

Project Development Phase

AI-based discourse for Banking Industry

TEAM ID: PNT2022TMID39294

TEAM MEMBERS

ROLE	TEAM MEMBERS NAME	ROLL NO
TEAM LEADER	JAGATHESWARAN .S	(422619104018)
TEAM MEMBER 1	DWARAGA. R.V	(422619104011)
TEAM MEMBER 2	JOSEPH VIKRAM. R	(422619104019)
TEAM MEMBER 3	DHIVYA .S	(422619104009)

Project Development - Delivery Of Sprint-4

Sprint	Functional Requirement (Epic)	User Story Number	Task	Story Points	Priority	Team Members
Sprint-4	Chatbot Deployment (HTML Website)	USN- 8	Build HTML Code	7	Medium	DHIVYA S
Sprint-4	Chatbot Deployment (Python Flask)	USN- 9	Build Python Code	7	Low	JOSEPH VIKRAM R
Sprint-4	Run the Application	USN- 10	Test the application	6	Medium	JOSEPH VIKRAM R

Delivery

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Hardware / Software requirements

Hardware:

Processor: Intel Pentium

Ram: 512 MB

Storage: 256 MB

Software:

IBM Watson Assistant, Browser

BUILD HTML CODE

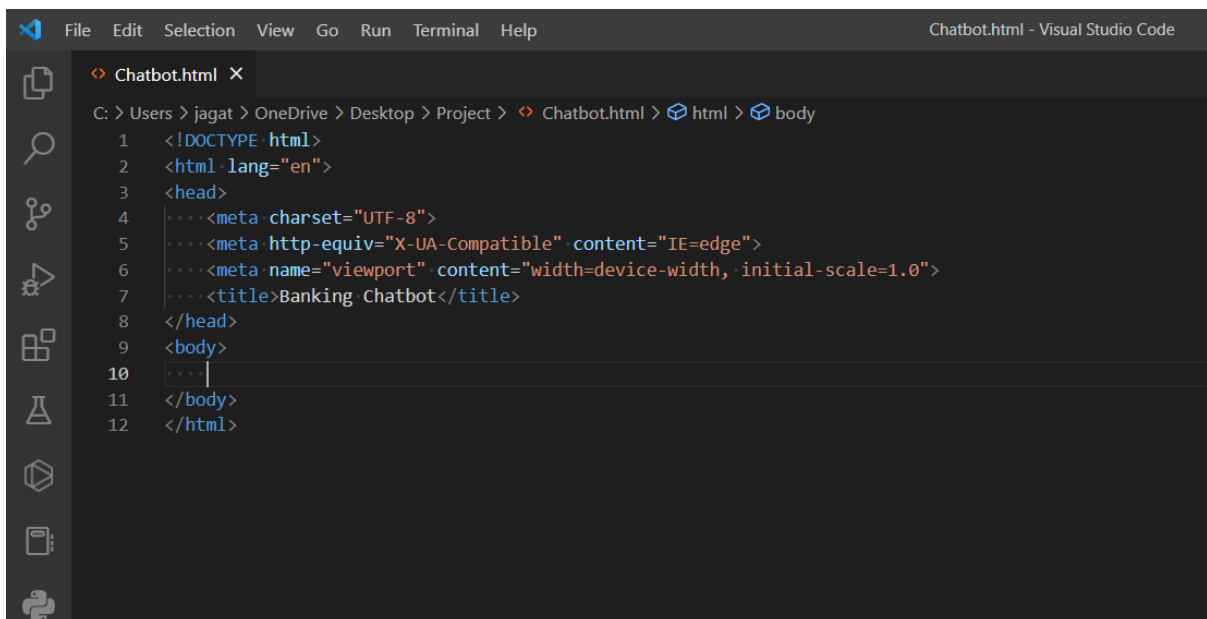
Task assigned: Dhivya. S

Task started on: 14 - 11 - 2022

Task completion date: - 11 - 2022

Progress 1: Building basic HTML structure code

In this step, we create the fundamental framework for the HTML code.

