

Project Development Phase

Delivery of Sprint - 4

Date	09 Nov 2022
Team ID	PNT2022TMID45515
Project Name	AI-based discourse for Banking Industry

Creating Assistant & Integrate With Flask Web Page

You will be creating a banking bot in this activity that has the following capabilities

1. The Bot should be able to guide a customer to create a bank account.
2. The Bot should be able to answer loan queries.
3. The Bot should be able to answer general banking queries.
4. The Bot should be able to answer queries regarding net banking.
5. With the help of this bot, you can get all the required details related to banking.

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

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

Build 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 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('/')
def bot():
    return render_template('chatbot.html')
```

Main Function This is used to run the application in localhost.

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

Build HTML Code

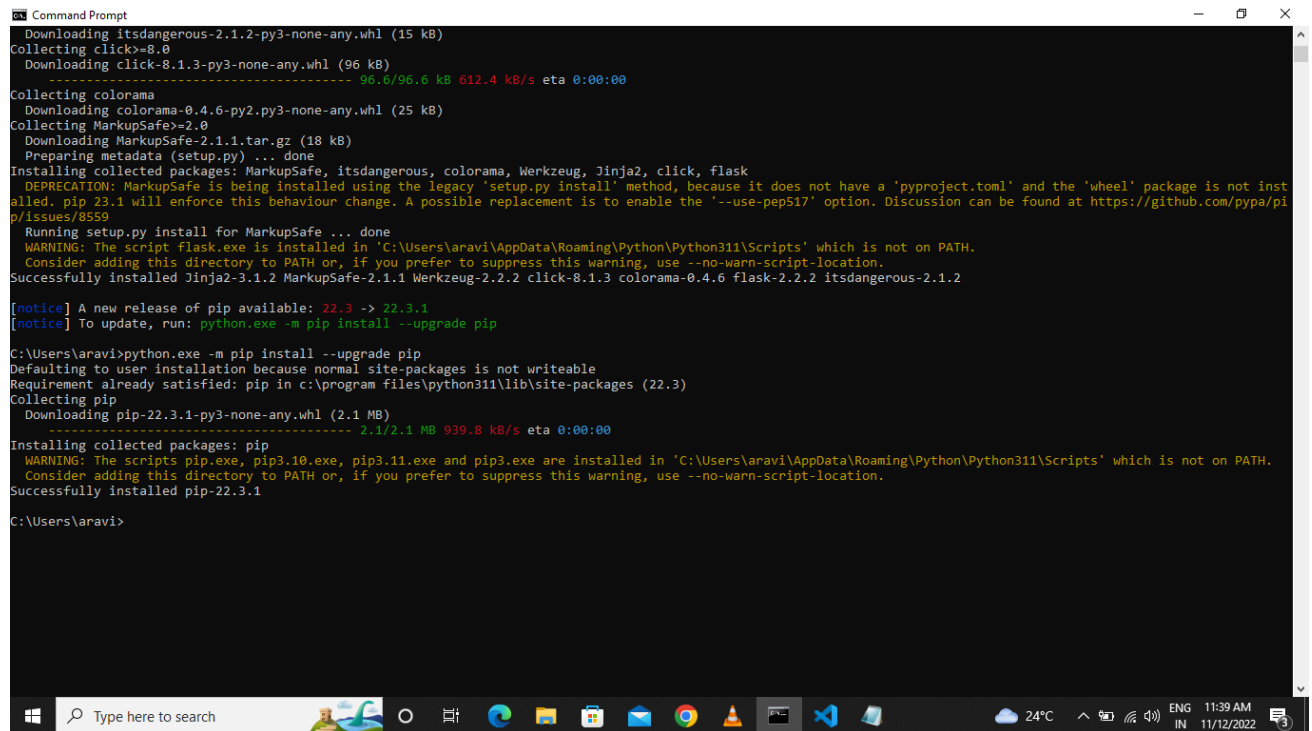
- ┆ We use HTML to create the front-end part of the web page.
- ┆ Here, we have created 1 HTML page-Chatbot.html
- Chatbot.html displays the home page which integrates with Watson Assistant.
- ┆ A simple HTML page is created. Auto-generated source code from IBM Watson Assistants is copied and pasted inside the body tag

Run The Application

Run the application

- Open the anaconda prompt from the start menu.
- Navigate to the folder where your app.py resides.
- Now type the “python app.py” command.
- It will show the local host where your app is running on <http://127.0.0.1:5000>
- Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.

Then it will run on localhost:5000

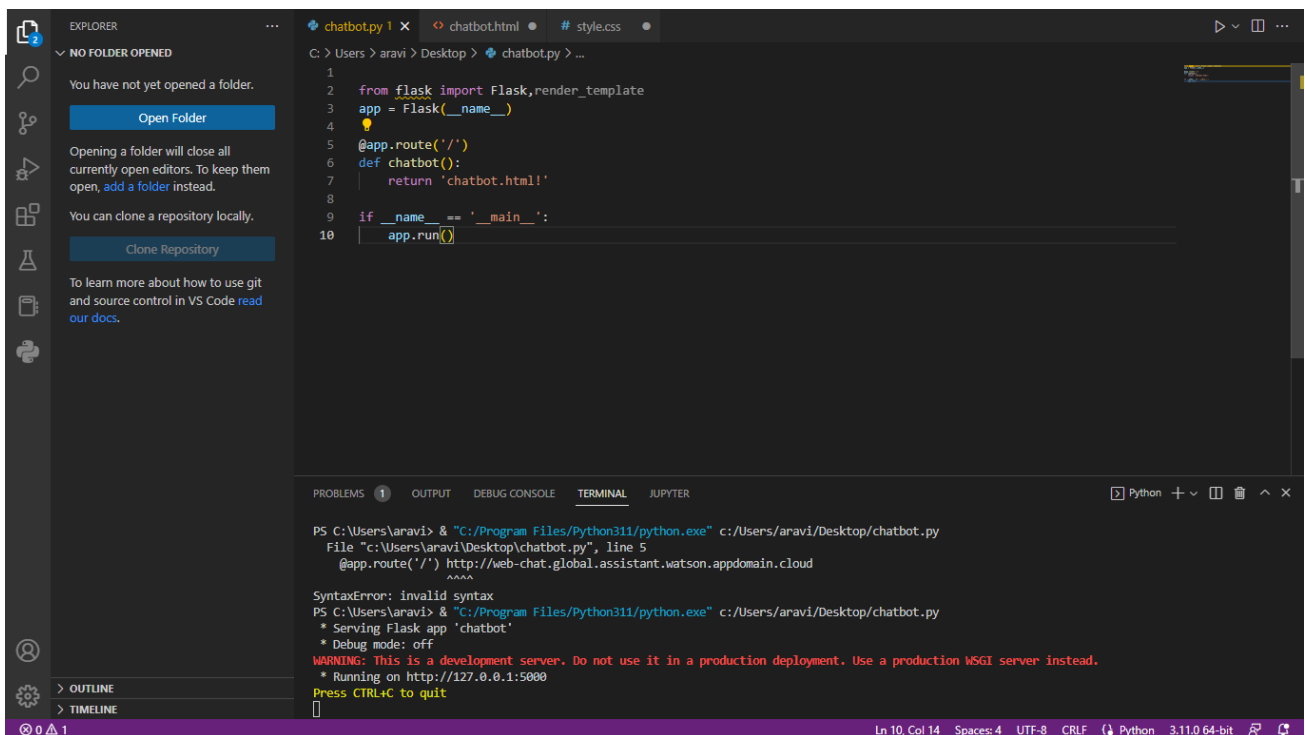


