**Fantasy Cricket Game Application**

a Project Submitted by,

**Saundarya Karli**

to,

**Internshala**

**Table of contents**

* Introduction 3
* Requirements 4
* Features 5
* Description 6
* Code Snippet 7
* Conclusion 14

**Introduction**

This project is a desktop application i.e. a Fantasy Cricket Game Application in python which allows creating virtual team of cricket players and scoring points on how the players perform in different matches. This is a game where you create a virtual team of real cricket players and score points depending on how you’re chosen players perform in real life matches. To win a tournament, one must try and get maximum points.

**Requirements**

* Python 3
* PyQt5
* Sqlite Studio

**Features**

* You will be able to create your own dream team by selecting your favorite players.
* Team selection will be based on basic cricket rules:
* No more than 5 batsmen allowed
* No more than 5 bowlers allowed
* No more than 3 all-rounder allowed
* No more than 1 wicket-keeper allowed
* You can create a new team or open an existing team.
* You can save your dream team containing of 11 players, no more or less.
* You can evaluate your team in a particular match.
* Initially 1500 points will be available to select players.
* Players will be selected according to the categories- BAT, BOW, AR and WK.

**Description**

Points for each player will be calculated according to the following rules:

* Batting:
* 1 point for 2 runs scored
* Additional 5 points for half century
* Additional 10 points for century
* 2 points for strike rate between 80-100
* Additional 4 points for strike rate >100
* 1 point for each four and 2 points for each six
* Bowling
* 10 points for each wicket
* Additional 5 points for three wickets
* Additional 10 points for 5 wickets or more
* 4 points for economy rate between 3.4 - 4.5
* 7 points for economy rate between 2 – 3.5
* 10 points for economy rate < 2
* Fielding
* 10 points for each catch/stumping/run out

**Code Snippet**

* Menu Functions:

def menuFunction(self,action):

txt=(action.text())

if txt=='NEW TEAM':

self.bat=0

self.bow=0

self.ar=0

self.wk=0

self.avl=1500

self.used=0

self.list1.clear()

self.list2.clear()

#pop-up window

text, ok=QtWidgets.QInputDialog.getText(MainWindow, "New Team", "Enter name of team:")

if ok and text:

self.t7.setText(str(text))

self.show()

if txt=='SAVE TEAM':

#Making a list of players in list2 separated by commas.

count=self.list2.count()

list\_save=""

for i in range(count):

list\_save+=self.list2.item(i).text()

if i<count:

list\_save+=","

self.save\_team(self.t7.text(),list\_save,self.used)

if txt=='OPEN TEAM':

#Clear all present values.

self.bat=0

self.bow=0

self.ar=0

self.wk=0

self.avl=1500

self.used=0

self.list1.clear()

self.list2.clear()

self.show()

#Open a team.

self.open\_team()

if txt=='EVALUATE TEAM':

#Importing contents of eval\_team.py

from eval\_team import Ui\_Dialog

Dialog = QtWidgets.QDialog()

ui = Ui\_Dialog()

ui.setupUi(Dialog)

result=Dialog.exec()

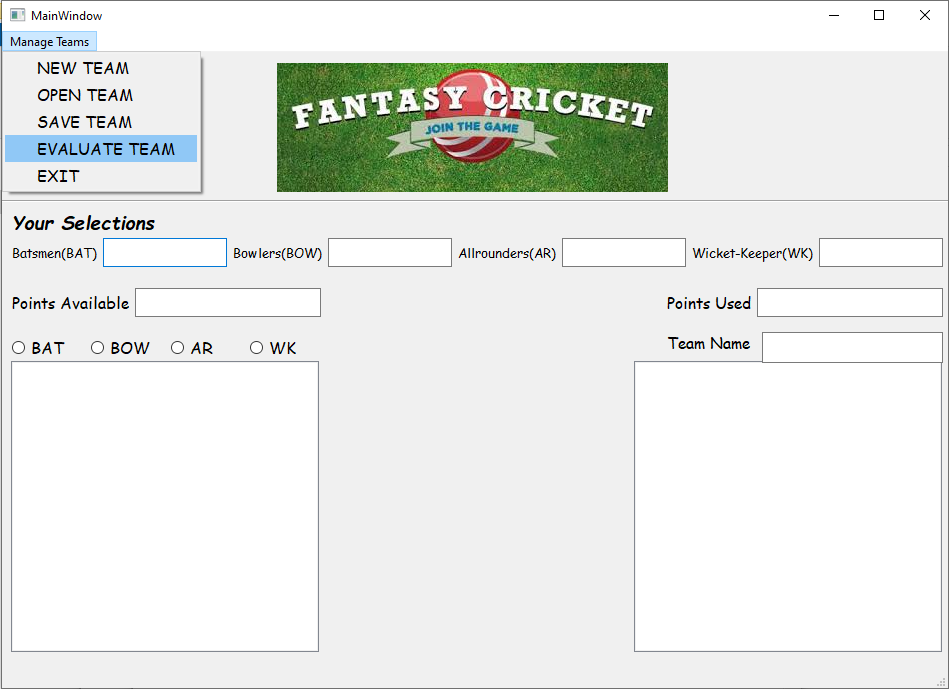
if txt=='EXIT':

