Functional Testing

Functional testing ensures that each feature of the website operates according to its intended functionality. Below are the steps on how to perform functional testing for each feature of the **Marketplace Ecommerce Website**:

Test Cases for Each Feature:

1. Data Fetching & Dynamic Routes:

Test Case 1: Verify Product Data Fetching

- **Test Objective**: Ensure product data is fetched correctly from the backend.
- **Test Steps**:
    1. Send a GET request to fetch product data via the API.
    2. Verify the product name, description, price, and image in the response.
    3. Ensure that products are displayed correctly on the frontend (product listing page).
- **Expected Result**: The product data should be returned and displayed as intended on the page.

Test Case 2: Verify Dynamic Product Page Routing

- **Test Objective**: Ensure dynamic routing to individual product detail pages works.
- **Test Steps**:
    1. Click on a product on the product listing page.
    2. Verify that the correct product details are displayed on the individual product page.
- **Expected Result**: Clicking on a product should route to the correct product details page, displaying accurate information.

2. Add to Cart Functionality:

Test Case 1: Add Product to Cart

- **Test Objective**: Ensure that clicking "Add to Cart" adds a product to the cart.
- **Test Steps**:
    1. Click "Add to Cart" on a product listing page.
    2. Verify that the product appears in the cart (use SweetAlert2 confirmation for feedback).

- **Expected Result**: A SweetAlert2 pop-up should appear confirming the addition of the product to the cart.

Test Case 2: Cart Persistence

- **Test Objective**: Ensure that cart data is persistent across page refreshes.
- **Test Steps**:
    1. Add a product to the cart.
    2. Refresh the page and check if the product remains in the cart.
- **Expected Result**: The product should remain in the cart even after the page is refreshed.

3. Gift Mechanism with Voucher System:

Test Case 1: Apply Valid Voucher Code

- **Test Objective**: Ensure a valid voucher code applies the gift mechanism during checkout.
- **Test Steps**:
    1. Enter a valid voucher code during checkout.
    2. Verify that the discount or gift mechanism is applied.
- **Expected Result**: The gift mechanism (discount or offer) should be successfully applied to the checkout amount.

Test Case 2: Apply Invalid Voucher Code

- **Test Objective**: Ensure an invalid voucher code returns an error message.
- **Test Steps**:
    1. Enter an invalid voucher code during checkout.
    2. Verify that an error message is displayed, indicating the voucher code is invalid.
- **Expected Result**: An error message should be shown indicating that the voucher code is invalid.

4. CRUD Operations on Products:

Test Case 1: Add Product (POST request)

- **Test Objective**: Ensure that a product can be added successfully via the admin panel.
- **Test Steps**:

1. Send a POST request with valid product data.
2. Verify that the new product appears in the product list on the frontend.

- **Expected Result**: The new product should be added to the backend and appear in the product listing.

Test Case 2: Edit Product (PUT request)

- **Test Objective**: Ensure that a product can be updated via the admin panel.
- **Test Steps**:
  1. Send a PUT request to update an existing product's information.
  2. Verify that the product details are updated on the frontend.
- **Expected Result**: The product details should reflect the updated information.

Test Case 3: Delete Product (DELETE request)

- **Test Objective**: Ensure that a product can be deleted via the admin panel.
- **Test Steps**:
  1. Send a DELETE request for a specific product.
  2. Verify that the product is removed from the product listing.
- **Expected Result**: The deleted product should no longer appear in the product list.

5. Pagination:

Test Case 1: Verify Pagination Navigation

- **Test Objective**: Ensure that pagination works and users can navigate through product pages.
- **Test Steps**:
  1. Go to the product listing page.
  2. Click "Next" or "Previous" to navigate between product pages.
  3. Verify that a new set of products is displayed after clicking the navigation buttons.
- **Expected Result**: Clicking "Next" or "Previous" should correctly navigate to the next or previous set of products.

6. Search Bar Functionality:

Test Case 1: Search for a Product by Name

- **Test Objective**: Ensure that users can search for products by name.

- **Test Steps**:
    1. Enter a product name in the search bar.
    2. Verify that the search results display products related to the query.
- **Expected Result**: The search results should display the correct filtered products.

Test Case 2: Search for Products by Category

- **Test Objective**: Ensure users can search products by category.
- **Test Steps**:
    1. Enter a category name in the search bar.
    2. Verify that the search results show products from that category.
- **Expected Result**: The correct filtered products should appear for the chosen category.

Test Case 3: No Results Found

- **Test Objective**: Ensure that the appropriate error message is displayed when no search results match.
- **Test Steps**:
    1. Enter a random or non-existent product name in the search bar.
    2. Verify that the message "No results found" is displayed.
- **Expected Result**: An error message indicating no matching products should be displayed.

7. User Authentication & Authorization:

Test Case 1: User Login

- **Test Objective**: Ensure users can log in securely.
- **Test Steps**:
    1. Enter valid login credentials (username and password).
    2. Click "Login" and verify successful redirection to the user dashboard or homepage.
- **Expected Result**: Users should be redirected to their dashboard/homepage after successful login.

Test Case 2: Unauthorized Access

- **Test Objective**: Ensure unauthorized users cannot access protected routes.
- **Test Steps**:
    1. Try accessing a protected route (e.g., user profile) without logging in.

2. Verify that the user is redirected to the login page or an error message is displayed.

- **Expected Result**: Unauthorized users should be redirected to the login page or shown an error message.

8. Checkout & Cart Flow Enhancements:

Test Case 1: Verify Cart Review and Checkout

- **Test Objective**: Ensure users can review their cart and proceed to checkout.
- **Test Steps**:
  1. Add items to the cart.
  2. Go to the cart page and verify that the cart contents are correct.
  3. Proceed to checkout and verify if all the data is passed correctly.
- **Expected Result**: The checkout page should show all the products added to the cart, including their details.

Test Case 2: Payment Integration

- **Test Objective**: Ensure smooth payment processing during checkout.
- **Test Steps**:
  1. Fill in all required fields in the checkout form.
  2. Select a payment method and complete the transaction.
  3. Verify that the payment is processed successfully.
- **Expected Result**: The payment should be processed smoothly, and users should receive a confirmation of their order.

Simulate User Actions:

- **Clicking**: Click on product listings, pagination buttons, and add-to-cart buttons to simulate user interaction with UI elements.
- **Form Submissions**: Submit search queries, voucher codes, and payment forms to test the system's response.
- **Navigation**: Simulate navigation between product pages, cart, and checkout process to ensure smooth transitions and data retention.

Validate Output Against Expected Results:

- **Check for Correct Data**: Ensure that the data displayed on the frontend matches the data received from the backend.

- **Verify UI Elements**: Confirm that all UI elements (buttons, images, text) are displayed correctly and function as intended.
- **Error Handling**: Ensure that appropriate error messages are shown when users perform incorrect actions (e.g., invalid voucher codes or attempting unauthorized actions).

By following the test cases and user action simulations listed above, functional testing will ensure that the Marketplace Ecommerce Website's features are working as expected and provide a seamless experience for the users.