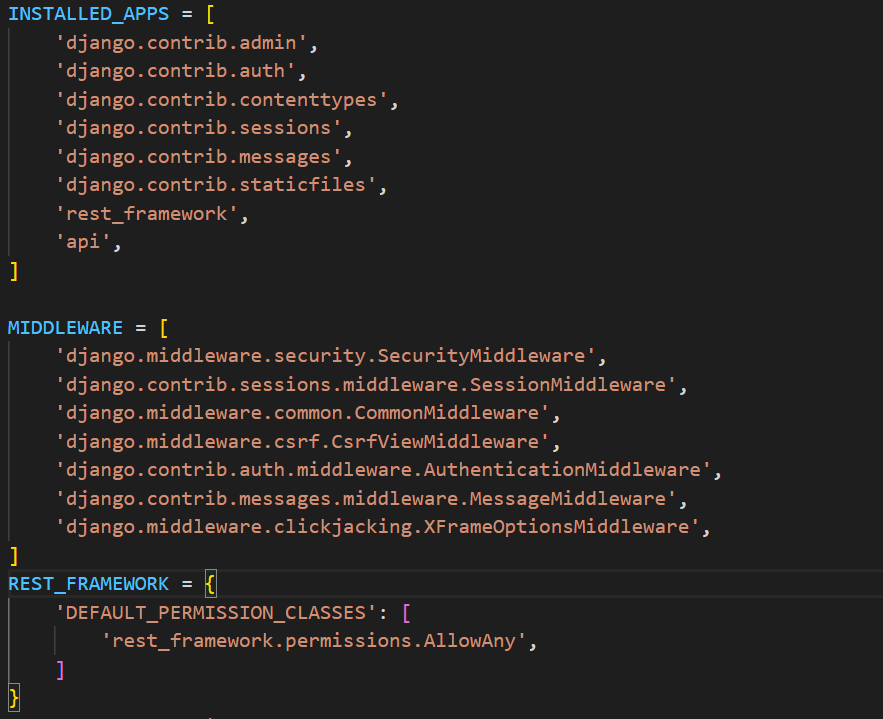
PRODUCT IO

CODE EXPLANATION

Step1: Create a Django project “pio” and start a new app “api” in it.

Step2: We are using “*Django Rest FrameWorks”* **and** our application “*api”* . We need to add those under applications in the INSTALLED\_APPLICATIONS in settings.py file under our project.

Step3:We also need to add REST\_FRAMEWORK in our settings.py as a separate entity as we need to give permissions inside it how we want to authoroize our api.



Step 4: We need to create a database for our api. We are using the inbuilt ORM(Object Relational Model) structure of the Django and creating a sqlllite3 database using models. We need to create Users model for user data, Sellers model for sellers data, Products model for product data, Product Seller model for the Seller who is selling that specific product data, Orders model for Order data, Address model for address data.

The attributes of different models are designed in such a way that we need to cover all the logical combinations of data in models.

For suppose a user may have multiple addresses like home, office etc,

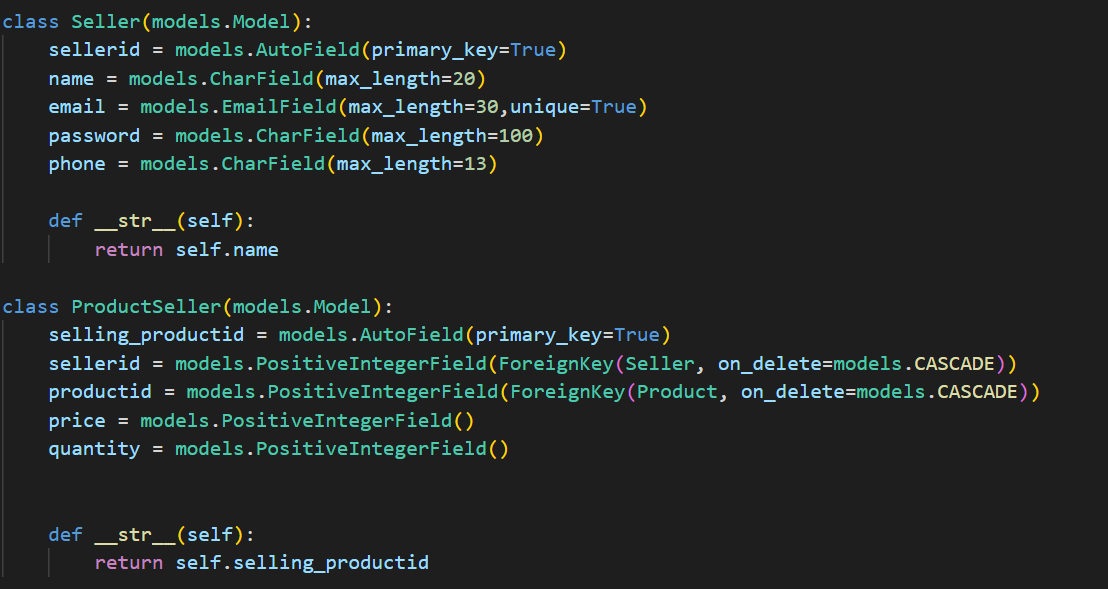
So we created a new address model to map multiple addresses to a user using userid.

