**Python Database Integration Project**

**Event Management System**

**PROBLEM DESCRIPTION:**

**EVENT MANAGEMENT SYSTEM:**

Event Management application keeps track of necessary information required for an event.

Information includes event details, organizer details, volunteer details, visitor details.

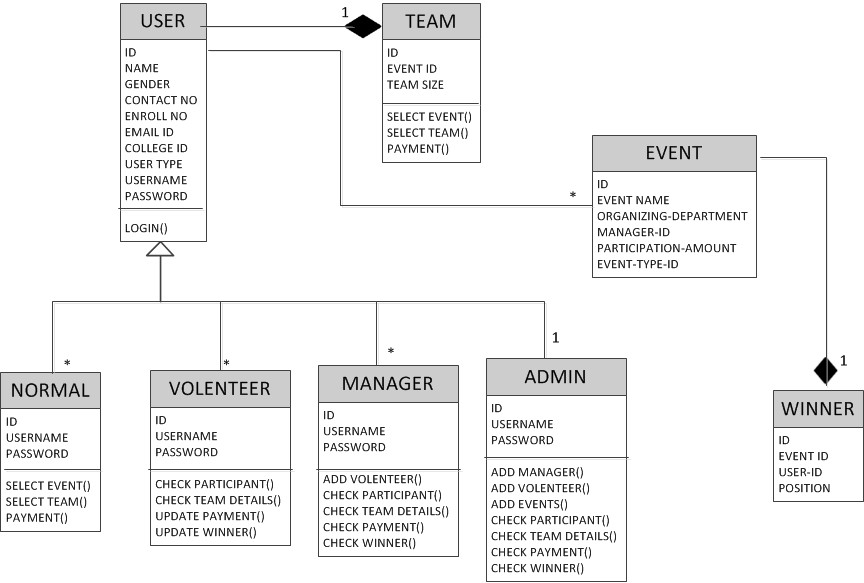
Event details contain Event Name, Date, Duration, Place, and Website.

Organizer details contain Event Name, Organizer Name, Address, Contact No, Email Address, and Role.

Volunteer details contain Event Name, Volunteer Name, Address, Contact No, Email Address, and Role.

Visitor details contain Event Name, Visitor Name, Address, Contact No, Email Address, Role.

**CLASS DIAGRAM:**

****

**Database:** Implement the above class diagram and connect it with database tables to maintain event details. Validate the event details and. Demo (Main class) must have the option to select for manipulating with database like insert/modify/delete/search/report.

**TABLE STRUCTURE:**

**TABLE USERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | userid | Varchar2 | 20 | Primary key |
| 2 | name | Varchar2 | 25 | Not null |
| 3 | gender | Varchar2 | 25 | Not null  Check in(‘male’,’female’,’other’) |
| 4 | contatcno | Number | 22 | Not null  Unique  Check length=10 |
| 5 | enrollno | Varchar2 | 10 | Not null |
| 6 | emailid | Varchar2 | 25 | Not null  unique |
| 7 | clgid | Varchar2 | 25 | Not null |
| 8 | teamid | Varchar2 | 10 | Default  null |
| 9 | usertype | Varchar2 | 25 | Not null  Check in(‘normal,’volunteer,’manager’,’admin’) |
| 10 | username | Varchar2 | 25 | Not null  unique |
| 11 | password | Varchar2 | 10 | Not null  Length>=8 |

**TABLE EVENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | eventid | Varchar2 | 20 | Primary key |
| 2 | eventname | Varchar2 | 25 | Not null |
| 3 | organisingdept | Varchar2 | 10 | Not null  Check in(‘IT’,’CS’,’ECE’,’EEE’,’MCT’,’MECH’) |
| 4 | managerid | Varchar2 | 20 | Not null |
| 5 | pamount | Number |  | Not null |
| 6 | etypeid | Varchar2 | 20 | Not null  Check in(‘technical,’non technical’) |

**TABLE TEAM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | teamid | Varchar2 | 10 | Primary key |
| 2 | eventid | Varchar2 | 10 | Foreign key |
| 3 | teamsize | number |  | Not null  Check (teamsize<=5) |
| 4 | payment | Varchar2 | 5 | Not null  Check in(‘yes’,’no’) |

