

## **PROJECT OVERVIEW**

**Project Title:** Chatbot Deployment with IBM Cloud Watson Assistant

**Domain** : Cloud Application Development –Group 4

**Assignment** : Project submission phase 5

### **SUBMITTED BY**

**Name** : P.vigneshwaran

**Mail id** : vigneshwaranp971@gmail.com

**COLLEGE NAME: P. R Engineering College**

**College code** : 8212

**Group 4** : Zone (13-16)

# CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT

## Problem Definition and Design Thinking

### Problem Statement:

Create a helpful virtual guide using IBM Cloud Watson Assistant. Customize the chatbot to assist users on popular messaging platforms like Facebook Messenger and Slack. Provide useful information, answer FAQs, and offer a friendly conversational experience. Empower users with quick access to information and create meaningful connections through your virtual guide!

In response to the growing demand for efficient and scalable customer support solutions, our organization has embarked on a project to deploy a chatbot that will serve as a valuable addition to our customer engagement strategy. The objective of this project is to design, develop, and deploy a highly functional and user-friendly chatbot that can assist our customers, answer inquiries, and provide support across various digital platforms.

### Problem Definition:

Chatbot deployment refers to the process of making a chatbot, also known as a conversational agent or virtual assistant, available for use by end-users or customers. It involves taking a chatbot from the development and testing phase and making it accessible through various communication channels or platforms, such as websites, mobile apps, messaging apps, or voice assistants.

The project involves creating a chatbot using IBM Cloud Watson Assistant. The goal is to develop a virtual guide that assists users on messaging platforms like Facebook Messenger and Slack. The chatbot should provide helpful information, answer frequently asked questions (FAQs), and offer a friendly conversational experience. The project includes designing the chatbot's persona, configuring responses, integrating with messaging platforms, and ensuring a seamless user experience.

Chatbot deployment is a critical phase in the lifecycle of a chatbot project, as it marks the transition from development to real-world use. A well-executed deployment ensures that the chatbot delivers the intended benefits, such as improved customer support, automation of routine tasks,

and enhanced user experiences. It also involves ongoing monitoring and maintenance to keep the chatbot performing optimally and meeting user expectations.

## **Design Thinking:**

Design thinking is a user-centered approach to problem-solving and innovation. When applying design thinking principles to chatbot deployment, the focus is on creating a chatbot that not only functions well but also meets the needs and expectations of its users. Here's a design thinking framework for chatbot deployment:

### **1. Empathize: Understand User Needs and Goals**

- Conduct user research to gain insights into the needs, behaviors, and pain points of potential chatbot users.
- Create user personas and customer journey maps to visualize the user experience.
- Gather feedback from existing customer support channels to identify common issues and questions.

### **2. Define: Frame the Problem and Set Objectives**

- Clearly define the problem you aim to solve with the chatbot. What specific user needs or pain points will it address?
- Set clear objectives and success criteria for the chatbot deployment. What are the key performance indicators (KPIs) for success?
- Identify user scenarios where the chatbot can add value, such as answering FAQs, providing product information, or troubleshooting common issues.

### **3. Ideate: Generate Creative Solutions**

- Brainstorm potential chatbot features and capabilities that align with user needs and business goals.

- Encourage cross-functional collaboration among designers, developers, subject matter experts, and customer support teams to generate ideas.
- Explore different conversational flows and dialogue design options.

#### **4. Prototype: Create a Chatbot Prototype**

- Develop a functional prototype of the chatbot to visualize how it will work and interact with users.
- Use prototyping tools to design the chatbot's user interface and conversation flows.
- Conduct usability testing with a small group of users to gather early feedback and identify areas for improvement.

#### **5. Test: Gather User Feedback**

- Launch a pilot version of the chatbot to a select group of users or within a controlled environment.
- Collect feedback from users about their experiences with the chatbot.
- Analyze user interactions, conversation logs, and user satisfaction to refine the chatbot's design and functionality.

#### **6. Implement: Develop and Deploy the Chatbot**

- Based on the feedback and insights gathered during testing, refine the chatbot's design, dialogues, and user interface.
- Collaborate with developers to build and configure the chatbot on the chosen deployment platform (e.g., website, messaging app, or voice assistant).
- Ensure that data security and privacy measures are in place, including compliance with relevant regulations.

#### **7. Iterate: Continuous Improvement**

- Monitor the chatbot's performance and gather ongoing user feedback after deployment.
- Use analytics to track KPIs and identify areas where the chatbot can be enhanced.
- Implement regular updates and improvements to the chatbot based on user insights and changing business needs.

## **8. Scale: Expand Usage and Reach**

- As the chatbot proves its value, consider expanding its availability to a broader user base and across additional platforms.
- Prepare for scalability by optimizing infrastructure and resources to handle increased user interactions.

By following this design thinking framework, we can create a chatbot deployment strategy that prioritizes user needs, aligns with business objectives, and continuously evolves to deliver a valuable and satisfying user experience.

