**Test Documentation:**

Unit Testing is a level of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit Testing is the first level of software testing. These tests are independent of each other.

This is achieved through white box testing which is essentially testing based on the analysis of the internal structure of the system.

Unit Testing Benefits:

* Codes are reusable.
* Development is faster in the long run.
* Code maintenance is easier.
* Code becomes more reliable.
* Debugging becomes easier.

In our project we used unit tests to ensure that the individual components in our code work accordingly for various possible inputs. We don’t create tests for everything, we instead focus on testing the part of our code that impact the behavior of our system.

These test cases are split into classes where a class contains test methods which test many features of one Module, for example we have a class TestUrls which tests all the possible urls which have been defined to be handled.

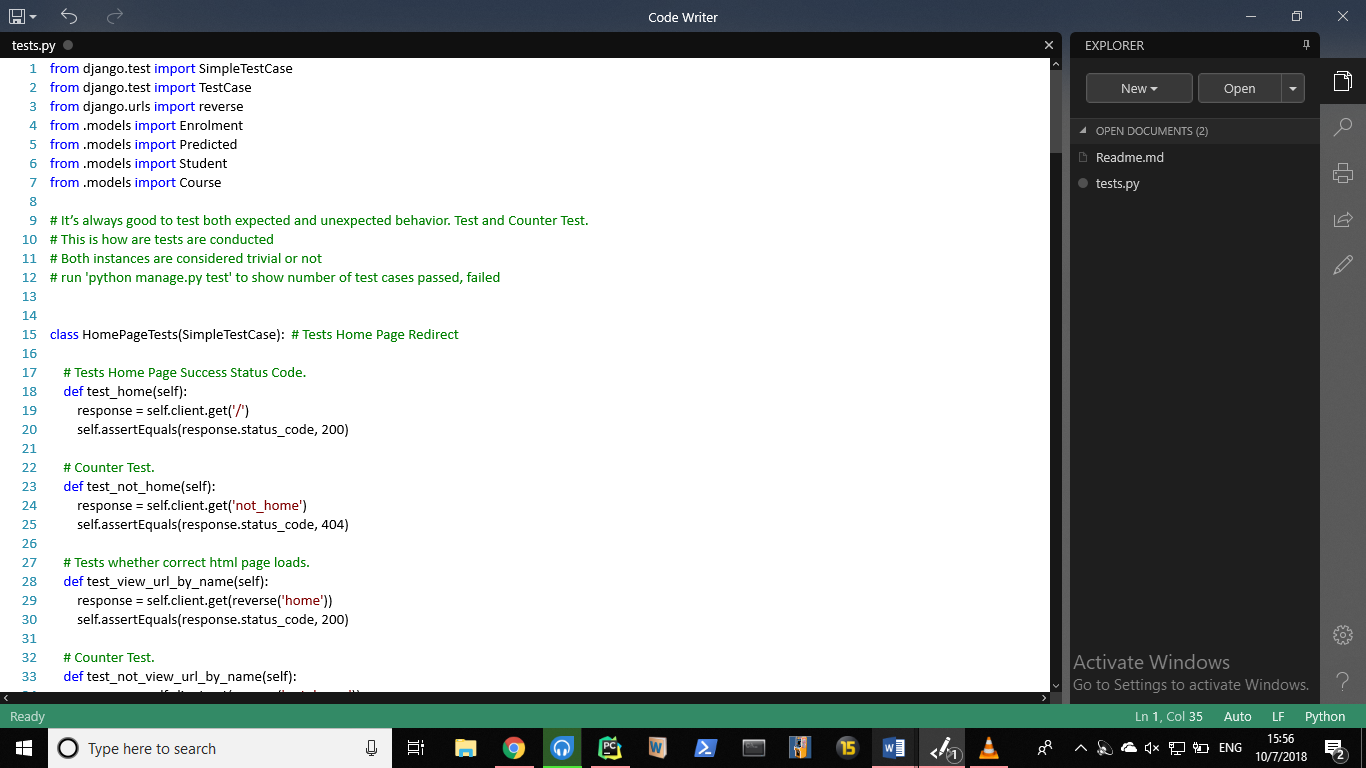
It’s always a good idea to test both expected and unexpected behavior. These are Commented as Test and Counter Test for most individual test cases. It is essential that both instances are considered whether they seem trivial or not. This is how our tests are conducted.

The tests are divided into two file in which one is conducted using the Django built in test modules in the file tests.py in the main app directory and the other is conducted using Pytest modules in the test\_pytest.py file under the tests folder.

