



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**UNIVERSITI TEKNOLOGI MALAYSIA, 81310, UTM JOHOR BAHRU, JOHOR,  
MALAYSIA**

**SECD2523-03 (DATABASE)**

**Section - 03**

**GROUP CAPYBARA**

**PROJECT PHASE 3**

<b>STUDENTS NAME</b>	<b>NO. MATRIC</b>
DANIAL ERFAN SHAH BIN NOR AZAM SHAH	A22EC0151
MEGAT MUHAMMAD ZAFRAN BIN MEGAT MUAZZAM	A22EC0194
MUHAMMAD ARIF FIKRY BIN NOOR KHARIZAN	A22EC0203
ADAM FAHMI BIN MOHD ADNAN	A22EC0032
RUBILLAN A/L SUKUMAHARAN	A22EC0266

**LECTURER'S NAME :**

**DR. IZYAN IZZATI BINTI KAMSANI**

# **Table Of Contents**

<b>1.0 Introduction</b>	<b>3</b>
<b>2.0 Overview</b>	<b>4</b>
<b>3.0 Database Conceptual Design</b>	<b>5</b>
<b>3.1 Updated Business Rule</b>	<b>5</b>
<b>3.2 Conceptual ERD</b>	<b>6</b>
<b>4.0 Database Logical Design</b>	<b>7</b>
<b>4.1 Logical ERD</b>	<b>7</b>
<b>4.3 Updated Data Dictionary</b>	<b>9</b>
<b>4.4 Normalization</b>	<b>11</b>
<b>5.0 Relational DB Schemas</b>	<b>13</b>
<b>6.0 SQL Statements (DDL &amp; DML)</b>	<b>14</b>
<b>Conclusion</b>	<b>34</b>

## **1.0 Introduction**

Nexscholar is a basic social platform that facilitates connections among students and clients. Some may say that Nexscholar is similar to social media giants like Facebook, Twitter, and Instagram due to its similar features. The current social media landscape faces criticism for its misuse among university students and spreading hate and anger. Nexscholar aims to harness the positive aspects of social media while incorporating an interactive learning approach. Becoming a social media platform is one of Nexscholar's primary objectives.

Besides that, the website features are divided into several categories, including post-graduate, undergraduate, researcher, and job search, each tailored to the specific needs of its users. Certain features may be unique to one another as all of them have different requirements from users.

Nexscholar aspires to be an all-in-one hub for students and lecturers to connect, share ideas, and engage in learning activities. The platform seeks to reduce the barriers between students and lecturers, cultivating inspiring and engaging interaction.

Despite its noble goals, Nexscholar faces challenges and is far from being a perfect website. As mentioned, it is a newborn website focusing on fundamental functionality without offering many advanced features. This report aims to address the issues Nexscholar is facing, particularly in event management, and propose efficient solutions to help it contrive into a better future.

## **2.0 Overview**

Nexscholar is a basic social platform that facilitates connections among students and clients. Some may say that Nexscholar is similar to social media giants like Facebook, Twitter, and Instagram due to its similar features. The current social media landscape faces criticism for its misuse among university students and the spewing of hate and anger. Nexscholar aims to harness the positive aspects of social media while incorporating an interactive learning approach. Becoming a social media platform is one of Nexscholar's primary objectives.

Besides that, the website features are divided into several categories, including post-graduate, undergraduate, researcher, and job search, each tailored to the specific needs of its users. Certain features may be unique to one another as all of them have different requirements from users.

Nexscholar aspires to serve as an all-in-one hub for students and lecturers to connect, share ideas, and engage in their learning activities. The platform seeks to reduce the barriers between students and lecturers, hence cultivating an inspiring and engaging interaction.

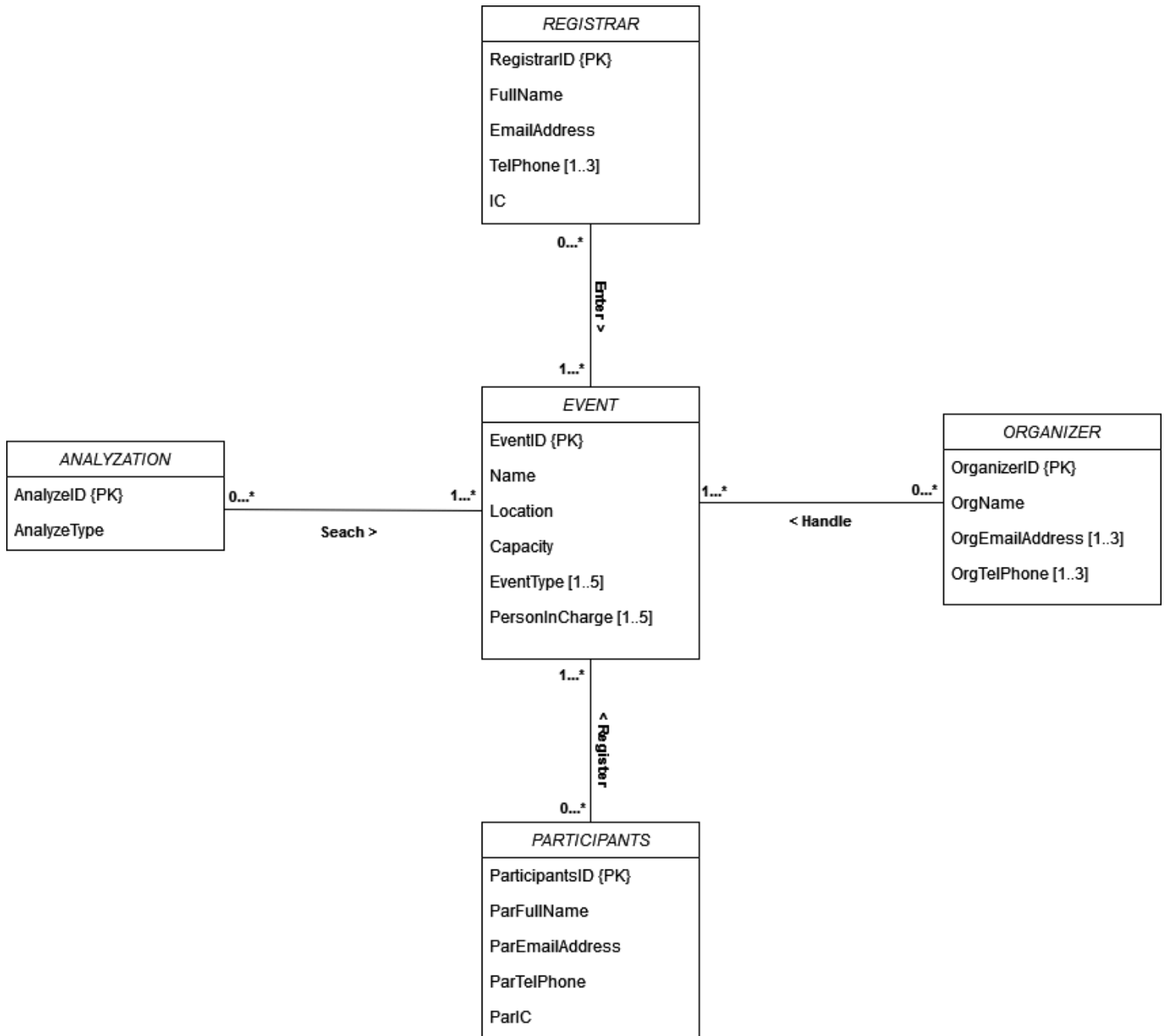
Despite its noble goals, Nexscholar faces challenges and is far from being a perfect website. As mentioned earlier, it is a newborn website that focuses on fundamental functionality without offering many advanced features. This report aims to address the issues Nexscholar is facing, particularly in event management, and propose efficient solutions to help it contrive into a better future.

