Rex Jones II

Selenium 4 How To Take WebElement & Full Page Screenshot

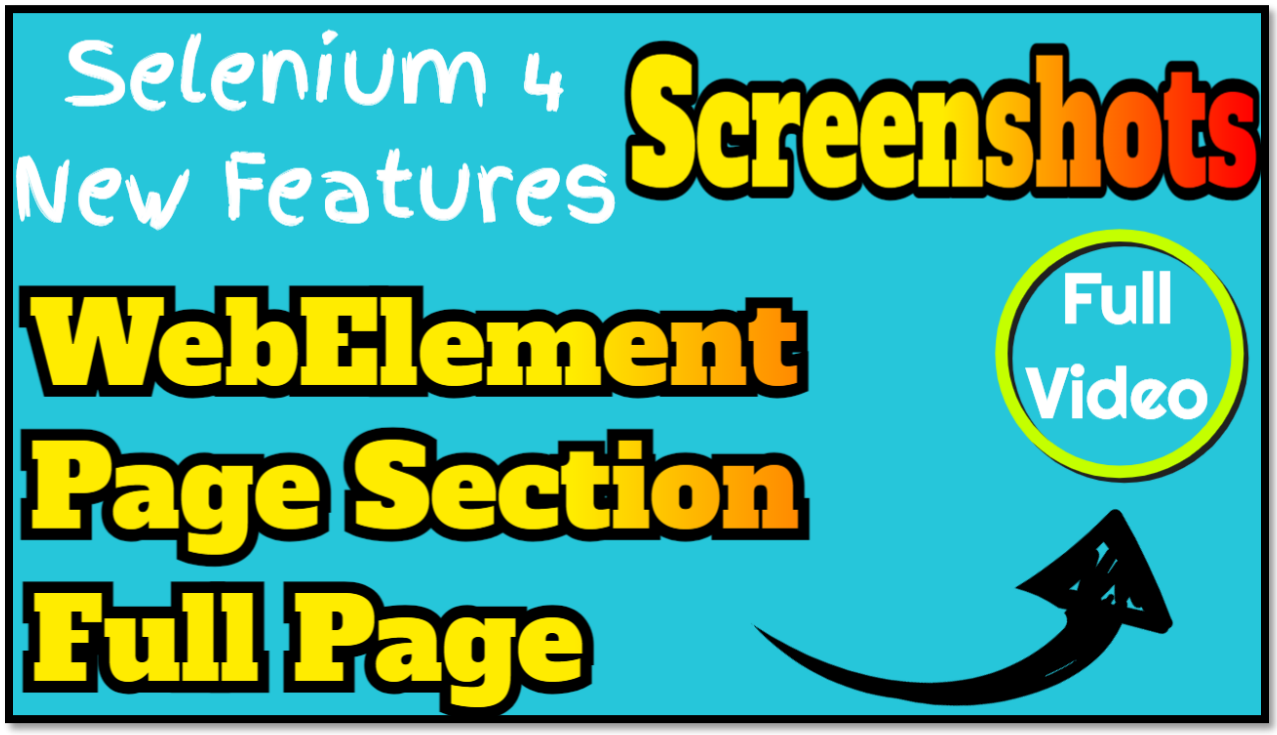


Table of Contents

Video Link

https://youtu.be/mIXhrlMmBs8

WebElement Screenshot

In this session, we are going to take a screenshot of a WebElement. Until now, Selenium already had a feature to take a screenshot of a page but not to take a screenshot of a WebElement. Before Selenium 4, we had to use an API called Ashot to take a screenshot of a WebElement. However, in this session I am going to demo how to take a screenshot of 1 WebElement and page section that includes more than 1 WebElement.

If you are interested in more videos, you can subscribe to my YouTube channel then click the bell icon. Follow me on Twitter and connect with me on LinkedIn and Facebook. Also, I'm going to place the transcript and code on GitHub.

We are going to use this Orange HRM site. I will take a screenshot of this logo for 1 WebElement. Do you see how this part of the page has more than 1 WebElement? We see Username, Password, Login Button, and the Forgot Password link plus this Orange image? I'm going to take screenshot of this entire section using Selenium 4. Let's start with the logo and Inspect then find the value by writing #divLogo. This is the id value for the parent > img is the tag name for the logo. Let's start by taking a screenshot of the WebElement logo.

