**CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT**

**PROBLEM DEFINITON**

1. **Goal:** Use IBM Cloud Watson Assistant to deploy a fully working chatbot that can comprehend and reply to customer inquiries, giving correct information and improving user experience on a digital platform.
2. **Background:** Chatbots have evolved into a crucial tool for improving user experience on digital platforms, thanks to the quick development of artificial intelligence. Utilizing IBM Cloud Watson Assistant's powerful platform for creating, training, and deploying chatbots is essential for providing users with high-quality service.

**3. Important Requirements:**

a. Data Security: Ensure that all user data and communications with the chatbot are handled securely and in accordance with applicable data protection laws.

b. Custom Training: Train the chatbot using data that is pertinent to the industry, making sure it is aware of the particular subtleties and jargon used there.

c. Integration: Ensure minimal downtime and a seamless user experience by seamlessly integrating the chatbot with the digital platform (such as a website or mobile app).

d. Monitoring & Feedback Loop: Create a system to track the chatbot's performance and take user comments into account to increase its efficacy and accuracy.

e. Scalability: As the platform expands, make sure the chatbot solution can handle an increasing volume of user enquiries.

**4. Restrictions:**

a. amount: A little amount has been set out for the deployment and upkeep of the chatbot.

b. Timing: To support an impending marketing campaign or event, the chatbot must be implemented quickly.

c. Technical: Potential barriers to connecting the Watson Assistant with current IT systems or outside software.

**5. Expected Results:**

A useful chatbot with over 90% accuracy in understanding and responding to user inquiries.

b. A substantial decrease in manual customer service efforts as a result of the automation of frequent enquiries.

c. Increased user engagement and satisfaction with the digital platform.

**6. Evaluation Criteria:**

a. User Satisfaction Rate: The proportion of users who were pleased with the chatbot's responses.

The average amount of time it takes the chatbot to answer to a user enquiry.

c. Resolution Rate: The percentage of inquiries that the chatbot effectively answers without requiring human assistance.

d. Error Rate: The proportion of queries for which the chatbot gave incorrect or unnecessary answers.

**7. Future Aims:**

a. Expand the chatbot's language support capabilities in order to serve a global user base.

b. Integration with other IBM Cloud AI services for a more customized user experience, such as sentiment analysis.

c. Ongoing development and update of the chatbot's knowledge base to keep up with developments in the market.

**DESIGN THINKING**

**Integrating Chatbot with Facebook Messenger**:

1. **Facebook Page and App:**

* Create a Facebook Page for your chatbot/service.
* Create a new App at Facebook Developers Portal.

1. **Set up Messenger:**

* In the App’s dashboard, under “Add a Product”, select “Messenger”.
* Generate a Page Access Token by selecting your page. Save this token; you’ll need it to set up your bot server.

1. **Set Up Webhooks:**

* Facebook will send messages that are sent to your page to a webhook of your choice.
* Set up an endpoint on your server that listens for POST requests.

In the Facebook App dashboard, under “Webhooks”, set up a new webhook with the callback URL pointing to your server endpoint and subscribe to messaging events.

1. **Connect Your Bot:**

* Using the Page Access Token and the Facebook Graph API, set up your server to respond to the messages sent to your Facebook Page.

1. **Review:**

* Before your chatbot goes live, it will have to be reviewed by Facebook. Ensure that it adheres to Facebook’s platform policies.

**Design Persona**

**Name**: Eddie (derived from "EventBuddy")

**Communication Style:**

Casual: Uses a mix of youthful jargon and simple language that resonates with college students..

Informative: While maintaining a casual tone, Eddie ensures that the core information about events is conveyed clearly.

**Catchphrases:**

"Are you prepared to slay that campus event?"

"I've got the details, so don't miss out!"

"Campus vibes are building up!"

**User Scenarios for the EventBuddy Chatbot ("Eddie")**

**Freshmen introduction**

User: Requests information about important yearly events.

Eddie: Offers a calendar of major college events.

**Check the event timing**

User: Requests the beginning time of a particular event.

Eddie: Announces when the event will begin.

**Location of Event Enquiry**

User: Requires instructions or the location of a place for an event.

Eddie: Gives the venue information and provides a map.

**Registration for the event**

User: Desires to sign up for a certain workshop or event.

Eddie: Provides a link for registration.

**Reactions to Events**

User: Would like to comment on a recent event and Eddie: Asking the user

**User Experience Guidelines for Chatbots**

**On boarding:** Greet users and introduce the chatbot functions.

**Language:** Avoid using technical jargon and speak in a nice manner.

**Quick responses:** Provide a succinct response.

**Error Handling:** Acknowledge misunderstandings politely and provide solutions.

Offer alternatives or succinct responses to steer talks.

**Feedback:** Allow people to comment on the performance of the chatbot.

**Human Handoff:** When necessary, provide a transition to a human representative.

**Context:** Remember user preferences or interactions during a session.

**Continuous Learning:** Keep an eye on interactions and update frequently.

**Privacy:** Prioritize user data privacy and educate users about data handling.

**Control:** Make it simple for users to halt or resume talks.

### EventBuddy Chatbot ("Eddie") - Conversation Flow

**2. Flow of Event Inquiry**

Who: "What events are coming up?"

Eddie: "These upcoming events are listed in [List of Events]. Want to know more about any of these?

**1. Salutation**

Hey there, Eddie! You’re EventBuddy, Eddie here. Check out some college events now. 🎓" **Recognizing User Intent** what upcoming events are there?

"I'd like to sign up for TechWeek."

**4. Flow of Event Updates**

"Any changes to the Debate Championship?" asked the user.

Yes, it has been moved to next Friday, Eddie. Make a note on your calendars.

**3. Flow of Registration**

User: "I want to register for the TechWeek AI workshop."

Eddie: "Awesome choice! 🤖 Click here to register. Let me know if you need any other info."

**6. Concluding the Dialogue**

"Thanks, Eddie!"

Eddie: "Thank you so much! I'm available to help if you have any further inquiries. Enjoy the festivities.

**5. Flow of Feedback and Support**

The user says, "The registration link isn't working."

Eddie: "I apologize for it! I have reported it. Try this direct connection in the interim: [Alternative connection]."

**7. Handling Errors**

When Eddie fails to understand the user's request:

Eddie: "I'm sorry, I missed that. Can you clarify or inquire about a different event? To get help if you're stuck, type 'help'.

**Response Configuration for Eventbuddy in Watson Assistant**

**Intents**

* #ask\_event\_info
* #event\_registration
* #ask\_event\_location
* #feedback\_or\_issue

**Entities**

* @event\_name
* @event\_type

**Dialog Nodes**

* Node: Ask Event Info
* Node: Event Registration
* Node: Ask Event Location
* Node: Feedback or Issues

**Fallback Node**

* For unrecognized queries or errors:
* Response: "Sorry, I'm not sure about that. Can you rephrase? Or type 'help' to see what I can assist with."

