~\Downloads\florence final.py

2/22/24, 4:13 PM

```
1
    # Imports and global declarations
 2
    import re
 3
    import zipfile
 4
    import time
 5
    import datetime
 6
    from zipfile import ZipFile as zipfile
 7
    from selenium import webdriver
 8
    from selenium.webdriver.common.keys import Keys
 9
    from selenium.webdriver.common.by import By
    from selenium.webdriver.support.ui import WebDriverWait
10
11
    from selenium.webdriver.support import expected_conditions as EC
    from selenium.common.exceptions import StaleElementReferenceException
12
13
    import time
14
    import csv
15
    import os
    import glob
16
17
    import pandas as pd
    from datetime import date
18
19
    from selenium.webdriver.common.action chains import ActionChains
20
    import getpass
21
    #need to change login stuff
22
    import warnings
23
24
    driver = webdriver.Chrome()
25
    driver.maximize_window()
26
    # driver.get("https://login.uatv2.researchbinders.com/u/login/identifier?state=
    hKFo2SBtZFFQa1FXRVVXNm93U1dFNWxyYTFpX0tGTGxUSkc4c6Fur3VuaXZlcnNhbC1sb2dpbqN0aWTZ
    IFlod2RDbV9rV19ja194VHVNQ1JTSUJŹWUFjTTBnejc0o2NpZNkgR0N3Z3ZHamF1YlNrMkRybwJycFpuS3VRVnNkUXhpYVA")
    driver.get("https://login.v2.researchbinders.com/u/login/identifier?state=
27
    hKFo2SA0Z0ZiNlhfaWxpamwtM0F5TG5nMS1XbGF5S1kySmI2NKFur3VuaXZlcnNhbC1sb2dpbqN0aWTZ
    IGVpNlFhcTN1YWxhb2h1cVZ10FdNN2VOWlJtNzdvT2pCo2NpZNkgQkJkbHpjR1oxWGZDNmxQeXdVSTFD
    UUJpWnpFbmJYRk8")
28
    def login(username, password):
        # 1: florence sign in page and process of entering username/password and goes though
29
    microsoft login to vumc
30
        trv:
31
            username_box = WebDriverWait(driver, 1).until(EC.element_to_be_clickable((By.XPATH,
    /html/body/div/dīv/div[1]/main/section/div/div/div/div[1]/div/form/div[1]/div/div/div/input')
    ))
32
            username box.send keys(username)
33
            next box = WebDriverWait(driver, 1).until(EC.element to be clickable((By.XPATH, '
34
    /html/body/div/div/div[1]/main/section/div/div/div[1]/div/form/div[2]/button')))
35
            next_box.click()
            try:
36
37
                time.sleep(1)
                 email_box = WebDriverWait(driver, 1).until(EC.element_to_be_clickable((By.XPATH,
38
    '/html/body/div/form[1]/div/div/div[2]/div[1]/div/div/div/div/div[1]/div[3]
    /div/div/div/div[2]/div[2]/div/input[1]')))
39
                 email box.send keys(username)
40
    next\_box = WebDriverWait(driver, \ \ 1).until(EC.element\_to\_be\_clickable((By.XPATH, 'html/body/div/form[1]/div/div/div[2]/div[1]/div/div/div/div[4]
41
    /div/div/div/input')))
                next box.click()
42
```

```
43
                 try:
44
                      time.sleep(1)
    password_box = WebDriverWait(driver, 1)
.until(EC.element_to_be_clickable((By.XPATH, '/html/body/div/form[1]/div/div/div[2]/div[1]
/div/div/div/div/div[3]/div/div[3]/div/div[3]/div/div[2]/input')))
45
46
                      password box.send keys(password)
47
48
                      next box = WebDriverWait(driver, 1)
    .until(EC.element to be clickable((By.XPATH, '/html/body/div/form[1]/div/div/div[2]/div[1]
    /div/div/div/div/div/div[3]/div/div[2]/div/div[4]/div[2]/div/div/div/div/input')))
                      next box.click()
49
50
51
                      next box = WebDriverWait(driver, 1)
    .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div/form/div/div/div[2]/div[1]
    /div/div/div/div/div/div[3]/div/div[2]/div/div[3]/div/div/div(2]/input )))
52
                      next box.click()
                      # reportclick = WebDriverWait(driver, 1)
53
    .until(EC.element_to_be_clickable((By.XPATH,'/html/body/div[1]/app-layout/div[2]/div/binder-
    index/nav-menu/nav/ul/li[4]/a')))
54
                 except:
55
                      return False
56
             except:
57
                 return False
58
        except:
59
             return False
        return True
60
61
62
    # Main program loop
    c = False
63
64
    while not c:
        username = input("Username: ")
65
        password = getpass.getpass("Password: ")
66
        #password = input("Password: ")
67
68
        c = login(username, password)
69
        if not c:
70
             # driver.get("https://login.uatv2.researchbinders.com/u/login/identifier?state=
