1. **Background**:

The Fédération Internationale de Football Association (FIFA) is a non-profit organization which describes itself as an international governing body of association football, fútsal, beach soccer, and e-football. It is the highest governing body of football.

There is also a video game called FIFA, which is a series of association football video games or football simulator, released annually by Electronic Arts under the EA Sports label. When the series began in late 1993, it was notable for being the first to have an official license from FIFA, the world governing body of football.

Motivation:

FIFA is a world-renowned football video game with a large video game player base. In reality, the number of football fans is also extremely large. A well-informed football player database and beautiful display interface can help the fans to understand the basic situation of each player's clubs and better participate in online and offline football matches. With enough information to support, fans can also predict the status of future players or events.

Goal:

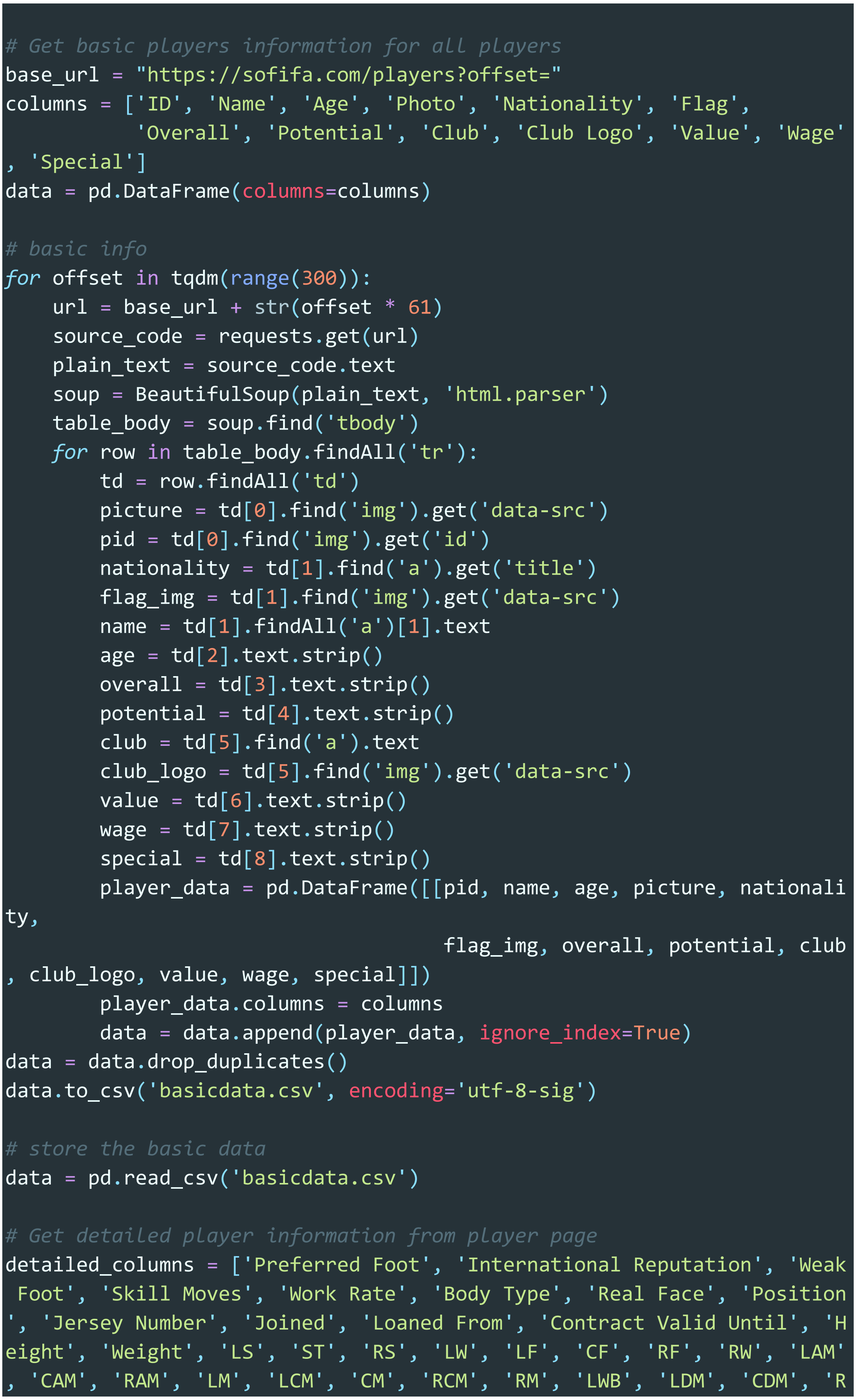
We hope to make a database with GUI to show the basic information of each football star to the football fans or football event audiences, including their photos, nationalities, clubs, height, weight, position, etc. We will use dynamic radar chart to display player's ability value. Based on it, we plan to introduce some machine learning methods to predict the value and wage of each player and recommend the appropriate position according to each player's ability value. Users can also change the player's ability value so that the system gives different predictions, in order to guide the actual ball game or electronic game.

