DELHI PRIVATE SCHOOL
DUBAI

PROJECT FILE

COMPUTER SCIENCE

NAME: GOKUL SANJEEV SHENOY

CLASS: 12 A

CERTIFICATE

DELHI PRIVATE SCHOOL, DUBAI

CERTIFICATE

Certified that the Project file is the bona fide work of Master/Miss Gokul Sanjeev Shenoy Class XII A Roll no D6343 recorded in the School laboratory during academic year 2025 to 2026.

Ms. Richi Rehlan	Ms. Rose Nimmi	
Teacher in-Charge	HOD	External Examiner

ACKNOWLEDGEMENT

I would like to express my gratitude and thanks to my Computer Science faculty Ms.Richi Rehlan and HOD Ms.Rose Nimmi for her guidance and unparalleled support throughout the process of making my Computer Science Project file.

I would also like to thank my headmistress Ms.Blessy supervisors and teachers for their patience and help that has enabled me to finish this Project file on time.

I'm obliged to the Principal, Ms.Rashmi Nandkeolyar for providing me with the opportunity and required environment.

Finally, I would like to thank my family and friends for giving me motivation and support to go ahead with the Project file and it's execution.

PROJECT AIM

The main goal of this project is to create "Hoopify," a powerful Python-based application for managing, organising, and recording basketball player statistics in an organised and approachable way.

By bringing the benefits of digital stat management to practice and game settings, the project aims to assist coaches, athletes, and fans in effectively tracking important performance metrics—like points, assists, and rebounds—across numerous players and teams.

Hoopify

More specifically, Hoopify is designed to:

- Simplify the process of player record keeping and performance analysis for individual athletes and teams.
- Enable easy data entry, review, and update of player stats, supporting improvement and progress monitoring over time.
- Provide a foundation for further enhancements, such as visual analytics, match history exports, and integration with advanced sports tracking solutions.

By bridging real-world sports requirements with hands-on programming skills, this project demonstrates how technology can empower player development and informed decision-making in basketball.

PROJECT DESCRIPTION

Hoopify is a Python-based fitness tracking application.
It enables the user to do the following:

- A new basketball player with stats (points, assists, and rebounds) can be added.
- Look up a player using their ID.
- Utilise fresh match data to update player performance.
- If a player departs the roster, remove the record.
- Present every player's statistics in an organised manner.

File handling, dictionary data structures, functions, and menu-driven programming are all demonstrated in this project. It is useful for managing sports data and demonstrates how to use Python in actual tracking situations.
The application also serves as a foundation for more complex database or GUI extensions by utilising fundamental Python programming ideas like functions, dictionaries, file handling (CSV read/write), and a loop-based menu system.

Hoopify's emphasis on dependability and accessibility is similar to that of professional basketball analytics platforms, where prompt and precise stat management aids in player development, training feedback, and decision-making.

At the same time, Hoopify's ability to give users basic performance insights is what distinguishes it from other stats trackers. The program can be readily expanded to analyse trends (like top scorer or most improved player) or flag players whose stats are dropping or demonstrating notable improvement based on the data entered. This innovation, which was inspired by professional analytics tools, not only helps coaches and players swiftly identify their strengths and weaknesses but also promotes data-driven decision making for training focus and lineup changes.