### **3.0 Database Conceptual Design**

#### **3.1 Updated Business Rule**

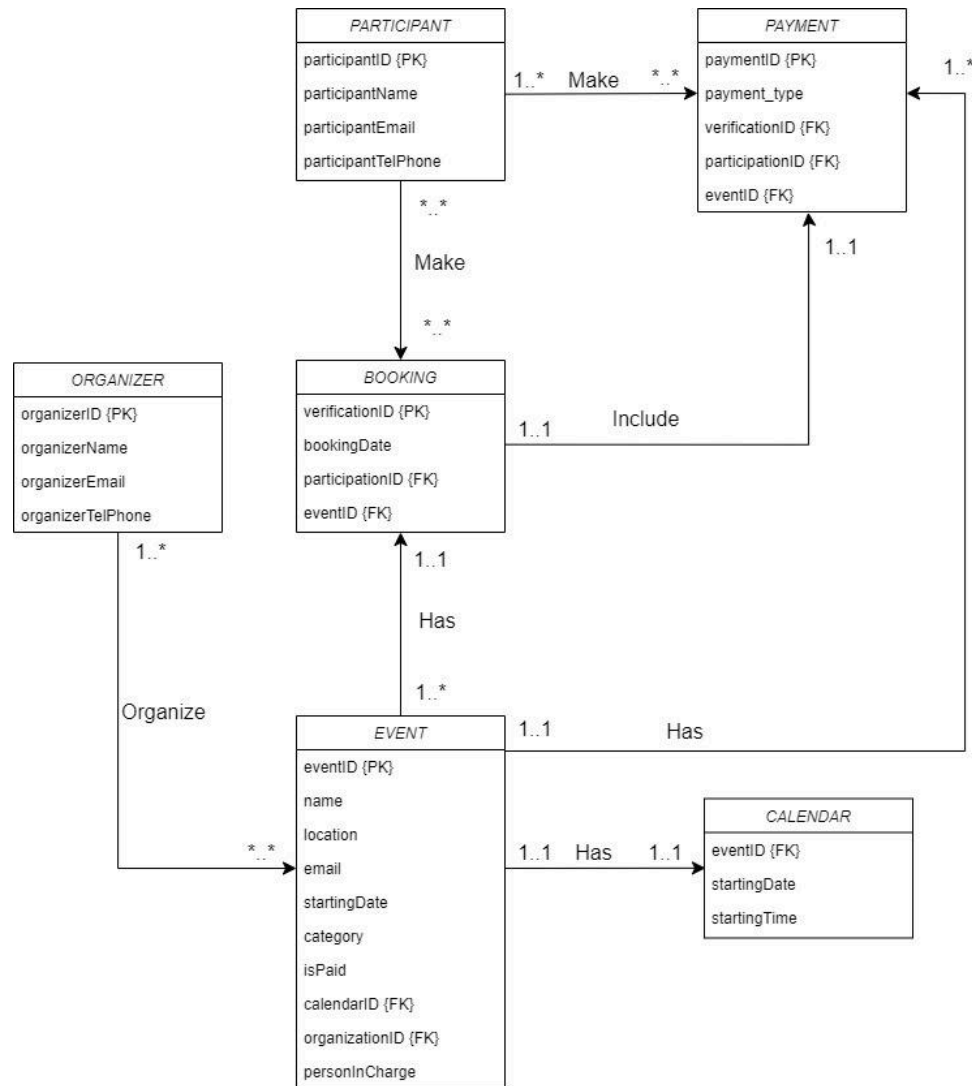
1. The system is working almost 23 hours nonstop
2. Maintenance can be done every 3 am until 4 am
3. Organizers need to contact the Registrar to publish an event
4. Organizers must provide the information regarding the Merchant details and Payment details which is mandatory for all Organizers who wish to promote their event with us.
5. The registrar must receive complete event information, including Person In Charge (PIC)
6. Only selected people in Registrar and Organizer are able to publish an event on the website.
7. Paid event will be included with merchant details that is encrypted
8. Participants will need to register themselves on the website before participating in any event
9. Participants' payment towards the organizer merchant will be recorded
10. Participants must have a valid bank account to pay to register for a paid event
11. Only one participant is able to pay for their own paid event, third party payment is not allowed.
12. Every registered participant will be included with a verification ID
13. Participants are able to see all the information regarding the events in the website but they are not able to modify it
14. Each participant will have a calendar interface that shows the upcoming events that they joined
15. Every event that exists in the website must have a valid date and time as it is mandatory.
16. All events can be analyzed
17. To do an analysis in the database, information of what to analyze must be inputted.
18. Participants are unable to use and see the analyzation feature as they do not have enough clearance

