A Mini-Project Report  
on

**CRICKET SCOREBOARD**

Submitted for partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING   
IN  
INFORMATION TECHNOLOGY**

BY

B. Mahathi

(1602-19-737-080)

and

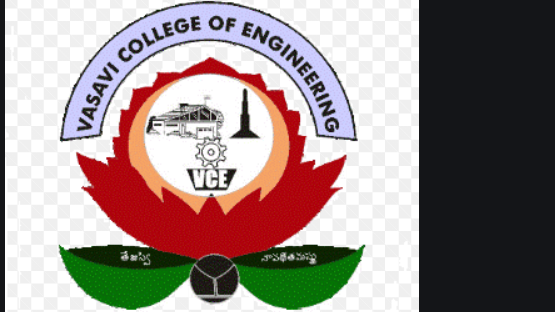
R. Sravya

(1602-19-737-109)

Under the guidance of

**Miss D.R.L PRASANNA**  
Assistant Professor.  
Department of IT

VCE,Hyderabad.



**Department of Information Technology   
Vasavi College of Engineering  
Hyderabad – 500075.**

# 

**ACKNOWLEDGEMENTS**

We would like to express our heartfelt gratitude to **Miss D.R.L Prasanna** mam, our project guide, for her valuable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, **DR. RAM MOHAN RAO** sir, for his steady support and the provision of every resource required for the completion of this project as the management of the institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project.

# ABSTRACT

This project implements an application which acts as a cricket score calculator, whose aims are to provide users with a smooth and easy way to calculate cricket score during any match.

The programming language used in this project is C.

The program asks for inputs on each ball of the match using which several data is calculated and stored. The information calculated consists of current run rate, required run rate, projected score, wickets remaining, individual stats of a player and obviously the present score. The duration of the match (overs) can be set by users.

The main objective of implementing this project is that, it can be used for storing and calculating data for short duration and local matches especially for college tournament matches and also for casual/friendly matches.

Though the scope of this project seems to be faded, but it becomes really helpful when the following facts are considered like prevents manipulation of data as stored in a soft copy which cannot be changed, fast and accurate calculations since this is done by a computer rather than human who often has to chance to miscalculate ,easy inputs through keyboard, can store huge amount of data regarding the match/tournaments without the risk of being damaged/losing it.

But, this, being a program can be only accessed by programmer.

The working and outputs can be understood in a better way by using this program.

**TABLE OF CONTENTS**

CHAPTER 1:

INTRODUCTION

* 1.1 Introduction
* 1.2 Limitations of the proposed system

CHAPTER 2:

LITERATURE SURVEY

* 2.1 Introduction
* 2.2 Existing System
* 2.3 Disadvantages of Existing system
* 2.4 Proposed System

CHAPTER 3:

ANALYSIS

* 3.1 Introduction
* 3.2 Software and Hardware Requirement
* 3.3 Block diagram
* 3.4 conclusion

CHAPTER 4:

IMPLEMENTATION & RESULTS

* 4.1 Method of Implementation
* 4.1.1 Classes
* 4.1.2 Output Screens
* 4.1.3 Result Analysis
* 4.1.4 future enhancements
* 4.1.5 Conclusion.

# CHAPTER-1

# INTRODUCTION

1. **1. Introduction:**

1) Our project “ CRICKET SCORE BOARD” is designed to calculate score , Strike Rate , Run Rate Of the cricket match occurred.

2) We can enter the runs scored per ball by an individual Batsman and

calculate the required score to win in given Overs of a match

3) To enhance this project , we use programming language i.e c.

4) This project introduces a cricket game score calculator .

5) This project enhances to calculate the Runrate automatically.

6). This project gives a perfect way to calculate cricket score.

## 1.2Limitations of the project:

1. Our Projects motto is to create an application which provides an easy calculation of cricket score in tournaments.
2. But the Data is entered manually with which much attention must be taken.
3. Calculations are done but it might take some time to access it.
4. The only limitation of this system is it takes input data manually which can’t be changed further.

**CHAPTER-2**

# LITERATURE SURVEY

## 2.1Introduction:

The cricket game was played in past many centuries .it was introduced in 18th century . Cricket is the national game of England. It reflects many athletics such as running ,long jump etc. The main aim of our Project is to reduce the difficulties of calculating cricket score related calculations. It is to design a proper way to calculate and record the scores of undergoing matches or tournaments.

## 2.2Existing System:

At present , we have no online website or an Application to store and calculate the score hoardings of cricket. Cricket score is calculated manually at present which may cause mistakes and improper expectations or graph of the game.

**2.3Drawbacks of Old System:**

The main drawback of the old system is the calculations are done manually.

It might invoke conflicts among players for wrong calculations.

This system can’t be accessed when ever required again i.e. we can’t see the past records of score and the properties of that respective match.

There exists some system , which are heavily equipped which can’t be used in tournaments. This might Take some time as it requires manual calculations.

## 2.4Proposed System:

In our project, we managed to calculate the score automatically when the score per ball is given manually. Here, we can obtain a proper graph of projected score and Run rate of the Batting Team.

