**BACK-END CODE**

from flask import Flask,render\_template,request,session,url\_for,redirect,flash

from flask\_mysqldb import MySQL

import mysql.connector

import smtplib

from email.message import EmailMessage

import razorpay

myapp = Flask(\_\_name\_\_)

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)

razorpay\_key\_id = 'rzp\_test\_dYStpQwu6bs3hh'

razorpay\_key\_secret = 'mTN9AofGfNT330nrs19Pm0RZ'

razorpay\_client = razorpay.Client(auth=(razorpay\_key\_id, razorpay\_key\_secret))

@myapp.route('/')

def index():

    return render\_template('login.html')

@myapp.route('/home')

def home():

    if 'loggedin' in session:

        return render\_template('home.html', username=session['username'])

    return redirect(url\_for('index'))

@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)

@myapp.route('/logout')

def logout():

    session.pop('loggedin', None)

    session.pop('username', None)

    return redirect(url\_for('index'))

@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')

def registration\_mail(us\_id):

    smtp\_server = 'smtp.gmail.com'

    smtp\_port = 587

    smtp\_username = 'superstarsuperstar123123@gmail.com'

    smtp\_password = 'vfim ugim qljz jedd'

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT uname, email FROM login WHERE userid=%s', (us\_id,))

    player = cursor.fetchone()

    if player:

        msg = EmailMessage()

        msg.set\_content(f"Hello {player[0]},\n\n"

                        f"Welcome To Connectport Buddy! "

                        f"\n Hope you had a great time"

                        f"\nBest Regards"

                        f"\nCONNECTSPORT TEAM")

        msg['Subject'] = 'Tournament Participation Details'

        msg['From'] = smtp\_username

        msg['To'] = player[1]

        print(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'

@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."

@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'))

@myapp.route('/view\_conducted')

def view\_conducted():

    conduct\_id = session['username']

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT \* FROM tournament WHERE org\_id = %s', (conduct\_id,))

    conducted\_tournaments = cursor.fetchall()

    formatted\_tournaments = []

    for tournament in conducted\_tournaments:

        tournament\_id = tournament[0]

        cursor.execute('''

            SELECT COUNT(\*) AS participant\_count

            FROM playertournamentinfo ti

            WHERE ti.tournament\_id = %s

        ''', (tournament\_id,))

        participant\_count = cursor.fetchone()[0]

        formatted\_tournament = {

            'tournament\_details': tournament,

            'participant\_count': participant\_count

        }

        formatted\_tournaments.append(formatted\_tournament)

    cursor.close()

    return render\_template('concards.html', conducted\_tournaments=formatted\_tournaments)

@myapp.route('/about')

def about():

    return render\_template('about.html')

@myapp.route('/tournament\_card' )

def tournament\_card():

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT \* FROM tournament')

    tournaments = cursor.fetchall()

    cursor.close()

    return render\_template('card.html', tournaments=tournaments)

def send\_email(name, email, message):

    smtp\_server = 'smtp.gmail.com'

    smtp\_port = 587

    smtp\_username = 'superstarsuperstar123123@gmail.com'

    smtp\_password = 'vfim ugim qljz jedd'

    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)

@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')

@myapp.route('/message', methods=['GET', 'POST'])

def message():

    if request.method == 'POST':

        try:

            cursor = mysql.connection.cursor()

            uid = session['username']

            msgs = request.form['content']

            print(uid)

            print(msgs)

            cursor.execute('INSERT INTO connectsport.posts(user\_id, content) VALUES (%s, %s)', (uid, msgs,))

            mysql.connection.commit()

            cursor.close()

            flash('Message Posted!', 'post\_success')

            return redirect('/message')

        except Exception as e:

            flash('An error occurred while posting the message', 'merror')

            print('Error:', e)

    return render\_template('message.html')

@myapp.route('/postmsg')

def postmsg():

    cursor=mysql.connection.cursor()

    cursor.execute('SELECT \* FROM posts')

    mesgs = cursor.fetchall()

    cursor.close()

    return render\_template('postsmsg.html',mesgs=mesgs)

def user\_check(user\_id):

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT role FROM login WHERE userid=%s', (user\_id,))

    user\_role = cursor.fetchone()

    cursor.close()

    if user\_role and user\_role[0] == 'tcc':

        return True

    else:

        return False

@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'

@myapp.route('/payment', methods=['GET', 'POST'])

def payment():

    if request.method == 'POST':

        payment\_id = request.form['razorpay\_payment\_id']

        payment\_order\_id = session.get('payment\_order\_id')

        if payment\_order\_id:

            try:

                payment = razorpay\_client.payment.fetch(payment\_id)

                if payment['status'] == 'captured':

