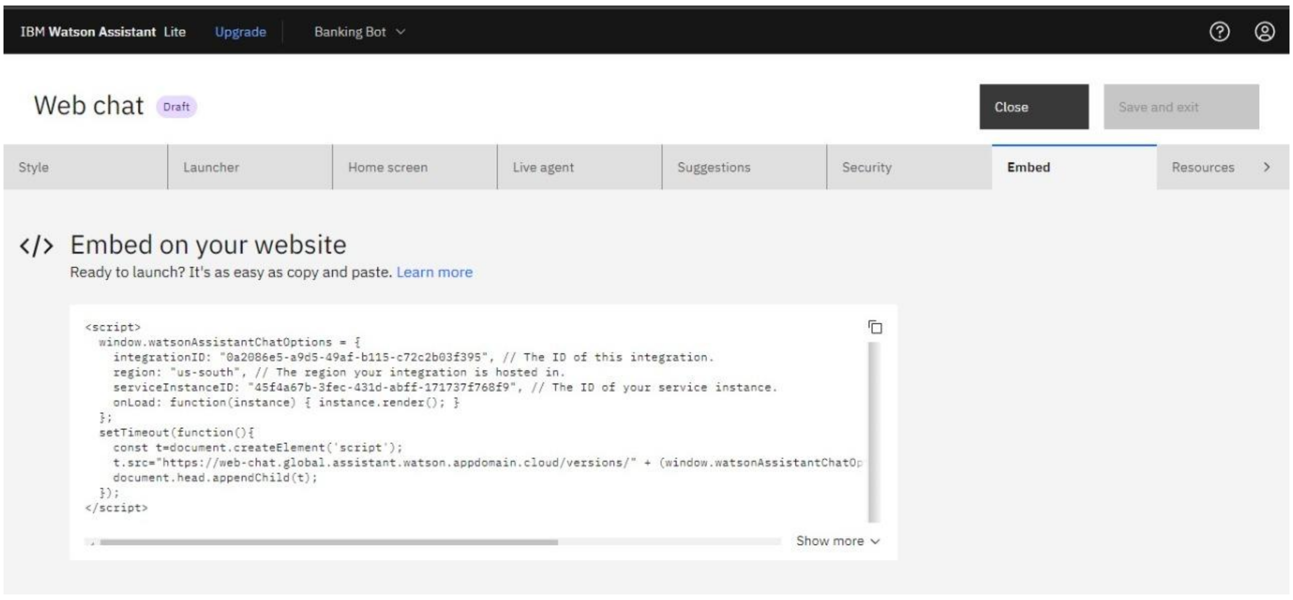
7. Coding & solution

7.1 Feature 1

Create the IBM Watson Assistant services to create the actions for greeting, index and end actions are created. And creating the chatbot skills, savings account action, Current account action, Loan account action, General query account action, Net Banking account action are successfully created and link the all action to index choose the preview to perform the chatbot action. Create the Assistant and choose the assistant name, choose the primary color, assent color, secondary color choose then go to embed choose the JavaScript file copy and paste the files in the html code.

