1st January 2018

IT Academy – UI/UX Training

AngularJS Assignment – Ecommerce Application

Create an e-commerce application using AngularJS, Bootstrap 4, HTML5 and CSS3.

e.g. amazon.com, aliexpress.com etc.

The application must have the following features:

- Featured products on the home page
- List of categories on the home page
- Category Page with category details and product list for that category
- Product Detail page with products details
- Ability to add product to cart from any page
- Cart icon with count on the header section
- Ability to set quantity on product detail page
- Ability to update quantity of products in cart
- Cart page with ability to increase quantity of products
- Ability to checkout from cart page and add products to orders
- Orders page with the past order details

Please Note:

- The app must have AngularJS ideal folder structure
- Should include multi-module structure
- Must use as many AngularJS concepts as possible
- Must use routes, factories, services, directives, filters etc.
- Avoid use of third party libraries
- Should be responsive UI with good user experience
- Should include comments in code

API

1. **Fetch all products**

   Method: GET
   URL: http://**{host}**/ecommerceapp/products

2. **Fetch a single product**

   Method: GET
   URL: http://**{host}**/ecommerceapp/products/**{id}**

3. **Add new product**

   Method: POST
   URL: http://**{host}**/ecommerceapp/products
   Request Type: JSON
   Request Data Sample (Add more properties if required):

   ```
   {
           "name": "here goes a product name",
           "description": "here goes a product description",
           "image": "path/to/file",
           "created_date": "datetime",
           "updated_date": "datetime"
   }
   ```

4. **Update existing product**

   Method: PATCH
   URL: http://**{host}**/ecommerceapp/products/**{id}**/update
   Request Type: JSON
   Request Data Sample:

   ```
   {
           "name": "here goes the updated product name",
           "updated_date": "datetime"
   }
   ```

5. **Delete existing product**

   Method: DELETE
   URL: http://**{host}**/ecommerceapp/products/**{id}**/delete

6. **Fetch all categories**

Method: GET
URL: http://**{host}**/ecommerceapp/categories

7. **Fetch a single category**

Method: GET
URL: http://**{host}**/ecommerceapp/categories/**{id}**

8. **Add new category**

Method: POST
URL: http://**{host}**/ecommerceapp/categories
Request Type: JSON
Request Data Sample (Add more properties if required):

```
{
        "name": "here goes a product name",
        "description": "here goes a product description",
        "parent_id": "here goes a product description",
        "image": "path/to/file",
        "created_date": "datetime",
        "updated_date": "datetime"
},
```

9. **Update existing category**

Method: PATCH
URL: http://**{host}**/ecommerceapp/categories/**{id}**/update
Request Type: JSON
Request Data Sample:

```
{
        "name": "here goes the updated category name",
        "updated_date": "datetime"
}
```

10. **Delete existing categories**

Method: DELETE
URL: http://**{host}**/ecommerceapp/categories/**{id}**/delete

11. **Fetch all cart items**

    Method: GET
    URL: http://**{host}**/ecommerceapp/cart

12. **Add product to cart**

    Method: POST
    URL: http://**{host}**/ecommerceapp/cart
    Request Type: JSON
    Request Data Sample (Add more properties if required):

```
        {
                "product_id": "",
                "quantity": "",
                "created_date": "datetime",
                "updated_date": "datetime"
        },
```

13. **Update product in cart**

    Method: PATCH
    URL: http://**{host}**/ecommerceapp/cart/**{id}**/update
    Request Type: JSON
    Request Data Sample:

```
        {
                "quantity": "",
                "updated_date": "datetime"
        }
```

14. **Delete product in cart**

    Method: DELETE
    URL: http://**{host}**/ecommerceapp/cart/**{id}**/delete

15. **Fetch order**

    Method: GET
    URL: http://**{host}**/ecommerceapp/order/**{id}**

16. **Add new order**

    Method: POST
    URL: http://**{host}**/ecommerceapp/order
    Request Type: JSON
    Request Data Sample (Add more properties if required):

```
{
        "items": [],
        "created_date": "datetime",
        "updated_date": "datetime"
}
```

Setting up the Mock Server

Requirements:
Nodejs, Git

Steps:

1. Run ‘npm install’ in the server folder to install the server dependencies

2. Run ‘npm start’ in the server folder to start the mock server

3. Test the REST API using postman or any HTTP REST Client