

FANTASY CRICKET GAME

Project Report

INDUSTRIAL TRAINING (ECS599)

B.TECH CSE (AI+ML+DL)

PROJECT GUIDE:

Mr. VIVEK KUMAR (Internal)

Mr. SARVESH AGRAWAL (External)

SUBMITTED BY:

RISHANT RAJPOOT

TCA1959009

DECEMBER/2021



**COLLEGE OF COMPUTING SCIENCES AND IT
TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance. First and foremost, I wish to record our sincere gratitude to Internshala Coordinators for their constant support and encouragement in preparation of this report and for making available videos and interface facilities needed to prepare this report. The training on “Python” was very helpful for me in giving the necessary background information and inspiration in choosing this topic for the training. Their contributions and technical support in preparing this report are greatly acknowledged. Last but not the least, I wish to thank my parents for financing my studies in this college as well as for constantly encouraging me to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

RISHANT RAJPOOT

Place: TMU(MORADABAD)

Date: 20/12/2021

DECLARATION

I hereby declare that this Project Report titled “FANTASY CRICKET GAME” submitted by me and approved by my project guide, the College of Computing Sciences and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by me.

Student Name: RISHANT RAJPOOT

Project Guide: Mr. SARVESH AGARWAL
(External)

Project Guide: Mr. VIVEK KUMAR
(Internal)

CERTIFICATION



Brief About the Company

Internshala is an online trainings provider company, located at Gurugram (GURGAON).

Internshala provides a platform from where anyone can access the available material for the desired need as per individual.

Internshala provides the certification for the trainings enrolled in.

The certificates provided by the company are valid in the fields for which the certification approved.

Company have some prerequisite and some qualifying criteria to get certified.

Internshala not just provide online training certification, it also provides the platform where we can access to the internships, for internships Internshala does not ply any charge to get hired.

Along with this all, Internshala is a dot com business with the heart of dot org. We are a technology company on a mission to equip students with relevant skills & practical exposure to help them get the best possible start to their careers. Imagine a world full of freedom and possibilities. A world where you can discover your passion and turn it into your career. A world where you graduate fully assured, confident, and prepared to stake a claim on your place in the world.

Table of Contents

1	Project Title.....	7
2	Domain.....	7
3	Problem Statement.....	7
4	Problem Description.....	7
4.1	Scope of the Work.....	10
4.2	Project Modules.....	10
5	Implementation Language.....	11
6	Description about Language.....	11
7	Implementation Methodology.....	14
8	Technologies to be used.....	14
8.1	Software Platform.....	14
8.2	Hardware Platform.....	14
9	Description of the database.....	15
10	Description of the GUI library.....	17
11	Advantages of this Project.....	20
12	Future Scope and further enhancement of the Project.....	20
13	GUI of the application.....	21
14	Database GUI.....	24
15	Conclusion.....	26
16	References.....	26h

Appendix

A:Data Flow Diagram (DFD)

B: Coding Screenshots

1 Project Title

Fantasy Cricket Game

What is Fantasy Cricket Game?

Amidst the wonderous sports that are popular globally, fantasy cricket apps have taken the digital sporting world by storm. In India, especially online fantasy cricket is a craze due to the unending love that Indian fans have for the sport. But the question is what exactly is the concept of fantasy cricket?

In layman's terms, fantasy cricket is an online cricket gaming platform that allows participants to create virtual cricket teams consisting on 11 players based on the real players playing in the live matches. Thus, this is an extremely enticing platform for all die-hard cricketing fans who will never get the opportunity of experiencing the real-life on-field thrill

For playing fantasy cricket games in India, the participants are required to have at least the basic knowledge of the game and the players. Unlike other fantasy games, here luck or probability doesn't play any role. Rather based on the past performances of the players, physical fitness, run rates, wickets and playing style you need to create your own team. Further based on the performance of the real-life cricketers on the field, your fantasy cricket team gets the points. The more points your players secure, the more real time cash you can earn.

2 Domain

Computer Application

3 Problem Statement

Create Fantasy Cricket Game in python. The game should have all the features as described in project descriptions section. Which will work to add players, make teams, save the teams, retrieve the teams from database and evaluate the teams as per the points available for the teams and per players. All the evaluated will get saved into the database.

Everything can be retrieved from the database in this case "SQLite3 studio application". Application must have some constraints which must be satisfied to make the teams.

All further details will be described in the project description section.

4 Project Description

Manage Teams

Your Selections

Batsmen (BAT) ### Bowlers (BOW) ### Allrounders (AR) ### Wicket-keeper (WK) ###

Points Available ##### Points Used #####

BAT BOW AR WK >

Team Name Displayed Here



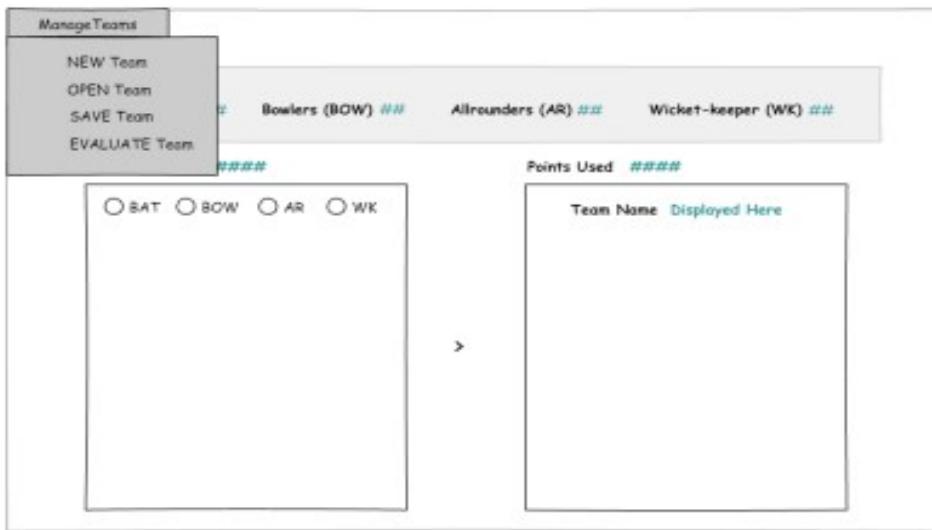
Manage Teams

NEW Team OPEN Team SAVE Team EVALUATE Team ##### Points Used #####

Bowlers (BOW) ### Allrounders (AR) ### Wicket-keeper (WK) ###

BAT BOW AR WK >

Team Name Displayed Here



Manage Teams

Your Selections

Batsmen (BAT) 0 Bowlers (BOW) 0 Allrounders (AR) 0 Wicket-keeper (WK) 0

Points Available 1000 Points Used 0

BAT BOW AR WK

Virat Kohli
Shikhar Dhawan
Rohit Sharma
Ajinkya Rahane
Yuvraj Singh

Team Name InternshalaII

Manage Teams

Your Selections

Batsmen (BAT) 4 Bowlers (BOW) 3 Allrounders (AR) 3 Wicket-keeper (WK) 1

Points Available 50 Points Used 950

BAT BOW AR WK

Rohit Sharma

Team Name InternshalaII

Virat Kohli
Shikhar Dhawan
Ajinkya Rahane
Yuvraj Singh
Bhuvneshwar Kumar
Umesh Yadav
Kedar Jadhav
Hardik Pandya
M.S.Dhoni
Ravindra Jadeja
Jasprit Bumrah

Manage Teams

Your Selections

Batsmen (BAT) 4 Bowlers (BOW) 3 Allrounders (AR) 3 Wicket-keeper (WK) 1

Points Available 50 Points Used 950

BAT BOW AR WK

Dinesh Karthik

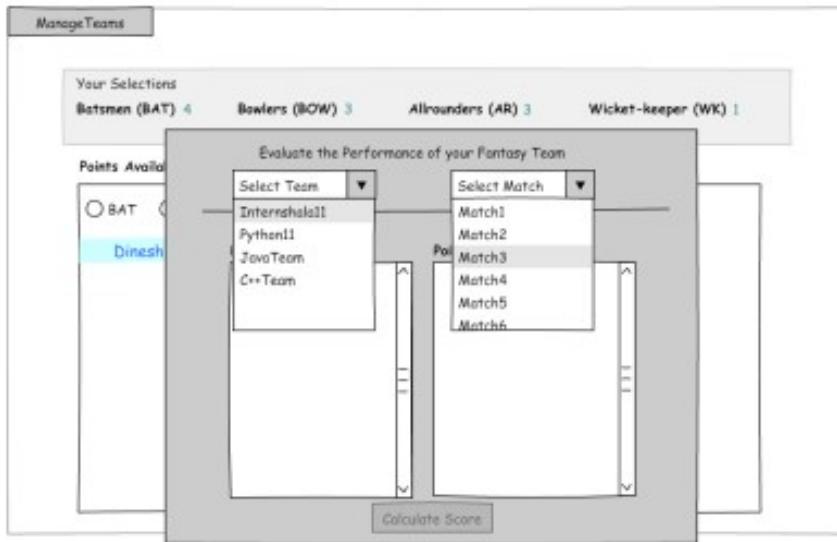
You can't select more than one wicket-keeper.

OK

Team Name InternshalaII

Virat Kohli
Dhawan
Rahane
Singh
Kumar
Yadav
Kedar Jadhav
Hardik Pandya
M.S.Dhoni
Ravindra Jadeja
Jasprit Bumrah

5 - Message if the game logic is not followed



To calculate the points for each player, you 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, you are required to use three tables – match, stats and teams.

match

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

*Run Out

teams

name	players	value

stats

player	matches	runs	100s	50s	value	ctg

Note: The teams table will be populated after score calculation. 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	rns	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	3569	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
Choni	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	130	1914	0	10	AR
Umesh	0	0	0	0	54	0	50	4	1	0	0	110	206	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	299	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

4.1 Scope of the Work

The project can be used in future online gaming systems for building the teams and playing online and also for calculating their values. Everything can be retrieved from the database.

4.2 Project Modules

- New team
- Open Team
- Save Team
- Evaluate team

5 Implementation Language

Python (v:3)

6 Description about language

History of Python: The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to ABC capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). (However, van Rossum stepped down as leader on July 12, 2018. Python was named after the BBC TV show Monty Python's Flying Circus.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible, though now-unsupported, Python 2.6 and 2.7.

Python 3.0 broke backward compatibility, and much Python 2 code does not run unmodified on Python 3. Python's dynamic typing combined with the plans to change the semantics of certain methods of dictionaries, for example, made perfect mechanical translation from Python 2.x to Python 3.0 very difficult. A tool called "2to3" does the parts of translation that can be done automatically. At this, 2to3 appeared to be fairly successful, though an early review noted that there were aspects of translation that such a tool would never be able to handle.^[33] Prior to the roll-out of Python 3, projects requiring compatibility with both the 2.x and 3.x series were recommended to have one source (for the 2.x series), and produce releases for the Python 3.x platform using 2to3. Edits to the Python 3.x code were discouraged for so long as the code needed to run on Python 2.x. This is no longer recommended; as of 2012 the preferred approach was to create a single code base that can run under both Python 2 and 3 using compatibility modules.

What the PYTHON is?

Python is an interpreted high-level general-purpose programming language.

Its design philosophy emphasizes code readability with its use of significant indentation.

Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Data Types in Python:

Like other programming languages, Python also have its own data types as follows:

- Python List

Lists are used to store multiple items in a single variable.

Lists are created using square brackets:

```
thislist = ["apple", "banana", "cherry"]
```

- Python Dictionary

Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

Dictionaries are written with curly brackets, and have keys and values:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

- Python Tuple

Tuples are used to store multiple items in a single variable.

Tuple is one of 4 built-in data types in Python used to store collections of data. A tuple is a collection which is ordered and unchangeable. Tuples are written with round brackets.

```
thistuple = ("apple", "banana", "cherry")
```

- Python Sets

Sets are used to store multiple items in a single variable. A set is a collection which is *unordered*, *unchangeable**, and *unindexed*. Sets are written with curly brackets.

```
thisset = {"apple", "banana", "cherry"}
```

Python Operators:

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

Conditional statement in Python:

Python also have conditional statements in the procedural programming.

- if
- elif
- else

Loops in Python:

Python includes iterative programming methods.

- While
- For

Other Features of Python:

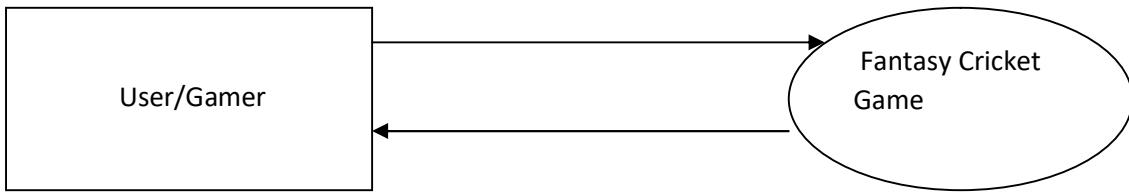
Python includes functional approaches, Object Oriented Programming approach, class and objects, inheritance Modules i.e: we need to import the module in inline or into the file code line.

