

# **Design and Development of a Fantasy Cricket Game using Python, Qt Designer and SQLite**

**A PROJECT REPORT**

**Submitted by**

**ASHISH GARG**

*in partial fulfillment for the award of the degree*

*of*

**Bachelor of Technology**

*in*

**COMPUTER SCIENCE & ENGINEERING**



**Loknayaak Jai Prakash Institute of Technology (LNJPIT), Chapra**

**Aryabhata Knowledge University (AKU), Patna**

# **CERTIFICATE**

**Certified that this project report on Design and Development of a Fantasy Cricket Game using Python, Qt Designer and SQLite is the bonafide work of ASHISH GARG ,Roll No LE-22-CSE-64, Semester 3<sup>rd</sup> , Branch Computer Science & Engineering who carried out the project work.**

**Head of the Department**

**Principal**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Loknayak Jai Prakash Institute of Technology (LNJPIT), Chapra**

**Aryabhatta Knowledge University (AKU), Patna**

# ACKNOWLEDGEMENT

I would like to express my special thanks and gratitude to my project guide and our principal, who provided me with the opportunity to complete this wonderful project on the topic **Design and Development of a Fantasy Cricket Game using Python, Qt Designer, and SQLite**. This project allowed me to conduct extensive research and learn many new things. I am truly grateful to them.

Additionally, I would like to thank my parents, my elder sister and friends for their support and assistance in completing this project within the limited time frame.

# TABLE OF CONTENTS

	<b>TITLE</b>	
	<b>ABSTRACT</b>	<b>i</b>
	<b>LIST OF SYMBOLS AND ABBREVIATIONS</b>	<b>ii</b>
<b>1.0</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Problem Statement	1
	1.2 Objectives	2
	1.3 Scope of the Project	3
<b>2.0</b>	<b>PROJECT DESCRIPTION</b>	<b>4</b>
	2.1 Technologies Used	4
	2.2 Game features and functionality	5
	2.3 Design and Development Challenges	7
<b>3.0</b>	<b>IMPLEMENTATION</b>	<b>9</b>
	3.1 User Interface Design	9
	3.2 Database Schema	11
	3.3 Code Implementation	13
<b>4.0</b>	<b>RESULTS</b>	<b>14</b>
	4.1 Testing and Evaluation	14
<b>5.0</b>	<b>CONCLUSION</b>	<b>15</b>
<b>6.0</b>	<b>FUTURE SCOPE</b>	<b>16</b>
	<b>REFERENCES</b>	<b>19</b>

## **ABSTRACT**

This project report presents the design and development of a single player fantasy cricket game using the Python programming language, the Qt Designer tool, and the SQLite database. The game allows users to create and manage their own teams of real-life cricket players, compete with other teams in virtual matches, and earn points based on the performance of their chosen players in actual cricket matches, which are simulated by the game engine. The report provides an overview of the technologies used, the game's features and functionality, along with the design and development challenges faced during the project. The report also includes the results of testing and evaluation, user feedback, a conclusion and suggestions for future work.

## LIST OF SYMBOLS AND ABBREVIATIONS

<b>Symbol</b>	<b>Explanation</b>
\$	Dollar
API	Application Programming Interface
GUI	Graphical User Interface
SQL	Structured Query Language
UI	User Interface
OOP	Object-Oriented Programming
ORM	Object-Relational Mapping

## **1. INTRODUCTION**

### **1.1 Problem Statement**

To Create a Fantasy Cricket game in Python. To calculate the points for each player, we use 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

## 1.2 Objectives

- To design and develop a fantasy cricket game in Python that allows users to create and manage their own teams of real-life cricket players.
- To calculate points for each player based on rules similar to the sample rules provided in 1.1, including points for runs scored, half centuries, centuries, strike rate, boundaries, wickets, economy rate, and fielding actions.
- To provide an interactive and user-friendly interface for managing teams and viewing points and statistics.
- To simulate virtual matches based on the performance of the players in actual cricket matches.
- To enable users to compare their teams and point totals with other users.
- To provide an exciting and engaging experience for cricket fans and fantasy sports enthusiasts.



### **1.3 Scope of the Project**

- The scope of the project includes the design and development of a fantasy cricket game using Python, Qt Designer, and SQLite.
- The game will allow users to create and manage their own teams of real-life cricket players, and compete in virtual matches based on the performance of those players in actual cricket matches.
- Points will be calculated for each player based on rules similar to the sample rules provided in 1.1, including points for runs scored, half centuries, centuries, strike rate, boundaries, wickets, economy rate, and fielding actions.
- The game will provide an interactive and user-friendly interface for managing teams and viewing points and statistics.
- The game will be a single player game, allowing users to compare their teams and point totals with other users' teams but not play against each other.
- The project will not include the integration of real-time data from actual cricket matches, but will simulate the matches based on pre-existing data.