```
Command Prompt
Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.0
Downloading click-8.1.3-py3-none-any.whl (96 kB)
----- 96.6/96.6 kB 612.4 kB/s eta 0:00:00
Collecting colorama
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
Downloading MarkupSafe-2.1.1.tar.gz (18 kB)
Preparing metadata (setup.py) ... done
Installing collected packages: MarkupSafe, itsdangerous, colorama, Werkzeug, Jinja2, click, flask
DEPRECATION: MarkupSafe is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 22.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
Running setup.py install for MarkupSafe ... done
WARNING: The script flask.exe is installed in 'C:\Users\aravi\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 colorama-0.4.6 flask-2.2.2 itsdangerous-2.1.2

[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\aravi>python.exe -m pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in c:\program files\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 939.8 kB/s eta 0:00:00
Installing collected packages: pip
WARNING: The scripts pip.exe, pip3.10.exe, pip3.11.exe and pip3.exe are installed in 'C:\Users\aravi\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-22.3.1

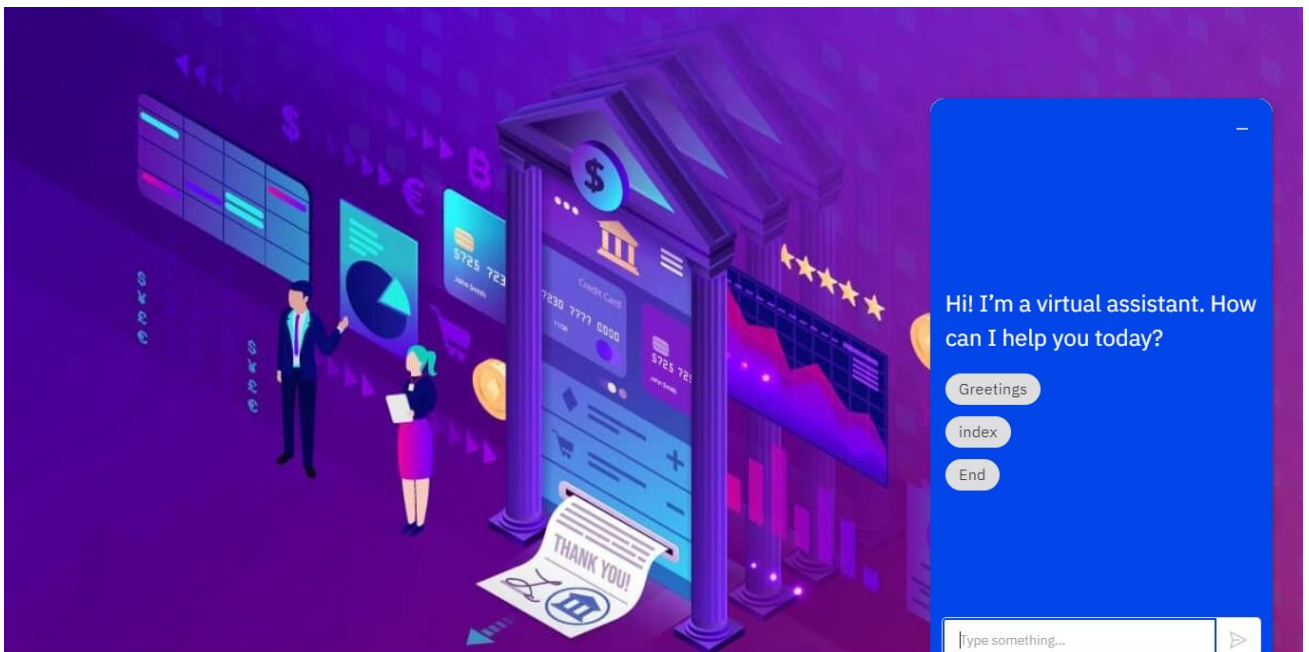
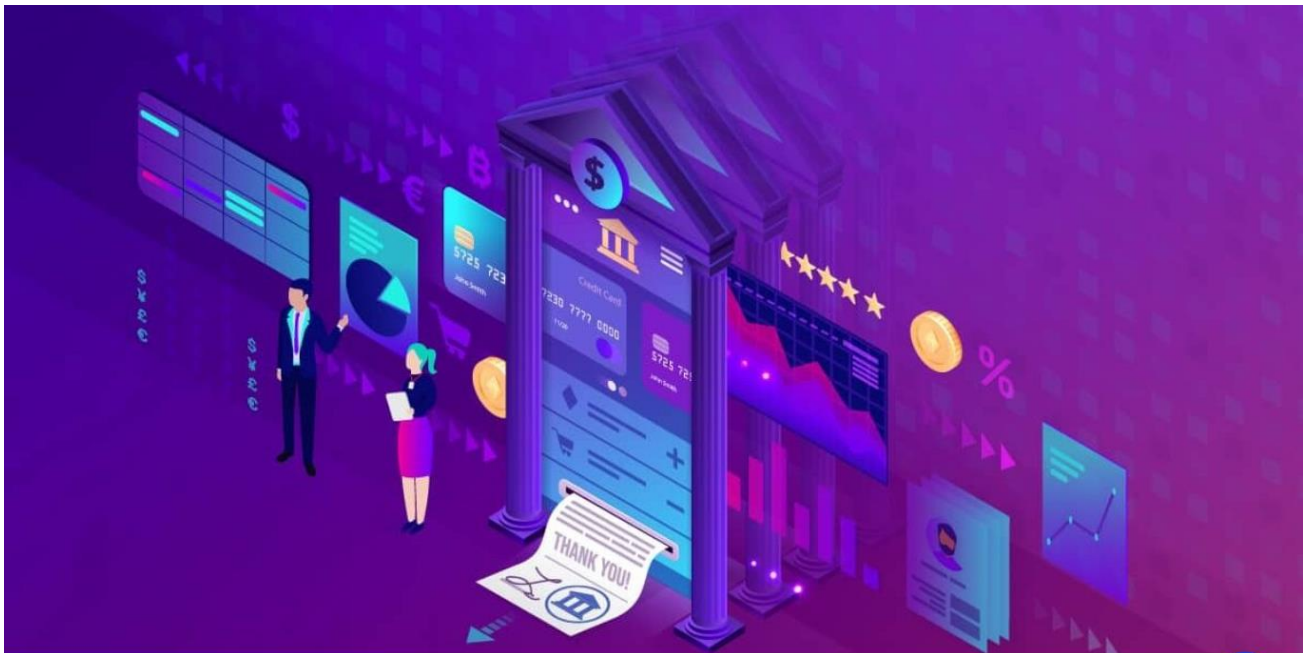
C:\Users\aravi>
```



