

Phase 4

PROJECT: CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT

PROBLEM STATEMENT: (DEVELOPMENT PART2)

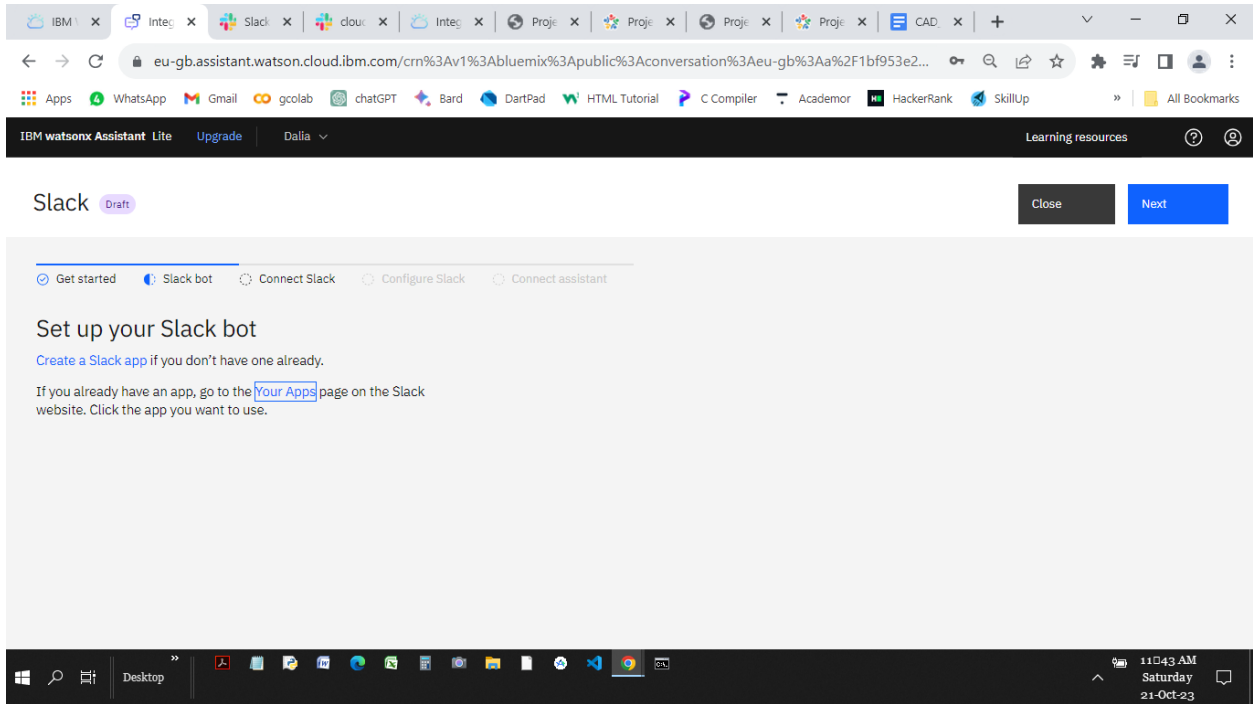
In this part you will continue building your project.

- **Continue building the chatbot by integrating it with messaging platforms and refining the responses.**
- **Integrate the chatbot with Facebook Messenger and Slack using respective APIs.**
- **Ensure that the conversation flows naturally and that the chatbot's responses are informative and accurate.**

