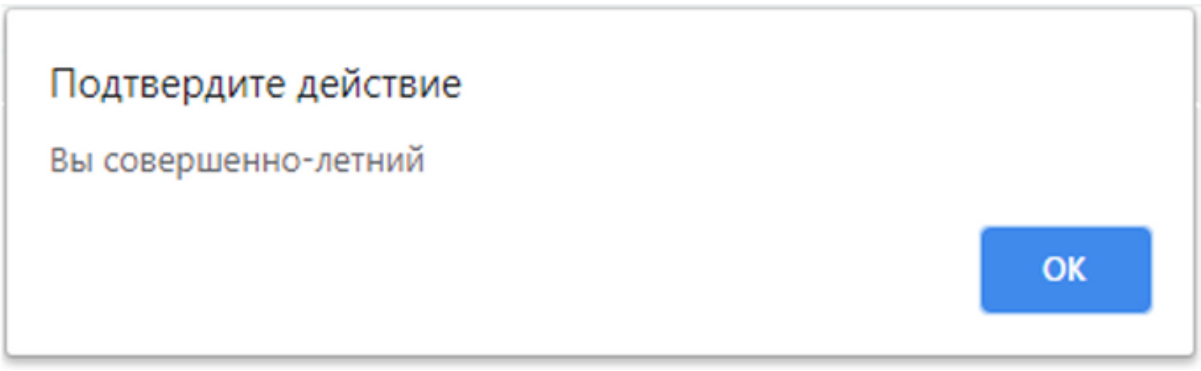
Выполнила: *Анисимова Татьяна*

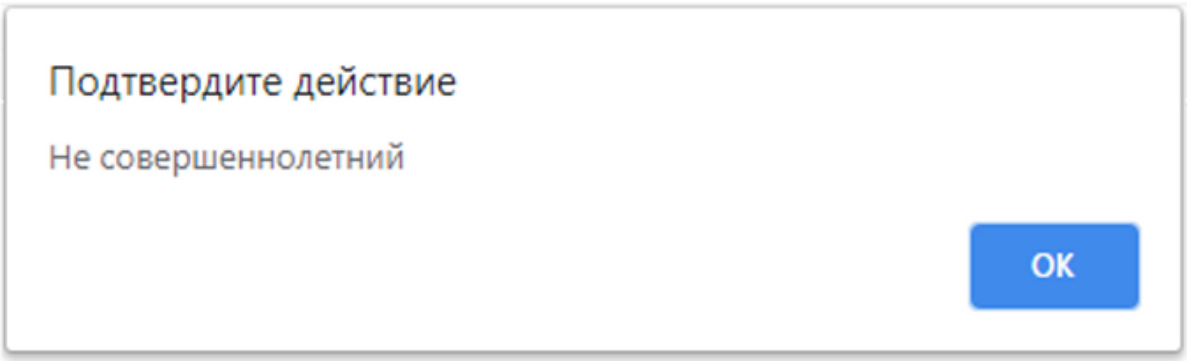
Тестовое задание для старта трудоустройства (Яндекс погода)

Задание 1.

ТЗ от заказчика: “Реализовать форму, которая по введенным данным определяет, является ли человек совершеннолетним. Приложение должно быть с архитектурой клиент-сервер.”
Предположим, что разработчик реализовал данную форму.
Начальный вид формы (в соответствии с макетом):

При нажатии на кнопку 0К появляется одно из сообщений (для вывода используется функция JS alert):





Ответьте на приведенные вопросы:

1. Какие могут возникнуть вопросы к требованиям?
2. Какие виды и типы тестирования стоит проводить? Объясните, почему.

3.Что больше подойдет для организации тестовой документации по задаче - тест-кейсы или чек-листы? Почему?

4.Какие техники тест-дизайна Вы бы использовали? Почему?

5.Какие можно выделить негативные и позитивные входные данные?

6.Какие теоретически могут быть баги? Опишите 4-6 возможных бага.

7.Есть ли баги любого типа на приведенных скриншотах? Если есть, опишите все.

Ответ должен быть прислан в формате документа, в котором есть ответы на все вопросы.

Оценивается:

1.Оформление

2.Правильность ответов на вопросы

Ответы на вопросы

1.Какие могут возникнуть вопросы к требованиям?

	Вопросы к требованиям
1	В форме “Введите свой Возраст?” кнопка “ОК” должна быть бесцветной или голубой как в последующих формах?
2	Во всех формах “Подтвердите действие”должно стоять местоимение “ВЫ” или его вообще не должно быть?
3	Какие числа вводятся: целые (18), дробные через запятую (18,5) или дробные через точку (18.5)?
4	Какое максимальное количество лет можно вводить?

