Тестовое задание по вебу

Часть 1 (верстка, базовое программирование на стороне клиента)

**Задание**: По данному дизайн-макету сверстать клиентский экран с формой обратной связи и экраны панели администрирования. На форме обратной связи добавить валидацию полей в реальном времени на JS.

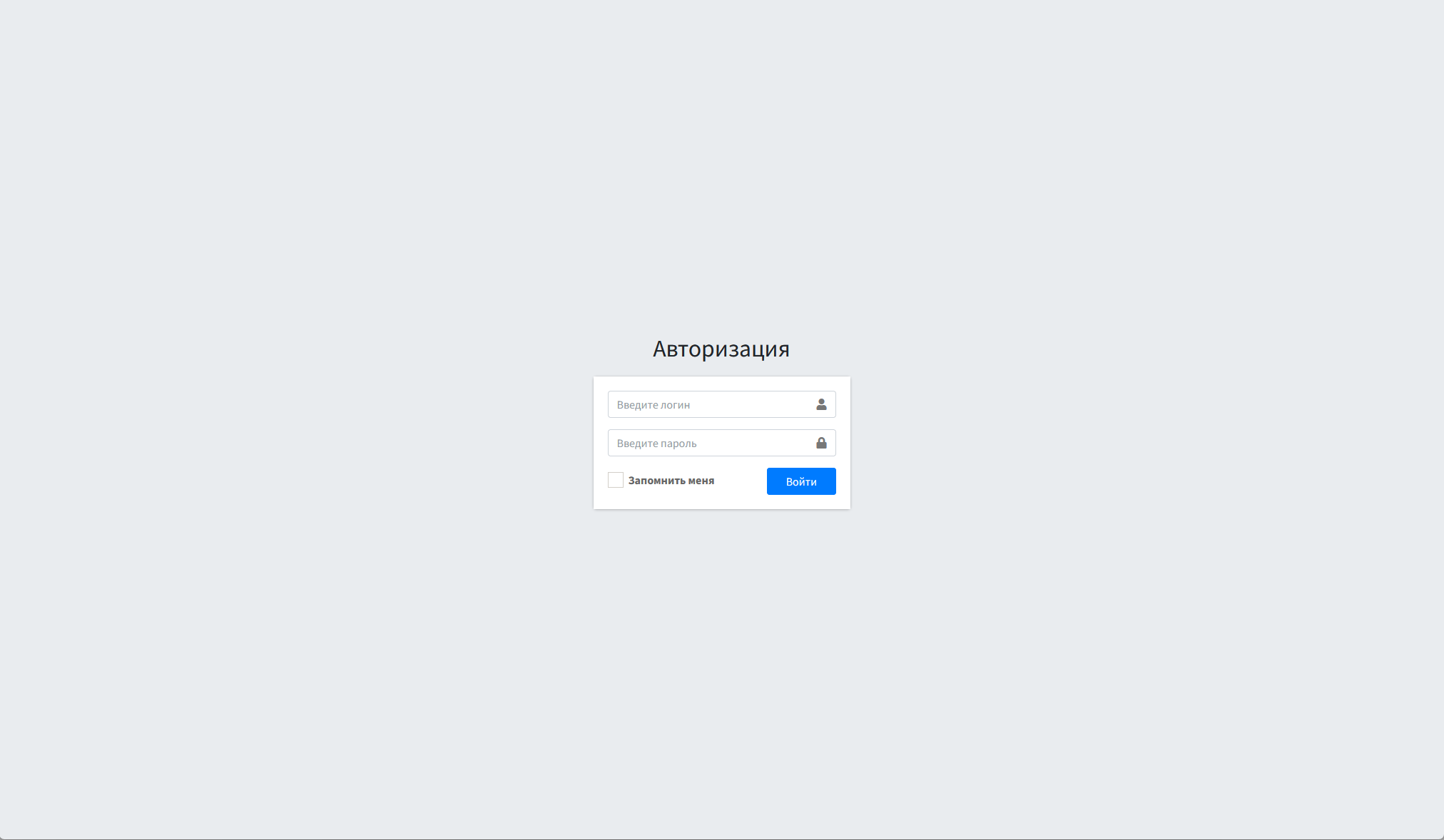
**Требования**: HTML, CSS, JS. Адаптация под широкоэкранные устройства (1600px+). Адаптация под мобильные устройства не требуется. Использование библиотек и фреймворков (Bootstrap, Tailwind, JQuery и т. п.) не разрешается.

