

CLOUD APPLICATION DEVELOPMENT

CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT

OBJECTIVE:

To create chatbot-helpful virtual guide using IBM Cloud Watson Assistant to assist the users on popular messaging platforms like Facebook and Slack, to provide useful information, answers FAQs.

DESIGN OF CHATBOT:

The chatbot was designed using the IBM Cloud Watson Assistant.

DEVELOPMENT PHASE:

Phase-1:

CHATBOT PERSONA:

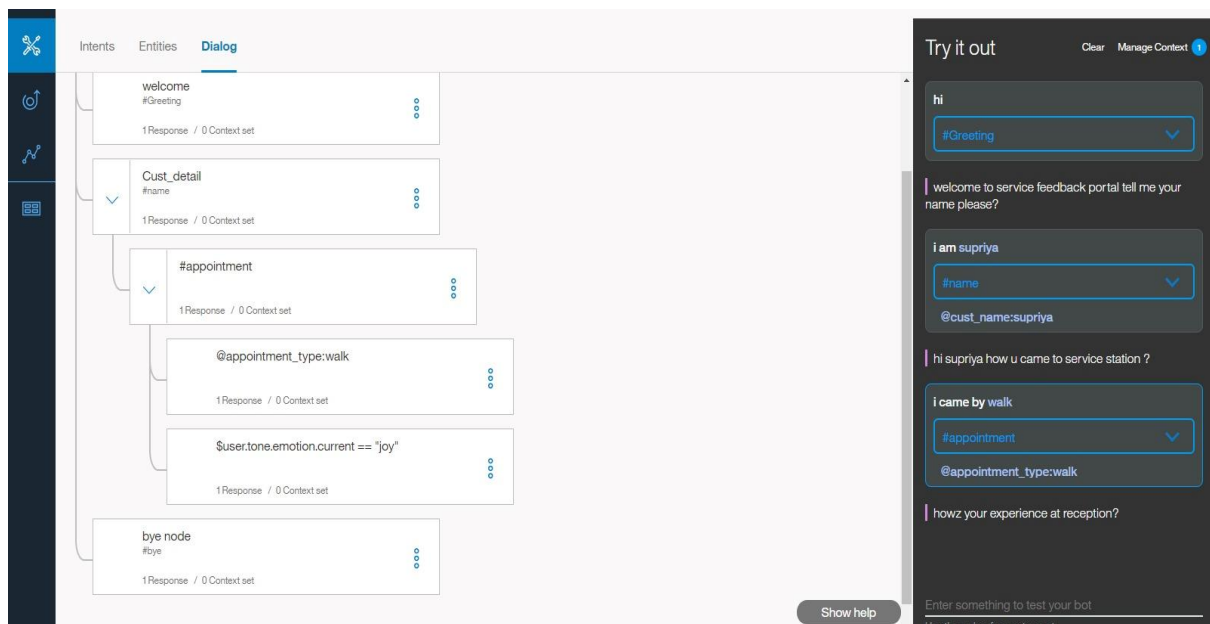
Chatbot Name: AmazBot

Personality:

- AmazBot is friendly, witty and professional in answering the doubts /queries of the user.
- The voice of the chatbot will be polite and enthusiastic.

Conversation flow of AmazBot:

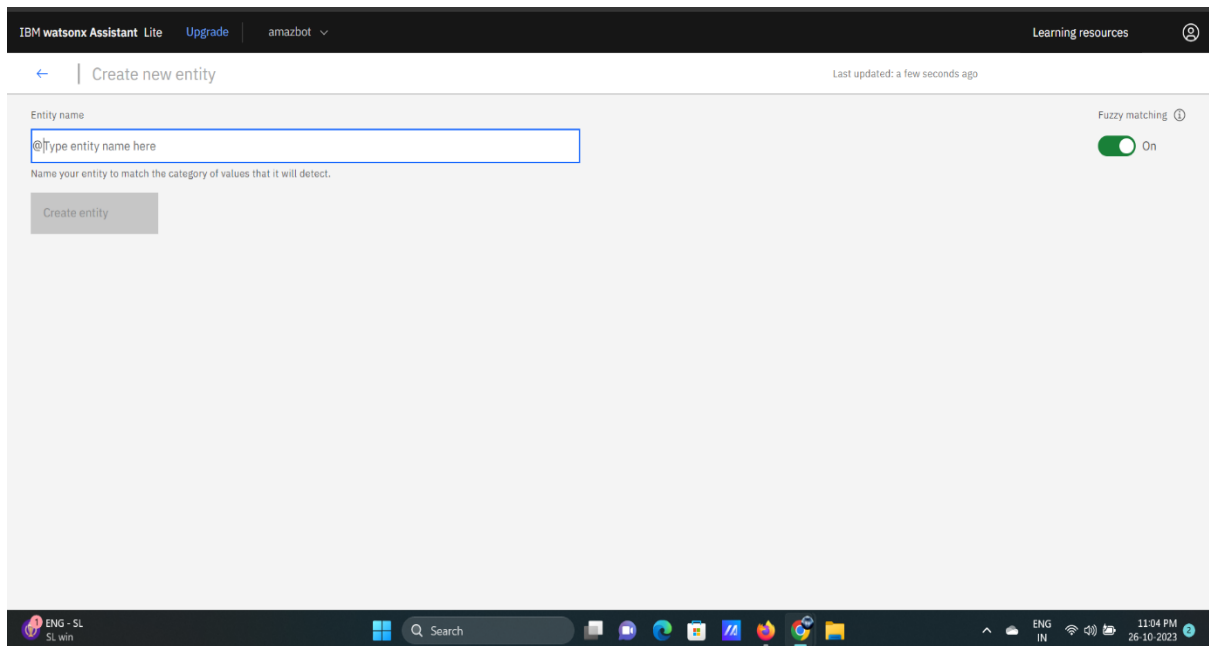
AmazBot is able to converse with the user in the friendly manner by starting the conversation with Greetings. And it maintains its tone in a polite voice as it offers a friendly conversational experience to the users. It provides the user with the information that may be a FAQ or queries.



