SIX WEEKS SUMMER TRAINING REPORT

On

"LPU - Object Oriented Programming using Python – Internship"

Submitted by:

Satya Diresh

Registeration No: 11901637

Programme Name : BTech(CSE).

Under the Guidance of E-box

School of Computer Science & Engineering
Lovely Professional University , Phagwara

(May –Aug 2021)

DECLARATION

I here by declare that I have completed my six weeks summer training at E- box from 01 May,2021 to 30 Aug,2021. I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes full fill the requirements of training for the award of degree of B.Tech(CSE) Lovely Professional University, Phagwara.

`Signature of Student

Name: Satya Diresh

Reg.No: 11901637

Start Date : 01 May 2021

End Date : 30 Aug 2021

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my mentor whose contribution in stimulating suggestions and encouragement helped me to coordinate my project especially in writing this report. I express my thanks to my institution Lovely Professional University for giving me an opportunity to learn this interesting topic. I also convey my regards to my faculty assistance all through this training named “Object Oriented Programming”. Once again I would like to thank all my supporters from the core of my heart.

Training certificate from organization

INTRODUCTION TO PYTHON

Python Language Introduction

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It

uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

PYTHON FEATURES

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below − □ It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

TRAINING CONTENTS

1. **Introduction to Python**

Learn how to install Python, distinguish between important data types and use basic features of the Python interpreter, IDLE.

2. **Using Variables in Python**

   Learn about numeric, string, sequence and dictionary data types and relevant operations while practicing Python syntax.

3. **Basics of Programming in Python**

   Learn how to write programs using conditionals, loops, iterators and generators, functions and modules and packages.

4. **Principles of Object-oriented Programming (OOP)**

   Learn about the important features of Object-oriented Programming while using Classes and Objects, two main aspects of the OOP paradigm.

5. **Connecting to SQLite Database**

   Learn about relational databases while learning how to store and retrieve data from an SQLite database through Python.

6. **Developing a GUI with PyQT**

   Learn how to install PyQt5 toolkit, Qt Designer and create a graphical user interface using common widgets and menu systems.

7. **Application of Python in Various Disciplines**

   Learn about various resources to extend your learning for the Python programming language.

PROFILE OF THE PROBLEM

Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, we can use rules similar to the sample rules displayed below.

Sample of Rules

Batting

- 1 point for 2 runs scored
- Additional 5 points for half century
- Additional 10 points for century
- 2 points for strike rate (runs/balls faced) of 80-100
- Additional 4 points for strike rate>100
- 1 point for hitting a boundary (four) and 2 points for over boundary (six) **Bowling**
- 10 points for each wicket
- Additional 5 points for three wickets per innings
- Additional 10 points for 5 wickets or more in innings
- 4 points for economy rate (runs given per over) between 3.5 and 4.5
- 7 points for economy rate between 2 and 3.5
- 10 points for economy rate less than 2 **Fielding**
- 10 points each for catch/stumping/run out

DATABASE DESIGN

For the database, we are required to use three tables – match, stats and teams.