                    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('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

                    send\_participation\_mail(session['username'])

                    flash('Payment successful. Player information inserted successfully.', 'success')

                    return redirect(url\_for('home'))

                else:

                    flash('Payment verification failed. Please contact support.', 'error')

                    return redirect(url\_for('home'))

            except Exception as e:

                print('Error in payment verification:', e)

                flash('Payment verification failed. Please contact support.', 'error')

                return redirect(url\_for('home'))

        else:

            flash('Payment order not found. Please try again.', 'error')

            return redirect(url\_for('home'))

    else:

        # Render payment.html template for displaying Razorpay checkout form

        return render\_template('payment.html')

@myapp.route('/payment\_failure')

def payment\_failure():

    flash('Payment failed. Please try again.', 'error')

    return redirect(url\_for('home'))

@myapp.route('/ply\_tournaments')

def ply\_tournaments():

    ply\_id = session['username']

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT ti.userid, ti.tournament\_id, ti.name AS player\_name, ti.teamname, ti.location AS player\_location, '

                   't.org\_id, t.tname AS tournament\_name, t.gameName AS game\_name, t.match\_date, '

                   't.location AS tournament\_location, t.entry\_fee AS efee '

                   'FROM playertournamentinfo ti '

                   'JOIN tournament t ON ti.tournament\_id = t.tournament\_id '

                   'WHERE ti.userid = %s', (ply\_id,))

    player\_tournament\_details = cursor.fetchall()

    cursor.close()

    return render\_template('ply\_cards.html', player\_tournament\_details=player\_tournament\_details)

def send\_participation\_mail(player\_id):

    smtp\_server = 'smtp.gmail.com'

    smtp\_port = 587

    smtp\_username = 'superstarsuperstar123123@gmail.com'

    smtp\_password = 'vfim ugim qljz jedd'

    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"\nCONNECTSPORT TEAM")

        msg['Subject'] = 'Tournament Participation Details'

        msg['From'] = smtp\_username

        msg['To'] = player[1]

        print(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'

if \_\_name\_\_  == "\_\_main\_\_":

    myapp.run(debug=True)

 {% 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>

    <script>

        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.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": "Your Company Name",

                "description": "Tournament Registration Fee",

                "handler": function (response) {

                    alert("Payment successful: " + response.razorpay\_payment\_id);

                    document.getElementById('registrationForm').submit();

                },

                "prefill": {

                    "name": yourName,

                    "email": "connectsport18@gmail.com",

                    "contact": "9500797305"

                }

            };

            var rzp = new Razorpay(options);

            rzp.open();

        };

    </script>

</body>

    {% for formatted\_tournament in conducted\_tournaments %}

        {% if loop.index % 4 == 1 %}

            <div class="row">

        {% endif %}

        <div class="card">

            <h2>Tournament Name: {{ formatted\_tournament['tournament\_details'][1] }}</h2>

            <p>Game: {{ formatted\_tournament['tournament\_details'][4] }}</p>

            <p>Registration End Date: {{ formatted\_tournament['tournament\_details'][6] }}</p>

            <p>Match Date: {{ formatted\_tournament['tournament\_details'][7] }}</p>

            <p>Entry Fee: {{ formatted\_tournament['tournament\_details'][9] }}</p>

            <p>Participant Count: {{ formatted\_tournament['participant\_count'] }}</p>

        </div>

        {% if loop.index % 4 == 0 or loop.last %}

            </div>

        {% endif %}

    {% endfor %}

div class="profile-info">

            <h2>Profile Information</h2>

            <p>userid: {{ user\_details[0] }}</p>

            <p>role: {{ user\_details[3] }}</p>

            <p>name:{{ user\_details[2] }}</p>

            <p>Email: {{ user\_details[1] }}</p>

            <p>Phone: {{ user\_details[4] }}</p>

        </div>

    </div>

 <h2>COMMUNICATION CENTER</h2>

    <div class="message-container">

        {% for ms in mesgs %}

        <div class="message">

            <span class="username">Username :{{ ms[0] }}</span>

            <span class="content">Message :{{ ms[1] }}</span>

        </div>

        {% endfor %}

document.addEventListener("DOMContentLoaded", function () {

  var flashMessages = document.querySelectorAll(".flash-message");

  flashMessages.forEach(function (message) {

    setTimeout(function () {

      message.style.display = "none";

}, 5000);

  });

});