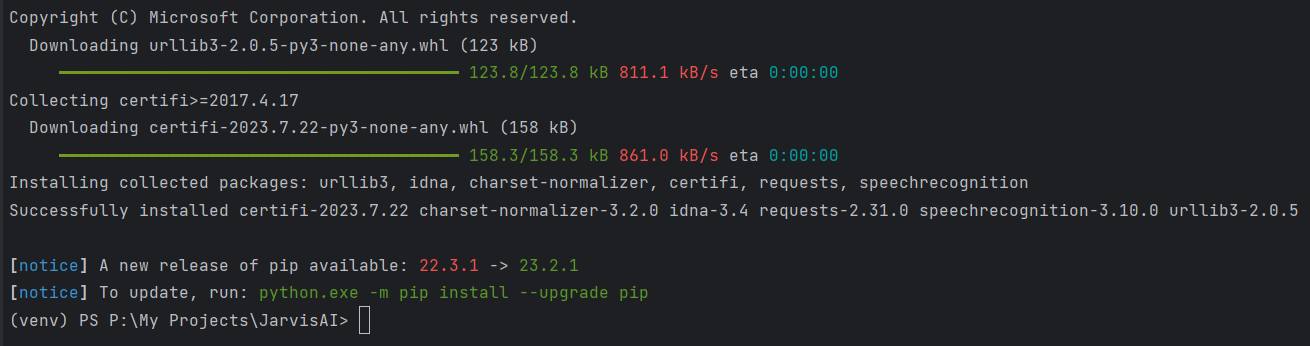
**My OpenAI API powered desktop assistant (Application AI) –**

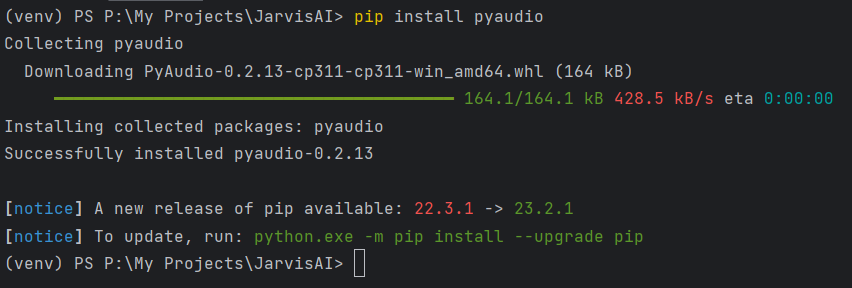
Open project in PyCharm – creating virtual env (as all the packages will be installed)

1.)Will install the ‘speech recognition’ package in terminal



Similarly we will install ‘wikipedia’ and ‘openai’ packages.

2.) See net up will install ‘pip install pyaudio’ in Pycharm



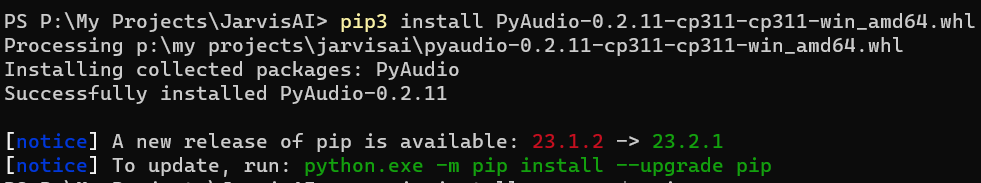
3.) Now we need to install the Pyaudio in windows.

For this the steps are as follows –

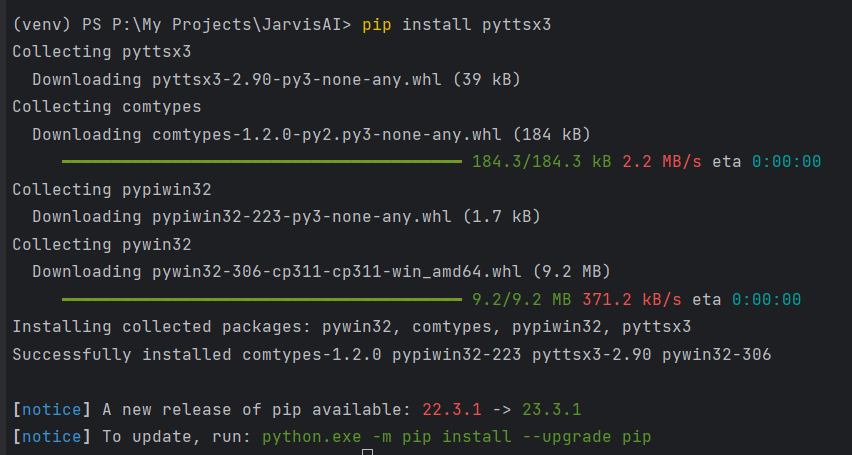
1. Search the ***unofficial python binaries.***
2. Install the ***64 bit*** version pack ***whl*** for windows from the site –

<https://www.lfd.uci.edu/~gohlke/pythonlibs/>

1. Remember to install whl of the same version as of your systems python version.
2. Re-locate the installed file to the same location as of Jarvis AI folder has.
3. In that right click and open it as terminal.
4. After that type the command for installing, ‘pip3 install <pyaudio.whl> ’, at the pyaudio.whl is the installed file name here.



