Galen: Proof of Concept

Task to accomplish

Use Galen to ensure that the responsive website works as expected on various devices and platforms.

What is Galen?

Galen Framework is an open-source layout and functional testing framework for websites, which allows testing the look and feel of responsive websites. It has its special language Galen Specs for describing the positioning, alignment and CSS properties of elements on a Web page. It uses Selenium for interacting with elements on page and getting their locations and dimensions.

What is Galen Specs?

Galen Specs is the language which defines the tests that have to performed for different devices. The language is quite advanced and allows to express complete layout of your website with minimal text.

```
= Main section =
    @on *
        header:
            height 100px
            inside screen 0px top
            width 100% of screen/width

        menu:
            height 50 to 60px
            width 100% of screen/width
            below header ~ 0px

        search-button:
            inside menu 20 to 50px left, 0 to 10px top
```

How Galen runs?

Galen Framework was made with responsive design in mind. It works in a following way:

- Opens a page in browser
- Navigate to desired page
- Resizes browser to specified size
- Tests the layout with Galen Specs

Key Features

1. Galen a tool for cross-browser layout testing
2. Inbuilt HTML Error Reporting with screenshots for Pass and Fail
3. Multiple browser support for testing on various platforms
4. Supports functional testing
5. Can be integrated with Browser Stack, and Sauce labs
6. Has rich JavaScript API that gives you a chance to work with WebDriver specifically

Galen Reporting

Galen has inbuilt Html reporting

- Error Reporting
  Generates Html reports where we can see all test objects and their status.
- Screenshots
  Highlights the failed testcases.
- Image Comparison
  Compares images and shows differences

How to set up?

1.Set up a maven project and add Junit, hamcrest and galen dependency in pom.xml

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.2</version>
</dependency>

<dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest-all</artifactId>
    <version>1.3</version>
</dependency>

<dependency>
    <groupId>com.galenframework</groupId>
    <artifactId>galen-java-support</artifactId>
    <version>2.3.0</version>
</dependency>
```

2.Extend GalenJUnitTestBase class and set the web driver path

```
public class GalenLayoutTesting extends GalenJUnitTestBase {


    @Override
    public WebDriver createDriver() {
        System.setProperty("webdriver.chrome.driver","//webdriver path//");
        return new ChromeDriver();
    }
```

3.Under Parameterized tag define list of devices on which test has to performed with respective tag name and dimensions

```
@Parameterized.Parameters
public static Iterable<Object[]> devices() {
    return Arrays.asList(new Object[][] {

            //device name and size
            { new DeviceSetup(new Dimension( width: 1920,  height: 1080),  ...tags: "normal", "desktop") },

            { new DeviceSetup(new Dimension( width: 320,  height: 800),  ...tags: "normal-phone", "phone", "mobile") }

    });
}
```

4. Under Test tag perform automation to navigate to desired webpage using selenium if required and use checkLayout function to perform layout testing

```
  resize(device.getScreenSize().getWidth(), device.getScreenSize().getHeight());
  checkLayout( specPath: "HomePage.spec", device.getTags());
```

How to write spec sheet?

Step 1: Define objects under @Object tag

```
@objects
    main                id      main-container
    menu                css     ul.menu
    comments            css     div.comments
```

Step 2: Define the respective checks to be performed in a section

```
= Main section =
    menu:
        width ~ 100 % of main/width

    comments:
        width 40 % of main/width
        height 90 to 100 % of main/height
```

Common test that can be performed

- Absent
- Near
- Above
- Below
- Inside
- Aligned
- Centered
- Color scheme
- Component
- Visible
- Text
- On
- Width
- Height
- Contains

Capabilities of Galen

- Testing relative location of elements in web page
- Checking visible text
- Integration with Selenium Grid
- Inject JavaScript into code
- Color scheme testing
- Image Comparision

Disadvantage of Galen

- Writing Spec sheets take a lot of time
- Code maintenance is costly when there are lot of UI changes