

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

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

**Итоговый проект
по курсу «Базы данных»**

Сервис бронирования авиабилетов

Выполнил: Халимов И. И.

Группа: М8О-308Б-22

Преподаватель: Малахов А. В.

Москва, 2024

Оглавление

1. Постановка задачи
2. Описание
3. Описание и схема моделей данных
4. Описание моделей уровня инфраструктуры
5. Разработка запросов к базе данных
6. Графический интерфейс
7. Вывод
8. Список использованных источников

Постановка задачи

Общие требования к итоговому проекту:

- При реализации курсового проекта допускается только использование СУБД PostgreSQL.
- Необходимо выбрать предметную область для создания базы данных. Выбранная предметная область должна быть уникальной для всего потока, а не только в рамках учебной группы.
- Необходимо описать модели предметной области и уровня инфраструктуры и их назначение в рамках реализуемого проекта (минимальное количество моделей предметной области и уровня инфраструктуры - 5). Также необходимо выполнить проектирование логической структуры базы данных. Все таблицы, связанные с описанными моделями предметной области, должны находиться в 3NF или выше. База данных должна иметь минимум 7 таблиц.
- Клиентское приложение должно быть в виде WEB или оконного приложения.
- Необходимо организовать различные роли пользователей и права доступа к данным (например: администратор, редактор, рядовой пользователь). Клиентское приложение, взаимодействующее с базой данных, должно предоставлять функционал для авторизации пользователя по логину и паролю (хранение непосредственно пароля в базе данных запрещено, надо хранить HASH от этого пароля и проверять его).
- При разработке функционала базы данных следует организовать логику обработки данных не на стороне клиента (Frontend), а на стороне серверного приложения (Backend). Все обработки «SQL запросов», «работа бизнес-логики должны находиться в BACKEND части. FRONT только для отображения.

- Запросы должны быть асинхронны. То есть, при нажатии на форму она не должна зависать. При нажатии на форму одним пользователем, другой должен иметь возможность свободно пользоваться приложением. То есть действия разных пользователей независимы.

Описание

Проект представляет собой веб-сервис, который помогает футболистам и любителям футбола находить футбольные клубы на основе различных критериев. Пользователи могут фильтровать клубы по городу, стадиону, рейтингу и другим параметрам, чтобы выбрать наиболее подходящие варианты. Сервис также предоставляет детальную информацию о клубах, включая данные об игроках, выигранных трофеях и финансовых показателях. Интуитивно понятный интерфейс позволяет пользователям легко взаимодействовать с базой данных для получения и анализа данных.

Вся информация о странах, городах, стадионах, клубах, игроках и трофеях хранится в базе данных PostgreSQL. Веб-интерфейс для доступа к сервису реализован с использованием Streamlit, а для взаимодействия с базой данных используется библиотека psycopg2.

Функциональность проекта включает:

- Отображение списка футбольных клубов с подробной информацией.
- Фильтрацию клубов по различным атрибутам, таким как рейтинг, город и стадион.
- Предоставление списка игроков, связанных с каждым клубом.
- Отображение выигранных клубами трофеев с указанием года победы.
- Интерактивный механизм поиска, позволяющий пользователям уточнять результаты в реальном времени.

Описание и схема моделей предметной области

Проект представляет собой сервис для поиска футбольных клубов, который позволяет игрокам фильтровать и находить подходящие футбольные клубы по различным параметрам, таким как регион, уровень, наличие трофеев и др. Вся информация хранится в базе данных PostgreSQL, а взаимодействие с базой данных осуществляется через psycopg2. Веб-интерфейс реализован с использованием Streamlit.

Модели предметной области

Country (Страны)

Таблица содержит информацию о странах и их регионах.

Атрибуты:

country_id: Уникальный идентификатор страны.

name: Название страны.

region: Регион, к которому относится страна.

Связи:

Связана с таблицей City через столбец country_id.

```
CREATE TABLE Country (  
    country_id SERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    region VARCHAR(255)  
);
```

City (Города)

Таблица содержит информацию о городах.

