EVENT MANAGEMENT SYSTEM

18CSC209J - Database Management System and Cloud Integration Services

Mini Project Report

Submitted by

MADUNURI HARSHINI[RA2111028010112]
B.Tech. CSE – CLOUD COMPUTING

YELURI BADRINATH[RA2111028010111]
B.Tech. CSE – CLOUD COMPUTING

DEPARTMENT OF NETWORKING AND COMMUNICATION
SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT

MAY 2023

BONAFIDE

This is to certify that **18CSC209J – DATABASE MANAGEMENT SYSTEM AND CLOUD INTEGRATION SERVICES LABORATORY Mini Project report** titled " Event Management System"is the bonafide work of **MADUNURI HARSHINI (RA2111028010112),YELUR1 BADRINATH (RA2111028010111)** who undertook the task of completing the project within the allotted time.

SIGNATURE

Dr. B. BALAKIRUTHIGA

Assistant Professor
Department of Networking and Communication
SRM Institute of Science and Technology

SIGNATURE

Dr. Annapurani Panaiyappan K

Professor and Head
Department of Networking and Communication
SRM Institute of Science and Technology

TABLE OF CONTENTS

ABSTRACT:

The purpose of Event Management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Event Management System, as described above, can lead to error free, secure, reliable and fast management systems. It can assist the user to coelenterate on their other activities rather than concentrating on the record keeping. Thus it will help organizations in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

INTRODUCTION:

The "Event Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Event Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Activity, Event, Attendees, Payment, Conductors. Every Event Management System has different Event needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

PROBLEM STATEMENT:

Organizing events can be a challenging and time-consuming task, especially when managing a large number of attendees, speakers, sponsors, and vendors. Event planners often struggle with coordinating multiple aspects of an event such as scheduling, ticketing, logistics, and communication.

Therefore, there is a need for an efficient and comprehensive event management system that can streamline the entire process of organizing events. Such a system should be able to handle all aspects of an event, from initial planning to post-event analysis. It should be user-friendly, customizable, and able to integrate with other tools and software.

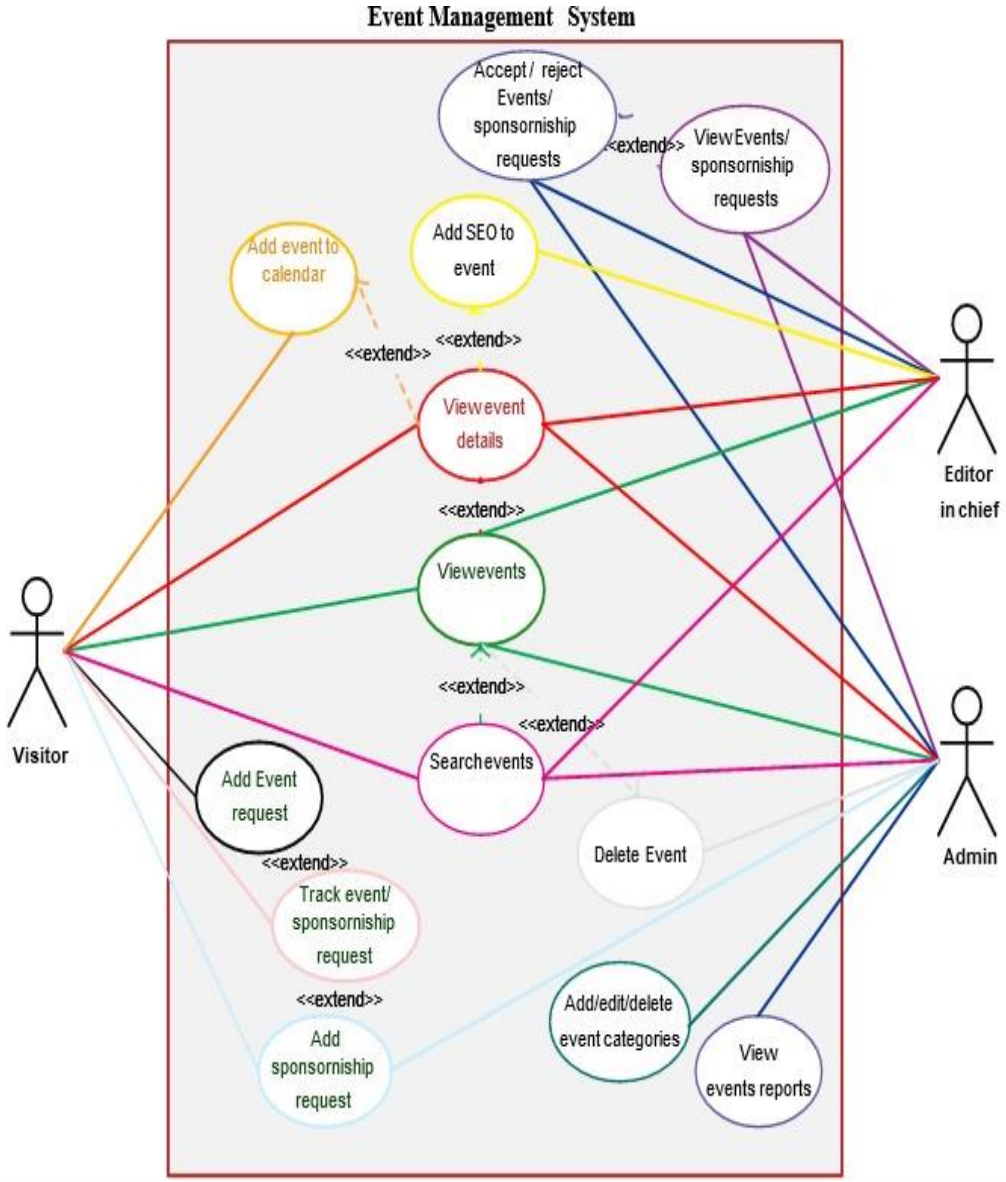
The system should also provide real-time data on attendee engagement, event performance, and revenue generation, enabling event planners to make data-driven decisions and continuously improve their events.

Overall, an effective event management system should simplify the event planning process, save time and resources, and enhance the attendee experience, making it a valuable investment for any organization that hosts events.

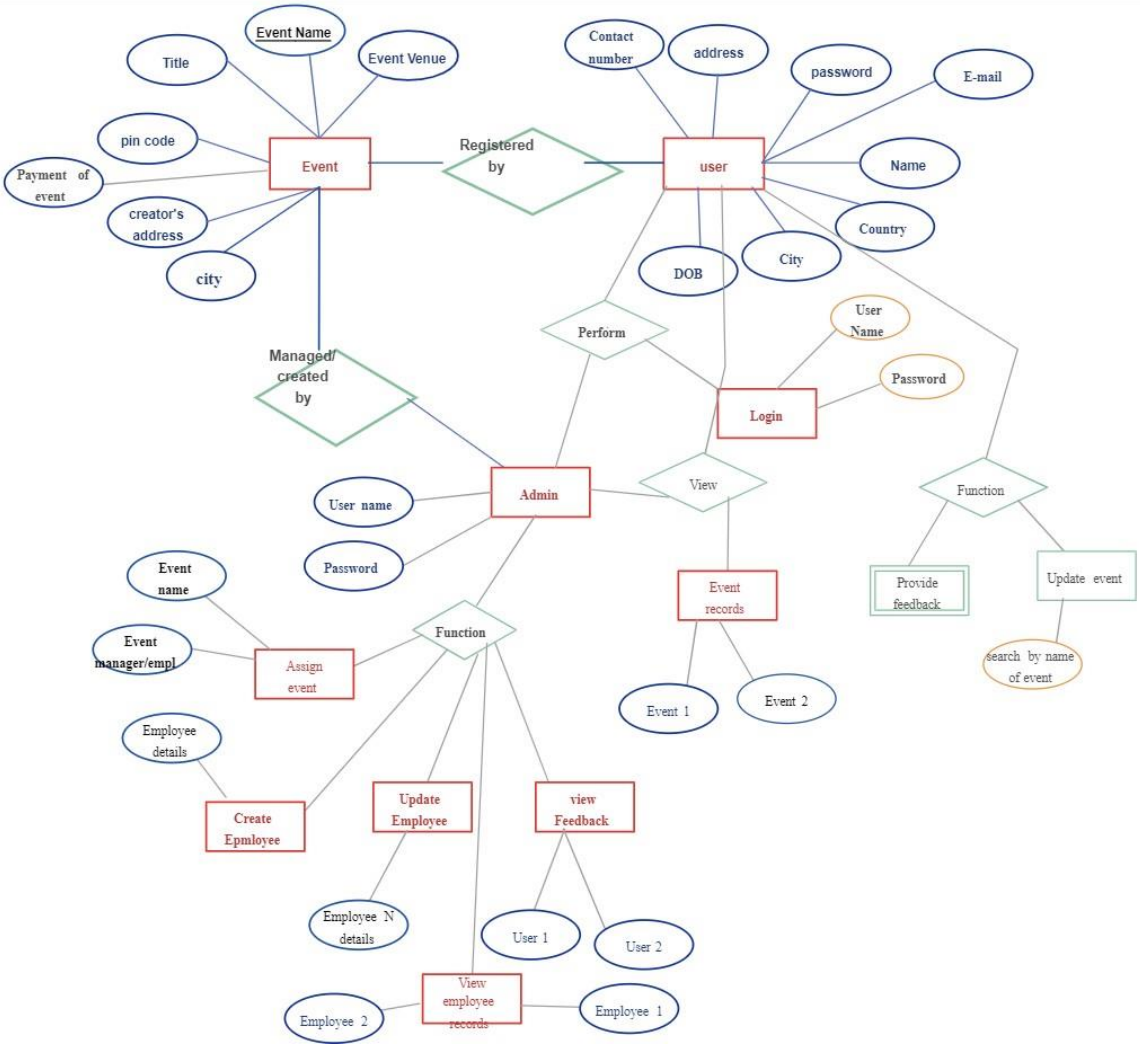
MODULE DESCRIPTION:

