

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE



1. Set Up the Development Environment

For web deployment, NLTK, SpaCy, Flask, or Streamlit are essential libraries, while Streamlit is suitable for a simple UI.

```
pip install nltk spacy flask streamlit  
python -m spacy download en_core_web_sm
```

2. Prepare the FAQ Data

The task involves gathering a list of frequently asked questions and their corresponding answers.

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE

```
1 faqs = {
2     "What are your hours of operation?": "We are open from 9 AM to 5 PM, Monday to Friday.",
3     "Where are you located?": "Our office is located at 123 Main Street, Springfield.",
4     "How can I contact customer support?": "You can contact customer support via email at support@example.com or call us at 123-456-7890."
5 }
6
7
```

3. Preprocess the Data

Utilize SpaCy or NLTK to preprocess text by tokenizing, removing stop words, and lemmatizing.

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE

```
Python3 script.py
1  import spacy
2
3  nlp = spacy.load('en_core_web_sm')
4
5  Codeium: Refactor | Explain | Generate Docstring | X
6  def preprocess_text(text):
7      doc = nlp(text.lower())
8      return ' '.join([token.lemma_ for token in doc if not token.is_stop and token.is_alpha])
9
10 processed_faqs = {preprocess_text(question): answer for question, answer in faqs.items()}
11
12
```

4. Create a Function to Match User Queries

The user's question is compared with preprocessed FAQ questions using similarity measures.

Cosine similarity, word vectors, or simple keyword matching are all methods that can be utilized for data analysis.

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE

```
PythonTask.py > ...
1  from sklearn.feature_extraction.text import TfidfVectorizer
2  from sklearn.metrics.pairwise import cosine_similarity
3
4  vectorizer = TfidfVectorizer()
5
6  Codeium: Refactor | Explain | Generate Docstring | X
7  def find_best_match(user_query):
8      user_query_processed = preprocess_text(user_query)
9      faqs_vectorized = vectorizer.fit_transform(preprocessed_faqs.keys())
10     user_vectorized = vectorizer.transform([user_query_processed])
11     similarities = cosine_similarity(user_vectorized, faqs_vectorized).flatten()
12     best_match_idx = similarities.argmax()
13     return list(faqs.values())[best_match_idx]
14
15 # Example Usage
16 user_input = "What time do you open?"
17 response = find_best_match(user_input)
18 print(response)
19
```

5. Create the Chatbot Interface

Web Interface with Flask

```
PythonTask.py > ...
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  Codeium: Refactor | Explain | Generate Docstring | X
6  @app.route("/", methods=["GET", "POST"])
7  def chat():
8      if request.method == "POST":
9          user_input = request.form["user_input"]
10         response = find_best_match(user_input)
11         return render_template("chat.html", user_input=user_input, response=response)
12         return render_template("chat.html")
13
14 if __name__ == "__main__":
15     app.run(debug=True)
16
```

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE

HTML Template (chat.html):

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>FAQ Chatbot</title>
5 </head>
6 <body>
7   <h1>FAQ Chatbot</h1>
8   <form method="post">
9     <label for="user_input">Ask a question:</label><br><br>
10    <input type="text" id="user_input" name="user_input"><br><br>
11    <input type="submit" value="Ask">
12  </form>
13  {% if user_input %}
14  <h2>Your question: {{ user_input }}</h2>
15  <h2>Answer: {{ response }}</h2>
16  {% endif %}
17 </body>
18 </html>
19
20
```

Simple UI with Streamlit

```
1 import streamlit as st
2
3 st.title("FAQ Chatbot")
4
5 user_input = st.text_input("Ask a question")
6 if user_input:
7     response = find_best_match(user_input)
8     st.write("Answer:", response)
9
```

Name M Hammad Hussain

Email hammadhussain1972@gmail.com

Domain ARTIFICIAL INTELLIGENCE

6. Test the Chatbot

The chatbot should be used to interact with users by asking various questions to ensure it provides the correct responses.

The study aims to examine the handling of edge cases, such as questions not included in the FAQ list.

7. Deploy the Chatbot

The Flask app can be easily deployed on platforms like Heroku or shared with the Streamlit app for easy access.

8. Maintain and Update

The FAQ list should be regularly updated with new common questions as they arise.

Improve the NLP processing by fine-tuning or integrating more advanced models if necessary.