**Creating a Virtual Environment**

1-) directory Name$ python3 -m venv ll\_env

**Activating the Virtual Environment**

1-) source ll\_env/bin/activate

**Stop using the Virtual Environment**

1-) deactivate

2-) close the terminal

**Installing Django in the Virtual Environment**

1-) pip install Django

**Creating a Project in Django Virtual Environment**

1-) django-admin startproject learning\_\_log . **(Don’t forget the dot)**

2-) **ls** [learning\_log, ll\_env, manage.py]

3-) **ls learning\_log** [\_\_init\_\_.py, settings.py, urls.py, wsgi.py]

**Creating the Database**

1-) python/python3 manage.py migrate

**Verifying the Project has been setup properly**

1-) python/python3 manage.py runserver (8000, 8001)

2-) control-C (to stop the server)

**Starting and App**

1-) python/python3 manage.py startapp learning\_logs

**Continues …**

**1-) Defining Models (Classes in Django)**

1-) That is done in the models.py file of the app directory just created.

**Activating Models**

1-) That is done in the settings.py file of main project directory.

**Adding our App to the Database**

1-) python/python3 manage.py makemigrations learning\_logs

**Applying our Topic to the Database**

1-) python/python3 manage.py migrate

**Whenever we want to modify the Database**

1-) Defining Models

2-) Adding our App to the Database

3-) Applying our Topic/Change to the Database

**Setting Up a Superuser**

1-) python/python3 manage.py createsuperuser

**Registering a Model with the Admin Site**

1-) That is done in the admin.py file of app project main directory.

**2-) Defining the Entry Model (many-to-one-relationship)**

1-) That is done in the models.py file of app project main directory.

**Migrating the Entry Model (Each time you modify Models.py)**

1-) python/python3 manage.py makemigrations learning\_logs

2-) python/python3 manage.py migrate

3-) That is done in the admin.py file of app project main directory.

**Continues …**

**The Django Shell**

1-) python/python3 manage.py shell

Note: iterating over a queryset it’s like looping over a list.

Example:

from learning\_logs.models import Topic

topics = Topic.objects.all()

for topic in topics:

print(topic.id, topic)

Note: Examine any attribute object

Example:

t = Topic.ojects.get(id=1)

t.text

t.date\_added

Note: Look all entries related to a certain topic.

Example:

t.entry\_set.all()

2-) control-D (to exit a shell session)

**Making Pages (The Learning Log Home Page)**

1-) Map a URL (That’s done in the urls.py file of the main project folder,

then in the url.py file of the app folder for every single web page)

2-) Write a View (That’s done in the views.py file of the app folder)

3-) Write a Template (That’s done by creating a templates folder inside

of the app folder and then putting all html files of the web page

inside)