1. INTRODUCTION

The objective of "Employee Management System" is designing a Scheduling System for a work centre. Scheduling is such a tool with which the process of intimating activities and notifications will be easy and even online in the organization where it is installed. But these task of scheduling the different activities if manually done whether they may be personal or official is time consuming and also may lead to confusion if not properly scheduled.

Scheduling becomes such an easy task such that it reduces much time when compared to previous methods. This enables for the employee to check the task that is assigned to them. The Supervisor is already having the account on the server and therefore will have login username and password. The Supervisor is allowed to delete employees,update details, view details etc. The Employee will have to register in the database and once registered he/she would be logged in to the system through his/her username and password. The employee is allowed to view details.

1.1 Existing System

The existing system is based on the manual work carried out by the different department, where you have to do all jobs manually and do not allow the automation of system and transparency to all users of system. Disadvantage of existing system is that everything is on paper, like for storing the details of the particular employee, to do so either that employee should be in contact or should be known by some other source resulting in the wastage of time. So, this whole task of scheduling the details is very time consuming and may lead to confusion if not properly scheduled.The system which was developed now, makes this process of scheduling much easier and computerized. By this system the manager or top level designated employee can fix the schedule of any employee working under him. Thus, the top-level management can easily fix the process of scheduling , saving a lot of time and work overhead.

1.1.1 Disadvantage

The main disadvantages in the existing system are the existing system can be accessed only through one place i.e., it means can be accessed only when either the employee or the supervisor can access only through the local host. The Employee or the supervisor cannot access it outside the office. This caused a huge problem in the problem in the present Corporate/Government world.

1.2 Proposed system

Our Proposed system is

- A new technology project.
- It can reduce the work time.
- It has a special ability that if a heavy amount of users visit our website even also our system does not hang or the server doesn't get busy .

1.2.1 ADVANTAGE OF THE SYSTEM

This project offers employees to enter the data through simple & interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
New Employees when registering themselves register according to category wise (i.e. Developer, Tester and Designer). Here, Employee Id is automatically generated once the user got registered. So, user logged in to the system and access to its profile and rights using that Employee Id and password.

The front view of the system consists of two modules i.e. Supervisor and Employee. When clicking on the Employee tab the pop-up window would open up and ask for the login, if

the employee is not yet register then clicking on **"New User Register Here",** user can register themselves. After registering they can log in to the system using password and Employee Id which is automatically generated after registration and access their rights like view schedule, view attendance, update details, view details etc.

Supervisor can log in to the system using username and password and can set schedule, mark attendance, check category wise description of the employees etc. Hence, the system has various advantages:

- Transparency to all the user of system.
- Less paper use and removal of redundancy.
- Less prone to errors.
- The whole system is interactive.

1.2.2 DEMERITS

The "Employee Management System" has following demerits.

- Currently it is an information system for supervisor and employee only.
- Limited to some basic operation.
- Security level is low.

1.2.3 Technology used for Implementation

"Employee Management System" will be adopting **3-Tier Architecture.** The front-end will be HTML pages with Java Script for client-side validation whereas all business customization will be in CSS (Cascading Style Sheets) reside for styling the webpage purpose. And these layers will interact with third layer of database, which will be '**My php login' database**. The web server will be Xampp (Apache Tomcat). To start working on this project environment required is a server having Tomcat as web server, My php Login as database.

"Employee Management System" software shall be designed and developed on Open Platform i.e. HTML, PHP, CSS, Xampp Server shall be used to maintain the database. To achieve a high degree of modularity, scalability and maintainability, it is recommended to adopt n-tier architecture while designing Employee Management System. Accordingly, it is proposed that the entire application development logic, the database logic and presentation logic shall be segregated. It is proposed that EMS software package shall be running on Apache Tomcat Web Server. The Web Server will be responsible for rendering the pages and result is shown back to the end-user.

1.2.4 OBJECTIVE

This report documents the process of designing, building and testing a software system to be used in a company. The piece of software, and therefore the project, is known as a **"Employee Management System".** This project basically includes two modules i.e. Supervisor and Employee.

The system will do the following:

1) **Supervisor:**

- Supervisor can delete the employees.
- Supervisor can accept employee registrations
- Supervisor can update the details of an employee.
- Supervisor can insert an employee.
- Supervisor can view the details category wise also.

2) Employee:

- Registration giving the details.
- Employee can view his/her details once the supervisor accepts the registration.
- Employee can update his/her details.

Hardware Requirements

Model	:	SUN Ultra SPARC IV, INTEL class
Processor	:	Pentium (or above)
RAM	:	1 GB
Hard Disk	:	20 GB

Software Requirements

Operating System	:	Windows XP or above
Database Server	:	My PHP Admin
Web Server	:	Apache Tomcat
Browser	:	Java Script enabed Browser

2. ARCHITECTURE AND IMPLEMENTATION

2.1 Introduction

The purpose of System Design is to create a technical solution that satisfies the functional requirements for the system. At this point in the project lifecycle there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organizational entities that will use the new system.

Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built.

At Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

2.2 Modularization Details

It is a technique of dividing the whole system into smaller independent modules.

There are two modules in our project:

1.<u>Supervisor Module</u>: Supervisor can do the following operations:

- Delete registered employee
- View category wise details.
- Update the details.

- Accept/Reject the details of the new user

2.<u>Employee Module</u>: - Employee can do the following operations:

- Register
- View Details.

2.3 Flowcharts

A flowchart is simply a graphical representation of steps. It shows steps in a sequential order, and is widely used in presenting flow of algorithms, workflow or processes. Typically, flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

Flowcharts are graphical representation of steps. It was originated from computer science as a tool for representing algorithms and programming logic, but had extended to use in all other kinds of processes. Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes, or make explicit the structure of problems and tasks. A flowchart can also be used to define a process or project to be implemented.

Flow

Lines represent flow of the sequence and direction of a process.

Data

It represents information entering or leaving the system. An input might be an order from a customer. An output can be a product to be delivered.

Terminator

The terminator symbol represents the starting or ending point of the system.

Process

A box indicates some particular operation.

