SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(An Autonomous Institution)
Adyar, Mangaluru-575007
2023-24

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A PROJECT REPORT

ON

Department Activities and Resources Management System

BY

CHAITHANYA K J 4SF21CS032

PRITHVI P SHETTY 4SF21CS115

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

DBMS LABORATORY WITH MINI PROJECT

Under the guidance of

Ms. Ashwini C S
Assistant Professor, Dept. of CS&E

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(An Autonomous Institution)
Adyar, Mangaluru-575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**Department Activities and Resources Management System"** is submitted in partial fulfillment for the requirement of V sem. B. E. (Computer Science & Engineering), "DATABASE MANAGEMENT SYSTEMS LABORATORY" during the year 2023 – 24 is a result of bonafide work carried out by

Chaithanya K J	**4SF21CS032**
Prithvi P Shetty	**4SF21CS115**

……	……..
Ms. Ashwini C S	**Dr. Mustafa Basthikodi**
Asst. Prof. Dept. of CS&E	**HOD, Dept. of CS&E**
SCEM, Mangaluru	**SCEM, Mangaluru**

Signature of the Examiners

1. **…….**
2. **…….**

ABSTRACT

A Database Management System (DBMS) refers to the technology for creating and managing databases. DBMS is a collection of inter-related data and a set of programs and a software tool to organize create, retrieve, update and manage data in a database. The DBMS software additionally comprises the core facilities to administer the database. The main aim of DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient.

This project introduces a Department Resource and Activity Management System centered around facilitating efficient management of the departmental activities like webinars, seminars, workshops, and skill labs. The DBMS enables users to insert, delete, update, sort, search and access information regarding these events. Through a user-friendly interface, individuals can keep a record of activities, monitor them. The implementation of this system has significantly streamlined the management of departmental resources and activities, fostering better coordination, resource utilization, and overall productivity within the organization.

ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **"Department Activities and Resources Management System"** We have completed it as a part of the V semester DATABASE MANAGEMENT SYSTEMS LABORATORY (21CSL55) of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Ms. Ashwini C S, Assistant Professor, Department of Computer Science & Engineering** for her innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We express our sincere gratitude to **Dr. Mustafa Basthikodi, Professor & Head of the Department of Computer Science & Engineering** for his invaluable support and guidance.

We sincerely thankful to our beloved **Principal Dr. S S Injaganeri, Sahyadri College of Engineering & Management,** who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Chaithanya K J
4SF21CS032
V Sem, B.E., CSE
SCEM, Mangaluru

Prithvi P Shetty
4SF21CS115
V Sem, B.E., CSE
SCEM, Mangaluru

PAGE INDEX

CHAPTER 1

INTRODUCTION

1.1. Introduction to Database Management System

A Database Management System (DBMS) is a vital software tool designed for cataloging, retrieving, and executing queries on data in an organized manner. Essentially, it serves as the backbone of data management, handling incoming data, organizing it efficiently, and offering means for modification and extraction by users or other applications.

Users interact with a DBMS to create, read, update, and delete data within a database, ensuring consistent organization and easy accessibility. Acting as an intermediary between the database and end users or applications, the DBMS plays a crucial role in maintaining data integrity and facilitating smooth data operations.

Key functionalities of a DBMS include change management, performance monitoring and tuning, and backup and recovery. Additionally, many DBMSs automate rollbacks, restarts, and recovery processes, along with logging and auditing user activity. Moreover, DBMSs provide both logical and physical data independence, shielding users and applications from the complexities of data storage and hardware changes.

By utilizing the application programming interface (API) provided by the DBMS, developers can seamlessly integrate database functionality into their programs without the need for constant modifications. In relational DBMSs (RDBMSs), SQL serves as the standard programming language accessed through the API, ensuring compatibility and ease of use across different applications and systems.

1.2. MySQL Database Management:

MySQL served as the cornerstone for storing and managing data essential to the Cleanliness Management System. Leveraging its robustness and scalability, MySQL provided a reliable platform for efficient data storage and retrieval.

1.3. phpMyAdmin for Database Administration:

phpMyAdmin, a web-based administration tool, facilitated the administration of MySQL databases. Its intuitive interface and comprehensive features streamlined database management tasks such as querying, importing, and exporting data. PHP was employed for server-side scripting, enabling dynamic interaction with the MySQL database.

1.4. Visual Studio Code (VSCode) as the IDE:

VSCode emerged as the primary Integrated Development Environment (IDE) for coding and managing project files. Its lightweight yet powerful features, coupled with an extensive ecosystem of extensions, provided developers with a versatile platform for writing, debugging, and organizing code efficiently.

By adopting a modular approach and leveraging the strengths of each software component, the integration approach ensured efficient development and seamless connectivity between the database and application logic. This cohesive integration facilitated streamlined workflows, enhanced productivity, and ultimately contributed to the successful deployment of the Cleanliness Management System within the organization.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1. Hardware Requirements:

- Processor: Any processor above 500 MHz
- RAM: 4GB
- Hard Disk: 500GB
- Input Device: Standard keyboard and Mouse
- Output Device: Monitor

2.2. Software Requirements:

- Database: MySQL (phpMyAdmin)
- Programming Language: php
- IDE: Visual Studio Code (VSCode)

CHAPTER 3

DESIGN

3.1. ER diagram

An entity-relationship mode describes interrelated things of interest in a specific domain of knowledge. The ER Diagram of our project is shown in the Figure: 3.1.2.

3.1.1. Symbols in Entity Relationship:

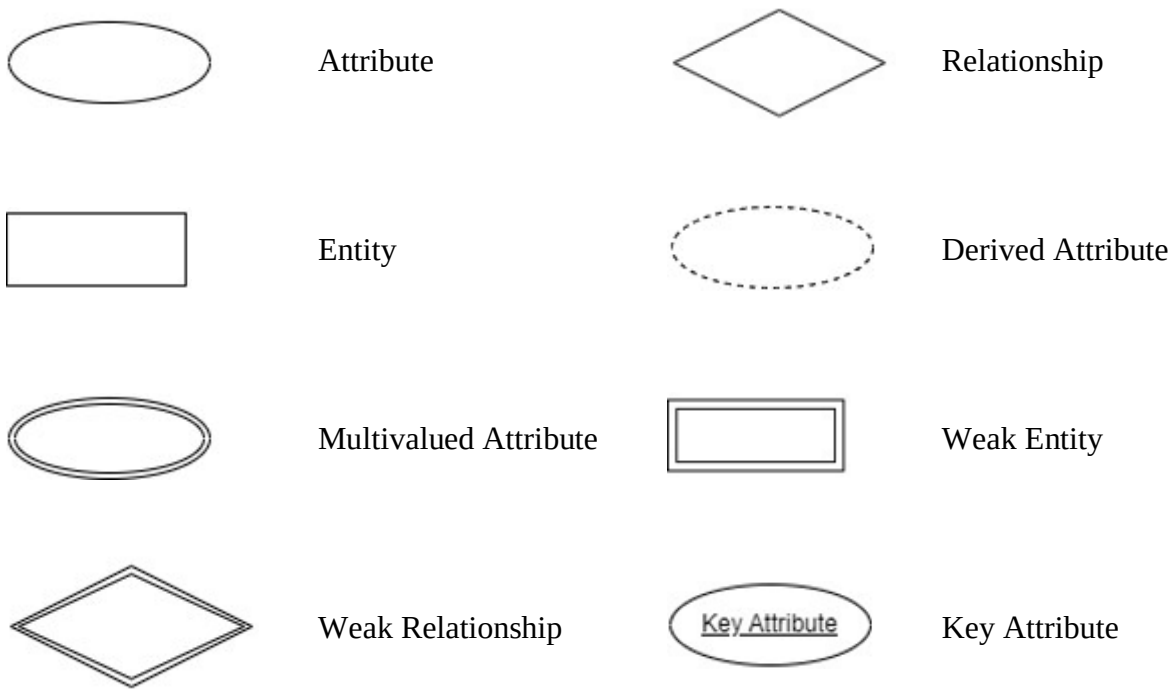


Fig: 3.1.1 ER Notations used.

Attribute: Describes a characteristic or property of an entity.

Entity: Represents a real-world object or concept in a database.

Derived Attribute: Calculated or derived from other attributes within the database.

Multivalued Attribute: Can hold multiple values for a single entity instance.

Weak Entity: Depends on another entity for its existence and typically does not have a primary key.

Weak Relationship: Connects a weak entity to its identifying strong entity.

Key Attribute: An attribute that uniquely identifies an entity within a database relation.

