AI BASED DISCOURSE FOR BANKING INDUSTRY

INTRODUCTION:

Overview:

Industries are forced to evolve and update their practices due to technological advances and the contemporary market. The banking sector is one of the most developed sectors and is always looking for the latest technological solutions that improve its efficiency.

Net banking websites are complex and involve navigating through a lot of pages to find the information you need. Bank staff undergoes a lot of stressful situations when communicating with clients directly. Such situations can be avoided gracefully by using chatbots.

Only 32% of companies in the finance industry currently use AI chatbots, and 37% are planning to start using them within 18 months said a report from Salesforce. This results in a potential growth rate of 118% which indicates the demand in the industry.

A smart chatbot takes a query from the user in natural language and gives the appropriate response for the same. This paper aims to discuss the relevance of chatbots in the banking sector and explore how chatbots can be implemented using natural language processing techniques that can be used in the banking industry.

Proposed solution:

- The solution to the problem is Artificial intelligence in the banking sector makes banks efficient, trustworthy, helpful, and more understanding. It is strengthening the competitive edge of modern banks in this digital era. The growing impact of AI in banking sector minimizes operational costs improves customer support and process automation.

- Nearly 40% to 50% of financial and banking service providers are using AI in their processes to harness the power of next-generation AI capabilities. The companies believe that AI is the future of banking sector which can perform a range of banking operations in faster, easier, and more secure ways.

- AI banking Chatbots help customers in many ways. AI-based chatbot service for financial industry is one of the significant use cases of AI in banking sector. AI chatbots in banking are modernizing the way how businesses provide services to their customers

- AI chatbots in the banking industry can assist customers 24*7 and give accurate responses to their queries. These chatbots provide a personalized experience to users.

- AI chatbots in banking is providing a better customer experience.

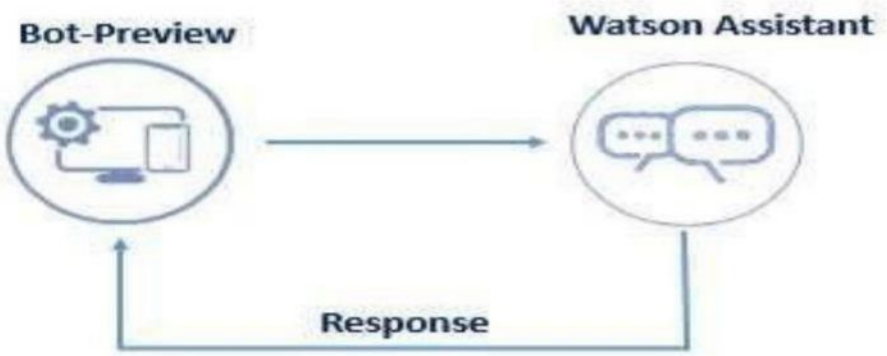
➢ Hence, AI chatbots for banking and finance operations let banks attract customer attention, optimize service quality, and expand the brand mark in the market.

THEORETICAL ANALYSIS:

Services Used:

- IBM Watson Assistant

Block diagram:



Hardware / Software designing:

To complete this project, you should have the following software and packages.

Softwares:

- Visual studio code
- IBM Watson studio

Packages:

- Flask

FLOWCHART:

To accomplish the above task, you must complete the below activities and tasks:

- Create IBM Services.
- Creating skills & Assistant for Chatbot.
- Creating Savings account action.
- Creating Current account action.
- Creating Loan account action.
- Creating a general query action.
- Creating a Net banking action.
- Create HTML web page.
- Integrate the Watson Chatbot with web page.

ADVANTAGES & DISADVANTAGES:

Advantages:

- Round-the-clock service.
- Brand Consistency.
- Increased Productivity.
- Reduced Staffing Needs.
- Consistent Response Rate and Availability.
- Helps with Fraud Prevention.
- Chats can be saved.
- Lower costs.

Disadvantages:

- Questions must be programmed beforehand.
- Impersonal
- Must keep information up-to-date.
- Technology issues.
- Needs additional measures to protect identities.

APPLICATIONS:

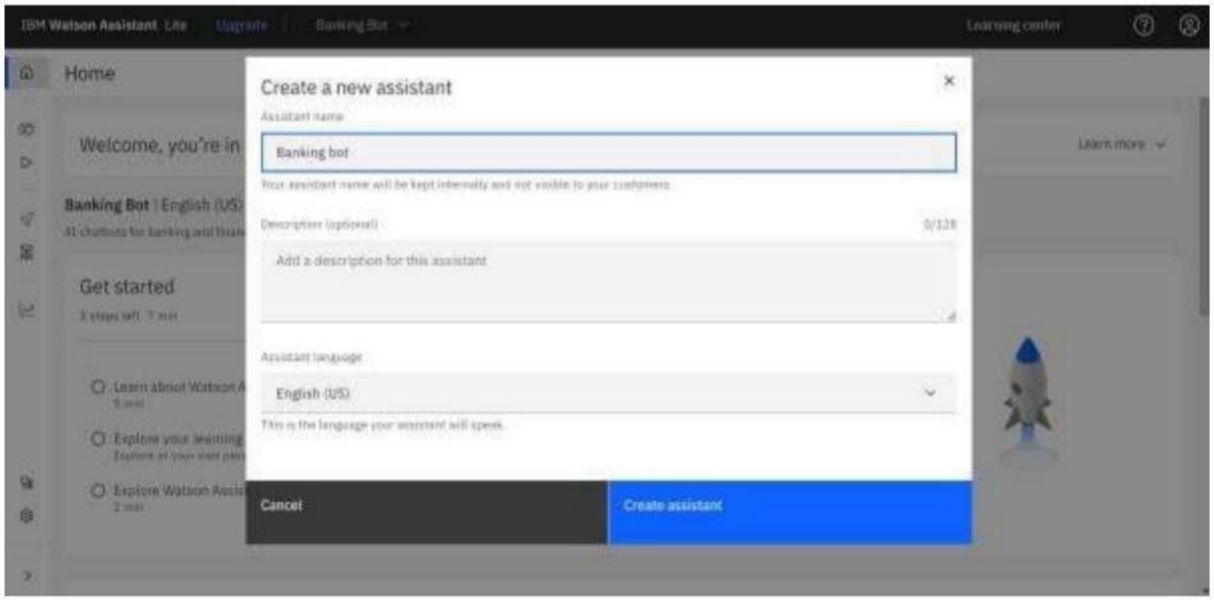
- Banking chatbots have all the data to predict the spending habits of customers and help them keep their finances on track.

