DOCUMENTACION DE PROYECTO BOOTCAMPS

Autores
Leonardo David Díaz León
Juan Sebastián Tirado Verbel

Docente
Boris Espitia

Asesor
Emilio Ayazo

Curso
Programación

Montería, Colombia
2024

Contenido

Introduction 3

1. Technologies Used 5

2. Architecture Diagram 5
Client (Browser) 6
Server (Flask) 6
Database (SQLite) 6
Project Structure 6
Installation and Configuration 8
Database 10
Main Functionalities 15
User Interface 16
Responsive Design 17
Known Issues and Limitations 23
Performance 23
Security 23
Usability 24
Availability 24
Future Enhancements 24
Credits and License 26

Introduction

Purpose
The KICK OFF project is an innovative and efficient web application specifically designed to simplify the management of sports tournaments. This platform serves as a comprehensive tool aimed at administrators and organizers of sports events, providing a practical and robust solution for registering teams, scheduling matches, and keeping a detailed record of results and standings. The primary goal of KICK OFF is to streamline and optimize the tournament organization process by offering an intuitive and user-friendly interface with flexible, customizable access to meet the diverse needs of its users.

Project Name: Tournament Organizer – KICK OFF

Description:
KICK OFF is a web platform focused on the efficient organization and management of sports tournaments, with a special emphasis on soccer. It enables administrators and organizers to effortlessly create and manage tournaments, from registering teams and scheduling matches to collecting and displaying results and standings. The platform is designed to optimize the coordination of sports events and enhance the experience for both participants and spectators.

Objective:
The objective of KICK OFF is to develop a web platform that allows users to organize, manage, and participate in soccer tournaments flexibly and efficiently. The solution aims to provide both free access with basic functionalities and premium plans offering advanced features and greater control over tournament management. The goal is to deliver a personalized and accessible experience for all types of users, from amateur organizers to professional sports management professionals.

Project Architecture
The KICK OFF project is built using a client-server architecture, where users interact through a web interface, and business logic and data management are processed on the

server. This approach ensures a clear separation between the frontend and backend, optimizing the information flow and scalability of the application.

1. Technologies Used

Backend:

- **Python:** Used as the primary programming language for server development, providing a robust and versatile foundation to handle the application's business logic.
- **Flask:** A Python microframework that enables the straightforward and efficient creation of web applications. Flask handles client requests, processes the business logic, and sends appropriate responses.

Database:

- **SQLite:** A lightweight and easy-to-configure database used to store and efficiently manage information about tournaments, teams, players, and results. SQLite integrates seamlessly with Flask, facilitating read and write operations.

Frontend:

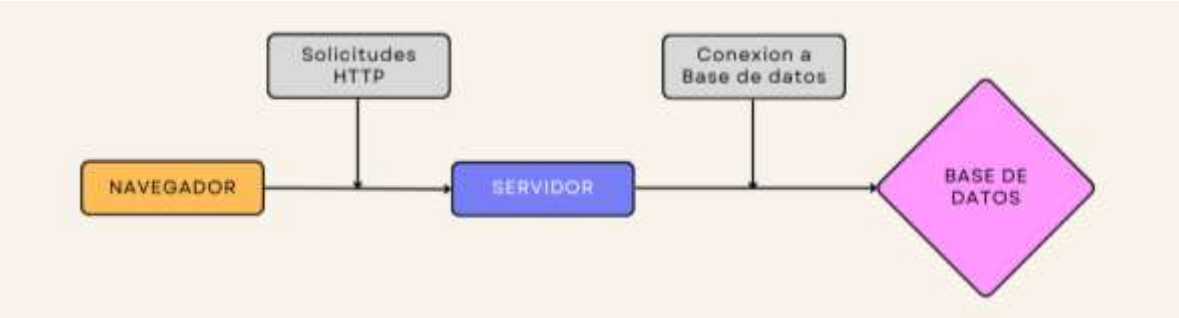
- **HTML, CSS, JavaScript:** These are used to create the application's user interface, providing an interactive and user-friendly experience. HTML structures the web page, CSS styles it, and JavaScript adds dynamic and interactive functionalities.

Styling Framework:

- **Bootstrap:** A front-end design framework that helps quickly develop responsive and aesthetically pleasing web interfaces. Bootstrap is used to facilitate the creation of a modern and adaptable design for various devices.

2. Architecture Diagram

The following diagram illustrates the flow of information between the components of the client-server architecture:



Client (Browser)

The user interacts with the web interface to perform actions such as registering teams, viewing results, or scheduling matches.

Server (Flask)

The server receives client requests, processes the necessary business logic, and handles database read and write operations.

Database (SQLite)

Stores and manages information about tournaments, teams, and results, enabling the server to efficiently access and update data.

This modular and well-defined architecture allows KICK OFF to deliver a robust and scalable platform for sports tournament management, ensuring a smooth user experience and optimized performance.

