

A series of thin, black, overlapping lines forming various geometric shapes like triangles and polygons, creating a complex, abstract pattern in the upper left portion of the slide.

# CUSTOMER SUPPORT SYSTEM

Prachi Sethi

19963

CS589

# INDEX

INTRODUCTION

APP ARCHITECTURE

DESIGN - CLI-BASED APPLICATION

DESIGN - PYTHON WEB SERVER

IMPLEMENTATION - CLI-BASED APPLICATION

IMPLEMENTATION - PYTHON WEB SERVER

FUTURE ENHANCEMENTS

CONCLUSION

REFERENCES

# INTRODUCTION

- **Objective:** The Flask app aims to provide an interactive platform for users to ask questions and receive intelligent answers generated by OpenAI's GPT-3.5 model.
- **Use Case:** It serves as a powerful tool for customer support system, currently implemented to have a cli and python-webserver based solution for a Q&A based application.

# APP ARCHITECTURE

## *Overview of the App's Architecture*

- **Flask Server:**
  - The backbone of the application, responsible for handling HTTP requests and responses.
  - Utilizes the Flask web framework to create a robust and scalable server.
- **OpenAI GPT-3.5 Integration:**
  - Incorporates the OpenAI GPT-3.5 model for natural language processing tasks.
  - The model is accessed through the OpenAI API to generate intelligent responses based on user queries.
- **CSV Data Processing:**
  - The app reads and processes data from a CSV file containing pre-processed embeddings and relevant information.
  - CSV data serves as a source of context for generating responses, enhancing the app's understanding of user queries.

# DESIGN - CLI-BASED APPLICATION

Our Command Line Interface (CLI) Question Answering Tool offers a streamlined text-based interaction, providing users with quick and precise information retrieval. Built with Python, this CLI application leverages simplicity and command-driven efficiency, making it an ideal choice for users who prefer a terminal environment.

# DESIGN - PYTHON WEB SERVER

Our Flask-based Question Answering App, powered by OpenAI's GPT-3.5, seamlessly combines Python's efficiency with Flask's simplicity. Users can effortlessly ask questions, receiving intelligent and context-aware responses. The app's intuitive interface, real-time interaction, and integration with GPT-3.5 showcase a powerful tool for information retrieval in a user-friendly environment.

# IMPLEMENTATION - CLI-BASED APP

```
(base) rumnraisin@Prachis-Air web-crawl-q-and-a-example % python3 -m venv webcrawler
wler
(base) rumnraisin@Prachis-Air web-crawl-q-and-a-example % source webcrawler/bin/activate
(webcrawler) (base) rumnraisin@Prachis-Air web-crawl-q-and-a-example % pip3 install -r requirements.txt
Collecting aiohttp==3.8.5 (from -r requirements.txt (line 1))
  Downloading aiohttp-3.8.5.tar.gz (7.4 MB)
    7.4/7.4 MB 2.2 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Installing backend dependencies ... done
Preparing metadata (pyproject.toml) ... done
Collecting aiosignal==1.3.1 (from -r requirements.txt (line 2))
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting appnope==0.1.3 (from -r requirements.txt (line 3))
```

```
import tiktoken

# Load the cl100k_base tokenizer which is designed to work with the ada-002 model
tokenizer = tiktoken.get_encoding("cl100k_base")

df = pd.read_csv('processed/scraped.csv', index_col=0)
df.columns = ['title', 'text']

# Tokenize the text and save the number of tokens to a new column
df['n_tokens'] = df.text.apply(lambda x: len(tokenizer.encode(x)))

# Visualize the distribution of the number of tokens per row using a histogram
```

PROBLEMS (18) OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

```
Downloading bs4-0.0.1.tar.gz (1.1 kB)
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting certifi==2023.7.22 (from -r requirements.txt (line 11))
Obtaining dependency information for certifi==2023.7.22 from https://files.pythonhosted.org/packages/4c/dd/2234eab22353ffc7d94e8d13177aa805013286e93e7b40eae01fb7c3d9/certifi-2023.7.22-py3-none-any.whl.metadata
Downloading certifi-2023.7.22-py3-none-any.whl.metadata (2.2 kB)
Collecting charset-normalizer==2.1.1 (from -r requirements.txt (line 12))
Downloading charset-normalizer-2.1.1-py3-none-any.whl (39 kB)
Collecting comm==0.1.2 (from -r requirements.txt (line 13))
Downloading comm-0.1.2-py3-none-any.whl (6.5 kB)
Collecting contourpy==1.0.7 (from -r requirements.txt (line 14))
Downloading contourpy-1.0.7.tar.gz (13.4 MB)
    7.9/13.4 MB 451.5 kB/s eta 0:00:13
```

```
(base) rumnraisin@Prachis-Air python-solution % source venv/bin/activate
(venv) (base) rumnraisin@Prachis-Air python-solution % ./run
Requirement already satisfied: autopep8==1.6.0 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 1)) (1.6.0)
Requirement already satisfied: click==8.1.3 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 2)) (8.1.3)
Requirement already satisfied: et-xmlfile==1.1.0 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 3)) (1.1.0)
Requirement already satisfied: Flask==3.0.1 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 4)) (3.0.1)
Requirement already satisfied: itsdangerous==2.1.2 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 5)) (2.1.2)
Requirement already satisfied: Jinja2==3.1.2 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 6)) (3.1.2)
Requirement already satisfied: MarkupSafe==2.1.1 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 7)) (2.1.1)
Requirement already satisfied: openpyxl==3.0.9 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 8)) (3.0.9)
```

