# Урок 23

| **Products** | | | | | | |  | **Orders** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Price | Quantity | Allergic | Promotion | Made\_in \_Ukraine |  | Id | ProductId | Amount | Customer | Address | Phone | Total pay |
| 1 | Bread | 12,00 | 25 | TRUE | FALSE | TRUE |  | 1 | 14 | 1 | Tom | Central streest 45 | 380631112233 | 12,00 |
| 2 | Baggett | 20,00 | 20 | TRUE | FALSE | TRUE |  | 2 | 19 | 2 | Anna | Back street 158 | 380985557744 | 150,00 |
| 3 | Croissant | 8,00 | 15 | TRUE | FALSE | TRUE |  | 3 | 3 | 10 | Sam | Central streest 458 | 380501235588 | 80,00 |
| 4 | Butter 200 g | 65,00 | 45 | TRUE | TRUE | TRUE |  | 4 | 7 | 2 | Sara | Park avenue 145 | 380454545879 | 110,00 |
| 5 | Milk 0,5 litr 1% | 25,00 | 10 | TRUE | FALSE | TRUE |  | 5 | 15 | 1 | Oscar | Back street 74 | 380569994712 | 15,00 |
| 6 | Milk 1 litr 1% | 30,00 | 20 | TRUE | FALSE | TRUE |  |  |  |  |  |  |  |  |
| 7 | Milk 2 litrs 1% | 55,00 | 15 | TRUE | TRUE | TRUE |  | **Напишіть запити SQL для відображення даних:** | | | | | | |
| 8 | Milk 0,5 litr 2,5% | 30,00 | 20 | TRUE | FALSE | TRUE |  | 1 | Order Id, Product name, Amount, Price, Total pay | | | | | |
| 9 | Milk 1 litr 2,5% | 35,00 | 30 | TRUE | FALSE | TRUE |  | 2 | Order Id, Customer, Address, Phone, Promotion | | | | | |
| 10 | Milk 2 litrs 2,5% | 65,00 | 20 | TRUE | TRUE | TRUE |  | 3 | Order Id, Customer, Total pay, Allergic де Allergic = 'True' | | | | | |
| 11 | Milk 0,5 litr 3,2% | 35,00 | 10 | TRUE | FALSE | TRUE |  |  |  |  |  |  |  |  |
| 12 | Milk 1 litr 3,2% | 40,00 | 10 | TRUE | FALSE | TRUE |  | **Запити SQL для відображення даних:** | | | | | | |
| 13 | Milk 2 litrs 3,2% | 75,00 | 5 | TRUE | TRUE | TRUE |  | 1 | SELECT Orders.ProductId, Products.ProductName, Orders.Amount, Products.Price, Orders.TotalPay  FROM Orders  INNER JOIN Products ON Orders.ProductId=Products.ProductId; | | | | | |
| 14 | Egg 10 | 55,00 | 10 | TRUE | FALSE | TRUE |  |
| 15 | Green apple 1 kg | 15,00 | 11 | FALSE | FALSE | FALSE |  | 2 | SELECT Orders.ProductId, Orders.Customer, Orders.Address, Orders.Phone, Products.Promotion  FROM Products  LEFT JOIN Orders ON Products.ProductId = Orders.ProductId  ORDER BY Orders.ProductId; | | | | | |
| 16 | Yellow apple 1 kg | 20,00 | 17 | FALSE | FALSE | FALSE |  |
| 17 | Red apple 1 kg | 22,00 | 21 | FALSE | FALSE | FALSE |  |
| 18 | Potatoes 1 kd | 17,00 | 37 | FALSE | FALSE | TRUE |  | 3 | SELECT Orders.ProductId, Orders.Customer, Orders.Total pay, Products.Allergic  FROM Products  INNER JOIN Orders ON Orders.ProductId=Products.ProductId WHERE Allergic= 'True'; | | | | | |
| 19 | Tomat 1 kg | 75,00 | 14 | TRUE | FALSE | FALSE |  |
| 20 | Chicken meat 1 kg | 165,00 | 14 | FALSE | TRUE | TRUE |  |

# Урок 22

| **Products** | | | | | | |  |  | **Напишіть запити SQL для відображення даних з таблиці Products:** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Price | Quantity | Allergic | Promotion | Made\_in \_Ukraine |  | 1 | Знайти мінімальну ціну |
| 1 | Bread | 12,00 | 25 | TRUE | FALSE | TRUE |  | 2 | Знайти максимальну кількість |
| 2 | Baggett | 20,00 | 20 | TRUE | FALSE | TRUE |  | 3 | Порахувати кількість товарів з алергенами |
| 3 | Croissant | 8,00 | 15 | TRUE | FALSE | TRUE |  | 4 | Порахувати кількість товарів з промо |
| 4 | Butter 200 g | 65,00 | 45 | TRUE | TRUE | TRUE |  | 5 | Порахувати суму вартості товарів |
| 5 | Milk 0,5 litr 1% | 25,00 | 10 | TRUE | FALSE | TRUE |  | 6 | Порахувати суму кількості товарів |
| 6 | Milk 1 litr 1% | 30,00 | 20 | TRUE | FALSE | TRUE |  | 7 | Порахувати суму вартості товарів, вироблених в Україні |
| 7 | Milk 2 litrs 1% | 55,00 | 15 | TRUE | TRUE | TRUE |  | 8 | Знайти середню ціну |
| 8 | Milk 0,5 litr 2,5% | 30,00 | 20 | TRUE | FALSE | TRUE |  | 9 | Відобразити лише ті товари, які вимірюються у літрах |
| 9 | Milk 1 litr 2,5% | 35,00 | 30 | TRUE | FALSE | TRUE |  | 10 | Порахувати суму вартості яблук |
| 10 | Milk 2 litrs 2,5% | 65,00 | 20 | TRUE | TRUE | TRUE |  |  |  |
| 11 | Milk 0,5 litr 3,2% | 35,00 | 10 | TRUE | FALSE | TRUE |  | 1 | SELECT MIN(Price)FROM Products; |
| 12 | Milk 1 litr 3,2% | 40,00 | 10 | TRUE | FALSE | TRUE |  | 2 | SELECT MAX(Quantity)FROM Products; |
| 13 | Milk 2 litrs 3,2% | 75,00 | 5 | TRUE | TRUE | TRUE |  | 3 | SELECT COUNT(Allergic)FROM Products WHERE Allergic =’TRUE’; |
| 14 | Egg 10 | 55,00 | 10 | TRUE | FALSE | TRUE |  | 4 | SELECT COUNT(Promotion)FROM Products WHERE Promotion =’TRUE’; |
| 15 | Green apple 1 kg | 15,00 | 11 | FALSE | FALSE | FALSE |  | 5 | SELECT SUM(Price)FROM Products; |
| 16 | Yellow apple 1 kg | 20,00 | 17 | FALSE | FALSE | FALSE |  | 6 | SELECT SUM(Quantity)FROM Products; |
| 17 | Red apple 1 kg | 22,00 | 21 | FALSE | FALSE | FALSE |  | 7 | SELECT SUM(Price)FROM Products WHERE Made\_in \_Ukraine =’TRUE’; |
| 18 | Potatoes 1 kd | 17,00 | 37 | FALSE | FALSE | TRUE |  | 8 | SELECT AVG(Price)FROM Products |
| 19 | Tomat 1 kg | 75,00 | 14 | TRUE | FALSE | FALSE |  | 9 | SELECT \* FROM Products WHERE *Name* LIKE *'%litr%'*; |
| 20 | Chicken meat 1 kg | 165,00 | 14 | FALSE | TRUE | TRUE |  | 10 | SELECT SUM(Price) FROM Products WHERE *Name* LIKE *'%apple%'*; |

# 

# Урок 21

| **Products** | | | | | | |  |  | **Напишіть запити SQL для відображення даних з таблиці Products:** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Price | Quantity | Allergic | Promotion | Made\_in \_Ukraine |  | 1 | Вивести усі дані з таблиці |
| 1 | Bread | 12,00 | 25 | TRUE | FALSE | TRUE |  | 2 | Список продуктів з алергенами (усі дані) |
| 2 | Baggett | 20,00 | 20 | TRUE | FALSE | TRUE |  | 3 | Список продуктів з промо (усі дані) |
| 3 | Croissant | 8,00 | 15 | TRUE | FALSE | TRUE |  | 4 | Список продуктів без алергенів (усі дані) |
| 4 | Butter 200 g | 65,00 | 45 | TRUE | TRUE | TRUE |  | 5 | Список продуктів без промо (усі дані) |
| 5 | Milk 0,5 litr 1% | 25,00 | 10 | TRUE | FALSE | TRUE |  | 6 | Список продуктів вироблених в Україні (усі дані) |
| 6 | Milk 1 litr 1% | 30,00 | 20 | TRUE | FALSE | TRUE |  | 7 | Список продуктів вироблених поза межами України (усі дані) |
| 7 | Milk 2 litrs 1% | 55,00 | 15 | TRUE | TRUE | TRUE |  | 8 | Вивести лише назви усіх продуктів |
| 8 | Milk 0,5 litr 2,5% | 30,00 | 20 | TRUE | FALSE | TRUE |  | 9 | Вивести лише назви усіх продуктів, ціну та кількість |
| 9 | Milk 1 litr 2,5% | 35,00 | 30 | TRUE | FALSE | TRUE |  | 10 | Вивести усі дані з таблиці з сортуванням по ціні від меншої до більшої |
| 10 | Milk 2 litrs 2,5% | 65,00 | 20 | TRUE | TRUE | TRUE |  | 11 | Вивести усі дані з таблиці з сортуванням по ціні від більшої до меншої |
| 11 | Milk 0,5 litr 3,2% | 35,00 | 10 | TRUE | FALSE | TRUE |  | 12 | Вивести усі дані з таблиці з сортуванням по назві за алфавітом |
| 12 | Milk 1 litr 3,2% | 40,00 | 10 | TRUE | FALSE | TRUE |  | 13 | Вивести усі дані з таблиці де є алергени і промо |
| 13 | Milk 2 litrs 3,2% | 75,00 | 5 | TRUE | TRUE | TRUE |  | 14 | Вивести усі дані з таблиці де є алергени і вироблені в Україні |
| 14 | Egg 10 | 55,00 | 10 | TRUE | FALSE | TRUE |  | 15 | Вивести усі дані з таблиці де є алергени або промо |
| 15 | Green apple 1 kg | 15,00 | 11 | FALSE | FALSE | FALSE |  | 16 | Вивести усі дані з таблиці де є алергени або вироблені в Україні |
| 16 | Yellow apple 1 kg | 20,00 | 17 | FALSE | FALSE | FALSE |  | 17 | Вивести усі дані з таблиці де немає алергенів (оператор NOT) |
| 17 | Red apple 1 kg | 22,00 | 21 | FALSE | FALSE | FALSE |  | 18 | Оновити назву яєць на Chicken eggs 10 |
| 18 | Potatoes 1 kd | 17,00 | 37 | FALSE | FALSE | TRUE |  | 19 | Додати новий продукт Wine 0,7 litr / 250 UAH / 10 botles / Allergic / Promo / EU |
| 19 | Tomat 1 kg | 75,00 | 14 | TRUE | FALSE | FALSE |  | 20 | Видалити хліб з таблиці |
| 20 | Chicken meat 1 kg | 165,00 | 14 | FALSE | TRUE | TRUE |  | 21 | Видалити багет і масло з таблиці |

