

# Project Development

## Phase

### Delivery of Sprint- 4

Date	05 November 2022
Team ID	PNT2022TMID28540
Project Name	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 net banking.
- 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:ImportingLibraries

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__)
```

## • Routing to the HTML Page

Here, the declared construct or issued to route to the HTML page created earlier.

The '/' route is bound with the bot function. Hence, when the home page of a 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 HTML Code

- We use HTML to create the front-end part of the webpage.
- Here, we have created 1 HTMLpage-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

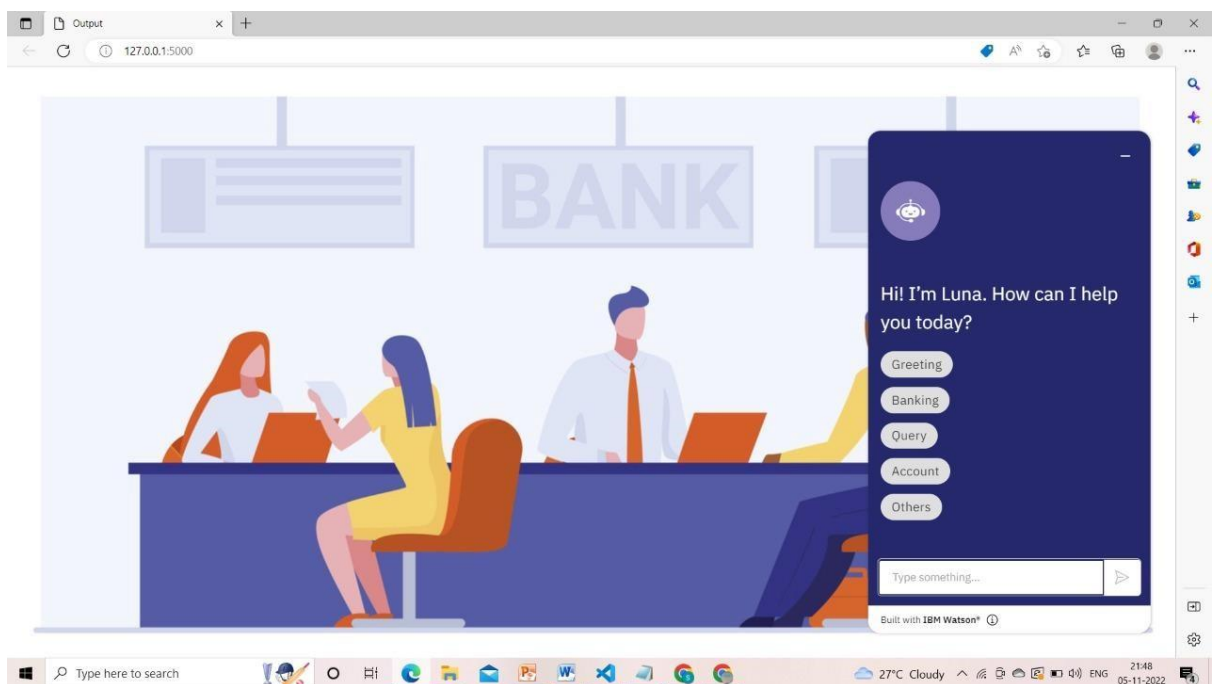
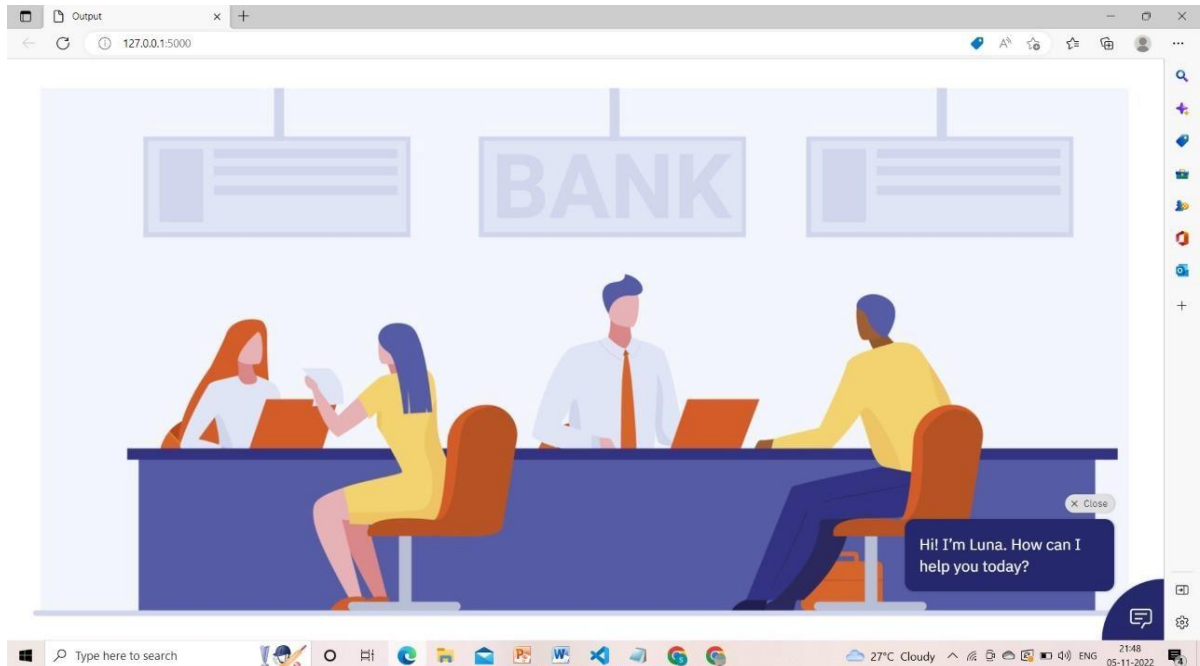
## Run The Application

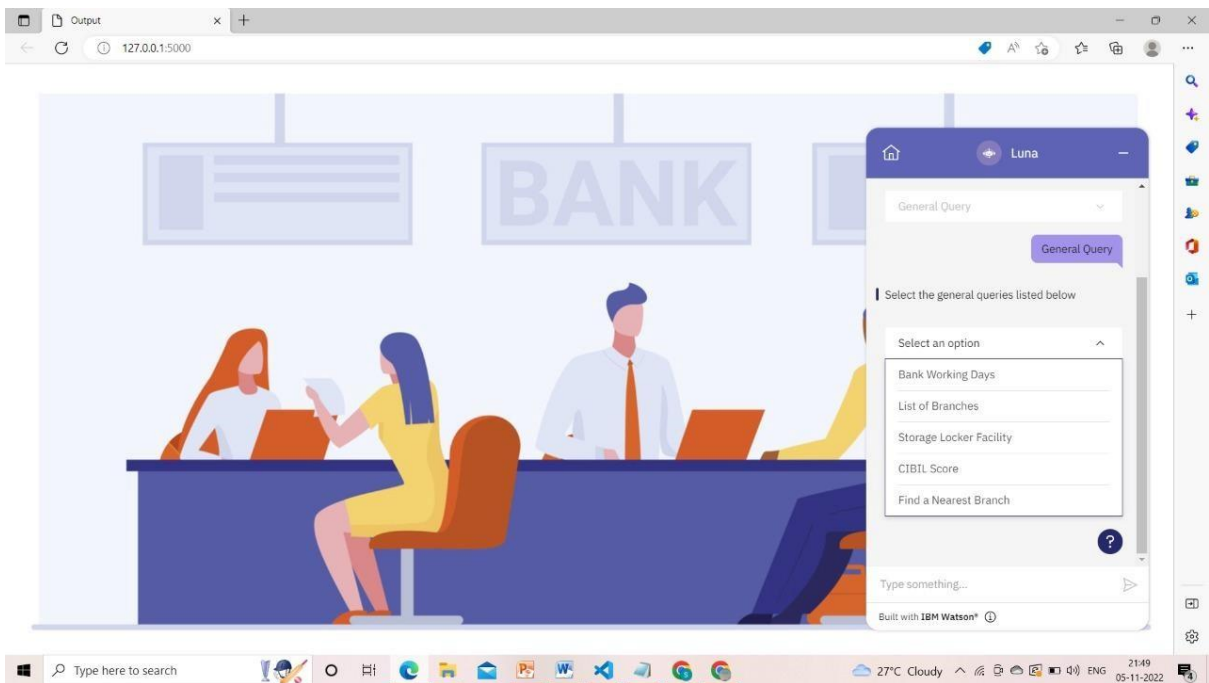
