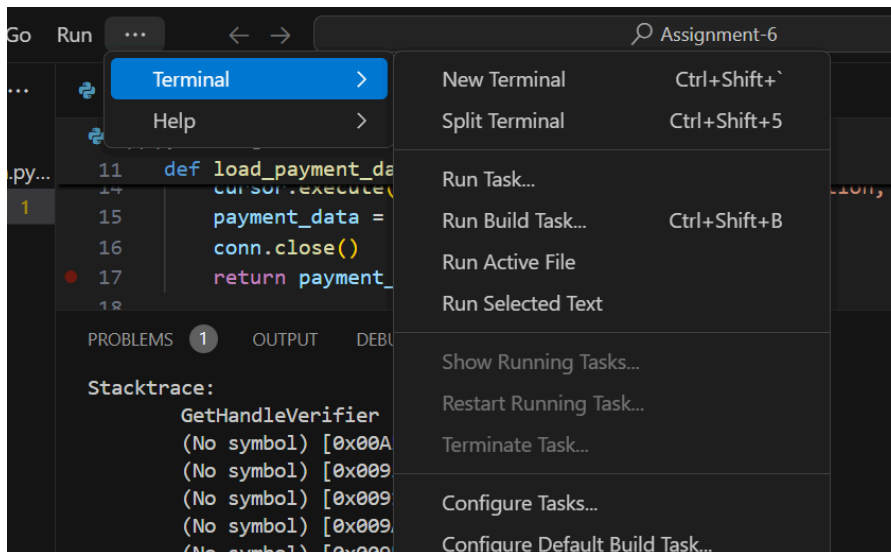


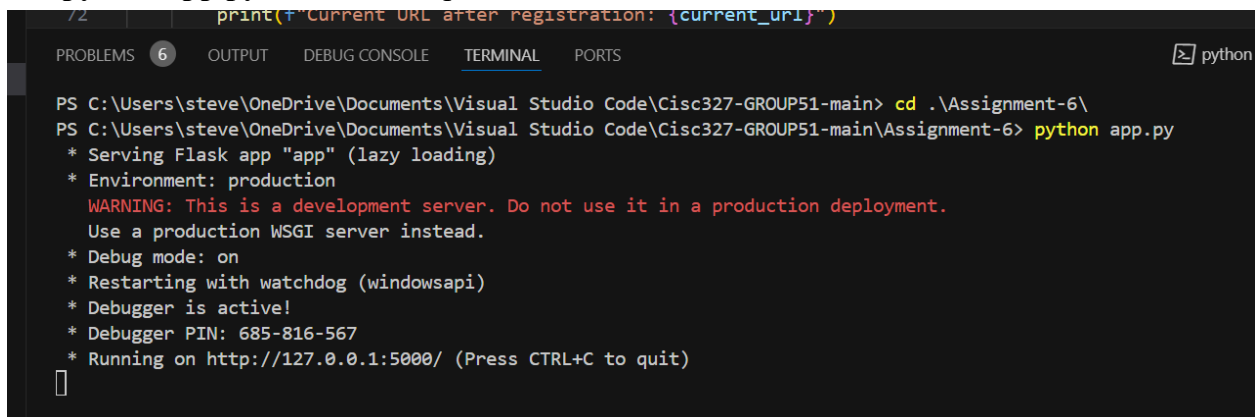
Running the test_e2e_registration.py - Steven Guan

- This end to end test simulates user registration in 4 circumstances: 1-successful registration, 2-registration using a duplicate email, 3-mismatching password for verification and 4-having missing fields in the form.