3.1.2. ER diagram of Department Activites and Resources Management System

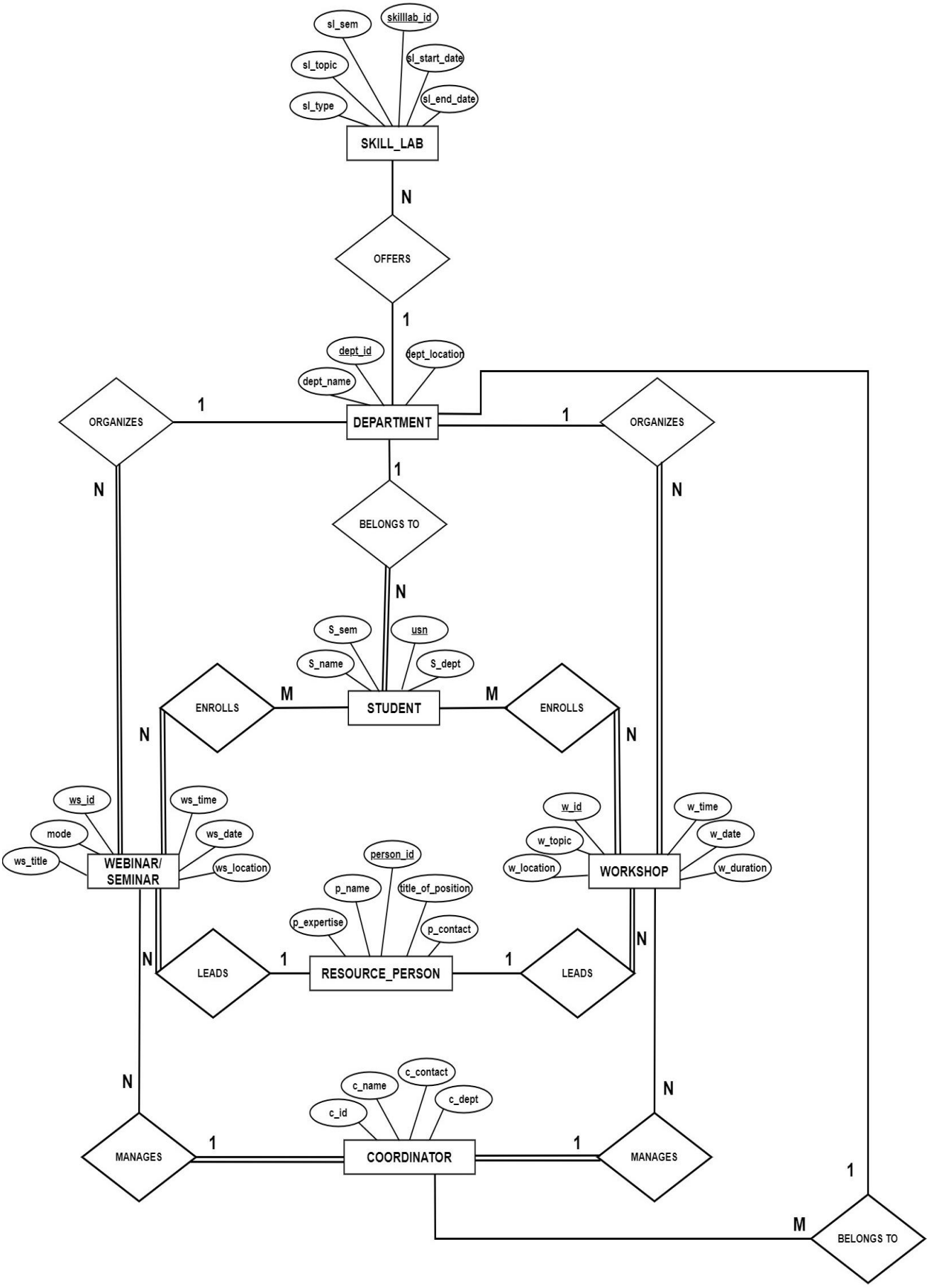


Figure 3.1.2 ER diagram of Department Activity and Resources Management System

3.2. Relational Schema

3.2.1. Mapping of Regular Entity Types

For every entity in our relationship diagram, we have created a separate relation. These created relations contain the respected attributes and respected primary key.

COORDINATOR

C_ID	C_NAME	C_DEPT	C_CONTACT	DEPT_ID

DEPARTMENT

DEPT_ID	DEPT_NAME	DEPT_LOCATION

RESOURCE_PERSON

PERSON_ID	P_NAME	P_CONTACT	TITLE_OF_POSITION	P_EXPERTISE

SEMINAR_WEBINAR

WS_ID	MODE	WS_TITLE	WS_TIME	WS_DATE	WS_LOCATION	DEPT_ID	PERSON_ID	C_ID

SKILL_LAB

SKILLLAB_ID	SL_TYPE	SL_SEM	SL_TOPIC	SL_START_DATE	SL_SEND_DATE	DEPT_ID

STUDENT

USN	S_NAME	S_SEM	S_DEPT	DEPT_ID

SEMINAR_WEBINAR

WS_ID	MODE	WS_TITLE	WS_TIME	WS_DATE	WS_LOCATION	DEPT_ID	PERSON_ID	C_ID

WORKSHOP

W_ID	W_TOPIC	W_LOCATION	W_TIME	W_DATE	W_DURATION	DEPT_ID	PERSON_ID	C_ID

Figure 3.2.1 Mapping regular entites for Department Activity and Resources Management System

3.2.2. Mapping of Weak Entity Types

The ER Diagram contains no weak entities, this step is ignored in this project.

3.2.3. Mapping of Binary 1:1 Relationship Types

The ER diagram contains no 1:1 entity, this step is ignored in the project.

3.2.4. Mapping of Binary 1: N Relationship Types

There are seven 1: N relation in schema diagram. There is a foreign key approach and have included the primary key of one relation as the foreign key in the N-sided relation.

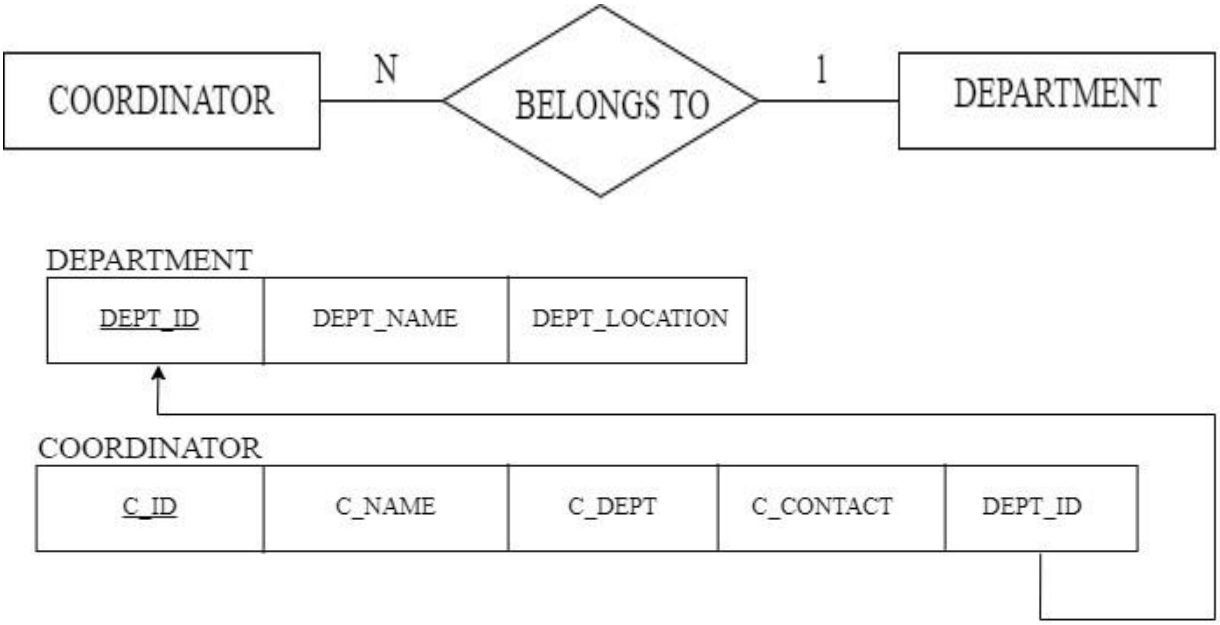


Fig 3.2.2 1:N relationship between Department and Coordinator.

