ASSIGNMENT-1 MINI PROJECT

Title of The Project: Cricket Player Ranking System

Group No : 16

Name:

Prathweesh(4MW24CS110)

Preethish (4MW24CS111)

Manish(4MWDIP004)

Rohan (4MWDIP006)

Abstract:

1) The Cricket Ranking System is a menu-driven C program used to store and manage cricket player performance details.

2) It uses structures and arrays to record player name, runs, wickets, and calculated ranking score.

3) The system identifies the top batsman, top bowler, and dynamically determines the top ranked player.

4) Additional operations such as display, search, and delete improve data management and usability.

Application Description:

*Used to store and manage cricket player performance details.

*Calculates player ranking based on runs and wickets.

*Helps identify top batsman, top bowler, and top ranked player.

*Useful for performance comparison and player selection.

Algorithm:

1.Menu Display and Input:

*Display a menu with options to Add Player, Show Top Batsman, Show Top Bowler, Show Top Ranked Player, Display All Players, Search Player, Delete Player, and Exit.

*Prompt the user to enter their choice.

2. Action Based on User Choice:

*Use a switch statement to execute the corresponding operation based on the user’s input.

*For 1 (Add Player):

Prompt for player name, runs, and wickets, calculate the ranking score, and store the details.

*For 2 (Show Top Batsman):

Compare the runs of all players and display the player with the highest runs.

*For 3 (Show Top Bowler):

Compare the wickets of all players and display the player with the highest wickets.

*For 4 (Show Top Ranked Player):

Compare the ranking scores of all players and display the player with the highest score.

*For 5 (Display All Players):

Display the details of all stored players.

*For 6 (Search Player):

Search for a player by name and display the details if found.

*For 7 (Delete Player):

Remove the selected player from the records.

*For 8 (Exit):

Terminate the program.

3. Function Implementation:

Each function performs a specific task as follows:

*Add Player: Stores player details and computes ranking score.

*Top Batsman: Identifies the player with maximum runs.

*Top Bowler: Identifies the player with maximum wickets.

*Top Ranked Player: Identifies the player with highest ranking score.

*Display/Search/Delete: Manage and maintain player records.

Methodology:

*The system is developed using the C programming language with a menu-driven approach to ensure ease of use.

*Player details such as name, runs, and wickets are stored using structures and arrays.

*A ranking score is calculated for each player based on both batting and bowling performance.

*Searching and deletion operations are implemented using linear search techniques.

*The system continuously accepts user input until the exit option is selected.

Source Code:

```
 #include <stdio.h>

#include <string.h>

#define MAX 10
```

```
struct Player {
    char name[30];
    int runs, wickets, score;
};
struct Player p[MAX];
int n = 0;
void addPlayer() {
    if (n == MAX) { printf("Limit reached!\n"); return; }
    printf("Name: "); scanf("%s", p[n].name);
    printf("Runs: "); scanf("%d", &p[n].runs);
    printf("Wickets: "); scanf("%d", &p[n].wickets);
    p[n].score = p[n].runs + p[n].wickets * 20;
    n++;
    printf("Player added!\n");
}
void topBatsman() {
    if (n == 0) { printf("No players!\n"); return; }
    int m = 0;
    for (int i = 1; i < n; i++)
        if (p[i].runs > p[m].runs) m = i;
    printf("Top Batsman: %s (%d runs)\n", p[m].name, p[m].runs);
}
void topBowler() {
    if (n == 0) { printf("No players!\n"); return; }
    int m = 0;
    for (int i = 1; i < n; i++)
        if (p[i].wickets > p[m].wickets) m = i;
    printf("Top Bowler: %s (%d wickets)\n", p[m].name, p[m].wickets);
}
void topRanked() {
    if (n == 0) { printf("No players!\n"); return; }
    int m = 0;
    for (int i = 1; i < n; i++)
        if (p[i].score > p[m].score) m = i;
    printf("\n--- Top Ranked Player ---\n");
    printf("Name    : %s\n", p[m].name);
    printf("Runs    : %d\n", p[m].runs);
    printf("Wickets : %d\n", p[m].wickets);
    printf("Score   : %d\n", p[m].score);
}
void displayAll() {
    if (n == 0) { printf("No players!\n"); return; }
    for (int i = 0; i < n; i++)
        printf("%d. %s R:%d W:%d S:%d\n",
               i+1, p[i].name, p[i].runs, p[i].wickets, p[i].score);
}
void searchPlayer() {
    char key[30];
    printf("Enter name: "); scanf("%s", key);
    for (int i = 0; i < n; i++)
        if (strcmp(p[i].name, key) == 0) {
            printf("Found: %s R:%d W:%d S:%d\n",
                   p[i].name, p[i].runs, p[i].wickets, p[i].score);
            return;
        }
    printf("Player not found!\n");
}
void deletePlayer() {
    char key[30];
    printf("Enter name: "); scanf("%s", key);
    for (int i = 0; i < n; i++)
        if (strcmp(p[i].name, key) == 0) {
            for (int j = i; j < n-1; j++) p[j] = p[j+1];
            n--;
            printf("Player deleted!\n");
            return;
        }
    printf("Player not found!\n");
}
int main() {
    int ch;
    while (1) {
        printf("\n--- Cricket Ranking System ---\n");
        printf("1.Add Player\n2.Top Batsman\n3.Top Bowler\n");
        printf("4.Top Ranked Player\n5.Display All\n6.Search\n");
        printf("7.Delete\n8.Exit\n");
        printf("Choice: "); scanf("%d", &ch);
        switch (ch) {
            case 1: addPlayer(); break;
            case 2: topBatsman(); break;
            case 3: topBowler(); break;
            case 4: topRanked(); break;
            case 5: displayAll(); break;
            case 6: searchPlayer(); break;
            case 7: deletePlayer(); break;
            case 8: return 0;
            default: printf("Invalid choice!\n");
        }
    }
}
```

