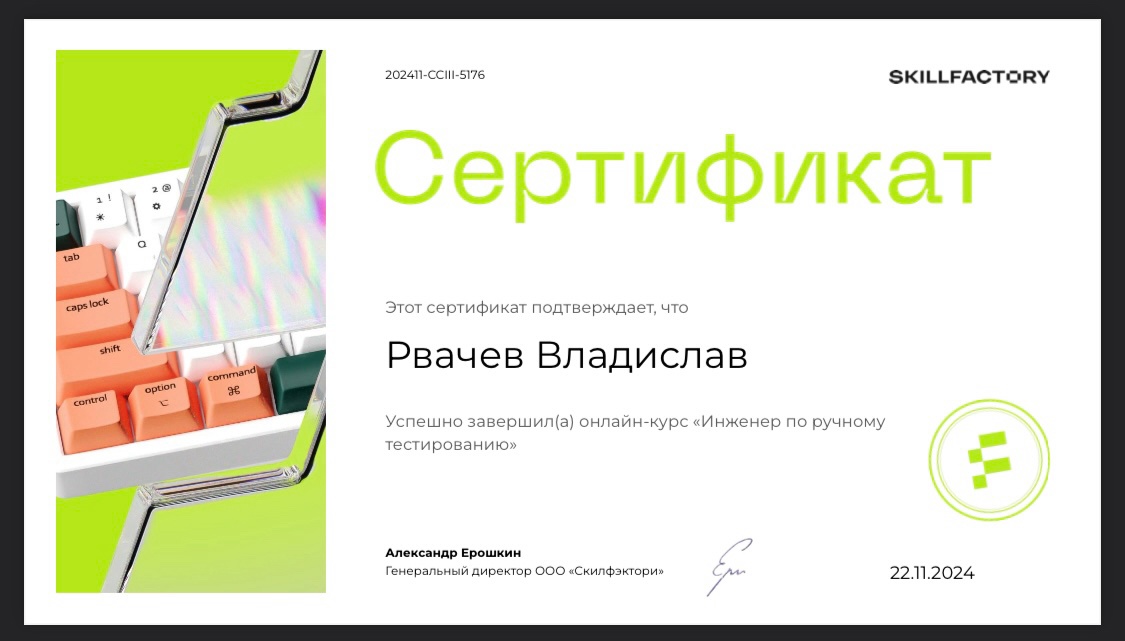
**Портфолио тестировщика.**

**Рвачев Владислав**

Немного о себе: мне 27 лет, живу в городе Санкт-Петербурге. Окончил среднее специальное образование в ВТАЛК в городе Выборг, (год окончания 2015) профессия - “Автомеханик, слесарь по ремонту автомобилей, водитель категории В”. Но профессия “Автомеханик” меня не сильно интересовала, больше интересовался инновационными технологиями. Не так давно понял что хочу заниматься тестированием.

На данный момент успешно закончил курсы “Инженера по ручному тестированию”

Во время учебы я успешно и с энтузиазмом проходил все задания которые были доступны по теме. Заслуженно брал хорошие оценки от менторов курса. Помогал студентам справится с заданиями. Хотелось поделится знаниями. Я выложил все значимые для моего становления работы. Их можно посмотреть в приложенных файлах.

Пройдем по списку, все что тут перечислено сделано мною:  
**Итоговый проект SCAN в Postman.**

1.Нужно было создать переменные логина и пароля, и с помощью csv-файла передать тестовые данные для них. Было сделано 3 позитивных и 5 негативных проверок.

2. Настроить авторизацию для всех заголовков коллекции, динамически получая значение токена с помощью скрипта в запросе «Авторизация пользователя».

3. Получить данные пользователя (Баланс) и записать их в переменную коллекции.

4. Найти соответствующие запрос в Swagger (Темы публикаций новостей). Создать переменную коллекции и записать в неё общее количество публикаций.

5. Получить документы в формате ScanDoc по идентификаторам одним запросом. Передать на сервер два идентификатора в виде переменных.

**Мобилка итоговый проект.**

Тестирование мобильного приложения.

Нужно было сделать тест-кейсы

1. Проводилось тестирование связи.
2. Тестирование прерываний.
3. Взаимодействие нон-стоп.
4. Тестирование геолокации с помощью приложения GPS Emulator.
5. Тестирование потребления ресурсов.
6. Инсталляционное тестирование.
7. Заведение багов в баг-репорт.

**В HTML формате сделал базовую страницу в браузере.**

**Postman collection**

Нужно было создать коллекцию с определенным сценарием.

