

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра №806 «Вычислительная математика и программирование»

**Курсовая работа
по курсу «Базы данных»**

Сервис для подбора и покупки недвижимости

Выполнил: В.А. Комбаров
Группа: 8О-312Б
Преподаватель: А.В. Малахов

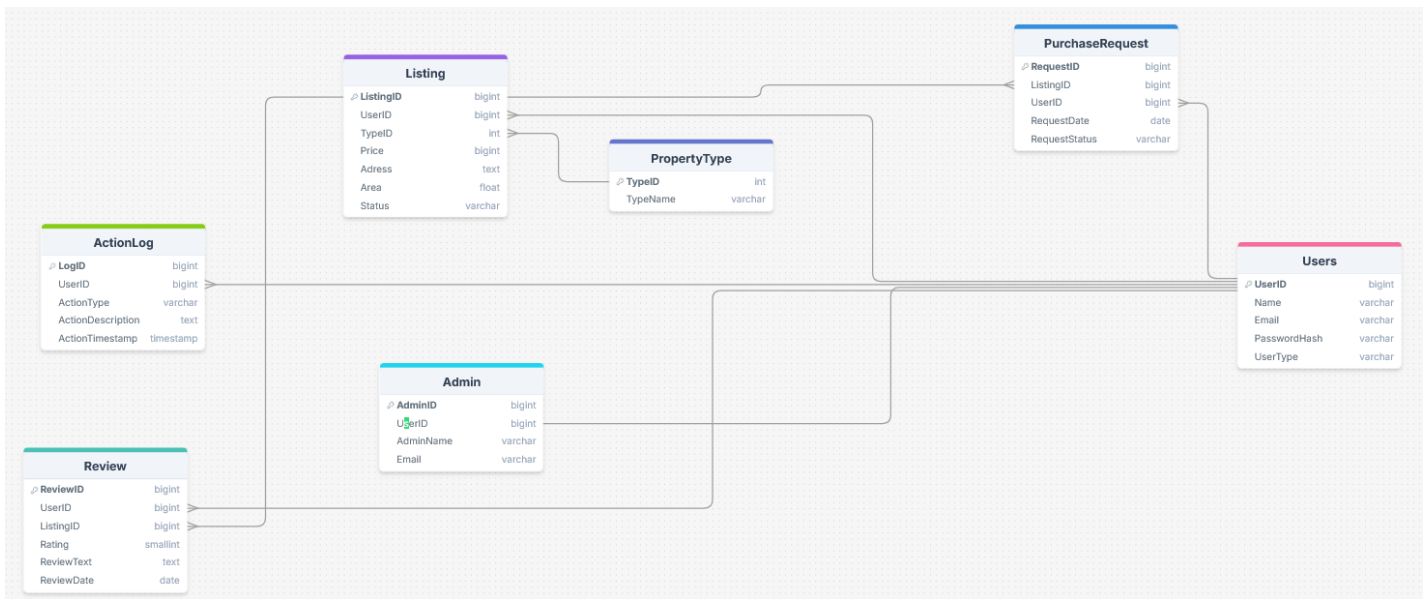
Москва, 2024

Обзор предметной области

Сервис для подбора и покупки недвижимости — это онлайн-платформа для пользователей, которые хотят приобрести желаемую недвижимость, а также для агентов, которые могут эту недвижимость порекомендовать своим клиентам. Платформа позволяет пользователям размещать объявления, отслеживать их текущий статус (объявление может быть продано или снято с публикации), отправлять запросы на покупку, оставлять отзывы на объявления. В системе предусмотрены две роли пользователей: обычный пользователь и администратор, который может удалять ставшие ненужными объявления и имеет доступ к таблице логов сервиса, а также ко всей базе данных.

Сервис ориентирован на пользователей, которым необходимо быстро и достаточно удобно приобрести недвижимость желаемого типа. Система удобно спроектирована, предоставляет простые опции для оформления покупки и обратной связи, а также поддерживает различные роли пользователей для эффективного использования сервиса и поддержания в нём порядка.

Схема и описание базы данных



СУБД – PostgreSQL

Таблица **users** — хранит информацию о пользователях системы. Содержит следующие поля:

- **userid**— уникальный идентификатор пользователя.
- **name** — имя пользователя.
- **email** – электронная почта пользователя.
- **passwordhash** — хэш пароля пользователя.
- **usertype**- роль пользователя (“user”, “admin”).

Таблица **propertytype** — хранит информацию о типах недвижимости. Содержит следующие поля:

- **typeid**— уникальный идентификатор типа.
- **typename**— наименование типа.

Таблица **admin** — хранит информацию об администраторах сервиса. Содержит следующие поля:

- **adminid**— уникальный идентификатор администратора.
- **userid**— идентификатор пользователя (ссылается на таблицу users).
- **adminname**— имя администратора.
- **email**— электронная почта администратора.

