comp0034_flask_testing

COMP0034 Code to accompany the lecture covering Flask testing

Exercise 1: Unittest tests

1. Open `test/backend_tests.py`
2. Run the unittest tests and check all run
3. Add a new test in `class TestMain(BaseTestCase):` :
    - GIVEN a Flask application
    - WHEN the '/view_profile' page is requested (GET) when the user is not logged in
    - THEN the user is redirected to the login page and the message 'You must be logged in to view that page.' is displayed

Exercise 2: pytest tests

1. Open the `test/pytest_tests` folder
2. Run the pytest tests and check all run
3. Add a new test in `test_main.py` :
    - GIVEN a Flask application
    - WHEN the '/view_profile' page is requested (GET) when the user is not logged in
    - THEN the user is redirected to the login page and the message 'You must be logged in to view that page.' is displayed

Exercise 3: Selenium webdriver tests

1. Stop any running Flask app before starting the tests (as the Selenium tests run on the same port unless you change the config)
2. Selenium should be installed, if not in the terminal for the venv of the project: `pip install selenium`
3. Download the browser driver https://sites.google.com/a/chromium.org/chromedriver/ (you MUST choose the version that matches your version of Chrome) and save it to the `test` directory of this project
4. Open `browser_tests.py`
5. Run the tests (3 tests, should all pass)
6. Add a new test to test the login is successful for an existing user (try: email="cs1234567@ucl.ac.uk", password="cs1234567")

Exercise 4: Selenium IDE tests

1. Install the Chrome or Firefox extension using the instructions on the Selenium site
2. Click on the Selenium IDE plugin icon in Chrome and then open the `comp0034_testing.side` file from the `test` directory
3. Run the Flask app (e.g. in PyCharm)
4. Run the `signup_test`
5. Try adding a new test to test the login process is successful (use the person signed up in the signup test)