Tunisian Fantasy API

Name: Becher Zribi

Major: BA

Module: IT 325

Supervised by: Montassar Ben Messaoud

Abstract

The Tunisian Fantasy API is an innovative web-based platform with the ambition of bringing fantasy sports to Tunisian football fans. It provides users with a secure, scalable, and intuitive experience for managing their fantasy teams, tracking player performance, and engaging in friendly competition, leveraging state-of-the-art technologies such as **Flask**, **SQLite**, and JWT (JSON Web Tokens). This project addresses a significant gap in Tunisia's sports technology ecosystem by delivering a localized fantasy sports solution tailored specifically to the Tunisian Ligue 1.

Developed using a modular architecture, the platform ensures flexibility and long-term maintainability, paving the way for future enhancements such as real-time match data integration, mobile application development, and advanced analytics. The Tunisian Fantasy API not only enhances fan engagement but also celebrates Tunisia's dynamic football culture, offering a unique and immersive experience for users to connect with their favorite teams and players.

In this project, we use API to interact with the backend, JWT for secure authentication, and ORM for database management.

Glossary

API Application Programming Interface 2

JWT JSON Web Token 2, 17

ORM Object-Relational Mapping 2

Contents

List of Figures

List of Tables

1 General Introduction

1.1 Motivation

Fantasy sports have been a global phenomenon, with platforms like the Premier League Fantasy attracting millions of users across the world. In Tunisia, football is more than just a sport, it's a passionate event that unites people all over the country. Many Tunisian fans actively participate in international fantasy leagues, of which the Premier League Fantasy is among the most popular and highly competitive ones.

Yet, despite this enthusiasm for fantasy sports, there is no dedicated platform for the Tunisian Ligue 1. This gap presents a unique opportunity to build a country-specific fantasy sports application tailored to Tunisian fans. By developing the **Tunisian Fantasy API**, we aim to provide a platform that not only enhances fan engagement but also celebrates the rich football culture of Tunisia. This project seeks to bring the excitement of fantasy sports to the Tunisian Ligue 1, enabling fans to connect with their favorite teams and players in a more engaging and meaningful way.

1.2 Research Goals

The core objectives of the Tunisian Fantasy API include:

- Provide a secure and user-friendly platform for managing fantasy teams.
- Allow users to track player performances and match results.
- Integrate social media sharing to facilitate additional user engagement.
- Create a leaderboard system to foster competition among users.

1.3 Target Audience

The platform is designed for Tunisian football fans who want to engage more deeply with the league by managing their own fantasy teams and competing with others.

1.4 Conclusion

The Tunisian Fantasy API aims to fill the gap in the Tunisian sports technology landscape by providing a localized fantasy sports platform. Using modern web technologies, the project seeks to increase fan engagement and celebrate the rich football culture of Tunisia.

2 System Design and Architecture

2.1 High-Level Architecture

The Tunisian Fantasy API is built using a **microservices architecture**, ensuring modularity, scalability, and maintainability. The system is divided into the following key components:

- **Backend**: A Flask-based API server that handles all core functionalities, including user authentication, fantasy team management, and player performance tracking.
- **Database**: SQLite is used for lightweight data storage, with SQLAlchemy as the Object-Relational Mapping (ORM) tool to streamline database operations.
- **Authentication**: A JWT (JSON Web Token)-based system is implemented to ensure secure user access and session management.

