ProjectDevelopmentPha
se
DeliveryofSprint- 4

Date	04November2022
TeamID	PNT2022TMID38920
ProjectName	AI-based discourse for Banking Industry

Creating Assistant & Integrate With Flask WebPage

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

- The Bot should be able to guide a customer to create a bank account.
- The Bot should be able to answer loan queries.
- The Bot should be able to answer general banking queries.
- The Bot should be able to answer queries regarding netbanking.
- 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 auser interface.

In the flask application ,users will interact with the chatbot, and based on the userqueries they willgettheoutcomes.

Build PythonCode

1:ImportingLibraries

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

```
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 thecurrent module(name).

2 . Creating our flask application and loading

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

- **Routing to the HtmlPage**

Here,the declared construct or isused to route to the HTMLpagecreatedearlier.

The ‘’ route is bound with the bot function. Hence, when the home page ofa webserver 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 HTMLCode

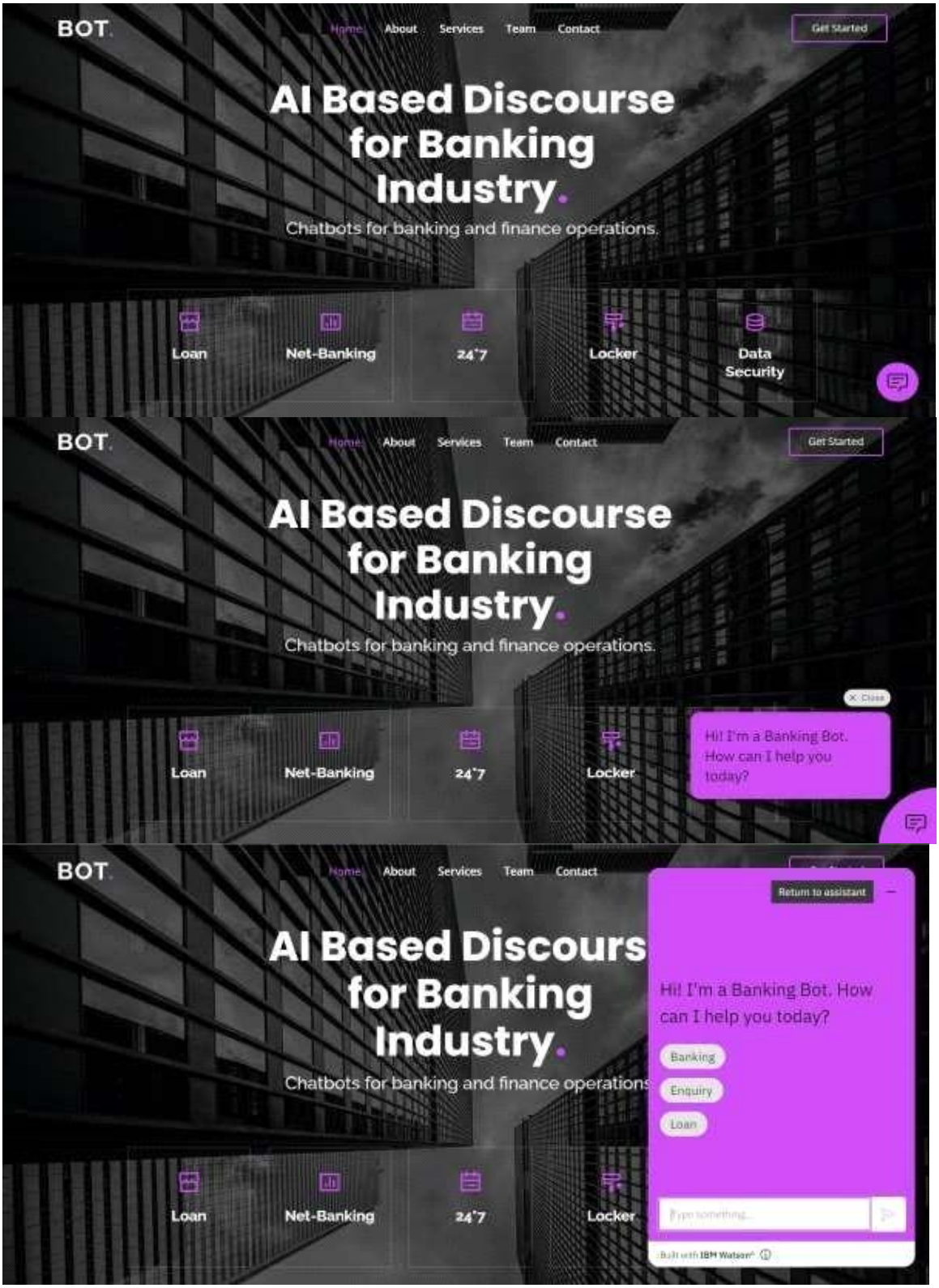
