SPORTS PORTAL

A Report

Submitted in partial fulfilment of the Requirements for the completion of

THEME BASED PROJECT

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY

By

P JYOSHNA REDDY 1602-21-737-023
B KIRAN KUMAR 1602-21-737-025
M SAI RISHIK 1602-21-737-302

Under the guidance of
C SIREESHA
ASSISTANT PROFESSOR

Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE.

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

2024

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

Department of Information Technology

DECLARATION BY CANDIDATES

We, **P JYOSHNA REDDY, B KIRAN KUMAR, M SAI RISHIK** bearing hall ticket number, **1602-21-737-023, 1602-21-737-025, 1602-21-737-302** hereby declare that the project report entitled **"SPORTS PORTAL"** under the guidance of C SIREESHA , ASSISTANT PROFESSOR, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the completion of Themebased project , VI semester, Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other institutes.

P JYOSHNA REDDY 1602-21-737-023
B KIRAN KUMAR 1602-21-737-025
M SAI RISHIK 1602-21-737-302

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**SPORTS PORTAL**" being submitted by **P JYOSHNA REDDY, B KIRAN KUMAR, M SAI RISHIK** bearing **1602-21737-023, 1602-21-737-025, 1602-21-737-302,** in partial fulfillment of the requirements for the completion of Theme-based project of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

C Sireesha
Assistant Professor

External Examiner

Dr. K. Ram Mohan Rao

Professor, HOD IT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **C SIREESHA , ASSISTANT PROFESSOR, Information Technology** under whom we executed this project. Her constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor and HOD, Information Technology,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to Dr. S. V. Ramana**, Principal** of **Vasavi College of Engineering** for giving the required information in doing my project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

We also express our sincere thanks to the Management for providing excellent facilities. Finally, we wish to convey our gratitude to our family who fostered all the facilities that we need.

ABSTRACT

Designed to elevate athletes' performance, prevent injuries, and facilitate seamless access to training resources. Our innovative Injury Prediction Module employs cutting-edge predictive analytics to anticipate potential injuries, empowering athletes to proactively address risks and maintain peak physical condition. Through our Nearest Academy Locator feature, athletes can easily discover nearby training centers and coaching facilities tailored to their specific needs and preferences, ensuring they receive expert guidance and access to state-of-the-art amenities. Our directory of Coaching Centers further streamlines the search process, offering a curated selection of reputable institutions across various sports disciplines. Additionally, our platform provides a wealth of self-learning guides and resources curated by industry experts, covering topics ranging from technique refinement to mental conditioning and nutrition. With the Sports Prodigy Portal, athletes can embark on a journey of continuous improvement, equipped with the tools and knowledge needed to excel in their chosen sport while staying injury-free.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

1.1 OVERVIEW

The Sports Portal is a groundbreaking platform designed to revolutionize the landscape of sports training and athlete development. In an era where the demands on athletes are ever increasing, our project addresses a critical gap in the current system by integrating performance enhancement and injury prevention into a single, cohesive solution. The Sports Prodigy Portal is built on the pillars of advanced predictive analytics, seamless access to training resources, and comprehensive self-learning materials, ensuring athletes receive holistic support throughout their athletic journey. One of the core features of our platform is the Injury Prediction Module, which leverages state-of-the-art predictive analytics to forecast potential injuries. This proactive approach empowers athletes to identify and mitigate risks before they become detrimental, allowing for sustained peak performance and reduced downtime. By anticipating injuries, athletes can focus on their training with confidence, knowing they have a tool that helps safeguard their physical well-being.

Additionally, The Sports Portal offers an unparalleled suite of training resources, curated for athletes of all levels. From expert video tutorials to interactive workout plans, our platform provides access to best practices and cutting-edge techniques. Our robust library of self-learning materials, including articles and research papers, empowers athletes with knowledge. Furthermore, our community feature fosters a collaborative environment where athletes can share experiences and seek advice, creating a supportive network. This comprehensive approach enhances physical performance while nurturing mental and educational growth, equipping athletes to excel both on and off the field.

1.2 PROBLEM STATEMENT

The current landscape of sports training and athlete development lacks an integrated solution that addresses both performance enhancement and injury prevention. Athletes often struggle to find reliable resources for locating nearby academies, accessing quality coaching, and obtaining self-learning materials. Additionally, the absence of predictive tools for injury prevention leaves athletes vulnerable to unforeseen setbacks that could hinder their progress and jeopardize their long-term athletic careers. This gap highlights the need for a comprehensive sports portal that seamlessly integrates features for injury prediction, academy location, coaching center discovery, and self-learning guidance, providing athletes with the necessary support to optimize their performance and safeguard their physical well-being.

The Sports Portal is a groundbreaking platform designed to revolutionize the landscape of sports training and athlete development. Our project addresses this critical gap by integrating performance enhancement and injury prevention into a single, cohesive solution. Built on the pillars of advanced predictive analytics, seamless access to training resources, and comprehensive self-learning materials, the platform ensures athletes receive holistic support throughout their athletic journey. One of the core features is the Injury Prediction Module, leveraging state-of-the-art analytics to forecast potential injuries, empowering athletes to identify and mitigate risks proactively. By anticipating injuries, athletes can train confidently, knowing they have a tool that helps safeguard their physical well-being.

1.3 MOTIVATION OF THEME & TITLE

The motivation behind the theme and title of the project stems from a desire to empower athletes to reach their full potential while prioritizing their health and well-being. "Sports Prodigy Portal" embodies the vision of creating a platform where athletes can access the tools, resources, and support they need to excel in their chosen sport. The term "Prodigy" evokes a sense of exceptional talent and potential, reflecting our commitment to nurturing athletes' innate abilities. "Portal" suggests a gateway or access point, highlighting our goal of providing athletes with a centralized hub for all their training and development needs.

By combining these elements, the theme and title convey a sense of ambition, innovation, and inclusivity. We aim to inspire athletes of all levels to embark on a journey of continuous improvement, guided by expert coaching, cutting-edge technology, and a supportive community. Ultimately, the Sports Prodigy Portal seeks to redefine the sports training experience, empowering athletes to unleash their full potential and achieve greatness both on and off the field.

2. LITERATURE SURVEY

The literature survey for the Sports Prodigy Portal encompasses an in-depth review of current research and existing solutions in the areas of sports training, injury prevention, and resource accessibility, highlighting the need for an integrated platform. In the realm of sports training and performance enhancement, recent studies underscore the importance of personalized training regimens and the use of data-driven methodologies. Kellmann et al. (2018) demonstrate that individualized training programs incorporating continuous athlete monitoring and feedback significantly boost performance outcomes. Similarly, wearable technologies and mobile applications have become crucial in tracking performance metrics such as heart rate, speed, and biomechanics, providing athletes with real-time data to optimize their training .Despite these advancements, the existing tools often function in isolation, lacking integration into a unified system that can holistically support an athlete's development journey.

- Kellmann, M., Bertollo, M., Bosquet, L., Brink, M., Coutts, A. J., Duffield, R., ... & Robazza, C. (2018). Recovery and performance in sport: consensus statement. International Journal of Sports Physiology and Performance.
- Baca, A., Pansi, W., & Kornfeind, P. (2018). Wearable technology in team sports. Advances in Intelligent Systems and Computing,
- Claudino, J. G., Capanema, D. O., de Souza, T. V., Serrão, J. C., Machado Pereira, A. C., & Nassis, G. P. (2019). Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review.
- Baker, J., Schorer, J., & Wattie, N. (2017). Compromising talent: Issues in identifying and selecting talent in sport.
- Gabbett, T. J., & Ullah, S. (2012). Relationship between running loads and soft-tissue injury in elite team sport athletes. Journal of Strength and Conditioning Research.
- Figueiredo, P., Seifert, L., & Vilas-Boas, J. P. (2014). Individual profiles of spatio-temporal coordination in swimming: Finis Tempo Trainer Pro™ assessment tool. Journal of Sports Scienc

3. EXISTING SYSTEM

The current landscape of sports training and athlete development is plagued by a lack of cohesion and accessibility, presenting multiple challenges. Athletes often face fragmented resources, having to navigate multiple platforms or sources to find comprehensive training materials, coaching, and facilities. This disjointed approach hinders their ability to access the support they need effectively. Moreover, existing systems generally lack predictive tools for injury prevention, leaving athletes susceptible to unforeseen setbacks that can disrupt their progress and performance. Finding nearby academies and coaching centers is also daunting, as existing directories are often outdated or incomplete, complicating the search for suitable training opportunities. Additionally, the availability of high-quality self-learning resources is limited, impeding athletes' ability to independently enhance their skills. Overall, the current system fails to provide a unified platform that comprehensively addresses athletes' diverse needs for training, injury prevention, and self-improvement. This highlights the urgent need for an integrated solution that offers holistic support and resources to athletes at all levels.