    hKFo2SBtZFFQa1FXRVVXNm93U1dFNWxyYTFpX0tGTGxUSkc4c6Fur3VuaXZ1cnNhbC1sb2dpbqN0aWTZ
    IFlod2RDbV9rV19ja194VHVNQ1JTSUJŹWUFjTTBnejc0o2NpZNkgR0N3Z3ZHamF1Y1NrMkRybwJycFpu
    S3VRVnNkUXhpYVA")
             driver.get("https://login.v2.researchbinders.com/u/login/identifier?state=
71
    hKFo2SA0Z0ZiNlhfaWxpamwtM0F5TG5nMS1XbGF5S1kySmI2NKFur3VuaXZlcnNhbC1sb2dpbqN0aWTZ
    IGVpNlFhcTN1YWxhb2h1cVZ10FdNN2VOWlJtNzdvT2pCo2NpZNkgQkJkbHpjR1oxWGZDNmxQeXdVSTFD
    UUJpWnpFbmJYRk8")
             print('--Incorrect Username/Password--')
72
73
        else:
74
             break
        print('----')
75
76
77
    # download report
78
79
    #template for explicit wait clicks
    # reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,)))
80
81
    # reportclick.click()
    reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
82
    /html/body/div[1]/app-layout/div[2]/div/binder-index/nav-menu/nav/ul/li[4]/a')))
    reportclick.click()
83
84
    reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
/html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div/button')))
85
    reportclick.click()
```

```
87
     reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
 88
      /html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div/ul/li[7]/a')))
     reportclick.click()
 29
 90
     reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
/html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div[2]/button/span')))
 91
 92
     reportclick.click()
 93
      reportclick = WebDriverWait(driver, 20).until(EC.element to be clickable((By.XPATH,'
 94
      /html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div[2]/ul/li[1]/span/a')))
 95
     reportclick.click()
 96
 97
     reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
      /html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div[2]/button')))
 98
     reportclick.click()
 99
     reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
/html/body/div[1]/app-layout/div[2]/div/reports-index/section/reports-
table/section/div/div/div/div[1]/a')))
100
101
     reportclick.click()
102
103
     time.sleep(1)
104
     reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
/html/body/div[1]/app-layout/div[2]/div/reports-index/nav-menu/nav/ul/li[6]/a')))
105
106
     reportclick.click()
107
108
     wait = WebDriverWait(driver, 360)
     wait.until(EC.text_to_be_present_in_element((By.XPATH, '/html/body/div[1]/app-layout/div[2]
/div/downloads-index/section/table/tbody/tr[1]/td[4]'), 'Downloaded'))
109
110
     # box for automation of duplicating into osr holding binder
111
112
113
     time.sleep(5)
114
     # Set the path where the CSV files are located
115
     path = "C:\\Users\\cruzrl\\Downloads" # Subject to change when user is confirmed
116
117
     # Find the most recently uploaded CSV file in the specified path
     file = max(glob.iglob(path + "\\*.csv"), key=os.path.getmtime)
118
119
120
     # Print the path of the most recently uploaded CSV file
     print(file)
121
122
     # Cleaning of untouched downloads from Florence
123
124
125
     # Read the CSV file into a DataFrame
126
     df2 = pd.read csv(file)
127
     # Filter the DataFrame to include only rows where the 'Location' column contains 'OSR'
128
     df2 = df2[df2['Location'].str.contains('OSR')]
129
130
     # Create an empty DataFrame with columns 'Row Labels' and 'Count of Location'
131
     clean = pd.DataFrame(columns=['Row Labels', 'Count of Location'])
132
133
134
     # Iterate over the 'Location' column of df2
135
     for i in df2['Location']:
```

```
if i.split('/')[0] != 'OSR Holding': # Check if the first part of the location is not '
136
     OSR Holding'
             if i.split('/')[0] not in clean['Row Labels'].values: # Check if the location is not
137
     already in 'clean'
                 # Append a new row to 'clean' with the location and count as 1
138
                 clean = clean.append({'Row Labels': i.split('/')[0], 'Count of Location': 1},
139
     ignore index=True)
140
             else:
141
                 # Increment the count for the existing location in 'clean'
142
                 clean.loc[clean['Row Labels'] == i.split('/')[0], 'Count of Location'] += 1
143
144
     # Remove rows with 'Row Labels' equal to '1.0 Central Study Documents' from 'clean'
     clean = clean[clean['Row Labels'] != '1.0 Central Study Documents']
145
146
     # Reset the index of 'clean' and drop the 'index' column
