

# PhiMart Planning

Here's a comprehensive list of REST API endpoints for your e-commerce project. These endpoints are designed to cover the core functionalities for products, categories, orders, and user management in the e-commerce platform.

## 1. Authentication Endpoints

These will handle user registration, login, and token management.

HTTP Method	Endpoint	Description
POST	/auth/register/	Register a new user.
POST	/auth/login/	Log in and obtain a token.
POST	/auth/logout/	Log out the current user.
POST	/auth/token/refresh/	Refresh an expired JWT token.

## 2. Category Endpoints

Manage and retrieve product categories.

HTTP Method	Endpoint	Description
GET	/api/categories/	List all categories.
GET	/api/categories/<id>/	Retrieve a specific category.
POST	/api/categories/	Create a new category. (Admin)
PUT	/api/categories/<id>/	Update an existing category.
DELETE	/api/categories/<id>/	Delete a category.

## 3. Product Endpoints

Manage and retrieve products.

HTTP Method	Endpoint	Description
GET	/api/products/	List all products.
GET	/api/products/<id>/	Retrieve a specific product.
GET	/api/products/?search=	Search products by name or description.
POST	/api/products/	Create a new product. (Admin)
PUT	/api/products/<id>/	Update an existing product. (Admin)
DELETE	/api/products/<id>/	Delete a product. (Admin)
GET	/api/products/?category=	Filter products by category.

## 4. Shopping Cart Endpoints

Manage a user's shopping cart.

HTTP Method	Endpoint	Description
GET	/api/cart/	Retrieve the user's current cart.
POST	/api/cart/add/	Add an item to the cart.
PUT	/api/cart/update/<item_id>/	Update the quantity of a cart item.
DELETE	/api/cart/remove/<item_id>/	Remove an item from the cart.
DELETE	/api/cart/clear/	Clear all items from the cart.

## 5. Order Endpoints

Handle order placement and management.

HTTP Method	Endpoint	Description
GET	/api/orders/	List all orders for the current user.
GET	/api/orders/<id>/	Retrieve details of a specific order.

POST	/api/orders/	Place a new order based on the current cart.
PUT	/api/orders/<id>/status/	Update the status of an order (Admin).
DELETE	/api/orders/<id>/	Cancel an order.

## 6. User Profile Endpoints

Allow users to manage their profile.

HTTP Method	Endpoint	Description
GET	/api/profile/	Retrieve the logged-in user's profile.
PUT	/api/profile/	Update the user's profile.

## 7. Dashboard Endpoints (Admin)

Provide administrative data for managing the platform.

HTTP Method	Endpoint	Description
GET	/api/dashboard/total-users/	Get the total number of registered users.
GET	/api/dashboard/total-orders/	Get the total number of orders placed.
GET	/api/dashboard/total-products/	Get the total number of available products.

## Models

Here's a **detailed plan** for each model, including the field names, data types, and relationships. This covers all the entities and their interactions for your **PhitMart** e-commerce project:

### 1. User Model

Represents customers and platform users.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the user.
first_name	CharField	Required FirstName for user
last_name	CharField	Required LastName for user
email	EmailField	Unique email for the user.
address	TextField	Optional address of the user.
phone_number	CharField	Optional phone number.
password	CharField (hashed)	Hashed password for authentication.

### 2. Category Model

Groups products into categories.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the category.
name	CharField	Name of the category.
description	TextField	Optional description.

Relationships:

- Category to Product:** One-to-Many (One category has multiple products).

### 3. Product Model

Contains details about products available for purchase.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the product.
name	CharField	Name of the product.
description	TextField	Detailed product description.

price	DecimalField	Price of the product.
stock	IntegerField	Available stock quantity.
image	ImageField	Optional image of the product.
category	ForeignKey	Links to the Category model.
created_at	DateTimeField	Timestamp when the product was created.
updated_at	DateTimeField	Timestamp when the product was last updated.

Relationships:

- Product to Category: Many-to-One.

## 4. Cart Model

Represents the shopping cart of a user.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the cart.
user	OneToOneField	Links to the User model (One cart per user).
created_at	DateTimeField	Timestamp when the cart was created.

Relationships:

- Cart to User: One-to-One.

## 5. CartItem Model

Represents items added to a cart.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the cart item.
cart	ForeignKey	Links to the Cart model.
product	ForeignKey	Links to the Product model.
quantity	IntegerField	Quantity of the product.

Relationships:

- CartItem to Cart: Many-to-One.
- CartItem to Product: Many-to-One.

## 6. Order Model

Represents completed purchases.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the order.
user	ForeignKey	Links to the User model.
status	CharField	Current status of the order (Pending, Shipped, Delivered).
total_price	DecimalField	Total price of the order.
created_at	DateTimeField	Timestamp when the order was created.
updated_at	DateTimeField	Timestamp when the order was last updated.

Relationships:

- Order to User: Many-to-One.

## 7. OrderItem Model

Contains details about items in an order.

Field Name	Data Type	Description
id	Primary Key	Unique identifier for the order item.
order	ForeignKey	Links to the Order model.

product	ForeignKey	Links to the Product model.
quantity	IntegerField	Quantity of the product in the order.
price	DecimalField	Price of the product at the time of purchase.

Relationships:

- **OrderItem** to **Order**: Many-to-One.
- **OrderItem** to **Product**: Many-to-One.

---

Relationships Summary

1. **User**:
  - One-to-One with **Cart**.
  - Many-to-One with **Order**.
2. **Category**:
  - One-to-Many with **Product**.
3. **Product**:
  - Many-to-One with **Category**.
  - Many-to-One with **CartItem** and **OrderItem**.
4. **Cart**:
  - One-to-Many with **CartItem**.
  - One-to-One with **User**.
5. **Order**:
  - Many-to-One with **User**.
  - One-to-Many with **OrderItem**.