Create environment

**Python -m venv <environment name>**

Activatre the environment

**ll\_env/Scripts/activate**

To deactivate the environment

**deactivate**

To install Django in virtual environment

**pip install Django**

To create the Django project you are currently in

**django-admin startproject <Project name> .**

command to create a database

**python manage.py migrate**

To view the project we first run the server

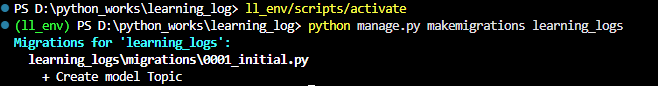
**python manage.py runserver**

Cammand to create an app that will work with project is

**python manage.py startapp <appname>**

To tell Django to modify the database so it can store information related to the model Topic

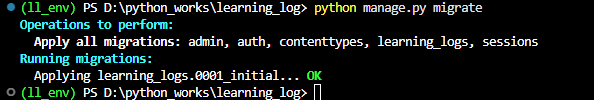
**python manage.py makemigration learning\_logs**

****

The makemigration cammand tell Django to figure out how to modify the database so it can store the data associated with any new models we’ve defined. The output here shows that Django has created a migration file called 0001\_initial.py. The migration will create a table for model Topic in the database

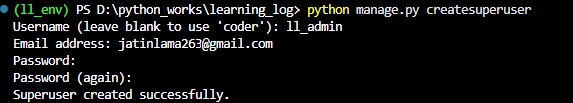
Now we will apply this migration and have Django modify the database for us

**Python manage.py migrate**

****

To create a superuser(which have all the access basically admin) in Django, enter the following and respond to the prompts

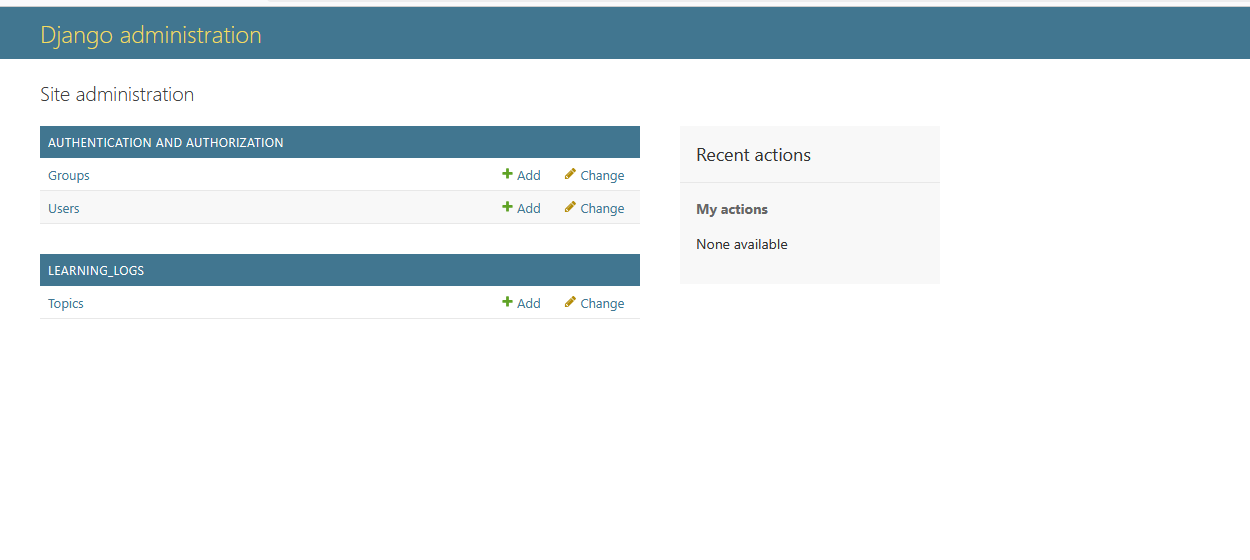
**python manage.py createsuperuser**

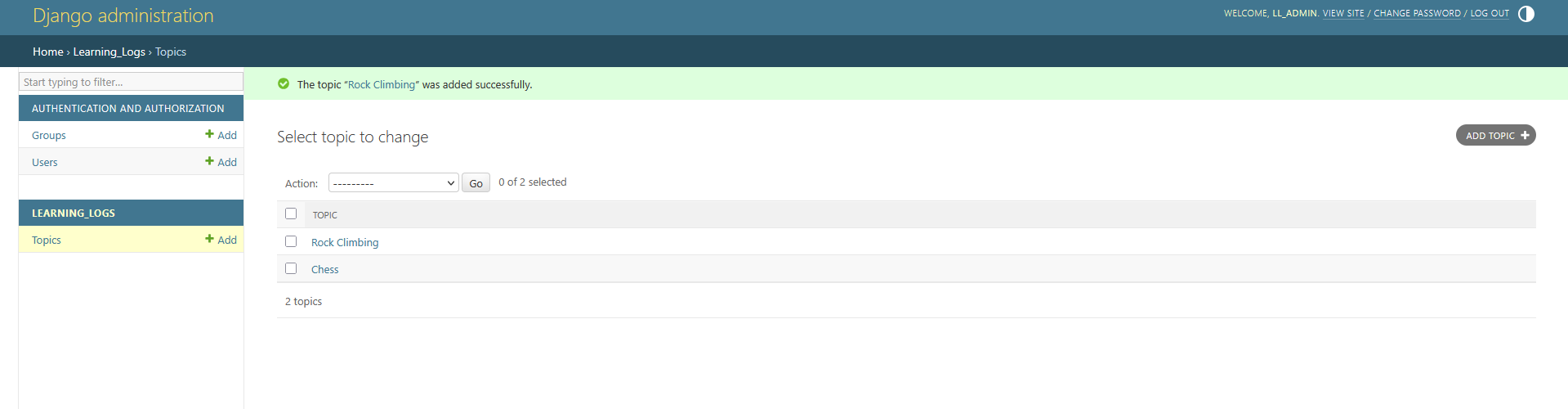


Now we need to register models in the admin.py

Reach the admin url and then enter username and password

Add topic , user and groups





Everytime you make changes in the models you need to migrate it

So that database can reflect it

We can examine data we entered by admin pragmatically through an interactive terminal session. This interactive environment is called the Django Shell, and its great environment for testing and troubleshooting your project

To start this type

**python manage.py shell**

for example

**>>> from learning\_logs.models import Topic**

**>>> Topic.objects.all()**

**>>> topics = Topic.objects.all()**

**>>> for topic in topics:**

**... print(topic.id,topic)**

**...**

**1 Chess**

**2 Rock Climbing**

get any object related to a model class

**>>> t = Topic.objects.get(id=1)**

**>>> t.text**

**'Chess'**

**>>> t.date\_added**

**datetime.datetime(2025, 4, 16, 13, 37, 0, 265631, tzinfo=datetime.timezone.utc)**

If you want to see all the entries related to the topic

**t.entry\_set.all()**

**<QuerySet [<Entry: Entry object (1)>, <Entry: Entry object (2)>]>**

**(Here I have done a mistake related to \_\_str\_\_)**