



NORTHWESTERN POLYTECHNIC
UNIVERSITY

Online Jukebox + DOS attack using Selenium JUnit/WebDriver + Python Web Server

Prepared For:

Mr. Henry Chang
Software Quality Assurance and Test Automation CS522
Fall 2021
Northwestern Polytechnic University

Prepared By:

Ms. Nagalla, Santhi Sree ID:19568

Table of Contents

- *Online Jukebox Code*
- *Online Jukebox form*
- *DOS attack using Selenium IDE*
- *DOS attack using Selenium Webdriver*
- *Run webdriver code from command line*
- *Shell script*
- *Cron Job*
- *Cron Job Mail Notice*

Online Jukebox Code

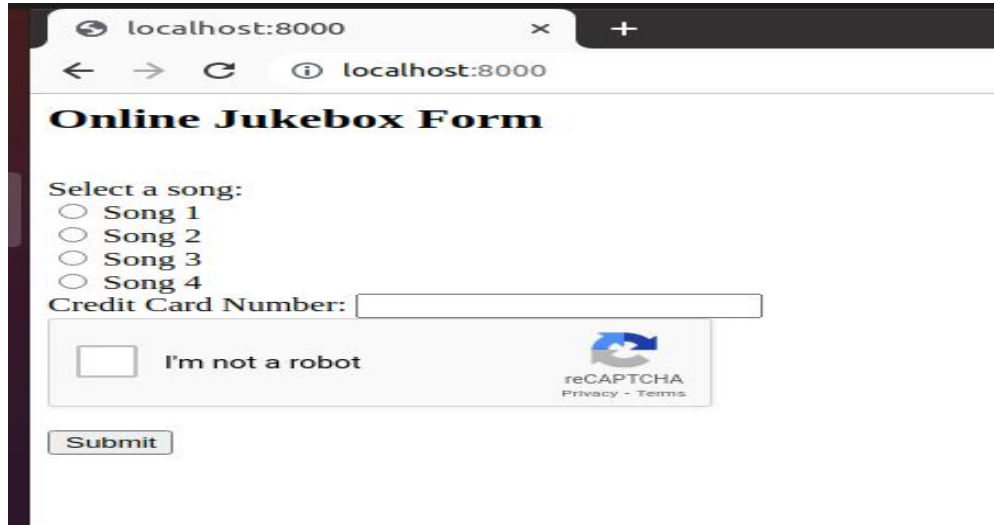
- Jukebox code is available in my Github link -
 - <https://github.com/santhinagalla/Software-Quality-Assurance-and-Test-Automation/tree/main/Jukebox>

Steps to run Code

- Make sure the mysql server up and running
- And start python3 server using below command.
python3 -m http.server 8000 --bind 127.0.0.1 --cgi 8000
- Save and change the mode of the Files using command ,for Ex - `chmod 755 dos.sh`
- Go to <https://www.google.com/recaptcha/admin/create> to create a reCAPTCHA to prevent the attack.
- Copy the client-side key and use in index.html.
- Copy the Server-side key and use in songlist.py.

Online Jukebox form

- Launch the browser and Open `http://localhost:8000/` to see Online Jukebox form.
The browser looks like below -

