Lab Key: Automating 113 Attendance Check

Every class in 113, the TAs make a TinyURL that will link you to the daily attendance form. There, you fill out your AndrewID and submit to confirm attendance, and add comments if you feel like it. This demo will pull up the form, fill out your AndrewID, then submit the form with no additional comments. The only modification necessary for different forms is to change the TinyURL link. So far, you should have the following code in your file:

```
from selenium import webdriver

PATH = “<copy path here>”
driver = webdriver.Chrome(PATH)
```

For this demo, we are going to want to import some other things. Copy and paste the following code under the webdriver import line.

```
from selenium.webdriver.common.by import By
import time
```

Importing **By** allows us to search for elements and is required. Importing **time** is to make results easier to view.

The first thing we need to do is to find the link for the attendance form. All links for 113 attendance have the form of “https://tinyurl.com/” followed by a word or phrase related to the day’s lecture. Thus, we can separate the link into two parts: the constant tinyurl part and the variable word. The link for the February 7th, 2023 lecture will use the phrase “selenium-test”. Using driver.get(“<link here>”), we will open the Google form. Below is the code we used. We defined the demo here to be in the function attendanceFiller, taking in the variable word as an argument.

```
def attendanceFiller(dailyPhrase):
     tinyURL = “https://tinyurl.com/”

     driver.get(tinyURL + dailyPhrase)
```

After we have our link, we define what exactly we want to do with the form. We want to (a) fill out the AndrewID section of the form with our AndrewID, and (b) submit the form.

In our code we defined a variable to be our AndrewID.

```
andrewID = “<insert AndrewID>”
```

Now, to complete (a), we must find the HTML element of the AndrewID input box. Using the inspect tool to find the XPath of the input box, we can store it in a variable, then create the element:

```
textBoxXPATH =
"/html/body/div/div[2]/form/div[2]/div/div[2]/div[1]/div/div/div[2]/div/div[1]/div/div[1]/input"

textBox = driver.find_element(By.XPATH, textBoxXPATH)
```

Now, we can send the keys to the input box.

```
textBox.send_keys(andrewID)
```

Using this loop through every possible input element, the correct input box will be found, selected, and have your AndrewID inputted!

To end the demo, we want to find the submit button and click it. To do this, we found the button using its XPath. We used the inspect tool to find it. As a tip, CTRL + F “span”. As you go through the instances, the inspect tool will highlight the element that corresponds to the instance. We found the path, and stored it in an element buttonPath.

```
buttonPath =
‘/html/body/div/div[2]/form/div[2]/div/div[3]/div[1]/div[1]/div/span/span’
```

Then, we can click the button.

```
submitButton = driver.find_element(By.XPATH, buttonPath)
submitButton.click()
```

And now, we have an automated 113 Attendance filler! The full code is shown below. time.sleep() commands are to give you time to see what is happening, and can be removed.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
```

```
PATH = "C:\Program Files (x86)\chromedriver_win32\chromedriver.exe"
driver = webdriver.Chrome(PATH)

def attendanceFiller(dailyPhrase):

    tinyURL = 'https://tinyurl.com/'

    driver.get(tinyURL + dailyPhrase)
    time.sleep(2)

    buttonPath = "/html/body/div/div[2]/form/div[2]/div/div[3]/div[1]/div[1]/div/span/span"

    andrewID = "andrewID"

    textBoxXPATH = "/html/body/div/div[2]/form/div[2]/div/div[2]/div[1]/div/div/div[2]/div/div[1]/div/div[1]/input"

    textBox = driver.find_element(By.XPATH, textBoxXPATH)

    textBox.send_keys(andrewID)

    submitButton = driver.find_element(By.XPATH, buttonPath)
    submitButton.click()
    time.sleep(2)

    driver.quit()

attendanceFiller(“selenium-test”)
```