

~\Downloads\florence final.py

```

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
10 from selenium.webdriver.support.ui import WebDriverWait
11 from selenium.webdriver.support import expected_conditions as EC
12 from selenium.common.exceptions import StaleElementReferenceException
13 import time
14 import csv
15 import os
16 import glob
17 import pandas as pd
18 from datetime import date
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=
hKfo2SBtZFFQa1FXRVVXNm93U1dFNWxyYTFpX0tGTGxUSkc4c6Fur3VuaXZ1cnNhbc1sb2dpbqN0aWTZ
IFlod2RDbV9rV19ja194VHVNQ1JTSUJZWUFjTTBnejc0o2NpZnkgR0N3Z3HamF1YlNrMkRybWJycFpu
S3VRVnNkUXhpYVA")
27 driver.get("https://login.v2.researchbinders.com/u/login/identifier?state=
hKfo2SA0Z0ZiNlhfaWxpamwtM0F5TG5nMS1XbGF5S1kySmI2NKf3VuaXZ1cnNhbc1sb2dpbqN0aWTZ
IGVpNlFhcTN1Ywxb2h1cVZ1OFdNN2VOWlJtNzdvt2pCo2NpZnkgQkKbHpjR1oxWGZDNmxQeXdVSTFD
UUJpWnpFbmJYRk8")
28 def login(username, password):
29     # 1: florence sign in page and process of entering username/password and goes though
microsoft login to vumc
30     try:
31         username_box = WebDriverWait(driver, 1).until(EC.element_to_be_clickable((By.XPATH, '
/html/body/div/div/div[1]/main/section/div/div/div/div[1]/div/form/div[1]/div/div/div/input'
)))
32         username_box.send_keys(username)
33
34         next_box = WebDriverWait(driver, 1).until(EC.element_to_be_clickable((By.XPATH, '
/html/body/div/div/div[1]/main/section/div/div/div/div[1]/div/form/div[2]/button'
))))
35         next_box.click()
36         try:
37             time.sleep(1)
38             email_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[1]/div[3]
/div/div/div/div[2]/div[2]/div/input[1]'))))
39             email_box.send_keys(username)
40
41             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[1]/div[3]/div/div/div/div[4]
/div/div/div/div/input'))))
42             next_box.click()

```

```

43     try:
44         time.sleep(1)
45         password_box = WebDriverWait(driver, 1)
46         .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div/form[1]/div/div/div[2]/div[1]
47         /div/div/div/div/div/div/div[3]/div/div[2]/div/div[3]/div/div[2]/input')))
48         password_box.send_keys(password)
49
50         next_box = WebDriverWait(driver, 1)
51         .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div/form[1]/div/div/div[2]/div[1]
52         /div/div/div/div/div/div/div[3]/div/div[2]/div/div[4]/div[2]/div/div/div/div/input')))
53         next_box.click()
54
55         next_box = WebDriverWait(driver, 1)
56         .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div/form/div/div/div[2]/div[1]
57         /div/div/div/div/div/div/div[3]/div/div[2]/div/div[3]/div[2]/div/div/div[2]/input')))
58         next_box.click()
59         # reportclick = WebDriverWait(driver, 1)
60         .until(EC.element_to_be_clickable((By.XPATH, '/html/body/div[1]/app-layout/div[2]/div/binder-
61         index/nav-menu/nav/u1/li[4]/a')))
62     except:
63         return False
64     except:
65         return False
66     except:
67         return False
68     return True
69
70 # Main program loop
71 c = False
72 while not c:
73     username = input("Username: ")
74     password = getpass.getpass("Password: ")
75     #password = input("Password: ")
76     c = login(username, password)
77     if not c:
78         # driver.get("https://login.uatv2.researchbinders.com/u/login/identifier?state=
79         hKFo2SBtZFFQa1FXRVVXNm93U1dFNWxyYTFpX0tGTGxUSKc4c6Fur3VuaXZ1cnNhbc1sb2dpbqN0aWTZ
80         IFlod2RDv9rV19ja194VHVNQ1JTSUJZWUFjTTBnejc0o2NpZNkgR0N3Z3HamF1Y1NrMkRyBwJycFpu
81         S3VRVnNkUXhpYVA")
82         driver.get("https://login.v2.researchbinders.com/u/login/identifier?state=
83         hKFo2SA0Z0ZiNlhfaWxpamwtM0F5TG5nMS1XbGF5S1kySmI2NKFur3VuaXZ1cnNhbc1sb2dpbqN0aWTZ
84         IGVpNlFhcTN1YXh2h1cVZ1OFdNN2VOW1JtNzdvt2pCo2NpZNkgQkKbHpjR1oxWGZDNmxQeXdVSTFD
85         UUpWnpFbmJYRk8")
86         print('--Incorrect Username/Password--')
87     else:
88         break
89     print('-----')
90
91 # download report
92
93 #template for explicit wait clicks
94 # reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,)))
95 # reportclick.click()
96 reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
97 /html/body/div[1]/app-layout/div[2]/div/binder-index/nav-menu/nav/u1/li[4]/a')))
98 reportclick.click()
99
100 reportclick = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
101 /html/body/div[1]/app-layout/div[2]/div/reports-index/section/div/div/button')))
102 reportclick.click()

