##### CONFERENCE ROOM BOOKING SYSTEM

##### INTERNSHIP REPORT

**Submitted by**

|  |  |
| --- | --- |
| **RAMANAN M** | **21DX17** |
| **BARATH K N** | **22XH04** |
| **MUKESH BABU A** | **22XH06** |
| **THILLAIARASU T** | **21XH07** |

##### Under the guidance of

In partial fulfillment of the requirement for the award of

DIPLOMA IN COMPUTER ENGINEERING

**STATE BOARD OF TECHNICAL EDUCATION**

**GOVERNMENT OF TAMILNADU**



##### JUNE 2023

##### DEPARTMENT OF COMPUTER ENGINEERING

##### PSG POLYTECHNIC COLLEGE

(Autonomous and an ISO 9001 certified Institution)

COIMBATORE – 641 004

##### PSG POLYTECHNIC COLLEGE

(Autonomous and an ISO 9001 certified Institution)

##### DEPARTMENT OF COMPUTER ENGINEERING

COIMBATORE – 641 004

**CERTIFICATE**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **RAMANAN M** | **REGISTER NUMBER** | **21DX17** |
| **NAME** | **BARATH K N** | **REGISTER NUMBER** | **22XH04** |
| **NAME** | **MUKESH BABU A** | **REGISTER NUMBER** | **22XH06** |
| **NAME** | **THILLAIARASU T** | **REGISTER NUMBER** | **21XH07** |

This is to certify that the Project report entitled

##### CONFERENCE ROOM BOOKING SYSTEM

has been submitted by

**Mr. RAMANAN M**

**Mr. BARATH K N**

**Mr. MUKESH BABU A**

**Mr. THILLAIARASU T**

In partial fulfillment for the award of

DIPLOMA IN COMPUTER ENGINEERING

of the State Board of Technical Education,

Government of Tamil Nadu.

during the academic year 2022 - 2023

Faculty guide Head of the Department

Certified that the candidate was examined by us in the Project work viva-voce examination held on ………………….

(Internal Examiner) (External Examiner)

**ABSTRACT**

This abstract aims to provide a concise overview of the process and considerations involved in booking a conference hall for an event. Organizing a successful conference requires careful planning and attention to detail. One critical aspect is securing an appropriate conference hall that meets the event's requirements. This abstract highlights the essential factors to consider during the conference hall booking process. The first step is to identify the specific needs of the event. This includes determining the estimated number of attendees, the duration of the conference, desired amenities and any additional requirements such as breakout rooms or catering facilities. Thorough research is vital to identify suitable conference halls that align with the event's requirements. Factors to consider include location, capacity, availability, cost, accessibility, and reputation.

**SQL QUERIES**

create database makemymeeting default character set utf8 collate utf8\_general\_ci;

use makemymeeting;

create table accounts(

id int(11) not null auto\_increment,

username varchar(50) not null,

password varchar(255) not null,

email varchar(255) not null,

location varchar(100) not null,

company varchar(100) not null ,

positon varchar(50) not null,

primary key (id)

) engine = InnoDB auto\_increment = 2 default charset = utf8;

SELECT \* FROM meetings ;

truncate table company;

drop table accounts;

insert into accounts values ( 1, "Admin" , "12345678" ,"Admin@gmail.com" , "Coimbatore" , "psg", "Admin");

insert into accounts values (2, ”Emp1” , ”12345678” , “emp1@gmail.com” , “Coimbatore”,”psg”,”Employee” );

create table company(

    name varchar(100) not null,

    location varchar(100) not null,

    no\_of\_rooms int not null,

    primary key (name)

)engine = InnoDB default charset = utf8;

insert into company values ("psg","Coimbatore",4);

select \* from company ;

create table room(

    company varchar(100) not null,

    name varchar(100) ,

    time1 varchar(50) ,

    time2 varchar(50) ,

    time3 varchar(50) ,

    time4 varchar(50) ,

    time5 varchar(50) ,

    foreign key (company) references company(name)

