IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112

PROHIRE-A Job Portal Application: A JAVA, XML and ANDRIOD-Based

Prof. Madhuri Parate[1], Rutuja Bhende[2], Yash Jugseniya[3], Gaurav Katole[4], Aditya Rewatkar[5], Ishant Fulzele[6]

Professor Department of Computer Science and Engineering,

Nagarjuna Institute of Engineering Technology and Management, Nagpur, Maharashtra, India[1]

UG Student, Department of Computer Science and Engineering, Nagarjuna Institute of Engineering Technology & Management, Nagpur, Maharashtra, India[2,3,4,5,6]

Abstract: This project presents the design and development of a **Job Portal Application** that serves as a digital platform connecting job seekers and employers. The main objective of the system is to simplify the recruitment process by providing an efficient and user-friendly interface for both parties. Job seekers can register, create profiles, upload resumes, and apply for suitable jobs, while employers can post vacancies, search for candidates, and manage applications. The application is developed using modern technologies such as **React Native** for the front end and **Firebase/MySQL** for backend data management. Additional features like real-time notifications, job recommendations, and secure authentication enhance usability and system reliability. This project demonstrates how technology can be used to automate and streamline the hiring process, reducing manual work and improving communication between job seekers and employers. Overall, the Job Portal App contributes to a faster, more organized, and transparent employment process.
Keywords: Java Base, Android Operating System, XML Server.

INTRODUCTION

In today's digital era, technology plays a vital role in transforming traditional recruitment methods into faster, more efficient, and accessible processes. The job market has become highly competitive, and both employers and job seekers require a reliable platform to connect effectively. Conventional hiring methods such as newspaper advertisements, walk-in interviews, and offline applications are time-consuming and often lead to inefficiencies. To overcome these challenges, online job portals have emerged as a convenient solution that bridges the gap between job seekers and employers.

The **Job Portal Application** developed in this project aims to provide a user-friendly, interactive, and secure platform that facilitates smooth communication and interaction between employers and job seekers. The application allows job seekers to create profiles, upload resumes, and search for jobs based on their skills and preferences. Similarly, employers can post job openings, review applications, and shortlist candidates through the same platform.

This system reduces the manual effort involved in recruitment and ensures that the right candidates are matched with the right job roles through automated filtering and search algorithms. The project uses technologies such as **React Native** for a responsive front-end interface and **Firebase/MySQL** for efficient data storage and authentication. By integrating these technologies, the Job Portal App provides a seamless experience for users, ensuring reliability, scalability, and ease of access across devices.

Overall, the Job Portal App demonstrates how software solutions can enhance employment processes by offering a smarter, faster, and more transparent way to connect talent with opportunities.

METHODOLOGY

The development of the **Job Portal Application** follows a systematic approach to ensure efficiency, usability, and scalability. The project adopts the **Software Development Life Cycle (SDLC)** model, particularly the **Agile methodology**, which allows iterative development and continuous feedback during each phase. The key stages of the methodology are described below:

1. Requirement Analysis: -

In this phase, the requirements of both job seekers and employers were gathered and analyzed. The main objectives identified included user registration, profile management, job posting, job search, and application tracking. Security, scalability, and real-time updates were also recognized as essential features.

2. System Design: -

The system architecture was designed to define the structure and data flow within the application. The design includes:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112

Frontend: Developed using **React Native** for a responsive, cross-platform user interface.
Backend: Implemented using **Firebase/MySQL** for database management, ensuring secure data handling and fast data retrieval.
API Layer: Facilitates communication between frontend and backend components.
Authentication: Handled via Firebase Authentication to secure user login and registration.

