PROJECT REPORT

on

Cricket Scorecard

submitted towards the partial fulfillment of

the requirement for the award of the degree of

Bachelor of Technology

in

Object Oriented Programming

Submitted by

Shailesh (2K20/SE/124)

Tanishka Anand (2K20/SE/141)

Under the Subject Teacher

Dr. Sonali Chawla

Delhi Technological University

Bawana Road. Delhi -110042

March -2021

2021

DECLARATION

We hereby certify that the work, which is presented in the Project entitled "The Cricket Scorecard" in fulfilment of the requirement for the award of the Degree of Bachelor of Technology in Object Oriented Programming, Delhi Technological University,Delhi is an authentic record of our own, carried out under the supervision of Dr. Sonali Chawla. The work presented in this report has not been submitted and not under consideration for the award for any other course/degree of this or any other Institute/University.

Tanishka Anand

2K20/SE/141

B.Tech, Object Oriented Programming

Shailesh

2K20/SE/124

B.Tech, Object Oriented Programming

Place: Delhi

Date:-8/11/2021

Subject Teacher Name and Signature

Dr. Sonali Chawla

INDEX

INTRODUCTION

The project cricket scorecard developed to provide user with an update of the cricket even when the user is not watching the mach. The user can use this anytime, anywhere to see the teams, matches, player's squad, runs scored by each player and can also view the boundaries hit by the batsmen as well as the wickets taken by the bowler. This gives original experience of watching the match by the user.

Each and every match details such as the description about the team and team members will be stored in the repository system in the form of database. That database could be utilized by the mislaid by the users. Each and every match can be updated lively using this system. As soon as someone checks the scoreboard, details of a particular player can be viewed such as the number of runs the player scored, boundaries hit by the player and wickets taken by the player. This software is error free anyone can use this software

OBJECTIVE

The main aim of this Project is to design and include statistics of each player and store the scores of each batsmen along with the wickets of each bowler. Cricket being a special part of the lives of many people, there will be many takers for such a system and the ability to follow the match without seeing the video will make it interesting for many. A user who is unable to watch the event like someone who is busy with their work, can easily check the score on a regular basis to get updates on what is happening. The system will keep posting updated scores and the team line-up during the match. The admin will store upcoming match details ,this will help the admin easily load information at the time of the match. This project will help the people who need to improve their performance of the event which will help to progress the betterment.

This system has been developed to override the problems prevailing in the practicing manual system. This system is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. Every organization, whether big or small, has challenges to overcome and managing the information of matches, players, etc. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. It’s a user-friendly Management System which leads to error free, secure, reliable and fast management system.

MANUAL SYSTEM

The existing system saves all the team and team members games format system manually. Manage the activities like manual decision making, processing, announcement, scoring data and handling players & team information are very tough process. Moreover it will make lot of confusions and risks to make further process. This leads to wrong decision making in the event.

The existing system is to manually alerts the system to customer and maintains the player details, and status are in records. It will be more difficult to maintain and gathering information about specific records. It will take more time. As there is lot of data work involved, skilled staffs are used. So it becomes dependable for the management on these people. The reports are not verified to the highest extend to avoid any miscommunication and misfortune of the center.

The existing system of watching cricket is generally on the television. Most matches are not scheduled on holidays and this will allow people access to the match regardless of their location. Some sites do exist that display text commentary but they are very impersonal.

PROPOSED SYSTEM VS MANUAL SYSTEM

The proposed system "Cricket Score Card System" is utilized by the particular person (administrator). Main objective of the project is to develop the software for the event requirement. In this project used to maintain the details in database so easily retrieve the details from the database. This system also having the details of player and match are maintained in the repository management system. The reports are useful to maintain the match and and details about the players and complete the work as simple and as quick. Report is generated and saved in non-editable format.

Advantages of the Proposed System :-

- Easily maintain all the player details.
- Report generation is easier.
- Easy to maintain score details.
- Ensure user security.

Disadvantages of the existing system :-

- *Time consuming:* The manual processing is taking more time. It takes lots of time to record the process and transaction into a paper.
- *Security is not assured:* Security is not assured for the records of the organization. The need for computerizing arises in order assure the security of the records from fire or other destruction.
- *Space consuming:* A lot of space is required to maintain the record physically. To solve the problem they are going for computerization.