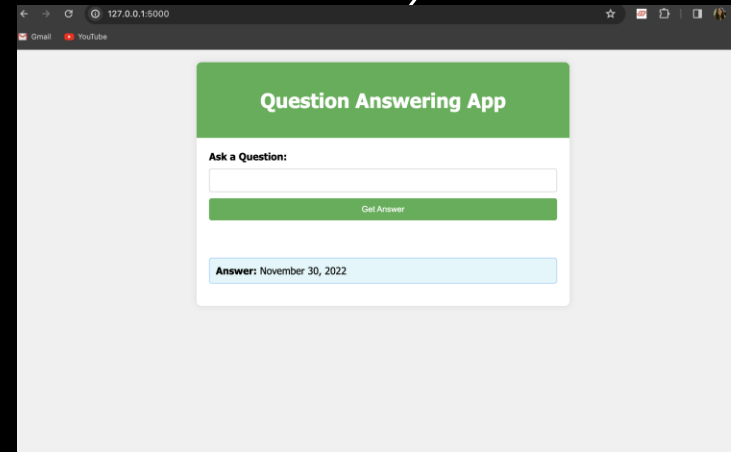
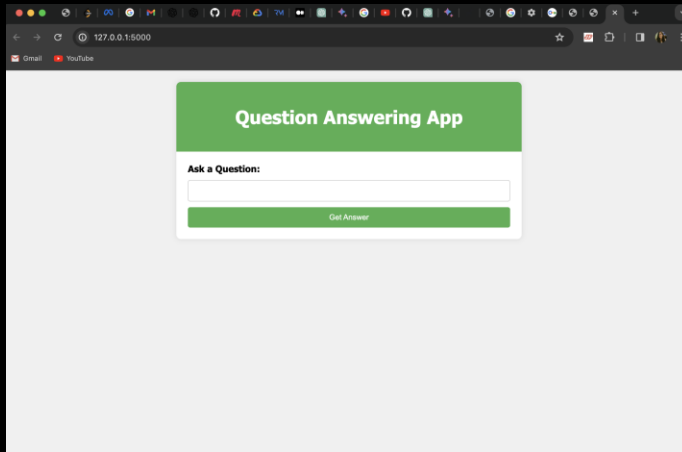
```
(venv) (base) rumnraisin@Prachis-Air cli-solution % python3 web-crawler.py
/Users/rumnraisin/Documents/sfbu/genAI/web-crawler/cli-solution/web-crawler.py:14: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466
```

```
import pandas as pd
https://openai.com/
https://openai.com/research/overview
https://openai.com/research/language-models-are-few-shot-learners
https://openai.com/research?authors=jeffrey-wu
https://openai.com/research?topics=transformers
```

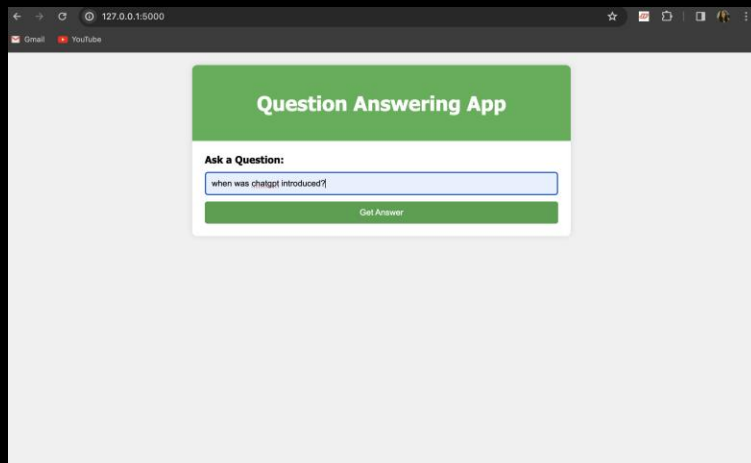
```
(venv) (base) rumnraisin@Prachis-Air cli-solution % python3 web-crawler.py
/Users/rumnraisin/Documents/sfbu/genAI/web-crawler/cli-solution/web-crawler.py:12: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466
```

```
import pandas as pd
I don't know.
We are introducing two new embedding models: a smaller and highly efficient text-embedding-3-small model, and a larger and more powerful text-embedding-3-large model.
```