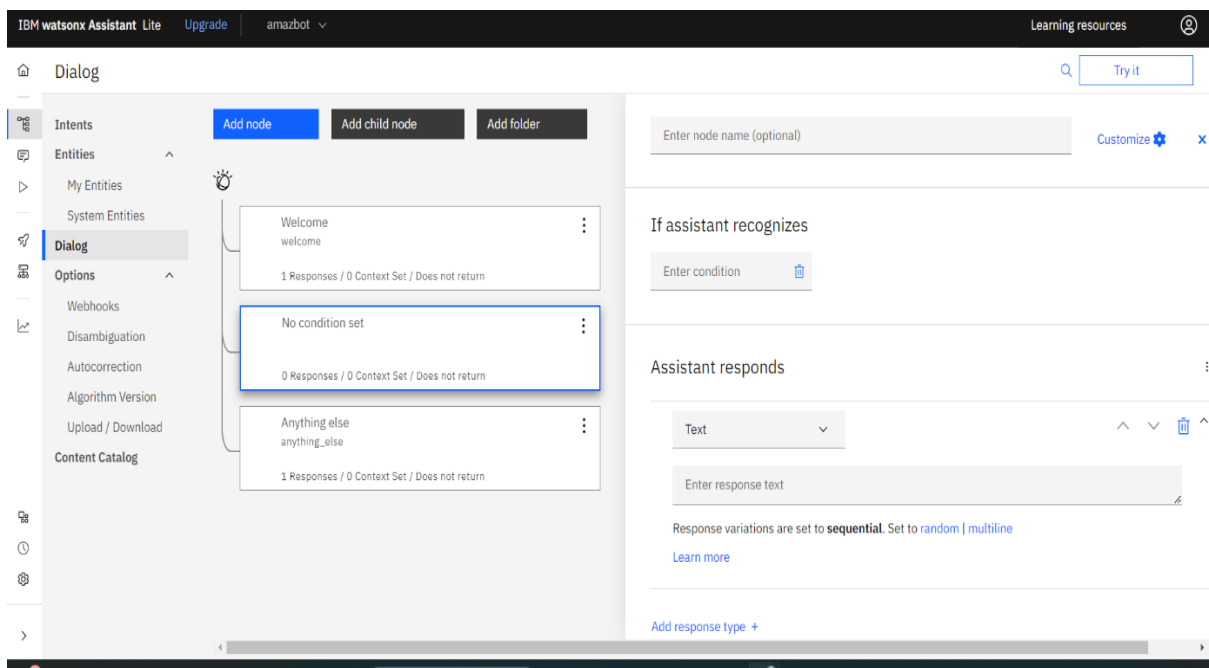
Configuration of intents, entities and dialogue:

The chatbot intents, entities and dialogues are configured using the IBM Cloud Watson Assistant.

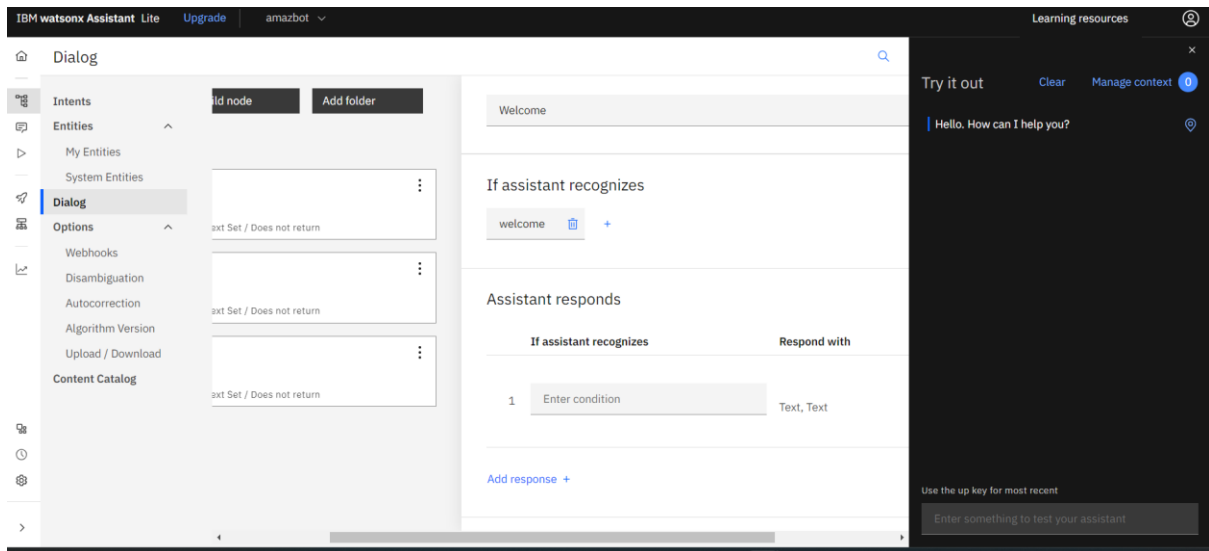
Intents are purposes or goals that are expressed in a customer's input, such as answering a question or processing a bill payment. By recognizing the intent expressed in a customer's input, the Watson Assistant service can choose the correct dialog flow for responding to it.



Dialogues are the core component of a chatbot that you create using IBM Cloud Watson Assistant. They define the flow of the conversation and the logic behind it. They also determine how the chatbot responds to user input and requests

