

# Інструкція до використання веб-сервісу

[Запуск сервісу](#)

[Автоматичне заповнення бази даних](#)

[Авторизація](#)

[Інструкція для Вчителя](#)

[Створення групи](#)

[Створення лекції](#)

[Створення задачі](#)

[Перевірка задач](#)

## Запуск сервісу

Спочатку необхідно клонувати проект собі на комп'ютер або завантажити:

```
git clone https://github.com/andriidem308/django_eolimp
```

Для простоти виконання наступних команд, рекомендуємо встановити підтримку "мейкфайлів". Введіть команду в терміналі:

```
sudo apt install make
```

Після цього ми можемо виконувати усі наступні команди з використанням слова make. З правого боку вказано оригінальні команди, якщо у вас не вдасться встановити цей пакет для своєї операційної системи.

Відкриваємо проект, в терміналі переходимо в директорію проекту та встановлюємо всі потрібні бібліотеки:

```
make requirements      pip install -r requirements.txt
```

Робимо міграції бази даних:

```
make migrate_tst      python3 manage.py makemigrations testing
```

```
make migrate_acc      python3 manage.py makemigrations accounts
```

```
make migrate          python3 manage.py migrate
```

Після цього в базі даних проекту створюються необхідні таблиці та проект готовий до запуску. Тому ми запускаємо проект:

*make run*

*python3 manage.py runserver*

В терміналі повинно було з'явитись посилання, за яким слід перейти, натиснувши на нього. Також це можна зробити вручну у адресному рядку в будь-якому браузері, увівши цю адресу: <http://127.0.0.1:8000/>

### **Автоматичне заповнення бази даних**

Для повноцінного функціонування сервісу алгоритм дій такий: зареєструвати вчителя, потім групу, потім зробити задачі та лекції для цієї групи і врешті-решт зареєструвати студента, який, зайшовши на сайт, зможе вирішувати задачі та читати лекції.

Це можна зробити вручну, однак, ми передбачили та реалізували таку можливість як автоматичне заповнення бази даних, яке можна зробити, перейшовши послідовно за цими посиланнями:

[http://127.0.0.1:8000/create\\_teachers](http://127.0.0.1:8000/create_teachers)

[http://127.0.0.1:8000/create\\_groups](http://127.0.0.1:8000/create_groups)

[http://127.0.0.1:8000/create\\_problems](http://127.0.0.1:8000/create_problems)

[http://127.0.0.1:8000/create\\_lectures](http://127.0.0.1:8000/create_lectures)

[http://127.0.0.1:8000/create\\_students](http://127.0.0.1:8000/create_students)

Після цього, у корені нашого проекту з'явиться файл passwords.txt, де буде записана інформація про кожного створеного вчителя та студента з їхніми поштами та паролями для входу.

### **Авторизація**

Перейшовши за <http://127.0.0.1:8000/>, перед собою бачимо домашню сторінку сервіса:

Тут ми можемо увійти в систему, якщо в нас вже є акаунт, натиснувши на

# Testing Service

## Hello! This is web-testing service.

If you already have an account, go to [Authorization page](#).

If you do not have an account, please sign up as [Student](#) or [Teacher](#).



**“Authorization page”**. Якщо акаунту поки немає, можемо зареєструватись у якості Студента або Вчителя, натиснувши на відповідні посилання - **“Student”** або **“Teacher”**.

Для того, щоб зареєструватись у якості Вчителя, користувачу необхідно заповнити всі поля та ввести секретний ключ, записаний у settings.py в полі **SECRET\_KEY\_TEACHER**, у поле **Access code**:

**Sign up as Teacher**

Процедура для реєстрації студента відрізняється лише тим, що йому не треба вводити секретний ключ, проте, треба обрати групу, до якої він належить.

## Інструкція для Вчителя

### Створення групи

Спочатку зверху (на навігаційній панелі) переходимо за посиланням “Groups”. Далі натискаємо кнопку “**New group**”, після чого вводимо назву групи в текстовому полі:

**Add group**

Group name:

## Створення лекції

Спочатку на навігаційній панелі переходимо за посиланням “Lectures”. Далі натискаємо кнопку “New lecture”, після чого обираємо групу, для якої хочемо додати лекцію та заповнюємо поля Назва та Опис. Також при необхідності, можна додати файл, який стосується лекції, натиснувши кнопку **Choose file**:

### New Lecture

Group:	<div>Computer Mathematics-3</div>
Title:	<div>OOP Principles</div>
Description:	<div><p>Object-Oriented Programming (<u>OOP</u>) is a programming paradigm that organizes code into objects, which are instances of classes. It focuses on the concept of objects, their properties, and their interactions to design and develop software systems.</p><p>In <u>OOP</u>, objects represent real-world entities or abstract concepts. They encapsulate both data (attributes or properties) and behaviors (methods or functions). Classes act as blueprints or templates for creating objects, defining their structure and behavior.</p><p>The core principles of <u>OOP</u> include:</p></div>
Attachment:	<div><div>Choose File</div> oop_principles.pdf</div>
<div><div>Save</div><div>Cancel</div></div>	

## Створення задачі

Спочатку на навігаційній панелі переходимо за посиланням “Problems”. Далі натискаємо кнопку “New problem”, після чого обираємо групу, для якої хочемо додати задачу та заповнюємо поля Назва, Опис, Бал за задачу, Максимальний час виконання, Дедлайн:

### Add problem

Group:

Title:

Description:

Problem value:

Max execution time:

Deadline:

Test file:

після чого обираємо \*.json файл з тестами, який має такий формат:

```
1  [
2    {
3      "inputs": ["3","2"],
4      "outputs": ["9"]
5    },
6    {
7      "inputs": ["4", "3"],
8      "outputs": ["64"]
9    },
10   {
11     "inputs": ["2", "4"],
12     "outputs": ["16"]
13   },
14   {
15     "inputs": ["5", "2"],
16     "outputs": ["25"]
17   }
18 ]
```

## Перевірка задач

На панелі зліва у вчителя показуються розв'язки студентів, які цей вчитель ще не перевінив:

Unchecked solutions	
Powered number	Brent Smith
Difference of numbers	Austin Gutierrez
Strings 2	Samuel Gonzalez
Difference of numbers	Samuel Gonzalez
<a href="#">See all problems</a>	

Це є необов'язковим кроком, тому що сервіс автоматично перевіряє задачі та виставляє бали, тому це зроблено для того, якщо вчитель хоче знизити чи підвищити оцінку студента.