Document of Selenium Automation Testing

Department of Information Technology

Faculty of Computing

Software Quality Assurance

Individual Assignment

(BSc) Hons Software Engineering UGC batch 3

A.G.N.K. Sandali Wijesooriya -IT_UGC/001/B003/0013

08/2024

Contents

List of Figures

List of Tables

Introduction

In today's rapidly developed software development environment, ensuring quality, accuracy, and reliability is crucial for every software. This testing can be executed mainly in two ways either manual or automated. However, automated testing can achieve this goal efficiently, effectively, and accurately. There are several automation testing stools such as Cenorex, Cypress, playwrite, selenium, etc. Among all those tools Selenium, an open-source framework for web application testing, is widely recognized for its flexibility, robustness, and ease of integration with various development environments. Therefore, here this automation testing was done by using selenium.

This report provides everything in detail about the implementation of Selenium WebDriver in IntelliJ IDE to automate the login process for LinkedIn, a professional networking platform. This report will highlight every key aspect from the beginning to the final output in testing. This document not only highlights Selenium's practical application in automating web interactions but also explores the various challenges encountered during the setup and execution phases. By using this selenium web driver here this project aims to identify the core features, procedures, and key aspects of automation testing.

Selection and Identification

Why Selenium? [1]

- Selenium was selected as the automation tool for this project due to its adaptability feature, its free of use, and higher support for web application automation testing.
- Selenium provides flexibility and cost-effectiveness, making it accessible to all beginner-level developers, single person as well as for large enterprises.
- support multiple programming languages such as Java, Python, and C#, along with its cooperation with major browsers like Chrome, Firefox, and Edge.
- A core component of the Selenium suite, allows for direct communication with web browsers, enabling highly accurate and reliable test execution.

Why LinkedIn?

- LinkedIn was chosen as the target web application for automation due to its globally recognized highly used platform with over 700 million users, Therefore, its' login process represents the capability of selenium WebDriver in automation testing.
- Automating the login process for LinkedIn allows for the identification of several Selenium features, such as locating web elements, verifying credentials, and validating successful user authentication.
- As a beginner in automation testing, choosing this one core feature in a popular platform helped me get basic knowledge clearly on the procedure.

Why IntelliJ IDEA? [2]

- IntelliJ IDEA was selected as the Integrated Development Environment (IDE) for this project due to its integration with Java the programming language which is used for Selenium automation.
- Its built-in support for Maven, a project management and comprehension tool, simplifies dependency management and project configuration, making it easier to set up and maintain Selenium projects.
- IntelliJ IDEA's user-friendly interface and extensive plugin ecosystem provide a clean environment for both beginner and experienced developers to efficiently develop, test, and debug their automation scripts.

Test Case Design

For the LinkedIn login functionality, here have designed mainly 4 test cases as below, [3]

 TC001: Valid username and valid password

- ***Objective:*** Verify that a user with valid credentials can successfully log in to LinkedIn.
- ***Expected Result*:** The user is redirected to their LinkedIn feed upon successful login and the test displays a successful log

 TC002: Invalid username and invalid password

- ***Objective:*** Verify that a user with invalid credentials can't log into LinkedIn.
- ***Expected Result:*** An error message indicates that the credentials are incorrect on the LinkedIn site and the test displays logging failed.

 TC003: Valid username and invalid password

- ***Objective:*** Verify that a user with a valid username but an invalid password cannot log in.
- ***Expected Result:*** An error message indicates that the password or username is incorrect on the LinkedIn site and the test displays logging failed.

 TC004: Invalid username and valid password

- ***Objective:*** Verify that a user with an invalid username but a valid password cannot log in.
- ***Expected Result:*** An error message indicates that the username or password is incorrect on the LinkedIn site and the test displays logging failed.