Python also have the libraries to work with data science, Machine learning and the Deep learning, Computer Vision.

Python also have libraries to develop and deploy the GUI based application and the web designing also.

Python can also work with databases to insert and retrieve the data from the python inline commands only no need to get deep dive into database softwares itself.

7 Implementation Methodology



8 Technologies to be used

8.1 Software Platform

a) Front-end

Python, QtDesigner and functions of PyQt5 module.

b) Back-end

Sqlite3

8.2 Hardware Platform

- RAM
- Hard Disk
- OS
- Python

8.3 Tools, if any

- QtDesigner
- PyQt5
- SQL Studio

9 Description of the database

What the Data Base is?

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

What is Structured Query Language (SQL)?

SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.

SQLite:

SQLite3 is a compact free database you can use easily create and use a database. Though SQLite3 is not a full-featured database, it supports a surprisingly large set of the SQL standard, and is *ideal for those just starting to learn SQL* as well for developers that need a simple database engine to plug into their applications. As such, SQLite has become very popular with smart phone developers. The first is that you don't need to be an expert to install and configure it. In fact, getting SQLite3 to run is as simple as downloading the program and then running a simple command. Second, the software is a really simple command line interface. Now you may think that is a fault, but it isn't, as our goal is to learn SQL. And to really learn SQL you need to understand its commands and instructions (syntax). Also, SQLite runs many different computer systems such as Apple OS X, Linux, and Windows. some of the ways SQLite is suited for learning SQL; however, its benefits go beyond that. Though SQLite is a compact database, it is meant for serious applications! Airbus, the airplane manufacturer, even uses it on some flight software for their A350 line of aircraft. Some benefits of SQLite3 include:

- Small and Self-Contained

No additional programs or components are required for it to run. The database engine can run on any modern PC or Smartphone

SQLite doesn't require a separate computer process to run. IT doesn't rely on windows services, background daemons, nor separate computer hardware.

- Portable

It is really simple to share databases. Just copy one file to do so. This is what makes it easy for me to provide you with the essential SQL sample database.

SQLite Cons

There are disadvantages to SQLite. Since it is so compact and meant to run in a small space, some trade-offs were made.

- Limited Data types

SQLite doesn't support the date or timestamp data types. This is a disadvantage as many SQL puzzles, thus learning opportunities, are out of our reach in the beginning.

- SQL Language Support

There are several features not supported in SQLite. Right Outer Joins aren't supported nor are Full Outer Joins. Since Left Outer Joins are supported, and are similar to Right Outer Joins, the opportunity to learn isn't lost.

It also isn't easy to alter an existing table nor are there some referential integrity checks. These are pretty important features to have to manage a production database.

- Size Limitations

SQLite isn't meant to support extremely large databases. It isn't able to scale to hundreds of users nor store gigabytes of data. It isn't suitable where you expect a large number of users to simultaneously modify data (high concurrency), nor where there is a large volume of transactions.

Why SQLite?

- SQLite does not require a separate server process or system to operate (serverless).
- SQLite comes with zero-configuration, which means no setup or administration needed.
- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.
- SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

10 Description of GUI library

What the GUI is?

The graphical user interface (GUI jee-you-eye or) is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

PyQt:

PyQt is a GUI widgets toolkit. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt was developed by RiverBank Computing Ltd. PyQt API is a set of modules containing a large number of classes and functions. While QtCore module contains non-GUI functionality for working with file and directory etc., QtGui module contains all the graphical controls. In addition, there are modules for working with XML (QtXml), SVG (QtSvg), and SQL (QtSql), etc. PyQt is compatible with all the popular operating systems including Windows, Linux, and Mac OS. It is dual licensed, available under GPL as well as commercial license. PyQt submodules or widgets:

- **QtCore**
contains the core non-GUI code.
- **QtGui**
Has everything for window management like event handling and graphics.
- **QtWidgets**
Has a many UI widgets like buttons, labels, textinput and other things you'd see in a desktop window.
- **QtMultimedia**
For multimedia content and camera.
- **QtBluetooth**
Scan bluetooth devices and connect.
- **QtNetwork**
A cross-platform solution for network programming. Set up a socket server or client that works on all desktop systems. Supports both the TCP/IP stack and UDP.
- **QtPositioning**
Determine a position by using a position (WiFi, Satellite)
- **QtWebSockets**
Implementation of the websocket protocol.

- **QtWebKit**

Web browser implementation. You can use this to render a webpage. This is based on WebKit2. WebKit is used in the Safari browser, by KDE and others.

- **QtWebKitWidgets**

Deprecated. WebKit1 version of web browser implementation

- **QtXml**

Use XML files, reading/writing and so on.

- **QtSvg**

Svg graphics (Scalable Vector Graphics (SVG)). A type of image format.

- **QtSql**

Work with databases.

- **QtTest**

Unit testing

PyQt version used:

I have used PyQt5 for the development of the application using python.

QtDesigner:

The PyQt installer comes with a GUI builder tool called **Qt Designer**. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

Creation of a GUI interface using Qt Designer starts with choosing a top level window for the application.

You can then drag and drop required widgets from the widget box on the left pane. You can also assign value to properties of widget laid on the form.

The designed form is saved as demo.ui. This ui file contains XML representation of widgets and their properties in the design. This design is translated into Python

equivalent by using pyuic4 command line utility. This utility is a wrapper for uic module. The usage of pyuic4 is as follows –

```
pyuic4 -x demo.ui -o demo.py
```

11 Advantages of this Project

Cricket being an integral part and parcel of life, all fantasy cricket games apps are popular among the general Indian population. However, there's many who still wonders about the benefits behind indulging in such online games. Passion for cricket combined with extensive knowledge and backed with cash prizes are enough reasons to trigger the desire for playing fantasy cricket in apps.

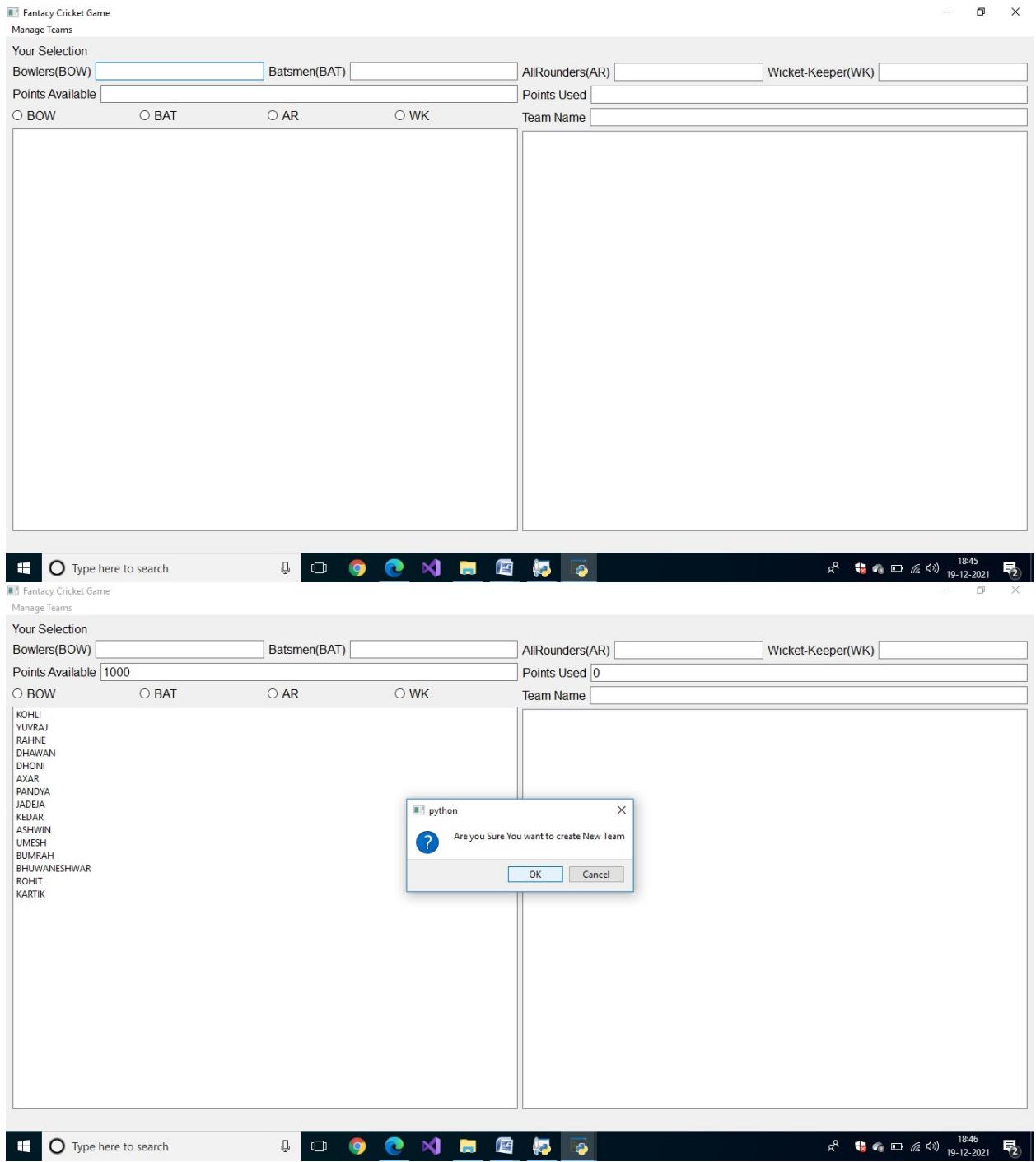
Some people get an adrenaline rush by playing online fantasy cricket. The love for challenges provokes the cricket lovers to start this type of game. However, if you're still having double thoughts about the same, then here's a list of benefits that will surely help you set your mind for the sports.

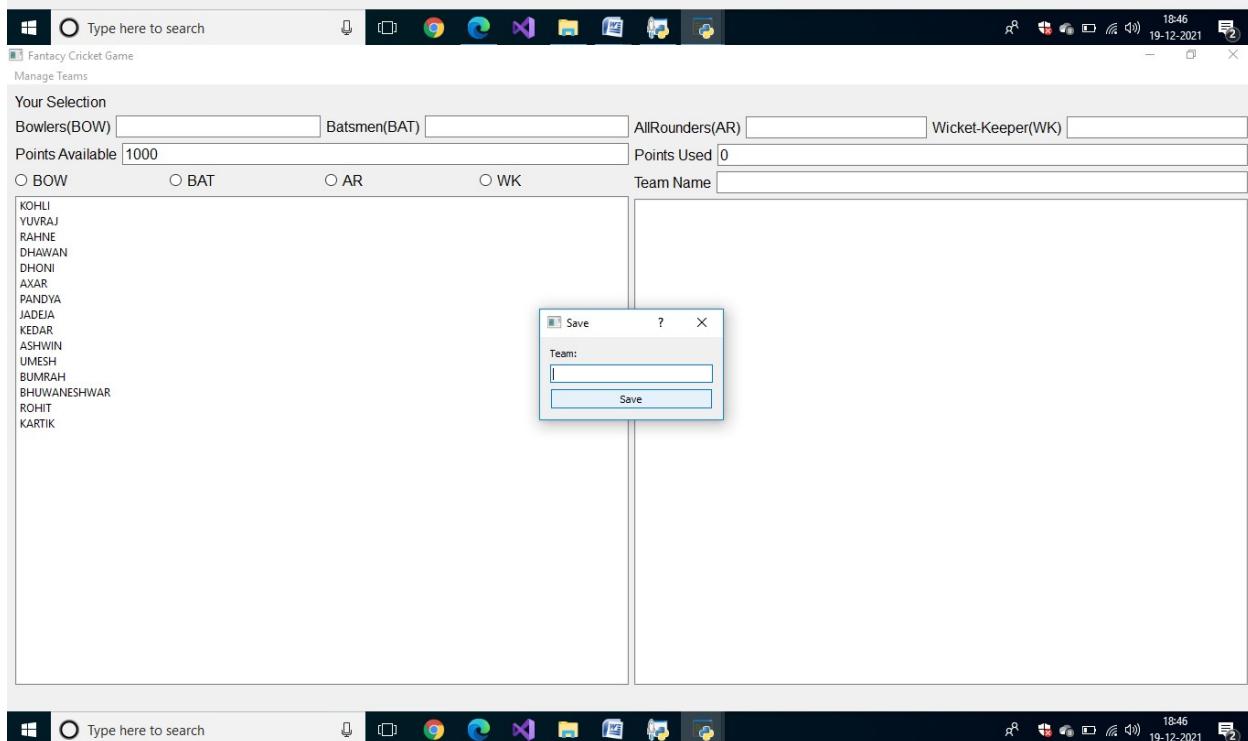
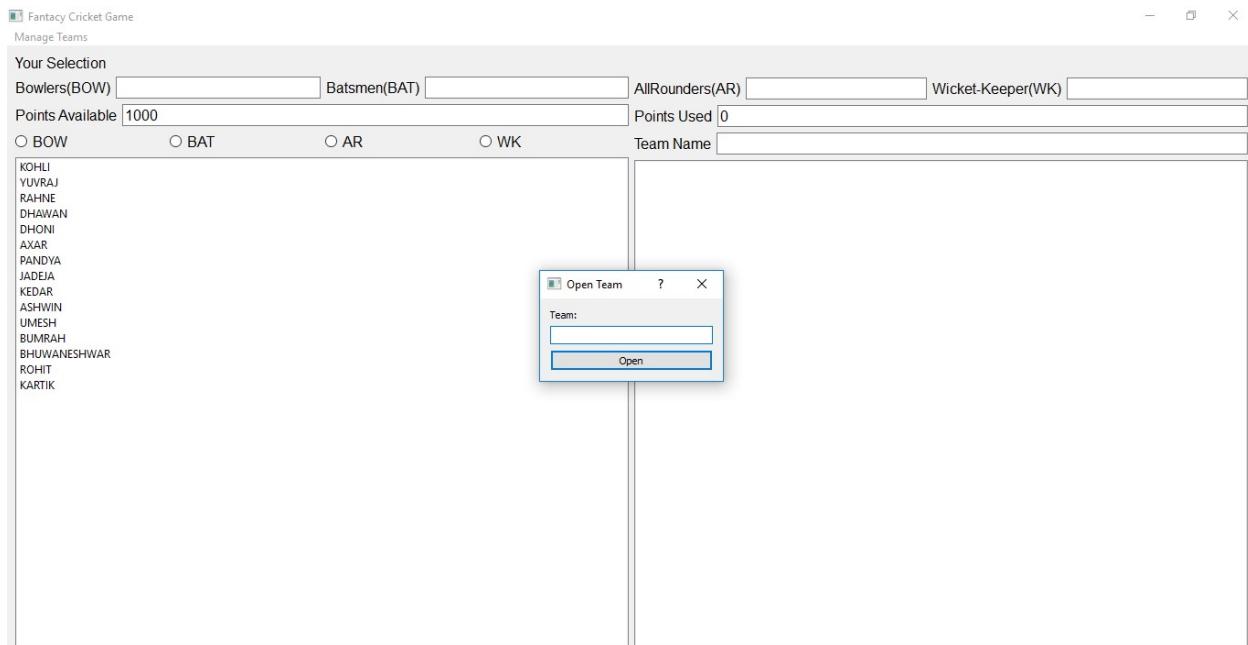
- Playing fantasy cricket helps in broadening the knowledge about the game since the selectors are required to make extensive research before playing the game.
- Ability to show your cricketing skills by creating teams and selecting the suitable players.
- High excitement and challenging game giving the brains a booster.
- Helps in improvement of decision-making ability in personal life.
- Makes you good in time management strategies and increases your risk-taking abilities.
- Reduces boredom of daily life and induces entertainment factor.
- Provides the opportunity of earning money while playing the game.

