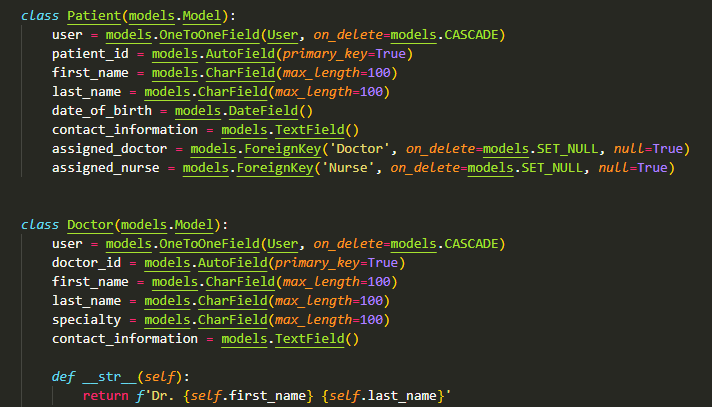
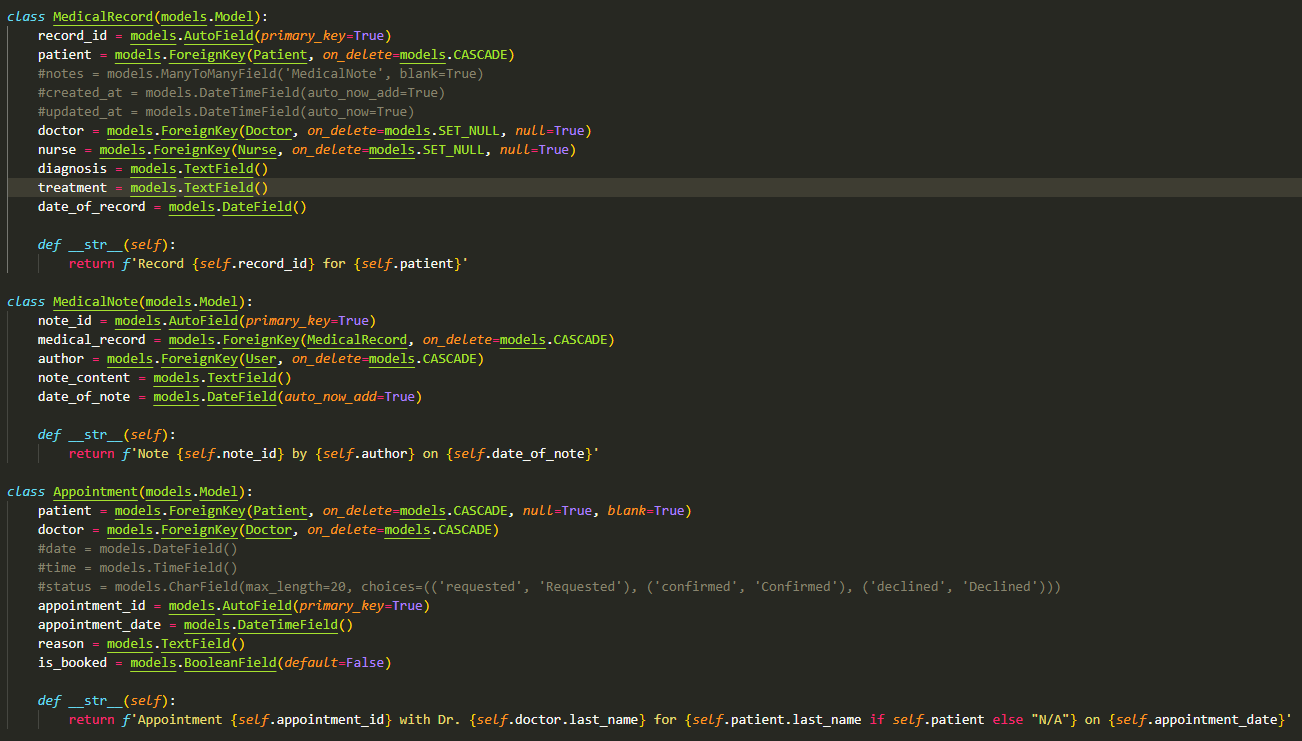
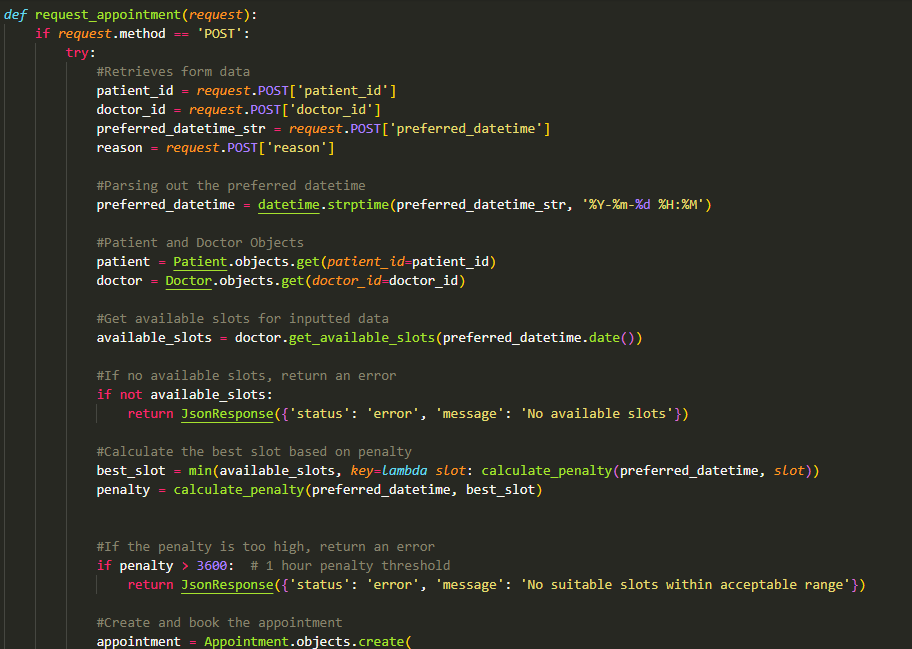
The artifact I’m submitting are the changes based on Milestone 4’s feedback and also the database items. The database items will include the actual MongoDB database with Jupyter Notebook as well (Jupyter was an easy way to be able to show you that the database was actually created – I did seed it via Python Django, however, there’s no image that the database was created correctly. So Jupyter is just a way for your assurance).

