

A
Project Report
On
PROJECT: PLACEMENT RECORD
MANAGEMENT SYSTEM

Bachelor of Computer Application
(2023 2026)



Submitted by :

Harsheekesh Tiwari (1323466)

Utosh Narayan (1323461)

Ankit Raj (1323467)

Submitted to :

Mrs. Preeti Sharma

Asst. Prof.

M.M. Institute of Computer Technology & Business Management Maharishi Markandeshwar
(Deemed to be University), Mullana (Ambala) (NAAC accredited Grade 'A++' University)

INDEX

1. Introduction	03
a. Overview.....	04
b. Rationale: Justification, why needed	04-05
c. Objective	05
d. Literature Review	05
e. Feasibility Study	06
I. Technical Feasibility	
II. Economic Feasibility	
2. Software and Hardware requirements	07-09
3. Software Requirement Analysis	10-13
4. Software Design	14-18
5. Coding/Code Templates	19-48
6. Testing.....	49-50
7. Simulation/Output Screens.....	51-52
8. Conclusions.....	53
9. References.....	54

Introduction:

Welcome to the **MMDU Placement Records portal**, where you can discover the remarkable placement successes of our students at **MMDU Mullana**. With great pride, our department has long been known for giving students outstanding chances to launch their careers in top companies worldwide. This platform showcases detailed information about our past placement records, giving insights into the companies that hire from us, the job roles offered, and the success stories of our graduates.

This site offers a thorough overview of the professional outcomes and opportunities that the MMICT & BM department offers, whether you are a recruiter, a potential student, or an existing member of the MMU community. The data spans from **2018 to 2024**, covering:

- The **top companies** that hire from our department
- The diverse **job roles** offered to our graduates
- **Salary packages** and trends over the years
- A breakdown of placement percentages and student achievements

To provide you with a better visual representation of the data and to make it easier for you to analyze trends and patterns, we have also included interactive graphs and charts. See how our department prepares students for a bright future in the working world and learn about the amazing journeys that our graduates pursue.

Overview:

The Overview provides a general description of your project, highlighting its purpose and scope. In this case, the "Placement Record Management System" is an automated solution designed to manage and streamline the process of student placements. This section should provide readers with a high-level understanding of what the system does, its key features, and how it benefits both students and educational institutions.

For example:

- A brief introduction to the problem the system addresses (inefficient placement record management).
- The purpose of the system (to automate and simplify the placement process).
- An overview of its key components (student profiles, company details, placement tracking, etc.).

Rationale: Justification, Why Needed:

The Placement Record Management System is a crucial solution to address several challenges faced by educational institutions in managing placement records effectively. In today's fast-paced academic environment, the placement process involves a large volume of data that must be handled with precision, timeliness, and efficiency. Traditional manual or semi-automated systems often result in inefficiencies, inaccuracies, and delays that negatively impact the placement experience for both students and organizations.

Justification:

- Automates the placement process, reducing manual errors.
- Improves efficiency in managing large volumes of placement data.
- Streamlines record tracking and reporting for better decision-making

Why Needed:

- Showcase Placement Achievements
- Enhance Transparency for Upcoming Enrollment of Students
- Facilitate Recruiters with Comprehensive Data
- Data Analytics for Continuous Improvement

Objective:

The Objective section outlines the primary goals and outcomes your project intends to achieve. This section should be concise and focused on the measurable results you hope to accomplish with your system.

- To automate the placement record management process: Minimize human intervention and reduce errors.
- To streamline communication between students, faculty, and companies: Facilitate real-time updates and feedback.
- To improve data accessibility and reporting: Provide instant access to placement data for all stakeholders.
- To ensure accurate tracking and management of placement status: Maintain a real-time, updated record of student placements.

Literature Review:

- **Existing systems:** Discuss how other systems handle placement management, whether manual or automated.
- **Technological advances:** Explain the technologies that have improved placement management, such as databases, cloud computing, or mobile apps.
- **Challenges in the field:** Mention common issues such as data redundancy, lack of real-time tracking, and poor user experience that your project addresses.

Feasibility Study:

I. Technical Feasibility

- **Technology Stack:** Identify the programming languages, databases, and frameworks you'll use (e.g., Java, Python, MySQL, HTML, CSS, etc.).
- **Development Tools:** Outline the tools for development and testing, such as IDEs (Integrated Development Environments), version control systems (e.g., Git), and testing frameworks.
- **Infrastructure Requirements:** Discuss the hardware or cloud services required to deploy and host the system.
- **Technical Skills:** Ensure your team (or yourself) has the required technical expertise to develop the system.

II. Economic Feasibility

- **Cost Estimates:** Provide a breakdown of the expected costs for software development, hardware (if any), and ongoing maintenance.
- **Potential Savings:** Compare the costs with the expected savings (e.g., time saved, improved efficiency, reduced errors, and reduced administrative overhead).
- **Return on Investment (ROI):** Estimate the financial benefit your system will bring to the institution. This might include reduced time spent on administrative tasks, better placement success rates, and improved student satisfaction.

Software and Hardware Requirements:

I. Software Requirements:

For the successful implementation of the Placement Record Management System, various software tools and technologies are needed. These tools span across frontend development, backend processing, database management, and overall system deployment. Each plays a pivotal role in ensuring smooth functioning and user interaction.

Frontend Development Tools:

➤ **HTML5 (Hyper Text Markup Language):**

HTML forms the structural foundation of the web pages. It defines the layout and structure of the website, including the placement of elements like text, buttons, input fields, etc.

➤ **CSS3 (Cascading Style Sheets):**

CSS is responsible for styling the website, ensuring that the pages are visually appealing. It controls elements like colors, fonts, layouts, and responsive design for different screen sizes.

➤ **JavaScript (JS):**

JavaScript adds interactivity to the website. It handles client-side functionalities, such as form validation, dynamic content loading, and creating interactive charts.

Backend Development Tools:

➤ **PHP:**

PHP can be used as an alternative or in combination with Node.js to handle server-side logic. It is widely used for form submissions, database interaction, and generating dynamic page content.

Database Management Systems:

➤ **MySQL:**

MySQL is the relational database management system (RDBMS) chosen for this project to store and manage student placement records. It offers fast query processing, data integrity, and ease of use.

Other Tools:

➤ **Version Control System - Git:**

Git will be used for source code version control, enabling team collaboration and tracking of changes made to the codebase. GitHub or GitLab can be used to host the repositories.

➤ **IDE (Integrated Development Environment):**

For coding, IDEs like Visual Studio Code (VS Code), JetBrains WebStorm, or Sublime Text can be used to write, debug, and manage the project code efficiently.

II. **Hardware Requirements:**

The hardware requirements for the development and deployment of the Placement Record Management System are outlined below:

Development Machine:

- **Processor:** A minimum of Intel Core i5 or AMD Ryzen 5 processor is recommended for smooth development.
- **RAM:** At least 8 GB of RAM is required to handle multiple development tools simultaneously (IDE, browser, local server, etc.).
- **Storage:** A Solid-State Drive (SSD) with at least 256 GB capacity is recommended to ensure fast read/write speeds during development.
- **Graphics:** While integrated graphics are sufficient, a dedicated graphics card (e.g., NVIDIA GTX 1050 or higher) will improve performance if there's a need for graphical processing.
- **Operating System:** Windows 10/11, macOS, or Linux—the choice of OS depends on the developer's preference.

Backup and Security:

- A cloud backup solution (such as AWS S3 or Google Cloud Storage) should be in place to ensure regular backups of critical data.
- Firewall and SSL Certificates will be installed to ensure secure communication between users and the web server.

Software Requirement Analysis:

Software Requirement Analysis (SRA) is a critical phase in the development life cycle of the Placement Record Management System (PRMS). It involves gathering and specifying the functional and non-functional requirements necessary for the design and implementation of the system. This phase ensures that the software system meets the needs of its users, including administrators, students, and potential employers, while adhering to organizational standards.

I. Functional Requirements:

Functional requirements define the specific behavior or functions of the system. In the context of the Placement Record Management System, these are the capabilities that the system must provide to fulfill user needs and achieve its objectives.