## CHAPTER-3

## ANALYSIS

**3.1 Introduction:**

In this chapter, the description of the system requirements i.e. Hard-ware and Soft-ware for executive the program will be shelled out for the best user experience. This will also include the system requirements needed by the admin to alter the source code.

**3.2 Hardware Requirements & Software Requirements:**

* The System need some basic hardware to run .

Those are:-

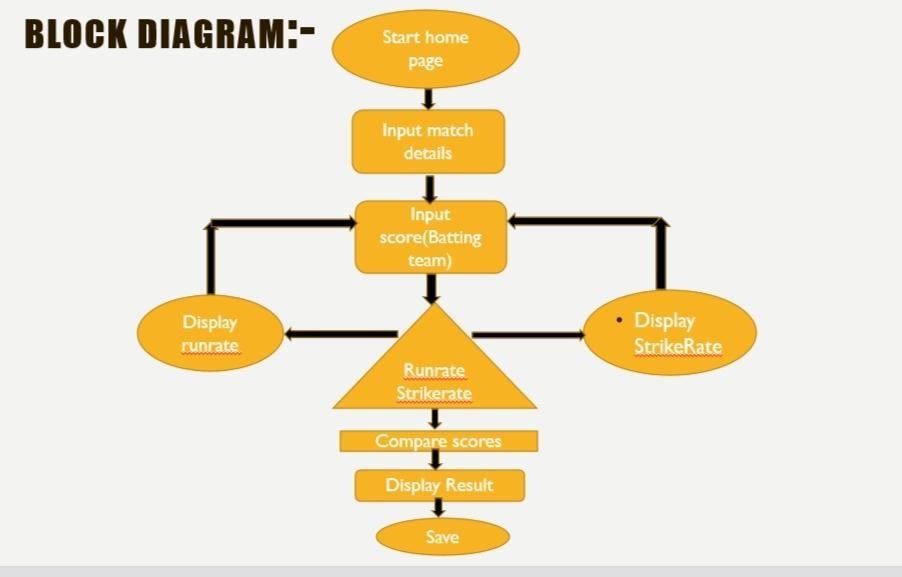
1. Good memory hard disk.
2. 8GB RAM.
3. 64 bit Machine
4. Good display facilities

* To run this project we need some basic software in computer/laptop.

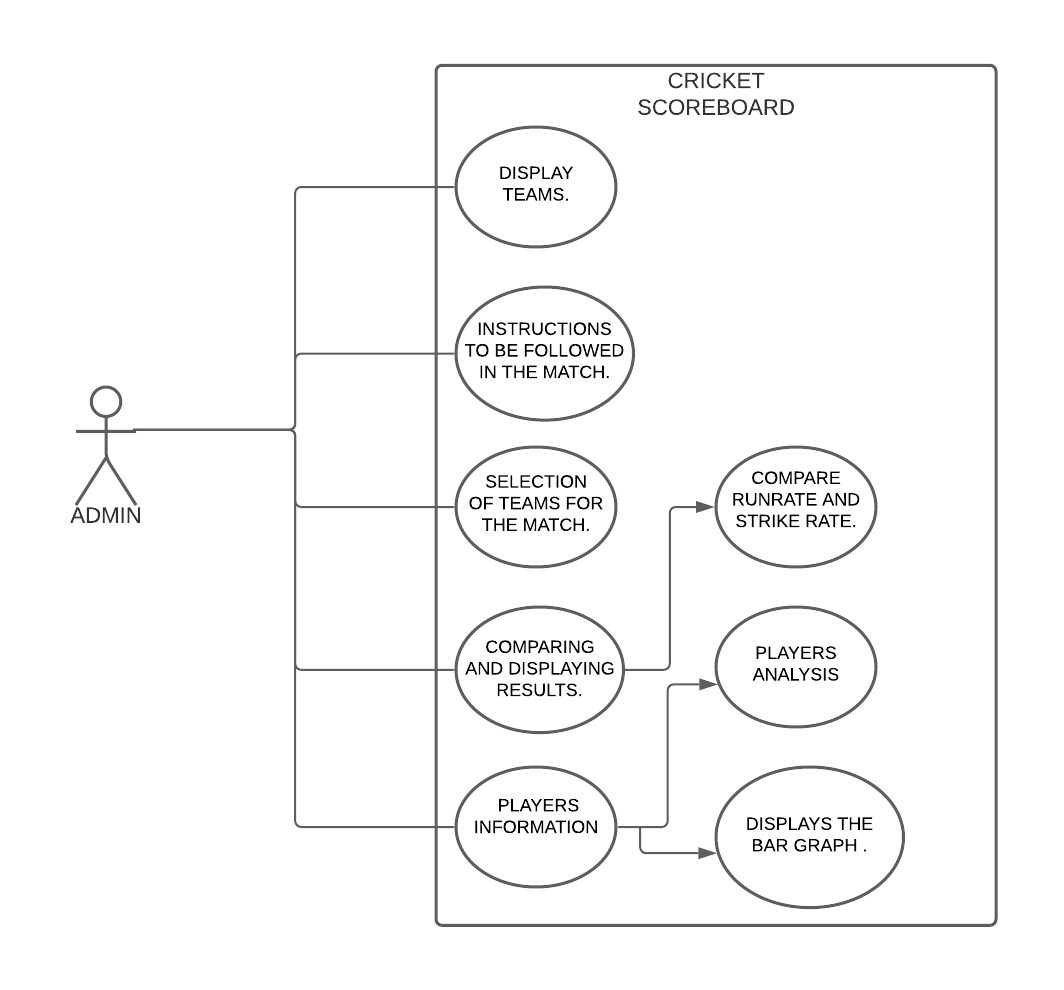
Those are:-

1. Text editor/Notepad.
2. An application to support and execute the Programming Language.

**3.3 Block Diagram:**



**USE CASE DIAGRAM:**



**3.4 Conclusion**

These are the system requirements for the project to run smoothly and efficiently and should be used for the alteration of the source code of this project.