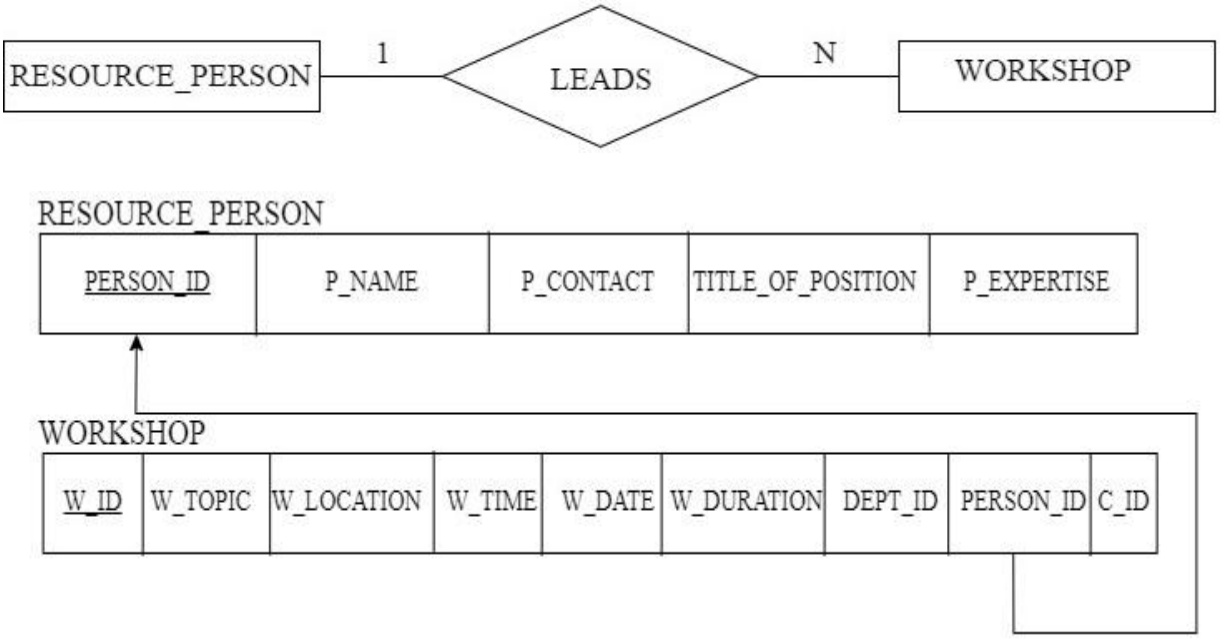


Fig 3.2.3 1:N relationship between Resource_person and Workshop.

3.2.5. Mapping of Binary M: N Relationship Types

There are two M: N relation in schema diagram. Mapping of Binary M: N Relationship Types involves the introduction of a junction table to manage the many-to-many relationship between two entities. This junction table contains foreign keys referencing the primary keys of the entities involved, facilitating seamless association and retrieval of data across the relationship.

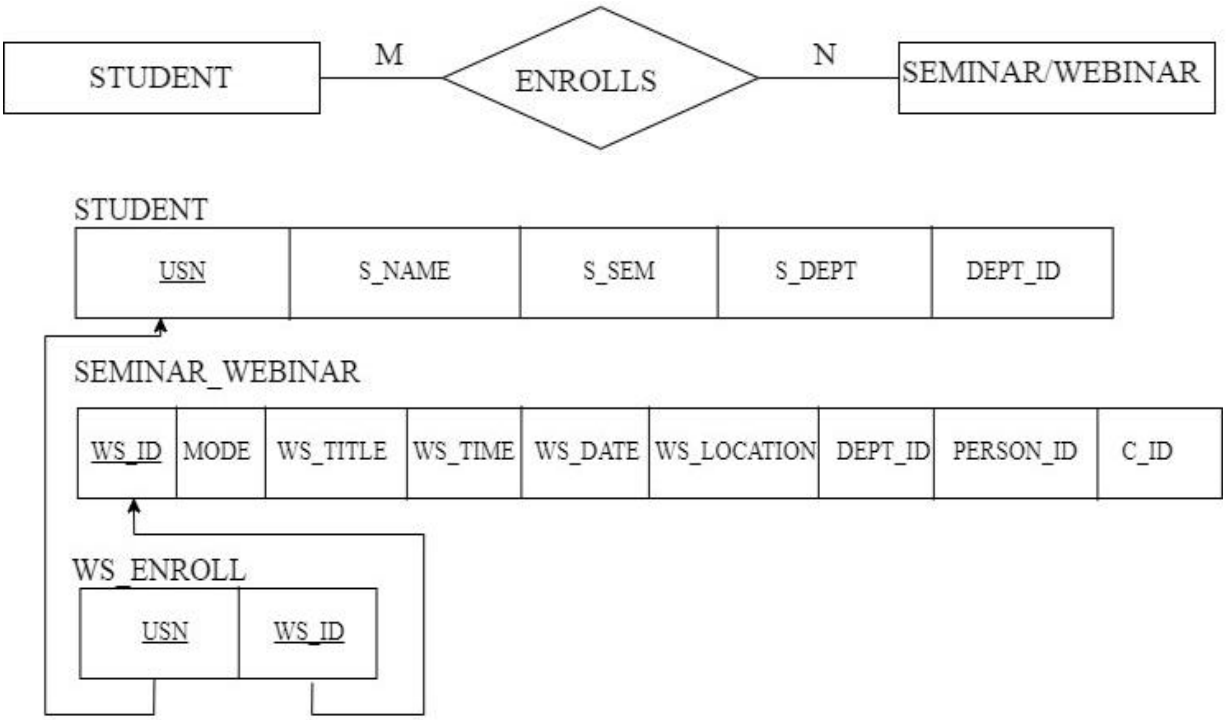


Fig 3.2.4 M: N relationship between Student and Seminar_Webinar

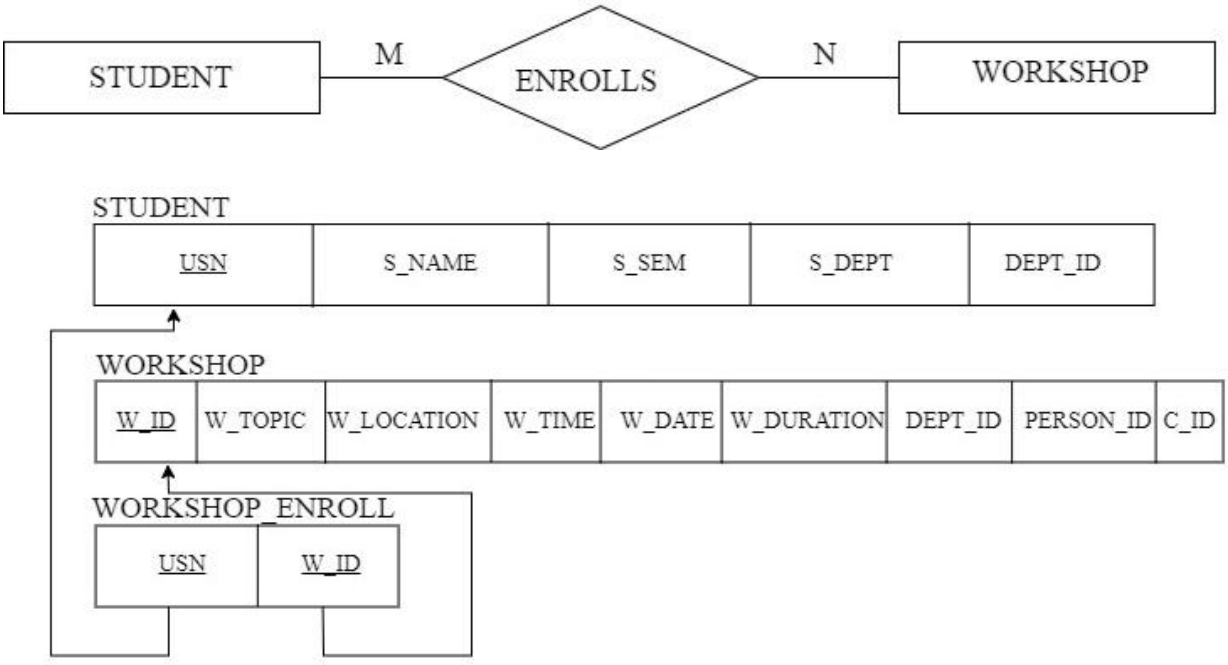


Fig 3.2.5 M:N relationship between Student and Workshop.

3.2.6. Mapping of Multivalued Attributes

The ER Diagram contains no multivalued attributes, this step is ignored for our project.

3.2.7. Mapping of N-ary Relationship Types

The ER Diagram contains no N-ary relationship, this step is ignored in this project.