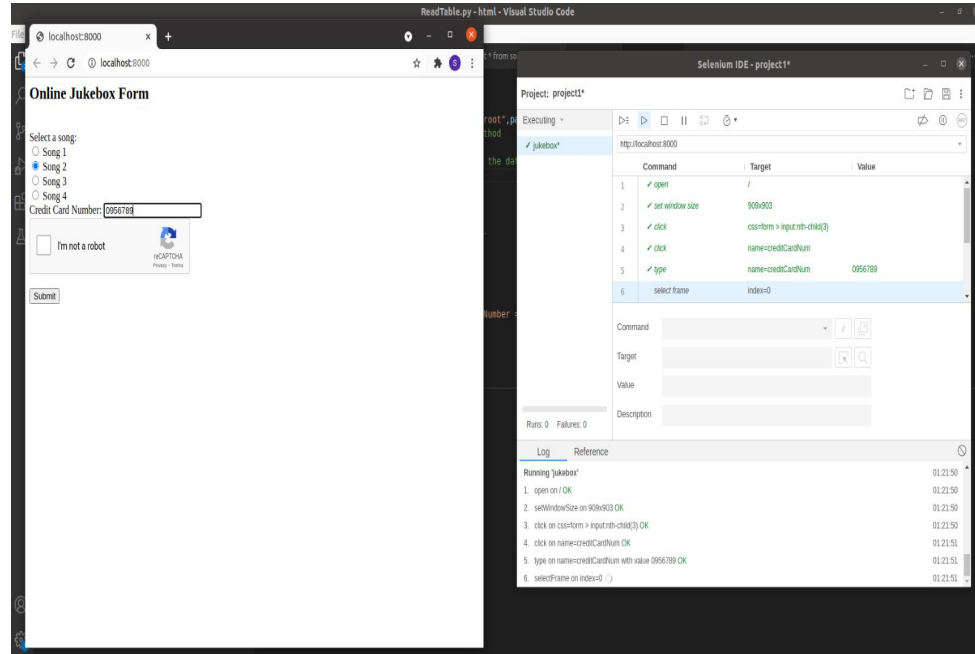


The screenshot shows a web browser window with the address bar displaying `localhost:8000`. The page title is "Online Jukebox Form". The form contains the following elements:

- A heading "Online Jukebox Form" in bold black text.
- A label "Select a song:" followed by four radio button options: "Song 1", "Song 2", "Song 3", and "Song 4".
- A label "Credit Card Number:" followed by a text input field.
- A reCAPTCHA widget with a checkbox labeled "I'm not a robot" and the reCAPTCHA logo.
- A "Submit" button at the bottom left.

DOS attack using Selenium IDE

- Open IDE and record the steps by using the Base URL - **http://localhost:8000/** to test Online Jukebox form.
- Select song and enter credit card number then select the simple captcha checkbox then click on submit button.
- Once you are done the recording, stop the recording and playback and check whether the test case running successfully.



Project: project1*

Tests ▾ + ▶ ▶ ⌂ ⌚ ▾

Search tests... Run current test Ctrl+R ▾

✓ jukebox*	Command	Target	Value
5	✓ type	name=creditCardNum	0956789
6	✓ select frame	index=0	
7	✓ click	css=.recaptcha-checkbox-border	
8	✓ select frame	relative=parent	
9	✓ click	id=button1	
10	✓ close		

Command //

Target

Value

Description

Log Reference

5. type on name=creditCardNum with value 0956789 OK 01:21:51

6. selectFrame on index=0 OK 01:21:51

7. click on css=.recaptcha-checkbox-border OK 01:21:52

8. selectFrame on relative=parent OK 01:21:52

9. click on id=button1 OK 01:21:52

10. close OK 01:21:53

'jukebox' completed successfully 01:21:53

```
[1] ... python3 ReadTable.py  
sans@sans-vb:~/Desktop/html$ python3 ReadTable.py  
songName = song1, Credit Card Number = 123456789  
songName = song2, Credit Card Number = 987654321
```

```
songName = song2, Credit Card Number = 0956789  
songName = song2, Credit Card Number = 0956789  
songName = song2, Credit Card Number = 0956789
```