Java script for Watson Assistant

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "0a2086e5-a9d5-49af-b115-c72c2b03f395", // The ID of this integration.
    region: "us-south", // The region your integration is hosted in.
    serviceInstanceID: "45f4a67b-3fec-431d-abff-171737f768f9", // 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);
  });
```



Embed on your website

Ready to launch? It's as easy as copy and paste. Learn more

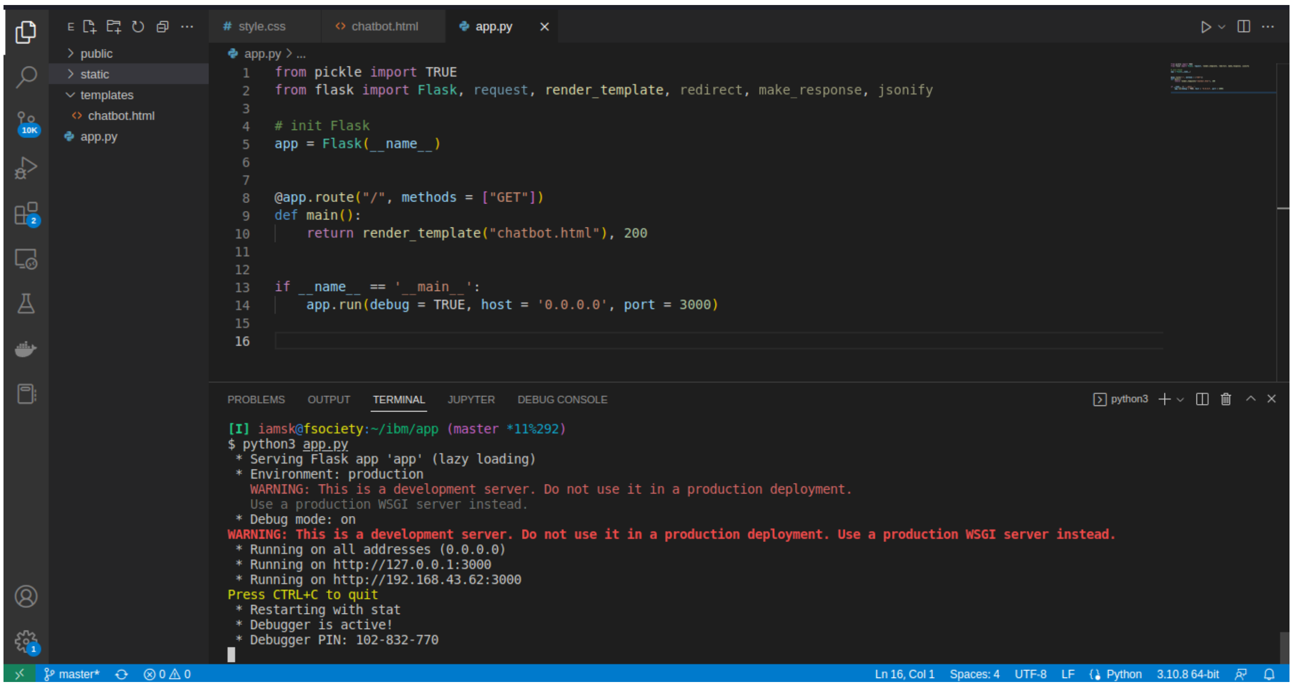
```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "0a2086e5-a9d5-49af-b115-c72c2b03f395", // The ID of this integration.
    region: "us-south", // The region your integration is hosted in.
    serviceInstanceID: "45f4a67b-3fec-431d-abff-171737f768f9", // The ID of your service 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.watsonAssistantChatOp
    document.head.appendChild(t);
  });
</script>
```

7.2 Feature 2

Integrate the web page in the python flask.

Flask

```
from pickle import TRUE
from flask import Flask, request, render_template, redirect, make_response, jsonify
# init Flask
app = Flask(__name__)
@app.route("/", methods = ["GET"])
def main():
        return render_template("chatbot.html"), 200
if __name__ == '__main__':
    app.run(debug = TRUE, host = '0.0.0.0', port = 3000)
```



```
from pickle import TRUE
from flask import Flask, request, render_template, redirect, make_response, jsonify

# init Flask
app = Flask(__name__)


@app.route("/", methods = ["GET"])
def main():
    return render_template("chatbot.html"), 200


if __name__ == '__main__':
    app.run(debug = TRUE, host = '0.0.0.0', port = 3000)

```