3.3. Schema diagram

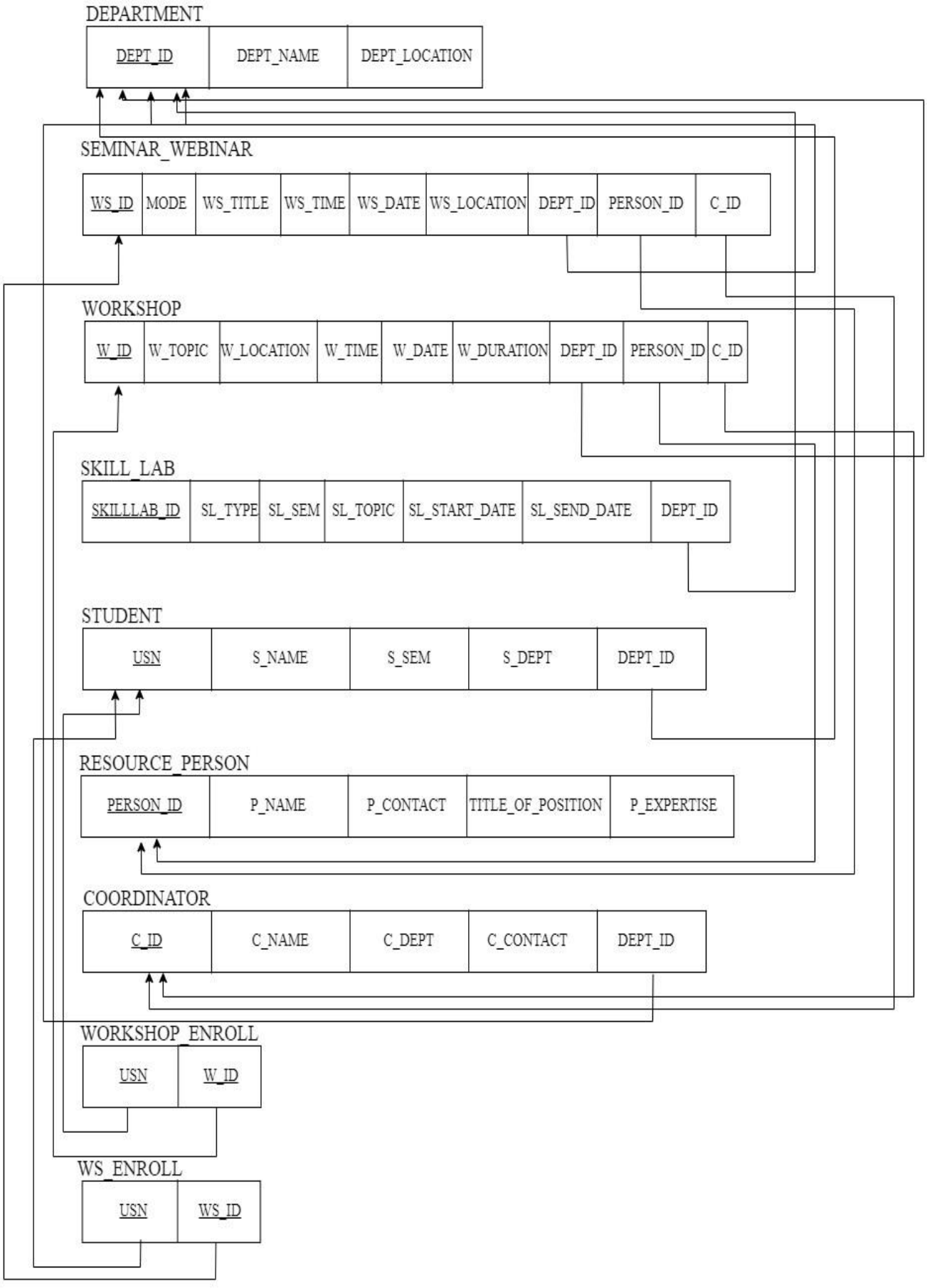
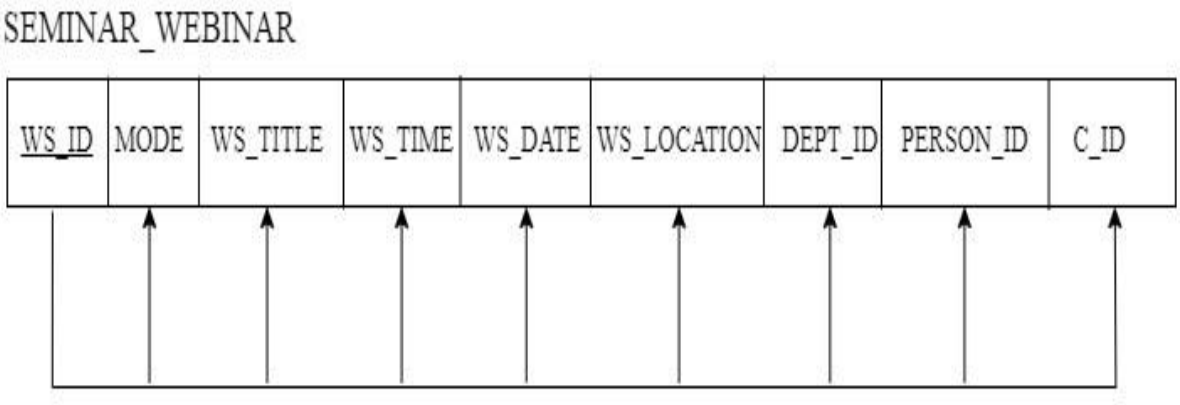


Fig 3.3. Schema Diagram.

CHAPTER 4

NORMALIZATION

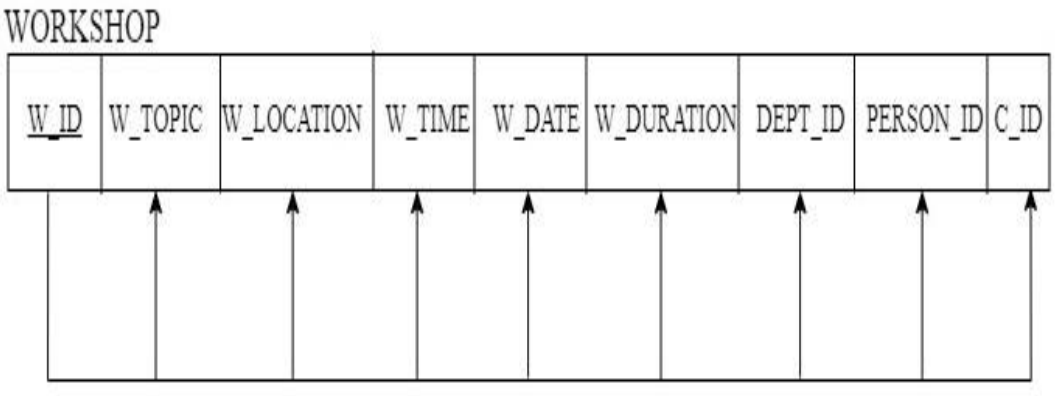
SEMINAR_WEBINAR



FD1:{WS_ID} ->
{MODE,WS_TITLE,WS_TIME,WS_DATE,WS_LOCATION,DEPT_ID,PERSON_ID,C_ID}

- The SEMINAR_WEBINAR relation is in 1NF because there are no multivalued attributes in the relation schema.
- It is in 2NF because the relation is in 1NF and all the attributes in the relation schema are fully functionally dependent on the primary key.
- It is in 3NF because the relation is in 2NF and there are no transitive dependencies.

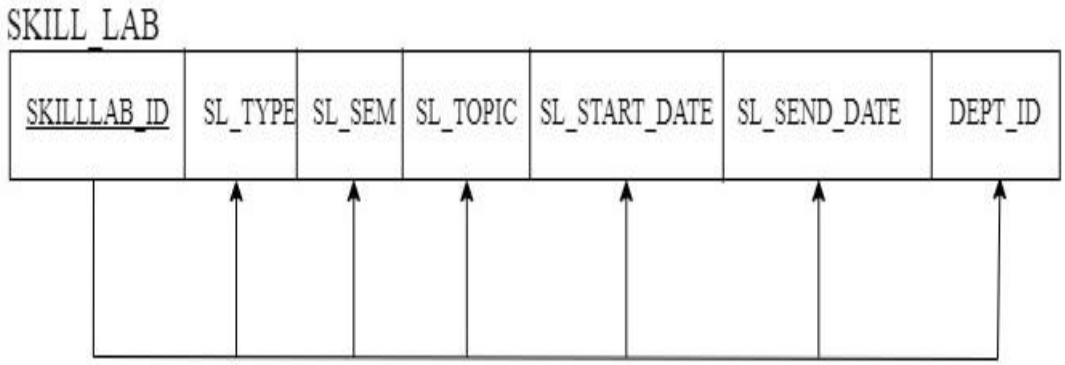
WORKSHOP



FD2: {W_ID }-> {W_TOPIC, W_LOCATION, W_TIME,
W_DATE,W_DURATION,DEPT_ID,PERSON_ID,C_ID}

- The WORKSHOP relation is in 1NF because there are no multivalued attributes in the relation schema.
- It is in 2NF because the relation is in 1NF and all the attributes in the relation schema are fully functionally dependent on the primary key.
- It is in 3NF because the relation is in 2NF and there are no transitive dependencies.

SKILL_LAB



FD3: {SKILLLAB_ID}->
{SL_TYPE,SL_SEM,SL_TOPIC,SL_START_DATE,SL_END_DATE,DEPT_ID}

- The SKILL_LAB relation is in 1NF because there are no multivalued attributes in the relation schema.
- It is in 2NF because the relation is in 1NF and all the attributes in the relation schema are fully functionally dependent on the primary key.
- It is in 3NF because the relation is in 2NF and there are no transitive dependencies.