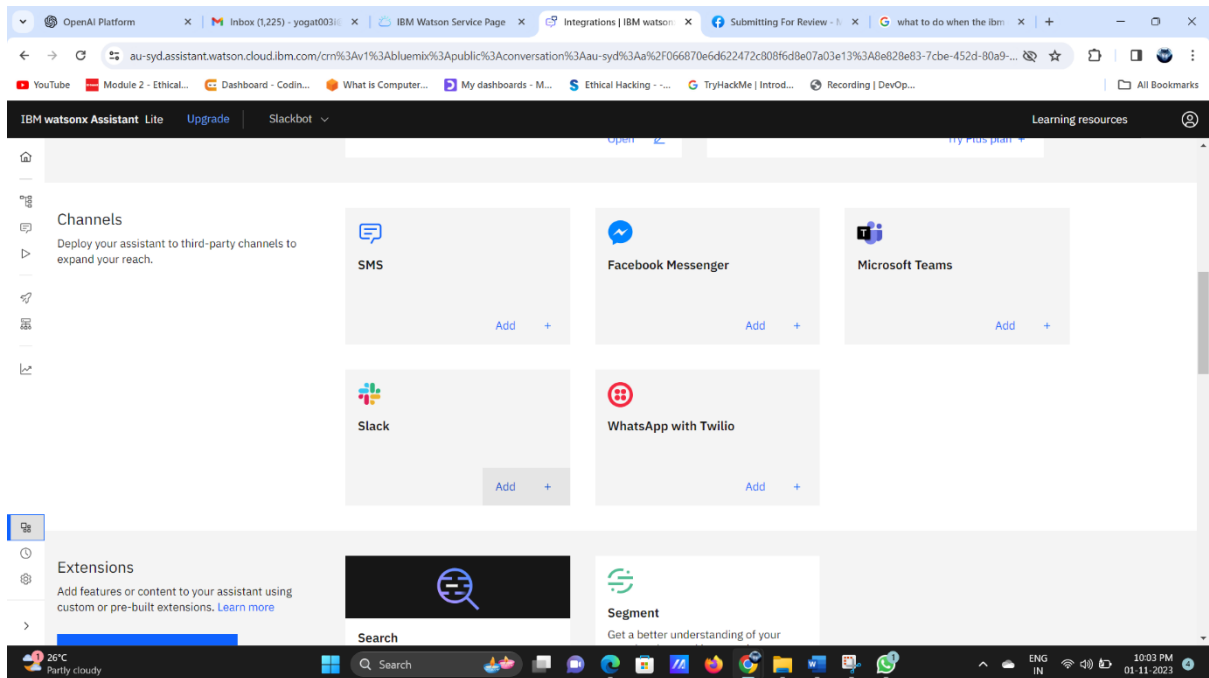


So the Chatbot is further devolved with the flow to conversate with the users.



PHASE-2

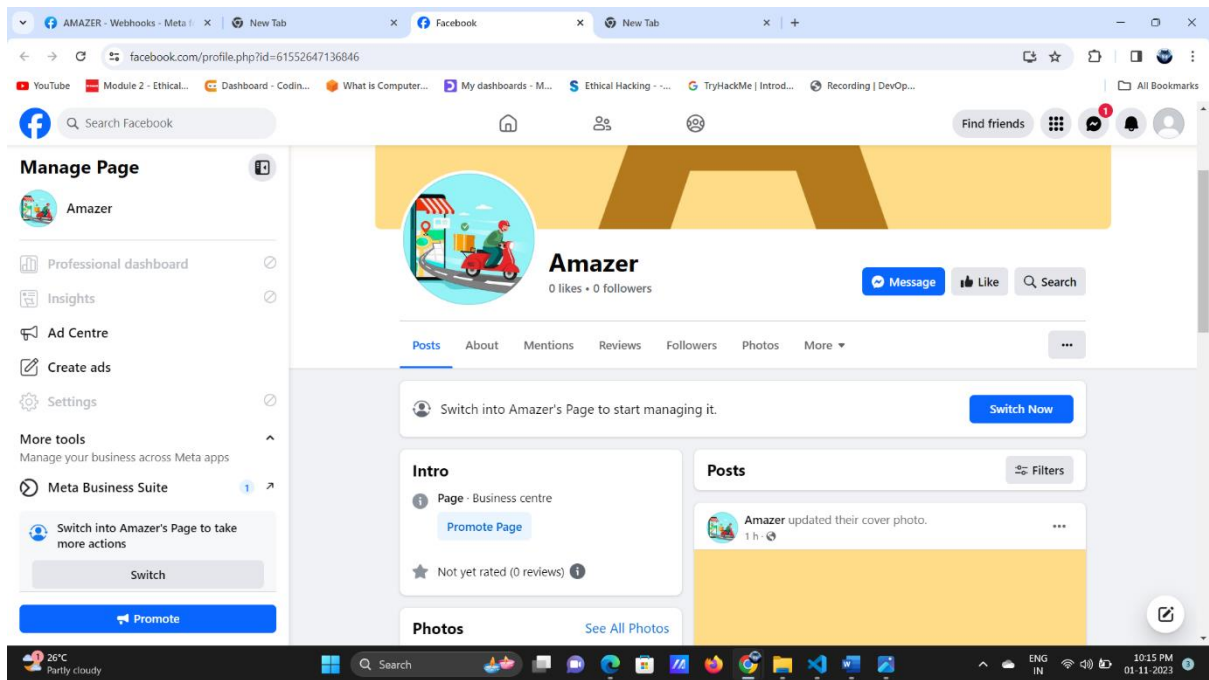
And in the left-bottom corner you find the Integration. Then scroll-down, you find the Facebook messenger and Slack. Click add in the Facebook



Now you will be directed to the integration steps follow it carefully.

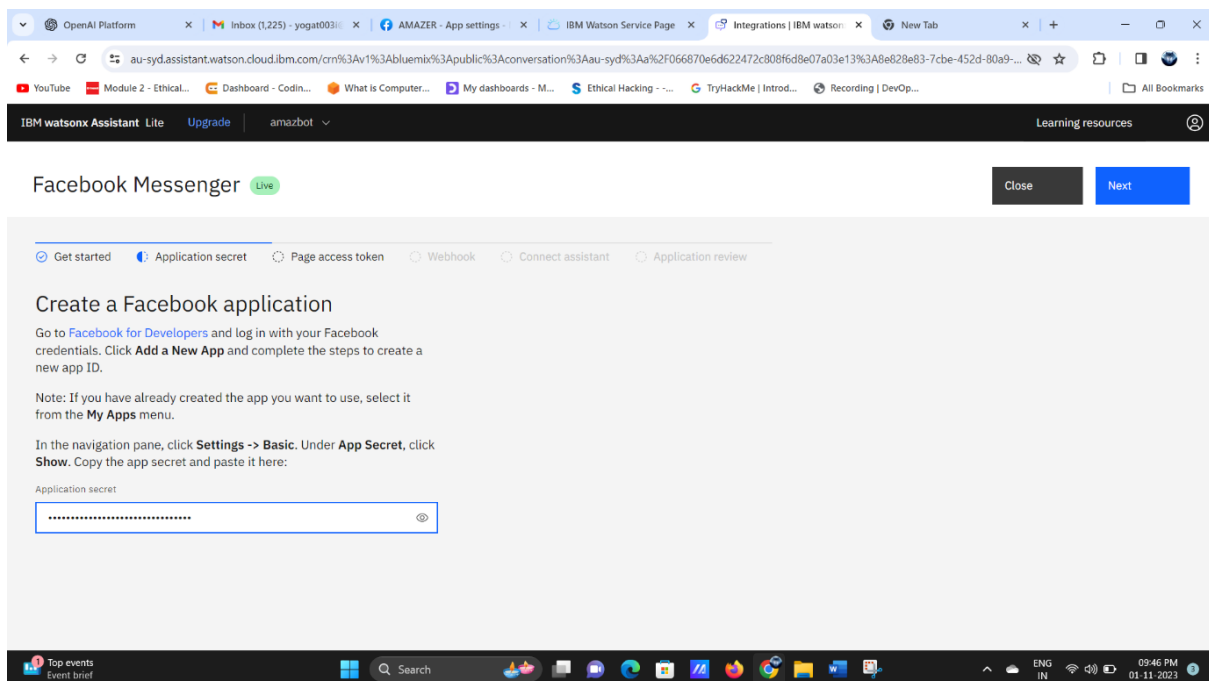
You will need to create your Facebook page by using the Meta for Developers.