Атрибуты:

city_id: Уникальный идентификатор города.

name: Название города.

country_id: Уникальный идентификатор страны, к которой относится город.

Связи:

Связана с таблицей Stadium через столбец city_id.

```
CREATE TABLE City (  
    city_id SERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    country_id INT NOT NULL REFERENCES Country(country_id)  
);
```

Stadium (Стадионы)

Таблица содержит информацию о стадионах.

Атрибуты:

stadium_id: Уникальный идентификатор стадиона.

name: Название стадиона.

type_of_pitch: Тип покрытия поля.

city_id: Уникальный идентификатор города, в котором расположен стадион.

Связи:

Связана с таблицей Club через столбец stadium_id.

```
CREATE TABLE Stadium (  
    stadium_id SERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    type_of_pitch VARCHAR(255),  
    city_id INT NOT NULL REFERENCES City(city_id)  
);
```

Club (Клубы)

Таблица содержит информацию о футбольных клубах.

Атрибуты:

club_id: Уникальный идентификатор клуба.

name: Название клуба.

stadium_id: Уникальный идентификатор стадиона, где базируется клуб.

rating: Рейтинг клуба.

Связи:

Связана с таблицей Player через столбец club_id.

```
CREATE TABLE Club (  
  club_id SERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  stadium_id INT REFERENCES Stadium(stadium_id),  
  rating INT CHECK (rating BETWEEN 0 AND 100)  
);
```

Player (Игроки)

Таблица содержит информацию об игроках.

Атрибуты:

player_id: Уникальный идентификатор игрока.

name: Имя игрока.

age: Возраст игрока.

salary: Зарплата игрока.

position: Позиция игрока на поле.

club_id: Уникальный идентификатор клуба, в котором играет игрок.

Связи:

Связана с таблицей Club через столбец club_id.

```
CREATE TABLE Player (  
  player_id SERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  age INT CHECK (age > 0),  
  salary NUMERIC(10, 2) CHECK (salary >= 0),  
  position VARCHAR(255),  
  club_id INT REFERENCES Club(club_id)  
);
```

Trophy (Трофеи)

Таблица содержит информацию о трофеях.

Атрибуты:

trophy_id: Уникальный идентификатор трофея.

name: Название трофея.

prize_fund: Призовой фонд трофея.

Связи:

Связана с таблицей Club_Trophy через столбец trophy_id.

```
CREATE TABLE Trophy (  
    trophy_id SERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    prize_fund NUMERIC(10, 2) CHECK (prize_fund >= 0)  
);
```

Club_Trophy (Трофеи клубов)

Таблица связывает клубы и их завоеванные трофеи.

Атрибуты:

club_id: Уникальный идентификатор клуба.

trophy_id: Уникальный идентификатор трофея.

year_won: Год завоевания трофея.

Связи:

Связана с таблицами Club и Trophy через соответствующие столбцы.

```
CREATE TABLE Club_Trophy (  
    club_id INT NOT NULL REFERENCES Club(club_id),  
    trophy_id INT NOT NULL REFERENCES Trophy(trophy_id),  
    year_won INT CHECK (year_won > 0),  
    PRIMARY KEY (club_id, trophy_id, year_won)  
);
```

Users (Пользователи)

Таблица содержит информацию о пользователях системы.

Атрибуты:

user_id: Уникальный идентификатор пользователя.

username: Уникальное имя пользователя.

password: Хэш пароля пользователя.

role: Роль пользователя (например, 'user').