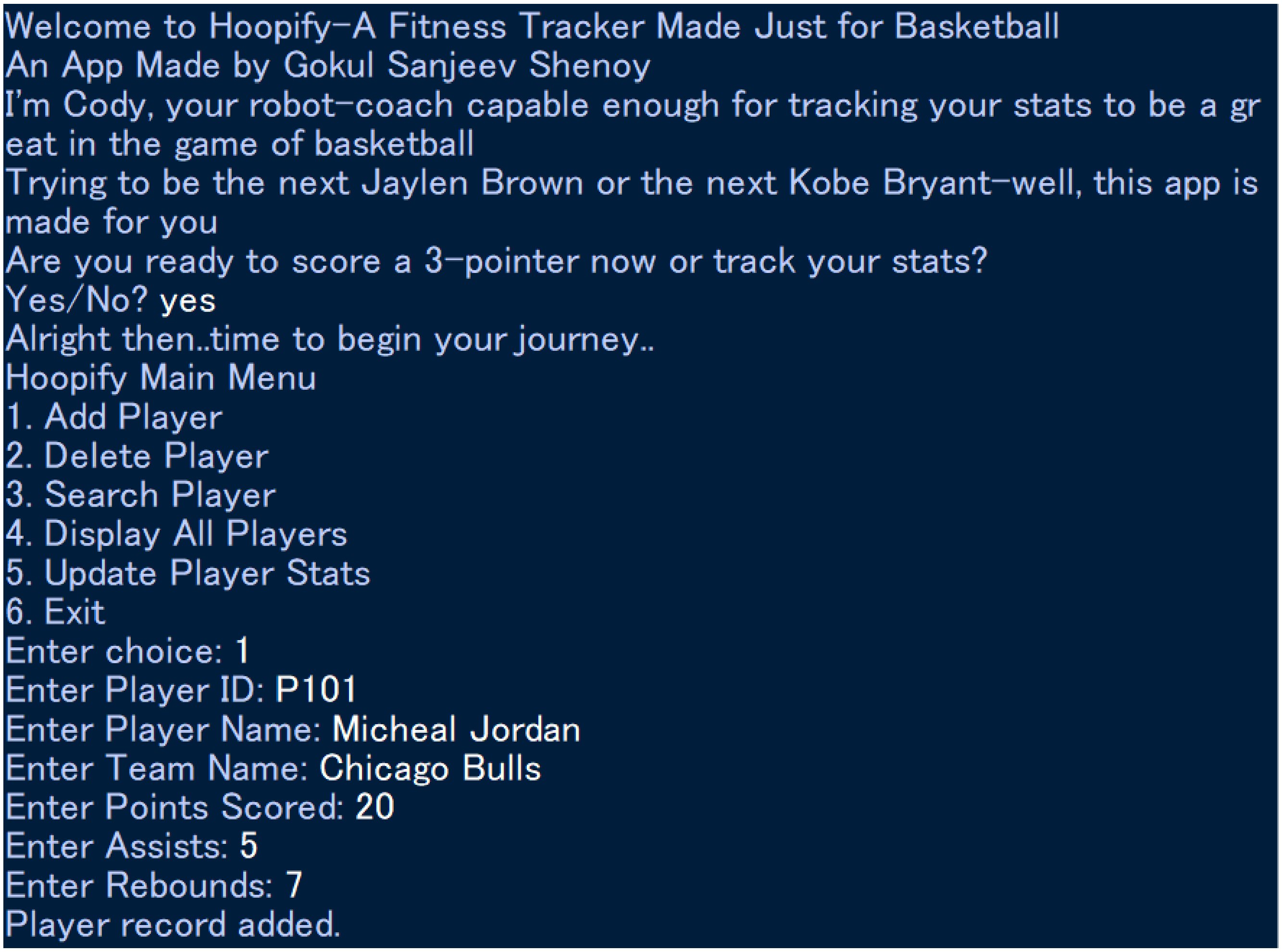
PROGRAM SOURCE CODE

```
import csv
filename = "hoopifyapp.csv"
def load_data():
    data = {}
    try:
        with open(filename, "r") as f:
            reader = csv.reader(f)
            next(reader)
            for row in reader:
                data[row[0]] = {
                    "Name": row[1],
                    "Team": row[2],
                    "Points": int(row[3]),
                    "Assists": int(row[4]),
                    "Rebounds": int(row[5])
                }
    except FileNotFoundError:
        pass
    return data

def save_data(data):
    with open(filename, "w", newline="") as f:
        writer = csv.writer(f)
        writer.writerow(["PlayerID", "Name", "Team", "Points", "Assists", "Rebounds"])
        for pid, stats in data.items():
            writer.writerow([pid, stats["Name"], stats["Team"], stats["Points"], stats["Assists"], stats["Rebounds"]])
```

