

**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 PRITHVI P SHETTY**

4SF21CS032 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 No.** | **Topic** | **Page No.** |
|  | Introduction | 6-7 |
|  | Requirement Specification | 8 |
| 3.1  3.2  3.4 | Design  E-R diagram  Relational Schema (ER to relational schema)  Schema diagram | 9-10  11-14  15 |
|  | Normalization | 16-17 |
|  | Implementation | 18-25 |
|  | Results | 26-29 |
|  | Conclusion | 30 |
|  | References | 31 |

**CHAPTER 1**

**INTRODUCTION**

* 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. **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. **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. **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**

* 1. **Hardware Requirements:**
* Processor: Any processor above 500 MHz
* RAM: 4GB
* Hard Disk: 500GB
* Input Device: Standard keyboard and Mouse
* Output Device: Monitor
  1. **Software Requirements:**
* Database: MySQL (phpMyAdmin)
* Programming Language: php
* IDE: Visual Studio Code (VSCode)

**CHAPTER 3**

**DESIGN**

* 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.

* + 1. **Symbols in Entity Relationship:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Attribute |  | Relationship |
|  | Entity |  | Derived Attribute |
|  | Multivalued Attribute |  | Weak Entity |
|  | Weak Relationship |  | Key Attribute |

**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.

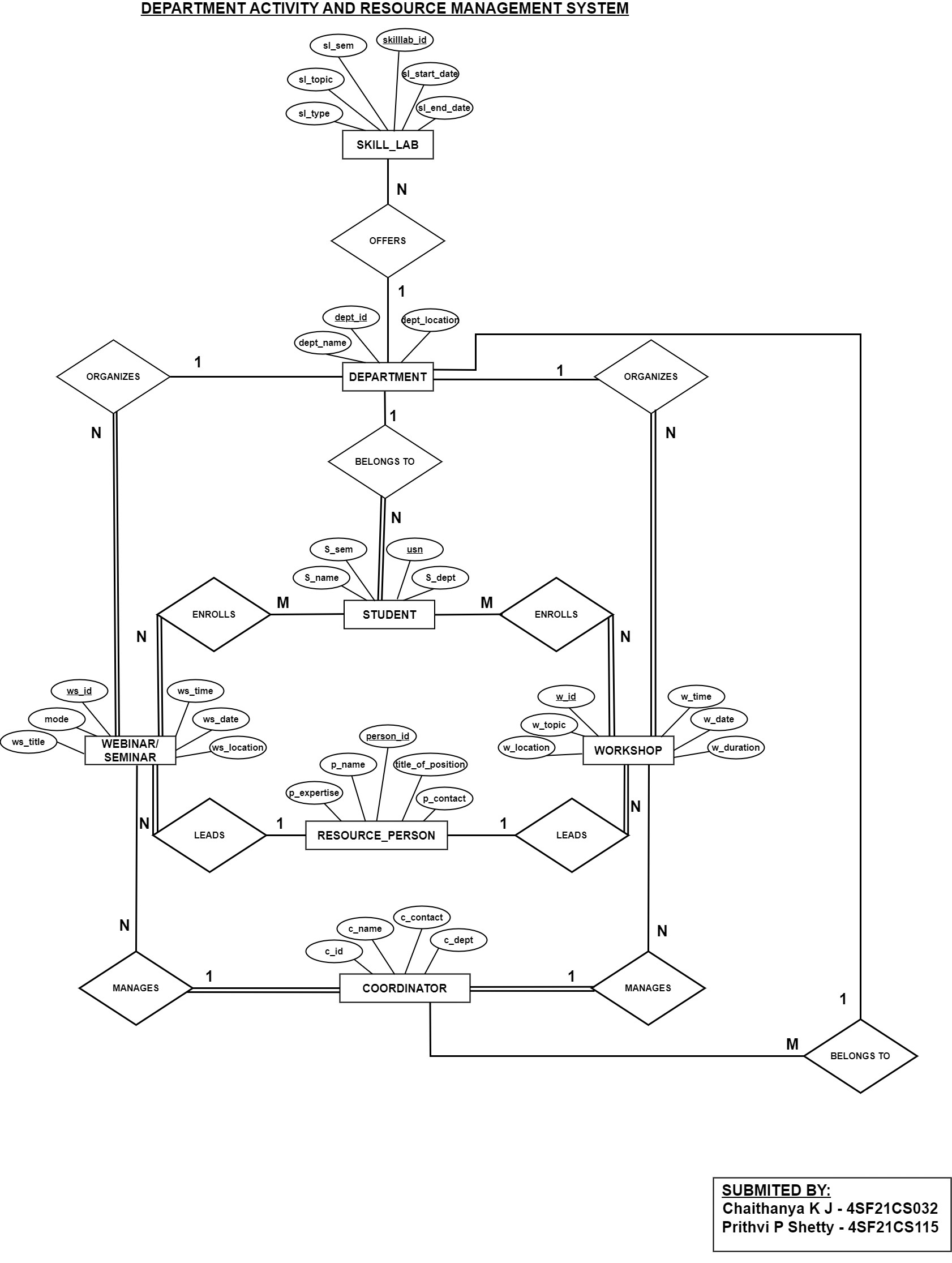
**Multivalued Attribute:** Can hold multiple values for a single entity instance.

**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.

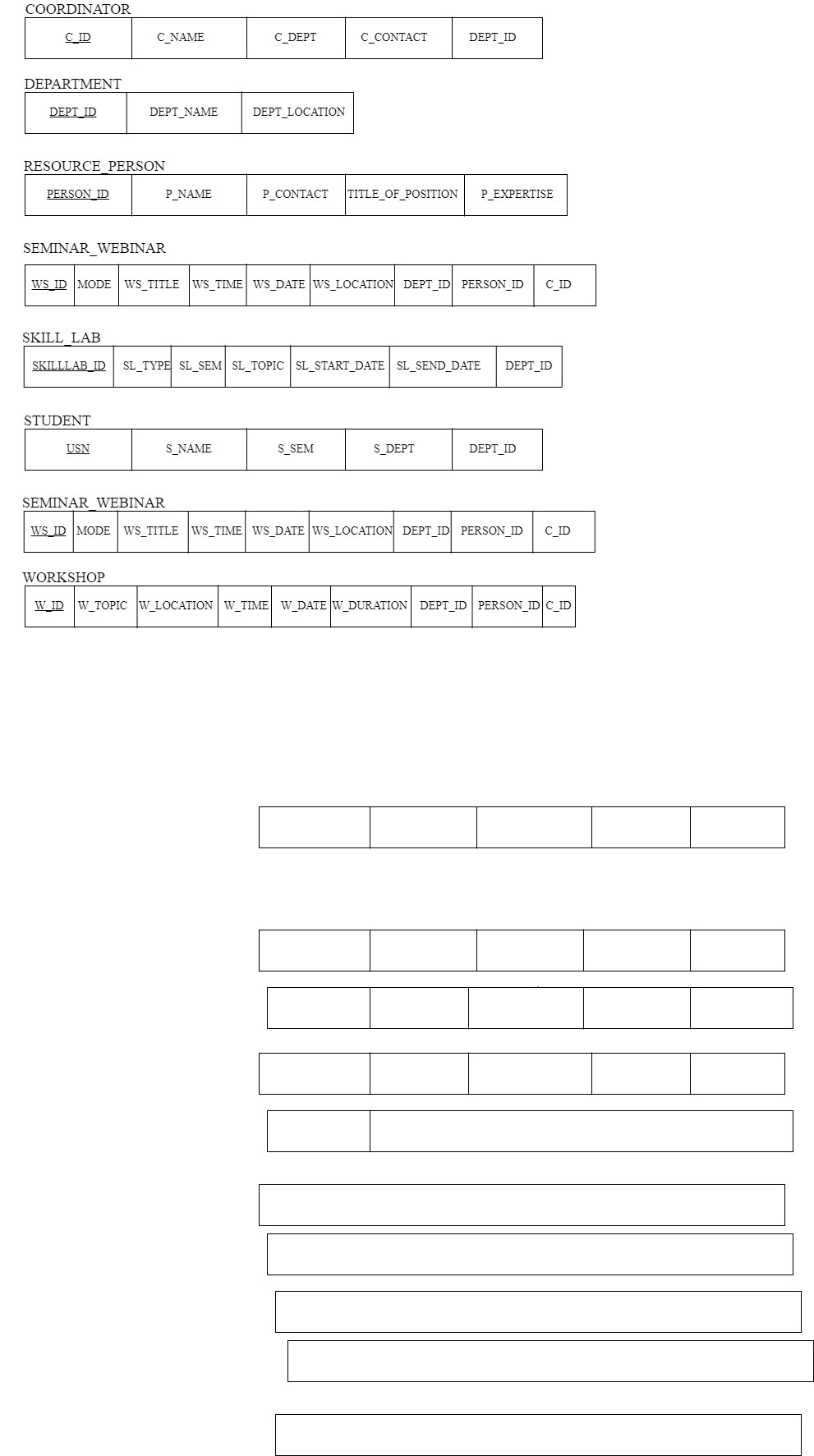
* + 1. **ER diagram of Department Activites and Resources Management System**



**Figure 3.1.2 ER diagram of Department Activity and Resources Management System**

* 1. **Relational Schema**
     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.



