A Project Report On

Hackathon Management System

DEVELOPED BY:

IT027 – SHAILY FADADU
IT037 – HIMANSHI THAKKAR
IT021 – DEVANSHI PATEL

Guided By
Internal Guide:
Prof. Ravindra Vyas

Department of Information Technology Faculty of Technology
DD University

Department of Information Technology Faculty of Technology,
Dharmsinh Desai University College Road, Nadiad-387001
April-2024

CERTIFICATE

This is to certify that the project entitled “Hackathon Management system” is a bonafide report of the work carried out by

1) Shaily Fadadu Student ID No: 22ITUOS150
2) Himanshi Thakkar Student ID No: 23ITUOD013
3) Devanshi Patel Student ID No: 22ITUON101 of

Department of Information Technology, semester IV, under the guidance and supervision for the subject Database Management System. They were involved in Project training during the academic year 2023-2024.

Prof. Ravindra Vyas
Project Guide, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:24/03/2024

Prof. Vipul Dabhi
Head, Department of Information Technology

INDEX

1) SYSTEM OVERVIEW

A) System Name: Hackathon Manager

Hackathons are events where individuals or teams collaborate intensively on software or hardware projects over a defined period, often leading to innovative solutions and prototypes.

Purpose: To streamline the organization and management of hackathon events, including participant registration, team formation, judge assignment, prize distribution, and sponsor management.

Key Features:

Participant Management:

User profiles: Store basic information about participants, including name, email, and hackathon history.

Team formation: Allow participants to create teams, invite others, and manage team rosters.

Registration: Handle the registration process, including collecting participant details and hackathon preferences.

Judge Management:

Judge profiles: Store information about judges, including expertise and availability.

Assignment: Automatically assign judges to hackathon events based on their expertise and availability.

Feedback collection: Facilitate the collection of judge feedback on participants and teams.

Prize Management:

Prize details: Store information about available prizes, including category, sponsor, and prize value.

Distribution: Track prize assignments and ensure all teams receive their appropriate prizes.

Sponsor Management:

Sponsor profiles: Store information about sponsors, including company name, website, and sponsorship level.

Sponsorship tracking: Track sponsor involvement, including logos, booth assignments, and promotional materials.

Event Management:
Schedule: Manage the event schedule, including hackathon start and end times, judging windows, and prize distribution times.
Communication: Send email notifications to participants, judges, sponsors, and organizers about important event details.
Registration: Handle on-site registration for walk-in participants.

Scalability: The system is designed to scale horizontally and handle large-scale hackathon events with thousands of participants, judges, and sponsors.

B) ADVANTAGES OF THE PROPOSED SYSTEM

1) Efficiency: Automates many administrative tasks, saving organizers time and effort. Judges can easily assign scores, and sponsors can track their involvement in real-time.

2) Centralized Management: Keeps all hackathon details in one place, making it easy for organizers to oversee the event and communicate with participants, judges, and sponsors.

3) Improved Judge Experience: Streamlines the judge assignment and feedback collection process, ensuring fair evaluations. Judges can focus on providing constructive feedback rather than administrative tasks.

