

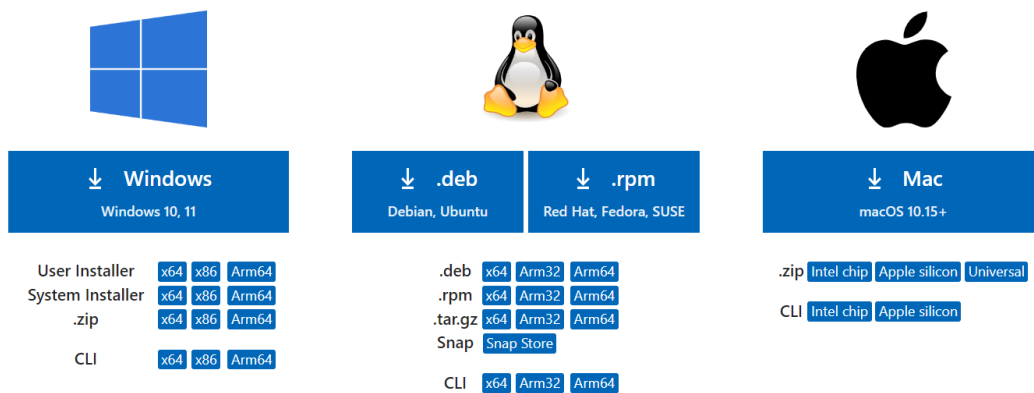
Turing GPU and Environment Tutorial

1. Vscode Installation:

- Download and setup Vscode on your laptop using this link:
<https://code.visualstudio.com/download>.
- Choose the version appropriate for your system (Windows or Mac).

