**Project Development Phase**

**Delivery of Sprint - 4**

|  |  |
| --- | --- |
| **Date** | 07 November 2022 |
| **Team ID** | PNT2022TMID10041 |
| **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 userinterface.

In the flask application, users will interact with the chat bot, and based on the user queriesthey will get the chatbotresponses.

**Building Python Code**

# 1: Importing Libraries

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



Importing the flask module into the project is mandatory. An object of theFlask class is ourWSGI application. Flask constructortakes the name of thecurrentmodule ( name ).

# 2: Creating our flask application and loading



# 3: Routing to the Html Page

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

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.



# Main Function

This is used to run the application in local host.



**Building HTML Code**

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

Here, wehave created“index.html” displays the home page which gets integrated with WatsonAssistant.

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

# Run the application

Open Jupyter notebook (anaconda3)

Navigate to the folder where app.ipynb resides.

Run the python code

Open a browser and type this URL <http://127.0.0.1:5000/>

It launches the application integrated with IBM Watson Assistant.