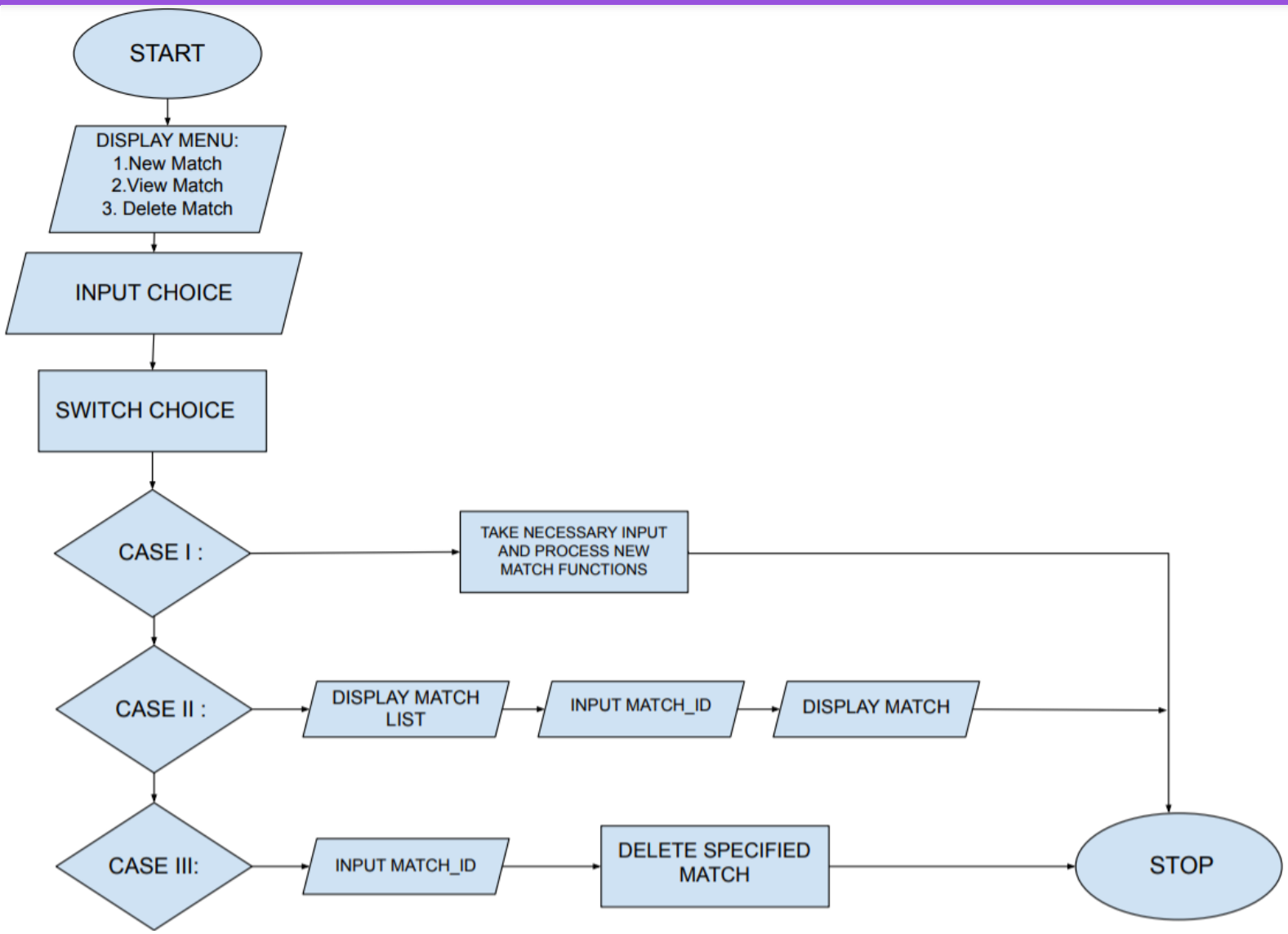
METHODOLOGY

The project begins by presenting the user with a menu from which they can create a new tabular structure, which will be used to store all of the data related to the scorecard which will get stored in the database. Initial input which we take from the user are the Match details such as Competition, venue, Toss won by, Date of the match to which we assign relevant data types.

We then provide a table-like structure in which we collect batsman and bowler information. After gathering all of the relevant information, we begin assigning the runs achieved by the batsman and the bowler's performance to their respective playing turns. As runs are assigned to the batsman, simultaneously batsman's total runs and number of _4s and _6s are displayed. Similarly, for the bowler (now bowling), his overs, maidens, economy, number of no balls (if any), and total runs provided by the bowler are presented in conjunction with
each input.

The admin will update details of upcoming cricket matches. The admin play the major role as the admin is responsible for carrying out the major operations such as updating match details, score updates etc., It maintains information regarding players as well.

BASIC FLOW



The Program will execute in the following way taking necessary input and adding them from temporary variable storages to Database. All the necessary table generation and output view has been specified in the functions defined in the program and the user can easily manage all the information without any issues.

OVERVIEW

The Project consists of 3 major classes Match, Team_Batsman and Team_Bowler. Basic Match details such as Competition, Date, Venue, Umpires, Teams, and their total scores are defined by Match Class. This class consists of 4 Functions each of which is defined to perform significant such as take Input, database entries, View Matches and Delete Matches.

Further Remaining two classes consists of consists of three functions each which defines Batsmen and Bowler details and performances. Functions are defined for carrying out table creation, take batsmen and bowler performance inputs and registering of taken inputs into their respected created tables in the database.

Each match contains one main record in the Match Details table with a unique ID allocated (specified as the database's primary key), and each match input generates four tables (two for each team in terms of batting and bowling).

REPOSITORY STRUCTURE

MATCH DETAILS TABLE :

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	MATCH_ID	varchar(7)	utf8mb4_general_ci		No	*None*
☐	2	COMPETITION	varchar(70)	utf8mb4_general_ci		Yes	*NULL*
☐	3	TEAM_1	varchar(50)	utf8mb4_general_ci		Yes	*NULL*
☐	4	TEAM_2	varchar(50)	utf8mb4_general_ci		Yes	*NULL*
☐	5	DATE	varchar(10)	utf8mb4_general_ci		Yes	*NULL*
☐	6	VENUE	varchar(70)	utf8mb4_general_ci		Yes	*NULL*
☐	7	MATCH_UMPIRES	varchar(60)	utf8mb4_general_ci		Yes	*NULL*
☐	8	THIRD_UMPIRE	varchar(70)	utf8mb4_general_ci		Yes	*NULL*
☐	9	MATCH_REFREE	varchar(50)	utf8mb4_general_ci		Yes	*NULL*
☐	10	TEAM1_RUNS	int(6)			Yes	*NULL*
☐	11	TEAM1_WKTS	int(2)			Yes	*NULL*
☐	12	TEAM2_RUNS	int(6)			Yes	*NULL*
☐	13	TEAM2_WKTS	int(2)			Yes	*NULL*

Match ID (Primary Key) is being specified as varchar datatype has no numeric restrictions and thus identifies each and every match uniquely.