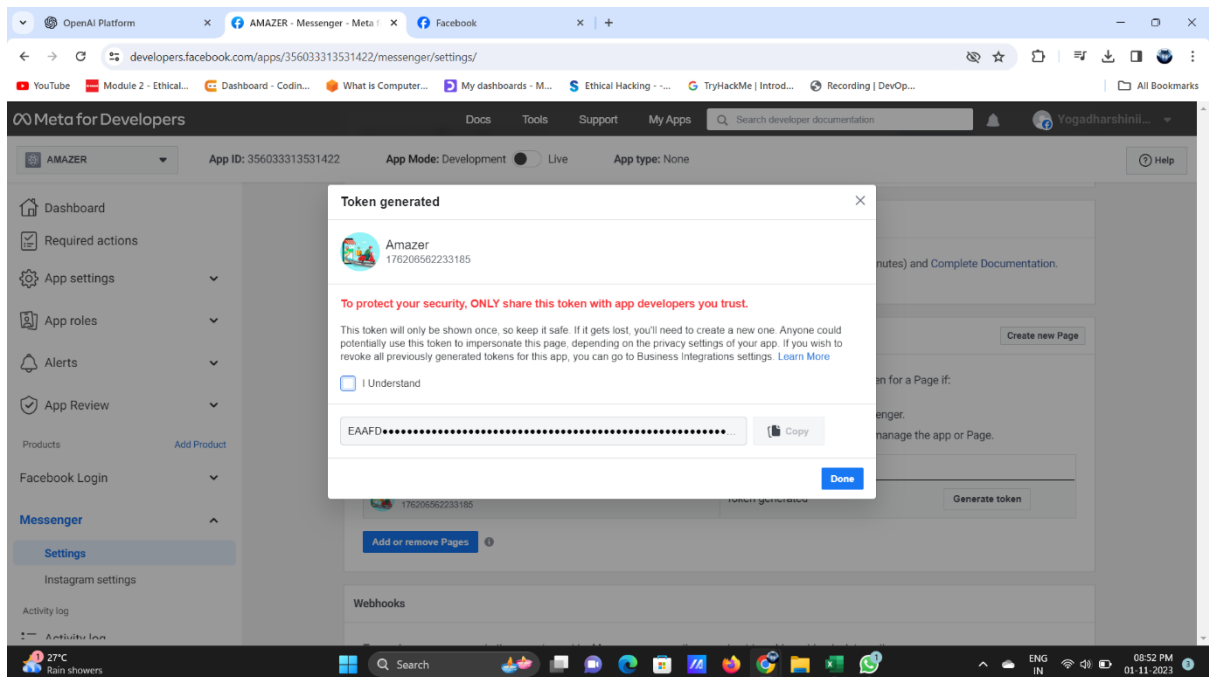
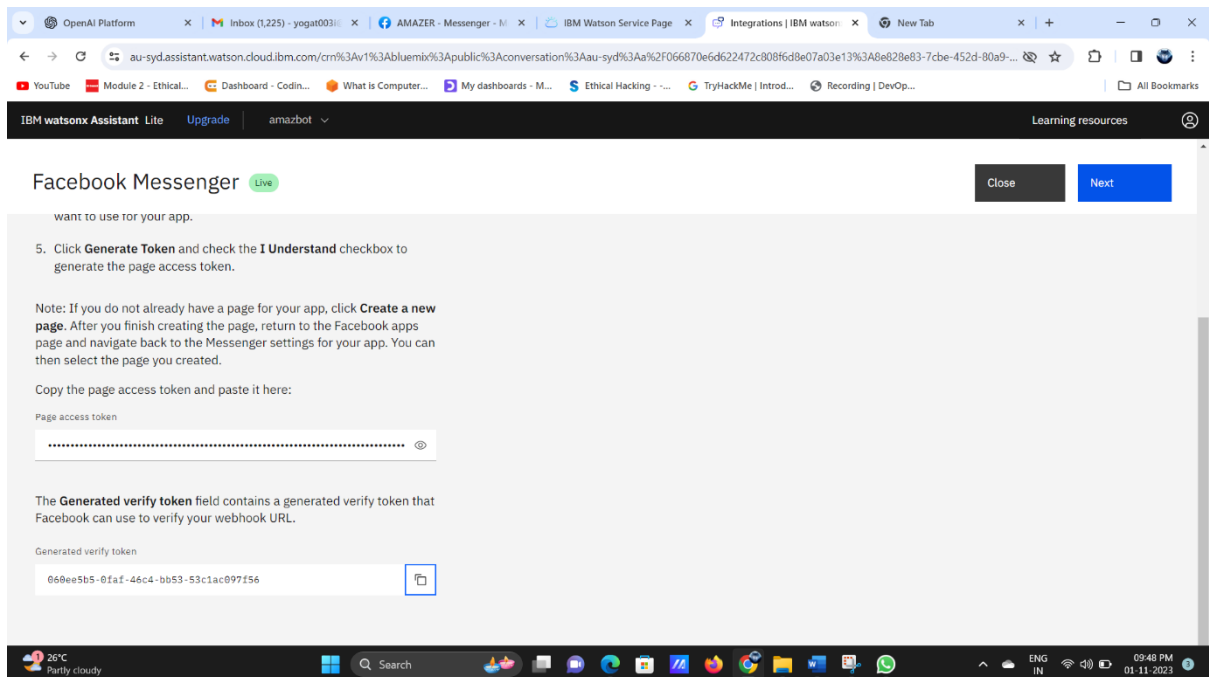
And the proceed with the given steps.