```
[I] iamsk@fsociety:~/ibm/app (master *11%292)
$ python3 app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * 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 all addresses (0.0.0.0)
 * Running on http://127.0.0.1:3000
 * Running on http://192.168.43.62:3000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 102-832-770
```

8. Testing

8.1 Test Scenarios

1. Verify user is able to see the chatbot icon when website is launched
2. Verify the UI elements in chatbot icon popup
3. Verify user is able to see the greeting from chatbot "Hi! I'm a Banking Bot. How can I help you today? Banking Enquiry Loan
4. Verify user is able to type query in text field.
5. Verify user is able to get the response from chatbot
6. Verify user whether get the response if the user enter the wrong query also

Search

1. Chatbot icon should display.
2. After 30 seconds Information about chatbot popup displayed
3. User should see the greeting message from chatbot
4. User able to type the query in text field.
5. Users get the response from chatbot.
6. Kindly reach out to our customer care executive. Contact Us @944xxxxx36

8.2 User Acceptance Testing (UAT)

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the AI-based discourse for Banking Industry project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Security 1	Security 2	Security 3	Security 4	Subtotal
By Design	1	0	0	0	1
Duplicate	3	1	0	1	5
External	1	3	0	1	5
Fixed	2	5	3	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won’t Fix	0	0	0	0	0
Totals	7	9	3	5	24

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1
Security	1	0	0	1
Outsource Shipping	0	0	0	0
Exception Reporting	1	0	0	1
Final Report Output	1	0	0	1
Version Control	1	0	0	1

Source Code:

HTML Code

Chatbot.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Internet Banking</title>
    <link rel="icon" type="image/x-icon" href="/static/favicon.png">
    <link rel="stylesheet" href="/static/style.css">
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "0a2086e5-a9d5-49af-b115-c72c2b03f395", // The ID of this integration.
    region: "us-south", // The region your integration is hosted in.
    serviceInstanceID: "45f4a67b-3fec-431d-abff-171737f768f9", // 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>
</head>
<body>
    <div class="landing-page">
        <div class="container">
            <div class="header-area">
                <div class="logo"> <h2>Banking&Finance</h2></div>
                <ul class="links">
                    <li>Home</li>
                    <li>About Us</li>
                    <li>Services</li>
                    <li>Contact Us</li>
                </ul>
            </div>
            <div class="info">
                <h1>Banking Services</h1>
                <p>Banking includes a wide variety of financial institutions that store the money of
individuals, businesses and other entities. Banks provide financial services that help people save,
manage and invest their money.</p>
                <button>Read More</button>
            </div>
            <div class="image">
                <img src="static/home.png" alt="">
            </div>
            <div class="clearfix"></div>
        </div>
    </div>
</body>
</html>
```

<u>CSS Code:</u>

Style.css

```
*{
   box-sizing: border-box;
}
body{
   font-family: sans-serif;
}
.container{
   width: 1170px;
   padding-right: 15px;
   padding-left: 15px;
   margin: auto;
}
.landing-page{
   position: relative;
   background-color: white;
}
.landing-page .header-area{
   display: flex;
   padding: 25px 0 0;
   position: relative;
}
.landing-page .header-area .logo{
   font-style: Times New Roman;
   margin-top: 10px;
   font-size: 19px;
   width: 300px;
   color: #5d5d5d;
}
.landing-page .header-area .links{
   list-style: none;
   padding: 0;
   margin: 0;
   width: 100%;
```

```
    text-align: right;
}
.landing-page .header-area .links li{
    display: inline-block;
    margin-left: 30px;
    color: #5d5d5d;
    cursor: pointer;
}
.landing-page .header-area .links li:last-child{
    border: 0;
    border-radius: 20px;
    padding: 10px 18px;
    color: white;
    background-color: #6c63ff;
}
.landing-page .info{
    width: 35%;
    float: left;
    margin-top: 130px;
}
.landing-page .info h1{
    font-size: 44px;
    margin: 0 0 20px;
    line-height: 1.4;
    color: #5d5d5d;
}
.landing-page .info p{
    margin: 0;
    line-height: 1.6;
    font-size: 15px;
    color: #5d5d5d;
}
.landing-page .info button{
    border: 0;
    border-radius: 20px;
    padding: 12px 30px;
    margin-top: 30px;
    cursor: pointer;
    color: white;
    background-color: #6c63ff;
}
.landing-page .image{
    width: 50%;
    float: right;
    margin-top: 35px;
}
.landing-page .image img{
    max-width: 100%;
}
.clearfix{
    clear: both;
}
```

Python Flask:

App.py

```
from pickle import TRUE
from flask import Flask, request, render_template, redirect, make_response, jsonify
# init Flask
app = Flask(__name__)
@app.route("/", methods = ["GET"])
def main():
    return render_template("chatbot.html"), 200
if __name__ == '__main__':
    app.run(debug = TRUE, host = '0.0.0.0', port = 3000)
```

GITHUB LINK: PNT2022TMID48085

PREVIEW: Chatbot preview link