4. PROPOSED SOLUTION

Creating an integrated hub that consolidates all necessary resources for athletes, including training materials, coaching, and facilities.

Implementing advanced predictive analytics to foresee potential injuries, enabling athletes to take preventative measures.

Developing a feature that helps athletes easily find nearby training academies and coaching centers with up-to-date and comprehensive listings.

Providing a detailed directory of reputable coaching centers across various sports disciplines, simplifying the search for quality coaching.

4.1. SYSTEM DESIGN

4.1.1 ARCHITECTURE DESIGN

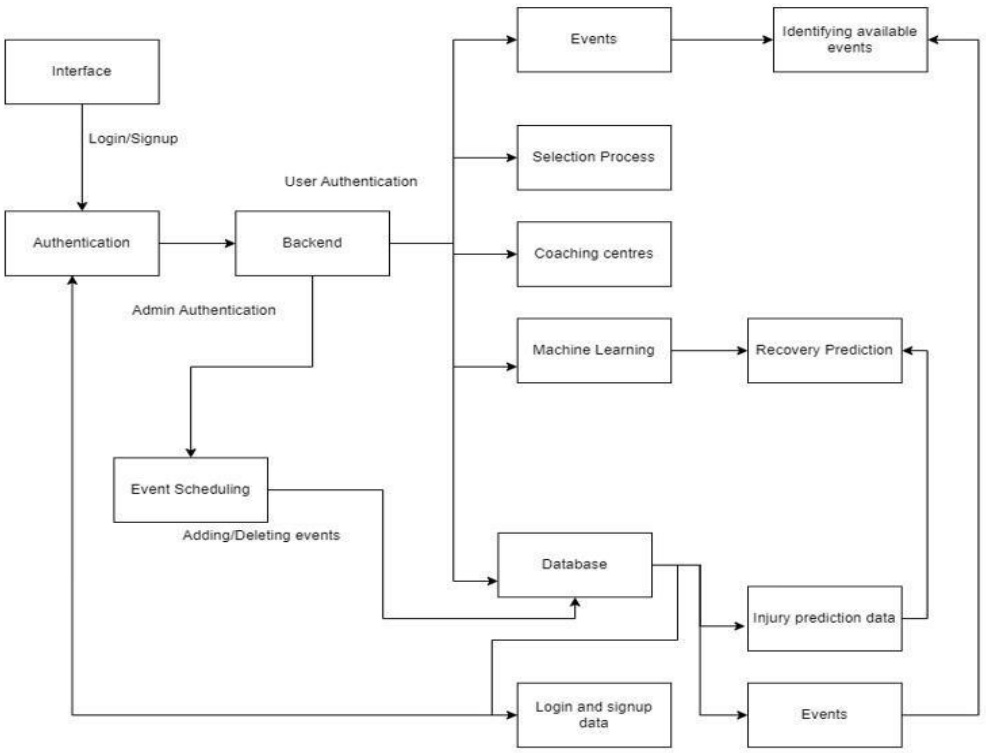

Fig: Architecture diagram

Architecture diagram : The architecture diagram for the sports portal is structured into three essential layers. The User Interface layer is where users interact, managing tasks like login, event scheduling, athlete selection, and coaching access. Behind this, the Application layer handles core functionalities such as injury prediction, recovery forecasting, event integration, and athlete selection processes. Supporting all these operations is the Data Management layer, which stores critical data such as user profiles, event details, athlete records, and machine learning models. This architecture ensures the portal operates seamlessly, providing users with a reliable and efficient experience.

4.1.2 USE-CASE DIAGRAM

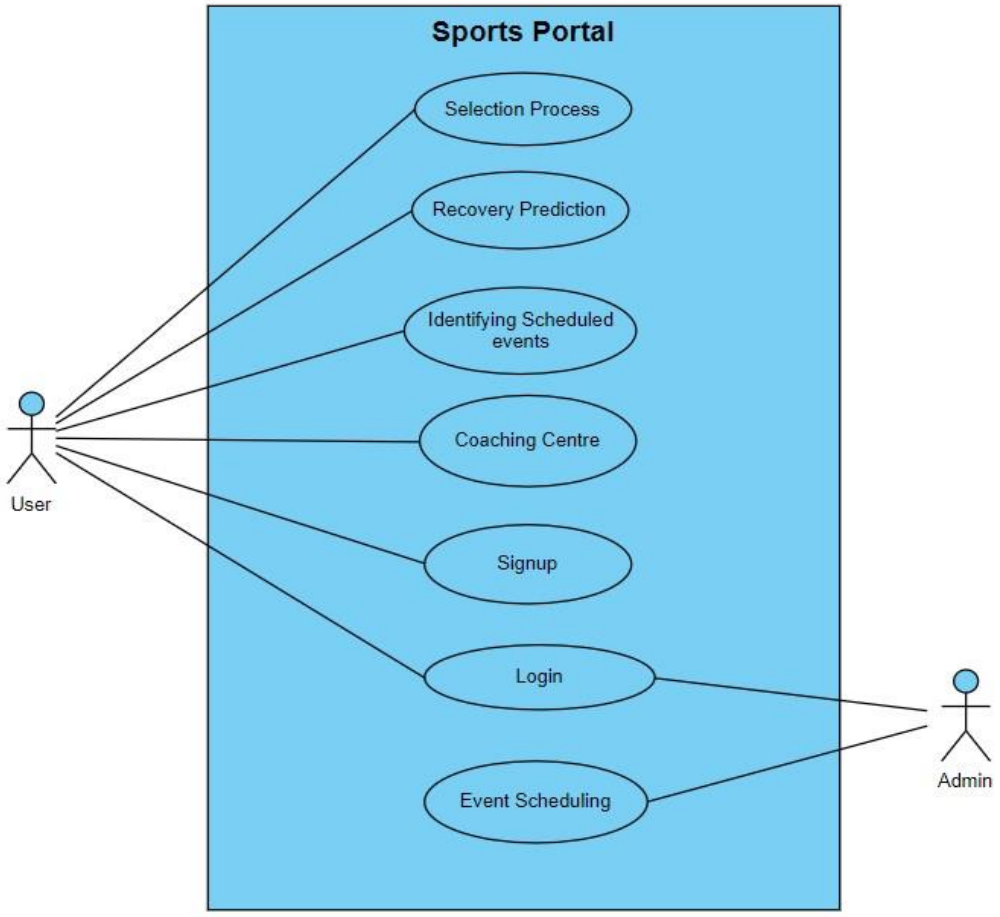

Fig : Use Case Diagram

Use Case Diagram : In the sports portal, athletes can easily register for upcoming events with a streamlined process. After logging in, athletes browse event listings, select their desired event, and review detailed information like dates and locations. Upon choosing to participate, they simply click to register, receive immediate confirmation, and, if required, make secure payments through integrated gateways. The portal updates their dashboard to reflect the event registration, ensuring athletes stay informed and prepared for their upcoming sports engagements effortlessly.

4.1.2.1 USE-CASE DESCRIPTION

User id : id001

Name : login

Actors : user, system

Description : user can securely login to their account.

User	System
Give the credentials that are required.	
	Verify the details and secure the logins
After verification securely logged in the account	

Login : 4.1.2.1.1

User id: id002
Name : Event integration

Actors : admin, system

Description: admin can add events and update any sports and display updates .

Admin	System
Admin can add events and save changes.	
	Save changes and update in the system.
Check the details and updates , modify any changes required.	

Event Integration : 4.1.2.1.2

User id:id003

Name: Event registration

Actors: user, system

Description: User have to register for the event to participate in the event(sport)

User	System
Search for updates and events.	Update the system with the updates .
Give the details required for registration	Save the details that given by user
Register for the sport with team participation or individual.	
	Update with team for the sport

Event Registration : 4.1.2.1.3

User id : id004

Name : Injury recovery prediction

Actors:user, system

Description: The user has to give the injury type, age , gender to predict number of days of recovery.

User	System
Give injury type, age and gender.	
	Save the updates and search the type of injury using ML algorithm,
Check for number of days, search for another recovery, or go to the homepage.	

