

# TASK P – 5.1 C

LUIS and QnA Maker

Evolution and hereditary  
QnA app

Target - C

Saurabh Dharmadhikari  
[Saurabh.dharma01@gmail.com](mailto:Saurabh.dharma01@gmail.com)

## Introduction:

We are required to build bot question and answering chat bot and link it to Facebook messenger. The domain on which we are working is organic evolution. Custom built bot is built using Azure Cognitive Services Language Studio, it will be deployed to Azure and attached to Facebook Messenger.

This bot will be able to answer questions related to evolution and how hereditary traits are transferred from one generation to other. It is a simple bot with limited data base on evolution and much complicated bots can be built using the same tools. This is to demonstrate how these QnA bots are built. Building process is explained step by step and attached to other applications for public use.

## Step 1: Creating a language resource group

The screenshot shows the Microsoft Azure portal homepage. At the top, there's a search bar and a user profile for 's223624833@deakin.edu.. DEAKIN UNIVERSITY (DEAKIN36..)'. Below the header, the 'Azure services' section features several icons: 'Create a resource', 'Resource groups', 'Language' (which is circled in blue), 'All resources', 'Cognitive Services', 'Azure Machine Learning', 'Subscriptions', 'Cost Management ...', and 'Cost analysis (preview)' followed by a 'More services' link. Under the 'Language' icon, there's a table showing a single item: 'Azure for Students' (Subscription type, last viewed 2 days ago). The 'Resources' and 'Tools' sections are also visible.

Open Azure portal and log in to your account then search language and select it.

The screenshot shows the 'Cognitive Services | Language service' page within the Azure portal. The left sidebar lists categories like 'Overview', 'All Cognitive Services', 'Azure OpenAI', 'Speech', 'Language' (which is selected and has a grey background), and 'Vision'. The main content area displays a message: 'No language to display. Try changing or clearing your filters.' Below this, there's a large 'Create language' button, which is highlighted with a blue box. Other buttons include 'Learn more' and 'Give feedback'.

Click on create language in language service provided by Azure.

Default features

- Sentiment analysis
- Key phrase extraction
- Pre-built question answering
- Conversational language understanding
- Named entity recognition
- Text Summarization
- Text analytics for Health

Custom features

- Custom question answering**  
Use this feature to answer user's questions over your data corpus. Requires Azure Cognitive Search. [Learn more](#).
- Custom text classification & Custom named entity recognition** ⓘ  
Use this feature for custom text classification or custom named entity extraction. Requires Azure Storage. [Learn more](#).

[Continue to create your resource](#)

We select default settings for our bot but if required we can select custom features. We continue to create our resource.

Project Details

Subscription \* ⓘ Azure for Students

Resource group \* ⓘ (New) LanguageQnA [Create new](#)

Instance Details

Region ⓘ East US

Name \* ⓘ LanModu

Pricing tier \* ⓘ Free F0 (5K Transactions per 30 days)

[View full pricing details](#)

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it.

[Responsible Use of AI documentation for Text Analytics for Health](#)

[Responsible Use of AI documentation for PII](#)

[Responsible Use of AI documentation for Language](#)

By checking this box I certify that I have reviewed and acknowledge the terms in the Responsible AI Notice. \*

[Review + create](#) [< Previous](#) [Next: Network >](#) [Give feedback](#)

We fill in all the required details selecting region as East US and pricing tier Free and click on next.

Once validation is passed, we are ready to create this resource group.

This will complete the deployment and creation of resource group. When we click on go to resource it will take us to the language resource that is created from where we can go to Language Studio.

Microsoft Azure

Home > TextAnalyticsCreate-20230430125036 | Overview >

**LanModu** Language

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource Management

Features

Keys and Endpoint

Encryption

Pricing tier

Networking

Identity

Cost analysis

Properties

Locks

Monitoring

Alerts

Metrics

Diagnostic settings

Logs

Automation

Tasks (preview)

Resource group (move) : LanguageQnA

Status : Active

Location : East US

Subscription (move) : Azure For Students

Subscription ID : 351fd358-bda7-4287-bda4-efb3375f0ab6

Tags (edit) : Click here to add tags

API type : Language

Pricing tier : Free

Endpoint : https://lanmodu.cognitiveservices.azure.com/

Manage keys : Click here to manage keys

Get Started Discover Develop Deploy

Text Analytics has been rebranded and incorporated into Azure Cognitive Service for Language. [Learn More](#)

**Get Started with Language service**

Azure Cognitive Service for Language enables you to build apps with industry-leading natural language understanding capabilities without machine learning expertise. [Learn More](#)

**Discover** **Develop** **Deploy**

**Language Studio**

Language Studio is a set of UI-based tools that lets you explore, build, and integrate features from Azure Cognitive Service for Language into your applications.

[Get started with Language Studio](#) [Get started with Language Studio](#)

<https://go.microsoft.com/fwlink/?linkid=2194422>

## Step 2: Create Azure search resource in Language Studio

Click on *Get started with Language Studio* to launch Language Studio.

Cognitive Services | Language Studio

Welcome to Language Studio

Recent custom projects you've worked on

Featured Extract information

Post call transcription and analytics (Preview)

Batch transcribe call center recordings and extract valuable information such as Personal Identifiable Information (PII), sentiment, and call summary.

Try it now

Select an Azure resource

To access Language studio, you need an Azure resource. You can use an existing resource or create a new one. You can change the resource you use at any time. [Learn more about resources in Azure](#).

Azure directory \* Deakin University

Note: Switching directory will cause the page to refresh.

Azure subscription \* Azure for Students

Select an existing Azure subscription or [create a free account](#) and then refresh

Resource type \* Language

Language or Azure Cognitive Services resources can be used for any capability except for translation, which requires a Translator resource

Resource name \* LanModu

Select an existing resource or create a new one

Pricing tier: Free (F0)

Managed identity: Enabled

Create a new Language resource

Done Cancel

Log into Language Studio and select the resource we just created and press done to select this as our resource.

Welcome to Language Studio

Recent custom projects you've worked on

You don't have any recent projects yet. Start with one of the custom capabilities to create a new project. The list of recent projects you've worked on will then appear here.

**Create new**

Featured   Extract information   Classify text   **Understand questions and conversational language**   Summarize text   Translate text

Retrieve the most appropriate answer to questions using question answering (CQA) or classify intents and extract entities for conversational utterances using conversational language understanding (CLU). Use orchestration workflow to create one project that routes queries between multiple CQA and CLU projects. [Learn more about understanding conversational language.](#)

**Answer questions**  
Use the prebuilt question answering API to get answers to questions over unstructured text.  
[Try it out](#)

**Custom question answering**  
Next generation of QnAMaker  
Customize the list of questions and answers extracted from your content corpus to provide a conversational experience that suits your needs.  
[Open Custom question answering](#)

**Conversational language understanding**  
Next generation of LUIS  
Classify utterances into intents and extract information with entities to build natural language into apps, bots, and IoT devices.  
[Open Conversational language understanding](#)

**Orchestration workflow**  
Connect and orchestrate CLU, Custom question answering & LUIS projects together in one single project.  
[Open Orchestration workflow](#)

Select **Understand questions and conversational language**, under it we can now select **Open Custom question answering**.

