Django Developer Test Assignment (Advanced Analytics API)

Objective:
Build a small **Sales Analytics API** using **Django REST Framework**.
This test checks your ability to design models, write analytics queries, and structure APIs.

1. Models

Customer
- id: AutoField (Primary Key)
- name: CharField(max_length=100)
- email: EmailField(unique=True)
- joined_on: DateTimeField(auto_now_add=True)

Product
- id: AutoField (Primary Key)
- name: CharField(max_length=100)
- price: DecimalField(max_digits=10, decimal_places=2)

Order
- id: AutoField (Primary Key)
- customer: ForeignKey(Customer, on_delete=CASCADE)
- order_date: DateTimeField(auto_now_add=True)

OrderItem
- id: AutoField (Primary Key)
- order: ForeignKey(Order, on_delete=CASCADE, related_name='items')
- product: ForeignKey(Product, on_delete=CASCADE)
- quantity: IntegerField(default=1)

2. API Endpoints

Endpoint	Method	Description
/api/customers/	GET, POST	List or create customers
/api/products/	GET, POST	List or create products
/api/orders/	GET, POST	List or create orders (with nested items)
/api/analytics/sales-summary/	GET	Get total sales, customers, products sold
/api/analytics/top-customers/	GET	Get top 5 customers by purchase amount
/api/analytics/top-products/	GET	Get top 5 most sold products

3. Requirements

- Implement nested serializers for Order and OrderItem.
- Optimize queries using select_related and annotate.
- Quantity must be ≥ 1.
- Each order must have at least one item.
- Include total price per order in response.

- Use proper HTTP status codes.

4. Bonus (Optional)

- Add JWT authentication.
- Add date range filters for analytics (e.g. ?from=2024-01-01&to;=2024-12-31).
- Write unit test for one analytics endpoint.

5. Submission

- Upload to public GitHub repository.
- Include README.md with setup steps.
- Include example Postman requests or API docs.

6. Evaluation Criteria

Area	Description
Models	Correct relationships and constraints
Query Optimization	Efficient use of ORM methods
Code Quality	Readable, structured, modular
API Responses	Clear and RESTful
Validation	Proper input and business rules
Bonus	JWT, filters, analytics tests