User Roles and Responsibilities:

➤ Admin:

- **Manage Placement Records:** Admins can create, update, or delete placement records, upload student details, assign job roles, and manage the overall database.
- **Job Postings:** Admins can post new job opportunities, edit existing job listings, and assign them to relevant students or departments.
- **User Management:** Admins control access to the system, create new user accounts for faculty and employers, and assign or revoke roles.
- **Data Monitoring:** Admins can view reports, access system logs, and ensure the integrity of the data stored in the system.

➤ **Faculty:**

- **Read-only Access:** Faculty members can view placement records, monitor students' progress, and provide feedback without having the authority to modify data.
- **Departmental Reports:** Faculty can generate department-wise reports to analyze trends in placements and job opportunities.

➤ **Students:**

- **View Job Listings:** Students can browse job listings, view company profiles, and apply for relevant job roles based on their qualifications.
- **Placement Records:** Students can view their own placement history and track job applications through the system.

➤ **Companies:**

- **Read-only Access:** Companies can view the profiles of students who have applied for jobs and verify qualifications before conducting interviews.

System Feature:

The core features of the Placement Record Management System are designed to simplify the placement process for administrators and make it easier for students and companies to access and manage relevant data.

➤ **Placement Record Management:**

- The system will allow the admin to add, update, or delete placement records, such as student details, job roles, and companies.
- It will provide options to import data from Excel files or manually input the data into a web-based form.

➤ **Job Posting Management:**

- Admins can create new job postings by entering details such as job role, company name, salary packages, and eligibility criteria.
- Job postings can be filtered by department, qualification, or skill set to ensure that only relevant students are notified.

➤ **Report Generation:**

- The system will generate reports that summarize placement statistics, such as the number of students placed per department, highest packages, and overall placement percentages.
- These reports can be exported in various formats (e.g., PDF, Excel) for internal use and sharing with stakeholders.

II. Non-Functional Requirements:

Non-functional requirements refer to the performance, reliability, usability, and other constraints that the system must meet to ensure smooth operation. These are critical for enhancing the user experience and overall efficiency of the system.

Usability:

- **Intuitive Interface:** The system should have an easy-to-use and responsive interface for all user roles, including admins, faculty, students, and companies. Minimal training should be required to navigate the platform.
- **Mobile Responsiveness:** The system must be mobile-friendly, allowing students and faculty to access the portal on smartphones, tablets, and desktop computers. The use of CSS frameworks like Tailwind or Bootstrap ensures seamless scaling across devices.

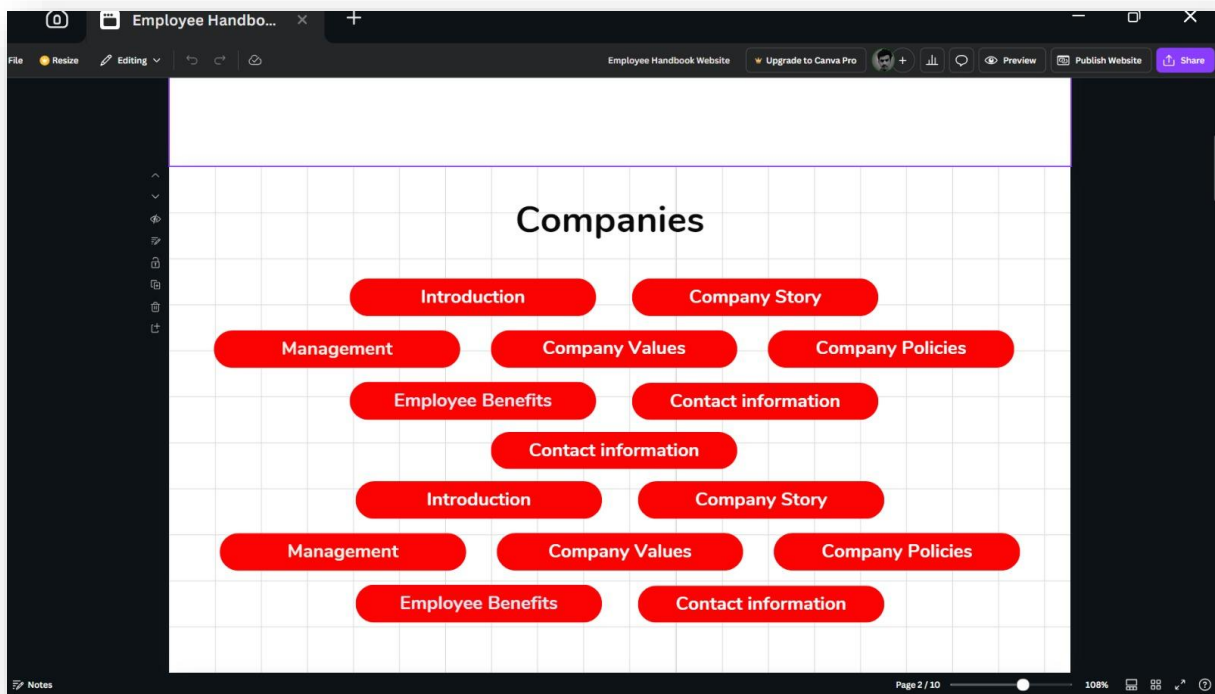
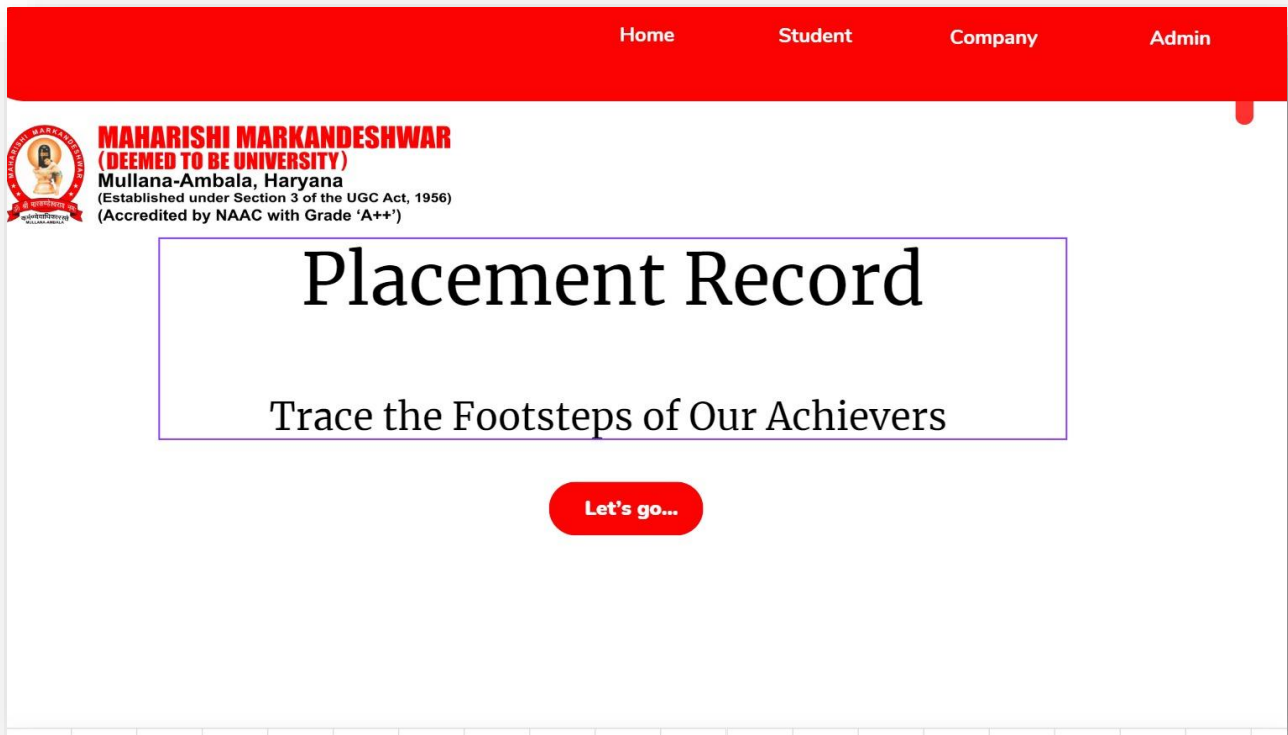
Security:

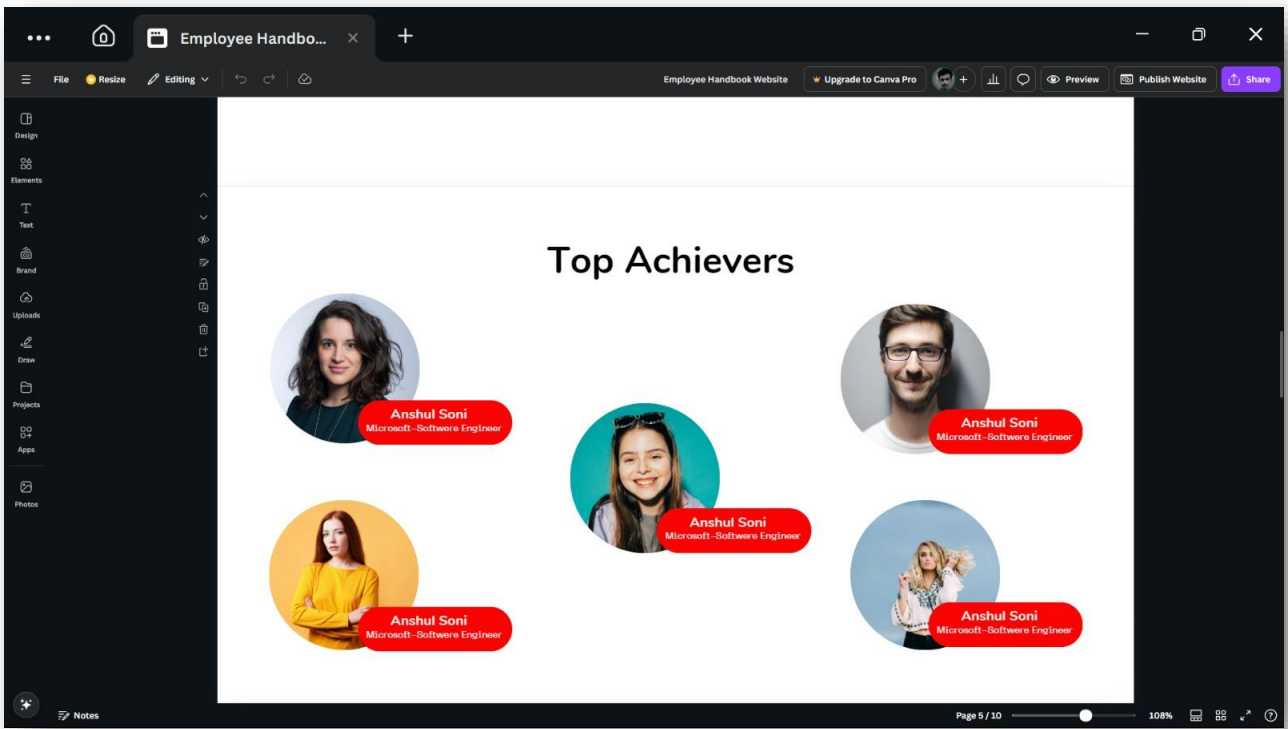
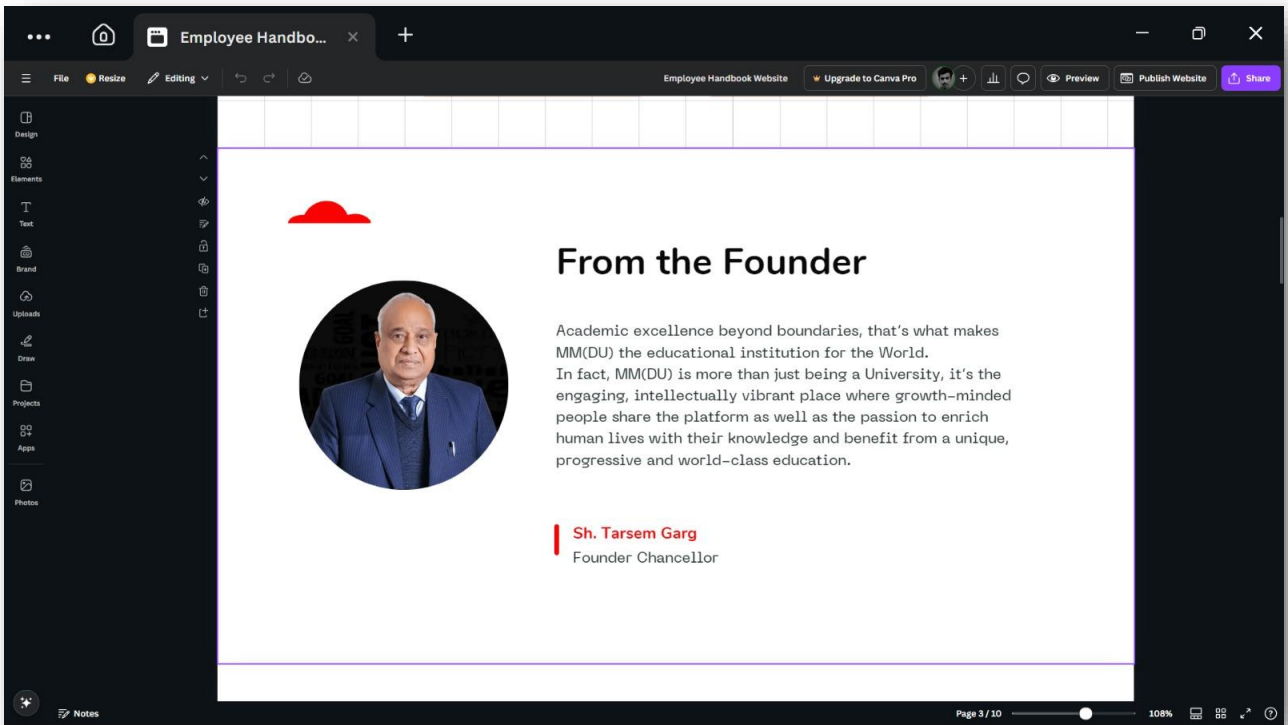
- **User Authentication:** The system must provide a secure authentication mechanism for all users, including multi-factor authentication (MFA) for admins. This prevents unauthorized access to sensitive student placement records and job postings.
- **Role-Based Access Control (RBAC):** Different user roles (Admin, Faculty, Student, Company) should have access to different parts of the system based on their permissions. Admins will have complete control over the system, while students and companies will have limited, read-only access.
- **Data Encryption:** Sensitive data such as student details, placement records, and job offers must be encrypted in transit (using HTTPS) and at rest (using encryption algorithms like AES-256).