DOS attack using Selenium Webdriver

- Write test cases using “Selenium JUnit/WebDriver” to access the Jukebox form or export the code in “Java JUnit” using Selenium IDE.

```
1 import org.junit.Test;
2 import org.junit.Before;
3 import org.junit.After;
4 //import static org.junit.Assert.*;
5 //import static org.hamcrest.CoreMatchers.is;
6 //import static org.hamcrest.core.IsNot.not;
7 import org.openqa.selenium.By;
8 import org.openqa.selenium.WebDriver;
9 import org.openqa.selenium.firefox.FirefoxDriver;
10 import org.openqa.selenium.chrome.ChromeDriver;
11 import org.openqa.selenium.chrome.ChromeDriverService;
12 import org.openqa.selenium.remote.RemoteWebDriver;
13 import org.openqa.selenium.remote.DesiredCapabilities;
14 import org.openqa.selenium.Dimension;
15 import org.openqa.selenium.WebElement;
16 import org.openqa.selenium.interactions.Actions;
17 import org.openqa.selenium.support.ui.ExpectedConditions;
18 import org.openqa.selenium.support.ui.WebDriverWait;
19 import org.openqa.selenium.JavascriptExecutor;
20 import org.openqa.selenium.Alert;
21 import org.openqa.selenium.Keys;
22 import java.util.*;
23 import java.net.MalformedURLException;
24 import java.net.URL;
25 public class Jukeboxselenium {
26     private WebDriver driver;
27     private Map<String, Object> vars;
28     JavascriptExecutor js;
29     @Before
30     public void setUp() {
31         System.setProperty(ChromeDriverService.CHROME_DRIVER_EXE_PROPERTY, "/home/sans/Downloads/jar/chromedriver");
32         driver = new ChromeDriver();
33         js = (JavascriptExecutor) driver;
34         vars = new HashMap<String, Object>();
35     }
36     @After
37     public void tearDown() {
38         driver.quit();
39     }
40     @Test
41     public void test1() {
42         try{
43             driver.get("http://localhost:8000/");
44             driver.manage().window().setSize(new Dimension(909, 903));
45             driver.findElement(By.name("song")).click();
46             driver.findElement(By.name("creditCardNum")).click();
47             driver.findElement(By.name("creditCardNum")).sendKeys("1111111123");
```

Run webdriver code from command line

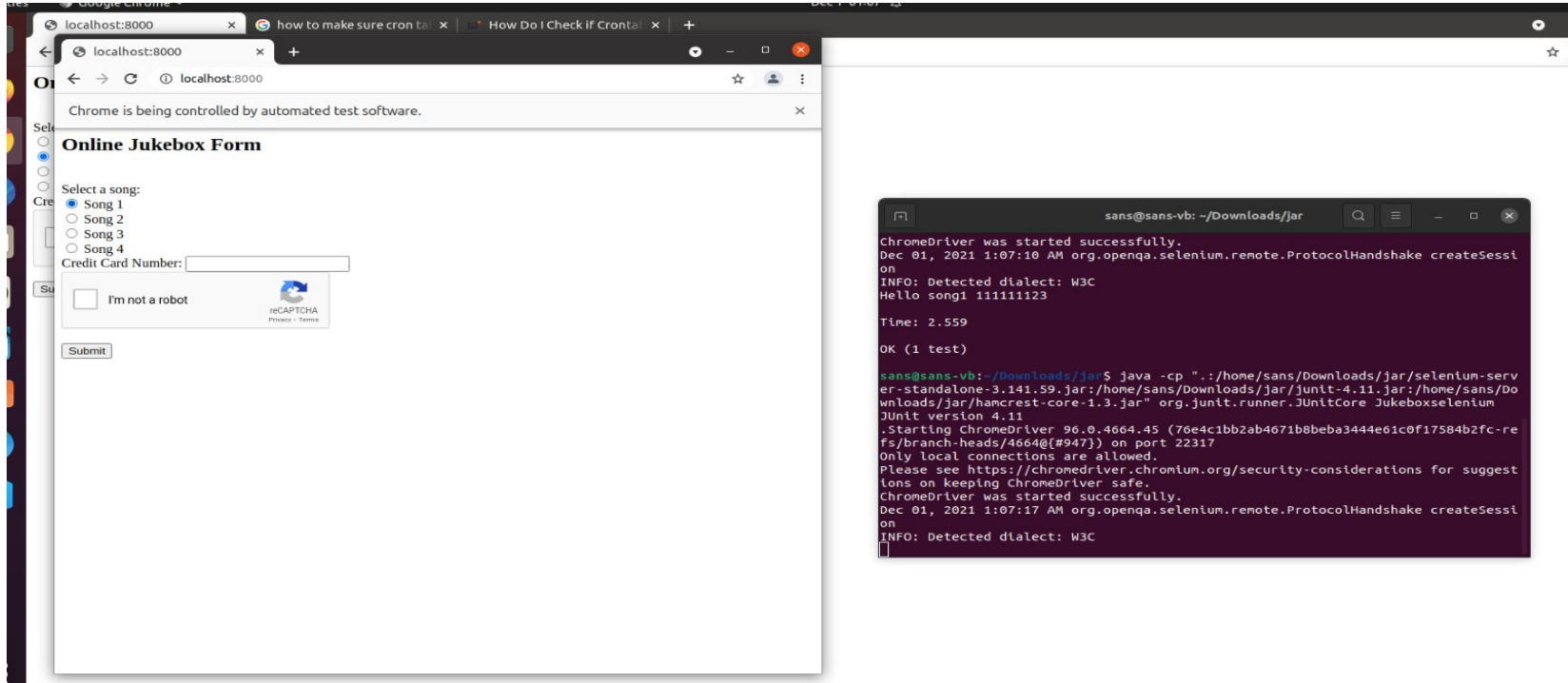
- To run the above code in command line need below Jar files.
- Download the Jar files before execute the code in command line.
 - **Selenium Standalone server** - <https://selenium-release.storage.googleapis.com/index.html?path=3.141/selenium-server-standalone-3.141.59.jar>
 - **Chrome Driver** - <https://chromedriver.chromium.org/downloads>
 - **junit-4.11** - <http://www.java2s.com/Code/Jar/j/Downloadjunit411jar.htm>
 - **hamcrest-core-1.3** - <http://www.java2s.com/Code/Jar/h/Downloadhamcrestcore13jar.htm>