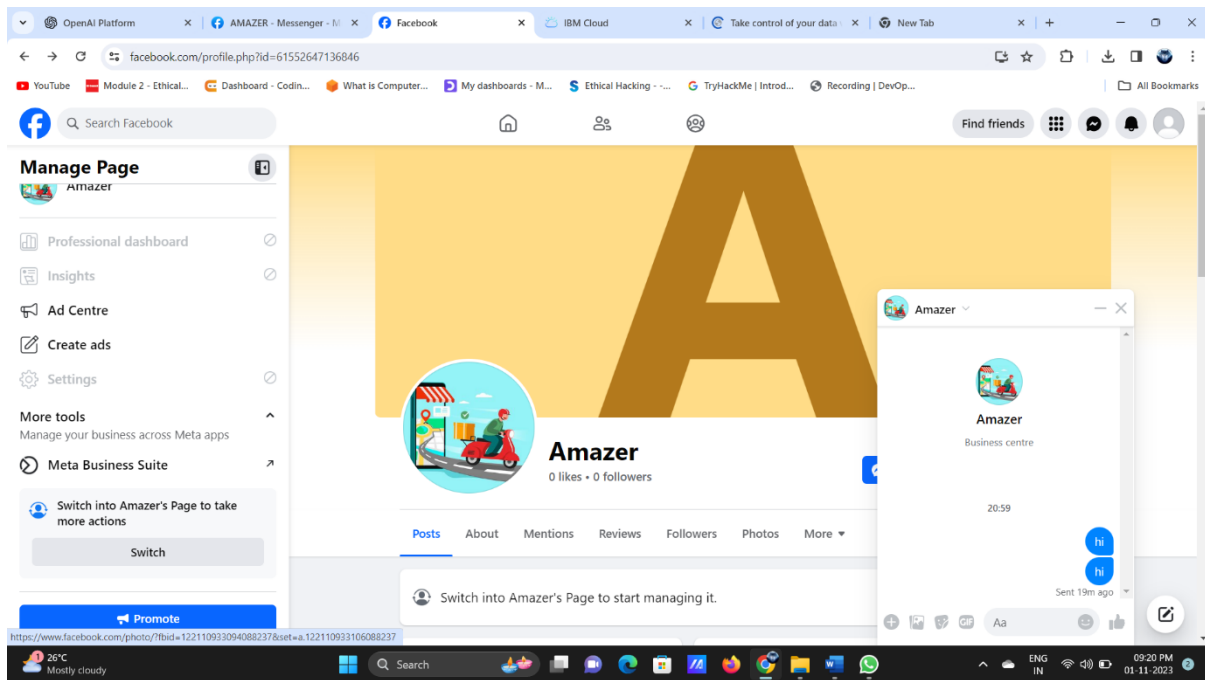
This is the Facebook page we created.

Application Secret is obtained :





Then the chat is then tested as a messenger to answer the queries.



Now for the Slack you need to choose the chatbot you created for it from the dashboard. And the opt for Slack from Integration. And you need to prepare you app by log in to slack and follow the steps are followed:

1. From the Assistants page, click to open the assistant tile that you want to deploy.
2. From the Integrations section, click **Add integration**.
3. Click **Slack**.
4. You need to have a Slack app to connect to.

If you don't have a Slack app, create one now. See [Starting with Slack apps](#).

5. Go to the [Your Apps](#) page on the Slack website, and then click the app you want to use.

Open the Slack app in a new browser tab, so you can easily switch back and forth between the Slack app settings page and Watson Assistant Slack integration configuration page.

6. From the settings page for your Slack app, open the **App Home** page.
7. Add access scopes for your Slack app.

The button label might be *Review Scopes to Add Or Update scopes* depending on whether you are creating a new app or editing an app that you created before February 2020.

The method for Slack access changed. For more information about it, read the [Slack blog post](#) about it.

8. Assign bot token scopes to your Slack app. At a minimum, apply the following scopes:

- `app_mentions:read`
- `chat:write`
- `im:history`
- `im:read`
- `im:write`

9. Click *Install App to Workspace*, and then allow the installation when prompted.

If you are editing scopes for an existing application, reinstall it.

10. From the Slack settings App Home page, enable the *Always Show My Bot As Online* setting.
11. Go to the *OAuth and Permissions* page in Slack, copy the *Bot User OAuth Access Token*.
12. From the Watson Assistant Slack integration configuration page, paste the token that you copied in the previous step into both the **OAuth access token** and **Bot user OAuth access token** fields.
13. On the Slack app settings page, go to the *Basic Information* page, and then find the *App Credentials* section. Copy the app credential verification token.
14. From the Watson Assistant Slack integration configuration page, paste the verification token that you copied in the previous step into the **Verification token** field.
15. Click **Generate request URL**, and then copy the generated request URL.