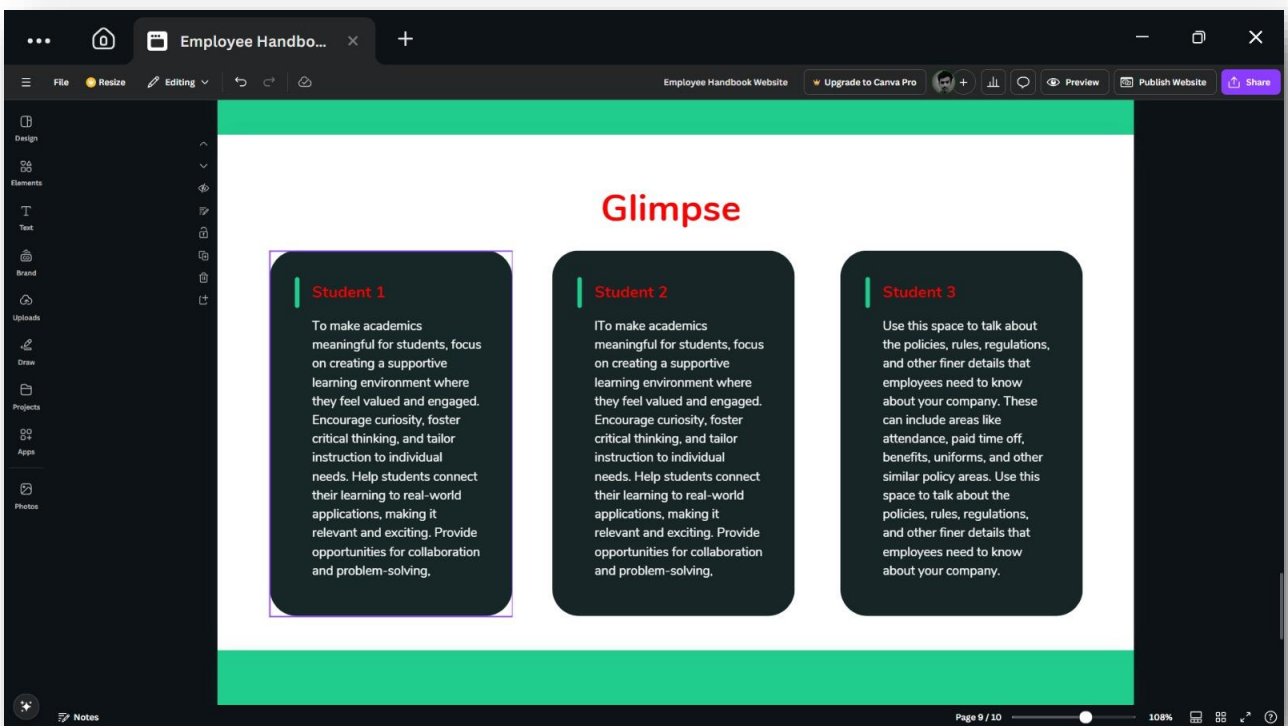
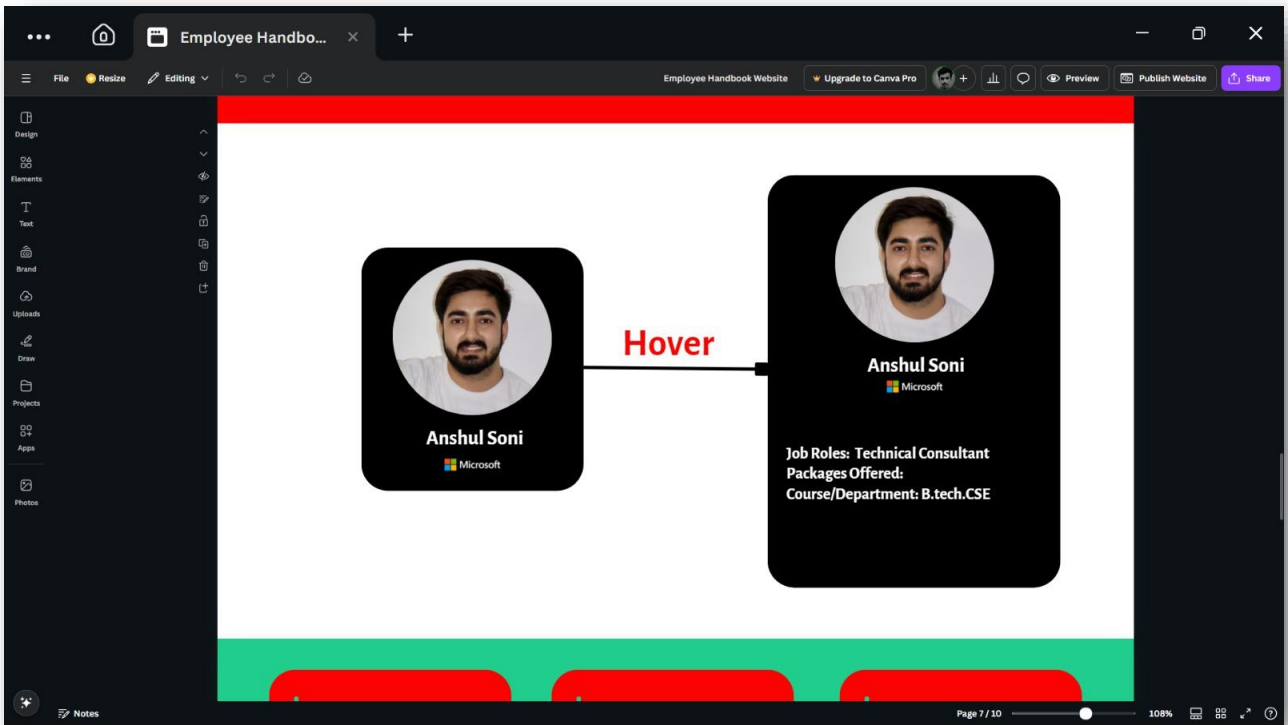
Performance:

- **Scalability:** The system should be scalable to accommodate an increasing number of users and data entries. As the number of students and placement records grows, the system should continue to perform efficiently.
- **Response Time:** The system should provide near real-time responses when accessing, filtering, or searching through placement records and job listings. Ideally, response times should not exceed 2-3 seconds for common operations.
- **Data Storage and Backup:** The database (MySQL or PostgreSQL) should be optimized for handling large datasets with quick query execution. Regular backups should be automated to ensure data recovery in case of failure.

Software Design:







Used Images:



**Training
& Placements**



**Industry Academia
Partnerships**



**Career
Development**



Coding/ Code Template:

Main Page: index.php

```
<?php
// Database connection details
$host = "localhost";
$user = "root";
$password = "basu2004";
$db = "admin";

$conn = new mysqli($host, $user, $password, $db);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT student_name, course, student_email, student_image, company_name, job_role,
salary_package, placement_year FROM student_placement_record LIMIT 5"; // Adjust the query
based on your table structure
$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Placement Record</title>
    <link rel="stylesheet" href="style1.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Merriweather:ital,wght@0,300;0,400;0,700;0,900;
1,300;1,400;1,700;1,900&display=swap" rel="stylesheet">
</head>
<body>
    <header>
        <div class="logo">
            
```

```

<div class="university-info">
  <h1>MAHARISHI MARKANDESHWAR</h1>
  <p>(DEEMED TO BE UNIVERSITY)</p>
  <p>Mullana-Ambala, Haryana</p>
  <p>(Accredited by NAAC with Grade 'A++')</p>
</div>
</div>
<nav>
  <ul>
    <li><a href="#hero">Home</a></li>
    <li><a href="#Student">Student</a></li>
    <li><a href="#companies">Company</a></li>
    <li><a href="http://localhost/Placement%20Record%20Project/Placement%20Record%20Project/Placement%20Record%20Project/Admin-Pannel/login.php">Admin</a></li>
  </ul>
</nav>
</header>

```

```

<section class="hero">

```

```

  <h2>Placement Record</h2>
  <p>Trace the Footsteps of Our Achievers</p>
  <!-- <p>We at MM(DU) strongly believe that each and every MMDU student would spread
his/her golden dust with the progressive learning attitude that they have been sculptured with and
would come out victoriously with flying colors in each field of their life.</p> -->
  <a href="#companies" class="cta-button">Let's Go</a>
</section>

```

```

<div class="red-data"></div>
<div class="quote-container">
  <div class="quote">
    <p>Creating Impeccable Placement Opportunities</p>
  </div>
</div>
</main>
<section id="companies">
  <div class="container">
    <h1 id="comp">Companies</h1>
    <div class="box-container">
      <div class="box">Infosys Ltd</div>
      <div class="box">TCS</div>
      <div class="box">IBM</div>
    </div>
  </div>
</section>