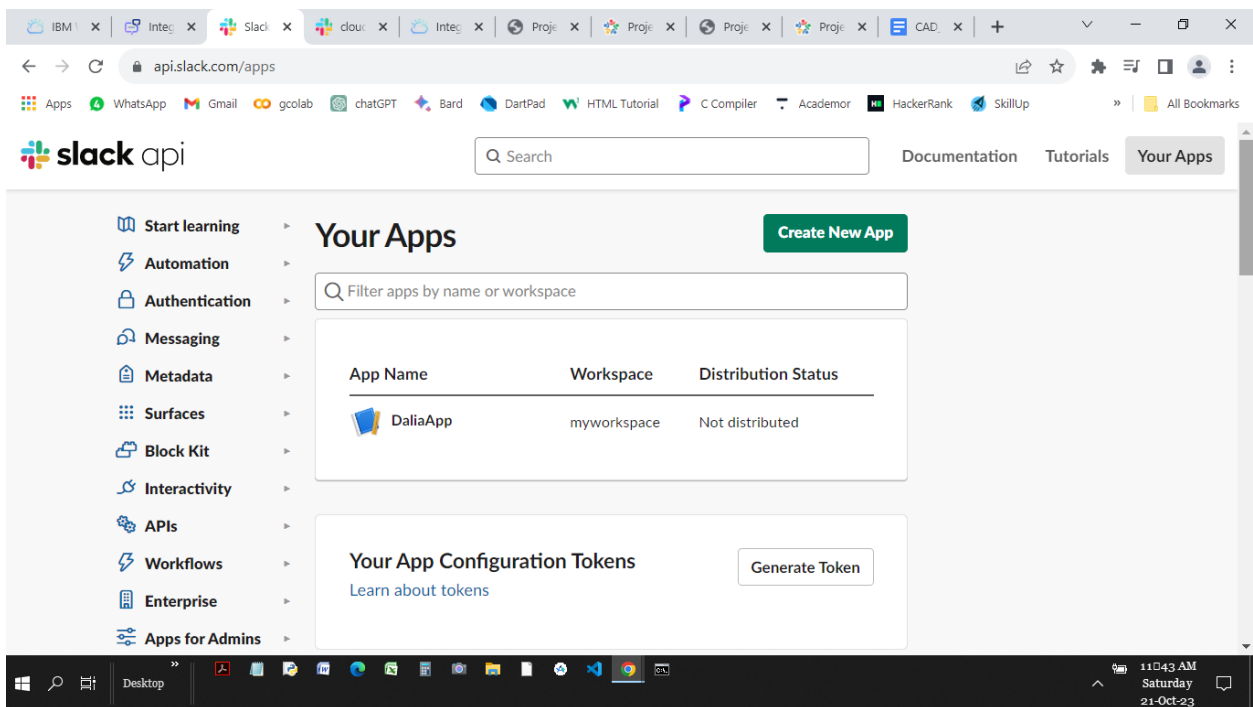
Integration with Slack:

Steps to integrate chatbot with slack:

1. **First we will create a slack app and**
2. **Associate the Slack app with the Slack workspace then**
3. **Connecting watson assistant with slack and**
4. **Configuring the slack**
5. **Finally ,testing the chatbot's responses are accurate**



Creating a slack app called DaliaApp and associating it with slack workspace



Connecting watson assistant to slack using the token and providing permissions

The screenshot shows the IBM Watson Assistant web interface. The browser's address bar displays the URL: `eu-gb.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F1bf953e2...`. The page title is "Slack" with a "Draft" label. A "Close" button and a "Next" button are visible in the top right corner.

The main content area is titled "Connect watsonx Assistant to Slack" and contains a progress bar with five steps: "Get started", "Slack bot", "Connect Slack" (current step), "Configure Slack", and "Connect assistant".

The instructions for the "Connect Slack" step are as follows:

- On the Slack app settings page, go to the **Basic Information** tab and find the **App Credentials** section. Copy your verification token from that section to the field below.

A text input field for the "Verification token" is shown, containing a masked token (dots) and a copy icon.

- 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

- 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.

A text input field for the "Bot user OAuth access token" is shown, containing a masked token (dots) and a copy icon. A "Show password" button is located below the field.

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock displaying "11:49 AM Saturday 21-Oct-23".

Slack confirm... x Project Subm... x Project Subm... x Ajishwaryap... x CAD_Phase4... x IBM Watson S... x Integrations |... x Slack API App... x (D) Messenger... x Where work h... x general - my... x +

api.slack.com/apps/A06292JKPDJ/general? | Gmail | YouTube | Maps

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

Manage distribution

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID
A06292JKPDJ

Date of App Creation
October 20, 2023

Client ID
6074530130179.6077086669460

Client Secret
***** [Show](#) [Regenerate](#)

You'll need to send this secret along with your client ID when making your `oauth.v2.access` request.

Signing Secret
***** [Show](#) [Regenerate](#)

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token
buqaTfMyszMtVoVCvhFCJzy [Regenerate](#)

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.

App-Level Tokens

[Discard Changes](#) [Save Changes](#)

Slack confirm... x Project Subm... x Project Subm... x Ajishwaryap... x CAD_Phase4... x IBM Watson S... x Integrations |... x Slack API App... x (D) Messenger... x Where work h... x general - my... x +

api.slack.com/apps/A06292JKPDJ/oauth? | Gmail | YouTube | Maps

slack api

Q Search Documentation Tutorials Your Apps

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	
app_mentions:read	View messages that directly mention @DallaApp in conversations that the app is in	✕
chat:write	Send messages as @DallaApp	✕
im:history	View messages and other content in direct messages that DallaApp has been added to	✕
im:read	View basic information about direct messages that DallaApp has been added to	✕
im:write	Start direct messages with people	✕

[Add an OAuth Scope](#)

User Token Scopes

Scopes that access user data and act on behalf of users that authorize them.

