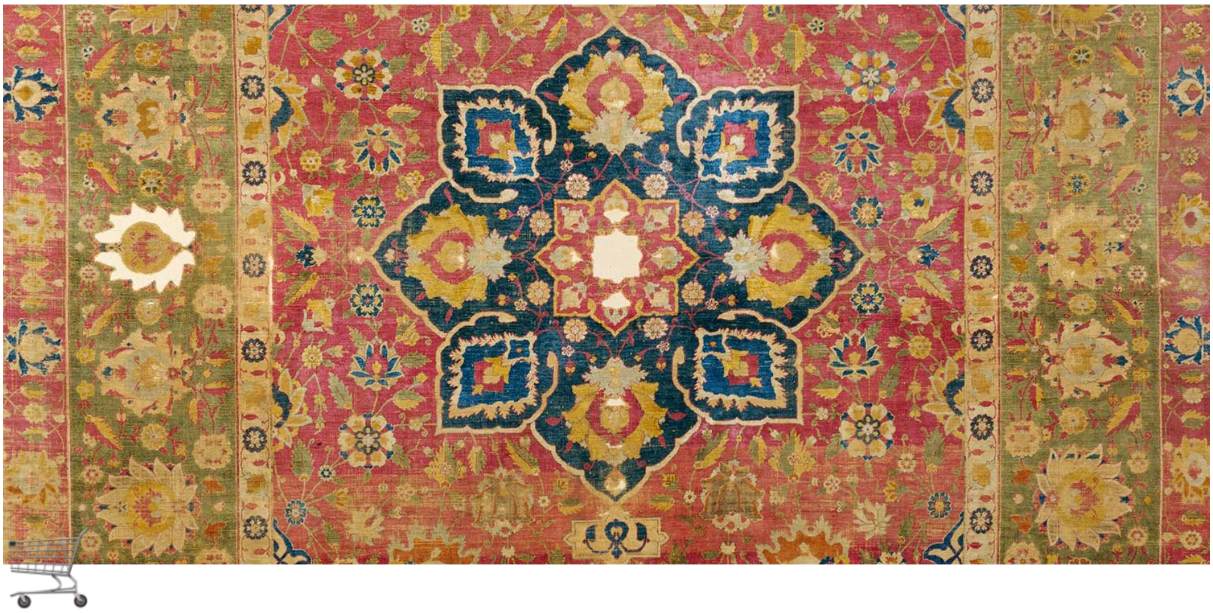

Order Inventory

Order Inventory Project Overview

> Welcome to the documentation for of Order Inventory project! This initiative stems from the need for a streamlined and efficient system to manage and track orders, inventory, and related processes. The primary purpose of the Order Inventory system is to enhance the overall order management experience, providing users with a centralized platform to optimize inventory control and improve order fulfillment processes.

Project Scope

Features and Functionalities:

1. **Order Management:**
    - Creation of new orders.
    - Real-time tracking of order status.
    - Order history and tracking.
2. **Inventory Management:**
    - Comprehensive database of inventory items.
    - Stock level updates and monitoring.
    - Integration with order management for seamless updates.

Database schema:

▼ ER Diagram

https://prod-files-secure.s3.us-west-2.amazonaws.com/31282dbc-a6a3-454c-930f-bedcf45363ec/0df30a00-1748-4940-8c7f-91e12cac4eb4/diag.pdf

▼ MySQL script files

script_one.sql

script_two.sql

Relationships:

1. **Order-Customer Relationship:**
    - Each order is associated with a customer through the CustomerID foreign key in the Orders table.
2. **Order-InventoryItem Relationship:**
    - The OrderDetails table links orders to the items included in each order, using the OrderID and ItemID.

Data Validation Rules:

1. **Order Status Validation:**
    - OrderStatus in the Orders table is restricted to predefined values (e.g., "Pending," "Processing," "Shipped," "Delivered") to maintain consistency.
2. **Stock Quantity Validation:**
    - StockQuantity in the InventoryItems table must be a non-negative integer to ensure accurate representation of available stock.
3. **Customer Email Format Validation:**
    - The Email attribute in the Customers table is validated against a standard email format to ensure the correctness of customer email addresses.
4. **Foreign Key Constraints:**
    - Foreign key constraints are applied to ensure referential integrity. For example, the CustomerID in the Orders table must match a valid CustomerID in the Customers table.

REST API Design:

All REST API requests and responses follow the JSON format. The project employs standard HTTP status codes (200, 201, 400, 404, 401, 500, 505) appropriately in response objects to convey the outcome of API requests

Http Status Code:

HTTP Status Code	Error Response

400	Bad Request
401	Unauthorized
404	Not Found
500	Internal Server Error
505	HTTP Version Not Supported

Working with GitHub:

▼ **Create a New Repository on GitHub:**

- Log in to your GitHub account.
- Click on the "+" icon in the top right corner and select "New repository."
- Provide a name, description, and other relevant details.

