Contents

[**User Authentication** 2](#_Toc153304563)

[**Register User** 2](#_Toc153304564)

[**Login User** 2](#_Toc153304565)

[**User Profile** 2](#_Toc153304566)

[**Product Catalog** 3](#_Toc153304567)

[**Manage Category** 3](#_Toc153304568)

[**Manage Brand** 4](#_Toc153304569)

[**Manage Product** 4](#_Toc153304570)

[**Get Suggestions** 5](#_Toc153304571)

[**Shopping Cart** 5](#_Toc153304572)

[**Manage Cart** 5](#_Toc153304573)

[**Checkout** 5](#_Toc153304574)

[**Orders** 6](#_Toc153304575)

[**Reviews** 6](#_Toc153304576)

[**Add Review** 6](#_Toc153304577)

[**Get Reviews** 6](#_Toc153304578)

[**Search History** 7](#_Toc153304579)

[**Get Search History** 7](#_Toc153304580)

[**Advanced Search and Filtering** 7](#_Toc153304581)

# **User Authentication**

To register, you need to provide the necessary details. Subsequently, you can log in by entering your username and password, obtaining a token that will be used for authentication purposes.

## **Register User**

Endpoint: /register/

Method: POST

Header:

Description: Register a new user.

Request:

{

"username": "example\_user",

"email": "user@example.com",

"password": "secure\_password",

"profile": {

"phone": "1234567890", // your mobile number

"address": "your address"

},

"role": "your\_role" // admin, staff, customers

}

## **Login User**

Endpoint: /login/

Method: POST

Header:

Description: Log in an existing user.

Request:

{

"username": "example\_user",

"password": "secure\_password"

}

# **User Profile**

Endpoint: /profile/

Method: GET (Get user profile) | PUT (Update user profile)

Header: key = Authorization, value = Token your\_token

Description: Get or update the user's profile.

Request (GET): // no need to add the any data just pass the token to headers

Request (PUT):

{

"phone": "9876543210", // your updated mobile number

"address": " your updated address"

}

# **Product Catalog**

You can execute CRUD operations for categories, brands, and products. It's possible to add numerous categories and brands directly to the database. When adding a product, valid category and brand IDs must be provided, indicating that a category and brand must exist in the database before adding a product.

## **Manage Category**

Endpoint: http://product-catalog/categories/

Method: GET (Get all categories) | POST (Create a new category) | PUT (Update a category) | DELETE (Delete a category)

Header: key = Authorization, value = Token admin\_or\_staff\_token

Description: Get all categories, create a new category, update, or delete an existing category.

Request (POST):

{

"name": "your\_category"

}

Request (GET): // no need to add the any data.

Request (PUT):

{

"id": 1, // id of the category

"name": "your\_updated\_category"

}

Request (DELETE):

{

"id": 1, // id of the category

}

## **Manage Brand**

Endpoint: /product-catalog/brands/

All other functionalities mirror those of the category.

## **Manage Product**

Endpoint: /product-catalog/products/

Method: GET (Get all products) | POST (Create a new product) | PUT (Update a product) | DELETE (Delete a product)

Header: key = Authorization, value = Token admin\_or\_staff\_token

Description: Get all products, create a new product, update, or delete an existing product.

Request (POST):

{

"name": "product\_name",

"description": "some\_discription\_of\_product",

"category\_id": 1, // id should be in the database

"brand\_id": 1, // id should be in the database

"price": 799.99,

"inventory": 50,

"attributes": [

{

"name": " attribute\_name",

"value": " attribute\_value"

}

// you can add as many attributes as you want in this list

]

}

Request (GET): // no need to add the any data.

Request (PUT): // same as POST just add the id of the product.

Request (DELETE):

{

"id": 1, // id of the product

}

# **Get Suggestions**

You can retrieve product suggestions, encompassing items from your historical searches and previous orders.

Endpoint: /suggestions/

Method: GET

Header: key = Authorization, value = Token customer\_token

Description: Get product suggestions based on user search history and previous orders.

Request (GET): // no need to add the any data.

# **Shopping Cart**

You can add products to your shopping cart, maintaining the cart's current state for potential modifications. During the checkout process, you need to supply the transaction ID for the payment and the chosen payment method. Subsequently, an order will be generated for all items in the cart, saving it for future reference. Furthermore, the inventory of purchased products will be reduced.

## **Manage Cart**

Endpoint: /cart/

Method: GET (Get user's shopping cart) | POST (Add product to cart) | DELETE (Delete user's shopping cart)

Header: key = Authorization, value = Token customer\_token

Description: Get the user's shopping cart, add a product to the cart, or delete the entire cart.

Request (POST):

{

"product\_id": 1

}

Request (GET): // no need to add the any data.

Request (DELETE): // no need to add the any data. It will get the user from request and will delete that user’s cart.

## **Checkout**

Endpoint: /checkout/

Method: POST

Header: key = Authorization, value = Token customer\_token

Description: Checkout and place an order.

Request (POST):

{

"transaction\_id": "txn\_id",

"payment\_method": "payment\_method"

}

# **Orders**

You can check your order history any time.

Endpoint: /orders/

Method: GET

Header: key = Authorization, value = Token customer\_token

Description: Get a list of orders placed by the user.

Request (DELETE): // no need to add the any data. It will get the user from request and will return that user’s order.

# **Reviews**

Customers have the option to add reviews and ratings for their orders after completing the checkout process. These reviews are then accessible to all other users on the platform, providing transparency and insight into the experiences shared by customers.

**Add Review**

Endpoint: /add\_review/

Method: POST

Header: key = Authorization, value = Token customer\_token

Description: Add a review for a specific order.

Request (POST):

{

"order\_id": 1,

"rating": 5,

"review": "Great product, fast delivery!"

}

**Get Reviews**

Endpoint: /reviews/

Method: GET

Header: key = Authorization, value = Token your\_token

Description:

Get a list of all reviews.

Request (GET): // no need add any data.

**Search History**

**Get Search History**

Endpoint: /search\_history/

Method: GET

Description: Get a list of search keywords used by the user.

Request (GET): // no need add any data.

# **Advanced Search and Filtering**

Users can apply advanced search filters to refine their product searches. Parameters can be included in the request to filter products, but it's important to note that all parameters are optional. If no parameters are provided, the system will return all available products.

Endpoint: /product-catalog/products/

Method: GET

Header: key = Authorization, value = Token your\_token

Parameters:

keyword: some\_key\_word

category\_id: some\_category\_id

brand\_id: some\_brand\_id

min\_price: some\_value

max\_price: some\_vlaue

Description:

Get a list of all reviews.

Request (GET): // no need add any data.