

## **Практическая работа 2. Продвинутые инструменты работы с клиентской частью**

В рамках этой работы нужно выполнить указанные далее задания в любом удобном вам порядке.

### **Задание 1.**

Переработать приложение из ПР 1 в виде SPA (если не было сразу реализовано в виде SPA) используя vue-cli. Для обеспечения навигации между страницами используйте vue-router.

### **Задание 2.**

Настроить цвета страниц разработанного приложения. Палитру можно сгенерировать с помощью специального сервиса, например, coolors (<https://coolors.co/>). Стилизацию нужно выполнять с помощью кастомизации используемого css-фреймворка ИЛИ собственных классов. Для создания таких классов необходимо использовать css-препроцессор (SASS, LESS и т. д.).

### **Задание 3.**

Должна быть реализована функция смены темы, т.е. замена настроенного в задании 2 набора цветов на другой набор по выбору пользователя. Можно определить допустимые темы заранее и предоставлять пользователю возможность выбора с помощью, например, простого селекта. При желании можно реализовать более глубокую стилизацию в темах.

### **Задание 4.**

Реализовать сохранение состояния страницы в LocalStorage. Т.е. при повторном открытии страницы установленные рейтинги/отзывы или другие не сохраняемые до сих пор параметры, равно как и незавершённая работа в конструкторе, должны сохраняться.

По желанию можно сохранять итоговые результаты использования конструктора (шаблоны или карточки) и использовать их при выводе каталога.

Помимо этого, нужно реализовать какую-либо функцию, использующую один из Web API (<https://developer.mozilla.org/ru/docs/Web/API>). Это, например, может быть запрос данных с другого сайта, воспроизведение видео/звука, text-to-speech и т.д.

### **Задание 5.**

Выполнить одно из следующих заданий на выбор:

- Доработать приложение так, чтобы оно соответствовало стандартам PWA (соответственно могло быть установлено на устройство пользователя)
- Переработать приложение для соответствия JamStack
- Использовать WebAssembly для реализации какой-либо функции (например, автоматического расчёта цены). Необязательно, чтобы для этой функции WebAssembly был действительно необходим.

#### **Задание 6.**

Разместить приложение на каком-либо бесплатном cdn, например CloudFlare (<https://www.cloudflare.com/ru-ru/>).

Все разработанные функции в js должны быть продокументированы в формате jsDoc.

Готовое приложение нужно показать преподавателю на паре или консультации. Когда работа принята, необходимо загрузить в Мудл архив с исходным кодом программы, а также отчёт. В отчёте нужно указать краткую формулировку задания, описать особенности конкретной реализации, если они есть, и привести скриншоты работы приложения.