

## \*MongoDB Response Structure :

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: localhost:27017 > productCatalogueSystem > products. The main area shows the 'Documents' tab for the 'products' collection. Two documents are visible:

```
{
  "_id": "2",
  "name": "Smartphones",
  "description": "Product Description",
  "price": 29.99,
  "categories": Array (2),
  "attributes": Array (3),
  "availability": Object,
  "ratings": Array (2),
  "_class": "com.LeadToRev.productCatalogueSystem.model.Product"
}
```

```
{
  "_id": "18",
  "name": "Lava",
  "description": "Product Description",
  "price": 49.99,
  "categories": Array (2),
  "attributes": Array (3),
  "availability": Object,
  "ratings": Array (2),
  "_class": "com.LeadToRev.productCatalogueSystem.model.Product"
}
```

## \*Add Product :

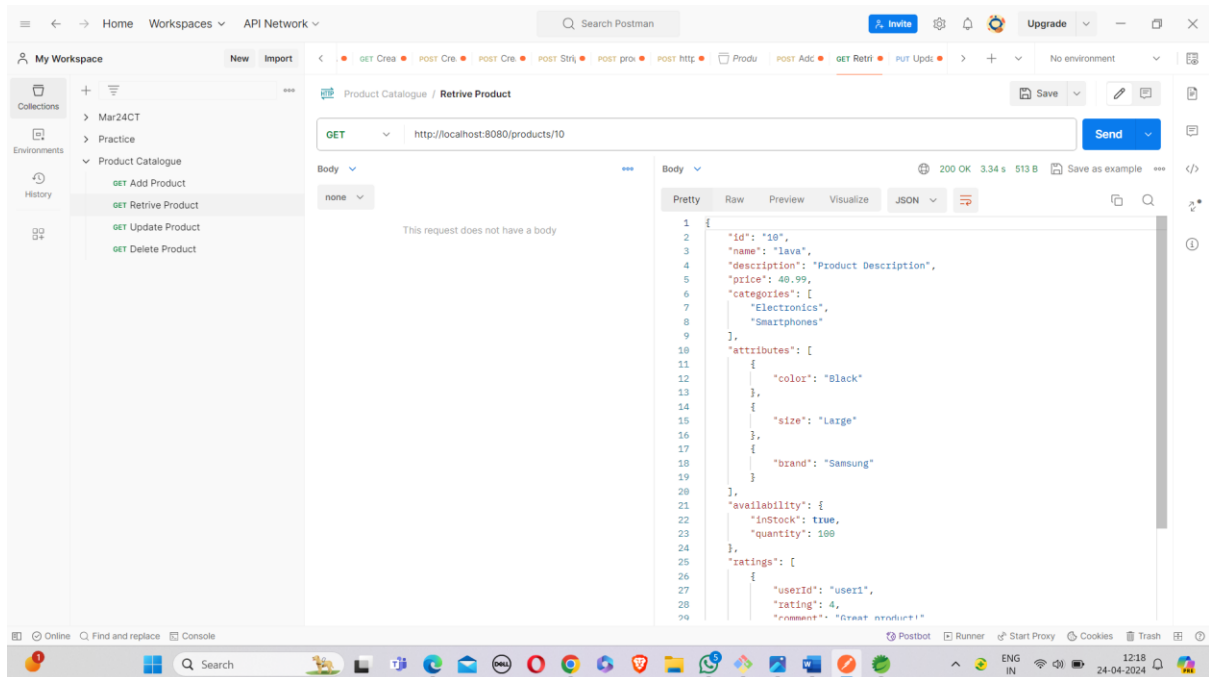
The screenshot shows the Postman interface. A POST request is configured to http://localhost:8080/products. The request body is a JSON object representing a product:

```
{
  "id": "20",
  "name": "Mobile",
  "description": "Product Description",
  "price": 29.99,
  "categories": ["Laptop", "Smartphones"],
  "attributes": [
    { "color": "Black", "size": "Large", "brand": "Samsung" }
  ],
  "availability": {
    "inStock": true,
    "quantity": 100
  },
  "ratings": [
    { "userId": "user1", "rating": 4, "comment": "Great product!" },
    { "userId": "user2", "rating": 5, "comment": null }
  ]
}
```