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

* + 1. **Mapping of Weak Entity Types**

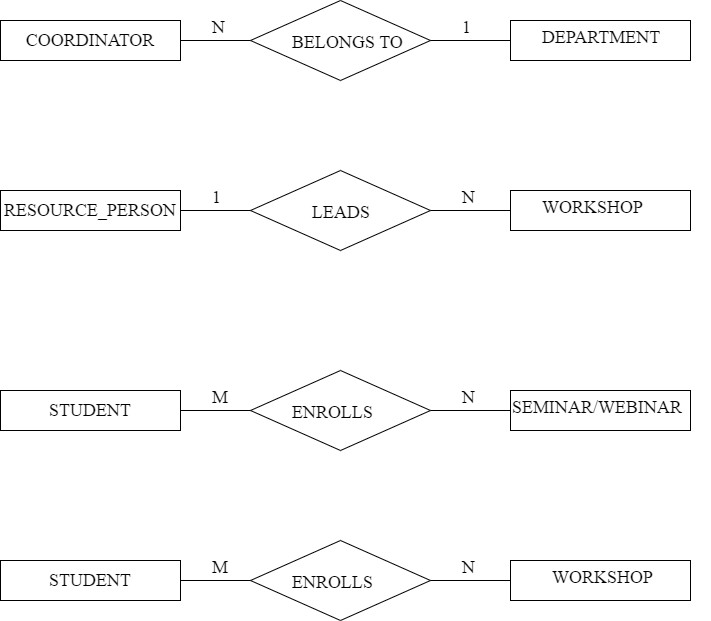
The ER Diagram contains no weak entities, this step is ignored in this project.

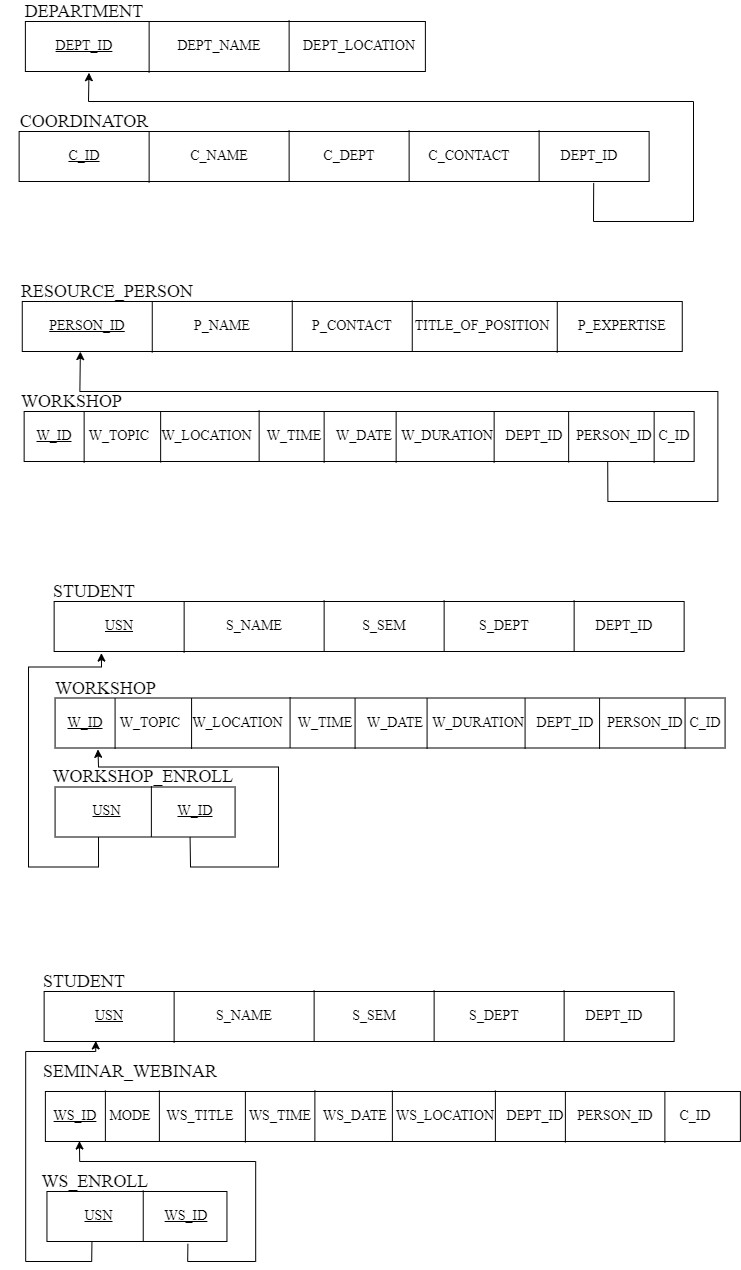
* + 1. **Mapping of Binary 1:1 Relationship Types**

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

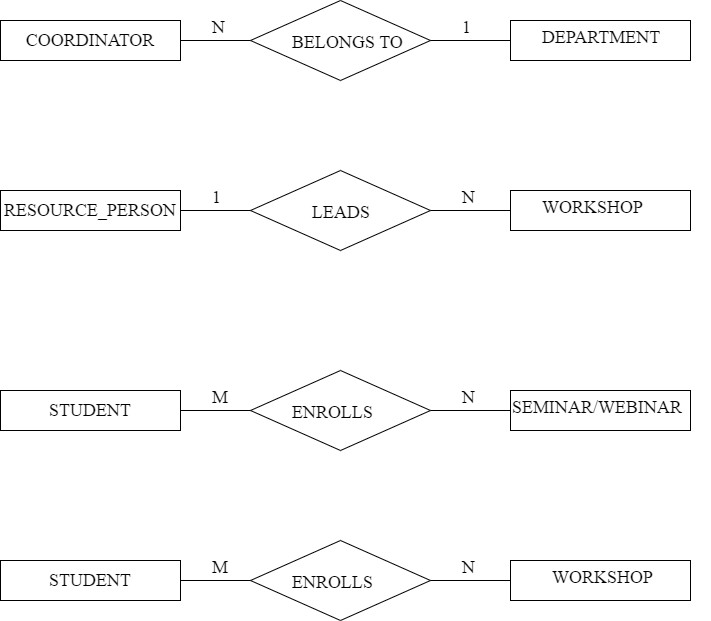
* + 1. **Mapping of Binary 1: N Relationship Types**

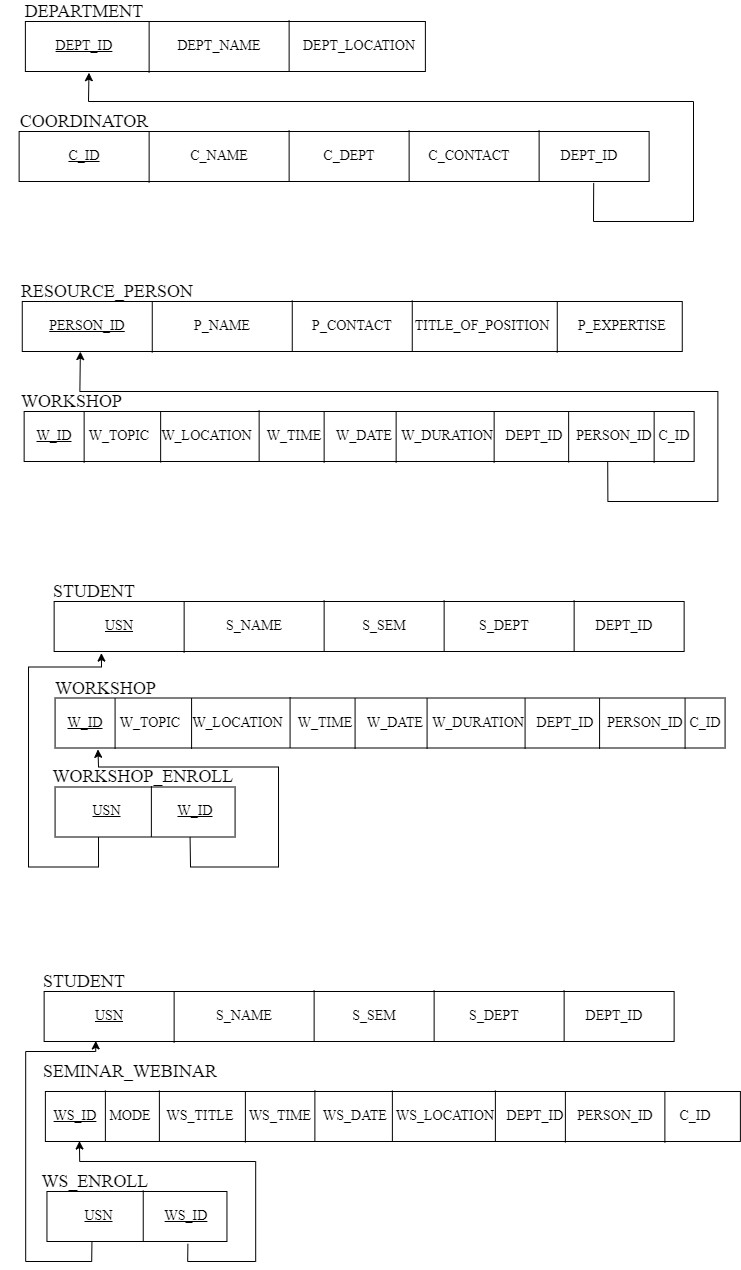
There are seven 1: N relation in schema diagram. There is a foreign key approach and have included the primary key of one relation as the foreign key in the N-sided relation.





