**ЛР 4 REACT и NODE.JS**

**Тема:**

1. Библиотека для создания пользовательских интерфейсов [React](https://ru.reactjs.org/);
2. Веб-фреймворк Express (Node.js/JavaScript)

**Цель работы:**

Закрепить знания и навыки по работе с библиотекой JavaScript для создания пользовательских интерфейсов [React](https://ru.reactjs.org/), фреймворком Express (Node.js).

**Cправочная информация по React:**

Документация: [**https://ru.react.js.org**](https://ru.react.js.org)**,** [Начало работы – React](https://ru.react.js.org/docs/getting-started.html)

[Руководство по React (metanit.com)](https://metanit.com/web/react/?ysclid=loiov4ig95727252426)

[Getting started with React - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)

[Быстрый старт React с примерами кода (reactdev.ru)](https://reactdev.ru/learn/)

[React Reference Overview – React](https://react.dev/reference/react)

[Учебник. React для начинающих | Microsoft Learn](https://learn.microsoft.com/ru-ru/windows/dev-environment/javascript/react-beginners-tutorial)

**Cправочная информация по Node.js/** **Express:**

<https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs>

[Node.JS | Введение и начало работы (metanit.com)](https://metanit.com/web/nodejs/1.1.php) [Node.JS | Начало работы с Express (metanit.com)](https://metanit.com/web/nodejs/4.1.php)

[Guides | Node.js (nodejs.org)](https://direct.nodejs.org/en/docs/guides)

[Полное руководство по Node.js — Изучите Node для начинающих ⚡️ Node.js с примерами кода (nodejsdev.ru)](https://nodejsdev.ru/guides/freecodecamp/?ysclid=loiwxadofi708106430)

[Полный справочник по Node.js (questu.ru)](https://questu.ru/articles/204850/?ysclid=loiwxxebvl210950948)

**Задание.**

Разработать веб-сайт с использованием Node.js(Express) для реализации сервера и [React](https://ru.reactjs.org/) для клиента.

**Постановка задачи**:

* Определить и реализовать **минимум четыре модели** в соответствии с предметной областью предыдущих ЛР (использовать встроенные валидаторы, подходящие типы данных). В качестве базы данных рекомендована MongoDB (<https://metanit.com/web/nodejs/6.1.php>, <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose> ), но можно другую по желанию.
* Реализовать механизмы авторизации/аутентификации с помощью логина/пароля и через аккаунты Google и Facebook ([Аутентификация ⚡️ Node.js с примерами кода (nodejsdev.ru)](https://nodejsdev.ru/guides/webdraftt/authentication/));
* Для авторизованного юзера реализовать CRUD (create, read, update, delete) операции и возможность поиска, сортировки записей <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose>; Использовать формы, созданные с использованием React, для добавления записей. <https://metanit.com/web/react/3.3.php>
* Для неавторизованного пользователя обеспечить только просмотр, поиск, сортировку и авторизацию;
* Обеспечить наполнение данными для демонстрации не менее 10 записей в списке товаров/услуг/объектов/клиентов.
* Подключить и использовать как минимум 2 сторонние API, какие именно - выбрать на свой вкус.
* отображать тайм зону пользователя, текущую дату, дату добавления/изменения данных в таблицы для тайм зоны пользователя и для UTC.
* Для создания всех **страниц (минимум четыре – выбрать любые из предыдущих ЛР(**среди страниц обязательно должен быть каталог, просмотр инфо об объекте из каталога**), переключение между ними с помощью панели навигации)** использовать только React.
* Создать компоненты
  + с использованием декларативных функций, стрелочных функций и классов.
  + Применить props, значения по умолчанию.
  + Использовать разработанные компоненты внутри других компонентов.
  + Добавить обработчики событий (минимум семь) – в функциональных компонентах, компонентах классах, с передачей параметров в обработчик события. <https://metanit.com/web/react/2.5.php>
  + Продемонстрировать работу со state (<https://metanit.com/web/react/2.4.php> )
  + Продемонстрировать использование любого из хуков.
* Приемлемый внешний вид. css (без <https://www.bootstrapcdn.com/>);
* Валидация форм как на стороне сервера, так и на стороне клиента;
* Ограничить использование API проекта для неавторизованных запросов.