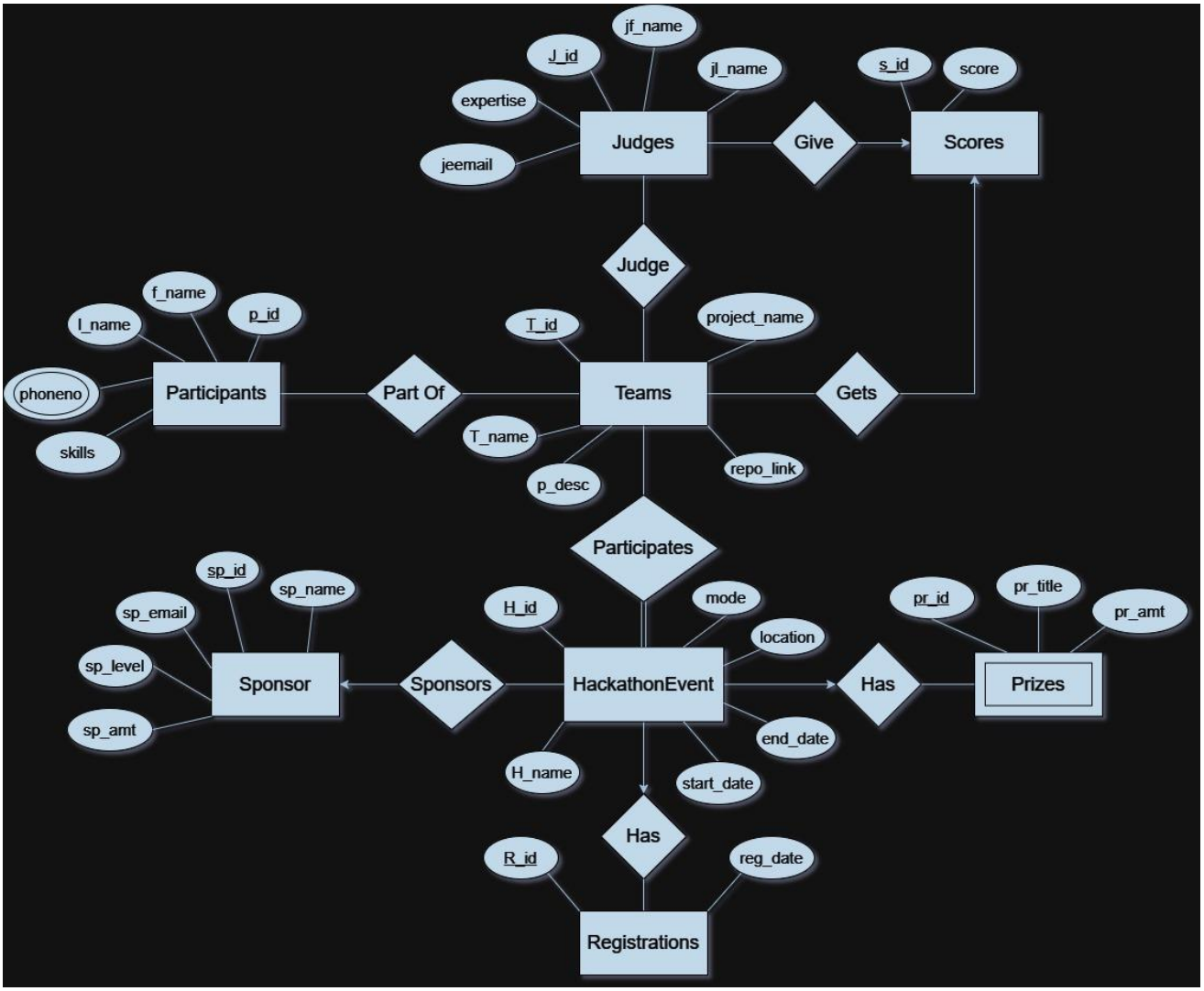
4) Scalability: Designed to handle large-scale hackathon events with thousands of participants, judges, and sponsors, making it a flexible solution for events of all sizes.

5) User-Friendly Interface: Clear workflows and an intuitive interface make it easy for organizers, judges, sponsors, and participants to use the system.

6) Security: Robust security measures protect sensitive data like participant information and judge feedback, building trust with users.

Overall, Hackathon Manager aims to significantly reduce the administrative burden of organizing hackathon events, allowing organizers to focus more on creating a rewarding and inclusive experience for participants. By automating many of the logistical details, the system enables a smoother, more efficient event management process.

2) ENTITY-RELATIONSHIP MODEL



3) Data Dictionary

3.1 Teams

```
postgres=# \d teams
                Table "public.teams"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 t_id         | character varying(3)   | not null
 t_name       | character varying(25)  | not null
 project_name | character varying(25)  | not null
 repo_link    | character varying(300) | not null
 p_desc       | character varying(200) |
Indexes:
    "teams_pkey" PRIMARY KEY, btree (t_id)
Referenced by:
    TABLE "participants" CONSTRAINT "participants_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
    TABLE "registration" CONSTRAINT "registration_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
    TABLE "scores" CONSTRAINT "scores_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
```

3.2 HackathonEvent

```
postgres=# \d HackathonEvent
         Table "public.hackathonevent"
   Column   |          Type          | Modifiers
------------+------------------------+-----------
 h_id       | character varying(3)   | not null
 h_name     | character varying(25)  | not null
 start_date | date                   | not null
 end_date   | date                   | not null
 location   | character varying(50)  |
 mode       | character varying(20)  | not null
Indexes:
    "hackathonevent_pkey" PRIMARY KEY, btree (h_id)
Referenced by:
    TABLE "prizes" CONSTRAINT "prizes_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
    TABLE "registration" CONSTRAINT "registration_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
    TABLE "scores" CONSTRAINT "scores_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
```