Project Structure

This section outlines the organization of files and folders within the project.

app

- __pycache__
  - app.cpython-312.pyc
  - config.cpython-312.pyc

- models
  - __pycache__
  - database
  - db.py
- routes
  - __pycache__
  - __init__.py
  - main.py
- static
  - css
    - style.css
  - img
    - escudos
    - avatar.svg
    - img-torneo.svg
  - js
    - login.js
    - logout.js
    - partidos.js
    - registroEquipos.js
    - registroJugadores.js
    - registroTorneo.js
    - script.js
    - singUp.js
- templates
  - dashboard.html
  - dashboardLanding.html
  - equipos.html

            - fixtures.html
            - index.html
            - layout.html
            - login.html
            - registerTorneo.html
            - registroEquipos.html
            - registroJugadores.html
            - singUp.html
            - torneo.html
            - verEquipo.html
        - utils
            - __pycache__
                - funtions.cpython-312.pyc
                - funtions.cpython-313.pyc
            - funtions.py
        - __init__.py
        - app.py
        - config.py
- README.md

Installation and Configuration

To set up the KICK OFF project and ensure its proper functioning, follow these installation and configuration steps. Below is a detailed guide:

Prerequisites
Before starting the installation and setup, ensure that the following are present on your system:

- **Python:** You need to have Python installed on your computer. It is recommended to use Python 3.7 or higher. Check if Python is installed by running the following command in your terminal or command line:
- `python --version`

  If Python is not installed, download and install it from python.org.

- **pip:** This is Python's package manager and should be included with the Python installation. Verify its installation with:
- `pip --version`

  If pip is not installed, follow the official documentation to install it.

Environment Setup

To maintain a clean development environment and manage dependencies effectively, it is recommended to use a virtual environment. Follow these steps:

1. **Create a virtual environment**: Navigate to your project directory and run:
2. `python -m venv venv`
3. **Activate the virtual environment**:
   - On Windows:
   - `venv\Scripts\activate`
   - On macOS and Linux:
   - `source venv/bin/activate`
4. **Install dependencies**: Create a `requirements.txt` file with the following lines (or ensure it contains the necessary dependencies):
5. `Flask==2.0.3`
6. `Flask-SQLAlchemy==2.5.1`
7. **Install the dependencies**:
8. `pip install -r requirements.txt`

   Alternatively, you can manually install Flask and SQLite with:

   ```
   pip install Flask Flask-SQLAlchemy
   ```

Database Configuration

The SQLite database will be used to store tournament information. To set it up:

1. **Create the database file**: In your project, create an SQLite database file named `databaseTorneo.db` (or any other name you prefer).
2. **Connect Flask to SQLite**: In your Flask configuration file (`app.py` or `config.py`), add the following line to set up the database connection:
3. `app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///databaseTorneo.db'`

4. **Initialize the database**: Create the necessary tables by running a database initialization script in Python:

```
from app import db
# Create all tables defined in the model
db.create_all()
```

   Ensure your application has the following initialization code:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///databaseTorneo.db'
db = SQLAlchemy(app)

