1.INTRODUCTION

1.Introduction

This Sports Intermediary application is your go-to place for organizing and joining sports tournaments. It's like the middleman between people who run tournaments and those who love playing indoor and outdoor games. We're here to make sports events super easy and fun for everyone involved**.**

1.1 Background Of This Project:

Usually, planning tournaments means dealing with lots of paperwork and keeping track of who's playing against who. This Application takes care of all that. It's like your one-stop-shop where tournament organizers can easily manage their events, and players can find games to join.

1.2 Proposed System:

1.2.1 Sports Intermediary Application:

This provides a modern and user-friendly interface to address the limitations of the existing system, offering a range of features to enhance the overall experience of sports events organization and participation.

1.2.2 Advantage of Proposed System:

- The platform streamlines the process of organizing events, offering organizers a centralized hub to create, manage, and promote matches or tournaments seamlessly.
- Simplifies the registration process for players, offering an intuitive and user-friend interface.
- Implements secure registration processes and payment gateways to protect participant information and ensure safe financial transactions.
- Provides robust communication channels, including real-time messaging and announcements, fostering effective communication between organizers and players. This ensures participants stay informed about event details and updates promptly.

1.3 Existing System:

1.3.1 TeamSnap:

TeamSnap offers features for organizing and managing sports leagues and tournaments.

1.3.2 Disadvantages of Existing System:

- While TeamSnap is versatile and supports various sports, its primary focus is on team management.
- User experience can be subjective, and some users might find that the focus on team management in TeamSnap results in a user –interface that may not be as intuitive for tournament-specific tasks.
- Depending on the scale of tournaments and the number of participants, TeamSnap might face limitations related to scalability when handling larger-scale events.
- While TeamSnap provides communication features for teams, its focus might be on internal team communication rather than broader communication features needed for tournament-wide announcements and updates.

2.SOFTWARE REQUIREMENTS SPECIFICATION

2.SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Introduction:

2.1.1 Purpose

The purpose of this document to describe the detailed requirement specification for “Sports intermediate System” it will explain the purpose and features of system. The interface of the system, what the system do. The constraints under which it must operate and how the system will react to external stimuli.

2.1.2 Document Conventions

Font: Times New Roman

Font Style: Bold

Heading: 14 Sizes

Subheading: 12 sizes

Description: 12 sizes

Line spacing: 1.5

2.2 Overall Description:

2.2.1 Product Perspective:

The application allows authorized admin, representative, customer to access the data such as viewing tournament detail and conductor also viewing the existing Tournament detail. The output will consist of the result based on the operation performed.

2.2.2 Product Function:

The function of this project is that it produces result accurately, The Conductor can conduct tournament, view tournament and players can view tournament, search games , register tournament and post comments those information displayed in the screen.

2.2.3 User Classes and Characteristics:

Admin:

Role	Permission
	Insert Record
	Login Record
	View Record
	Delete Record
	Edit Record

Tournament Conductors:

Role	Permission
	View Record
	Edit Record
	Insert Record
	Login Record

Players:

Role	Permission
	View Record
	Login Record
	Insert Record

2.3 System Features:

This Sports Intermediary application offer seamless tournament management, including registration, and secure payment processing, all accessible through a user-friendly interface with robust customer support.

2.3.1 Functional Requirements:

FR No.	Functional requirement	Description
1	User Registration	Registration through Form
2	User Confirmation	Confirmation via Email
3	User Login	Login through Form
4	Admin Login	Login with username and password
5	Users Complaint	Enter the complaints

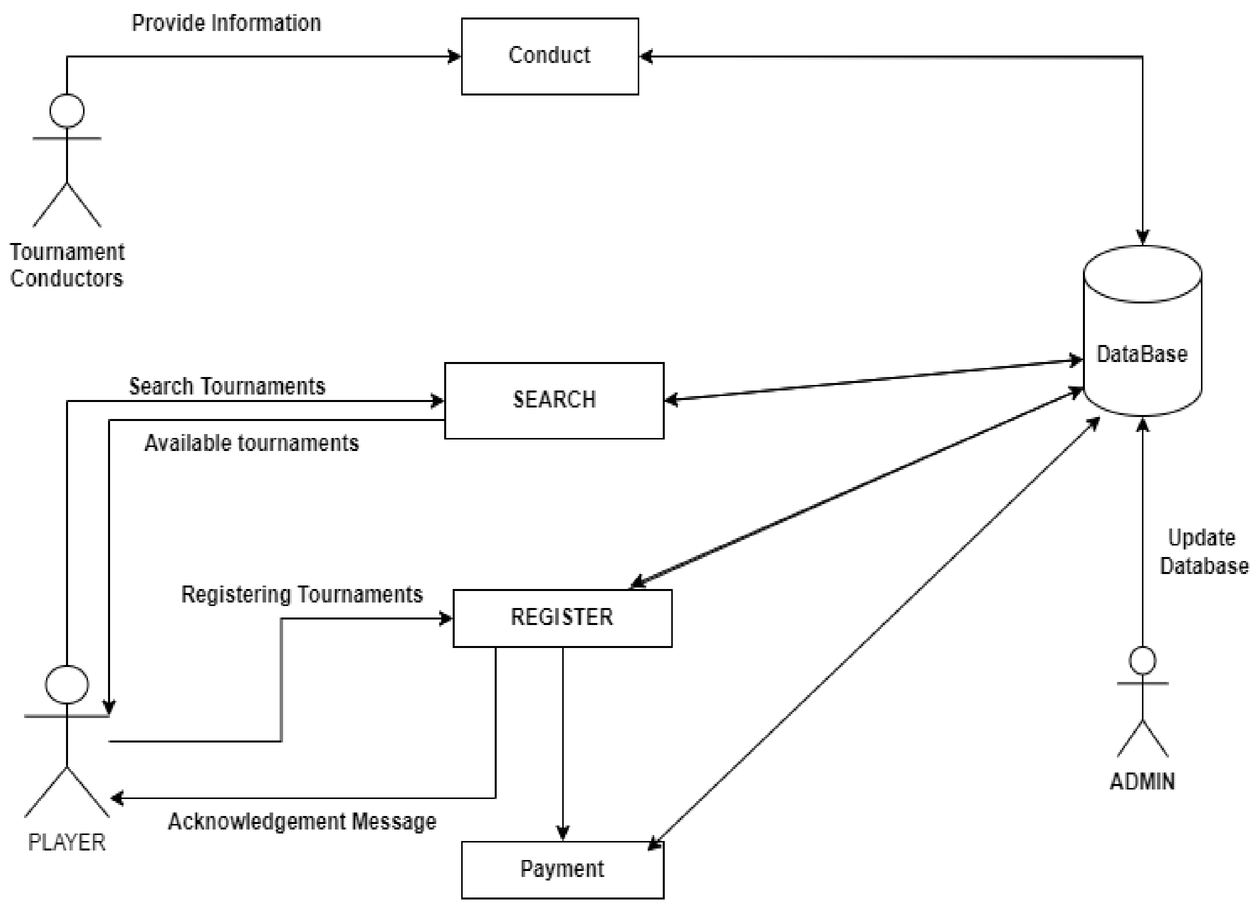
2.3.2 Non-Functional Requirements:

S. No	Non-Functional Requirements	Description
1	Usability	User can easily interact with the website
2	Security	The information given by the user will be secure
3	Reliability	The system will allow the user to contact the agent if the user didn’t get the satisfied solution
4	Performance	The user interface page will be loaded within few seconds
5	Availability	New module deployment must not impact front page and main page.
6	Scalability	The website traffic limit must be scalable.