**TABLE WINNER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | winnerid | Varchar2 | 10 | Primary key |
| 2 | eventid | Varchar2 | 20 | Not null  Foreign key |
| 3 | teamid | Varchar2 | 10 | Not null  Foreign key |
| 4 | position | Number | 22 | Not null  Check position in (1,2,3) |

**TABLE NORMAL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | userid | Varchar2 | 20 | Primary key |
| 2 | username | Varchar2 | 25 | Not null  unique |
| 3 | password | Varchar2 | 10 | Not null  Check(length(password)>=8) |

**TABLE VOLUNTEER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | userid | Varchar2 | 20 | Primary key |
| 2 | username | Varchar2 | 25 | Not null  unique |
| 3 | password | Varchar2 | 10 | Not null  Check(length(password)>=8) |

**TABLE MANAGER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | userid | Varchar2 | 20 | Primary key |
| 2 | username | Varchar2 | 25 | Not null  unique |
| 3 | password | Varchar2 | 10 | Not null  Check(length(password)>=8) |

**TABLE ADMIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Column Name** | **Type** | **Size** | **Description** |
| 1 | userid | Varchar2 | 20 | Primary key |
| 2 | username | Varchar2 | 25 | Not null  unique |
| 3 | password | Varchar2 | 10 | Not null  Check(length(password)>=8) |

**SAMPLE USERS TABLE**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| USERID | NAME | | GENDER | CONTACTNO | | | ENROLLNO | | EMAILID |
| us006 | ram | | male | 8457383737 | | | en006 | | 16euit003@skcet.ac.in |
| us001 | priya | | female | 9876544210 | | | en001 | | 16eucs160@skcet.ac.in |
| us002 | rohit | | male | 9234549090 | | | en002 | | 16euec035@skcet.ac.in |
| us003 | suresh | | male | 9357986420 | | | en003 | | 16tuit081@skcet.edu.in |
| us004 | sachin | | male | 9956903478 | | | en004 | | 16tuec048@skct.edu.in |
| us005 | sathya | | female | 9753444680 | | | en005 | | 16euit163@skcet.ac.in |
| CLGID | TEAMID | USERTYPE | | | USERNAME | PASSWORD | |
| c002 | t002 | NORMAL | | | normal | 222346656 | |
| c003 |  | ADMIN | | | priya | 222366456 | |
| c002 | t001 | NORMAL | | | rohit | 222346656 | |
| c002 | t001 | NORMAL | | | sachin | 222346656 | |
| c001 |  | MANAGER | | | sathya | 222346656 | |

**SAMPLE EVENTS TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EVENTID | EVENTNAME | ORGANISINGDEPT | MANAGERID | PAMOUNT | ETYPEID |
| e001 | Hacking | ECE | u005 | 500 | technical |
| e002 | Singing | MCT | u005 | 300 | non-technical |

**SAMPLE TEAM TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| TEAMID | EVENTID | TEAMSIZE | PAYMENT |
| t001 | e001 | 2 | YES |

**SAMPLE WINNER TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| TEAMID | EVENTID | TEAMSIZE | PAYMENT |
| t001 | e001 | 2 | YES |

**SAMPLE NORMAL TABLE**

|  |  |  |
| --- | --- | --- |
| USERID | USERNAME | PASSWORD |
| 1 | priya | 1234567890 |

**SAMPLE VOLUNTEER TABLE**

|  |  |  |
| --- | --- | --- |
| USERID | USERNAME | PASSWORD |
| 1 | priya | 1234567890 |

**SAMPLE MANAGER TABLE**

|  |  |  |
| --- | --- | --- |
| USERID | USERNAME | PASSWORD |
| 1 | priya | 1234567890 |

**SAMPLE MANAGER TABLE**

|  |  |  |
| --- | --- | --- |
| USERID | USERNAME | PASSWORD |
| 1 | priya | 1234567890 |

**Steps to be followed in MySQL db**

**1. Create a new database with name ‘management’**

mysql > create database management;

**2. Change to management database**

mysql> use management;  
Database changed

**3. Create four tables in management database with the following names**

i. event

ii. Organizer

iii. Volunteer

iv. Visitor

**4. Event table should contain the following attributes with one primary key**

a. Event Name

b. Date

c. Duration

d. Place

e. Website

Primary Key – Event Name

create table event (  
Event Name varchar(30) NOT NULL,  
Date date NOT NULL,  
Duration time NOT NULL,  
Place varchar(30) NOT NULL,  
Website varchar(30) NOT NULL,  
PRIMARY KEY  (Event Name));