BATSMEN PERFORMACE TABLE :

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Player**	varchar(70)	utf8mb4_general_ci		Yes	*NULL*
☐	2	**Runs**	int(11)			Yes	*NULL*
☐	3	**Balls**	int(11)			Yes	*NULL*
☐	4	**4s**	int(11)			Yes	*NULL*
☐	5	**6s**	int(11)			Yes	*NULL*
☐	6	**SR**	float			Yes	*NULL*
☐	7	**Status**	varchar(70)	utf8mb4_general_ci		Yes	*NULL*

Each time a team is defined a new batsmen table is automatically generated via SQL query with he following structure. The structure is unique to each match since table name consists of MATCH_ID which is the primary key to the main table and unique to each team as table name also contains team number.

BOWLER PERFORMACE TABLE :

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Bowler**	varchar(70)	utf8mb4_general_ci		Yes	*NULL*
☐	2	**Overs**	int(11)			Yes	*NULL*
☐	3	**Maidens**	int(11)			Yes	*NULL*
☐	4	**Runs**	int(11)			Yes	*NULL*
☐	5	**Wickets**	int(11)			Yes	*NULL*

This table structure and name is defined similar to Batsmen table structure and therefore is unique to every match and team.

IMPLEMENTATION

```
#include <bits/stdc++.h>
//to include basic input out stream, math functions string functions
#include <windows.h>
#include <mysql.h> // to invoke Database connectivity
#include <sstream> //to pass string as queries in database


using namespace std;

/*
Classes Defined :
1.Match for Match Details : Connected with Match_Details Table in Database
2.Team_Batsman    :   for Generating Batting Details of the match
3.Team_Bowler   :    for Generating Bowling Details of the match
 */

 //Forward Declaration of classes
class Team_Batsman;
class Team_Bowler;

   //Forward Declaration of View functions
     void Batting_View(string Match_ID, string a);
     void Bowling_View(string Match_ID, string a);



class Match
{
    private:
        //Defining variables for all necessary match specific details
```

```
        string Competition;
        string Team1,Team2 ;
        int Team1_runs, Team1_wickets;
        int Team2_runs, Team2_wickets;
        string date;
        string venue;
        string Umpires, Third_Umpire,Match_refree;

        //making other two classes friends to enable private data sharing
        friend class Team_Batsman;
        friend class Team_Bowler;

    public:
        string Match_ID;
        /*defining Match_ID as public since it is passed as argument in
table creation
        and match view functions */
        void setdata_Match(void); //Input Match Details
        void Match_Database_Entry(void); //Enter data in database
        friend void MatchList_View(void); //Extract and view list of
matches from database
        friend void Match_View(void);//View a specific match selected via
MATCH_ID
        friend void Delete_Match(void);//Delete a specific match related
tables

};


//Match All Functions


/*Following function Inputs all the necessary match details taken from the
user
and assigns to defined variables in the class*/
void Match :: setdata_Match(void)
{

//Initiating Data Input
    cout<<endl<<"----------NEW MATCH-----------"<<endl<<endl;
```

```
    cout<<"Enter Match ID : "; //ID unique to every match
    cin>>Match_ID;
    cout<<"Enter Competition : "; //mention Competition
    getline(cin, Competition);
    getline(cin, Competition);
    cout<<"Match Between ==> \nTeam 1 :";
   //taking team details date and match venue
    getline(cin, Team1);
    cout<<"Team 2: ";
    getline(cin, Team2);
    cout<<"DATE : ";
    getline(cin, date);
    cout<<"Match Venue : ";
    getline(cin, venue);
    //match refree and umpire inputs
    cout<<"Match Umpires : ";
    getline(cin, Umpires);

    cout<<"Third Umpire : ";
    getline(cin, Third_Umpire);

    cout<<"Match Refree : ";
    getline(cin, Match_refree);
    //Taking final score input
    cout<<endl<<"Enter "<<Team1<<" Runs / Wickets : ";
    cin>>Team1_runs>>Team1_wickets;
    cout<<"Enter "<<Team2<<" Runs / Wickets : ";
    cin>>Team2_runs>>Team2_wickets;


}

/*Following function stacks up all the match input data taken from the
user
which has been stored in temporary assigned variables in the tables
created
in the database for permanent storage purposes*/
void Match :: Match_Database_Entry(void){
    //Initiating Database connection
    MYSQL* conn;
```

```
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);
    int qstate=0 ;
    //Checking if connection is successful
        if(conn)
    {
        cout<<"DATABASE CONNECTED......."<<endl;
    }
    else{
        cout<<"DATABASE NOT CONNECTED........."<<endl;
    }
    stringstream ss;//Defining variable to pass query in the database for
data insertion
    ss << " INSERT INTO Match_Details VALUES('" << Match_ID<< "','" <<
Competition << "','" << Team1 << "','" << Team2 << "','" << date << "','"
<< venue << "','" << Umpires << "','" << Third_Umpire << "','" <<
Match_refree << "','" << Team1_runs<<  "','" << Team1_wickets <<  "','" <<
Team2_runs <<  "','" << Team2_wickets <<  "')";
    string query = ss.str();
    const char* q =query.c_str();
    qstate = mysql_query(conn, q);
    //Checking if record successfully generated or not
    if(qstate ==0){
        cout<<"Record Inserted..."<<endl;
    }
    else{
        cout<<"Failed"<<endl;
    }

}


/*Following function is defined for data extraction for display purposes
It invokes database connection and display specific details to let the
user know
all the matches which are there in database with their unique assigned
MATCH_IDs*/
void MatchList_View(void){
    //Initiating Database connection