2.2 Database Schema

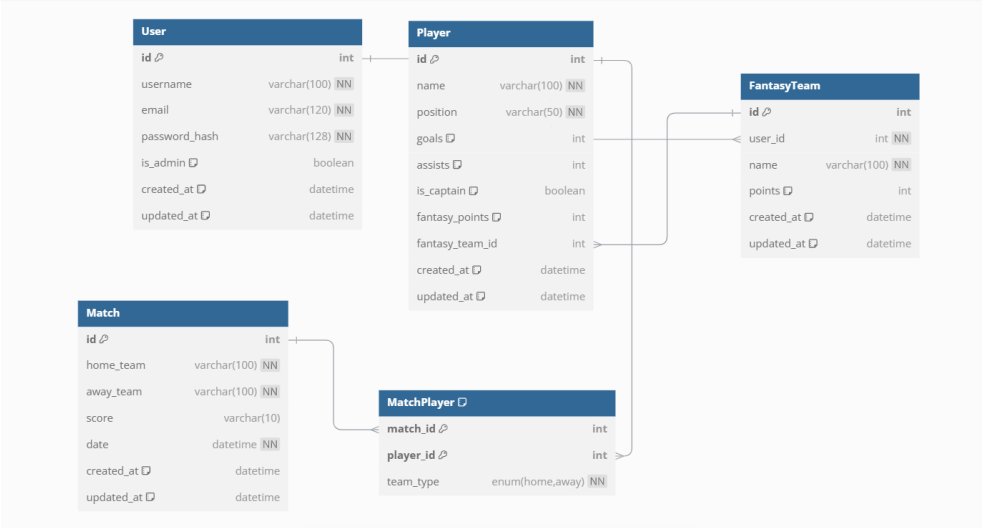

Figure 1: Database Schema Diagram

The database schema, illustrated in Figure 1, is designed to support the core features of the API. It includes tables for **users**, **fantasy teams**, **players**, and **matches**. This schema ensures efficient data management and retrieval, enabling seamless functionality for users.

2.3 Tech Stack

The Tunisian Fantasy API leverages the following technologies:

- **Backend**: Python Flask for the core API, Flask-Smorest for RESTful API development, and Flask-Limiter for rate limiting to ensure optimal performance.

- **Database**: SQLite for lightweight and efficient data storage, with SQLAlchemy as the ORM to simplify database interactions.
- **Authentication**: JWT (JSON Web Tokens) for secure and scalable user authentication.
- **API Documentation**: Swagger UI is integrated to provide an interactive and user-friendly interface for API testing and documentation.

2.4 Conclusion

The architecture of the Tunisian Fantasy API is designed to provide a **scalable**, **secure**, and **user-friendly** platform for fantasy sports enthusiasts. By leveraging modern technologies like Flask, SQLite, and JWT, the system ensures efficient data management, seamless user authentication, and robust performance. This modular design not only meets current requirements but also lays a strong foundation for future enhancements, such as real-time match updates and mobile app integration.

3 Features and Endpoints

3.1 Authentication and Authorization

The API provides the following authentication and authorization endpoints:

Endpoint	Description
/auth/register	Registers a new user
/auth/login	Logs in a user
/auth/refresh	Refreshes the access token
/auth/logout	Logs out a user

Table 1: API Endpoints for Authentication

3.2 Admin Management

The API provides the following endpoints for admin management:

Endpoint	Description
/admin/register	Registers a new admin user
/admin/users	Retrieves a list of all users
/admin/users/<user_id>	Deletes a specific user
/admin/users/<user_id>/make_admin	Promotes a user to admin status

Table 2: API Endpoints for Admin Management

3.3 Fantasy Team Management

The API provides the following endpoints for managing fantasy teams:

Endpoint	Description
/fantasy/create	Creates a new fantasy team
/fantasy/update/<team_id>	Updates a fantasy team
/fantasy/delete/<team_id>	Deletes a fantasy team

Table 3: API Endpoints for Fantasy Team Management

3.4 Player Management

The API provides the following endpoints for managing players:

Endpoint	Description
/player/add	Adds a player to a fantasy team
/player/<team_id>/players	Retrieves all players in a team
/player/update/<player_id>	Updates a player's details
/player/delete/<player_id>	Deletes a player

Table 4: API Endpoints for Player Management

3.5 Leaderboard

The API provides the following endpoint for retrieving the leaderboard:

Endpoint	Description
/leaderboard/get	Retrieves the leaderboard with team rankings

Table 5: API Endpoints for Leaderboard

3.6 Match Details

The API provides the following endpoint for retrieving match details:

Endpoint	Description
/match/<match_id>	Retrieves details of a specific match

Table 6: API Endpoints for Match Details

3.7 Twitter Integration

The API provides the following endpoint for Twitter integration:

Endpoint	Description
/twitter/share	Shares a message on Twitter

Table 7: API Endpoints for Twitter Integration

3.8 Conclusion

The Tunisian Fantasy API provides a comprehensive set of features and endpoints to support fantasy team management, player tracking, and social media integration. These features are designed to enhance user engagement and provide a seamless experience.

4 Implementation Details

4.1 Core Modules

The Tunisian Fantasy API is implemented using the following core modules, each responsible for specific functionality:

- **auth_routes.py**: Handles user authentication, token generation, and session management.
- **admin_routes.py**: Manages admin-specific operations, such as user management and role assignments.
- **fantasy_routes.py**: Manages operations related to fantasy teams, such as creation, updates, and deletions.
- **player_routes.py**: Handles player-related operations, including adding, updating, and deleting players.
- **leaderboard_routes.py**: Generates and manages the leaderboard, ranking teams based on performance.
- **match_routes.py**: Retrieves and processes match details, including scores and player statistics.
- **twitter_routes.py**: Integrates with the Twitter API to enable social media sharing of team achievements.