# **INNOVATION DESIGN TO SOLVE THE PROBLEM**

To deploy a chatbot using IBM Cloud Watson Assistant with a focus on design and innovation to solve a specific problem, you'll want to follow a structured approach. Let's break down the process into detailed steps:

## **1. Problem Definition and Ideation**

- Define the problem you want to solve with your chatbot. For instance, let's say you want to create a chatbot that helps customers troubleshoot common tech issues.
- Identify the pain points, user needs, and goals.
- Brainstorm innovative features and solutions to address these needs.

## **2. User-Centric Design**

- Create user personas to understand your target audience.
- Map out user journeys and identify touchpoints where the chatbot can assist users.
- Design a conversational flow that addresses user queries and provides a seamless experience.
- Incorporate design thinking principles to ensure a user-centric approach to problem-solving.

## **3. Watson Assistant Configuration**

- Set up a Watson Assistant instance on IBM Cloud.
- Define intents (user queries) and entities (data points) that the chatbot should understand.
- Create a dialog tree that guides the chatbot's responses based on user inputs.
- Train the chatbot using sample conversations and refine its understanding over time.

## **4: Innovative Features**

- Consider adding innovative features to your chatbot. For tech troubleshooting, this might include:

- Visual Recognition: Enable the chatbot to analyze user-submitted images to diagnose issues.
- Voice Commands: Implement voice-based interactions for users who prefer not to type.
- Integration with IoT Devices: Connect the chatbot to IoT devices for remote troubleshooting.

## **5. Testing and User Feedback**

- Test your chatbot extensively to ensure it provides accurate and helpful responses.
- Collect user feedback through beta testing or pilot deployments to refine the chatbot's performance.

## **6: Integration**

- Integrate the chatbot into your website or app using the Watson Assistant web widget, API, or other suitable methods.
- Ensure a seamless user experience by matching the chatbot's appearance and behaviour with your platform's design.

## **7. Data Security and Compliance**

- Address data security and compliance concerns, especially if you're handling sensitive tech-related data. Ensure your chatbot follows industry standards and regulations.

## **8. Scalability and Performance**

- Optimise the chatbot's performance for scale. IBM Cloud offers scalability options to handle increased traffic.

## **9. Analytics and Continuous Improvement**

- Implement analytics to monitor user interactions and chatbot performance.

- Continuously improve the chatbot by analysing user data, identifying pain points, and making adjustments to the dialog and features.

## **10. User Education and Support**

- Provide users with documentation and support channels to help them get the most from the chatbot.
- Offer guidance on how to interact with the chatbot effectively.

## **11. Innovation Iteration**

- Periodically revisit the chatbot's design and features to introduce innovative improvements that enhance the user experience and solve the problem more effectively.

## **12. Documentation and Knowledge Sharing**

- Document your design and development process, especially any innovative solutions you've implemented.
- Share your insights with the community to contribute to the field of chatbot development and AI innovation.

Throughout this process, collaboration and an innovative mindset are crucial. Seek input from various team members, including designers, developers, domain experts, and end-users, to ensure that your chatbot effectively addresses the problem and offers an innovative solution. Regularly review and iterate on your chatbot to stay at the forefront of technology and user needs.

## **DEVELOPMENT PART 1:**

In IBM Watson Assistant, which is a cloud-based conversational AI platform, entities, intents, and dialogs are key components used to build and train chatbots or virtual assistants. Here's a brief explanation of each:

### **Entities:**

- In Watson Assistant, an entity represents a specific piece of information within user input. It is used to extract relevant data from user messages. Entities can be things like dates, numbers, product names, or any other data you want to capture. You define entities to help the assistant understand and process user queries more effectively.
- For example, if you're building a chatbot for a restaurant, you might define an entity named "cuisine" to extract the type of cuisine the user is interested in (e.g., Italian, Chinese, Mexican).

### **Intents:**

- An intent is the purpose or goal expressed in a user's message. It represents what the user is trying to achieve or communicate. Intents are essential for routing user requests to the appropriate responses or actions. You define intents to help the assistant recognize and categorize user input accurately.
- For instance, in a virtual assistant for a bank, you might define intents like "Check Account Balance," "Transfer Funds," or "Report Lost Card" to identify the user's intentions.

