|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (УрФУ)  ИРИТ-РТФ  Базовая кафедра «Аналитика больших данных и методы видео анализа» |

Оценка по проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель проекта Медведев М. А.

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет по проекту на тему

**«Сайт управляющей компании «Лесной»»**

Студенты: Крапивин Михаил Александрович \_\_\_\_\_\_\_\_\_\_\_\_\_

Лубин Евгений Игоревич \_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО) (Подпись)

Группа : \_\_\_\_\_\_\_\_РИ-311055\_\_\_\_\_\_\_\_\_\_\_\_\_

Екатеринбург

2024

СОДЕРЖАНИЕ

[Введение 3](#_Toc168789268)

[1 Анализ предметной области 4](#_Toc168789269)

[2 Проектирование приложения 6](#_Toc168789270)

[3 Выбор инструментальных средств разработки 7](#_Toc168789271)

[4 Разработка прототипов и дизайн-шаблонов 8](#_Toc168789272)

[5 Разработка клиентской части приложения 18](#_Toc168789273)

[5.1 index.html 18](#_Toc168789274)

[5.2 register.html 23](#_Toc168789275)

[5.3 login.html 25](#_Toc168789276)

[5.4 send\_complaint.html 26](#_Toc168789277)

[5.5 complaints.html 28](#_Toc168789278)

[5.6 my\_complaints.html 28](#_Toc168789279)

[5.7 send\_meterreadings.html 29](#_Toc168789280)

[5.8 send\_message.html 32](#_Toc168789281)

[5.9 user\_messages.html 34](#_Toc168789282)

[5.10 admin\_messages.html 36](#_Toc168789283)

[5.11 reply.html 37](#_Toc168789284)

[5.12 news.html 37](#_Toc168789285)

[6 Разработка серверной части приложения 39](#_Toc168789286)

[6.1 Структура проекта 39](#_Toc168789287)

[6.2 База данных 42](#_Toc168789288)

[6.3 Процессы взаимодействия сервера с клиентом 46](#_Toc168789289)

[Заключение 61](#_Toc168789290)

# Введение

В условиях современной городской инфраструктуры управляющие компании играют ключевую роль в поддержании благосостояния жителей. Эффективное управление жилищным фондом требует наличия удобных и функциональных инструментов для взаимодействия с жителями, решения их проблем и повышения уровня удовлетворенности услугами. Создание сайта управляющей компании позволит улучшить коммуникацию с жителями, автоматизировать процессы подачи и обработки жалоб, передачи показаний счетчиков и получения новостей, а также повысить прозрачность деятельности компании.

**Целью** данного проекта является разработка функционального веб-сайта для управляющей компании «Лесной», который обеспечит удобный доступ жителей к информации и возможность взаимодействия с компанией.

**Поставленные задачи**:

1. Разработать главную страницу сайта с основной информацией о компании.
2. Создать формы для регистрации и авторизации пользователей.
3. Реализовать функционал подачи жалоб пользователями.
4. Разработать страницы для отображения всех жалоб и жалоб, поданных конкретным пользователем.
5. Реализовать возможность передачи показаний счетчиков (сохранение их в базе данных).
6. Добавить функцию просмотра новостей.
7. Реализовать функционал обмена сообщениями между пользователем и компанией.
8. Оформить стили сайта с использованием CSS.

# 1 Анализ предметной области

Проект представляет собой веб-сайт управляющей компании «Лесной». Основными функциями сайта являются предоставление информации о компании, регистрация и авторизация пользователей, возможность подачи и просмотра жалоб, связь со службой поддержки, просмотр новостей, а также передача показаний счетчиков.

Сфера деятельности проекта – жилищно-коммунальное хозяйство (ЖКХ). В данной сфере важны оперативность и точность в обработке обращений жителей, прозрачность деятельности управляющей компании и доступность информации.

**Предметами** и **объектами** в сфере деятельности, затрагиваемой нашим проектом, являются:

1. Жильцы: пользователи системы, которые могут подавать жалобы и просматривать их, отправлять сообщения в службу поддержки и просматривать их, следить за новостями, отправлять показания счетчиков.
2. Управляющая компания: организация, обрабатывающая жалобы, новостные публикации и взаимодействующая с жильцами.
3. Жалобы: обращения жителей, требующие решения.
4. Новости: просматриваются пользователями, публикуются компанией.
5. Показания счетчиков: передаются пользователями (жильцами) посредством заполнения соответствующей формы.
6. Сообщения от пользователей/службы поддержки: отправляются пользователями/службой поддержки.

**Взаимодействия:**

* регистрация пользователей;
* авторизация пользователей;
* подача жалоб;
* просмотр жалоб;
* отправка показаний счетчиков;
* отправка сообщений в службу поддержки и получение обратной связи;
* просмотр новостей.

# 2 Проектирование приложения

Проект построен на базе Django Framework и включает в себя следующие компоненты:

* **Клиентская часть**: фронтенд, реализующий интерфейс взаимодействия с пользователями.
* **Серверная часть**: бэкенд, обрабатывающий запросы от клиентской части и взаимодействующий с базой данных.
* **База данных**: хранилище данных о пользователях, их жалобах, сообщениях и показаниях счетчиков, а также новостях организации.

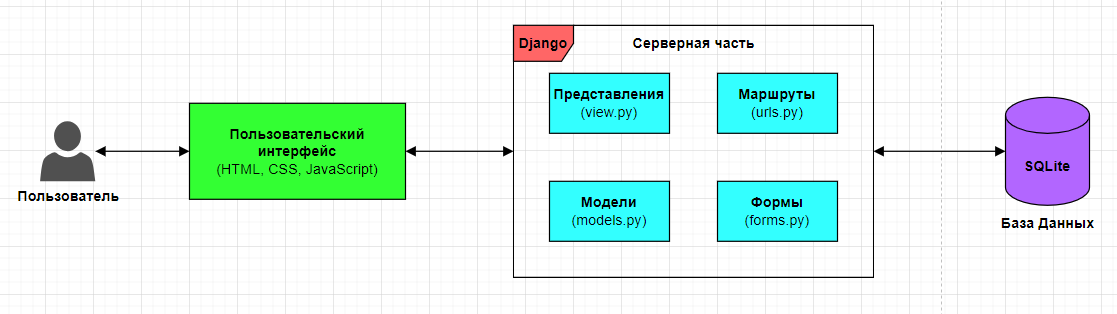
Для наглядности была создана Архитектурная диаграмма продукта (рис. 1)

Рисунок 2.1 – Архитектурная диаграмма

# 3 Выбор инструментальных средств разработки

В качестве редактора кода был выбран Visual Studio Code (с расширениями «SQLite Viewer», «Python», «HTML CSS Support»). Для разработки проекта были выбраны следующие технологии:

1. Django Framework – это фреймворк для Python, с помощью которого можно создавать сайты и приложения, работающие в браузере. Он будет использован для разработки серверной части.
2. HTML/CSS – это технологии для создания и стилизации веб-страниц. Также для фронтенда был выбран HTML, CSS фреймворк Bootstrap.
3. JavaScript – это язык программирования, используемый в веб-разработке. Он будет использован для валидации форм.
4. SQLite – это быстрая и встраиваемая однофайловая СУБД. Она будет использована для хранения необходимых данных.

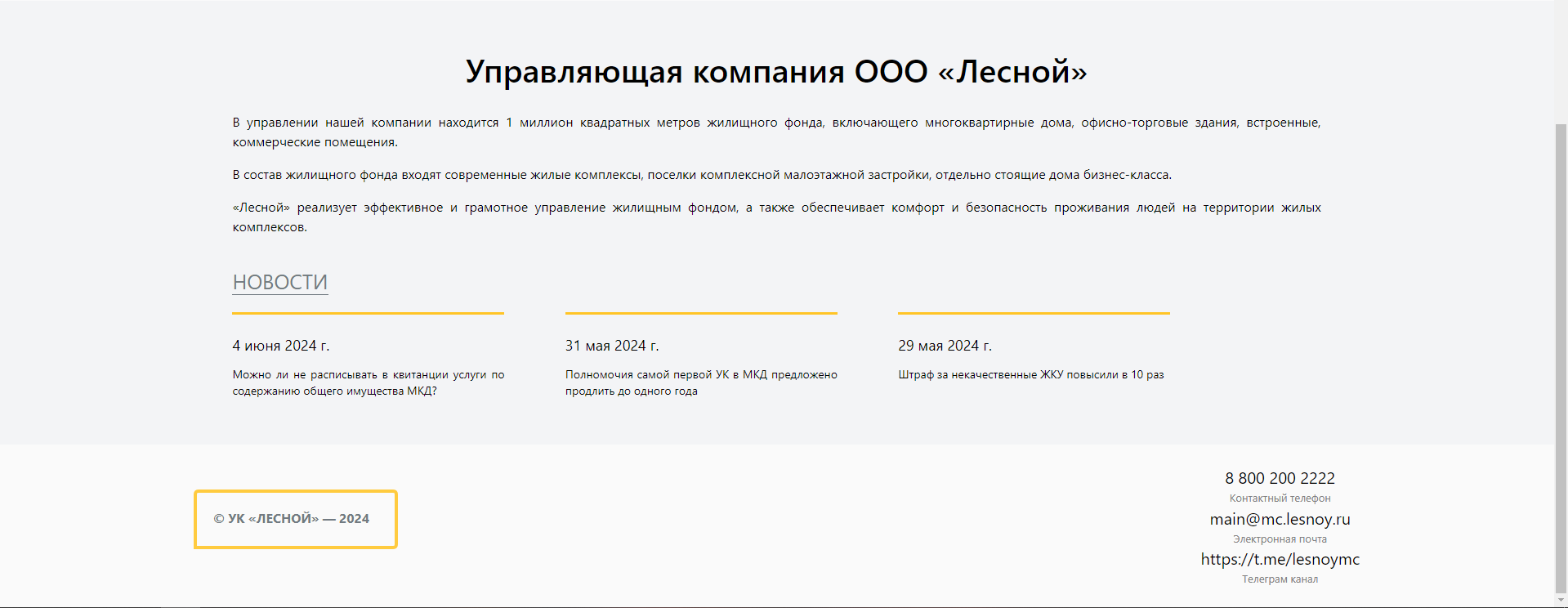
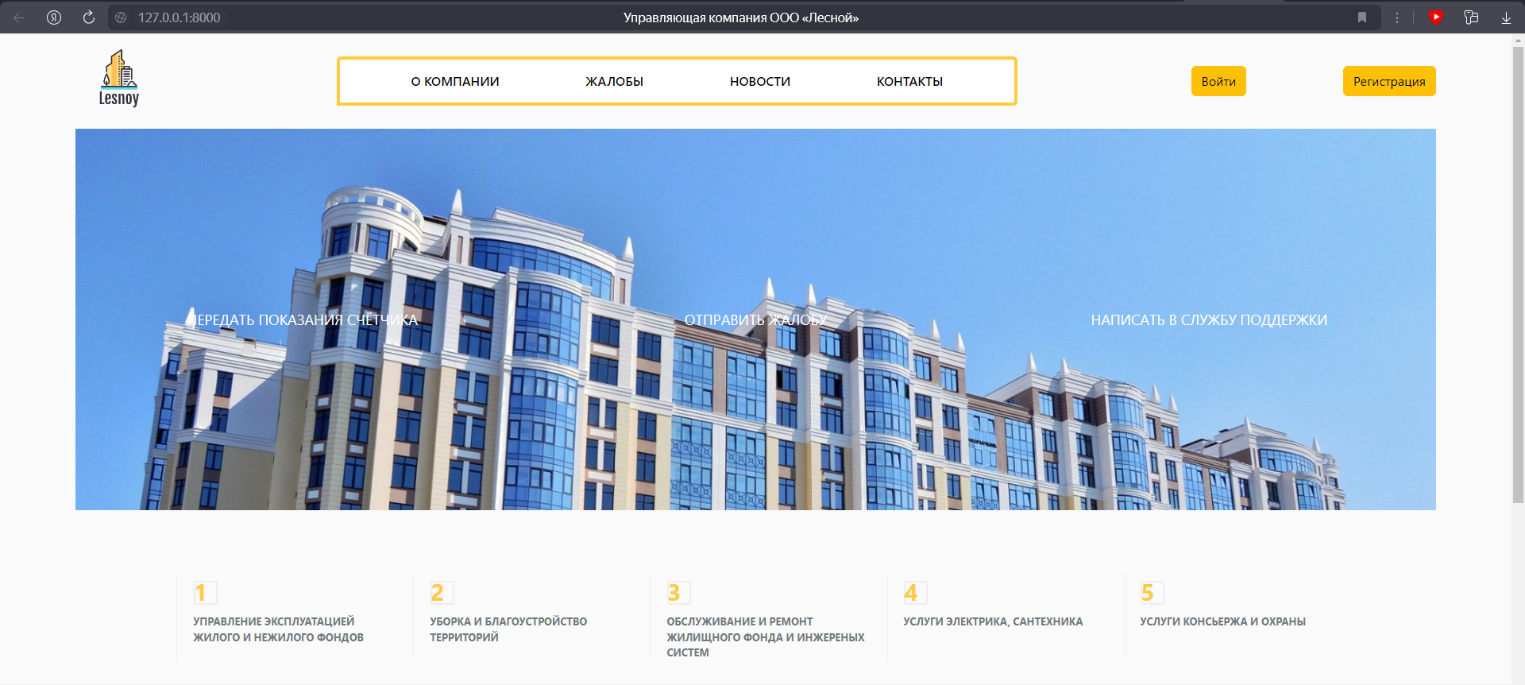
# 4 Разработка прототипов и дизайн-шаблонов

На каждой странице предусмотрены основные блоки: шапка сайта с навигацией, основной контент и подвал сайта.

**Основные элементы:**

1. Главная страница (рис. 4.1).

Рисунок 4.1 – Главная страница



На главной странице в шапке находится логотип компании. Для удобства использования сайта в ней расположена навигационная панель, а также кнопки со ссылками на страницы авторизации и регистрации.

В основном блоке находиться анимация (смена двух картинок), на переднем плане которой расположены основные опции сайта: «Передать показания счётчика», «Подать жалобу» и «Написать в службу поддержки». Ниже перечислен список услуг компании.

В разделе «О компании» расположено резюме об организации, а также блок свежих новостей компании.

Подвал страницы содержит контактные данные для обратной связи с компанией.

1. Окно регистрации (рис. 4.2).

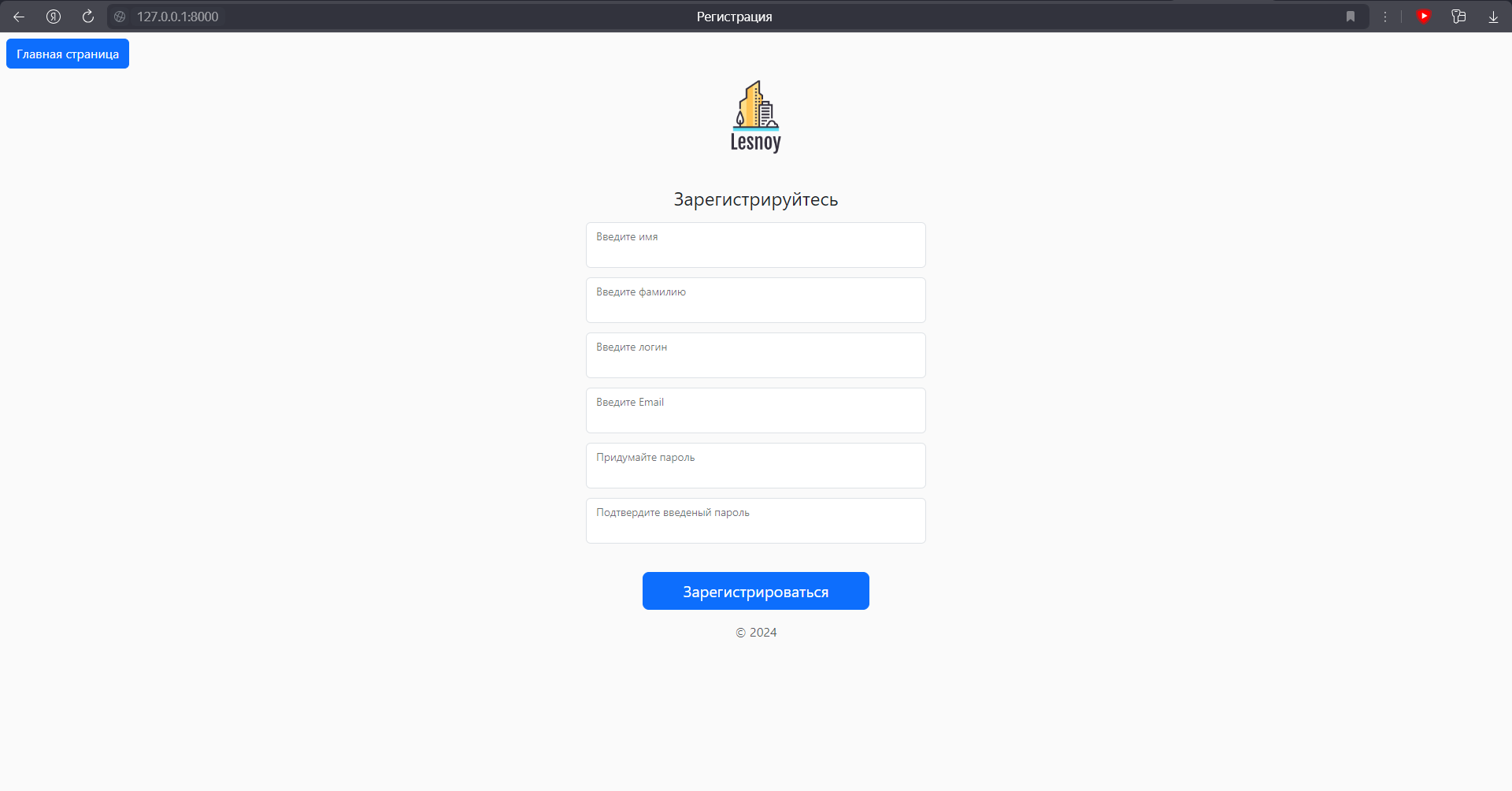


Рисунок 4.2 – Окно регистрации

При нажатии на кнопку «Регистрация» открывается страница для ввода пользовательских данных (имя, фамилия, логин, email, пароль и его подтверждение). Если все поля заполнены правильно, пользователь добавляется в базу данных компании.

1. Окно авторизации (рис. 4.3).