**Fig 3.2.2 1:N relationship between Department and Coordinator.**

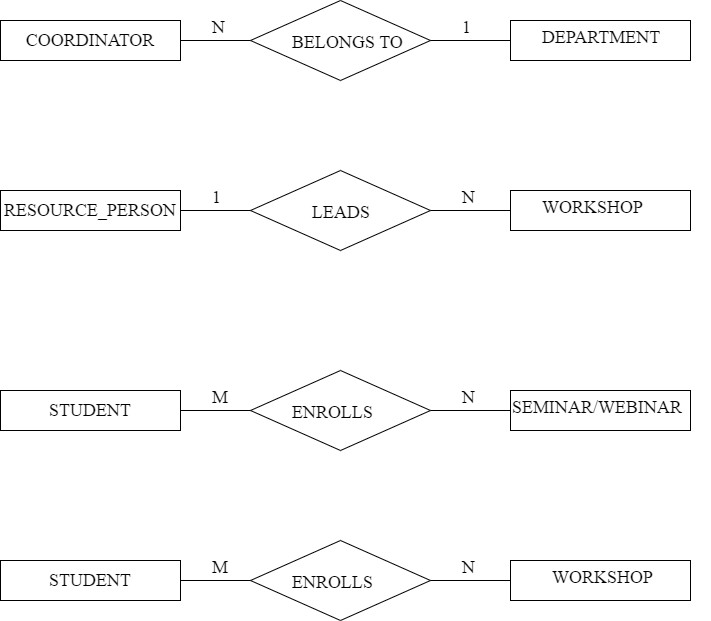


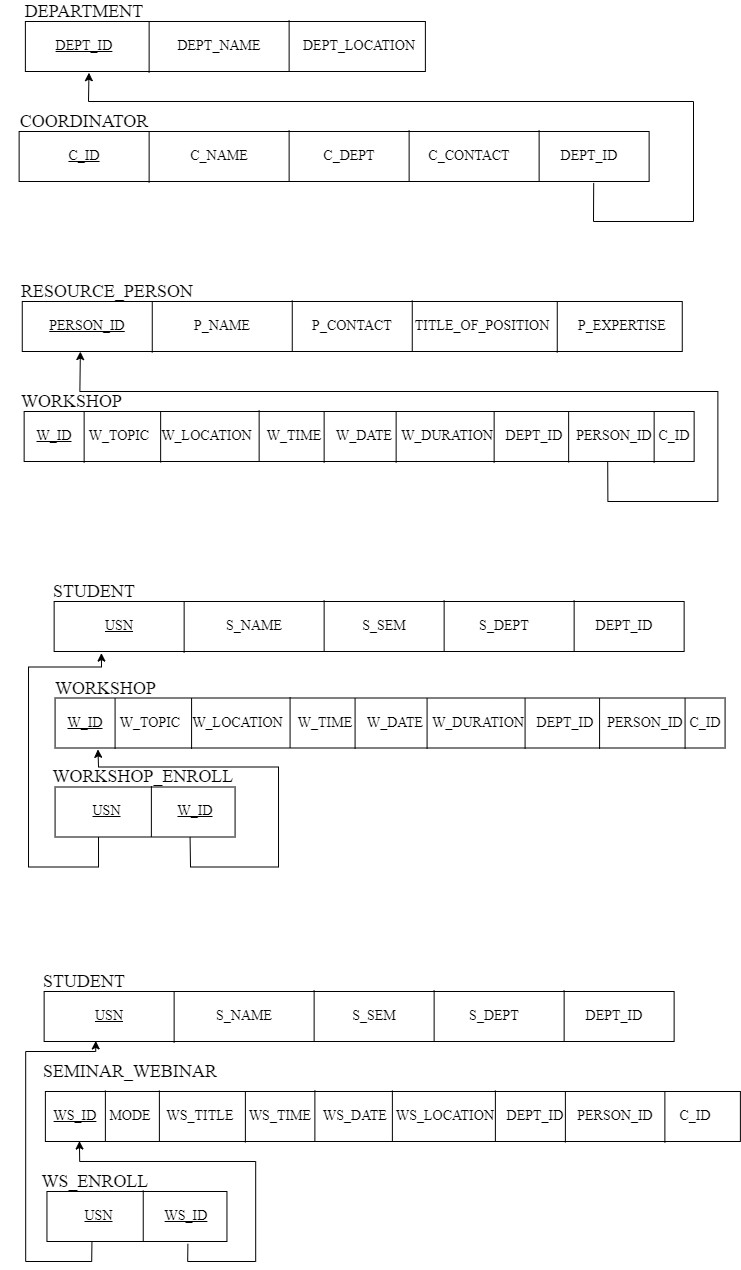


**Fig 3.2.3 1:N relationship between Resource\_person and Workshop.**

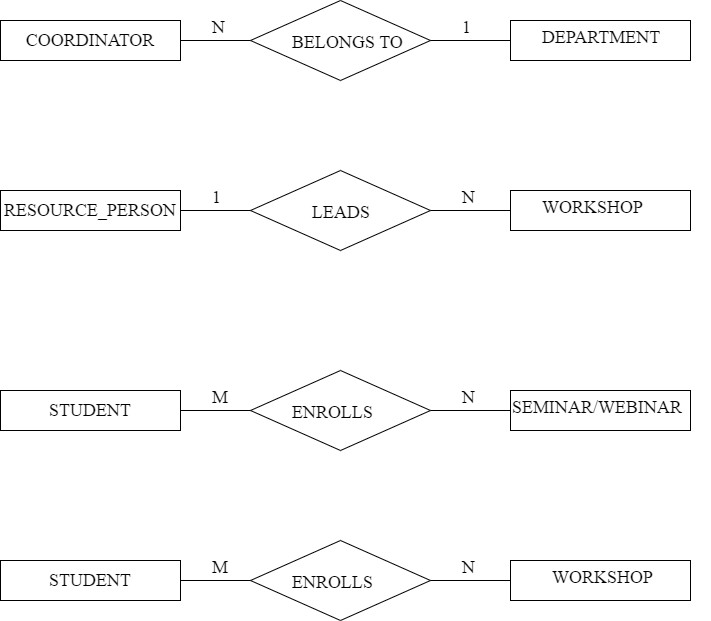
* + 1. **Mapping of Binary M: N Relationship Types**

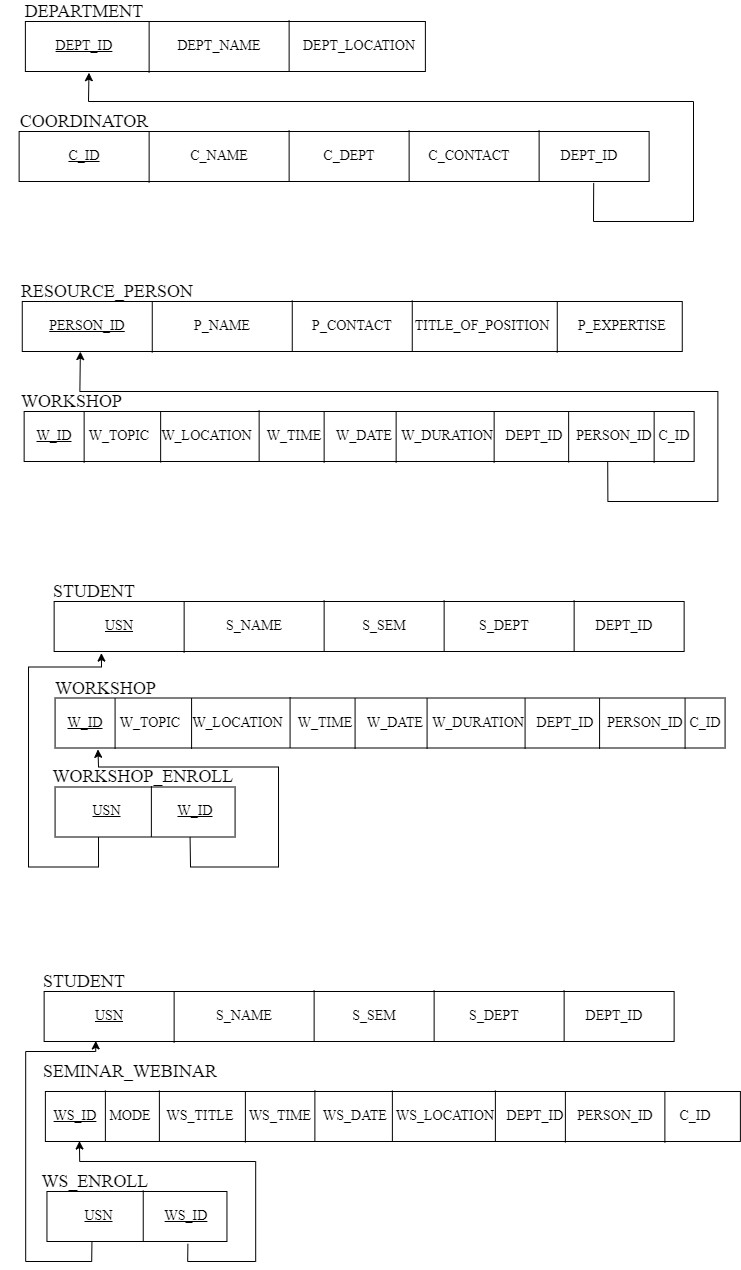
There are two M: N relation in schema diagram. Mapping of Binary M: N Relationship Types involves the introduction of a junction table to manage the many-to-many relationship between two entities. This junction table contains foreign keys referencing the primary keys of the entities involved, facilitating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