3.3 Participants

```
postgres=# \d participants
        Table "public.participants"
 Column  |          Type          | Modifiers
---------+------------------------+-----------
 p_id    | character varying(3)   | not null
 f_name  | character varying(25)  |
 l_name  | character varying(25)  |
 phoneno | numeric(10,0)          |
 skills  | character varying(100) |
 t_id    | character varying(3)   |
Indexes:
    "participants_pkey" PRIMARY KEY, btree (p_id)
Foreign-key constraints:
    "participants_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
Referenced by:
    TABLE "registration" CONSTRAINT "registration_p_id_fkey" FOREIGN KEY (p_id) REFERENCES participants(p_id)
```

3.4 Registration

```
postgres=# \d Registration
         Table "public.registration"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 r_id     | character varying(3) | not null
 reg_date | date                  | not null
 p_id     | character varying(3) |
 t_id     | character varying(3) |
 h_id     | character varying(3) |
Indexes:
    "registration_pkey" PRIMARY KEY, btree (r_id)
Foreign-key constraints:
    "registration_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
    "registration_p_id_fkey" FOREIGN KEY (p_id) REFERENCES participants(p_id)
    "registration_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
```

3.5 Judges

```
postgres=# \d Judges
            Table "public.judges"
  Column   |          Type          | Modifiers
-----------+------------------------+-----------
 j_id      | character varying(3)   | not null
 jf_name   | character varying(25)  | not null
 jl_name   | character varying(25)  | not null
 expertize | character varying(100) |
 jemail    | character varying(50)  |
Indexes:
    "judges_pkey" PRIMARY KEY, btree (j_id)
Referenced by:
    TABLE "scores" CONSTRAINT "scores_j_id_fkey" FOREIGN KEY (j_id) REFERENCES judges(j_id)
```

3.6 Prizes

```
postgres=# \d Prizes
            Table "public.prizes"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 pr_id    | character varying(3)  | not null
 pr_title | character varying(25) | not null
 pr_amt   | numeric(6,0)          | not null
 sp_id    | character varying(3)  |
 h_id     | character varying(3)  |
Indexes:
    "prizes_pkey" PRIMARY KEY, btree (pr_id)
Foreign-key constraints:
    "prizes_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
    "prizes_sp_id_fkey" FOREIGN KEY (sp_id) REFERENCES sponsor(sp_id)
```

3.7 Sponsor

```
postgres=# \d Sponsor
           Table "public.sponsor"
  Column  |         Type          | Modifiers
----------+-----------------------+-----------
 sp_id    | character varying(3)  | not null
 sp_name  | character varying(50) | not null
 sp_email | character varying(50) | not null
 sp_level | character varying(10) | not null
 sp_amt   | numeric(6,0)          |
 h_id     | character varying(3)  |
Indexes:
    "sponsor_pkey" PRIMARY KEY, btree (sp_id)
Foreign-key constraints:
    "sponsor_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
Referenced by:
    TABLE "prizes" CONSTRAINT "prizes_sp_id_fkey" FOREIGN KEY (sp_id) REFERENCES sponsor(sp_id)
```

3.8 Scores

```
postgres=# \d Scores
           Table "public.scores"
 Column |         Type          | Modifiers
--------+-----------------------+-----------
 s_id   | character varying(3) | not null
 h_id   | character varying(3) |
 t_id   | character varying(3) |
 j_id   | character varying(3) |
 score  | numeric(2,1)         |
Indexes:
    "scores_pkey" PRIMARY KEY, btree (s_id)
Foreign-key constraints:
    "scores_h_id_fkey" FOREIGN KEY (h_id) REFERENCES hackathonevent(h_id)
    "scores_j_id_fkey" FOREIGN KEY (j_id) REFERENCES judges(j_id)
    "scores_t_id_fkey" FOREIGN KEY (t_id) REFERENCES teams(t_id)
```

4) DATA IMPLEMENTATION

4.1) SCHEMA

4.1.1 Teams

CREATE TABLE teams (T_id varchar(3) PRIMARY KEY, T_name varchar(25) NOT NULL, project_name varchar(25) NOT NULL, repo_link varchar(300) NOT NULL, p_desc varchar(200));

