НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем та мереж

**Звіт**

З дисципліни «Екстримальне програмування»

Лабораторної робота №4

**Мультиплеєрна гра ПвпКалькулятор**

Варіант №21

Виконав:

Студент гр. КН-311

Нанівський О.І.

Прийняв:

Щербак С.С.

Львів – 2020

Програма пвп калькулятор.

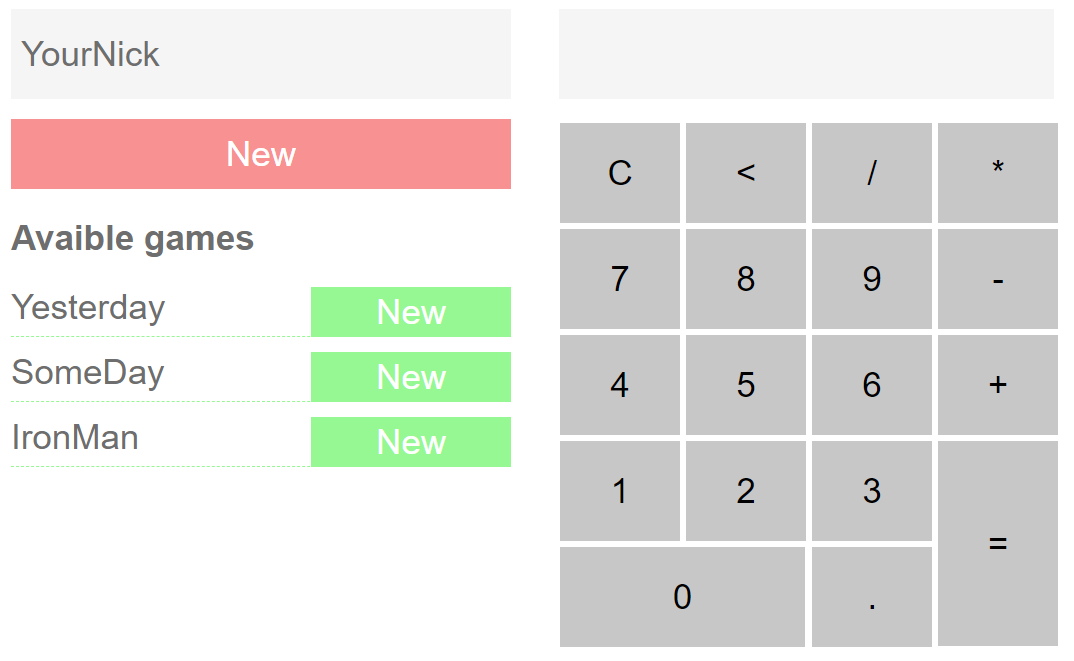
План: проект написаний на django, мультиплеєр відбувається за допомогою бібліотеки channels, написані відповідні тести за допомогою selenium та unittest.

Для запуску тестів в консолі:

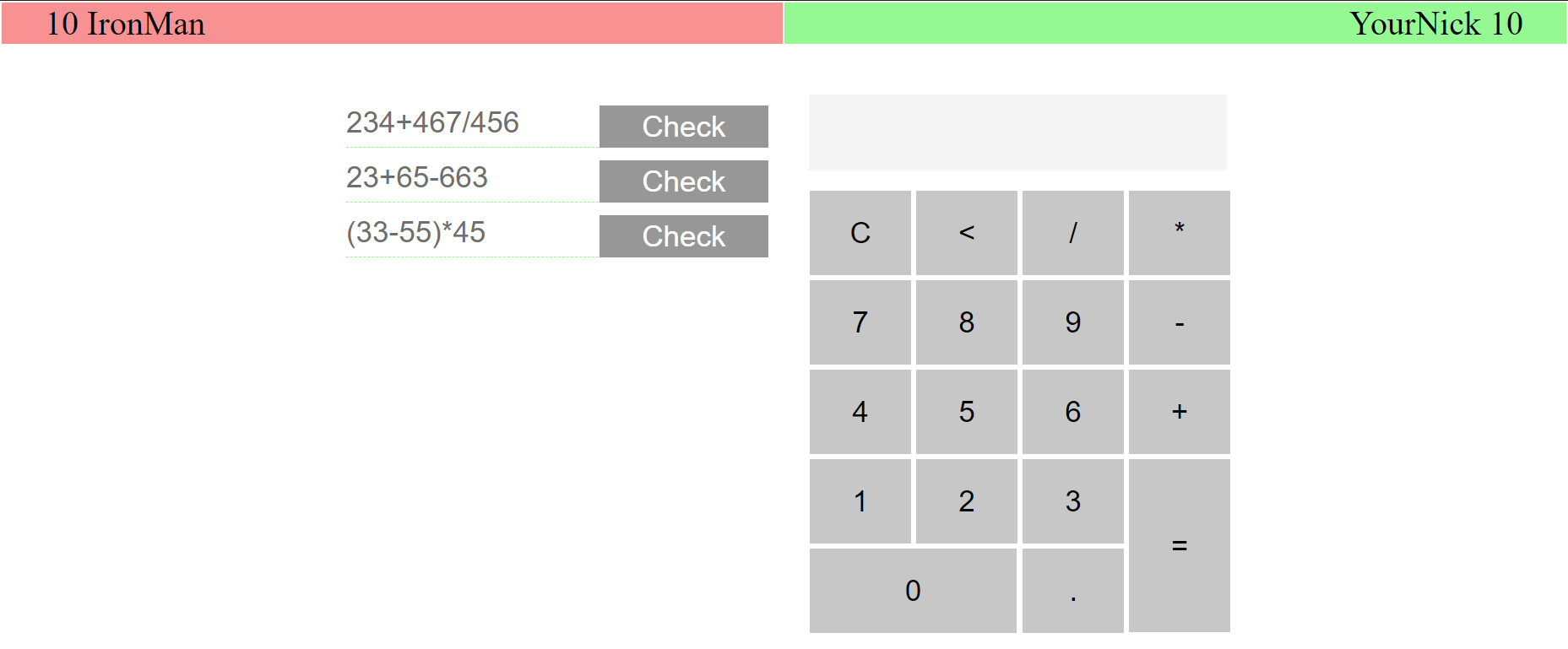
py manage.py test calc

py manage.py test pvp

**Головна сторінка**



**Сторінка гри**

****

**Модель даних**

from django.db import models

# Create your models here.

class Game(models.Model):

    scores\_red\_player = models.CharField(max\_length=2, default="10")

    scores\_green\_player = models.CharField(max\_length=2, default="10")

    nick\_red\_player = models.CharField(max\_length=15, default="Anonim")

    nick\_green\_player = models.CharField(max\_length=15, default="Anonim")

    status = models.CharField(max\_length=20, choices=(("free", "FREE"), ("ingame", "INGAME")), default="free")

class Task(models.Model):

    task = models.CharField(max\_length=25, default="0")

    answer = models.CharField(max\_length=25, default="0")

**Тести**

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')

**Логіка**

var checked = "";

function play\_green(id) {

    nick = document.getElementById("nickview").value;

    document.location.href = "game/" + id + "/" + nick + "/" + "green";

}

function play\_red(id) {

    nick = document.getElementById("nickview").value;

    document.location.href = "game/" + id + "/" + nick + "/" + "red";

}

function check(answer, id) {

    attemp = document.form.textview.value;

    if (checked.includes(id)) {

        return;

    }

    color = document.location.href.split('/')[6];

    if (attemp == answer) {

        checked += id;

        if (color == "green") {

            val = document.getElementById("scores-red").innerHTML;

            document.getElementById("scores-red").innerHTML = val - 1;

        }

        else {

            val = document.getElementById("scores-green").innerHTML;

            document.getElementById("scores-green").innerHTML = val - 1;

        }

        document.getElementById("task-butt-"+id).innerHTML = "Solved";

        document.getElementById("task-butt-"+id).style.background = "#6e6e6e";

    }

    else {

        document.getElementById("task-butt-"+id).innerHTML = "Again";

    }

}

**Відповіді серевера**

from django.shortcuts import render

from .models import Game, Task

# Create your views here.

def home(request):

    games\_list = Game.objects.all()

    max\_id = 1 + Game.objects.values\_list('id', flat=True).order\_by('id').reverse()[0]

    return render(request, 'pvp/home.html', {'games\_list': games\_list, 'max\_id': max\_id})

def game(request, id, nick, color):

    tasks\_list = Task.objects.all()

    if Game.objects.filter(id=id).exists():

        game = Game.objects.get(id=id)

        game.nick\_green\_player = nick

        game.status = "INGAME"

    else:

        game = Game.objects.create()

        game.nick\_red\_player = nick

        game.save()

    return render(request, 'pvp/game.html', {'tasks\_list': tasks\_list, 'game': game})

def loser(request):

    return render(request, 'pvp/loser.html')

def winner(request):

    return render(request, 'pvp/winner.html')