Configuring the slack bot and enabling events

The screenshot shows a web browser window with multiple tabs open, including IBM, Integ, Slack, cloud, and several Project and CAD tabs. The address bar shows a URL from eu-gb.assistant.watson.cloud.ibm.com. The browser's bookmark bar contains links to Apps, WhatsApp, Gmail, gcolab, chatGPT, Bard, DartPad, HTML Tutorial, C Compiler, Academor, HackerRank, and SkillUp. The page title is "IBM watsonx Assistant Life Upgrade" with a user profile "Dalia".

The main content area is titled "Slack" with a "Draft" label. It features a "Close" button and a "Next" button. Below the title is a progress bar with five steps: "Get started", "Slack bot", "Connect Slack", "Configure Slack" (the current step), and "Connect assistant".

The "Configure Slack" section is titled "Configure your Slack bot" and contains two numbered instructions:

1. On the Slack app settings page, go to the **Event Subscriptions** tab. Switch the **Enable Events** toggle to the **On** position. In the provided text entry field, paste your request URL.

Generated request URL
740-3f3c-4be0-ab8f-fd0accf409fa7/message?version=2023-09-15
4

The generated request URL will be using the latest service API version. [Learn more](#)
2. On the **Event Subscriptions** tab, 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.impressions for message events that are posted in...

The Windows taskbar at the bottom shows the system clock as 11:05 AM on Saturday, 21-Oct-23.

IBM x Integ x Slack x clou x Integ x Proje x Proje x Proje x CAD x +

eu-gb.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F1bf953e2...

Apps WhatsApp Gmail gcolab chatGPT Bard DartPad HTML Tutorial C Compiler Academor HackerRank SkillUp All Bookmarks

IBM watsonx Assistant Lite Upgrade Dalia Learning resources

Slack Draft

Close Next

Switch the **Enable Events** toggle to the **On** position. In the provided text entry field, paste your request URL.

Generated request URL

740-3f3c-4be0-ab8f-fd0ac409fa7/message?version=2023-09-15

The generated request URL will be using the latest service API version. [Learn more](#)

2. On the **Event Subscriptions** tab, 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.

3. Click **Save Changes**.

Desktop 11:56 AM Saturday 21-Oct-23

Gmail YouTube Maps

Settings

Basic InformationCollaboratorsSocket ModeInstall AppManage Distribution

Features

App HomeOrg Level AppsIncoming WebhooksInteractivity & ShortcutsSlash CommandsWorkflow StepsOAuth & Permissions

Event Subscriptions

User ID TranslationApp Manifest NEW

Beta Features

Submit to App Directory

Review & Submit

Give feedback

Slack HelpContactPoliciesOur Blog

Enable Events

On

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more](#).

Request URL Verified

https://integrations.eu-gb.assistant.watson.appdomain.cloud/public/slack/69

Change

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a **challenge** parameter, and your endpoint must respond with the challenge value. [Learn more](#).

New event authorization format

Recent changes to Events API payloads

The Events API now sends information about authorized users and workspaces in a new, compact format. [Learn more](#).

