K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi.
ISO 9001:2015, 14001:2015 certified institution. Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

A Project Report

on

SCHOOL SPORTS TEAM MANAGEMENT

Submitted in partial fulfillment of requirements for the award of the course

of

MGB1201 – PYTHON PROGRAMMING

Under the guidance of

Mrs.S.RAJESWARI M.E.

Assistant Professor / CSE

Submitted By

SWASTHI K (8115U23ME053)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K.RAMAKRISHNAN COLLEGE OF ENGINEERING
(Autonomous)

TRICHY-621 112

DECEMBER 2024

K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

TRICHY-621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“SCHOOL SPORTS TEAM MANAGEMENT”** is the bonafide work of **SWASTHI K(8115U23ME053)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature

Mrs .S.RAJESWARI M.E.

SUPERVISOR,

Department of Computer Science and Engineering,

K. Ramakrishnan College of Engineering
Samayapuram, Trichy-621 112.

.

Signature

Dr. T.M. NITHYA M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Computer Science and Engineering,

K. Ramakrishnan College of Engineering
Samayapuram, Trichy-621 112.

Submitted for the End Semester Examination held on…………….

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“SCHOOL SPORTS TEAM MANAGEMENT”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **MGB1201 – PYTHON PROGRAMMING**

.

Signature

SWASTHI K

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Engineering (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S.RAJESWARI M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions

MISSION OF THE INSTITUTION

M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

VISION OF THE DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF THE DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successful career.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write
11. effective reports and design documentation, make effective presentations, and give and receive clear instructions.
12. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
13. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

ABSTRACT

The School Sports Team Management Program is a user-friendly Python-based application designed to assist in organizing and managing sports events efficiently. The program allows users to perform key operations such as adding new events, viewing a list of all events, searching for specific events, and deleting outdated or unnecessary events. Each event is stored in a dictionary, with details including the event name, number of participants, judge, and location, ensuring efficient and structured data management. This program emphasizes simplicity and accessibility, making it suitable for school staff and even programming beginners. Its modular design ensures that each function operates independently, allowing for easy customization or expansion, such as adding new attributes like event dates or times. By automating manual tasks like record-keeping and retrieval, the program enhances accuracy, reduces errors, and saves time.

.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
The School Sports Team Management Program is a user-friendly Python-based application designed to assist in organizing and managing sports events efficiently. The program allows users to perform key operations such as adding new events, viewing a list of all events, searching for specific events, and deleting outdated or unnecessary events. Each event is stored in a dictionary, with details including the event name, number of participants, judge, and location, ensuring efficient and structured data management. This program emphasizes simplicity and accessibility, making it suitable for school staff and even programming beginners. Its modular design ensures that each function operates independently, allowing for easy customization or expansion, such as adding new attributes like event dates or times. By automating manual tasks like record-keeping and retrieval, the program enhances accuracy, reduces errors, and saves time.	PO1,PO2, PO3,PO12	PSO1

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of the School Sports Team Management Program is to provide an efficient and user-friendly solution for organizing and managing sports events in schools. It aims to streamline operations by enabling users to add, view, search, and delete event details, all stored in a centralized and structured format using dictionaries. By automating manual record-keeping tasks, the program enhances accuracy, ensures data integrity, and reduces redundancy. Its scalable design allows for future expansions, such as adding new attributes or functionalities to meet evolving needs. Additionally, the program is designed to be beginner-friendly, showcasing fundamental Python concepts like dictionaries, loops, and conditionals, making it accessible to users with basic programming knowledge. Ultimately, it simplifies event management, saves time, and serves as a reliable tool for school staff or administrators.

Overview

The School Sports Team Management Program is a Python-based application designed to simplify the management of school sports events. The program allows users to perform essential tasks such as adding, viewing, searching, and deleting event details. Each event is stored in a structured format using dictionaries, where details like the event name, number of participants, judge, and location are recorded. This ensures efficient data organization and quick retrieval. It is suitable for school staff or administrators to handle event records efficiently, as well as for programming beginners to understand key Python concepts such as data storage, loops, and conditionals. By automating manual tasks like record-keeping and retrieval, the program reduces errors, saves time, and ensures data accuracy

1.2 Python Programming Concepts

1. Dictionaries: Used for storing event details with the event name as the key and event information (e.g., participants, judge, location) as the value.

