**Задание**

Вам нужно создать страницу, на которой Секретарша сможет создавать карточки, описывающие запланированные визиты к врачам.

На странице должны присутствовать:

1. Header (шапка) страницы:
   * в левом верхнем углу - логотип. Можно взять любой
   * в правом углу - Кнопка "Вход". После успешной авторизации она должна меняться на Кнопку "Создать визит".
2. Под Header - форма для фильтрации визитов. В этой форме должно быть 3 поля:
   * поиск по заголовку/содержимому визита
   * поиск по статусу (Open/Done) (визит прошел или еще нет)
   * срочность визита (High, Normal, Low)
3. Под формой фильтров - список созданных визитов.

**Командная работа**

На данном проекте все студенты разделены на группы по три человека. Студенты могут распределять между собой задачи самостоятельно. При сдаче проекта необходимо в файле Readme.md указать, кто выполнял какую часть задания.

**Технические требования**

* При первом посещении пользователем страницы, на доске должна быть надпись No items have been added. Эта же надпись должна быть, если у пользователя нету ни одной добавленной карточки (например, он их все удалил).
* По клику на кнопку **Вход** появляется модальное окно, в котором пользователь вводит свой email и пароль. Если он верный - пользователю на странице выводится список ранее созданных визитов.
* По клику на кнопку **Создать визит** появляется модальное окно, в котором можно создать новую карточку.
* Для создания классов нужно использовать синтаксис class из ES6.
* Для выполнения AJAX запросов можно использовать fetch или axios.
* После выполнения любых AJAX запросов, страница не должна перезагружаться. При добавлении/удалении карточки и других подобных операциях, с сервера **не должен** заново загружаться весь список карточек. Необходимо использовать данные из ответа сервера и Javascript для обновления информации на странице.
* При обновлении страницы или ее закрытии, ранее добавленные заметки не должны пропадать.
* Желательно разделить проект на модули с помощью ES6 modules.

**Модальное окно "Создать визит"**

В модальном окне должны присутствовать:

* Выпадающий список (select) с выбором врача. В зависимости от выбранного врача, под этим выпадающим списком будут появляться поля, которые нужно дозаполнить для визита к этому врачу.
* В выпадающем списке должно быть три опции - **Кардиолог**, **Стоматолог**, **Терапевт**.
* После выбора доктора из выпадающего списка, под ним должны появиться поля для записи к этому доктору. Несколько полей являются одинаковыми для всех трех докторов:
  + цель визита
  + краткое описание визита
  + выпадающее поле - срочность (обычная, приоритетная, неотложная)
  + ФИО
* Также, каждый из докторов имеет свои уникальные поля для заполнения. Если выбрана опция **Кардиолог**, дополнительно появляются следующие поля для ввода информации:
  + обычное давление
  + индекс массы тела
  + перенесенные заболевания сердечно-сосудистой системы
  + возраст
* Если выбрана опция **Стоматолог**, дополнительно необходимо заполнить:
  + дата последнего посещения
* Если выбрана опция Терапевт, дополнительно необходимо заполнить:
  + возраст
* Кнопка Создать. При клике на кнопку отправляется AJAX запрос на соответствующий адрес, и если в ответе пришла информация о новосозданной карточке - создается карточка в Доске визитов на странице, модальное окно закрывается.
* Кнопка Закрыть - закрывает модальное окно без сохранения информации и создания карточки. По клику на область вне модального окна - модальное окно также закрывается.
* Все поля ввода, независимо от выбранной опции, кроме поля для дополнительных комментариев - обязательны для ввода данных. Валидацию на корректность данных делать не обязательно.

**Карточка, описывающая визит**

Карточка, которая создается по клику, появляется на доске задач. Это должно выглядеть примерно так:

В ней должны присутствовать:

* ФИО, которые были введены при создании карточки
* Врач, к которому человек записан на прием
* Кнопка Показать больше. По клику на нее карточка расширяется, и появляется остальная информация, которая была введена при создании визита
* Кнопка Редактировать. При нажатии на нее, вместо текстового содержимого карточки появляется форма, где можно отредактировать введенные поля. Такая же, как в модальном окне при создании карточки
* Иконка с крестиком в верхнем правом углу, при нажатии на которую карточка будет удалена

**Фильтры визитов**

Фильтр карточек (поле input для ввода текста поиска по заголовку или описанию визита, выпадающий список по статусу, выпадающий список по приоритету) вам нужно делать на фронт-енде - то есть при изменении value любого элемента формы (выбран пункт в выпадающем списке, было введено что-то в input) вы фильтруете список ранее полученных с сервера карточек, и отображаете на экране новую информацию.

По принципу работы система должна быть похожа на фильтры в интернет-магазинах (например, слева [здесь](https://rozetka.com.ua/notebooks/c80004/)).

**Классы**

В JavaScript коде обязательно должны быть такие классы:

* класс Modal (всплывающее окно);
* класс Visit (описывающий общие для всех визитов к любому врачу поля и методы);
* дочерние классы VisitDentist, VisitCardiologist, VisitTherapist;

Методы и свойства каждого класса вам нужно продумать самостоятельно. При необходимости, вы можете добавлять также другие классы.

**Требования к реализации**

Дизайн может быть любой, но должен быть.

**AJAX часть**

Вся необходимая документация по взаимодействию с AJAX сервером находится [здесь](https://ajax.test-danit.com/api-pages/cards.html).

**Необязательное задание продвинутой сложности**

* При создании карточки визита, выполнять валидацию корректности введенных данных. Правила для валидации можно придумать самостоятельно (например, обычное давление должно являться числом, и находиться в диапазоне от 50 до 160)
* Добавить возможность пользователю перемещать карточки по доске методом Drag&Drop. Такие манипуляции с карточкой не влияют на месторасположение остальных карточек. После перетягивания карточки, не нужно "запоминать" ее новое местоположение. При перезагружке страницы она может вернуться на свое первоначальное место.