Subscribe to bot events

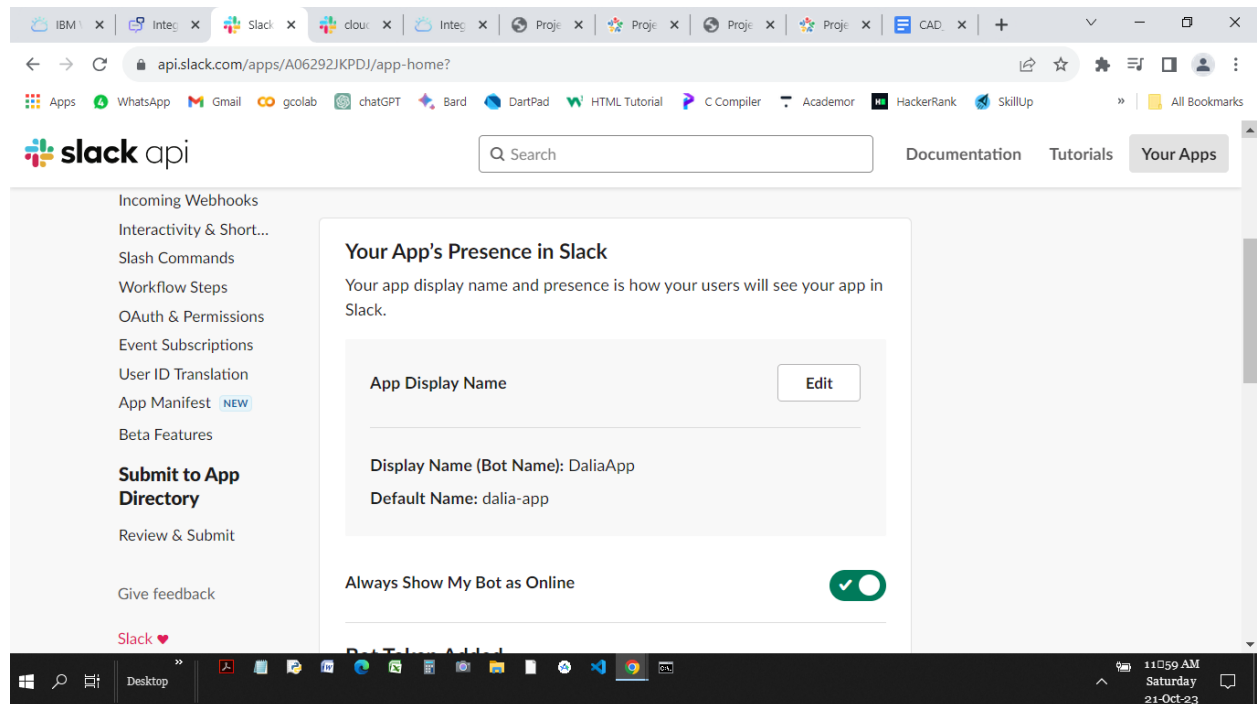
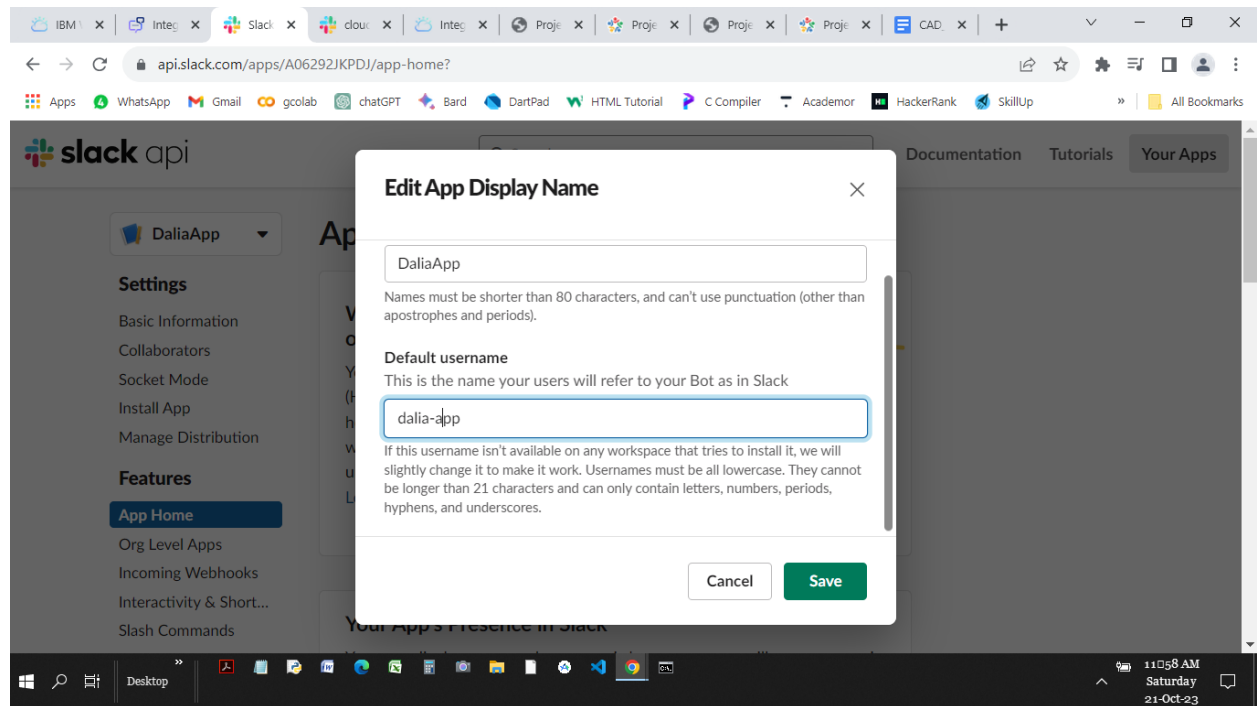
Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary OAuth scope for you.