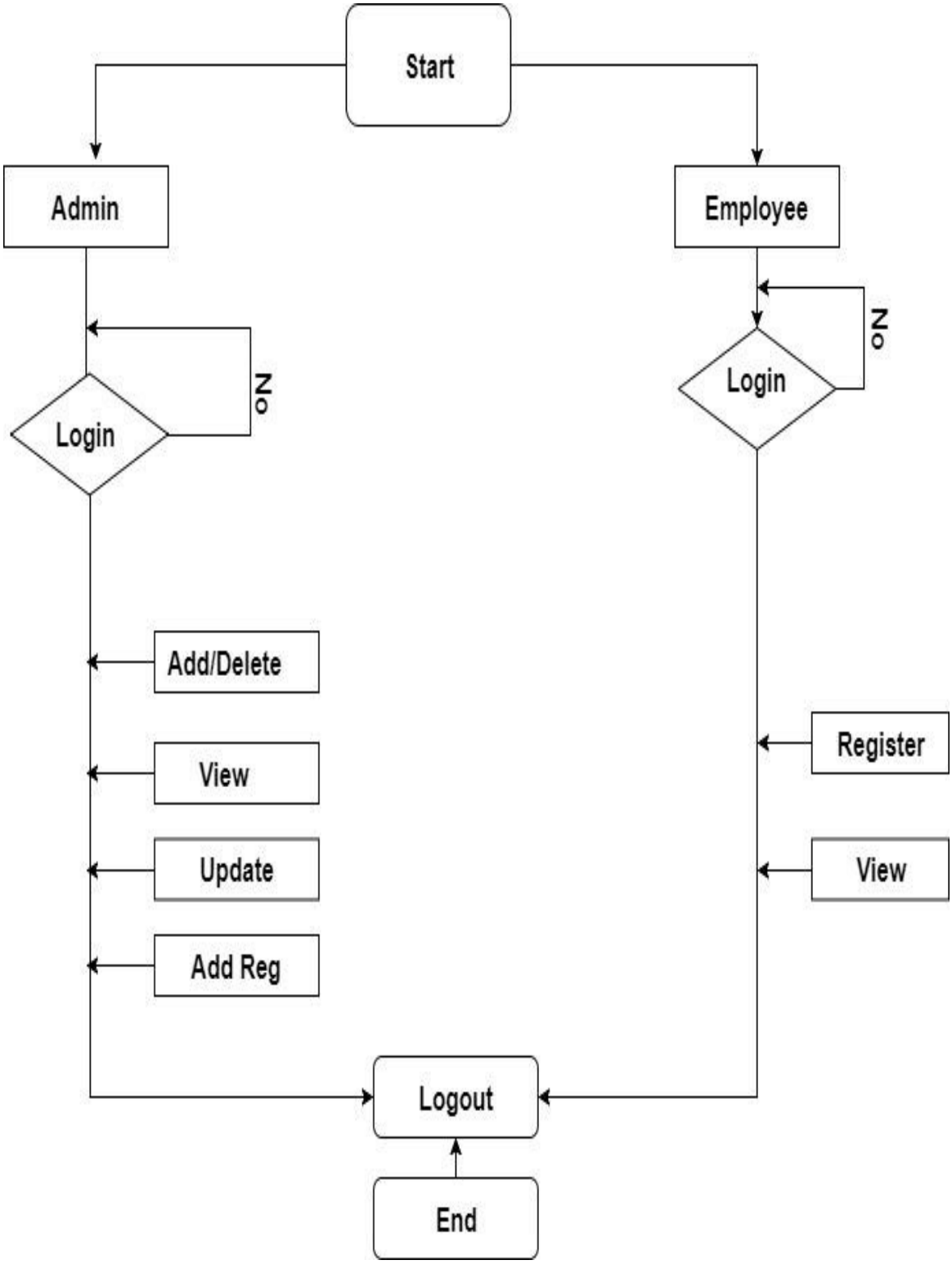


Fig 2.1 Structure

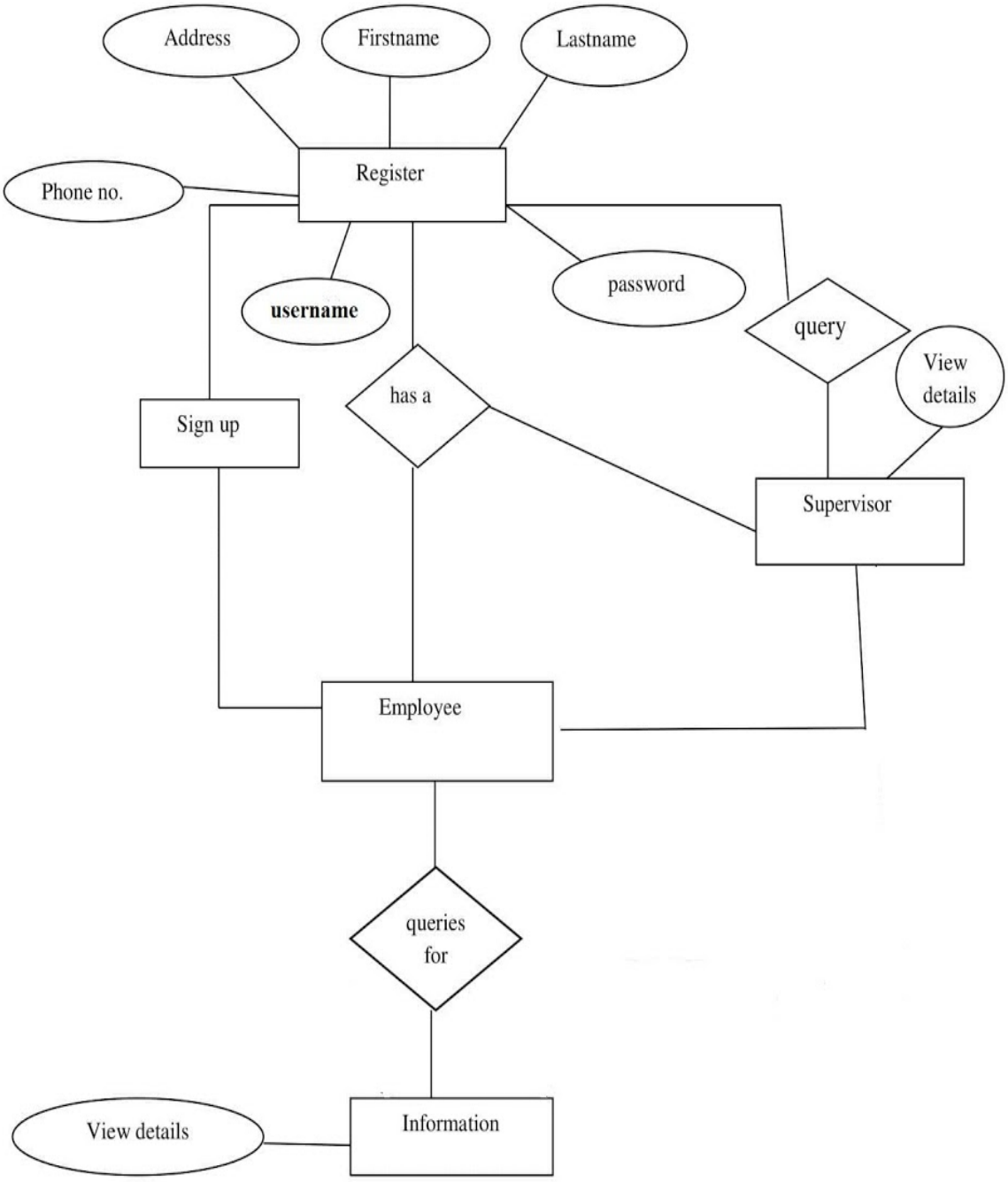


Fig 2.2: Flow chart for EMP system

2.5 DATABASE AND CONSTRAINTS

Data

In simple words data can be facts related to any object in consideration.For example your name, age, height, weight, etc are some data related to you.A picture , image , file , pdf etc can also be considered data.