```

```

    <div class="box">HCL</div>
    <div class="box">Wipro Technologies Ltd.</div>
    <div class="box">LAVA International</div>
    <div class="box">Cognizant</div>
    <div class="box">Accenture</div>
    <div class="box">Tech Mahindra</div>
    <div class="box">Capagimini</div>
    <div class="box">JP Morgan</div>
    <div class="box">Vivo</div>
    <div class="box">Bosch</div>
    <div class="box">Infogain</div>
    <div class="box">eClerx</div>
    <div class="box">Airtel</div>
    <div class="box">VI</div>
    <div class="box">Samsung</div>
    <div class="box">Byju's</div>
    <div class="box">Informatica</div>
    <div class="box">Pepsico</div>
  </div>
</div>
</section>

<!-- Graph -->
<div class="graph-container">
  <h1>Placement Statics</h1>
</div>
  <div class="graph-img1">
    <div class="img-1"></div>
    <div class="img-2"></div>
  </div>
  <br>
  <div class="graph-img2">
    <div class="img-1"></div>
    <div class="img-2"></div>
  </div>

</div>
<section id="Student">
  <div class="top-achievers">
    <h2>Top Achievers</h2>
    <div class="achievers-container">
      <div class="achiever">
        
        <div class="name-tag">
          <p>Anshul Soni</p>

```

```

        <p>Microsoft - Software Engineer</p>
    </div>
</div>
<div class="achiever">
    
    <div class="name-tag">
        <p>Anshul Soni</p>
        <p>Microsoft - Software Engineer</p>
    </div>
</div>
<div class="achiever">
    
    <div class="name-tag">
        <p>Anshul Soni</p>
        <p>Microsoft - Software Engineer</p>
    </div>
</div>
<div class="achiever">
    
    <div class="name-tag">
        <p>Anshul Soni</p>
        <p>Microsoft - Software Engineer</p>
    </div>
</div>
<div class="achiever">
    
    <div class="name-tag">
        <p>Anshul Soni</p>
        <p>Microsoft - Software Engineer</p>
    </div>
</div>
</div>
</div>

```

```

<!--<section id="Student">
<div class="top-achievers">
<h2>Top Achievers</h2>
<div class="achievers-container">
    <?php if ($result->num_rows > 0): ?>
        <?php while ($row = $result->fetch_assoc()): ?>
            <div class="achiever">

                <?php if (!empty($row['student_image'])): ?>

```



```

        " />
        <?php else: ?>
        
        <?php endif; ?>
        <div class="name-tag">
        <p><?php echo htmlspecialchars($row['student_name']); ?></p>
        <p><?php echo htmlspecialchars($row['job_role']); ?> at <?php echo
htmlspecialchars($row['company_name']); ?></p>
        <p>Course: <?php echo htmlspecialchars($row['course']); ?></p>
        <p>Email: <?php echo htmlspecialchars($row['student_email']); ?></p>
        <p>Salary Package: <?php echo number_format($row['salary_package'], 2); ?>
LAKHS</p>
        <p>Placement Year: <?php echo htmlspecialchars($row['placement_year']);
?></p>
        </div>
    </div>
    <?php endwhile; ?>
    <?php else: ?>
    <p>No achievers found.</p>
    <?php endif; ?>
</div>
</div>
</section>-->

</section>
</main>
<div class="founder-section">
    <div class="founder-image">
        
    </div>
    <div class="founder-message">
        <h2 id="from-founder">From the Founder</h2>
        <p>
            Academic excellence beyond boundaries, that's what makes MM(DU) the educational
            institution for the World.
            In fact, MM(DU) is more than just being a University, it's the engaging, intellectually
            vibrant place where growth-minded people share the platform as well as the passion to enrich
            human lives with their knowledge and benefit from a unique, progressive and world-class
            education.
        </p>
        <p class="founder-signature">
            <span class="founder-name">Sh. Tarsem Garg</span><br>

```

```

        Founder Chancellor
    </p>
</div>
</div>

<footer>

    <div class="footer-container2">
        <div class="contact-info">
            <h2>Connect</h2>
            <p>Maharishi Markandeshwar (DEEMED TO BE UNIVERSITY)<br>
            Mullana, Ambala (Haryana) +91-1731-274475, 76, 77, 78 | Toll Free: 1800 2740 240<br>
            <a href="mailto:info@mmumullana.org">info@mmumullana.org</a></p>
            <div class="social-icons">
                <a href="https://www.facebook.com/MMDUMullana/"></a>
                <a href="https://www.instagram.com/mmdumullana/"></a>
                <a href="https://www.linkedin.com/school/maharishi-markandeshwar-university-
mullana"></a>
                <a
href="https://www.youtube.com/MaharishiMarkandeshwarUniversityMullana"></a>
                <a href="https://api.whatsapp.com/send?phone=8059930001"></a>
            </div>
        </div>

    </div>
    <div class="footer-bottom">
        <p>© Copyright <strong>MMDU</strong>. All Rights Reserved</p>
        <p>Designed by <strong>Harsheekesh Tiwari</strong></p>
    </div>
</footer>

<?php $conn->close(); ?>
</body>
</html>

```


Main Page: style.css

```
*{
  margin: 0;
  padding: 0;
}
body {
  margin: 0;
  margin-bottom: 0;
  font-family: Arial, sans-serif;
}

header {
  font-family: sans-serif;
  width: 100%;
  display: flex;
  justify-content: space-around;
  align-items: center;
  background-color: #e30513;
  padding: 10px ;
  margin-bottom: 5px;
  color: white;
  /* position: fixed; */
}

#logo{
  border-radius: 5px;
}
.logo {
  display: flex;
  align-items: center;
}

.logo img {
  width: 60px;
  height: 60px;
  margin-right: 10px;
}

.university-info h1 {
  font-size: 20px;
```

```

    margin: 0;
}

.university-info p {
    margin: 0;
    font-size: 12px;
}

nav ul {
    margin-left: 10px;
    list-style: none;
    margin: 0;
    padding: 0;
    display: flex;
}

nav ul li {
    margin-left: 30px;
}

nav ul li a {
    color: rgb(255, 255, 255);
    text-decoration: none;
    font-size: 16px;
    padding: 6px;
    border-radius: -100px;
}

nav ul li a:hover{

    background-color: white;
    color: #000000;
    border-radius: 70px;
    opacity: 1;
    transition: border-radius 1.5s ease;
}

.hero {
    background-image: url(Images/placement-bg.jpg);
    background-size: cover;
    text-align: center;
    margin-top: 0;
    width: 100%;
}

```

```

.hero h2 {
  color: #14213d;
  height: 40px;
  padding: 20px;
  font-size: 48px;
  margin-bottom: 10px;
}

.hero p {
  font-size: 30px;
  padding-bottom: 150px;
  font-weight: bold;
  color: #b5e2fa;
  margin-bottom: 250px;
}

.cta-button {
  background-color: #e30513;
  color: white;
  padding: 15px;
  border-radius: 30px;
  text-decoration: none;
  font-size: 18px;
  /* margin-top: 100px; */
}

.red-data {
  border: 2px solid black;
  border-radius: 20px;
  margin: auto;
  margin-top: 50px;
  background-image: url(Images/reddata.png);
  width: 88%;
  background-repeat: no-repeat;
  height: 250px;
  box-shadow: 5px 10px 10px black;
}

.quote-container {
  display: flex;
  justify-content: center;
  align-items: center;
}

.quote {
  display: flex;
  font-family: "Merriweather", serif;

```

```

    font-weight: 300;
    font-style: normal;
    justify-content: center;
    margin-top: 100px;
    border: 3px solid black;
    border-radius: 30px;
    box-shadow: 5px 10px 10px black;
    width: 60%;
    height: 30px;
    height: fit-content;
    font-size: xx-large;
    transition: width 0.5s ease;
}

.quote:hover
{
    width: 95%;
}

/* */
.container {
    text-align: center;
    margin-top: 100px;
}

h1 {
    font-size: 36px;
    color: white;
    margin-bottom: 20px;
}

#comp{
    color: black;
}

.box-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    margin-bottom: 50px;
}

.box {
    background-color: #f00;
    color: #fff;

```

```

padding: 10px 20px;
margin: 10px;
border-radius: 20px;
font-size: 18px;
text-align: center;
min-width: 150px;
cursor: pointer;
box-shadow: 2px 3px 3px black;
transition: transform 0.5s;
}

.box:hover {
background-color: black;
color: white;
transform: scale(1.1);
}

/*-----CHart----- */

.graph-container{
width: 100%;
height: auto;
margin-top: 50px;
display: flex;
justify-content: center;
transition: width 0.5s ease;
}

.graph-container h1 {

color: #000000;
margin: 50px;
}

.graph-img1 {
margin: auto;
padding: 10px;
display: flex;
flex-direction: row;
align-self: center;
justify-content: space-between;
width: 95%;
border: 5px solid black;
border-radius: 10px;
margin-bottom: 30px;
}