Injury Recovery prediction : 4.1.2.1.4

4.2 FUNCTIONAL MODULES

4.2.1 SCREENSHOTS & RESULTS AND PSEUDOCODE

4.2.1.1 LOGIN AND SIGNUP

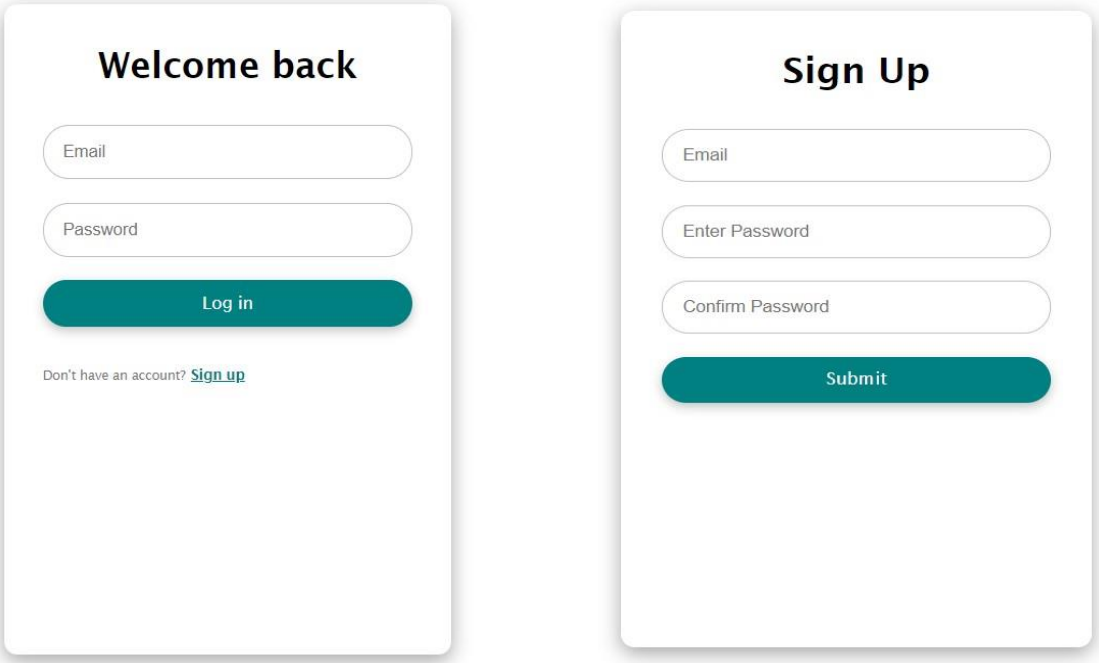

Fig 4.2.1.1.1 Login & signup

Login & Signup : The Sports Portal offers a seamless login and signup experience for users. Athletes can easily create an account to access personalized features such as event registration, coaching center selection, and event calendar management. The platform ensures secure login credentials and straightforward signup steps, enabling athletes to quickly engage with all the portal's resources and functionalities.

4.2.1.2 HOME PAGE

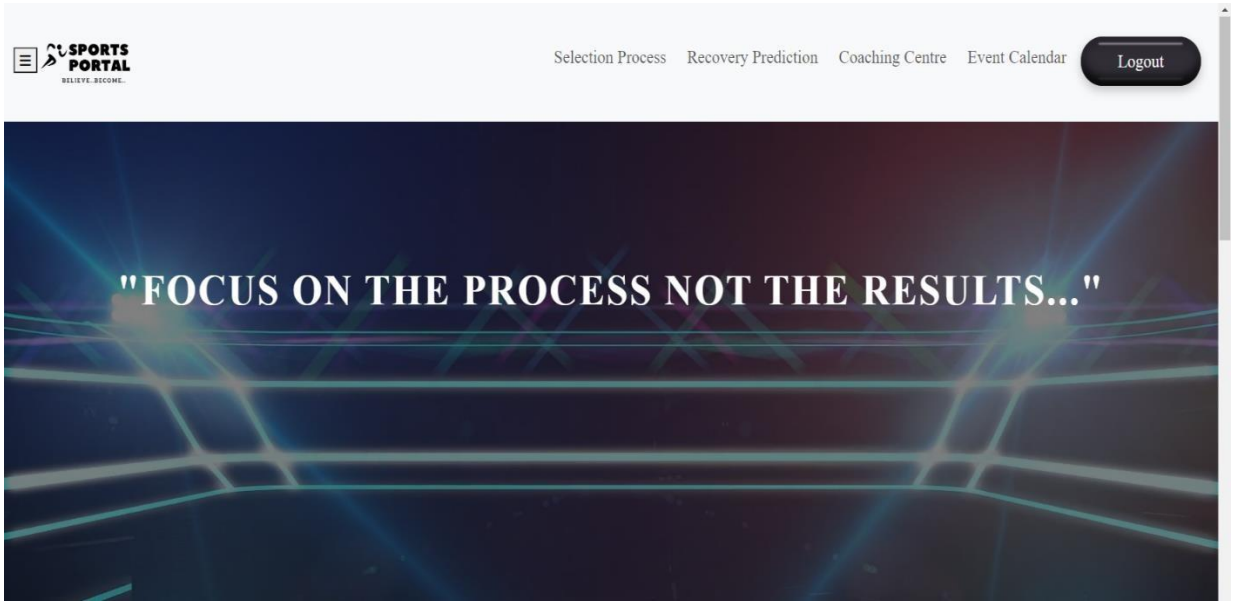

Fig 4.2.1.1.2 Home Page

Home Page : The Sports Portal homepage offers athletes a user-friendly interface with quick access to upcoming events, featured coaching centers, and registration links. It highlights athlete testimonials, promoting community engagement and success stories.

4.2.1.3 SELECTION PROCESS:

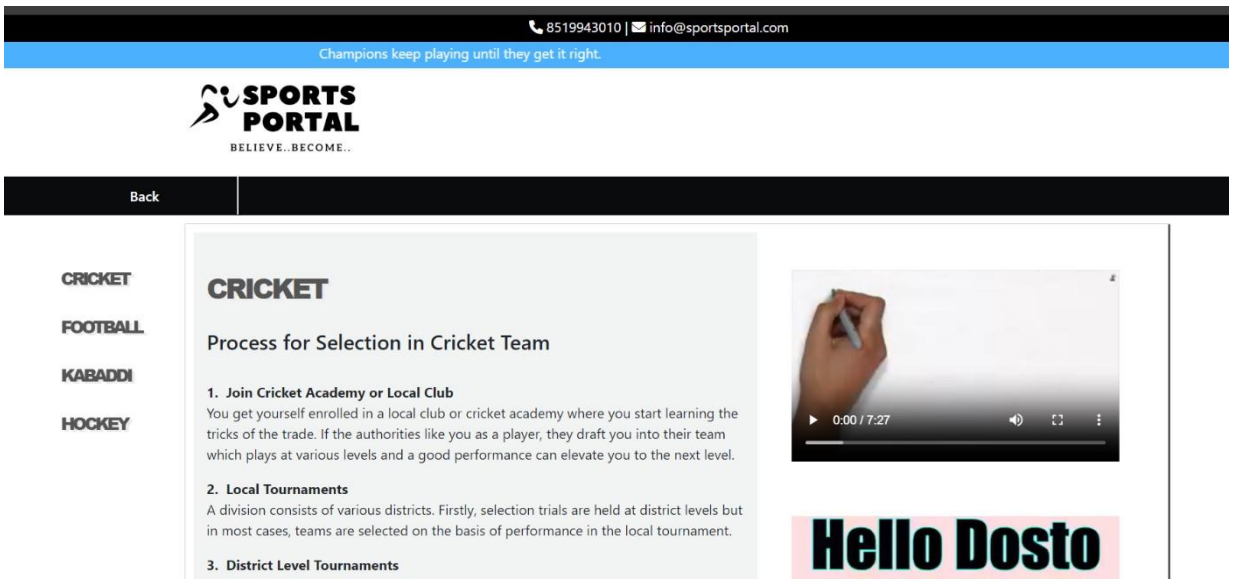

Fig 4.2.1.3 Selection Process

Selection Process : The selection process on The Sports Portal is streamlined and user-friendly. Athletes can browse detailed profiles of coaching centers, including location, specialties, and user ratings. This allows them to make informed decisions based on their specific training needs.

