**КУРСОВАЯ РАБОТА**

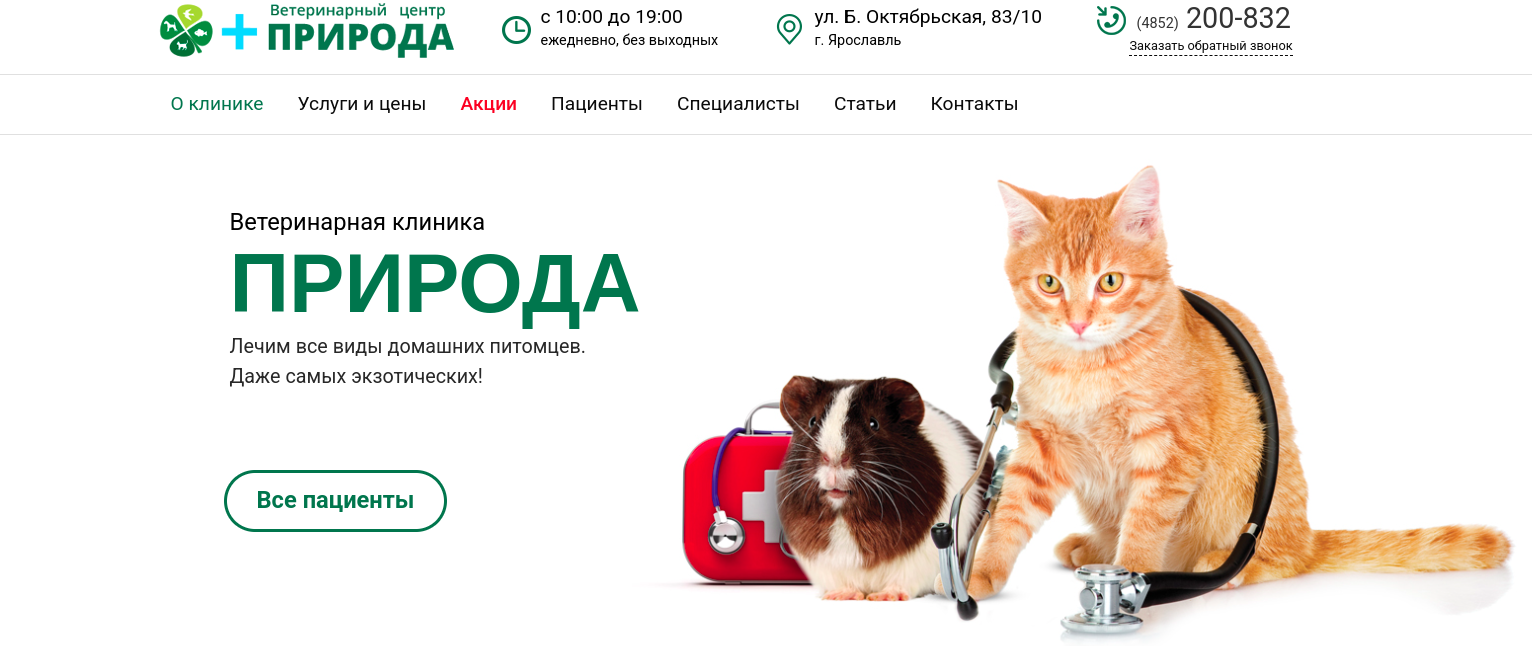
**ПО ТЕМЕ "ВЕТЕРИНАРНАЯ КЛИНИКА"**

**Выполнил Косолапов