One WebElement Screenshot

We have our test setup to load Chrome and the AUT.

```
@BeforeClass
public void setUp () {
 WebDriverManager.chromedriver().setup();
 driver = new ChromeDriver();
 driver.manage().window().maximize();
 driver.get("https://opensource-demo.orangehrmlive.com/");
}

@AfterClass
public void tearDown () {
 driver.quit();
}
```

Now, let's take a screenshot of the logo by writing @Test / public void takeWebElementScreenshot () { } Let's find the WebElement by writing WebElement logoOrangeHRM =

driver.findElement(By.cssSelector("#divLogo > img")); Import the WebElement and @Test annotation from TestNG. Now that we have the logo for OrangeHRM logoOrangeHRM., we get the screenshotAs(OutputType.FILE). The getScreenshotAs method Capture the screenshot and store it in the specified location. Notice the return type is File. Let's hover the OutputType and it defines the output type for a screenshot. FILE is used to obtain the screenshot into a temporary file. Therefore, we assign the screenshot to a File and name it source =.

Now we need a File for our destination = new File called ("Orange HRM Logo.png"); We have our source file and destination file. A class called FileUtils. has a method to copyFile. The description shows it copies a file to a new location. So, for the source file we are going to have source and the destination file we are going to have our destination. Depending on how your Selenium 4 is setup. You may have to download the Apache Commons IO jar from Maven's Repository to get this FileUtils class then import the jar. It's also a dependency hierarchy for WebDriverManager. Last step is to add the throws declaration for IOException.

```
@Test
Run | Debug
public void takeWebElementScreenshot () throws IOException {
  WebElement logoOrangeHRM = driver.findElement(By.cssSelector("#divLogo > img"));
  File source = logoOrangeHRM.getScreenshotAs(OutputType.FILE);
  File destination = new File ("Orange HRM Logo.png");
  FileUtils.copyFile(source, destination);
}
```

That's it. Let's Run. We see the screenshot in our project. I setup Eclipse to auto refresh so I did not have to manually right click the project and select F5. Video 72 will show you how to Auto Refresh your project automatically. Right click, select Open, and here's the logo WebElement screenshot.

Next is to take a screenshot of a page section.

Page Section Screenshot

In a section, there are multiple WebElements so we need to find the parent tag of those WebElements. Inspect this section and we see the parent tag has an id value of divLoginImage. On this page, another section is the Social Media icons that can also be used to take a screenshot of multiple WebElements. All we need to find is the parent tag and Selenium will take a screenshot of all 4 Social Media icons.

In this example, social-icons is the parent tag. Go back to our IDE and write our test for the page section. So, I'm going to write @Test / public void takePageSectionScreenshot () { } The page section is a WebElement so we write WebElement pageSectionOrangeHRM = driver.findElement(By.id("divLoginImage")); Next is to store the screenshot in a file. Therefore, we write File source = What is the source? The source pageSectionOrangeHRM.getScreenshotAs(OutputType.FILE);

This statement is an abstract representation of the File class and will save the screenshot in memory. We need a physical file since this is an abstract representation. Therefore, we copy the source file to the destination. I'm going to use these 2 statements and combine them into 1 line. So, I'm going to write FileUtils.copyFile(source,). The destination file will contain the name of the new File("Orange HRM Page Section.png"). You can also use .jpg file but I like to use png. This statement will convert the abstract source file into a physical .png file so we can view the screenshot. Add the throws declaration and Run. We see the Page Section screenshot. Let's open and all of the WebElements are located in this screenshot.

That's it and Thanks for watching.

In this session, I am going to take a full-page screenshot using Selenium 4. Right now, it's only available for Firefox. There's a difference between the full-page screenshot and the existing screenshot feature. The existing screenshot feature takes a screenshot of the page we can see in our web page. However, the full-page screenshot takes a screenshot of the entire page whether we can see it or not.