**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 touch points 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 behavior 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**

* Optimize 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 analyzing 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.

**DEVELOPMENT PART 1**

**1. Set Up IBM Cloud Account:**

* Register or log in to your IBM Cloud account.
* Navigate to the dashboard.
* Click on **Create resource** and select **Watson Assistant**.

**2. Design Your Event Management Chatbot:**

* Launch the Watson Assistant and **create a new Assistant**.
* Name it something relevant, like **EventBot**.
* Add a dialog skill to define how the chatbot will converse.

**3. Define Intents:**

* **#inquire\_event\_details:** To ask about specific event details.
* **#book\_event\_ticket:** To book a ticket or register for an event.
* **#cancel\_registration:** To cancel a previously made booking.
* **#event\_feedback:** To provide feedback on an event.
* **#ask\_venue\_directions:** To inquire about venue directions and any other intent pertinent to event management.

**4. Define Entities:**

* **@event\_name:** Recognize and store the names of different events (e.g., "Summer Gala", "Tech Conference").
* **@date:** Identify specific dates.
* **@event\_type:** Different types of events like workshops, conferences, galas, etc.

**5. Craft Dialog Nodes:**

* **Welcome Node:** "Hello! Welcome to XYZ Event Management. How can I assist you with your event needs today?"
* **Event Inquiry:** If the intent detected is #inquire\_event\_details, the bot can respond with details about the event. If a @event\_name entity is detected, it can provide specific details about that event.
* **Booking:** For **#book\_event\_ticket**, guide the user through the booking process.
* **Cancellation:** For **#cancel\_registration**, ask for details like registration number and then process the cancellation and so on for each intent and its related entities.

**6. Incorporate Prompts:**

* Use clear prompts to guide users, especially for tasks like booking where a sequence of steps might be required.
* "Would you like to know more about the **@event\_name**?"
* "Please provide your preferred date for **@event\_name**."

**7. Integrate and Deploy:**

* Once you're satisfied with the bot's dialog flow and responses, you can integrate it with various channels where your audience might interact, such as a website, event app, or social media platform.
* Use the Watson Assistant's **Integrations tab** to help with deployment.

**8. Monitor and Refine:**

* Use Watson Assistant's analytics to monitor user interactions, recognized intents, and any questions the bot couldn't answer.
* Refine the bot over time based on feedback and interaction patterns.

**9. Feedback Loop:**

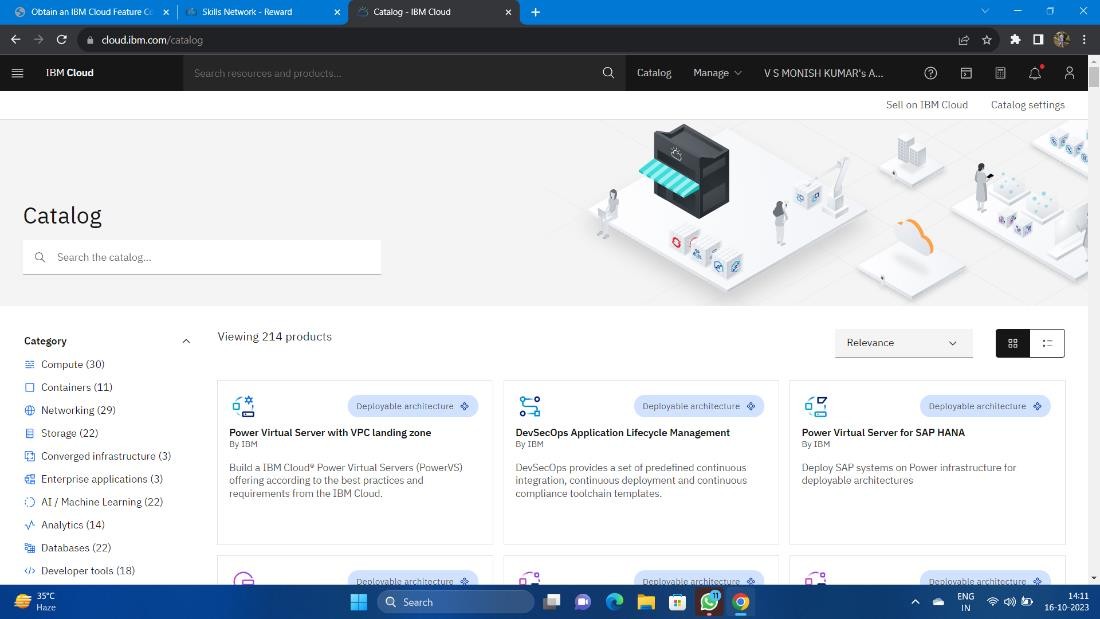
* Encourage users to provide feedback after attending events.
* This can help the event management team improve future events and offer a better experience.

**10. Additional Features:**

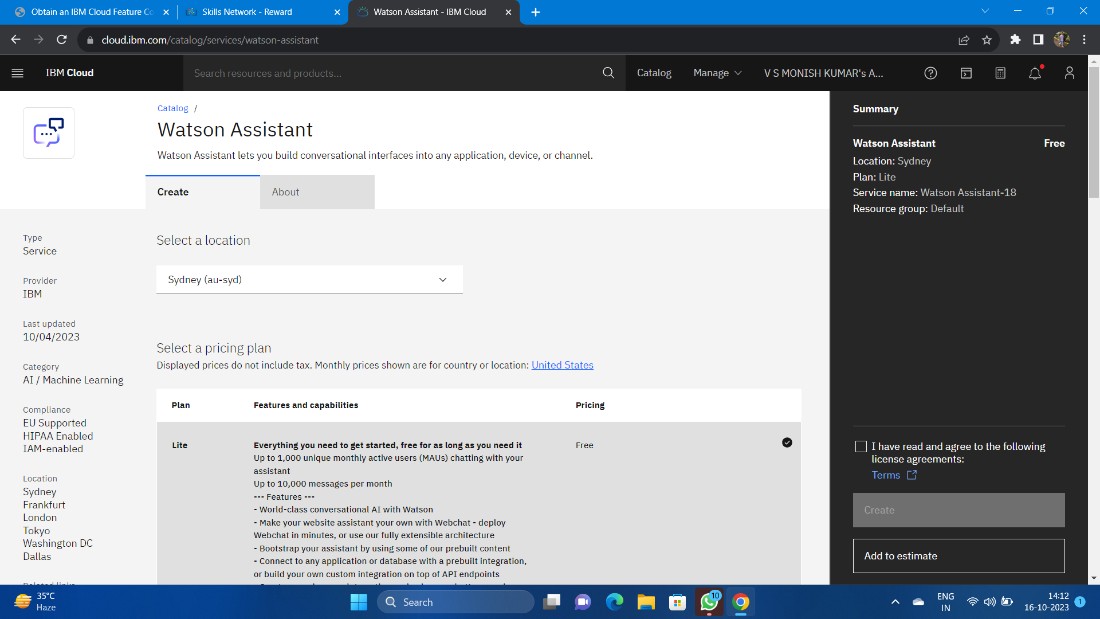
* **Reminders:** Integrate with email or SMS services to remind users of their booked events.
* **Suggest Events:** Based on user interactions, suggest upcoming events they might be interested in.

