**Basavarajeswarigroupof institutions**

**Ballari Institute of Technology&Management**

**AUTONOMOUS INSTITUTE UNDER VISVESVARAYATECHNOLOGICAL UNIVERSITYJNANA SANGAMA,BELAGAVI590018**

**INTERNSHIP**

**Report On**

**PHYSICAL THERAPY SCHEDULING SYSTEM**

Submitted in partial fulfilment of the requirements for the award of degree of

**Bachelor of Engineering In**

# COMPUTER SCIENCE AND ENGINEERING –

**DATA SCIENCE**

**Submitted by**

J.Nitya

3BR23CD032

**Internship Carried Out By**

## EZ TRAININGS & TECHNOLOGIES PVT.LTD

## HYDERABAD

**Internal Guide External Guide**

**MRS.PARVATHI MR.VISHAL KUMAR**

**Asst.prof,CSE-DS Technical Trainer**

**MRS.KAVYASHREE**

**Asst.prof,CSE-DS**

**BALLARIINSTITUTEOFTECHNOLOGY&MANAGEMENT**

NACCAccreditedInstitution\*

**(RecognizedbyGovt.ofKarnataka,approvedbyAICTE,NewDelhi&AffiliatedtoVisvesvarayaTechnological University, Belagavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road,Allipur, Ballar1-583 104 (Karnataka) (India) Ph: 08392 –237100 / 237190, Fax: 08392 – 237197**

**2024-2025**

**Basavarajeswarigroupofinstitutions**

# BALLARIINSTITUTEOFTECHNOLOGY&MANAGEMENT

**Autonomous instituteunderVISVESVARAYATECHNOLOGICALUNIVERSITYJNANASANGAMA,**

**BELAGAVI590018**

NACCAccreditedInstitution\*

**(RecognizedbyGovt.ofKarnataka,approvedbyAICTE,NewDelhi&Affiliated to**

**VisvesvarayaTechnologicalUniversity,Belagavi)**

**"JnanaGangotri"Campus,No.873/2,Ballari-HospetRoad,Allipur,**

**Ballar1-583104(Karnataka)(India)**

**Ph: 08392–237100/237190,Fax:08392–237197**

# DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING

CERTIFICATE

This is to certify that the Internship entitled **“ PHYSICAL THERAPY SCHEDULING SYSTEM”** has been successfully completed by  **J.Nitya** bearing USN **3BR23CD032** a bonafide student of Ballari Institute of Technology and Management,

Ballari.Forthepartial fulfillmentoftherequirementsforthe**Bachelor’sDegreein Computer**

**Science and Engineering-Data science** of theVISVESVARAYATECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2024-2025.

**Signature of Internship SignatureofHOD**

**Co-ordinator**

**MRS.PARVATHI DR.ARADHANA D**

**Asst.prof,CSE-DS Prof.and HOD**

**(CSE-DS)**

**MRS.KAVYASHREE**

**Asst.prof,CSE-DS**

**DECLARATION**

I, J.Nitya**,** second year student of Computer Science and Engineering, Ballari Institute of Technology, Ballari, declare that Internship entitled **PHYSICAL THERAPY SCHEDULING SYSTEM** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD ,Hyderabad** at “**BITM,BALLARI”.** This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

**Date: 28/09/2024 Signature of theStudent**

**Place:BALLARI**

**ACKNOWLEDGEMENT**

The satisfactions that a company the successful completion of my internship on “ PHYSICAL THERAPY SCHEDULING SYSTEM ” would be incompletewithout the mentionof people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinators **“MRS.PARVATHI (Asst.prof,CSE-DS) , MRS.KAVYASHREE (Asst.prof,CSE-DS)”** for their noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **DR. ARADHANA D,** H.O.D. Department of **Computer Science and Engineering- Data Science** for extending all her valuable support and encouragement.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **ChapterName** | **PageNo.** |
| **1** | **Day to day activity (student diary extract)** | **01** |
| **2** | **Day to day activity(student diary extract)** | **02** |
| **3** | **Abstract** | **03** |
| **4** | **Introduction of the project** | **04-05** |
| **5** | **Description** | **06-09** |
| **6** | **Algorithm** | **10-17** |
| **7** | **Output** | **18-22** |
| **8** | **Conclusion** | **23** |
| **9** | **References** | **24** |