### 3.2 Conceptual ERD

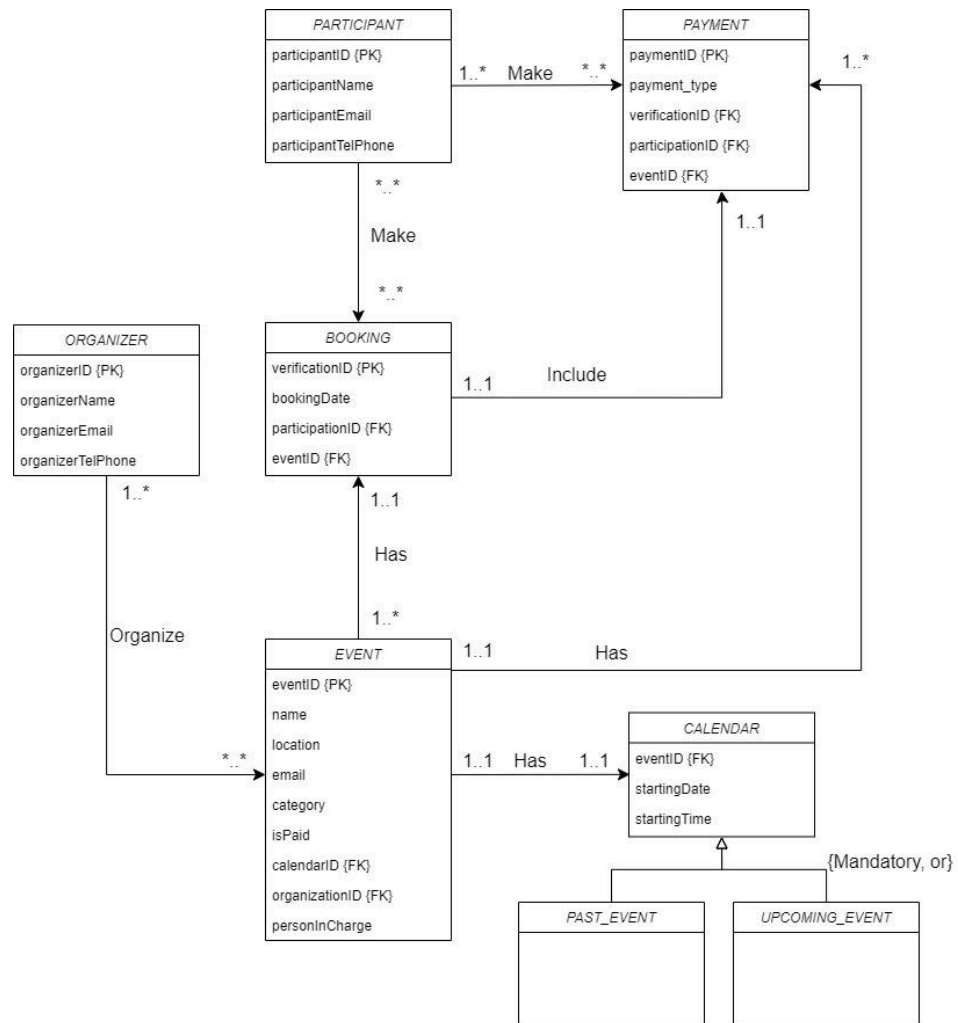


## 4.0 Database Logical Design

### 4.1 Logical ERD



## 4.2 Enhanced ERD





### 4.3 Updated Data Dictionary

Entity Name	Attributes	Data Type & Length	Nulls	Constraint
EVENT	eventID	VARCHAR2(15)	No	PRIMARY KEY
	name	VARCHAR2(100)	No	NOT NULL
	location	VARCHAR2(100)	No	NOT NULL
	email	VARCHAR2(100)	No	NOT NULL
	category	VARCHAR2(100)	No	NOT NULL
	isPaid	VARCHAR2(1)	No	NOT NULL
	organizationID	VARCHAR2(15)	No	FOREIGN KEY
	personInCharge	VARCHAR2(100)	No	NOT NULL
PAYMENT	paymentID	VARCHAR2(15)	No	PRIMARY KEY
	payment_type	VARCHAR2(15)	No	NOT NULL
	verificationID	VARCHAR2(15)	No	FOREIGN KEY
	participationID	VARCHAR2(15)	No	FOREIGN KEY
	eventID	VARCHAR2(15)	No	FOREIGN KEY
BOOKING	verificationID	VARCHAR2(15)	No	PRIMARY KEY
	bookingDate	DATE	No	NOT NULL
	participantID	VARCHAR2(15)	No	FOREIGN KEY
	eventID	VARCHAR2(15)	No	FOREIGN KEY

ORGANIZER	organizerID	VARCHAR2(15)	No	PRIMARY KEY
	organizerName	VARCHAR2(100)	No	NOT NULL
	organizerEmail	VARCHAR2(100)	No	NOT NULL
	organizerTelPhone	VARCHAR2(12)	No	NOT NULL
PARTICIPANT	participantID	VARCHAR2(15)	No	PRIMARY KEY
	participantName	VARCHAR2(100)	No	NOT NULL
	participantEmail	VARCHAR2(100)	No	NOT NULL
	participantTelPhone	VARCHAR2(15)	No	NOT NULL
CALENDAR	calendarID	VARCHAR2(15)	No	PRIMARY KEY
	startingDate	DATE	No	NOT NULL
	eventID	VARCHAR2(15)	No	FOREIGN KEY

