



בית הספר  
להייינט וסיבר

המערך לתוכניות וייעודיות  
אוניברסיטת בר-אילן

# AUTOMATION TESTING - FINAL PROJECT

EDEN TSABARI REUVEN

GET STARTED





# הציג פרויקט - קורסAQ בדיקות אוטומציה

במהלך העבודה הפרויקט, כתבתי תסריטי בדיקה אוטומטיים באמצעות **SELENIUM**, שהיו מיועדים לבדוק את התוכן של האתר, כולל אלמנטים שונים כמו כפתורים, תפריטים וטקסטים. בנוסף, בדקתי את הפונקציות השונות של האתר כדי לוודא שהן פועלות כמצופה.

## הרצת הבדיקות

### שפט תכנות:

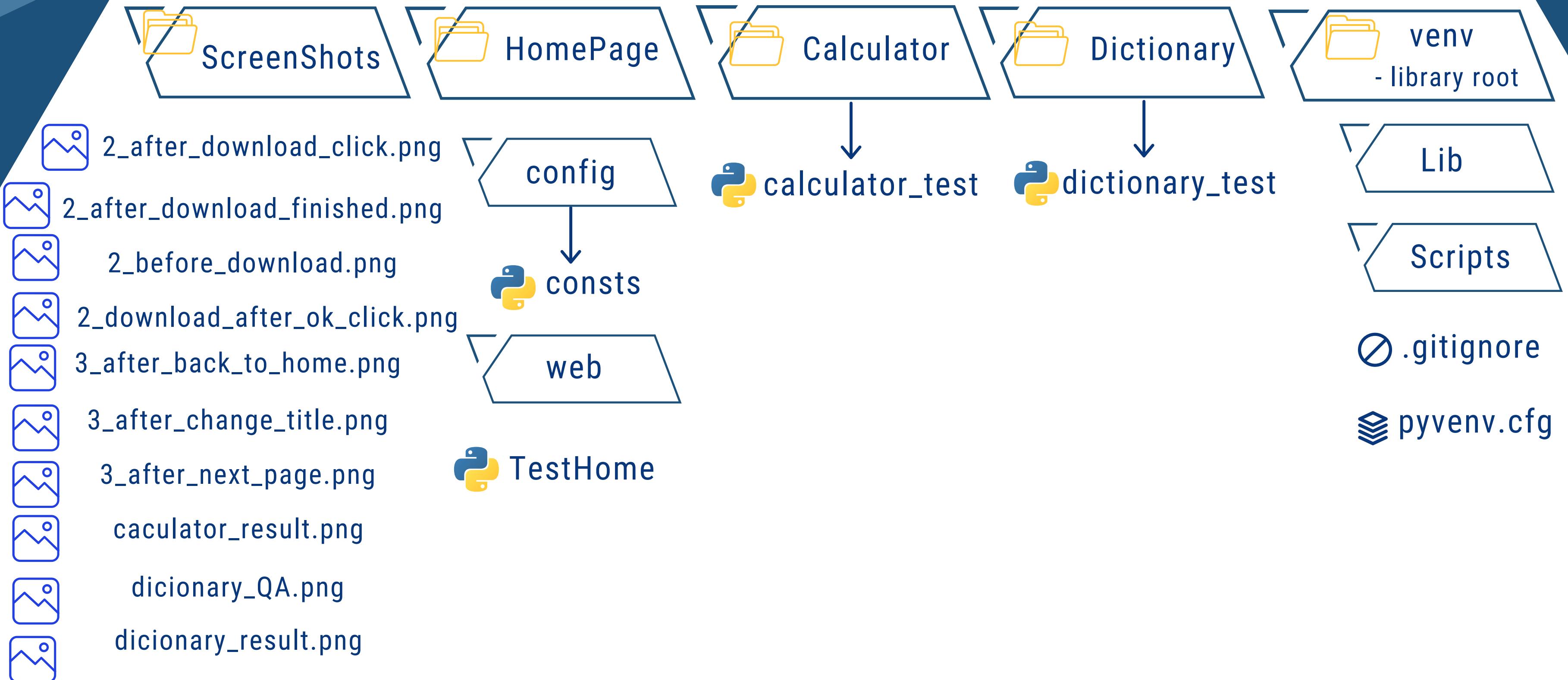
הרצת הבדיקות בוצעה בשפט פיתון.

### סביבה עבודה:

השתמשתי ב-**PYCHARM** כ סביבת עבודה.



# automation\_project\_eden



## config

**מטרת התקינה:**

ארגון: לרכז את כל הגדירות למקומות אחד, מה שמקל על התחזוקה.

גמישות: ניתן לשנות את הגדירות מבלוי לגעת בקוד.

נוחות: ניתן להוסיף או לשנות פרמטרים בקלות, במיוחד כמיוחד כשייש צורך לבודד בסביבות שונות.

```
consts.py  × :  
1  BASE_URL= "file:///C:/programin/python/workspice/automation_project_eden/tests/web/home.html" ⚠ 2 ✓ 6  ⌂  
2  NEXT_PAGE_URL = "file:///C:/programin/python/workspice/automation_project_eden/tests/web/nextpage.html"  
3  BROWSER_TITLE = "Automation Project"  
4  PAGE_MAIN_TITLE = "Automation Project - Main Page"  
5  CITIES_TABLE = "Cities of the World"  
6  STUDENTS_TABLE = "Student Details"  
7  FIRST_NAME_COLUMN = "First Name"  
8  STUDENTS_FIRST_NAME = ['John', 'Jane', 'Alice', 'Michael', 'Emily']  
9  FORM_DATA = {"first_name": 'Eden', 'last_name': 'tsabari',  
10     'email': '1234eden@gmail.com', 'mobile': 2565887}  
11  DICTIONARY_PAGE_URL = "https://www.merriam-webster.com/"  
12  
13
```



# בדיקות תוכן

Automation Project - Main Page

## Cities of the World

USA	Europe	Israel
Washington, D.C.	London	Jerusalem
New York City	Manchester	Tel Aviv
San Francisco	Berlin	Haifa

## Student Details

ID	First Name	Last Name	Grade
1	John	Smith	A
2	Jane	Doe	B+
3	Alice	Johnson	A-
4	Michael	Williams	B
5	Emily	Brown	A

## Form

All Rights Reserved ©



## Links

- [Next Page](#)
- [Google](#)
- [Merriam-Webster Dictionary](#)
- [Jerusalem Municipality](#)



python™ TestHome.py ×

# TestHome

⚠ 1 ⚠ 38 ✅ 53

```
1 import unittest
2 from selenium import webdriver
3 from config import consts
4 from selenium.webdriver.common.by import By
5 from selenium.webdriver.support.ui import Select
6 from selenium.common.exceptions import NoSuchElementException
7 import random
8 from time import sleep
9
10 class TestHome(unittest.TestCase):
11     driver = None
12
13     @classmethod
14     def setUpClass(cls):
15         cls.driver = webdriver.Chrome()
16         cls.driver.maximize_window()
17
18     @classmethod
19     def tearDownClass(cls):
20         cls.driver.quit()
21
22     def setUp(self):
23         self.driver.get(consts.BASE_URL)
```

מетодת מחלקה  
מетодה זו רצה פעם אחת לפני כל הבדיקות במחלקה.  
היא נועדה להכין סביבה שנדרשת לכל הבדיקות במחלקה.