**CHAPTER-1**

**DAYTODAYACTIVITIES**



**Internship Program on Python for BE-3rdSem students**

**From 9th September to 28th September 2024 (During 3rd semester vacations).**

**Student Name: J. Nitya USN No: 3BR23CD032 Branch:CSE-DS**

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEXPAGE** | | | |
| **Day** | **Date** | **ContentCovered** | **Signatureofthe**  **facultyin-charge** |
| **1** | **09.09.24** | **Introduction to Python, Setup & Installation,FirstPythonProgram, Variables, Data Types,**  **andBasicI/O** |  |
| **2** | **10.09.24** | **Control Structures: If-else, Loops, Functions and Modules** |  |
| **3** | **11.09.24** | **Lists,Tuples,and Dictionaries,File Handling** |  |
| **4** | **12.09.24** | **Exception Handling,Practice exercises on Python basics** |  |
| **5** | **13.09.24** | **Introduction to OOP,Classes,and Objects** |  |
| **6** | **14.09.24** | **Inheritance,Polymorphism,and Encapsulation** |  |
| **7** | **15.09.24** | **Abstract Classes and Interfaces** |  |
| **8** | **17.09.24** | **Practice exercises on OOPconcepts** |  |
| **9** | **18.09.24** | **Introduction to DSA, Arrays, and Linked Lists** |  |
| **10** | **19.09.24** | **Introduction to DSA, Arrays, and Linked Lists** |  |
| **11** | **20.09.24** | **Introduction to Stacks and Queues** |  |
| **12** | **21.09.24** | **Practice exercise on basic concept**  **(Reduce,Lambda Function,List Comprehension)** |  |
| **13** | **23.09.24** | **Introduction to Trees Data structure** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **14** | **24.09.24** | **Introduction to Graph Data structure** |  |
| **15** | **25.09.24** | **Searching Algorithms**  **Project Building & Presentations** |  |
| **16** | **26.09.24** | **Project Building & Presentations** |  |
| **17** | **27.09.24** | **Project Building & Presentations** |  |
| **18** | **28.09.24** | **Project Building & Presentations** |  |

**CHAPTER-2**

**COMPANYPROFILE**

## CompanyName: EZ Trainings and Technologies Pvt.Ltd.

**Introduction:**

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

## Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

## Services:

**CollegeTrainings:**

* Tailored training programs designed to enhance the employ ability of students.
* Industry-aligned curriculum covering technical and soft skills.
* Placement assistance and career guidance.

## DevelopmentProjects:

* End-to-end development services, from ideation to execution.
* Expertise in diverse technologies and frameworks.
* Custom solutions to meet specific business needs.

**Locations:**Hyderabad |DelhiNCR

At EZ Trainings and Technologies Pvt.Ltd.,we believe in transforming potential into excellence

**CHAPTER-3**

**ABSTRACT**

This code represents a simple therapy scheduling system for athletes using object-orientedprogramming principles. It is composed of three main classes, each with a distinct responsibility:

1. Athlete: This class encapsulates the data and behavior related to an athlete.It stores the athlete's name, sport, age, injury details, and a list of therapy sessions.The class provides methods to add new therapy

Sessions and display existing ones.

1. TherapySession:This class represents an individual therapy session.It stores the therapist's name and session notes. Each session is linked to a specific athlete.
2. Therapy Scheduler: This class acts as a manager that handles the system’s primary functionality. It maintains a list of athletes and provides methods to add athletes,schedule therapy sessions, and display all

athletes'sessions.It also searches for athletes by name and handles user interaction.