Database

Database is a systematic collection of data. Databases support storage and manipulation of data. Databases make data management easy.A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information.Database records and files must be organized to allow retrieval of the information. Queries are the main way users retrieve database information. The power of a DBMS comes from its ability to define new relationships from the basic ones given by the tables and to use them to get responses to queries. Typically, the user provides a string of characters, and the computer searches the database for a corresponding sequence and provides the source materials in which those characters appear; a user can request, for example, all records in which the contents of the field for

Data Constraints

Constraints means we enforce some rules to enter the data in a correct formatted way which gives the desired output to the user. This attribute measures the system ability to calculate and generate the desired and the correct output for the user. Here, this system provides both integrity and constraints to the user. In this system, wrong data entry can't be permissible as no fields cannot be left blank, password should be greater than 6 digits etc.

2.6 DATABASE DESIGN AND TABLES

In relational databases, and flat file databases, atable is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows, the cell being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows.

Queries are one of the things that make databases so powerful. A "query" refers to the action of retrieving data from your database. Usually, you will be selective with how much data you want returned. If you have a lot of data in your database, you probably don't want to see everything. More likely, you'll only want to see data that fits a certain criteria.

2.6 Description of Tables used in Employee Management System

Table 2.6.1: Employee registration

S.no	Field name	Data type	Description
1.	Username	varchar	Not null
2.	Email	varchar	Not null
3.	Password	text	Not null
4.	Repeat password	text	Not null
5.	Age	number	Not null
6.	Salary	number	Not null
7.	Qualification	text	Not null
8.	Phone	number	Not null
9.	Address	text	Not null
10.	Date of birth	date	Not null
11.	Date of join	date	Not null

Table 2.6.2: Supervisor login

S.no	Field name	Data type	Description
1.	Username	varchar	varchar
2.	Password	text	text

Table 2.6.3: User login

S.no	Field name	Data type	Description
1.	Username	varchar	varchar
2.	Password	text	text

Table 2.6.4: Employee insertion:

Description: Table to insert the employees by the supervisor in his page.

S.no	Field name	Data type	Description
1.	Username	varchar	Not null
2.	Email	varchar	Not null
3.	Password	text	Not null
4.	Repeat password	text	Not null
5.	Age	number	Not null
6.	Salary	number	Not null
7.	Qualification	text	Not null
8.	Phone	number	Not null
9.	Address	text	Not null
10.	Date of birth	date	Not null
11.	Date of join	date	Not null

3.CODE

The contents in the code are:

1.login

2.log

3.Admin login page

4.Admin view

5.View pending registrations

6.Admin update

7.Insert employee

8.Delete employee

9.Registration

10.Reg

11.Employee logged in page

1.login:

```
<body>

<form action='log.php' meth0od='post'>

<center>

<div class="header">

<h1>Login</h1></div>

<br><br>

<div class="c2">

  User Name :<input type="text" name="username" id="username" ><br><br>

Password :<input type="password" name="password" id="pasword"><br><br>

</div>

<button class="button1 button2">Submit</button>

<a href="register.php">Register?</a>

</center>

</form>

</body>
```

2.log

```
<?php
```

```
 include("sql.php");
 session_start();
function Fix($str) {
 $str = trim($str);
if(get_magic_quotes_gpc()) {
$str = stripslashes($str);
}
return mysql_real_escape_string($str);
}
 $errmsg = array(); //Array to store errors
 $errflag = false; //Error flag
 $username = Fix($_POST['username']);
$password =md5( Fix($_POST['password']));
if($username == '') {
$errmsg[] = 'Username missing';
$errflag = true;
 }
if($password == '') {
$errmsg[] = 'Password missing';
$errflag = true;   }
 if($errflag) {
```

```
 $_SESSION['ERRMSG'] = $errmsg;
echo "Error";
session_write_close();
    }
if($username=='admin'){
header('Location: dispalayuser.php');
}
else{
 $qry = "SELECT * FROM users WHERE username = '$username' AND password =
'md5($password)';";
   $result = mysqli_query($conn,$qry);
  $qr1="SELECT * FROM employee WHERE name='$username'; ";
  $res = mysqli_query($conn,$qr1);
  $res = $res->fetch_assoc();
  $_SESSION['username'] = $username;
  $_SESSION['age'] = $res["age"];
  $_SESSION['salary'] = $res["salary"];
  $_SESSION['qualification'] = $res["qualification"];
  $_SESSION['date_of_birth'] = $res["date_of_birth"];
  $_SESSION['date_of_join'] = $res["date_of_join"];
  $_SESSION['email']=$res["email"];
  $_SESSION['adress']=$res["adress"];
```

```
header('Location: viewuser.php');}?>
```

3.Admin login page

```
<?php
include 'sql.php';
session_start();
?>
<body>
<div class="s">
<center><h2>Displaying Records</h2></center>
<a href="logout.php"><button class="btn" name="logout">logout</button></a>
<a href="pending.php" ><button class="btn1" name="pending">pending
registrations</button></a>
<a href="inserte.php" ><button class="btn " name="logout">insert</button></a><br>
<table style="margin-top:32px;" >
<tr>
<th>ID<th>
<th>Name</th>
<th>Age</th>
<th>Salary</th>
<th>Qualification</th>
<th>address</th>
<th>email</th>
```

```
<th>phone</th>

<th>Date of Birth</th>

<th>Date of Join</th>

<th>View profile</th>

<th>Update</th></tr>

<?php

$q="SELECT * FROM employee WHERE pending='true'";

$query = mysqli_query($conn,$q);

while ($res = mysqli_fetch_array($query)) {

?>

<tr>

<td><?php echo $res['id'] ?></td>

<td><?php echo $res['name'] ?></td>

<td><?php echo $res['age'] ?></td>

<td><?php echo $res['salary'] ?></td>

<td><?php echo $res['qualification'] ?></td>

<td><?php echo $res['adress'] ?></td>

<td><?php echo $res['email'] ?></td>

<td><?php echo $res['phone'] ?></td>

<td><?php echo $res['date_of_birth'] ?></td>

<td><?php echo $res['date_of_join'] ?></td>

<td><a  href="viewuseradmin.php?id=<?php echo $res['id']; ?>">
```

```
<button class="btn">View</button></a></td>

<td><a  href="updatet.php"><button class="btn1">Update</button></a></td></tr>

<?php

    }

    ?>

</table>

</div></body>
```

