A

MINI PROJECT REPORT ON

Placement News Management System

Prepared By-

Kadam Vivek (70)

Kolhe Yash (78)

Pachore Prashant (92)

(SY-A B-Tech. Computer)

SAVITRIBAIPHULE PUNE UNIVERSITY

In the academic year 2022-23

Department of Computer Engineering

Sanjivani Rural Education Society's

Sanjivani College of Engineering

Kopargaon - 423 603.

Sanjivani college of Engineering, Kopargoan

CERTIFICATE

This is to certify that

Kadam Vivek (70)
Kolhe Yash (78)
Pachore Prashant (92)

Has successfully completed MPL report on

Placement News Management System

Towords the partial fulfilment of
Bachelor's Degree In Computer Engineering
During the Academic year 2022-23

Prof. I.B.Tirse
[Guide]

Dr .D. B. Kshirsagar
[H.O.D Comp Eng]

Dr. A. G. Thakur
[Director]

ACKNOWLEDGEMENT

The entire session of MPL completion phase so far was a great experience providing me with great insight and innovation into learning various concepts and achievement of it. As is rightly said, for the successful completion of any work, people are the most important asset my seminar would not be materialized without the cooperation of many of the people involved.

First and foremost,I am very thankful to seminar guide **Prof.I.B.Tirse** and seminar coordinator **Prof. T.Bhaskar** for their leading guidance and sincere efforts in finalizing this topic. They took deep interest in correcting the minor mistakes and guided us through my journey so far. Also they has been persistent source of inspiration for me.

I am also very thankful of **Dr. D. B. Kshirsagar**, Head of Dept. of Computer Engineering for the symmetric guidance and providing necessary facilities and I Express deep gratitude to all the staff members and our department's technical Staff for providing me needed help.

Date : / /2023

Place : Kopargoan

Roll No. 70,78,92

Div. B

S.Y. B.Tech

Contents

Sr. No	Title	Page No.
1	Introduction	4
2	Proposed System	5
3	Scope and Objective	5
4	Literature Review	6
5	System Overview	9
7	Sytem Snapshots	20
8	Project Program	23
9	Result Analysis	39
10	Conclusion	40

INTRODUCTION

The Placement Management System is a software solution designed to simplify the process of connecting college students with potential employers for internships and job placements. The system is designed to provide a platform for students to create a profile, search and apply for relevant opportunities, and receive notifications about new job postings. The system also includes a login system that allows administrators to manage user accounts and job postings. The system is designed to be user-friendly, intuitive, and efficient, making the job search process easier for college students.

Currently, many placement officers post job opportunities on WhatsApp groups for students to access. However, these posts often get lost among the numerous messages on the group, making it difficult for students to keep track of job opportunities. As a result, students often miss out on potentially valuable job opportunities that could help them build their careers.

Proposed System

The proposed Placement Management System addresses this shortcoming of the current system by providing a centralized platform for job postings. The system will allow placement officers to post job opportunities on the platform, and students can easily access these postings through their profiles. The system will use advanced algorithms to match job seekers with potential employers based on their skills and experience, making the job search process more efficient and relevant to the students.

The system will also include a notification system that will alert students to new job postings that match their profile, ensuring that they never miss out on a job opportunity. The system will have a user-friendly interface that is easy to navigate and will provide all the necessary information about the job posting, such as job description, required qualifications, and application process.

Scope

edit
edi
t

- Centralized platform: A system that provides a centralized platform for managing placement news and updates.

- Real-time updates: A system that allows placement officers to post real-time updates about job vacancies, interview schedules, and other relevant information.

- Customizable notifications: A system that enables students to customize notifications according to their interests and skills, so they receive relevant job updates.

Objective

- Centralized platform: A system that provides a centralized platform for managing placement news and updates.

- Real-time updates: A system that allows placement officers to post real-time updates about job vacancies, interview schedules, and other relevant information.

- Customizable notifications: A system that enables students to customize notifications according to their interests and skills, so they receive relevant job updates.

literature review 01

COLLEGE PLACEMENT MANAGEMENT SYSTEM

Farheen Taqi Rizvi1 , Naushin Arif Khan2 , Saurabh Sanjay Upadhyay3 , Prof. Sonali Suryawanshi4 1, 2, 3, 4Dept. Of Computer Engineering, Rizvi College of Engineering, Mumbai, India

- The paper discusses the challenges faced by traditional placement systems in colleges, including insufficient details, less security, and problems with manual working. The authors propose a Placement Management System website that aims to address these challenges by providing a quick and efficient platform for placement management.
- The website allows end users to register online through their CMSys account, which is a website created for college management purposes such as attendance and term test marks. Students can read and apply for the company of their choice and receive frequent updates regarding the placements from the college TPO.
- One of the key advantages of the proposed system is that there is no chance of missing any placement opportunity updates. The college placement officers will not have to separately collect information of every student, as it will be automatically updated when the student registers. The project is a website that can be easily accessed through mobile devices.
- This paper highlights the need for an automated and efficient placement management system in colleges, which can overcome the limitations of traditional systems. The proposed system can provide benefits to both students and college placement officers by enabling them to manage placement-related tasks in a more organized and streamlined manner. Overall, this paper provides a valuable insight into the benefits of an automated placement management system in colleges.

literature review 02

Placement Management System

Maryam Sayyed, Faiza Umatiya,Seemab Zehera, Prof. Shiburaj Pappu 1Student, 2 Student, 3 Student, 4 Assistant professor 1Computer engineering dept. RCOE, 1Rizvi College of Engineering, Mumbai, India

- Automated systems have become increasingly popular in various industries, including education. The Placement Management System project aims to automate the placement process in colleges by developing a application that can be used by both students and placement officers. This system provides a single platform for all placement-related tasks, eliminating the need for students to visit multiple platforms to find job opportunities and for placement officers to manage data across different systems.
- Previous research has shown the benefits of automated systems in various industries. For example, in a study by Suryanarayanan et al. (2015), it was found that an automated placement system significantly reduced the time and effort required for placement officers to manage data and coordinate with companies. Another study by Ahmad et al. (2018) found that an automated job recommendation system improved the job search process for job seekers, reducing the time and effort required to find relevant job opportunities.
- In conclusion, the Placement Management System project aims to automate the placement process in colleges and provide an efficient and effective platform for students and placement officers. By incorporating features such as personalized job recommendations and email alerts, the system aims to simplify the job search process and improve communication between students and placement officers.