I create videos every week and if you are interested in more videos, feel free to connect with me on LinkedIn and Facebook. You can also follow me on Twitter and GitHub. If you're on YouTube, you can subscribe to my YouTube channel and click the bell icon. I'll make sure to place the transcript and code on GitHub.

Our AUT is going to be Amazon. On this page, we see it's very long and has a lot of products. Scroll and scroll and we see more and more. Now, let's go to our code.

Full Page Screenshot

We have a setup method for Amazon using FirefoxDriver and a tear down method to quit the driver.

```
@BeforeTest
public void setUp () {
 WebDriverManager.firefoxdriver().setup();
 driver = new FirefoxDriver ();
 driver.manage().window().maximize();
 driver.get("https://www.amazon.com/");
}

@AfterTest
public void tearDown () {
 driver.quit();
}
```

TakesScreenshot Interface

Start by writing @Test / public void takeFullPageScreenshot () { }. The 1st step is to capture the screenshot using ((FirefoxDriver)driver). We are casting FirefoxDriver so this statement is a downcast. The plan is to use driver which is a variable of type WebDriver. We are going to use driver to call a method that's only available to FirefoxDriver and that method is getFullPageScreenshotAs(OutputType.FILE); Let's save the FILE and name it source =. Import the @Test Annotation from TestNG and File.

If you want, feel free to separate this same statement into 2 statements. However, I'm going continue getting straight to the point. Also, with the next statement we can separate into 2 separate statements but I will write 1 line. Let's handle the file with FileHandler.copy(source,). The destination is a new File() named ("Amazon Full Page Screenshot.png"); That's all we write to capture the full page. Add a throws declaration and select IOException.

```
@Test
Run | Debug
public void takeFullPageScreenshot () throws IOException {
  File source = ((FirefoxDriver)driver).getFullPageScreenshotAs(OutputType.FILE);
  FileHandler.copy(source, new File("Amazon Full Page Screenshot.png"));
}
```

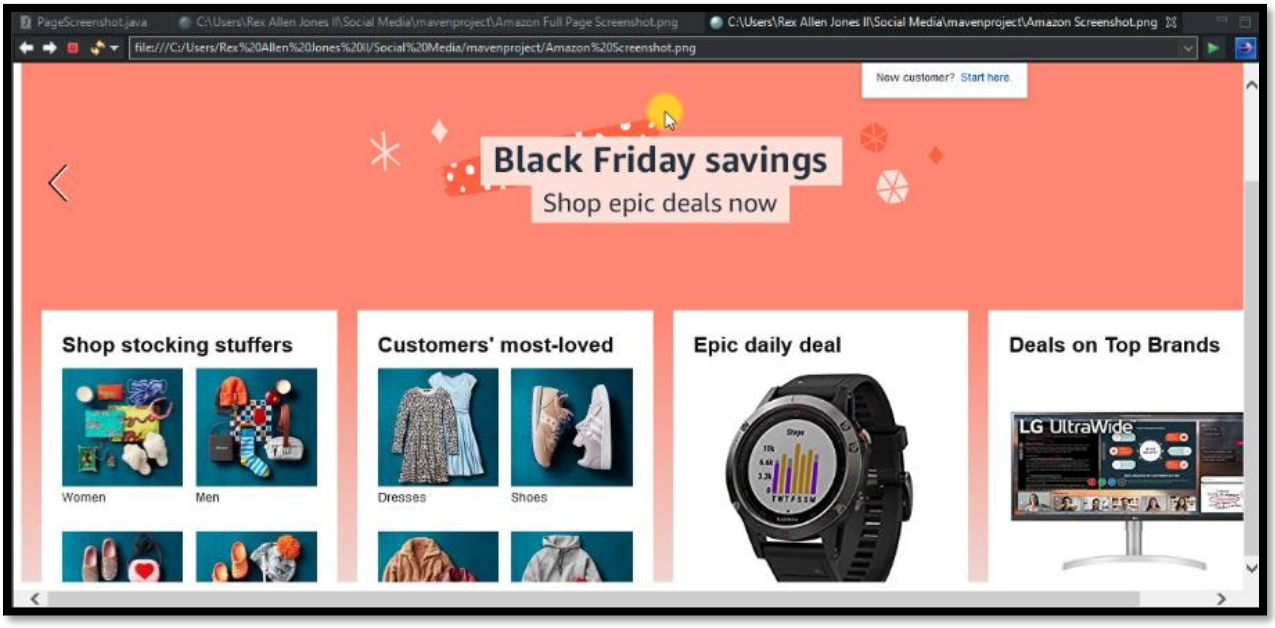
Let me also capture a screenshot but not the full page. It won't take long. I want to show you the difference between full page screenshot and the existing screenshot feature.

