

СПП (часть 2)

Лабораторная работа №1

Клиент - сервер

Разработать простое приложения с рендерингом на сервере. Например, список задач со статусом их выполнения, фильтрацией по статусу и выставлением ожидаемой даты завершения, а также возможностью прикреплять файлы к каждой задаче. Сервер должен отдавать клиенту готовую разметку, отправка данных серверу должна осуществляться через отправку форм. Обязательно использование `NodeJS`, конкретные библиотеки могут отличаться. Например, подойдут `Express + EJS`.

Лабораторная работа №2

Rest API + SPA

Простое приложение, как в лабораторной работе №1, но с другой архитектурой. На сервере должен быть реализован `REST API`, на клиенте - `Single Page Application`. Обмен данных должен осуществляться путем отправки/принятия `HTTP`-запросов с данными в формате `JSON` или файлов в формате `multipart/form-data`. Обновление данных на клиенте не должно приводить к перегрузке страницы. Серверный `REST API` должен поддерживать ожидаемую семантику: правильно использовать `HTTP`-методы (`GET` для чтения данных, `POST/PUT` для изменения, `DELETE` для удаления и т.п.) и возвращать правильные коды ответов (200 в случае успешного чтения/изменения данных, 404 если ресурс не найдет и т.п.). Обязательно использование `NodeJS` на сервере. На клиенте можно использовать что угодно, `React/Angular/Vue` или вообще без библиотеки.

Лабораторная работа №3

JWT

Добавить к приложению из лабораторной №2 аутентификацию на базе `JWT`-токенов. Токен должен передаваться через `httponly cookie` на клиент и так же отправляться на сервер. При попытке прочитать/изменить данные на сервере без валидного токена, клиенту должен возвращаться 401 код. При получении кода 401 клиент должен потребовать от пользователя ввода логина/пароля. Для формирования `JWT`-токена можно использовать только пакеты `jsonwebtoken` и `bcrypt`. Логику аутентификации нужно описать в виде отдельного `middleware` той библиотеки, на которой написан сервер (например, `Express`).

Лабораторная работа №4

WebSockets

Как лабораторная работа №3, но заменить REST API на обмен данных через WebSockets .
Можно использовать библиотеку `Socket.IO` .

Лабораторная работа №5

GraphQL

Как лабораторные работы №3 и №4, но на сервере сделать API на GraphQL .