if __name__ == '__main__':
    with app.app_context():
        db.create_all()  # Creates the tables in the database
```

8. **Verify the connection and structure**: Start your application and check that the database has been created correctly and that the tables are in place. You can use an SQLite management tool like **DB Browser for SQLite** to inspect the database and confirm that the structure has been properly implemented.

Database

Database: Structure and Design
The KICK OFF system uses SQLite as the database to efficiently manage information related to teams, players, matches, and results. SQLite is ideal for this project due to its lightweight nature, ease of integration with Flask, and its ability to handle read and write operations quickly and securely.

Database Schema Design
The database schema is designed with interrelated tables that store information in an organized manner. This structure facilitates data retrieval and updates, ensuring the platform runs efficiently and without redundancy. Below is a description of the main tables and their relationships:

1. **Teams Table**
   - **Fields**: id_team, team_name, city, state, creation_date
   - **Relationships**: Related to the players table through `id_team`, and to the matches table to record participating teams.
2. **Players Table**
   - **Fields**: id_player, first_name, last_name, age, position, id_team
   - **Relationships**: Related to the teams table through `id_team` to assign players to their respective teams.
3. **Matches Table**
   - **Fields**: id_match, id_home_team, id_away_team, match_date, location

    - **Relationships**: Related to the teams table to register participating teams and to the results table to store match results.
4. **Results Table**
