Data pipeline showcase

Step 1: Get list of all match reports links

1: Create request object

2: Use bs4 to parse html

3: Search all HTML object that starts with ‘a’ (link object)

4: if the link’s text is “Match Report”, append it to the empty list

5: You now have a list that contains every link for match reports

Scores & Fixtures 2022-2023 Premier League Share & Export ▾ Glossary

Wk	Day	Date	Time	Home	xG	Score	xG	Away	Attendance	Venue	Referee	Match Report	Notes
1	Fri	2022-08-05	20:00 (15:00)	Crystal Palace	1.2	0–2	1.0	**Arsenal**	25,286	Selhurst Park	Anthony Taylo	Match Report	
	Sat	2022-08-06	12:30 (07:30)	Fulham	1.2	2–2	1.2	Liverpool	22,207	Craven Cottage	Andy Madley	Match Report	
			15:00 (10:00)	**Tottenham**	1.5	4–1	0.5	Southampton	61,732	Tottenham Hotspur Stadium	Andre Marrine	Match Report	
			15:00 (10:00)	**Newcastle Utd**	1.7	2–0	0.3	Nott'ham Forest	52,245	St. James' Park	Simon Hooper	Match Report	
			15:00 (10:00)	**Leeds United**	0.8	2–1	1.3	Wolves	36,347	Elland Road	Robert Jones	Match Report	
			15:00 (10:00)	**Bournemouth**	0.6	2–0	0.7	Aston Villa	11,013	Vitality Stadium	Peter Bankes	Match Report	
			17:30 (12:30)	Everton	0.7	0–1	1.5	**Chelsea**	39,254	Goodison Park	Craig Pawson	Match Report	
	Sun	2022-08-07	14:00 (09:00)	Leicester City	0.6	2–2	0.8	Brentford	31,794	King Power Stadium	Jarred Gillett	Match Report	
			14:00 (09:00)	Manchester Utd	1.4	1–2	1.5	**Brighton**	73,711	Old Trafford	Paul Tierney	Match Report	
			16:30 (11:30)	West Ham	0.5	0–2	2.2	**Manchester City**	62,443	London Stadium	Michael Oliver	Match Report	
2	Sat	2022-08-13	12:30 (07:30)	**Aston Villa**	2.3	2–1	1.6	Everton	41,883	Villa Park	Michael Oliver	Match Report	
			15:00 (10:00)	**Manchester City**	1.7	4–0	0.1	Bournemouth	53,453	Etihad Stadium	David Coote	Match Report	
			15:00 (10:00)	Southampton	1.2	2–2	1.8	Leeds United	30,815	St. Mary's Stadium	Tony Harringt	Match Report	
			15:00 (10:00)	Wolves	0.9	0–0	1.5	Fulham	31,178	Molineux Stadium	John Brooks	Match Report	
			15:00 (10:00)	**Arsenal**	2.7	4–2	0.5	Leicester City	60,033	Emirates Stadium	Darren Englar	Match Report	
			15:00 (10:00)	Brighton	1.5	0–0	0.2	Newcastle Utd	31,552	The American Express Community Stadium	Graham Scott	Match Report	
			17:30 (12:30)	**Brentford**	1.6	4–0	0.9	Manchester Utd	17,051	Brentford Community Stadium	Stuart Attwell	Match Report	
	Sun	2022-08-14	14:00 (09:00)	**Nott'ham Forest**	1.3	1–0	2.1	West Ham	29,281	The City Ground	Robert Jones	Match Report	
			16:30 (11:30)	Chelsea	1.6	2–2	1.0	Tottenham	39,946	Stamford Bridge	Anthony Taylo	Match Report	
	Mon	2022-08-15	20:00 (15:00)	Liverpool	2.0	1–1	1.0	Crystal Palace	52,970	Anfield	Paul Tierney	Match Report	
3	Sat	2022-08-20	12:30 (07:30)	**Tottenham**	1.7	1–0	0.7	Wolves	61,298	Tottenham Hotspur Stadium	Simon Hooper	Match Report	

```python
#CREATING LIST OF MATCH REPORT LINKS
#-------------------------------------------------------------------------------#
URL = (r"https://fbref.com/en/comps/9/schedule/Premier-League-Scores-and-Fixtures") #this is the website with match results
req = requests.get(URL) #requesting
print("Connection status:",req.status_code) #Checking Connection
soup = bs4.BeautifulSoup(req.text, 'html.parser') #Parsing HTML

tablesoup = soup.select('table.stats_table')[0] #the table with the link is named stats_table + table class
matchsoup = tablesoup.find_all('a') #all the links are under "<a"
matchlink = [] #empty list to add links to
for i in matchsoup:
    if i.text == "Match Report": #we only want links that are described as Match Report
        matchlink.append("http://www.fbref.com"+i.get("href"))
print("Match Link List Created")
```