Environment Set Up

Key Steps [4]

1. Download IntelliJ IDE, Set up SDK and JDK
2. Download and install a webdriver.exe (here, msedgedriver.exe)
3. Import selenium jar to IntelliJ IDE
4. Create a new java file
5. Import relevant sources
6. Inspect the selected site for Id s and for names (here, inspect the LinkedIn login page)
7. Implement the code
8. Run the code
9. Get results
10. Continue with a set of different data

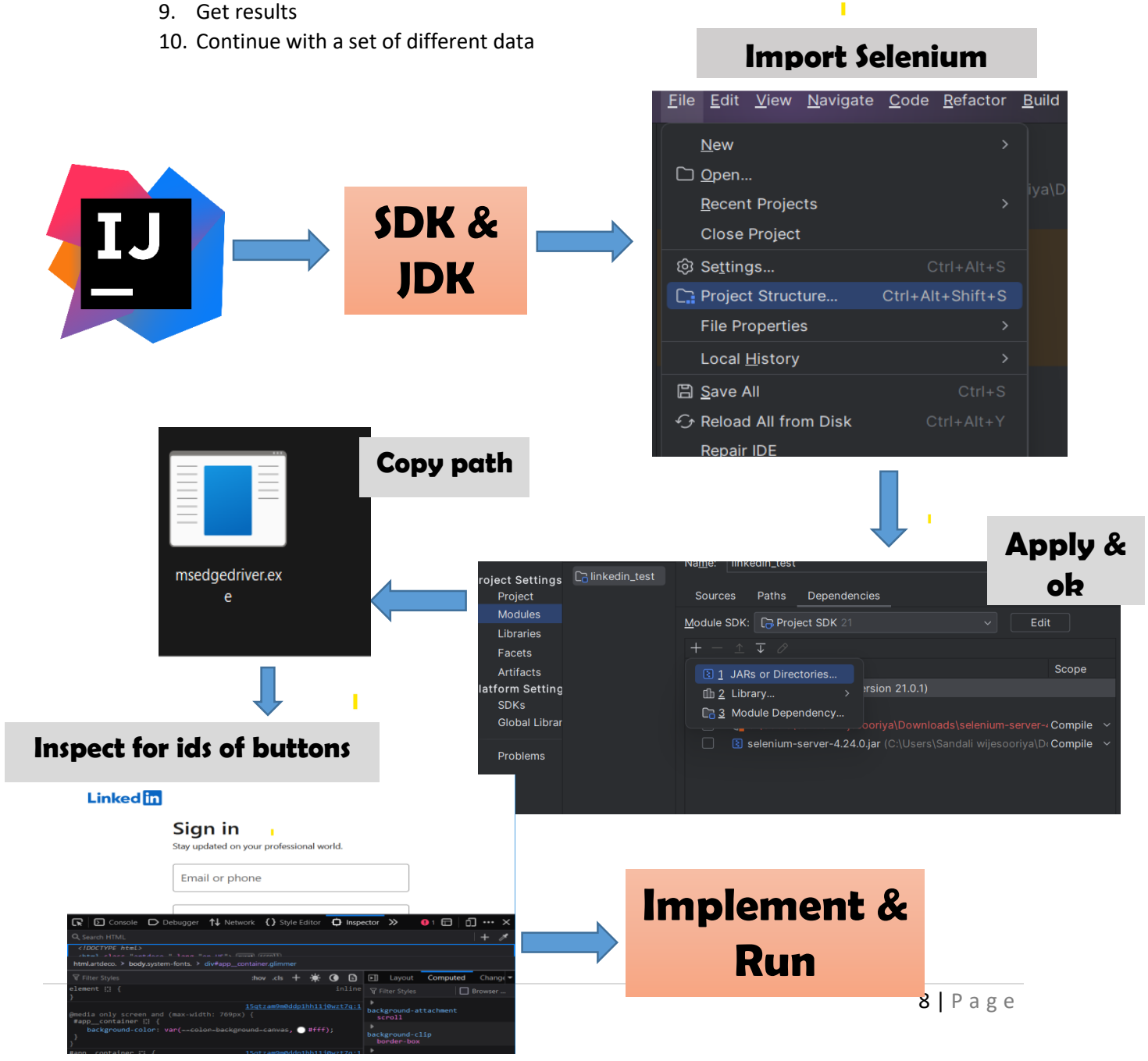


Figure 1- implementation procedure

Implementation

Imported elements

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.edge.EdgeOptions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import java.time.Duration;
```

Figure 2- code part 1

- **import org.openqa.selenium.By;** --used to locate elements on a web page, such as by ID, name, class name, XPath, or CSS selector.
- **import org.openqa.selenium.WebDriver;**-- It allows to perform actions like navigating to a webpage, clicking buttons, etc.
- **import org.openqa.selenium.WebElement;**-- It provides methods to interact with these elements, like clicking, typing, and retrieving text.
- **import org.openqa.selenium.edge.EdgeDriver;**-- It allows Selenium to control and automate interactions in Edge.
- **import org.openqa.selenium.edge.EdgeOptions;**-- used to configure various settings for the Edge browser, such as disabling notifications, running in headless mode, or setting preferences, before running it with Edge.
- **import org.openqa.selenium.support.ui.ExpectedConditions;**-- used to wait for certain states or events on the web page, such as the appearance of an element or the visibility of text.
- **import org.openqa.selenium.support.ui.WebDriverWait;**-- useful for handling dynamic content or delays in web page loading.
- **import java.time.Duration;** -- to define how long to wait for a condition.

Set path to Web Drive