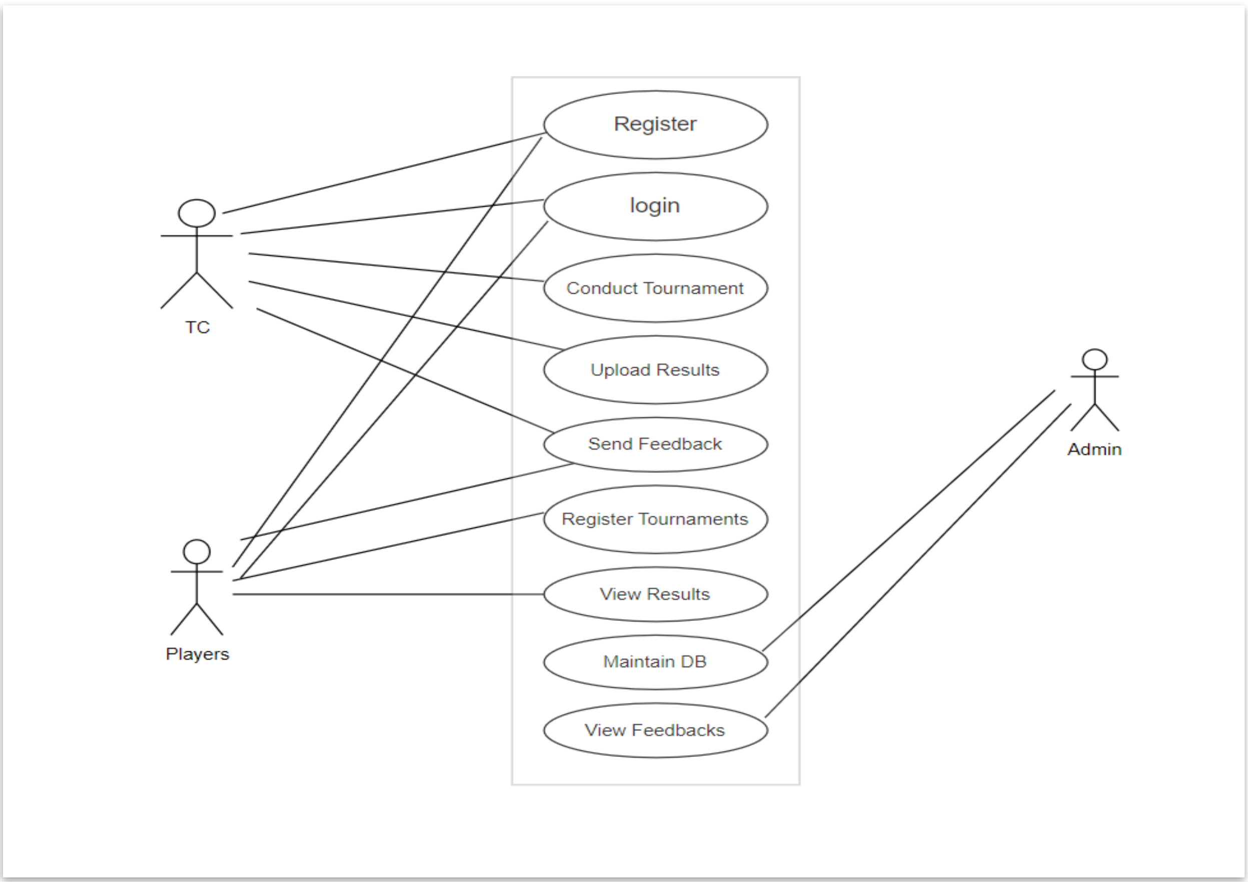
3.SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

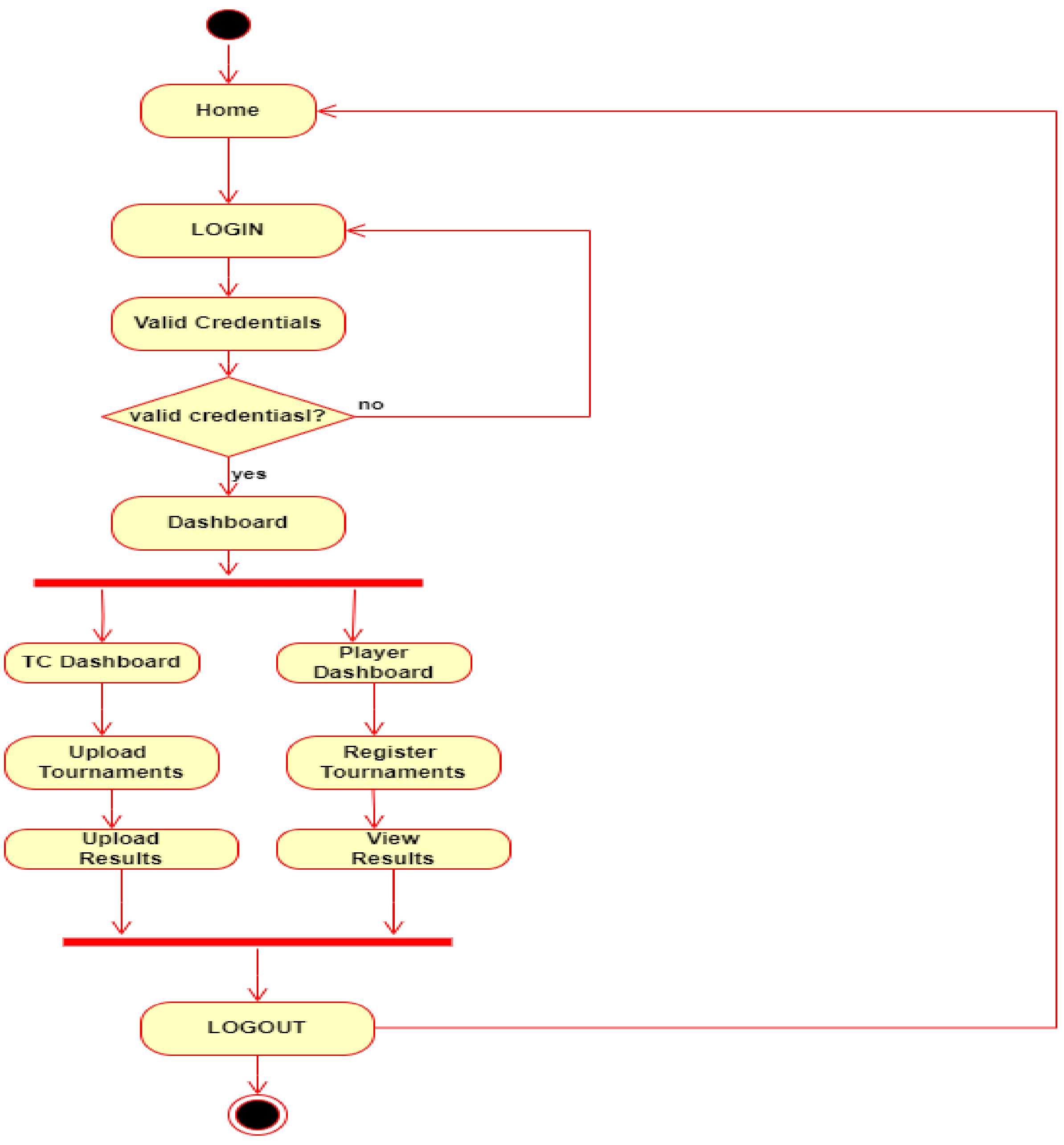
3.1 Architecture Diagram



3.2 Use Case Diagram



3.3 Activity Diagram



3.4 Modules and Description:

3.4.1Admin Module:

Responsible for managing user accounts, permissions, and overall system settings.

3.4.2 Conductor Module:

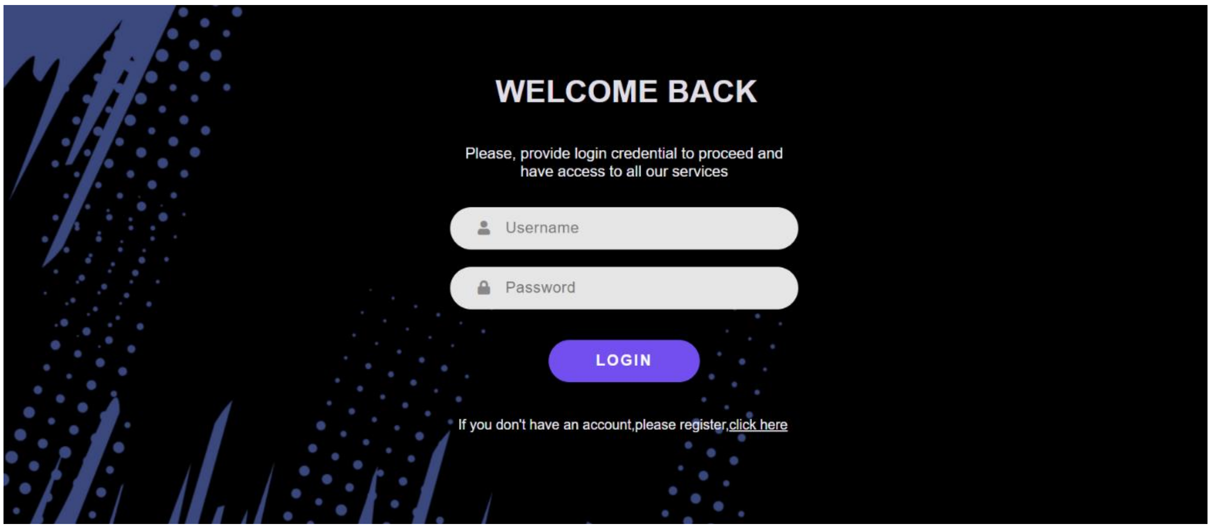
Handles the organization and management of sports tournaments, including creating tournaments, scheduling matches, and managing participant registrations.