4.1.2 HackathonEvent

```
CREATE TABLE HackathonEvent (
H_id varchar(3) PRIMARY KEYy,
H_name varchar(25) NOT NULL,
start_date date NOT NULL,
end_date date NOT NULL,
location varchar(50),
mode varchar(20) NOT NULL);
```

4.1.3 participants

```
CREATE TABLE participants(
p_id varchar(3) PRIMARY KEY,
f_name varchar(25),
l_name varchar(25),
phoneno numeric(10),
skills varchar(100),
T_id varchar(3) REFERENCES teams(T_id));
```

4.1.4 Registration

```
CREATE TABLE Registration(
R_id varchar(3) PRIMARY KEY,
reg_date date NOT NULL,
p_id varchar(3) REFERENCES participants(p_id),
```

T_id varchar(3) REFERENCES teams(T_id),
H_id varchar(3) REFERENCES HackathonEvent(H_id));

4.1.5 Judges

CREATE TABLE Judges(J_id varchar(3) PRIMARY KEY, jf_name varchar(25) NOT NULL, jl_name varchar(25) NOT NULL, Expertize varchar(100));

4.1.6 Prizes

CREATE Prizes(
pr_id varchar(3) PRIMARY KEY,
pr_title varchar(25) NOT NULL,
pr_amt numeric(6) NOT NULL,
Sp_id varchar(3) REFERENCES Sponsor(Sp_id),
H_id varchar(3) REFERENCES HackathonEvent(H_id));

4.1.7 Sponsor

CREATE TABLE Sponsor(
Sp_id varchar(3) PRIMARY KEY,
sp_name varchar(50) NOT NULL,
sp_email varchar(50) NOT NULL,
sp_level varchar(10) NOT NULL,
sp_amt numeric(6),
H_id varchar(3) REFERENCES HackathonEvent(H_id));

4.1.8 Scores

CREATE TABLE Scores(
s_id varchar(3) PRIMARY KEY,
H_id varchar(3) REFERENCES HackathonEvent(H_id),
T_id varchar(3) REFERENCES teams(T_id),
J_id varchar(3) REFERENCES Judges(J_id));

4.2) DATA INSERTION

4.2.1 teams

```
INSERT INTO teams values ('T01', 'Hacksquad', 'Cypher Link',
'https://github.com/HIMANSHIthakkar/himanshi', 'Empowering Indie
Developers Across Blockchains');
INSERT INTO teams values
('T02','ByteBattles','AlumInsights','https://github.com/HIMANSHIthakkar/himan
shi','Alumni Insights for Placement Preparations');
INSERT INTO teams values
('T03','HackHeroes','Gitsume','https://github.com/shailifadadu','Crafting the
perfect resume just got easier');
INSERT INTO teams values
('T04','WebWarriors','InterviewMap','https://github.com/shailifadadu','Crafting
the perfect resume just got easier');
```

4.2.2 Participants

```
INSERT INTO participants
values('P01','Himanshi','Thakkar',8780230308,'c++,java','T01');
INSERT INTO participants
values('P02','Srushti','Gol',8780230308,'c++,java','T01');
INSERT INTO participants
values('P03','Manan','Patel',8780230208,'Dbms,python','T01');
INSERT INTO participants
values('P04','Isha','Jagani',8780230209,'React,C','T01');
INSERT INTO participants values('P05','Mansi','Shah',8780222209,'Node
js,js','T02');
INSERT INTO participants
values('P06','Shyam','Shah',9780222209,'sql,js,React','T02');
INSERT INTO participants values('P07','Mann','Jain',9730220209,'next
js,React','T02');
INSERT INTO participants
values('P08','Shaily','Fadadu',9730223209,'AWS,Docker','T03');
INSERT INTO participants
values('P09','Neha','Patel',9730123209,'HTML,CSS,JS','T03');
INSERT INTO participants values('P10','Meet','soni',9730123289,'React, next
JS','T03');
```

```
INSERT INTO participants values('P11','Disha','Modi',9730893289,'AI/ML,node JS','T03');
INSERT INTO participants values('P12','Devanshi','Patel',9830893289,'React,node JS','T04');
INSERT INTO participants values('P13','kunj','Patel',9830593289,'GCP,mernstack','T04');
INSERT INTO participants values('P14','krish','Desai',9830593279,'AI/ML,mongo DB','T04');
```

