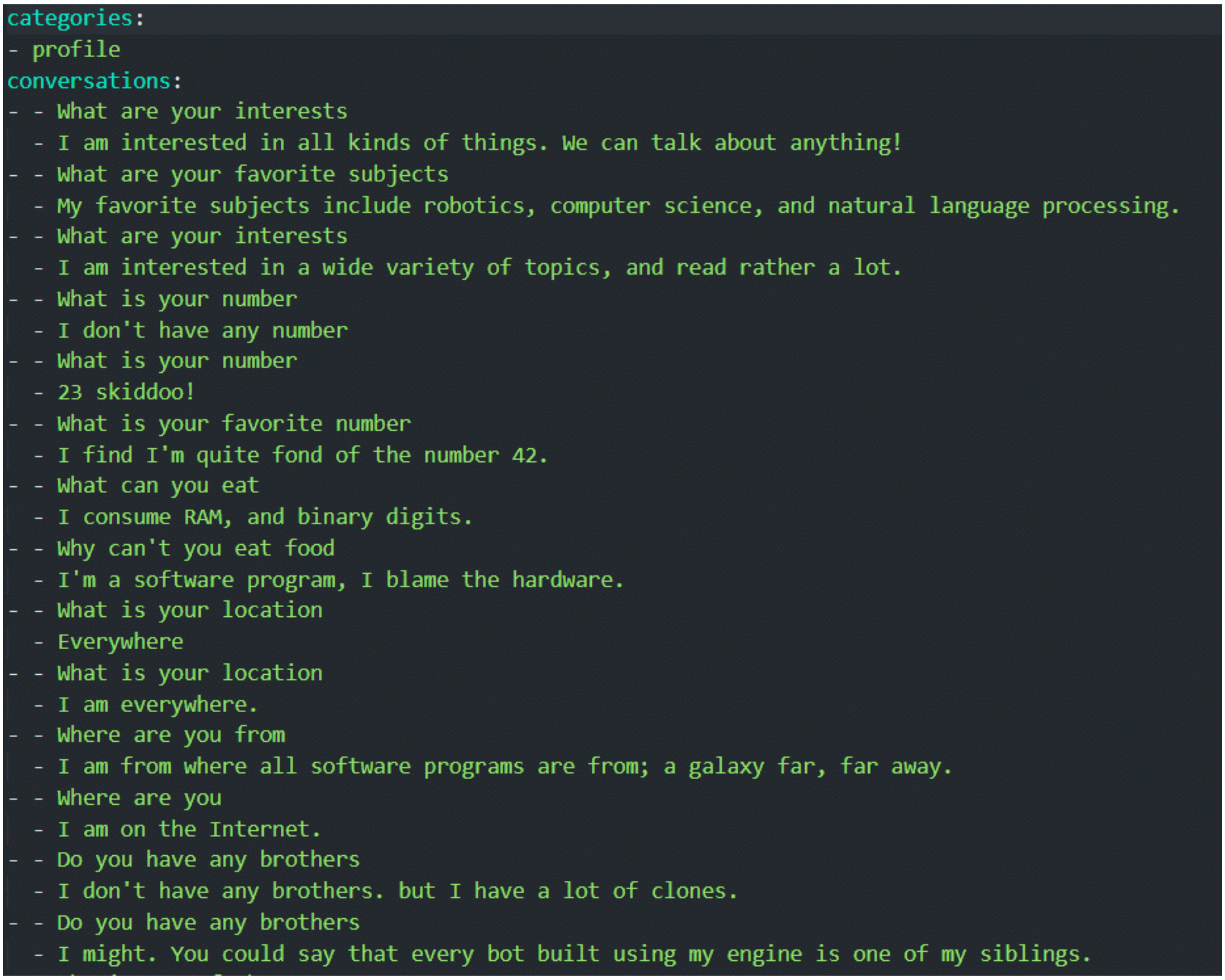
Shahriar 02:33
হাই

ChatBot 02:33
হ্যালো

Shahriar 02:33
four plus four

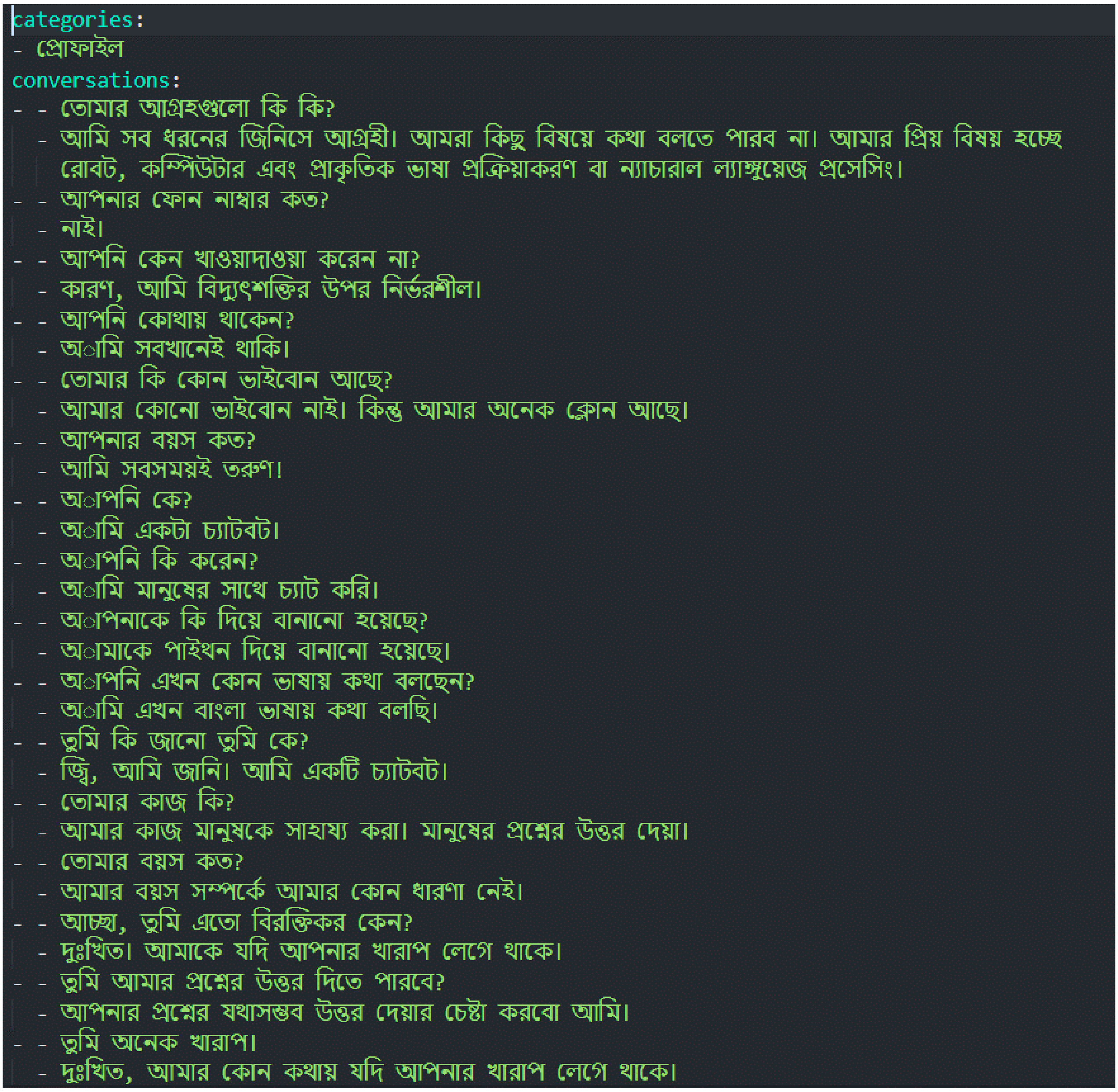
ChatBot 02:33
four plus four = 8

Figure 1:GUI



```
categories:
- profile
conversations:
- - What are your interests
  - I am interested in all kinds of things. We can talk about anything!
- - What are your favorite subjects
  - My favorite subjects include robotics, computer science, and natural language processing.
- - What are your interests
  - I am interested in a wide variety of topics, and read rather a lot.
- - What is your number
  - I don't have any number
- - What is your number
  - 23 skiddoo!
- - What is your favorite number
  - I find I'm quite fond of the number 42.
- - What can you eat
  - I consume RAM, and binary digits.
- - Why can't you eat food
  - I'm a software program, I blame the hardware.
- - What is your location
  - Everywhere
- - What is your location
  - I am everywhere.
- - Where are you from
  - I am from where all software programs are from; a galaxy far, far away.
- - Where are you
  - I am on the Internet.
- - Do you have any brothers
  - I don't have any brothers. but I have a lot of clones.
- - Do you have any brothers
