*Synopsis of*

# BUILDING A DESKTOP APPLICATION FOR FANTASY CRICKET GAME

**Submitted as a part of INTERNSHALA's Online Training**

**on**

**Programming with Python**

(Jun 2021-Aug 2021)

***submitted by***

**Charulatha K**

# BUILDING A DESKTOP APPLICATION FOR FANTASY CRICKET GAME

## 1.INTRODUCTION

Python is **an interpreted high-level programming language**. It supports multiple programming paradigms. It is a dynamically typed and extensible language.

As part of Internshala's online training on programming with python, a desktop GUI application which is a fantasy cricket game was developed by creating a virtual team of cricket players and scored points could be calculated based on how the players perform in different matches. Fantasy cricket game is an online game in which virtual teams of real cricket players are created. Based on the performance of the selected players, points are scored.

## 2.METHODOLOGY

The database of players was created using a SQLiteStudio ,a desktop application for performing CRUD (create, retrieve, update, delete)operations on database files. SQLite is a relational database management system contained in a C library. The Python Standard Library includes a module called "sqlite3" intended for working with this database. This module is a SQL interface compliant with the DB-API 2.0 specification.

For the database, three tables (entities)– match, stats and teams was created with its required attributes and record. The code to create database is shown in Fig.1.

### Common SQL commands used:

- CREATE TABLE - creates a new table
- INSERT INTO - inserts new data into a database
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database.

*myteam.py - C:\Python36\myteam.py (3.6.2)*

File Edit Format Run Options Window Help

```
import sqlite3
MyTeam=sqlite3.connect('cricket.db')
curteam=MyTeam.cursor()
curteam.execute('''CREATE TABLE match(Player,Scored,Faced,Fours,Sixes,Bowled,
                # Maiden,Given,Wkts,Catches,Stumping,RunOut);''')
MyTeam.commit()
curteam.execute('''CREATE TABLE teams(Name,Players,Value);''')
curteam.execute('''CREATE TABLE stats (Player,Matches,Runs,"100s","50s",Value,ctg);''')
MyTeam.commit()
player=input("Enter player name:")
scored=int(input("Enter score:"))
faced=int(input("Enter faced value:"))
fours=int(input("No. of 4s:"))
sixes=int(input("No.of 6s:"))
bowled=int(input("No.of balls bowled:"))
maiden=int(input("Enter Maiden:"))
given=int(input("Given:"))
wkts=int(input("Enter no. of Wickets:"))
catches=int(input("Enter no. of catches:"))
stmp=int(input("Enter no.of stumping:"))
ro=int(input("Run Out:"))
value=int(input("Enter value:"))
match=int(input("Matches:"))
runs=int(input("Runs:"))
century=int(input("100s:"))
half_century=int(input("50s:"))
ctg=input("Role:")
curteam.execute('''INSERT INTO match(Player,Scored,Faced,Fours,Sixes,Bowled,Maiden,Given,Wkts,Catches,Stumping,RunOut) VALUES(?,?,?,?,?,?,?,?,?,?,?,?);''',(player,scored,faced,fours,sixes,
curteam.execute('''INSERT INTO stats(Player,Matches,Runs,"100s","50s",Value,ctg) VALUES(?,?,?,?,?,?,?);''',(player,match,runs,century,half_century,value,ctg))
MyTeam.commit()
```

Fig.1 Database Code

After creating database, GUI application was created using QtDesigner. It is a Qt tool for designing and building graphical user interfaces (GUIs) with Qt Widgets.GUI libraries like Qt can be ported to python.

**Common PyQt widgets used are** QLabel, QLineEdit, QPushButton, QRadioButton, QComboBox, QlistWidget, Horizontal Layout

## 2.1 Designing Main Window

1. After opening QtDesigner, a Main Window was created ” Manage Teams” was typed in menu bar and different menu were created.

2. Display widgets were placed in a horizontal layout for displaying the category of players and to display the number of each category players and also to display the points available and used while selecting the players.

3. Image is placed into Qlabel object by using pixmap property.

4. Four Radiobuttons were used to select any one category of players.

5. Two ListWidgets were used in a horizontal layout.List1 gets filled with corresponding players when radio buttons were pressed whereas list2 is empty first. It is used to select an item in one list, remove the selected item from it and add inother list. For this purpose we use itemDoubleClicked signal emitted by QListWidget widget and connect it to event handler methods.
   The opening of application window looks like as shown in Fig.2.

