

PROJECT OVERVIEW

This project is an e-commerce web application. It provides a platform(marketplace) for vendors to sell multiple products and for users(buyers) to purchase from a variety of products.

Vendors can track all of their sales based on their different products. From this, they can decipher which product has a better market performance.

FEATURES

Product Management:

- Add a new product
- Retrieve product details
- Update product details
- Delete a product

Order Management:

- Place an order
- Retrieve order details
- Update order status
- Cancel an order

API Endpoints

➤ User Management:

Create a new user:

HTTP Verb: POST

Endpoint URL: /api/users/

Details: Accepts a JSON payload with user details.

Retrieve user details:

HTTP Verb: GET

Endpoint URL: /api/users/{user_id}/

Details: Requires the user ID as a parameter.

Delete a user:

HTTP Verb: DELETE

Endpoint URL: /api/users/{user_id}/

Details: Requires the user ID as a parameter.

➤ Vendor Management:

Create a new vendor:

HTTP Verb: POST

Endpoint URL: /api/vendors/

Details: Accepts a JSON payload with vendor details.

Retrieve vendor details:

HTTP Verb: GET

Endpoint URL: /api/vendors/{vendor_id}/

Details: Requires the vendor ID as a parameter.

Delete a vendor:

HTTP Verb: DELETE

Endpoint URL: /api/vendors/{vendor_id}/

Details: Requires the vendor ID as a parameter.

➤ **Product Management:**

Add a new product:

HTTP Verb: POST

Endpoint URL: /api/products/

Details: Accepts a JSON payload with product details.

Retrieve product details:

HTTP Verb: GET

Endpoint URL: /api/products/{product_id}/

Details: Requires the product ID as a parameter.

Update product details:

HTTP Verb: PUT

Endpoint URL: /api/products/{product_id}/

Details: Requires the product ID as a parameter. Accepts a JSON payload with updated product details.

Delete a product:

HTTP Verb: DELETE

Endpoint URL: /api/products/{product_id}/

Details: Requires the product ID as a parameter.

➤ **Order Management:**

Place a new order:

HTTP Verb: POST

Endpoint URL: /api/orders/

Details: Accepts a JSON payload with order details.

Retrieve order details:

HTTP Verb: GET

Endpoint URL: /api/orders/{order_id}/

Details: Requires the order ID as a parameter.

Update order status:

HTTP Verb: PUT

Endpoint URL: /api/orders/{order_id}/

Details: Requires the order ID as a parameter. Accepts a JSON payload with updated order status.

Cancel an order:

HTTP Verb: DELETE

Endpoint URL: /api/orders/{order_id}/

Details: Requires the order ID as a parameter.

➤ **Product Order Management:**

Add a product to an order:

HTTP Verb: POST

Endpoint URL: /api/product_orders/

Details: Accepts a JSON payload with order item details.

Retrieve product order details:

HTTP Verb: GET

Endpoint URL: /api/product_orders/{order_item_id}/

Details: Requires the order item ID as a parameter.

Cancel a product order:

HTTP Verb: DELETE

Endpoint URL: /api/product_orders/{order_item_id}/

Details: Requires the order item ID as a parameter.

DATABASE TABLES

Database Name: e-commerce

USERS***Columns:***

user_id (Primary Key, Integer)

username (String)

email (String, Unique)

password (String)

VENDORS***Columns:***

vendor_id (Primary Key, Integer)

name (String)

email (String, Unique)

PRODUCTS

Columns:

product_id (Primary Key, Integer)
vendor_id (Foreign Key to vendors)
category (String)
price (Decimal)
description (Text)

ORDERS

Columns:

order_id (Primary Key, Integer)
user_id (Foreign Key to users)
amount (Decimal)
date_time (Timestamp)

PRODUCT_ORDER

Columns:

order_id (Primary Key to orders)
product_id (Primary Key to products)
amount (Decimal)
date_time (Timestamp)

E-commerce

This is an entity relational diagram for an e-commerce store which has multiple vendors that sell different products registered under them.