#### STEP1:

* Login to the IBM account and click on the Catalog and then search for Watson Assistant and give enter.

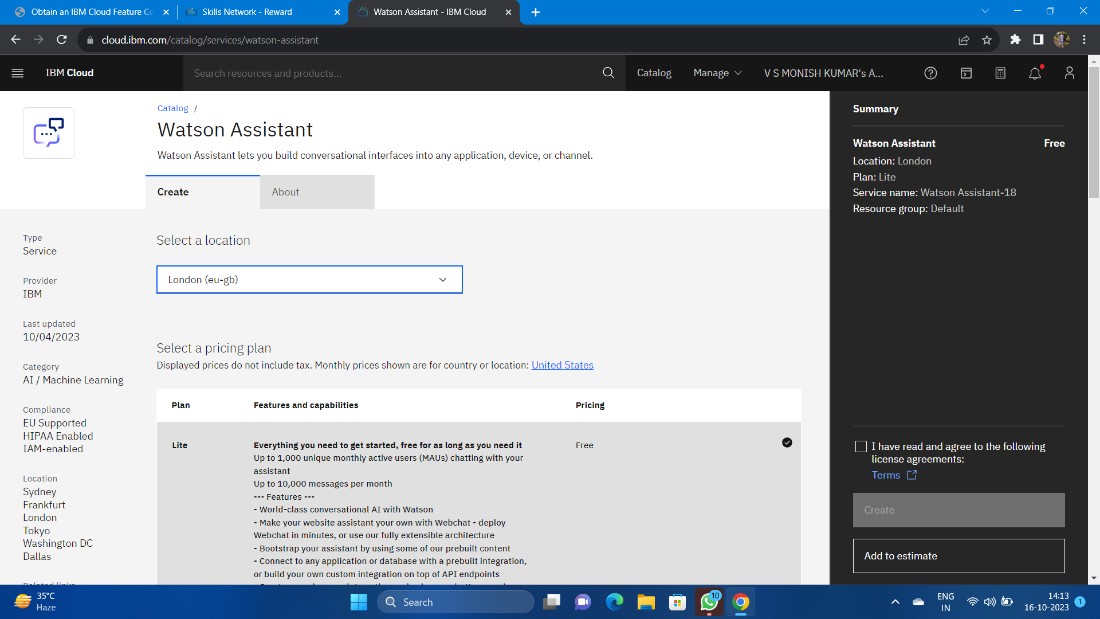


* You will get the Watson Assistant There By default you will have this

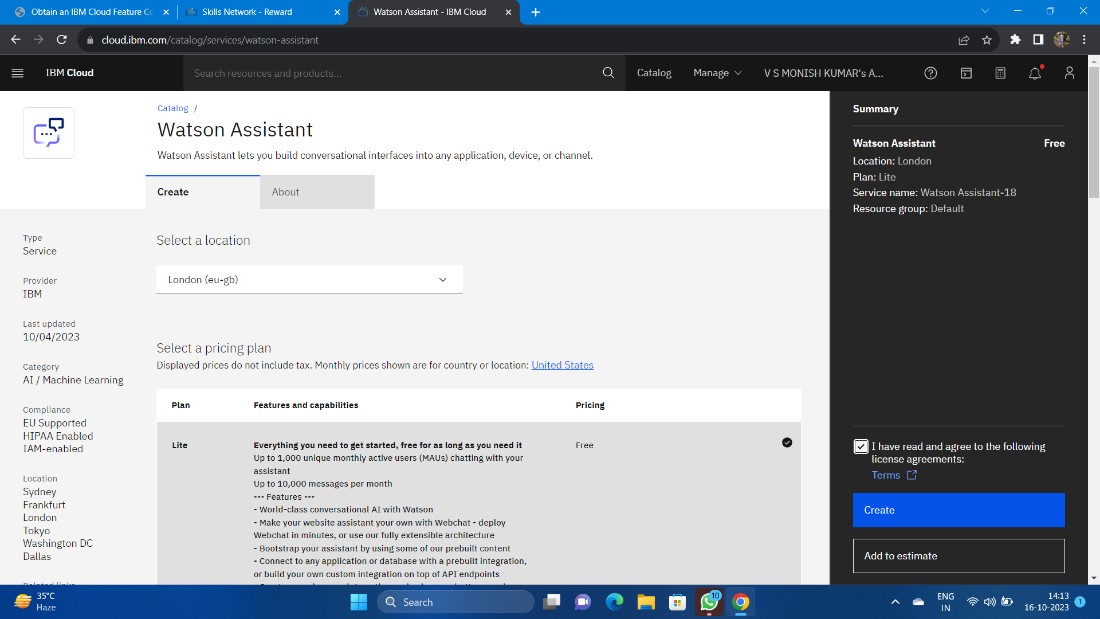


#### Step 2:

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

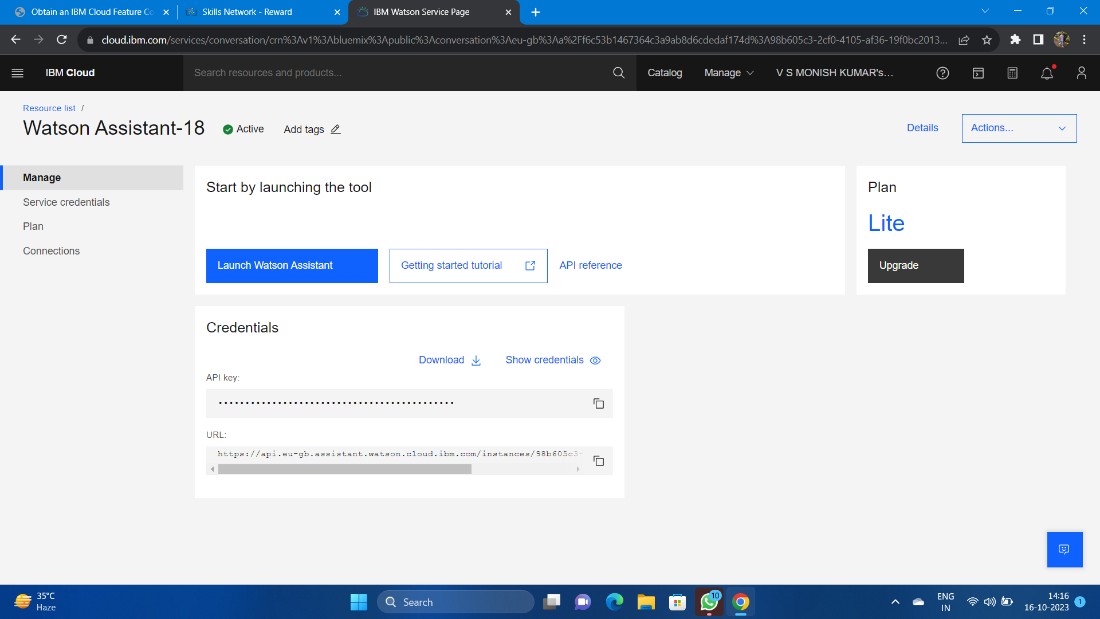


* Give tick mark for I Have read and agree to the following license agreement
* Now click on create it will create an instance for you



