

Srabone Raxit

```
package Testing;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;

public class RegisterAutomation {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\hp\\libs\\chromedriver-
win64\\chromedriver.exe");

        ChromeOptions options = new ChromeOptions();
        options.addArguments("--remote-allow-origins=*");
        WebDriver driver = new ChromeDriver(options);

        try {
            // Navigate to the register page
            driver.get("http://productedge.fwh.is/views/register.php");
            driver.manage().window().maximize();
            Thread.sleep(2000);

            // === Test 1: Submit without filling any field ===
            System.out.println("Testing empty form submission...");
            WebElement submitButton = driver.findElement(By.xpath("//button[@type='submit']"));
            submitButton.click();
        }
    }
}
```

```

Thread.sleep(1000);

// Check error messages for required fields
verifyErrorMessage(driver, "name", "Name is required");
verifyErrorMessage(driver, "username", "Username is required");
verifyErrorMessage(driver, "password", "Password must contain at least one lowercase
letter, one uppercase letter, one special character, and be at least 8 characters long");
verifyErrorMessage(driver, "role", "Please select a role");
verifyErrorMessage(driver, "email", "Email is required");
verifyErrorMessage(driver, "phone_number", "Phone Number is required");
verifyErrorMessage(driver, "address", "Address is required");

// === Test 2: Invalid password format ===
System.out.println("\nTesting invalid password format...");
fillInputField(driver, "name", "");
fillInputField(driver, "username", "srabone12");
fillInputField(driver, "password", "12345"); // Invalid password
fillSelectField(driver, "role", "customer");
fillInputField(driver, "email", "srabone@gmail.com");
fillInputField(driver, "phone_number", "1234567890");
fillInputField(driver, "address", "Gazipur");
submitButton.click();
Thread.sleep(1000);

verifyErrorMessage(driver, "password", "Password must contain at least one lowercase
letter, one uppercase letter, one special character, and be at least 8 characters long");

// === Test 3: Successful form submission ===
System.out.println("\nTesting successful form submission...");

```

```

        fillInputField(driver, "password", "Password@123"); // Valid password
        submitButton.click();
        Thread.sleep(2000);

        // Verify successful submission (you may check for redirection or success message)
        String currentUrl = driver.getCurrentUrl();
        if (currentUrl.contains("register_controller.php")) {
            System.out.println("Test Passed: Form submitted successfully.");
        } else {
            System.out.println("Test Failed: Form submission failed.");
        }

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // Close the browser
        driver.quit();
    }
}

// Helper method to fill input fields
public static void fillInputField(WebDriver driver, String fieldId, String value) {
    WebElement field = driver.findElement(By.id(fieldId));
    field.clear();
    field.sendKeys(value);
}

```

```

// Helper method to fill select dropdowns

public static void fillSelectField(WebDriver driver, String fieldId, String value) {

    WebElement dropdown = driver.findElement(By.id(fieldId));

    dropdown.click();

    WebElement option = driver.findElement(By.xpath("//select[@id='" + fieldId +
    "']/option[@value='" + value + "']"));

    option.click();

}

// Helper method to verify error messages

public static void verifyErrorMessage(WebDriver driver, String fieldId, String
expectedMessage) {

    WebElement field = driver.findElement(By.id(fieldId));

    WebElement parent = field.findElement(By.xpath("../..")); // Get the parent element

    WebElement errorMessage = parent.findElement(By.className("error-message"));

    if (errorMessage != null && errorMessage.getText().equals(expectedMessage)) {

        System.out.println("Test Passed: Correct error message for " + fieldId);

    } else {

        System.out.println("Test Failed: Incorrect or missing error message for " + fieldId);

    }

}

}

```