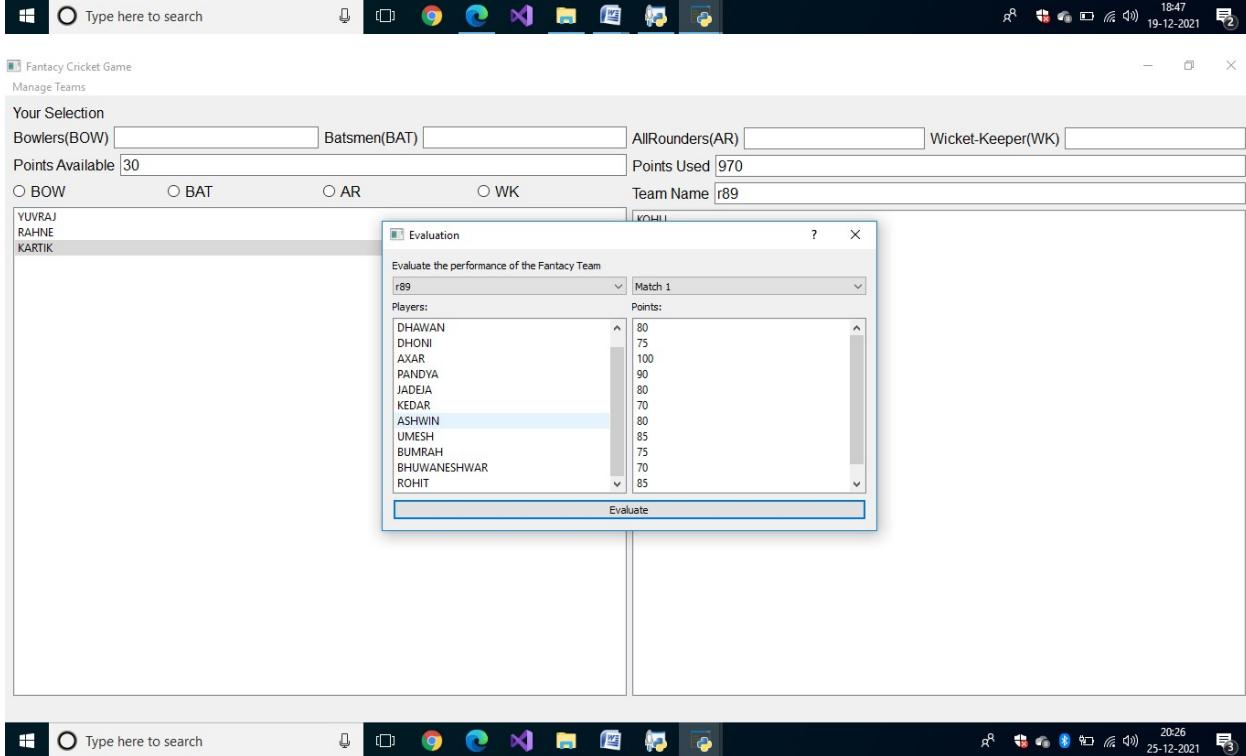
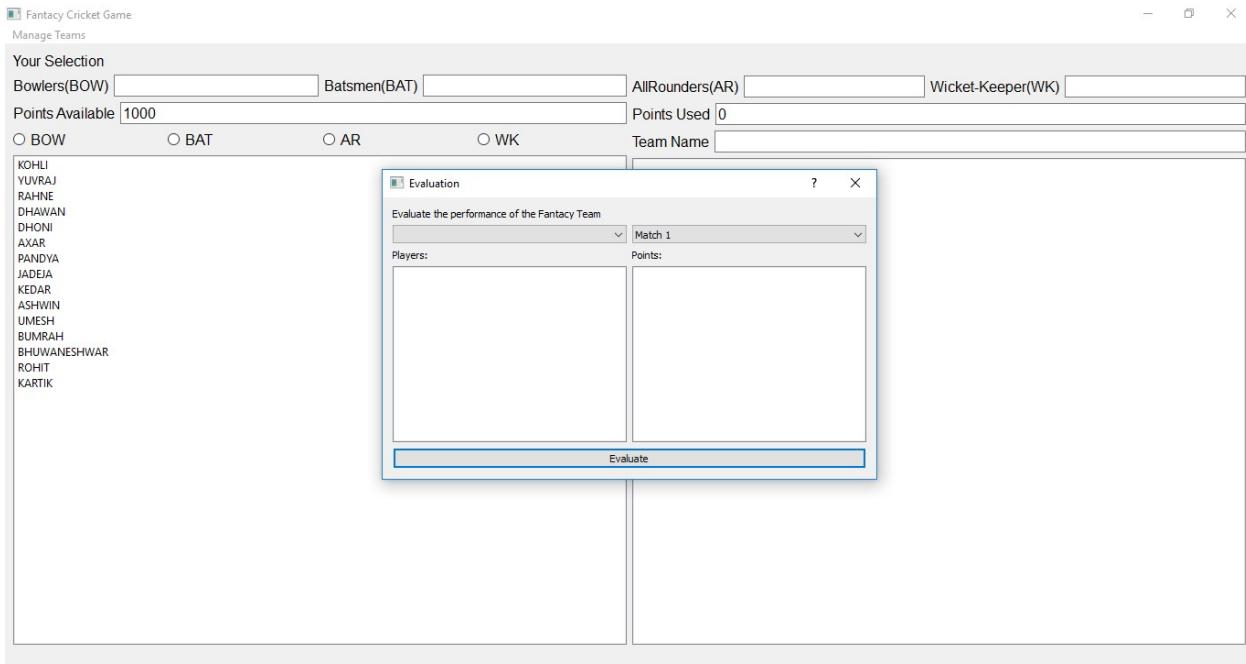
12 Future Scope and further enhancement of the Project

In future, this project may include the module that will display the name of teams and the players in them that will help the user to display the data in application window only there will not be any need to open database again and again to access the data.

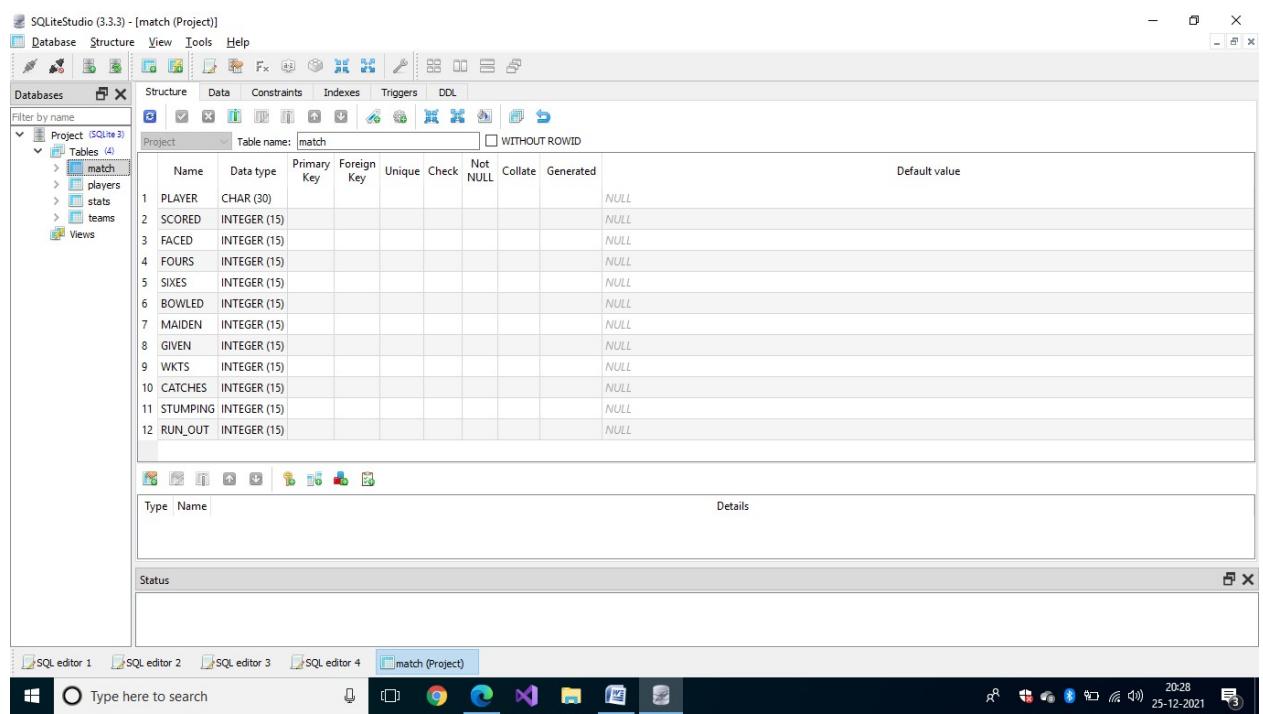
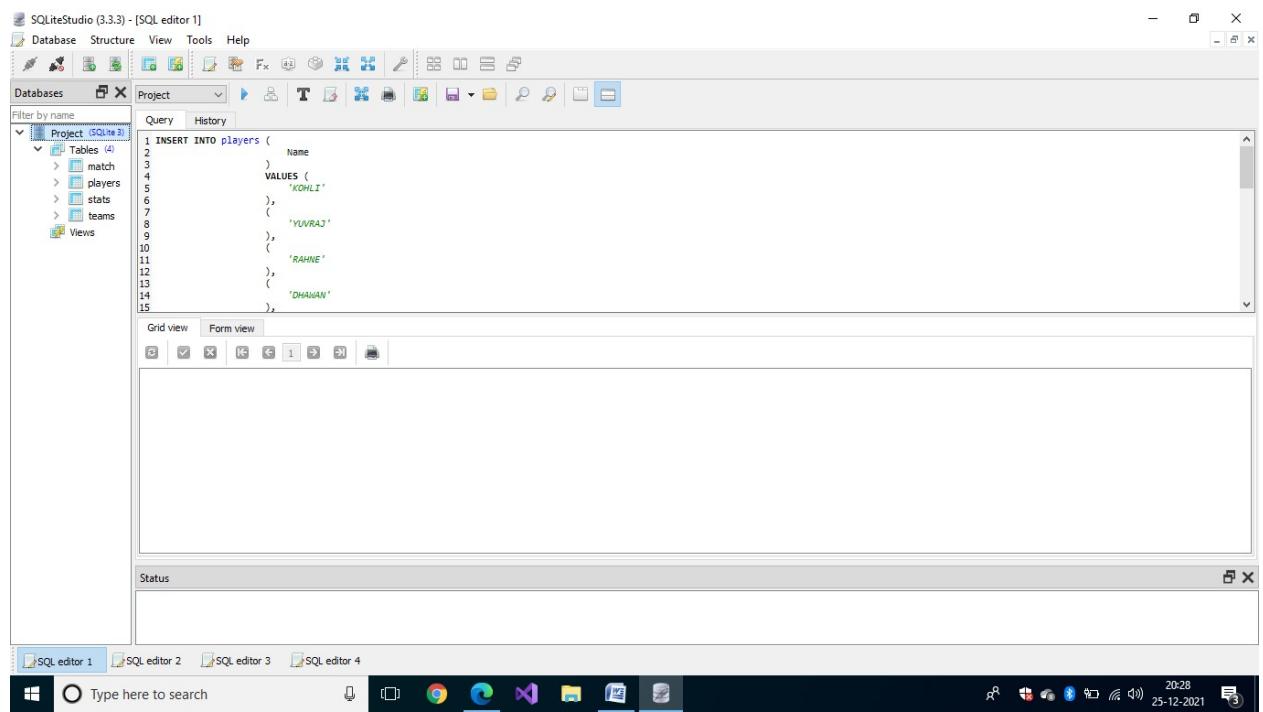
13 GUI of the application