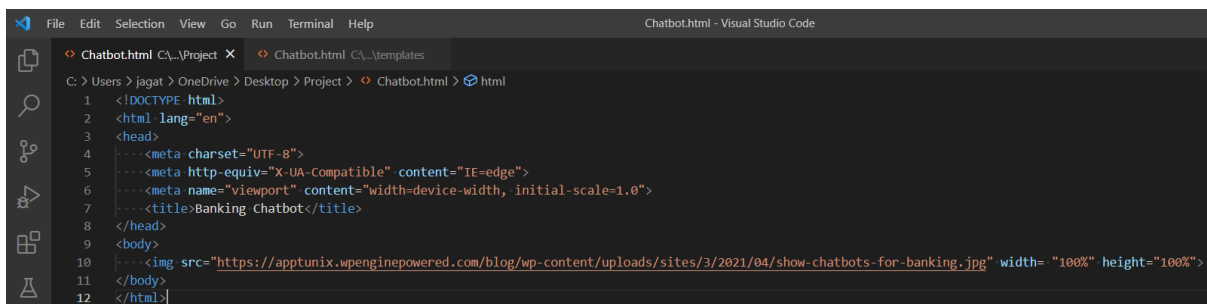


The screenshot shows the Visual Studio Code editor with the file 'Chatbot.html' open. The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Banking Chatbot</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Progress 2: Adding background image using URL

This step involves using a URL to add the background image for the Home page.

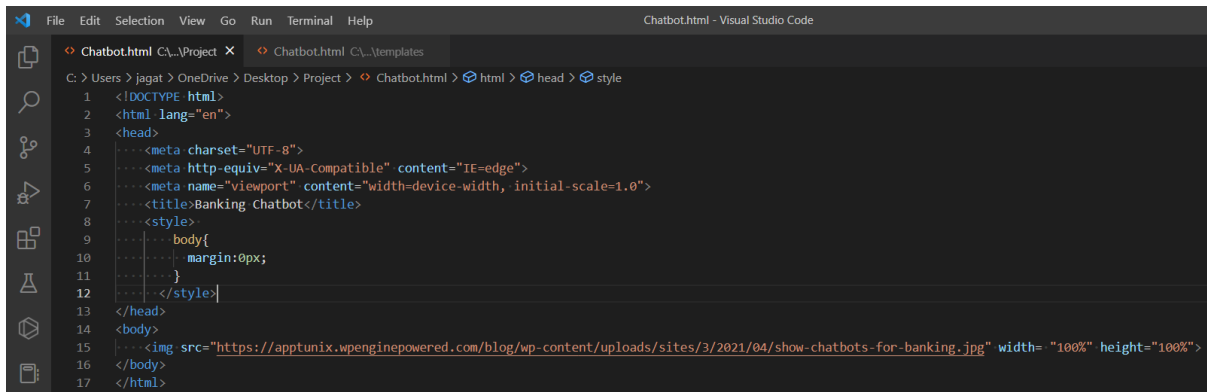


The screenshot shows the Visual Studio Code editor with the file 'Chatbot.html' open. The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Banking Chatbot</title>
8 </head>
9 <body>
10     
11 </body>
12 </html>
```

Progress 3: Adding CSS to the image

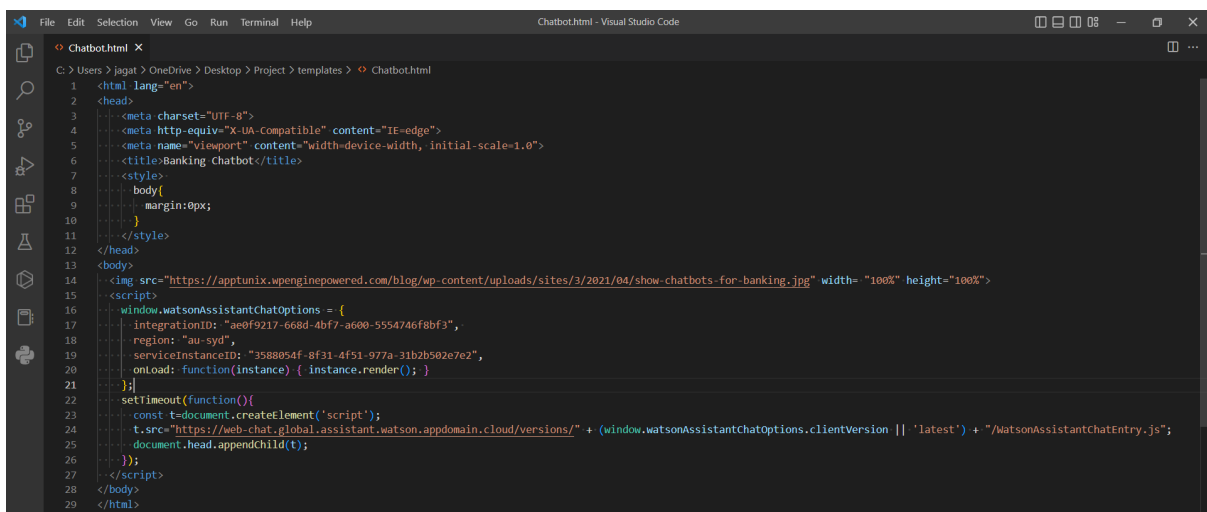
For picture repositioning in this stage, we're using CSS.