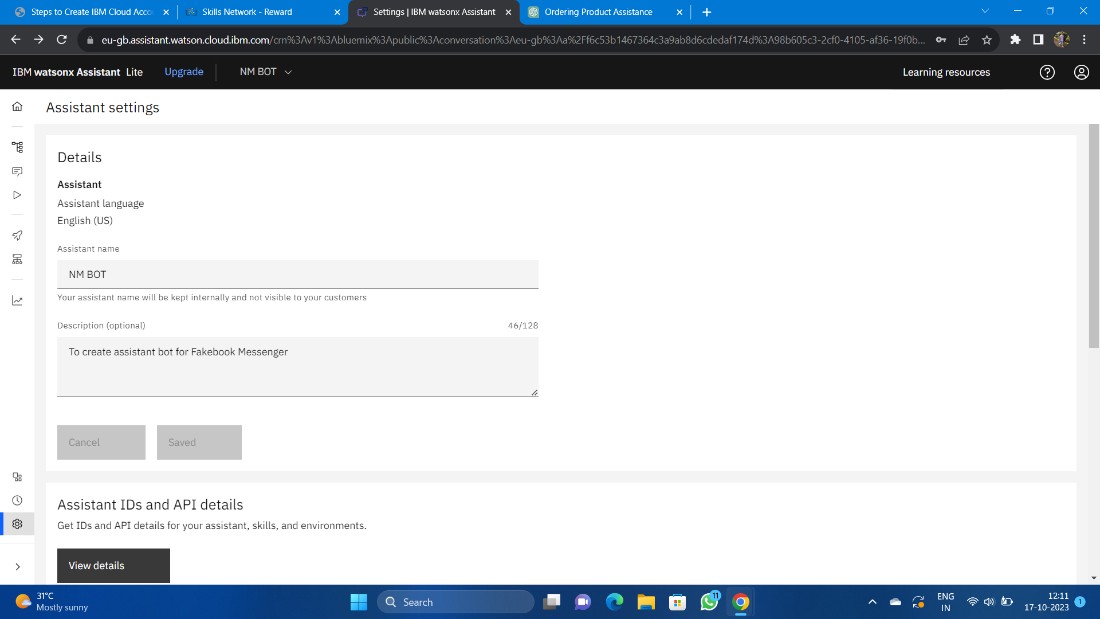
#### Step 3:

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

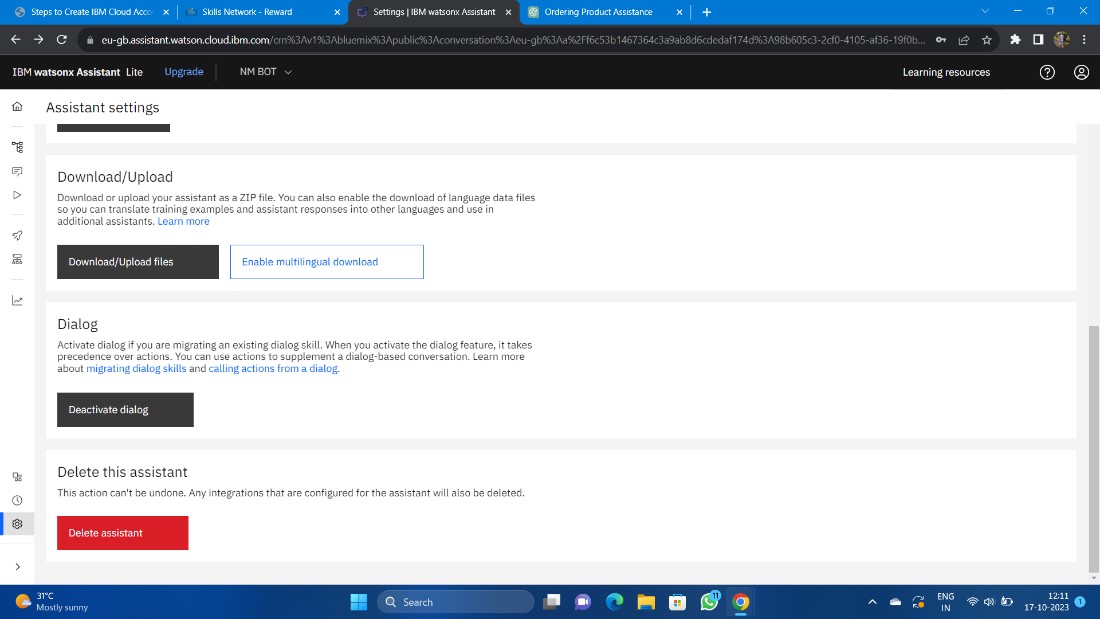


#### Step 4:

* 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.