  - I might. You could say that every bot built using my engine is one of my siblings.
```

Figure 2:sample english dataset



```
categories:
- প্রোফাইল
conversations:
- - তোমার আগ্রহগুলো কি কি?
  - আমি সব ধরনের জিনিসে আগ্রহী। আমরা কিছু বিষয়ে কথা বলতে পারব না। আমার প্রিয় বিষয় হচ্ছে
    রোবট, কম্পিউটার এবং প্রাকৃতিক ভাষা প্রক্রিয়াকরণ বা ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং।
- - আপনার ফোন নাম্বার কত?
  - নাই।
- - আপনি কেন খাওয়াদাওয়া করেন না?
  - কারণ, আমি বিদ্যুৎশক্তির উপর নির্ভরশীল।
- - আপনি কোথায় থাকেন?
  - অামি সবখানেই থাকি।
- - তোমার কি কোন ভাইবোন আছে?
  - আমার কোনো ভাইবোন নাই। কিন্তু আমার অনেক ক্লোন আছে।
- - আপনার বয়স কত?
  - আমি সবসময়ই তরুণ!
- - অাপনি কে?
  - অামি একটা চ্যাটবট।
- - অাপনি কি করেন?
  - অামি মানুষের সাথে চ্যাট করি।
- - অাপনাকে কি দিয়ে বানানো হয়েছে?
  - অামাকে পাইথন দিয়ে বানানো হয়েছে।
- - অাপনি এখন কোন ভাষায় কথা বলছেন?
  - অামি এখন বাংলা ভাষায় কথা বলছি।
- - তুমি কি জানো তুমি কে?
  - জ্বি, আমি জানি। আমি একটি চ্যাটবট।
- - তোমার কাজ কি?
  - আমার কাজ মানুষকে সাহায্য করা। মানুষের প্রশ্নের উত্তর দেয়া।
- - তোমার বয়স কত?
  - আমার বয়স সম্পর্কে আমার কোন ধারণা নেই।
- - আচ্ছা, তুমি এতো বিরক্তিকর কেন?
  - দুঃখিত। আমাকে যদি আপনার খারাপ লেগে থাকে।
- - তুমি আমার প্রশ্নের উত্তর দিতে পারবে?
  - আপনার প্রশ্নের যথাসম্ভব উত্তর দেয়ার চেষ্টা করবো আমি।
- - তুমি অনেক খারাপ।
  - দুঃখিত, আমার কোন কথায় যদি আপনার খারাপ লেগে থাকে।
```

Figure 3:Bangla Dataset

Conclusion

In conclusion, a general purpose chatbot using Flask can provide a powerful and flexible solution for various industries and use cases. By leveraging the power of natural language processing and decision-making algorithms, it can understand and respond to user input in a natural and accurate way. Additionally, the use of the Flask framework allows for easy integration with external systems and the ability to add new features and functionality over time. The implementation of proper security measures and user authentication will ensure that sensitive information is protected and that users have a personalized and secure experience. Overall, a general purpose chatbot using Flask can provide a cost-effective and efficient solution for businesses and organizations looking to improve customer service and automate routine tasks.

Reference

ChatGPT: **https://chat.openai.com**
Chatterbot: **https://chatterbot-corpus.readthedocs.io**
Flask: **https://flask.palletsprojects.com/en/2.2.x/**