)engine = InnoDB default charset = utf8;

insert into room values ("psg",null,null,null,null,null,null);

drop table room;

create table meetings(

    user varchar(50) not null,

    meeting1 varchar(50),

    meeting2 varchar(50),

    meeting3 varchar(50),

    meeting4 varchar(50),

    meeting5 varchar(50)

)engine = InnoDB default charset = utf8;

drop table meetings;

select \* from meetings

**PYTHON PROGRAM**

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

from flask\_mysqldb import MySQL

import MySQLdb.cursors

import re

import json

app = Flask(\_\_name\_\_)

global account ,room, room1, room2, room3,room4,room5, meet

app.secret\_key = 'RAMANAN'

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'Ramanan1836@'

app.config['MYSQL\_DB'] = 'makemymeeting'

mysql = MySQL(app)

@app.route('/')

def home():

    msg = ''

    return render\_template('Index.html', msg = msg)

@app.route('/login', methods =['GET', 'POST'])

def login():

    print ("login")

    msg = " "

    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:

        global account ,room, room1, room2, room3,room4,room5

        username = request.form['username']

        password = request.form['password']

        print ("requested")

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        cursor.execute('SELECT \* FROM accounts WHERE username = % s AND password = % s', (username, password, ))

        print ("cursor")

        account = cursor.fetchone()

        print ("fetched")

        if account:

            session['loggedin'] = True

            session['id'] = account['id']

            session['username'] = account['username']

            if account["positon"] == "Employee":

                cursor.execute('SELECT name,time1,time2,time3,time4,time5 FROM room WHERE company = % s', (account['company'], ))

                room = cursor.fetchall()

                print(room)

                room1 = room[0]

                room2 = room[1]

                room3 = room[2]

                room4 = room[3]

                room5 = room[4]

                return render\_template('E-HOMEPAGE.html',room1 = room1,room2 = room2,room3 =room3,room4 = room4,room5 = room5 )

            else:

                return render\_template('A-HOMEPAGE.html',msg =msg)

        else:

            msg = 'Incorrect username / password !'

    return render\_template('LOGIN.html')

@app.route('/logout')

def logout():

    session.pop('loggedin', None)

    session.pop('id', None)

    session.pop('username', None)

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

@app.route('/register', methods =['GET', 'POST'])

def register():

    msg = ''

    if request.method == 'POST':

        usertype = request.form['usertype']

        username = request.form['username']

        company = request.form['company']

        location = request.form['location']

        email = request.form['email']

        password = request.form['password']

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        cursor.execute('SELECT \* FROM accounts WHERE username = % s', (username, ))

        account = cursor.fetchone()

        cursor.execute('SELECT \* FROM accounts WHERE company = % s', (company, ))

        com = cursor.fetchone()

        if account:

            msg = 'Account already exists !'

        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

            msg = 'Invalid email address !'

        elif not re.match(r'[A-Za-z0-9]+', username):

            msg = 'Username must contain only characters and numbers !'

        elif not username or not password or not email or not location or not company:

            msg = 'Please fill out the form !'

        elif usertype == 'Admin' and com:

            msg = 'Your Company is already registered!!'

        else:

            if usertype == 'Employee':

                cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s, % s, % s, % s)', (username, password, email, location, company, usertype, ))

                mysql.connection.commit()

                msg = 'You have successfully registered !'

                for i in range(0,5):

                    cursor.execute('insert into meetings values (%s ,null,null,null,null,null)',(username,))

                    mysql.connection.commit()

                return render\_template('LOGIN.html', msg = msg)

            elif usertype == 'Admin':

                cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s, % s, % s, % s)', (username, password, email, location, company, usertype, ))

                mysql.connection.commit()

                cursor.execute('INSERT INTO company VALUES ( % s, % s, 0)', ( company, location, ))

                mysql.connection.commit()

                i = 0

                for i in range(0,5):

                    room\_name = "ROOM "+str(i+1)

                    cursor.execute('insert into room values (%s ,%s ,"available","available","available","available","available")',(company,room\_name,))

                    mysql.connection.commit()

                    cursor.execute('insert into meetings values (%s ,null,null,null,null,null)',(username,))

                    mysql.connection.commit()

                msg = 'You have successfully registered !'