language.cognitive.azure.com/questionAnswering/projects

Cognitive Services | Language Studio

Language Studio > Custom question answering

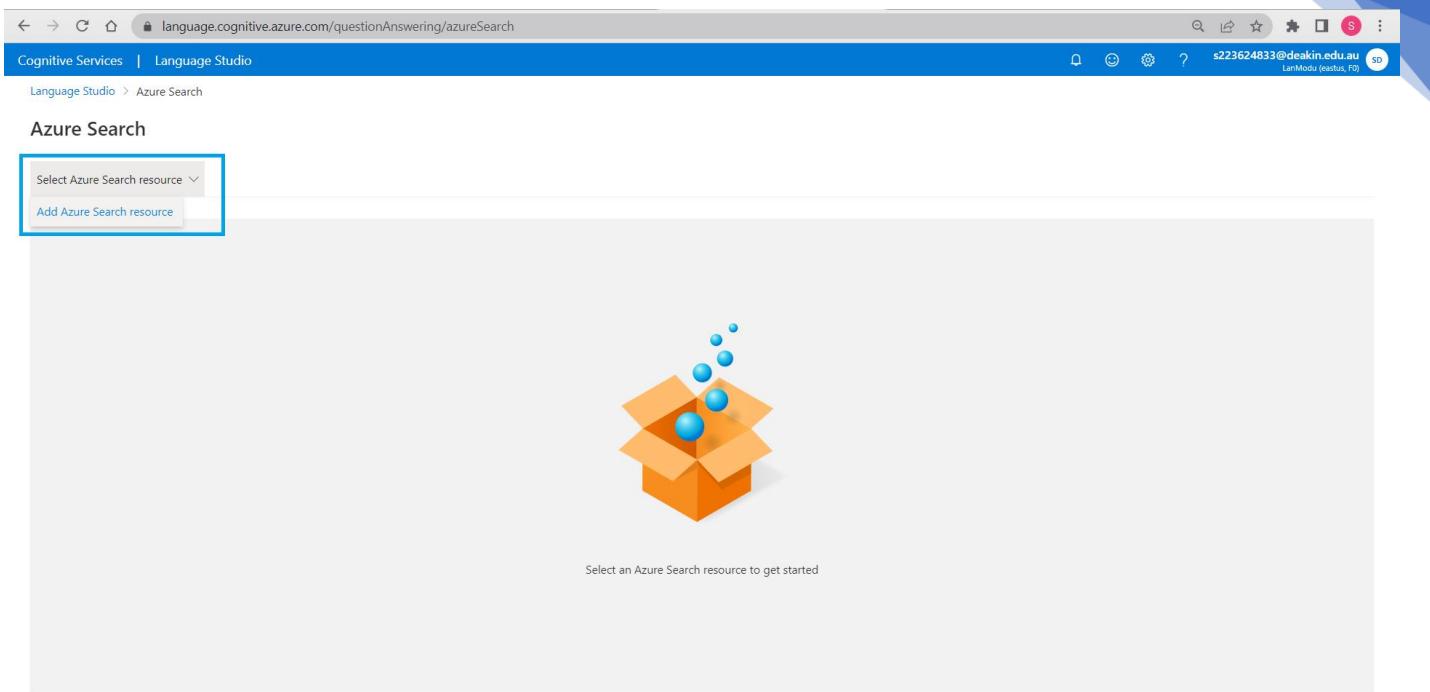
Select the project you want to work with

+ Create new project   Export   Import   Delete   0 items in list   Search projects...

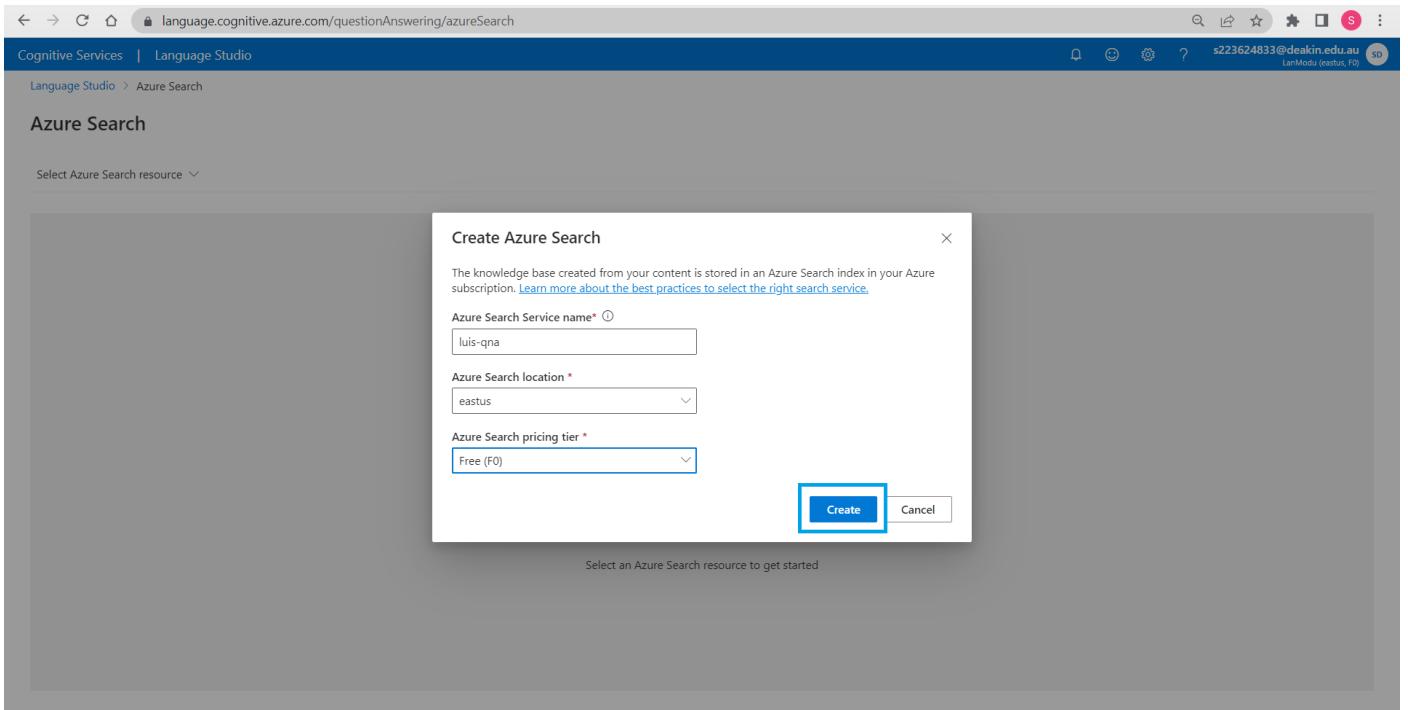
**Start by connecting Azure search to the language resource**

To create a custom question answering project and add data sources to it, you would need to link an Azure search resource to the language resource. [Connect to Azure search](#). After connecting your resource to Azure Search, please refresh this page to continue with project creation.

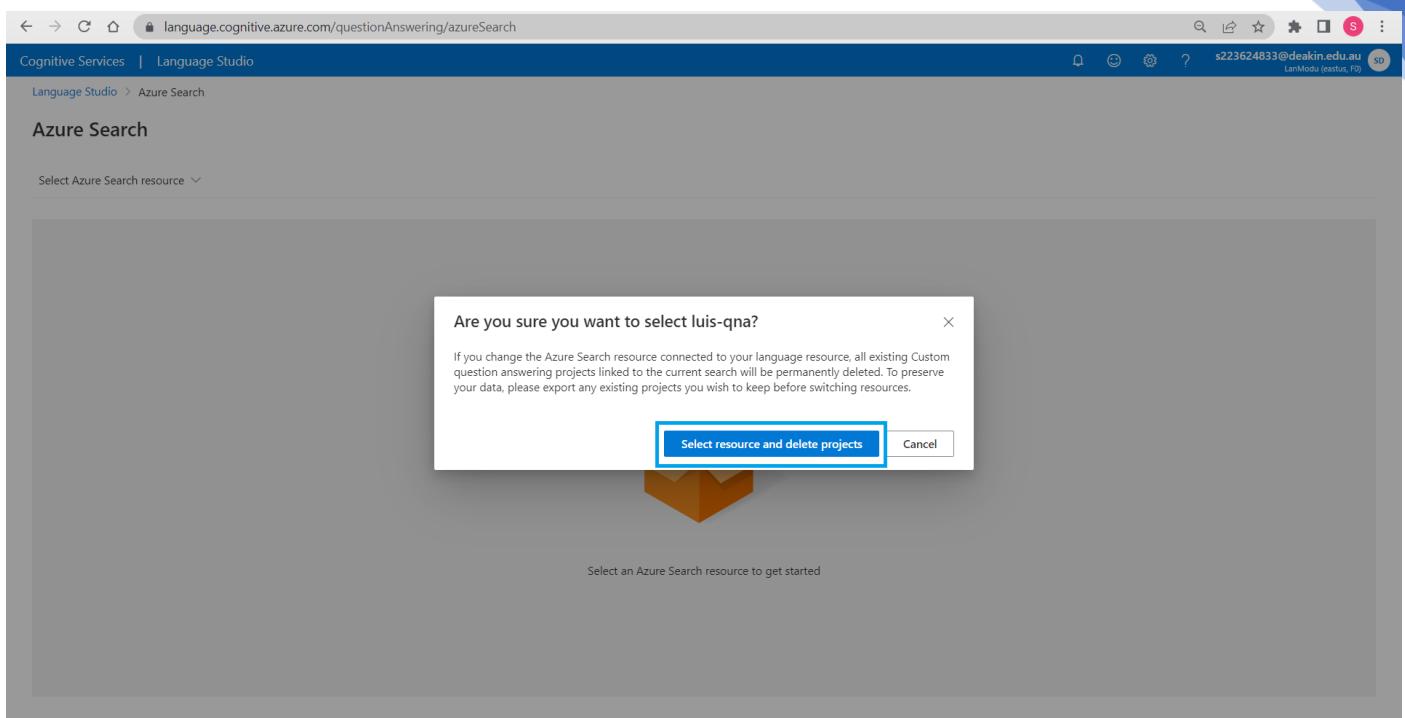
Now go to **Connect to Azure search**.