**CHAPTER 4**

**IMPLEMENTATION &RESULTS**

**4.1** **Method of Implementation:**

**\*\*coding\*\***

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include <windows.h>

///This will set the forground color for printing in a console window.

void SetColor(int ForgC)

{

WORD wColor;

///We will need this handle to get the current background attribute

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

///We use csbi for the wAttributes word.

if(GetConsoleScreenBufferInfo(hStdOut, &csbi))

{

///Mask out all but the background attribute, and add in the forgournd color

wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);

SetConsoleTextAttribute(hStdOut, wColor);

}

return;

}

void ClearConsoleToColors(int ForgC, int BackC)

{

WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);

///Get the handle to the current output buffer...

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

///This is used to reset the carat/cursor to the top left.

COORD coord = {0, 0};

///A return value... indicating how many chars were written

/// not used but we need to capture this since it will be

/// written anyway (passing NULL causes an access violation).

DWORD count;

///This is a structure containing all of the console info

/// it is used here to find the size of the console.

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

///Here we will set the current color

SetConsoleTextAttribute(hStdOut, wColor);

if(GetConsoleScreenBufferInfo(hStdOut, &csbi))

{

///This fills the buffer with a given character (in this case 32=space).

FillConsoleOutputCharacter(hStdOut, (TCHAR) 32, csbi.dwSize.X \* csbi.dwSize.Y, coord, &count);

FillConsoleOutputAttribute(hStdOut, csbi.wAttributes, csbi.dwSize.X \* csbi.dwSize.Y, coord, &count );

///This will set our cursor position for the next print statement.

SetConsoleCursorPosition(hStdOut, coord);

}

return;

}

void SetColorAndBackground(int ForgC, int BackC)

{

WORD wColor = ((BackC & 0x0F) << 4) + (ForgC & 0x0F);;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), wColor);

return;

}

COORD coord = {0,0}; ///set the cordinate to 0, 0 (top-left corner of window);

void gotoxy(int x, int y){

coord.X = x; coord.Y = y; /// X and Y coordinates

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

void clearWindow(){

int i,j;

for(i = 37; i < 78; i++){

for(j = 7; j < 25; j++){

gotoxy(i,j);printf(" ");

}

}

return;

}

void window(){

SetColor(35);

printf("\n\t\t\t\*\*\*");printf("\n\t\t\tCRICKET PREMIERE LEAGUE\n ");printf("\n\t\t\t\*\*");

printf("\n\t\t\tVASAVI COLLEGE OF ENGINEERING\n");

printf("\n\t\t\t\*\*\*\*\n");

SetColor(17);

}

int sum1=0;

int sum2=0;

int w2=0;

int w1=0;

int b2=1;

int b1=1;

int main()

{

main:

system("cls"); /\* \*\*\*\*\*Main menu \*\*\*\*\*\*\*\* \*/

ClearConsoleToColors(17,15);

SetConsoleTitle("\n\tCRICKET SCORE READER");

window();

printf("\n\n\n\t\tMAIN MENU\n\t\t=====================\n\t\t[1] INSTRUCTIONS\n\t\t[2] CHECK WHEATHER\n\t\t[3] SELECTION OF TEAMS\n\t\t[4] PREVIOUS MATCH DETAILS\n\t\t[5] GRAPH REPRESENTATION\n\t\t[0] EXIT\n\t\t=================\n\t\t");

int ch;

printf("ENTER THE CHOICE:");

scanf("%d",&ch);

switch(ch)

{

case 0:

printf("\n\n\t\tAre you sure you want to exit?");

break;

/\* \*\*\*\*\*\*\*\*Instructions\*\*\*\*\* \*/

case 1:

system("cls");

printf("\n\t\t================================================================================================\n\t\t\t\t\tINSTRUCTIONS TO BE FOLLOWED IN THE MATCH\n\t\t================================================================================================\n\n");

printf("0:Dot Ball\n1:Single Run\n2:Two Runs\n3:Three Runs\n4:Four\n5:No Ball/Wide/LB/B+4\n6:Six\n7;No Ball+6\n8:Wicket\n");

break;

/\* \*\*\*\*\*\*\*\*check whether\*\*\*\*\*\*\*\* \*/

case 2:

system("cls");

printf("\n\t\t================================================================================================\n\t\t\t\t\tCHECK WHEATHER\n\t\t================================================================================================\n\n");

float Temp;

printf("ENTER THE TEMPERATURE:\n");

scanf("%f",&Temp);

if(Temp<10)

