**Automate an E-Commerce Web Application**

**Step 1: Setup the Project**

1. Create a new Java project in your preferred Integrated Development Environment (IDE).

2. Add the necessary Selenium WebDriver and TestNG dependencies to your project's build path or Maven/Gradle configuration.

**Step 2: WebDriver Initialization**

1. Import the required packages for Selenium WebDriver, TestNG, and Java utilities.

2. Set the system property for the specific WebDriver executable (e.g., ChromeDriver).

3. Initialize the WebDriver instance (e.g., ChromeDriver) and maximize the window.

**Step 3: Navigate to Flipkart Homepage and Measure Page Load Time**

1. Use the WebDriver to navigate to the Flipkart homepage (https://www.flipkart.com/).

2. Calculate the page load time by noting the start time and end time of the navigation.

3. Use TestNG's `Assert` to verify that the correct URL has been loaded.

**Step 4: Search for a Product (e.g., "iPhone 13" under "Mobile" Category)**

1. Find the search box element on the page and enter the search query (e.g., "iPhone 13").

2. Submit the search query and wait for the search results to load.

**Step 5: Check Images Visibility till Screen Height**

1. Use JavaScriptExecutor to determine if the last image on the page is visible till the screen height.

2. Use TestNG's `Assert` to check if the image is visible till the screen height.

**Step 6: Check for Page Scroll Feature**

1. Use JavaScriptExecutor to check if the page is scrollable (i.e., the document height is greater than the window height).

2. Use TestNG's `Assert` to check if the page has a scroll feature.

**Step 7: Verify Image Download and Display Time**

1. This step is a bit challenging to automate through WebDriver alone.

2. The verification of image download and display time would depend on the website's implementation and may require other tools or manual validation.

**Step 8: Scroll to the Bottom of the Page**

1. Use JavaScriptExecutor to scroll to the bottom of the page.

2. Use TestNG's `Assert` to verify that the page has navigated to the bottom.

**Step 9: Check Rendering on Different Browsers and Screen Resolutions**

1. To check rendering on different browsers, you can use WebDriver's capabilities to instantiate different browser drivers (e.g., FirefoxDriver for Firefox) and run the same tests.

2. For checking different screen resolutions, you can use WebDriver's `Dimension` class to resize the browser window to different dimensions and observe the rendering.

**Step 10: Run the Test**

1. Create a TestNG XML file to specify the test suite configuration.

2. Add the test class containing the test methods to the TestNG XML file.

3. Right-click on the XML file and run it as a TestNG suite.

**Step 11: Analyze Test Results**

1. After running the test, check the TestNG test report to see the results of each step.

2. Review any failures or errors and investigate potential issues.

**Step 12: Refactor and Improve**

1. If necessary, refactor the code to make it more efficient, maintainable, and readable.

2. Add error handling and retries to make the test robust.

3. Explore additional Selenium WebDriver features or third-party libraries that may aid in testing and validation.