



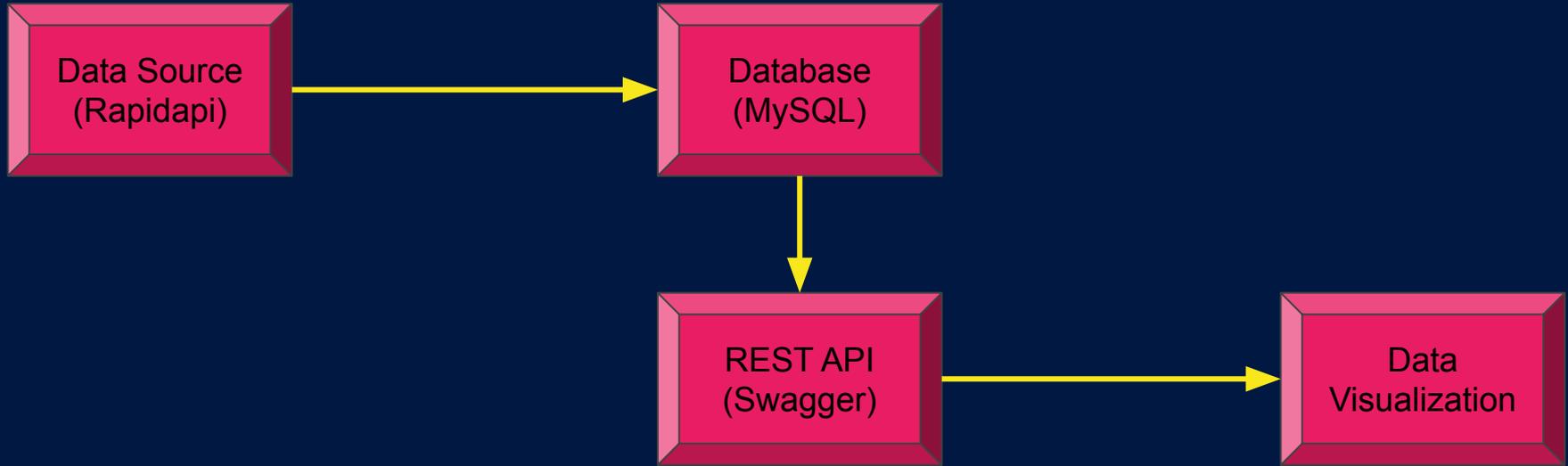
Premier
League

My Football

Overview

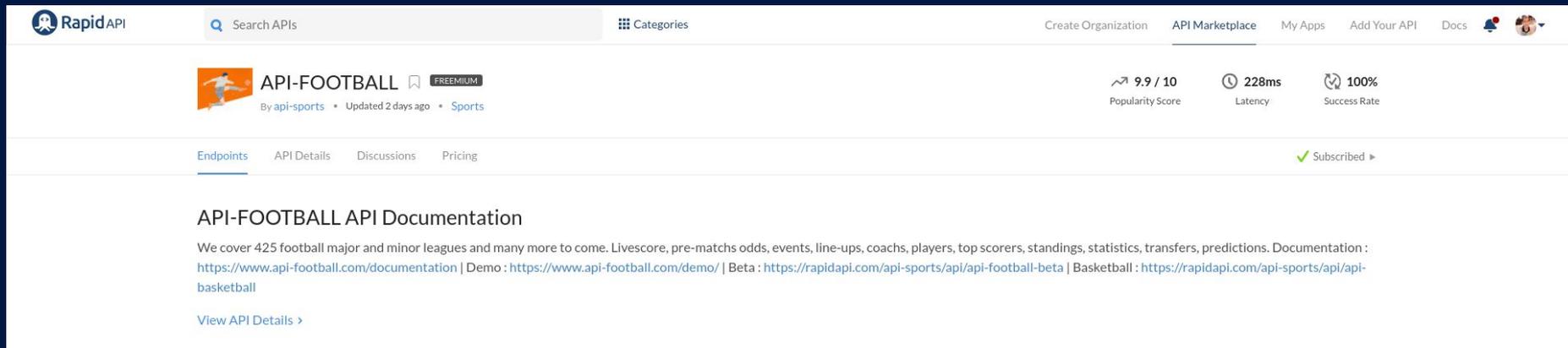
This project is determined to analyze match result of Football Premier League from the between 2016 - 2019, and visualize data on the webpage by chart.

