Project Development Phase
Delivery of Sprint - 4

Date	11 November 2022
Team ID	PNT2022TMID10377
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 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('/')
def bot():
    return render_template('chatbot.html')
```

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

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

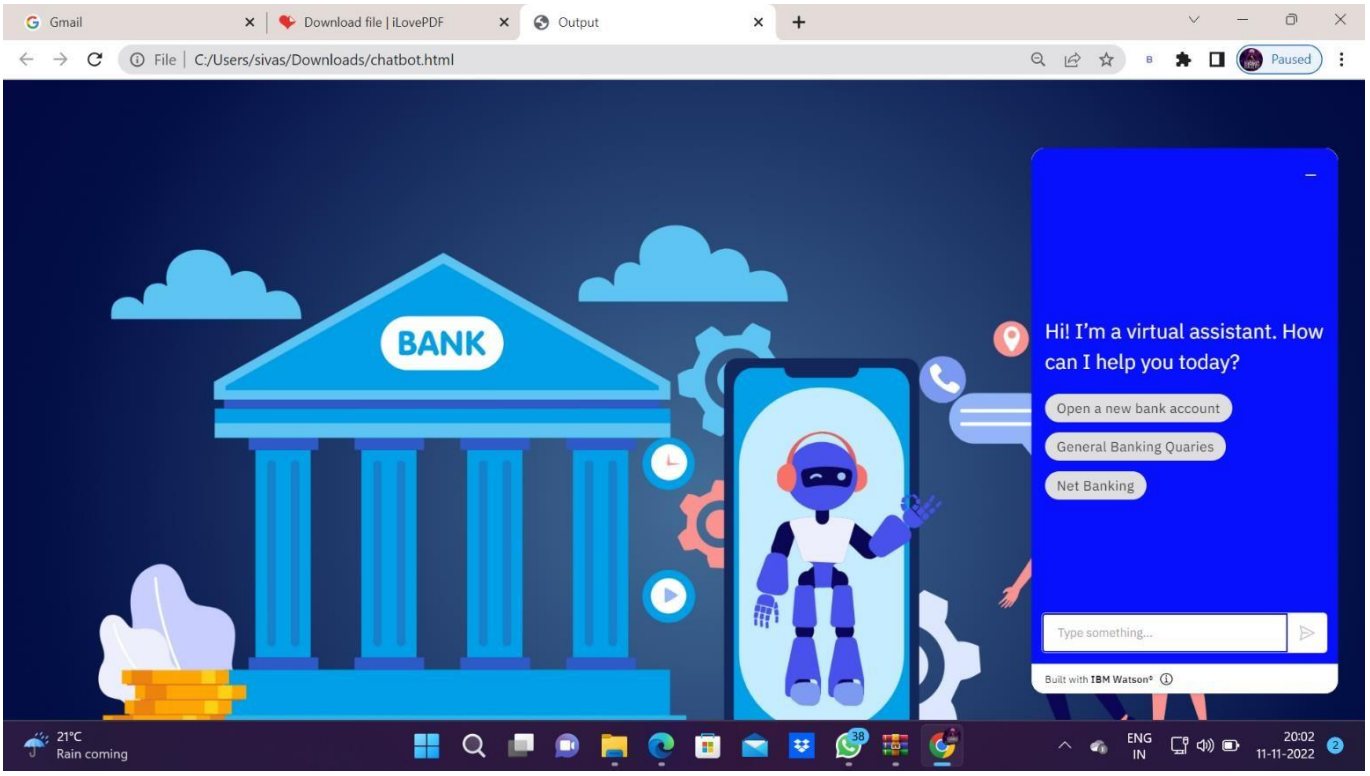
Build HTML Code

- 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

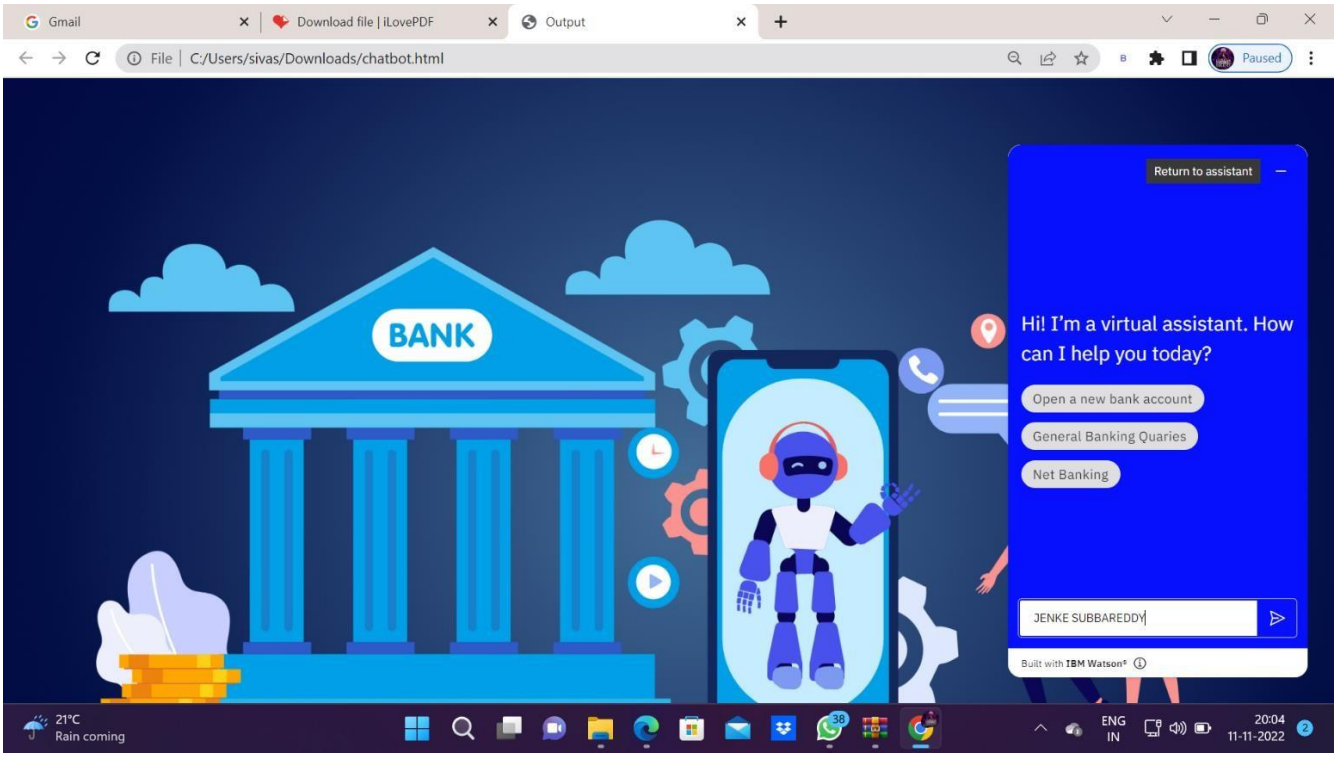
Run The Application

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 localhost 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



HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Output</title>
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
   <style>
      body
      {
      background-image: url("https://www.apptunix.com/blog/wp-content/uploads/sites/3/2021/04/show-
chatbots-for-banking.jpg");
      background-size: cover;
      }
   </style>
</head>
<body>
<script>
  window.watsonAssistantChatOptions = {
   integrationID: "1bd939b8-5c5c-4781-884c-93a6025f5955", // The ID of this integration.
   region: "us-south", // The region your integration is hosted in.
   serviceInstanceID: "4691697a-e720-498a-b0fd-72b57086e3f9", // The ID of your service instance.
   onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
   const t=document.createElement('script');
   t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
   document.head.appendChild(t);
  });
</script>
</body>
</html>
```

BOOTSTRAP :

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
```

NOTE : Here we are not using any kind of CSS Inplace of that we have used BootStrap.