# 

|  | **Запити SQL для відображення даних з таблиці Products:** |
| --- | --- |
| 1 | SELECT \* FROM Products; |
| 2 | SELECT \* FROM Products WHERE Allergic =’TRUE’; |
| 3 | SELECT \* FROM Products WHERE Promotion =’TRUE’; |
| 4 | SELECT \* FROM Products WHERE Allergic =’FALSE’; |
| 5 | SELECT \* FROM Products WHERE Promotion =’FALSE’; |
| 6 | SELECT \* FROM Products WHERE Made\_in \_Ukraine =’TRUE’; |
| 7 | SELECT \* FROM Products WHERE Made\_in \_Ukraine =’FALSE’; |
| 8 | SELECT *Name* FROM Products; |
| 9 | SELECT *Name,* Price, Quantity FROM Products; |
| 10 | SELECT \* FROM Products ORDER BY Price; |
| 11 | SELECT \* FROM Products ORDER BY Price DESC; |
| 12 | SELECT \* FROM Products ORDER BY *Name*; |
| 13 | SELECT \* FROM Products WHERE Allergic = ’TRUE’ AND Promotion = ’TRUE’; |
| 14 | ​​SELECT \* FROM Products WHERE Allergic = ’TRUE’ AND Made\_in \_Ukraine = ’TRUE’; |
| 15 | SELECT \* FROM Products WHERE Allergic = ’TRUE’ OR Promotion = ’TRUE’; |
| 16 | SELECT \* FROM Products WHERE Allergic = ’TRUE’ OR Made\_in \_Ukraine = ’TRUE’; |
| 17 | SELECT \* FROM Products WHERE NOT Allergic = ’TRUE’; |
| 18 | UPDATE *Products* SET *Name = ‘Chicken eggs 10’* WHERE *Id=14;* |
| 19 | INSERT INTO Products (*Name,* Price, Quantity,Allergic, Promotion, Made\_in \_Ukraine)  VALUES (’Wine 0,7 litr’, ’250,00’, ’10’, ’TRUE’, ’TRUE’, ’FALSE’); |
| 20 | DELETE FROM ProductsWHERE *Name=‘Bread’*; |
| 21 | DELETE FROM ProductsWHERE *Name=‘Baggett’* OR *Name=‘Butter’*; |

# 

# 

# Урок 19

# Урок 18

**За допомогою** [API для навчання. Урок 17](https://docs.google.com/document/d/1WUu2YWRtvDOHk4GbIS2ZekMnJzUzuKk7/edit?usp=sharing&ouid=115143950622232718184&rtpof=true&sd=true) **побудуй POST, GET, PATCH, DELETE запити з використанням:**

* змінних в тілі запиту;
* змінних в параметрах запиту;
* мінімум 2-х тестів для кожного запиту.

Експортовану колекцію додай в LMS.

**Для кожного з методів POST, GET, PATCH, DELETE створи 1 позитивну та 1 негативну перевірку у вигляді запитів з використанням:**

* змінних в тілі запиту
* змінних в параметрах запиту
* мінімум 2 тестів для кожного запиту

# Урок 17

**1. В Google docs створи 5 тест-кейсів для функціонального тестування сайту** [**https://automoto.ua/uk/**](https://automoto.ua/uk/) **(5 найбільш пріоритетних тестів). Посилання на документ додай до ЛМС.**

1. **Перевірка роботи пошуку автомобіля за параметрами:**

Кроки:

* Відкрити домашню сторінку сайту automoto.ua.
* Знайти поле пошуку та ввести параметри пошуку (наприклад, марка, модель, рік випуску, ціна тощо).
* Натиснути кнопку пошуку.

Очікуваний результат: Список автомобілів відображається згідно з введеними параметрами.

1. **Перевірка переходу на сторінку конкретного автомобіля:**

Кроки:

* Відкрити домашню сторінку сайту automoto.ua.
* Вибрати будь-який автомобіль зі списку.
* Клацнути на посилання, щоб перейти на сторінку автомобіля.

Очікуваний результат: Користувач переходить на сторінку з детальною інформацією про обраний автомобіль.

1. **Перевірка можливості сортування результатів пошуку:**

Кроки:

* Відкрити домашню сторінку сайту automoto.ua.
* Виконати пошук автомобіля з будь-якими параметрами.
* Вибрати опцію сортування за ціною або іншими параметрами.

Очікуваний результат: Результати пошуку відсортовані відповідно до обраного параметра.

1. **Перевірка роботи функції фільтрації результатів пошуку:**

Кроки:

* Відкрити домашню сторінку сайту automoto.ua.
* Виконати пошук автомобіля з будь-якими параметрами.
* Використати фільтри для обмеження результатів (наприклад, за типом кузова, роком випуску, об'ємом двигуна тощо).

Очікуваний результат: Список автомобілів показується, враховуючи обрані фільтри.

1. **Перевірка роботи контактної форми:**

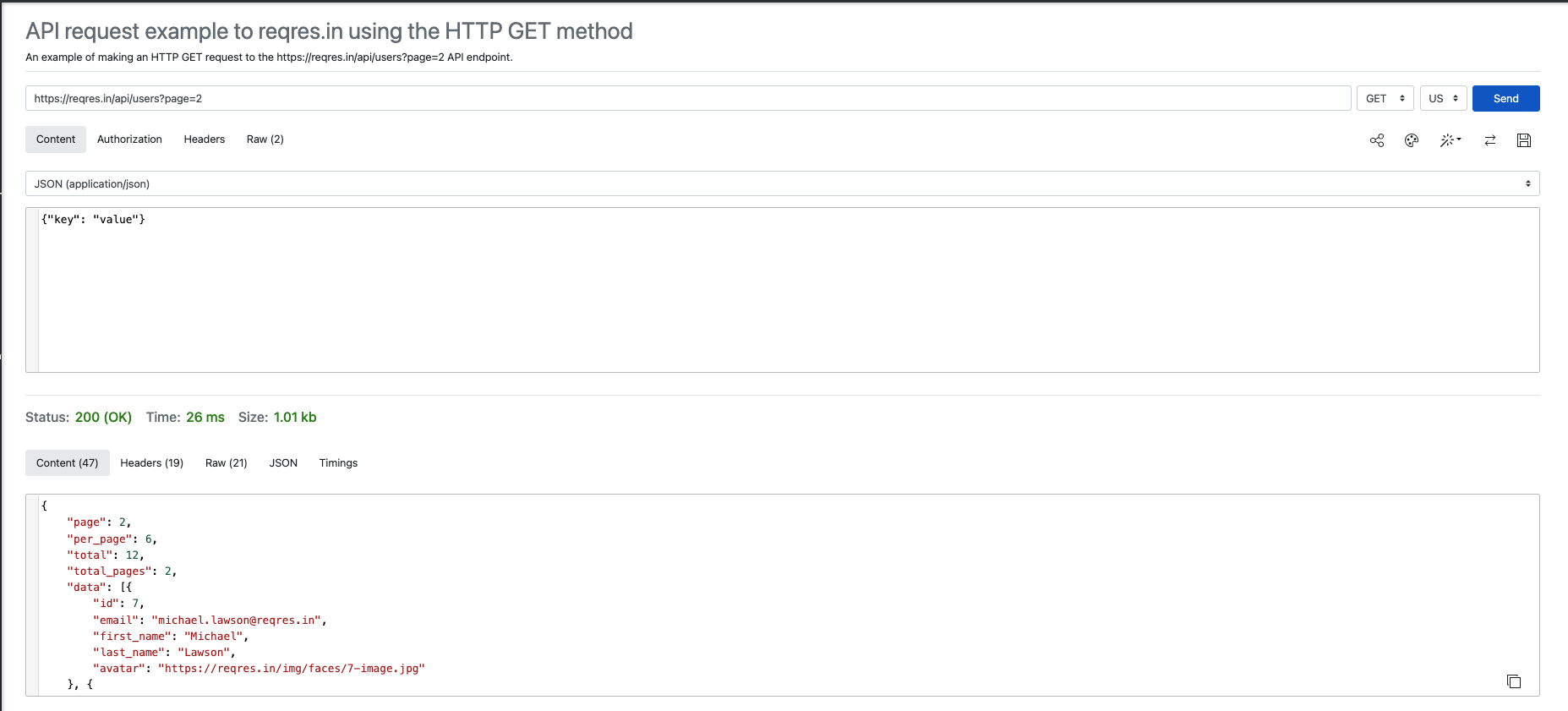
Кроки:

* Відкрити домашню сторінку сайту automoto.ua.
* Знайти контактну форму або сторінку з контактами.
* Заповнити всі обов'язкові поля форми (ім'я, електронна адреса, телефон, повідомлення).
* Надіслати заповнену форму.

Очікуваний результат: Після відправлення форми користувач отримує підтвердження успішної відправки або повідомлення про помилку, якщо така сталася.

**2. За допомогою** [**https://reqbin.com/**](https://reqbin.com/) **протестуй****наступні методи API** [**Reqres**](https://reqres.in/)**:**