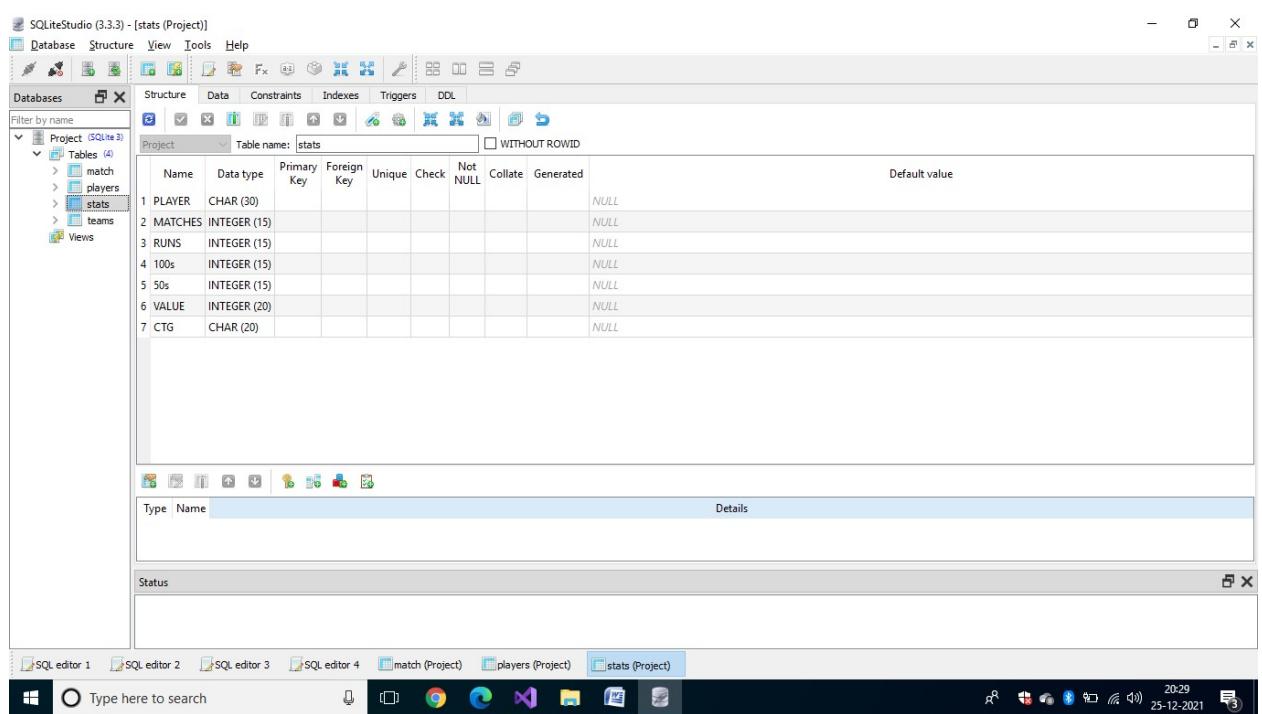
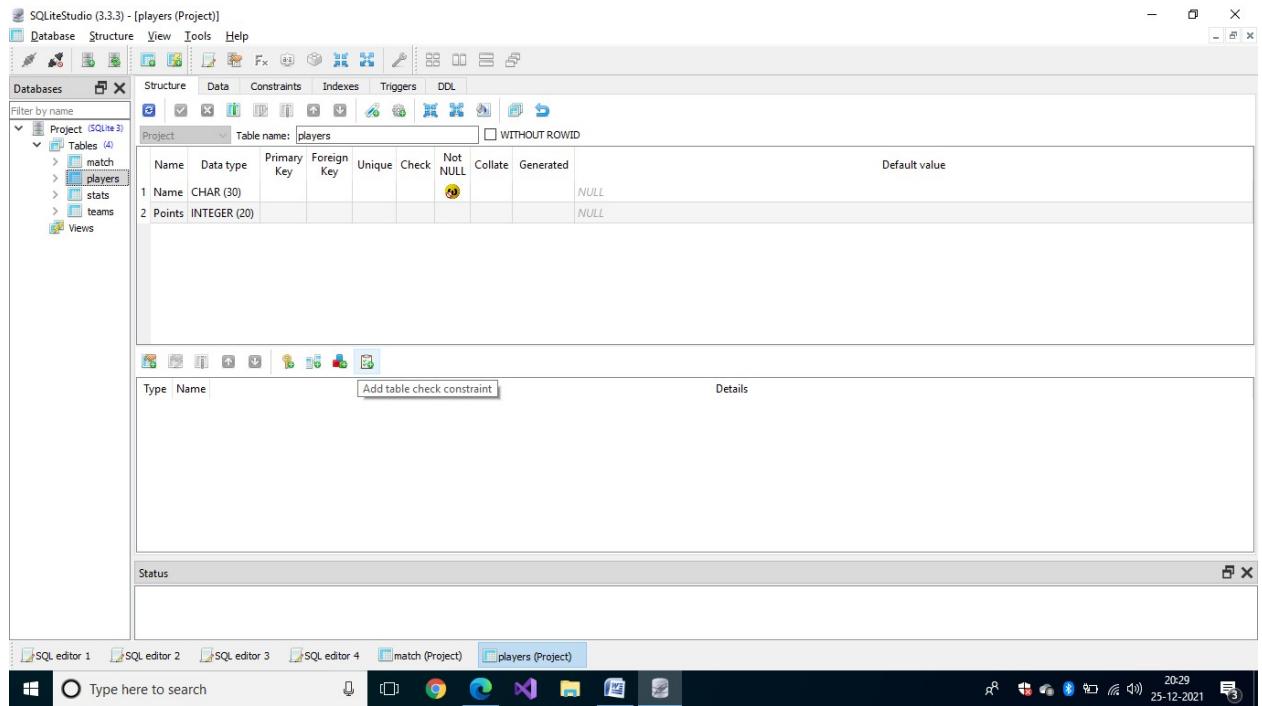


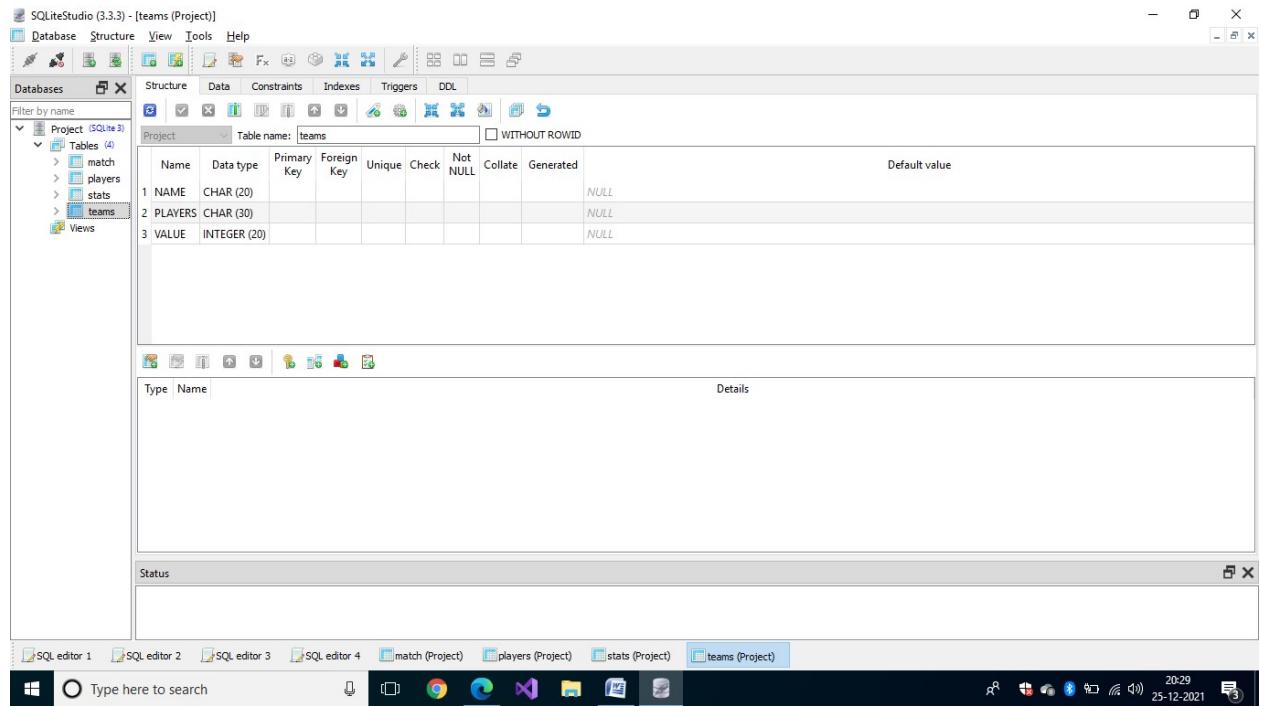




14 GUI of Database







15 Conclusion

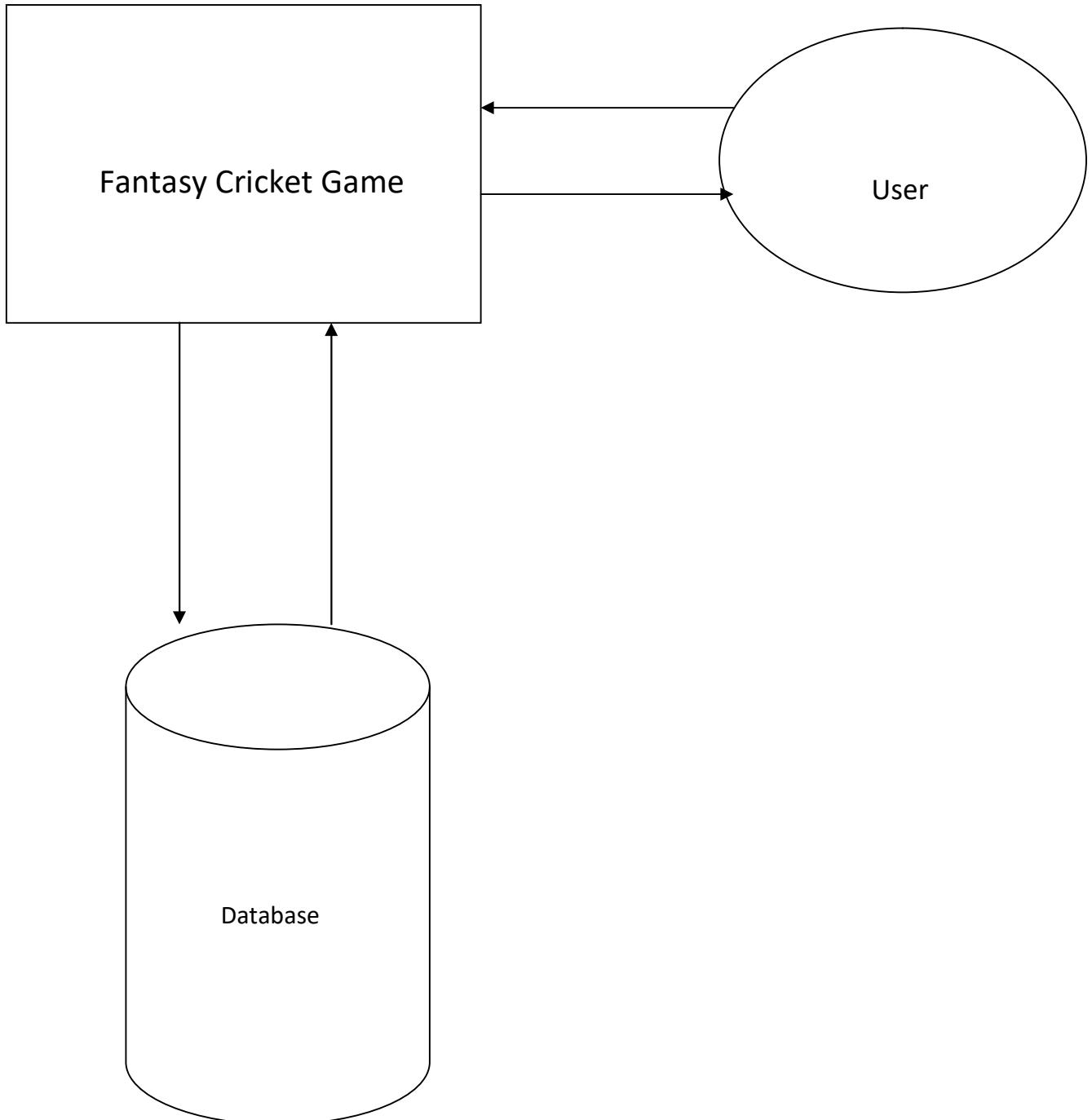
The main objective of this project is to provide base line application which will be user friendly and very much simple to use. The main feature include the evaluation of teams and saving the team into the data base.

