Summary of Installation and Setup Steps For Miniconda3 (a python distribution)

Summary of Installation and Setup Steps For Miniconda3 (a python distribution)

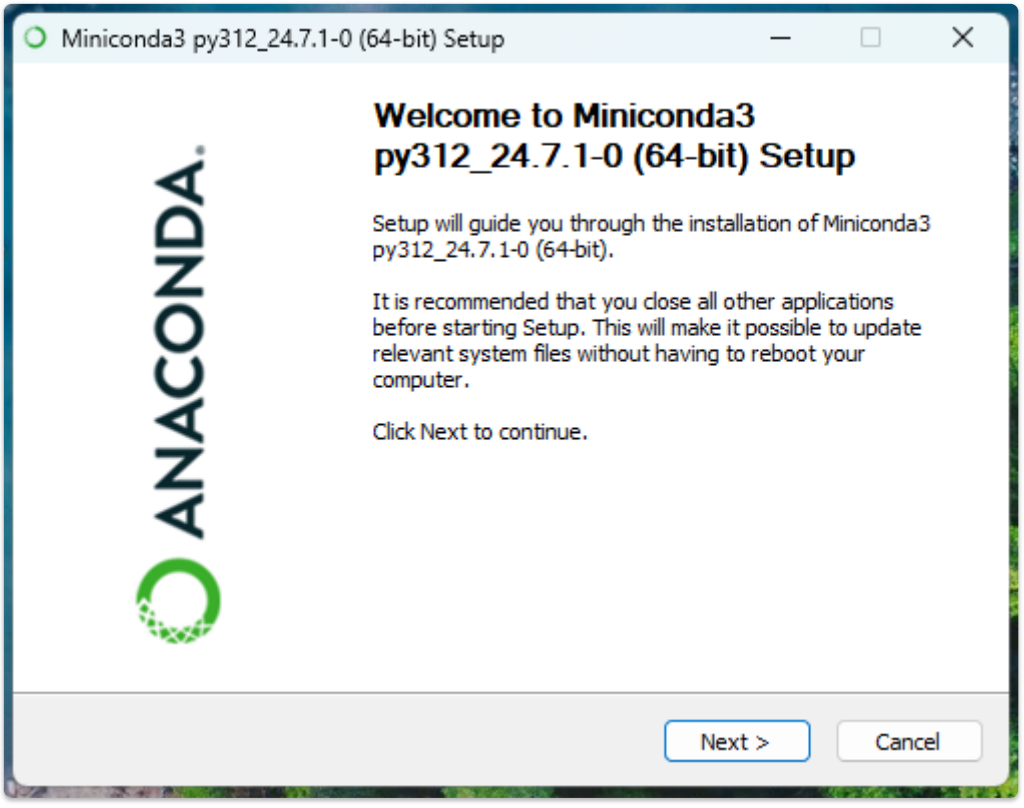
1.) Installing Miniconda3:-

A) Via Graphical User Interface (GUI):-

In this step we download the latest miniconda3 installer from [anaconda website](Miniconda — Anaconda documentation) based on your Operating System whether windows, macOS and Linux and run it to install miniconda3.

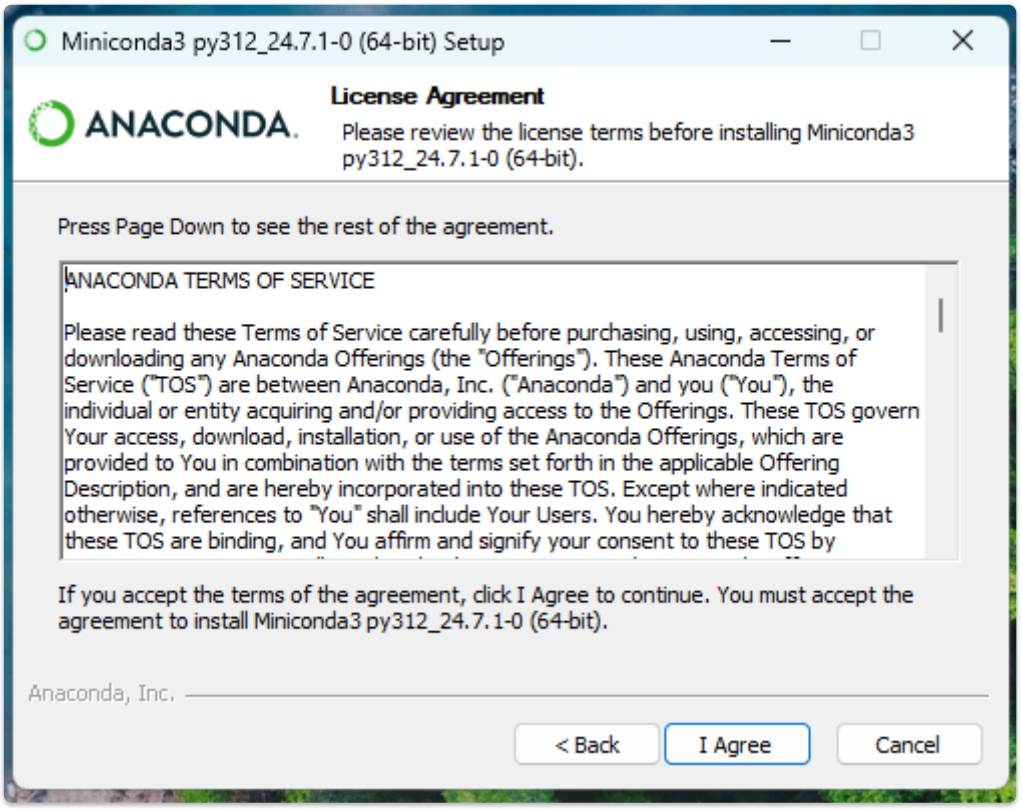
Let's look at GUI installer steps to install Miniconda3 on windows:-
After running the downloaded installer and running it you get:

Step-1:



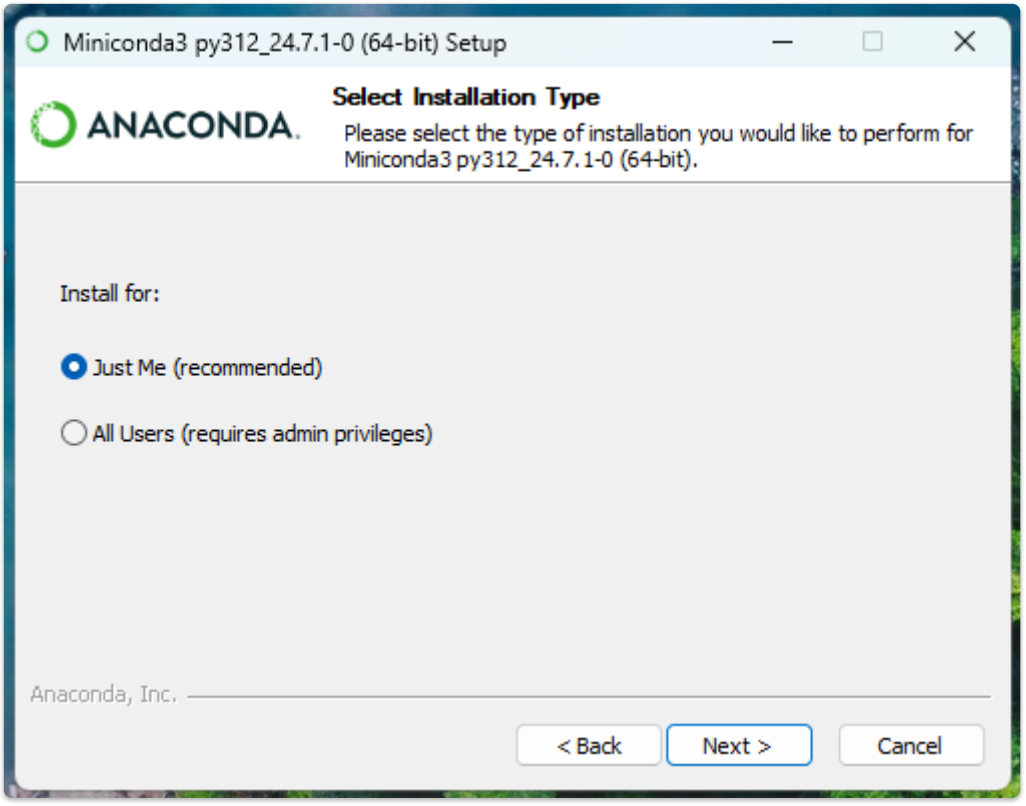
This is the main window that you get after running the installer, now press `Next >` to continue installation.

Step-2:



After pressing `Next >` in the previous step you get this window where `License Agreement` is shown which outlines the *ANACONDA TERMS OF SERVICE* which is required by us to agree to if we want to continue with installation and use the miniconda, so now pause and read the `License Agreement` and then press `I Agree` to continue with installation.

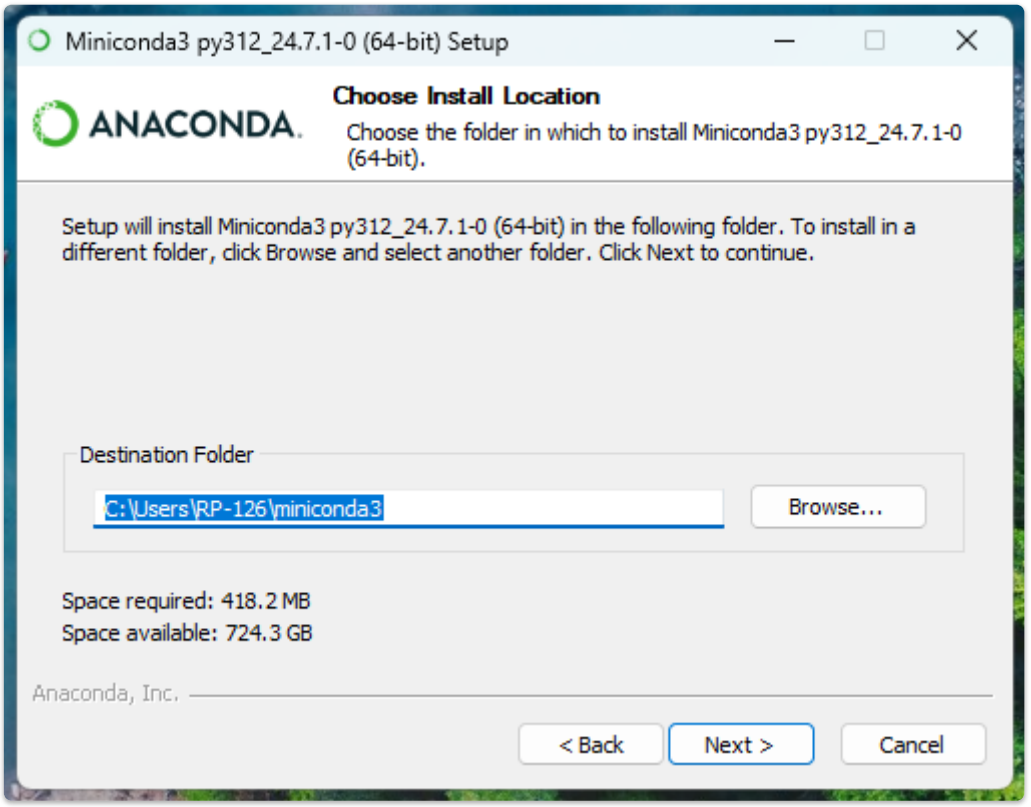
Step-3:



After Pressing `I Agree` in the previous step you get this window. Here by default `Just Me (recomended)` option is selected, now we don't have to

