

Backend Developer Assignment – Sabhyasha Retail Tech Pvt. Ltd.

Role: Django Developer

Company: Sabhyasha Retail Tech Pvt. Ltd.

Submission Deadline: As mentioned on email .

Objective

Build a basic e-commerce-style backend system using **Django**, **Django REST Framework (DRF)**, and **MySQL** (configured using XAMPP). This assignment is meant to assess your backend development skills, API architecture, and familiarity with the required technologies.

Project Requirements

1. Database Models (Using MySQL via XAMPP)

Set up your MySQL database using **phpMyAdmin (XAMPP)** and define the following models in Django:

Custom User Model

Do **NOT** use Django's built-in **User** model. Instead, create your own by extending **AbstractUser**.

Fields:

- email
- username
- password

- `is_seller` (Boolean)
- `is_buyer` (Boolean)

✓ **Product**

- `name, description, price, stock,`
- `seller` → ForeignKey to **User**

✓ **Order**

- `buyer` → ForeignKey to **User**
- `order_date`
- `status`

✓ **OrderItem**

- `order` → ForeignKey
 - `product` → ForeignKey
 - `quantity`
 - `price`
-

2. API Endpoints (Using Django REST Framework)

Develop RESTful APIs using **DRF** to support the following:

- ☒ **User Registration** (Register as buyer or seller)
 - ☒ **Login (Use JWT Authentication via SimpleJWT ONLY)**
 - ☒ **Add Product** (Access limited to sellers)
 - ☒ **List Products** (Accessible to all users)
 - ☒ **Create Order** (Access limited to buyers)
 - ☒ **List Orders**
 - Buyer: See own orders
 - Seller: See orders that include their products
 - ☒ **Order Details with Order Items**
-

3. Bonus Features (Extra Points)

- ☒ Pagination on product listing
 - ☒ Filter products by name or price
 - ☒ Basic validation and permission checks (e.g., role-based access control)
-

Tech Stack

- Django
 - Django REST Framework
 - MySQL (via XAMPP/phpMyAdmin)
 - SimpleJWT (for Token Authentication)
 - Postman (for API Testing)
-

Submission Guidelines

✦ Push the entire project to a **public GitHub repository** and include:

- **README.md** file with:
 - Clear setup instructions (env, DB, etc.)
 - API documentation (endpoints, auth, request/response samples)
 - Export the MySQL database as a **.sql file** and include it in the repo
-

Important Notes

⚠️ Use JWT authentication **ONLY** via [SimpleJWT](#)

⚠️ Do not use Django's built-in **User** model – create a custom one

✅ **Good Luck!**

We look forward to seeing your implementation.

Best Regards,

Sabhyasha Retail Tech Pvt. Ltd.
