

# Введение

Вы закончили базовые части курсов по JavaScript и вёрстке. Вы узнали, как работать с HTML, задавать внешний вид интерфейсам с помощью CSS, а также добавлять интерактивность элементам интерфейса через JavaScript. Ваших знаний достаточно для того, чтобы создавать полноценные приложения в браузере, которые могут решать реальные проблемы бизнеса.

Вы успешно прошли этот сложный путь, так что вполне можете гордиться своими трудолюбием и упорством! А впереди вас ждет не менее полезный и интересный этап - дипломная работа.

До сих пор вы выполняли задания на отработку тех знаний, которые вы получали в рамках определённого модуля. Теперь же пришло время объединить весь пройденный материал в едином большом задании, где вы сможете понять, как они применяются совместно. Этим заданием как раз и является данная дипломная работа.

Некоторые считают диплом формальностью: "Я получил знания, а бумажка мне не важна". Но диплом - это не просто свидетельство. Выполненная работа многое даст вам как специалисту:

- ◆ Опыт разработки приложения, максимально приближенного к тому, что бы вы могли делать, устроившись на работу.
- ◆ Уверенность в своих силах, так как вы будете иметь на руках рабочее приложение, которое не стыдно будет добавить в портфолио.
- ◆ Обратную связь от преподавателей, которая поможет вам не просто выполнить задание, а сделать это правильно. Ведь существует множество способов реализовать одно и то же, и не всегда очевидно, как сделать оптимальный выбор.



**Так что предлагаем вам собраться и сделать это!  
Будет очень полезно и интересно.**

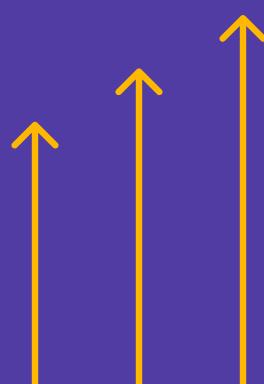
# Тема вашего проекта

## Система управления контактными данными клиентов.



Перед вами подробное описание вашего будущего проекта, который будет отличным дополнением для портфолио. Здесь есть всё, что вам нужно, чтобы справиться с поставленной задачей:

- ◆ [Описание задачи](#)
- ◆ [Техническое задание](#)
- ◆ [Описание экранов приложения](#)
  - ◆ [Таблица со списком клиентов](#)
  - ◆ [Сортировка](#)
  - ◆ [Поиск](#)
  - ◆ [Отображение контактов клиента](#)
  - ◆ [Действия над клиентами](#)
  - ◆ [Формы создания или редактирования данных о клиентах](#)
  - ◆ [Контакты клиента](#)
  - ◆ [Кнопки](#)
- ◆ [Инструкция по использованию инфраструктуры](#)
- ◆ [Технические требования](#)
- ◆ [Улучшения и дополнения к заданию](#)
- ◆ [Формат сдачи задания](#)
- ◆ [Критерии оценки](#)
- ◆ [Рекомендации к выполнению](#)
- ◆ [Полезные ссылки и материалы](#)



**Последовательно изучите каждую часть.**

Периодически обращайтесь к актуальным для вас разделам в этом документе.

# Описание задачи

Вы работаете frontend-разработчиком в компании Skillbus, которая занимается офлайн-курсами. До недавнего времени компания была небольшая: всего 1 аудитория, 3-4 параллельно идущих курса и небольшой поток учеников. Но недавно всё изменилось: вы получили инвестиции, арендовали ещё 5 помещений под аудитории, а количество курсов увеличилось в несколько раз. Учеников теперь стало столько, что надеяться на тетрадки и Excel-таблицы продажников уже не представляется возможным.

Поэтому к вам в отдел разработки пришёл руководитель отдела продаж Николай, чтобы понять, что делать дальше.

В отделе разработки с Николаем вели переговоры Татьяна. Она является вашим руководителем, а также фактически исполняет обязанности менеджера по проектам.



**Николай**

Руководитель отдела продаж



**Татьяна**

Менеджер проектов



После обсуждения стала очевидна необходимость программы, в которой можно будет работать с контактной информацией всех учеников, то есть ваших клиентов.

Постепенно стоит превратить эту программу в полноценную CRM-систему, но есть понимание, что внедрение сложного ПО в вашем случае займёт слишком много времени. Поэтому решено было сначала запустить хотя бы базовый функционал, а затем дорабатывать его.

**Разумеется, затяжная разработка подобной системы - дело не из лёгких.**

Поэтому ваш отдел будет расширяться, а вы должны будете создать сложную CRM-систему, а в перспективе вы сможете возглавить команду разработки этой системы. Выглядит как отличная возможность для развития: вы перейдёте из простой вёрстки и несложных скриптов, которые требовались для сайта вашей компании, в полноценную frontend-разработку.

Татьяна совместно с Николаем разработала набор требований к системе на момент её запуска. В ней должны быть следующие функции:

- ◆ Просмотр списка клиентов в виде таблицы
- ◆ Добавление нового клиента
- ◆ Изменение информации о существующем клиенте (ФИО и контактная информация)

На данный момент уже разработана backend-часть проекта, а также есть утверждённый дизайн. Вы, как frontend-разработчик, должны будете создать интерфейсную часть проекта и передать её Татьяне.

Татьяна сказала вам следующее:

Татьяна

На данном этапе продажникам важно иметь возможность добавлять и просматривать контактную информацию о клиентах, то есть иметь что-то вроде телефонной книги. Для этого достаточно отобразить таблицу со всеми клиентами, а также предоставить формы для создания и изменения клиентов. Для базового функционала этого будет достаточно.  
Само собой, система будет развиваться, для этого мы пока разрабатываем roadmap. Так что нужно предусмотреть то, что программа со временем будет дорабатываться и видоизменяться.



