|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Теоретическая информатика и компьютерные технологии (ИУ9)

**ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Студент *Костоусов Данил Витальевич*

*фамилия, имя, отчество*

Группа *ИУ9-21Б*

Тип практики проектно-технологическая практика

Название предприятия МГТУ имени Н. Э. Баумана

Команда *Сайт: форум*

Студент *Костоусов Д. В.*

*подпись, дата фамилия, и.о.*

Руководитель практики *Коновалов А. В.*

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2023 г*

**Введение:**

Идея написания сайта была выбрана, так как хотелось изучить такую категорию программирования, как Web, чтобы иметь не только общее представление, но и некоторый опыт в этой сфере. Это бы помогло понять, есть ли желание дальше развиваться в данной области. А по причине того, что в музыке мне интересен такой жанр, как «рэп», было принято решение совместить эти идеи и в качестве летней практики написать сайт – форум по рэпу и хип-хопу. Он должен включать в себя следующий функционал:

1. Лента с редактором для написания новостей, где админы сайта могли бы писать и выкладывать различные статьи по тематике сайта
2. Форум с некоторым количеством веток, каждая из которых отвечала бы определенной подтеме
3. Возможность создания и удаления авторизованным пользователям собственного «треда» в определенной ветке
4. Возможность оставлять сообщения в определенном «треде», отвечать на сообщения других пользователей, удалять собственные сообщения
5. Система регистрации с капчей и подтверждением через почту собственного аккаунта на сайте, входа в личный аккаунт и смены пароля
6. Страница аккаунта пользователя с возможностью написания на ней собственных статьей, редактирования личной информации, просмотр статей других пользователей
7. Форма для пожертвования материальных средств
8. Система администрирования сайта

**Описание проекта**

Для создания сайта были выбраны такие средства, как:

1. HTML, CSS – для написания верстки сайта
2. Язык программирования Python, фреймворк Flask и библиотека sqlalchemy для написания серверной части (были выбраны по причинам того, что они просты в изучении и использовании, гибкие, а также имеют кучу инструментов для работы с разными частями бекенда)
3. Приложение SQLite для работы с базой данных
4. Paint3D для создания логотипа и стандартного аватара пользователя
5. А также несколько других мелких библиотек для небольших задач, например captcha.image для создания и проверки капчи.

Была написана система регистрации, при которой после корректного заполнения формы и верной введенной капчи на почту, указанную в форме, от лица рабочей почты сайта [pihpoh188@gmail.com](mailto:pihpoh188@gmail.com) приходит письмо с уникальной ссылкой на страницу сайта для подтверждения почты пользователя. Только после этого аккаунт считается подтвержденным и появляется возможность в него зайти через специальную форму с указанием данных, оставленных в форме регистрации. На этой же странице есть возможность восстановление пароля, для этого надо указать почту от аккаунта, и, если аккаунт с данной почтой существует, то на эту почту также придет письмо с уникальной ссылкой на форму для указания нового пароля. После смены пароля появляется возможность зайти в аккаунт с этой же почтой, но с новым паролем. В добавлении была реализована возможность сменить пароль на собственной странице пользователя на новый.

Также была написана системы администрирования, в которой пользователь со статусом «АДМИН» может назначить админом любого другого пользователя. Данный статус дает возможность написания новостей в общей ленте, удаления тредов и сообщений других пользователей.

Для организации системы пожертвования владельцу сайта материальных средств была выбрана платежная система Юmoney, для подключения которой достаточно было вставить специальную форму от Yandex Forms в HTML код страницы, предназначенных для пожертвований. После заполнения формы с указанием суммы пользователя отправляет на сайт Юmoney, через который и осуществляется перевод средств на личный кошелек автора.

Далее разберем архитектуру проекта. Файл main.py запускает сервер с сайтом, а также в нем описаны маршруты и функции со всей логикой работы:

1. page\_not\_found – функция, которая возвращает шаблон при ошибке 404 (при попытке зайди на страничку сайта с адресом, которого не существует)
2. page\_not\_found\_500 – обработчик ошибки 500
3. article\_line – функция для обработки страницы со списком всех новостей (доступна в том числе и для неавторизированных пользователей)
4. сreate\_article – создание новости (доступно только для пользователей, имеющих статус «АДМИН»)
5. article – просмотр конкретной новости
6. delete\_article – функция для удаления новости (доступно только для пользователей, имеющих статус «АДМИН»)
7. forum – обработчик для просмотра страницы с ветками
8. sect - обработчик для просмотра страницы с конкретной ветки со списком всех существующих «тредов» в нем
9. create\_thread – функция для обработки страницы с созданием и написанием нового «треда» (доступно только для авторизированных пользователей)
10. thread – обработчик страницы с определенным «тредом» и списком сообщений, оставленных в нем, а также формой для написания собственного сообщения.
11. delete\_thread – функция для удаления конкретного «треда» («тред» может удалить только пользователь, являющийся его автором или админ)
12. delete\_message – обработчик для удаления сообщения (сообщение может удалить только пользователь, являющийся его автором или админ)
13. edit message – обработчик страницы для редактирования сообщения пользователем, оставившим его
14. about – обработка страницы с небольшим описанием проекта
15. donate - обработка страницы с формой для пожертвования материальных средств автору сайта
16. register – функция для обработки страницы с формой для регистрации нового аккаунта
17. login - функция для обработки страницы с формой для входа пользователя в аккаунт
18. recover\_password – обработчик страницы для восстановления или изменения пароля
19. mail\_recover\_password – функция для создания уникального токена и отправки письма на почту с ссылкой на страницу для восстановления пароля
20. logout – функция для выхода пользователя из аккаунта
21. confirm\_email – функция для обработки страницы, ссылка на которую указывается в письме для подтверждения почты созданного аккаунта
22. change\_pass - функция для обработки страницы, ссылка на которую указывается в письме для смены забытого пароля
23. user – обработчик страницы пользователя
24. give\_admin – функция для обработки события – присвоения другому пользователю статуса «АДМИН» (доступно только админам)
25. edit\_page – обработчик страницы с изменением личной информации пользователя
26. all\_user\_threads – обработчик страницы со всеми «тредами» определенного пользователя (доступно из страницы пользователя)
27. all\_user\_articles – обработчик страницы со всеми статьями определенного пользователя (доступно из страницы пользователя)

Также в файле main.py прописывается все настройки проекта и несколько дополнительных функций, упрощающих логику работы определенных частей программы.

