Following up the workflow on the Youtube channel below:

<https://www.youtube.com/watch?v=S_F_c9e2bz4&list=PLZoTAELRMXVPS-dOaVbAux22vzqdgoGhG>

Create Main folder. Link the Main Folder to a Git and Github.

Main Folder Name: Cell Voltage

* Files:
  + .gitignore
  + README.md
  + requirements.txt
  + app.py
  + setup.py
* Folders:
  + .venv
  + src
    - \_\_init\_\_.py
  + data
    - Files:
      * Cell\_Voltage.xlsx
  + Notebooks
    - Files:
      * Cell\_Voltage\_EDA

# Create a working folder

Filename: Cell Voltage (will do the same for hydrogen production).

# Create a virtual environment.

(powershell terminal of VSCode) >> python -m venv .venv

## Activate the venv

>> .venv\Scripts\Activate

# Create a requirements.txt file

## Add the important packages

* numpy
* pandas
* scikit-learn
* matplotlib
* seaborn
* jupyter
* etc

## Run the following command after every package addition

>>pip install -r requirements.txt

## Use the following command to check if dependencies are installed

>>pip list

# Now add folder to GIT local repository

>>git init

## Create a .gitignore file

### Add files to ignore

* .venv/
* \_\_pycache\_\_/
* .ipynb\_checkpoints/

## Add files to be tracked

>>git add .

## Save changes to the local git repository

>> git commit “message”

# Create an online Github repository

* Check README.md
* Check .gitignore

## Get repository URL (to Clone)

* <>Code => HTTPS (copy url)

## Link the repositories

>>git remote add origin HTTPS

## Check which Github repo is connected

>>git remote -v

# Version Control

## Add files to track

>> git add .

## Check files tracked

>> git status

## Commit changes on local repository (Git)

>> git commit -m “Message”

## Commit changes on remote repository (Github)

>> git push -u origin master (or main)

# Setup project structure

## Add these files

### App.py

## Add these folders

### Data

### Notebooks

### SRC

# EDA

## Loading Data to Python

* Import necessary modules i.e. pandas, numpy, matplotlib, seaborn
* Since the data is modular, need to combine all sheets into one. Firstly copy path of the excel workbook and use the following code

>> file\_path = r"C:\Users\jmasa\OneDrive - University of Johannesburg\Thesis\ML Projects\Python\Cell Voltage\data\Cell\_Voltage\_Data\_6Mar.xlsx"

* Follow this with uploading the specific sheet.

>> deGroot = pd.read\_excel(file\_path, sheet\_name= "deGroot et al (2022)")

>> Emam1 = pd.read\_excel(file\_path, sheet\_name= "Emam et al (2024)fig5")

* Repeat for the remaining sheets. (Emam2, Emam3).
* Now use the following code to combine the 4 datasets.

>> df = pd.concat([deGroot,Emam1,Emam2,Emam3 ], ignore\_index=True)

* Index = True just means the resulting DataFrame gets a new integer index starting from 0, instead of keeping the original indices from the individual DataFrames.
* Finally save the new combined dataset using the following code.

>> df.to\_excel('Cell\_Voltage\_Combined\_9Mar.xlsx', index=False)

* Index=False ensures that the DataFrame is saved to an Excel file without writing the index column.