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

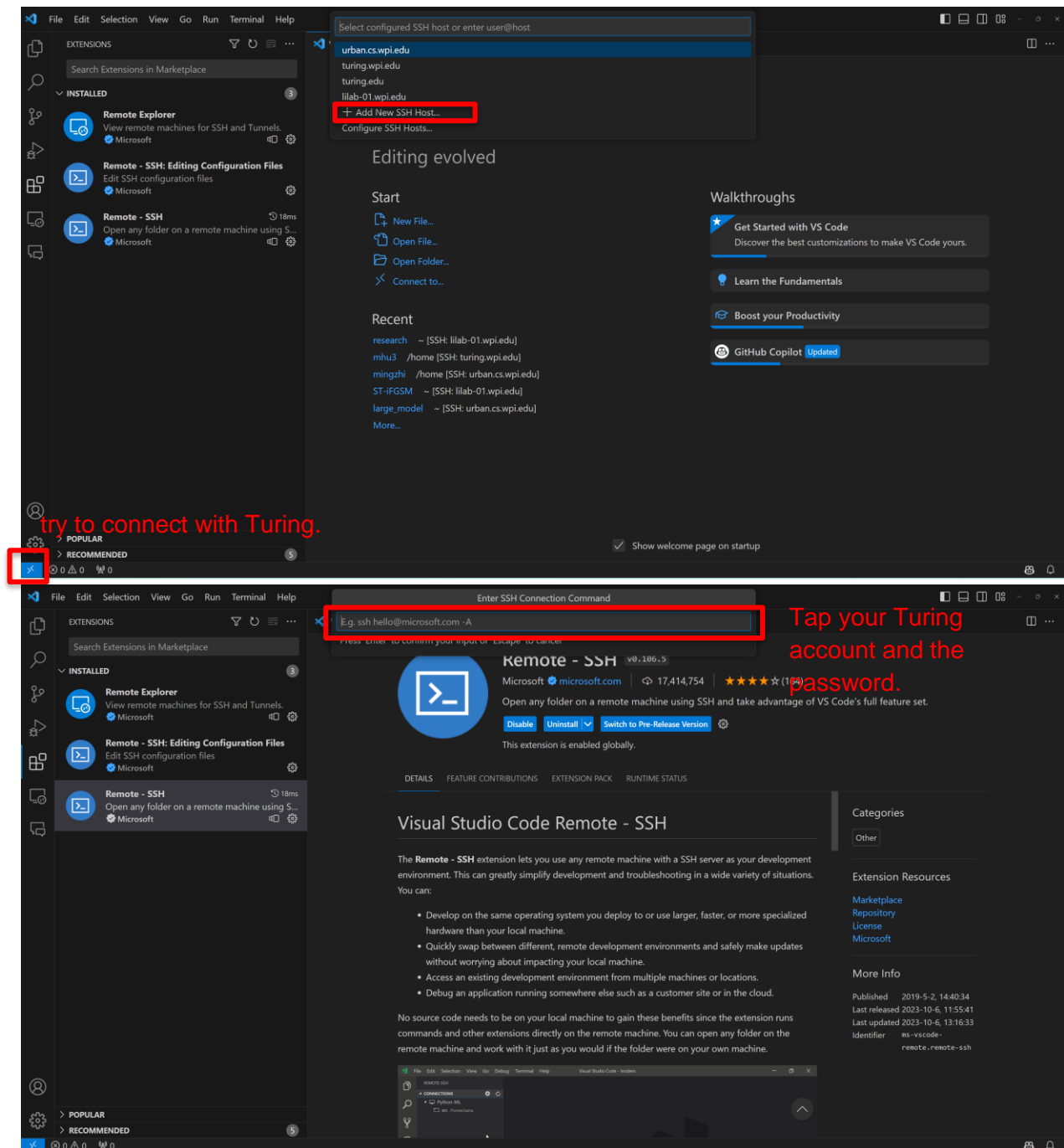


2. Connect to the WPI Network

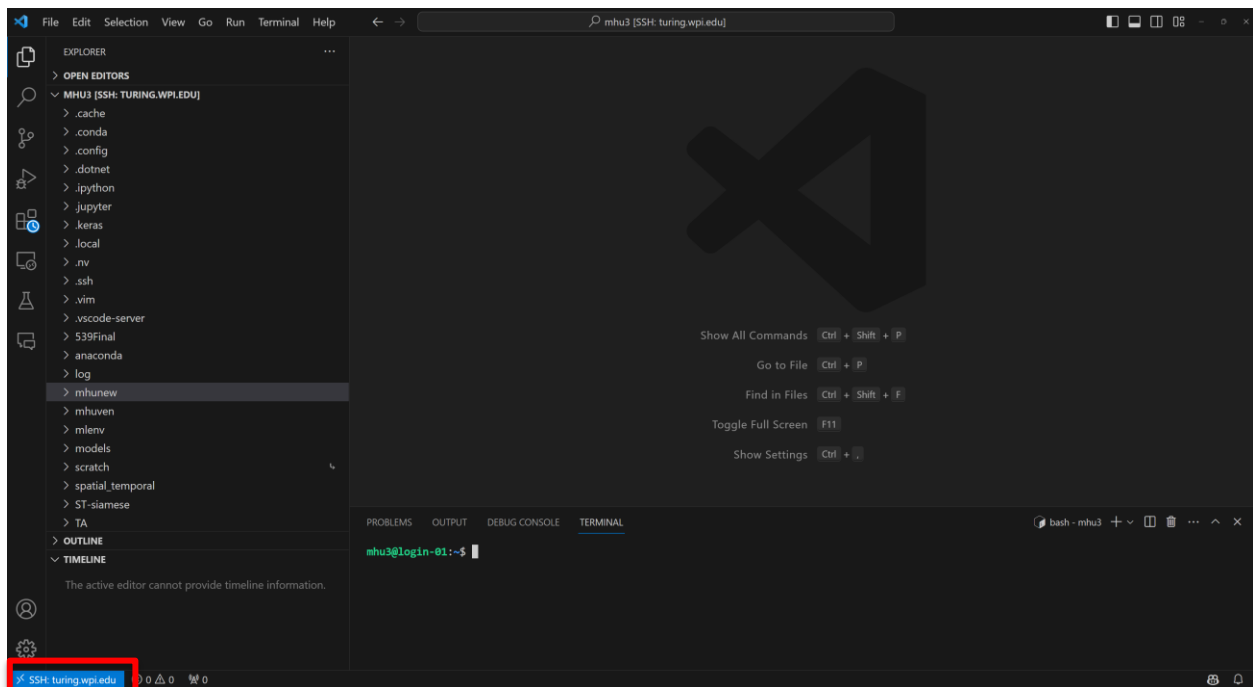
- Connect directly if you're on campus.
- Use the VPN following (<https://hub.wpi.edu/article/444/globalprotect-vpn-client-configuration>).

3. Open Vscode and try to connect to the Turing:

Vscode will automatically install the extensions of SSH for you.