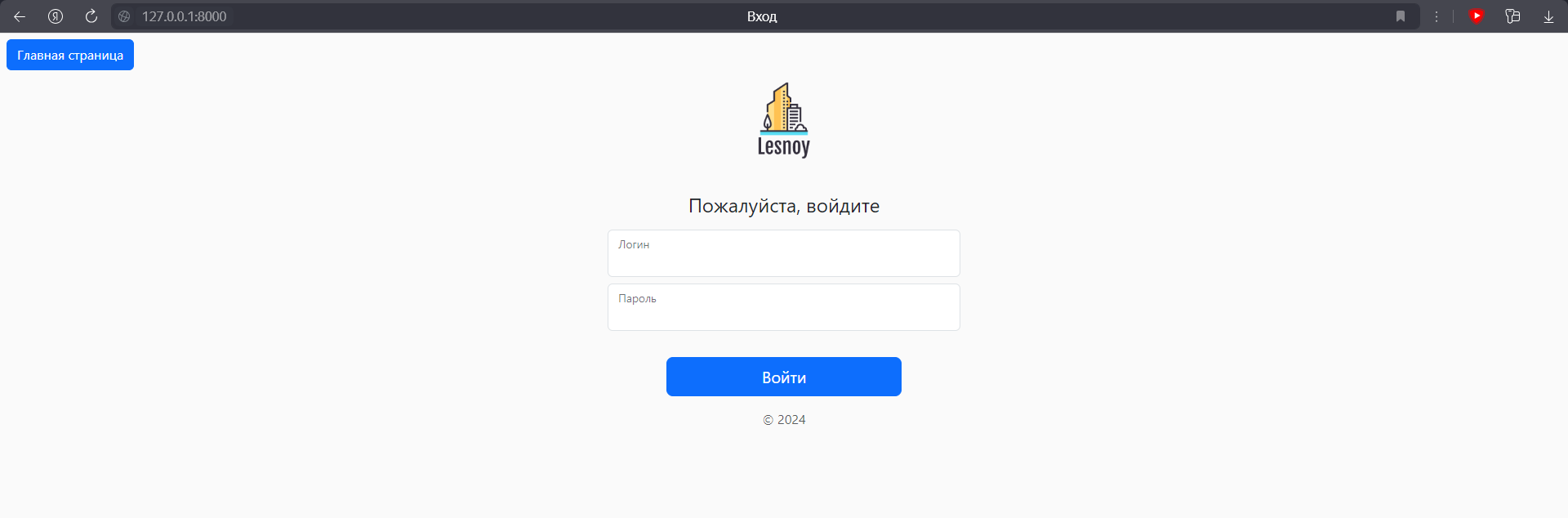


Рисунок 4.3 – Окно авторизации

Зарегистрированные пользователи могут сразу нажать на кнопку «Войти», и их перенесет на страницу входа, где нужно ввести логин и пароль. После этого они возвращаются на основную страницу и могут пользоваться опциями, которые не были им доступны в неавторизованном режиме.

1. Отправка жалобы (рис. 4.4).

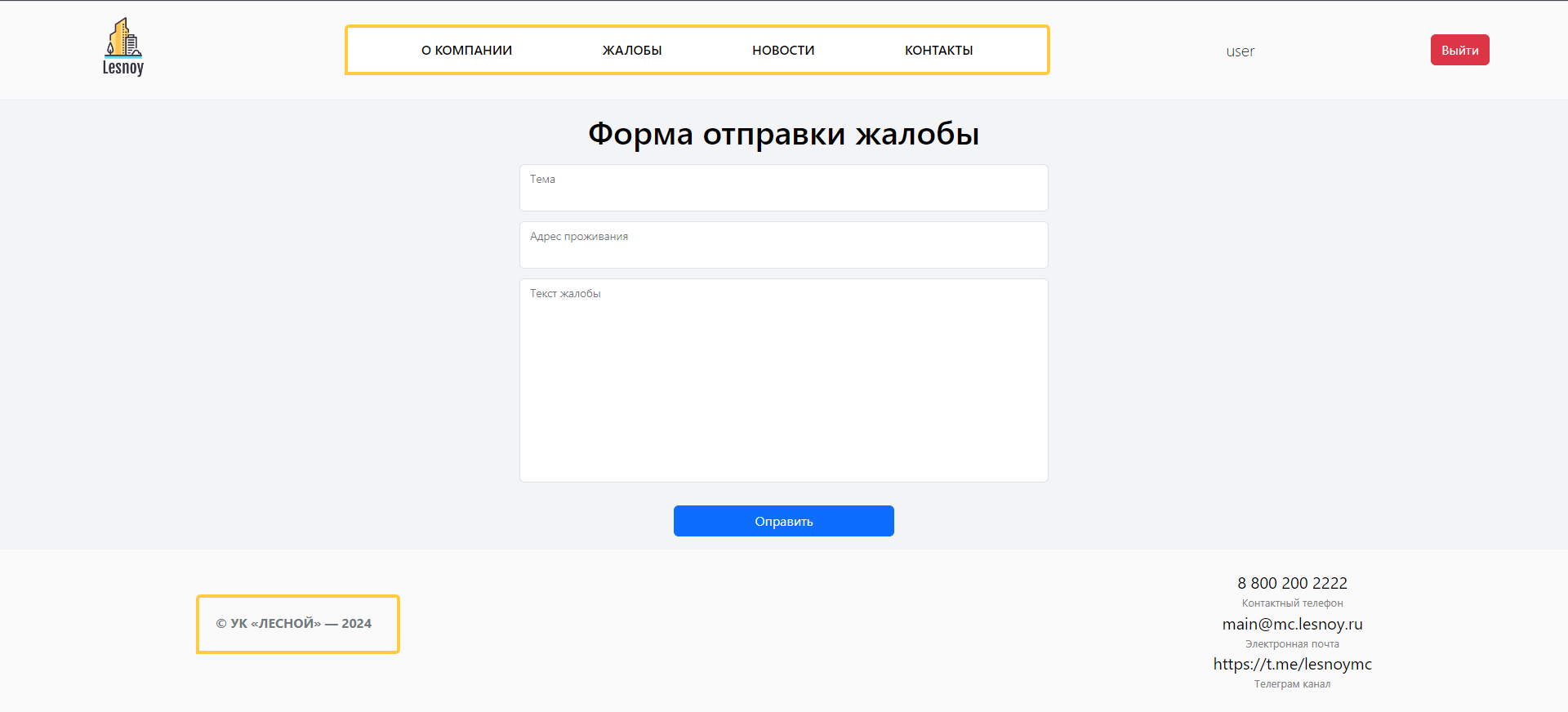


Рисунок 4.4 – Окно отправки жалобы

Если пользователь хочет отправить жалобу, то он может нажать на соответствующую опцию, расположенную на блоке с анимацией. После нажатия он попадет на данную страницу, где ему надо указать тему жалобы, адрес проживания и содержание жалобы.

1. Списки жалоб (рис. 4.5, 4.6).

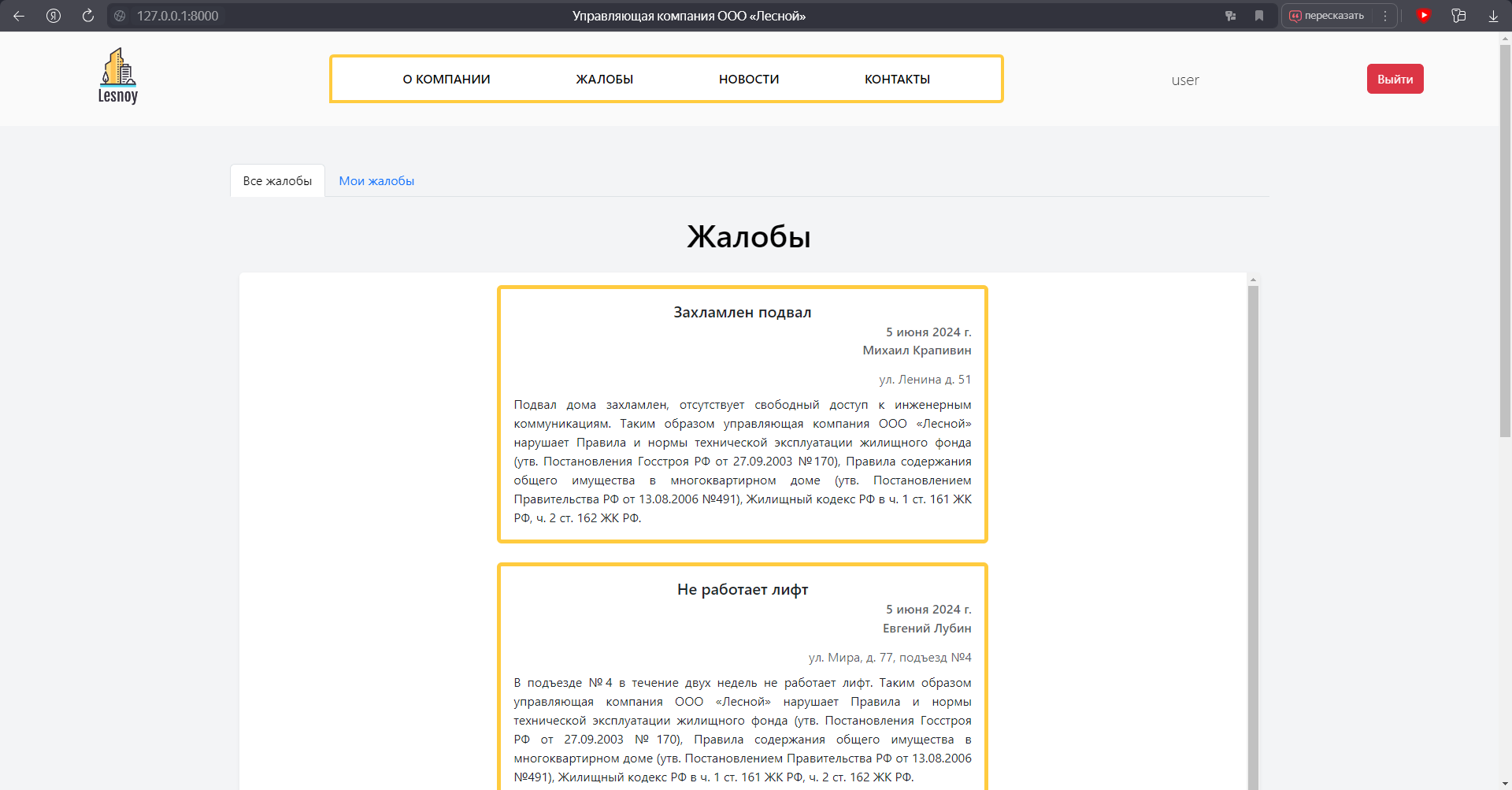


Рисунок 4.5 – Список всех жалоб

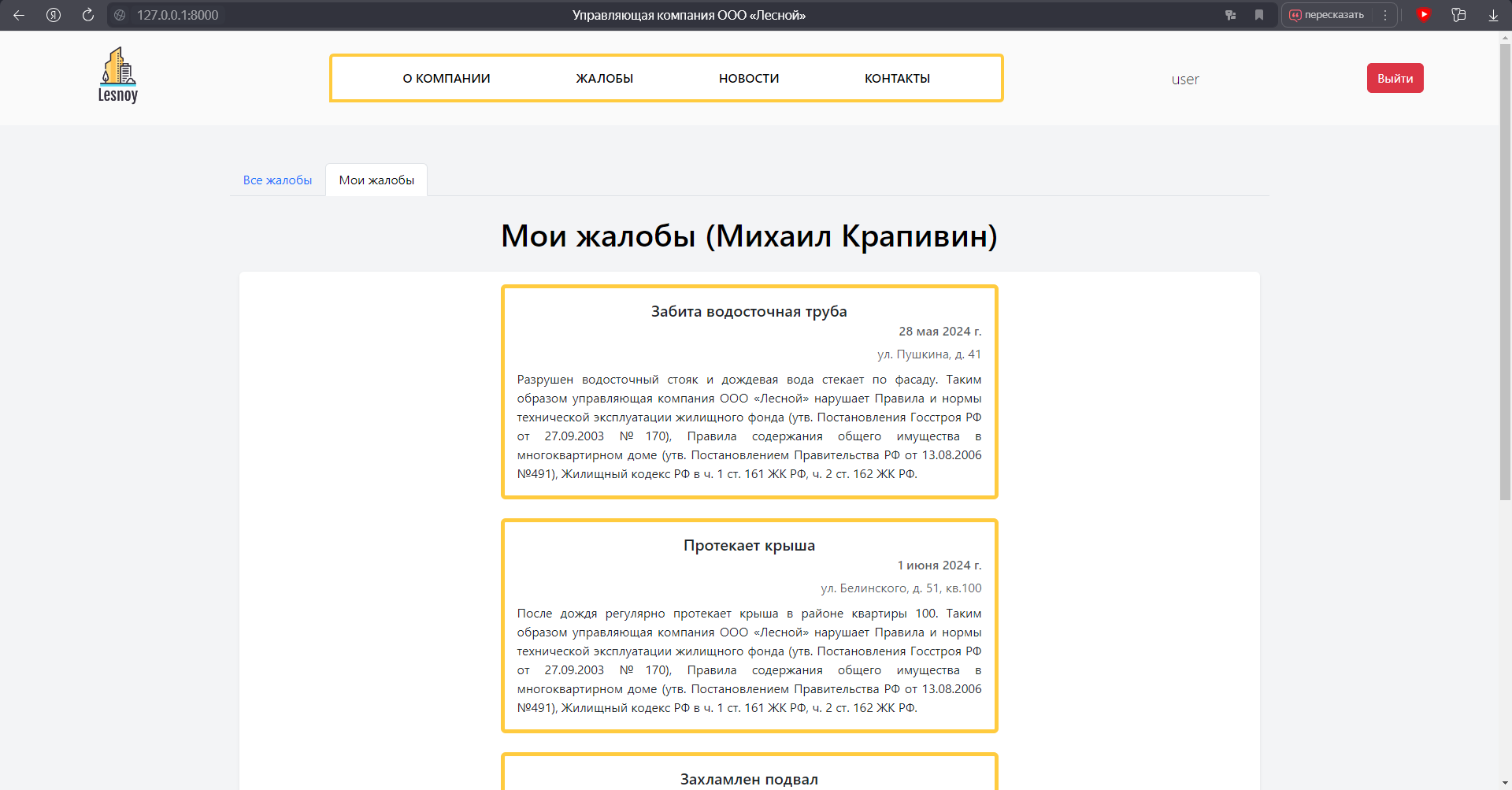


Рисунок 4.6 – Список жалоб пользователя

Чтобы увидеть список жалоб, нужно нажать на кнопку «Жалобы», находящуюся на навигационной панели. На странице жалоб пользователь может выбрать в навигационном меню то, что его интересует: все жалобы, поданные пользователями, или его жалобы. Жалобы оформлены в виде карточек с темой, датой подачи, адресом проживания заявителя и текстом жалобы.

1. Передача показаний счетчиков (рис. 4.7).

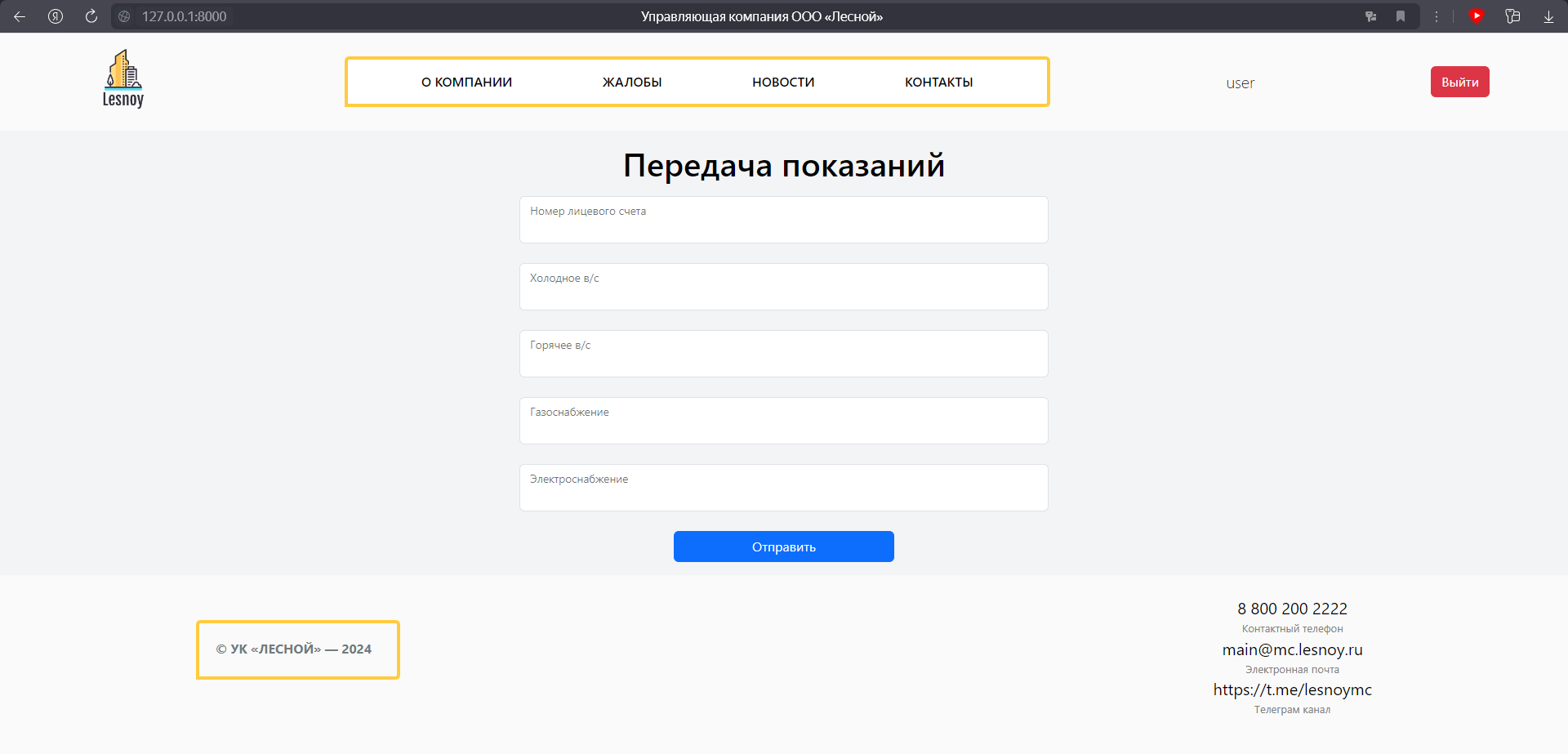


Рисунок 4.7 – Передача показаний счетчиков

На эту страницу пользователь может попасть, если нажмет на соответствующую кнопу опций. Перейдя сюда, пользователь должен ввести номер лицевого счета, состоящий из 9 цифр, и далее ввести в соответствующие поля формы показания по воде (горячей и холодной), газу и электроэнергии.

1. Новости (рис. 4.8).