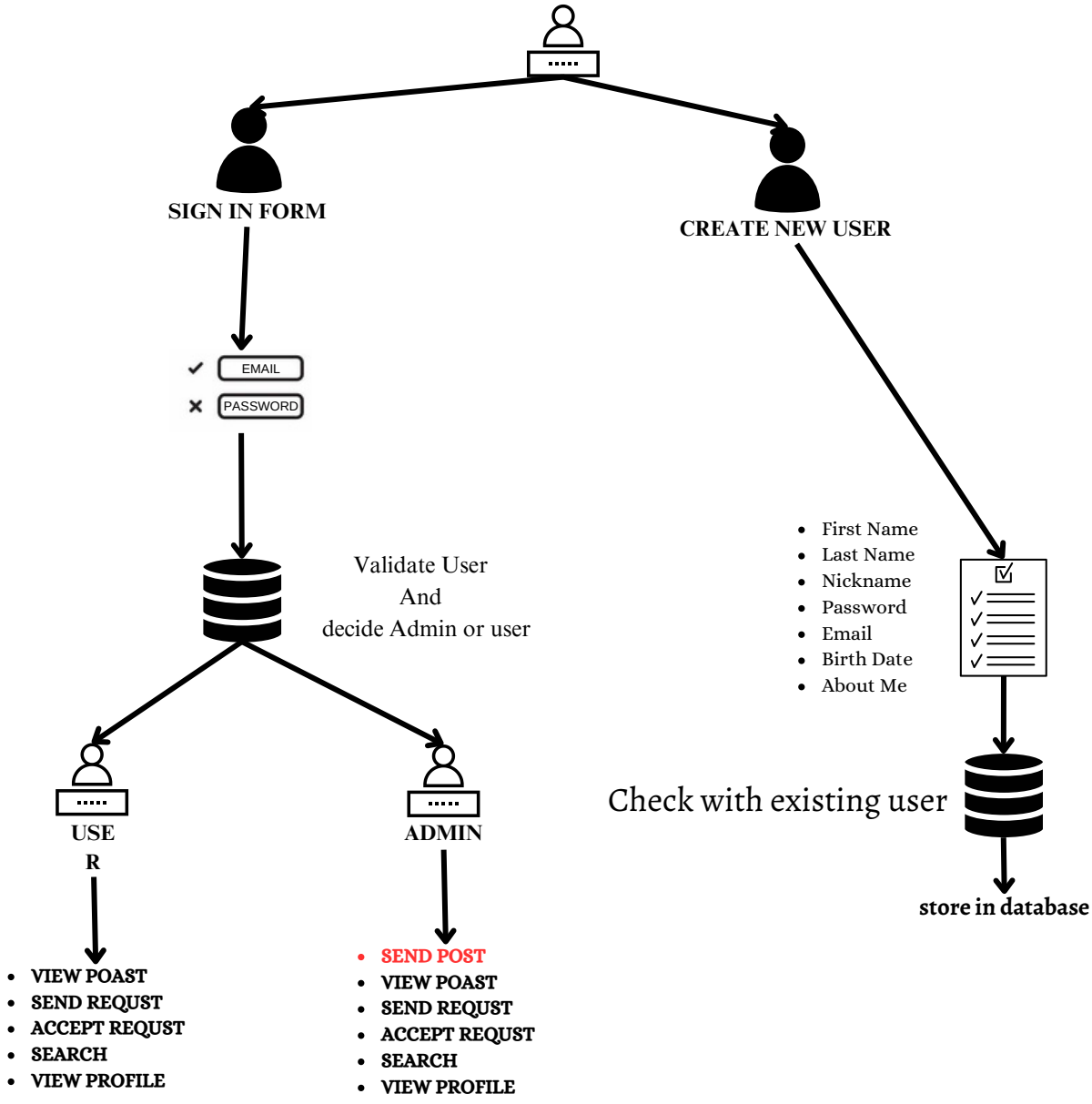
literature review 03

Placement Management System for Campus Recruitment

Ajeena Sunny1 Aneena Felix1 Angelin Saji1 Christina Sebastian1 Praseetha V.M2 1UG Student, Dept. of CSE, SJCET, Palai, Kerala, India 2Associate Professor, Dept. of CSE, SJCET, Palai, Kerala, India

- The traditional manual placement management system in colleges requires a lot of manpower and time, making it a cumbersome process for both students and placement officers. With the development of a web portal using Laravel framework, this project aims to streamline the process by providing a centralized platform for managing placements. This platform can be accessed by students and placement officers with proper login credentials, and it offers features such as viewing personal and academic information, applying for companies, and receiving updates about placement opportunities.
- The use of technology in placement management not only reduces the burden of manual work but also provides benefits such as increased efficiency, transparency, and accuracy. The system also enables placement officers to easily manage student information and generate lists of eligible candidates for companies. By automating the placement process, this web portal can help colleges save time and resources while providing better placement opportunities for their students.
- Overall, the development of a web portal for placement management in colleges using Laravel framework is a step towards modernizing the education system and improving the prospects of students. The system's features and benefits make it an attractive option for colleges looking to enhance their placement process and provide better opportunities for their students.

System Overview



Module 1: SIGN IN FORM

1. **Email Field:**

The email field is typically used to collect the user's email address. This field can be implemented using HTML input type "email" to validate that the user has entered a valid email address. You can also add placeholder text that provides examples of the correct format for an email address.

2. **Password Field:**

The password field is used to collect the user's password. This field should be implemented using HTML input type "password" to hide the characters entered by the user. Additionally, you can add password strength validation to ensure that the password meets certain criteria (e.g., minimum length, use of special characters).