12:48

Требования к проекту Татьяна разделила на 2 части.

- 1 Первая часть - [техническое задание](#). В нём описан функционал, который обязательно должен быть на момент запуска.
- 2 Вторая часть описана в разделе "[Улучшения и дополнения](#)". Там находятся пожелания, которые необязательно должны быть, но значительно повысят ценность итогового продукта.

# Техническое задание

Разработать web-интерфейс для CRM системы, в которой должны быть следующие возможности:

- Просмотр списка людей в виде таблицы
- Добавление нового клиента
- Изменение информации о существующем клиенте

Каждый контакт представляет из себя следующий набор данных:

- Имя
- Фамилия
- Отчество
- Массив объектов с контактными данными, где каждый объект содержит:
  - Тип контакта (телефон, email, VK и т.п.)
  - Значение контакта (номер телефона, адрес email, ссылка на страницу в VK и т.п.)

Интерфейс представляет из себя единственную страницу, на которой располагается таблица клиентов, кнопка для добавления нового клиента, а также шапка с логотипом компании и строкой поиска клиентов.

The screenshot shows a web-based CRM application. At the top left is a purple circular logo with the letters 'skb.'. To its right is a search bar with the placeholder text 'Введите запрос' (Enter query). Below the header is a light gray navigation bar containing the word 'Клиенты' (Clients) in bold. The main content area features a table titled 'Клиенты' (Clients) with the following columns: ID, Фамилия Имя Отчество (Family Name First Name Middle Name), Дата и время создания (Creation Date and Time), Последние изменения (Last Change), Контакты (Contacts), and Действия (Actions). The table lists five client entries with their respective details and contact icons. At the bottom of the table is a blue button labeled 'Добавить клиента' (Add Client) with a plus sign icon.

ID	Фамилия Имя Отчество	Дата и время создания	Последние изменения	Контакты	Действия
123455	Скворцов Денис Юрьевич	21.02.2021 12:41	21.02.2021 14:50		<a href="#">Изменить</a> <a href="#">Удалить</a>
123456	Куприянов Арсений Валерьевич	28.01.2021 16:40	02.02.2021 17:01		<a href="#">Изменить</a> <a href="#">Удалить</a>
123457	Константинопольская Людмила Александровна	20.01.2021 14:01	21.01.2021 14:59		<a href="#">Изменить</a> <a href="#">Удалить</a>
123458	Дмитриевский Олег Алексеевич	14.01.2021 16:50	18.01.2021 16:55		<a href="#">Изменить</a> <a href="#">Удалить</a>
123459	Александрова Татьяна Павловна	11.01.2021 12:45	11.01.2021 14:11		<a href="#">Изменить</a> <a href="#">Удалить</a>

## Описание экранов приложения

### Таблица со списком клиентов

Таблица со списком людей имеет следующие колонки:

- ID
- ФИО (Фамилия Имя Отчество через пробел)
- Дата и время создания

- Дата и время последнего изменения
- Контакты
- Действия (кнопки)
  - Изменить клиента
  - Удалить клиента

ID ↑	Фамилия Имя Отчество ↓ A-Я	Дата и время создания ↓	Последние изменения ↓	Контакты	Действия
123455	Скворцов Денис Юрьевич	21.02.2021 12:41	21.02.2021 14:50	+6	Изменить  Удалить
123456	Куприянов Арсений Валерьевич	28.01.2021 16:40	02.02.2021 17:01		Изменить  Удалить
123457	Константинопольская Людмила Александровна	20.01.2021 14:01	21.01.2021 14:59		Изменить  Удалить
123458	Дмитриевский Олег Алексеевич	14.01.2021 16:50	18.01.2021 16:55	 +7 (985) 443-00-00	Изменить  Удалить
123459	Александрова Татьяна Павловна	11.01.2021 12:45	11.01.2021 14:11		Изменить  Удалить

Таблица должна строиться на основе данных из API. При первичной загрузке нужно отображать индикатор загрузки, пока таблица с данными не будет построена.

1	4
2	3
3	2
4	1

## Сортировка

Все заголовки колонок, кроме контактов и действий, можно нажать, чтобы установить сортировку по соответствующему полю. Первое нажатие устанавливает сортировку по возрастанию, повторное - по убыванию. Сортировка должна происходить из JavaScript, то есть API передаёт данные в неотсортированном виде.

По умолчанию должна быть установлена сортировка по возрастанию по ID.

Состояние сортировки должно корректно отображаться в виде соответствующих иконок около заголовков.

## Поиск

При вводе текста в поле для поиска данные для таблицы должны быть перезапрошены из API с введённым поисковым запросом. При этом запрос должен отправляться только по прошествии 300мс с момента последнего ввода символа в поле (то есть нужно ожидать, пока пользователь не завершит ввод поискового запроса).