```
File Edit Selection View Go Run Terminal Help Chatbot.html - Visual Studio Code
C:\Users\jagat> OneDrive > Desktop > Project > Chatbot.html > html > head > style
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Banking Chatbot</title>
8 <style>
9 <body{
10 <body{
11 <body{
12 </style>
13 </head>
14 <body>
15 
16 </body>
17 </html>
```

Progress 4: Plug-in Watson assistant in HTML website

Using Javascript, we are integrating Watson Assistant in this stage.



```
File Edit Selection View Go Run Terminal Help Chatbot.html - Visual Studio Code
C:\Users\jagat> OneDrive > Desktop > Project > templates > Chatbot.html
1 <html lang="en">
2 <head>
3 <meta charset="UTF-8">
4 <meta http-equiv="X-UA-Compatible" content="IE=edge">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Banking Chatbot</title>
7 <style>
8 <body{
9 <body{
10 <body{
11 </style>
12 </head>
13 <body>
14 
15 <script>
16 window.watsonAssistantChatOptions = {
17 integrationID: "ae0f9217-668d-4bf7-a600-5554746f8bf3",
18 regions: "au-syd",
19 serviceInstanceID: "3588054f-8f31-4f51-977a-31b2b502e7e2",
20 onLoad: function(instance) { instance.render(); }
21 };
22 setTimeout(function(){
23 const t=document.createElement('script');
24 t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
25 document.head.appendChild(t);
26 });
27 </script>
28 </body>
29 </html>
```

BUILD PYTHON CODE

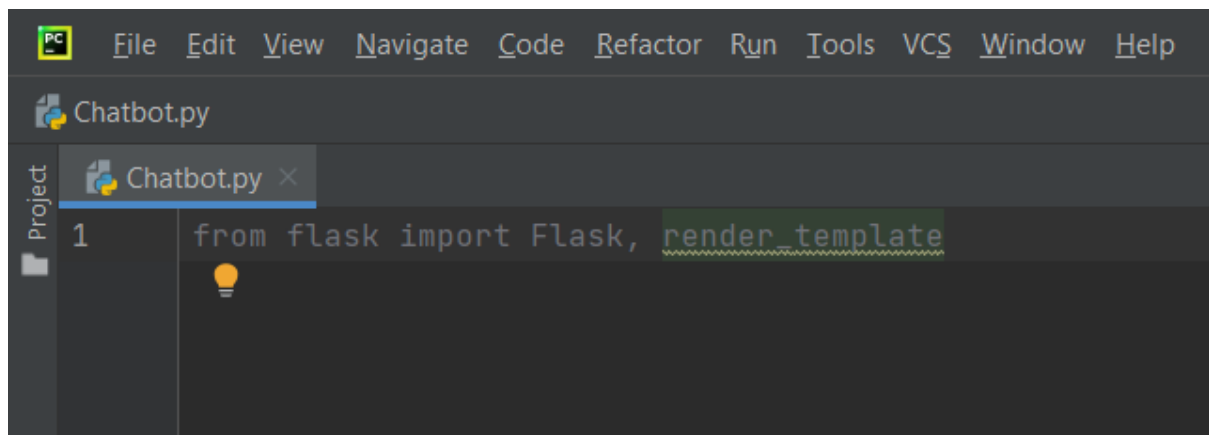
Task assigned: Joseph Vikram. R

Task started on: 14 - 11 - 2022

Task completion date: 14 - 11 - 2022

Progress 1: Importing Libraries

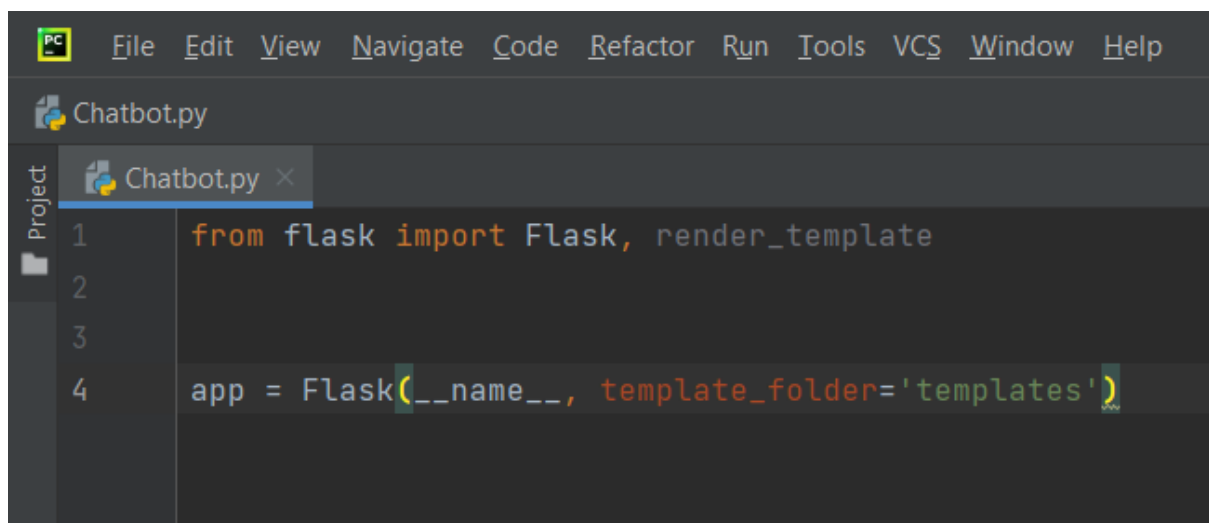
The first step is usually importing the libraries that will be needed in the program.



```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
```

Progress 2: Creating our flask application and loading

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. The Flask constructor takes the name of the current module (`__name__`).

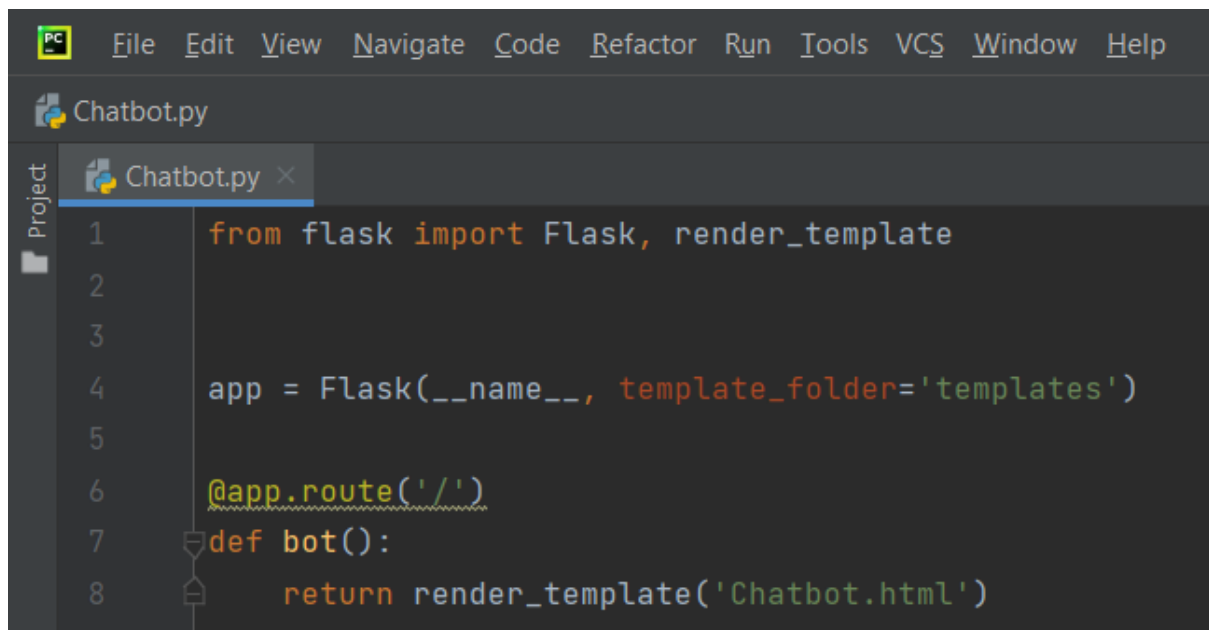