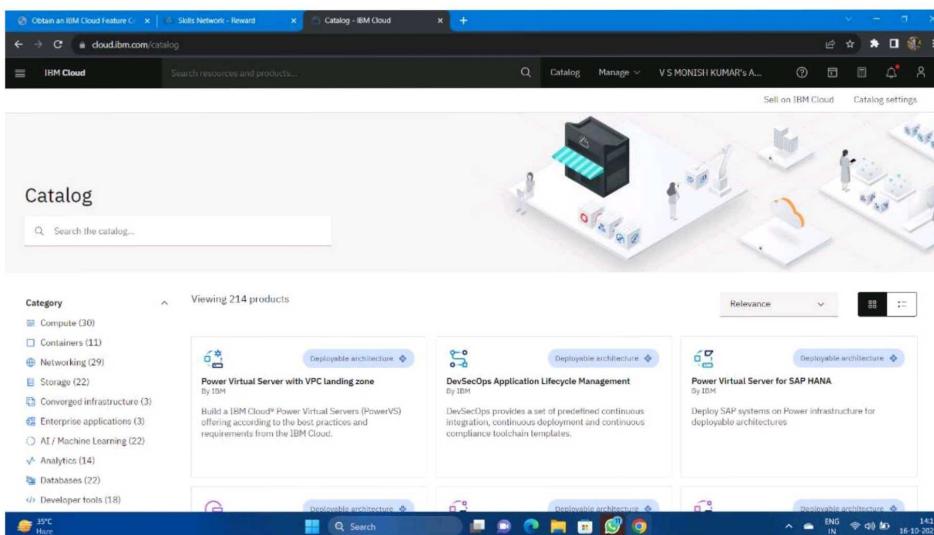
### **Dialogs:**

- Dialogs in Watson Assistant are used to structure the conversation flow between the user and the chatbot. You create dialog nodes to define how the assistant should respond to user input based on detected intents and entities. Dialogs help in creating dynamic and context-aware interactions.

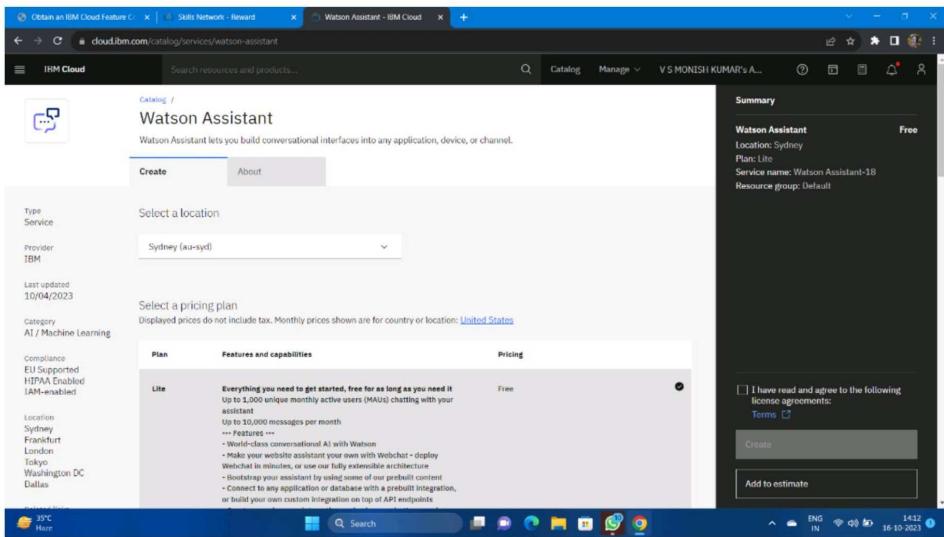
- Within a dialog node, you can define responses, conditions, and actions to take. You can also incorporate variables to store and retrieve information throughout the conversation, enabling personalized interactions.
- The typical workflow in Watson Assistant involves defining entities and intents, building dialog nodes to handle different conversation paths, and training the assistant using historical data or sample conversations. This training helps the assistant understand user input better, recognize intents and entities accurately, and respond appropriately.
- Entities, intents, and dialogs work together to enable natural and context-aware conversations between users and your chatbot or virtual assistant built with IBM Watson Assistant. By correctly defining and configuring these components, you can create effective and intelligent conversational interfaces.
- Now we are going to create the chatbot for that we will do the primary steps now.

## STEP1:

- Login To The IBM account and click on the Catalog and then search for Watson Assistant and give enter.