**Part One: Using Django Tests modules**

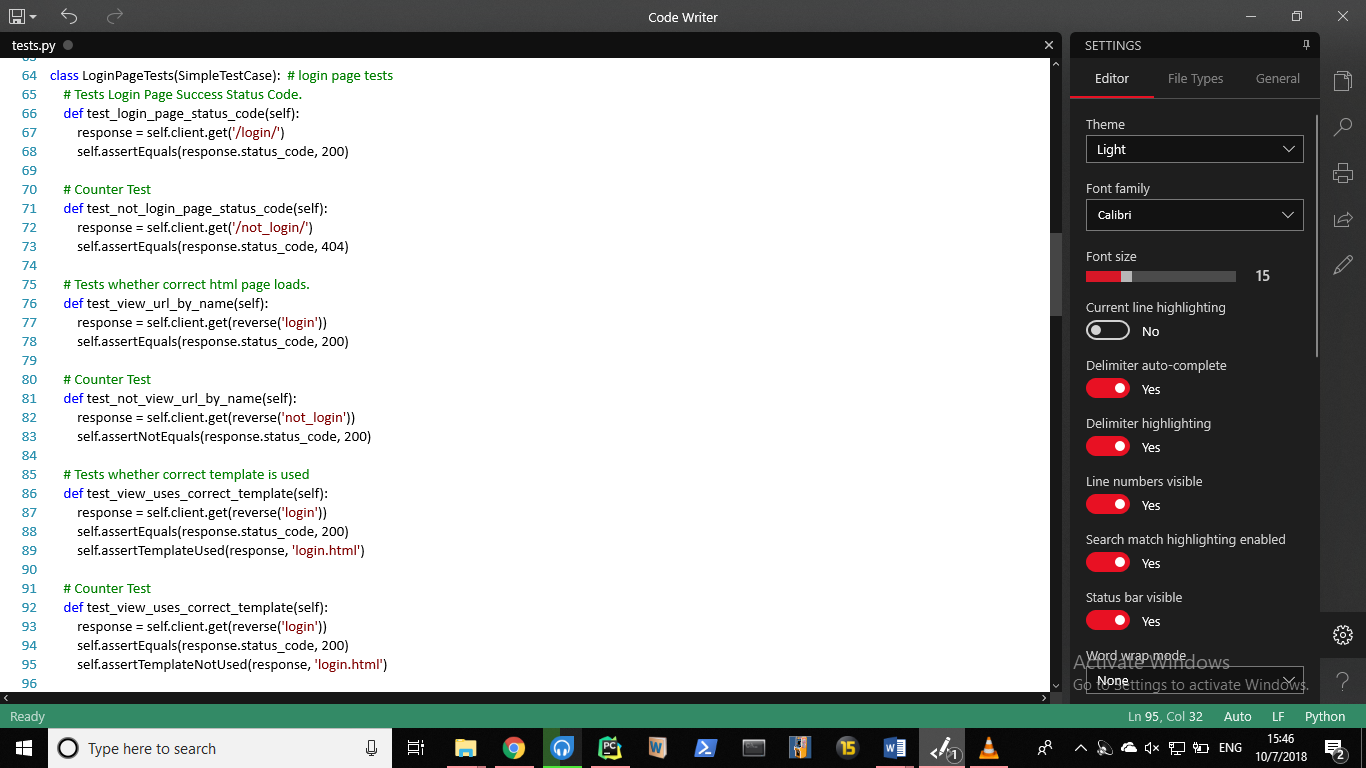
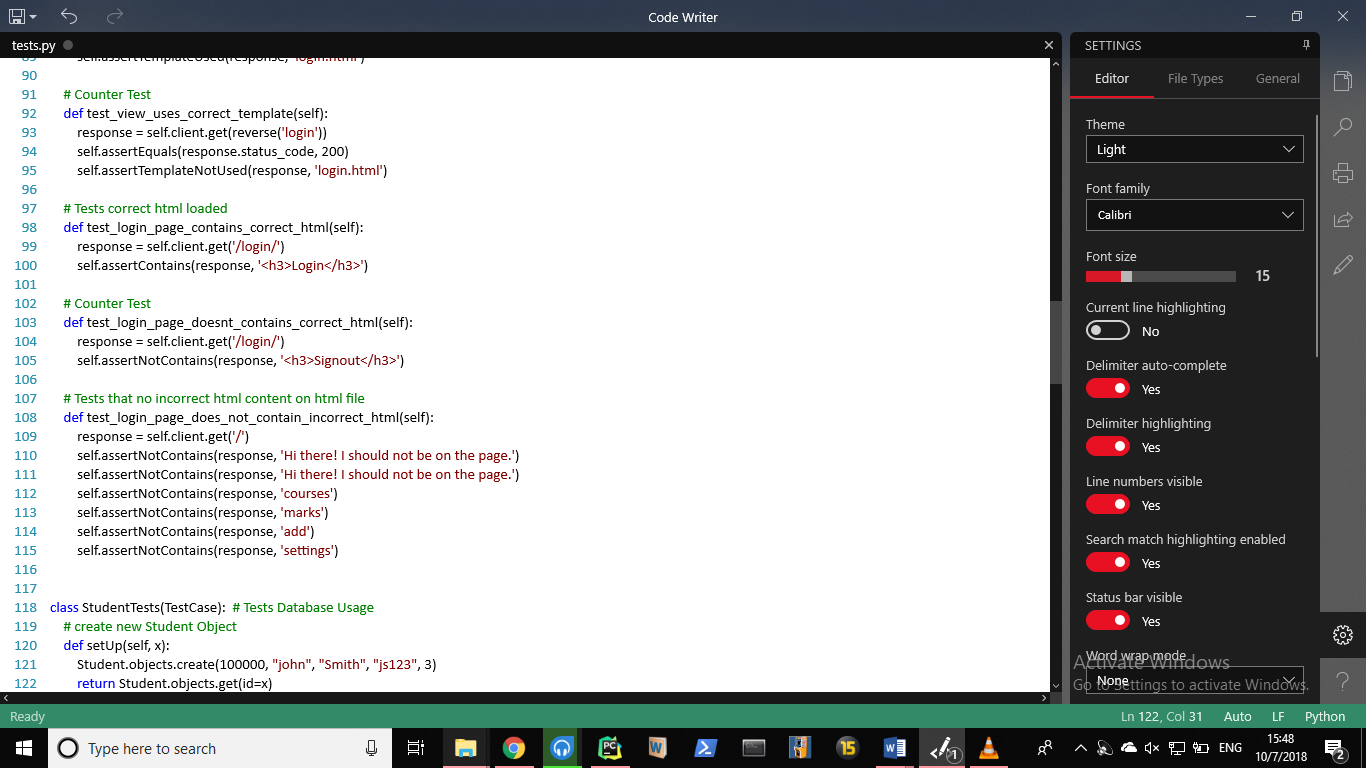
Codes extracted from tests.py.