4.Admin view:

```
<?php

session_start();

error_reporting(0);

include 'sql.php';

$id = $_GET['id'];

$name  = ucfirst($_POST['user']);

$age = $_POST['age'];

$salary = $_POST['salary'];

$qualification = ucfirst($_POST['qualification']);

$email=$_POST['email'];

$phone=$_POST['phone'];

$adress=$_POST['adress'];

$day = $_POST['day'];

$date_of_birth = $_POST['day2'];

$q="select * from employee where id = $id";

$query = mysqli_query($conn,$q);

$res=mysqli_fetch_array($query);

?>

<body>

<div class="container">

<div class="col-lg-6 m-auto">
```

```
<form method="post">

<br><div>

<div class="card-header bg-dark">

<center><h2 >Displaying Employee Details</h2></center>

</div><br>

<input type="hidden" name="id" value="<?php echo $res['id']; ?>">

<table border=2 align="center">

<tr>

<td align="center" colspan=9></td></tr>

<tr>

<td align="center">Name of the employee</td>

<td  align="center" ><?php echo $res['name']; ?></td></tr>

<tr>

<td align="center">Age of the employee</td>

<td align="center" ><?php echo $res['age']; ?></td>

<tr>

<td align="center">Salary of the employee</td>

<td align="center"><?php echo $res['salary']; ?></td></tr

<td align="center">Qualification of the employee</td></tr>

<tr>

<td align="center">Mailid of the employee</td>

<td align="center"><?php echo $res['email']; ?></td></tr>
```

```
<tr>
<td align="center">Date of birth of the employee</td>
<td align="center"><?php echo $res['date_of_birth']; ?></td></tr>
<tr>
<td align="center">Date of join of the employee</td>
<td align="center"><?php echo $res['date_of_join']; ?></td></tr>
<tr>
<td align="center">Address of join of the employee</td>
<td align="center"><?php echo $res['adress']; ?></td><?tr>
<table>
<a href="dispalayuser.php"><input type="button" name="" value="Back to records" class="btn btn-primary col-lg-12"></a>
                    </div>
          </form>
<?php session_write_close();?>
</div>
</body>
```

5.<u>View pending registrations</u>:

```
    <body>
    <center>
    <div class="container">
          <div class="col-lg-12"><br>
```

```
<div class="row">

<h1>Displaying Registered Records</h1>

<a href="logout.php" class="col-lg-3"><button class="btn1"
name="logout">logout</button></a>

<a href="dispalayuser.php" ><button class="btn1" name="back to records">Back to
records</button></a>

</div><br>

<table class="table table-stripped table-hover table-bordered">

<tr class="text-dark">

<th>ID</th>

<th>Name</th>

<th>Age</th>

<th>Salary</th>

<th>Qualification</th>

<th>Address</th>

<th>Email</th>

<th>Phone</th>

<th>Date of Birth</th>

<th>Date of Join</th>

<th>Accept</th>

<th>Reject</th>

</tr>
```

```
<?php
 $qry="SELECT * FROM employee WHERE pending ='false'";
 $result = mysqli_query($conn,$qry);
 $num  = mysqli_num_rows($result);
while($num > 0)
 {
 $row = mysqli_fetch_array($result);
 ?>
<tr>
      <th><?php echo $row['id'] ?></th>
      <th><?php echo $row['name'] ?></th>
      <th><?php echo $row['age'] ?></th>
      <th><?php echo $row['salary'] ?></th>
      <th><?php echo $row['qualification'] ?></th>
<th><?php echo $row['adress'] ?></th>
<th><?php echo $row['email'] ?></th>
<th><?php echo $row['phone'] ?></th>
      <th><?php echo $row['date_of_birth'] ?></th>
      <th><?php echo $row['date_of_join'] ?></th>
<th><button name="accept" type="button" class="btn1"><a
href="updateUser.php?id=<?php echo
$row['id'];?>&op=acc">Accept</a></button></th>
```

```
<th><button name="reject" type="button" class="btn2"><a
href="updateUser.php?id=<?php echo
$row['id'];?>&op=del">Reject</a></button></th></tr>

<?php

 $num = $num - 1;

   }

?>

      </table>          </div></div></center>

</body>
```

6.<u>Admin update</u>:

```
<?php

session_start();

error_reporting(0);

include 'sqls.php';

$Id = $_GET['Id'];

$name  = ucfirst($_POST['name']);

$age = $_POST['age'];

$salary = $_POST['salary'];

$qualification = ucfirst($_POST['qualification']);

$email=$_POST['email'];

$phone=$_POST['phone'];

$adress=$_POST['adress'];
```

```
$day = $_POST['day'];

$date_of_birth = $_POST['day2'];

$q="select * from employee where Id = $Id";

$query = mysqli_query($conn,$q);

$res=mysqli_fetch_array($query);?>

<body>

<div class="container">

<div class="col-lg-6 m-auto">

      <form method="post">

      <div class="card-header bg-dark">

      <center><h2>Displaying Employee Details</h2></center></div><br>

      <input type="hidden" name="Id" value="<?php echo $res['Id']; ?>">

      <label>Name</label>

      <input type="text" name="name" class="form-control" value="<?php echo
$res['name']; ?>" required><br>

      <label>age</label>

      <input type="text" name="age" class="form-control" value="<?php echo
$res['age']; ?>" required pattern="[0-9]{1,15}"

  title="this field accepts only numbers"><br>

      <label>salary</label>

      <input type="text" name="salary" class="form-control" value="<?php echo
$res['salary']; ?>" required pattern="[0-9]{1,15}"

  title="this field accepts only numbers"><br>
```

```
<label>qualification</label>

<input type="text" name="qualification" class="form-control" value="<?php echo
$res['qualification']; ?>" required><br>

<label>email</label>

<input type="text" name="email" class="form-control" value="<?php echo
$res['email'];?>" required><br>

<label>adress</label><input type="text" name="adress" class="form-control"
value="<?php echo $res['adress'];?>" required><br>

<label>phone</label>

<input type="text" name="phone" class="form-control"value="<?php echo
$res['phone'];?>" required pattern="[0-9]{10}"><br>

<label>date of Birth</label>

<input type="date" name="day2" class="form-control"

value="<?php echo $res['date_of_birth'];?>"placeholder="date-month-year"
required><br>

<label>date of Joining</label>

<input type="date" name="day" class="form-control" value="<?php echo
$res['date_of_join']; ?>" placeholder="date-month-year" required><br>

<div class="row">

<div class="col-md-3"><button class="btn " name="done">update</button></div>

<div class="col-md-3"><input type="button" class="btn " name="delete"
value="Delete" onClick "deleteme(<?php $_SESSION['Id'] = $Id; ?>)"></div>

<script type="text/javascript">

function deleteme(del id)
```

```
{
if(confirm("Are you sure you want to delete ?"))
 {
      window.location.href="deletete.php";}}
      </script> </div><br>
         
      <a href="dispalayuser.php"><input type="button" name="" value="Back to records" class="btn"></a><?div></form></div>
<?php
if(isset($_POST["done"]))
{
$q2="UPDATE employee SET name='$name',age='$age',salary='$salary' , qualification='$qualification'
,email='$email',phone='$phone',adress='$adress',date_of_birth = '$date_of_birth' , date_of_join = '$day' WHERE Id = $Id";
mysqli_query($conn,$q2);
header("refresh:0");
}
if(isset($_POST['delete']))
{
      $_SESSION['id'] = $Id;
      header("location:deletete.php");}
?>
```

```
</body>
```