Overall Architecture



Data Source

rapidapi.com



The screenshot shows the RapidAPI website interface. At the top, there is a search bar with the text "Search APIs" and a "Categories" button. The main header includes navigation links: "Create Organization", "API Marketplace", "My Apps", "Add Your API", "Docs", and a user profile icon. The main content area features the "API-FOOTBALL" listing, which is marked as "FREEMIUM". The listing includes a small icon of a person running, the text "By api-sports", "Updated 2 days ago", and "Sports". To the right of the listing, there are three performance metrics: "Popularity Score" of 9.9 / 10, "Latency" of 228ms, and "Success Rate" of 100%. Below the listing, there are tabs for "Endpoints", "API Details", "Discussions", and "Pricing". A "Subscribed" status is indicated with a green checkmark. The main content area displays the title "API-FOOTBALL API Documentation" and a paragraph of text: "We cover 425 football major and minor leagues and many more to come. Livescore, pre-matches odds, events, line-ups, coaches, players, top scorers, standings, statistics, transfers, predictions. Documentation : <https://www.api-football.com/documentation> | Demo : <https://www.api-football.com/demo/> | Beta : <https://rapidapi.com/api-sports/api/api-football-beta> | Basketball : <https://rapidapi.com/api-sports/api/api-basketball>". A "View API Details" link is located at the bottom of the content area.

Collection Method

```
const unirest = require("unirest");
const mysql = require("mysql");
const dbConfig = require("../config/dbconfig");
const apiConfig = require("../config/apiconfig");
```

```
const leagueId = ["524", "2", "37", "56"];
const db = mysql.createConnection(dbConfig.dbOptions);
```

```
leagueId.map(id => {
  var req = unirest(
    "GET",
    `https://api-football-v1.p.rapidapi.com/v2/fixtures/league/${id}`
  );

  req.query({
    timezone: "Asia/Bangkok"
  });

  req.headers({
    "x-rapidapi-host": "api-football-v1.p.rapidapi.com",
    "x-rapidapi-key": `${apiConfig.apiKey}`
  });

  req.end(function(res) {
    if (res.error) throw new Error(res.error);

    var json = res.body.api.fixtures;

    json.map(data => {
      db.connect(() => {
        db.query(
          `insert into competition (compId, seasonId, eventDate, status, home, away, goals, scores)
          values ('${data.fixture_id}', '${id}', '${data.event_date}', '${data.status}', '${data.homeTeam.team_id}',
          '${data.awayTeam.team_id}', '{"home": "${data.goalsHomeTeam}", "away": "${data.goalsAwayTeam}"}, '${data.score.fulltime}')
          `);
      });
    });
  });
});
```

```
const db = mysql.createConnection(dbConfig.dbOptions);
```

```
leagueId.map(id => {
  var req = unirest(
    "GET",
    `https://api-football-v1.p.rapidapi.com/v2/teams/league/${id}`
  );

  req.headers({
    "x-rapidapi-host": "api-football-v1.p.rapidapi.com",
    "x-rapidapi-key": `${apiConfig.apiKey}`
  });

  req.end(res => {
    if (res.error) throw new Error(res.error);

    var json = res.body.api.teams;

    json.map(data => {
      db.connect(() => {
        db.query(
          `insert into team (teamId, teamName, teamLogo)
          values ('${data.team_id}', '${data.name}', '${data.logo}')
          on duplicate key update teamId = '${data.team_id}'`
        );
      });
    });
  });
});
```

Database

MySQL

rankingId	int
seasonId	int
teamId	int
order	int
win	int
loss	int
draw	int
point	int
goalDiff	int

teamId	int
teamName	string
teamLogo	string

seasonId	int
year	int
seasonStart	timestamp
seasonEnd	timestamp
teams	array
matches	array
ranking	array

compId	int
seasonId	int
eventDate	Date
home	int
away	int
goal	array
score	text
status	string

Statistic ID	int
Shots on goal	array
Shots off goal	array
Total shots	array
Blocked shots	array
Shots insidebox	array
Shots outsidebox	array
Fouls	array
Corner kicks	array
Offsides	array
Ball possession	array
Yello card	array
Red card	array
Goalkeeper saves	array
Total passes	array
Pass accurate	hayato kawai
Passes%	array

API

-GraphQL

-SWAGGER

Swagger UI for Premier League API (v2.0.0, 0413). The response body shows a JSON array of team statistics:

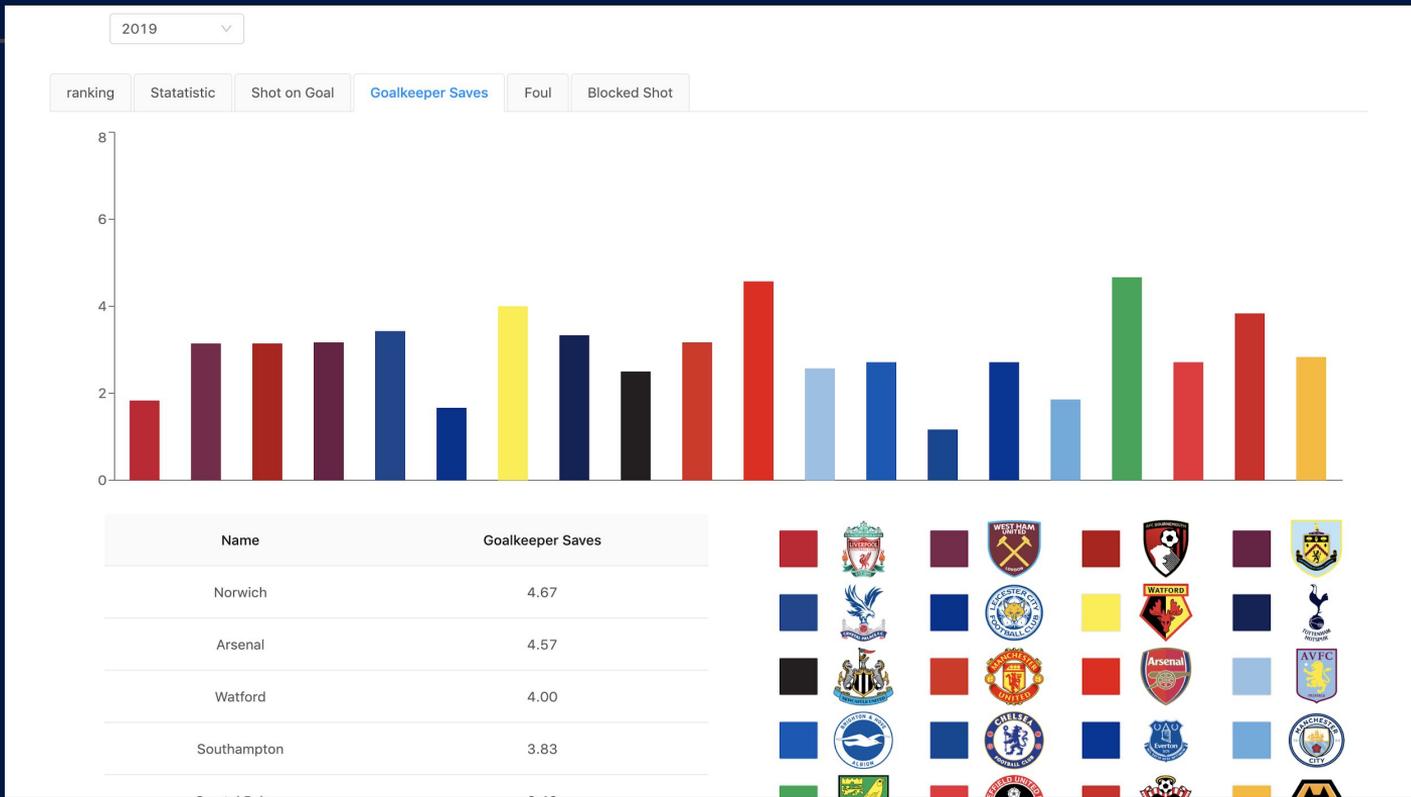
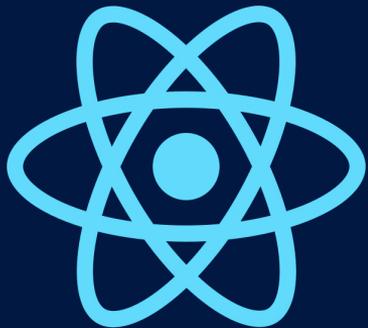
```
[
  {
    "draw": 1,
    "goalsDiff": 19,
    "loss": 0,
    "points": 37,
    "ranking": 1,
    "teamId": 40,
    "teamLogo": "https://media.api-football.com/teams/40.png",
    "teamName": "Liverpool",
    "win": 12
  },
  {
    "draw": 2,
    "goalsDiff": 23,
```

```
def get_team_by_season(seasonId):
    with db_cursor() as cs:
        cs.execute("""
            SELECT ranking.order,team.teamId,team.teamName, team.teamLogo,ranking.win, ranking.lose, ranking.draw , ranking.goalsDiff, ranking.points
            FROM team
            INNER JOIN ranking
            ON team.teamId = ranking.teamId
            WHERE ranking.seasonId = %s
            """,[seasonId])

        result = [models.TeamBySeason(*row) for row in cs.fetchall()]
        return result
```

<Recharts />

Data Visualization



Thank You