# Отображение контактов клиента

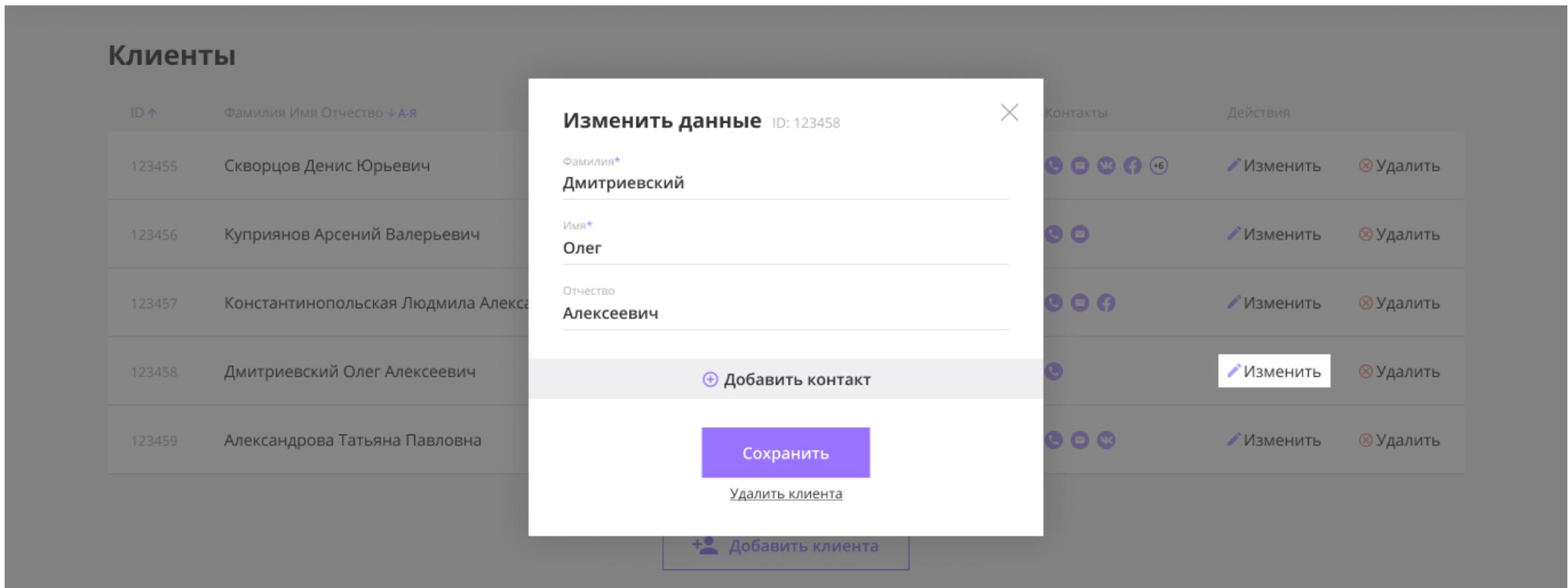
В колонке с контактами для контактов VK, Facebook, телефона и email должны отображаться соответствующие иконки. Все остальные виды контактов отображаются с одинаковыми иконками с человечком.



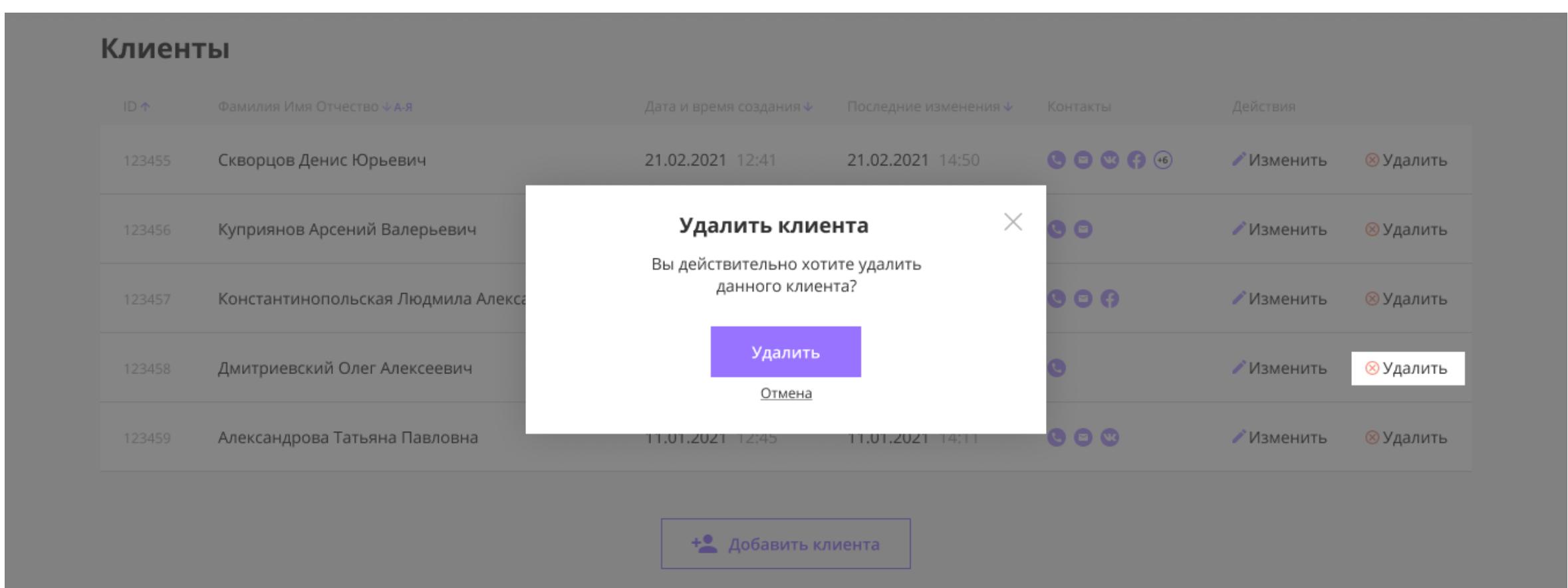
При наведении указателя на контакт должна показываться всплывающая подсказка с типом и значением этого контакта в формате "Тип": значение" (Например: "Email": abc@abc.ru", "Телефон": +71234567890", "Twitter": @xyz").

## Действия над клиентами

При нажатии на кнопку "Изменить" должно появиться модальное окно с формой изменения клиента. Подробное описание поведения формы приведено ниже.



При нажатии на кнопку "Удалить" должно появиться модальное окно с подтверждением действия. Если пользователь подтверждает удаление, то человек должен быть удалён из списка. Также на сервер с API должен посыпаться запрос на удаление.