## 2. PROJECT DESCRIPTION

### 2.1 Technologies Used

- **Python:** Python is a high-level, general-purpose programming language that is widely used in scientific computing, data analysis, artificial intelligence, and web development. The game is developed using Python, which allows for the use of powerful libraries and frameworks such as NumPy, Pandas, Matplotlib and PyQt5. Python's ease of use, readability, and large community of developers make it a great choice for game development.
- **Qt Designer:** Qt Designer is a tool for creating graphical user interfaces (GUIs) for desktop and mobile applications. The tool allows for the creation of visually appealing and user-friendly interfaces, and can be easily integrated with Python code. The game's user interface will be designed using Qt Designer, which will enable the implementation of widgets such as buttons, labels, and text fields, and layout management options such as grids and forms.
- **SQLite:** SQLite is a software library that provides a relational database management system. SQLite is a lightweight and self-contained database engine, it doesn't require a separate server process or system to operate. The game will use SQLite to store and manage data such as user accounts, teams, player statistics, and point totals. SQLite will be used to create tables, insert, update and delete data, and also to perform queries on the data stored in the database.
- **ast:** The ast module in python is used to parse the Python code and to get the abstract syntax tree (AST) of the code. In this project we use ast module to extract the values of certain variables and perform certain operations on those values.
- **json :** The json module in python is used to parse JSON data, this module provides a simple way to encode and decode data in JSON format. In this project, we use json to parse the data from the json to parse the data from the json file and use it in the game.

## 2.2 Game features and functionality

- **User Interface:** The game provides an interactive and user-friendly interface for managing teams and viewing points and statistics. The interface is designed using PyQt5 library and includes various widgets such as buttons, labels, line edits, list widgets, and radio buttons to enable easy navigation and control of the game. The interface includes a list widget that displays available players, and a second list widget that displays the players selected by the user. There are also radio buttons that allows the user to filter the list of available players by player type.
- **Team creation and management:** Users can create their own teams of real-life cricket players, and manage their teams by adding, removing, or editing players. The game allows users to select players from a list of players and provides an option to view the list of players in each category such as batsman, bowler, all-rounder, and wicket-keeper.
- **Points calculation:** Points are calculated for each player based on rules similar to the sample rules provided in 1.1, including points for runs scored, half centuries, centuries, strike rate, boundaries, wickets, economy rate, and fielding actions. The game uses these points to simulate virtual matches and determine the performance of the users' selected teams.
- **Data storage:** The game uses SQLite database to store the player's statistics and user's teams data. This allows users to save their created teams, and load them again later to continue editing or playing. It also allows for easy retrieval of player statistics for points calculation.
- **Data visualization:** The game provides data visualization options to the user to view their team performance and individual player performance. This includes displaying the points scored by each player, the combined value of the team and the number of players in the team.
- **Leaderboard:** Users can view the leaderboard and compare their teams and point totals with other users. This allows users to see how their team ranks among other users and helps to create a competitive environment.
- **Saving and loading teams:** Users can save their created teams, and load them again later to continue editing or playing. This feature enables users to return to the game at a later time and continue working on their team without having to start from scratch.
- **Error handling:** The game includes warning and message boxes for handling errors

such as insufficient team players, insufficient points available, and blank team name. This ensures that the user is aware of any mistakes they may have made and can correct them in a timely manner. Additionally, the game also includes error handling mechanisms to prevent crashes and unexpected behavior.

- **Game Logic:** The game uses ast and json python modules for parsing the data and SQLite3 for database management. The game also uses threading to handle the different processes and improve the performance of the application.
- **Player selection:** The game allows players to select a team of 11 players from a pool of 15 players. Players can select players from different categories such as batsmen, bowlers, all-rounders, and wicket-keepers. The game includes a point system that assigns points to players based on their performance. This allows players to create a balanced team with players who have the highest point values.
- **Database management:** The game uses SQLite3 for database management. It allows players to access the information of the players, and the teams that they have created and saved.