מетодת מחלקה  
מетодה זו רצה פעם אחת לאחר שכל הבדיקות במחלקה הושלמו

מетодה זו רצה לפני כל בדיקה.

python™

TestHome.py ×

```

10  class TestHome(unittest.TestCase):
25      #title בדיקת題
26  ▶  def test_browser_title(self):
27      self.assertEqual(consts.BROWSER_TITLE, self.driver.title)
28
29      #h1 בדיקת題 הגדיש ושם ירים רק אלמנט אחד מסויים
30  ▶  def test_page_title(self):
31      h1_list = self.driver.find_elements(By.TAG_NAME, value: 'h1')
32      self.assertEqual(first: 1, len(h1_list))
33      self.assertEqual(consts.PAGE_MAIN_TITLE, h1_list[0].text)
34
35      # לבדוק שגאים 2 טבלאות בדיקות
36  ▶  def test_existence_of_two_tables(self):
37      table_list = self.driver.find_elements(By.TAG_NAME, value: 'table')
38      self.assertEqual(first: 2, len(table_list))
39
40      # לבדוק שהכותרת של הטבלה הראשונה מכילה את הטעסט הגדיש
41  ▶  def test_cities_table_name(self):
42      cities_table = self.driver.find_element(By.ID, value: 'citiesOfTheWorldTitle')
43      self.assertEqual(consts.CITIES_TABLE, cities_table.text)
44
45  ▶  def test_students_table_name(self):
46      students_table = self.driver.find_element(By.ID, value: 'studentDetailsTitle')
47      self.assertEqual(consts.STUDENTS_TABLE, students_table.text)

```

⚠ 1 ⚠ 37 ✅ 53 ^

x Automation Project

Automation Project - Main Page

Cities of the World			
USA	Europe	Israel	
Washington, D.C.	London	Jerusalem	
New York City	Manchester	Tel Aviv	
San Francisco	Berlin	Haifa	

Student Details			
ID	First Name	Last Name	Grade
1	John	Smith	A
2	Jane	Doe	B+
3	Alice	Johnson	A-
4	Michael	Williams	B
5	Emily	Brown	A

Cities of the World

Student Details

## TestHome.py ×

```
10  class TestHome(unittest.TestCase):
48
49      # בדיקת 5 תלמידים ושהם נכונים
50  ▶  def test_students_table_content(self):
51      students_name = consts.STUDENTS_FIRST_NAME
52      first_name_column_idx = None
53      table = self.driver.find_element(By.CSS_SELECTOR, value: '#studentsTable')
54      table_body_rows = table.find_elements(By.CSS_SELECTOR, value: 'tbody tr')
55      self.assertEqual(first: 5, len(table_body_rows))
56      names = []
57      for row in table_body_rows:
58          name = row.find_elements(By.TAG_NAME, value: 'td')[1].text
59          names.append(name)
60      self.assertEqual(consts.STUDENTS_FIRST_NAME, names)
61
```

ID	First Name
1	John
2	Jane
3	Alice
4	Michael
5	Emily

# בדיקות - עבודה עם טופס

## Form

### Personal Information Form

First Name

The Last Name

City

Email

Mobile

Gender

 Male |  Female |  Other

Preferred Professions

 Math |  Physics |  Biology |  Chemistry |  English

```
10  class TestHome(unittest.TestCase):  
11  
12      #עבודה עם טסט  
13      2 usages  
14      #פונקציה למציאת אלמנט מהטפט  
15      self.first_name = self.driver.find_element(By.ID, value: "first-name")  
16      self.last_name = self.driver.find_element(By.ID, value: "last-name")  
17      self.email = self.driver.find_element(By.ID, value: "email")  
18      self.mobile = self.driver.find_element(By.ID, value: "mobile")  
19  
20      self.city = self.driver.find_element(By.ID, value: "city")  
21      self.area_code = self.driver.find_element(By.ID, value: "area-code")  
22  
23      self.female_btn = self.driver.find_element(By.ID, value: "female")  
24      self.male_btn = self.driver.find_element(By.ID, value: "male")  
25      self.other_btn = self.driver.find_element(By.ID, value: "other")  
26      self.math_btn = self.driver.find_element(By.ID, value: "math")  
27      self.physics_btn = self.driver.find_element(By.ID, value: "physics")  
28      self.biology_btn = self.driver.find_element(By.ID, value: "biology")  
29      self.chemistry_btn = self.driver.find_element(By.ID, value: "chemistry")  
30      self.english_btn = self.driver.find_element(By.ID, value: "english")  
31
```

```
82     def fill_form(self):|  
83         # פונקציה למילוי פרטיים באולםנווט  
84         self.find_form_elements()  
85         city_dropdown = Select(self.city)  
86         city_dropdown_options = self.driver.find_elements(By.CSS_SELECTOR, value: "#city option")  
87  
88         area_code = Select(self.area_code)  
89         area_code_dropdown_options = self.driver.find_elements(By.CSS_SELECTOR, value: "#area-code option")  
90         selected_idx_city = random.randint(1, len(city_dropdown_options) - 1)  
91         selected_idx_area_code = random.randint(1, len(area_code_dropdown_options) - 1)  
92         self.first_name.send_keys(consts.FORM_DATA.get('first_name'))  
93         sleep(0.5)  
94         self.last_name.send_keys(consts.FORM_DATA.get('last_name'))  
95         sleep(0.5)  
96         self.email.send_keys(consts.FORM_DATA.get('email'))  
97         sleep(0.5)  
98         self.mobile.send_keys(consts.FORM_DATA.get('mobile'))  
99         sleep(0.5)  
100        self.female_btn.click()  
101        sleep(0.5)  
102        self.biology_btn.click()  
103        sleep(0.5)  
104        self.chemistry_btn.click()  
105        sleep(0.5)  
106        city_dropdown.select_by_index(selected_idx_city)  
107        sleep(0.5)  
108        self.assertTrue(city_dropdown_options[selected_idx_city].is_selected())  
109        area_code.select_by_index(selected_idx_area_code)  
110        sleep(3)  
111        self.assertTrue(area_code_dropdown_options[selected_idx_area_code].is_selected())
```

```
10 class TestHome(unittest.TestCase):
113     def check_form(self):
114         #רְגַשֵּׁת מִזְרָבְּלָה
115         self.find_form_elements()
116         first_name_value = self.first_name.get_attribute("value")
117         last_name_value = self.last_name.get_attribute("value")
118         email_value = self.email.get_attribute("value")
119         city_value = self.city.get_attribute("value")
120         area_code_value = self.area_code.get_attribute("value")
121         mobile_value = self.mobile.get_attribute("value")
122         sleep(2)
123         self.assertEqual(first: "", first_name_value)
124         self.assertEqual(first: "", last_name_value)
125         self.assertEqual(first: "", email_value)
126         self.assertEqual(first: "Tel Aviv", city_value)
127         self.assertEqual(first: "050", area_code_value)
128         self.assertEqual(first: "", mobile_value)
129         sleep(2)
130         self.assertFalse(self.male_btn.is_selected())
131         self.assertFalse(self.female_btn.is_selected())
132         self.assertFalse(self.other_btn.is_selected())
133         self.assertFalse(self.math_btn.is_selected())
134         self.assertFalse(self.physics_btn.is_selected())
135         self.assertFalse(self.biology_btn.is_selected())
136         self.assertFalse(self.chemistry_btn.is_selected())
137         self.assertFalse(self.english_btn.is_selected())
138         sleep(5)
```

```
140  
141 ▶ def test_form(self):  
142     # פעולה ששולחת את הטעוס מעווצה מנוקה  
143     submit_form = self.driver.find_element(By.ID, value: "submit-form")  
144     sleep(2)  
145     self.fill_form()  
146  
147     self.driver.save_screenshot("../ScreenShots/1_after_fill_form.png")  
148     submit_form.click()  
149  
150     reset_form = self.driver.find_element(By.ID, value: "reset-form")  
151     sleep(2)  
152     reset_form.click()  
153     self.driver.save_screenshot("../ScreenShots/1_after_reset_form.png")  
154     self.check_form()  
155     sleep(2)  
156
```