Кирилл, студент группы ИС 1-31**

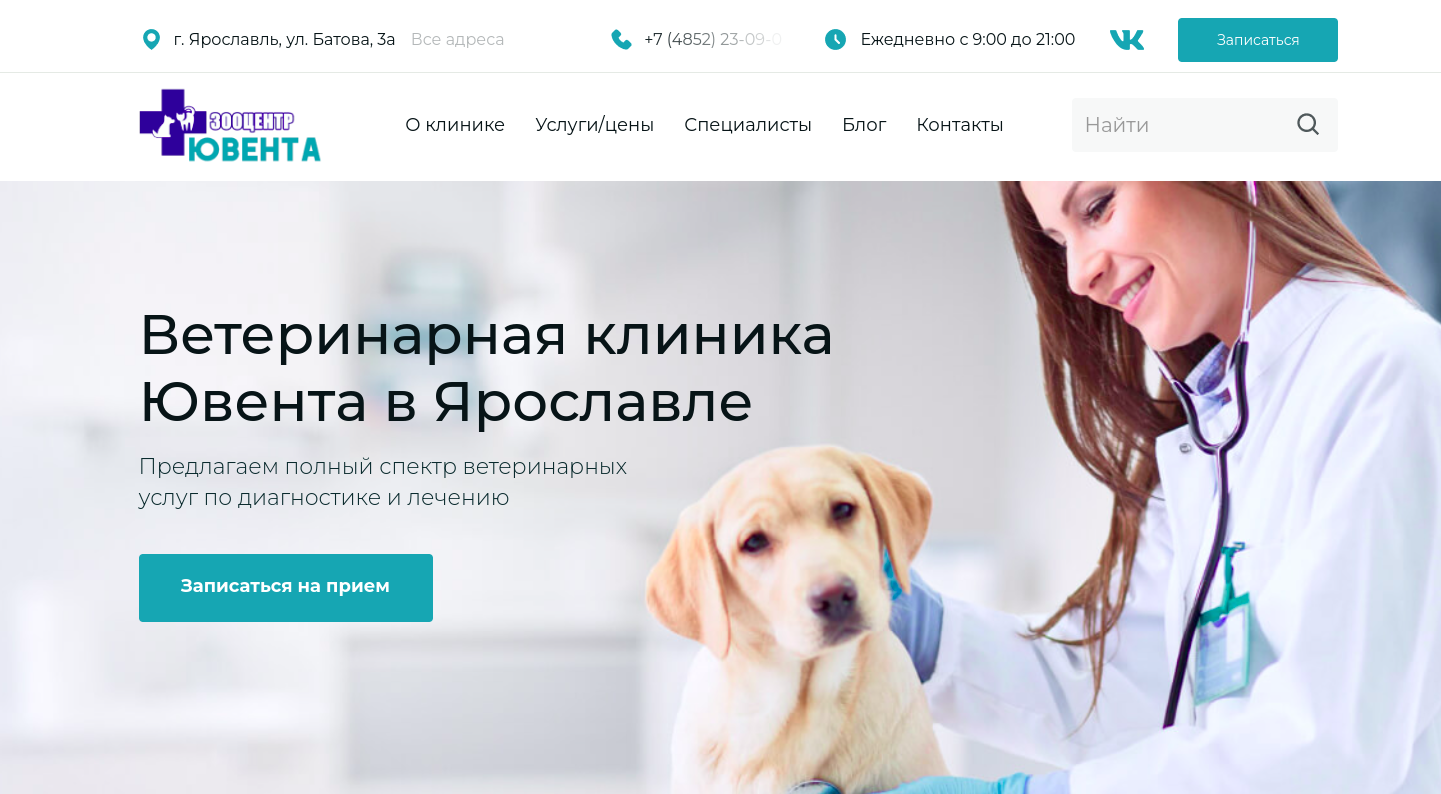
**Сайты-аналоги**

Ветеринарный центр “ПРИРОДА”