3. Sign-in Button:

The sign-in button is the final element of the sign-in form module. This button should be implemented using HTML input type "submit" and should be labeled with a clear and concise call to action, such as "Sign In" or "Log In."

4. **Error Messaging:**

Finally, it is important to provide clear and concise error messaging to the user in case of any errors or invalid input. You can use HTML5 validation to ensure that users enter valid input, and provide error messages in the form of pop-ups or inline messaging.

Module 2: CREATE NEW USER

1.Create New User:

This module should allow new users to create an account by filling in the required fields, including **First Name, Last Name, class, Password, Email, Birth Date, and About Me**. You can use HTML input types to validate that users enter the correct information in each field.

2.**Check with Existing User:**

To ensure that there are no duplicate accounts, you can add a check to see if the user already exists in the database before creating a new account. This can be done using server-side scripting languages such as PHP. If the user already exists, you can display an error message asking the user to log in instead of creating a new account.

3.**Store in Database:**

Once the user has filled in all the required fields and their account has been confirmed as unique, their information can be stored in a database. You can use a database management system such as MySQL or PostgreSQL to store user information securely.

Module 3: USER

1.View Posts:

This module should allow users to view posts made by other users on the platform. You can use server-side scripting languages such as PHP or Python to fetch posts from the database and display them in a user-friendly format.

2.**Send and Accept Requests:**

Users should be able to send friend requests to other users on the platform. Once a request has been sent, the recipient should be able to accept or reject the request. You can use server-side scripting languages to handle the logic for sending and accepting/rejecting requests.

3.**Search:**

Users should be able to search for other users on the platform using keywords such as first name, last name, or nickname. You can use server-side scripting languages to handle the search logic and display search results in a user-friendly format.

4.**View Profile:**

Users should be able to view other users' profiles on the platform. You can use server-side scripting languages to fetch profile information from the database and display it in a user-friendly format.

5.Manage User Account:

Users should be able to manage their own user account, including updating their profile information, changing their password, and deleting their account. You can use server-side scripting languages to handle the logic for updating and deleting user accounts.

By incorporating these elements, you can create a user-friendly module that allows users to view and interact with content on the platform, as well as manage their own user account.

Module 4: ADMIN

The admin should be able to send posts to all users on the platform. These posts will be visible to all users except the admin who sent the post. Users on the platform will not have the ability to create posts themselves.

To implement this feature, you can create a form that allows the admin to enter the content of the post they wish to send. This form can be handled with server-side scripting languages such as PHP or Python to store the post in the database and send it to all users on the platform.

When a user logs in to their account, they will be able to view the posts created by the admin. This can be done by fetching the posts from the database using server-side scripting languages and displaying them on the user's dashboard or profile page.

It is important to note that users on the platform will not have the ability to create their own posts, as this feature is only available to the admin user.

Process Steps

1.User Registration:

a. The student accesses the application or website.

b. The student clicks on the "Sign Up" button.

c. The student fills out the registration form with details like name, email, and hometown.

d. The student submits the registration form.

e. The system validates the information and creates a new student account.

2.User Login:

a. The student accesses the application or website.

b. The student clicks on the "Login" button.

c. The student enters their login credentials (email and password).

d. The student submits the login form.

e. The system verifies the credentials and logs the student into their account.

3.Posting Option:

a. After logging in, the student sees a "Post" option in the navigation menu.

b. The student clicks on the "Post" option.

c. The student can enter the desired information or news related to the placement.

d. The student submits the post.

e. The system stores the post in the database.

4.Viewing Posts:

a. After logging in, both students and the teacher can see the posts on their respective dashboards.

b. The student or teacher can click on the "Posts" section.

c. The system retrieves the posts from the database and displays them on the screen.

5.Admin Posting:

a. The teacher (admin) accesses the application or website.

b. The teacher logs in using their credentials.

c. The teacher sees an additional "Admin" option in the navigation menu.

d. The teacher clicks on the "Admin" option.

e. The teacher can enter the desired information or news related to the placement.

f. The teacher submits the post.

g. The system stores the post in the database.

6.Restrictions on Student Posting:

a. The system ensures that students do not have the ability to post any content except for viewing posts.

7.Sending and Accepting Requests:

a. Both students and the teacher can search for other users (students) based on email, name, or hometown.

b. They can send connection requests to each other.

c. The recipient user (student or teacher) receives the request notification.

d. The recipient can choose to accept or decline the request.

e. If the recipient accepts the request, a connection is established between the users.

Existing System (WhatsApp Chat):

1.Registration and User Management:

- The existing system does not have a formal registration process or user management. It relies on the phone numbers of participants.
- User details and profiles are not available within the system.
- There is no distinction between different roles or access levels (e.g., students and college placement officers).

2.Posting and Updates:

- News and updates related to placements are posted in a chat format.
- Information may easily get lost or buried in the chat history.
- Users may miss important updates if they join the chat late or do not actively follow the conversations.
- There is no organized categorization or search functionality for posts.

3.Automation and Integration:

- The existing system lacks automation and integration with other college management systems.
- The college placement officers need to manually collect information from each student separately.

Proposed Placement Management System Website:

1.Registration and User Management:

- The proposed system has a formal registration process, allowing students to register online through their CMSys account.
- User profiles are created, providing a centralized and organized database of students.
- College placement officers can access student information without the need for separate data collection.

2.Posting and Updates:

- The website provides a dedicated platform for posting placement-related news and updates.
- Students can read and apply for companies of their choice, ensuring they do not miss any opportunities.
- The system can send frequent updates and notifications to students regarding placements.
- Information is organized, categorized, and easily accessible through the website.

