

Project Development Phase

Delivery of Sprint – 4

Date	09 – Nov - 2022
Team ID	PNT2022TMID18803
Project Name	AI based discourse for Banking Industry

Creating Assistant & Integrate With Flask Web Page

Let us build our flask application which will be running in our local browser as an user interface.

In the flask application, users will interact with the chat bot, and based on the user queries they will get the chatbot responses.

Building Python Code

1: Importing Libraries

The first step is usually importing the libraries that will be needed in the program.

```
from flask import Flask,render_template
```

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (name).

2: Creating our flask application and loading

```
app = Flask(__name__)
```

3: Routing to the Html Page

Here, the declared constructor is used to route to the HTML page created earlier.

The '/' route is bound with the bot function. Hence, when the home page of a web server is opened in the browser, the HTML page will be rendered.

```
@app.route('/Chatbot', methods = ['GET','POST'])
def Chatbot():
    return render_template('Chatbot.html')
```

Main Function

This is used to run the application in local host.

```
if __name__ == '__main__':
    app.run()
```

Building HTML Code

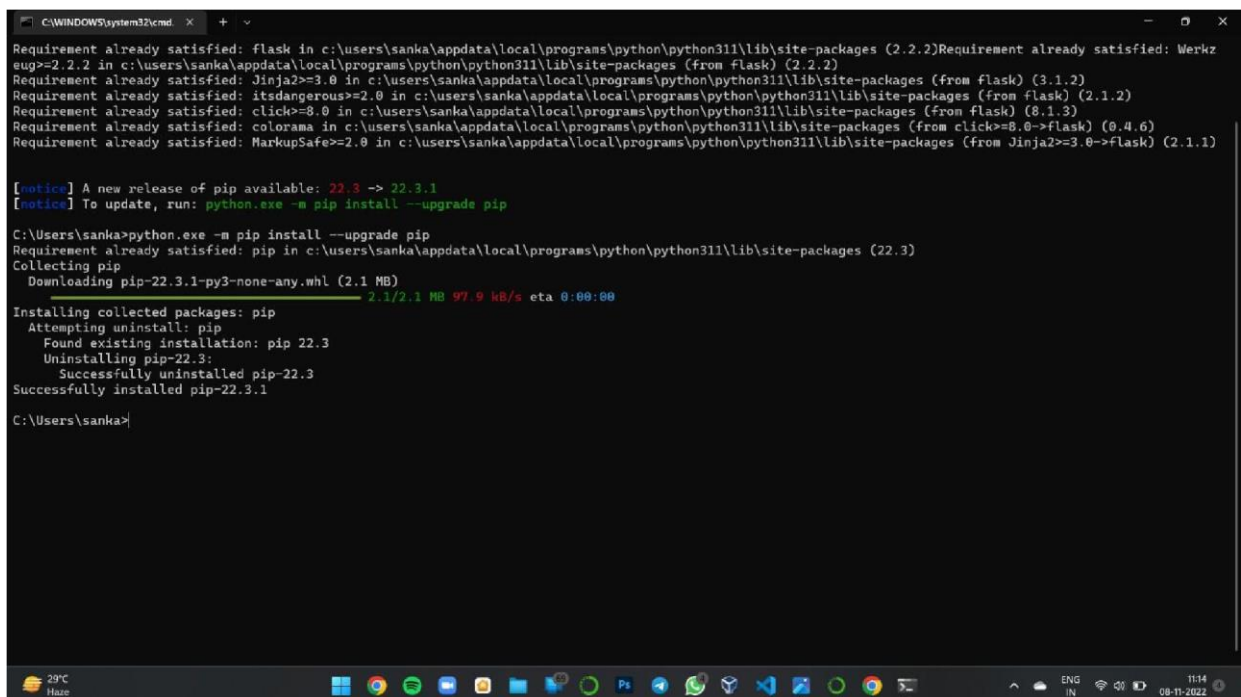
We have used HTML to create the front-end part of the web page.

Here, we have created “index.html” displays the home page which gets integrated with Watson Assistant.

Auto-generated source code which contains the Integration ID of IBM Watson Assistants is copied and pasted inside the body tag.

Run the application

1. Open Vs Code
2. Navigate to the folder where app.py resides.
3. Run the python code
4. Open a browser and type this URL <http://127.0.0.1:5500/>.
5. [Click Chatbot.html](#) and then it shows <http://127.0.0.1:5500/Chatbot.html>
6. It launches the application integrated with IBM Watson Assistant.