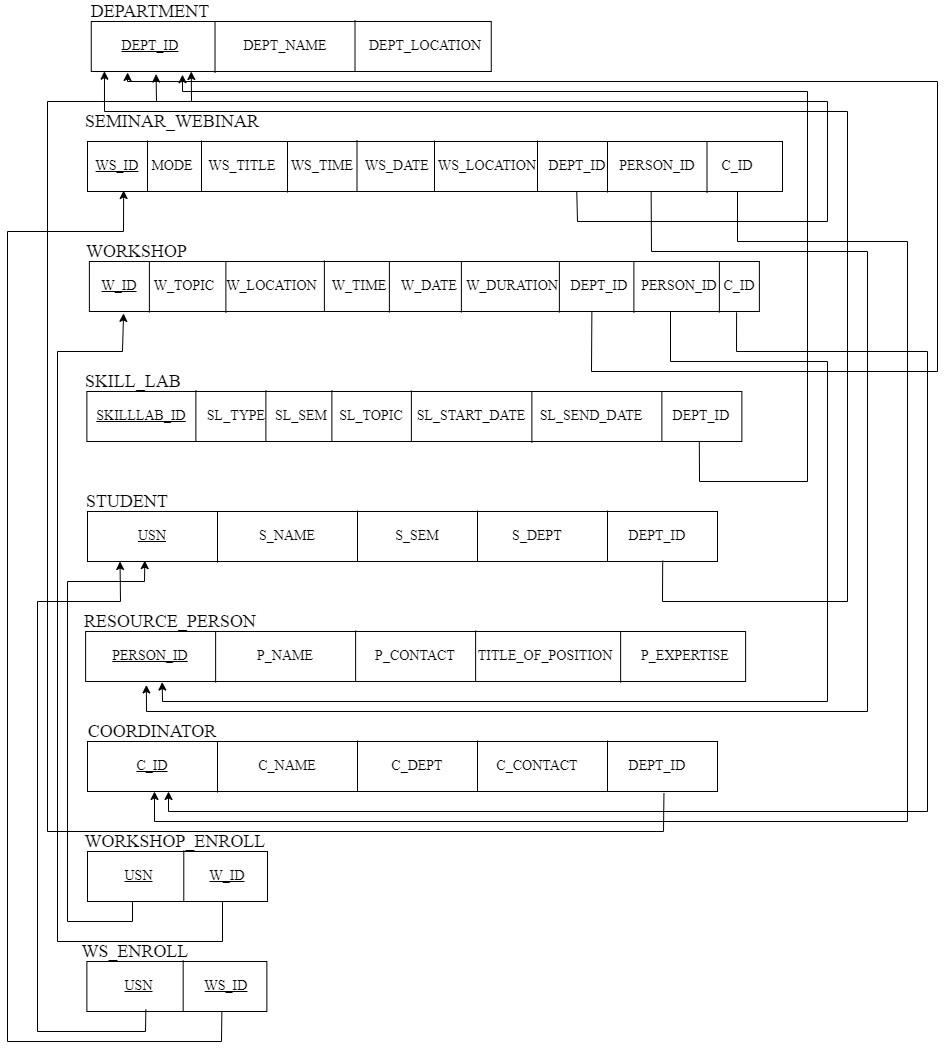
* + 1. **Mapping of Multivalued Attributes**

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

* + 1. **Mapping of N-ary Relationship Types**

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

* 1. **Schema diagram**

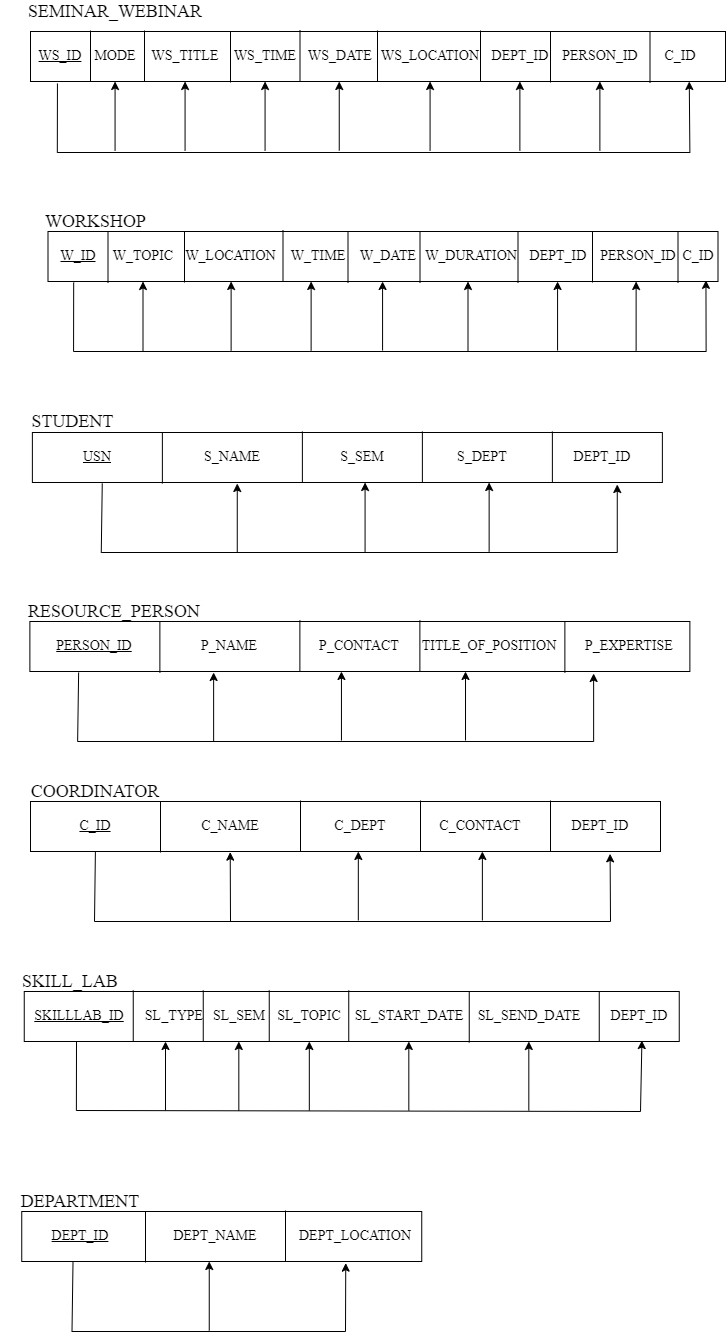


**Fig 3.3. Schema Diagram.**

**CHAPTER 4**

**NORMALIZATION**

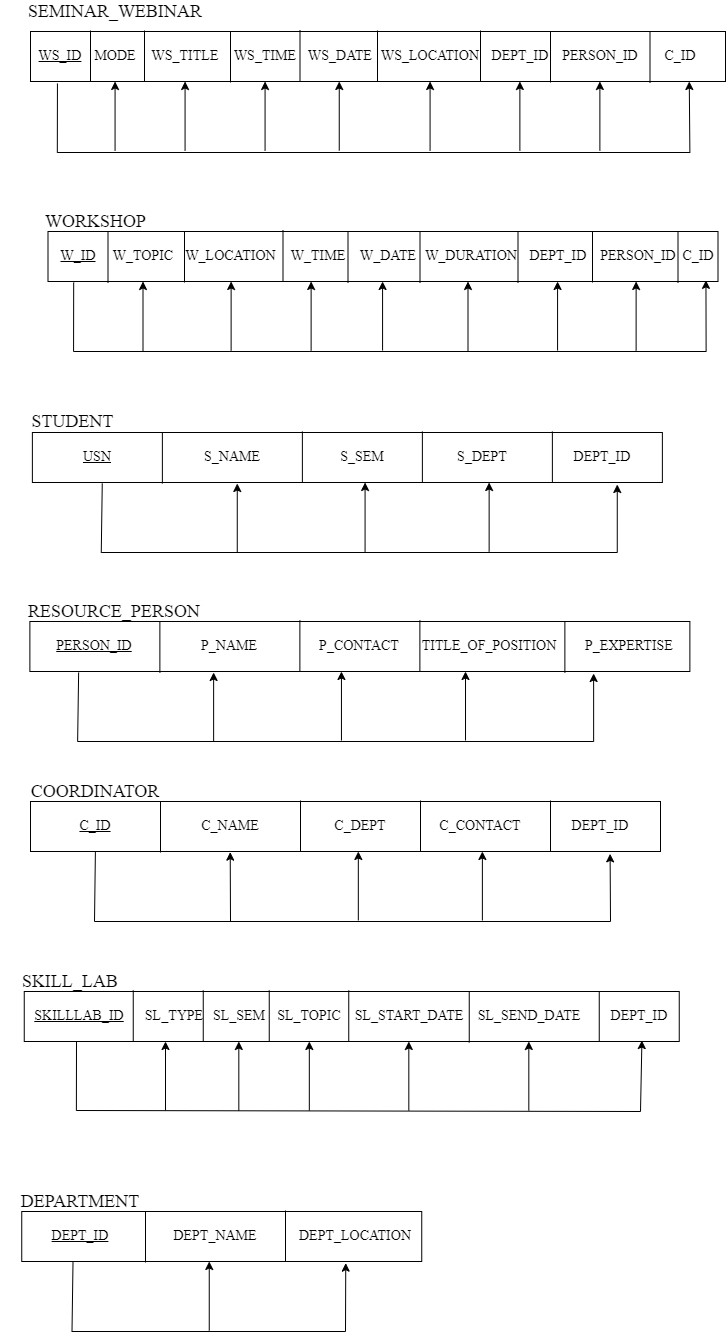
**SEMINAR\_WEBINAR**



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.

