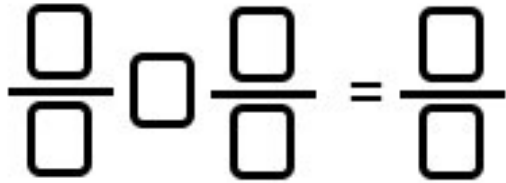


Необходимо реализовать одностраничное приложение с несколькими секциями, используя фреймворк VueJS 2.x.

Каждой секции должен соответствовать свой url.

Отдельные секции должны быть реализованы в виде подгружаемых модулей. Т.е. подгрузка кода/ресурсов, относящихся только к одной из секций, должна происходить после перехода на нее.

Задание №1 калькулятор дробей



ADD FRACTION

Необходимо реализовать калькулятор обыкновенных дробей. Слева от знака «=» расположено выражение. Минимальный набор операторов +, -, *, /, скобки будут плюсом, справа от знака равно находится результат в виде несократимой обыкновенной дроби. При нажатии на ссылку add fraction в выражении слева должен добавиться один знак и еще одна дробь. При любом изменении данных выражение должно пересчитываться. Необходимо предусмотреть всевозможные ошибочные варианты ввода.

Задание №2 Взаимодействие с сервером по протоколу websocket

Имеется веб сервер, подключение к которому происходит по протоколу websocket. На сервер посылаются запросы. Каждый запрос содержит в себе свой номер. Сервер высылает в ответ на каждый запрос номер запроса и некоторый ответ.

Пример:

Запрос «1, тело запроса» ответ «1, тело ответа»

Запрос «2, тело запроса» ответ «2, тело ответа»

Особо стоит отметить, что как на серверной стороне, так и на клиентской протокол websocket предусматривает асинхронную обработку. Т.е. сначала клиент может послать 5 запросов, а потом получить 5 ответов.

Проблема: сервер может ответить не в том порядке, в котором были сделаны запросы. Каждый ответ сервера требует уникальных действий на стороне клиента. На какой-то из запросов сервер вообще может не ответить, необходима проверка (возможность указать таймаут - опционально)

Адрес сервера: `ws://echo.websocket.org/`

Задача: реализовать асинхронную обработку ответов сервера.

Далее описание того, что необходимо реализовать.

На странице выведен список из 10 различных комментариев.

Рядом с каждым словом есть кнопка «удалить».

При нажатии на эту кнопку формируется запрос на сервер.

В начале каждого запроса указывается его номер.

(т.к. задача тестовая, то кроме номера ничего указывать не надо).

Сервер ответ сервера так же содержит номер запроса.

(т.е. задача тестовая, то кроме номера запроса в ответе ничего нет).

При получении корректного ответа от сервера на запрос с определенным номером, нужный комментарий удаляется из списка.

Пример развития событий:

Пользователь видит список комментариев

- Тестовый коммент 1 (удалить)
- Это шедевр (удалить)
- Это прекрасно (удалить)
- Лучшее, что я видел (удалить)
- Два чая этому автору (удалить)

Пользователь нажимает удалить комментарий «Это шедевр».

Далее составляется запрос, содержащий только номер 1.

Запрос отправляется на сервер. Сервер возвращает 1.

Комментарий «Это шедевр» должен удалиться.

Пользователь нажимает удалить комментарий «Два чая этому автору»

Далее составляется запрос, содержащий только номер 2.

Запрос отправляется на сервер. Сервер не ответил. Ничего не происходит

Пользователь нажимает удалить комментарий «Два чая этому автору»

Далее составляется запрос, содержащий только номер 3.

Запрос отправляется на сервер. Сервер возвращает 3.

Комментарий «Два чая этому автору» должен удалиться.

Дополнительно по всем секциям:

Особое внимание будет уделено стилистике и расширяемости кода.

По каждой секции необходимо указывать список используемых библиотек/ресурсов. Обосновать свой выбор используемых алгоритмов и структур данных.