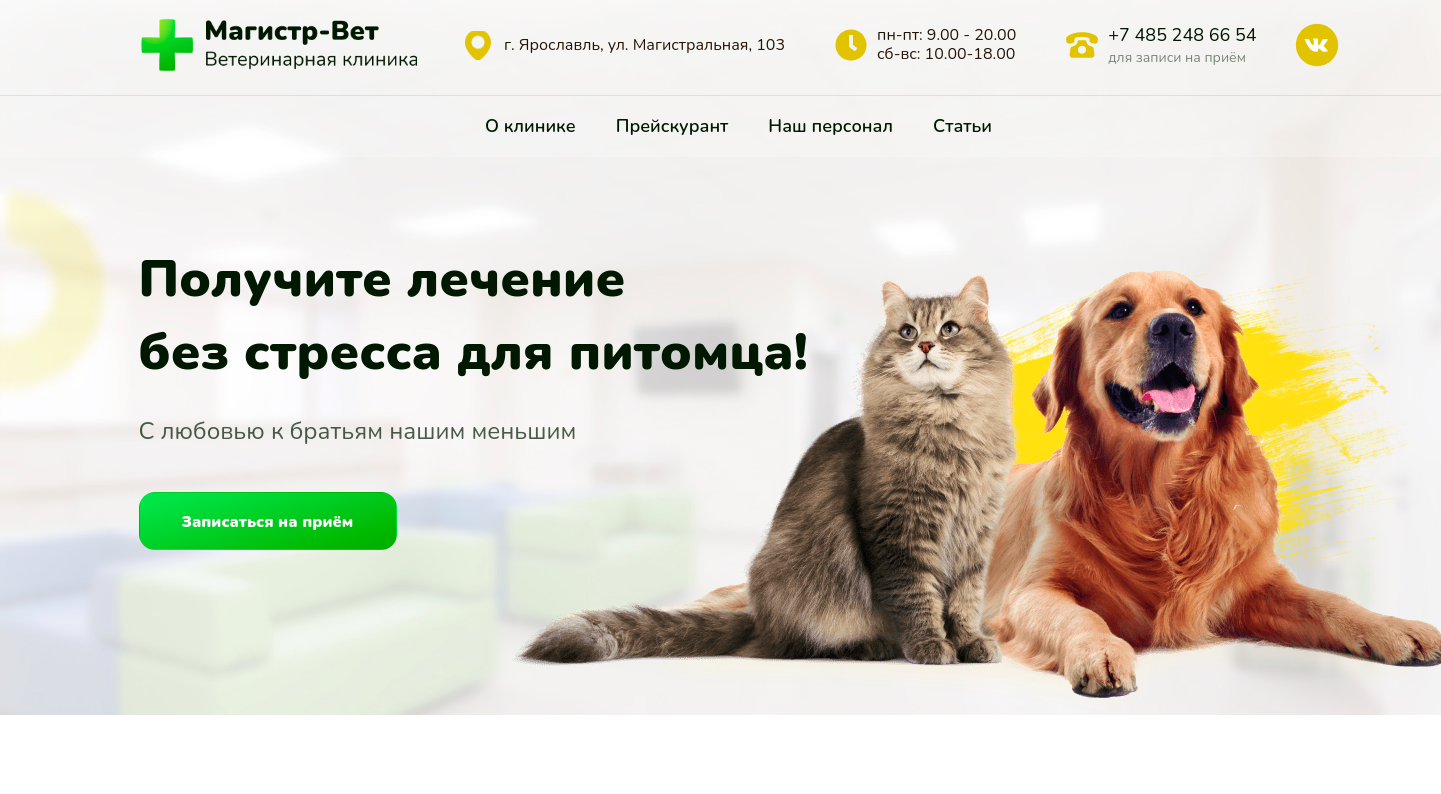
Ссылка - <https://vet-76.ru/>

  
Зооцентр "ЮВЕНТА"

Ссылка - <https://uventa76.ru/>

  
Ветеринарная клиника Магистр-Вет

Ссылка - <https://vet-mataeva.ru/>

Что я предпочту использовать в своем сайте?

* Картинка на заднем фоне сайта
* Вкладке в "шапке" сайта
* Контактные данные на главной странице

Чего бы я предпочел избежать?

