**Перший рівень — відпрацюй навички на базовому рівні.**

**1. Склади порівняльну таблицю трьох видів тестової документації:**

| Назва тестової документації | Основні характеристики | Переваги | Недоліки |
| --- | --- | --- | --- |
| Чек-ліст | Список перевірок необхідних для тестування системи, котрі бувають залежні від послідовності та незалежні. Містить опис та перевіряє функціональність усього додатка. | 1. Значно прискорює підготовку для тестування. 2. Структурування інформації. 3. Структурованість може вказати на ієрархію ідей з тестування. 4. Гарно підходить до проектів, для яких є менше часу на тестування. 5. Легко відстежувати знаходження помилок та послідовності роботи системи. 6. Валідація для кожного пункту. | 1. Немає детального опису перевірки функціоналу. 2. Не вказується параметрів та кроків проходження щодо перевірки певних функцій. 3. Ми не вказуємо очікувані результати, а вихідні результати описуються як “pass/fail” без певної конкретизації. |
| Тест-кейс | Строгий документ із чітко поставленими кроками виповнення тестування певного функціоналу додатка з чітко окресленими критеріями структуризації. | 1. Детально описує крокування за перевіркою функціональності чи його частини.. 2. Кожний крок показує послідовність та залежність від попереднього. 3. Для кожного функціоналу використовуються окремі тест-кейси 4. Кожний тест-кейс дає тестувальнику конкретну інформацію щодо працездатності та належного функціонування. 5. Легше відстежити шлях до певної помилки завдяки чіткості формування тест-кейсу. | 1. Займає більше часу на конкретизацію усіх кроків тестування. 2. Може бути можливість дублювання тест-кейсів, чого не потрібно допускати. 3. Парадокс пестициду. У всякому випадку постійно потрібно використовувати різні тест-кейси з перевірки функціоналу. 4. На кожний окремий функціонал потрібно розробляти окремий тест-кейс. |
| Користувальницький сценарій | Опис поведінки системи на зовнішні запити, відповіді на питання: який юзер і як може користуватися функціоналом | 1. Дає розуміння працездатності функціоналу зі сторони користувача. 2. Гарно підходить до документування вимог, зв’язаних із системою | 1. Шаблони не гарантують якість. Все залежить від навичок творця. 2. Може бути неповне розуміння різновидів використання. 3. Складно у визначенні рівня опису користувальницького інтерфейсу. 4. Виключення потенційно цінних методик аналізу вимог. 5. Деякі розробники вважають це формальними та складними документами. |

2. Склади чек-ліст для перевірки головної сторінки свого улюбленого інтернет-магазину.

Чекліст зробив у гугл таблиці за цим посиланням:

[Чек-ліст shafa.ua .xlsx](https://docs.google.com/spreadsheets/d/1eOQRTXabgxLgDYj0ZW9PLzEfQEl-ohOR/edit#gid=182204203)

**Другий рівень**

Склади 5 позитивних і 5 негативних тест-кейсів для перевірки основної функціональності твого улюбленого інтернет-магазину

TestCase зробив у гугл таблиці за цим посиланням:  
[testcase shafa](https://docs.google.com/spreadsheets/d/1raeg45BP4f2hc-LG2EdWBN-Vp3KCUFBodW5eLhm_bes/edit#gid=0)