Dr. P. A. Inamdar University, Pune

Allana Institute of Computer Application and IT

Synopsis

on

PMS

Mini Project

By

Abul Fazl Rafeeque Ahmed

MCA I (Sem-3rd)

Roll No: 78

Academic Year 2025-26

Guide Name: Pankaj Kulkarni sir

Objectives:

Secure Authentication & Authorization – Provide a secure login, registration, and password recovery system for users.

Admin Management – Enable admins to perform CRUD operations and manage Student, Admin, Placement Officer.

Student Management – Manage and Update Their Profile and Apply for Opening Added by Placement Officer.

Placement Officer Management – Manage students and Post Job Applications.

User Role-Based Access – Ensure different functionalities for Admin, Student, Placement Officer based on their roles.

Efficient Record Keeping – Maintain systematic records of Student information and Placement Records.

Backend Task – Automatic send emails **SENDGRID** to student who are willing to apply for job will be using **cronjob** for automatic sending email every 1 hrs.

Updated Number of Modules in the Project:

Authentication Module – Handles login, registration, and password recovery.

Admin Module – Manages CRUD operations Student, Admin, Placement Officer.

Placement Officer Module – Add Job Posting and Manage it.

Student Module – Allows doctors to view appointments, update progress, and generate invoices.

Detailed Scope and Functionality of Placement Management System:

Student Management

Students can register and maintain their profile with details like education, skills, and job preferences.

Students can update skills, certifications, and placement status.

Students can view placement notifications and available opportunities.

Students can apply for jobs and track their application history.

Placement Officer Management

Placement officers can post job opportunities with details (company, role, eligibility, salary, deadline).

They can update or delete job postings.

They can manage and update student placement status.

Placement officers can view student profiles and filter them based on eligibility.

Admin Management

Admin can manage user roles (student, placement officer, admin).

Admin has complete access to view, approve, and monitor student and job data.

Admin ensures smooth functioning by monitoring placement activities.

Job & Notification Management

Students receive job notifications based on eligibility and preferences.

Toggle notification settings available for students.

Students can view job details and apply directly.

Placement officers and admins can broadcast important announcements.

Application & Placement Tracking:

Students can track the jobs they have applied for.

Placement officers can update the selection status of students (applied, shortlisted, placed, rejected).

Students can view their complete placement history.

Reports & Analytics

System can generate reports like number of students placed, company-wise placements, and job trends.

Placement officer and admin can analyze placement data for better decision-making.

Admin → Manage overall system, users, and data.

Placement Officer → Manage jobs, placements, and student updates.

Student → Manage own profile, apply for jobs, view opportunities.

Technology Used:

Front-end: React, HTML, CSS (with Vite 7 for bundling).

Back-end: Node JS and Mysql

Operating System: Windows 10 & Above.

Local Host: Chrome Browser , etc.

Role of MySQL in Your Placement Management System:

Storing User Information:
MySQL will store details of users, including admins, placement officers, and students. This data will help in managing user authentication, roles, and personalized access.

Student Records & Placement Data:
The database will store student details, including personal information, education, skills, job preferences, and placement history, ensuring easy access for placement officers and admins.

Placement Officer & Admin Management:
MySQL will keep records of placement officers and admins, along with their roles, permissions, and responsibilities, helping in efficient system management.

Job Postings & Applications:
All job opportunities posted by placement officers will be stored in the database. Students' job applications, their statuses (applied, shortlisted, selected, rejected), and deadlines will be tracked in MySQL for smooth job management.

Notifications & Updates:
The system will store placement notifications and announcements for students. Students can toggle their notification preferences, and the data will be managed in the database.

Placement Tracking & Reports:
MySQL will store student placement updates and history, allowing admins and placement officers to generate reports like total students placed, company-wise placements, and application trends.

User Role Management:
Different roles (admin, placement officer, and student) will have access to specific functionalities based on their database records, ensuring a structured and secure workflow.

Hardware Specifications:

Minimum Hardware Requirements:

Processor:

Intel Core i3 or equivalent.

A basic processor such as the Intel Core i3 is sufficient for running the development environment and testing the project. While higher-end processors (like i5 or i7) can improve performance, i3 is the minimum requirement for a smooth experience in local development.

RAM: 4 GB (minimum).For React development, 4 GB of RAM is sufficient for handling basic tasks like running the code editor, browser, and development server simultaneously. However, more RAM (8 GB or higher) is recommended for handling larger projects or when running multiple tools and processes in parallel.

Storage: At least 128 GB SSD.An SSD (Solid-State Drive) is recommended for faster read/write speeds compared to traditional hard drives. This ensures quicker file access, faster boot times, and improved performance when running the development server or database locally.