Chapter 1

INTRODUCTION

1.1 Introduction to Project

An ordered collection is necessary when important data i.e, projects are submitted to an organisation. Our project is an ordered collection of projects of all students of V sem CSE, VKIT who have created projects for their DBMS Laboratory.

Our project is a UI friendly way of entering, updating and deleting information of their individual projects. This is done by creating distinct login credentials through the interface. A user is then led onto a gateway page where he/she is presented with a number of options. These options are EDIT, VIEW, SEARCH and OPERATIONS.

The EDIT page is an interface provided to the user to enter data related to their project. These data are then stored into a database. This data is later retrieved and displayed in a presentable format. In this way our program is able to acquire all the data that it needs to perform display.

The VIEW page leads the user to a static view of their data i.e., their project. This page is connected to the database and the data retrieved is displayed here.

The SEARCH page is a comprehensive layout where the user is given the choice to search for a particular project. This page is coded in a way that it accepts either the project name or the creator of the project as a parameter to retrieve the project from the database. The output of this page is that it displays a link to the particular searched project's display section.

The OPERATIONS page is a page where the page is created to display SQL friendly operations. The page is coded in a way that is used as a Demonstration of how SQL works with PHP and HTML. This page is an example of multiple SQL queries.

The way that our project works is a way in which we have an interface built to please the eye and make an informational friendly. Project Showcase is the normalized method to display projects. Our project has a beautiful GUI mainly made by Cascading Style Sheets and JavaScript.

A brief introduction to our project is more than enough for the reader to understand the how our project works superficially.

1.2 DBMS

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Any piece of information can be a data, for example name of your school. Database is actually a place where related piece of information is stored and various operations can be performed on it.

A DBMS is a software that allows creation, definition and manipulation of database. Dbms is actually a tool used to perform any kind of operation on data in database. Dbms also provides protection and security to database. Here are some examples of popular dbms, MySQL, Oracle, Sybase, Microsoft Access and IBM DB2 etc.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical Database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.2.1 Components of Database System

The database system can be divided into four components.

- **Users:** Users may be of various type such as DB administrator, System developer and End users.
- **Database application:** Database application may be Personal, Departmental, Enterprise and Internal.

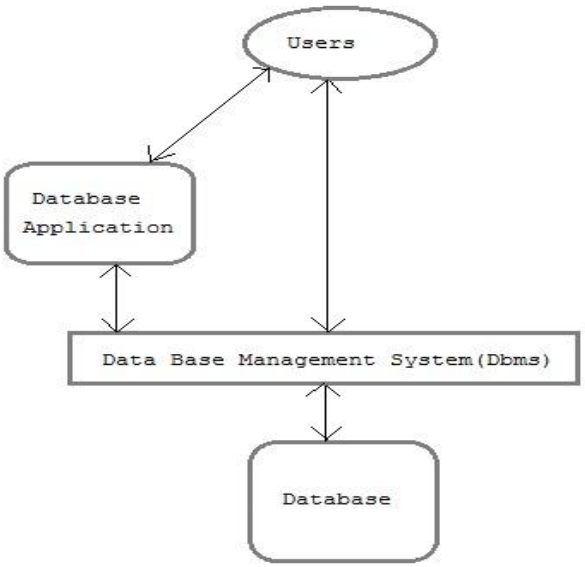

Fig 1.1 Components of database system

- **DBMS:** Software that allow users to define, create and manages database access, Ex: MySQL, Oracle etc.

- **Database:** Collection related data and facts and figures that can be processed to produce information.

A DBMS can limit what data the end user sees, as well as how that end user can view the data, providing many views of a single database schema. End users and software programs are free from having to understand where the data is physically located or on what type of storage media it resides because the DBMS handles all requests.

1.3 Structured Query Language

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. This tutorial will give you a quick start to SQL. It covers most of the topics required for a basic understanding of SQL and to get a feel of how it works.

SQL (Structured Query Language) is a standardized programming language used for managing relational databases and performing various operations on the data in them. Initially created in the 1970s, SQL is regularly used by database administrators, as well as by developers writing data integration scripts and data analysts looking to set up and run analytical queries.

The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements and commonly used SQL statements include select, add, insert, update, delete, create, alter and truncate.

SQL commands are divided into several different types, among them data manipulation language (DML) and data definition language (DDL) statements, transaction controls and security measures. The DML vocabulary is used to retrieve and manipulate data, while DDL statements are for defining and modifying database structures. The transaction controls help manage transaction processing, ensuring that transactions are either completed or rolled back if errors or problems occur. The security statements are used to control database access as well as to create user roles and permissions.

1.4 FRONT END DEVELOPMENT USING HTML5/CSS AND PHP

HTML 5 is a markup language used for structuring and presenting content on the World Wide Web. HTML 5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements are added. New attributes are introduced, some elements and attributes have been removed, and others have been changed, redefined, or standardized.

CSS (cascading style sheets) explains how using them with HTML pages is a user interface (UI) development best practice that complies with the separation of concerns design pattern.CSS is the standard and preferred mechanism for formatting HTML pages. Conforming with the separation of concerns design pattern and best practice, cascading style sheets provide a central location in which information about what various fonts, foreground colors, background colors, italicization and emphasization should be applied to various HTML elements within a webpage. Cascading style sheets can also control how various parts of a page, such as the header, footer, body, article content, sections and asides, are laid out on the page. This is extremely helpful when content must be laid out in

dramatically different fashion depending upon whether it is being viewed on a desktop, tablet or a smartphone.