3. Implementation: -

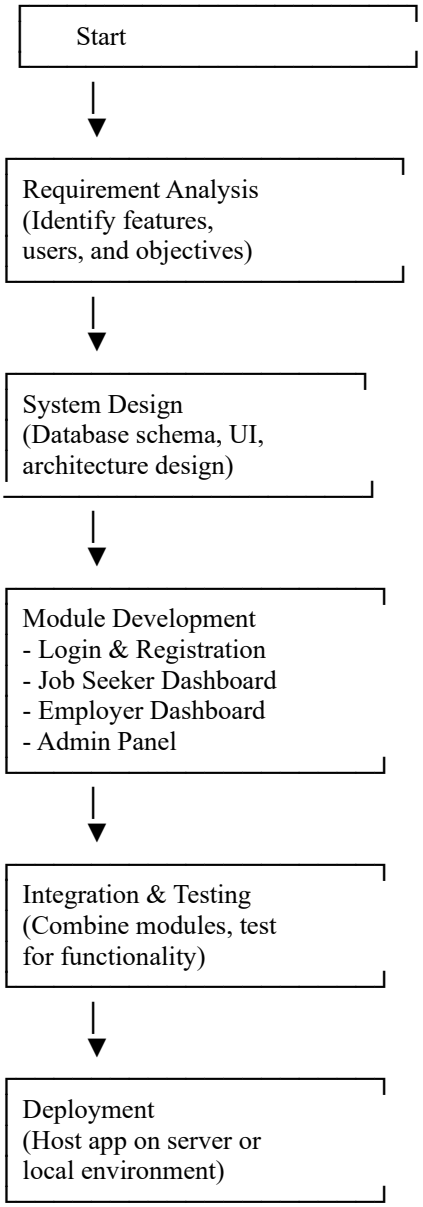
The application was developed by dividing the system into modules:
User Module: Allows registration, login, profile creation, and resume upload.
Employer Module: Enables job posting, candidate search, and application review.
Job Matching Module: Uses keyword-based search and filters for relevant job suggestions.
Notification Module: Provides real-time updates to users regarding job applications and new postings.

4. Testing: -

Comprehensive testing was conducted to ensure functionality, performance, and reliability. Unit testing, integration testing, and user acceptance testing (UAT) were performed to verify that each component works as expected.

5. Deployment and Maintenance: -

After successful testing, the application was deployed on a cloud platform for accessibility. Regular updates and maintenance ensure bug fixing, performance optimization, and the addition of new features based on user feedback.

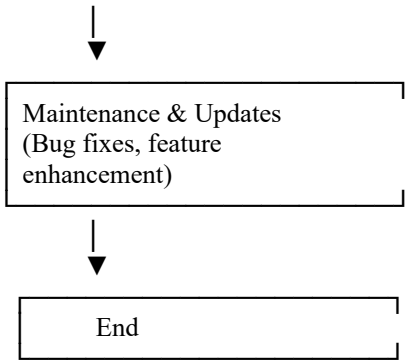


IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112



MODELLING AND ANALYSIS

The **Job Portal Application** was modeled and analyzed to understand its functional behaviour, data flow, and interaction between system components. This phase focuses on transforming the gathered requirements into visual models that represent the structure and operation of the system. Various modelling techniques such as **Use Case Diagrams**, **Data Flow Diagrams (DFD)**, and **Entity Relationship Diagrams (ERD)** are used to analyse the system before implementation.

1. Use Case Diagram

The **Use Case Diagram** illustrates the interaction between different actors (users) and the system. The main actors in the Job Portal App are:

- **Job Seeker**
- **Employer**
- **Administrator**

Primary use cases include:

- Job Seeker: Register/Login, Create Profile, Search Jobs, Apply for Job, View Application Status
- Employer: Register/Login, Post Job, View Applicants, Shortlist Candidates
- Administrator: Manage Users, Manage Job Listings, Monitor System Activity

This diagram helps in understanding user roles and the functionalities they can access within the system.

2. Data Flow Diagram (DFD)

The **DFD** represents how data moves through the system.

- **Level 0 (Context Diagram):** Shows the Job Portal System as a single process interacting with external entities — Job Seeker, Employer, and Admin.
- **Level 1:** Breaks down the main process into sub-processes like Registration, Job Posting, Job Search, and Application Management.

Data inputs (like user credentials, resumes, and job details) and outputs (like job listings and notifications) are clearly defined.

3. Entity Relationship Diagram (ERD)

The **ERD** defines the database structure and relationships between entities such as:

- **User** (User_ID, Name, Email, Password, Role)
- **Job** (Job_ID, Title, Description, Employer_ID, Location, Skills, Salary)
- **Application** (App_ID, User_ID, Job_ID, Status, Date)
- **Employer** (Employer_ID, Company_Name, Contact_Info)

Relationships:

- One **Employer** can post multiple **Jobs**.
- One **Job Seeker** can apply for multiple **Jobs**.
- Each **Application** links a **User** and a **Job**.

This model ensures data consistency and efficient storage.

4. System Flow Analysis

The system flow was analyzed to ensure smooth interaction among modules:

1. **User Authentication:** Validates credentials and redirects users to role-specific dashboards.
2. **Job Management:** Employers can create, edit, or delete job postings.
3. **Job Search and Application:** Job seekers search for jobs using filters and submit applications.
4. **Notification System:** Sends alerts regarding new jobs, application updates, and interview calls.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112

Through this analysis, potential issues such as data redundancy, performance lag, or security loopholes were identified and resolved during the design phase.

