

Тестовое задание — Проверка мини-приложения, SQL и API

Задание №1 — Проверка мини-приложения бронирования столика

Часть 1: Анализ и уточнение

Неясности:

1. Не указано, можно ли бронировать на текущую дату.
2. Не уточнён часовой пояс.
3. Неясно, можно ли повторно бронировать на один слот.
4. Неизвестно, что происходит после бронирования.
5. Не указано, какие символы разрешены в имени.

Уточняющие вопросы:

1. Можно ли бронировать на текущую дату, если время ещё не наступило?
2. Какие символы допустимы в поле «Имя»?
3. Что происходит после успешного бронирования?
4. Что делать при повторной брони на один и тот же слот?
5. Как обрабатываются ноль или отрицательные значения гостей?

Часть 2: Тест-кейсы

ТС-01 — Успешная бронь□ Шаги: Иван, 4 гостя, завтра, 18:00, комментарий: День рождения□

Успешное бронирование

ТС-02 — Количество гостей = 0□ Шаги: Олег, 0 гостей, +3 дня, 14:00□ ОР: Ошибка — поле не

ТС-03 — Дата в прошлом□ Шаги: Мария, 2 гостя, вчера, 16:00□ ОР: Ошибка — нельзя выбрать дату

ТС-04 — Имя из спецсимволов□ Шаги: @#\$%^, 2 гостя□ ОР: Ошибка валидации

ТС-05 — Комментарий 250 символов□ Шаги: Антон, комментарий: 250 символов□ ОР: Успешна

Часть 3: Баг-репорты

Bug-01 — Можно выбрать текущую дату и прошедшее время□ Приоритет: P1□ Обоснование: В прошлое недопустимо

Bug-02 — Поле «Имя» ограничено 10 символами□ Приоритет: P2□ Обоснование: Недостаточно имён

Bug-03 — Кнопка активна при пустом поле «Имя»□ Приоритет: P1□ Обоснование: Обязательно игнорируется

Bug-04 — Комментарий принимает 1000+ символов□ Приоритет: P3□ Обоснование: Возможны UI/БД

Часть 4: UX и улучшения

1. Автозаполнение имени□2. Блокировка прошедшего времени□3. Уведомление после брони

Задание №2 — SQL

1. `SELECT * FROM USERS WHERE is_active = true AND age > 25;`
2. `SELECT COUNT(*) FROM USERS WHERE is_active = false;`
3. `SELECT name, email FROM USERS ORDER BY age DESC;`
4. Удаление пользователя и проверка связей в orders.
5. Баги: отрицательный возраст, дублирующийся email.

Задание №3 — API

1. Валидация: name, email, age.□ Примеры:
 - Некорректный email□- Отрицательный возраст□- Пустое имя
2. Headers: Content-Type, Authorization, CORS.
3. Типы тестов в Postman: позитивные, негативные, валидация, структура, авторизация.
4. Статусы:□ - 201 Created□ - 400 Bad Request□ - 409 Conflict□ - 401 Unauthorized
5. Тест-кейс: TC-API-01 — успешное создание пользователя.