CERTIFICATE

This is to certify that Aritro Dutta a bonafide student of Class XII-J has successfully completed the project titled “Human V/S Bot Cricket Match” in the Computer Lab during the Year 2021 – 2022 for the A.I.S.S.C.E Computer Science Practical Term-2 Examination -2022.

It is further certified that this project is the individual work of the Candidate.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner Internal Examiner

DATE: \_\_\_\_\_\_\_\_

SEAL: \_\_\_\_\_\_\_\_

Acknowledgements

I gratefully acknowledge my sincere thanks to our Computer Science Teacher Mr. Sudarshan Kumar Manna for his remarkable, valuable guidance and supervision throughout the project work. I’m also utmost indebted to all my batch mates for their encouragement, help, suggestion and readily helpful service in the successful completion of the project.

I wish to express my deep gratitude and sincere thanks to the Principal, Mrs. Joyoti Chaudhuri, Delhi Public School, School, Ruby Park for her encouragement and for all the facilities that she provided for this project work.

Aritro Dutta

INDEX

1. Certificate

2. Acknowledgment

3. Hardware and Software Requirements

4. Objectives of the Project

5. Project Design

6. Database Design and Source Code

7. Modules and Function Documentation

8. Source Code of Modules

9. Output Screens

10. Limitations

11. Bibliography & References

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

Processor: Intel® Pentium® CPU G4560 3.5 GHz

Memory: 4GB RAM

Hard Disk:

SOFTWARE REQUIREMENTS

Operating System: Microsoft Windows 10 Pro

Programming IDE: Spyder (Python 3.9), MySQL

OBJECTIVES OF THE PROJECT

This project aims to create a Human vs Bot T20 Cricket game in which a Human team will play against the Bot team. It stores and manages data of the teams. It will display the scoreboard to the user. It is a simulation of a live cricket match played between two teams. In this project, Team-A represents Human beings and Team-B represents Bots. Based on the final total batting score, the Winner of the match is decided.

PROJECT DESIGN

HUMAN V/S BOT CRICKET MATCH

CREATE TEAM-B

CREATE TEAM-A

SHOW TEAM-B

SHOW TEAM-A

START MATCH

SCOREBOARD TEAM-B

SCOREBOARD TEAM-A

FOURS

FOURS

SIXES

RUNS

SIXES

RUNS

TOTAL RUNS TEAM-B

TOTAL RUNS TEAM-A

SCORE TEAM-A V/S SCORE TEAM-B

NO

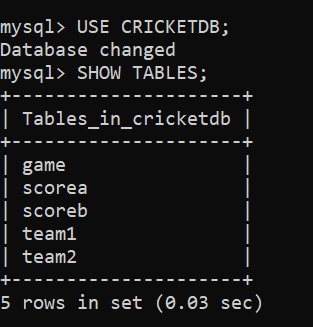
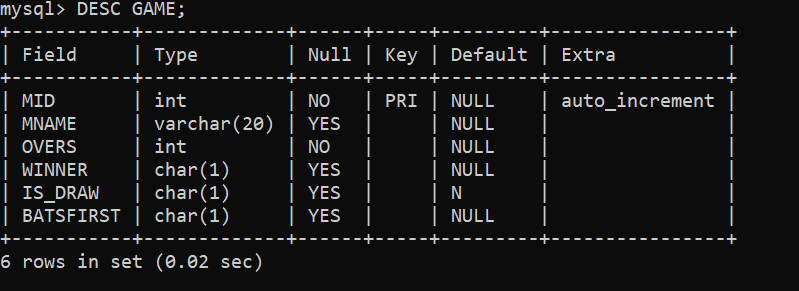
YES