Match1

Player	Scored	Faced	Fours	Sixes	Bowled	Maiden	Given	Wkts	Catches	Stumping	RO*

***Run Out**

Team

name	players

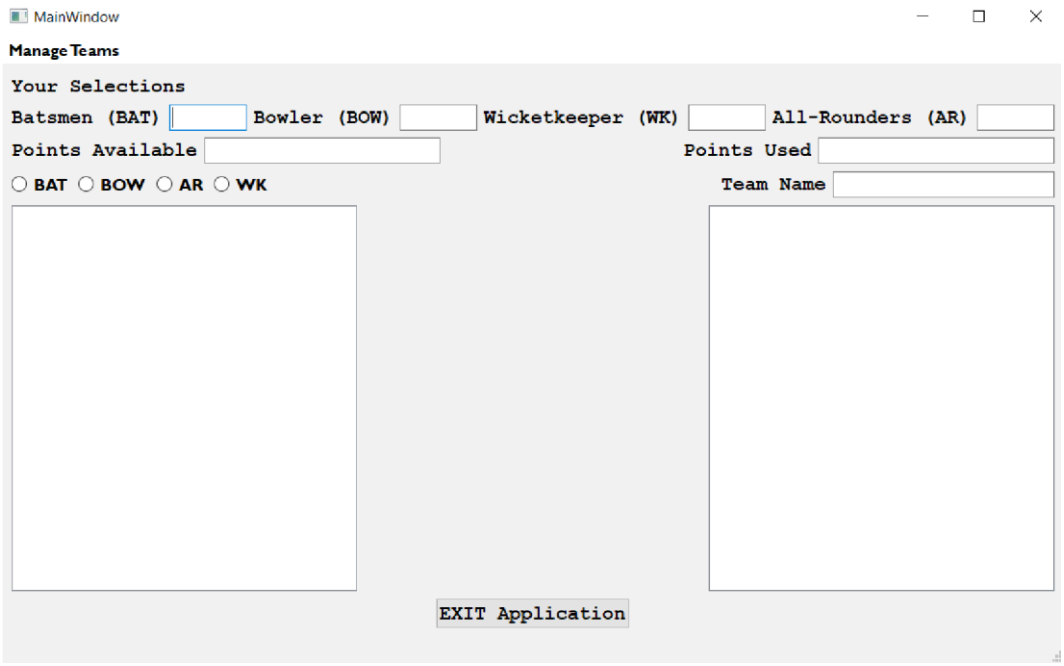
Stats

player	matches	runs	100s	50s	value	ctg

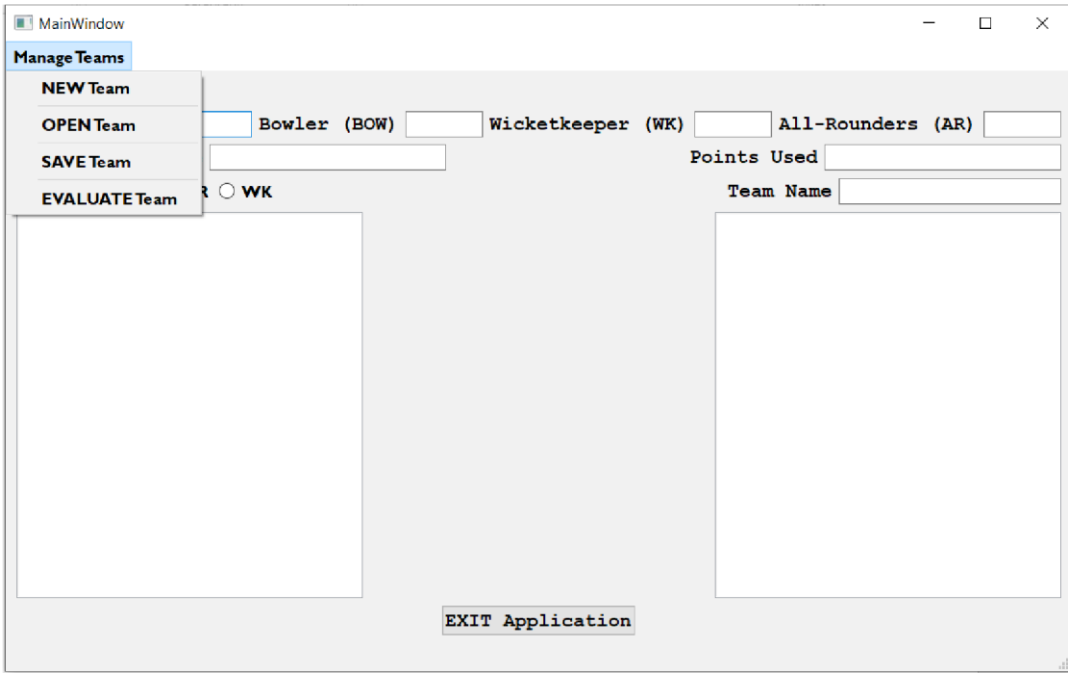
The data to enter in the remaining two tables is given below:

player	scored	faced	fours	sixes	bowled	maiden	given	wkts	catches	stumping	ro	value	matches	runs	100s	50s	ctg
Kohli	102	98	8	2	0	0	0	0	0	0	1	120	189	8257	28	43	BAT
Yuvraj	12	20	1	0	48	0	36	1	0	0	0	100	86	3589	10	21	BAT
Rahane	49	75	3	0	0	0	0	0	1	0	0	100	158	5435	11	31	BAT
Dhawan	32	35	4	0	0	0	0	0	0	0	0	85	25	565	2	1	AR
Dhoni	56	45	3	1	0	0	0	0	3	2	0	75	78	2573	3	19	BAT
Axar	8	4	2	0	48	2	35	1	0	0	0	100	67	208	0	0	BWL
Pandya	42	36	3	3	30	0	25	0	1	0	0	75	70	77	0	0	BWL
Jadeja	18	10	1	1	60	3	50	2	1	0	1	85	16	1	0	0	BWL
Kedar	65	60	7	0	24	0	24	0	0	0	0	90	111	675	0	1	BWL
Ashwin	23	42	3	0	60	2	45	6	0	0	0	100	136	1914	0	10	AR
Umesh	0	0	0	0	54	0	50	4	1	0	0	110	296	9496	10	64	WK
Bumrah	0	0	0	0	60	2	49	1	0	0	0	60	73	1365	0	8	WK
Bhuwaneshwar	15	12	2	0	60	1	46	2	0	0	0	75	17	289	0	2	AR
Rohit	46	65	5	1	0	0	0	0	1	0	0	85	304	8701	14	52	BAT
Kartick	29	42	3	0	0	0	0	0	2	0	1	75	11	111	0	0	AR

