

Індивідуальне завдання.

“Tickets”

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, manager1@test.com, manager2@test.com, customer1@test.com, customer2@test.com, partner1@test.com, 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).

Створіть колекції в спільній панелі STCRM_Tickets_Exploratory.

Підзавдання 1: Дослідження RBAC та Multi-tenancy

Завдання: Визначте через API, чи може **Зовнішній користувач** (клієнт) бачити тікети, створені іншими клієнтами того ж орендаря, та чи має він доступ до тікетів іншого орендаря (tenant).

Підзавдання 2: Екстремальна валідація даних

Завдання: Дослідіть валідація даних поля при створенні тікета.

Підзавдання 3: Аналіз бізнес-логіки статусів

Завдання: Шляхом запитів **PATCH** або **PUT** визначте "карту переходів" статусів.

2. E2E Testing (Playwright).

Завдання: Напишіть декілька автоматизований сценарій, що імітує реальну взаємодію в системі.

Приклади сценаріїв: "Конфіденційність та Життєвий цикл"

- *Приклади 1 (Customer):* Створити новий тикет через UI, прикріпити файл, перевірити появу тикета в "Мої заявки".
- *Приклади 2 (Agent):* Авторизуватися, знайти цей тикет, додати **Внутрішню нотатку** (Internal Note) та змінити статус на "In Progress".
- *Приклади 3 (Validation):* Повернутися в сесію Клієнта. Перевірити, що статус оновився, але текст внутрішньої нотатки відсутній у DOM-дереві сторінки.

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, браузері, ролі).
- **Checklist/Test Cases** модулів які ми тестимо.
- **API Blueprint (якщо немає Swagger):** Опис полів, заголовків та статус-кодів для модуля *Tickets*.
- **Bug Reports:** Детальні звіти у Trello (доступ пізніше надам) з вкладеними скріншотами з Playwright та JSON-відповідями з Postman.