```

```
    MYSQL* conn; //defining connection variable
    MYSQL_ROW row;
    MYSQL_RES* res;
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);
    if(conn){
            //Passing query to generate match lists from database
        int qstate = mysql_query(conn, "SELECT MATCH_ID , COMPETITION
FROM Match_Details  ");

        if(!qstate){
            res = mysql_store_result(conn);
        //specifying format of matches list
                cout<<"-------------------------------------------------
-----------------------"<<endl;
                cout<<"  Match ID    |                   LIST OF ALL
MATCHES                     "<<endl;
                cout<<"-------------------------------------------------
-----------------------"<<endl;

            while(row = mysql_fetch_row(res)){
              //specifying display of matches list
                    cout<<"    "<<setw(5)<<row[0]<<"     |         " <<"
"<<row[1]<<endl;  //Display All matches
                     cout<<"-----------------------------------------
----------------------------"<<endl;
            }
        }
    }
}



/*Following function is defined for data extraction for display purposes
It invokes database connection followed by asking for MATCH_ID and
displays all the associated records and tables associated with match in
the specified manner*/
        void Match_View(void){
            MYSQL* conn; //defining connection variable
```

```
            MYSQL_ROW row;
            MYSQL_RES* res;
            conn= mysql_init(0);
            conn=mysql_real_connect(conn, "192.168.29.53"
,"shailesh","1234", "cricket", 0, NULL, 0);
            if(conn){
                    /*Taking Match_ID as input to pass Match specific data
generation query into
                    the database */
                string sqlQuery;
                cout<<"Enter Match_ID value : "<<endl;
                cin>>sqlQuery;

                string id; //creating variable to retain Match ID
                id=sqlQuery;
             //Query to select data from a specific table for display
            sqlQuery= "SELECT * FROM Match_Details WHERE  MATCH_ID =  "+
sqlQuery;
            int qstate =mysql_query(conn, sqlQuery.c_str());

        if(!qstate){
                res = mysql_store_result(conn);
                while(row = mysql_fetch_row(res)){

                        //Specifying display structure of match details
display
                           cout<<"----------------------------------------
---------------------------------"<<endl;
                           cout<<"|"<<"COMPETITION : "<<row[1]<<"
||        "<<"DATE : "<<row[4]<<endl;
                           cout<<endl;
                           cout<<"|"<<"MATCH BETWEEN : "<<row[2]<<" V/S
"<<row[3]<<"    "<<endl;
                          cout<<endl;
                            cout<<"|"<<"VENUE : "<<row[5]<<endl;
                         cout<<endl;
                            cout<<"|"<<"MATCH UMPIRES : "<<row[6]<<endl;
                             cout<<endl;
                            cout<<"|"<<"THIRD UMPIRE : "<<row[7]<<"  |
"<<"MATCH REFREE : "<<row[8]<<endl;
```

```
                            cout<<endl;
                            cout<<"|"<<row[2]<<" :
"<<row[9]<<"/"<<row[10]<<"     |     "<<row[3]<<" :
"<<row[11]<<"/"<<row[12]<<endl;


                             cout<<"-------------------------------------
-----------------------------------"<<endl<<endl;

                        }

                }
            //calling function from batsmen and bowler classes and
displaying
            //result extracted from database
          cout<<"\n\t---TEAM 1 BATTING PERFORMANCE ----"<<endl;
         Batting_View(id, "1");
         cout<<"\n\t---TEAM 1 BOWLING PERFORMANCE ----"<<endl;
         Bowling_View(id, "1");
         cout<<endl;
         cout<<"\n\t---TEAM 2 BATTING PERFORMANCE ----"<<endl;
         Batting_View(id, "2");
         cout<<"\n\t---TEAM 2 BOWLING PERFORMANCE ----"<<endl;
         Bowling_View(id, "2");

        }
    }



/*Following function is defined for Match Deletion purposes
It asks the user for Match specific ID and Deletes the Match record from
the
main table and also deletes all batsmen and bowler tables associated with
specified match from the database at once*/
void Delete_match(void){
    //Initiating Database connection
    MYSQL* conn;//Defining connection variable
    conn= mysql_init(0);
```

```
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);

        if(conn){
        string sqlQuery;
        //asking for Match_Id
        cout<<"Enter Match_ID value : "<<endl;
        cin>>sqlQuery;
        string id;
        //retaining match_id to pass as query for deletion of associated
tables
        id=sqlQuery;
        //Query to Delete Match record
        sqlQuery = "DELETE FROM Match_Details WHERE  MATCH_ID =  "+
sqlQuery;
         int qstate =mysql_query(conn, sqlQuery.c_str());
        stringstream ss;
        //Query to drop tables
        ss << " DROP TABLE
"+id+"_team_1_batting,"+id+"_team_1_bowling,"+id+"_team_2_batting,"+id+"_t
eam_2_bowling ;";
        string query = ss.str();
        const char* q =query.c_str();
        int rstate = mysql_query(conn, q);

        }
    }


    //-----------------------------------------------------------------------
-----------------------------------------------

//Batsmen Information Class
    class Team_Batsman{
    private:
        //Defining information variables for Batsman
        string player;