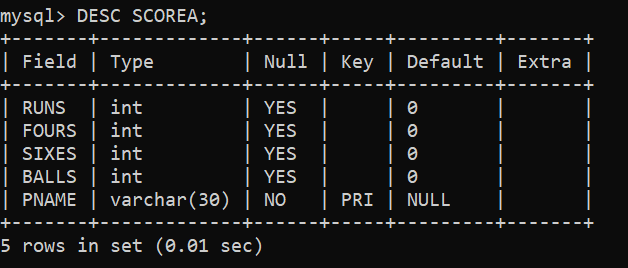
IS GREATER

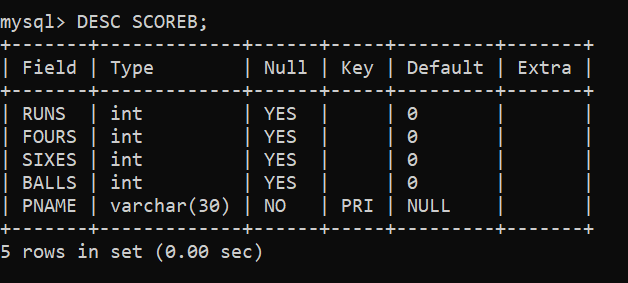
BOT TEAM IS WINNER

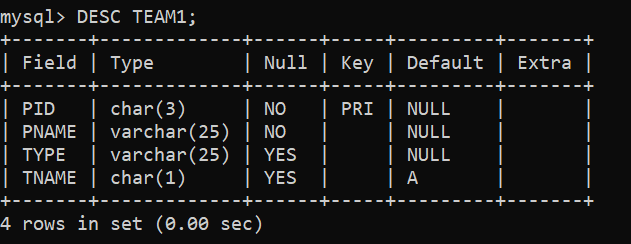
HUMAN TEAM IS WINNER

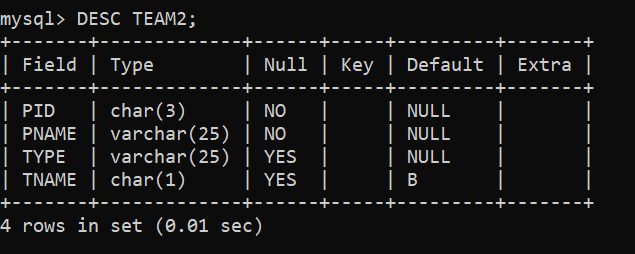
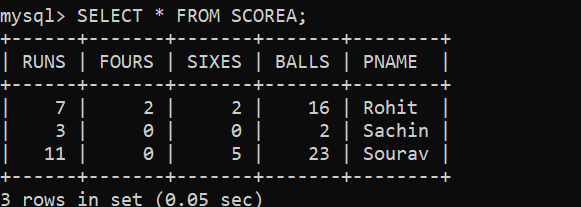
DATABASE DESIGN

TABLE STRUCTURE







For Game Table:

mysql> CREATE TABLE GAME (MID INT PRIMARY KEY AUTO\_INCREMENT, MNAME VARCHAR(20), OVERS INT NOT NULL, WINNER CHAR(1), IS\_DRAW CHAR(1), BATSFIRST CHAR(1));

For ScoreA Table:

mysql> CREATE TABLE SCOREA (RUNS INT, FOURS INT, SIXES INT, BALLS INT, PNAME VARCHAR(30) PRIMARY KEY);

For ScoreB Table:

mysql> CREATE TABLE SCOREB (RUNS INT, FOURS INT, SIXES INT, BALLS INT, PNAME VARCHAR(30) PRIMARY KEY);

For Team1 Table:

mysql> CREATE TABLE TEAM1 (PID CHAR(3) PRIMARY KEY, PNAME VARCHAR(25) NOT NULL, TYPE VARCHAR(25), TNAME CHAR(1) DEFAULT A);

For Team2 Table:

mysql> CREATE TABLE TEAM2 (PID CHAR(3) PRIMARY KEY, PNAME VARCHAR(25) NOT NULL, TYPE VARCHAR(25), TNAME CHAR(1) DEFAULT B);

MODULE DOCUMENTATION

MODULE NAME: random

DESCRIPTION: This module implements pseudo-random number generators for various distributions.

FUNCTIONS:

choice(seq): Choose a random element from a non-empty sequence.

randint(a, b): Returns random integer in range [a, b], including both end points.

random(): Returns a random number in the interval [0, 1).

randrange(start, stop=None, step=1): Choose a random item from range(start, stop[, step]).

MODULE NAME: mysql.connector

DESCRIPTION: This module provides top-level methods and properties.