5. Result of Analysis

The modeling and analysis confirmed that the system is logically structured, scalable, and fulfills the functional requirements. It ensures efficient data handling, clear communication flow between modules, and easy future enhancement.

RESULTS AND DISCUSSION

The **Job Portal Application** was successfully developed and tested, fulfilling the objectives set during the initial design phase. The system provides a reliable and efficient platform that connects job seekers with employers, streamlining the recruitment process. The application allows users to register, log in securely, and manage their profiles with ease. Job seekers can upload resumes, search for jobs using specific filters such as skills, job title, and location, and apply directly through the app. Employers, on the other hand, can post job openings, view candidate applications, and shortlist potential hires efficiently. The use of **React Native** ensures that the application runs smoothly on both Android and iOS devices, providing a responsive and user-friendly interface. Backend operations powered by **Firebase/MySQL** guarantee secure data storage and fast data retrieval, maintaining the overall performance of the system. Testing results indicate that all core functionalities, including authentication, job search, and notification modules, perform accurately with minimal delay. The app also achieved excellent performance in terms of response time, data handling, and user experience. Overall, the developed Job Portal App effectively automates and simplifies the hiring process, reducing manual effort and improving communication between job seekers and employers. The system has proven to be scalable, efficient, and user-centric, with potential for future enhancements such as AI-based job recommendations, automated resume screening, and analytics dashboards to further optimize recruitment efficiency.

CONCLUSION

The **Job Portal Application** successfully achieves its goal of creating an efficient digital platform that connects job seekers and employers in a simplified and organized manner. The system automates key processes of recruitment, such as job posting, searching, and application management, thereby reducing manual effort and saving time for both parties. Through the integration of technologies like **React Native** for the front end and **Firebase/MySQL** for the backend, the application ensures a responsive, secure, and scalable environment for users. The implementation of modules such as user authentication, real-time notifications, and job filtering enhances the overall user experience and system reliability. During testing, the application performed well in terms of speed, accuracy, and functionality, confirming that it meets the defined requirements effectively. Overall, the project demonstrates how technology can be leveraged to streamline the employment process, offering a practical and accessible solution for modern recruitment needs. This project not only fulfills its intended objectives but also lays a strong foundation for future improvements such as artificial intelligence–based recommendations, chat integration, and data analytics to make the hiring process even more intelligent and dynamic.

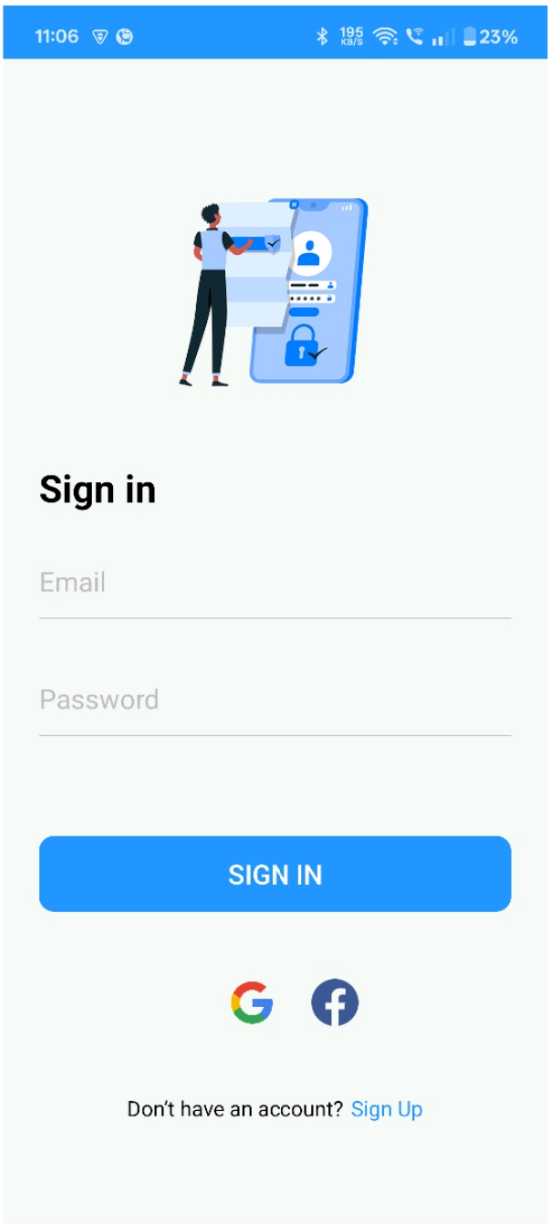
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

