**Lab 16 - Building a Q&A app**

**Introduction**

In the ever-evolving world of technology, the fusion of artificial intelligence and cloud computing has opened new horizons for developers. This lab is designed to provide you with hands-on experience in creating an application using Azure OpenAI services. The lab will guide you through the process of setting up your development environment, configuring necessary parameters, and creating a functional app.

This notebook helps you to build a simple Q&A demo application by doing the following steps:

* **Data preparation** - you will need to adapt this code to have it work with your data
* **Embedding creation** - this will mostly work out of the box
* **Prompt creation** - this will mostly work out of the box, but you could adapt this a little bit
* **App creation** - this will mostly work out of the box, but you can make changes if needed

**Objectives**

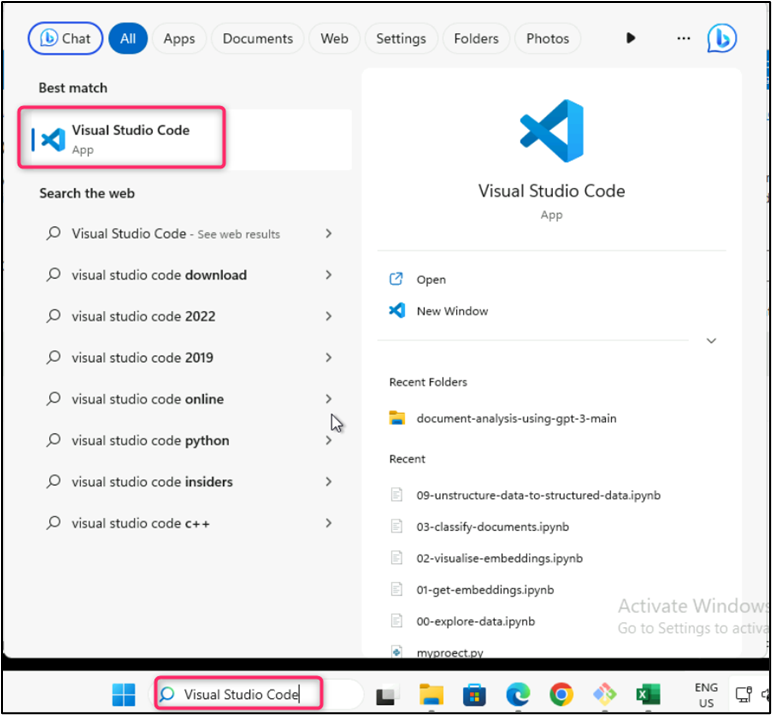
The primary objective of this lab is to familiarize you with the Azure OpenAI service and its capabilities by guiding you through the process of creating an application. Specifically, you will learn how to:

* Install required dependencies and libraries.
* Prepare data for your application.
* Create functions to embed documents.
* Develop a functional app template.

By the end of this lab, you will have a practical understanding of how to leverage Azure OpenAI to develop intelligent applications and gain insights into the potential of AI-powered question-and-answer systems.

**Task 1: App creation using Azure OpenAI service**

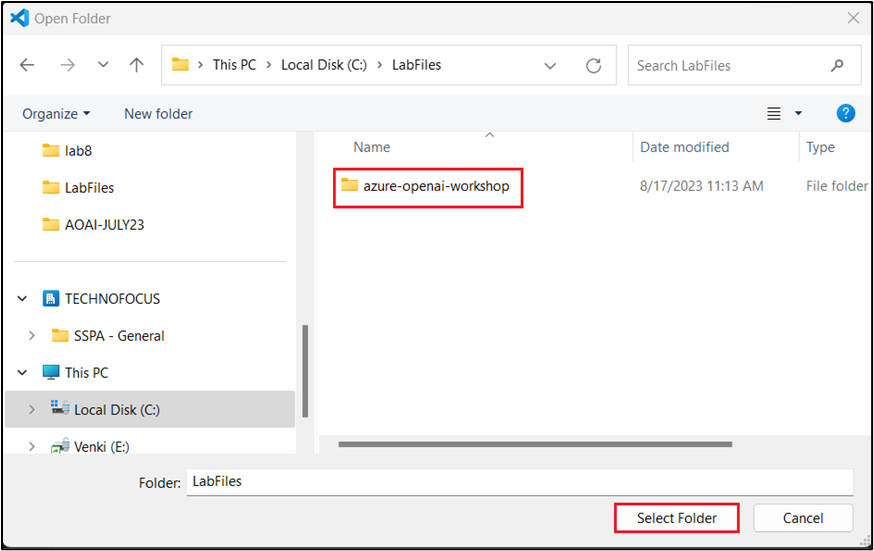
1. In your Windows search box, type Visual Studio Code, then click on **Visual Studio Code**.



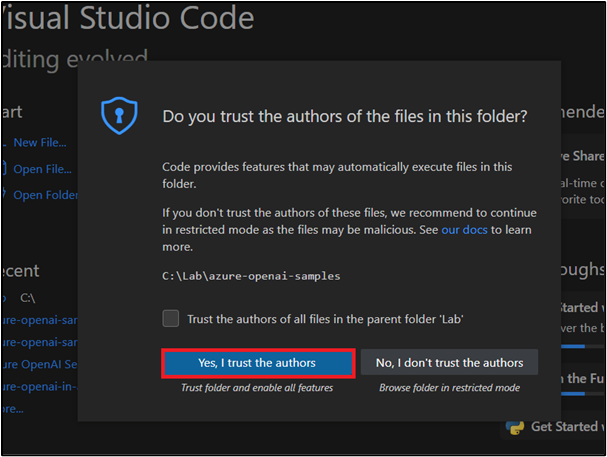
1. In the **Visual Studio Code** editor, click on **File**, then navigate and click on **Open Folder**.