2. Functions: Modular design with separate functions for adding, viewing, searching, and deleting events (`add_event()`, `view_events()`, etc.).

3. For Loops: Iterates over the dictionary to display all events and their details.

4. While Loop: Keeps the program running until the user chooses to exit.

5. If-Else Statements: Handles menu choices, checks if events exist, and provides appropriate feedback.

6. Input/Output Operations:

Input: Collects data from the user (event details, menu choices).

Output: Displays messages, lists of events, and event details on the console.

7. Error Handling: Checks for conditions like empty dictionaries or events not found and handles them gracefully.

8. Nested Data Structures: Uses dictionaries within dictionaries to store and organize event-related data effectively.

9. String Manipulation: Formats output to display event details clearly (e.g., `f-strings` for formatted text).

10. Menu-Driven Interface: Provides an interactive user interface to navigate through the program's functionalities.

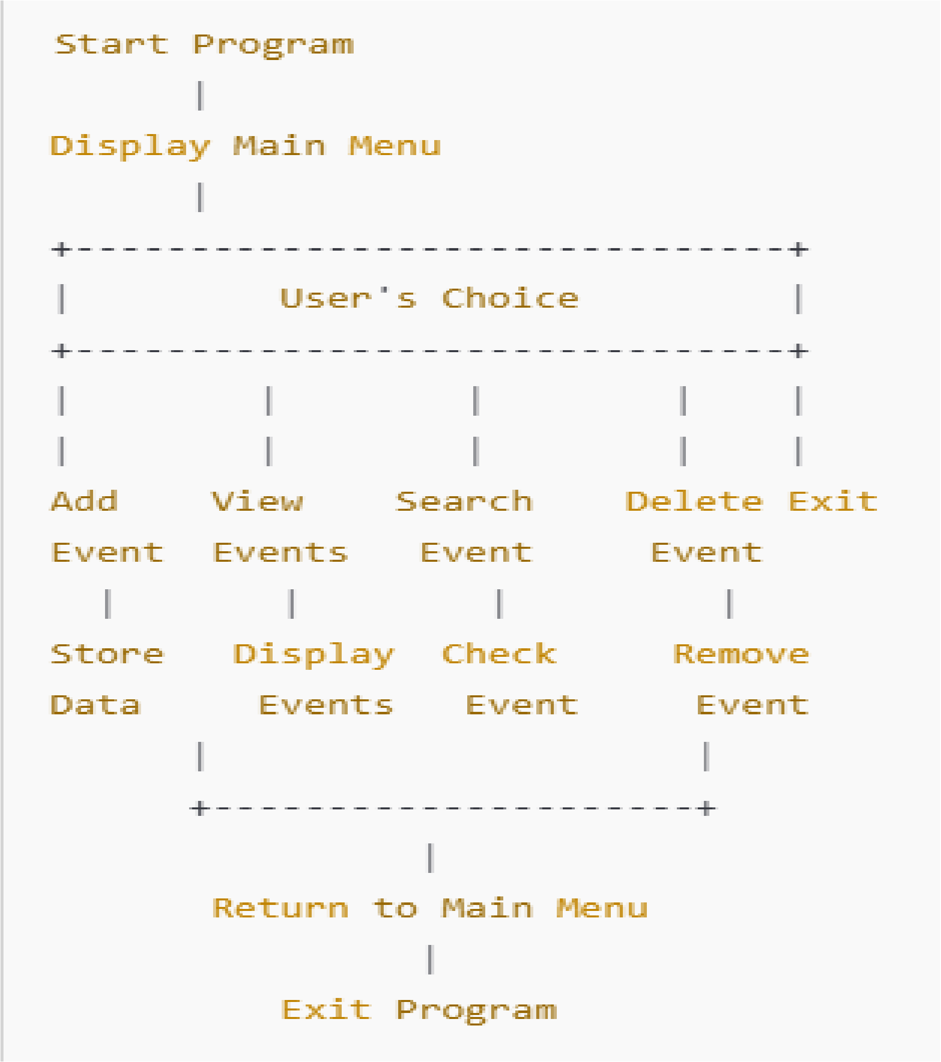
CHAPTER 2

PROJECT METHODOLOGY

2.1Proposed Work

1. **Core Functionalities:** Enable users to add event details such as event name, number of participants, judge, and location. Provide options to view all events, search for specific events, and delete events.
2. **Centralized Data Storage:** Use a dictionary to efficiently store and manage event information, ensuring quick data retrieval and updates.
3. **Menu-Driven Interface:** Design an intuitive menu system to allow easy navigation between different operations.
4. **Error Handling:** Implement checks for invalid inputs, empty records, or duplicate entries to maintain data integrity.
5. **Scalability:** Develop the program with modular functions to allow for future enhancements, such as adding event dates, generating reports, or integrating with databases.
6. **Ease of Use:** Keep the interface and operations simple, making the program accessible to school administrators, teachers, or students with basic computer skills.
7. **Educational Value:** Demonstrate key Python programming concepts, including dictionaries, loops, functions, and conditional statements, to make the program a learning resource for beginners.

2.2 Block Diagram



```
Start Program
      |
Display Main Menu
      |
+-------------------------------+
|         User's Choice         |
+-------------------------------+
|        |        |        |    |
|        |        |        |    |
Add    View    Search    Delete Exit
Event  Events   Event     Event
  |       |        |         |
Store   Display  Check     Remove
Data     Events   Event     Event
      |                     |
      +---------------------+
                |
       Return to Main Menu
                |
          Exit Program
```

CHAPTER 3

MODULE DESCRIPTION

3.1 Module 1 : Main Menu

Function Name: main()

Description: Controls the flow of the program, displaying the menu and calling the respective functions based on user input.

Steps:

1. Display the main menu with options: Add Event, View Events, Search Event, Delete Event, Exit.
2. Prompt the user to input their choice.
3. Based on the user’s input, navigate to the corresponding function (add_event(), view_events(), search_event(), delete_event(), or exit the program).
4. Repeat the process until the user selects to exit the program.