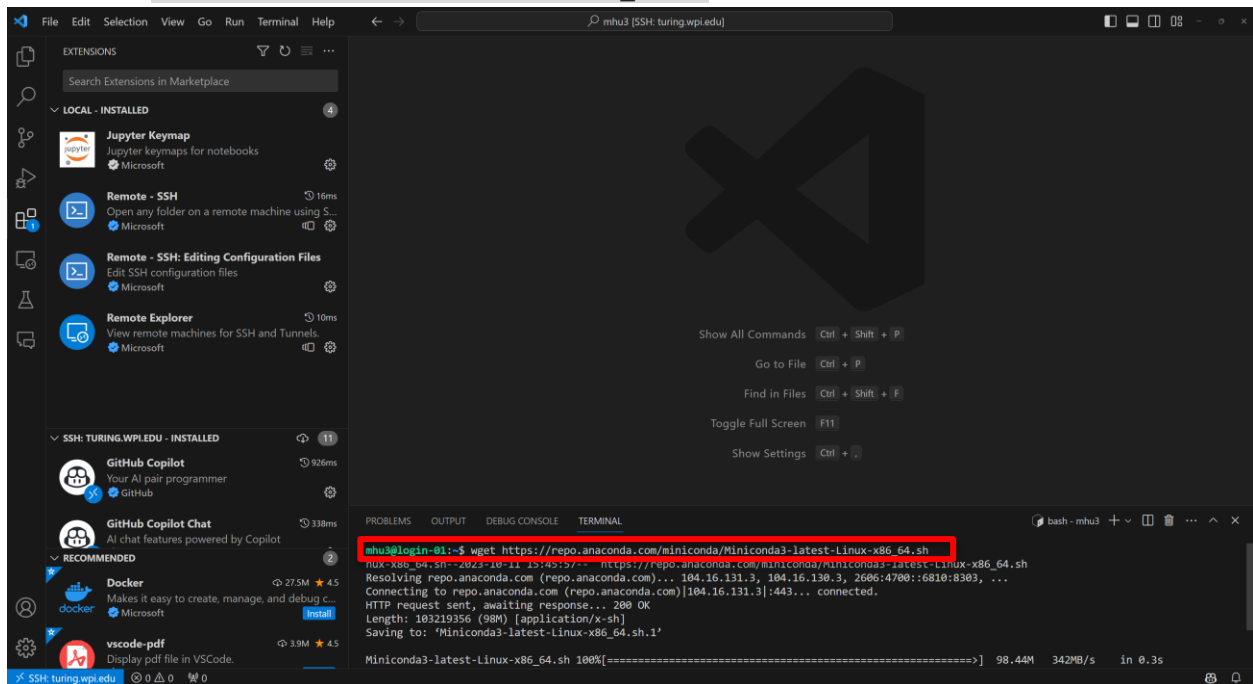


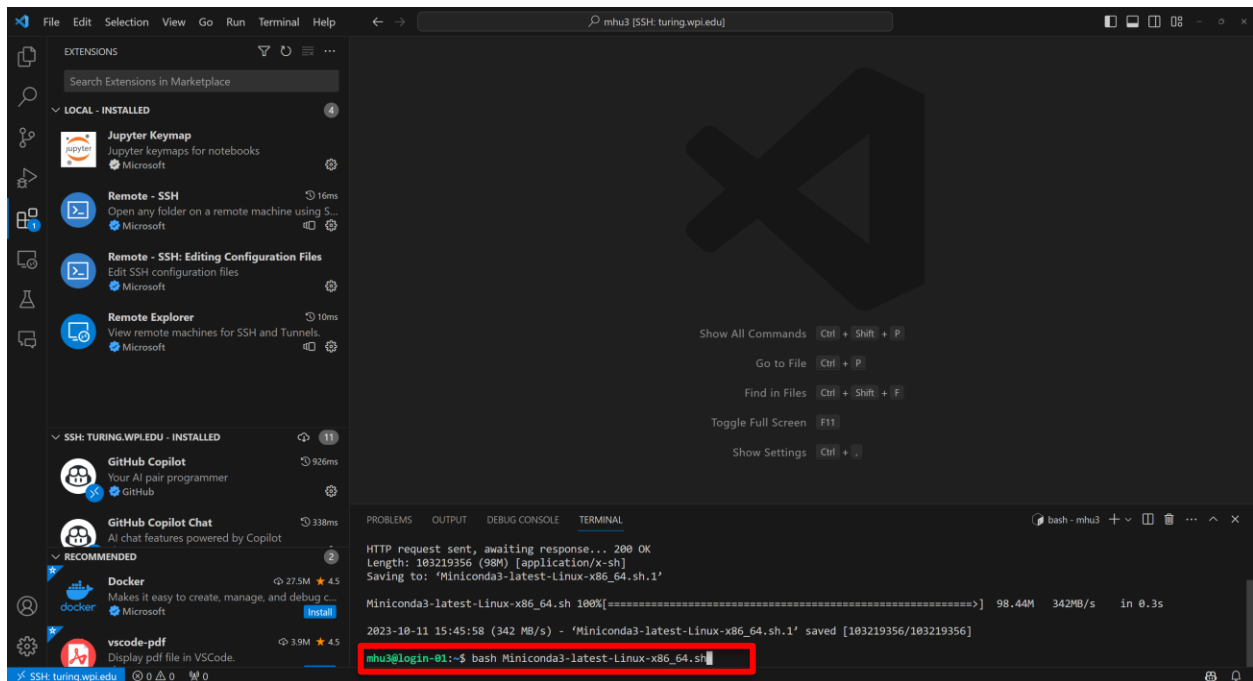
Now you are connected to Turing.