PHP is a script language and interpreter that is freely available and used primarily on Linux Web servers. PHP stands for *Hypertext Pre-processor*, which the PHP FAQ describes as a "recursive acronym."PHP executes on the server, while a comparable alternative, JavaScript, executes on the client. PHP is an alternative to Microsoft's Active Server Page (ASP) technology. As with ASP, the PHP script is embedded within a Web page along with its HTML. Before the page is sent to a user that has requested it, the Web server calls PHP to interpret and perform the operations called for in the PHP script.

An HTML page that includes a PHP script is typically given a file name suffix of ".php" ".php7," or ".phtml". Like ASP, PHP can be thought of as "dynamic HTML pages," since content will vary based on the results of interpreting the script.

1.5 Three Schema Architecture

The goal of the three-schema architecture, illustrated in Figure 2.2, is to separate the user applications from the physical database. The three-schema architecture is a convenient tool with which the user can visualize the schema levels in a database system

Following are the three levels of database architecture

1. Physical or Internal Level
2. Conceptual Level
3. External or View Level

The **internal level** has an **internal schema**, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

The **conceptual level** has a **conceptual schema**, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is

implemented. This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.

The **external** or **view level** includes a number of **external schemas** or **user views**. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level data model.

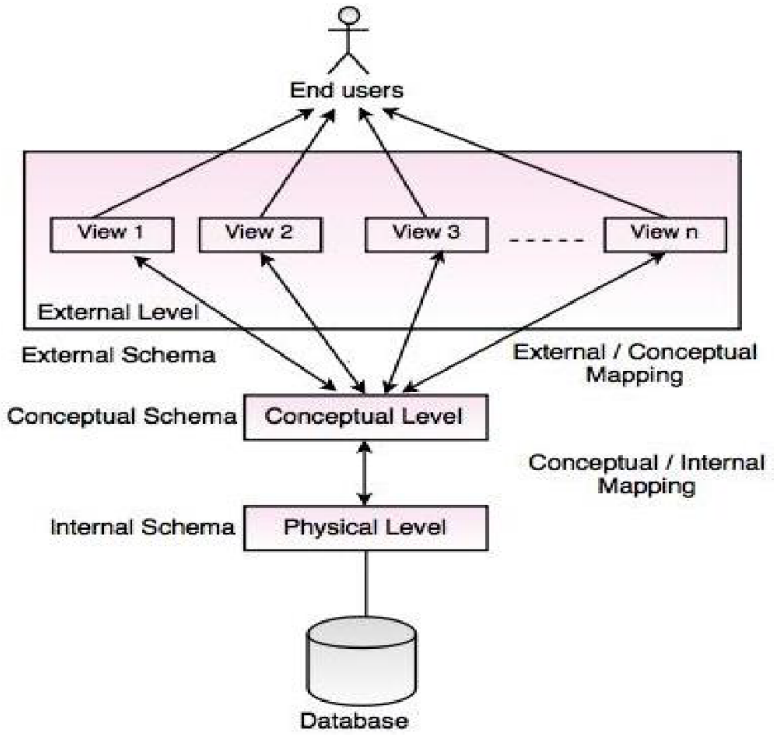

Fig 1.2 The Three Tier Architecture

Most DBMSs do not separate the three levels completely and explicitly, but support the three-schema architecture to some extent. Some older DBMSs may include physical-level details in the conceptual schema. The three-level ANSI architecture has an important place in database technology development because it clearly separates the users' external level, the database's conceptual level, and the internal storage level for designing a database.

It is very much applicable in the design of DBMSs, even today. In most DBMSs that support user views, external schemas are specified in the same data model that describes the conceptual-level information (for example, a relational DBMS like Oracle uses SQL for this). Some DBMSs allow different data models to be used at the conceptual and external levels. An example is Universal Data

Base (UDB), a DBMS from IBM, which uses the relational model to describe the conceptual schema, but may use an object-oriented model to describe an external schema.

Notice that the three schemas are only descriptions of data; the stored data that actually exists is at the physical level only. In a DBMS based on the three-schema architecture, each user group refers to its own external schema. Hence, the DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database. If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view. The processes of transforming requests and results between levels are called **mappings**. These mappings may be time-consuming, so some DBMSs—especially those that are meant to support small databases—do not support external views. Even in such systems, however, a certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

1.6 Normalization of Database

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes:

- Eliminating redundant (useless) data.
- Ensuring data dependencies make sense i.e. data is logically stored

Here are the most commonly used normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF

1.5.1 First Normal Form (1NF)

As per First Normal Form, no two Rows of data must contain repeating group of information i.e. each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. This rule defines that all the attributes in a relation must have atomic domains. The values in an

atomic domain are indivisible units. Each attribute must contain only a single value from its pre-defined domain.

1.5.2 Second Normal Form (2NF)

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.

1.5.3 Third Normal Form (3NF)

Third Normal form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this transitive functional dependency should be removed from the table and also the table must be in Second Normal form.

For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy –

- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency, X → A, then either –
- X is a super key or,
- A is prime attribute.

1.5.4 Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

- R must be in 3rd Normal Form
- And, for each functional dependency (X → Y), X should be a super Key.

Chapter 2

REQUIREMENTS SPECIFICATION

2.1 Requirements Specification:

2.1.1 Software Requirements Specification

A software requirements specification (SRS) – a requirements specification for a software system – is a complete description of the behaviour of a system to be developed. In addition to a description of the software functions, the SRS also contains non-functional requirements. Software requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The system software requirements are: -

1. Operating system requirements: The operating system that can be used Windows 7, Windows 8, Windows 10.
2. IDE requirements: Sublime text editor, Chrome browser, Windows/ MacOS/Linux.
3. Programming language: For coding purpose HTML5 and PHP is used.
4. Other Tools: MySQL.