APPENDIX:

Create IBM Service

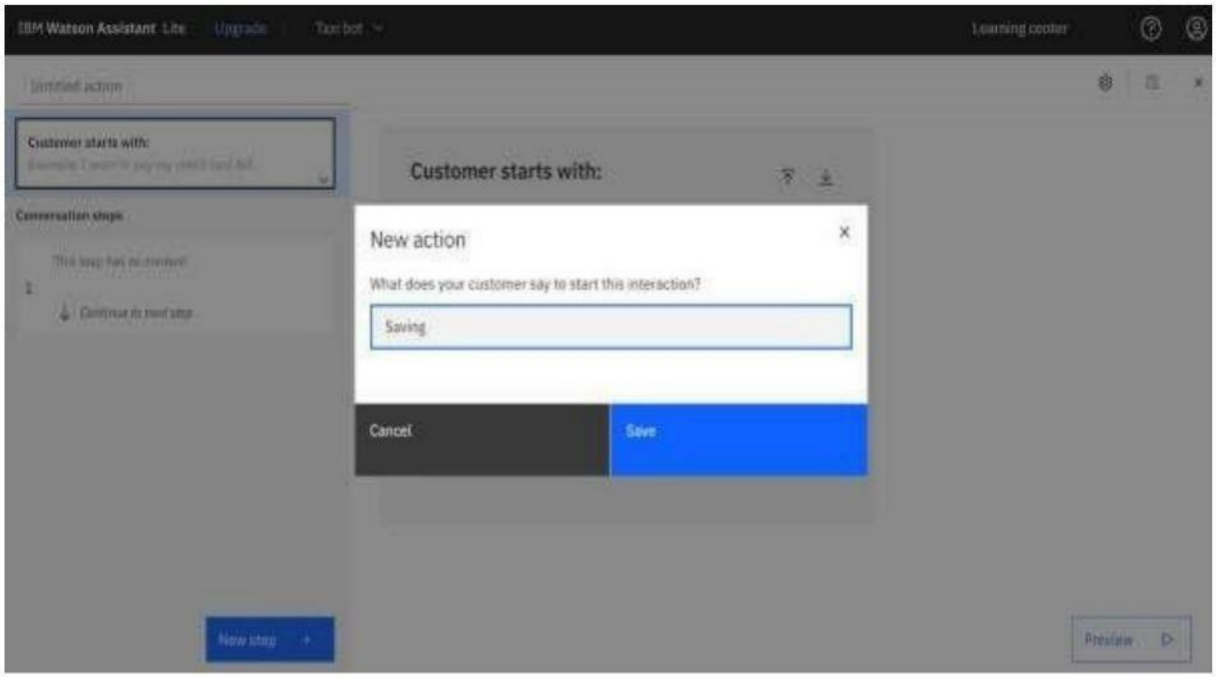
In this activity, you will be creating the Necessary IBM service. The following are the service that you have to create.