1. Получение ключа, script - тест проверка статус-кода и запись полученного API-ключа в переменную *auth\_key*.
2. Добавление питомца. В теле запроса значения ключей — переменные. Script - проверка статус-кода, добавление *ID* питомца из ответа в переменную *petid* коллекции *Postman* и запись *name* питомца из ответа в переменную *name* коллекции *Postman.*
3. Изменение имени питомца запросом *PUT*. В теле запроса значения ключей — переменные. Script - проверка предположения, что новое имя, полученное в ответе, не равно старому имени, запись нового имени сохранить в переменную *name* коллекции *Postman*.
4. Получение списка своих питомцев. Script - проверка предположения, что переменная *name* коллекции *Postman* равна значению *name* первого ([0]) питомца, полученного из ответа после выполнения запроса.

**Пироговый дворик итоговая работа**

Был составлен тест-план, сделаны тест-кейсы и чек-лист согласно плану тестирования.

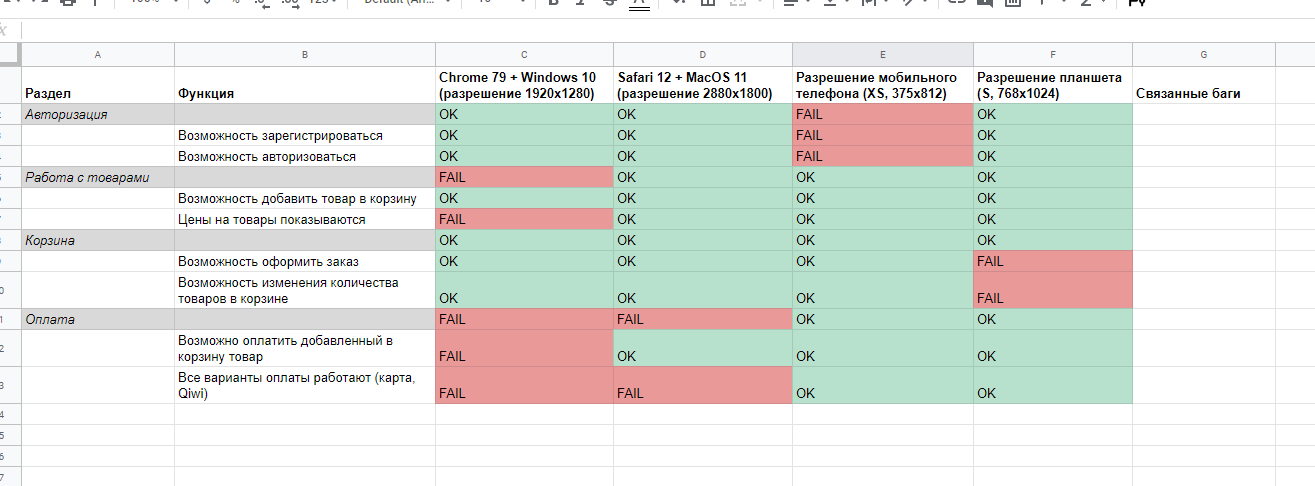
**Кроссбраузерное и кроссплатформенное тестирование.**

1. На основе топ используемых браузеров в России для десктопной ОС выбрать две самые популярные пары браузер+десктопная ОС за 2019 год, используя [сайт](https://gs.statcounter.com/).

2. На основе топ используемых браузеров для мобильных ОС в мире выбрать две самые популярные пары браузер+мобильная ОС за 2019 год.

3. Изучив функционал сайта [Лента.Ру](https://lenta.ru/) или <https://tass.ru/> (на выбор), составить чек-лист для проверки критичного функционала (не нужно составлять чек-лист на весь функционал).

4. Добавить к чек-листу критичного функционала строки на проверку кроссбраузерности, кроссплатформенности и адаптивности. Пример чек-листа из учебной программы прилагаю



5. Провести тестирование по составленному чек-листу на одном сочетании десктопной ОС + браузер и на другом сочетании мобильная ОС + браузер. Сопроводить каждый пункт чек-листа скриншотом сделанной проверки.

6. Использовать для тестирования пробный период онлайн-сервисов для кроссплатформенного тестирования, либо личные ресурсы.

**Три теста ВК.** Можно было взять любой сайт. а к этому времени я нашел несколько неприятных багов в этой социальной сети.

**Экзаменационная работа от заказчика.** Тестирование сайта по требованиям заказчика.

1. Чек-лист
2. Тест-кейсы
3. Баг-репорт.

Все это нужно выполнить по внутренним стандартам заказчика.

**Баг-репорт слот** это приложение игровых автоматов веб-версия. Нужно было найти баги в этой игре, я не справился с этой задачей, нашел слишком мало багов.

**Мир кораблей** - компьютерная игра.

1. Был создан чек-лист по основному функционалу верхних кнопок “В порту”
2. И проверка корабля в бою.