2.1.2 Hardware Requirements

1. System: Intel core i7 below and above
2. Hard disk: 250GB-1TB
3. Floppy drive: 1.44 Mb
4. Monitor: 15 VGA color
5. Ram: 2GB-16GB.

2.2 Connectivity

To connect the MySQL with PHP, the following steps listed below must be followed. The five main steps to make PHP database interaction is,

- To create connection
- To select the Database

- Perform database query
- Use return data
- Close Connection

Code for connectivity:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Chapter 3

DATABASE DESIGN

3.1 ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types.

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
- Choose one of the key attributes of E as the primary key for R.
- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Step 2: Mapping of Weak Entity Types

- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
- Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
- The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any

Step 3: Mapping of Binary 1:1 Relation Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

Step 4: Mapping of Binary 1:N Relationship Types.

- For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
- Include any simple attributes of the 1:N relation type as attributes of S.

Step 5: Mapping of Binary M:N Relationship Types.

- For each regular binary M:N relationship type R, create a new relation S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
- Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Step 6: Mapping of Multivalued attributes.

- For each multivalued attribute A, create a new relation R.
- This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute. The primary key of R is the combination of A and K.
- If the multivalued attribute is composite, we include its simple components.

Step 7: Mapping of N-ary Relationship Types.

- For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

3.2.1 Notations

The following figure shows the different notations used in the ER Diagrams.

