**Тест-план сайта**

[**https://rubikon-test-quest.tilda.ws**](https://rubikon-test-quest.tilda.ws)

Исполнитель Рвачев Владислав

Дата: 07.02.2025

Версия: 1.0

Статус: В работе

Оглавление.

1. *Введение………………………………………………………1*
2. *Рамки проекта……………………………………………….2*
3. *План тестирования……………………………...…………2*
4. *Стратегия тестирования………………………………..2*
5. *Процедура тестирования…………………………………3*
6. *Сообщения об ошибках…………………………………….3*
7. *Основные инструменты и окружения…………………..4*
8. *Критерии качества…………………………………………4*
9. *Результат……………………………………………………4*

**1.Введение.**

1.1 Будет использован метод черного и серого ящика.

1.2 Должно быть проведено функциональное, нефункциональное тестирование сайта [**https://rubikon-test-quest.tilda.ws**](https://rubikon-test-quest.tilda.ws) так же, нужно сделать кроссбраузерное и кроссплатформенное тестирование. Провести тестирование всех кнопок сайта и сделать отчет по тестированию. Найденные баги занести в баг-репорт.

1.3 Цель проекта - выявить уязвимые точки сайта и показать их.

**2 Рамки проекта.**

2.1 Тестирование всего сайта, рамок нет.

**3. План тестирования.**

3.1 В начале провести функциональное тестирование основных кнопок на сайте (чек-лист по всем кнопкам), учитывая нефункциональные части этих кнопок.  
3.2 Сделать тест-кейсы на формы для заполнения, провести основной пользовательский сценарий.

3.3 Проверить запросы через DevTools. Проверить верстку сайта на адаптивность.

3.4 Проверить кроссбраузерность и кроссплатформенность на наличие ошибок.

**4. Стратегия тестирования.**

Цель функционального тестирования состоит в том, чтобы убедиться, что весь программный продукт работает в соответствии с требованиями, и в веб-приложении не появляется существенных ошибок. Функциональное тестирование является наиболее существенной частью тестирования ПО, включающее в себя проверку различных аспектов системы. Программный продукт должен пройти все запланированные тесты. Только в этом случае можно быть уверенным в его качестве.

**5. Процедура тестирования.**

Основные типы тестирования, которые будут выполнены:

* ~~Функциональное тестирование.~~
* ~~Нефункциональное тестирование.~~
* ~~Кроссбраузерность сайта.~~
* ~~Кроссплатформенность сайта.~~

Техники тест-дизайна

* ~~Классы эквивалентности (формы для заполнения)~~

Метод серого ящика

* ~~Используя DevTools провести тестирование запросов и просмотр ошибок в консоли.~~

**6.Сообщения об ошибках.**

Степень серьезности ошибок можно разделить на четыре категории:

● Критические (блокирующие) дефекты - это сбой всей программной системы или критической подсистемы, тестирование не может быть выполнено после возникновения дефекта. Это также относится к сбоям при потере данных и процессам, которые оставляют несогласованные данные, хранящиеся в базе данных.

● Мажорные дефекты — ошибки, которые также приводят к выходу из строя всей или части системы, но существуют некоторые альтернативы обработки, которые позволяют продолжить работу системы. Это также относится к сбою или прерыванию работы системы во время нормальной работы некритического потока.

● Незначительные дефекты не приводят к сбою, но приводят к тому, что система показывает неправильные, неполные или противоречивые результаты.

● Тривиальные дефекты - это небольшие ошибки, которые не влияют на функциональность: опечатки, грамматические ошибки, неправильная терминология и т.д.

**7.Основные инструменты и окружение.**

Инструменты:

* Google.docs
* Sitechco.ru

Окружение:

* Браузер - Microsoft Edge Версия 131.0.2903.112
* Браузер - Opera One(версия: 116.0.5366.71)
* Браузер - Google Chrome Версия 132.0.6834.160
* Windows 11 Pro версия: 23H2
* Galaxy S24 Android 14

**8.Критерии качества.**

Продукт не должен содержать критических и блокирующих дефектов в окончательной версии проекта.

**9.Результаты.**

* План тестирования.
* Баг-репорты.
* Отчет с рекомендациями.