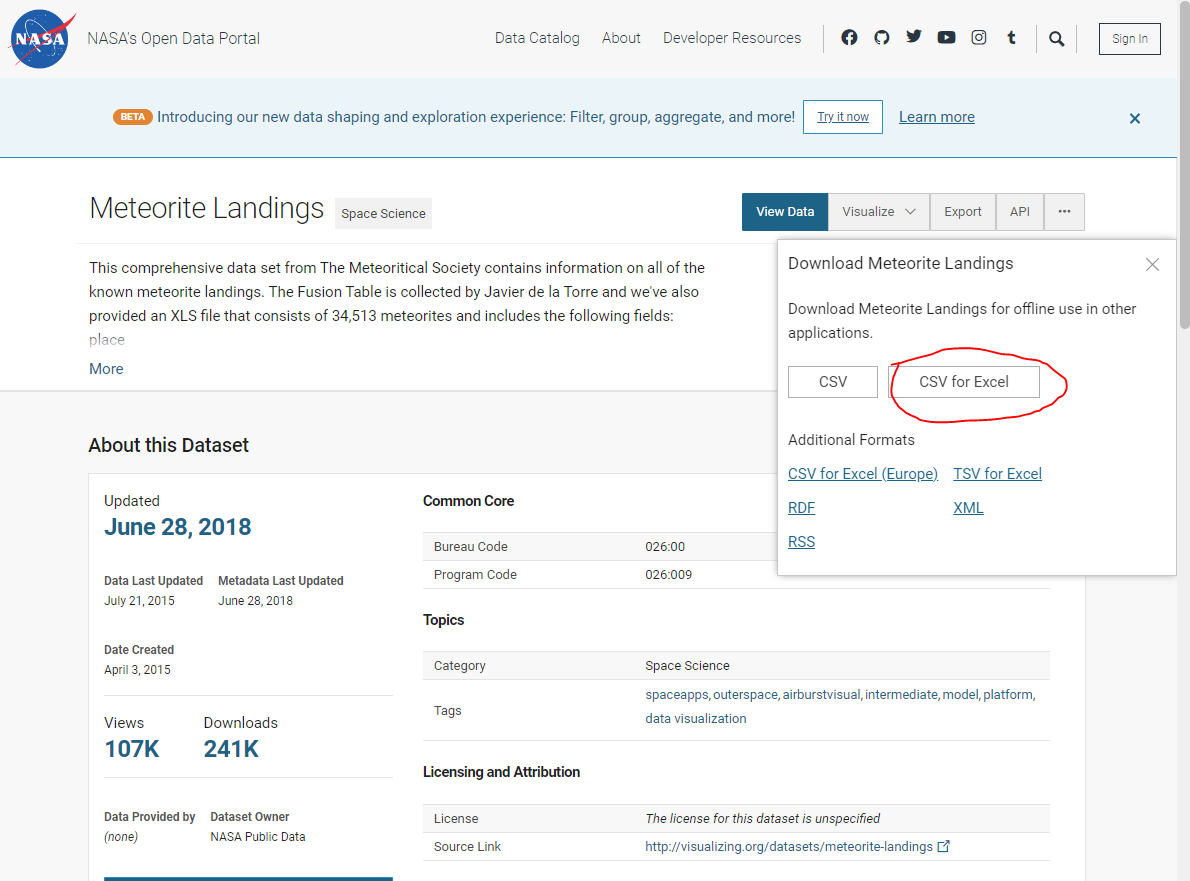
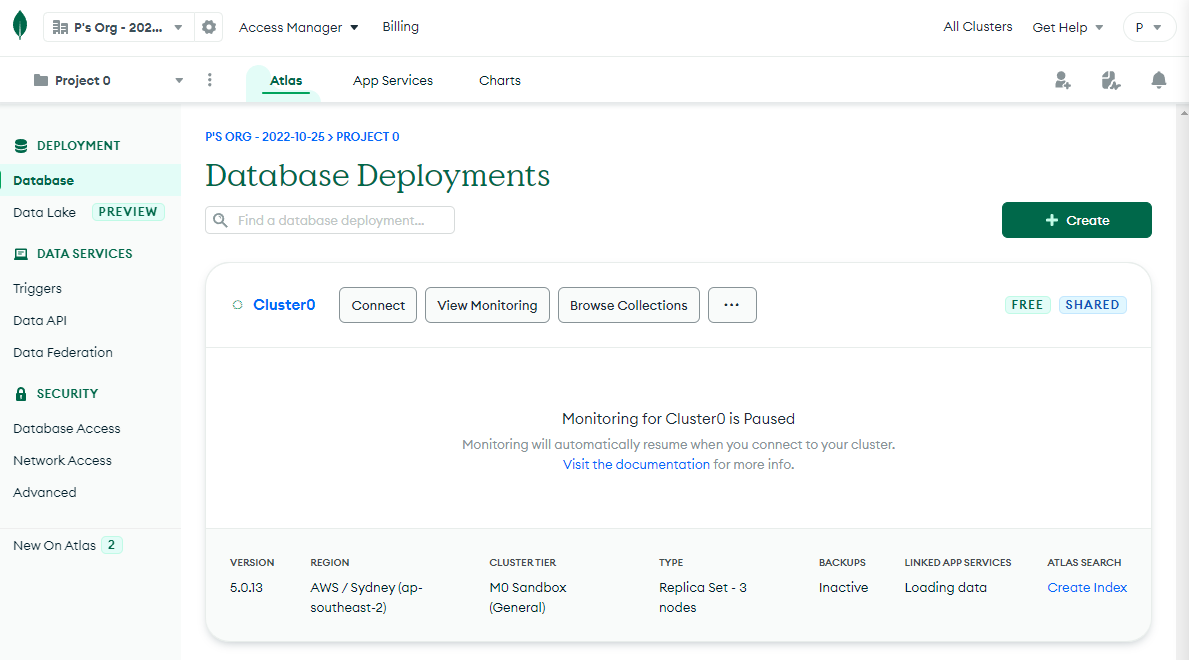
# Connecting Django to MongoDB via PyMongo Example

## 1.0 Creating the Dataset

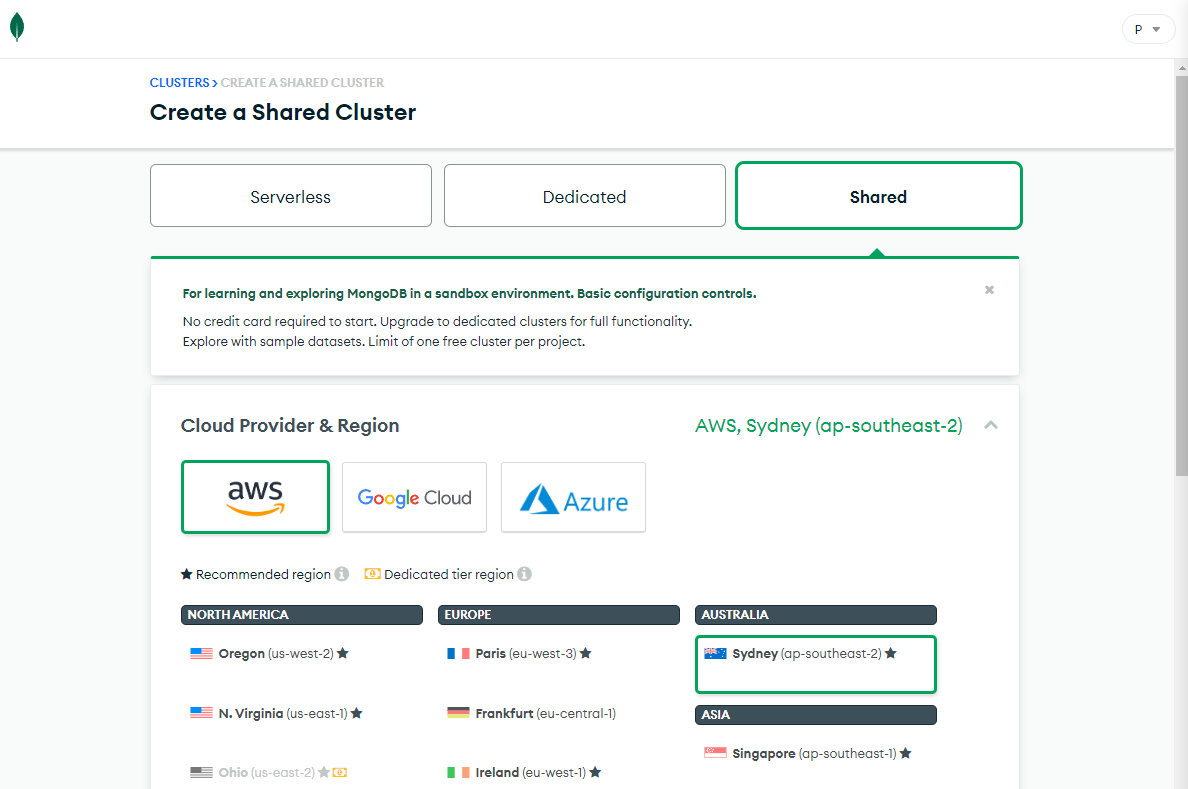
**Step 1.1:** Download the dataset <https://data.nasa.gov/Space-Science/Meteorite-Landings/gh4g-9sfh>



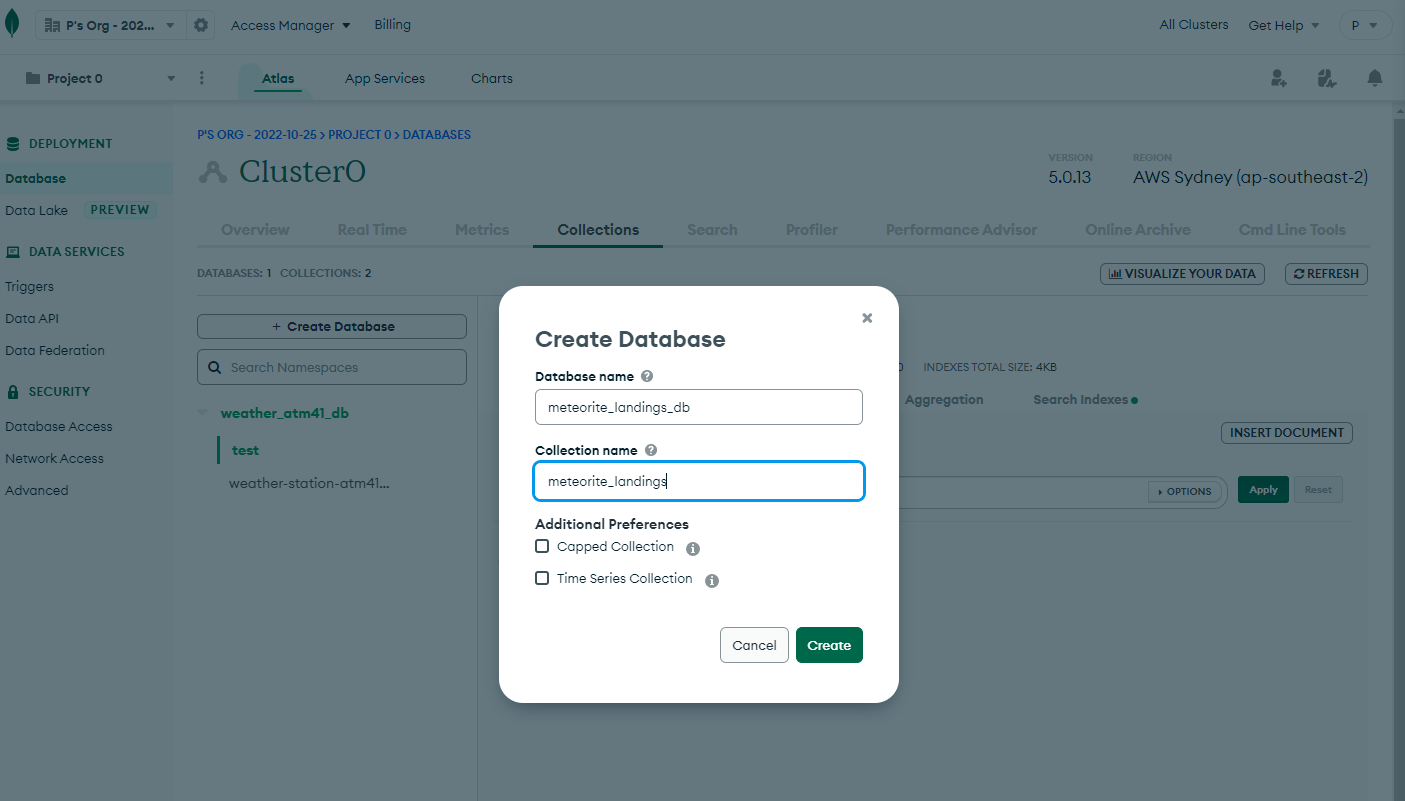
**Step 1.2:** Log into Atlas <https://www.mongodb.com/atlas/database>