2.Какие виды и типы тестирования стоит проводить? Объясните, почему.

	Виды и типы тестирования
1	Функциональное тестирование. Проверка корректности работы функциональности приложения. -*Модульное тестирование*. Проверка модуля. -*Smoke test*. Проверка приложения на старте на выполнение основных функций.

2	Нефункциональное тестирование. -*Тестирование UI*. Расположение, размер, цвет, ширина, длина элементов; возможность ввода цифр; проверка текста на орфографические, пунктуационные ошибки. -*Юзабилити-тестирование*. Проверка удобства использования, понятности для пользователей разрабатываемого приложения в контексте заданных условий. -*Тестирование с различными разрешениями экрана*. -*Тестирование кросс-браузерности или совместимости* с разными интернет-браузерами и их версиями.
3	Регрессионное тестирование. Проверка изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб-сервер или сервер приложения), для подтверждения, что всё работает, как и прежде. Регрессионными могут быть как функциональные, так и нефункциональные тесты.

3.Что больше подойдет для организации тестовой документации по задаче - тест-кейсы или чек-листы? Почему?

Тест-кейс от чек-листа отличается наличием шагов, приводящие к конкретному итогу, который описывается в атрибуте “Ожидаемый результат”. А чек-лист в свою очередь состоит из заголовка и результата проверки. В этом случае подойдет чек-лист, так как для простых полей можно один раз написать чек-лист проверок, а потом переиспользовать , лишь меняя под “свое” поле.

4.Какие техники тест-дизайна Вы бы использовали? Почему?

1	**Граничные значения.** Именно на границах чаще всего встречаются баги, поэтому тестируются значения, которые отделяют один интервал от другого.
2	**Класс эквивалентности.** Эта техника позволяет протестировать основные категории чисел, которые могут быть введены в форму, чтобы быть уверенными, что форма определяет разные типы чисел; и мы тестируем нечисловое значение, чтобы быть уверенными, что форма определяет их как не числовые.

3	**Ноль.** Ноль, пустота или состояние, равное нулю, — это отдельный класс эквивалентности, который нужно тестировать всегда.

5.Какие можно выделить негативные и позитивные входные данные?

Позитивные	Негативные
целые числа 18 - 99	целые числа 1-17
	ноль
	спецсимволы
	буквенные значения
	иероглифы
	отрицательные числа
	целые числа лишенные смысла (200 лет)
	пустая строка
	дробные числа
	римские числа

6.Какие теоретически могут быть баги? Опишите 4-6 возможных бага.

- принимаются числа меньше 18;
- принимаются буквенные значения;
- кнопка “ОК” не кликабельна;
- не совмещается с разными интернет-браузерами;
- грамматические ошибки в заголовке формы;
- при нажатии на кнопку “0К” НЕ появляется одно из сообщений.

7.Есть ли баги любого типа на приведенных скриншотах? Если есть, опишите все.

	Описание бага
1	В первой форме “Подтвердите действие” грамматическая ошибка. ФР - совершенно-летний (через тире) ОР - совершеннолетний (слитно)
2	Во второй форме “Подтвердите действие” грамматическая

	ошибка. ФР - не совершеннолетний(НЕ отдельно) ОР - несовершеннолетний(НЕ слитно)
4	В форме “Введите ваш Возраст ? ” пунктуационная ошибка. ФР - знак “?” стоит ОР - знак “?” не стоит
5	В форме “Введите ваш Возраст” грамматическая ошибка. ФР - слово “Возраст” написано с большой буквы ОР - слово “возраст” написано с маленькой буквы

Задание 2.

Составить чек-лист проверок для главной страницы сайта

Результат приложить в виде doc файла или скриншотов созданных чек-листов в любой известной вам TMS.

