МИРЭА – Российский Технологический Университет

**Семестровая работа по дисциплине**

**«Фронтенд и бекенд разработка»**

**ПЛАТФОРМА ДЛЯ КОММУНИКАЦИИ ЖИЛЬЦОВ И АДМИНИСТРАЦИИ ОБЩЕЖИТИЯ**

**Сведения об авторе:**

**Столяров Даниил Шикхарович,**

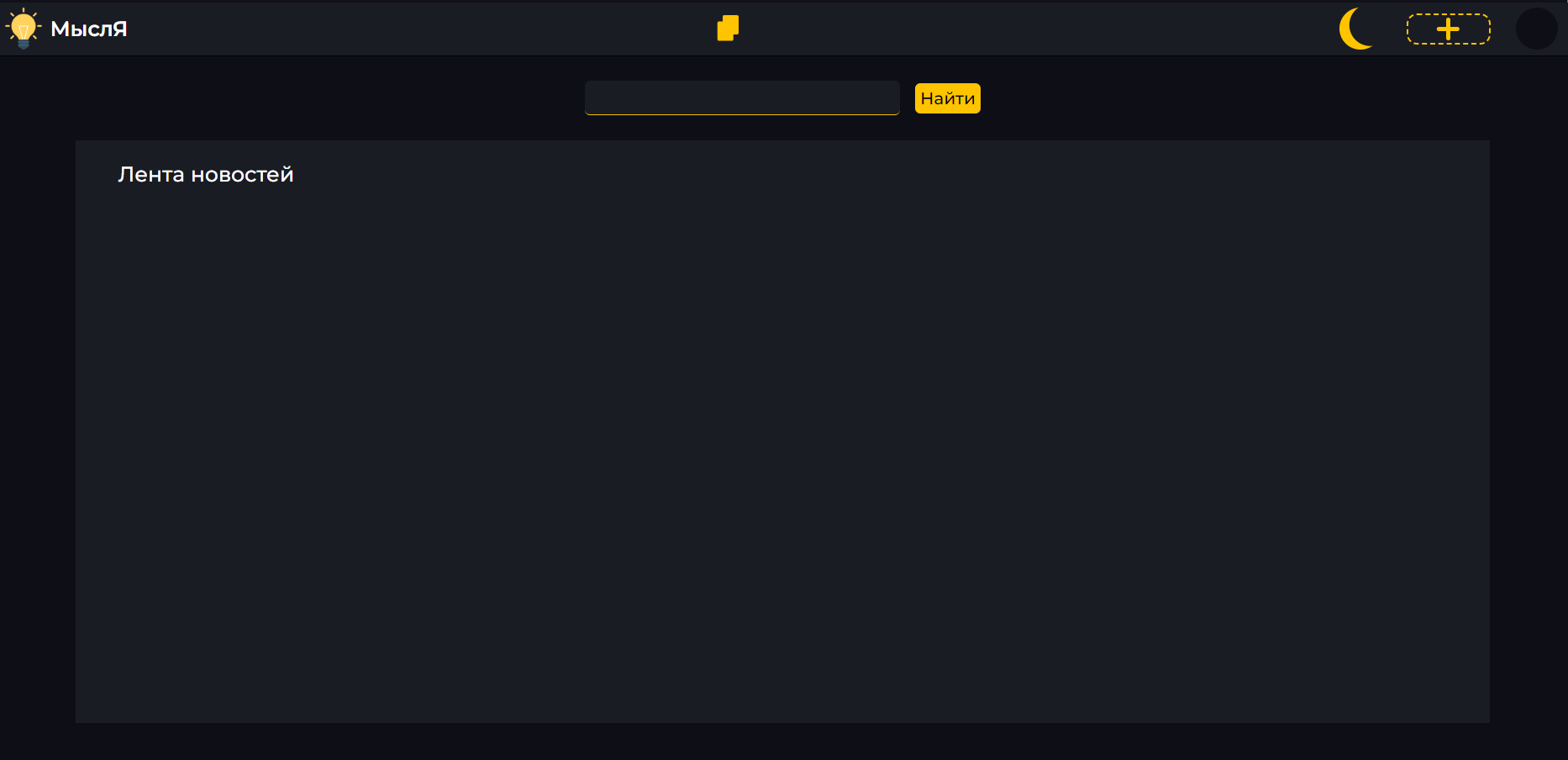
**ЭФБО-01-22**

2023 год

# Описание страниц

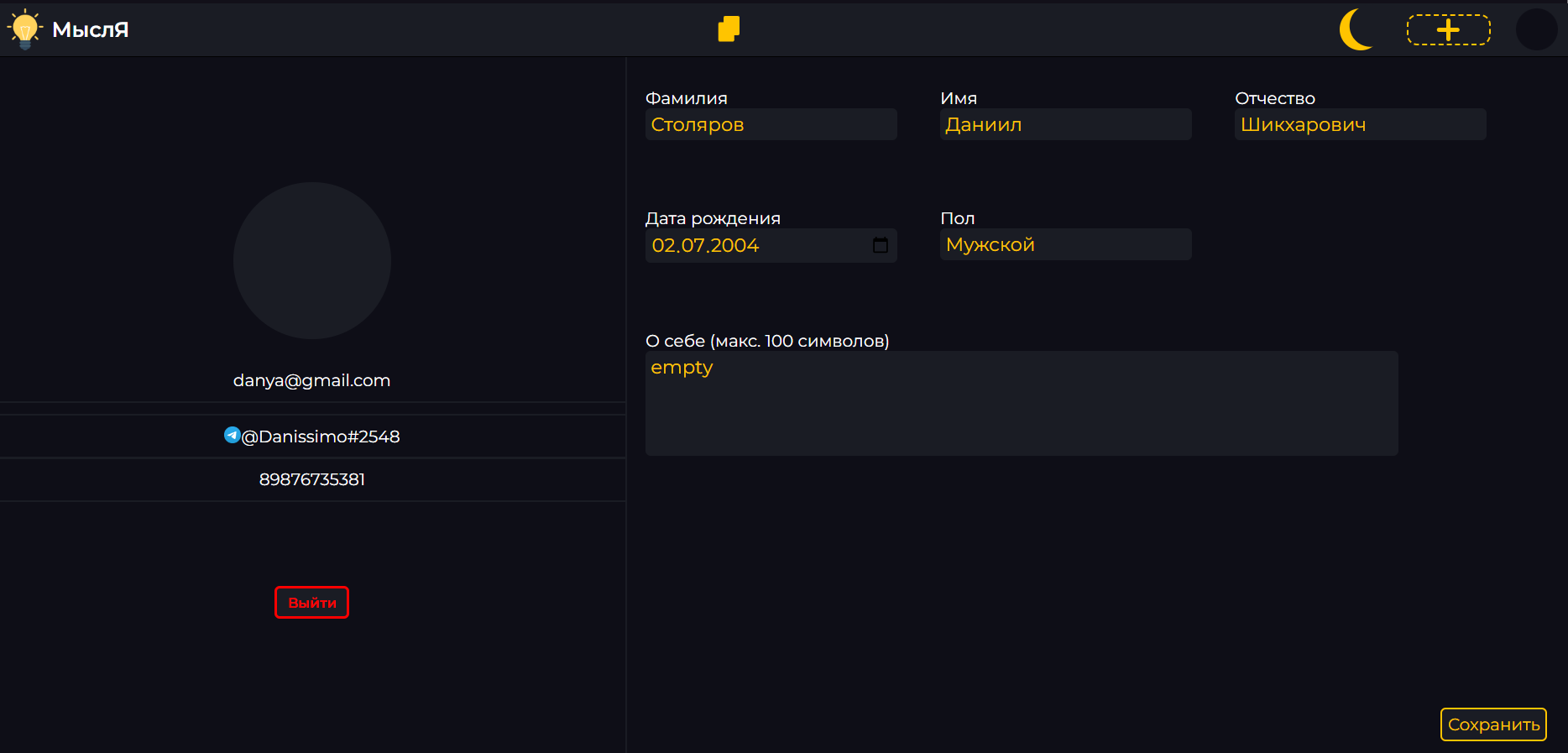
## Страница «Лента новостей»

На данной странице будут отображаться уведомления от администрации общежития для всех жильцов. Верхняя панель имеет кнопки (перейти на главную страницу, посмотреть заявки, переключить цветовую тему в режим для слабовидящих, добавить заявку, открыть профиль)



## Страница «Профиль»

На данной странице пользователь может увидеть хранящуюся информацию о себе. Также он может осуществить выход из учётной записи.