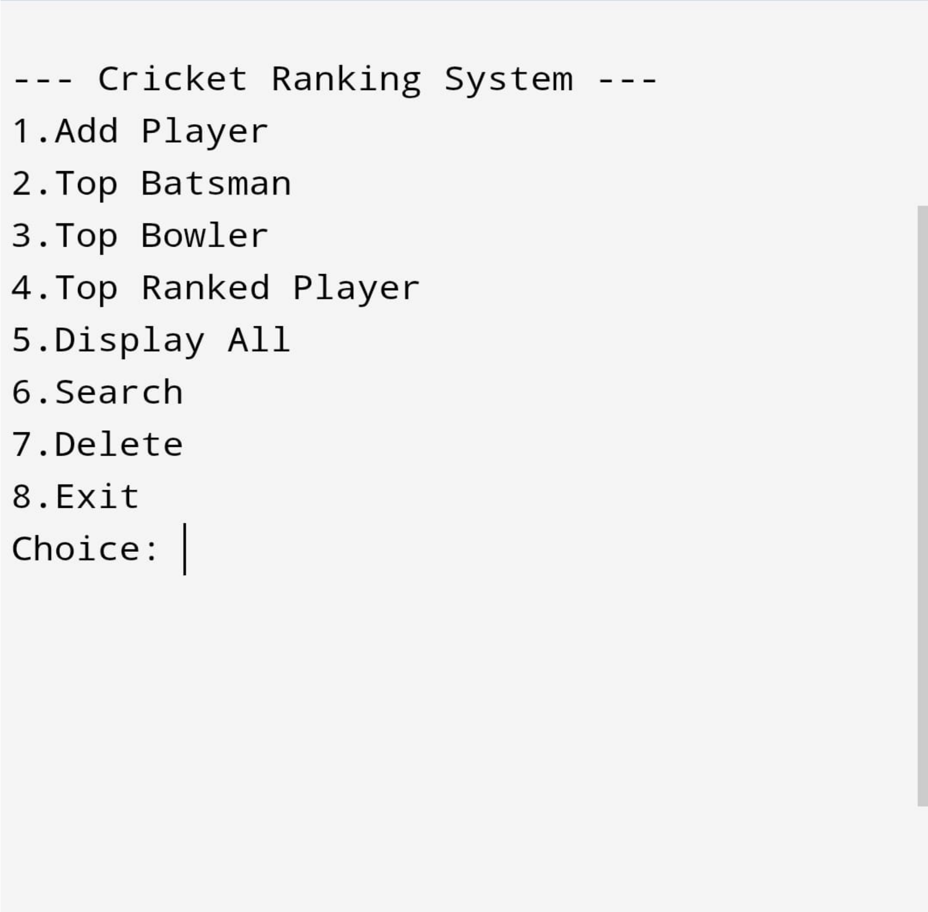
Results and discussion:

1) Main menu:

*The main menu provides options to add player details, calculate scores, and manage rankings.

*It allows users to view top-ranked players based on performance (runs and wickets).

*Users can also display all players and exit the program safely from this menu.