**Importing Django modules and other important imports**:



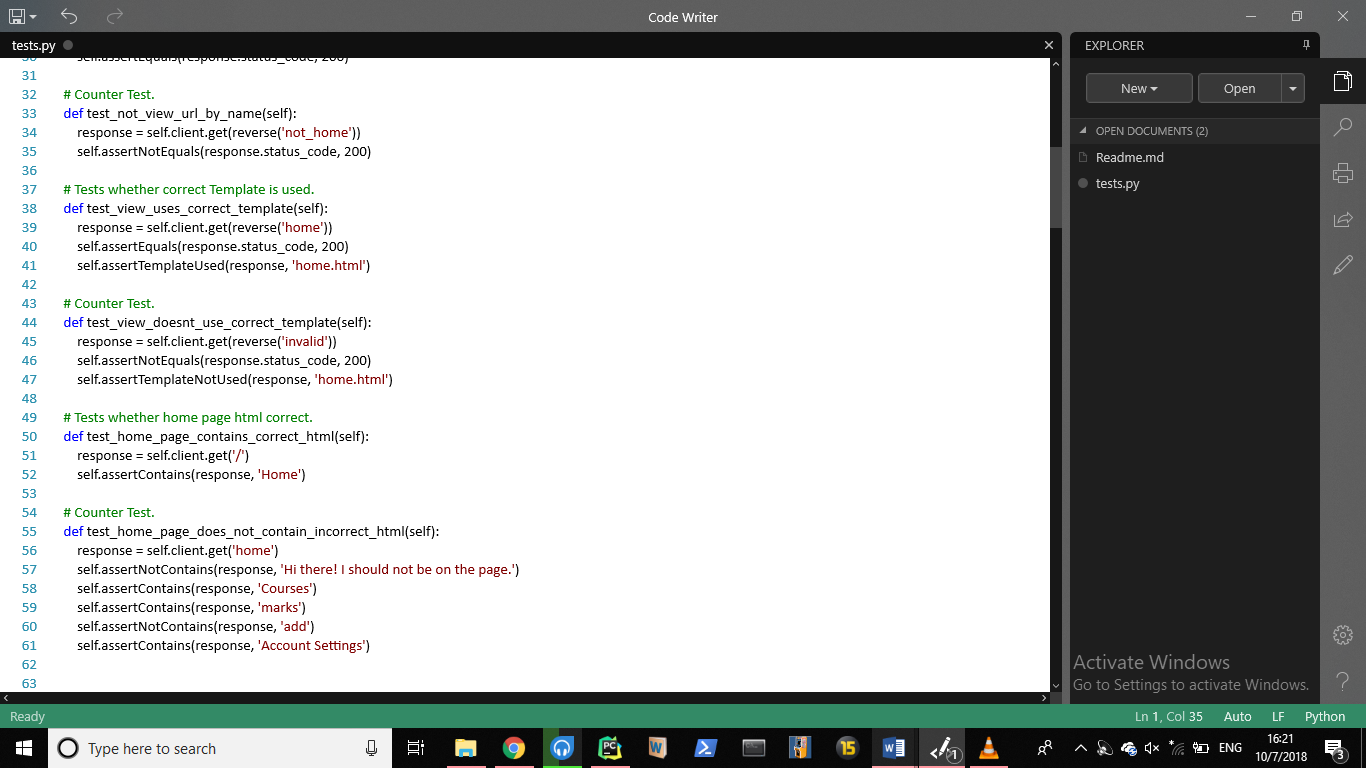
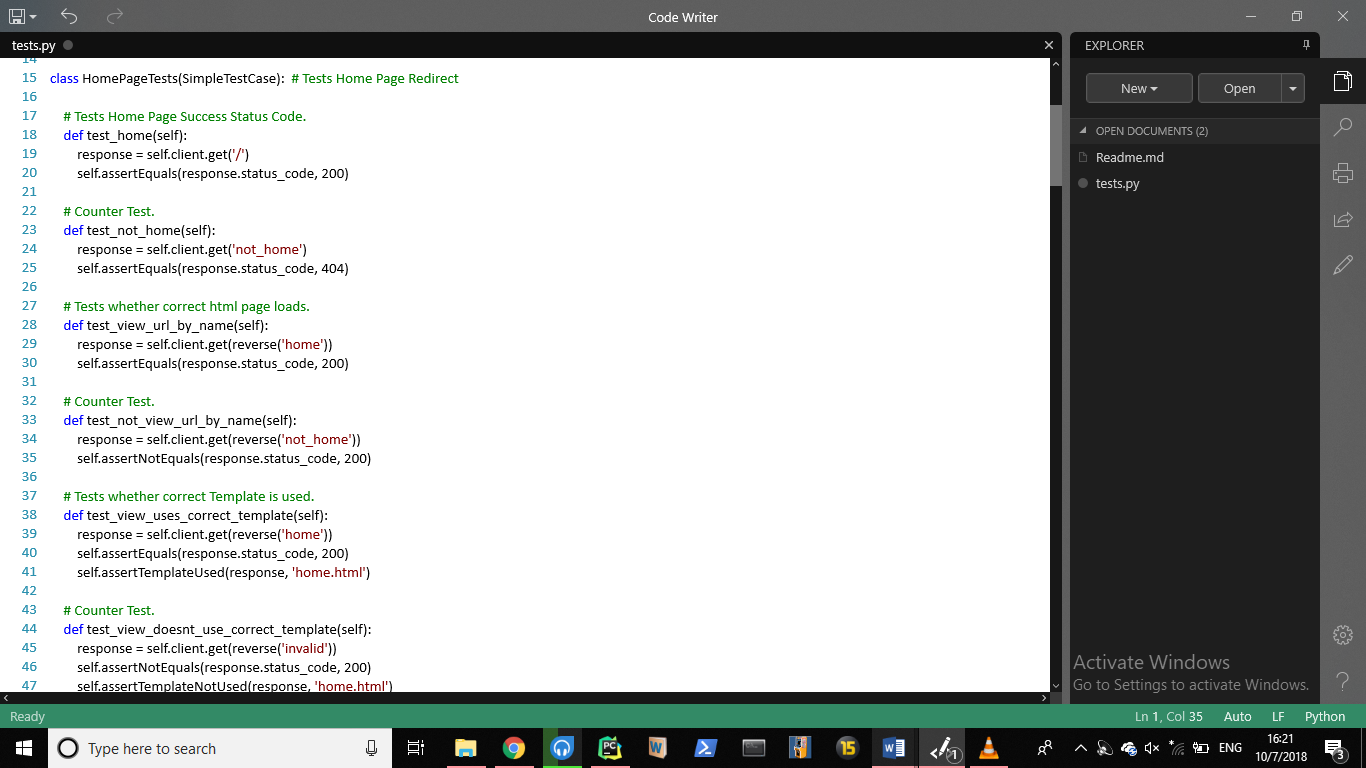
We import the models we are going to test as well. (Line 4-7)