- We use HTML to create the front-endpart of the webpage.
- Here, we have created 1 HTMLpage-Chatbot.html
- Chatbot.html displays the home pagewhich integrates with WatsonAssistant.
- A simple HTML page is created.Auto-generated source code fromIBMWatson Assistants is copied and pasted inside the body tag

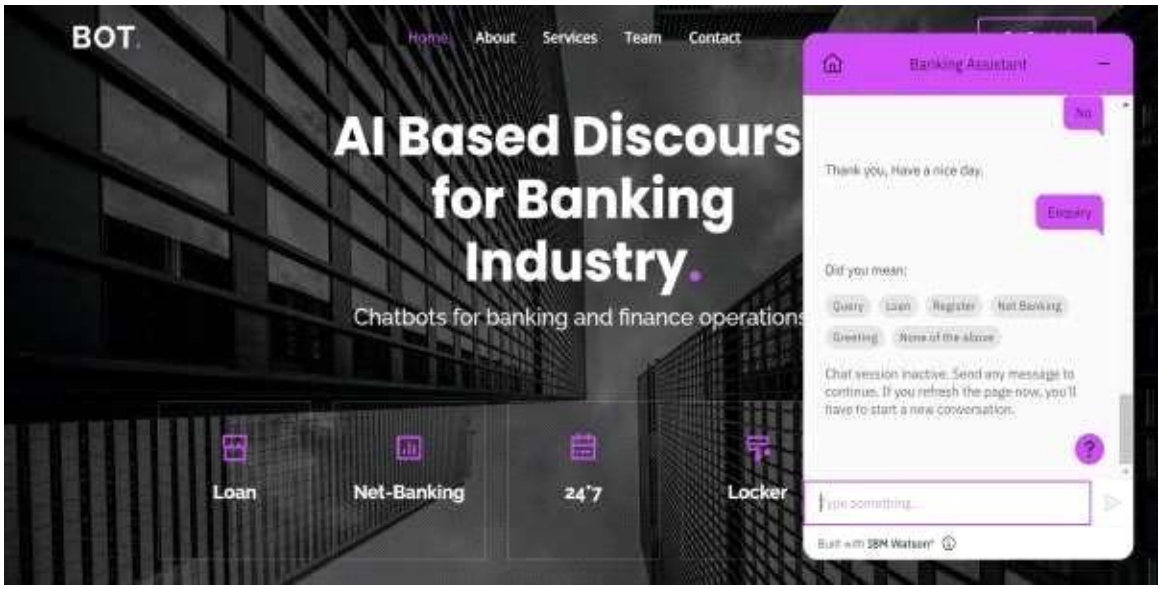
Run The Application

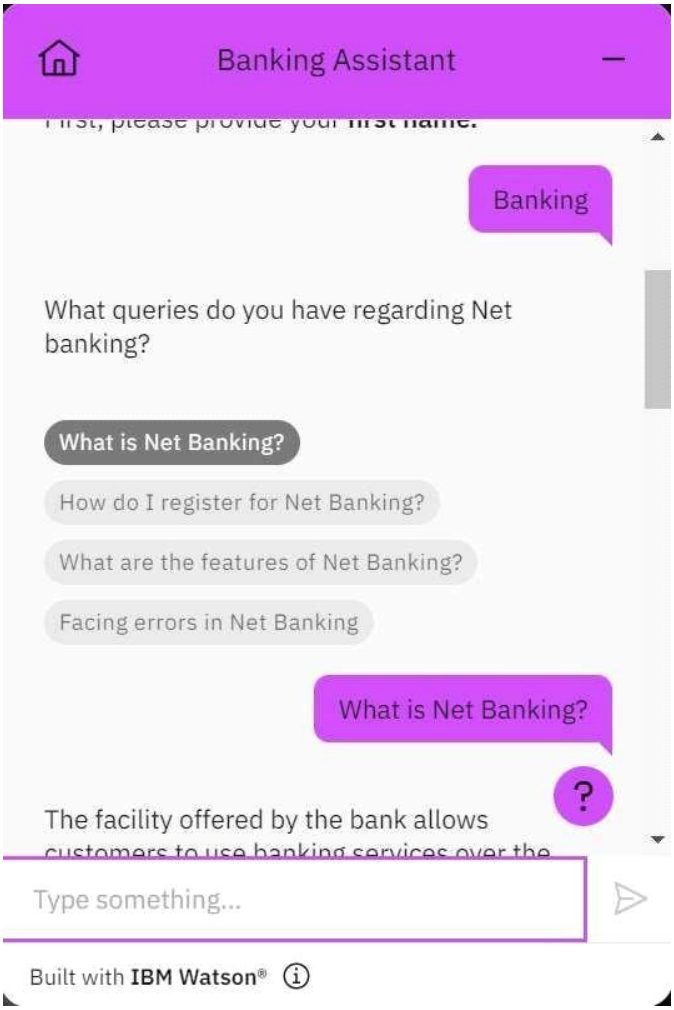
Run the application

- Open the anacondaprompt from the start menu.
- Navigate to the folder where your app.pyresides.
- Now type the"pythonapp.py"command.
- It will show the localhost where your app is running on http://127.0.0.1.5000/
- Copy that localhost URL and open that URL in the browser.Itdoes navigate me to where you can view yourwebpage.

Then itwill run on localhost:5000


BOT.
Home
About
Services
Team
Contact
Get Started
AI Based Discourse for Banking Industry.
Chatbots for banking and finance operations.
Loan
Net-Banking
24*7
Locker
Data Security
BOT.
Home
About
Services
Team
Contact
Get Started
AI Based Discourse for Banking Industry.
Chatbots for banking and finance operations.
Close
Hi! I'm a Banking Bot. How can I help you today?
Loan
Net-Banking
24*7
Locker
BOT.
Home
About
Services
Team
Contact
AI Based Discours for Banking Industry.
Chatbots for banking and finance operations