    - **Fields**: id_result, id_match, home_team_goals, away_team_goals, comments
    - **Relationships**: Associated with the matches table through `id_match` to link results with their respective match.
5. **Tournaments Table**
    - **Fields**: id_tournament, tournament_name, start_date, end_date, tournament_type
    - **Relationships**: Related to the teams and matches tables to manage participating teams and scheduled matches during the tournament.

Table Relationships
Tables are interconnected via foreign keys to maintain referential integrity and data consistency. For example:

- The players table has a foreign key `id_team` that relates to the teams table, assigning players to a specific team.
- The matches table uses foreign keys `id_home_team` and `id_away_team` to link the teams participating in a match.
- The results table is linked to the matches table via the foreign key `id_match`, ensuring each result corresponds to a specific match.

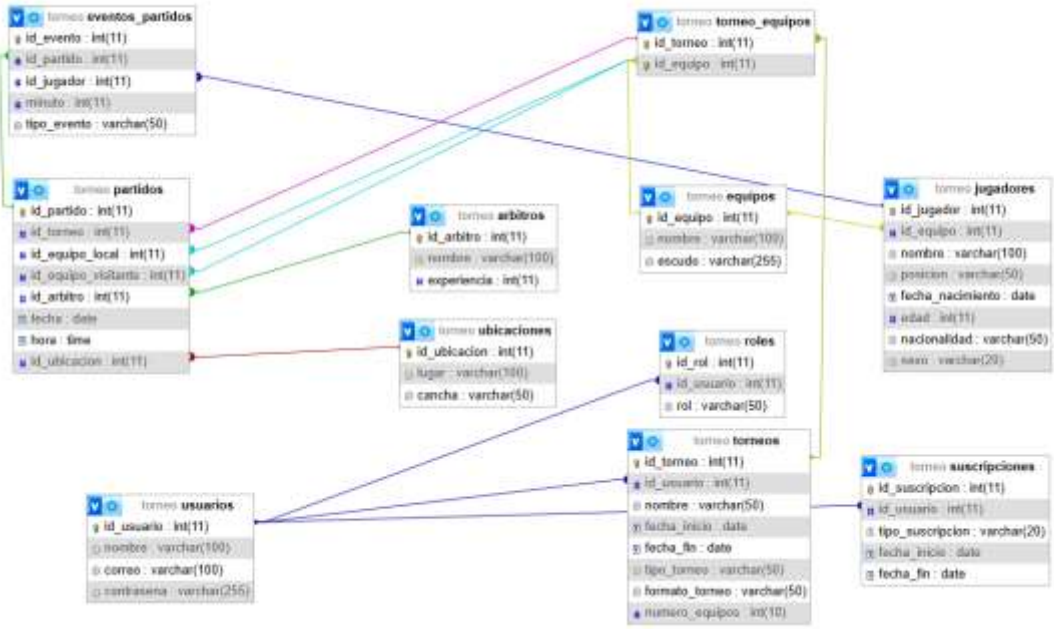
Advantages of the Schema Design

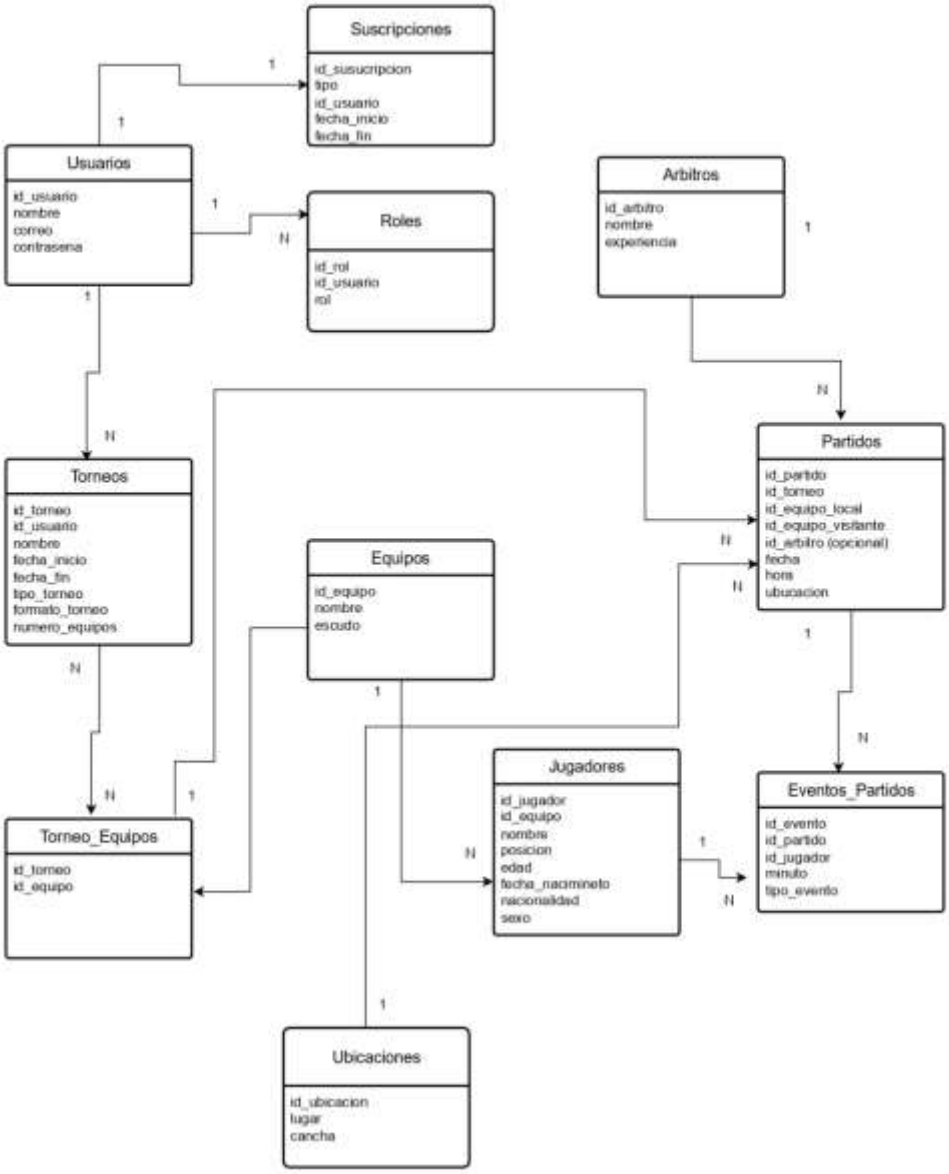
- **Normalized Structure**: The database is designed following normalization principles to prevent data redundancy and ensure data integrity.
- **Efficient Queries**: The interconnection between tables allows for complex queries to be performed efficiently, such as retrieving a list of players by team, results of a specific tournament, or scheduled match details.
- **Update Flexibility**: Data updates can be performed in a straightforward and consistent manner due to the modular structure of tables and their relationships.

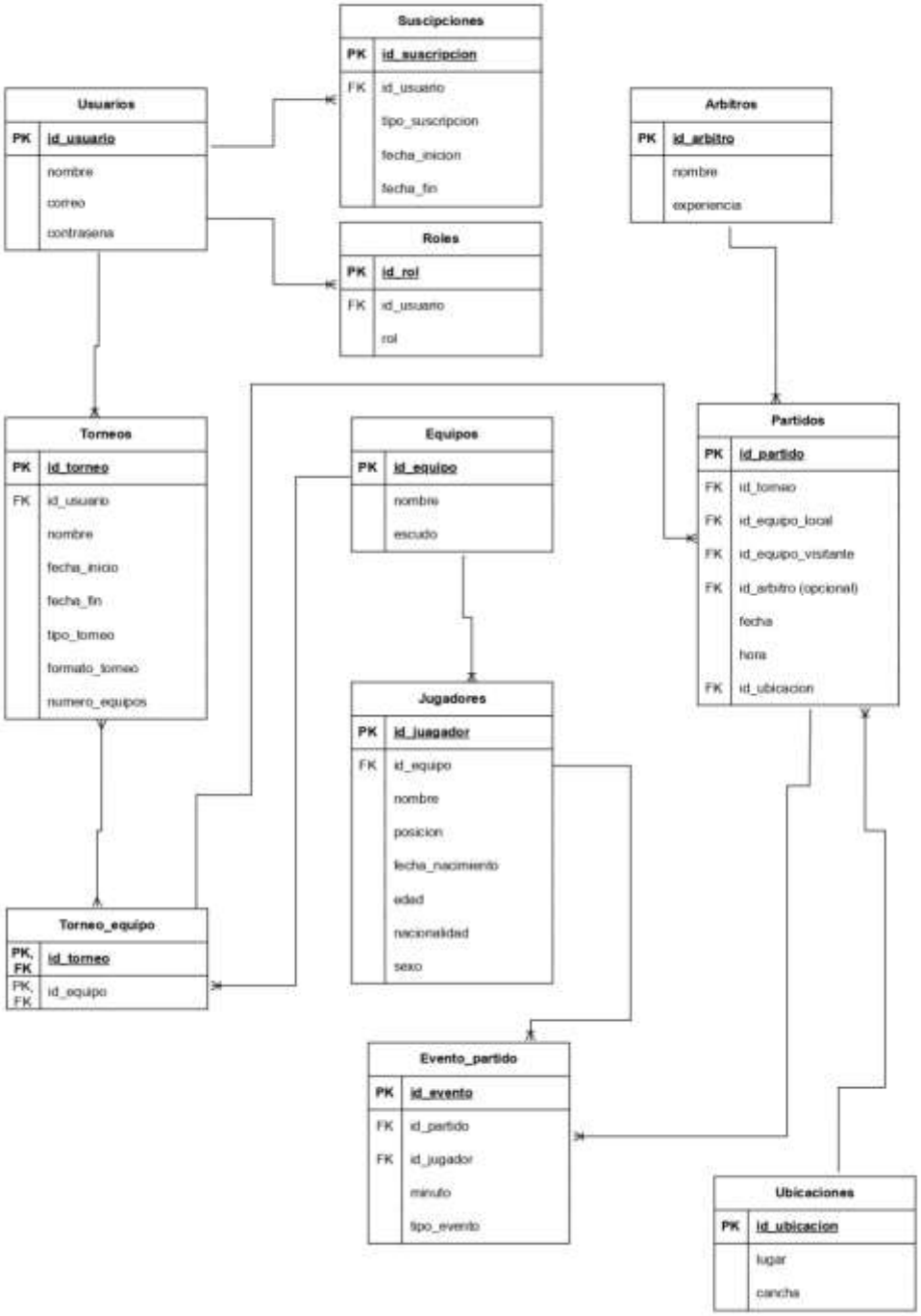
Example SQL Query
Here is an example SQL query that could be run on this schema to retrieve the results of a specific tournament:

```
SELECT t.tournament_name, p.match_date, e1.team_name AS home_team,
e2.team_name AS away_team,
       r.home_team_goals, r.away_team_goals
FROM tournaments t
JOIN matches p ON t.id_tournament = p.id_tournament
JOIN teams e1 ON p.id_home_team = e1.id_team
JOIN teams e2 ON p.id_away_team = e2.id_team
JOIN results r ON p.id_match = r.id_match
WHERE t.id_tournament = ?;
```

This query retrieves the tournament name, match date, participating teams, and corresponding match results.




Suscripciones
id_susucripcion
tipo
id_usuario
fecha_inicio
fecha_fin
Usuarios
id_usuario
nombre
correo
contrasena
Roles
id_rol
id_usuario
rol
Arbitros
id_arbitro
nombre
experiencia
Torneos
id_torneo
id_usuario
nombre
fecha_inicio
fecha_fin
tipo_torneo
formato_torneo
numero_equipos
Partidos
id_partido
id_torneo
id_equipo_local
id_equipo_visitante
id_arbitro (opcional)
fecha
hora
ubicacion
Equipos
id_equipo
nombre
escudo
Jugadores
id_jugador
id_equipo
nombre
posicion
edad
fecha_nacimiento
nacionalidad
sexo
Eventos_Partidos
id_evento
id_partido
id_jugador
minuto
tipo_evento
Torneo_Equipos
id_torneo
id_equipo
Ubicaciones
id_ubicacion
lugar
cancha


Suscripciones
PK id_suscripcion
FK id_usuario
tipo_suscripcion
fecha_inicion
fecha_fin
Usuarios
PK id_usuario
nombre
correo
contrasena
Arbitros
PK id_arbitro
nombre
experiencia
Roles
PK id_rol
FK id_usuario
rol
Torneos
PK id_torneo
FK id_usuario
nombre
fecha_inicio
fecha_fin
tipo_torneo
formato_torneo
numero_equipos
Equipos
PK id_equipo
nombre
escudo
Partidos
PK id_partido
FK id_torneo
FK id_equipo_local
FK id_equipo_visitante
FK id_arbitro (opcional)
fecha
hora
FK id_ubicacion
Jugadores
PK id_jugador
FK id_equipo
nombre
posicion
fecha_nacimiento
edad
nacionalidad
sexo
Torneo_equipo
PK, FK id_torneo
PK, FK id_equipo
Evento_partido
PK id_evento
FK id_partido
FK id_jugador
minuto
tipo_evento
Ubicaciones
PK id_ubicacion
lugar
cancha

Main Functionalities

System Functionalities

The KICK OFF system is designed to provide a comprehensive solution for organizing and managing sports tournaments, allowing administrators and participants to interact efficiently and securely. Below is a detailed breakdown of the main functionalities of the system:

User Management

The user management module provides complete control over account creation and management, as well as assigning permissions and access to various platform functionalities.

- **Registration and Login**: Users can sign up on the platform through an easy registration process and log in securely. This ensures each user has exclusive access to their data and settings.
- **Role Assignment (Super Administrator, Tournament Administrator, Spectator)**: Different roles with specific access levels and permissions can be assigned. Super Administrators have full control over the platform, while Tournament Administrators manage events, and Spectators only access tournament information.
- **Plan Management (Free and Premium)**: Users can choose between free and premium access plans, each offering different features and benefits. Premium plans may include advanced functionalities like detailed statistics, tournament customization, and more.

Tournament Creation

This module allows organizers to create and manage tournaments with flexibility and control, adapting to different needs and types of sports events.

- **Tournament Customization**: Organizers can configure the specific details of their tournaments, such as name, date, location, and other essential characteristics defining the event.
- **Game Mode Selection**: The appropriate game mode for each tournament can be chosen, such as knockout matches, leagues, group tournaments, etc.
- **Tournament Format Configuration**: The platform allows the setting of competition formats, specifying the number of rounds, match structures, and other key aspects.
- **Team and Player Management**: Organizers can add and manage teams and players, assigning members to teams and updating participant information as tournament phases progress.

Statistics System

The statistics system allows administrators and participants to analyze and visualize tournament performance and results.

