iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2026 **Issue:** Conference **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.83332

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com
National Conference on Innovation in Engineering and Technology-(NCIET2026)

Scalable Architectural Framework for Automated Campus Recruitment Management: A Multi-Module Android and MVI-Based Approach

Neha Rai[1], Pranjal Patil[2], Shruti Patil[3], Atharva Pajgade[4], Rohan Dharia[5]
[1]*Assistant Professor,* [2, 3, 4, 5]*Student, Department of Computer Engineering, All India Shri Shivaji Memorial Society College of Engineering Pune, Maharashtra, India-411001*

Abstract: ***The transition from academic environments to professional careers is a critical stage in higher education. Traditionally, Training and Placement Offices (TPOs) rely on fragmented manual tools such as spreadsheets, emails, and social messaging platforms to manage recruitment activities, often resulting in communication gaps, data inconsistency, and increased administrative effort.***
This paper presents "Clave," an Android-based campus recruitment management system designed to streamline and automate the recruitment process. The proposed solution employs a multi-layered backend architecture and a modular frontend based on Model-View-Intent (MVI) and Model-View-ViewModel (MVVM) architectural patterns to ensure scalability, maintainability, and efficient state management.
The system incorporates secure OTP-based authentication, JSON Web Token (JWT) authorization, and a flexible cloud-based data model powered by MongoDB Atlas. By centralizing student, recruiter, and placement cell interactions, the platform enhances communication, simplifies application tracking, and reduces dependence on manual processes. The proposed architecture provides a secure, scalable, and modern framework for managing campus recruitment activities and serves as a foundation for future intelligent placement management systems.
Keywords: Android Application, Campus Recruitment Management, Multi-Module Architecture, MVI Pattern, MVVM, MongoDB Atlas, Recruitment Automation.

I. INTRODUCTION

The transition from academic education to professional employment is a critical phase in a student's career. In large educational institutions, managing campus recruitment activities involves coordinating thousands of students, recruiters, and placement administrators. Traditional recruitment processes often depend on fragmented communication channels such as spreadsheets, emails, and social messaging platforms. These approaches frequently result in communication gaps, data inconsistency, delayed information dissemination, and increased administrative workload.

The lack of a centralized recruitment platform creates significant operational challenges for Training and Placement Offices (TPOs). Placement coordinators spend considerable effort managing student records, verifying eligibility criteria, and communicating recruitment updates manually. Furthermore, the absence of real-time analytics limits the ability of institutions to monitor placement trends and make data-driven decisions.

To address these challenges, this paper presents "Clave," an Android-based campus recruitment management system designed to automate and streamline placement-related activities. The proposed solution employs a modular software architecture based on Model-View-Intent (MVI) and Model-View-ViewModel (MVVM) patterns, supported by a scalable backend infrastructure using MongoDB Atlas.

The platform integrates secure OTP-based authentication, JWT-based authorization, centralized job management, and automated application tracking to improve the efficiency of campus recruitment processes.

The objective of the proposed system is to provide a secure, scalable, and user-friendly platform that enhances communication among students, recruiters, and placement coordinators while reducing dependence on manual administrative procedures.

©IJRASET: All Rights are Reserved

A. Problem Statement

Despite the increasing number of students participating in campus recruitment processes, most Training and Placement Offices (TPOs) still rely on fragmented systems such as spreadsheets, emails, and messaging platforms. These systems lack real-time synchronization, centralized data management, and automated eligibility filtering, resulting in inefficient communication, delayed responses, and increased administrative workload.

Furthermore, the absence of intelligent analytics prevents institutions from deriving meaningful insights into recruitment trends. Therefore, there is a need for a scalable, automated, and mobile-first solution that integrates all placement-related activities into a unified platform.

B. Contributions of the Proposed System:

The primary contributions of the proposed Clave platform are as follows:

1) Development of a centralized Android-based platform for campus recruitment management.
2) Integration of secure OTP-based authentication and JWT-based authorization mechanisms to ensure data security.
3) Adoption of MVI and MVVM architectural patterns to improve application scalability, maintainability, and state management.
4) Implementation of a cloud-based backend infrastructure using MongoDB Atlas for efficient data storage and synchronization.
5) Automation of recruitment workflows, including job posting, eligibility verification, application tracking, and notification management.
6) Design of a modular architecture that supports future enhancements and integration with advanced recruitment technologies.

II. LITERATURE REVIEW

The digitization of placement management has progressed from simple web-based repositories to sophisticated mobile-first applications.

