# Лабораторная работа №6

В данной лабораторной работе вам предлагается сделать свой аналог сервиса Emoji Kitchen.

<https://www.google.com/search?q=emoji+kitchen+google>



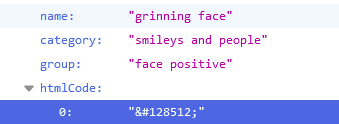
Предлагается сделать два варианта на выбор.

**Вариант 1**: Вы делаете приложение полностью по шагам, описанным ниже.

**Вариант 2**: Вы доделываете частично готовое приложение, которое реализовано мною.

## Вариант 1.

1. Напишите функцию, которая сделает при помощи fetch GET запрос по адресу <https://emojihub.yurace.pro/api/all> . Этот URL возвращает список всех возможных эмоджи. Вызовите эту функцию в onMount App компонента. Поместите ответ в реактивную переменную emojis.
2. Создайте новый компонент Emoji Данный компонент будет отвечать за отрисовку смайла. В качестве входного параметра он принимает единичный эмоджи. Реализуйте это при помощи defineProps. В разметке данного компонента выведите при помощи интерполяции свойство htmlCode[0] (оно содержит код эмоджи)



1. Проверьте работоспособность этого компонента. Подключите его в App.vue, добавив импорт, и после этого выведите какой-либо emoji из массива emojis.
2. Создайте компонент EmojiList. Он предназначен для вывода списка emoji. Входной параметр - массив emojis. Отрисуйте в разметке этого компонента каждый emoji при помощи директивы v-for.

Пользовательское событие в данном компоненте - selectEmoji. Оно происходит когда пользователь кликает по какому-либо из emoji.

1. Отрисуйте в App компоненте EmojiList дважды. В первом EmojiList пользователь выбирает первый смайл, а втором, соответственно, второй. Для хранения выбранных смайлов заведите реактивную переменную. Например, const selectedSmiles = ref([null, null]).

Далее в обработчиках события selectEmoji необходимо менять элемент selectedSmiles с соответствующим индексом.

1. Создайте компонент EmojiMixer. Он принимает два входных параметра - firstEmoji и secondEmoji. Создайте computed свойство, которое будет получать смиксованный emoji. Его код следующий:

const mixedEmoji= computed(() => {

if (props.firstEmoji && props.secondEmoji) {

const [unicodeFirst, unicodeSecond] = [props.firstEmoji.unicode[0],props.secondEmoji.unicode[0]].map(u => u.replace('U+', '').toLowerCase());

return `https://emk.vercel.app/s/${unicodeFirst}\_${unicodeSecond}?size=96`

}

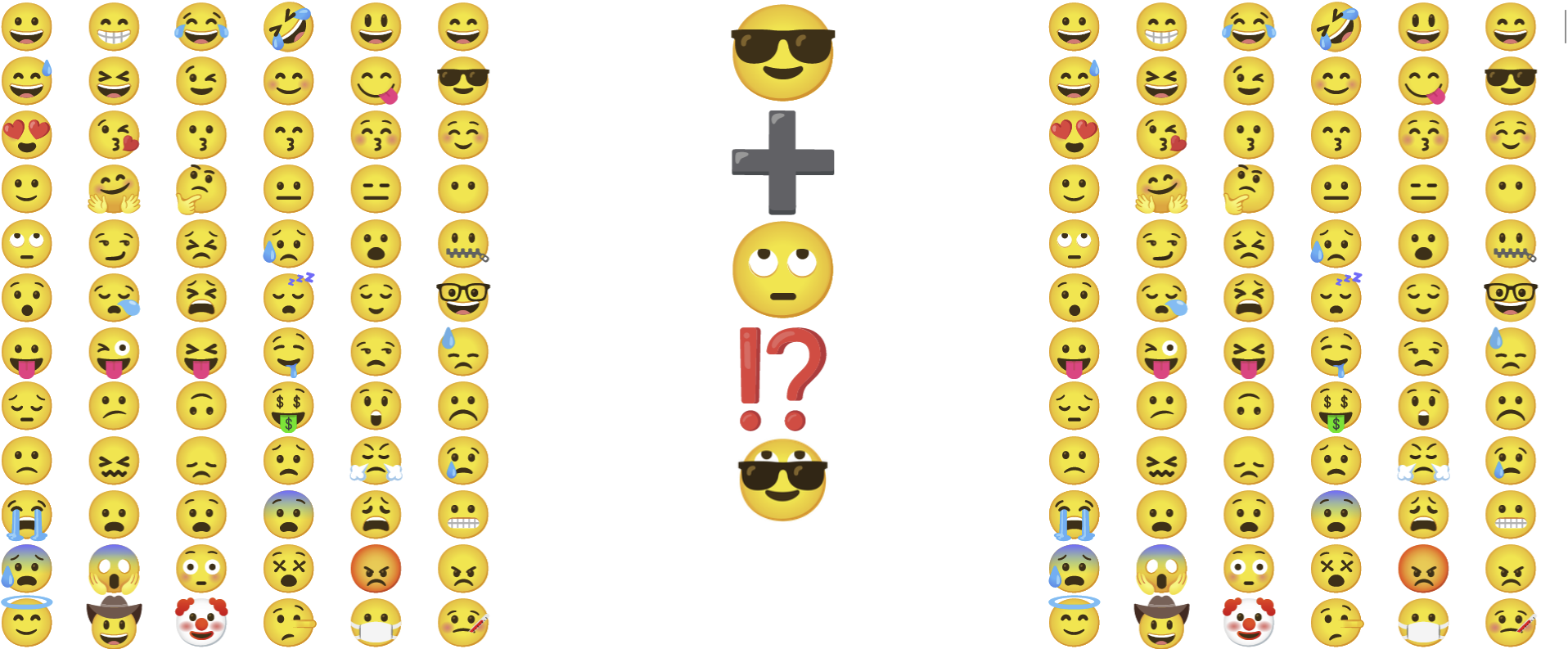
})

Смиксованное emoji мы получаем при помощи следующего API: <https://emk.vercel.app/>

1. В разметке EmojiMixer выведите два компонента эмоджи, соответствующие первой и второй эмоджи, а также картинку итогового эмоджи. Вы получите примерно следующую разметку:

<Emoji :emoji=”props.firstEmoji” /> + <Emoji :emoji=”props.secondEmoji” /> = <img :src=”mixedEmoji” />

1. Выведите компонент EmojiMixer в App компоненте. Передайте в него в качестве параметров элементы массива selectedEmoji.
2. В итоге получится что-то такое:



## Вариант 2

Скачайте шаблон с [Гугл Диска](https://drive.google.com/drive/folders/1YHCuakXkRKJBNvmwBqACJFZ0e3RYSsPY?usp=drive_link). Не забудьте выполнить **npm i**

1. Реализуйте компонент *Emoji*. Входным параметром он принимает объект *emoji*. В шаблоне компонента выведите свойство *htmlCode[0]* входного параметра *emoji*.
2. Модифицируйте компонент EmojiList. Добавьте ему пользовательское событие *selectEmoji.* Запускайте данное событие по нажатию на какое-либо *Emoji.* В качестве полезной нагрузки события передайте объект смайла, по которому был произведён клик.
3. Подключите компонент EmojiList в EmojiGame. Добавьте его в шаблон сверху и снизу относительно EmojiMixer. Передайте массив emojis в EmojiList. Добавьте функцию handleSelectEmoji в качестве обработчика события selectEmoji. Передайте в обработчик объект события и индекс листа на котором произошло событие. В обработчике модифицируйте массив selectedEmojis, установив значение элемента по индексу.