When creating a test automation framework for a web application like an online shopping site, my approach will include the following steps:

1. Understanding Requirements:

- Understand the key functionalities of the web application, such as user login, registration, product search, add to cart, checkout, and order history.

- Identify important areas that need to be automated based on the application's importance and user interactions.

2. Choosing the Right Automation Tools:

- Select appropriate tools for automation. Common choices for web applications include Selenium WebDriver for UI automation, TestNG for test management, and Maven build management.

3. Designing the Automation Framework:

- Choose the type of framework based on the application's complexity and team experience.

- Use Page Object Model (POM):
  Create separate classes for each web page to divide page-specific behaviors and elements.

4. Setting Up the Test Environment:

- Create a configuration file to manage test settings like browser types, URLs, timeout.

- Prepare test data sets if needed.(e.g., Excel)

- Use a version control system like Git to manage and track changes.

5. Write Test scripts and Execute:

- Start by automating high-priority test cases, such as those related to critical functionalities like login, checkout, and payment.

- Update and maintain the test scripts as the application evolves, ensuring that the automated tests remain relevant and effective.