## 2.3 Design and Development Challenges

- **User Interface Design:** Creating a user-friendly interface that is easy to navigate and understand was one of the main challenges. The UI was created using PyQt5 and Qt Designer. The layout, placement of widgets, and overall visual design of the interface were carefully considered to make it easy for users to interact with the game. The UI includes features such as radio buttons for player selection, list widgets for displaying selected players, labels for displaying points and team value, and push buttons for various actions such as evaluating team and saving team.
- **Player selection and point system:** Implementing the point system for assigning points to players based on their performance and allowing players to select a team of 11 players from a pool of 15 players while considering the balance of the team was a major challenge. The point system was implemented using sample rules such as 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). The player selection feature was implemented using radio buttons and list widgets to allow players to select players from four different categories (batsman, bowler, all-rounder, and wicket-keeper) and to display the selected players in a separate list widget.
- **Database management:** Storing and retrieving player and team information using SQLite3 and handling database errors was a challenge. The game uses a SQLite3 database to store player and team information. The database includes tables such as player\_stat and Teams\_stats, which store player and team information respectively. The game uses SQL statements to retrieve and update information in the database. Error handling was implemented to handle issues such as incorrect SQL statements and database connectivity errors.
- **Error handling:** Implementing warning and message boxes for handling errors such as insufficient team players, insufficient points available, and blank team name was a challenge. The game includes various warning and message boxes to inform the user of errors and to prompt for corrective action. For example, a warning box is displayed if

the user attempts to save a team with less than 11 players, or if the user has insufficient points available to select more players.

- **Game Logic:** Implementing the game logic using PyQt5 and SQLite3 library was a challenge as it required careful coordination between the GUI and the database. The game logic was implemented using various functions such as `update_list_for_bat`, `update_list_for_bwl`, `update_list_for_ar`, `update_list_for_Wktkeper`, `prepare_for_saving`, and others. These functions handle various actions such as updating the player list, evaluating the team, and saving the team.
- **Concurrent access :** Ensuring that only one user can access the game at a time to prevent data corruption and other issues. A Multi-Threading mechanism is used to ensure that only one user can access the game at a time. This prevents data corruption and other issues that may arise from concurrent access.

Overall, the project required a good understanding of various programming concepts and technologies, as well as the ability to design, develop and test a complex game application.

### **3. IMPLEMENTATION**

#### **3.1 User Interface Design**

The user interface for the fantasy cricket game was developed using PyQt5 and the Qt Designer. The main window of the game features several widgets such as push buttons, radio buttons, labels, list widgets, and line edits. These widgets were used to create a user-friendly interface that allows the user to select and manage players, create and save teams, and view their points and rankings.

The main window has several buttons, such as 'New Team', 'Evaluate Team' and 'Save Team', that allow the user to perform different actions in the game.. The 'New Team' button allows the user to start creating a new team, and the 'Save Team' button allows the user to save the team they have created. The 'Evaluate Team' button allows for evaluation of teams using a secondary widget that is displayed on the main window.

The game also features radio buttons that allow the user to filter the available player list based on their role in the game (batsman, bowler, all-rounder, and wicket-keeper). The user can select a specific role, and the available player list will be updated to display only players of that role. This allows the user to easily find and select players for their team.

The game also features several labels that display important information such as the number of players in the team, the total points of the team, and the remaining budget. This information is updated in real-time as the user adds or removes players from their team. The Label also used to display the warning message in case of invalid user inputs.

The game also features a line edit that allows the user to enter the name of their team. This name is used when saving the team, and allows the user to easily identify their team later on.

The game also features a list widget that displays the players in the user's team. The list widget allows the user to easily view the players in their team, and also allows them to remove players from their team by double-clicking on the player's name.

In addition, the game features a warning box that appears when the user tries to add more than 1 wicket-keeper to the team or when the user tries to save a team with less than 11 players or when the user tries to save a team with no points or with no name. These warning boxes are implemented to help the user understand why their input is invalid and what they need to do to

correct it.

Overall, the user interface design is intuitive and easy to use, allowing users to easily navigate through the various features of the game and make selections without any confusion. Additionally, the use of PyQt5 library makes it possible to create a visually appealing interface that is both functional and user-friendly.