Step 2: Get home/away team name

1: Create request object for the match report link

2: Make a soup object

3: Search for all H2 Text

4: if H2 text contains “Player Stats” append it to an empty list

5: list[0] is home, list[1] is away

Crystal Palace Player Stats Share & Export ▾ Glossary

Summary | Passing | Pass Types | Defensive Actions | Possession | Miscellaneous Stats

						Performance												Expected			SCA		Passes				Dribbles	
Player	#	Nation	Pos	Age	Min	Gls	Ast	PK	PKatt	Sh	SoT	CrdY	CrdR	Touches	Tkl	Int	Blocks	xG	npxG	xAG	SCA	GCA	Cmp	Att	Cmp%	Prog	Succ	Att
Odsonne Édouard	22	FRA	FW	24-201	57	0	0	0	0	3	1	0	0	20	0	0	0	0.2	0.2	0.0	1	0	3	6	50.0	0	1	2
Jean-Philippe Mateta	14	FRA	FW	25-038	33	0	0	0	0	1	0	0	0	13	0	0	0	0.1	0.1	0.2	0	0	8	10	80.0	0	0	0
Wilfried Zaha	11	CIV	LW	29-268	90	0	0	0	0	1	0	0	0	51	3	0	0	0.1	0.1	0.6	3	0	30	39	76.9	2	3	5
Jordan Ayew	9	GHA	RW,AM	30-328	90	0	0	0	0	1	0	0	0	44	2	0	0	0.1	0.1	0.1	7	0	27	34	79.4	0	6	9
Eberechi Eze	10	ENG	AM	24-037	85	0	0	0	0	1	1	0	0	53	1	0	1	0.5	0.5	0.0	1	0	32	41	78.0	0	1	4
Malcolm Ebiowei	23	ENG	RW	18-335	5	0	0	0	0	0	0	0	0	5	0	0	0	0.0	0.0	0.0	0	0	2	3	66.7	0	1	3
Jeffrey Schlupp	15	GHA	DM	29-225	85	0	0	0	0	3	0	0	0	52	4	1	2	0.2	0.2	0.0	1	0	30	35	85.7	1	3	4
Will Hughes	19	ENG	DM	27-110	5	0	0	0	0	0	0	0	0	4	0	0	0	0.0	0.0	0.0	0	0	3	5	60.0	0	0	0
Cheick Doucouré	28	MLI	DM	22-209	74	0	0	0	0	0	0	0	0	39	1	1	1	0.0	0.0	0.0	1	0	32	34	94.1	4	0	0
Luka Milivojević	4	SRB	DM	31-120	16	0	0	0	0	0	0	0	0	17	0	0	0	0.0	0.0	0.0	0	0	15	17	88.2	3	0	0
Tyrick Mitchell	3	ENG	LB	22-338	90	0	0	0	0	0	0	0	0	75	1	2	2	0.0	0.0	0.0	0	0	50	66	75.8	4	0	1
Marc Guéhi	6	ENG	CB	22-023	90	0	0	0	0	0	0	0	0	114	0	1	4	0.0	0.0	0.0	0	0	93	105	88.6	6	0	0
Joachim Andersen	16	DEN	CB	26-066	90	0	0	0	0	0	0	0	0	123	6	2	0	0.0	0.0	0.1	2	0	93	107	86.9	2	0	0
Nathaniel Clyne	17	ENG	RB	31-122	90	0	0	0	0	0	0	1	0	73	0	1	0	0.0	0.0	0.1	2	0	59	68	86.8	1	1	1
Vicente Guaita	13	ESP	GK	35-207	90	0	0	0	0	0	0	0	0	40	0	0	0	0.0	0.0	0.0	0	0	37	39	94.9	0	0	0
15 Players					**990**	**0**	**0**	**0**	**0**	**10**	**2**	**1**	**0**	**723**	**18**	**8**	**10**	**1.2**	**1.2**	**1.1**	**18**	**0**	**514**	**609**	**84.4**	**23**	**16**	**29**