@Test / public void takePageScreenshot () { } It's very similar to the full page screenshot but I will write TakesScreenshot and not FirefixDriver. Alright, File source = then we cast ((TakesScreenhot)driver). Import TakesScreenshot from Selenium package and .screenshot notice we only see getScreenshotAs but we do not see getFullPageScreenshotAs. That's how we know getFullPageScreenshotAs is only available for the FirefoxDriver. Select getScreenshotAs(OutputType.FILE). Next, is File CTRL + SPACE. We have the option of using FileHandler or FileUtils. FileHandler is a class from Selenium and FileUtils is a class from apache.commons. I'm going to use FileUtils this time since I used FileHandler in the previous Test Script and select copyFile(source, new File("Amazon Screenshot.png")); Import throws declaration and we are going to select IOException.

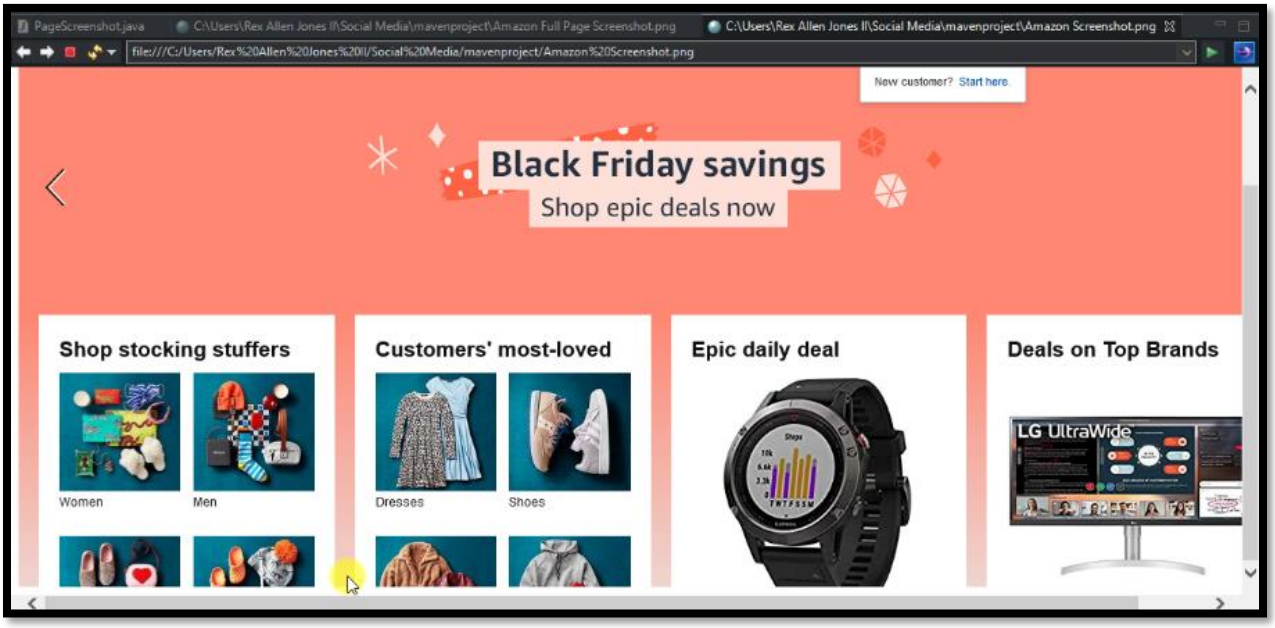
```
@Test
Run | Debug
public void takePageScreenshot () throws IOException {
  File source = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
  FileUtils.copyFile(source, new File("Amazon Screenshot.png"));
}
```