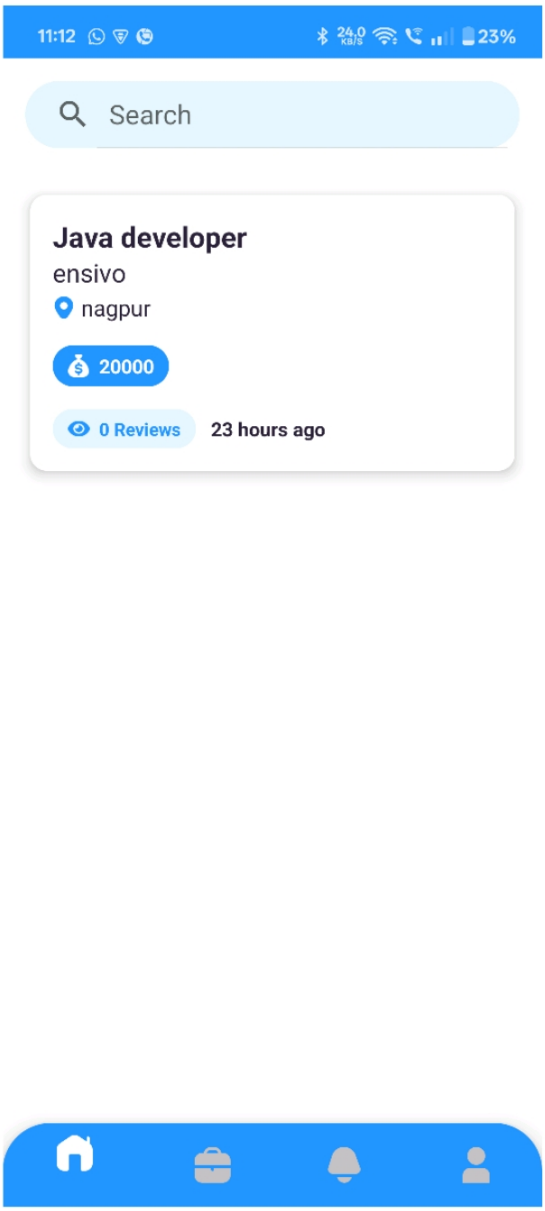
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112