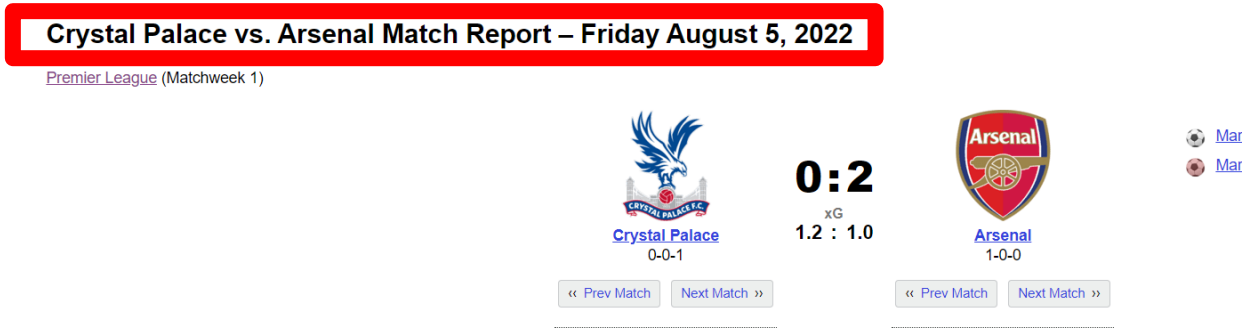
```python
def dataExtract(link):
    matchurl = (link)  # this is the website with match results
    matchreq = requests.get(matchurl)  # requesting
    print("Starting new connection...")
    print("Connection status:", matchreq.status_code)  # Checking Connection

    #Really dumb way of getting home team and away team name + Date
    soup = bs4.BeautifulSoup(matchreq.text, 'html.parser')
    teamnamefinder = soup.select("h2")
    list1 = []
    for i in teamnamefinder:
        if "Player" in i.text:
            list1.append(i.text)
        else:
            pass
    TeamNames = list(set(list1))
    hometeam = TeamNames[0]
    awayteam = TeamNames[1]
```

Step 3: Get match name

1: use the soup object created for step 2

2: the only H1 text is the name of the game




```
gamenamefinder = soup.select("h1")
list2 = []
for i in gamenamefinder:
    gamename = i.text

print(f"Start Scrapping the game {gamename}") #shows what game we are scrapping
```

Step 4: Get Home Team Stats

1: use the soup object created for step 2

2: player stat table is actually
6 different tables

3: Create data frame of all 6 tables

4: concatenate all 6 tables

5: Create data frame of goalkeeper stat table

6: Save result as home team stat data frame

Crystal Palace Player Stats Share & Export ▾ Glossary

Summary | Passing | Pass Types | Defensive Actions | Possession | Miscellaneous Stats

						Performance												Expected			SCA		Passes				Dribbles	
Player	#	Nation	Pos	Age	Min	Gls	Ast	PK	PKatt	Sh	SoT	CrdY	CrdR	Touches	Tkl	Int	Blocks	xG	npxG	xAG	SCA	GCA	Cmp	Att	Cmp%	Prog	Succ	Att
Odsonne Édouard	22	FRA	FW	24-201	57	0	0	0	0	3	1	0	0	20	0	0	0	0.2	0.2	0.0	1	0	3	6	50.0	0	1	2
Jean-Philippe Mateta	14	FRA	FW	25-038	33	0	0	0	0	1	0	0	0	13	0	0	0	0.1	0.1	0.2	0	0	8	10	80.0	0	0	0
Wilfried Zaha	11	CIV	LW	29-268	90	0	0	0	0	1	0	0	0	51	3	0	0	0.1	0.1	0.6	3	0	30	39	76.9	2	3	5
Jordan Ayew	9	GHA	RW,AM	30-328	90	0	0	0	0	1	0	0	0	44	2	0	0	0.1	0.1	0.1	7	0	27	34	79.4	0	6	9
Eberechi Eze	10	ENG	AM	24-037	85	0	0	0	0	1	1	0	0	53	1	0	1	0.5	0.5	0.0	1	0	32	41	78.0	0	1	4
Malcolm Ebiowei	23	ENG	RW	18-335	5	0	0	0	0	0	0	0	0	5	0	0	0	0.0	0.0	0.0	0	0	2	3	66.7	0	1	3
Jeffrey Schlupp	15	GHA	DM	29-225	85	0	0	0	0	3	0	0	0	52	4	1	2	0.2	0.2	0.0	1	0	30	35	85.7	1	3	4

