INSIDER CASE

İremsu Özdemir

Base URL https://insider-league-app-00404103e92e.herokuapp.com
Github Link : https://github.com/iremsuozdemir/InsiderCase

Description

My project is a full-featured Football League Simulator built with a Go backend and a vanilla JavaScript frontend. It allows users to create and manage realistic football league competitions. I've implemented functionality to add teams with customizable strength ratings (from 1 to 100), and the system automatically generates fair fixtures using a round-robin scheduling algorithm.

The match simulation engine I developed incorporates probabilistic logic that considers team strength, home-field advantage, recent performance bonuses, and realistic scoring behavior. While stronger teams are more likely to win, the simulation still allows for upsets and draws based on strength differences. All match results are recorded, and the league table is updated following standard Premier League rules—3 points for a win, 1 for a draw.

I also store all match and team data in a PostgreSQL database with a properly normalized relational schema. The frontend provides an intuitive user interface to manage teams, simulate matches (week-by-week or an entire season), view live-updating standings, and browse match results by week. I've exposed all core operations through RESTful API endpoints, added CORS support for frontend-backend communication, and configured the app for deployment with environment-based port handling. Overall, it's a complete end-to-end platform for simulating and managing football leagues.

Project Setup

Option 1: Running Locally

If you'd like to run the project on your own machine, start by making sure you have Go version 1.24.4 or higher installed. Clone the repository and navigate to the insider-league directory, then run go mod tidy to install the necessary dependencies including the PostgreSQL driver. You'll need PostgreSQL running locally with a football_league database and appropriate user privileges set up.Before running the application, set the DATABASE_URL environment variable to your local PostgreSQL connection string by running export DATABASE_URL="postgres://username:password@localhost:5432/football_league?sslmode=disable" (replace username and password with your actual credentials). This environment variable is required as the application won't start without it. Then launch the server by running go run main.go router.go from the project directory.Once running, you'll see confirmation messages and the server will start on port 8080. Open http://localhost:8080 in your browser to access the web interface where you can create leagues, add teams, simulate matches, and view real-time standings. You can also use this URL as the base for API testing with Postman. The local version runs independently from any deployed version with its own separate database.

Option 2: Accessing the Deployed Application

https://insider-league-app-00404103e92e.herokuapp.com

SQL Schema

```
CREATE TABLE IF NOT EXISTS teams (
        id SERIAL PRIMARY KEY,
        name VARCHAR(100) NOT NULL UNIQUE,
        strength INTEGER NOT NULL CHECK (strength >= 1 AND strength <= 100)
);

CREATE TABLE IF NOT EXISTS leagues (
        id SERIAL PRIMARY KEY,
        name VARCHAR(100) NOT NULL,
        current_week INTEGER DEFAULT 0,
        total_weeks INTEGER NOT NULL,
        status VARCHAR(20) DEFAULT 'active' CHECK (status IN ('active', 'completed', 'paused'))
);

CREATE TABLE IF NOT EXISTS league_teams (
        id SERIAL PRIMARY KEY,
        league_id INTEGER REFERENCES leagues(id) ON DELETE CASCADE,
        team_id INTEGER REFERENCES teams(id) ON DELETE CASCADE,
        UNIQUE(league_id, team_id)
);

CREATE TABLE IF NOT EXISTS matches (
        id SERIAL PRIMARY KEY,
        league_id INTEGER REFERENCES leagues(id) ON DELETE CASCADE,
        week_number INTEGER NOT NULL,
        home_team_id INTEGER REFERENCES teams(id),
        away_team_id INTEGER REFERENCES teams(id),
        home_score INTEGER DEFAULT NULL,
        away_score INTEGER DEFAULT NULL,
        played BOOLEAN DEFAULT FALSE,
        played_at INTEGER DEFAULT NULL,
        CHECK (home_team_id != away_team_id)
);

CREATE TABLE IF NOT EXISTS team_stats (
        id SERIAL PRIMARY KEY,
        league_id INTEGER REFERENCES leagues(id) ON DELETE CASCADE,
        team_id INTEGER REFERENCES teams(id) ON DELETE CASCADE,
        played INTEGER DEFAULT 0,
        won INTEGER DEFAULT 0,
        drawn INTEGER DEFAULT 0,
        lost INTEGER DEFAULT 0,
        goals_for INTEGER DEFAULT 0,
        goals_against INTEGER DEFAULT 0,
        points INTEGER DEFAULT 0,
        goal_difference INTEGER GENERATED ALWAYS AS (goals_for - goals_against) STORED,
        UNIQUE(league_id, team_id)
);

INSERT INTO teams (name, strength) VALUES
        ('Arsenal', 70),
        ('Chelsea', 85),
        ('Liverpool', 75),
        ('Manchester City', 92)
ON CONFLICT (name) DO NOTHING;
```