4. Download conda to the server and install it.

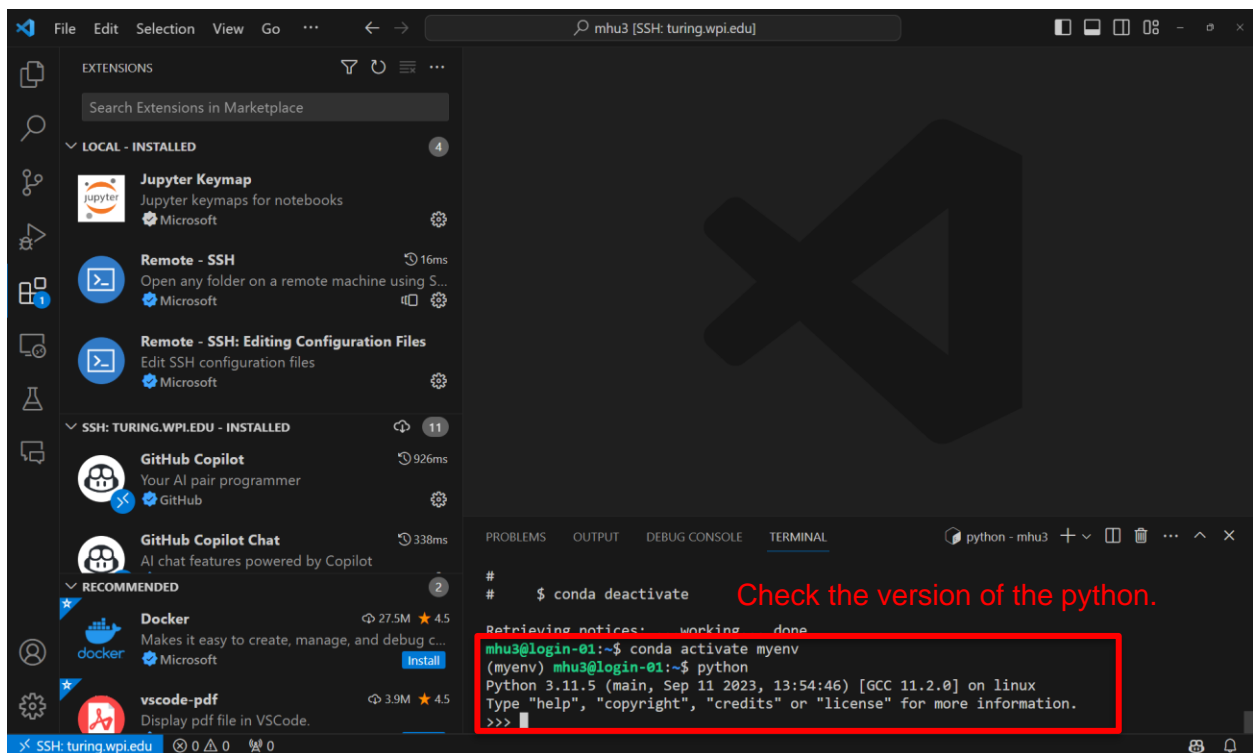
- `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`
- `bash Miniconda3-latest-Linux-x86_64.sh`