**Step 1.3a:** If you do not have a cluster already, create a new Shared Cluster called “NasaData”. Follow step 3b after this with this new cluster…

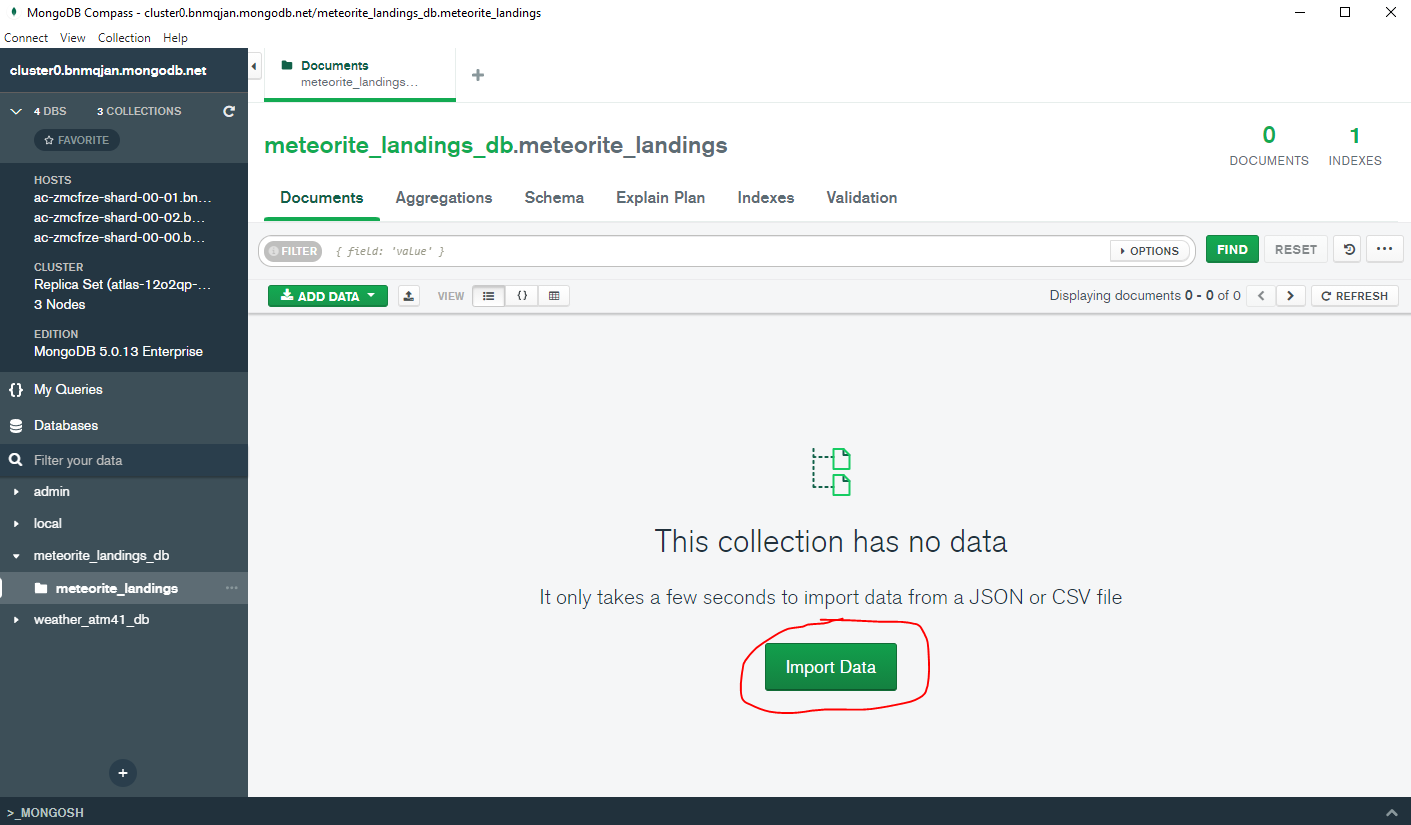


**Step 1.3b:** If you have a Shared cluster already, click on it, go to the “Collections” tab and create a new database called “meteorite\_landings\_db”. Call the table “meteorite\_landings”.



**Step 1.4:** Open up MongoDB Atlas Compass and connect to this new database/collection. Click the “Import Data” button

Download compass from here if you do not already have it: [MongoDB Compass Download | MongoDB](https://www.mongodb.com/try/download/compass)

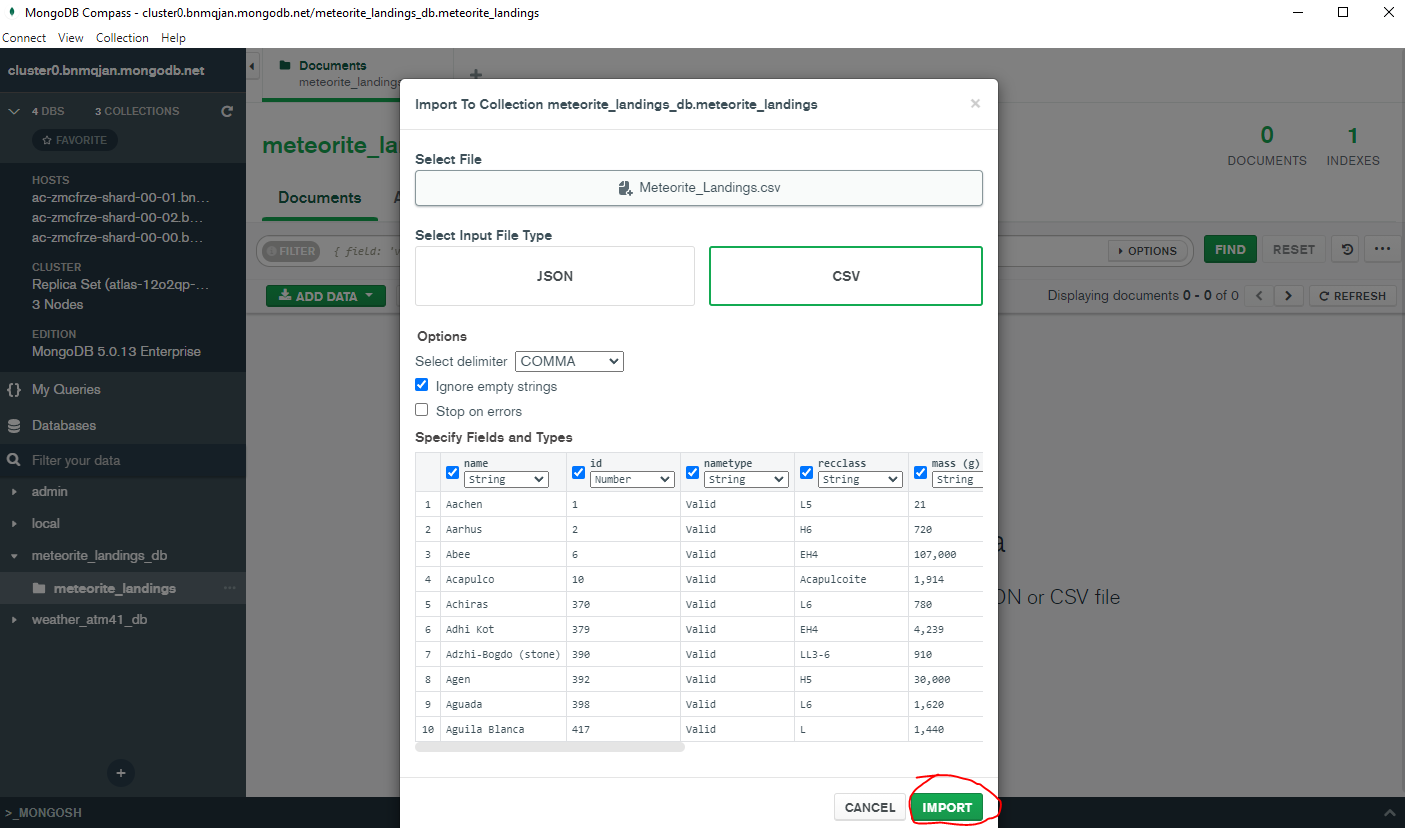


**Step 1.5:** When importing, select the “Meteorite\_Landings.csv” file downloaded earlier, select the input file type as CSV, leave the delimiter set to COMMA, and set the following fields to these data

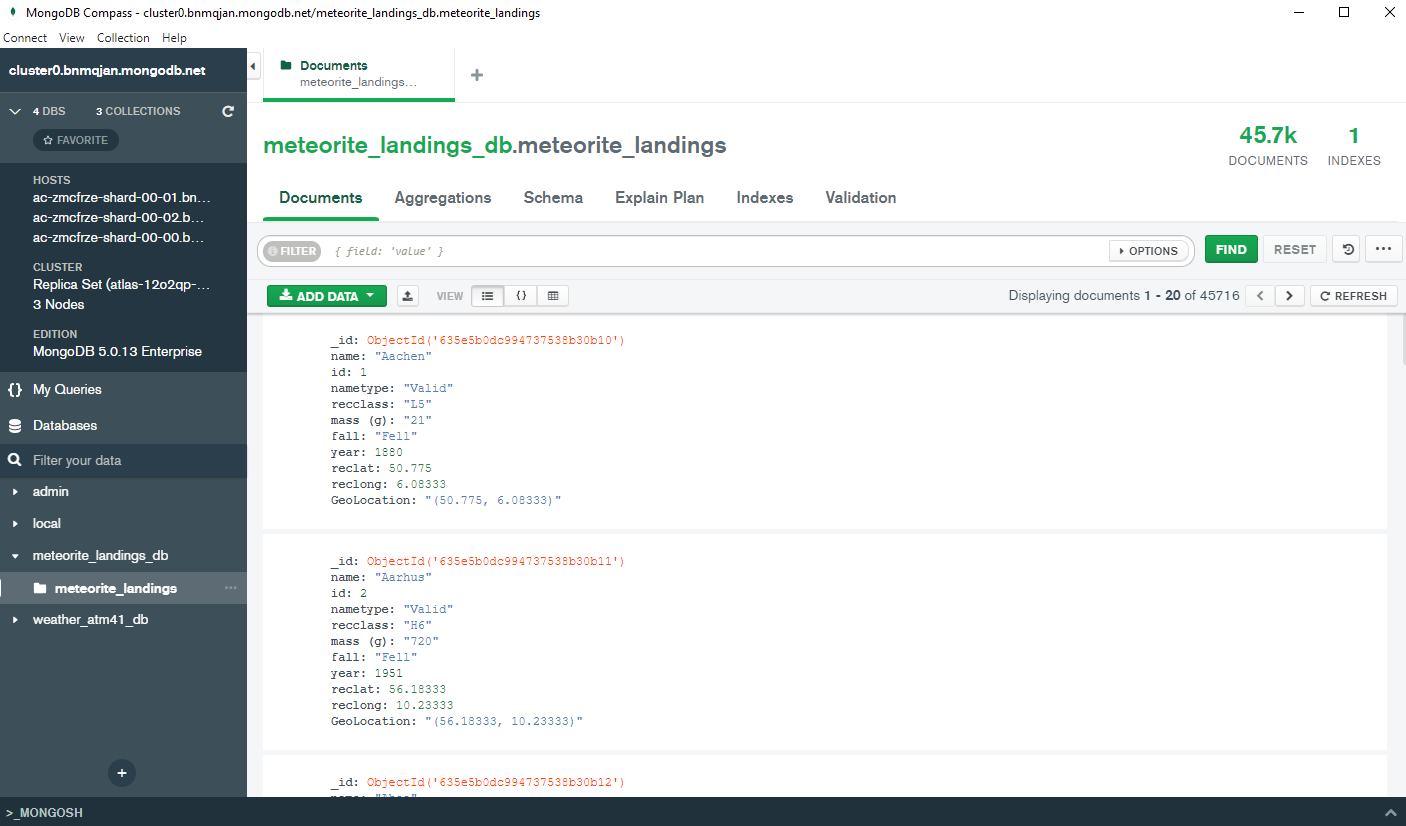
types:

* name: String
* id: Number
* nametype: String
* recclass: String
* mass (g): String
* fall: String
* reclat: Double
* reclong: Double
* GeoLocation: String

After this, click the “Import” button.