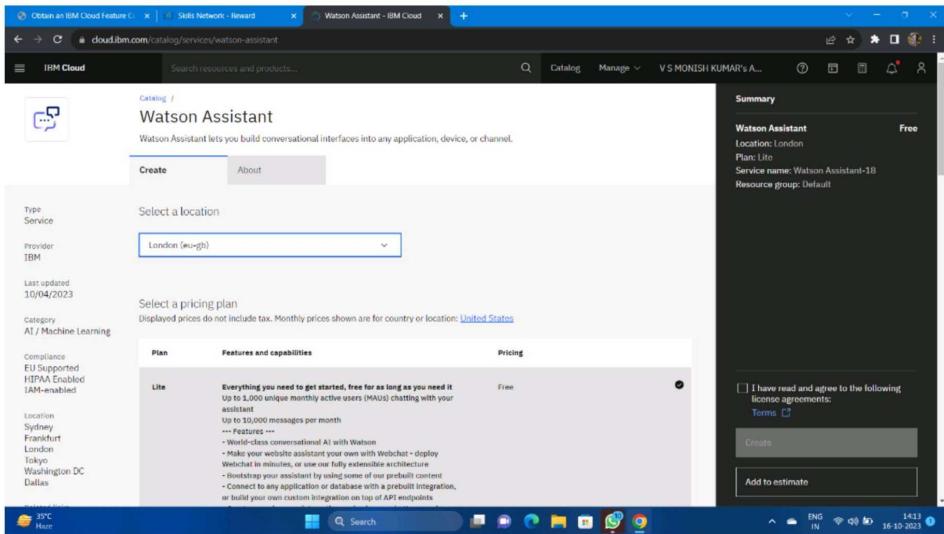


- You will get the Watson Assistant There By default you will have this

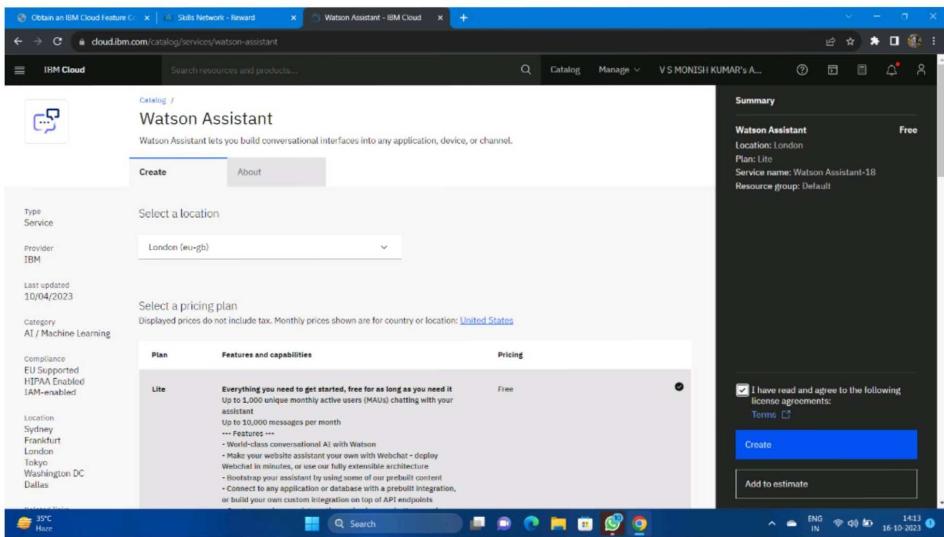


## Step 2:

- Change the default location and give the location as London(eu-gb) and select the plan as Lite

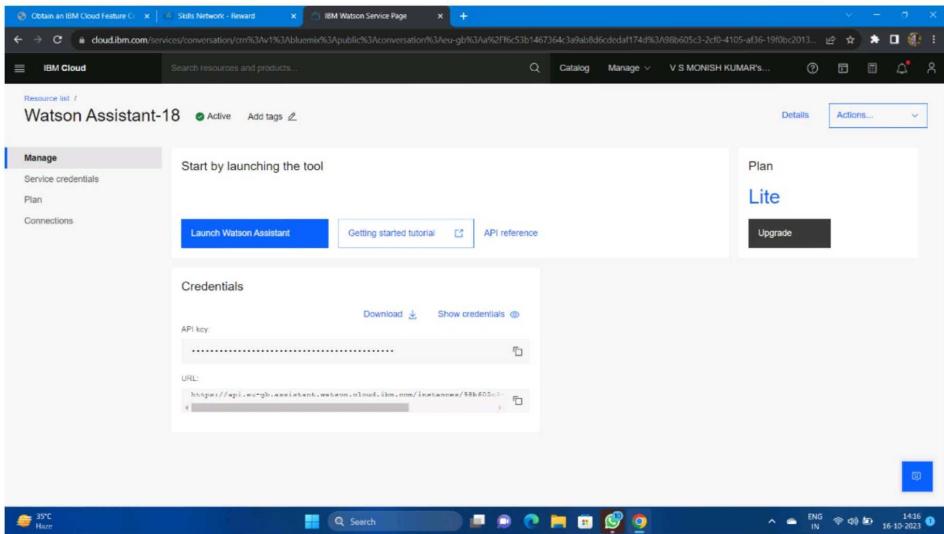


- Give tick mark for I Have read and agree to the following license agreement
- Now click on create it will create an instance for you



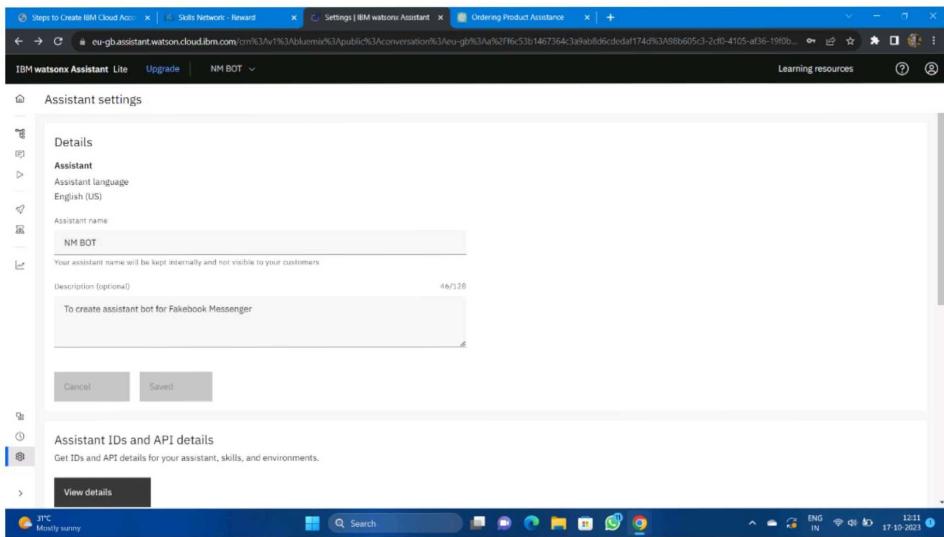
### Step 3:

- After creating an instance for Watson Assistant you need to launch the Watson Assistant by clicking the launch the assistant

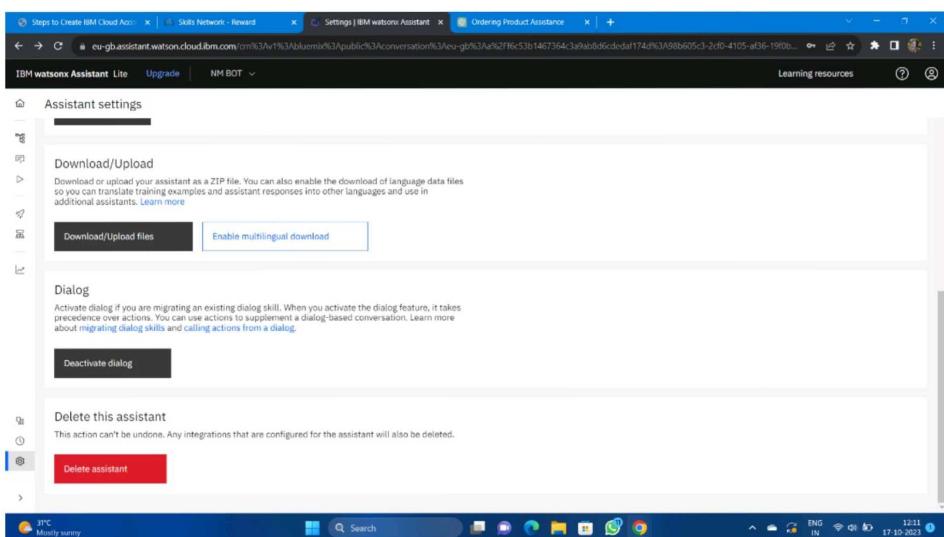


### Step 4:

- It will give the access to create the assistant give the name for the Assistant and give the description for that assistant it's completely optional click on create and save it.
- Here I have been created NM BOT as my chat bot assistant name .



- Now scroll down and then activate the dialog



## Step 5:

- After activating the Dialog, you will get the Intents, Entities, Dialog, and Content catalog like shown below

The screenshot shows the IBM Watson Assistant Lite interface. The left sidebar has a tree view with 'Intents' selected, followed by 'Entities', 'Dialog' (which is expanded), 'Options', 'Webhooks', 'Disambiguation', 'Autocorrection', 'Algorithm Version', 'Upload / Download', and 'Content Catalog'. Under 'Dialog', there's a sub-menu with 'My Entities' and 'System Entities'. The main content area has a heading 'What is an intent?' with a brief description: 'An intent is a collection of user statements that have the same meaning. By creating intents, you train your assistant to understand the variety of ways users express a goal.' Below this are buttons for 'Create intent' and 'Upload intents'. At the bottom of the page, there's a footer with the URL 'https://eu-gb-assistant.watson.cloud.ibm.com/ai' and a status bar showing '17-10-2023'.

## Step 6:

- Create the Entities first and some variables for the entities you have been created.
- Here I have been created the Entity with the name Entertainment and added variables as channels and star with some variable value.