4.2.3 HackathonEvent

```
INSERT INTO HackathonEvent values('H01','DUHacks','24/02/2024','26/02/2024','ddu college','offline');

INSERT INTO HackathonEvent values('H02','Hack&Chill','16/01/2024','17/01/2024','ddu college','offline');

INSERT INTO HackathonEvent values('H03','HackNITR','8/03/2024','10/03/2024','none','online');

INSERT INTO HackathonEvent values('H04','DataThrone','7/04/2024','8/04/2024','none','online');
```

4.2.4 Judges

```
INSERT INTO judges (j_id, jf_name, jl_name, expertise, jemail)
VALUES ('J01', 'Juhi', 'Jaiswal', 'AI/ML', 'juhi890@gmail.com'),
    ('J02', 'Harsh', 'Thakkar', 'Web', 'hthakkar@gmail.com');
```

4.2.5 Sponsor

```
INSERT INTO Sponsor (sp_id, sp_name, sp_email, sp_level, sp_amt, h_id)
VALUES ('S01','Apex Education', 'hthakkar374@gmail.com', 'Bronze', 50000, 'H01'),
    ('S02','Sucess overseas', 'hthakkar374@gmail.com', 'Diamond', 150000, 'H01'),
    ('S03','Jivan vikas Education', 'hthakkar374@gmail.com', 'Silver', 55000, 'H02');
```

4.2.6 Scores

```
INSERT INTO Scores (s_id, h_id, t_id, score)
VALUES ('S01', 'H01', 'T01', 'J01', '4.0'),
    ('S02', 'H01', 'T02', 'J01', '3.0'),
    ('S03', 'H02', 'T01', 'J02', '4.0');
```

4.2.7 Prizes

```
INSERT INTO prizes (pr_id, pr_title, pr_amt, sp_id, h_id)
VALUES ('Z01','1st', 8000, 'S01', 'H01'),
    ('Z02','2nd', 8000, 'S01', 'H02'),
    ('Z03','3rd', 8000, 'S01', 'H01'),
    ('Z04','1st con', 8000, 'S02', 'H03');
```

4.2.8 Registration

```
INSERT INTO registration values('R01','1/3/2024','P01','T01','H01');
INSERT INTO registration values('R02','2/3/2024','P02','T01','H01');
INSERT INTO registration values('R03','2/3/2024','P03','T01','H01');
INSERT INTO registration values('R04','3/3/2024','P04','T01','H01');
INSERT INTO registration values('R05','3/3/2024','P05','T02','H01');
INSERT INTO registration values('R06','4/3/2024','P06','T02','H01');
INSERT INTO registration values('R07','4/3/2024','P07','T02','H01');
INSERT INTO registration values('R08','4/3/2024','P08','T03','H01');
INSERT INTO registration values('R09','4/3/2024','P09','T03','H01');
INSERT INTO registration values('R10','4/3/2024','P10','T03','H01');
INSERT INTO registration values('R11','4/3/2024','P11','T03','H01');
INSERT INTO registration values('R12','4/3/2024','P12','T04','H01');
INSERT INTO registration values('R13','4/3/2024','P13','T04','H01');
INSERT INTO registration values('R14','4/3/2024','P14','T04','H01');
```

INSERTION OUTPUT:

4.2.1 Teams

```
postgres=# select * from teams;
 t_id |   t_name    | project_name |                repo_link                |                  p_desc
------+-------------+--------------+-----------------------------------------+-------------------------------------------
 T01  | Hacksquad   | Cypher Link  | https://github.com/HIMANSHIthakkar/himanshi | Empowering Indie Developers Across Blockchains
 T02  | ByteBattles | AlumInsights | https://github.com/HIMANSHIthakkar/himanshi | Alumni Insights for Placement Preparations
 T03  | HackHeroes  | Gitsume      | https://github.com/shailifadadu             | Crafting the perfect resume just got easier
 T04  | WebWarriors | InterviewMap | https://github.com/shailifadadu             | Crafting the perfect resume just got easier
(4 rows)
```

