**MATCH RESULTS TRACKER:**

Import streamlit as st

Import json

Import os

Import matplotlib.pyplot as plt

Import pandas as pd

# Load existing match data from a JSON file

Def load\_data(filename):

If os.path.exists(filename):

With open(filename, “r”) as file:

Return json.load(file)

Else:

Return {}

# Save match data to a JSON file

Def save\_data(data, filename):

With open(filename, “w”) as file:

Json.dump(data, file)

# Function to add a match

Def add\_match():

Team1 = st.text\_input(“Team 1”)

Team2 = st.text\_input(“Team 2”)

Score1 = st.number\_input(“Score 1”, min\_value=0, step=1)

Score2 = st.number\_input(“Score 2”, min\_value=0, step=1)

Man\_of\_match = st.text\_input(“Man of the Match”)

Highest\_wicket\_taker = st.text\_input(“Highest Wicket Taker”)

If st.button(“Add Match”):

Match\_data[f”{team1} vs {team2}”] = {

“Team1”: team1,

“Team2”: team2,

“Score1”: score1,

“Score2”: score2,

“Man of the Match”: man\_of\_match,

“Highest Wicket Taker”: highest\_wicket\_taker

}

Save\_data(match\_data, “match\_data.json”)

St.success(“Match added successfully!”)

# Function to display match results in a table

Def display\_results():

St.header(“Match Results”)

Df = pd.DataFrame.from\_dict(match\_data, orient=’index’)

St.write(df)

# Function to display team performance graph

Def team\_performance\_graph():

St.header(“Team Performance Graph”)

Team\_scores = {} # Dictionary to store total scores for each team

# Aggregate scores for each team

For match, details in match\_data.items():

Team1 = details[‘Team1’]

Team2 = details[‘Team2’]

Score1 = int(details[‘Score1’])

Score2 = int(details[‘Score2’])

If team1 not in team\_scores:

Team\_scores[team1] = 0

If team2 not in team\_scores:

Team\_scores[team2] = 0

Team\_scores[team1] += score1

Team\_scores[team2] += score2

# Convert dictionary to DataFrame

Team\_df = pd.DataFrame.from\_dict(team\_scores, orient=’index’, columns=[‘Total Runs’])

Team\_df.reset\_index(inplace=True)

Team\_df.rename(columns={‘index’: ‘Team’}, inplace=True)

# Plot team performance graph

St.bar\_chart(team\_df.set\_index(‘Team’))

# Function to update match data

Def update\_match():

Match\_to\_update = st.selectbox(“Select match to update:”, list(match\_data.keys()))

If match\_to\_update:

Team1 = st.text\_input(“Team 1 – Update”, match\_data[match\_to\_update][“Team1”])

Team2 = st.text\_input(“Team 2 – Update”, match\_data[match\_to\_update][“Team2”])

Score1 = st.number\_input(“Score 1 – Update”, min\_value=0, step=1, value=match\_data[match\_to\_update][“Score1”])

Score2 = st.number\_input(“Score 2 – Update”, min\_value=0, step=1, value=match\_data[match\_to\_update][“Score2”])

Man\_of\_match = st.text\_input(“Man of the Match – Update”, match\_data[match\_to\_update].get(“Man of the Match”, “”))

Highest\_wicket\_taker = st.text\_input(“Highest Wicket Taker – Update”, match\_data[match\_to\_update].get(“Highest Wicket Taker”, “”))

If st.button(“Update Match”):

Match\_data[match\_to\_update] = {

“Team1”: team1,

“Team2”: team2,

“Score1”: score1,

“Score2”: score2,

“Man of the Match”: man\_of\_match,

“Highest Wicket Taker”: highest\_wicket\_taker

}

Save\_data(match\_data, “match\_data.json”)

St.success(“Match updated successfully!”)

# Function to search for a match

Def search\_match():

Search\_query = st.text\_input(“Search for a match”)

If st.button(“Search”):

Found = False

For match, details in match\_data.items():

If search\_query.lower() in match.lower():

St.subheader(“Match Found:”)

St.write(f” {details[‘Team1’]} {details[‘Score1’]} – {details[‘Score2’]} {details[‘Team2’]}”)

St.write(f” Man of the Match: {details.get(‘Man of the Match’, ‘N/A’)}”)

St.write(f” Highest Wicket Taker: {details.get(‘Highest Wicket Taker’, ‘N/A’)}”)

Found = True

If not found:

St.warning(“Match not found.”)

# Function to delete a particular match data

Def delete\_match():

Match\_to\_delete = st.selectbox(“Select match to delete:”, list(match\_data.keys()))

If match\_to\_delete:

If st.button(“Delete Match”):

Del match\_data[match\_to\_delete]

Save\_data(match\_data, “match\_data.json”)

St.success(“Match deleted successfully!”)

# Main Streamlit UI

St.title(“Match Results Tracker”)

# Load existing match data from a JSON file

Match\_data = load\_data(“match\_data.json”)

# Create a menu to navigate options

Menu\_option = st.sidebar.radio(“Select Option”, [“Add”, “Display”, “Search”, “Update”, “Delete”, “Team Performance Graph”])

# Based on the selected option, call the corresponding function

If menu\_option == “Add”:

Add\_match()

Elif menu\_option == “Display”:

Display\_results()

Elif menu\_option == “Search”:

Search\_match()

Elif menu\_option == “Update”:

Update\_match()

Elif menu\_option == “Delete”:

Delete\_match()

Elif menu\_option == “Team Performance Graph”:

Team\_performance\_graph()