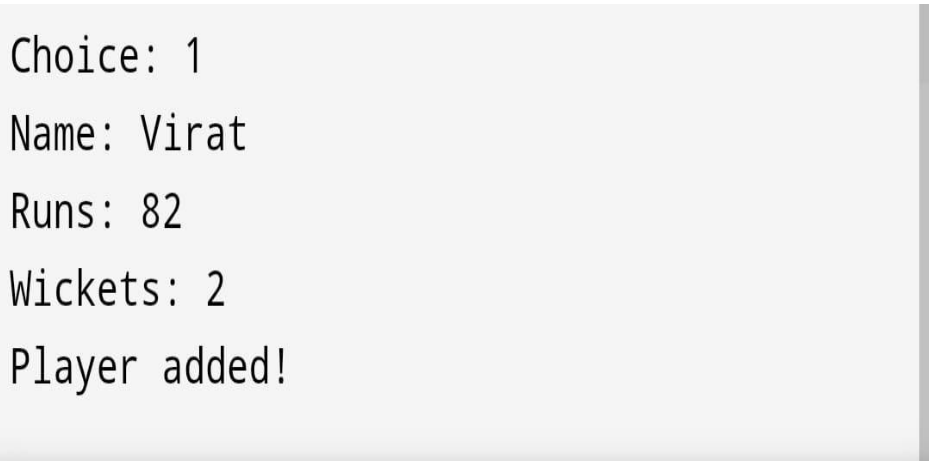
```
--- Cricket Ranking System ---
1.Add Player
2.Top Batsman
3.Top Bowler
4.Top Ranked Player
5.Display All
6.Search
7.Delete
8.Exit
Choice: 
```

1) (i)Adding player:

*Allows the user to enter a player’s name, runs scored, and wickets taken.

*Stores the entered details in the system for ranking.

*Confirms successful addition of the player.



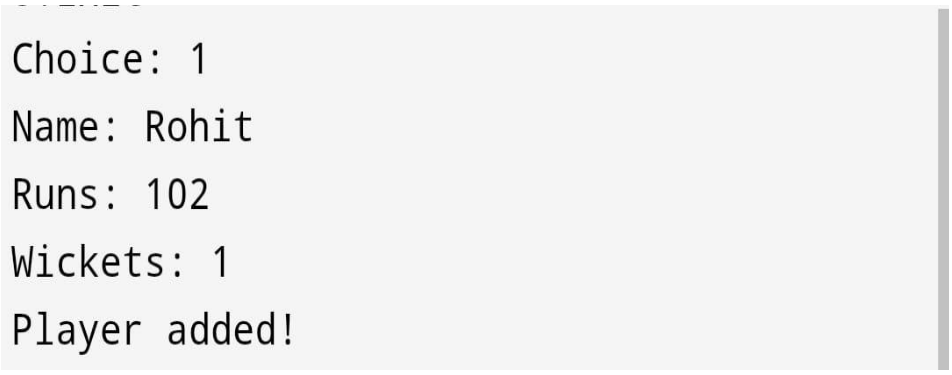
```
Choice: 1
Name: Virat
Runs: 82
Wickets: 2
Player added!
```

1) (ii)Add Player :

*Accepts the player’s name, runs scored, and wickets taken as input.

*Stores Rohit's performance details in the system.

*Displays a confirmation message indicating the player was added successfully



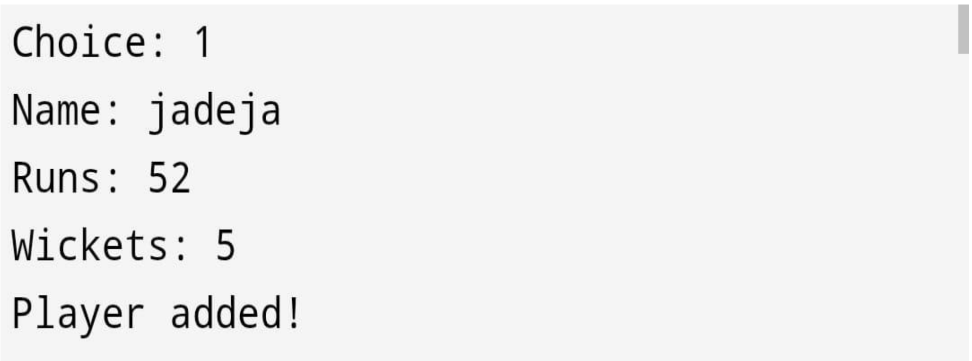
```
Choice: 1
Name: Rohit
Runs: 102
Wickets: 1
Player added!
```

1) (iii) Add player :

*Accepts the player's name, runs scored, and wickets taken as input.