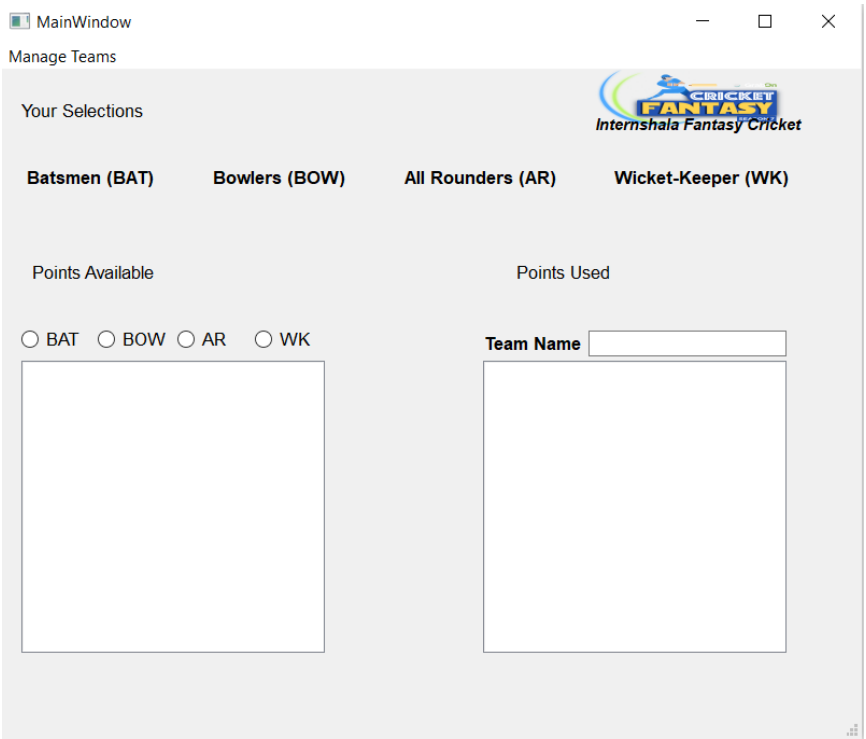


Fig.2 Opening application

The players of each category can be seen by selecting the category. The selection is disabled until a new team is created from the Manage Teams menu. When NEW Team is clicked a pop up asking the team name appears as shown in Fig.3.

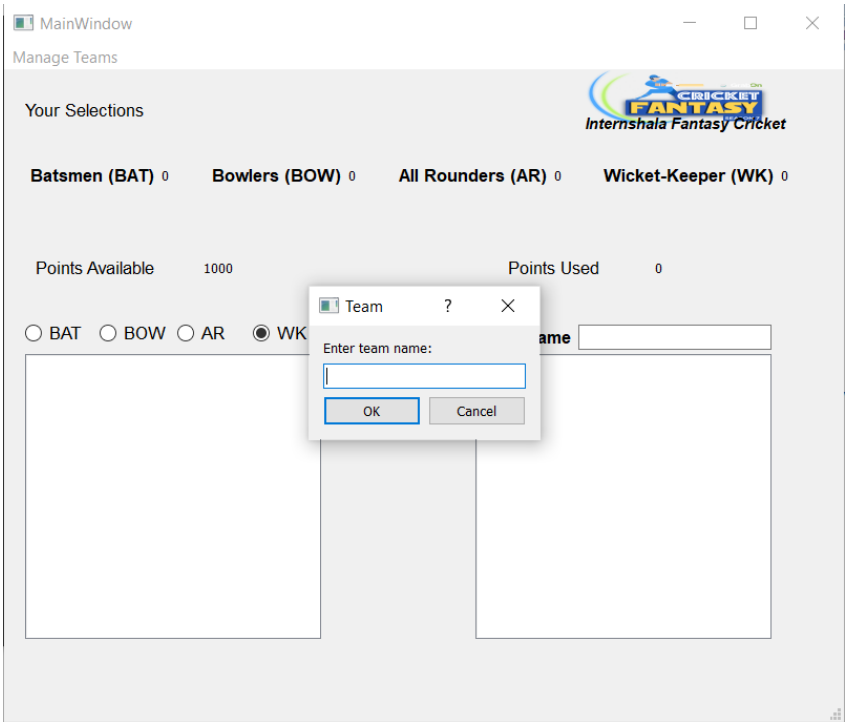


Fig.3 NEW Team Window

After entering a team name, the players can be selected from the corresponding category. On double-clicking each player name, the right box gets populated. Points available and used are displayed accordingly.