2. **Data Sources**:

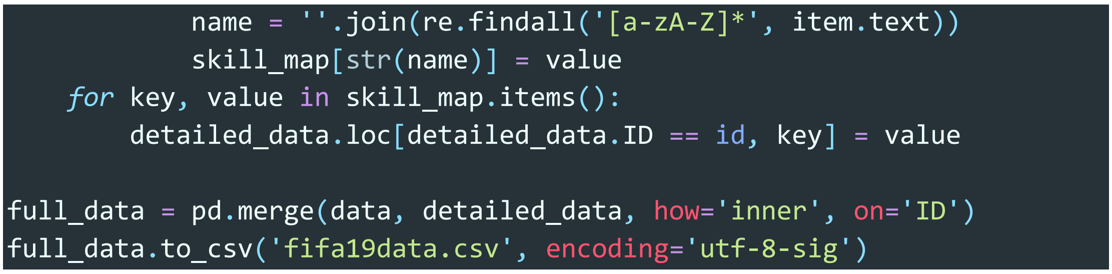
Our data is scraped from https://sofifa.com/, which is a professional FIFA information website.

Attached is our Python WebCrawler code.

```python
import pandas as pd
import re
import requests
from bs4 import BeautifulSoup
from tqdm import tqdm
```

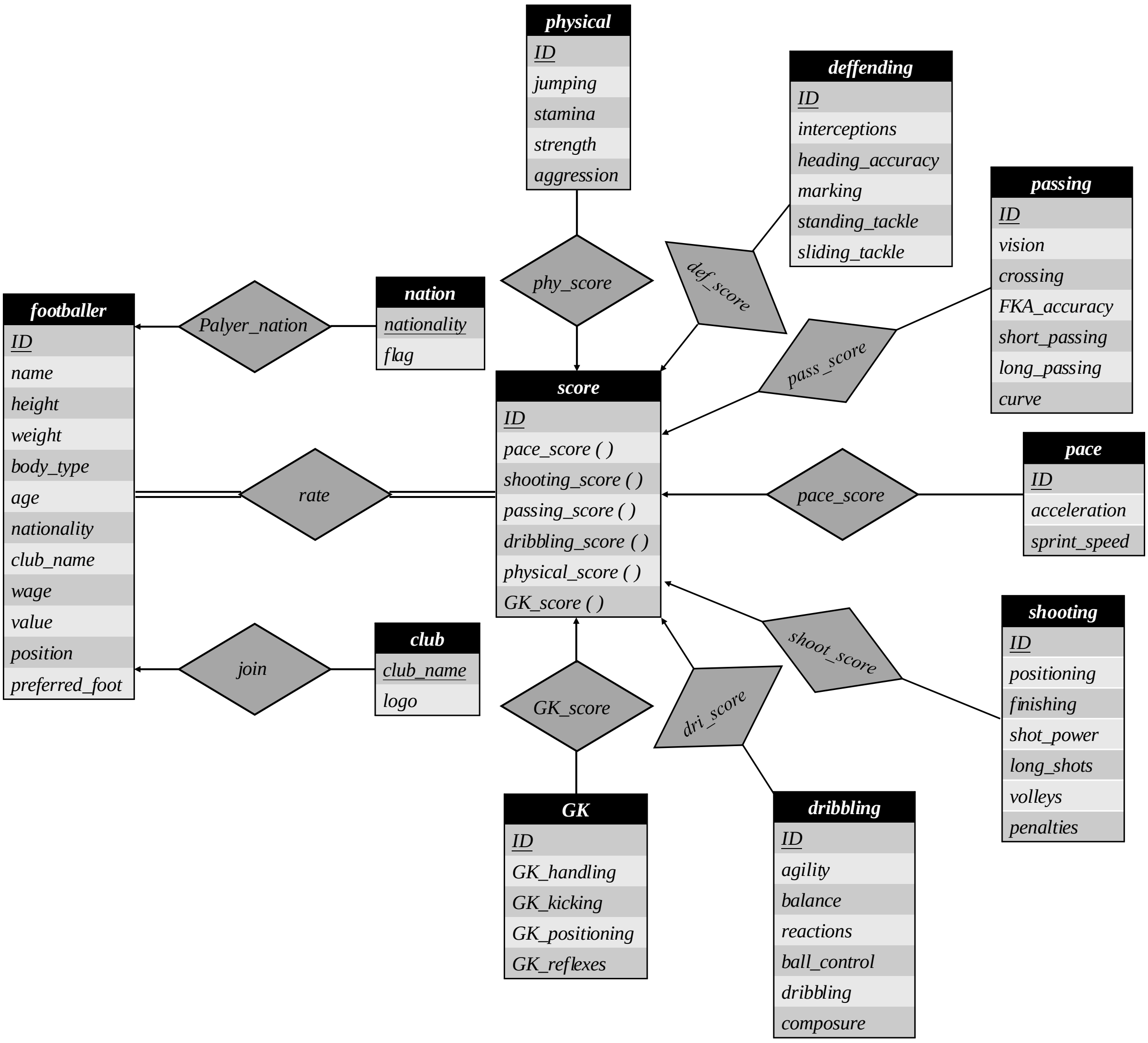


```
# Get basic players information for all players
base_url = "https://sofifa.com/players?offset="
columns = ['ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
           'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage'
, 'Special']
data = pd.DataFrame(columns=columns)

# basic info
for offset in tqdm(range(300)):
    url = base_url + str(offset * 61)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    table_body = soup.find('tbody')
    for row in table_body.findAll('tr'):
        td = row.findAll('td')
        picture = td[0].find('img').get('data-src')
        pid = td[0].find('img').get('id')
        nationality = td[1].find('a').get('title')
        flag_img = td[1].find('img').get('data-src')
        name = td[1].findAll('a')[1].text
        age = td[2].text.strip()
        overall = td[3].text.strip()
        potential = td[4].text.strip()
        club = td[5].find('a').text
        club_logo = td[5].find('img').get('data-src')
        value = td[6].text.strip()
        wage = td[7].text.strip()
        special = td[8].text.strip()
        player_data = pd.DataFrame([[pid, name, age, picture, nationali
ty,
                                     flag_img, overall, potential, club
, club_logo, value, wage, special]])
        player_data.columns = columns
        data = data.append(player_data, ignore_index=True)
data = data.drop_duplicates()
data.to_csv('basicdata.csv', encoding='utf-8-sig')