4.2.1.4 RECOVERY PREDICTION:

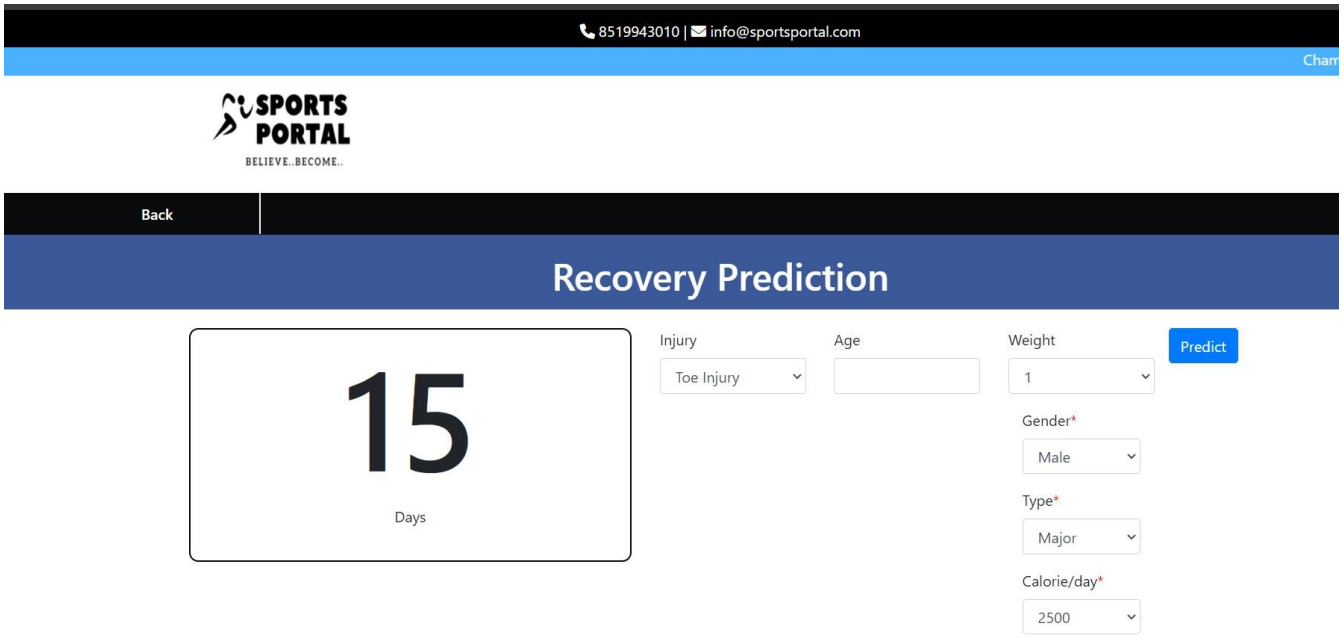

Fig 4.2.1.1.4 Recovery Prediction

Reduced Injury Rates: The Injury Prediction Module will lead to a significant reduction in injury rates by allowing athletes to proactively manage their health. By identifying potential risks and implementing preventative measures, athletes can avoid common injuries and extend their careers.

4.2.1.5 COACHING CENTER:

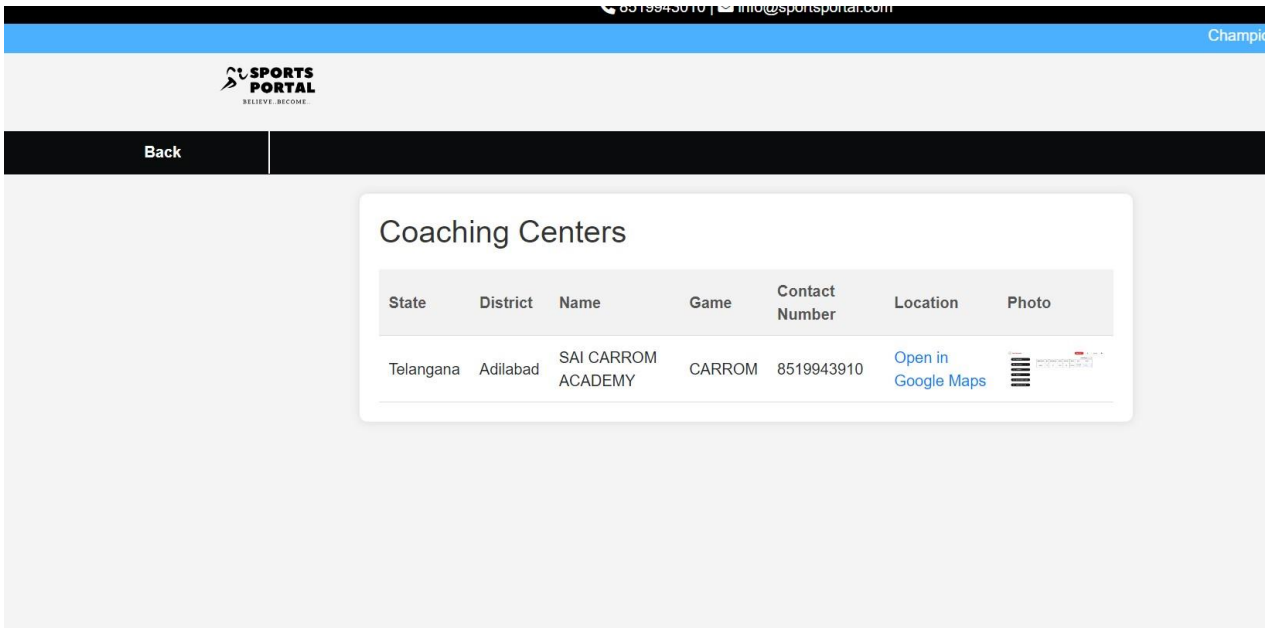

Fig 4.2.1.1.5 Coaching center

Improved Accessibility to Training Resources: The platform will simplify the process of finding nearby academies and coaching centers, providing athletes with easy access to quality training facilities. This will ensure that athletes can find the best possible training environments tailored to their needs.

4.2.1.6 EVENT CALENDAR

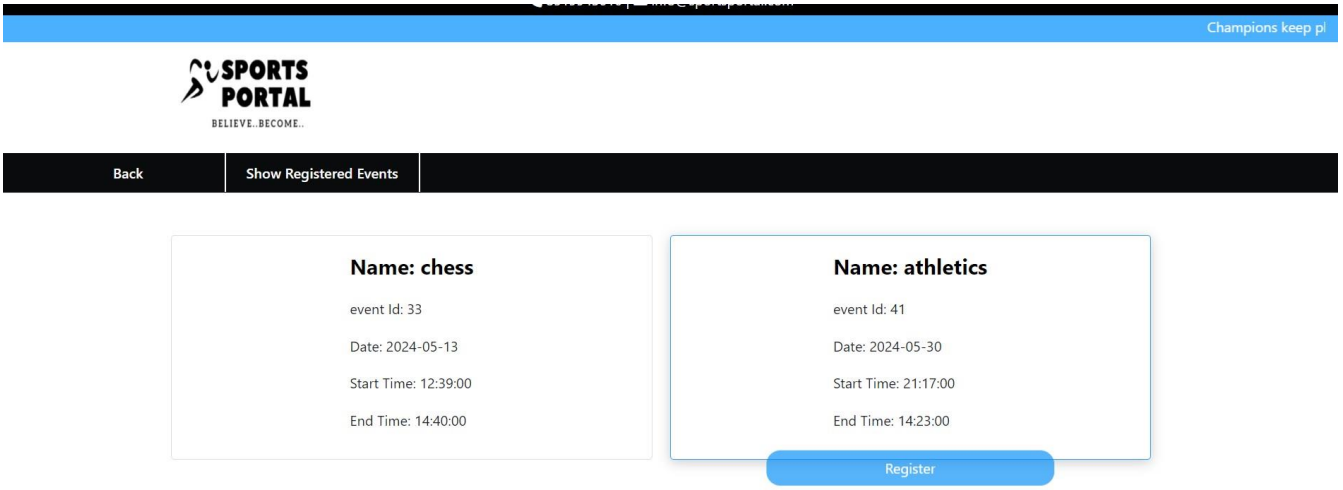

Fig 4.2.1.1.6 Event Calendar

Event Calendar : The Sports Portal features a user-friendly Event Calendar, helping athletes stay informed about upcoming competitions and training camps. Users can search by location, sport, and date, with real-time updates on schedules and registration deadlines. Reminders and notifications ensure athletes never miss an important event, keeping them organized and prepared.

