Tunisian Fantasy API

A Web-Based Platform for Tunisian Football Fans

Becher Zribi

Module: IT 325
Supervised by: Montassar Ben Messaoud

TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Introduction

- Fantasy sports are a global phenomenon.
- Tunisia lacks a platform for its football league, the Tunisian Ligue 1.
- The **Tunisian Fantasy API** fills this gap with a localized fantasy sports platform.

Motivation

- Football unites people across Tunisia.
- Fans participate in international fantasy leagues (e.g., Premier League Fantasy).
- No platform exists for the Tunisian Ligue 1.
- Opportunity to create a localized fantasy sports application.

Tech Stack

- **Backend**: Python Flask, Flask-Smorest, Flask-Limiter.
- **Database**: SQLite with SQLAlchemy ORM.
- **Authentication**: JWT for secure user access.
- **API Documentation**: Swagger UI.

Features

- **Authentication**: Register, login, refresh, logout.
- **Admin Features**: Manage users, promote admins, delete users.
- **Fantasy Team Management**: Create, update, delete teams.
- **Player Management**: Add, update, delete players.
- **Match Management**: Retrieve match details.
- **Leaderboard**: Track team rankings.
- **Twitter Integration**: Share achievements on Twitter.

Code Snippet: JWT Token Creation

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

def create_access_token(user_id):
    expiration = timedelta(minutes=15)
    payload = {
        "sub": str(user_id),  # User ID as the subject
        "exp": datetime.utcnow() + expiration,  # Token expiration
        "type": "access",  # Token type
    }
    return jwt.encode(payload, current_app.config["SECRET_KEY"], algorithm="HS256")

```

Code Snippet: Authentication

```
@bp.route('/register', methods=['POST'])
def register():
    data = request.get_json()
    username = data.get('username')
    email = data.get('email')
    password = data.get('password')


@bp.route('/login', methods=['POST'])
def login():
    data = request.get_json()
    email = data.get('email')
    password = data.get('password')


@bp.route('/refresh', methods=['POST'])
def refresh_token():
    refresh_token = request.cookies.get("refresh_token")
bp.route('/logout', methods=['POST'])

```

Code Snippet: Admin Features

```python
# Register a new admin ( AdminS only )
@bp.route('/register', methods=['POST'])
@admin_required
def register_admin():
    data = request.get_json()
    username = data.get('username')
    email = data.get('email')
    password = data.get('password')
# Get all users (AdminS only)
@bp.route('/users', methods=['GET'])
@admin_required
def get_all_users():
    users = User.query.all()
    result = [{"id": user.id, "username": user.username, "
    email": user.email, "is_admin": user.is_admin} for user
    in users]
# Delete a user (AdminS only)
@bp.route('/users/<int:user_id>', methods=['DELETE'])
@admin_required
def delete_user(user_id):
```

Code Snippet: Fantasy Team Creation

```
@bp.route('/create', methods=['POST'])
def create_fantasy_team():
    data = request.get_json()
    user_id = data.get('user_id')
    team_name = data.get('team_name')

@bp.route('/update/<int:team_id>', methods=['PUT'])
def update_fantasy_team(team_id):
    data = request.get_json()
    new_team_name = data.get('team_name')
    new_points = data.get('points')


@bp.route('/delete/<int:team_id>', methods=['DELETE'])
def delete_fantasy_team(team_id):

    team = FantasyTeam.query.get(team_id)

```

Code Snippet: Player Management

```python
@bp.route('/add', methods=['POST'])
@user_required
def add_player():
    data = request.get_json()
    team_id = data.get('team_id')
    name = data.get('name')
    position = data.get('position')

@bp.route('/<int:team_id>/players', methods=['GET'])
def get_players(team_id):
    team = FantasyTeam.query.get(team_id)

@bp.route('/update/<int:player_id>', methods=['PUT'])
def update_player(player_id):
    data = request.get_json()
    player = Player.query.get(player_id)

@bp.route('/delete/<int:player_id>', methods=['DELETE'])
def delete_player(player_id):
    player = Player.query.get(player_id)
```

Code Snippet: Match Management

```
@bp.route('/<int:match_id>', methods=['GET'])
@user_required
def get_match(match_id):
    match = Match.query.get(match_id)
    if match:
        return jsonify({
            "home_team": match.home_team,
            "away_team": match.away_team,
            "score": match.score
        }), 200
    return jsonify({"message": "Match not found"}), 404

```

Code Snippet: Leaderboard

```
@bp.route('/get', methods=['GET'])
@user_required
def get_leaderboard():
    teams = FantasyTeam.query.options(joinedload(FantasyTeam
    .user)) \
        .order_by(FantasyTeam.points.desc()).all()

    leaderboard = [
        {
            "score": team.points,
            "team_name": team.name,
            "user_id": team.user.id,
            "username": team.user.username
        }
        for team in teams
    ]
    return jsonify(leaderboard), 200

```

Code Snippet: Twitter Integration

```
@bp.route('/share', methods=['POST'])
@user_required
def share_on_twitter():
    data = request.get_json()
    message = data.get('message')

    if not message:
        return jsonify({"message": "Message content is required"}), 400

    try:
        twitter_api.update_status(status=message)
        return jsonify({"message": "Shared successfully on Twitter"}), 200
    except tweepy.errors.Forbidden as e:
        return jsonify({"message": f"Twitter API error (Forbidden): {e}"}), 403
    except tweepy.errors.TweepyException as e:
        return jsonify({"message": f"Failed to share on Twitter: {e}"}), 500
```

Challenges

- Lack of free external APIs for real-time data.
- Reliance on seed data for testing.
- Performance optimization under high load.
- Future work: Integrate live match data and improve scalability.

Conclusion

- The Tunisian Fantasy API delivers a secure, scalable, and user-friendly platform.
- Enhances fan engagement and celebrates Tunisia's football culture.
- Future enhancements: Live match data, mobile app, advanced analytics.

Questions?

Thank You!

Your feedback and questions are welcome!