While selecting, certain set of rules of cricket are followed, if we try to input more than one wicket-keeper an error pops up. Message is displayed if logic of the game is not followed. Fig.4(a),(b) shows the input window for creating a new team.

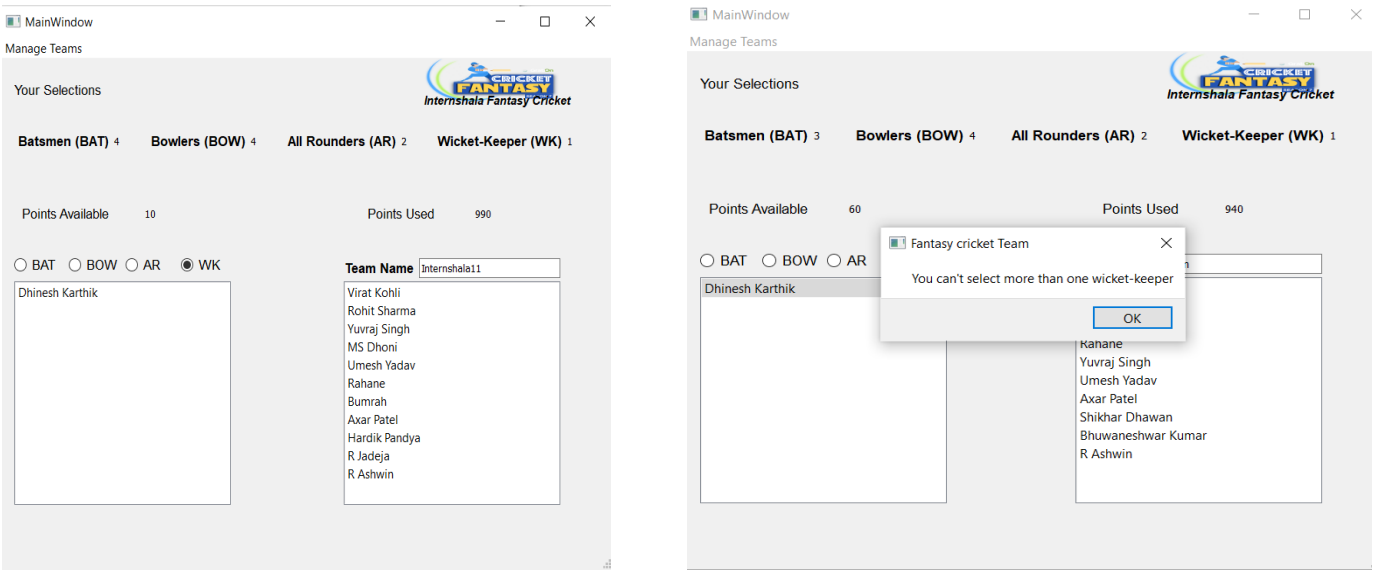


Fig.4(a),(b) Creating a new team

After creating a new team it can be saved using SAVE Team option from Manage Teams Menu. Upon successful saving, a pop up appears saying ,“Team Saved Successfully”.

Saved Teams can be accessed from OPEN Team option. When it is clicked a pop up appears which allows to choose a team from saved teams and opens it as shown in Fig.5.