147
148
     clean = clean.reset_index()
     clean = clean.drop(['index'], axis=1)
149
150
151
     # Separate entries in 'clean' into two DataFrames based on the count value
152
153
     # Create df0 DataFrame with entries having count <= 100
     df0 = clean[clean['Count of Location'] <= 100]</pre>
154
155
     # Create df1 DataFrame with entries having count > 100
156
     df1 = clean[clean['Count of Location'] > 100]
157
158
     # Reset the index of 'df0' and 'df1' and drop the 'index' column
159
160
     df0 = df0.reset index()
     df1 = df1.reset_index()
161
     df0 = df0.drop(['index'], axis=1)
162
     df1 = df1.drop(['index'], axis=1)
163
164
     # Print the 'Row Labels' from 'clean'
165
     for i in clean['Row Labels']:
166
         print(i)
167
168
     # Print the length of 'clean' DataFrame
169
170
     print(len(clean))
171
     warnings.simplefilter(action='ignore', category=FutureWarning)
172
     # go to binders page
173
     homec = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
     /html/body/div[1]/app-layout/navbar/section/nav/a[1]/img<sup>T</sup>)))
174
     homec.click()
175
     # loops through study names
176
177
     for i in range(0, len(clean.index)):
         studyName = clean['Row Labels'][i]
178
179
180
         # search study
181
         search_box = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[2]/table-filter/div/input'))
182
         # search box= driver.find element(By.XPATH,'/html/body/div[1]/app-layout/div[2]
     /div/binder-index/section/div[2]/table-filter/div/input')
         search_box.send_keys(studyName)
183
184
185
         # click study
```

```
study = WebDriverWait(driver, \begin{tabular}{l} 20 \end{tabular}. until(EC.element\_to\_be\_clickable((By.XPATH, 'html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[3]/div[2]/md-virtual-repeat-layout/div[2]/div/binder-index/section/div[3]/div[2]/md-virtual-repeat-layout/div[3]/div[2]/md-virtual-repeat-layout/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/div[3]/d
186
         container/div/div[2]/div/div[2]/a')))
                # study=driver.find_element(By.XPATH,'/html/body/div[1]/app-layout/div[2]/div/binder-
187
         index/section/div[3]/div[2]/md-virtual-repeat-container/div/div[2]/div/div[2]/a')
                studv.click()
188
189
190
                # variables to determine if osr is found and if it can be completed
191
                skip = False
192
                checkbox_element = None
193
                element_found = False
194
                completed check = False
195
196
                # finds osr folder in study
197
                while not element_found:
198
                        # Wait for elements matching the CSS selector to become clickable
199
                        elements = WebDriverWait(driver, 20)
         .until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, '.c-infinite-table__row.u-user-
         select-none.ng-scope.folder-show__item--loaded')))
200
                        # Loop through the elements
201
202
                        for i in elements:
203
                               matching elements = i.find elements(By.CSS SELECTOR, '.c-infinite-table cell .u-
         d-inline-block.u-word-break-break-all[class*="test-itemNameLink-"]')
204
                               for element in matching elements:
205
                                       if element.text == 'OSR' or element.text == 'OSRs':
                                              class name = element.get attribute('class')
206
207
                                              alphanumeric code = re.search(r'test-itemNameLink-(\w+)', class name)
         .group(1)
208
                                              checkbox = i.find_element(By.CSS_SELECTOR, f'.c-infinite-table cell
          .checkbox-icon.far.fa-fw.fa-square.test-itemCheckbox-{alphanumeric_code}')
209
                                              checkbox_element = checkbox
                                              # Scroll to the element
210
211
                                              driver.execute_script("arguments[0].scrollIntoView();", checkbox)
212
213
                                              # Click the checkbox using ActionChains
214
                                              actions = ActionChains(driver)
215
                                              actions.move_to_element(checkbox).click().perform()
216
217
                                              # Retrieve the document count
                                              document count element = i.find element(By.CSS SELECTOR, f'.test-
218
         counterLabel-{alphanumeric_code}')
219
                                              document_count = document_count_element.text
220
                                              if document_count.isdigit():
221
                                                      document count = int(document_count)
222
223
                                                      print('Checkbox found and clicked for element with text OSR or OSRs
         and alphanumeric code:', alphanumeric_code)
                                                      print('Document count:', document_count)
224
225
                                              else:
226
                                                      print('Invalid document count:', document_count)
227
                                              if document count <= 100:</pre>
                                                      completed_check = True
228
229
                                              element_found = True
230
                                              break # Exit the inner loop after finding and clicking the checkbox
231
                               if element_found:
232
                                       break # Exit the outer loop if the desired element is found
233
```

```
if not element_found:
234
235
                 # Scroll down using the scroller XPath
236
                 scroll_element = driver.find_element(By.XPATH, '/html/body/div[1]/app-
    layout/div[2]/div/folder-show/section/div[4]/div[2]/md-virtual-repeat-container/div')
237
                 driver.execute_script("arguments[0].scrollBy(0, 100);", scroll_element)
                 print('Element with text OSR or OSRs not found. Scrolling down...')
238
239
240
         action_box = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/app-layout/div[2]/div/folder-show/section/div[2]/div[2]/button')))
         # action_box=driver.find_element(By.XPATH, '/html/body/div[1]/app-layout/div[2]
241
     /div/folder-show/section/div[2]/div[2]/button')
242
         action box.click()
243
244
         duplicate = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/app-layout/div[2]/div/folder-show/section/div[2]/div[2]/ul/li[3]/a')))
    # duplicate=driver.find_element(By.XPATH, '/html/body/div[1]/app-layout/div[2]
/div/folder-show/section/div[2]/div[2]/ul/li[3]/a')
245
246
         duplicate.click()
247
248
         find_study = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[1]/virtual-
     tree/div/div[2]/table-filter/div/input')))
249
         # find study = driver.find element(By.XPATH, '/html/body/div[1]/div/div/duplicate-modal-
    wrapper/duplicate/section/div[1]/virtual-tree/div/div[2]/table-filter/div/input')
250
         find study.send keys('OSR Holding')
251
         select for copy = WebDriverWait(driver, 20).until(EC.element to be clickable((By.XPATH, '
252
     /html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[1]/virtual-tree/cdk-
    virtual-scroll-viewport/div[1]/div[2]/div/div/div')))
         # select_for_copy = driver.find_element(By.XPATH, '/html/body/div[1]/div/div/duplicate-
253
    modal-wrapper/duplicate/section/div[1]/virtual-tree/cdk-virtual-scroll-viewport/div[1]/div[2]
     /div/div/div')
254
         select for copy.click()
255
         name_copy = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
256
     /html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[1]/form/div[3]/div[1]
     /input')))
         # name copy = driver.find_element(By.XPATH, '/html/body/div[1]/div/div/duplicate-modal-
257
    wrapper/duplicate/section/div[1]/form/div[3]/div[1]/input')
258
         actionChains = ActionChains(driver)
         actionChains.double_click(name_copy).perform()
259
260
         name_copy.send_keys(studyName)
261
262
         # hit duplicate
         duplicate = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
263
     /html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[2]/div/button')))
         # duplicate = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
264
     /html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[2]/button')))
