Experiment-12

Implement a simple chatbot for answering python questions from text file.

```
Command Prompt                    ×    +  ∨                        –  □  ×

(myenv) C:\Users\student\Desktop\Chatbot_Flask>pip list
Package         Version
-------------   -------
blinker         1.6.2
click           8.1.7
colorama        0.4.6
Flask           2.3.3
itsdangerous    2.1.2
Jinja2          3.1.2
MarkupSafe      2.1.3
pip             23.2.1
setuptools      65.5.0
Werkzeug        2.3.7

(myenv) C:\Users\student\Desktop\Chatbot_Flask>
```

```
Command Prompt                    ×    +  ∨                        –  □  ×

(myenv) C:\Users\student\Desktop\Chatbot_Flask>pip list
Package         Version
-------------   -------
blinker         1.6.2
click           8.1.7
colorama        0.4.6
Flask           2.3.3
itsdangerous    2.1.2
Jinja2          3.1.2
MarkupSafe      2.1.3
pip             23.2.1
setuptools      65.5.0
Werkzeug        2.3.7

(myenv) C:\Users\student\Desktop\Chatbot_Flask>pip freeze>requirements.txt

(myenv) C:\Users\student\Desktop\Chatbot_Flask>
```

```
Chatbot_Flask              ×    +                                 –  □  ×

⊕ New ∨    ✕   ⧉  ⬚  ⬚  ⬚  🗑   ↑↓ Sort ∨  ☰ View ∨  ...

←  →  ∨  ↑   > Chatbot_Flask                    ⟳   Search Chatbot_Flask  ⌕

⌂ Home          Name          Date modified     Type           Size
> ☁ OneDrive    ⬚ myenv        02-09-2023 09:54  File folder
                ▤ requirements 02-09-2023 10:00  Text Document   1 KB
🖥 Desktop    ⭐
⬇ Downloads  ⭐
▤ Documents  ⭐
🖼 Pictures   ⭐
🎵 Music      ⭐
```

```
Command Prompt - flask run    ×    +  ⌄                                    –  ☐  ✕

(myenv) C:\Users\student\Desktop\Chatbot_Flask>flask run
 * Debug mode: off
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [02/Sep/2023 10:09:15] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [02/Sep/2023 10:09:29] "POST /ask HTTP/1.1" 200 -
127.0.0.1 - - [02/Sep/2023 10:09:29] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [02/Sep/2023 10:09:29] "GET /favicon.ico HTTP/1.1" 404 -
```

app.py

```python
from flask import Flask, request, render_template
app = Flask(__name__)
# Load python questions and answering from the text file
def load_questions():
    questions = {}
    with open("Python_questions.text", 'r') as file:
        lines = file.readlines()
        for line in lines:
            question, answer = line.strip().split("|")
            questions[question.strip()] = answer.strip()
    return questions
Python_questions = load_questions()
# Define a route for the Chatbot Page
@app.route("/")
def chatbot_page():
    return render_template("Chatbot.html")
```

```
# Define a route for answering questions
@app.route("/ask", methods = ["POST"])
def answer_question():
    user_question = request.form.get("user_question")
    if user_question in Python_questions:
        answer = Python_questions[user_question]
    else:
        answer = "Sorry, I don't know the answer
                   to that question."
    return render_template("chatbot.html",
    user_question = user_question, answer = answer)

app.py run(debug = True)
```

Chartbot.html

```html
<!DOCTYPE html>
<html>
<head>
    <title> Python Chatbot </title>
    <link rel="Stylesheet"
    href="https://stackPath.bootstrapcdn.Com/bootstrap/4.5.2/
        css/bootstrap.min.css">
</head>
<body>
    <div class="Container mt-5">
        <h1> Python Chatbot </h1>
        <form method="POST" action="/ask">
            <div class="form-group">
                <label for="user_question">Ask a Python
                        Question:</label>
                <input type="text" class="form-Control"
                        id="user_question"
    name="user_question" required>
            </div>
            <button type="Submit" class="btn
btn-Primary">Submit</button>
        </form>
        {% if user_question %}
        <h3> Your Question:</h3>
```

```
<P>{{ user-question}}</P>
<h3> Answer:</h3>
<P>{{answer}}</P>
{% endif %}
</div>
</body>
</html>
```
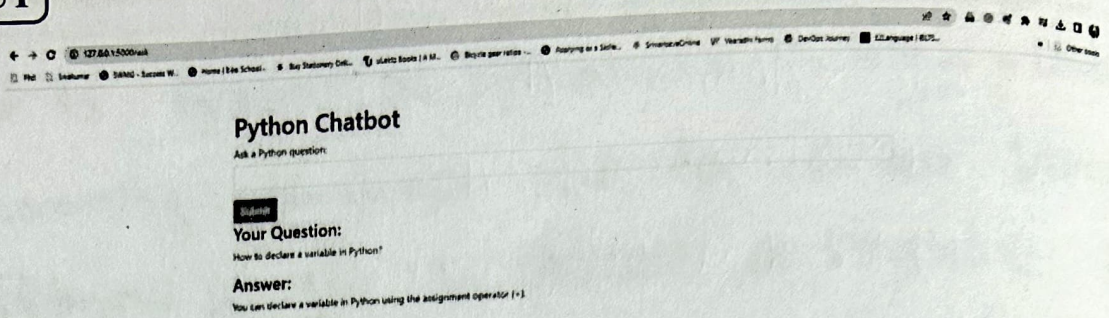
python_question.tst

What is Python? III Python is a high-level Programming Language. How to declare Variable in Python? III you can declare a Variable in Python using the assignment operator (=).

OUTPUT

Python Chatbot
Ask a Python question:

Your Question:
How to declare a variable in Python?

Answer:
You can declare a variable in Python using the assignment operator (=).

VIVA QUESTIONS

1. What is a chatbot?

Ans. A chatbot is a software application designed to simulate conversation with users, typically through text or voice, using predefined responses or artificial Intelligence.

2. Which extension is used to connect to a database in Flask?

Ans. In flask, the most commonly used extension to connect to database is flask SQL Alchemy. It Provides an easy way to work with database.

3. What is PYTHONPATH?

Ans. PYTHONPATH is an environment variable in Python that specifies list of directories where the python interpreter should look for modules and Packages.

4. What are Python Modules?

Ans. Python modules are individual files containing Python code that can be imported and reused in other Python Program. Each module in Python is essential.

5. What are the common built-in data types in Python?

Ans. The common built-in data types include
1) Numeric Type   2) Sequence Type   3) Set Type
4) Mapping Type   5) Boolean Type   6) None Type