Связи:

Не связана с другими таблицами.

```
CREATE TABLE Users (  
  user_id SERIAL PRIMARY KEY,  
  username VARCHAR(50) UNIQUE NOT NULL,  
  password TEXT NOT NULL,  
  role VARCHAR(10) DEFAULT 'user'  
);
```

Общая схема моделей предметной области:

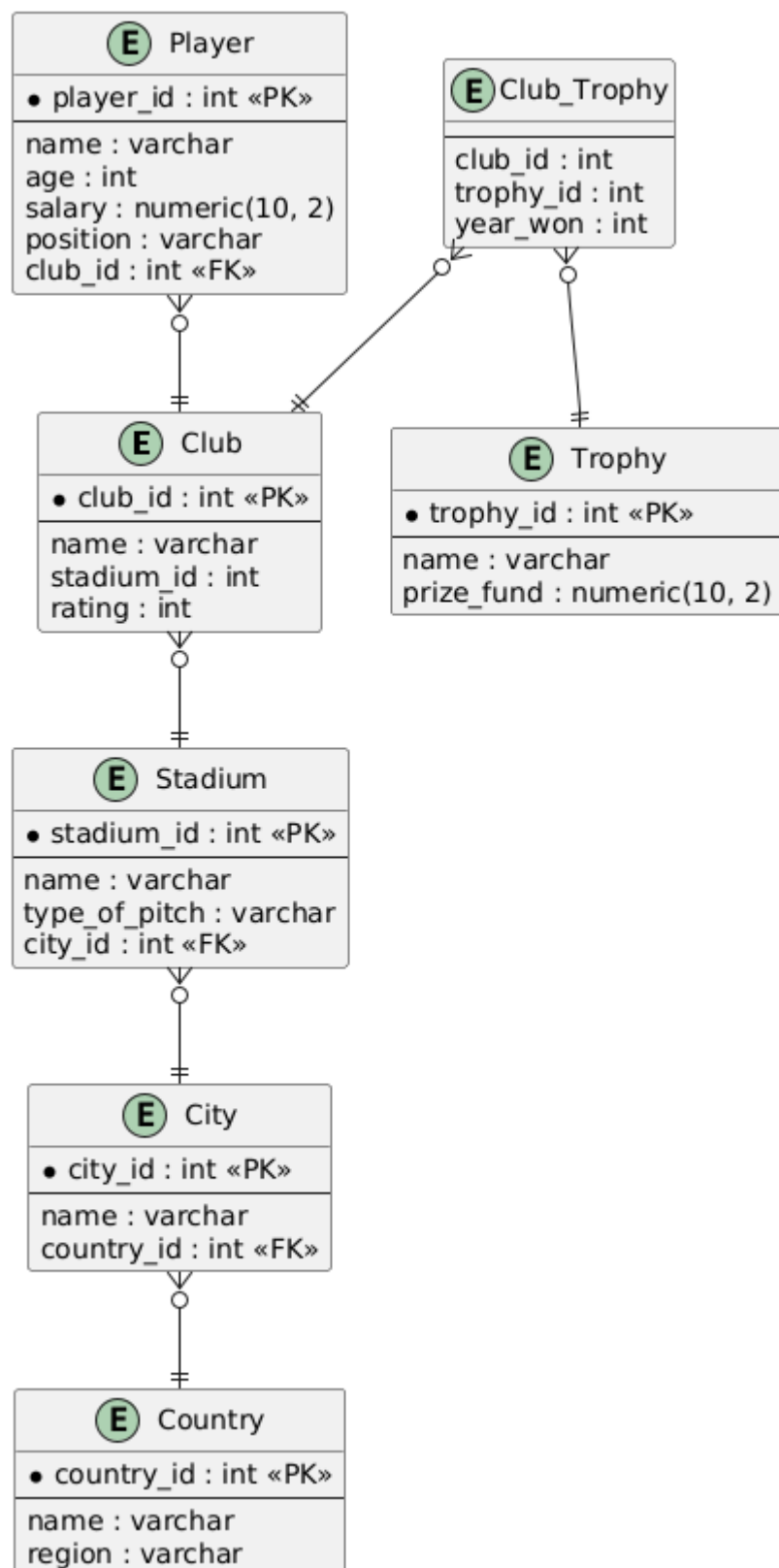


рис.1 “Схема моделей предметной области”

Описание моделей уровня инфраструктуры

Сервис авторизации/регистрации

Сервис авторизации и регистрации пользователей предоставляет функциональность для создания новых аккаунтов, проверки учетных данных и управления доступом к системе. Важной частью сервиса является безопасное хранение и обработка пользовательских паролей.