Parmar et al. [1] developed a basic Android application for student registration and resume uploads. While it successfully reduced physical paperwork, the system lacked advanced analytics and real-time synchronization. Kamble et al. [2] improved upon this by utilizing Firebase for real-time notifications, though the research did not address high-traffic scalability or modular software design.

More recently, Dubey and Narang [3] proposed theoretical machine learning models for candidate matching, but practical implementation in live university ecosystems remained limited. Mali et al. [4] examined smart placement systems using sentiment analysis but highlighted that the lack of real-time cloud synchronization remains a significant barrier to adoption. Furthermore, studies by Singh and Verma [5] and Gupta [6] emphasize that automated filtering and mobile-first accessibility are essential for reducing the administrative burden on institutional staff.

Recent work by Patil et al. [15] explored a centralized Android-based campus recruitment platform and highlighted the importance of secure authentication, cloud-based infrastructure, and recruitment workflow automation in educational institutions.

Clave synthesizes these findings, addressing previous limitations by employing a modern, reactive multi-module Android architecture combined with a robust, cloud-native backend.

A. Research Gap

The review of existing literature indicates that while several placement management solutions have addressed individual aspects such as student registration, notification delivery, and candidate matching, few systems provide a unified platform combining secure authentication, automated eligibility verification, real-time synchronization, modular Android architecture, and cloud-native scalability.

Additionally, limited research has focused on integrating modern architectural patterns such as MVI and MVVM within campus recruitment applications. The proposed Clave system addresses these gaps by offering a centralized, scalable, and mobile-first recruitment management platform.

III.SYSTEM ARCHITECTURE

The Clave system is predicated on modularity and separation of concerns to ensure institutional-level scalability.

