Тестовое задание на вакансию frontend разработчик

Средствами Vue.js реализуйте небольшое SPA приложение для заметок.

Каждая заметка имеет название и список задач (todo list), далее - Todo. Каждый пункт Todo состоит из чекбокса и относящейся к нему текстовой подписи.

Приложение состоит всего из 2х страниц. (Приложение состоит из одной страницы – роутер подключен)

На главной странице отображается список всех заметок. Для каждой заметки отображается заголовок и Todo, сокращенный до нескольких пунктов, без возможности отмечать. Действия на главной:

* перейти к созданию новой заметки
* перейти к изменению
* удалить (необходимо подтверждение) – диалоговое окно есть – но удаляется все сразу же без подтверждения

Страница изменения заметки позволяет определенную заметку отредактировать, отметить пункты Todo, а после сохранить изменения. Действия с заметкой:

* сохранить изменения
* отменить редактирование (необходимо подтверждение) – работает без подтверждения
* удалить (необходимо подтверждение)
* отменить внесенное изменение
* повторить отмененное изменение Действия с пунктами Todo:
  + добавить (не сохраняется после обновления)
  + удалить (удаление возможно, однако после окончания редактирования – оно не сохраняется)
  + отредактировать текст
  + отметить как выполненный – чек бокс работает при редактировании, но не сохраняется после отмены редакции

Требования к функционалу:

* Все действия на сайте должны происходить без перезагрузки страницы.
* Подтверждение действий (удалить заметку) выполняется с помощью диалогового окна. – окно и подтверждение есть, но задача удаляется сразу же
* Интерфейс должен отвечать требованиям usability.
* После перезагрузки страницы состояние списка заметок должно сохраняться.
* Можно пренебречь несоответствием редактирования текста с помощью кнопок отменить/повторить и аналогичным действиям с помощью комбинации клавиш (Ctrl+Z, etc.).

Технические требования:

* Диалоговые окна должны быть реализованы без использования "alert", "prompt" и "confirm". – есть вариант с сonfirm (рабочий) – а так реализованы диалоговое окна без вышеуказанных способов
* В качестве языка разработки допускается использовать JavaScript или TypeScript.
* В качестве сборщика, используйте Webpack.
* Верстка должна быть выполнена без использования UI библиотек (например Vuetify).
* Адаптивность не обязательна, но приветствуется. Проект неполноценный (не готов – поэтому не использовал адаптивность, только max-min-width)
* Логика приложения должна быть разбита на разумное количество самодостаточных Vue-компонентов.

Особое внимание стоит обратить на следующие моменты:

* Код должен быть написан понятно и аккуратно, с соблюдением табуляции и прочих элементов написания, без лишних элементов и функций, не имеющих отношения к функционалу тестового задания, снабжен понятными комментариями.
* Читабельность и наличие элементарной архитектуры.
* Чистота и оформление кода — не менее важный фактор. Код должен быть написан в едином стиле (желательно в рекомендуемом для конкретного языка). Также к чистоте относятся отсутствие копипаста и дублирования логики.

Тестовое задание должно быть представлено в следующем виде:

* Ссылка на публичный репозиторий (GitHub, BitBucket, GitLab) с исходным кодом.
* Ссылка на сайт для тестирования функционала. Или Dockerfile и docker-compose.yaml, позволяющие развернуть локально командой docker-compose up работоспособную копию сайта.