1. Encapsulation: Each class(Athlete,TherapySession,TherapyScheduler) encapsulates specific data and behavior, keeping internal details private and providing interfaces for interacting with them.
2. Modularity:The system is divided into modular classes,each with a single responsibility.
3. Abstraction:The internal details of how sessions are stored or retrieve dare hidden from the user.The user inter acts with higher-level commands like"add athlete" or "schedule session"without needing to know how the underlying data structures work.

**CHAPTER-4**

**INTRODUCTION OF THE PROJECT**

Introduction to Physical Therapy Scheduling System

In sports and athletics, injuries area common occurrence, and the recovery process is often just as important as the training regimen itself. Physical therapy plays a crucial role in ensuring that athletes

Regain the ir strength and return to peak performance. However, managing physical therapy sessions for a large number of athletes and tracking their progress can be a complex and time-consuming task. To address these challenges, this project proposes the development of a Physical Therapy Scheduling System using Python.

The Physical Therapy Scheduling System is designed to streamline the process of managing therapy sessions for injured athletes. It provides a structured platform where therapy sessions can be efficiently scheduled, and the progress of athletes can be tracked over time. This system will be built as a Proof of Concept (POC), showcasing how an application can be used to simplify the scheduling and tracking of therapy outcomes, thereby improving the overall recovery process.

One of the primary objectives of the system is to implement the CRUD (Create, Read, Update, Delete) operations necessary for managing therapy sessions. These operations will include:

SchedulingPhysicalTherapy Sessions:Users can create and schedule sessionsfor athletes, ensuring that the therapy process is organized and timely.

Tracking Therapy Outcomes: The system will allow users to record and monitor the outcomes of each session, enabling healthcare providers and coaches to assess the recovery progress of each athlete over time.

The system will be developed usingObject-OrientedProgramming(OOP) principles.Thisapproachwill ensure that the system is both modular and extensible, allowing for future updates and enhancements. OOPalso enables efficient data management through the use of standard data structures and algorithms, which will be essential for managing large volumes of athlete data and therapy session records. In

**CHAPTER-5**

**MODULEDESCRIPTION**

This code represents a Physical Therapy Scheduling System developed in Python, designed to help healthcare providers, trainers, and coaches efficiently manage therapy sessions for injured athletes. By employing Object-Oriented Programming (OOP) principles, the system is structured to facilitate the creation, management, and tracking of therapy sessions. Below is a comprehensive breakdown of each class, its attributes, methods, and the overall functionality of the system.

* 1. AthleteClass

The Athlete class serves as ablueprint for creating athlete objects, encapsulating all relevant information about an athlete's identity and therapy sessions.

Attributes:

name:A string representing the athlete's fullname.

sport:A string denoting the sport the athlete participates in,such as basketball, football,or soccer.

age:An integer representing the athlete's age.

injury\_details: A string providing specifics about the athlete's injury, including type, severity, and recovery status.

sessions: A list that stores instances of TherapySession, allowing for multiple sessions to be associated with each athlete.

Methods:

init(self,name,sport,age,injury\_details):The constructor initializes an athlete object with the provided name, sport, age, and injury details.

add\_session(self, session): This method allows the addition of a TherapySession object to the athlete's sessions list, ensuring that all therapy sessions can be tracked for each athlete.

display\_sessions(self): This method prints a formatted list of all therapy sessions associated with the athlete, displaying the therapist's name and any notes taken during the sessions. This functionality is essential for monitoring the athlete's recovery progress.

* 1. TherapySessionClass

The Therapy Session class models individualtherapy sessions that athletes undergo during their recovery process.

Attributes:

therapist:Astring containing the name ofthe therapist responsible for conducting the session.This helps in identifying who is overseeing the athlete's therapy.

notes:An optional string that allows the therapist to document important observations, treatment details, or recommendations during the session. This provides valuable context for future sessions.

Methods:

init(self, therapist, notes=""): The constructor initializes a new therapy session with the therapist's name and optional notes. By default, notes are set to an empty string if not provided.

* 1. TherapyScheduler Class

The TherapyScheduler class acts as the central controller for managing athletes and their therapysessions. This class is vital for the overall functionality of the system.