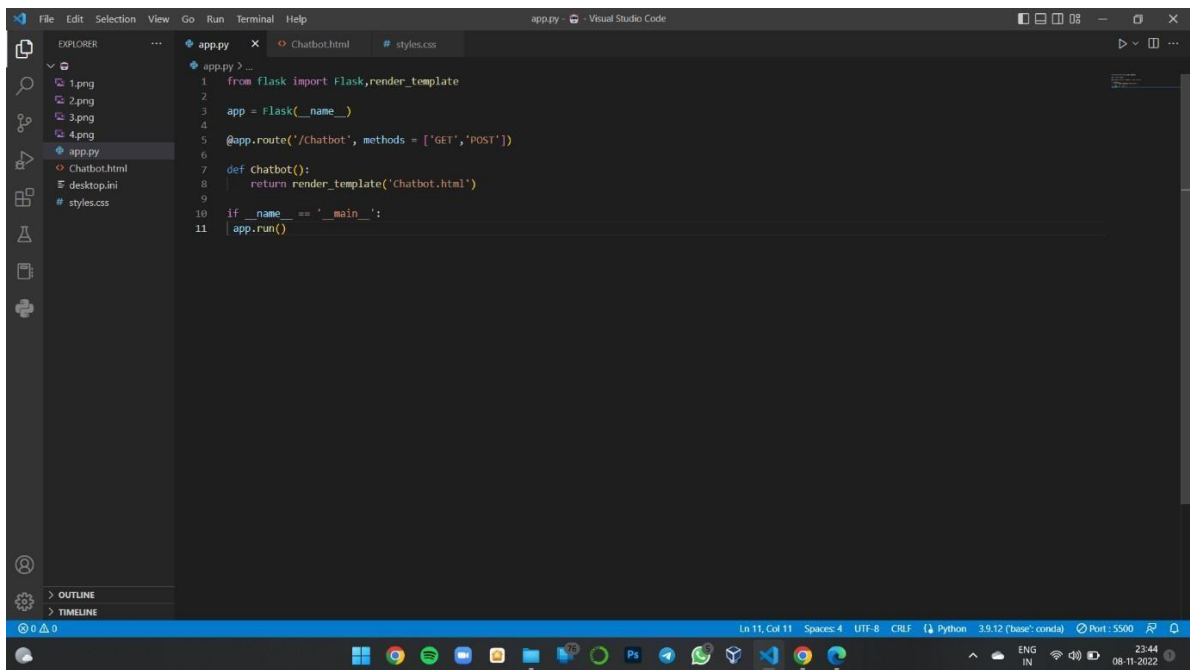


```
C:\WINDOWS\system32\cmd x + v
Requirement already satisfied: flask in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (2.2.2)
Requirement already satisfied: Werkzeug in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (2.2.2)
Requirement already satisfied: Jinja2>=3.0 in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (from flask) (8.1.3)
Requirement already satisfied: colorama in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (from click>=8.0->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (from Jinja2>=3.0->flask) (2.1.1)

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\sanka>python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\sanka\appdata\local\programs\python\python311\lib\site-packages (22.3)
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
    2.1/2.1 MB 97.9 kB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3
    Uninstalling pip-22.3:
      Successfully uninstalled pip-22.3
  Successfully installed pip-22.3.1

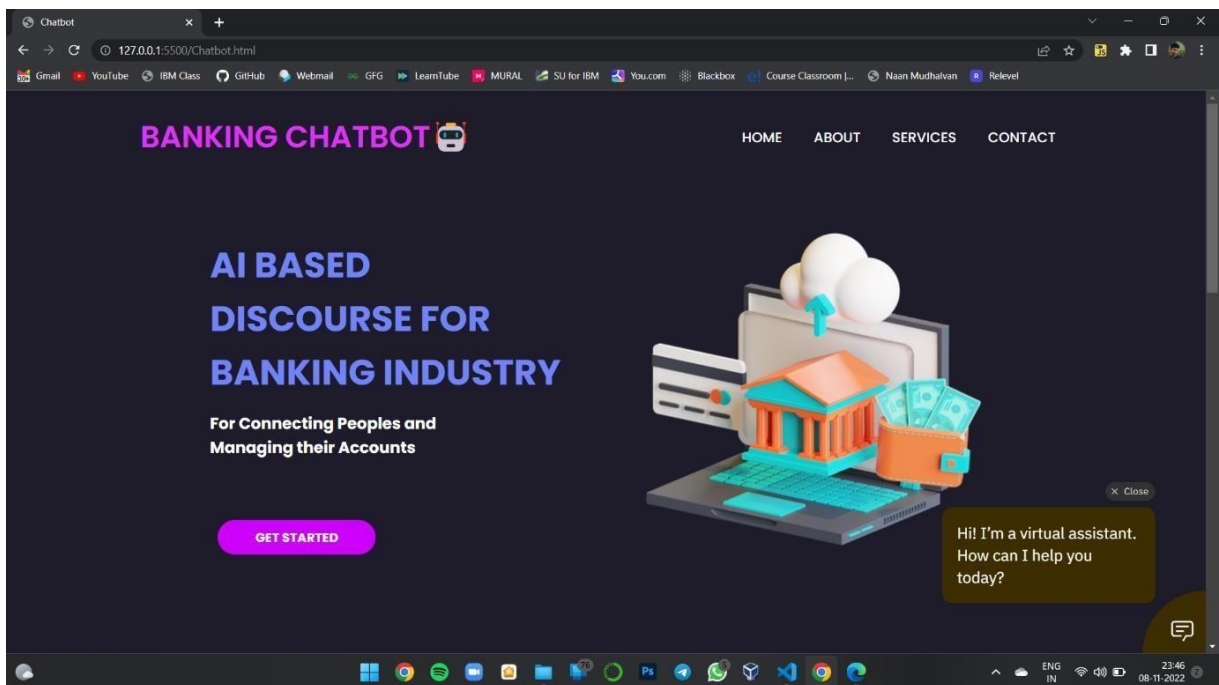
C:\Users\sanka>
```