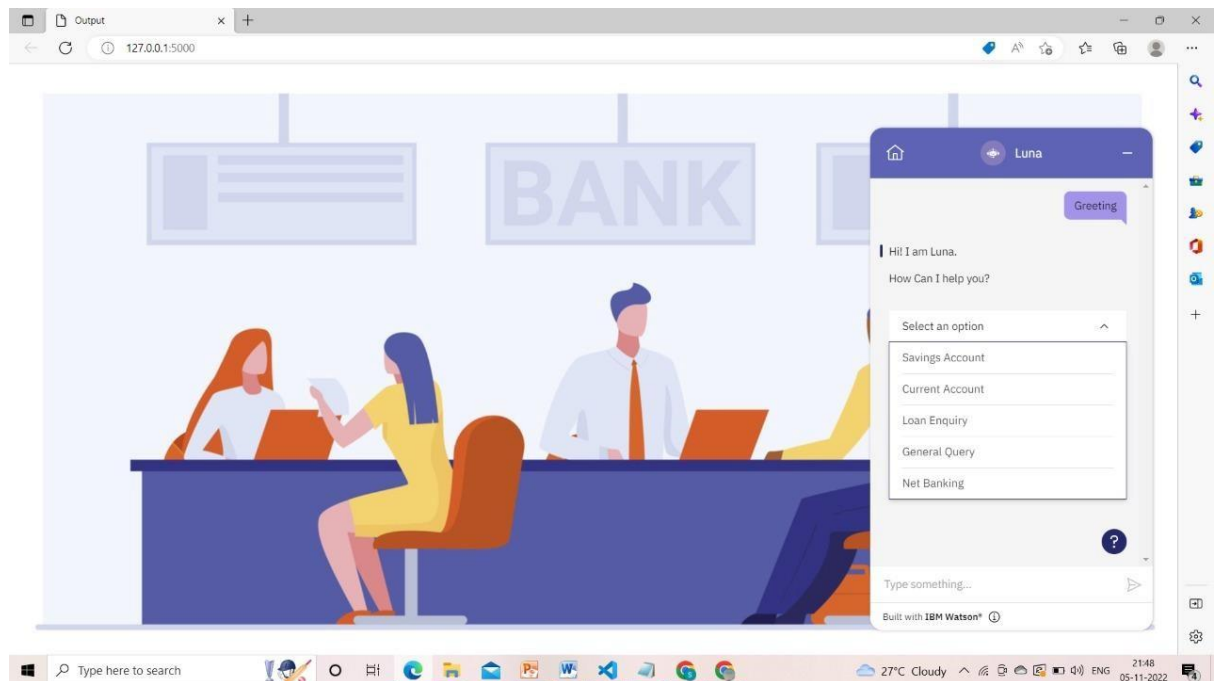
### Run the application

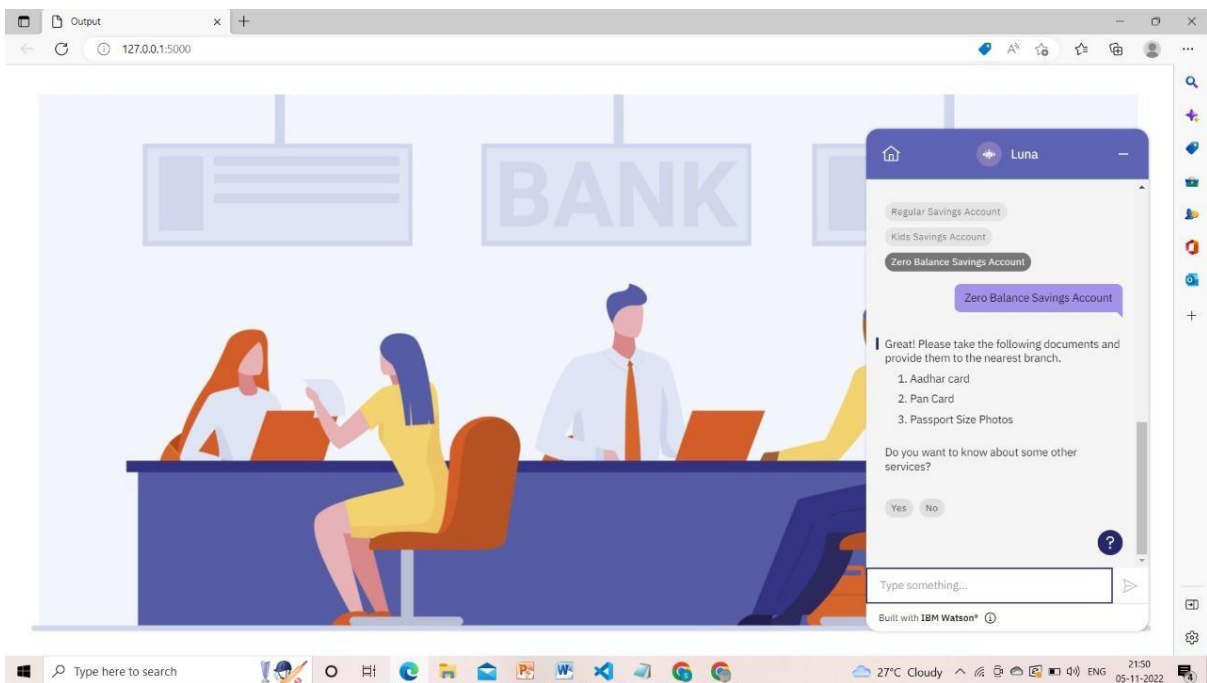
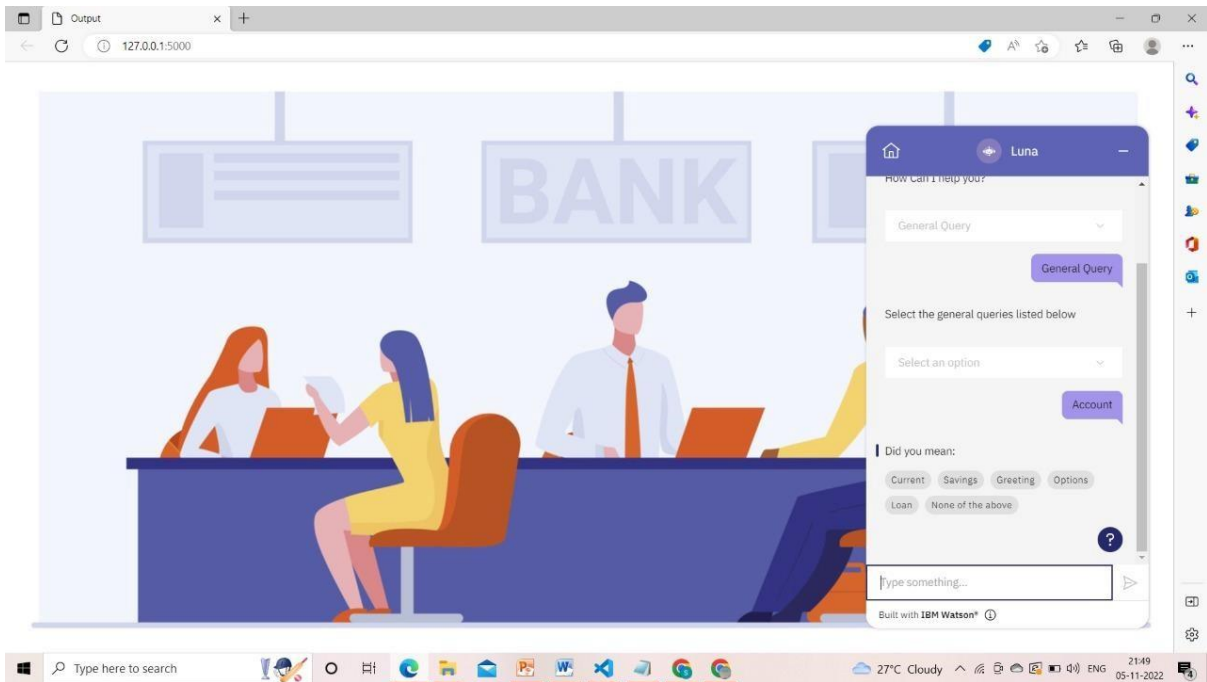
- Open the anaconda prompt from the start menu. □
- Navigate to the folder where your app.py resides.