4.2.1.7 USER REGISTERED EVENTS:

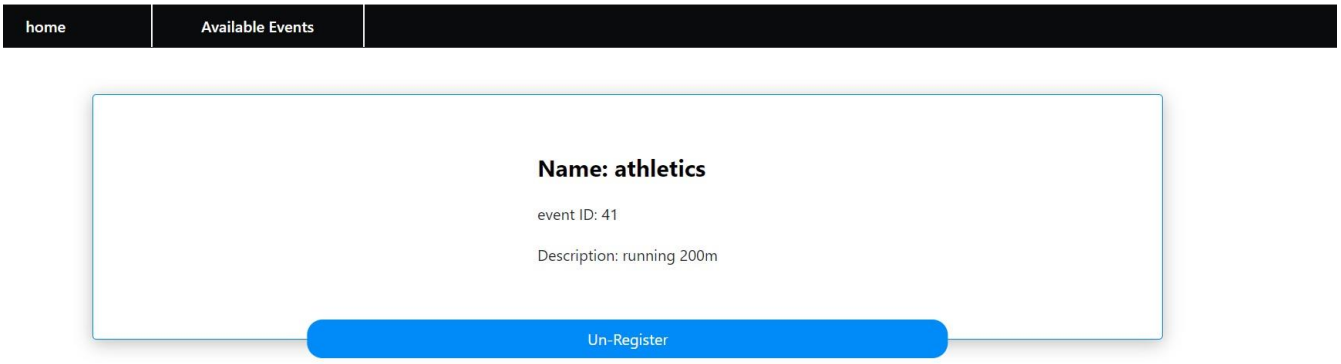

Fig 4.2.1.1.7 User Registered events

User Registered Events : The Sports Portal allows athletes to register for events directly. Users can easily sign up for competitions and training camps, then view and manage their events in one place. They receive reminders and notifications, ensuring they stay organized and never miss an important event.

4.2.1.8 ADD EVENT

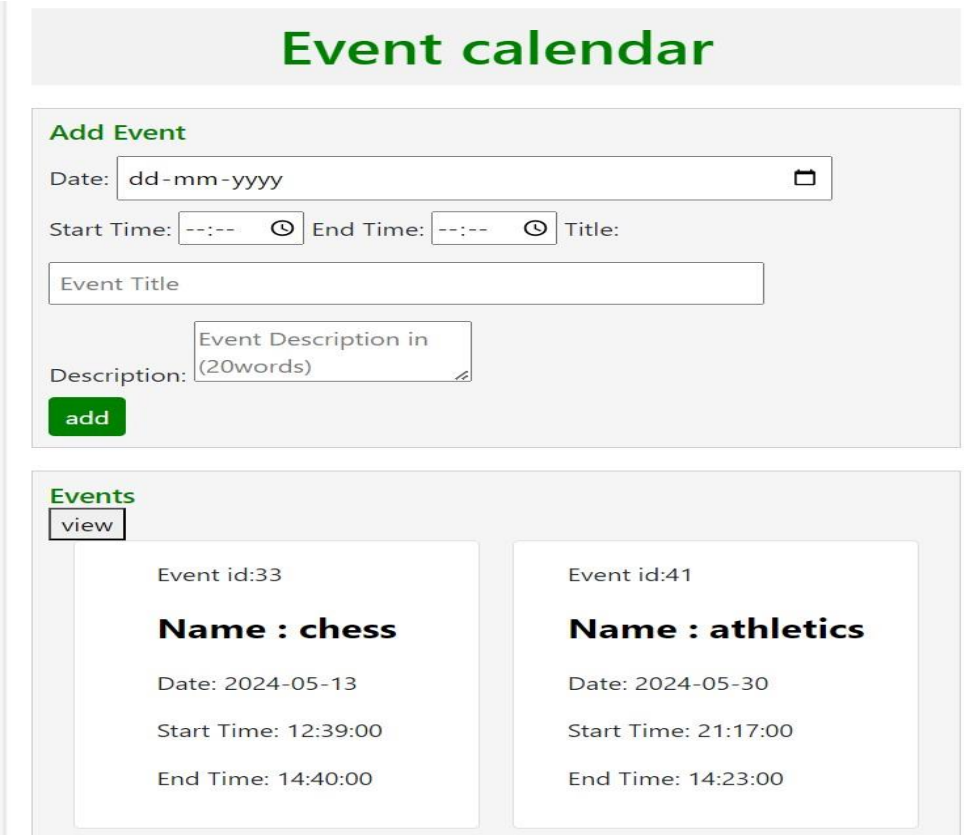

Fig 4.2.1.1.8 Add Event

Add Event : The Sports Portal also allows athletes to add events to their personal calendar. Users can easily sign up for competitions and training camps, then view and manage their events in one place. They receive reminders and notifications, ensuring they stay organized and never miss an important event.

4.2.1.9 ADD COACHING CENTER

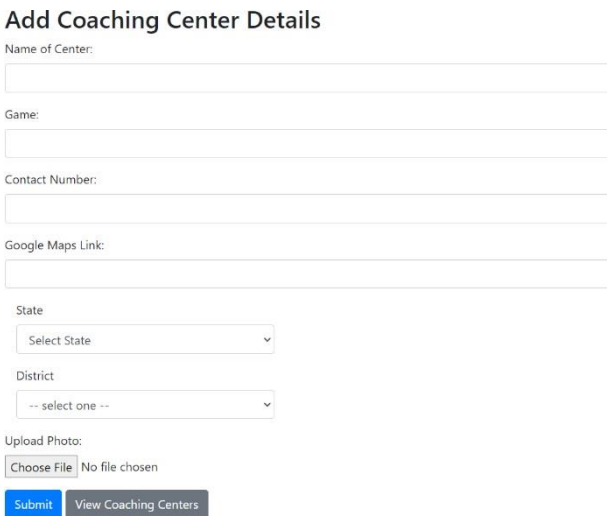

Fig 4.2.1.1.9 Add Coaching Center

Add Coaching center : The Sports Portal also features a directory of coaching centers, making it convenient for athletes to find and connect with quality coaching facilities. Users can search for centers based on location, sport, and specialization, ensuring they find the right coaching support. This integrated feature helps athletes access reliable training resources to enhance their skills and performance effectively.