FUNCTIONS:

connect(\*args, \*\*kwargs): Create or get a MySQL connection object.

custom\_error\_exception(error=None, exception=None): Define custom exceptions for MySQL server errors

MODULE NAME: prettytable

DESCRIPTION: This module is used to create relational tables in Python.

FUNCTIONS:

from\_html(html\_code, \*\*kwargs): Generates a list of PrettyTables from a string of HTML code. Each <table> in the HTML becomes one PrettyTable object.

from\_html\_one(html\_code, \*\*kwargs): Generates a PrettyTables from a string of HTML code which contains only a single <table>.

SOURCE CODE

File Name: game.py

import random as rd

import mysql.connector as mydb

import prettytable as pt

cnn=mydb.connect(host='localhost',user='root',password='root',database='cricketdb')

if cnn.is\_connected():

print("Connection Successful")

else:

print("Connection Failed")

def toss():

ch=input("Your choice <h-head | t-tail>:")

coin=rd.choice(["h","t"])

if ch==coin:

print("Human Team has won the toss")

return "A"

else:

print("Bot Team has won the toss")

return "B"

def clearscore():

cur=cnn.cursor()

strsql='delete from scorea'

cur.execute(strsql)

cnn.commit()

strsql='delete from scoreb'

cur.execute(strsql)

cnn.commit()

cur.close()

def addscore(row,team):

if team=="A":

tname='scorea'

else:

tname='scoreb'

strsql='insert into'+tname+'(pname,runs,fours,sixes,balls) values(%s,%s,%s,%s,%s)'

cur=cnn.cursor()

cur.execute(strsql,row)

cnn.commit()

cur.close()

def gameplay(batsmen,overs,result):

count=len(batsmen)

cnt=0

fours=sixes=runs=balls=0

allout=False

totalruns=0

batsman=batsmen[cnt]

print("Batsman:",batsmen[cnt])

for over in range(1,overs+1):

print("Over:",over)

for ball in range(1,7):

run=rd.choice([0,1,2,3,4,6,7,8,9,-1])

balls+=1

if run==4:

fours+=1

totalruns+=run

print("Hit a FOUR")

elif run==6:

sixes+=1

totalruns+=run

print("Hit a SIX")

elif run==1 or run==2 or run==3:

runs+=run

print("Run scored",run)

totalruns+=run

elif run==0:

print("No run scored")

if run==-1:

print("Batsman OUT",batsman)

print("No of Fours:",fours)

print("No of Sixes:",sixes)

print("Total Scored:",totalruns)

cnt+=1

row=(batsman,runs,fours,sixes,balls)

addscore(row,result)

fours=sixes=runs=balls=0

totalruns=0

if cnt>=count:

allout=True

break

else:

batsman=batsmen[cnt]

print("Next Batsman:",batsmen[cnt])

if allout:

print("All Players Out")

break

if not allout:

print("Batsman NOT OUT:",batsman)

row=(batsman,runs,fours,sixes,balls)

addscore(row,result)

def start():

result=toss()

clearscore()

if result=="A":

team="team1"

print("Human Team will bat first")

else:

team="team2"

print("Bot Team will bat first")

cur=cnn.cursor()

mname=input("Enter Match Name:")

overs=int(input("Enter No of overs:"))

strsql="insert into game(mname,overs,batsfirst) values('{}',{},'{}')".format(mname,overs,result)

cur.execute(strsql)

cnn.commit()

strsql='select mid from game order by mid desc limit 1'

cur.execute(strsql)

row=cur.fetchone()

matchid=row[0]

#players of team-A

strsql='select \* from team1'

cur.execute(strsql)

playersA=[]

rows=cur.fetchall()

for row in rows:

playersA.append(row[1])

print(playersA)

#players of team-B

strsql='select \* from team2'

cur.execute(strsql)

playersB=[]

rows=cur.fetchall()

for row in rows:

playersB.append(row[1])

print(playersB)

if result=="A":

batsmen1=playersA

batsmen2=playersB

else:

batsmen1=playersB

batsmen2=playersA

gameplay(batsmen1,overs,result)

if result=="A":

print("Human Team batting over")

nextbat="BOT Team will bat now"

else:

team="team2"

print("Bot Team batting over")

nextbat="Human Team will bat now"

print("\*"\*100)

print(nextbat.center(100))

print("\*"\*100)

ch=input("Start Match <y/n>:")

if result=="A":

gameplay(batsmen2,overs,"B")

else:

gameplay(batsmen2,overs,"A")

def scoreboard(team):

cur=cnn.cursor()