- Event Management Module: Used for managing the Event details.
- Conductors Module: Used for managing the details of Conductors
- Payment Module: Used for managing the details of Payment
- Activity Management Module: Used for managing the information and details of the Activity.
- Organizers Module: Used for managing the Organizers details
- Attendees Module: Used for managing the Attendees information
- Login Module: Used for managing the login details
- Users Module: Used for managing the users of the system

USE CASE DIAGRAM:



ER DIAGRAM:


Event Name
Title
Event Venue
pin code
Payment of event
creator's address
city
Event
Registered by
Contact number
address
password
E-mail
user
Name
Country
City
DOB
Perform
User Name
Password
Login
Managed/ created by
Admin
User name
Password
View
Function
Event name
Event manager/empl
Assign event
Function
Event records
Event 1
Event 2
Provide feedback
Update event
search by name of event
Employee details
Create Epmloyee
Update Employee
view Feedback
Employee N details
User 1
User 2
View employee records
Employee 2
Employee 1

DATABASE CREATION USING DDL AND DML:

DDL (Data Definition Language) and DML (Data Manipulation Language) are two types of SQL commands used to manage and manipulate database objects. Here are some examples of DDL and DML commands that can be used in an event management system:

DDL Commands:

1. CREATE TABLE: This command can be used to create a table to store information about events such as event name, date, time, location, attendees, etc.

Example:

```
CREATE TABLE events (
  event_id INT PRIMARY KEY,
  event_name VARCHAR(50) NOT NULL,
  event_date DATE,
  venue_id INT,
  organizer_id INT
);
```

2. ALTER TABLE: This command is used to modify an existing table. In an event management system, you may use this command to add or remove columns from a table.

Example:

```
ALTER TABLE events ADD event_description TEXT;
```

3. DROP TABLE: This command is used to delete a table from the database.

Example:

```
DROP TABLE events;
```

DML Commands:

1. INSERT INTO: This command is used to add new data to a table.

   Example:

   INSERT INTO events (event_id, event_name, event_date, venue_id, organizer_id)
   VALUES (1, 'Tech Conference', '2023-05-20', 1, 1);

2. UPDATE: This command is used to modify existing data in a table.

   Example:

   UPDATE events SET event_name = 'Digital Marketing Conference'
   WHERE event_id = 1;

3. DELETE: This command is used to delete data from a table.

   Example:

   DELETE FROM events WHERE event_id = 1;

   These are just a few examples of DDL and DML commands that can be used in an event management system. The specific commands you use will depend on the requirements of your system.

NORMALIZATION OF DATABASE:

Normalization is the process of organizing a database to minimize redundancy and improve data integrity. In an event management system, there are several tables that could benefit from normalization, such as events, attendees, venues, and organizers. Here's an example of how you could normalize the events table:

Original events table:

Event_id	Event_name	Event_date	Venue_name	Venue_address	Organizer_name
1	concert	2023-06-01	Madison square	4 pennsylvania piaza	Live nation
2	conference	2023-06-15	hilton	1335 avenue of the americas	Acme Inc

The above table violates the first normal form (1NF) because it has repeating groups of data in the columns for venue_name, venue_address, and organizer_name.

To normalize the events table, you could split it into multiple tables:

Table 1: Events

Event_id	Event_name	Event_data	Venue_id	Organizer_id
1	concert	2023-06-01	1	1
2	conference	2023-06-15	2	2

Table 2: Venues

Venue_id	Venue_name	Venue_address
1	Madison square	4 pennsyvania piaza
2	hilton	1335 avenue of the americans

Table 3: Organizers

Organizer_id	Organizer_name
1	Live nation
2	Acme Inc

In this normalized design, the Events table only stores the foreign keys for the Venue and Organizer tables. This eliminates the redundancy in the original table, and ensures that any updates to the Venue or Organizer data are reflected across all events associated with them.

This design also allows for easier querying and reporting on specific subsets of data, since the data is now split across multiple tables.

Implementation using Dynamo DB:

Experiment -14 (Scan Operations in DynamoDB)

DynamoDB:
Creating DynamoDb table using AWS CLI:
1. Open AWS CloudShell and select the region where you want to create the table. You can choose the region from the dropdown menu in the top-right corner of the CloudShell window.

2. Launch the AWS CLI terminal in CloudShell by clicking on the "Open Terminal" button in the top-right corner of the CloudShell window.

3. Create a JSON file with the table schema. For example, create a file named table_schema.json with the following contents:

JSON - Using a JSON document database, you can store each user's profile efficiently by storing only the attributes that are specific to each user.
EX: echo '{
"TableName": "my_table",
"KeySchema": [
{
"AttributeName": "my_partition_key",
"KeyType": "HASH"
},
{

"AttributeName": "my_sort_key",
"KeyType": "RANGE"
}
],
"AttributeDefinitions": [
{

"AttributeName": "my_partition_key",
"AttributeType": "S"
},
{
"AttributeName": "my_sort_key",
"AttributeType": "N"
}
],
"BillingMode": "PAY_PER_REQUEST"
}' > table_schema.json

Create the table by running the following AWS CLI command:
aws dynamodb create-table --cli-input-json file://table_schema.json
To check the status of the table by running the following AWS CLI command:
aws dynamodb describe-table --table-name my_table
To add data:
{
"my_partition_key": {"S": "my_partition_key_value"},
"my_sort_key": {"N": "1"},
"attribute1": {"S": "value1"},
"attribute2": {"N": "2"}
}
Use the ‘aws dynamodb put-item’ command to add the item to the table. Run the following command:
aws dynamodb put-item --table-name my_table --item file://item.json
Verify that the item has been added by using the aws dynamodb get-item command. Run
the following command:
aws dynamodb get-item --table-name my_table --key '{"my_partition_key": {"S":
"my_partition_key_value"}, "my_sort_key": {"N": "1"}}'
You can add more items by repeating the steps above with different data in the ‘ item.json’
file.