# Формы создания или редактирования данных о клиентах

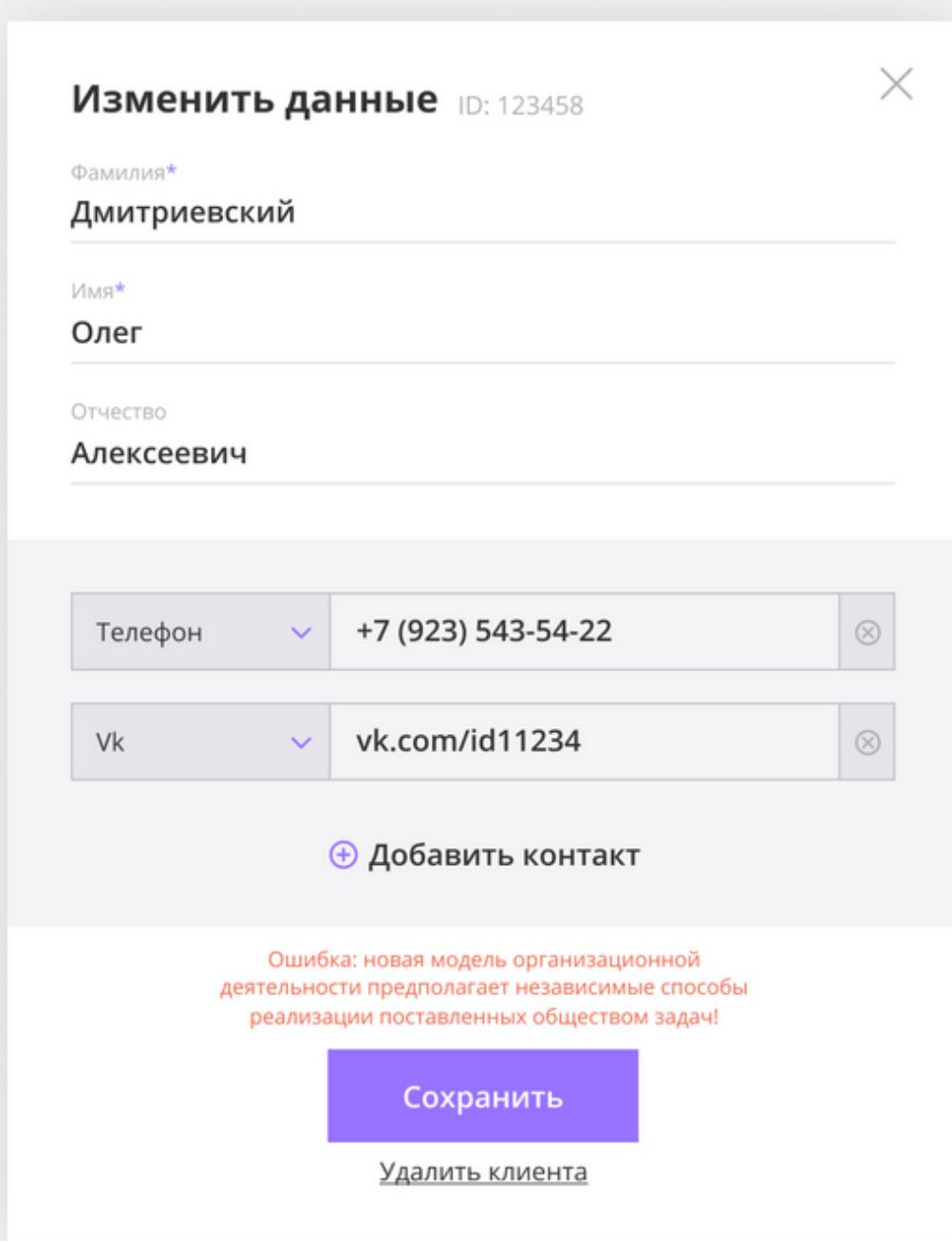
Форма создания клиента должна открываться в виде модального окна по нажатию на кнопку "Добавить клиента", находящуюся под таблицей. Форма редактирования клиента должна открываться по нажатию на кнопку "Изменить" в таблице клиентов.

Форма создания клиента открывается сразу с незаполненными полями.

При изменении клиента перед открытием формы из API должны быть запрошены свежие данные клиента, только после получения этих данных должна открыться форма. При этом форма должна быть заполнена соответствующими данными клиента.

## Контакты клиента

В блоке контактов нужно предусмотреть возможность добавления до 10 контактов включительно. Для этого под добавленными контактами должна быть кнопка "Добавить контакт". Если у клиента уже добавлено 10 контактов, кнопка не должна отображаться.



The screenshot shows a modal window titled "Изменить данные" (Edit data) with ID: 123458. It contains fields for姓氏 (Family name) with value "Дмитриевский", 名前 (First name) with value "Олег", Отчество (Middle name) with value "Алексеевич", and two contact entries. The first entry is a phone number "+7 (923) 543-54-22" and the second is a VK link "vk.com/id11234". Below these is a button "+ Добавить контакт" (Add contact). A note at the bottom states: "Ошибка: новая модель организационной деятельности предполагает независимые способы реализации поставленных обществом задач!" (Error: the new organizational model assumes independent ways of implementing tasks set by society!). At the bottom are "Сохранить" (Save) and "Удалить клиента" (Delete client) buttons.