3.4.3 player Module:

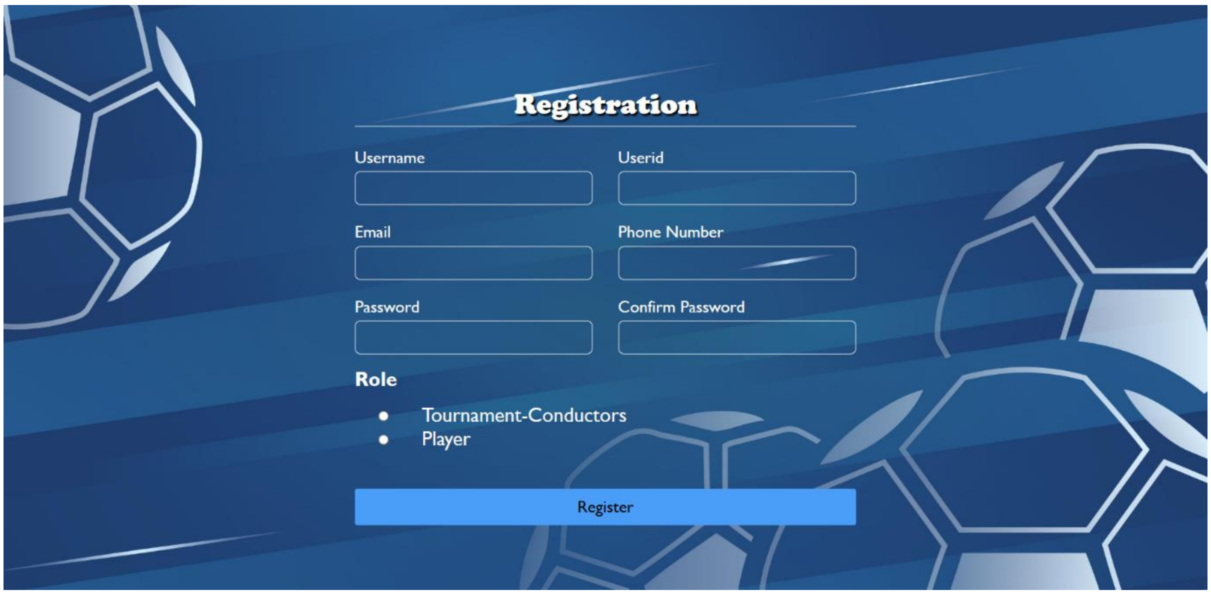
Provides features for players to register for tournaments, view match schedules and results, and communicate with tournament conductors.

4.Design

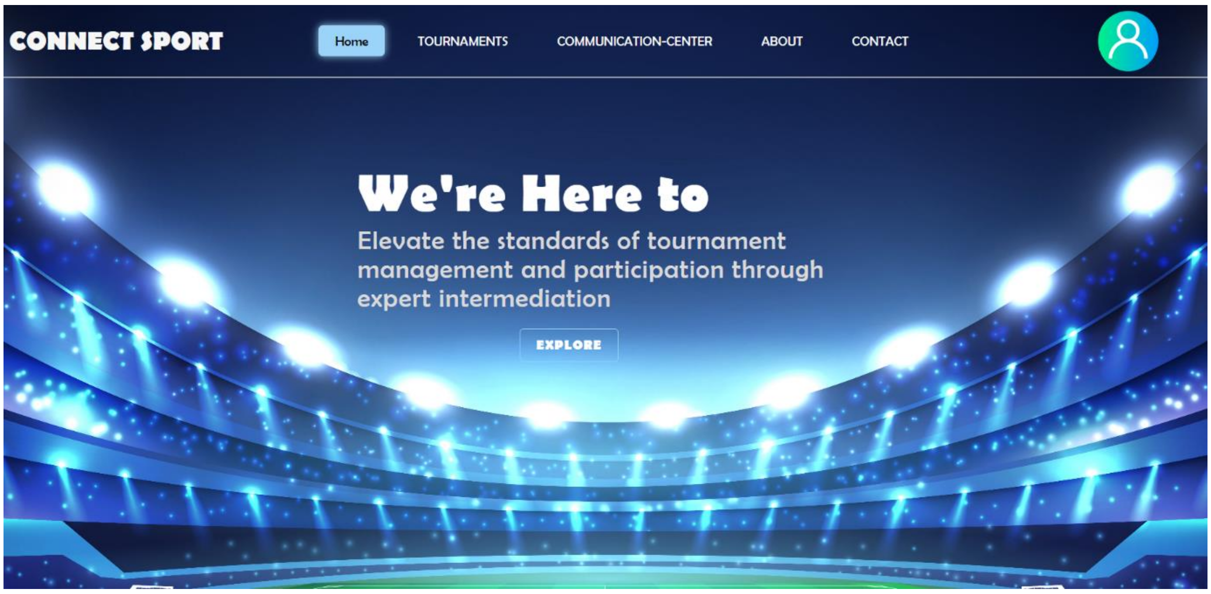
Login Page:


WELCOME BACK
Please, provide login credential to proceed and have access to all our services
Username
Password
LOGIN
If you don't have an account,please register,click here

Registration Page:



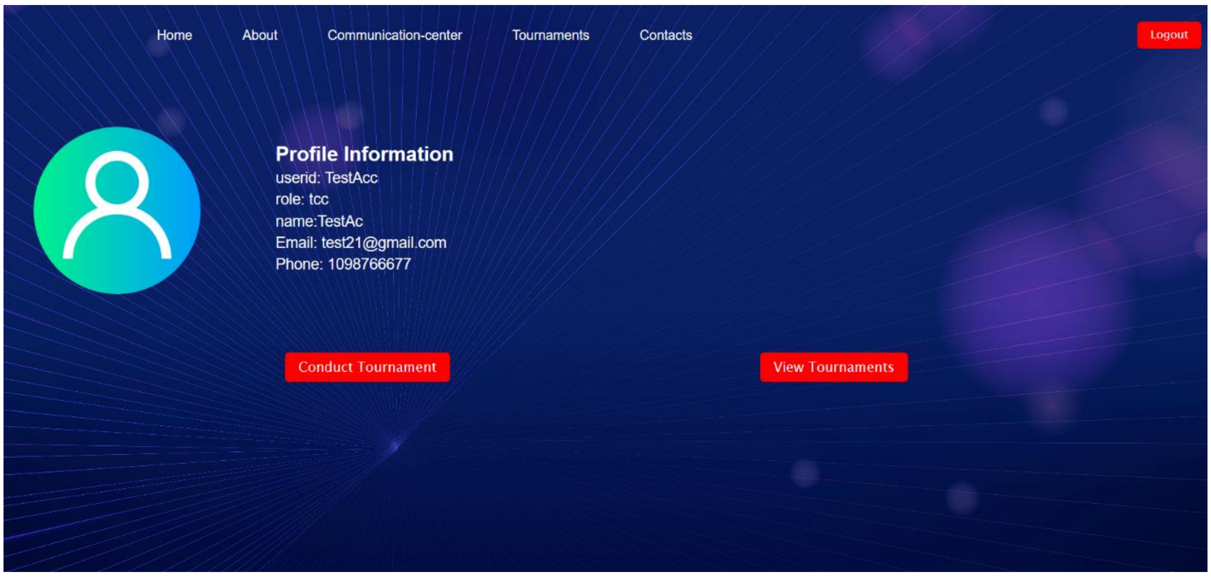
Home Page:



We're Here to

Elevate the standards of tournament management and participation through expert intermediation

Tournament conductor Profile:



Profile Information

userid: TestAcc

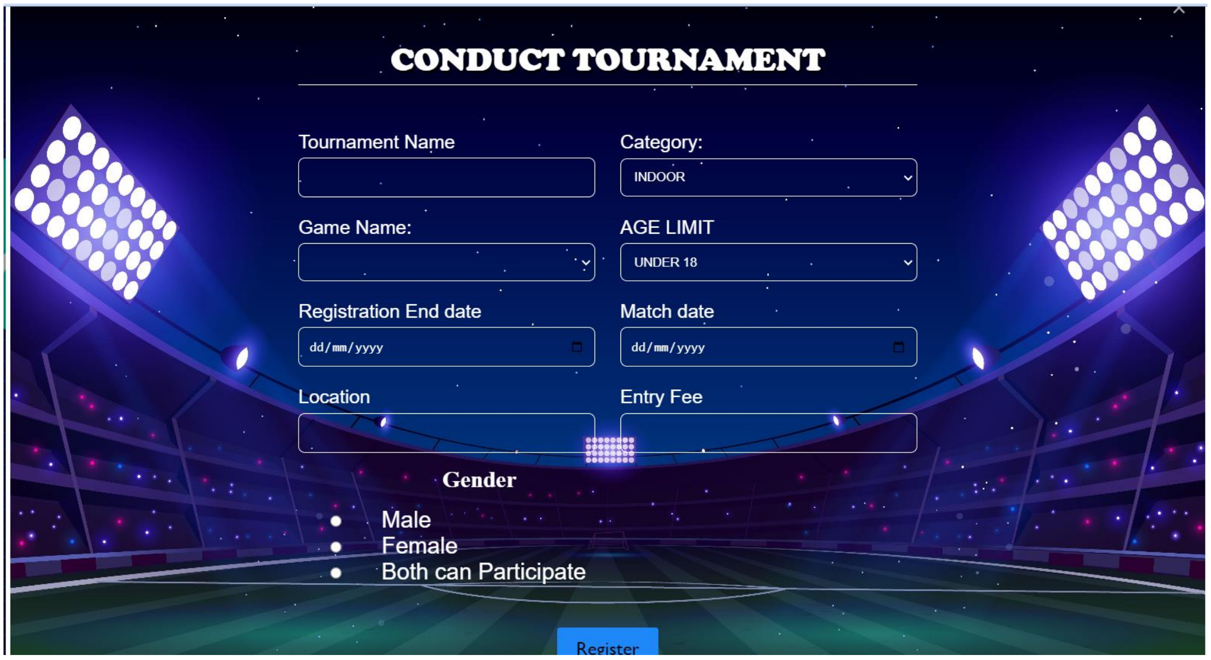
role: tcc

name:TestAc

Email: test21@gmail.com

Phone: 1098766677

Conduct Tournament:



CONDUCT TOURNAMENT

Tournament Name

Category: INDOOR

Game Name:

AGE LIMIT: UNDER 18

Registration End date: dd/mm/yyyy

Match date: dd/mm/yyyy

Location

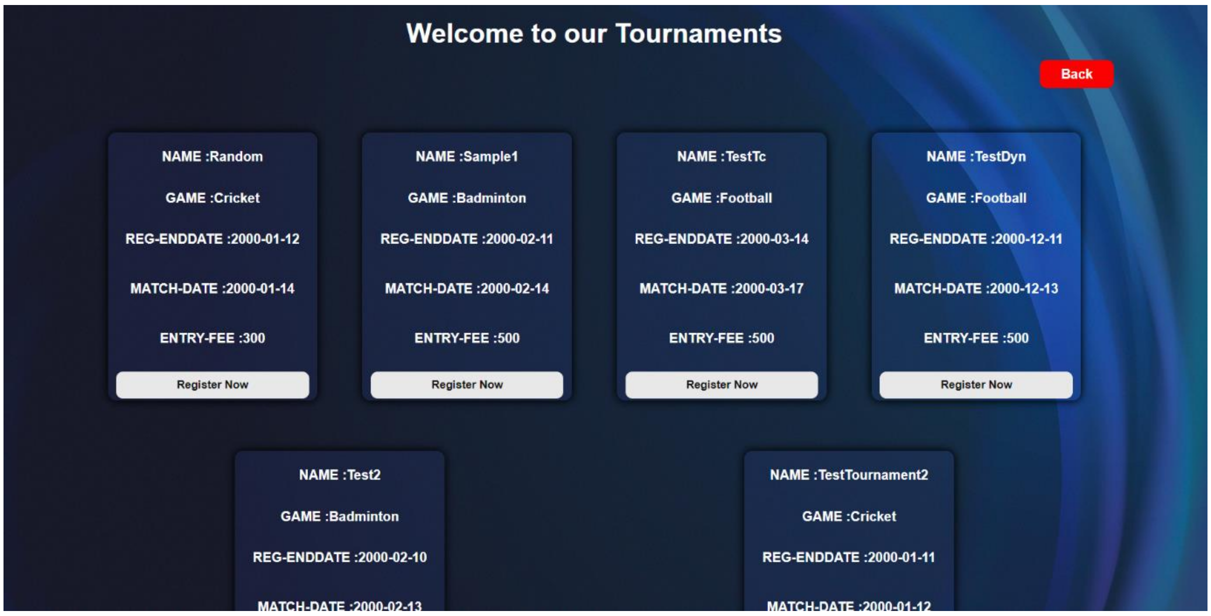
Entry Fee

Gender

- Male
- Female
- Both can Participate

Register

Tournament Page:



Welcome to our Tournaments

Back

NAME :Random
GAME :Cricket
REG-ENDDATE :2000-01-12
MATCH-DATE :2000-01-14
ENTRY-FEE :300
Register Now

NAME :Sample1
GAME :Badminton
REG-ENDDATE :2000-02-11
MATCH-DATE :2000-02-14
ENTRY-FEE :500
Register Now

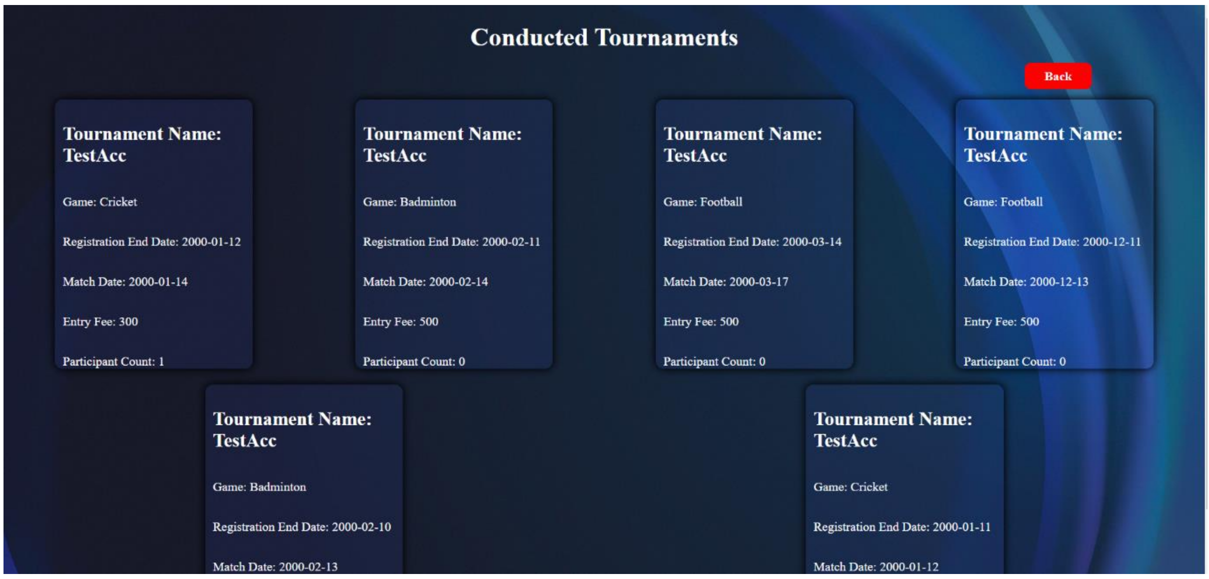
NAME :TestTc
GAME :Football
REG-ENDDATE :2000-03-14
MATCH-DATE :2000-03-17
ENTRY-FEE :500
Register Now

NAME :TestDyn
GAME :Football
REG-ENDDATE :2000-12-11
MATCH-DATE :2000-12-13
ENTRY-FEE :500
Register Now

NAME :Test2
GAME :Badminton
REG-ENDDATE :2000-02-10
MATCH-DATE :2000-02-13

NAME :TestTournament2
GAME :Cricket
REG-ENDDATE :2000-01-11
MATCH-DATE :2000-01-12

Personal Tournament:


Conducted Tournaments

Tournament Name: TestAcc
Game: Cricket
Registration End Date: 2000-01-12
Match Date: 2000-01-14
Entry Fee: 300
Participant Count: 1

Tournament Name: TestAcc
Game: Badminton
Registration End Date: 2000-02-11
Match Date: 2000-02-14
Entry Fee: 500
Participant Count: 0

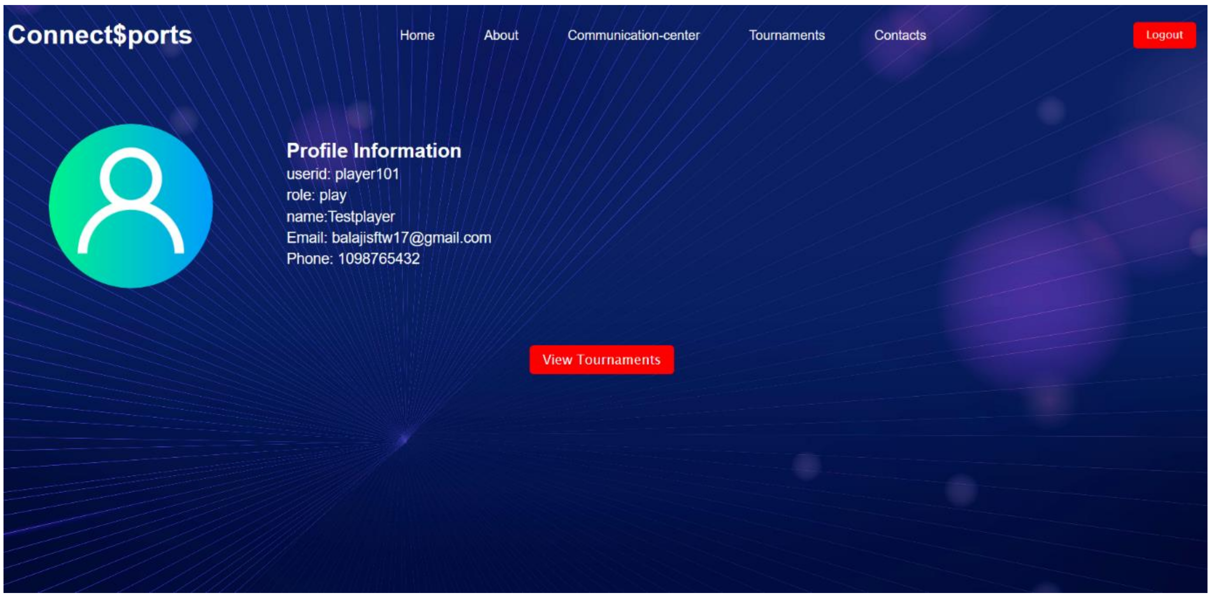
Tournament Name: TestAcc
Game: Football
Registration End Date: 2000-03-14
Match Date: 2000-03-17
Entry Fee: 500
Participant Count: 0