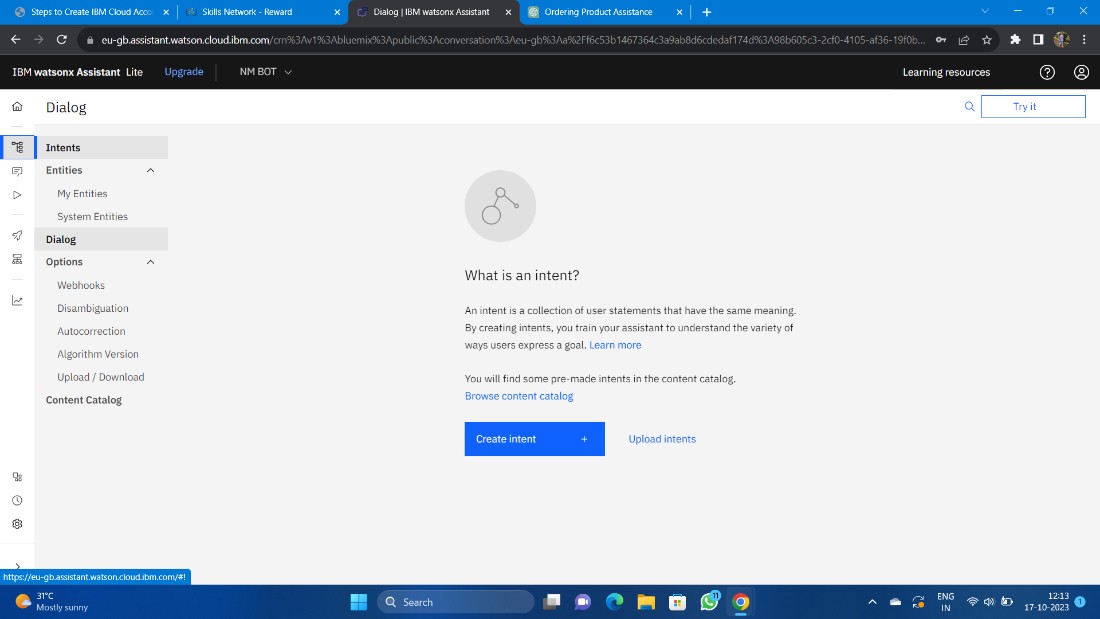


* Now scroll down and then activate the dialog

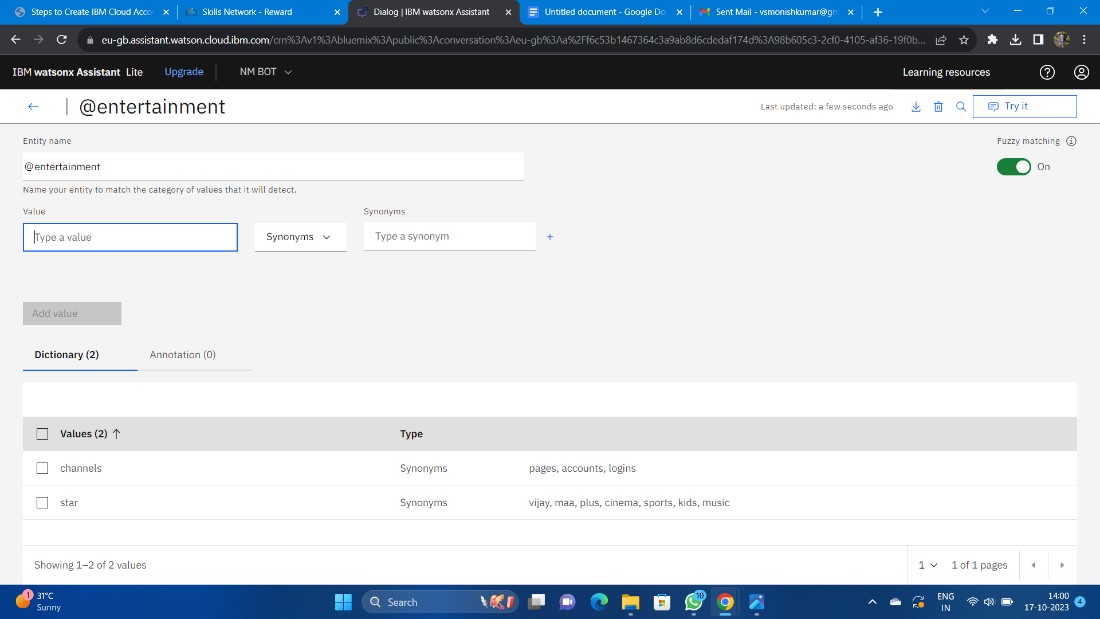


#### Step 5:

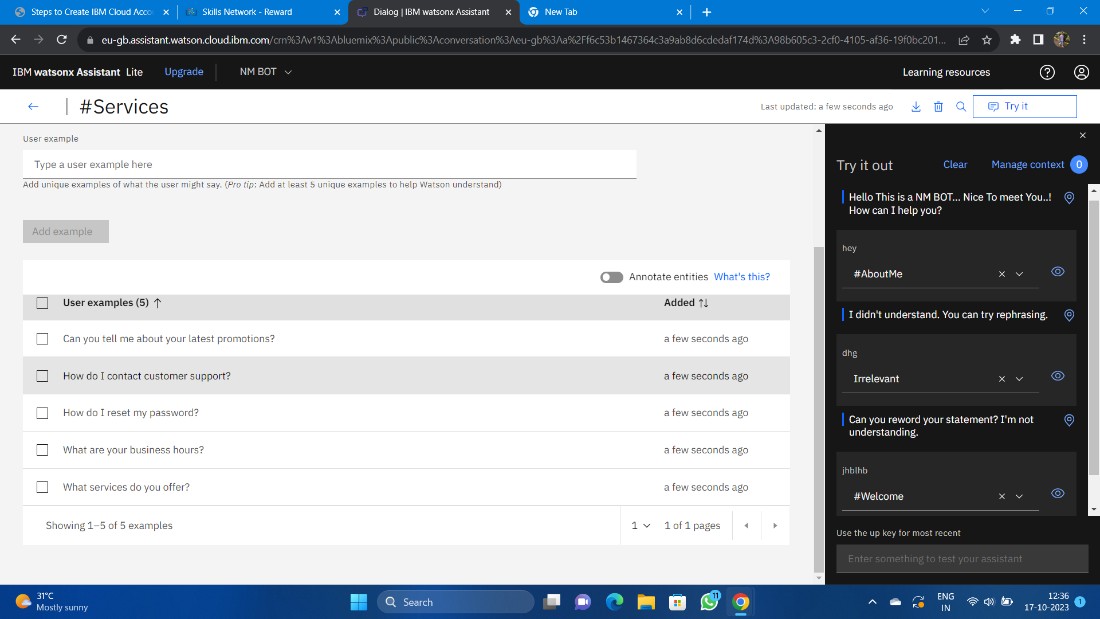
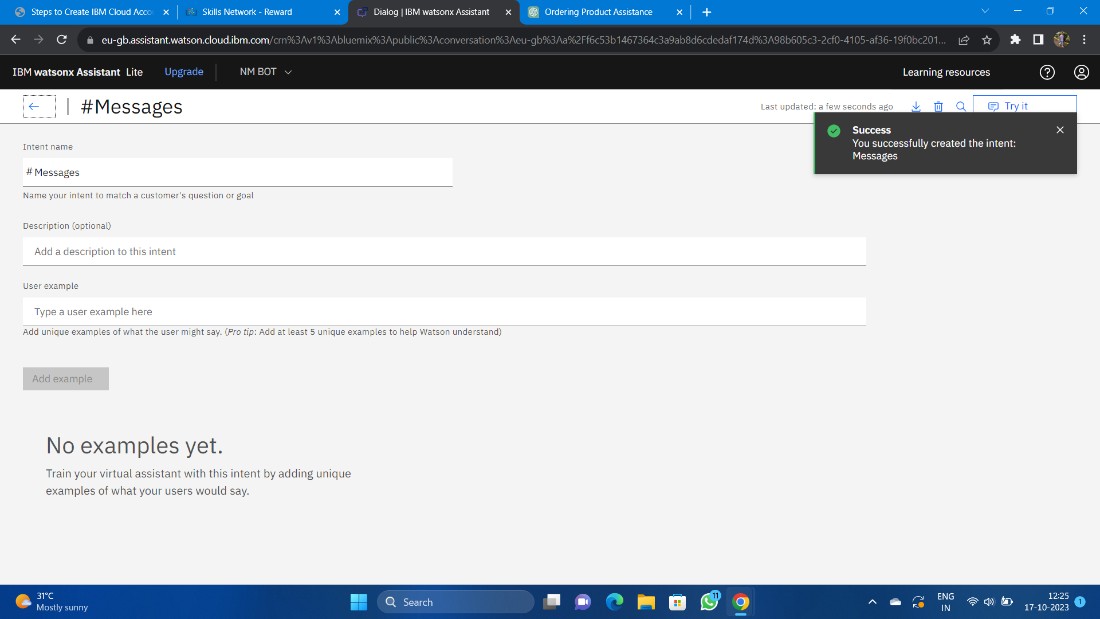
* After activating the Dialog, you will get the Intents, Entities, Dialog, and Content catalog like shown below