Таблица **listing** — хранит информацию об объявлениях. Содержит следующие поля:

- listingid— уникальный идентификатор объявления.
- userid— идентификатор пользователя (ссылается на таблицу users).
- typeid— идентификатор типа недвижимости (ссылается на таблицу propertytype).
- price— цена объекта недвижимости.
- address— адрес объекта.
- area— площадь объекта.
- status— статус объявления (“активно”, “продано”, “снято с публикации”).

Таблица **purchaserequest** — хранит информацию о запросах на покупку конкретного объекта. Содержит следующие поля:

- requestid— уникальный идентификатор запроса.
- listingid— идентификатор объявления (ссылается на таблицу listing).
- userid— идентификатор пользователя (ссылается на таблицу users).
- requestdate— дата создания запроса.
- requeststatus— статус запроса (“рассматривается”, “принят”, “отклонён”).

Таблица **review** — хранит информацию об отзывах. Содержит следующие поля:

- reviewid— уникальный идентификатор отзыва.
- userid— идентификатор пользователя (ссылается на таблицу users).
- listingid— идентификатор объявления (ссылается на таблицу listing).
- rating— выставленный рейтинг (может быть только от 1 до 10).
- reviewtext— текст отзыва.
- reviewdate— дата оставления отзыва.

Таблица **actionlog** — хранит информацию о логах всех пользователей. Содержит следующие поля:

- logid— уникальный идентификатор записи.
- userid— идентификатор пользователя (ссылается на таблицу users).
- actiontype— тип действия, о котором повествует запись.
- actiondescription— описание действия.
- actiontimestamp— время записи.

Триггеры / функции: в роли триггера выступает таблица actionlog, которая доступна только администраторам. Эта таблица заполняется автоматически во время какого-либо действия какого-либо пользователя, будь то регистрация / логин или создание / просмотр объявлений. После каждого успешного выполнения запроса на сервере автоматически вызывается функция, заполняющая таблицу actionlog соответствующими данными.

Нормализация: Все таблицы соответствуют третьей нормальной форме.

Описание приложения

1. Регистрация и авторизация пользователей

- Покупатели начинают использование сервиса с регистрации на платформе. Для этого им необходимо указать имя, электронную почту и пароль. В случае успешной регистрации введённая информация сохраняется в базе данных.
- После регистрации пользователь может войти в систему, используя свои email и пароль. Это позволяет ему получить доступ к списку объявлений и начать полноценную работу на платформе.

Name: Email: Password:

[Already have an account? Login here.](#)

Email: Password:

[Don't have an account? Register here.](#)

🌐 127.0.0.1:5000

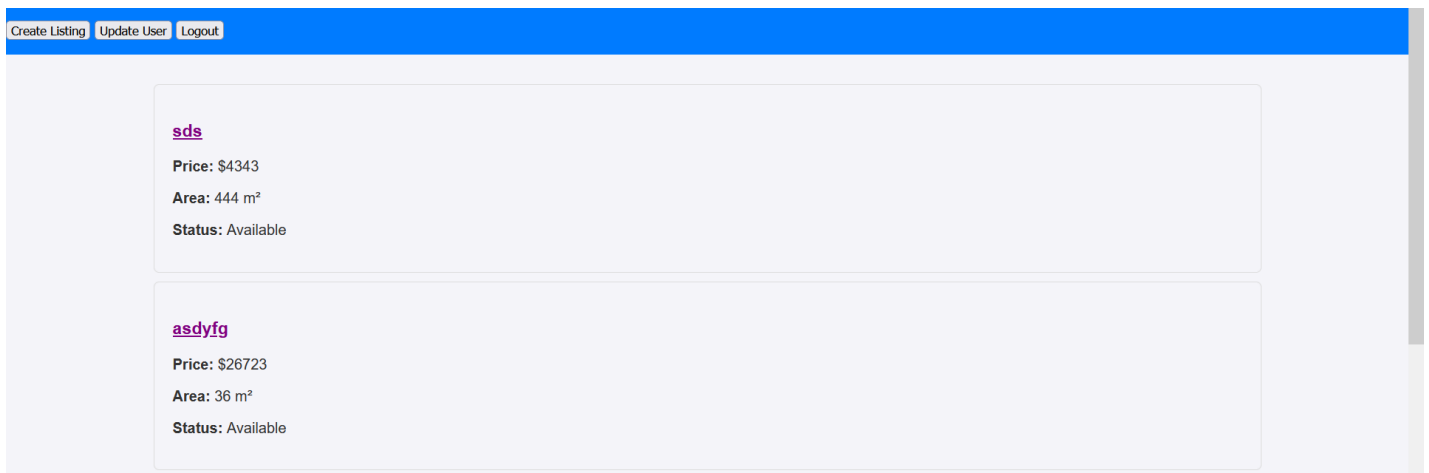
Login successfull!