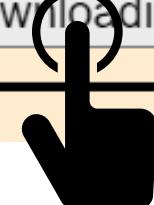
שליחה איפוא

## בדיקה - לחיצה על כפתור לדימוי פעולה הורדת הקובץ

**A demonstration of how to access a PROGRESS element**

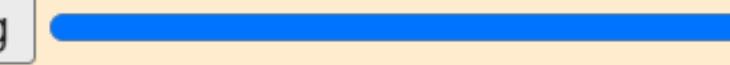
Downloading Progress

Start Downloading  Progress: 0%



**A demonstration of how to access a PROGRESS element**

Downloading Progress

Start Downloading  Progress: 100% Download Finished! 



**A demonstration of how to access a PROGRESS element**

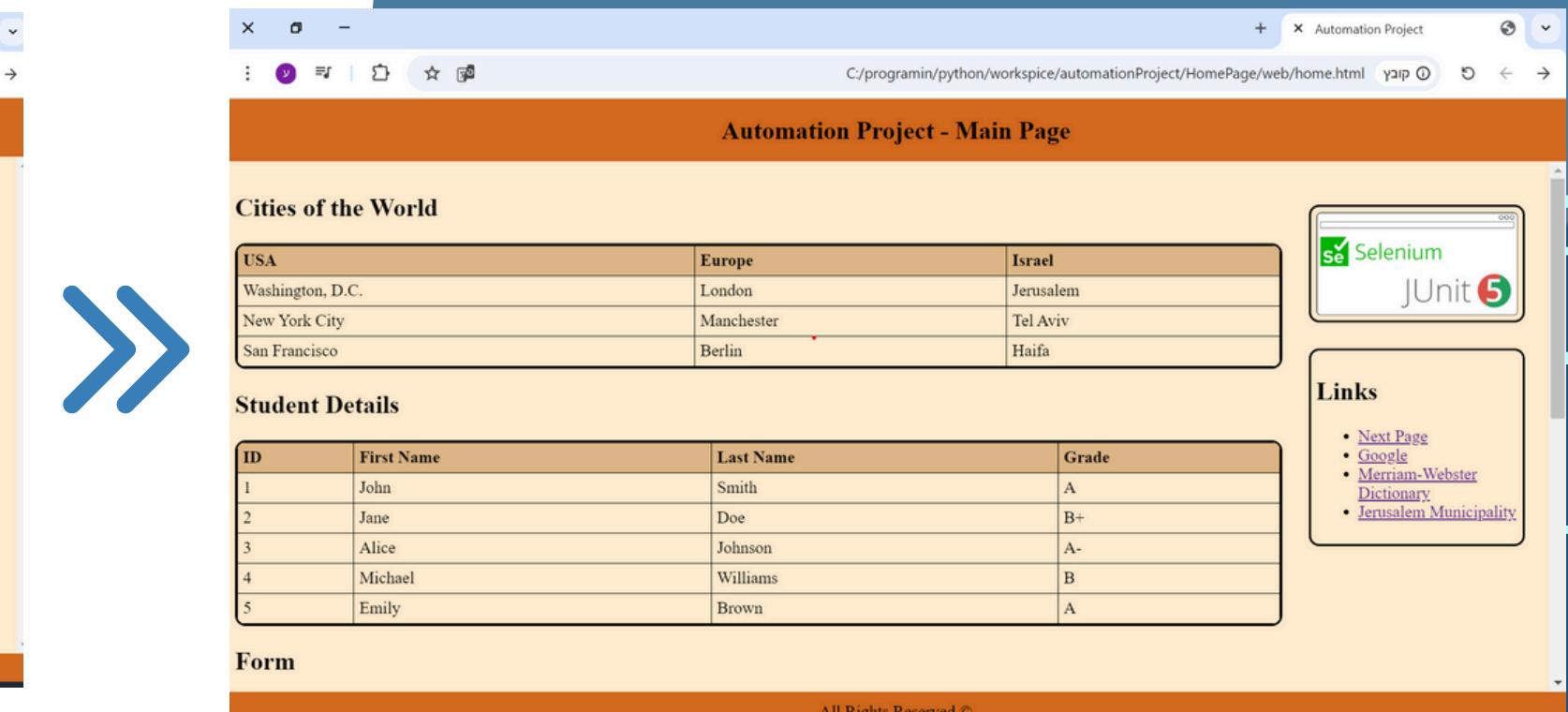
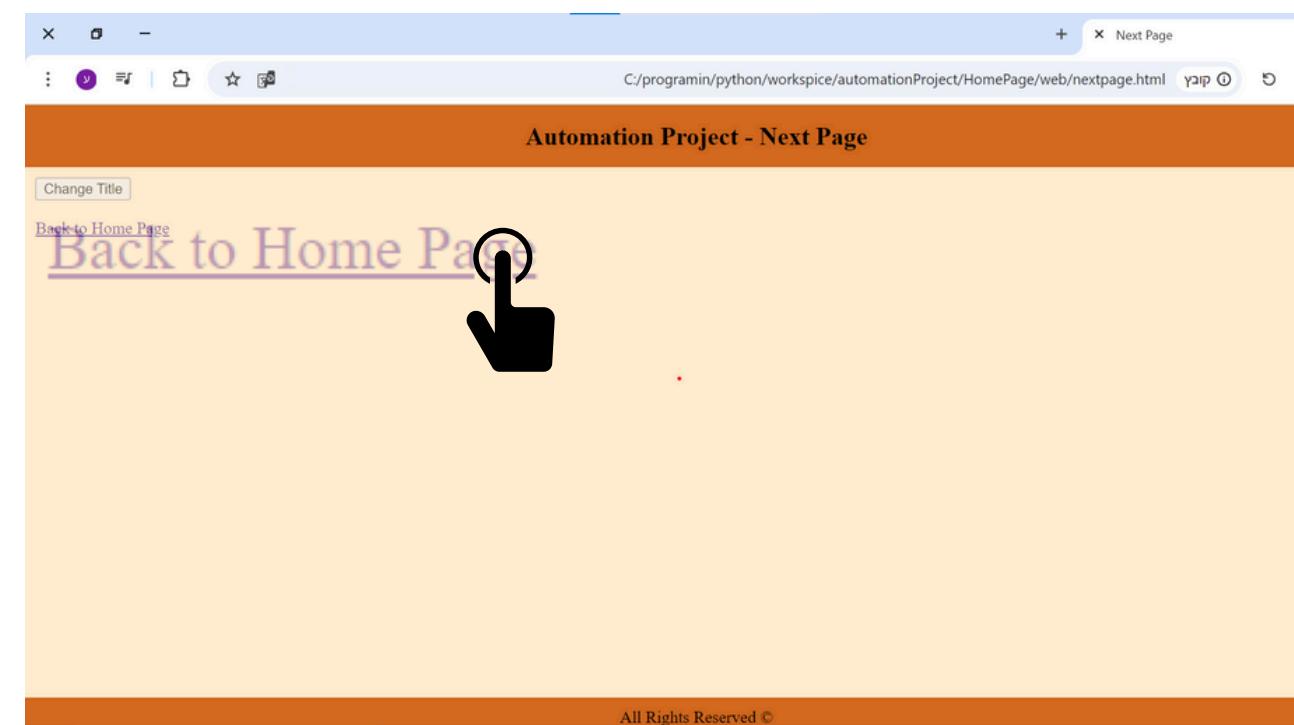
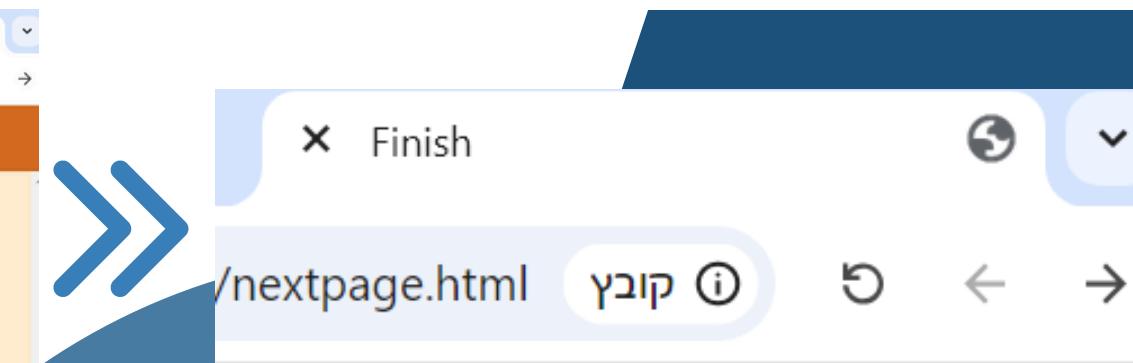
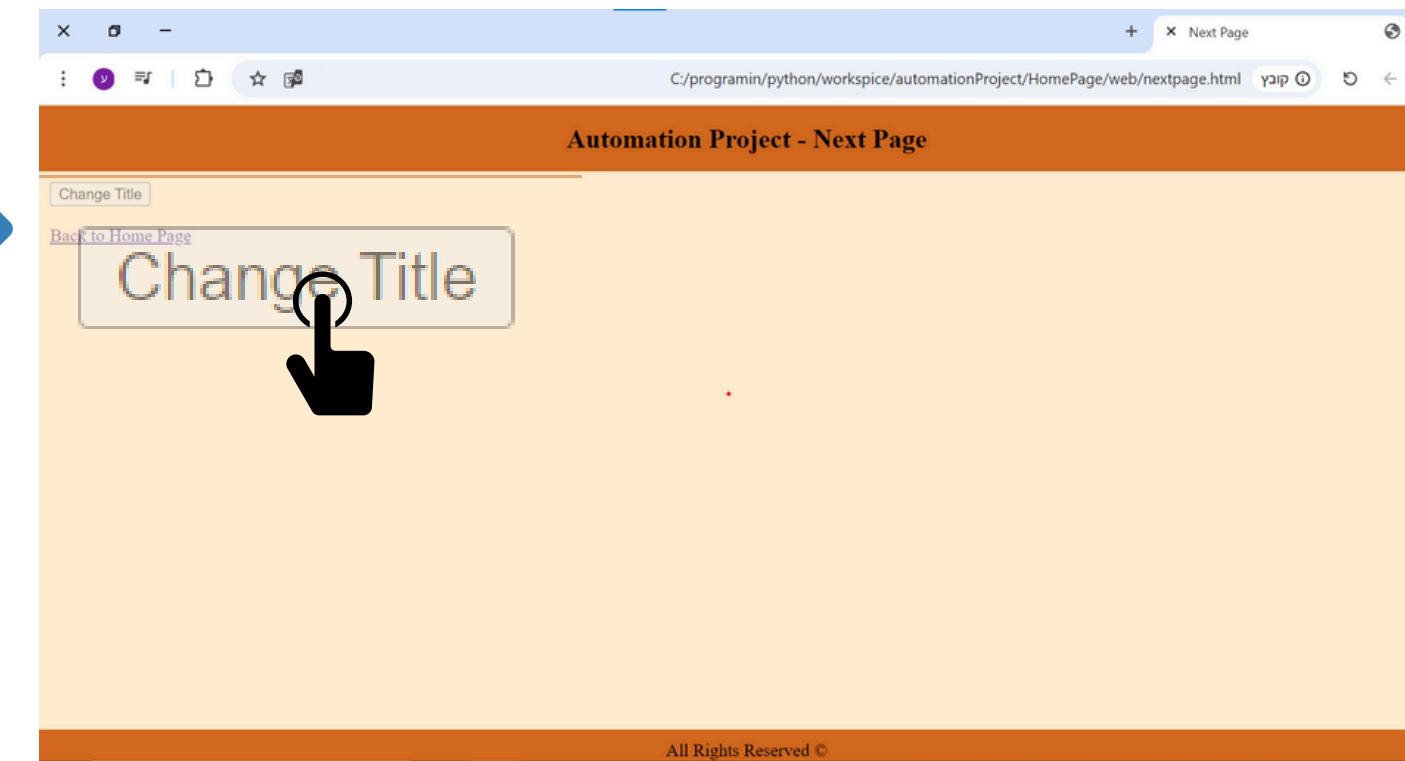
Downloading Progress