```
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.

chatbot.py | chatbot.html | style.css
C:\Users\aravi\Desktop> chatbot.py > ...
1
2 from flask import Flask, render_template
3 app = Flask(__name__)
4
5 @app.route('/')
6 def chatbot():
7     return 'chatbot.html!'
8
9 if __name__ == '__main__':
10     app.run()

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS C:\Users\aravi> & "C:/Program Files/Python311/python.exe" c:/Users/aravi/Desktop/chatbot.py
File "c:/Users/aravi/Desktop/chatbot.py", line 5
    @app.route('/') http://web-chat.global.assistant.watson.appdomain.cloud
                    ^^^^^
SyntaxError: invalid syntax
PS C:\Users\aravi> & "C:/Program Files/Python311/python.exe" c:/Users/aravi/Desktop/chatbot.py
* Serving Flask app 'chatbot'
* Debug mode: off
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
```



Hi! I'm a virtual assistant. How can I help you today?

Greetings

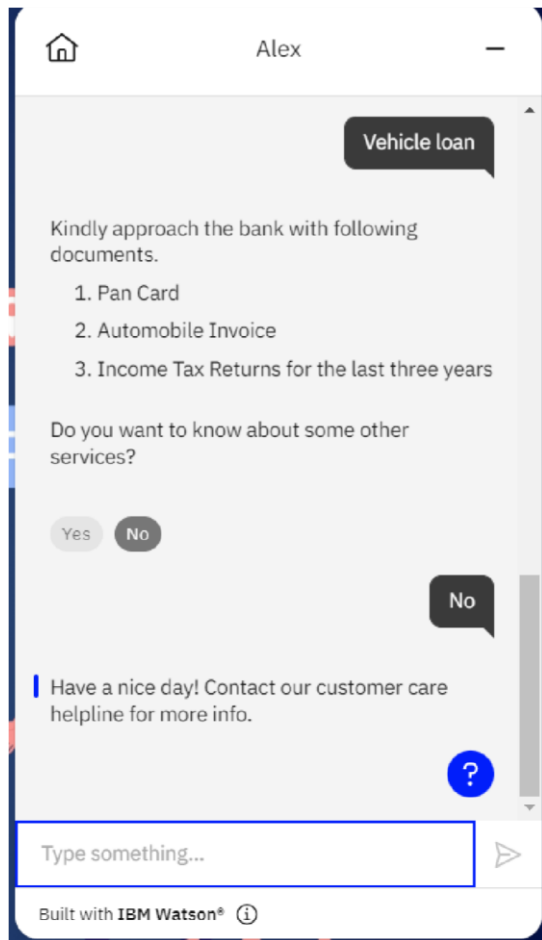
index

End

Type something...



Built with IBM Watson® ⓘ



PREVIEW OF CHATBOT:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-e5a1d21e-2939-4d14-9a23-401a94fea6ac%3A%3A6de5293c-6d8e-4556-b898-07f808debfc4&integrationID=6827dfd4-2308-4e42-9b8d-e73b8c965670®ion=us-south&serviceInstanceID=e5a1d21e-2939-4d14-9a23-401a94fea6ac>

Source code is attached in Final Deliverables.

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