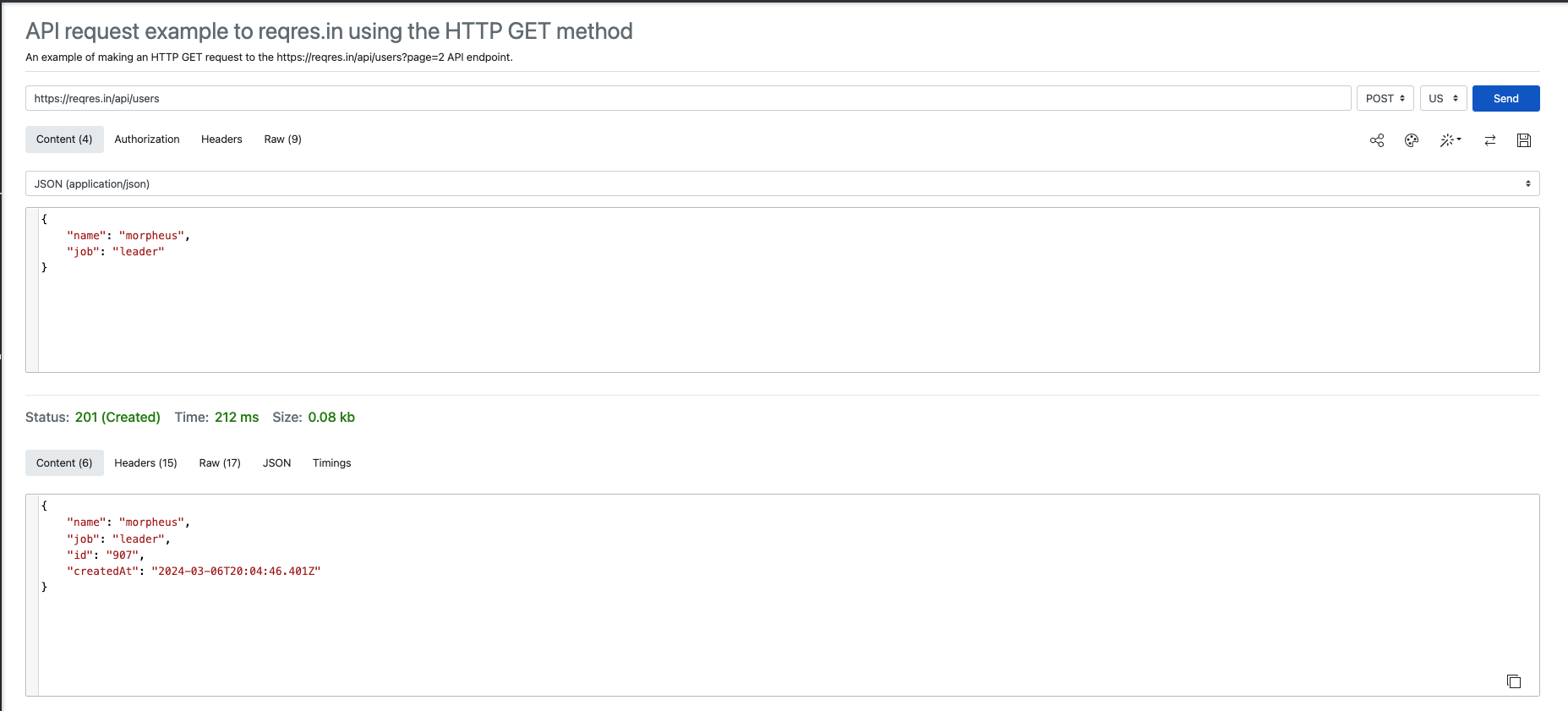
* LIST USERS



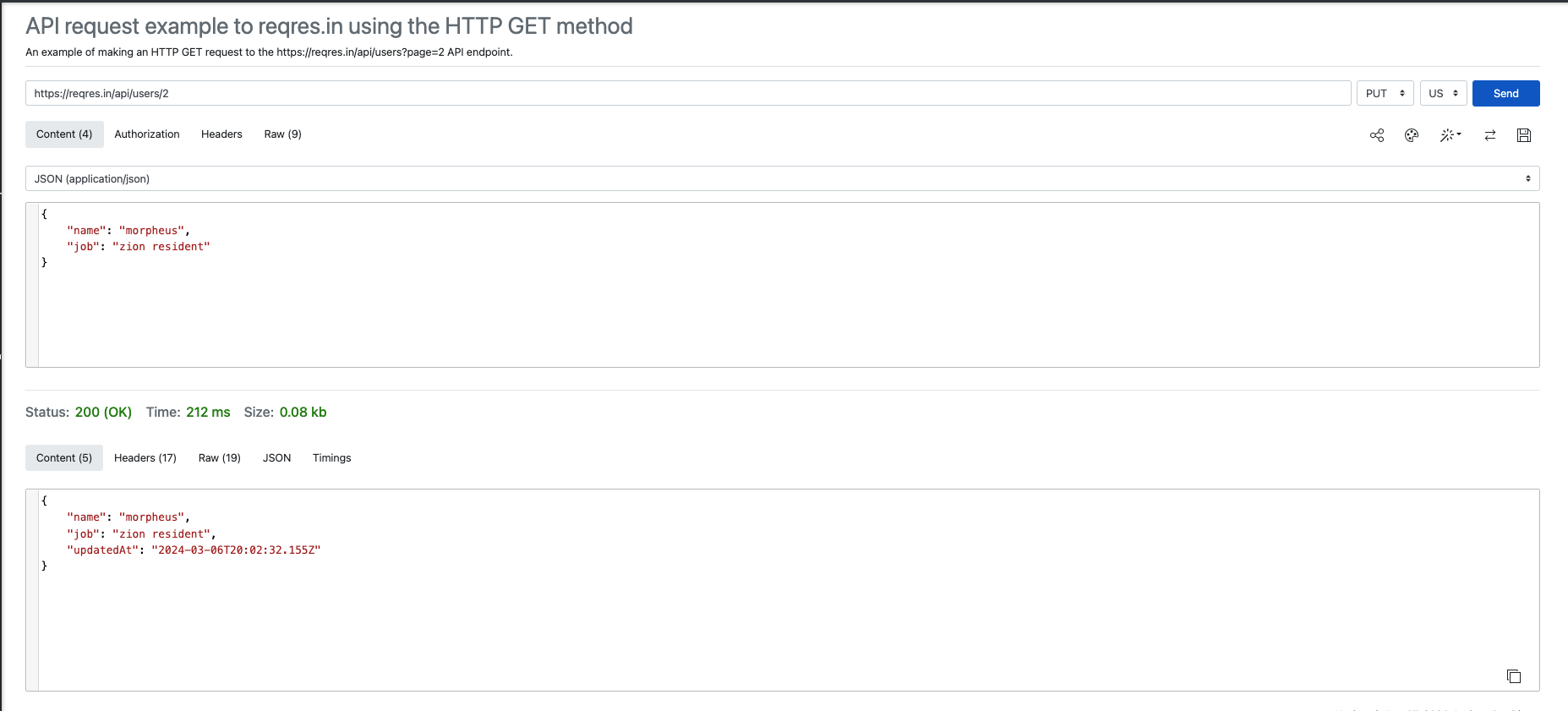
* SINGLE USER



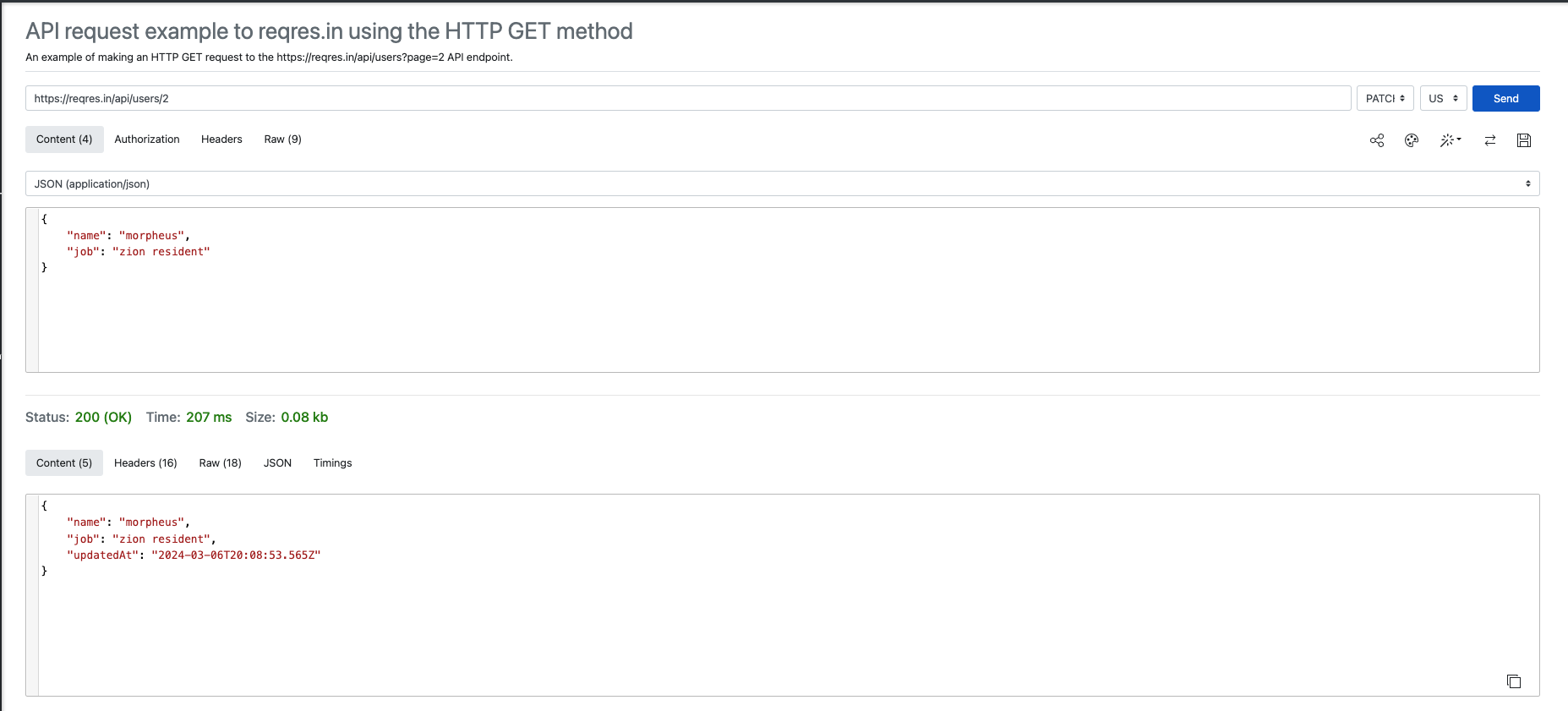
* CREATE



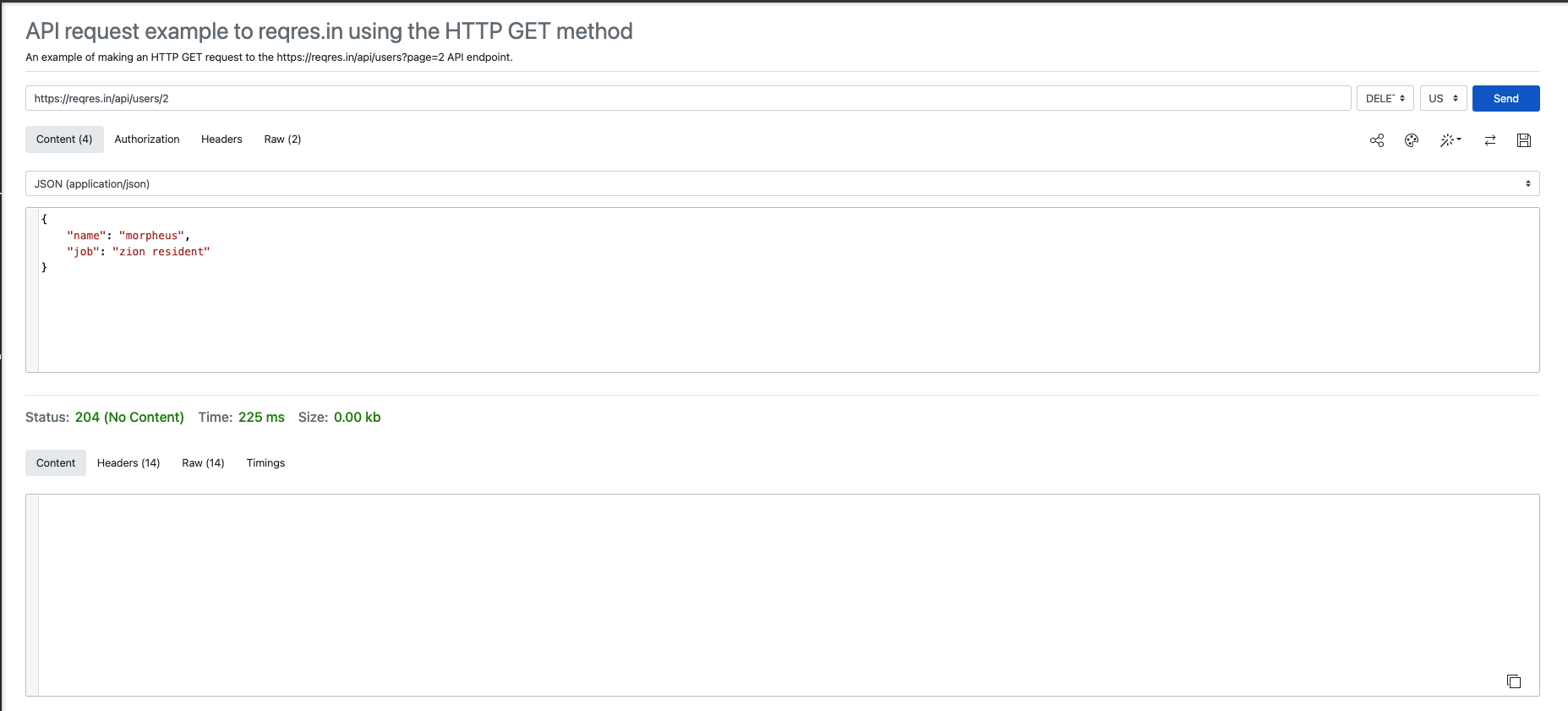
* UPDATE (PUT)