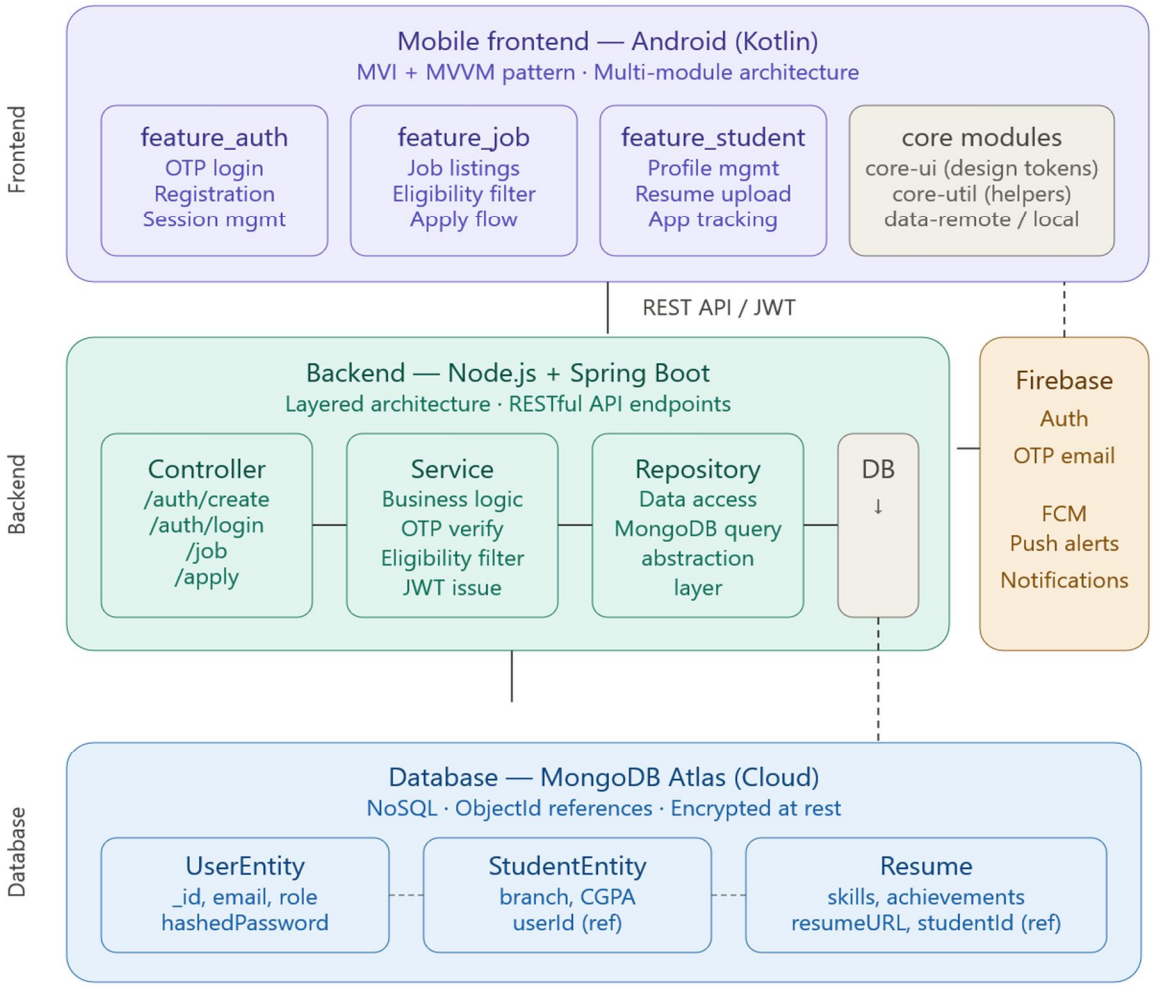


Fig 1: System Architecture

A. Backend: Layered Persistence and Logic

The backend architecture follows a Database → Repository → Service → Controller flow.

- Controller: Manages RESTful API endpoints (e.g., /auth/create, /auth/login, /job) and maps incoming HTTP requests to specific service methods.
- Service: Implements core business logic, such as automated eligibility filtering and OTP verification processing.
- Repository: Acts as a data access layer, abstracting MongoDB queries from the services to facilitate database scaling.

B. Frontend: Multi-Module Android Structure

As identified in the project structure (Image 4 & 5), the Android application is partitioned into independent modules to optimize build times and prevent dependency entanglement:

- Core Modules: The core-ui and core-util modules house shared design tokens, Figma-based standardized UI components, and common utility functions.
- Data Modules: The data-local and data-remote modules handle persistent storage and network operations, isolating API logic from the UI.

- Feature Modules: Individual user flows are encapsulated in modules like feature_auth for authentication, feature_job for listings, and feature_student for profile tracking.

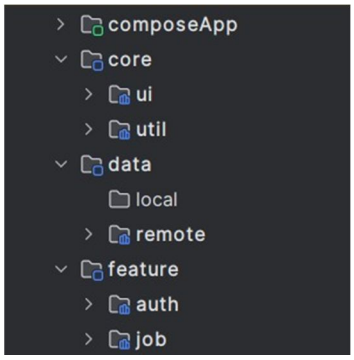


Fig. 2. Multi-module project architecture showing core, data, and feature modules.

C. Pattern Synergy: MVI and MVVM

The application employs a hybrid of MVI and MVVM patterns (Image 3). MVVM handles the screen lifecycle and preserves state during configuration changes. Simultaneously, the Model-View-Intent (MVI) pattern enforces a unidirectional data flow (Image 1, Image 2):

- Intent: User actions (e.g., "Apply for Job") are captured as intents.
- Presenter/Intent Handler: Processes the intent through the domain layer.
- State Update: The system produces an immutable "New State."
- View
- Rendering: The UI renders the new state, eliminating race conditions and ensuring predictable behavior.

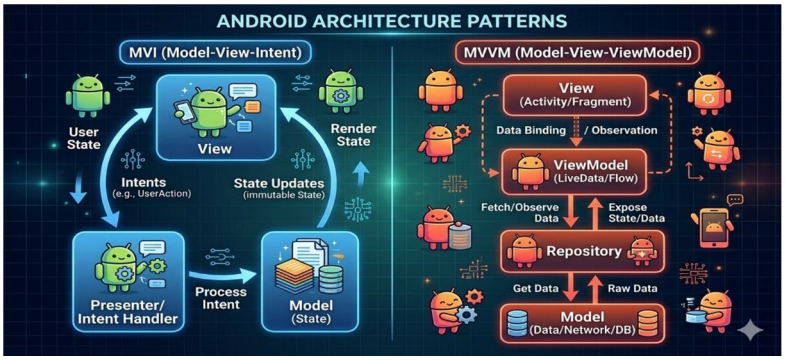


Fig. 3. Architectural comparison: MVI Unidirectional Flow vs. MVVM Observation Pattern.

The combination of layered backend architecture, modular Android development, and modern architectural patterns enables the Clave platform to achieve high maintainability, scalability, and separation of concerns. This design approach facilitates future feature expansion while ensuring consistent application performance across different user workflows.