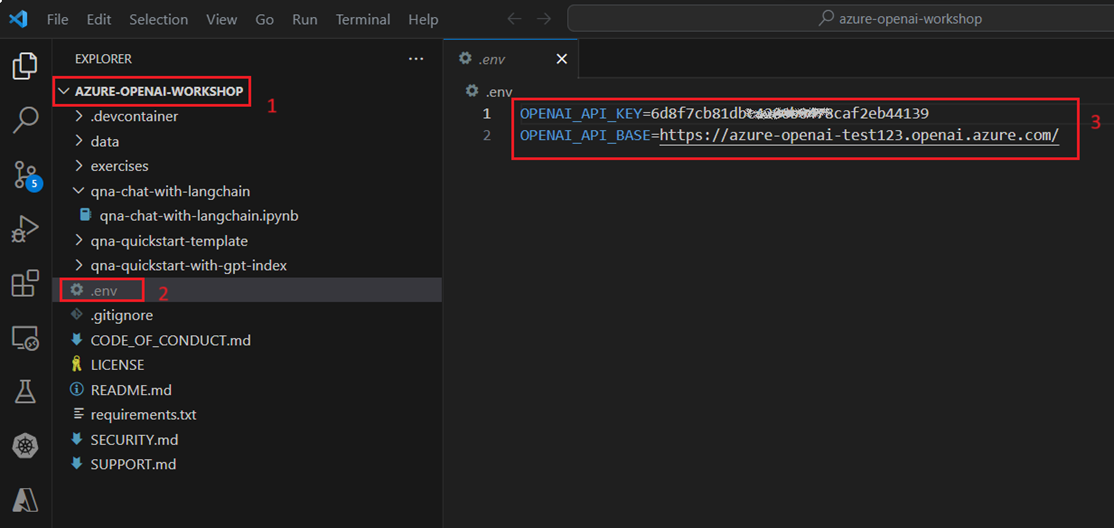
1. Navigate to **C:\LabFiles**, select **azure-openai-workshop** folder and click on the **Select Folder** button.



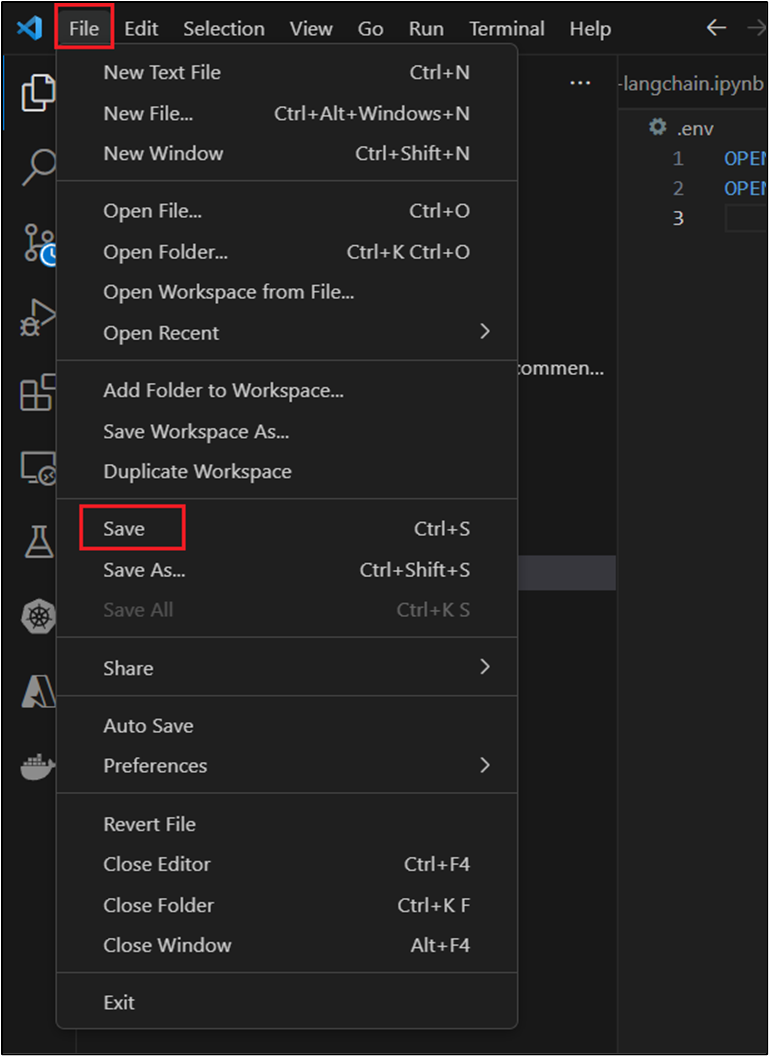
1. If you see a dialog box - **Do you trust the authors of the files in this folder?** then click on **Yes, I trust the author**.



