Type of Testing used in Eshop application:

1. Swagger API Testing
2. UI Automation Testing

**Approach used in Testing:**

**Swagger API Testing:**

In API Testing, endpoints are provided in swagger. Get, put, post, Delete API are being used in testing of this application.

Get: For Get Apis, endpoints are being hit and response is generated. We don’t require Body and parameter for GET API.

Put: For Put Api, along with endpoint body and parameters are required to update any partial information on the server and response code 200 is validated.

Post: For Post Api, along with endpoint body and parameters are required to update any information on the server and response code 200 is validated.

Delete: For Delete Api, endpoint is being hit and response is generated. It doesn’t require any body and parameter to delete any information on the server.

In API Automation testing, Endpoint is being hit via automation code and response is generated and validated via assertions. Rest Assured is used for validating the responses and for passing any parameters in the API request.

We have used Gherkin language for creating the test cases in automation as it is easy to understand by anyone like BA, developers etc.

TestCases which are executed in the automation are:

1. Get: /api/catalog-brands : This API gives us all the catalog brands like Azure, .Net, Visual Studio, SQL Server, Other after hitting the endpoint with status code 200.
2. Post: /api/authenticate: This API authenticates a user by providing username and password. In the request Body, Username and password is being provided to authenticate the user. After passing all the information in the body and request parameters and headers, it gives us the response that user is authenticated.
3. Put : /api/catalog-items: This API updates any information passed in the body for that particular catalog items and provides us the response that item is being updated with status code 200.

**UI Automation testing:**

While UI Automation, we are using Hybrid framework in which test cases are written in feature files in Gherkin language to make it easily readable for everyone. Every step in feature file is linked to a step definition layer where it implements the java method as we are using Java as Object oriented programming language.

Page Object Model is being used in the framework as application contains various pages, so every page in application is linked to a java page. We have used Page Object Model (POM) to avoid redundancy in case of any changes in elements in future. If there is any change by the developers in any of the page then it would be easy in page object model to make those change in our framework by navigating to that particular changed page in framework and changing the particular element. Hence it improves efficiency of code and saves a lot of time for a tester to incorporate any change.

Rest Assured is being implemented in the same framework in pom.xml file as a dependency so that we can use API automation with the same framework.

All the elements are placed in Object Repository file and test runner is created to run any test case and then html report is generated with all the execution results.

Test Cases executed in UI Automation:

1. **Add Product to basket and Checkout**: This is end to end happy flow for adding any product to the basket and then checkout.
2. Login into the application by entering the credentials.
3. Select any product i.e. .Net bot black sweatshirt.
4. Add it into the basket
5. Proceed to checkout by entering all the relavant details.
6. Product must be checked out successfully.
7. Validate the product, address, Value of Product after checkout.
8. All the fields should be correctly entered.
9. **Checking the filters on homepage:** This is both happy as well as negative flow to check the filters on webpage.
10. Login into the application by entering the credentials
11. Select the filters from Brand i.e. .Net and Type is All.
12. Validate the page should be refreshed with all the applied filters.
13. All the product on webpage should be of the selected brand i.e. .Net
14. **Login Logout Functionality**: This is happy and positive flow to check Login and Logout functionality.
15. Login into the Application by entering the valid credentials.
16. User should be logged in to the application successfully.
17. After successful Login, Click on Logout button.
18. Application should get logged out successfully