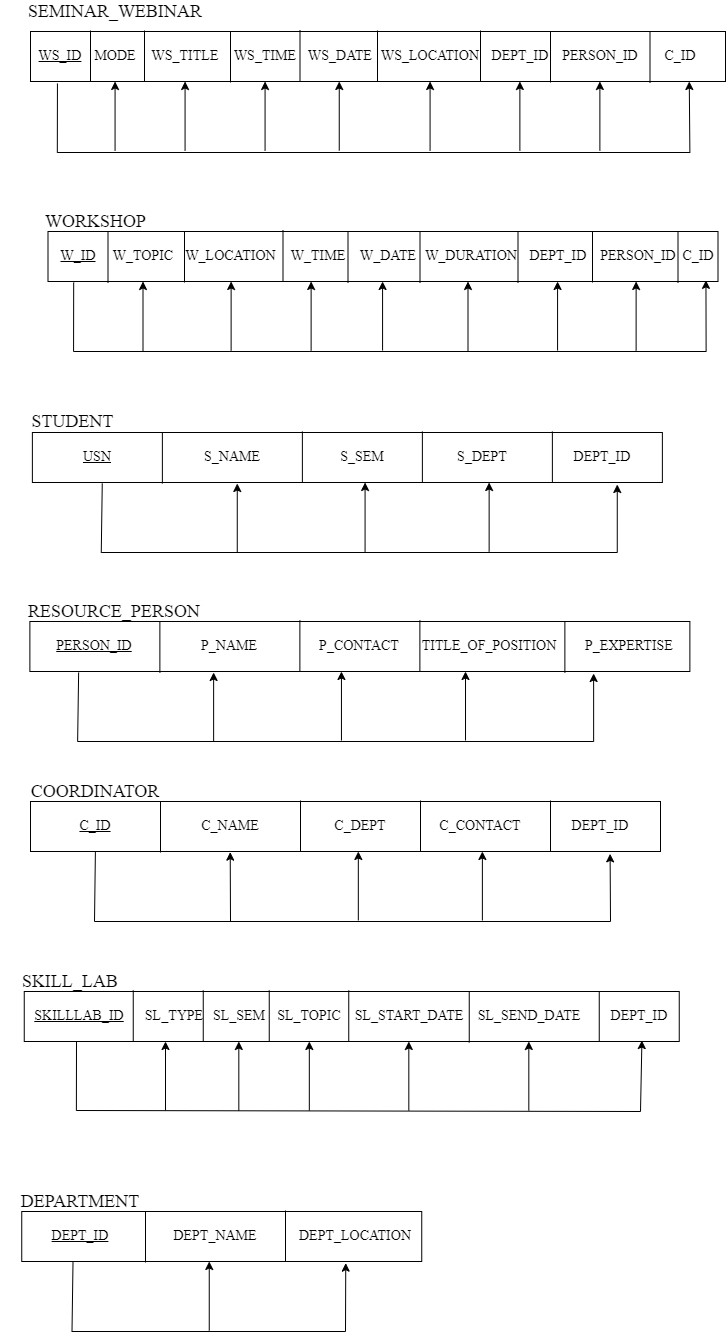
**WORKSHOP**



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**

* 1. **TABLE STRUCTURE**
     1. **DEPARTMENT RELATION**

CREATE TABLE `department` (

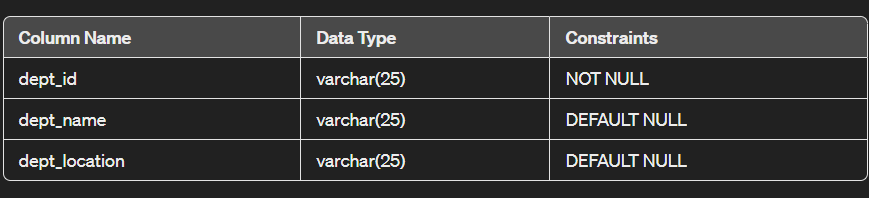
  `dept\_id` varchar(25) NOT NULL,

  `dept\_name` varchar(25) DEFAULT NULL,

  `dept\_location` varchar(25) DEFAULT NULL

PRIMARY KEY(dept\_id)

);



**Fig 5.1.1 Department Relation**

* + 1. **RESOURCE\_PERSON RELATION**

CREATE TABLE `resource\_person` (

  `person\_id` varchar(25) NOT NULL,

  `p\_name` varchar(25) DEFAULT NULL,

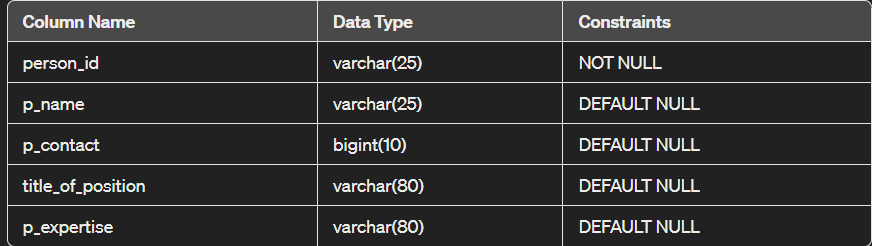
  `p\_contact` bigint(10) DEFAULT NULL,

  `title\_of\_position` varchar(80) DEFAULT NULL,

  `p\_expertise` varchar(80) DEFAULT NULL

PRIMARY KEY(person\_id)

);



**Fig 5.1.2 Resource\_person Relation**

* + 1. **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,

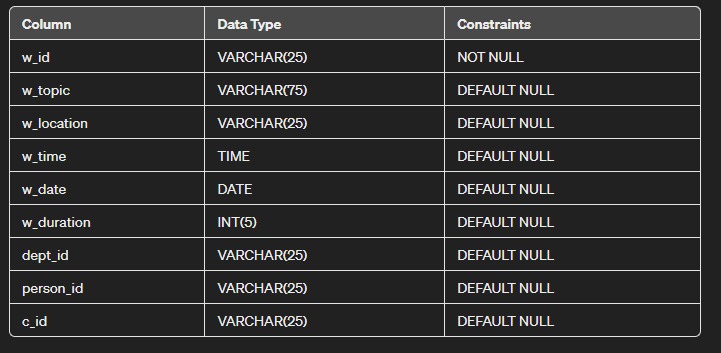
  `dept\_id` varchar(25) DEFAULT NULL references department(dept\_id),

  `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)

);



**Fig 5.1.3 Workshop Relation**

* + 1. **WORKSHOP\_ENROLL RELATION**

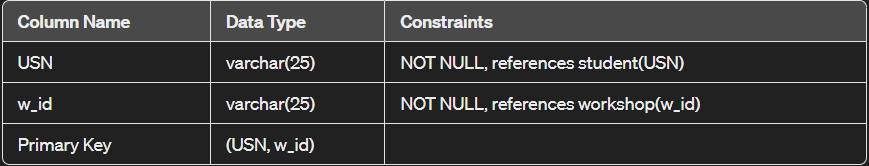
CREATE TABLE `workshop\_enroll` (

  `USN` varchar(25) NOT NULL references student(USN),

  `w\_id` varchar(25) NOT NULL references workshop(w\_id),

PRIMARY KEY (USN, w\_id)

) ;



**Fig 5.1.4 workshop\_enroll Relation**

* + 1. **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,

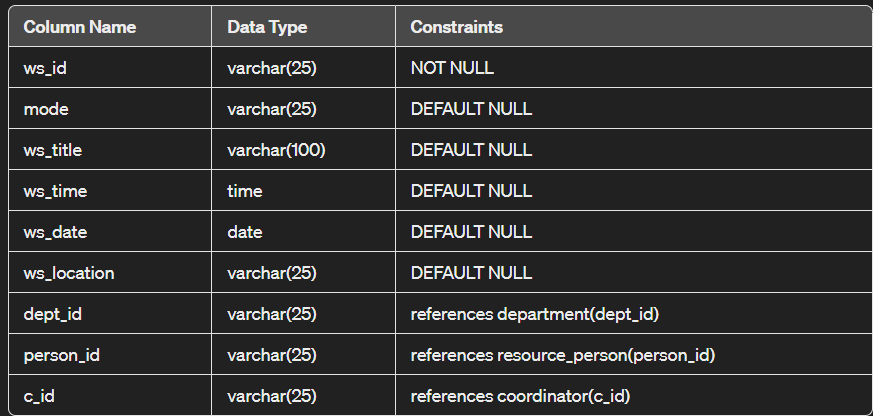
  `dept\_id` varchar(25) DEFAULT NULL references department(dept\_id),

  `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)

);



**Fig 5.1.5 Seminar\_Webinar Relation**

* + 1. **WS\_ENROLL RELATION**

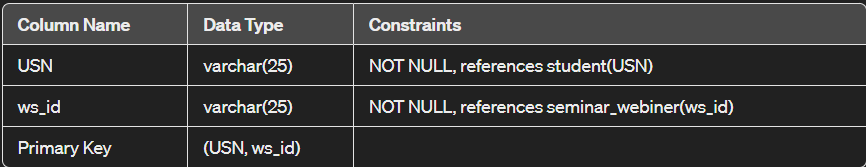
CREATE TABLE `ws\_enroll` (

  `USN` varchar(25) NOT NULL references student(USN),

  `ws\_id` varchar(25) NOT NULL references seminar\_webiner(ws\_id),

PRIMARY KEY (USN, ws\_id)

) ;



**Fig 5.1.6 ws\_enroll Relation**

* + 1. **COORDINATOR RELATION**

CREATE TABLE `coordinator` (

  `c\_id` varchar(25) NOT NULL,

  `c\_name` varchar(25) DEFAULT NULL,

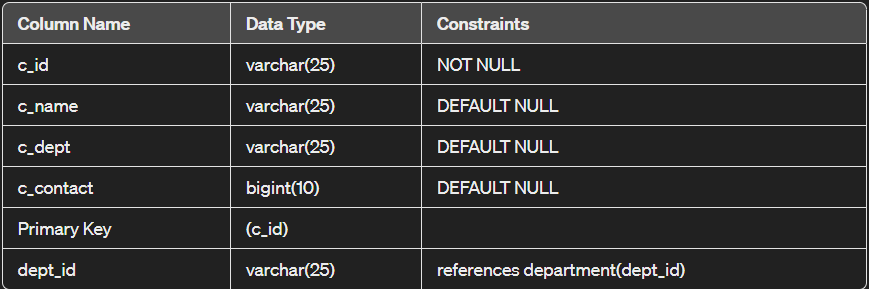
  `c\_dept` varchar(25) DEFAULT NULL,

  `c\_contact` bigint(10) DEFAULT NULL,

PRIMARY KEY (c\_id),

  `dept\_id` varchar(25) DEFAULT NULL references department(dept\_id)

);



**Fig 5.1.7 Coordinator Relation**

* + 1. **STUDENT RELATION**

CREATE TABLE `student` (

  `USN` varchar(25) NOT NULL,

  `s\_name` varchar(25) DEFAULT NULL,

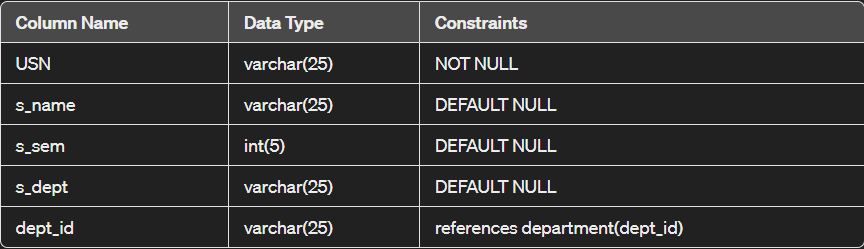
  `s\_sem` int(5) DEFAULT NULL,

  `s\_dept` varchar(25) DEFAULT NULL,

  `dept\_id` varchar(25) DEFAULT NULL references department(dept\_id),

PRIMARY KEY(USN)