## 4.4 Normalization

1. ORGANIZER (organizerID, organizerName, organizerEmail, organizerTelPhone)  
**fd1:** organizerID  $\rightarrow$  organizerName, organizerEmail, organizerTelPhone  
**1NF & 2NF & 3NF & BCNF:**  
ORGANIZER (organizerID, organizerName, organizerEmail, organizerTelPhone)
2. PARTICIPANT (participantID, participantName, participantEmail, participantTelPhone)  
**fd1:** participantID  $\rightarrow$  participantName, participantEmail, participantTelPhone  
**1NF & 2NF & 3NF & BCNF:**  
PARTICIPANT (participantID, participantName, participantEmail, participantTelPhone)
3. EVENT (eventID, name, location, email, category, isPaid, organizationID, personInCharge)  
**fd1:** eventID  $\rightarrow$  name, location, email, category, isPaid, organizationID  
**1NF & 2NF & 3NF & BCNF:**  
EVENT (eventID, name, location, email, category, isPaid, organizationID, personInCharge)
4. BOOKING (verificationID, bookingDate, participationID, eventID)  
**fd1:** verificationID  $\rightarrow$  bookingDate, participationID, eventID  
**1NF & 2NF & 3NF & BCNF:**  
BOOKING (verificationID, bookingDate, participationID, eventID)
5. PAYMENT (paymentID, payment\_type, verificationID, participationID, eventID)  
**fd1:** paymentID  $\rightarrow$  payment\_type, verificationID, participationID, eventID  
**1NF & 2NF & 3NF & BCNF:**  
PAYMENT (paymentID, payment\_type, verificationID, participationID, eventID)
6. CALENDAR (calendarID, eventID, startingDate)  
**fd1:** eventID  $\rightarrow$  startingDate  
**1NF & 2NF & 3NF & BCNF:**  
CALENDAR (calendarID, eventID, startingDate)
7. PAST\_EVENT (calendarID, eventID, startingDate)  
**fd1:** eventID  $\rightarrow$  startingDate  
**1NF & 2NF & 3NF & BCNF:**  
PAST\_EVENT (calendarID, eventID, startingDate)

8. UPCOMING\_EVENT (calendarID, eventID, startingDate)

**fd1:** eventID  $\rightarrow$  startingDate

**1NF & 2NF & 3NF & BCNF:**

UPCOMING\_EVENT (calendarID, eventID, startingDate)

## 5.0 Relational DB Schemas

Right after the normalization process here is what he have come up with for relational database schema of this project.

ORGANIZER (organizerID, organizerName, organizerEmail, organizerTelPhone)

PARTICIPANT (participantID, participantName, participantEmail, participantTelPhone)

EVENT (eventID, name, location, email, category, isPaid, calendarID, organizationID, personInCharge)

BOOKING (verificationID, bookingDate, participationID, eventID)

PAYMENT (paymentID, payment\_type, verificationID, participationID, eventID)

CALENDAR (calendarID, startingDate, eventID)

PAST\_EVENT (calendarID, startingDate, eventID)

UPCOMING\_EVENT(calendarID, startingDate, eventID)

## 6.0 SQL Statements (DDL & DML)

```
CREATE TABLE EVENT(  
    eventID VARCHAR2(15),  
    name     VARCHAR2(100) NOT NULL,  
    location VARCHAR2(100) NOT NULL,  
    email    VARCHAR2(100) NOT NULL,  
    category VARCHAR2(100) NOT NULL,  
    isPaid   VARCHAR2(1) NOT NULL,  
    organizationID VARCHAR2(15) NOT NULL,  
    personInCharge VARCHAR2(100) NOT NULL,  
    CONSTRAINT eventID_pk PRIMARY KEY (eventID)  
);  
  
CREATE TABLE PAYMENT(  
    paymentID VARCHAR2(15),  
    payment_type VARCHAR2(15) NOT NULL,  
    verificationID VARCHAR2(15) NOT NULL,  
    participantID VARCHAR2(15) NOT NULL,  
    eventID VARCHAR2(15) NOT NULL,  
    CONSTRAINT paymentID_pk PRIMARY KEY (paymentID)  
);  
  
CREATE TABLE BOOKING(  
    verificationID VARCHAR2(15) NOT NULL,  
    bookingDate DATE NOT NULL,  
    participantID VARCHAR2(15) NOT NULL,  
    eventID VARCHAR2(15) NOT NULL,  
    CONSTRAINT verificationID_pk PRIMARY KEY (verificationID)  
);  
  
CREATE TABLE ORGANIZER(  
    organizerID VARCHAR2(15) NOT NULL,  
    organizerName VARCHAR2(100) NOT NULL,  
    organizerEmail VARCHAR2(100) NOT NULL,  
    organizerTelPhone VARCHAR2(12) NOT NULL,  
    CONSTRAINT organizerID_pk PRIMARY KEY (organizerID)  
);
```

```
CREATE TABLE PARTICIPANT(  
    participantID VARCHAR2(15) NOT NULL,  
    participantName VARCHAR2(100) NOT NULL,  
    participantEmail VARCHAR2(100) NOT NULL,  
    participantTelPhone VARCHAR2(15) NOT NULL,  
    CONSTRAINT participantID_pk PRIMARY KEY (participantID)  
);  
  
CREATE TABLE CALENDAR(  
    calendarID VARCHAR2(15) NOT NULL,  
    startingDate DATE NOT NULL,  
    eventID VARCHAR2(15) NOT NULL,  
    CONSTRAINT calendarID_pk PRIMARY KEY (calendarID)  
);  
  
CREATE TABLE UPCOMING_EVENT(  
    calendarID VARCHAR2(15) NOT NULL,  
    eventID VARCHAR2(15) NOT NULL,  
    startingDate DATE NOT NULL  
);
```

Table EVENT created.

Table PAYMENT created.

Table BOOKING created.

Table ORGANIZER created.

Table PARTICIPANT created.

Table CALENDAR created.

Table UPCOMING\_EVENT created.