Attributes:

athletes:A list that stores instances of the Athlete class.This allows the system to maintain a comprehensive record of all athletes being tracked.

Methods:

init(self):The constructor initializes an empty list of athletes.

add\_athlete(self, athlete): This method allows users to add a newAthlete object to the scheduler's list. It validates that the athlete is a recognized object before appending it.

schedule\_session(self, athlete\_name, therapist, notes=""): This method allows the scheduling of a new therapy session for a specific athlete. It first retrieves the athlete by name using the get\_athlete method and,iffound,createsanewTherapySessionobject.Thesessionisthenaddedtotheathlete’ssessionlist, and a confirmation message is printed.

get\_athlete(self, name): This method searches for an athlete by name and returns the corresponding Athlete object. If no matching athlete is found, it returns None, ensuring that the system can validate requests effectively.

display\_all\_sessions(self):This method iterates through all athletes in the system, invoking each athlete's display\_sessions method to print all therapy sessions for each athlete. This provides a comprehensive overview of the therapy activities within the system.

* 1. Menu and Main Program Logic

At the heart of the system is a text-based command-line interface (CLI) that facilitates user interaction. The menu() function presents options for users to choose from, providing a user-friendly experience.The menu includes:

AddAthlete: Prompts the user to enter details about a new athlete, such as name, sport, age, and injury details, and adds the athlete to the system.

Schedule Therapy Session: Allows the user to schedule a therapy session for a specified athlete,inputting the therapist's name and any relevant notes.

Display All Therapy Sessions: Displays all scheduled therapy sessions for all athletes in the system, allowing for easy monitoring of therapy activities.

Exit:Terminates the program cleanly.

The program runs in a loop, continuously displaying the menu until the user opts to exit. This design ensuresaseamless experienceforusers whoneed tomanagemultipleathletesand theirtherapysessions.

Key Features and Functionalities

Object-Oriented Structure: The use of classes promotes modularity and ease of maintenance. Each class has distinct responsibilities, making the codebase cleaner and more manageable.

CRUD Functionality:The system supports fundamental CRUD operations:

Create:Newathletesandtherapysessionscanbeadded easily.

**CHAPTER-6**



**ALGORITHM**

1. Initialization
2. DefineClasses:Create the Athlete class with attributes:name,sport,age,injury\_details,andsessions. Create the TherapySession class with attributes: therapist and notes.

Create the Therapy Scheduler class to manage athletes and their therapy sessions.

1. InitializeObjects:Createan instanceofTherapyScheduler.

2.LoadApplication

1.Starttheapplication:Display the main menu with options:

AddAthlete

Schedule Therapy Session DisplayAllTherapySessions Exit

3.Create Application

1.Add Athlete: Input: Get athlete details(name,sport,age,injury details). Process:Create an instance of Athlete with the provided details.

Call the add\_athlete method of Therapy Scheduler to store the athlete. Output: Confirm that the athlete has been added.

2. Schedule Therapy Session:Input: Get athlete name, therapist name, and session notes. Process:Call the schedule\_sessions method of Therapy Scheduler with the provided details. Output:Confirm that the session has been scheduled or notify if the athlete was not found.

1. ReadApplication

1.Display All Therapy Sessions:

Process:Call the display\_all\_sessions method of TherapyScheduler.

Output:Show all therapy sessions for each athlete,including therapist details and notes.

1. UpdateApplication
2. UpdateAthleteDetails(Optionalforfutureimplementation):

Input:Get the athlete's name to update.

Process:Find the athlete using the get\_athlete method. Update the desired details (e.g., injury details).

Output:Confirm that the athlete's details have been updated.

1. UpdateTherapy SessionDetails(Optional for future implementation):

Input:Get session details to update.

Process:Find the relevant session in the athlete's session list. Update therapist or notes as needed.

Output:Confirm that the session has been updated.

1. DeleteApplication
2. DeleteAthlete(Optionalforfutureimplementation):

Input: Get athlete name to delete. Process:Find the athlete using get\_athlete.

Remove the athlete from the athletes list inTherapyScheduler. Output: Confirm that the athlete has been deleted.