Tournament Name: TestAcc
Game: Football
Registration End Date: 2000-12-11
Match Date: 2000-12-13
Entry Fee: 500
Participant Count: 0

Tournament Name: TestAcc
Game: Badminton
Registration End Date: 2000-02-10
Match Date: 2000-02-13

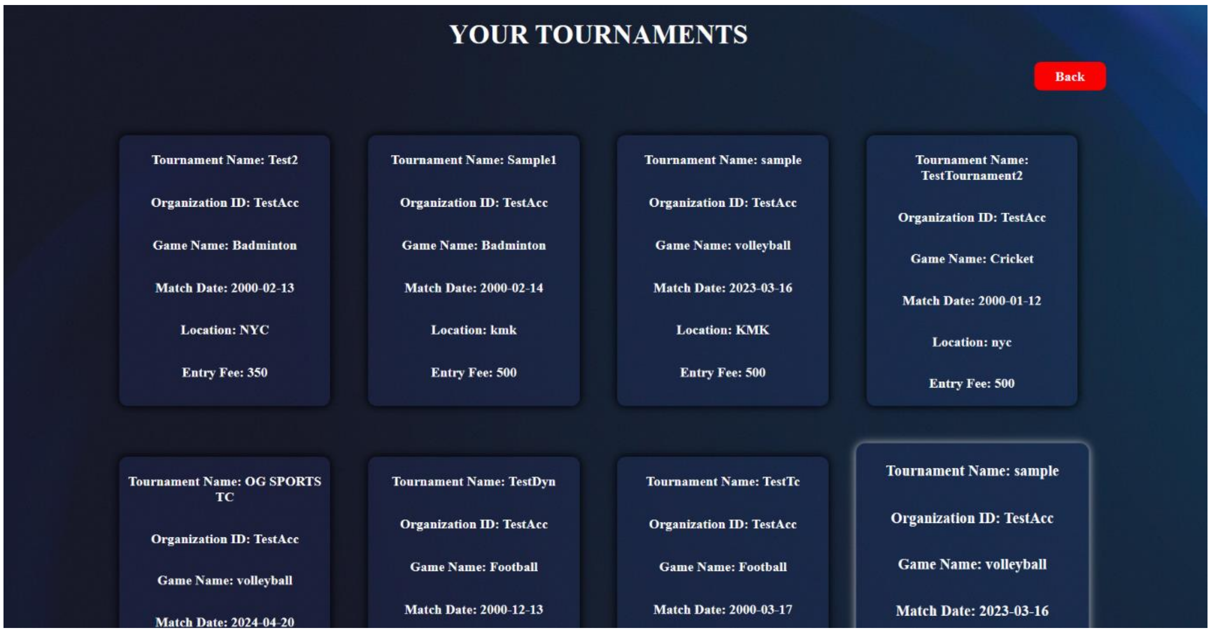
Tournament Name: TestAcc
Game: Cricket
Registration End Date: 2000-01-11
Match Date: 2000-01-12

Player profile:



Profile Information
userid: player101
role: play
name:Testplayer
Email: balajisftw17@gmail.com
Phone: 1098765432

Participated tournaments:



Tournament Name: Test2
Organization ID: TestAcc
Game Name: Badminton
Match Date: 2000-02-13
Location: NYC
Entry Fee: 350

Tournament Name: Sample1
Organization ID: TestAcc
Game Name: Badminton
Match Date: 2000-02-14
Location: kmk
Entry Fee: 500

Tournament Name: sample
Organization ID: TestAcc
Game Name: volleyball
Match Date: 2023-03-16
Location: KMK
Entry Fee: 500

Tournament Name: TestTournament2
Organization ID: TestAcc
Game Name: Cricket
Match Date: 2000-01-12
Location: nyc
Entry Fee: 500

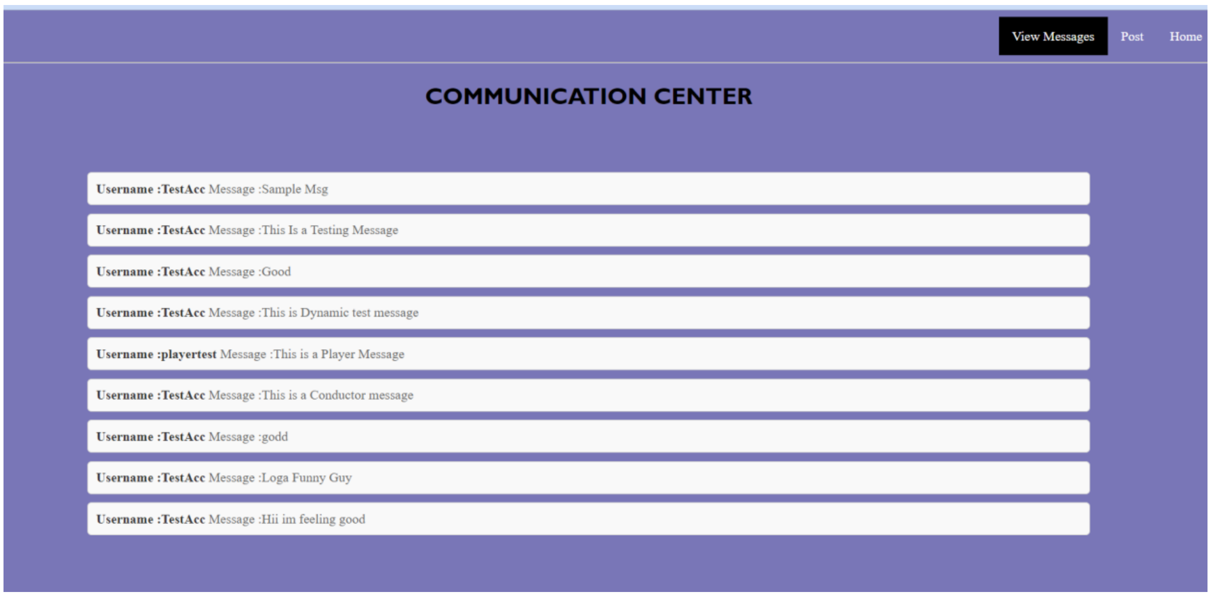
Tournament Name: OG SPORTS TC
Organization ID: TestAcc
Game Name: volleyball
Match Date: 2024-04-20

Tournament Name: TestDyn
Organization ID: TestAcc
Game Name: Football
Match Date: 2000-12-13

Tournament Name: TestTc
Organization ID: TestAcc
Game Name: Football
Match Date: 2000-03-17

Tournament Name: sample
Organization ID: TestAcc
Game Name: volleyball
Match Date: 2023-03-16

Communication center:



Username :TestAcc Message :Sample Msg

Username :TestAcc Message :This Is a Testing Message

Username :TestAcc Message :Good

Username :TestAcc Message :This is Dynamic test message

Username :playertest Message :This is a Player Message

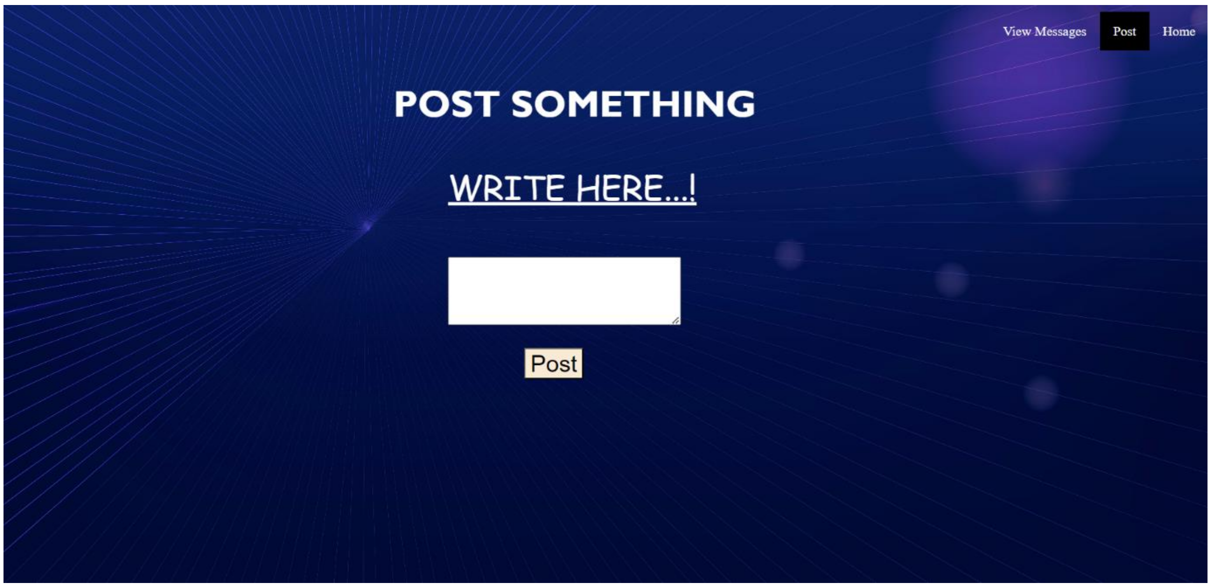
Username :TestAcc Message :This is a Conductor message

