1. Automate any web login flow using Selenium

Step 1: Add Selenium to your pom.xml (if using Maven)

```
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.21.0</version>
    </dependency>
</dependencies>
```

Step 2: Java Code to Automate Login Flow

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

public class WebLoginAutomation {

    public static void main(String[] args) {
        // Set the path to your chromedriver executable
        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

        // Initialize the Chrome driver
        WebDriver driver = new ChromeDriver();

        try {
            // Navigate to the login page
            driver.get("https://example.com/login");

            // Maximize the browser window
```

```
        driver.manage().window().maximize();

        // Find username and password fields and input credentials
        WebElement username = driver.findElement(By.id("username"));
        WebElement password = driver.findElement(By.id("password"));

        username.sendKeys("yourUsername");
        password.sendKeys("yourPassword");

        // Click the login button
        WebElement loginButton = driver.findElement(By.id("loginBtn"));
        loginButton.click();

        // Optionally, add a simple wait or check to verify login success
        Thread.sleep(3000); // Simple wait, replace with WebDriverWait in real use

        if (driver.getCurrentUrl().contains("dashboard")) {
            System.out.println("Login successful!");
        } else {
            System.out.println("Login failed or redirected incorrectly.");
        }

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