

WebLab#3

SPA

Продолжение [WebLab#1](#) и [WebLab#2](#). Необходимо для выбранного приложения (к этому моменту должен быть готов бэкенд с REST API и макет интерфейса) разработать пользовательский интерфейс в виде одностраничного приложения (SPA).

1. Приложение должно быть разработано на JS-фреймворке, обеспечивающем компонентный подход (например, Angular, React, Vue и др.), либо на Vanilla JS с теми же требованиями.
2. Компоненты должны быть реализованы на базе одного из общепринятых архитектурных паттернов (MVC/MVP/MVVM/аналоги), запрещается использование “классического” js-подхода с неконтролируемыми подписками и модификациями DOM.
3. Весь UI должен быть действительно декомпозирован на компоненты, все повторяющиеся элементы интерфейса должны быть вынесены в качестве компонентов (панели, кнопки, поля и формы ввода).
4. Должна поддерживаться полноценная маршрутизация, принятая в Web: состояние любой страницы полностью кодируется URL и может быть из него восстановлено (например при переходе по ссылке из мессенджера).
Примечание: методика наименования URL страниц и состояний идеологически схожа с принципами REST.
5. По умолчанию - достаточно адаптивной десктопной версии (корректное отображение в диапазоне от 1280 x 768 до 4k).

Дополнительное задание #1 (Рекомендуемое)

Приложение полностью разработано на TypeScript.

Дополнительное задание #2

Поддержка jwt-авторизации (при условии ее реализации на бекенде).

Дополнительное задание #3

Разработан полноценный CI/CD у приложения: сборка docker-образов, запуск тестов (при наличии), развертывание (на локальной машине/хостинге/облаке).

Дополнительное задание #4

Приложение работает с сохранением своего состояния (частичная автономность), которое сохраняется во внутреннем хранилище (localStorage, IndexedDB). Например, сохраняется частично заполненная, но не отправленная форма и т.д.

Дополнительное задание #5

Адаптив под мобильную версию.