**Задание №1**

**Часть 1 :**

**Неясности и противоречия:**

1. **Неясно,возможно ли выбирать текущую дату**
2. **Не указано, какие символы допустимы для поля «Имя»(кириллица,латиница , иероглифы и спецсимволы)**
3. **Не указано, по какому часовому поясу ориентироваться**
4. **Не указано, как форма должна повести себя в случае бронирования более чем за 30 дней,либо каким образом ограничение в 30 дней должно соблюдаться**
5. **Не указано, как должна повести себя форма в случае, если бронь, которую планирует клиент, уже занята**

**Вопросы :**

1. **Возможна ли бронь на текущую дату ?**
2. **Какие символы допустимы для поля «Имя»(кириллица,латиница , иероглифы и спецсимволы)?**

**3) По какому часовому поясу ориентироваться?**

**4) Как форма должна повести себя в случае бронирования более чем за 30 дней,либо каким образом ограничение в 30 дней должно соблюдаться?**

**5) Как должна повести себя форма в случае, если бронь, которую планирует клиент, уже занята?**

**Часть 2 - тест-кейсы:**

**Тест-кейс 1**  
**Название:** Успешное бронирование с валидными значениями  
**ID:** TC\_001  
**Предусловие:** Форма открыта  
**Шаги:**

1. Имя: "Алексей"
2. Количество гостей: 4
3. Дата: через 5 дней
4. )Время: 18:30
5. Комментарий: "С красивым видом,пожалуйста"
6. Нажать "Забронировать"

**Ожидаемый результат:** Бронь успешна

**Тест-кейс 2**  
**Название:** Бронирование на ближайшую допустимую дату (завтра)  
**ID:** TC\_002  
**Предусловие:** Форма открыта  
**Шаги:**

1. Имя: "Петр"
2. Кол-во гостей: 2
3. Дата: завтра
4. Время: 13:00
5. Комментарий: "День рождения"
6. Нажать "Забронировать"

**Ожидаемый результат:** Бронь успешна

**Тест-кейс 3**  
**Название:** Проверка предельной длины комментария (250 символов)  
**ID:** TC\_003  
**Предусловие:** Форма открыта  
**Шаги:**

1. Имя: "Анастасия"
2. Гости: 2
3. Дата: через 10 дней
4. Время: 17:30
5. Комментарий: строка длиной 250 символов
6. Нажать "Забронировать"

**Ожидаемый результат:** Бронь успешна

**Тест-кейс 4**  
**Название:** Ввод недопустимого количества гостей (0)  
**ID:** TC\_004  
**Предусловие:** Форма открыта  
**Шаги:**

1. Имя: "Иван"
2. Гости: **0**
3. Дата: через 3 дня
4. Время: 14:00
5. Комментарий: "Проба"
6. Нажать "Забронировать"

**Ожидаемый результат:** Кнопка неактивна или сообщение об ошибке

**Тест-кейс 5**  
**Название:** Кнопка активна при пустом поле "Имя"  
**ID:** TC\_005  
**Предусловие:** Форма открыта  
**Шаги:**

1. Оставить поле "Имя" пустым
2. Гости: 2
3. Дата: через 4 дня
4. Время: 15:30
5. Комментарий: "Без имени"
6. Проверить кнопку "Забронировать"

**Ожидаемый результат:** Кнопка должна быть неактивна

**Часть 3 - Баги :**

**Баг-репорты находятся в отдельном Excel-файле(см. В архиве с данным файлом)**

**Часть 4 -  Оценка UX и улучшения :**

1. **Автоматическое отключение занятых слотов - при выборе даты скрывать занятые временные интервалы**
2. **Подсказки для пользователей в полях - например в поле «Имя» - «Введите ваше имя»**
3. **Помечать обязательные поля символом «\*»,а у кнопки «Забронировать»-подсказку,какие обязательные поля не заполнены,если таковые имеются**

**Задание 2 - SQL**

1. **SELECT \* FROM Users WHERE Status = 'active' AND Age > 25; -** запрос для выборки всех активных пользователей старше 25 лет.
2. **SELECT COUNT(\*) FROM Users WHERE Status = 'inactive';-** получить количество неактивных пользователей
3. **SELECT Name, Email FROM Users ORDER BY Age DESC;-**запрос покажет имена и email всех пользователей, отсортированных по возрасту по убыванию.
4. **Проверка каскадного удаления -** сначала найти **id** пользователей, потом проверить наличие заказов в таблице **orders .** Далее удалить запись из таблицы **users(DELETE FROM users WHERE id= 1;),** и в конце отправить запрос в **orders** с id удаленного пользователя**.Если все прошло как надо,записи из обеих страниц будут удалены**
5. **Наличие записи с нулевым или отрицательным значением Age - SELECT \* FROM users WHERE Age <0 OR Age = 0;.**
6. **Наличие записи с некорректным полем e-mail - SELECT \* FROM users WHERE Email NOT LIKE '%@%';**

**Задание 3 - API**

1. **Проверять следует следующие поля:**
2. **name - на длину, спецсимволы и пустое поле**
3. **Email - корректность и уникальность**
4. **Age - тип данных и допустимый диапазон**

**Примеры:**

1. **Невалидный email:**

**{**

**"name": "Анна",**

**"email": "annaexample.com",**

**"age": 27**

**}**

1. **Отрицательный возраст:**

**{**

**"name": "Анна",**

**"email": "anna@example.com",**

**"age": -5**

**}**

1. **Пустое поле name :**

**{**

**"name": "",**

**"email": "anna@example.com",**

**"age": 27**

**}**

1. **В запросе**
   1. **корректность токена и его наличие (Authorization)**
   2. **соответствует ли формат тела запроса(Content-Type: application/json)**

**В ответе:**

1. **правильный формат ответа(Content-Type: application/json)**
2. **техническая информация(Date, Server)**
3. **В Postman можно проводить такие типы тестирования ,как:**
   1. **Функциональное тестирования -** проверка, работает ли эндпоинт как ожидается
   2. **Тестирование валидации -** проверка неправильных/пограничных данных
   3. **Регрессионное -** автоматизация повторных проверок с использованием скриптов
   4. **Интеграционное -** проверка взаимодействия с другими API
4. 1. **201 Created -Успешное создание**
   2. **400 Bad Request-Ошибка валидации**
   3. **409 Conflict -Email уже существует**
5. **Тест-кейс 1**

**Название:** Проверка успешного создания пользователя

**ID:** TC\_\_008

**Предусловие:** Эндпоинт /api/v1/users доступен, авторизация (если требуется) выполнена

**Шаги:**

1)Отправить POST-запрос на /api/v1/users

В теле запроса передать корректные данные:

{

"name": "Анна",

"email": "anna@example.com",

"age":

}

1. Отправить запрос

**Ожидаемый результат:**  
Ответ с кодом 201 Created, в теле ответа возвращаются поля:

{

"id": <число>,

"name": "Анна",

"email": "anna@example.com",

"age": 27,

"status": "active"}