1. DeleteTherapy Session(Optionalforfutureimplementation):

Input:Get athlete name and session details to delete. Process:Find the athlete and the specific session.

Remove the session from the athlete's session list. Output:Confirm that the session has been deleted.

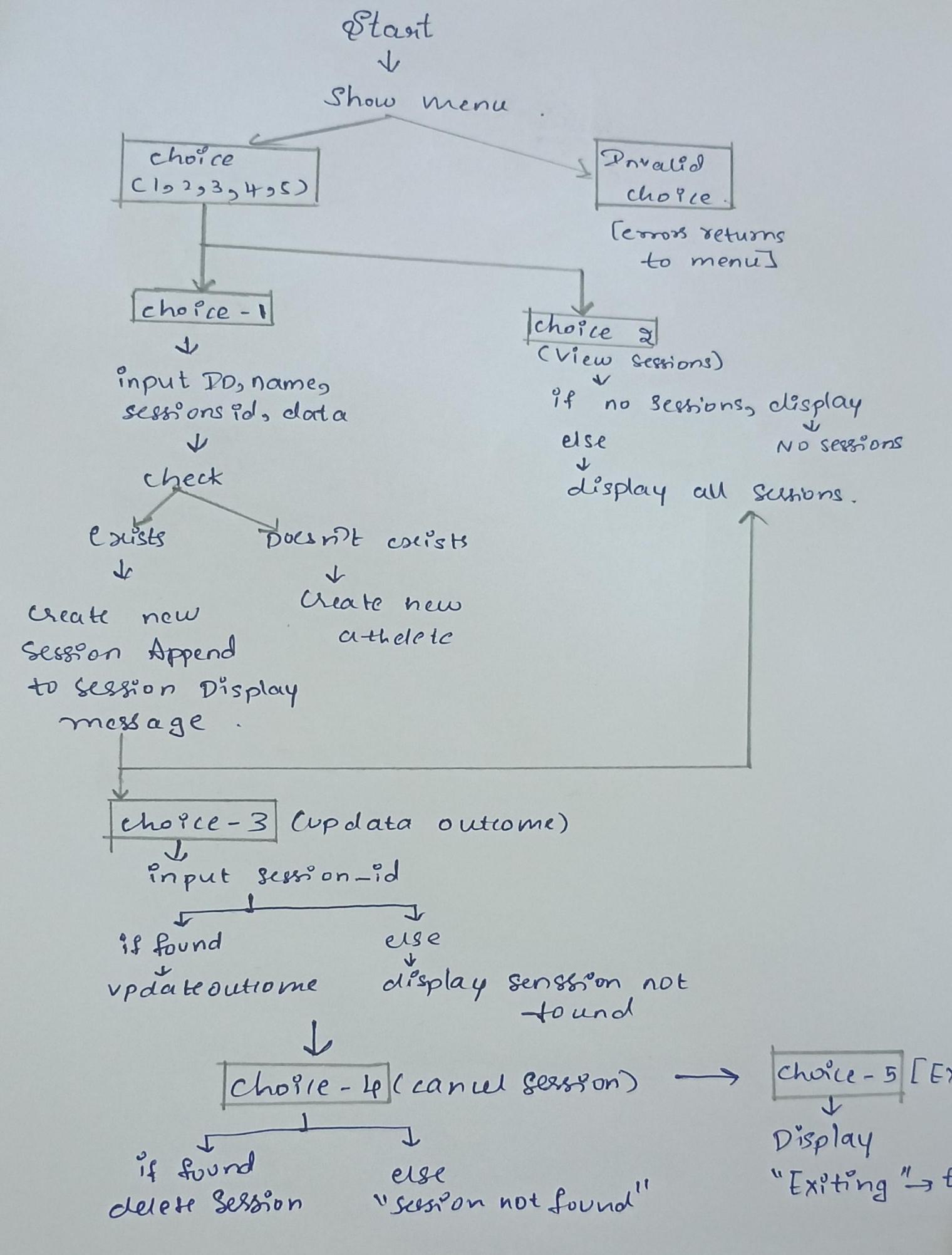
1. Save Application(Optionalforfutureimplementation):
   1. PersistData:Implement functionality to save athlete and session data to a file(e.g.,JSON,CSV)forfuture loading.