Table PAST\_EVENT created.



```
-- Here lies constraint foreign key --

-- Payment

ALTER TABLE PAYMENT
ADD CONSTRAINT verificationID_fk2 FOREIGN KEY (verificationID)
REFERENCES BOOKING;

ALTER TABLE PAYMENT
ADD CONSTRAINT participantID_fk FOREIGN KEY (participantID)
REFERENCES PARTICIPANT;

ALTER TABLE PAYMENT
ADD CONSTRAINT eventID_fk2 FOREIGN KEY (eventID) REFERENCES EVENT;

-- Booking

ALTER TABLE BOOKING
ADD CONSTRAINT participantID_fk2 FOREIGN KEY (participantID)
REFERENCES PARTICIPANT;

ALTER TABLE BOOKING
ADD CONSTRAINT eventID_fk FOREIGN KEY (eventID) REFERENCES EVENT;

-- Event

ALTER TABLE EVENT
ADD CONSTRAINT organizationID_fk FOREIGN KEY (organizationID)
REFERENCES ORGANIZER;

-- Calendar

ALTER TABLE CALENDAR
ADD CONSTRAINT eventID_fk3 FOREIGN KEY (eventID) REFERENCES EVENT;

-- Upcoming Event

ALTER TABLE UPCOMING_EVENT
ADD CONSTRAINT eventID_fk4 FOREIGN KEY (eventID) REFERENCES EVENT;
```

```
-- Past Event
```

```
ALTER TABLE PAST_EVENT  
ADD CONSTRAINT eventID_fk5 FOREIGN KEY (eventID) REFERENCES EVENT;
```

Table PAYMENT altered.

Table PAYMENT altered.

Table PAYMENT altered.

Table BOOKING altered.

---

Script Output

Table BOOKING altered.

Table EVENT altered.

Table CALENDAR altered.

Table UPCOMING\_EVENT altered.

Table PAST\_EVENT altered.