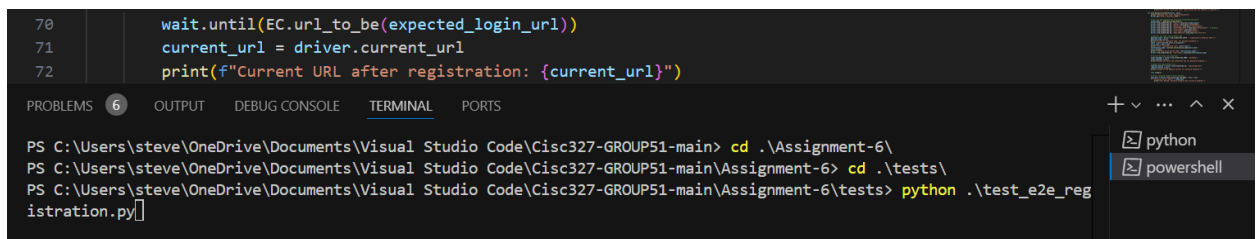
1. Install: **pip install selenium webdriver-manager** - in your terminal
2. Create two new terminals



3. In the first terminal: make the assignment6 folder the current working directory and then run: **python app.py** - in order to open a connection to the local host



4. In the second terminal change the working directory to the “end_to_end_tests” folder which is located in the tests directory and run **python test_e2e_registration.py**



5. The results should look as so:

```
PS C:\Users\steve\OneDrive\Documents\Visual Studio Code\Cisc327-GROUP51-main> cd .\Assignment-5\  
PS C:\Users\steve\OneDrive\Documents\Visual Studio Code\Cisc327-GROUP51-main\Assignment-5> cd .\tests\  
PS C:\Users\steve\OneDrive\Documents\Visual Studio Code\Cisc327-GROUP51-main\Assignment-5\tests> python .\sel_test_registration.py  
  
DevTools listening on ws://127.0.0.1:52385/devtools/browser/50902998-410f-46e8-a4a2-fa1b3ad4ef91  
  
Running Test Case: Successful Registration  
Clicked on 'Generate Code'.  
Alert received with text: Your verification code is: 848011  
Captured Verification Code: 848011  
Checked the Terms and Conditions box.  
Clicked on the Register button.  
Test Passed: Registration was successful.  
  
Running Test Case: Duplicate Email Registration  
Clicked on 'Generate Code'.  
Alert received with text: Your verification code is: 824316  
Captured Verification Code: 824316  
Checked the Terms and Conditions box.  
Clicked on the Register button.  
Clicked on 'Generate Code' for duplicate registration.  
Alert received with text: Your verification code is: 178265  
Captured Verification Code: 178265  
Checked the Terms and Conditions box for duplicate registration.  
Clicked on the Register button for duplicate registration.  
Test Passed: Duplicate email registration was correctly handled.  
  
Running Test Case: Password Mismatch  
Clicked on 'Generate Code' for password mismatch.  
Alert received with text: Your verification code is: 829853  
Captured Verification Code: 829853  
Checked the Terms and Conditions box for password mismatch.  
Clicked on the Register button for password mismatch.  
Test Passed: Password mismatch was correctly handled.  
  
Running Test Case: Missing Required Fields  
Created TensorFlow Lite XNNPACK delegate for CPU.  
Clicked on 'Generate Code' for missing fields.  
Alert received with text: Your verification code is: 490093  
Captured Verification Code: 490093  
Did not check the Terms and Conditions box.  
Clicked on the Register button.  
Attempting to use a delegate that only supports static-sized tensors with a graph that has dynamic-sized tensors (tensor#58 is a dynamic-sized tensor).  
Test Passed: Missing required fields were correctly handled.  
PS C:\Users\steve\OneDrive\Documents\Visual Studio Code\Cisc327-GROUP51-main\Assignment-5\tests> |
```

Running the test_e2e_login.py - Hassan Hamid

- This end-to-end test is going to simulate a user logging in. It contains four tests: logging in with correct information, logging in with invalid email, logging in with invalid password, and finally testing that the forgot password page loads properly.