1. Now will import the ‘os’and run pyaudio. To allow text-to-speech functionality we will install ‘pyttsx3’ package.



1. Since the function in the code ‘takeCommand’ is returning the same text we said, we need to handle the infinite times speaking and listening query and answering query using the “try catch” method. This is the exceptional handling.
2. Now will ask Jarvis to do certain task for us, like will import webbrowser and allow it to open sites in browser.
3. Moving forward we will now include the OpenAI stuff here, which allow user to command like GPT and get results stored in folder we create “OpenAIoutput”, where results will be stored in .txt files.
4. For this to accomplish we will do the following –
5. Create an OpenAI account
6. Create our own API key, which is very secret to us and will be used to directly call GPT versions to generate results for us.
7. Then paste this secret key in config.py file in project, which later will be accessed for use to generate answers.

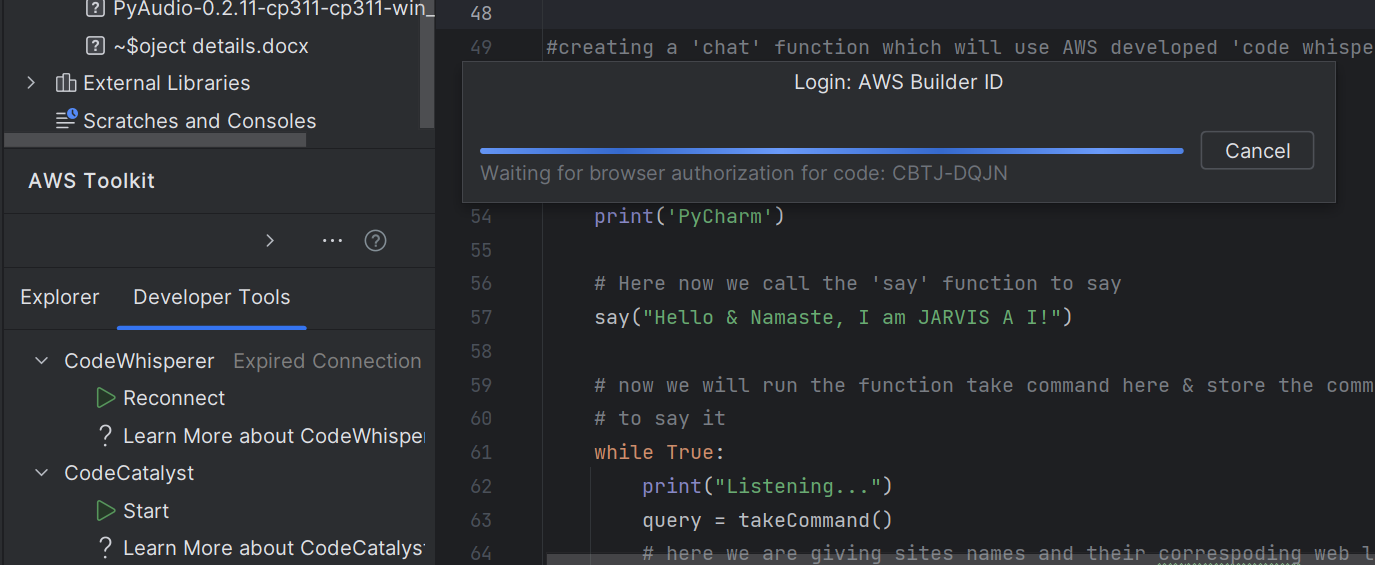
j) ***“Believer$$02GPT#AI”*** This is my secret API key name.

This is secret key generated 🡪

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

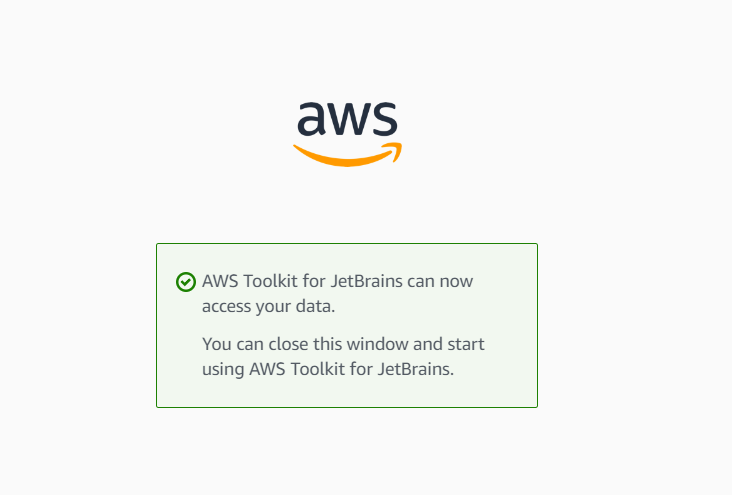
Remember this is very very important.