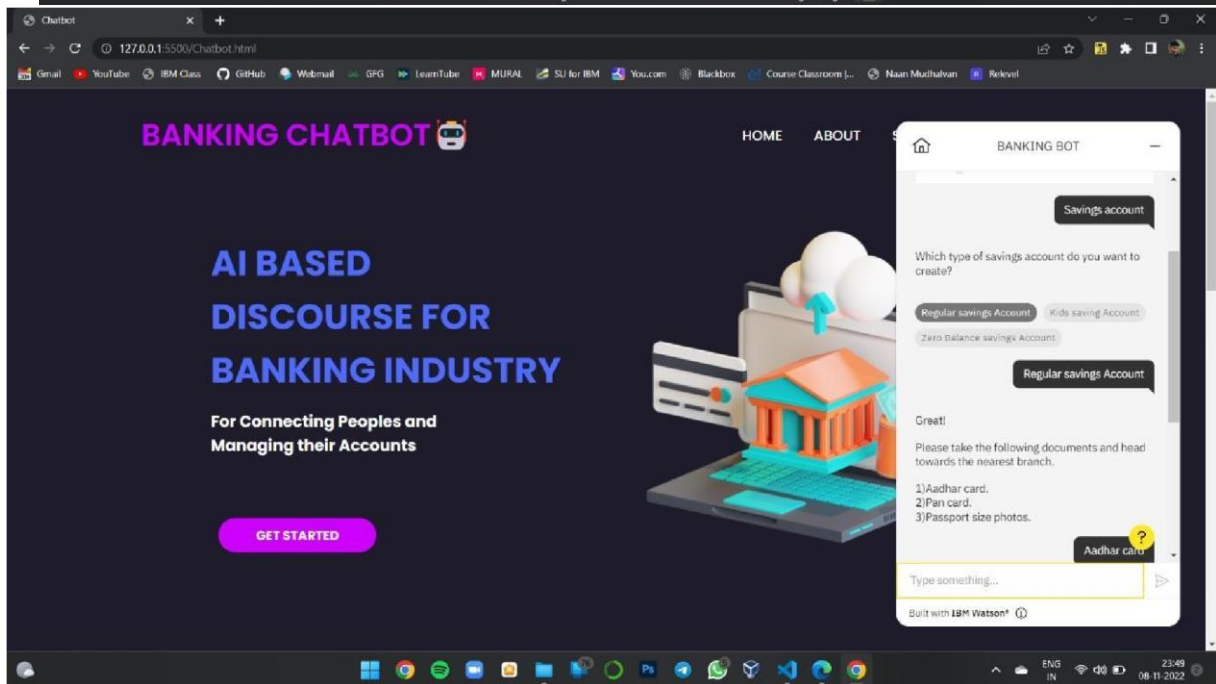
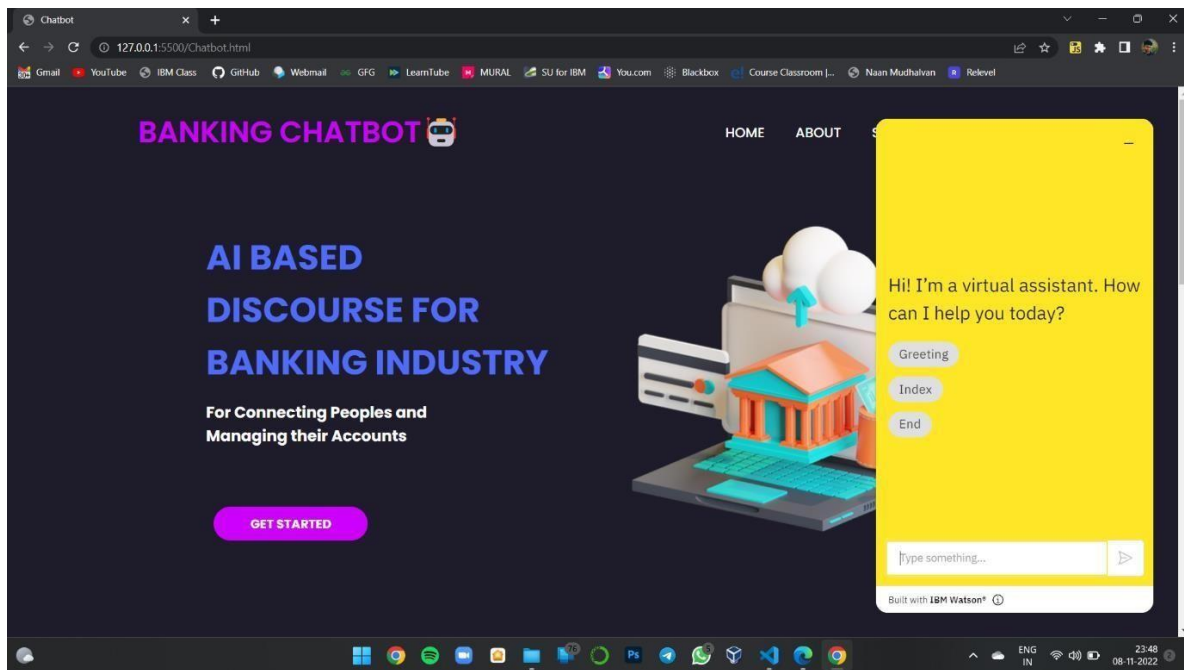