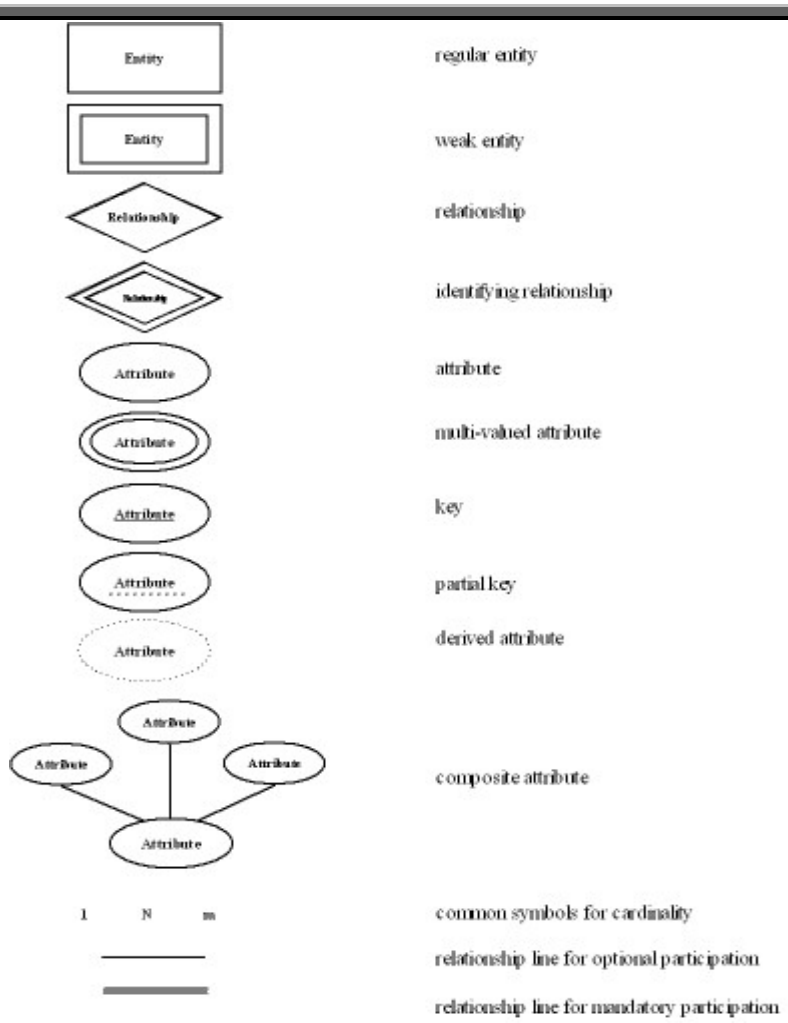

Fig 3.1 ER diagram notations

3.2 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

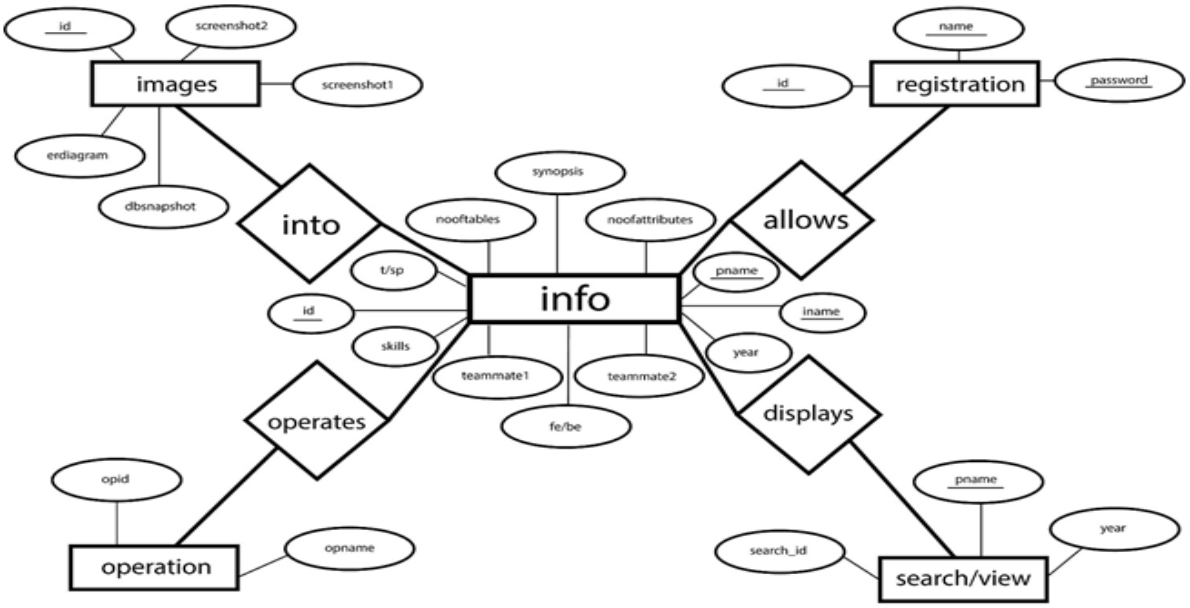

Fig 3.2 ER Diagram for Project Showcase

3.3 Schema

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed.

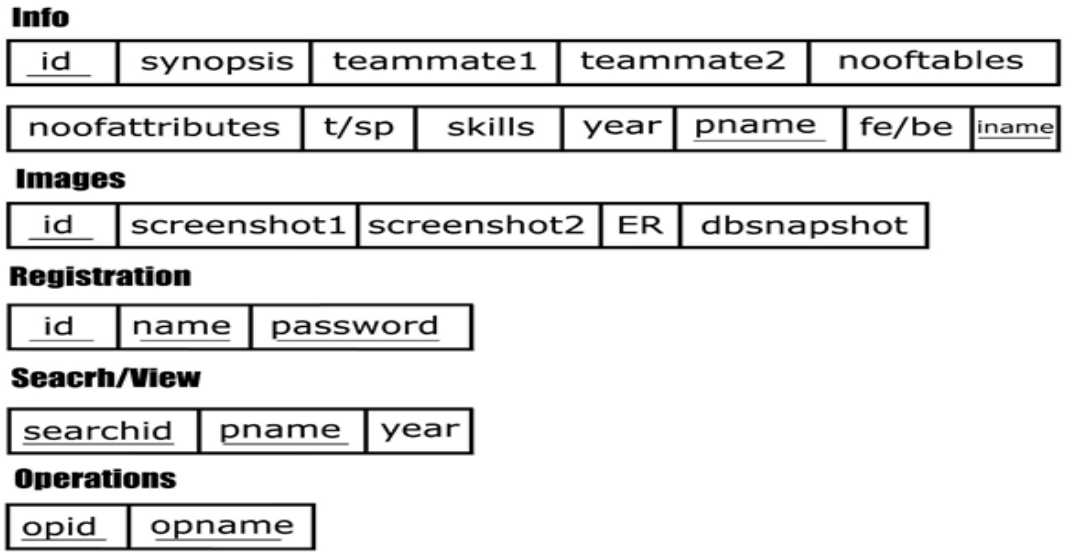

Fig.3.3 Schema Diagram for Project Showcase DBMS

Chapter 4

IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

Implementation is one of the most important phases of the Software Development Life Cycle (SDLC). It encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, and running, testing, and making necessary changes. Specifically, it involves coding the system using a particular programming language and transferring the design into an actual working system.

This phase of the system is conducted with the idea that whatever is designed should be implemented; keeping in mind that it fulfils user requirements, objective and scope of the system. The implementation phase produces the solution to the user problem.

4.1 Source code

The following gives some of the code snippets at the front end used for inserting, updating, deleting, and viewing the operations.

4.1.1 Code for insertion

The below code creates a connection with the backend- SQL Server. This connection is used to perform the insertion operation. The values inserted are taken from the textboxes on the form.

```
<div class="row portfolio-content"

  <div class="col-twelve">

   <!-- portfolio-wrapper -->

    <div id="folio-wrapper" class="block-1-2 block-mob-full stack">
```

```
<div class="bgrid folio-item">

 <div class="item-wrap">

        <img src="images/portfolio/edit.jpg" alt="Liberty">

        <a href="#modal-01" class="overlay">

          <div class="folio-item-table">

            <div class="folio-item-cell">

              <h3 class="folio-title">EDIT</h3>

              <span class="folio-types">

              </span>

            </div>

          </div>

        </a>

      </div>

    </div> <!-- /folio-item -->
```

4.1.2 Code for Updating

This below code again uses the connection for updating the information about the user as well as the project.

```
//ADD UPDATE TABLES

  if(isset($_POST['autables'])){ //check if form was submitted

    $nt = $_POST['notables']; //get input text
```

```
    $na=$_POST['noattributes'];

    $tsp=$_POST['tsp'];

    $s="UPDATE `info` SET `notables`='$nt',`noattributes`='$na',`tsp`='$tsp' WHERE
id='".$_SESSION['ids']."' ";

    mysqli_query($con,$s);

  }

//ADD UPDATE TABLES

   if(isset($_POST['auskill'])){ //check if form was submitted

    $skill = $_POST['skill']; //get input text

    $s="UPDATE `info` SET `skills`='$skill' WHERE id='".$_SESSION['ids']."' ";

    mysqli_query($con,$s);

   }
```