*Stores jadeja's performance details in the system (specifically, 52 runs and 5 wickets).

*Displays a confirmation message indicating the player was added successfully.

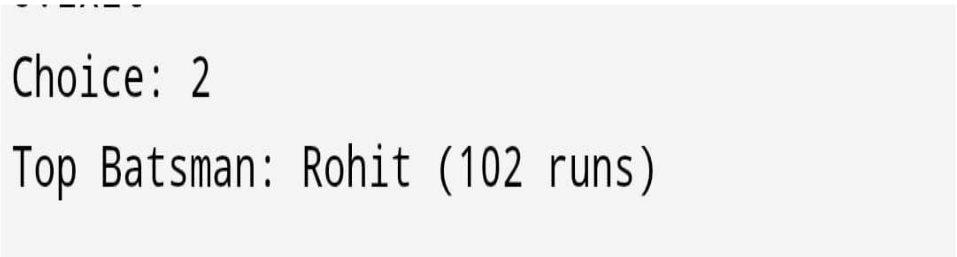


```
Choice: 1
Name: jadeja
Runs: 52
Wickets: 5
Player added!
```

2) Top batsman:

*Choice: A value of '2' has been selected or recorded.

*Top Batsman: The top batsman is identified as Rohit, who scored 102 runs.



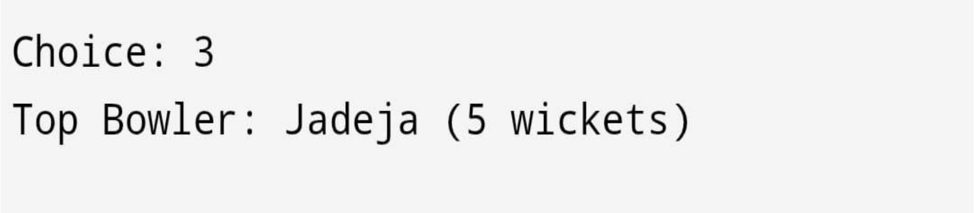
```
Choice: 2
Top Batsman: Rohit (102 runs)
```

3)Top bowler:

*Displays the bowler with the highest number of wickets.

*Shows the player’s name and total wickets taken.

*Helps identify the best-performing bowler in the system.



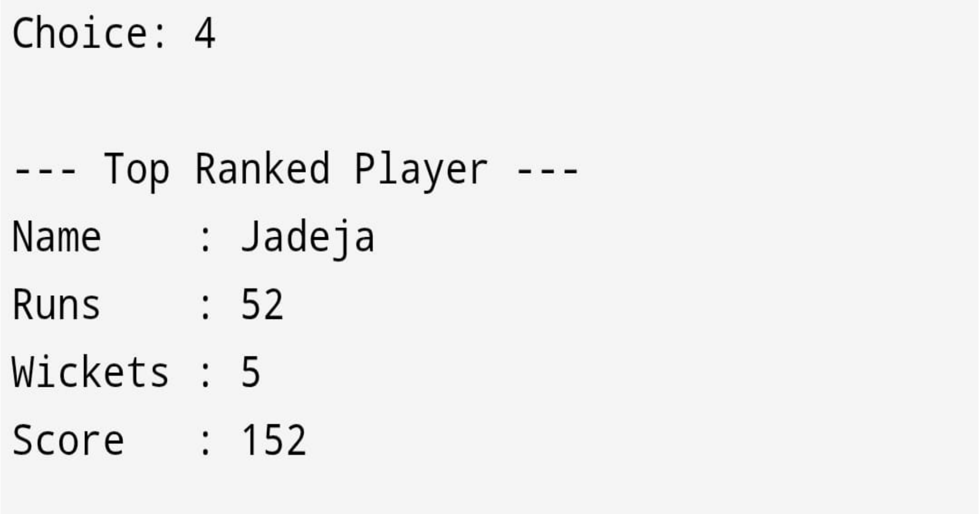
```
Choice: 3
Top Bowler: Jadeja (5 wickets)
```

4) Top ranked player:

*Displays the player with the highest overall performance score.

*Shows detailed information including name, runs, wickets, and total score.

*Helps quickly identify the best all-round performer in the system.