- Watson Assistant

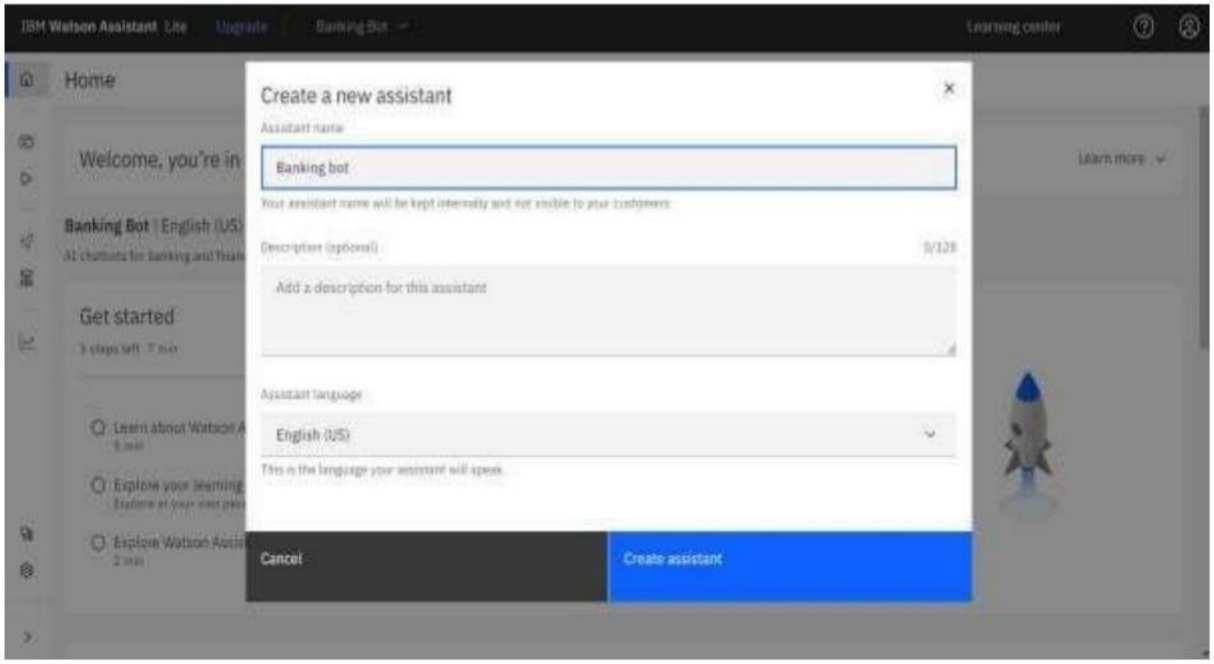


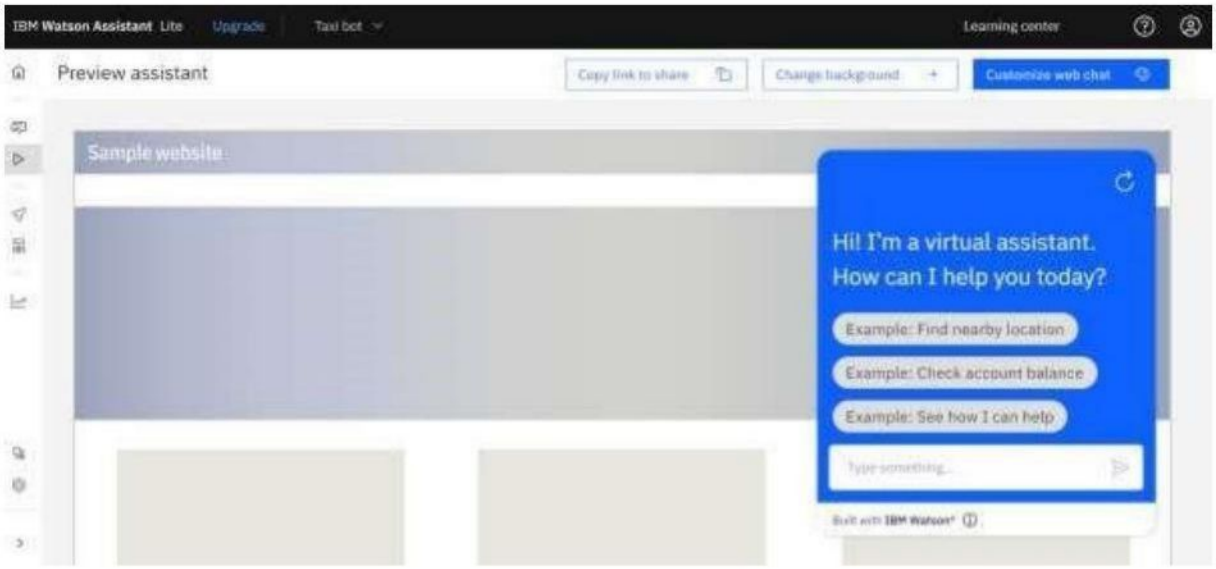
Creating Skills & Assistant For Chatbot

Skills are nothing but actions and steps. Steps are the subset of actions where conversations are built and Assistant is used to integrate skills.



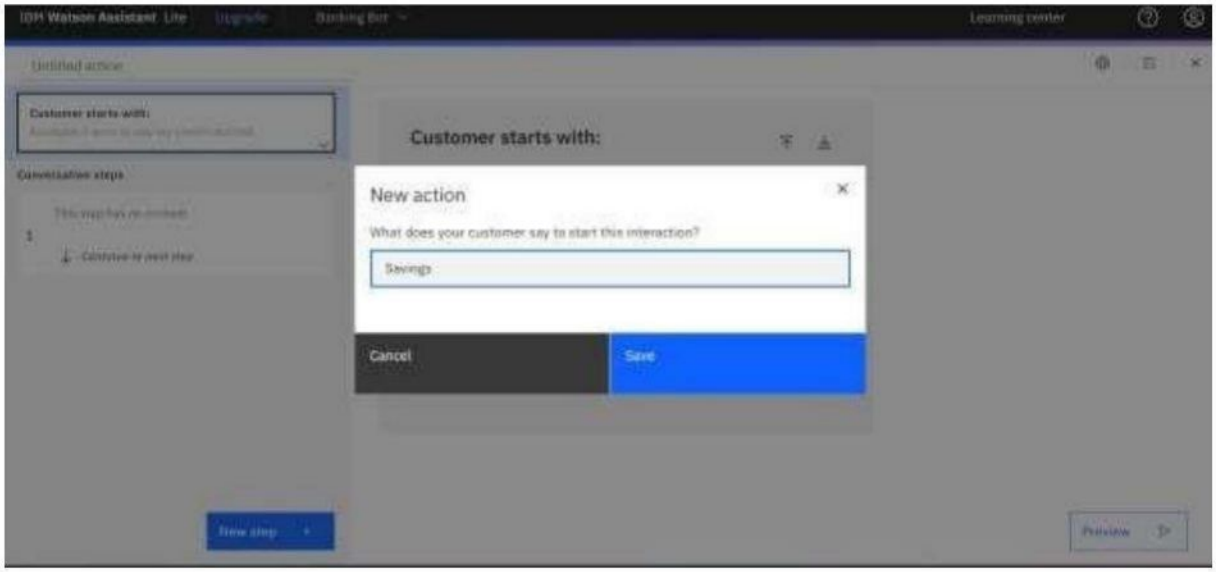
A default template chatbot is created. Need to add actions.



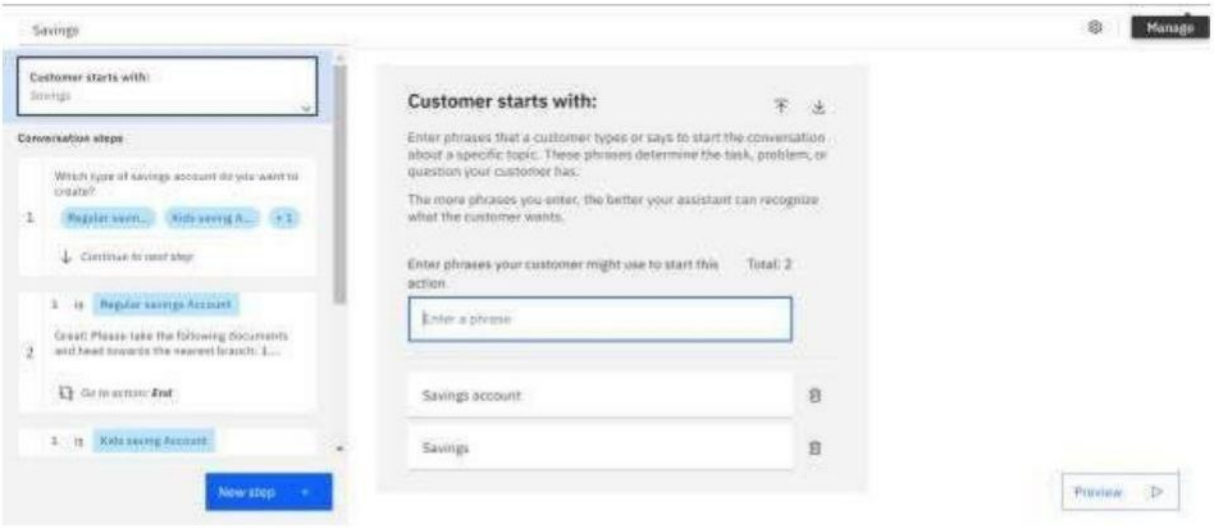


Creating Saving Account Action

Create a saving account in IBM Watson. Create new **Action** Saving.