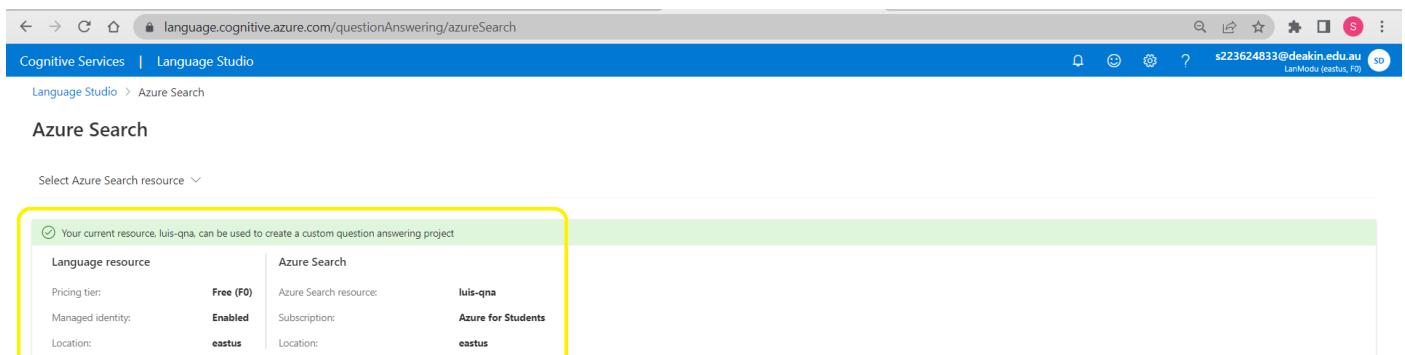
Now we can *Add Azure Search resource*.



Filled required details and create Azure Search resource.

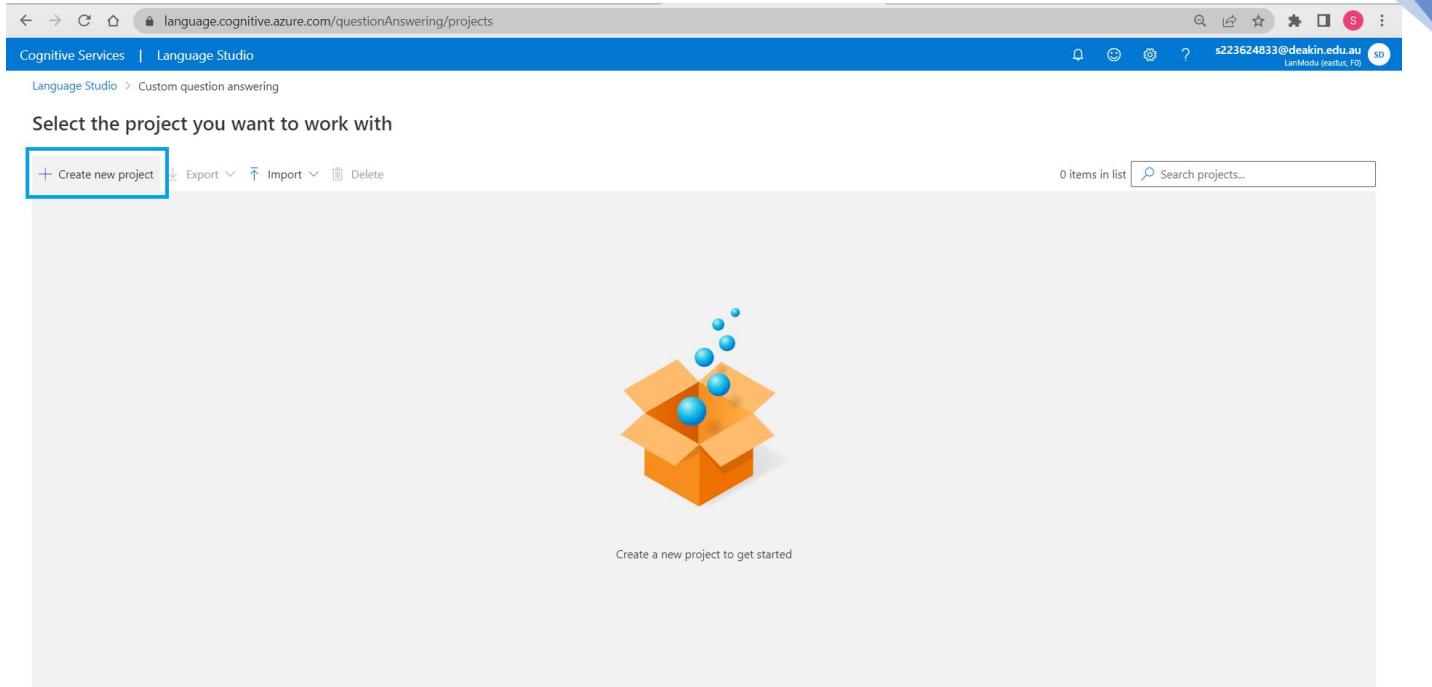


Go ahead with *Select resource and delete projects*.