{

printf("CANNOT PLAY\n");

exit(0);

}

else

{

printf("GOOD LUCK FOR THE MATCH\n");

}

break;

/\* \*\*\*\*\*\*selection of teams\*\*\*\*\*\*\* \*/

case 3:

system("cls");

printf("\n\n\t ..::SELECTION OF TEAMS..:: \n\t===========================\n\t");

char t1[10];

char t2[10];

char filename[15];

printf("\n ENTER THE TEAMS PARTICIPATING:\n");

printf("ENTER THE FIRST TEAM\n");

scanf("%s",t1);

printf("ENTER THE SECOND TEAM\n");

scanf("%s",t2);

/\* \*\*\*\*\*\*\*\*MAIN MATCH\*\*\*\*\*\*\*\*\*/

FILE \*fp;

printf("Enter the filename to be opened with date(match 12-07-20)\n");

scanf("%s",filename);

fp = fopen(filename,"ab+");

//int b1=1;

//int b2=1;

//int sum1=0;

//int sum2=0;

//int w1=0;

//int w2=0;

int r1;

int r2;

SetColor(17);

if(fp == NULL){

MessageBox(0,"Error in Opening file\nMake sure your file is not write protected","Warning",0);

}else{

printf("%s innings:\n",t1);

fprintf(fp,"%s innings:\n",t1);

while(b1<=120)

{

printf("status of ball %d:",b1);

fprintf(fp,"status of ball %d:",b1);

scanf("%d",&r1);

fprintf(fp,"%d\n",r1);

if(r1==8)

{

w1++;

if(w1<=9)

{

printf("%s loses a wicket,%d wickets remaining\n",t1,(10-w1)); fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

}

if(w1==10)

{

printf("%s innings came to an end\n",t1);

break;

}

}

else if(r1>8)

{

printf("invalid no.\n please read the instructions carefully \n");

continue;

}

else if(r1==7)

{

printf("free hit so no wicket will be considered even if wicket falls\n");

b1--;

sum1=sum1+r1;

}

else

{

sum1=sum1+r1;

}

b1++;

}

}

printf("%s requires %d runs to win in 120 balls at a run rate of %d\n",t2,sum1+1,(sum1+1)/20);

fprintf(fp,"%s requires %d runs to win in 120 balls at a run rate of %d\n",t2,sum1+1,(sum1+1)/20);

printf("%s innings:\n",t2);

fprintf(fp,"%s innings:\n",t2);

while(b2<=120 && sum2<=sum1)

{

printf("status of ball %d:",b2);

fprintf(fp,"status of ball %d:",b2);

scanf("%d",&r2);

fprintf(fp,"%d\n",r2);

if(r2==8)

{

w2++;

if(w2<=9)

{

printf("%s loses a wicket,%d wickets remaining\n",t2,(10-w2));

}

if(w2==10)

{

printf("%s innings came to an end\n",t2);

break;

}

}

else if(r2>8)

{

printf("invalid no.\n please read the instructions carefully\n");

continue;

}

else if(r2==7)

{

printf("free hit so no wicket will be considered even if wicket falls\n");

b2--;

sum2=sum2+r2;

}

else

{

sum2=sum2+r2;

}

b2++;

}

printf("%s total is %d\n",t1,sum1);

fprintf(fp,"%s total is %d\n",t1,sum1);

printf("%s total is %d\n",t2,sum2);

fprintf(fp,"%s total is %d\n",t2,sum2);

if(sum1>sum2)

{

printf("%s won the match over %s by %d wickets in %d balls\n",t1,t2,w1,b1);

fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t1,t2,w1,b1);

}

if(sum2>sum1)

{

printf("%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

}

SetColor(17);

fclose(fp); fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2); fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

a: if(sum2==sum1)

{

FILE \*fp;

char filename[15];

printf("Enter the filename to be opened with date(match 12-07-20)\n");

scanf("%s", filename);

fp = fopen(filename,"ab+");

int b1=1;

int b2=1;

int sum1=0;

int sum2=0;

int w1=0;

int w2=0;

int r1;

int r2;

SetColor(17);

if(fp == NULL){

MessageBox(0,"Error in Opening file\nMake sure your file is not write protected","Warning",0);

}else

{

printf("%s innings:\n",t1);

fprintf(fp,"%s innings:\n",t1);

printf("Let's Begin Super Over\n");

fprintf(fp,"Let's Begin Super Over\n");

printf("%s innings\n",t1);

while(b1<=6)

{

printf("status of ball %d:",b1);

fprintf(fp,"status of ball %d:",b1);

scanf("%d",&r1);

fprintf(fp,"%d\n",r1);

if(r1==8)

{

w1++;

if(w1==1)

{

printf("%s loses a wicket,one wicket remaining\n",t1);

}

if(w1==2)

{

printf("%s innings has came to an end\n",t1);

printf("%s requires %d runs to win in 6 balls at a run rate of %d\n",t2,sum1+1,sum1+1);

break;

}

}

else if(r1>8)

{

printf("invalid no.\n please read the instructions carefully\n");

continue;

}

else if(r1==7)

{

printf("free hit\n");

b1--;

sum1=sum1+r1;

}

else

{

sum1=sum1+r1;