Username :TestAcc Message :godd

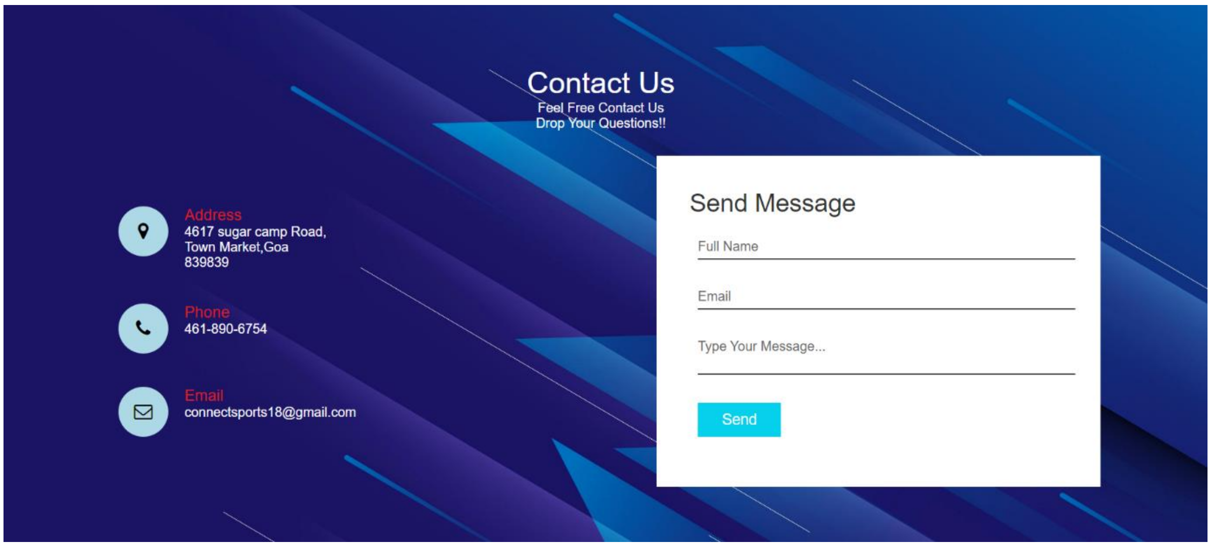
Username :TestAcc Message :Loga Funny Guy

Username :TestAcc Message :Hii im feeling good

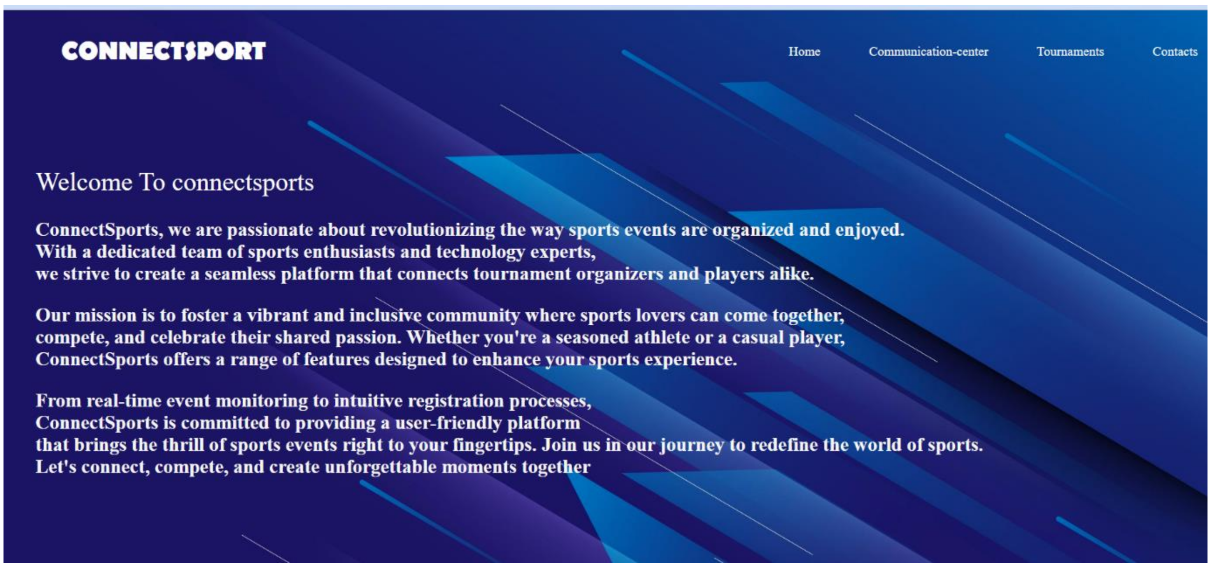
Messaging:



Contact us:



About:



CONNECTSPORT

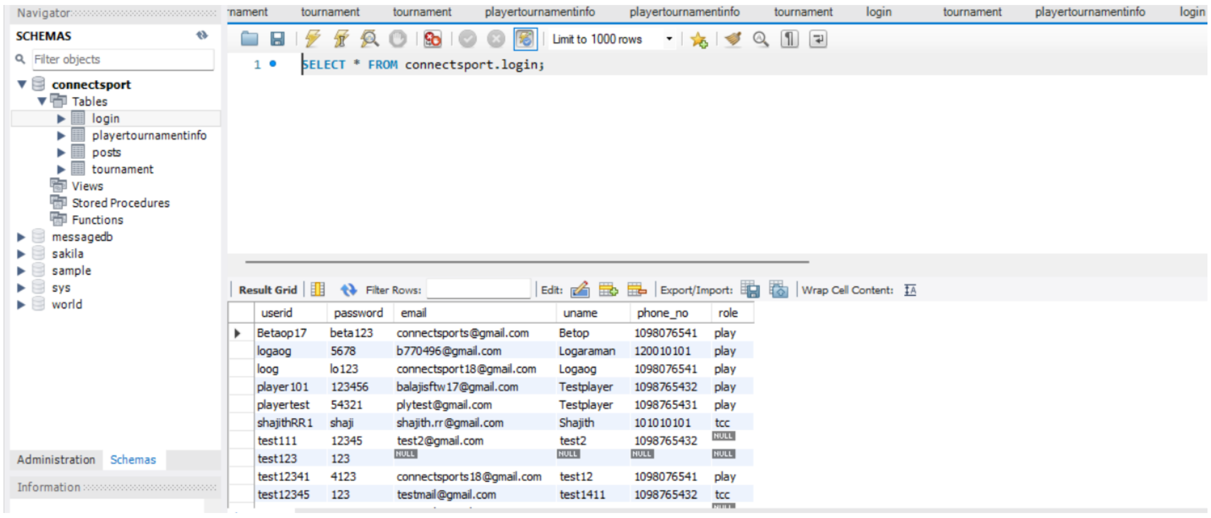
Welcome To connectsports

ConnectSports, we are passionate about revolutionizing the way sports events are organized and enjoyed.
With a dedicated team of sports enthusiasts and technology experts,
we strive to create a seamless platform that connects tournament organizers and players alike.

Our mission is to foster a vibrant and inclusive community where sports lovers can come together,
compete, and celebrate their shared passion. Whether you're a seasoned athlete or a casual player,
ConnectSports offers a range of features designed to enhance your sports experience.

