

```
// pages/NavigationLinksPage.java  
package pages;
```

```
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import  
org.openqa.selenium.support.ui.ExpectedCon  
ditions;  
import  
org.openqa.selenium.support.ui.WebDriverWa  
it;  
import java.time.Duration;
```

```
public class NavigationLinksPage {
```

```
    WebDriver driver;  
    WebDriverWait wait;
```

```
    public NavigationLinksPage(WebDriver  
driver) {  
        this.driver = driver;  
        this.wait = new WebDriverWait(driver,
```

```
Duration.ofSeconds(10));  
}
```

// Locators for Bar Links (Top Navigation)

```
private final By contactBarLink =
```

```
By.xpath("//li[contains(@class,'nav-item')]//  
a[normalize-space()='Contact']");
```

```
private final By homepageBarLink =
```

```
By.xpath("//li[contains(@class,'nav-item')]//  
a[normalize-space()='Homepage']");
```

```
private final By locationBarLink =
```

```
By.xpath("//li[contains(@class,'nav-item')]//  
a[normalize-space()='Location']");
```

// Locators for Quick Links (Footer)

```
private final By quickLinksSection =
```

```
By.xpath("//h2[normalize-space()='Quick  
Links']");
```

```
private final By footerContactLink =
```

```
By.xpath("//div[@id='root']//a[normalize-  
space()='contact']");
```

// Methods

```
public void clickBarLink(String linkName) {  
    By locator;  
    switch (linkName.toLowerCase()) {  
        case "contact" -> locator =  
contactBarLink;  
        case "homepage" -> locator =  
homepageBarLink;  
        case "location" -> locator =  
locationBarLink;  
        default -> throw new  
IllegalArgumentException("Invalid link name:  
" + linkName);  
    }  
  
    wait.until(ExpectedConditions.elementToBeCl  
ickable(locator)).click();  
}  
  
public void clickQuickLink(String linkName)  
{
```

```
By locator;  
switch (linkName.toLowerCase()) {  
    case "contact" -> locator =  
footerContactLink;  
    default -> throw new  
IllegalArgumentException("Invalid quick link  
name: " + linkName);  
}
```

```
wait.until(ExpectedConditions.elementToBeCl  
ickable(locator)).click();  
}
```

```
public String getCurrentURL() {  
    return driver.getCurrentUrl();  
}
```

```
public boolean  
isQuickLinksSectionDisplayed() {  
    try {  
        return  
wait.until(ExpectedConditions.visibilityOfElem
```

```
entLocated(quickLinksSection)).isDisplayed();  
    } catch (Exception e) {  
        return false;  
    }  
}  
}
```