EVENTEASE - EVENT MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

SIDDHARTH SINGH [Reg No: RA2211028010074]

LAVANYA PAKHALE [Reg No: RA2211028010132]

DEEPSHIKHA KUMARI [Reg No: RA2211028010134]

Under the Guidance of

Dr. V.Nallarasan

Assistant Professor, Department of Networking and Communications

in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

With specialization in CLOUD COMPUTING

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR– 603 203

MAY 2024

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Register no. RA2211028010074, RA2211028010132 and RA2211028010134, Certified to be the bonafide work done by Siddharth Singh , Lavanya Pakhale, Deepshikha Kumari of II year/IV semester B. Tech Degree Course in the Project Course – **21CSC205P Database Management Systems** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY,** Kattankulathur for the academic year 2023-2024.

[illegible] 06/05/24

Faculty in Charge
Dr. V. Nallarasan
Assistant Professor
Department of Networking
and Communications
SRMIST - KTR

Dept. of Networking and Communications
Kattankulathur
603 203
• SRM IST •

[illegible]

HEAD OF THE DEPARTMENT
Dr. Annapurani Panaiyappan K
Professor and H.O.D
Department of Networking
and Communications
SRMIST - KTR

ABSTRACT

In today's dynamic event management landscape, efficient organization is paramount for businesses, organizations, and individuals alike. The EventEase system offers a comprehensive solution to streamline event planning, participant engagement, and resource allocation. With features like venue selection, budget management, guest list coordination, and marketing strategies, EventEase simplifies the event planning process while providing insights into event performance. Additionally, seamless registration and ticketing processes enhance participant engagement, while feedback collection tools enable continuous improvement. It also excels in budget management EventEase ensures scalability and flexibility to handle diverse event management needs. Targeting event planners, businesses, educational institutions, non-profit organizations, and individuals hosting various events, EventEase empowers users to execute flawless events effortlessly. Through its user-friendly interface and robust functionality, EventEase aims to revolutionize the event planning industry and deliver exceptional experiences for organizers and participants alike.

TABLE OF CONTENTS

CHAPTER-1

☐ Problem Understanding:

In today's dynamic world, organizing events efficiently is crucial, yet many organizations face challenges with manual processes, communication gaps, and resource mismanagement. To address these issues, a comprehensive Event Management System (EMS) is needed to streamline event planning, participant engagement, and resource allocation while providing insights into event performance. This system should consolidate various processes into a centralized platform, enabling event organizers to efficiently manage tasks, collaborate with stakeholders, facilitate seamless participant interactions, optimize resource utilization, and leverage data-driven analytics for continuous improvement. By automating and integrating key aspects of event management, the EMS can enhance the overall event experience, foster better collaboration, and drive operational efficiencies for businesses, organizations, and individuals alike.

☐ Identification of Entity and Relationships:

1. Events:

- Event ID (Primary Key)
- Event Name
- Date
- Time
- Venue ID (Foreign Key referencing Venues table)

- Description

2. Venues:
   - Venue ID (Primary Key)
   - Venue Name
   - Address
   - Capacity
   - Contact Person
   - Contact Email
   - Contact Phone
3. Participants:
   - Participant ID (Primary Key)
   - Name
   - Email
   - Phone
   - Organization (if applicable)
4. Tickets:
   - Ticket ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Price
   - Type (e.g., General Admission, VIP)
   - Availability
   - Sale Date
5. Staff:
   - Staff ID (Primary Key)

   - Name
   - Email
   - Phone
   - Role (e.g., Coordinator, Volunteer)

6. Sponsors:
   - Sponsor ID (Primary Key)
   - Sponsor Name
   - Contact Person
   - Contact Email
   - Contact Phone
   - Sponsorship Level
   - Amount Sponsored

7. Budget:
   - Budget ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Total Budget
   - Allocation Details
   - Expenditure

8. Sessions or Agenda:
   - Session ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Title
   - Description
   - Date

- Time

9. Speakers:
   - Speaker ID (Primary Key)
   - Name
   - Bio
   - Contact Information

10. Registrations:
    - Registration ID (Primary Key)
    - Event ID (Foreign Key referencing Events table)
    - Participant ID (Foreign Key referencing Participants table)
    - Registration Date
    - Status (e.g., Confirmed, Pending)

11. Feedback:
    - Feedback ID (Primary Key)
    - Event ID (Foreign Key referencing Events table)
    - Participant ID (Foreign Key referencing Participants table)
    - Rating
    - Comments

12. Vendors:
    - Vendor ID (Primary Key)
    - Vendor Name
    - Contact Person
    - Contact Email

- Contact Phone
- Services Provided

13. Transportation:
   - Transportation ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Mode of Transportation
   - Departure Date and Time
   - Arrival Date and Time
   - Pickup/Drop-off Locations

14. Tasks:
   - Task ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Description
   - Assigned To
   - Deadline
   - Status

15. Contracts:
   - Contract ID (Primary Key)
   - Event ID (Foreign Key referencing Events table)
   - Counterparty
   - Contract Type
   - Terms and Conditions
   - Start Date
   - End Date

☐ MIND MAP:

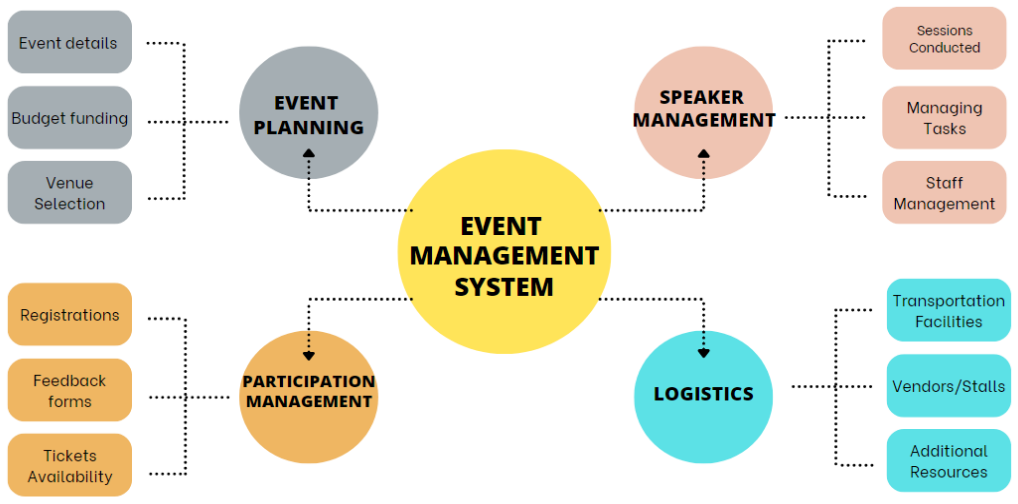


Fig:1.1

☐ ER Diagram:

The ER diagram illustrates how users, such as event organizers, participants, and vendors, interact with events through various relationships, enabling the Event Management System to facilitate efficient event planning

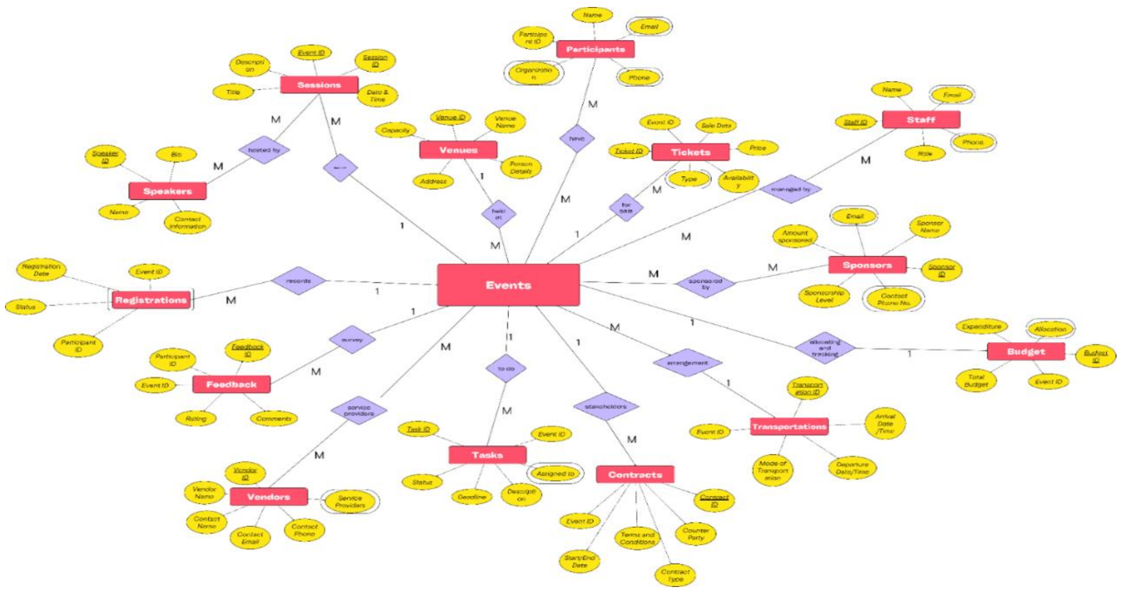


Fig 1.2

CHAPTER-2

☐ Creation of Database Tables for the project:

Events ID (primary key)	Event Name	Venue ID (foreign key)	Date	Time	Description

Schema: Event (Event ID, Event Name, Date, Time, Venue ID, Description)

Venue ID (primary key)	Venue Name	City	Capacity	Contact Name

Schema: Venue (Venue ID, Venue Name, Contact Name)

Venue ID (primary key)	Address	Email	Contact No.

Schema: Venue(Venue ID, Address, Email, Contact no.)

Participant ID (primary key)	Name

Schema: Participants (Participant ID, Name, Email, Phone No.)

Participant ID (primary key)	Email	Phone no.

Schema: Participants(Participant ID, Email, Phone no.)

Staff ID (primary key)	Name	Role

Schema: Staff (Staff ID, Name, Role)

Staff ID (primary key)	Email	Phone no.

Schema: Staff(Staff ID, Email, Phone no.)

Sponsor ID (primary key)	Sponsor Name	Contact Person	Sponsorship Level	Amount Sponsored

Schema: Sponsors (Sponsor ID, Sponsor Name, Sponsorship Level, Amount Sponsored)

Sponsor ID (primary key)	Email	Phone no.

Schema: Sponsor(Sponsor ID, Email, Phone no.)

Budget ID (primary key)	Event ID (foreign key)	Total Budget	Allocation Details	Expenditure

Schema: Budget (Budget ID, Event ID, Total Budget, Allocation Details, Expenditure)

Session ID (primary key)	Event ID (foreign key)	Title	Description	Date	Time

Schema: Session (Session ID, Event ID, Title, Description, Date, Time)

Speaker ID (primary key)	First_Name	Last_Name	Bio	Contact

Schema: Speaker (Speaker ID, First_Name, Last_Name, Bio, Contact)

Registration ID (primary key)	Event ID (foreign key)	Participant ID (foreign key)	Registration Date	Status

Schema: Registrations (Registration ID, Event ID, Participant ID, Registration Date, Status)

Feedback ID (primary key)	Event ID (foreign key)	Participant ID	Rating	Comment

Schema: Feedback (Feedback ID, Event ID, Participant ID, Rating, Comments)

Vendor ID (primary key)	Vendor First_Name	Vendor Last_Name	Contact First_Name	Contact Last_Name	Service Providers

Schema: Vendor (Vendor ID, Vendor First_Name, Vendor Last_Name, Contact First_name, contact last_name, service provider)

Vendor ID (primary key)	Email	Phone no.

Schema: Vendor (Vendor ID, Email, Phone no.)