К сожалению обратной связи не дождался потому что, в письме написал что не готов менять свое место работы…

**Тестирование «SafeFrome» тестовое задание**

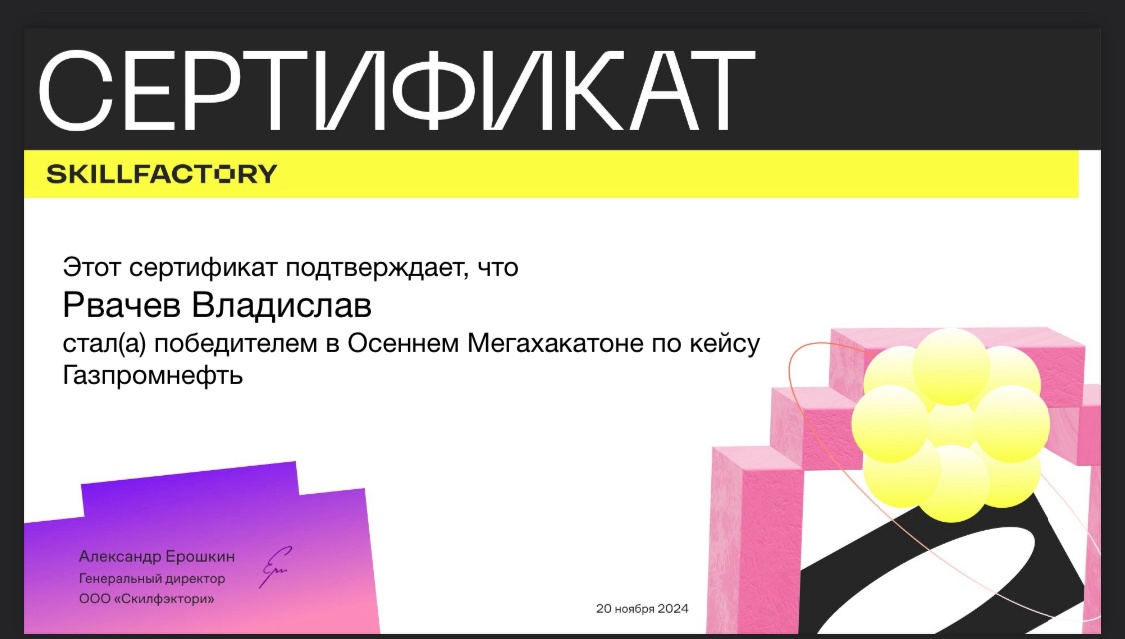
Создание чек-листа и тест-кейсов для популярного сервиса - расширения [SaveFrom.net](https://vk.com/away.php?to=http%3A%2F%2FSaveFrom.net&cc_key=)Helper

1. Разработайте полноценный чек-лист, включающий в себя функциональное, UI и UX тестирование.

2. Тестирование: - На основе вашего чек-листа, проведите тестирование сервиса. - В случае обнаружения багов, создайте баг-репорт. - Создайте тест-кейс про то, как функционал должен был работать без бага.

3.Создание баг-репортов и тест-кейсов: - Для каждого найденного бага создайте отдельный лист в xlsx документе.

**Проект “Хакатон” от SkillFactory** наша команда состояла из 3-х человек и стала победителем!



Задача проекта - провести исследовательское функциональное и нефункциональное тестирование веб-приложения, выявить соответствия сценариям и сложности на пути пользователей, предложить варианты доработки дизайна.

В ходе реализации проекта было сделано следующее:

- при подготовке был сделан тест-план

- тестировалась кроссбраузерность и кроссплатформенность.

В ходе реализации проекта команда подготовила

- чек-листы и заводили баг-репорты на найденные ошибки/баги

- тест-кейсы с помощью техник тест-дизайна по проверке формы авторизации,

- тест-кейсы по проверки веб-приложения с помощью инструментов DevTools,

- план и сценарий проведения юзабилити-тестирования.

- была подготовлена презентация для заказчика, в которой коротко и по делу были озвучены: где уязвимые места приложения, способы улучшения UI и юзабельности.

В процессе проведения тестирования:

- было обнаружено 46 багов с разным приоритетом, исправление которых, в дальнейшем позволит улучшить производительность веб и мобильного приложения.

- было проведено немоделируемое удаленное проверочное юзабилити-тестирование, сформирован отчет по замечаниям респондентов, что позволит улучшить веб-приложения с точки зрения UI/UX. Наша команда состояла из трёх студентов.

Во всем этом я принимал активное участие. То что сделал лично я приложено файлами с пометкой “Хакатон”.

На этом пока все…

Мои личные данные и ссылки:

VK - <https://vk.com/mr.brothers>

Gmail - [bsusbebbdvlad@gmail.com](mailto:bsusbebbdvlad@gmail.com)

Номер телефона - 89817024948