```

```
87
88 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/ul/li[7]/a')))
89 reportclick.click()
90
91 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')))
92 reportclick.click()
93
94 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]/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
100 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[1]/a')))
101 reportclick.click()
102
103 time.sleep(1)
104
105 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')))
106 reportclick.click()
107
108 wait = WebDriverWait(driver, 360)
109 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'))
110
111 # box for automation of duplicating into osr holding binder
112
113 time.sleep(5)
114 # Set the path where the CSV files are located
115 path = "C:\\Users\\cruzr1\\Downloads" # Subject to change when user is confirmed
116
117 # Find the most recently uploaded CSV file in the specified path
118 file = max(glob.iglob(path + "\\*.csv"), key=os.path.getmtime)
119
120 # Print the path of the most recently uploaded CSV file
121 print(file)
122
123 # Cleaning of untouched downloads from Florence
124
125 # Read the CSV file into a DataFrame
126 df2 = pd.read_csv(file)
127
128 # Filter the DataFrame to include only rows where the 'Location' column contains 'OSR'
129 df2 = df2[df2['Location'].str.contains('OSR')]
130
131 # Create an empty DataFrame with columns 'Row Labels' and 'Count of Location'
132 clean = pd.DataFrame(columns=['Row Labels', 'Count of Location'])
133
134 # Iterate over the 'Location' column of df2
135 for i in df2['Location']:
```

```
136     if i.split('/')[0] != 'OSR Holding': # Check if the first part of the location is not '
OSR Holding'
137     if i.split('/')[0] not in clean['Row Labels'].values: # Check if the location is not
already in 'clean'
138         # Append a new row to 'clean' with the location and count as 1
139         clean = clean.append({'Row Labels': i.split('/')[0], 'Count of Location': 1},
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'
145 clean = clean[clean['Row Labels'] != '1.0 Central Study Documents']
146
147 # Reset the index of 'clean' and drop the 'index' column
148 clean = clean.reset_index()
149 clean = clean.drop(['index'], axis=1)
150
151 # Separate entries in 'clean' into two DataFrames based on the count value
152
153 # Create df0 DataFrame with entries having count <= 100
154 df0 = clean[clean['Count of Location'] <= 100]
155
156 # Create df1 DataFrame with entries having count > 100
157 df1 = clean[clean['Count of Location'] > 100]
158
159 # Reset the index of 'df0' and 'df1' and drop the 'index' column
160 df0 = df0.reset_index()
161 df1 = df1.reset_index()
162 df0 = df0.drop(['index'], axis=1)
163 df1 = df1.drop(['index'], axis=1)
164
165 # Print the 'Row Labels' from 'clean'
166 for i in clean['Row Labels']:
167     print(i)
168
169 # Print the length of 'clean' DataFrame
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')))
174 homec.click()
175
176 # loops through study names
177 for i in range(0, len(clean.index)):
178     studyName = clean['Row Labels'][i]
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')
183     search_box.send_keys(studyName)
184
185     # click study
```

```

186     study = 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')))
187     # study=driver.find_element(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')
188     study.click()
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
201         # Loop through the elements
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':
206                     class_name = element.get_attribute('class')
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
210                     # Scroll to the element
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
218                     document_count_element = i.find_element(By.CSS_SELECTOR, f'.test-
counterLabel-{alphanumeric_code}')
219                     document_count = document_count_element.text
220
221                     if document_count.isdigit():
222                         document_count = int(document_count)
223                         print('Checkbox found and clicked for element with text OSR or OSRs
and alphanumeric code:', alphanumeric_code)
224                         print('Document count:', document_count)
225                     else:
226                         print('Invalid document count:', document_count)
227                     if document_count <= 100:
228                         completed_check = True
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

