**Задание:** создать git репозиторий, в котором будет лежать тестовый проект. В проекте должны находиться папки *feature*, в которой хранится тест-кейс, и *step\_defs*, в которой будет хранится сам скрипт, и в корне проекта файл *requirements.txt*, в котором перечислены все используемые библиотеки с их версиями.

Используя **pytest (v6.2.4)**, **pytest\_bdd (v4.0.2)**, **webdriver-manager (v3.4.2), selenium,** а также браузер **Chrome,** выполнить следующее:

1. Написать тест-кейс при помощи [Gherkin](https://wellbehaved.readthedocs.io/Gherkin.html):
   1. Открыть браузер и развернуть на весь экран.
   2. Зайти на [market.yandex.ru](file:///C:\Users\natalia\Downloads\Telegram%20Desktop\market.yandex.ru)
   3. В разделе «Каталог → Электроника» выбрать «Смартфоны».
   4. Перейти в «Все фильтры».
   5. Задать параметр поиска до 20000 рублей и Диагональ экрана от 3 дюймов.
   6. Выбрать не менее 5 любых производителей.
   7. Нажать кнопку «Показать».
   8. Посчитать кол-во смартфонов на одной странице.
   9. Запомнить последний из списка.
   10. Изменить Сортировку на другую (по цене/ по рейтингу/ по скидке).
   11. Найти и нажать по имени запомненного объекта.
   12. Вывести рейтинг выбранного товара.
   13. Закрыть браузер.
2. Используя **pytest\_bdd,** написать автоматизированный скрипт и связать его с тест-кейсом([readme](https://github.com/pytest-dev/pytest-bdd#readme)).

**Доп. задание:**

1. Реализовать логирование действий при помощи библиотеки logging.
2. Создать *conftest.py*, в котором будут определены тестовые конфигурации. Создать минимум 1 fixture.

**Ожидаемый результат:** при клонировании проекта и запуске теста той командой, которая указана в readme тестового проекта кандидата, отобразится браузер и выполнятся действия из тест-кейса, шаги которого описаны выше.