2. Страница объявлений

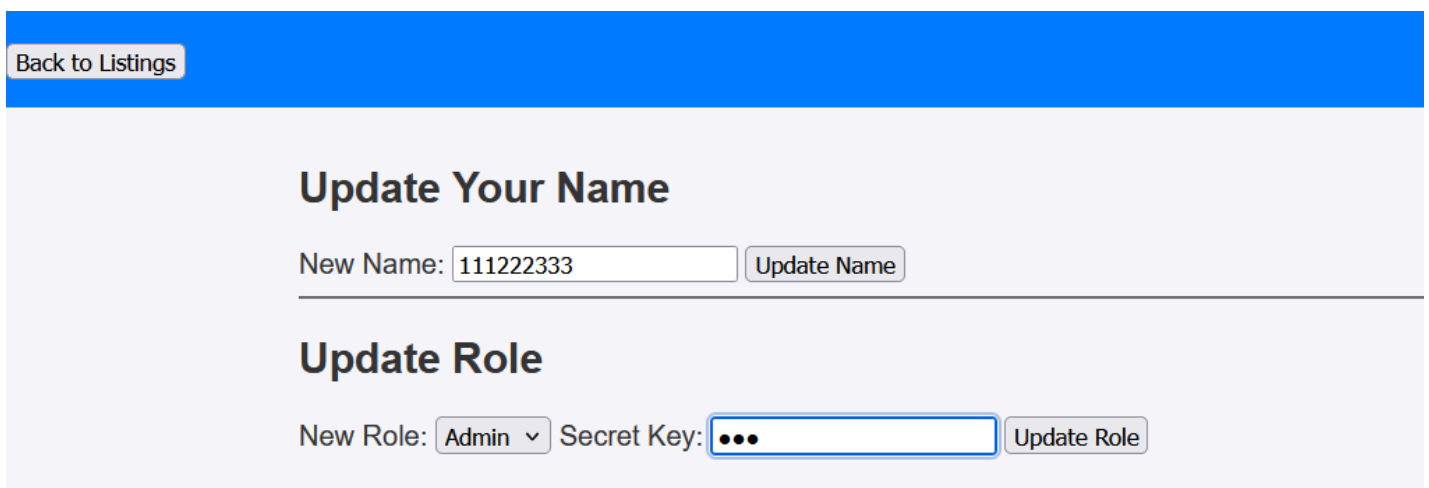
- После успешной авторизации пользователь попадает на страницу с объявлениями. С этой страницы можно перейти на следующие вкладки:
- Страница конкретного объявления — для дальнейших действий с конкретным объектом недвижимости.
- Кнопка логута – возможность выйти из своего профиля
- Страница создания нового объявления
- Вкладка, где возможно изменить пользователя

3. Изменение пользователя

- На странице изменения пользователя можно быстро сменить своё имя, а также поменять свой статус, однако для этого нужно ввести секретный ключ, который знают только администраторы. Предполагается, что этот ключ не будет разглашаться на широкую аудиторию, в случае форс-мажорной ситуации ключ можно будет легко поменять в коде приложения (файл .env)



Фрагмент страницы с объявлениями (введены тестовые неправдоподобные данные)



Фрагмент страницы с изменением пользователя

4. Создание объявления

При создании объявления нужно указать:

- Тип недвижимости (id поставлен в соответствие конкретному типу, например, id = 1 – значит тип – квартира)
- Цена
- Адрес
- Площадь
- Также статус объекта автоматически определяется как “доступный”
- После ввода всех параметров покупатель может нажать кнопку "Создать объявление", тогда его перенаправит на страницу объявлений.

Create New Listing

Type

Price

Address

Area

Status

5. Операции с объявлением и объектом

На странице объявления можно:

- Отправить запрос на покупку, а также посмотреть все свои запросы
- Изменить / удалить объявление, если оно принадлежит создателю объявления плюс для полного удаления объявления нужно обладать правами администратора. Как правило, объявления должны удаляться после того, как объект недвижимости был продан.

[Back to Listings](#)

Address: Area: Price: Status: Type:

🌐 127.0.0.1:5000

Listing updated successfully.