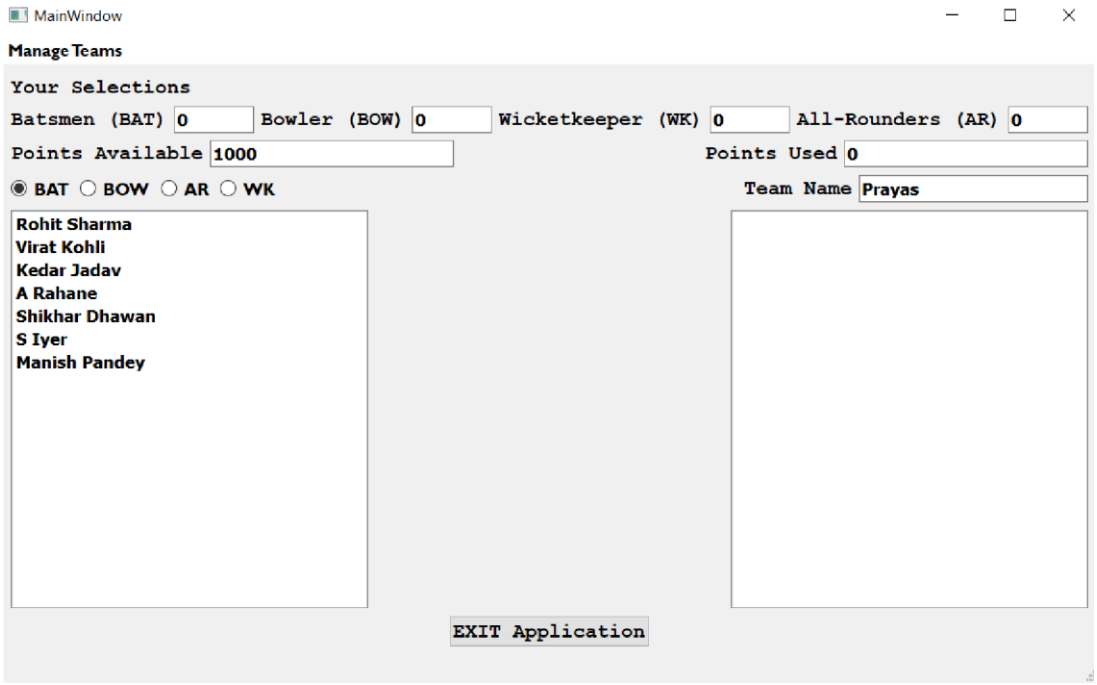
Testing /Deployment



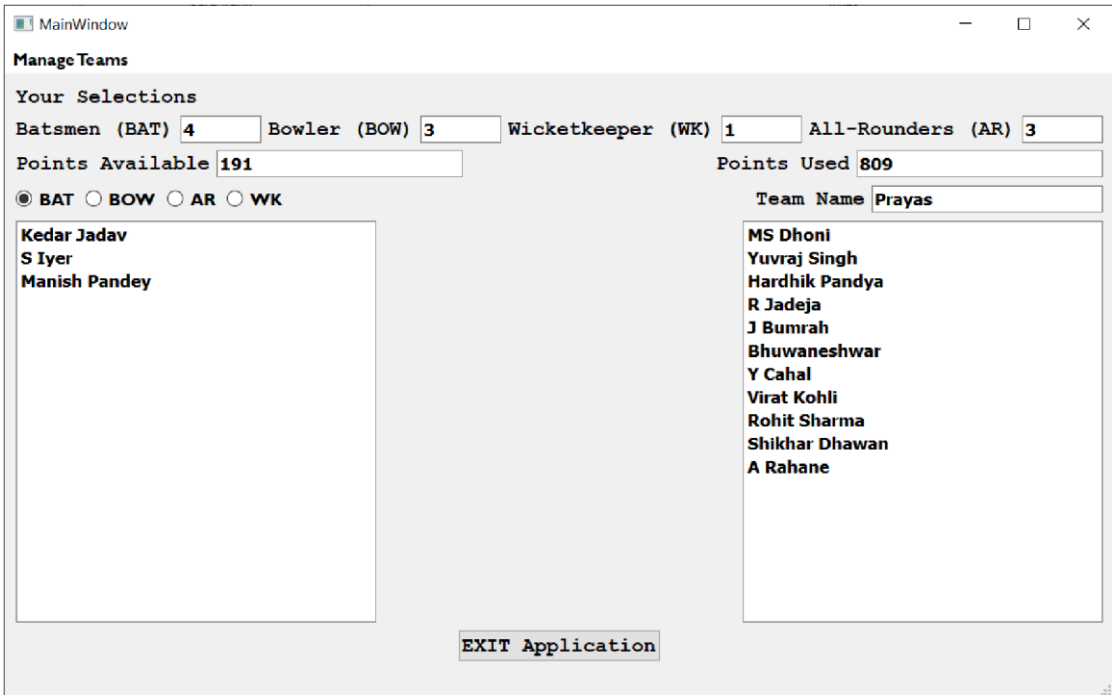
- Opening screen of the application. You can see the players of each category by selecting the category. To begin with, the selection is disabled until a new team is created from the Manage Teams menu. A pop up asking the name of the team appears.