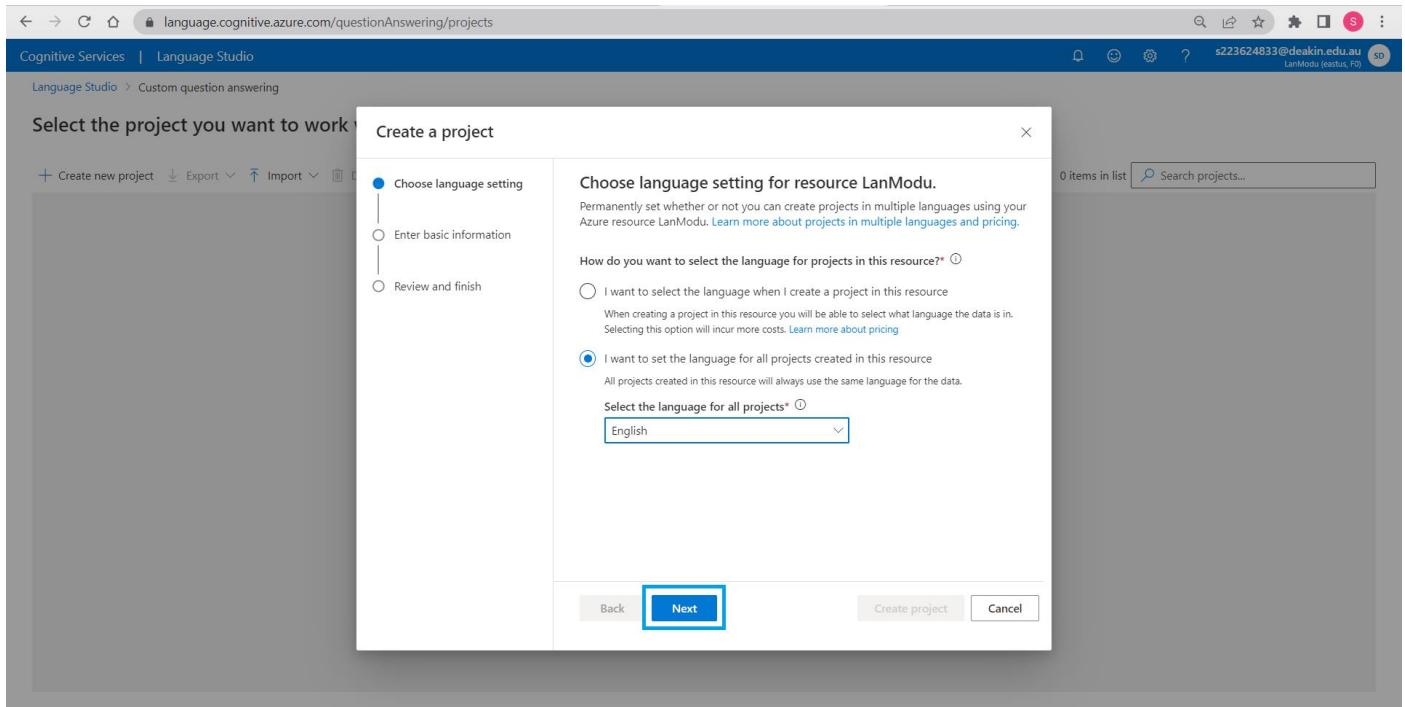


A screen like above screen shot will be displayed once created.

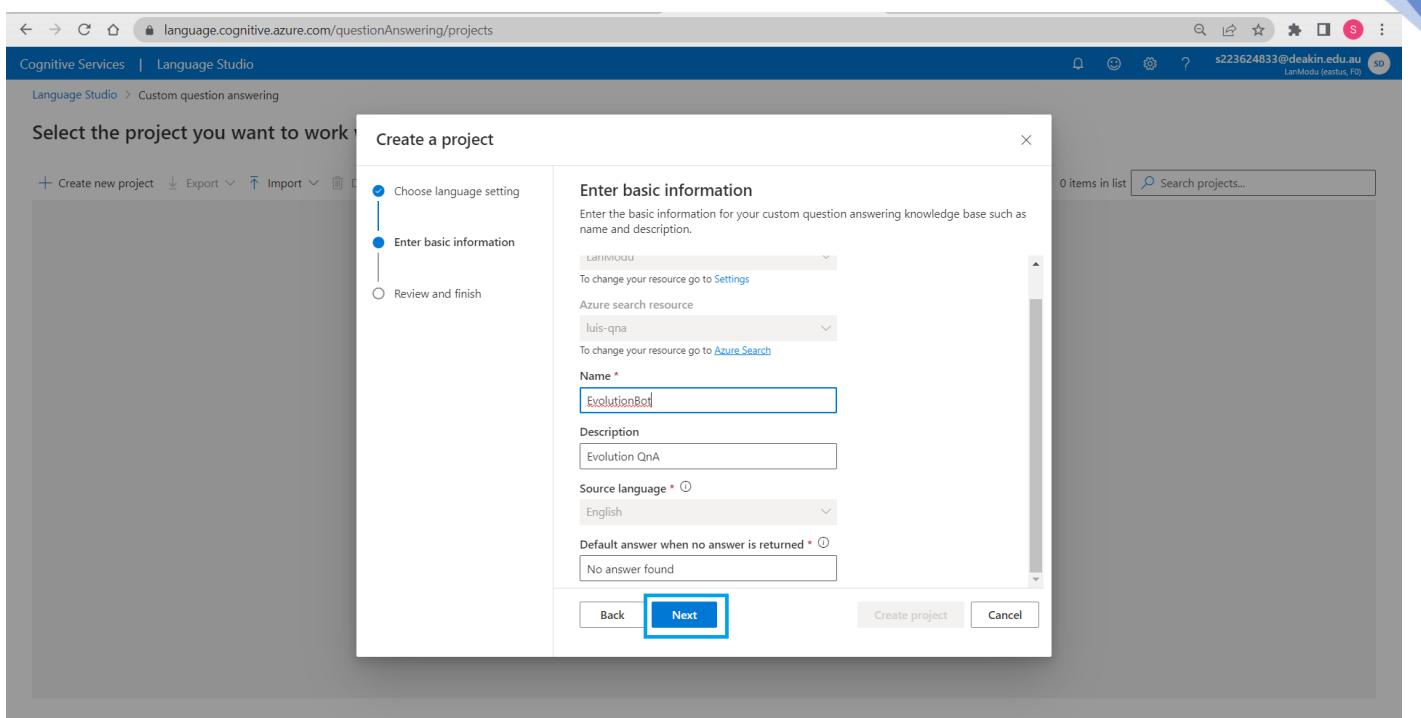
## Step 3: Creating a new project



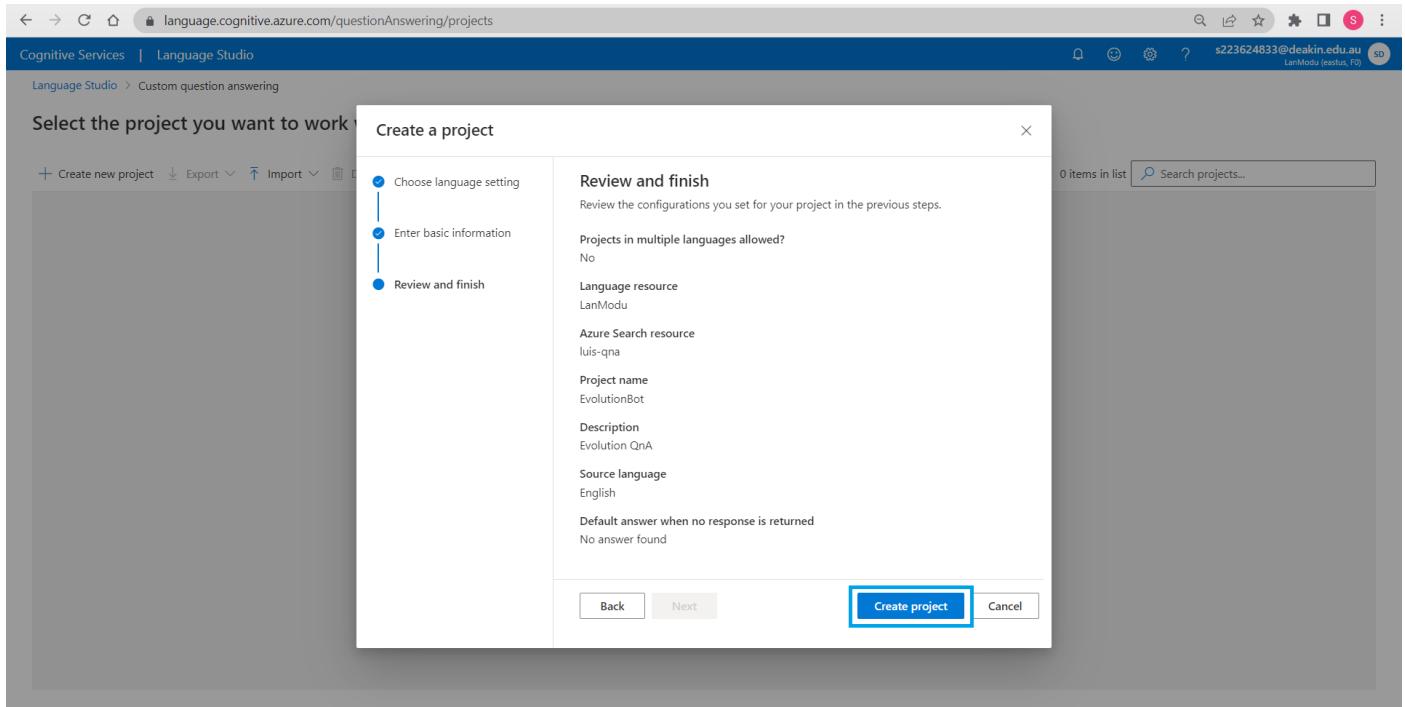
Go back to Language Studio and create a new project.