* Нарушение цветовой гаммы и шрифтов сайта в элементах (особенно в контактных данных)
* Неадаптивные элементы на сайте

**Целевая аудитория**

Люди, которые заботятся о своих питомцах и озабоченны их здоровьем

Пол: Любой

Возраст: 20-50 (средний возраст)

Средний доход: 50000 руб в месяц и больше

**Структура сайта**

Главная страница

* Конткатные данные
* Врачи-сотрудники
* Новости
  + Страница выбранной новости

Фотогалерея

* Фото из поликлинники

О нас

* О нашей ветеринарной клинике
* Наша миссия
* Наша команда
* Наш подход

Контакты

* Контактные данные клиники (расширенные)

Отзывы

* Отзывы клиники
* Форма для написания отзыва

Авторизация в админ панель

* Форма для авторизации в админ панеле (Логин и пароль – admin)
* После авторизации все элементы сайта становятся редактируемыми для администратора

**Комментарии по коду**

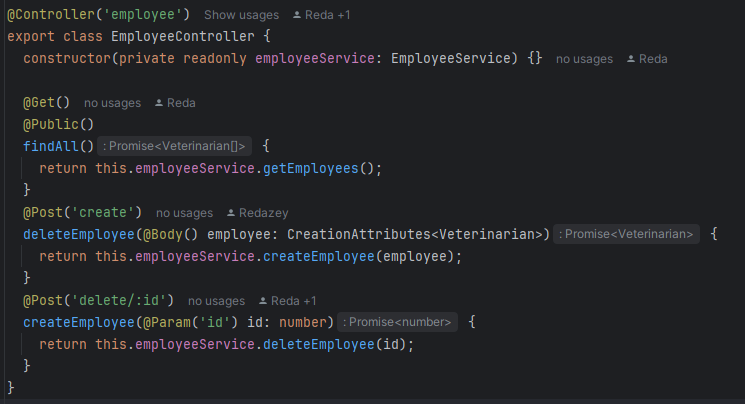
**Backend**

Контроллер, отвечающий за CRUD операции с новостями  


Сервисы, которые использует контроллер



Контроллер, отвечающий за CRUD операции с ветеринарами (сотрудниками)



Сервисы, которые использует контроллер



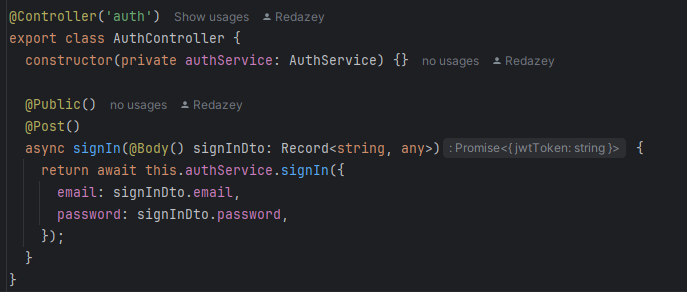
Контроллер, отвечающий за CRUD операции с отзывами