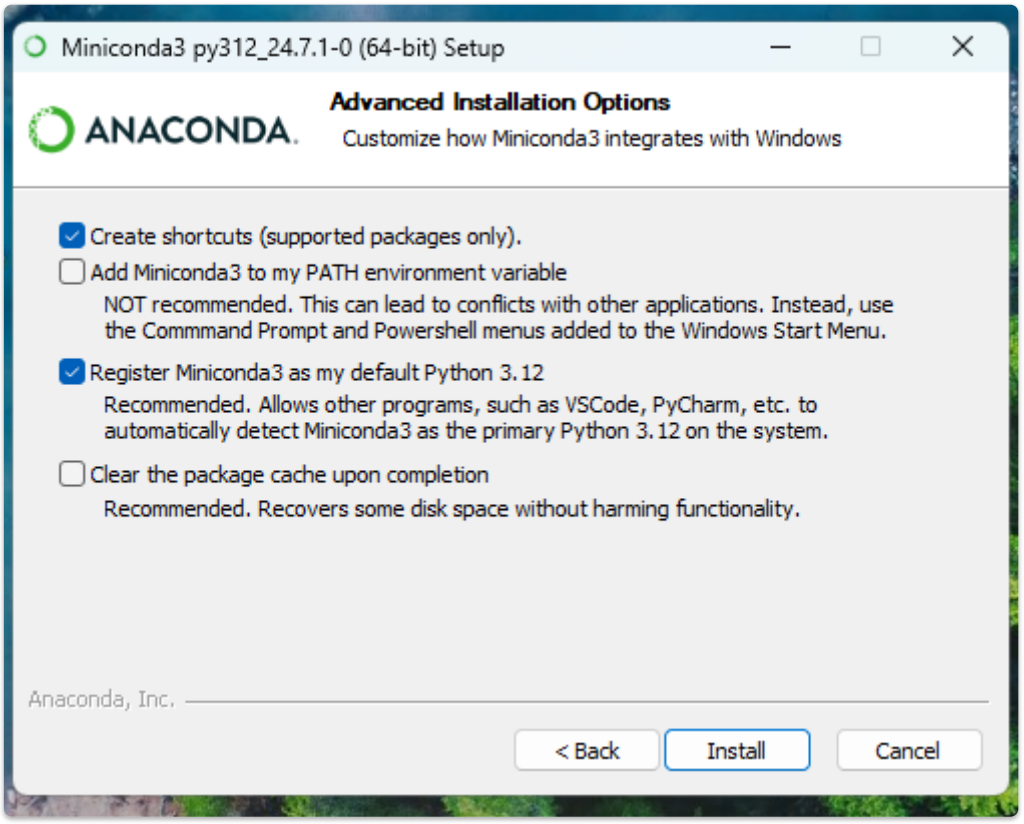
change anything and just click `Next >` to continue with installation.

Step-4:



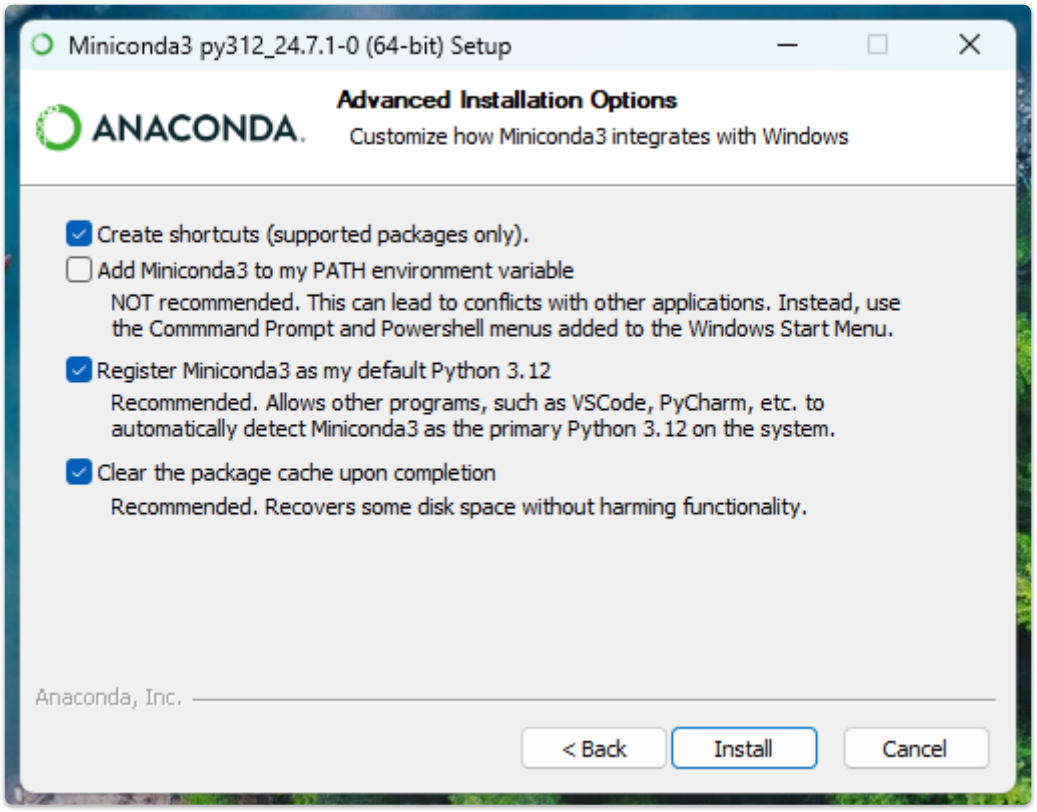
After pressing `Next >` in previous step you get this window. Here we have to `Choose Install Location` where we want miniconda to install. By default Miniconda will install to `C:\User\<Your_UserName>\miniconda3` location. Here we don't need to change anything but if you have space in your username folder for example like `C:\Users\Tavneet singh\miniconda3` or if your username folder doesn't have the space required (which is displayed on the window) to install the miniconda then it is recommended to install to miniconda to a different folder which you can choose by using the `Browse ...` option, this folder can be on some different drive on your system or on the same `C:\` drive for example like `C:\miniconda3` or `D:\programs\miniconda3`. In most cases you don't need to change `Destination Folder` and if you have to do then after selecting the new `Destination Folder` press `Next >` to continue installation.