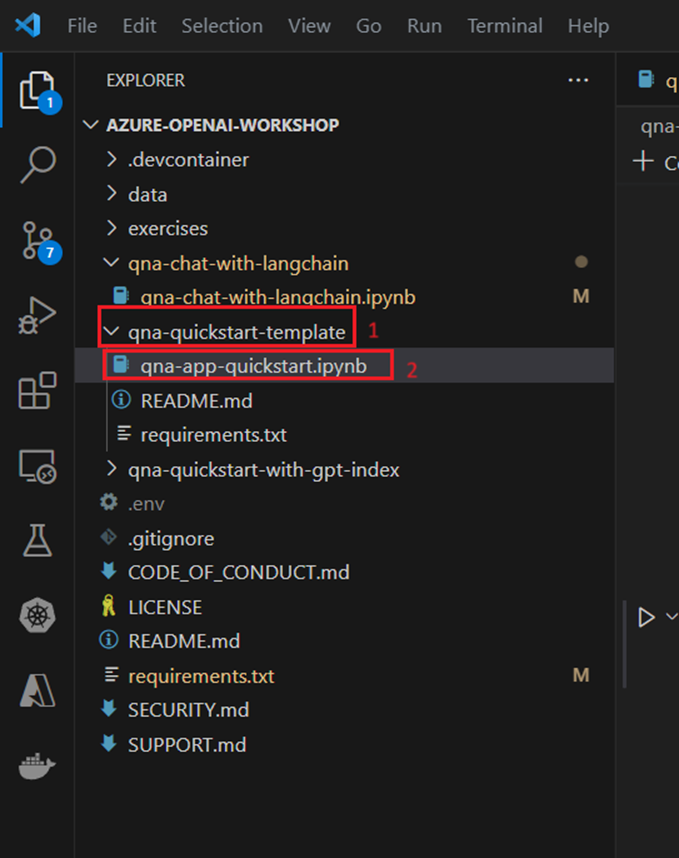
1. In Visual Studio Code, navigate and click on **.env** file.
2. In the **.env** file, replace **Azure OpenAI Endpoint, Azure OpenAI Key,** with the values that you have saved in your notepad in the **Lab 1> Task 2**.



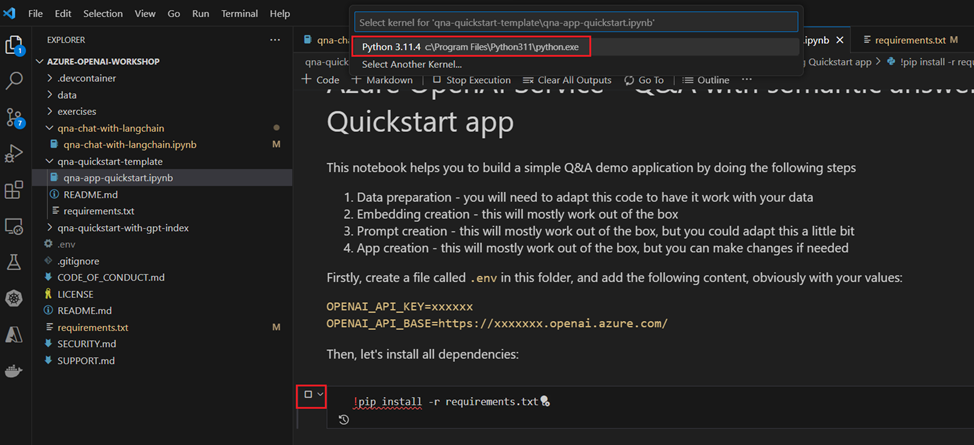
1. Click on **File** and the click on **Save**.



1. In Visual Studio Code, under **AZURE-OPENAI-WORKSHOP**, navigate and click on **qna-quickstart-template** and click on **qna-app-quickstart.ipynb** notebook.



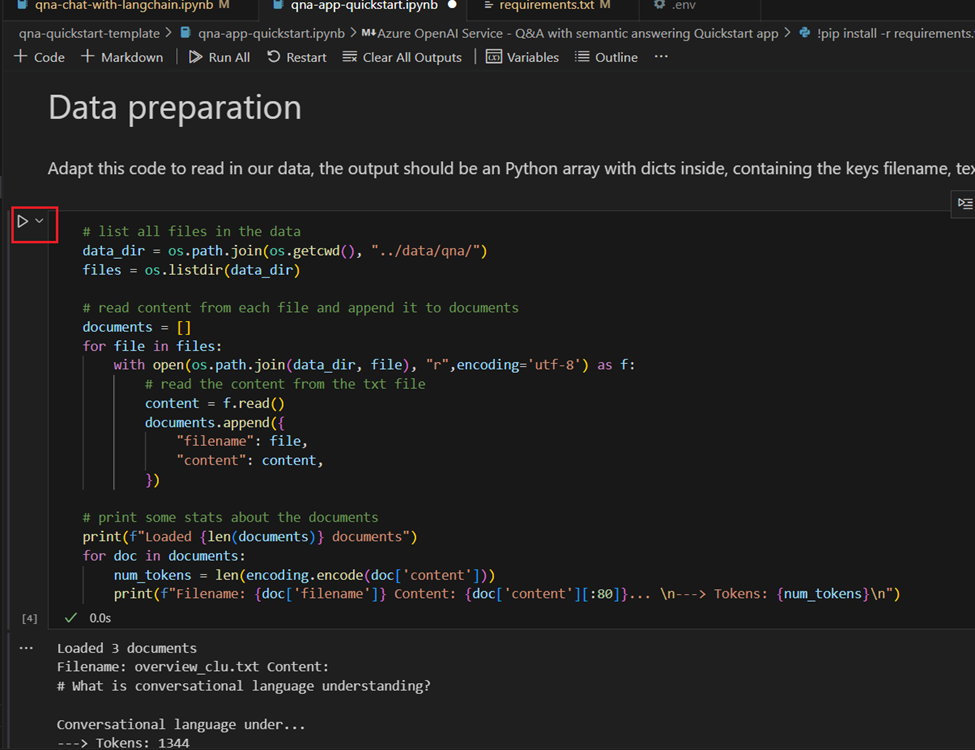
1. In the main page of Visual Studio Code editor, click on the **Play** button of the 1st and 2nd cells to install all dependencies. If prompted to select the environment, then select **Python Environments** as shown in the below image.