3.2 Module 2 : Add Event

Function Name : add_event()

Description:- The Add Event module enables the user to add a new sports event with its details (event name, number of participants, judge, and location).The event details are stored in the sports_events dictionary, with the event name as the key and a dictionary containing the number of participants, judge name, and location as values.This function validates the input (such as ensuring the number of participants is a valid integer) and checks if the event already exists to prevent duplicates.

Steps:

1. Prompt the user to enter the event name .
2. Ask for the number of participants and convert it to an integer .
3. Prompt for the judge's name and location of the event.
4. Store all these details in the sports_events dictionary using the event name as the key.
5. Display a confirmation message like, "Event added successfully."
6. Return to the main menu after adding the event.

3.3 Module 3 : View Events

Function Name : view_events()

Description:- The View Events module displays all the stored events and their details (number of participants, judge, and location). It handles the case where no events have been added by checking if the dictionary is empty. The program loops through the sports_events dictionary and prints the details of each event in a clear and readable format.

Steps:

1. Check if the sports_events dictionary is empty.
2. If no events exist, print "No events have been added yet."
3. If events are available, iterate over the dictionary and display the event name followed by its details (number of participants, judge, and location).
4. Return to the main menu after displaying the events.

3.4 Module 4 : Search Event

Function Name : search_events()

Description:- The Search Event module allows the user to find a specific event by its name. It prompts the user for the event name and checks if that event exists in the dictionary. If the event is found, it displays the event's details; if not, it shows a message saying the event is not found.

Steps:

1. Prompt the user to enter the event name to search for.
2. Check if the event name exists in the sports_events dictionary.
3. If the event is found, display its details (number of participants, judge, and location).
4. If not found, print an error message like, "Event not found."
5. Return to the main menu after the search operation.

3.5 Module 5 : Delete Event

Function Name : delete_events()

Description:- The Delete Event module allows the user to remove an event from the stored records.The user is asked for the event name, and the program checks if the event exists.If the event is found, it is deleted from the sports_events dictionary, and a success message is displayed. If not found, an error message is shown.

Steps:

1. Prompt the user to enter the event name to delete.
2. Check if the event exists in the sports_events dictionary.
3. If found, delete the event from the dictionary and display a success message: "Event deleted successfully."
4. If not found, display an error message: "Event not found."
5. Return to the main menu after deleting the event.

3.7 Module 6 : Exit

Function Name : exit()

Description:- The Exit module handles the program's termination. It ensures that the user is properly notified when exiting and that no other operations are performed after exiting.