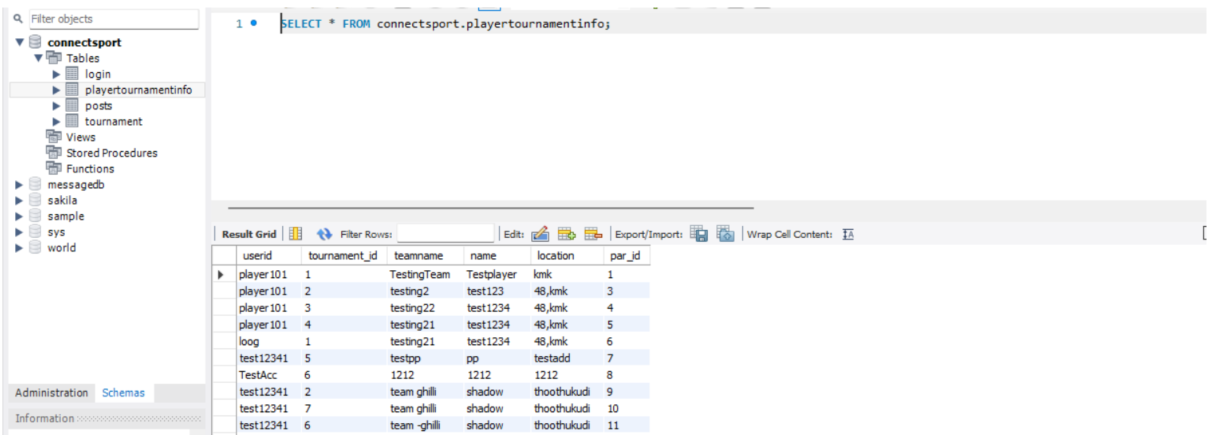
From real-time event monitoring to intuitive registration processes,
ConnectSports is committed to providing a user-friendly platform
that brings the thrill of sports events right to your fingertips. Join us in our journey to redefine the world of sports.
Let's connect, compete, and create unforgettable moments together

Output Design:



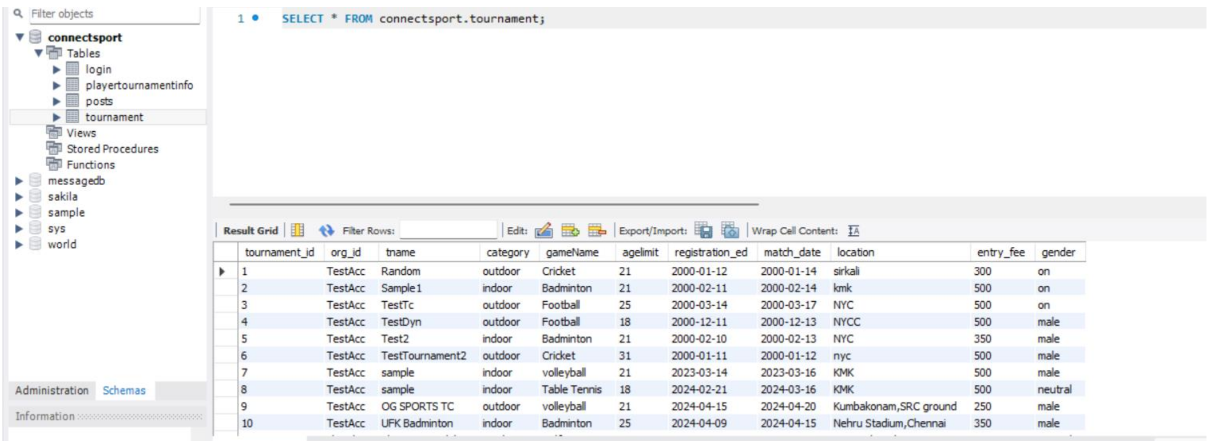
```
SELECT * FROM connectsport.login;
```

userid	password	email	uname	phone_no	role
Betaop17	beta123	connectsports@gmail.com	Betop	1098076541	play
logaog	5678	b770496@gmail.com	Logaraman	120010101	play
loog	lo123	connectsport18@gmail.com	Logaog	1098076541	play
player101	123456	balajisftw17@gmail.com	Testplayer	1098765432	play
playertest	54321	plytest@gmail.com	Testplayer	1098765431	play
shajithRR1	shaji	shajith.rr@gmail.com	Shajith	101010101	tcc
test111	12345	test2@gmail.com	test2	1098765432	NULL
test123	123	NULL	NULL	NULL	NULL
test12341	4123	connectsports18@gmail.com	test12	1098076541	play
test12345	123	testmail@gmail.com	test1411	1098765432	tcc



```sql
SELECT * FROM connectsport.playertournamentinfo;
```

userid	tournament_id	teamname	name	location	par_id
player101	1	TestingTeam	Testplayer	kmk	1
player101	2	testing2	test123	48,kmk	3
player101	3	testing22	test1234	48,kmk	4
player101	4	testing21	test1234	48,kmk	5
loog	1	testing21	test1234	48,kmk	6
test12341	5	testpp	pp	testadd	7
TestAcc	6	1212	1212	1212	8
test12341	2	team ghilli	shadow	thoothukudi	9
test12341	7	team ghilli	shadow	thoothukudi	10
test12341	6	team -ghilli	shadow	thoothukudi	11



```sql
SELECT * FROM connectsport.tournament;
```

tournament_id	org_id	tname	category	gameName	agelimit	registration_ed	match_date	location	entry_fee	gender
1	TestAcc	Random	outdoor	Cricket	21	2000-01-12	2000-01-14	sirkali	300	on
2	TestAcc	Sample1	indoor	Badminton	21	2000-02-11	2000-02-14	kmk	500	on
3	TestAcc	TestTc	outdoor	Football	25	2000-03-14	2000-03-17	NYC	500	on
4	TestAcc	TestDyn	outdoor	Football	18	2000-12-11	2000-12-13	NYCC	500	male
5	TestAcc	Test2	indoor	Badminton	21	2000-02-10	2000-02-13	NYC	350	male
6	TestAcc	TestTournament2	outdoor	Cricket	31	2000-01-11	2000-01-12	nyc	500	male
7	TestAcc	sample	indoor	volleyball	21	2023-03-14	2023-03-16	KMK	500	male
8	TestAcc	sample	indoor	Table Tennis	18	2024-02-21	2024-03-16	KMK	500	neutral
9	TestAcc	OG SPORTS TC	outdoor	volleyball	21	2024-04-15	2024-04-20	Kumbakonam,SRC ground	250	male
10	TestAcc	UFK Badminton	indoor	Badminton	25	2024-04-09	2024-04-15	Nehru Stadium,Chennai	350	male

5.CODING

LOGIN

```
@myapp.route('/login',methods=['GET','POST'])
def login():
    msg=''
    if request.method=='POST':
        username = request.form['username']
        password = request.form['passw']
        cursor = mysql.connection.cursor()
        cursor.execute('SELECT * FROM login WHERE email=%s AND password=%s',(username,password))
        record = cursor.fetchone()
        if record:
            session['loggedin']=True
            session['username']=record[0]
            return redirect(url_for('home'))
        else:
            msg='Incorrect username/password,Try again'
            flash(msg, 'error')
    return  render_template('login.html',msg=msg)
```

REGISTRATION

```
@myapp.route('/registration', methods=['GET', 'POST'])
def registration():
    msgr = ''
    if request.method == 'POST':
        username = request.form['userid']
        password = request.form['passw']

        email = request.form['email']

        uname = request.form['uname']

        phone_no = request.form['phnum']

        roles=request.form['role']

        cursor = mysql.connection.cursor()

        cursor.execute('SELECT * FROM login WHERE userid=%s', (username,))

        record = cursor.fetchone()

        if record:

            flash('Username already exists, please choose a different one.')

            return redirect(url_for('registration'))

        cursor.execute('INSERT INTO login (userid, password, email, uname, phone_no,role) 
VALUES (%s, %s, %s, %s, %s,%s)',

                        (username, password, email, uname, phone_no,roles,))

        mysql.connection.commit()

        cursor.close()

        flash('Registration successful! You can now log in.')

        return redirect(url_for('index'))

    return render_template('reg.html')
```

PROFILE/DASHBOARD

```
@myapp.route('/profiles')

def profiles():

    if 'loggedin' in session:

        usname = session['username']

        cursor = mysql.connection.cursor()
        cursor.execute('SELECT userid, email, uname, role, phone_no FROM login WHERE
userid = %s', (usname,))
        user_details = cursor.fetchone()
        cursor.close()
        if user_details:
            role = user_details[3]
            if role == 'tcc':
                return render_template('cprofile.html', user_details=user_details)
            elif role == 'play':
                return render_template('plyprofile.html', user_details=user_details)
            else:
                return "Invalid user role."
        else:
            return "User details not found."
```

TOURNAMENT CREATION:

```
@myapp.route('/create_tournament', methods=['POST'])
def create_tournament():
    if request.method == 'POST':
            cursor = mysql.connection.cursor()
            orgid = session['username']
            tournament_name = request.form['tname']
            category = request.form['category']
            game_name = request.form['gameName']
            age_limit = request.form['agelimit']
            reg_end_date = request.form['rd']
```

```
        match_date = request.form['td']

        location = request.form['location']

        entry_fee = request.form['efee']

        gender = request.form['gender']

        cursor.execute('INSERT INTO tournament(org_id,tname, category, gameName, 
agelimit, registration_ed, match_date, location, entry_fee, gender) VALUES (%s,%s, %s, %s, 
%s, %s, %s, %s, %s, %s)',

                (orgid,tournament_name, category, game_name, age_limit, reg_end_date, 
match_date, location, entry_fee, gender,))

        mysql.connection.commit()

        cursor.close()

        flash('Tournament created successfully!', 'success')

        return redirect(url_for('profiles'))
```

