**Automate a Web Application(Selenium)**

**Algorithm:**

**Login.java:**

1. Imports necessary Selenium packages.

2. Sets up the WebDriver using ChromeDriver.

3. Navigates to the login page by loading a local HTML file "login.html" from the classpath.

4. Calls the `login()` method to perform the login operation with provided email and password.

5. Pauses execution for 1 second (1000 milliseconds) before closing the browser.

6. The `login()` method locates the email, password input fields, and the submit button on the login page, fills in the provided email and password, and clicks the submit button.

7. It then waits for 1 second (1000 milliseconds) before printing a success message.

**Registration.java:**

1. Imports necessary Selenium packages.

2. Sets up the WebDriver using ChromeDriver.

3. Displays the welcome page by loading a local HTML file "welcome.html" from the classpath.

4. Pauses execution for 2 seconds (2000 milliseconds) for manual interaction with the page (for demonstration purposes).

5. Prints a "Register Success" message after the pause.

6. Pauses for 2 seconds (2000 milliseconds) before displaying the registration page.

7. Calls the `registerNewEmployee()` method to automate the registration process with provided name, email, and password.

8. The `registerNewEmployee()` method navigates to the registration page by loading a local HTML f ile "registration.html" from the classpath.

9. It fills out the registration form using the provided name, email, and password and clicks the register button.

10. The method then checks if the registration was successful by looking for the success message element. If the message is displayed, it prints a "Register Success" message; otherwise, it prints a "Registration page closed" message.