**5. Organizer table should contain the following attributes with one foreign key**

a. Event Name

b. Organizer Name

c. Address

d. Contact No

e. Email Address

f. Role

Foreign Key – Event Name

create table organizer (  
Event Name varchar(30) NOT NULL,  
Organizer Namevarchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
Contact No int(11) NOT NULL,  
Email Address varchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(Event Name) REFERENCES event (Event Name));

**6. Volunteer table should contain the following attributes with one foreign key**

a. EventName

b. VolunteerName

c. Address

d. ContactNo

e. EmailAddress

f. Role

Foreign Key – EventName

create table volunteer (  
EventNamevarchar(30) NOT NULL,  
VolunteerNamevarchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
ContactNoint(11) NOT NULL,  
EmailAddressvarchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(EventName) REFERENCES event (EventName));

**7. Visitor table should contain the following attributes**

a. EventName

b. Visitor Name

c. Address

d. ContactNo

e. EmailAddress

f. Role

Foreign Key – EventName

create table visitor (  
EventNamevarchar(30) NOT NULL,  
VolunteerNamevarchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
ContactNoint(11) NOT NULL,  
EmailAddressvarchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(EventName) REFERENCES event (EventName));

**Note:**

1. Use **case insensitive String** comparison wherever applicable.
2. Perform **case sensitive character** comparison wherever applicable.
3. Do not include any extra instance variables or member methods in the given classes.
4. Do typecasting wherever appropriate
5. The order of passing the values to the child class constructor would be member variables of parent class followed by the child class
6. Implement Getter and Setter methods for all the variables

**Implementation Details:**

**Class: event**

Add code to event based on the class diagram. Wherever needed, the additional implementation details are given below.

* Add eventname of new event to be organized and date,duration,place, website of that particular event should be included in this class

**Class:** organizer

**Organizer name()**

* This method takes organizer name and all the details of the organizer to the database using variables

**Class: Volunteer**

* This method takes volunteer for each event with volunteer name as object. This also includes the details of the volunteer.

**Demo Class:**

* Demo class is the starter class
* Code for the Demo class is provided to you
* Read and understand the functionality of the code
* You can modify the code of Demo class for testing purpose but ensure that you are submitting it compilation error free
* Demo class will not be evaluated
* Demo class will have the option to insert/modify/delete/search customer details

**CODE:**

import cx\_Oracle

class User(object):

def \_\_init\_\_(self, id, name, gender, contact\_no, enroll\_no, college\_id, team\_id, user\_type, username, password):

self.id = id

self.name = name

self.gender = gender

self.contact\_no = contact\_no

self.enroll\_no = enroll\_no

self.college\_id = college\_id

self.team\_id = team\_id

self.user\_type = user\_type

self.username = username

self.password = password

def \_\_str\_\_(self):

print("User registered successfully\n")

class Normal(User):

def \_\_init\_\_(self):

pass

def \_\_str\_\_(self):

print("Welcome " + self.name)

def select\_event(self):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT eventid,eventname,organisingdept,managerid,pamount,etypeid from events")

temp=cur.fetchall()

print(temp)

cur.close()

con.close()

team\_id = input("Enter the Team ID : ")

event\_id = input("Enter the Event ID : ")

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("UPDATE team SET eventid='%s' WHERE teamid='%s'"%(event\_id,team\_id))

con.commit()

cur.close()

con.close()

def select\_team(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT \* from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchall()

print(temp)

cur.close()

con.close()

def payment(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT payment from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

payment = temp[0]

print(payment)

cur.close()

con.close()

class Volenteer(User):

def \_\_init\_\_(self):

pass

def \_\_str\_\_(self):

print("Welcome " + self.name + " Type: volunteer ")

def check\_participant(self, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT teamid,eventid from team")

print(cur.fetchall())

cur.close()

con.close()

#Check Team\_Details

def check\_team\_details(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT \* from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

print(temp)

cur.close()

con.close()

def update\_payment(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("UPDATE team SET payment='YES' WHERE teamid='%s'"%(team\_id))

con.commit()

cur.close()

con.close()

def update\_winner(self, winner\_id, team\_id, event\_id, pos, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("INSERT INTO winner VALUES('%s','%s','%s','%d')"%(winner\_id,eventid,teamid,pos))

con.commit()

cur.close()

con.close()

class Manager(User):

def \_\_init\_\_(self):

pass

def \_\_str\_\_(self):

print("Welcome " + self.name + " Type: manager ")

def add\_volunteer(self, user\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("UPDATE users SET usertype='VOLUNTEER' WHERE userid='%s'"%(user\_id))

con.commit()

cur.close()

con.close()

def check\_participant(self, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT teamid,eventid from team")

print(cur.fetchall())

cur.close()

con.close()