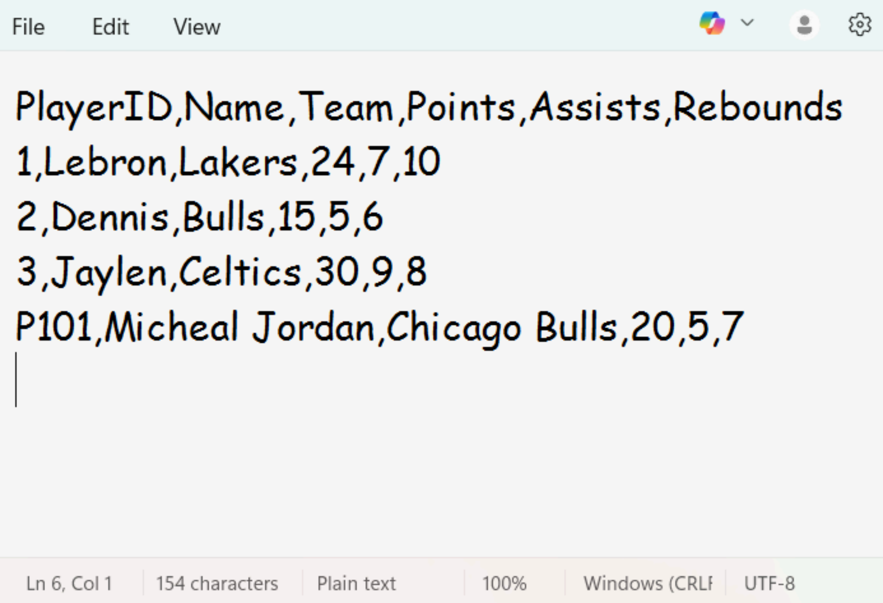
```
def add_player(data):
    pid = input("Enter Player ID: ")
    if pid in data:
        print("Player already exists.")
    else:
        name = input("Enter Player Name: ")
        team = input("Enter Team Name: ")
        points = int(input("Enter Points Scored: "))
        assists = int(input("Enter Assists: "))
        rebounds = int(input("Enter Rebounds: "))
        data[pid] = {"Name": name, "Team": team, "Points": points, "Assists": assists, "Rebounds": rebounds}
        save_data(data)
        print("Player record added.")


def delete_player(data):
    pid = input("Enter Player ID to delete: ")
    if pid in data:
        data.pop(pid)
        save_data(data)
        print("Player record deleted.")
    else:
        print("Player not found.")


def search_player(data):
    pid = input("Enter Player ID to search: ")
    if pid in data:
        p = data[pid]
```

```
  print("Found:", pid, p["Name"], "| Team:", p["Team"], "| Points:", p["Points"], "| Assists:",
   p["Assists"], "| Rebounds:", p["Rebounds"])
    else:
        print("Player not found.")


def display_players(data):
    if data:
        print("\nBasketball Player Stats:")
        for pid, p in data.items():
         print(pid, p["Name"], "| Team:", p["Team"], "| Points:", p["Points"], "| Assists:", p["Assists"], "|
Rebounds:", p["Rebounds"])
    else:
        print("No player records.")


def update_stats(data):
    pid = input("Enter Player ID to update stats: ")
    if pid in data:
        points = int(input("Enter New Points: "))
        assists = int(input("Enter New Assists: "))
        rebounds = int(input("Enter New Rebounds: "))
        data[pid]["Points"] = points
        data[pid]["Assists"] = assists
        data[pid]["Rebounds"] = rebounds
        save_data(data)
        print("Stats updated.")
    else:
        print("Player not found.")
def main_menu():
    data = load_data()
    while True:
        print("Hoopify Main Menu")
        print("1. Add Player")
        print("2. Delete Player")
        print("3. Search Player")
        print("4. Display All Players")
        print("5. Update Player Stats")
        print("6. Exit")
        choice = input("Enter choice: ")
        if choice == "1":
            add_player(data)
        elif choice == "2":
            delete_player(data)
        elif choice == "3":
            search_player(data)
        elif choice == "4":
            display_players(data)
        elif choice == "5":
            update_stats(data)
        elif choice == "6":
            print("Exiting...")
            break
        else:
            print("Invalid choice.")
#Opening
print("Welcome to Hoopify-A Fitness Tracker Made Just for Basketball")
print("An App Made by Gokul Sanjeev Shenoy")
print("I'm Cody, your robot-coach capable enough for tracking your stats to be a great in the game of basketball")
print("Trying to be the next Jaylen Brown or the next Kobe Bryant-well, this app is made for you")
print("Are you ready to score a 3-pointer now or track your stats?")
check = input("Yes/No? ").strip().lower()
if check == "yes":
    print("Alright then..time to begin your journey..")
    main_menu()
else:
    print("Alright then, keep playing and keep showing up at the court")
    print("Goodbye!")
```

