



**DEVI  
SANSTHAN**  
Dignity Education  
Vision International  
*Leave no one behind*

Github Link: [github.com/sandyawijayaa/DSS-x-TWB-Devi-S](https://github.com/sandyawijayaa/DSS-x-TWB-Devi-S)

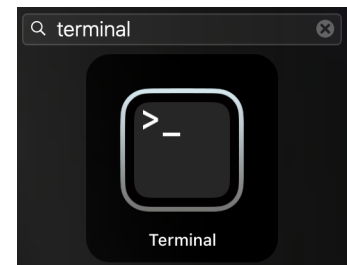
Video Tutorial by Zack: [DeviSansthan Dashboard Tutorial.mp4](#)

## Step-by-step: Mac

### 1) Install a terminal

The terminal is a program that allows you to interact with your computer by entering commands.

If you're on a Mac, you already have a program called `Terminal` or something similar on your computer. Open that up and you should be good to go.



### 2) Install Python 3

Python 3 is the programming language we used in our work. If you already have an older version of Python installed, please make sure to download and install Python3. You can check your Python version with `python3 --version`. [Download Python3 Here](#)

Download and install Python 3 (64-bit). You may need to right-click the download icon and select "Open". After installing, please close and reopen your Terminal.

### 3) Open terminal and see current location

Open "Terminal" and you'll see something like this

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
(base) Zackooooon:~ alexoon$  
(base) Zackooooon:~ alexoon$
```

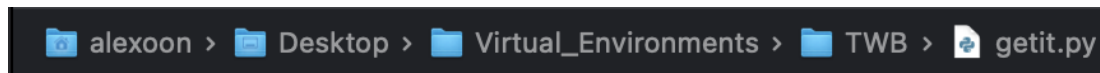
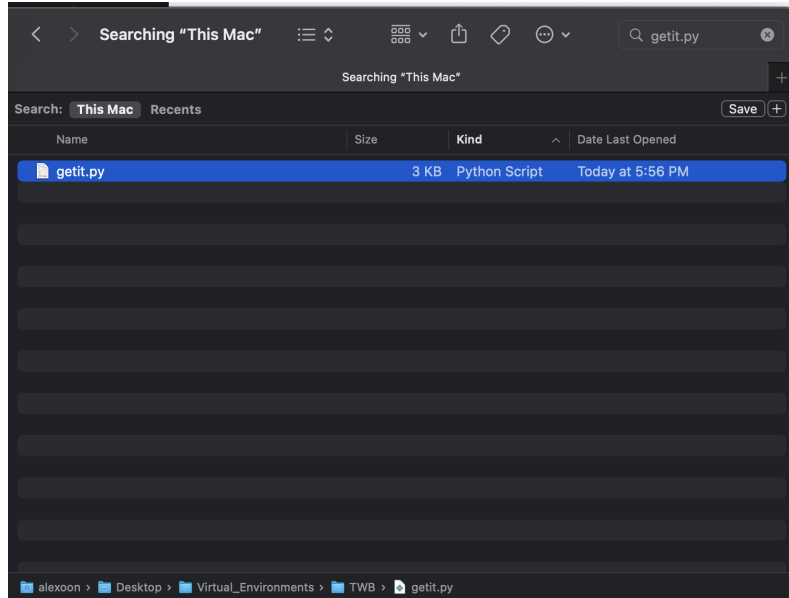
The `~` sign tells us that we are the "home" directory, which is the outermost folder of your file organization system. In the example above, my home directory is named "alexoon"

### 4) Open the project folder

We now want to go to the folder in which your program was saved. You can check where it was saved through your file explorer.



If we go to our file explorer and type in the name of the program that was downloaded (ours was `getit.py`), **[INSERT NAME]**, then **single-click** on the file/folder name, it'll show at the bottom how to get from the “home” directory to the file.



To go from outer folders to the other folders listed above, we will use the following commands:

```
cd ~
```

```
cd Desktop/Virtual_Environments/TWB
```

Now, you will be in the innermost folder (in this example, it is called “TWB”).

You'll only need to do step 5 once. Once you have done it, you won't need to do it even if you update the data. Step 6 should be carried out whenever you want to analyze updated data! If it isn't the first time going through this manual on your computer and you aren't using new data, you can *skip to part 7*.

## 5) Download required dependencies

Before we can open up our dashboard, we will need to download the dependencies required to run our code. It isn't advised to do this directly onto your laptop, so what we want to do is create something called a “virtual environment.” This will download the packages in an isolated folder, so that it doesn't clutter your computer.

To create a virtual environment, in the terminal type (note the last phrase, `twb_venv` is just the name. You can name it anything, as long as it doesn't have a space. Just make sure the name



is consistent throughout the commands below—wherever it says “twb\_venv”, replace it with your chosen name):

```
python3 -m venv twb_venv
```

Now, to download dependencies, do:

```
source twb_venv/bin/activate  
pip install -r requirements.txt
```

Now, all the required dependencies should be installed!

## 6) Load in your data

Finally, before we can use the dashboard, we just need to import the data! The datasets were not published in the github repository since we wouldn't want that information to be public in case it was sensitive. As a result, you'll need to get the data from someone who has access to it (the data should come in a .csv file format).

Once you have obtained the data, drag and drop the data files into the folder labeled “data” within the project folder. If you are having difficulties, the video tutorial will also show you alternative ways besides just dragging and dropping the files. Note that the data folder has a placeholder file called “data\_placeholder.txt” You can ignore that—it's just there to tell you that the data is in the right place!

Finally, now that all the data is in the “data” folder, we want to update the file titles within our code! This may seem very intimidating, but don't worry, it is just a couple copy-pastes.

There are two options: you can (1) change the data file names in your “data” folder — this is the easiest way to go about it, or (2) you can update the file “paths” (where it is located in your computer) in our code!

### For (1): Change file name in your folder

The exact file names we can use are:

- mdl\_h5pactivity\_attempts.csv
- mdl\_student.csv
- mdl\_learner.csv
- learnerRequestData.csv

### For (2): Update file paths in code

To update the file paths, you can do so in a text editor. Specific instructions are found in this section of the document [here](#).

## 7) Generate dashboard

To go from outer folders to the other folders listed above, we will use the following commands:



and you will now type the following command to generate your dashboard, which should open in your browser:

**python3 getit.py**

These instructions don't use the same file names (rather than `getit.py`, it is **demographicsDashboard.py** and **performanceDashboard.py**), but give the same idea to apply to the file names we have provided in the project. If you would like a more in-depth video, we have created one linked here.

## Step-by-step: Windows

### 1) Install a terminal

PowerShell comes pre-installed on Windows and requires no extra setup. You can simply launch it from the Start menu. Simple commands like `cd` and `ls` will work (`python` will work after the setup), which encompass most of the Bash commands needed.

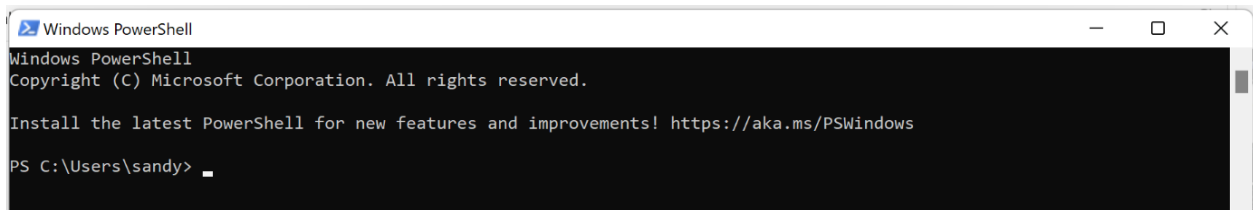


### 2) Install Python 3

**Open the Microsoft Store and search for "python".** Install Python 3.9 by the Python Software Foundation (this should be the first result). You can then skip the rest of this section. (Important: If you later decide to reinstall Python differently, **uninstall it from the Microsoft Store first.**)

### 3) Open terminal and see current location

Open "Terminal" and you'll see something like this



This is our "home" directory, which is the outermost folder of your file organization system. In the example above, my home directory is named "sandy"

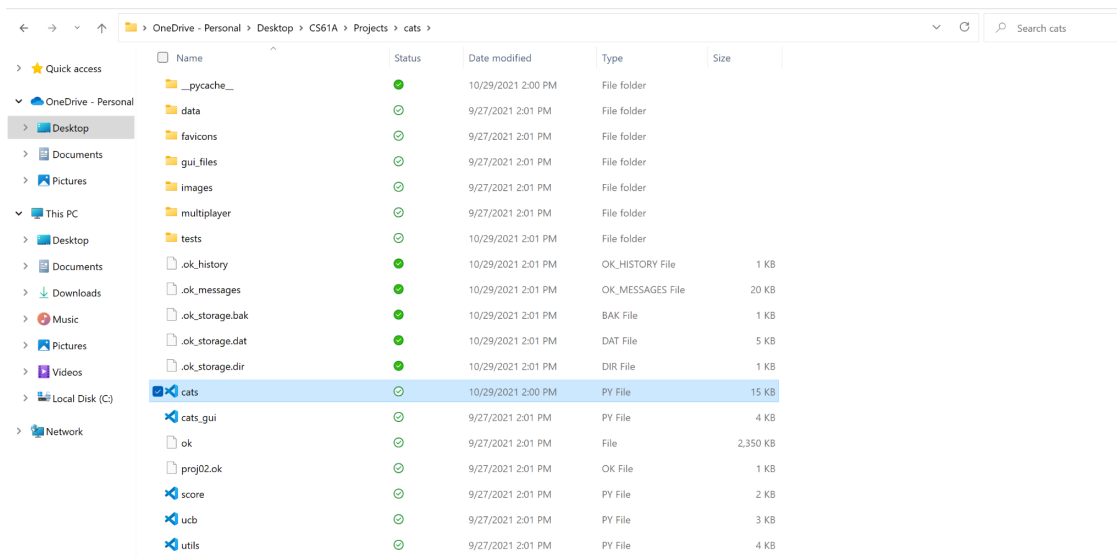


## 4) Open the project folder

We now want to go to the folder in which your program was saved. You can check where it was saved through your file explorer.

If we go to our file explorer and type in the name of the program that was downloaded (ours was `cats.py`), **[INSERT NAME]**, then **single-click** on the file/folder name, it'll show at the top how to get from the “home” directory to the file.

(In this case, the path is OneDrive → Personal → Desktop → CS61A → Projects → cats → `cats.py`)



You'll only need to do step 5 once. Once you have done it, you won't need to do it even if you update the data. Step 6 should be carried out whenever you want to analyze updated data! If it isn't the first time going through this manual on your computer and you aren't using new data, you can *skip to part 7*.

## 5) Download required dependencies

Before we can open up our dashboard, we will need to download the dependencies required to run our code. It isn't advised to do this directly onto your laptop, so what we want to do is create something called a “virtual environment.” This will download the packages in an isolated folder, so that it doesn't clutter your computer.

To create a virtual environment, in the terminal type (note the last phrase, `twb_venv` is just the name. You can name it anything, as long as it doesn't have a space. Just make sure the name is consistent throughout the commands below—wherever it says “`twb_venv`”, replace it with your chosen name):

```
python3 -m venv twb_venv
```



Now, to download dependencies, do:

```
source twb_venv/bin/activate  
pip install -r requirements.txt
```

Now, all the required dependencies should be installed!

## 6) Load in your data

Finally, before we can use the dashboard, we just need to import the data! The datasets were not published in the github repository since we wouldn't want that information to be public in case it was sensitive. As a result, you'll need to get the data from someone who has access to it (the data should come in a .csv file format).

Once you have obtained the data, drag and drop the data files into the folder labeled "data" within the project folder. If you are having difficulties, the video tutorial will also show you alternative ways besides just dragging and dropping the files. Note that the data folder has a placeholder file called "data\_placeholder.txt" You can ignore that—it's just there to tell you that the data is in the right place!

Finally, now that all the data is in the "data" folder, we want to update the file titles within our code! This may seem very intimidating, but don't worry, it is just a couple copy-pastes.

There are two options: you can (1) change the data file names in your "data" folder — this is the easiest way to go about it, or (2) you can update the file "paths" (where it is located in your computer) in our code!

### For (1): Change file name in your folder

The exact file names we can use are:

- mdl\_h5pactivity\_attempts.csv
- mdl\_student.csv
- mdl\_learner.csv
- learnerRequestData.csv

### For (2): Update file paths in code

To update the file paths, you can do so in a text editor. Specific instructions are found in this section of the document [here](#).

## 7) Generate dashboard

To go from outer folders to other folders in, we will use the following command:

```
cd [NEXT DIRECTORY]
```



**DEVI  
SANSTHAN**  
Dignity Education  
Vision International  
*Leave no one behind*

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sandy> cd onedrive
PS C:\Users\sandy\onedrive> cd Desktop
PS C:\Users\sandy\onedrive\Desktop> cd CS61A
PS C:\Users\sandy\onedrive\Desktop\CS61A> cd Projects
PS C:\Users\sandy\onedrive\Desktop\CS61A\Projects> cd cats
PS C:\Users\sandy\onedrive\Desktop\CS61A\Projects\cats> cd cats.py
```

Now, you will be in the innermost folder (in this example, it is called “cats,” but you would type **demographicsDashboard.py** and **performanceDashboard.py**), and you will now type the following command to generate your dashboard, which should open in your browser:

**python3 cats.py**

## Updating file names in code:

### 1) Install a text editor

The **Python interpreter** that you just installed allows you to *run* Python code. You will also need a **text editor**, where you will *write* Python code.

[Visual Studio Code \(VS Code\)](#) is the most popular choice among the staff for this course for writing Python. Some other editors that are used among staff are listed below as well.

**If you're using Windows** and followed our Python setup process, VS Code will work best for you (since it has WSL support). After installing VS Code, install the [Remote Development extension pack](#). Then, you can use the instructions in [this section](#) of the VS Code docs to open WSL folders in VS Code.

**We highly recommend using VS Code.**

Another nice feature of VS Code is that it features an "embedded terminal". So, when running terminal commands for this class, you can manage everything in VS Code rather than navigating back and forth between VS Code and a separate terminal application. You can open an embedded terminal by going to [Terminal > New Terminal](#) in VS Code's navigation bar.

**Warning:** Please do not use word processors such as Microsoft Word to edit programs, as it'll add/remove extra characters.

For your reference, here are some guides on using popular text editors:

- [Visual Studio Code](#): A full-featured desktop editor with many extensions available to support different languages.
- [Atom](#): A more lightweight desktop editor.



**DEVI  
SANSTHAN**  
Dignity Education  
Vision International  
*Leave no one behind*

## 2) Change path name

For example, for the demographicsDashboard.py, we need 4 file “paths” to be updated. Make sure the name of the file after “data/” is exactly matching with that of the corresponding file in your “data” folder.

The four categories of data this dashboard takes in are: h5p\_activity\_attempts, student, learner, and learnerRequestData:

```
h5p_activity_path = "data/mdl_h5pactivity_attempts.csv"
student_data_path = "data/mdl_student.csv"
learner_data_path = "data/mdl_learner.csv"
request_data_path = "data/learnerRequestData.csv"
```

In our case here, our student dataset is saved as “mdl\_student.csv” in the directory ‘/Users/alexoon/Desktop/DSS-x-TWB-Devi-S/data/mdl\_student.csv’ — so this is what we set as student\_data\_path in performanceDashboard.py (can be shortened to data/mdl\_student.csv as the code is within the DSS-x-TWB-Devi-S folder).

```
#-----
# change file paths and browser here

h5p_activity_path = "data/mdl_h5pactivity_attempts.csv"
student_data_path = "data/mdl_student.csv"
learner_data_path = "data/mdl_learner.csv"
request_data_path = "data/learnerRequestData.csv"
browser = "chrome"

#-----
```