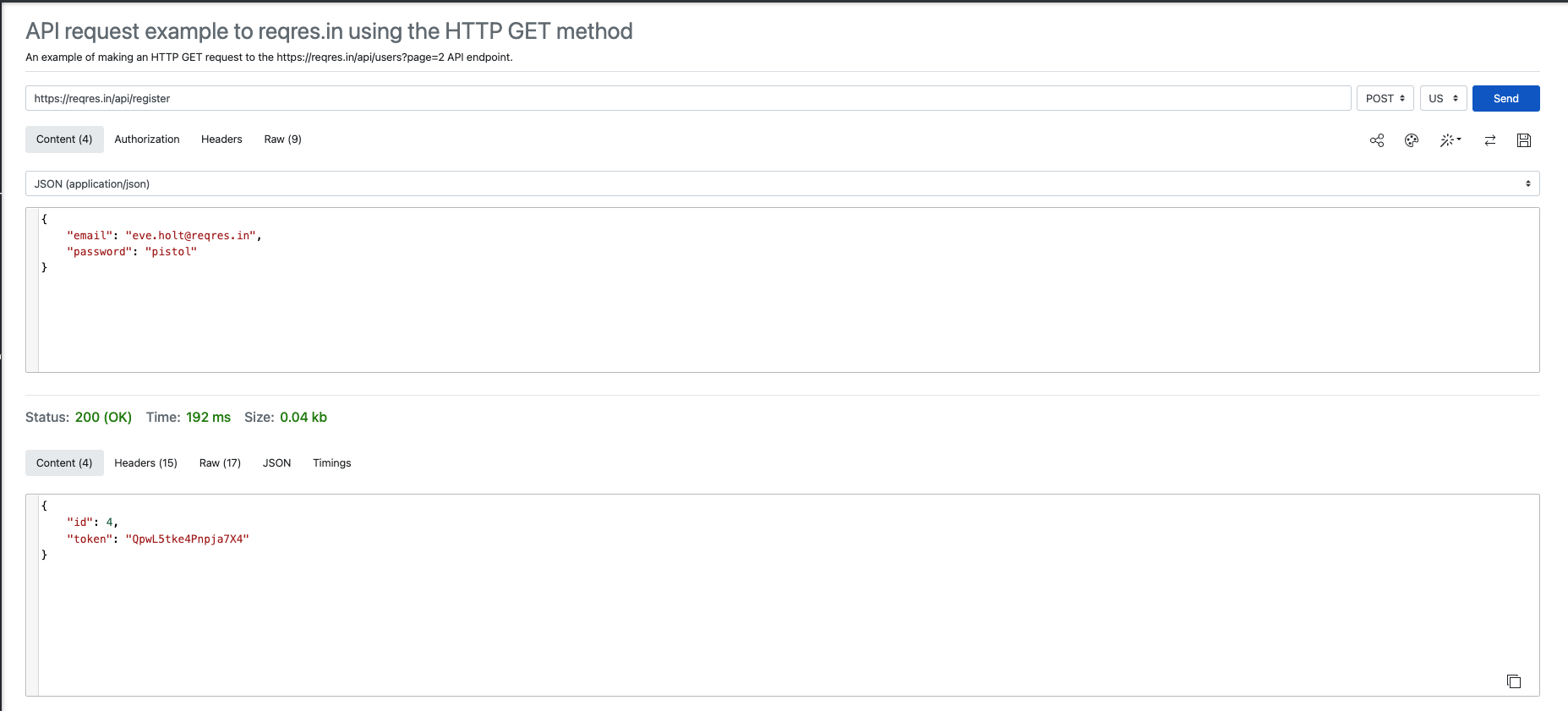
* UPDATE (PATCH)



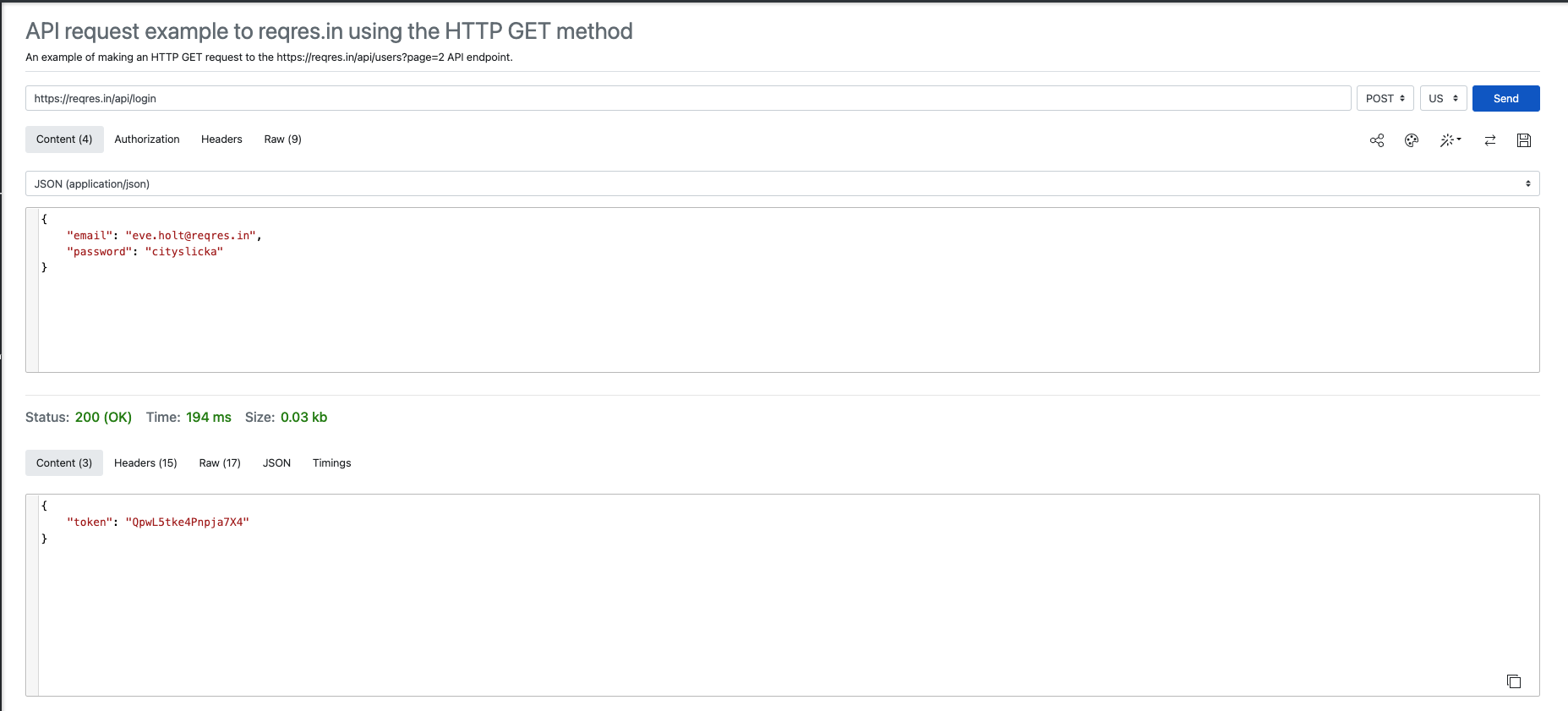
* DELETE



* REGISTER - SUCCESSFUL



* LOGIN - SUCCESSFUL



# 

# Урок 16

1. Протестуй роботу сайту “[Київська міська лікарня ветеринарної медицини](https://kmlvm.kyivcity.gov.ua/)” на мобільному пристрої.
2. Заведи 5 відповідних баг репортів у файл Google Docs. Додай посилання на нього в LMS.

| Bug-reports |  |  |  |
| --- | --- | --- | --- |
| Summary | У розділі Шукають домівку не відцентровано назву розділу | У розділі Шукають домівку не відцентровано кнопки | У розділі Шукають домівку при натисканні на іконку фільтрація зʼїзжає інформація |
| Pre-conditions | вхід на сайт | вхід на сайт | вхід на сайт |
|  | відкрити розділ Шукають домівку | відкрити розділ Шукають домівку | відкрити розділ Шукають домівку |
| Steps | 1. Відкрити розділ шукають домівку 2. Впевнитись, що назва розділу не відцентрована | 1. Відкрити розділ шукають домівку 2. Впевнитись, що кнопки впираються в банер з назвою розділу та знизу залишається багато вільного простору | 1. В розділі Шукають домівку спуститись до іконки фільтрації 2. Натиснути на іконку 3. Неможливо відфільтрувати, бо всі дані перекрити карткою тварини |
| Expected result | Назва розділу посередині та відцентрована. Зайвий пустий простір видалено. | Кнопки посередині та відцентровані. Зайвий пустий простір видалено. | Всю інформацію видно, та можна обирати та натискати фільтри |
| Actual result |  |  |  |
| Post-conditions |  |  |  |
| Environment | Google Chrome 305.610629488 | Google Chrome 305.610629488 | Google Chrome 305.610629488 |
| Severity | Low | Low | Medium |
| Priority | Low | Low | Medium |
| Type of bug | ui/ux | ui/ux | Functional |
| Attachment |  |  |  |

1. Встанови AndroId Studio за посиланням [Download AndroId Studio and SDK tools](https://developer.android.com/studio).
2. Запусти Емулятор, використовуючи [інструкцію](https://developer.android.com/studio/run/emulator).
3. Знайди 3 баги на сайті “[Звірополіс](https://zpolis.com.ua/)”. Додай їх в документ з баг репортами з попереднього рівня.

\*В баг репортах на скриншотах/відео повинно бути зрозуміло, що тестування проводилось саме з емулятора.

# Урок 15

1. **Протестуй UI/UX сайту** [**Оксфорд Медикал**](https://oxford-med.com.ua/)**. В Google docs створи 5 багів з дотриманням усіх вимог до оформлення багу.**

| Bug-reports |  |  |  |
| --- | --- | --- | --- |
| Summary | При невірно заповненій формі підписки попап іншою мовою | Відсутня інформація про лікарів в англійській версії сайту | При зміні міста у вкладці Ціни зникає інформація |
| Pre-conditions | вхід на сайт | вхід на сайт | вхід на сайт |
|  | на головній сторінці, нижній правий кут | відкрити вкладку [Лікарі](https://oxford-med.com.ua/ua/doctors/) | на головній сторінці верхній лівий кут |
| Steps | 1. Знайти в правому нижньому куті іконку подарунка 2. Натиснути на неї 3. Ввести не валідний емейл | 1. Обрати мову сайту українську 2. Обрати будь якого лікаря (наприклад ГАЛИЧ СЕРГІЙ ПЕТРОВИЧ) натиснути на його картку 3. Відкривається попап з Інформаціʼю про лікаря 4. Змінити мову на англійську 5. Відкривається попап з інформацією про лікаря | 1. Відкрити головну сторінку 2. Обрати місто (наприклад Вінниця) 3. Обрати розділ Ціни |
| Expected result | Підказка має бути відповідно до обраної мови сайту | Інформація про лікарів має бути на всіх мовних версіях сайту | Розділ Ціни має бути заповнена у всіх містах |
| Actual result |  |  |  |
| Post-conditions |  |  |  |
| Environment | Google Chrome Версія 122.0.6261.69 (Офіційне складання) (x86\_64) | Google Chrome Версія 122.0.6261.69 (Офіційне складання) (x86\_64) | Google Chrome Версія 122.0.6261.69 (Офіційне складання) (x86\_64) |
| Severity | Low | Low | Low |
| Priority | Low | Medium | Low |
| Type of bug | Functional | Functional | Functional |
| Attachment |  |  |  |