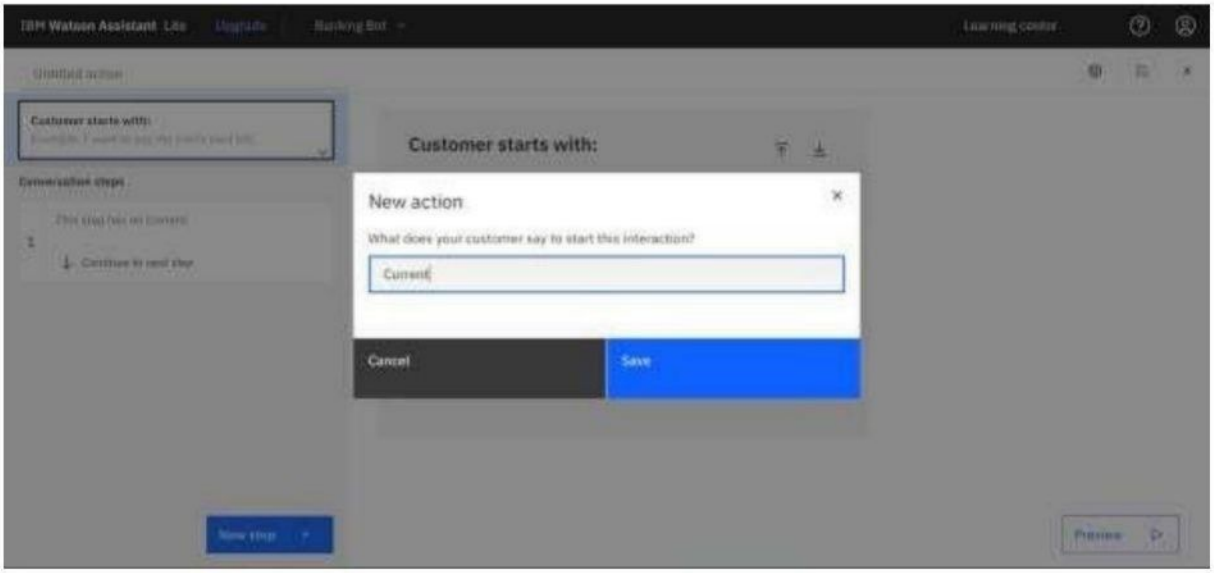


Add steps in savings action.

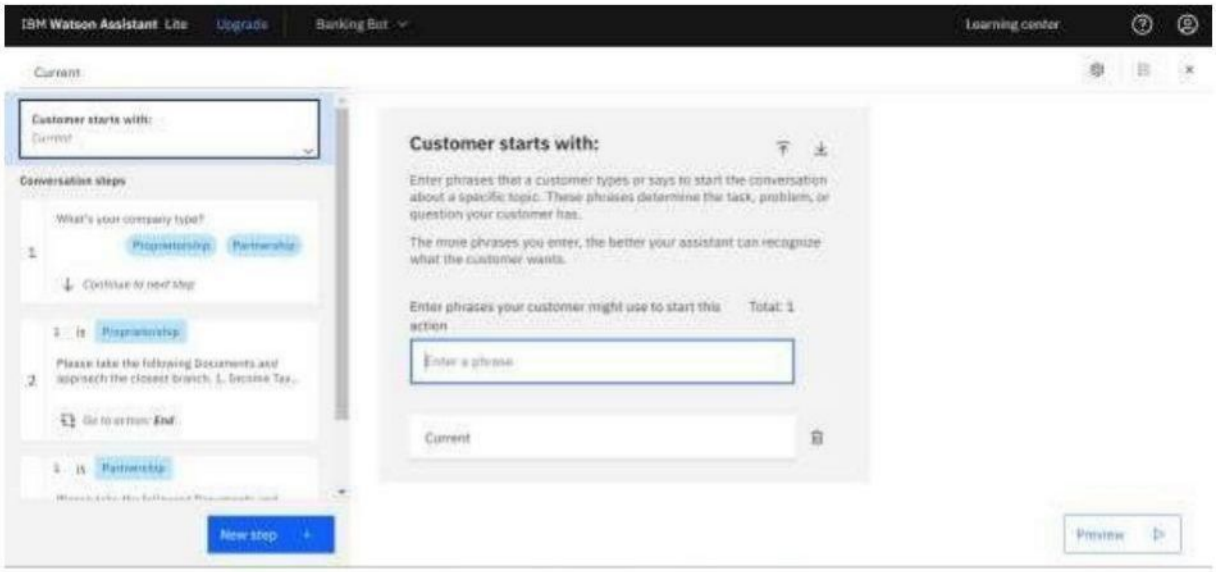


Creating Current Account Action

Create a new **Action** Current for the current account action.

