# Програма ProblemSet

## Вступ

Дає змогу викладачам створювати задачники, а студентам вирішувати задачі з задачників.

Студенті можуть отримувати умови задач і надсилати вирішення на перевірку в той час коли задачник відкритий.

## Загальні функції

**Вхід**

Зміна пароля

Вихід

## Функції викладача

**Перегляд своїх задачників** problemset/list

**Створення нового задачника** problemset/new

**Редагування існуючого задачника** problemset/edit/{id}

**Видалення задачника** problemset/del/{id}

**Перегляд результатів по задачнику** problemset/show

**Копіювання задачі** problem/copy/{id}

**Редагування задачі** problem/edit/{id}

**Видалення задачі** problem/del/{id}

## Функції студента

**Перегляд активних задач** /to\_solve

**Відкриття задачі** /to\_solve/problem/{prob\_id}/{pset\_title}

**Надсилання рішення** /check (ajax)

## Функції адміна

Підтримка списку юзерів (реєстрація, видалення)

Скидання паролю

## Сутності

### Задача

Id (PK: uuid),

назва,

атрибути,

умова,

мова програмування,

презентація,

повний код із авторським вирішенням

підказка,

автор,

момент створення.

### Користувач

Ім’я користувача (PK: str),

хеш паролю,

роль (student, tutor, admin).

### Задачник

Назва(PK: str),

володар,

id задач у вигляді рядків, поділених пробільними символами

момент активації,

кількість хвилин активності,

фільтр студентів

### Тікет

Id (PK: int),

ім’я юзера,

id задачі,

записи (answer, check, when),

момент втрати чинності

коментар

стан (0 – задача не вирішена, 1 – задача вирішена, )

#### Пояснення

Записи це текстове поле. Один запис зберігає вирішення студента, відповідь tss і момент перевірки. Формат запису: "~0~ answer ~1~ check ~2~ when ~3~"

Тікет створюється, коли студент вперше відкриває сторінку з умовою задачі. При створенні в поле «момент» вноситься момент закриття задачника, і робиться перший запис тікету задля збереження моменту створення тікету.

Кожна перевірка рішення додає запис - (answer, check, when). Якщо check = OK, стан змінюється на 1, така зміна відбувається лише один раз.

# Встановлення

## Підготовка

cd c:\git\dock\psscont

docker build -t pss\_img .

docker save -o pss.tar pss\_img

cd c:\git\dock\problemset

docker build -t pset\_img .

docker save -o pset.tar pset\_img

## Перенесення

Створити папки ~/dock і ~/data.

Скоп в папку ~/dock

pss.tar, problemset.tar

.env

Docker-compose.yml

Скоп в папку ~/data:

pss.db

## Розгортання

docker network create mynet

Перейдіть у dock - директорію з docker-compose.yml.

cd ~\dock

docker load -i pss.tar

docker load -i pset.tar

docker compose up -d

## Корисні команди

docker compose down

docker compose logs

## docker-compose.yml

services:

pss:

image: pss\_img

container\_name: pss\_cont

env\_file: ./.env

networks:

- mynet

ports:

- "7000:7000"

volumes:

- ~/data:/data

restart: unless-stopped

pset:

image: pset\_img

container\_name: pset\_cont

env\_file: ./.env

networks:

- mynet

ports:

- "7001:7001"

volumes:

- ~/data:/data

restart: unless-stopped

networks:

mynet:

external: true