



Тестовое задание: QA

Проверка базовых навыков.

Мануальное тестирование (Junior)

Задача: Оценить функциональность и адаптивность сайта компании LLC "dZENcode IT-Company" (<https://dzencode.com>) на различных устройствах и в разных браузерах.

Цель тестирования: Проверьте функциональность и удобство использования сайта компании LLC "dZENcode IT-Company" (<https://dzencode.com>) предоставляющей различные услуги через веб-сайт.

Основные требования:

- Разработка тест-кейсов (10-15 покрывающих основной функционал)
- Поиск и документирование багов (составить БР разной степени критичности)
- Отчёт о тестировании

Разработка тест-кейсов

- Создайте 10-15 тест-кейсов для проверки функционала web-сайта.
- Разделить тест-кейсы на категории (функциональные, UI/UX (проверка интерфейса), доступность)
- Оформите их в табличном виде (например, в Excel, Google Sheets или другой удобной для вас форме).5
- Укажите: название тест-кейса, шаги для выполнения, ожидаемый результат.

Поиск и документирование багов

- Если в процессе тестирования вы обнаружите баги, оформите их в баг-репорты.
- В каждом баг-репорте укажите: название бага, шаги воспроизведения, фактический результат, ожидаемый результат, приоритет.

Ожидаемый результат:

- Подготовьте краткий отчет о проделанной работе.
- Укажите количество найденных багов, их краткое описание, а также успешность прохождения тест-кейсов.
- Включите таблицу или список с указанием всех инструментов, примененных для тестирования (например, браузеры, плагины для анализа, средства для записи багов и т. д.).
- Опишите, для каких задач использовался каждый инструмент (например, Chrome DevTools для анализа производительности)
- Запишите краткое сопроводительное видео с пояснением важных моментов выполнения задания (3 - 5 мин)

Автоматизированное тестирование (Junior+)

Задача: Посетить сайт <https://rahulshettyacademy.com/seleniumPractise/#/> и разработать автоматизированные тесты для проверки функциональности сайта. Тесты должны быть написаны на JavaScript с использованием Cypress или Playwright.

Основные требования:

- Использовать Cypress или Playwright для написания и выполнения тестов.
- Язык написания тестов – JavaScript.
- Разместить проект в публичном репозитории на GitHub или GitLab.
- Подготовить детальный README-файл с инструкциями по развертыванию проекта, включая необходимые зависимости, шаги установки и команды для запуска тестов.

Описание тестов:

1. Перейти на сайт <https://rahulshettyacademy.com/seleniumPractise/#/>.
2. В строке поиска ввести значение **“ro”**.
3. Найти элемент **Carrot** в результатах поиска и установить количество равным **5** с помощью поля ввода (input).
4. Найти элемент **Mushroom** в результатах поиска и установить количество равным **3**, используя нажатия на кнопку увеличения (increment).
5. Добавить **Carrot** в корзину.
6. Добавить **Mushroom** в корзину.
7. Нажать на иконку корзины для открытия списка товаров.
8. Удалить **Carrot** из корзины.