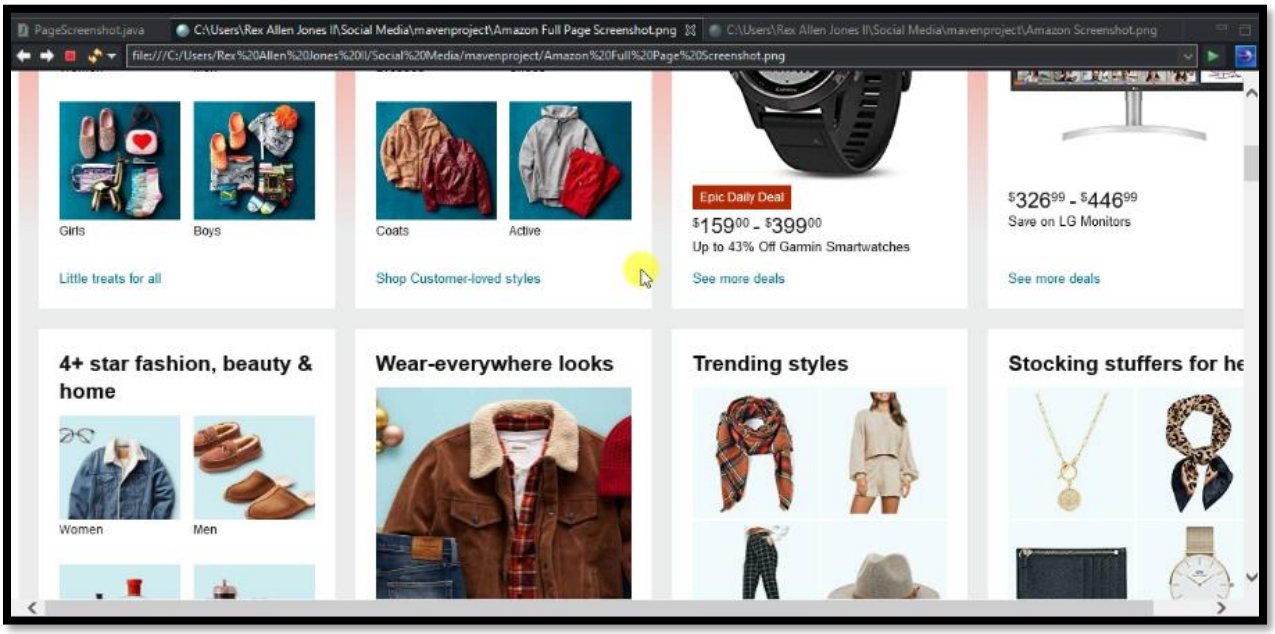
I'm going to save it and this time Run All and not an individual Test Script. We see both screenshots: Amazon Full Page Screenshot and Amazon Screenshot. Like I mentioned in the previous video, I set it up where it refreshes automatically.

