

Індивідуальне завдання.

*deals*

**Мета:** Це комплексне завдання розроблено для перевірки навичок тестування повного життєвого циклу тикетів у системі **STCRM**, враховуючи її архітектуру ізольованих орендарів (multi-tenancy) та рольову модель.

### **Огляд об'єкта тестування: Модуль Tickets**

Тікети в STCRM — це основний інструмент взаємодії між Клієнтом та Співробітником. Кожен тикет має статус, пріоритет, категорію та історію коментарів.

#### **Тестове середовище:**

- Проект: STCRM-система
- URL (Test Server): <https://test2.stcentr.com/>, <https://test3.stcentr.com/>, <https://test4.stcentr.com/>, <https://test5.stcentr.com>
- Доступ: login: [uperadmin@test.com](mailto:uperadmin@test.com), [manager1@test.com](mailto:manager1@test.com), [manager2@test.com](mailto:manager2@test.com), [customer1@test.com](mailto:customer1@test.com), [customer2@test.com](mailto:customer2@test.com), [partner1@test.com](mailto:partner1@test.com), [partner2@test.com](mailto:partner2@test.com), pass: 12345678
- STCRM Документація: <https://stcrm-test-sopyw.ondigitalocean.app/documentation>

# **Завдання.**

- 1. API Тестування (Postman)**
- 2. E2E Testing (Playwright)**
- 3. Документація: Фіксація результатів**

# 1. API Testing (Postman): Інтеграційний рівень та бізнес-логіка.

**Мета:** Перевірка повного життєвого циклу угоди в системі STCRM.

Створіть колекції в спільній папці `STCRM_Deals_Exploratory`.

Підзавдання 1: RBAC та Multi-tenancy.

Перевірити доступ до угод через API:

- Чи може користувач бачити:
  - угоди інших користувачів у межах одного tenant
  - угоди іншого tenant
- Чи можна:
  - редагувати чужі угоди
  - видаляти чужі угоди

# 1. API Testing (Postman): Інтеграційний рівень та бізнес-логіка.

Підзавдання 2: Валідація даних (Create Deal) - протестувати створення угоди з різними даними:

Перевірити:

- пусті поля (title, amount)
- негативні значення amount
- дуже великі числа
- некоректні типи (string замість number)
- відсутній contact / company
- XSS / SQL injection payloads

Підзавдання 3: Бізнес-логіка стадій (Pipeline) - Через PATCH / PUT визначити дозволені переходи між стадіями угоди.

Наприклад:

- Lead → Qualified → Proposal → Won / Lost

Перевірити:

- чи можна перескочити стадії
- чи можна повернутись назад

## 2. E2E Testing (Playwright): Клієнтський шлях та взаємодія.

*Мета: Написати автоматизовані сценарії, що імітують роботу менеджера продажів.*

*Приклади сценаріїв: "Конвертація ліда Зовнішнім користувачем та обробка Власником"*

- **Сценарій 1 (Sales Agent – створення угоди)**
  - Авторизація
  - Створити нову угоду через UI
  - Вибрати клієнта
  - Вказати суму
  - Перевірити:
    - i. угода з'явилась у списку
    - ii. правильна стадія (наприклад: Lead)
- **Сценарій 2 (Pipeline Management)**
- **Сценарій 3 (Access Control)**
- **Сценарій 4 (Закриття угоди)**
- **Сценарій 5 (Validation)**

### 3. Документація: Фіксація результатів.

*Мета: Описати логіку роботи модуля та забезпечити підтримку тестів у майбутньому.*

Необхідні документи:

- **Test Plan (Lead Module):** Короткий опис того, *що* ми тестуємо (критичні шляхи, браузерери, ролі).
- **Checklist/Test Cases** модулів які ми тестимо.
- **API Blueprint (якщо немає Swagger):** Опис полів, заголовків та статус-кодів для модуля Deals.
- **Bug Reports:** Детальні звіти у Trello (доступ пізніше надам) з вкладеними скріншотами з Playwright та JSON-відповідями з Postman.