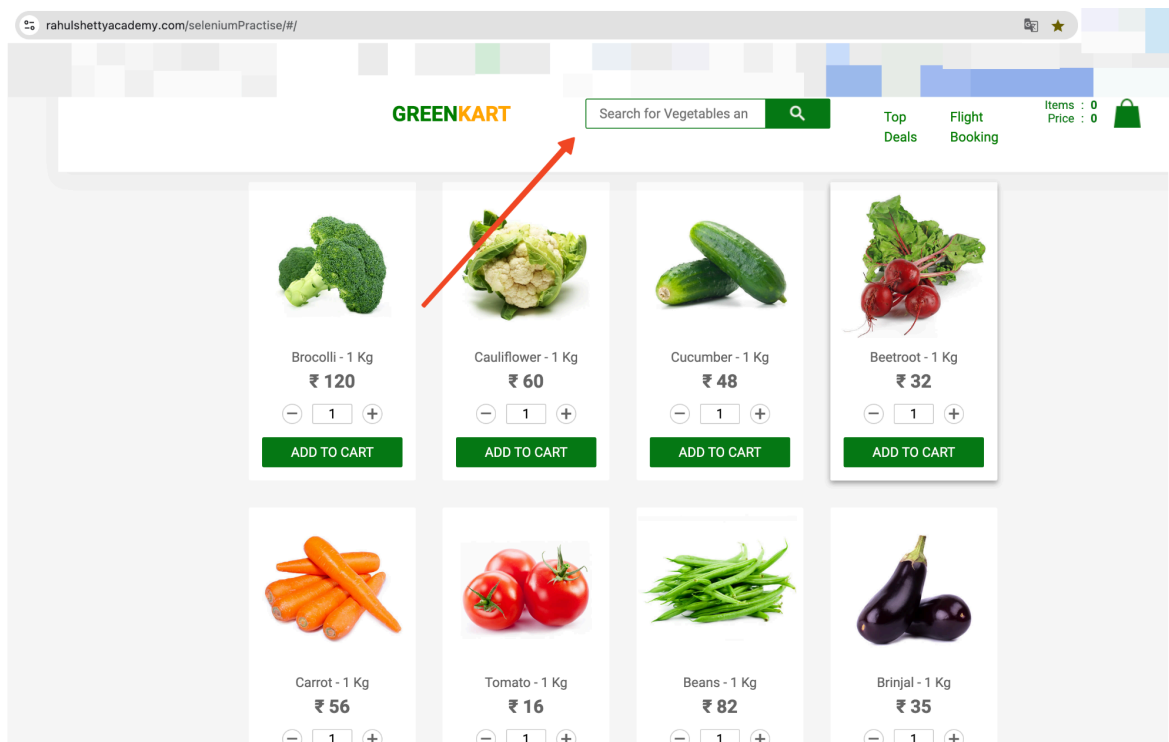
Ожидаемый результат:

- Публичный репозиторий с проектом на GitHub или GitLab, содержащий все необходимые файлы.
- Краткое сопроводительное видео (3–5 минут) с демонстрацией экрана и пояснением ключевых моментов выполнения задания.

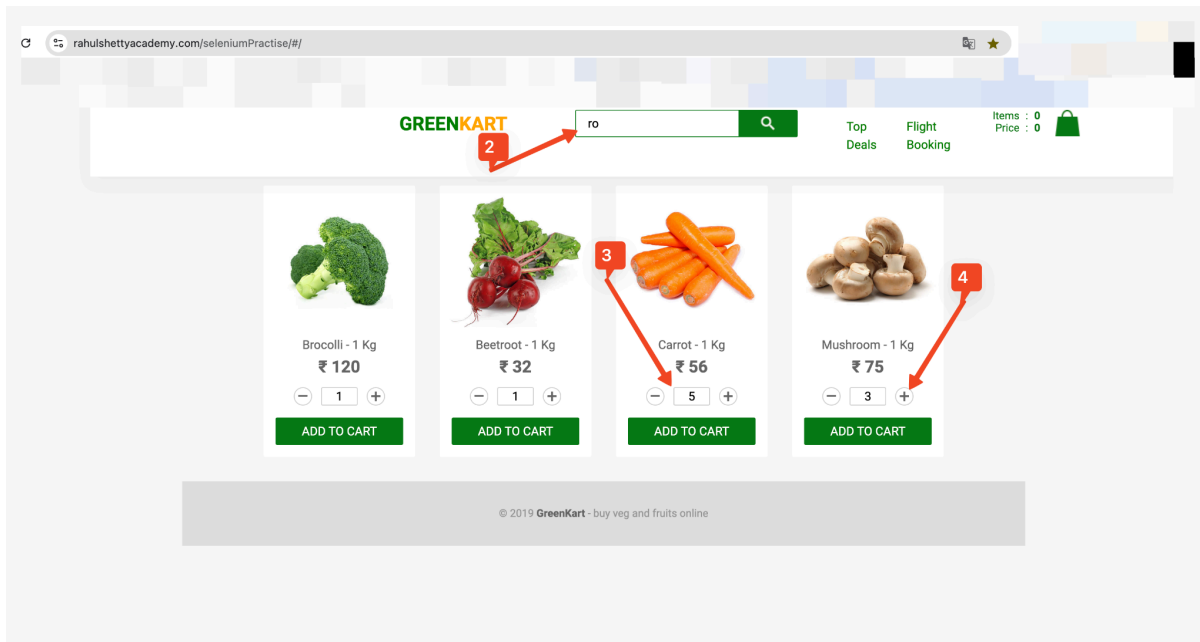
Инструкция:

1. Перейти на сайт <https://rahulshettyacademy.com/seleniumPractise/#/>.
2. В строке поиска ввести значение **"ro"**.
3. Найти элемент **Carrot** в результатах поиска и установить количество равным **5** с помощью поля ввода (input).
4. Найти элемент **Mushroom** в результатах поиска и установить количество равным **3**, используя нажатия на кнопку увеличения (increment).
5. Добавить **Carrot** в корзину.
6. Добавить **Mushroom** в корзину.
7. Нажать на иконку корзины для открытия списка товаров.
8. Удалить **Carrot** из корзины.

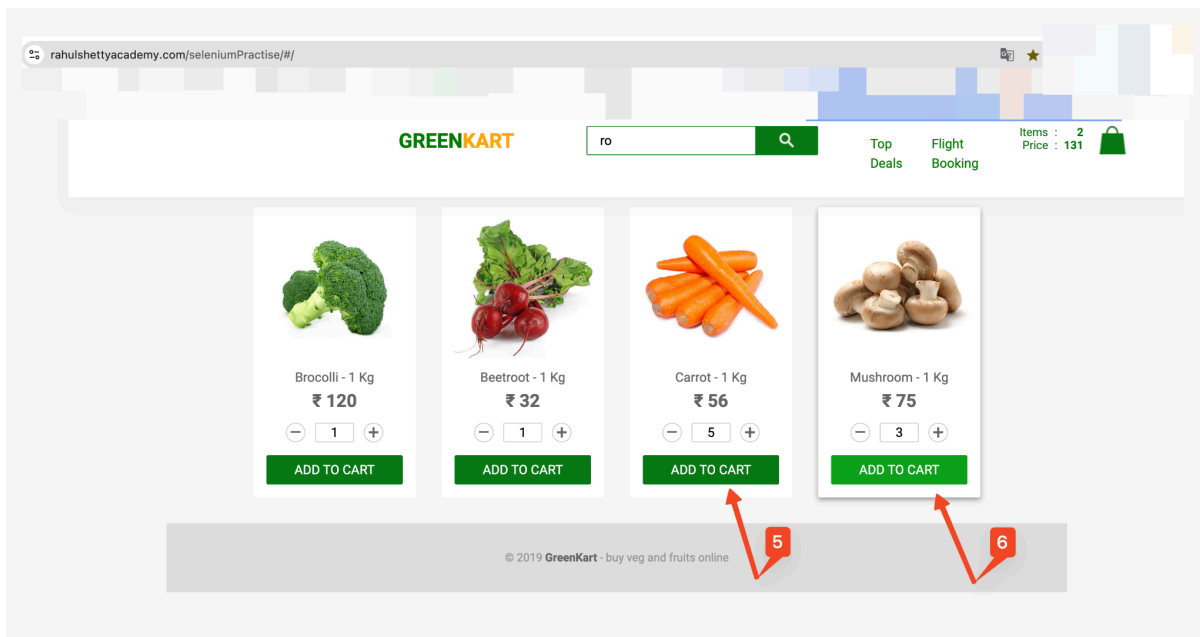
п.1:



п.2-4:



n.5-6:



n.7-8:

