LESS 5

horizontal line

Опис ситуації: компанія має намір розробити повний аналог rozetka.ua.

# Завдання 1: Ключові області тестування, які не варто автоматизувати на цьому проекті

1. **Тестування UI (інтерфейсу користувача)**: Інтерфейс постійно змінюється, і автоматизація цих тестів може бути невиправдано складною та трудомісткою, особливо на початкових етапах проекту, коли дизайн активно доопрацьовується.
2. **Ad-hoc тестування**: Це неконтрольоване тестування без чіткої структури та сценаріїв. Автоматизація такого типу тестів безглузда, оскільки її суть полягає у вільному дослідженні програми.
3. **Тестування на юзабіліті**: Автоматизація не може оцінити зручність використання сайту з точки зору користувача. Це завжди залишатиметься завданням для людей.
4. **Креативне тестування**: Пошук нестандартних або нових сценаріїв використання системи, де інтуїція тестувальника відіграє велику роль. Така робота не піддається автоматизації.
5. **Тестування локалізації та контенту**: Автоматизовані системи не можуть ефективно перевіряти правильність перекладу, змісту текстів та адаптацію до місцевих культурних особливостей.

# Завдання 2: Ключові області тестування, які мають сенс піддати автоматизації

1. **Регресійне тестування**: Це один із найкращих кандидатів для автоматизації, оскільки з кожним новим релізом повторюється велика кількість тестів для перевірки, що нові зміни не порушили вже існуючу функціональність.
2. **Модульне тестування**: Автоматизація тестів на рівні окремих модулів дозволить перевіряти правильність їх роботи ізольовано, що важливо при розробці великих систем.
3. **Тестування продуктивності (Performance testing)**: Вимірювання часу завантаження сторінок, обробки запитів, і витривалості системи під навантаженням — це задачі, які найкраще автоматизувати для постійного контролю.
4. **Тестування API**: Оскільки API зазвичай не має інтерфейсу користувача і його функціональність рідше змінюється, такі тести легко автоматизуються і дають високий рівень надійності.
5. **Тестування сумісності**: Якщо потрібно перевіряти, як система працює на різних платформах і браузерах, автоматизація тут значно прискорить процес.
6. **Конфігураційне тестування**: Автоматизація налаштування та перевірки різних конфігурацій системи дозволить швидко перевіряти її роботу в різних умовах.
7. **Налаштування тестового оточення**: Створення та підготовка оточення для тестування є рутинним завданням, яке можна автоматизувати для економії часу.
8. **Інтеграційне тестування**: Тестування взаємодії між різними модулями системи також можна автоматизувати, щоб швидше виявляти проблеми в інтеграціях.
9. **Комбінаторні техніки тестування**: Для складних сценаріїв з великою кількістю можливих варіантів введення автоматизовані тестові сценарії можуть охопити більше випадків, ніж ручне тестування.