Основные функции:

- Регистрация новых пользователей.
- Авторизация пользователей (проверка логина и пароля).
- Хранение паролей в зашифрованном виде с использованием алгоритмов хеширования.
- Восстановление доступа и изменение паролей.
- Управление ролями пользователей, включая "футболист" и "администратор".

Сервис фильтрации и подбора клуба

Сервис обеспечивает интерактивный поиск футбольных клубов, основываясь на заданных пользователем критериях. Пользователи могут выбирать клубы, соответствующие их предпочтениям, и просматривать подробную информацию.

Основные функции:

- Фильтрация клубов по заданным параметрам (уровень лиги, трофеи, финансовое состояние и пр.).
- Предоставление сведений о клубах, включая их достижения, местоположение и состав команды.
- Обеспечение быстрого отклика на запросы пользователя.

Сервис управления базой данных

Сервис взаимодействует с базой данных PostgreSQL, обеспечивая выполнение операций на стороне сервера, таких как добавление, обновление и удаление записей, а также управление структурой базы данных.

Основные функции:

- Обработка запросов на выборку данных о клубах, футболистах, трофеях и других объектах.
- Поддержание базы данных в нормальной форме (3НФ и выше).
- Автоматизация процедур администрирования данных, таких как оптимизация запросов и индексация.

Дополнение о функционале "Админа" и "Журналиста"

Функционал "Журналиста"

Журналист обладает расширенными правами по сравнению с обычным пользователем. Его основной задачей является поддержание актуальности данных о трофеях футбольных клубов.

Основные функции:

- Добавление информации о новых трофеях, выигранных клубами.
- Обновление данных о трофеях клубов, таких как название турнира, год победы и описание достижения.
- Просмотр и редактирование уже существующей информации о трофеях (при необходимости, в рамках своих прав).

Журналист взаимодействует с системой через удобный интерфейс, который позволяет минимизировать риск ошибок при внесении данных.

Функционал "Админа"

Администратор имеет максимальный доступ к функциональности системы. Он может выполнять все действия, доступные пользователю и журналисту, а также имеет дополнительные привилегии для управления базой данных.

Основные функции:

- Управление данными о трансферах игроков:
 - Добавление информации о переходе игрока в другой клуб.
 - Указание условий трансфера, таких как сумма сделки
- Выполнение всех действий журналиста:

Разработка запросов к базе данных

Основные запросы к базе данных, используемые для реализации сервиса бронирования авиабилетов.

Добавление нового пользователя

```
INSERT INTO users (username, password, role)
VALUES (%(username)s,
%(password)s, %(role)s);
```

Запрос добавляет нового пользователя в таблицу users, записывая его имя, фамилию, email, хеш пароля и роль. По умолчанию роль устанавливается как "user".

Получение списка клубов с указанием города и стадиона

```
SELECT c.name, c.rating, s.name, ci.name
FROM Club c
      JOIN Stadium s USING(stadium_id)
      JOIN City ci USING(city_id)
      JOIN COUNTRY co USING(country_id)
```

Получение клубов из определённой страны

```
SELECT c.name, ci.name  
FROM Club c  
JOIN Stadium s ON c.stadium_id = s.stadium_id  
JOIN City ci ON s.city_id = ci.city_id  
JOIN Country co ON ci.country_id = co.country_id  
WHERE co.name = %s  
ORDER BY 1;
```

Запрос находит все клубы расположенные в выбранной пользователем стране и выводит название клуба вместе с названием города

Добавление выигрыша трофея в клуб

```
INSERT INTO Club_Trophy (club_id, trophy_id, year_won)  
VALUES (%s, %s, %s);
```

Запрос добавляет новый трофей в таблицу Club_Trophy для отображения победы на определенном турнире для выбранного клуба

Получение списка городов

```
SELECT DISTINCT name FROM city;
```

Запрос извлекает уникальные города из таблицы city, чтобы предоставить список всех доступных городов для выбора пользователю.