#### Step 6:

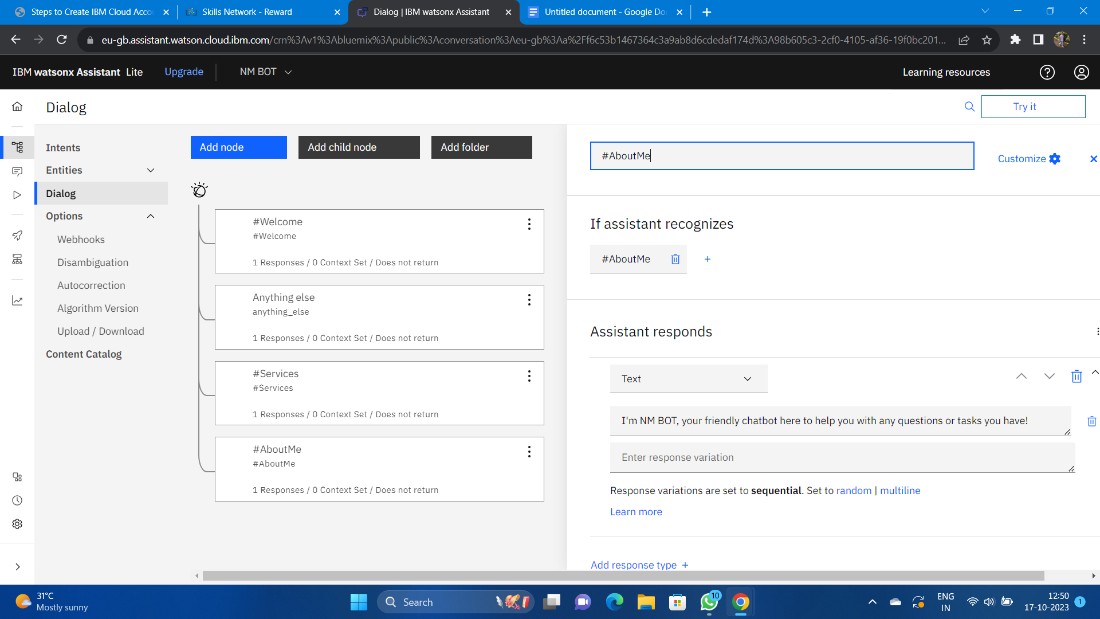
* Create the Entities first and some variables for the entities you have been created.
* Here I have been created the Entity with the name Entertainment and added variables as channels .

#### Step 7:

* Open the Intents and then create the Intents for Messages, Services, AboutMe .

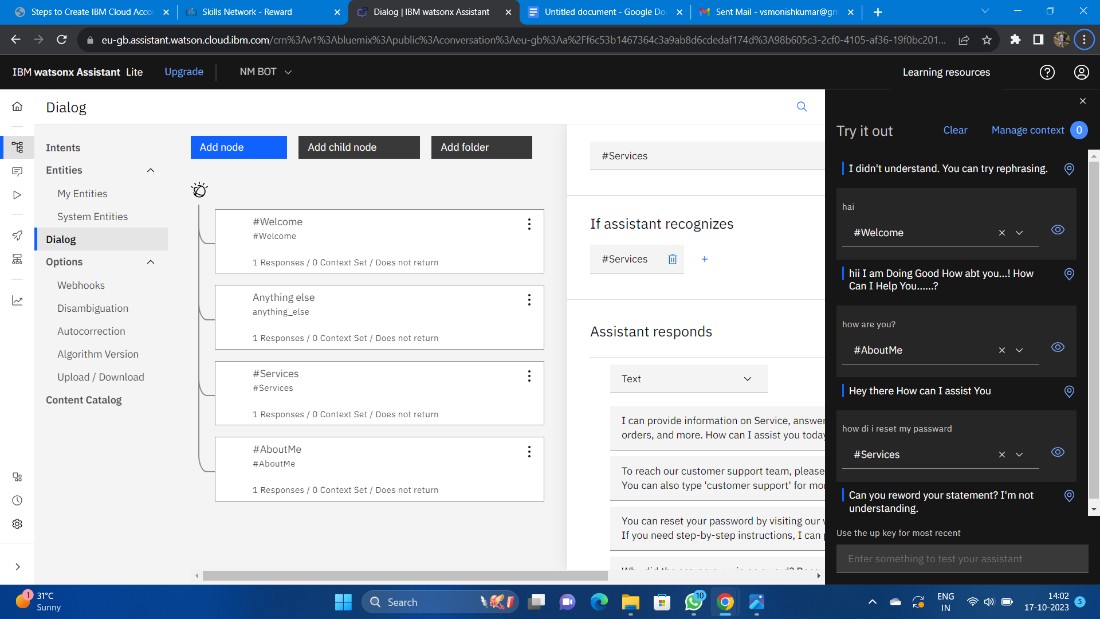
#### Step 8:

* 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.