#Check Team\_Details

def check\_team\_details(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT \* from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

print(temp)

cur.close()

con.close()

#Check Payment

def check\_payment(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT payment from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

payment = temp[0]

print(payment)

cur.close()

con.close()

#Check Winner

def check\_winner(self, event\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT teamid from winner WHERE eventid='%s'"%(event\_id))

temp=cur.fetchone()

winner = temp[0]

print(winner)

cur.close()

con.close()

class Admin(User):

def \_\_init\_\_(self):

pass

def \_\_str\_\_(self):

print("Welcome " + self.name + " Type: administrator ")

def add\_manager(self, user\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("UPDATE users SET user\_type='MANAGER' WHERE userid='%s' "%(user\_id))

con.commit()

cur.close()

con.close()

def add\_volunteer(self, user\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

cur.execute("UPDATE users SET user\_type='VOLUNTEER' WHERE userid='%s'"%(user\_id))

cur.close()

con.close()

def add\_event(self, \*args, \*\*kwargs):

print ("New Event Addition")

print ("------------")

event\_id = input("Enter Event Id : ")

ename = input("Enter Event Name : ")

odept = input("Enter Organising Department[IT/CS/ECE/EEE/MCT/MECH] : ")

managerid = input("Enter Manager id : ")

pamount = int(input("Enter Payment Amount : "))

etype = input("Enter Event Type[technical/non-technical] : ")

con = cx\_Oracle.Connection("supriya/supriya10")

cur=con.cursor()

log=cur.execute("INSERT INTO events VALUES('%s','%s','%s','%s','%d','%s')"%(event\_id,ename,odept,managerid,pamount,etype))

con.commit()

print ("Event Registration Successful\n")

cur.close()

con.close()

def check\_participant(self, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT teamid,eventid from team")

print(cur.fetchall())

cur.close()

con.close()

#Check Team\_Details

def check\_team\_details(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT \* from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

print(temp)

cur.close()

con.close()

#Check Payment

def check\_payment(self, team\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT payment from team WHERE teamid='%s'"%(team\_id))

temp=cur.fetchone()

payment = temp[0]

print(payment)

cur.close()

con.close()

#Check Winner

def check\_winner(self, event\_id, \*args, \*\*kwargs):

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT teamid from winner WHERE eventid='%s'"%(event\_id))

temp=cur.fetchone()

winner = temp[0]

print(winner)

cur.close()

con.close()

#--> MAIN MENU :

print("\*"\*50)

print("\t\tEvent Management System")

print("\*"\*50)

def login():

login\_choice=int(input("1.New Registration\n2. Login\n3. Exit\nEnter Your Choice : "))

if login\_choice==1:

print ("------------")

print ("Registration")

print ("------------")

user\_id = input("Enter User Id : ")

name = input("Enter your Name : ")

gender = input("Enter Gender[male/female/other] : ")

contact = int(input("Enter Your Contact No : "))

enroll = input("Enter Your Enroll No : ")

email\_id = input("Enter Your Email Id : ")

college\_id = input("Enter Your College Id : ")

team\_id = input("Enter the Team ID : ")

user\_type = "NORMAL"

username = input("Enter the User Name : ")

password = input("Enter the Password : ")

con = cx\_Oracle.Connection("supriya/supriya10")

cur=con.cursor()

log=cur.execute("INSERT INTO users VALUES('%s','%s','%s',%d,'%s','%s','%s','%s','%s','%s','%s')"%(user\_id,name,gender,contact,enroll,email\_id,college\_id,team\_id,user\_type,username,password))

con.commit()

print ("Registration Successful\n")

cur.close()

con.close()

login()

if login\_choice==2:

username = input("Enter your username : ")

password = input("Enter the password : ")

con=cx\_Oracle.connect("supriya/supriya10")

cur=con.cursor()

log=cur.execute("SELECT \* from users WHERE username ='%s' and password='%s'"%(username,password))

if (len(log.fetchall())):

print("Login success\n")

else:

print("You are not a registered user\n")

cur.execute("select usertype from users where username = '%s'"%(username))

temp=cur.fetchone()

user\_type = temp[0]

cur.close()

con.close()