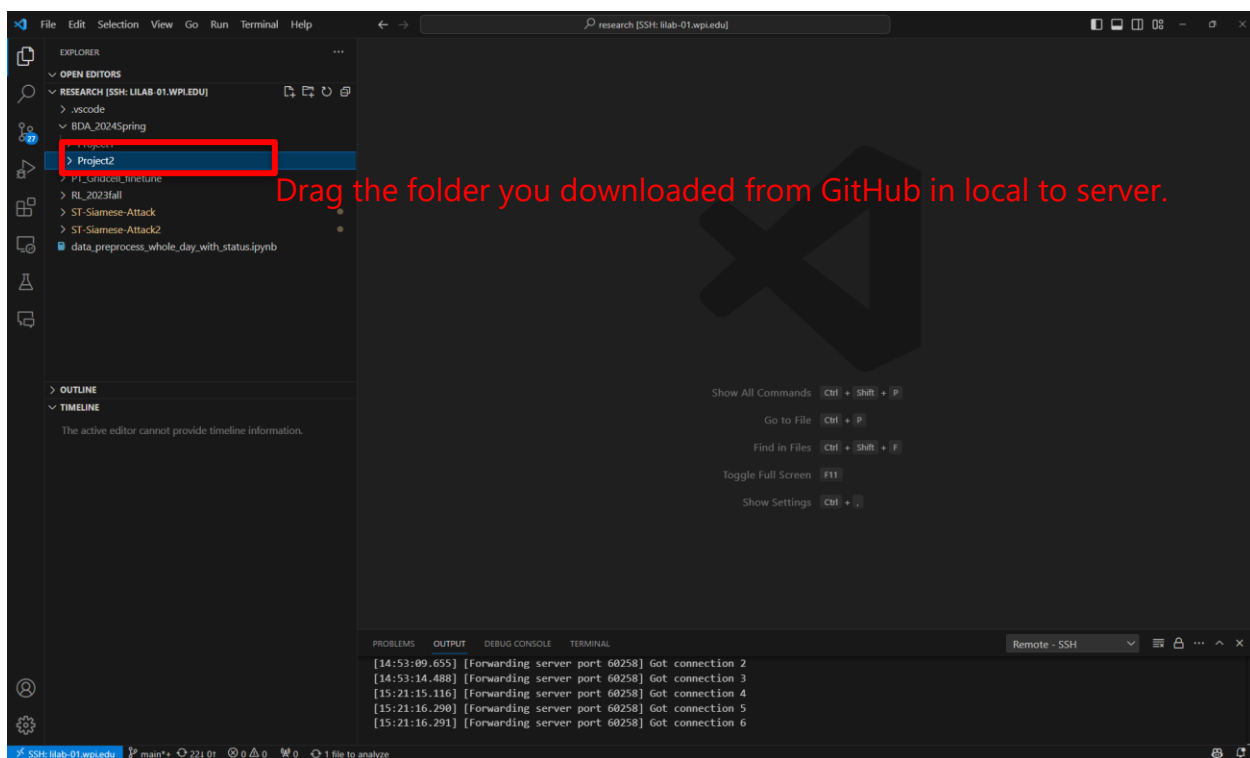
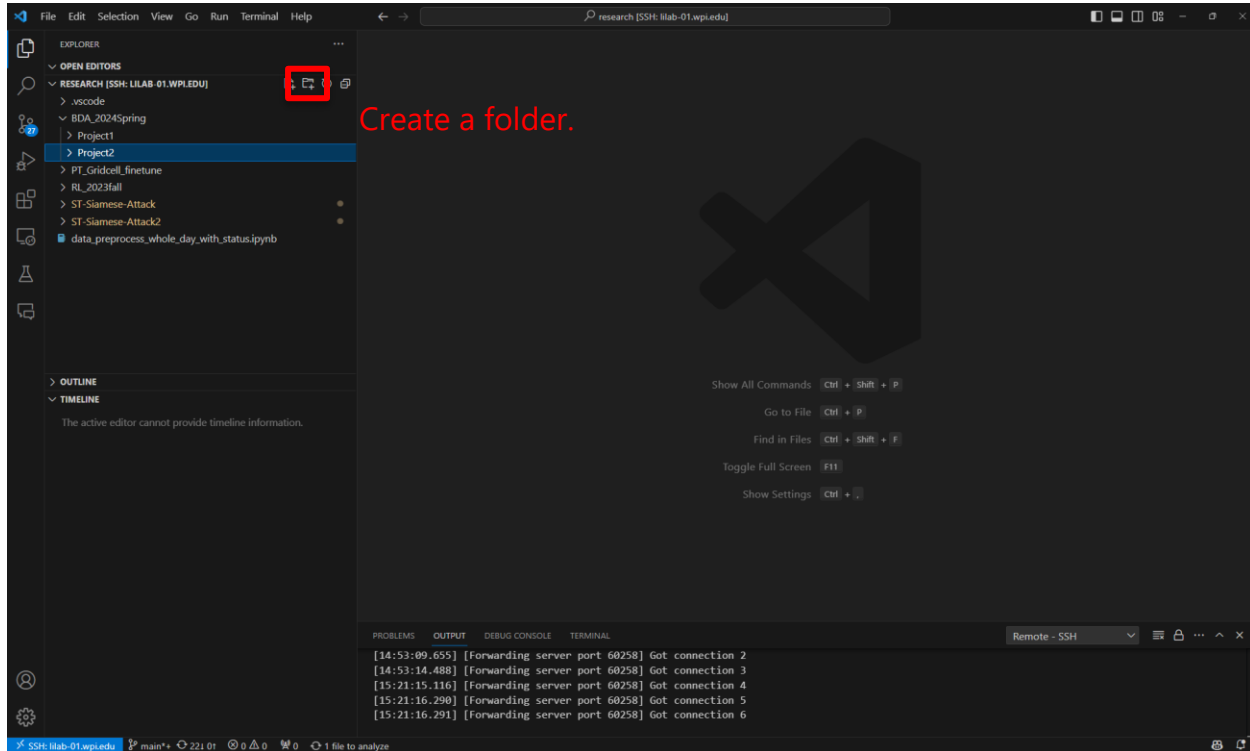