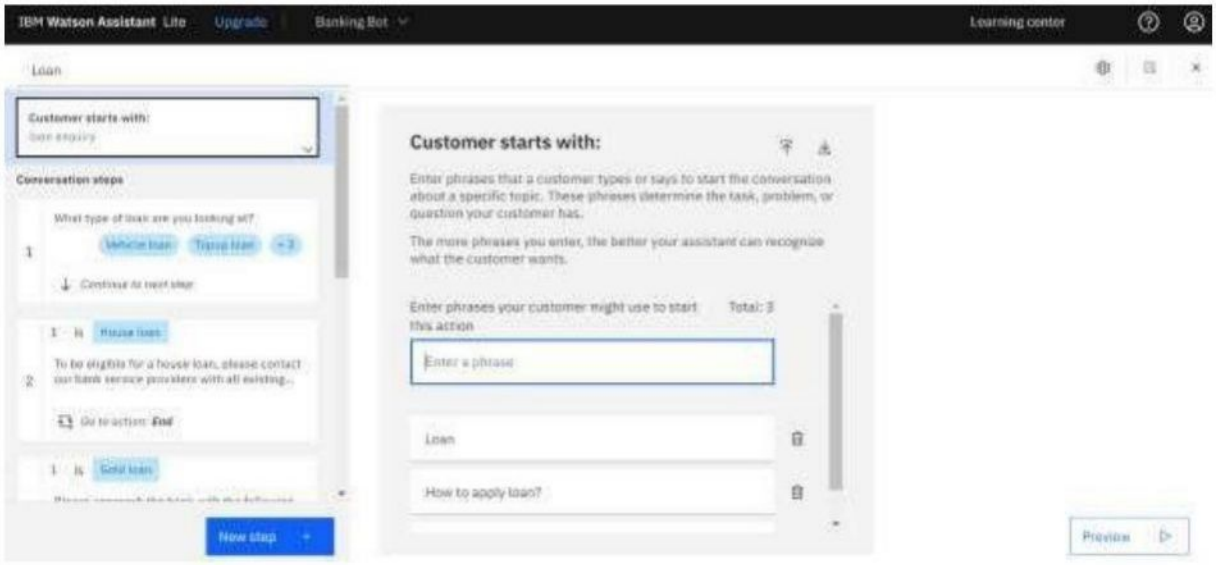


Add steps in current action.



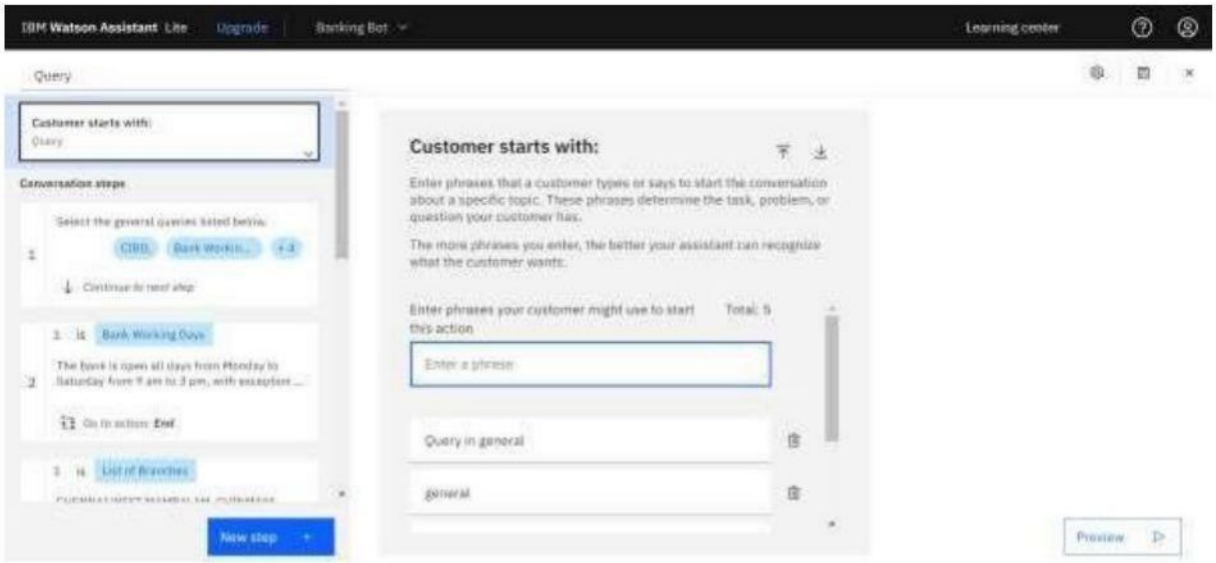
Creating Loan Account Action

Loan action is created with the necessary steps.



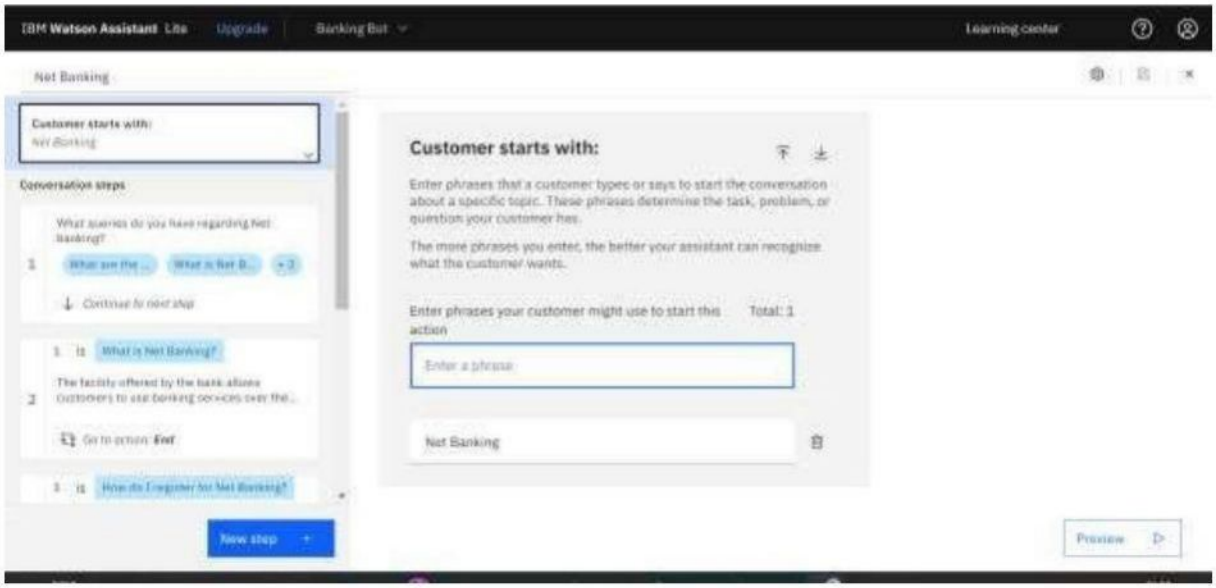
Creating General Query Action

General query action is created with the necessary steps.

