Tool Selection: Playwright
Application Selection: Online Store

a. Test Case Description:

In this test case we test the main and mostly user functionality od the product which are
- Welcome page
- User can login with the valid credentials
- User can search for a product
- User can add a product to his/her cart
- User can proceed to the checkout page
- User can submit the purchases

Test Case ID: TC_001
Test Title: Verify a product purchase
Test Priority: High
Preconditions:
- The application is accessible and running.
- A valid user account exists in the system
- There are products in the database which tests can run.

Test Description:
As a client, I can login, search for a product and purchase it through the website. Test Steps:
1. Navigate to the login page of the application.
2. Login with given credentials
3. Search for a product and add to the cart
4. Do a checkout
5. Submit order

Test Data:
- User's username: anshika@gmail.com
- User's password: Iamking@000
- Product name: ADIDAS ORIGINAL

Expected Result:
- The user is successfully logged in.
- User can search for a product.
- User can add products to his/her cart.
- User can go to checkout page.
- User can submit order.

Actual Result:
- The user can successfully log in.
- User can search for a product.
- User can add products to his/her cart.
- User can go to checkout page.
- User can submit

B - Test Design and Strategy

Test Strategy for Verifying a Product Purchase Feature
The goal is to ensure the "verify a product purchase" feature works correctly, reliably, and securely across all intended use cases. This includes confirming that purchases are accurately recorded, validated, and reflected in the system, while providing a seamless user experience.

Scope

- Verification of purchase initiation (e.g., clicking "buy" or "checkout").
- Validation of payment processing (successful, failed, or canceled transactions).
- Confirmation of purchase completion (e.g., receipt generation, user notifications).
- Backend data updates (e.g., order history, inventory).

Test Approach

- **Manual Testing**:
  - Exploratory testing to identify unexpected issues (e.g., UI glitches during verification).
  - Test with real-world scenarios (e.g., slow internet, interrupted transactions).
- **Automated Testing**:
  - Write unit tests for individual components (e.g., payment validation logic).
  - Create end-to-end tests using tools like Selenium or Cypress to simulate a full purchase flow.
  - Automate API tests to verify backend responses (e.g., status codes, data consistency).

Success Criteria

- 100% of critical test cases pass (e.g., successful purchase, failed payment handling).
- No severity-1 bugs (e.g., data loss, security breaches).
- Feature meets performance benchmarks (e.g., <2-second verification time).
- Positive feedback from usability testers.