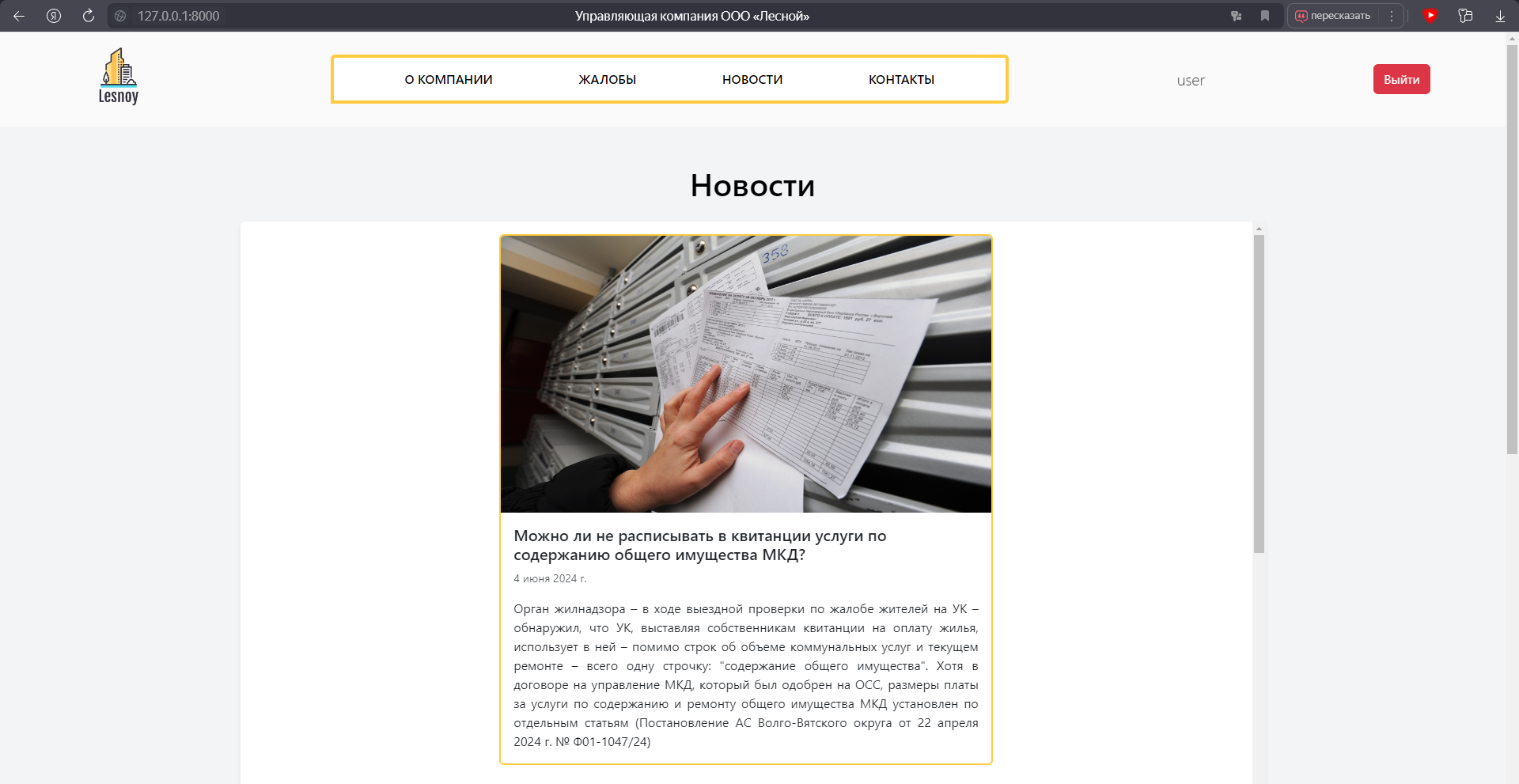


Рисунок 4.8 – Страница новостей

Помимо «Жалоб» на навигационной панели расположены «Новости». Если нажать на данную кнопку, можно увидеть список всех новостей в виде карточек с темой, датой и текстом новости.

1. Отправка сообщений в службу поддержки (рис. 4.9).

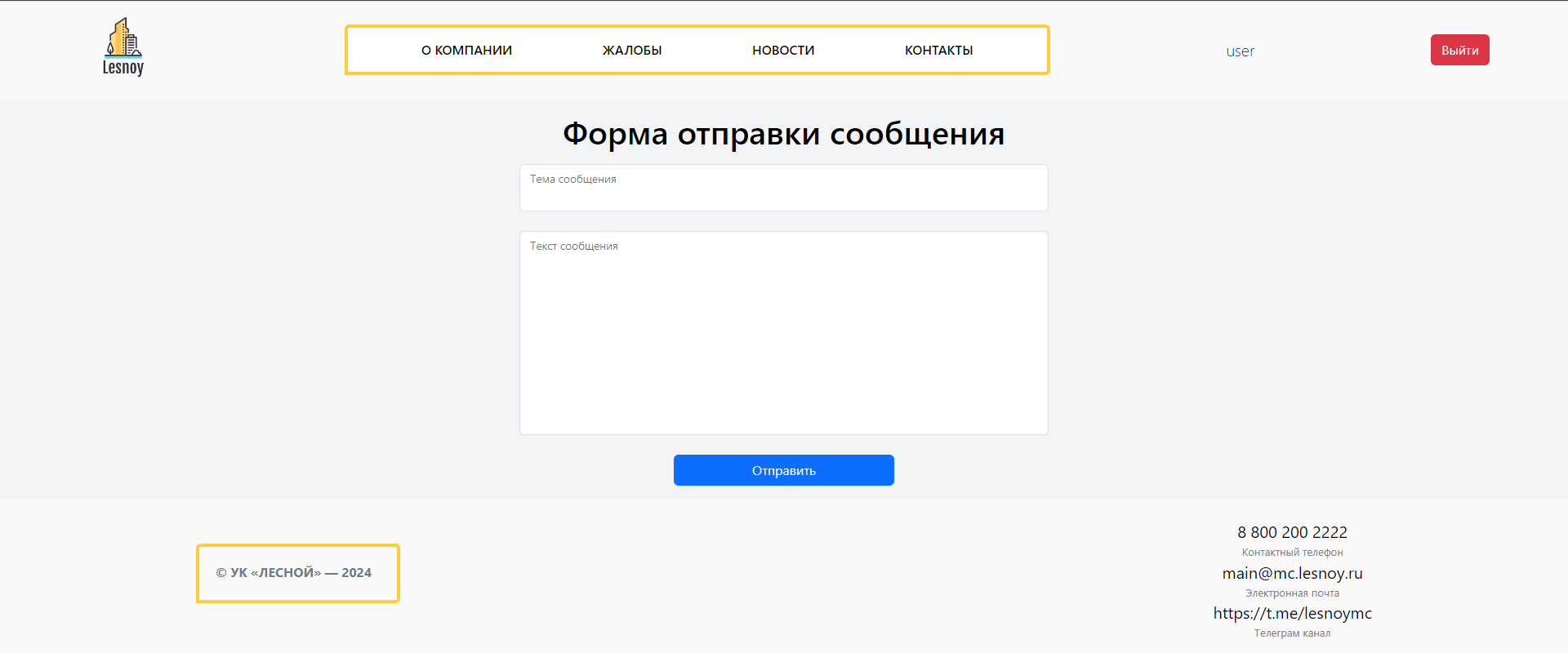


Рисунок 4.9 – Отправка сообщений в службу поддержки

Чтобы перейти на страницу отправки сообщений, нужно нажать на аналогичную кнопку на блоке с анимацией. Здесь пользователь вводит тему своего вопроса и сам вопрос.

1. Список личных сообщений (рис. 4.10).

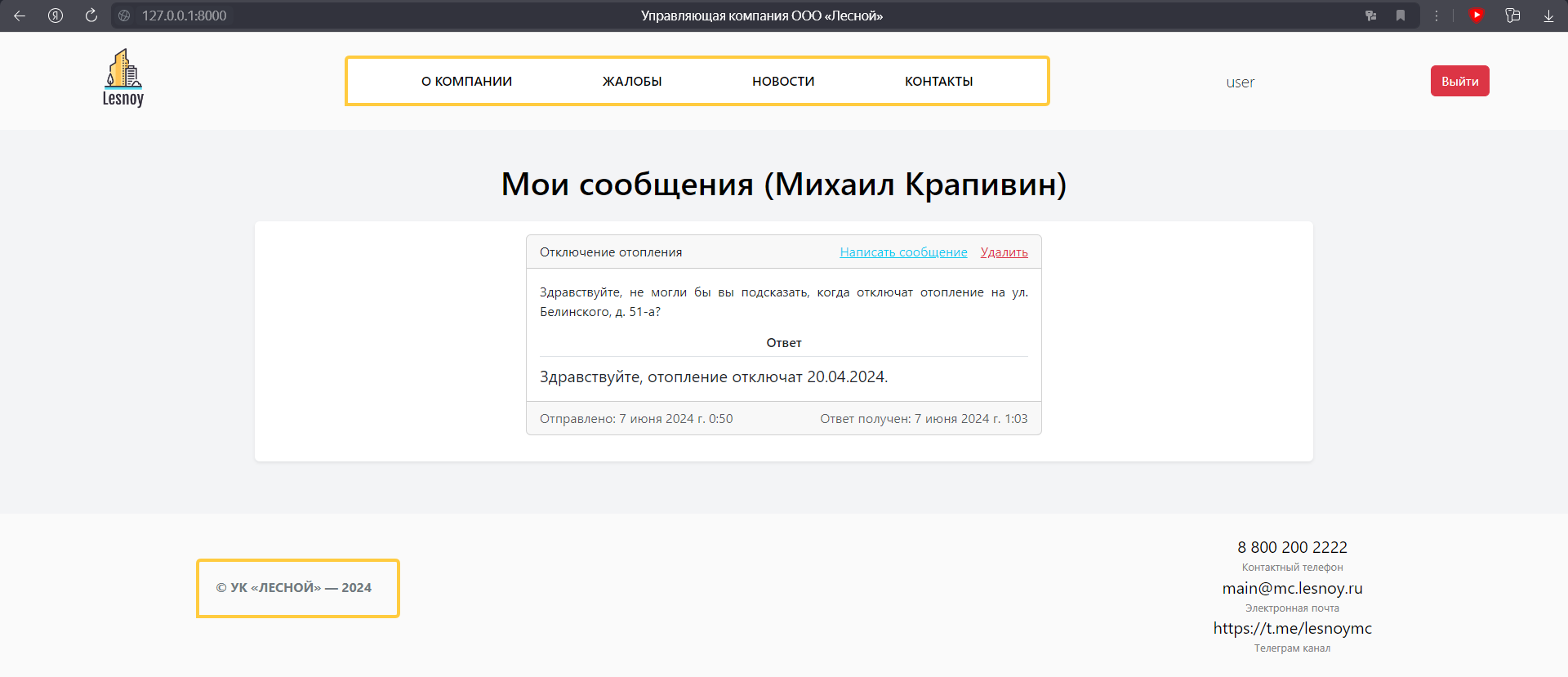


Рисунок 4.10 – Список сообщений

После авторизации справа от навигационного меню появится кнопка «Сообщения», при нажатии на которую откроется список сообщений, являющихся ответами службы поддержки.

1. Сообщения, полученные службой поддержки (рис. 4.11).

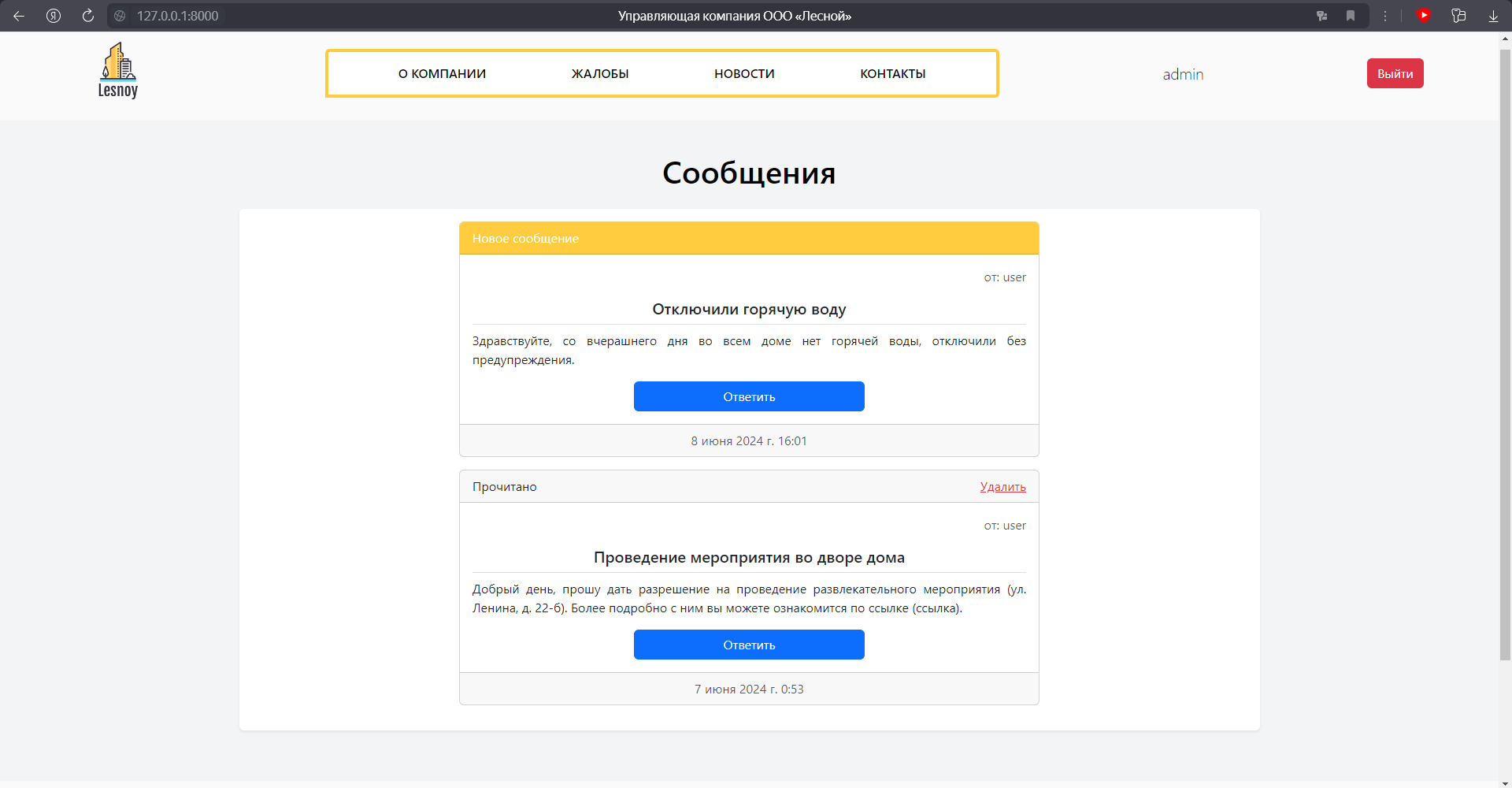


Рисунок 4.11 – Список сообщений (служба поддержи)

Для того, чтобы можно было получать ответы на сообщения, были добавлены дополнительные страницы, доступные службе поддержки (администраторам сайта). Если зайти на сайт под пользователем admin, то на странице «Сообщения» будут отображаться сообщения, присланные пользователями. При нажатии на кнопку ответить, admin переходит на страницу ответа. (рис. 4.12):



Рисунок 4.12 – Ответ на сообщение (служба поддержи)

После ввода текста сообщения и нажатия на кнопку «Отправить», данное сообщение удалится из списка (запись о нем в базе данных останется).

# 5 Разработка клиентской части приложения

Проект включает несколько веб-страниц, каждая из которых имеет своё назначение и свою функциональность. В этом разделе будет подробно описана клиентская часть проекта, включающая разработку интерфейса, назначение каждого блока, его код, реализацию адаптивности и валидацию форм.

## 5.1 index.html

Страница «index.html» является главной страницей веб-приложения компании. Она содержит основную информацию о компании, контактные данные и ссылки на другие страницы сайта.

1. Заголовок и подключение стилей (рис. 5.1.1).

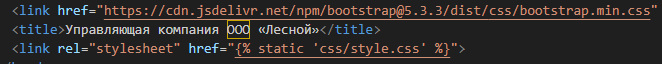
****

Рисунок 5.1.1 – Заголовок и подключение стилей

Эти строки задают заголовок окна страницы и подключают фреймворк Bootstrap и файл стилей проекта, что обеспечивает стилизацию и адаптивность страницы.

1. Шапка страницы и навигационная панель (рис. 5.1.2).

Рисунок 5.1.2 – Шапка страницы и навигационная панель

В этом блоке расположена навигация по сайту. В зависимости от статуса пользователя (авторизован или нет) показываются различные ссылки.

1. Блок авторизации (рис. 5.1.3).



Рисунок 5.1.3 – Блок авторизации

Здесь реализована логика отображения кнопок для входа, регистрации или выхода из аккаунта в зависимости от состояния пользователя. Также реализовано отображение количества непрочитанных сообщений (кружок с цифрой в правой верхней части кнопки «Сообщения»).

1. Основной блок сайта (рис. 5.1.4).

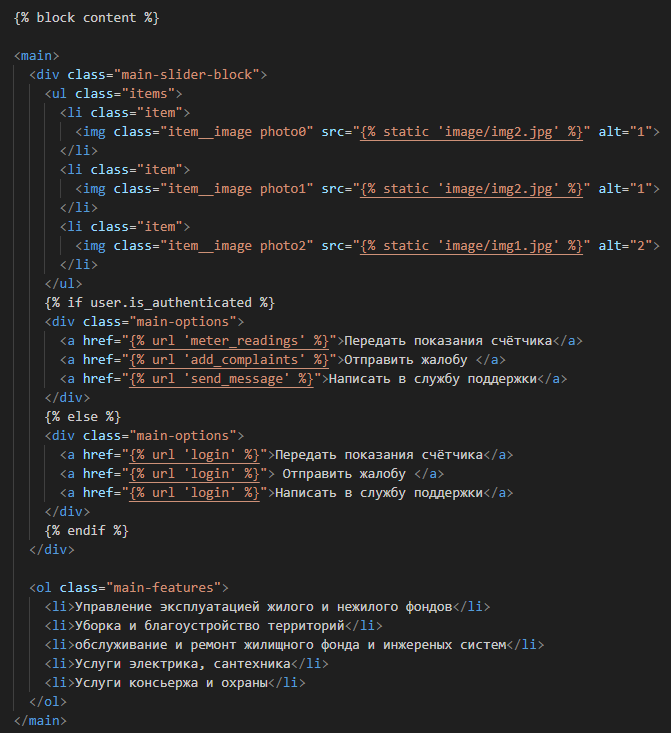


Рисунок 5.1.4 – Основной блок сайта

Основной контент включает блок с анимацией, блок с основными действиями для пользователей (передача показаний, отправка жалоб и связь со службой поддержки) и список услуг компании.

1. Раздел «О компании» (рис. 5.1.5).