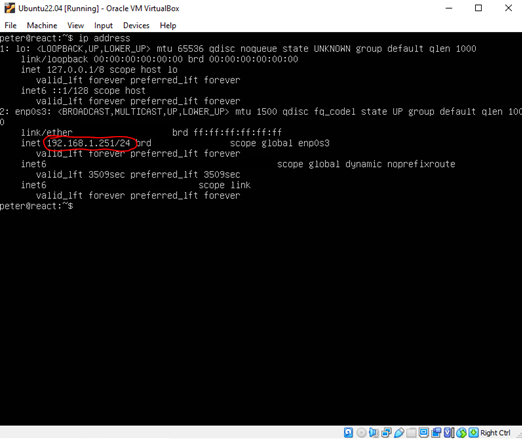


**Step 1.6:** Confirm that the data is all present.

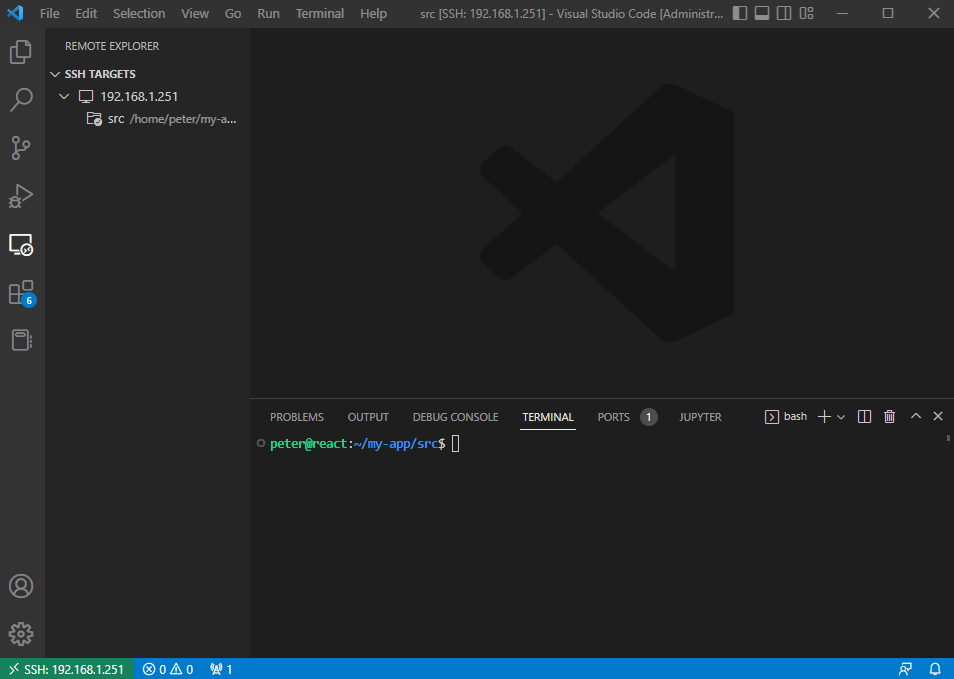


## 2.0 Creating the Django App

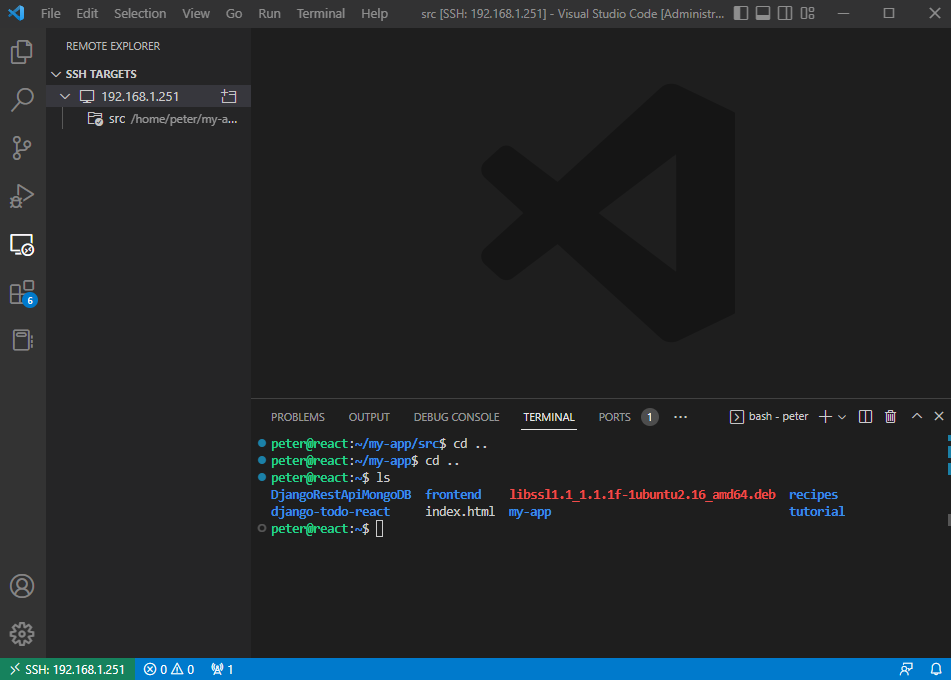
**Step 2.1:** Open up Virtual Box and spin up the Linux Virtual Machine, get the IP Address (if needed) using the command “ip address”



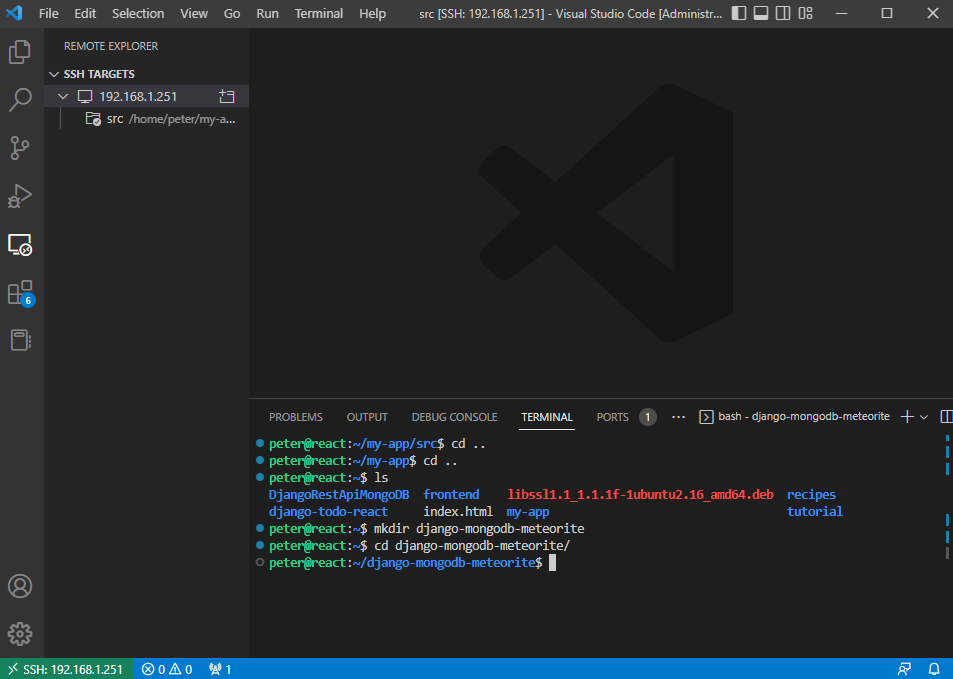
**Step 2.2:** Connect to the server using Visual Studio Code and the remote explorer extension (make sure that you are running this program as an administrator). Open up the Terminal View.



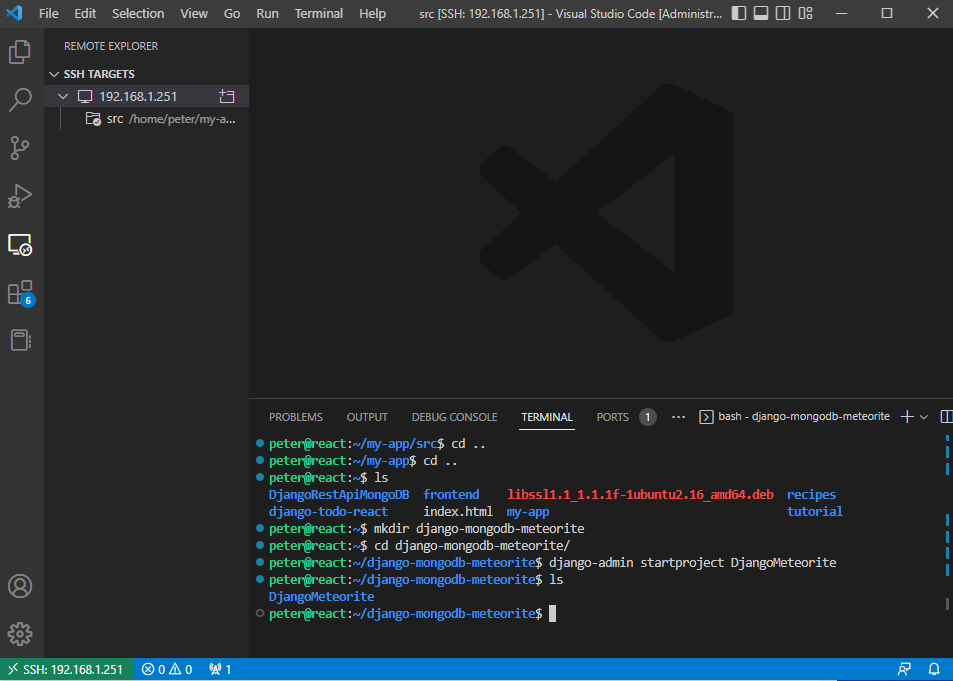
**Step 2.3:** If not done so already, navigate back to the main folder of the linux server with all the projects.



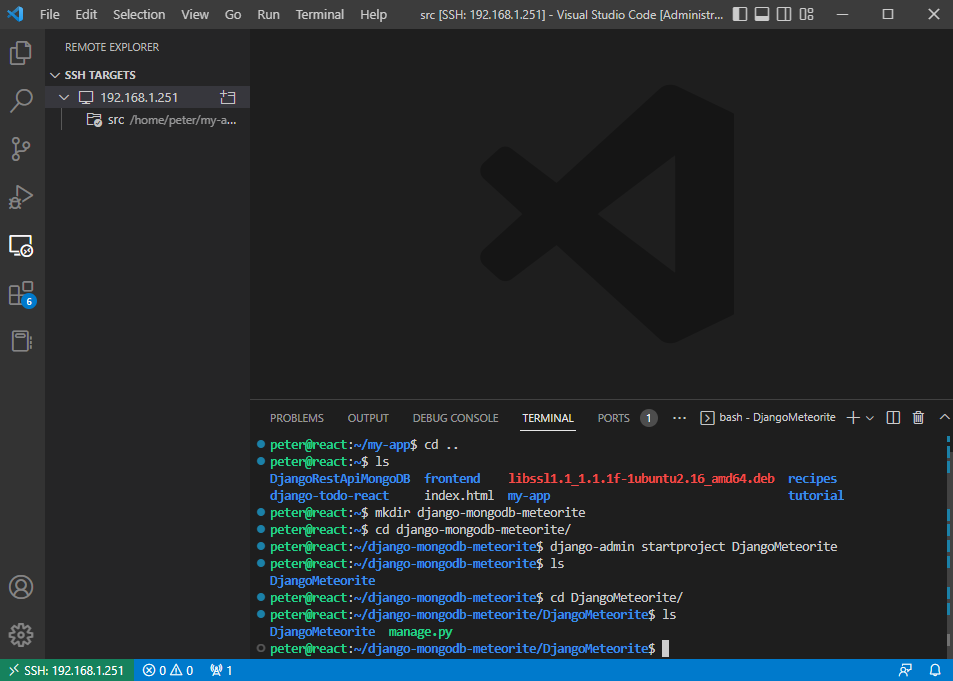
**Step 2.4:** Make a new directory with the command “mkdir django-mongodb-meteorite” then change into that directory with “cd django-mongodb-meteorite”:



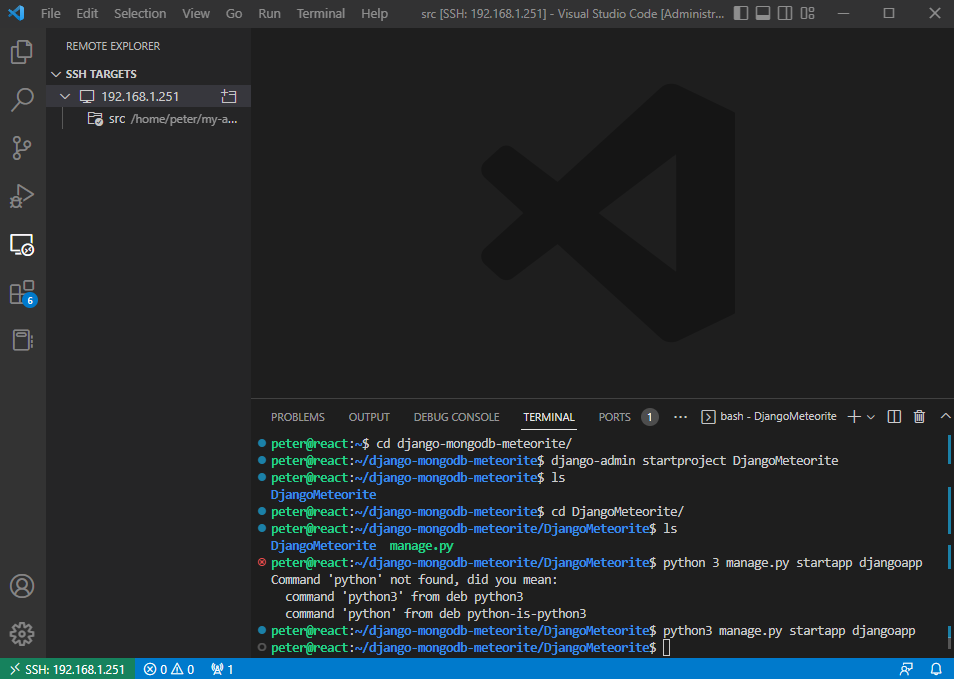
**Step 2.5:** Start the new django project by using the command ”django-admin startproject DjangoMeteorite” (with “DjangoMeteorite” being the name of the project).



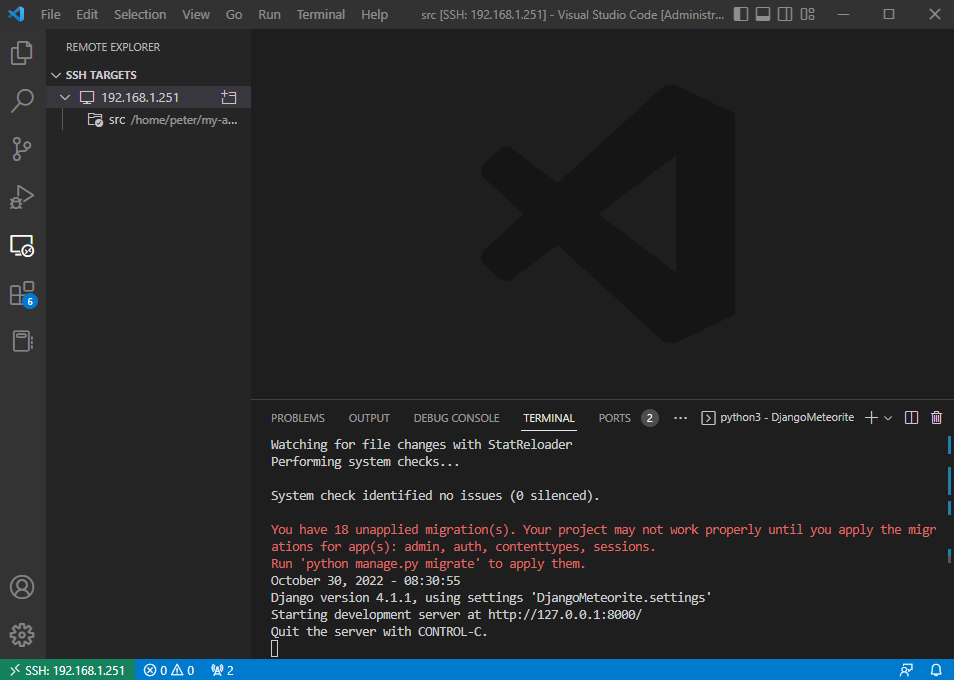
**Step 2.6:** Change directories into this new directory created with the command “cd DjangoMeteorite”.



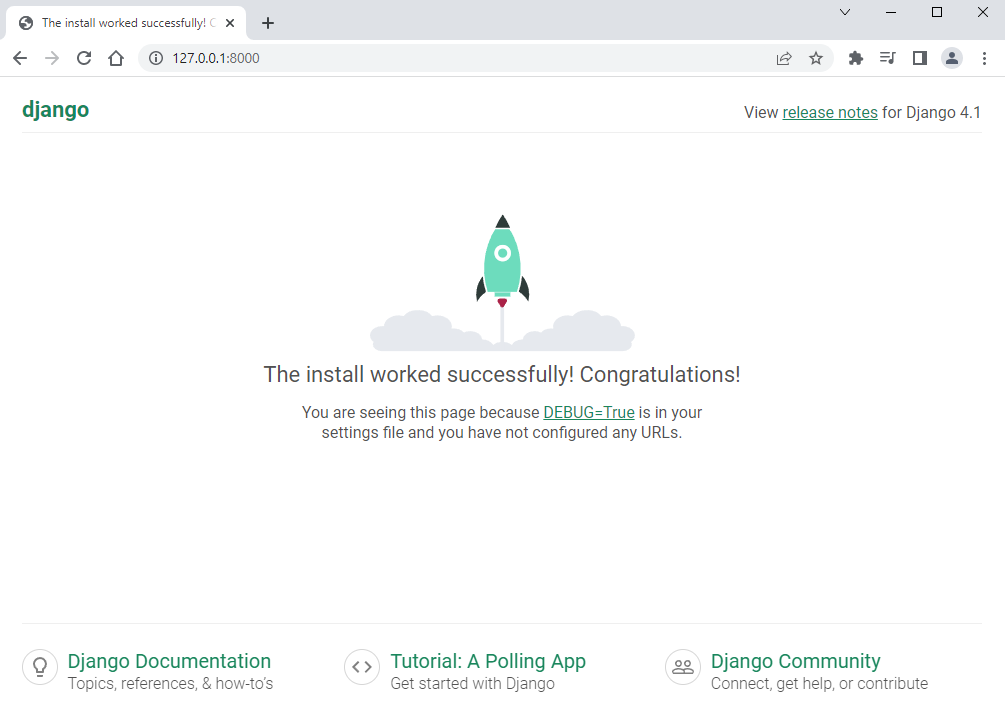
**Step 2.7:** Create the Django application while in this folder by using the command “python3 manage.py startapp djangoapp”



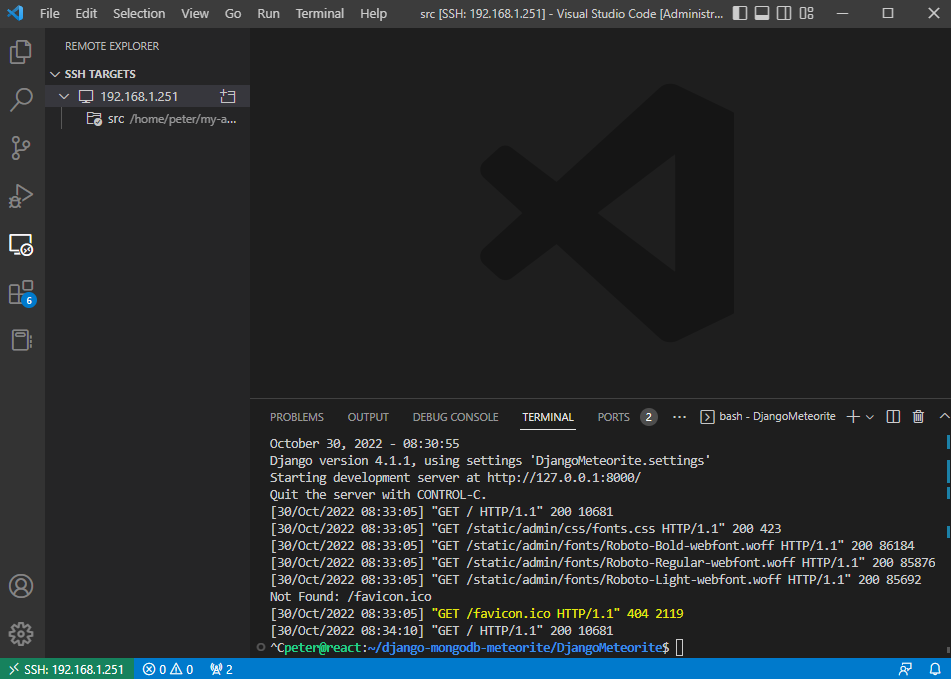
**Step 2.8:** Run the server by using the following command “python3 manage.py runserver”



**Step 2.9:** Go to “http://127.0.0.1:8000/” (home on port 8000) to confirm that you are running django.

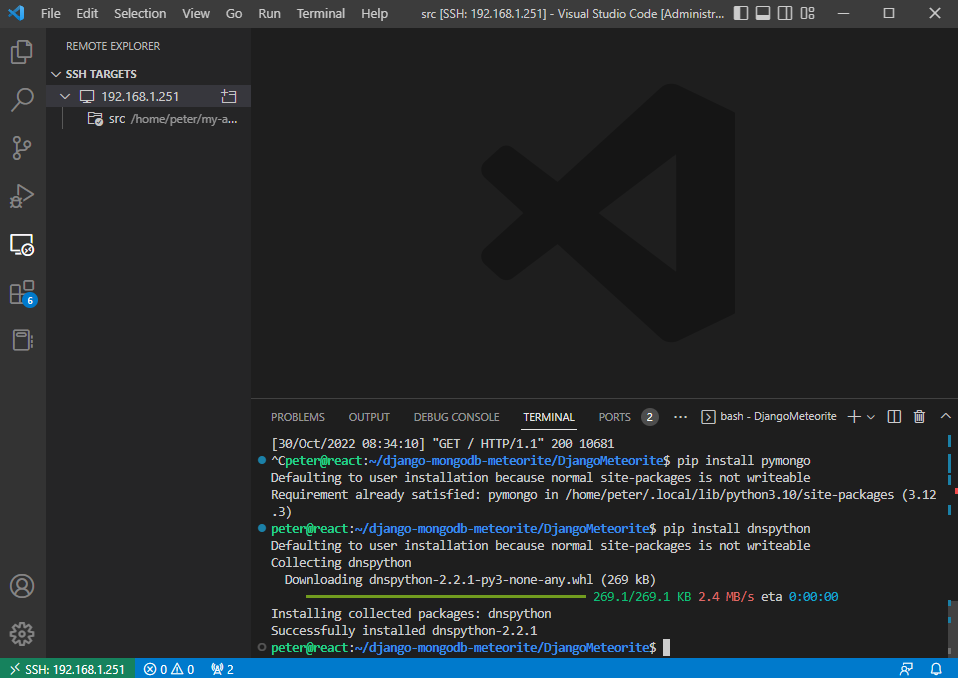


**Step 2.10:** After this, kill the server with control + c when on the terminal in Visual Studio Code. This should also prevent you from accessing the congratulations screen on the browser.