1. **Login Page Tests:**

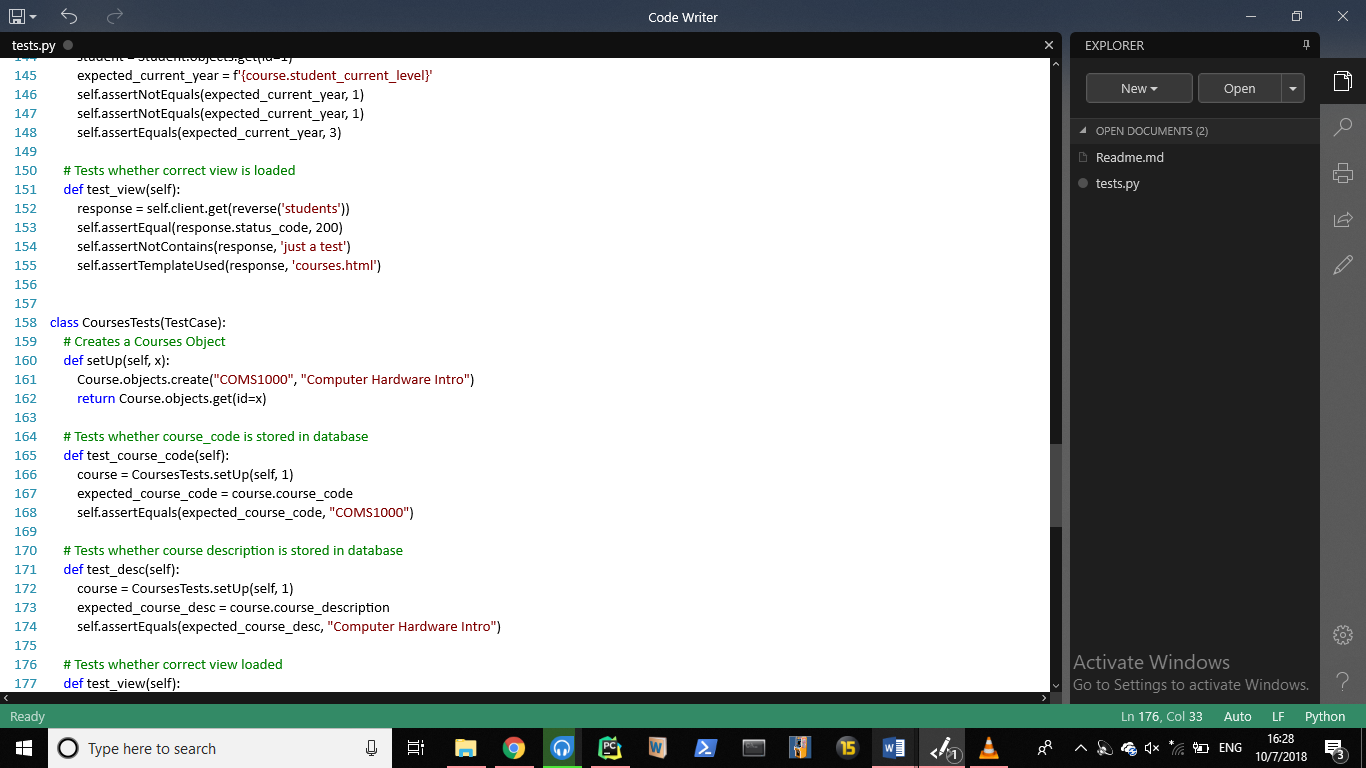
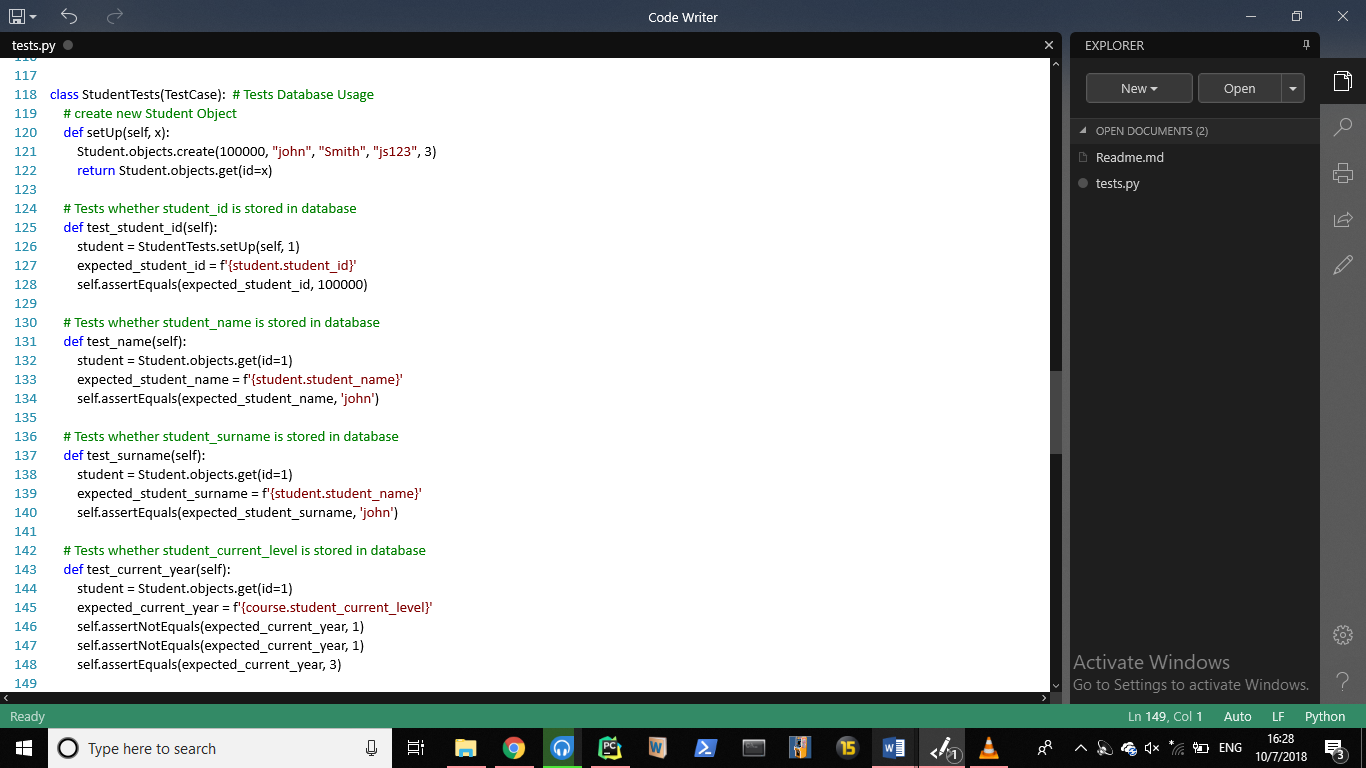
* The login tests are defined in the class ‘LoginPageTests’ - (line 64)
* The first test and counter test ensures that the login page status code is returned whenever the login url is invoked and that an error status code is returned whenever an invalid login request (possibly due to a typing error) is passed.- (line 65-73)
* The second test ensures that the correct html file is returned whenever login url is passed, the counter test ensures that ensures that its not returned anywhere else.-(lines 75-83)
* The third test ensures that the correct template is used by the login url, its counter test ensures that that template is not used elsewhere. - (line 86-95)
* The forth test ensures that the html file contains the word “Login” heading, its counter test ensures that the html doesn’t contain any other random headings. – (line 97-105)
* The fifth test ensures that the html doesn’t contain any strings from other html files. - (line 107 - 115)