Event Name	Description	Required Scope
app_mention	Subscribe to only the message events that mention your app or bot	app_mentions:read
message.im	A message was posted in a direct message channel	im:history

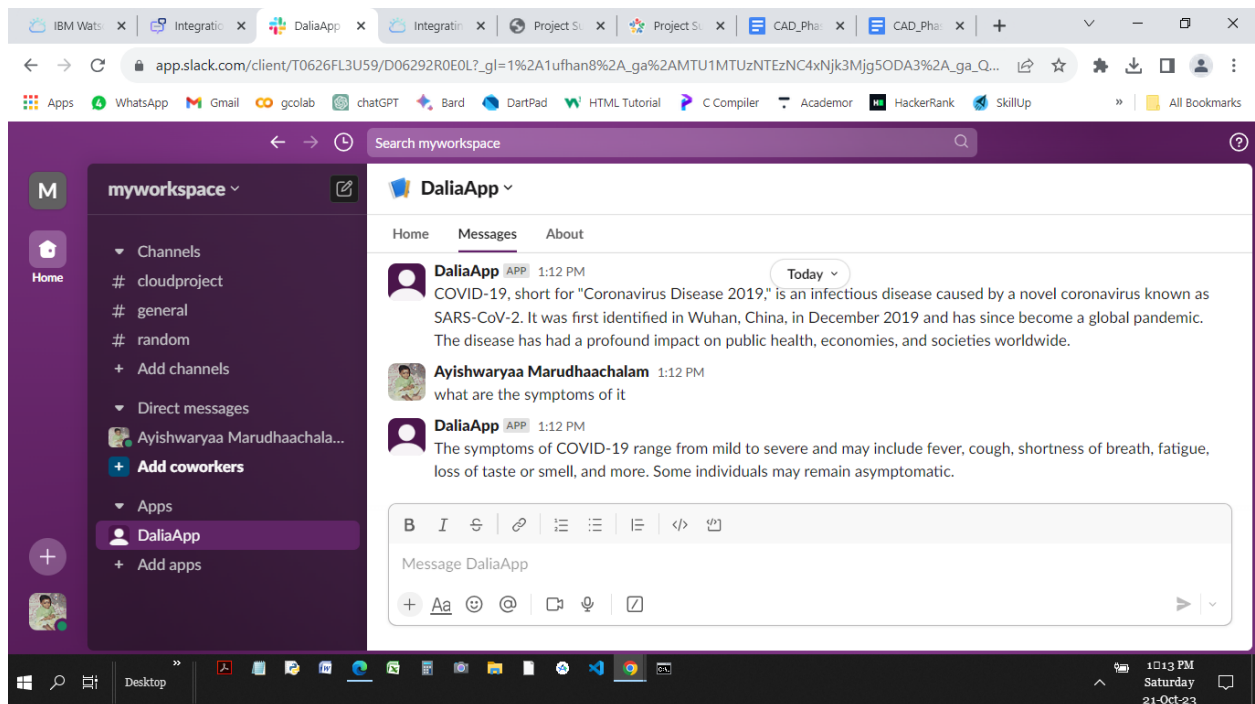
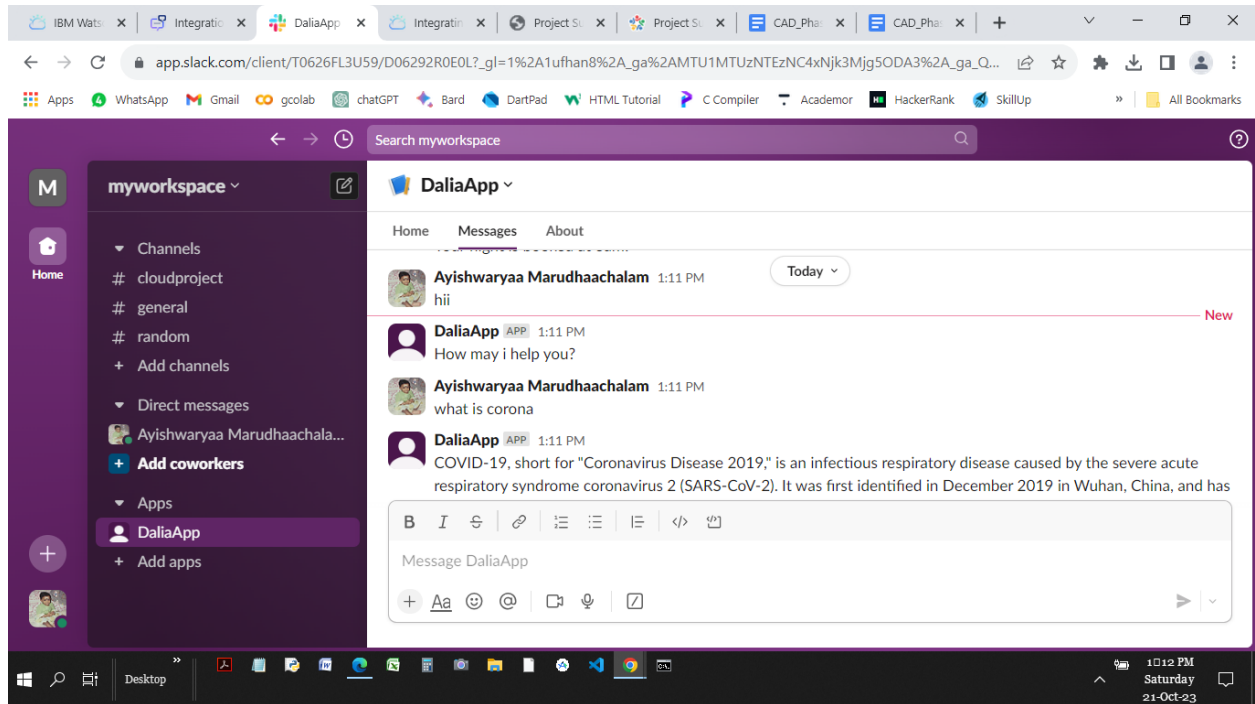
Add Bot User Event