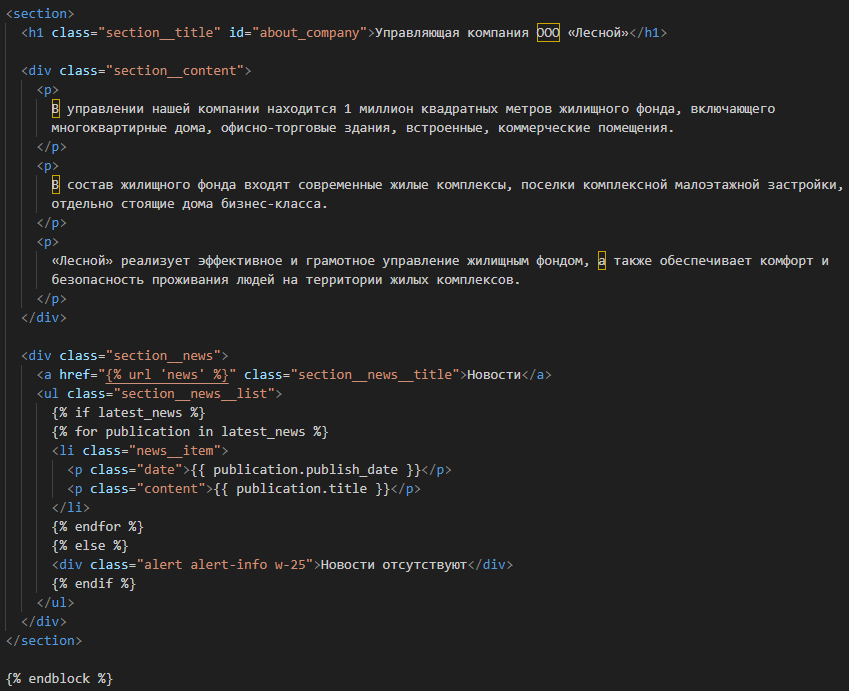


Рисунок 5.1.5 – Раздел «О компании»

Это раздел с описанием компании, ее сферы деятельности и подхода к управлению. В этом разделе также расположен блок новостей, который динамически обновляется (в нем отображаются 3 последних новости).

1. Подвал страницы (рис. 5.1.6).

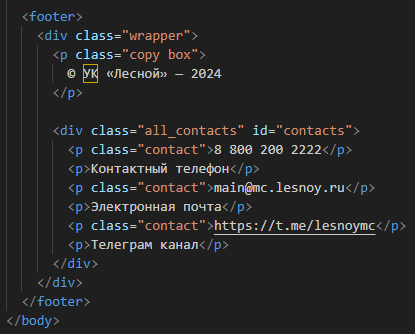


Рисунок 5.1.6 – Подвал страницы

В этом блоке указаны контактные данные компании.

**Реализация адаптивности**

Адаптивность данной страницы была реализована по принципу Desktop First. Она была настроена с помощью медиазапросов в CSS. Адаптивность позволяет странице корректно отображаться на устройствах с разными размерами экранов. В медиазапросах изменяются свойства некоторых CSS классов для устройств с шириной экрана в диапазонах: от 1400px до 1200px, от 1200px до 992px, от 992px до 768px, от 768px до 576px и от 576px до 0.

Например, ниже представлена часть медиазапроса для экранов с шириной, находящейся в диапазоне от 992px до 768px (рис. 5.1.7):

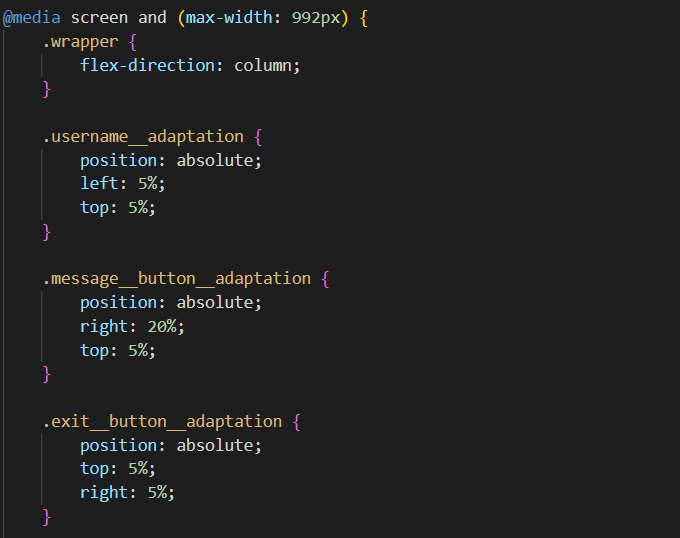
В данном случае меняется расположение элементов в шапке страницы (имя пользователя переносится в левый верхний угол, кнопка «Сообщения» и «Выйти» – в правый верхний угол).

Рисунок 5.1.7 – Пример реализации адаптивности

Большинство страниц расширяют шаблон index.html (с помощью инструкции *{% extends "index.html" %}*). Таким образом, они будут состоять из шапки, футера шаблона «index.html» и блока собственного шаблона.

Для **остальных страниц** адаптивность была настроена с помощью Bootstrap классов «container», «col-\*», «col-md-\*», «col-lg-\*», «col-xl-\*» (вместо знака \* могут стоять цифры от 1-12).

## 5.2 register.html

Страница регистрации «register.html» позволяет новым пользователям создать аккаунт в системе. Ниже приведены основные компоненты и их объяснение:

1. Заголовок и подключение стилей (рис. 5.2.1).



Рисунок 5.2.1 – Заголовок и подключение стилей в register.html

Здесь задается название и подключается фреймворк Bootstrap.

1. Ссылка на главную страницу (рис. 5.2.2).



Рисунок 5.2.2 – Ссылка на главную страницу в register.html

Добавление кнопки, которая позволяет пользователю вернуться на главную страницу сайта.

1. Основная часть (рис. 5.2.3).

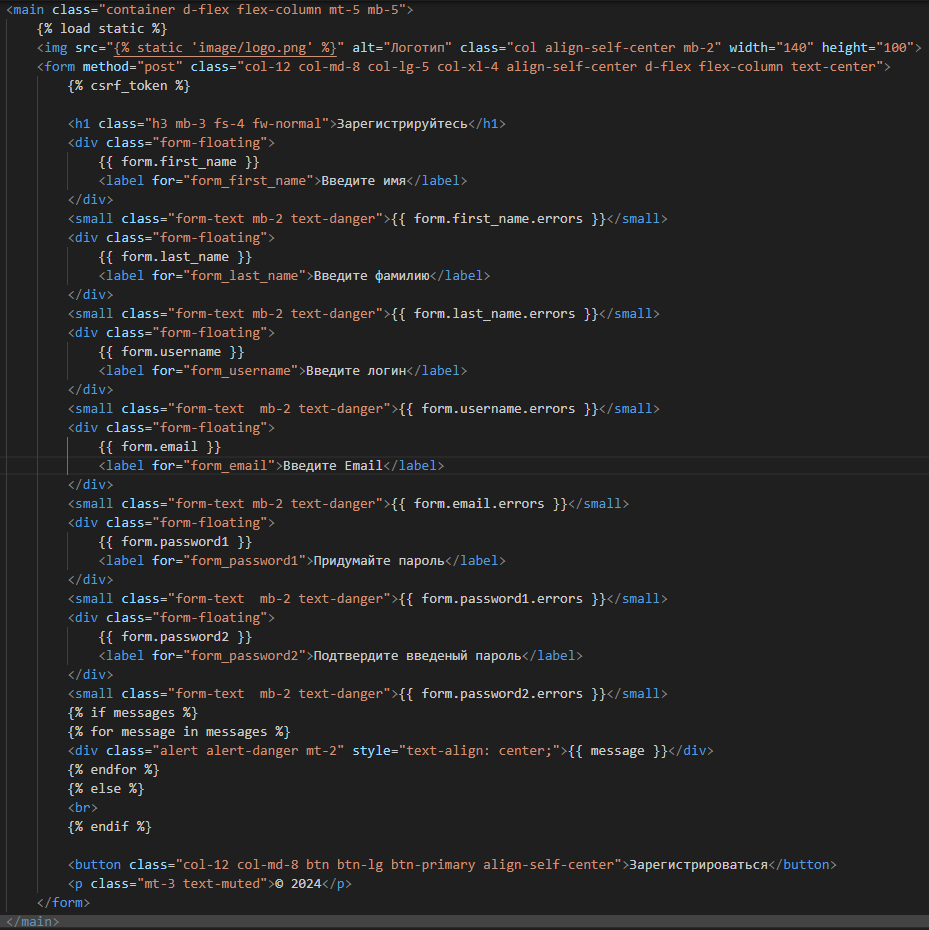


Рисунок 5.2.3 – Основная часть в «register.html»

Основная часть содержит форму регистрации с полями для ввода имени, фамилии, логина, email, пароля и его подтверждения. Также выводятся сообщения об ошибках, если есть ошибки.

## 5.3 login.html

Страница авторизации «login.html»позволяет уже зарегистрированным пользователям входить в систему. Она включает следующие компоненты:

1. Заголовок и подключение стилей (рис. 5.3.1).



Рисунок 5.3.1 – Заголовок и подключение стилей в login.html

Здесь задается название и подключается фреймворк Bootstrap.

1. Ссылка на главную страницу (рис. 5.3.2).



Рисунок 5.3.2 – Ссылка на главную страницу в login.html

1. Основная часть (рис. 5.3.3).

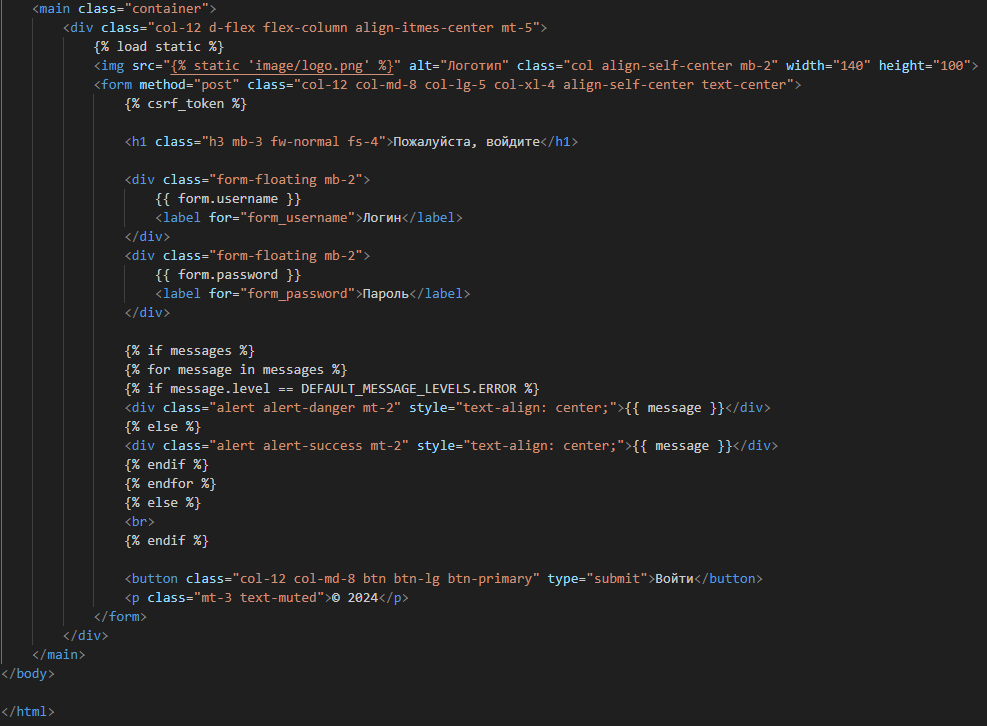


Рисунок 5.3.3 – Основная часть в login.html

в login.html

Основной блок содержит форму для ввода логина и пароля, а также отображает сообщения об ошибках или успешном входе.

## 5.4 send\_complaint.html

Эта страница позволяет пользователю отправить жалобу, заполнив форму с необходимыми полями (рис. 5.4.1):

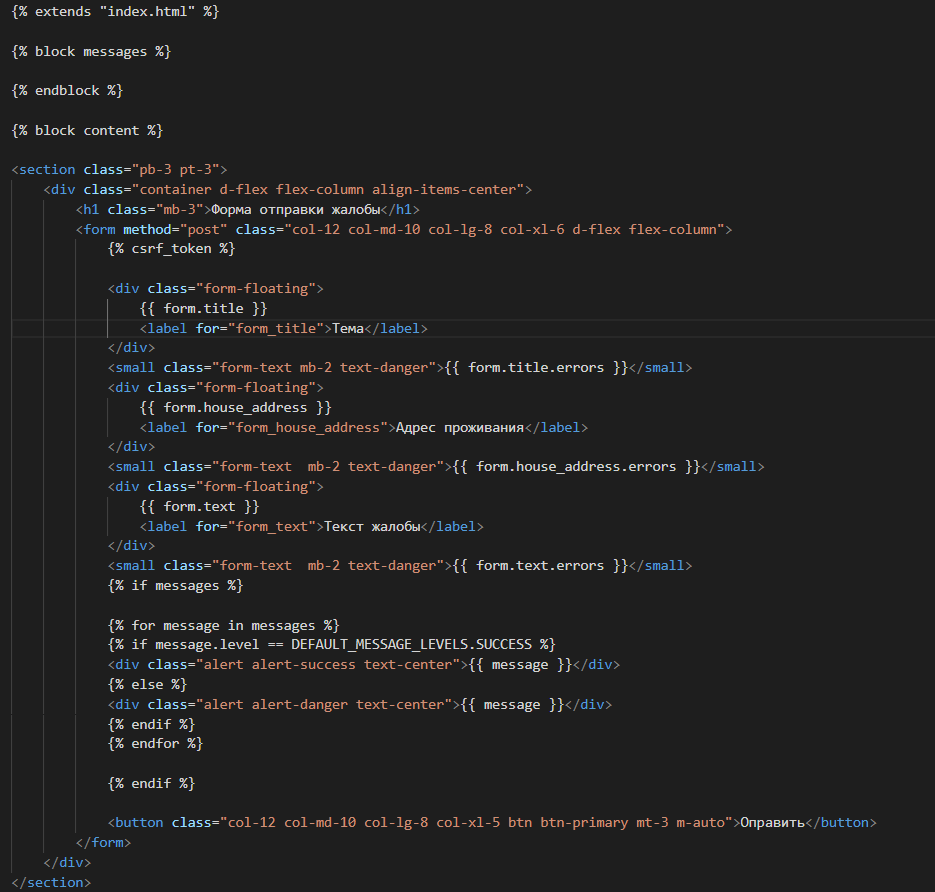
**Основные элементы страницы**:

Рисунок 5.4.1 – send\_complaint.html

1. Заголовок страницы.
2. Поля для ввода темы, адреса проживания и текста жалобы.
3. Сообщения об ошибках под соответствующими полями.
4. Сообщения об успешной или неудачной отправке жалобы.

## 5.5 complaints.html

Эта страница отображает все жалобы, которые были отправлены пользователями (рис. 5.5.1).

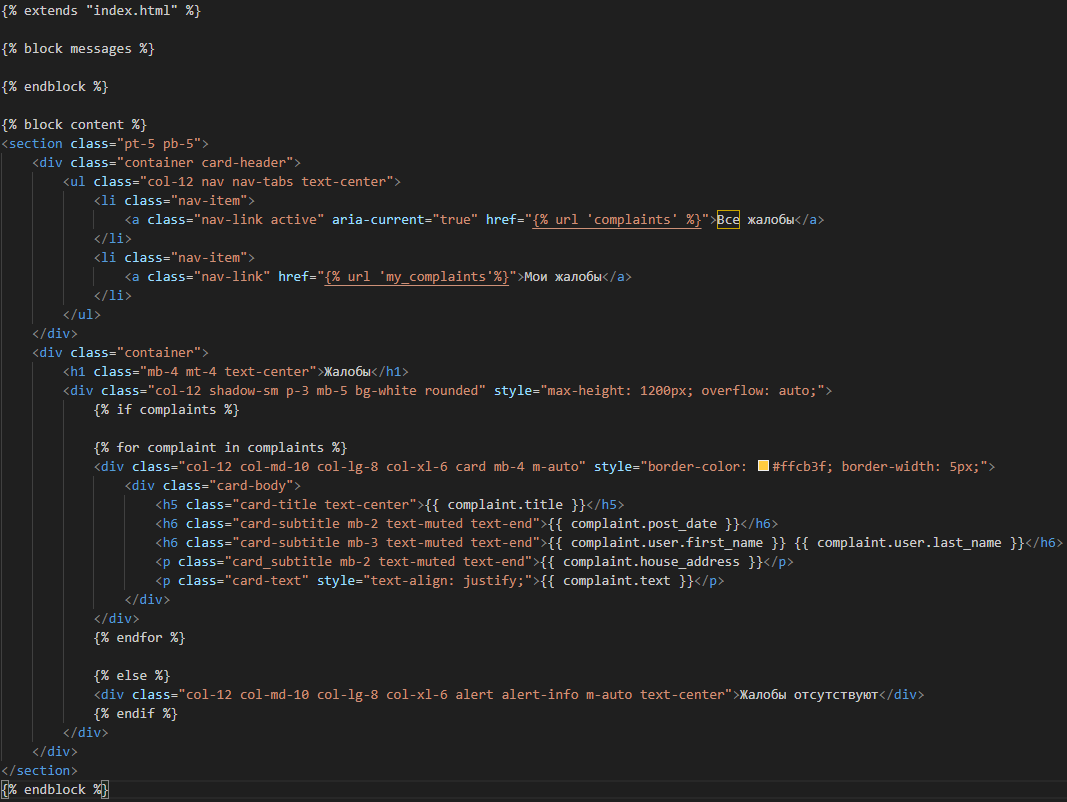


Рисунок 5.5.1 –complaints.html

в login.html

**Основные элементы страницы**:

1. Навигационные вкладки: «Все жалобы» (текущая страница) и «Мои жалобы» (Ссылка на страницу с жалобами текущего пользователя)
2. Список жалоб: Отображение всех жалоб в виде карточек с подробной информацией (заголовок, дата, имя и фамилия, адрес проживания, текст жалобы).

## 5.6 my\_complaints.html

Эта страница отображает жалобы, отправленные текущим пользователем (рис. 5.6.1).