        /*Rather than creating individual integer variables
        for runs, balls, 4s and 6s
```

```
        Creating an Array for taking all the numeric input */
        int p[4];
        //Defining Strike Rate as SR
        float SR;
        string status;

    public:
        void Create_Table_Batting(string Match_ID, string a);// Function
to create table

        //Function to enter data from temporary assigned variables to
database
        void Batting_Database_Entry(string Match_ID, string a, int num);
        //Function to display batting info
        friend void Batting_View(string Match_ID, string a);

    };

//Team_Batsman Functions   :


/*Following function is defined for Creating team specific batting
performance
table via SQL query*/
    void Team_Batsman :: Create_Table_Batting(string Match_ID, string a)
{
    //Initiating Database Connection
    MYSQL* conn;
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);
    //Query for Table creation
    stringstream ss;
    ss << "CREATE TABLE "<< Match_ID<<"_Team_"<<a<<"_Batting  ("
    "Player varchar(70),"
    " Runs int , "
    "Balls int , "
    "4s int , "
    "6s int , "
    "SR float , "
```

```
    "Status varchar(70) ) ;";
    string query = ss.str();
    const char* q =query.c_str();
    int qstate = mysql_query(conn, q);
    //Checking if table created successfully
    if(qstate ==0){
        cout<<"Table Created..."<<endl;
    }
    else{
        cout<<"Failed!!"<<endl;
    }
}

/*Following function stacks up all the Batsmen input data taken from the
user
which has been stored in temporary assigned variables in the tables
created
in the database for permanent storage purposes
The function takes arguments MATCH_ID  to know for which match and team
table
is being created and also an integer argument to know the number of
players*/
void Team_Batsman :: Batting_Database_Entry(string Match_ID, string a, int
num)
{
    //Initiating Database Connection
    MYSQL* conn;//Defining connection variable
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);

    cout<<"\n------TEAM"<<a<<" BATTING RECORD INPUT ------\n";
    //Taking input for various batsmen in loop
    for(int j=0;j<num;j++){
    fflush(stdin);
    cout<<"Enter Player Name : ";
    getline(cin, player);
    cout<<"Input \n  Runs   Balls   4s   6s  :"<<endl;
  for(int i=0;i<4;i++)
  {
```

```
      cout<<"\t";
      cin>>p[i];//taking numeric input into array
  }
  //Strike rate calculation for each batsman
  SR=((float) p[0]/(float) p[1])*100;

  fflush(stdin);
  cout<<"Enter Status : ";
  getline(cin, status);

    stringstream ss;
    //Query to transfer data from variables and array to database
    ss << " INSERT INTO "<< Match_ID<<"_Team_"<<a<<"_Batting VALUES('" <<
player<< "','" << p[0] << "','" << p[1] << "','" << p[2] << "','" << p[3]
<< "','" << SR<< "','" << status <<  "')";
    string query = ss.str();
    const char* q =query.c_str();
    int qstate = mysql_query(conn, q);
     //Checking if record successfully generated or not
    if(qstate ==0){
        cout<<"Record Inserted...\n"<<endl;
       }
    else{
        cout<<"Failed!!\n"<<endl;
       }
    }

}
/*Following function is defined to view data extracted from database*/
void Batting_View(string Match_ID, string a){
    //Initiating Database Connection
    MYSQL* conn; //Defining connection variable
    MYSQL_ROW row;
    MYSQL_RES* res;
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);
    if(conn){
            string sqlQuery;
            //Query to extract data from tables
```

```
    sqlQuery ="SELECT *  FROM  "+ Match_ID+"_Team_"+a+"_Batting";
        int qstate = mysql_query(conn, sqlQuery.c_str() );

        if(!qstate){
            res = mysql_store_result(conn);
                 cout<<"-------------------------------------------------
-----------------------------------"<<endl;
                 cout<<"       Player    |     R(B)      |   4s  |   6s  |
SR     |    Status    "<<endl;
                 cout<<"-------------------------------------------------
-----------------------------------"<<endl;

            while(row = mysql_fetch_row(res)){
                    //Specifying format for displaying extracted data from
database
                     cout<<setw(15)<<row[0]<<"  |  "
<<setw(5)<<row[1]<<"("<<setw(3)<<row[2]<<") "<<"  |  "<<setw(3)<<row[3]<<"
|  "<<setw(3)<<row[4]<<"  |  "<<setw(9)<<row[5]<<"  |    "<<row[6]<<endl;
//Display Batsmen Performances
                     cout<<"-------------------------------------------
---------------------------------------"<<endl;
            }
        }
    }
}


//-----------------------------------------------------------------------
---------------------------------------

    //Bowler Information Class
    class Team_Bowler{
        private:
        string bowler;
        int p[4];

     public:
     void Create_Table_Bowling(string Match_ID, string a);//Function to
create table
     void Bowling_Database_Entry(string Match_ID, string a, int num);
```

```
     friend void Bowling_View(string Match_ID, string a);

    };
//Team_Bowler Functions   :

/*Following function is defined for Creating team specific bowling
performance
table via SQL query*/
    void Team_Bowler :: Create_Table_Bowling(string Match_ID, string a)
{
    //Initiating Database Connection
    MYSQL* conn;//Defining connection variable
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);

     stringstream ss;
    ss << "CREATE TABLE "<< Match_ID<<"_Team_"<<a<<"_Bowling ("
    "Bowler varchar(70),"
    " Overs int , "