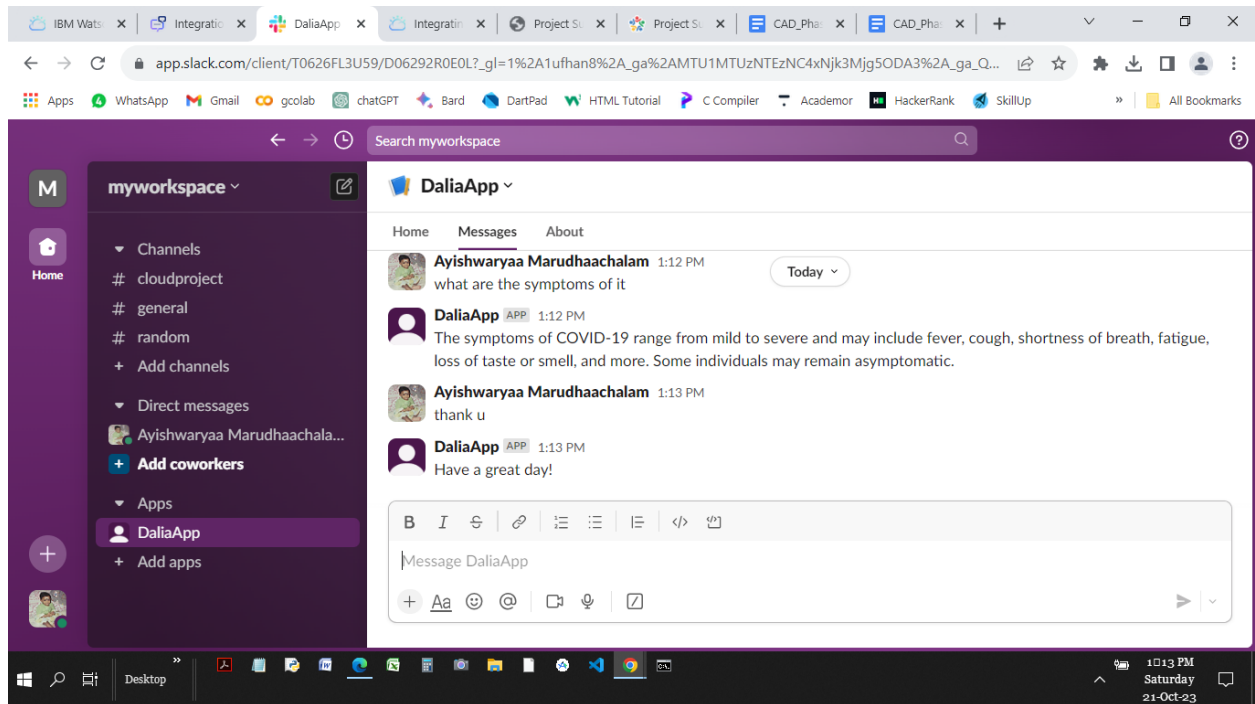
Discard Changes Save Changes

Setting a display name for the app “DaliaApp”



After integrating the chatbot with slack..