Удаление пользователя

```
DELETE FROM users  
WHERE username = %(username)s;
```

Запрос удаляет рейс из базы данных по его flight_id

Получение списка клубов после фильтрации

```
SELECT c.name, s.name AS stadium, ci.name AS city, c.rating, COUNT(player_id)  
FROM Club c  
JOIN Stadium s ON c.stadium_id = s.stadium_id  
JOIN City ci ON s.city_id = ci.city_id  
JOIN Player p ON c.club_id = p.club_id  
WHERE c.rating >= %(rating)s AND ci.name = %(ci_name)s AND s.name = %(s_name)s  
GROUP BY 1, 2, 3, 4  
ORDER BY c.rating DESC;
```

Запрос отбирает только те записи, которые удовлетворяют условиям (минимальный рейтинг, город клуба, стадион клуба) и группирует записи для подсчёта количества игроков в каждом клубе

Графический интерфейс

1. Главная страница

Навигация

Действие

☒ Войти

☐ Регистрация

Авторизация

Имя пользователя

Пароль

👁

Войти

2. Авторизация

Авторизация

Имя пользователя

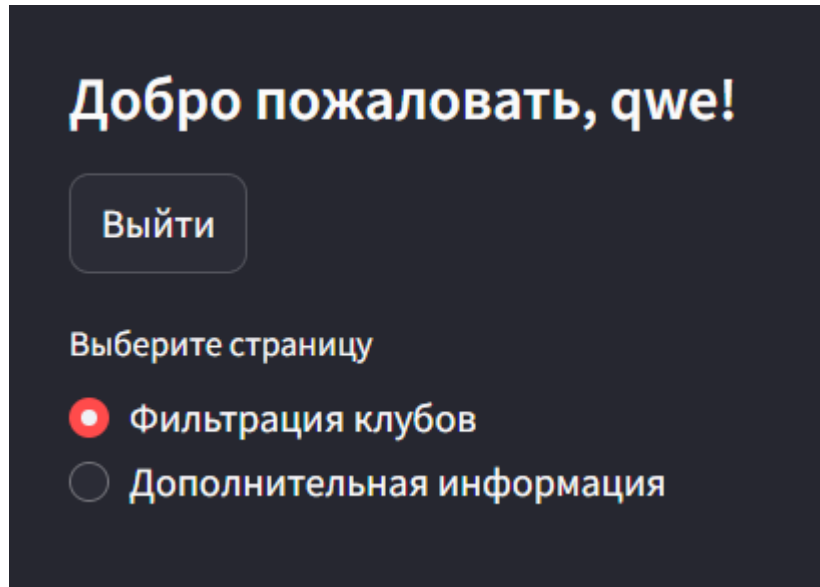
Пароль

👁

Войти

Интерфейс для пользователя:

Основные функции



1. Фильтрация клубов

Добро пожаловать, isma!

Выйти

Выберите страницу

- ☒ Фильтрация клубов
- ☐ Дополнительная информация
- ☐ Добавить трофей клубу
- ☐ Оформить трансфер игрока в другой клуб

Фильтры

Минимальный рейтинг клуба

0 84 100

Город

Все

Стадион

Все

Применить фильтры

Результаты фильтрации:

	Название клуба	Стадион	Город	Рейтинг	Кол-во игроков
0	Real Madrid	Santiago Bernabéu	Madrid	98	9
1	Bayern Munich	Allianz Arena	Munich	95	9
2	Paris Saint-Germain	Parc des Princes	Paris	94	11
3	AC Milan	San Siro	Milan	93	5
4	Ajax	Johan Cruyff Arena	Amsterdam	92	4
5	Chelsea FC	Wembley Stadium	London	90	9
6	Benfica	Estádio da Luz	Lisbon	89	4
7	Flamengo	Maracanã	Rio de Janeiro	88	9
8	Club América	Estadio Azteca	Mexico City	86	5
9	Boca Juniors	La Bombonera	Buenos Aires	85	9
10	Spartak Moscow	Luzhniki Stadium	Moscow	84	1

2. Список игроков клуба

Добро пожаловать, isma!

