**Тестовое задание на должность frontend developer (Vue.js)**

Требуется средствами Vue.js реализовать следующее SPA-приложение:

**Приложение состоит из 4х страниц:**

1. **Страница “Авторизации”** (страница имитирующая авторизацию). Пользователю предоставлена форма из 4-х полей: Имя (текстовое поле), Email(тестовое поле), Пароль (засекреченное текстовое поле), файл ( Выбрать фотографию пользователя).

Валидация для полей:

* Имя: Не менее 3-х и не более 15-ти символов
* Email: введенный email должен быть в формате example@host.domen
* Пароль: Не менее 8-ми символов, латиница, должны присутствовать цифры, первый символ в верхнем регистре.
* Все поля должны быть обязательными.

При нажатии на кнопку “Войти” данные проверяются и в случае если все корректно сохраняются в локальное хранилище, иначе не валидные поля получают красную обводку и выводиться под этим полем сообщение с ошибкой.

1. **Страница “Главная”**. После “Авторизации” пользователь попадает на главную страницу.

* В шапке этой страницы располагается: аватарка пользователя с именем при нажатии на этот блок попадаем на страницу пользователя и поле для поиска статьи по имени.
* На этой странице выводятся карточки (статьи), в которой находиться дата добавления, фотография, тема статьи, контент не более 25 символов и если контента больше показать “...”, автора статьи (Имя автора).
* При нажатии на карточку пользователь попадает на страницу статьи.
* Поиск происходит в момент ввода текста, НЕ по нажатию на кнопку найти (ее не должно быть).
* Сделать сортировку по дате добавления и алфавиту.

1. **Страница “Статьи”**. На этой странице выводим всю информацию о статье, и кнопка с возможностью редактирования статьи, при ее нажатии всплывает попап с полям в которые вписана вся информация и ее можно редактировать. При нажатии на кнопку сохранить информация сохраняется.

1. **Страница “Профиля”**. На этой странице надо вывести всю информацию о пользователе с возможностью ее редактирования. Редактирование происходит так же как со статьей в попапе. Также на этой странице есть возможность добавления новой статьи, при нажатии на соответствующую кнопку появляется попап с полями для добавления. Все поля обязательные. При нажатии на кнопку добавить, статья сохраняется в хранилище в JSON формате, дата добавления создается автоматически.

* **Требования по функционалу:**

Шапка на странице авторизации скрыта.

Сделать защиту роутов, чтобы неавторизованный пользователь не мог попасть на другие страницы.

При перезагрузки странице данные не удаляются.

Дизайн приложения не важен, но желательно чтобы выглядело более менее красиво.

(не обязательно) Сделать на 2-х языках RU/ENG

* **Технические требования:**

Для сборки приложения использовать Vue-CLI.

Хранилище - Vuex.

Роуты - Vue-Router.

Верстка без библиотек.

Для стилизации использовать sass/scss и все его фитчи.

После выполнения прислать:

1. Ссылку на репозиторий с исходным кодом.
2. Ссылку на загруженный сайт на git-page.

*Тестовое задание рассчитано на 1-2 дня*

[*https://developer.mozilla.org/ru/docs/Learn/Common\_questions/Using\_Github\_pages*](https://developer.mozilla.org/ru/docs/Learn/Common_questions/Using_Github_pages)

echo "# vue-project" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/sasha-is-permi/vue-project.git

git push -u origin main

**…or push an existing repository from the command line**

git remote add origin https://github.com/sasha-is-permi/vue-project.git

git branch -M main

git push -u origin main

[*https://netology-code.github.io/guides/github-pages/*](https://netology-code.github.io/guides/github-pages/)

npm run build

cd dist

git init

git add -A

git commit -m 'deploy'

git push