                return render\_template('LOGIN.html', msg = msg)

    return render\_template('SIGNUP.html', msg = msg)

@app.route('/time11')

def time11():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time1 = null where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting1 = %s where user = %s',(room1['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time12')

def time12():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time2 = null where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting2 = %s where user = %s',(room1['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time13')

def time13():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time3 = null where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting3 = %s where user = %s',(room1['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time14')

def time14():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time4 = null where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting4 = %s where user = %s',(room1['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time15')

def time15():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time5 = null where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting5 = %s where user = %s',(room1['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time21')

def time21():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time1 = null where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting1 = %s where user = %s',(room2['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time22')

def time22():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time2 = null where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting2 = %s where user = %s',(room2['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time23')

def time23():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time3 = null where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting3 = %s where user = %s',(room2['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time24')

def time24():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time4 = null where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting4 = %s where user = %s',(room2['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time25')

def time25():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time5 = null where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting5 = %s where user = %s',(room2['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time31')

def time31():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time1 = null where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting1 = %s where user = %s',(room3['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time32')

def time32():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time2 = null where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting2 = %s where user = %s',(room3['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time33')

def time33():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time3 = null where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting3 = %s where user = %s',(room3['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time34')

def time34():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time4 = null where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting4 = %s where user = %s',(room3['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time35')

def time35():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time5 = null where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting5 = %s where user = %s',(room3['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time41')

def time41():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time1 = null where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting1 = %s where user = %s',(room4['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time42')

def time42():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time2 = null where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting2 = %s where user = %s',(room4['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time43')

def time43():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time3 = null where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting3 = %s where user = %s',(room4['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time44')

def time44():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time4 = null where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting4 = %s where user = %s',(room4['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time45')

def time45():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time5 = null where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting5 = %s where user = %s',(room4['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time51')

def time51():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time1 = null where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting1 = %s where user = %s',(room5['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time52')

def time52():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time2 = null where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting2 = %s where user = %s',(room5['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time53')

def time53():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time3 = null where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting3 = %s where user = %s',(room5['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time54')

def time54():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time4 = null where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting4 = %s where user = %s',(room5['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/time55')

def time55():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update room set time5 = null where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('update meetings set meeting5 = %s where user = %s',(room5['name'],account['username'],))

    mysql.connection.commit()

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

@app.route('/meetings')

def meetings():

    print("meetings")

    global meet

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchone()

    print(meet)

    return render\_template('meetings.html', meet = meet)

@app.route('/remove1')

def remove1():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update meetings set meeting1 = null where user = %s',(account['username'],))

    mysql.connection.commit()

    cursor.execute('update room set time1 = "available" where company = %s and name = %s',(account['company'],room1['name'],))

    mysql.connection.commit()

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchall()

    return render\_template('meetings.html', meet = meet)

@app.route('/remove2')

def remove2():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update meetings set meeting2 = null where user = %s',(account['username'],))

    mysql.connection.commit()

    cursor.execute('update room set time2 = "available" where company = %s and name = %s',(account['company'],room2['name'],))

    mysql.connection.commit()

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchall()

    return render\_template('meetings.html', meet = meet)

@app.route('/remove3')

def remove3():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update meetings set meeting3 = null where user = %s',(account['username'],))

    mysql.connection.commit()

    cursor.execute('update room set time3 = "available" where company = %s and name = %s',(account['company'],room3['name'],))

    mysql.connection.commit()

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchall()

    return render\_template('meetings.html', meet = meet)

@app.route('/remove4')

def remove4():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update meetings set meeting4 = null where user = %s',(account['username'],))

    mysql.connection.commit()

    cursor.execute('update room set time4 = "available" where company = %s and name = %s',(account['company'],room4['name'],))

    mysql.connection.commit()

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchall()

    return render\_template('meetings.html', meet = meet)

@app.route('/remove5')

def remove5():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute('update meetings set meeting5 = null where user = %s',(account['username'],))

    mysql.connection.commit()

    cursor.execute('update room set time5 = "available" where company = %s and name = %s',(account['company'],room5['name'],))

    mysql.connection.commit()

    cursor.execute('SELECT \* FROM meetings WHERE user = %s', (account['username'], ))

    meet = cursor.fetchall()

  return render\_template('meetings.html', meet = meet)