Transportation id (primary key)	Event id (foreign key)	Mode of transportation	Departure date	Departure time	Arrival date	Arrival time	Pick up location	Drop off location

Schema: Transportation (transportation id, event id, mode of transportation, departure date, departure time, arrival date, arrival time, pick up location, drop off location)

Task id (primary key)	Event id (foreign key)	description	Assigned to	status	deadline

Schema: tasks(task id, event id, description, assigned to, status, deadline)

Contract id (primary key)	Event id (foreign key)	Counter party	Contract type	Terms and conditions	Start date	End date

Schema: contract(contract id, event id, counter party, contract type, terms and conditions start date, end date)

Participant ID	Event ID	have

Schema: have(Participant ID, Event ID)

Sponsor ID	Event ID	Sponsored by

Schema: Sponsored by(Sponsor ID, Event ID)

Staff ID	Event ID	Managed by

Schema: Managed by(Staff ID, Event ID)

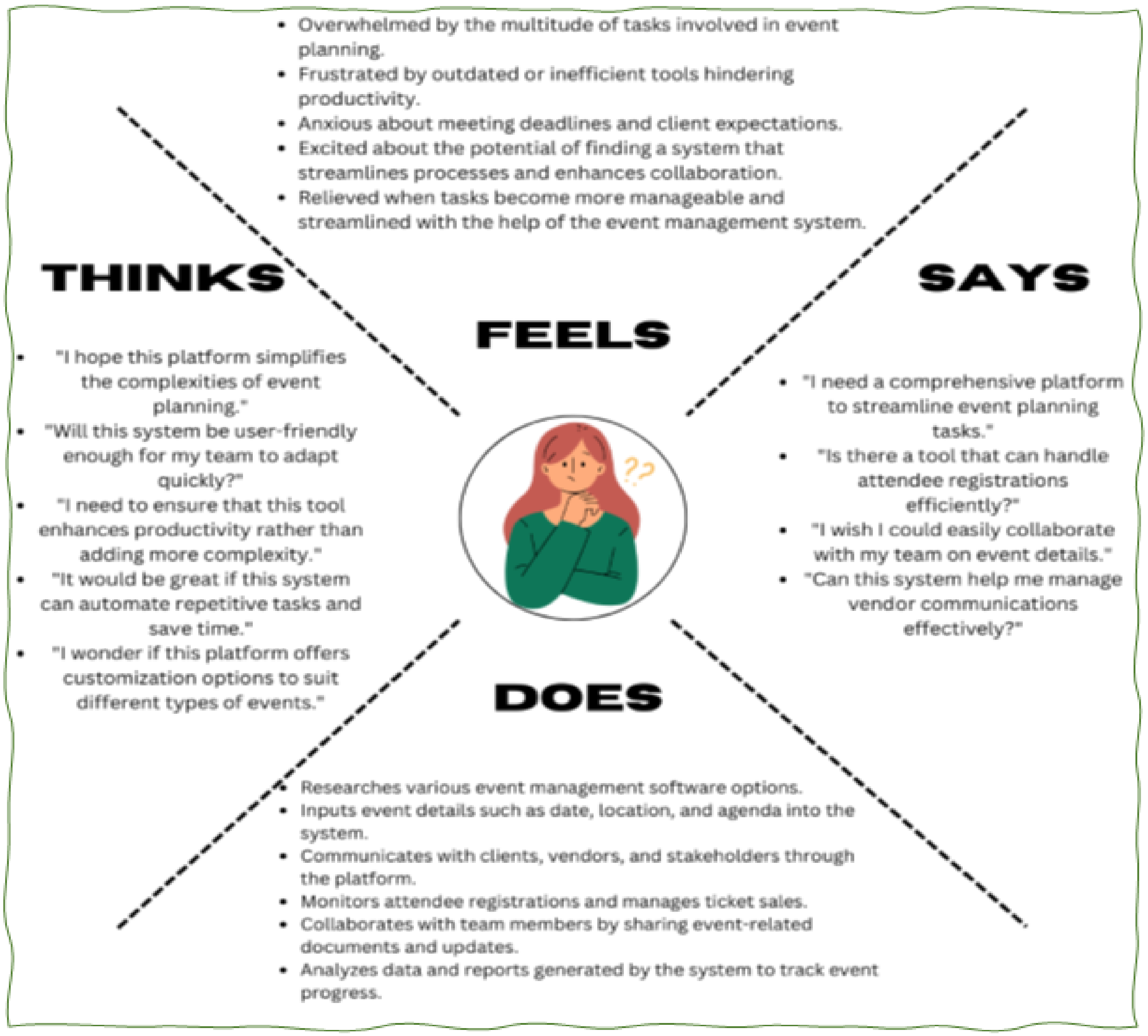
Vendor ID	Event ID	Service provided

Schema: Service Provided(Vendor ID, Event ID)

Session ID	Speaker ID	Hosted by

Schema: Hosted by(Session ID, Speaker ID)

☐ Empathy Map:



- Overwhelmed by the multitude of tasks involved in event planning.
- Frustrated by outdated or inefficient tools hindering productivity.
- Anxious about meeting deadlines and client expectations.
- Excited about the potential of finding a system that streamlines processes and enhances collaboration.
- Relieved when tasks become more manageable and streamlined with the help of the event management system.
THINKS
SAYS
FEELS
- "I hope this platform simplifies the complexities of event planning."
- "Will this system be user-friendly enough for my team to adapt quickly?"
- "I need to ensure that this tool enhances productivity rather than adding more complexity."
- "It would be great if this system can automate repetitive tasks and save time."
- "I wonder if this platform offers customization options to suit different types of events."
- "I need a comprehensive platform to streamline event planning tasks."
- "Is there a tool that can handle attendee registrations efficiently?"
- "I wish I could easily collaborate with my team on event details."
- "Can this system help me manage vendor communications effectively?"
??
DOES
- Researches various event management software options.
- Inputs event details such as date, location, and agenda into the system.
- Communicates with clients, vendors, and stakeholders through the platform.
- Monitors attendee registrations and manages ticket sales.
- Collaborates with team members by sharing event-related documents and updates.
- Analyzes data and reports generated by the system to track event progress.

CHAPTER-3

☐ Table Creation with Constraints:

```
CREATE TABLE Venues (
  Venue_ID INT PRIMARY KEY,
  Venue_Name VARCHAR2(100) NOT NULL,
  Address VARCHAR2(200) NOT NULL,
  Capacity INT NOT NULL,
  Contact_Person VARCHAR2(100),
  Contact_Email VARCHAR2(100),
  Contact_Phone VARCHAR2(20)
);

CREATE TABLE Participants (
  Participant_ID INT PRIMARY KEY,
  Name VARCHAR2(100) NOT NULL,
  Email VARCHAR2(100) NOT NULL UNIQUE,
  Phone VARCHAR2(20),
  Organization VARCHAR2(100)
);
 CREATE TABLE Speakers (
  Speaker_ID INT PRIMARY KEY,
  Name VARCHAR2(100) NOT NULL,
  Bio VARCHAR2(500),
  Contact_Information VARCHAR2(200)
);
 CREATE TABLE Staff (
  Staff_ID INT PRIMARY KEY,
  Name VARCHAR2(100) NOT NULL,
  Email VARCHAR2(100) NOT NULL UNIQUE,
  Phone VARCHAR2(20),
  Role VARCHAR2(50)
);
 CREATE TABLE Sponsors (
  Sponsor_ID INT PRIMARY KEY,
  Sponsor_Name VARCHAR2(100) NOT NULL,
  Contact_Person VARCHAR2(100),
  Contact_Email VARCHAR2(100),
  Contact_Phone VARCHAR2(20),
  Sponsorship_Level VARCHAR2(50),
  Amount_Sponsored NUMBER(10, 2) DEFAULT 0,
  CONSTRAINT chk_Amount_Sponsored CHECK (Amount_Sponsored >= 0)
);
 CREATE TABLE Events (
  Event_ID INT PRIMARY KEY,
  Event_Name VARCHAR2(100) NOT NULL,
  Event_Date DATE NOT NULL,
  Event_Time TIMESTAMP NOT NULL,
  Venue_ID INT NOT NULL,
  Description VARCHAR2(500),
  CONSTRAINT fk_Events_Venues FOREIGN KEY (Venue_ID) REFERENCES
Venues(Venue_ID)
);
 CREATE TABLE Tickets (
  Ticket_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Price NUMBER(10, 2) NOT NULL,
  Type VARCHAR2(50) NOT NULL,
  Availability INT NOT NULL DEFAULT 0,
  Sale_Date DATE,
  CONSTRAINT fk_Tickets_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID),
  CONSTRAINT chk_Ticket_Price CHECK (Price > 0),
  CONSTRAINT chk_Ticket_Availability CHECK (Availability >= 0)
```

```
);
 CREATE TABLE Budget (
  Budget_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Total_Budget NUMBER(10, 2) NOT NULL,
  Allocation_Details VARCHAR2(500),
  Expenditure NUMBER(10, 2),
  CONSTRAINT fk_Budget_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID),
  CONSTRAINT chk_Total_Budget CHECK (Total_Budget >= 0),
  CONSTRAINT chk_Expenditure CHECK (Expenditure >= 0)
);
CREATE TABLE Registrations (
  Registration_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Participant_ID INT NOT NULL,
  Registration_Date DATE NOT NULL,
  Status VARCHAR2(50) NOT NULL,
  CONSTRAINT fk_Registrations_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID),
  CONSTRAINT fk_Registrations_Participants FOREIGN KEY (Participant_ID)
REFERENCES Participants(Participant_ID)
);
 CREATE TABLE Feedback (
  Feedback_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Participant_ID INT NOT NULL,
  Rating INT NOT NULL,
  Comments VARCHAR2(500),
  CONSTRAINT fk_Feedback_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID),
  CONSTRAINT fk_Feedback_Participants FOREIGN KEY (Participant_ID)
REFERENCES Participants(Participant_ID),
  CONSTRAINT chk_Rating CHECK (Rating >= 1 AND Rating <= 5)
```

```
);
 CREATE TABLE Vendors (
  Vendor_ID INT PRIMARY KEY,
  Vendor_Name VARCHAR2(100) NOT NULL,
  Contact_Person VARCHAR2(100),
  Contact_Email VARCHAR2(100),
  Contact_Phone VARCHAR2(20),
  Services_Provided VARCHAR2(200)
);
 CREATE TABLE Transportation (
  Transportation_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Mode_of_Transportation VARCHAR2(100) NOT NULL,
  Departure_Date TIMESTAMP NOT NULL,
  Arrival_Date TIMESTAMP NOT NULL,
  Pickup_Dropoff_Locations VARCHAR2(200),
  CONSTRAINT fk_Transportation_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID),
  CONSTRAINT chk_Transportation_Dates CHECK (Departure_Date < Arrival_Date)
);
 CREATE TABLE Tasks (
  Task_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Description VARCHAR2(500) NOT NULL,
  Assigned_To VARCHAR2(100) NOT NULL,
  Deadline DATE NOT NULL,
  Status VARCHAR2(50) NOT NULL,
  CONSTRAINT fk_Tasks_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID)
);
 CREATE TABLE Sessions (
  Session_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Title VARCHAR2(100) NOT NULL,
  Description VARCHAR2(500),
  Session_Date DATE NOT NULL,
  Session_Time TIMESTAMP NOT NULL,
  CONSTRAINT fk_Sessions_Events FOREIGN KEY (Event_ID) REFERENCES
Events(Event_ID)
);
 CREATE TABLE Contracts (
  Contract_ID INT PRIMARY KEY,
  Event_ID INT NOT NULL,
  Counterparty VARCHAR2(100),
  Contract_Type VARCHAR2(50),
  Terms_and_Conditions VARCHAR2(500),
  Start_Date DATE,
  End_Date DATE,
  CONSTRAINT fk_Contracts_Events FOREIGN KEY
```

☐ DML Commands to insert entries:

-- Inserting data into the venue table:

```
INSERT INTO Venues (Venue_Name, Address, Capacity, Contact_Person, Contact_Email, Contact_Phone)
VALUES
  ('Convention Center', '1 Main St, Cityville', 2000, 'John Smith', 'john@conventioncenter.com', '123-
456-7890'),
  ('Auditorium', '5 Park Ave, Townsville', 500, 'Jane Doe', 'jane@auditorium.com', '987-654-3210'),
  ('Art Gallery', '10 Museum Rd, Villagetown', 150, 'Alice Johnson', 'alice@artgallery.com', '111-222-
3333'),
  ('Coworking Space', '15 Tech Lane, Hamletville', 100, 'Bob Brown', 'bob@coworking.com', '444-555-
6666'),
  ('Hotel Grand', '20 Central Ave, Lodgetown', 800, 'Carol White', 'carol@hotelgrand.com', '777-888-
9999');
```

-- Inserting data into the participant table:

```
INSERT INTO Participants (Name, Email, Phone, Organization)
```

VALUES

('Michael Lee', 'michael.lee@gmail.com', '555-123-4567', 'Tech Company Inc.'),

('Sarah Jones', 'sarah.jones@yahoo.com', '222-333-4444', 'Marketing Agency'),

('David Williams', 'david.williams@outlook.com', '111-555-7777', 'University'),

('Emily Brown', 'emily.brown@hotmail.com', '999-888-6666', 'Freelancer'),

('Daniel Garcia', 'daniel.garcia@aol.com', '333-222-1111', 'Non-Profit Organization');