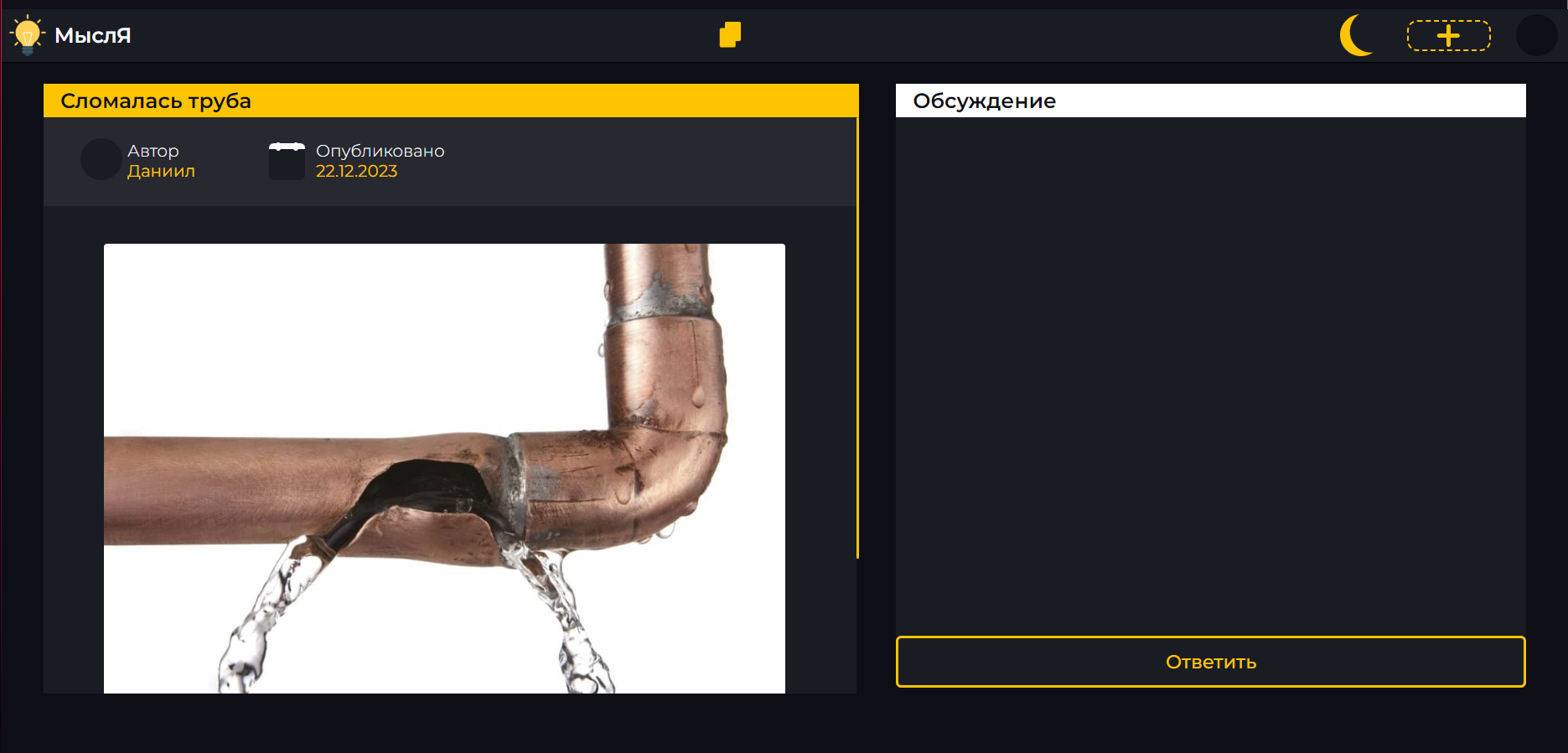
## Страница «Мои заявки»

На данной странице выведен список из поданных данным пользователем заявок. При клике на заголовок заявки пользователь попадает на страницу с заявкой.