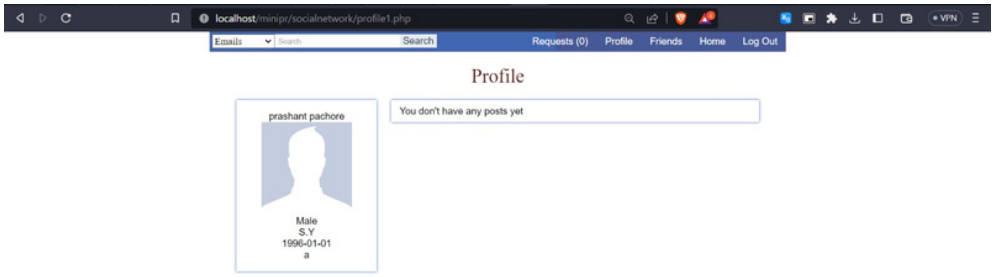
3.Automation and Integration:

The proposed system integrates with CMSys, the college management system, leveraging existing data and ensuring automatic updates when students register.

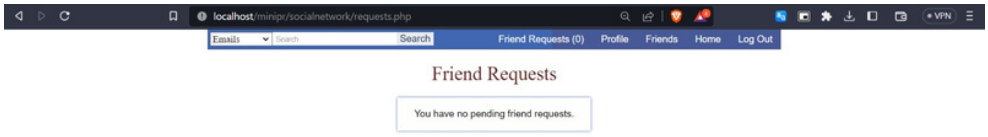
The system eliminates the need for manual data collection and reduces administrative workload.

3.Automation and Integration:

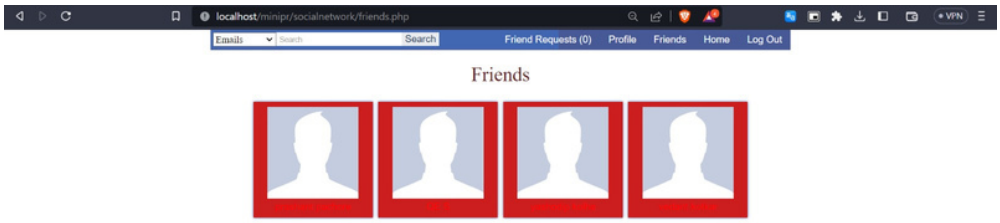
Profile



Frienf Requsts

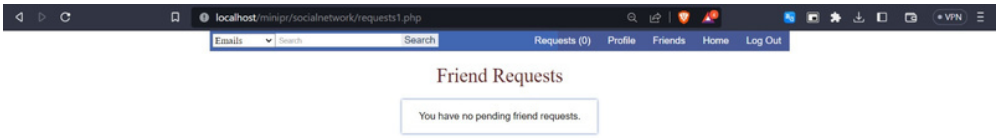


Friends



edit

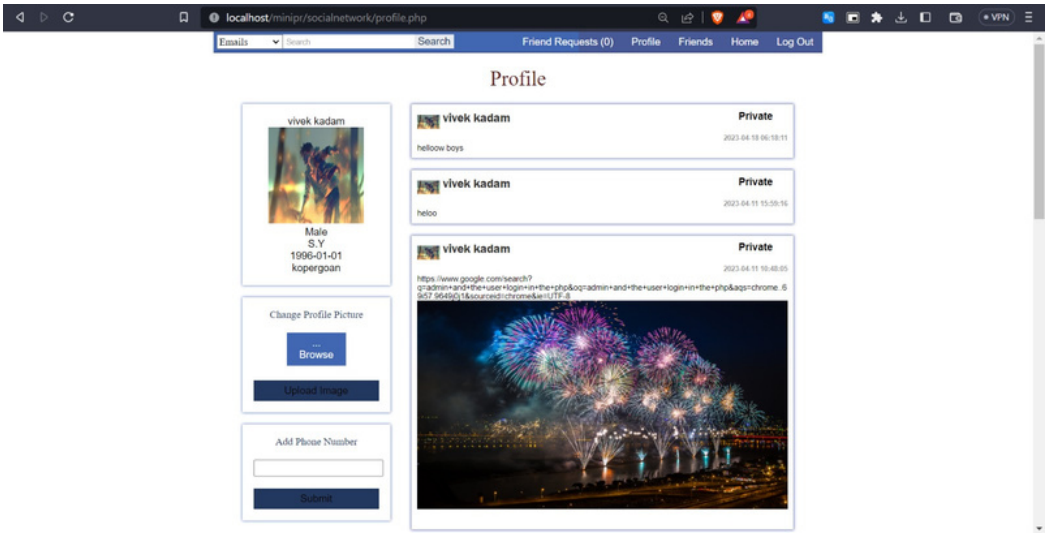
Friends requst admin



Login Pge

New Feeds

profile


localhost/minipr/socialnetwork/profile.php
Emails
Search
Search
Friend Requests (0)
Profile
Friends
Home
Log Out
Profile
vivek kadam
Male
S.Y
1996-01-01
kopergoan
Change Profile Picture
Browse
Upload Image
Add Phone Number
Submit
vivek kadam
Private
helloow boys
vivek kadam
Private
heloo
vivek kadam
Private