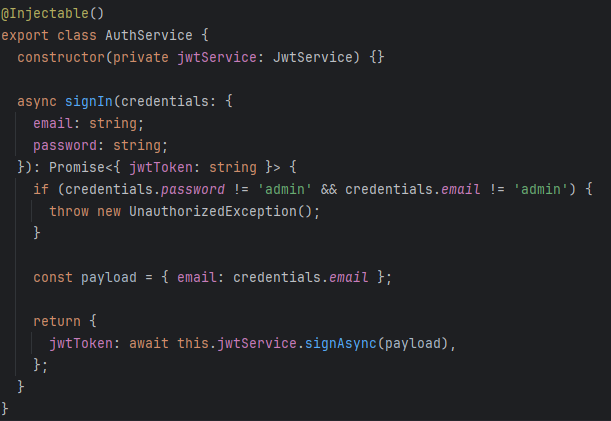
Сервисы, которые использует контроллер



Контроллер, отвечающий за получение JWT – токена для авторизации в админ панеле

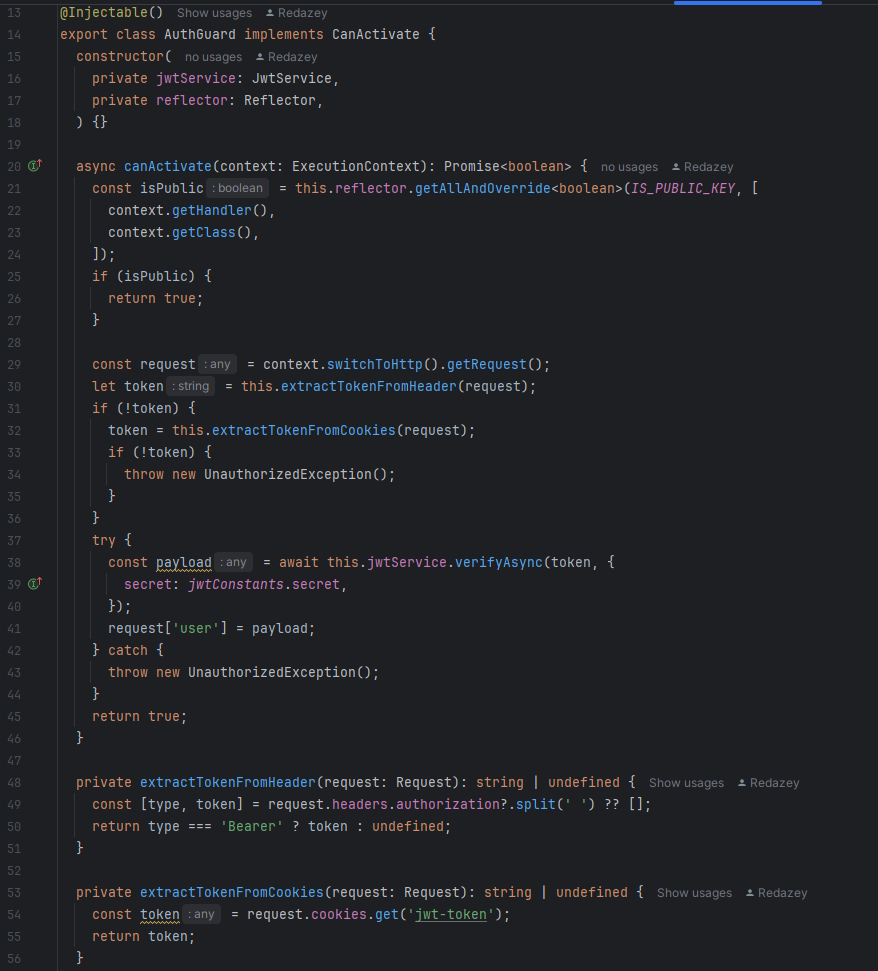


Сервис, который использует контроллер



Метод защиты непубличных (админ) роутов путем проверки JWT токена

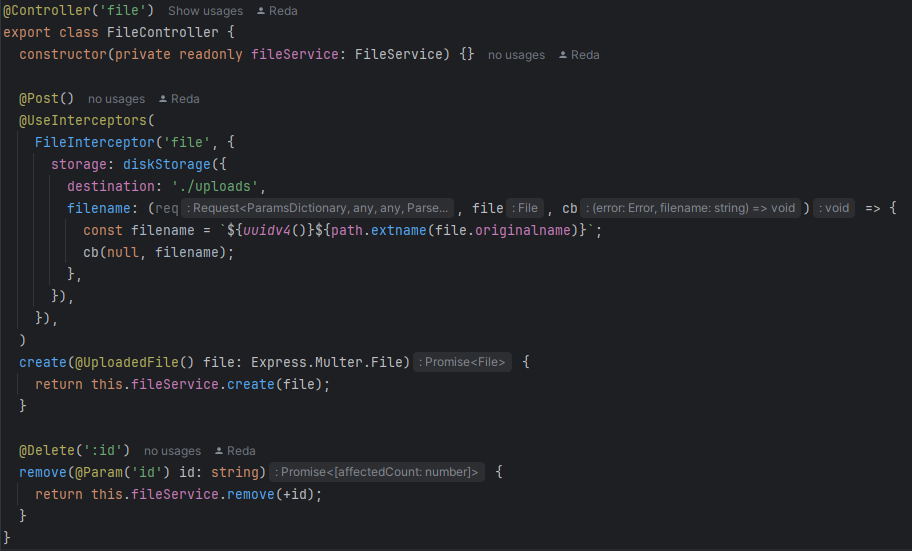
Он встраивается как middleware на все роуты, которые не обозначены как @Public() и проверяет заголовки запроса на наличие Authorization: Bearer [токен], затем валидирует токен и если он корректный, то пускает запрос на роут, в ином случае возвращает статус 403 (Неавторизован)



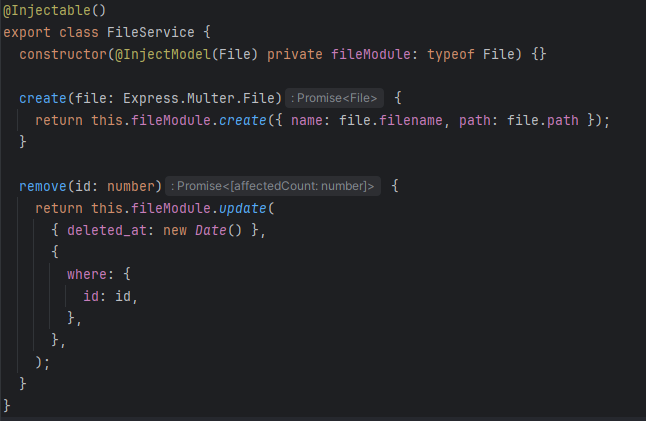