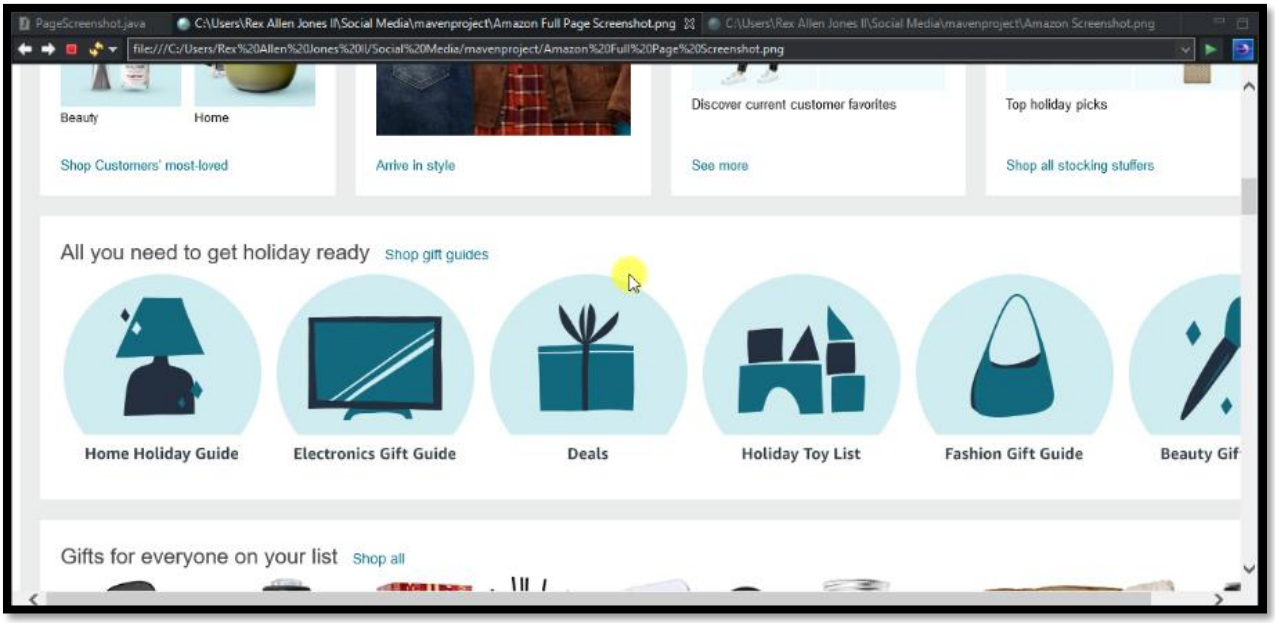
This here is the screenshot which shows what we only saw in the webpage and it stops around this point.