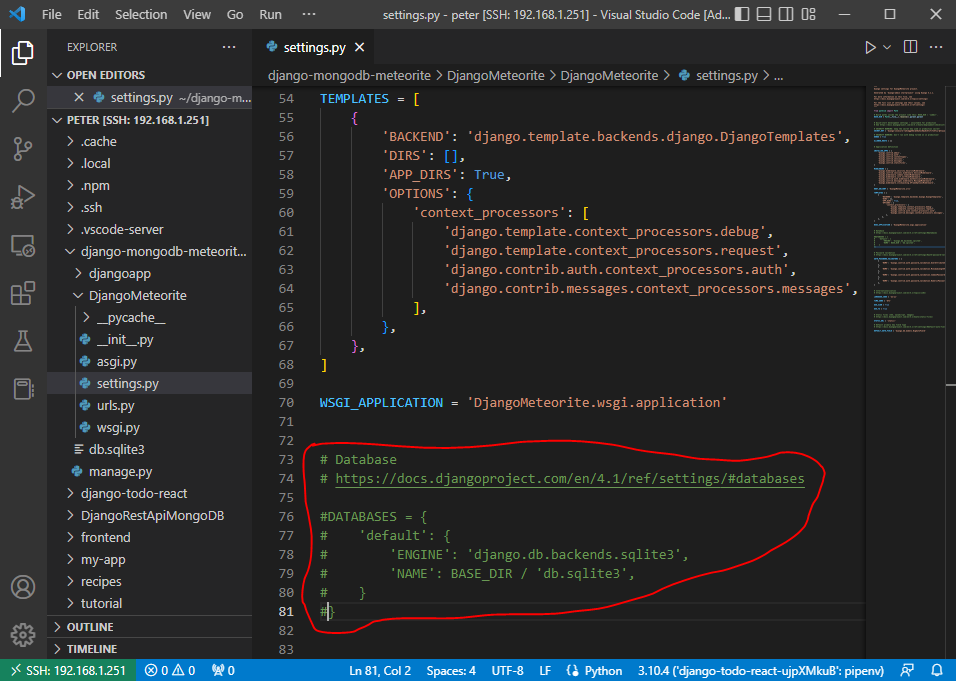


## 3.0 Setting up PyMongo

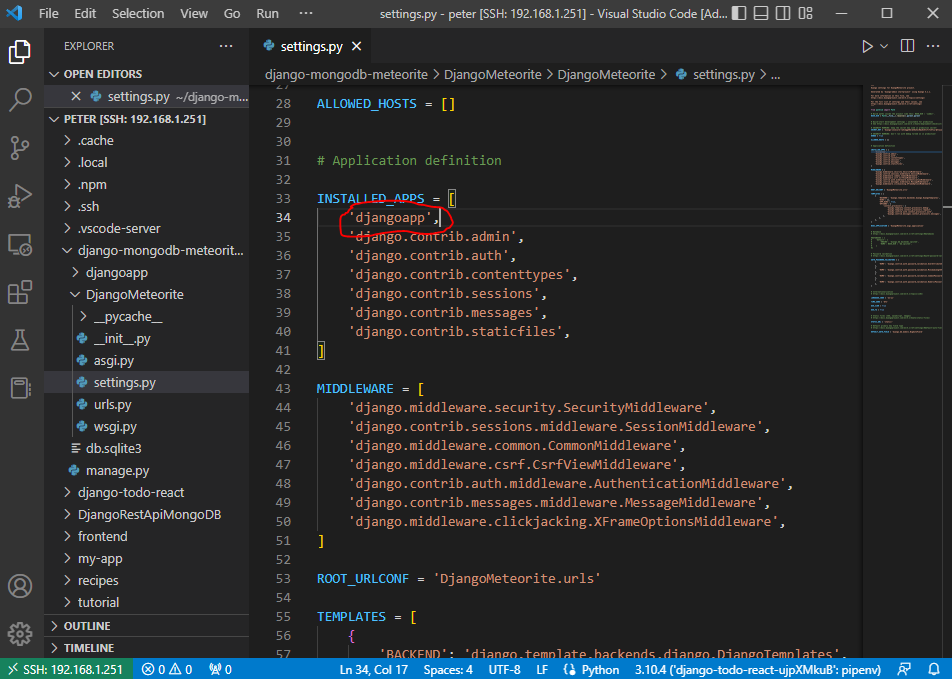
**Step 3.1:** On the terminal in Visual Studio Code connected to the Linux server, install both PyMongo and Dnspython with the “pip install pymongo” and “pip install dnspython” commands.



**Step 3.2:** Get into the “DjangoMeteorite” directory via the Explorer menu and open “settings.py”. Scroll down to the “DATABASES section and comment out all of it.

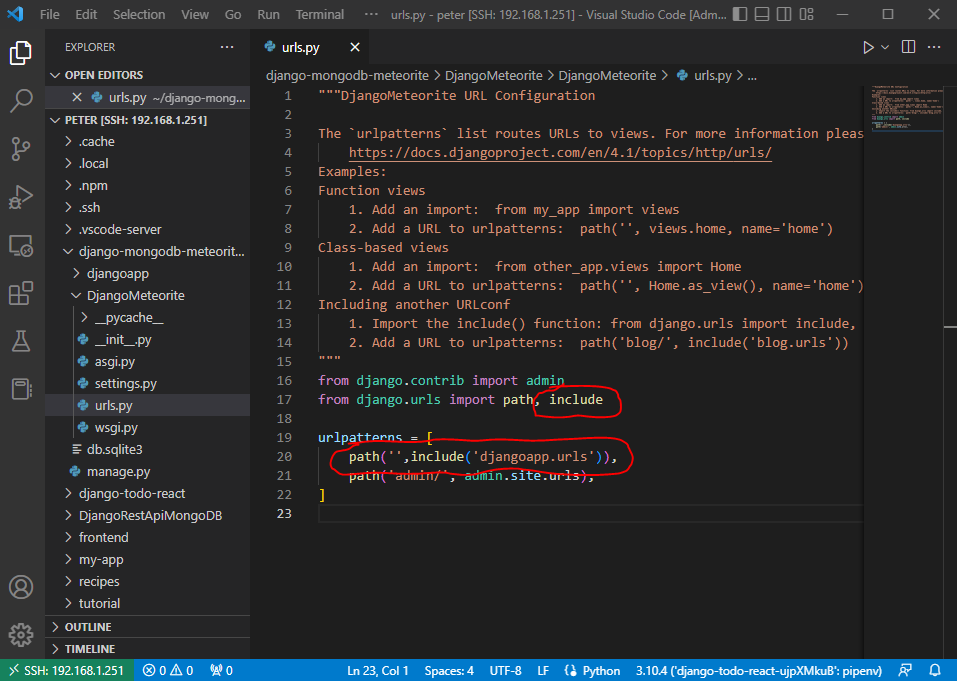


**Step 3.3:** Find the “INSTALLED\_APPS” and add “ ‘djangoapp’, ” to it.

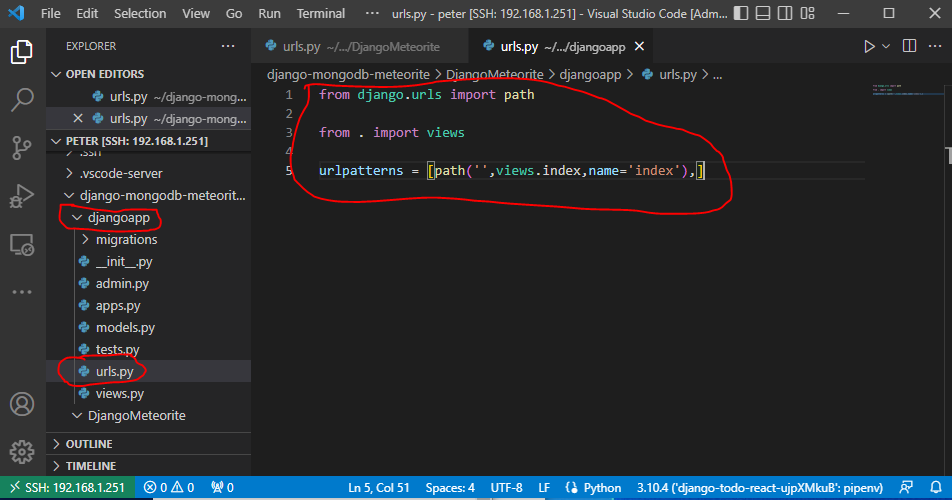


## 4.0 Configuring PyMongo

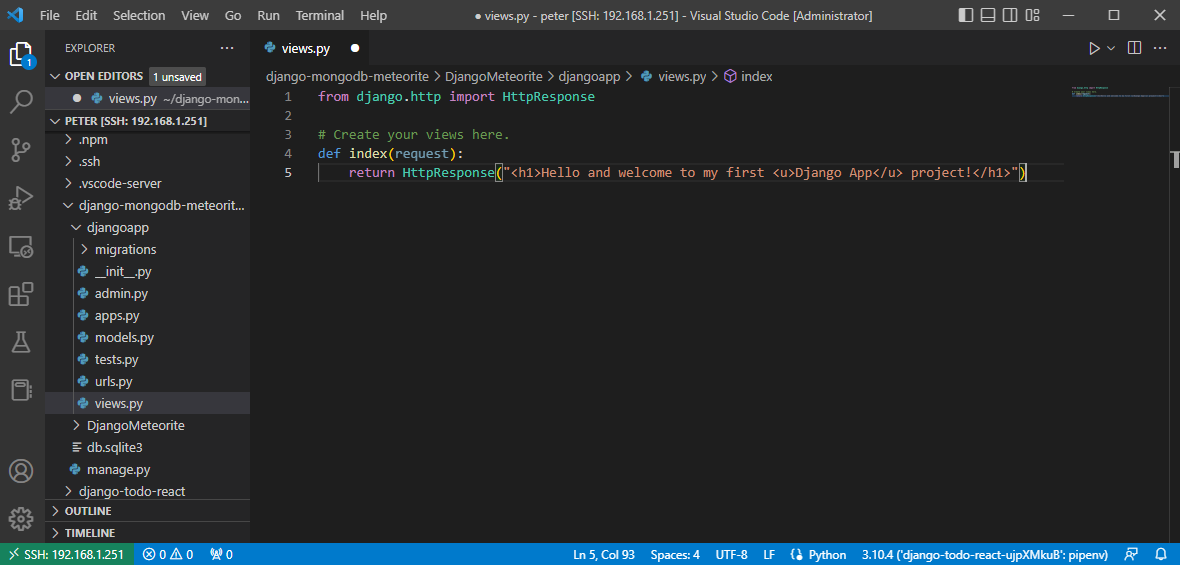
**Step 4.1:** Close “settings.py” and instead open up “urls.py”. At the end of the “from django.urls import path”, add at the end of this “, include”. After this, add a new “urlpattern” being “path(‘’,include(‘djangoapp.urls’)),”. Save this file, then close it.



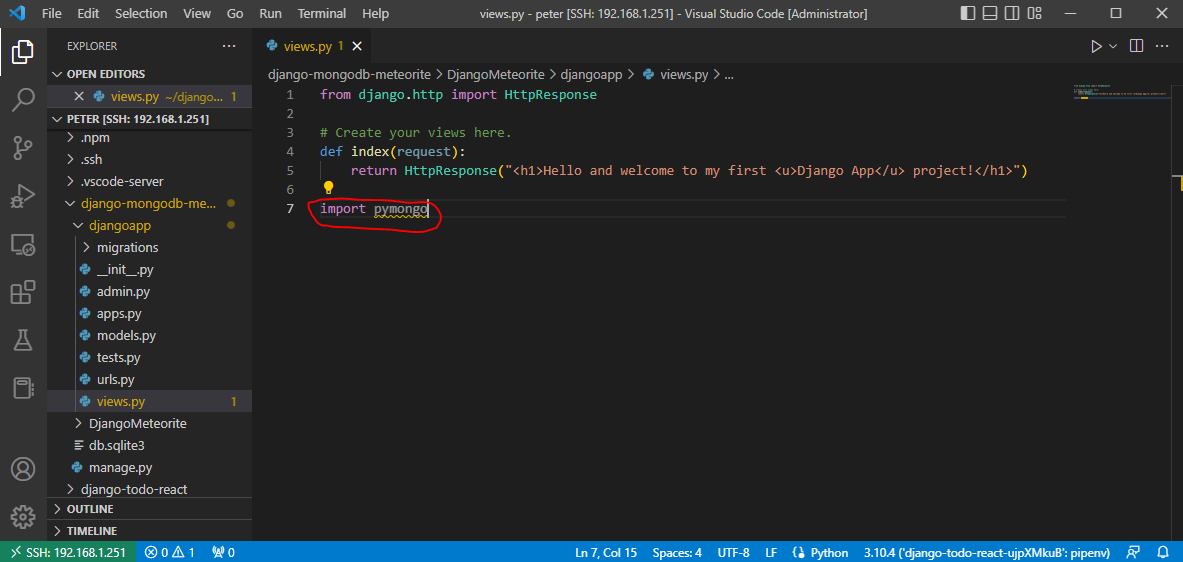
**Step 4.2:** Go to the “djangoapp” directory (as opposed to the DjangoMeteorite” directory we were in before) and create a new file called “urls.py”. Inside of this file, copy the code you see below:



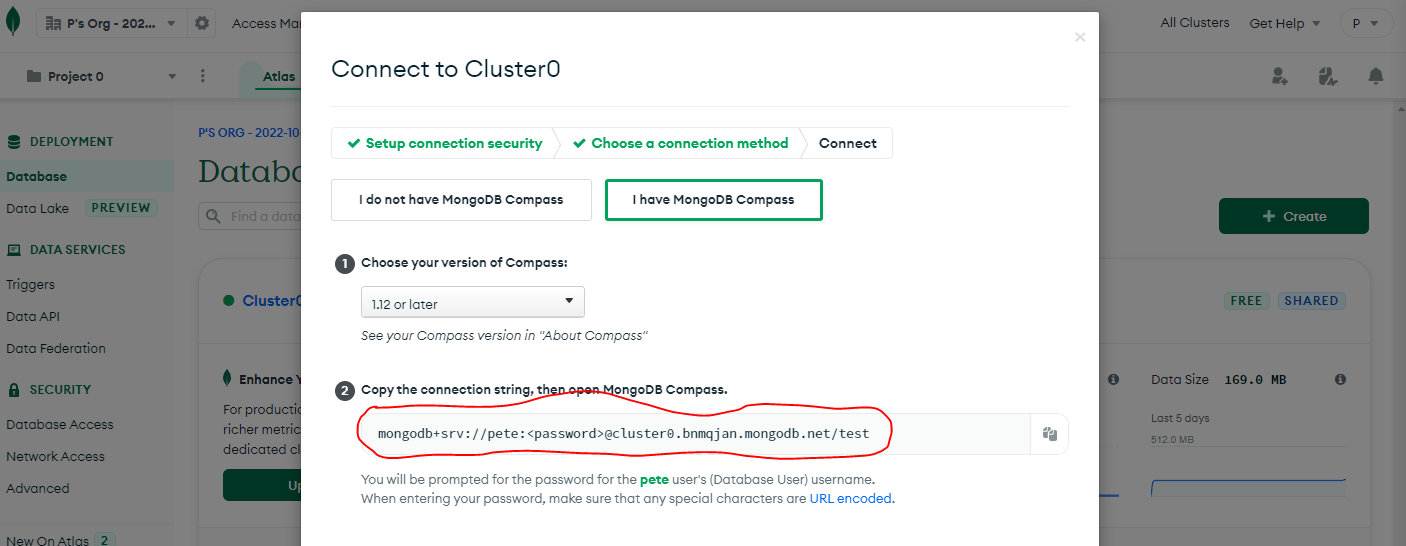
**Step 4.3:** After this code has been entered, save the file and close it. Open the “views.py” file in the djangoapp directory. Replace the code that is in this file with the code found below:



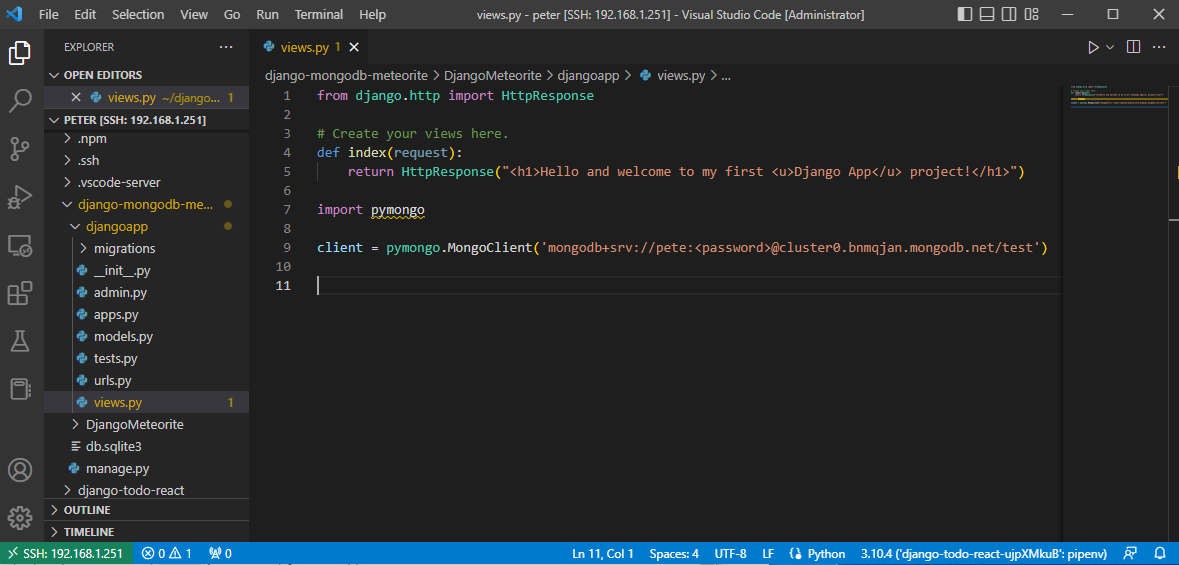
**Step 4.4:** Add the following command “import pymongo” to the bottom of this file to import the PyMongo library.



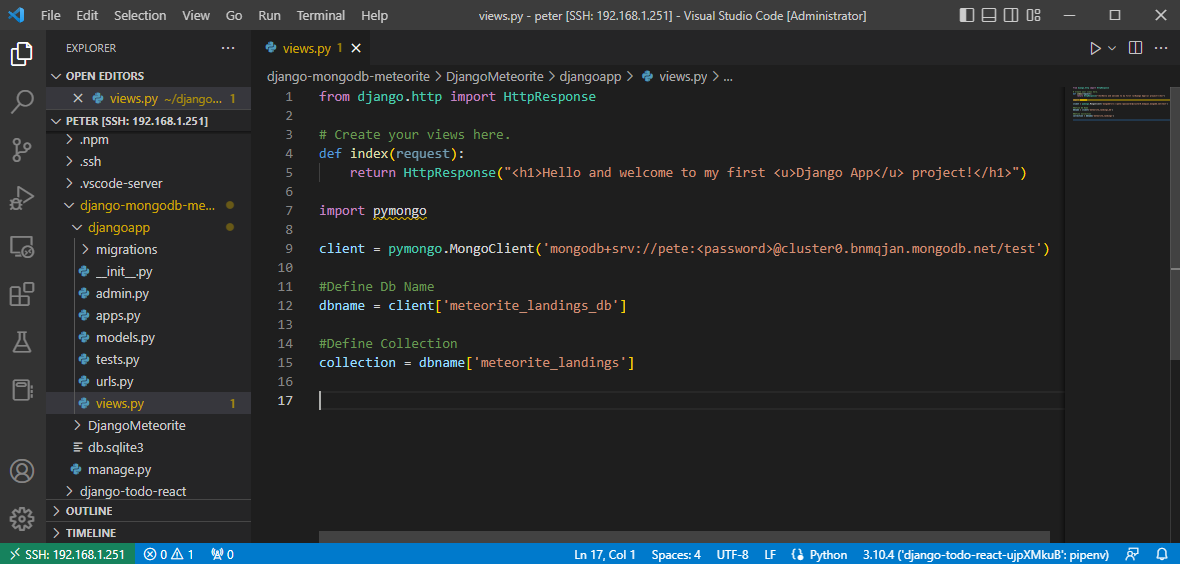
**Step 4.5:** Next, go back to the MongoDB and grab the connection string for the database. To get the connection string, click “Connect” and then “I have MongoDB Compass”.



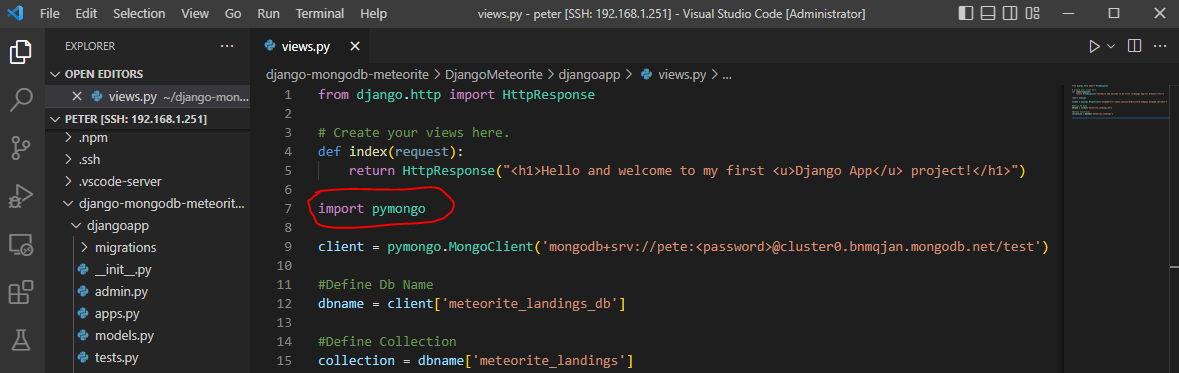
**Step 4.6:** Back on Visual Code Studio in the “views.py”, instantiate the client by typing in “client = pyMongo.MongoClient(‘XXXX’)” where “XXXX” is replaced with the connection string.



**Step 4.7:** Define the database name (meteorite\_landings\_db) and the collection (meteorite\_landings) under the current code as seen below:



**Note:** The error for the pymongo was due to the fact that the wrong python interpreter was selected. To solve this, I opened the command palette by typing ctrl + shift + p, then typed “Python select interpreter” and changed it from the to-do list app to the global one. After this, there were no problems with picking up pymongo.



## 5.0 Adding a Document into a Collection

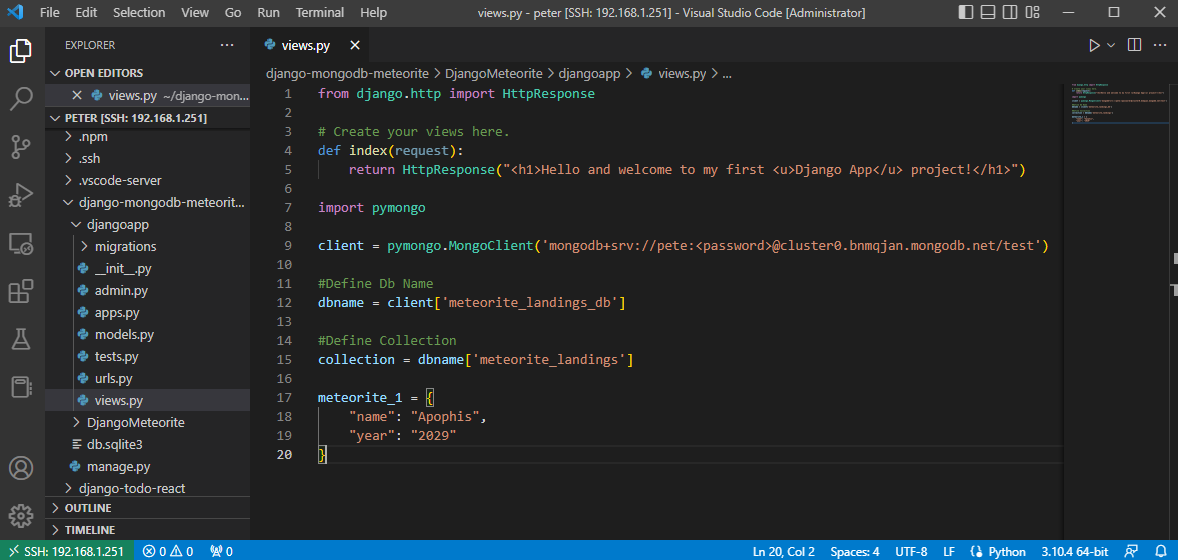
**Step 5.1**: While still on the views.py, type in the following script to add a new meteorite to the database…

meteorite\_1 = {

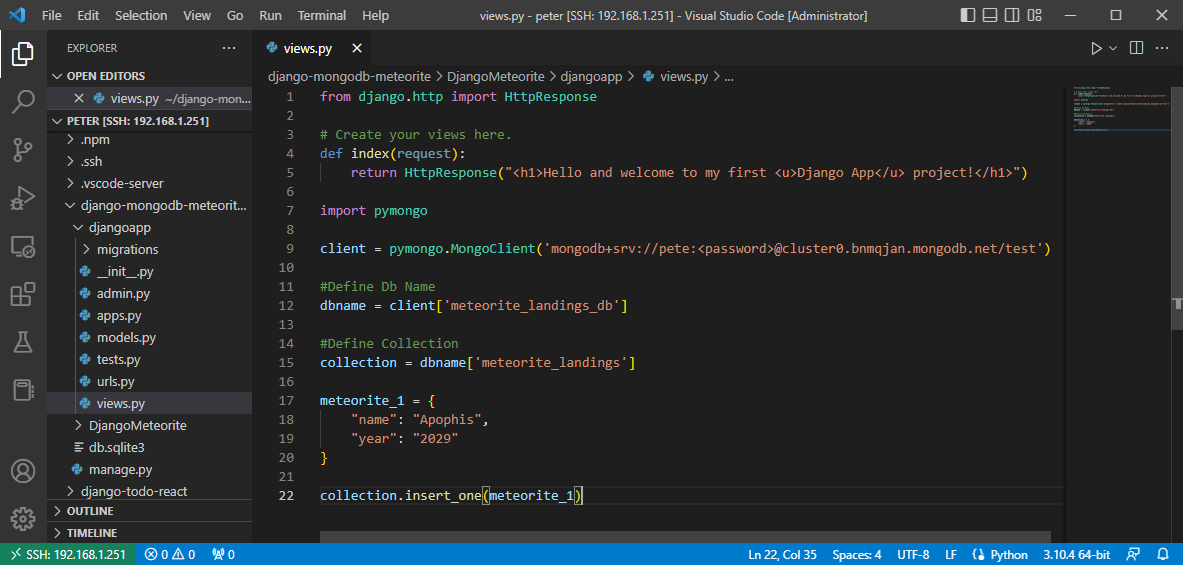
“name”: “Apophis”,

“year”: “2029”