#PID,PName,Type,TName

if team=="A":

strsql='select pname,fours,sixes,balls,runs+fours\*4+sixes\*6 as totruns from scorea'

else:

strsql='select pname,fours,sixes,balls,runs+fours\*4+sixes\*6 as totruns from scoreb'

cur.execute(strsql)

rows=cur.fetchall()

count=cur.rowcount

rec=[]

print("Team:"+team)

totalruns=0

totalballs=0

tbl=pt.PrettyTable(["Player Name","Fours","Sixes","Balls Faced","Runs Scored"])

for row in rows:

tbl.add\_row([row[0],row[1],row[2],row[3],row[4]])

totalruns+=int(row[4])

totalballs+=int(row[3])

print(tbl)

overs=totalballs//6

print("="\*60)

score="Final Score {} /{}".format(totalruns,count)

print(score.center(60))

print("Total Overs Played {}".format(overs).center(60))

print("="\*60)

cur.close()

def finalresult():

cur=cnn.cursor()

strsql='select count(pname) as pcount,sum(balls) as balls,sum(runs+fours\*4+sixes\*6) as totruns from scorea'

cur.execute(strsql)

row=cur.fetchone()

runs=int(row[2])

count=int(row[0])

balls=int(row[1]%6)

overs=int(row[1])//6

print("\*"\*60)

scorea=int(row[2])

print("Final Score Board".center(60))

print()

print("Team-A Human Team Scoreboard".center(60))

if balls>0:

score="Scored {} runs in {} overs and {} balls - Wickets out {}".format(runs,overs,balls,count)

else:

score="Scored {} runs in {} overs - Wickets out {}".format(runs,overs,count)

print(score)

print()

strsql='select count(pname) as pcount,sum(balls) as balls,sum(runs+fours\*4+sixes\*6) as totruns from scoreb'

cur.execute(strsql)

row=cur.fetchone()

runs=int(row[2])

balls=int(row[1])

overs=int(row[1])//6

print("Team-B BOT Team Scoreboard".center(60))

if balls>0:

score="Scored {} runs in {} overs and {} balls - Wickets out {}".format(runs,overs,balls,count)

else:

score="Scored {} runs in {} overs - Wickets out {}".format(runs,overs,count)

print(score)

print()

scoreb=int(row[2])

print()

if scorea>scoreb:

print("Team Human Won - Team Bot Lost".center(60))

else:

print("Team BOT Won - Team Human Lost".center(60))

print("\*"\*60)

cur.close()

def menu():

print("1. Create Team-A")

print("2. Create Team-B")

print("3. Show Team-A")

print("4. Show Team-B")

print("5. Start Game")

print("6. Score Board")

print("7. Final Result")

print("8. Exit")

def create(team):

cur=cnn.cursor()

if team=="A":

strsql='delete from team1'

else:

strsql='delete from team2'

cur.execute(strsql)

cnn.commit()

#PID, PName, Type, TName

n=int(input("Enter no of players:"))

for i in range(1,n+1):

pid=team+str(i)

pname=input("Enter player Name:")

print("Select Type 1. Batsman 2. Bowler 3. All Rounder")

opt=int(input("Select Type <1-3>:"))

if opt==1:

ptype="Batsman"

elif opt==2:

ptype="Bowler"

else:

ptype="All Rounder"

if team=="A":

strsql="insert into team1 values('{}','{}','{}','{}')".format(pid,pname,ptype,"A")

else:

strsql="insert into team2 values('{}','{}','{}','{}')".format(pid,pname,ptype,"B")

cur.execute(strsql)

cnn.commit()

cur.close()

def show(team):

cur=cnn.cursor()