Step-5(a):



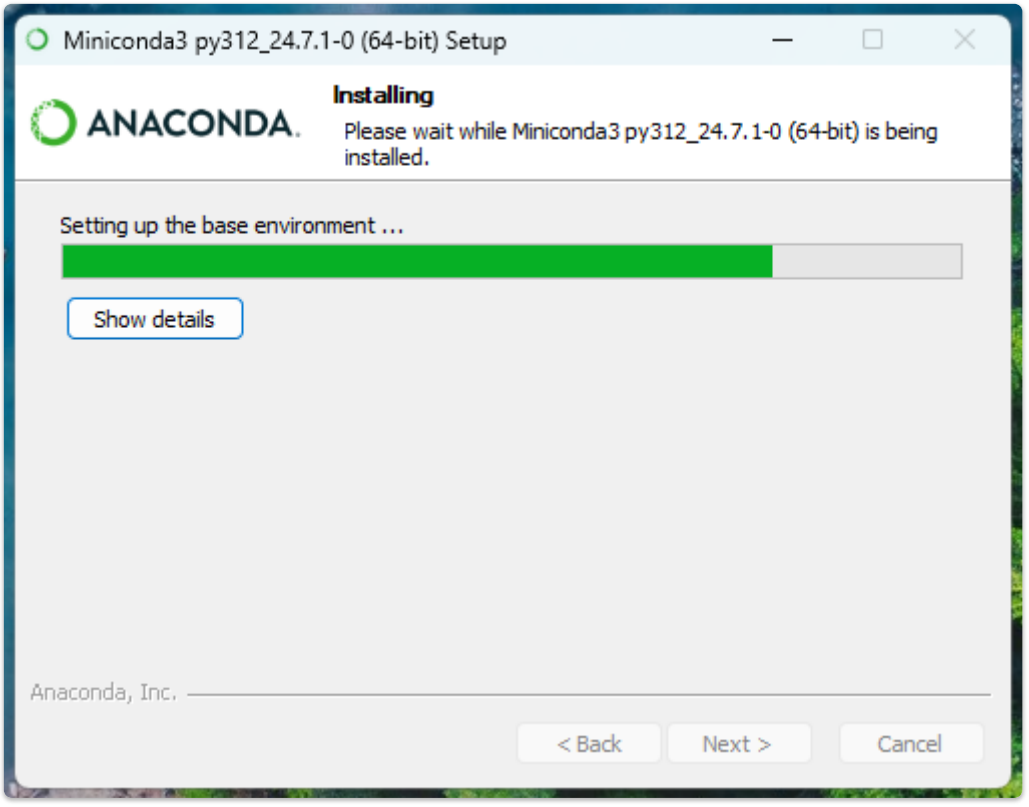
After pressing `Next >` in previous step you get this window, Here you get `Advance Istallation Options` which controls how Miniconda Integrates with Windows. Now, here by default the 2 options i.e; `Create Shortcuts` and `Register Minicoda3 as my default Python 3.12` that are shown in the screenshot of the installation wizard are selected, here we just keep them selected as it is and just tick one another option `Clear the package cache upon completion` which just clean the cache of packages like `.tar` files that gets downloaded and now not required anymore when we install any new package. This helps us to manage space on our storage device. After selecting this option the above window will look like below:

Step-5(b):



Now press `Install` to continue with installation.

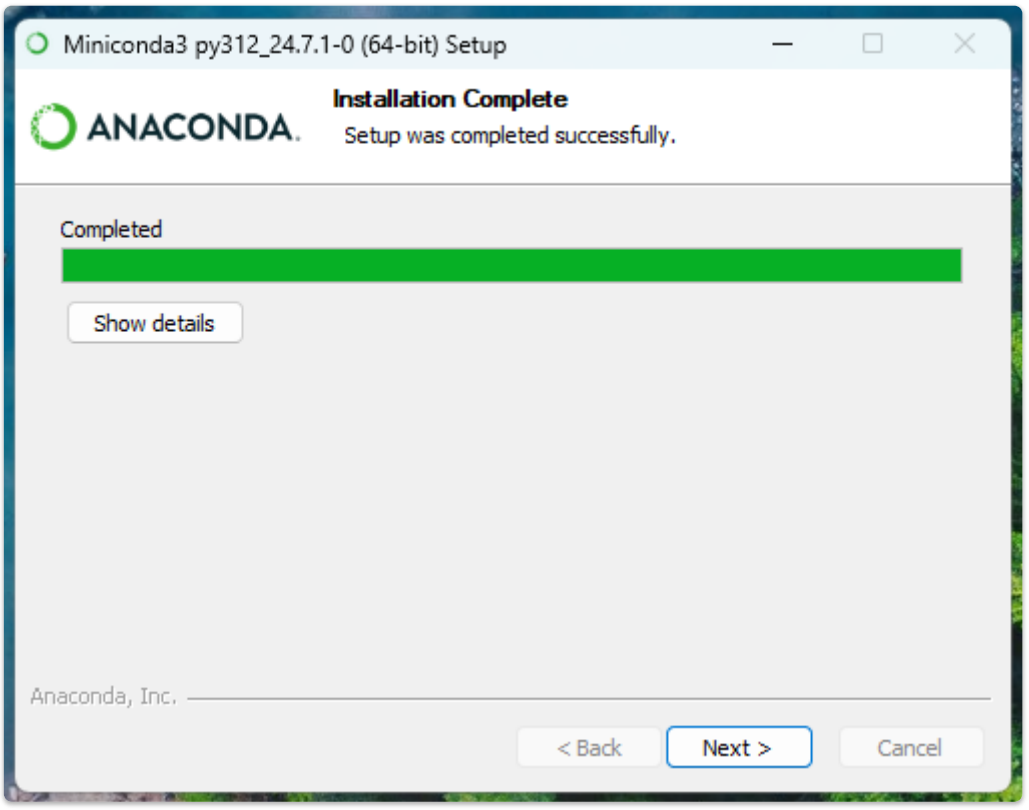
Step-6(a):



After pressing `Install` in previous window the installer continues with the installation process and installs the miniconda on your system and the progress bar for that is also shown on the installation window.