Process:Serialize the athletes list and write it to a file. Output:Confirm that the data has been saved successfully.

1. ExitApplication:
   1. TerminateProgram: Exit the application cleanly when the user selects the exit option.

**CHAPTER-7**

**FLOWCHART**



**CHAPTER-8**

**SOURCE CODE**

classAthlete:

definit(self,athlete\_id,name): self.athlete\_id=athlete\_id self.name=name

classTherapySession:

definit(self,session\_id,athlete,date): self.session\_id = session\_id

self.athlete=athlete self.date = date self.outcome=None

athletes = [] sessions=[]

def show\_menu():

print("\n---PhysicalTherapyScheduling---") print("1. Schedule a Therapy Session")

print("2.ViewallscheduledTherapySessions") print("3. Update Therapy Outcome")

print("4.CancelaTherapySession") print("5. Exit")

def schedule\_therapy(): athlete\_id=input("Enter athlete ID: ") name=input("Enter athlete name: ") session\_id=int(input("EntersessionID:")) date =input("Enter session date: ")

athlete\_exists=False forathlete inathletes:

ifathlete.athlete\_id==athlete\_id: athlete\_exists = True athlete\_name = athlete.name break

ifnot athlete\_exists:

athlete=Athlete(athlete\_id,name) athletes.append(athlete) athlete\_name = athlete.name

else:

print(f"Athletealreadyexists:{athlete\_name}")

session=TherapySession(session\_id,athlete,date) sessions.append(session)

print(f"Scheduledsession{session\_id}for{athlete.name} on {date}")

defview\_scheduled\_sessions(): if not sessions:

print("Nosessions scheduled.")

return

print("\nAllTherapySessions:") for session in sessions:

print(f"SessionID:{session.session\_id},Athlete:{session.athlete.name},Date:{session.date}, Outcome: {session.outcome}")

def therapy\_outcome():

session\_id=int(input("EntersessionIDtoupdateoutcome:")) session=None

fors in sessions:

ifs.session\_id==session\_id: session = s

break

ifsession:

outcome\_data=input("Enteroutcomedata:") session.outcome = outcome\_data

print(f"Updatedoutcomeforsession{session\_id}:{outcome\_data}") else:

print("Sessionnot found.")

def delete\_session(sessions):

session\_id=int(input("EntersessionIDtodeletesession:"))

updated\_sessions=[s for sinsessionsifs.session\_id!=session\_id]

if len(updated\_sessions) < len(sessions): print(f"Deletedsession{session\_id}.")

else:

print("Sessionnot found.")

returnupdated\_sessions

def main():globalsessions

while True: show\_menu()

choice=int(input("Enteryourchoice: "))

if choice == 1: schedule\_therapy()

elif choice == 2: view\_scheduled\_sessions()

elif choice == 3: therapy\_outcome()

elif choice == 4: sessions=delete\_session(sessions)

elif choice == 5: print("Exiting...") break

else:

print("Invalidchoice!")

main()

**CHAPTER-9**

---PhysicalTherapyScheduling---

1. Schedule a Therapy Session
2. View all scheduled Therapy Sessions
3. Update Therapy Outcome
4. Cancel a TherapySession
5. Exit

Enter your choice: 1 Enter athlete ID:cd086

Enter athlete name:sanjana

Enter session ID: 234

Enter session date:22/09/2024

**OUTPUT**

Scheduled session 234 for sanjana on 22/09/2024

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. View all scheduledTherapySessions
3. Update Therapy Outcome
4. Cancel a Therapy Session
5. Exit

Enter your choice:3

Enter sessionID to update outcome:243

Session not found.