```
Set the correct path to microsoft edge driverG
tem.setProperty("webdriver.chrome.driver", "C:\\Users\\Sandali wijesooriya\\Desktop\\1\\msedgedriver.exe");

eOptions options = new EdgeOptions();
ions.addArguments("--disable-notifications");

Initialize WebDriver
Driver driver = new EdgeDriver(options);
```

Figure 3-code part 2

Initialize by passing data to be checked

```
// Open LinkedIn login page
driver.get("https://www.linkedin.com/checkpoint/lg/sign-in-another-account?trk=guest_homepage-basic_nav-header-signin"


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

// inspect id and add, then add data need to be passed as the username
WebElement emailField = driver.findElement(By.id("username"));
emailField.sendKeys( ...keysToSend: "menuznethuzhewage@gmail.com");

// inspect id and add, then add data need to be passed as the password
WebElement passwordField = driver.findElement(By.id("password"));
passwordField.sendKeys( ...keysToSend: "Pqck13877*");

// Searching for sign-in button and click it
WebElement loginButton = driver.findElement(By.xpath( xpathExpression: "//button[@type='submit']"));
loginButton.click();
```

Figure 4- Code Part 3

Get output

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));
boolean loginSuccessful = wait.until(ExpectedConditions.or(
        ExpectedConditions.urlContains( fraction: "/feed/"), // URL changes on successful login
        ExpectedConditions.visibilityOfElementLocated(By.id("error-for-username")) // Error message for incorrect lo
));

if (loginSuccessful) {
    if (driver.getCurrentUrl().contains("/feed/")) {
        System.out.println("Login successful!");
    } else {
        System.out.println("Login failed!");
    }
} else {
    System.out.println("Login failed due to an unexpected reason!");
}
```

Figure 5-code part 4

Exception handing

```
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Login failed due to an exception!");
        } finally {
            // Close the browser
            //driver.quit();
        }
    }
}
```

Figure 6-code part 5

Project Format

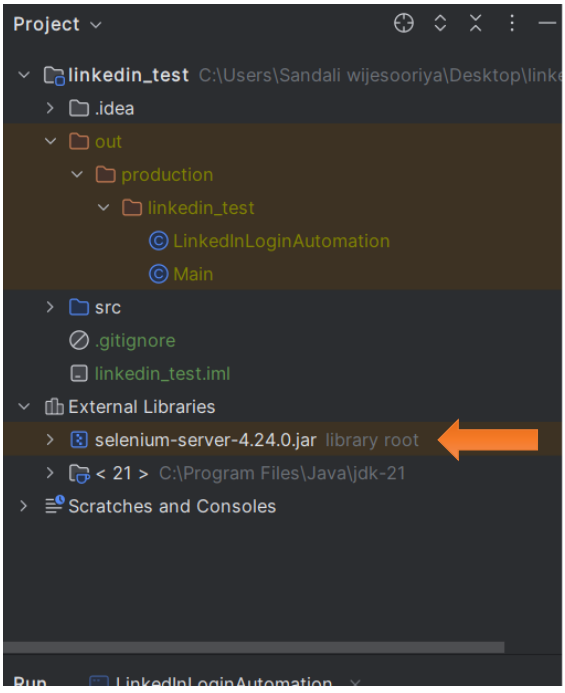


Figure 7- file structure

Test Cases Automation

Test 01- valid Username and password

```
// inspect id and add, then add data need to be passed as the username
WebElement emailField = driver.findElement(By.id("username"));
emailField.sendKeys( ...keysToSend: "menuznethuzhewage@gmail.com");

// inspect id and add, then add data need to be passed as the password
WebElement passwordField = driver.findElement(By.id("password"));
passwordField.sendKeys( ...keysToSend: "Pgck13877*");

// Searching for sign-in button and click it
WebElement loginButton = driver.findElement(By.xpath( xpathExpression: "//button[@type='submit']"));
loginButton.click();
```