16 References

S#	Reference Details	Owner	Date
1.	Project Synopsis	Rishant Rajpoot	10-09-21
2.	Project Requirements	Mr. Sarvesh Agarwal	15-07-21

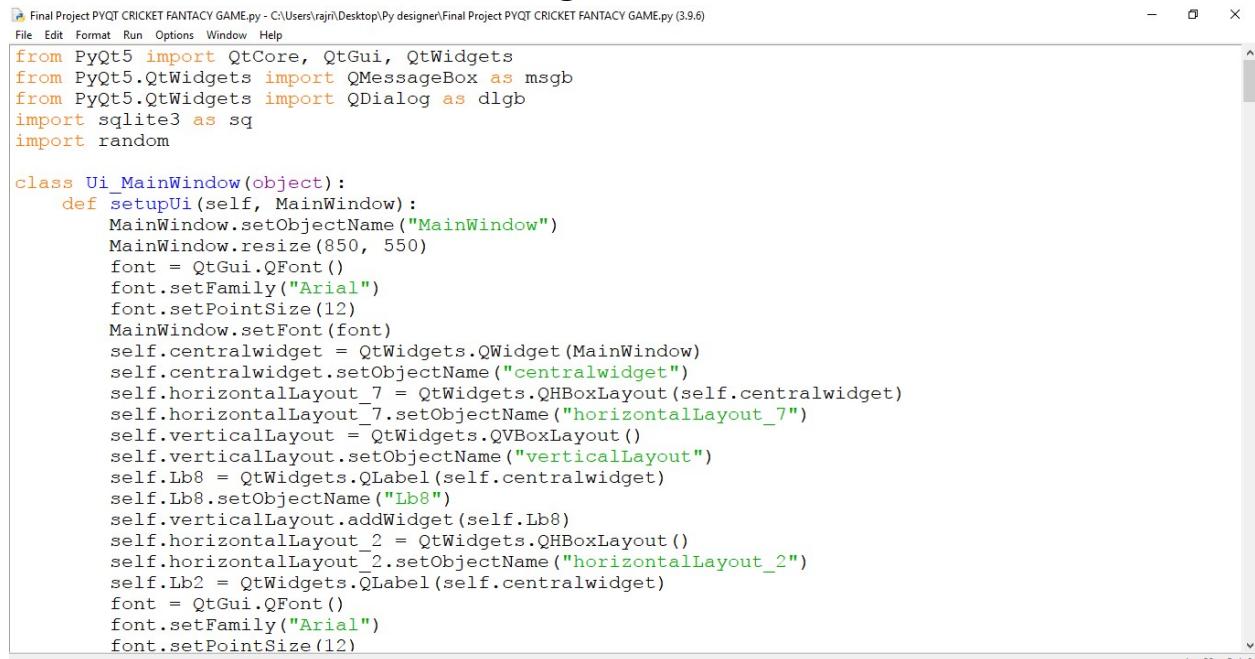
Annexure A

Data Flow Diagram (DFD)



Annexure:C

Coding Screenshots



```

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox as msgb
from PyQt5.QtWidgets import QDialog as dlgb
import sqlite3 as sq
import random

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(850, 550)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPointSize(12)
        MainWindow.setFont(font)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.horizontalLayout_7 = QtWidgets.QHBoxLayout(self.centralwidget)
        self.horizontalLayout_7.setObjectName("horizontalLayout_7")
        self.verticalLayout = QtWidgets.QVBoxLayout()
        self.verticalLayout.setObjectName("verticalLayout")
        self.Lb8 = QtWidgets.QLabel(self.centralwidget)
        self.Lb8.setObjectName("Lb8")
        self.verticalLayout.addWidget(self.Lb8)
        self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_2.setObjectName("horizontalLayout_2")
        self.Lb2 = QtWidgets.QLabel(self.centralwidget)
        self.Lb2.setObjectName("Lb2")
        self.horizontalLayout_2.addWidget(self.Lb2)
        self.t1 = QtWidgets.QLineEdit(self.centralwidget)
        self.t1.setEnabled(True)
        self.t1.setObjectName("t1")
        self.horizontalLayout_2.addWidget(self.t1)
        self.Lbl = QtWidgets.QLabel(self.centralwidget)
        self.Lbl.setObjectName("Lbl")
        self.horizontalLayout_2.addWidget(self.Lbl)
        self.t2 = QtWidgets.QLineEdit(self.centralwidget)
        self.t2.setObjectName("t2")
        self.horizontalLayout_2.addWidget(self.t2)
        self.verticalLayout.addLayout(self.horizontalLayout_2)
        self.horizontalLayout_7.addLayout(self.verticalLayout)
        MainWindow.setCentralWidget(self.centralwidget)
        self.retranslateUi(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.Lb8.setText(_translate("MainWindow", "Label"))
        self.Lb2.setText(_translate("MainWindow", "Label"))
        self.t1.setPlaceholderText(_translate("MainWindow", "Text Input"))
        self.Lbl.setText(_translate("MainWindow", "Label"))
        self.t2.setPlaceholderText(_translate("MainWindow", "Text Input"))

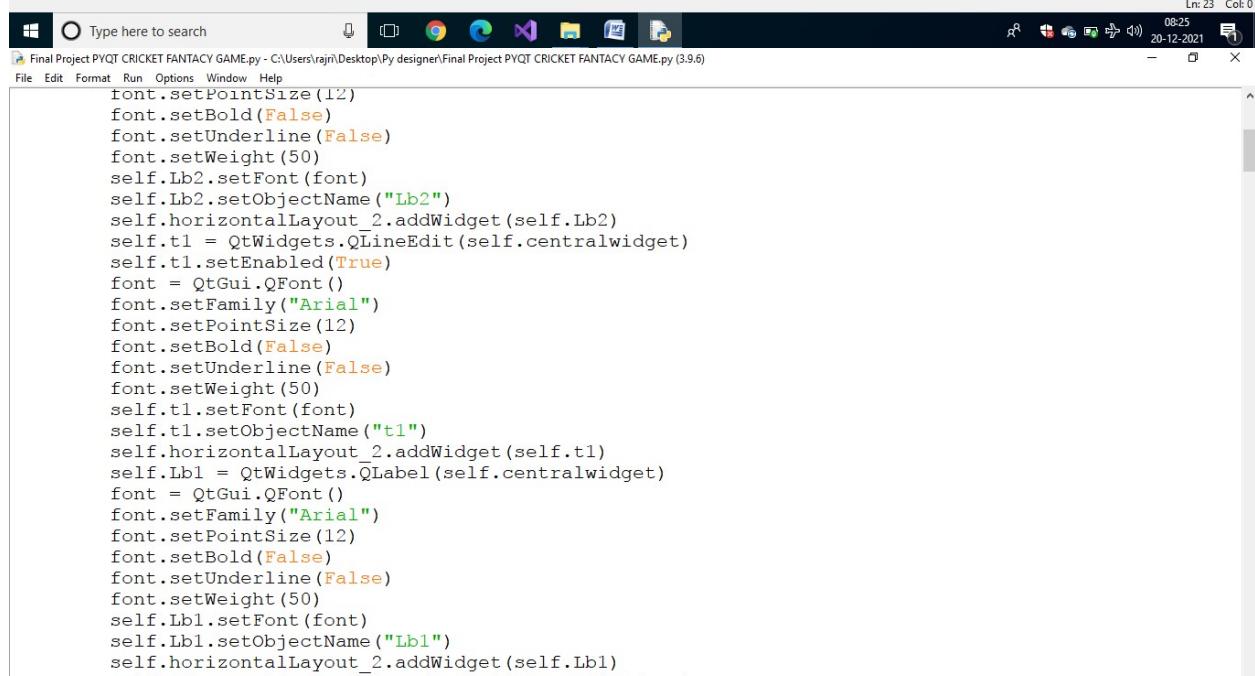
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(850, 550)
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(False)
        font.setUnderline(False)
        font.setWeight(50)
        self.Lb2.setFont(font)
        self.Lb2.setObjectName("Lb2")
        self.horizontalLayout_2.addWidget(self.Lb2)
        self.t1 = QtWidgets.QLineEdit(self.centralwidget)
        self.t1.setEnabled(True)
        self.t1.setObjectName("t1")
        self.horizontalLayout_2.addWidget(self.t1)
        self.Lbl = QtWidgets.QLabel(self.centralwidget)
        self.Lbl.setObjectName("Lbl")
        self.horizontalLayout_2.addWidget(self.Lbl)
        self.t2 = QtWidgets.QLineEdit(self.centralwidget)
        self.t2.setObjectName("t2")
        self.horizontalLayout_2.addWidget(self.t2)
        self.verticalLayout.addLayout(self.horizontalLayout_2)
        self.horizontalLayout_7.addLayout(self.verticalLayout)
        MainWindow.setCentralWidget(self.centralwidget)
        self.retranslateUi(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.Lb8.setText(_translate("MainWindow", "Label"))
        self.Lb2.setText(_translate("MainWindow", "Label"))
        self.t1.setPlaceholderText(_translate("MainWindow", "Text Input"))
        self.Lbl.setText(_translate("MainWindow", "Label"))
        self.t2.setPlaceholderText(_translate("MainWindow", "Text Input"))

    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(850, 550)
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(False)
        font.setUnderline(False)
        font.setWeight(50)
        self.t1.setFont(font)
        self.t1.setObjectName("t1")
        self.horizontalLayout_2.addWidget(self.t1)
        self.Lbl = QtWidgets.QLabel(self.centralwidget)
        self.Lbl.setObjectName("Lbl")
        self.horizontalLayout_2.addWidget(self.Lbl)
        self.t2 = QtWidgets.QLineEdit(self.centralwidget)
        self.t2.setObjectName("t2")
        self.horizontalLayout_2.addWidget(self.t2)
        self.verticalLayout.addLayout(self.horizontalLayout_2)
        self.horizontalLayout_7.addLayout(self.verticalLayout)
        MainWindow.setCentralWidget(self.centralwidget)
        self.retranslateUi(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.t1.setPlaceholderText(_translate("MainWindow", "Text Input"))
        self.Lbl.setText(_translate("MainWindow", "Label"))
        self.t2.setPlaceholderText(_translate("MainWindow", "Text Input"))

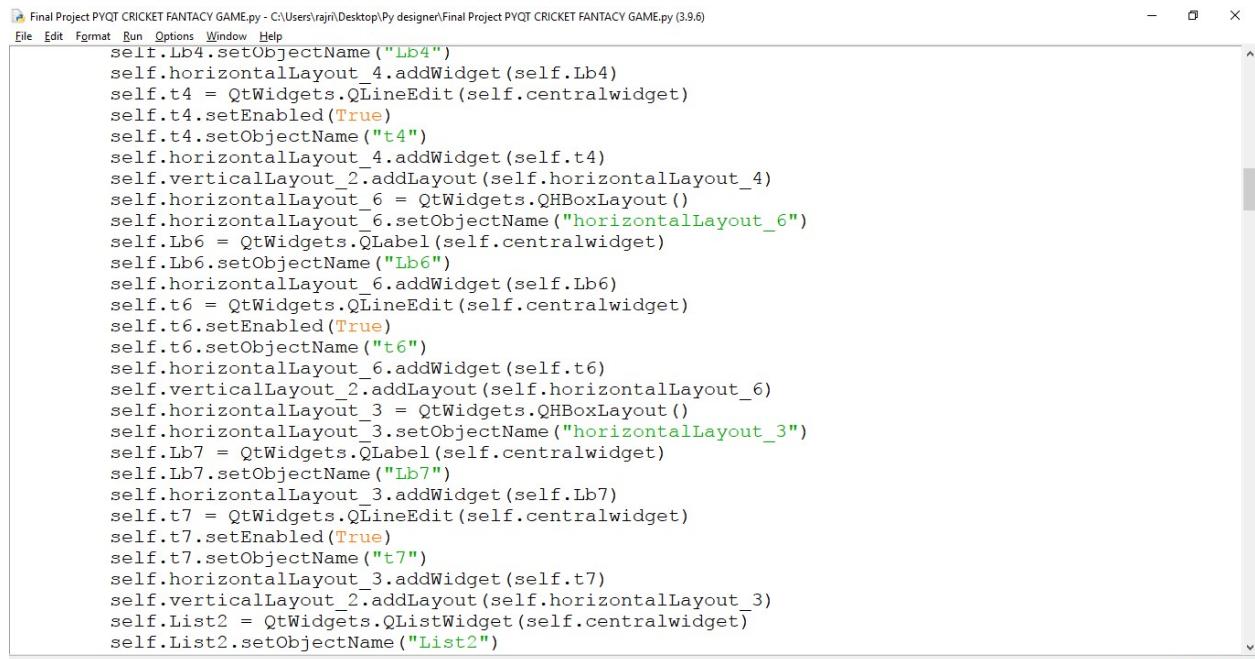
```