4.1.3 Code for deletion

Again here are some of the lines of code that perform the delete operation .

```
  //Delete Name

    if(isset($_POST['namedel'])){ //check if form was submitted

      $s="UPDATE `info` SET `name`=' ' WHERE id='".$_SESSION['ids']."' ";

      mysqli_query($con,$s);

    }

   //Delete Synopsis
```

```
    if(isset($_POST['syndel'])){ //check if form was submitted
      $s="UPDATE `info` SET `synopsis`=' ' WHERE id='".$_SESSION['ids']."' ";
      mysqli_query($con,$s);
    }
  if(isset($_POST['erdelete'])){ //check if form was submitted
      $s="UPDATE `images` SET `er`=' ' WHERE id='".$_SESSION['ids']."' ";
      mysqli_query($con,$s);
    }
     if(isset($_POST['db_delete'])){ //check if form was submitted
      $s="UPDATE `images` SET `dbsnap`=' ' WHERE id='".$_SESSION['ids']."' ";
      mysqli_query($con,$s);
    }
     if(isset($_POST['sc_delete2'])){ //check if form was submitted
      $s="UPDATE `images` SET `screenshot2`=' ' WHERE id='".$_SESSION['ids']."' ";
      mysqli_query($con,$s);
  if(isset($_POST['sc_delete'])){ //check if form was submitted
      $s="UPDATE `images` SET `screenshot`=' ' WHERE id='".$_SESSION['ids']."' ";
      mysqli_query($con,$s);
    }
```

4.1.4 Code for Operating

This is the code for the operation of the code.it deals with both coordination of HTML5/CSS and PHP. And also MySQL.

```
//echo $result[0];

?>

<!DOCTYPE html>

<!--[if IE 8 ]><html class="no-js oldie ie8" lang="en"> <![endif]-->

<!--[if IE 9 ]><html class="no-js oldie ie9" lang="en"> <![endif]-->

<!--[if (gte IE 9)|!(IE)]><!--><html class="no-js" lang="en"> <!--<![endif]-->

<head>

  <!--- basic page needs

        ================================================== -->

        <meta charset="utf-8">

        <title></title>

        <meta name="description" content="">

        <meta name="author" content="">

        <!-- mobile specific metas

        ================================================== --

       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

        <!-- CSS
```

```
        ================================================== -->

        <link rel="stylesheet" href="css/base.css">

        <link rel="stylesheet" href="css/main.css">

        <link rel="stylesheet" href="css/vendor.css">

    <!-- script

    ================================================== -->

    <script src="js/modernizr.js"></script>

    <script src="js/pace.min.js"></script>

    <!-- favicons

    ================================================== -->

    <link rel="icon" type="image/png" href="favicon.png">

    </head>
```

4.1.5 Code for viewing

This is the code that executes the contents of stored procedure, which includes a select query for retrieving the details for reference by the respective project.

```
<?php

  session_start();

  if(!isset($_SESSION['username'])){

      header('Location: loginerror.html');

   }
```

```
 $con = mysqli_connect("localhost", "root", "jesus123", "spm");

 $info = mysqli_query($con,"SELECT * FROM info WHERE id='".$_SESSION['ids']."'; ");

   $sql = "SELECT * FROM images WHERE id='".$_SESSION['ids']."';";

$sth = $con->query($sql);

$result=mysqli_fetch_array($sth);

?>

<!DOCTYPE html>

<!--[if IE 8 ]><html class="no-js oldie ie8" lang="en"> <![endif]-->

<!--[if IE 9 ]><html class="no-js oldie ie9" lang="en"> <![endif]-->

<!--[if (gte IE 9)|!(IE)]><!--><html class="no-js" lang="en"> <!--<![endif]-->

<head>

  <!--- basic page needs

  ================================================== -->

  <meta charset="utf-8">

        <title></title>

        <meta name="description" content="">

        <meta name="author" content="">

 <!-- mobile specific metas

  ================================================== -->

        <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
```

```
<!-- CSS

  ================================================== -->

  <link rel="stylesheet" href="css/base.css">

  <link rel="stylesheet" href="css/main.css">

  <link rel="stylesheet" href="css/vendor.css">       <!-- script
================================================== -->

        <script src="js/modernizr.js"></script>

        <script src="js/pace.min.js"></script>

 <!-- favicons

        ================================================== -->

        <link rel="icon" type="image/png" href="favicon.png">

</head>
```

4.1.5 Code for search

This code is used for searching projects and their description using the user name , password and the USN.

```
<!-- contact

    ================================================== -->

    <section id="contact">

 <div class="row section-intro">

       <div class="col-twelve">
```

```
<h5>Search</h5>

   <h1><?php echo $_SESSION['username'];?> logged in!</h1>

  <p class="lead"></p>

 </div>

</div>

<div>

<form method="post" action="searchpage.php">

 <center>

  <input  type="text" name="query" style="text-align:center;">

      <br>

  <input  type="submit" value="Search">

 </center>

</form>

<div class="row contact-form">

 <div class="row contact-info">

</div>
```

4.2 Queries

The following are some of the queries that can be executed to retrieve information for different need and purpose of the database.

“SELECT COUNT(id) FROM info;"

"SELECT pname FROM info WHERE fe LIKE '%html%'"

"SELECT pname,year FROM info ORDER BY year ASC;"

"UPDATE `images` SET `screenshot2`='$image' WHERE id='".$_SESSION['ids']."' ";

"UPDATE `info` SET `notables`='$nt',`noattributes`='$na',`tsp`='$tsp' WHERE id='".$_SESSION['ids']."' ";

Chapter 5

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is a process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Testing Techniques

5.1 White box testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Unit testing

Unit testing involves the design of test cases that validate the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is dine after the completion of an individual unit before integration.

This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2 Black box testing

Black Box Testing is testing software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you

cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.3 Grey box testing

Involves having knowledge of internal data structures and algorithms for purpose of designing tests, while executing those tests at the user, or black-box level. Manipulating input data and formatting output do not qualify as grey box, because the input and output are clearly outside of the “black box” that we are calling the system under test.