Тип контакта выбирается из выпадающего списка с выбором одного из значений:

- Телефон
- Email
- Facebook
- VK
- Другое

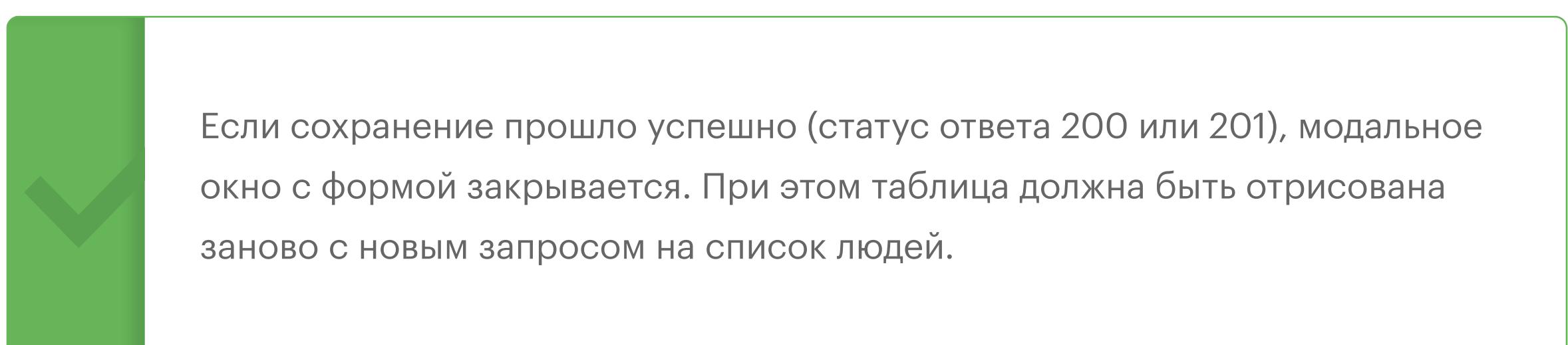
Каждый контакт можно удалить из списка по нажатию на крестик справа от него.

## Кнопки

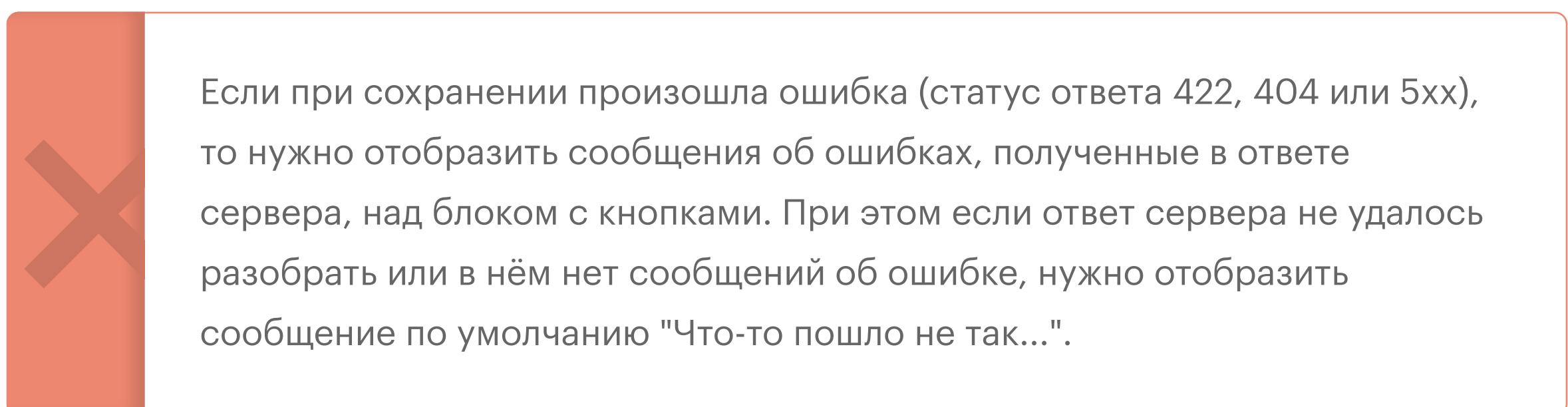
Под формой должна располагаться кнопка "Сохранить". Кнопка "Удалить клиента" добавляется, если это форма редактирования существующего в базе данных человека.

По нажатию на кнопку удаления клиент должен быть удалён из таблицы, а на сервер API должен быть отправлен запрос на удаление клиента. Модальное окно с формой должно закрыться

По нажатию на кнопку сохранения изменения должны быть отправлены на сервер с использованием метода создания или изменения существующего клиента. Далее в зависимости от ответа сервера:

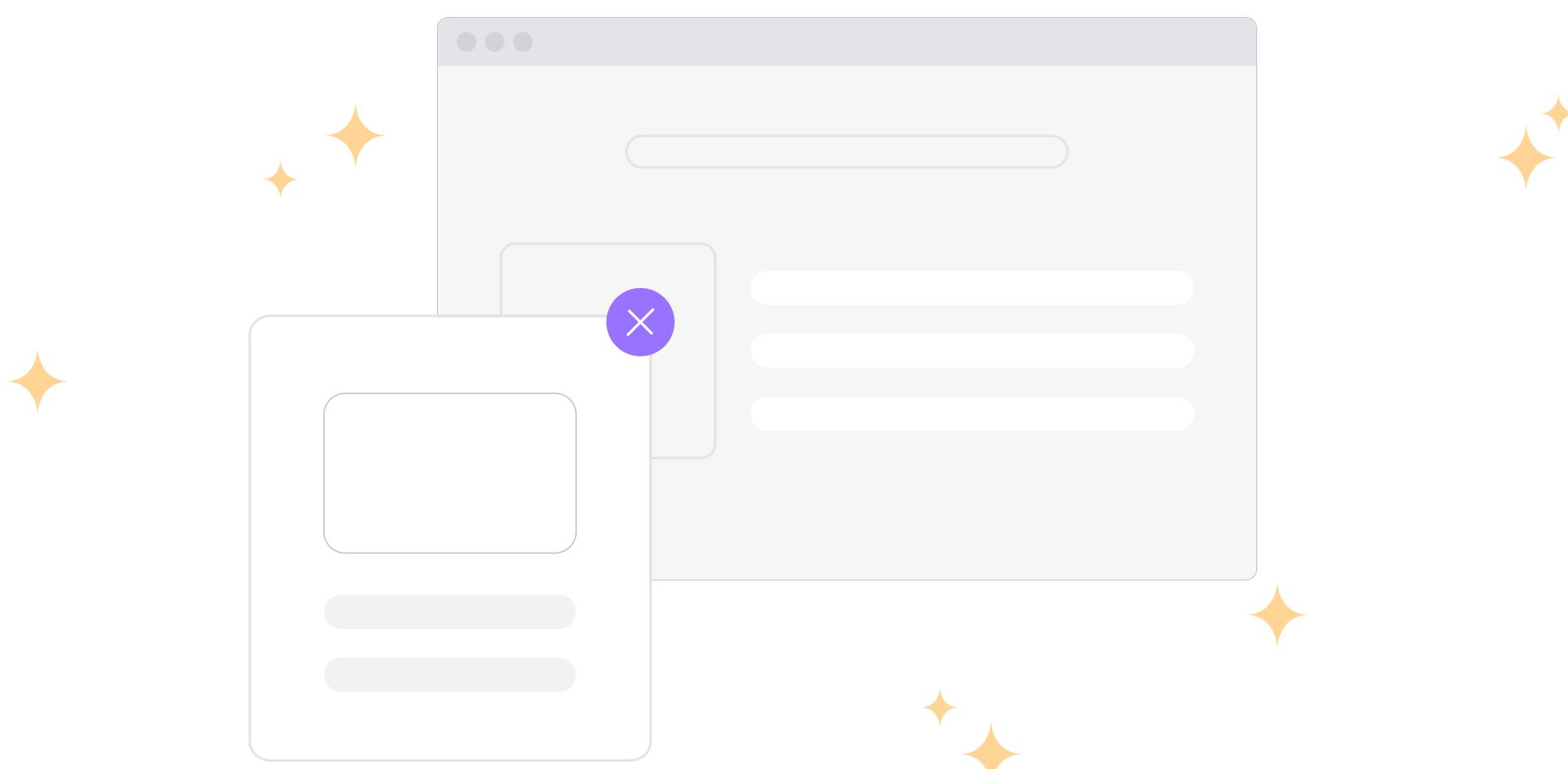


Если сохранение прошло успешно (статус ответа 200 или 201), модальное окно с формой закрывается. При этом таблица должна быть отрисована заново с новым запросом на список людей.



Если при сохранении произошла ошибка (статус ответа 422, 404 или 5xx), то нужно отобразить сообщения об ошибках, полученные в ответе сервера, над блоком с кнопками. При этом если ответ сервера не удалось разобрать или в нём нет сообщений об ошибке, нужно отобразить сообщение по умолчанию "Что-то пошло не так...".

По нажатию на кнопку с крестиком, кнопку отмены или фон модального окна модальное окно должно закрыться.



# Инструкция по использованию инфраструктуры

Дизайн доступен в виде макета в Figma по ссылке:

<https://www.figma.com/file/rcta5K2ySOhnskjG1D82jL/CRM?node-id=121%3A485>

## Технические требования

Выполненную работу необходимо предоставить в виде отдельного репозитория.

Репозиторий должен включать в себя страницу **index.html** в качестве точки входа в приложения.

## Улучшения и дополнения к заданию

Если вы уже выполнили задание по ТЗ, вы можете попробовать свои силы в дополнительных заданиях. Они расположены по сложности — от более простых к более сложным. По просьбе Татьяны их нужно сделать без привлечения дополнительных ресурсов, поэтому мы не привлекаем дизайнера. Вам нужно будет продумать внешний вид и реализовать задачи самостоятельно.  

### 1. Анимация открытия модального окна

Сложность · 1

Татьяна предложила вам добавить анимацию открытия модального окна. Сходу она смогла вам предложить следующие варианты реализации:

- ◆ Выход из прозрачности
- ◆ Смещение окна (`transition: translateY` в CSS) с формой сверху-вниз
- ◆ Масштабирование (`transform: scale` в CSS)

Вы понимаете, что выход из прозрачности должен быть в любом случае, в остальном же вы можете сделать всё, что считаете красивым.

## 2. Ссылка на карточку клиента

Сложность • 2

Николай пришёл к вам с просьбой добавить отдельный вид страницы с карточкой клиента. Ему важно, чтобы была возможность обмениваться ссылкой на конкретного клиента, так как сейчас каждый раз приходится искать нужный контакт в таблице.

**Николай сказал:**

Николай

Внешний вид страницы вторичен, но вся информация о клиенте должна отображаться в удобном для чтения виде. То, как будет выглядеть ссылка, его тоже не интересует. Важно только то, что при открытии ссылки пользователь сразу видел карточку клиента, а саму ссылку можно было скопировать из соответствующей строки в таблице.



15:40

**Вы обсудили функционал с Татьяной и нашли неплохой вариант реализации задачи:**

Татьяна

Думаю, можно связать hash-часть пути страницы с открытием модального окна редактирования контакта. На форме отображается всё необходимое, сама форма уже свёрстана, так что это можно назвать "обойтись малой кровью". Здесь вам нужно найти информацию о том, как работать с hash-частью пути страницы, а также о событии hashchange. Это нужно, чтобы изменять hash страницы при открытии модального окна, а также чтобы сразу открывать нужное модальное окно при первичной загрузке страницы с указанной hash-частью.



16:04

## 3. Валидация формы перед отправкой на сервер

Сложность • 2

Для браузерных приложений почти всегда делают базовую валидацию данных перед отправкой на сервер. Это позволяет чуть быстрее показать ошибки пользователю, а также немного уменьшает нагрузку на сервер.