''' Normal User functions '''

if user\_type == "NORMAL":

print("1. Select Event\n2. Display Team\n3. Payment\nEnter Your Choice : ")

choice = int(input())

obj = Normal()

if choice == 1:

obj.select\_event()

elif choice == 2:

team\_id = input("Enter the Team ID : ")

obj.select\_team(team\_id)

elif choice == 3:

team\_id = input("Enter the Team ID : ")

obj.payment(team\_id)

else:

print("Invalid Option\n")

elif user\_type == "VOLENTEER":

print("1. Participant list\n2. Team details\n3. Update Payment\n4. Update Winner\nEnter Your Choice : ")

choice = int(input())

obj=Volenteer()

if choice == 1:

obj.check\_participant()

elif choice == 2:

team\_id = input("Enter the Team ID : ")

obj.check\_team\_details(team\_id)

elif choice == 3:

team\_id = input("Enter the Team ID : ")

obj.update\_payment(team\_id)

elif choice == 4:

winner\_id= input("Enter the Winner ID : ")

team\_id = input("Enter the Team ID : ")

event\_id = input("Enter the Event ID : ")

pos = int(input("Enter the winning position[1/2/3]"))

obj.update\_winner(winner\_id, team\_id, event\_id, pos)

else:

print("Invalid Option.....Try Again\n")

elif user\_type == "MANAGER":

print("1. Add Volunteer\n2. Check Participant\n3. Team Details\n4. Check Payment\n5.Winner Details\nEnter Your Choice : ")

choice = int(input())

obj = Manager()

if choice == 1:

user\_id = input("Enter the User ID : ")

obj.add\_volunteer(user\_id)

elif choice == 2:

obj.check\_participant()

elif choice == 3:

team\_id = input("Enter the Team ID : ")

obj.check\_team\_details(team\_id)

elif choice == 4:

team\_id = input("Enter the Team ID : ")

obj.check\_payment(team\_id)

elif choice == 5:

event\_id = input("Enter the Event ID : ")

obj.check\_winner(event\_id)

else:

print("Invalid Option.....Try Again\n")

elif user\_type == "ADMIN":

print("1. Add Manager\n2. Add Volunteer\n3. Add Event\n4. Check Participant\n5. Team Details\n6. Check Payment\n7. Check Winner\nEnter Your Choice : ")

choice = int(input())

obj = Admin()

if choice == 1:

user\_id = input("Enter the User ID : ")

obj.add\_manager(user\_id)

elif choice == 2:

user\_id = input("Enter the User ID : ")

obj.add\_volunteer(user\_id)

elif choice == 3:

obj.add\_event()

elif choice == 4:

obj.check\_participant()

elif choice == 5:

team\_id = input("Enter the Team ID : ")

obj.check\_team\_details(team\_id)

elif choice == 6:

team\_id = input("Enter the Team ID : ")

obj.check\_payment(team\_id)

elif choice == 7:

event\_id = input("Enter the Event ID : ")

obj.check\_winner(event\_id)

else:

print("Invalid Option........Try Again\n")

login()

if login\_choice == 3:

print("Thank You\n")

''' Main function here '''

login()

**SCRIPT:**

drop table users;

drop table team;

drop table event;

drop table winner;

Drop table normal;

drop table volunteer;

drop table manager;

drop table admin;

Create table users(userid varchar2(20) primary key,name varchar2(25) not null,gender varchar2(25) not null check(Gender in('male','female','other')),contactno number not null UNIQUE check(length(Contactno)=10),enrollno varchar2(10) not null,emailid varchar2(25) not null UNIQUE,clgid varchar2(25) not null,teamid varchar2(10) default null,usertype varchar2(25) not null check(usertype in ('NORMAL','VOLUNTEER','MANAGER','ADMIN')),username varchar2(25) not null UNIQUE,password varchar2(10) not null check(length(password)>=8));

Create table team(teamid varchar2(10) primary key,eventid varchar2(10),foreign key(eventid) references event(eventid),teamsize number(1) not null check(teamsize<=5),payment varchar2(5) not null check(payment in ('YES','NO')));

