installation

Great! ✅ Now that you've installed **Python 3.10**, let's walk through the full step-by-step process to:

1. ✅ Create a virtual environment with Python 3.10
2. ✅ Activate it
3. ✅ Install libraries from `requirements.txt`

🔧 Step-by-Step Instructions

✅ Step 1: Confirm Python 3.10 is available

Open your terminal and run:

```
py -3.10 --version
```

Expected output:

```
Python 3.10.x
```

✅ Step 2: Create a virtual environment using Python 3.10

```
py -3.10 -m venv myenv
```

> This will create a virtual environment folder named myenv.

✅ Step 3: Activate the virtual environment

On **Windows** (PowerShell or CMD):

```
myenv\Scripts\activate
```

You should see your prompt change to:

```
(myenv) PS F:\path\to\your\project>
```

✅ Step 4: Create or update your `requirements.txt`

Create a file named `requirements.txt` in your project folder (if not already created) and add this:

```
tensorflow>=2.10,<2.13
pandas
numpy
scikit-learn
matplotlib
streamlit
scikeras
nltk
```

✅ Step 5: Install all required libraries

Once the virtual environment is activated, run:

```
pip install -r requirements.txt
```

This will install all the required packages into your virtual environment.

✅ Step 6: Test installation

Run Python inside the environment:

```
python
```

Then test TensorFlow and Pandas:

```
import tensorflow as tf
import pandas as pd
print(tf.__version__)
print(pd.__version__)
```

To exit Python:

```
exit()
```

🛑 To deactivate the environment later

```
deactivate
```

Let me know if you want to also add Jupyter Notebook or VS Code integration with this virtual environment!

Perfect! Let's install **TensorFlow and all other libraries** from your `requirements.txt` the correct way.

✅ Step-by-Step: Install from `requirements.txt`

✅ 1. Make sure your virtual environment is activated

In terminal:

```
myenv\Scripts\activate
```

You should see something like:

```
(myenv) PS F:\0vocational training\classes\classes_ka\tensorflow>
```

✅ 2. Edit your `requirements.txt`

Open the file and make sure it has the correct TensorFlow version range that supports Python 3.10:

```
tensorflow>=2.10,<2.13
pandas
numpy
scikit-learn
matplotlib
streamlit
scikeras
nltk
```

Make sure there's **no** `tensorflow==2.1`, as that version will not work.

✅ 3. Install all packages

Run this:

```
pip install -r requirements.txt
```

Let the installation finish (it may take a few minutes because of TensorFlow).

✅ 4. Verify that everything installed

You can run:

```
pip list
```

Or check TensorFlow specifically:

```
pip show tensorflow
```

Or run:

```
python
```

Then inside Python:

```
import tensorflow as tf
print(tf.__version__)
```

Let me know if you'd like me to verify your current `requirements.txt` content or help debug errors during installation.

Perfect! You're all set up. Here's how to **organize your project** and **run your first program** step-by-step.

🗂️ Current Folder Structure:

```
F:\0vocational training\classes\classes_ka\tensorflow\
│
├── requirements.txt      ✅ Contains your dependencies
└── myenv\                ✅ Your virtual environment (do NOT write code here)
```

✅ Step 1: Create a folder for your code

Create a folder for your scripts to keep things organized.

In terminal (PowerShell):

```
mkdir app
```

Now your structure looks like:

```
F:\0vocational training\classes\classes_ka\tensorflow\
├── app\                ✅ ✅ Write your Python programs here
├── myenv\              ❌ Do NOT write anything here
└── requirements.txt
```

✅ Step 2: Create a simple program

Inside `app/`, create a file called `main.py`.

📄 `main.py` content:

```
import tensorflow as tf
import numpy as np
import pandas as pd

print("TensorFlow version:", tf.__version__)
print("NumPy version:", np.__version__)
print("Pandas version:", pd.__version__)
```

✅ Step 3: Run the program

Make sure your virtual environment is active:

```
myenv\Scripts\activate
```

Then run:

```
python app\main.py
```

✅ You should see:

```
TensorFlow version: 2.12.1
NumPy version: 1.24.3
Pandas version: 2.3.0
```

📁 Bonus: Ideal Project Structure

Later, you can structure your app like this:

```
tensorflow_project/
├── myenv/              (virtual env)
├── app/
│   ├── main.py         (your scripts)
│   └── utils.py        (helper functions if needed)
├── data/               (for datasets)
├── models/             (saved models)
├── requirements.txt
└── README.md
```

Let me know if you want to build a simple ML model next, or need a sample notebook or Streamlit app!