**HTML**

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Welcome</title>

        <link rel="icon" type="image/x-icon" href="{{ url\_for('static', filename='ICON.png') }}">

        <style>

            .di{

                background-color: white;

                padding: 20px 50px 30px 50px;

                border-radius:50px;

                text-align: "right";

            }

            .loginbutton{

                background-color: #258cda;

                color: white;

                padding: 5px 25px 5px 25px;

                border: none;

                margin-top:10px;

                font-family: "Roboto";

                position: relative;

                bottom: 0;

                transition: bottom 0.5s ease;

                border-radius:10px;

            }

            .loginbutton:hover{

                bottom:20px;

            }

            body{

                background-image: url("{{ url\_for('static', filename='signup\_background.jpg') }}");

                background-repeat: no-repeat;

                background-attachment: fixed;

                background-size: cover;

            }

            .rect{

                background-color: white;

                width: 350px;

                height: 300px;

            }

            .content{

                border: 5px solid white;

                padding: 20px 50px;

                height: 520px;

                border-radius:50px;

            }

            .image{

                float:"right";

                display: block;

                padding: 60px 0px;

                margin-top:auto;

                margin-bottom: auto;

                width: 55%;

            }

            .text{

                padding: 40px 40px 0px 40px;

                font-family: "Roboto";

                font-size:200%;

            }

            .logo{

                width: 171px;

                height: 40px;

                padding: 0px 0px 30px 10px;

                margin-top:3px;

            }

        </style>

    </head>

    <body>

        <div class = di align="right">

            <img class = "logo" align = "left" src="{{ url\_for('static', filename='LOGO.png') }}" alt="ICON">

            <a align = "right"  href = "{{url\_for('login')}}" class = "loginbutton" style = " text-decoration:none; font-size:small;">Login</a>

        </div>

        <br>

        <br>

        <div class = "content">

            <img align = "right" class = "image" src="{{ url\_for('static', filename='welcome.png') }}" alt="Conference room">

            <div style = "padding:80px 80px">

                <div class = "rect">

                    <b>

                        <pre class = "text">

YOU CREATE

IDEAS,

WE CREATE

YOUR MEETINGS.</b>

<a href = "{{url\_for('register')}}" style = "padding: 10px 25px; text-decoration:none; font-size:small;" class = "loginbutton">Get Started</a>