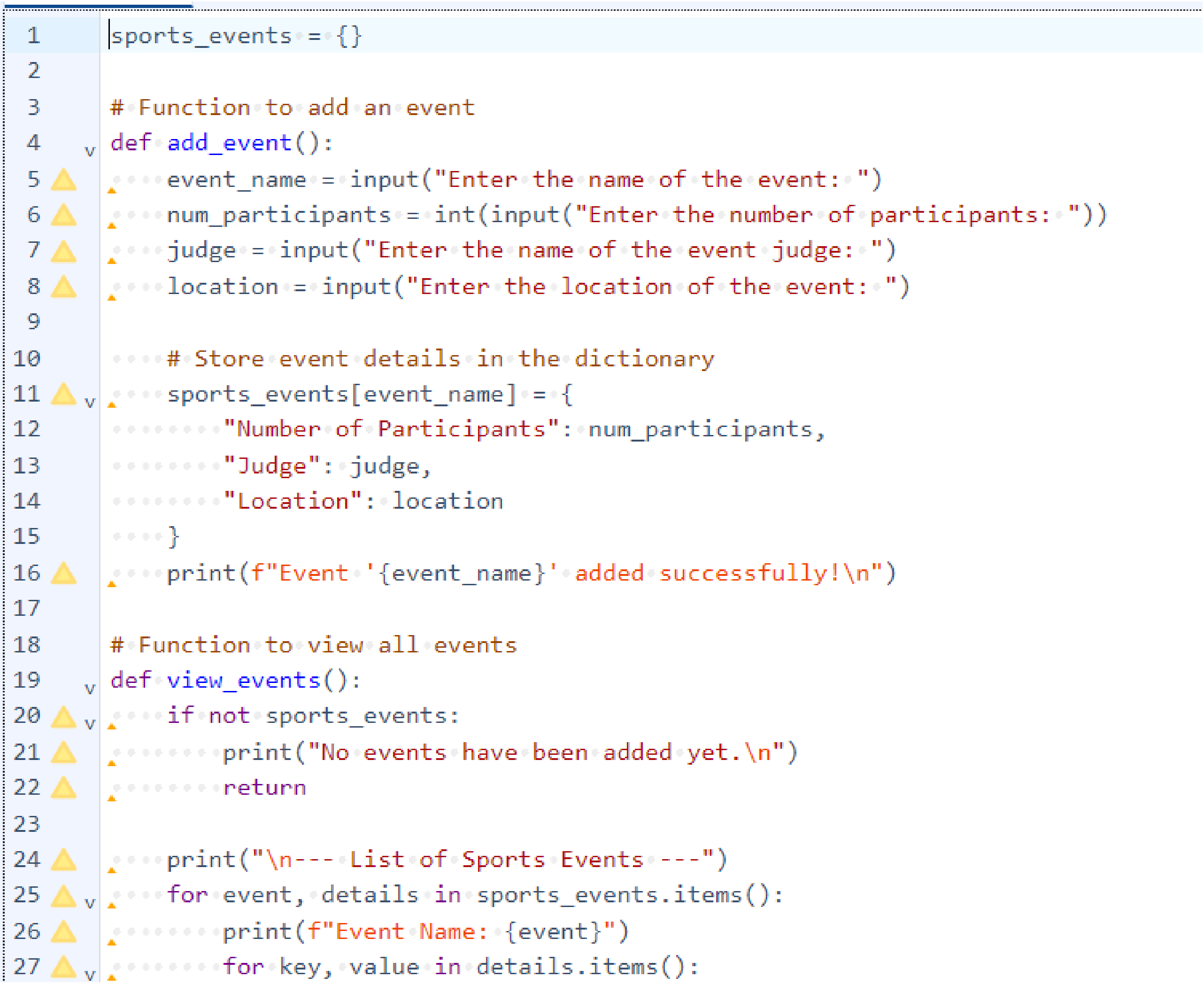
Steps:

1. Display a farewell message, e.g., "Exiting the program. Goodbye!"
2. End the program by exiting the main loop
3. Close the program and terminate the application.