}



**Step 5.2:** Add the command “collection.insert\_one(meteorite\_1)” to the bottom of the code.



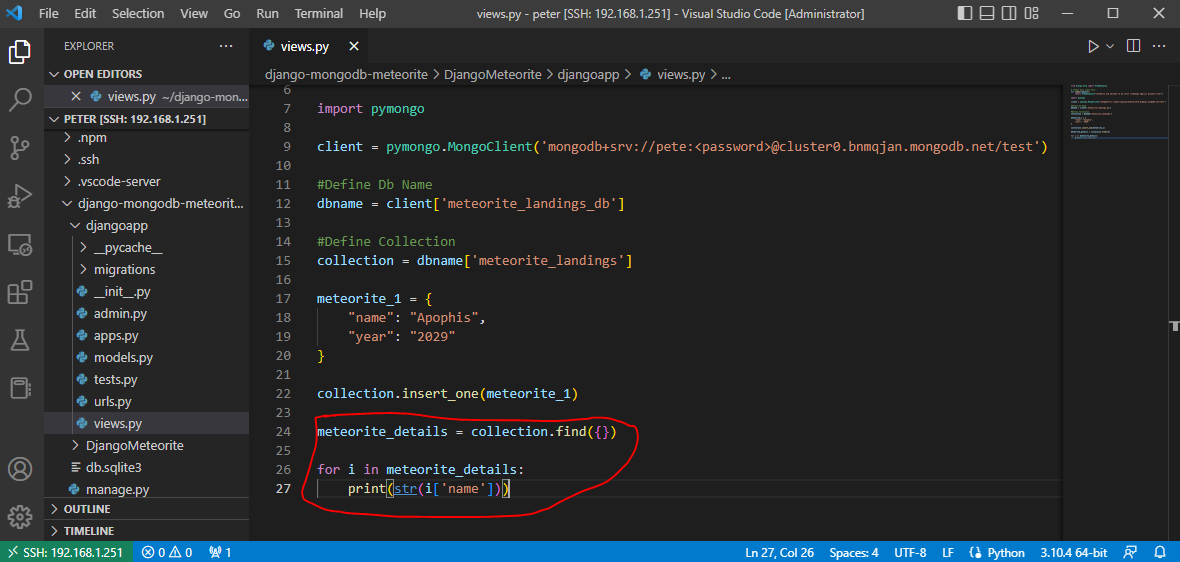
## 6.0 Viewing the Names of the Documents in the Database

Step 6.1: Finally, add the following code to retrieve all the names and years of the meteorites in the database to confirm if the newest meteorite has been added in:

meteorite\_details = collection.find({})

for i in meteorite\_details:

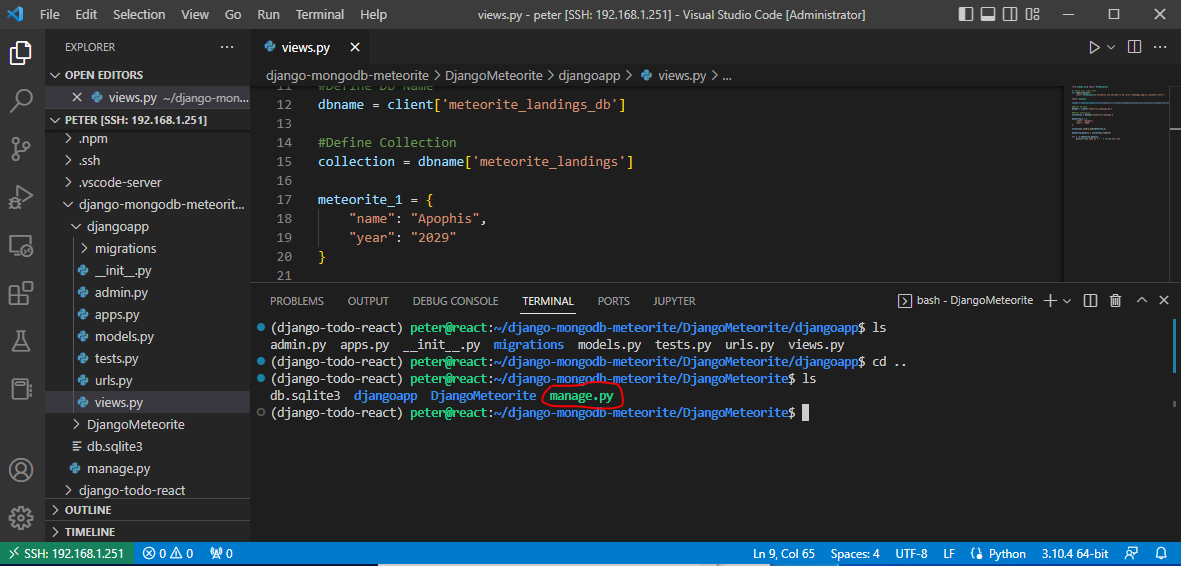
print(str(i[‘name’]))



## 7.0 Test if it Works

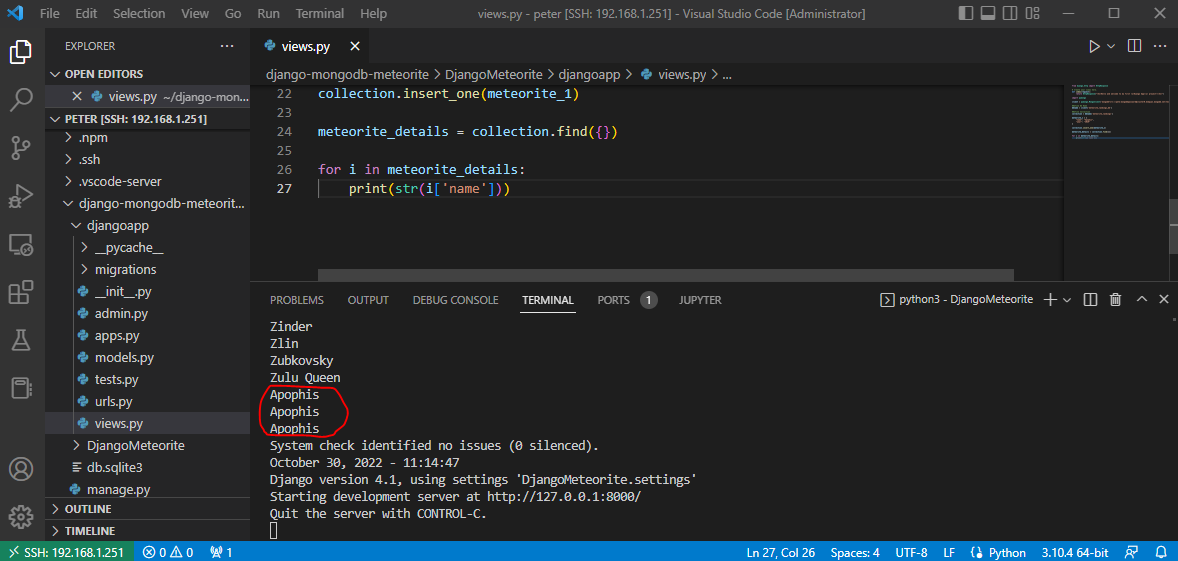
**Note:** Make sure you connection string has the correct password in it instead of “<password>”.

**Step 7.1:** Now, you should be able to go back into terminal, into the directory “DjangoMeteorites” with manage.py.

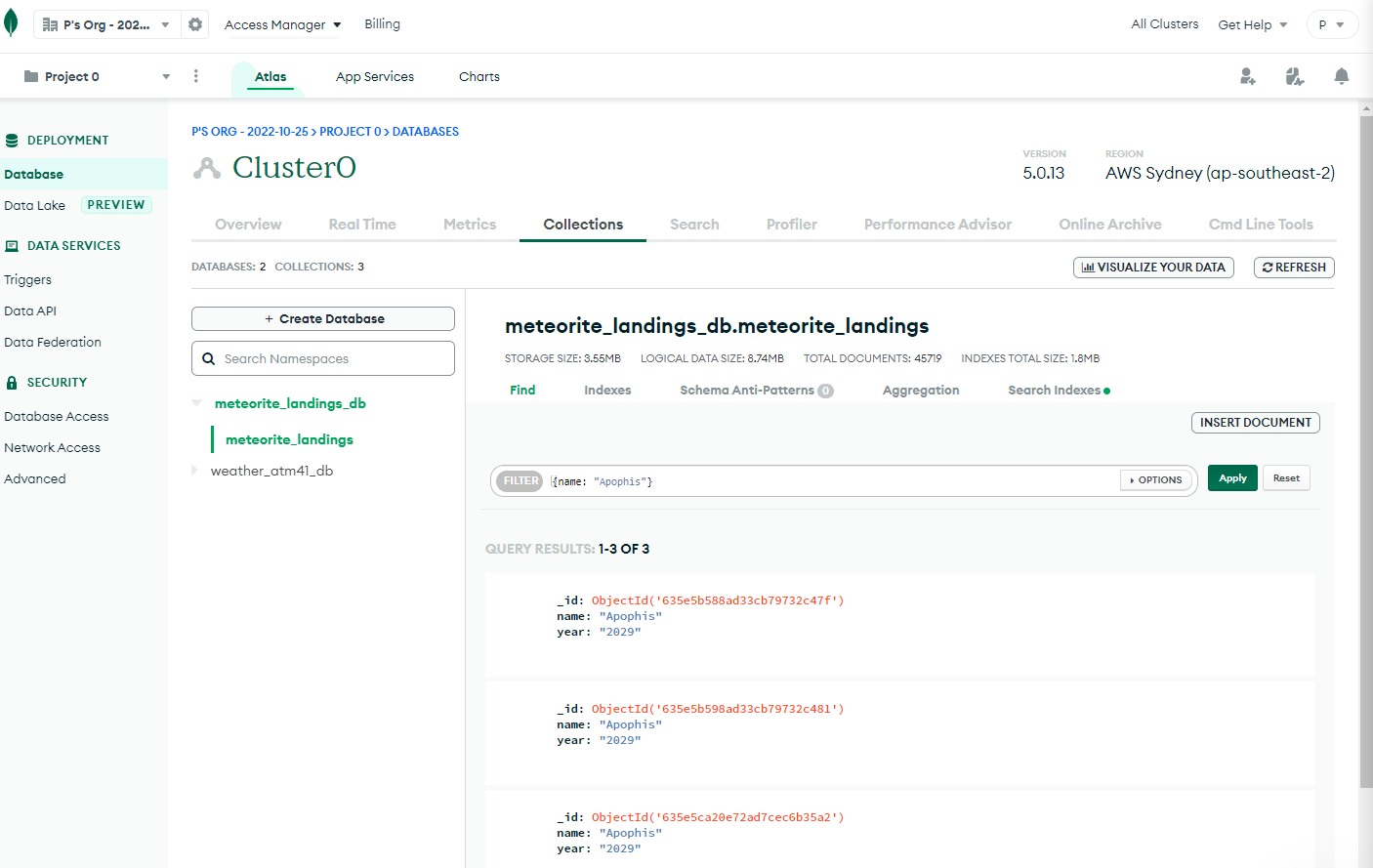


**Step 7.2:** Run the server with the following command “python3 manage.py runserver” as seen below.

**Note:** Did this 3 times re-attempting to get it working, hence 3 documents.



You should see that the Apophis meteorite for 2029 has been added to the collection and printed out. On top of this, if you go to



**Note:** This was found using the filter bar at the top searching for {name: “Apophis”}. Three documents appear due to the fact that this was run three times during debugging.

