**Задание 1.**

1. Вопросы к требованиям:

- Не указаны ограничения на возможный диапазон вводимых данных.

- Не указано, как система будет взаимодействовать с сервером (внутренний механизм).

- Не уточнено, какие сообщения должны появляться при возникновении ошибок.

2. Виды и типы тестирования:

- Функциональное тестирование: проверка корректности работы формы и правильности вывода сообщений.

- Нагрузочное тестирование: проверка, как система будет обрабатывать большое количество запросов одновременно.

- Тестирование безопасности: проверка на возможность ввода вредоносных данных и попыток взлома.

3. Тест-кейсы или чек-листы:

- Для данной задачи можно использовать как тест-кейсы, так и чек-листы. Тест-кейсы подходят, если требуется детальное описание последовательности действий и ожидаемых результатов. Чек-листы подойдут, если необходимо быстро проверить основные аспекты функциональности.

4. Техники тест-дизайна:

- Эквивалентное разбиение: проверка на разных диапазонах входных данных (отрицательные, нулевые, положительные).

- Граничное значение: тестирование на границе диапазона возраста (например, 17 и 18 лет).

- Тестирование на основе таблицы решений: создание таблицы с входными данными и ожидаемыми результатами.

5. Негативные и позитивные входные данные:

- Позитивные: корректный возраст (например, 25 лет).

- Негативные: отрицательный возраст, нулевой возраст, возраст с плавающей точкой, символы, строка вместо числа.

6. Возможные баги:

- Отсутствие обработки ошибок при вводе некорректных данных (например, букв вместо чисел).

- Неправильное определение совершеннолетия (ошибки в логике вычислений).

- Проблемы с интерфейсом (например, неправильное отображение сообщений).

- Проблемы с перегрузкой сервера при большом количестве запросов.

7. Баги на скриншотах (макетах):

- На первом скриншоте «Введите ваш Возраст?», с точки зрения орфографии и пунктуации рекомендуется написать так: «Введите Ваш возраст.».

- На втором и третьем скриншотах ошибки в словах «совершеннолетний» и «несовершеннолетний».

Помимо этого, рекомендуется также провести ручное тестирование на различных браузерах и устройствах, чтобы убедиться, что функциональность работает стабильно в разных условиях.

## Задание 2.

1. Внешний вид и дизайн:

- Логотип и название компании на видном месте.

- Основное меню (навигация) присутствует и отображается корректно.

- Все элементы дизайна (цвета, шрифты, иконки) соответствуют корпоративному стилю.

- Изображения и графика отображаются без искажений.

- Страница адаптивно отображается на разных устройствах (мобильные, планшеты, десктопы).

2. Содержание:

- Понятный заголовок или слоган, передающий цель и уникальное предложение компании.

- Краткое описание основных услуг или продуктов компании.

- Контактная информация (телефон, адрес, электронная почта) присутствует и актуальна.

3. Функциональность:

- Отсутствие битых ссылок и ошибок 404.

- Формы работают корректно и отправляют данные на сервер.

- Работоспособность всех интерактивных элементов (кнопки, меню, слайдеры).

- Виджеты социальных сетей и другие сторонние компоненты функционируют правильно.

- Поиск на сайте работает и возвращает соответствующие результаты.

- Корректное отображение видео или медиа-файлов.

4. Безопасность:

- Сайт использует HTTPS-протокол для обеспечения безопасного соединения.

5. SEO и производительность:

- Заголовок страницы соответствует контенту и содержит ключевые слова.

- Мета-теги (description, keywords) заполнены информативно.

- Сайт имеет адекватную скорость загрузки на разных устройствах и соединениях.

6. Социальные ссылки и отзывы:

- Ссылки на официальные страницы в социальных сетях актуальны.

- Отзывы или рекомендации клиентов, отображаются корректно.

7. Кросс-браузерная совместимость:

- Сайт корректно отображается и функционирует в популярных браузерах (Chrome, Firefox, Safari, Edge).

8. Язык и локализация:

- Проверить, что тексты на странице переведены и локализованы правильно.

## Задание 3.

|  |  |
| --- | --- |
| Postman Collection: |  |

Негативные сценарии (помечены значком «-»):

- Запрос без указания ключа.

- Измените формат координат на некорректный (например, строки вместо чисел).

- Изменить язык на неподдерживаемый.

## Задание 4.

* SQL-запросы для создания и заполнения таблиц:

-- Создание таблицы animal\_classes и заполнение ее данными

CREATE TABLE animal\_classes (

id INT PRIMARY KEY,

class VARCHAR(50)

);

INSERT INTO animal\_classes (id, class) VALUES

(1, 'кошка'),

(2, 'собака');

-- Создание таблицы animal\_info и заполнение ее данными

CREATE TABLE animal\_info (

name VARCHAR(50),

class INT,

owner VARCHAR(50)

);

INSERT INTO animal\_info (name, class, owner) VALUES

('Кити', 1, 'Ваня'),

('Мити', 2, 'Ваня'),

('Пити', 1, 'Петя');

* SQL-запросы для решения каждого из вопросов:

-- Сколько всего животных у Вани

SELECT COUNT(\*) AS total\_animals

FROM animal\_info

WHERE owner = 'Ваня';

-- Уникальные имена всех кошек отсортированные по алфавиту

SELECT DISTINCT name

FROM animal\_info

INNER JOIN animal\_classes ON animal\_info.class = animal\_classes.id

WHERE animal\_classes.class = 'кошка'

ORDER BY name;

--Найти количество животных каждого класса. Вывести количество и имя класса

SELECT animal\_classes.class, COUNT(\*) AS total\_count

FROM animal\_info

INNER JOIN animal\_classes ON animal\_info.class = animal\_classes.id

GROUP BY animal\_classes.class;