IV. SYSTEM WORKFLOW

The Clave system follows a structured workflow to ensure seamless interaction between students, administrators, and recruiters.

Step 1: User Registration and Authentication

Students register using their institutional credentials and verify their identity through OTP-based authentication.

Step 2: Profile Creation

Users complete their profiles by adding academic details, skills, and resumes.

Step 3: Job Posting

The Training and Placement Office (TPO) uploads job opportunities with specific eligibility criteria such as CGPA, branch, and required skills.

Step 4: Eligibility Filtering
The system automatically filters eligible students based on predefined criteria.
Step 5: Notification System
Eligible students receive real-time notifications via push notifications.
Step 6: Application Submission
Students apply for jobs directly through the platform.
Step 7: Data Tracking and Analytics
The system tracks applications and generates reports for administrators.

This workflow ensures automation, reduces manual intervention, and improves overall efficiency.

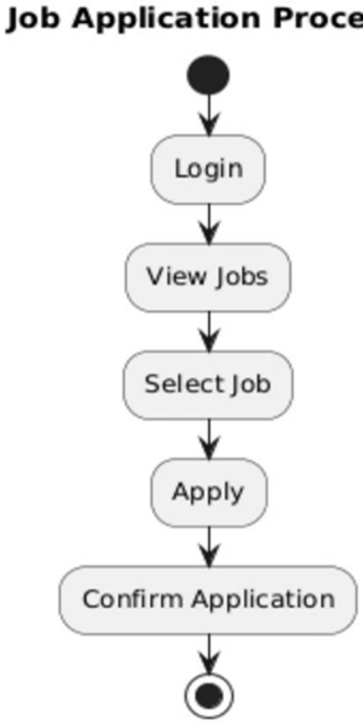


Fig. 4. Activity Diagram of the Clave Campus Recruitment Workflow

V. TECHNICAL IMPLEMENTATION

A. Authentication and Session Security

Given the sensitive nature of academic data, Clave implements a two-factor verification process:

- OTP Verification: Users enter a roll number and institutional email to receive a code via email.
- JWT Management: Upon validation, the backend issues JSON Web Tokens (JWT) for secure, stateless session management. Sensitive database fields are hashed to prevent unauthorized access.

B. Database Schema Design

Clave utilizes a collection-based schema in MongoDB Atlas, linked via **ObjectId** references:

- UserEntity: Root entity for authentication and identity.
- StudentEntity: Manages branch-specific details, CGPA, and contact data.
- Resume: Stores skills, achievements, and URLs for student profiles.

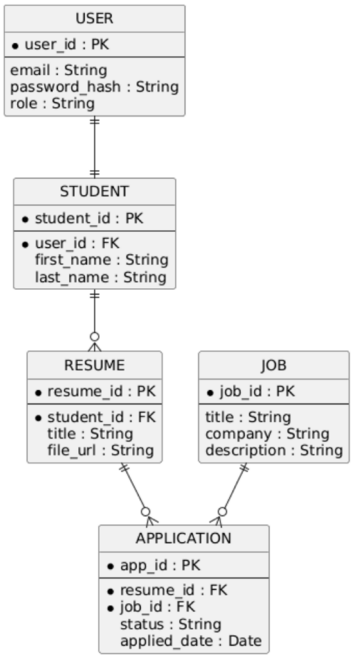


Fig. 5. Database Schema of the Clave System

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Performance Metrics:

Testing conducted on the Clave prototype showed the system meets industry standards for high-performance mobile applications.

Metric	Observed Outcome	Target/Benchmark
App Load Time	Fast Application Startup	Instant access for mobile users
API Response Time	Low Latency Communication	Optimized for real-time interaction
Notification Delay	Near Real-Time Delivery	Instant delivery via Firebase

Table 1. Performance Evaluation of the Clave System

B. Operational Efficiency Gains

A comparative study revealed dramatic improvements over traditional manual systems.