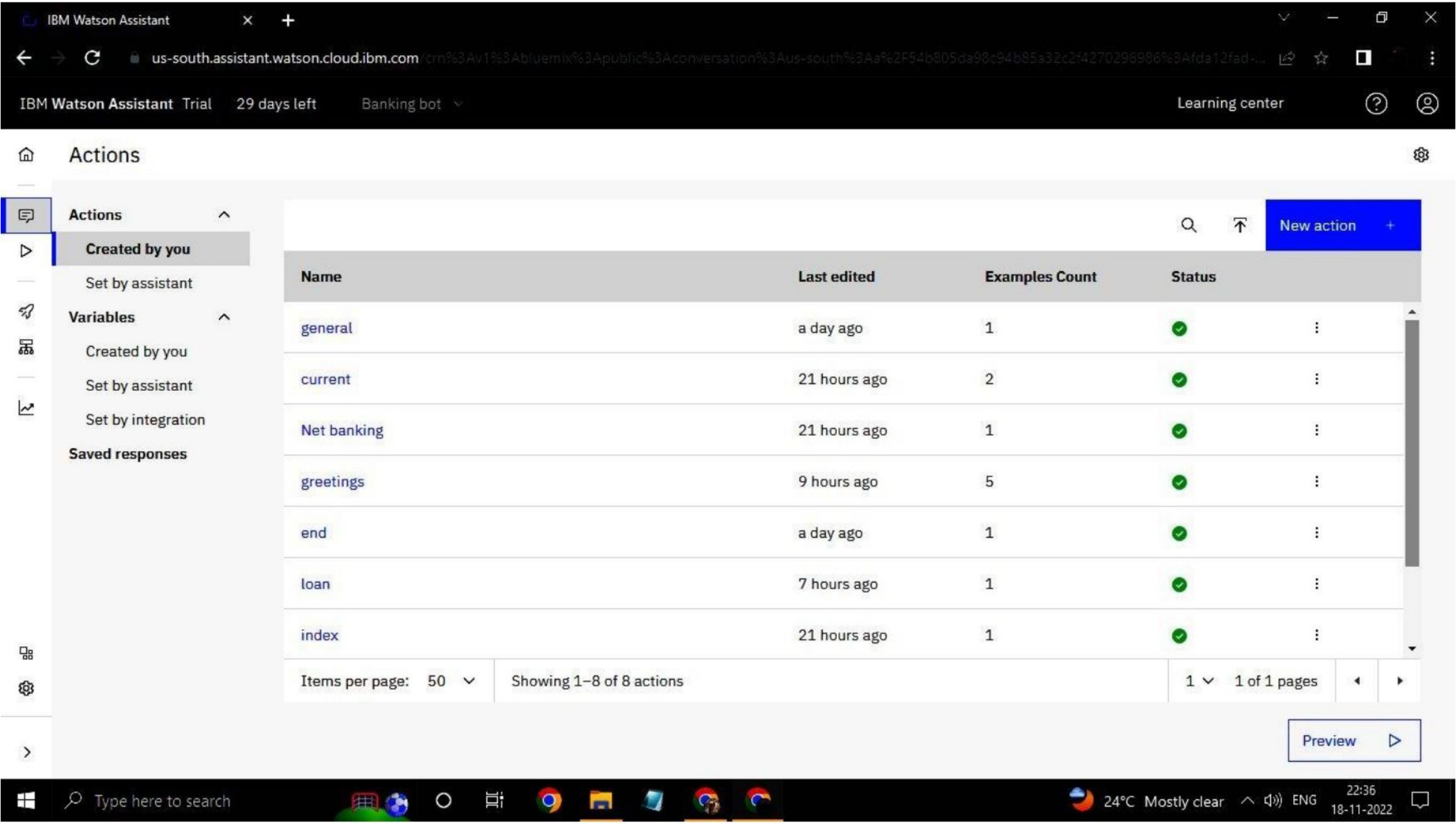


Creating Net Banking Action

Net banking action is created with the necessary steps.



In addition to this greeting, end greeting ,index and end actions are also created.


IBM Watson Assistant
us-south.assistant.watson.cloud.ibm.com
IBM Watson Assistant Trial
29 days left
Banking bot
Learning center
Actions
Actions
Created by you
Set by assistant
Variables
Created by you
Set by assistant
Set by integration
Saved responses
New action
Name
Last edited
Examples Count
Status
general
a day ago
1
current
21 hours ago
2
Net banking
21 hours ago
1
greetings
9 hours ago
5
end
a day ago
1
loan
7 hours ago
1
index
21 hours ago
1
Items per page: 50
Showing 1–8 of 8 actions
1
1 of 1 pages
Preview
Type here to search
24°C Mostly clear
ENG
22:36
18-11-2022

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.

Project report

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.