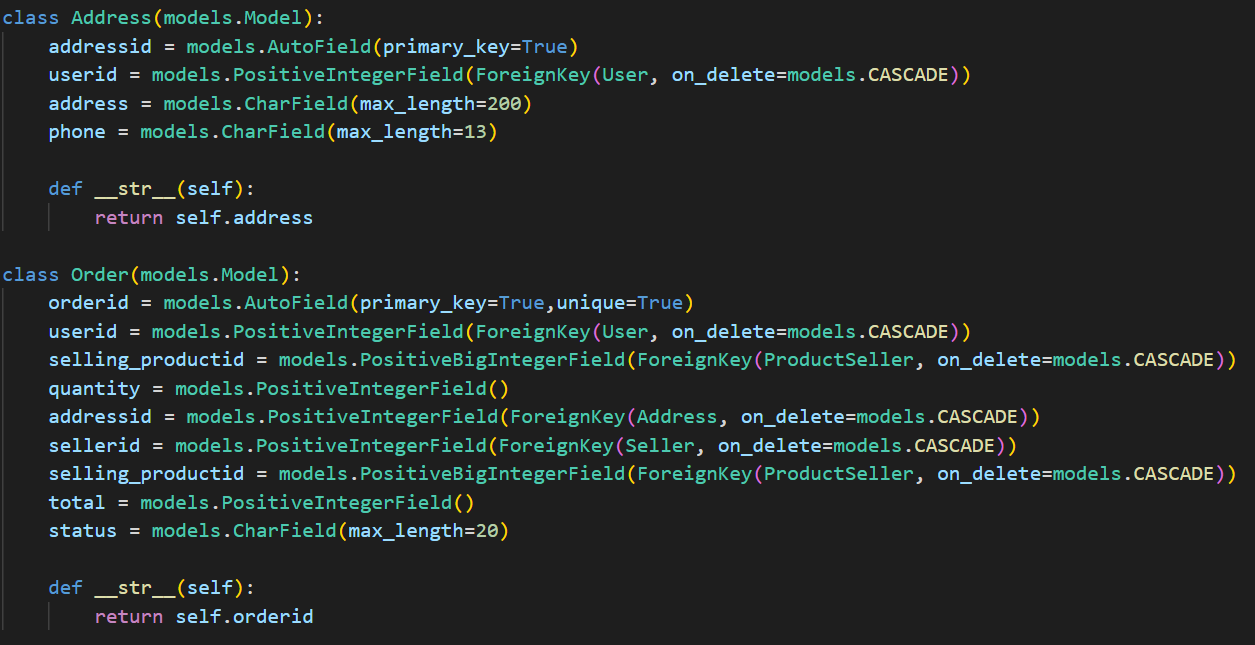
Similarly a product can be sold by different sellers, So we created a new model of Product Seller with the respective attributes. As each seller may want to sell that products at a different price we need to map the ProductSeller model to Product and Seller models using product and seller id respectively

Similarly a user may place multiple orders at a time so we created a new Orders model and map it to User model using userid.

And etc







Step5: We need to create a serializers.py file to include different serializers for each model as we need to make the complex model data into user readable and easily interpretable json data in our case.

For example UserSerializer works on the model User and changes all its fields to JSON data. Similarly others



Step6:We n