

Models

RoleModel

Field Name	Datatype	Primary Key	Comments
id	Integer	Yes	Autoincrement
rolename	String	No	Values: "CUSTOMER", "ADMIN"

UserModel

Field Name	Datatype	Primary Key	Foreign Key	Comments
id	Integer	Yes	No	Autoincrement
username	String	No	No	-
email	String	No	No	Unique
password	String	No	No	-
role	Integer	No	Yes	FK to RoleModel

OrderModel

Field Name	Datatype	Primary Key	Foreign Key	Comments
id	Integer	Yes	No	Autoincrement
customerId	Integer	No	Yes	FK to UserModel
productName	String	No	No	-
quantity	Integer	No	No	-
unitPrice	Double	No	No	-
discountPct	Double	No	No	Default: 0.0
taxPct	Double	No	No	Default: 18.0
totalPrice	Double	No	No	Calculated: (quantity * unitPrice) - discount + tax
orderDate	String	No	No	Default: Current date (dd/MM/yyyy)
status	String	No	No	Default: "pending"

JWT Authentication

- JWT token required in Authorization header:

Authorization: Bearer <JWT_TOKEN>

- Roles (CUSTOMER / ADMIN) are identified from the token.
- All endpoints except /login require authentication.

Endpoints

1. POST /login

- Purpose: Authenticate user and return JWT.

- Request Body:

```
'json
{
  "email": "cust1@gmail.com",
  "password": "cust123$"
}'
```

- Success Response (200 OK):

```
'json
{
  "email": "cust1@gmail.com",
  "access": "<access_token>",
  "refresh": "<refresh_token>",
  "status": 200
}'
```

2. POST /order/create (CUSTOMER only)

- Creates a new order.

- Request Body:

```
'json
{
  "productName": "Smartphone",
  "quantity": 2,
  "unitPrice": 15000,
  "discountPct": 10,
  "taxPct": 18
}'
```

- Manipulations:

- discount = (quantity * unitPrice) * (discountPct / 100)
- tax = ((quantity * unitPrice) - discount) * (taxPct / 100)
- totalPrice = (quantity * unitPrice) - discount + tax

- Success Response (201 CREATED):

```
'json
{
  "id": 1001,
  "customerId": 1,
  "productName": "Smartphone",
  "quantity": 2,
  "unitPrice": 15000,
```

```
        "discountPct": 10,  
        "taxPct": 18,  
        "totalPrice": 29700,  
        "orderDate": "30/11/2025",  
        "status": "pending"  
    }  
  
---
```

3. GET /order/list?minTotal=20000&maxTotal=50000&status=pending (Both roles)

- Returns orders filtered by:
 - minTotal (orders with totalPrice ≥ given value)
 - maxTotal (orders with totalPrice ≤ given value)
 - status (pending/confirmed/shipped)
- Success: 200 OK with list.
- Error: 400 Bad Request → "no orders found"

4. PATCH /order/update/{id} (ADMIN only)

- Updates order status.
- Request Body:

```
json  
{ "status": "shipped" }
```
- Success Response (200 OK): Updated order object.
- Error: 400 Bad Request → "order not found"

5. DELETE /order/delete/{id} (Conditional access)

- Accessible by:
 - Admin
 - Customer who created the order
- Success: 204 No Content → "deleted successfully"
- Error:
 - 400 Bad Request → "order not found"
 - 403 Forbidden → "you don't have permission"

Response Codes

- 200 OK – Success
- 201 CREATED – Resource created
- 204 No Content – Deleted successfully
- 400 Bad Request – Invalid input / Not found
- 403 Forbidden – Permission denied

- ✓ This new problem statement adds extra manipulations:
- Discount and tax calculations before arriving at totalPrice.
- Query params with three filters: minTotal, maxTotal, status.