Program : index.php

```
<?php
require 'functions/functions.php';
//add admin.phpfile in this code
//require 'admin.php';
session_start();
if (isset($_SESSION['user_id'])) {
 header("location:home.php");
}
session_destroy();
session_start();
ob_start();
?>
<!DOCTYPE html>
<html>
<head>
 <title>Social Network</title>
 <link rel="stylesheet" type="text/css" href="resources/css/main.css">
 <style>
 .container{
 margin: 40px auto;
 width: 400px;
   }
 .content {
 padding: 30px;
 background-color: white;
 box-shadow: 0 0 5px #4267b2;
   }
 </style>
</head>
<body>
 <h1>Welcome to SPN</h1>
 <div class="container">
 <div class="tab">
 <button class="tablink active" onclick="openTab(event,'signin')"
id="link1">Login</button>
 <button class="tablink" onclick="openTab(event,'signup')" id="link2">Sign
Up</button>
 </div>

<div class="content">
 <div class="tabcontent" id="signin">
 <form method="post" onsubmit="return validateLogin()">
 <label>Email<span>*</span></label><br>
 <input type="text" name="useremail" id="loginuseremail">
 <div class="required"></div>
 <br>
 <label>Password<span>*</span></label><br>
    <input     type="password"     name="userpass"     id="loginuserpass">          <div
class="required"></div>
 <br><br>
```

```
<input type="submit" value="Login" name="login">
</form>
</div>
<div class="tabcontent" id="signup">
<form method="post" onsubmit="return validateRegister()">
<!--Package One-->
<h2>Highly Required Information</h2>
<hr>
<!--First Name-->
<label>First Name<span>*</span></label><br>
<input type="text" name="userfirstname" id="userfirstname">  <div class="required">
</div>
<br>
<!--Last Name-->
<label>Last Name<span>*</span></label><br>
<input type="text" name="userlastname" id="userlastname">
<div class="required"></div>
<br>
<!--Nickname-->
<label>Nickname</label><br>
<input type="text" name="usernickname" id="usernickname">
<div class="required"></div>
<br>
<!--Password-->
<label>Password<span>*</span></label><br>
<input type="password" name="userpass" id="userpass">
<div class="required"></div>
<br>
<!--Confirm Password-->
<label>Confirm Password<span>*</span></label><br>
<input type="password" name="userpassconfirm" id="userpassconfirm">
<div class="required"></div>
<br>
<!--Email-->
<label>Email<span>*</span></label><br>
<input type="text" name="useremail" id="useremail">
<div class="required"></div>
<br>
<!--Birth Date-->
Birth Date<span>*</span><br>
<select name="selectday">
<?php
for($i=1; $i<=31; $i++){
echo '<option value="'. $i .'">'. $i .'</option>';
  }
?>
</select>
<select name="selectmonth">
<?php
```

```
echo '<option value="1">January</option>';
echo '<option value="2">February</option>';
echo '<option value="3">March</option>';
echo '<option value="4">April</option>';
echo '<option value="5">May</option>';
echo '<option value="6">June</option>';
echo '<option value="7">July</option>';
echo '<option value="8">August</option>';
echo '<option value="9">September</option>';
echo '<option value="10">October</option>';
echo '<option value="11">Novemeber</option>';
echo '<option value="12">December</option>';
?>
</select>
<select name="selectyear">
<?php
for($i=2017; $i>=1900; $i--){
if($i == 1996){
echo '<option value="'. $i .'" selected>'. $i .'</option>';
  }
echo '<option value="'. $i .'">'. $i .'</option>';
  }
?>
</select>
<br><br>
<!--Gender-->                                    Home
      <input type="radio" name="usergender" value="M" id="malegender"
class="usergender">
<label>Male</label>
<input type="radio" name="usergender" value="F"
id="femalegender" class="usergender">
<label>Female</label>
<div class="required"></div>
<br>
<!--Hometown-->
<label>Hometown</label><br>
<input type="text" name="userhometown" id="userhometown">
<br>
<!--Package Two-->
<h2>Additional Information</h2>
<hr>
<!--Marital Status-->
<input type="radio" name="userstatus" value="S" id="singlestatus">
<label>Single</label>
<input type="radio" name="userstatus" value="E" id="engagedstatus">
<label>Engaged</label>
<input type="radio" name="userstatus" value="M"
```

```
id="marriedstatus">
 <label>Married</label>
 <br><br>
 <!--About Me-->
 <label>About Me</label><br>
 <textarea rows="12" name="userabout" id="userabout"></textarea>  <br><br>
 <input type="submit" value="Create Account" name="register">
 </form>
 </div>
 </div>
 </div>
 <script src="resources/js/main.js"></script>
</body>
</html>

<?php
$conn = connect();
if ($_SERVER['REQUEST_METHOD'] == 'POST') { // A form is posted
 if (isset($_POST['login'])) { // Login process
 $useremail = $_POST['useremail'];
 $userpass = md5($_POST['userpass']);
 $query = mysqli_query($conn, "SELECT * FROM users WHERE user_email = '$useremail'
AND user_password = '$userpass'");
 if($query){
 if(mysqli_num_rows($query) == 1) {
 $row = mysqli_fetch_assoc($query);
 $_SESSION['user_id'] = $row['user_id'];
 $_SESSION['user_name'] = $row['user_firstname'] . " " . $row['user_lastname'];
 if($row['user_type'] == 'admin') {
 header("location:home.php");
 } else {
 header("location:home1.php");
   }
   }
 else {
 ?> <script>
 document.getElementsByClassName("required")[0].innerHTML = "Invalid Login
Credentials.";
 document.getElementsByClassName("required")[1].innerHTML = "Invalid Login
Credentials.";
 </script> <?php
   }
 } else {
 echo mysqli_error($conn);
   }
   }

}
```

```
 if (isset($_POST['register'])) { // Register process
 // Retrieve Data
$userfirstname = $_POST['userfirstname'];
 $userlastname = $_POST['userlastname'];
 $usernickname = $_POST['usernickname'];
 $userpassword = md5($_POST['userpass']);
 $useremail = $_POST['useremail'];
    $userbirthdate = $_POST['selectyear'] . '-' . $_POST['selectmonth'] . '-' .
$_POST['selectday'];
 $usergender = $_POST['usergender'];
 $userhometown = $_POST['userhometown'];
 $userabout = $_POST['userabout'];
 if (isset($_POST['userstatus'])){
 $userstatus = $_POST['userstatus'];
   }
 else{
 $userstatus = NULL;
   }
 // Check for Some Unique Constraints
 $query = mysqli_query($conn, "SELECT user_nickname, user_email FROM users WHERE
user_nickname = '$usernickname' OR user_email = '$useremail'");
 if(mysqli_num_rows($query) > 0){
 $row = mysqli_fetch_assoc($query);
 if($usernickname == $row['user_nickname'] && !empty($usernickname)){
 ?> <script>
 document.getElementsByClassName("required")[4].innerHTML = "This Nickname
already exists.";
 </script> <?php
   }
 if($useremail == $row['user_email']){
 ?> <script>
 document.getElementsByClassName("required")[7].innerHTML = "This Email already
exists.";
 </script> <?php
   }
   }
 // Insert Data
 $sql = "INSERT INTO users(user_firstname, user_lastname, user_nickname,
user_password, user_email, user_gender, user_birthdate, user_status, user_about,
user_hometown)
 VALUES ('$userfirstname', '$userlastname', '$usernickname', '$userpassword',
'$useremail', '$usergender', '$userbirthdate', '$userstatus', '$userabout',
'$userhometown')";
 $query = mysqli_query($conn, $sql);
 if($query){
 $query = mysqli_query($conn, "SELECT user_id FROM users WHERE
user_email = '$useremail'");
 $row = mysqli_fetch_assoc($query);
 $_SESSION['user_id'] = $row['user_id'];
 header("location:home.php");
```