Валидации для формы обратной связи:

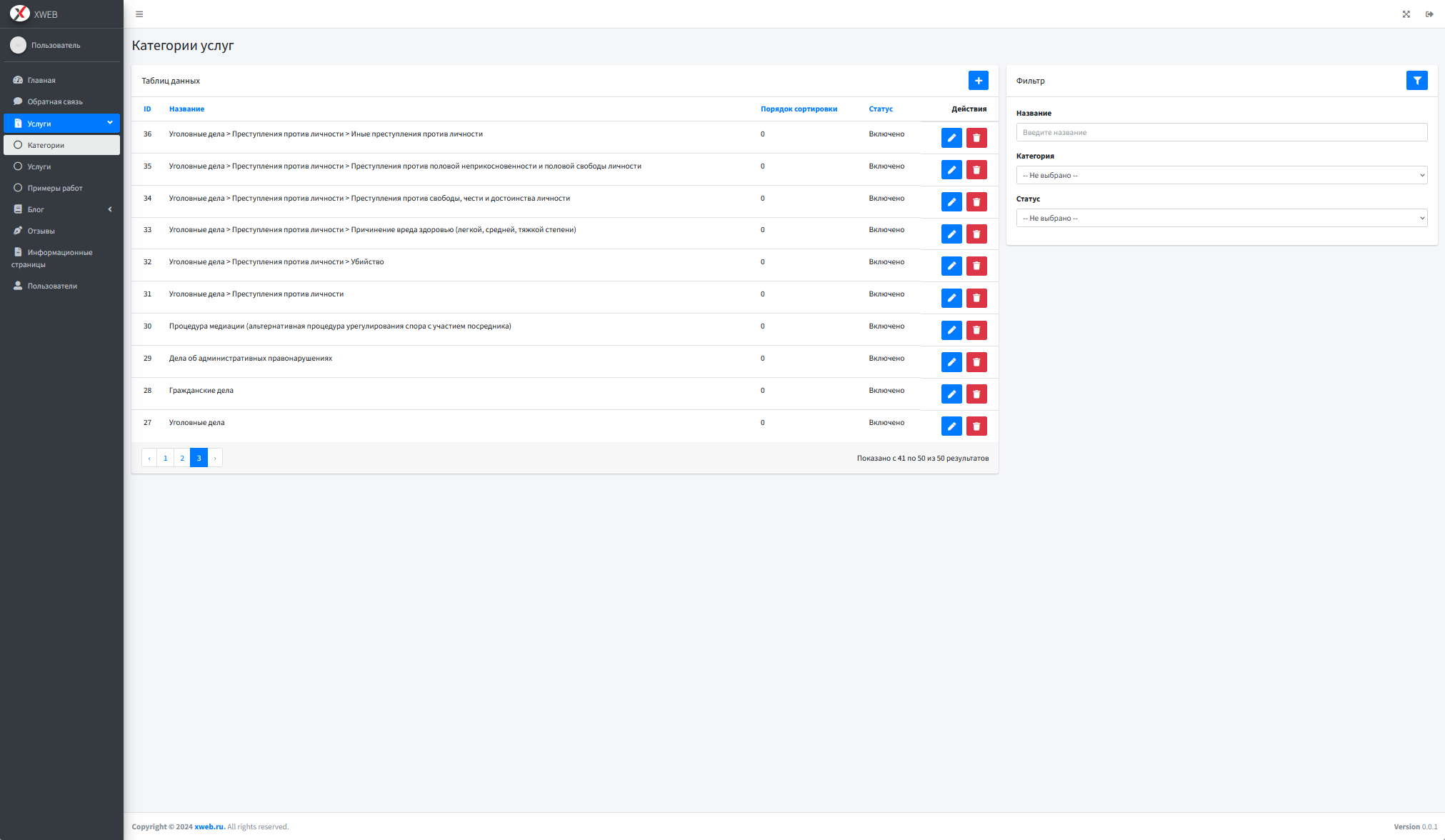
* Поле “Имя” - обязательное поле, только буквы, длина от 2 до 16 символов [здесь и далее проверка на заполненность поля должна быть сделана на JS]
* Поле “E-mail” - необязательное поле, соответствует формату email (“\*@\*.\*\*”) [тут так же дефолтная валидация HTML не считается]
* Поле “Номер телефона” - обязательное поле, соответствует формату “+7 (###) ###**-**##**-**##”. Дополнительно: сделать маску ввода для соответствия формату
* Поле “Сообщение” - необязательное поле, до 255 символов.
* Чекбокс “Согласие с политикой обработки персональных данных” - обязательное поле (должно быть прожато)