Run webdriver code from command line

- Compile the jukebox code using below commands with jar files.
- .class file will be generated.
 - **javac -cp**
"./:/home/sans/Downloads/jar/selenium-server-standalone-3.141.59.jar:/home/sans/Downloads/jar/junit-4.11.jar:/home/sans/Downloads/jar/hamcrest-core-1.3.jar" -d . Jukeboxselenium.java
- Now run the .class file using below command.
 - **java -cp**
"./:/home/sans/Downloads/jar/selenium-server-standalone-3.141.59.jar:/home/sans/Downloads/jar/junit-4.11.jar:/home/sans/Downloads/jar/hamcrest-core-1.3.jar" org.junit.runner.JUnitCore Jukeboxselenium

Run webdriver code from command line

- After running the above commands browser will launch and execute the steps.



Output

```
sans@sans-vb: ~/Downloads/jar
Time: 2.559
OK (1 test)

sans@sans-vb:~/Downloads/jar$ java -cp ".:~/home/sans/Downloads/jar/selenium-server-standalone-3.141.59.jar:~/home/sans/Downloads/jar/junit-4.11.jar:~/home/sans/Downloads/jar/hamcrest-core-1.3.jar" org.junit.runner.JUnitCore Jukeboxselenium
JUnit version 4.11
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{#947}) on port 22317
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Dec 01, 2021 1:07:17 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Hello song1 111111123

Time: 2.745
OK (1 test)

sans@sans-vb:~/Downloads/jar$
```

Shell script

- Write a shell script to invoke your java program with the necessary arguments : Ex- Juke.sh

```
#!/bin/bash
```

```
echo "Running script."
```

```
cd /home/sans/Downloads/jar
```

```
#javac -cp
```

```
".:/home/sans/Downloads/jar/selenium-server-standalone-3.141.59.jar:/home/sans/Downloads/jar/junit-4.11.jar:/home/sans/Downloads/jar/hamcrest-core-1.3.jar" -d .
```