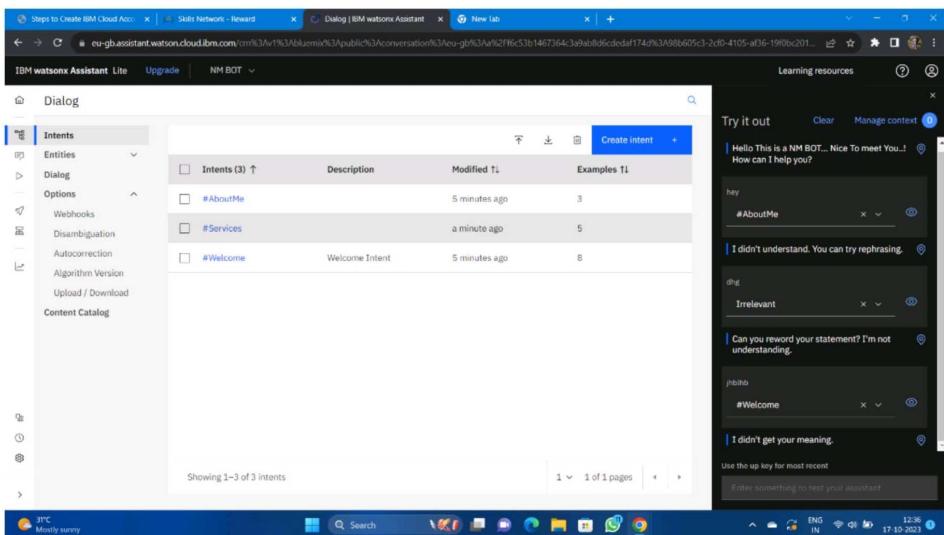
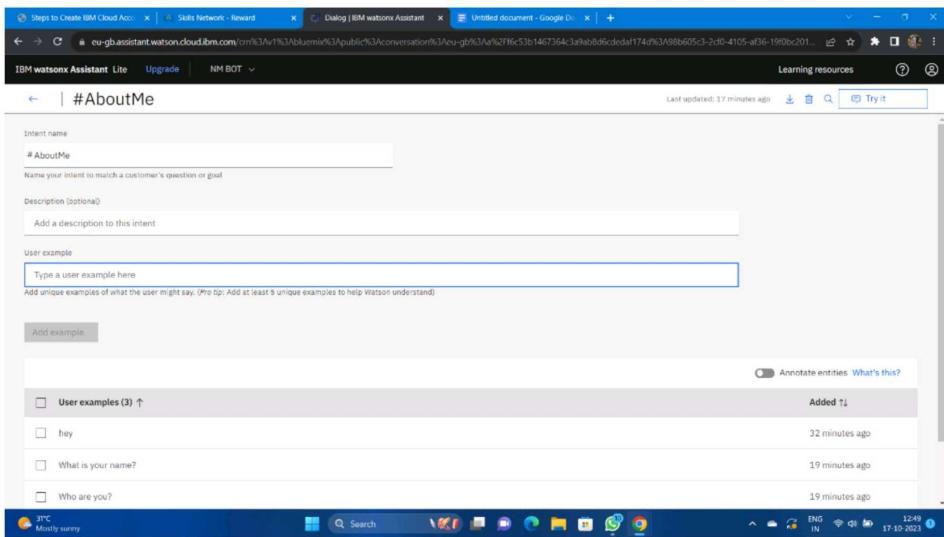
The screenshot shows the 'Entities' section of the IBM Watson Assistant Lite interface. A new entity named '@entertainment' has been created. The 'Value' field contains 'Type a value' and the 'Synonyms' field contains 'Type a synonym'. Below the input fields, there's a 'Dictionary (2)' tab which lists two values: 'channels' and 'star'. Both entries have their 'Type' listed as 'Synonyms' and provide examples like 'pages, accounts, logins' for 'channels' and 'vijay, maa, plus, cinema, sports, kids, music' for 'star'. The status bar at the bottom shows '17-10-2023'.

## Step 7:

- Open the Intents and then create the Intents for Messages, Services, AboutMe give some example queries for them

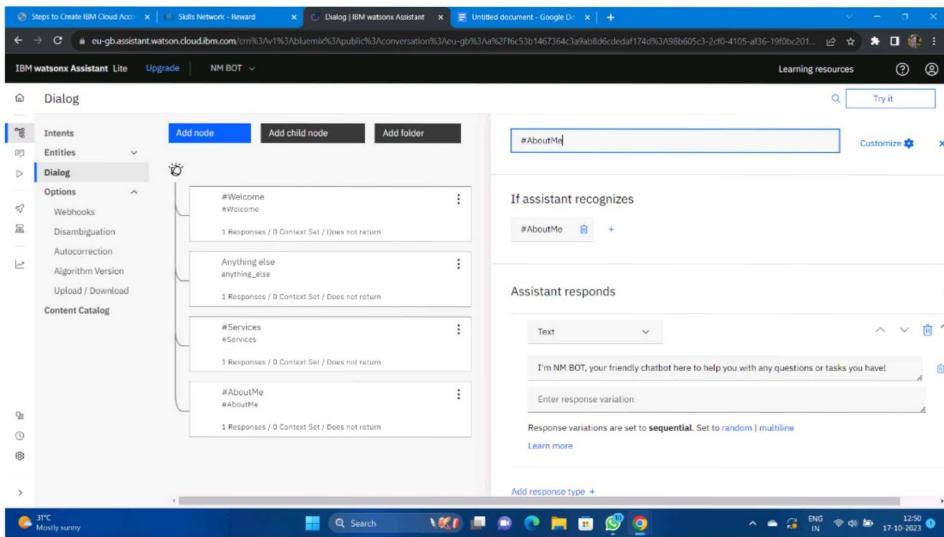
The screenshot shows the IBM Watson Assistant Lite interface. A modal window is open, confirming the successful creation of an intent named '#Messages'. The main interface shows fields for Intent name (#Messages), Description (optional), and User example (Type a user example here). Below these fields, a message says 'No examples yet.' and provides a note: 'Train your virtual assistant with this intent by adding unique examples of what your users would say.'

The screenshot shows the IBM Watson Assistant Lite interface. A modal window is open, confirming the successful creation of an intent named '#Services'. The main interface shows a list of user examples under the heading 'User examples (5) ↑'. Examples include: 'Can you tell me about your latest promotions?', 'How do I contact customer support?', 'How do I reset my password?', 'What are your business hours?', and 'What services do you offer?'. To the right, a 'Try it out' panel shows a conversation with the bot, with messages like 'Hello This is a NM BOT... Nice To meet You...', 'I didn't understand. You can try rephrasing.', 'Irrelevant', and 'Can you reword your statement? I'm not understanding.'