profile.php

Home.php

```
<?php
require 'functions/functions.php';
session_start();

// Check whether user is logged on or not
if (!isset($_SESSION['user_id'])) {
 header("location:index.php");
}
$temp = $_SESSION['user_id'];
session_destroy();
session_start();
$_SESSION['user_id'] = $temp;
ob_start();
// Establish Database Connection
$conn = connect();
?>

<!DOCTYPE html>
<html>
<head>
 <title>Social Network</title>
 <link rel="stylesheet" type="text/css" href="resources/css/main.css">
</head>
<body>
 <div class="container">
 <?php include 'includes/navbar.php'; ?>
 <br>
 <div class="createpost">
 <form method="post" action="" onsubmit="return validatePost()"
enctype="multipart/form-data">
 <h2>Make Post</h2>
 <hr>
 <span style="float:right; color:black">
 <input type="checkbox" id="public" name="public">
 <label for="public">Public</label>
 </span>
 Caption <span class="required" style="display:none;"> *You can't Leave the Caption
Empty.</span><br>
 <textarea rows="6" name="caption"></textarea>
 <center><img src="" id="preview" style="max-width:580px; display:none;"></center>
 <div class="createpostbuttons">
 <!--<form action="" method="post" enctype="multipart/form-data" id="imageform">-->
 <label>
 <img src="images/photo.png">
 <input type="file" name="fileUpload" id="imagefile">
 <!--<input type="submit" style="display:none;">-->
 </label>
 <input type="submit" value="Post" name="post">
```

```
<!--</form>-->
</div>
</form>
</div>
<h1>News Feed</h1>
<?php
// Public Posts Union Friends' Private Posts
    $sql = "SELECT posts.post_caption, posts.post_time, posts.post_public,
users.user_firstname,
users.user_lastname, users.user_id, users.user_gender, posts.post_id
FROM posts
JOIN users
ON posts.post_by = users.user_id
WHERE posts.post_public = 'Y' OR users.user_id = {$_SESSION['user_id']}
UNION
        SELECT posts.post_caption, posts.post_time, posts.post_public,
users.user_firstname,
users.user_lastname, users.user_id, users.user_gender, posts.post_id
FROM posts
JOIN users
ON posts.post_by = users.user_id
JOIN (
SELECT friendship.user1_id AS user_id
FROM friendship
WHERE friendship.user2_id = {$_SESSION['user_id']} AND
friendship.friendship_status = 1
UNION
SELECT friendship.user2_id AS user_id
FROM friendship
WHERE friendship.user1_id = {$_SESSION['user_id']} AND
friendship.friendship_status = 1
) userfriends
ON userfriends.user_id = posts.post_by
WHERE posts.post_public = 'N'
ORDER BY post_time DESC";
$query = mysqli_query($conn, $sql);
if(!$query){
echo mysqli_error($conn);
  }
if(mysqli_num_rows($query) == 0){
echo '<div class="post">';
echo 'There are no posts yet to show.';
echo '</div>';
  }
else{
$width = '40px'; // Profile Image Dimensions
$height = '40px';
```

```
while($row = mysqli_fetch_assoc($query)){
include 'includes/post.php';
echo '<br>';
  }
  }
?>
<br><br><br>
</div>
<script src="resources/js/jquery.js"></script>
<script>
// Invoke preview when an image file is choosen.
$(document).ready(function(){
$('#imagefile').change(function(){
preview(this);
});
});
// Preview function
function preview(input){
if (input.files && input.files[0]) {
var reader = new FileReader();
reader.onload = function (event){
$('#preview').attr('src', event.target.result);
$('#preview').css('display', 'initial');
  }
reader.readAsDataURL(input.files[0]);
  }
  }
// Form Validation
function validatePost(){
var required = document.getElementsByClassName("required");
   var caption = document.getElementsByTagName("textarea")[0].value;
required[0].style.display = "none";
if(caption == ""){
required[0].style.display = "initial";
return false;
  }
return true;
  }
</script>
</body>
</html>

<?php
if($_SERVER['REQUEST_METHOD'] == 'POST') { // Form is Posted
 // Assign Variables
 $caption = $_POST['caption'];
 if(isset($_POST['public'])) {
 $public = "Y";
 } else {
 $public = "N";
```

```
  }
 $poster = $_SESSION['user_id'];
 // Apply Insertion Query
  $sql = "INSERT INTO posts (post_caption, post_public, post_time, post_by)
 VALUES ('$caption', '$public', NOW(), $poster)";
 $query = mysqli_query($conn, $sql);
 // Action on Successful Query
 if($query){
 // Upload Post Image If a file was choosen
 if (!empty($_FILES['fileUpload']['name'])) {
 echo 'FUUUQ';
 // Retrieve Post ID
 $last_id = mysqli_insert_id($conn);
 include 'functions/upload.php';
  }
 header("location: home.php");
  }
}
?>
```

