# ТЕСТОВОЕ ЗАДАНИЕ на позицию

# Junior Frontend разработчик

## **Реализация SPA**

В данном задании необходимо реализовать приложение для поиска youtube-видео по ключевым словам, а также сохранение запросов поиска.

Реализовать интерфейс необходимо по макетам без использования сторонних ui-библиотек и css-фреймворков.

**Требования**

1. Реализовать страницу авторизации пользователя с формой входа: логин, пароль. Для аутентификации можно использовать json-файл со списком пользователей. После входа, необходимо сгенерировать и сохранить токен. Полученный токен использовать для авторизации - проверка наличия токена в localStorage, если токена нет, значит пользователь не авторизован (видит только страницу авторизации).
2. После авторизации пользователь должен попадать на главную страницу сервиса, на которой расположена строка поиска. Необходимо реализовать поиск youtube-видео, при помощи Youtube API. Результаты поиска вывести на той же странице в виде карточек видео (подобно тому, как оно реализовано в youtube). Список видео может принимать два вида: список (одно видео в строку с описанием) и карточки (несколько видео в строку). **По умолчанию необходимо вывести 12 видео в виде списка.**
3. Также, в приложении должна быть реализована страница “Сохраненные запросы”, на которой выводится список сохраненных запросов. Запросы этого списка можно редактировать или выполнить.

Чтобы выполнить запрос, необходимо нажать кнопку “Выполнить”, после чего происходит переход на главную страницу, на которой отображаются результаты запроса, при этом, **количество выводимых видео равно количеству, указанному при сохранении запроса** (см. ниже, Максимальное количество)

Для сохранения запроса необходимо добавить в строке поиска на главной странице иконку “Сохранить поиск”, при нажатии на которую открывается форма.

В форме заранее заполнено значение из строки поиска в неизменяемом поле “Запрос” и отображается пустое поле “Название”, а также дополнительные пустые поля: **Сортировать по** (выпадающий список, доступные значения взять из API YouTube), **Максимальное количество** (числовое значение от 0 до 50).

Пользователь должен ввести название запроса, и, после этого, может сохранить запрос, остальные поля - опциональные, т.е. могут быть не заполнены.

Редактирование запроса происходит по нажатию на кнопку “Редактировать”. После этого открывается форма редактирования. Для редактирования использовать тот же компонент, что и для сохранения запроса. **Обратите внимание,** что при редактировании запроса можно изменять все поля формы, включая сам запрос.

1. Реализовать выход пользователя из сервиса.
2. После повторного входа пользователя сохраненные запросы должны отображаться. Если пользователь - другой, то отображать запросы соответствующие для него (другого). Для хранения можно использовать localStorage браузера. Либо другие методы на усмотрение исполнителя.
3. Интерфейс должен корректно отображаться в последних версиях следующих браузеров: Safari, Chrome, Mozilla, Opera. А также в мобильных: Safari, Chrome.
4. Интерфейс должен адаптироваться под разрешения устройств в соответствии с макетами.

**Рекомендуется**

Использование методологии БЭМ.

Использование библиотеки axios для запросов.

**Ссылка на макет:** [**https://tinyurl.com/sibdev-front**](https://tinyurl.com/sibdev-front)

Рекомендуемое время выполнения тестового задания: 3–4 дня

**Полезное**

1. YouTube API: <https://developers.google.com/youtube/v3/docs/>
2. Axios: <https://github.com/axios/axios>
3. БЭМ: <https://en.bem.info/>
4. Управление состоянием приложения:
   1. Redux: [https://redux.js.org/](https://redux.js.org/api/api-reference)
   2. Mobx: <https://mobx.js.org/README.html>
   3. Vuex: <https://vuex.vuejs.org/>
5. Маршрутизация:
   1. React Router: <https://reacttraining.com/react-router/>
   2. Vue Router: <https://router.vuejs.org/>