#PID, PName, Type, TName

if team=="A":

strsql='select \* from team1'

else:

strsql='select \* from team2'

cur.execute(strsql)

rows=cur.fetchall()

rec=[]

print("Team:"+team)

tbl=pt.PrettyTable(["Player ID","Player Name","Type"])

for row in rows:

tbl.add\_row([row[0],row[1],row[2]])

print(tbl)

cur.close()

while True:

menu()

ch=int(input("Enter your choice:"))

if ch==1:

create("A")

elif ch==2:

create("B")

elif ch==3:

show("A")

elif ch==4:

show("B")

elif ch==5:

start()

elif ch==6:

print("H - Human Team,B - Bot Team")

ch=input("Select Team Scoreboard to View <H|B>:").upper()

if ch=="H":

scoreboard("A")

else:

scoreboard("B")

elif ch==7:

finalresult()

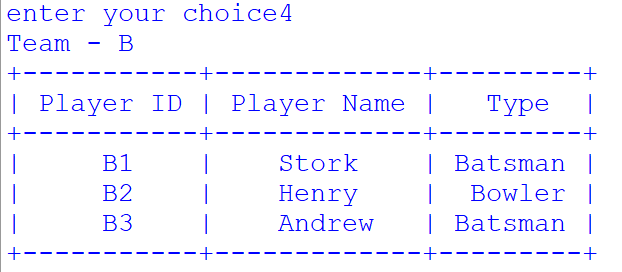
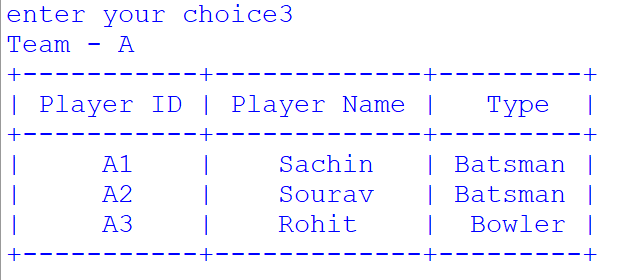
elif ch==8:

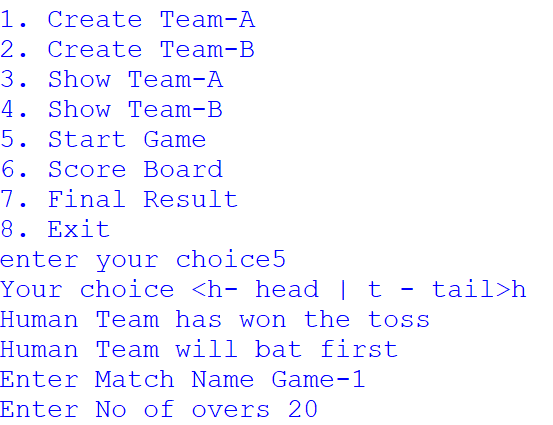
break

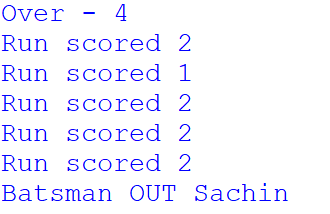
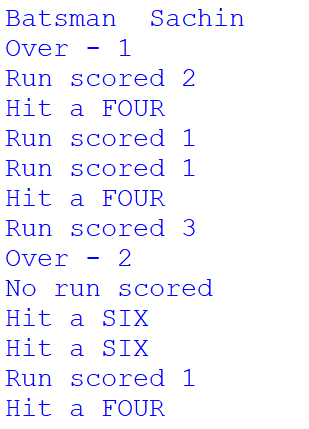
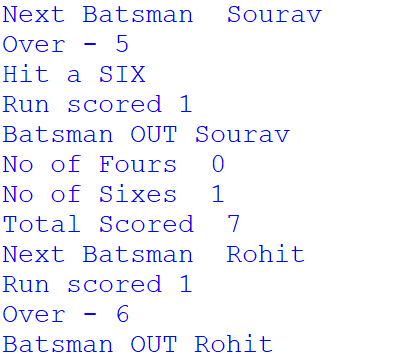
else:

print("Error - select correct option <1-5>")