            </pre>

                </div>

            </div>

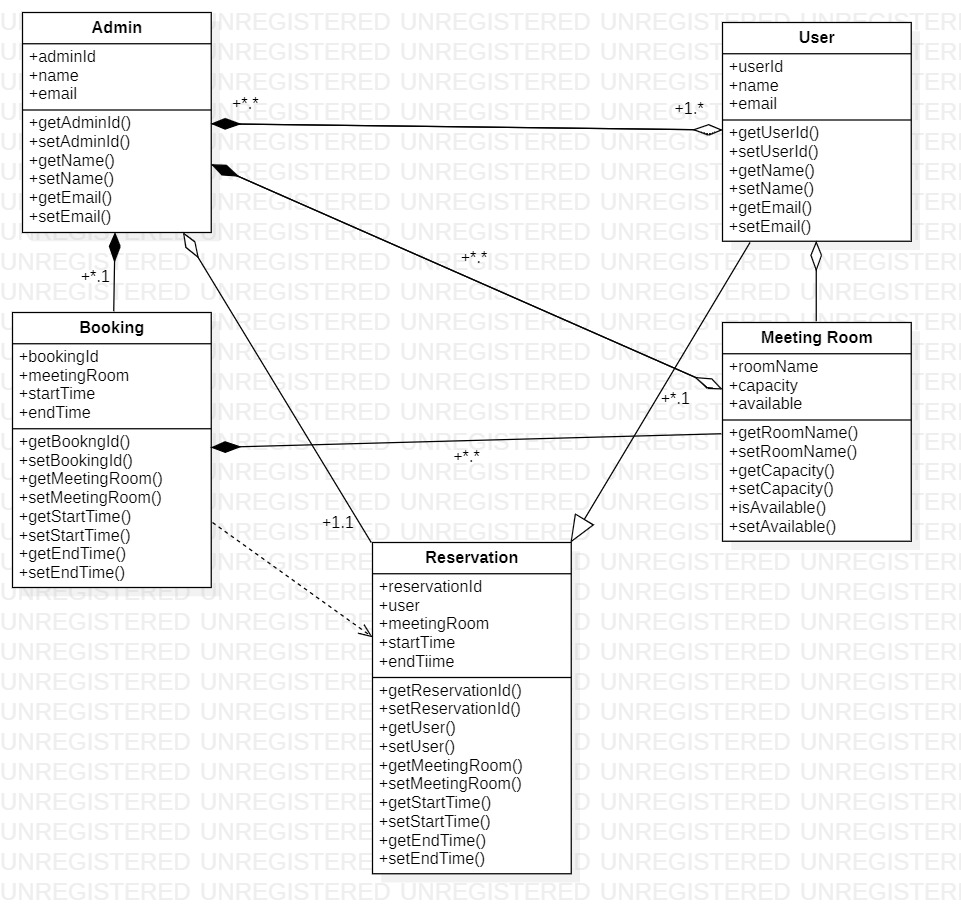
        </div>

    </body>

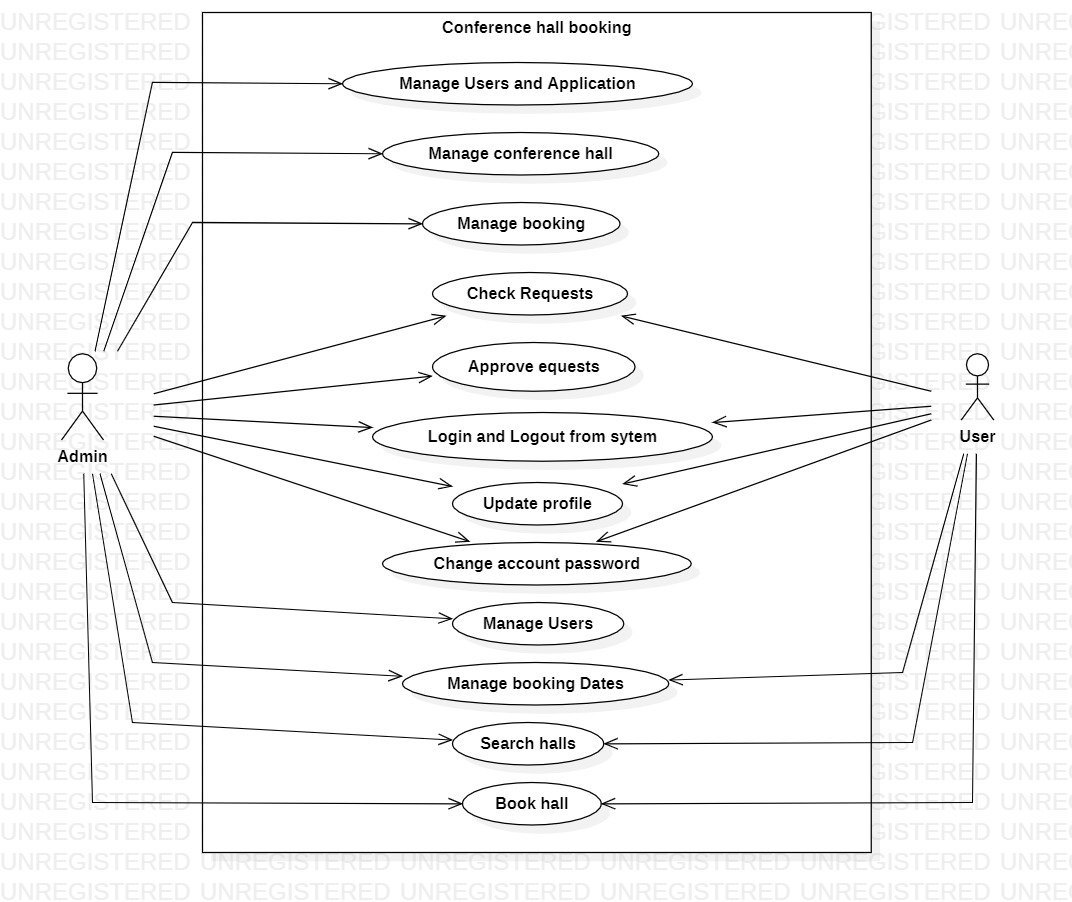