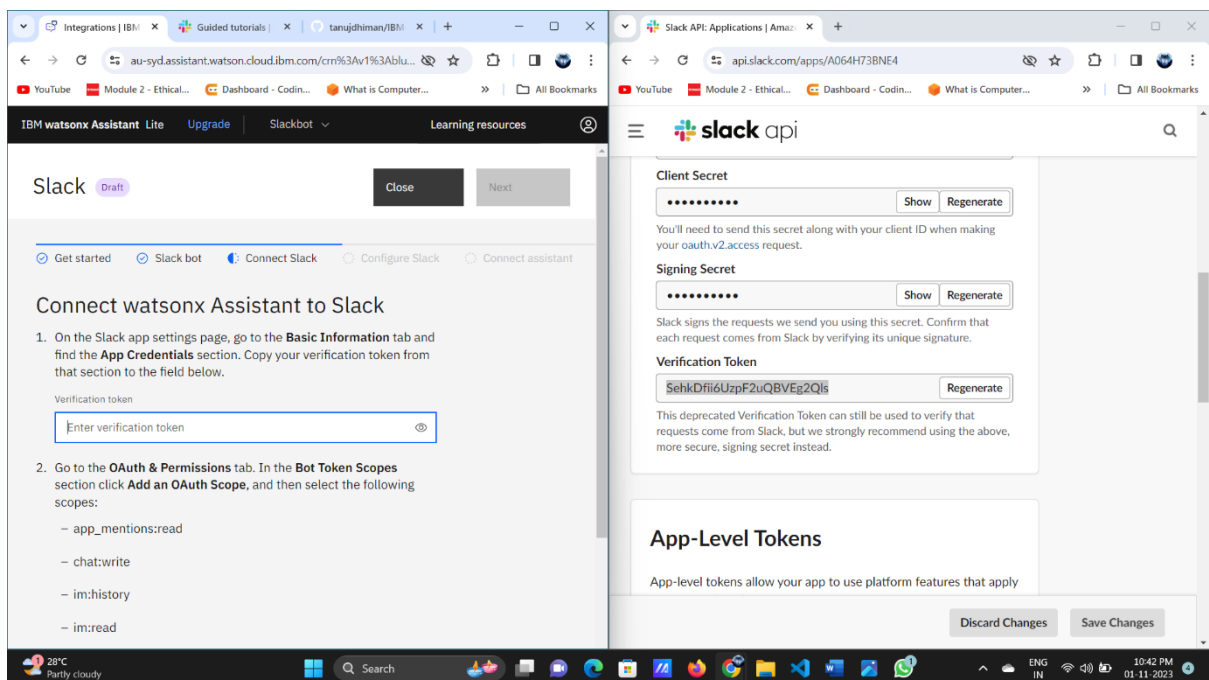
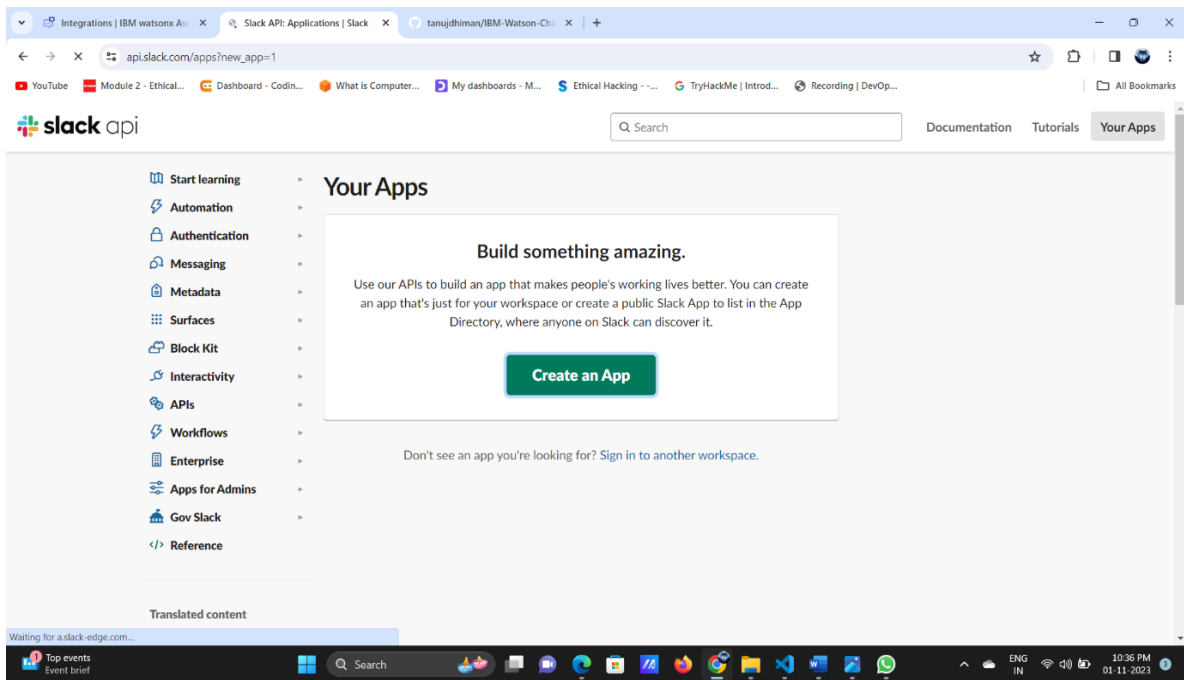
16. Return to the Slack app settings page. Open the *Event Subscriptions* page, and then turn on *Enable Events*. Paste the request URL that you copied in the previous step into the field.
17. On the *Event Subscriptions* page in Slack, find the *Subscribe to Bot Events* section. Click *Add Bot User Event*, and then select the event types you want to subscribe to. You must select at least one of the following types:
 - `message.im`: Listens for message events that are posted in a direct message channel.
 - `app_mention`: Listens for only message events that mention your app or bot.

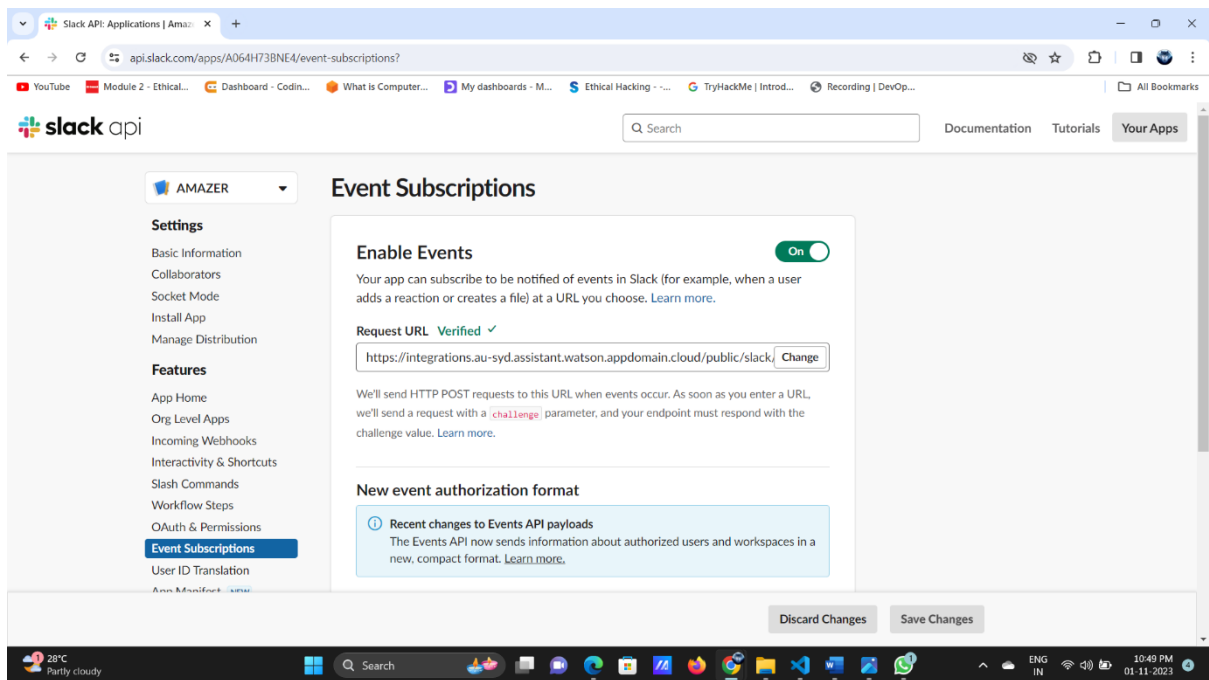
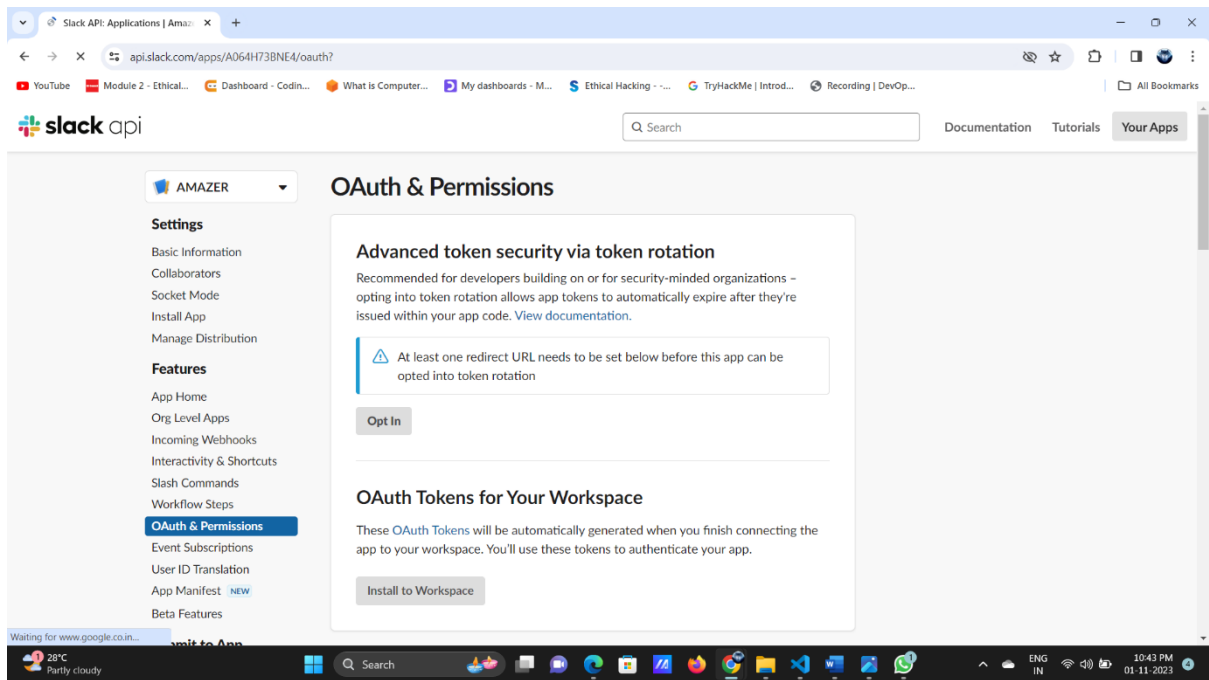
Choose the *app_mention* entry in normal font, *not* the `app_mention` entry that is in bold font.
18. Click *Save Changes*.
19. Optional: To add support for showing buttons, menus, and disambiguation options in the Slack app, go to the *Interactive Components* tab and enable the feature. Paste your request URL in the provided text entry field, and then click *Enable Interactive Components*.