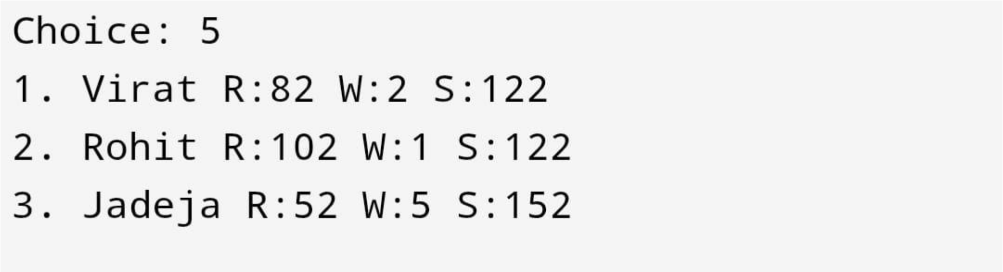
```
Choice: 4

--- Top Ranked Player ---
Name    : Jadeja
Runs    : 52
Wickets : 5
Score   : 152
```

5) Display all:

*All Players Displayed: Virat, Rohit, and Jadeja with their Runs (R), Wickets (W), and Score (S) are listed.

"Score Comparison: Jadeja has the highest score (152), while Virat and Rohit have 122 each.

*Top-Ranked Player: Jadeja ranks first due to his combined performance in batting and bowling.



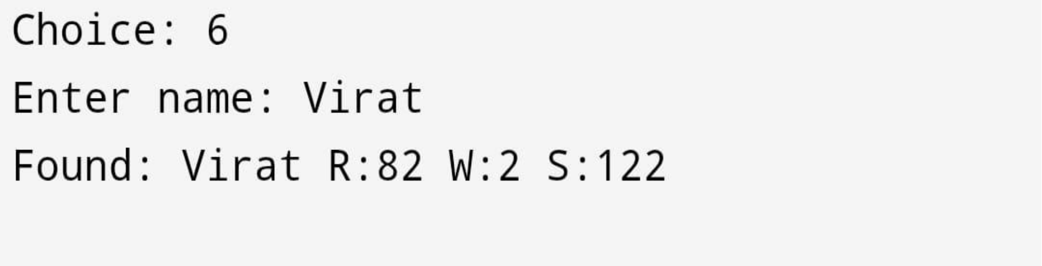
```
Choice: 5
1. Virat R:82 W:2 S:122
2. Rohit R:102 W:1 S:122
3. Jadeja R:52 W:5 S:152
```

6) Search:

*Player Search: The program searches for the player name entered (Virat).

*Player Found: Virat’s stats are displayed – Runs: 82, Wickets: 2, Score: 122.

*Function Purpose: Choice 6 allows searching and displaying a specific player’s details from the list.



```
Choice: 6
Enter name: Virat
Found: Virat R:82 W:2 S:122
```

7) Delete:

*Player Deletion: The program searches for the entered name (Jadeja).

*Action Taken: Jadeja’s record is removed from the player list.

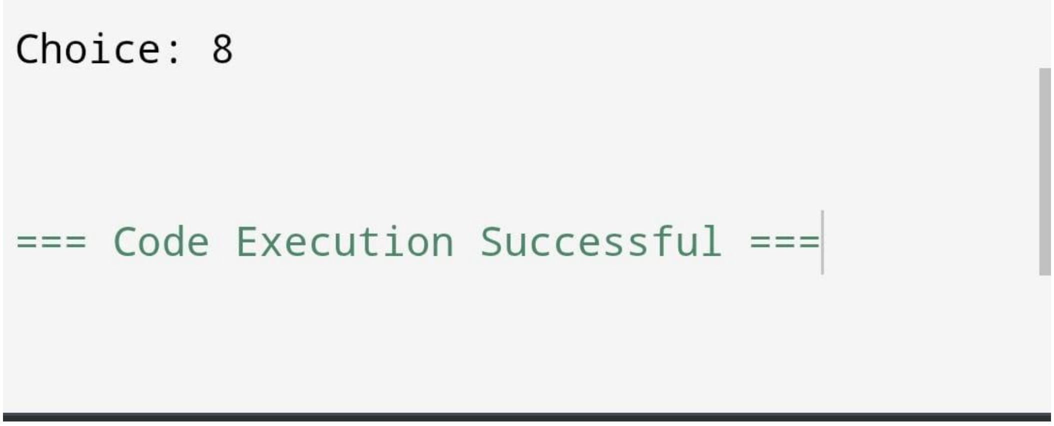
*Confirmation: A message “Player deleted!” confirms successful deletion.

8) Exit:

*Code Execution: This option indicates that the program has run successfully without errors.

*Purpose: Confirms that all previous operations (add, display, search, delete) executed correctly.

*Feedback: Provides the user with a success message: “Code Execution Successful.”



```
Choice: 8


=== Code Execution Successful ===
```