Выйти

Выберите страницу

Фильтрация клубов

Дополнительная информация

Добавить трофей клубу

Оформить трансфер игрока в другой клуб

Выберите тип информации

Что вы хотите узнать?

Список игроков клуба

Дополнительная информация о клубах

Выберите клуб из списка

AC Milan

Показать список

	Имя игрока	Позиция	Возраст	Зарплата
0	Elijah Brown	Goalkeeper	27	65000
1	Liam Stone	Forward	25	60000
2	Noah Griffin	Midfielder	28	55000
3	Oliver Wood	Defender	22	47000
4	Zlatan Ibrahimović	Forward	41	1500000

	Название клуба	Рейтинг клуба	Количество игроков
0	AC Milan	93	5

3. Список трофеев клуба

Добро пожаловать, isma!

Выйти

Выберите страницу

Фильтрация клубов

Дополнительная информация

Добавить трофей клубу

Оформить трансфер игрока в другой клуб

Выберите тип информации

Что вы хотите узнать?

Список трофеев клуба

Дополнительная информация о клубах

Выберите клуб из списка

AC Milan

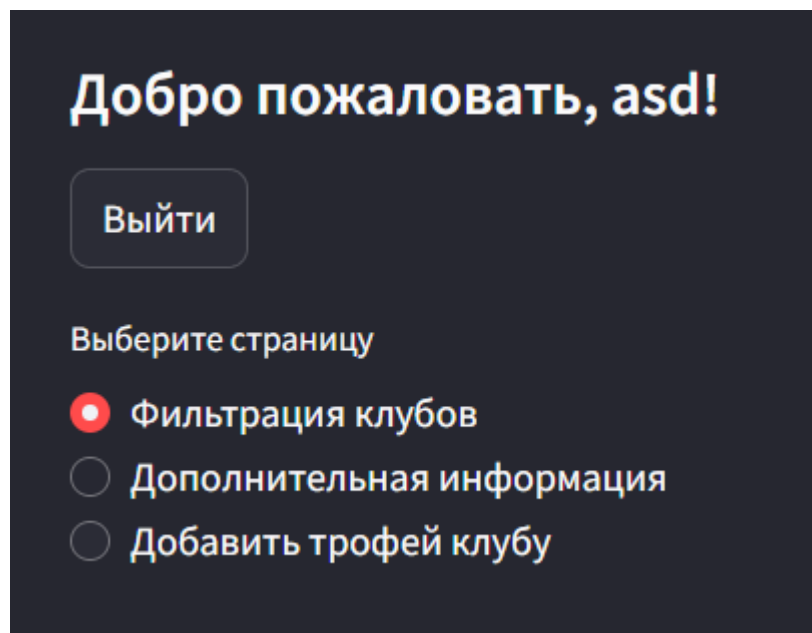
Показать список

	Название трофея	Год получения	Призовой фонд
0	UEFA Champions League	2019	15000000
1	Serie A	2021	9000000
2	Serie A	2023	9000000