**Добавьте предварительную проверку данных с помощью JavaScript:**

- ◆ Имя и фамилия обязательны для заполнения
- ◆ Каждый добавленный контакт должен быть полностью заполнен

При неуспешной проверке нужно добавить над кнопкой соответствующие сообщения об ошибках, а также подсветить ошибочные поля в соответствии с макетом. Отправка ошибочных данных на сервер происходить не должна.

Поле должно перестать подсвечиваться как ошибочное, как только пользователь начал изменять в нём значение.

## 4. Индикация загрузки

Сложность • 3

Бывает, что из-за загруженности серверов, на которых находится API, запросы с вашего интерфейса довольно долго обрабатываются (бывают по 3 или даже по 5 секунд) 😬

Поэтому к вам пришла Татьяна и попросила добавить индикацию загрузки, так как пользователи часто не понимают, что при нажатии на кнопки сохранения или при открытии списка клиентов на самом деле что-то происходит.

Татьяна

**Индикация загрузки нужна для 3-х случаев:**

- ◆ Загрузка таблицы со списком клиентов при открытии страницы, после ввода значения в поле для поиска и т.п. Возможный способ реализации: отображать большой спиннер с полупрозрачным слоем поверх области с таблицей
- ◆ Загрузка данных клиента при открытии формы редактирования. Возможные способы реализации: выводить пустую форму с заблокированными атрибутом `disabled` полями или спиннер на кнопке редактирования с открытием модального окна только после загрузки клиента.
- ◆ Отправка формы создания или редактирования клиента. Здесь важно предотвратить изменения в полях для ввода, пока идёт взаимодействие с сервером. Это можно сделать как минимум 2-мя способами: поставить на поля атрибуты `disabled` или вывести поверх формы абсолютно позиционированный полупрозрачный слой.



16:21

Способ реализации остаётся на ваше усмотрение, выше указаны лишь варианты, которые вам сходу предложила Татьяна. В вашей компании инициатива приветствуется, так что если вы можете предложить вариант получше - вы можете без проблем реализовать его.

## 5. Поиск с автодополнением

Сложность · 4

Николай пришёл к вам с просьбой реализовать поиск с автодополнением, если это не отвлекает вас от других рабочих задач.

Николай

Это более сложнореализуемый, но при этом более привычный для пользователей способ работы с поиском. Наличие автодополнения означает, что список найденных элементов должен отображаться непосредственно под полем для ввода, без изменения самой таблицы. Похожим образом реализованы поисковые системы Google, Яндекс и т.п.

В таком случае имеет смысл при выборе элемента в списке каким-то образом подсветить соответствующий контакт в таблице и, при необходимости, прокрутить страницу до выделенного контакта, чтобы пользователь смог его увидеть.

Сами элементы можно, как и раньше, получить из API списка элементов, а в каждой строке показывать "Имя Фамилия" найденного контакта.



17:14

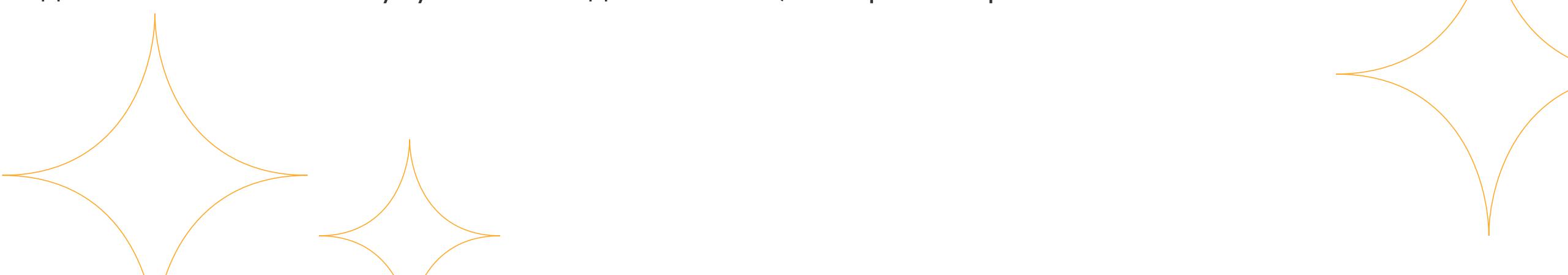
Очень важно не забыть добавить возможность выбора с помощью клавиш стрелок вверх/вниз и Enter на клавиатуре.

## Формат сдачи задания

Исходный код выполненного задания должен быть выложен на GtiLab.

HTML-файл, который необходимо открыть для работы с интерфейсом, должен быть назван **index.html** и находиться в корне репозитория.

В корень репозитория также должен быть добавлен текстовый файл README.md, в котором могут содержаться различные пояснения для проверяющих преподавателей, которые вы считаете важным передать. Также здесь должен быть добавлен список тех улучшений и дополнений, которые вы реализовали.



# Критерии оценки

Преподаватель проверяет работу по следующим критериям:

- ◆ Соответствие разработанного интерфейса требованиям в техническом задании.
- ◆ Понятность написанного кода. Сюда же можно отнести корректное именование переменных, констант и функций, которое должно соответствовать тому, что в них содержится.
- ◆ Единый стиль написания кода, в том числе:
  - ◆ нет смешанного именования (camelCase, snake\_case);
  - ◆ везде используются одинаковые отступы (2 пробела, 4 пробела или 1 табуляция);
  - ◆ единый стиль объявления функций (function declaration или function expression);
  - ◆ единый стиль работы с массивами (цикл for of или метод forEach).