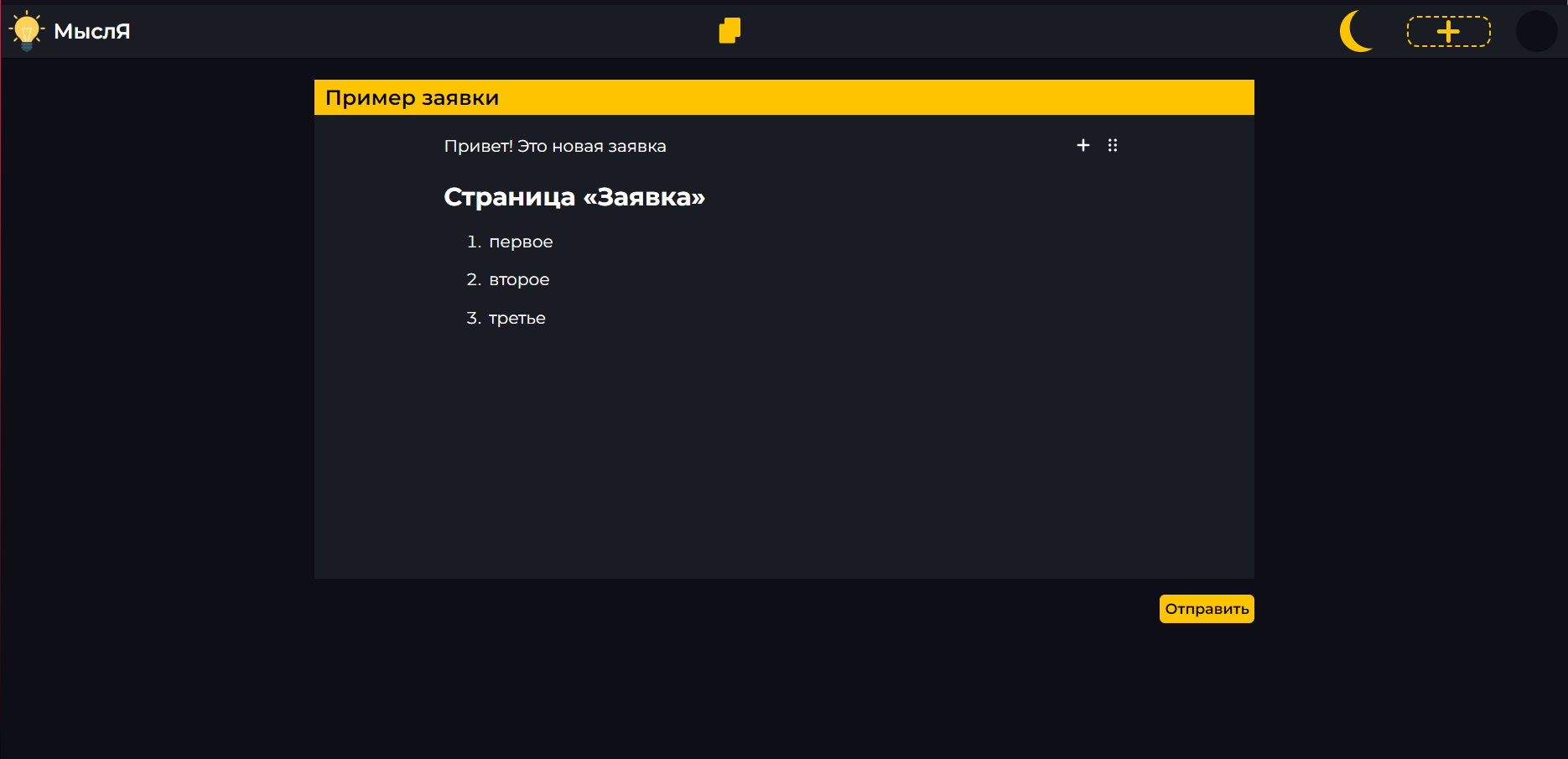


## Страница «Заявка»

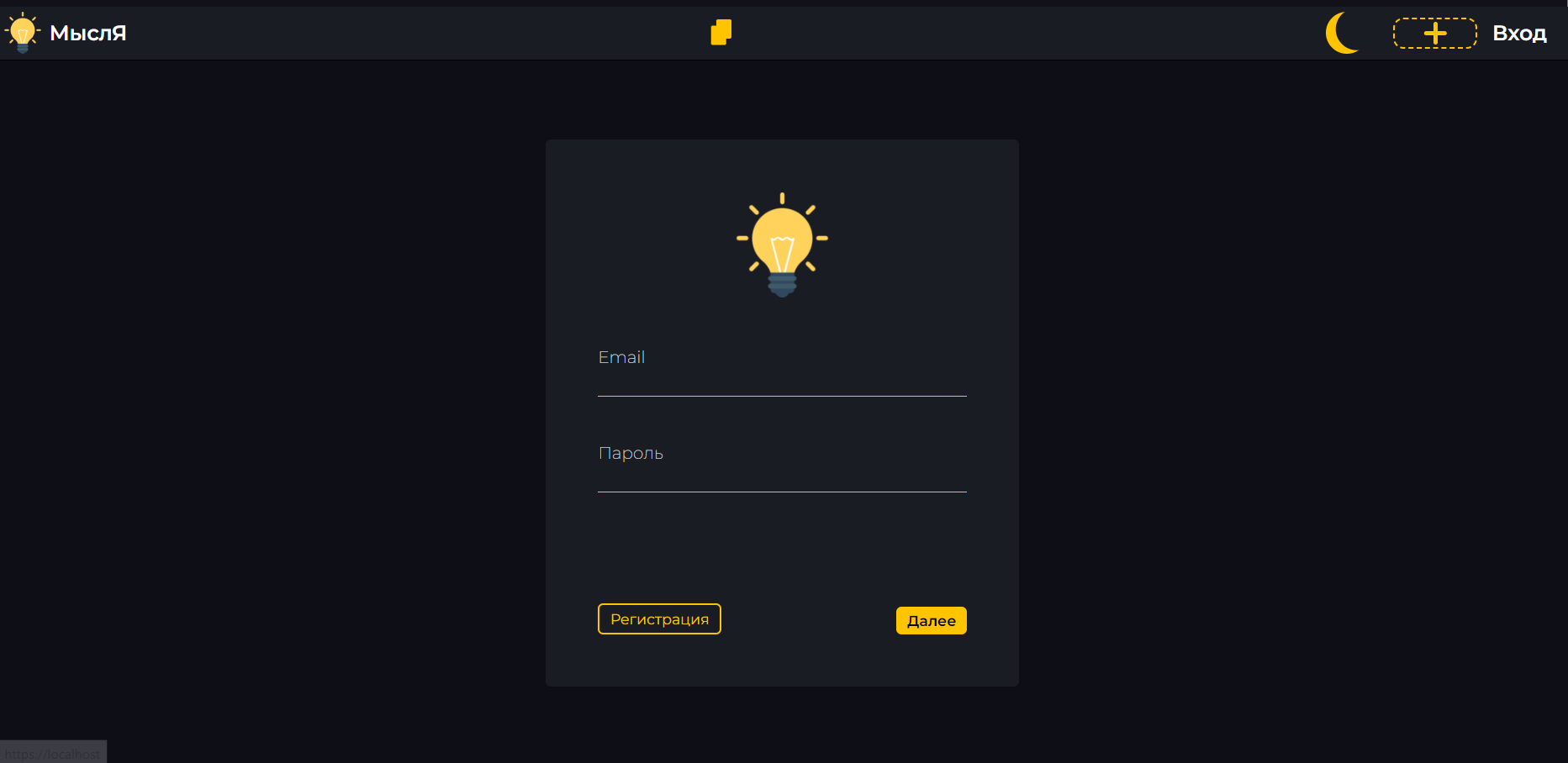
На данной странице есть два окна: окно с исходной заявкой (информация об авторе, название и описание заявки, которое может включать картинки) и окно с обсуждением, где администрация и пользователь могут обсудить заявку.