Количество трофеев у клуба AC Milan: 3

Интерфейс для Журналиста

Основные функции



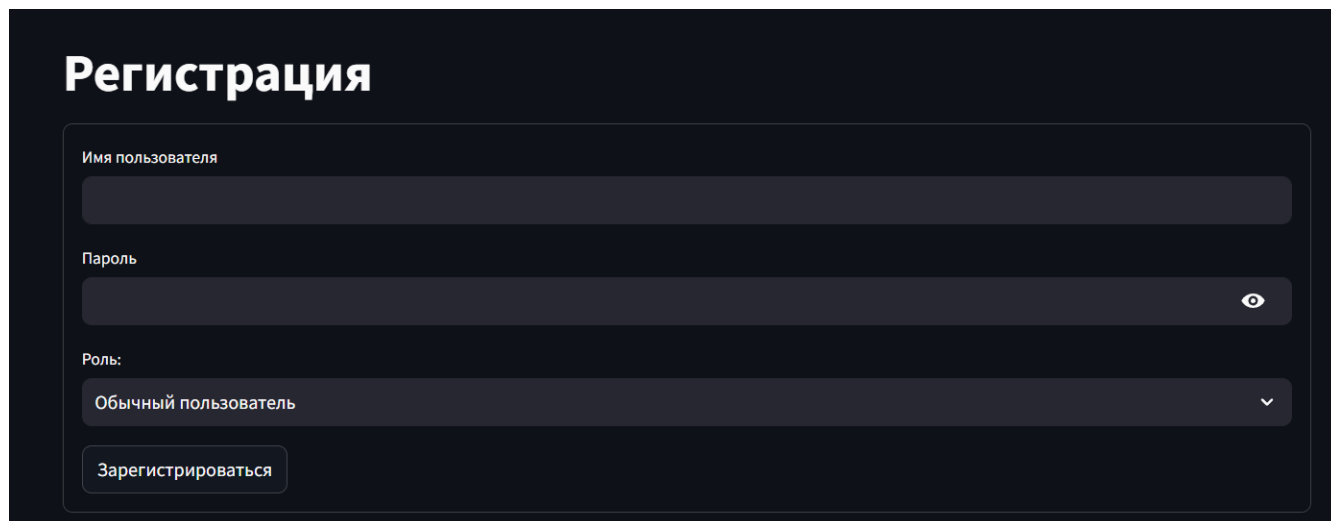
Добро пожаловать, asd!

Выйти

Выберите страницу

- ☒ Фильтрация клубов
- ☐ Дополнительная информация
- ☐ Добавить трофей клубу

1. Регистрация



Регистрация

Имя пользователя

Пароль

Роль:

Обычный пользователь

Зарегистрироваться

2. Добавление трофея клубу

Добавить трофей клубу

Выберите клуб

AC Milan

Выберите трофей

A-League Men

Введите год получения трофея

2023

Добавить трофей

3. Оформление трансфера игрока

Оформить трансфер игрока в другой клуб

Выберите клуб, в котором игрок сейчас играет

AC Milan

Выберите игрока

Elijah Brown

Введите новую зарплату

100000

Выберите клуб, в который игрок перейдёт

AC Milan

Оформить трансфер игрока

Вывод

В рамках проекта "Сервис для поиска футбольного клуба для футболистов" был разработан функционал для взаимодействия игроков с базой данных футбольных клубов. Система позволяет пользователям регистрироваться, авторизоваться и находить подходящие клубы с использованием фильтров, таких как локация, достижения клуба, возрастные ограничения и другие критерии.

Система поддерживает три роли: **пользователь**, **журналист** и **администратор**, что позволяет разграничить доступ к функционалу и обязанностям:

- **Пользователь** может искать клубы, просматривать их данные и оценивать их достижения.
- **Журналист** имеет возможность добавлять информацию о новых трофеях, выигранных клубами.
- **Администратор** обладает полным доступом к системе, включая управление трансферами игроков и данными о клубах, а также назначение ролей другим пользователям.

Для обеспечения безопасности, все пароли пользователей хешируются, исключая их хранение в открытом виде. Система реализована с использованием базы данных PostgreSQL, соблюдающей требования третьей нормальной формы, что обеспечивает целостность и надежность данных. Взаимодействие с базой данных выполняется через библиотеку psycopg2.

Разработанный функционал предоставляет удобный интерфейс для пользователей, позволяя игрокам в интерактивном режиме находить подходящие клубы и просматривать данные о них. Это делает систему доступной и удобной для широкой аудитории.

Проект представляет собой масштабируемую и надежную платформу, способную поддерживать высокую нагрузку и предоставлять удобные инструменты для поиска клубов и управления информацией о них.

Список использованных источников

1. <https://docs.streamlit.io/develop>
2. <https://www.postgresql.org/docs/17/index.html>
3. <https://www.psycopg.org/docs/usage.html>
4. [GitHub - apftf777/sample_course_work](#)