Scan Operations:
Scanning a table for all items:

This command scans the my_table table and returns all items in the table
aws dynamodb scan --table-name my_table

If the table has many items, you may need to paginate the results to retrieve all items. To do
this, add the --page-size parameter to the aws dynamodb scan command and specify the
maximum number of items to retrieve per page. For example, to retrieve 100 items per
page, run the following command:

aws dynamodb scan --table-name my_table --page-size 100
Scanning a table with filter expression:
First run this command to scan with a filter expression.

aws dynamodb scan --table-name my_table --filter-expression "attribute1 = :value" --
expression-attribute-values '{":value":{"S":"value1"}}'

Next verify the results by this command

aws dynamodb scan --table-name my_table --filter-expression "attribute1 = :value" --
expression-attribute-values '{":value":{"S":"value1"}}' | jq '.Items[].{attribute1:.attribute1.S,
attribute2:.attribute2.N}'
Scanning a table with a projection expression:

Run the aws dynamodb scan command with the --table-name parameter to specify the
name of the table you want to scan, and the --projection-expression parameter to specify
the attributes you want to include in the scan. For example, to scan a table named my_table
and include only the attribute1 and attribute2 attributes in the results, run the following
command:

aws dynamodb scan --table-name my_table --projection-expression "attribute1, attribute2" -

-max-items 100

Scanning a table with a limit:

aws dynamodb scan --table-name my_table --limit 10

Scanning table with a parallel scan:

aws dynamodb scan --table-name my_table --total-segments 10

and use this for scanning the first segment:

aws dynamodb scan --table-name my_table --total-segments 10 --segment 0.

Table 1: event

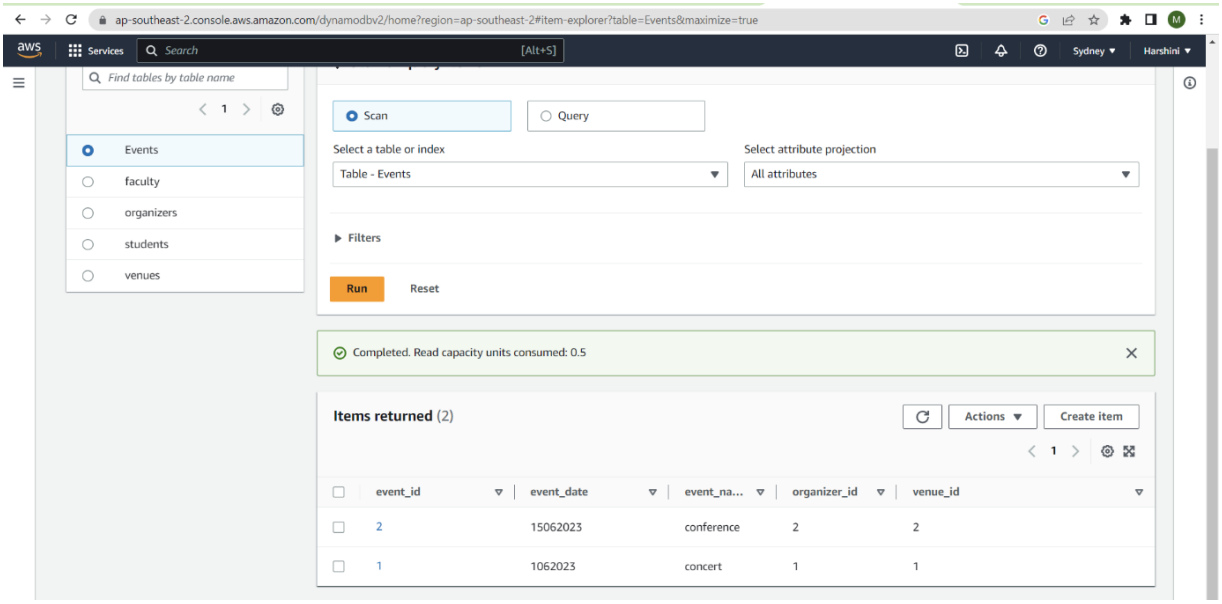


Table 2: venues

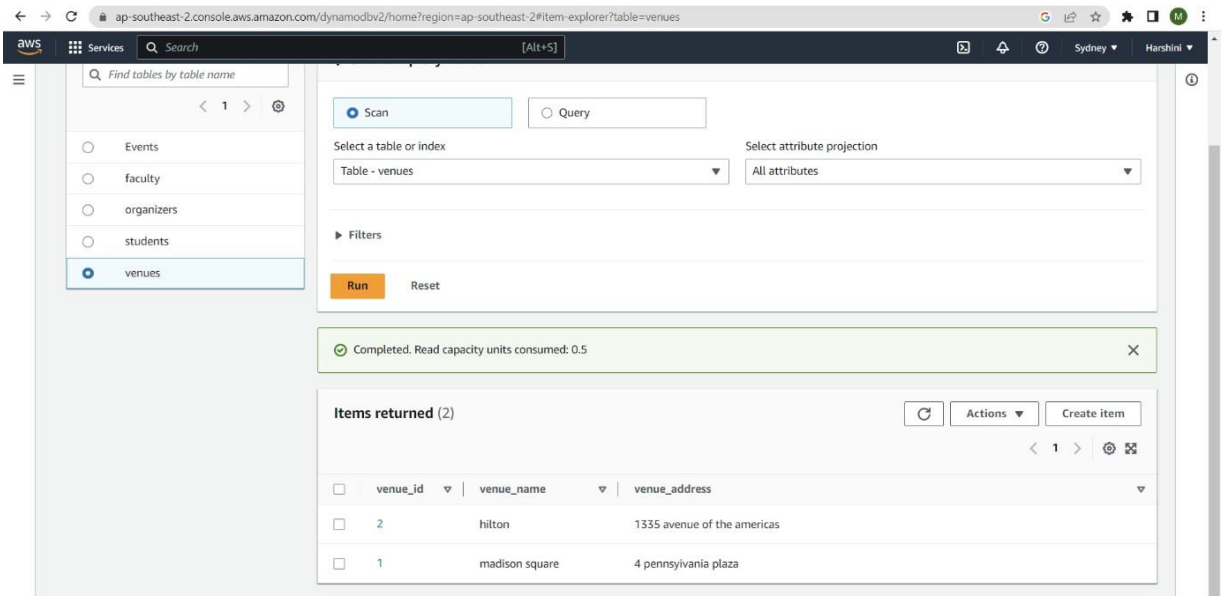
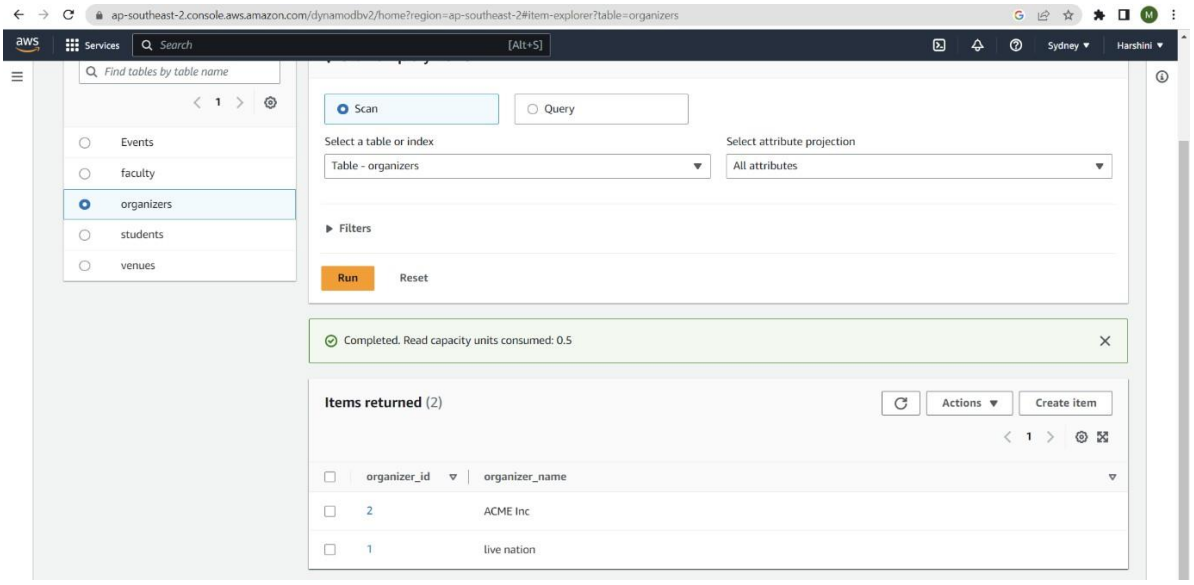


