- Complete Guide to Django, MongoDB, and Your Project
- What is Django?

Django is a **Python web framework** that helps build web applications. It allows handling **databases**, creating **APIs**, and managing **data easily**. Django follows **MVT** (**Model-View-Template**) architecture:

- Model

 → Defines database structure.
- View → Handles user requests and returns data.
- Template → Displays data (not needed for your assignment).

For your project, you'll mostly work with:

- Models (MongoDB structure)
- Views (APIs to fetch and modify data)
- Serializers (to format data for APIs)
- Django Admin (to manage data easily)

Folder & File Structure in Your Project

Your Django project contains two main parts:

- 1. data_platform/ (Django Project)
- 2. analytics/ (Django App)
- data_platform/ (Django Project)

This is the main Django project folder.

File	Purpose
initpy	Marks this as a Python package .
settings.py	Main configuration file (database, apps, security settings, etc.).
urls.py	Manages URL routing (connects APIs and pages).
wsgi.py	Helps deploy Django on web servers .
asgi.py	Alternative to wsgi.py (for async applications).

analytics/ (Django App)

This is where your main code lives.

Django apps are like "modules" that handle different features.

This is where your main code lives . Django apps are like "modules" that handle different features .		
File	Purpose	
initpy	Marks this as a Python package .	
admin.py	Registers models for the Django Admin Panel .	
apps.py	Basic configurations for the app .	
models.py	Defines database structure (MongoDB models).	
views.py	Contains functions to process API requests.	
tests.py	Used for testing the application (not needed for now).	
migrations/	Stores database changes (not needed for MongoDB).	

How Will You Use Django & MongoDB for Your Assignment?

Step 1: Store Data in MongoDB

• The CSV files (amazon_sale_report.csv, etc.) contain **e-commerce sales data**.

Goal: Import this data into MongoDB so Django can access it.

Step 2: Define a MongoDB Model

Edit analytics/models.py and define a model:

```
python
                                                                       Copy
from djongo import models
class AmazonSaleReport(models.Model):
    order_id = models.CharField(max_length=100)
    date = models.DateField()
    sales_channel = models.CharField(max_length=100)
    sku = models.CharField(max_length=100)
    category = models.CharField(max_length=100)
    amount = models.FloatField()
    objects = models.DjongoManager()
    def __str__(self):
    return f"Order {self.order_id} - {self.category}"
```

This creates a table in MongoDB!

Step 3: Connect Django to MongoDB

Modify data_platform/settings.py:

```
python
DATABASES = {
     'default': {
         'ENGINE': 'djongo',
'NAME': 'data_analysis_db',
         'CLIENT': {
              'host': 'mongodb://127.0.0.1:27017/',
```

✓ Start MongoDB on macOS:

Step 4: Create an API to Fetch Data

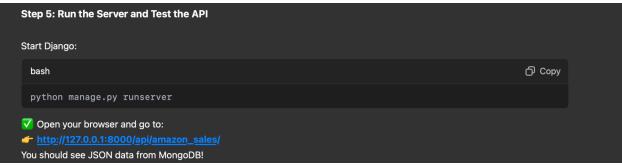
```
Edit analytics/views.py:
                                                                                                              〇 Copy
  python
  from rest_framework import viewsets
  from .models import AmazonSaleReport from .serializers import AmazonSaleReportSerializer
  class AmazonSaleReportViewSet(viewsets.ModelViewSet):
       queryset = AmazonSaleReport.objects.all()
       serializer_class = AmazonSaleReportSerializer
 Create analytics/serializers.py:
  python

    Copy

  from rest_framework import serializers
from .models import AmazonSaleReport
  class AmazonSaleReportSerializer(serializers.ModelSerializer):
       class Meta:
            model = AmazonSaleReport
            fields = '__all__
Define API URLs in analytics/urls.py:
 python
                                                                                                              〇 Copy
 from django.urls import path, include
from rest_framework.routers import DefaultRouter
 from .views import AmazonSaleReportViewSet
 router = DefaultRouter()
 router.register(r'amazon_sales', AmazonSaleReportViewSet)
 urlpatterns = [
      path('', include(router.urls)),
Connect APIs in data_platform/urls.py:

    ○ Copy

 python
 from django.contrib import admin
  from django.urls import path, include
 urlpatterns = [
      path('admin/', admin.site.urls),
path('api/', include('analytics.urls')),
```



Summary: How Everything Works Together Component **Role in Your Project** Django Project (data_platform) Main project, settings, URLs. Django App (analytics) Handles database and APIs. MongoDB Stores sales data. Models (models.py) Defines data structure in MongoDB. Views (views.py) Creates API endpoints. Serializers (serializers.py) Formats data for API. Admin (admin.py) Manages data via Django admin. **o** Next Steps Complete all the above steps. Verify MongoDB is connected (mongo --eval "db.runCommand({ connectionStatus: 1 })"). Run Django server and test API (http://127.0.0.1:8000/api/amazon_sales/). Import CSV data into MongoDB (I'll guide you on this next!).