---PhysicalTherapyScheduling---

1. Schedule a Therapy Session
2. View all scheduled Therapy Sessions
3. Update Therapy Outcome
4. CancelaTherapySession
5. Exit

Enter your choice:3

Enter session ID to update outcome: 234

Enter outcome data:sprain in ankle. Recommended rest for 5days

Updated outcome for session 234:sprain in ankle.Recommended rest for 5days

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:2

days

All Therapy Sessions:

SessionID: 234,Athlete: sanjana,Date: 22/09/2024, Outcome: sprain in ankle.Recommended rest for 5

---PhysicalTherapyScheduling---

1. Schedule a Therapy Session
2. View all scheduled Therapy Sessions
3. Update Therapy Outcome
4. Cancel a TherapySession
5. Exit

Enteryourchoice:4

Enter session ID to delete session:243 Session not found.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:4

EntersessionIDtodeletesession:234 Deleted session 234.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enter your choice: 1 EnterathleteID:R07

Enterathletename:George

Enter session ID: 347

Enter session date:28/09/2024

Scheduled session 347 for George on 28/09/2024

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. Cancel a Therapy Session
5. Exit

Enter your choice:3

Enter session ID to update outcome:247 Session not found.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:3

Enter session ID to update outcome:347

Enter outcome data:Sprained ankle of right leg. Recommended rest for 3days

Updated outcome for session347:Sprained ankle of right leg.Recommended rest for 3days

---PhysicalTherapyScheduling---

1. Schedule aTherapy Session
2. View all scheduled Therapy Sessions
3. Update Therapy Outcome
4. Cancel a Therapy Session
5. Exit

Enter your choice:2

All Therapy Sessions:

Session ID: 347,Athlete: George, Date: 28/09/2024, Outcome: Sprained ankle of right leg.Recommended rest for 3 days

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:4

EntersessionIDtodeletesession:234 Session not found.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:4

EntersessionIDtodeletesession:347 Deleted session 347.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:4

Enter session ID to delete session:347 Session not found.

---PhysicalTherapyScheduling---

1. ScheduleaTherapy Session
2. ViewallscheduledTherapySessions
3. UpdateTherapyOutcome
4. CancelaTherapySession
5. Exit

Enteryourchoice:5 Exiting...

**CHAPTER-10**

**CONCLUSION**

The therapy scheduling system for athletes is a well-structured application that illustrates the principles ofobject-orientedprogramming(OOP)andeffectivedatamanagement.Thiscodeeffectively modelsthe interactionbetween athletes and theirtherapy sessions,enabling a streamlined process for managing and tracking therapeutic care.

1. Object-Oriented Design: The code employs OOP principles, encapsulating the properties and behaviors related toathletes and therapy sessionswithindedicatedclasses.TheAthlete,TherapySession, and TherapyScheduler classes work together seamlessly, promoting modularity and enhancing the maintainability of the code. This design allows for easy extensions and modifications in the future, should the requirements evolve.
2. Functional User Interface: The program features an interactive menu system that facilitates user engagement. Users can easily navigate through the options to add athletes, schedule therapy sessions, and view all scheduled sessions. This simplicity enhances user experience and reduces the likelihood of errors, making the application accessible to users who may not be technically inclined.
3. Efficient Data Management: The TherapyScheduler class acts as a central repository for managing athletesandtheircorrespondingsessions.Thisclearseparationofconcernsensuresthatdatahandlingis

efficient and organized. The ability to search for athletes by name and associate multiple therapy sessions with each athlete demonstrates a thoughtful approach to data relationships.

1. Error Handling: The code includes basic error handling mechanisms, providing users with feedback when attempting to schedule a session for a non-existent athlete. While this is a foundational approach,it lays the groundwork for further enhancements, such as more robust validation or exception handlingto manage user input effectively.

The therapy scheduling system for athletes is a well-structured application that illustrates the principles ofobject-orientedprogramming (OOP) and effective data management.This code effectively modelsthe interactionbetween athletes and theirtherapy sessions,enabling a streamlined process for managing and tracking therapeutic care.

**CHAPTER-11**

**REFERENCES**

Google

ChatGPT(PythonGpt)