SQL Queries

Get all teams:

```
SELECT id, name, strength FROM teams ORDER BY name
```

Add team:

```
INSERT INTO teams (name, strength) VALUES ($1, $2)
```

Get team by ID:

```
SELECT id, name, strength FROM teams WHERE id = $1
```

Get team by name:

```
SELECT id, name, strength FROM teams WHERE name = $1
```

Update team:

```
UPDATE teams SET name = $1, strength = $2 WHERE id = $3
```

Delete team:

```
DELETE FROM teams WHERE id = $1
```

Check if team exists:

```
SELECT EXISTS(SELECT 1 FROM teams WHERE name = $1)
```

Count total teams:

```
SELECT COUNT(*) FROM teams
```

Create league:

```
INSERT INTO leagues (name, total_weeks) VALUES ($1, $2) RETURNING id
```

Get league status:

```
SELECT current_week, total_weeks, status FROM leagues WHERE id = $1
```

Update current week:

```
UPDATE leagues SET current_week = $1 WHERE id = $2
```

Add teams to league:

```
INSERT INTO league_teams (league_id, team_id) VALUES ($1, $2)
```

Save played match:

```
INSERT INTO matches (league_id, week_number, home_team_id, away_team_id, home_score,
away_score, played)
VALUES ($1, $2, $3, $4, $5, $6, true)
```

Get all matches:

```
SELECT ht.name, at.name, m.home_score, m.away_score, m.week_number
FROM matches m
JOIN teams ht ON m.home_team_id = ht.id
JOIN teams at ON m.away_team_id = at.id
WHERE m.league_id = $1 AND m.played = true
ORDER BY m.week_number, m.id
```

Get matches for a specific week:

```
SELECT ht.name, at.name, m.home_score, m.away_score, m.week_number
FROM matches m
```

```
JOIN teams ht ON m.home_team_id = ht.id
JOIN teams at ON m.away_team_id = at.id
WHERE m.league_id = $1 AND m.week_number = $2 AND m.played = true
ORDER BY m.id
```

Save fixture (not yet played):

```
INSERT INTO matches (league_id, week_number, home_team_id, away_team_id, played)
VALUES ($1, $2, $3, $4, false)
```

Get match schedule:

```
SELECT ht.name, at.name, m.week_number
FROM matches m
JOIN teams ht ON m.home_team_id = ht.id
JOIN teams at ON m.away_team_id = at.id
WHERE m.league_id = $1
ORDER BY m.week_number, m.id
```

Initialize team stats:

```
INSERT INTO team_stats (league_id, team_id) VALUES ($1, $2)
```

Update team stats:

```
UPDATE team_stats
SET played = $1, won = $2, drawn = $3, lost = $4,
   goals_for = $5, goals_against = $6, points = $7
WHERE league_id = $8 AND team_id = (SELECT id FROM teams WHERE name = $9)
```

Get league standings:

```
SELECT t.name, ts.played, ts.won, ts.drawn, ts.lost,
      ts.goals_for, ts.goals_against, ts.points, ts.goal_difference
FROM team_stats ts
JOIN teams t ON ts.team_id = t.id
WHERE ts.league_id = $1
ORDER BY ts.points DESC, ts.goal_difference DESC, ts.goals_for DESC
```