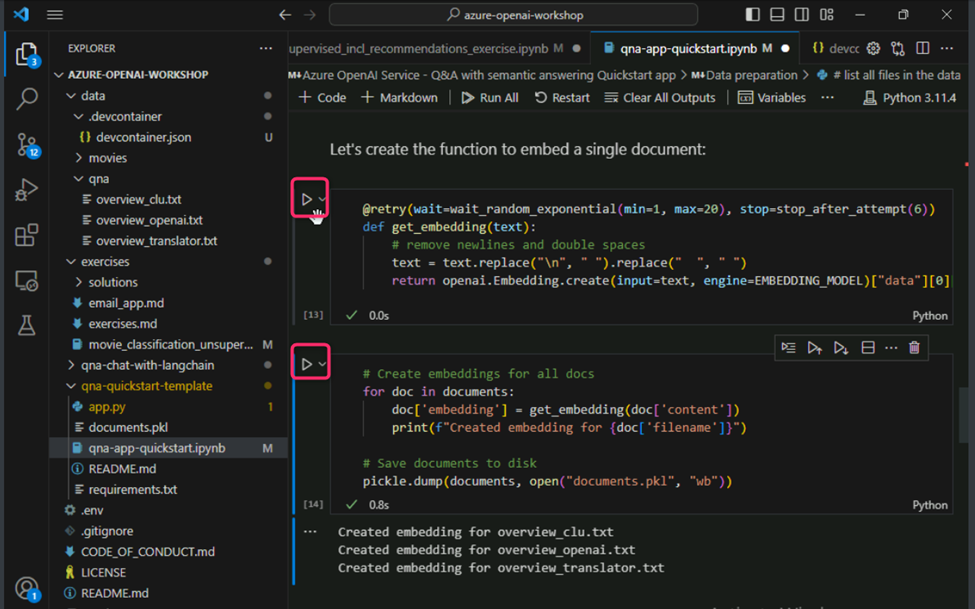




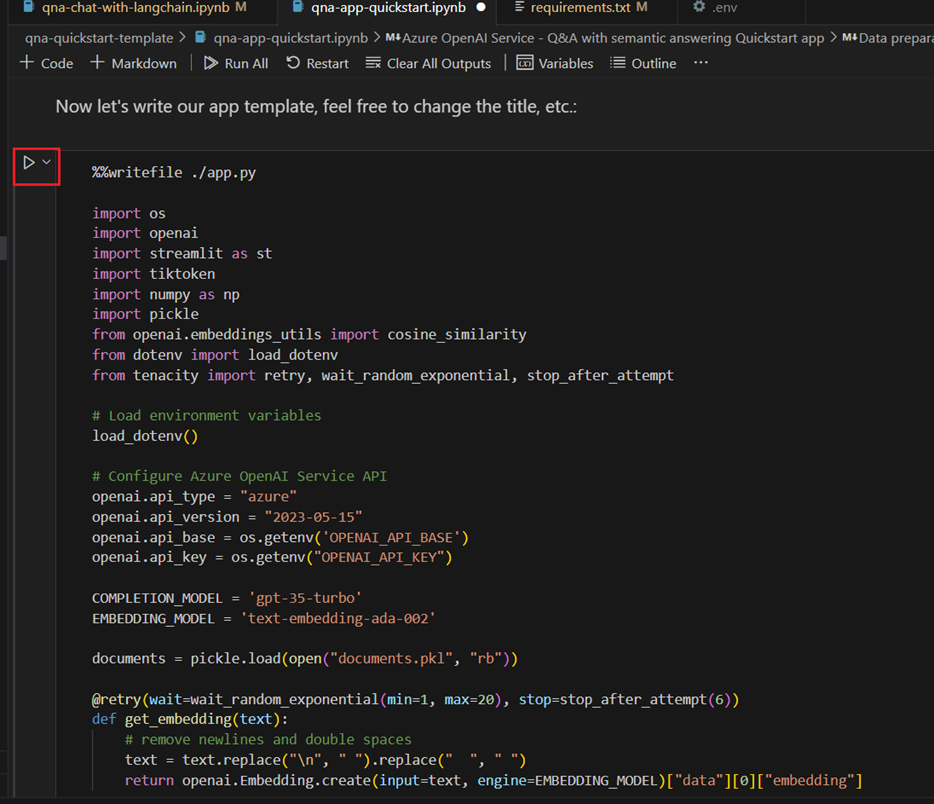
1. For data preparation, select and execute the 3^rd^ cell by clicking on the **Play** button as shown in the below image.