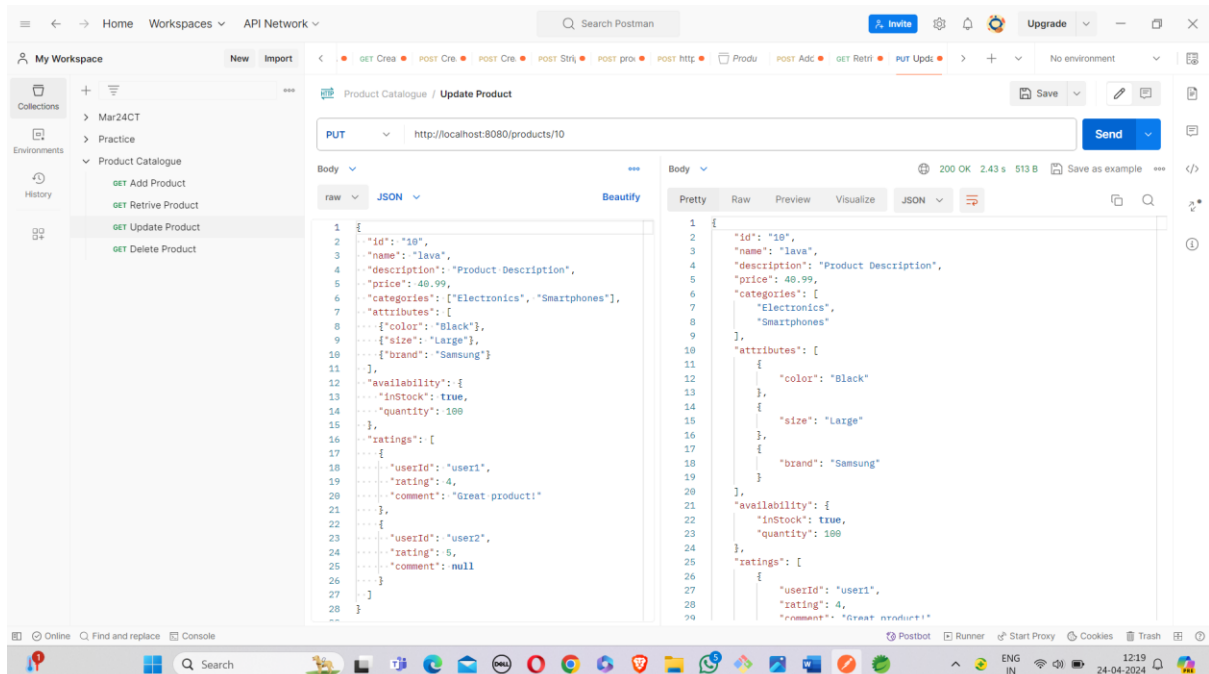
The response body is also shown in a pretty-printed format:

```
{
  "attributes": [
    {
      "color": "Black",
      "size": "Large",
      "brand": "Samsung"
    }
  ],
  "availability": {
    "inStock": true,
    "quantity": 100
  },
  "ratings": [
    {
      "userId": "user1",
      "rating": 4,
      "comment": "Great product!"
    },
    {
      "userId": "user2",
      "rating": 5,
      "comment": null
    }
  ]
}
```

## \*Retrieve Product :



## \*Update Product :



## \*Delete Product :

The screenshot shows the Postman interface with a DELETE request configured. The request is sent to the URL `http://localhost:8080/products/1`. The response is a 200 OK status with a response time of 1494 ms and a body of `Product is deleted....`.

**Request Details:**

- Method: DELETE
- URL: `http://localhost:8080/products/1`
- Params: None
- Body: None

**Response Details:**

- Status: 200 OK
- Time: 1494 ms
- Size: 186 B
- Body: `Product is deleted....`

The left sidebar shows the 'Product Catalogue' collection with the following items:

- GET Add Product
- GET Retrieve Product
- GET Update Product
- GET Delete Product