#Exiting the application.

self.showdlg("Hope you enjoyed!!!\nVisit Again!!")

import sys

sys.exit()



* Open team:

def open\_team(self):

sql="select name from teams;"

curcricket=Mycricket.execute(sql)

teams=[]

for row in curcricket:

teams.append(row[0])

#Pop-up window showing a list of team to be selected.

team, ok=QtWidgets.QInputDialog.getItem(MainWindow,"Open Team","Select Team",teams,0,False)

if ok and team:

self.t7.setText(team)

#Fetching list of players from database to display in list2.

res="SELECT players,tot\_value from teams where name='"+team+"';"

curcricket=Mycricket.execute(res)

row=curcricket.fetchone()

list\_open=row[0].split(',')

self.list2.addItems(list\_open)

self.used=row[1]

self.avl=1500-row[1]

count=self.list2.count()

#Setting appropriate values for lineEdits.

for i in range(count-1):

player=self.list2.item(i).text()

sql="select ctg from stats where player='"+player+"';"

curcricket=Mycricket.execute(sql)

row=curcricket.fetchone()

ctgr=row[0]

if ctgr=="BAT":

self.bat+=1

if ctgr=="BOW":

self.bow+=1

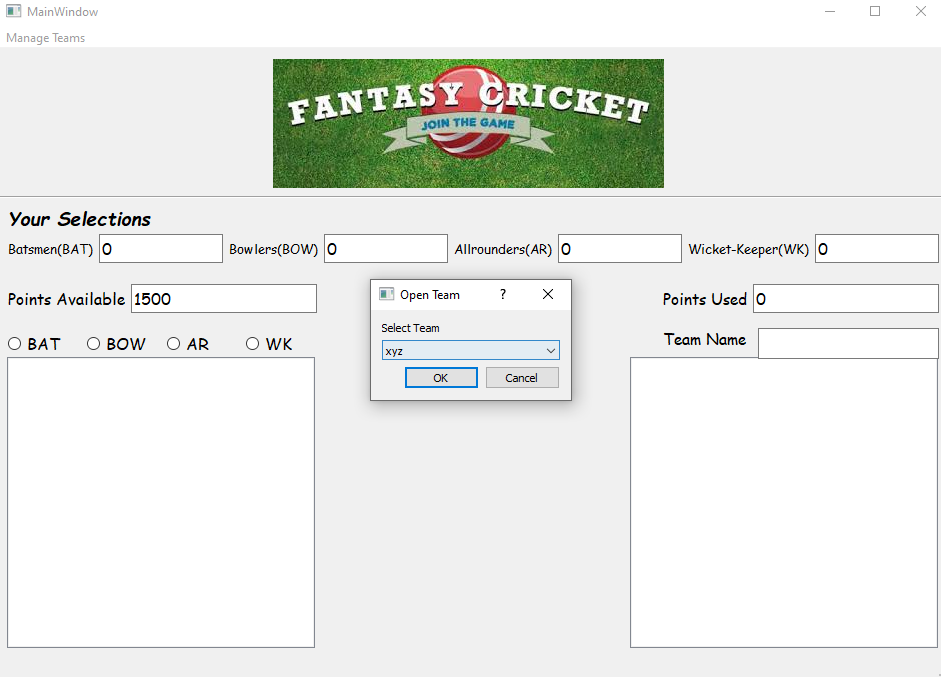
if ctgr=="AR":