--- Inserting data into the events table:

INSERT INTO Events (Event_Name, Event_Date, Event_Time, Venue_ID, Description)

VALUES

('Tech Conference 2024', '2024-06-15', '09-00-00', 1, 'A gathering of tech enthusiasts and industry leaders'),

('Art Exhibition Opening', '2024-05-20', '18-00-00', 3, 'Showcase of local and international contemporary art'),

('Community Workshop', '2024-07-10', '10-00-00', 4, 'Interactive session on sustainable living practices'),

('Networking Mixer', '2024-04-25', '17-00-00', 2, 'Opportunity to connect with professionals across various fields'),

('Summer Music Festival', '2024-08-12', '14-00-00', 5, 'Live music performances by popular artists');

-- Inserting data into the Registrations table:

INSERT INTO Registrations (Registration_ID, Event_ID, Participant_ID, Registration_Date, Status) VALUES

(1, 1, 1, '2024-03-01', 'Confirmed'),

(2, 1, 2, '2024-03-02', 'Confirmed'),

(3, 2, 3, '2024-03-03', 'Pending'),

(4, 3, 4, '2024-03-04', 'Confirmed'),

(5, 3, 5, '2024-03-05', 'Cancelled');

-- Inserting data into the Feedback table:

INSERT INTO Feedback (Event_ID, Participant_ID, Rating, Comments) VALUES

(1, 1, 4, 'Great event overall. Enjoyed the sessions.'),

(1, 2, 5, 'Excellent organization and content. Would attend again.'),

(2, 3, 3, 'Good event, but could have been more interactive.'),

(3, 4, 5, 'Fantastic experience. Highly recommended.'),

(3, 5, 2, 'Disappointed with the lack of variety in sessions.');

-- Inserting data into the Sessions table:

INSERT INTO Sessions (Event_ID, Title, Description, Session_Date, Session_Time) VALUES

(1, 'Keynote Address', 'Opening keynote by renowned speaker', '2024-04-01', '2024-04-01 09:00:00'),

(1, 'Panel Discussion', 'Panel discussion on industry trends', '2024-04-01', '2024-04-01 11:00:00'), (2, 'Workshop: Data Analytics', 'Hands-on workshop on data analysis techniques', '2024-03-15', '2024-03-15 10:00:00'),

(2, 'Networking Lunch', 'Networking session over lunch', '2024-03-15', '2024-03-15 12:30:00'),

(3, 'Product Showcase', 'Showcasing latest products and innovations', '2024-04-10', '2024-04-10 11:00:00');

◻ TRIGGERS:

1. To create a trigger that gets invoked if the numbers entered in the **Phone** column are less than 10 characters long

```
CREATE OR REPLACE TRIGGER check_phone_length
BEFORE INSERT ON Participants
FOR EACH ROW
BEGIN
  IF LENGTH(:NEW.Phone) < 10 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Phone number must be at least 10 characters long');
  END IF;
END;
/
```

2. To create a trigger that gets invoked when the format of entering an email is incorrect, you can use a regular expression to validate the email format.

```
CREATE OR REPLACE TRIGGER check_email_format
BEFORE INSERT ON Staff
FOR EACH ROW
DECLARE
  email_pattern VARCHAR(100) := '[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}';
BEGIN
  IF NOT REGEXP_LIKE(:NEW.Email, email_pattern) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid email format');
  END IF;
END;
```

◻ VIEWS:

1. The "EventRegistrations" view combines data from the Events, Registrations, and Participants tables, providing details such as event name, participant name, registration date, and status for each registration.

```
CREATE VIEW EventRegistrations AS

SELECT e.Event_Name, p.Name, r.Registration_Date, r.Status

FROM Events e

INNER JOIN Registrations r ON e.Event_ID = r.Event_ID

INNER JOIN Participants p ON r.Participant_ID = p.Participant_ID;

describe EventRegistrations;
```

2. The "AvailableTickets" view displays event names, ticket types, prices, and availability for tickets with availability greater than zero, by joining the Events and Tickets tables.

```
CREATE VIEW AvailableTickets AS

SELECT e.Event_Name, t.Type, t.Price, t.Availability

FROM Events e

INNER JOIN Tickets t ON e.Event_ID = t.Event_ID

WHERE t.Availability > 0;

describe AvailableTickets;
```

☐ CURSORS:

1. This declares a cursor named event_cursor for the Events table. It then opens the cursor, fetches each row of data from the Events table one by one, processes it and continues until there are no more rows to fetch. Finally, it closes the cursor.

```
DECLARE

  -- Declare the cursor

  CURSOR event_cursor IS

    SELECT * FROM Events;

  -- Declare variables to store column values

  event_id Events.Event_ID%TYPE;

  event_name Events.Event_Name%TYPE;

  event_date Events.Event_Date%TYPE;
  event_time Events.Event_Time%TYPE;
  venue_id Events.Venue_ID%TYPE;
  event_description Events.Description%TYPE;
BEGIN
  -- Open the cursor
  OPEN event_cursor;
  -- Fetch rows from the cursor one by one
  FETCH event_cursor INTO event_id, event_name, event_date, event_time, venue_id, 
event_description;
  -- Loop through the cursor to fetch all rows
  WHILE event_cursor%FOUND LOOP
    -- Print or use the fetched values as needed
    DBMS_OUTPUT.PUT_LINE(event_id || ', ' || event_name || ', ' || TO_CHAR(event_date, 
'DD-MON-YY') || ', ' || TO_CHAR(event_time, 'DD-MON-YY HH:MI:SS AM') || ', ' || 
venue_id || ', ' || event_description);
    -- Fetch the next row
    FETCH event_cursor INTO event_id, event_name, event_date, event_time, venue_id, 
event_description;
  END LOOP;
  -- Close the cursor
  CLOSE event_cursor;
END;
/
```

2. This cursor fetches data from the `Participants` table row by row and prints each row's details, including participant ID, name, email, phone, and organization, using `DBMS_OUTPUT.PUT_LINE` in a PL/SQL environment.

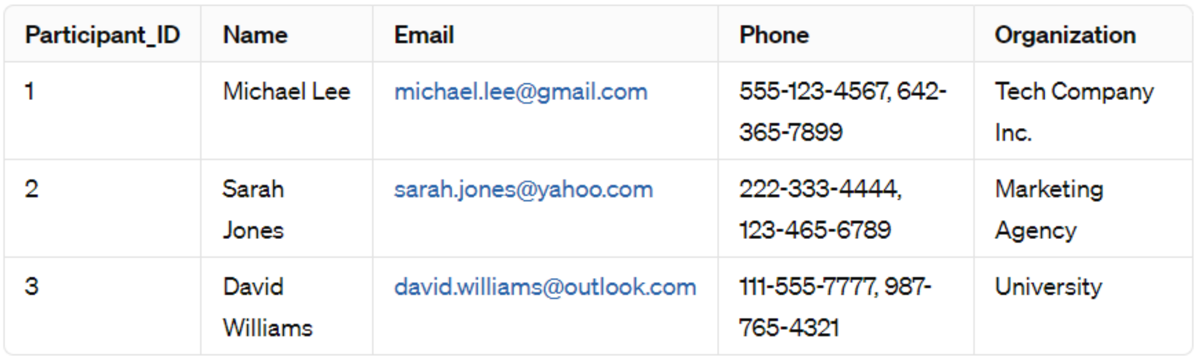
```
DECLARE
  -- Declare the cursor
  CURSOR participant_cursor IS
    SELECT * FROM Participants;
  -- Declare variables to store column values
  participant_id Participants.Participant_ID%TYPE;
  participant_name Participants.Name%TYPE;
  participant_email Participants.Email%TYPE;
  participant_phone Participants.Phone%TYPE;
  participant_organization Participants.Organization%TYPE;
BEGIN
  -- Open the cursor
  OPEN participant_cursor;
  -- Fetch rows from the cursor one by one
  FETCH participant_cursor INTO participant_id, participant_name, participant_email,
participant_phone, participant_organization;
  -- Loop through the cursor to fetch all rows
  WHILE participant_cursor%FOUND LOOP
      -- Print or use the fetched values as needed
      DBMS_OUTPUT.PUT_LINE(participant_id || ', ' || participant_name || ', ' ||
participant_email || ', ' || participant_phone || ', ' || participant_organization);
      -- Fetch the next row
      FETCH participant_cursor INTO participant_id, participant_name, participant_email,
participant_phone, participant_organization;
  END LOOP;
  -- Close the cursor
  CLOSE participant_cursor;
END;
/
```

CHAPTER-4

1 NF:

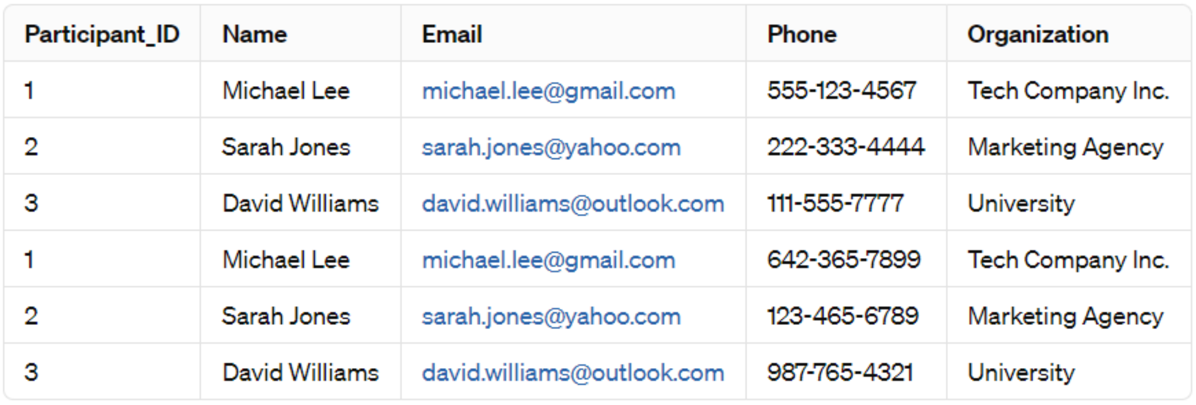
- The table must have a primary key.
- All columns must contain atomic (single) values, with no repeating groups.
- The table must not have any multi-valued attributes.

This table is in unnormalize form as it contains multi-valued attributes.



Participant_ID	Name	Email	Phone	Organization
1	Michael Lee	michael.lee@gmail.com	555-123-4567, 642-365-7899	Tech Company Inc.
2	Sarah Jones	sarah.jones@yahoo.com	222-333-4444, 123-465-6789	Marketing Agency
3	David Williams	david.williams@outlook.com	111-555-7777, 987-765-4321	University

After Normalization:



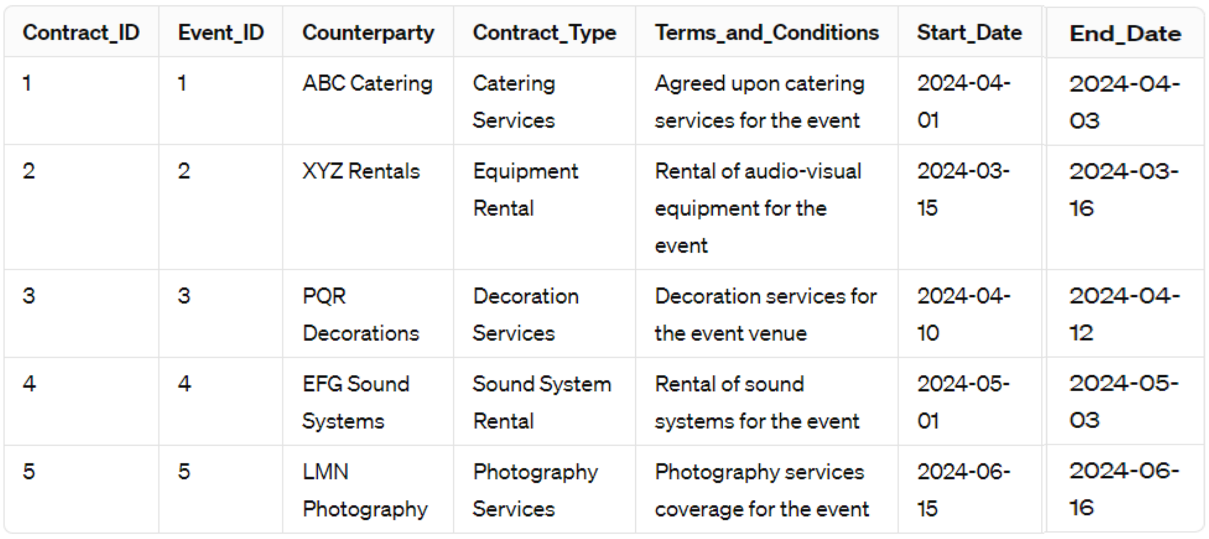
Participant_ID	Name	Email	Phone	Organization
1	Michael Lee	michael.lee@gmail.com	555-123-4567	Tech Company Inc.
2	Sarah Jones	sarah.jones@yahoo.com	222-333-4444	Marketing Agency
3	David Williams	david.williams@outlook.com	111-555-7777	University
1	Michael Lee	michael.lee@gmail.com	642-365-7899	Tech Company Inc.
2	Sarah Jones	sarah.jones@yahoo.com	123-465-6789	Marketing Agency
3	David Williams	david.williams@outlook.com	987-765-4321	University

The “Participants” table is in the First Normal Form (1NF) because:

- All values in each column are indivisible.
- There are no repeating groups of columns.
- Each column has a unique name.