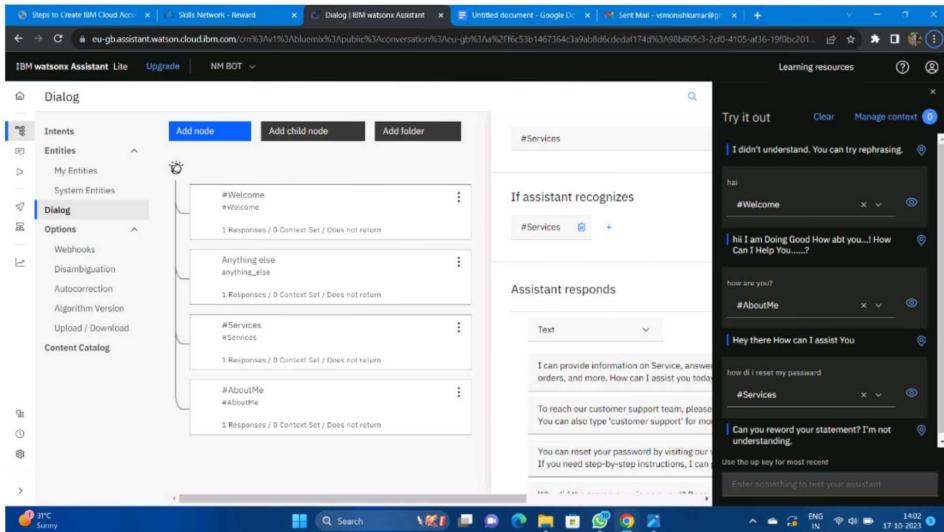
## Step 8:

- Next open the Dialog and then add nodes for all the Intents you have created where we need to give the responses for the selected queries.
- Whereby default we will have Anything else node.



## Step 9:

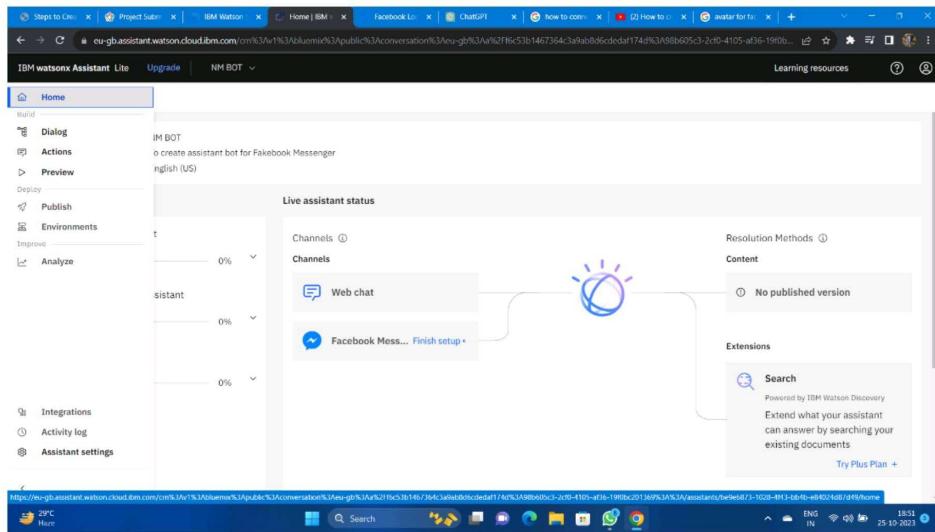
- Check the chat bot by clicking the try it before connecting the Facebook Messenger.



## DEVELOPMENT PART 2:

Integrating the chatbot with Facebook Messenger and Slack using respective APIs.

Initially go to IBM and login and then go to the already existing Watson instance and then open it and launch the assistance. It shows already existence bot that we created in the before phase open the home page in it and then start connecting the channels.



### Step 1: Create a Facebook Page

Log in to your Facebook account.

Go to Facebook's Create a Page and follow the steps to create a Facebook Page. This Page will be used to host your chatbot.

### Step 2: Set Up a Facebook App

Go to the Facebook for Developers website.