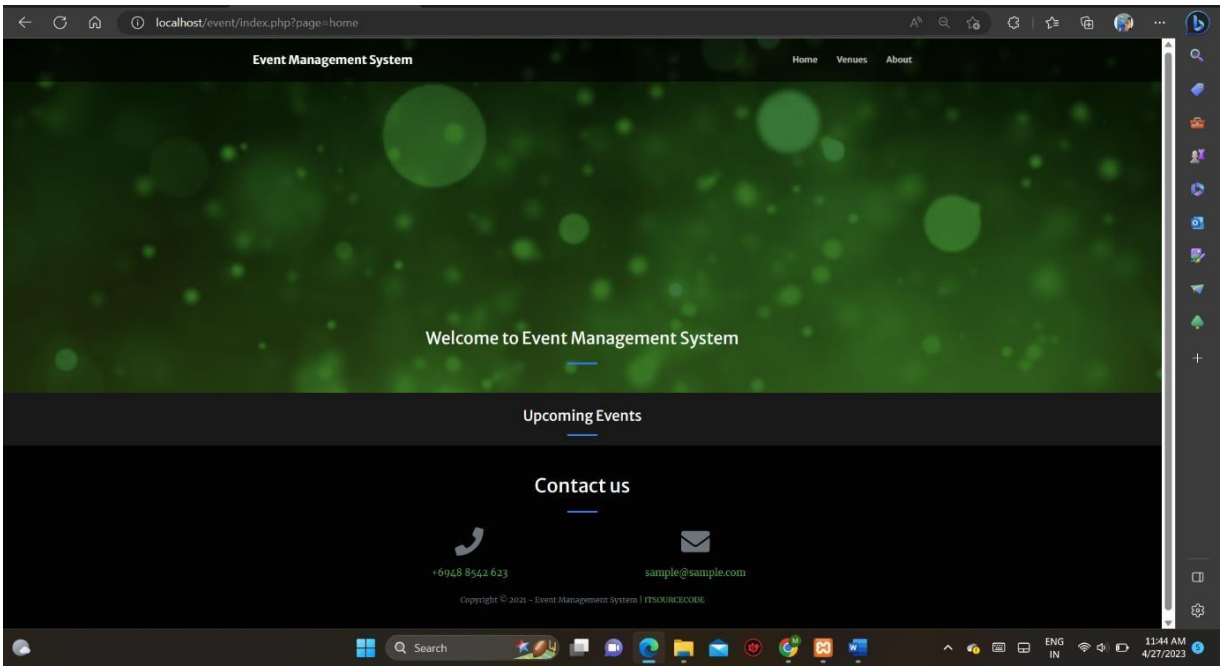
Table 3: organizer

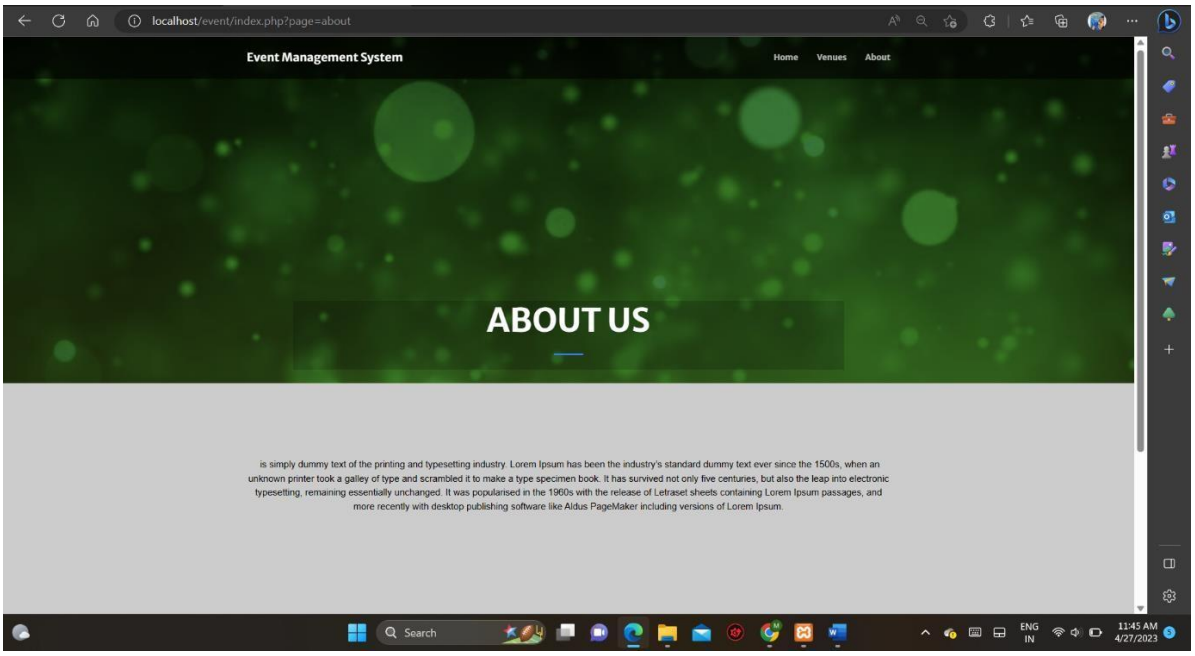


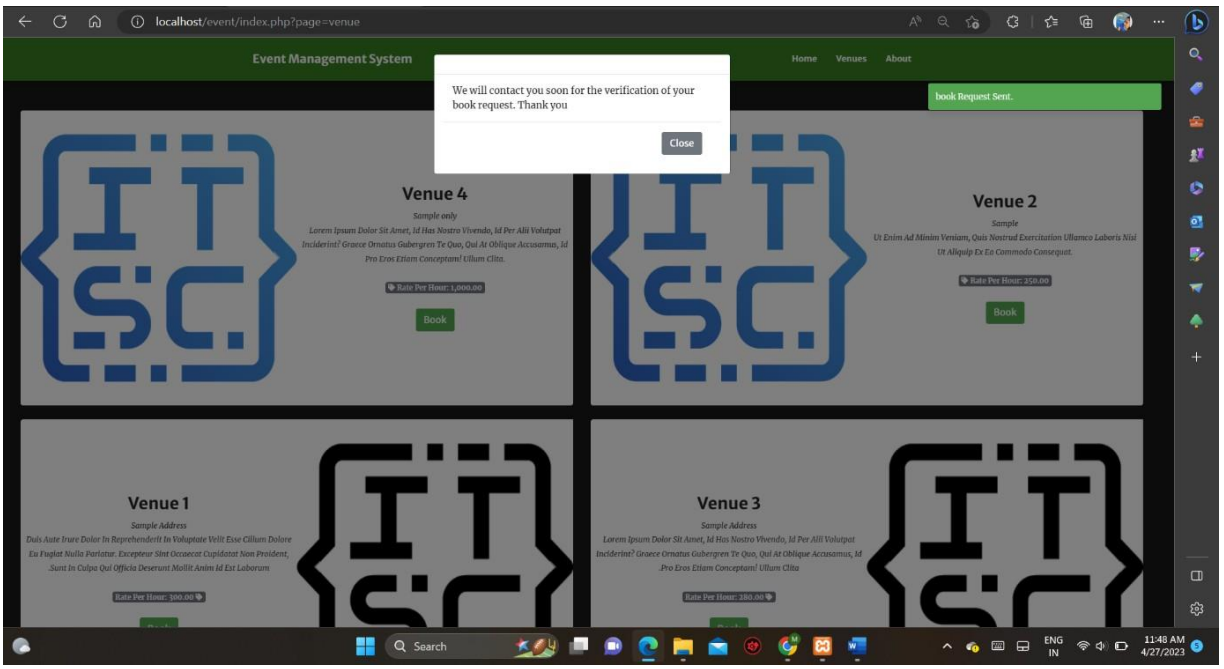
CONCLUSION:

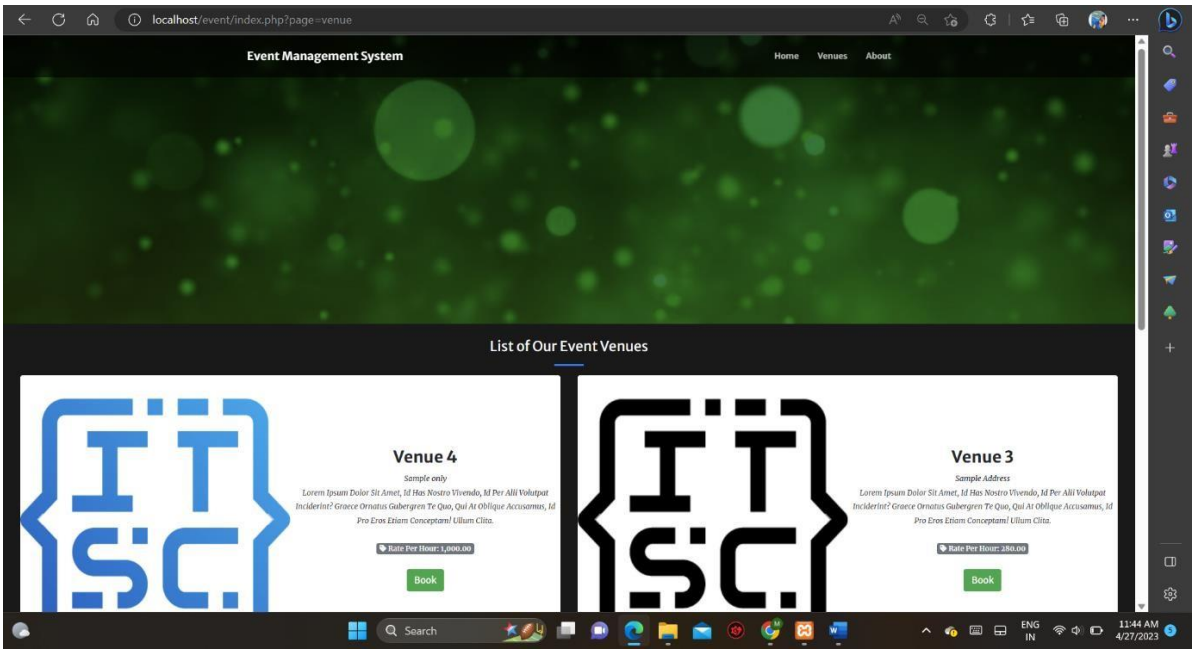
Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

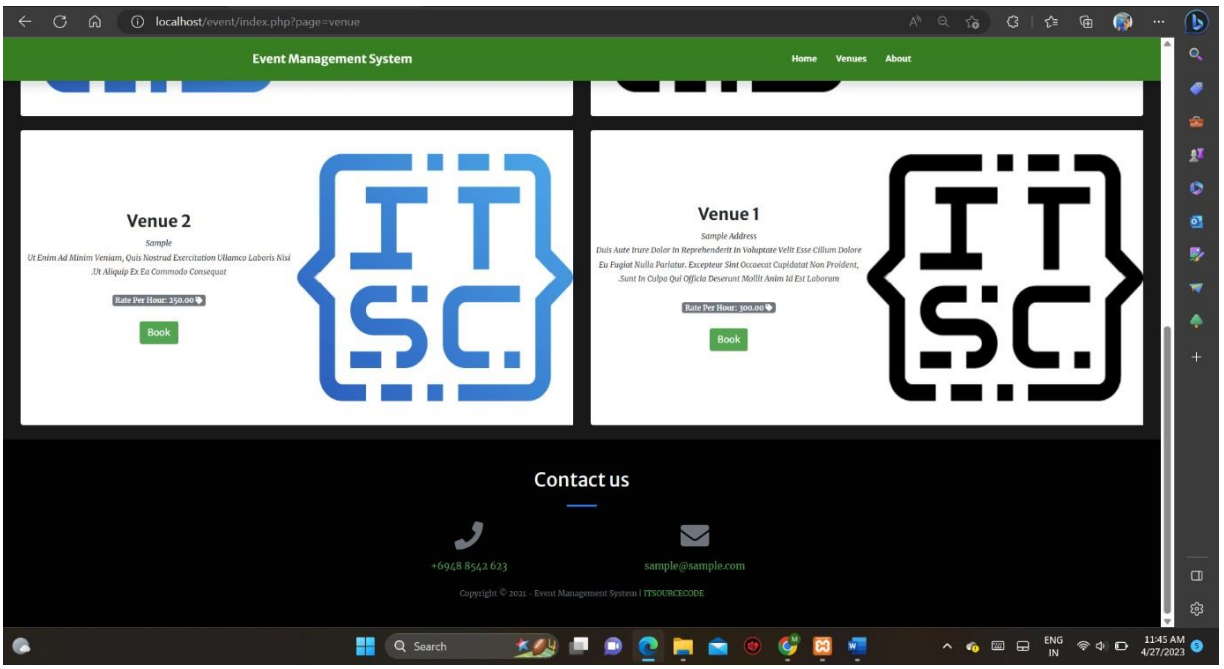
SCREENSHOT:

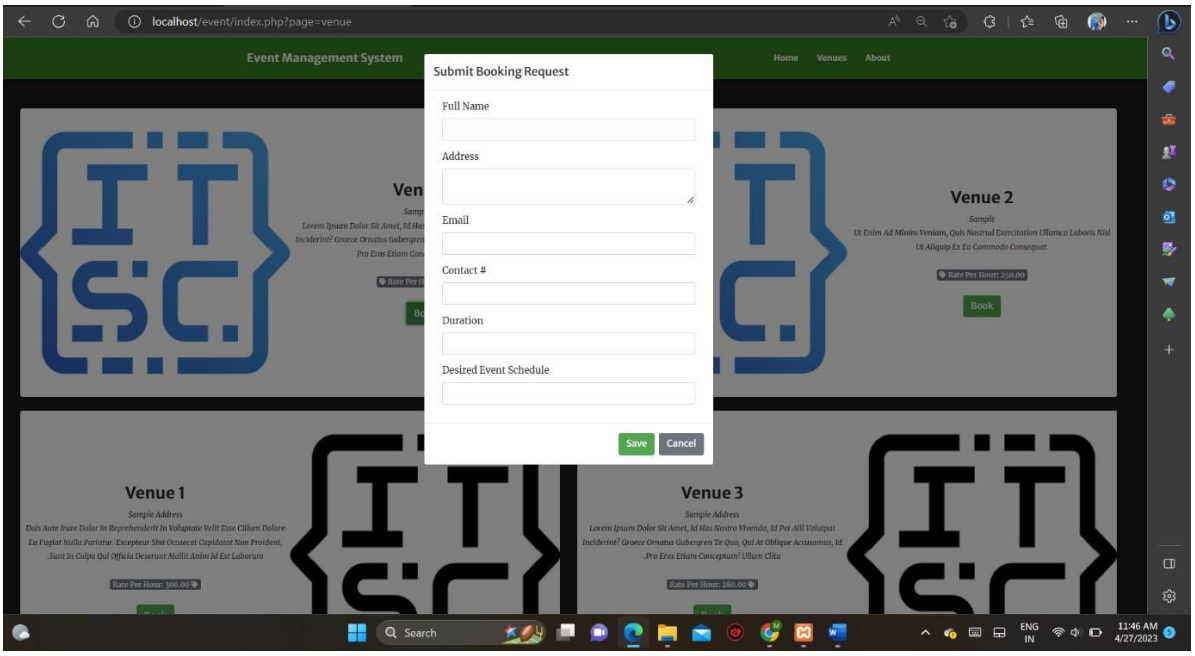

localhost/event/index.php?page=home
Event Management System
Home
Venues
About
Welcome to Event Management System
Upcoming Events
Contact us
+6948 8542 623
sample@sample.com
Copyright © 2021 - Event Management System | ITSOURCECODE
Search
ENG
IN
11:44 AM
4/27/2023