```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
```

Progress 3: Routing to the Html Page

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

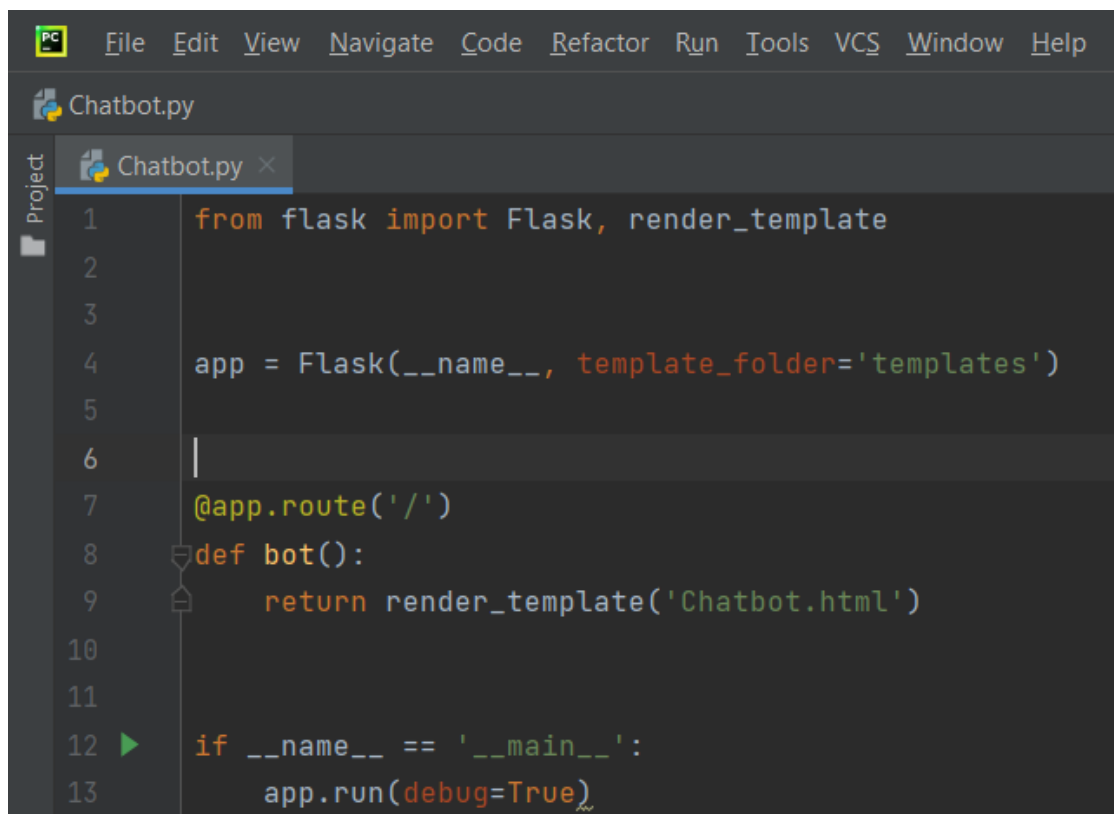
The '/' route is bound with the bot function. Hence, when the home page of a web server is opened in the browser, the HTML page will be rendered.