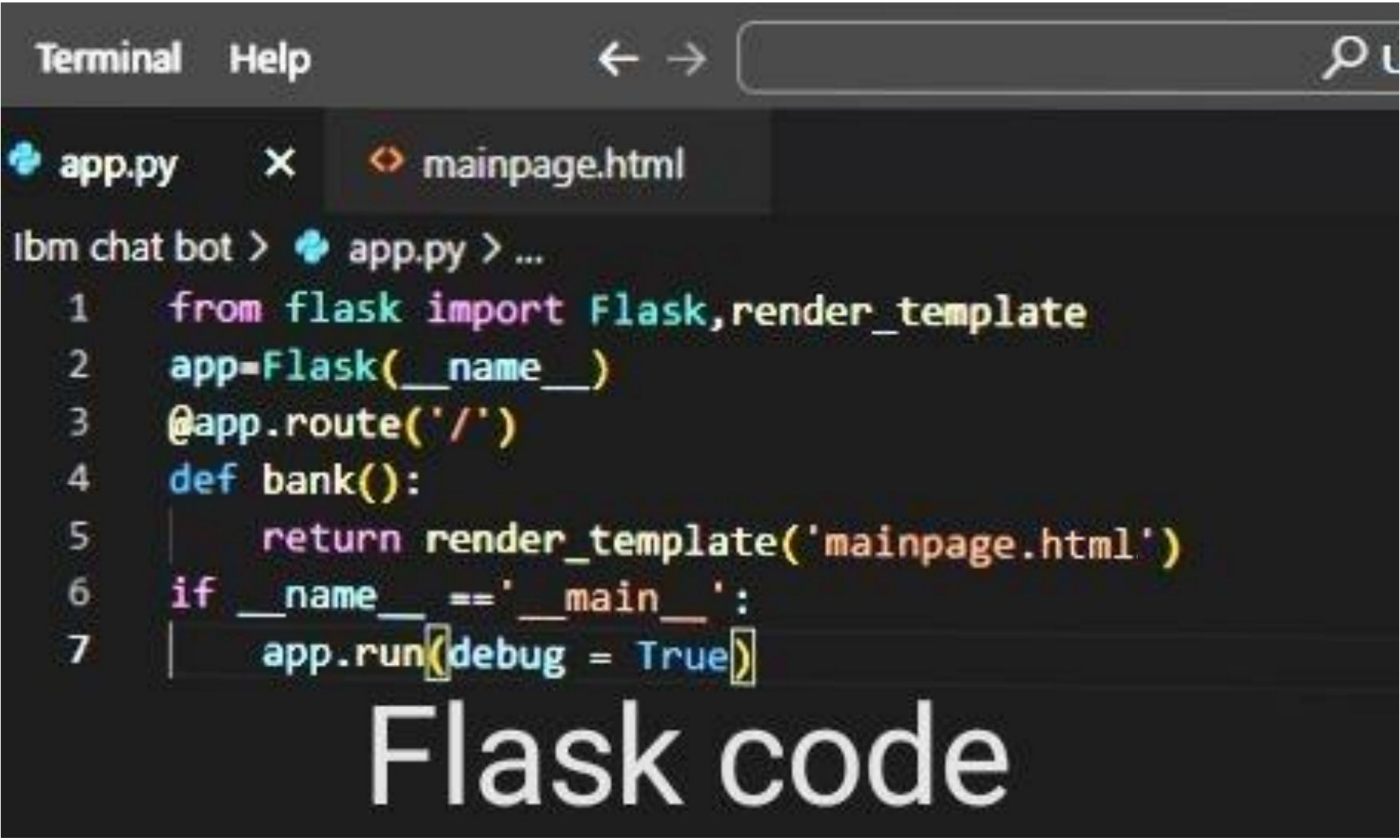
```
if __name__ == '__main__':
    app.run()
```