5.4 Test cases

5.4.1 Test case for user interface

Steps	Test Action	Results
Step 1	Run the program	Successful display of the Login page.
Step 2	Press login button	Takes you to the login page. Enter your name and password. If account not yet created, then create an account.
Step 3	Select any of them out of Synopsis, screenshots etc,.	Takes you to the respective option and displays the information.
Step 4	Edit the information	Helps you to edit the information of the respective project. Update can also be done.
Step 5	View and ratings	Helps us to view the project of the given student and we can also rate that project with your view of perspective.

Table 5.1 Test case for user interface

5.4.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors with an interface. This testing is done with simple data and developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

Steps to perform integration testing:-

Step 1: Create a test plan

Step 2: Create Test Cases and Test Data

Step 3: Once the components have been integrated execute the test cases

Step 4: Fix the bugs if any and re test the code

Step 5: Repeat the test cycle until the components have been successfully integrated

Name of the Test	Integration testing
Test plan	To check whether system works properly when all the modules are integrated.
Test data	The data entered by the students.

Table 5.2 test case for integration testing

5.4.3 System Testing

Ultimately, software is included with other system components and the set of system validation and integration tests are performed. System testing is a series of different tests whose main aim is to fully exercise the computer-based system. Although each test has a different role all work should verify that all system elements are properly integrated and formed allocated functions.

Name of the Test	**System Testing**
Item being tested	The working of the forms for each request made by the user.
Sample Input	User details.
Expected Output	Redirection to the appropriate form upon request which includes reserving, updating and cancelling.
Actual Output	Shows the entries made by the user in a separate grid and the input is reflected in the database.
Remarks	Successful

Table 5.3 Test case for input-output

5.4.4 Validation Testing

At the culmination of black box testing, software is completely assembled is as a package. Interfacing errors have been uncovered and the correct and final series of tests, i.e., validation tests begins. Validation test is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

5.4.5 Output Testing

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system is required tests the output displayed or generated by the system under consideration. The output format is considered in two ways, one is on screen format and the other is printed format. The output format on the screen is found to be corrected as the format was designated in the system has according to the user needs. As for the hard copy the output comes according to the specification requested by the user. The output testing does not result in any correction in the system.

5.4.6 Test data and Output

Taking various kind soft data plays a vital role in system testing. After preparing the test data system under study is tested using the test data. While testing, errors are again uncovered and corrected by using the above steps and corrections are also noted for future use.

5.4.7 User acceptance Testing

User acceptance testing of the system is the key factor for the success of the system. A system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

5.4.8 GUI Testing

GUI testing is use to ensure the visual clarity of the system, flexibility of the system, user friendliness of the system. The various components which are to be tested are:

• Relative layout

• Various Links and Buttons

Chapter 6

RESULTS AND SNAPSHOTS

The experimental results are shown by giving some snapshots of the output of this project.

The following are some of them.

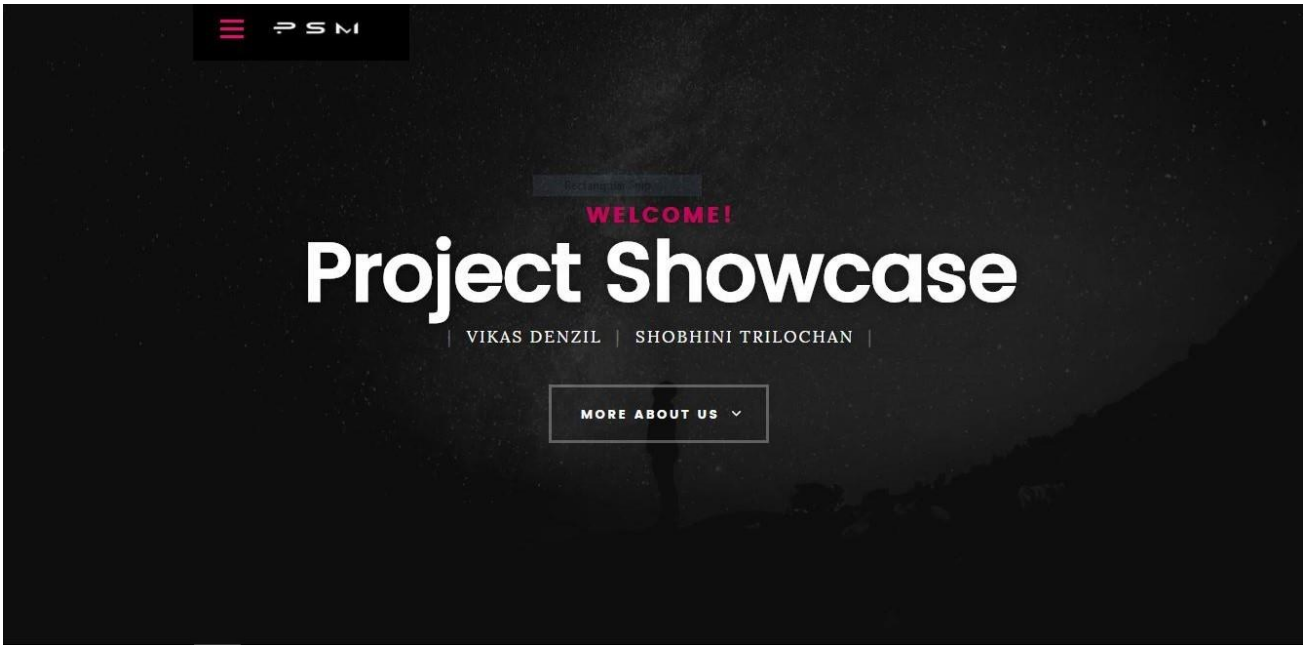

Fig6.1 Welcome page

This snapshot is the immediate display when the program is executed. The ‘more about us’ button leads to the display of our information. And the three points on the left-top of the page gives two options, home and login.