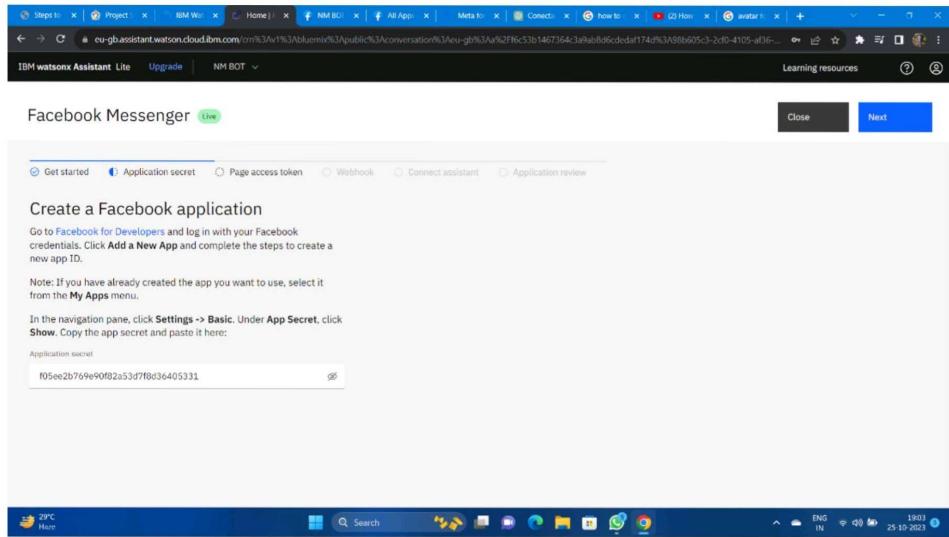
Create a new app by clicking on "My Apps" and then "Create App."

The screenshots illustrate the process of creating a Facebook app. The first step shows the 'Create an app' wizard, where the developer has chosen to allow users to log in with their Facebook accounts. This selection triggers a list of secondary use cases: Personalize your users' app experience by requesting access to their information, Get real-time notifications with Webhooks, and And more... Below this, there's a section for 'Use case requirements' which includes Business Verification, Annual data use checkup, Annual data protection assessment, and Data handling questions. The second screenshot shows the 'Basic' settings page for the newly created app, with the app ID 843882300620550. It displays fields for the app's basic information, including the app ID, app secret, display name (NM BOT), app domains, privacy policy URL, terms of service URL, and an uploaded app icon.

### Step 3: Configure the Facebook App for Messenger

In your Facebook App dashboard:

- Go to the "Messenger" section.
- Under "Access Tokens," generate a Page Access Token. You'll need this token to connect your Watson Assistant to Facebook Messenger.



#### Step 4: Create or Access IBM Watson Assistant

If you don't have an instance of IBM Watson Assistant, you can create one on the IBM Cloud.

#### Step 5: Configure Watson Assistant

In your IBM Watson Assistant instance:

- Create a new assistant or use an existing one.
- Configure your assistant by adding intents, entities, and dialog flows that suit your chatbot's purpose.

(Note: Step4 and Step5 already completed in phase 3 )

#### Step 6: Integrate Watson Assistant with Facebook Messenger

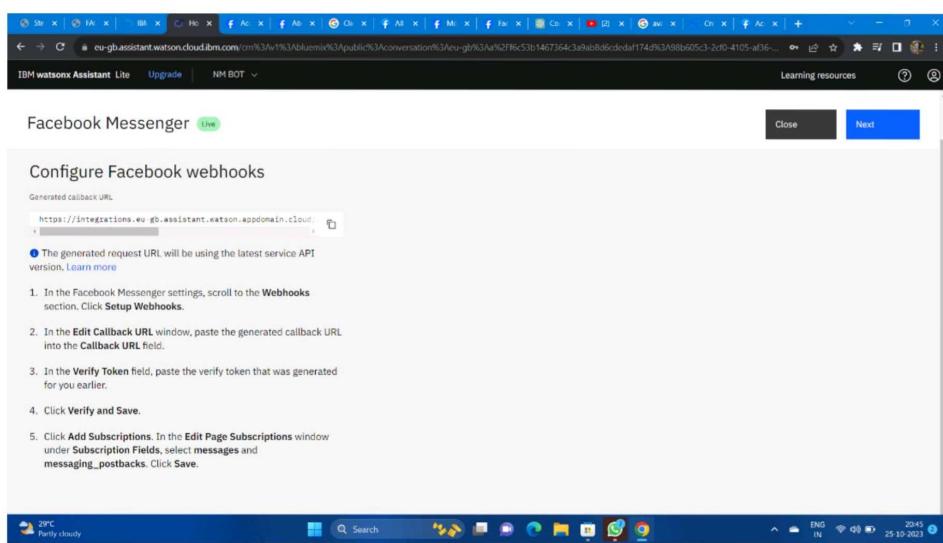
In the Watson Assistant Dashboard:

- Go to the "Skills" section.
- Add a new skill or use an existing one.
- In the skill settings, go to the "Integrations" tab.
- Click on "Facebook Messenger."

#### Step 7: Set Up the Webhook

In the Watson Assistant integration settings for Facebook Messenger:

- a. Enter the Facebook Page Access Token you obtained in Step 3.
- b. Specify the Webhook URL. This URL should be the same as the callback URL you set up in the Facebook App's Webhook settings.
- c. Save the changes.



#### Step 8: Subscribe to the Facebook Webhook

In the Facebook App settings:

- a. In the "Webhooks" section, click on "Edit Subscription."
- b. Select the Facebook Page you created in Step 1.
- c. Subscribe your Page to the webhooks events.

After completing all the steps complete it and click finish the connection now your bot is connected with the face book messenger



The implementation will work in the Facebook messenger where and which the queries that has been mentioned in the entities, dialogs and intents will be given as reply in my messenger.