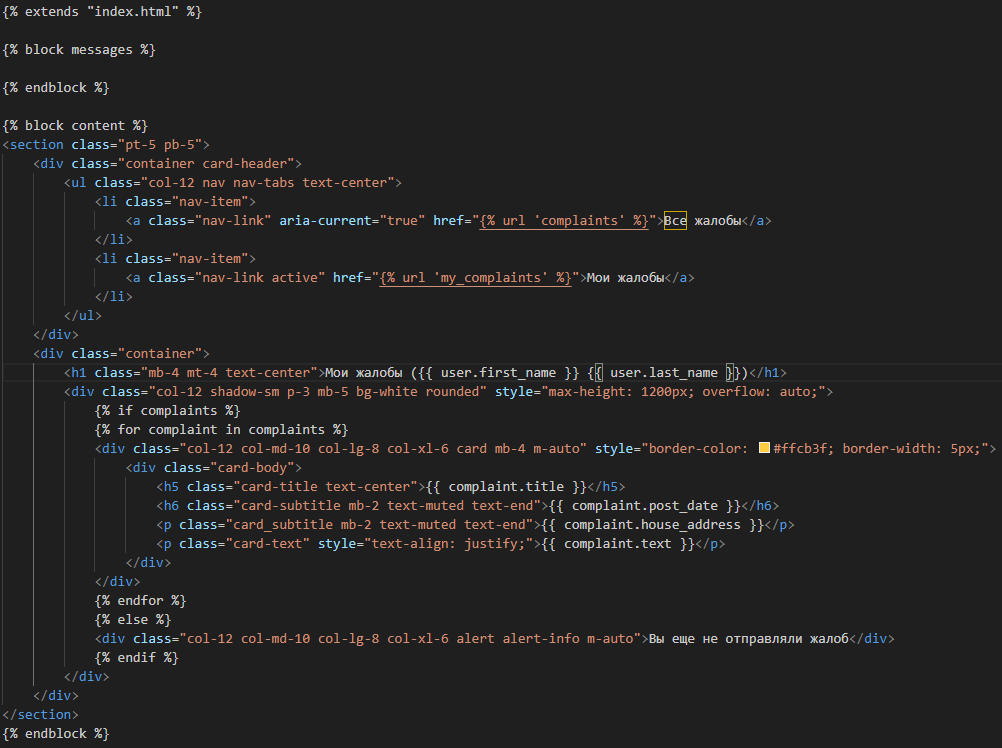


Рисунок 5.6.1 –my\_complaints.html

в login.html

**Основные элементы страницы**:

1. Навигационные вкладки: «Все жалобы» (Ссылка на страницу со всеми жалобами) и «Мои жалобы» (текущая страница)
2. Список жалоб: Отображение личных жалоб в виде карточек с подробной информацией (заголовок, дата, адрес проживания, текст жалобы).

## 5.7 send\_meterreadings.html

Эта страница позволяет пользователям ввести номер лицевого счета и передать показания счетчиков холодного и горячего водоснабжения, газа и электроэнергии (рис. 5.7.1).

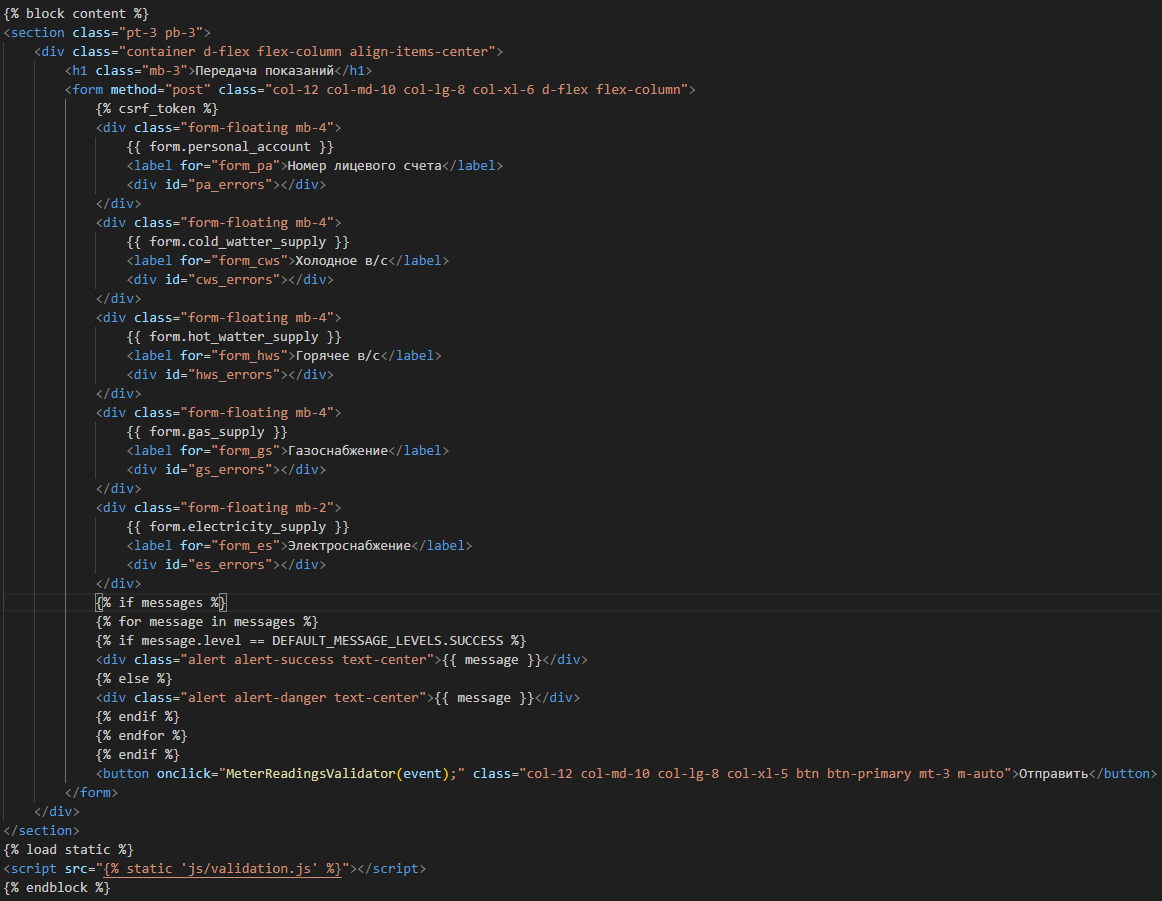


Рисунок 5.7.1 – send\_meterreadings.html

в login.html

**Основные элементы страницы**:

1. Заголовок страницы
2. Форма: содержит поля для ввода номера лицевого счета, показаний холодного и горячего водоснабжения, газо- и электроснабжения.
3. Сообщения об успехе или ошибке
4. Кнопка отправки показаний
5. Подключение валидации

**Валидация** (рис. 5.7.2).

Для того, чтобы не писать много раз один и тот же код, была создана функция «regexCheck()» (рис. 5.7.3).

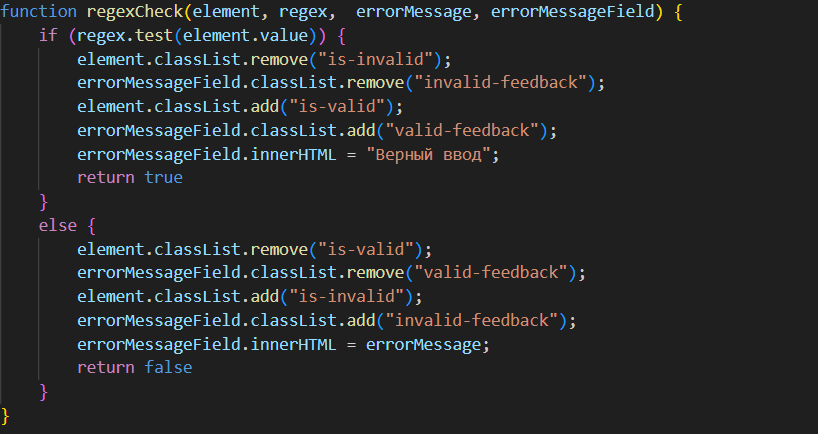
В ней проверяется соответствие введенной пользователем строки регулярному выражению. Если строка соответствует выражению, то возвращается «true», добавляются и удаляются некоторые классы элемента, вставляется надпись «Все верно». Если строка не соответствует регулярному выражению, то возвращается «false», также добавляются и удаляются некоторые классы элемента, вставляется уведомление об ошибке.

Рисунок 5.7.3 – Функция regexCheck()



Рисунок 5.7.2 – Валидация формы в send\_meterreadings.html

в login.html

Здесь устанавливаются условия чтобы:

1. Номер лицевого счета (personal\_account) содержал девять цифр (0-9).
2. Показание по холодному водоснабжению (cold\_watter\_supply), горячему водоснабжению (hot\_watter\_supply), газоснабжению (gas\_supply) и электроснабжению (electricity\_supply) были целыми числами (0-99999).

## 5.8 send\_message.html

Эта страница позволяет пользователям отправлять сообщения в службу поддержки, заполнив форму с необходимыми полями (рис. 5.8.1).

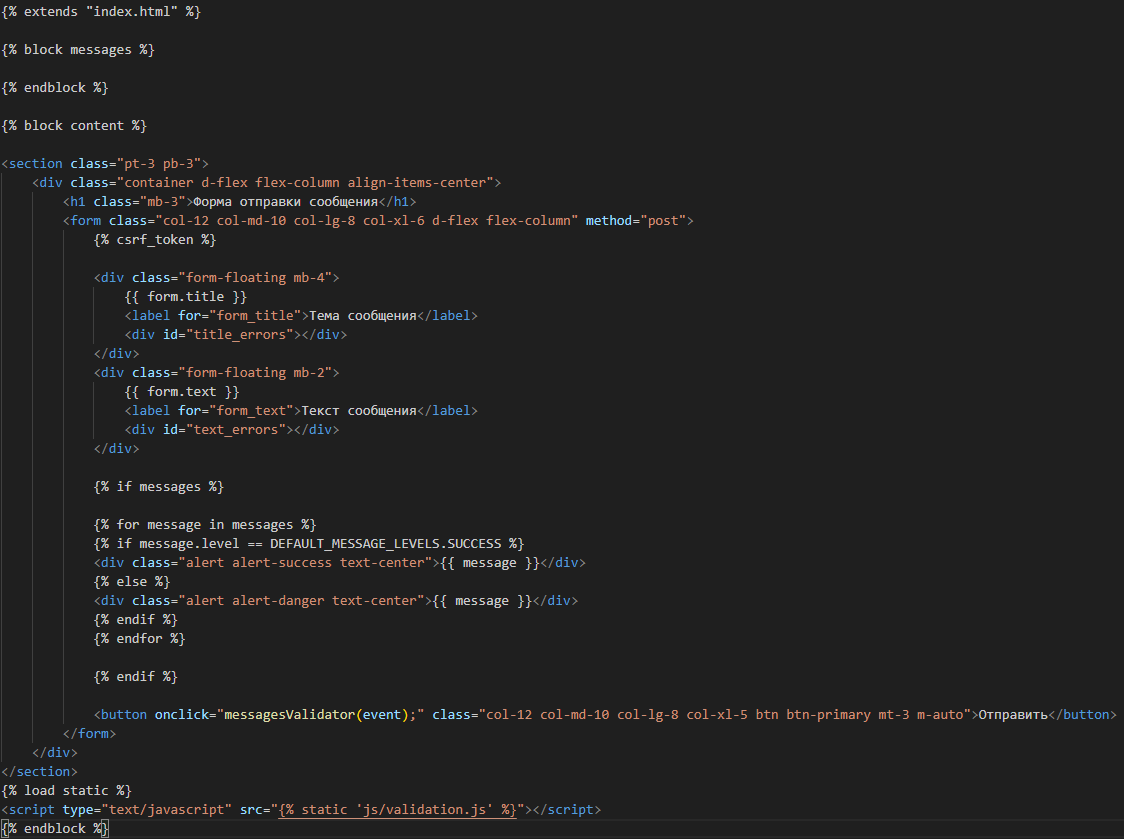


Рисунок 5.8.1 – send\_message.html

в login.html

**Основные элементы страницы**:

1. Заголовок страницы.
2. Форма: два основных поля title (тема сообщения) и text (текст сообщения).
3. Сообщения об успехе или ошибке.
4. Кнопка отправки сообщения.
5. Подключение валидации.

**Валидация** (рис. 5.8.2).

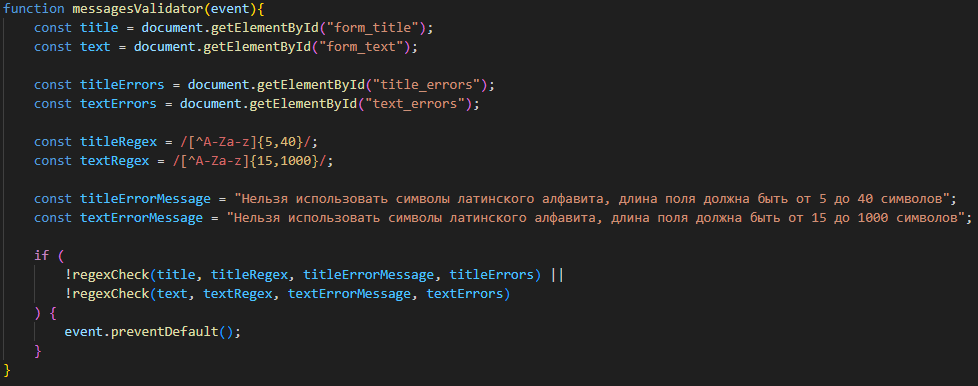


Рисунок 5.8.2 – Валидация формы в send\_message.html

в login.html

Здесь устанавливаются условия чтобы:

1. В теме сообщения (title) не использовались символы латинского алфавита, и длина сообщения была от 5 до 40 символов.
2. В тексте сообщения (text) не использовались символы латинского алфавита, и длина сообщения была от 15 до 1000 символов.

## 5.9 user\_messages.html

Эта страница отображает список сообщений, полученных пользователем от службы поддержки в качестве ответа на посланные им сообщения (рис. 5.9.1).

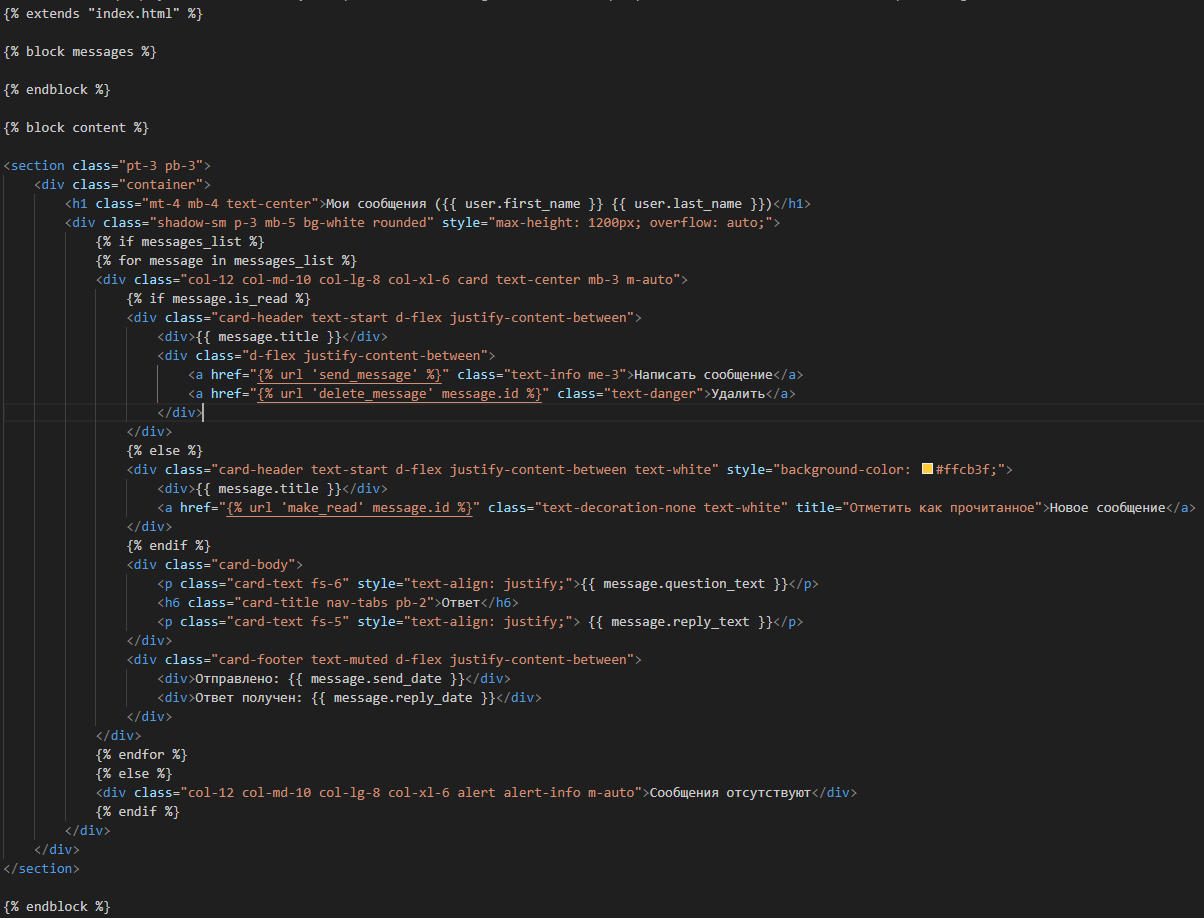


Рисунок 5.9.1 – user\_message.html

в login.html

**Основные элементы страницы**:

1. Заголовок.
2. Список сообщений:
   1. Сообщения отображаются в виде карточек.
   2. Прочитанные сообщения имеют обычный стиль заголовка, а непрочитанные выделены желтым цветом.
   3. Каждое сообщение содержит заголовок, текст вопроса, текст ответа, дату отправки и дату получения ответа.
   4. Также предусмотрены ссылки для написания нового сообщения и удаления текущего сообщения.
3. Отображение отсутствия сообщений (Если сообщений нет, отображается уведомление об их отсутствии).

## 5.10 admin\_messages.html

Эта страница позволяет службе поддержки просматривать и управлять сообщениями пользователей (рис. 5.10.1).