Если пользователь нажимает кнопку “Отправить” и в форме есть ошибки валидации, то рядом с полем должно выводиться сообщение (или сообщения), поясняющее ошибку.

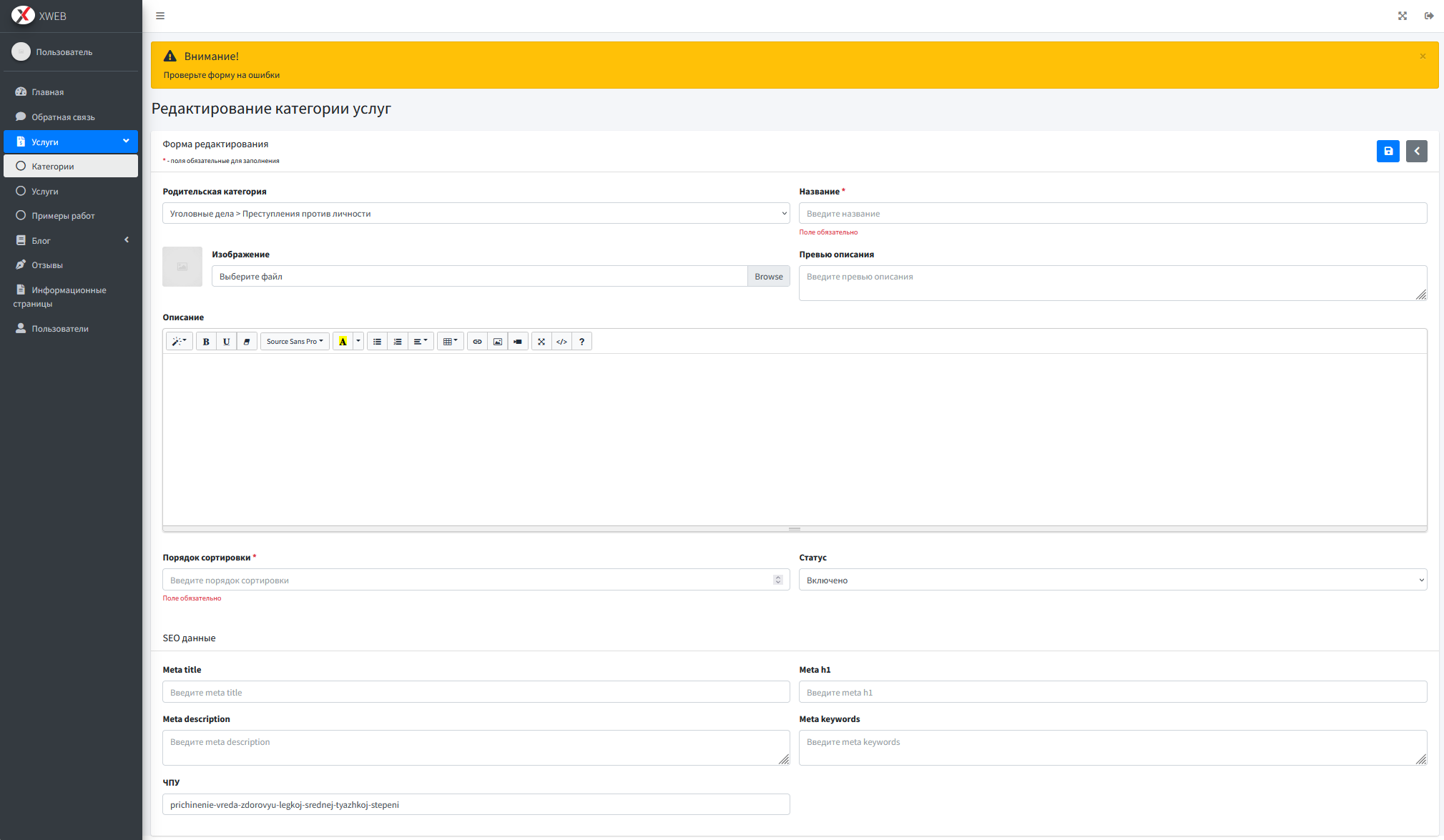
**Изображения примерного макета**:

****

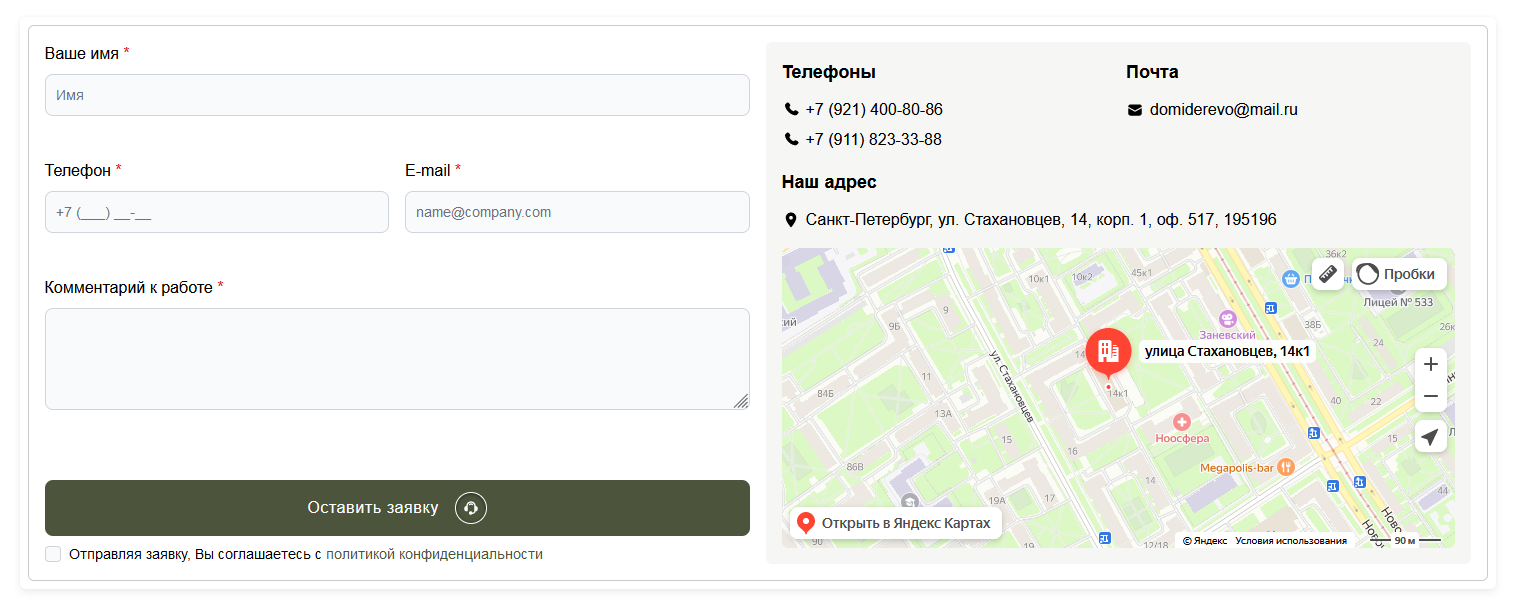
Пример страницы авторизации в админ-панель

****

Пример страницы со всеми данными раздела админ-панели

****

Пример формы редактирования/создания записи из раздела админ-панели

****

Пример формы обратной связи на клиентской части сайта

Часть 2 (работа с БД, программирование логики на стороне сервера)

**Задание**: Создать БД с необходимыми таблицами и полями, разработать базовый функционал панели администрирования, применить сверстанные в задании 1 экраны.

**Требования**: PHP/Node.js, использование фреймворка Laravel/Express (можно с библиотеками) настоятельно рекомендуется, СУБД - на свое усмотрение.