Integration with Facebook Messenger:

Steps to integrate chatbot with facebook:

1. First we will create a Facebook App in (<https://developers.facebook.com/>) and create a new app.
2. Next we will set up the basic configuration and we'll note down the App ID and App Secret.
3. Then we will create a Facebook Page.
4. Connecting the Page to the App.
5. Setting up Webhooks
6. Configuring IBM Watson Assistant (creating a chatbot, training it with the necessary dialog and intents)
7. Integrating Watson Assistant
8. Linking the Bot.
9. Finally,Testing the Integration by sending messages to the Facebook Page, which will be handled by the Watson Assistant chatbot.

Creating a facebook app and providing the app secret code to integrate the chatbot with facebook.

The screenshot shows the IBM Watson Assistant web interface. At the top, there's a navigation bar with tabs for 'IBM Watson Servi', 'Integrations | IBM', 'Dalia - App settin', 'Project Submissi', 'CAD_Phase4 - Go', and '(1) Facebook'. Below the navigation bar, there's a search bar and a list of applications including 'Apps', 'WhatsApp', 'Gmail', 'gcolab', 'chatGPT', 'Bard', 'DartPad', 'HTML Tutorial', 'C Compiler', 'Academor', 'HackerRank', and 'SkillUp'. The main content area is titled 'Facebook Messenger' with a 'Draft' label. It includes a 'Close' button and a 'Next' button. The 'Next' button is highlighted in blue. Below the title, there's a progress bar with steps: 'Get started', 'Application secret', 'Page access token', 'Webhook', 'Connect assistant', and 'Application review'. The 'Application secret' step is currently selected. The main text area contains instructions on how to create a Facebook application, including a note about selecting an existing app and a section for the 'Application secret' which is currently masked with dots.

The screenshot shows the Meta for Developers dashboard. At the top, there's a navigation bar with links for 'Docs', 'Tools', 'Support', and 'My Apps'. A search bar is also present. The main content area is titled 'Apps' and features a 'Filter by' section with options: 'All Apps (1)', 'Archived', and 'Required actions'. The 'All Apps (1)' option is selected. Below the filter section, there's a list of apps. The first app listed is 'Dalia', which has an App ID of 840965134699925, is in 'Development' mode, and is of type 'Business'. The app is owned by 'Administrator'. To the right of the app list, there's a 'Recently Used' dropdown menu. At the bottom of the dashboard, there's a footer section with the Meta logo, 'Follow Us' links (Facebook, Instagram, Twitter, LinkedIn, YouTube), and a list of links categorized under 'Products', 'Programs', 'Support', 'News', and 'Terms and Policies'.

Creating a facebook page “chat corner” where the chatbot will respond for user queries

The image shows a two-part process for creating a Facebook page for a chatbot. The top part is a screenshot of the IBM Watson Assistant web interface, and the bottom part is a screenshot of the resulting Facebook page.

IBM Watson Assistant Interface:

- Page Header:** IBM watsonx Assistant | Life | Upgrade | Dalia | Learning resources
- Section:** Facebook Messenger (Draft)
- Progress Bar:** Get started (active) | Application secret | Page access token | Webhook | Connect assistant | Application review
- Section:** Connect to Facebook page
- Instructions:**
 1. On the Facebook app page, click the plus sign (+) next to **Products** in the left side navigation.
 2. Under **Add a product**, find the **Messenger** tile and click **Set Up**.
 3. In the Messenger settings, scroll down to **Access Tokens**.
 4. Click **Add or Remove Pages** and choose the Facebook page you want to use for your app.
 5. Click **Generate Token** and check the **I Understand** checkbox to generate the page access token.
- Note:** If you do not already have a page for your app, click **Create a new page**. After you finish creating the page, return to the Facebook apps page and navigate back to the Messenger settings for your app. You can then select the page you created.

Facebook Page Interface:

- Page Name:** Chat Corner
- Page Type:** Information technology company
- Intro:** chat with me!
- Page Info:** Page · Information technology company, Not yet rated (0 reviews)
- Photos:** See All Photos
- Posts:** No posts available

Providing the access token of the page created to connect watson with facebook.

The screenshot shows the IBM Watson Assistant interface in a web browser. The browser's address bar displays the URL: `eu-gb.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F1bf953e2...`. The page title is "Facebook Messenger" with a "Draft" status. The main content area contains instructions for generating a page access token. It includes a note about creating a new page if one does not already exist, followed by a text input field for the "Page access token". Below this, it states that the "Generated verify token" field contains a token for Facebook verification, with a text input field showing the value: `93f7504-ba73-4cb7-b293-f9a856a56dd`. The interface also features a "Close" button and a "Next" button. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 7:52 PM on Saturday, 21-Oct-23.