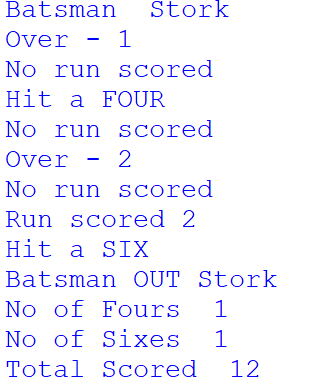
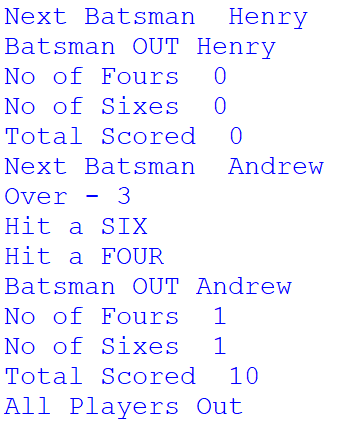
OUTPUT SCREEN

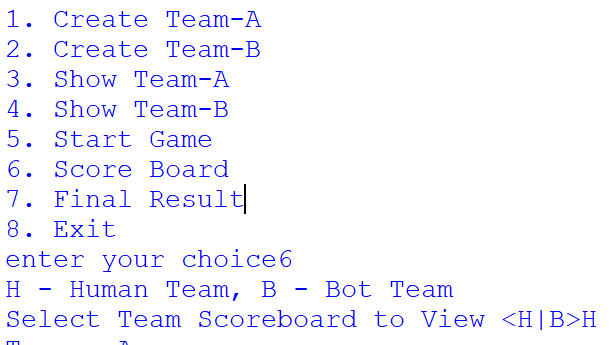
b bnncncnbc
ncncncn
bcbcb
vbbvb

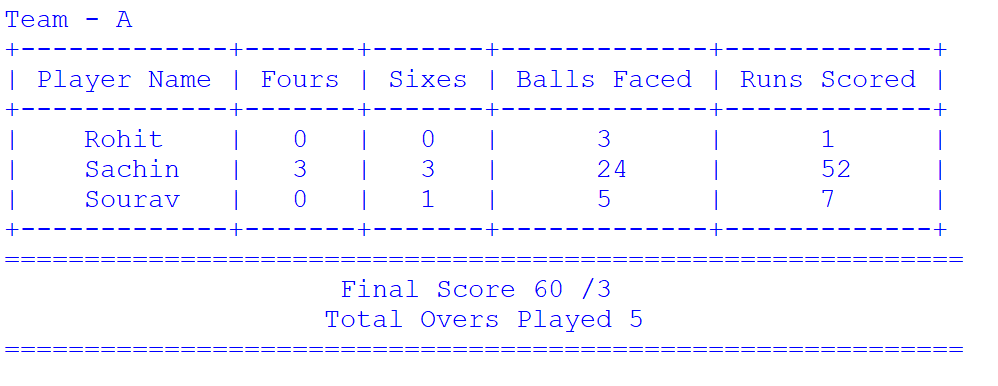


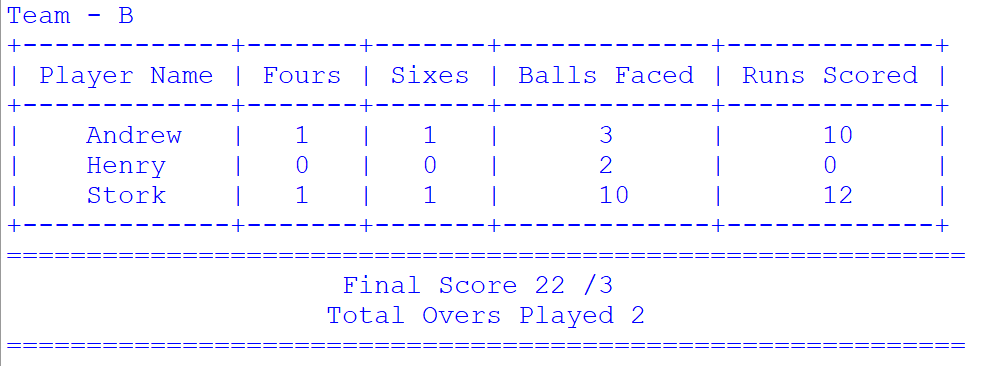












LIMITATIONS

1. It is a prototype of the real live cricket match scoreboard.
2. It cannot be played by two users at the same time.
3. It cannot be played online.

BIBLIOGRAPHY & REFERENCES

1. Computer Science for class XII by Sumita Arora.
2. <http://www.icc-cricket.com>