**Вы можете делать проект в соответствии с кодстайлом Skillbox, а можете практиковать собственный стиль. Главное - его единообразие.**

- ◆ Правильность использования элементов HTML.
- ◆ Правильность использования CSS стилей.
- ◆ Уместность использования средств языка JavaScript. Сюда могут относиться следующие **обязательные моменты** (но не только они):
  - ◆ для переменных используется let;
  - ◆ для констант const;
  - ◆ предпочтение отдается const всегда, когда это возможно;
  - ◆ изоляция области видимости;

**И рекомендации:**

- ◆ дублирующиеся участки кода вынесены в функции;
- ◆ правильное использование методов массива (forEach, filter, find, reduce);
- ◆ осознанная работа с объектами как со ссылочным типом;
- ◆ разделение логики и представления (это значит, что в работе вы должны отделять отрисовку интерфейса и работу с сервером).

## Работа возвращается на доработку сразу, если:

- ◆ отсутствует или не работает базовая функциональность, указанная в техническом задании;
- ◆ имеются грубые ошибки в организации или структуре проекта;
- ◆ преподаватель не может запустить проект локально.

## Рекомендации к выполнению

- ▣ Внимательно изучите документацию. Убедитесь, что вам всё понятно. Это первый и очень важный шаг в начале работы над любым проектом. Если у вас возникают вопросы, фиксируйте их для себя и обязательно задавайте своему куратору дипломного проекта. В реальных проектах уточнять требования у вашего заказчика - это совершенно нормально. Можно даже сказать, что это часть вашей работы.
  - ▣ Не пытайтесь делать всё и сразу, разделите работы по вёрстке и по реализации логики программы. Так вы сможете сфокусировать своё внимание на чём-то одном, что значительно упрощает работу. Каждый вид работ тоже можно разделить на подзадачи, вполне вероятно вам так будет проще и удобнее работать.
  - ▣ Составьте план работы по датам. Исходя из нашего опыта, продуктивнее выделять на дипломный проект по 2–3 часа несколько дней в неделю, чем делать этот же объём за один подход. Рекомендуем придерживаться этого графика и обязательно выделять время для отдыха.
- Отмечайте свой прогресс по мере выполнения плана. Это полезно по нескольким причинам: во-первых, вы будете держать ритм, во-вторых — контролировать ситуацию. И самое главное: каждый выполненный этап — это ваша маленькая победа. Чем больше таких побед вы будете замечать, тем большее удовольствие от выполненного проекта получите в итоге.
- ☑ Прежде чем отправлять каждую часть проекта на проверку преподавателю (а в реальной работе — тестировщику), всегда проверяйте за собой. Мы рекомендуем вам проверить весь проект целиком, когда вы его закончите. Идеально, если между завершением работы и вашей проверкой пройдёт некоторое время — это позволит вам отключиться и в результате посмотреть на работу свежим незамыленным взглядом. И помните, что ошибки при работе — это нормально, их совершают даже самые матёрые разработчики. Но по-настоящему хорошие специалисты отличаются тем, что способны самостоятельно найти свои ошибки и исправить их ещё до того, как их работа будет передана дальше.

- Помните о том, что логику отрисовки интерфейса (создания DOM-дерева) и логику работы с данными нужно разделять, если вы хотите, чтобы ваш код был простым и понятным. Например, вы можете сделать открытие модального окна с формой функцией, которая принимает объект с клиентом и создаёт соответствующее дерево элементов. А логику для обработки событий нажатия на кнопку вы можете передать дополнительными аргументами в виде функций по аналогии с приложением TODO. И будет совсем круто, если в эти функции вы будете передавать необходимую для обработки информацию: ID клиента для удаления или готовый объект с данными о клиенте для сохранения. Вы особенно почувствуете пользу такого подхода, если будете выполнять задания из раздела [Улучшения и дополнения](#). Примерная структура кода:

```
/**  
 * Функция создания модального окна.  
 */  
function createModalWithForm(client, { onSave, onClose }) {  
  const modalElement = document.createElement(...);  
  const form = document.createElement('form');  
  
  // ...создание DOM-дерева модального окна с формой...  
  
  form.addEventListener('submit', e => {  
    e.preventDefault();  
    const data = { ... }; // собираем объект из значений в полях для ввода  
    onSave(data, modalElement);  
  });  
  
  // ...  
  
  closeButton.addEventListener('click', () => {  
    onClose(modalElement);  
  });  
  
  // ...  
  
  return modalElement;  
}  
  
/**  
 * Что должно произойти при сохранении формы.  
 */  
async function onSave(formData, modalElement) {  
  const response = await fetch(...);  
  // ...  
  modalElement.remove();  
}  
  
/**  
 * Что должно произойти при закрытии формы.  
 */  
function onClose(modalElement) {  
  modalElement.remove();  
}  
  
// ...  
  
// добавляем обработчик нажатия на кнопку изменения клиента  
editButton.addEventListener('click', () => {  
  document.body.append(  
    createModalWithForm(client, { onSave, onClose })  
  );  
});  
  
// ...
```

- ☰ Для сложной структуры HTML вы можете разбить код на несколько вспомогательных функций, чтобы было проще ориентироваться. Например, для таблицы вы могли бы сделать функции для создания заголовочной строки и строки с клиентом. Также было бы удобно выделить в отдельную функцию создание иконки контакта со всплывающей подсказкой.

# ✨ Полезные ссылки и материалы ✨

- ◆ Если у вас возникают трудности при работе со страницей в браузере, вернитесь **к модулям 7 и 8**, где разбирается работа с DOM и событиями.
- ◆ Если вы не помните, как работать с сервером API, обратитесь **к 11-му модулю**, где разбирается клиент-серверное взаимодействие с помощью функции `fetch`.
- ◆ **В последнем уроке 5-го модуля** (области видимости, константы и стрелочные функции) разбирается работа с отладчиком Chrome. Он очень поможет вам понять, как работает ваш код и где в нём ошибки.



**У вас всё получится!**