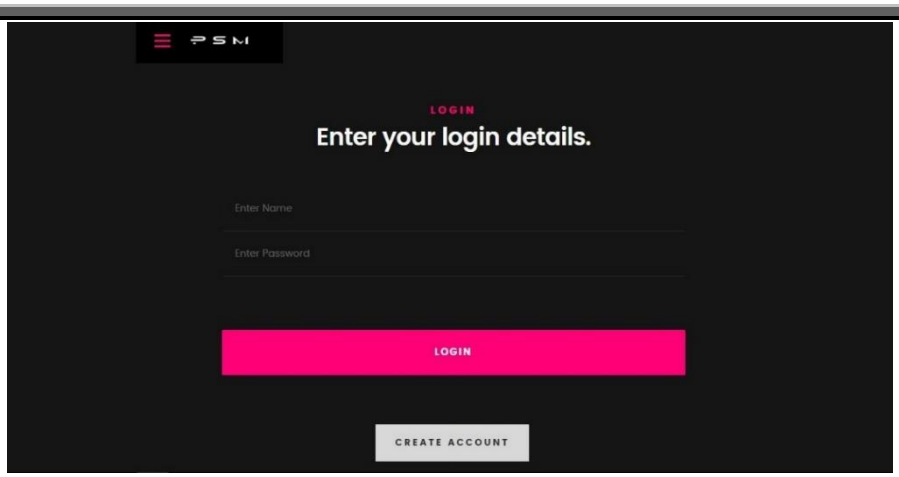

Fig 6.2 The Login Page

This page helps the user to login to our project. For getting into this project, they will have to enter the User name and User password and then enter the login button. If the account is not created then, enter Create Account button and create an account.

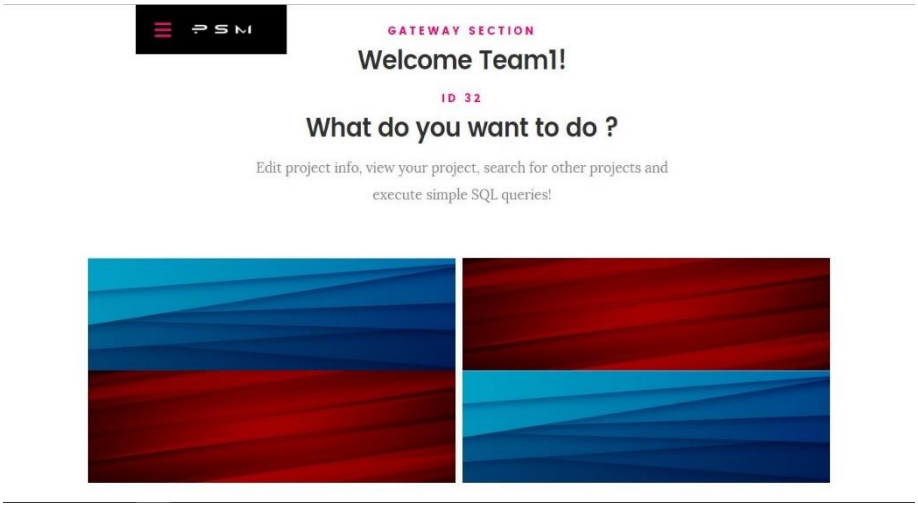

Fig 6.3 Gateway Section

After logging into the project, we enter the Gateway Section where we have four options. The four options are EDIT, VIEW, SEARCH and, OPERATION. Each one on clicking takes us to its respective page.

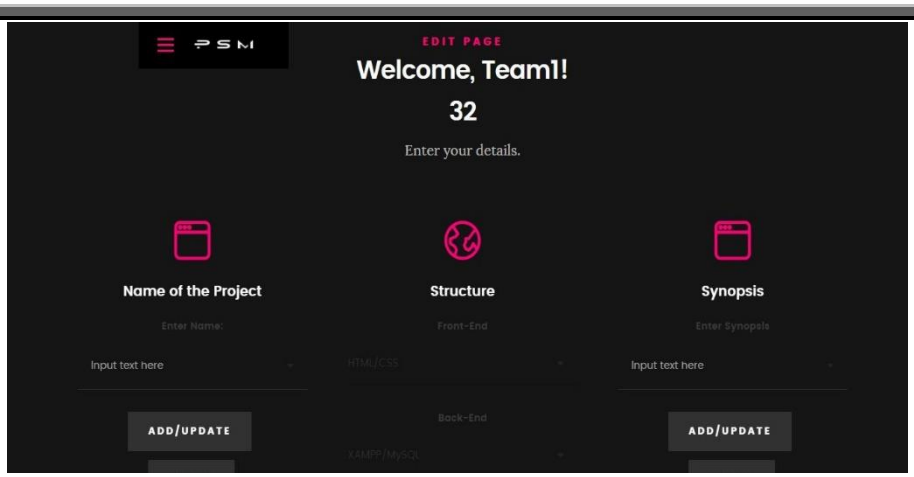

Fig 6.4 Edit Page

On pressing the EDIT button, we enter into Edit page where we can upload the information and make the updates. The information that we upload here is Name of the project, Structure of the project and Synopsis of the project.