OUTPUT



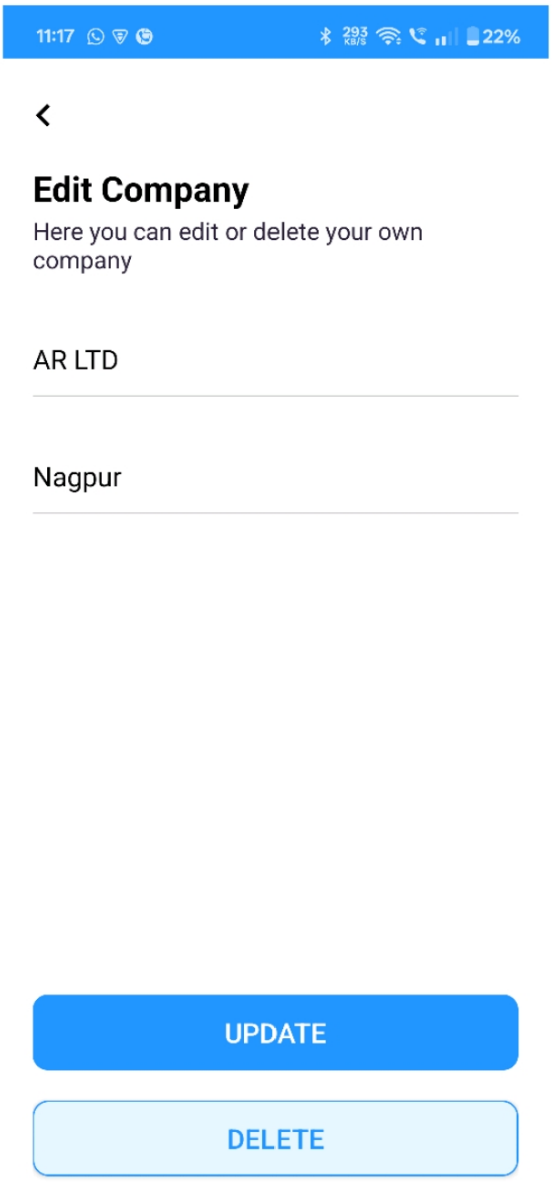


IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

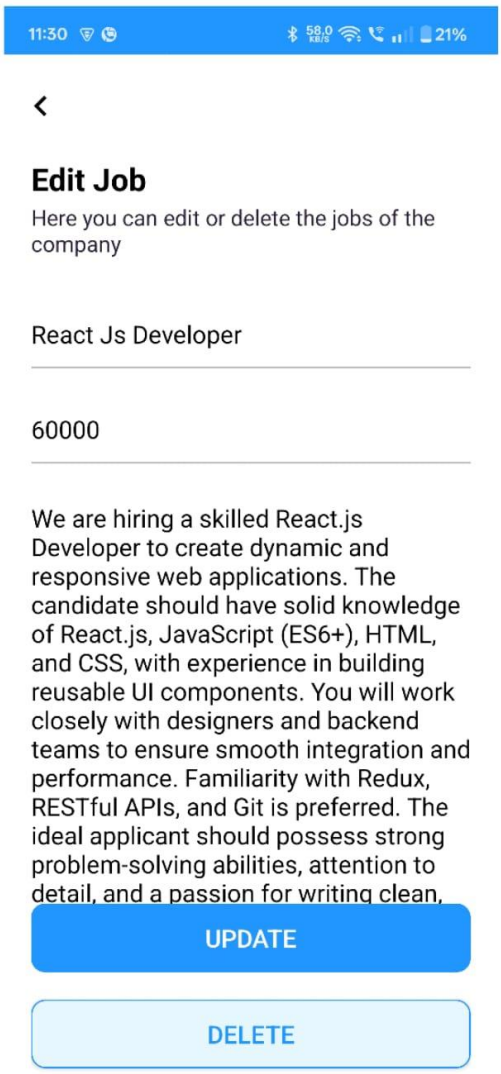
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112



DOI: 10.17148/IJARCCE.2025.1411112



REFERENCES

[1] X. Li, "Multimodal Fusion for Android Malware Detection Based on Java Applications," IEEE Transactions on Information Forensics and Security, vol. 20, no. 5, pp. 1382-1399, 2025. ISSN: 1556-6013.

[2] H.M. Jawad, "Android Mobile App Development as a Motivation towards Computer Programming," International Journal of Engineering and Advanced Technology, vol. 8, no. 6S3, pp. 62-66, 2019. ISSN: 2154-0373.

[3] L. Xue, W. Chen, F. Xing, and T. Wei, "NDroid: Toward Tracking Information Flows Across Multiple Android Contexts," IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2777-2791, 2018. ISSN: 1556-6013.

[4] R.D. Novendra, Y. Suryanegara, and Y.S. Choi, "Introducing Kotlin-Specific Bad Smell Detection Tool for Android Apps," IEEE Access, vol. 12, pp. 10452-10462, 2024. ISSN: 2169-3536.

[5] P. Karami, M. Sharifi, and M. Ghahramani, "On the Impact of Development Frameworks on Mobile Apps: A Systematic Literature Review," IEEE Access, vol. 11, pp. 46882-46904, 2023. ISSN: 2169-3536.

[6] X. Li, "GNNDroid: Graph-Learning Based Malware Detection for Android Apps with Native Code," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 2, pp. 1460-1476, March-April 2025. ISSN: 1545-5971.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411112

[7] M. Martinez, "Why Did Developers Migrate Android Applications from Java to Kotlin?" IEEE Transactions on Software Engineering, vol. 47, no. 6, pp. 1262-1275, 2021. ISSN: 0098-5589.

[8] P. Kong, X. Bai, K. Chen, and Z. Hu, "Automated Testing of Android Apps: A Systematic Literature Review," IEEE Transactions on Software Engineering, vol. 44, no. 2, pp. 122-146, 2018. ISSN: 0098-5589.

[9] H. van der Merwe, B. van der Merwe, and W. Visser, "Verifying Android Applications Using Java PathFinder," ACM Transactions on Software Engineering and Methodology, vol. 21, no. 4, pp. 1-27, 2012. ISSN: 1049-331X.

[10] S.Y. Fiawoo, "Design and development of an Android application to retrieve information from a remote database," International Journal of Computer Applications, vol. 60, no. 19, pp. 1-6, 2012. ISSN: 0975-8887.