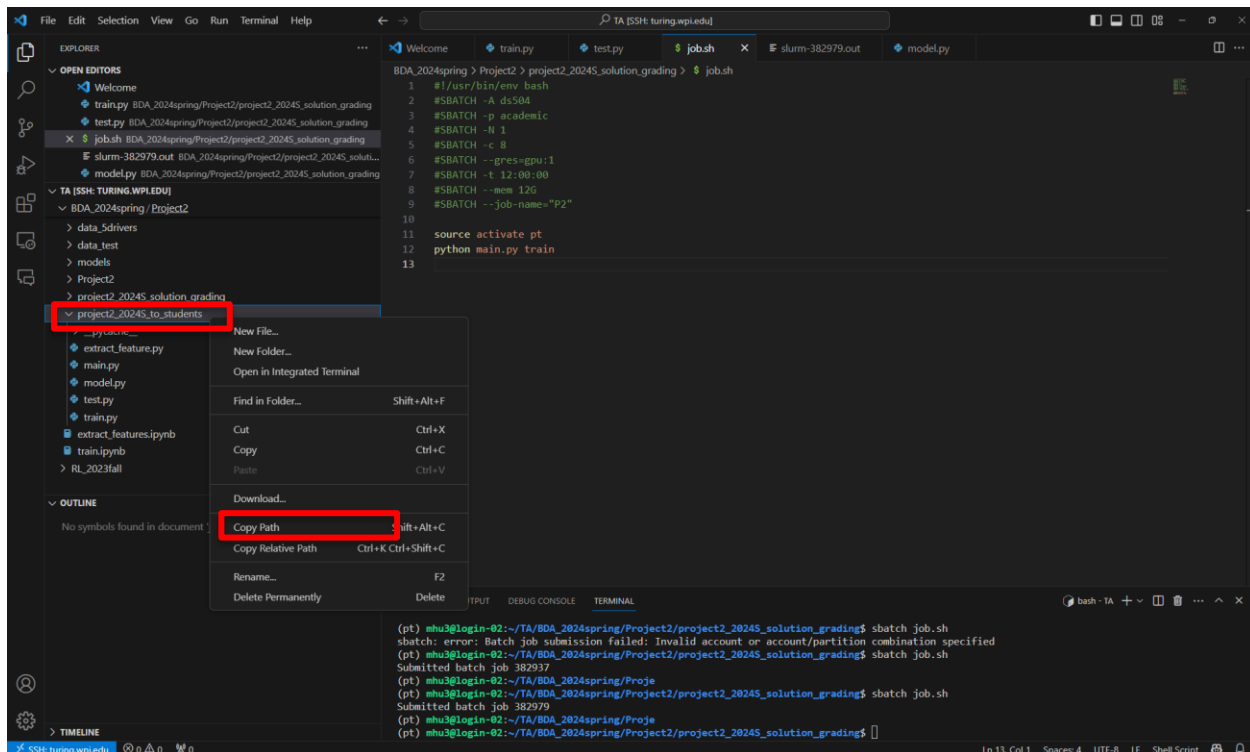
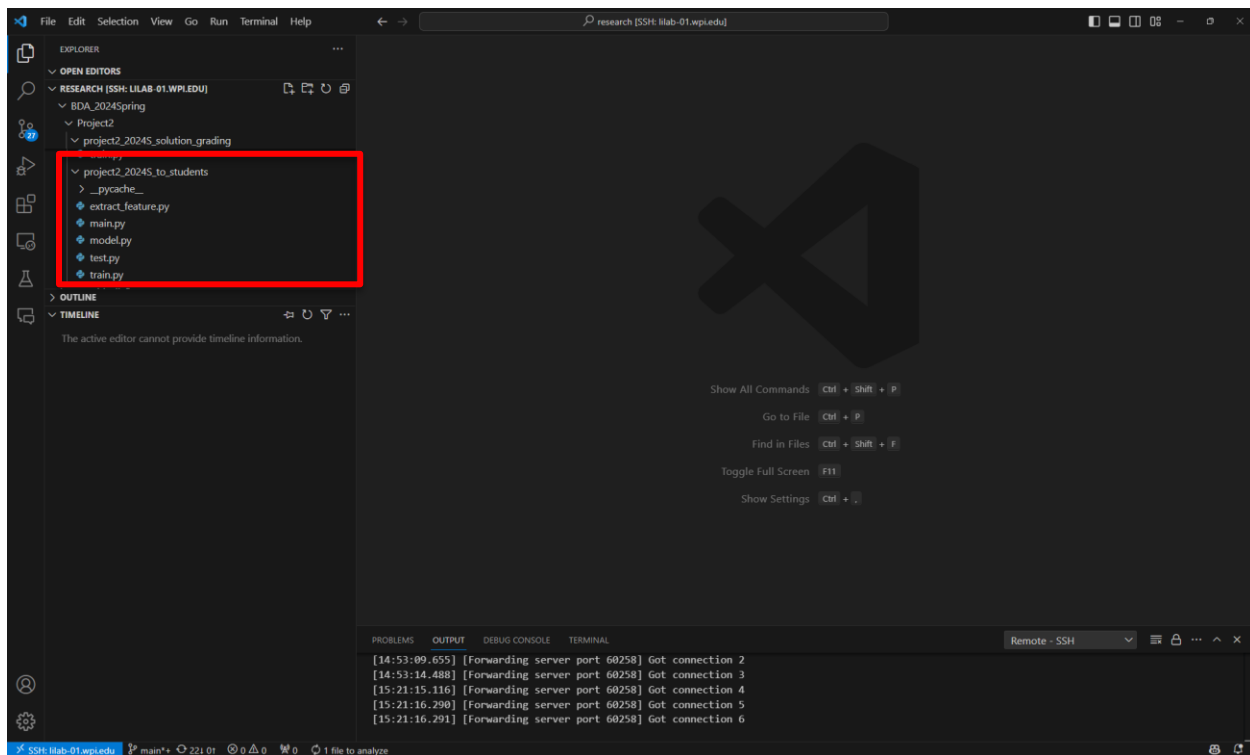
After the installation is done, restart your bash by typing **bash** if needed. conda will create an environment named "base" for you.

5. Create virtual environment: `conda create -n myenv python==3.11.5.`
6. After creating the virtual environment, activate it: `conda activate myenv.` Please install the required packages in the environment.