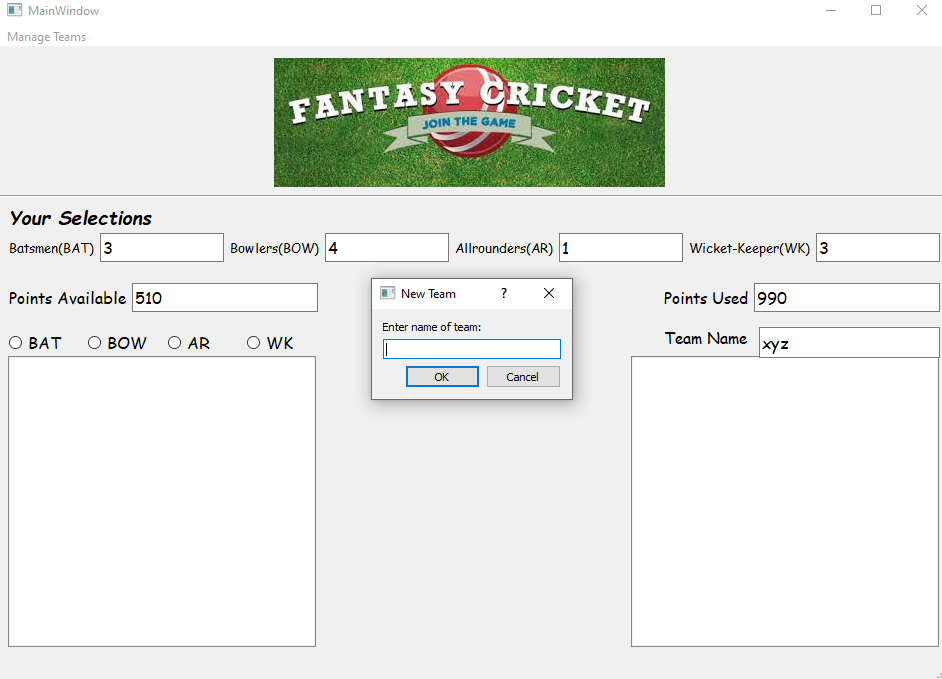
self.ar+=1

if ctgr=="WK":

self.wk+=1

self.show()





* Save Team:

#Setting conditions for players: not > or < 11.

count=self.list2.count()

if count!=11:

self.showdlg("No. of Players should be equal to 11!!")

return

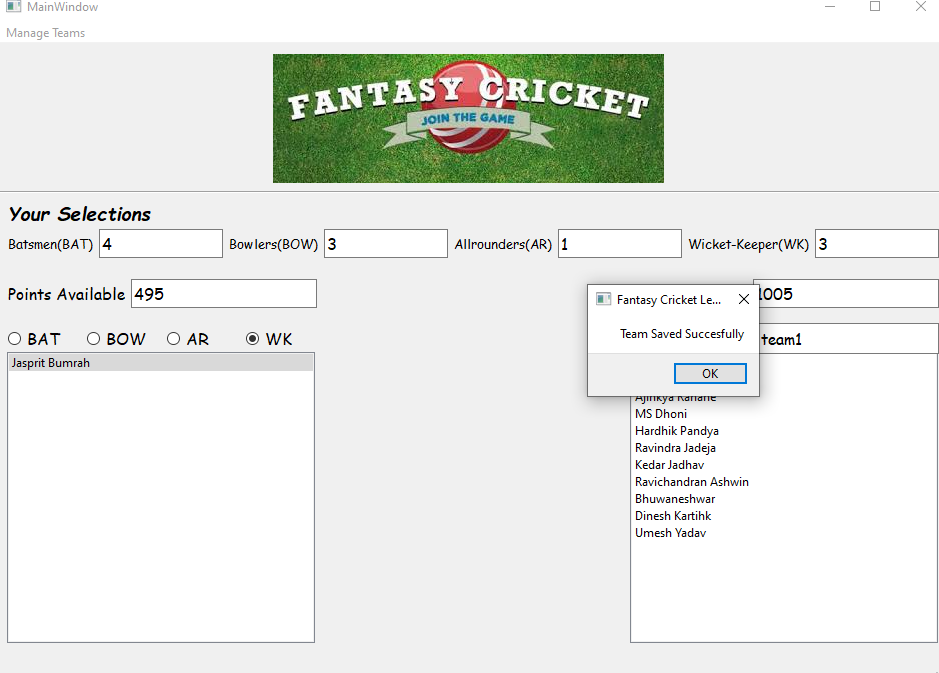
#Inserting name of team,players list and their total value in database.