В файле forms.py прописаны все формы для заполнения данных, использующихся на определенных страницах сайта, а конкретно:

1. LoginForm – форма для заполнения данных при входе пользователя в личный аккаунт
2. RegisterForm – форма для регистрации нового аккаунта
3. ThreadForm – форма для создания нового «треда» (с указанием названия и описания) или же редактирования существующего
4. MessageForm – форма для написания нового сообщения под определенным «тредом» или же редактирования существующего
5. EmailForm – форма для указания почты при восстановлении пароля от аккаунта
6. PasswordForm – форма для указания нового пароля при смене
7. EditForm – форма для редактирования личной информации от аккаунта пользователя
8. ArticleForm – форма для написания статьи или же редактирования существующой

В папке templates содержатся все html файлы страниц сайта. Стоит упомянуть, что был использован инструмент для шаблонов страниц, с помощью которого можно написать 1 base.html файл, использующийся на всех страницах, а в остальных файлах описывать лишь ту часть страницы, которая отличается от остальных.

В папке static содержится css файл со стилями и все изображения, использующиеся в сайте.

Далее нужно сказать, что для удобства разработки данного проекта были отдельно вынесены функции, разделенные по категориям (сервисам), каждая из которых отвечала за собственный аспект работы сайта, и каждая категория функций описана в собственном .py файле. А все модели описывающие таблицы базы данных вынесены каждая в свой .py файл и собраны в папке data. В целом вся архитектура проекта показана на рис. 1.

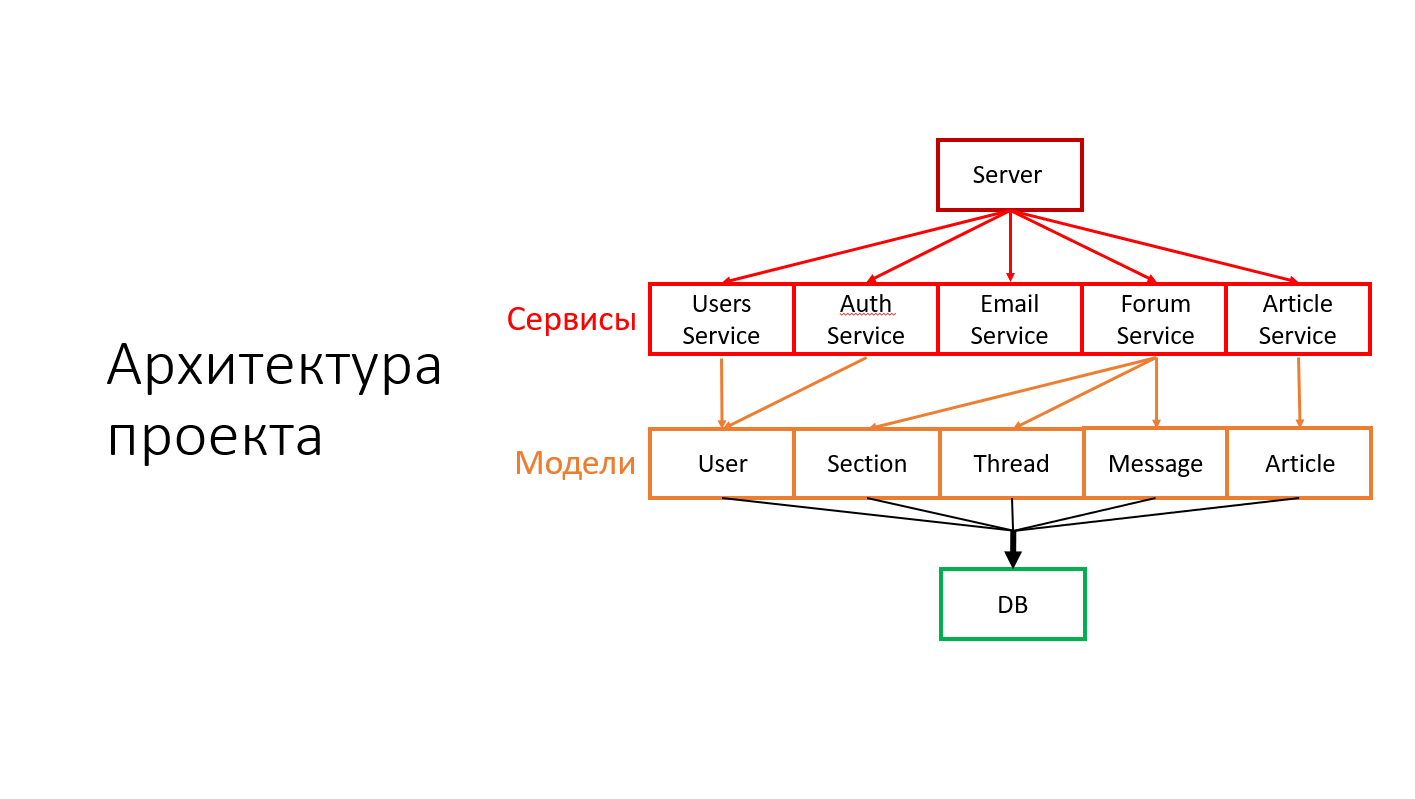


рис. 1

Как можно заметить в проекте существует 5 сервисов, и, как уже было написано выше, каждый из них отвечал за работу определенной части сайта.

1. Users Service – функции для работы с данными пользователей или же контентом, написанном и выложенным на сайте пользователями
2. Auth Service – функции для работы с авторизацией пользователей
3. Email Service – функции для отправки писем на почтовые ящики пользователей и генерации уникальных ссылок и токенов
4. Forum Service – функции для работы форума
5. Article Service – функции для работы ленты новостей со статьями

Выше упомянуто, что для работы с базой данных использовалась библиотека sqlalchemy, так как она удобна в использовании с Flask и проста в использовании, а также модули, описывающие таблицы, вынесены отдельно каждый в свой .py файл. Для корректной работы проекта понадобилось создать следующие таблицы:

1. Users – таблица, в которой хранится информация об аккаунтах пользователей (id, ник, личная информация о пользователе, дата рождения, статус, аватар (картинка или .gif файл ввиде набора байт), кеш аватара, почта, пароль (ввиде хеша), дата создания аккаунта и поле, указывающее на то, подтвержден ли аккаунт через почту)
2. Articles – таблица с данными статей (id, название, описание, дата создания, id автора, поле, указывающее на то, написана ли статья админом(для разделения личных статей пользователей и новостных статей, отображающихся в ленте))
3. Threads – таблица с данными о «тредах» (id, название, описание, дата создания, время написания последнего сообщения в «треде» (нужно для определения статуса активности «треда»), количество сообщений в «треде», поле со статусом активности «треда», id автора и id ветки)
4. Messages – таблица с данными сообщений (id, содержание сообщения, дата создания, поле с id другого пользователя (если это сообщение – ответ на сообщение другого пользователя), id автора, id «треда», в котором написано сообщение и время последнего редактирования
5. Sections – таблица с данными о ветках (id, название, адрес для создания ссылки на ветку, описание, количество «тредов», количество активных «тредов», время создания последнего «треда»)

