

College Placement System Using Python

**Biradar Triveni, Sandeep, Kanakshi, Mohammad Shahbaaz¹
Dr. Sharanbasappa Shetkar²**

¹UG Students of Department of Electronics & Communication,

²Faculty of Department of Electronics & Communication,
Lingaraj Appa Engineering College Bidar, Karnataka.
Emails: sshetkar@gmail.com

ABSTRACT

Campus placement is a critical activity in higher educational institutions that connects students with employment opportunities. Traditional manual placement processes suffer from inefficiencies such as data inconsistency, delayed communication, and lack of transparency. This paper presents the design and implementation of a web-based College Placement Management System using Python and Flask framework with a MySQL database back-end. The proposed system automates the end-to-end placement workflow for students, recruiters, and placement officers through role-based access, centralized data management, and real-time notifications. The system improves operational efficiency, minimizes manual effort, and enhances coordination among stakeholders. Experimental evaluation shows that the proposed system provides a scalable, secure, and efficient platform for modern campus placement management.

Keywords— Placement Management System, Flask, MySQL, Web Application, Campus Recruitment, Automation, Role-Based Access Control.

I. INTRODUCTION

Campus placement plays a vital role in bridging the gap between academic education and professional employment. Educational institutions handle large volumes of student data, recruiter requirements, application tracking, and communication processes. In many colleges, these activities are still managed manually using physical documents, emails, and spreadsheets, leading to inefficiencies, delays, and data inconsistencies.

To overcome these challenges, a web-based Placement Management System is proposed to automate the entire placement life-cycle. The system provides a centralized platform for students, recruiters, and placement officers to interact through structured workflows. The use of Flask as a back-end framework and MySQL as a relational database ensures scalability, data consistency, and secure access control. This system modernizes placement operations by enabling real-time updates, automated notifications, and role-based access to information.

II. RELATED WORK

Several researchers have proposed web-based placement management systems to improve campus recruitment processes. Khandelwal *et al.*

[1] presented a web-based system to automate recruitment workflows with role-based access. Bhuvaneswari *et al.*

[2] emphasized centralized portals for training and placement officers and students. Shradha *et al.*

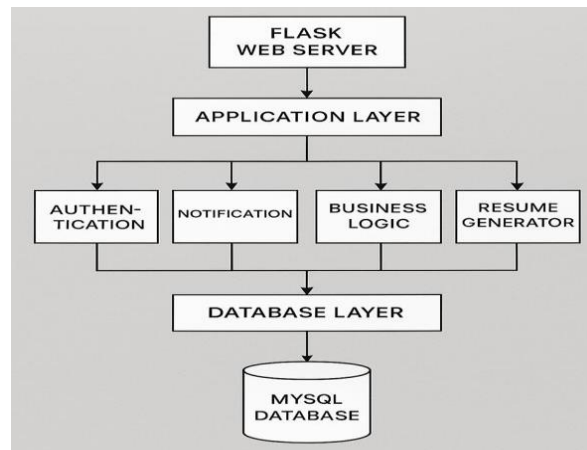
[3] implemented a PHP-MySQL based placement portal for real-time updates. Singh *et al.* [9] proposed a MERN-stack based placement workflow system with analytics and bulk data handling.

These studies highlight the importance of automation, centralized databases, and role-based access. However, many systems lack flexible modular design and lightweight frameworks. The proposed system addresses these gaps using Flask for modular backend design, integrated notification services, and dynamic resume generation.

III. PROBLEM STATEMENT

Manual placement systems result in scattered student records, inconsistent application formats, slow communication, and lack of real-time tracking. Recruiters face difficulties in candidate shortlisting, while placement officers manage large datasets without proper analytical tools. These issues lead to operational inefficiency, reduced transparency, and delayed recruitment decisions.

IV. PROPOSED SYSTEM ARCHITECTURE



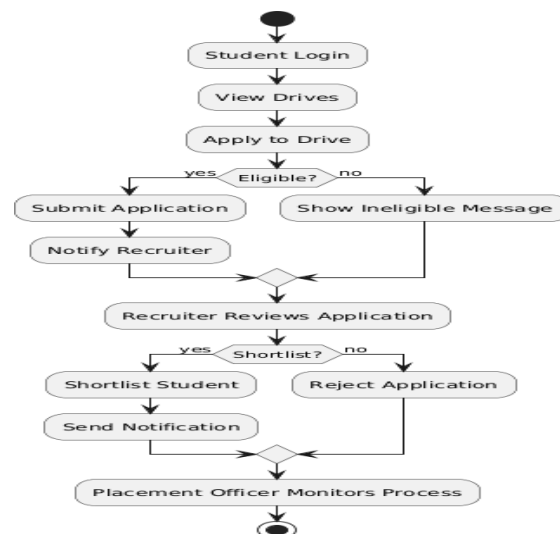
The proposed system follows a layered architecture consisting of:

1. **Presentation Layer:** HTML, CSS, JavaScript templates for user interaction.
2. **Application Layer:** Flask framework handling routing, authentication, business logic, and notifications.
3. **Database Layer:** MySQL database storing users, drives, applications, interviews, results, and notifications.

The architecture supports three user roles:

- Students
- Recruiters
- Placement Officers

Role-based access control ensures secure and restricted access to system features.



V. METHODOLOGY

The system development follows the following stages:

System Analysis: Identification of limitations in manual placement processes and requirement gathering.

System Design: Modular architecture and normalized database schema design.

Backend Development: Flask-based server-side logic and role-based authentication.

Frontend Development: HTML templates using Jinja2 for dynamic content rendering.

Database Integration: MySQL integration with optimized queries and indexing.

Resume Generation: Automated PDF resume generation using ReportLab.

Notification System: Real-time alerts for drives, applications, interviews, and results.

Testing and Validation: Functional, performance, and security testing.

VI. FUNCTIONAL MODULES

A. Student Module

Students can register, update profiles, generate resumes, view placement drives, apply for jobs, and track application status.

B. Recruiter Module

Recruiters can create placement drives, define eligibility criteria, shortlist candidates, schedule interviews, and publish results.

C. Placement Officer Module

Placement officers verify profiles, activate drives, manage recruiters, and monitor analytics and placement statistics.

VII. RESULTS AND DISCUSSION

The implemented system successfully automates placement workflows and improves communication through real-time notifications. Centralized storage reduces data inconsistency and improves retrieval efficiency. Role-based access ensures secure and structured operations. The system demonstrates improved placement cycle time and reduced administrative workload compared to manual processes.

VIII. CONCLUSION

This paper presented a web-based Placement Management System using Flask and MySQL to automate campus recruitment processes. The system enhances transparency, reduces manual effort, and improves coordination among stakeholders.

FUTURE WORK

Future enhancements may include integration of machine learning for candidate recommendation, mobile application support, and advanced analytics dashboards for predictive placement trends.

REFERENCES

- [1] V. Khandelwal *et al.*, “Placement Management System for Campus Recruitment,” 2020.
- [2] [2] R. Bhuvaneswari *et al.*, “A Research on Placement Management System,” 2022.
- [3] [3] S. R. Shradha *et al.*, “Web-Based Application for College Campus Placement,” 2023.
- [4] [4] K. Basha *et al.*, “Web-Based Placement Management System,” 2022.
- [5] [9] M. Singh *et al.*, “Placement Workflow Management System Using MERN,” 2023.