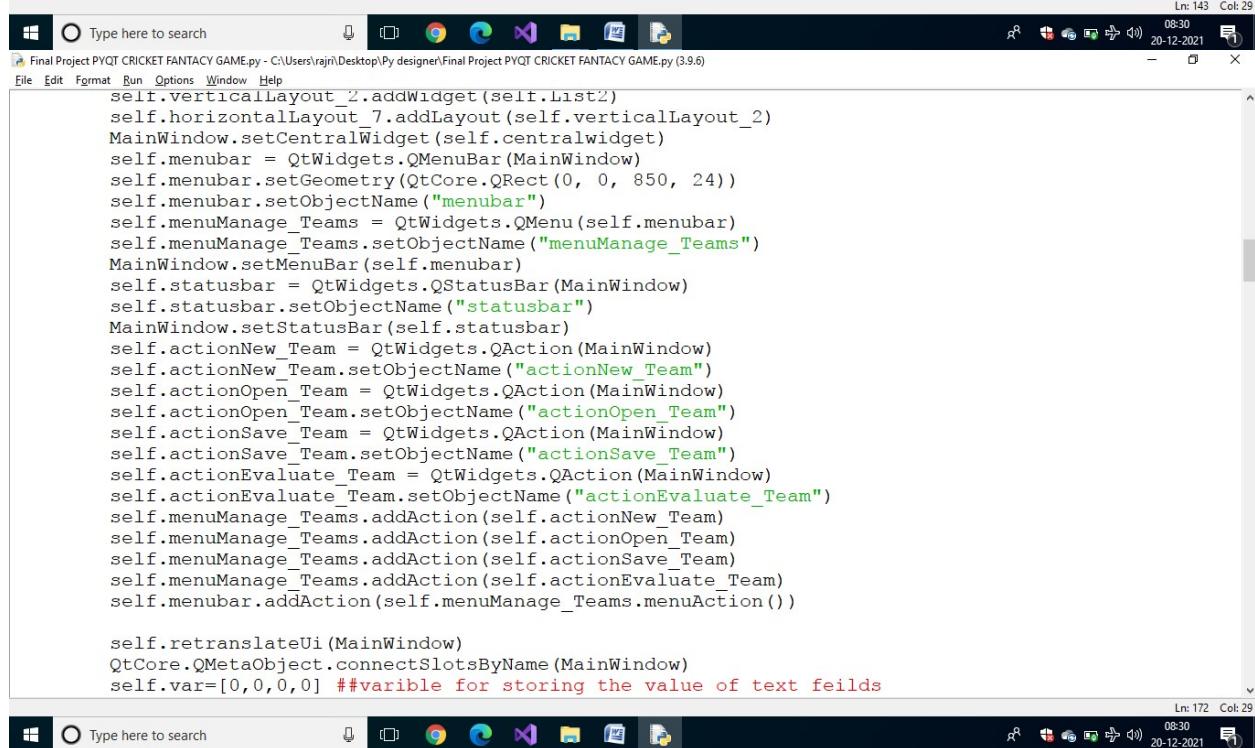


```
Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajni\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
    self.t2 = QtWidgets.QLineEdit(self.centralwidget)
    self.t2.setEnabled(True)
    font = QtGui.QFont()
    font.setFamily("Arial")
    font.setPointSize(12)
    font.setBold(False)
    font.setUnderline(False)
    font.setWeight(50)
    self.t2.setFont(font)
    self.t2.setObjectName("t2")
    self.horizontalLayout_2.addWidget(self.t2)
    self.verticalLayout.addLayout(self.horizontalLayout_2)
    self.horizontalLayout_5 = QtWidgets.QHBoxLayout()
    self.horizontalLayout_5.setObjectName("horizontalLayout_5")
    self.horizontalLayout_5.addWidget(self.Lb5)
    self.Lb5 = QtWidgets.QLabel(self.centralwidget)
    self.Lb5.setObjectName("Lb5")
    self.horizontalLayout_5.addWidget(self.Lb5)
    self.t5 = QtWidgets.QLineEdit(self.centralwidget)
    self.t5.setEnabled(True)
    self.t5.setObjectName("t5")
    self.horizontalLayout_5.addWidget(self.t5)
    self.verticalLayout.addLayout(self.horizontalLayout_5)
    self.horizontalLayout = QtWidgets.QHBoxLayout()
    self.horizontalLayout.setObjectName("horizontalLayout")
    self.horizontalLayout.addWidget(self.rb2)
    self.rb2 = QtWidgets.QRadioButton(self.centralwidget)
    self.rb2.setObjectName("rb2")
    self.horizontalLayout.addWidget(self.rb2)
    self.rbl = QtWidgets.QRadioButton(self.centralwidget)
    self.rbl.setIconSize(QtCore.QSize(10, 16))
    self.horizontalLayout.addWidget(self.rbl)
```

```
Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
    self.rbl.setObjectName("rbl")
    self.horizontalLayout.addWidget(self.rbl)
    self.rb4 = QtWidgets.QRadioButton(self.centralwidget)
    self.rb4.setObjectName("rb4")
    self.horizontalLayout.addWidget(self.rb4)
    self.rb3 = QtWidgets.QRadioButton(self.centralwidget)
    self.rb3.setObjectName("rb3")
    self.horizontalLayout.addWidget(self.rb3)
    self.verticalLayout.addLayout(self.horizontalLayout)
    self.List1 = QtWidgets.QListWidget(self.centralwidget)
    self.List1.setObjectName("List1")
    self.verticalLayout.addWidget(self.List1)
    self.horizontalLayout_7.setLayout(self.verticalLayout)
    self.verticalLayout_2 = QtWidgets.QVBoxLayout()
    self.verticalLayout_2.setObjectName("verticalLayout_2")
    self.label = QtWidgets.QLabel(self.centralwidget)
    self.label.setText("")
    self.label.setObjectName("label")
    self.verticalLayout_2.addWidget(self.label)
    self.horizontalLayout_4 = QtWidgets.QHBoxLayout()
    self.horizontalLayout_4.setObjectName("horizontalLayout_4")
    self.Lb3 = QtWidgets.QLabel(self.centralwidget)
    self.Lb3.setObjectName("Lb3")
    self.horizontalLayout_4.addWidget(self.Lb3)
    self.t3 = QtWidgets.QLineEdit(self.centralwidget)
    self.t3.setEnabled(True)
    self.t3.setObjectName("t3")
    self.horizontalLayout_4.addWidget(self.t3)
    self.horizontalLayout_4.addWidget(self.Lb4 = QtWidgets.QLabel(self.centralwidget))
```

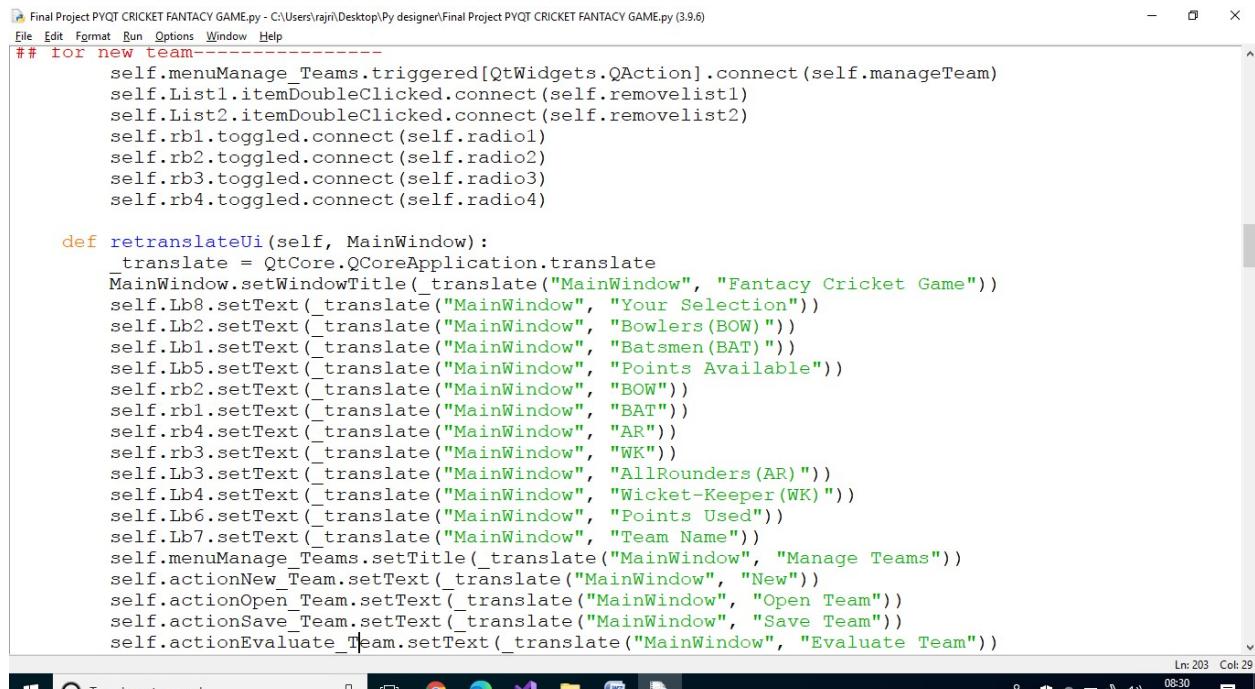
```


File Edit Format Run Options Window Help
self.Lb4.setObjectName("Lb4")
self.horizontalLayout_4.addWidget(self.Lb4)
self.t4 = QtWidgets.QLineEdit(self.centralwidget)
self.t4.setEnabled(True)
self.t4.setObjectName("t4")
self.horizontalLayout_4.addWidget(self.t4)
self.verticalLayout_2.addLayout(self.horizontalLayout_4)
self.horizontalLayout_6 = QtWidgets.QHBoxLayout()
self.horizontalLayout_6.setObjectName("horizontalLayout_6")
self.Lb6 = QtWidgets.QLabel(self.centralwidget)
self.Lb6.setObjectName("Lb6")
self.horizontalLayout_6.addWidget(self.Lb6)
self.t6 = QtWidgets.QLineEdit(self.centralwidget)
self.t6.setEnabled(True)
self.t6.setObjectName("t6")
self.horizontalLayout_6.addWidget(self.t6)
self.verticalLayout_2.addLayout(self.horizontalLayout_6)
self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
self.horizontalLayout_3.setObjectName("horizontalLayout_3")
self.Lb7 = QtWidgets.QLabel(self.centralwidget)
self.Lb7.setObjectName("Lb7")
self.horizontalLayout_3.addWidget(self.Lb7)
self.t7 = QtWidgets.QLineEdit(self.centralwidget)
self.t7.setEnabled(True)
self.t7.setObjectName("t7")
self.horizontalLayout_3.addWidget(self.t7)
self.verticalLayout_2.addLayout(self.horizontalLayout_3)
self.List2 = QtWidgets.QListWidget(self.centralwidget)
self.List2.setObjectName("List2")


File Edit Format Run Options Window Help
self.verticalLayout_2.addWidget(self.List2)
self.horizontalLayout_7.addWidget(self.verticalLayout_2)
MainWindow.setCentralWidget(self.centralwidget)
self.menuubar = QtWidgets.QMenuBar(MainWindow)
self.menuubar.setGeometry(QtCore.QRect(0, 0, 850, 24))
self.menuubar.setObjectName("menuubar")
self.menuManage_Teams = QtWidgets.QMenu(self.menuubar)
self.menuManage_Teams.setObjectName("menuManage_Teams")
MainWindow.setMenuBar(self.menuubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)
self.actionNew_Team = QtWidgets.QAction(MainWindow)
self.actionNew_Team.setObjectName("actionNew_Team")
self.actionOpen_Team = QtWidgets.QAction(MainWindow)
self.actionOpen_Team.setObjectName("actionOpen_Team")
self.actionSave_Team = QtWidgets.QAction(MainWindow)
self.actionSave_Team.setObjectName("actionSave_Team")
self.actionEvaluate_Team = QtWidgets.QAction(MainWindow)
self.actionEvaluate_Team.setObjectName("actionEvaluate_Team")
self.menuManage_Teams.addAction(self.actionNew_Team)
self.menuManage_Teams.addAction(self.actionOpen_Team)
self.menuManage_Teams.addAction(self.actionSave_Team)
self.menuManage_Teams.addAction(self.actionEvaluate_Team)
self.menuubar.addAction(self.menuManage_Teams.menuAction())

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
self.var=[0,0,0,0] ##variable for storing the value of text feilds

