Shopperstack Application API Testing

1. Project Overview/Introduction

Objective:

The objective of this project is to test and validate various modules of the Shopperstack application using API testing methodologies. The project focuses on verifying key functionalities such as Shopper Profile, Product View Action, Shopper Address, Shopper Wishlist, Shopper Cart, and Shopper Order. Postman is utilized to interact with the APIs, ensuring that the responses meet the expected behaviors in real-world scenarios.

Modules Tested:

- Shopper Profile
- Product View
- Shopper Address
- Shopper Wishlist
- Shopper Cart
- Shopper Order

Technologies Used:

- **Postman**: For sending HTTP requests and validating API responses.
- **Swagger Documentation**: Reference for API endpoints and expected request/response formats.
- **Bearer Token Authentication**: Used for securing API access where required.

2. Test Plan

Scope:

This project covers API testing for the following key functionalities in the Shopperstack application:

1. User Profile Management
2. Product View Actions
3. Address Management
4. Wishlist Management
5. Shopping Cart Operations
6. Order Placement and Management

Test Strategy:
Manual testing is conducted using Postman to execute HTTP requests for each module. Both positive and negative test cases are written to validate various scenarios and error handling. The testing includes response validation, error messages, and ensuring compliance with the Swagger documentation.

Test Environment:
- Postman: For manual request executions.
- Swagger Documentation: Reference for API details.
- Test Data: Data prepared for testing various modules, including valid and invalid inputs.

Tools Used:
- Postman
- Swagger Documentation (https://www.shoppersstack.com/shopping/swagger-ui.html)

3. Test Cases Document

Detailed test cases for the Shopperstack application have been prepared for each module:
- **Shopper Profile**: Includes test cases for creating, updating, and viewing shopper profiles.
- **Product View**: Tests the API response for viewing all available products.
- **Shopper Address**: Validates address creation, update, and deletion functionalities.
- **Shopper Wishlist**: Includes test cases for adding/removing items from the wishlist.
- **Shopper Cart**: Verifies adding, updating, and removing products from the cart.
- **Shopper Order**: Includes test cases for order placement, viewing, and cancellation.

4. API Documentation

The Shopperstack application APIs were tested using the Swagger documentation provided. Below are some key API operations:

Shopper Profile API Operations:
- Create Profile (POST): /shopper/profile/create
- Update Profile (PUT): /shopper/profile/update
- Get Profile (GET): /shopper/profile/{id}

Product View Action API:
- View All Products (GET): /products/view

Shopper Wishlist API:
- Add to Wishlist (POST): /wishlist/add
- Remove from Wishlist (DELETE): /wishlist/{id}

5. Challenges & Learnings

Challenges:
1. **Authentication Handling**: Modules required Bearer token authentication, & its expiry time is only 2 hours, so after every 2 hours we need to generate new token, which had to be managed effectively to ensure valid access during testing.
2. Internal Server Error: Most of the time we get 500 status code while making API call, which create lot of issue while performing API testing.

Learnings:
- Developed a strong understanding of how to handle real-world API scenarios in retail applications, including managing shopper sessions, validating product inventories, and handling error cases in wishlist and cart functionalities.
- Improved proficiency in using Postman for API testing, including setting up authentication and writing Post-request validations for responses.

6. Test Summary Report

Test Module	Total Test Cases	Passed	Failed
Shopper Profile	5	5	0
Product View	4	4	0
Shopper Address	5	5	0
Shopper Wishlist	9	7	2
Shopper Cart	6	6	0
Shopper Order	4	4	0
Overall	33	31	2

7. Conclusion

The Shopperstack API testing project validated multiple functionalities across different modules of the application. Positive and negative test cases were executed for six key modules, and responses were validated for both correct functionality and proper error handling.

By leveraging Postman and Swagger documentation, I gained hands-on experience in API testing, authentication, and response validation. This project also helped deepen my understanding of real-world API testing for e-commerce applications, ensuring a solid foundation for future roles in software testing.