BASAVARAJ ESWARI GROUP OF INSTITUTIONS



# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

**AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELAGAVI 590018**

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**Intership Report on**

# “Referee Scheduling System”

# For the course: Python

**Name Usn no**

**Guru Kiran 3BR23EC048**

**Mohammad Ithesham 3BR23EC099**

**Shanmuka 3BR23EC151**

**Siddeshwar Pujeri 3BR23EC155**

**Manish 3BR23EC171**

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited institution\*

(Recognised by Govt.of karnataka,approved by AICTE,NewDelhi & Affiliated to Visvesvaraya

Technological University, Belgavi)

“Jnana Gangotri” Campus,No.873/2,Ballari-Hospet Road,Allipur,Ballari-583104

Karnataka,India.

Ph: 08392-237100/23719, Fax:08392-237197

2023-2024

**Introduction:**

The Referee Schedule System is designed to automate the process of assigning referees to sporting events based on criteria such as availability, location, and level of expertise. This system aims to streamline scheduling, reduce human errors, and improve efficiency in sports event management. Traditional methods of scheduling referees are time-consuming and prone to mismanagement. The automated system can save valuable time for administrators and ensure fair distribution of assignments among referees.

**Objective:**

The primary objectives of the Referee Schedule System are:

* To automate the scheduling process for referees.
* To ensure fair and balanced assignments based on predefined criteria (e.g., availability, location, qualifications).
* To provide an easy-to-use interface for both referees and administrators.
* To reduce manual errors and conflicts in scheduling.
* To improve efficiency and transparency in the assignment process.

**System Requirements:**

a. Software Requirements

* **Operating System**: Windows 10, Linux, or macOS.
* **Programming Language**: Python (or Java, C# depending on choice).
* **Database**: MySQL, SQLite, or PostgreSQL for managing user and schedule data.
* **Web Framework (optional):** Django (Python), Flask (Python), or ASP.NET (C#) for web-based interfaces.
* **IDE:** Visual Studio Code

b. Hardware Requirements

* **Processor**: Dual Core or higher.
* **RAM:** Minimum 4GB (8GB recommended).
* **Storage:** Minimum 500MB for the application (depending on data size).

**Algorithm:**

1**. Input:**

* Referee Data: Name, availability, location, qualifications.
* Event Data: Event type, location, date, time, required expertise.

2. **Process:**

* **Fetch the Event Details:** The system retrieves the event details that need a referee assignment.
* **Filter Referees Based on Availability:**The system checks which referees are available during the time slot of the event.
* **Match Referees by Location:** The system filters the referees based on proximity to the event location to minimize travel costs.
* **Filter by Expertise:** The system matches the referee’s expertise level to the type of event.
* **Avoid Conflict**: Ensure the referee has no prior assignments that overlap with the new event.

3**. Output:**

* Generate a list of suitable referees for the event.
* Automatically assign the most suitable referee to each event.
* Notify referees and administrators about the assignments.

**Algorithm:**

BEGIN

INPUT event details (time, date, location, expertise level required)

INPUT referee details (availability, location, expertise level)

FOR each event:

FILTER referees by availability

FILTER referees by location proximity

FILTER referees by required expertise level

IF conflict exists (referee is already assigned to another event) THEN

REMOVE referee from the list

END IF

ASSIGN the most suitable referee to the event

NOTIFY the referee and administrator about the assignment

END FOR

END

5. **Code (Python Example)**

Below is a simplified version of the system using Python:

class RefereeSchedule:

def \_init\_(self):

self.schedule = {}

def add\_referee(self):

name = input("Enter referee's name: ")

date = input("Enter match date (YYYY-MM-DD): ")

time = input("Enter match time (HH:MM): ")

team1 = input("Enter team 1 name: ")

team2 = input("Enter team 2 name: ")

self.schedule[name] = {

"date": date,

"time": time,

"team1": team1,

"team2": team2 }

print(f"Referee {name} added successfully.")

def update\_referee(self):

name = input("Enter referee's name to update: ")

if name in self.schedule:

print("Enter new details (press Enter to skip):")

self.schedule[name]["date"] = input(f"Date ({self.schedule[name]['date']}):") or self.schedule[name]["date"]

self.schedule[name]["time"] = input(f"Time ({self.schedule[name]['time']}): ") or self.schedule[name]["time"]

self.schedule[name]["team1"] = input(f"Team 1 ({self.schedule[name]['team1']}): ") or self.schedule[name]["team1"]

self.schedule[name]["team2"] = input(f"Team 2 ({self.schedule[name]['team2']}): ") or self.schedule[name]["team2"]

print(f"Referee {name} updated successfully.")

else:

print(f"Referee {name} not found.")

def delete\_referee(self):

name = input("Enter referee's name to delete: ")

if name in self.schedule:

del self.schedule[name]

print(f"Referee {name} deleted successfully.")

else:

print(f"Referee {name} not found.")

def display\_schedule(self):

print("Referee Schedule:")

for name, details in self.schedule.items():

print(f"Referee: {name}")

print(f"Date: {details['date']}")

print(f"Time: {details['time']}")

print(f"Teams: {details['team1']} vs {details['team2']}")

print("------------------------")

def main():

schedule = RefereeSchedule()

while True:

print("Referee Schedule System")

print("1. Add Referee")

print("2. Update Referee")

print("3. Delete Referee")

print("4. Display Schedule")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

schedule.add\_referee()

elif choice == "2":

schedule.update\_referee()

elif choice == "3":

schedule.delete\_referee()

elif choice == "4":

schedule.display\_schedule()

elif choice == "5":

break

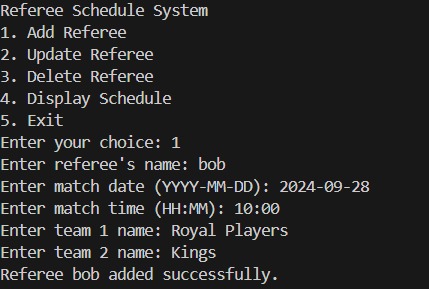
else:

print("Invalid choice. Please try again.")

if \_name\_ == "\_main\_":

main()

**output:**

****

**6. Conclusion:**

The Referee Schedule System is a valuable tool for automating the scheduling process, improving efficiency, and reducing human error in assigning referees to events. By considering availability, location, and expertise, this system ensures that events are managed smoothly and effectively. This solution can be further enhanced by integrating mobile notifications, web-based dashboards, and dynamic assignment rules to accommodate large-scale sporting events.

**Future Enhancement:**

1. AI-Powered Auto-Scheduling

Implement machine learning algorithms to analyze historical data and automatically assign referees based on factors like availability, performance, proximity to the venue, and preferences.

The system can learn from patterns and improve scheduling accuracy over time.

2. Mobile App Integration

Build a mobile app for referees to easily view schedules, accept or decline assignments, get reminders, and communicate with coordinators.

Add push notifications for schedule changes, cancellations, and new assignments.