НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем та мереж

**Звіті**

З дисципліни «Екстримальне програмування»

Лабораторної робота №3

**Тестування**

Варіант №21

Виконав:

Студент гр. КН-311

Нанівський О.І.

Прийняв:

Щербак С.С.

Львів – 2020

Тестування відбувається за допомогою додатка unittest.

from django.test import SimpleTestCase, TestCase, Client

from django.urls import resolve, reverse

from .views import calc

# Create your tests here.

class TestUrls(SimpleTestCase):

    def test\_calc\_url\_resolves(self):

        url = reverse('calc')

        self.assertEquals(resolve(url).func, calc)

class TestViews(TestCase):

    def setUp(self):

        self.client = Client()

        self.calc\_url = reverse('calc')

    def test\_calc\_GET(self):

        response = self.client.get(self.calc\_url)

        self.assertEquals(response.status\_code, 200)

        self.assertTemplateUsed(response, 'calc/main.html')

Приклади тестування для додатку pvp. Відповідно тестування urls та views.

from django.test import SimpleTestCase, TestCase, Client

from django.urls import resolve, reverse

from .views import home, game, loser, winner

from .models import Game

# Create your tests here.

class TestUrls(SimpleTestCase):

    def test\_home\_url\_resolves(self):

        url = reverse('home')

        self.assertEquals(resolve(url).func, home)

    def test\_game\_url\_resolves(self):

        url = reverse('game', args=['1', 'nick'])

        self.assertEquals(resolve(url).func, game)

    def test\_loser\_url\_resolves(self):

        url = reverse('loser')

        self.assertEquals(resolve(url).func, loser)

    def test\_winner\_url\_resolves(self):

        url = reverse('winner')

        self.assertEquals(resolve(url).func, winner)

class TestViews(TestCase):

    def setUp(self):

        self.client = Client()

        Game.objects.create()

        self.home\_url = reverse('home')

        self.game\_url = reverse('game', args=['1', 'nick'])

        self.loser\_url = reverse('loser')

        self.winner\_url = reverse('winner')

    def test\_home\_GET(self):

        response = self.client.get(self.home\_url)

        self.assertEquals(response.status\_code, 200)

        self.assertTemplateUsed(response, 'pvp/home.html')

    def test\_game\_GET(self):

        response = self.client.get(self.game\_url)

        self.assertEquals(response.status\_code, 200)

        self.assertTemplateUsed(response, 'pvp/game.html')

    def test\_loser\_GET(self):

        response = self.client.get(self.loser\_url)

        self.assertEquals(response.status\_code, 200)

        self.assertTemplateUsed(response, 'pvp/loser.html')

    def test\_winner\_GET(self):

        response = self.client.get(self.winner\_url)

        self.assertEquals(response.status\_code, 200)

        self.assertTemplateUsed(response, 'pvp/winner.html')

Також можна тестувати за допомогою бібліотеки selenium. Крім неї також потрібно скачати двигун браузера.

Приклад файлу для тестування.

from selenium import webdriver

from pvp.models import Game, Task

from django.contrib.staticfiles.testing import StaticLiveServerTestCase

from django.urls import reverse

import time

class TestPvpHomePage(StaticLiveServerCase):

    def setUp():

        self.browser =  webdriver.Chrome('function\_tests/chtomedriver.exe')