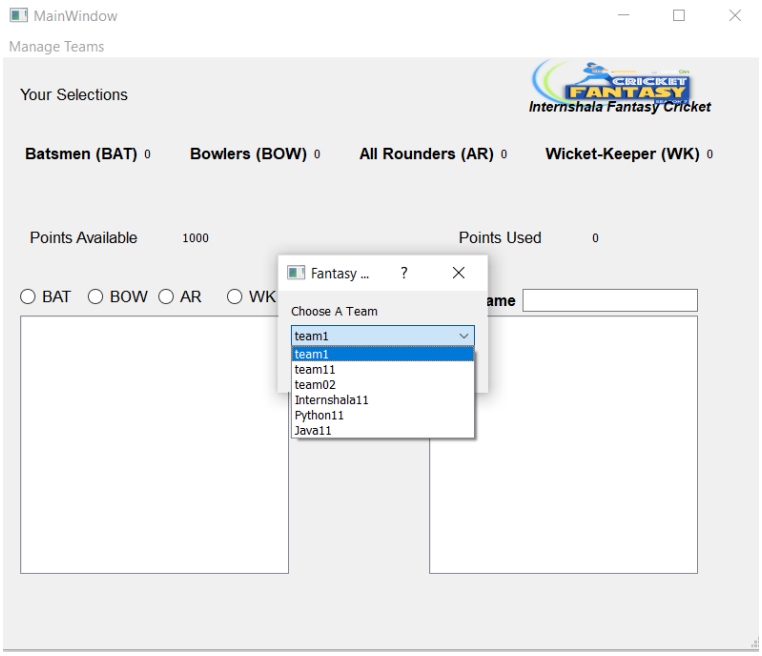


Fig.5 Opening a team

Pop up appears on clicking EVALUATE Team. It is designed in QtDesigner by dragging and dropping two combo box for selecting team and match, two listWidget in a horizontal layout.List1 displays the name of players and list2 displays their respective score.

You can select your team from select team combo box and the match from select match combo box. Clicking on the Calculate score pushbutton evaluates the performance of every player and the total team score i.e, the sum of points scored by each player is displayed in display widget. While calculating the points of each player certain set of batting, bowling and fielding rules are followed to calculate points. The code for calculating the score is shown in Fig.6 and the application window for evaluating the performance of a team is shown below in Fig.7.

scorecal.py - C:\Python36\Scripts\Charulatha_Python Final Project\scorecal.py (3.6.2)

File Edit Format Run Options Window Help

```
    def calculate_score(self):
        import sqlite3
        MyTeam= sqlite3.connect('cricket.db')
        curteam=MyTeam.cursor()
        team=self.comboBox1.currentText()
        self.listWidget1.clear()
        sql="select Players, Value from teams where Name='"+team+"'"
        cur=curteam.execute(sql)
        row=cur.fetchone()
        selected=row[0].split(',')
        self.listWidget1.addItems(selected)
        team_total=0
        self.listWidget2.clear()
        match=self.comboBox2.currentText()
        for i in range(self.listWidget1.count()-1):
            total=0
            batscore=0
            bowlscore=0
            fieldscore=0
            nm=self.listWidget1.item(i).text()
            cur=curteam.execute("select * from "+match+" where Player='"+nm+"'")
            row=cur.fetchone()
            batscore=row[1]//2
            if batscore>=50: batscore+=5
            if batscore>=100: batscore+=10
            if row[1]>0:
                strike_rate=row[1]/row[2]
                if strike_rate>=80 and strike_rate<100: batscore+=2
                if strike_rate>=100:batscore+=4
            batscore=batscore+row[3]
            batscore=batscore+2*row[4]
            bowlscore=row[8]*10
            if row[8]>=3: bowlscore=bowlscore+5
            if row[8]>=5: bowlscore=bowlscore=bowlscore+10
            if row[7]>0:
                economy_rate=6*row[7]/row[5]
                if economy_rate<=2: bowlscore=bowlscore+10
                if economy_rate>2 and economy_rate<=3.5: bowlscore=bowlscore+7
                if economy_rate>3.5 and economy_rate<=4.5: bowlscore=bowlscore+4
            fieldscore=(row[9]+row[10]+row[11])*10
            total=batscore+bowlscore+fieldscore
            self.listWidget2.addItem(str(total))
            team_total=team_total+total


        self.ptlabel.setText(str(team_total))
```