7.Insert employee:

```
<?php
include 'sql.php';
if(isset($_POST['done'])){
        $name  = ucfirst($_POST['user']);
        $age = $_POST['age'];
        $salary = $_POST['salary'];
        $qualification = ucfirst($_POST['qualification']);
        $email=$_POST['email'];
        $phone=$_POST['phone'];
 $adress=$_POST['adress'];
$day = $_POST['day1'];
$date_of_birth = $_POST['day2'];
        $q="INSERT INTO `employee`( `name`, `age`, `salary`, `qualification`,`email`,
`phone`, `adress`, `date_of_birth`, `date_of_join`) VALUES
('$name','$age','$salary','$qualification', '$email',
'$phone','$adress','$date_of_birth',NOW(),)";
$query = mysqli_query($conn,$q);
echo "UPDATED SUCCESSFULLY!";
header("inserte.php");}
?>
```

```
<body>
<div class="container">
<center><h2>ADD AN EMPLOYEE</h2></center>
<form method="post">
        <label>Name</label>
        <input type="text" name="user" class="form-control" required><br>
        <label>age</label>
        <input type="text" name="age" class="form-control" required
pattern="[0-9]{1,15}"
 title="this field accepts only numbers"><br>
        <label>salary</label>
        <input type="text" name="salary" class="form-control" required
pattern="[0-9]{1,15}"
  title="this field accepts only numbers"><br>
        <label>qualification</label>
        <input type="text" name="qualification" class="form-control" required><br>
        <label>email</label>
        <input type="text" name="email" class="form-control" required><br>
        <label>adress</label>
        <input type="text" name="adress" class="form-control" required><br>
        <label>phone</label>
        <input type="text" name="phone" class="form-control" required pattern="[0-
9]{10}"><br>
```

<td width ="500" align="left">Date of Birth :</td>

<td><input type="date" name="day2" class="form-control" placeholder="date" required pattern="[0-9]{1,2}" title="this field accepts only numbers and two characters">
</td>

<label>date of Joining</label>

<td><input type="date" name="day1" class="form-control" placeholder="date" required pattern="[0-9]{1,2}" title="this field accepts only numbers and two characters">
</td>

<button class="btn btn-success col-lg-12" name="done">Add</button>

<input type="button" name="" value="Back to records" class="btn btn-danger col-lg-12"> <?div></form></script>

</body>

8.Delete employee:

<?php

session_start();

include 'sql.php';

$id = $_SESSION['id'] ;

$q = " DELETE FROM employee WHERE id = '$id' ";

mysqli_query($conn,$q);

header("location:dispalayuser.php");

?>

9.Registration:

```
<?php
 session_start(); //Start the session
if(isset($_SESSION['ERRMSG'])&& is_array($_SESSION['ERRMSG']) &&
count($_SESSION['ERRMSG']) >0 ) {
$err = "<table>";
foreach($_SESSION['ERRMSG'] as $msg) {
 $err .= "<tr><td>" . $msg . "</td></tr>";     }
 $err .= "</table>";
 unset($_SESSION['ERRMSG']);   }
 ?>
 <body>
 <div class="container">
 <center><h2>Registration Form</h2></center>
 <form action='reg.php' method='post'>
 <table align="center">
 <tr>
 <td><?php //echo $err; ?></td></tr>
 <tr>
 <td>Username</td>
 <td><input type='text' name='username'></td><?tr>
 <tr>
 <td>Email</td>
```

```
<td><input type='text' name='email'></td></tr>

<tr>

<td>Password</td>

<td><input type='password' name='password'></td><?tr>

<tr>

<td>Repeat Password</td>

<td><input type='password' name='rpassword'></td></tr>

<tr>

<td>Age</td>

<td><input type='number' name='age'></td></tr>

<tr>

<td>Salary</td>

<td><input type='number' name='salary'></td></tr>

<tr>

<td>Qualification</td>

<td><input type='text' name='qualification'></td></tr>

<tr>

<td>Phone</td>

<td><input type='number' name='phone'></td>

</tr>

<tr>

<td>Address</td>
```

```
<td><input type= 'text' name='adress'></td>
</tr>
<tr>
<td>Date of birth</td>
<td><input type='date' name='date_of_birth'></td>
</tr>
<tr>
<td>Date of join</td>
<td><input type='date' name='date_of_join'></td>
</tr>
<tr>
<td><button type="submit" class="btn" >Register</button>
</td>
<td><a href="login.php">I have an account</a></td>
</tr>
</table></form></div></body>
```

10.Reg:

```
<?php
include("sql.php");
session_start();
function Fix($str) {
$str = @trim($str);
```

```
    if(get_magic_quotes_gpc()) {
   $str = stripslashes($str);
  }
      return mysql_real_escape_string($str);
   }
  $errmsg = array();
  $errflag = false;
   $UID = "12323543534523453451465685454";
   $name = Fix($_POST['username']);
   $email = $_POST['email'];
   $password = Fix($_POST['password']);
   $rpassword = Fix($_POST['rpassword']);
   $add=$_POST['adress'];
   $sal=$_POST['salary'];
   $qal=$_POST['qualification'];
   $ph=$_POST['phone'];
$dob=$_POST['date_of_birth'];
$doj=$_POST['date_of_join'];
$age = $_POST['age'];
if($name == '') {
$errmsg[] = 'Username missing';
$errflag = true;   }
```

```
if($password == '') {
$errmsg[] = 'Password missing';
$errflag = true; }
if($rpassword == '') {
 $errmsg[] = 'Repeated password missing';
$errflag = true;
  }
if(strcmp($password, $rpassword) != 0 ) {
 $errmsg[] = 'Passwords do not match';
$errflag = true;
  }
if($name != '') {
$qry = "SELECT * FROM `employee` WHERE `username` = '$name'";
$result = mysql_query($qry);
if($result) {
 if(mysql_num_rows($result) > 0) {
$errmsg[] = 'Username already in use';
$errflag = true;
   }
mysql_free_result($result);}
 if($errflag) {
$_SESSION['ERRMSG'] = $errmsg;
```

```
 }
$status = "false";
 $pasd=md5($password);
 $q="INSERT INTO `users` VALUES(NULL,'$name','$email','$pasd')";
 $res = mysqli_query($conn,$q);
 $qry="INSERTINTO`employee’VALUES(NULL,'$name','$age','$sal','$qal','$email','
ph','$add','$dob','$doj','$status')";
$result = mysqli_query($conn,$qry);
if($result != FALSE)
{
echo "Thank you for registering, " . $name . ". Please login <a
href=\"login.php\">Here</a>";
}
else
{
die('There was an error, try again later");
}
?>
```