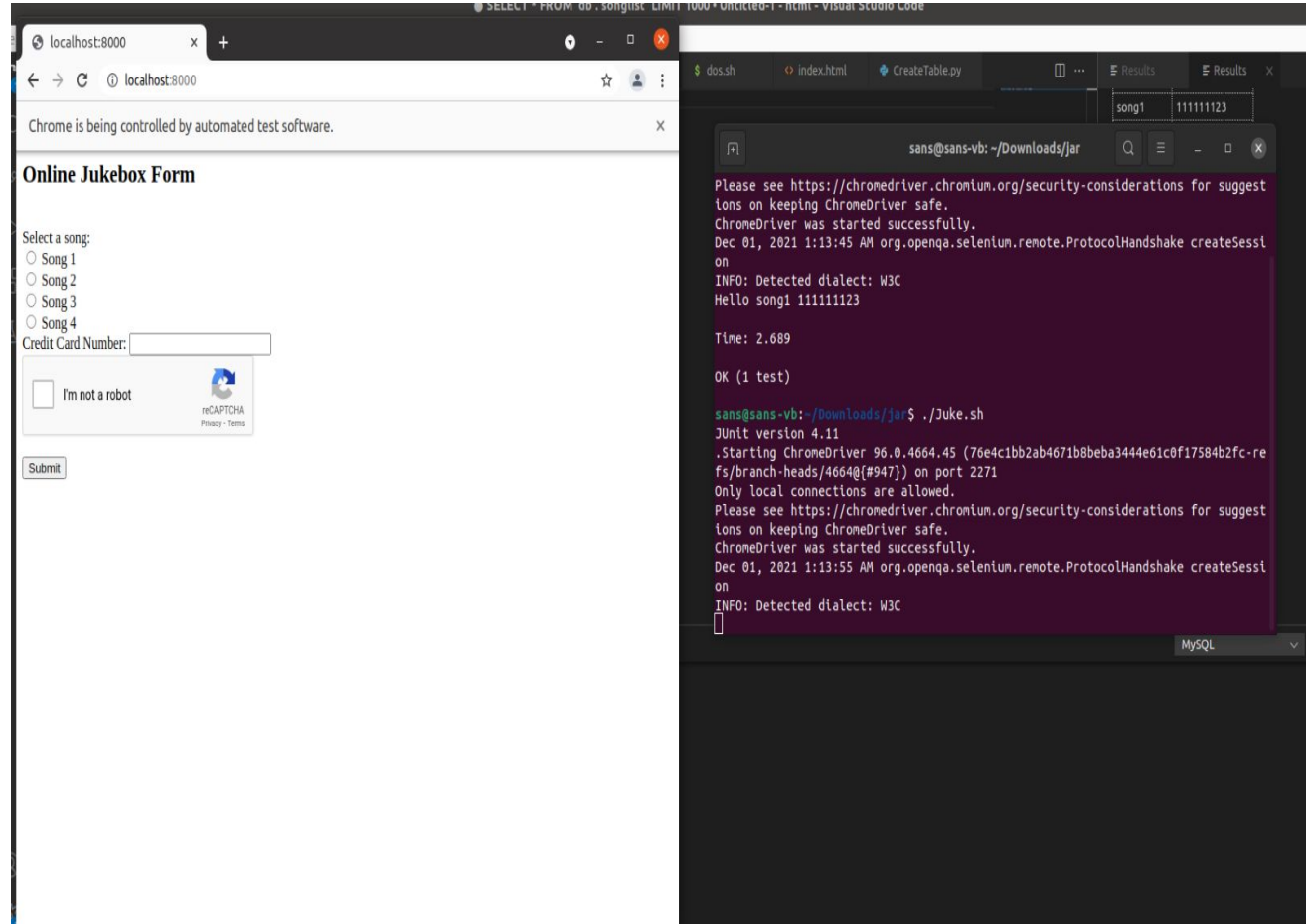
```
Jukeboxselenium.java
```

```
java -cp
```

```
".:/home/sans/Downloads/jar/selenium-server-standalone-3.141.59.jar:/home/sans/Downloads/jar/junit-4.11.jar:/home/sans/Downloads/jar/hamcrest-core-1.3.jar
```

```
org.junit.runner.JUnitCore Jukeboxselenium
```