2 NF:

- The table must be in 1NF.
- All non-key attributes must be fully dependent on the primary key, with no partial dependencies.

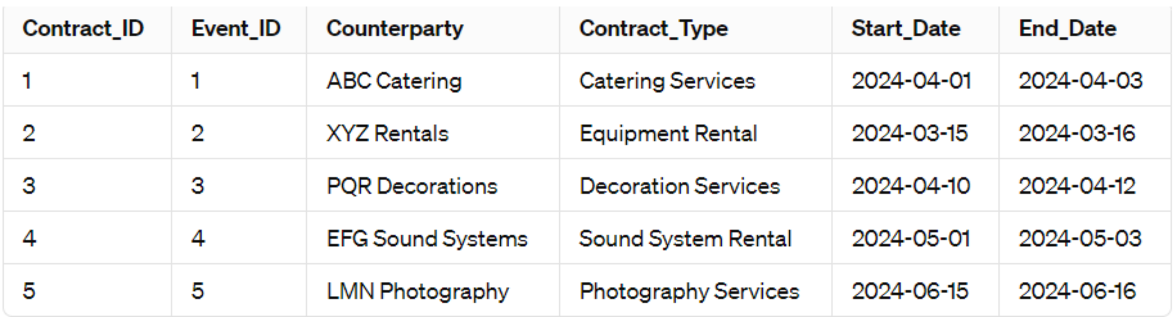


Contract_ID	Event_ID	Counterparty	Contract_Type	Terms_and_Conditions	Start_Date	End_Date
1	1	ABC Catering	Catering Services	Agreed upon catering services for the event	2024-04-01	2024-04-03
2	2	XYZ Rentals	Equipment Rental	Rental of audio-visual equipment for the event	2024-03-15	2024-03-16
3	3	PQR Decorations	Decoration Services	Decoration services for the event venue	2024-04-10	2024-04-12
4	4	EFG Sound Systems	Sound System Rental	Rental of sound systems for the event	2024-05-01	2024-05-03
5	5	LMN Photography	Photography Services	Photography services coverage for the event	2024-06-15	2024-06-16

In this table, the Terms_and_Conditions attribute appears to be partially dependent on the Contract_ID and Event_ID, as it describes aspects of the contract specific to each contract and event combination.

Let's split the Contracts table into two tables:

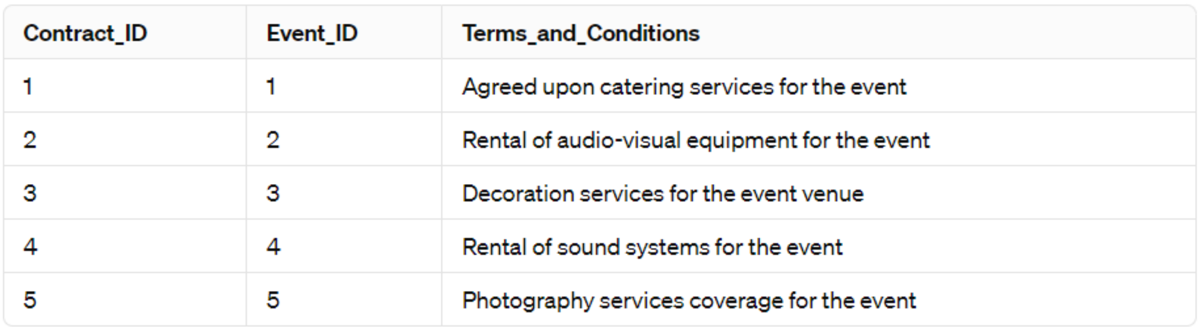
Contracts_Info table:



Contract_ID	Event_ID	Counterparty	Contract_Type	Start_Date	End_Date
1	1	ABC Catering	Catering Services	2024-04-01	2024-04-03
2	2	XYZ Rentals	Equipment Rental	2024-03-15	2024-03-16
3	3	PQR Decorations	Decoration Services	2024-04-10	2024-04-12
4	4	EFG Sound Systems	Sound System Rental	2024-05-01	2024-05-03
5	5	LMN Photography	Photography Services	2024-06-15	2024-06-16

Contracts_Info contains attributes directly related to the Contract_ID and Event_ID, such as Counterparty, Contract_Type, Start_Date, and End_Date. There are no partial dependencies since all attributes are fully dependent on the entire primary key.

Contracts Details:

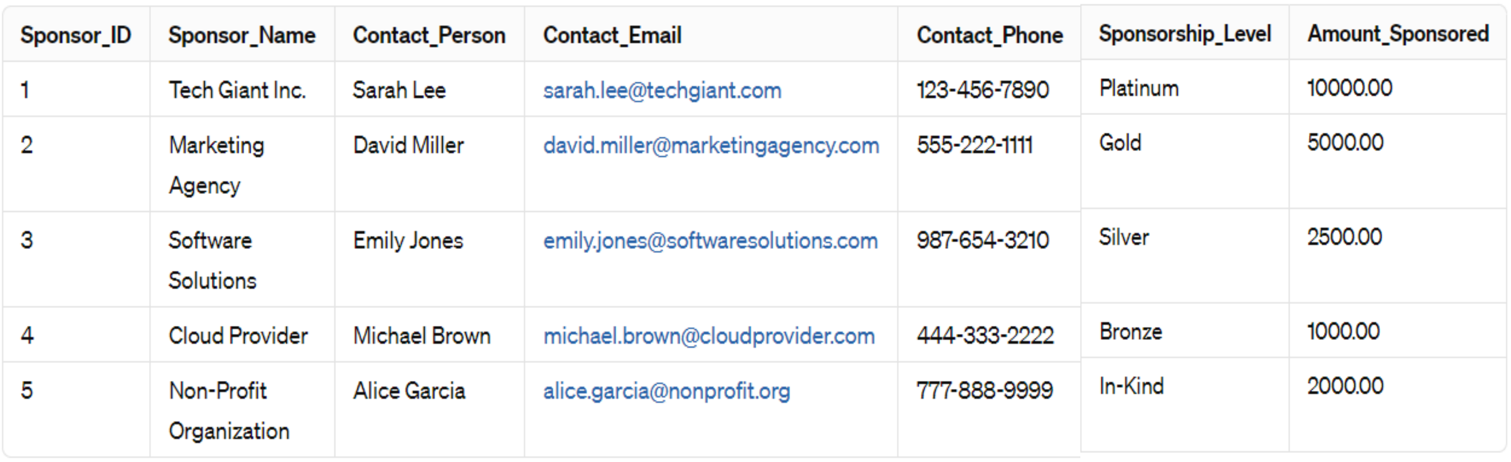


Contract_ID	Event_ID	Terms_and_Conditions
1	1	Agreed upon catering services for the event
2	2	Rental of audio-visual equipment for the event
3	3	Decoration services for the event venue
4	4	Rental of sound systems for the event
5	5	Photography services coverage for the event

Contracts_Details contains the Terms_and_Conditions attribute, which is related only to the Contract_ID and Event_ID primary key combination, removing any partial dependencies from the original Contracts table.

<u>3 NF:</u>

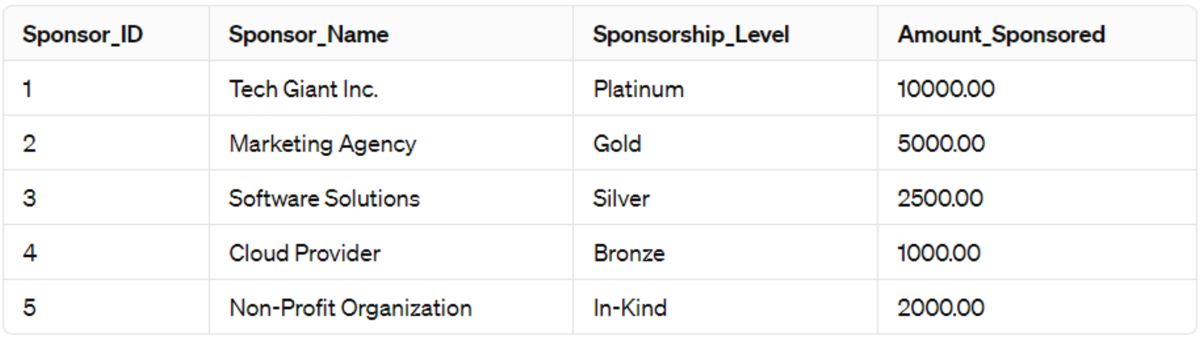
- The table must be in 2NF.
- All non-key attributes must be independent of each other and must depend only on the primary key, with no transitive dependencies.



Sponsor_ID	Sponsor_Name	Contact_Person	Contact_Email	Contact_Phone	Sponsorship_Level	Amount_Sponsored
1	Tech Giant Inc.	Sarah Lee	sarah.lee@techgiant.com	123-456-7890	Platinum	10000.00
2	Marketing Agency	David Miller	david.miller@marketingagency.com	555-222-1111	Gold	5000.00
3	Software Solutions	Emily Jones	emily.jones@softwaresolutions.com	987-654-3210	Silver	2500.00
4	Cloud Provider	Michael Brown	michael.brown@cloudprovider.com	444-333-2222	Bronze	1000.00
5	Non-Profit Organization	Alice Garcia	alice.garcia@nonprofit.org	777-888-9999	In-Kind	2000.00

The original table had a redundancy issue because contact information was repeated for each sponsor. For example, if a sponsor changed their contact email or phone number, it would need to be updated in multiple rows, leading to potential inconsistencies.

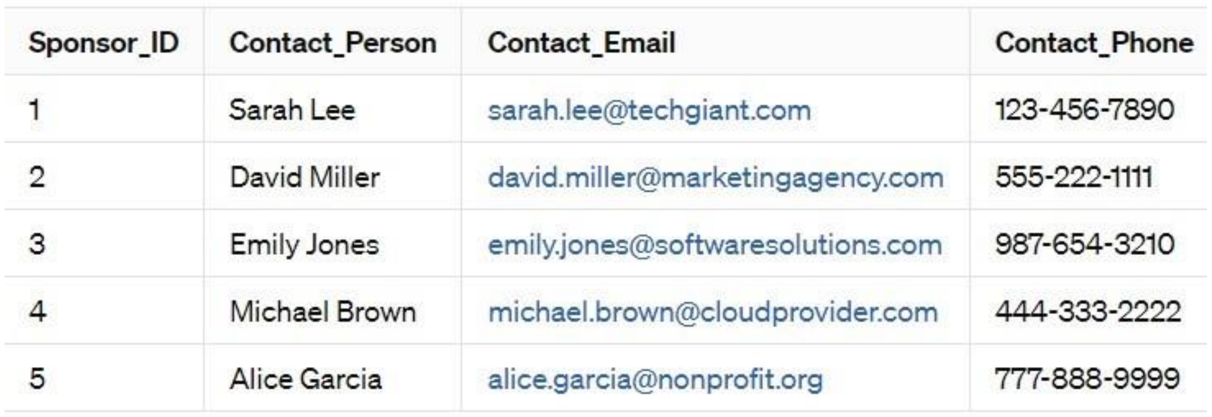
Sponsors Table:



Sponsor_ID	Sponsor_Name	Sponsorship_Level	Amount_Sponsored
1	Tech Giant Inc.	Platinum	10000.00
2	Marketing Agency	Gold	5000.00
3	Software Solutions	Silver	2500.00
4	Cloud Provider	Bronze	1000.00
5	Non-Profit Organization	In-Kind	2000.00

In the Sponsors table, we have the Sponsor_ID as the primary key, and the Sponsor_Name, Sponsorship_Level, and Amount_Sponsored as non-key attributes. These attributes are independent of each other.

