API Endpoints

1. Home Page API

- **Endpoint:** `/` (GET)
- **Description:** Fetch new arrivals and featured clothes.
- **Response:** JSON array of products categorized as "new arrivals" or "featured".
- **Query Parameters:** `?limit=10` for paginated responses.
- **Example Response:**

```
{
    "newArrivals": [{ "id": 1, "name": "Dress A", "price": 50 }],
    "featured": [{ "id": 2, "name": "Jacket B", "price": 80 }]
}
```

2. Products API

- **Endpoint:** `/products` (GET)
- **Description:** Fetch products by category and apply filters.
- **Request Parameters:**
  - `?category=women` → Filter by category.
  - `?priceRange=100-500` → Filter by price range.
  - `?sort=price` → Sort by price or relevance.
- **Response:** JSON array of products matching filters.
- **Example Response:**

```
[
    { "id": 1, "name": "T-shirt", "price": 20, "category": "women" },
    { "id": 2, "name": "Skirt", "price": 35, "category": "women" }
]
```

3. Single Product API

- **Endpoint:** `/product/:id` (GET)
- **Description:** Fetch specific product details.
- **Request Parameters:** `:id` → Product ID (e.g., `/product/123`).
- **Response:** JSON object with detailed product info.
- **Example Response:**

```
{
    "id": 123,
    "name": "Winter Coat",
    "description": "Warm and stylish coat.",
    "price": 150,
    "stock": 25,
    "images": ["image1.jpg", "image2.jpg"]
}
```

3.1. Cart Management (State Management)

- **Description:** Manage user's shopping cart using state management (Zustand).
- **Actions:**
  - **Add to Cart:** Add a product to the user's cart.
  - **Remove from Cart:** Remove a product from the user's cart.
  - **Fetch Cart Items:** Retrieve all products in the user's cart.
- **State Structure:**

```
{
    "cart": [
        { "productId": 101, "name": "T-shirt", "price": 20, "quantity": 2 },
        { "productId": 202, "name": "Skirt", "price": 35, "quantity": 1 }
    ]
}
```

- **Example Actions:**
  - **Add to Cart:**

```
addToCart({ productId: 101, name: "T-shirt", price: 20, quantity: 2 });
```

        - **Remove from Cart:**

          ```
          removeFromCart(101);
          ```

        - **Fetch Cart Items:**

          ```
          const cartItems = getCartItems();
          ```

    - **Example State Update:**
        - **Add to Cart:**

          ```
          {
              "message": "Product added to cart successfully."
          }
          ```

        - **Remove from Cart:**

          ```
          {
              "message": "Product removed from cart successfully."
          }
          ```

4. Orders API

- **Endpoint:** `/orders` (POST)
- **Description:** Save the order and process payment.
- **Request Body:**

  ```
  {
      "user": { "id": 1, "name": "John Doe", "email": "john@example.com" },
      "products": [
          { "id": 101, "quantity": 2 },
          { "id": 202, "quantity": 1 }
      ],
      "paymentInfo": { "method": "credit_card", "status": "paid" }
  }
  ```

- **Response:** Confirm order status and payment.
- **Example Response:**

  ```
  {
      "orderId": "ORD12345",
      "status": "success",
      "message": "Order placed successfully."
  }
  ```

5. Shipment API

- **Endpoint:** `/shipment` (GET)
- **Description:** Fetch all orders and their payment status.
- **Request Parameters:** : `?userId=1` → Fetch shipments for a specific user.
- **Response:** JSON array of orders with shipment and payment status.
- **Example Response:**

  ```
  [
      {
          "orderId": "ORD12345",
          "shipmentStatus": "shipped",
          "paymentStatus": "paid",
          "trackingNumber": "TRK123"
      },
      {
          "orderId": "ORD67890",
          "shipmentStatus": "pending",
          "paymentStatus": "pending",
          "trackingNumber": null
      }
  ]
  ```