Coding Tips and How-tos

# CMD Commands

* Add 'C:\Users\tomos\AppData\Roaming\Python\Python39\Scripts' to directory PATH
* Start virtual environment type in CMD C:\Users\tomos\source\repos\Platform\_v1\_0\Platform\_v1\_0\env\Scripts\activate.bat'
* Then to install Django import export in CMD type 'py -m pip install django-import-export'
* To start Python shell in CMD first navigate to where the manage.py is located then type 'py manage.py shell'
* To access model data type within the Python shell 'from app.models import Bids', where Bids is just an example of a model name
* The data can then be queried / manipulated using the Django ORM
* Country.objects.filter(name\_\_exact="Peru").values('name')
* City.objects.filter(id=1).values('country')
* City.objects.filter(id=1).values('') to show all field values
* City.objects.filter(id=1).values('country\_id') also works as country\_id is the ForeignKey added by Django
* Filters for a specific database entry: Bids.objects.filter(id=4940).values()
* Save a new value to the database: Bids(bid\_value=2005).save()
* Item.objects.values('menu\_\_name\_menu') or Item.objects.values('menu\_\_name\_menu').values()
* ForeignKey has to be a number and cannot be text
* bids=Bids.objects.values()[0:10]
* sortedbids=list(bids.values\_list('bid\_value', flat=True))
* sortedbids.sort()

# PostgreSQL Databases

* Set up process followed was taken from:

<https://medium.com/django-unleashed/complete-tutorial-set-up-postgresql-database-with-django-application-d9e789ffa384>

* The creation of Django project and Django app can be skipped, as these already exist
* Just the simple CMD from Windows OS was used (rather than accessing PowerShell or similar via Visual Studio) and works
* Venv WAS used when setting up PostgreSQL and works; uses the ‘activate’ file saved at [C:\Users\tomos\OneDrive\Platform\GitHub\Platform\_v1\_0\env\Scripts](file:///C:\Users\tomos\OneDrive\Platform\GitHub\Platform_v1_0\env\Scripts)
* Database creation was performed in the main app folder i.e. [C:\Users\tomos\OneDrive\Platform\GitHub\Platform\_v1\_0](file:///C:\Users\tomos\OneDrive\Platform\GitHub\Platform_v1_0)
* User and superuser can and usually do have different names and passwords
* When creating a user within Postgres, the additional piece of code needs to be included:

ALTER DATABASE platformdb OWNER TO myuser;

* + This extra piece of code was taken from <https://stackoverflow.com/questions/74110708/postgres-15-permission-denied-for-schema-public/74111630#comment130849690_74110708>

# MySQL Databases

* Open non-env version PowerShell from within Visual Studio
* Navigate to main folder location i.e. C:\Users\tomos\OneDrive\Platform\GitHub\Platform\_v1\_0
* Log into MySQL using command ‘mysql -u root -p’ and then enter password ‘Lampost\_220783’
* To see existing databases enter ‘SHOW DATABASES;’
* To create a new database, enter command ‘CREATE DATABASE database\_name;’
* Exit MySQL using the command ‘exit’
* Amend the database name details in the settings.py file
* To update and confirm the new database details, ensure still in the main folder i.e. ‘…GitHub\Platform\_v1\_0’ and first enter the command:
  + ‘py manage.py makemigrations’ and then command
  + ‘py manage.py migrate’
* Create a super user by entering the command ‘py manage.py createsuperuser’
* Start running the code by entering the command ‘py manage.py runserver’ and enter within a browser the website address referenced within PowerShell i.e. ‘http://127.0.0.1:8000/’
* To upload data from the website to a database i.e. not using the Admin page, then forms.py needs to be amended, however, if data is only downloaded then forms.py is not needed

# Python

* Python on my machine I think exists in a virtual environment, meaning it must be started via PowerShell in Microsoft Visual Studio
* Matplotlib initial steps have been followed using the below <https://www.w3schools.com/python/matplotlib_getting_started.asp>
* PIP is accessed directly via the CMD and not within Python
* Pip is already installed BUT can only be accessed within venv
* Matplotlib must be imported each time Python is started
* Numpy is the fundamental package for scientific computing with Python and should be used to create and handle matrices and arrays

# Heroku / Git

* List git branches, all, remote and details
  + Git branch -a
  + Git branch -r
  + Git branch -v
* List the external / remote details
  + git remote -v
* List Heroku addons and apps
  + heroku addons –all
  + heroku apps --all
* When pushing to Heroku:
  + git push Heroku main:main - does not work
  + git push heroku HEAD:master – does not work