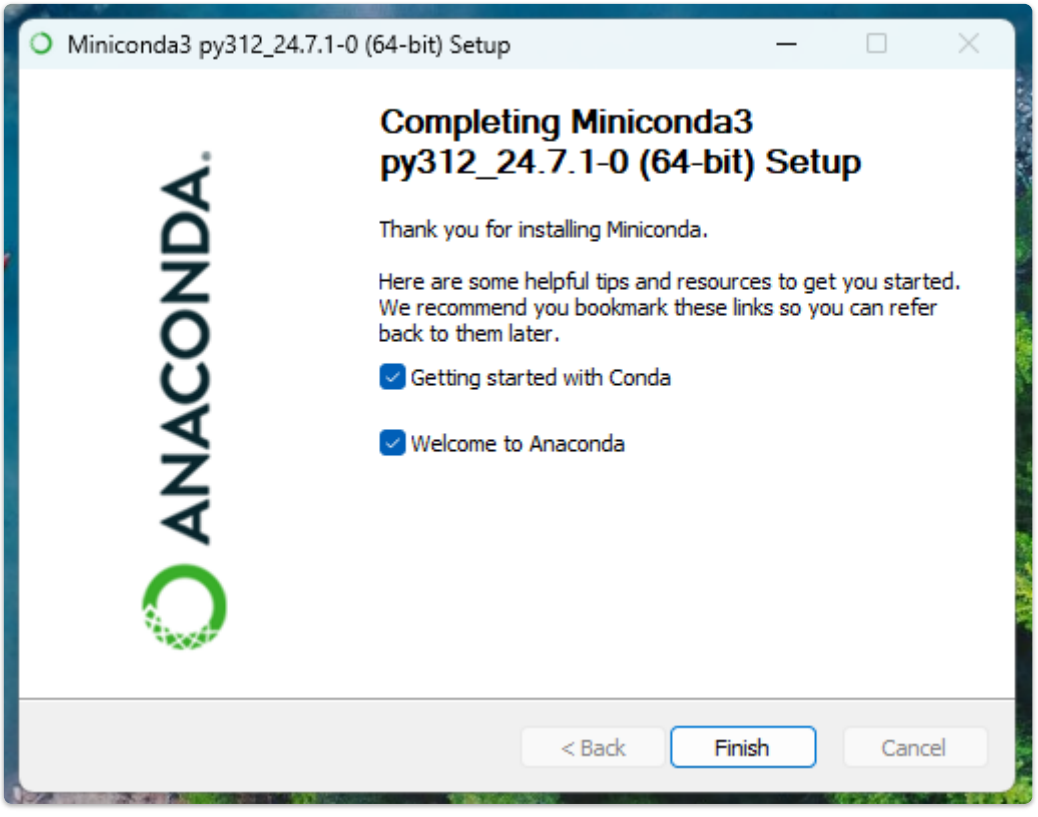
Also, if you want to know what is currently happening in the installation process then you can look at that by clicking `Show details` button.

Step-6(b):



When the installation process is completed successfully then this window gets displayed where `completed` is written on progress bar which indicts that installation has completed without any errors. Now press `Next >` to continue.

Step-7:



After pressing `Next >` in the previous window then this window gets displayed. Here it thanks you for installing Miniconda and also list two options which is selected by default which includes resources that we can refer to get more information on how to use conda. These are just webpages that will open in our default browser after we press `Finish` button which will close the installation wizard and this concludes the installation process.

Now at this stage the miniconda3 has been properly installed on your machine.

B) Via Command Line Instruction

Windows

Alternatively, if on *Windows11* or *Windows10 Build 1809* or higher *(and this can be checked by running* `winver` *after opening start menu)* we can use `winget` which is a new cli tool introduces by Microsoft. For installing via this method open start menu and type PowerShell and open that, a new terminal window will appear and in that window type the following command:-

```
winget install Anaconda.Miniconda3
```

This above command will install Miniconda3 on your system, and after running the above command you might get asked for `yes/no` for accepting agreement so, you should type `yes` and accept the agreement for using the `winget` cli.

MacOS

If on *macOS* open the terminal app and then copy and run the following four commands which download the latest M1 version of the MacOS installer, rename it to a shorter file name, silently install, and then delete the installer:

> **Note:**
>
> For best results, copy all of the commands in each code block and run them all at once.

```
mkdir -p ~/miniconda3
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-arm64.sh -o
~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh
```

Linux

If on *Linux* open the terminal app and then copy and run the following four commands which download the latest 64-bit version of the Linux installer, rename it to a shorter file name, silently install, and then delete the installer:

> **Note:**
>
> For best results, copy all of the commands in each code block and run them all at once.

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O
~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh
```

Shell Initialization

After running these above commands (on *macOS* and *Linux* respectively) miniconda3 will be installed on your system and after installation you need to initialize the installed miniconda for your shell whether it is `bash` or `zsh` which is necessary for running `conda` commands via command-line.

> **Note:**
>
> By default the macOS has `zsh` shell and Linux has `bash`, if you have don't change it manually.

Also, If you are not sure about your what is your current macOS or Linux shell you can use the following command to check what is your current shell:

```
echo $0
```

If your current shell is bash then the above command will return

```
bash
```

And if your current shell is `zsh` then the above command will return

```
zsh
```

or if you have installed any other shell then the `echo $0` command will return the name of that shell.

After figuring out the current shell then you can use the following commands for initialization miniconda for your shell `bash` or `zsh`:-