- Now type the "pythonapp.py" command.
- It will show the localhost where your app is running on <http://127.0.0.1:5000/>

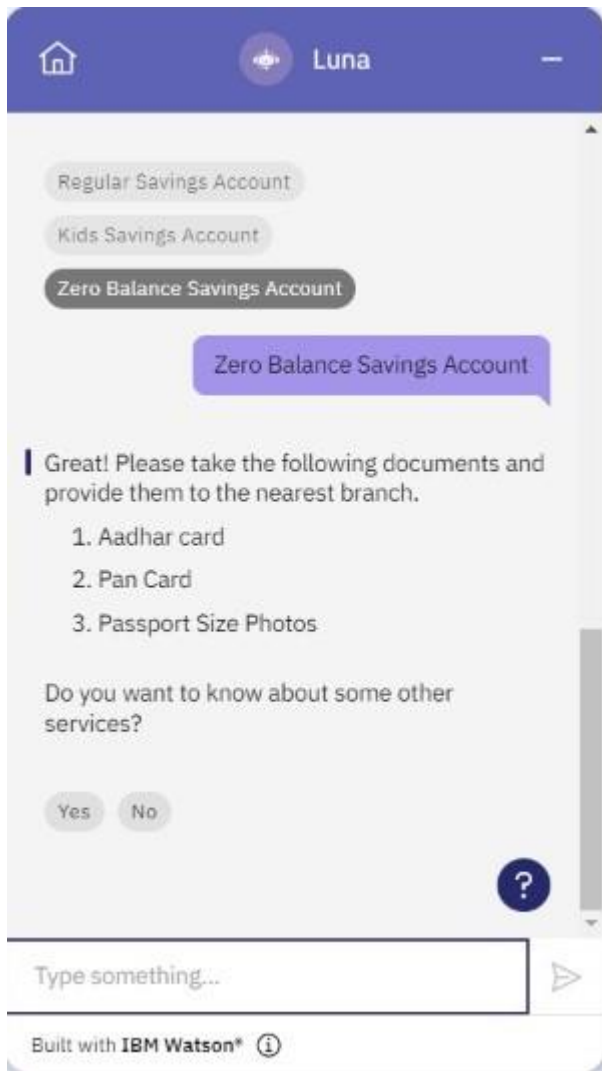
- Copy that localhost URL and open that URL in the browser. It does navigate me to where you can view your webpage.

Then it will run on localhost:5000









**PROJECT:** <https://chatbotprojectibm.000webhostapp.com/>

### **PREVIEW OF CHATBOT:**

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fu-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-8980c29e-0ec5-4b3f-b3ca-d81d8ec41e2b%3A%3A15a6900d-b12d-4882-926e-8e65badc9abc&integrationID=fcd8c7f8-333c-4518-9c8a-53d9b5ad1073&region=us-south&serviceInstanceID=8980c29e-0ec5-4b3f-b3ca-d81d8ec41e2b>