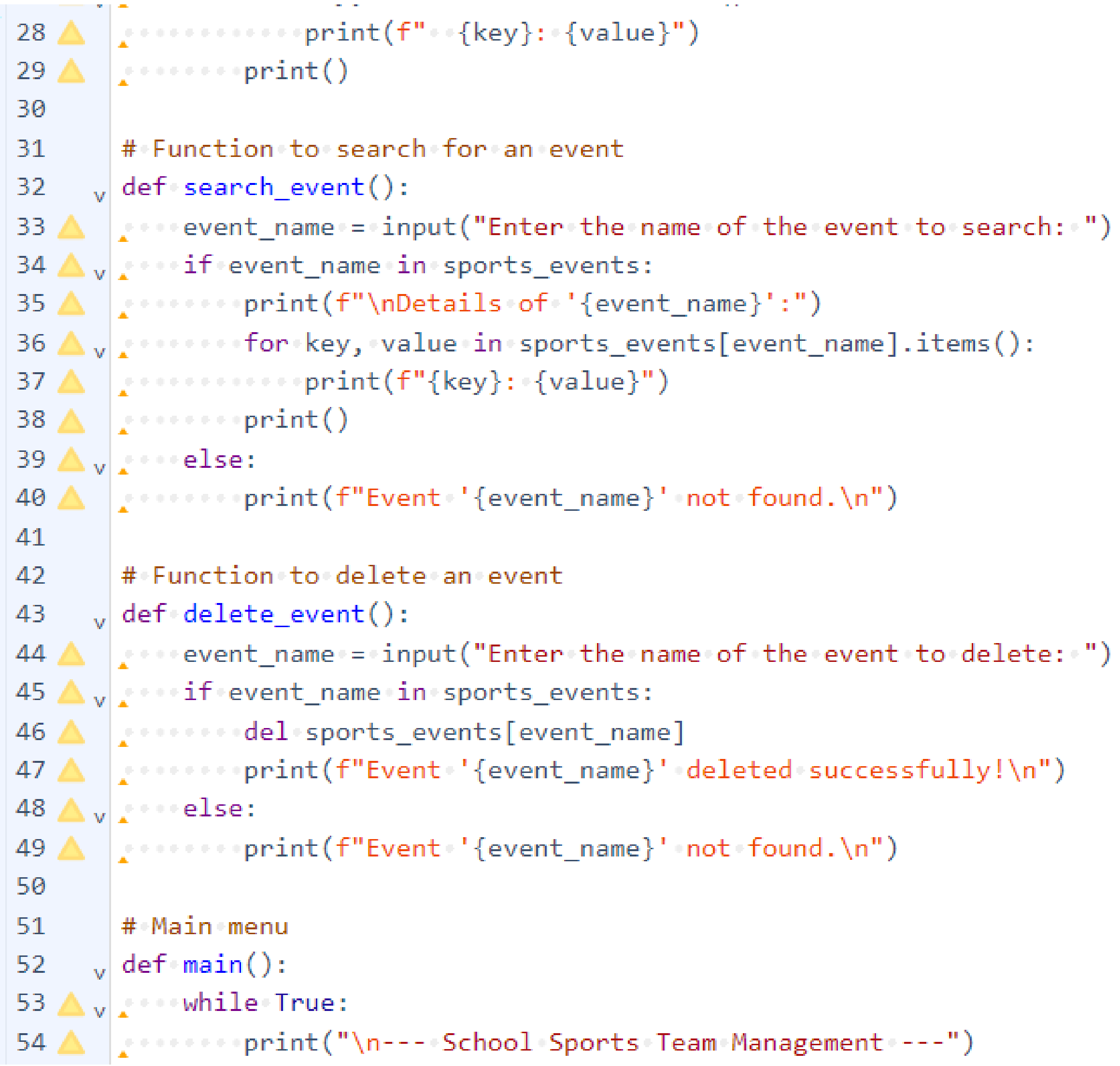
CHAPTER 4

RESULTS AND DISCUSSION

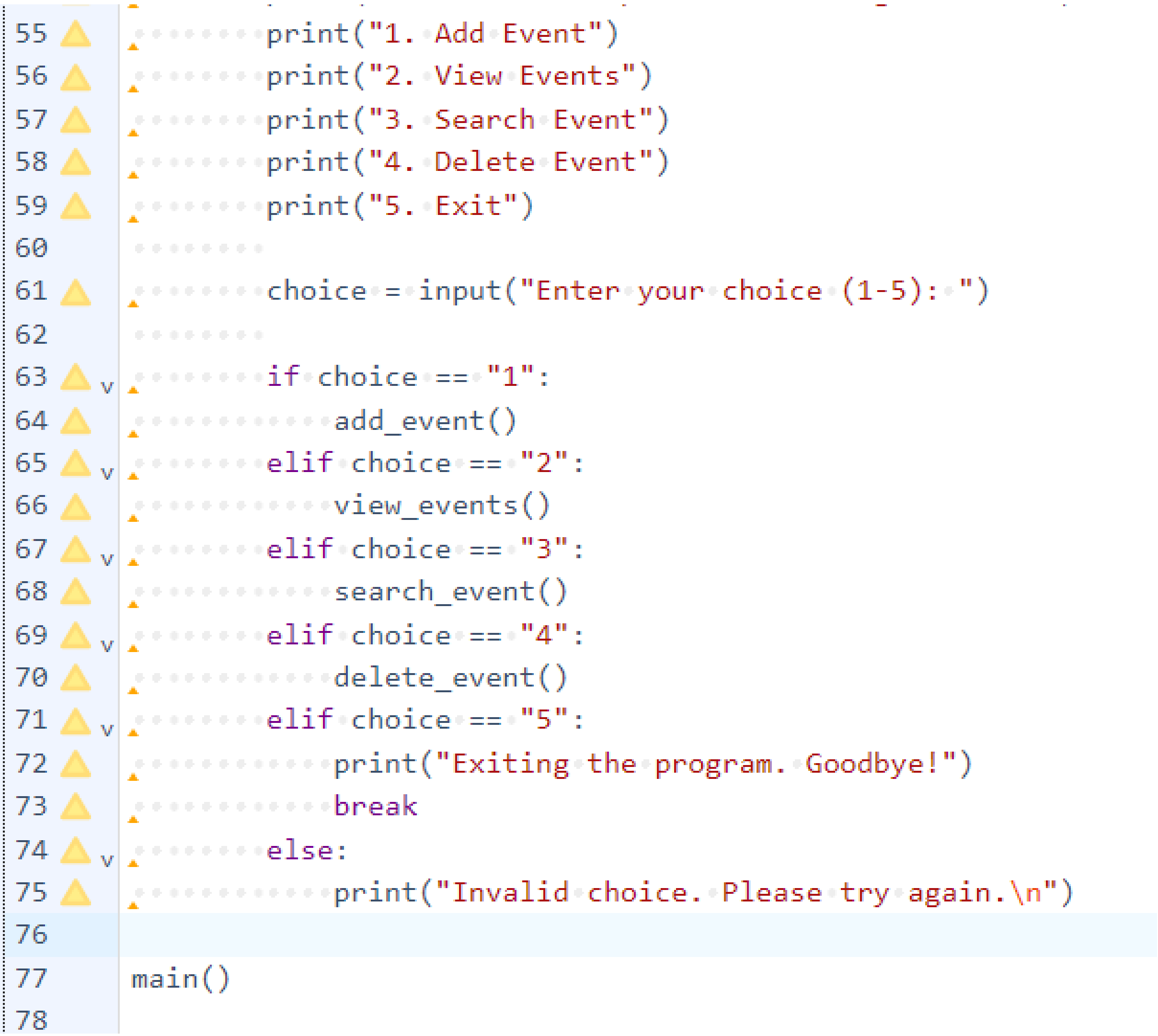
PROGRAM



```
sports_events = {}

# Function to add an event
def add_event():
    event_name = input("Enter the name of the event: ")
    num_participants = int(input("Enter the number of participants: "))
    judge = input("Enter the name of the event judge: ")
    location = input("Enter the location of the event: ")

    # Store event details in the dictionary
    sports_events[event_name] = {
        "Number of Participants": num_participants,
        "Judge": judge,
        "Location": location
    }
    print(f"Event '{event_name}' added successfully!\n")

# Function to view all events
def view_events():
    if not sports_events:
        print("No events have been added yet.\n")
        return

    print("\n--- List of Sports Events ---")
    for event, details in sports_events.items():
        print(f"Event Name: {event}")
        for key, value in details.items():
```