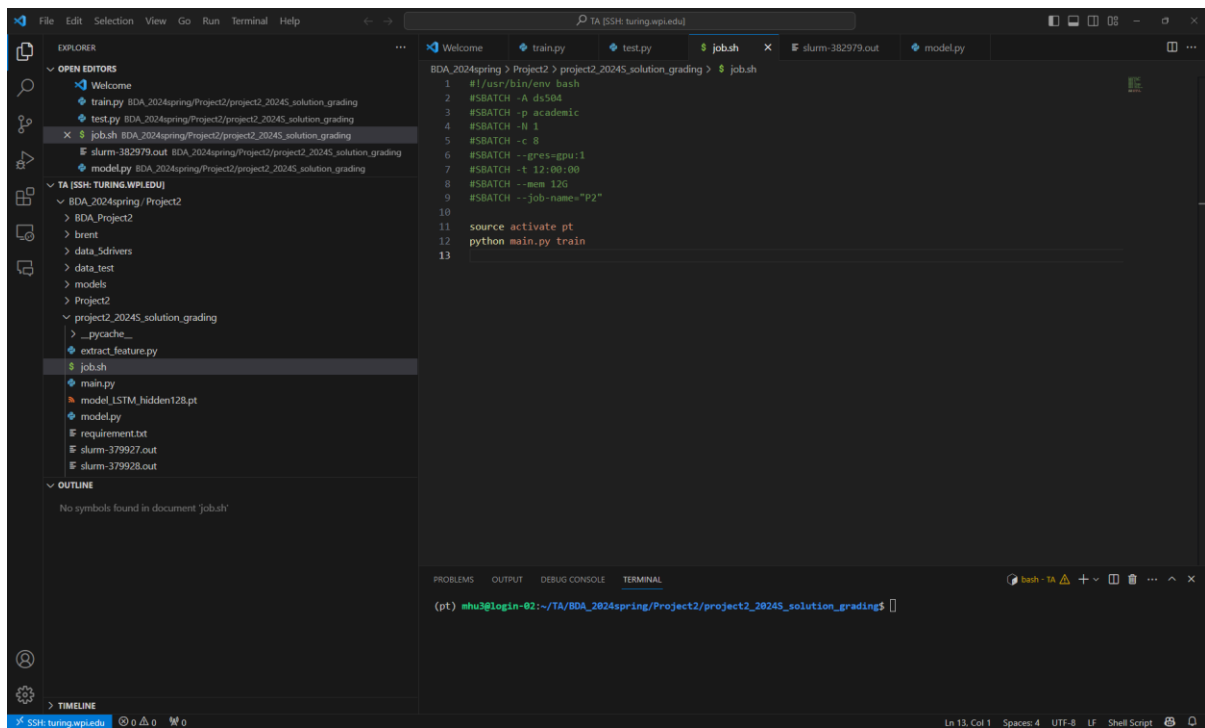


7. Project Directory Setup. Create a folder for the project.





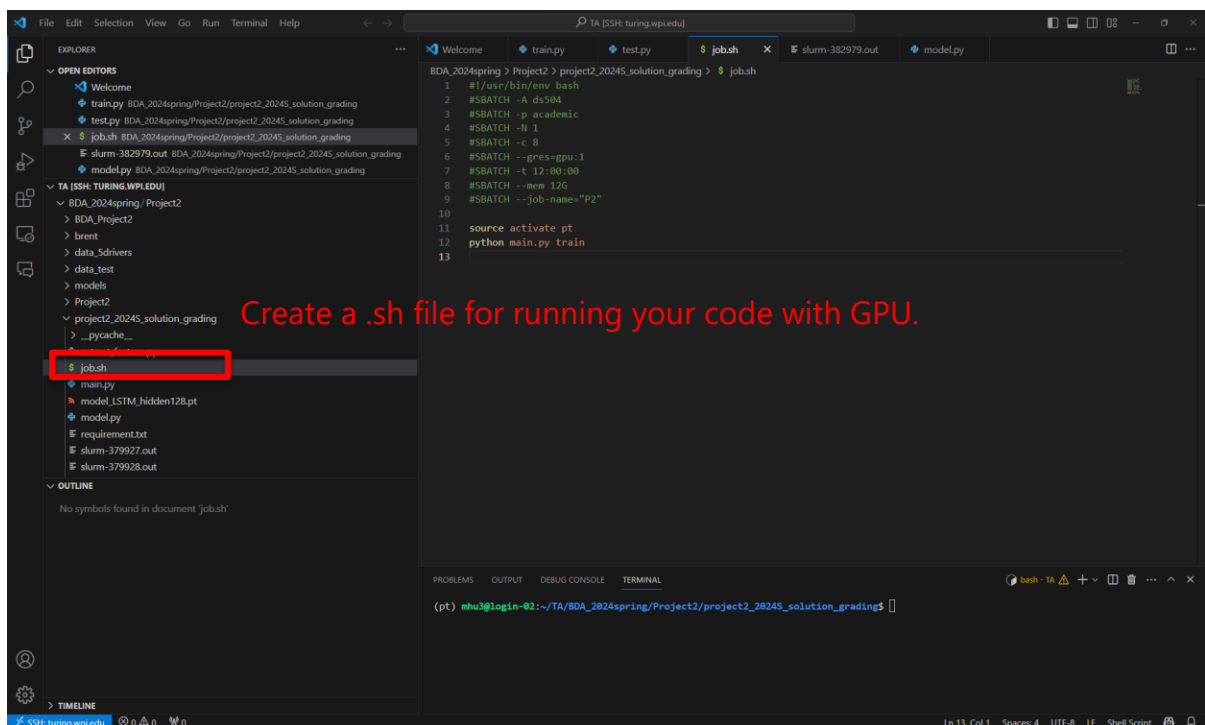
8. Navigate to the Directory using `cd path` ("path" is the dictionary you have copied.)