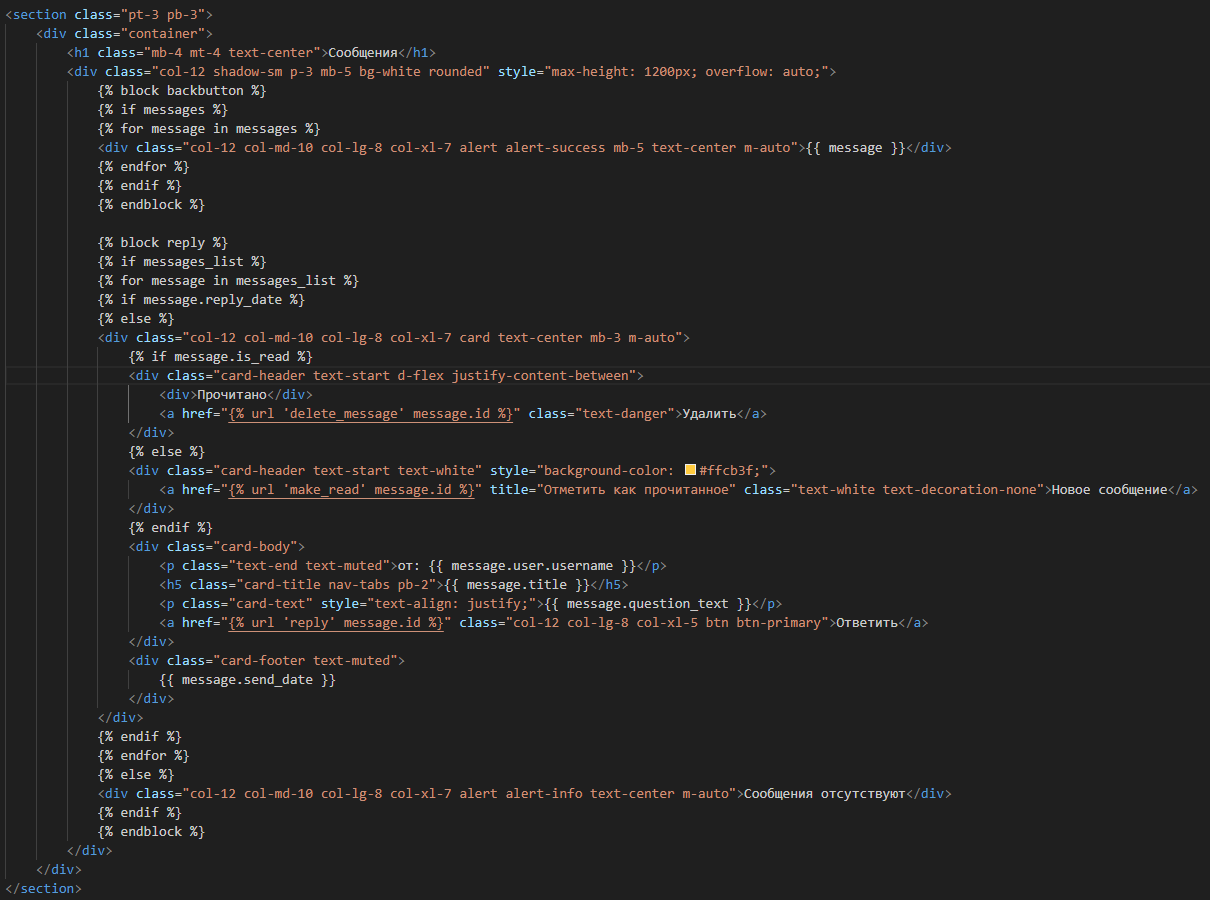


Рисунок 5.10.1 – admin\_messages.html

в login.html

**Основные элементы страницы**:

1. Заголовок.
2. Если имеются сообщения:
   1. Сообщения пользователей отображаются в виде карточек
   2. Сообщения, помеченные как непрочитанные, выделены желтым цветом
   3. Прочитанные сообщения можно удалить.
   4. Кнопка «Ответить» ведет на страницу ответа для соответствующего сообщения.
3. Если сообщений нет, отображается уведомление об их отсутствии

## 5.11 reply.html

Эта страница позволяет администратору ответить на сообщение пользователя (рис. 5.11.1).

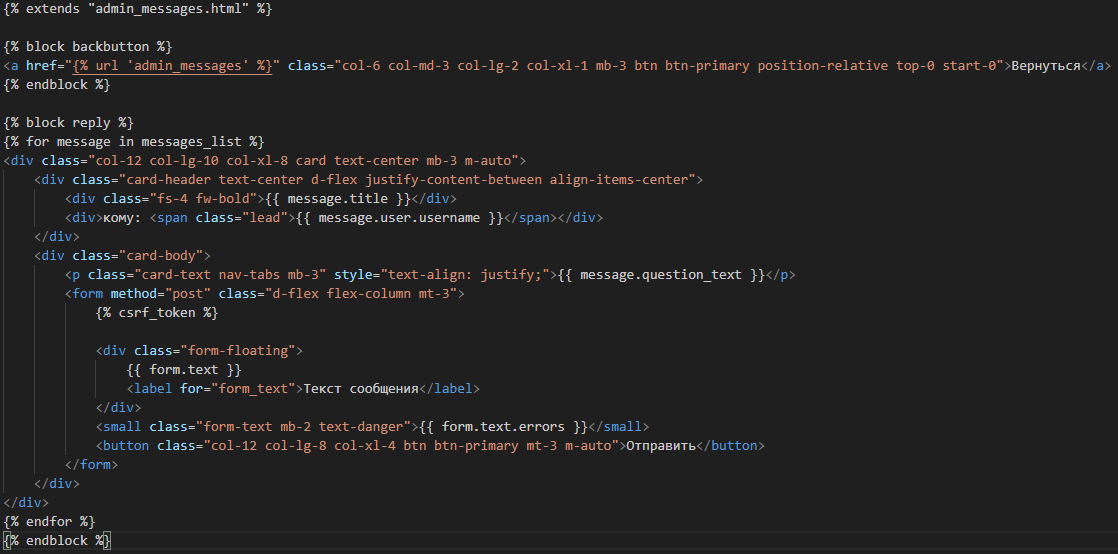


Рисунок 5.11.1 – reply.html

в login.html

**Основные элементы страницы**:

1. Кнопка возвращения на страницу со списком сообщений.
2. Сообщение, выбранное для ответа (содержит заголовок, имя пользователя и текст вопроса, а также форму, содержащую текстовое поле для ввода ответа и кнопку для отправки формы).

## 5.12 news.html

Эта страница отображает новости компании (рис. 5.12.1).

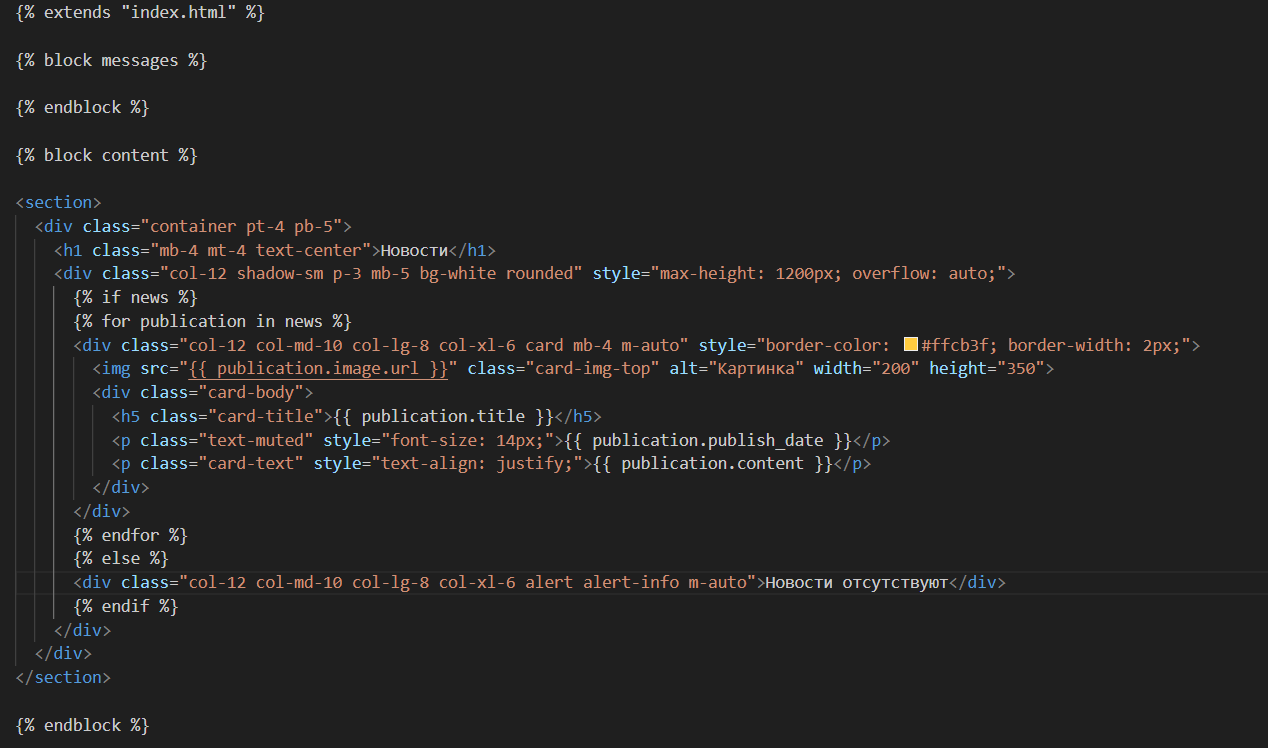
**Основные элементы страницы**:

Рисунок 5.12.1 – news.html

1. Заголовок.
2. Список новостей. Новости отображаются в виде карточек с заголовком, изображением, датой публикации и содержанием.
3. Если новостей нет, отображается уведомление об их отсутствии.

# 6 Разработка серверной части приложения

Серверная часть приложения реализована на базе Django Framework.

## 6.1 Структура проекта

Структура каталога проекта представлена на рисунке 6.1.1.

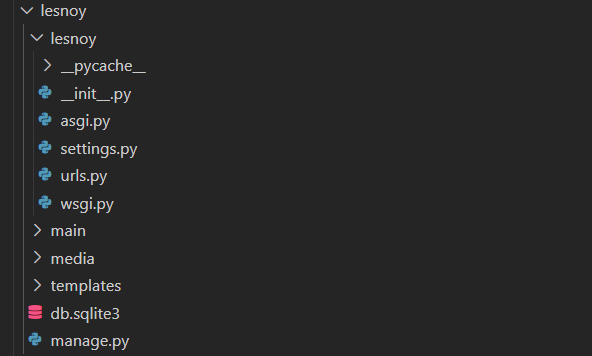
В каталоге «lesnoy» находятся конфигурационные файлы проекта, определяющие его работу. Среди них можно выделить файл «settings.py», в котором указаны основные настройки проекта в виде переменных Python, и файл «urls.py» (рис. 6.1.2), содержащий маршруты проекта (пути к его частям).

Рисунок 6.1.1 – Структура проекта

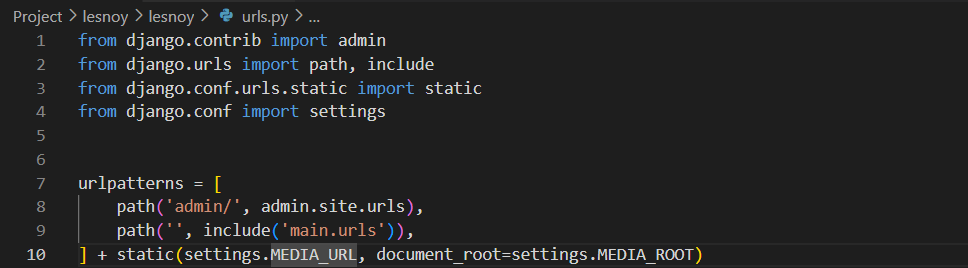
Во втором маршруте указано (с помощью функции «include»), что настройка маршрутов, ответвляющихся от него, находится в файле «urls.py» каталога приложения «main» (рис. 6.1.3).

Рисунок 6.1.2 – Файл «urls.py»

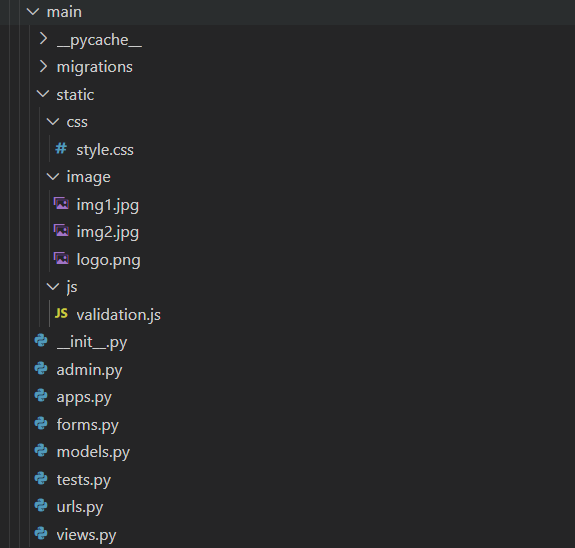
В данном каталоге за взаимодействие с базой данных отвечают файл «models.py», в котором с помощью классов, их полей и методов создаются таблицы в базе, и каталог «migrations» (также команды «makemigrations», «migrate»), отвечающий за распространение изменений в файле «models.py» в схему базы данных.

Рисунок 6.1.3 – Структура приложения «main»

За логику работы процессов в приложении отвечают представления, представляющие функции Python и хранящиеся в файле «views.py».

За настройку части сайта (путь «/admin»), предназначенной для создания, просмотра, обновления и удаления записей в таблицах, отвечает файл «admin.py». В нем, например, можно указать модели, которые будут отображаться в данной части сайта (рис. 6.1.4).

Статические файлы (CSS, JS) и изображения хранятся в каталоге «static». В шаблонах к ним можно обратиться с помощью инструкции «{% load static %}» и конструкции «{% static ‘<относительный путь к файлу>’ %}».

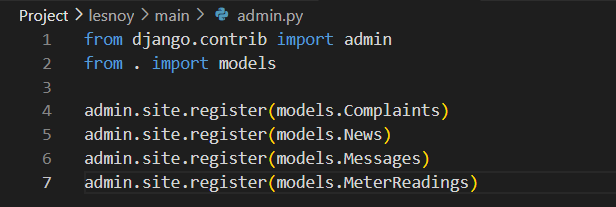
За настройку структуры форм (форма регистрации, авторизации, отправки сообщений и т. д.) на сайте отвечает файл «forms.py».

Рисунок 6.1.4 – Файл «admin.py»

Также стоит упомянуть каталоги «templates» и «media» (рис. 6.1.5), в которых находятся шаблоны и изображения, использующиеся во вкладке новости, соответственно.

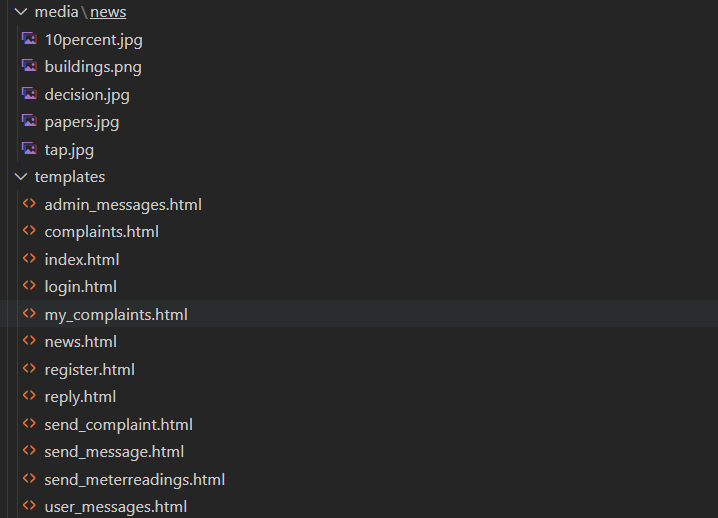
Файл «db.sqlite3» представляет из себя базу данных.

Рисунок 6.1.5 – Содержимое каталогов «templates» и «media»

## 6.2 База данных

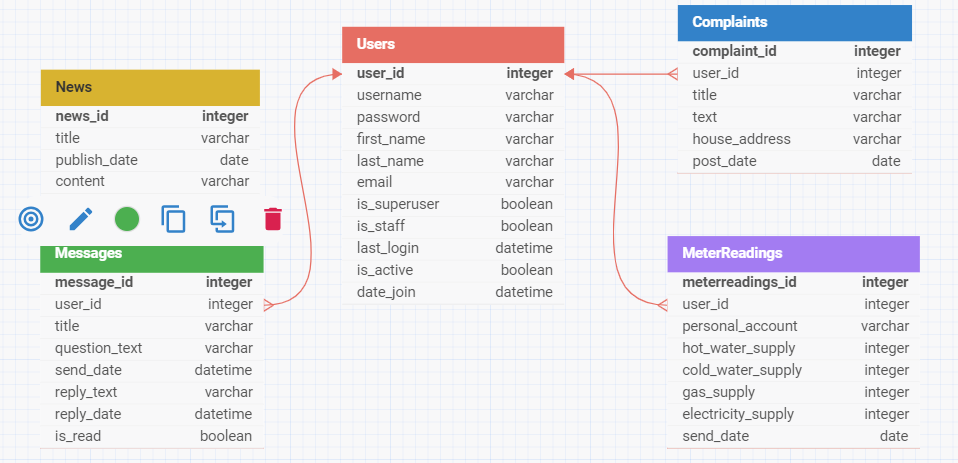
В качестве СУБД использовался SQLite. Схема базы данных представлена на рисунке 6.2.1.

Рисунок 6.2.1 – Схема базы данных

В данной схеме 3 таблицы имеют связь «многие к одному» с таблицей «Users» (в качестве внешнего ключа выступает атрибут «user\_id»).

В таблице «Users» хранятся данные о зарегистрированных пользователях, например, идентификатор пользователя (user\_id), имя пользователя (username), его пароль (password), его имя и фамилия (first\_name и last\_name), адрес электронной почты (email) и т. д. Обращение к ней происходит в процессе регистрации, аутентификации. Данная таблица не создается вручную посредством добавления модели, Django Framework создает ее автоматически.

В таблице «Complaints» хранятся данные о жалобах пользователей: идентификатор жалобы (complaint\_id), тема жалобы (title), содержание жалобы (text), дата ее подачи (post\_date), адрес проживания пользователя (house\_address), подавшего ее, его идентификатор (user\_id). Обращение к ней происходит в процессе подачи жалобы и просмотра списка жалоб. Модель, устанавливающая структуру таблицы «Complaints», представлена на рисунке 6.2.2.

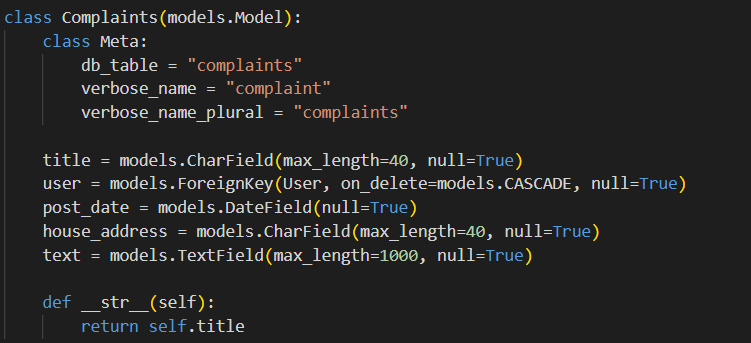
В таблице «MeterReadings» хранится информация о переданных показаниях счетчиков. Структура таблицы: идентификатор показания (meterreadings\_id), номер лицевого счета (personal\_account), данные счетчиков горячей (hot\_water\_supply) и холодной (cold\_water\_supply) воды, газа (gas\_supply) и электроэнергии (electricity\_supply), идентификатор пользователя (user\_id), передавшего показания, дата передачи (send\_date). Обращение к данной таблице происходит в процессе передачи показаний. Модель, устанавливающая структуру таблицы «MeterReadings», представлена на рисунке 6.2.3.