Start Downloading  Progress: 0%



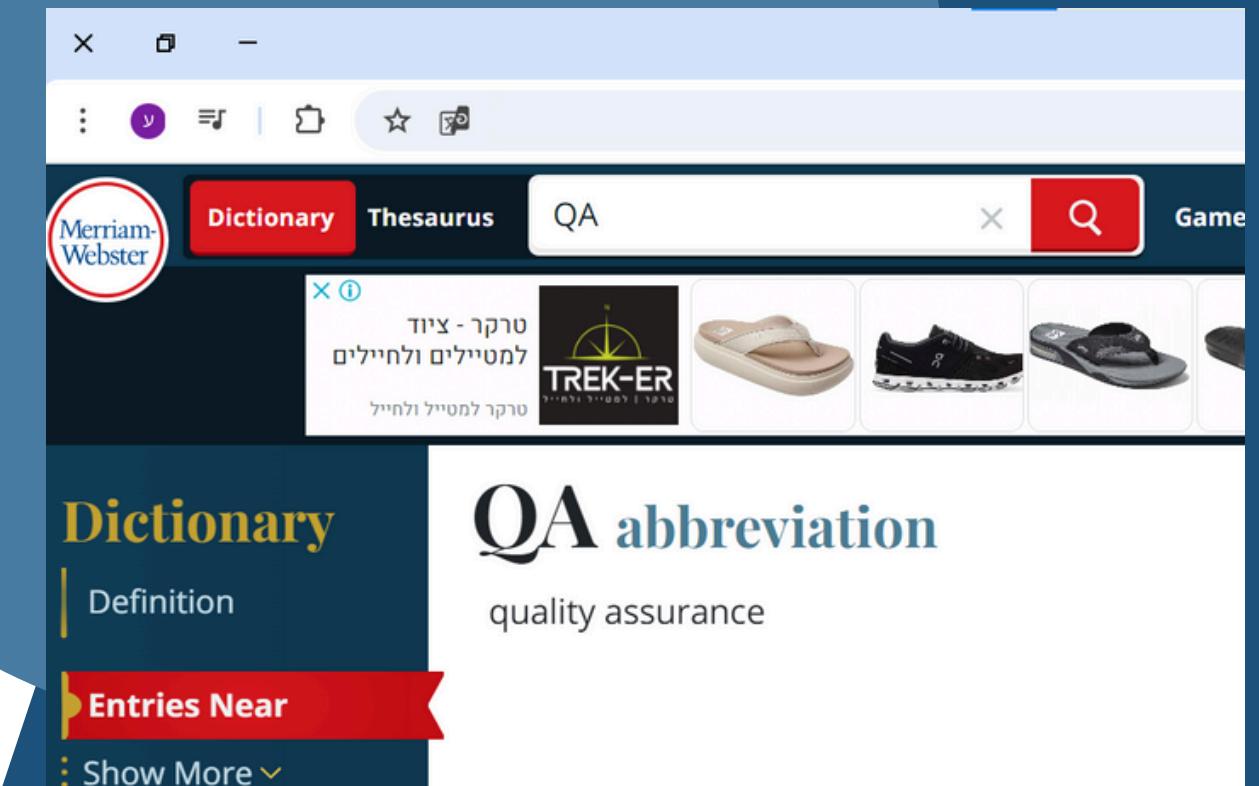
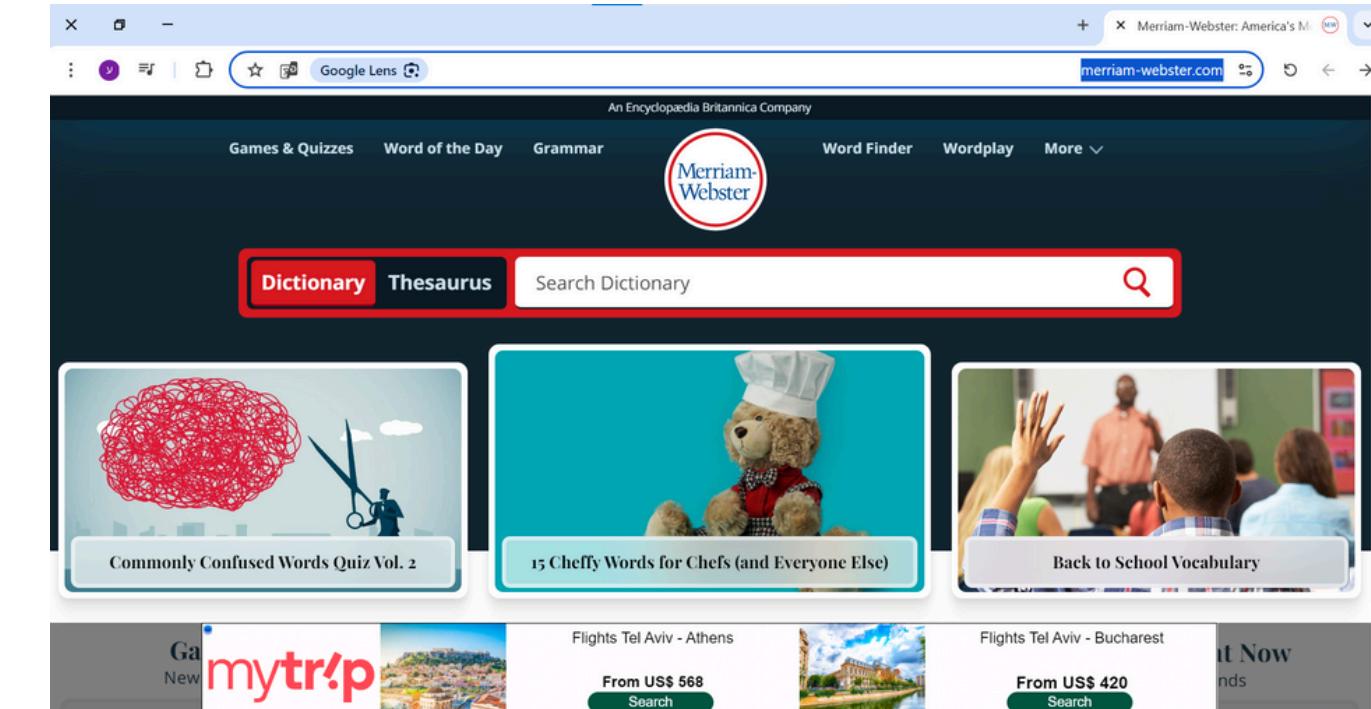
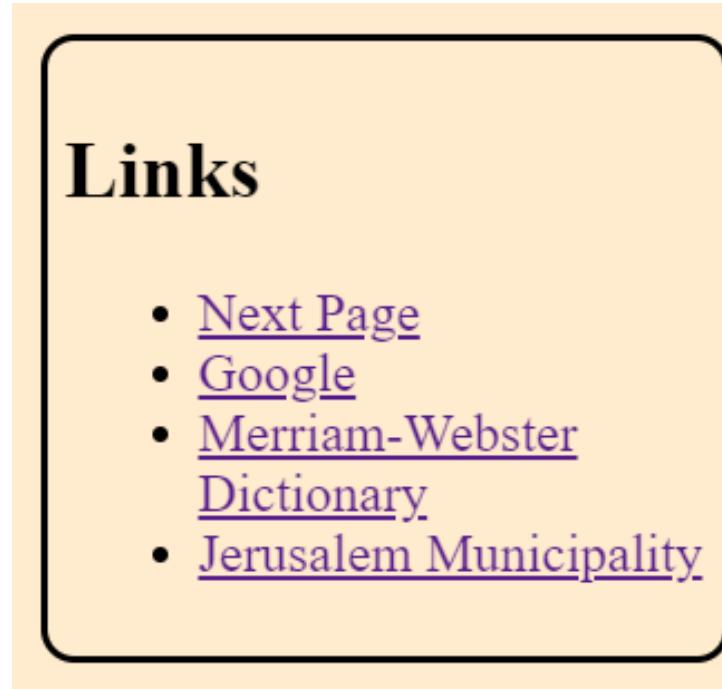
```
158 ▶ def test_download(self):
159     download = self.driver.find_element(By.CSS_SELECTOR, value: "[onclick='startDownload()']")
160
161     self.driver.execute_script( script: 'arguments[0].scrollIntoView()', *args: download)
162
163     self.driver.save_screenshot("../ScreenShots/2_before_download.png")
164     download.click()
165     sleep(2)
166     self.driver.save_screenshot("../ScreenShots/2_after_download_click.png")
167     progress = self.driver.find_element(By.ID, value: "progress-txt")
168     # Wait for 100%
169     while progress.text != '100':
170         pass
171
172     self.driver.save_screenshot("../ScreenShots/2_after_download_finished.png")
173     try:
174         download_finished = self.driver.find_element(By.ID, value: "download-finished-msg")
175
176         self.assertEqual(first: "Download Finished! OK", download_finished.text)
177     except NoSuchElementException:
178         print("The 'Download Finished' element is not found")
179         self.assertTrue(False)
180
181     ok = self.driver.find_element(By.CSS_SELECTOR, value: "#download-finished-msg>button")
182     sleep(5)
183     ok.click()
184
185     self.driver.save_screenshot("../ScreenShots/2_download_after_ok_click.png")
186     # Check that the message disappeared
187     self.assertEqual(first: "", download_finished.text)
```