▼ **Fork the Repository:**

- Go to the GitHub page of the repository you want to contribute to.
- Click on the "Fork" button in the upper right corner of the page.

▼ **Clone Your Forked Repository:**

Clone your forked repository to your local machine.

```
git clone https://github.com/your-username/repository.git
```

▼ **Make Changes, Commit, and Push:**

- Make the necessary changes to the code.
- Stage and commit your changes.

```
git add .
git commit -m "Your descriptive commit message"
git push
```

▼ **Create a Pull Request:**

- Go to the GitHub page of your forked repository.
- Switch to the branch you just pushed.
- Click on the "New Pull Request" button.

▼ **Review and Merge:**

- Describe your changes and submit the pull request.
- The repository owner or maintainers will review your changes.

▼ **Update Your Fork (Optional):**

Endpoints

Endpoints

Aa End Points	HTTP Methods	Description	Success Response	Error Response
/api/v1/customers	GET	Fetch all customers	Collection of Customers	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching customers."`
/api/v1/customers	POST	Add new Customer object in DB	String: Record Created Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid customer data."`
/api/v1/customers	PUT	Update customer by object	String: Record Updated Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid customer data for updating."`
/api/v1/customers/{customerId)	DELETE	Delete customer by object	String: Record deleted Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid customer ID for deletion."`
/api/v1/customers/{emailId}	GET	Search Customers Matching emailid	Collection of Customers	• HTTP Status Code: 404 Not Found • Body: `"Customer wit the provided email ID not found."`
/api/v1/customers/{name}	GET	Search Customers Matching name wildcard	Collection of Customers	• HTTP Status Code: 404 Not Found • Body: `"Customer wit the provided name wildcard not found."`
/api/v1/customers/shipment/status	GET	Shipment status wise	Custom	• HTTP Status Code: 500

End Points	HTTP Methods	Description	Success Response	Error Response
		count of customers	Object of Shipment Status and count of customers	Internal Server Error • Body: `"An internal server error occurred while fetching shipment status count."`
/api/v1/customers/{custId}/order	GET	Fetch orders for the respective customer by id	Custom object of customer with list of Order(s)	• HTTP Status Code: 404 Not Found • Body: `"Orders for the specified customer ID not found."`
/api/v1/customers/{custId}/shipment	GET	Fetch shipment history for the respective customer by id	Custom object of customer with list of shipment(s)	• HTTP Status Code: 404 Not Found • Body: `"Shipment history for the specified customer ID not found."`
/api/v1/customers/shipments/pending	GET	Retrieve customers with pending shipments	Collection of Customers with pending shipments	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching customers with pending shipments."`
/api/v1/customers/orders/completed	GET	Retrieve customers with completed orders	Collection of Customers with completed orders	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching customers with completed orders."`
/api/v1/customers/shipments/overdue	GET	Retrieve customers with overdue shipments	Collection of Customers with overdue shipments	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching customers with`

Aa End Points	HTTP Methods	Description	Success Response	Error Response
				`overdue shipments."`
/api/v1/customers/orders/quantity/{min}/{max}	GET	Retrieve customers with order quantities within a specified range	Collection of Customers with order quantities in the specified range	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid minimum and maximum quantities for orders."`
/api/v1/products	GET	Fetch all products	Collection of products	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching all products."`
/api/v1/products	POST	Add new product object in DB	String: Record Created Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid product data for creation."`
/api/v1/products	PUT	Update products by object	String: Record Updated Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid product data for updating."`
/api/v1/products /{id}	DELETE	Delete product	String: Record deleted Successfully	• HTTP Status Code: 404 Not Found • Body: `"Product with the specified ID not found for deletion."`
/api/v1/products /{productname}	GET	Search Products Matching product name	Collection of Products	• HTTP Status Code: 404 Not Found • Body: `"Product(s) with the specified name not found."`

End Points	HTTP Methods	Description	Success Response	Error Response
/api/v1/products/brand/{brand}	GET	Retrieve products based on a specific brand	Collection of Products for the specified brand	• HTTP Status Code: 404 Not Found • Body: `"Products wit the specified brand not found."`
/api/v1/products/colour/{colour}	GET	Retrieve products based on a specific colour	Collection of Products for the specified colour	• HTTP Status Code: 404 Not Found • Body: `"Products wit the specified color not found."`
/api/v1/products /unitprice?min=value&max=value	GET	Fetch List of Products matching between min and max unit price	Collection of Products	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid minimum and maximum unit prices."`
/api/v1/products/sort?field=value	GET	Get list of products order by query field	Collection of Product Object ordered by query field	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide a valid field for sorting."`
/api/v1/orders	GET	Fetch all orders	Collection of Orders	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching all orders."`
/api/v1/orders	POST	Create new Order object in DB	String: Record Created Successfully	• HTTP Status Code: 400 Bad Request • Body: `"Invalid request. Please provide valid order data for creation."`
/api/v1/orders	PUT	Update Order by object	String: Record Updated	• HTTP Status Code: 400 Bad Request • Body: `"Invalid`