-- Here lies Adding Ton of Data

```
INSERT INTO PARTICIPANT VALUES ('P001', 'John Doe',  
'john.doe@example.com', '123-456-7890');  
INSERT INTO PARTICIPANT VALUES ('P002', 'Jane Smith',  
'jane.smith@example.com', '987-654-3210');  
INSERT INTO PARTICIPANT VALUES ('P003', 'Bob Johnson',  
'bob.johnson@example.com', '111-222-3333');  
INSERT INTO PARTICIPANT VALUES ('P004', 'Alice Brown',  
'alice.brown@example.com', '444-555-6666');  
INSERT INTO PARTICIPANT VALUES ('P005', 'Charlie White',  
'charlie.white@example.com', '777-888-9999');  
INSERT INTO PARTICIPANT VALUES ('P006', 'Eva Davis',  
'eva.davis@example.com', '333-444-5555');  
INSERT INTO PARTICIPANT VALUES ('P007', 'Frank Miller',  
'frank.miller@example.com', '666-777-8888');  
INSERT INTO PARTICIPANT VALUES ('P008', 'Grace Wilson',  
'grace.wilson@example.com', '222-333-4444');  
INSERT INTO PARTICIPANT VALUES ('P009', 'Henry Lee',  
'henry.lee@example.com', '999-000-1111');  
INSERT INTO PARTICIPANT VALUES ('P010', 'Ivy Taylor',  
'ivy.taylor@example.com', '555-666-7777');  
INSERT INTO PARTICIPANT VALUES ('P011', 'Jack Robinson',  
'jack.robinson@example.com', '888-999-0000');  
INSERT INTO PARTICIPANT VALUES ('P012', 'Kelly Turner',  
'kelly.turner@example.com', '444-555-6666');  
INSERT INTO PARTICIPANT VALUES ('P013', 'Leo Harris',  
'leo.harris@example.com', '111-222-3333');  
INSERT INTO PARTICIPANT VALUES ('P014', 'Mia Garcia',  
'mia.garcia@example.com', '777-888-9999');  
INSERT INTO PARTICIPANT VALUES ('P015', 'Nora Evans',  
'nora.evans@example.com', '123-456-7890');  
INSERT INTO PARTICIPANT VALUES ('P016', 'Oscar Wright',  
'oscar.wright@example.com', '987-654-3210');  
INSERT INTO PARTICIPANT VALUES ('P017', 'Pamela Adams',  
'pamela.adams@example.com', '111-222-3333');  
INSERT INTO PARTICIPANT VALUES ('P018', 'Quincy Green',  
'quincy.green@example.com', '555-666-7777');  
INSERT INTO PARTICIPANT VALUES ('P019', 'Rachel Hall',  
'rachel.hall@example.com', '777-888-9999');
```

```
INSERT INTO PARTICIPANT VALUES ('P020', 'Samuel Lewis',
'samuel.lewis@example.com', '222-333-4444');

INSERT INTO ORGANIZER VALUES ('ORG001', 'ABC Events',
'abc.events@example.com', '123-456-7890');
INSERT INTO ORGANIZER VALUES ('ORG002', 'XYZ Productions',
'xyz.productions@example.com', '987-654-3210');
INSERT INTO ORGANIZER VALUES ('ORG003', 'EventCo Enterprises',
'eventco@example.com', '555-123-4567');
INSERT INTO ORGANIZER VALUES ('ORG004', 'Dream Events',
'dream.events@example.com', '111-222-3333');
INSERT INTO ORGANIZER VALUES ('ORG005', 'Elite Planners',
'elite.planners@example.com', '444-555-6666');
INSERT INTO ORGANIZER VALUES ('ORG006', 'City Celebrations',
'city.celebrations@example.com', '777-888-9999');
INSERT INTO ORGANIZER VALUES ('ORG007', 'Gala Creations',
'gala.creations@example.com', '999-888-7777');
INSERT INTO ORGANIZER VALUES ('ORG008', 'Star Productions',
'star.productions@example.com', '333-222-1111');
INSERT INTO ORGANIZER VALUES ('ORG009', 'Epic Events',
'epic.events@example.com', '666-555-4444');
INSERT INTO ORGANIZER VALUES ('ORG010', 'Global Celebrations',
'global.celebrations@example.com', '000-999-8888');
INSERT INTO ORGANIZER VALUES ('ORG011', 'Royal Gatherings',
'royal.gatherings@example.com', '222-333-4444');
INSERT INTO ORGANIZER VALUES ('ORG012', 'Creative Minds Events',
'creative.minds@example.com', '777-666-555');
INSERT INTO ORGANIZER VALUES ('ORG013', 'Grand Affairs',
'grand.affairs@example.com', '444-333-2222');
INSERT INTO ORGANIZER VALUES ('ORG014', 'Majestic Events',
'majestic.events@example.com', '111-000-9999');
INSERT INTO ORGANIZER VALUES ('ORG015', 'Elegance Productions',
'elegance.productions@example.com', '888-777-6666');
INSERT INTO ORGANIZER VALUES ('ORG016', 'Premier Planners',
'premier.planners@example.com', '555-444-3333');
INSERT INTO ORGANIZER VALUES ('ORG017', 'Harmony Events',
'harmony.events@example.com', '222-111-0000');
INSERT INTO ORGANIZER VALUES ('ORG018', 'Prestige Productions',
'prestige.productions@example.com', '999-888-7777');
```

```
INSERT INTO ORGANIZER VALUES ('ORG019', 'Chic Celebrations',
'chic.celebrations@example.com', '666-555-4444');
INSERT INTO ORGANIZER VALUES ('ORG020', 'Signature Events',
'signature.events@example.com', '333-222-1111');

INSERT INTO CALENDAR VALUES ('CAL001', TO_DATE('2024-01-15',
'YYYY-MM-DD'), 'EVT001');
INSERT INTO CALENDAR VALUES ('CAL002', TO_DATE('2024-02-20',
'YYYY-MM-DD'), 'EVT002');
INSERT INTO CALENDAR VALUES ('CAL003', TO_DATE('2024-03-10',
'YYYY-MM-DD'), 'EVT003');
INSERT INTO CALENDAR VALUES ('CAL004', TO_DATE('2024-04-05',
'YYYY-MM-DD'), 'EVT004');
INSERT INTO CALENDAR VALUES ('CAL005', TO_DATE('2024-05-15',
'YYYY-MM-DD'), 'EVT005');
INSERT INTO CALENDAR VALUES ('CAL006', TO_DATE('2024-06-08',
'YYYY-MM-DD'), 'EVT006');
INSERT INTO CALENDAR VALUES ('CAL007', TO_DATE('2024-07-20',
'YYYY-MM-DD'), 'EVT007');
INSERT INTO CALENDAR VALUES ('CAL008', TO_DATE('2024-08-12',
'YYYY-MM-DD'), 'EVT008');
INSERT INTO CALENDAR VALUES ('CAL009', TO_DATE('2024-09-25',
'YYYY-MM-DD'), 'EVT009');
INSERT INTO CALENDAR VALUES ('CAL010', TO_DATE('2024-10-18',
'YYYY-MM-DD'), 'EVT010');
INSERT INTO CALENDAR VALUES ('CAL011', TO_DATE('2024-11-08',
'YYYY-MM-DD'), 'EVT011');
INSERT INTO CALENDAR VALUES ('CAL012', TO_DATE('2024-12-01',
'YYYY-MM-DD'), 'EVT012');
INSERT INTO CALENDAR VALUES ('CAL013', TO_DATE('2024-12-20',
'YYYY-MM-DD'), 'EVT013');
INSERT INTO CALENDAR VALUES ('CAL014', TO_DATE('2025-01-15',
'YYYY-MM-DD'), 'EVT014');
INSERT INTO CALENDAR VALUES ('CAL015', TO_DATE('2025-02-10',
'YYYY-MM-DD'), 'EVT015');
INSERT INTO CALENDAR VALUES ('CAL016', TO_DATE('2025-03-05',
'YYYY-MM-DD'), 'EVT016');
INSERT INTO CALENDAR VALUES ('CAL017', TO_DATE('2025-04-18',
'YYYY-MM-DD'), 'EVT017');
```

```
INSERT INTO CALENDAR VALUES ('CAL018', TO_DATE('2025-05-20',
'YYYY-MM-DD'), 'EVT018');
INSERT INTO CALENDAR VALUES ('CAL019', TO_DATE('2025-06-08',
'YYYY-MM-DD'), 'EVT019');
INSERT INTO CALENDAR VALUES ('CAL020', TO_DATE('2025-07-15',
'YYYY-MM-DD'), 'EVT020');

INSERT INTO EVENT VALUES ('EVT001', 'Summer Gala', 'Grand Ballroom',
'summer.gala@example.com', 'Social', 'Y', 'ORG001', 'Alex
Anderson');
INSERT INTO EVENT VALUES ('EVT002', 'Tech Conference 2024',
'Convention Center', 'tech.conference@example.com', 'Technology',
'N', 'ORG002', 'Bella Clark');
INSERT INTO EVENT VALUES ('EVT003', 'Art Exhibition', 'City Art
Gallery', 'art.exhibition@example.com', 'Arts', 'Y', 'ORG003',
'Carter Davis');
INSERT INTO EVENT VALUES ('EVT004', 'Corporate Retreat', 'Luxury
Resort', 'corporate.retreat@example.com', 'Business', 'N', 'ORG004',
'Diana Edwards');
INSERT INTO EVENT VALUES ('EVT005', 'Music Festival', 'Outdoor
Stadium', 'music.festival@example.com', 'Entertainment', 'Y',
'ORG005', 'Ethan Foster');
INSERT INTO EVENT VALUES ('EVT006', 'Health and Wellness Expo',
'Exhibition Hall', 'wellness.expo@example.com', 'Health', 'N',
'ORG006', 'Fiona Garcia');
INSERT INTO EVENT VALUES ('EVT007', 'Fashion Show', 'Fashion
Center', 'fashion.show@example.com', 'Fashion', 'Y', 'ORG007',
'George Harris');
INSERT INTO EVENT VALUES ('EVT008', 'Community Fair', 'Local Park',
'community.fair@example.com', 'Community', 'N', 'ORG008', 'Hannah
Ingram');
INSERT INTO EVENT VALUES ('EVT009', 'Sports Championship', 'Sports
Arena', 'sports.championship@example.com', 'Sports', 'Y', 'ORG009',
'Isaac Johnson');
INSERT INTO EVENT VALUES ('EVT010', 'Wedding Celebration', 'Luxury
Hotel', 'wedding.celebration@example.com', 'Wedding', 'N', 'ORG010',
'Jasmine Kim');
INSERT INTO EVENT VALUES ('EVT011', 'Food and Wine Tasting',
'Vineyard', 'food.wine@example.com', 'Culinary', 'Y', 'ORG011',
'Kevin Lopez');
```

```
INSERT INTO EVENT VALUES ('EVT012', 'Technology Expo', 'Exhibition Center', 'tech.expo@example.com', 'Technology', 'N', 'ORG012', 'Lily Morgan');
INSERT INTO EVENT VALUES ('EVT013', 'Film Festival', 'Movie Theater', 'film.festival@example.com', 'Entertainment', 'Y', 'ORG013', 'Max Nelson');
INSERT INTO EVENT VALUES ('EVT014', 'Educational Summit', 'University Auditorium', 'education.summit@example.com', 'Education', 'N', 'ORG014', 'Nina Oliver');
INSERT INTO EVENT VALUES ('EVT015', 'Charity Auction', 'Event Hall', 'charity.auction@example.com', 'Charity', 'Y', 'ORG015', 'Owen Parker');
INSERT INTO EVENT VALUES ('EVT016', 'Holiday Celebration', 'Community Center', 'holiday.celebration@example.com', 'Holiday', 'N', 'ORG016', 'Penny Quinn');
INSERT INTO EVENT VALUES ('EVT017', 'Science Fair', 'Science Museum', 'science.fair@example.com', 'Science', 'Y', 'ORG017', 'Quinn Rivera');
INSERT INTO EVENT VALUES ('EVT018', 'Gaming Convention', 'Convention Center', 'gaming.convention@example.com', 'Gaming', 'N', 'ORG018', 'Ryan Silva');
INSERT INTO EVENT VALUES ('EVT019', 'Auto Show', 'Convention Center', 'auto.show@example.com', 'Automotive', 'Y', 'ORG019', 'Samantha Turner');
INSERT INTO EVENT VALUES ('EVT020', 'Dance Performance', 'Dance Theater', 'dance.performance@example.com', 'Arts', 'N', 'ORG020', 'Tyler Walker');
```

```
-- Free Event
-- Lets say user with the id 'P001' until 'P009' registered for
event with id 'EVT002'
-- Their verification id will be automatically generated
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P001', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P002', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P003', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P004', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P005', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P006', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P007', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P008', 'EVT002');
INSERT INTO BOOKING VALUES (DBMS_RANDOM.STRING('L', 15),
SYSDATE, 'P009', 'EVT002');
```