4.2 Code Snippets

Below are some critical code snippets that demonstrate key functionalities of the API:

```python
from datetime import datetime, timedelta
import jwt
from flask import current_app

def create_access_token(identity, expires_in=15):
    expiration = timedelta(minutes=expires_in)
    payload = {
        "sub": identity,  # Subject (user identifier)
        "exp": datetime.utcnow() + expiration,  # Expiration time
        "type": "access",  # Token type
    }
    # Encode the payload using the app's secret key
    return jwt.encode(payload, current_app.config["SECRET_KEY"],
   algorithm="HS256")
```

Listing 1: JWT Token Creation

```python
@bp.route('/create', methods=['POST'])
def create_fantasy_team():
    """
    Creates a new fantasy team for a user.

    Returns:
        JSON: A success message or an error if the user or team name is invalid.
    """
    data = request.get_json()
    user_id = data.get('user_id')
    team_name = data.get('team_name')

    # Validate user existence
    user = User.query.get(user_id)
    if not user:
        return jsonify({"message": "User not found"}), 404

    # Validate team name
    if not team_name:
        return jsonify({"message": "Team name is required"}), 400

    # Create and save the new fantasy team
    new_team = FantasyTeam(user_id=user_id, name=team_name)
    db.session.add(new_team)
    db.session.commit()

    return jsonify({"message": "Fantasy team created successfully"}), 201
```

Listing 2: Fantasy Team Creation

4.3 Conclusion

The implementation of the Tunisian Fantasy API demonstrates the effective use of modern web technologies to create a secure, scalable, and user-friendly platform. The modular design ensures flexibility and maintainability, allowing for easy updates and future enhancements. By leveraging Python Flask, SQLite, and JWT, the API provides a robust foundation for managing fantasy teams, tracking player performance, and fostering user engagement.

5 Declaration of Originality

I, **Becher Zribi**, declare that the work presented in this project, titled **Tunisian Fantasy API**, is entirely my own creation. This project represents my original ideas, efforts, and contributions, and it has not been submitted, either in whole or in part, for any other academic degree, qualification, or purpose.

Throughout the development of this project, I have made every effort to acknowledge and properly cite all sources of information, inspiration, and reference materials used.

This project is a reflection of my dedication, hard work, and passion for creating innovative solutions that bridge technology and sports. It has been an incredible journey, and I am proud to present this work as a testament to my skills, creativity, and commitment to excellence.

6 Threats to Validity

6.1 Challenges in Data Benchmarking and Performance

One of the significant challenges faced during the development of the Tunisian Fantasy API was the lack of access to free external APIs for fetching real-time data. Most available APIs for sports data, such as player statistics and match results, were either prohibitively expensive or required complex licensing agreements. This limitation forced us to create a seed data system to simulate real-world scenarios.

6.2 Handling Data Challenges

To address the absence of real-time data, we developed a seed data mechanism that populates the database with mock player and match data. This approach allowed us to test and validate the API's functionality effectively. However, it also introduced limitations in terms of data accuracy and real-time updates, which are critical for a fantasy sports platform.

6.3 Performance Considerations

Another challenge was ensuring the API's performance under high load, especially during peak usage times such as match days. To mitigate this, we implemented rate limiting and optimized database queries to reduce latency. Additionally, we used SQLite for its simplicity during development, but a more scalable database solution like PostgreSQL or MySQL would be necessary for a production environment.

6.4 Conclusion

While the Tunisian Fantasy API successfully addresses many technical challenges, the reliance on seed data and the absence of real-time updates remain limitations. Future work will focus on integrating reliable data sources and improving performance to provide a more robust and realistic user experience.

7 Conclusion

The Tunisian Fantasy API represents a significant step forward in bringing the excitement of fantasy sports to Tunisian football fans. By leveraging modern web technologies such as **Flask**, **SQLite**, and JWT (JSON Web Tokens), the platform delivers a secure, scalable, and user-friendly experience tailored specifically to the Tunisian Ligue 1. This project not only bridges a critical gap in Tunisia's sports technology landscape but also celebrates the nation's vibrant football culture, offering fans a unique and immersive way to engage with their favorite teams and players.

The modular and flexible architecture of the API ensures long-term maintainability and paves the way for future enhancements, such as real-time match data integration, mobile app development, and advanced analytics. By combining cutting-edge technology with a deep understanding of Tunisian football fandom, the Tunisian Fantasy API sets a new standard for fan engagement, fostering a stronger connection between fans and the sport they love.

8 Perspectives

The Tunisian Fantasy API is just the beginning of a journey to revolutionize how Tunisian football fans engage with their favorite sport. While the current version of the API provides a solid foundation for fantasy team management, player tracking, and social interaction, there are several exciting opportunities for future enhancements. These planned improvements aim to make the platform more dynamic, accessible, and insightful, ensuring it remains at the forefront of sports technology in Tunisia.