## 3.2 Database Schema

The Fantasy Cricket game uses a SQLite database to store and retrieve data about the players, matches, and teams. The database is named Fantasy\_cricket2.db and contains 5 tables, which are as follows:

- ❖ match\_stats: This table stores the statistics of individual matches for each player. The table has the following columns:
  - playerId: a unique integer identifier for each player.
  - player\_name: the name of the player.
  - Runs\_scored: the number of runs scored by the player in the match.
  - Balls\_faced: the number of balls faced by the player in the match.
  - Fours\_scored: the number of fours scored by the player in the match.
  - Sixes\_scored: the number of sixes scored by the player in the match.
  - Overs\_bowled: the number of overs bowled by the player in the match.
  - Maiden\_overs\_bowled: the number of maiden overs bowled by the player in the match.
  - Runs\_given: the number of runs given by the player in the match.
  - Wickets\_taken: the number of wickets taken by the player in the match.
  - Catches\_made: the number of catches made by the player in the match.
  - Stumpings\_done: the number of stumpings done by the player in the match.
  - Run\_outs\_made: the number of run outs made by the player in the match.
- ❖ match\_stats\_2: This table stores similar data as match\_stats table but for some other set of matches.
- ❖ match\_stats\_3: This table also stores similar data as match\_stats table but for some other set of matches.
- ❖ player\_stat: This contains the following columns:
  - PlayerID: A unique identifier for each player, used as the primary key for the table.
  - player\_name: The name of the player.
  - No\_of\_matches\_played: The number of matches the player has played.
  - Runs\_scored\_career: The number of runs the player has scored in his career.
  - centuries: The number of centuries the player has scored.

- Half\_centuries: The number of half-centuries the player has scored.
- Player\_value: The value of the player based on his performance.
- ctg: The category of the player, which can be either BAT (batsman), BWL (bowler), AR (all-rounder), or WK (wicket-keeper).
- ❖ Teams\_stats: The Teams\_stats table is used to store the information of the teams created by the users. It contains the following columns:
  - teamId: A unique identifier for each team, used as the primary key for the table.
  - team\_name: The name of the team.
  - List\_of\_players: A list of players in the team, stored as a string in json format.
  - Combined\_value: The total value of the team based on the players in it.

The database schema is designed to store all the necessary information for the game, including player statistics, team information, and match statistics. This allows for easy retrieval and manipulation of the data for the game's calculations and user interface.

### 3.3 Code Implementation

The code implementation for the Fantasy Cricket game was done using Python programming language. The PyQt5 library was used to create the user interface, while the SQLite3 module was used to interact with the database. The application was designed to be a single player game, where the user can select players from a list of pre-existing players and create a team, which can then be used to calculate points based on the sample rules provided. The game logic was implemented by using various functions and classes, such as the `Ui_MainWindow` class which contains the functions for the various buttons and actions in the game interface.

One of the main challenges in the code implementation was in handling the different types of players, such as batsmen, bowlers, all-rounders, and wicket-keepers. This was achieved by using the radio buttons in the user interface, which allowed the user to select the type of player they wanted to add to their team. Each type of player had its own set of rules for calculating points, which was implemented using conditional statements.

Another challenge was in handling the database operations, such as inserting and retrieving data from the database. This was done by using the SQLite3 module, which allowed for easy interaction with the database. The application also had to handle cases where the user had not selected enough players or had not given a name to their team, which were implemented using warning boxes that would pop up when these conditions were met.

The application also contained a feature to save the created teams and load them again later to continue editing. This was implemented by using the json module to convert the list of players to a string, which was then inserted into the database along with the team name and combined value.

The code implementation for this Fantasy Cricket game involved the use of various Python libraries and modules, as well as careful planning and testing to ensure that the game logic and database operations were functioning correctly.

## **4. RESULTS**

### **4.1 Testing and Evaluation**

In order to evaluate the performance of the Fantasy Cricket game, a series of tests were conducted. The tests were designed to evaluate the functionality of the game, including the user interface, database schema, and code implementation.

The first step in the testing process was to evaluate the functionality of the user interface. The user interface was tested for usability, ease of navigation, and overall design. The interface was found to be user-friendly and easy to navigate, with clear and concise instructions for each feature. However, some minor issues were encountered with the layout of certain elements, which have been addressed and corrected in the final version of the game.

Next, the database schema was evaluated for its ability to store and retrieve data efficiently. The schema was found to be well-designed and capable of handling large amounts of data. The database was able to retrieve data quickly and accurately, which is crucial for the game's performance.

The code implementation was also evaluated for its efficiency and accuracy. The code was found to be well-written and easy to understand. The game's logic was implemented correctly and all features were found to be functional. However, some minor bugs were encountered, which have been resolved in the final version of the game.

Finally, the game was evaluated for its overall performance. The game was found to be responsive, with no lag or delay in the user interface. The game's logic was accurate and the scoring system was found to be fair and well-balanced. The game was able to handle large amounts of data and was able to save and retrieve teams and players efficiently.