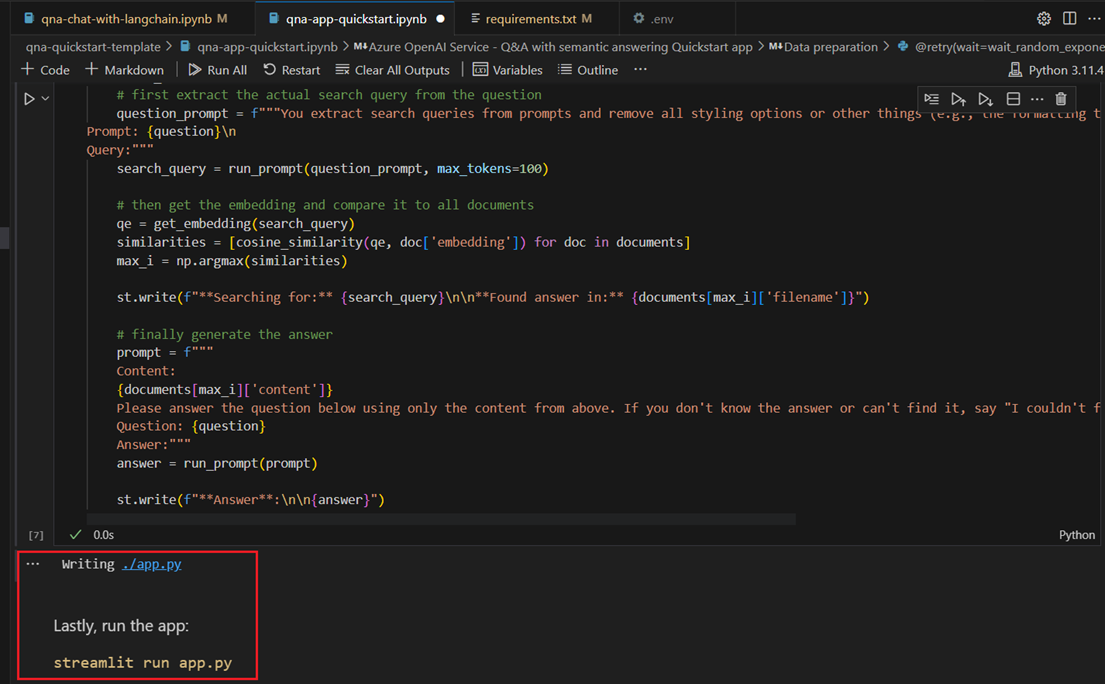


1. Create the function to embed a single document by selecting and executing the 4th and 5th cells.

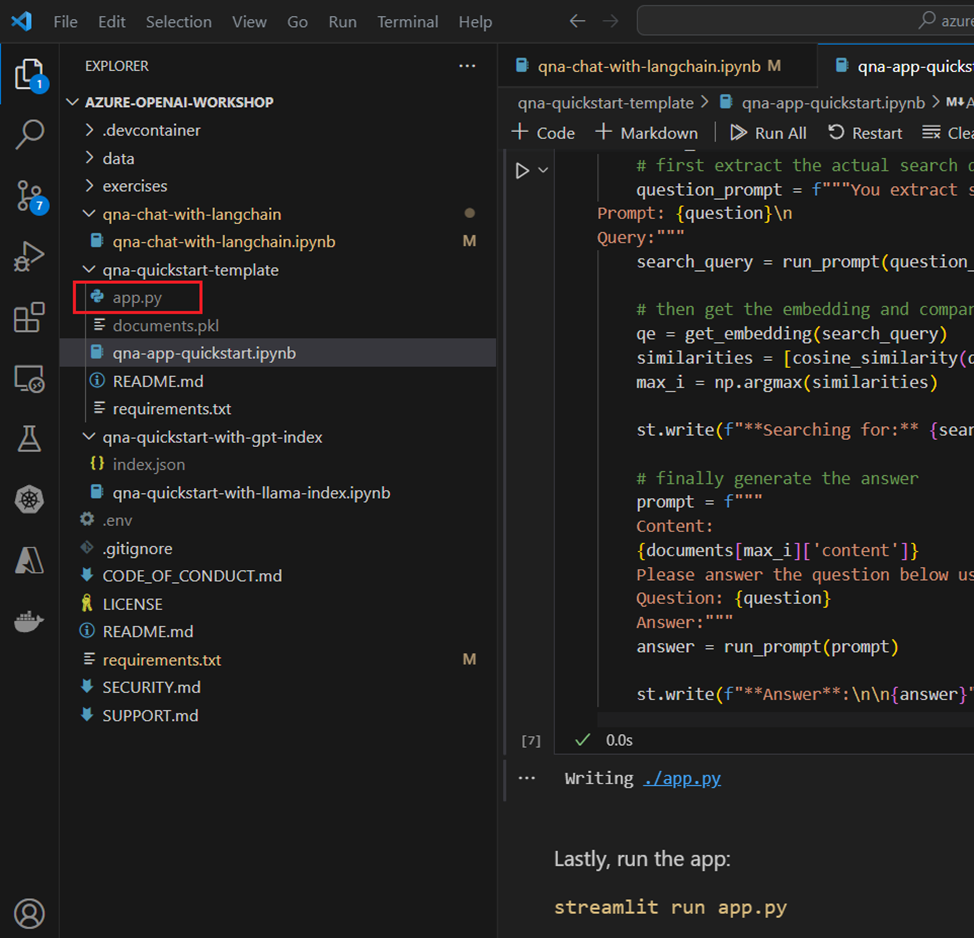


1. To write app template, select and execute the 6th cell by clicking on the **Play** button.

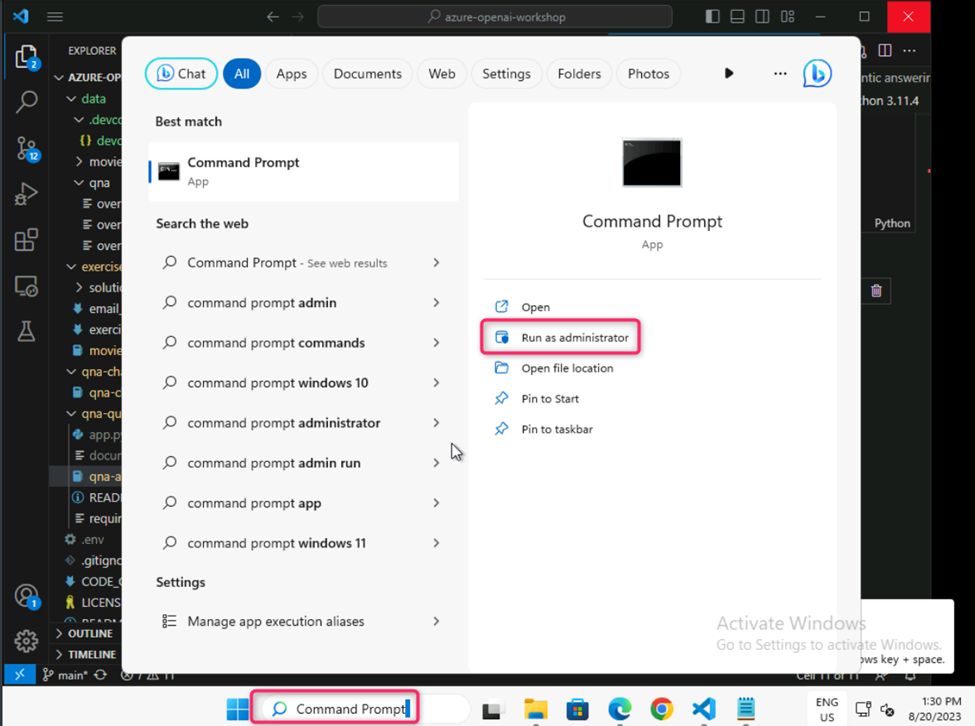




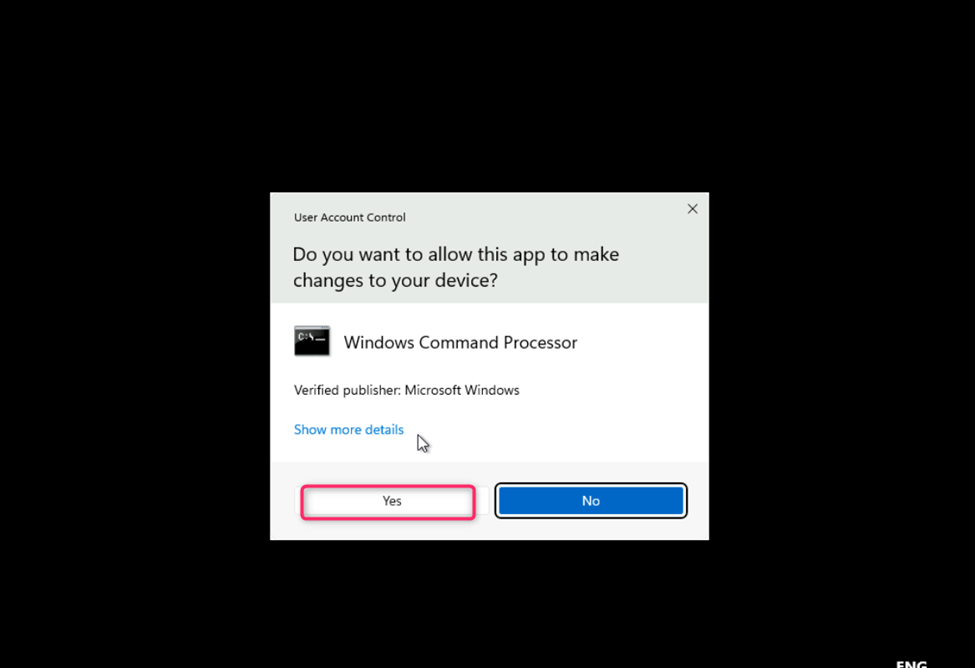