);



**Fig 5.1.8 Student Relation**

* + 1. **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,

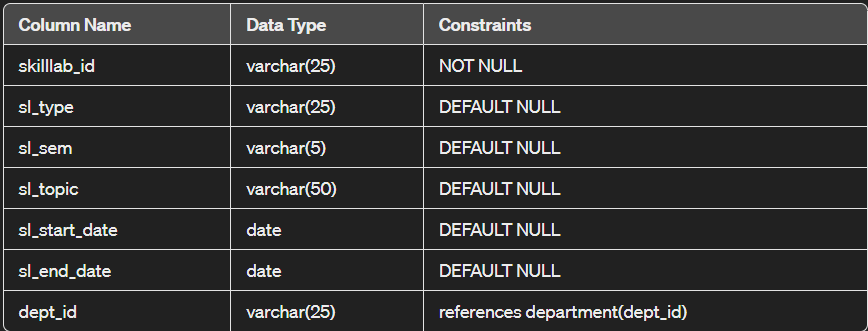
  `sl\_start\_date` date DEFAULT NULL,

  `sl\_end\_date` date DEFAULT NULL,

  `dept\_id` varchar(25) DEFAULT NULL references department(dept\_id),

PRIMARY KEY(skilllab\_idtta )

);



**Fig 5.1.9 skill\_lab Relation**

* 1. **FUNCTIONALITY**
     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));

}

?>

* + 1. **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);

?>

* + 1. **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);

    }

}

?>

* + 1. **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));

    }

}

?>

* + 1. **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));

    }

}

?>

* + 1. **SORTING**

<?php

$sort = isset($\_GET['sort']) ? $\_GET['sort'] : 'latest';

$sortOptions = ['latest', 'oldest'];

if (!in\_array($sort, $sortOptions)) {

    $sort = 'latest';

}

?>

* + 1. **SEARCHING**

<?php

$search = isset($\_GET['search']) ? $\_GET['search'] : '';

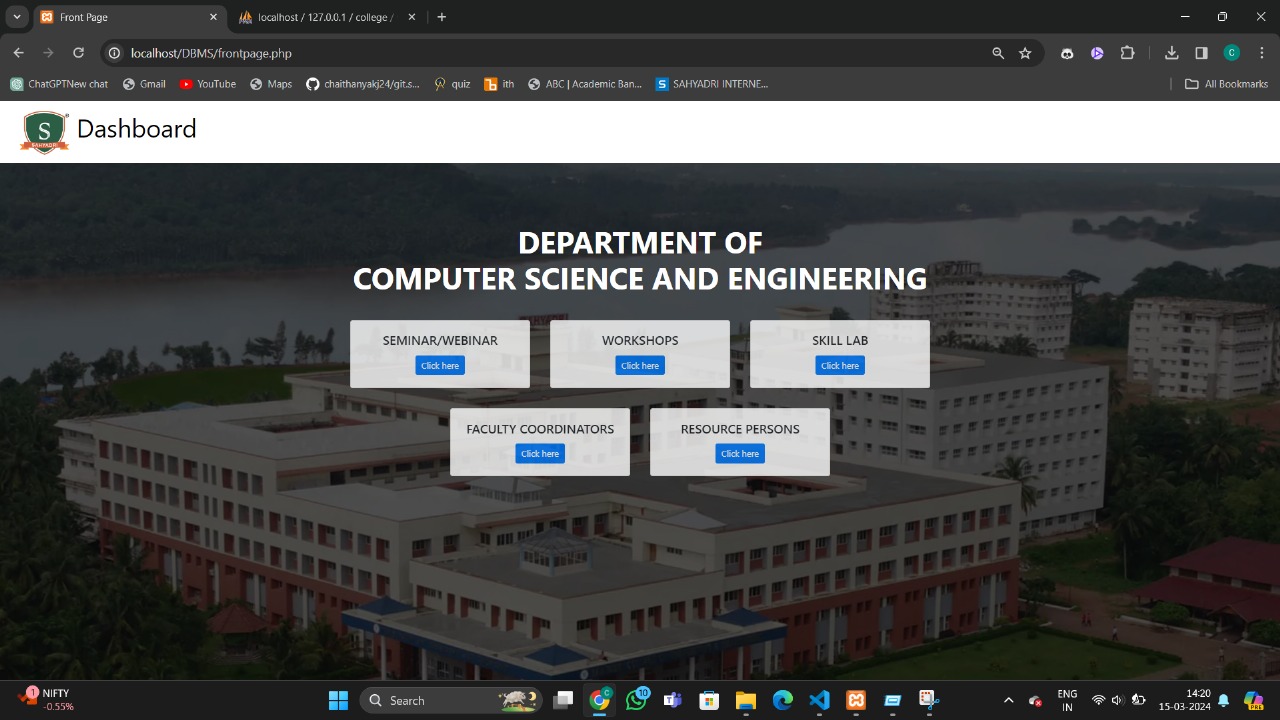
$searchQuery = $search ? "WHERE w.w\_topic LIKE '%$search%'" : '';

