



Software Testing Project

Selenium (React.js BMI Calculator App)

Nima Moeintaghavi
Mohammad Mahdi Rahneshin

9822762211
9812762145



Project being tested:

<https://github.com/GermaVinsmoke/bmi-calculator>

Selenium project:

<https://github.com/nimamt/selenium>

Run the BMI-calculator project and type:

python main.py (Windows) / python3 main.py (Linux/Mac)

مقدمه

برنامه مقدار BMI را محاسبه می کند و می توان این مقادیر را در یک لیست در پایین صفحه مشاهده کرد.

همچنین، با استفاده از یک نمودار در وسط صفحه می توان تاریخچه مقادیر را به همراه مقادیر دقیق مشاهده کرد و تغییرات را با چشم مشاهده کرد.

علاوه بر این Document، تمام تست ها و دسته ها کامنت گذاری شده اند.

تست ها

7 تست در مجموع طراحی شده است، 3 اشکال کشف شده، و بقیه تست ها برای بررسی عملکرد برنامه و تست رگرسیون مناسب هستند.

در کل می توان تست ها را به دو دسته تقسیم کرد که این تقسیم بندی در کد قابل مشاهده است.

دسته اول تست های محاسبه BMI هستند که اشکال ها در این بخش می باشند. دسته دوم تست های بررسی نگهداری مقادیر محاسبه شده برای BMI ها هستند.

دسته اول

برنامه محدودیتی برای مقادیر ورودی قرار داده است که برای هر کدام مقدار بیشتر از 999 نمی توان قرار داد.

در این تست کیس ها برنامه را به چالش می کشیم و به این نتیجه می رسیم که اشکال های متعددی دارد.

تست 1

```
def test_bmi_nan(self):
    """Test an invalid BMI result that should result
    in NaN and should not be accepted."""
    self.weight.send_keys('0')
    self.height.send_keys('0')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CSS_SELECTOR, ".data-container .col")
    self.assertEqual(len(container_items), 0)
```

عملیات: مقدار 0 تقسیم بر 0 را ایجاد می کنیم. نتیجه مورد انتظار: BMI به لیست نتایج اضافه نگردد

تست 2

```
def test_bmi_inf(self):
    """Test an invalid BMI result that should result
    in Inf and should not be accepted."""
    self.weight.send_keys('10')
    self.height.send_keys('0')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CSS_SELECTOR, ".data-container .col")
    self.assertEqual(len(container_items), 0)
```

عملیات: مقدار عدد بزرگتر از 0 تقسیم بر 0 را ایجاد می کنیم.

نتیجه مورد انتظار: BMI به لیست نتایج اضافه نگردد

```
class BMICalculation(unittest.TestCase):
    """A sample test class to test how the BMI calculation works"""
```

```
def setup(self):
    self.driver = webdriver.Chrome(
        "C:\Program Files (x86)\chromedriver.exe")
    self.driver.maximize_window()
    self.driver.delete_all_cookies()
    self.driver.get("http://localhost:3000")
    time.sleep(1)
    self.weight = self.driver.find_element(By.ID, "weight")
    self.height = self.driver.find_element(By.ID, "height")
    self.calculate_btn = self.driver.find_element(
        By.CLASS_NAME, "calculate-btn")
```

```
def tearDown(self):
    self.driver.close()
```



تست 3

```
def setUp(self):
    self.driver = webdriver.Chrome(
        "C:\\Program Files (x86)\\chromedriver.exe")
    self.driver.maximize_window()
    self.driver.delete_all_cookies()
    self.driver.get("http://localhost:3000")
    time.sleep(3)
    self.weight = self.driver.find_element(By.ID, "weight")
    self.height = self.driver.find_element(By.ID, "height")
    self.calculate_btn = self.driver.find_element(
        By.CLASS_NAME, "calculate-btn")
```

```
def tearDown(self):
    self.driver.close()
```

تست 1

عملیات: مقادیر درست و عادی وارد می کنیم
و پس از اضافه شدن نتیجه به لیست نتایج آن
را با دکمه مربوطه حذف می کنیم.
نتیجه مورد انتظار: نتیجه به درستی حذف
گردد.

```
def test_bmi_normal_add_remove(self):
    """Test whether the delete functionality works
    correctly"""
    self.weight.send_keys('80')
    self.height.send_keys('185')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    del_btn = self.driver.find_element(By.CLASS_NAME,
        "delete-btn")
    del_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CSS_SELECTOR, ".data-container .col")
    self.assertEqual(len(container_items), 0)
```

تست 2

```
def test_bmi_undo_hidden(self):
    """Test whether the undo button remains hidden
    until the right time."""
    self.weight.send_keys('80')
    self.height.send_keys('185')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CLASS_NAME, "calculate-btn")
    self.assertEqual(len(container_items), 1)
```

عملیات: دکمه Undo نباید تا موقعی که حذفی
انجام نشده نباید قابل مشاهده باشد.
نتیجه مورد انتظار: نتوان این دکمه را مشاهده
کرد

```
def test_bmi_zero(self):
    """Test an invalid BMI result that should result
    in 0 and should not be accepted."""
    self.weight.send_keys('0')
    self.height.send_keys('10')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CSS_SELECTOR, ".data-container .col")
    self.assertEqual(len(container_items), 0)
```

عملیات: مقدار 0 تقسیم بر عدد بزرگتر از 0 را
ایجاد می کنیم.
نتیجه مورد انتظار: BMI به لیست نتایج اضافه
نگردد

تست 4

```
def test_bmi_normal(self):
    """Test a valid BMI score."""
    self.weight.send_keys('80')
    self.height.send_keys('185')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CSS_SELECTOR, ".data-container .col")
    self.assertEqual(len(container_items), 1)
```

عملیات: مقدار عدد بزرگتر از صفر تقسیم بر
عدد بزرگتر از صفر را ایجاد می کنیم.
نتیجه مورد انتظار: BMI به لیست نتایج اضافه
گردد

دسته دوم

در این قسمت بررسی می کنیم که آیا فرایند
ذخیره سازی، حذف و Undo کردن به درستی
انجام می گردد یا خیر.

```
class BMIStorage(unittest.TestCase):
    """A sample test class to test how the BMI results storage works"""
```




```
def test_bmi_undo_visible(self):
    """Test whether the undo button is being shown in the
    right time."""
    self.weight.send_keys('80')
    self.height.send_keys('185')
    time.sleep(1)
    self.calculate_btn.click()
    time.sleep(1)
    del_btn = self.driver.find_element(By.CLASS_NAME,
                                       "delete-btn")
    del_btn.click()
    time.sleep(1)
    container_items = self.driver.find_elements(
        By.CLASS_NAME, "calculate-btn")
    self.assertEqual(len(container_items), 2)
```

عملیات: زمانی که یک حذف انجام شده باشد
نتیجه مورد انتظار: دکمه Undo قابل مشاهده
باشد.

تشکر و قدردانی
از توجه شما سپاسگذاریم.