Crystal Palace Goalkeeper Stats Share & Export ▾ Glossary

				Shot Stopping					Launched			Passes				Goal Kicks			Crosses			Sweeper	
Player	Nation	Age	Min	SoTA	GA	Saves	Save%	PSxG	Cmp	Att	Cmp%	Att	Thr	Launch%	AvgLen	Att	Launch%	AvgLen	Opp	Stp	Stp%	#OPA	AvgDist
Vicente Guaita	ESP	35-207	90	2	2	1	0.0	0.3	7	9	77.8	35	2	22.9	25.7	4	25.0	29.5	9	1	11.1	2	15.4

```python
#scrapping home team player stats
#not using for loop for future reference
#stat is divided in 7 different tables 6 for field players 1 for goalkeeper
homefieldstat1 = pd.read_html(matchreq.text,match='Stats Table')[0]
homefieldstat2 = pd.read_html(matchreq.text, match='Stats Table')[1]
homefieldstat3 = pd.read_html(matchreq.text, match='Stats Table')[2]
homefieldstat4 = pd.read_html(matchreq.text, match='Stats Table')[3]
homefieldstat5 = pd.read_html(matchreq.text, match='Stats Table')[4]
homefieldstat6 = pd.read_html(matchreq.text, match='Stats Table')[5]

homegoalkeeperstat = pd.read_html(matchreq.text, match='Stats Table')[6]

homefieldstat = pd.concat([homefieldstat1,homefieldstat2.iloc[:,6:],homefieldstat3.iloc[:,6:],homefieldstat4.iloc[:,6:],
                           homefieldstat5.iloc[:,6:],homefieldstat6.iloc[:,6:]],axis=1) #combining dataframes into one dataframe

homegoalkeeperstat.columns = homegoalkeeperstat.columns.map('|'.join).str.strip('|')  # flattening the multiindex column
homegoalkeeperstat["Home|Away"] = "Home" #adding home away
homegoalkeeperstat["Team"] = hometeam  # adding team name

homefieldstat.columns = homefieldstat.columns.map('|'.join).str.strip('|') #flattening the multiindex column
homefieldstat["Home|Away"] = "Home" #adding home away
homefieldstat["Team"] = hometeam  # adding team name
```

Step 5: Get Away Team Stats

1: Repeat step 4 but for the away team

Arsenal Player Stats Share & Export ▾ Glossary

Summary | Passing | Pass Types | Defensive Actions | Possession | Miscellaneous Stats

						Performance												Expected			SCA		Passes				Dribbles	
Player	#	Nation	Pos	Age	Min	Gls	Ast	PK	PKatt	Sh	SoT	CrdY	CrdR	Touches	Tkl	Int	Blocks	xG	npxG	xAG	SCA	GCA	Cmp	Att	Cmp%	Prog	Succ	Att
Gabriel Jesus	9	BRA	FW	25-124	82	0	0	0	0	1	0	0	0	40	1	1	0	0.1	0.1	0.0	4	0	20	28	71.4	3	6	6
Eddie Nketiah	14	ENG	FW	23-067	8	0	0	0	0	0	0	0	0	4	0	0	0	0.0	0.0	0.1	2	0	3	3	100.0	1	1	1
Martinelli	11	BRA	LW	21-048	90	1	0	0	0	2	1	0	0	35	0	0	1	0.4	0.4	0.1	1	0	24	28	85.7	1	3	3
Bukayo Saka	7	ENG	RW	20-334	90	0	0	0	0	3	0	0	0	48	2	0	3	0.3	0.3	0.0	3	1	25	35	71.4	1	1	3