localhost/event/index.php?page=about
Event Management System
Home
Venues
About
ABOUT US
is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
Search
ENG
IN
11:45 AM
4/27/2023


localhost/event/index.php?page=venue
Event Management System
Home
Venues
About
We will contact you soon for the verification of your book request. Thank you
Close
book Request Sent.
Venue 4
Sample only
Lorem Ipsum Dolor Sit Amet, Id Has Nostro Vivendo, Id Per Alii Volutpat Inciderint? Graece Ornatus Gubergren Te Quo, Qui At Oblique Accusamus, Id Pro Eros Etiam Conceptam! Ullum Clita.
Rate Per Hour: 1,000.00
Book
Venue 2
Sample
Ut Enim Ad Minim Veniam, Quis Nostrud Exercitation Ullamco Laboris Nisi Ut Aliquip Ex Ea Commodo Consequat.
Rate Per Hour: 250.00
Book
Venue 1
Sample Address
Duis Aute Irure Dolor In Reprehenderit In Voluptate Velit Esse Cillum Dolore Eu Fugiat Nulla Pariatur. Excepteur Sint Occaecat Cupidatat Non Proident, Sunt In Culpa Qui Officia Deserunt Mollit Anim Id Est Laborum
Rate Per Hour: 300.00
Venue 3
Sample Address
Lorem Ipsum Dolor Sit Amet, Id Has Nostro Vivendo, Id Per Alii Volutpat Inciderint? Graece Ornatus Gubergren Te Quo, Qui At Oblique Accusamus, Id Pro Eros Etiam Conceptam! Ullum Clita
Rate Per Hour: 280.00
Search
ENG
IN
11:48 AM
4/27/2023

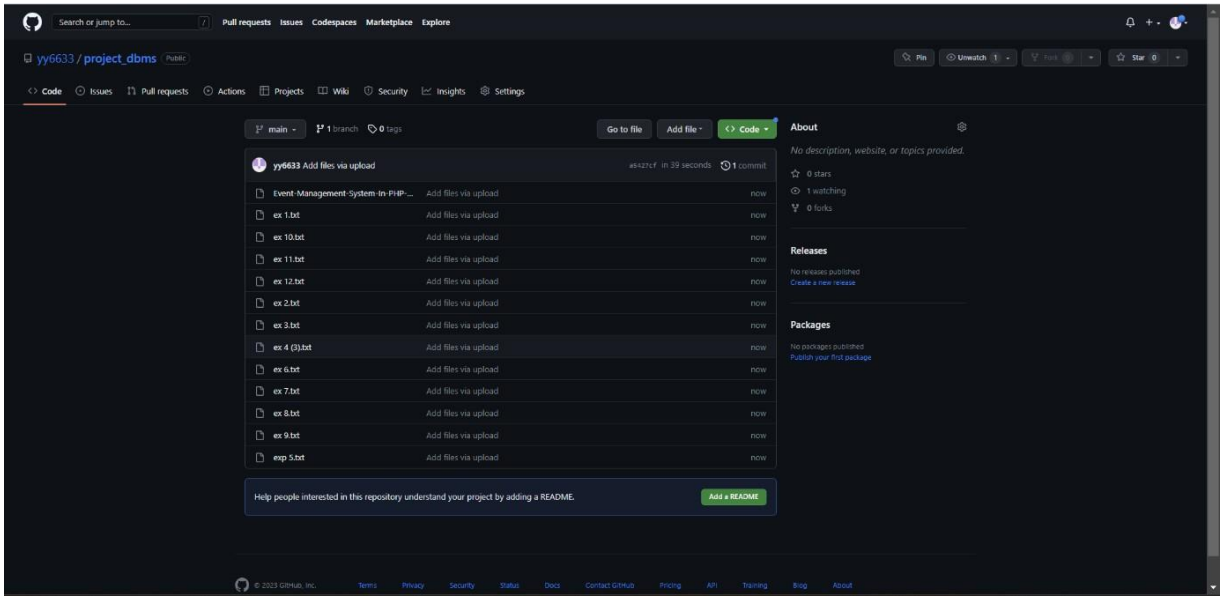

localhost/event/index.php?page=venue
Event Management System
Home
Venues
About
List of Our Event Venues
Venue 4
Sample only
Lorem Ipsum Dolor Sit Amet, Id Has Nostro Vivendo, Id Per Alii Volutpat Inciderint? Graece Ornatus Gubergren Te Quo, Qui At Oblique Accusamus, Id Pro Eros Etiam Conceptam! Ullum Clita.
Rate Per Hour: 1,000.00
Book
Venue 3
Sample Address
Lorem Ipsum Dolor Sit Amet, Id Has Nostro Vivendo, Id Per Alii Volutpat Inciderint? Graece Ornatus Gubergren Te Quo, Qui At Oblique Accusamus, Id Pro Eros Etiam Conceptam! Ullum Clita.
Rate Per Hour: 280.00
Book
Search
ENG
IN
11:44 AM
4/27/2023


localhost/event/index.php?page=venue
Event Management System
Home
Venues
About
Venue 2
Sample
Ut Enim Ad Minim Veniam, Quis Nostrud Exercitation Ullamco Laboris Nisi
Ut Aliquip Ex Ea Commodo Consequat
Rate Per Hour: 250.00
Book
Venue 1
Sample Address
Duis Aute Irure Dolor In Reprehenderit In Voluptate Velit Esse Cillum Dolore
Eu Fugiat Nulla Pariatur. Excepteur Sint Occaecat Cupidatat Non Proident,
Sunt In Culpa Qui Officia Deserunt Mollit Anim Id Est Laborum
Rate Per Hour: 300.00
Book
Contact us
+6948 8542 623
sample@sample.com
Copyright © 2021 - Event Management System | ITSOURCECODE
Search
ENG
IN
11:45 AM
4/27/2023

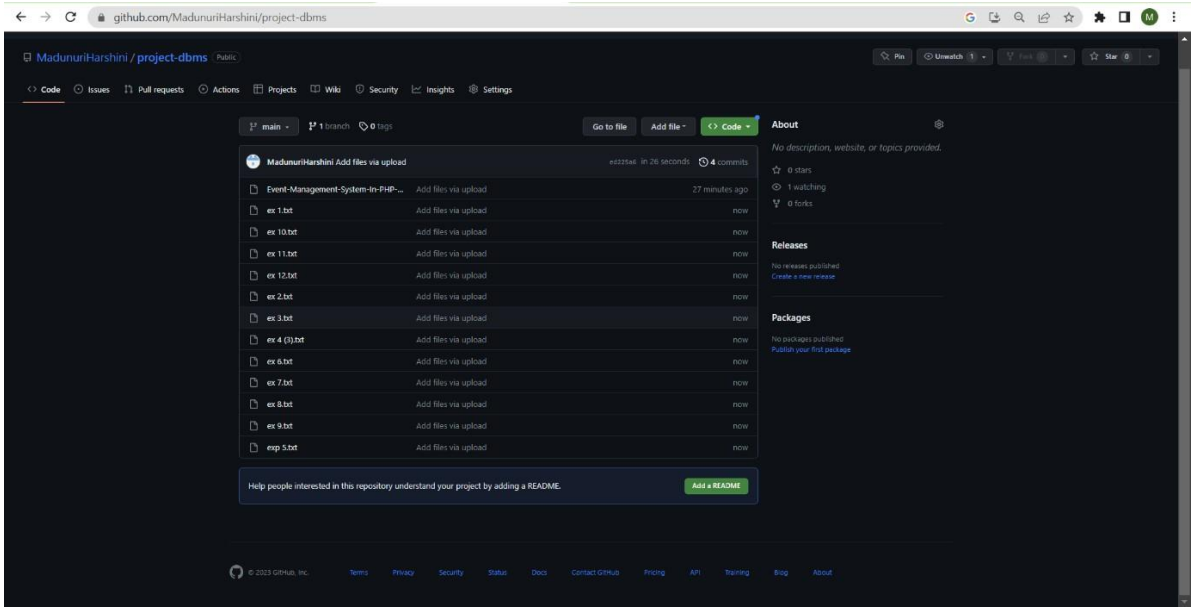

localhost/event/index.php?page=venue
Event Management System
Home
Venues
About
Submit Booking Request
Full Name
Address
Email
Contact #
Duration
Desired Event Schedule
Save
Cancel
Venue 2
Sample
Ut Enim Ad Minim Veniam, Quis Nostrud Exercitation Ullamco Laboris Nisi
Ut Aliquip Ex Ea Commodo Consequat.
Rate Per Hour: 250.00
Book
Venue 1
Sample Address
Duis Aute Irure Dolor In Reprehenderit In Voluptate Velit Esse Cillum Dolore
Eu Fugiat Nulla Pariatur. Excepteur Sint Occaecat Cupidatat Non Proident,
Sunt In Culpa Qui Officia Deserunt Mollit Anim Id Est Laborum
Rate Per Hour: 300.00
Venue 3
Sample Address
Rate Per Hour: 280.00
Search
ENG
IN
11:46 AM
4/27/2023