Create table event(eventid varchar2(20) primary key,eventname varchar2(25) not null,organisingdept varchar2(10) not null check(organisingdept in ('IT','CS','ECE','EEE','MCT','MECH') ),managerid varchar2(20) not null,Pamount number(3) not null,etypeid varchar2(20) not null check(etypeid in('technical','non-technical')));

Create table winner(WinnerId varchar2(10) primary key,eventid varchar2(20) not null,foreign key(eventid) references event(eventid),teamid varchar2(10) not null,foreign key(teamid) references team(teamid),position number not null check(position in(1,2,3)));

Create table normal(userid varchar2(20) primary key,username varchar2(25)not null UNIQUE,password varchar2(10)not null check(length(password)>=8));

Create table volunteer(userid varchar2(20) primary key,username varchar2(25)not null UNIQUE,password varchar2(10)not null check(length(password)>=8));

Create table manager(userid varchar2(20) primary key,username varchar2(25)not null UNIQUE,password varchar2(10)not null check(length(password)>=8));

Create table admin(userid varchar2(20) primary key,username varchar2(25)not null UNIQUE,password varchar2(10)not null check(length(password)>=8));

insert into users values('uS001','priyanka','female',9876543210,'en001','16eucs100@skcet.ac.in','c003','','ADMIN','priyanka','12345678');

insert into users values('uS002','rohit','male',1234567890,'en002','16euit095@skcet.ac.in','c002','t001','NORMAL','rohit','12345678');

insert into users values('uS003','ram','male',1357986420,'en003','16euee189@skcet.ac.in','c001','','VOLUNTEER','ram','12345678');

insert into users values('uS004','suresh','male',1256903478,'en004','16tuec048@skct.edu.in','c002','t001','NORMAL','suresh','12345678');

insert into users values('uS005','sathya','female',9753124680,'en005','16tuee166@skct.edu.in','c001','','MANAGER','sathya','12345678');

insert into event values('e001','hacking','IT','u005',500,'technical');

insert into event values('e002','singing','ECE','u005',300,'non-technical');

insert into team values('t001','e001',2,'YES');

insert into winner values('w001','e001','t001',1);

select \* from users;

select \* from event;

select \* from team;

select \* from winner;

insert into Normal values(1,'priya','1234567890');

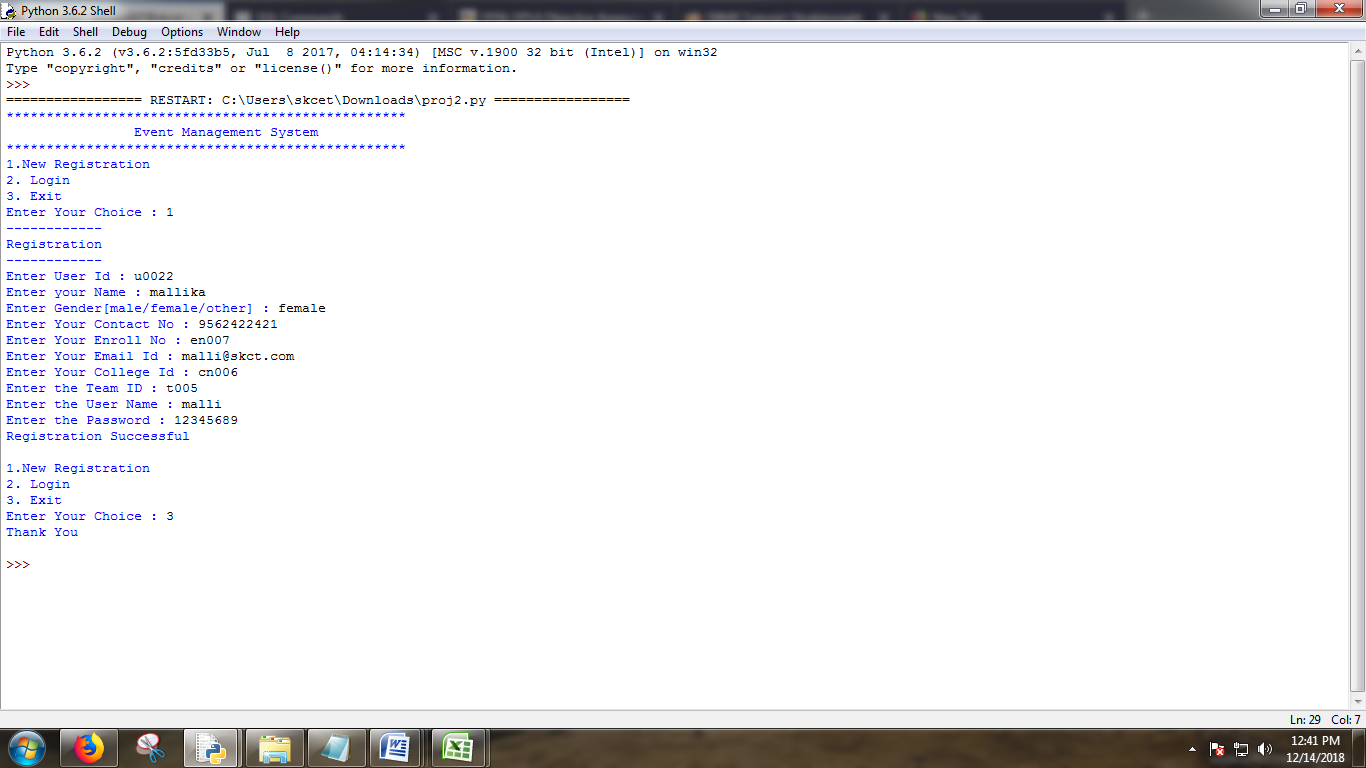
insert into Volunteer values(1,'priya','1234567890');