4.2.2 Participants

```
postgres=# select * from participants;
 p_id |  f_name  | l_name  |  phoneno   |       skills       | t_id
------+----------+---------+------------+--------------------+------
 P01  | Himanshi | Thakkar | 8780230308 | c++,java           | T01
 P02  | Srushti  | Gol     | 8780230308 | c++,java           | T01
 P03  | Manan    | Patel   | 8780230208 | Dbms,python        | T01
 P04  | Isha     | Jagani  | 8780230209 | React,C            | T01
 P05  | Mansi    | Shah    | 8780222209 | Node js,js         | T02
 P06  | Shyam    | Shah    | 9780222209 | sql,js,React       | T02
 P07  | Mann     | Jain    | 9730220209 | next js,React      | T02
 P08  | Shaily   | Fadadu  | 9730223209 | AWS,Docker         | T03
 P09  | Neha     | Patel   | 9730123209 | HTML,CSS,JS        | T03
 P10  | Meet     | soni    | 9730123289 | React, next     JS | T03
 P11  | Disha    | Modi    | 9730893289 | AI/ML,node JS      | T03
 P12  | Devanshi | Patel   | 9830893289 | React,node JS      | T04
 P13  | kunj     | Patel   | 9830593289 | GCP,mernstack      | T04
 P14  | krish    | Desai   | 9830593279 | AI/ML,mongo DB     | T04
(14 rows)
```

4.2.3 HackathonEvent

```
postgres=# select *from HackathonEvent;
 h_id |   h_name   | start_date |  end_date  |  location   |  mode
------+------------+------------+------------+-------------+---------
 H01  | DUHacks    | 2024-02-24 | 2024-02-26 | ddu college | offline
 H02  | Hack&Chill | 2024-01-16 | 2024-01-17 | ddu college | offline
 H03  | HackNITR   | 2024-03-08 | 2024-03-10 | none        | online
 H04  | DataThrone | 2024-04-07 | 2024-04-08 | none        | online
(4 rows)
```

4.2.4 Judges

```
postgres=# select * from judges;
 j_id | jf_name | jl_name | expertize |       jemail
------+---------+---------+-----------+--------------------
 J01  | Juhi    | Jaiswal | AI/ML     | juhi890@gmail.com
 J02  | Harsh   | Thakkar | Web       | hthakkar@gmail.com
(2 rows)
```

4.2.5 Sponsor

```
postgres=# select * from sponsor;
 sp_id |        sp_name        |       sp_email        | sp_level | sp_amt | h_id
-------+-----------------------+-----------------------+----------+--------+------
 S01   | Apex Education        | hthakkar374@gmail.com | bronze   |  50000 | H01
 S02   | Sucess overseas       | hthakkar374@gmail.com | Diamond  | 150000 | H01
 S03   | Jivan vikas Education | hthakkar374@gmail.com | Silver   |  55000 | H02
(3 rows)
```

4.2.6 Scores

```
postgres=# select* from scores;
 s_id | h_id | t_id | j_id | score
------+------+------+------+-------
 S01  | H01  | T01  | J01  |   4.0
 S02  | H01  | T02  | J01  |   3.0
 S03  | H02  | T01  | J02  |   4.0
(3 rows)
```

4.2.7 Prizes

```
postgres=# select* from prizes;
 pr_id | pr_title | pr_amt | sp_id | h_id
-------+----------+--------+-------+------
 Z01   | 1st      |   8000 | S01   | H01
 Z02   | 2st      |   8000 | S01   | H02
 Z03   | 3rd      |   8000 | S01   | H01
 Z04   | 1st con  |   8000 | S02   | H03
(4 rows)
```

4.2.8 Registration

```
postgres=# select * from registration;
 r_id |  reg_date  | p_id | t_id | h_id
------+------------+------+------+------
 R01  | 2024-03-01 | P01  | T01  | H01
 R02  | 2024-03-02 | P02  | T01  | H01
 R03  | 2024-03-02 | P03  | T01  | H01
 R04  | 2024-03-03 | P04  | T01  | H01
 R05  | 2024-03-03 | P05  | T02  | H01
 R06  | 2024-03-04 | P06  | T02  | H01
 R07  | 2024-03-04 | P07  | T02  | H01
 R08  | 2024-03-04 | P08  | T03  | H01
 R09  | 2024-03-04 | P09  | T03  | H01
 R10  | 2024-03-04 | P10  | T03  | H01
 R11  | 2024-03-04 | P11  | T03  | H01
 R12  | 2024-03-04 | P12  | T04  | H01
 R13  | 2024-03-04 | P13  | T04  | H01
 R14  | 2024-03-04 | P14  | T04  | H01
(14 rows)
```