Fig.6 Score calculation code

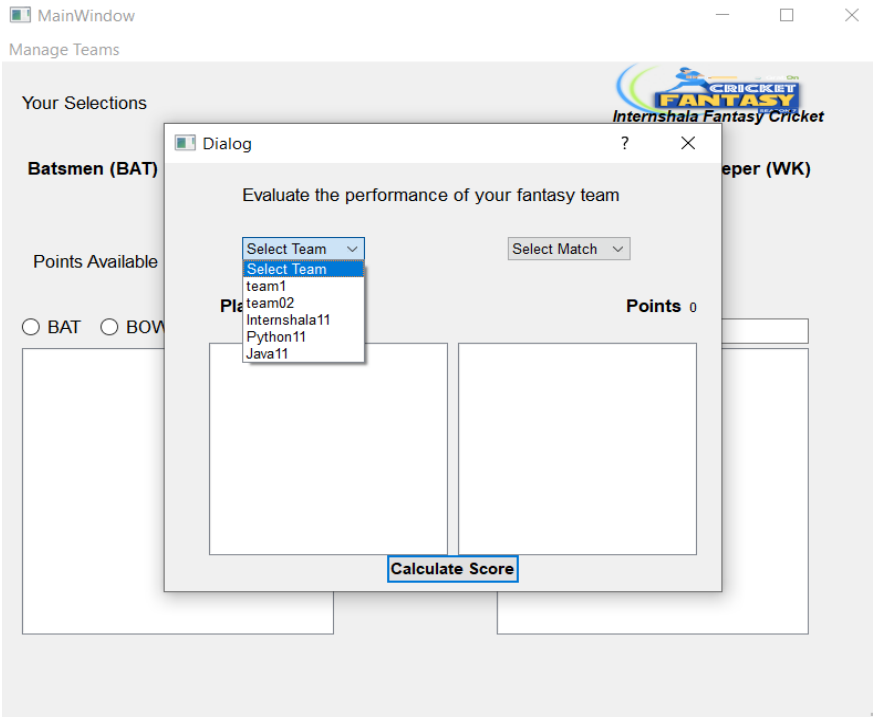


Fig.7 Team Evaluation

After clicking on calculate score pushbutton , the points are displayed as shown in Fig.8.

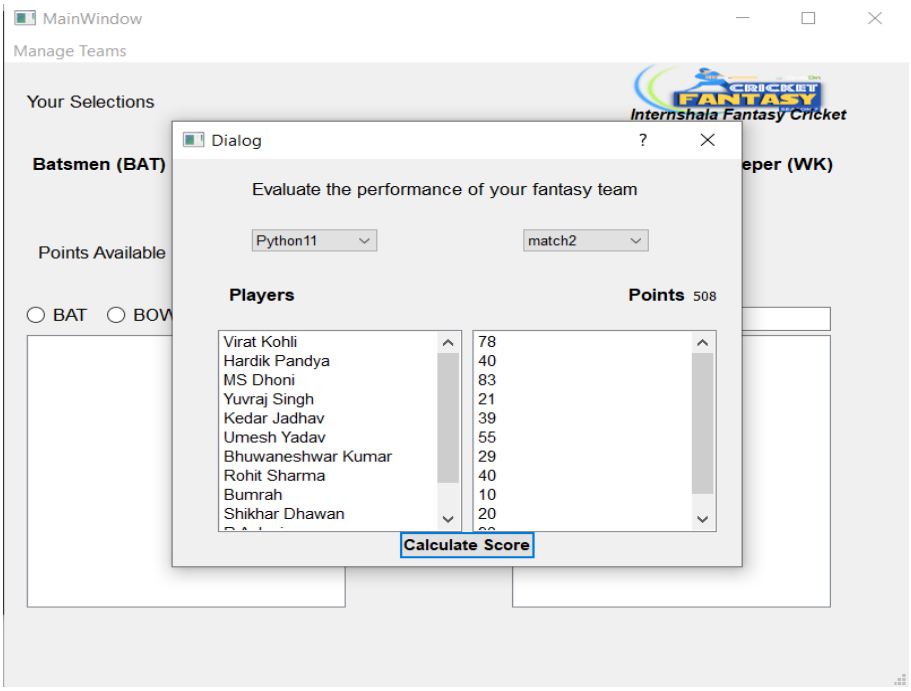


Fig.8 Displaying Score

pyuic5 –x fantasyCricket.ui –o FantasyCricketProject(Main Script).py command is used to convert ui file into py file.

## 3. CONCLUSION/LEARNING OUTCOME

Overall this project gives an insight into the various disciplines learned in Internshala's six week online training program on programming with python that comprises of various modues such as Introduction to Python, Using Variables inPython, Basics of Programming in Python, Principles of Object-oriented Programming (OOP), Connecting to SQLite Database, Developing a GUI withPyQT .