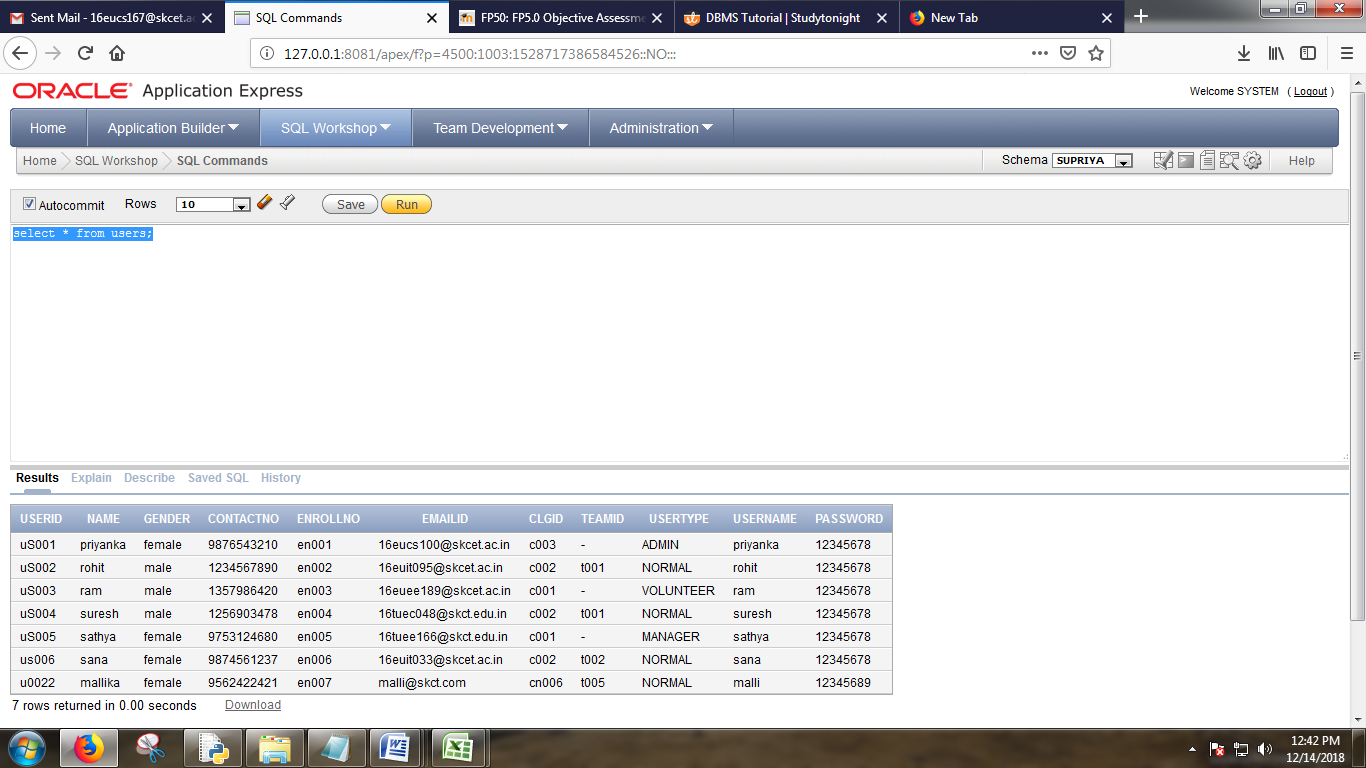
insert into Manager values(1,'priya','1234567890');