SAMPLE OUTPUT



```
Welcome to Hoopify–A Fitness Tracker Made Just for Basketball
An App Made by Gokul Sanjeev Shenoy
I'm Cody, your robot–coach capable enough for tracking your stats to be a gr
eat in the game of basketball
Trying to be the next Jaylen Brown or the next Kobe Bryant–well, this app is
made for you
Are you ready to score a 3–pointer now or track your stats?
Yes/No? yes
Alright then..time to begin your journey..
Hoopify Main Menu
1. Add Player
2. Delete Player
3. Search Player
4. Display All Players
5. Update Player Stats
6. Exit
Enter choice: 1
Enter Player ID: P101
Enter Player Name: Micheal Jordan
Enter Team Name: Chicago Bulls
Enter Points Scored: 20
Enter Assists: 5
Enter Rebounds: 7
Player record added.
```



```
PlayerID,Name,Team,Points,Assists,Rebounds
1,Lebron,Lakers,24,7,10
2,Dennis,Bulls,15,5,6
3,Jaylen,Celtics,30,9,8
P101,Micheal Jordan,Chicago Bulls,20,5,7
```

POSSIBLE IMPROVEMENTS

1. For an interactive visual experience, use Tkinter/PyQt to add a Graphical User Interface (GUI).
2. For real-time stat tracking, incorporate live match data imports using web scraping or APIs.
3. Use Performance Trend Analysis to create charts and graphs that show each player's development.
4. Turn on AI-driven recommendations and Automated Performance Insights (e.g., flag declining stats, highlight top performers).
5. Use machine learning models to forecast future game results or player progress based on past performance data.
6. For more comprehensive athlete management, incorporate player fitness and injury tracking.
7. Facilitate bulk data import/export capabilities (compatibility with Excel and CSV) to streamline team onboarding and reporting.
8. Include role-based access control to differentiate between admin, coach, and player users.
9. Use OpenCV to integrate video analysis and extract movements or statistics from games that have been recorded.
10. Give players motivational feedback messages that change according to their recent performance.
11. Create mobile/web app extensions for multi-device access by utilising frameworks such as Flask or Django.
12. In the future, the project might even link to wearable technology APIs to provide real-time stat updates during live games.

BIBLIOGRAPHY

I have referred to various books and sites on the internet for text related to my project. These include:-

1. Python Official Documentation – File Handling and CSV
2. NCERT Computer Science Class XII Textbook (Python File Handling, Structured Data)
3. Al Sweigart, *Automate the Boring Stuff with Python* (No Starch Press, 2015; 3rd Ed., 2022)
4. Al Sweigart, *The Big Book of Small Python Projects* (No Starch Press, 2021)
5. Preeti Arora Class 12 Computer Science Textbook
6. GeeksforGeeks, "Python Projects for Beginners"
7. W3Schools, "Python File Handling"
8. McKay Johns, "6 Sports Analytics Project Ideas"
9. Moldstud.com, "Python in Sports Analytics: Leveraging Data for Performance Insights"
10. Career Karma, "Sports Analytics Projects"

These references help me develop my project in the best way possible. The Internet provides many sites we can go into to search for ideas and reference materials related to our research work.