

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ АЛТАЙСКОГО КРАЯ
КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ЯРОВСКОЙ ПОЛИТЕХНИЧЕСКИЙ ТЕХНИКУМ»

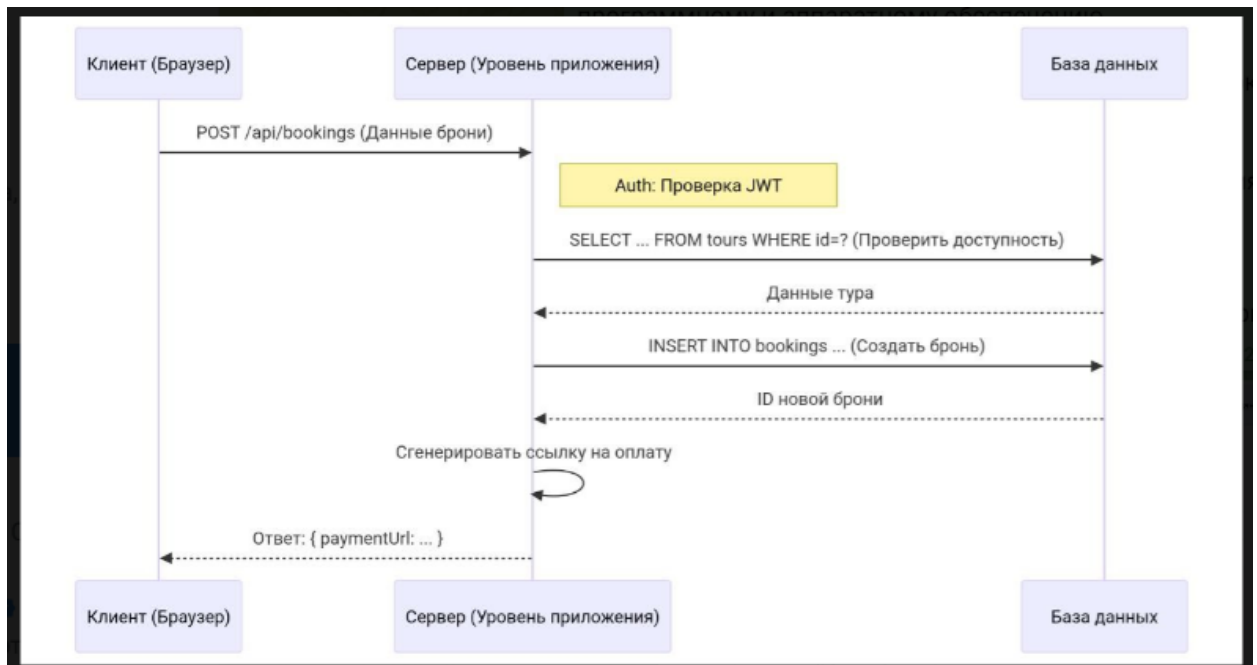
Практическая работа №3

Тема: Построение архитектуры программного средства.

Цель: Развить навыки проектирования архитектуры программного обеспечения.

Работу выполнил(а):	Крутиков С. В. , Сухих А.Н.
Работу проверил:	Олюшин В. В.

Яровое 2025



1. Уровень представления (Клиент)

- Роль: Интерфейс для пользователя. Все, что видно в браузере: кнопки, формы, картинки. Его задача — показать данные и принять команды от человека.
- Взаимодействие: Отправляет HTTP-запросы (GET, POST) на Уровень приложения (сервер) и получает от него ответы (обычно в формате JSON), чтобы отобразить результат. Пример: нажал кнопку "Найти туры" -> отправил запрос GET /api/tours?country=Turkey.

2. Уровень приложения (Сервер)

- Роль: Мозг системы. Обрабатывает логику: бронирование, расчеты, проверку прав доступа. Не имеет своего интерфейса.
- Взаимодействие:
 1. Принимает запросы от Уровня представления.
 2. Обрабатывает их (считает стоимость, проверяет логин/пароль).
 3. Формирует и отправляет запросы к БД (найти тур, сохранить бронь).
 4. Получив данные из БД, отправляет ответ обратно на Уровень представления.

3. Уровень данных (База данных)

- Роль: Надежное хранилище. Как большой цифровой склад. Хранит все: данные о турах, пользователях, бронированиях.

- Взаимодействие: Отвечает только на запросы от Уровня приложения (например, выполнить SQL-команду `SELECT * FROM tours`). Никогда не общается напрямую с клиентом.

Итог: Запрос всегда идет по цепочке Клиент -> Сервер -> База данных. Ответ возвращается в обратном порядке. Это обеспечивает безопасность (к БД нет прямого доступа извне) и возможность развивать каждый уровень независимо.