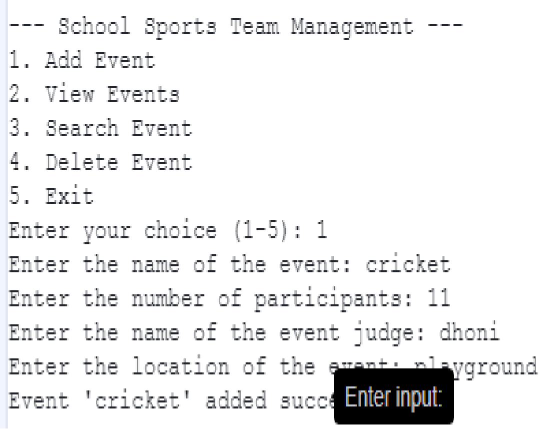


```
            print(f"  {key}: {value}")
        print()

# Function to search for an event
def search_event():
    event_name = input("Enter the name of the event to search: ")
    if event_name in sports_events:
        print(f"\nDetails of '{event_name}':")
        for key, value in sports_events[event_name].items():
            print(f"{key}: {value}")
        print()
    else:
        print(f"Event '{event_name}' not found.\n")

# Function to delete an event
def delete_event():
    event_name = input("Enter the name of the event to delete: ")
    if event_name in sports_events:
        del sports_events[event_name]
        print(f"Event '{event_name}' deleted successfully!\n")
    else:
        print(f"Event '{event_name}' not found.\n")

# Main menu
def main():
    while True:
        print("\n--- School Sports Team Management ---")
```

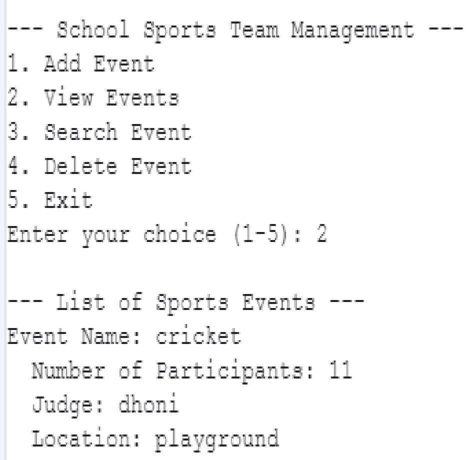


```
        print("1. Add Event")
        print("2. View Events")
        print("3. Search Event")
        print("4. Delete Event")
        print("5. Exit")

        choice = input("Enter your choice (1-5): ")

        if choice == "1":
            add_event()
        elif choice == "2":
            view_events()
        elif choice == "3":
            search_event()
        elif choice == "4":
            delete_event()
        elif choice == "5":
            print("Exiting the program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.\n")

main()
```

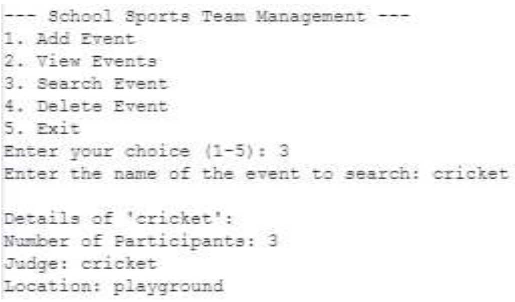
OUTPUT



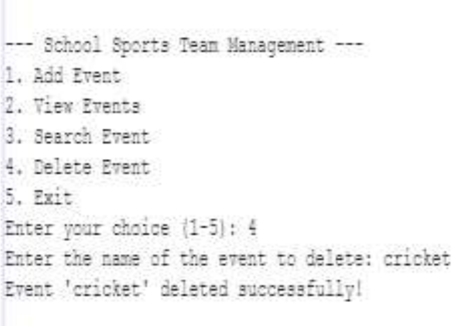
```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 1
Enter the name of the event: cricket
Enter the number of participants: 11
Enter the name of the event judge: dhoni
Enter the location of the event: playground
Event 'cricket' added succe
```