Delete league with all related data (Transactional):

```
BEGIN;
DELETE FROM matches WHERE league_id = $1;
DELETE FROM team_stats WHERE league_id = $1;
DELETE FROM league_teams WHERE league_id = $1;
DELETE FROM leagues WHERE id = $1;
COMMIT;
```

Delete all teams and related data (Transactional):

```
BEGIN;
DELETE FROM matches;
DELETE FROM team_stats;
DELETE FROM league_teams;
DELETE FROM leagues;
DELETE FROM teams;
COMMIT;
```

Count total fixtures:

```
SELECT COUNT(*) FROM matches WHERE league_id = $1
```

Get current week:

```
SELECT current_week FROM leagues WHERE id = $1
```

Get total weeks:

```
SELECT total_weeks FROM leagues WHERE id = $1
```

API Endpoints

League Operations:
- `POST /api/league` - Create new league
- `DELETE /api/league` - Clear league
- `GET /api/league/status` - Get league info
- `POST /api/init-db` - Setup database

Playing Games:
- `POST /api/league/play-week` - Play one week
- `POST /api/league/play-all` - Play entire season

View Results:
- `GET /api/league/table` - League standings

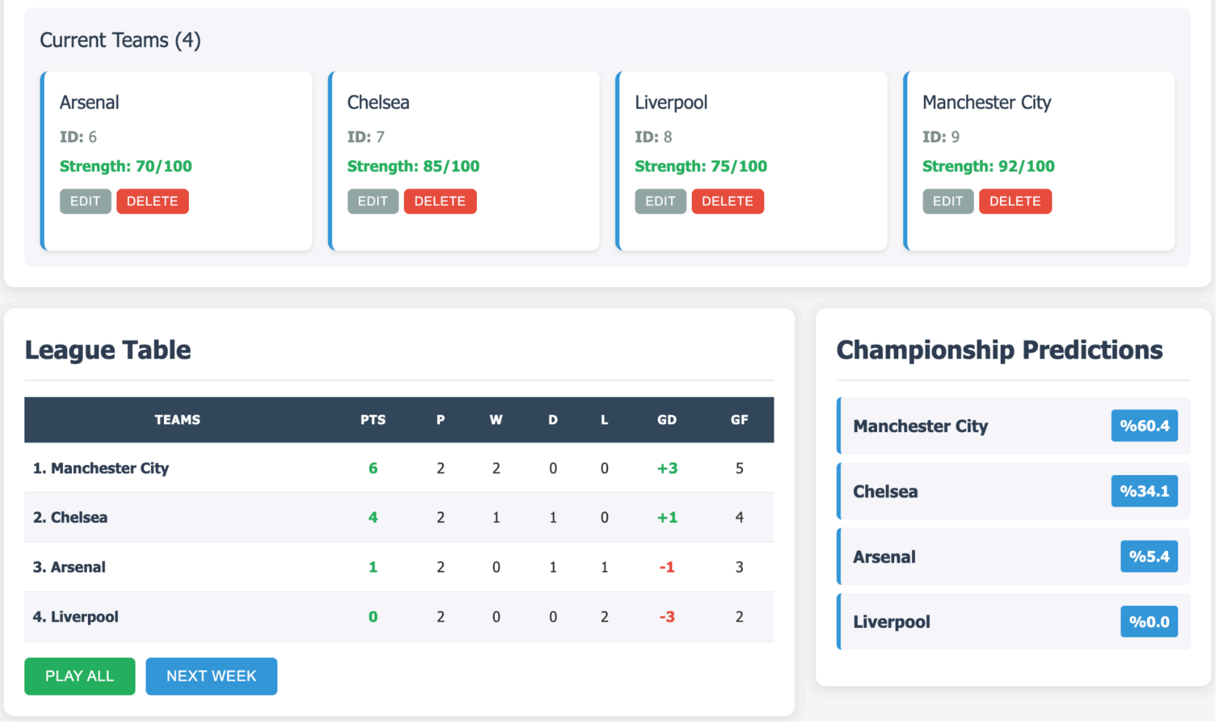
- `GET /api/league/matches` - All match results
- `GET /api/league/matches/week/{week}` - Specific week results
- `GET /api/league/schedule` - Upcoming matches
- `GET /api/league/predictions` - Championship predictions