CHAPTER 5

IMPLEMENTATION

5.1. TABLE STRUCTURE

5.1.1. DEPARTMENT RELATION

```
CREATE TABLE `department` (
  `dept_id` varchar(25) NOT NULL,
  `dept_name` varchar(25) DEFAULT NULL,
  `dept_location` varchar(25) DEFAULT NULL
   PRIMARY KEY(dept_id)
);
```

Column Name	Data Type	Constraints
dept_id	varchar(25)	NOT NULL
dept_name	varchar(25)	DEFAULT NULL
dept_location	varchar(25)	DEFAULT NULL

Fig 5.1.1 Department Relation

5.1.2. RESOURCE_PERSON RELATION

```
CREATE TABLE `resource_person` (
  `person_id` varchar(25) NOT NULL,
  `p_name` varchar(25) DEFAULT NULL,
  `p_contact` bigint(10) DEFAULT NULL,
  `title_of_position` varchar(80) DEFAULT NULL,
  `p_expertise` varchar(80) DEFAULT NULL
   PRIMARY KEY(person_id)
);
```

Column Name	Data Type	Constraints
person_id	varchar(25)	NOT NULL
p_name	varchar(25)	DEFAULT NULL
p_contact	bigint(10)	DEFAULT NULL
title_of_position	varchar(80)	DEFAULT NULL
p_expertise	varchar(80)	DEFAULT NULL

Fig 5.1.2 Resource_person Relation

5.1.3. WORKSHOP RELATION

```
CREATE TABLE `workshop` (
  `w_id` varchar(25) NOT NULL,
  `w_topic` varchar(75) DEFAULT NULL,
  `w_location` varchar(25) DEFAULT NULL,
  `w_time` time DEFAULT NULL,
  `w_date` date DEFAULT NULL,
  `w_duration` int(5) DEFAULT NULL,
  `dept_id` varchar(25) DEFAULT NULL references department(dept_id),
  `person_id` varchar(25) DEFAULT NULL references resource_person(person_id),
  `c_id` varchar(25) DEFAULT NULL references coordinator(c_id),
   PRIMARY KEY(w_id)
);
```

Column	Data Type	Constraints
w_id	VARCHAR(25)	NOT NULL
w_topic	VARCHAR(75)	DEFAULT NULL
w_location	VARCHAR(25)	DEFAULT NULL
w_time	TIME	DEFAULT NULL
w_date	DATE	DEFAULT NULL
w_duration	INT(5)	DEFAULT NULL
dept_id	VARCHAR(25)	DEFAULT NULL
person_id	VARCHAR(25)	DEFAULT NULL
c_id	VARCHAR(25)	DEFAULT NULL

Fig 5.1.3 Workshop Relation

5.1.4. WORKSHOP_ENROLL RELATION

```
CREATE TABLE `workshop_enroll` (
  `USN` varchar(25) NOT NULL references student(USN),
  `w_id` varchar(25) NOT NULL references workshop(w_id),
   PRIMARY KEY (USN, w_id)
) ;
```

Column Name	Data Type	Constraints
USN	varchar(25)	NOT NULL, references student(USN)
w_id	varchar(25)	NOT NULL, references workshop(w_id)
Primary Key	(USN, w_id)	

Fig 5.1.4 workshop_enroll Relation

5.1.5. SEMINAR_WEBINAR RELATION

```
CREATE TABLE `seminar_webiner` (
  `ws_id` varchar(25) NOT NULL,
  `mode` varchar(25) DEFAULT NULL,
  `ws_title` varchar(100) DEFAULT NULL,
  `ws_time` time DEFAULT NULL,
  `ws_date` date DEFAULT NULL,
  `ws_location` varchar(25) DEFAULT NULL,
  `dept_id` varchar(25) DEFAULT NULL references department(dept_id),
  `person_id` varchar(25) DEFAULT NULL references resource_person(person_id),
  `c_id` varchar(25) DEFAULT NULL references coordinator(c_id),
   PRIMARY KEY(ws_id)
);
```

Column Name	Data Type	Constraints
ws_id	varchar(25)	NOT NULL
mode	varchar(25)	DEFAULT NULL
ws_title	varchar(100)	DEFAULT NULL
ws_time	time	DEFAULT NULL
ws_date	date	DEFAULT NULL
ws_location	varchar(25)	DEFAULT NULL
dept_id	varchar(25)	references department(dept_id)
person_id	varchar(25)	references resource_person(person_id)
c_id	varchar(25)	references coordinator(c_id)

Fig 5.1.5 Seminar_Webinar Relation

5.1.6. WS_ENROLL RELATION

```
CREATE TABLE `ws_enroll` (
  `USN` varchar(25) NOT NULL references student(USN),
  `ws_id` varchar(25) NOT NULL references seminar_webiner(ws_id),
   PRIMARY KEY (USN, ws_id)
) ;
```

Column Name	Data Type	Constraints
USN	varchar(25)	NOT NULL, references student(USN)
ws_id	varchar(25)	NOT NULL, references seminar_webiner(ws_id)
Primary Key	(USN, ws_id)	

Fig 5.1.6 ws_enroll Relation

5.1.7. COORDINATOR RELATION

```
CREATE TABLE `coordinator` (
  `c_id` varchar(25) NOT NULL,
  `c_name` varchar(25) DEFAULT NULL,
  `c_dept` varchar(25) DEFAULT NULL,
  `c_contact` bigint(10) DEFAULT NULL,
   PRIMARY KEY (c_id),
  `dept_id` varchar(25) DEFAULT NULL references department(dept_id)
);
```

Column Name	Data Type	Constraints
c_id	varchar(25)	NOT NULL
c_name	varchar(25)	DEFAULT NULL
c_dept	varchar(25)	DEFAULT NULL
c_contact	bigint(10)	DEFAULT NULL
Primary Key	(c_id)	
dept_id	varchar(25)	references department(dept_id)

Fig 5.1.7 Coordinator Relation

5.1.8. STUDENT RELATION

```
CREATE TABLE `student` (
  `USN` varchar(25) NOT NULL,
  `s_name` varchar(25) DEFAULT NULL,
  `s_sem` int(5) DEFAULT NULL,
  `s_dept` varchar(25) DEFAULT NULL,
  `dept_id` varchar(25) DEFAULT NULL references department(dept_id),
  PRIMARY KEY(USN)
);
```

Column Name	Data Type	Constraints
USN	varchar(25)	NOT NULL
s_name	varchar(25)	DEFAULT NULL
s_sem	int(5)	DEFAULT NULL
s_dept	varchar(25)	DEFAULT NULL
dept_id	varchar(25)	references department(dept_id)

Fig 5.1.8 Student Relation

5.1.9. SKILL_LAB RELATION

```
CREATE TABLE `skill_lab` (
  `skilllab_id` varchar(25) NOT NULL,
  `sl_type` varchar(25) DEFAULT NULL,
  `sl_sem` varchar(5) DEFAULT NULL,
  `sl_topic` varchar(50) DEFAULT NULL,
  `sl_start_date` date DEFAULT NULL,
  `sl_end_date` date DEFAULT NULL,
  `dept_id` varchar(25) DEFAULT NULL references department(dept_id),
   PRIMARY KEY(skilllab_idtta )
);
```

Column Name	Data Type	Constraints
skilllab_id	varchar(25)	NOT NULL
sl_type	varchar(25)	DEFAULT NULL
sl_sem	varchar(5)	DEFAULT NULL
sl_topic	varchar(50)	DEFAULT NULL
sl_start_date	date	DEFAULT NULL
sl_end_date	date	DEFAULT NULL
dept_id	varchar(25)	references department(dept_id)

Fig 5.1.9 skill_lab Relation

5.2. FUNCTIONALITY

5.2.1. CONNECTING TO DATABASE

```
<?php
$HOSTNAME='localhost';
$USERNAME='root';
$PASSWORD='';
$DATABaSE='college';
$con=mysqli_connect($HOSTNAME,$USERNAME,$PASSWORD,$DATABaSE);
if($con){
   echo "connection successful";
}else{
   die(mysqli_error($con));
}
?>
```

5.2.2. SELECTION

```
<?php
$search = isset($_GET['search']) ? $_GET['search'] : '';
$searchQuery = $search ? "WHERE w.w_topic LIKE '%$search%'" : '';

$sql = "SELECT w.*, COUNT(we.USN) AS enrollment
       FROM workshop w
       LEFT JOIN workshop_enroll we ON w.w_id = we.w_id
       $searchQuery
       GROUP BY w.w_id
       ORDER BY w.w_date " . ($sort == 'latest' ? 'DESC' : 'ASC');

$result = mysqli_query($con, $sql);
?>
```

5.2.3. INSERTION

```
<?php
include 'connect.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $w_id = $_POST['w_id'];
   $w_topic = $_POST['w_topic'];
   $w_location = $_POST['w_location'];
   $w_time = $_POST['w_time'];
   $w_date = $_POST['w_date'];
   $w_duration = $_POST['w_duration'];
   $person_id = $_POST['person_id'];
   $c_id = $_POST['c_id'];
   $sql_workshop = "INSERT INTO workshop (w_id, w_topic, w_location, w_time, w_date, w_duration,
person_id, c_id)
                VALUES ('$w_id', '$w_topic', '$w_location', '$w_time', '$w_date', '$w_duration', '$person_id',
'$c_id')";
   if (mysqli_query($con, $sql_workshop)) {
      $enrollments = $_POST['enrollments'];
      $usns = $_POST['usn'];
      foreach ($usns as $usn) {
         $sql_enroll = "INSERT INTO workshop_enroll (USN, w_id) VALUES ('$usn', '$w_id')";
         mysqli_query($con, $sql_enroll);
```

```
        }
        header("Location: index.php");
        exit();
    } else {
        echo "Error: " . $sql_workshop . "<br>" . mysqli_error($con);
    }
}
?>
```