Chatting with the assistant

To start a chat with the assistant, complete the following steps:

1. Open Slack, and go to the workspace associated with your app.
2. Click the application that you created from the Apps section.
3. Chat with the assistant.





Integrations | IBM watsonx Assistant | Guided tutorials | Slack | tanujdhiran/IBM-Watson-Ch...

au-syd.assistant.watson.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2F066870e6d622472c808f6d8e07a03e13%3A8e828e63-7cbe-452d-80a9-...

IBM watsonx Assistant Lite Upgrade Slackbot Learning resources

Slack Draft


Close Next

2. Go to the **OAuth & Permissions** tab. In the **Bot Token Scopes** section click **Add an OAuth Scope**, and then select the following scopes:

- app_mentions:read
- chat:write
- im:history
- im:read
- im:write

3. On the **OAuth & Permissions** tab, Click **Install App to Workspace**, and then click **Allow**. You should be redirected back to the OAuth & Permissions page. Copy and paste your **Bot user OAuth access token** to the following field.

Bot user OAuth access token

xxxb-6153033897376-6153281985328-azuqNsM5fGKY18iu9yZ5 

Hide password

28°C Partly cloudy Search 10:48 PM 01-11-2023

Slack API: Applications | Amazon

api.slack.com/apps/A064H73BNE4/app-home?

App display name saved!

App Home

- Org Level Apps
- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- Workflow Steps
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- App Manifest NEW
- Beta Features
- Submit to App Directory**
- Review & Submit
- Give feedback
- Slack ♥
- Help
- Contact
- Policies
- Our Blog

Your App's Presence in Slack

Your app display name and presence is how your users will see your app in Slack.