Общая логика связи таблиц отображена на рис. 2

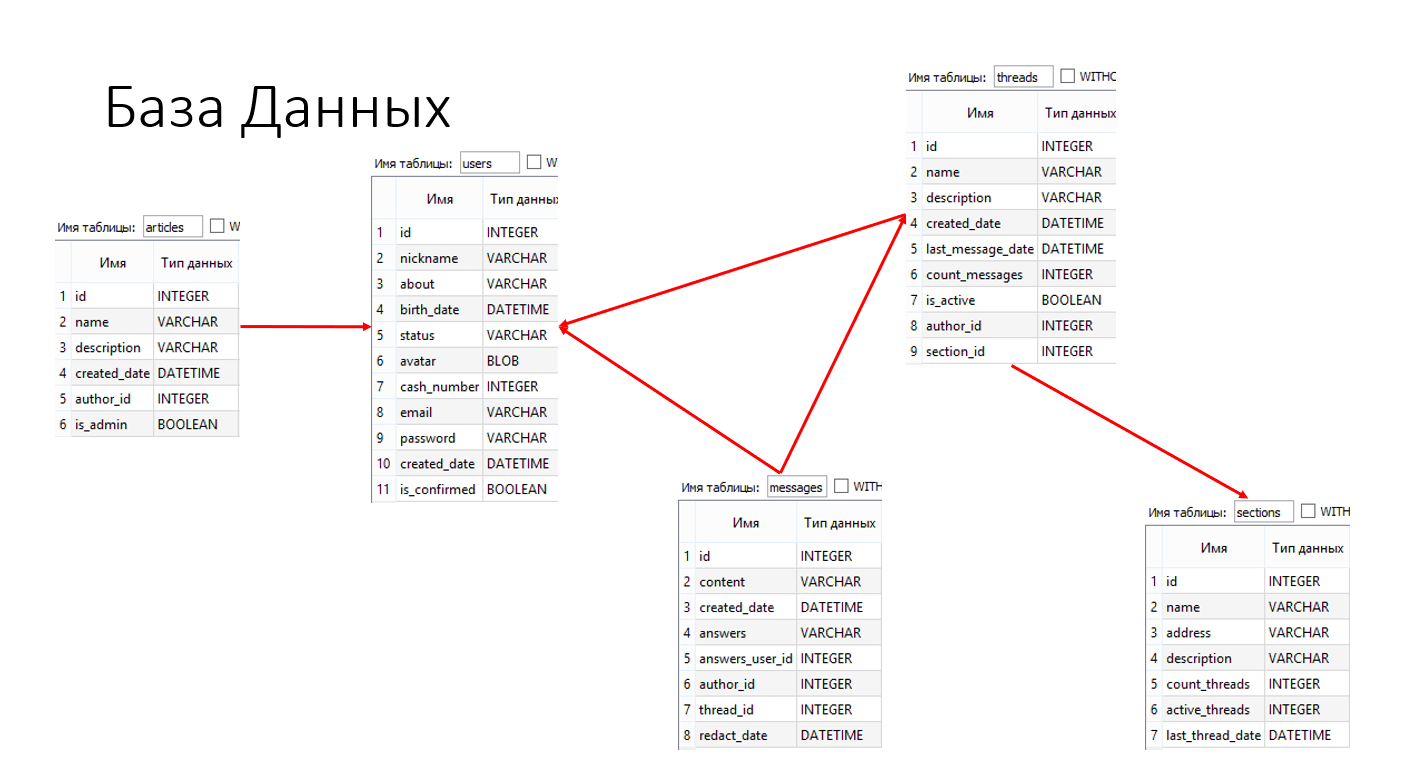
