College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
IN608: Intermediate Application Development Concepts
Level 6, Credits 15

Practical 11 Django 5: Automation Testing

Due Date: 07/09/2020 at 5pm

In this practical, you will complete a series of tasks covering today's lecture. This practical is worth 1% of the final mark for the IN608: Intermediate Application Development Concepts course.

Before you start, in your practicals repository, create a new branch called **11-practical**.

Task 1

Copy & paste the `dog` Django project from **10-practical**. Run the virtual environment by running the command `pipenv shell`. In `tests.py`, create three `TestCase` classes called `TestDog`, `TestIndexView` & `TestResults`.

`TestDog` will contain the following tests:

- `Dog` object's `breed` equals some value
- `Dog` object's `height` equals some value
- `Dog` object's `adaptability` equals `High`
- Field's `max_length` equals some value

`TestIndexView` will contain the following tests:

- Status code equals 200
- Template used is `practical10cdns/index.html`

`TestResults` will contain the following tests:

- Status code equals 200
- Template used is `practical10cdns/results.html`
- QuerySet returns no `Dog` objects

- QuerySet returns one or more `Dog` objects

Create a `StaticLiveServerTestCase` called `TestResultSelenium`. Install selenium by running the command `pipenv install selenium`. Check the `Pipfile` & `Pipfile.lock` to ensure it has install successfully. `TestResultSelenium` will contain the following tests:

- The text **1 dog matched your search.** is found in the page source
- The text **2 dogs matched your search.** is found in the page source
- The text **No dogs available.** is found in the page source

Run the command `python manage.py tests` to run the tests.

Task 2

Use Coverage.py to measure the code coverage of the project. Coverage.py analyses a program to identify code that was executed & code that was not executed, but could be executed. Coverage measurement is an excellent way to gauge the effectiveness of tests. To install `Coverage.py`, run the command `pipenv install coverage`. Generate a coverage report of the Django project in a nice HTML format.

Resources

- Selenium with Python
- Coverage.py