App Display Name Edit

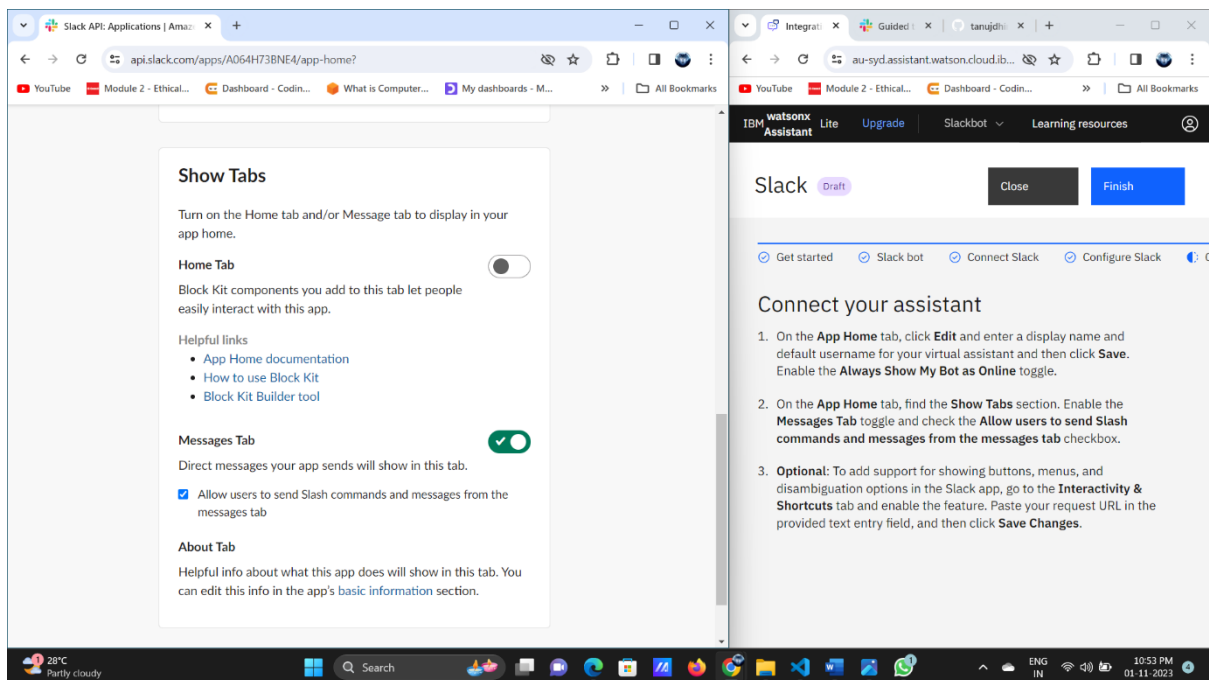
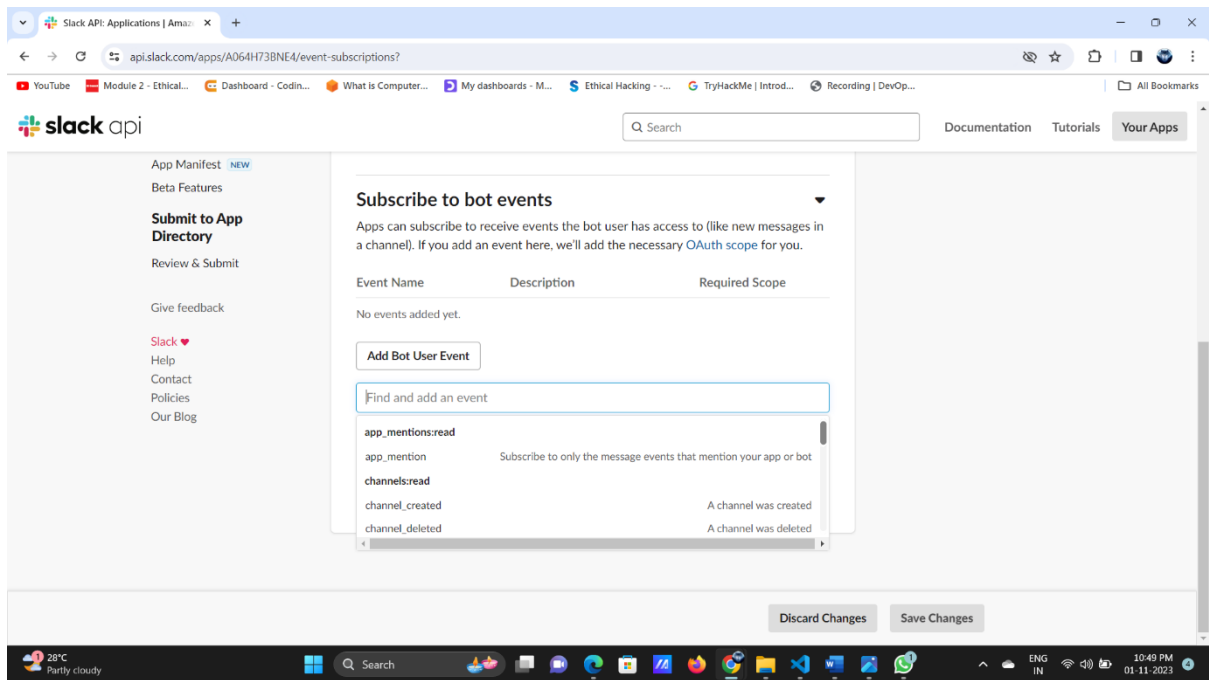
Display Name (Bot Name): AMAZER
Default Name: amazer

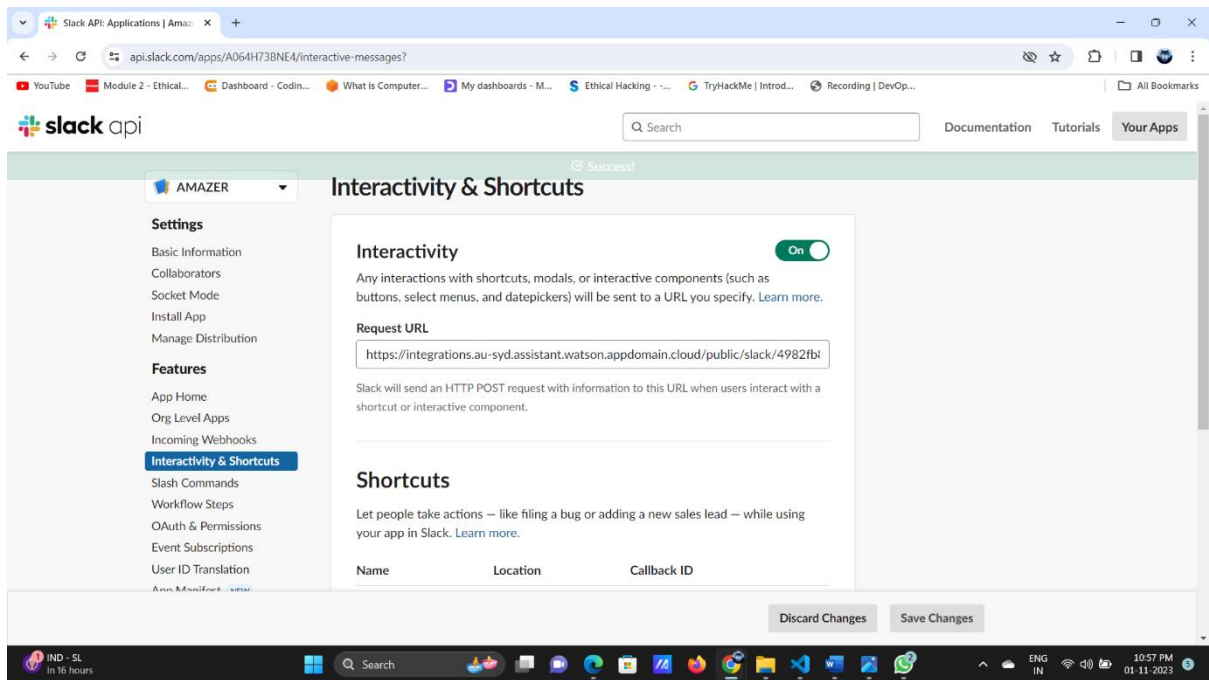
Always Show My Bot as Online Toggle

Bot Token Added

A [bot token](#) in your app lets users at-mention it, add it to channels and conversations, and allows you to turn on tabs in your app's home. It makes it possible for users to interact with your app in Slack. You can adjust your Bot Token's permissions in [OAuth & Permissions](#).

28°C Partly cloudy Search 10:51 PM 01-11-2023





By this we have integrated the chatbot with Facebook and Slack. And the link for chatbot of the Facebook Messenger:

<https://www.facebook.com/profile.php?id=61552647136846>.

Link for my repository:

<https://github.com/Yoga-2003/Phase1.git>