PLAYER REGISTRATION

```
@myapp.route('/playerinfo', methods=['POST'])

def playerinfo():

  if request.method == 'POST':

      u_id = session['username']

      t_id = request.form['tournamentId']

      teamname = request.form['teamName']

      user = request.form['yourName']

      loca = request.form['address']

      cursor = mysql.connection.cursor()

      cursor.execute('SELECT * FROM login WHERE userid=%s', (u_id,))

      user_role = cursor.fetchone()
```

```
        role = user_role[5]

        print(role)

        if role == 'tcc':

            flash('Tournament conductor cannot participate in the tournaments','error')

            return redirect(url_for('home'))

        elif role == 'play':

            cursor.execute('INSERT INTO playertournamentinfo (userid, tournament_id, 
teamname, name, location) VALUES (%s, %s, %s, %s, %s)', (u_id, t_id, teamname, user, 
loca))

            mysql.connection.commit()

            cursor.close()

            send_participation_mail(session['username'])

        else:

            return 'Invalid User Role'

        return 'Player information inserted successfully'

    else:

        return 'Method not allowed'
```

MySql Configuration

```
myapp.config['SECRET_KEY'] = '_5#y2L"F4Q8z\n\xec]/'

myapp.config['MYSQL_HOST'] = 'localhost'

myapp.config['MYSQL_USER'] = 'root'

myapp.config['MYSQL_PASSWORD'] = 'Betaop$17'

myapp.config['MYSQL_DB'] = 'connectsport'

mysql = MySQL(myapp)
```

CONTACT

```
@myapp.route('/contact',methods=['GET','POST'])
def contact():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        message = request.form['message']
        send_email(name, email, message)
        return 'Thank you for your message!'
    return render_template('contact.html')
def send_email(name, email, message):
    smtp_server = 'smtp.gmail.com'
    smtp_port = 587
    smtp_username = 'connectsports18@gmail.com'
    smtp_password = 'pjyv vosj ulgk vjsa'
    msg = EmailMessage()
    msg.set_content(f'From: {name}\nEmail: {email}\n\n{message}')
    msg['Subject'] = 'Contact Form Submission'
    msg['From'] = email
    msg['To'] = 'connectsports18@gmail.com'
    with smtplib.SMTP(smtp_server, smtp_port) as server:
        server.starttls()
        server.login(smtp_username, smtp_password)
        server.send_message(msg)
```

```
def send_participation_mail(player_id):
    smtp_server = 'smtp.gmail.com'
    smtp_port = 587
    smtp_username = ‘connectsports18@gmail.com’
    smtp_password = "pjyv vosj ulgk vjsa'
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT uname, email FROM login WHERE userid=%s', (player_id,))
    player = cursor.fetchone()
    if player:
        msg = EmailMessage()
        msg.set_content(f"Hello {player[0]},\n\n"
               f"Thank you for participating in the tournament.Use this mail in Matchday"
                f"\n Hope you had a great time" f"\nBest Regards" f"\nCONNECTSPORTTEAM")
        msg['Subject'] = 'Tournament Participation Details'
        msg['From'] = smtp_username
        msg['To'] = player[1]
        with smtplib.SMTP(smtp_server, smtp_port) as server:
            server.starttls()
            server.login(smtp_username, smtp_password)
            server.send_message(msg)
        cursor.close()
        return 'successful'
    else:
        return 'Player not found'
```

```
{% for chunk in tournaments|batch(4) %}
    <div class="card-container">
        {% for tournament in chunk %}
        <div class="card">
            <h1 class="tournament-name">NAME :{{ tournament[2] }}</h1>
            <h1 class="tournament-name">GAME :{{ tournament[4] }}</h1>
            <h1 class="tournament-date">REG-ENDDATE :{{ tournament[6] }}</h1>
            <h1 class="tournament-date">MATCH-DATE :{{ tournament[7] }}</h1>
            <h1 class="tournament-date">ENTRY-FEE :{{ tournament[9] }}</h1>
            <button type="submit" onclick="url_for('playerinfo')" class="register-btn"
                data-tournament-id="{{ tournament[0] }}" data-entry-fee="{{ tournament[9]
}}">Register Now</button>
        </div>
        {% endfor %}
    </div>
    {% endfor %}
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
document.getElementById('payButton').onclick = function () {
            var teamName = document.getElementById('teamName').value;
            var yourName = document.getElementById('yourName').value;
            var address = document.getElementById('address').value;
            var tournamentId = document.getElementById('tournamentId').value;
            var entryFee = document.getElementById('entryFee').value;
            var options = {
                "key": "rzp_test_dYStpQwu6bs3hh",
                "amount": entryFee * 100,
                "currency": "INR",
                "name": "connectsport",
                "description": "Tournament Registration Fee",
                "handler": function (response) {
                    alert("Payment successful: " + response.razorpay_payment_id);
                    document.getElementById('registrationForm').submit();
                },
                "prefill": {
                    "name": Developer,
                    "email": "connectsport18@gmail.com",
                    "contact": "9500797305"
                }
            };
            var rzp = new Razorpay(options);
            rzp.open();
        };
document.addEventListener("DOMContentLoaded", function () {
            var popupContainer = document.getElementById('popupContainer');
            var registerBtns = document.getElementsByClassName("register-btn");
            console.log("Register buttons found:", registerBtns.length);
            for (var i = 0; i < registerBtns.length; i++) {
                registerBtns[i].addEventListener('click', function () {
                    console.log("Register button clicked");
                    var tournamentId = this.getAttribute("data-tournament-id");
                    var entryFee = this.getAttribute("data-entry-fee");
                    popupContainer.style.display = "flex";
                    document.getElementById('tournamentId').value = tournamentId;
                    document.getElementById('entryFee').value = entryFee;
                });
            }
        });
        window.onclick = function (event) {
            if (event.target == popupContainer) {
                popupContainer.style.display = "none";
            }
        }
document.addEventListener("DOMContentLoaded", function () {
  var flashMessages = document.querySelectorAll(".flash-message");
  flashMessages.forEach(function (message) {
        setTimeout(function () {
          message.style.display = "none";
}, 5000);
  });
});
```

6.TESTING

6.Testing

Testing is the process of evaluating a software application or system to ensure that it meets specified requirements, functions correctly, and satisfies user expectations. It involves systematically executing the software under controlled conditions and comparing actual results against expected outcomes to identify defects, errors, or deviations from expected behaviour.

6.1 Unit Testing:

Unit testing is essential for the verification of the code produced during phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried the programming stage itself. All units were successfully tested.

6.2 Integration testing:

Integration testing is a systematic technique for constructing the software architecture while at the same time conducting tests to uncover error associated with interfacing. The objective is to take unit tested components and build a program structure.

7.Implementation

7.Implementation

7.1 Problems Faced

This is the first time we are using backend, so it takes too much of time to refer it, then the error occurred in backend is not simply solvable, we referred many websites to solve the backend errors and making payment system was the difficult part of this project.

7.2 Lessons Learnt

While developing this project we came across many lessons. They are as follows:

- Before starting this project, we must have proper plan about the project.
- We should not jump into coding directly. First project should be analyzed thoroughly.
- We include session concept in our system. Learn some more ideas and information about the project

8.Conclusion and Future Enhancements

8.Conclusion and Future Enhancements

8.1 Conclusion

In conclusion, This Sports intermediary Application offers a comprehensive solution for organizing and participating in sports tournaments with its user-friendly interface, real-time communication features, and efficient event management tools. This Application aims to enhance the overall experience of real-time sports events by addressing challenges such as authentication, data management, and performance,

8.2 Future Enhancements

In the future, we want to improve our payment system making it easier to use and safer for everyone who uses our app. We also want to provide options for conducting various unique games. These changes will make our app more enjoyable and useful for everyone who uses it.

9.REFERENCES

9.References

1. Trinity Software Academy: https://youtu.be/P5dESEQ-ce8?si=vYx6Lf0XweX7GcJL

2. freeCodeCamp.org: https://youtu.be/Qr4QMBUPxWo?si=0ikdwyhrrWgn9z5B

3. Bro code: https://youtube.com/playlist?list=PLZPZq0r_RZOMskz6MdsMOgxzheIyjo-BZ&si=2KunftYkT29wHuhL(Mysql)

4. https://flask.palletsprojects.com/en/3.0.x/

5. https://www.tutorialspoint.com/flask

6. https://razorpay.com/docs/payments/server-integration/python/install