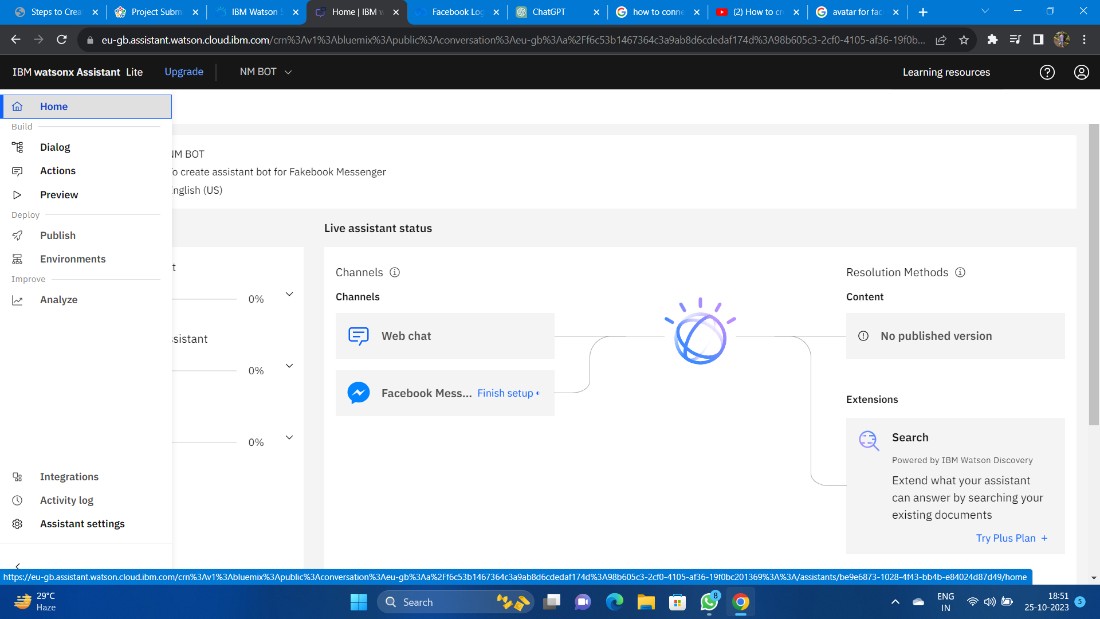
#### Step 9:

* Check the chat bot by clicking the try it before connecting the Face book 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.



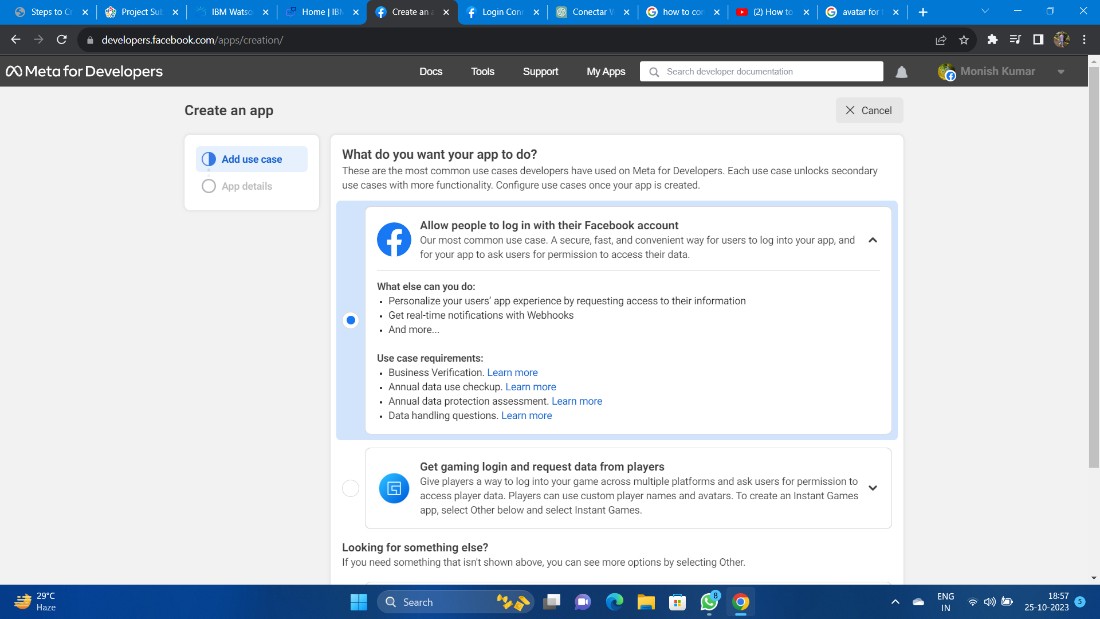
**Step 1:** Create a Facebook Page Log in to your Facebook account.

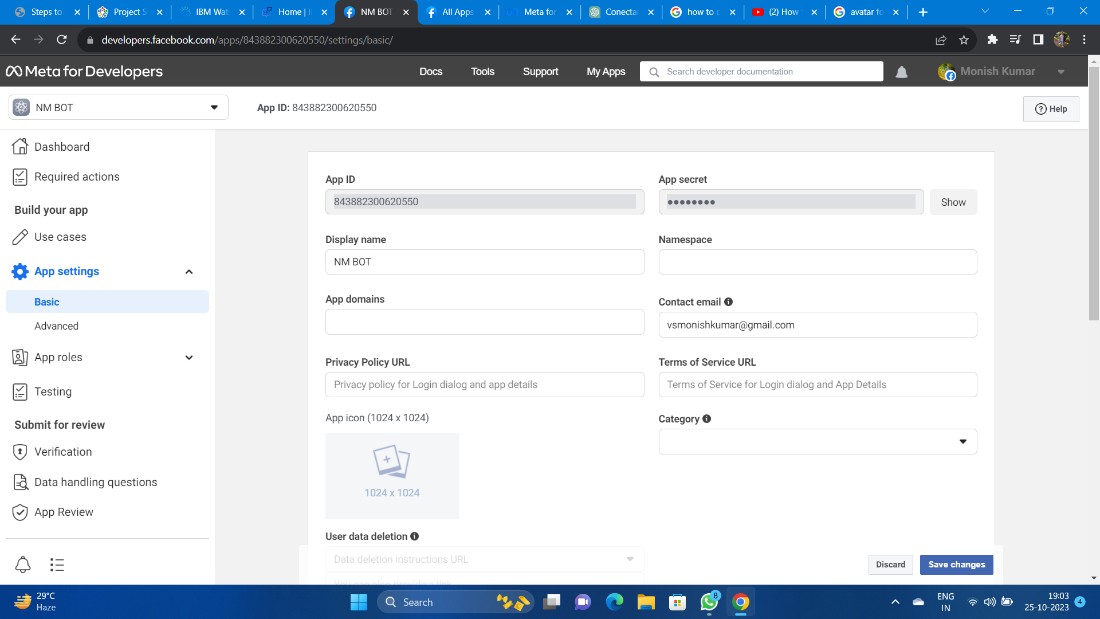
Go to Facebook's Create a Page and follow the steps to create a Facebook Page. This Page will be used to host your chatbot.

**Step 2:** Set Up a Facebook App

Go to the Facebook for Developers website.

