

## Описание тестового задания “Сервис по изучению алфавита”

Необходимо реализовать веб-приложение - сервис по изучению алфавита, состоящий из двух страниц:

/) Индексная страница, в которой расположена кнопка “Начать”, по нажатию которой осуществляется переход на стр. 2.

/challenge) На данной странице отображается 3 блока: выбранная буква из алфавита с картинкой и подпись под ней, тестовое слово, алфавит. Общий вид представлен на скриншоте 1.

лок отображения выбранной буквы.

лок с тестовым заданием. Отображается слово (одно из трех для данной буквы, выбирается случайным образом). При клике по букве необходимо вывести ответ, правильно ли была выбрана буква (см. скриншот 2). При правильно выбранной букве, помимо отображения плашки-индикатора, необходимо выделить её в слове отличным от других букв цветом. Если буква выбрана неправильно, то она никак не выделяется; выводится плашка красного цвета с соответствующим комментарием (желательно скрывать по таймауту).

лок отображения алфавита. При клике на букву должно обновляться содержимое блоков 2.a и 2.b. Выбранная пользователем буква должна быть выделена - например, подчёркиванием

Аа



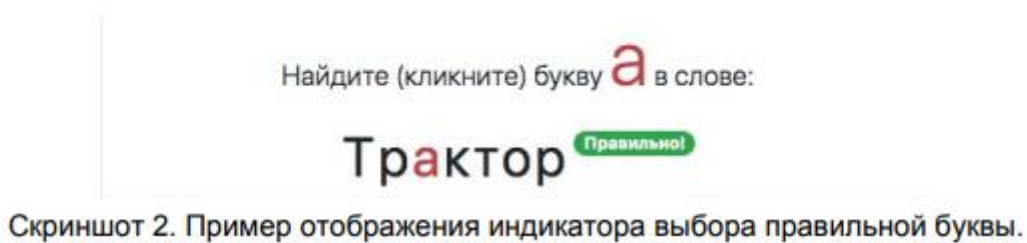
Арбуз

Найдите (кликните) букву **а** в слове:

Трактор

Аа Бб Вв Гг Дд Ее Ёё Жж Зз Ии Йй Кк  
Лл Мм Нн Оо Пп Рр Сс Тт Уу Фф Хх Цц  
Чч Шш Щщ Ъъ Ыы Ьь Ээ Юю Яя

Скриншот 1. Общий вид страницы /challenge



Скриншот 2. Пример отображения индикатора выбора правильной буквы.

## Требования к реализации и оформлению

Реализация должна быть выполнена с использованием фреймворка Vue JS 3. Верстка желательно адаптивная, можно использовать любой CSS Kit/библиотеку компонентов (PrimeVue, Vuetify...).

Алфавит и сопутствующие данные хранятся в LocalStorage. Различное отображение гласных и согласных букв не обязательно, кроме того, не обязательно задавать алфавит целиком - достаточно 3-5 букв. Для каждой буквы должен храниться список из не менее 3-х слов, используемых в задании на поиск. В случае, если localStorage клиента пуст, его необходимо заполнить данными по умолчанию (способ хранения этих данных - на усмотрение программиста). Картинки необходимо хранить и загружать из каталога assets.

Приложение должно быть выполнено в виде SPA. В Url на стр. 2 всегда отображается выбранная буква алфавита, например, в виде /challenge/, где - выбранная буква. Обновление страницы или переход по Url подобного вида должен работать корректно - т.е. отображать состояние страницы, соответствующее выбранной букве.

Тестирование работы приложения будет выполнено в последней стабильной версии браузера Chrome ( $\geq 106$ ). При этом разметка не должна "ехать", в консоль не должны падать ошибки. При оценке исходного кода будет учитываться умение использовать возможности TypeScript, владение средствами реактивного программирования (если используются), разбиение проекта на подсистемы (компоненты). Плюсом будет следование Style Guides Ангуляра с использованием линтинга исходного кода.

Исходный код приложения должен быть размещен в публичном репозитории исходных кодов (github, gitlab, bitbucket, sourceforge, ...) и дополнен инструкцией по установке и запуску. Желательно указать общее количество затраченного времени на выполнение задания.