?>

**CHAPTER 6**

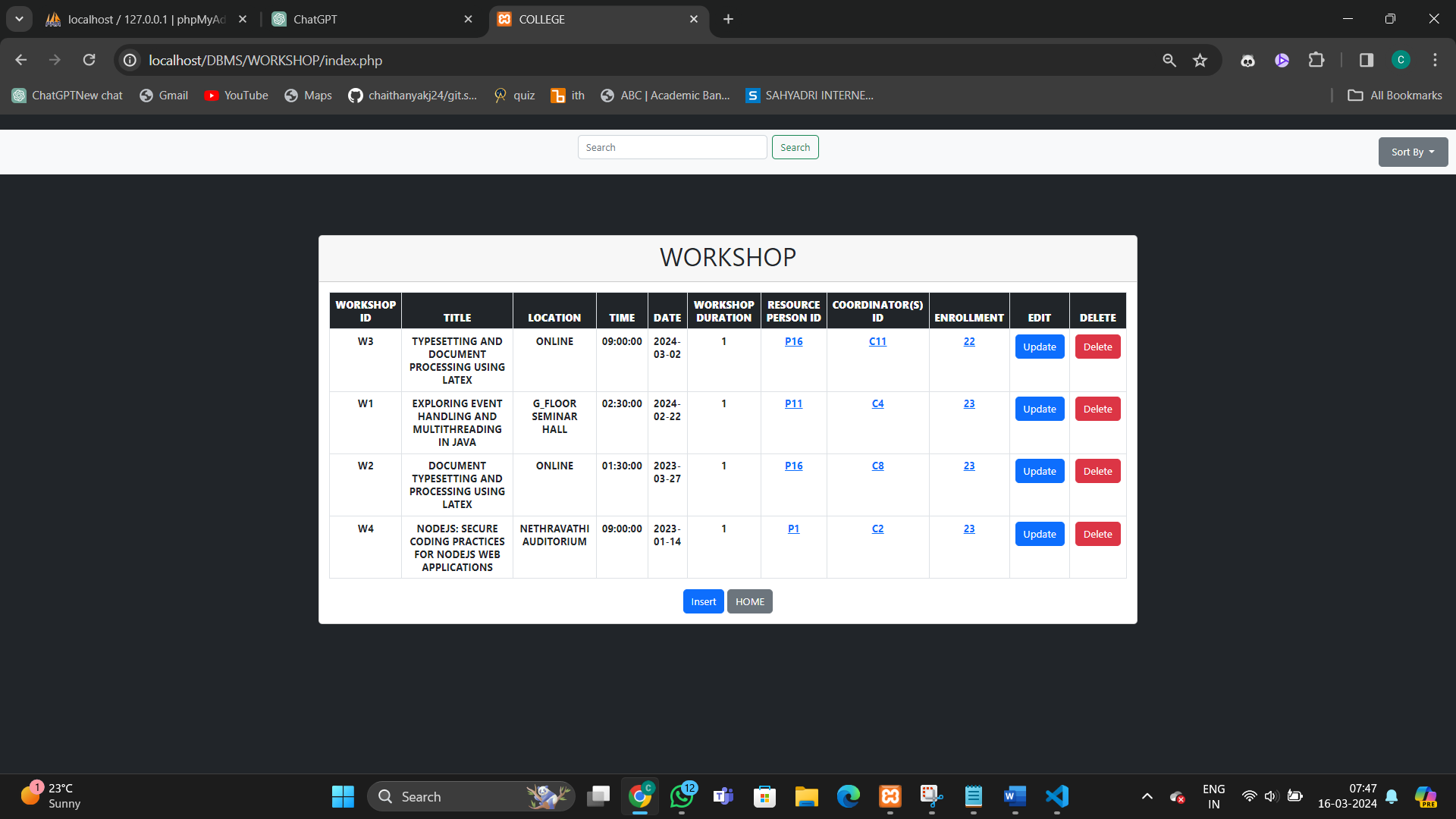
**RESULT**

**6.1 HOME PAGE**



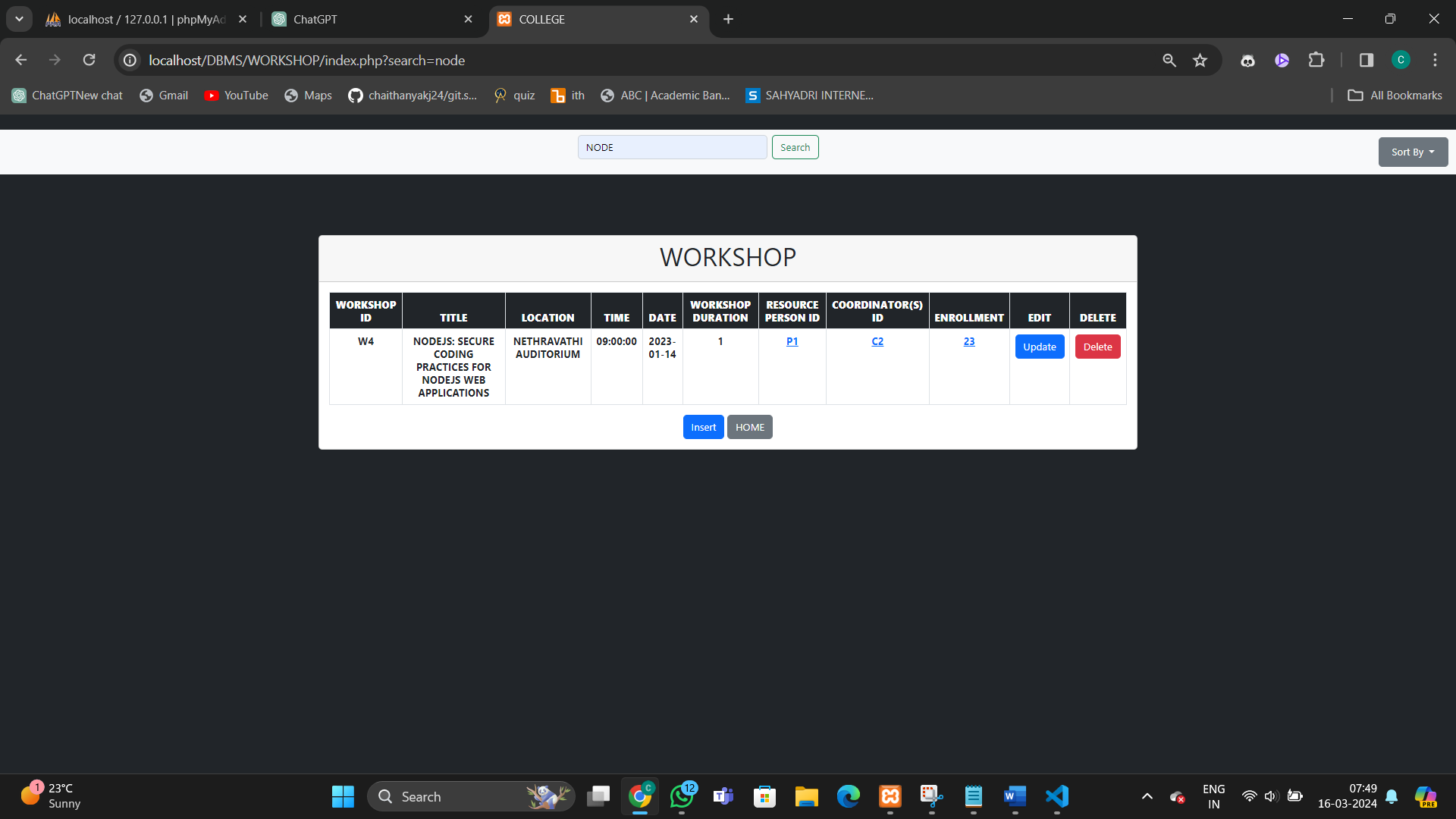
**Fig 6.1 Home Page**

**6.2 WORKSHOP VIEW PAGE**

****

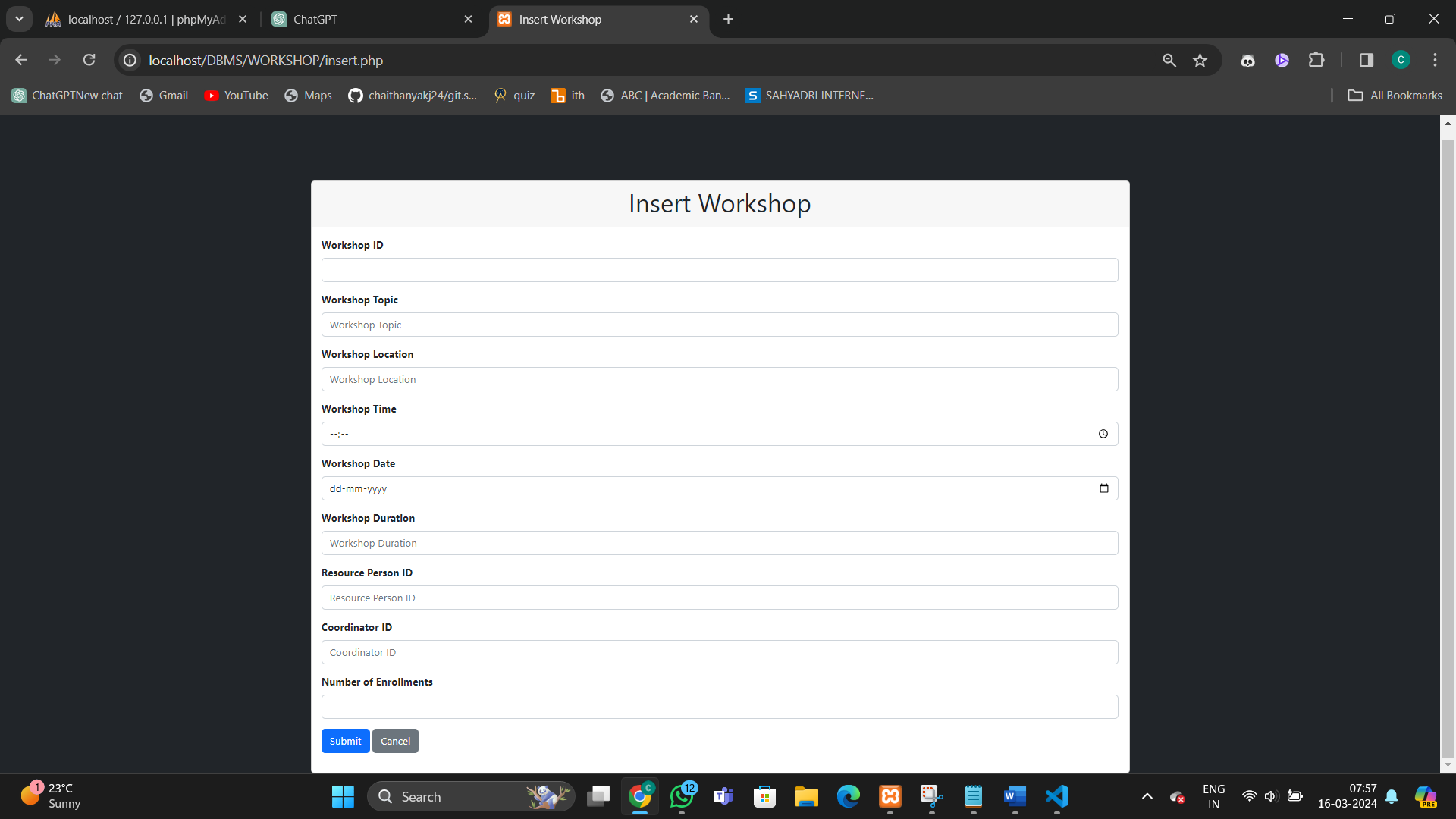
**Fig 6.2 Workshop View Page**

**6.3 WORKSHOP SEARCH PAGE**

****

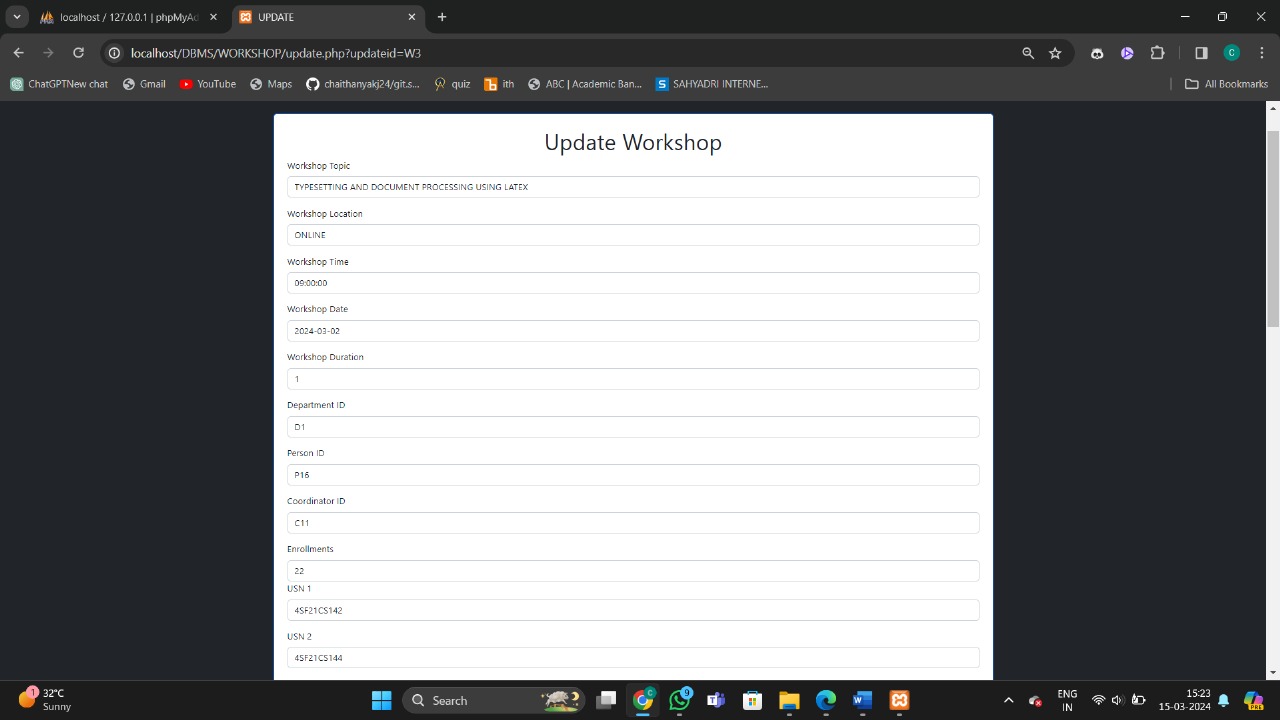