Figure 8- Test Case 01

```
Aug 30, 2024 2:34:18 PM org.openqa.selenium.devtools.CdpVersionFin
WARNING: Unable to find CDP implementation matching 128
Aug 30, 2024 2:34:18 PM org.openqa.selenium.chromium.ChromiumDrive
WARNING: Unable to find version of CDP to use for 128.0.2739.42. Y
Login successful!

Process finished with exit code 0
```

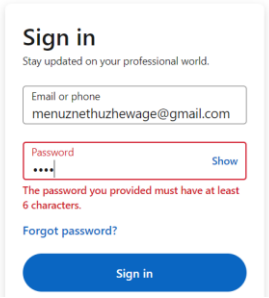
Test 02- valid username and invalid password

```
// inspect id and add, then add data need to be passed as the username
WebElement emailField = driver.findElement(By.id("username"));
emailField.sendKeys( ...keysToSend: "menuznethuzhewage@gmail.com");

// inspect id and add, then add data need to be passed as the password
WebElement passwordField = driver.findElement(By.id("password"));
passwordField.sendKeys( ...keysToSend: "Pgck");

// Searching for sign-in button and click it
WebElement loginButton = driver.findElement(By.xpath( xpathExpression: "//button[@type='submit']"));
loginButton.click();
```

Figure 9- Test Case 02




```
Session ID: 1f21c236388acd06ff5aeb20077e15d7
    at org.openqa.selenium.support.ui.WebDriverWait.timeoutException(W
    at org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java
    at LinkedInLoginAutomation.main(LinkedInLoginAutomation.java:43)
Login failed due to an exception!

Process finished with exit code 0
```

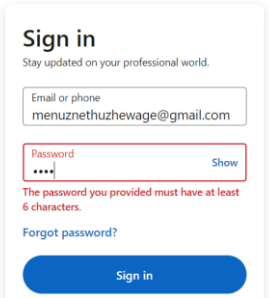
Test 03- invalid username and valid password

```
// inspect id and add, then add data need to be passed as the username
WebElement emailField = driver.findElement(By.id("username"));
emailField.sendKeys( ...keysToSend: "sandali@gmail.com");

// inspect id and add, then add data need to be passed as the password
WebElement passwordField = driver.findElement(By.id("password"));
passwordField.sendKeys( ...keysToSend: "Pgck13877*");

// Searching for sign-in button and click it
WebElement loginButton = driver.findElement(By.xpath( xpathExpression: "//button[@type='submit']"));
loginButton.click();
```

Figure 10- Test Case 03




```
Session ID: 1f21c236388acd06ff5aeb20077e15d7
    at org.openqa.selenium.support.ui.WebDriverWait.timeoutException(W
    at org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java
    at LinkedInLoginAutomation.main(LinkedInLoginAutomation.java:43)
Login failed due to an exception!

Process finished with exit code 0
```

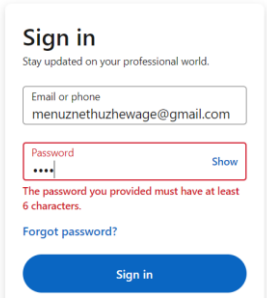
Test 04 invalid username and password

```
// inspect id and add, then add data need to be passed as the username
WebElement emailField = driver.findElement(By.id("username"));
emailField.sendKeys( ...keysToSend: "sandali@gmail.com");

// inspect id and add, then add data need to be passed as the password
WebElement passwordField = driver.findElement(By.id("password"));
passwordField.sendKeys( ...keysToSend: "Pg3877*");

// Searching for sign-in button and click it
WebElement loginButton = driver.findElement(By.xpath( xpathExpression: "//button[@type='submit']"));
loginButton.click();
```

Figure 11- Test Case 04




```
Session ID: 1f21c236388acd06ff5aeb20077e15d7
    at org.openqa.selenium.support.ui.WebDriverWait.timeoutException(W
    at org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java
    at LinkedInLoginAutomation.main(LinkedInLoginAutomation.java:43)
Login failed due to an exception!

Process finished with exit code 0
```

Why Warnings??