4.2.2 PSEUDO CODE:

4.2.2.1 App.py:

```
from flask import Flask, flash, render_template, request, redirect, url_for, jsonify, session
import mysql.connector import logging,sys from recovery import train_linear_regression
from datetime import timedelta from werkzeug.utils import secure_filename import os


app = Flask(__name__) app.secret_key=
'123'

UPLOAD_FOLDER = 'static/uploads' app.config['UPLOAD_FOLDER']
= UPLOAD_FOLDER

# Database connection configuration
mydb = mysql.connector.connect(
host="localhost",     user="root",
password="Sairishik2019@",
database="rishik"
)

cursor = mydb.cursor()

@app.route('/') def
login():
   return render_template('login.html')


@app.route('/login', methods=['POST'])
def login_post():     if request.method
== 'POST':
```

```
    # Check if the logout button was clicked        if
request.form.get('logout') == 'logout':
session.clear()  # Clear the session data            return
redirect('/login')  # Redirect to the login page

    # Check if the signup button was clicked
signup_value = request.form.get('signup')        if
signup_value == 'signup':
        return redirect(url_for('signup'))

    # Proceed with login process
username = request.form['username']
password = request.form['password']
session['username'] = username        cursor
= mydb.cursor()
    cursor.execute("SELECT * FROM user WHERE username = %s AND password =
%s", (username, password))
user = cursor.fetchone()        if
user:
        # User exists, redirect to the appropriate page
if username == "admin@gmail.com":
return redirect(url_for('admin'))          else:
return redirect(url_for('home'))        else:
        # User doesn't exist or incorrect password, redirect back to login page
return redirect(url_for('login'))

@app.route('/logout1')
def logout():
   # Clear the session or perform any other logout logic
# For example:
   # session.clear()
   # Redirect the user to the login page
return redirect(url_for('login'))
```

```
@app.route('/add_coaching_center1', methods=['GET','POST']) def
add_coaching_center1():
   return render_template('coaching.html')

@app.route('/view_coaching_centers', methods=['GET']) def
view_coaching_centers():

   cursor = mydb.cursor()     cursor.execute("SELECT *
FROM coaching_centers")     coaching_centers =
cursor.fetchall()     cursor.close()



   return render_template('coachingadmin.html', coaching_centers=coaching_centers)
@app.route('/delete_coaching_center/<int:center_id>', methods=['POST'])
def delete_coaching_center(center_id):     if request.method == 'POST':
cursor = mydb.cursor()
      # Delete the coaching center from the database       cursor.execute("DELETE FROM
coaching_centers WHERE id = %s", (center_id,))       mydb.commit()       cursor.close()
return redirect(url_for('view_coaching_centers'))  # Redirect to the coaching centers page
else:
      return "Method Not Allowed"

@app.route('/add_coaching_center', methods=['GET', 'POST']) def
add_coaching_center():
   # Check if the user is logged in and is an administrator
# if not is_admin():
   #     # If not an admin, redirect to an appropriate page or show an error message
#     return "You are not authorized to access this page."

   if request.method == 'POST':        center_name =
request.form['center_name']        game =
request.form['game']        contact_number =
request.form['contact_number']        maps_link =
request.form['maps_link']        state =
```

```
request.form['inputState']        district =
request.form['inputDistrict']

      # Handle file upload
if 'photo' in request.files:
          photo = request.files['photo']
if photo.filename != '':                #
Save photo to a directory
              photo_filename = secure_filename(photo.filename)
photo.save(os.path.join(app.config['UPLOAD_FOLDER'], photo_filename))            else:
              photo_filename = ''  # No photo uploaded

      # Insert data into the database
cursor = mydb.cursor()
      cursor.execute("INSERT INTO coaching_centers (center_name, game,
contact_number, maps_link,photo_filename, state, district) VALUES (%s, %s, %s, %s,%s,
%s, %s)", (center_name, game, contact_number, maps_link,photo_filename, state, district))
mydb.commit()        cursor.close()

      return redirect(url_for('add_coaching_center1'))
   else:
      # Render the form for adding coaching centers
return render_template('add_coaching_center.html')




@app.route('/signup', methods=['GET', 'POST'])
def signup():     if request.method=='POST':
      signup_value = request.form.get('signup')
print(signup_value)        username =
request.form['username']        password =
request.form['password']
session['username'] = username
```

```
        if signup_value == 'signup':
            print("Successfully Signed Up")
# Insert new user into database
cursor = mydb.cursor()
            cursor.execute("INSERT INTO user (username, password) VALUES (%s, %s)",
(username, password))
mydb.commit()

            return redirect(url_for('login'))     return
render_template('signup.html')
@app.route('/admin', methods=['GET', 'POST'])
def admin():
    return render_template('admin.html')



@app.route('/home', methods=['GET', 'POST']) def
home():
    print("entered home")
sys.stdout.flush()     if request.method ==
'POST':         print("enetered post method
in home")         sel =
request.form.get('selection')
print("selection1")         if sel=='selection':
print("selection1")             return
redirect(url_for('selection'))         lgin =
request.form.get('loginl')         if
lgin=='loginl':             print("lgout")
return redirect(url_for('login'))
rec=request.form.get('recovery')         if
rec=='recovery':
            return redirect(url_for('recovery'))
```

```
        coach=request.form.get('coaching')
if coach=='coaching':
            return redirect(url_for('coaching'))

        eve=request.form.get('eventcal')
if coach=='eventcal':
            return redirect(url_for('eventcal'))



    return render_template('home.html')
@app.route('/selection', methods=['POST']) def
selection():
    return render_template('selection.html')
@app.route('/recovery', methods=['GET', 'POST']) def
recovery():
    return render_template('recovery.html') @app.route('/coaching', methods=['GET', 'POST'])
def coaching():     if request.method == 'POST':         # Handle form submission if needed
pass

    # Query the database to fetch all coaching center details     cursor =
mydb.cursor()     cursor.execute("SELECT * FROM coaching_centers")
coaching_centers = cursor.fetchall()     print(coaching_centers)     cursor.close()
return render_template('coachingdat.html', coaching_centers=coaching_centers)

@app.route('/add', methods=['POST'])
def add():     if request.method ==
'POST':
        print("entered post")         # Get the form data
eventName = request.form['eventTitle']         eventDate =
request.form['eventDate']         startTime =
request.form['eventStartTime']          endTime =
request.form['eventEndTime']         eventDescription =
request.form['eventDescription']          cursor =
mydb.cursor()
```

```
        # Insert the data into the database
        cursor.execute("INSERT INTO events (eventName, eventDate, startTime, endTime,
eventDescription) VALUES (%s, %s, %s, %s, %s)", (eventName, eventDate, startTime,
endTime, eventDescription))

        # Commit the changes
mydb.commit()     return
render_template('admin.html')
@app.route('/login22',
methods=['GET', 'POST']) def
login22():
   return render_template('login.html')
@app.route('/submit_data', methods=['GET', 'POST'])
def predit():     result=0     if
request.method=='POST':
        data=request.form
print(data)         calorie =
data['calorie']         l=[]
        l.append(int(data['calorie']))
        l.append(int(data['age']))
        l.append(int(data['weight']))
        l.append(int(data['injury']))
        l.append(int(data['gender']))
        l.append(int(data['type']))
print(l)
        # cursor = mydb.cursor()
        # cursor.execute("select * from injurytable")
        # dataset=cursor.fetchall()         # mydb.commit()         result =
train_linear_regression("newinjury.csv", "Recovery_Period",
new_data_point=l)
        print(result)         return
render_template('recovery.html',result=result)     return
render_template('recovery.html',result=result)
```

```
@app.route('/eventcalendar') def
user_events():
   cursor = mydb.cursor()     cursor.execute("SELECT * FROM events
ORDER BY eventDate asc")     events = cursor.fetchall()     event_list = []
for event in events:         event_dict = {
          'id': event[0],
          'eventDate': str(event[1]),  # Convert timedelta to string
'startTime': str(event[2]),   # Convert timedelta to string
          'endTime': str(event[3]),    # Convert time to string
          'eventName': event[4],
          'eventDescription': event[5]
       }
       event_list.append(event_dict)     return
render_template('events.html', events=event_list)

@app.route('/adminviews', methods=['GET', 'POST']) def
admin_events():
   cursor = mydb.cursor()     cursor.execute("SELECT * FROM events
ORDER BY eventDate asc")     events = cursor.fetchall()     event_list = []
for event in events:         event_dict = {
          'id': event[0],
          'eventDate': str(event[1]),  # Convert timedelta to string
'startTime': str(event[2]),   # Convert timedelta to string
          'endTime': str(event[3]),    # Convert time to string
          'eventName': event[4],
          'eventDescription': event[5]
       }
       event_list.append(event_dict)         print(event_list)
return render_template('admin.html', events=event_list)
@app.route('/adminviews/delete_event/<int:event_id>',
methods=['POST']) def delete_event(event_id):     try:
       cursor = mydb.cursor()
print(event_id)
```

```
        cursor.execute("DELETE FROM events WHERE eventId = %s", (event_id,))
mydb.commit()         return jsonify({'message': 'Event deleted successfully'}), 200
except Exception as e:
        return jsonify({'error': str(e)}), 500



@app.route('/register') def
register():
   return render_template('form.html')




@app.route('/register_event/<int:event_id>', methods=['POST'])
def register_event(event_id):     if 'username' in session:
        username = session['username']
print("Username from session:", username)
print(event_id)         try:
            cursor = mydb.cursor()
            # Check if the user is already registered for the event
cursor.execute("SELECT * FROM registrations WHERE username = %s AND
event_id = %s", (username, event_id))             registration_exists = cursor.fetchone()
if registration_exists:
                flash('You are already registered for this event!', 'warning')
            else:
                # Fetch game and description associated with the event_id
cursor.execute("SELECT eventName, eventDescription FROM events WHERE eventId =
%s", (event_id,))                 event_data = cursor.fetchone()                 if event_data:
                    game = event_data[0]
description = event_data[1]
                    # Insert registration data into the registrations table
cursor.execute("INSERT INTO registrations (username, event_id, game, description)
VALUES (%s, %s, %s, %s)",
```

```
                        (username, event_id, game, description))
mydb.commit()  # Commit the transaction
flash('Event registered successfully!', 'success')              else:
                flash('Event not found!', 'error')
except mysql.connector.Error as err:
# Handle any database errors
flash(f"Database error: {err}", 'error')
finally:
        cursor.close()  # Close the cursor
else:
    flash('Please login to register for events.', 'error')
return redirect(url_for('user_events'))

####
@app.route('/unregister_event/<int:event_id>', methods=['POST']) def
unregister_event(event_id):
  if 'username' in session:
    username = session['username']
print(username)      print(event_id)
try:
        cursor = mydb.cursor()
        # Delete the registration record for the logged-in user and the specific event
cursor.execute("DELETE FROM registrations WHERE username = %s AND
event_id = %s", (username, event_id))          mydb.commit()  # Commit the
transaction          flash('Successfully unregistered from the event!', 'success')
except mysql.connector.Error as err:          # Handle any database errors
flash(f"Database error: {err}", 'error')      finally:
        cursor.close()  # Close the cursor
else:
    flash('Please login to unregister from events.', 'error')
return redirect(url_for('show_registered_events'))
```

```
####
@app.route('/Show_Registered_Events', methods=['GET','POST'])
def show_registered_events():     if 'username' in session:
      username = session['username']
try:
         cursor = mydb.cursor()
         # Fetch registered events for the logged-in user            cursor.execute("SELECT
game, description,event_id FROM registrations WHERE username = %s", (username,))
         registered_events = cursor.fetchall()
print(registered_events)           reglist=[]
for reg in registered_events:
            reg_dict={
               'name':str(reg[0]),
               'desc':str(reg[1]),
               'event_id':str(reg[2])
            }
            reglist.append(reg_dict)               print(reglist)           # Close cursor
cursor.close()           return render_template('registered_events.html',
registered_events=reglist)        except mysql.connector.Error as err:
flash(f"Database error: {err}", 'error')     else:
      flash('Please login to view registered events.', 'error')
return redirect(url_for('login'))
```

```
@app.route('/eventcal', methods=['GET','POST']) def
add_event():
   if request.method == 'POST':
try:
```

```
        print("DATABASE EVENT")          # Extract event
data from request          event_data = request.json
event_date = event_data.get('event_date')          event_name
= event_data.get('event_name')          event_time_from =
event_data.get('event_time_from')          event_time_to =
event_data.get('event_time_to')

        print("Event date",event_date)
print(event_name)          print(event_time_from)
print(event_time_to)

        # Insert event into database          cursor = mydb.cursor()
print("Connection Successful")          cursor.execute("INSERT INTO events (event_date,
event_name, event_time_from, event_time_to) VALUES (%s, %s, %s, %s)",
                    (event_date, event_name, event_time_from, event_time_to))
mydb.commit()          print("Data Saved")          cursor.close()
return jsonify({"success": True}), 200        except Exception as e:
        return jsonify({"error": str(e)}), 500

   return render_template('event.html')

# Configure Flask logger to output to terminal app.logger.setLevel(logging.DEBUG)

@app.route('/') def
index():
   app.logger.debug('This is a debug message')
app.logger.info('This is an info message')
app.logger.warning('This is a warning message')
app.logger.error('This is an error message')
app.logger.critical('This is a critical message')
return 'Check terminal for log messages' if
__name__ == '__main__':     app.run(debug=True)
```