There are many options for language but we have selected English and we want to set English for all the projects created in this resource.



When we click on next, we will have to fill in some basic information about the project we are creating like its name and description and we are setting default answer as *No answer found*.



Review the information again and create the project.

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

### Manage sources

+ Add source ▾ Edit name Refresh URL Delete

Add sources to your knowledge base

Extract question-and-answer pairs from an online FAQ, product manuals, or add unstructured content to enable search over text. Supported formats are .tsv, .pdf, .doc, .docx, .xlsx. [Learn more about knowledge base sources.](#)

**Notifications**

Dismiss All

- Loading project completed 1:18 PM  
Project loaded successfully

Project is created and we have more options on the left-hand side of the screen.

#### Step 4: Adding source to our project, testing and deployment

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

### Manage sources

+ Add source ▾ Edit name Refresh URL Delete

0 items in list Filter

URLs

Files

Chitchat

Add sources to your knowledge base

Extract question-and-answer pairs from an online FAQ, product manuals, or add unstructured content to enable search over text. Supported formats are .tsv, .pdf, .doc, .docx, .xlsx. [Learn more about knowledge base sources.](#)

We now go to add source and add file.

language.cognitive.azure.com/questionAnswering/projects/EvolutionBot/manageSources

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

Manage sources

+ Add source ▾ Edit name Refresh URL Delete

0 items in list Filter

Add files

You can add up to 10 files at a time. Classification of file structure is auto-detected by default, but you can set your own file structure classification if preferred.

Source name ↓	File name ↓	Classify file structure ↓
Evolution and heredity	Heredity and evolution_compressed.pdf	Auto-detect

Add all Cancel

Click Add file and browse in local drive to open it and give a name to source. We have christened our source as Evolution and heredity, it is a pdf file. You may have multiple source file. Now click add all.

language.cognitive.azure.com/questionAnswering/projects/EvolutionBot/manageSources

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

Manage sources

+ Add source ▾ Edit name Refresh URL Delete

Source ↓	Source name ↓	Unstructured ↓	Source type ↓
Heredit and evolution_compressed.pdf	Evolution and heredity	No	file

Adding sources... 12:45 PM

Our file source is added.

language.cognitive.azure.com/questionAnswering/projects/EvolutionBot/manageSources

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

### Manage sources

+ Add source ▾ Edit name Refresh URL Delete

	Source name	Unstructured	Source type
URLs			
Files			
Chitchat	Evolution and heredity	No	file

We are also adding a chit chat source available in Language Studio to make small talk to give it a better experience to the user.

language.cognitive.azure.com/questionAnswering/projects/EvolutionBot/manageSources

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

### Manage sources

+ Add source ▾ Edit name Refresh URL Delete

	Source type
Source	Unstructured
Heredity and evolution_compressed.pdf	No

**Add chit chat**

When you select a chit chat personality, we add a curated list of question and answer pairs to the knowledge base. [Learn more](#)

Chit chat is currently available in the following languages: Chinese Simplified, English, French, German, Italian, Japanese, Korean, Portuguese, and Spanish.

Professional  
 Friendly  
 Witty  
 Caring  
 Enthusiastic

**Add chit chat** Cancel

There are multiple options to choose from. We have selected *Friendly* and added chit chat.

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Manage sources

Manage sources

+ Add source ▾ Edit name Refresh URL Delete

Source name ▾ Unstructured ▾ Source type ▾

Source name	Unstructured	Source type
qna_chitchat_Friendly	No	file
Heredity and evolution_compressed.pdf	No	file

Friendly chit chat and file source are added now. These will serve as database for our bot to extract info when questions are asked.

## Step 5: Editing and testing project created

Cognitive Services | Language Studio

Language Studio > Custom question answering > EvolutionBot - Edit knowledge base

Question answer pairs (132) Synonyms (0)

+ Add Test Delete Save Import/Export Discard

I'm so bored of you  
Swing and a miss.

Are you a woman?  
That doesn't really apply to me.

Who's older?  
I don't really have an age.

Will you ask me anything about myself?  
I'm a much better answerer than asker.

Do you have fingers?  
I don't have the hardware for that.

Do you have a boss?  
I'm here for you!

Can you jump rope?  
That's not one of my talents

Search pairs

Edit knowledge base

0 unstructured sources and 2 structured sources. [View sources](#)

I'm so bored of you

Source: qna\_chitchat\_Friendly

Enable rich text Show context tree

Answer

Edit answer

Swing and a miss.

Alternate questions (112)

Follow up prompts (0)

Metadata (1)

Go to *Edit knowledge base* and check the question answer pairs created by custom Azure, we can edit questions to add similar questions. We now click *Test* to check project that is created.

Here we test the project created by asking question related to evolution and testing chit chat source by giving inputs like hi or how are you. If it is replying correctly we can assume that it is fit to go ahead otherwise we can make relevant changes.

## Step 6: Deployment of the project

Now we go to Deploy knowledge base and click on deploy.

The screenshot shows the Azure Language Studio interface for deploying a knowledge base. The left sidebar lists options like Language Studio, Custom question answering, Azure Search, EvolutionBot, Manage sources, Edit knowledge base, Deploy knowledge base (which is selected), Review suggestions, and Project settings. The main content area shows a success message: "Your knowledge base is now deployed. You can get your prediction URL or create a bot." Below this, the "Knowledge base status" section displays the following details:

State	Deployed	Resource
①	<input checked="" type="checkbox"/>	LanModu
Deployment Date	4/30/2023	Location
①	<input checked="" type="checkbox"/>	eastus
Deployment Time	2:55:53 PM	Tier
①	<input checked="" type="checkbox"/>	Free (F0)

Below the status, there's a "Next steps: Create a bot" section with two steps: "Step 1: [Read the documentation](#) to learn more about creating bots." and "Step 2: Go to Azure to create a bot." A blue "Create a bot" button is at the bottom.

We will be able to see the knowledge base status details after successful deployment and we can go ahead to *Create a bot*.

## Step 6: Create a bot

This screenshot shows the "Custom deployment" step in the Azure portal. The form is divided into several sections:

- Project details:** Includes fields for "Subscription" (set to "Azure for Students") and "Resource group" (set to "LanguageQnA").
- Instance details:** Includes "Resource group location" (set to "(US) East US") and "Azure Bot" (set to "LanModu-bot").
- Choose your pricing tier:** Shows "Standard" selected for the pricing tier.
- Microsoft App ID:** Shows "Create new User-assigned managed identity" selected for creation type.

At the bottom, there are "Review + create" and "Next : Web App >" buttons.

Again, fill in the necessary details and review and create.

The screenshot shows the Microsoft Azure Deployment Details blade for a deployment named "Microsoft.Template-20230430152619". The main message is "Your deployment is complete". Deployment details include the name, subscription (Azure for Students), resource group (LanguageQnA), start time (4/30/2023, 3:25:19 PM), and correlation ID (ca79674d-1097-4679-91d7-2fb4ad4b652d). A "Go to resource group" button is present. On the right, there are promotional cards for Cost Management, Microsoft Defender for Cloud, Free Microsoft tutorials, and Work with an expert.

Bot is created and deployment completed now we can go to resource group to see the bot we have created.

## Step 7: Testing our bot in Azure resource

The screenshot shows the Azure Resource Group overview page for the "LanguageQnA" resource group. The "Essentials" section displays the subscription (Azure for Students), subscription ID (351fd358-bda7-4287-bda4-efb3375f0ab6), and location (East US). The "Resources" section lists the "LanModu-bot" resource, which is highlighted with a blue border. The "Properties" table provides detailed information about the bot:

Type	Value	Location	Actions
Language	East US	Global	...
Azure Bot		Global	...
Managed identity		East US	...
App Service		East US	...
App Service plan		East US	...
Search service		East US	...

