PLAYER MANAGEMENT SYSTEM

Mini Project Documentation (C Language)

---

1. Introduction

The **Player Management System** is a console-based application developed using the **C programming language**.
The purpose of this project is to manage player records efficiently by performing basic CRUD (Create, Read, Update, Delete) operations along with searching and sorting functionalities.

This system is suitable for managing players in sports teams and demonstrates strong fundamentals of **structures, dynamic memory allocation, functions, pointers, and file-independent data handling**.

---

2. Objectives of the Project

- To store and manage player information efficiently
- To implement CRUD operations using dynamic memory
- To provide search and sort functionalities
- To improve understanding of structures and pointers in C
- To build a real-world console-based management system

---

3. Technologies Used

Component	Description
Programming Language	C
Compiler	Dev C++
Platform	Console-based application
Libraries Used	stdio.h, stdlib.h, string.h

---

4. Features of the System

- Add multiple player records
- Add a single player dynamically
- Display all players
- Display a specific player
- Search player by ID or Name
- Update player details
- Delete player records
- Sort players based on performance
- Dynamic memory management using `malloc` and `realloc`

---

5. Data Structures Used

5.1 Date Structure

Stores date-related information such as:

- Day
- Month
- Year

5.2 Team Structure

Stores team-related details:

- Team ID
- Team Name
- Player Role
- Captain Status
- Active Status

5.3 PerformanceStats Structure

Stores performance-related data:

- Matches Played
- Total Score
- Best Performance Score

5.4 System Structure

Stores system-level information:

- Data upload date
- Last updated date
- Remarks
- Contact value (charges)

5.5 Player Structure

Main structure combining all player-related data:

- Personal details
- Contact information
- Team information
- Performance statistics
- System metadata

6. Functional Modules

6.1 Add Players

Allows the user to add multiple player records at once.

6.2 Add Single Player

Dynamically adds a new player record using memory reallocation.

6.3 Display Players

Displays details of all players stored in the system.

6.4 Display Player by Index

Displays details of a specific player based on search result.

6.5 Search Player

Searches player using:

- Player ID
- Player Name (case-insensitive)

6.6 Update Player

Updates player details after locating the record using search functionality.

6.7 Delete Player

Deletes a player record and shifts remaining records to maintain array continuity.

6.8 Sort Player Data

Sorts player records based on:

- Score
- Matches Played
- Best Performance

Sorting supports:

- Ascending order
- Descending order

7. Memory Management

- Dynamic memory allocation using `malloc`
- Memory resizing using `realloc`
- Safe deletion by shifting array elements
- Efficient memory usage without memory leaks

---

8. Error Handling

- Displays appropriate messages when data is not available
- Prevents invalid updates and deletions
- Handles invalid menu choices gracefully

---

9. Limitations

- Data is stored in memory (no file storage)
- Console-based user interface
- Uses `gets()` which is not recommended for production systems

---

10. Future Enhancements

- File handling for permanent data storage
- Input validation improvements
- Authentication (Admin/User roles)
- GUI-based interface
- Integration with SMS or notification systems

---

11. Conclusion

The **Player Management System** successfully demonstrates the practical application of C programming concepts such as structures, pointers, dynamic memory allocation, and modular programming.

This project reflects a strong understanding of real-world problem solving and can be extended further for advanced use cases.

---

12. Developer Details

Developer Name: Pritish Ramesh Pawar
Course: Java Full Stack
Project Type: Mini Project (C Language)
Year: 2026