Run the Shell script
on command line
`./Juke.sh`



Output - Shell script

```
sans@sans-vb: ~/Downloads/jar
Hello song1 111111123
Time: 2.689
OK (1 test)

sans@sans-vb:~/Downloads/jar$ ./Juke.sh
JUnit version 4.11
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-re
fs/branch-heads/4664@{#947}) on port 2271
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggest
ions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Dec 01, 2021 1:13:55 AM org.openqa.selenium.remote.ProtocolHandshake createSessi
on
INFO: Detected dialect: W3C
Hello song1 111111123
Time: 2.794
OK (1 test)

sans@sans-vb:~/Downloads/jar$
```

Cron Job

- Schedule the script to be invoked by setting up a cron job.
- Run from a terminal: **crontab -e**
- This will open your crontab editor.

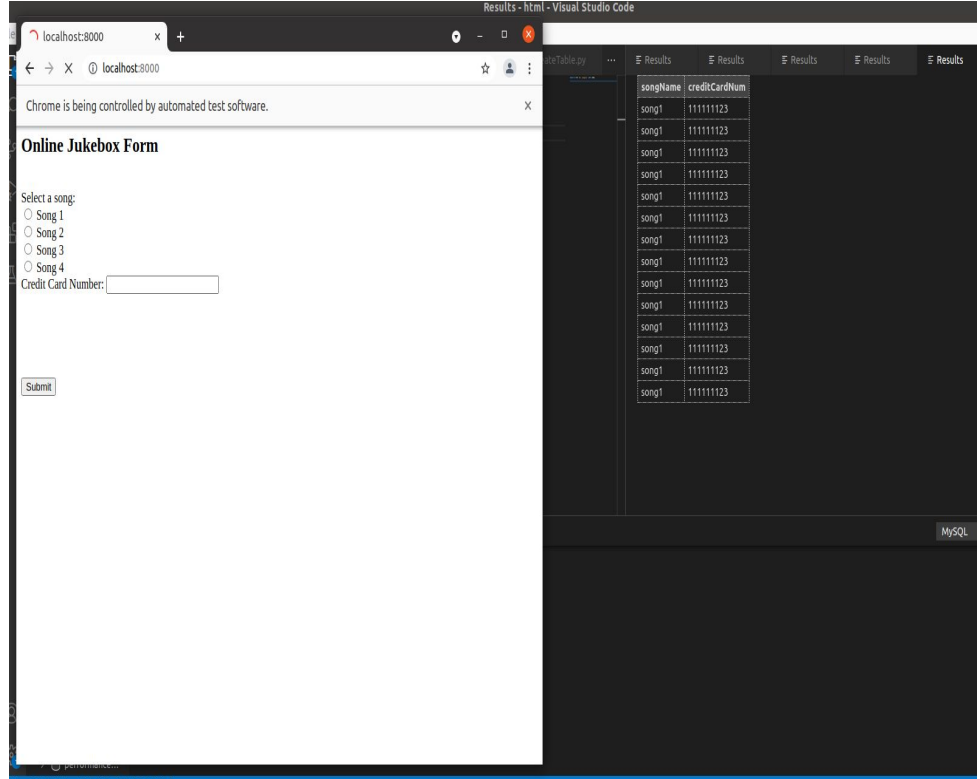
You can add a job in this way:

MAILTO="santhi@gmail.com"