5.2.4. UPDATE

```
<?php
include 'connect.php';
$id = isset($_GET['updateid']) ? $_GET['updateid'] : null;
function fetchWorkshopDetails($con, $id) {
    $details = array();
    $sql = "SELECT * FROM `workshop` WHERE w_id='$id'";
    $result = mysqli_query($con, $sql);
    if (!$result) {
        die("Error: " . mysqli_error($con));
    }
    if (mysqli_num_rows($result) > 0) {
        $details['workshop'] = mysqli_fetch_assoc($result);


        $sql_enrollments = "SELECT * FROM workshop_enroll WHERE w_id='$id'";
        $result_enrollments = mysqli_query($con, $sql_enrollments);
        if ($result_enrollments) {
            $enrollments = array();
            while ($row_enrollment = mysqli_fetch_assoc($result_enrollments)) {
                $enrollments[] = $row_enrollment['USN'];
            }
            $details['enrollments'] = $enrollments;
        }
    } else {
        echo "No workshop found with the provided ID.";
    }
    return $details;
}
$workshopDetails = fetchWorkshopDetails($con, $id);
$workshop = $workshopDetails['workshop'];
$enrollments = isset($workshopDetails['enrollments']) ? $workshopDetails['enrollments'] : array();
if (isset($_POST['update'])) {
    $w_topic = $_POST['w_topic'];
    $w_location = $_POST['w_location'];
    $w_time = $_POST['w_time'];
    $w_date = $_POST['w_date'];
    $w_duration = $_POST['w_duration'];
    $dept_id = $_POST['dept_id'];
    $person_id = $_POST['person_id'];
    $c_id = $_POST['c_id'];
    $enrollments = isset($_POST['usn']) ? $_POST['usn'] : array();

    $person_check_sql = "SELECT * FROM resource_person WHERE person_id='$person_id'";
    $person_check_result = mysqli_query($con, $person_check_sql);
```

```
    if (mysqli_num_rows($person_check_result) == 0) {
        die("Error: Person with ID $person_id does not exist.");
    }
    $sql = "UPDATE `workshop` SET w_topic='$w_topic', w_location='$w_location', w_time='$w_time',
w_date='$w_date', w_duration='$w_duration', dept_id='$dept_id', person_id='$person_id', c_id='$c_id'
WHERE w_id='$id'";
    $result = mysqli_query($con, $sql);
    if ($result) {
        $delete_sql = "DELETE FROM workshop_enroll WHERE w_id='$id'";
        mysqli_query($con, $delete_sql);

        foreach ($enrollments as $usn) {
            $insert_sql = "INSERT INTO workshop_enroll (w_id, USN) VALUES ('$id', '$usn')";
            mysqli_query($con, $insert_sql);
        }
        header("Location: index.php");
        exit();
    } else {
        die(mysqli_error($con));
    }
}
?>
```

5.2.5. DELETION

```
<?php
include 'connect.php';
if(isset($_GET['deleteid'])){
    $ws_id=$_GET['deleteid'];
    $sql="delete from workshop where w_id='$w_id'";
    $result=mysqli_query($con,$sql);
    if($result){
       header('location:index.php');
    }
    else{
        die(mysqli_error($con));
    }
}
?>
```

5.2.6. SORTING

```
<?php
$sort = isset($_GET['sort']) ? $_GET['sort'] : 'latest';
$sortOptions = ['latest', 'oldest'];
if (!in_array($sort, $sortOptions)) {
    $sort = 'latest';
}
?>
```

5.2.7. SEARCHING

```
<?php
$search = isset($_GET['search']) ? $_GET['search'] : '';
$searchQuery = $search ? "WHERE w.w_topic LIKE '%$search%'" : '';
?>
```