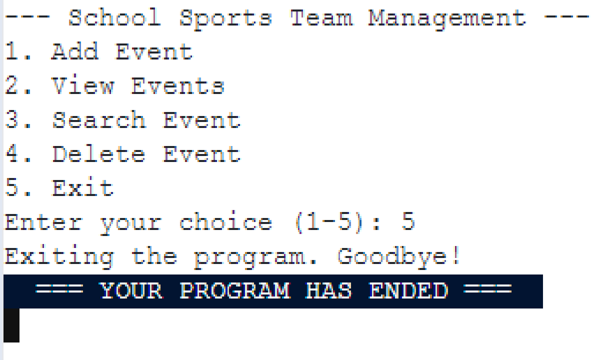
```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 2

--- List of Sports Events ---
Event Name: cricket
  Number of Participants: 11
  Judge: dhoni
  Location: playground
```



```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 3
Enter the name of the event to search: cricket

Details of 'cricket':
Number of Participants: 3
Judge: cricket
Location: playground
```



```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 4
Enter the name of the event to delete: cricket
Event 'cricket' deleted successfully!
```



```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 5
Exiting the program. Goodbye!
  === YOUR PROGRAM HAS ENDED ===
```

CHAPTER 5

CONCLUSION

The School Sports Team Management Program provides an efficient and user-friendly system for managing sports events within a school environment. By leveraging basic Python programming concepts such as functions, dictionaries, loops, and conditionals, the program allows users to perform essential tasks such as adding, viewing, searching, and deleting sports events. The modular approach ensures that each operation is handled independently, making the program flexible and easy to maintain. The program is designed with a simple, intuitive interface that guides users through each step of managing events, allowing them to input necessary details like event names, participants, judges, and locations. It also provides error handling to ensure smooth user interaction, such as checking for invalid input or events that do not exist.

REFERENCES:

1. Python Software Foundation. Retrieved from https://docs.python.

2. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/python-dictionary/

3.W3Schools. Retrieved from https://www.w3schools.com

APPENDIX

(Coding)

```
sports_events = {}

def add_event():
    event_name = input("Enter the name of the event: ")
    num_participants = int(input("Enter the number of participants: "))
    judge = input("Enter the name of the event judge: ")
    location = input("Enter the location of the event: ")

    # Store event details in the dictionary
    sports_events[event_name] = {
        "Number of Participants": num_participants,
        "Judge": judge,
        "Location": location
    }
    print(f"Event '{event_name}' added successfully!\n")

# Function to view all events
def view_events():
    if not sports_events:
        print("No events have been added yet.\n")
        return

    print("\n--- List of Sports Events ---")
    for event, details in sports_events.items():
        print(f"Event Name: {event}")
        for key, value in details.items():
            print(f"  {key}: {value}")
        print()

# Function to search for an event
def search_event():
    event_name = input("Enter the name of the event to search: ")
    if event_name in sports_events:
        print(f"\nDetails of '{event_name}':")
        for key, value in sports_events[event_name].items():
            print(f"{key}: {value}")
        print()
    else:
        print(f"Event '{event_name}' not found.\n")

# Function to delete an event
def delete_event():
    event_name = input("Enter the name of the event to delete: ")
    if event_name in sports_events:
        del sports_events[event_name]
        print(f"Event '{event_name}' deleted successfully!\n")
    else:
        print(f"Event '{event_name}' not found.\n")

# Main menu
def main():
    while True:
        print("\n--- School Sports Team Management ---")
        print("1. Add Event")
        print("2. View Events")
        print("3. Search Event")
        print("4. Delete Event")
        print("5. Exit")

        choice = input("Enter your choice (1-5): ")

        if choice == "1":
            add_event()
        elif choice == "2":
            view_events()
        elif choice == "3":
            search_event()
        elif choice == "4":
            delete_event()
        elif choice == "5":
            print("Exiting the program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.\n")

# Run the program
main()
```

APPENDIX

(Output)

```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 1
Enter the name of the event: cricket
Enter the number of participants: 22
Enter the name of the event judge: dhoni
Enter the location of the event: play ground
Event cricket  added successfully!
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5):2
--- List of Sports Events ---
Event Name: cricket
Number of participants: 22
Judge: dhoni
Location : play ground
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5): 3
Enter the name of the event to search: cricket
Details of ‘cricket’:
Number of participants: 22
Judge: dhoni
Location : play ground
```

```
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5):4
Enter the name of the event to delete: cricket
Event cricket deleted successfully!
--- School Sports Team Management ---
1. Add Event
2. View Events
3. Search Event
4. Delete Event
5. Exit
Enter your choice (1-5):5
Exiting the program. Goodbye!
```