1. Navigate and click on **app.py** under **qna-quickstart-template** to view the application code.



1. Type Command Prompt in your windows VM search box and then open the Command Prompt by clicking on **Run as administrator**.



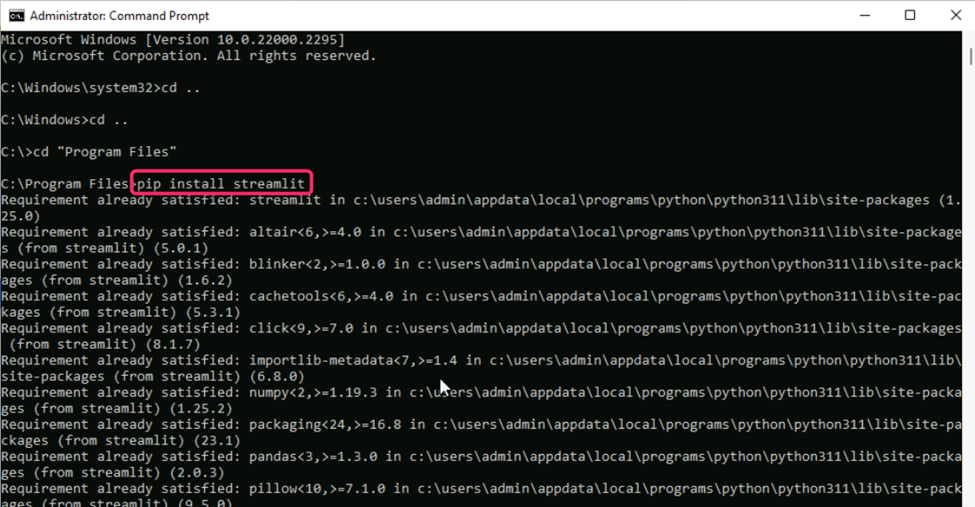
1. On **Do you want to allow this app to make changes to your device?** dialog box, click on **Yes** button.