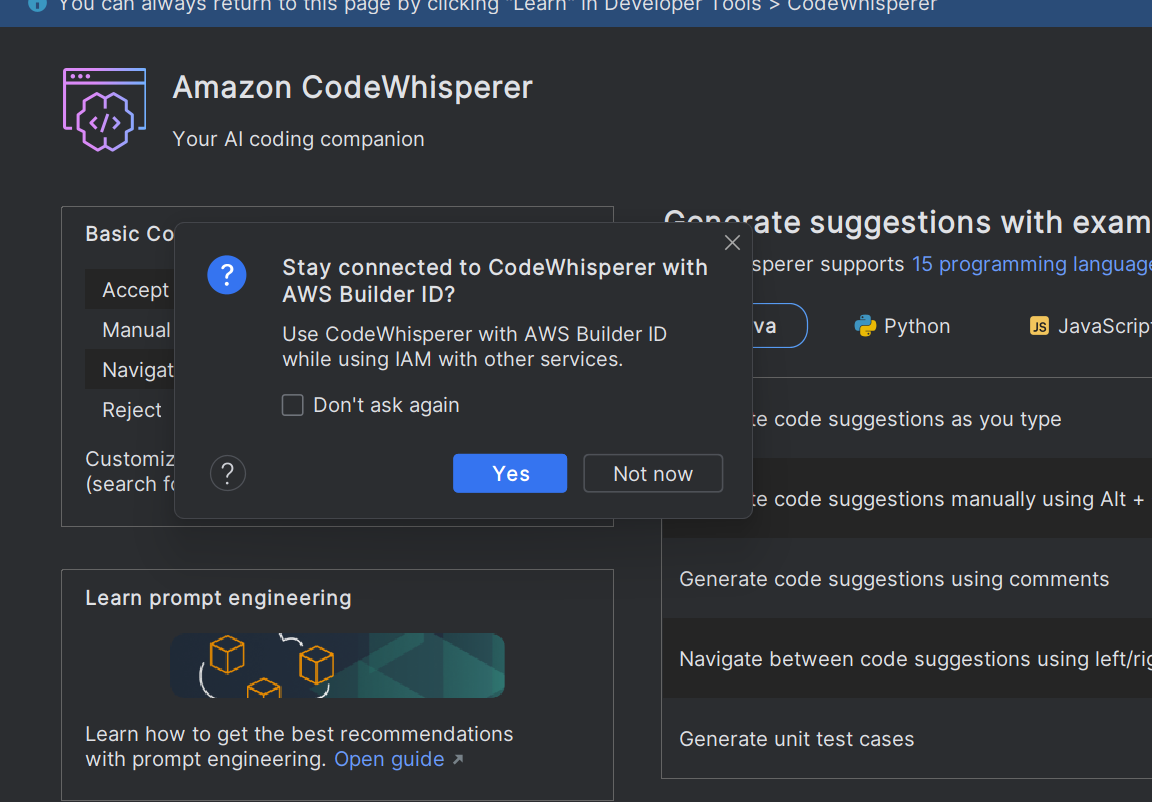
* Next will install ‘AWS toolkit’ plugin in PyCharm for using amazons free ‘code whisperer’. We will double click on start and create an account for AWS builder.
* This will allow our code to auto suggest code and complete codes for us like Blackbox AI does.



Like this it looks while setup.

Now the profile is created successfully as –





Overall the steps are summarized here –

✦How to make an AI desktop assistant using Python and OpenAI

* Download and install PyCharm, an advanced IDE for Python development
* Install required packages like speech recognition, Wikipedia, and OpenAI for voice recognition, accessing data from Wikipedia, and utilizing OpenAI's API respectively

✦Download and install PyCharm on Windows

* Download PyCharm from the provided links in the description
* Launch the downloaded setup and follow the installation process

✦Jarvis AI can recognize voice commands and perform actions like opening websites.

* Jarvis AI can handle errors while recognizing voice commands.
* Jarvis AI can open websites like YouTube and Wikipedia.

✦Jarvis AI can open and play music files and provide the current time

* Jarvis AI can open and play music files by using the 'os' module and the 'subprocess' package
* Jarvis AI can provide the current time by importing the 'datetime' package

✦Jarvis AI successfully opens FaceTime and other applications

* Jarvis AI successfully opens FaceTime and follows commands to open other applications like Paskey
* Jarvis AI is integrated with OpenAI to generate responses and automate tasks

✦Use artificial intelligence to generate text prompts and save them in a file.

* Create a function called defai that takes the prompt and returns the response.
* Ensure the directory for storing the prompt files exists and create it if it doesn't.

✦Create a functionality for chat in the JarvisAI using Artificial Intelligence.

* Introduce a plugin called Code Whisperer for code completion in JetBrains.
* Implement the chat function in the JarvisAI to enable conversation with Jarvis.

✦Build a desktop assistant using OpenAI's ChatGPT model

* The assistant can respond to voice commands and perform various tasks
* Additional features like fetching weather and news can be added