```

```

.graph-img1 img1 img2{

    border: 2px solid black;
    width: 45%;
    height: 100%;
}

.graph-img2{
    margin: auto;
    display: flex;
    flex-direction: row;
    align-self: center;
    justify-content: space-around;
    width: 95%;
    border: 5px solid black;
    border-radius: 10px;
    margin-bottom: 30px;
}
/* -----chart-image----- */

/* top achievers : students */
.top-achievers {
    text-align: center;
    padding: 50px 0;
    z-index: 0;
}

.top-achievers h2 {
    font-size: 36px;
    margin-bottom: 30px;
    color: #000000;
}

.achievers-container {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
    gap: 40px;
}

.achiever {
    position: relative;
    width: 350px;

```

```

    text-align: center;
}

.achiever img {
    width: 180px;
    height: 180px;
    border-radius: 50%;
    object-fit: cover;
}

.achiever img:hover {
    transform: scale(1.1);
    transition: transform 0.5s;
}

.name-tag {
    position: absolute;
    bottom: -20px;
    left: 50%;
    transform: translateX(-50%);
    background-color: #ff1100;
    color: white;
    border-radius: 20px;
    padding: 5px 15px;
    width: 220px;
}

.name-tag p {
    margin: 0;
    font-size: 14px;
    font-weight: bold;
}

.name-tag p:last-child {
    font-size: 12px;
    font-weight: normal;
}

/* Founder Section */
.founder-section {
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 40px;
    max-width: 800px;
    margin: 0 auto;
}

```

```
}

#from-founder{
  color: red;
}

.founder-image img {
  border-radius: 50%;
  width: 150px;
  height: 150px;
  object-fit: cover;
  box-shadow: 20px 40px 80px rgba(233, 9, 9, 0.461);
}

.founder-message {
  margin-left: 20px;
}

.founder-message h2 {
  font-size: 32px;
  margin: 0 0 10px;
}

.founder-message p {
  font-size: 18px;
  line-height: 1.6;
  margin: 0 0 20px;
}

.founder-signature {
  font-weight: bold;
  color: red;
}

.founder-name {
  font-size: 20px;
}

.founder-signature span {
  display: block;
}

.founder-signature small {
  color: #777;
  font-size: 14px;
}
```



```

/* Extra styles for bottom graphics */
.founder-section::after {
  content: "";
  display: block;
  width: 80px;
  height: 80px;
  background-image: url('path-to-graphic.png'); /* Replace with the actual path to your graphic */
  background-size: contain;
  background-repeat: no-repeat;
  position: absolute;
  right: 20px;
  bottom: 20px;
}

footer {
  position: relative;
  background-color: #000000;
  color: #fff;
  padding: 10px;
  text-align: center;
  overflow: hidden;
}

.footer-container {
  position: relative;
  width: 100px;
  height: 5px;
  margin: 0 auto;
}

/* footer */
.footer-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 5px;
}

.social-icons:hover {
  transform: scale(1.1);
  transition: transform 0.5s;
}

.social-icons::after {
  content: "";
  display: block;

```

```

    width: 50px;
    height: 3px;
    background-color: black;
    margin-top: 10px;
}

.contact-info h2 {
    font-size: 24px;
    margin-bottom: 10px;
    color: #fff;
}

.contact-info p {
    margin: 5px 0;
    line-height: 1.6;
}

.contact-info a {
    color: #fff;
    text-decoration: none;
}

.social-icons {
    margin-top: 10px;
}

.social-icons a {
    margin: 0 10px;
    display: inline-block;
}

.social-icons img {
    width: 30px;
    height: 30px;
    /* filter: invert(100%); */
}

.footer-bottom {
    text-align: center;
    margin-top: 20px;
    border-top: 1px solid #000000;
    padding-top: 0px;
}

```

```
.footer-bottom p {  
    margin: 0px;  
    font-size: 14px;  
}
```

```
.footer-bottom strong {  
    color: #f44336;  
}
```

Login Page : login.php

```
<?php
```

```
// Database connection details
```

```
$host = "localhost";
```

```
$user = "root";
```

```
$password = "basu2004";
```

```
$db = "admin";
```

```
$user_type="admin";
```

```
// Create connection
```

```
$conn = new mysqli($host, $user, $password, $db);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

```
// Initialize login variables
```

```
$username = $password = $role = "";
```

```
$login_error = "";
```

```
// Check if form is submitted
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    // Get form data
```

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

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

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

```
// Use prepared statement to prevent SQL injection
```

```
$sql = "SELECT * FROM users WHERE username = ? AND password = ? AND user_type = ?";
```

```
$stmt = $conn->prepare($sql);
```

```
if ($stmt === false) {
```

```
    die("Error preparing the statement: " . $conn->error);
```

```
}
```

```

// Bind parameters
$stmt->bind_param("sss", $username, $password, $role);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    // Successful login
    echo "Login successful!";
    // Redirect to specific role page
    if ($role === "admin") {
        header("Location: post.php");
    } elseif ($role === "faculty") {
        header("Location: faculty_dashboard.php");
    } elseif ($role === "student") {
        header("Location: student_dashboard.php");
    }
    exit();
} else {
    // Failed login
    $login_error = "Invalid username, password, or role!";
}

$stmt->close();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Login Page</title>
    <style>
        /* Basic styling for the login page */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }

        header {
            font-family: sans-serif;
            width: 100%;
            display: flex;

```

```
justify-content:space-around;
align-items: center;
background-color: #e30513;
padding: 10px ;
margin-bottom: 5px;
color: white;
/* position: fixed; */
}
```

```
#logo{
border-radius: 5px;
}
```

```
.logo {
display: flex;
align-items: center;
}
```

```
.logo img {
width: 60px;
height: 60px;
margin-right: 10px;
}
```

```
.university-info h1 {
font-size: 20px;
margin: 0;
}
```

```
.university-info p {
margin: 0;
font-size: 12px;
}
```

```
nav ul {
margin-left: 10px;
list-style: none;
margin: 0;
padding: 0;
display: flex;
}
```

```
nav ul li {
margin-left: 30px;
}
```

```

nav ul li a {
  color: rgb(255, 255, 255);
  text-decoration: none;
  font-size: 16px;
  padding: 6px;
  border-radius: -100px;
}

.login-container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.login-box {
  background-color: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
  width: 300px;
}

.announcement h2 {
  color: red;
  text-align: center;
}

.login-form {
  display: flex;
  flex-direction: column;
}

.login-form input,
.login-form select {
  margin-bottom: 15px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 14px;
}

.login-form button {
  padding: 10px;
}

```

```

        background-color: red;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }

    .login-form button:hover {
        background-color: darkred;
    }
</style>
</head>
<body>

<header>
    <div class="logo">
        
        <div class="university-info">
            <h1>MAHARISHI MARKANDESHWAR</h1>
            <p>(DEEMED TO BE UNIVERSITY)</p>
            <p>Mullana-Ambala, Haryana</p>
            <p>(Accredited by NAAC with Grade 'A++')</p>
        </div>
    </div>
    <nav>
        <ul>
            <li><a
href="http://localhost/Placement%20Record%20Project/Placement%20Record%20Project/Placement%20Record%20Project/index.php#hero">Home</a></li>
            <li><a
href="http://localhost/Placement%20Record%20Project/Placement%20Record%20Project/Placement%20Record%20Project/index.php#top-achievers">Student</a></li>
            <li><a
href="http://localhost/Placement%20Record%20Project/Placement%20Record%20Project/Placement%20Record%20Project/index.php#companies">Company</a></li>
            <li><a
href="http://localhost/Placement%20Record%20Project/Placement%20Record%20Project/Placement%20Record%20Project/Admin-Pannel/login.php">Admin</a></li>
        </ul>
    </nav>
</header>
<div class="login-container">
    <div class="login-box">
        <div class="announcement">

```