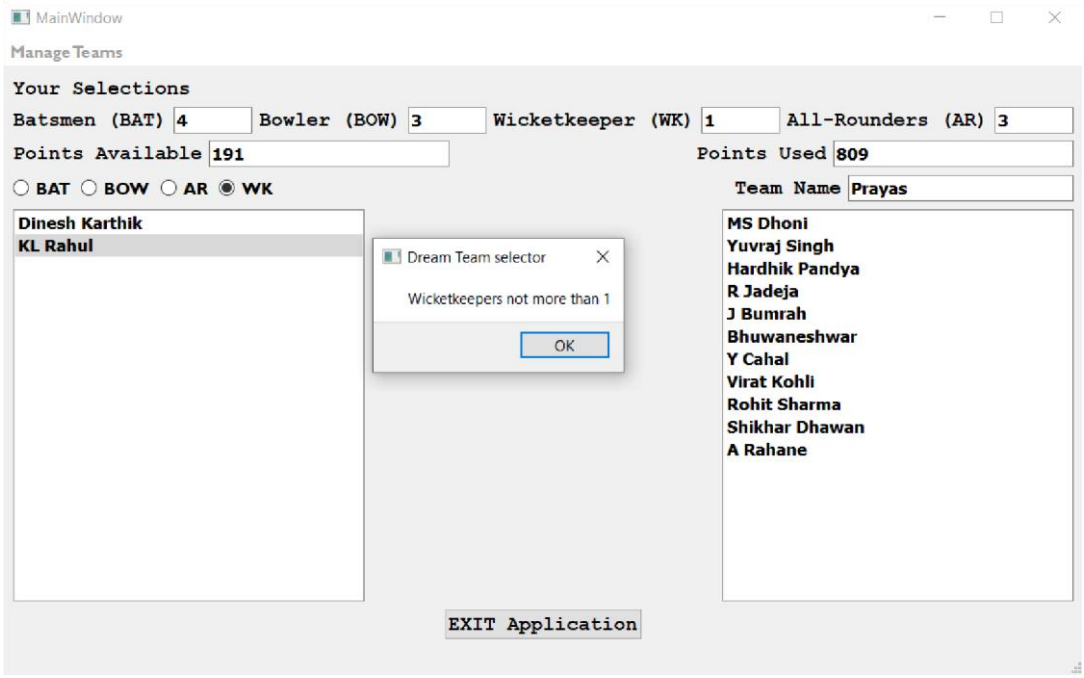
- The toolbar menu options which allow you to create a new team, open an existing team, save your team and finally evaluate the score of a saved team.



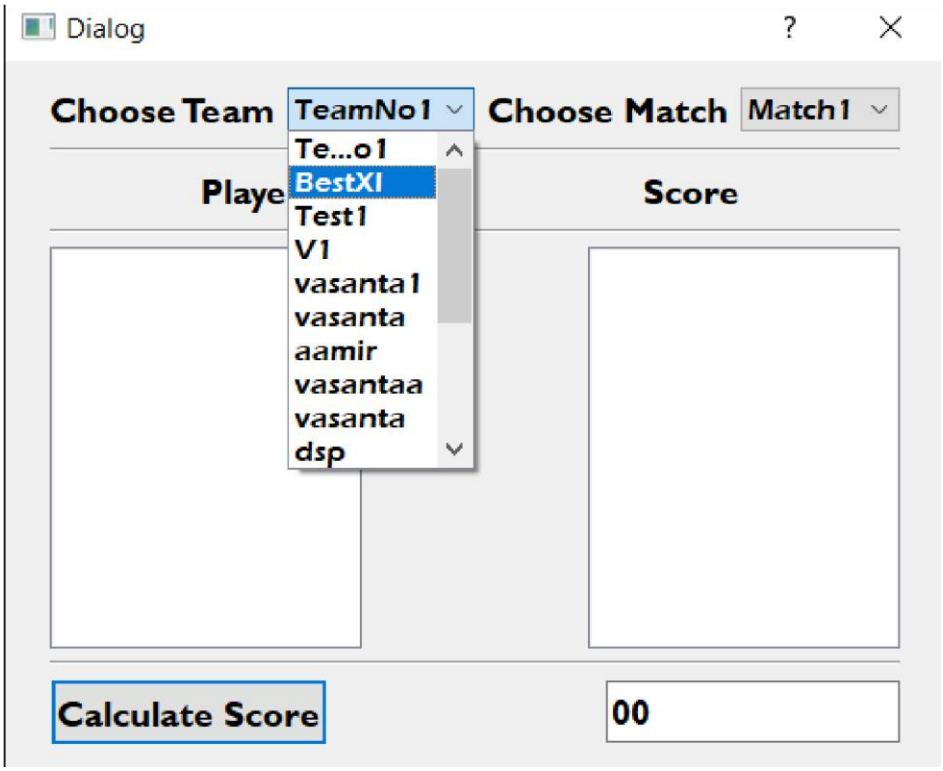
- After clicking New Team, the left box is populated with player names. As you select a different category, the corresponding list of players is displayed.