}

b1++;

}

}

b1=b1-1;

fprintf(fp,"%s requires %d runs to win in 6 balls at a run rate of %d\n",t2,sum1+1,(sum1+1)/20);

printf("%s innings\n",t2);

fprintf(fp,"%s innings:\n",t2);

while(b2<=6 && sum2<=sum1)

{

printf("status of ball %d:",b2);

fprintf(fp,"status of ball %d:",b2);

scanf("%d",&r2);

fprintf(fp,"%d\n",r2);

if(r2==8)

{ w2++;

if(w2==1)

{

printf("%s loses a wicket, one wicket remaining\n",t2);

}

if(w2==2)

{

printf("%s innings has come to an end\n",t2);

break;

}

}

else if(r2>8)

{

printf("invalid no.\nplease carefully read the instructions\n");

continue;

}

else if(r2==7)

{

printf("free hit so no wicket will be considered even if wicket falls.Hence do not enter 8\n");

b2--;

sum2=sum2+r2;

}

else

{

sum2=sum2+r2;

}

b2++;

}

b2=b2-1;

printf("A total: %d\n",sum1);

fprintf(fp,"%s total is %d\n",t1,sum1);

printf("B total:%d\n",sum2);

fprintf(fp,"%s total is %d\n",t2,sum2);

if(sum1>sum2)

{

printf("%s won over %s by %d wickets in %d balls\n",t1,t2,w1,b1);

fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t1,t2,w1,b1);

}

if(sum1<sum2)

{

printf("%s won over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

fprintf(fp,"%s won the match over %s by %d wickets in %d balls\n",t2,t1,w2,b2);

}

if(sum1==sum2)

{

goto a;

}

fclose(fp);

break;

}

case 4:

system("cls");

printf("\n\t\t================================================================================================\n\t\t\t\t\tPREVIOUS MATCH DETAILS\n\t\t================================================================================================\n\n");

FILE \*fptr;

char filename1[100];

char c;

printf("Enter the filename to open \n");

scanf("%s", filename1);

// Open file

fptr= fopen(filename1, "r");

if (fptr == NULL)

{

printf("Cannot open file \n");

exit(0);

}

// Read contents from file

c = fgetc(fptr);

while (c != EOF)

{

printf ("%c", c);

c = fgetc(fptr);

}

fclose(fptr);

break;

/\* \*\*\*\*\*\*\*\*BAR GRAPH OF TWO TEAMS\*\*\*\*\*\*\*\* \*/

case 5:

system("cls");

printf("\n\t\t================================================================================================\n\t\t\t\t\tBAR GRAPH\n\t\t================================================================================================\n\n");

// gm is Graphics mode which is

// a computer display mode that

// generates image using pixels.

// DETECT is a macro defined in

// "graphics.h" header file

int gd = DETECT, gm;

// initgraph initializes the

// graphics system by loading a

// graphics driver from disk

initgraph(&gd, &gm, "");

settextstyle(10,1,3);

// location of sides

int left, top, right, bottom;

// left, top, right, bottom denotes

// location of rectangular bar

bar(left = 150, top = 350-sum1,

right = 190, bottom = 350);

outtextxy(192,370,t1);

bar(left = 260, top = 350-sum2,

right = 300, bottom = 350);

outtextxy(302,370,t2);

//bar(left = 290, top = 200,

//right = 330, bottom = 350);

// y axis line

SetColor(45);

outtextxy(90,70,"SCORES");

line(100, 20, 100, 350);

settextstyle(10,0,3);

// x axis line

line(100, 350, 500, 350);

outtextxy(490,390,"INNINGS");

getch();

// closegraph function closes the

// graphics mode and deallocates

// all memory allocated by

// graphics system .

closegraph();

break;

/\*default:

printf("INVALID CHOICE");

break;\*/

}

printf("\n\n\n..::ENTER THE CHOICE:\n\n\t[1] MAIN MENU\t\t[0] EXIT\n");

scanf("%d",&ch);

switch (ch)

{

case 1:

goto main;

case 0:

break;

default:

printf("INVALID CHOICE");

break;