Рисунок 6.2.2 – Модель «Complaints»

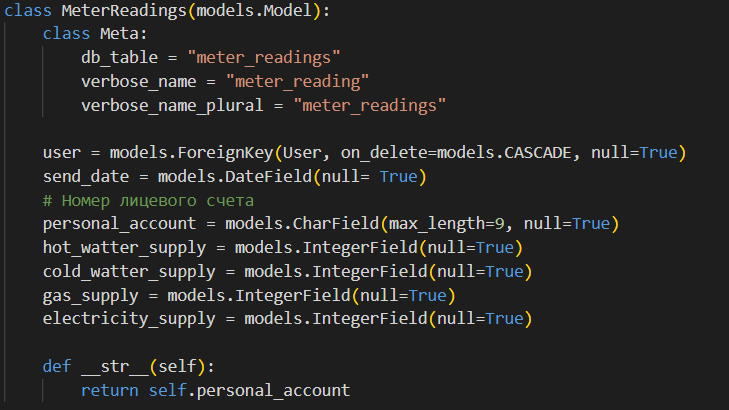
В таблице «Messages» хранятся данные о сообщениях, посылаемых в службу поддержки. Структура таблицы: идентификатор сообщения (message\_id), тема сообщения (title), текст сообщения (question\_text), дата его отправки (send\_date), текст ответа на сообщение (reply\_text), дата получения ответа на сообщение (reply\_date), флаг прочтения сообщения (is\_read), идентификатор пользователя, отправившего сообщение (user\_id). Обращение к этой таблице происходит в процессе отправки, просмотра сообщений и отправки ответов на них. Модель, устанавливающая структуру таблицы «Messages», представлена на рисунке 6.2.4.

Рисунок 6.2.3 – Модель «MeterReadings»

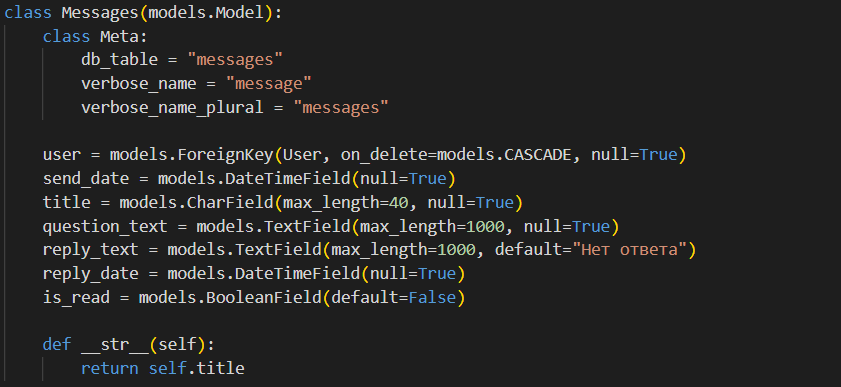
В таблице «News» хранится информация об опубликованных новостях. Структура таблицы: идентификатор новости (news\_id), ее заголовок (title) и текст (content), дата публикации (publish\_date), изображение (image). Обращение к данной таблице происходит в процессе просмотра новостей. Модель, устанавливающая структуру таблицы «News», представлена на рисунке 6.2.5.

Рисунок 6.2.4 – Модель «Messages»

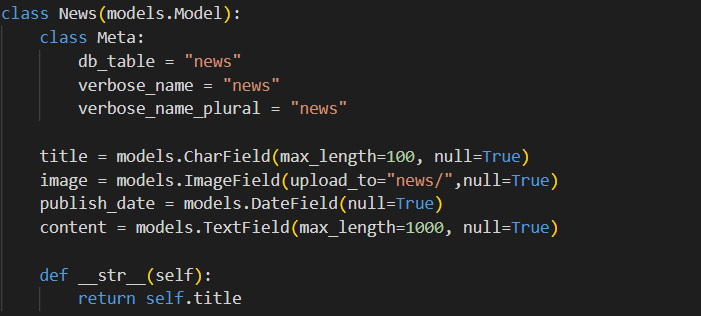
Особенностью хранения изображений является то, что при загрузке в базу данных картинки сохраняются в каталог «media» (расположение их в «media» определяется аргументом «upload\_to»), а в нее записывается абсолютный путь к ним.

Рисунок 6.2.5 – Модель «News»

## 6.3 Процессы взаимодействия сервера с клиентом

**Переход на главную страницу**

За логику процесса перехода на главную страницу отвечает представление «main\_view()» (рис. 6.3.1).

В данном представлении данные о трех последних новостях выгружаются из таблицы «News» и рассчитывается количество непрочитанных сообщений (в зависимости от типа пользователя).

Рисунок 6.3.1 – Представление «main\_view()»

Элементы главной страницы содержатся в шаблоне «index.html» (пункт 5.1). Маршрут к этой странице представлен на рисунке 6.3.2.

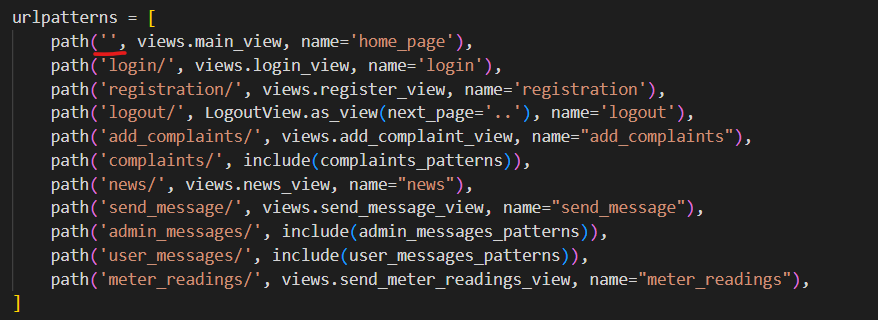
**Регистрация пользователей**

Рисунок 6.3.2 – Маршрут к главной странице

За логику процесса регистрации отвечает представление «register\_view()» (рис. 6.3.3).

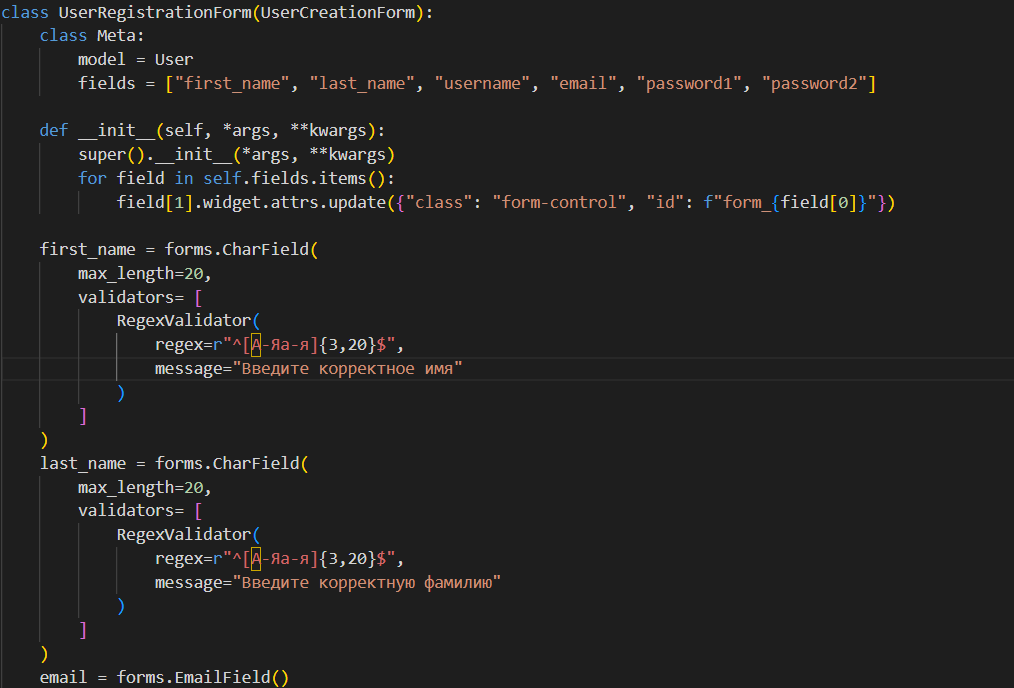
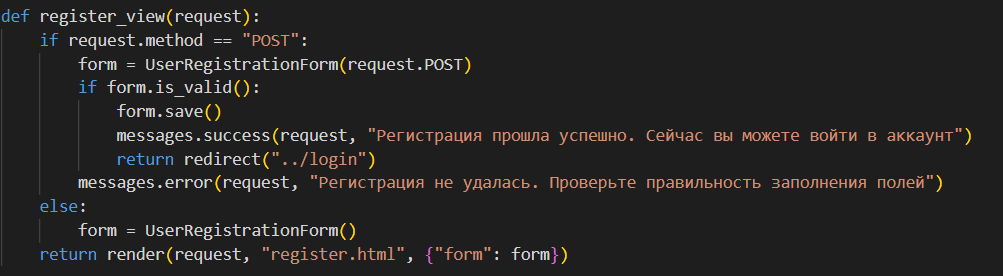
При регистрации пользователь вводит данные в форму, которая реализована с помощью класса «UserRegistrationForm()» (рис. 6.3.4).

Рисунок 6.3.4 – Класс «UserRegistrationForm»

Рисунок 6.3.3 – Представление «register\_view()»

Данные передаются на сервер с помощью HTTP POST запроса и сохраняются в базе данных «Users». Элементы интерфейса страницы регистрации содержатся в шаблоне «register.html» (пункт 5.2). Маршрут к данной странице установлен в файле «urls.py» каталога «main» (рис. 6.3.5).

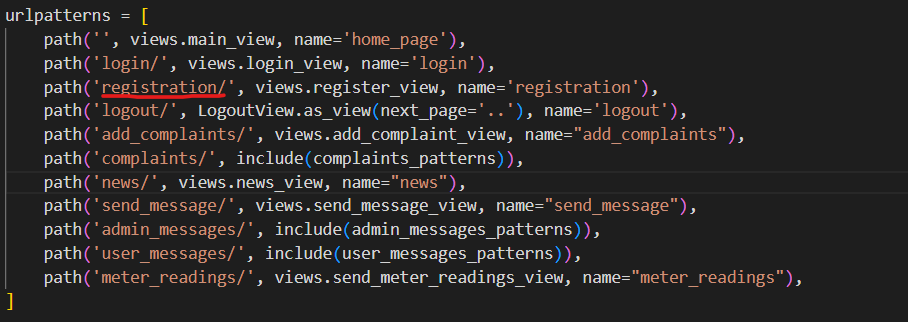
**Аутентификация пользователей**

Рисунок 6.3.5 – Маршрут к странице регистрации

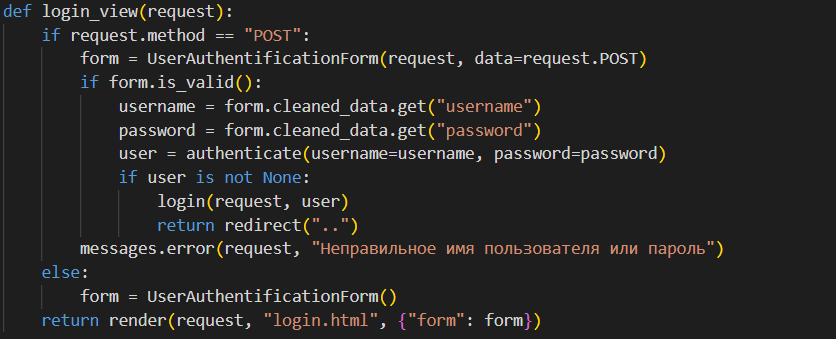
За логику процесса аутентификации отвечает представление «login\_view()» (рис. 6.3.6).

Рисунок 6.3.6 – Представление «login\_view()»

При аутентификации пользователь вводит данные в форму, структура которой устанавливается классом «UserAuthentificationForm()» (рис. 6.3.7).

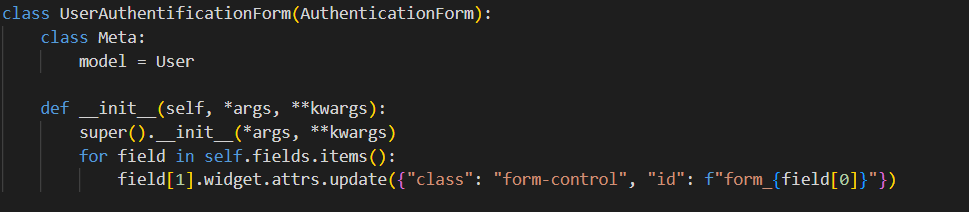
Данные передаются на сервер с помощью HTTP POST запроса и сохраняются в таблицу «User». Элементы страницы аутентификации содержатся в шаблоне «login.html» (пункт 5.3). Маршрут к данной странице представлен на рисунке 6.3.8.

Рисунок 6.3.7 – Класс «UserAuthentificationForm()»

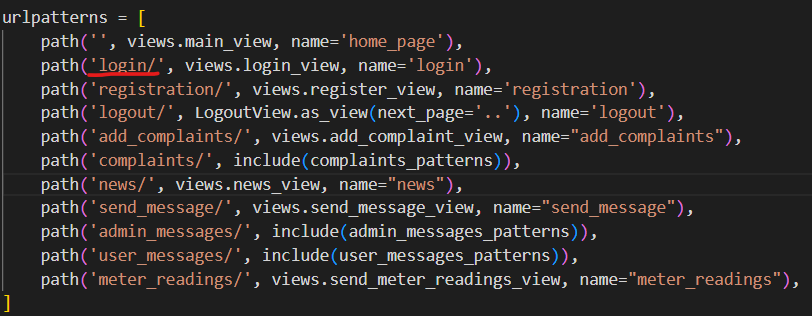
**Подача жалоб**

Рисунок 6.3.8 – Путь к странице входа

За логику процесса подачи жалоб отвечает представление «add\_complaint\_view()» (рис. 6.3.9).

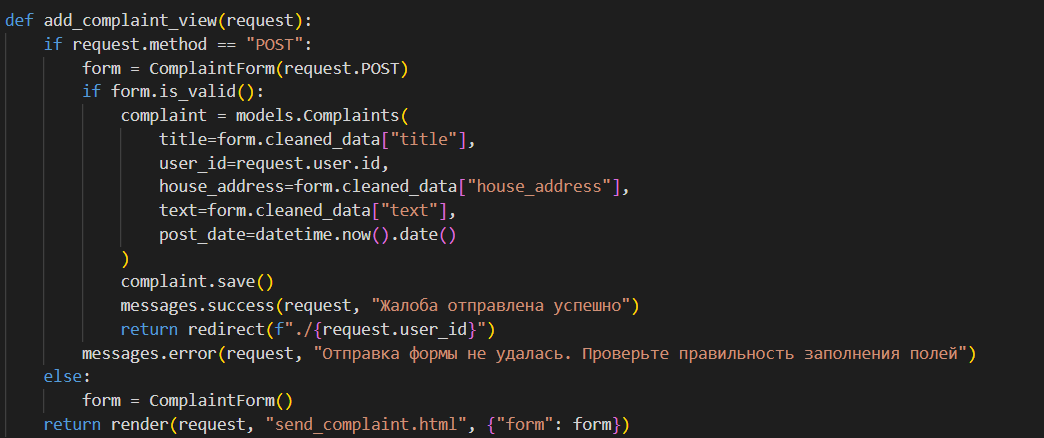
При подаче жалобы пользователь заполняет соответствующую форму, структура которой устанавливается классом «ComplaintForm()» (рис. 6.3.10).

Рисунок 6.3.9 – Представление «add\_complaint\_view()»

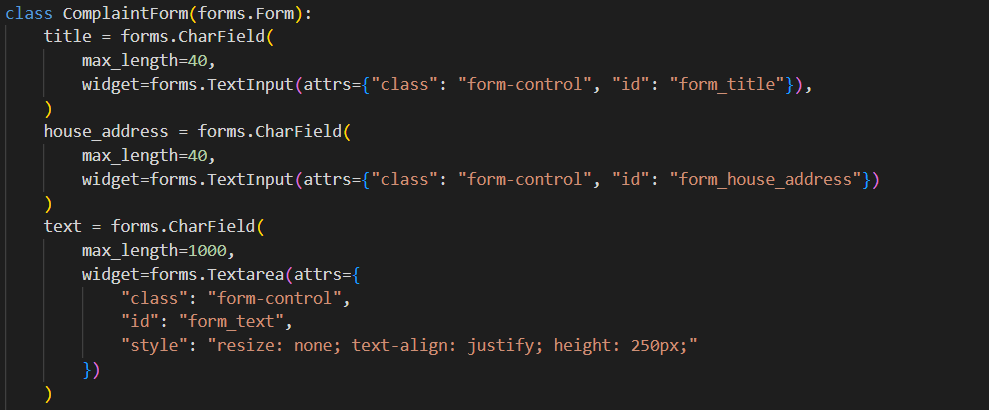
Валидация этой формы реализована с помощью инструментов Django (функция вида clean\_<название поля формы>). Ее реализация представлена на рисунке 6.3.11.

Рисунок 6.3.10 – Класс «ComplaintForm()»

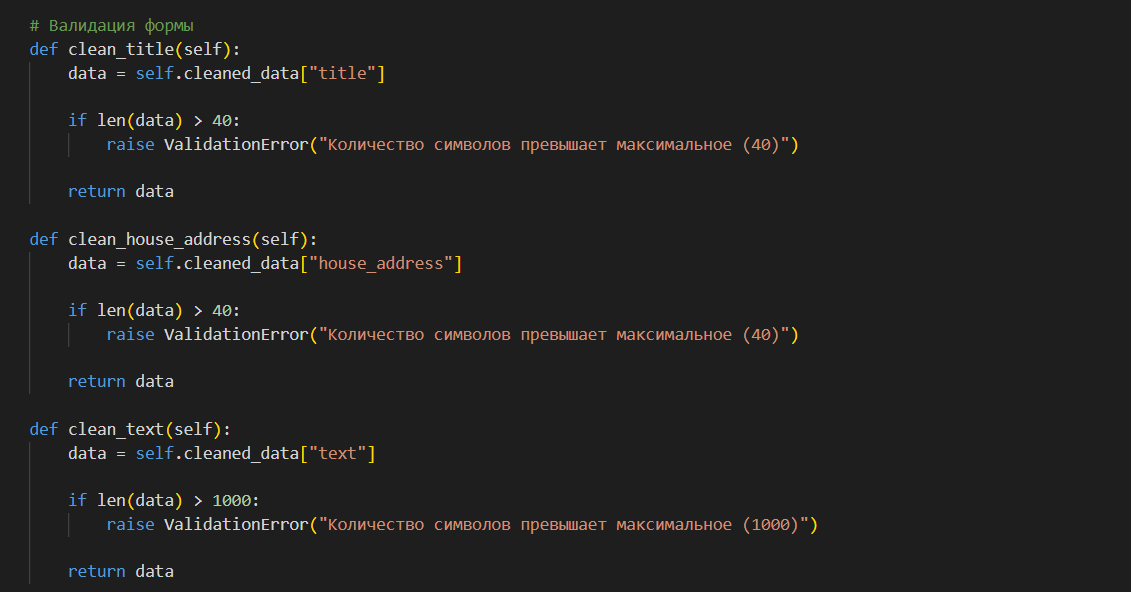
Данные передаются на сервер с помощью HTTP POST запроса и записываются в таблицу «Complaints», структуру которой устанавливает модель «Complaints» (рис. 6.2.2). Элементы страницы подачи жалоб содержатся в шаблоне «send\_complaint.html» (пункт 5.4). Маршрут к этой странице представлен на рисунке 6.3.12.