```

        <h2>Placement Record</h2>
        <p>Login to manage, update and manipulate</p>
    </div>
    <form class="login-form" method="POST" action="">
        <select name="role" required>
            <option value="">Select Role</option>
            <option value="admin">Admin</option>
        </select>
        <input type="text" name="username" placeholder="Username" required>
        <input type="password" name="password" placeholder="Password" required>
        <button type="submit">Login</button>
        <button type="button" onclick="window.location.href='C:\xampp\htdocs\Placement Record
Project\Placement Record Project\index.php';">Close</button>
    </form>

    <!-- Display PHP login error -->
    <?php if (!empty($login_error)): ?>
        <p style="color: red;"><?php echo $login_error; ?></p>
    <?php endif; ?>
</div>
</div>

</body>
</html>

```

Login Page: style.css

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: 'Arial', sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.login-container {
    background-image: url('background.jpg'); /* Add background image */

```



```

    background-size: cover;
    background-position: center;
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
}

.login-box {
    background-color: rgba(255, 255, 255, 0.8);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);
    width: 100%;
    max-width: 400px;
}

.announcement {
    background-color: rgba(0, 0, 0, 0.7);
    color: #fff;
    padding: 10px;
    border-radius: 5px;
    text-align: center;
    margin-bottom: 20px;
}

.announcement h2 {
    margin-bottom: 10px;
}

.login-form {
    display: flex;
    flex-direction: column;
}

.login-form select,
.login-form input {
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
}

#login-btn {

```

```

padding: 10px;
margin-bottom: 10px;
background-color: #1f221f;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s ease;
}
#close-btn {
padding: 10px;
background-color: #d0021b;
color: white;
border: none;
margin-bottom: 10px;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s ease;
}

#login-btn:hover {
background-color: green;
}

/* Responsive Design */
@media (max-width: 768px) {
.login-box {
max-width: 300px;
}
}

@media (max-width: 480px) {
.login-box {
padding: 15px;
}
}

```

Data Registration: Post.php

```
<?php
// Database connection
$conn = new mysqli("localhost", "root", "basu2004", "admin");

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

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get form data and match with database column names
    $student_name = $_POST['student_name'];
    $course = $_POST['course'];
    $student_email = $_POST['student_email'];
    $company_name = $_POST['company_name'];
    $job_role = $_POST['job_role'];
    $salary_package = $_POST['salary_package'];
    $placement_year = $_POST['placement_year'];

    // Handle file upload for 'student_image'
    if (isset($_FILES['student_image']) && $_FILES['student_image']['error'] == 0) {
        $file_tmp = $_FILES['student_image']['tmp_name'];
        $file_type = mime_content_type($file_tmp);

        // Validate MIME type
        $allowed_types = ['image/jpeg', 'image/png', 'image/jpg'];
        if (!in_array($file_type, $allowed_types)) {
            die("Invalid file type. Only JPEG, PNG, and JPG files are allowed.");
        }

        // Read file contents as binary
        $student_image = addslashes(file_get_contents($file_tmp));
    } else {
        die("Error uploading image: " . $_FILES['student_image']['error']);
    }

    // SQL query to insert the data into the database
    $sql = "INSERT INTO student_placement_record
        (student_name, course, student_email, student_image, company_name, job_role,
        salary_package, placement_year)
```

```

VALUES (?, ?, ?, ?, ?, ?, ?, ?);

// Prepare statement and bind parameters
if ($stmt = $conn->prepare($sql)) {
    $stmt->bind_param(
        "ssssssds", // Data types: string, string, string, string (longblob), string, string, decimal,
string
        $student_name,
        $course,
        $student_email,
        $student_image,
        $company_name,
        $job_role,
        $salary_package,
        $placement_year
    );

    // Execute the statement
    if ($stmt->execute()) {
        echo "Record saved successfully!";
    } else {
        echo "Error: " . $stmt->error;
    }

    // Close the statement
    $stmt->close();
} else {
    echo "Error preparing the statement: " . $conn->error;
}

// Close the database connection
$conn->close();
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Placement Record Updation</title>
    <style>

```

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #fff;  
  color: #000;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 800px;  
  margin: 50px auto;  
  padding: 20px;  
  background-color: #f4f4f4;  
  border-radius: 10px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
}  
  
h2 {  
  color: #d32f2f;  
  text-align: center;  
}  
  
label {  
  font-weight: bold;  
  display: block;  
  margin-top: 20px;  
  color: #d32f2f;  
}  
  
input[type="text"],  
input[type="file"],  
input[type="number"],  
input[type="email"],  
select {  
  width: 100%;  
  padding: 10px;  
  margin-top: 5px;  
  border: 2px solid #d32f2f;  
  border-radius: 5px;  
  box-sizing: border-box;  
}  
  
input[type="submit"] {  
  width: 100%;  
  background-color: #d32f2f;
```

```

        color: #fff;
        padding: 10px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
        margin-top: 20px;
    }

    input[type="submit"]:hover {
        background-color: #b71c1c;
    }

    .responsive-image {
        max-width: 100px;
        height: auto;
    }

    @media screen and (max-width: 600px) {
        .container {
            padding: 10px;
        }

        input[type="submit"] {
            font-size: 14px;
        }
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Update Student Placement Record</h2>
        <form action="post.php" method="POST" enctype="multipart/form-data">
        <label for="student_name">Student Name:</label>
        <input type="text" id="student_name" name="student_name" required>

        <label for="course">Course:</label>
        <input type="text" id="course" name="course" required>

        <label for="student_email">Student Email:</label>
        <input type="email" id="student_email" name="student_email" required>

        <label for="student_image">Upload Student Image:</label>
        <input type="file" id="student_image" name="student_image" accept="image/jpeg, image/png,
image/jpg" required>

```

```

<label for="company_name">Company Name:</label>
<input type="text" id="company_name" name="company_name" required>

<label for="job_role">Job Role:</label>
<input type="text" id="job_role" name="job_role" required>

<label for="salary_package">Salary Package (in Lakhs):</label>
<input type="number" id="salary_package" name="salary_package" required min="0"
step="0.01">

<label for="placement_year">Placement Year:</label>
<select id="placement_year" name="placement_year" required>
  <option value="2024">2024</option>
  <option value="2023">2023</option>
  <option value="2022">2022</option>
</select>

<input type="submit" value="Save Record">
</form>

```

```

</div>

```

```

<!-- JavaScript to handle form submission -->

```

```

<script>
  document.getElementById('placementForm').addEventListener('submit', function(event) {
    event.preventDefault(); // Prevent form from submitting the default way

    // Get form data
    const studentName = document.getElementById('student_name').value;
    const course = document.getElementById('course').value;
    const studentEmail = document.getElementById('student_email').value;
    const companyName = document.getElementById('company_name').value;
    const jobRole = document.getElementById('job_role').value;
    const salaryPackage = document.getElementById('salary_package').value;
    const placementYear = document.getElementById('placement_year').value;

    // Validate data (optional)
    if (!studentName || !course || !studentEmail || !companyName || !jobRole || !salaryPackage ||
!placementYear) {
      alert("Please fill out all required fields.");
      return;
    }
  });

```

```
// Send the form data to the backend (for simplicity, we'll log it to the console here)
const formData = {
  studentName,
  course,
  studentEmail,
  companyName,
  jobRole,
  salaryPackage,
  placementYear,
};

console.log("Form Data Submitted:", formData);

// Simulate saving the data to a database
alert("Student Placement Record saved successfully!");