Все картинки, которые загружаются через формы хранятся в backend/uploads/ и подгружаются по адресу <http://localhost:3001/uploads/имя_файла>

Это возможно благодаря папке, которая обозначена как доступная для загрузки с нее файлов

  
Контроллер, отвечающий за загрузку файлов на сервер backend

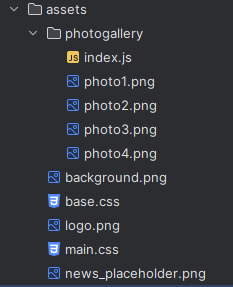


Сервисы, которые использует контроллер

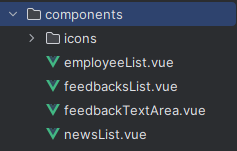


**Frontend**

Стили, статичные картинки (логотип, фотогалерея) и т.д. хранятся в папке assets



**Компоненты, из которых состоят страницы**

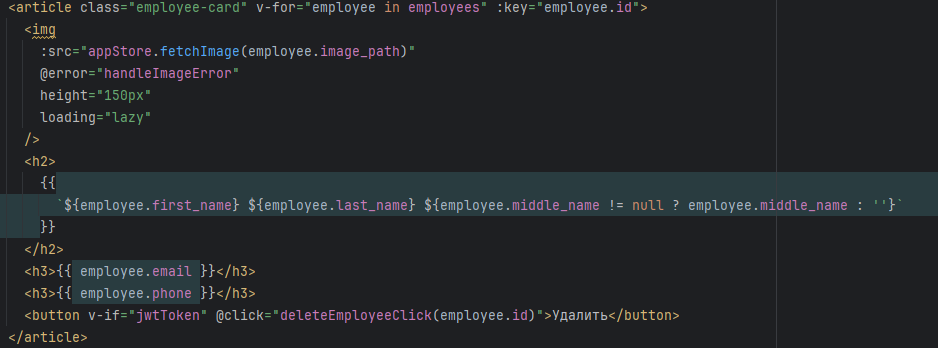


Компонент «сотрудники» (employeeList.vue)

