

# Опис та вимоги до курсowego проекту

Курсовий проект

QA Automation

Лектор  
Володимир Троць

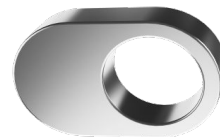
The logo consists of the lowercase letters 'r\_d' in a white, sans-serif font, positioned inside a dark gray rounded rectangle that has a folded top-right corner.

# ОПИС ТА МЕТА ПРОЄКТУ

**Покрити автоматизованими тестами програмне забезпечення для управління проєктами — Kanboard.**

На основі дослідження функціональності UI- або API-документації розробити набір тестів та інфраструктуру їхнього запуску для тестування програми.

Для розробки тестів використовувати Kanban-аплікацію, підняту в Docker.  
Це основний спосіб.

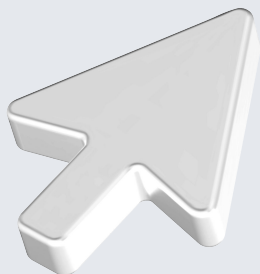


# ФАЗИ КУРСОВОГО ПРОЄКТУ

1. Студент(ка) отримує дані доступу до навчального застосунку та бази даних.
2. Студент(ка) досліджує функціональність і документацію, складає тестовий план.
3. Студент(ка) формує набір тестів для автоматизації (коротко, без покрокового опису), вказує, яким чином проєктуватиме тести (через UI, API, базу даних).
4. Студент(ка) створює тестовий проєкт, імплементує описані тести, описує Jenkins job конфігурацію, перевіряє працездатність і надає на перевірку.
5. Фінальне обговорення.

# ІНСТРУКЦІЯ

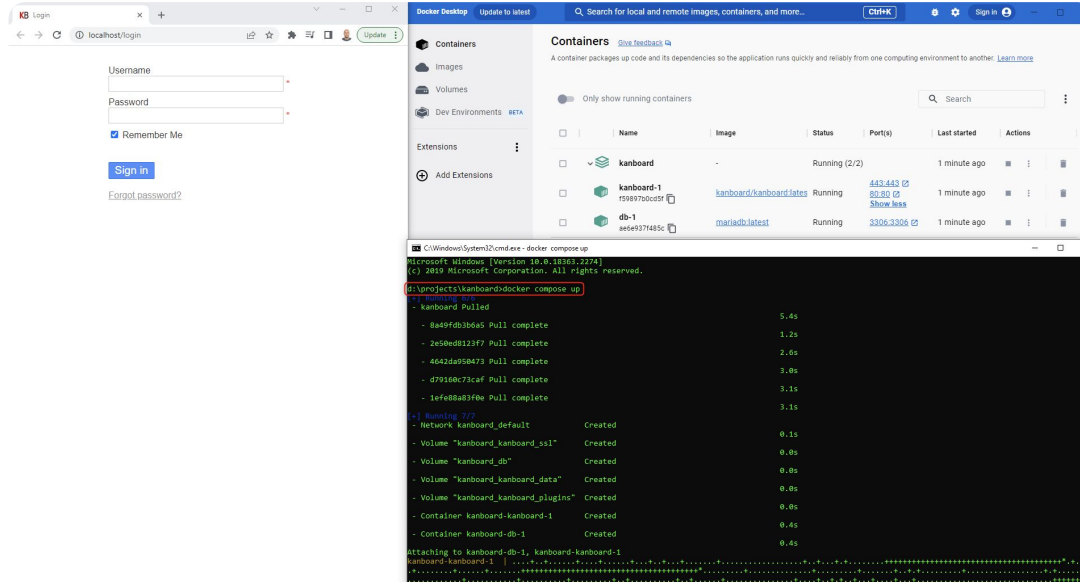
1. Завантажити й зберегти файл **docker-compose.yml** за [посиланням](#).



```
docker-compose.yml
1  version: '2'
2  services:
3    kanboard:
4      image: kanboard/kanboard:latest
5      ports:
6        - "80:80"
7        - "443:443"
8      volumes:
9        - kanboard_data:/var/www/app/data
10       - kanboard_plugins:/var/www/app/plugins
11       - kanboard_ssl:/etc/nginx/ssl
12      environment:
13        DATABASE_URL: mysql://kanboard:kanboard-secret@db/kanboard
14    db:
15      image: mariadb:latest
16      ports:
17        - "3306:3306"
18      command: --default-authentication-plugin=mysql_native_password
19      environment:
20        MYSQL_ROOT_PASSWORD: secret
21        MYSQL_DATABASE: kanboard
22        MYSQL_USER: kanboard
23        MYSQL_PASSWORD: kanboard-secret
24      volumes:
25        - db:/var/lib/mysql
26  volumes:
27    kanboard_data:
28    kanboard_plugins:
29    kanboard_ssl:
30    db:
```