11.<u>Employee loggedin page</u>:

```
<?php
session_start();
error_reporting(0);
```

```
include 'sql.php';

$id = $_GET['id'];

$name  = ucfirst($_POST['user']);

$age = $_POST['age'];

$salary = $_POST['salary'];

$qualification = ucfirst($_POST['qualification']);

$email=$_POST['email'];

$phone=$_POST['phone'];

$adress=$_POST['adress'];

$day = $_POST['day'];

$date_of_birth = $_POST['day2'];

$q="select * from employee where id = $id";

$query = mysqli_query($conn,$q);

$res=mysqli_fetch_array($query);

?>
```

4.SCREENSHOTS:

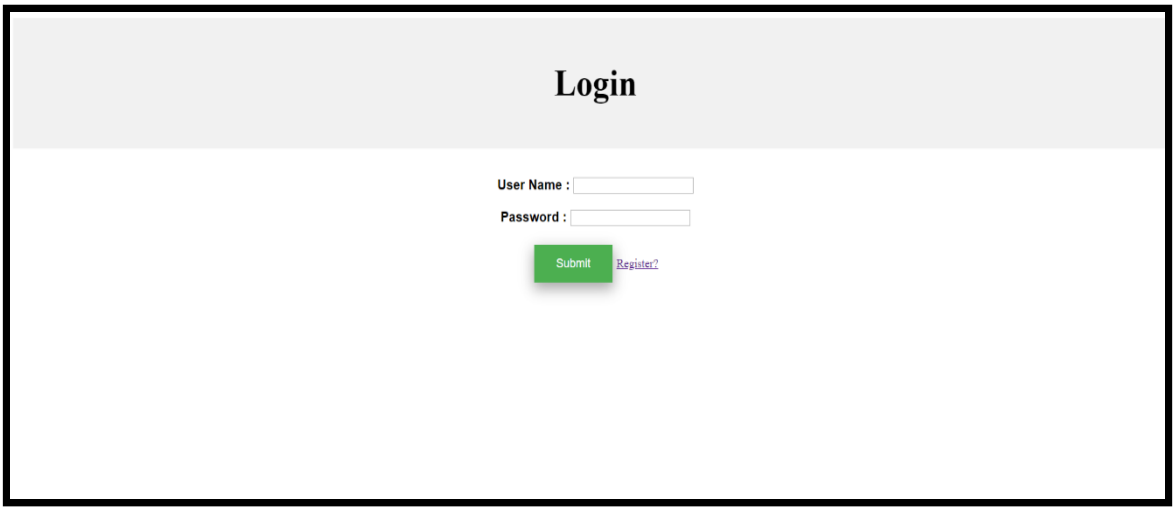
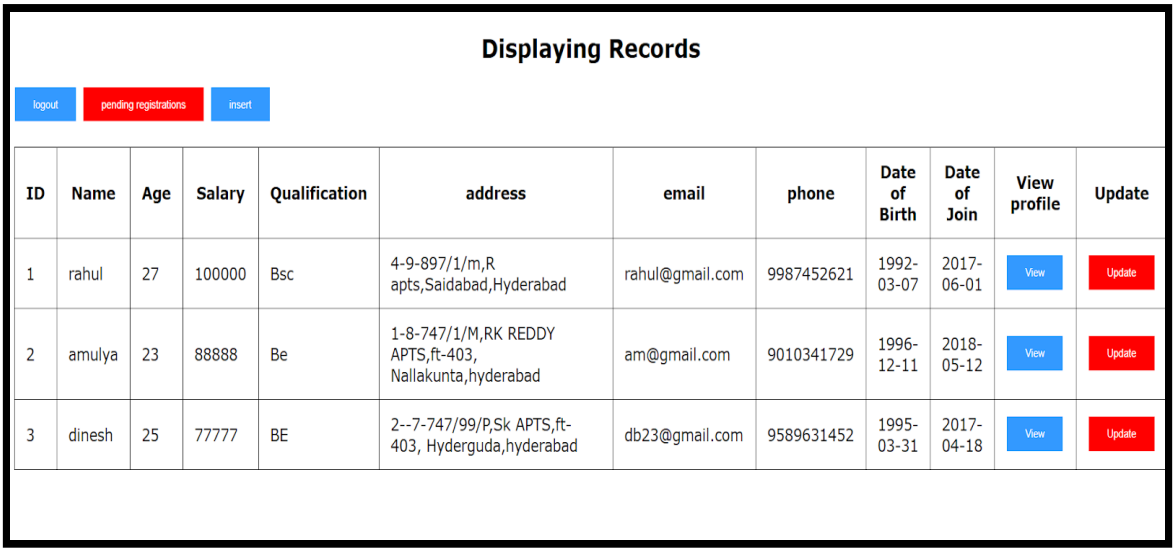


Fig 4.1:login page



ID	Name	Age	Salary	Qualification	address	email	phone	Date of Birth	Date of Join	View profile	Update
1	rahul	27	100000	Bsc	4-9-897/1/m,R apts,Saidabad,Hyderabad	rahul@gmail.com	9987452621	1992-03-07	2017-06-01	View	Update
2	amulya	23	88888	Be	1-8-747/1/M,RK REDDY APTS,ft-403, Nallakunta,hyderabad	am@gmail.com	9010341729	1996-12-11	2018-05-12	View	Update
3	dinesh	25	77777	BE	2--7-747/99/P,Sk APTS,ft-403, Hyderguda,hyderabad	db23@gmail.com	9589631452	1995-03-31	2017-04-18	View	Update