End Points	HTTP Methods	Description	Success Response	Error Response
			Successfully	`request. Please provide valid order data for updating."`
/api/v1/orders/{id}	DELETE	Delete Order	String: Record deleted Successfully	• HTTP Status Code: 404 Not Found • Body: `"Order with the specified ID not found for deletion."`
/api/v1/orders/status	GET	Count of orders by status	Custom Object for status of order and count	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching the count of orders by status."`
/api/v1/orders/{store}	GET	Fetch order details matching storename	Object of Order having orderid, orderstatus, storename and webaddress	• HTTP Status Code: 404 Not Found • Body: `"Orders with the specified store name not found."`
/api/v1/orders/{id}	GET	Fetch details of a specific order by its ID.	Order details.	• HTTP Status Code: 404 Not Found • Body: `"Order with the specified ID not found."`
/api/v1/orders/customer/{customerId}	GET	Retrieve a list of orders placed by a specific customer.	List of orders.	• HTTP Status Code: 404 Not Found • Body: `"Orders for the specified customer ID not found."`
/api/v1/orders/{id}/cancel	GET	Mark an order as canceled.	Success message.	• HTTP Status Code: 404 Not Found • Body: `"Order with the specified ID not found for cancellation."`

Aa End Points	HTTP Methods	Description	Success Response	Error Response
/api/v1/orders/{orderId}	GET	Retrieve order details by ID	Order details (Single Order)	• HTTP Status Code: 404 Not Found • Body: `"Order with the specified order ID not found."`
/api/v1/orders/status/{status}	GET	Retrieve orders based on a specific status.	Collection of Orders with the specified status	• HTTP Status Code: 404 Not Found • Body: `"Orders with the specified status not found."`
/api/v1/orders/date/{startDate}/{endDate}	GET	Retrieve orders placed within the specified date range	Collection of Orders placed within the specified date range	• HTTP Status Code: 404 Not Found • Body: `"Orders withi the specified date range not found."`
/api/v1/orders/customer/{email}	GET	Retrieve orders associated with a specific customer's email	Collection of Orders associated with the specified customer email	• HTTP Status Code: 404 Not Found • Body: `"Orders for the specified customer email not found."`
Untitled				
/api/v1/inventory	GET	Fetch all inventory records	Custom Object of Inventory records	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching all inventory records."`
/api/v1/inventory?storeid=value	GET	Fetch productdata, storedata, orderstatus matching storied	Custom Object Of inventory contains product, store and order details	• HTTP Status Code: 404 Not Found • Body: `"Inventory records matching the specified store ID not found."`
/api/v1/inventory/shipment	GET	Fetch inventories	Collection of Inventory	• HTTP Status Code: 500 Internal Server

End Points	HTTP Methods	Description	Success Response	Error Response
		and matching shipments	records matching shipments	Error • Body: `"An internal server error occurred while fetching inventory records matching shipments."`
/api/v1/inventory/{orderid}	GET	Fetch store, product and customer data matching order id	Custom Object to return product, customer and store details matching ordered	• HTTP Status Code: 404 Not Found • Body: `"Store, product, and customer data for the specified order ID not found."`
/api/v1/inventory/shipment	GET	Count shipment status wise count of total products sold	Custom Object of Inventory to print shipment status and count of sold products	• HTTP Status Code: 500 Internal Server Error • Body: `"An internal server error occurred while fetching the count of shipment status and sold products."`
/api/v1/inventory/{orderid}/details	GET	Fetch list of Products in a order matching ordered with store details, shipment status and total amount	Custom List of object for inventory details	Body: `"List of products in the specified order ID not found with store details, shipment status, and total amount."`
/api/v1/inventory/product/{productId}/store/{storeId}	GET	Retrieve inventory records for a specific product and store	Custom Object of Inventory records	• HTTP Status Code: 404 Not Found • Body: `"Inventory records for the specified product and store not found."`
/api/v1/inventory/category/{category}	GET	Retrieve	Custom	• HTTP Status Code: 404 Not

Aa End Points	HTTP Methods	Description	Success Response	Error Response
		inventory records for products in a specific category	Object of Inventory records for the specified category	Found • Body: `"Inventory records for the specified category not found."`