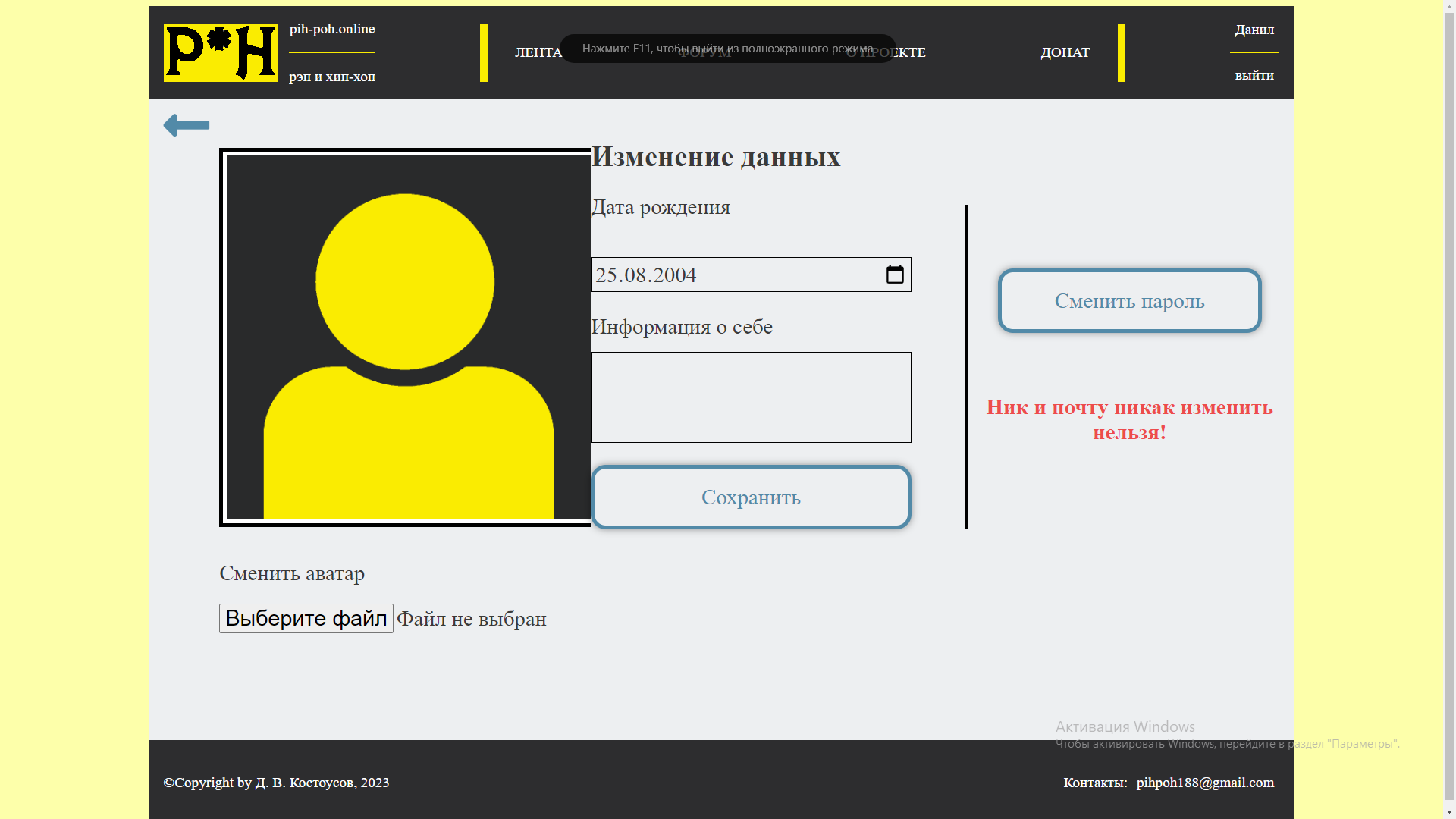
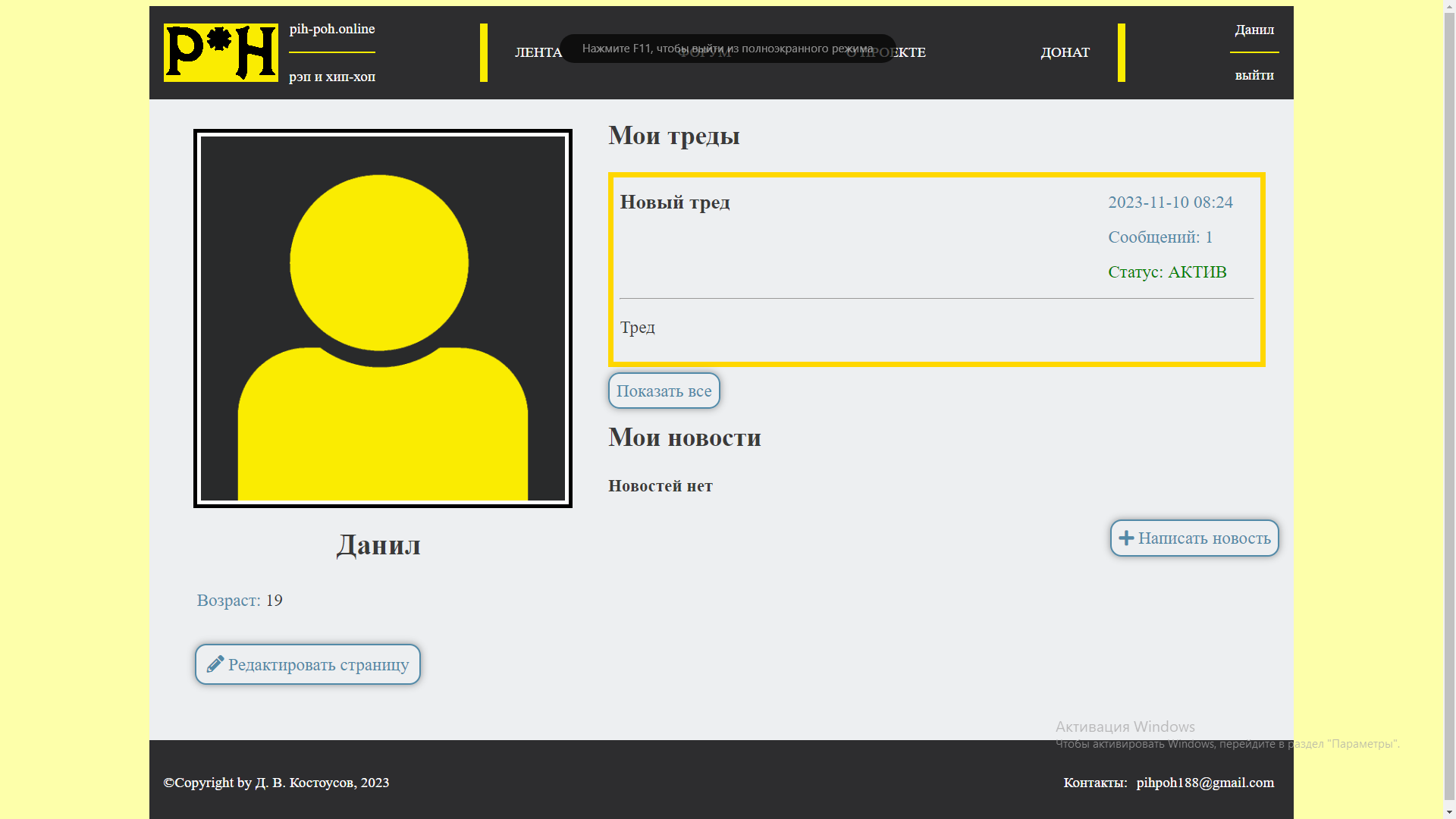