         # duplicate = driver.find element(By.XPATH, '/html/body/div[1]/div/div/duplicate-modal-
265
    wrapper/duplicate/section/div[2]/div/button')
266
         duplicate.click()
267
268
         # process to rename osr to osr + date and create new osr folder if successfully
    duplicated
         if completed check:
269
270
             time.sleep(20)
271
             already_complete = False
272
             element found = False
273
             date = date.today()
             while not element_found:
274
```

```
275
                 elements = WebDriverWait(driver, 20)
     .until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, '.c-infinite-table__row.u-user-
    select-none.ng-scope.folder-show__item--loaded')))
276
                 for i in elements:
                     matching elements = i.find elements(By.CSS SELECTOR, '.c-infinite-table cell
277
     .u-d-inline-block.u-word-break-break-all[class*="test-itemNameLink-"]')
                     for element in matching elements:
278
                         if element.text == ("Completed OSR " + date.strftime("%m-%d-%y")):
279
                             print("OSRs already moved to holding")
280
281
                             already complete = True
                             element_found = True
282
283
                             break
284
                         if element.text == 'OSR' or element.text == 'OSRs':
285
                             class_name = element.get_attribute('class')
286
                             alphanumeric code = re.search(r'test-itemNameLink-(\w+)', class name)
     .group(1)
287
                             time.sleep(1)
                             checkbox = i.find_element(By.CSS_SELECTOR, f'.c-infinite-table__cell
288
     .checkbox-icon.far.fa-fw.fa-square.test-iTemCheckbox-{alphanumeric_code}')
289
                             checkbox element = checkbox
290
                             driver.execute_script("arguments[0].scrollIntoView();", checkbox)
291
                             actions = ActionChains(driver)
292
                             actions.move_to_element(checkbox).click().perform()
293
                             if not already_complete:
                                  dropdown = i.find_element(By.CSS_SELECTOR, "div.c-infinite-
294
     table__cell.mr-3.dropdown-menu-right-nested.dropdown a.test-actionsLink")
295
                                  driver.execute script("arguments[0].scrollIntoView();", dropdown)
296
                                  actions = ActionChains(driver)
297
                                  actions.move_to_element(dropdown).click().perform()
298
                                  wait = WebDriverWait(driver, 10)
299
                                  time.sleep(2)
300
                                 ul element = i.find element(By.CSS SELECTOR, 'div.c-infinite-
301
     table__cell.mr-3.dropdown-menu-right-nested.dropdown.open_ul.dropdown-menu.ng-scope')
302
                                  li elements = WebDriverWait(ul element, 10)
     .until(EC.presence of all elements located((By.CSS SELECTOR, 'ul.dropdown-menu.ng-scope
     a.test-editLink.ng-scope')))
303
                                  for li_element in li_elements:
304
                                      try:
305
                                          print(li element.text)
306
                                          if "Rename/Update" in li_element.text:
307
                                              li element.click()
308
                                              break
309
                                      except NoSuchElementException:
310
311
                                  time.sleep(1)
312
                                  name copy = WebDriverWait(driver, 20)
     .until(EC.element_to_be_clickable((By.XPATH, '/html/body/modal-container/div/div/folder-
     form/form/div[1]/div/input')))
313
                                  actionChains = ActionChains(driver)
314
                                  actionChains.double_click(name_copy).perform()
315
                                  name_copy.send_keys("Completed OSR " + date.strftime("%m-%d-%y"))
316
                                  time.sleep(1)
317
                                  save = WebDriverWait(driver, 20)
     .until(EC.element to be clickable((By.XPATH, '/html/body/modal-container/div/div/folder-