Select the bot we have created in the resource group.

LanModu-bot | Azure Bot

**Overview**

Activity log Access control (IAM) Tags

**Settings**

- Bot profile Configuration Channels Pricing
- Test in Web Chat
- Encryption Networking Properties Locks
- Monitoring
- Conversational analytics Alerts Metrics Diagnostic settings Logs
- Automation Tasks (preview)
- Export template

Help us improve Bot Service. Take our survey!

**Essentials**

Resource group (move) : LanguageQnA  
Subscription (move) : Azure for Students  
Subscription ID : 351fd358-bda7-4287-bda4-efb3375f0ab6  
Tags (edit) : WebAppLinked : true

Bot Service pricing tier : F0  
Messaging endpoint : https://LanModu-bot-Evolution.azurewebsites.net/api/messages  
Location : Global

**Build enterprise-grade conversational AI**

Develop enterprise-grade, conversational AI experiences, while maintaining control of your data. Edit your bot in SDK, host your bot in any environment and enable customers from a variety of channels, such as your app or website, Direct Line Speech, Microsoft Teams and more. [Learn more](#)

**Get started with the Bot Framework**  
Build, test, deploy, and manage intelligent bots, all in one place. The Bot Framework includes a comprehensive set of tools for building bots, as well as tools, templates, and related AI services. Note: For User-Assigned Managed Identity and Single Tenant app, BotFramework SDK (C# or Javascript) version 4.15.0 or higher is needed for these app types. [Learn the Bot Framework](#)

**Test and refine your bot**  
Refined and debug locally with Emulator, and test your bot online in Web Chat. Learn more about testing and debugging.

**Publish to Azure**  
When you are ready, publish your bot to Azure and connect to channels. Learn about publishing directly to Azure and continuous deployment.

**Connect to channels**

**Download bot source code**

Select **Test in Web Chat** to open testing space to test our bot.

LanModu-bot | Test in Web Chat

Start over

hi

A minute ago

evolution

A minute ago

Hi!

A minute ago

hello

A minute ago

Hi!

A minute ago

how are you

A minute ago

I'm doing great, thanks for asking!

A minute ago

Type your message ➤

We test our bot and as it is giving relevant answers, we are good to go ahead.

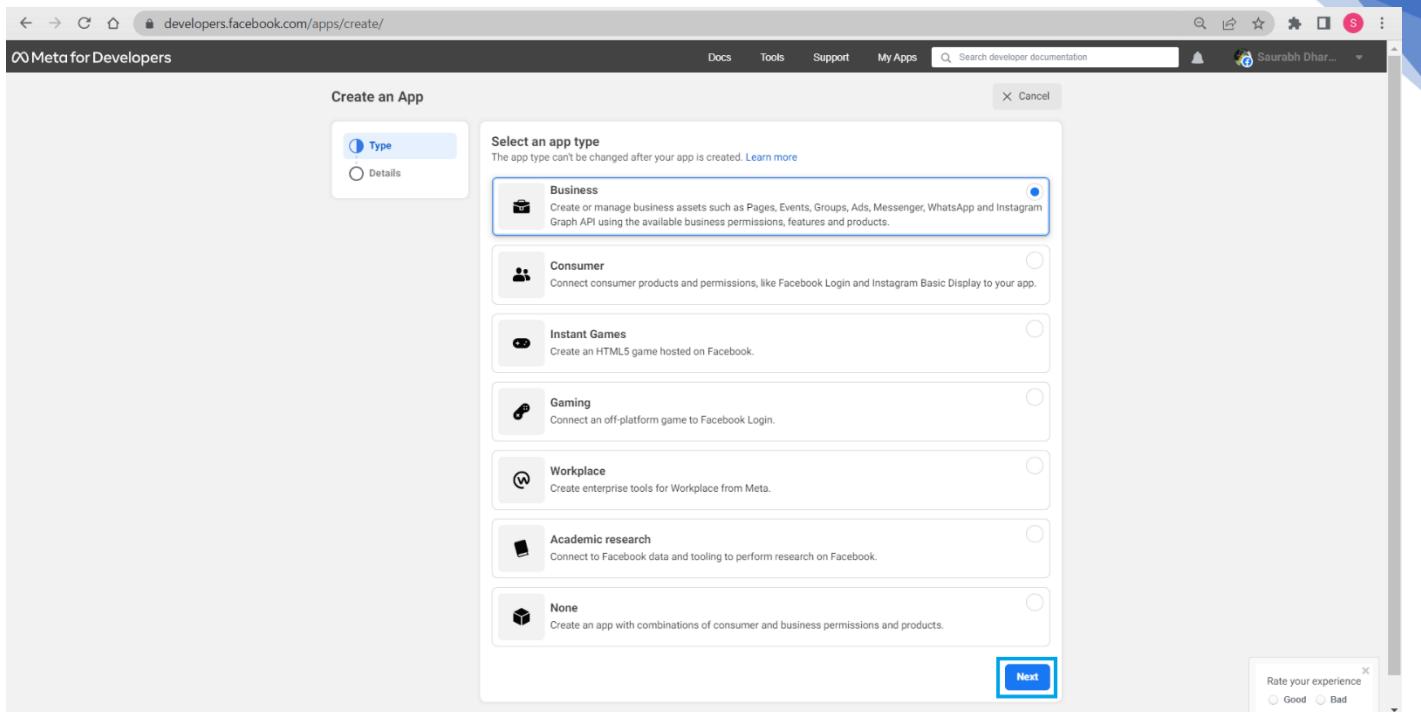
## Step 8: Creating an app on Meta for Developers

The screenshot shows the 'Meta for Developers' website at [developers.facebook.com/apps/](https://developers.facebook.com/apps/). The top navigation bar includes links for Docs, Tools, Support, and My Apps, along with a search bar and user profile information. On the left, a sidebar titled 'Apps' has a 'Filter by' section with 'All Apps' selected (indicated by a blue background) and 'Archived' as an option. The main content area features a large illustration of a person working on a computer. Below the illustration, the text 'No apps yet' is displayed, followed by the subtext 'To get started, create your first app.' A prominent green 'Create App' button is centered in this area.

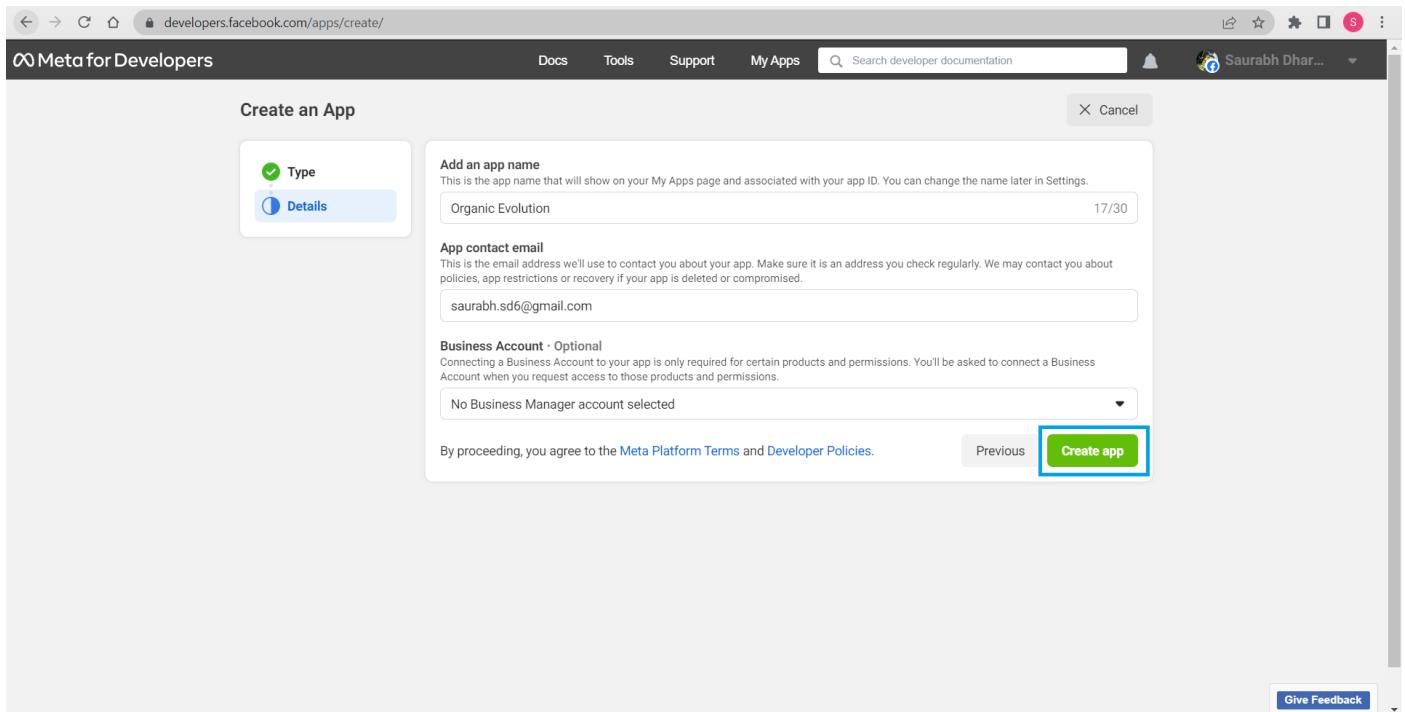
Go to my apps in Meta for Developers and press *Create App*.