```
~/miniconda3/bin/conda init bash
```

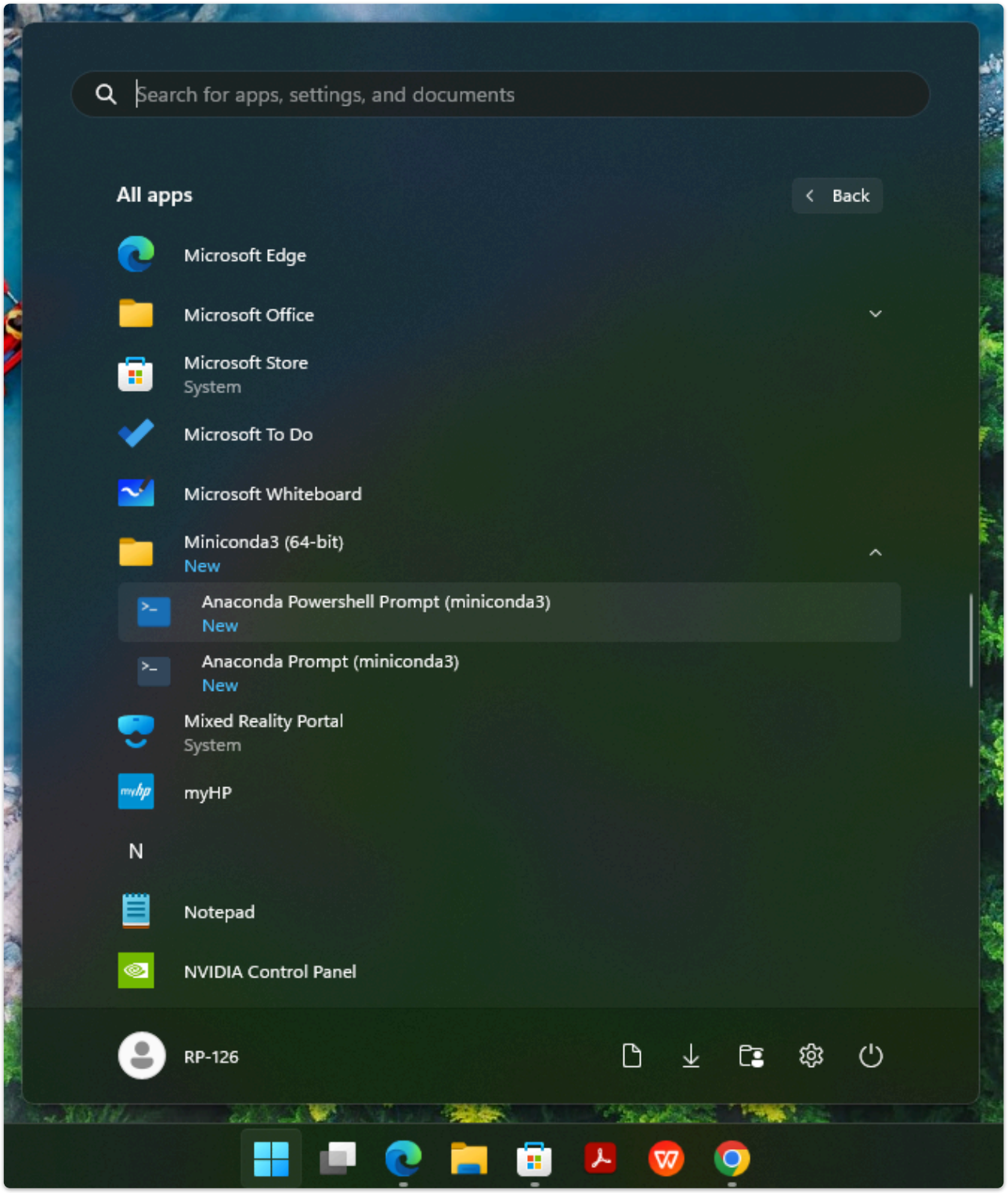
```
~/miniconda3/bin/conda init zsh
```

if you get another shell name after running `echo $0` command the replace shell name in above command after `init` to the shell name that you get.

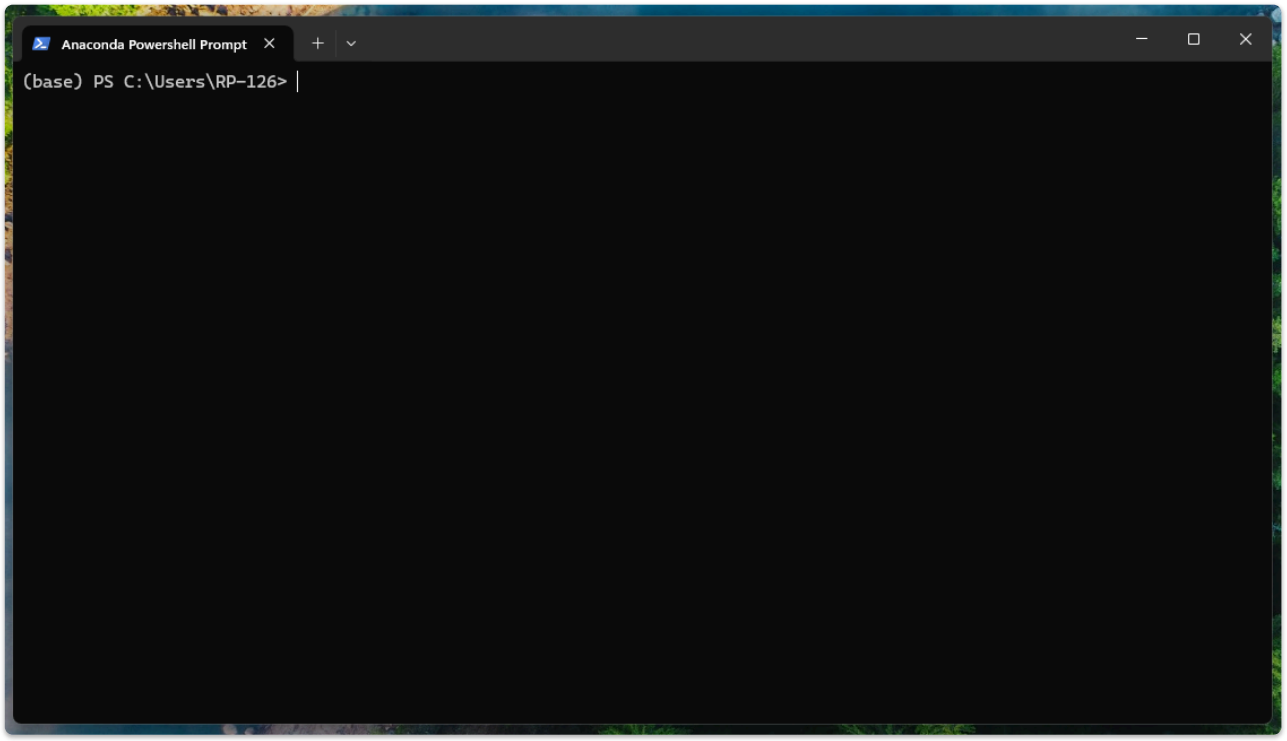
```
~/miniconda3/bin/conda init <your shell name>`
```

2.) Open Anaconda PowerShell Prompt(Miniconda3):-

After installation of miniconda3, in your program list you will get a folder created with name of *Miniconda3* this will contains some applications but we have to only deal with *Anaconda PowerShell Prompt(Miniconda3)* so now we will open this.



Anaconda Powershell Prompt (miniconda3) in start menu



Miniconda PowerShell Prompt

3.)Checking if conda is installed successfully:-

After this we can check if this is the conda prompt or not by running `conda --version` and it will return the conda version information which confirms that we have conda in our prompt. Also, alternatively you can run `python --version` command and this will return the python version which is installed by default in the base or global environment when we have installed miniconda3 as miniconda3 comes with a base python.

4.) Adding `conda-forge` Channel:-

After this in this same prompt we will run the following command:-