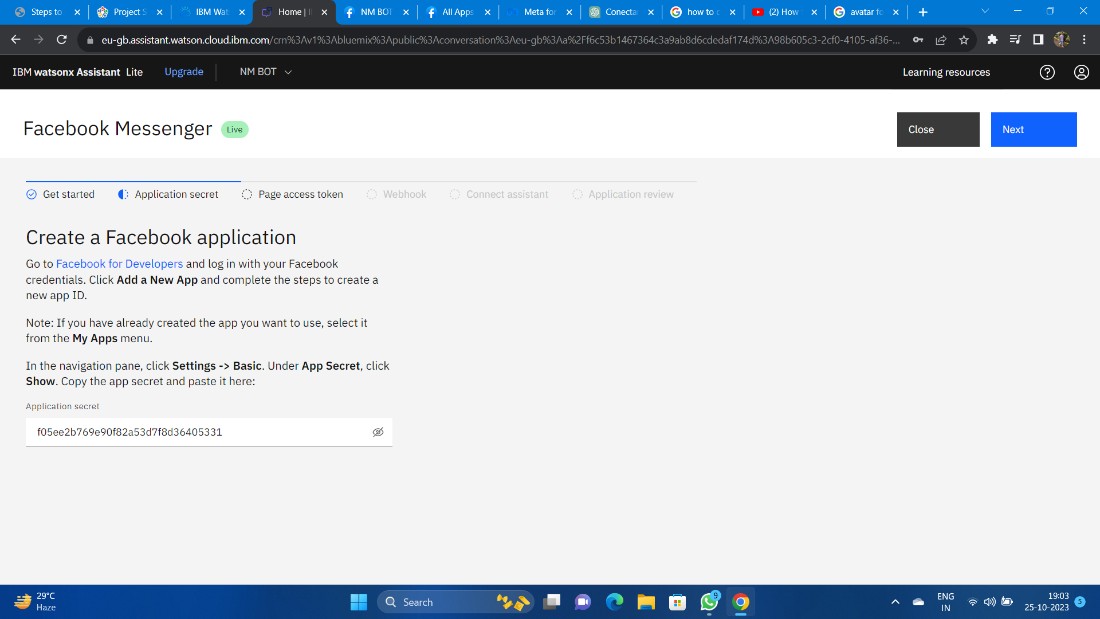
Create a new app by clicking on "My Apps" and then "Create App."





**Step 3:** Configure the Facebook App for Messenger In your Facebook App dashboard:

1. Go to the "Messenger" section.
2. 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:

1. Create a new assistant or use an existing one.
2. 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)

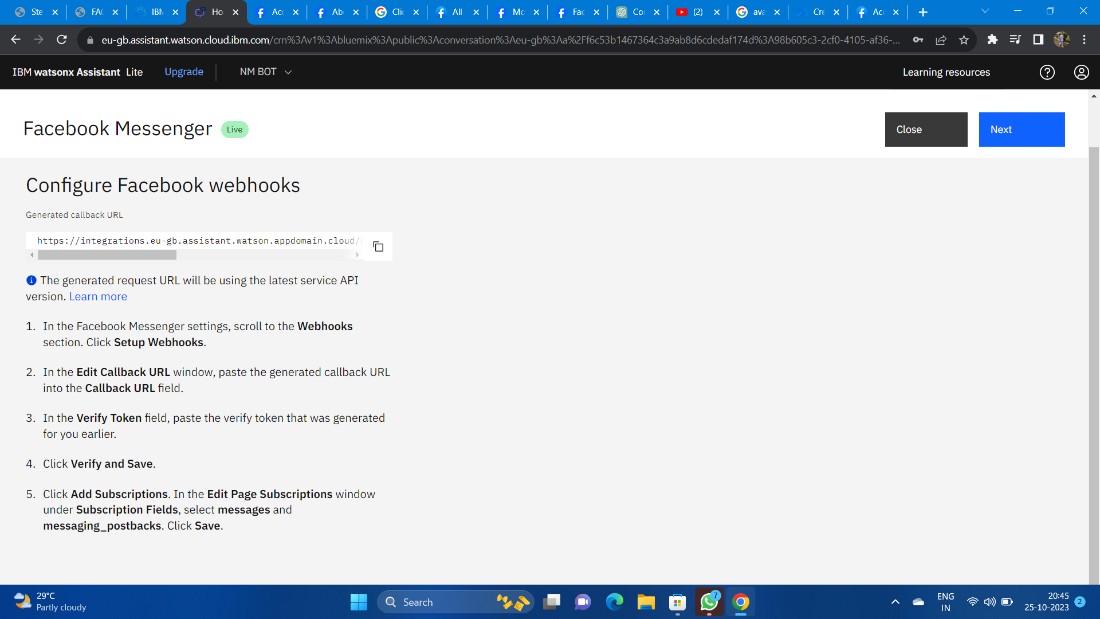
**Step 6:** Integrate Watson Assistant with Facebook Messenger In the Watson Assistant Dashboard:

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

**Step 7:** Set Up the Webhook

In the Watson Assistant integration settings for Facebook Messenger:

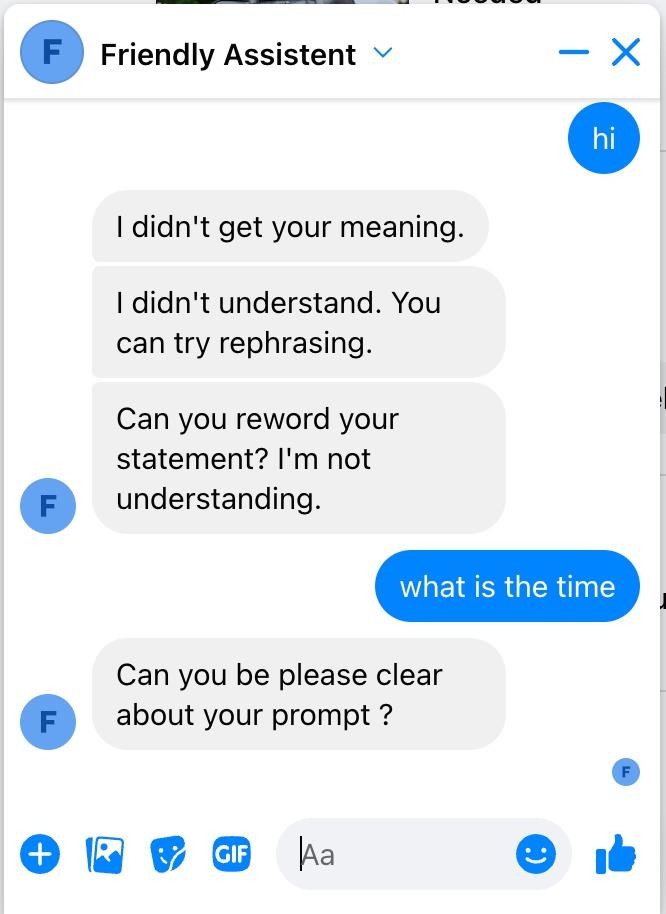
1. Enter the Facebook Page Access Token you obtained in Step 3.
2. Specify the Webhook URL. This URL should be the same as the callback URL you set up in the Facebook App's Webhook settings.
3. Save the changes.



**Step 8:** Subscribe to the Facebook Webhook In the Facebook App settings:

1. In the "Webhooks" section, click on "Edit Subscription."
2. Select the Facebook Page you created in Step 1.
3. Subscribe your Page to the webbooks events.

After competing 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 Facebooks messenger where and which the queries that has been mentioned in the entities, dialogs and interns will be given as reply in my messenger.