With feedback given from the previous module, I first ended up updating **models.py** to incorporate some different elements. Under the Doctor class, you’ll now see a “specialty”. The “Patient” class received an update to include the assigned doctor/nurse to the patient



Also, everyone else received minor updates as well. The MedicalRecord class I tagged in the doctor, nurse, diagnosis, treatment, and date\_of\_record. Now looking back on this, though I commented out for this task, It’s best that I incorporate back in the note as logically, it doesn’t make sense to leave it out of the entire record, but the whole record doesn’t need to show within the note.



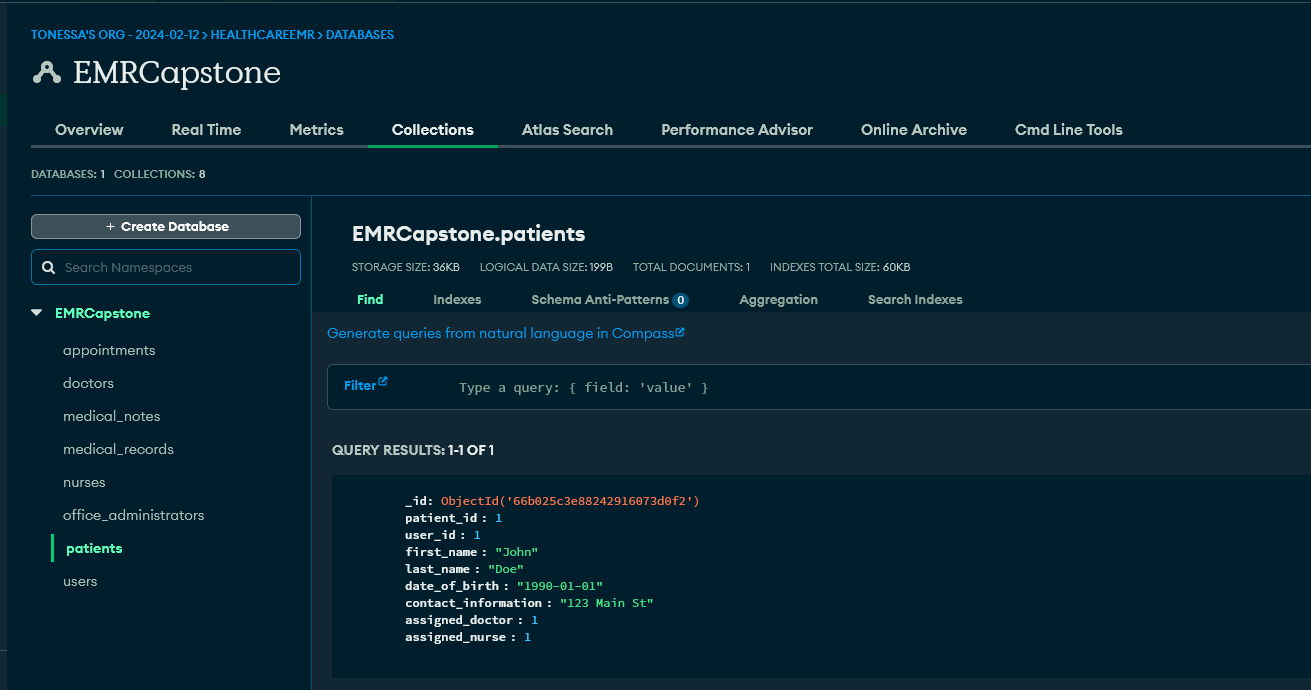
Within the views.py file, I incorporated some proper error handling and as you see below, incorporated more comments within this section. I will be going back to add more comments into other sections before the final. I did receive feedback to move the penalty portion of the code into the .utils for cleaner code organization. I can see that but with the error handling, I’m confused on if that should also move over with it as well?  
  