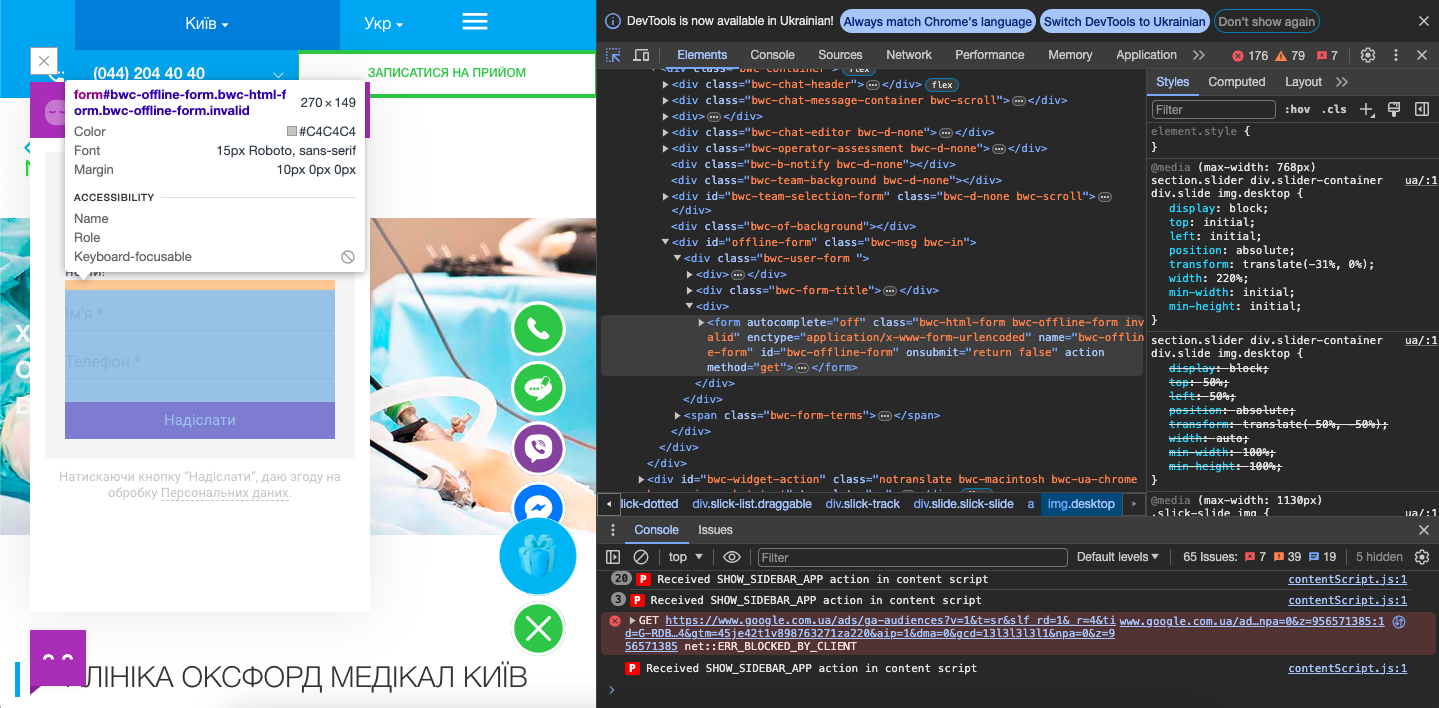
1. **Проведи кросбраузерне тестування знайдених раніше багів у трьох різних браузерах.**
2. **Додай результати до зафіксованих раніше багів.**

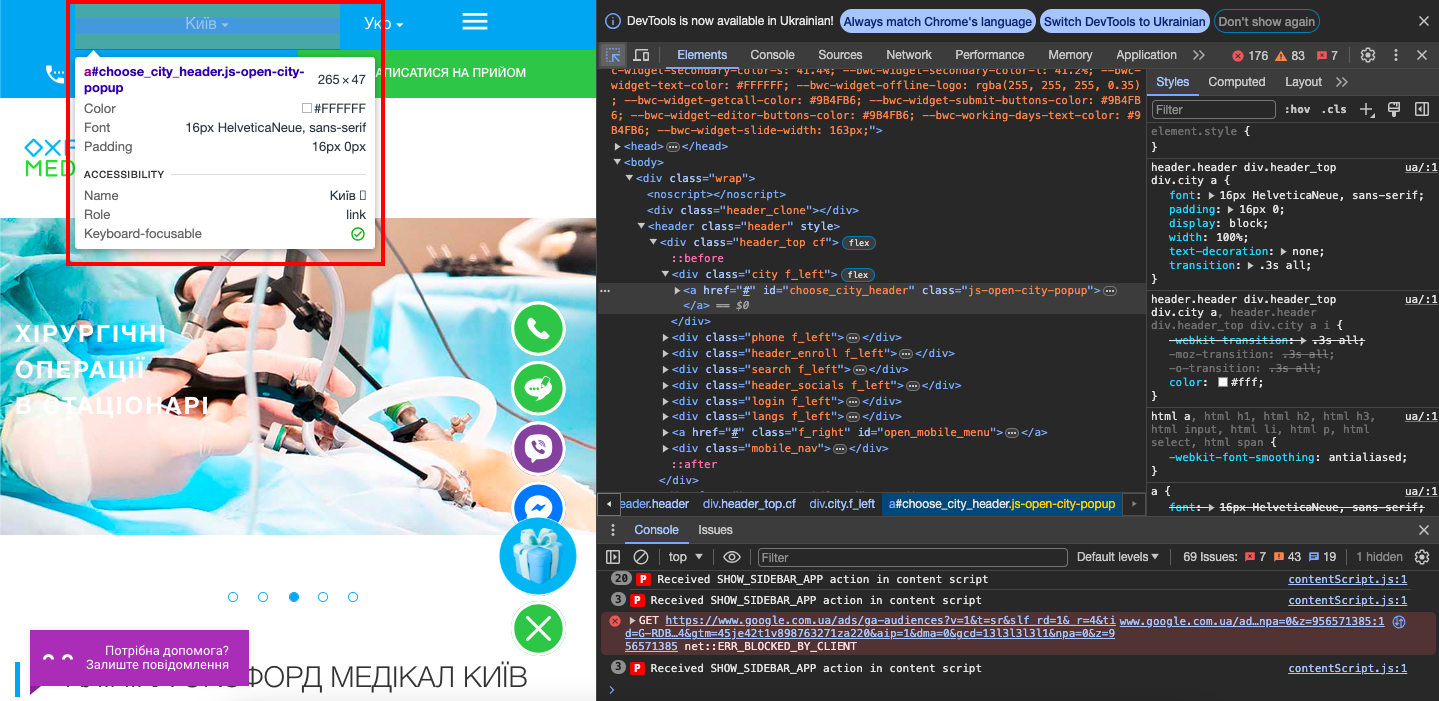
| Safari 5.1.7 | Safari 5.1.7 | Safari 5.1.7 |
| --- | --- | --- |
|  |  | Коли обираєш інші міста крім відсутності інформації в розділі Ціни зникає Англійська версія сайту, та колір кнопки стає в тон хедера |

1. **За допомогою Developer Tools зафіксуй використаний шрифт, його розмір, колір та колір бекграунду наступних елементів сайту** [**Оксфорд Медикал**](https://oxford-med.com.ua/)**:**

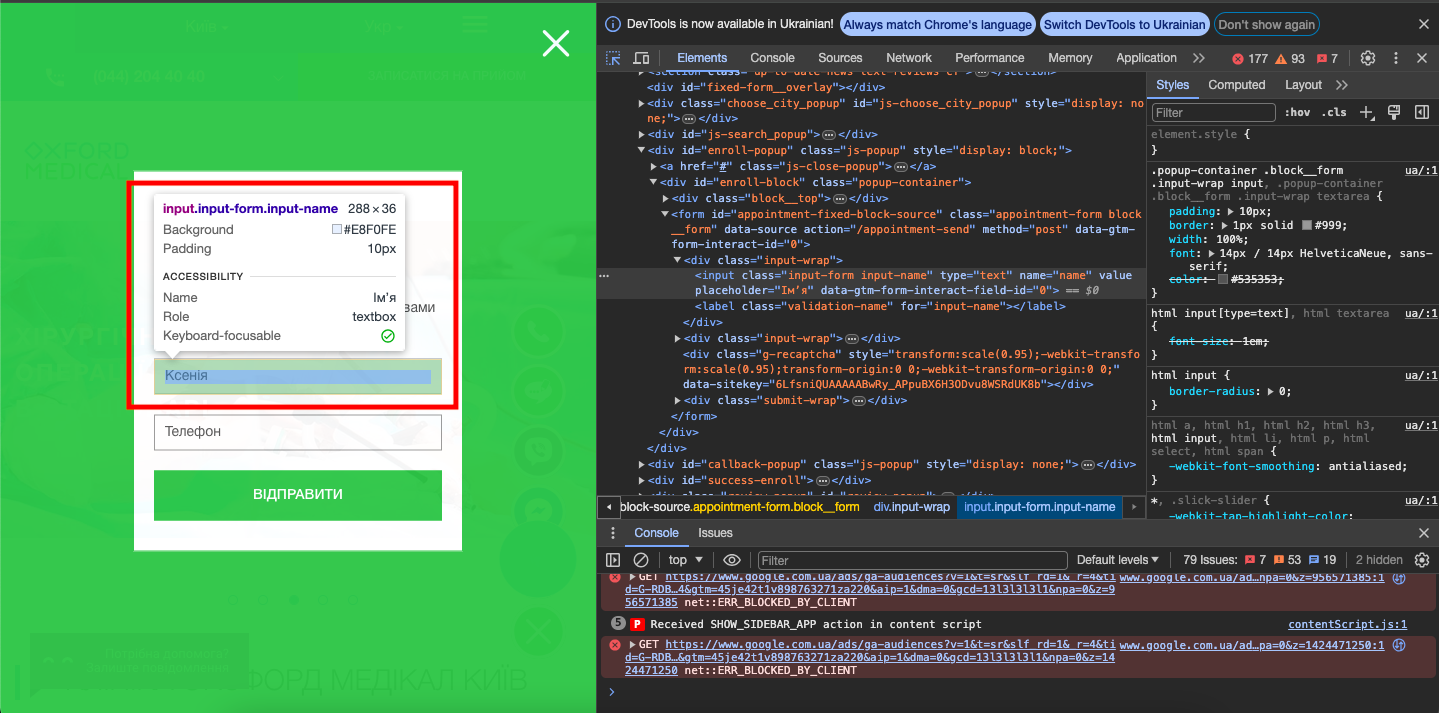
* номер телефону у хедері



* ім’я асистента при виклику чата допомоги (час коли робила домашку, асистенти спали)
* кнопка “інші міста” у футері сайту



* Placeholder “ім’я” у формі “ЗАПИСАТИСЯ НА ПРИЙОМ”