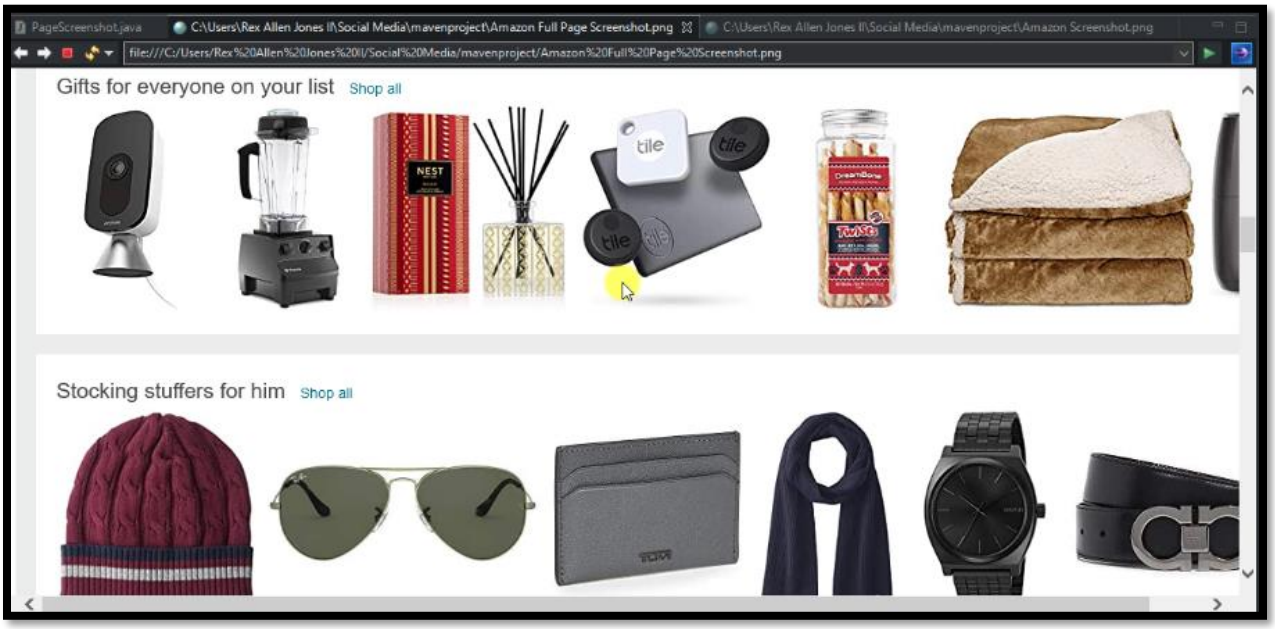


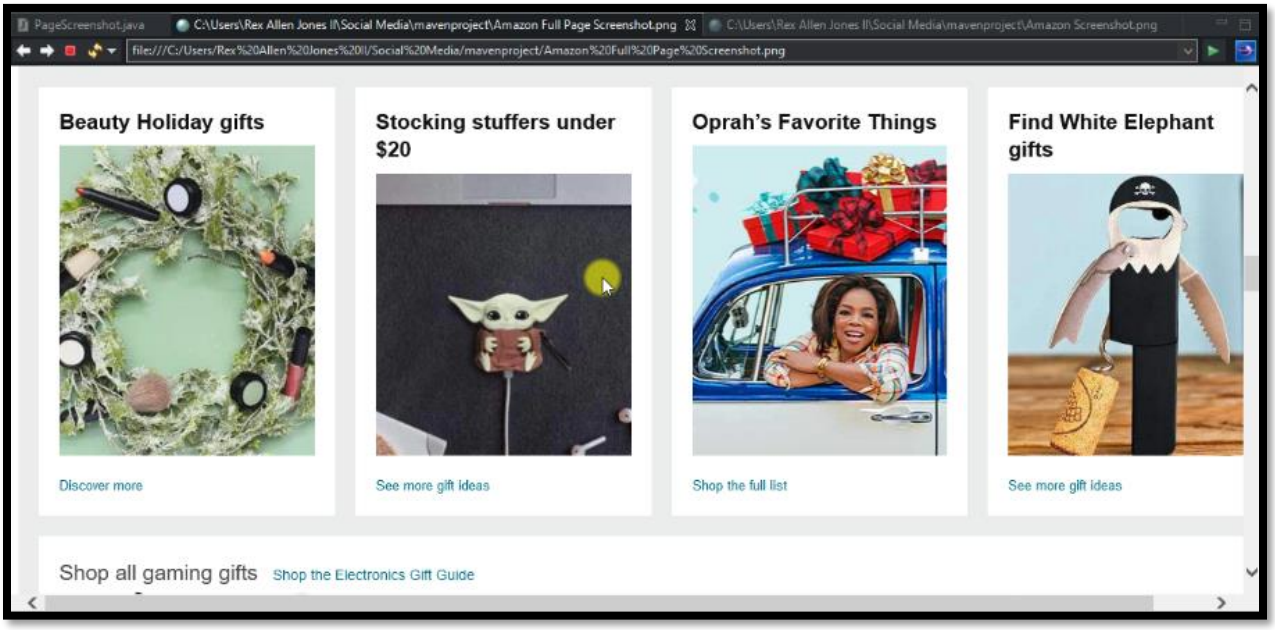
Select the other screenshot and we keep on scrolling and the full-page screenshot took a screenshot of the complete page from top to bottom but not including the tab and address bar. Taking a normal screenshot only captured what we see in the browser page. That's it and thank you for watching and I'll see you in the next session.