**Fig 6.3 Workshop Search Page**

**6.4 WORKSHOP INSERT PAGE**

****

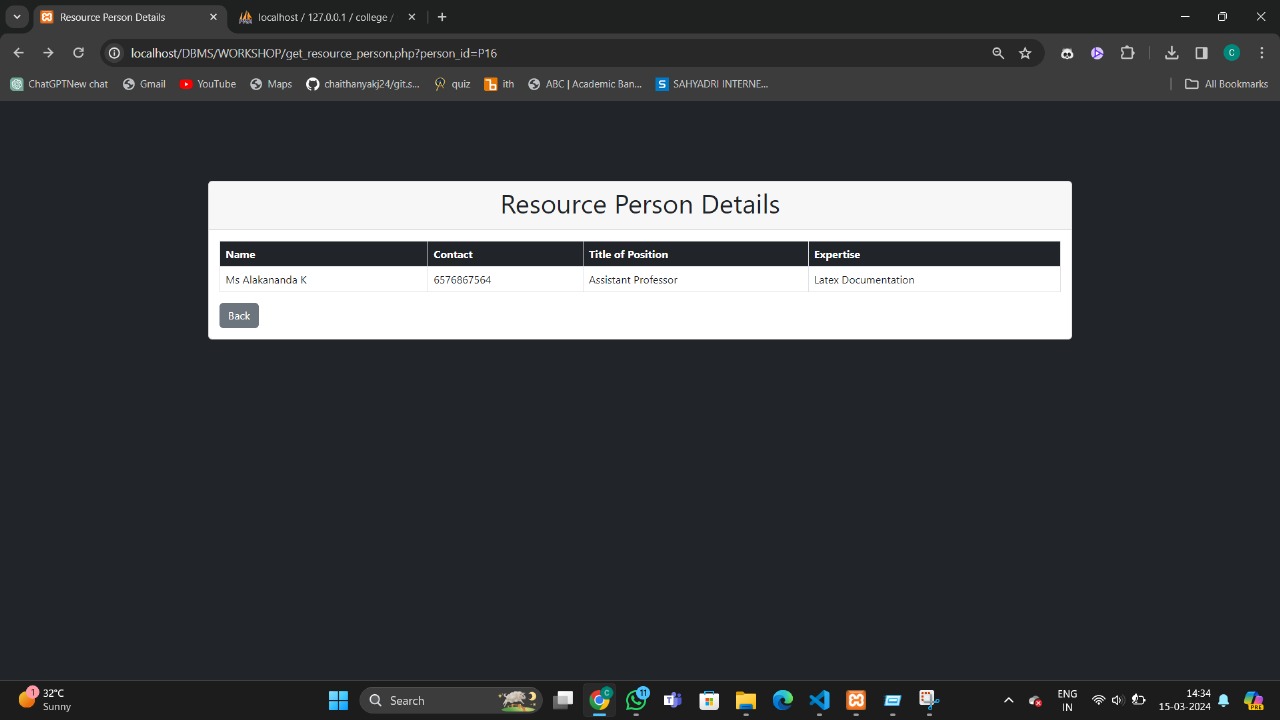
**Fig 6.4 Workshop Insert Page**

**6.5 WORKSHOP UPDATE PAGE**



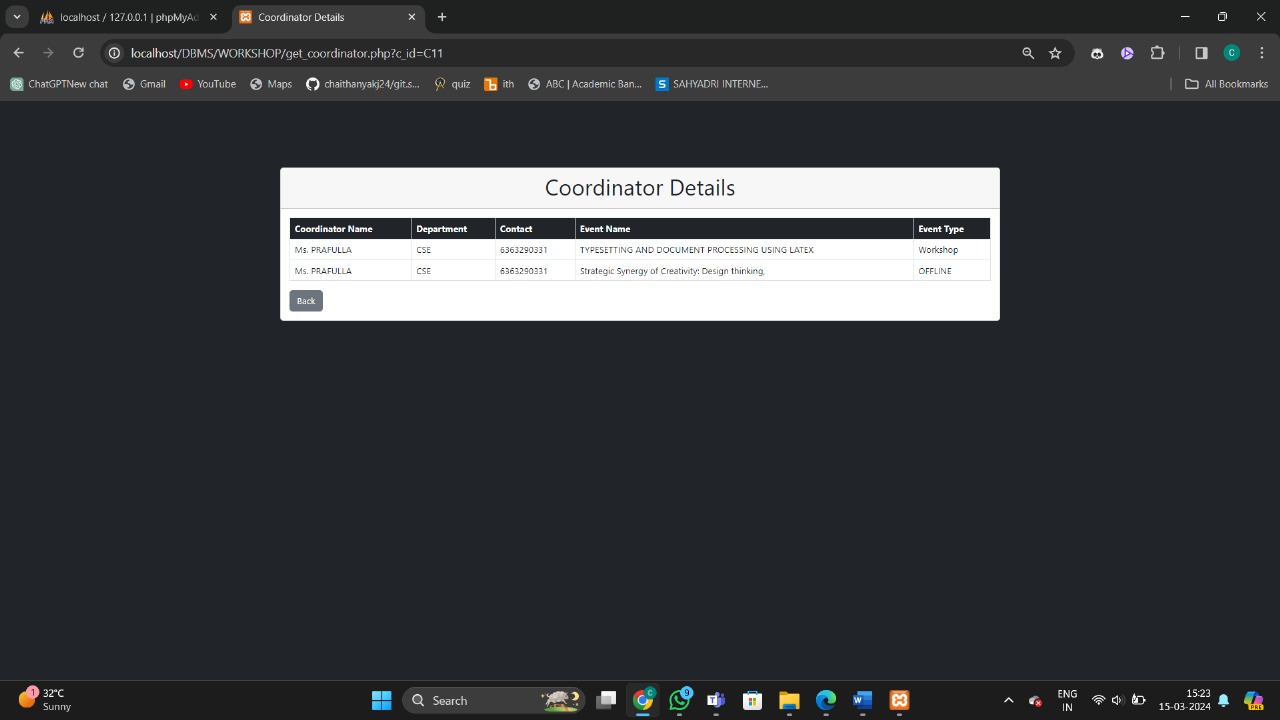
**Fig 6.5 Workshop Update Page**

**6.6 RESOURCE PERSON DETAILS PAGE**



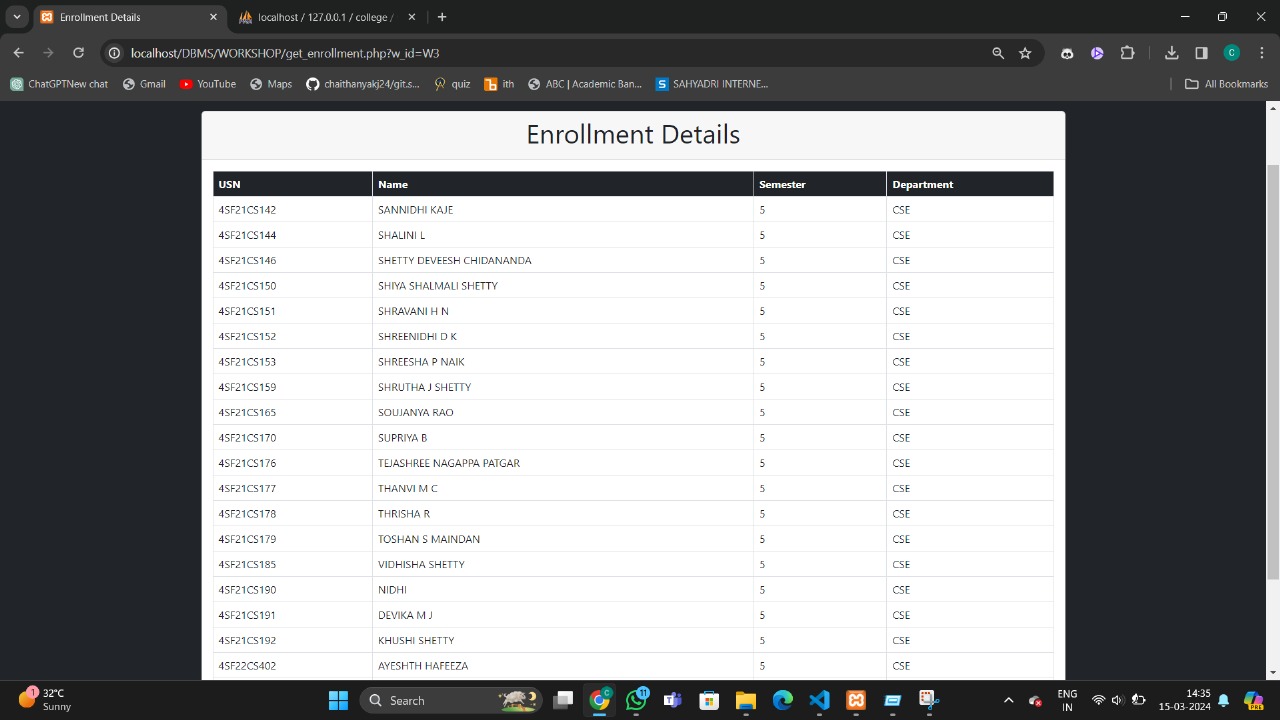
**Fig 6.6 Resource Person Details Page**

**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>