- Написать отзыв на объявление, а также посмотреть все отзывы на него
- Также можно перейти обратно на страницу объявлений

[Back to Listing](#)

Rating: 10 / 10

Review: Great

Reviewed by User ID: 13

Date: 2024-12-19

[Edit](#)

[Delete](#)

Фрагмент страницы с отзывами на объявлениях (на данное объявление всего 1 отзыв)

[Back to Listings](#)

[Read Listing Reviews](#)

[View Your Purchase Requests](#)

main

Price: \$15000000


Area: 55 m²

Status: Available

[Request Purchase](#)

[Edit Listing](#)

[Delete Listing](#)

Rating (1 to 10): 10 

Leave a Review

Great

[Submit Review](#)

Фрагмент страницы конкретного объявления

[Back to Listings](#)

Your Purchase Requests

- **Listing ID:** 24

Status: Pending

Request Date: 19.12.2024, 03:00:00

[Update Request Status](#)

- **Listing ID:** 20

Status: Pending

Request Date: 19.12.2024, 03:00:00

[Update Request Status](#)

Фрагмент страницы с запросами пользователя на покупку

Со страницы запросов на покупку можно перейти на страницу изменения одного запроса, так можно изменить каждый из них. Это разработано прежде всего для структурирования данных о покупках и удобства самих покупателей, если покупатель изменит запрос на принятый, но на самом деле продавец против, то изменённый запрос ни на что не влияет, последнее слово всё рано за продавцом объекта.

На странице отзывов можно изменять и удалять отзывы, но только свои.

[Back to Reviews](#)

Rating (1-10):

Review Text:

good

[Update Review](#)

Фрагмент страницы с изменением отзыва

6. Роли пользователей

Обычный пользователь имеет возможность:

- Создавать, редактировать свои объявления, в том числе меняя статус.
- Управлять своими персональными данными.
- Оставлять отзывы, делать запросы на покупку, имея возможность редактировать их и удалять неудачные отзывы

Администратор обладает расширенными правами:

- Полноценное удаление объявления (например, проданные или подозрительные или снятые с публикации и долго не возвращающиеся на рынок)
- Полный доступ к базе данных: возможность отслеживать действия пользователей на платформе, смотреть список всех зарегистрированных пользователей, отслеживать все запросы на покупку того или иного объекта и т.д.

```
estimatedb=# select * from actionlog;
```

logid	userid	actiontype	actiondescription	actiontimestamp
309	15	register	New user registered with email: asd@example.com	2024-12-13 11:03:53.690904
310	15	login	User 15 logged in.	2024-12-13 11:04:18.857755
311	15	update	User 15 updated their name from 'Vlad' to 'qwerty'	2024-12-13 11:05:15.647905
312	15	update	User 15 updated their name from 'qwerty' to 'Vlad'	2024-12-13 11:05:44.402303
313	15	update_role	User 15 updated their role from user to admin	2024-12-13 11:05:53.710283
314	15	logout	User 15 logged out.	2024-12-13 11:06:35.639159
315	15	login	User 15 logged in.	2024-12-13 11:06:55.205295
316	15	create	User 15 created a listing with ID 22	2024-12-13 11:07:26.604184
317	15	update	User 15 updated a listing with ID 22	2024-12-13 11:08:11.286786
318	15	delete	User 15 deleted a listing with ID 22	2024-12-13 11:08:32.949021
319	15	create	User 15 created a purchase request for listing 19	2024-12-13 11:08:43.771539
320	15	read	User 15 accessed their purchase requests	2024-12-13 11:08:48.25241

:04.806904

Фрагмент таблицы actionlog непосредственно в самой базе данных

Сервер (backend) написан на Python с использованием Flask и вспомогательных инструментов, взаимодействующих с SQL. Frontend был написан с помощью HTML, JavaScript и CSS.

Также предусмотрена обработка различных типов ошибок, таких как:

1. Ошибки входа в систему

- Ошибка неверного логина / пароля.
- Ошибка неверного указания данных при регистрации (например, некорректный email).
- Ошибка отсутствия прав доступа (не вошедший через логин пользователь практически ничего не сможет сделать на платформе, обычный пользователь не сможет удалить объявление)

2. Ошибки при работе с объявлениями

- Нельзя изменить / удалить то, что не принадлежит пользователю.
- Все данные при создании / изменении объявления или смежных компонент

должны быть корректными и не пустыми.

3. Ошибки с запросом на покупку

- Нельзя сделать запрос на покупку своего объявления.

4. Ошибки работы с базой данных

- Ошибка соединения с базой данных.
- Ошибка отсутствия данных.

Репозиторий на Github



Выводы

В ходе работы был успешно реализован сервис для подбора и покупки недвижимости, который включает в себя функционал для регистрации и авторизации пользователей, работы со своими и чужими объявлениями, работы с отзывами, изменения роли и имени пользователя, подачи и редактирования запросов на покупку. Была построена структурированная база данных с использованием СУБД PostgreSQL, что позволило эффективно управлять данными о пользователях, объявлениях, типах недвижимости и т.д. Также в своей БД есть специальная таблица с логами всех действий на платформе. Все процессы были реализованы с использованием популярных, достаточно простых в обучении и проверенных технологий, что позволило пошагово создать систему с понятным интерфейсом и хорошей производительностью.