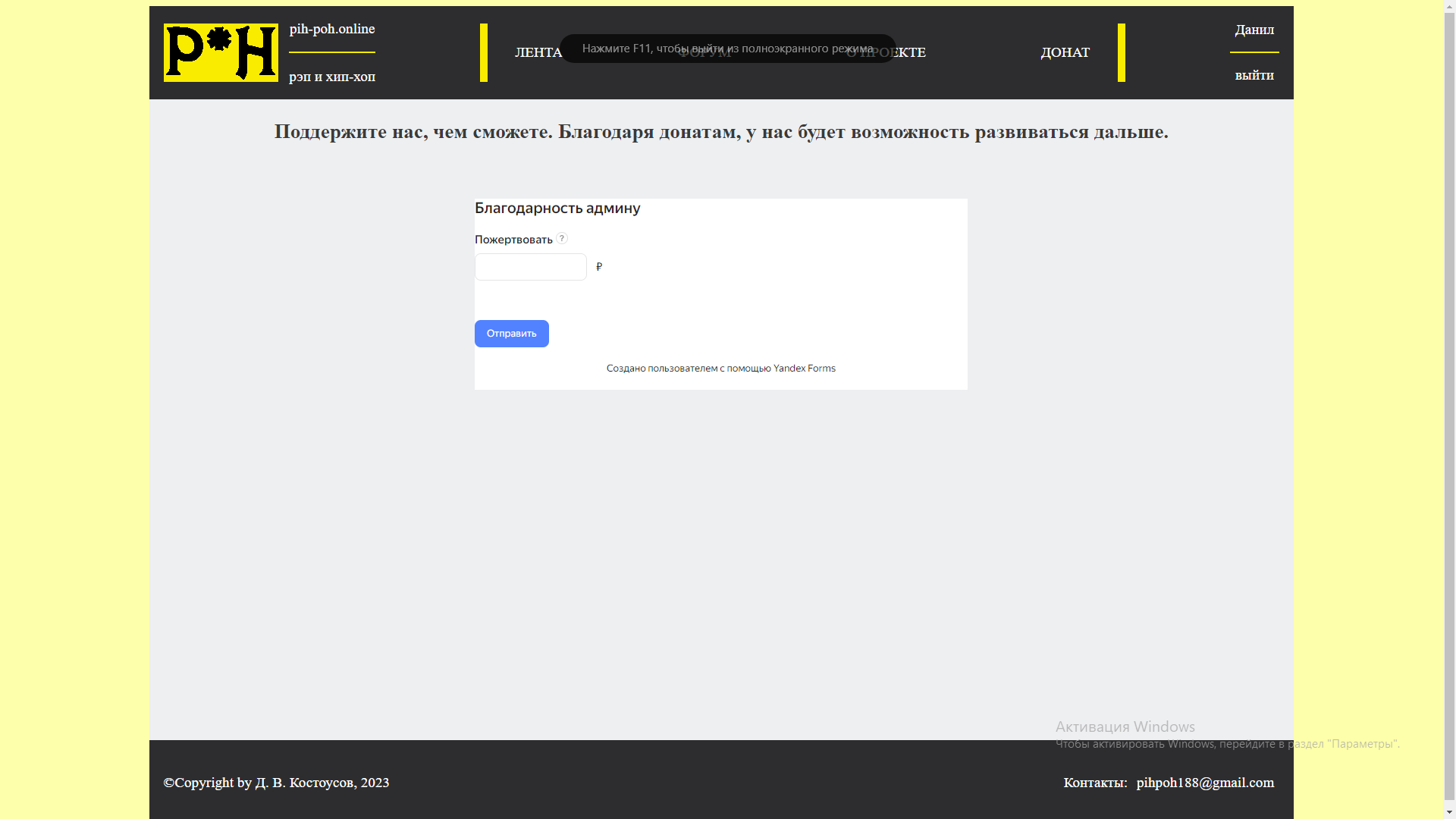
рис. 2

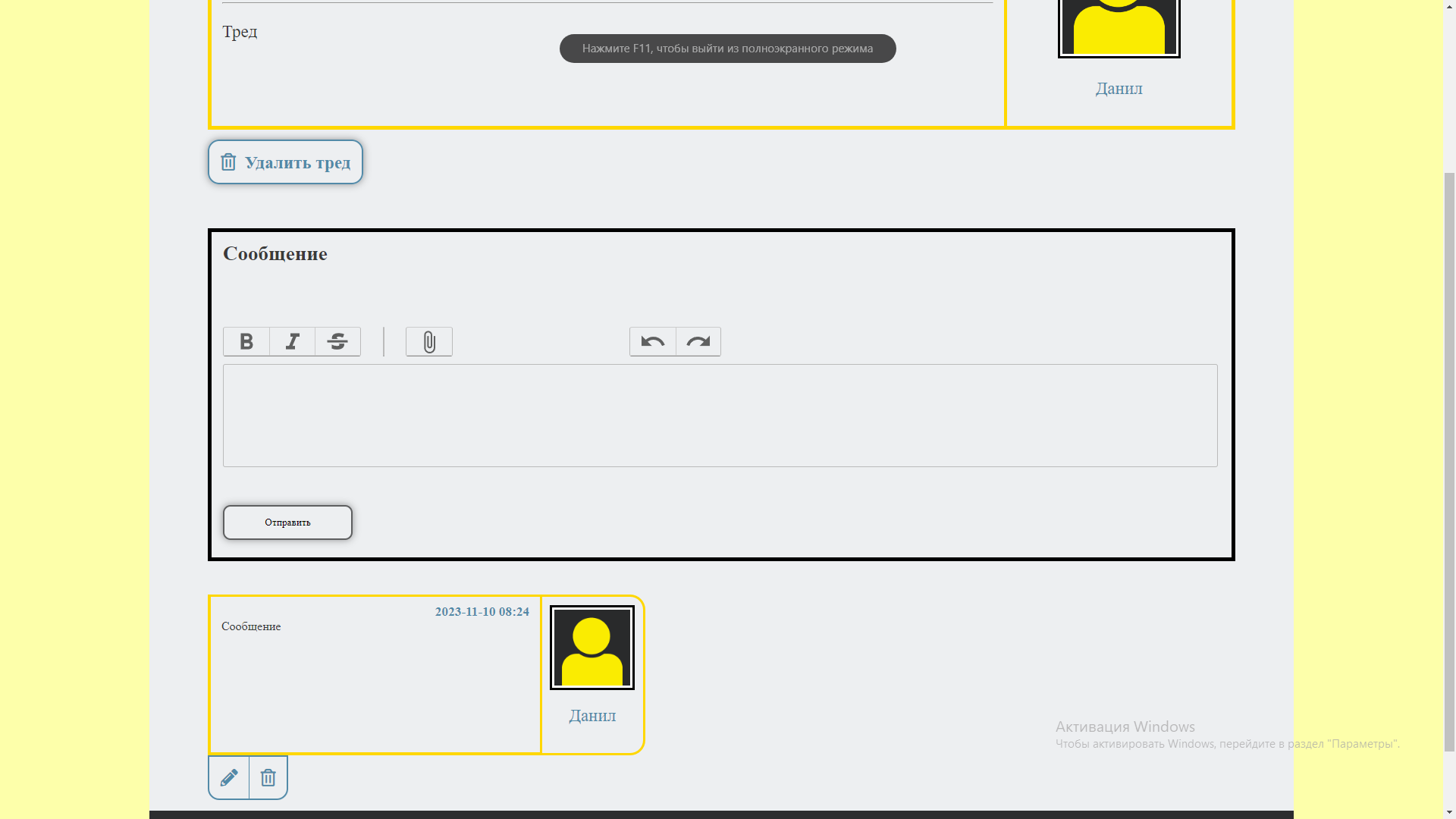
Также в проекте есть файл db\_session.py для настройки и более удобной работы с базой данных и файл seeder.py, для заполнения таблицы с ветками при запуске сервера.

