A Project Report On

CRICKET CLUB MANAGENT SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Computer Application
BCA

Academic Year 2025 – 26

Ankan Khanda **Snehendu Bhunia**
92300527161 **92300527163**

Internal Guide
Tirth Bhadeshiya

Marwadi University
Marwadi Chandarana Group
NAAC A+

Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.

Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled

<CRICKET CLUB MANAGENT SYSTEM>

submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Computer Application

BCA

of the

Marwadi University

is a result of the bonafide work carried out by

Ankan Khanda(92300527161)

Snehendu Bhunia (92300527163)

during the academic year 2025-26

Faculty Guide	HOD	Dean

DECLARATION

We hereby declare that this project work entitled **Cricket club management system** is a record done by us.

we also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place : Marwadi University

Date :

Ankan Khanda(92300527161) Signature : _____

Snehendu Bhunia (92300527163) Signature :_____

CONTENTS

Chapters	Particulars	Page No.
1	**SYNOPSIS**	
2	**PREAMBLE** General Introduction Module description	
3	**TECHNICAL DESCRIPTION** Hardware Requirement Software Requirement	
4	**SYSTEM DESIGN AND DEVELOPMENT** **(Only applicable diagrams)** • Flowchart • Data Flow Diagram, Use Case Diagram, • Sequential Diagram, Class Diagram, • Activity Diagram • Database Design (If applicable) Screen Design & Coding	
5	**CONCLUSION**	
6	**LEARNING DURING SIP**	
7	**BIBLIOGRAPHY** Online References Offline References	

1.SYNOPSIS

The Cricket Club Management System is a desktop-based application developed in Python using the Tkinter library for GUI and SQLite for database management. This application allows club administrators to maintain and manage player records efficiently. Features include adding, viewing, deleting players, and resetting input fields. The system is user-friendly, supports essential CRUD (Create, Read, Update, Delete) operations, and is designed for scalability and simplicity.

2. PREAMBLE

General Introduction

Managing players in a cricket club manually is time-consuming and error-prone. This project automates player record handling by providing an intuitive graphical interface to manage player data. It helps maintain contact information, player roles, joining dates, and more, ensuring that records are accurate and up to date.

Module Description

The system allows users to:

- Add new player records with details like name, contact, email, role, etc.
- View existing records in a tabular format.
- Delete individual players or clear the entire database.
- Reset form fields for quick data entry.

The application uses SQLite as the backend database, making it lightweight and easy to deploy on any system.

3.TECHNICAL DESCRIPTION

Hardware Requirements

- Processor: Intel i3 or equivalent
- RAM: 2 GB or more
- Storage: 100 MB free space
- Operating System: Windows 10 or higher / Linux / macOS

Software Requirements

- Python 3.x
- Tkinter (built into Python)
- SQLite3 (built into Python)
- Code Editor: VS Code, PyCharm, or any text editor