1. If you've already launched the application after testing **test_e2e_registration.py**, simply run **python test_e2e_login.py** in the same terminal as where you ran the first test. This should be the result:

```
Test Passed: Missing Required Fields were correctly handled.  
● (base) hassan@MacBookAir end_to_end_tests % python test_e2e_login.py  
Forgot password page load test passed!  
.Invalid email test passed!  
.Invalid password test passed!  
.Valid login test passed!  
.  
-----  
Ran 4 tests in 5.915s  
  
OK
```

2. If you haven't ran the first test yet, install: **pip install selenium webdriver-manager** - in your terminal
3. Create two new terminals
4. In the first terminal: make the assignment6 folder the current working directory and then run: **python app.py** - in order to open a connection to the local host
5. In the second terminal change the working directory to the "end_to_end_tests" folder which is located in the tests directory and run **python test_e2e_login.py**

Running the test_e2e_payment.py - Sungmoon Choi

- This end-to-end test is to simulate real user inputting the payment information and it contains 3 tests: valid payment with sufficient fund, invalid payment, valid payment but insufficient fund
1. Make sure that selenium is installed before running the test, if not just run "**pip install selenium**"

```
C:\Users\Sungpoon\Documents\GitHub\Cisc327-GROUP51\Assignment-6>pip install selenium
Requirement already satisfied: selenium in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (4.27.1)
Requirement already satisfied: urllib3<3,>=1.26 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (2.1.0)
Requirement already satisfied: trio~=0.17 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from selenium) (0.27.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from selenium) (2023.11.17)
Requirement already satisfied: typing_extensions~=4.9 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from selenium) (4.12.2)
Requirement already satisfied: websocket-client~=1.8 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from selenium) (1.8.0)
Requirement already satisfied: attrs>=23.2.0 in c:\users\sungpoon\appdata\local\programs\python\python312\lib\site-packages (from trio~=0.17->selenium) (24.2.0)
```

2. Go to Assignment-6 folder directory and run "**python app.py**" to launch the server

```
C:\Users\Sungpoon\Documents\GitHub\Cisc327-GROUP51\Assignment-6>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 580-871-992
```

3. In another terminal, change the directory to end_to_end_tests folder inside tests folder

```
C:\Users\Sungpoon>cd C:\Users\Sungpoon\Documents\GitHub\Cisc327-GROUP51\Assignment-6\tests\end-to-end tests
C:\Users\Sungpoon\Documents\GitHub\Cisc327-GROUP51\Assignment-6\tests\end-to-end tests>
```

4. Now run the command “**python test_e2e_payment.py**”

If everything is done properly, this should be the result

```
C:\Users\Sungmoon\Documents\GitHub\Cisc327-GROUP51\Assignment-6\tests\end-to-end tests>python test_e2e_payment.py

DevTools listening on ws://127.0.0.1:52781/devtools/browser/c45335be-35cc-4890-86f9-9354b2e8945d
This should be valid payment with sufficient fund
Flash message: Payment of CAD $1244.13 successful! Remaining balance: CAD $55.86999999999999.

DevTools listening on ws://127.0.0.1:52803/devtools/browser/eabb8230-d107-41fb-bb7c-f60bd7e35985
This should be invalid payment
Flash message: Invalid payment details! Please try again.

DevTools listening on ws://127.0.0.1:52826/devtools/browser/f1bfc29d-8d78-4cf7-858d-0134b1ee5b9c
This should be valid payment but insufficient fund
Flash message: Insufficient funds! Available balance: CAD $900.0.
```

Task distribution:

Sungmoon Choi:

- Listed necessary features to test for end-to-end test
- Developed test_e2e_payment.py
- Recorded the steps and screenshots of the result

Steven Guan:

- Created Selenium testing for the registration functionality
- Added explanation to their test case in the report document
- Contributed to the instructions to run the respective testing file.

Hassan Hamid:

- Created Selenium testing for the login functionality
- Added explanation to this test case in the report document
- Contributed to the instructions to running the respective testing file