4.2.2.2 HOME PAGE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"     integrity="sha384-
</head>

<body>
  <section class="top-area">
    <div class="header-area navigation">
      <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
          <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse"
datatarget="#navbar-menu">
              <i class="fa fa-bars"></i>
            </button>
            <a class="navbar-brand" href="home.html">
              <img src="static\images\sportsportal.png" alt="Sports Portal Logo" style="width:
25%; height: 25%;">
            </a>
          </div>

          <div class="collapse navbar-collapse d-flex flex-row justify-content-end"
id="navbarNav"            style="font-family: 'Times New Roman', Times, serif">
```

```
        <ul class="navbar-nav" style="font-size: 20px">
          <li class="scroll" style="margin-right: 10px">
            <form id="selectionForm" action="/selection" method="POST" class="form">
              <input type="hidden" name="selection" value="selection">
              <button type="submit" class="nav-link" id="selection">Selection
Process</button>
            </form>

            <script>
              document.getElementById("selection").addEventListener("click", function (event)
{
                // No need for event.preventDefault() as the button type is "submit"
console.log("Selection button clicked!"); // Log message to the console
              });
            </script>

          </li>

          <script>
document.getElementById("selection").addEventListener("click", function (event)
{              event.preventDefault(); // Prevent default behavior of the anchor tag
document.getElementById("selectionForm").submit(); // Submit the form
            });
          </script>


          <li class="scroll" style="margin-right: 10px">
            <form id="recoveryForm" action="/recovery" method="GET" class="form">
              <input type="hidden" name="recovery" value="recovery">
            </form>
            <a class="nav-link" id="recovery" href="#">Recovery Prediction</a>
          </li>

          <script>
```

```
        document.getElementById("recovery").addEventListener("click", function (event)
{           event.preventDefault(); // Prevent default behavior of the anchor tag
document.getElementById("recoveryForm").submit(); // Submit the form
        });
      </script>



      <li class="scroll" style="margin-right: 10px">
        <form id="coachingForm" action="/coaching" method="GET" class="form">
          <input type="hidden" name="coaching" value="coaching">
        </form>
        <a class="nav-link" id="coaching" href="#">Coaching Centre</a>
      </li>

      <script>
        document.getElementById("coaching").addEventListener("click", function (event)
{           event.preventDefault(); // Prevent default behavior of the anchor tag
document.getElementById("coachingForm").submit(); // Submit the form
        });
      </script>

      <li class="scroll" style="margin-right: 10px">
        <form id="eventForm" action="/eventcalendar" method="GET" class="form">
          <input type="hidden" name="eventcalendar" value="eventcalendar">
        </form>
        <a class="nav-link" id="eventcalendar" href="#">Event Calendar</a>
      </li>

      <script>
        document.getElementById("eventcalendar").addEventListener("click", function
```

```
(event) {                event.preventDefault(); // Prevent default behavior of the
anchor tag                 document.getElementById("eventForm").submit(); //
Submit the form
            });
          </script>

          <form class="form" action="/login22" method="post">
            <li class="scroll" style="margin-right: 10px">
              <button class="Btn" type="submit" name="loginl" value="login">
                Logout
              </button>
            </li>
          </form>
          <form class="form" action="/similar" method="post">
          </form>
        </ul>
      </div>
    </div>
  </nav>
 </div>
</section>
<section id="Home" class="welcome-hero">
  <video autoplay muted loop id="bg-video">
    <source src="static\video\homepagevdo.mp4" type="video/mp4">
    Your browser does not support HTML5 video.
  </video>
  <div class="container">
<div class="welcome-hero-txt">
      <h2>
        "Focus on the Process Not the Results..." <br />
        <!-- And if you take care of the process, you will get the results.." -->
      </h2>
      <p></p>
    </div>
```

```
  </div>
 </section>
 <br>
<section id="works" class="works" style="padding: 20px">
  <div class="container">
   <div class="section-header" align="center">
    <h2 style="color: #4a646b">About</h2>
    <p>Lets get a info about how,what and why..?</p>
   </div>
   <!--/.section-header-->
   <div class="works-content">
    <div class="row">
     <div class="col-md-4 col-sm-6">
      <div class="single-how-works">
       <img src="static\images\injurytabhome.png" alt="Recovery picture" class="src" />

       <h2>
        <a href="#">injury <span> Recovery</span> Prediction</a>
       </h2>
       <br>
       <p>
        Experience streamlined injury recovery prediction on our platform! Simply select injury details including type, age, and gender, then click predict. Our tool calculates estimated recovery time, making your decision-making effortless. Get back in the game faster with our precise injury recovery predictions. Try it now!.
       </p>
      </div>
     </div>
     <div class="col-md-4 col-sm-6">
      <div class="single-how-works">
       <img src="static\images\homeselectionprocess.png" alt="selection process" class="src" />
       <h2>
        <a href="#">Selection <span> process</span></a>
```

```
            </h2><br>
<p>
                Discover the secrets behind sporting success with our Selection Process feature! Explore how teams and athletes are chosen for the game. Uncover the strategies and criteria that shape sporting excellence.
                Get ready to dive deep into the heart of competition on our sports website!
              </p>
            </div>
          </div>
          <div class="col-md-4 col-sm-6">
            <div class="single-how-works">
              <img src="static\images\calendar.jpg" alt="event Calendar" class="src" />
<h2>
                <a href="#">Event <span> Calendar</span> </a>
              </h2>
              <br>
              <p>
                Stay in the loop with our Event Calendar feature! Explore a comprehensive lineup of upcoming events, from tournaments to championships. Never miss a beat with our user-friendly calendar, keeping you informed and engaged. Whether you're a player, fan, or enthusiast, our calendar has something for everyone. Dive in and discover!
              </p>
            </div>
          </div>