The screenshot shows the 'Create an app' wizard at [developers.facebook.com/apps/creation/](https://developers.facebook.com/apps/creation/). The title bar says 'Create an app'. The main question is 'Which use case do you want to add to your app?'. It provides two options: 'Set up Facebook Login' (with a blue 'f' icon) and 'Other' (with a blue building icon). The 'Other' option is selected, indicated by a blue dot next to its name. A 'Learn More' link is also present under 'Set up Facebook Login'. At the bottom right of the wizard, a blue 'Next' button is visible.

Select other and then next.



Select business from the various options available.



Enter the name of the app that you may wish and a contact email address and then create app.

The screenshot shows the Facebook Developers Dashboard for an app named "Organic Evolution". The dashboard includes sections for "Required actions", "Application Rate Limit" (0% used), "User Rate Limit" (0 users throttled), and "Add products to your app" (options for App Events, Audience Network, and Fundraisers). The left sidebar lists various developer tools and logs.

Now we can check app created in our dashboard.

The screenshot shows the "Advanced" settings page for the "Organic Evolution" app. It includes fields for "Client token" (4afa216b8f0ac41e49e2b05ae5edda16), "Require app secret" (disabled), "Require 2-factor reauthorization" (disabled), "Allow API Access to app settings" (enabled), "Domain manager" (no domains set), and "App Page" (name field empty). Buttons for "Discard" and "Save changes" are at the bottom.

Check advanced settings enable *Allow API Access to app settings*, save changes.

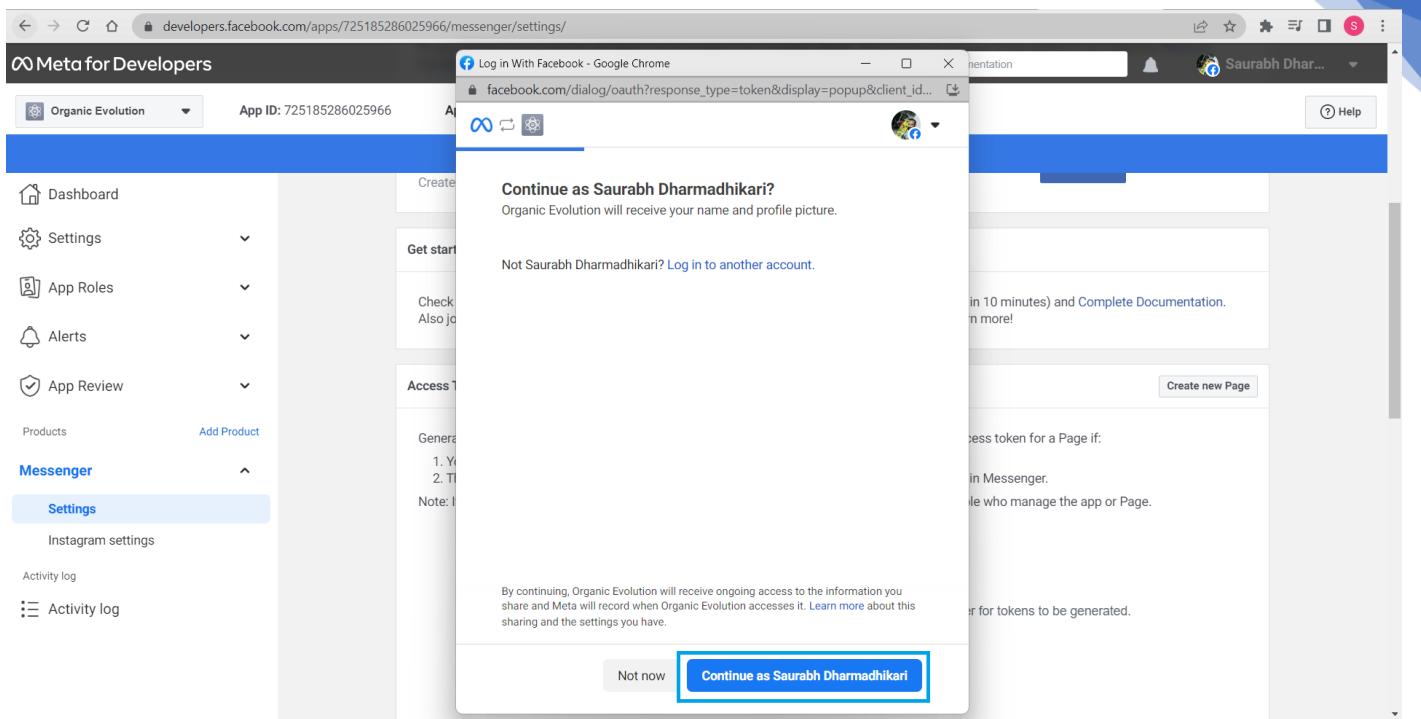
The screenshot shows the Facebook Developers Basic Settings page. On the left, there's a sidebar with options like Dashboard, Settings (which is selected), Basic, Advanced, App Roles, Alerts, App Review, Products, Activity log, and Activity log. The main area displays app configuration fields. Two fields are highlighted with yellow boxes: 'App ID' containing '725185286025966' and 'App secret' containing '\*\*\*\*\*'. Other visible fields include 'Display name' (Organic Evolution), 'Namespace', 'App domains', 'Contact email' (saurabh.sd6@gmail.com), 'Privacy Policy URL', 'Terms of Service URL', 'Category' (Choose a category), and an 'App icon (1024 x 1024)' placeholder. At the bottom right are 'Discard' and 'Save changes' buttons.

Under basic settings we can find App ID and App secret. Do not share App secret with anyone and copy it and keep it safe.