9. Prepare Your Job:

Ensure you have a script (e.g., job.sh) ready. This script should specify the necessary resources (like CPU cores, RAM, GPU) and contain the instructions for running your job. Please make sure you have the following lines in the script.

```
#SBATCH -A ds504
#SBATCH -p academic
```

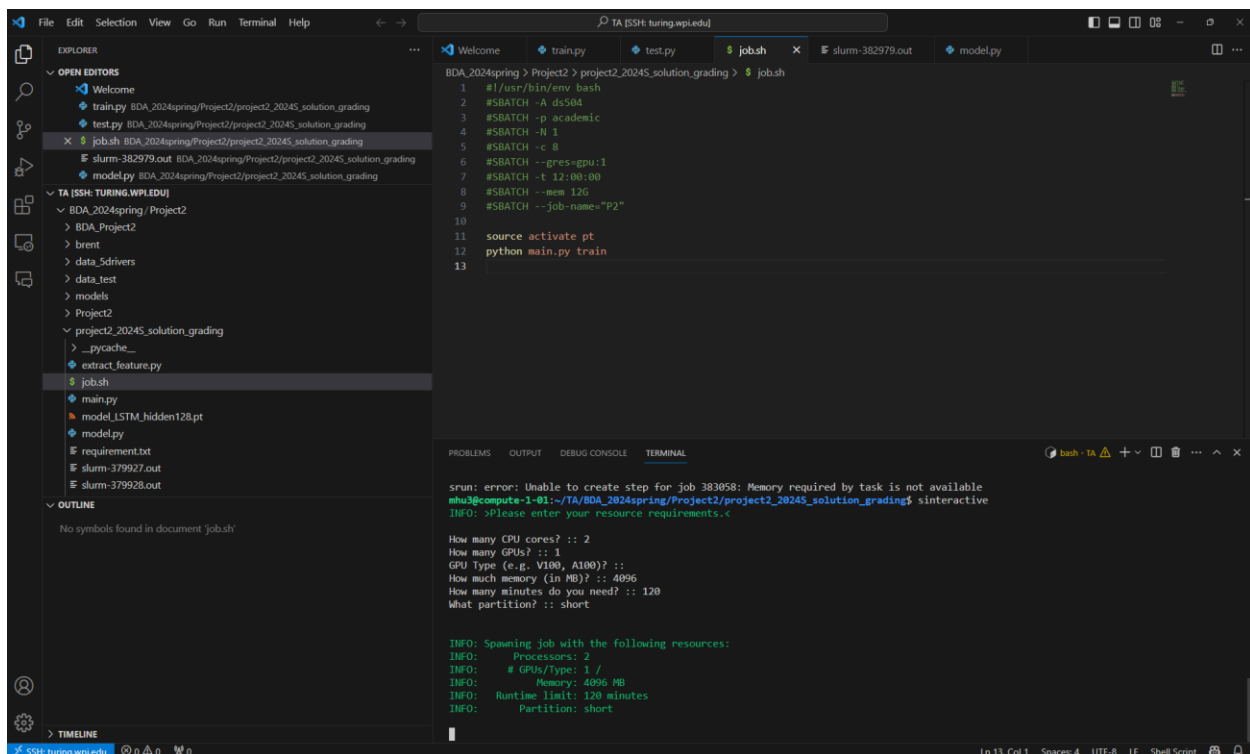


10. Use the following command to submit your job: `sbatch job.sh`.
11. Launch an Interactive Session for Testing: Use `sinteractive`.

This command is particularly useful when you want to test your code on the server's resources before running the actual job. Note that in an interactive session:

- Your terminal session must remain active for the duration of the session.
- Any code you run will execute immediately, unlike the batch mode (`sbatch`), which queues your job to be run based on the server's schedule and available resources.

After running this command, wait for the server (in this case, Turing) to allocate resources to you, such as a GPU. Once resources are assigned and your prompt changes, you're in an interactive session and can run commands as if you're on the allocated node.



The screenshot shows a VS Code editor interface with a terminal window open. The terminal displays the following commands and output:

```
1 #!/usr/bin/env bash
2 #SBATCH -A ds504
3 #SBATCH -p academic
4 #SBATCH -N 1
5 #SBATCH -c 8
6 #SBATCH --gres-gpu:1
7 #SBATCH --time=12:00:00
8 #SBATCH --mem=126
9 #SBATCH --job-name="P2"
10
11 source activate pt
12 python main.py train
13
```

The terminal output shows an error message from `srund` indicating that the memory required by the task is not available. The user then runs `sinteractive`, which prompts them to enter their resource requirements. The user enters the following requirements:

```
How many CPU cores? :: 2
How many GPUs? :: 1
GPU Type (e.g. V100, A100)? ::
How much memory (in MB)? :: 4096
How many minutes do you need? :: 120
What partition? :: short
```

The terminal then displays the following information about the resources allocated for the job:

```
INFO: Spawning job with the following resources:
INFO:   Processors: 2
INFO:   # GPUs/Type: 1 /
INFO:   Memory: 4096 MB
INFO:   Runtime limit: 120 minutes
INFO:   Partition: short
```