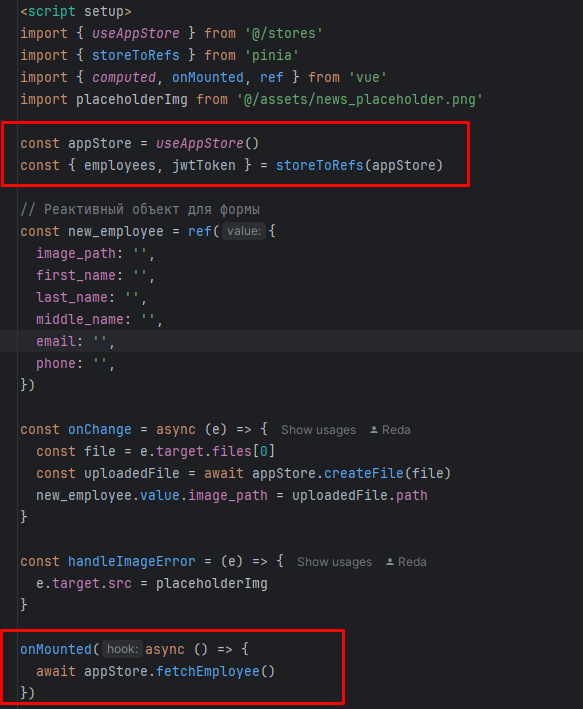
Он очень обьемный, так что обозначу основные моменты

«Карточки» сотрудников, которые генерируются из списка сотрудников, получаемого через API

Под каждой есть кнопка удалить, в случае если пользователь авторизован

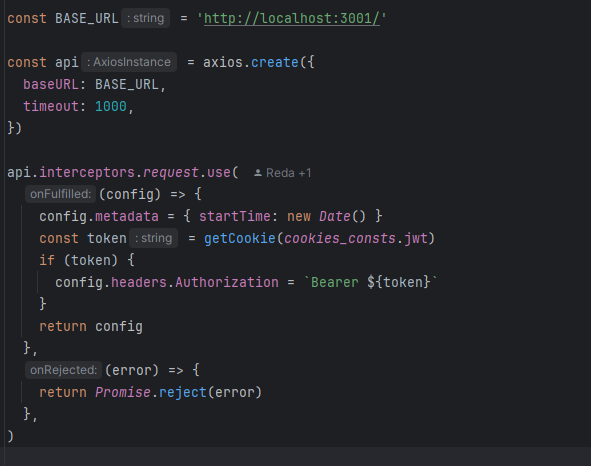


Получение сотрудников по API



Методы, использованные для этого

Создание Axios обьекта, для упрощенного обращения к API и автоматического использования Cookie



CRUD операции с сотрудниками



Компонент «Новости» (newsList.vue)



«Карточки» новостей, которые генерируются из списка новостей, получаемого через API (по тому же принципу, что и сотрудники, отличаются только роуты и названия переменных)