****

# Урок 14

**1. Протестуй сайт** [**http://opencart.qatestlab.net/index.php?route=common/home**](http://opencart.qatestlab.net/index.php?route=common/home) **, знайшовши 3 баги; запиши їх в Google docs.**

| Bug-reports |  |  |  |
| --- | --- | --- | --- |
| Summary | Неможливо обрати обовʼязковий параметр при купівлі товару [Easipet Fabric Pet Carrier, Medium,Blue](http://opencart.qatestlab.net/index.php?route=product/product&product_id=30) | При додаванні товари у кошику дублюються | Не на всіх товарах є іконки (додати в обране, швидкий перегляд та порівняння) |
| Pre-conditions | вхід на сайт | вхід на сайт | вхід на сайт |
|  | відкрити товар [Easipet Fabric Pet Carrier, Medium,Blue](http://opencart.qatestlab.net/index.php?route=product/product&product_id=30) | відкрити товар будь який (наприклад [Hing Designs The Bone Bowl with Non Slip Rubber Feet and Dishwasher Safe Removable Stainless Steel Bowls, Pink](http://opencart.qatestlab.net/index.php?route=product/product&path=20&product_id=34)) | обрати категорію (наприклад [Dogs](http://opencart.qatestlab.net/index.php?route=product/category&path=20)) |
| Steps | 1. Знайти Drop-down list “SIZE” 2. Натиснути на Drop-down list “SIZE” | 1. Впевнитись, що корзина пуста 2. Відкрити товар 3. Обрати Колір та Розмір 4. Натиснути ADD TO CART | 1. Відкрити категорію [Dogs](http://opencart.qatestlab.net/index.php?route=product/category&path=20) 2. Спуститись вниз до карток товарів 3. Навести курсор на перший товар 4. Навести курсор на другий товар |
| Expected result | Мають з'явитися варіанти розмірів для обраного товару | Має бути лише той товар, який був обраний для покупки | Іконки мають бути та працювати на всіх товарах |
| Actual result |  |  |  |
| Post-conditions |  |  |  |
| Environment | Версія 122.0.6261.69 (Офіційне складання) (x86\_64) | Версія 122.0.6261.69 (Офіційне складання) (x86\_64) | Версія 122.0.6261.69 (Офіційне складання) (x86\_64) |
| Severity | Low | Low | Low |
| Priority | Low | Medium | Low |
| Type of bug | Functional | Functional | Functional |
| Attachment |  |  |  |

**2. Створи єдиний HTML-документ з підключенням в ньому стилів трьома різними способами:**

* В середині специфічного тегу (inline)

<https://drive.google.com/file/d/10250gICMsV-4QkLRM_l4kkmirnFg_AQO/view?usp=drive_link>

**4. За допомогою Developer Tools визнач кодування символів, використаних на таких вебсторінках:**

| * [https://beetroot.academy](https://beetroot.academy/) | **utf-8** |
| --- | --- |
| * <https://microseniors76.com/> | **iso-8859-1** |
| * [Tennis Warehouse](https://www.tennis-warehouse.com/) | **UTF-8** |
| * [FIdelity Investments](https://www.fidelity.com/) | **UTF-8** |

# Урок 12

**1. Створи високорівневий тест-кейс для перевірки функціоналу мобільного застосунка (наприклад, Дія або МоноБанк).**

Перевірити основний функціонал мобільного застосунка для користувача.

Кроки:

1. Відкрити мобільний застосунок на смартфоні.
2. Перевірити, що стартовий екран відображається коректно і без затримок.
3. Перевірити наявність основних елементів управління (наприклад, кнопки меню, пошуку, налаштувань тощо) на головному екрані.
4. Натиснути кнопку для переходу на екран входу/реєстрації.
5. Перевірити, що користувач може ввести свої дані для входу або зареєструвати нового.
6. Здійснити вхід у систему з використанням існуючого облікового запису або створити новий.
7. Перевірити доступність головного меню після входу у систему.
8. Вибрати одну з основних функцій застосунка (наприклад, перегляд списку продуктів або послуг).
9. Перевірити коректність відображення вмісту на обраному екрані.
10. Виконати дії відповідно до обраної функції (наприклад, додати продукт до кошика чи здійснити покупку).
11. Перевірити, що дії користувача виконуються без помилок та затримок.
12. Перевірити коректність виведення повідомлень про стан системи або результат виконаних дій.
13. Завершити роботу з мобільним застосунком, вийшовши з програми або перехід на інший екран.

Очікуваний результат:

Усі кроки тест-кейсу виконуються успішно, без помилок або затримок. Весь функціонал мобільного застосунка працює коректно і відповідає очікуванням користувача.

**2. Створи 3 низькорівневі тест-кейси на основі високорівневого.**

**Низькорівневий тест-кейс №1**: Перевірка відображення стартового екрану

Перевірити, що стартовий екран мобільного застосунка відображається коректно та без затримок.

Пре-умови:

* Мобільний застосунок встановлено на пристрої.

Кроки:

1. Відкрити мобільний застосунок на смартфоні.
2. Перевірити, що стартовий екран відображається коректно.
3. Перевірити наявність основних елементів управління (наприклад, кнопки меню, пошуку, налаштувань тощо) на стартовому екрані.
4. Перевірити відображення зображень або іконок, які використовуються на стартовому екрані.
5. Перевірити коректність виведення текстових елементів на стартовому екрані.
6. Перевірити, що кнопки та інші елементи управління реагують на дотик користувача.

Очікуваний результат:

Стартовий екран мобільного застосунка відображається без помилок та затримок. Всі елементи управління працюють коректно, і користувач може без перешкод взаємодіяти з ними.

**Низькорівневий тест-кейс №2**: Перевірка входу в систему

Перевірити коректність процесу входу в систему за допомогою існуючого облікового запису.

Пре-умови:

* Мобільний застосунок встановлено на пристрої.

Кроки:

1. Відкрити мобільний застосунок на смартфоні.
2. Натиснути кнопку для переходу на екран входу/реєстрації.
3. Ввести існуючі облікові дані (логін та пароль).
4. Натиснути кнопку для входу в систему.
5. Перевірити, що користувач успішно ввійшов у систему.
6. Перевірити, що користувач перенаправлений на головний екран після входу в систему.
7. Перевірити наявність основних елементів управління на головному екрані після входу в систему.

Очікуваний результат:

Користувач успішно входить у систему за допомогою існуючого облікового запису. Після входу в систему користувач перенаправляється на головний екран, де він може використовувати основний функціонал застосунка.

**Низькорівневий тест-кейс №3**: Перевірка функціоналу авторизації користувача

Перевірити функціонал авторизації користувача в мобільному застосунку.

Пре-умови:

* Мобільний застосунок встановлено на пристрої.
* Користувач має обліковий запис для входу в систему.

Кроки:

1. Відкрити мобільний застосунок на смартфоні.
2. Натиснути кнопку "Увійти" на екрані вітаємо.
3. Ввести валідні облікові дані (ім'я користувача та пароль).
4. Натиснути кнопку "Увійти" або виконати відповідну дію для підтвердження введених даних.
5. Перевірити, що користувач успішно авторизується і переходить на головний екран.
6. Перевірити наявність основних елементів управління на головному екрані після авторизації (наприклад, меню, кнопки, вміст тощо).

Очікуваний результат:

Користувач успішно авторизується в мобільному застосунку, і після цього відбувається перехід на головний екран, де він може користуватися всіма доступними функціями.

**3. Проестимуй, скільки часу знадобиться для виконання тест-кейсів з попереднього завдання.**

**4. Для оцінки використай як мінімум 2 техніки (наприклад, WBS та трьохточкову естимацію).**

# 

# Урок 11

**1. Яке твердження щодо діаграми переходу станів і таблиці з тест-кейсами є вірним?**



| Тест-кейс | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- |
| Стан початку | S1 | S2 | S2 | S3 | S3 |
| Живлення | Power On | Power Off | RC On | RC off | Power Off |
| Стан завершення | S2 | S1 | S3 | S2 | S1 |

1. Дані тест-кейси покривають валідні і невалідні переходи на діаграмі.
2. **Дані тест-кейси показують всі можливі валідні переходи на діаграмі.**
3. Дані тест-кейси покривають деякі валідні переходи на діаграмі.
4. Дані тест-кейси покривають пари переходів на діаграмі.

**2. Співробітникам компанії виплачують бонуси за умови, що вони пропрацювати більше ніж рік та виконали заздалегідь поставлені цілі.**

Ці умови можна подати у вигляді таблиці рішень:

| Тест |  | Т1 | Т2 | Т3 | Т4 |
| --- | --- | --- | --- | --- | --- |
| Умова 1: | Стаж більше року? | Так | Ні | Ні | Так |
| Умова 2 | Ціль поставлена? | Ні | Ні | Так | Так |
| Умова 3 | Ціль досягнута? | Ні | Ні | Так | Так |
| Дія | Виплата бонуса | Ні | Ні | Ні | Так |

Який сценарій, що є ймовірним в реальному житті, пропущений в таблиці?

1. Умова 1 = ТАК, Умова 2 = НІ, Умова 3 = ТАК, Дія = НІ
2. Умова 1 = ТАК, Умова 2 = ТАК, Умова 3 = НІ, Дія = ТАК
3. Умова 1 = НІ, Умова 2 = НІ, Умова 3 = ТАК, Дія = НІ
4. **Умова 1 = НІ, Умова 2 = ТАК, Умова 3 = НІ, Дія = НІ**

**2. Склади діаграму станів і переходів для тестування відеогри:**

Після запуску гри ти опиняєшся в «Замку загадок», у кімнаті з двома коридорами — направо і наліво.

Якщо піти направо — потрапляєш до дракона. Дракон загадує тобі загадку. Якщо відповідь правильна — виходиш із замку та виграєш. Якщо неправильна — дракон загадує ще одну загадку. Якщо ти вдруге відповідаєш неправильно — дракон тебе з'їдає і ти програєш.

Якщо піти наліво — потрапиш до відьми. Відьма загадує загадку, якщо відповідь правильна — виходиш із замку і виграєш. Якщо відповідь неправильна — відьма переносить тебе до дракона.



**3. Скільки тест-кейсів, відповідно до складеної діаграми, буде достатньо, щоб протестувати цю гру?**

**7 тест-кейсів**

# Урок 10

**1. Система контролю за швидкістю руху має наступні характеристики:**

* при швидкості 50 і менше км/год – система не реагує
* при швидкості більше ніж 50, але менше ніж 55 км/год – система видає попередження
* при швидкості більше ніж 55, але менше ніж 60 км/год – система випише штраф
* при швидкості більше, ніж 60 км/год – водій отримає штраф та штрафний бал у водійське посвідчення

Швидкість в системі вимірюється цілими значеннями. Який з наборів тестових даних можна використати для перевірки всіх граничних значень еквівалентних класів?

1. 0, 49, 50, 54, 59, 60
2. 50, 55, 60
3. 49, 50, 54, 55, 60, 62
4. **50, 51, 55, 56, 60, 61**

**2. Фітнес застосунок рахує кількість кроків і надсилає користувачам повідомлення, щоб заохотити їх рухатися. В залежності від кількості кроків фідбек буде таким:**

* до 1000 кроків включно – “Житель дивана”
* від 1000 до 2000 кроків включно – “Лежибока”
* від 2000 до 4000 кроків включно – “Рухай тілом!”
* від 4000 до 6000 кроків включно – “Непогано!”
* більше 6000 – “Молодець, так тримати!”

Який набір тестових даних надасть найкраще покриття класів еквівалентності?

1. 0, 1000, 2000, 3000, 4000
2. 1000, 2001, 4000, 4001, 6000
3. 123, 2345, 3456, 4567, 5678
4. **666, 1999, 2222, 5555, 6666**

**Пристрій, що вимірює час та інтенсивність сонячного світла, яке отримала рослина, рахує комбінацію параметрів – час на сонці (менше ніж 3 години, від 3 до 6 годин та понад 6 годин) та інтенсивність світла (дуже низька, низька, середня, висока).**

Є наступний набір тестів:

|  | Години | Інтенсивність | Індекс опромінення |
| --- | --- | --- | --- |
| Тест 1 | 1,5 | дуже низька | 10 |
| Тест 2 | 7 | середня | 60 |
| Тест 3 | 0,5 | дуже низька | 10 |

Яка **мінімальна** кількість додаткових тест-кейсів потрібна, щоб упевнитися, що всі валідні класи еквівалентності покриті?

