

3. Test Automation Basics

Assume you are asked to create a test automation framework for a web application of an online shopping site. The application includes functionalities such as user login, registration, product search, add to cart, checkout, and order history etc...

What is your approach? (No need to write scripts/codes)

Approach to create a test automation framework for a Web Application of an Online Shopping Site

1. Define the scope of the automation:
Identify key functionalities to automate, such as user login, registration, product search, add to cart, checkout, and order history.
Prioritize test cases for automation, focusing on repetitive and high-risk areas.
2. Select tools and technologies:
Choose automation tools like Selenium or Cypress.
Use a language compatible with the chosen tool (E.g: Java, JavaScript, Python).
CI/CD integration.
3. Design the test automation framework:
Organize the framework with a clear folder structure.
4. Implement reusable components:
Use the page object model for creating reusable page classes.
Develop utility functions for common tasks.
5. Develop and manage test cases:
Write simple, independent test cases for each functionality.
Implement data-driven testing to cover various scenarios.
6. Reporting and continuous integration:
Integrate reporting tools (Allure, ExtentReports) for detailed test execution reports.
Set up a CI/CD pipeline to run automated tests on code commits or scheduled intervals.

7. Maintainability and continuous improvement:

Regularly update test cases to reflect application changes.

Refactor code for readability and maintainability.

Collect feedback and continuously enhance the framework.

8. Documentation and training:

Document the framework setup, design, and usage.

Provide training to ensure the team can effectively use and maintain the framework.