// You can replace the above with actual AJAX or API call to save the data to your
backend/database
});
</script>
</body>
</html>
```


Testing:

Using XAMPP Server on Localhost for the Placement Record Management System

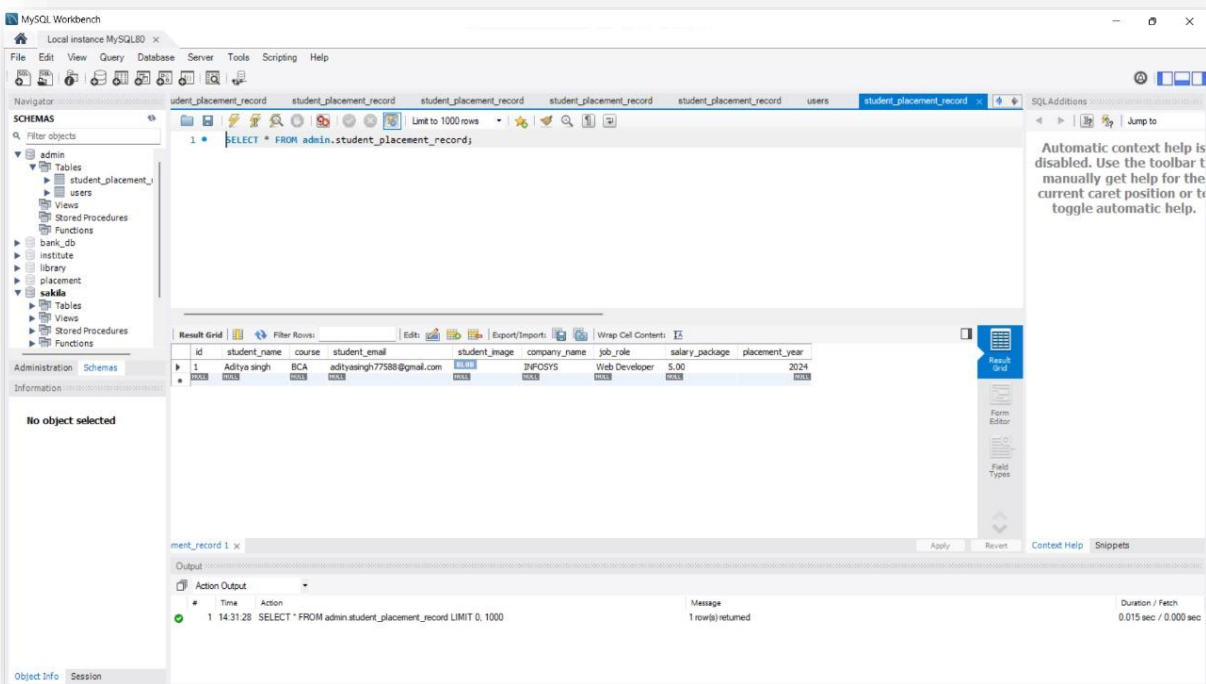
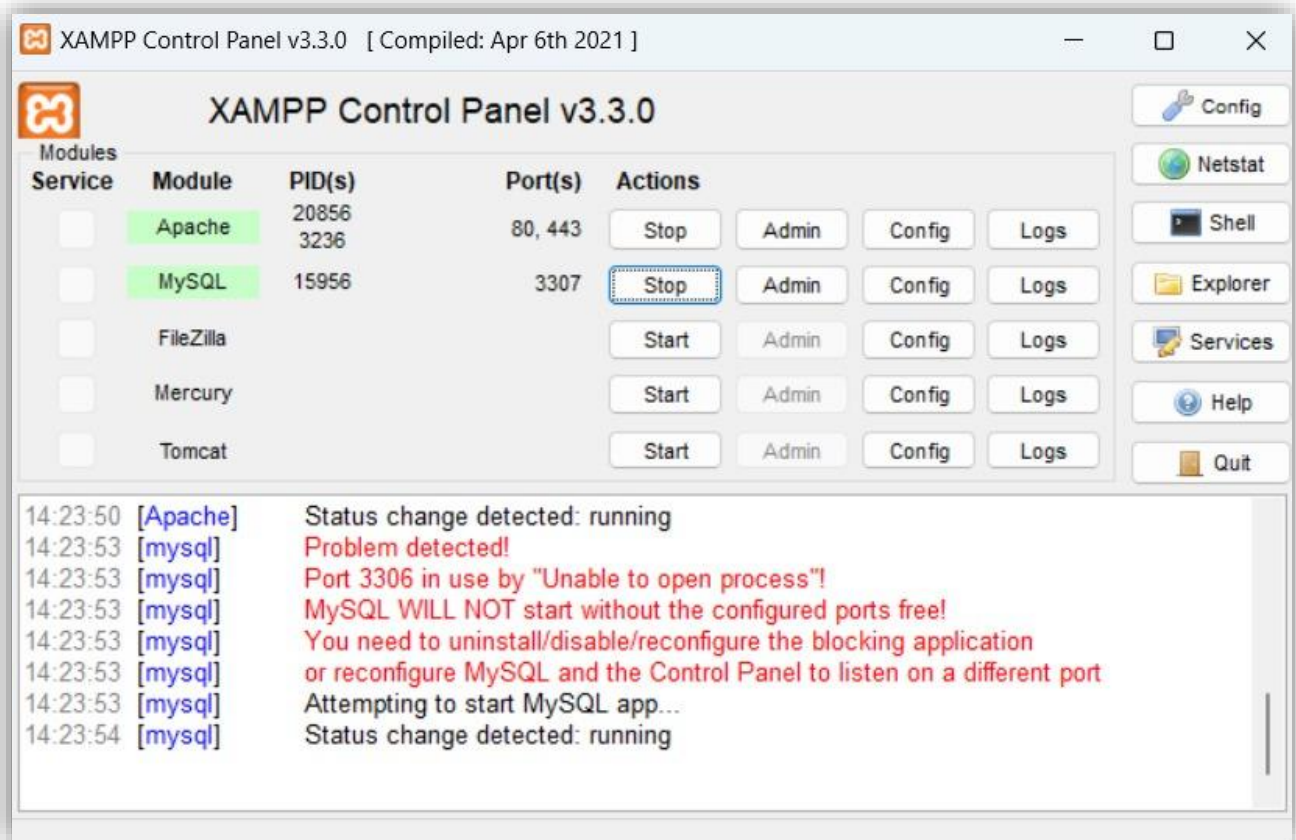
XAMPP (Cross-Platform, Apache, MySQL, PHP, and Perl) is a popular, easy-to-install web server solution that allows developers to run a local server environment on their own computers. When building and testing a web-based project like the Placement Record Management System, XAMPP serves as a critical tool for simulating a live web environment before deploying the system to a real server.

XAMPP:

XAMPP provides an all-in-one package that includes:


- **Apache:** The web server that processes HTTP requests from clients (browsers) and serves HTML, PHP, and other files.
- **MySQL:** A relational database management system used to store and manage the placement records and other data related to the system.
- **PHP:** The server-side scripting language used to generate dynamic content and interact with the MySQL database.
- **phpMyAdmin:** A web-based interface for managing MySQL databases, making it easier to create tables, run queries, and manage data without needing to use the command line.

XAMPP runs all these components on localhost, which refers to the local computer that is running the XAMPP server. This setup simulates the environment of a live web server, enabling developers to build, test, and debug their projects before uploading them to a remote server.



Simulation/Output Screens:





MAHARISHI MARKANDESHWAR
(DEEMED TO BE UNIVERSITY)
Mullana Ambala, Haryana
(Accredited by NAAC with Grade 'A++')

[Home](#)[Student](#)[Company](#)[Admin](#)

Placement Record

Login to manage, update and manipulate

Select Role

Username

Password

Login

Close

Update Student Placement Record

Student Name:

Course:

Student Email:

Upload Student Image:

Choose File

No file chosen

Company Name:

Job Role:

Salary Package (in Lakhs):

Placement Year:

2024

Save Record

Conclusion:

The Placement Record Management System has successfully streamlined the process of managing placement data for universities, providing a centralized, user-friendly platform for administrators, students, and companies. By automating tasks such as student record management, job postings, and real-time access to placement statistics, the system has significantly improved efficiency and accuracy, reducing administrative overhead and minimizing human error. The use of technologies like PHP, MySQL, JavaScript, and XAMPP ensures robust performance, scalability, and secure data management. The system's role-based access controls maintain data integrity, while its flexibility allows for future expansion, such as cloud hosting or AI-driven analytics. In conclusion, this system enhances accessibility, transparency, and the overall placement process, providing a reliable tool that meets the evolving needs of academic institutions and their stakeholders.

Reference:

MMDU Official: <https://www.mmumullana.org/>

MMDU Placement Section: <https://www.mmumullana.org/placement/>