SponsorContacts Table:

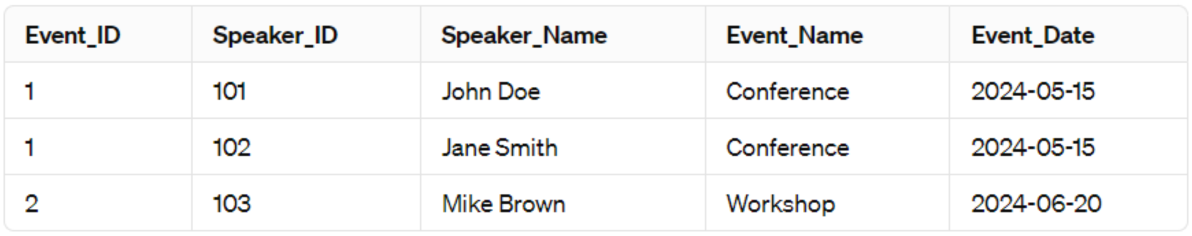


Sponsor_ID	Contact_Person	Contact_Email	Contact_Phone
1	Sarah Lee	sarah.lee@techgiant.com	123-456-7890
2	David Miller	david.miller@marketingagency.com	555-222-1111
3	Emily Jones	emily.jones@softwaresolutions.com	987-654-3210
4	Michael Brown	michael.brown@cloudprovider.com	444-333-2222
5	Alice Garcia	alice.garcia@nonprofit.org	777-888-9999

In the SponsorContacts table, we have the SponsorContact_ID as the primary key, Sponsor_ID as a foreign key referencing the Sponsors table, and the Contact_Person, Contact_Email, and Contact_Phone as non-key attributes. These attributes are dependent on the Sponsor_ID, ensuring that there are no transitive dependencies.

BCNF:

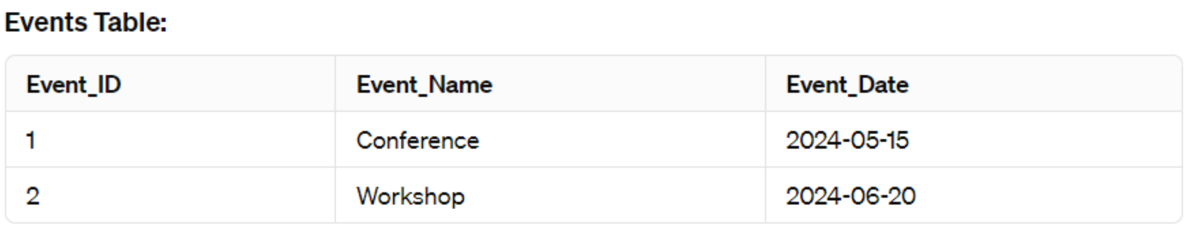
- The table must be in 3NF
- A table is in BCNF if every functional dependency X → Y, X is the super key of the table.



Event_ID	Speaker_ID	Speaker_Name	Event_Name	Event_Date
1	101	John Doe	Conference	2024-05-15
1	102	Jane Smith	Conference	2024-05-15
2	103	Mike Brown	Workshop	2024-06-20

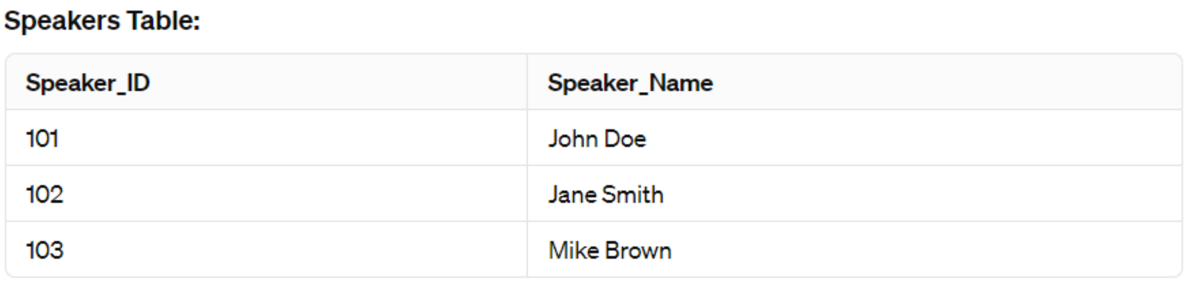
The original Event_Speakers table violated BCNF because Event_Name and Event_Date were functionally dependent on only Event_ID, not the full primary key. To fix this, we will decompose the table.

This new design will satisfy BCNF, as each table has a primary key that determines all other attributes



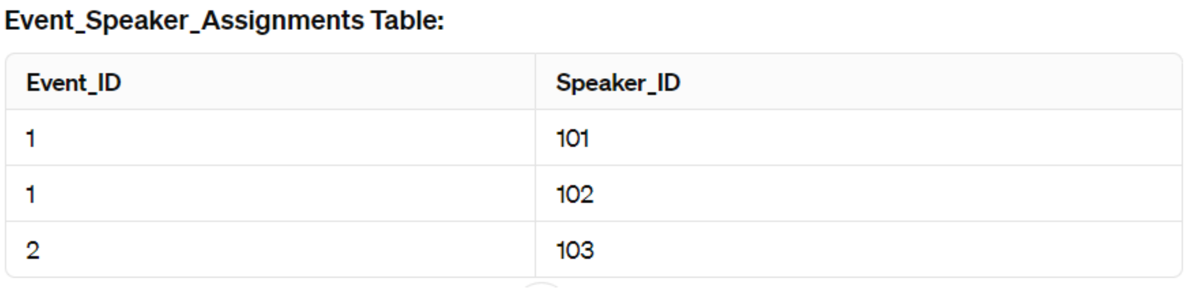
Events Table:

Event_ID	Event_Name	Event_Date
1	Conference	2024-05-15
2	Workshop	2024-06-20



Speakers Table:

Speaker_ID	Speaker_Name
101	John Doe
102	Jane Smith
103	Mike Brown



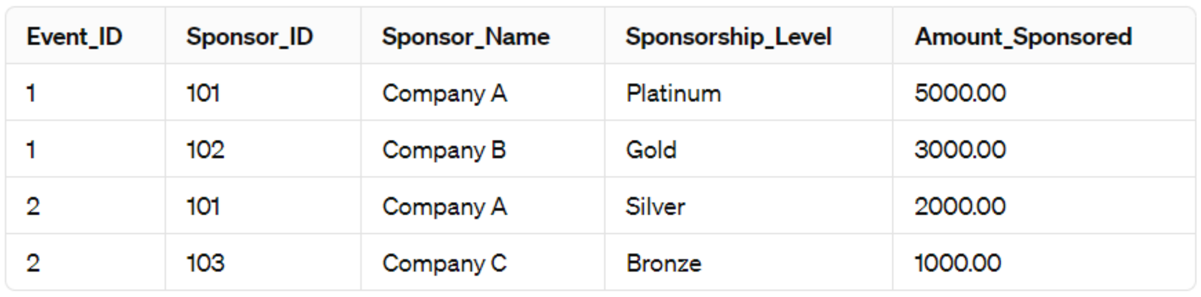
Event_Speaker_Assignments Table:

Event_ID	Speaker_ID
1	101
1	102
2	103

4NF:

- The table must be in BCNF.
- There should be no multi-valued dependency.

In the original Event_Sponsors table, the Sponsorship_Level was determined by Sponsor_ID, and the Amount_Sponsored was determined by Event_ID, violating 4NF.



Event_ID	Sponsor_ID	Sponsor_Name	Sponsorship_Level	Amount_Sponsored
1	101	Company A	Platinum	5000.00
1	102	Company B	Gold	3000.00
2	101	Company A	Silver	2000.00
2	103	Company C	Bronze	1000.00

To achieve 4NF, we decomposed the table into:

1. Event_Sponsorship_Levels: Holds Event_ID, Sponsor_ID, and Sponsorship_Level. This satisfies 4NF as the composite primary key determines all attributes.

2. Event_Sponsorship_Amounts: Holds Event_ID, Sponsor_ID, and Amount_Sponsored. This also satisfies 4NF as the composite primary key determines all attributes.

Event_Sponsorship_Levels Table:

Event_ID	Sponsor_ID	Sponsorship_Level
1	101	Platinum
1	102	Gold
2	101	Silver
2	103	Bronze

Event_Sponsorship_Amounts Table:

Event_ID	Sponsor_ID	Amount_Sponsored
1	101	5000.00
1	102	3000.00
2	101	2000.00
2	103	1000.00

5NF:

- The table must be in 4CNF.
- It should not contain any join dependency and joining should be lossless

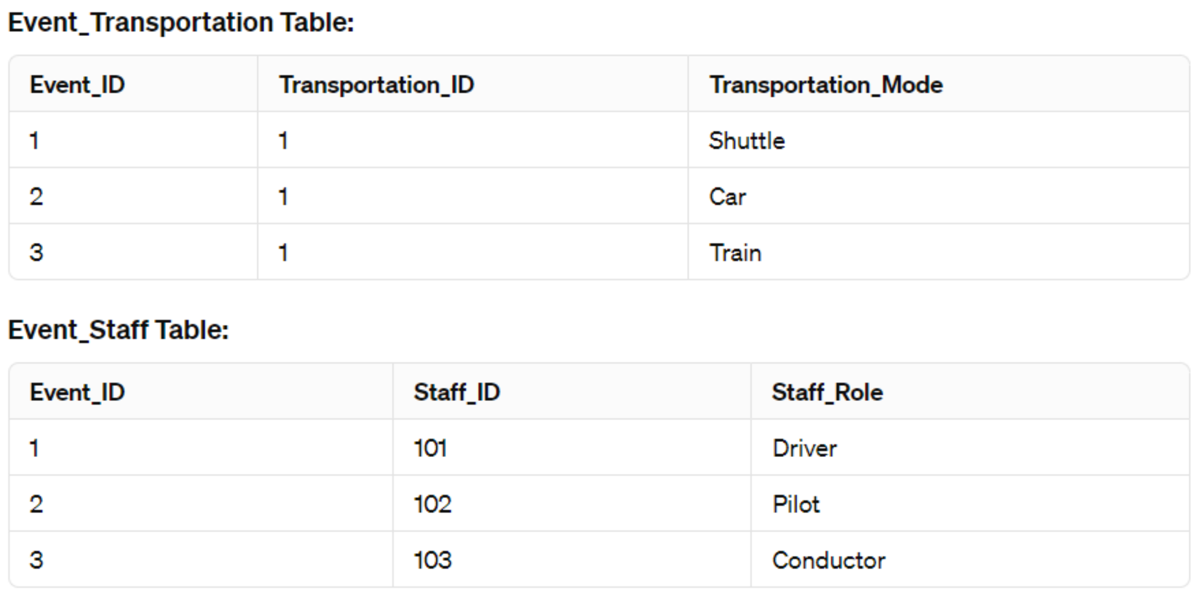
In the original Event_Logistics table, there were potential issues with multiple transportation modes and staff assignments per event, violating 5NF.

Event_ID	Transportation_Mode	Staff_ID
1	Shuttle	101
2	Car	102
3	Train	103

To achieve 5NF, we decomposed the table into:

1. Event_Transportation: Holds Event_ID, Transportation_ID, and Transportation_Mode. This table can have multiple rows per event, satisfying 5NF.

2. Event_Staff: Holds Event_ID, Staff_ID, and Staff_Role. This table can also have multiple rows per event, satisfying 5NF.