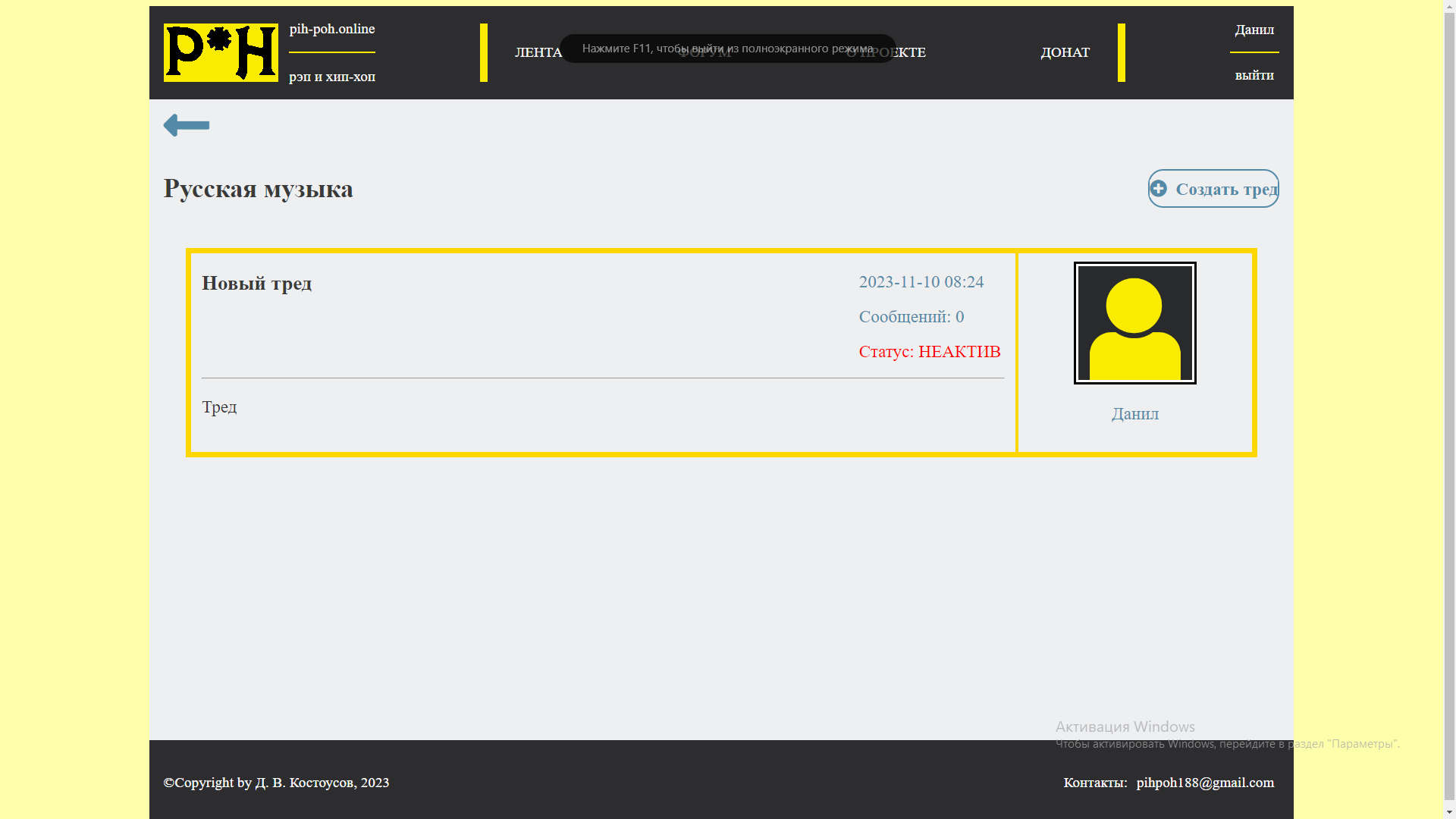
Некоторая часть работы сайта отображена на рис. ниже:

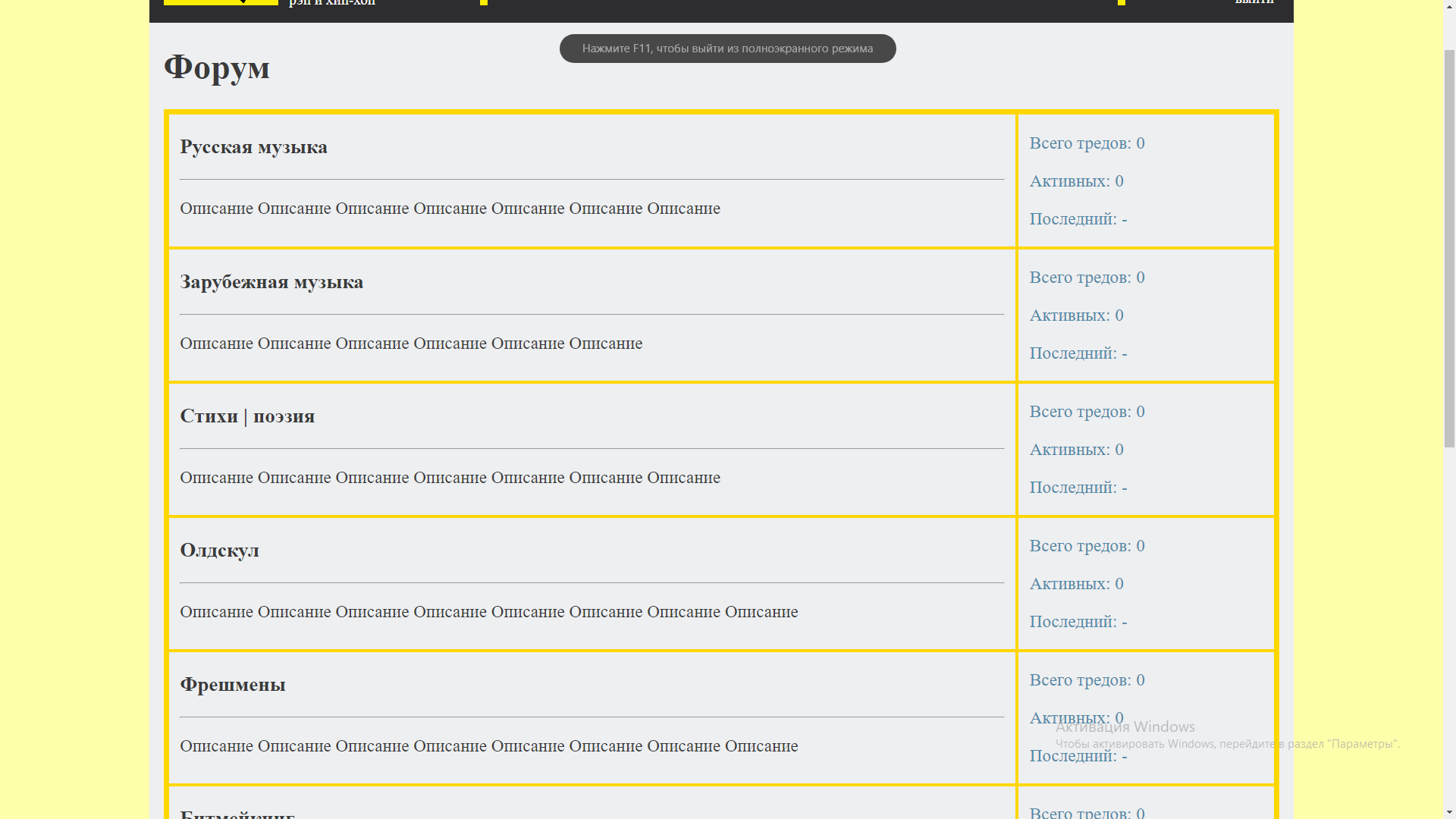


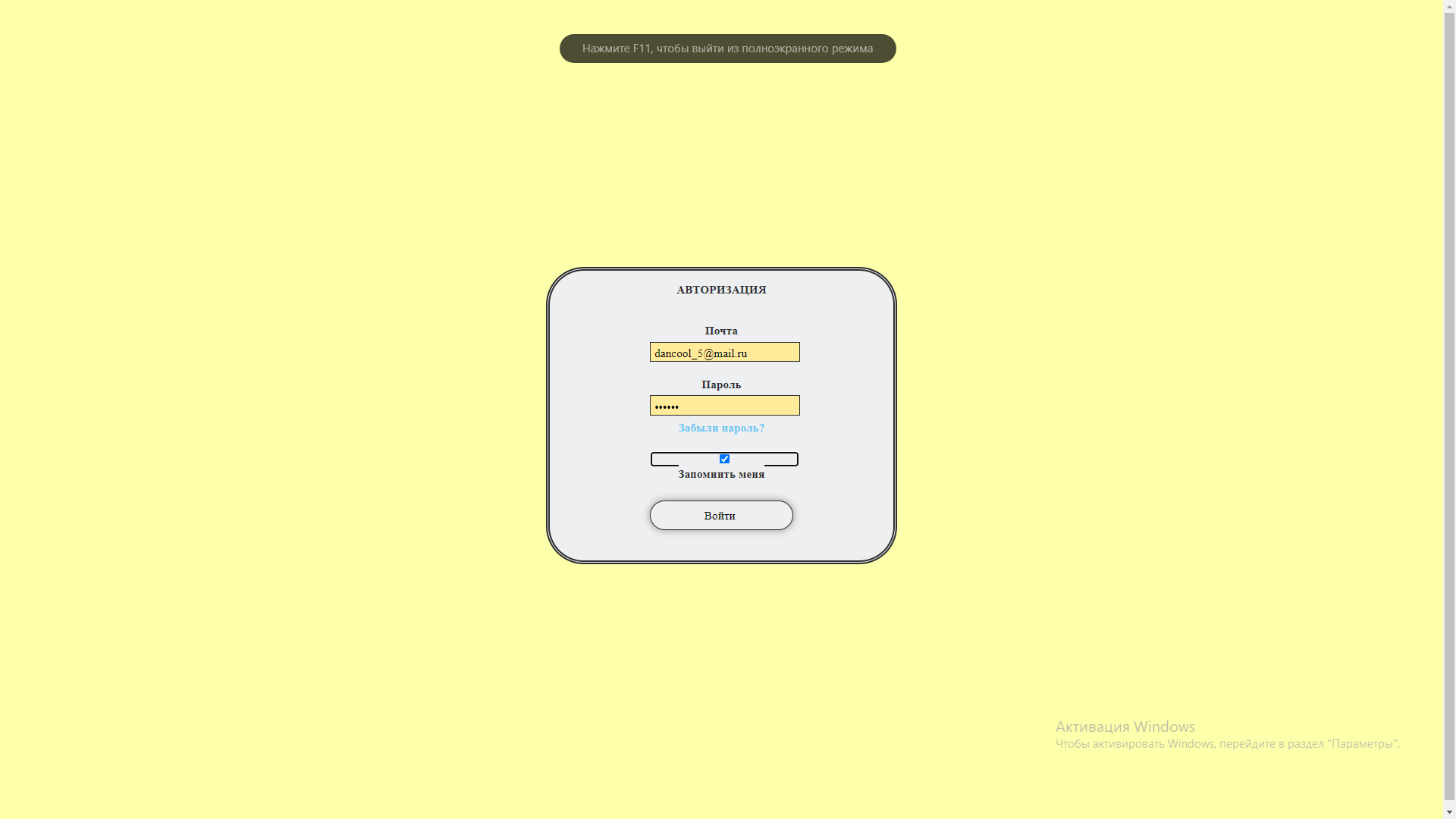


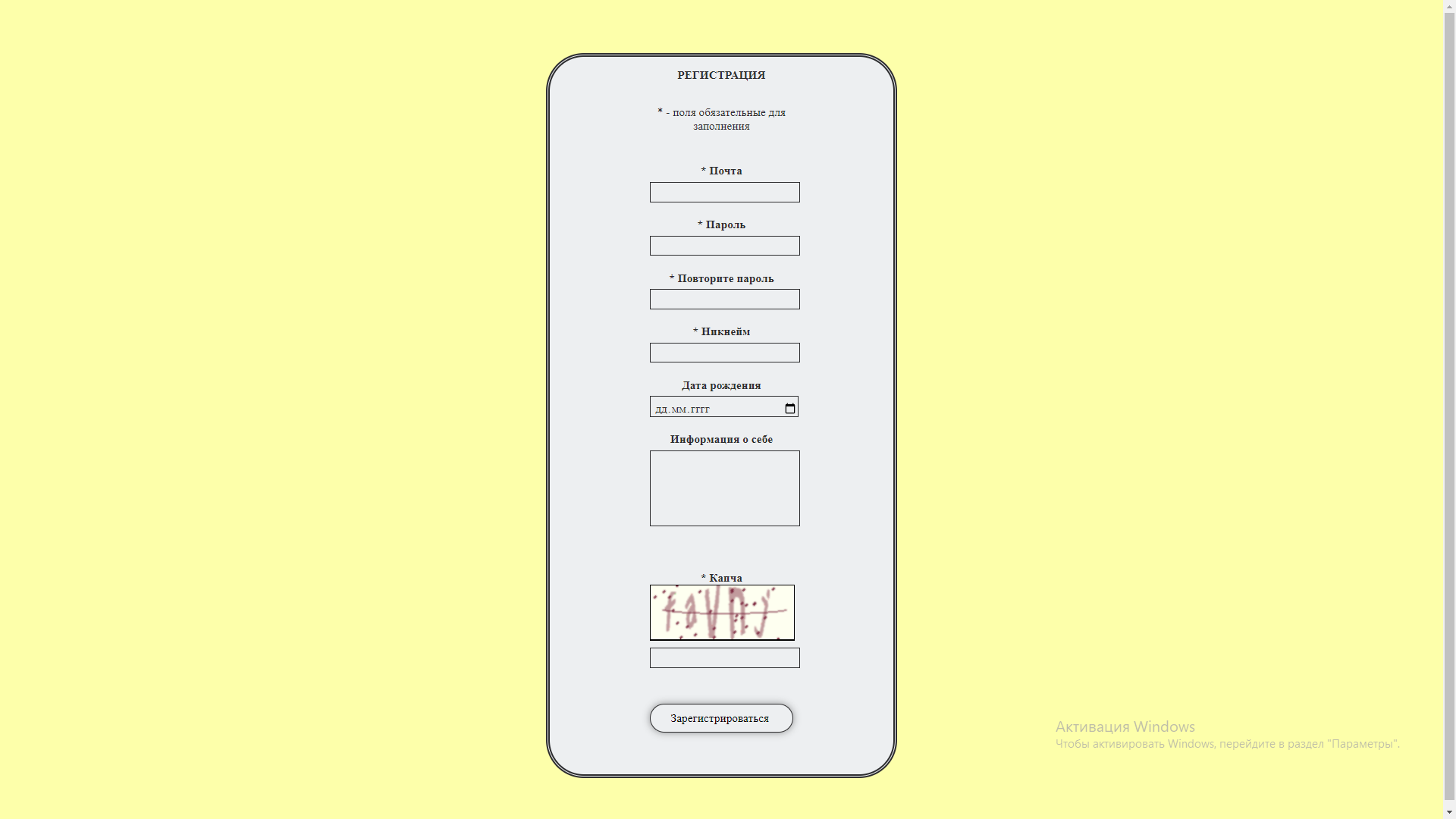




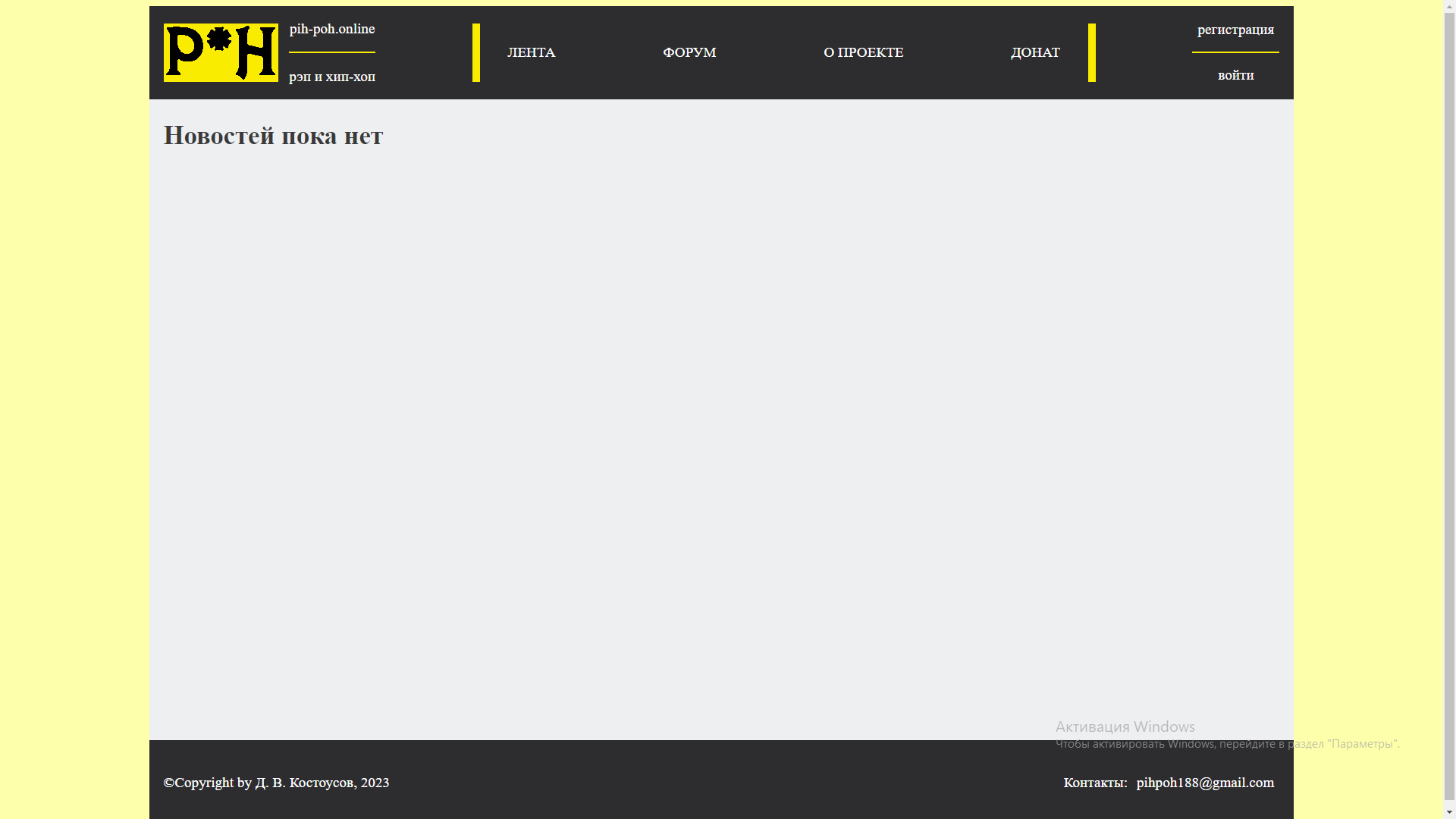








\



**Заключение**

В ходе работы были изучены и использованы все средства, описанные в главной части. Также я понял, что сфера Web разработки действительно интересна мне, но была изучена лишь малая часть этой области IT. Мне бы хотелось и дальше в ней развиваться, тем более, я считаю, что данный проект нельзя назвать окончательно завершенным. Есть еще несколько аспектов, которые я бы хотел доделать, а именно:

1. Исправление багов
2. Исправление верстки, а также реализация адаптивной верстки сайта
3. Улучшение системы администрирование
4. Возможность прикреплять файлы в сообщениях и статьях
5. Улучшение редактора для написания сообщений и статей

В целом опыт, полученный в ходе разработки данного проекта, считаю успешный и довольно занимательный и, исходя из всего вышенаписанного, хотелось бы и дальше изучать сферу Web-разработки, например, в сторону фронтенда, чтобы этот и дальнейшие проекты визуально были более привлекательными, а также в сторону бекенда, конкретно в написании защищенных сайтов и приложений, способных выдержать большую нагрузку.

Ссылка на Github репозиторий проекта: https://github.com/dvkostousov/pih-poh.online