    "Maidens int , "
    "Runs int , "
    "Wickets int "
    " ) ;";
    string query = ss.str();
    const char* q =query.c_str();
    int qstate = mysql_query(conn, q);
    //Checking if table created successfully
    if(qstate ==0){
        cout<<"Table Created..."<<endl;
    }
    else{
        cout<<"Failed!!"<<endl;
    }

}
/*Following function stacks up all the Bowler input data taken from the
user
which has been stored in temporary assigned variables in the tables
created
```

```
in the database for permanent storage purposes
The function takes arguments MATCH_ID  to know for which match and team
table
is being created and also an integer argument to know the number of
players*/
void Team_Bowler :: Bowling_Database_Entry(string Match_ID, string a, int
num)
{
    //Initiating Database Connection
    MYSQL* conn; //Defining connection variable
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);

    cout<<"\n------TEAM_"<<a<<" BOWLING RECORD INPUT ------\n";
    //Taking bowler inputs in loop
    for(int j=0;j<num;j++){
    fflush(stdin);
    cout<<"Enter Bowler Name : ";
    getline(cin, bowler);
    cout<<"Input \n  Overs   Maidens   Runs  Wickets  :"<<endl;
  for(int i=0;i<4;i++)
  {
      cin>>p[i]; //taking Numeric Input
  }

    stringstream ss;//variable to write query
     //Query to transfer data from variables and array to database
    ss << " INSERT INTO "<< Match_ID<<"_Team_"<<a<<"_Bowling VALUES('" <<
bowler<< "','" << p[0] << "','" << p[1] << "','" << p[2] << "','" << p[3]
<<  "')";
    string query = ss.str();
    const char* q =query.c_str();
    int qstate = mysql_query(conn, q);
    if(qstate ==0){
        cout<<"Record Inserted...\n"<<endl;
    }
    else{
        cout<<"Failed!!\n"<<endl;
}
```

```
    }

}
/*Following function is defined to view data extracted from database*/
void Bowling_View(string Match_ID, string a){
    //Initiating Database Connection
    MYSQL* conn; //Defining connection variable
    MYSQL_ROW row;
    MYSQL_RES* res;
    conn= mysql_init(0);
    conn=mysql_real_connect(conn, "192.168.29.53" ,"shailesh","1234",
"cricket", 0, NULL, 0);
    if(conn){
            string sqlQuery;
        //Query to extract data from table
    sqlQuery ="SELECT *  FROM  "+ Match_ID+"_Team_"+a+"_Bowling";
        int qstate = mysql_query(conn, sqlQuery.c_str() );

        if(!qstate){
            res = mysql_store_result(conn);
                 cout<<"-------------------------------------------------
-----"<<endl;
                 cout<<"       Bowler    |    O   |   M  |   R   |    W
"<<endl;
                 cout<<"-------------------------------------------------
-----"<<endl;

            while(row = mysql_fetch_row(res)){
                     //Specifying format for displaying extracted data
from database
                     cout<<setw(15)<<row[0]<<"  |  " <<setw(5)<<row[1]<<"
|"<<setw(4)<<row[2]<<"  |  "<<setw(3)<<row[3]<<"  |
"<<setw(3)<<row[4]<<endl;  //Display Bowler Performances
                     cout<<"---------------------------------------------
---------"<<endl;
            }
      }
    }
}
```

```
//-------------------------------------------------------------------------
---------------------------------------------

//Main_Function
int main()
{
    //Setting display text colour
    system("Color 0B");
    fflush(stdin);
    int choice;
    cout << endl
         << endl
         << "--------------------WELCOME TO CRICKET SCORE MANAGEMENT
SYSTEM----------------------" << endl;
    while (1)
    {
        cout << endl
             << "CHOOSE OPERATION : " << endl;
        cout << "1.Match Details Input\n2.View Match\n3.Delete Match
\nEnter choice : ";
        cin >> choice;

        if (choice == 1)
        {

            //Match Details Input
            Match TD;
            //New Match
            TD.setdata_Match();

            //Match Details Database Entry (New Record)
            TD.Match_Database_Entry();
            //Creating objects for Batting and Bowling teams
            Team_Batsman A, B;
            Team_Bowler C, D;

           //Input number of batsmen and bowlers in each team
            int batsmen, bowlers;
            cout << endl
                 << "--------INPUT INFO----" << endl;
```

```
            cout << "Enter No. of Batsmen in Each Team : ";
            cin >> batsmen;
            cout << "Enter No. of Bowler in Each Team : ";
            cin >> bowlers;

            /*Team specific Batting and Bowling table creation and
            calling neccesary functions for input
            Also passing MATCH_ID , team no. and no of player
            as arguments*/
            A.Create_Table_Batting(TD.Match_ID, "1");
            A.Batting_Database_Entry(TD.Match_ID, "1", batsmen);
            C.Create_Table_Bowling(TD.Match_ID, "1");
            C.Bowling_Database_Entry(TD.Match_ID, "1", bowlers);

            B.Create_Table_Batting(TD.Match_ID, "2");
            B.Batting_Database_Entry(TD.Match_ID, "2", batsmen);
            D.Create_Table_Bowling(TD.Match_ID, "2");
            D.Bowling_Database_Entry(TD.Match_ID, "2", bowlers);
        }

        else if (choice == 2)
        {
            //View Matches List
            MatchList_View();
            //View Specific match
            Match_View();
        }
        else if (choice == 3)
        {
            //Delete Match Record and associated tables
            Delete_match();
        }
        else
        {
            cout << "Invalid choice !!!";
        }
    }
   return 0;
}
```