Team Management:
- `GET /api/teams` - List all teams
- `POST /api/teams` - Add team
- `GET /api/teams/{id}` - Get team details
- `PUT /api/teams/{id}` - Update team
- `DELETE /api/teams/{id}` - Delete team
- `DELETE /api/clear-teams` - Delete all teams

System:
- `GET /api/health` - Check if API is working

Main Pages:
- `GET /` - Web interface (frontend)
- `GET /static/` - CSS/JS files

Week 2/6



Current Teams (4)

Arsenal
ID: 6
Strength: 70/100

Chelsea
ID: 7
Strength: 85/100

Liverpool
ID: 8
Strength: 75/100

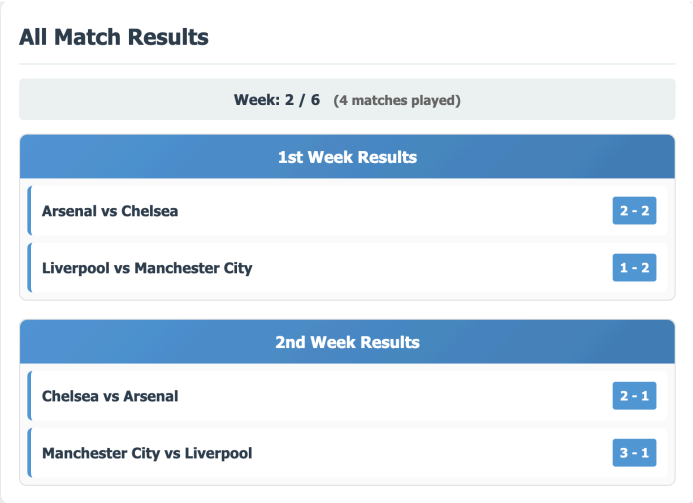
Manchester City
ID: 9
Strength: 92/100

League Table

TEAMS	PTS	P	W	D	L	GD	GF
1. Manchester City	6	2	2	0	0	+3	5
2. Chelsea	4	2	1	1	0	+1	4
3. Arsenal	1	2	0	1	1	-1	3
4. Liverpool	0	2	0	0	2	-3	2

Championship Predictions

Manchester City	%60.4
Chelsea	%34.1
Arsenal	%5.4
Liverpool	%0.0



All Match Results

Week: 2 / 6 (4 matches played)