# בדיקות הדף הבא - Next Page



```
10  class TestHome(unittest.TestCase):  
189 >  def test_next_page(self):  
190      next_page_link = self.driver.find_element(By.LINK_TEXT, value: 'Next Page')  
191      next_page_link.click()  
192      sleep(2)  
193      self.driver.save_screenshot("../ScreenShots/3_after_next_page.png")  
194  
195      self.assertEqual(consts.NEXT_PAGE_URL, self.driver.current_url)  
196  
197      self.assertEqual(first: "Next Page", self.driver.title)  
198  
199      change_title = self.driver.find_element(By.CSS_SELECTOR, value: "[onclick='changeTitle(); ']")  
200      change_title.click()  
201      sleep(5)  
202      self.driver.save_screenshot("../ScreenShots/3_after_change_title.png")  
203  
204      self.assertEqual(first: "Finish", self.driver.title)  
205  
206      back_to_home = self.driver.find_element(By.LINK_TEXT, value: "Back to Home Page")  
207      back_to_home.click()  
208      sleep(5)  
209      self.driver.save_screenshot("../ScreenShots/3_after_back_to_home.png")  
210  
211      self.assertEqual(consts.BASE_URL, self.driver.current_url)
```

# משימת בונוס - Merriam-Webster Dictionary



**QA abbreviation**  
quality assurance



```
1 import time
2 import unittest
3 from selenium import webdriver
4 from selenium.webdriver.common.keys import Keys
5 from selenium.webdriver.common.by import By
6 from HomePage.config import consts
7
8 ▶ class Dictionary(unittest.TestCase):
9     driver = None
10
11     @classmethod
12     @↑ def setUpClass(cls):
13         cls.driver = webdriver.Chrome()
14         cls.driver.maximize_window()
15
16     @classmethod
17     @↑ def tearDownClass(cls):
18         cls.driver.quit()
19
20     @↑ def setUp(self):
21         self.driver.get(consts.BASE_URL)
22         time.sleep(2)
23
```

dictionary\_test.py ×

⚠ 6 ✅ 10 ⌂ ⌃

```
8 class Dictionary(unittest.TestCase):
9
10    def test_dictionary(self):
11
12        # מילון QA זמין על הלינק ומציננו את הערך
13        dictionary_page_link = self.driver.find_element(By.LINK_TEXT, value: 'Merriam-Webster Dictionary')
14        dictionary_page_link.click()
15        time.sleep(3)
16
17        self.assertEqual(consts.DICTIONARY_PAGE_URL, self.driver.current_url)
18
19
20        search_line = self.driver.find_element(By.ID, value: "home-search-term")
21        search_line.send_keys("QA")
22        self.driver.save_screenshot("../ScreenShots/dictionary_QA.png")
23        search_line.send_keys(Keys.ENTER)
24        result = self.driver.find_element(By.CSS_SELECTOR, value: ".dt>.dtText")
25        self.driver.save_screenshot("../ScreenShots/dictionary_result.png")
26        self.assertEqual(first: "quality assurance", result.text)
```

# машימת בונוס - Calculator.net

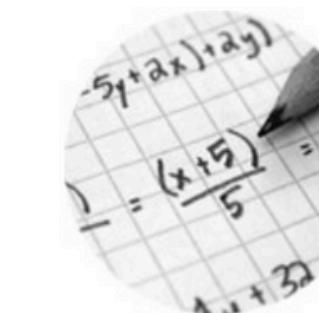
The screenshot shows the homepage of Calculator.net. At the top, there is a search bar with the placeholder "calculator.net" and a "Search" button. Below the search bar, the text "Free Online Calculators" is displayed. On the left side, there is a large image of a scientific calculator with a numeric keypad and various mathematical functions like sin, cos, tan, and exponential operations. To the right of the calculator image, there are four circular icons representing different calculator categories: Financial Calculators (coins and a graph), Fitness & Health Calculators (a measuring tape and fitness equipment), Math Calculators (a notepad with mathematical equations), and Other Calculators (a close-up of a calculator screen). The background features a blue and white color scheme with abstract circuit board patterns.