Чек-лист главной страницы сайта https://cloud.ru/ru

	Проверка	**Результат**
	Проверить Хедер	
1	select “Почему Cloud.ru”	
2	select “Решения”	
3	select “Сервисы”	
4	аккордеон “Цены”	
5	select “Партнеры”	
6	select “Ресурсы”	
7	аккордеон “Консультация”	
8	select “Начать работу”	
9	строка “Поиск”	
10	select смены языка	
11	аккордеон “Документация”	
12	аккордеон “Кэшбэк 200% на облачные сервисы Cloud/ru”	
13	кнопка “Начать работу”	
14	кнопка “Консультация”	
15	Проверить слайдер по партнерам	
	Блок “Сервисы”	
1	Популярное	
2	Инфраструктура	
3	Сети	
4	Хранение и резервное копирование	
5	Облако VMware	
6	Безопасность	
7	AI@ML	
8	Контейнеры	
9	Базы данных	
10	Миграция	
11	Разработка	
12	Serverless	

13	Аналитика данных	
14	Управление и администрирование	
15	Услуги	
	Блок “Решаем задачи”	
	По отраслям	
1	Ритейл	
2	IT-Разработка	
3	Транспорт и логистика	
4	Образование	
5	Медиа и реклама	
6	Энергетика	
7	Здравоохранение	
8	Государство	
	По задачам	
1	Чат-боты на Serverless	
2	Сайт в облаке	
3	1С в облаке	
4	Разработка и тестирование	
	Проверить слайд “Читать кейс”	
	Проверить кнопку “Начать работу” в блоке “Интерфейс”	
	Проверить кнопку “Больше о нас” в блоке “Наше преимущество”	
	Проверить ссылки на соцсети, телефон и почту	
	Блок “Ресурсы”	
1	Проверить слайд	
2	Проверить ссылки на отдельные статьи под слайдом	
	Проверить “Полезные ссылки”	
	Проверить поле ввода почты	
	Проверить кнопку “Подписаться”	
	Проверить поле согласия на сообщения информации и рекламы	
	Проверить футер	
1	Техническая поддержка	
2	Решения	
3	Документация	
4	Платформы	
	Проверить ссылку на cookie в всплывающем окне	
	Проверить кнопку “Понятно” в всплывающем окне	

Оцениваются:

1.Полнота проверок

2.Качество оформления и системность материалов

Задание 3.

API Яндекс.Погоды позволяет получать погодные данные в автоматизированном режиме. Объем предоставляемых данных зависит от выбранного тарифа доступа

Необходимо с помощью Postman протестировать работу метода со следующими параметрами:

-тариф Тестовый

-lon 50

-lat - 55

-язык русский
Также создать несколько негативных сценариев и убедиться, что при неправильных данных запрос возвращает соответствующий результат
P.S. Для начала работы с методами вам потребуется создать API ключ на соответствующее приложение.
Результат необходимо прикрепить в виде готовой json коллекции Postman
Оценивается:
-оформление
-правильность запросов
-использование и передача переменных между запросами (если это необходимо)
-проверка возвращаемых данных (написаны тестовые скрипты на проверку статус кода и других необходимых параметров(если это необходимо)
Выполненное задание отдельно прикрепила в виде готовой json коллекции Postman!

Задание 4.

Есть вот такие вводные данные:
table animal_classes

name	**class**	**owner**
Кити	1	Ваня
Мити	2	Ваня
Пити	1	Петя

table animal_info

id	**class**
1	кошка
2	собака

Оценивается:
-умение создавать SQL запросы
Ответь, используя запросы SQL, на следующие вопросы:
1.Сколько всего животных у Вани;
2.Уникальные имена всех кошек отсортированные по алфавиту;
3.Найти количество животных каждого класса. Вывести количество и имя класса.
Результат приложить в виде ответов на вопросы ***в формате SQL***

Выполнение задания:

```
-- Создать и заполнить таблицу table animal_classes

CREATE TABLE animal_classes (
    id          INTEGER,
    class       TEXT
);

INSERT INTO animal_classes (id, class)
VALUES
(1, 'кошка'),
(2, 'собака');

 --Создать и заполнить таблицу table animal_info

        CREATE TABLE animal_info (
          name       TEXT,
       class    INTEGER,
       owner      TEXT
);


INSERT INTO animal_info (name, class, owner)
VALUES
('Кити', 1, 'Ваня'),
('Мити', 2, 'Ваня'),
('Пити', 1, 'Петя');

                                                --Задания--

/*1.Сколько всего животных у Вани*/

SELECT COUNT (*) FROM animal_info
WHERE owner = 'Ваня';

/*2.Уникальные имена всех кошек отсортированные по алфавиту.*/

SELECT DISTINCT a.name
FROM animal_classes AS ac JOIN animal_info AS a ON ac.id = a.class
WHERE ac.class = 'кошка'
```

```
ORDER BY a. name    ASC;
```

/*3.Найти количество животных каждого класса. Вывести количество и имя класса.*/

```
SELECT COUNT (*), ac.class
FROM animal_classes AS ac JOIN animal_info AS a ON ac.id = a.class
GROUP BY ac.class
```

Также приложила результат выполненного задания в виде отдельного файла!