**1) Beet Seed** — відпрацюй навички на базовому рівні.

1. Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.   
Порівняння має містити такі блоки:

* що перевіряється;
* коли застосовується;
* обмеження;
* особливості

| **№** | **Категорії порівняння** | **Функціональне тестування** | **Нефункціональне тестування** | **Тестування, що повʼязане зі змінами** |
| --- | --- | --- | --- | --- |
| 1 | що перевіряється | чи відповідає продукт вимогам | якість роботи системи/коду | що проблема роботи функцій/системи/продукту/ була вирішена після внесення змін. |
| 2 | коли застосовується | - коли є частина готового коду чи продукту. частина, яку ми вже можемо протестувати  - після інтеграції частин продукту  -після внесення змін  - перед релізом продукту | - після функціонального тестування  - коли треба перевірити швидкість, сумісність, навантаження та інші функції НФ тестування, які покращують продуктивність продукту | - після збірки коду  - після внесення будь-яких змін у код  - після випуску оновленої версії додатку |
| 3 | обмеження | тестує лише відповідність вимогам. навантаження, безпеку, зручність, словом, внутрішні помилки - не може протестувати | не перевіряє правильність функцій та їх логіку, а перевіряє радше їх характеристику | - не можуть перевірити усі помилки, бо працюють лише з конкретними виправленнями |
| 4 | особливості | - проводиться одним з перших та є базовим  - спочатку треба впевнюємось, що у позитивному тестуванні все ок  - тестуємо згідно вимог  - дивимось як звичайний користувач, відтворюємо його кроки  - більш рутинний процес | без цього виду тестування продукт може бути створений згідно вимог, але бути нестабільним або навіть небезпечним  - деякі види цього тестування вимагають автоматизації | - повторюється багато разів через часті оновлення  - залежать від якості написаних тестів (?) |

2. Поясни, в чому різниця між регресією та ретестингом (5 речень).

Ретестинг, як і витікає з назви, це повторне тестування конкретної помилки, тестування для підтвердження її виправлення. регресійне тестування - це тестування певної кількості функцій саме після внесення змін. регресійне тестування - це більш широке поняття.

ретестинг має вищий пріоритет, бо виправляється дійсний баг. а регресія - це більш широке поняття, яке охоплює інші частини продукту, навіть ті, які працюють коректно, щоб впевнитись, що зміни не вплинули на правильність роботи продукту.

**2) Beet Sprout** — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Як ти вважаєш, чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог?

* Якщо так – в яких випадках?
* Якщо ні – чому?
* Обґрунтуй свою відповідь.

Щоб отримати якісний продукт потрібно провести повний цикл тестування, тому нефункціональне тестування буде невідʼємною частиною функціонального тестування. якщо я випустила машину, виконала функціональне тестування і перевірила, що вона їде, але НЕ протестувала при якій силі удару повинна спрацювати подушка безпеки, що є вже нефункціональним видом, наслідки можуть бути критичними.

Таким чином, проведення тільки функціонально тестування цілком можливо. продукт буде “якось” працювати. але питання наскільки зручно буде користуватись таким продуктом? чи безпечно? та чи є сенс випускати такий продукт? на мою думку ні. без нефункціонального тестування не можна вважати продукт повністю перевіреним.

3. Як ти розумієш необхідність проведення smoke (димового) тестування? Чи завжди воно є доречним?

Димове тестування потрібне, щоб перевірити, що усі важливі функції працюють справно після кожного оновлення. можна зіставити його з принципом раннього тестування. димове тестування допомагає побачити чи запускається взагалі система та чи можна взагалі працювати з продуктом далі.

по аналогії до прикладу з машиною, перед тим як їхати мені треба перевірити чи є у мене пальне та чи запускається двигун, чи працюють гальма. якщо щось з цього не працює - поїхати я або не зможу, або це знову ж таки буде небезпечно.

**3) Mighty Beet** — різнобічно опануй тематику уроку.

1. Виконай завдання двох попередніх рівнів.

2. Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв.

Користувачі можуть завантажувати фотографії котиків. Але не можуть завантажувати фотографії інших тварин/людей/об’єктів. Користувачі можуть додавати друзів, ставити “вподобайки”, залишати коментарі.  
  
Завдання: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.

посилання на тест-кейси:

<https://docs.google.com/spreadsheets/d/15PJPlq85mgXgiqWfRkE9sFwYAb7wKD7ItXklQ7BpHcI/edit?usp=sharing>

так само як в попередньому дз, кожен тест кейс на окремому листі та виділені для зручності оновлені дані

3. Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.   
Опиши перевірки, які б їх перевіряли (3-5 прикладів).

враховуючи, що це стартап та застосунок поки не має високого потоку користувачів:

1) load: застосунок має дозволяти публікувати фотографію 1000 людям одночасно (змоделювати ситуацію, коли 1000 користувачів одночасно завантажують фото і перевірити чи система працює стабільно)

2) надійність: застосунок має працювати без збоїв 24 години на добу з 00:00 до 23:59 щодня (запустити додаток працювати без зупинки на 7-10 днів і перевірити чи він не дає збоїв)

3) безпека: застосунок не має передавати особисті дані користувачів (ПІБ, номер телефону, адресу електронної пошти тощо) третім сторонам без згоди користувачів

4) usability: перевірити, що інтерфейс застосунку інтуїтивно зрозумілий та зручний у використанні (провести бета-тестування)

5) compatibility: перевірити, що застосунок працює без помилок на платформах ios та android (перевірити застосунок на різних моделях телефонів з різними версіями iOS і Android)