- **Statistics Generation**: The platform generates detailed statistics, including team performance, scores, match analysis, and more. This helps organizers evaluate and monitor tournament progress.
- **Restricted Visualization**: The viewing of specific statistics and data is limited based on user access permissions, ensuring the confidentiality and integrity of information.
- **Role-Based Access Control**: Access to analysis functions and statistical visualization is controlled according to user roles, ensuring that only users with the appropriate permissions can access sensitive information.

User Interface

Web Interface of the KICK OFF System

The KICK OFF system has been designed with an intuitive and user-friendly web interface, facilitating user interaction with the platform. The interface is organized so that administrators and tournament organizers can quickly access main functionalities and efficiently manage sports events. Below are the key sections of the web interface:

Main Pages of the Interface

1. **Tournament Management Section**
    - **Functionalities**: Allows users to view active and past tournaments, as well as create, edit, and delete tournaments.
    - **Features**: Each tournament can include details such as name, start and end dates, game mode, and tournament status.
    - **Usability**: Easy-to-use buttons and forms for adding new tournaments and updating existing information.
2. **Team and Player Registration Section**
    - **Functionalities**: Facilitates the registration of new teams and players, assigning players to the corresponding teams and managing their data.
    - **Features**: Data entry forms with fields for team name, player name, position, age, and other relevant details.
    - **Usability**: Clean interfaces and forms that allow the addition of multiple players and teams in a single operation.
3. **Statistics and Results Visualization Section**
    - **Functionalities**: Displays match statistics, rankings, and tournament results in a clear and understandable format.
    - **Features**: Graphs, tables, and match summaries, with filtering options to view statistics for specific teams, players, and tournaments.
    - **Usability**: An interactive section with options to sort and filter data by different criteria such as date, team, or match type.

Responsive Design

The system is designed with a responsive approach, ensuring that the platform is fully functional and visually appealing across different devices and screen sizes. This allows users to access the application from desktop computers, tablets, and mobile phones without display or usability issues.

Responsive Design Features:

- **Adaptability**: Interface elements, such as menus, buttons, and tables, automatically adjust to the device's screen size.
- **Compatibility**: Technologies like HTML5, CSS3, and Bootstrap are used to ensure content adapts and maintains an appropriate presentation on all devices.
- **User Experience (UX)**: Interactive elements are efficiently redesigned to enhance navigation and interaction on both small and large screens.

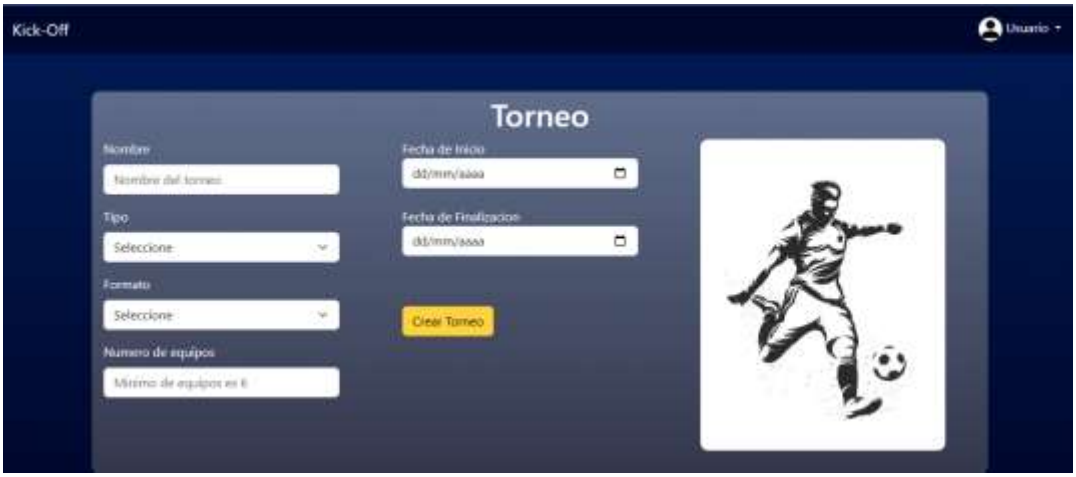
Inicio

Login

Registro

dashboard

register-torneo



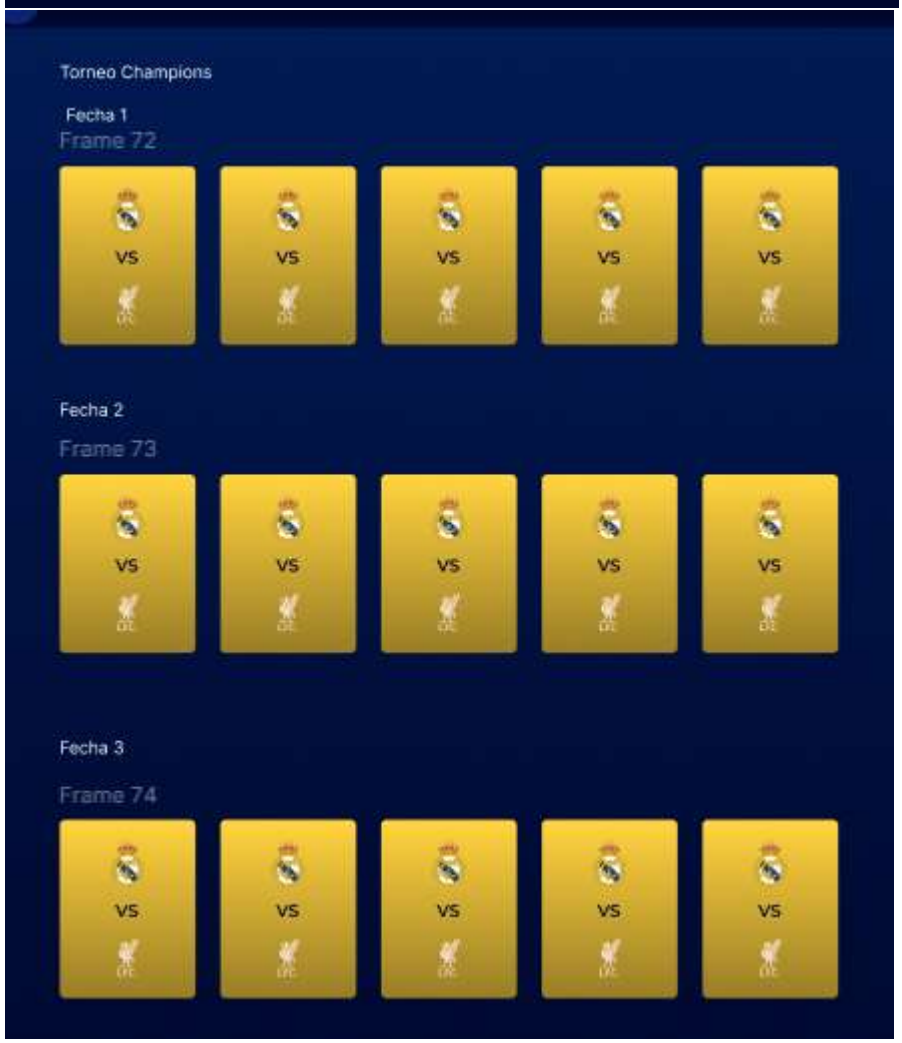

Kick-Off
Usuario
Equipos
Equipo Escudo Acciones
1 Obcaecati quia nostr
2 Minus aute sit poss
3 Nemo qui perferendis
4 Velit aut adipisicin
5 Alias est maiores al
6 Labore accusamus deb
Registrar Jugadores
Ver Equipo
Eliminar
← Back
Generar Cruces


Kick-Off
Usuario
Torneo
Equipos
Ver
Partidos
Ver
Estadisticas
Ver
← Back


LOGO
INICIO
CONFIGURACION
CHAMPIONS
EQUIPO
ESCUDO
1
2
3
4
5
6
CREAR EQUIPOS


LOGO
INICIO
CONFIGURACION
PARTIDOS
PARTIDO FECHA HORA UBICACION ARBITRO
1 Real Madrid vs Liverpool 2024-11-19 07:00 pm Cacha Bernabeu Jorge Rodriguez
2 River vs Barcelona 2024-11-19 07:00 pm Cacha Bernabeu Jorge Rodriguez
3 Everton vs Chelsea 2024-11-19 07:00 pm Cacha Bernabeu Jorge Rodriguez
4 Liverpool vs River 2024-11-21 07:00 pm Cacha Bernabeu Jorge Rodriguez
5 Real Madrid vs Chelsea 2024-11-21 07:00 pm Cacha Bernabeu Jorge Rodriguez
6 Everton vs Barcelona 2024-11-21 07:00 pm Cacha Bernabeu Jorge Rodriguez
7 Barcelona vs Real Madrid 2024-11-24 07:00 pm Cacha Bernabeu Jorge Rodriguez
8 Chelsea vs Liverpool 2024-11-24 07:00 pm Cacha Bernabeu Jorge Rodriguez
9 River vs Everton 2024-11-24 07:00 pm Cacha Bernabeu Jorge Rodriguez
10 River vs Real Madrid 2024-11-27 07:00 pm Cacha Bernabeu Jorge Rodriguez
11 Everton vs Liverpool 2024-11-27 07:00 pm Cacha Bernabeu Jorge Rodriguez
12 Barcelona vs Chelsea 2024-11-27 07:00 pm Cacha Bernabeu Jorge Rodriguez
13 Real Madrid vs Everton 2024-11-30 07:00 pm Cacha Bernabeu Jorge Rodriguez
14 Liverpool vs Barcelona 2024-11-30 07:00 pm Cacha Bernabeu Jorge Rodriguez
15 River vs Chelsea 2024-11-30 07:00 pm Cacha Bernabeu Jorge Rodriguez


LOGO
INICIO
CONFIGURACION
Real Madrid
Jugador
IDENTIFICACION
NOMBRE
FECHA NACIMIENTO
EDAD
NACIONALIDAD
SEXO
HOMBRE
MUJER
IDENTIFICACION NOMBRE FECHA NACIMIENTO EDAD SEXO
1 1069854852 Segio Ramos 2000-02-05 24 M
2 1254699532 Juan Perez 2001-04-04 23 M
3 1263654789 Luca Modrid 2001-02-05 23 M
4 1187458963 Pedro Ramos 2001-01-20 23 M
5 1023021254 Luis Diaz 2000-02-30 24 M
6 1525245874 Omar Perez 2000-10-05 24 M
7 1023323365 Leonardo Diaz 2000-08-05 24 M
8 1789632364 Mario Ruiz 2004-09-20 20 M
9 1254896285 Segio Santos 2004-12-10 20 M
10 1595142365 Felipe Lopez 2004-01-05 20 M
11 1234563254 Javier Arias 2000-07-05 24 M
12 1597535741 Juan Char 2000-05-15 24 M
13 1201201478 Arlan Ramos 2000-03-10 24 M


Torneo Champions
Fecha 1
Frame 72
VS
Fecha 2
Frame 73
VS
Fecha 3
Frame 74
VS


LOGO
HOME
FIXTURES
RESULTS
TABLE
PLAYERS
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool
5/09/24 07:00 pm Real Madrid 2 VS 1 Liverpool


LOGO
HOME
FIXTURES
RESULTS
TABLE
PLAYERS
1 - Liverpool 11 8 1 1 21 6 15 25
2 - Manchester City 11 7 2 2 22 13 9 23
3 - Chelsea 11 5 4 2 21 15 6 19
4 - Arsenal 11 5 4 2 18 12 6 19
5 - Nottingham Forest 11 5 4 2 15 10 5 19
6 - Brighton And Hove Albion 11 5 4 2 19 15 4 19

Known Issues and Limitations

The project has several key areas that need improvement and optimization to ensure its functionality, scalability, and security. The following outlines these aspects in more detail:

Identified Limitations

1. **Multilingual Support**
   Currently, the platform does not support multiple languages, limiting accessibility for users who do not speak the default language. Adding a localization system that translates both the interface and dynamic content would expand the platform's global reach.
2. **Mobile Device Interface**
   Although functional, the current interface needs optimization for smaller screens. This includes an improved responsive design and extensive testing across different mobile device resolutions to ensure an optimal user experience.
3. **Manual Team Configuration**
   Manually adjusting the number of teams for each tournament is inefficient. Implementing a dynamic system that allows administrators to define teams with greater flexibility would significantly reduce the time needed to set up a tournament.

Performance

Strong performance is critical for a successful application. Key goals include:

- **Page Load Time (< 3 seconds)**: Implement script optimization, caching, file compression, and well-structured databases to maintain page load times under 3 seconds.
- **Support for Concurrent Users (500 users)**: The infrastructure must handle significant concurrent traffic. This can be achieved through load balancers, scalable servers, and backend optimization for managing multiple requests.
- **Scalability**: The system architecture must be designed with future growth in mind, allowing for the addition of new modules and services without compromising current performance.

Security

Protecting user data and maintaining user trust are essential:

- **Data Encryption**: Sensitive data, such as passwords and personal information, should be encrypted using modern standards like AES-256.
- **Two-Factor Authentication (2FA)**: Adding this method provides an additional layer of security, ensuring that only legitimate users can access their accounts.
- **Protection Against Common Threats (OWASP Top 10)**: Implement input validation, SQL injection prevention, and data sanitization to guard against common attacks.
- **Regulatory Compliance**: The application should comply with regulations like GDPR (in Europe) or local data protection laws depending on the region of operation.

Usability

User experience is crucial for adoption and retention:

- **Intuitive Interface**: Design a user-friendly experience with simple navigation and intuitive visual elements.
- **Compatibility with Modern Browsers**: The application should be tested and functional on browsers such as Chrome, Firefox, Edge, and Safari.
- **Accessibility (WCAG 2.1 Level AA)**: Include accessibility features such as screen reader support, high contrast, and keyboard navigation for users with disabilities.

Availability

To ensure a reliable and consistent experience, availability objectives include:

- **Uptime (99.5%)**: The system should be available most of the time, using redundant infrastructure to minimize downtimes.
- **Failure Recovery (< 30 minutes)**: Implement procedures and tools for quick recovery in the event of a failure, including automatic backups and real-time monitoring.

Future Enhancements

The purpose of these future enhancements is to enrich user experience, optimize system functionality, and expand the platform's capabilities to better meet the needs of users and sports tournament organizers. These updates aim to make the platform more accessible, functional, and adaptable, as well as to facilitate its use in different contexts and regions.

Planned Features for Future Versions

1. **Multilingual Support**:

    - Implement a localization and translation system to enable the platform to be used in various languages, broadening its accessibility and offering a better user experience to audiences from different regions.
2. **Integration with Payment Platforms for Registrations**:
    - Integrate secure and efficient payment methods for participant registration in tournaments. This would include online payment options via credit/debit cards and digital payment services like PayPal and other mobile payment systems.
3. **Complementary Mobile Application**:
    - Develop a mobile app to complement the web platform, enabling users to manage and follow their tournaments from mobile devices. The app will feature a user-friendly interface for organizers and participants to stay updated on events in real-time.

Additional Potential Improvements

4. **Support for Tournament Customization by Categories or Modes**:
    - Allow organizers to customize tournaments based on different categories or game modes, adapting to various sports and competition formats for greater event flexibility.
5. **Email Notification System for Key Events**:
    - Integrate an automated email notification system that alerts participants and organizers of key events, such as schedule changes, result updates, registration reminders, and other important notifications.
6. **Integration with Map APIs for Venue Location**:
    - Add a location system that allows users to see and get directions to match venues, improving participant and spectator accessibility and event travel planning.
7. **API Development for Future Integrations**:
    - Create a robust API to facilitate integration with other systems and applications, such as social media platforms for real-time result sharing or tools for player and team performance analysis.

Summary of Additional Functionalities

- **Payment System Integration**: Ensure the platform supports various payment methods for convenient and secure tournament registrations.
- **Multilingual Support**: Expand platform accessibility by adding multiple language options.
- **API Development for Future Integrations**: Build a flexible infrastructure that allows interaction with other applications, enhancing functionality and interoperability.

These improvements are designed to make the platform more comprehensive and adaptable, providing an enriched experience for sports tournament organizers and participants, optimizing both event management and participation.

Credits and License

Credits: Project developed by Leonardo David Díaz León and Juan Sebastián Tirado Verbel.

The project was developed by Leonardo David Díaz León and Juan Sebastián Tirado.
The use and distribution of the project are subject to the license specified in the repository's license file.
GitHub: https://github.com/JuanT20/Torneo
Figma: https://www.figma.com/design/yACg2KKf6fkEyBEP3P0lLw/TORNEO?node-id=0-1&t=ySZestCipyNzOaYl-1