        </div>
      </div>
    </div>
    <!--/.container-->
  </section>
<br>
  <section id="contact" style="font-family: 'Times New Roman', Times, serif; padding-top: 20px">
    <div class="contact-top" align="center">
```

```
    <h1 style="font-weight: bold; color:#1c2123">Connect With Us</h1>
    <p>We'd Love to Hear from You.....</p>
   </div>
   <div class="container">
    <div class="row">
     <div class="col-4">
      <div align="center" class="d-flex flex-row" style="padding: 100px; background-color:
#344044">
       <div>
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24"
height="24">
         <path
          d="M18 2H6a2 2 0 0 0-2 2v16a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V4a2 2 0 0 0-2-
2zM9 20H7v-2h2zm4 0h-2v-2h2zm4 0h-2v-2h2zm2-4H5V6h14z" />
        </svg>
       </div>
       <div>
        <h2 style="color: #ffffff">Call</h2>
        <p style="color: #f0f0f0;">+91 8519943987</p>
       </div>
      </div>
     </div>
     <div class="col-4">
      <div align="center" class="d-flex flex-row" style="padding: 100px; background-color:
#0ea4d2">
<div>
       <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24"
height="24">
         <path
          d="M18 2H6a2 2 0 0 0-2 2v16a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V4a2 2 0 0 0-2-
2zM9 20H7v-2h2zm4 0h-2v-2h2zm4 0h-2v-2h2zm2-4H5V6h14z" />
        </svg>
       </div>
       <div>
```

```
            <h2 style="color: #ffffff">Instagram</h2>
            <p><a href="www.instagram.com">Instagram</a></p>
           </div>
          </div>
        </div>
        <div class="col-4">
          <div align="center" class="d-flex flex-row" style="padding: 100px; background-color:
#eeb32b">
            <div>
              <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24"
height="24">
                <path
                  d="M18 2H6a2 2 0 0 0-2 2v16a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V4a2 2 0 0 0-2-
2zM9 20H7v-2h2zm4 0h-2v-2h2zm4 0h-2v-2h2zm2-4H5V6h14z" />
              </svg>
            </div>
            <div>
              <h2 style="color: #ffffff">Email</h2>
              <p>sportsportal@gmail.com</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>


</body>

</html>
```

5. EXPERIMENTAL SETUP & IMPLEMENTATION

5.1 System Specifications

5.1.1 Hardware Requirements

Processor: A multi-core processor, preferably an Intel Core i7 or AMD Ryzen 7, with a minimum clock speed of 3.0 GHz. The processor should support parallel processing capabilities to handle computationally intensive tasks efficiently.

Memory (RAM): At least 16 GB of DDR4 RAM is required to ensure smooth execution of programs and to accommodate large datasets in memory without causing significant performance degradation. For more complex experiments, 32 GB or more may be necessary.

Storage: A Solid-State Drive (SSD) with a minimum capacity of 512 GB is recommended for fast data access and reduced I/O latency. An additional 1 TB HDD can be used for long-term data storage and backups.

Graphics Processing Unit (GPU): A dedicated GPU, such as an NVIDIA GeForce GTX 1080 or higher, with at least 8 GB of VRAM is essential for tasks involving deep learning and other GPU-accelerated computations.

Power Supply: A reliable power supply unit (PSU) with sufficient wattage to support all the hardware components, ensuring stable and uninterrupted operation of the system.

5.1.2 Software Requirements

Operating System:

A stable and widely supported operating system such as Ubuntu 20.04 LTS (or later) or Windows 10 Professional.

Programming Languages:

JavaScript: For client-side scripting and dynamic content.

HTML: For structuring the web pages.

CSS: For styling the web pages.

Python: Python 3.8 (or later) for backend development with Flask.

Development Environment:

Visual Studio Code (VS Code): A powerful and flexible IDE with support for a wide range of extensions, enabling development in JavaScript, HTML, CSS, and Python.

Frameworks and Libraries:

Flask: A lightweight Python web framework for building web applications.

Bootstrap: A CSS framework for responsive and mobile-first web development.

Jinja2: A templating engine for Flask to create dynamic web pages.

MySQL-Connector-Python: A Python library for connecting to and interacting with MySQL databases.

Database:

MySQL: For structured data storage and management. MySQL Workbench can be used for database design and administration.

5.2 DATASETS

Callore	Age	Weight	Fitness Level	Injury	Gender	Type	Recovery_Perd
2500	28	65	0.8	1	1	1	2
2400	31	70	0.7	1	1	1	2
3300	29	67	0.8	1	1	1	2
2500	30	68	0.8	1	1	1	3
2400	25	55	0.7	1	1	2	6
2300	28	60	0.5	1	2	2	8
2300	21	50	0.6	1	2	1	2
2300	29	65	0.6	1	2	2	5
2350	32	60	0.6	1	1	2	6
3200	29	54	0.8	1	1	1	4
2350	26	65	0.7	2	1	2	3
2400	29	70	0.8	2	1	2	12
2000	32	67	0.6	2	2	1	2
2400	29	68	0.6	2	2	1	2
2200	32	55	0.8	2	1	1	3
2550	28	70	0.7	2	1	2	10
3000	29	68	0.8	2	1	2	20
2500	30	67	0.7	2	1	2	16
2400	34	50	0.6	2	2	1	3
2300	27	65	0.8	2	1	1	2
2500	27	60	0.6	2	1	1	2
2300	33	65	0.7	2	1	2	3
2500	29	65	0.6	3	2	1	2
2400	30	70	0.5	3	2	2	8
2300	34	67	0.7	3	2	1	1
2500	27	68	0.5	3	2	1	2
2400	27	55	0.7	3	1	2	6
2200	27	58	0.8	3	1	2	6
2550	33	72	0.7	3	1	1	1

6. CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION

The Sports Prodigy Portal represents a significant advancement in the realm of sports training and athlete development by addressing critical gaps in the current system. By providing a centralized, integrated platform, it offers athletes comprehensive support that enhances performance, prevents injuries, and simplifies access to quality training resources. The innovative Injury Prediction Module empowers athletes to proactively manage their health, while the Nearest Academy Locator and Coaching Center Directory ensure they can easily find and access top-tier training facilities and expert coaching. Furthermore, the extensive self-learning resources enable athletes to independently refine their skills and knowledge. Ultimately, the Sports Prodigy Portal is designed to foster continuous improvement and safeguard the well-being of athletes, helping them achieve excellence in their chosen sports.

6.2 FUTURE SCOPE

Enhanced Predictive Analytics: Continuously improve the accuracy and sophistication of the Injury Prediction Module by incorporating machine learning algorithms and larger datasets.

Virtual Coaching and Training: Integrate virtual reality (VR) and augmented reality (AR) technologies to offer immersive training experiences and remote coaching sessions.

AI-Powered Personalization: Implement artificial intelligence (AI) to create highly personalized training programs based on individual athlete profiles, including their performance data, injury history, and specific goals. This can optimize training efficiency and effectiveness.

Expanded Resource Library: Continuously update and expand the self-learning resources with the latest research, training techniques, and best practices across various sports disciplines. Including interactive content, video tutorials, and webinars from industry experts can further enrich the learning experience

7. REFERENCES

- Kellmann, M., Bertollo, M., Bosquet, L., Brink, M., Coutts, A. J., Duffield, R., ... & Robazza, C. (2018). Recovery and performance in sport: consensus statement. International Journal of Sports Physiology and Performance.

- Baca, A., Pansi, W., & Kornfeind, P. (2018). Wearable technology in team sports. Advances in Intelligent Systems and Computing, 778, *203-213.*

- Claudino, J. G., Capanema, D. O., de Souza, T. V., Serrão, J. C., Machado Pereira, A. C., & Nassis, G. P. (2019). Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. Sports Medicine.

- Baker, J., Schorer, J., & Wattie, N. (2017). Compromising talent: Issues in identifying and selecting talent in sport..

- Gabbett, T. J., & Ullah, S. (2012). Relationship between running loads and soft-tissue injury in elite team sport athletes. Journal of Strength and Conditioning Research.

- Figueiredo, P., Seifert, L., & Vilas-Boas, J. P. (2014). Individual profiles of spatio-temporal coordination in swimming

- Joyce, D., & Lewindon, D. (2014). *High-performance training for sports.* Human Kinetics.

- Haugen, T. A., Tønnessen, E., & Seiler, S. (2015). Physical and physiological characteristics of male handball players: Influence of playing position and competitive level. Journal of Sports Medicine and Physical Fitness