Return to assistant
Hi! I'm a Banking Bot. How can I help you today?
Banking
Enquiry
Loan
Type something...
Built with IBM Watson
Loan
Net-Banking
24*7
Locker


BOT.
Home
About
Services
Team
Contact
AI Based Discours
for Banking
Industry.
Chatbots for banking and finance operations
Loan
Net-Banking
24*7
Locker
Banking Assistant
Thank you, Have a nice day.
Did you mean:
Chat session inactive. Send any message to continue. If you refresh the page now, you'll have to start a new conversation.
Type something...
Built with IBM Watson®


Banking Assistant
Banking
What queries do you have regarding Net banking?
What is Net Banking?
How do I register for Net Banking?
What are the features of Net Banking?
Facing errors in Net Banking
What is Net Banking?
The facility offered by the bank allows
Type something...
Built with IBM Watson®

PROJECT:https://chatbotprojectibm.000webhostapp.com/

PREVIEW OF CHATBOT:

https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImag eURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-785992fb-b6cf-4d51-b222-23f37f3cee20%3A%3A33c532ec-f7b3-46f0-becb-d89ad77b3d68&integrationID=fafa4141-555c-427c-9e44-66a101cbb178&region=us-south&serviceInstanceID=785992fb-b6cf-4d51-b222-23f37f3cee20

Source code will be attached in Final Deliverables.