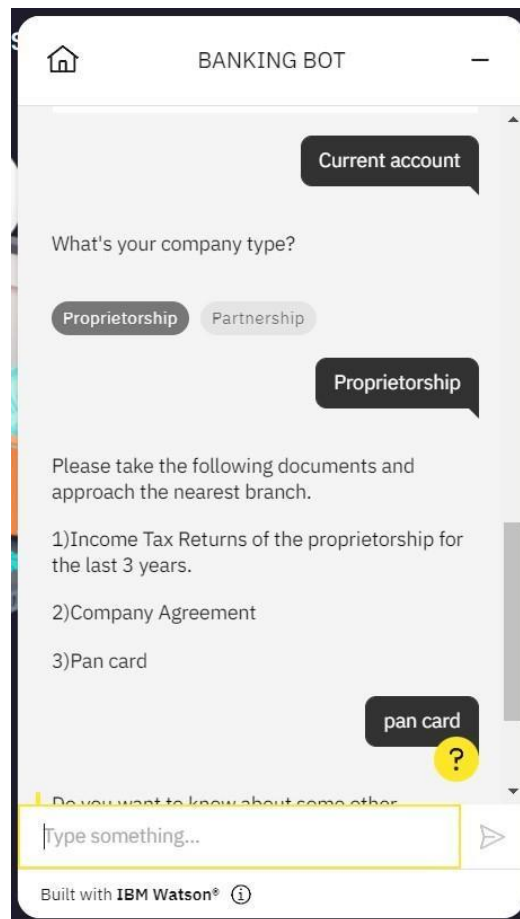
The screenshot shows the Visual Studio Code editor with a Python file named `app.py` open. The code is a simple Flask web application that serves a chatbot interface. The code is as follows:

```
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4
5 @app.route('/chatbot', methods = ['GET', 'POST'])
6
7 def chatbot():
8     return render_template('chatbot.html')
9
10 if __name__ == '__main__':
11     app.run()
```

The Explorer sidebar on the left shows the project structure with files `1.png`, `2.png`, `3.png`, `4.png`, `app.py`, `chatbot.html`, `desktop.ini`, and `styles.css`. The status bar at the bottom indicates the file is at line 11, column 11, using UTF-8 encoding and CRLF line endings. The Python interpreter is set to 3.9.12 (base: conda) and the port is 5500.







Preview of Chatbot

https://webchat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImage_URL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-3533978938d5-451f-8f05-3ddb5dbbe57e%3A%3A49d7f8d1-0500-4610-9518-068ec885f252&integrationID=fc789460-2e93-472d-b324-c85bef5219f5®ion=us-south&serviceInstanceID=35339789-38d5-451f-8f05-3ddb5dbbe57e

Source code is attached in Final Deliverables.

Note: No code for this project. So, I attached the screenshot and step to build it.