profile.php

```
<?php
echo '<div class="profile">';
echo '<center>';
$row = mysqli_fetch_assoc($profilequery);
// Name and Nickname
if(!empty($row['user_nickname']))
 echo $row['user_firstname'] . ' ' . $row['user_lastname'] . ' (' .
$row['user_nickname'] . ')';
else
 echo $row['user_firstname'] . ' ' . $row['user_lastname'];
echo '<br>';
// Profile Info & View
$width = '168px';
$height = '168px';
include 'includes/profile_picture.php';
echo '<br>';
// Gender
if($row['user_gender'] == "M")
 echo 'Male';
else if($row['user_gender'] == "F")
 echo 'Female';
echo '<br>';
// Status
if(!empty($row['user_status'])){
 if($row['user_status'] == "S")
 echo 'Single';
 else if($row['user_status'] == "E")
 echo 'Engaged';
 else if($row['user_status'] == "M")
 echo 'Married';
 echo '<br>';
}
// Birthdate
echo $row['user_birthdate'];
// Additional Information
if(!empty($row['user_hometown'])){
 echo '<br>';
 echo $row['user_hometown'];
}
if(!empty($row['user_about'])){
 echo '<br>';
 echo $row['user_about'];
}
// Friendship Status
if($flag == 1){
 echo '<br>';
 if(isset($row['friendship_status'])) {
 if($row['friendship_status'] == 1){
 echo '<form method="post">';
 echo '<input type="submit" value="Friends" disabled="disabled"
```

```
id="special">';
 echo '</form>';
 } else if ($row['friendship_status'] == 0){
 echo '<form method="post">';
    echo '<input type="submit" value="Request Pending" disabled="disabled"
id="special">';
 echo '</form>';
   }
 } else {
 echo '<form method="post">';
 echo '<input type="submit" value="Send Friend Request" name="request">';
 echo'</form>';
   }
}

echo '<center>';
echo'</div>';

$query4 = mysqli_query($conn, "SELECT * FROM user_phone WHERE user_id =
{$row['user_id']}");
if(!$query4){
 echo mysqli_error($conn);
}
if(mysqli_num_rows($query4) > 0){
 echo '<br>';
 echo '<div class="profile">';
 echo '<center class="changeprofile">';
echo 'Phones:';
 echo '<br>';
 while($row4 = mysqli_fetch_assoc($query4)){
 echo $row4['user_phone'];
 echo '<br>';
   }
 echo '</center>';
 echo '</div>';
}

?>
```

Results

```
 WHERE (posts.post_public = 'Y' OR users.user_id = {$_SESSION['user_id']}) AND
posts.post_caption LIKE '%$key%'
 UNION
 SELECT posts.post_caption, posts.post_time, posts.post_public, users.user_firstname,
 users.user_lastname, users.user_id, users.user_gender, posts.post_id
 FROM posts
 JOIN users
 ON posts.post_by = users.user_id
 JOIN (
 SELECT friendship.user1_id AS user_id
 FROM friendship
 WHERE friendship.user2_id = {$_SESSION['user_id']} AND friendship.friendship_status =
1
 UNION
 SELECT friendship.user2_id AS user_id
 FROM friendship
 WHERE friendship.user1_id = {$_SESSION['user_id']} AND friendship.friendship_status =
1
 ) userfriends
 ON userfriends.user_id = posts.post_by
 WHERE posts.post_public = 'N' AND posts.post_caption LIKE '%$key%'
 ORDER BY post_time DESC";
 $query = mysqli_query($conn, $sql);
 $width = '40px'; // Profile Image Dimensions
 $height = '40px';
 if(!$query){
 echo mysqli_error($conn);
   }
 if(mysqli_num_rows($query) == 0){
 echo '<div class="post">';
 echo 'There is no results given the keyword, try to widen your search query.';
 echo '</div>';
 echo '<br>';
   }
 while($row = mysqli_fetch_assoc($query)){
 include 'includes/post.php';
 echo '<br>';
   }
 }
?>
 </div>
</body>
</html>
```

Home

search.php

```
<?php
//
require 'functions/functions.php';
session_start();
// Check whether user is logged on or not
if (!isset($_SESSION['user_id'])) {
 header("location:index.php");
}
// Establish Database Connection
$conn = connect();

?>

<!DOCTYPE html>
<html>
<head>
 <title>Social Network</title>
 <link rel="stylesheet" type="text/css" href="resources/css/main.css">
</head>
<body>
 <div class="container">
 <?php include 'includes/navbar.php'; ?>
 <h1>Search Results</h1>
 <?php
 $location = $_GET['location'];
 $key = $_GET['query'];
 if($location == 'emails') {
 $sql = "SELECT * FROM users WHERE users.user_email = '$key'";
 include 'includes/userquery.php';
 } else if($location == 'names') {
 $name = explode(' ', $key, 2); // Break String into Array.
 if(empty($name[1])) {
 $sql = "SELECT * FROM users WHERE users.user_firstname = '$name[0]' OR
users.user_lastname= '$name[0]'";
 } else {
 $sql = "SELECT * FROM users WHERE users.user_firstname = '$name[0]' AND
users.user_lastname= '$name[1]'";
    }
 include 'includes/userquery.php';
 } else if($location == 'hometowns') {
  $sql = "SELECT * FROM users WHERE users.user_hometown = '$key'";   include
'includes/userquery.php';
 } else if($location == 'posts') {
 $sql = "SELECT posts.post_caption, posts.post_time, posts.post_public,
users.user_firstname,
 users.user_lastname, users.user_id, users.user_gender, posts.post_id
 FROM posts
 JOIN users
 ON posts.post_by = users.user_id
```

profile.php

friends.php

```
<?php
require 'functions/functions.php';
session_start();
// Check whether user is logged on or not
if (!isset($_SESSION['user_id'])) {
 header("location:index.php");
}
// Establish Database Connection
$conn = connect();
?>

<!DOCTYPE html>
<html>
<head>
 <title>Social Network</title>
 <link rel="stylesheet" type="text/css" href="resources/css/main.css">
 <style>
 .frame a{
 text-decoration: none;
 color: #4267b2;
   }
 .frame a:hover{
 text-decoration: underline;
   }
 </style>
</head>
<body>
 <div class="container">
 <?php include 'includes/navbar.php'; ?>
 <h1>Friends</h1>
 <?php
 echo '<center>';
$sql = "SELECT users.user_id, users.user_firstname,
users.user_lastname, users.user_gender
 FROM users
 JOIN (
 SELECT friendship.user1_id AS user_id
 FROM friendship
        WHERE        friendship.user2_id        =        {$_SESSION['user_id']}        AND
friendship.friendship_status = 1
 UNION
 SELECT friendship.user2_id AS user_id
 FROM friendship
        WHERE        friendship.user1_id        =        {$_SESSION['user_id']}        AND
friendship.friendship_status = 1
 ) userfriends
 ON userfriends.user_id = users.user_id";
 $query = mysqli_query($conn, $sql);
 $width = '168px';
 $height = '168px';
```

