Certainly! If your professor has suggested installing Miniconda on Moore and using it to manage your Python environment without requiring sudo access, here are the general steps you can follow:

### 1. Download Miniconda:

- Go to the Miniconda website: [https://docs.conda.io/en/latest/miniconda.html](https://docs.conda.io/en/latest/miniconda.html)

- Download the Miniconda installer for Linux (probably for 64-bit architecture).

- Use `wget` or `curl` to download the installer directly to Moore:

```bash

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

```

or

```bash

curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

```

### 2. Run the Installer:

- Run the Miniconda installer script:

```bash

bash Miniconda3-latest-Linux-x86\_64.sh

```

- Follow the prompts in the installer. You'll be asked to review the license terms and choose the installation location. It's often a good idea to accept the default options.

### 3. Initialize Miniconda:

- Close and reopen your terminal to activate the changes made by the installer.

- You should see `(base)` in your terminal prompt, indicating that the base environment is active.

### 4. Create a Conda Environment:

- Create a new Conda environment for your project. Replace `your\_env\_name` with a name for your environment and specify the Python version you want (e.g., `python=3.8`):

```bash

conda create --name your\_env\_name python=3.8

```

### 5. Activate the Environment:

- Activate your Conda environment:

```bash

conda activate your\_env\_name

```

- You should now see your environment name in the terminal prompt.

### 6. Install Packages:

- Use `conda` to install Python packages within your environment:

```bash

conda install package\_name

```

- Replace `package\_name` with the name of the package you want to install.

### 7. Deactivate the Environment:

- When you're done working in your Conda environment, deactivate it:

```bash

conda deactivate

```

These steps should help you set up and manage your Python environment using Miniconda on Moore without needing sudo access. If you have any specific requirements or encounter issues, feel free to provide more details, and I'll be happy to assist you further.

It appears that the error is indicating that the "jupyter" package is not installed in the specified virtual environment ("your\_env\_name"). Here are steps to address this issue:

1. \*\*Install Jupyter in the Virtual Environment:\*\*

Run the following command to install the Jupyter package in your virtual environment:

```bash

conda install -n your\_env\_name jupyter

```

This command will install Jupyter Notebook in the specified environment.

2. \*\*Activate the Virtual Environment:\*\*

After installing the Jupyter package in your environment, make sure to activate the environment before launching Jupyter Notebook:

```bash

conda activate your\_env\_name

```

3. \*\*Launch Jupyter Notebook:\*\*

Once the environment is activated, you can try launching Jupyter Notebook again:

```bash

jupyter notebook

```

This should start Jupyter Notebook in the specified environment.

If you encounter any issues during the installation process, make sure to check for error messages and address them accordingly. If the problem persists, consider creating a new virtual environment and repeating the steps above or seeking assistance from relevant forums or communities for further troubleshooting.

Using Moore which is an hpc

|  |
| --- |
| activate conda environment |
| write jupyter notebook |
| open another cmd |
| write ssh -L port number:localhost:portnumber moore |
| now copy local house list in another browser |

Creating conda environment in Windows:

To create a virtual environment in Windows, you can use Python's built-in tool called `venv`. Here's a step-by-step guide on how to do it:

1. \*\*Open Command Prompt\*\*: Press Win + R, type `cmd`, and hit Enter.

2. \*\*Navigate to Your Project Directory\*\*: Use the `cd` command to navigate to your project directory. For example, if your project is located in `C:\Users\YourUsername\Projects`, you would type:

```

cd C:\Users\YourUsername\Projects

```

3. \*\*Create Virtual Environment\*\*: Once you're in your project directory, run the following command to create a virtual environment named `env`:

```

python -m venv env

```

This command will create a new directory called `env` in your project folder, containing a copy of the Python interpreter, the standard library, and various supporting files.

4. \*\*Activate Virtual Environment\*\*: After the virtual environment is created, you need to activate it. In Command Prompt, run:

```

.\env\Scripts\activate

```

You'll notice that your command prompt changes, indicating that the virtual environment is active. You'll see the name of the virtual environment in parentheses at the beginning of your command prompt.

5. \*\*Install Packages\*\*: With the virtual environment activated, you can now install packages without affecting the global Python installation. For example:

```

pip install package\_name

```

6. \*\*Deactivate Virtual Environment\*\*: When you're done working in the virtual environment, you can deactivate it by running:

```

deactivate

```

That's it! You've successfully created a virtual environment in Windows using `venv`.