```

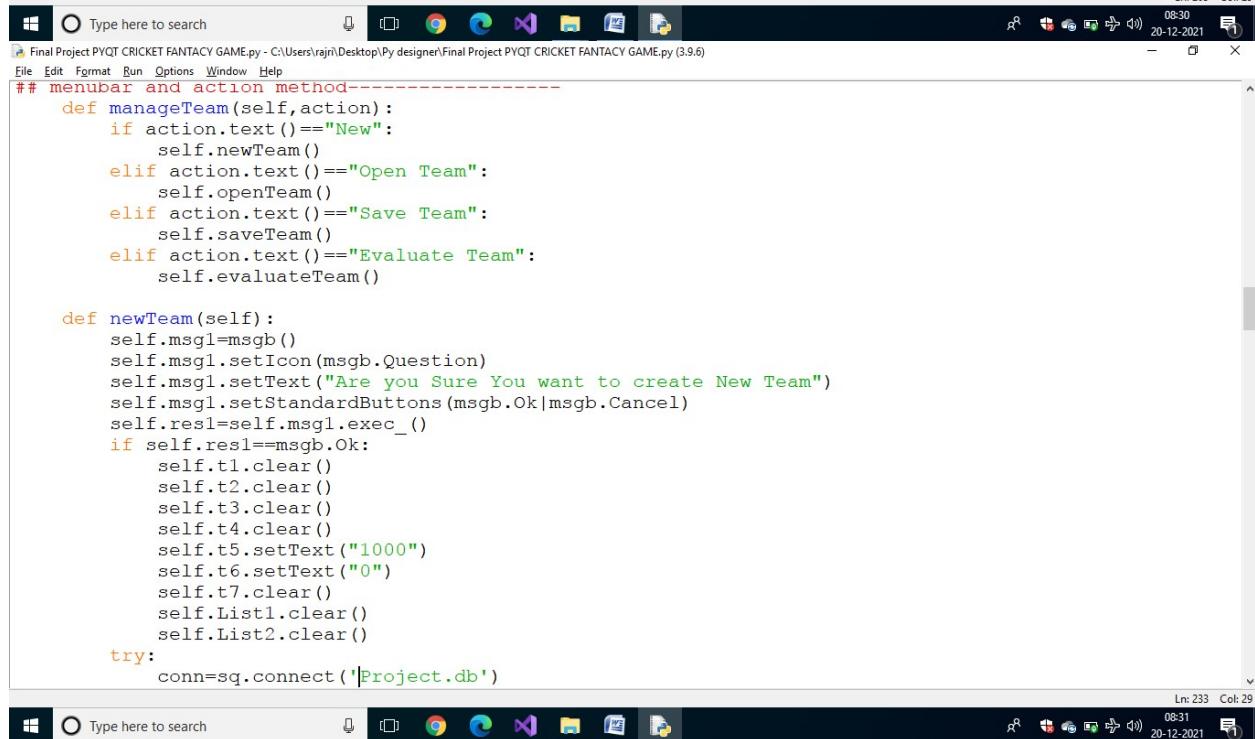


```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
## for new team-----
    self.menuManage_Teams.triggered[QtWidgets.QAction].connect(self.manageTeam)
    self.List1.itemDoubleClicked.connect(self.removeList1)
    self.List2.itemDoubleClicked.connect(self.removeList2)
    self.rbl.toggled.connect(self.radio1)
    self.rb2.toggled.connect(self.radio2)
    self.rb3.toggled.connect(self.radio3)
    self.rb4.toggled.connect(self.radio4)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Fantacy Cricket Game"))
    self.Lb8.setText(_translate("MainWindow", "Your Selection"))
    self.Lb2.setText(_translate("MainWindow", "Bowlers(BOW)"))
    self.Lb1.setText(_translate("MainWindow", "Batsmen(BAT)"))
    self.Lb5.setText(_translate("MainWindow", "Points Available"))
    self.rb2.setText(_translate("MainWindow", "BOW"))
    self.rbl.setText(_translate("MainWindow", "BAT"))
    self.rb4.setText(_translate("MainWindow", "AR"))
    self.rb3.setText(_translate("MainWindow", "WK"))
    self.Lb3.setText(_translate("MainWindow", "AllRounders(AR)"))
    self.Lb4.setText(_translate("MainWindow", "Wicket-Keeper(WK)"))
    self.Lb6.setText(_translate("MainWindow", "Points Used"))
    self.Lb7.setText(_translate("MainWindow", "Team Name"))
    self.menuManage_Teams.setTitle(_translate("MainWindow", "Manage Teams"))
    self.actionNew_Team.setText(_translate("MainWindow", "New"))
    self.actionOpen_Team.setText(_translate("MainWindow", "Open Team"))
    self.actionSave_Team.setText(_translate("MainWindow", "Save Team"))
    self.actionEvaluate_Team.setText(_translate("MainWindow", "Evaluate Team"))

```

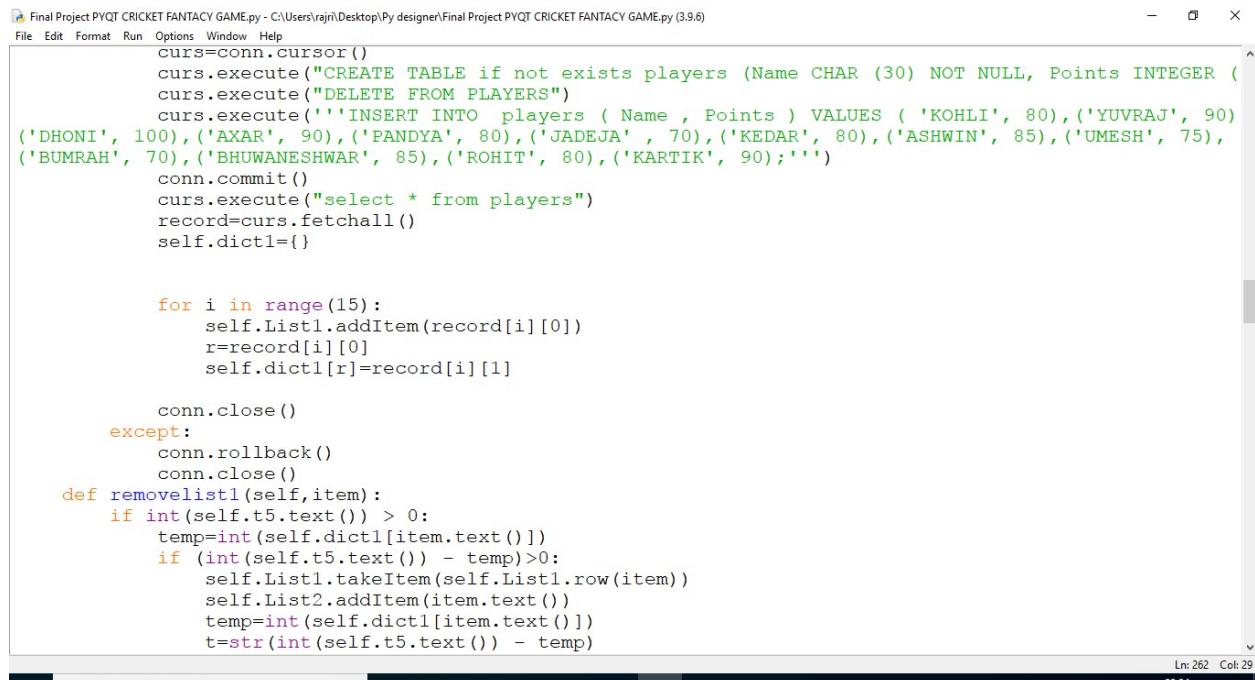


```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
## menubar and action method-----
def manageTeam(self,action):
    if action.text() == "New":
        self.newTeam()
    elif action.text() == "Open Team":
        self.openTeam()
    elif action.text() == "Save Team":
        self.saveTeam()
    elif action.text() == "Evaluate Team":
        self.evaluateTeam()

def newTeam(self):
    self.msg1=msgb()
    self.msg1.setIcon(msgb.Question)
    self.msg1.setText("Are you Sure You want to create New Team")
    self.msg1.setStandardButtons(msgb.Ok|msgb.Cancel)
    self.res1=self.msg1.exec_()
    if self.res1==msgb.Ok:
        self.t1.clear()
        self.t2.clear()
        self.t3.clear()
        self.t4.clear()
        self.t5.setText("1000")
        self.t6.setText("0")
        self.t7.clear()
        self.List1.clear()
        self.List2.clear()
    try:
        conn=sq.connect('Project.db')

```



```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajni\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
curs=conn.cursor()
curs.execute("CREATE TABLE if not exists players (Name CHAR (30) NOT NULL, Points INTEGER ")
curs.execute("DELETE FROM PLAYERS")
curs.execute('''INSERT INTO players ( Name , Points ) VALUES ( 'KOHLI' , 80 ), ('YUVRAJ' , 90)
('DHONI' , 100 ), ('AXAR' , 90 ), ('PANDYA' , 80 ), ('JADEJA' , 70 ), ('KEDAR' , 80 ), ('ASHWIN' , 85 ), ('UMESH' , 75),
('BUMRAH' , 70 ), ('BHUWANESHWAR' , 85 ), ('ROHIT' , 80 ), ('KARTIK' , 90 ); ''')
conn.commit()
curs.execute("select * from players")
record=curs.fetchall()
self.dict1={}

for i in range(15):
    self.List1.addItem(record[i][0])
    r=record[i][0]
    self.dict1[r]=record[i][1]

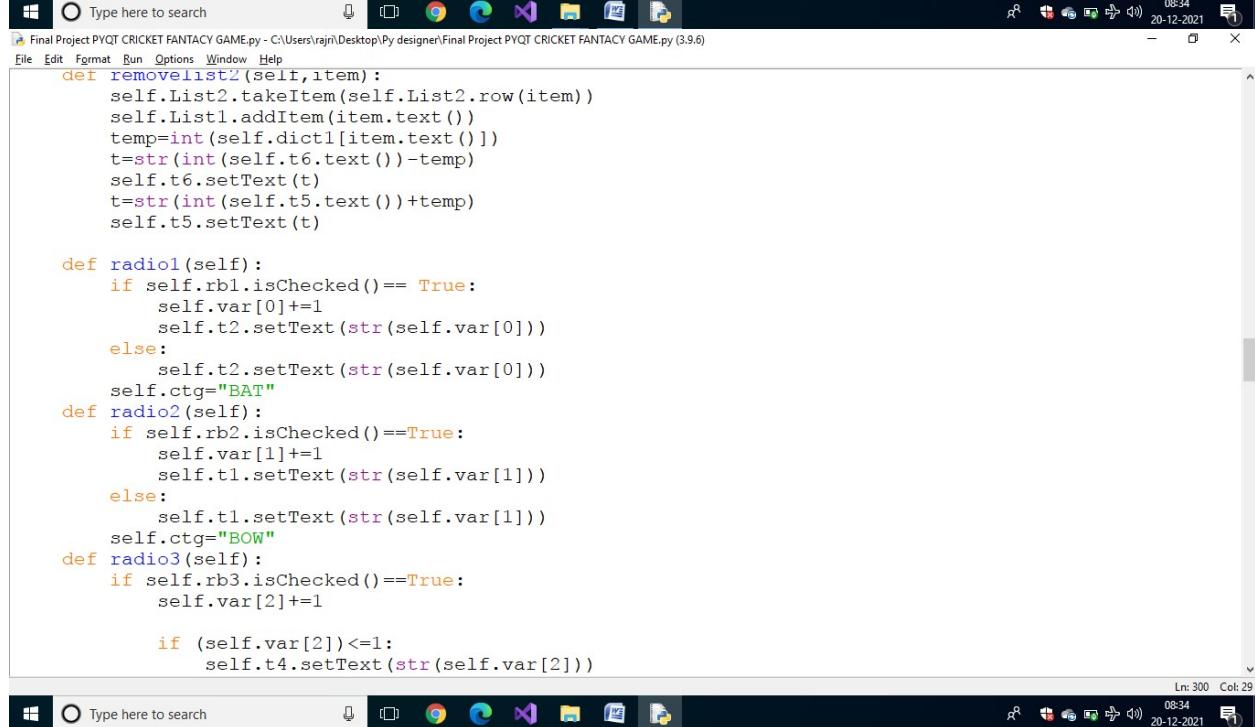
    conn.close()
except:
    conn.rollback()
    conn.close()
def removelist1(self,item):
    if int(self.t5.text()) > 0:
        temp=int(self.dict1[item.text()])
        if (int(self.t5.text()) - temp)>0:
            self.List1.takeItem(self.List1.row(item))
            self.List2.addItem(item.text())
            temp=int(self.dict1[item.text()])
            t=str(int(self.t5.text()) - temp)

def removelist2(self,item):
    self.List2.takeItem(self.List2.row(item))
    self.List1.addItem(item.text())
    temp=int(self.dict1[item.text()])
    t=str(int(self.t6.text())-temp)
    self.t6.setText(t)
    t=str(int(self.t5.text())+temp)
    self.t5.setText(t)

def radiol(self):
    if self.rbl.isChecked()== True:
        self.var[0]+=1
        self.t2.setText(str(self.var[0]))
    else:
        self.t2.setText(str(self.var[0]))
    self.ctg="BAT"
def radio2(self):
    if self.rb2.isChecked()==True:
        self.var[1]+=1
        self.t1.setText(str(self.var[1]))
    else:
        self.t1.setText(str(self.var[1]))
    self.ctg="BOW"
def radio3(self):
    if self.rb3.isChecked()==True:
        self.var[2]+=1

    if (self.var[2])<=1:
        self.t4.setText(str(self.var[2]))