1. 1
2. **2**
3. 3
4. 4

**2. Застосунок для відтворення відео має вимоги. Застосунок буде працювати на пристроях з такою розподільчою здатністю:**

1. 640x480
2. 1280x720
3. 1600x1200
4. 1920x1080

Який тест-кейс є результатом застосування техніки розподілення на класи еквівалентності? Обгрунтуй свою відповідь.

1. Перевірити, що застосунок відтворює відео на дисплеї 1920х1080 (1 тест-кейс)
2. Перевірити, що застосунок відтворює відео на дисплеї 640х480 та 1920х1080 (2 тест-кейси)
3. **Перевірити, що застосунок відтворює відео на дисплеях всіх розмірів, вказаних у вимогах (4 тест-кейси)**
4. Перевірити, що застосунок відтворює відео на дисплеї будь-якого розміру, що вказані у вимогах (1 тест-кейс)

Цей тест-кейс відображає клас еквівалентності для всіх можливих розподільчих здатностей пристроїв і перевіряє, чи відтворюється відео на всіх зазначених розподільчих здатностях безперервно та без втрати якості.

# Урок 9

1. **Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.**

|  | Статистична техніка тестування | Динамічна техніка тестування |
| --- | --- | --- |
| Основна інформація | Припускає, що програмний код під час тестування не буде виконуватися. При цьому саме тестування може бути як ручним, так і автоматизованим. | Аналізується поведінка програми під час її роботи. |
| Перевага №1 | Виявлення проблем на ранніх етапах. | Перевірка реальної роботи програми. |
| Перевага №2 | Ефективне виявлення аномалій в документації | Виявлення дефектів, які не можуть бути виявлені статичним тестуванням. |
| Перевага №3 | Можливість автоматизації аналізу коду. Може бути проведений на будь-якому етапі розробки. | Відповідність реальним умовам експлуатації. |
| Обмеження №1 | Не може гарантувати виявлення всіх дефектів. | Вимагає витратних ресурсів для виконання. Виконується після написання програмного забезпечення. |
| Обмеження №2 | Вимагає аналітичних навичок для ефективного проведення. | Обмежена можливість покриття коду. |
| Обмеження №3 | Може бути обмеженим доступ до вихідного коду. | Вимагає попереднього розроблення програмного забезпечення. |
| Висновок | Обидва підходи важливі для забезпечення якості програмного забезпечення, і їх використання в комбінації може забезпечити більш повну оцінку та виявлення дефектів. | |

1. **Наступне твердження стосується покриття рішень:**

*Коли код має одну ‘IF” умову, не має циклів (LOOP) або перемикачів (CASE), будь-який тест, який ми виконаємо, дасть результат 50% покриття рішень (decision coverage).*

**Яке твердження є коректним?**

* 1. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.
  2. **Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.**
  3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
  4. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.

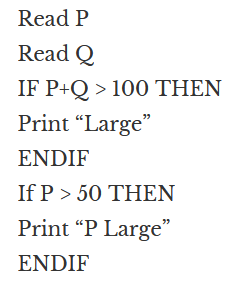
1. **Є псевдокод:** Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word.

Скільки тест кейсів знадобиться, щоб перевірити його функціонал?

* 1. **1 – для покриття операторів, 2 – для покриття рішень**
  2. 1 – для покриття операторів, 1 – для покриття рішень
  3. 2 – для покриття операторів, 2 – для покриття рішень
  4. 2 – для покриття операторів, 1 – для покриття рішень



1. **Скільки потрібно тестів для перевірки тверджень коду:**



* 1. 2
  2. 1
  3. 3
  4. 4

**Є алгоритм:**

*Запитай, якого улюбленця має користувач.*

*Якщо користувач відповість, що має кота, то запитай, яка порода його улюбленця: «короткошерста чи довгошерста?»*

*Якщо клієнт відповість «довгошерста», то запитай: «ви бажаєте отримати контакти найближчого грумера?»*

*Якщо клієнт відповість «так», то скажи: «Надайте адресу найближчої котячої перукарні»*

*Інакше*

*Скажи: «Запропонуй магазин з товарами по догляду за шерстю»*

*Закінчити*

*Інакше*

*Скажи «Запропонуй обрати магазин із зоотоварами»*

*Закінчити*

*Якщо клієнт не має кота*

*Скажи “Коли вирішите завести улюбленця – приходьте”*

*Закінчити*

**

**Завдання:**

1. Намалюй схему алгоритму Google Docs редакторі
2. Який потрібен мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані?

**Потрібно 4 тест-кейси, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані**

# Урок 8

В баг-трекінговій системі (Jira) опиши 3 баги, які знайдеш на сайті <https://www.headhunterhairstyling.com/>

<https://baqa126group.atlassian.net/browse/OTP-64?atlOrigin=eyJpIjoiMmFjODlmNDY4YmJiNDVjNGEzYTkxYWIyNTExM2U5ZGIiLCJwIjoiaiJ9>

<https://baqa126group.atlassian.net/browse/OTP-65?atlOrigin=eyJpIjoiYjFlMTM1NTY5ZjRlNDMzNWFmNzc4ZDEzZjJjNjBlODMiLCJwIjoiaiJ9>

Наведи власні приклади багів, які можуть мати такі комбінації::

* Severity - Critical/Priority - Low
* Severity - Minor / Priority - Highest

(на кожен варіант вкажи по 2 баги)

* **Severity - Critical / Priority - Low:**
  + Помилка в авторизації: Користувач може авторизуватись з введенням неправильних даних, але система надає доступ до облікового запису з обмеженими правами. Хоча це критичний недолік з точки зору безпеки, пріоритет його виправлення може бути визначений як низький, оскільки він не заважає основному функціоналу системи.
  + Проблема з відображенням зображень: Під час завантаження сторінки зображення не завантажуються правильно або взагалі не відображаються, що може призвести до поганого користувацького досвіду. Цей баг може бути важливим для виправлення, але якщо він не впливає на основний функціонал сайту, пріоритет виправлення може бути встановлений як низький.
* **Severity - Minor / Priority - Highest:**
  + Помилка валідації форми замовлення: При введенні невірних даних у формі замовлення (наприклад, невірний формат номера телефону або адреси електронної пошти), користувач не отримує повідомлення про помилку і замовлення не може бути оформлене. Це може стати причиною втрати потенційних клієнтів, тому виправлення цього багу може бути найвищим пріоритетом.
  + Помилка виведення ціни: Невірна відображення ціни товару на сторінці категорії або в кошику. Наприклад, вартість товару може бути вказана нижче або вище фактичної ціни. Цей баг, хоча і має невеликий вплив на функціонал сайту, але може вплинути на рішення користувачів щодо покупки, тому має найвищий пріоритет для виправлення.

# Урок 7

**На твою думку, в чому полягає найбільша користь використання систем тест-менеджменту? Які їх особливості дозволяють цього досягнути? Відповідь текстово обґрунтуй.**

Найбільша користь використання систем тест-менеджменту полягає у забезпеченні структурованого та ефективного управління процесом тестування програмного забезпечення. Основні особливості цих систем, що дозволяють досягти цього:

* Централізація інформації: Системи тест-менеджменту надають одне централізоване місце для зберігання всієї інформації, пов'язаної з тестуванням, включаючи тестові кейси, результати тестів, дефекти, плани тестування тощо. Це дозволяє всім учасникам процесу тестування легко отримувати доступ до потрібної інформації в будь-який час.
* Спрощення планування та організація тестування: Системи тест-менеджменту дозволяють створювати та організовувати тестові плани, розподіляти завдання між командою, призначати тести на виконання та встановлювати пріоритети. Це спрощує процес планування тестування та дозволяє ефективно розподіляти ресурси.
* Відстеження стану тестування: За допомогою систем тест-менеджменту можна легко відстежувати стан виконання тестів, результати тестування, прогрес вирішення дефектів та інші ключові метрики. Це дозволяє вчасно виявляти проблеми, швидко реагувати на них та приймати відповідні заходи.
* Контроль версій та історія змін: Системи тест-менеджменту зазвичай забезпечують можливість відстеження історії змін у тестових кейсах, дефектах та інших елементах. Це дозволяє зберігати контроль над змінами та відстежувати, як вони впливають на процес тестування.

Загалом, використання систем тест-менеджменту сприяє підвищенню ефективності, якості та організованості процесу тестування програмного забезпечення, що робить їх незамінними інструментами для розробників та тестувальників.

**Перенеси тест-кейси, які ти створював/ла на попередніх заняттях, в тест-менеджмент систему TestRail**

<https://qa126groupbeetroot.testrail.io/index.php?/suites/view/12&group_by=cases:section_id&group_order=asc&display_deleted_cases=0>

# Урок 6

**1. Склади порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування.**

|  | Функціональне тестування | Нефункціональне тестування | Тестування зі змінами |
| --- | --- | --- | --- |
| що перевіряється | перевірка функціональності програми відповідно до вимог та очікувань користувача | оцінка роботи програми з точки зору продуктивності, навантаження, безпеки та інших аспектів, які не пов'язані з функціональністю | перевірка програми на наявність помилок та забезпечення її коректності після внесення змін |
| коли застосовується | виконується під час розробки нового програмного забезпечення або при внесенні змін до існуючого | зазвичай проводиться після завершення функціонального тестування, але перед випуском продукта в експлуатацію | проводиться в момент внесення змін до програми або під час випуску нової версії |
| обмеження | не завжди може охопити всі можливі сценарії використання програми | можуть виникнути труднощі в оцінці об'єму навантаження, який може зазначатися в окремих сценаріях використання | може вимагати багаторазового тестування для переконання, що нові зміни не спричиняють проблем з існуючою функціональністю |
| особливості | дозволяє перевірити, чи відповідає програма вимогам з точки зору функціональності | оцінює продуктивність, надійність, безпеку та інші аспекти програми, які не пов'язані з конкретними функціональними вимогами | допомагає забезпечити стабільність та надійність програми після внесення змін, а також ідентифікувати можливі проблеми, які можуть виникнути в результаті змін |

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

Регресійне тестування та ретестінг - це дві різні стратегії тестування, які використовуються для різних цілей та в різний час у життєвому циклі продукту.

**Регресійне** тестування спрямоване на перевірку програми після внесення змін, щоб переконатися, що ці зміни не спричинили появу нових помилок або не порушили існуючий функціонал.

**Ретестінг** полягає в повторному виконанні тестів для перевірки, чи були виправлені помилки, які були виявлені та документовані під час попереднього тестування.

Основна відмінність полягає в тому, що **регресійне** тестування **виконується після змін** у програмі, а **ретестінг** - **після виправлення виявлених помилок**. Обидва види тестування допомагають забезпечити якість програмного забезпечення, але з різних перспектив: регресійне - забезпечує стабільність після змін, ретестінг - перевіряє, чи були виправлені виявлені проблеми.

**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.

Проведення тільки функціонального тестування без перевірки нефункціональних вимог може бути недостатнім для забезпечення якості продукта. Хоча функціональне тестування оцінює відповідність програмного забезпечення функціональним вимогам, нефункціональні вимоги, такі як продуктивність, безпека, надійність і ефективність, також є важливими аспектами для коректної роботи продукту.

Наприклад, прості веб-сайти з базовою функціональністю можуть не потребувати глибокого аналізу нефункціональних аспектів.