Event_Transportation Table:

Event_ID	Transportation_ID	Transportation_Mode
1	1	Shuttle
2	1	Car
3	1	Train

Event_Staff Table:

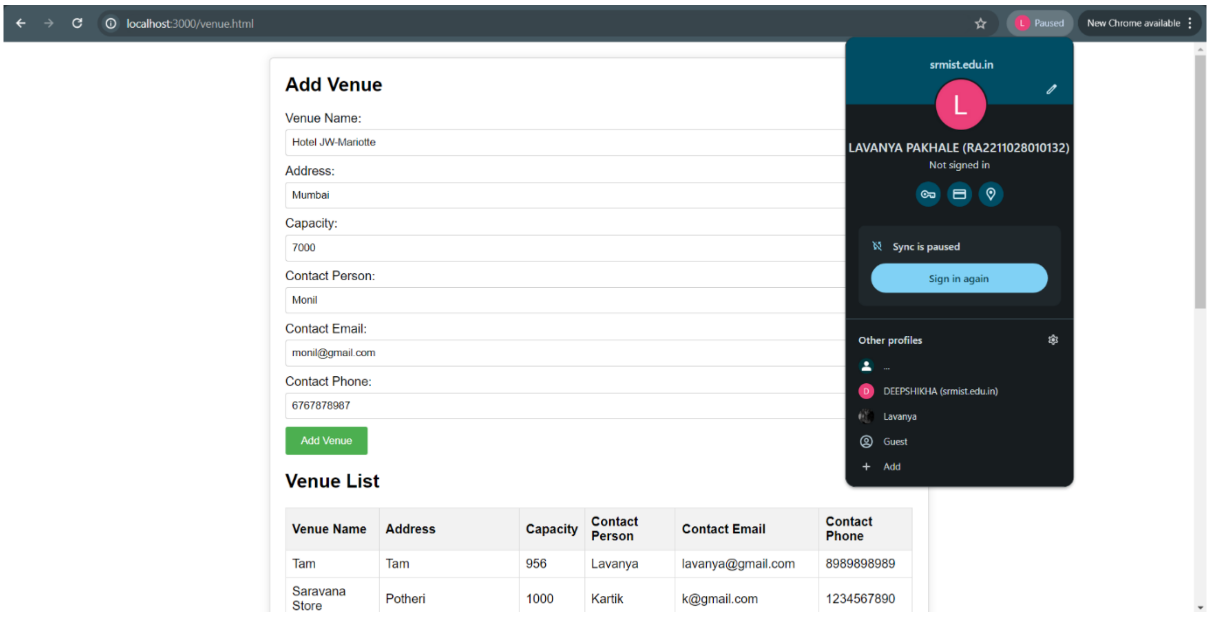
Event_ID	Staff_ID	Staff_Role
1	101	Driver
2	102	Pilot
3	103	Conductor

By separating the transportation and staff data into these two tables, we have eliminated the non-trivial join dependencies and ensured the design is in 5NF. This allows for more flexibility in managing the event logistics data, as transportation options and staff assignments can be added or modified without affecting the overall structure.

CHAPTER-5

CONCURRENCY CONTROL

Addition of Event details in two different systems:


localhost:3000/venue.html

srmist.edu.in
LAVANYA PAKHALE (RA2211028010132)
Not signed in
Sync is paused
Sign in again
Other profiles
DEEPSHIKHA (srmist.edu.in)
Lavanya
Guest
Add

Add Venue

Venue Name: Hotel JW-Mariotte

Address: Mumbai

Capacity: 7000

Contact Person: Monil

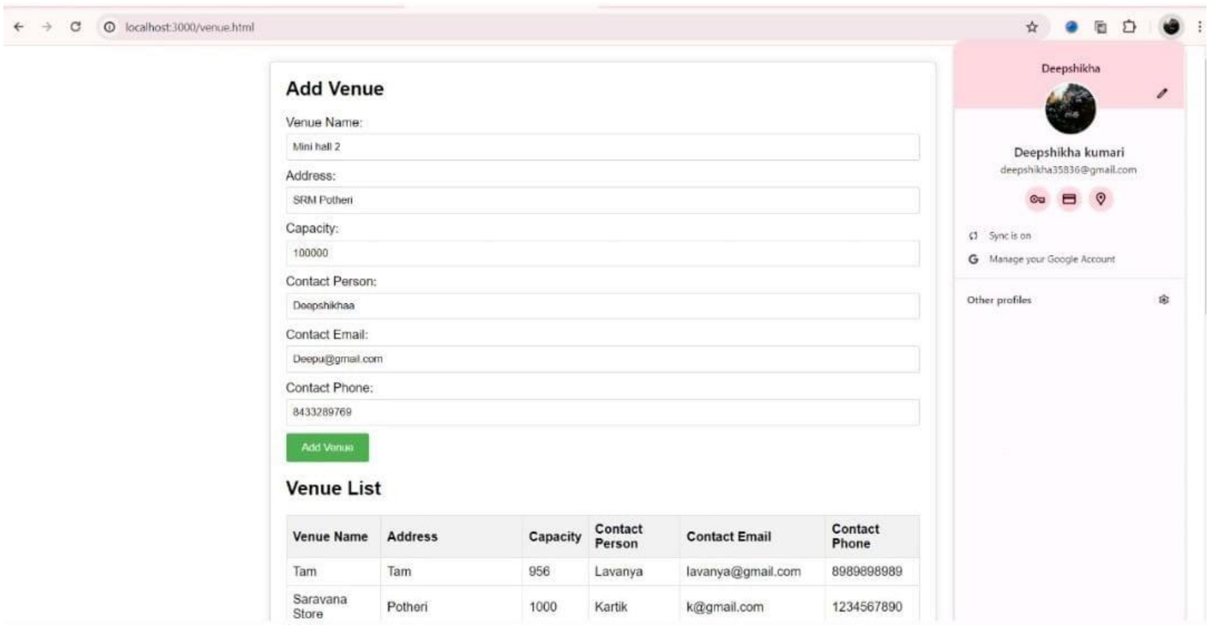
Contact Email: monil@gmail.com

Contact Phone: 6767878987

Add Venue

Venue List

Venue Name	Address	Capacity	Contact Person	Contact Email	Contact Phone
Tam	Tam	956	Lavanya	lavanya@gmail.com	8989898989
Saravana Store	Potheri	1000	Kartik	k@gmail.com	1234567890


localhost:3000/venue.html

Deepshikha
Deepshikha kumari
deepshikha35836@gmail.com
Sync is on
Manage your Google Account
Other profiles

Add Venue

Venue Name: Mini hall 2

Address: SRM Potheri

Capacity: 100000

Contact Person: Deepshikhaa

Contact Email: Deepu@gmail.com

Contact Phone: 8433289769

Add Venue

Venue List

Venue Name	Address	Capacity	Contact Person	Contact Email	Contact Phone
Tam	Tam	956	Lavanya	lavanya@gmail.com	8989898989
Saravana Store	Potheri	1000	Kartik	k@gmail.com	1234567890

CHAPTER – 6

Code for the project:

Events html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Event Management</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    .container {
      max-width: 800px;
      margin: 20px auto;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    h2 {
      margin-top: 0;
    }
    label {
      display: block;
      margin-bottom: 5px;
    }
    input[type="text"], input[type="date"], input[type="time"], input[type="number"], textarea
  {
      width: 100%;
      padding: 8px;
      margin-bottom: 10px;
      border: 1px solid #ccc;
      border-radius: 3px;
      box-sizing: border-box;
    }
    button {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 3px;
      cursor: pointer;
    }
    button:hover {
      background-color: #45a049;
    }
    .event-list {
      margin-top: 20px;
    }
    .event-item {
      margin-bottom: 10px;
      padding: 10px;
      border: 1px solid #eee;
      border-radius: 3px;
    }
```

```
    table {
      width: 100%;
      border-collapse: collapse;
    }
th, td {
      border: 1px solid #ddd;
      padding: 8px;
      text-align: left;
    }
 th {
      background-color: #f2f2f2;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Add Event</h2>
    <form id="addEventForm">
      <label for="eventName">Event Name:</label>
      <input type="text" id="eventName" name="eventName" required>
      <label for="eventDate">Event Date:</label>
      <input type="date" id="eventDate" name="eventDate" required>
      <label for="eventTime">Event Time:</label>
      <input type="time" id="eventTime" name="eventTime" required>
      <label for="venueName">Venue Name:</label>
      <input type="text" id="venueName" name="venueName" required>
      <label for="description">Description:</label>
      <textarea id="description" name="description" required></textarea>
      <button type="submit">Add Event</button>
    </form>
  <div class="event-list">
   <h2>Event List</h2>
   <table id="eventList">
    <thead>
     <tr>
      <th>Event Name</th>
      <th>Date</th>
      <th>Time</th>
      <th>Venue Name</th>
      <th>Description</th>
     </tr>
    </thead>
    <tbody></tbody>
   </table>
  </div>
 </div>
<script>
  document.addEventListener('DOMContentLoaded', async () => {
   await fetchEvents();
  });
  async function fetchEvents() {
   const response = await fetch('http://localhost:3000/event');
   const events = await response.json();
   const eventList = document.querySelector('#eventList tbody');
   eventList.innerHTML = '';
   events.forEach(event => {
    const row = document.createElement('tr');
    const date = new Date(event[1]);
    const year = date.getFullYear();
    const month = String(date.getMonth() + 1).padStart(2, '0');
    const day = String(date.getDate()).padStart(2, '0');
    const formattedDate = `${day}-${month}-${year}`;
     const time = new Date(event[2]);
    const hours = String(time.getHours()).padStart(2, '0');
    const minutes = String(time.getMinutes()).padStart(2, '0');
    const formattedTime = `${hours}:${minutes}`;
    row.innerHTML = `
      <td>${event[0]}</td>
      <td>${formattedDate}</td>
      <td>${formattedTime}</td>
      <td>${event[3]}</td>
      <td>${event[4]}</td>
    `;
    eventList.appendChild(row);
  });
}
document.getElementById('addEventForm').addEventListener('submit', async (event) => {
  event.preventDefault();
  const formData = new FormData(event.target);
  const eventData = Object.fromEntries(formData.entries());

  const response = await fetch('http://localhost:3000/event', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(eventData)
  });

  if (response.ok) {
        await fetchEvents();
        event.target.reset();
      } else {
        console.error('Failed to add event');
      }
    });
  </script>
</body>
</html>
```

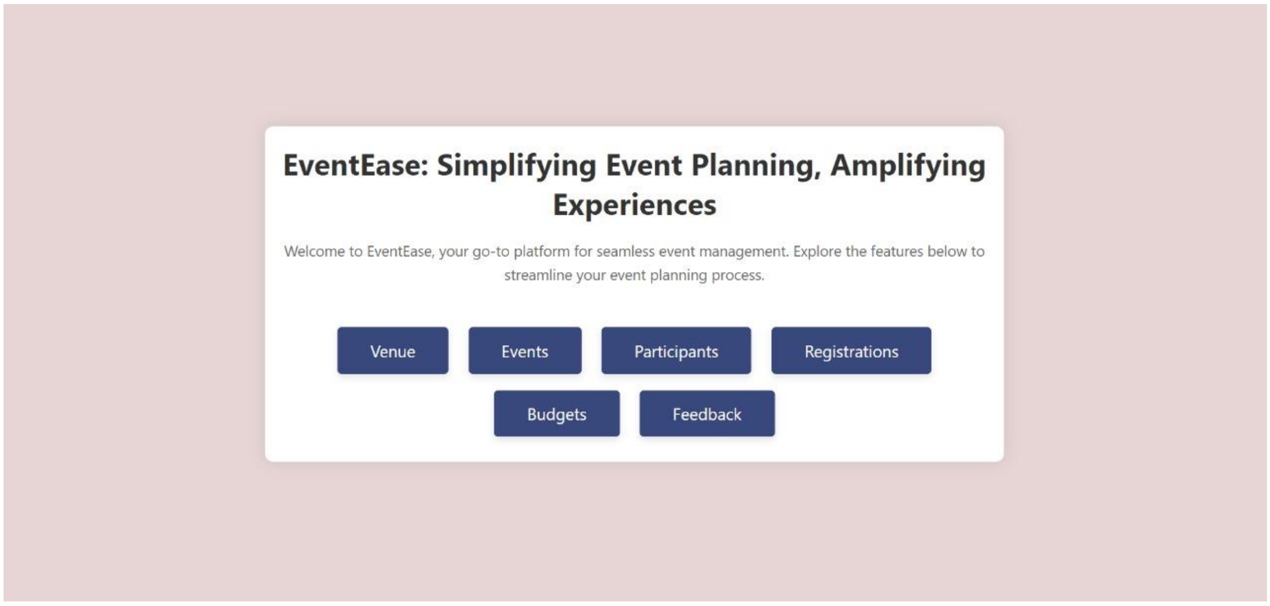
Node.js code:

```
app.get('/event', async (req, res) => {
  let connection;
  try {
    connection = await oracledb.getConnection();
    const result = await connection.execute('SELECT E.EVENT_NAME, E.EVENT_DATE,
E.EVENT_TIME, V.VENUE_NAME, E.DESCRIPTION FROM EVENTS E LEFT JOIN
VENUES V ON E.VENUE_ID = V.VENUE_ID');
    res.json(result.rows);
  } catch (err) {
    console.error('Error fetching events:', err);
    res.status(500).send('Internal Server Error');
  } finally {
    if (connection) {
      try {
        await connection.close();
      } catch (err) {
        console.error('Error closing connection:', err);
      }
    }
  }
```

```
});
app.post('/event', async (req, res) => {
  const { eventName, eventDate, eventTime, venueName, description } = req.body;
  let connection;
 try {
    connection = await oracledb.getConnection(); // Get a connection from the pool
    const venueIdQuery = `SELECT VENUE_ID FROM VENUES WHERE VENUE_NAME
= :venueName`;
    const venueIdResult = await connection.execute(venueIdQuery, [venueName]);
    const venue_id = venueIdResult.rows[0][0];
    const insertQuery = `
      INSERT INTO Events (Event_Name, Event_Date, Event_Time, Venue_ID, Description)
      VALUES (:1, TO_DATE(:2, 'YYYY-MM-DD'), TO_TIMESTAMP(:3, 'YYYY-MM-
DD"T"HH24:MI:SS'), :4, :5)`;
    const formattedTime = eventTime + ':00'; // Assuming eventTime is in HH:MM format
      const newEvent = [eventName, eventDate, `${eventDate}T${formattedTime}`, venue_id,
description];
    await connection.execute(insertQuery, newEvent);
    await connection.commit();
console.log('Event added successfully');
    res.sendStatus(200);
  } catch (err) {
    if (connection) {
      try {        await connection.execute(`ROLLBACK`);
      } catch (rollbackErr) {
        console.error('Error rolling back transaction:', rollbackErr);
      }
    }
    console.error('Error adding event:', err);
    res.status(500).send('Internal Server Error');
  } finally {
    if (connection) {
      try {
        await connection.close(); // Release the connection back to the pool
      } catch (closeErr) {
        console.error('Error closing connection:', closeErr);
      }
    }
  }
});
```

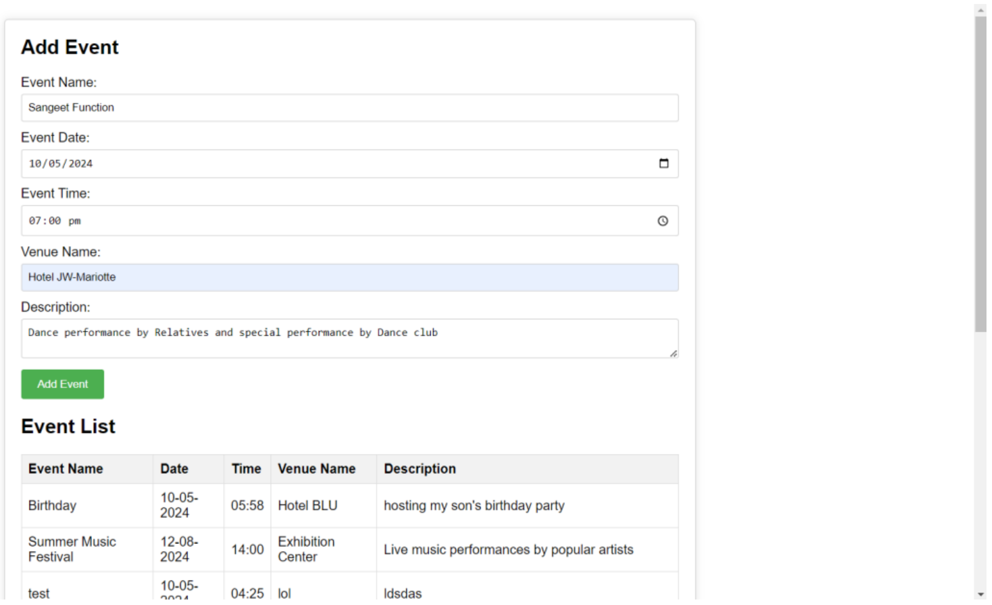
CHAPTER – 7

Result and Discussions:



EventEase: Simplifying Event Planning, Amplifying Experiences

Welcome to EventEase, your go-to platform for seamless event management. Explore the features below to streamline your event planning process.



Add Event

Event Name: Sangeet Function

Event Date: 10/05/2024

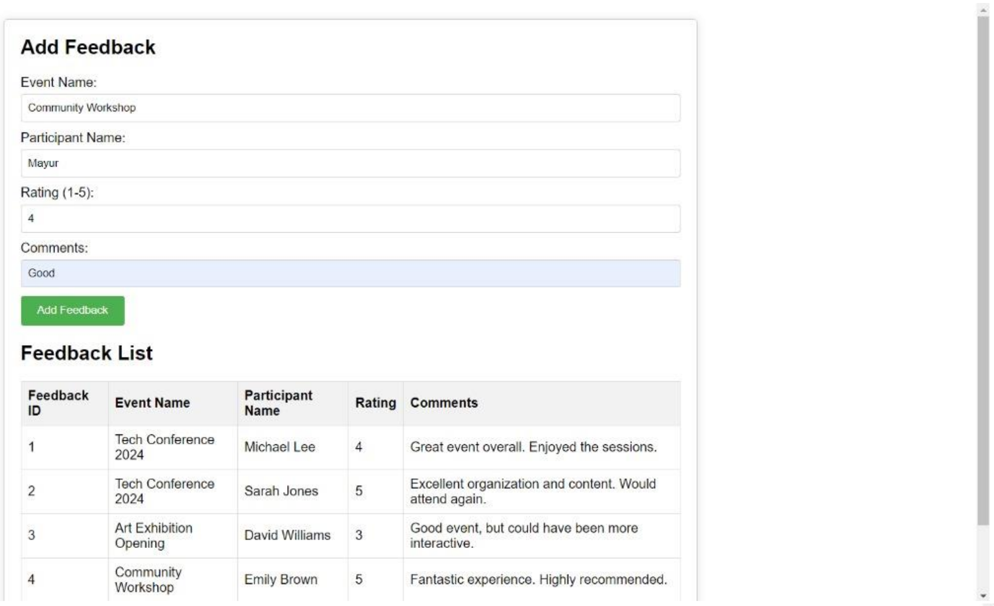
Event Time: 07:00 pm

Venue Name: Hotel JW-Mariotte

Description: Dance performance by Relatives and special performance by Dance club

Event List

Event Name	Date	Time	Venue Name	Description
Birthday	10-05-2024	05:58	Hotel BLU	hosting my son's birthday party
Summer Music Festival	12-08-2024	14:00	Exhibition Center	Live music performances by popular artists
test	10-05-	04:25	lol	ldsdas

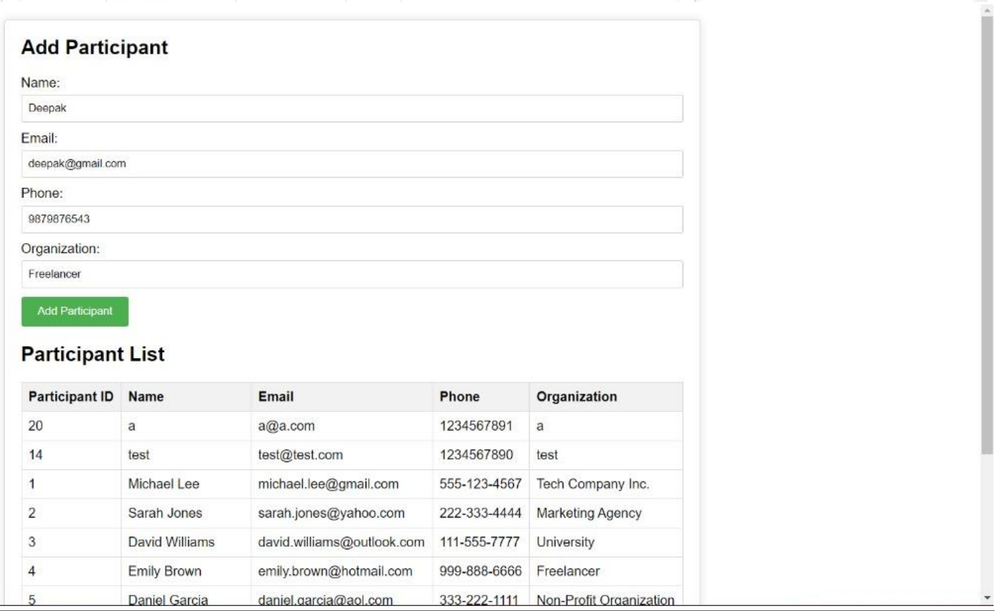


Add Feedback

Event Name:	Community Workshop
Participant Name:	Mayur
Rating (1-5):	4
Comments:	Good

Feedback List

Feedback ID	Event Name	Participant Name	Rating	Comments
1	Tech Conference 2024	Michael Lee	4	Great event overall. Enjoyed the sessions.
2	Tech Conference 2024	Sarah Jones	5	Excellent organization and content. Would attend again.
3	Art Exhibition Opening	David Williams	3	Good event, but could have been more interactive.
4	Community Workshop	Emily Brown	5	Fantastic experience. Highly recommended.

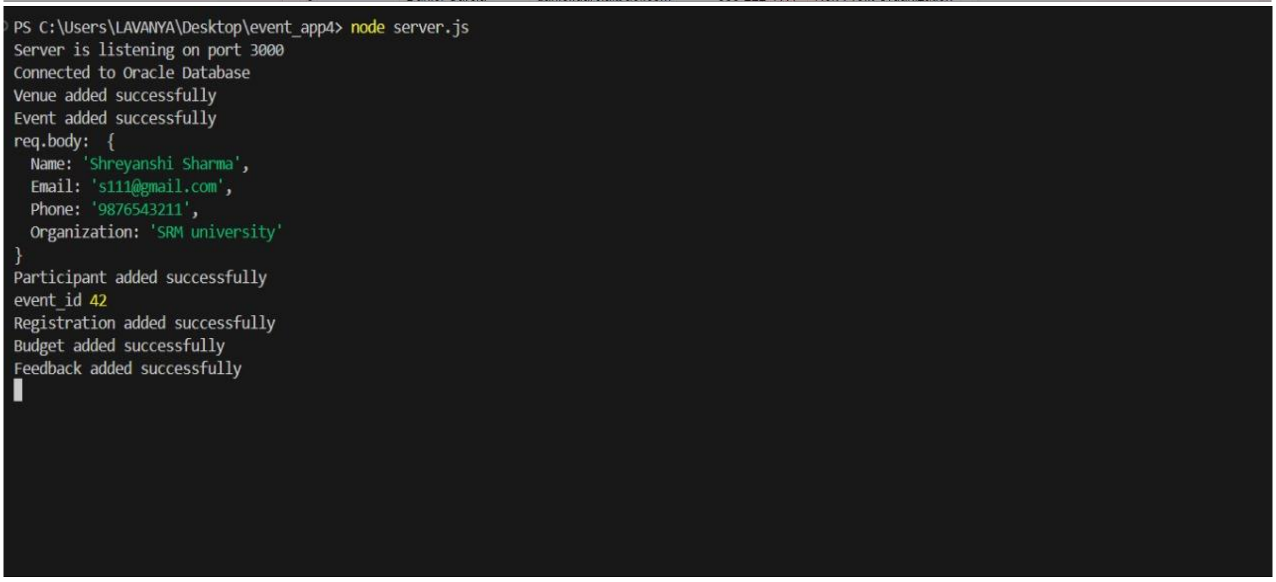


Add Participant

Name:	Deepak
Email:	deepak@gmail.com
Phone:	9879876543
Organization:	Freelancer

Participant List

Participant ID	Name	Email	Phone	Organization
20	a	a@a.com	1234567891	a
14	test	test@test.com	1234567890	test
1	Michael Lee	michael.lee@gmail.com	555-123-4567	Tech Company Inc.
2	Sarah Jones	sarah.jones@yahoo.com	222-333-4444	Marketing Agency
3	David Williams	david.williams@outlook.com	111-555-7777	University
4	Emily Brown	emily.brown@hotmail.com	999-888-6666	Freelancer
5	Daniel Garcia	daniel.garcia@aol.com	333-222-1111	Non-Profit Organization