GITHUB PROFILE:

Yeluri Badrinath[RA2111028010111]



Madunuri Harshini[RA2111028010112]



AWS Course Completion certificate:

MADUNURIHARSHINI[RA2111028010112]

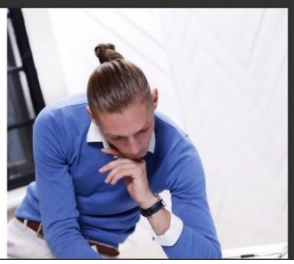
YeluriBadrinath[RA2111028010111]

PPT:

Event Management System

Introduction

Event Planning can be overwhelming. But you can **streamline** the process with an **efficient event management system**. Join us to learn how.



Challenges of Event Planning

Event planning involves **multiple tasks** such as **venue selection, vendor management, attendee registration**, and more. These tasks can become **chaotic** and **time-consuming** without a proper system in place.

The Benefits of an Event Management System

An **event management system** can **simplify** the event planning process by **automating** tasks, **centralizing** information, and **providing real-time updates**. This leads to **greater efficiency, accuracy** and **saves time**.

Key Features of an Event Management System

An event management system should include features such as **customizable registration forms**, **automated email communication**, **payment processing**, and **data analytics**. These features can help **streamline** the event planning process.

Implementing an Event Management System

When implementing an event management system, it is important to **train staff**, **test the system**, and **integrate it with other tools**. This will help ensure a **successful implementation** and **maximize benefits**.

Implementing an Event Management System

When implementing an event management system, it is important to **train staff**, **test the system**, and **integrate it with other tools**. This will help ensure a **successful implementation** and **maximize benefits**.

Thanks!

DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES – 18CSC109J

B.Tech CSE (Cloud Computing)

Student Portfolio

Name	:Madunuri Harshini
Reg No	:RA2111028010112
Year	:2ND Year
Section	:P1
Course Teacher	: Dr. B. Balakiruthiga

List of Assignments:

- SQL Query formulation
- ER Diagram
- Normalization of Database
- PLSQL Programs

Signature of the student

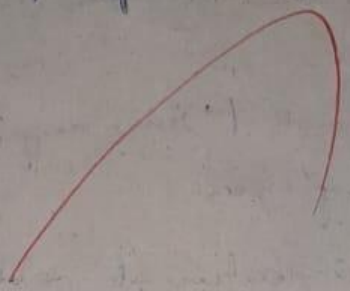
Harshini

Q2. Declare
dept_id number := 10;
emp_count Number;
BEGIN
SELECT count (*) into emp_count
FROM employees
WHERE department_id = dept_id;
DBMS_output.PUT_line
END;

Q1. CREATE OR REPLACE PROCEDURE
Count_employee_by_dept (dptid number)
IS
emp_counts NUMBER;
BEGIN
SELECT count (*) INTO emp_count
FROM employees
WHERE department_id = dept_id
DBMS_OUTPUT.PUT_LINE
END;

Q3. CREATE OR REPLACE PROCEDURE
get_hiredate(
p_job IN employees.job_id%TYPE,
P_department IN
employees.department_id%TYPE,
p_hiredate OUT employees.hire_date%TYPE) AS
BEGIN
SELECT hire_date INTO p_hiredate
FROM employees

```
WHERE job = p_job
AND department_id = p_department;
END;
/
```