Под каждой есть кнопка удалить, в случае если пользователь авторизован

Форма админ панели, через которую добавляются новые новости



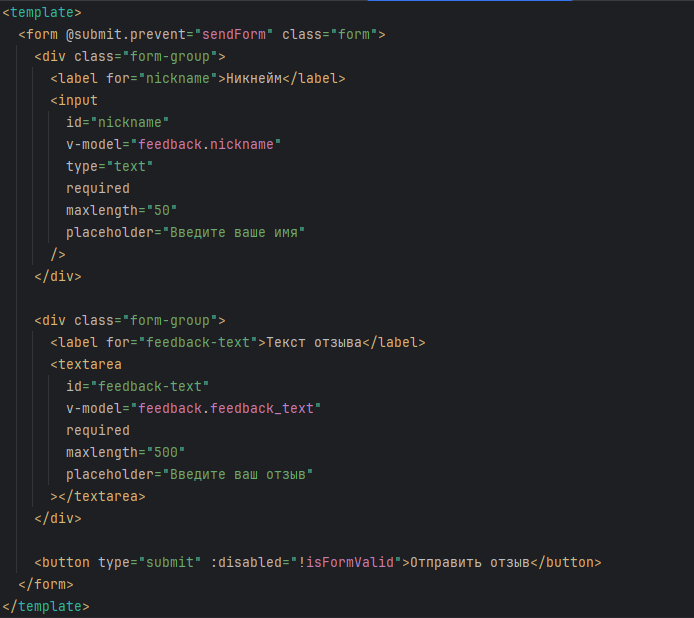
Компонент «Отзывы» (feedbacksList.vue)

Отзывы, которые генерируются из списка, получаемого через API

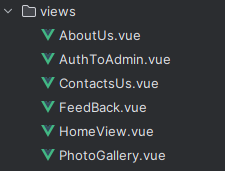
Под каждым есть кнопка удалить, в случае если пользователь авторизован



Форма добавления отзыва



Далее идут Views (фактически страницы, собранные из компонентов)



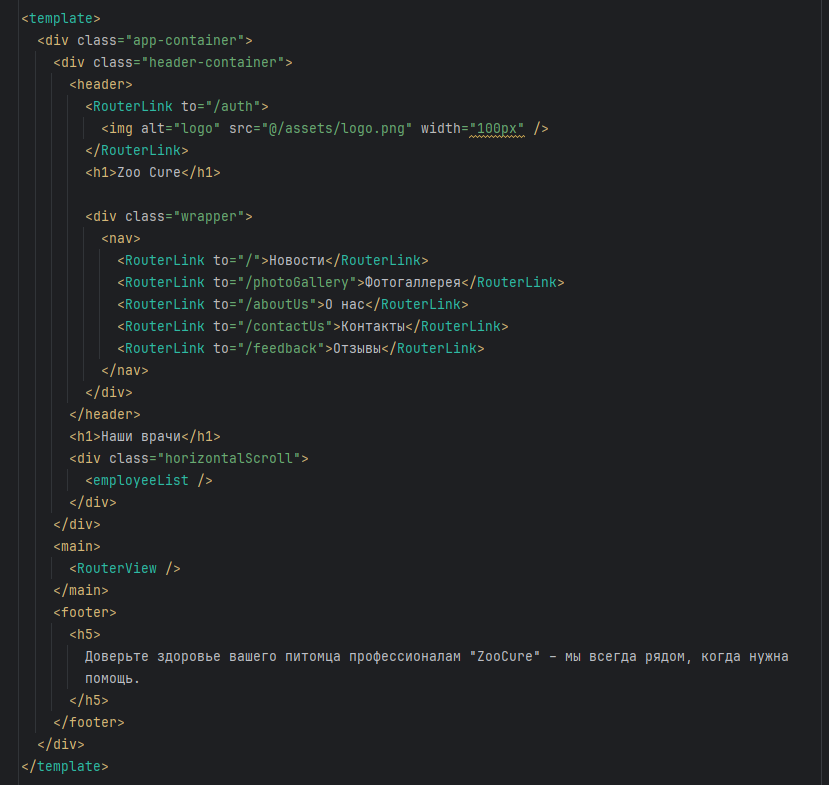
Форма авторизации в админ панель



Фотогалерея, реализованная в виде карусели, использует статичные данные



Главная страница (страница, в которой меняется контент, в ней же находится панель навигации)



Роутер, отвечающий за привязку views к определенным путям



LocalStorage сайта, или же PiniaStore, в котором определяются переменные, хранящие в себе данные, полученные через API, а так же методы получения этих данных