CHAPTER 6

RESULT

6.1 HOME PAGE

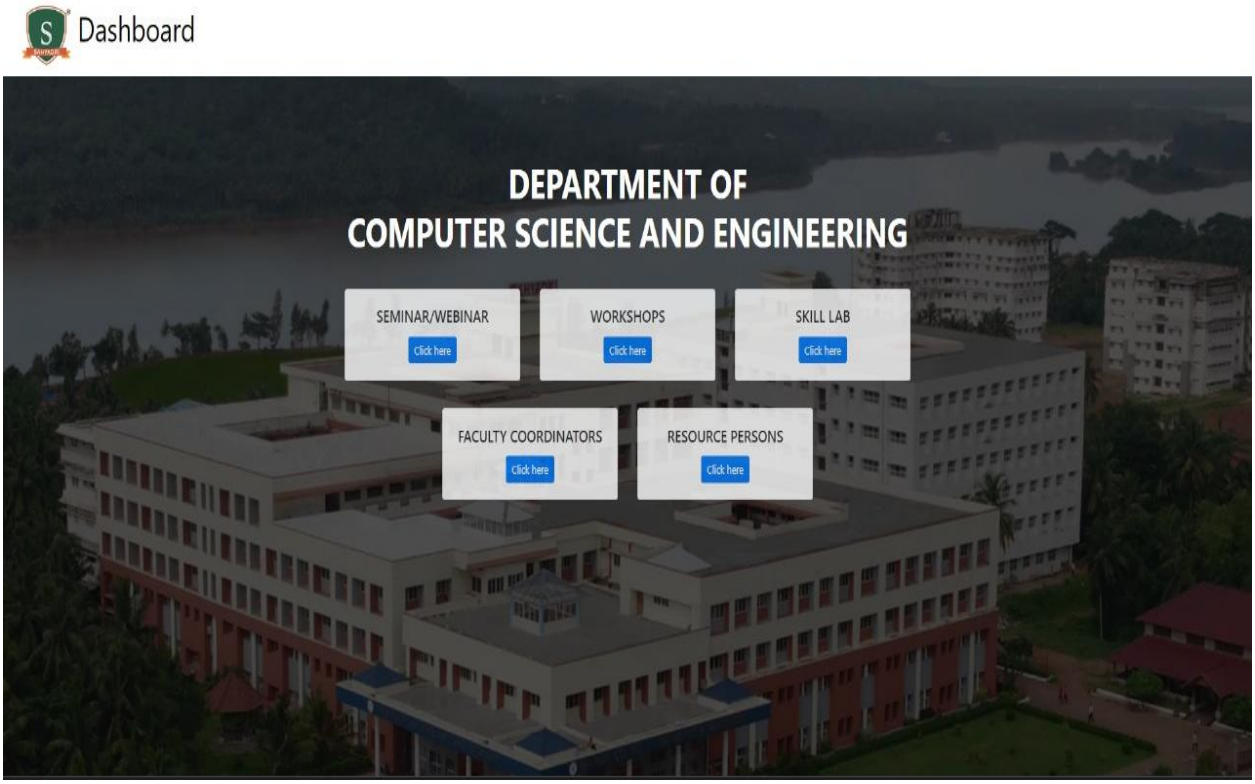


Fig 6.1 Home Page

6.2 WORKSHOP VIEW PAGE

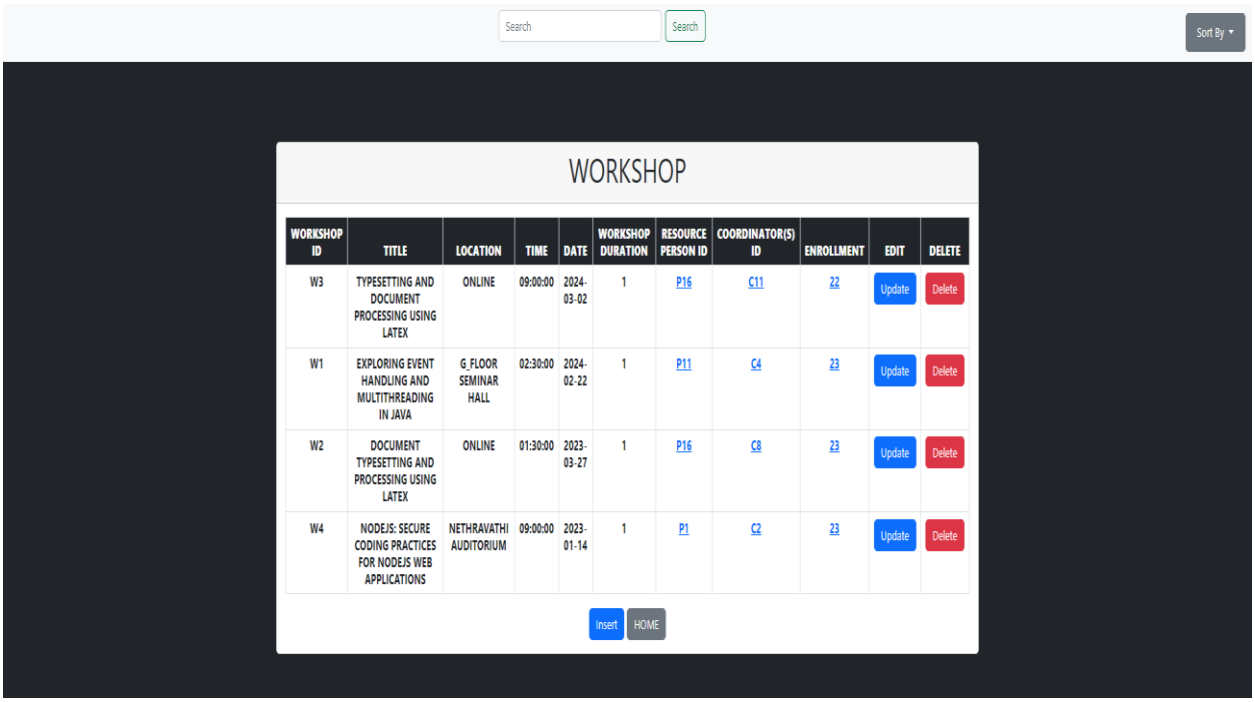


Fig 6.2 Workshop View Page

6.3 WORKSHOP SEARCH PAGE

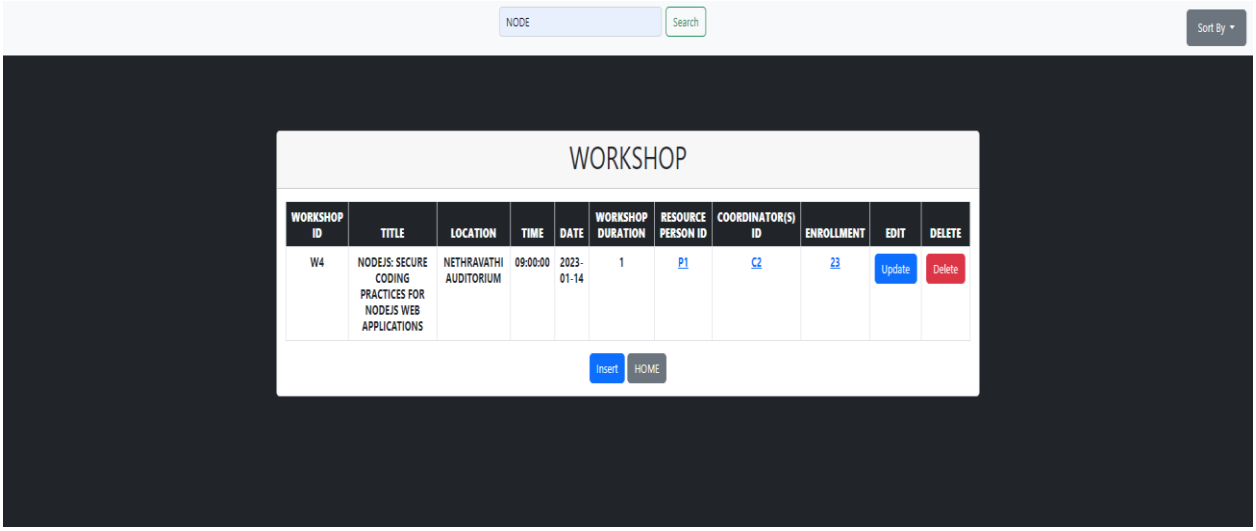


Fig 6.3 Workshop Search Page

6.4 WORKSHOP INSERT PAGE

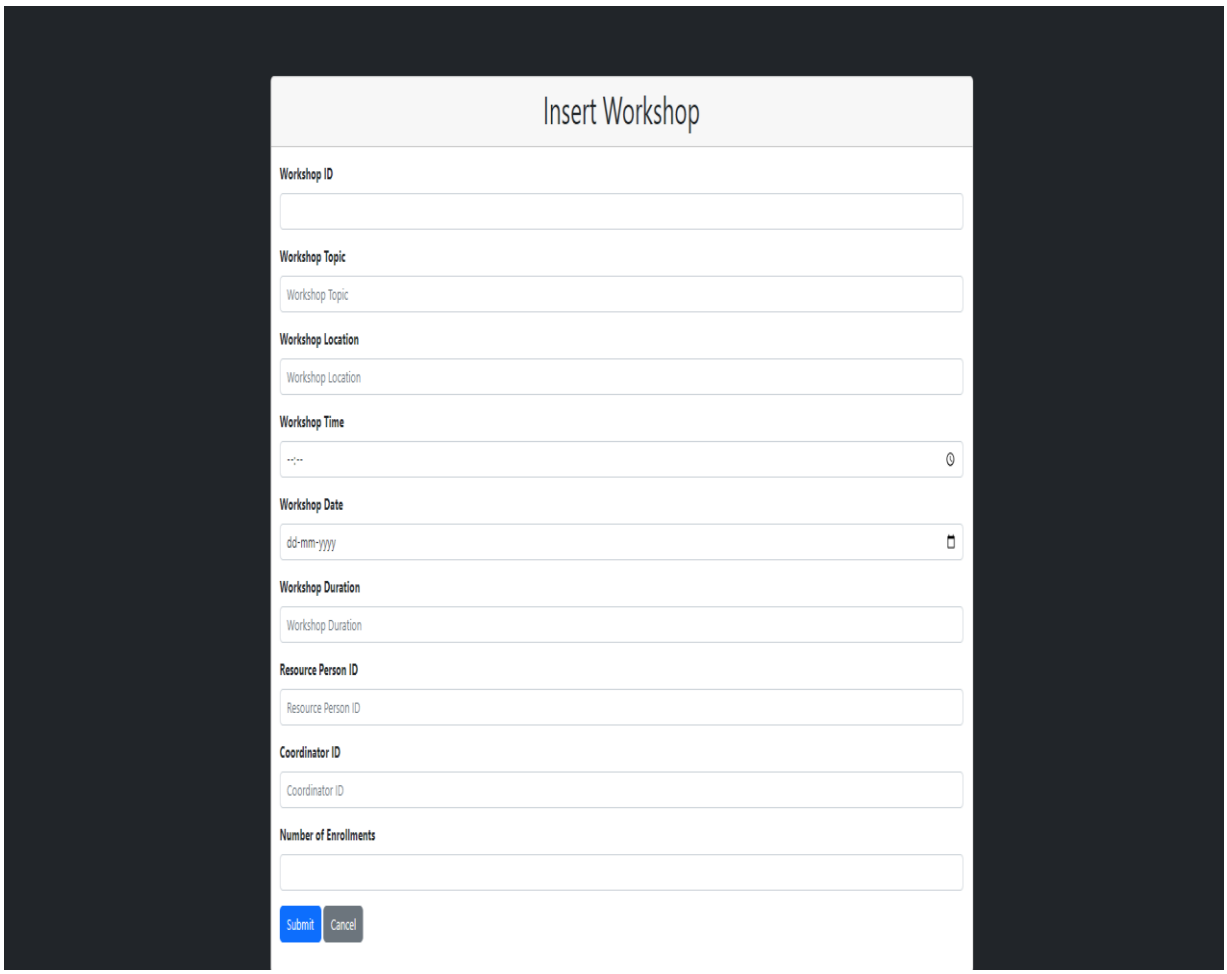


Fig 6.4 Workshop Insert Page

6.5 WORKSHOP UPDATE PAGE

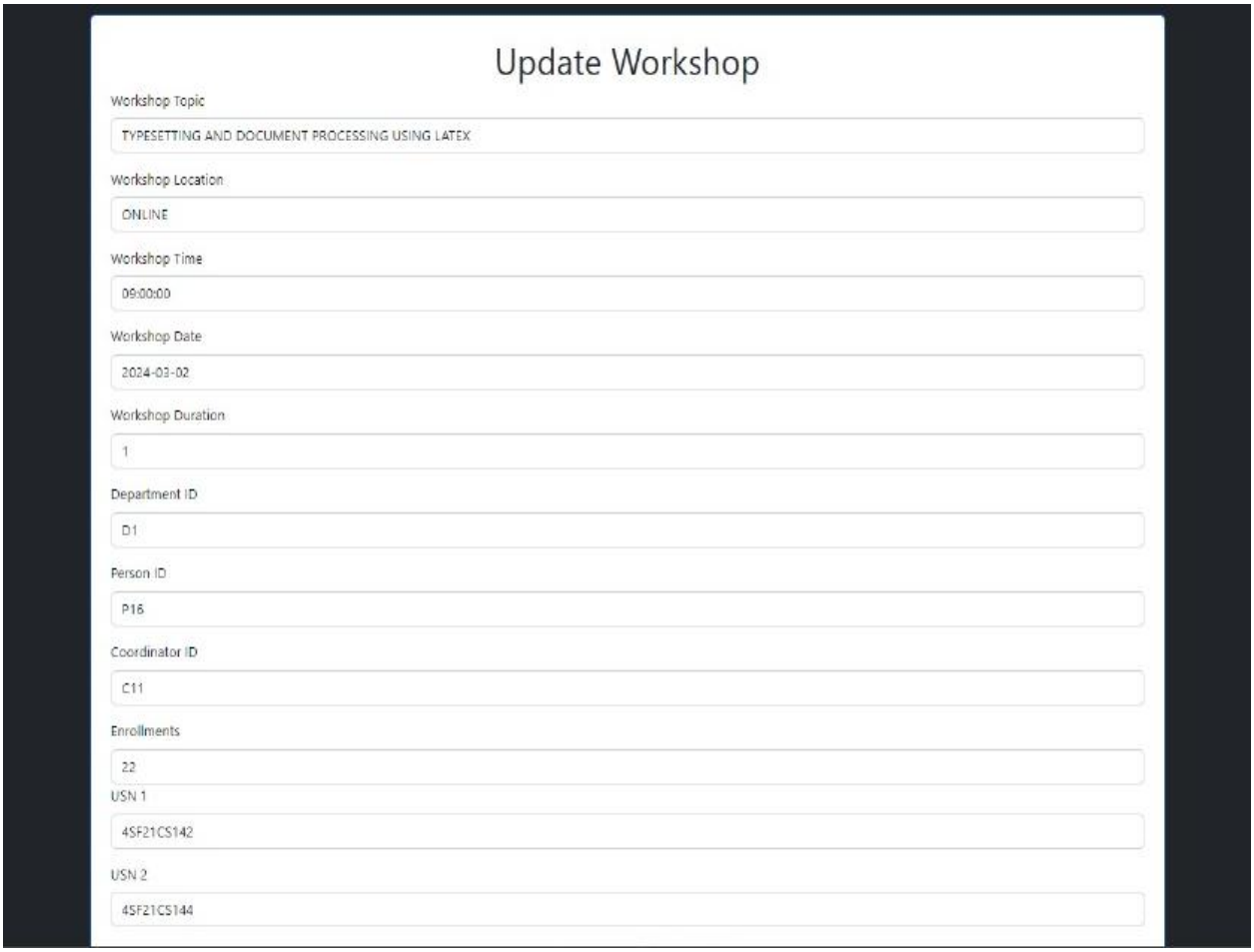


Fig 6.5 Workshop Update Page

6.6 RESOURCE PERSON DETAILS PAGE

Fig 6.6 Resource Person Details Page

6.7 COORDINATOR DETAILS PAGE

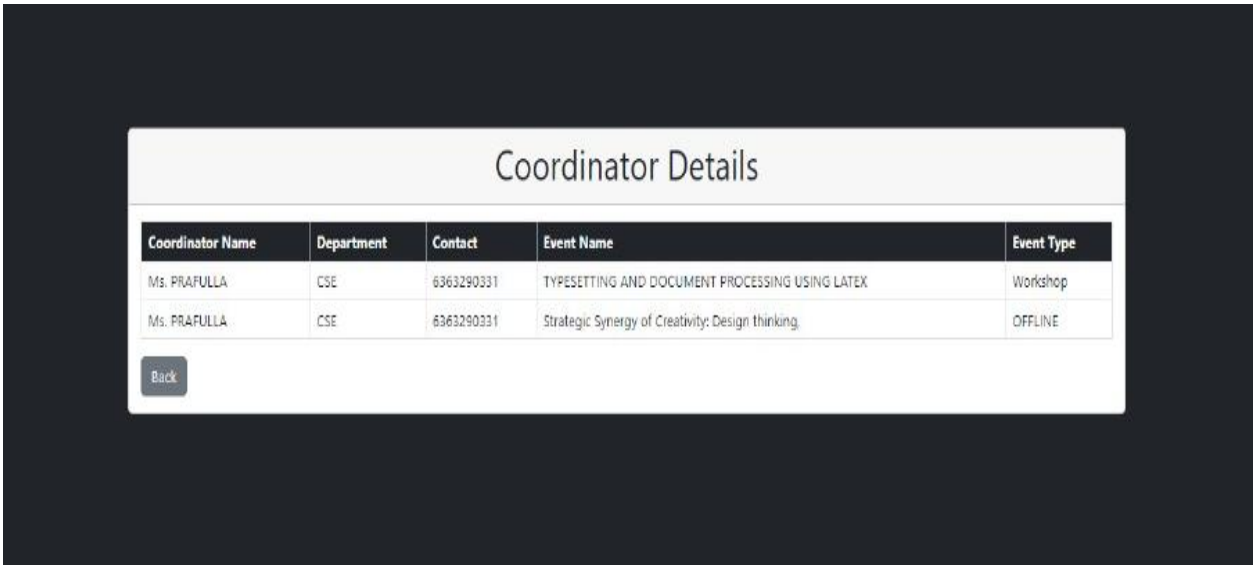


Fig 6.7 Coordinator Details Page

6.8 ENROLLMENT DETAILS PAGE

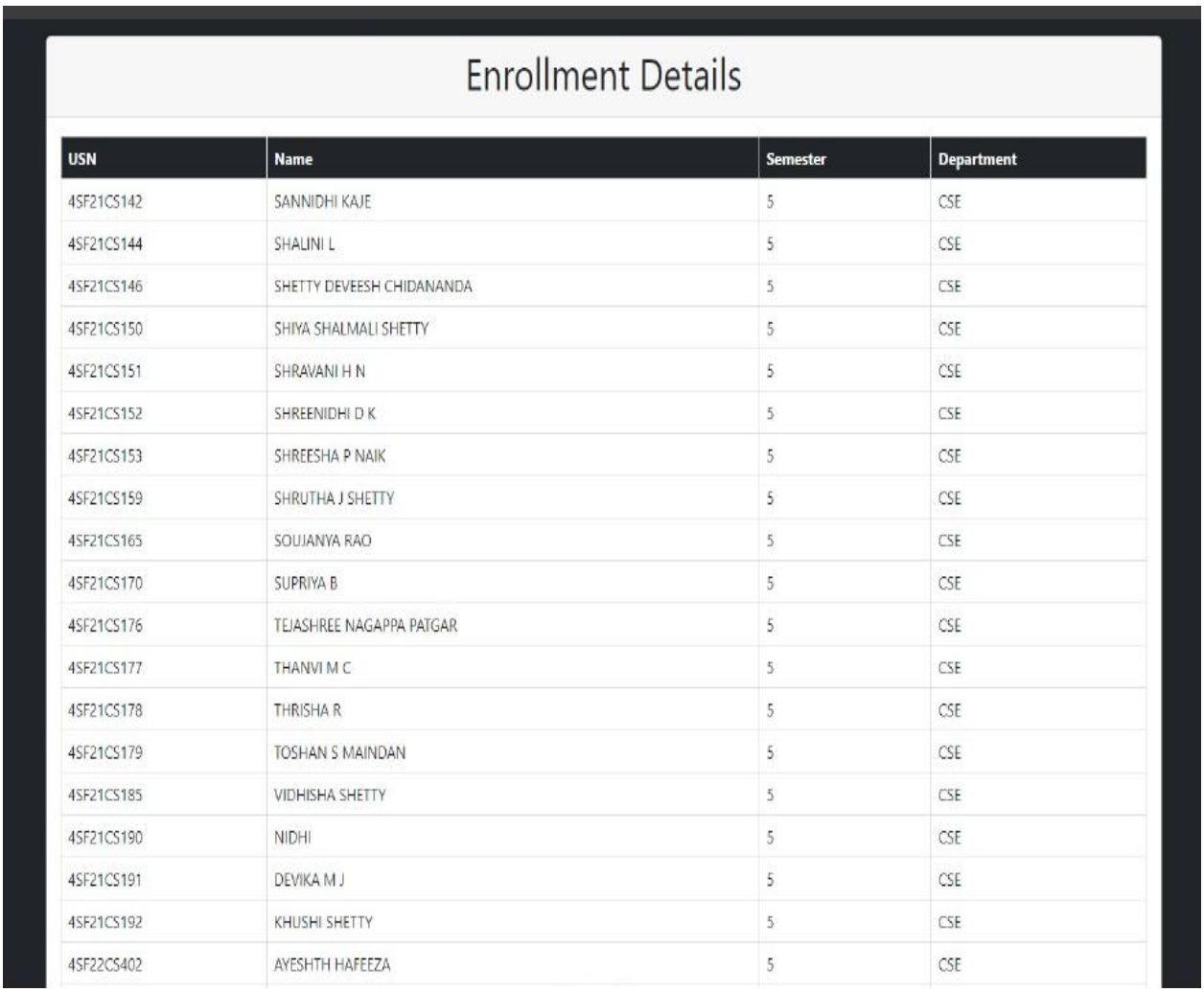


Fig 6.8 Enrollment Details Page

CHAPTER 7

CONCLUSION

It has been an enriching journey filled with challenges and rewards to embark on and successfully complete this project. In conclusion, the Department Resource and Activity Management System serves as a pivotal tool for efficiently organizing and overseeing various departmental activities. The incorporation of a user-friendly graphical interface ensures seamless navigation and data input, facilitating effortless database management for administrators.

This project stands as a comprehensive solution for optimizing the operations of the department, enhancing overall efficiency and productivity. A notable advantage lies in the system's capability to effortlessly monitor activities, resource utilization, and outcomes. Administrators can swiftly access information regarding activities, past engagements.

By emphasizing the importance of efficient resource management, this project aligns with broader initiatives aimed at enhancing departmental effectiveness and supporting organizational goals. It promotes the efficient utilization of departmental resources.

REFERENCES

1. "DEPARTMENT MANAGEMENT SYSTEM" Jhanvi Agarwal, Renuka Singh, Mansi Singh, Mansi Raghav, Student IMS Engineering College, Ghaziabad, India https://www.researchgate.net/publication/346208202_DEPARTMENT_MANAGEMENT_SYSTEM_-_WEB-BASED_APPLICATION