     form/form/div[2]/button[2]')))
318
                                  save.click()
319
                                  time.sleep(3)
                                  save = WebDriverWait(driver, 20)
320
     .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div[1]/app-layout/div[2]/div/folder-
```

```
show/section/div[2]/div[1]/button')))
321
                                  save.click()
322
                                  save = WebDriverWait(driver, 20)
     .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div[1]/app-layout/div[2]/div/folder-
     show/section/div[2]/div[1]/ul/li[1]/a')))
323
                                  save.click()
324
325
                                  save = WebDriverWait(driver, 20)
     .until(EC.element to be clickable((By.XPATH, '/html/body/modal-container/div/div/folder-
     form/form/div[1]/div/input')))
                                  save.send keys("OSR")
326
                                  save = WebDriverWait(driver, 20)
327
     .until(EC.element_to_be_clickable((By.XPATH, '/html/body/modal-container/div/div/folder-
     form/form/div[2]/button[2]')))
328
                                  save.click()
329
                                  # print("check")
330
                              element found = True
331
                              break
332
                     if element_found:
333
                         break
334
                 if not element_found:
335
                     scroll element = driver.find element(By.XPATH, '/html/body/div[1]/app-
     layout/div[2]/div/folder-show/section/div[4]/div[2]/md-virtual-repeat-container/div')
336
                     driver.execute_script("arguments[0].scrollBy(0, 100);", scroll_element)
                     print('Working...')
337
338
339
340
             completed_check = False
341
         time.sleep(1)
342
         max_retries = 3
343
         retry_count = 0
344
345
         # occasionally the binders tab is clickable but not loaded in DOM so this gives the code
     3 chances to click the tab
346
         while retry_count < max_retries:</pre>
347
                 homec = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,'
348
     /html/body/div[1]/app-layout/div[2]/div/folder-show/nav-menu/nav/ul/li[3]/a')))
349
                 homec.click()
350
                 break # If click is successful, exit the loop
351
             except StaleElementReferenceException:
352
                 # Increment retry count
353
                 retry_count += 1
354
                 print(f"StaleElementReferenceException occurred. Retrying ({retry count}
     /{max_retries})...")
355
                 continue
356
             except:
357
                 # Increment retry count
358
                 retry_count += 1
359
                 print(f"Error occurred. Retrying ({retry_count}/{max_retries})...")
360
                 continue
     search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
361
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[2]/table-filter/div/input'))
362
     search.send_keys('OSR Holding')
363
364
     search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[3]/div[2]/md-virtual-repeat-
     container/div/div[2]/div/div[1]/i')))
```

```
2/22/24, 4:13 PM
  365
  366
  367
  368
```

```
search.click()
     search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[2]/div[2]/button')))
     search.click()
369
370
     search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[2]/div[2]/ul/li[4]/a')))
371
     search.click()
372
     search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
373
     /html/body/div[1]/div/div/binder-download-wrapper/binder-download/section/div[2]/button[2]'))
374
     search.click()
375
376
     max retries = 3
377
     retry_count = 0
378
379
     while retry_count < max_retries:</pre>
380
             homec = WebDriverWait(driver, 20).until(EC.element to be clickable((By.XPATH,'
381
     /html/body/div[1]/app-layout/div[2]/div/binder-index/nav-menu/nav/ul/li[6]/a')))
382
             homec.click()
383
             break # If click is successful, exit the loop
384
         except StaleElementReferenceException:
385
             # Increment retry count
386
             retry_count += 1
387
             print(f"StaleElementReferenceException occurred. Retrying ({retry_count})
     /{max_retries})...")
