

Лабораторная работа №1-2. Typescript

Задание

Разработать приложение, состоящие из backend-части (nodejs + typescript) и frontend-части (ReactJS + typescript), использующих разделяемый набор типов.

Требования к реализации

- Работа выполняется в группе из 2 человек.
- Использовать github (добавить в отчет ссылку на репо, **в репо должны быть коммиты от каждого студента!**).
- Подготовить документацию на контракт API при помощи swagger (**добавить в отчет!**).
- Использовать микросервисную архитектуру построения backend-части приложения.
- Для реализации backend-части использовать одно из следующих решений (при желании различные решения для разных микросервисов): Express.js, Fastify, NestJS.
- Реализовать механизм авторизации на основе JWT, логику по работе с пользователями вынести в отдельный микросервис (при необходимости разделить роли пользователей).
- Frontend-часть приложения должна быть запакована в образ nginx (не использовать при сдаче работы dev-сервер).
- Для хранения данных использовать БД (PostgreSQL, MySQL или mongoDB – не более 3-4 таблиц/коллекций).
- Использовать Docker.

Варианты

1. Личный кабинет системы бронирования билетов на мероприятия.
2. Личный кабинет системы записи на прием к врачу.
3. Личный кабинет системы контроля успеваемости студентов.
4. Личный кабинет системы доставки продуктов питания.
5. Личный кабинет системы учета клиентов тренажерного зала.

Материалы

1. <https://www.typescriptlang.org/>
2. https://www.youtube.com/watch?v=J_BI_Q07XTk&t=373s
3. <https://swagger.io/>
4. <https://react.dev/>
5. <https://nodejs.org/en>
6. <https://expressjs.com/>
7. <https://fastify.dev/>
8. <https://nestjs.com/>