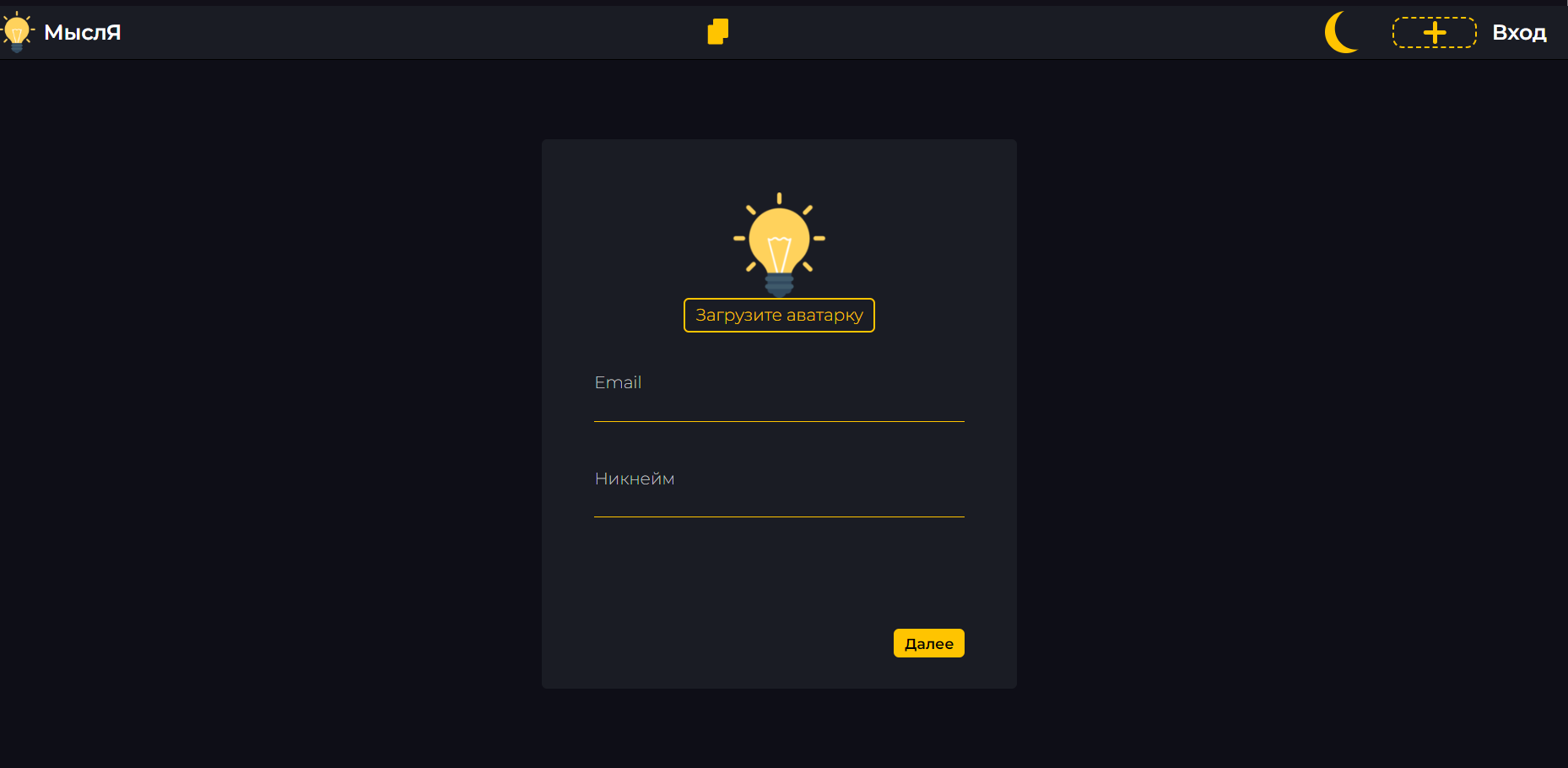


## Страница «Добавить заявку»

На данной странице пользователь может добавить заявку. Для этого предлагается использовать текстовый редактор, который поддерживает вставку картинок.

## Страницы «Регистрация» и «Логин»

На данных страницах можно соответственно зарегистрироваться и войти в учётную запись. 



# Схема маршрутизации

## REST API

**GET \*** - отправить /front/index.html (приложение использует паттерн Single Page Application)

**GET /images/\*** - отправить картинку с указанным uuid или сообщение об ошибке.

**POST /register** (отправка формы регистрации) – зарегистрирует пользователя или (в случае занятого email или некорректного пароля) вернёт ошибку.

**POST /login** (отправка формы входа) – авторизирует пользователя или (в случае некорректного пароля) вернёт ошибку.

**POST /upload-image-file** (отправка формы, к которой прикреплена картинка) – загрузит картинку на сервер, если та не превышает определённого размера (8 мб).

**POST /upload-image-url** (отправка формы, содержащей url картинки) – сохранит внешнюю ссылку на картинку, не загружая её саму.

**POST /update-user-data** (отправка формы с данными пользователя) – обновляет данные о пользователе.

## Socket.IO API

В данном проекте используется фреймворк Socket.IO. События названы по следующей схеме: ***<название>*** - вызов события на стороне клиента;  
***successful <название>*** - обработка при удачном сценарии обработки события;  
***failed <название>*** - обработка при неудачном сценарии обработки события;

На данный момент зарегистрированы следующие события.

**“application fetch”** – загружает заявку по её id.  
**“fetch mine applications”** – загружает список собственных заявок.  
**“application apply”** – создаёт новую заявку.  
**“fetch user”** – загрузка публичных данных о пользователе.  
**“fetch by key”** – загрузка полных данных о пользователе, используя ключ аутентификации.  
**“remove application”** – удаление заявки по её id.

# База данных

В качестве СУБД используется PostgreSQL. Схема базы данных представлена ниже.