OUTPUT

```
--------------------WELCOME TO CRICKET SCORE MANAGEMENT SYSTEM----------------------

CHOOSE OPERATION :
1.Match Details Input
2.View Match
3.Delete Match
Enter choice : 1
```

Output begins with the display of Menu.
On selection of choice 1, New Match entry will initiate.

NEW MATCH FUNCTIOANALITY:

```
----------NEW MATCH-----------

Enter Match ID : 101
Enter Competition : ICC Mens T20 WC 2021
Match Between ==>
Team 1 :South Africa
Team 2: West Indies
DATE : 26/10/2021
Match Venue : Dubai International Stadium,Dubai
Match Umpires : Aleem Dar(PAK), Paul Reiffel(AUS)
Third Umpire : Chris Brown(NZ)
Match Refree : David Boon(AUS)

Enter South Africa Runs / Wickets : 144 2
Enter West Indies Runs / Wickets : 143 8
DATABASE CONNECTED.......
Record Inserted...
```

101	ICC Mens T20 WC 2021	South Africa	West Indies	26/10/2021	Dubai International Stadium,Dubai	Aleem Dar(PAK), Paul Reiffel(AUS)	Chris Brown(NZ)	David Boon(AUS)

Match Details will be taken as shown and further database would be connected and match record would be inserted. Match_ID being primary key in the main table would play the main role through out the source code execution.

```
--------INPUT INFO----
Enter No. of Batsmen in Each Team : 5
Enter No. of Bowler in Each Team : 5
Table Created...

------TEAM1 BATTING RECORD INPUT ------
Enter Player Name : Bavuma
Input
  Runs   Balls   4s   6s  :
        2   3  0   0
                        Enter Status : run out(Russel)
Record Inserted...

Enter Player Name : Hendricks
Input
  Runs   Balls   4s   6s  :
        39  30  4  1
                        Enter Status : c.Akeal Hosein
Record Inserted...

Enter Player Name : Dussen
Input
  Runs   Balls   4s   6s  :
        43  51  3  0
                        Enter Status : not out
Record Inserted...

Enter Player Name : Pretorius
Input
  Runs   Balls   4s   6s  :
        2  3  0  0
                        Enter Status : not out
Record Inserted...

Enter Player Name : Miller
Input
  Runs   Balls   4s   6s  :
        21 30 0 0
                        Enter Status : b.Akeal Hosein
Record Inserted...

Table Created...
```

```
Table Created...

------TEAM2 BATTING RECORD INPUT ------
Enter Player Name : Simmons
Input
  Runs   Balls   4s   6s  :
        16  35   0  0
                        Enter Status : b.Rabada
Record Inserted...

Enter Player Name : Lewis
Input
  Runs   Balls   4s   6s  :
        56  35 3  6
                        Enter Status : c.Rabada
Record Inserted...

Enter Player Name : Pooran
Input
  Runs   Balls   4s   6s  :
        12  7  2  0
                        Enter Status : c.Miller
Record Inserted...

Enter Player Name : Pollard
Input
  Runs   Balls   4s   6s  :
        26  20  2  1
                        Enter Status : b.Pretorius
Record Inserted...

Enter Player Name : Russel
Input
  Runs   Balls   4s   6s  :
        5  4  1  0
                        Enter Status : b.Nortjie
Record Inserted...
```

Number of Batsmen and Bowler would be taken to not restrict the Project to official 11 players' match. Further Batsmen Details specific to team and Match(via Match ID) would be taken as shown.

```
Table Created...

------TEAM_1 BOWLING RECORD INPUT ------
Enter Bowler Name : Markram
Input
  Overs   Maidens   Runs  Wickets  :
   3   1   22   0
Record Inserted...

Enter Bowler Name : Rabada
Input
  Overs   Maidens   Runs  Wickets  :
   4   0   27   1
Record Inserted...

Enter Bowler Name : Nortje
Input
  Overs   Maidens   Runs  Wickets  :
 4   0   24   2
Record Inserted...

Enter Bowler Name : Shamsi
Input
  Overs   Maidens   Runs  Wickets  :
  3   0   37   0
Record Inserted...

Enter Bowler Name : Pretorius
Input
  Overs   Maidens   Runs  Wickets  :
  2   0   17   3
Record Inserted...
```

```
------TEAM_2 BOWLING RECORD INPUT ------
Enter Bowler Name : Hosein
Input
  Overs   Maidens   Runs  Wickets  :
   4   0   27   1
Record Inserted...

Enter Bowler Name : Rampaul
Input
  Overs   Maidens   Runs  Wickets  :
   3   0  22   0
Record Inserted...

Enter Bowler Name : Russell
Input
  Overs   Maidens   Runs  Wickets  :
   3   0   36   0
Record Inserted...

Enter Bowler Name : Walsh
Input
  Overs   Maidens   Runs  Wickets  :
    3   0    26   0
Record Inserted...

Enter Bowler Name : Bravo
Input
  Overs   Maidens   Runs  Wickets  :
   4   0   23    0
Record Inserted...
```

Similarly bowling details input would taken with respect to each team.