Arsenal Goalkeeper Stats Share & Export ▾ Glossary

				Shot Stopping					Launched			Passes				Goal Kicks			Crosses			Sweeper	
Player	Nation	Age	Min	SoTA	GA	Saves	Save%	PSxG	Cmp	Att	Cmp%	Att	Thr	Launch%	AvgLen	Att	Launch%	AvgLen	Opp	Stp	Stp%	#OPA	AvgDist
Aaron Ramsdale	ENG	24-083	90	2	0	2	100.0	0.3	8	15	53.3	30	4	46.7	36.5	2	50.0	43.5	16	2	12.5	1	15.7

```
#scrapping away team player stats
awayfieldstat1 = pd.read_html(matchreq.text,match='Stats Table')[7]
awayfieldstat2 = pd.read_html(matchreq.text, match='Stats Table')[8]
awayfieldstat3 = pd.read_html(matchreq.text, match='Stats Table')[9]
awayfieldstat4 = pd.read_html(matchreq.text, match='Stats Table')[10]
awayfieldstat5 = pd.read_html(matchreq.text, match='Stats Table')[11]
awayfieldstat6 = pd.read_html(matchreq.text, match='Stats Table')[12]

awaygoalkeeperstat = pd.read_html(matchreq.text, match='Stats Table')[13]

awayfieldstat = pd.concat([awayfieldstat1,awayfieldstat2.iloc[:,6:],awayfieldstat3.iloc[:,6:],awayfieldstat4.iloc[:,6:],
                           awayfieldstat5.iloc[:,6:],awayfieldstat6.iloc[:,6:]],axis=1) #combining dataframes into one dataframe

awaygoalkeeperstat.columns = awaygoalkeeperstat.columns.map('|'.join).str.strip('|')  # flattening the multiindex column
awaygoalkeeperstat["Home|Away"] = "Away"  # adding home away
awaygoalkeeperstat["Team"] = awayteam  # adding team name
awayfieldstat.columns = awayfieldstat.columns.map('|'.join).str.strip('|') #flattening the multiindex column
awayfieldstat["Home|Away"] = "Away"  # adding home away
awayfieldstat["Team"] = awayteam  # adding team name
```

Step 6: Combine Data and export as csv

1: Combine home and away data

2: Export the data frame as a csv file to a folder of your choice

```
#Combining Home and Away dataframe into one
fieldstat = pd.concat([homefieldstat.iloc[:-1],awayfieldstat.iloc[:-1]],axis = 0) #we don't need the last row of each dataframe
fieldstat["Game"] = gamename #Adding date
goalkeeperstat = pd.concat([homegoalkeeperstat,awaygoalkeeperstat],axis = 0)
goalkeeperstat["Game"] = gamename #Adding date
print("Dataframe is ready...Exporting as csv") #dataframe ready signal

space="_"
fieldstat.to_csv(rf"C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\Fieldstat\fieldstate_{gamename}.csv"
                 ,encoding='utf-8-sig') #ExportingData
goalkeeperstat.to_csv(rf"C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\Goalkeeperstat\goalkeeperstat_{gamename}.csv"
                      ,encoding='utf-8-sig') # ExportingData
print("Export completed")
print("------------------------------------------------------------------------------------------------------------------------------------------------------") #end of scrap
```

Step 7: Repeat using for loop

1: Use for loop to repeat process for all match report links

2: After one cycle is completed wait 7 to 12 seconds to avoid macro detection in the website

3: If done correctly, you will have many csv files in the export folder

```
for i in matchlink:
    dataExtract(i)
    print("Sleeping for a given time...")
    time.sleep(random.randint(7,13)) #this is avoid security detection
    print("I am awake!")

print("End process")
```