# store the basic data
data = pd.read_csv('basicdata.csv')

# Get detailed player information from player page
detailed_columns = ['Preferred Foot', 'International Reputation', 'Weak
 Foot', 'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position
', 'Jersey Number', 'Joined', 'Loaned From', 'Contract Valid Until', 'H
eight', 'Weight', 'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW', 'LAM'
, 'CAM', 'RAM', 'LM', 'LCM', 'CM', 'RCM', 'RM', 'LWB', 'LDM', 'CDM', 'R
DM', 'RWB', 'LB', 'LCB', 'CB', 'RCB', 'RB', 'Crossing', 'Finishing', 'H
eadingAccuracy',
                    'ShortPassing', 'Volleys', 'Dribbling', 'Curve', 'F
KAccuracy', 'LongPassing', 'BallControl', 'Acceleration', 'SprintSpeed'
, 'Agility', 'Reactions', 'Balance', 'ShotPower', 'Jumping', 'Stamina',
 'Strength', 'LongShots', 'Aggression', 'Interceptions', 'Positioning',
 'Vision', 'Penalties', 'Composure', 'Marking', 'StandingTackle', 'Slid
ingTackle', 'GKDiving', 'GKHandling', 'GKKicking', 'GKPositioning', 'GK
Reflexes', 'ID']
detailed_data = pd.DataFrame(index=range(
    0, data.count()[0]), columns=detailed_columns)
detailed_data.ID = data.ID.values

player_data_url = 'https://sofifa.com/player/'
for i in tqdm(range(data.ID.shape[0])):
    id = data.ID[i]

    url = player_data_url + str(id)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    skill_map = {}
    columns = soup.find('div', {'class': 'teams'}).find(
        'div', {'class': 'columns'}).findAll('div', {'class': 'column c
ol-6'})
    columns.append(soup.find('div', {'class': 'teams'}).find(
        'div', {'class': 'columns'}).findAll('div', {'class': 'bp3-
callout'})[0])
    for column in columns:
        skills = column.findAll('li')
        for skill in skills:
            if(skill.find('label') != None):
                label = skill.find('label').text