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 Flowchart

4.2 Data flow diagram

4.3Use Case Diagram

4.4 Sequential Diagram

4.5 Class digram

4.6 Activity digram

- **Coding**

```
import tkinter as tk
from tkinter import ttk, messagebox
import sqlite3

conn = sqlite3.connect("cricket_club.db")
cursor = conn.cursor()
cursor.execute("""
CREATE TABLE IF NOT EXISTS players (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT,
    contact TEXT,
    email TEXT,
    age INTEGER,
    role TEXT,
    doj TEXT,
    team TEXT
)
""")
conn.commit()

def add_player():
    name = entry_name.get()
    contact = entry_contact.get()
    email = entry_email.get()
    age = entry_age.get()
    role = entry_role.get()
    doj = entry_doj.get()
    team = entry_team.get()

    if name == "" or contact == "" or email == "":
        messagebox.showerror("Error", "Please fill all required fields")
        return

    cursor.execute("INSERT INTO players (name, contact, email, age, role, doj, team)
VALUES (?, ?, ?, ?, ?, ?, ?)",
                   (name, contact, email, age, role, doj, team))
    conn.commit()
    messagebox.showinfo("Success", "Player added successfully")
    view_players()
```

```
    reset_fields()

def delete_player():
    selected = tree.selection()
    if not selected:
        messagebox.showerror("Error", "Please select a player to delete")
        return
    player_id = tree.item(selected[0])['values'][0]
    cursor.execute("DELETE FROM players WHERE id=?", (player_id,))
    conn.commit()
    view_players()

def view_players():
    for row in tree.get_children():
        tree.delete(row)
    cursor.execute("SELECT * FROM players")
    rows = cursor.fetchall()
    for row in rows:
        tree.insert("", tk.END, values=row)

def reset_fields():
    entry_name.delete(0, tk.END)
    entry_contact.delete(0, tk.END)
    entry_email.delete(0, tk.END)
    entry_age.delete(0, tk.END)
    entry_role.delete(0, tk.END)
    entry_doj.delete(0, tk.END)
    entry_team.delete(0, tk.END)

def delete_database():
    cursor.execute("DELETE FROM players")
    conn.commit()
    view_players()
    messagebox.showinfo("Deleted", "All records deleted")

root = tk.Tk()
root.title("Cricket Club Management System")

root.state('zoomed')

root.config(bg="lightgreen")
```

```
title_label = tk.Label(root, text="🏏 Cricket Club Management System 🏏",
                       font=("Arial", 22, "bold"), bg="darkgreen", fg="white", pady=15)
title_label.pack(fill=tk.X)

frame_left = tk.Frame(root, bg="lightgreen")
frame_left.pack(side=tk.LEFT, padx=40, pady=20, fill=tk.Y)

tk.Label(frame_left, text="Player Name", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_name = tk.Entry(frame_left, font=("Arial", 12))
entry_name.pack(fill=tk.X)

tk.Label(frame_left, text="Contact Number", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_contact = tk.Entry(frame_left, font=("Arial", 12))
entry_contact.pack(fill=tk.X)

tk.Label(frame_left, text="Email Address", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_email = tk.Entry(frame_left, font=("Arial", 12))
entry_email.pack(fill=tk.X)

tk.Label(frame_left, text="Age", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_age = tk.Entry(frame_left, font=("Arial", 12))
entry_age.pack(fill=tk.X)

tk.Label(frame_left, text="Role (Batsman/Bowler/All-rounder/WK)", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_role = tk.Entry(frame_left, font=("Arial", 12))
entry_role.pack(fill=tk.X)

tk.Label(frame_left, text="Date of Joining", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_doj = tk.Entry(frame_left, font=("Arial", 12))
entry_doj.pack(fill=tk.X)

tk.Label(frame_left, text="Team", bg="lightgreen", font=("Arial", 12, "bold")).pack(anchor="w")
entry_team = tk.Entry(frame_left, font=("Arial", 12))
```

```
entry_team.pack(fill=tk.X)

tk.Button(frame_left, text="Add Player", command=add_player, bg="darkgreen",
fg="white", font=("Arial", 12, "bold")).pack(pady=5, fill=tk.X)
tk.Button(frame_left, text="Delete Player", command=delete_player, bg="red",
fg="white", font=("Arial", 12, "bold")).pack(pady=5, fill=tk.X)
tk.Button(frame_left, text="View Players", command=view_players, bg="blue",
fg="white", font=("Arial", 12, "bold")).pack(pady=5, fill=tk.X)
tk.Button(frame_left, text="Reset Fields", command=reset_fields, bg="orange",
fg="black", font=("Arial", 12, "bold")).pack(pady=5, fill=tk.X)
tk.Button(frame_left, text="Delete Database", command=delete_database, bg="black",
fg="white", font=("Arial", 12, "bold")).pack(pady=5, fill=tk.X)

frame_right = tk.Frame(root, bg="white")
frame_right.pack(side=tk.RIGHT, padx=20, pady=20, expand=True, fill=tk.BOTH)

columns = ("ID", "Name", "Contact", "Email", "Age", "Role", "DOJ", "Team")
tree = ttk.Treeview(frame_right, columns=columns, show="headings")

for col in columns:
    tree.heading(col, text=col)
    tree.column(col, width=150, anchor="center")

tree.pack(expand=True, fill=tk.BOTH)

view_players()

root.mainloop()
```

Screen Design

5. CONCLUSION

The Cricket Club Management System project has successfully demonstrated how to manage player information efficiently using Python, Tkinter, and SQLite. It provides an easy-to-use interface for performing essential operations like adding, viewing, and deleting records. The application simplifies data management, reduces manual errors, and allows administrators to focus on team development rather than paperwork. The system is scalable and can be enhanced further with additional features like performance tracking, reports, or user authentication. This project also helped strengthen skills in GUI development, database integration, and event-driven programming.

6. LEARNING DURING SIP

During the development of this project as part of the Summer Internship Program (SIP), the following key learnings were achieved:

- Python Programming: Gained hands-on experience with Python syntax, libraries, and functions.
- Tkinter GUI Design: Learned how to build interactive and user-friendly applications with proper layout management.
- Database Handling: Understood how to connect, create, and manipulate a SQLite database.
- Data Validation: Learned techniques to ensure that user inputs are correctly handled and validated before submission.
- Debugging & Testing: Improved troubleshooting skills through testing various scenarios and resolving bugs.
- Documentation & Reporting: Learned how to structure and present technical reports for software projects.
- Time Management: Learned how to plan tasks, set milestones, and complete the project within deadlines.

7. BIBLIOGRAPHY

➤ Online References

1. Python Official Documentation – https://docs.python.org/3/
2. Tkinter Documentation – https://docs.python.org/3/library/tkinter.html
3. SQLite Tutorial – https://www.sqlitetutorial.net/
4. GeeksforGeeks – https://www.geeksforgeeks.org/python-gui-tkinter/
5. W3Schools Python Tutorial – https://www.w3schools.com/python/

➤ Offline References

1. "Core Python Programming" by Wesley J. Chun
2. "Python GUI Programming with Tkinter" by Alan D. Moore
3. Internship mentor guidance and in-house training materials
4. Notes and classroom discussions during the SIP program