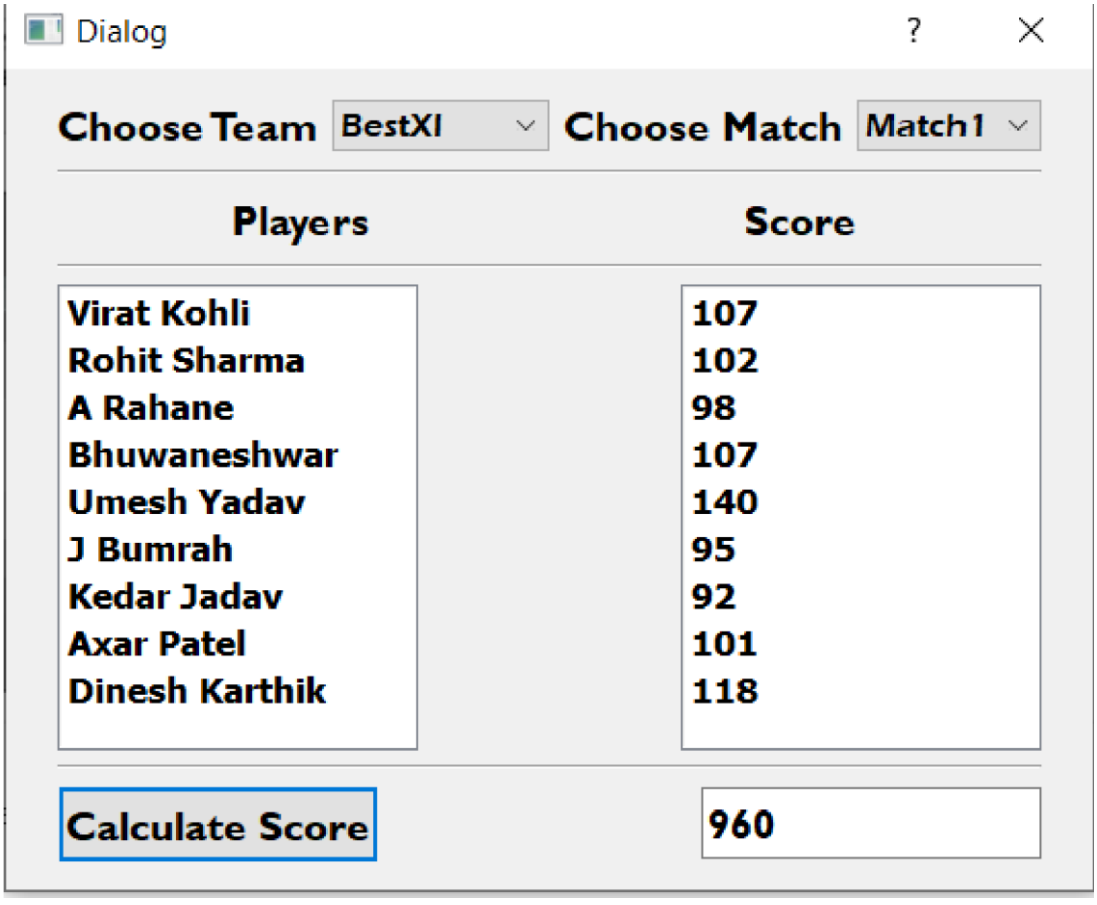
- On double-clicking each player name, the right box gets populated. Points available and used are displayed accordingly.