4.3) QUERIES USING BASIC DBMS CONSTRUCTS JOIN & SUBQUERIES:

4.3.1 Display Participants First name and Last name

```
postgres=# select f_name,l_name from participants;
  f_name   | l_name
-----------+---------
 Himanshi  | Thakkar
 Srushti   | Gol
 Manan     | Patel
 Isha      | Jagani
 Mansi     | Shah
 Shyam     | Shah
 Mann      | Jain
 Shaily    | Fadadu
 Neha      | Patel
 Meet      | soni
 Disha     | Modi
 Devanshi  | Patel
 kunj      | Patel
 krish     | Desai
 yash      | soni
 yash      | soni
(16 rows)
```

4.3.2 Find all the offline Hackathon Event and their corresponding sponsors

```
postgres=# select * from Hackathonevent h  inner join sponsor s on h.h_id=s.h_id;
 h_id |   h_name    | start_date |  end_date  |  location   |  mode   | sp_id |        sp_name        |       sp_email        | sp_level | sp_amt | h_id
------+-------------+------------+------------+-------------+---------+-------+-----------------------+-----------------------+----------+--------+------
 H01  | DUHacks     | 2024-02-24 | 2024-02-26 | ddu college | offline | S01   | Apex Education        | hthakkar374@gmail.com | bronze   |  50000 | H01
 H01  | DUHacks     | 2024-02-24 | 2024-02-26 | ddu college | offline | S02   | Sucess overseas       | hthakkar374@gmail.com | Diamond  | 150000 | H01
 H02  | Hack&Chill  | 2024-01-16 | 2024-01-17 | ddu college | offline | S03   | Jivan vikas Education | hthakkar374@gmail.com | Silver   |  55000 | H02
(3 rows)
```

4.3.3 Display the names of the participants and the teams they belong to, along with the project names

```
postgres=# select F_name,t_name,project_name from participants p inner join teams t on p.T_id=t.T_id;
  f_name  |   t_name    | project_name
----------+-------------+--------------
 Himanshi | Hacksquad   | Cypher Link
 Srushti  | Hacksquad   | Cypher Link
 Manan    | Hacksquad   | Cypher Link
 Isha     | Hacksquad   | Cypher Link
 Mansi    | ByteBattles | AlumInsights
 Shyam    | ByteBattles | AlumInsights
 Mann     | ByteBattles | AlumInsights
 Shaily   | HackHeroes  | Gitsume
 Neha     | HackHeroes  | Gitsume
 Meet     | HackHeroes  | Gitsume
 Disha    | HackHeroes  | Gitsume
 Devanshi | WebWarriors | InterviewMap
 kunj     | WebWarriors | InterviewMap
 krish    | WebWarriors | InterviewMap
(14 rows)
```

4.3.4 Display the scores for each project and the descriptions of those projects

```
postgres=# select project_name,t_name,p_desc,score from teams t inner join scores s on t.t_id=s.t_id;
 project_name |   t_name    |                    p_desc                    | score
--------------+-------------+----------------------------------------------+-------
 Cypher Link  | Hacksquad   | Empowering Indie Developers Across Blockchains |   4.0
 AlumInsights | ByteBattles | Alumni Insights for Placement Preparations     |   3.0
 Cypher Link  | Hacksquad   | Empowering Indie Developers Across Blockchains |   4.0
(3 rows)
```

4.3.5 Display the scores given by judges to the specific projects along with their Team Name

```
postgres=# select jf_name,t_name,project_name,score from judges j inner join scores s  on j.j_id=s.j_id inner join teams t on t.t_id=s.t_id;
 jf_name |   t_name    | project_name | score
---------+-------------+--------------+-------
 Juhi    | Hacksquad   | Cypher Link  |   4.0
 Juhi    | ByteBattles | AlumInsights |   3.0
 Harsh   | Hacksquad   | Cypher Link  |   4.0
(3 rows)
```

4.4) FUNCTION & TRIGGERS:

4.4.1 Create trigger on participants table so that on insertion, it should be reflected in dummy_participants.

Function:

```
CREATE OR REPLACE FUNCTION test1()
  RETURNS trigger AS
$$
BEGIN
INSERT INTO  dummy_participants
VALUES(old.p_id,old.f_name,old.l_name,old.phoneno,old.skills,old.T_id,curren
t_date);
   RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
```