```
conda config --add channels conda-forge
```

This will add `conda-forge` as a channel or in easy terms we can say source from which `conda` will install packages that we require.

5.) Project Folder Creation:-

After this we make a folder in our windows filesystem or Linux or macOS file system where we want out programs that we write to be stored.

Example a folder is created in documents folder

`C:\Users\tavne\Documents\data_science_lab_practicals`

> ⚠ **Caution**
>
> Try to avoid spaces in the folder name and instead of spaces use `_` instead.

6:) Changing the location of the terminal prompt to our newly created folder:-

Now we will go back to our terminal (*Anaconda PowerShell Prompt(Miniconda3)*) ; Now we will go to the folder which we have created in previous step by using following command:-

```
cd full\path\to\the\folder\created
```

Example of the above command in our scenario will be:-

```
cd C:\Users\tavne\Documents\data_science_lab_practicals
```

Now after running the above command you can see this whole path in before prompt in your terminal `>` this indicated that currently we are inside of our newly folder.

7.) New Virtual Environment Creation:-

Now we will create a new virtual environment which will help us to keep our environment separate from the global environment and this approach helps us to easily manage our projects and the libraries needed for them as each project will have its separate environments that contains packages of different versions. This separation made using virtual environment helps us to resolve conflicts between dependences/libraries of different projects that might occurs if we install dependencies in global space. Here we will create environment using `--prefix` which will allow us to keep the environment/ libraries in the specific location we want and here we keep our environment in subfolder of our project. Also, here we will also specify the python version that we want in our environment which is `pyhton=3.10`, this gives us more control over the environment. If we don't specify any python version then conda will create an environment with latest python version available.
Syntax for command for creating environment using `--prefix` is:-

```
conda create --prefix full\path\to\env pyhton=3.10
```

Example of above command in our scenario will be:-

```
conda create --prefix
C:\Users\tavne\Documents\data_science_lab_practicals\data_science_lab_practicals_
env pyhton=3.10
```

in above command the new virtual environment with `python=3.10` is created within our `data_science_lab_practicals` folder and all the libraries/ dependencies that we install further in this environment will be stored in `data_science_lab_practicals_env` folder which is inside our main project folder. So, here the environment is contained in our main project folder.

8.) Created Environment Activation:-

After creation of our environment we need to activate it in order to work on that environment and then we will install our required packages in this environment.
Syntax for this command is (for environment created using `--prefix` flag):-

```
conda activate full\path\to\environment
```

Example of above command in our scenario will be:-

```
conda activate
C:\Users\tavne\Documents\data_science_lab_practicals\data_science_lab_practicals_
env
```

After activation the further `conda` command we run will be executed under this environment.

Additionally, we can also check if our newly created environment is activated or not by using following command:-

```
coda info --envs
```

This will list out all the environments available on your system along with the environment locations within which the packages are installed. The environment with an asterisk `(*)` Infront of it is currently activated and in the case if our environment is properly activated the `*` will be Infront of our newly created environment.

9.) Installing Required packages/libraries:-

After Environment creation we will install libraries that we require on our program.
The following command is used to install required package:-

```
conda install packagename
```

Example
If we want to install scikit-learn, the above command for that will be:

```
conda search scikit-learn
```

After running the above command the `scikit-learn` library will be installed under our `C:\Users\tavne\Documents\data_science_lab_practicals\data_science_lab_practicals_env` environment.

similarly,