Рисунок 6.3.11 – Валидация формы подачи жалоб

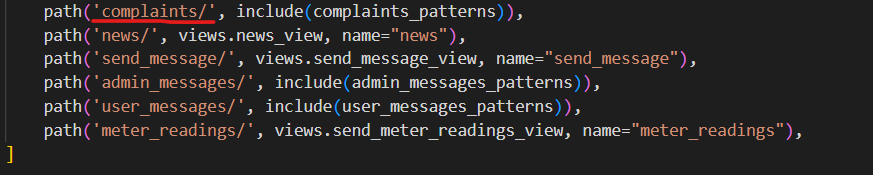
**Просмотр жалоб**

Рисунок 6.3.12 – Маршрут к странице подачи жалоб

На сайте можно просмотреть как все жалобы, так и жалобы, которые подал текущий пользователь. За логику процесса просмотра жалоб отвечает представление «all\_complaints\_view()» и «my\_complaints\_view()» (рис. 6.3.13).

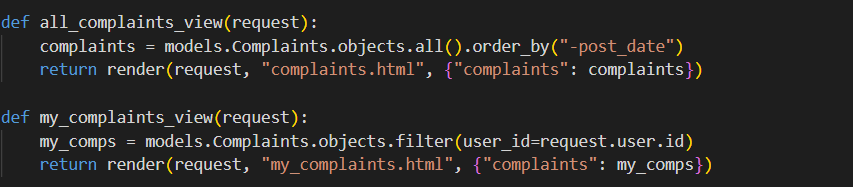
При входе на соответствующие страницы подгружаются данные из таблицы «Complaints». На странице «Мои жалобы» из базы данных выбираются жалобы, значение атрибута «user\_id» которых соответствует идентификатору авторизованного пользователя.

Рисунок 6.3.13 – Представления «all\_complaints\_view()», «my\_complaints\_view()»

Элементы страниц просмотра всех жалоб и жалоб конкретного пользователя содержатся в шаблонах «complaints.html» (пункт 5.5) и «my\_complaints.html» (пункт 5.6) соответственно. Маршруты к данным страницам представлены на рисунке 6.3.14.

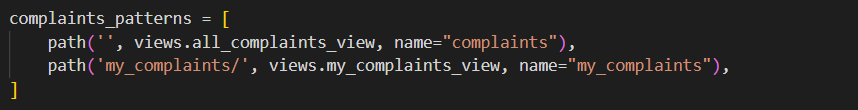
**Отправка показаний счетчиков**

Рисунок 6.3.14 – Маршруты к страницам просмотра жалоб

За логику процесса отправки показаний отвечает представление «send\_meter\_readings\_view()» (рис. 6.3.15).

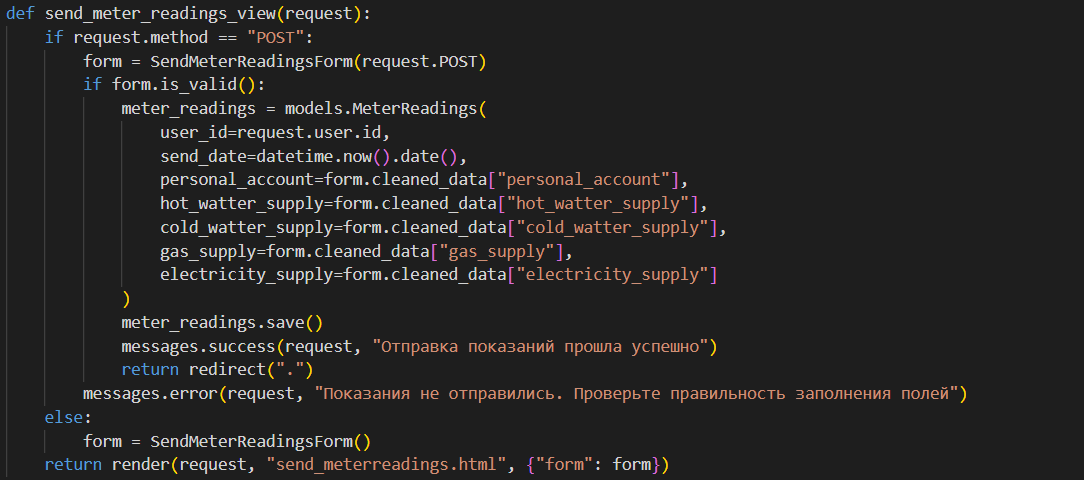
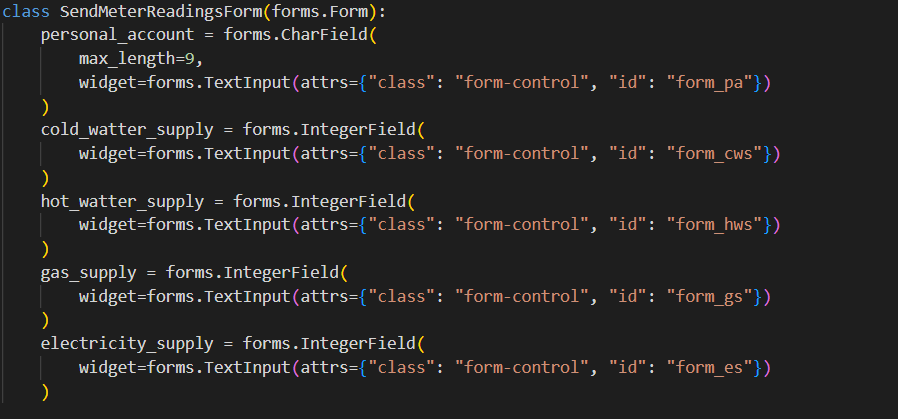
При передаче показаний счетчиков пользователь заполняет соответствующую форму, которая реализована с помощью класса «SendMeterReadingsForm()» (рис. 6.3.16).

Рисунок 6.3.16 – Класс «SendMeterReadingsForm()»

Рисунок 6.3.15 – Представление «send\_meter\_readings\_view()»

Данные передаются на сервер с помощью HTTP POST запроса и записываются в таблицу «MeterReadings», структуру которой устанавливает модель «MeterReadings» (рис. 6.2.3). Элементы страницы передачи показаний содержатся в шаблоне «send\_meterreadings.html» (пункт 5.7). Маршрут к этой странице представлен на рисунке 6.3.17.

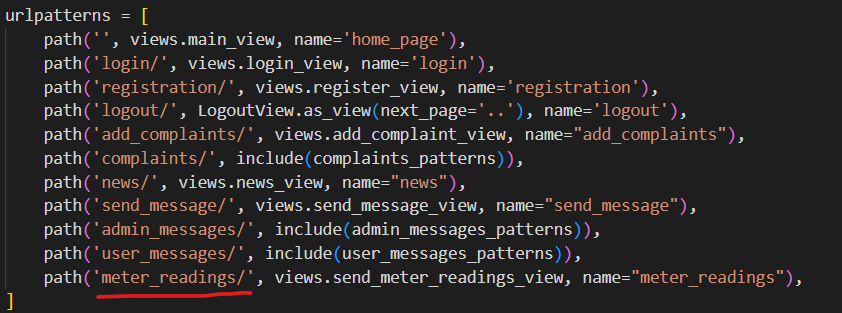
**Просмотр новостей**

Рисунок 6.3.17 – Маршрут к странице отправки показаний счетчиков

За логику процесса просмотра новостей отвечает представление «news\_view()» (рис. 6.3.18).

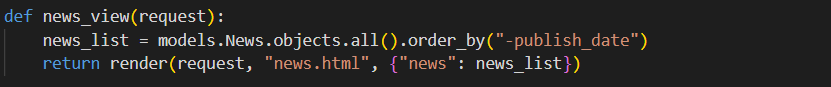
При входе на страницу «Новости» подгружаются данные из таблицы «News», структуру которой устанавливает модель «News» (рис. 6.2.5). Элементы данной страницы содержатся в шаблоне «news.html» (пункт 5.12). Маршрут к данной страницы представлен на рисунке 6.3.19.

Рисунок 6.3.18 – Представление «news\_view()»

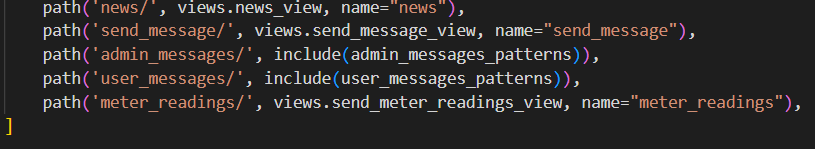
**Отправка сообщений в службу поддержки**

Рисунок 6.3.19 – Маршрут к странице просмотра новостей

За логику процесса отправки сообщений отвечает представление «send\_message\_view()» (рис. 6.3.20).

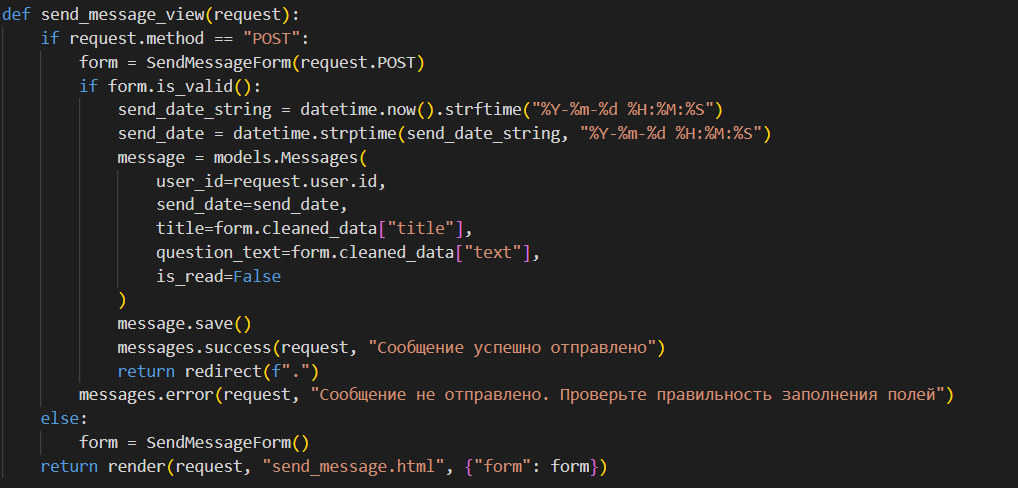
Перед отправкой сообщения пользователь заполняет соответствующую форму, которая реализована с помощью класса «SendMessageForm()» (рис. 6.3.21).

Рисунок 6.3.20 – Представление send\_message\_view()

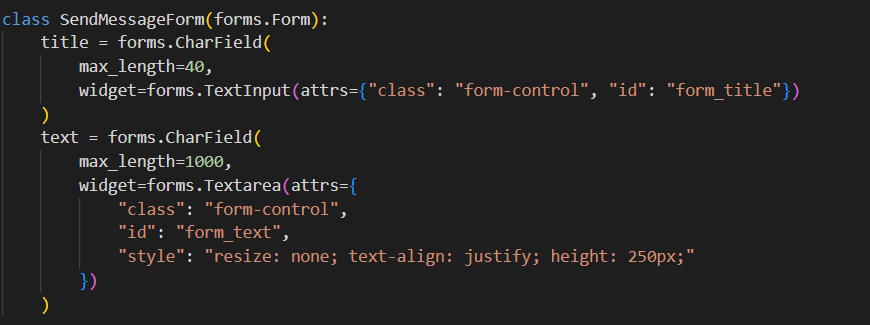
Данные передаются на сервер с помощью HTTP POST запроса и записываются в таблицу «Messages», структуру которой устанавливает модель «Messages» (рис 6.2.4).

Рисунок 6.3.21 – Класс «SendMessageForm()»

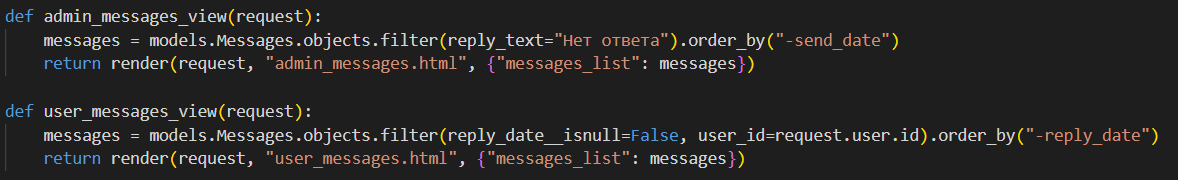
Элементы страницы отправки сообщений содержатся в шаблоне «send\_message.html» (пункт 5.8). Маршрут к этой странице представлен на рисунке 6.3.22.

**Просмотр сообщений**

Рисунок 6.3.22 – Маршрут к странице отправки сообщений

От лица службы поддержки входящие сообщения может посмотреть пользователь, обладающий правами администратора. Обычные пользователи также могут посмотреть входящие сообщения, являющиеся ответами службы поддержки. За логику данных процессов отвечают представления «admin\_messages\_view()» и «user\_messages\_view()» (рис. 6.3.23).

Рисунок 6.3.23 – Представления «admin\_messages\_view()», «user\_messages\_view()»

При входе на эти страницы из таблицы «Messages» выбираются нужные данные. Элементы страницы просмотра сообщений пользователем и страницы просмотра сообщений админом содержатся в шаблонах «user\_messages.html» и «admin\_messages.html» соответственно (пункты 5.9, 5.10). Маршруты к этим страницам представлены на рисунке 6.3.24.

На страницах просмотра сообщений есть функции «прочитать сообщение» и «удалить сообщение». Первая функция изменяет значение атрибута «is\_read» таблицы «Messages» с «False» на «True» (изменения отображаются на странице). Вторая функция удаляет запись о выбранном сообщении из таблицы «Messages». За логику работы функций отвечают представления «make\_read\_view()» и «delete\_message\_view()» (рис. 6.3.25).

Рисунок 6.3.24 – Маршруты к страницам просмотра сообщений

**Ответ службы поддержки на сообщение**

Рисунок 6.3.25 – Представления «make\_read\_view()», «delete\_message\_view()»

Админ, находясь на странице просмотра сообщений, может ответить на сообщение от пользователей, нажав на кнопку «Ответить». За логику процесса ответа на присланное сообщение отвечает представление «reply\_view()» (рис. 6.3.26).

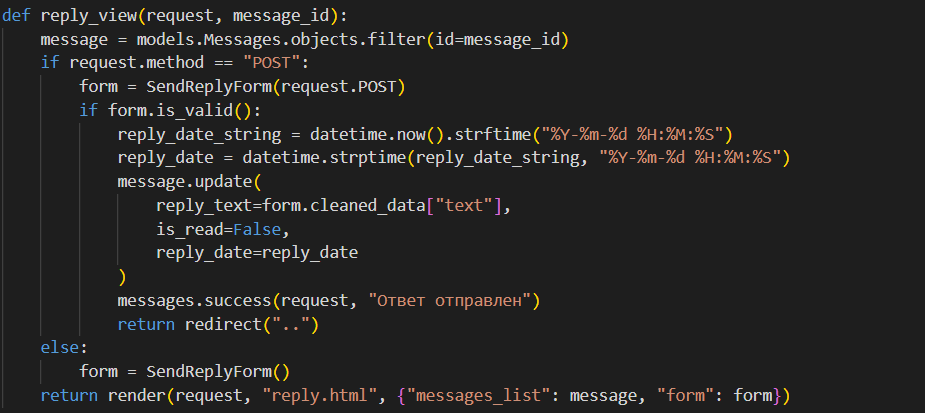
Перед отправкой сообщения-ответа пользователь заполняет форму, которая реализована классом «SendReplyForm()» (рис. 6.3.27).

Рисунок 6.3.26 – Представление «reply\_view()»

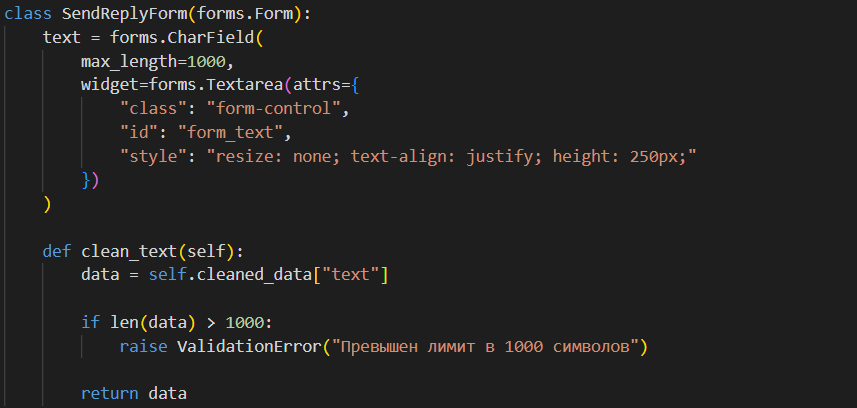
Валидация единственного поля формы реализована с помощью функции «clean\_text()» (рис. 6.3.27).

Рисунок 6.3.27 – Класс «SendReplyForm()»

Данные передаются на сервер с помощью HTTP POST запроса, заполняются колонки «reply\_text» и «reply\_date» таблицы «Messages», меняется значение атрибута «is\_read» с «True» на «False» (если до этого оно было изменено пользователем).

Элементы страницы ответа на сообщение содержатся в шаблоне «reply.html» (пункт 5.11). Маршрут к этой странице представлен на рисунке 6.3.28.

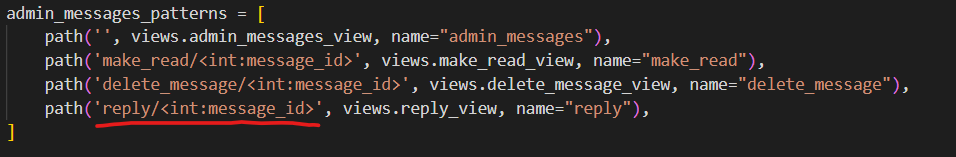


Рисунок 6.3.28 – Маршрут к странице ответа на сообщение

# Заключение

В ходе курсовой работы был разработан веб-сайт управляющей компании «Лесной» с использованием Django Framework. Проект включает в себя основные страницы, необходимые для взаимодействия жителей с компанией: главную страницу, формы для регистрации и авторизации, форму подачи жалоб и страницы для отображения всех жалоб и жалоб конкретного пользователя, новости компании, отправка сообщений и передача показаний счетчиков.

Сайт предоставляет необходимые инструменты для взаимодействия жильцов с управляющей компанией, повышая эффективность управления и качество обслуживания.