

Итоговый проект курса по фронтенд разработке в Ерат на Заставской, 22

Задание

Разработать SPA (Single-page application) на React.js.

Обязательные требования

Начало разработки

Рекомендуется использовать `create-react-app` для начала разработки.

Запуск приложения

Приложение должно запускаться после клонирования репозитория и установки зависимостей (`npm install`) с помощью только запуска `npm start`

Управление состоянием приложения

Управление состоянием приложения (state management) должно быть выполнено с помощью Redux.

Работа с данными

Данные должны загружаться из внешних ресурсов. AJAX-запросы должны выполняться с помощью Redux Middleware. Если приложение не использует внешние данные (крестики-нолики, шахматы, todo-app), необходимо реализовать хранение данных в Local Storage/Indexed DB.

Роутинг

Приложение должно представлять несколько (3-4 или больше) экранов (страниц), переход между ними должен осуществляться без перезагрузки страницы в браузере. Для реализации переходов между экранами необходимо использовать React Router.

Поддержка браузеров

Приложение должно корректно работать в текущих стабильных версиях всех браузеров из основного списка CanIUse (исключая Opera Mini):

- IE 11+
- Edge 16+
- Firefox 57+
- Chrome 62+

- Safari 11+
- Opera 48+
- iOS Safari 11+
- Android Browser 62+
- Chrome for Android 62+

Корректность работы выражается в отсутствии ошибок в консоли браузера, возможности выполнить одни и те же действия в приложении по одним и тем же сценариям и отсутствию значительных расхождений во внешнем виде приложения (верстка не должна разъезжаться).

Разрешения экранов устройств – от 320 до 2560 по ширине.

Если используется какая-то возможность, поддерживаемая не всеми браузерами (например, промисы в IE11), для таких браузеров (и только для них) должны подключаться полифиллы или функционал должен иметь фоллбек.

Использование сторонних библиотек и инструментов

Для верстки макета разрешается использовать Bootstrap, если необходимы готовые компоненты (например, кнопки, выпадающие списки и т.д.). Не разрешается использовать Bootstrap для колоночной верстки. Вместо этого следует использовать Flexbox, Grid и т.д.

Можно использовать Sass, CSS + PostCSS.

Примеры приложений

- TODO-апп с несколькими списками TODO (экран списка TODO-списков, экран просмотра одного TODO-списка, экран редактирования подробностей одного TODO (приоритет, дата выполнения задачи и т.д.))
- Tic-tac-toe (крестики-нолики: экран ввода настроек (размер поля), экран игры, экран завершения игры)
- Прогноз погоды (можно использовать сверстаный виджет; экран добавления/редактирования списка городов, экран списка городов, экран данных по одному городу и т.д.)
- Конвертер валют (экран добавления валют, экран выбора направления конвертации, экран ввода/редактирования данных для конвертации и т.д.)
- Агрегатор новостей (в качестве источников данных можно использовать RSS New York Times)

Будет плюсом

- Покрытие кода юнит-тестами
- Покрытие приложения e2e-тестами
- Написание технического задания на разработку своего приложения (1-2 страницы)
- Публикация приложения с помощью github pages
- Добавление инструментов контроля качества кода (eslint, editorconfig и т.д.)
- Адаптивный/отзывчивый дизайн

Представление результата

Код приложения должен быть залит на github. В проекте не должно быть кода сторонних библиотек.

Защита проекта

Защита будет состоять из демонстрации работающего приложения, объяснения, как оно работает, ответов на вопросы по React и темам, которые рассматривались на занятиях.