```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
5
6 @app.route('/')
7 def bot():
8     return render_template('Chatbot.html')
```

Progress 4: Main Function

This is used to run the application in localhost.



```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help
Chatbot.py
Project Chatbot.py x
1 from flask import Flask, render_template
2
3
4 app = Flask(__name__, template_folder='templates')
5
6 |
7 @app.route('/')
8 def bot():
9     return render_template('Chatbot.html')
10
11
12 if __name__ == '__main__':
13     app.run(debug=True)
```

RUN THE APPLICATION

Task assigned: Joseph Vikram. R

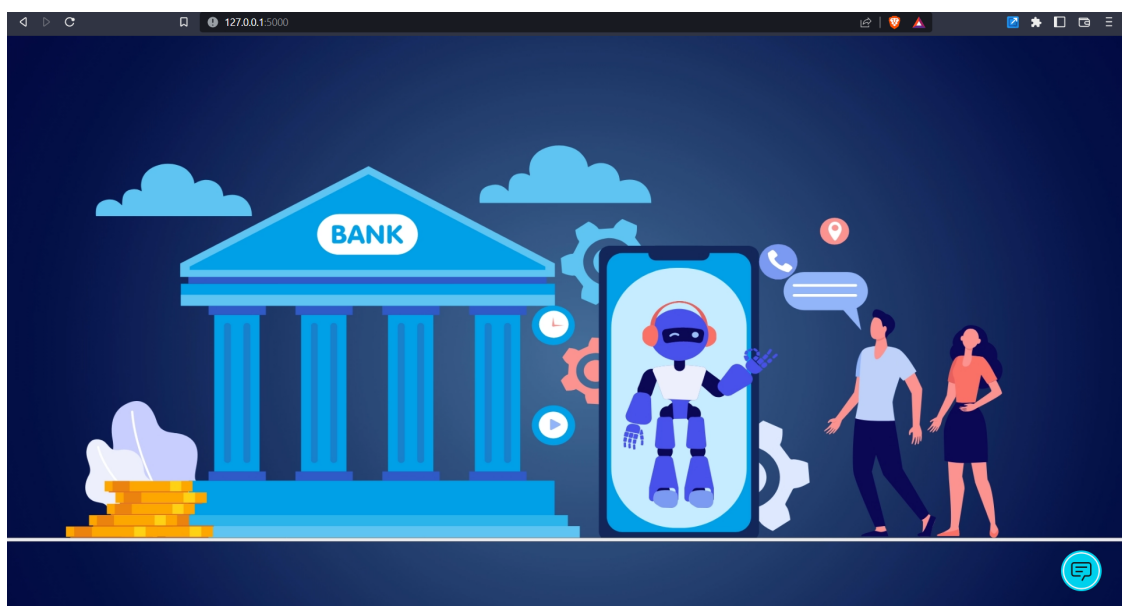
Task started on: 14 - 11 - 2022

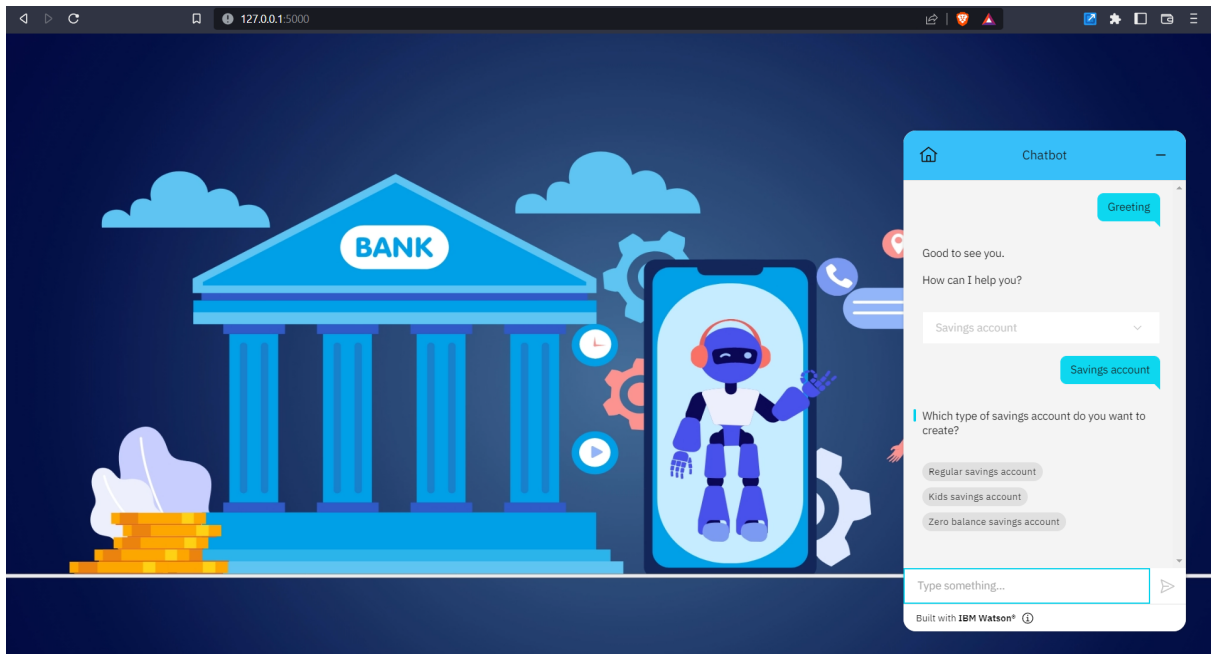
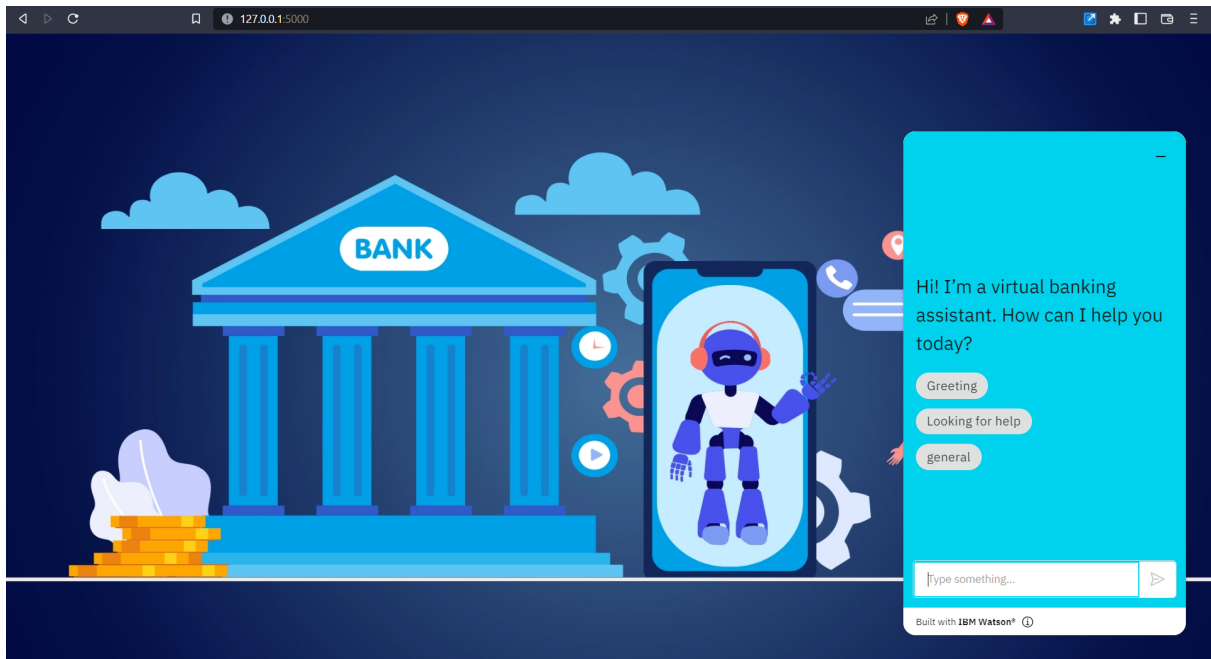
Task completion date: 14 - 11 - 2022

Progress 1: Run the python flask file. It will show the local host where your app is running on <http://127.0.0.1:5000/>

