## API Documentation (MVC Architecture)

**BaseUrl**: <https://wsa-mini.onrender.com>

1. `GET /products`

Retrieve all products, with optional query parameters for filtering by category, region, and price range.

**Request**

- **Method**: GET

- **Endpoint**: `/products`

- **Query Parameters**:

- `category\_id` (optional): Integer - Filter by category ID.

- `region\_id` (optional): Integer - Filter by region ID.

- `min\_price` (optional): Decimal - Minimum price for filtering.

- `max\_price` (optional): Decimal - Maximum price for filtering.

**Response**

- 200 OK

- Renders an EJS template `products.ejs` with the list of products.

- Example EJS rendering:

**```javascript**

**res.render('products', { products: products });**

**```**

## 2. `GET /products/{id}`

Retrieve details of a specific product.

**Request**

- **Method**: GET

- **Endpoint**: `/products/{id}`

- **Path Parameters**:

- `id`: Integer - The ID of the product to retrieve.

**Response**

- **200 OK**

- Renders an EJS template `product.ejs` with the product details.

- Example EJS rendering:

**```javascript**

**res.render('product', { product: product });**

**```**

- **404 Not Found**

- Renders an EJS template `error.ejs` with an error message.

- Example EJS rendering:

**```javascript**

**res.status(404).render('error', { message: 'Product not found' });**

**```**

## 3. `POST /cart`

Add a product to the cart.

**Request**

- **Method**: POST

- **Endpoint**: `/cart`

- **Body Parameters:**

- `product\_id`: Integer - The ID of the product to add.

- `quantity`: Integer - The quantity of the product to add.

**Response**

- **201 Created**

- Redirects to the cart page.

- Example redirection:

**```javascript**

**res.redirect('/cart');**

**```**

- **400 Bad Request**

- Renders an EJS template `error.ejs` with an error message.

- Example EJS rendering:

**```javascript**

**res.status(400).render('error', { message: 'Invalid input' });**

**```**

## 4. `GET /cart`

Retrieve the contents of the cart.

**Request**

- **Method**: GET

- **Endpoint**: `/cart`

**Response**

- **200 OK**

- Renders an EJS template `cart.ejs` with the cart contents.

- Example EJS rendering:

**```javascript**

**res.render('cart', { cart: cartItems });**

**```**

## 5. `PUT /cart/{id}`

Update the quantity of a product in the cart.

**Request**

- **Method**: PUT

- **Endpoint**: `/cart/{id}`

- **Path Parameters**:

- `id`: Integer - The ID of the cart item to update.

- **Body Parameters**:

- `quantity`: Integer - The new quantity of the product.

**Response**

- **200 OK**

- Redirects to the cart page.

- Example redirection:

**```javascript**

**res.redirect('/cart');**

**```**

- **400 Bad Request**

- Renders an EJS template `error.ejs` with an error message.

- Example EJS rendering:

**```javascript**

**res.status(400).render('error', { message: 'Invalid input' });**

**```**

- 404 Not Found

- Renders an EJS template `error.ejs` with an error message.

- Example EJS rendering:

**```javascript**

**res.status(404).render('error', { message: 'Cart item not found' });**

```

## 6. `DELETE /cart/{id}`

Remove a product from the cart.

**Request**

- **Method**: DELETE

- **Endpoint**: `/cart/{id}`

- **Path Parameters**:

- `id`: Integer - The ID of the cart item to remove.

**Response**

- **200 OK**

- Redirects to the cart page.

- Example redirection:

**```javascript**

**res.redirect('/cart');**

**```**

- **404 Not Found**

- Renders an EJS template `error.ejs` with an error message.

- Example EJS rendering:

**```javascript**

**res.status(404).render('error', { message: 'Cart item not found' });**

**```**

## 7. `GET /stock/{id}`

Check the stock quantity of a specific product.

**Request**

- **Method**: GET

- **Endpoint**: `/stock/{id}`

- **Path Parameters**:

- `id`: Integer - The ID of the product to check stock for.

**Response**

- **200 OK**

**```json**

**{**

**"product\_id": 1,**

**"stock\_quantity": 50**

**}**

**```**

- **404 Not Found**

**```json**

**{**

**"error": "Product not found"**

**}**

**```**