</html>

**UML DIAGRAMS**

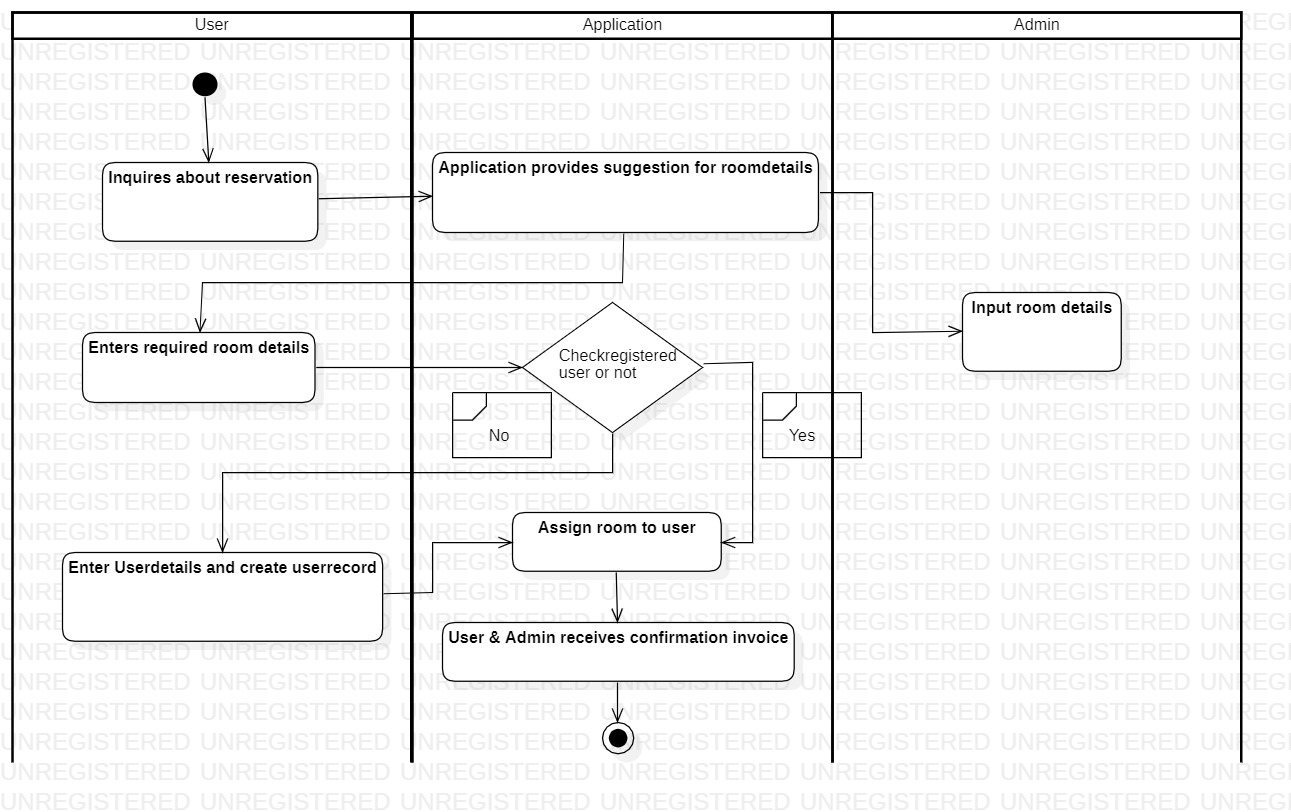
**CLASS DIAGRAM**

****

**USE CASE DIAGRAM**



**ACTIVITY DIAGRAM**



**DEPLOYMENT DIAGRAM**