# IMPLEMENTATION - PYTHON WEBSERVER



```
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 827-059-473
```



```
(base) rumnraisin@Prachis-Air python-solution % source venv/bin/activate
(venv) (base) rumnraisin@Prachis-Air python-solution % ./run
Requirement already satisfied: autopep8==1.6.0 in /Users/rumnraisin/Documents/sfbu/genAI/openai-q
uickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 1)) (1.6.0)
Requirement already satisfied: click==8.1.3 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quic
kstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 2)) (8.1.3)
Requirement already satisfied: et-xmlfile==1.1.0 in /Users/rumnraisin/Documents/sfbu/genAI/openai-
quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 3)) (1.1.0)
Requirement already satisfied: Flask==3.0.1 in /Users/rumnraisin/Documents/sfbu/genAI/openai-quic
kstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 4)) (3.0.1)
Requirement already satisfied: itsdangerous==2.1.2 in /Users/rumnraisin/Documents/sfbu/genAI/open
ai-quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 5)) (2.1.2)
Requirement already satisfied: Jinja2==3.1.2 in /Users/rumnraisin/Documents/sfbu/genAI/openai-qui
ckstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 6)) (3.1.2)
Requirement already satisfied: MarkupSafe==2.1.1 in /Users/rumnraisin/Documents/sfbu/genAI/openai-
quickstart-python/venv/lib/python3.11/site-packages (from -r requirements.txt (line 7)) (2.1.1)
Requirement already satisfied: opencv-python==4.8.0 in /Users/rumnraisin/Documents/sfbu/genAI/openai-q
```



# FUTURE ENHANCEMENTS

---

- **Multimodal Capabilities:** Explore incorporating multimodal features, enabling the app to process not only text but also images and possibly audio for more comprehensive responses.
- **User Personalization:** Implement user profiles and preferences to tailor responses based on individual user interactions, creating a more personalized experience.
- **Real-Time Collaboration:** Introduce collaborative features, allowing users to share and collaborate on contextual information in real-time, fostering collective intelligence.
- **Enhanced UI/UX:** Invest in a more visually appealing and intuitive user interface, making the app more engaging and accessible to a broader audience.
- **Continuous Model Training:** Implement a mechanism for continuous model training, ensuring the app adapts to evolving language patterns and stays up-to-date with the latest information.

# FUTURE ENHANCEMENTS

---

- **Support for Multiple Languages:** Extend language support beyond English to cater to a global audience, making the app more inclusive and versatile.
- **Integration with External APIs:** Explore integration with external APIs to fetch real-time data and further enrich the context for generating responses.
- **Feedback Mechanism:** Implement a user feedback mechanism to gather insights on the accuracy and usefulness of responses, facilitating continuous improvement.
- **Mobile Application Development:** Extend the app's reach by developing a mobile application, providing users with on-the-go access to intelligent question-answering capabilities.
- **Security and Privacy Measures:** Implement robust security measures and privacy controls, ensuring user data is handled securely and in compliance with relevant regulations.



## CONCLUSION

In summary, our Flask-based Question Answering App, powered by OpenAI's GPT-3.5, combines the efficiency of Python and Flask for web development with the intelligence of GPT-3.5 for context-aware responses. This project demonstrates the seamless integration of technology stacks, emphasizing simplicity and effectiveness in creating a user-friendly and intelligent application. As we look ahead, we're committed to refining and expanding the app's capabilities, ensuring it remains at the forefront of accessible and responsive information retrieval tools. Thank you for your interest and engagement.

# REFERENCES

[https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine\\_learning/chatgpt/slide/exercise\\_chatgpt.html](https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/chatgpt/slide/exercise_chatgpt.html)Links to an external site.

<https://platform.openai.com/docs/tutorials/web-qa-embeddings>

<https://openai.com/>

<https://flask.palletsprojects.com/en/3.0.x/>

A series of white, overlapping geometric lines and polygons on a black background, located on the left side of the slide.

# THANK YOU

Prachi Sethi

19963