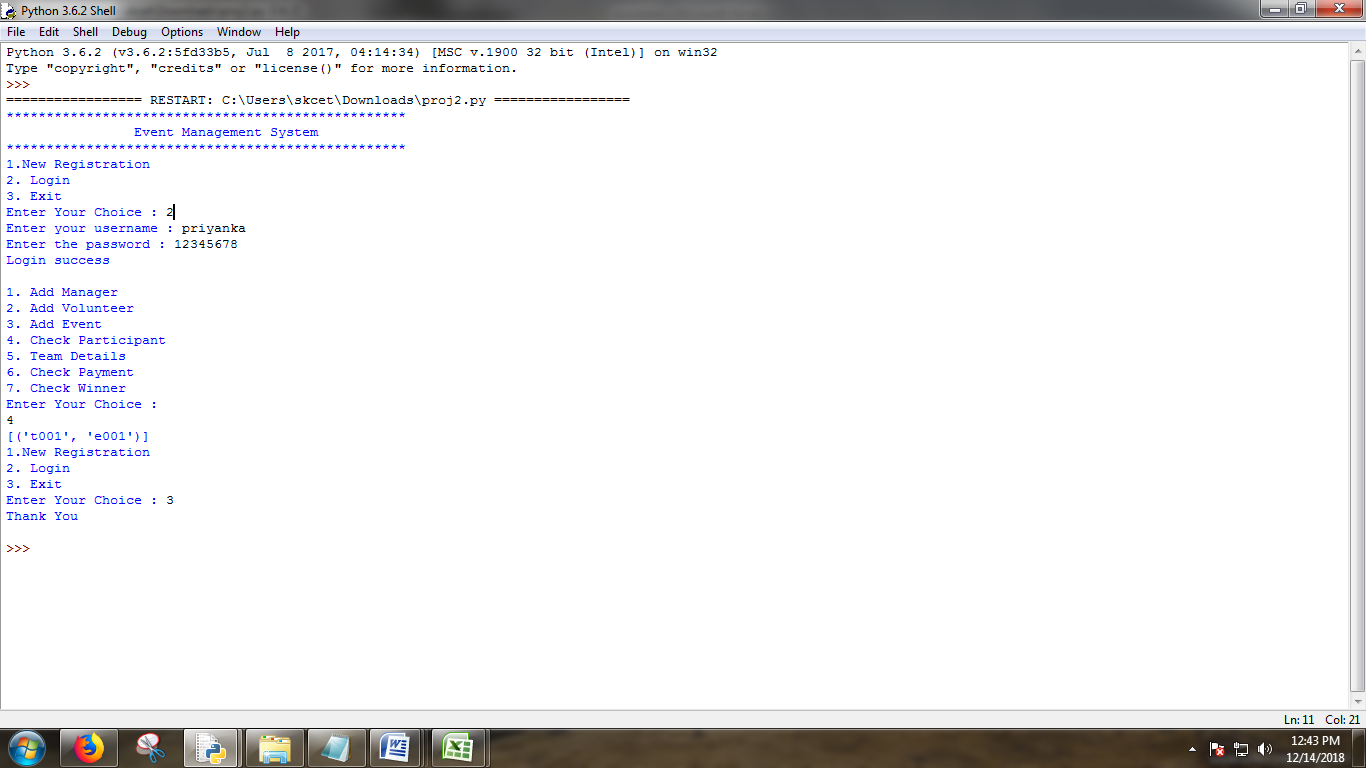
DESC Normal;

DESC User1;

**OUTPUT SCREENSHOTS:**

****

****

****

**CONCLUSION:**

The “Event Management” was successfully designed. During this project we have accomplished all the objectives and this project meets the needs of the organization. The developed will be used in searching, retrieving and generating information for the concerned requests.

**GOALS ACHIVIED**

* Reduced entry work
* Easy retrieval of information
* Reduced errors due to human intervention
* User friendly screens to enter the data
* Portable and flexible for further enhancement
* Web enabled.
* Fast finding of information reques