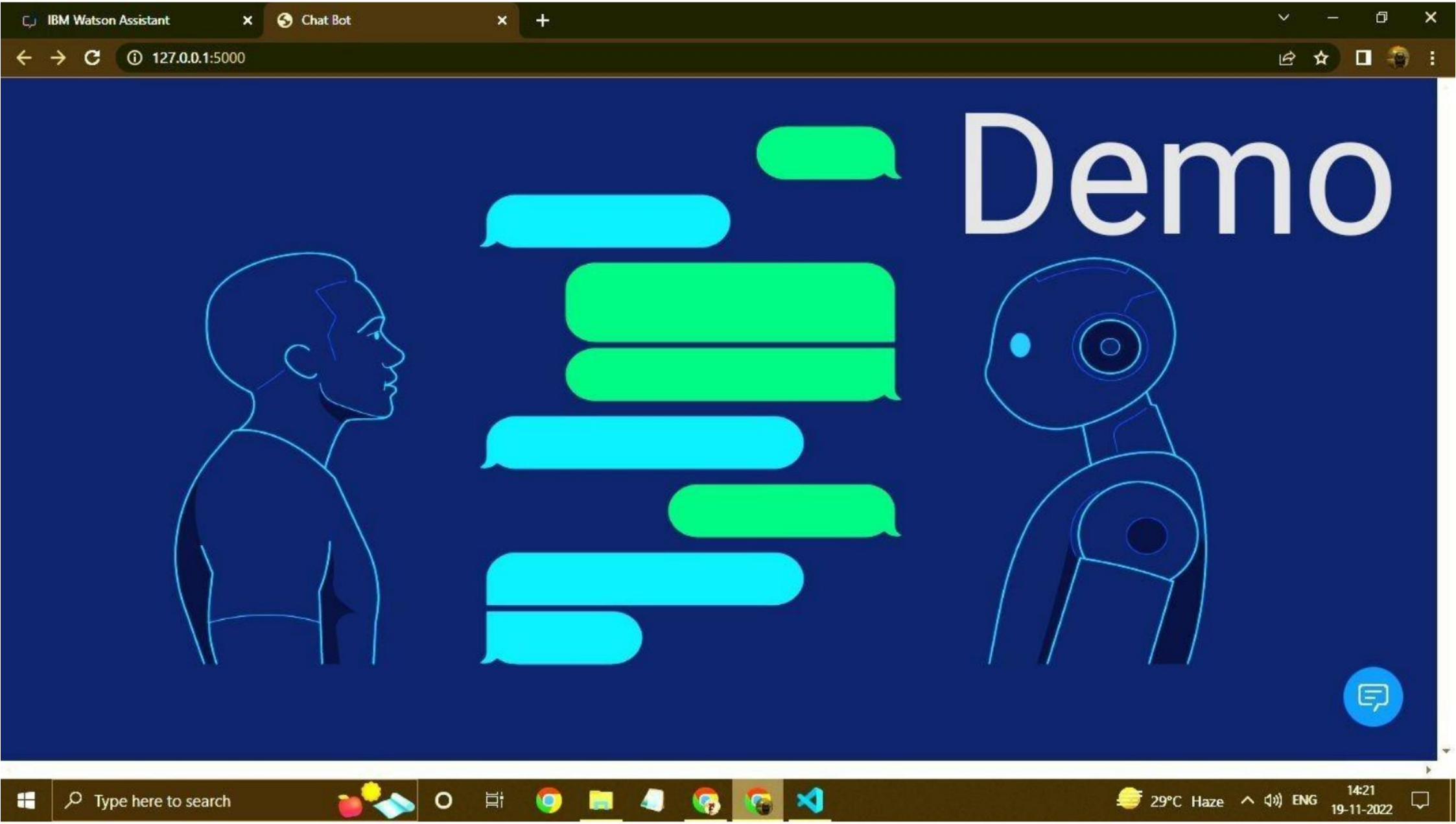
app.py mainpage.html

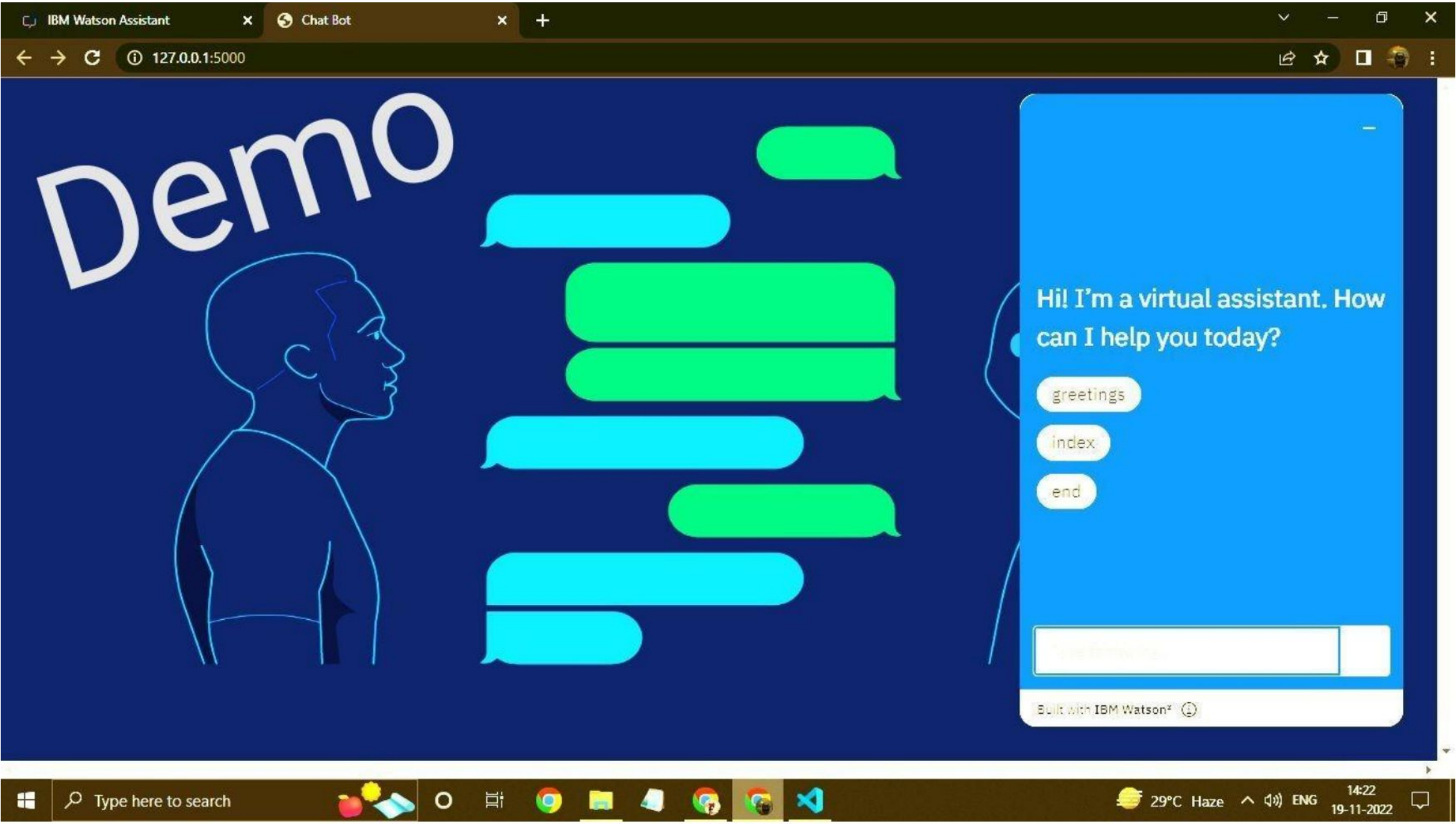
ibm chat bot > templates > mainpage.html > html > script

```
<html>

<head>
    <title>Chat Bot</title>
</head>

<script>
    window.watsonAssistantChatOptions = {
      integrationID: "821b10b0-73e7-4bae-b950-a2896ff892de", // The ID of this integration.
      region: "us-south", // The region your integration is hosted in.
      serviceInstanceID: "fda12fad-e9d7-4c46-b685-342c5dd73f3b", // 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') + "/WatsonAssista
      document.head.appendChild(t);
    });
</script>

<body>
  <img src="{{ url_for('static', filename='chatbotimg.jpg') }}" alt="chatbot image" style="position: relative;width: 1420px; height: 662px; top: -12px;left: -70px
</body>

</html>
```

HTML page code


Terminal Help
app.py
mainpage.html
Ibm chat bot > app.py > ...
1 from flask import Flask,render_template
2 app=Flask(__name__)
3 @app.route('/')
4 def bank():
5 return render_template('mainpage.html')
6 if __name__ =='__main__':
7 app.run(debug = True)
Flask code


IBM Watson Assistant
Chat Bot
127.0.0.1:5000
Demo
Type here to search
29°C Haze
ENG
14:21
19-11-2022


IBM Watson Assistant
Chat Bot
127.0.0.1:5000
Demo
Hi! I'm a virtual assistant. How can I help you today?
greetings
index
end
Built with IBM Watson
Type here to search
29°C Haze
ENG
14:22
19-11-2022

Preview Link

https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-fda12fad-e9d7-4c46-b685-342c5dd73f3b%3A%3A993ee716-488f-4609-9aa9-c811f8d20483&integrationID=821b10b0-73e7-4bae-b950-a2896ff892de&region=us-south&serviceInstanceID=fda12fad-e9d7-4c46-b685-342c5dd73f3b


Thank
You