1. **Sign Up Page Tests:**

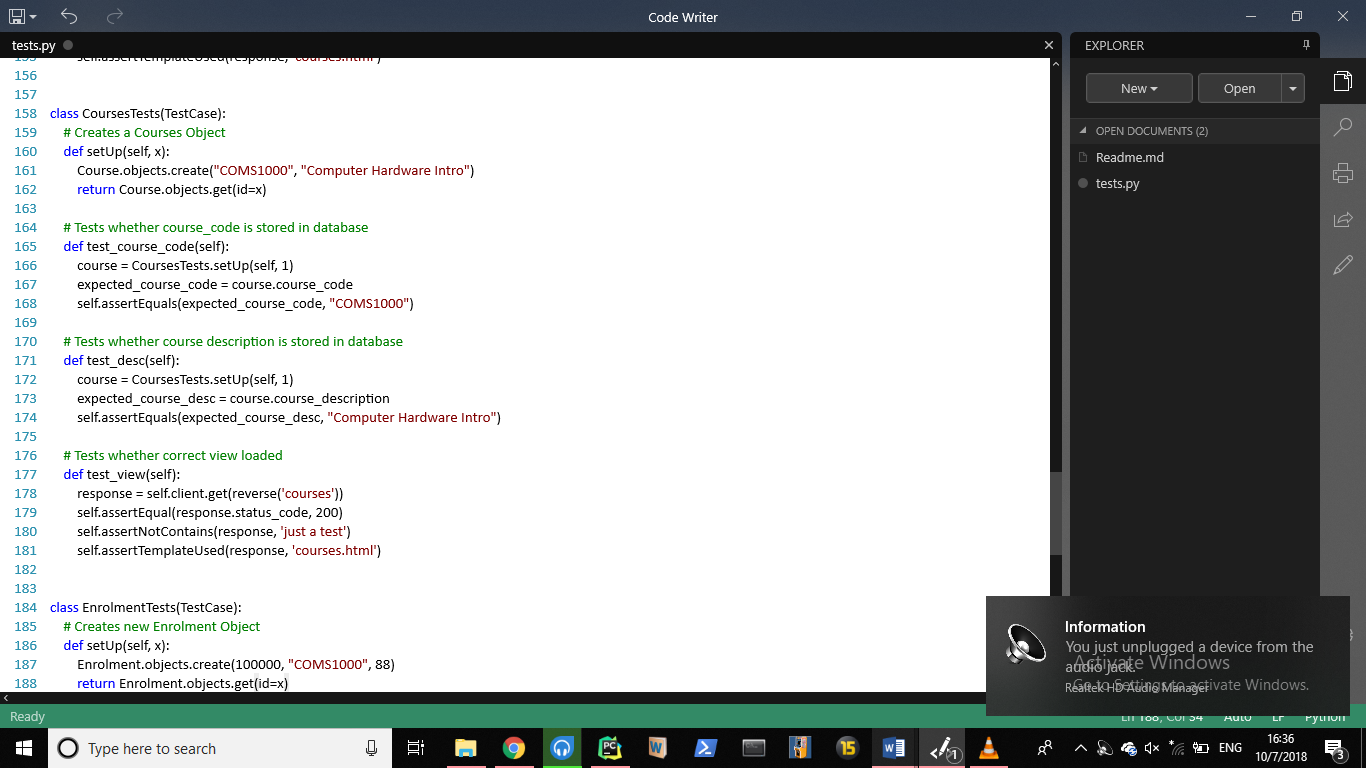
****

* The HomePage tests are defined in the class ‘HomePageTests’ - (line 15)
* The first test and counter test ensures that the home page status code is returned whenever the home url is invoked and that an error status code is returned whenever an invalid home request (possibly due to a typing error) is passed.- (line 17-25)
* The second test ensures that the correct html file is returned whenever home url is passed, the counter test ensures that ensures that its not returned anywhere else.-(lines 27-35)
* The third test ensures that the correct template is used by the home url, its counter test ensures that that template is not used elsewhere. - (line 37-47)
* The fourth test ensure that the correct home url is loaded and returned.