Parameter	Manual (WhatsApp/Email)	Clave Android System
Data Integrity	Prone to human error	Centralized cloud storage
Communication	Fragmented & Manual	Automated Push Notifications
Reporting	Manual Excel compilation	Automated Analytics Dashboard
Response Time	Delayed Information Flow	Faster Information Delivery
Staff Workload	High Administrative Effort	Reduced Manual Intervention

Table 2. Comparison Of Traditional And Proposed Recruitment Systems

The proposed system demonstrates the potential to improve communication efficiency, automate routine administrative tasks, and provide a centralized platform for managing recruitment activities. By integrating real-time notifications, automated eligibility verification, and cloud-based data management, Clave offers a more structured and scalable alternative to traditional placement management approaches

VII. CONCLUSION

The Clave Campus Placement App presents a modern and centralized approach to managing campus recruitment activities. By integrating student registration, profile management, job posting, eligibility verification, application tracking, and notification services into a single platform, the system reduces the complexity associated with traditional placement processes. The adoption of a modular Android architecture based on MVI and MVVM patterns enhances maintainability, scalability, and application performance. Furthermore, the integration of MongoDB Atlas, OTP-based authentication, and JWT authorization ensures secure and efficient data management. The proposed solution provides a strong foundation for digital transformation in campus recruitment and can be effectively adopted by educational institutions seeking to streamline placement operations.

VIII. FUTURE SCOPE

Future enhancements of the Clave system will focus on integrating advanced technologies to improve functionality and user experience. Artificial Intelligence and machine learning techniques can be incorporated for automated resume screening, candidate ranking, and predictive placement analytics. Intelligent recommendation systems can be developed to suggest suitable job opportunities based on student skills, academic performance, and interests. The platform can also be extended to iOS and web environments to provide cross-platform accessibility. Additional features such as real-time analytics dashboards, NLP-based resume parsing, and advanced reporting capabilities can further transform Clave into a comprehensive recruitment intelligence platform.

IX. ACKNOWLEDGMENT

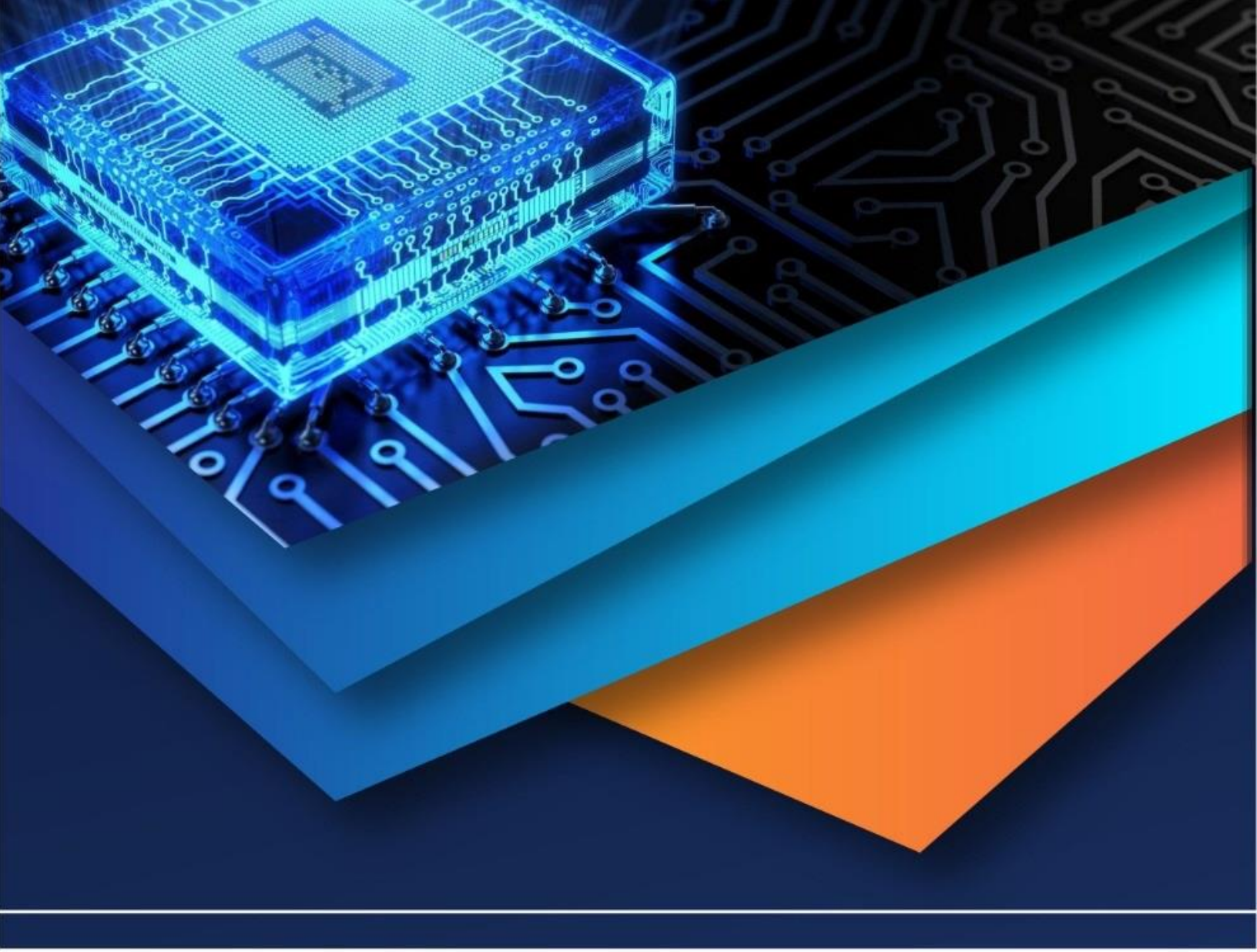
We express our sincere gratitude to our project guide, Prof. Neha Rai, Department of Computer Engineering, AISSMS College of Engineering, Pune, for her valuable guidance, support, and encouragement throughout this project.