```
C:\windows\py.exe x + v
* Serving Flask app 'Chatbot'
* Debug mode: on
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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 417-829-865
```

Progress 2: Copy that localhost URL and open that URL in the browser. and open Watson's assistant chatbot.





Testing the chat

Testing 1: Python Testing

```
C:\windows\py.exe x + v
* Serving Flask app 'Chatbot'
* Debug mode: on
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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 417-829-865
```

Test result: passed

Testing 2: HTML Testing



Test result: passed

Jira report:

Projects / AI Based Discourse For Banking Industry / Reports

Sprint burndown chart

[How to read this report](#)

Sprint

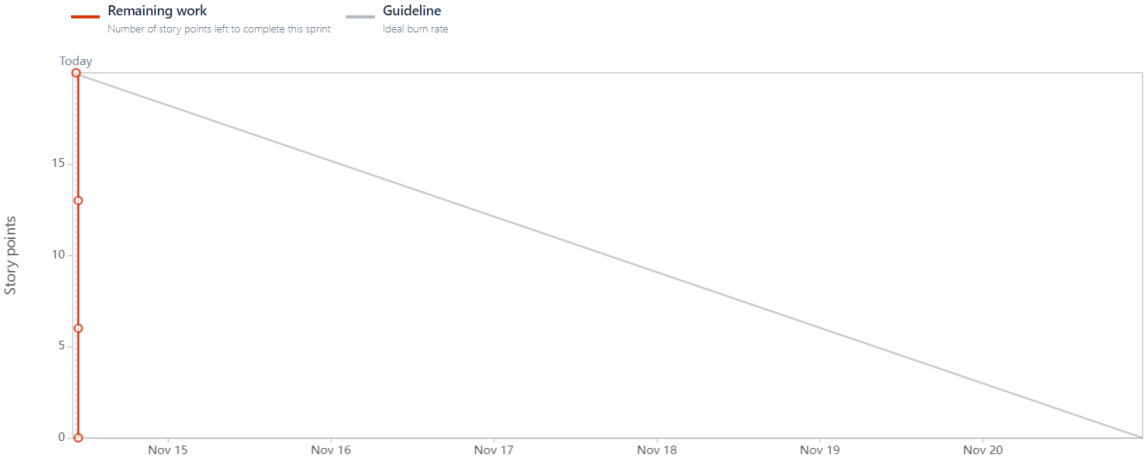
ABDFBI Sprint 4

Estimation field

Story points

...

Date - November 14th, 2022 - November 20th, 2022
Sprint goal - Build HTML Code, Build Python Code, and Test the application.



Backlog

...

Q

JS JV D

...

Epic

Insights

Epic

Issues without epic

Dashboard

Chatbot Skills

ABDFBI Sprint 4 14 Nov – 20 Nov (3 issues)

Build HTML Code, Build Python Code, and Test the application.

0 0 20 Complete sprint

ABDFBI-19 Build HTML Code CHATBOT DEPLOYMENT 7 DONE D

ABDFBI-20 Build Python Code CHATBOT DEPLOYMENT 7 DONE JV

ABDFBI-21 Test the application RUN THE APPLICATION 6 DONE JV

Projects / AI Based Discourse For Banking Industry

Roadmap

[Give feedback](#) [Share](#) [Export](#) ...

Q

JS JV D DR

...

Status category

Epic

View settings

	T	NOV	DEC
Sprints	ABDF...	ABDFBI... ABDFBI... ABDFBI...	
ABDFBI-6 Dashboard			
ABDFBI-7 Chatbot Skills			
ABDFBI-8 Chatbot Deployment			
ABDFBI-10 Run the Application			

Velocity report

[How to read this report](#)