```



```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
    self.t4.setText(str(self.var[2]))
else:
    self.msg3=msgb()
    self.msg3.setIcon(msgb.Critical)
    self.msg3.setWindowTitle("Warning")
    self.msg3.setText("You can not select more than 1 wicket keeper")
    self.msg3.exec_()
    self.var[2]=1
else:
    self.t4.setText(str(self.var[2]))
self.ctg="WK"
def radio4(self):
    if self.rb4.isChecked()==True:
        self.var[3]+=1
        self.t3.setText(str(self.var[3]))
    else:
        self.t3.setText(str(self.var[3]))
self.ctg="AR"

def openTeam(self):
    try:
        self.dlg3=dlg3()
        self.dlg3.setWindowTitle("Open Team")
        self.Lb13=QtWidgets.QLabel(self.dlg3)
        self.Lb13.setText("Team:")
        self.t9=QtWidgets.QLineEdit(self.dlg3)
        self.B3=QtWidgets.QPushButton("Open",self.dlg3)
        self.vBox3=QtWidgets.QVBoxLayout(self.dlg3)
        self.vBox3.addWidget(self.Lb13)

```

Ln: 328 Col: 29

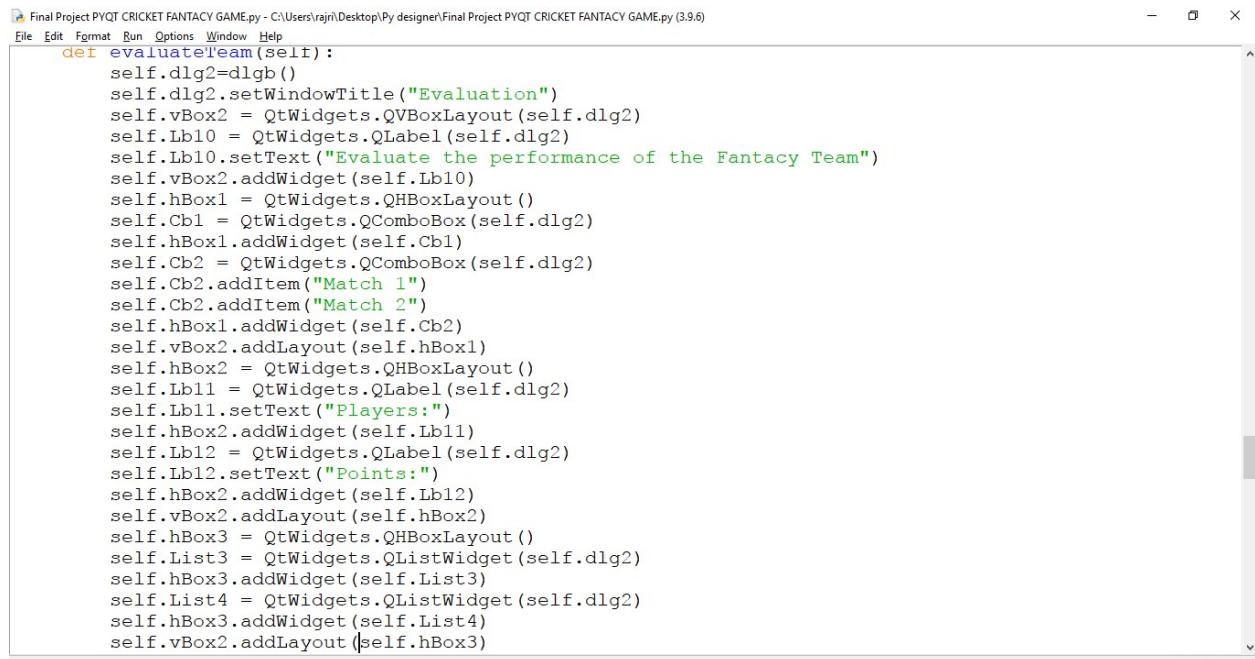
```

Type here to search  08:35 20-12-2021
Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Frgmat Run Options Window Help
    self.vBox3.addWidget(self.t9)
    self.vBox3.addWidget(self.B3)
    self.B3.clicked.connect(self.open)
    self.dlg3.exec_()

except:
    self.msg3=msgb()
    self.msg3.setIcon(msgb.Critical)
    self.msg3.setWindowTitle("Error")
    self.msg3.setText("Error While Fetching the Team Data")
    self.res3=self.msg3.exec_()
    if self.res3==msgb.Ok:
        conn.rollback()
        conn.close()
    else:
        return
def open(self):
    try:
        query="select * from teams where name = '"+self.t9.text()+"'"
        conn=sq.connect('Project.db')
        curs=conn.cursor()
        curs.execute(''CREATE TABLE if not exixts teams (
NAME CHAR (20),PLAYERS CHAR (30),VALUE INTEGER (20) );''')
        curs.execute(query)
        record=curs.fetchall()
        self.t7.setText(str(record[0][0]))
        dict3={}
        temp=0
        for i in range(15):

```

Ln: 357 Col: 29

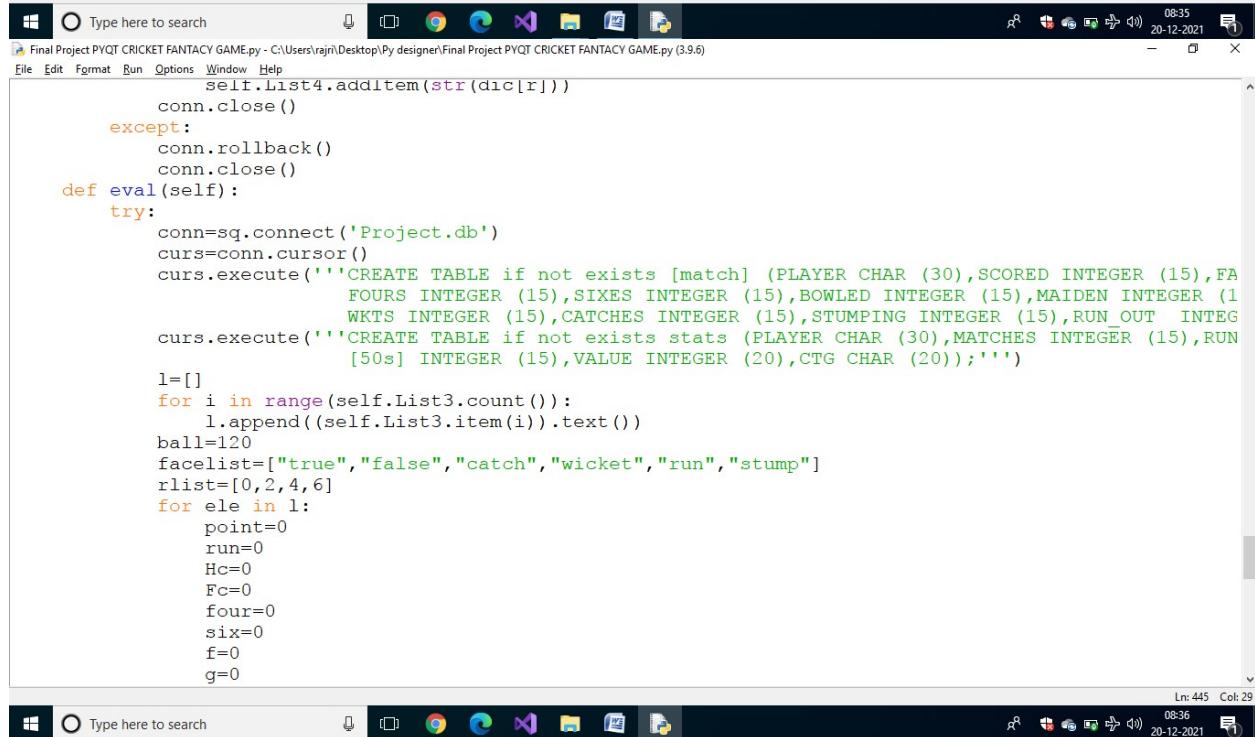


```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
def evaluateTeam(self):
    self.dlg2=dlg2()
    self.dlg2.setWindowTitle("Evaluation")
    self.vBox2 = QtWidgets.QVBoxLayout(self.dlg2)
    self.Lb10 = QtWidgets.QLabel(self.dlg2)
    self.Lb10.setText("Evaluate the performance of the Fantasy Team")
    self.vBox2.addWidget(self.Lb10)
    self.hBox1 = QtWidgets.QHBoxLayout()
    self.Cb1 = QtWidgets.QComboBox(self.dlg2)
    self.hBox1.addWidget(self.Cb1)
    self.Cb2 = QtWidgets.QComboBox(self.dlg2)
    self.Cb2.addItem("Match 1")
    self.Cb2.addItem("Match 2")
    self.hBox1.addWidget(self.Cb2)
    self.vBox2.addLayout(self.hBox1)
    self.hBox2 = QtWidgets.QHBoxLayout()
    self.Lb11 = QtWidgets.QLabel(self.dlg2)
    self.Lb11.setText("Players:")
    self.hBox2.addWidget(self.Lb11)
    self.Lb12 = QtWidgets.QLabel(self.dlg2)
    self.Lb12.setText("Points:")
    self.hBox2.addWidget(self.Lb12)
    self.vBox2.addLayout(self.hBox2)
    self.hBox3 = QtWidgets.QHBoxLayout()
    self.List3 = QtWidgets.QListWidget(self.dlg2)
    self.hBox3.addWidget(self.List3)
    self.List4 = QtWidgets.QListWidget(self.dlg2)
    self.hBox3.addWidget(self.List4)
    self.vBox2.addLayout(self.hBox3)

Ln: 428 Col: 29

```

```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
        self.List4.addItem(str(dic[r]))
    conn.close()
except:
    conn.rollback()
    conn.close()
def eval(self):
    try:
        conn=sq.connect('Project.db')
        curs=conn.cursor()
        curs.execute('''CREATE TABLE if not exists [match] (PLAYER CHAR (30), SCORED INTEGER (15), FA
FOURS INTEGER (15), SIXES INTEGER (15), BOWLED INTEGER (15), MAIDEN INTEGER (1
WKTS INTEGER (15), CATCHES INTEGER (15), STUMPING INTEGER (15), RUN_OUT INTEG
curs.execute('''CREATE TABLE if not exists stats (PLAYER CHAR (30), MATCHES INTEGER (15), RUN
[50s] INTEGER (15), VALUE INTEGER (20), CTG CHAR (20));'''')
l=[]
for i in range(self.List3.count()):
    l.append((self.List3.item(i)).text())
ball=120
facelist=["true","false","catch","wicket","run","stump"]
rlist=[0,2,4,6]
for ele in l:
    point=0
    run=0
    Hc=0
    Fc=0
    four=0
    six=0
    f=0
    g=0

Ln: 445 Col: 29

```

```

Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Format Run Options Window Help
s=0
    for i in range(ball):
        faced=random.choice(facelist)
        rpoint=random.choice(rlist)
        if faced == "true":
            if run<=(ball*6):

                run = run + rpoint

                if rpoint == 2:
                    point = point + 1

                elif rpoint == 4:
                    point = point + 1
                    four = four + 1
                elif rpoint == 6:
                    point = point + 2
                    six = six + 1
                elif run>=50 and run <100:
                    point = point + 5
                    Hc = Hc + 1

                elif run>=100:
                    point = point + 4*(run/100)
                    Fc = Fc +1

                elif run>0:
                    if (run/i) >=80 :
                        point = point + 2

```

```

Type here to search 08:37 20-12-2021
Final Project PYQT CRICKET FANTACY GAME.py - C:\Users\rajin\Desktop\Py designer\Final Project PYQT CRICKET FANTACY GAME.py (3.9.6)
File Edit Fgmat Run Options Window Help
    point = point + 10
    s = s + 1
    break
q="""+tele+","+str(run)+","+str(f)+" ,"+str(four)+" ,"+str(six)+" ,"+str(i)+" ,"+str(maidn)
#print(q)
curs.execute("""insert into [match](PLAYER ,SCORED ,FACED ,FOURS ,SIXES ,BOWLED ,MAIDEN
               WKTS ,CATCHES ,STUMPING ,RUN_OUT )
               values('''+q)
q1="""+tele+" ,"+str(1)+" ,"+str(run)+","+str(Fc)+" ,"+str(Hc)+" ,"+str(point)+" ,"+str("NA")+" ,"
#print(q1)

curs.execute("""insert into stats (PLAYER ,MATCHES ,RUNS ,[100s] ,[50s] ,VALUE ,CTG )
               values('''+q1)
conn.commit()
conn.close()
except:
    print("error")
    conn.rollback()
    conn.close()
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```