Проте для більш складних та критичних систем, таких як банківські системи, медичні програми або програмне забезпечення для авіаційних систем, необхідне проведення нефункціонального тестування, оскільки невиконання вимог щодо безпеки, надійності або продуктивності може призвести до серйозних наслідків, включаючи загрозу життю або масштабні витрати.

Таким чином, хоча функціональне тестування важливе, для забезпечення повної якості продукту важливо також враховувати та перевіряти нефункціональні аспекти.

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

**Смоук-тестування** – це мінімальний набір тестів, який перевіряє основні функції системи або компонента. Завдання – перевірити, чи працює система в загальних рисах, без детальної перевірки. Перевіряє систему на умовне “загоряння”, одразу після її встановлення або змін і перед запуском, щоб переконатися, що основні функції працюють справно. Такий тест дає змогу швидке виявлення серйозних проблем, економію часу та ресурсів, фокусування на ключових аспектах системи та позитивну оцінку її працездатності перед іншими етапами тестування.

Однак не завжди доречно використовувати смоук тестування, наприклад, якщо система має складну архітектуру або багато додаткових функцій, то може бути потрібне більш детальне тестування. Також, якщо зміни у програмі дуже значні, смоук тестування може бути недостатнім для виявлення всіх проблем. Тому важливо оцінювати ситуацію та використовувати смоук тестування там, де воно найбільш ефективне і доцільне.

# Урок 5

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

| Назва тестової документації | Основні характеристики | Переваги | Недоліки |
| --- | --- | --- | --- |
| Чек-ліст | список запитань або тверджень, які треба перевірити | * може бути застосований для будь якого типу тестування * простий та швидкий у використанні | * часові витрати * статичний і не завжди враховує зміни, які можуть бути під час розробки |
| Тест-кейс | набір вхідних значень, передумов виконання, очікуваних результатів і пострезультатів виконання | * включає в себе аспекти продуктивності та безпеки * деталізований * можуть бути основою для автоматизації | * підтримка великої кількості тест-кейсів може бути витратною та часомісткою * не гнучкі * якщо не оновлювати, то перестануть бути актуальними |
| Use Case | опис, як система взаємодіє з користувачем | * моделювання взаємодії користувача з системою * повний огляд, як система буде використана в реальному середовищі * структурований формат опису для різних сценаріїв * через структурність легкий в редагуванні | * обмеженість в описі деталей * складність для великіх проектів |

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

<https://www.readellion.com/>

[Melikova. Домашнє завдання\*](https://docs.google.com/spreadsheets/d/1pRbB-sAm9fKvKnh__eh6_5Q6J3YbGC-XA98fTp5fZ2E/edit?usp=sharing)

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

<https://www.readellion.com/>

[Melikova. Домашнє завдання\*](https://docs.google.com/spreadsheets/d/1pRbB-sAm9fKvKnh__eh6_5Q6J3YbGC-XA98fTp5fZ2E/edit?usp=sharing)

Ти – засновник/ця стартапу, який планує випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв. Стартап має команду з 3 розробників (back-end, iOS та Android). Вимоги та сценарії використання надаєш ти як головний стейкхолдер.

Завдання: Створи тест-план, який дозволить випустити продукт належної якості.

# Урок 4

**Наведи короткі приклади вимог (3-5) до будь-якого предмета з твого оточення, які б відповідали кожному з таких критеріїв оцінки якості.**

**Термопляшка**

* атомарність

зберігання та підтримка температури

легка та зручна для переносу

гарантувати герметичність

* несуперечність

зберігання температури не менше 4 годин

безпечні матеріали

точні розміри та обʼєм

легкий догляд

* тестованість

утримувати температуру рідини від 0 до 100 градусів

матеріали відповідають стандартам безпеки

абсолютна герметичність

легко мити вручну без ризику пошкодження

* відстежуваність

?

**На твою думку, яка з технік тестування вимог гарантує максимально можливу якість фінального результату. Відповідь обгрунтуй (3-5 речень).**

думаю окремо і повноцінно не справиться жодна, а от комбінації різних може допомогти забезпечити високу якість продукту.

**прототипування** без нього перевірити вимоги на реалізованість і на зручність користування інтерфейсом складно

написання **тест-кейсів та юз-кейсів** дозволяє подати вимоги у структурованій формі, на основі чого програмістам легко програмувати, а тестувальникам — тестувати. та не до всіх продуктів можна швидко та якісно їх написати

важливо **ставити питання**, щоб перевірити вимоги та зменшити ризики на ранніх етапах, та якщо питання ставить не дуже уважна людина, то перевірка буде не до кінця якісною

**Ти – засновник/ця стартапу, який планує випустити на ринок мобільний застосунок для обміну світлинами котиків.**

**Склади функціональні (5-7 од.) та нефункціональні (5-7 од.) вимоги до застосунку.**

функціональні

* авторизація (вхід для вже існуючих та реєстрація для нових)
* створювати та редагувати профіль користувача
* можливість додавати світлини котиків
* пошук та фільтрація
* взаємодія з контентом (вподобайки, коменти)

нефункціональні

* швидка реакція на дії користувача
* захист особистих данних
* стабільна робота при збільшенні кількості користувачів
* інтуїтивно зрозумілий для користувача
* сумісність з різними платформами

# Урок 3

**Склади порівняльну таблицю найбільш поширених методологій:**

| Назва методології | Сильні сторони | Слабкі сторони |
| --- | --- | --- |
| Waterfall | послідовність | жорсткість  важче впроваджувати зміни після початку розробки |
| детальна документація | тривалі терміни |
| передбачуваність | труднощі тестування  пізнє виявлення проблем |
|  | несумісність зі змінами |
| V-модель | раннє виявлення помилок  тестування на кожному етапі | вимагає добре запланованого графіку робіт і ресурсів |
| знижені ризики та витрати на виправлення помилок | неможливість швидко реагувати на зміни |
| відповідність стандартам  підходить для проектів де безпека та надійність критичні | немає проміжних релізів |
| фіксовані терміни та бюджет на кожному етапі | відсутність зворотного зв'язку з клієнтом |
| Спіральна | адаптація до змін на ранніх етапах | багато часу та ресурсу |
| поступовий розвиток | складність управління через велику кількість ітерацій |
| планування наступних кроків на основі попередніх помилок | не стійки вимоги через постійні зміни |
| зниження ризиків завдяки постійному фідбеку | можливі відхилення від плану |
| Scrum - це мистецтво робити вдвічі більше роботи за половину часу | документується лише те що необхідно | не підходить для великих проектів |
| розробка одразу без ретельного планування | недостатньо уваги до документації (не підійде до проектів медичного обладнання, авіації тощо) |
| обсяг роботи на спинт 2-3 тижні | залежність від комунікації |
| фокус на результатах |  |
| командна робота |  |
| Kanban | візуалізація  усі задачі додаються до плану та оновлюються в міру виконання | не має вираженої структури та постійно змінюється |
| групування  всі завдання поділяються за станом: на виконанні, виконується, виконане | не передбачений чіткий час для виконання задачі |
| гнучкість | не передбачено поділу на етапи |

* **На твою думку, чому з’явився Agile-маніфест?**

Аджайл придумали для того щоб встигати за змінами на ринку. Щоб уникнути довготривалої роботи над продуктом, який в результаті виявляється нікому не потрібним. Світ почав змінюватись настільки швидко, що попередня модель waterfall просто не дає можливості встигати та вчасно адаптуватися до нових умов. Завдяки Agile процеси відбуваються швидше (прототип, тестування, реліз). У той час як waterfall досить повільний і дорогий.

* **Які проблеми він мав вирішити і чи це вдалося?**

Це закономірне продовження еволюції підходів, які використовувалися раніше.

| основні проблеми | вирішення |
| --- | --- |
| неспроможність адаптації | підтримує гнучкість та ітеративний підхід, дозволяючи швидко реагувати на зміни |
| неефективна комунікація з клієнтом | активна взаємодія протягом всього процесу |
| низька залученість клієнта | акцентує на влюченні та активній участі клієнта |
| затримка випуску продукта | ітеративний підхід та регулярні випуски |
| неефективне використання ресурсів | увага на реалізацію цінних функцій для клієнта |

вважаю, що Аджайл вирішив існуючі на той час проблеми. Та кажуть, що більшість успіху досягають ті компанії, які комбінують різні інструменти та методології. Судячи з усього, в недалекому майбутньому agile-практики також будуть еволюціонувати і в управлінні з’являться нові ідеї.

**Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків.**

**Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй.**

вважаю, що для застосунку варто використовувати одну із Айджал систем. Бо вони гнучкі, швидки, та не потребують подробної документації, що важливо при розробці мобільного додатку.

оберу Scrum, бо:

* гнучкість. можливість адаптуватись до змін, що важливо в стартапі
* спринт. розбивка процесів на ітерації з чіткими відрізками часу, по закінченню кожного спринта вже є частина продукта, який можна презентувати користувачу
* включення користувачів. які нададуть фідбек по додатку і можна буде виправити помилки на ранніх стадіях
* комунікація. навряд на розробку додатку в стартапі буде велика команда і купа процесів/документів, які треба буде фіксувати
* наявність візуалізації процесів. дошки на яких будуть зафіксовані всі етапи створення

# 

# 

# Урок 2

1. **Вибери предмет з твого оточення (наприклад, чашку, ноутбук, машину, футболку тощо) і спробуй його протестувати. Поясни, чому ти обрав/ла саме таку перевірку.**

*Джинси*

*чи дійсно то штани і вони можуть захищати ноги та сідниці*

*чи є в них пʼять карманів та 6 шльовок*

*чи з джинсу вони зроблені*

*чи мають гудзики та блискавку*

*чи відповідає колір та матеріал*

*чи якісно зроблені шви*

*чи відповідний розмір*

*чи зносостійкі*

*такої перевірки достатньо для цього продукту, якби це були галузєві (наприклад для війскових чи дснс, то потрібні були б ще перевірки безпеки та відповідних стандартів)*

**

1. **Своїми словами поясни визначення валідації та верифікації.**

*верифікація відповідає вимогам*

*валідація відповідає потребам користувача*

1. **Склади порівняльну таблицю різних видів компаній. Вкажи плюси та мінуси кожної з них (з точки зору працівника).**

| вид компанії | плюси | мінуси |
| --- | --- | --- |
| product | *стабільність* | *великі терміни* |
|  | *карʼєрний розвиток* | *бюрократія* |
|  | *команда* |  |
|  | *сучасне робоче оточення* |  |
|  | *соцпакет* |  |
| outsource | *гнучкість графіку* | *нестабільний дохід* |
|  | *будь яке місце знаходження* | *відсутність соціальних вигід* |
|  | *вибір клієнтів* | *ризик шахрайств* |
|  | *великий ринок робочих можливостей* | *обмежені карʼєрні зростання* |
| startup | *творча свобода* | *ризик невдач* |
|  | *різноманітні завдання* | *фінансова нестабільність* |
|  | *енергія та атмосфера* | *невизначеність ролей* |
|  | *частка в успіху* |  |

1. **Наведи приклади невдалої валідації або верифікації продукту, з якими довелося зіткнутися в житті.**

*онлайн замовлення з аліекспресу то кожен раз невдала валідація та верифікація* 🙂 *незалежно від детального опису та наявних фото від продавця*

1. **Поясни важливість 2-3 принципів тестування на власний вибір. Наведи приклади з власного досвіду.**

*раннє тестування*

*якщо борщ на смак пробувати в процесі готування, то ймовірність що його зʼїдять, а не викинуть збільшується*