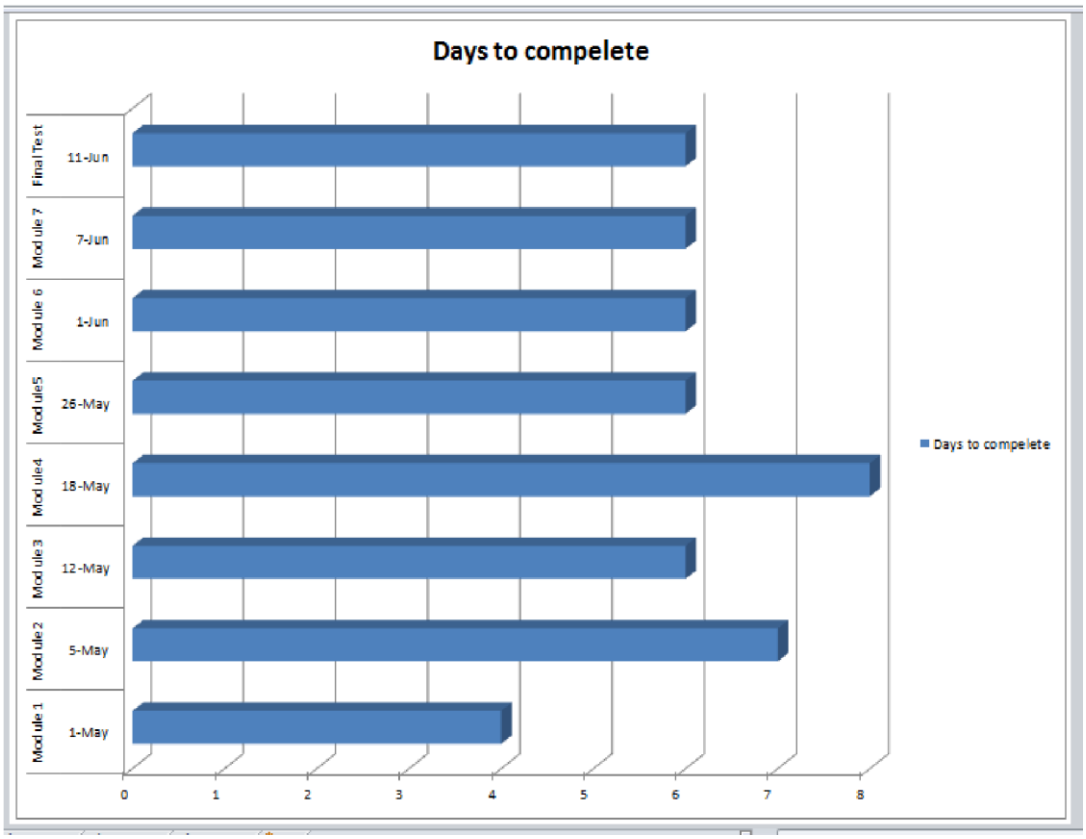
- Message if the game logic is not followed



- Upon opening the second file to evaluate the scores. You can select your team here and the match for which the players' performance is compared.



- The final score for your fantasy team based on the match selected.



Problem Analysis

PRODUCT DEFINATION:-It is an game where you create a team of real cricket players and score points depending on how your chosen players perform in real life matches. To win a tournament, you must try and get the maximum points and the No. 1 rank amongst other participants.

FEASIBILITY ANALYSIS:- I am building an software for gaming purposes using an specific technology named python. It is a game software where you can create virtual team according to your choice and score points to win an tournament.

This software is created for motivating street cricket and adding more fun and entertainment to cricket. The components that are used in this demo can be integrated to a high extent to provide statics to different components of cricket. This project helps in providing real time on field actions there by helping its user of the current actions happening on field.

Coding

fan1.py - C:\Users\Prayas\Downloads\Compressed\Summer-Training-Python-development-master\Summer-T...