```

```
234     if not element_found:
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)
238         print('Element with text OSR or OSRs not found. Scrolling down...')
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')))
241         # action_box=driver.find_element(By.XPATH, '/html/body/div[1]/app-layout/div[2]
/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')))
245         # 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')
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[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[2]/table-filter/div/input')
250         find_study.send_keys('OSR Holding')
251
252         select_for_copy = 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/cdk-
virtual-scroll-viewport/div[1]/div[2]/div/div/div')))
253         # select_for_copy = driver.find_element(By.XPATH, '/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')
254         select_for_copy.click()
255
256         name_copy = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[1]/form/div[3]/div[1]
/input')))
257         # name_copy = driver.find_element(By.XPATH, '/html/body/div[1]/div/div/duplicate-modal-
wrapper/duplicate/section/div[1]/form/div[3]/div[1]/input')
258         actionChains = ActionChains(driver)
259         actionChains.double_click(name_copy).perform()
260         name_copy.send_keys(studyName)
261
262         # hit duplicate
263         duplicate = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[2]/div/button')))
264         # duplicate = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/html/body/div[1]/div/div/duplicate-modal-wrapper/duplicate/section/div[2]/button')))
265         # duplicate = driver.find_element(By.XPATH, '/html/body/div[1]/div/div/duplicate-modal-
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
269         if completed_check:
270             time.sleep(20)
271             already_complete = False
272             element_found = False
273             date = date.today()
274             while not element_found:
```



```

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:
277         matching_elements = i.find_elements(By.CSS_SELECTOR, '.c-infinite-table__cell
.u-d-inline-block.u-word-break-break-all[class*="test-itemNameLink-"]')
278         for element in matching_elements:
279             if element.text == ("Completed OSR " + date.strftime("%m-%d-%y")):
280                 print("OSRs already moved to holding")
281                 already_complete = True
282                 element_found = True
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)
288                 checkbox = i.find_element(By.CSS_SELECTOR, f'.c-infinite-table__cell
.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:
294                     dropdown = i.find_element(By.CSS_SELECTOR, "div.c-infinite-
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
301                     ul_element = i.find_element(By.CSS_SELECTOR, 'div.c-infinite-
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                             pass
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)
320                     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]/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'))))
326         save.send_keys("OSR")
327         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]'))))
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)
337         print('Working...')
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:
347             try:
348                 homec = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/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
361         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]/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'))))

```



```
365 search.click()
366
367 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')))
368 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
373 search = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/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:
380     try:
381         homec = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/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)
392 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'))
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)
408 c = True
409 while c:
410     try:
```

```

411     time.sleep(1)
412     name = WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '
/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
413     delete = WebDriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH, '
/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'))))
414     delete.click()
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/folder-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
436     with zipfile('C:\\Users\\cruzrl\\Downloads\\OSR Holding.zip', 'r') as zip:
437         origFileNames = zip.namelist()
438         fileNames = []
439         for string in origFileNames:
440             print(string)
441             string = string[12:]
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:
449                     df1 = df1.append({'Row Labels': i, 'Count of Location': 1}, ignore_index = True)
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)

```

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