```
conda install numpy
```

```
conda install pandas
```

```
conda install jupyterlab
```

```
conda install matplotlib
```

the above separate commands will be executed individually and scikit-learn, NumPy, pandas, jupyterlab and matplotlib will be installed in our current active environment which is `C:\Users\tavne\Documents\data_science_lab_practicals\data_science_lab_practicals_env`
Also, we can install all these packages/ libraries in single command and that command will be:-

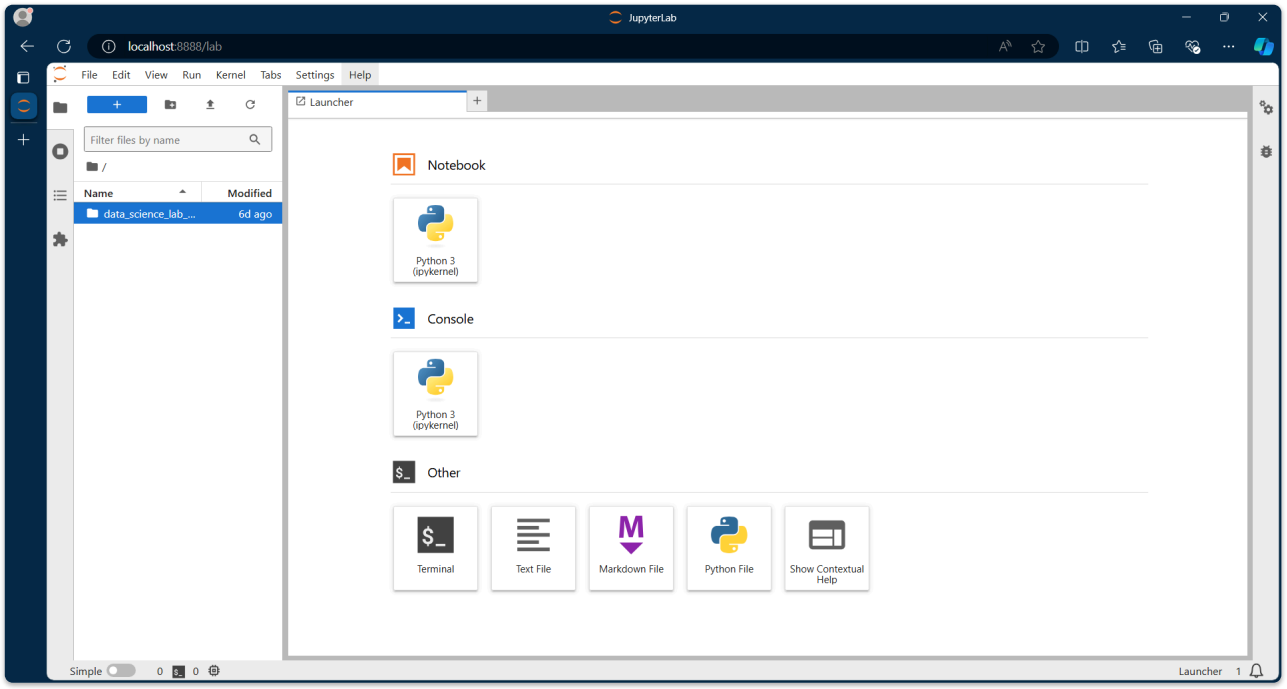
```
conda install scikit-learn numpy pandas jupyterlab matplotlib
```

this above command will install each the listed packages which are separated by space in just a single line. Also, we can modify this line and add more required packages in this command which are separated by spaces and this will allow us to install required packages in just a single command.

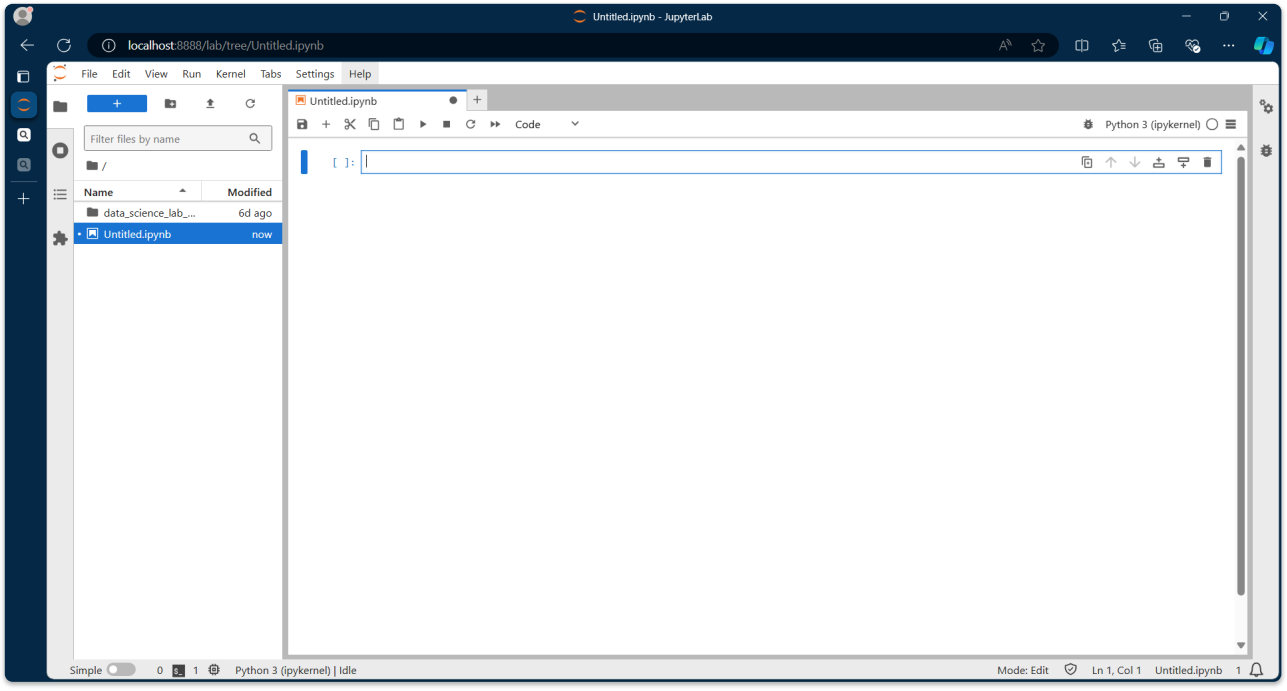
10.) Opening Jupyter-Lab:-

After all the required packages including jypyterlab is installed in our environment we then type on our terminal `jupyter-lab` which will open a web

based UI in the default browser (Following Picture shows the Jupyter Lab UI where we create `jupyter notebooks` in which we will write our programs.



Jupyter-Lab User-Interface(UI)



Jupyter-Notebook UI