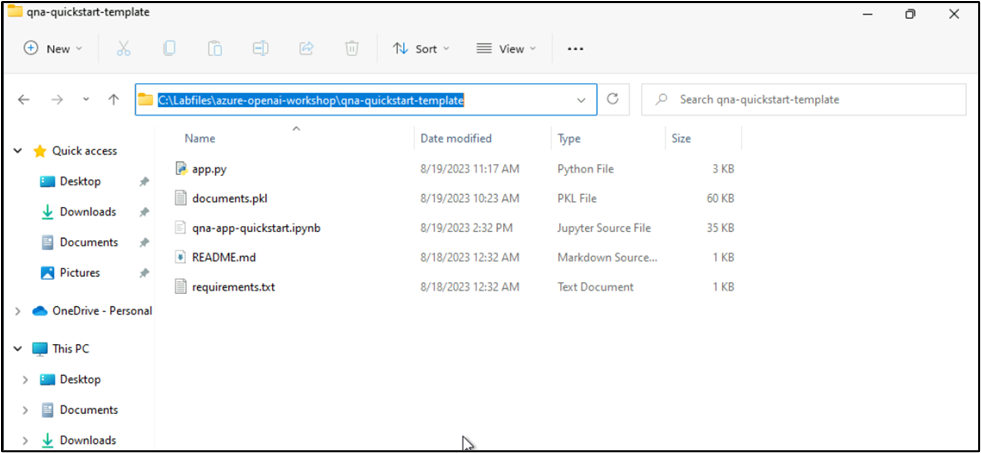
**Important Note**: You need to change the current directory to the **Program Files** directory (The command used to move back to the previous directory is **cd .. [space after cd then two dots],** the command used to move to the next directory is **cd \)**

1. Use the following command to install Streamlit in command prompt.

pip install streamlit



1. Navigate to the location of your application (here, in this lab, the application is located at **C:\Labfiles\azure-openai-workshop\qna-quicksart-template** folder).



1. Change your directory to application directory using the following command as shown in the below image:

cli

cd ..

(go back to C:\ drive)

