

## Order Management System (OMS) REST API

### Assignment Submission

Below are the screenshots from Postman demonstrating the successful execution of all required scenarios.

#### Scenario 1: Create an Order Endpoint: POST /orders

The screenshot shows the Postman interface with a collection named "campusmate" selected. A POST request is being made to `http://127.0.0.1:5000/orders`. The request body is a JSON object representing an order with two items:

```
1: { 2:   "customer_id": 1, 3:   "order_date": "2025-11-28", 4:   "shipping_contact_mech_id": 1, 5:   "billing_contact_mech_id": 2, 6:   "order_items": [ 7:     { 8:       "product_id": 1, 9:       "quantity": 2, 10:      "status": "Pending" 11:    }, 12:    { 13:      "product_id": 2, 14:      "quantity": 1, 15:      "status": "Pending" 16:    } 17:  ] 18: }
```

The response status is 201 CREATED, indicating success.

#### Scenario 2: Retrieve Order Details Endpoint: GET /orders/{id}

The screenshot shows the Postman interface with a collection named "campusmate" selected. A GET request is being made to `http://127.0.0.1:5000/orders/1`. The request includes a query parameter "Key" with the value "Value".

The response status is 200 OK, indicating success. The response body is a JSON object representing the retrieved order:

```
1: { 2:   "billing_contact_mech_id": 2, 3:   "customer_id": 1, 4:   "items": [ 5:     { 6:       "order_id": 1, 7:       "order_item_seq_id": 1, 8:       "product_id": 1, 9:       "quantity": 2, 10:      "status": "Pending" 11:    }, 12:    { 13:      "order_id": 1,
```

**Scenario 3: Update an Order Item** *Endpoint: PUT /orders/{id}/items/{item\_id}* *Action: Updated quantity of Jeans to 2.*

The screenshot shows the Postman interface with the following details:

- Collection:** campusmate
- Request Type:** PUT
- URL:** http://127.0.0.1:5000/orders/1/items/2
- Headers:** (9)
- Body:** raw JSON
- Body Content:**

```
1 {
2     "quantity": 2
3 }
```
- Response Status:** 200 OK
- Response Time:** 46 ms
- Response Size:** 271 B

**Scenario 4: Add an Order Item** *Endpoint: POST /orders/{id}/items* *Action: Added "Sneakers" to the order.*

The screenshot shows the Postman interface with the following details:

- Collection:** campusmate
- Request Type:** POST
- URL:** http://127.0.0.1:5000/orders/1/items
- Headers:** (9)
- Body:** raw JSON
- Body Content:**

```
1 {
2     "product_id": 3,
3     "quantity": 1,
4     "status": "Pending"
5 }
```
- Response Status:** 201 CREATED
- Response Time:** 39 ms
- Response Size:** 276 B

**Scenario 5: Delete an Order Item** *Endpoint: DELETE /orders/{id}/items/{item\_id}* *Action: Deleted "T-Shirt" from the order.*

The screenshot shows the Postman interface. On the left, the sidebar lists collections: 'campusmate' (with 'GET CampusMate API'), 'Hotwax JAVA' (with 'GET JAVA create order', 'GET JAVA get', 'GET New Request'), and 'Hotwax python' (with 'GET post python', 'GET get python', 'GET python put', 'GET python add item', 'GET python delete a item', 'GET New Request'). The main area shows a 'python delete a item' request. The 'DELETE' method is selected, and the URL is 'http://127.0.0.1:5000/orders/1/items/1'. The 'Params' tab is active, showing a table with one row: 'Key' (Value) and 'Description'. Below the table, the 'Body' tab is selected, showing a JSON response: { "message": "Item deleted successfully" }. The status bar at the bottom right indicates '200 OK' with a 35 ms response time and 210 B size.

**Scenario 6: Delete an Order** *Endpoint: DELETE /orders/{id}*

The screenshot shows the Postman interface. The sidebar lists the same collections as before. The main area shows a 'Delete order' request. The 'DELETE' method is selected, and the URL is 'http://127.0.0.1:5000/orders/1'. The 'Params' tab is active, showing an empty table. Below the table, the 'Response' tab is selected, showing a small cartoon character pointing upwards and the text 'Click Send to get a response'.

**JWT Authentication** Implemented secure login to protect all API endpoints.

## 1. User Registration Endpoint: POST /register

The screenshot shows the Postman interface with a collection named "campusmate". A POST request is made to `http://127.0.0.1:5000/register`. The request body contains the following JSON:

```
1 {
2     "username": "jayaditya",
3     "password": "mypassword123"
4 }
```

The response status is **201 CREATED**, with a response time of 3.18 s and a response size of 218 B. The response body is:

```
1 {
2     "message": "User registered successfully"
3 }
```

## 2. User Login (Token Generation) Endpoint: POST /login

The screenshot shows the Postman interface with the same "campusmate" collection. A POST request is made to `http://127.0.0.1:5000/login`. The request body contains the following JSON:

```
1 {
2     "username": "jayaditya",
3     "password": "mypassword123"
4 }
```

The response status is **200 OK**, with a response time of 2.76 s and a response size of 516 B. The response body is a JSON object containing an access token:

```
1 {
2     "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyImcnVzAcT6zfsc2UsmlhdI6MTI2NDM0OTcxMSwi��iOiTQzM2M4ZTMtMjY40C80MGFkLWE4NWUtMGYwN2ViO9hjYj3jiwidHlwZSI6ImFjY2VzcyIsInN1Yi6IjEiLCJu  
YmYiOjE3NjQ2MTk3MTEsImlzcwY101i1Nwfzmi0M0C8ZjBhLRHYTET0WniysI10TM6NDuyowZfzjUlc31eHai0je9Nj0zmjA2MF9.  
XXX4t7qeCnL_cjacgbwZEIQVcbLtc6alTn2xCIx-384"
3 }
```

### 3. Authorized Request Endpoint: GET /orders/{id} (Using Bearer Token)

The screenshot shows a REST API testing interface with the following details:

- Left Sidebar (API Structure):** Lists various endpoints categorized by service:
  - campusmate**: GET JAVA create order, GET JAVA get, GET update order, GET add new item, GET delete item, GET delete order, GET update order address.
  - Hotwax JAVA**: GET JAVA create order, GET JAVA get.
  - Hotwax python**: GET post python, GET get python, GET python put, GET python add item, GET python delete a item, GET Delete order, GET register user, GET New Request.
- Request Details:**
  - Method:** GET
  - URL:** http://127.0.0.1:5000/orders/1
  - Auth Type:** Bearer Token
  - Body:** JSON (empty)
  - Headers:** Authorization (containing a token)
- Response:** 200 OK, 33 ms, 586 B, Save Response, etc.
- Body Content (JSON Response):**

```
1  {
2   "billing_contact_mech_id": 2,
3   "customer_id": 1,
4   "items": [
5     {
6       "order_id": 1,
7       "order_item_seq_id": 2,
8       "product_id": 2,
9       "quantity": 2,
10      "status": "Pending"
11    },
12    {
13      "order_id": 1,
```