sql="INSERT INTO teams (name,players,tot\_value) VALUES ('"+nm+"','"+ply+"','"+str(val)+"');"

curcricket=Mycricket.execute(sql)

self.showdlg("Team Saved Succesfully")

Mycricket.commit()



* Evaluate Team:

def display(self):

import sqlite3

conn=sqlite3.connect('fantasy\_cricket.db')

team=self.cb1.currentText()

self.list01.clear()

try:

#Adding players to list1 according to team name.

sql1="SELECT players,tot\_value from teams where name='"+team+"';"

cur=conn.execute(sql1)

row=cur.fetchone()

list\_add=row[0].split(',')

self.list01.addItems(list\_add)

except:

conn.rollback()

team\_total=0

self.list02.clear()

match=self.cb2.currentText()

try:

#For every player in list1:

for i in range(self.list01.count()):

total=0

batscore=0

bowlscore=0

fieldscore=0

name=self.list01.item(i).text()

cursor1=cur.execute("SELECT \* from "+match+" WHERE player='"+name+"';")

row=cursor1.fetchone()

batscore=int(row[1]/2)

if batscore>=50:

batscore+=5

if batscore>=100:

batscore+=10

#Calculating strike rate.

if row[2]>0:

strike\_rate=row[1]/row[2]

if strike\_rate>=80 and strike\_rate<=100:

batscore+=2

if strike\_rate>100:

batscore+=4

#Calculating score for 4s and 6s.

batscore+=row[3]

batscore+=(2\*row[4])

bowlscore=10\*row[8]

if row[8]>=3:

bowlscore+=5

if row[8]>=5:

bowlscore+=10

#Calculating economy rate.

if row[5]>0:

eco\_rate=6\*row[7]/row[5]

if eco\_rate>=3.5 and eco\_rate<=4.5:

bowlscore+=4

if eco\_rate>=2 and eco\_rate<3.5:

bowlscore+=7

if eco\_rate<2:

bowlscore+=10

#Calculating fieldscore.

fieldscore=10\*(row[9]+row[10]+row[11])

#Calculating score of individual player,setting value in list2

#Adding to team total.

total=batscore+bowlscore+fieldscore

self.list02.addItem(str(total))

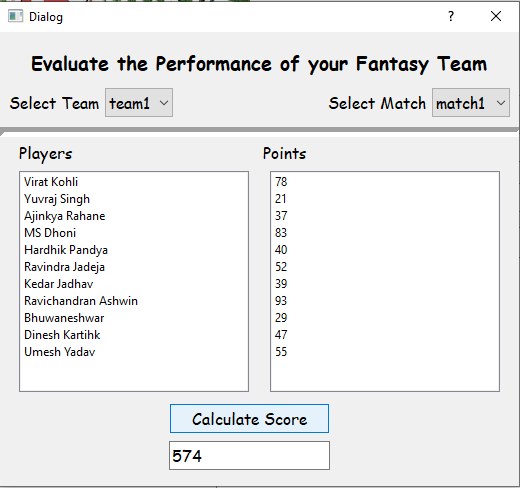
team\_total+=total

except:

conn.rollback()

#Setting total team score in textbox.

self.lineEdit.setText(str(team\_total))



**Conclusion**

This project is a desktop application for Fantasy Cricket Game build in python3 using PyQt5 libraries for GUI and sqlite studio for backend.

Select your dream team, analyze scores and have fun with this amazing application!