388
             continue
389
390
     # code waits up to 900 seconds for OSR Holding to download and will time out if longer (can
     be changed)
391
     wait = WebDriverWait(driver, 900)
     wait.until(EC.text_to_be_present_in_element((By.XPATH, '/html/body/div[1]/app-layout/div[2]
/div/downloads-index/section/table/tbody/tr[1]/td[4]'), 'Downloaded'))
392
393
     time.sleep(1)
394
395
     # delete from osr holding
396
     # deletes top row until nothing is left
397
398
     delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
     /html/body/div[1]/app-layout/div[2]/div/downloads-index/nav-menu/nav/ul/li[3]/a')))
399
     delete.click()
400
401
     delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[2]/table-filter/div/input'))
402
     delete.send_keys('OSR Holding')
403
404
     delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/app-layout/div[2]/div/binder-index/section/div[3]/div[2]/md-virtual-repeat-
     container/div/div[2]/div/div[2]/a')))
405
     delete.click()
406
407
     time.sleep(1)
     c = True
408
409
     while c:
410
         try:
```

```
411
             time.sleep(1)
             name = WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '
412
     /html/body/div[1]/app-layout/div[2]/div/folder-show/section/div[4]/div[2]/md-virtual-repeat-
     container/div/div[2]/div[1]/div[2]/a'))).text
             delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
413
     /html/body/div[1]/app-layout/div[2]/div/folder-show/section/div[4]/div[2]/md-virtual-repeat-
     container/div/div[2]/div[1]/div[9]/a')))
             delete.click()
414
415
416
             time.sleep(1)
417
418
             delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
     /html/body/div[1]/app-layout/div[2]/div/fólder-show/section/div[4]/div[2]/md-virtual-repeat-
     container/div/div[2]/div[1]/div[9]/ul/li[2]/a')))
419
             delete.click()
420
421
             delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/div/div/destroy-folder/section/div[1]/div/div/form/div/input')))
422
             delete.send keys(name)
423
424
             delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
     /html/body/div[1]/div/div/destroy-folder/section/div[1]/div/div/form/div/textarea')))
425
             delete.send keys("post automation deletion")
426
427
             delete = WebDriverWait(driver, 20).until(EC.element to be clickable((By.XPATH, '
     /html/body/div[1]/div/div/destroy-folder/section/div[2]/button[2]/span/span'
428
             delete.click()
429
             time.sleep(1)
430
         except:
431
             c = False
432
     # Close the browser
433
     driver.quit()
434
     # comparison from florence and osr binder
435
     with zipfile('C:\\Users\\cruzrl\\Downloads\\OSR Holding.zip', 'r') as zip:
436
437
         origFileNames = zip.namelist()
     fileNames = []
438
439
     for string in origFileNames:
440
         print(string)
         string = string[12:]
441
442
         string = string.split('/')[0]
443
         if string not in fileNames:
444
             fileNames.append(string)
445
     for i in fileNames:
446
         print(i)
447
         if i in df0['Row Labels'].to_numpy():
448
             if i not in fileNames:
                 df1 = df1.append({'Row Labels': i, 'Count of Location': 1}, ignore index = True)
449
450
     # for i in df1['Row Labels']:
451
           print(i)
452
     with zipfile('C:\\Users\\cruzrl\\Downloads\\OSR Holding.zip', 'r') as zip:
453
         origFileNames = zip.namelist()
454
     fileNames = []
455
     for string in origFileNames:
456
         string = string[12:]
457
         string = string.split('/')[0]
458
         if string not in fileNames:
459
             fileNames.append(string)
```

2/22/24, 4:13 PM

```
460
            # print(string)
461
    over = [] # Create a copy of the 'clean' list
462
463
464
465
    for i in clean['Row Labels']:
        if i not in fileNames:
466
467
            # print(i)
            over.append(i)
468
469
    print(len(over))
    print(len(clean))
470
471 print(len(fileNames))
472 # for i in df1['Row Labels']:
473 #
         print(i)
```