The screenshot displays the Meta for Developers interface. The top navigation bar includes links for "Docs", "Tools", "Support", and "My Apps", along with a search bar and a user profile for "Ayishu M". The main content area is titled "Access Tokens" and includes a "Create new Page" button. It provides instructions on how to generate a Page access token, listing two conditions: being a Page admin and having the necessary permissions. A table below shows the details of a generated token for the "Chat Corner" page (ID: 138807869315105). The table has two columns: "Pages" and "Tokens". The "Pages" column lists the page name and ID, and the "Tokens" column shows the status "Token generated" and a "Generate token" button. A blue button labeled "Add or remove Pages" is also visible. The left sidebar contains a navigation menu with options like "Dashboard", "Required actions", "App settings", "App roles", "Alerts", and "App Review". The bottom of the page shows a "Products" section with an "Add Product" link.

Pages	Tokens
Chat Corner 138807869315105	Token generated <button>Generate token</button>

Configuring webhooks (to get inputs from users)

The screenshot shows the 'Configure Facebook webhooks' interface in the IBM Watson Assistant console. The browser's address bar displays the URL: `eu-gb.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F1bf953e2...`. The page has a dark header with 'IBM watsonx Assistant Lite' and navigation links for 'Upgrade' and 'Dalia'. The main content area is titled 'Facebook Messenger' and includes a 'Draft' button. A progress bar at the top of the content area shows steps: 'Get started', 'Application secret', 'Page access token', 'Webhook' (current), 'Connect assistant', and 'Application review'. The 'Webhook' section is titled 'Configure Facebook webhooks' and shows a 'Generated callback URL' in a text box: `https://integrations.eu-gb.assistant.watson.appdomain.cloud/...`. Below this, a note states: 'The generated request URL will be using the latest service API version. Learn more'. A three-step instruction list follows: 1. In the Facebook Messenger settings, scroll to the **Webhooks** section. Click **Setup Webhooks**. 2. In the **Edit Callback URL** window, paste the generated callback URL into the **Callback URL** field. 3. In the **Verify Token** field, paste the verify token that was generated for you earlier. At the bottom of the browser window, the Windows taskbar is visible with the system clock showing 8:03 PM on Saturday, 21-Oct-23.

The screenshot shows the 'Webhooks' configuration page within the Facebook Messenger settings. The left sidebar contains a navigation menu with 'Products', 'Webhooks', 'Messenger' (selected), 'Settings', 'Instagram settings', 'Activity log', and 'Activity log'. The main content area is titled 'Webhooks' and includes the instruction: 'To receive messages and other events sent by Messenger users, the app should enable webhooks integration.' It features two input fields: 'Callback URL' with the value `https://integrations.eu-gb.assistant.watson.appdomain.cloud/public/face...` and 'Verify token' with a masked value. Below these fields, a note states: 'Validation requests and Webhook notifications for this object will be sent to this URL.' and 'Token that Meta will echo back to you as part of callback URL verification.' There are buttons for 'Edit callback URL' and 'Show recent errors'. A table below lists the configured webhooks for the 'Chat Corner' page (ID: 13887869315105). The table has two columns: 'Pages' and 'Webhooks'. The row shows '2 fields: messages, messaging_postbacks' and an 'Edit' button. At the bottom of the table is a link: 'Add or remove Pages'. The Windows taskbar at the bottom shows the system clock at 8:57 PM.

Pages	Webhooks
Chat Corner 13887869315105	2 fields messages, messaging_postbacks

Finally connecting the watson assistant chatbot with the facebook page created..

The screenshot shows a web browser window with multiple tabs. The active tab is titled "eu-gb.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F1bf953e2...". The browser's address bar and tabs are visible at the top. Below the browser window, a dark taskbar shows various application icons. The main content area of the browser displays the "Facebook Messenger" integration page. At the top of this page, there are buttons for "Close" and "Next". Below these, a progress bar indicates the current step is "Connect assistant". The main heading is "Connect your assistant". A note states: "Note: Subscribe to only one page. Multiple-page subscriptions are not currently supported." Below the note, a message says: "Congrats! You should be able to test your bot now in Messenger. You must be a Page admin to interact with it until you make it public."

The screenshot shows the same web browser window as the previous one, but the active tab now displays the "Application review" step of the integration process. The progress bar at the top of the page indicates that "Application review" is the current step. The main heading is "Start app review process". Below the heading, a message states: "Once you have confirmed your bot is working in Messenger, and you would like to make the bot public, go to [Submitting Your Messenger App](#) on Facebook for Developer's site and follow the instructions to start the app review process."

Testing the chatbot's responses for user inputs.