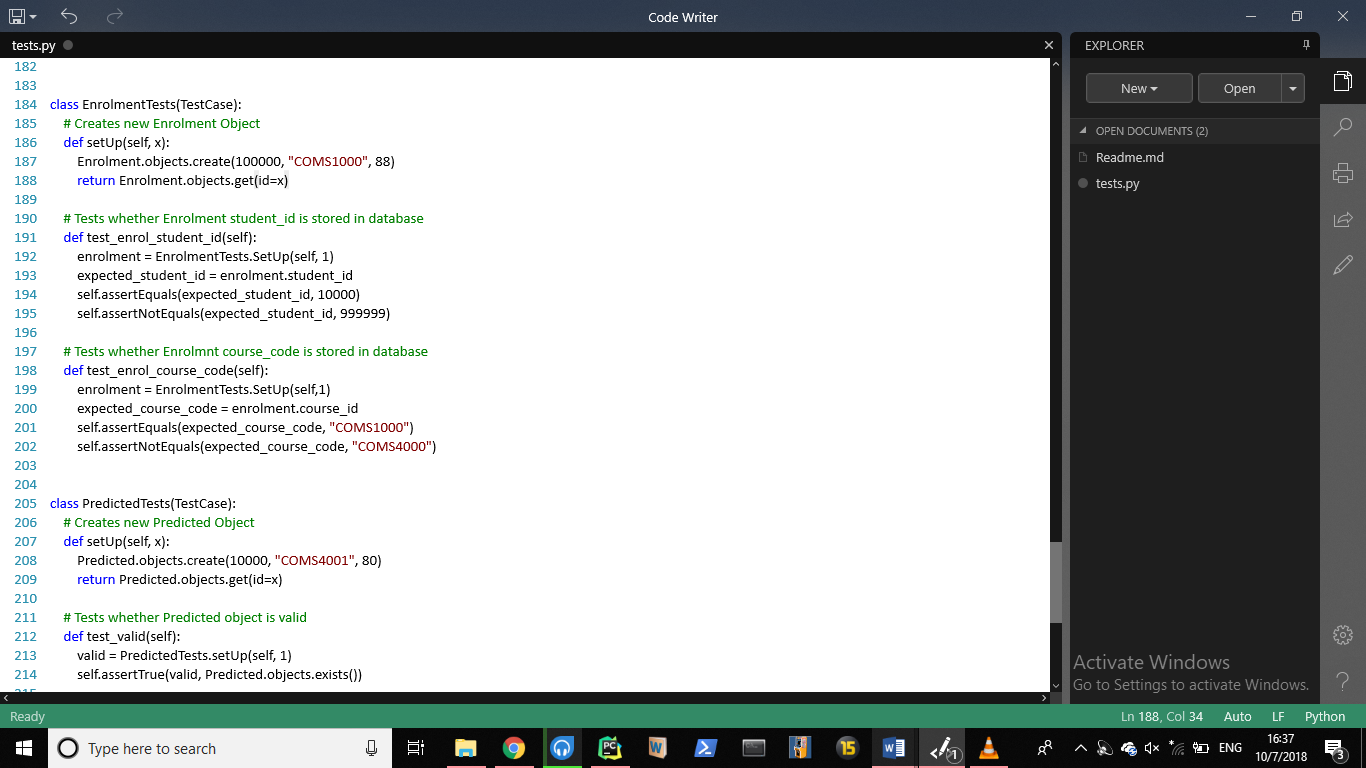
1. **Model Tests:**

****

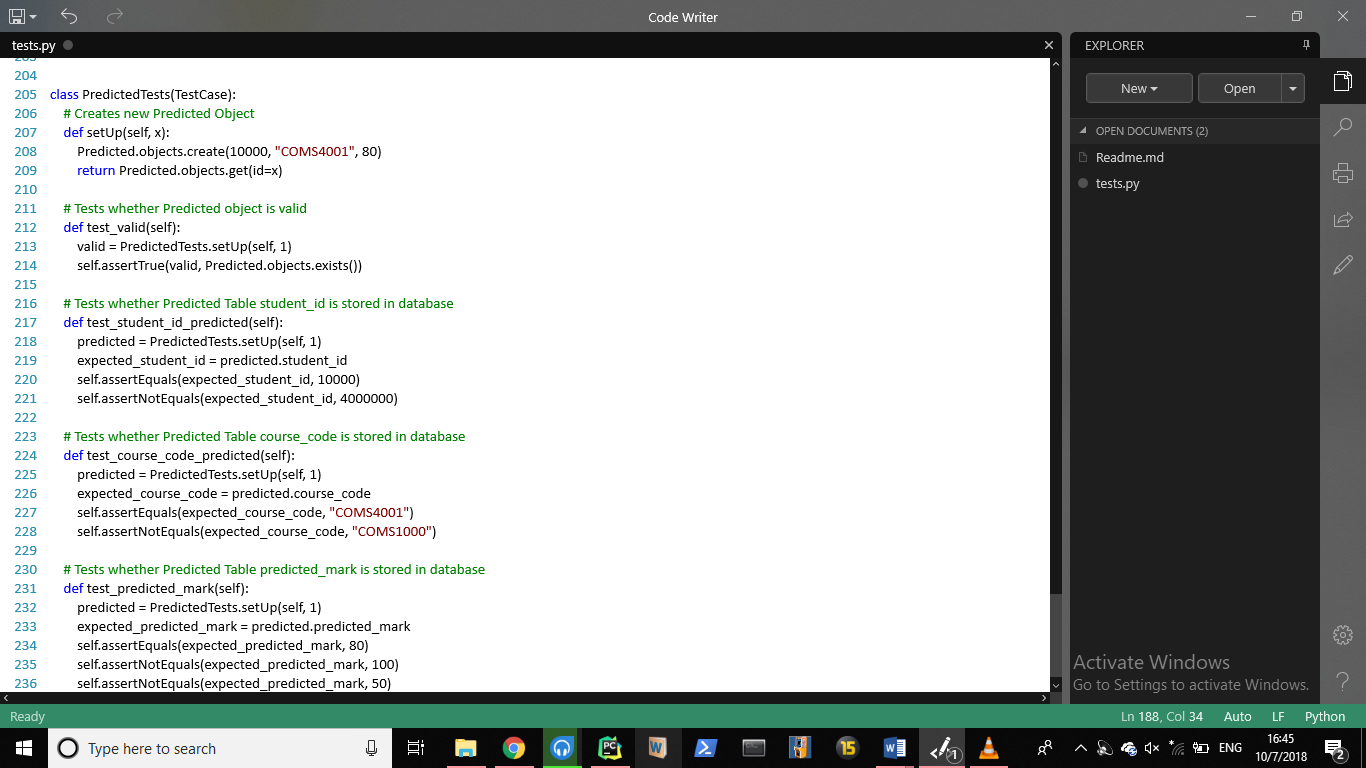
* The Student Model tests are defined in the class ‘StudentTests’ - (line 118)
* The first test ensures that the student\_id is stored in the database. – (line 125-128)
* The second test ensures that the student\_name is stored in the database. – (line 131-134)
* The third test ensures that the student\_surname is stored in the database. – (line 137-140)
* The fourth test ensures that the student\_current\_level is stored in the database. – (line 143-148)
* The fifth test ensures that the correct view is loaded.- (line 151-156)

****

* The Course Model tests are defined in the class ‘CourseTests’ - (line 158)
* The first test ensures that the course\_code is stored in the database. – (line 165-168)
* The second test ensures that the course\_desc is stored in the database. – (line 171-174)
* The third test ensures that the correct view is loaded.- (line 177-181)

****

* The Enrolment Model tests are defined in the class ‘EnrolmentTests’ - (line 184)
* The first test ensures that the student\_id (foreign key) is stored in the database. – (line 191-195)
* The second test ensures that the course\_code is stored in the database. – (line 198-202)

****

* The Predicted Model tests are defined in the class ‘PredictedTests’ - (line 184)
* The first test ensures that the predicted object which was created is valid.- (line212-214)
* The second test ensures that the student\_id (foreign key) is stored in the database. – (line 217-220)
* The third test ensures that the course\_code(foreign key) is stored in the database. – (line 224-228)
* The fourth test ensures that the predicted\_mark is stored in the database. – (line 224-228)