Results

```
if($query){
if(mysqli_num_rows($query) == 0){
echo '<div class="post">';
echo 'You don\'t yet have any friends.';
echo '</div>';
} else {
while($row = mysqli_fetch_assoc($query)){
echo '<div class="frame">';
echo '<center>';
include 'includes/profile_picture.php';
echo '<br>';
 echo  '<a  href="profile.php?id='  .  $row['user_id']  .  '">'  .  $row['user_firstname']  .  '  '  .
$row['user_lastname'] . '</a>';
echo '</center>';
echo '</div>';
   }
   }
   }
echo '</center>';
?>
</div>
</body>
</html>
```


profile.php

Logout.php

```
<?php
session_start();
session_destroy();
header("location:index.php");
?>
```

rEFRENCES

Results

The implementation of a Placement Management System for colleges can have several profound and far-reaching positive results. One of the primary benefits is increased efficiency in the placement process. By automating and streamlining various tasks, the system reduces the burden of manual work on administrators, allowing them to allocate their time and resources more effectively. This, in turn, saves valuable time and ensures a smoother and more streamlined placement process.

Another significant advantage is the enhanced transparency offered by the system. With a centralized platform, all relevant information regarding job postings, requirements, and eligibility criteria can be easily accessed by students. This transparency eliminates any ambiguity or confusion, ensuring that students have a clear understanding of the available opportunities. Moreover, the system can provide real-time updates and notifications about new job postings, keeping students informed and engaged throughout the placement process.

Furthermore, the accuracy of the placement process is greatly improved with the implementation of a Placement Management System. By digitizing data and employing intelligent algorithms, the system can match students with relevant job opportunities based on their profiles, skills, and preferences. This personalized approach increases the likelihood of successful matches and improves the overall quality of placements.

The user-friendly and intuitive design of the Placement Management System significantly enhances the job search experience for college students. With a user-friendly interface, students can easily create profiles, search for suitable job opportunities, and submit applications. This simplicity and ease of use contribute to a positive user experience, ultimately benefiting both students and administrators.

Looking to the future, the potential scope for incorporating a feature that allows students to directly apply to companies and submit their resumes through the app holds immense promise. This added functionality would further streamline the application process, reducing manual intervention and saving time for both students and employers. It would provide a seamless and efficient experience, aligning with modern hiring practices and enhancing the system's usefulness.

In conclusion, the implementation of a Placement Management System using modern technology can bring about significant positive results. It increases efficiency, transparency, and accuracy in the placement process, ultimately providing better placement opportunities for students. With its user-friendly design and potential for future expansion, this system has the potential to modernize the education system and greatly improve the prospects of students in their career journeys.

Conclusion:

Placement Management System is an exceptionally user-friendly and efficient platform that revolutionizes the process of connecting college students with potential employers for internships and job placements. With its intuitive interface and robust functionality, it simplifies the job search process, empowering students to take control of their career prospects.

The system offers students a seamless experience by allowing them to easily create a comprehensive profile, highlighting their skills, qualifications, and experiences. Through a sophisticated search feature, students can effortlessly browse through a vast database of job opportunities, filtering them based on their preferences, such as location, industry, or job type. Additionally, the system provides timely notifications to students about newly posted job openings, ensuring they stay updated with the latest opportunities.

Not only does the Placement Management System benefit students, but it also provides invaluable advantages for administrators. The inclusion of a secure login system enables administrators to efficiently manage user accounts and job postings, maintaining a well-organized and updated platform.

In summary, the Placement Management System serves as a powerful tool for colleges and students, revolutionizing the job search process and enhancing placement opportunities. Its user-friendly interface, comprehensive features, and potential for future expansion make it an invaluable asset in facilitating successful internships and job placements. By bridging the gap between students and employers, this system contributes to the overall growth and development of aspiring professionals, ultimately leading to successful and fulfilling careers.

REFRENCES

Research Papers :-

1. COLLEGE PLACEMENT MANAGEMENT SYSTEM

Farheen Taqi Rizvi1 , Naushin Arif Khan2 , Saurabh Sanjay Upadhyay3 , Prof. Sonali Suryawanshi4 1, 2, 3, 4Dept. Of Computer Engineering, Rizvi College of Engineering, Mumbai, India

2.PLACEMENT MANAGEMENT SYSTEM

Maryam Sayyed, Faiza Umatiya,Seemab Zehera, Prof. Shiburaj Pappu 1Student, 2 Student, 3 Student, 4 Assistant professor 1Computer engineering dept. RCOE, 1Rizvi College of Engineering, Mumbai, India

3.PLACEMENT MANAGEMENT SYSTEM FOR CAMPUS RECRUITMENT

Ajeena Sunny1 Aneena Felix1 Angelin Saji1 Christina Sebastian1 Praseetha V.M2 1UG Student, Dept. of CSE, SJCET, Palai, Kerala, India 2Associate Professor, Dept. of CSE, SJCET, Palai, Kerala, India

WEBSITES :

1) GEEKFORGEEKS

HTTPS://WWW.GEEKSFORGEEKS.ORG/

2)PROGRAMIZ

HTTPS://WWW.PROGRAMIZ.COM/