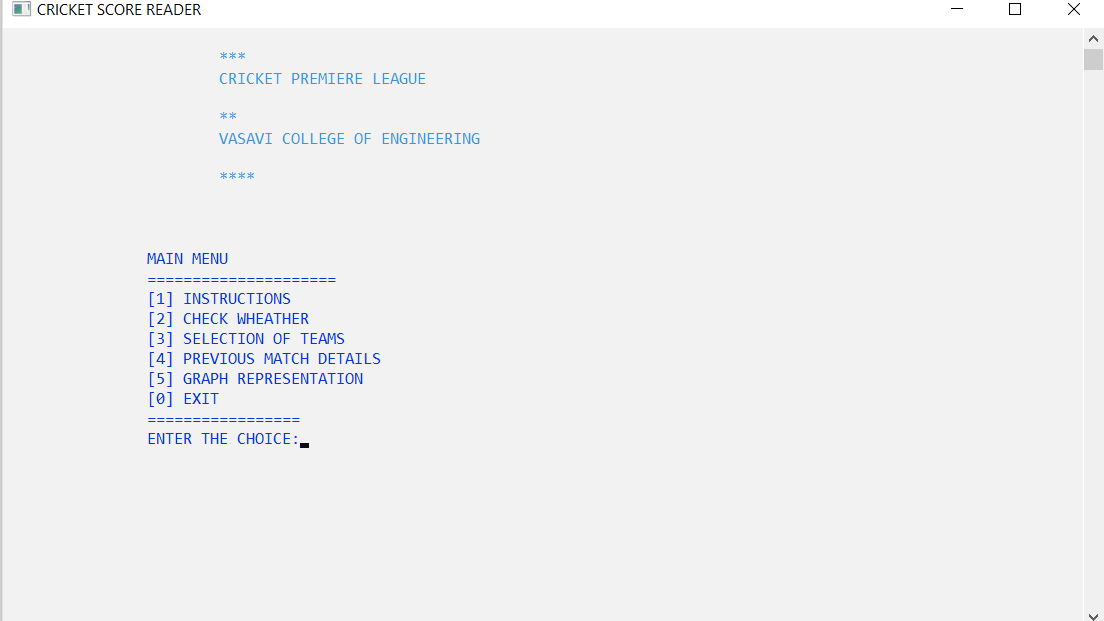
}

return 0;