**Функционал**:

1. Возможность пользователю авторизоваться/разлогиниться в админ панели.
   1. Поля для авторизации: E-mail, Пароль.
   2. Обязательная валидация введенных данных со стороны сервера и вывод сообщения на полях, где она не была пройдена.
   3. При перезагрузке страницы, закрытии браузера авторизация не должна слетать.
   4. Если введенного пользователя нет в системе или он со статусом 0 (false), то выводит в нижней части формы ошибка “Неверный логин или пароль”.
   5. Если после авторизации пользователя заблокируют или удалят, то должно произойти разлогирование, при попытке сделать что либо в системе.
   6. После разлогина сделать редирект на страницу авторизации.
   7. Если пользователь не авторизован, то при попытке попасть на любой другой экран, должен происходить редирект на страницу авторизации.
   8. Если пользователь авторизирован, то при попытке попасть повторно на страницу авторизации, его редиректит обратно на прошлую страницу.
2. Возможность добавлять/редактировать/удалять пользователей, которые могут работать в панели администратора.
   1. Поля для пользователей: Имя, E-mail, Пароль, Повторите пароль, Статус, Дата создания, Дата редактирования.
   2. Текущего пользователь нельзя удалить, а при редактировании заблокировать поле статуса.
   3. При удалении любой записи сделать вывод модального окна с подтверждением удаления (можно использовать встроенные JS-овские методы).
   4. Страница отображения всех пользователей.
      1. Обязательная пагинация, по умолчанию на странице отображается 25 записей.
      2. По возможности сделать выбор количества записей на одной странице.
      3. По возможности сделать сортировки по разным полям (поля для отображения сделать на свой выбор).
      4. По возможности добавить блок фильтрации данных по Имени, E-mail, Статусу, Дате создания (если заполнены все параметры, то должны выбираться те данные, которые удовлетворяют всему).
   5. Форма добавления/редактирования пользователей.
      1. По возможности сделать только одну форму и для редактирования, и для создания пользователей.
      2. Обязательная валидация введенных данных со стороны сервера и вывод сообщения на полях, где она не была пройдена.
3. Возможность добавлять/редактировать/удалять заявки клиентов, которые пришли с формы обратной связи.
   1. Поля для заявок: Имя, E-mail, Номер телефона, Сообщение, Статус, Дата создания, Дата редактирования.
   2. При удалении любой записи сделать вывод модального окна с подтверждением удаления (можно использовать встроенные JS-овские методы).
   3. Страница отображения всех заявок.
      1. По умолчанию сделать сортировку заявок по статусу и по дате создания (сначала идут от старых до новых необработанные, после идут от новых до старых обработанные).
      2. Обязательная пагинация, по умолчанию на странице отображается 25 записей.
      3. По возможности сделать выбор количества записей на одной странице.
      4. По возможности сделать сортировки по разным полям (поля для отображения сделать на свой выбор).
      5. По возможности добавить блок фильтрации данных по Имени, E-mail, Номеру телефона, Статусу, Дате создания (если заполнены все параметры, то должны выбираться те данные, которые удовлетворяют всему).
   4. Форма добавления/редактирования заявок.
      1. По возможности сделать только одну форму и для редактирования, и для создания заявок.
      2. Обязательная валидация введенных данных со стороны сервера и вывод сообщения на полях, где она не была пройдена.

Часть 3 (связка фронтенда и бекенда)

**Требования**: После нажатия кнопки отправки клиент должен отправлять данные формы на сервер, если хоть на одном поле произошла ошибка валидации, то на сервер ничего не отправляется.

Если на сервер все же было отправлено сообщение, то в случае успешной валидации на стороне сервера возвращается результирующая страница с сообщением об успешной отправкой формы или модальное окно (если выполнять дополнительное задание).

Дополнительное задание

**Требования**: После нажатия кнопки отправки клиент должен отправлять асинхронный запрос к серверу с сериализованными в JSON данными формы.

В случае не прохождении валидации на стороне сервера, ответ сервера должен содержать JSON данные с полями, в которых не прошла валидации и по какой причине.

Формат JSON данных обращения клиента и ответа сервера - делать на свое усмотрение.