Fig 4.2: Admin logged-in page

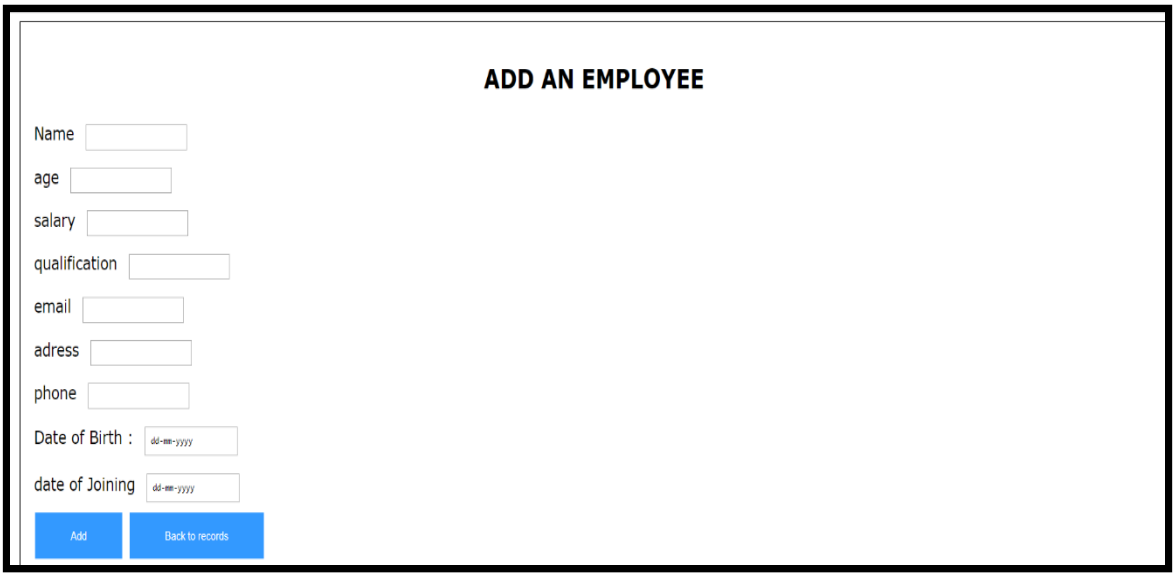


Fig 4.3: Employee insertion by Supervisor

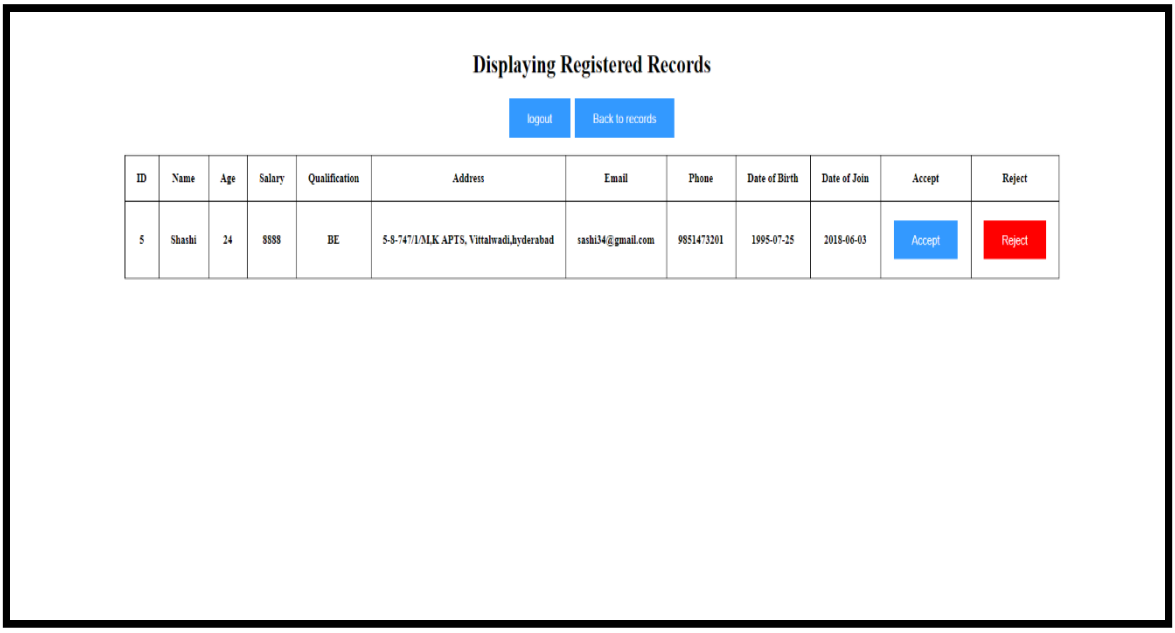


Fig 4.4: Pending registrations

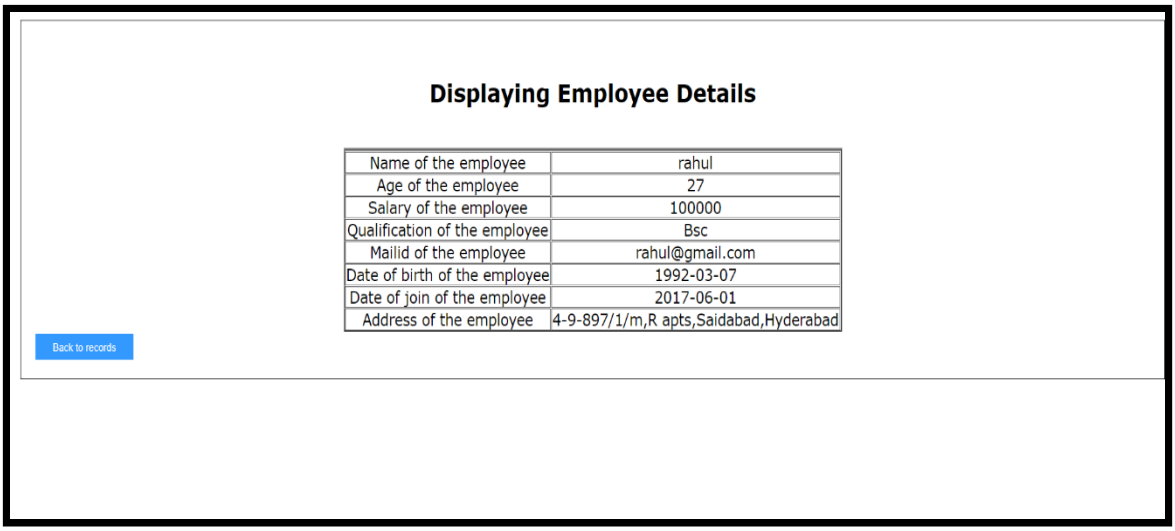


Fig 4.5: Admin view details

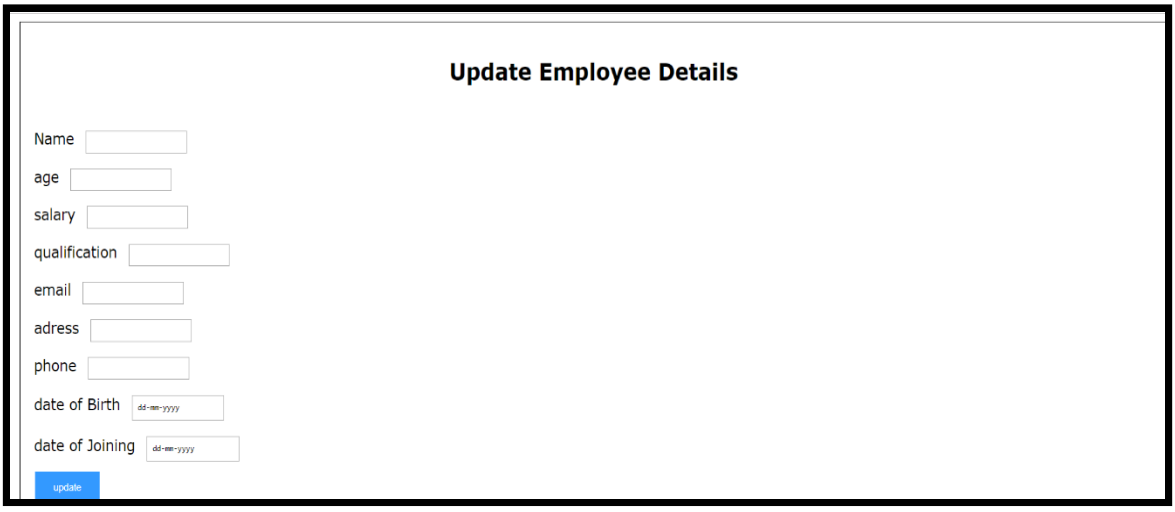


Fig 4.6: Update employee details

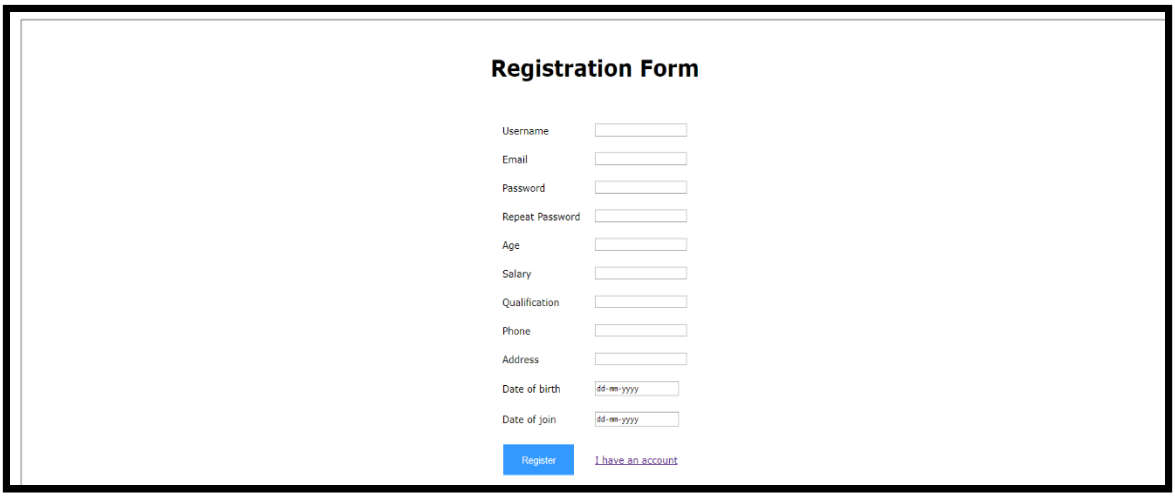


Fig 4.6: Registration form

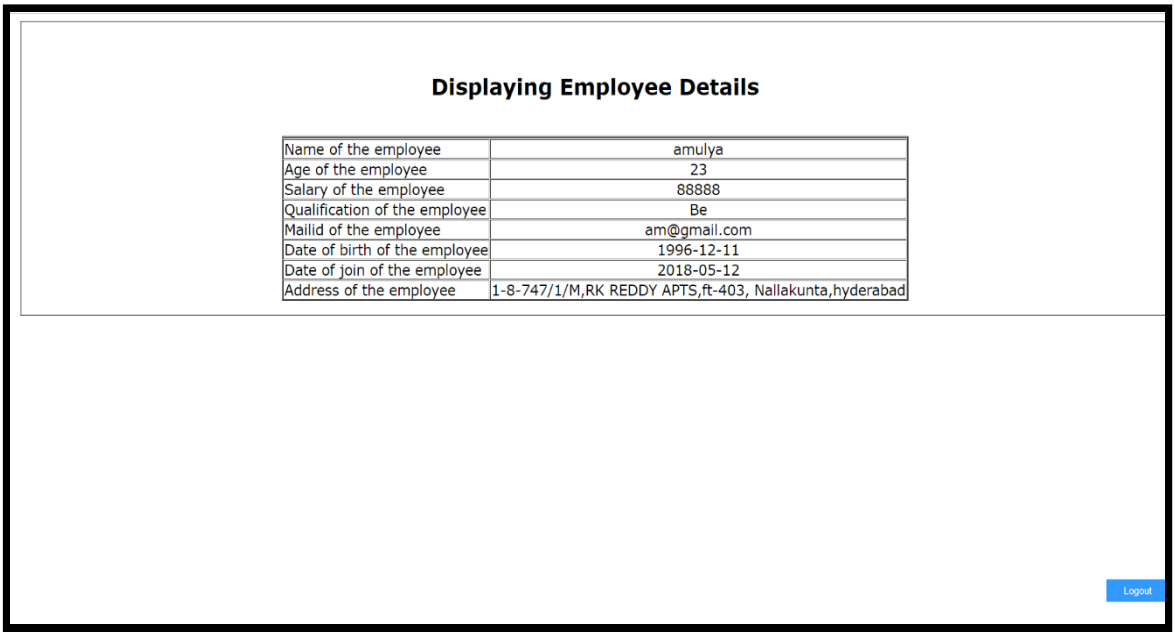


Fig 4.7: Employee logged-in page

5. CONCLUSION

Overall, the system is useful for all the users to maintain information at various levels. It connects supervisor and employee and thus easy to maintain. Now supervisor can easily set the employees without having a person to send to employees. It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for us as it provided practical knowledge of not only programming in PHP,HTML and XAMPP server based application, but also about all handling procedure related with Advance and new technology. We also learned to customize WebPages using CSS type of styling It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

6. FUTURE ENHANCEMENT

The developed system is flexible and changes can be made easily. The system is developed with an insight into the necessary modification that may be required in the future. Hence the system can be maintained successfully without much rework. Any product despite of its meticulous design and features needs enhancement with time.EMS being no exception needs active enhancement of features and functionality. Following features are proposed to be implemented in future to make the product more feature rich. Moreover, it is just a beginning; further the website can be enhanced by adding following facilities in the existing software:

- **Email:** In each organization, there is always a need of efficient paperless, secure,and private communication medium that has the retention capabilities. We are proposing to add Intranet messaging facility to fill this requirement as an independent module in future release.
- **Template Based Look & Feel:** Being a product, it is proposed to be implemented at the site of different clients. Each client has different set of preferences for look feel of the application. To minimize the customization process, look and feel of the application is proposed to be based on templates in future releases.
- **Complete Employee Record Management:** In future we propose to do all record management online or web based automated system which only requires officially needed paperwork not more than that.
- **Attendance through email:** A complete email module for sending the attendance to respective employees which is developed separately can be integrated to the system.

7.REFERENCES

World Wide Web:

1) https://www.w3schools.com/
2) https://www.w3.org/Style/CSS/Overview.en.html
3) https://stackoverflow.com/
4) https://www.plus2net.com/

Books:

1)SQL Queries for Mere Mortals

Michael James Hernandez, John Viescas

2)Learning Php, Mysql & Javascript ,Robin Nixon,Shroff Publishers.