# ІНСТРУКЦІЯ

2. Виконати команду `docker compose up` з папки, де лежить **docker-compose.yml** файл. Docker desktop має бути включеним.



# ВИМОГИ ДО ПРОЄКТУ

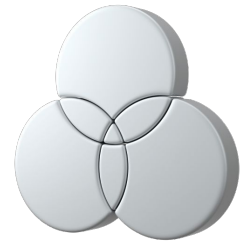
- Проєкт потрібно виконати як **Maven project**
- Треба використати: **testNG, Selenide, Rest Assured, Allure** та (optional) **Cucumber**
- Весь код проєкту має розташовуватися на **git-репозиторії**
- Набір тестів повинен містити **UI-** та **API-тести**
- UI-тести потрібно запускати для **Chrome headless browser, Chrome browser, Firefox browser**
- Запуск тестів варто виконувати на **Jenkins** за заданим графіком (один раз на день)
- За допомогою **конфігураційного файлу** має бути змога дозволяти запускати тести на вибір, які перевіряють аплікацію, підняту локально та доступну з інтернету
- Треба розділити тести за **різними функціональностями** та налаштувати **можливість запускати** окрему з них

# ТЕСТОВІ СЦЕНАРІЇ: UI

1. Вхід: два-три негативних і один позитивний кейс.
2. Користувач може створити проєкт.
3. Користувач може сформулювати завдання. Користувача, проєкт, посилання між користувачем і проєктом створюють за допомогою API як попередні умови та видаляють завдяки API як постумові.
4. Користувач може закрити завдання. Користувача, проєкт, посилання між користувачем і проєктом створюють за допомогою API як попередні умови та видаляють завдяки API як постумові.
5. Користувач може додавати коментарі до завдання. Користувача, проєкт, посилання між користувачем і проєктом створюють за допомогою API як попередні умови та видаляють завдяки API як постумові.

# ТЕСТОВІ СЦЕНАРІЇ: API

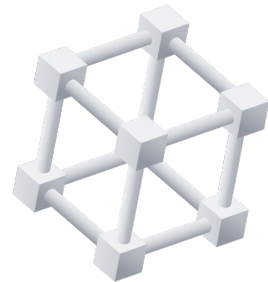
1. Створення та видалення користувача.
2. Створення та видалення завдання.
3. Створення та видалення проєкту.





# ДОДАТКОВІ ТЕСТОВІ СЦЕНАРІЇ (НЕОБОВ'ЯЗКОВІ ДЛЯ ВИКОНАННЯ) : UI

1. Користувач може переносити завдання з однієї фази на іншу.
2. Користувач може додавати коментарі до завдань.
3. Користувач може створювати підзавдання.
4. Користувач може копіювати завдання з іншого проєкту.
5. Користувач може закрити завдання.
6. Користувач може закрити проєкт.



# КРИТЕРІЇ ОЦІНЮВАННЯ

**Максимальна кількість балів за фінальний проєкт — 40.**

- 0–10 балів. Посилання на проєкт є, але головні вимоги не виконано, код не запускається
- 10–20 балів. Головні вимоги є, код запускається, не всі тестові сценарії створено.
- 20–30 балів. Код робочий, пропущено UI- чи API-кроки.
- 30–40 балів. Роботу виконано в повному обсязі, є деякі похибки в архітектурі, організації кроків.





**ЗАВЖДИ Є КУДИ  
ЗРОСТАТИ**