	❖ VERIFICATIONID	❖ BOOKINGDATE	❖ PARTICIPANTID	❖ EVENTID	
1	mfuqurchwmjomtl	15/01/2024	P001	EVT002	
2	mzxbqqdvpcviznu	15/01/2024	P002	EVT002	
3	dojfmrldkersbhox	15/01/2024	P003	EVT002	
4	kohbiodnrgpacqi	15/01/2024	P004	EVT002	
5	mgvgsvgpeqesyjf	15/01/2024	P005	EVT002	
6	nnkxddxdbmkdkid	15/01/2024	P006	EVT002	
7	apqaxlxoebbszar	15/01/2024	P007	EVT002	
8	urbcdsezujzqzqa	15/01/2024	P008	EVT002	
9	ygfvqbdwxjppdx	15/01/2024	P009	EVT002	

```

-- Paid Event
-- Lets say user with the id 'P001' registered for event with id 'EVT001'
-- Their verification id will be automatically generated
INSERT INTO BOOKING VALUES ('abecipqggkutwtu',SYSDATE,'P001','EVT001'); --
generated verification id abecipqggkutwtu

-- Pay Using Online Banking
INSERT INTO PAYMENT VALUES ('PYT001', 'Online Banking', 'abecipqggkutwtu',
'P001', 'EVT001');

```

10	ygvfqbdbwxjppdxd	15/01/2024	P009	EVT002
----	------------------	------------	------	--------

	⚡ PAYMENTID	⚡ PAYMENT_TYPE	⚡ VERIFICATIONID	⚡ PARTICIPANTID	⚡ EVENTID
1	PYT001	Online Banking	abecipqggkutwtu	P001	EVT001

```

-- Here lies user P001 refund money for event EVT001
DELETE FROM PAYMENT WHERE participantID='P001' AND eventID='EVT001' AND
verificationID='abecipqggkutwtu';
DELETE FROM BOOKING WHERE participantID='P001' AND
verificationID='abecipqggkutwtu';

```

1 row deleted.

1 row deleted.

```

-- Here lies organizer create a new event
-- Add one more calendarID
INSERT INTO EVENT VALUES ('EVT021', 'Defcon', 'Computer Science Lab',
'deflcon.performance@example.com', 'Cyber Security', 'N', 'ORG020',
'Megat');
INSERT INTO CALENDAR VALUES ('CAL021', TO_DATE('2025-07-18',
'YYYY-MM-DD'), 'EVT021');

```

19	EVT019	Auto Show	Convention Center	auto.show@example.com	Automotive	Y	ORG019	Samantha Turner
20	EVT020	Dance Performance	Dance Theater	dance.performance@example.com	Arts	N	ORG020	Tyler Walker
21	EVT021	Defcon	Computer Science Lab	deflcon.performance@example.com	Cyber Security	N	ORG020	Megat

20	CAL020	15/07/2025	EVT020
21	CAL021	18/07/2025	EVT021

```
-- Here lies finding out the trending event analyzation
```

```
SELECT
```

```
    E.category,
```

```
    COUNT(B.eventID) AS booking_count
```

```
FROM
```

```
    EVENT E
```

```
LEFT JOIN
```

```
    BOOKING B ON E.eventID = B.eventID
```

```
GROUP BY
```

```
    E.category;
```

CATEGORY	BOOKING_COUNT
Technology	9
Gaming	0
Fashion	0
Business	0
Education	0

```
-- Here lies finding out the top 10 active students
```

```
SELECT
    P.participantName,
    COUNT(B.participantID) AS booking_count
FROM
    PARTICIPANT P
JOIN
    BOOKING B ON P.participantID = B.participantID
GROUP BY
    P.participantName
ORDER BY
    booking_count DESC
FETCH FIRST 10 ROWS ONLY;
```