8.1 Integration with Live Match Data

One of the most anticipated features is the integration of live match data. Currently, the API relies on mock data to simulate player performances and match outcomes. However, by integrating real-time data feeds from Tunisian Ligue 1 matches, the platform can provide users with up-to-the-minute updates on player statistics, goals, assists, and other key metrics. This will not only enhance the realism of the fantasy experience but also allow users to make informed decisions about their teams during live matches. Imagine being able to adjust your fantasy lineup in real time as the action unfolds on the pitch!

8.2 Mobile App Development

While the API is designed to be accessible via web browsers, the future development of a dedicated mobile app will take user accessibility to the next level. A mobile app will allow fans to manage their fantasy teams, track player performances, and compete with friends on the go. With push notifications for match updates, player injuries, and league news, the app will keep users connected to the Tunisian Ligue 1 no matter where they are. The goal is to create a seamless and immersive experience that fits into the daily lives of football enthusiasts.

8.3 Advanced Analytics for Player Performance Predictions

To add a layer of strategy and depth to the fantasy experience, the platform will incorporate advanced analytics and machine learning models. These tools will analyze historical player data, current form, and other factors to generate performance predictions. Users will be able to access insights such as a player's likelihood of scoring in the next match, their expected points, and even recommendations for optimal team compositions. This feature will not only make the game more engaging but also empower users to make data-driven decisions, adding a new dimension to the fantasy sports experience.

8.4 Enhanced Social Features

Football is a social sport, and the Tunisian Fantasy API will continue to evolve to reflect this. Future updates will include enhanced social features, such as the ability to create private leagues with friends, share team achievements on social media, and participate in community challenges. These features will foster a sense of camaraderie and competition among users, making the platform not just a tool for managing fantasy teams but also a hub for connecting with fellow fans.

8.5 Expansion to Other Sports

While the current focus is on Tunisian Ligue 1, the platform has the potential to expand to other sports and leagues in the future. By adapting the API to support basketball, handball, or even international football leagues, the platform can cater to a broader audience and become a one-stop destination for fantasy sports enthusiasts in Tunisia and beyond.

8.6 Conclusion

The future of the Tunisian Fantasy API is bright, with a roadmap filled with innovative features and enhancements. By integrating live match data, developing a mobile app, introducing advanced analytics, and expanding social features, the platform will continue to evolve and meet the needs of its users. These improvements are not just about adding new functionalities; they are about creating a richer, more immersive experience that celebrates the passion and excitement of Tunisian football. With these enhancements, the Tunisian Fantasy API will remain a cutting-edge platform, bringing fans closer to the game they love and ensuring that the fantasy sports experience is as dynamic and engaging as the sport itself.

9 References

- **Flask Documentation**: https://flask.palletsprojects.com/
  Official documentation for Flask, the Python web framework used to build the backend of the Tunisian Fantasy API.
- **SQLAlchemy Documentation**: https://www.sqlalchemy.org/
  Official documentation for SQLAlchemy, the Object-Relational Mapping (ORM) tool used for database interactions in the project.
- **Swagger UI Documentation**: https://swagger.io/tools/swagger-ui/
  Documentation for Swagger UI, the tool used to create interactive API documentation for the Tunisian Fantasy API.
- **Premier League Fantasy App**: https://fantasy.premierleague.com/
  The official Premier League Fantasy App, which served as a source of inspiration for the design and functionality of the Tunisian Fantasy API.
- **JSON Web Tokens (JWT) Introduction**: https://jwt.io/introduction/
  A comprehensive guide to JSON Web Tokens (JWT), the authentication mechanism used in the Tunisian Fantasy API.
- **JWT.io - Token Debugger and Secret Key Generation**: https://jwt.io/
  A tool used for debugging JWT tokens and generating secure secret keys for the Tunisian Fantasy API.
- **Twitter API Tools**: https://developer.twitter.com/apitools/
  Official tools and documentation for the Twitter API, used to integrate social media sharing functionality into the Tunisian Fantasy API.
- **Fantasy Sports Industry Trends**: https://www.statista.com/topics/1354/fantasy-sports
  A statistical resource on fantasy sports trends, which helped shape the vision and goals of the Tunisian Fantasy API.
- **Tunisian Ligue 1 Professionnelle - Calendar and Results**: https://www.ftf.org.tn/fr/calendrier-et-resultats-ligue-1/
  Official calendar and results page for the Tunisian Ligue 1 Professionnelle, which served as the primary inspiration and reference for the Tunisian Fantasy API.