File Edit Format Run Options Window Help

```
    def menu(self,action):
        txt=(action.text())

        if txt=='NEW Team':
            self.bat=0
            self.bwl=0
            self.ar=0
            self.wk=0
            self.avl=1000
            self.used=0
            self.list1.clear()
            self.list2.clear()
            #self.t7.setText("???")

            #self.dial()
            text, ok=QtWidgets.QInputDialog.getText(MainWindow, "Team", "Enter name of team
            if ok:
                self.t7.setText(str(text))

            self.show()


        if txt=='SAVE Team':
            count=self.list2.count()
            selected=""
            for i in range(count):
                selected+=self.list2.item(i).text()
                if i<count:
                    selected+=","
            self.saveteam(self.t7.text(),selected,self.used)

        if txt=='OPEN Team':
            self.bat=0
            self.bwl=0
            self.ar=0
            self.wk=0
            self.avl=1000
            self.used=0
            self.list1.clear()
            self.list2.clear()
            self.show()
            #print("rgr")
            self.openteam()


        if txt=='EVALUATE Team':
            from dlgscore import Ui_Dialog
```

Ln: 334 Col: 27

*fan1.py - C:\Users\Prayas\Downloads\Compressed\Summer-Training-Python-development-master\Summer-...

File Edit Format Run Options Window Help

```
    def saveteam(self,nm,ply,val):
        #print("rvrv")

        if self.bat+self.bwl+self.ar+self.wk!=11:
            self.showdlg("Insufficient players")
            return

        #print("frbfj")
        sql="INSERT INTO teams (name,players,value) VALUES ('"+nm+"','"+ply+"','"+str(val)+'
        #print("f3f4")
        try:
            #print("bjr")
            cur=conn.execute(sql)
            #print("dehe")
            self.showdlg("Team Saved Succesfully")
            conn.commit()
        except:
            self.showdlg("Error in Operation")
            conn.rollback()


    def showdlg(self,msg):
        #print("ecb")
        Dialog=QtWidgets.QMessageBox()
        Dialog.setText(msg)
        Dialog.setWindowTitle("Dream Team selector")
        ret=Dialog.exec()


if __name__ == "__main__":
    import sqlite3
    conn = sqlite3.connect('fantasy.db')
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

Ln: 546 Col: 4

*fan1.py - C:\Users\Prayas\Downloads\Compressed\Summer-Training-Python-development-master\Summer-...

File Edit Format Run Options Window Help

```
    def show(self):
        #print("vvrv")
        self.t1.setText(str(self.bat))
        self.t2.setText(str(self.bwl))
        self.t3.setText(str(self.wk))
        self.t4.setText(str(self.ar))
        #print("rrrr")
        self.t5.setText(str(self.avl))
        self.t6.setText(str(self.used))
        #print("efef")



    def criteria(self,ctgr,item):
        msg=''
        if ctgr=="BAT" and self.bat>=5:msg="Batsmen not more than 5"
        if ctgr=="BWL" and self.bwl>=5:msg="bowlers not more than 5"
        if ctgr=="AR" and self.ar>=3:msg="Allrounders not more than 3"
        if ctgr=="WK" and self.wk>=1:msg="Wicketkeepers not more than 1"
        if msg!='':
            #msg="You Have Exhausted your Points"
            self.showdlg(msg)
            return False

        if self.avl<=0:
            msg="You Have Exhausted your Points"
            self.showdlg(msg)
            return False

        if ctgr=="BAT":self.bat+=1
        if ctgr=="BWL":self.bwl+=1
        if ctgr=="AR":self.ar+=1
        if ctgr=="WK":self.wk+=1


        sql="SELECT value from stats where player='"+item.text()+"'"
        cur=conn.execute(sql)
        row=cur.fetchone()
        self.avl-=int(row[0])
        self.used+=int(row[0])
        return True
```

Ln: 546 Col: 4

BIOGRAPHY

- https://www.w3schools.com/python/
- https://wiki.python.org/moin/PyQt/Tutorials
- https://www.tutorialspoint.com/pyqt/
- https://www.tutorialspoint.com/sqlite/sqlite_quick_guide.htm
- https://e-box.co.in