PARTICIPANTNAME	BOOKING_COUNT
John Doe	1
Jane Smith	1
Bob Johnson	1
Alice Brown	1
Henry Lee	1
Eva Davis	1
Frank Miller	1
Grace Wilson	1
Charlie White	1

```
-- Here lies finding out the organizer of event
```

```
SELECT
```

```
    O.organizerName,  
    E.eventID,  
    E.name AS eventName,  
    E.location,  
    E.category,  
    E.isPaid
```

```
FROM
```

```
    ORGANIZER O
```

```
JOIN
```

```
    EVENT E ON O.organizerID = E.organizationID
```

```
ORDER BY
```

```
    O.organizerName, E.eventID;
```

ORGANIZERNAME	EVENTID	EVENTNAME
ABC Events	EVT001	Summer Gala
Chic Celebrations	EVT019	Auto Show
City Celebrations	EVT006	Health and Wellness Expo
Creative Minds Events	EVT012	Technology Expo
Dream Events	EVT004	Corporate Retreat
Elegance Productions	EVT015	Charity Auction
Elite Planners	EVT005	Music Festival
Epic Events	EVT009	Sports Championship
EventCo Enterprises	EVT003	Art Exhibition
Gala Creations	EVT007	Fashion Show
Global Celebrations	EVT010	Wedding Celebration
ORGANIZERNAME	EVENTID	EVENTNAME
Grand Affairs	EVT013	Film Festival
Harmony Events	EVT017	Science Fair
Majestic Events	EVT014	Educational Summit
Premier Planners	EVT016	Holiday Celebration
Prestige Productions	EVT018	Gaming Convention
Royal Gatherings	EVT011	Food and Wine Tasting
Signature Events	EVT020	Dance Performance
Signature Events	EVT021	Deftoon
Star Productions	EVT008	Community Fair
XYZ Productions	EVT002	Tech Conference 2024

```

INSERT INTO BOOKING VALUES ('abecipqggkutwtz', SYSDATE, 'P002', 'EVT001');
-- generated verification id abecipqggkutwtz
INSERT INTO PAYMENT VALUES ('PYT002', 'Online Banking', 'abecipqggkutwtz',
'P002', 'EVT001');

INSERT INTO BOOKING VALUES ('abecipqggkutwtp', SYSDATE, 'P003', 'EVT001');
-- generated verification id abecipqggkutwtp
INSERT INTO PAYMENT VALUES ('PYT003', 'Online Banking', 'abecipqggkutwtp',
'P003', 'EVT001');

INSERT INTO BOOKING VALUES ('abecipqggkutwto', SYSDATE, 'P004', 'EVT001');
-- generated verification id abecipqggkutwto
INSERT INTO PAYMENT VALUES ('PYT004', 'Online Banking', 'abecipqggkutwto',
'P004', 'EVT001');

SELECT
    E.eventID,
    E.name AS eventName,
    E.category,
    COUNT(B.participantID) AS participantsCount
FROM
    EVENT E
JOIN
    BOOKING B ON E.eventID = B.eventID
WHERE
    E.isPaid = 'Y'
GROUP BY
    E.eventID, E.name, E.category
ORDER BY
    participantsCount DESC
FETCH FIRST 1 ROWS ONLY;

```

	EV...		EVENTNAME	CATEGORY	PARTICIPANTSCOUNT
1	EVT001		Summer Gala	Social	3

```
-- Populate UPCOMING_EVENT and PAST_EVENT tables based on the startingDate

INSERT INTO UPCOMING_EVENT (calendarID, eventID, startingDate)
SELECT c.calendarID, e.eventID, c.startingDate
FROM CALENDAR c
JOIN EVENT e ON c.eventID = e.eventID
WHERE c.startingDate > TO_DATE('2024-01-15', 'YYYY-MM-DD');

INSERT INTO PAST_EVENT (calendarID, eventID, startingDate)
SELECT c.calendarID, e.eventID, c.startingDate
FROM CALENDAR c
JOIN EVENT e ON c.eventID = e.eventID
WHERE c.startingDate <= TO_DATE('2024-01-15', 'YYYY-MM-DD');
```

### Upcoming Event

	CALENDARID	EVENTID	STARTINGDATE
1	CAL002	EVT002	20/02/2024
2	CAL003	EVT003	10/03/2024
3	CAL004	EVT004	05/04/2024
4	CAL005	EVT005	15/05/2024
5	CAL006	EVT006	08/06/2024
6	CAL007	EVT007	20/07/2024
7	CAL008	EVT008	12/08/2024
8	CAL009	EVT009	25/09/2024
9	CAL010	EVT010	18/10/2024
10	CAL011	EVT011	08/11/2024
11	CAL012	EVT012	01/12/2024
12	CAL013	EVT013	20/12/2024
13	CAL014	EVT014	15/01/2025
14	CAL015	EVT015	10/02/2025
15	CAL016	EVT016	05/03/2025
16	CAL017	EVT017	18/04/2025
17	CAL018	EVT018	20/05/2025
18	CAL019	EVT019	08/06/2025
19	CAL020	EVT020	15/07/2025



Past Event

	CALENDARID	EVENTID	STARTINGDATE
1	CAL001	EVT001	15/01/2024

## **Conclusion**

To put everything to bed, what we have been doing throughout this whole project is upgrading the basic functions in Nexscholar. We recognise a few flaws and lack of features that already exist within the system and come up with a couple of brilliant ideas to enhance and better enable the system to wider user

Such improvements to the system are not easy to bring about as there are often many parties that are involved in the system process such as administrator, participant, and event manager. Henceforth, the new or upgraded functionalities that were added to the system have to take into account all of the parties involved and how they interconnected with each other to provide a seamless user experience.

As a whole, we learn a lot about what it takes to develop a fully functional system and make it good enough to be accepted and used by the public. We learned about focusing on the details and how they matter in creating a system of this scale. That is why, our system presses on a lot of additional functionalities from the previous system, as we think a lot about the ‘what-if’, not only from the user perspective, but also from other entities as well.

The SQL implementation we did was mostly done based on the syllabus in the class, as it mostly required basic syntax to apply to our system. This syntax includes adding, joining applying constraints, etc. Hence, we find the cohesive similiarity between what we learn and the application of that knowledge in real-life scenarios.

Overall, this project was a success from our team's point of view, as the coordination from each team member was above and beyond the call of duty, and we believe that working on such a project further enhanced our understanding of the application and the use of database in our day to day life.