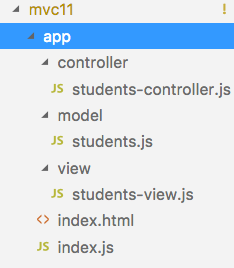
ДОМАШНЕЕ ЗАДАНИЕ. МОДУЛЬ 11

Если по какой-то причине (например, не хватает памяти на компьютере/ноутбуке) не удалось поставить VirtualBox + Ubuntu, то ставьте всё (гит, ноджс) на свою операционную систему и выполняйте в ней)

шаблон MVC имеет множество спорных реализаций и даже опытнейшие программисты могут использовать и трактовать его не верно. Воспринимайте в нашем случае MVC, как способ организации и хранения кода при котором Контроллер принимает запросы пользователя и отправляет в Модель, которая работает с данными и содержит основную логику приложения. На основании того, что модель отправит контроллеру, он сформирует Вид (Представление) и отобразит его пользователю. Вид - это не обязательно графическое представление, это может быть просто вывод в консоль или отданный пользователю файл.

1. Создайте папку (каталог, директорию) с любым названием, она будет хранить наш мини-MVC проект. Откройте ТЕРМИНАЛ или git-bash и дальше всё делайте через него. Создайте папку **app**, в которой создайте 3 папки: **model, view, controller**. Как создавать папки командами в терминале можно глянуть здесь <https://lifehacker.ru/terminal-commands/>

Перемещаясь в терминале по папкам, создайте в модели файл **students.js**, в контроллере **students-controller.js**, а в представлении **students-view.js**. Дополнительно в папке **app** создайте файл **index.html** и **index.js**. Конечный вид иерархии проекта будет таким:



Код файлов вы найдёте по ссылке

<https://jsfiddle.net/prozuyg4/>

Возьмите этот код и поместите в созданные файлы. Для запуска проекта не хватает подключить скрипты в правильной последовательности в файле **index.html**

2. Зарегистрируйтесь на сайте **github**, если ещё не сделали этого ранее. Создайте там репозиторий.

Перейдите в терминале в папку, в которой содержится **app** и выполните команду **git init**, чтобы проинициализировать репозиторий. Добавьте всё содержимое с помощью команды **git add** и сделайте 1-й коммит с текстом "initial commit". Подключите репозиторий с гитхаба в качестве удалённого и запушьте туда свой проект с компьютера.

3. Посмотрите на код проекта внимательно. Изучите, как он работает. Нам необходимо сделать пару изменений:

1. Сейчас список студентов находится внизу, а форма добавления вверху. Исправьте код так, чтобы было наоборот
2. Добавьте кнопку "REMOVE STUDENT", которая бы удаляла студента, если он есть в списке. Если студента нет в списке, то сейчас выбрасывается исключение и программа крашится. Оберните вызов, связанный с удалением студента в блок try/catch с выводом сообщения об ошибке в alert()
3. Добавьте проверку на дубликаты, чтобы нельзя было добавить несколько одинаковых студентов. Если есть совпадения, то будет alert(), что такой студент уже есть в списке
4. Добавьте функционал для замены одного студента на другого

Все крупные изменения коммитьте и не забывайте добавлять описание коммита. Чтобы было видно историю ваших коммитов.

ЗАДАНИЯ СО ЗВЁЗДОЧКОЙ

4. Возможно, что вы видите где-то несоблюдение принципов SOLID. Постарайтесь переделать код, например разбить код модели на несколько классов или код других папок, чтобы принципы соблюдались

**Задание присылать в виде ссылки на репозиторий гитхаб**