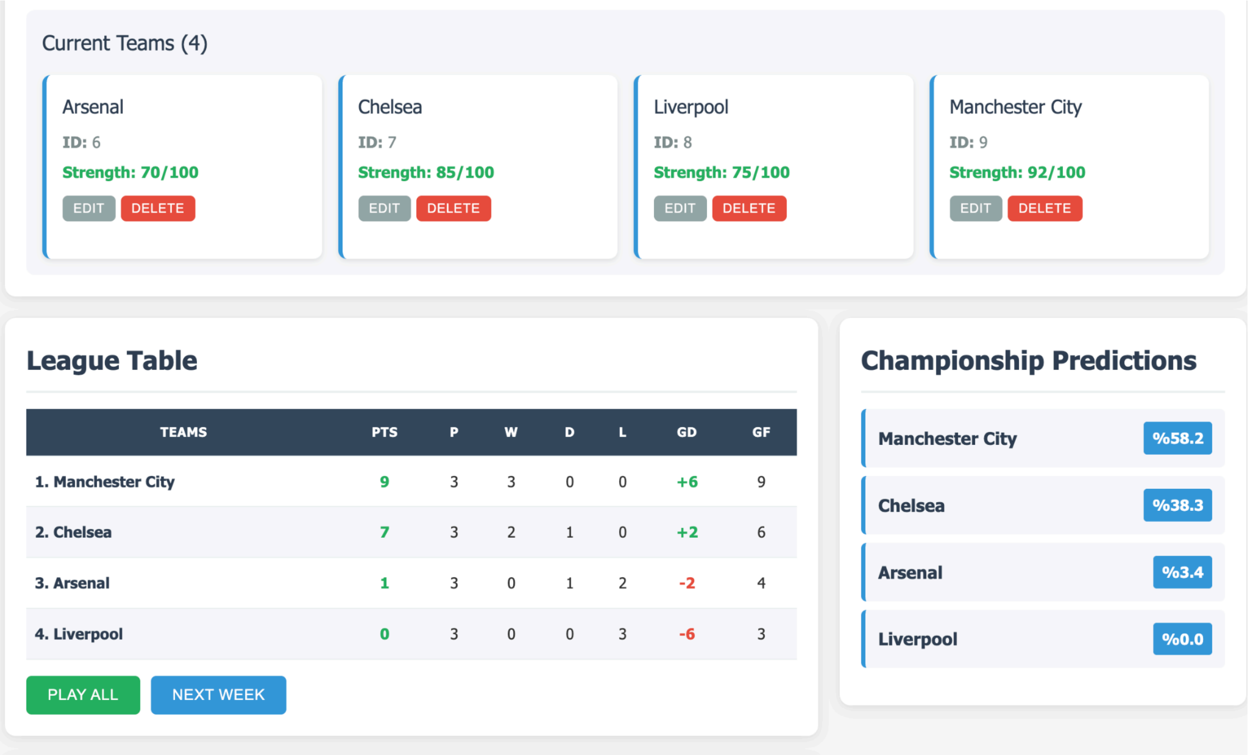
1st Week Results

Arsenal vs Chelsea	2 - 2
Liverpool vs Manchester City	1 - 2

2nd Week Results

Chelsea vs Arsenal	2 - 1
Manchester City vs Liverpool	3 - 1

Week 3/6



Current Teams (4)

Arsenal
ID: 6
Strength: 70/100

Chelsea
ID: 7
Strength: 85/100

Liverpool
ID: 8
Strength: 75/100

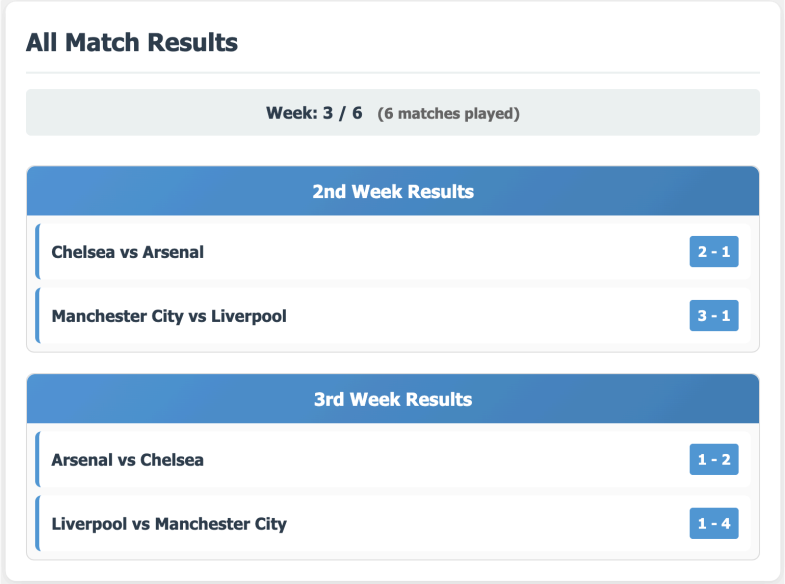
Manchester City
ID: 9
Strength: 92/100

League Table

TEAMS	PTS	P	W	D	L	GD	GF
1. Manchester City	9	3	3	0	0	+6	9
2. Chelsea	7	3	2	1	0	+2	6
3. Arsenal	1	3	0	1	2	-2	4
4. Liverpool	0	3	0	0	3	-6	3

Championship Predictions

Manchester City	%58.2
Chelsea	%38.3
Arsenal	%3.4
Liverpool	%0.0



All Match Results

Week: 3 / 6 (6 matches played)

2nd Week Results

Chelsea vs Arsenal	2 - 1
Manchester City vs Liverpool	3 - 1

3rd Week Results

Arsenal vs Chelsea	1 - 2
Liverpool vs Manchester City	1 - 4