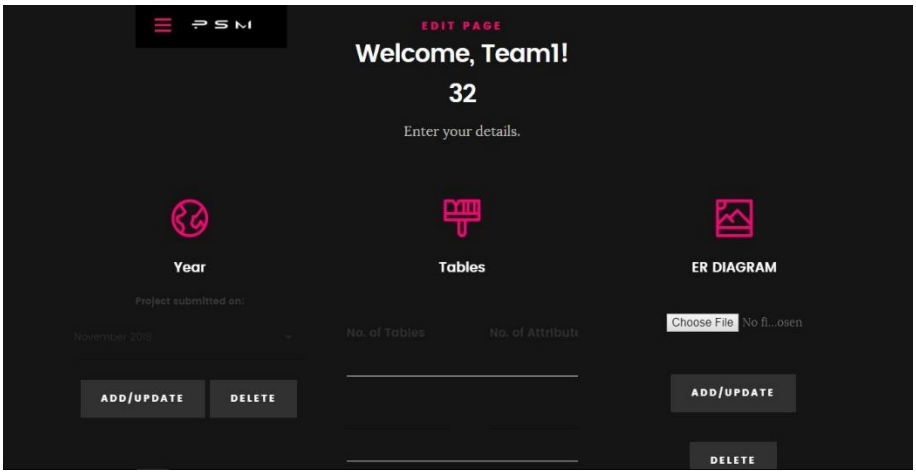

Fig6.5 Edit Page

This is another slide of Edit page where we will have upload the year, Tables and the ER diagram of the projects. And they can also be edited.

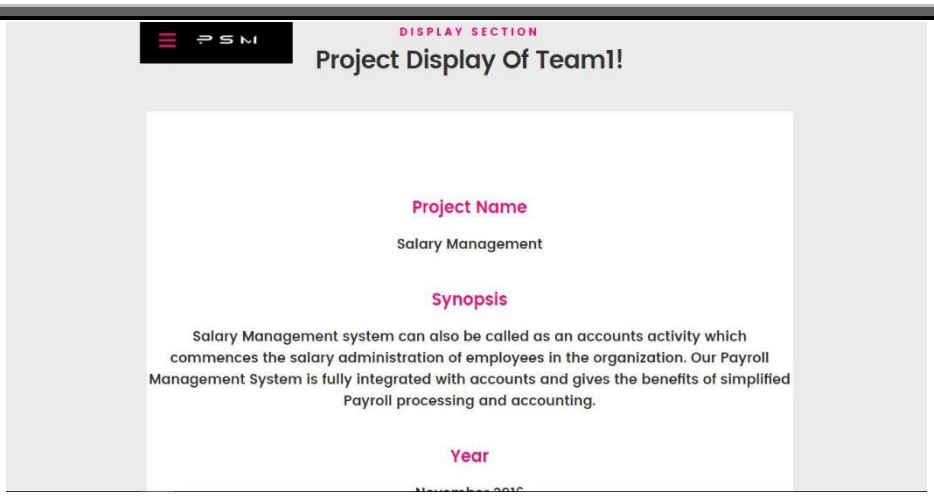

Fig 6.6 Display section

This page gives the information about the projects that have logged in.

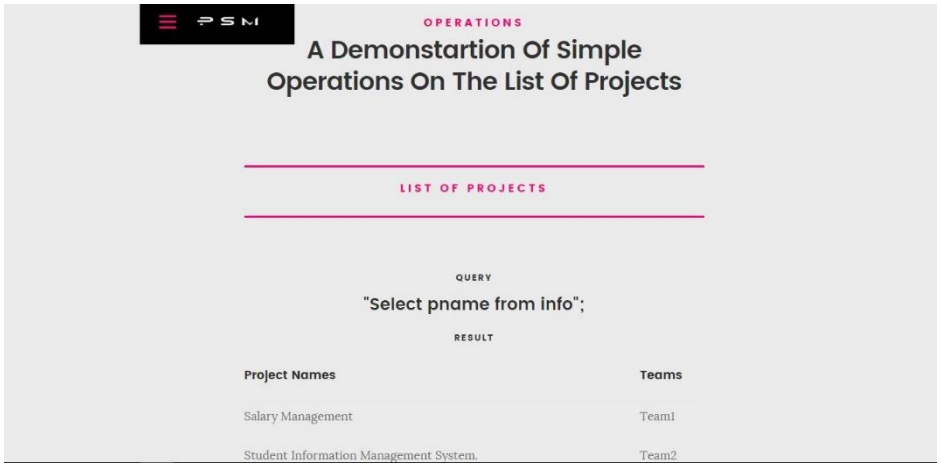

Fig 6.7 Operations page

This page is used to show the working of queries that we have inserted. The queries can be anything.

The information will be displayed on the screen according to the query.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENT

Project Showcase System has led to ease of viewing the description of all the projects and also provided a means for teachers and others to access and view the descriptions and details about the projects with ease and in time. It has also increased the speed with which information about students are retrieved and handled and project scheduling is tasked.

This report's content comprises the whole task solution, starting from the programming environments, going through the database, the application analysis and constructions, and finishing with code-implementation and test samples.

Our project gives a clear idea of a Project Showcase which provides an easy way for users to handle the database since they can directly insert, delete, update the entries without worrying about the backend codes.

Even though Project Showcase is not a business critical application, they are functionally quite complex, the operation of a Project management system is relatively expensive and it should be a continuous process. Some effective means can be adopted to make it effective and affordable as well.

Further, Project Showcase can be turned into a website where the students themselves can upload the changes done in their projects.

BIBLIOGRAPHY

1. Github- https://github.com
2. Codepen