	Table	Action						Rows
☐	101_team_1_batting		Browse	Structure	Search	Insert	Empty	Drop	
☐	101_team_1_bowling		Browse	Structure	Search	Insert	Empty	Drop	
☐	101_team_2_batting		Browse	Structure	Search	Insert	Empty	Drop	
☐	101_team_2_bowling		Browse	Structure	Search	Insert	Empty	Drop	

VIEW MATCH FUNCTIOANALITY:

```
CHOOSE OPERATION :
1.Match Details Input
2.View Match
3.Delete Match
Enter choice : 2
---------------------------------------------------------------------------
  Match ID    |                    LIST OF ALL MATCHES
---------------------------------------------------------------------------
      101     |                  ICC Mens T20 WC 2021
---------------------------------------------------------------------------
      102     |                  4 Day Franchise Series 2021-22
---------------------------------------------------------------------------
      103     |                  Women Big Bash League 2021
---------------------------------------------------------------------------
Enter Match_ID value :
101
```

On Selecting View Match choice, A list of Matches from Database would be displayed following which Match ID would be requested from the user.

```
---------------------------------------------------------------------------
|COMPETITION : ICC Mens T20 WC 2021        ||        DATE : 26/10/2021

|MATCH BETWEEN : South Africa V/S West Indies

|VENUE : Dubai International Stadium,Dubai

|MATCH UMPIRES : Aleem Dar(PAK), Paul Reiffel(AUS)

|THIRD UMPIRE : Chris Brown(NZ)  |  MATCH REFREE : David Boon(AUS)

|South Africa : 144/2     |     West Indies : 143/8
---------------------------------------------------------------------
```

On Entering the Match ID , All details respective to matches would be displayed.

```
       ---TEAM 1 BATTING PERFORMANCE ----
-------------------------------------------------------------------------------------
      Player    |     R(B)      |   4s  |   6s  |      SR     |    Status
-------------------------------------------------------------------------------------
        Bavuma  |      2(  3)   |    0  |    0  |    66.6667  |    run out(Russel)
-------------------------------------------------------------------------------------
     Hendricks  |     39( 30)   |    4  |    1  |        130  |    c.Akeal Hosein
-------------------------------------------------------------------------------------
        Dussen  |     43( 51)   |    3  |    0  |    84.3137  |    not out
-------------------------------------------------------------------------------------
     Pretorius  |      2(  3)   |    0  |    0  |    66.6667  |    not out
-------------------------------------------------------------------------------------
        Miller  |     21( 30)   |    0  |    0  |         70  |    b.Akeal Hosein
-------------------------------------------------------------------------------------

       ---TEAM 1 BOWLING PERFORMANCE ----
-------------------------------------------------------
      Bowler    |    O   |   M  |   R   |    W
-------------------------------------------------------
       Markram  |      3 |   1  |   22  |    0
-------------------------------------------------------
        Rabada  |      4 |   0  |   27  |    1
-------------------------------------------------------
        Nortje  |      4 |   0  |   24  |    2
-------------------------------------------------------
        Shamsi  |      3 |   0  |   37  |    0
-------------------------------------------------------
     Pretorius  |      2 |   0  |   17  |    3
-------------------------------------------------------


       ---TEAM 2 BATTING PERFORMANCE ----
-------------------------------------------------------------------------------------
      Player    |     R(B)      |   4s  |   6s  |      SR     |    Status
-------------------------------------------------------------------------------------
       Simmons  |     16( 35)   |    0  |    0  |    45.7143  |    b.Rabada
-------------------------------------------------------------------------------------
         Lewis  |     56( 35)   |    3  |    6  |        160  |    c.Rabada
-------------------------------------------------------------------------------------
        Pooran  |     12(  7)   |    2  |    0  |    171.429  |    c.Miller
-------------------------------------------------------------------------------------
       Pollard  |     26( 20)   |    2  |    1  |        130  |    b.Pretorius
-------------------------------------------------------------------------------------
        Russel  |      5(  4)   |    1  |    0  |        125  |    b.Nortjie
-------------------------------------------------------------------------------------
```

```
        ---TEAM 2 BOWLING PERFORMANCE ----
-------------------------------------------------------
       Bowler    |    O   |   M  |   R   |    W
-------------------------------------------------------
         Hosein  |      4 |   0  |   27  |    1
-------------------------------------------------------
        Rampaul  |      3 |   0  |   22  |    0
-------------------------------------------------------
        Russell  |      3 |   0  |   36  |    0
-------------------------------------------------------
          Walsh  |      3 |   0  |   26  |    0
-------------------------------------------------------
          Bravo  |      4 |   0  |   23  |    0
-------------------------------------------------------
```

All match related Batting and bowling data would be displayed in tabular form as shown.

DELETE MATCH FUNCTIOANALITY:

```
-------------------------------------------------------

CHOOSE OPERATION :
1.Match Details Input
2.View Match
3.Delete Match
Enter choice : 3
Enter Match_ID value :
103

CHOOSE OPERATION :
1.Match Details Input
2.View Match
3.Delete Match
Enter choice : 2
-------------------------------------------------------------------------
  Match ID    |                  LIST OF ALL MATCHES
-------------------------------------------------------------------------
      101     |                 ICC Mens T20 WC 2021
-------------------------------------------------------------------------
      102     |                 4 Day Franchise Series 2021-22
-------------------------------------------------------------------------
```

When a Match is deleted via ID , its record and all the associated tables are dropped collectively from the database.

CONCLUSION

It is concluded that the application works well and satisfy the end users. The application is tested very well and errors are properly debugged. This system eliminate and reduce the hardships faced by the manual system. Thus its more convenient than the manual system and it also reduces the confusion and problems faced in the manual system.

This system is user friendly so everyone can use easily. The end user can easily understand how the whole system is implemented. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully. Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one.