22/02/23

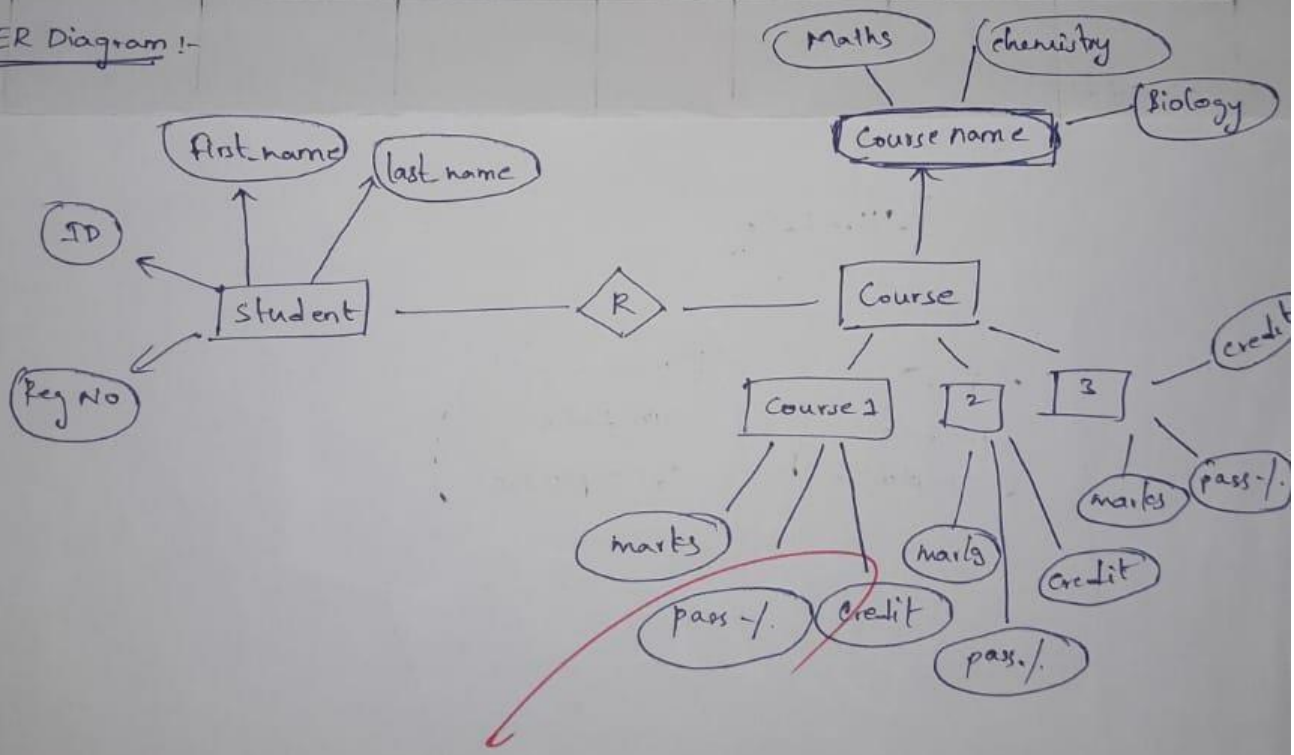
DBMS

MADUNURI HARSHINI
RA2111028010112

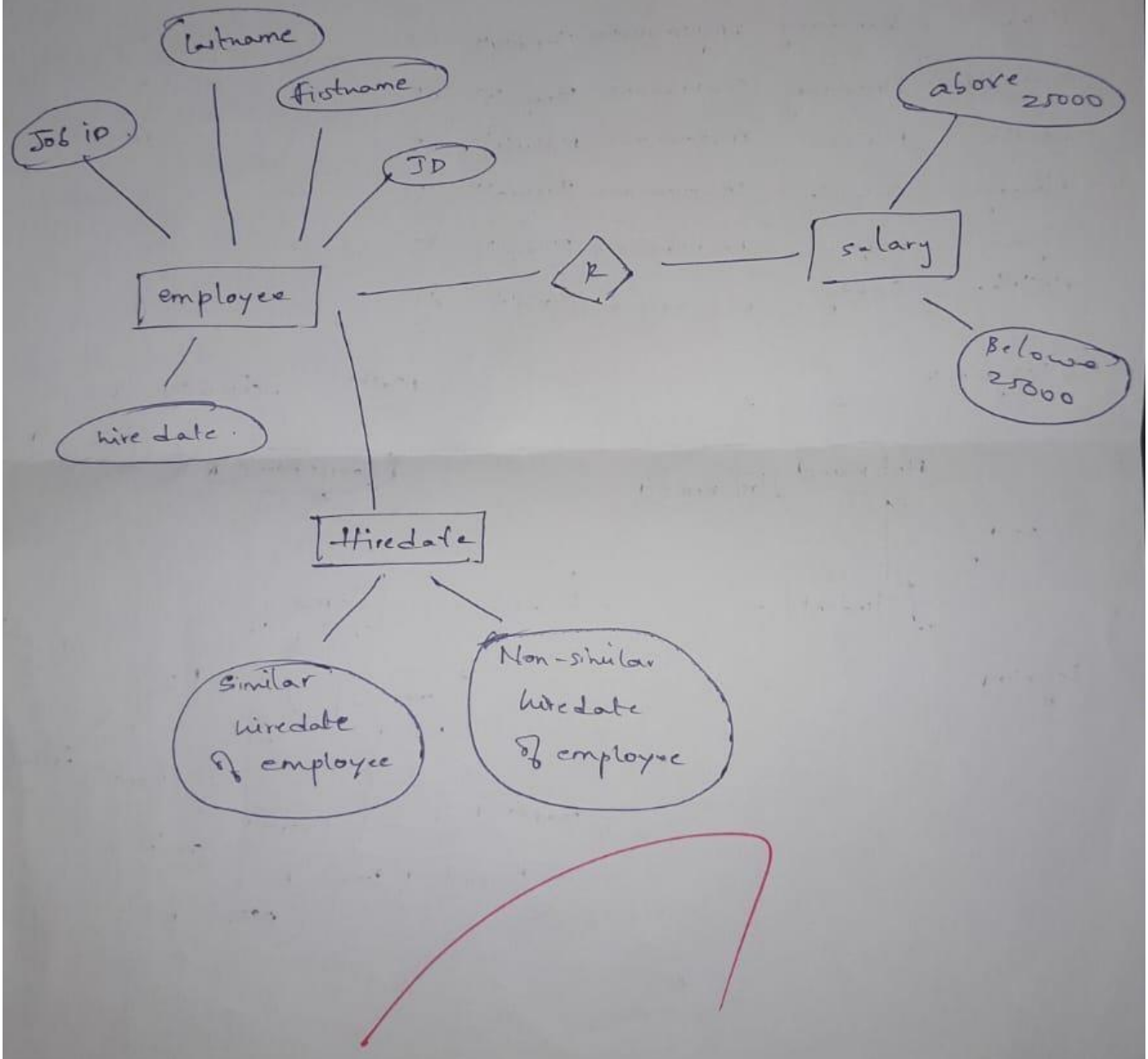
1. ⇒ Tablet

Student First Name	student last Name	Reg NO	Course 1/2/3	Marks	pass-/.	Credit
Madunuri	Harshini	RA2111028010112	Maths(1)	90	90-/.	4
Neha	Bhardwaj	RA2111028010104	chemistry(2)	80	80-/.	3
Sunandhini	chowdary	RA2111028010105	Biology(3)	85	80-/.	2
Bandla	Malini	RA2111028010113	Maths(1)	75	70-/.	4
Abohaya	Sowmithri	RA2111028010116	chemistry(2)	95	90-/.	3
Hyma	Khoyrnal	RA2111028010117	Biology(3)	93	90-/.	2
Nikhitha	Reddy.	RA2111028010112	Maths(1)	81	80-/.	4

⇒ ER Diagram :-



2. Consider employee database table with Attributes employee ID, firstname, lastname, job id and hire date and relationship b/w employee with similar hiredate and sal above 25000.



DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES – 18CSC109J

B.Tech CSE (Cloud Computing)

Student Portfolio

Name :Yeluri Badrinath

Reg No :RA2111028010111

Year :2ND YEAR

Section :P1

Course Teacher : Dr. B. Balakiruthiga

List of Assignments:

- SQL Query formulation
- ER Diagram
- Normalization of Database
- PLSQL Programs

Signature of the student

RA2111028010112
Harshini

Q2 Declare

dept - idNumber := 10,

Emp - CsumtNumber,

BEGIN

SELECT count (d) into Emp-count

FROM Employes

where departement-id = dept-id,

DBMS-output: put-line

END;

Q1. CREATE OR REPLACE PROCEPORE

count-Employe-by-dept-id dptid Number

;)

Emp-counti (Number);

BEGIN

SELECT count (*) INTO Emp-count

FROM Employes

WHERE department-d = dep-id

DBMS-Output put-LINE

END;

Q3. CREATE OR REPLACE PROCEDURE

get_hiredata(

i_job IN employes.job%Type,

p_department_id employes.department_id%Type,

- cursdate OUT demployee.we.date%Type) AS

BEGIN

SELECT hire_date INTO p_hiredate

FROM employees

WHERE job = p_job

AND department_id = p_department;

END;

/

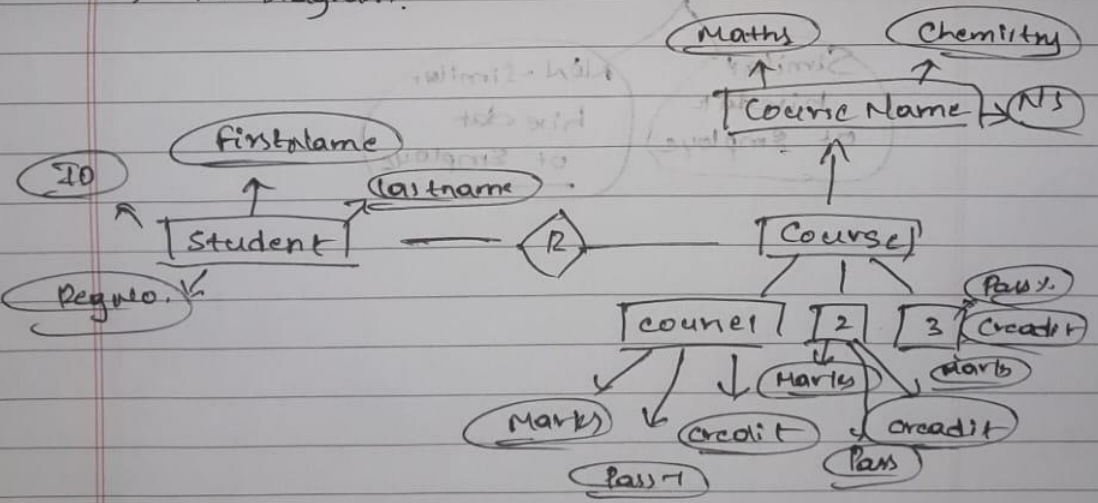
22/02/23 DBMS

Y. BADRINATH
RA2111028010111.

1.) Table:

Student first name	Student last name	Reg. No	course 1/2/3	Mark	Pass	Credit
Badrinath	Yeleni	RA2111028010111	Maths	90	90%	4
Harshin	Madumoni	RA2111028010112	Chemistry	80	80%	3
Neha	Bharadwaj	RA2111028010104	NS(3)	85	80%	2
Sai	Kerulla	RA2111028010110	Maths	75	70%	4
Maharthi	Paravasthu	RA2111028010	Chemistry	95	90%	3
Sai	Yakala	RA2111002804	NS(3)	93	90%	2
Buppy	Gunda	RA2111028010114	Maths	81	86	4

=) ER Diagram:



2. Consider employee database table with Attributes Employee ID, firstname, lastname, job id and hire data and Relationship bw Employee with similar hiredate and sal above 2000.