**Note:** Attempted to also print out the “year”, but as the year was not present for each of the documents, this was an error with the output when that key could not be found.

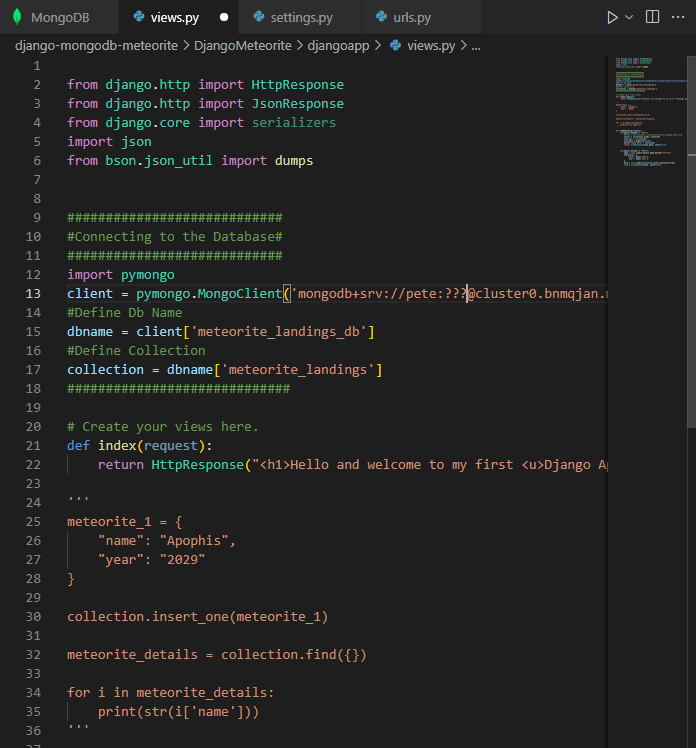
## Additional Resources

Below, you will find some other resources I found useful while putting together this guide:

* Why using MongoDB with PyMongo instead of Djongo: <https://www.youtube.com/watch?v=ORog5kwPd_E>
* A DigitalOcean Tutorial of connecting a Django App to MongoDB with PyMongo (ignore the digital ocean server, create the database on mongodb atlas): <https://www.digitalocean.com/community/tutorials/how-to-connect-a-django-app-to-mongodb-with-pymongo>
* PyMongo Additional Instructions: <https://pymongo.readthedocs.io/en/stable/>
* **[Not recommended, but another possible approach to connecting]** Getting Djongo working with MongoDB: <https://www.youtube.com/watch?v=FLZYLog369s>

# Meteorite API Example

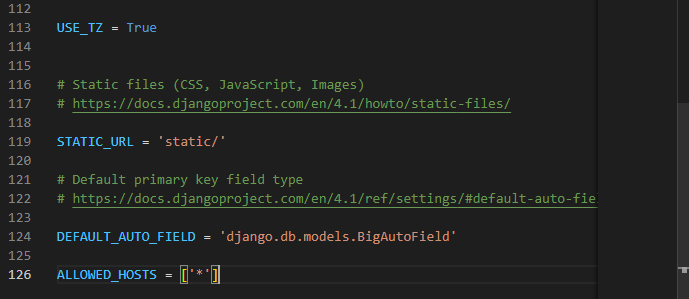
## Visual Studio Code views.py in djangoapp folder:



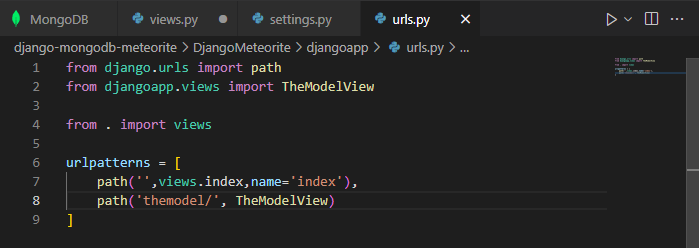


Note: POST method is not working just yet… still needs debugging with the end…

## Visual Studio Code settings.py at the bottom in the DjangoMeteorite:

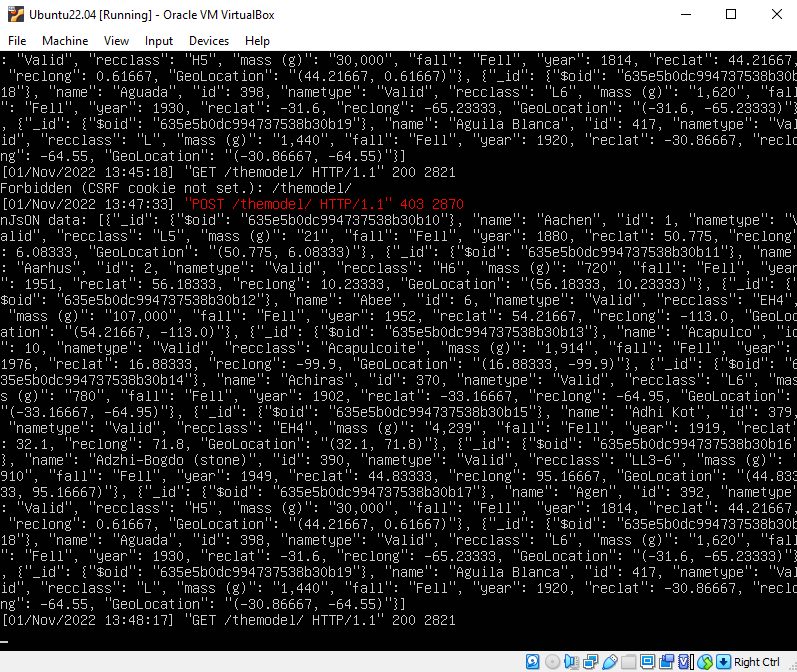


## Visual Studio Code urls.py in djangoapp:

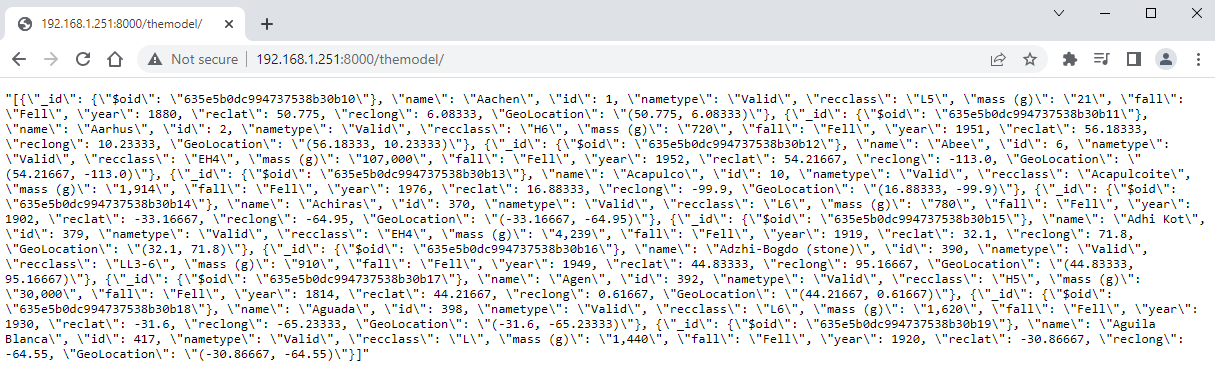


‘

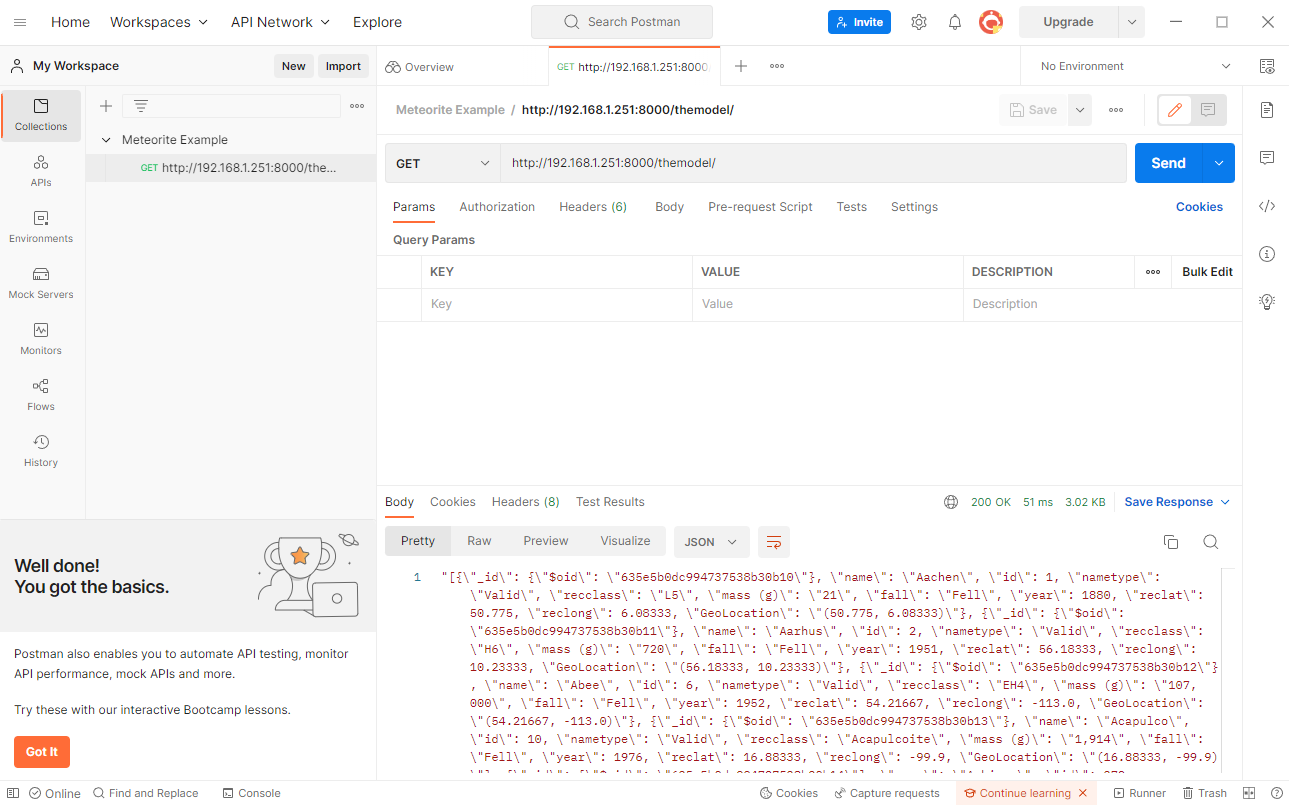
## Virtual Machine Output:



## Web Browser Output:



## Postman Desktop App Output:



## Useful Links:

<https://www.postman.com/downloads/>

<https://www.w3schools.com/python/python_mongodb_limit.asp>

<https://www.geeksforgeeks.org/convert-pymongo-cursor-to-json/>

<https://stackoverflow.com/questions/46909905/how-to-allow-others-to-connect-to-my-django-website>

<https://dev.to/alexmercedcoder/creating-a-restful-api-with-django-without-djangorestframework-17n7>

<https://medium.com/swlh/build-your-first-rest-api-with-django-rest-framework-e394e39a482c>

<http://docs.mongoengine.org/>

<https://www.analyticsvidhya.com/blog/2021/06/how-to-connect-mongodb-database-with-django/>

<https://www.mongodb.com/compatibility/mongodb-and-django>

<https://www.django-rest-framework.org/api-guide/views/>

<https://www.bezkoder.com/django-mongodb-crud-rest-framework/>