                value = skill.text.replace(label, '').strip()
                skill_map[label] = value
    meta_data = soup.find('div', {'class': 'meta'}).text.split(' ')
    length = len(meta_data)
    weight = meta_data[length - 1]
    height = meta_data[length - 2].split('\'')[0] + '\'' + \
        meta_data[length - 2].split('\'')[1].split('\"')[0]
    skill_map["Height"] = height
    skill_map['Weight'] = weight

    sections = soup.find('article').findAll(
        'div', {'class': 'column col-4'})[:-1]
    for section in sections:
        items = section.find('ul').findAll('li')
        for item in items:
            value = int(re.findall(r'\d+', item.text)[0])
```



```
            name = ''.join(re.findall('[a-zA-Z]*', item.text))
            skill_map[str(name)] = value
    for key, value in skill_map.items():
        detailed_data.loc[detailed_data.ID == id, key] = value

full_data = pd.merge(data, detailed_data, how='inner', on='ID')
full_data.to_csv('fifa19data.csv', encoding='utf-8-sig')
```

3. Database Schema:


physical
ID
jumping
stamina
strength
aggression
deffending
ID
interceptions
heading_accuracy
marking
standing_tackle
sliding_tackle
passing
ID
vision
crossing
FKA_accuracy
short_passing
long_passing
curve
phy_score
def_score
pass_score
footballer
ID
name
height
weight
body_type
age
nationality
club_name
wage
value
position
preferred_foot
Palyer_nation
nation
nationality
flag
score
ID
pace_score ( )
shooting_score ( )
passing_score ( )
dribbling_score ( )
physical_score ( )
GK_score ( )
rate
pace_score
pace
ID
acceleration
sprint_speed
join
club
club_name
logo
GK_score
shoot_score
dri_score
shooting
ID
positioning
finishing
shot_power
long_shots
volleys
penalties
GK
ID
GK_handling
GK_kicking
GK_positioning
GK_reflexes
dribbling
ID
agility
balance
reactions
ball_control
dribbling
composure