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.5\l
Aug 30, 2024 2:43:05 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 128
Aug 30, 2024 2:43:05 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 128.0.2739.42. You may need to include a dependency on a specific version o
org.openqa.selenium.TimeoutException: Expected condition failed: waiting for at least one condition to be valid: url to conta
Build info: version: '4.24.0', revision: '748ffc9bc3'
System info: os.name: 'Windows 11', os.arch: 'amd64', os.version: '10.0', java.version: '21.0.1'
Driver info: org.openqa.selenium.edge.EdgeDriver
Capabilities {acceptInsecureCerts: false, browserName: MicrosoftEdge, browserVersion: 128.0.2739.42, fedcm:accounts: true, ms
Session ID: 1f21c236388acd06ff5aeb20077e15d7
    at org.openqa.selenium.support.ui.WebDriverWait.timeoutException(WebDriverWait.java:84)
    at org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java:228)
    at LinkedInLoginAutomation.main(LinkedInLoginAutomation.java:43)
```

Figure 12- warnings in terminal

- **CDP Version Mismatch Warning**: Have used the most compatible version but not the latest version, all the warnings are due to that version but this is the version I have to use with my device
- **TimeoutException:** got this warning with an invalid username or password due to login failure

Testing results

Test case ID	Test case Description	Automated Test steps	Test data	Expected Output	Actual Output	Status
TC001	Valid username and valid password	1. Navigate to the LinkedIn login page 2. Enter a valid username 3. Enter a valid password 4. Click the "Sign in" button	Valid username “menuznethuzhewage @gmail.com” Valid password “Pgck13877*”	Successfully logged in and directed to LinkedIn home page. In the test, terminal prints were logged successfully	Successfully logged in and directed to LinkedIn home page. In the test, terminal prints were logged successfully	pass
TC002	A valid username and invalid password	1. Navigate to the LinkedIn login page 2. Enter a valid username 3. Enter an invalid password 4. Click the "Sign in" button	Valid username “menunethuzhewage @gmail.com” Invalid password “pgck”	Should not be able to log in and Display error too short password, terminal prints failed to log in	Not able to log in and Display error message too short password, (need 6 characters minimum) terminal prints failed to log in	pass
TC003	An invalid username and valid password	1. Navigate to the LinkedIn login page 2. Enter an invalid username 3. Enter a valid password 4. Click the "Sign in" button	invalid username “sandali@gmail.com” valid password “Pgck13877*”	Should not be able to log in and Display error message invalid username or password, terminal prints failed to log in	not able to log in and Display error message invalid username or password, terminal prints failed to log in	pass

TC004	An invalid username and an invalid password	1. Navigate to the LinkedIn login page 2. Enter an invalid username 3. Enter an invalid password 4. Click the "Sign in" button	invalid username “sandali@gmail.com” valid password “Pgck1367”	Should not be able to log in and Display error message invalid username or password, terminal prints failed to log in	not able to log in and Display error message invalid username or password, terminal prints failed to log in	pass

Conclusion

Here, this project successfully automated the login functionality of the LinkedIn platform using Selenium WebDriver along with Java using intelliJ IDE. By using the open-source automated testing tool selenium can test login functionality by simulating it as a real-world user scenario making it closely relatable to an actual login scenario.

Four different test cases were designed and tested here by using this tool. In there for every test was done by using different input data such as valid/ invalid usernames and passwords. All test scenarios got the same result as expected, therefore it proves that login functionality in LinkedIn works accurately and prevents users from login with invalid credentials, which describes the security side of this platform.

In the management of test scripts, selenium was a great tool and a source as it is compatible with different types of web browsers and a few different languages also IntelliJ was one of the best platforms to use Java in automation testing.

In conclusion, finally we can say that selenium is one of the best open sources for automated testing, it helped to test LinkedIn login functionality likewise it can be used to check many other platforms, and many different functionalities, it is the best toolkit from beginner to pro. This project helped a lot not only to get knowledge on testing automation but also about automation testing tools, test scripts, test designing as well as documentation

References

[1] selenium, "selenium," selenum, 28 08 2024. [Online]. Available: https://www.selenium.dev/. [Accessed 30 08 2024].

[2] intellij, "jetbrains," [Online]. Available: https://www.jetbrains.com/help/idea/discover-intellij-idea.html. [Accessed 30 08 2024].

[3] k. devaraj, "test sigma," 29 july 2024. [Online]. Available: https://testsigma.com/blog/test-case-design-techniques/. [Accessed 30 08 2024].

[4] j. point, "java point," [Online]. Available: https://www.javatpoint.com/selenium-tutorial. [Accessed 2024].

[5] [Sound Recording]. youtube. 2022.