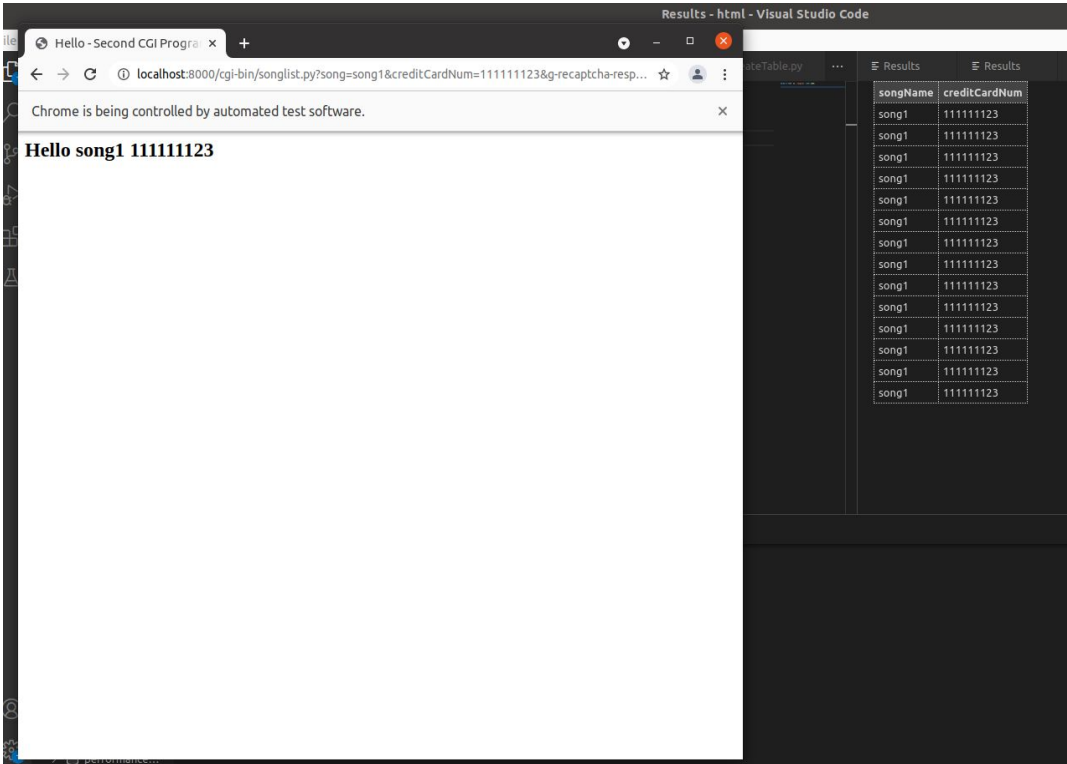
```
*/5 * * * * bash
```

`/home/sans/Downloads/jar/Juke.sh`

- This job is set to run every 5 minutes.



Cron Job Mail Notice



The screenshot shows a web browser window on the left and a Visual Studio Code editor on the right. The browser window displays the output of a CGI program: "Hello song1 111111123". The Visual Studio Code editor shows a file named "Results - html" with the following content:

songName	creditCardNum
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123
song1	111111123

Cron <sans@sans-vb> bash /home/sans/Downloads/jar/Juke.sh 



root <sans.nagalla@gmail.com>

to me ▾

Running script.

JUnit version 4.11

.Dec 01, 2021 11:18:02 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Hello song1 111111123

Time: 4.899

OK (1 test)

```
NO MODIFICATION MADE
sans@sans-vb:~$ crontab -r
sans@sans-vb:~$ crontab -l
no crontab for sans
sans@sans-vb:~$
```


References

- Github Link - [https://github.com/santhinagalla/Software-Quality-Assurance-and-Test-Automation/tree/main/QA/Selenium%20JUnit\(WebDriver%20%2B%20Python%20Web%20Server%20%2B%20Cron%20Job\)/JukeBox](https://github.com/santhinagalla/Software-Quality-Assurance-and-Test-Automation/tree/main/QA/Selenium%20JUnit(WebDriver%20%2B%20Python%20Web%20Server%20%2B%20Cron%20Job)/JukeBox)
- https://npu85.npu.edu/~henry/npu/classes/shell_script/backup/slide/cron.html
- <https://unix.stackexchange.com/questions/98914/how-to-use-crontab-for-a-java-file-in-linux>
- <https://stackoverflow.com/questions/7855666/cron-job-for-a-java-program>
- <https://www.browserstack.com/guide/selenium-webdriver-tutorial>