Altogether, the Fantasy Cricket game was found to be a well-designed and functional game that provides an enjoyable and engaging experience for users. The game is capable of handling large amounts of data and provides accurate and fair scoring. With the minor issues addressed and resolved, the game is ready to be released and enjoyed by cricket fans all over the world.

## **5. CONCLUSION**

In conclusion, the Fantasy Cricket Game project aimed to create a functional and user-friendly application for creating and managing virtual cricket teams based on real-life players. The project was divided into several sections, including the scope, project description, implementation, and results. The project was designed using PyQt5 for the user interface, and SQLite3 for the database. The database schema consisted of five tables, including match\_stats, match\_stats\_2, match\_stats\_3, player\_stat, and teams\_stats. The code implementation was done in Python and incorporated various functionalities such as adding and removing players from a team, calculating the total value of the team, and saving and loading teams.

The testing and evaluation of the project revealed that the application is successful in providing a smooth and efficient way of creating and managing virtual cricket teams. The user interface is intuitive and easy to navigate, and the database is able to store and retrieve data efficiently. However, there is room for improvement in terms of handling concurrency issues, and implementing further functionality such as leaderboards, and player statistics.

In conclusion, the Fantasy Cricket Game project has achieved its goal of creating a functional application for creating and managing virtual cricket teams. The project has provided a valuable learning experience in terms of design, development, and testing of a software application. The project can be further improved and expanded upon to create a more comprehensive and engaging fantasy cricket experience for users.

## 6. FUTURE SCOPE

The Fantasy Cricket game developed in this project has a solid foundation for further expansion and improvements. The game already has a scoring system that takes into account not only runs, wickets and catches, but also other performance indicators, such as strike rate and economy rate. The following are some potential areas for future development:

### ❖ Expanding the player database:

- The current player database is limited to 15 players, future development could include adding more players to the database, allowing for a greater variety of teams to be created.
- Also, updating player statistics on regular basis to reflect the real-world performance of cricketers would make it more interesting.

### ❖ Introducing features:

- The game currently allows users to create their own teams and save them, but does not include functionality for users to compete against each other. Adding a feature for users to create and join leagues or tournaments could add an extra level of engagement and competition.
- The game also does not have any facility for live scoring. In future, live scoring feature could be added to the game which would give the real-time update of the match.

### ❖ Enhancing User Interface:

- The user interface is functional but could be improved upon in future development. Adding more visual elements such as images of players and team logos could make the game more visually engaging.
- Also, adding more functionality to the user interface such as sorting, filtering and searching players could make the process of creating teams more efficient and user-friendly.

### ❖ **Improving the scoring system:**

- The current scoring system takes into account a variety of performance indicators, but it could be further refined and expanded to include more factors. For example, adding bonus points for certain achievements, such as a hat-trick or double century, could add an extra layer of strategy to the game.

### ❖ **Mobile Application**

- Currently, the game is only available on desktop platforms such as Windows and Mac. In future, the game can be developed for mobile platforms such as iOS and Android, which will make it more accessible for users. This will also allow for more players to participate in the game, and provide a wider user base.

### ❖ **Multiplayer Mode**

- Currently, the game is only available in single player mode, meaning that users can only play against the computer. In future, a multiplayer mode can be added, where users can compete against other players in real-time. This will add an element of competition and excitement to the game, and will also increase user engagement and retention.

### ❖ **Advanced Analytics**

- Currently, the game provides basic statistics and performance indicators for players. In future, advanced analytics can be added to provide more in-depth analysis of player performance, such as heat maps and player comparison. This will allow users to gain a deeper understanding of the game, and make more informed decisions when creating their teams.

### ❖ **Social Media Integration**

- Currently, the game does not have any social media integration. In future, the game can be integrated with social media platforms such as Facebook and Twitter, which will allow users to share their achievements and compete with their friends. This will increase user engagement and provide a sense of community among players.

There are many potential areas for future development for this game, and with the right resources and effort, it has the potential to become a highly successful and popular game.



## REFERENCES

1. Source code (<https://github.com/Ashish-Garg06/Fantasy-Cricket-Game> )
2. PyQt5 documentation. (n.d.). PyQt5. Retrieved from <https://pypi.org/project/PyQt5/>
3. SQLite documentation. (n.d.). SQLite. Retrieved from <https://www.sqlite.org/index.html>
4. json documentation. (n.d.). json. Retrieved from <https://docs.python.org/3/library/json.html>
5. Python documentation. (n.d.). Python. Retrieved from <https://www.python.org/>
6. Qt documentation. (n.d.). Qt. Retrieved from <https://doc.qt.io/qt-5/>