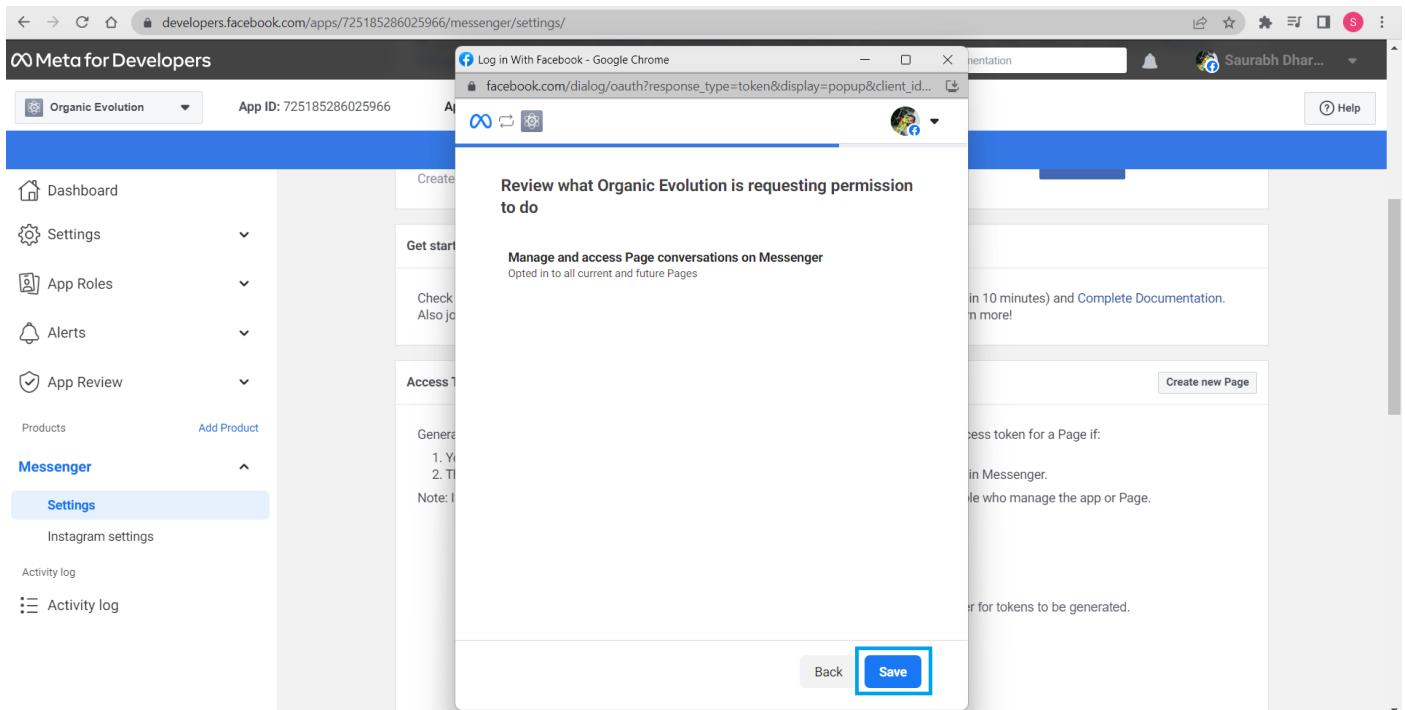
## Step 9: Messenger setup

The screenshot shows the Facebook Developers Messenger Settings page. The sidebar includes Dashboard, Settings, App Roles, Alerts, App Review, Products, Add Product, and Messenger (which is selected). Under Messenger, 'Settings' is selected. The main area has two sections: 'Access Tokens' and 'Webhooks'. The 'Access Tokens' section contains instructions to generate a Page access token for platform APIs, mentioning page admins and permissions. It also notes that tokens can still be generated in dev mode. A 'No page permissions granted' message with a 'Add or remove Pages' button is present. The 'Webhooks' section explains the need for webhook integration and has an 'Add Callback URL' button.

Once messenger setup is done, we can check the settings and add or remove pages for the app developed.



Continue with your account ID.



Select only for this page option and save the changes.

This app is now using Facebook Login for Business. X

Check out some resources that may help you in development: [Quick Start](#) (build a Messenger app in 10 minutes) and [Complete Documentation](#). Also join our [Messenger Platform Developers Community](#) on Facebook to get latest news and learn more!

### Access Tokens

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Pages ↑	Tokens
Organic evolution 102336336191938	Token generated

[Add or remove Pages](#) ⓘ

[Create new Page](#)

### Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

[Add Callback URL](#)

Scroll down to access token and generate token. Copy the token generated and keep it safe.

## Step 10: Attaching Facebook app to azure

Microsoft Azure

LanModu-bot | Channels

Overview Activity log Access control (IAM) Tags

Settings Bot profile Configuration Channels Pricing

Text in Web Chat Encryption Networking Properties Locks Monitoring Conversational analytics Alerts Metrics Diagnostic settings Logs Automation Tasks (preview)

Search resources, services, and docs (G+)

Connect the bot with channels. Learn more ⓘ

Channel Details

- Alexa Alexa Channel
- Communication Services - Chat Communication Services - Chat Channel
- Direct Line Speech Direct Line Speech Channel
- Email O365 Email Channel
- Facebook** Support for Text Messaging via Facebook
- GroupMe GroupMe Channel
- LINE Support for LINE Channel
- Microsoft 365 Enable message extensions in Outlook, and Microsoft 365 apps
- Microsoft Teams Microsoft Teams Channel
- Omnichannel Omnichannel Channel
- Outlook Outlook Channel
- Skype Skype Channel
- Slack Slack Channel
- Telegram Telegram Channel

<https://portal.azure.com/#@deakin365.onmicrosoft.com/resource/subscriptions/351fd358-bda7-4287-bda4-efb3375f0ab6/resourceGroups/LanguageQnA/providers/Microsoft.BotService/botServices/LanModu-bot/channelsReact>

Select Facebook on Azure portal in bot resource under channels.

Microsoft Azure

LanModu-bot | Facebook

Search

Overview

Activity log

Access control (IAM)

Tags

Settings

- Bot profile
- Configuration
- Channels**
- Pricing
- Test in Web Chat
- Encryption
- Networking
- Properties
- Locks
- Monitoring
- Conversational analytics
- Alerts
- Metrics
- Diagnostic settings
- Logs
- Automation
- Tasks (preview)
- Export template

Facebook credentials

Facebook Workplace

Pages credentials

Configure the page

Page ID: https://www.facebook.com/profile.php?id=100092393197334

Access Token: [REDACTED]

Apply Cancel

Fill App ID and secret previously copied and in page credentials fill the page id and access token and apply.

Meta for Developers

Organic Evolution App ID: 725185286025966 App Mode: Development Live App type: Business

This app is now using Facebook Login for Business. X

Dashboard

Settings

App Roles

Alerts

App Review

Products Add Product

Messenger

**Settings**

Instagram settings

Activity log

Callback URL: https://facebook.botframework.com/api/v1/bots/LanModu-bot

Verify token: [REDACTED]

Pages: Organic evolution 102336336191938

Webhooks

Validation requests and Webhook notifications for this object will be sent to this URL. Token that Meta will echo back to you as part of callback URL verification.

Edit callback URL Show recent errors

Add or remove Pages Add subscriptions

Webhook field version controls

Copy call-back URL and verify token generated in Azure.

This screenshot shows the Facebook Developer Portal's Messenger settings page. The app is named "Organic Evolution" (App ID: 725185286025966). The "Add subscriptions" button is highlighted with a blue box.

We can now add subscriptions.

This screenshot shows the "Edit page subscriptions" dialog for the "Organic evolution" app. Several subscription fields are checked, including "messages", "messaging\_postbacks", "messaging\_optins", "message\_reads", "message\_referrals", "message\_echoes", "messaging\_handovers", "inbox\_labels", and "messaging\_customer\_information". The "Save" button is highlighted with a blue box.

Tick on subscription fields as shown in the pic.

## Step 11: Testing app created

The screenshot shows the Facebook Developers portal interface. The left sidebar is titled 'Meta for Developers' and includes sections for Dashboard, Settings, App Roles, Alerts, App Review, Products, Add Product, Messenger, and Settings. Under Messenger, 'Settings' is selected. The main content area shows an app configuration for 'Organic evolution' (App ID: 725185286025966). The 'App Mode' is set to 'Development' and 'Live'. The 'App type' is 'Business'. A modal window titled 'Edit page subscriptions' is open, listing various subscription fields for the app. Fields like 'messages', 'messaging\_postbacks', and 'messaging\_optins' are checked. Other fields like 'messaging\_optouts', 'message\_deliveries', and 'message\_reads' are unchecked. At the bottom of the modal are 'Learn more', 'Cancel', and 'Save' buttons. Below the modal, there is a note about European privacy updates and a text input field for entering page IDs.

Now that Azure bot and Facebook messenger app are attached, we can see that it is working.

### Conclusion:

Making an Qna app on Facebook Messenger was successful and it is giving a satisfactory result as we chat. This bot can further be analysed and improved. Also, we can add more data sources to this machine son it is updated with knowledge source.