We are thankful to Dr. D. P. Gaikwad, Head of the Department, and Dr. D. S. Bormane, Principal, AISSMS College of Engineering, Pune, for providing the necessary facilities and support. We also thank all faculty members, our parents, friends, and classmates for their constant motivation and encouragement in the successful completion of this work.

REFERENCES

[1] K. Parmar, et al., "Android College Campus Placement Every Student's Dream," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 6, no. 4, Apr. 2019.
[2] D. R. Kamble, et al., "Campus Recruitment System Using Android App," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 2, no. 2, Jul. 2022.
[3] M. K. Dubey and B. Narang, "Online Campus Recruitment System – A Machine Learning Model," International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol. 11, no. 10S, Oct. 2023.
[4] J. N. Mali, et al., "A Review: Smart Campus Placement System," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), vol. 6, no. 10, Oct. 2024.
[5] Singh and Verma, "Web-Based Placement Automation and Data Management," ScienceXcel Journal, vol. 12, no. 2, 2019.
[6] Gupta, "Performance Analysis of Mobile-Based Institutional Portals," International Journal of Creative Research Thoughts (IJCRT), vol. 8, no. 3, 2020.
[7] R. Sharma, et al., "Cloud-Based Student Management Systems," IEEE Access, vol. 9, 2021.
[8] P. Kulkarni and A. Shah, "Secure Authentication Techniques for Mobile Applications," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 10, no. 5, 2022.
[9] A. Patil, et al., "Firebase Notification Systems in Android Applications," International Journal of Engineering Research and Technology (IJERT), vol. 10, no. 7, 2021.
[10] S. Roy and K. Das, "NoSQL Database Applications in Educational Platforms," Springer Journal of Computing, 2020.
[11] J. Peterson, "Modern Mobile UI/UX Design Principles," Elsevier Computer Systems, 2021.
[12] T. Wilson, et al., "Agile Development Methodologies in Software Projects," IEEE Conference Publications, 2019.
[13] M. Fernandes, "Role-Based Access Control in Web Applications," ACM Digital Library, 2020.
[14] N. Kapoor, et al., "Big Data Analytics in Recruitment Systems," IEEE Transactions on Data Engineering, 2023.
[15] P. Patil, S. Patil, A. Pajgade, and R. Dharia, "Campus Placement App: A Centralized Android-Based Solution for Automated College Recruitment Management," International Journal of Innovative Research and Technology (IJIRT), vol. 12, no. 6, Nov. 2025.
[16] A. Sharma and R. Kulkarni, "AI-Driven Recruitment and Placement Management Systems in Higher Education," International Journal of Scientific Research in Computer Science Engineering and Information Technology (IJSRCSEIT), vol. 11, no. 2, 2025.
[17] S. Mehta, P. Deshmukh, and V. Patil, "Cloud-Native Mobile Applications for Educational Management Systems," International Journal of Computer Applications (IJCA), vol. 187, no. 5, 2025.
[18] K. Joshi and M. Shah, "Modern Android Architecture Using MVVM and MVI Patterns for Scalable Applications," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 1, 2025.
[19] R. Verma and D. Singh, "Secure Authentication Frameworks for Mobile-Based Academic Platforms," Journal of Information Security and Applications, vol. 82, 2025.
[20] A. Gupta, S. Nair, and P. Rao, "Predictive Analytics and Intelligent Recommendation Systems in Campus Recruitment Platforms," International Journal of Data Science and Analytics, vol. 14, no. 3, 2025.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)