}

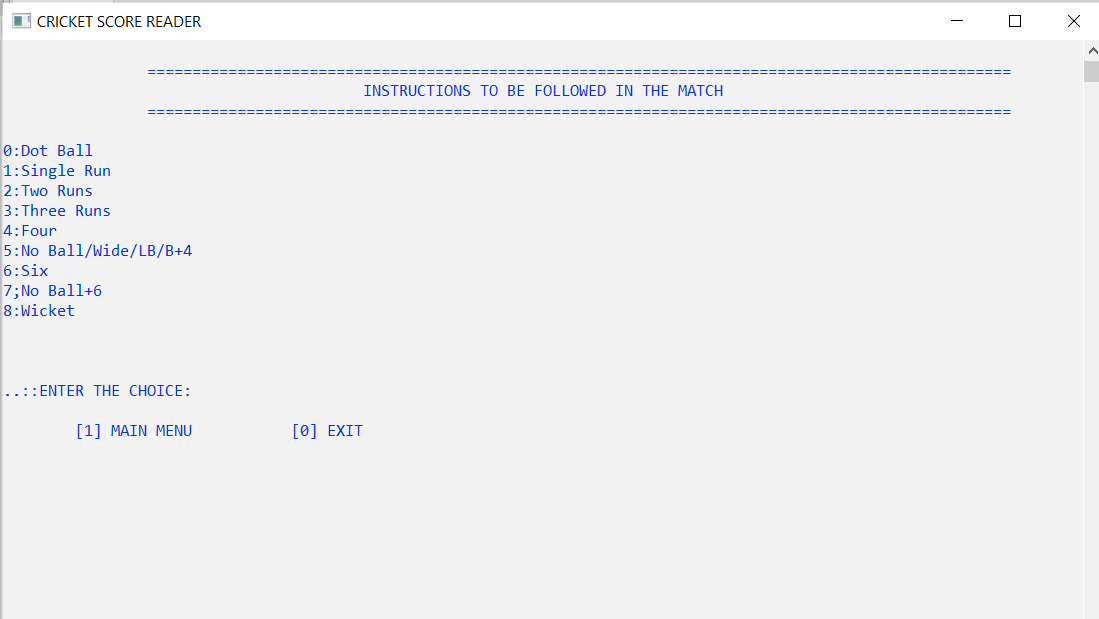
**4.2 Output Screens:**

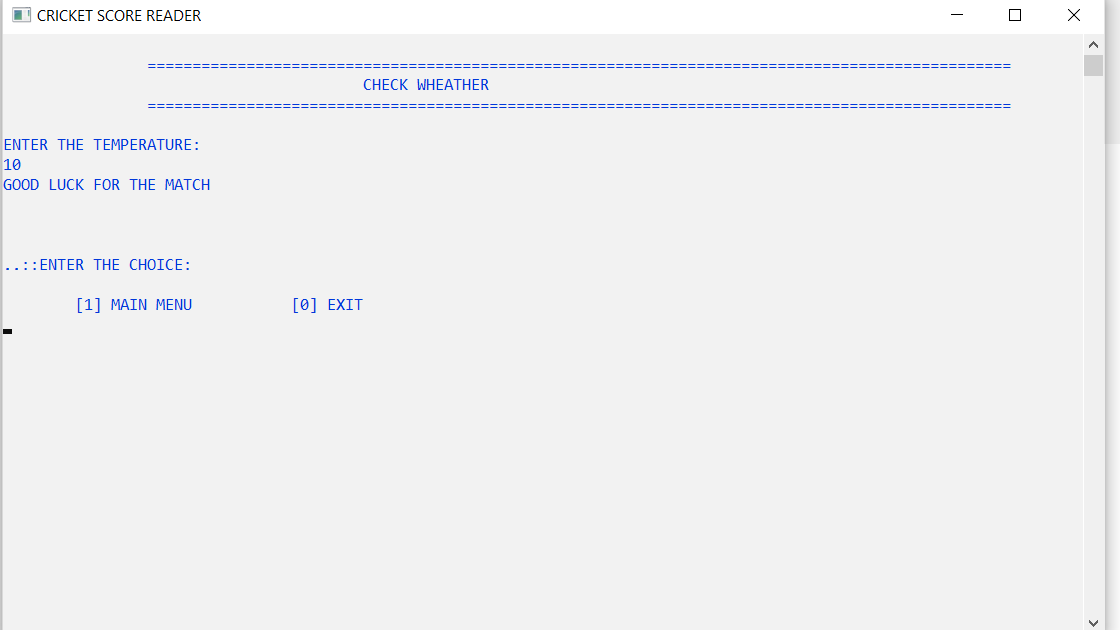
**Menu bar:**

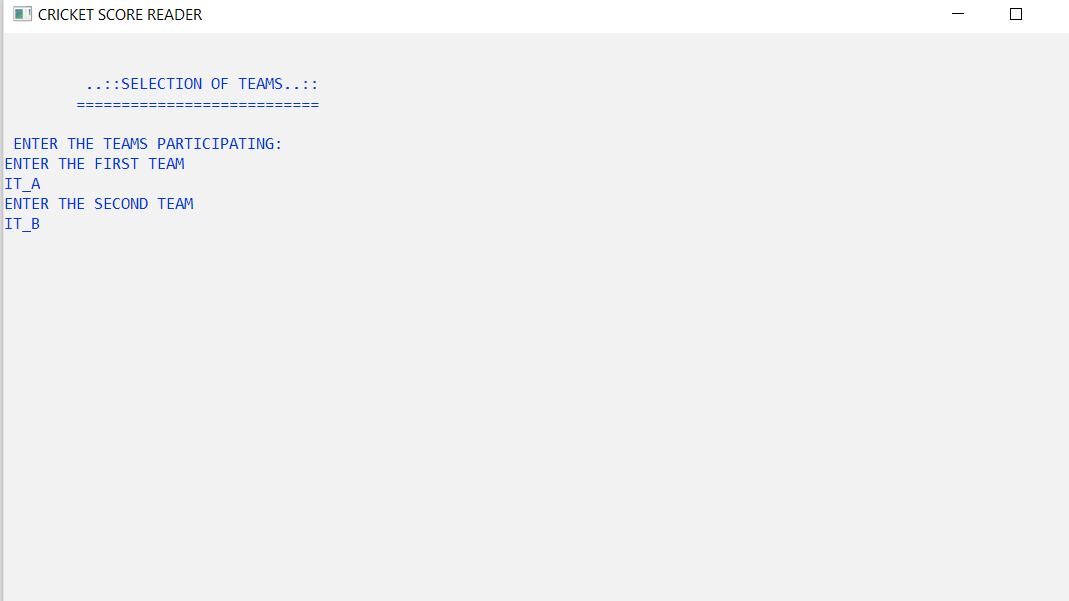
****

**Fig**-**4.1.2.a-Menu Bar**

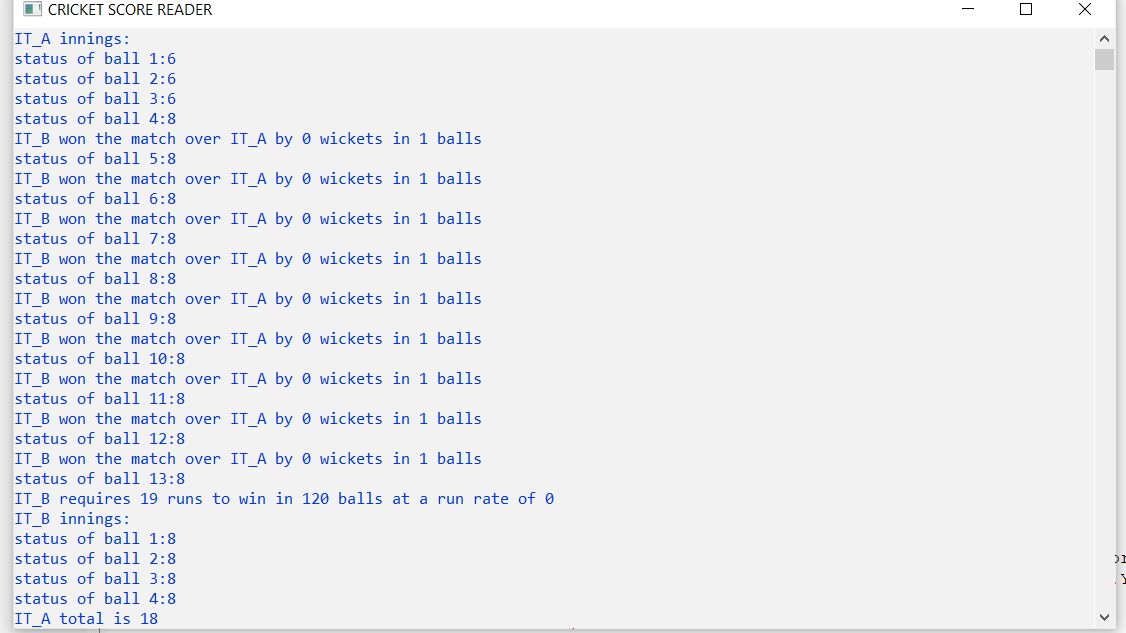
**Instructions followed in the match**

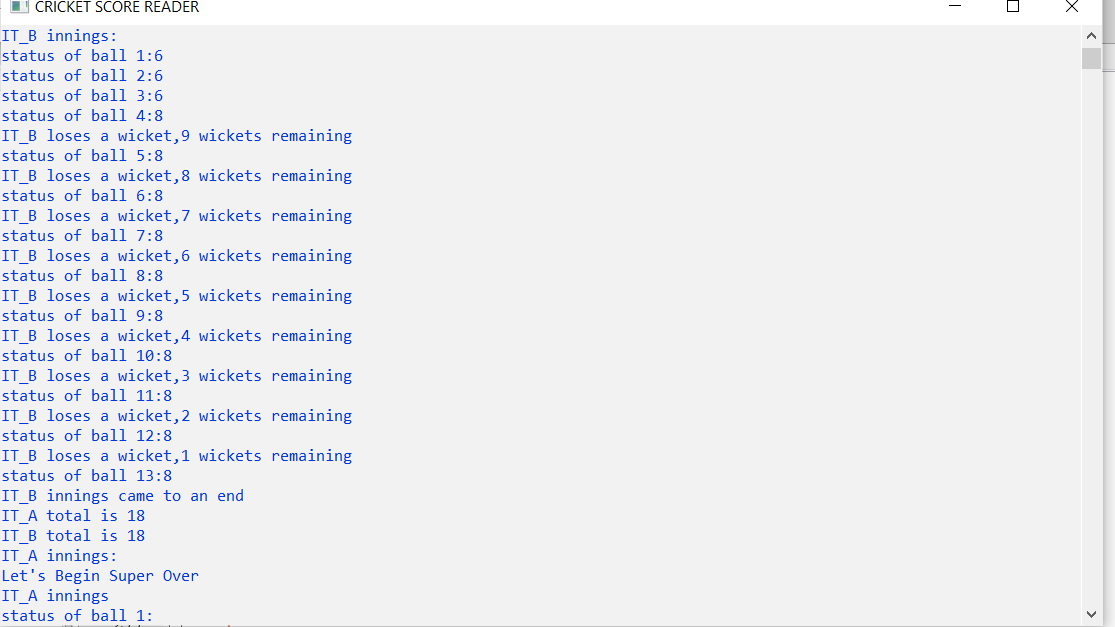
****

**Check weather :**

**Selection of Teams:**

**Main Match:**

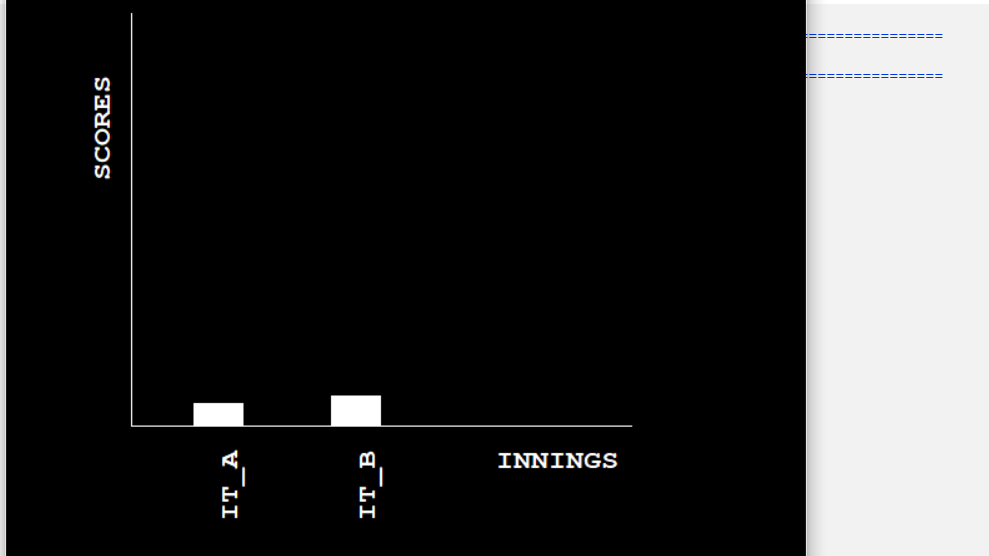
****

**Super-Over:**

**Previous Match Details:**

****

**Graph Representation:**

****

## 4.1.3 Result Analysis:

## Hence, we can observe the calculations occurred and the score is displayed here.

## This enables us to easily understand the cricket game and it enables to calculate the score, score board and analysis of teams using bargraphs.

## Future Enhancements:

* The future of our system might change the tournament level cricket

competitions.

* The cricket score calculations can be made easier without any waste of time.
* This system is easily available and its scope and lifetime is high

**4.1.5 Conclusion:**

* Our project and implementation is on **Cricket Score board.** We have successfully completed it. We take this opportunity to express our sense of indebtedness and gratitude to all those people who helped us in completing the project and implementation.
* We are immensely grateful to our esteemed project guidance without which this work would not have been possible.
* This project and implementation has contributed a lot to our knowledge that has proved to be a value addition for us.