cli

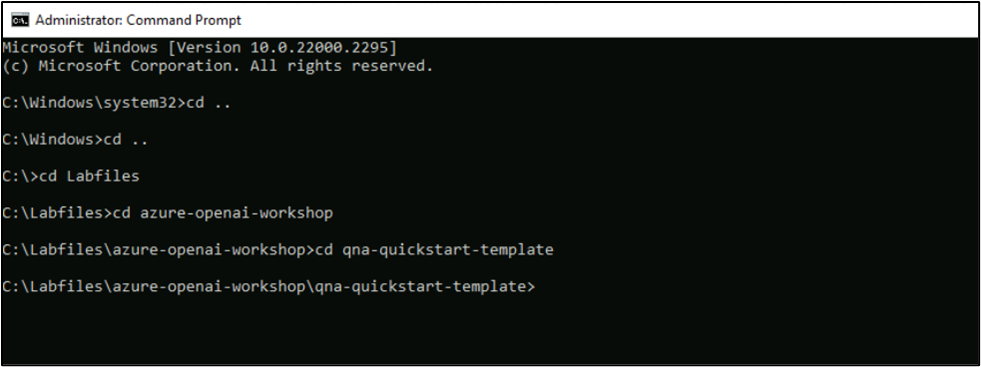
cd Labfiles

powershell

cd azure-openai-workshop

powershell

cd qna-quicksart-template

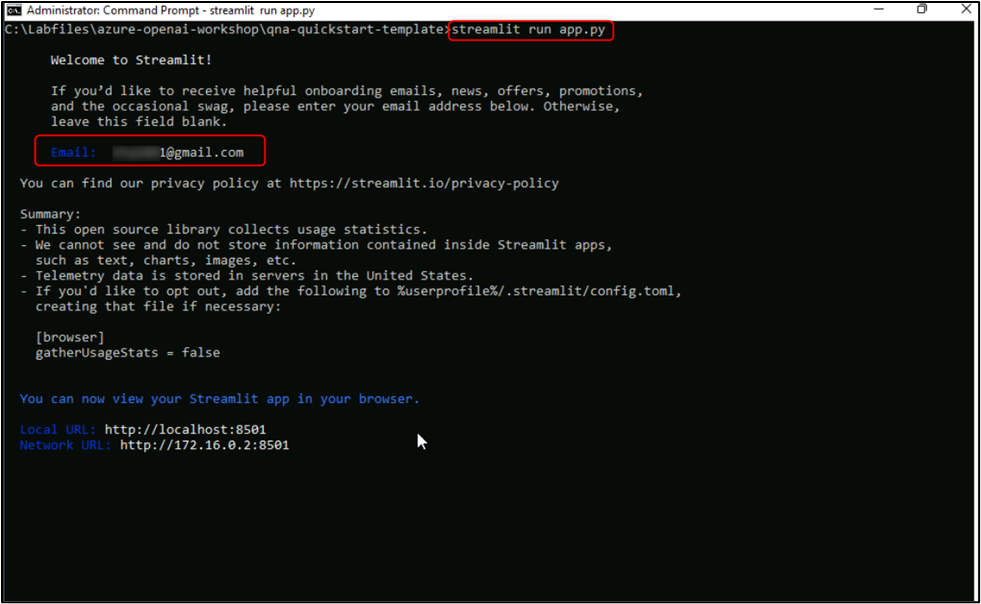


1. Now, run the application using the following command.

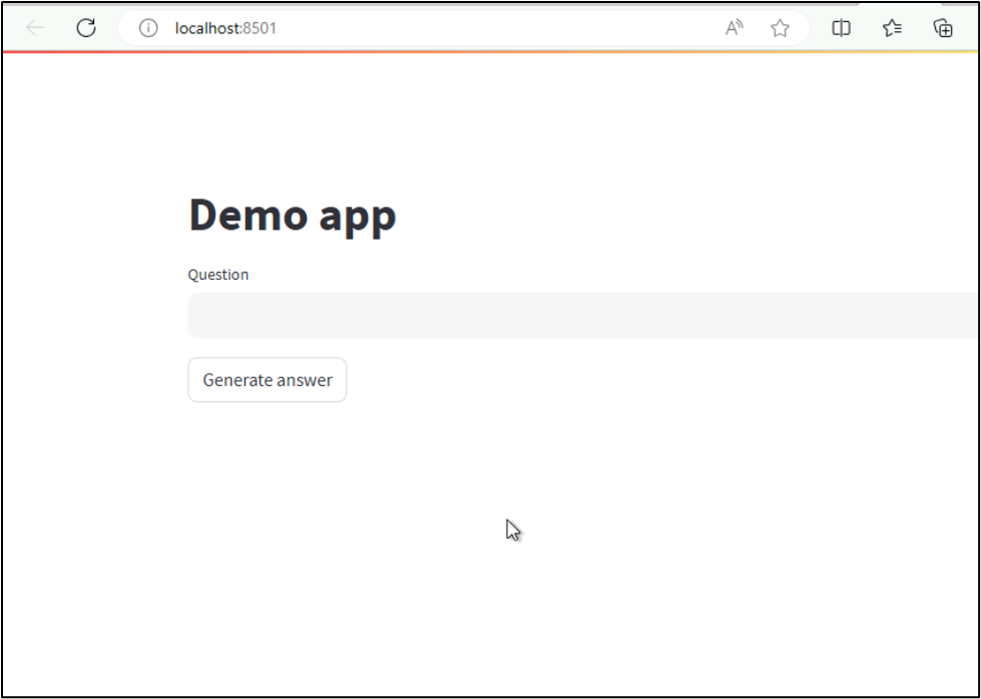
cli

streamlit run app.py

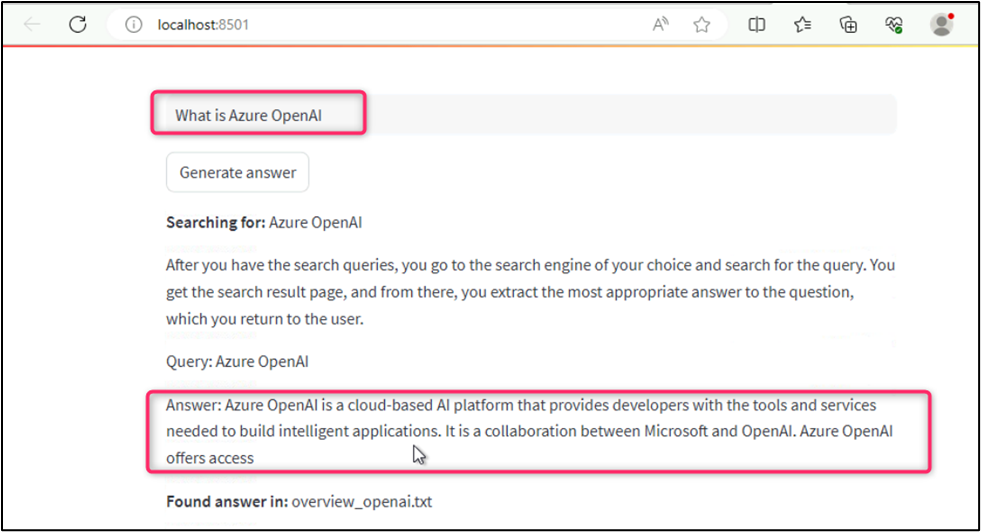
In case, you would like to receive helpful onboarding emails, news, offers, promotions from Streamlit, then enter your email address.



1. Your application will be opened in localhost.



1. In the Question field, enter **What is Azure OpenAI**, then press the **Generate answer** button. You'll receive the answer for your question.



**Summary**

In this lab, you've explored the capabilities of Azure OpenAI services through the creation of an intelligent application. You've installed the essential dependencies and libraries, prepared the data, created functions to embed documents, and then you've executed the Q&A application using Streamlit and obtained a response for your question. After successfully executing this lab, you will be able to harness the capabilities of AI-powered question-and-answer systems for various practical applications.

**Important Note: Please do not delete the Resource group and Azure OpenAI Service (Azure-openai-testXX). The same Resource group and AOAI service will be used throughout all the labs.**