Financial Calculators



Fitness & Health Calculators



Math Calculators



Other Calculators



```
8     class Calculator(unittest.TestCase):
24         def test_calc_result(self):
25
26         # בונקציה של חישוב נל 7 ו-5 ומחזב את הסכום#
27         num7 = self.driver.find_element(By.CSS_SELECTOR, value: "[onclick='r(7)']")
28         num5 = self.driver.find_element(By.CSS_SELECTOR, value: "[onclick='r(5)']")
29         mult_operator = self.driver.find_element(By.XPATH, value: "//*[text()='+']")
30         eq_operator = self.driver.find_element(By.XPATH, value: "//*[text()']='='")
31         result = self.driver.find_element(By.ID, value: "sciOutPut")
32
33         num7.click()
34         sleep(0.5)
35         mult_operator.click()
36         sleep(0.5)
37         num5.click()
38         sleep(0.5)
39
40         eq_operator.click()
41         sleep(0.5)
42
43         # ScreenShot
44         self.driver.save_screenshot('..../ScreenShots/calculator_result.png')
45
46         self.assertEqual(first: " 12", result.text)
47
```

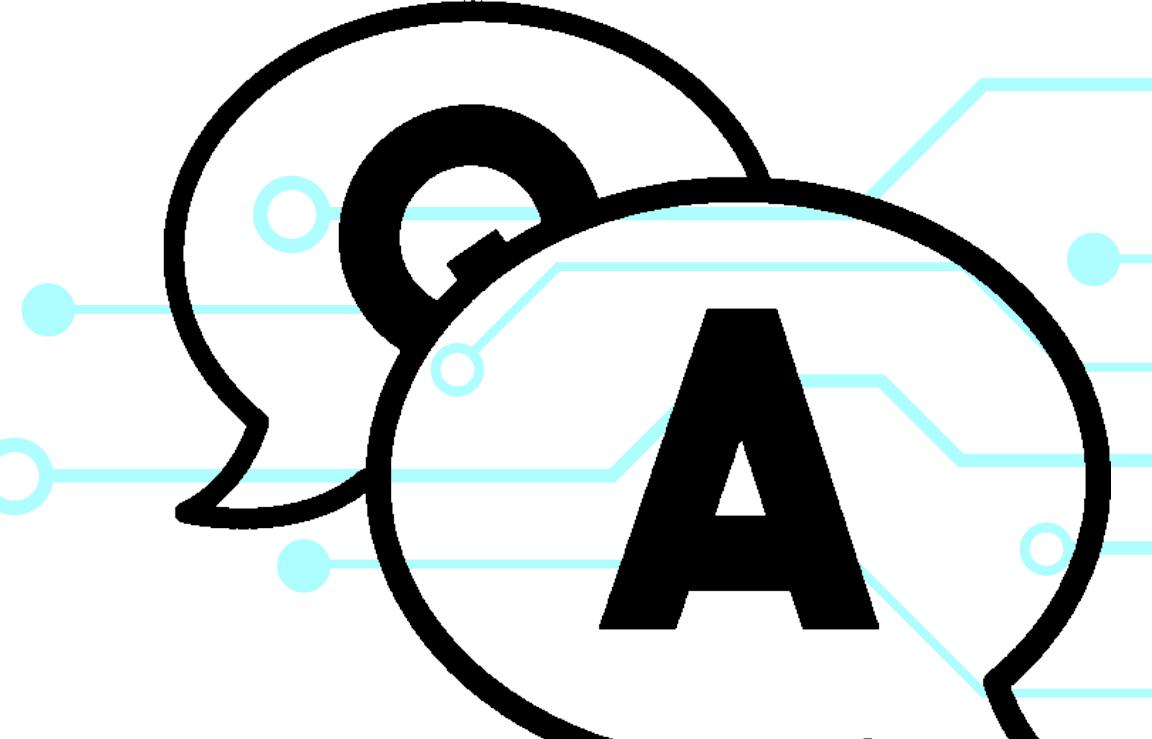
## כעת ניגש לצפות בהערכת הטסטים בזמן אמת



# ANY QUESTions?



תודה על ההקשבה!


Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14	Column 15	Column 16	Column 17	Column 18	Column 19	Column 20	Column 21	Column 22	Column 23	Column 24	Column 25	Column 26	Column 27	Column 28	Column 29	Column 30	Column 31	Column 32	Column 33	Column 34	Column 35	Column 36	Column 37	Column 38	Column 39	Column 40	Column 41	Column 42	Column 43	Column 44	Column 45	Column 46	Column 47	Column 48	Column 49	Column 50	Column 51	Column 52	Column 53	Column 54	Column 55	Column 56	Column 57	Column 58	Column 59	Column 60	Column 61	Column 62	Column 63	Column 64	Column 65	Column 66	Column 67	Column 68	Column 69	Column 70	Column 71	Column 72	Column 73	Column 74	Column 75	Column 76	Column 77	Column 78	Column 79	Column 80	Column 81	Column 82	Column 83	Column 84	Column 85	Column 86	Column 87	Column 88	Column 89	Column 90	Column 91	Column 92	Column 93	Column 94	Column 95	Column 96	Column 97	Column 98	Column 99	Column 100	Column 101	Column 102	Column 103	Column 104	Column 105	Column 106	Column 107	Column 108	Column 109	Column 110	Column 111	Column 112	Column 113	Column 114	Column 115	Column 116	Column 117	Column 118	Column 119	Column 120	Column 121	Column 122	Column 123	Column 124	Column 125	Column 126	Column 127	Column 128	Column 129	Column 130	Column 131	Column 132	Column 133	Column 134	Column 135	Column 136	Column 137	Column 138	Column 139	Column 140	Column 141	Column 142	Column 143	Column 144	Column 145	Column 146	Column 147	Column 148	Column 149	Column 150	Column 151	Column 152	Column 153	Column 154	Column 155	Column 156	Column 157	Column 158	Column 159	Column 159	Column 160	Column 161	Column 162	Column 163	Column 164	Column 165	Column 166	Column 167	Column 168	Column 169	Column 170	Column 171	Column 172	Column 173	Column 174	Column 175	Column 176	Column 177	Column 178	Column 179	Column 180	Column 181	Column 182	Column 183	Column 184	Column 185	Column 186	Column 187	Column 188	Column 189	Column 190	Column 191	Column 192	Column 193	Column 194	Column 195	Column 196	Column 197	Column 198	Column 199	Column 200	Column 201	Column 202	Column 203	Column 204	Column 205	Column 206	Column 207	Column 208	Column 209	Column 210	Column 211	Column 212	Column 213	Column 214	Column 215	Column 216	Column 217	Column 218	Column 219	Column 220	Column 221	Column 222	Column 223	Column 224	Column 225	Column 226	Column 227	Column 228	Column 229	Column 230	Column 231	Column 232	Column 233	Column 234	Column 235	Column 236	Column 237	Column 238	Column 239	Column 240	Column 241	Column 242	Column 243	Column 244	Column 245	Column 246	Column 247	Column 248	Column 249	Column 250	Column 251	Column 252	Column 253	Column 254	Column 255	Column 256	Column 257	Column 258	Column 259	Column 260	Column 261	Column 262	Column 263	Column 264	Column 265	Column 266	Column 267	Column 268	Column 269	Column 270	Column 271	Column 272	Column 273	Column 274	Column 275	Column 276	Column 277	Column 278	Column 279	Column 280	Column 281	Column 282	Column 283	Column 284	Column 285	Column 286	Column 287	Column 288	Column 289	Column 290	Column 291	Column 292	Column 293	Column 294	Column 295	Column 296	Column 297	Column 298	Column 299	Column 300	Column 301	Column 302	Column 303	Column 304	Column 305	Column 306	Column 307	Column 308	Column 309	Column 310	Column 311	Column 312	Column 313	Column 314	Column 315	Column 316	Column 317	Column 318	Column 319	Column 320	Column 321	Column 322	Column 323	Column 324	Column 325	Column 326	Column 327	Column 328	Column 329	Column 330	Column 331	Column 332	Column 333	Column 334	Column 335	Column 336	Column 337	Column 338	Column 339	Column 340	Column 341	Column 342	Column 343	Column 344	Column 345	Column 346	Column 347	Column 348	Column 349	Column 350	Column 351	Column 352	Column 353	Column 354	Column 355	Column 356	Column 357	Column 358	Column 359	Column 360	Column 361	Column 362	Column 363	Column 364	Column 365	Column 366	Column 367	Column 368	Column 369	Column 370	Column 371	Column 372	Column 373	Column 374	Column 375	Column 376	Column 377	Column 378	Column 379	Column 380	Column 381	Column 382	Column 383	Column 384	Column 385	Column 386	Column 387	Column 388	Column 389	Column 390	Column 391	Column 392	Column 393	Column 394	Column 395	Column 396	Column 397	Column 398	Column 399	Column 400	Column 401	Column 402	Column 403	Column 404	Column 405	Column 406	Column 407	Column 408	Column 409	Column 410	Column 411	Column 412	Column 413	Column 414	Column 415	Column 416	Column 417	Column 418	Column 419	Column 420	Column 421	Column 422	Column 423	Column 424	Column 425	Column 426	Column 427	Column 428	Column 429	Column 430	Column 431	Column 432	Column 433	Column 434	Column 435	Column 436	Column 437	Column 438	Column 439	Column 440	Column 441	Column 442	Column 443	Column 444	Column 445	Column 446	Column 447	Column 448	Column 449	Column 450	Column 451	Column 452	Column 453	Column 454	Column 455	Column 456	Column 457	Column 458	Column 459	Column 460	Column 461	Column 462	Column 463	Column 464	Column 465	Column 466	Column 467	Column 468	Column 469	Column 470	Column 471	Column 472	Column 473	Column 474	Column 475	Column 476	Column 477	Column 478	Column 479	Column 480	Column 481	Column 482	Column 483	Column 484	Column 485	Column 486	Column 487	Column 488	Column 489	Column 490	Column 491	Column 492	Column 493	Column 494	Column 495	Column 496	Column 497	Column 498	Column 499	Column 500	Column 501	Column 502	Column 503	Column 504	Column 505	Column 506	Column 507	Column 508	Column 509	Column 510	Column 511	Column 512	Column 513	Column 514	Column 515	Column 516	Column 517	Column 518	Column 519	Column 520	Column 521	Column 522	Column 523	Column 524	Column 525	Column 526	Column 527	Column 528	Column 529	Column 530	Column 531	Column 532	Column 533	Column 534	Column 535	Column 536	Column 537	Column 538	Column 539	Column 540	Column 541	Column 542	Column 543	Column 544	Column 545	Column 546	Column 547	Column 548	Column 549	Column 550	Column 551	Column 552	Column 553	Column 554	Column 555	Column 556	Column 557	Column 558	Column 559	Column 560	Column 561	Column 562	Column 563	Column 564	Column 565	Column 566	Column 567	Column 568	Column 569	Column 570	Column 571	Column 572	Column 573	Column 574	Column 575	Column 576	Column 577	Column 578	Column 579	Column 580	Column 581	Column 582	Column 583	Column 584	Column 585	Column 586	Column 587	Column 588	Column 589	Column 590	Column 591	Column 592	Column 593	Column 594	Column 595	Column 596	Column 597	Column 598	Column 599	Column 600	Column 601	Column 602	Column 603	Column 604	Column 605	Column 606	Column 607	Column 608	Column 609	Column 610	Column 611	Column 612	Column 613	Column 614	Column 615	Column 616	Column 617	Column 618	Column 619	Column 620	Column 621	Column 622	Column 623	Column 624	Column 625	Column 626	Column 627	Column 628	Column 629	Column 630	Column 631	Column 632	Column 633	Column 634	Column 635	Column 636	Column 637	Column 638	Column 639	Column 640	Column 641	Column 642	Column 643	Column 644	Column 645	Column 646	Column 647	Column 648	Column 649	Column 650	Column 651	Column 652	Column 653	Column 654	Column 655	Column 656	Column 657	Column 658	Column 659	Column 660	Column 661	Column 662	Column 663	Column 664	Column 665	Column 666	Column 667	Column 668	Column 669	Column 670	Column 671	Column 672	Column 673	Column 674	Column 675	Column 676	Column 677	Column 678	Column 679	Column 680	Column 681	Column 682	Column 683	Column 684	Column 685	Column 686	Column 687	Column 688	Column 689	Column 690	Column 691	Column 692	Column 693	Column 694	Column 695	Column 696	Column 697	Column 698	Column 699	Column 700	Column 701	Column 702	Column 703	Column 704	Column 705	Column 706	Column 707	Column 708	Column 709	Column 710	Column 711	Column 712	Column 713	Column 714	Column 715	Column 716	Column 717	Column 718	Column 719	Column 720	Column 721	Column 722	Column 723	Column 724	Column 725	Column 726	Column 727	Column 728	Column 729	Column 730	Column 731	Column 732	Column 733	Column 734	Column 735	Column 736	Column 737	Column 738	Column 739	Column 740	Column 741	Column 742	Column 743	Column 744	Column 745	Column 746	Column 747	Column 748	Column 749	Column 750	Column 751	Column 752	Column 753	Column