fieldstate_Arsenal vs. Aston Villa Match Report – Wednesday August 31, 2022.csv
fieldstate_Arsenal vs. Fulham Match Report – Saturday August 27, 2022.csv
fieldstate_Arsenal vs. Leicester City Match Report – Saturday August 13, 2022.csv
fieldstate_Arsenal vs. Liverpool Match Report – Sunday October 9, 2022.csv
fieldstate_Arsenal vs. Newcastle United Match Report – Tuesday January 3, 2023.csv
fieldstate_Arsenal vs. Nottingham Forest Match Report – Sunday October 30, 2022...
fieldstate_Arsenal vs. Tottenham Hotspur Match Report – Saturday October 1, 202...
fieldstate_Arsenal vs. West Ham United Match Report – Monday December 26, 20...
fieldstate_Aston Villa vs. Brentford Match Report – Sunday October 23, 2022.csv
fieldstate_Aston Villa vs. Chelsea Match Report – Sunday October 16, 2022.csv
fieldstate_Aston Villa vs. Everton Match Report – Saturday August 13, 2022.csv
fieldstate_Aston Villa vs. Liverpool Match Report – Monday December 26, 2022.csv
fieldstate_Aston Villa vs. Manchester City Match Report – Saturday September 3, ...
fieldstate_Aston Villa vs. Manchester United Match Report – Sunday November 6,...
fieldstate_Aston Villa vs. Southampton Match Report – Friday September 16, 2022...
fieldstate_Aston Villa vs. West Ham United Match Report – Sunday August 28, 20...
fieldstate_Aston Villa vs. Wolverhampton Wanderers Match Report – Wednesday J...
fieldstate_Bournemouth vs. Arsenal Match Report – Saturday August 20, 2022.csv
fieldstate_Bournemouth vs. Aston Villa Match Report – Saturday August 6, 2022.csv
fieldstate_Bournemouth vs. Brentford Match Report – Saturday October 1, 2022.csv
fieldstate_Bournemouth vs. Crystal Palace Match Report – Saturday December 31,...
fieldstate_Bournemouth vs. Everton Match Report – Saturday November 12, 2022....
fieldstate_Bournemouth vs. Leicester City Match Report – Saturday October 8, 202...
fieldstate_Bournemouth vs. Southampton Match Report – Wednesday October 19...
fieldstate_Bournemouth vs. Tottenham Hotspur Match Report – Saturday October...
fieldstate_Bournemouth vs. Wolverhampton Wanderers Match Report – Wednesd...
fieldstate_Brentford vs. Arsenal Match Report – Sunday September 18, 2022.csv
fieldstate_Brentford vs. Brighton & Hove Albion Match Report – Friday October 1...
fieldstate_Brentford vs. Chelsea Match Report – Wednesday October 19, 2022.csv
fieldstate_Brentford vs. Everton Match Report – Saturday August 27, 2022.csv
fieldstate_Brentford vs. Leeds United Match Report – Saturday September 3, 2022....
fieldstate_Brentford vs. Liverpool Match Report – Monday January 2, 2023.csv

Step 8: Get path of all CSV files

1: use glob and os to create list of all csv file path that we want to combined

```
Fieldstat = (r'C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\Fieldstat') #Input source data folder
Fieldstatcsv = glob.glob(os.path.join(Fieldstat, "*.csv")) #This gets all the csv file in the source data folder

Goalkeeperstat = (r'C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\Goalkeeperstat') #Input source data folder
Goalkeeperstatcsv = glob.glob(os.path.join(Goalkeeperstat, "*.csv")) #This gets all the csv file in the source data folder
```

Step 9: Combine all CSV file into 1 data frame

1: Create a base data frame
2: Append every other csv file into that base data frame
3: Export it as the main csv file
4: Repeat for goalkeeper csv files too

```
#Combining FieldStat CSV Files
#-----------------------------------------------------------------------------------#
df1 = pd.read_csv(Fieldstatcsv[0]) #Get a base csv
n = 0
for i in Fieldstatcsv:
    if n==0:
        pass
    else:
        df2 = pd.read_csv(i)
        df1 = pd.concat([df1,df2]) #combine base df to df we are currently reading
    n+=1
    print(f"Combining{n}th csv file")
print("Combining FieldStat CSV Completed")
df1.to_csv(r'C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\CombinedData\fielddata.csv',encoding='utf-8-sig') #export combined df
```

Step 10: Connect to MySQL database

1: Use mysql.connector to connect to your MySQL database

```
mydb = mysql.connector.connect(  # this is the credential to connect into mysql database
    host="########", user="#########", password="##########" #input your login details
)

connector = mydb.cursor()  # this is using the credential to create a connector
```

Step 11: Get combined CSV file

1: Truncate table in the MySQL database (to avoid duplicate values)
2: Change all nan value into null

```
deletequery = "truncate footballdatasets.field_data"  # this is query to reset the database
connector.execute(deletequery)  # I am executing the query

empdata = pd.read_csv(
    r"C:\Users\justi\OneDrive\Desktop\ScrapAndPipeline\Data\CombinedData\fielddata.csv", #read the csv you want to upload
    index_col=False,
    delimiter=",",
)
empdata.fillna("null",inplace=True) #change all NA value into null
```

Step 12: Get combined CSV file

1: Create string of SQL Query
2: Use for loop to create string variables (numbers of the columns on csv file)
3: Use for loop to insert each row into the MySQL database

```
for i in range(116): #create string to insert into sql query 116 is the number of columns
    text = text + "%s,"
text = text[:-1]

sql = f"INSERT INTO footballdatasets.field_data VALUES " \
      f"({text})"  # this is the query to insert data

for i,row,in empdata.iterrows():  # iterrows() and tuple(row) allows me to insert data row by row
    connector.execute(sql, tuple(row[1:]))  # row[1:] to not include the index
    # print(tuple(row[1:2]))
mydb.commit()  # commit the changes
```