Test Automation Basics

1. Requirement Analysis

- Identify key features like user login, registration, product search, add to cart, checkout, and order history.
- Prioritize test cases that need automation (e.g., login, checkout, payment).

2. Select the Right Tools

- **Automation Tool**: Use Selenium WebDriver for web automation.
- **Test Framework**: Select a test framework like TestNG (for Java) or Mocha (for JavaScript).

3. Design the Framework

- **Page Object Model (POM)**: Create separate classes for each web page to keep the code organized.
- **Data-Driven Testing**: Store test data (e.g., login credentials) in external files to reuse across tests.
- **Reusable Components**: Write reusable functions (e.g., login, add to cart) to avoid repeating code

4. Write Test Scenarios

- **Smoke Tests**: Quick tests to check major functions (e.g., login, add to cart).
- **Functional Tests**: Detailed tests for each feature (e.g., registration, product search).
- **Regression Tests**: Test all critical workflows (e.g., checkout, payment)

5. **Implement Logging and Reporting**

- Add logs and screenshots for failed tests to help with debugging.
- Generate test reports after each test run.

6. Run tests across different browsers (Chrome, Firefox, etc.) using tools like Browser Stack.

7. Maintain the Framework

- Regularly update the tests and framework as the website changes.
- Use Git for version control to manage updates.