```
PS C:\Users\LAVANYA\Desktop\event_app4> node server.js
Server is listening on port 3000
Connected to Oracle Database
Venue added successfully
Event added successfully
req.body:  {
  Name: 'Shreyanshi Sharma',
  Email: 's111@gmail.com',
  Phone: '9876543211',
  Organization: 'SRM university'
}
Participant added successfully
event_id 42
Registration added successfully
Budget added successfully
Feedback added successfully
```

CHAPTER – 8

In conclusion, our event management system, EventEase presents a comprehensive solution designed to address the various requirements of organizing and executing diverse events. Its robust architecture contains a variety of functionalities tastefully designed to cater to the needs of event planners, participants, sponsors, staff, vendors, and other stakeholders involved in the event ecosystem.

At its core, the system ensures efficient data management, serving as a centralized repository for crucial information pertaining to venues, participants, speakers, staff, sponsors, budgets, registrations, feedback, vendors, transportation, tasks, sessions, and contracts. This consolidation of data streamlines the planning and coordination process, facilitating seamless execution from inception to completion. Event planners benefit from the system's versatile tools, which enable them to meticulously plan and coordinate various aspects of events. From selecting suitable venues to allocating budgets, managing ticket sales, overseeing registrations, assigning tasks, scheduling sessions, arranging transportation, and finalizing vendor contracts, the system offers a comprehensive suite of features to support every stage of event management.

Moreover, the system fosters participant and speaker engagement by providing intuitive registration processes, ticketing options, and feedback mechanisms. It also empowers sponsors to contribute to events at different levels, while offering transparent visibility into sponsorship amounts and associated benefits. Additionally, staff roles and responsibilities are efficiently managed, ensuring effective collaboration and task execution. With its robust financial tracking capabilities, the system enables organizers to monitor event budgets, track expenditures, and manage financial transactions with clarity and accountability. Participant feedback mechanisms aid in evaluating event success and identifying areas for improvement, facilitating continual refinement of the event experience.

In summary, the event management system serves as a powerful tool for enhancing collaboration, streamlining processes, and optimizing the overall event experience. Through further refinements and enhancements, it has the potential to become an indispensable asset for event management professionals seeking to orchestrate successful events with precision and efficiency.

CHAPTER – 9

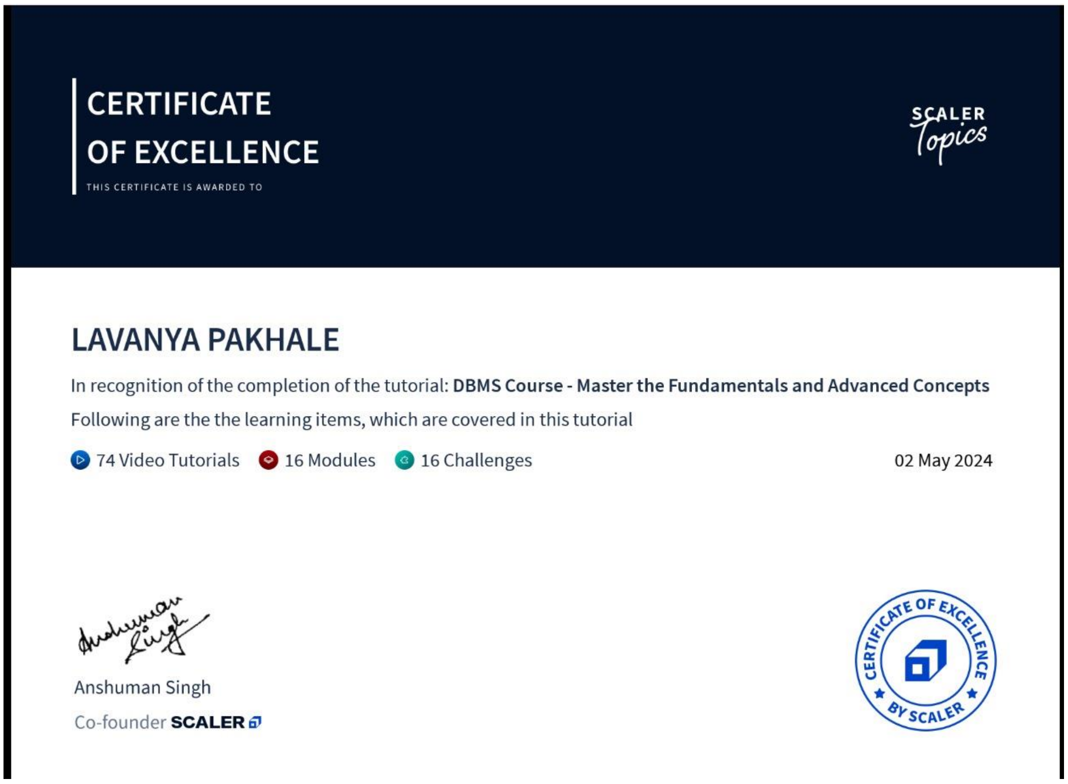
Future Scope of EventEase:

After analysing our event management system, we have found out that there's ample scope for future enhancements. Improving user interface intuitiveness and adding features like calendar views and drag-and-drop functionality could enhance user experience. Integration with payment gateways would enable online ticket sales, while a dedicated mobile application could extend accessibility.

Real-time analytics would aid in tracking event metrics, and social media integration could amplify event promotion. Personalization algorithms could tailor event suggestions, and multi-language support would widen accessibility. Automation and AI technologies could streamline tasks, and support for virtual and hybrid events would cater to evolving formats.

Lastly, incorporating sustainability initiatives would align with environmental concerns. These enhancements collectively promise a more efficient, engaging, and environmentally conscious event management system.

CHAPTER – 10



CERTIFICATE
OF EXCELLENCE
THIS CERTIFICATE IS AWARDED TO

SCALER Topics

LAVANYA PAKHALE

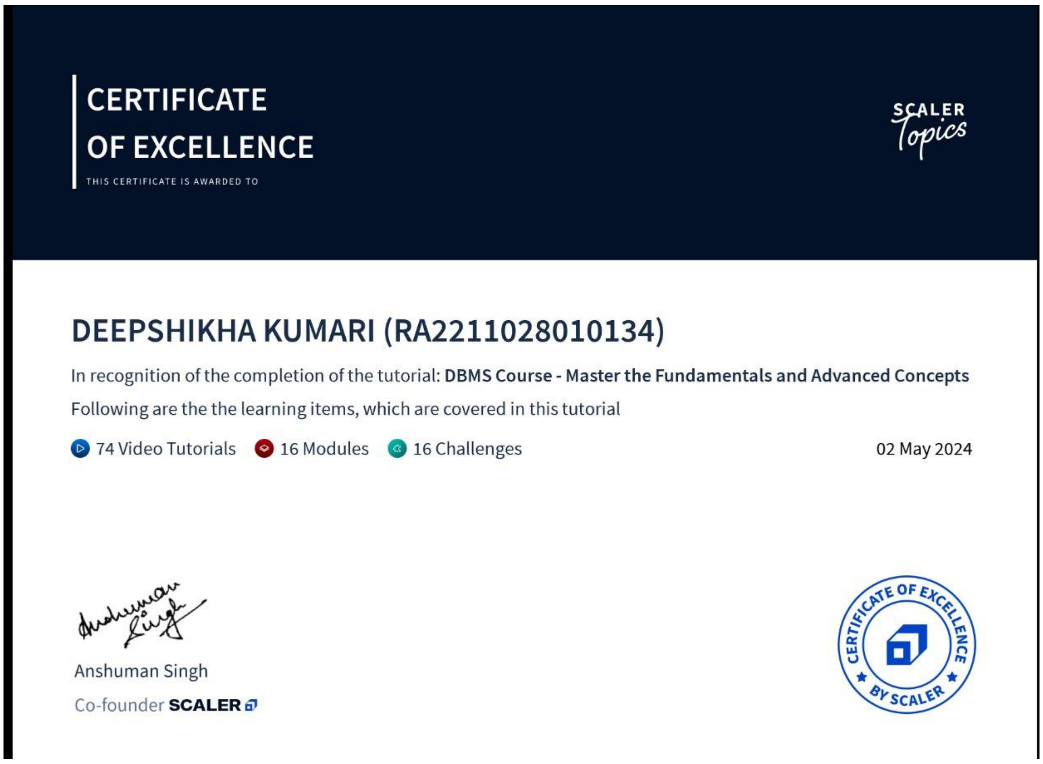
In recognition of the completion of the tutorial: **DBMS Course - Master the Fundamentals and Advanced Concepts**

Following are the the learning items, which are covered in this tutorial

74 Video Tutorials 16 Modules 16 Challenges 02 May 2024

[illegible]
Anshuman Singh
Co-founder **SCALER**

CERTIFICATE OF EXCELLENCE ★ BY SCALER ★



CERTIFICATE
OF EXCELLENCE
THIS CERTIFICATE IS AWARDED TO

SCALER Topics

DEEPSHIKHA KUMARI (RA2211028010134)

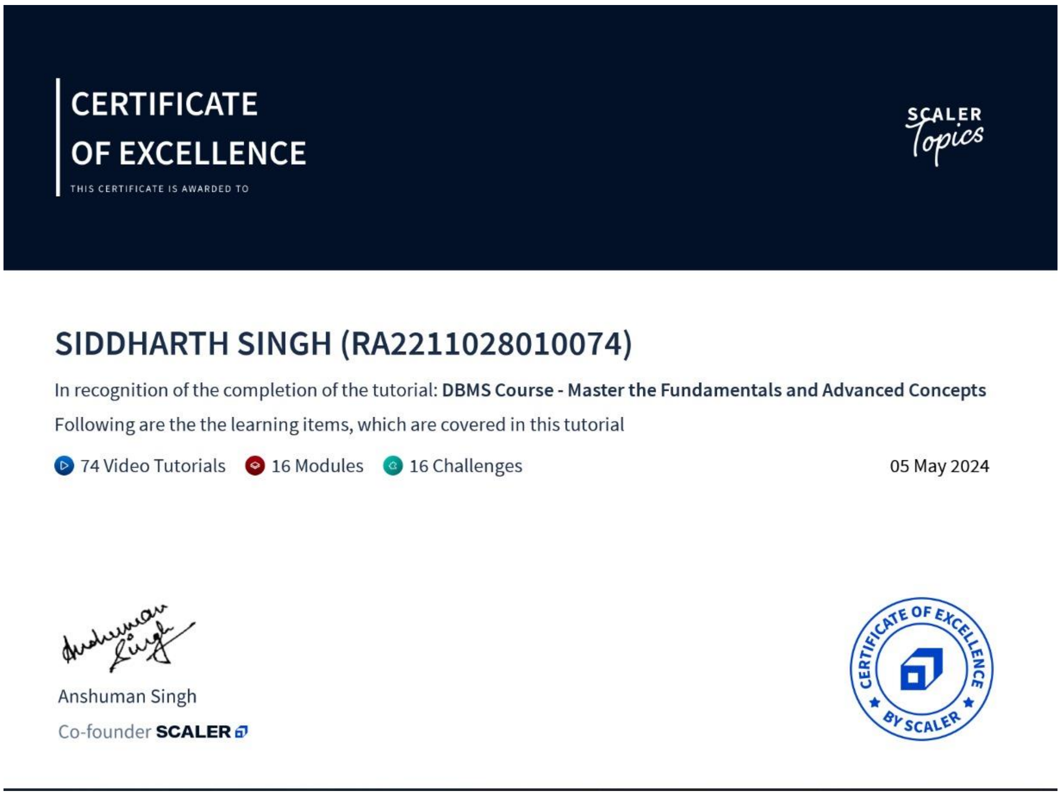
In recognition of the completion of the tutorial: **DBMS Course - Master the Fundamentals and Advanced Concepts**

Following are the the learning items, which are covered in this tutorial

74 Video Tutorials 16 Modules 16 Challenges 02 May 2024

[illegible]
Anshuman Singh
Co-founder **SCALER**

CERTIFICATE OF EXCELLENCE ★ BY SCALER ★



CERTIFICATE
OF EXCELLENCE

THIS CERTIFICATE IS AWARDED TO

SCALER Topics

SIDDHARTH SINGH (RA2211028010074)

In recognition of the completion of the tutorial: **DBMS Course - Master the Fundamentals and Advanced Concepts**

Following are the the learning items, which are covered in this tutorial

74 Video Tutorials 16 Modules 16 Challenges 05 May 2024

[illegible]

Anshuman Singh

Co-founder **SCALER**

CERTIFICATE OF EXCELLENCE BY SCALER