Trigger:

```
CREATE TRIGGER test_trigger1
AFTER INSERT
ON  dummy_participants
FOR EACH ROW
EXECUTE PROCEDURE test1();
```

```
postgres=# CREATE OR REPLACE FUNCTION test()
postgres-#   RETURNS trigger AS
postgres-# $$
postgres$# BEGIN
postgres$#          INSERT INTO  dummy_participants
postgres$#          VALUES(NEW.p_id,NEW.f_name,NEW.l_name,NEW.phoneno,NEW.skills,NEW.T_id,current_date);
postgres$#
postgres$#     RETURN NEW;
postgres$# END;
postgres$# $$
postgres-# LANGUAGE 'plpgsql';
CREATE FUNCTION
postgres=# CREATE TRIGGER test_trigger
postgres-#   AFTER INSERT
postgres-#   ON  dummy_participants
postgres-#   FOR EACH ROW
postgres-#   EXECUTE PROCEDURE test();
CREATE TRIGGER
```

4.5) CURSOR:

Create Cursor to display all the participants participating in Hackathon Event.

CURSOR SYNTAX:

```
CREATE OR REPLACE FUNCTION process_participants()
RETURNS void AS $$
DECLARE
  participant_cursor CURSOR FOR
    SELECT *
    FROM participants;
  participant participants%ROWTYPE;
BEGIN
  OPEN participant_cursor;
  LOOP
    FETCH participant_cursor INTO participant;
    EXIT WHEN NOT FOUND;
    RAISE NOTICE 'Participant ID: %', participant.p_id;
    RAISE NOTICE 'Name: % %', participant.f_name, participant.l_name;
    RAISE NOTICE 'Phone Number: %', participant.phoneno;
    RAISE NOTICE 'Skills: %', participant.skills;
    RAISE NOTICE 'Team ID: %', participant.t_id;
    RAISE NOTICE '--------------------------';
  END LOOP;
  CLOSE participant_cursor;
END;
$$ LANGUAGE plpgsql;
```

```
postgres=# CREATE OR REPLACE FUNCTION process_participants()
postgres-# RETURNS void AS $$
postgres$# DECLARE
postgres$#   participant_cursor CURSOR FOR
postgres$#     SELECT *
postgres$#     FROM participants;
postgres$#   participant participants%ROWTYPE;
postgres$#
postgres$# BEGIN
postgres$#   OPEN participant_cursor;
postgres$#
postgres$#   LOOP
postgres$#     FETCH participant_cursor INTO participant;
postgres$#
postgres$#     EXIT WHEN NOT FOUND;
postgres$#
postgres$#     RAISE NOTICE 'Participant ID: %', participant.p_id;
postgres$#     RAISE NOTICE 'Name: % %', participant.f_name, participant.l_name;
postgres$#     RAISE NOTICE 'Phone Number: %', participant.phoneno;
postgres$#     RAISE NOTICE 'Skills: %', participant.skills;
postgres$#     RAISE NOTICE 'Team ID: %', participant.t_id;
postgres$#     RAISE NOTICE '--------------------------';
postgres$#   END LOOP;
postgres$#   CLOSE participant_cursor;
postgres$#
postgres$# END;
postgres$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
postgres=#
postgres=# select process_participants();
NOTICE:  Participant ID: P01
NOTICE:  Name: Himanshi Thakkar
NOTICE:  Phone Number: 8780230308
NOTICE:  Skills: c++,java
NOTICE:  Team ID: T01
```

```
NOTICE:  Participant ID: P16
NOTICE:  Name: yash soni
NOTICE:  Phone Number: 9730123289
NOTICE:  Skills: React, next  JS
NOTICE:  Team ID: T03
NOTICE:  --------------------------
NOTICE:  Participant ID: P18
NOTICE:  Name: Meet soni
NOTICE:  Phone Number: 9730123289
NOTICE:  Skills: React, next  JS
NOTICE:  Team ID: T04
NOTICE:  --------------------------
NOTICE:  Participant ID: P19
NOTICE:  Name: Meet soni
NOTICE:  Phone Number: 9730123289
NOTICE:  Skills: React, next  JS
NOTICE:  Team ID: T04
NOTICE:  --------------------------
 process_participants
----------------------

(1 row)
```