NBA Player Stats Tracker - Python Project

Project: NBA Player Stats Tracker

Objective:

Build a Python application that allows users to fetch and analyze the performance stats of NBA players from the current season.

The app will fetch player data via an API and provide functionality for basic statistical analysis, such as comparing player stats or calculating averages for specific metrics.

Step 1: API Integration

Use a free NBA stats API like balldontlie.io to get player data.

Install necessary libraries: requests to handle API calls and json to process the response.

Example code for fetching player stats:

```
import requests


def get_player_stats(player_name):
    url = f"https://www.balldontlie.io/api/v1/players?search={player_name}"
    response = requests.get(url)
    player_data = response.json()
    if player_data['data']:
        player_id = player_data['data'][0]['id']
                                                                stats_url        =
```

```
f"https://www.balldontlie.io/api/v1/season_averages?season=2023&player_ids[]={player_id}
"
        stats_response = requests.get(stats_url)
        stats = stats_response.json()
        if stats['data']:
            return stats['data'][0]
        else:
            return f"No stats available for {player_name}"
    else:
        return f"Player {player_name} not found"
```

Step 2: Basic User Interface

Allow the user to input the name of an NBA player and fetch the player's season statistics like points, assists, rebounds, field goal percentage.

Example code for user interface:

```
def display_player_stats():
    player_name = input("Enter the name of the NBA player: ")
    stats = get_player_stats(player_name)
    if isinstance(stats, dict):
        print(f"Stats for {player_name}:")
        print(f"Points per Game: {stats['pts']}")
        print(f"Assists per Game: {stats['ast']}")
        print(f"Rebounds per Game: {stats['reb']}")
        print(f"Field Goal Percentage: {stats['fg_pct']}")
```

```
    else:
        print(stats)
```

Step 3: Statistical Analysis Features

Allow users to compare two players' stats side by side or calculate the average performance for a player across seasons.

Example code for comparing players:

```
def compare_players(player1, player2):
    stats1 = get_player_stats(player1)
    stats2 = get_player_stats(player2)

    if isinstance(stats1, dict) and isinstance(stats2, dict):
        print(f"{player1} vs {player2}:")
        print(f"Points per Game: {stats1['pts']} vs {stats2['pts']}")
        print(f"Assists per Game: {stats1['ast']} vs {stats2['ast']}")
        print(f"Rebounds per Game: {stats1['reb']} vs {stats2['reb']}")
    else:
        print("Error fetching player data.")
```

Step 4: Bonus Feature - Search by Team

Extend the functionality to allow searching for all players on a specific team.

Example code for team search:

```
def get_team_players(team_name):
    url = f"https://www.balldontlie.io/api/v1/teams?search={team_name}"
    response = requests.get(url)
    team_data = response.json()
    if team_data['data']:
        team_id = team_data['data'][0]['id']
        players_url = f"https://www.balldontlie.io/api/v1/players?team_ids[]={team_id}"
        players_response = requests.get(players_url)
        players = players_response.json()
        if players['data']:
            for player in players['data']:
                print(player['first_name'], player['last_name'])
        else:
            print(f"No players found for team {team_name}")
    else:
        print(f"Team {team_name} not found")
```