**DOMAIN NAME: CLOUD APPLICATION DEVELOPMENT**

**PROJECT NAME: CHATBOT USING IBM CLOUD WATSON ASSISTANT**

**Phase 4 Submission Document**

To continue building the chatbot and integrate it with a messaging platform, you can use a web framework to expose the chatbot as a web service, which can then be accessed by the messaging platform. In this example, we'll use Flask to create a simple HTTP server for your chatbot**.**

**Install Flask:**



**Extend the Python script to include Flask and set up an HTTP server:**

import json

from ibm\_watson import AssistantV2

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

from flask import Flask, request, jsonify

app = Flask(\_\_name)

# Replace with your IBM Cloud Assistant API key and URL

api\_key = 'YOUR\_API\_KEY'

api\_url = 'YOUR\_API\_URL'

# Initialize Watson Assistant with your API key and URL

authenticator = IAMAuthenticator(api\_key)

assistant = AssistantV2(

version='2021-06-14',

authenticator=authenticator

)

assistant.set\_service\_url(api\_url)

# Create a session with your assistant

session = assistant.create\_session(

assistant\_id='YOUR\_ASSISTANT\_ID'

).get\_result()

session\_id = session['session\_id']

# Define a function to send a message to the assistant

def send\_message(message):

response = assistant.message(

assistant\_id='YOUR\_ASSISTANT\_ID',

session\_id=session\_id,

input={

'message\_type': 'text',

'text': message

}

).get\_result()

return response

@app.route('/chat', methods=['POST'])

def chat():

try:

user\_input = request.json['message']

response = send\_message(user\_input)

if 'output' in response:

bot\_response = ""

for text in response['output']['generic']:

if text['response\_type'] == 'text':

bot\_response += text['text'] + " "

return jsonify({"response": bot\_response.strip()})

else:

return jsonify({"response": "I'm sorry, I didn't understand your request."})

except Exception as e:

return jsonify({"error": str(e)})

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

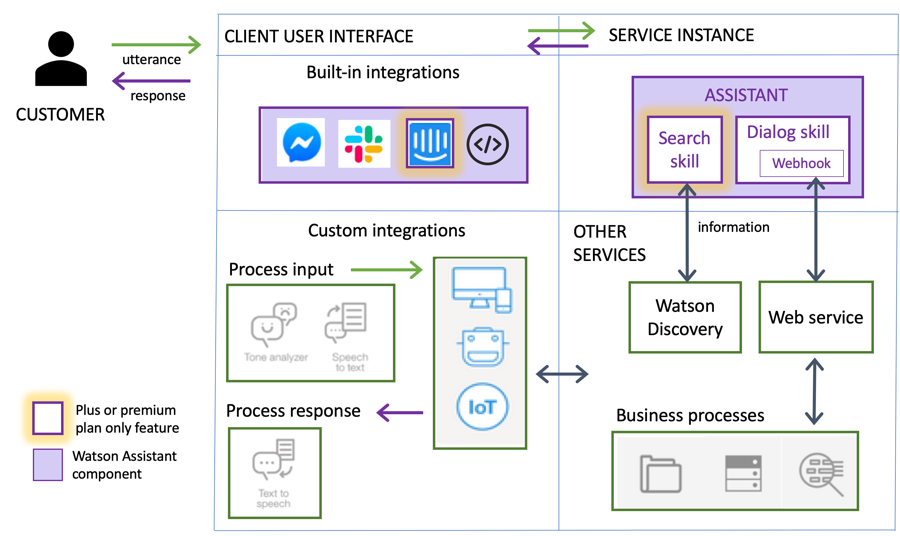
In this code:

* We create an **/chat** endpoint in Flask that accepts POST requests with JSON data. It expects a JSON object with a **message** field representing the user's input.
* The chatbot processes the user's input and returns a response.
* The chatbot's response is sent back as a JSON object.

Now, your chatbot is ready to be accessed over HTTP. You can use tools like ngrok to expose your local development server to the internet or deploy it to a web server. Then, you can integrate this chatbot with various messaging platforms (e.g., Facebook Messenger, Slack, or custom web chat interfaces) by sending POST requests to the **/chat** endpoint with user messages and receiving responses.

To refine the responses, you can improve your IBM Watson Assistant's dialog tree, add more intents and entities, and customize the responses to make the chatbot more conversational and context-aware.

**DESIGNING:**



**Steps to creating successful chatbots**

* Identify the problem to be solved and use case
* Choose the channel for your bot (e.g. Facebook Messenger, Skype, Slack)
* Choose the right use case for your bot
* Choose the services you will use to build your chatbot (in this case we are planning to use IBM Watson)
* Emulate conversations to train and retrain bot.
* Test
* Launch and learn

Included components:

* Build, test, and deploy a bot or virtual agent across mobile devices, messaging platforms, or even on a physical robot
* A cognitive search and content analytics engine for applications to identify patterns, trends, and actionable insights
* Analyze text to extract meta-data from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles, using natural language understanding
* Uses linguistic analysis to detect communication tones in written text
* An asynchronous event driven JavaScript runtime, designed to build scalable applications

IBM Watson has multiple tools to make the chatbot system works. One of the modules in Watson is Understand, which helps in understanding imaginary or unstructured data like humans understand. The reason is other tools that help in making form hypotheses and grasp underlying concepts. Learn is another module which helps in interactive learning so that new outcomes can be improved. Interact is the module helps in making chatbot interact like actual humans with abilities to talk, see, and hear.

**Conclusion**

In recent years, AI and machine learning have changed the way we go about our day-to-day business. Chatbots have conquered the market. No longer a nascent technology, chatbots are now mainstream. Its business impacts include breakthrough across different industries, such as financial services, retail, oilfield services, hospitals, and insurance companies.