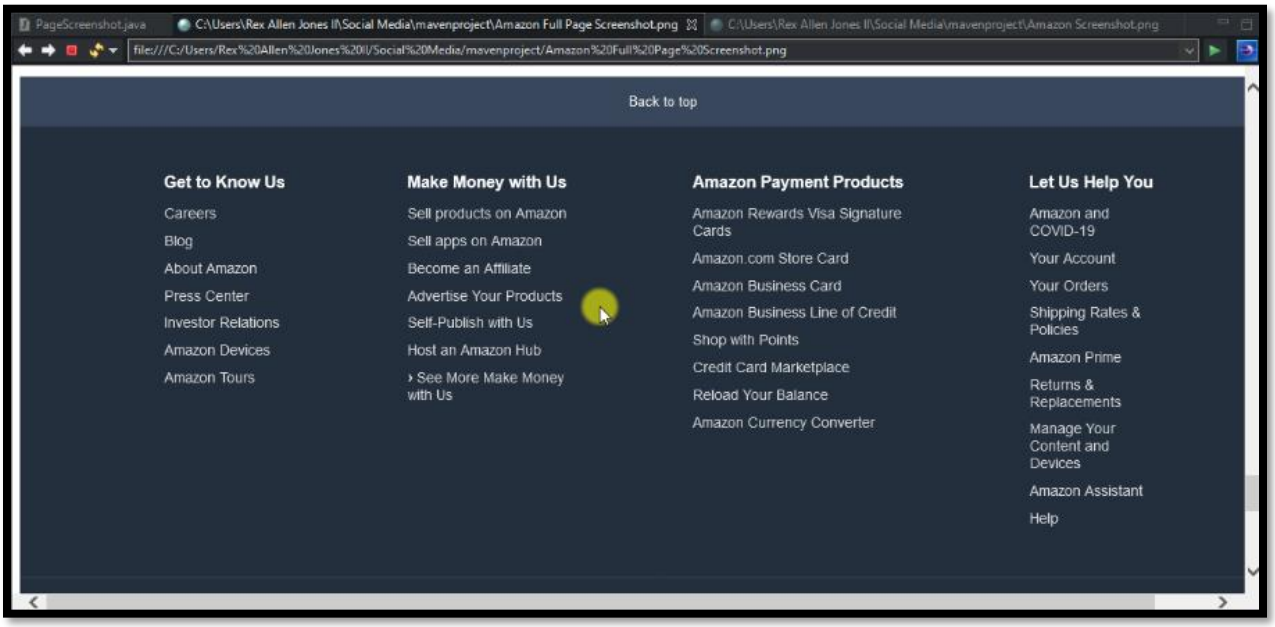

PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Screenshot.png
New customer? Start here.
Black Friday savings
Shop epic deals now
Shop stocking stuffers
Women
Men
Customers' most-loved
Dresses
Shoes
Epic daily deal
Deals on Top Brands
LG UltraWide


PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Full%20Page%20Screenshot.png
Girls
Boys
Little treats for all
Coats
Active
Shop Customer-loved styles
Epic Daily Deal
$159.00 - $399.00
Up to 43% Off Garmin Smartwatches
See more deals
$326.99 - $446.99
Save on LG Monitors
See more deals
4+ star fashion, beauty & home
Women
Men
Wear-everywhere looks
Trending styles
Stocking stuffers for he


PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Full%20Page%20Screenshot.png
Beauty
Home
Shop Customers' most-loved
Arrive in style
Discover current customer favorites
See more
Top holiday picks
Shop all stocking stuffers
All you need to get holiday ready
Shop gift guides
Home Holiday Guide
Electronics Gift Guide
Deals
Holiday Toy List
Fashion Gift Guide
Beauty Gif
Gifts for everyone on your list
Shop all


PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Full%20Page%20Screenshot.png
Gifts for everyone on your list
Shop all
NEST
tile
Stocking stuffers for him
Shop all


PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Full%20Page%20Screenshot.png
Beauty Holiday gifts
Discover more
Stocking stuffers under $20
See more gift ideas
Oprah's Favorite Things
Shop the full list
Find White Elephant gifts
See more gift ideas
Shop all gaming gifts
Shop the Electronics Gift Guide


PageScreenshot.java
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Full Page Screenshot.png
C:\Users\Rex Allen Jones II\Social Media\mavenproject\Amazon Screenshot.png
file:///C:/Users/Rex%20Allen%20Jones%20II/Social%20Media/mavenproject/Amazon%20Full%20Page%20Screenshot.png
Back to top
Get to Know Us
Careers
Blog
About Amazon
Press Center
Investor Relations
Amazon Devices
Amazon Tours
Make Money with Us
Sell products on Amazon
Sell apps on Amazon
Become an Affiliate
Advertise Your Products
Self-Publish with Us
Host an Amazon Hub
› See More Make Money with Us
Amazon Payment Products
Amazon Rewards Visa Signature Cards
Amazon.com Store Card
Amazon Business Card
Amazon Business Line of Credit
Shop with Points
Credit Card Marketplace
Reload Your Balance
Amazon Currency Converter
Let Us Help You
Amazon and COVID-19
Your Account
Your Orders
Shipping Rates & Policies
Amazon Prime
Returns & Replacements
Manage Your Content and Devices
Amazon Assistant
Help

Rex Jones II

Contact Info

- ✔ Email Rex.Jones@Test4Success.org
- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook http://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/