A black screen with yellow and green text

Description automatically generated



As far as the Database goes, I have the seven collections (essentially, I delved this out to what would need primary key type IDs) and have some records in for the patient, nurse, doctor, and even office administrator. I did the administrator different to show some security issues within the database currently:



A screenshot of a computer

Description automatically generated

A screenshot of a computer

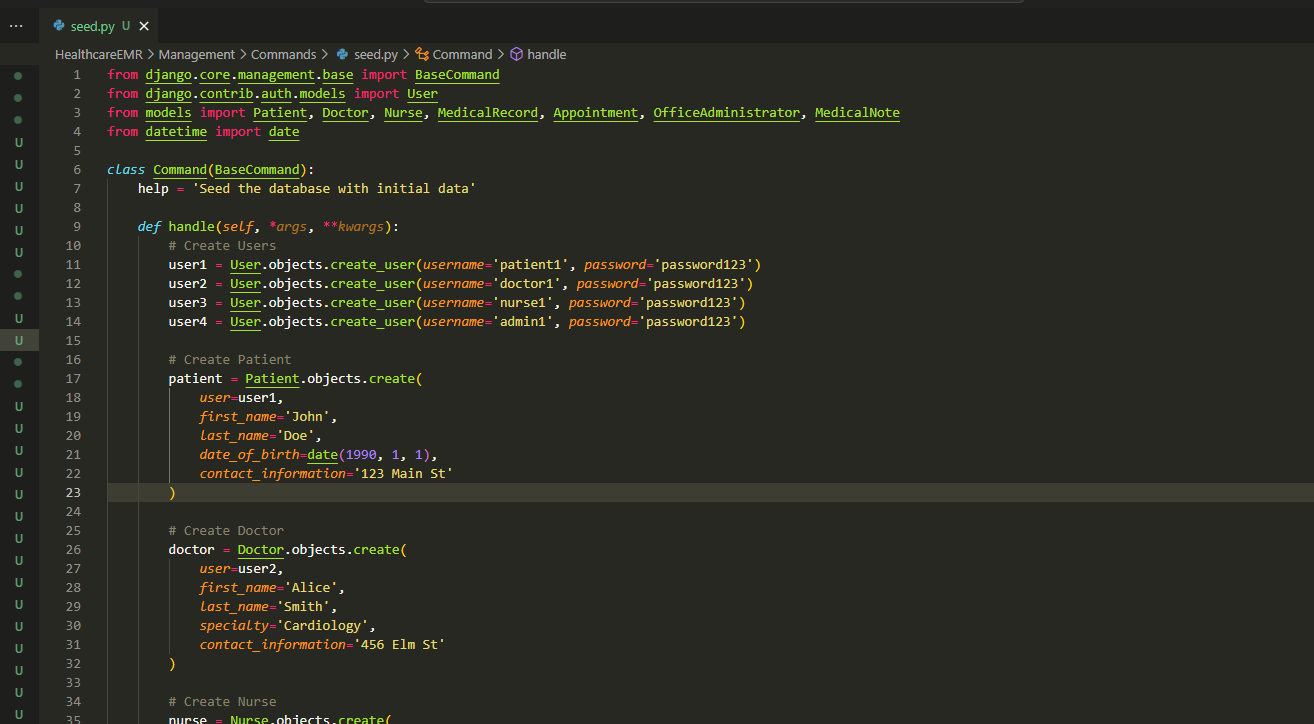
Description automatically generated

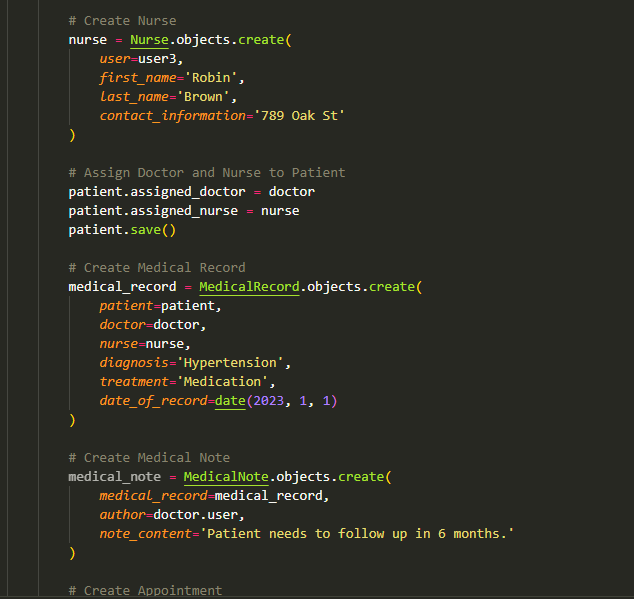
A screenshot of a computer

Description automatically generated

As you can see above, the main issue is that “contact\_information” is generic. Are we using address or are we going by email address? If there’s not a specific parameter for the field.

I also incorporated another file called **seed.py**. This seeds the information into the database from the terminal line:





A screen shot of a computer program

Description automatically generated

And I also created **init\_db.py** to initialize the database an create indexes within the collections (the lavender portion blocks out my actual password to the database):

