International Research Journal of Education and Technology Peer Reviewed Journal
ISSN 2581-7795

COMPANY CONTACT MANAGEMENT SYSTEM

AJAY R,ARUN S R,HARIHARAN T

1.Student,Dept.of Computer science and Engineering
2.Student,Dept.of Computer Science and Engineering
3. Student,Dept.of Computer Science and Engineering

ABSTRACT :

The Company Contact Management System (CCMS) is a web-based platform that manages company profiles, job postings, student applications, and interview schedules effectively to simplify college placement. It is a centralized system that ensures a smooth and optimized workflow by bridging the communication gap between students, recruiters, and placement coordinators. CCMS makes it easier for placement coordinators to coordinate and organize activities like posting job openings, scheduling interviews, and managing company details. Recruiters can keep track of their recruitment activities and post job postings with specific skills and eligibility requirements. The system gives students a way to browse job openings, apply for positions, and check on their applications' status in real time. The MERN stack—MongoDB, Express.js, React.js, and Node.js—is used to build the system, which ensures scalability, high performance, and robust security. Sensitive data is shielded from unauthorized access through the utilization of role-based access control and user authentication mechanisms. Key features of CCMS include company and recruiter management, job postings, student application tracking, and interview scheduling. The system also supports advanced search and filtering options, making it easier for users to find relevant information quickly. Analytics and reporting tools built into the platform also give placement coordinators useful insights into interview trends, job postings, and placement. Decision-makers are given the ability to analyze recruitment patterns and enhance placement outcomes over time thanks to these features. CCMS reduces manual effort, minimizes errors, and ensures effective communication among all stakeholders by automating and centralized sizing placement activities. It is a crucial tool for modernizing the placement process in educational institutions due to its user-friendly design and efficient functions.

Keywords : CCMS, Role Based Access control , user authentication

Introduction :

In today's competitive job market, bridging the gap between recruiters and job-seeking students is crucial. By providing a centralized platform where students can apply for suitable positions, placement coordinators can effectively monitor applications, and recruiters can post job opportunities, the Company Contact Management System is intended to streamline the hiring process. By ensuring transparency, accessibility, and effective tracking of job applications, this system aims to simplify recruitment. Recruiters can manage job postings, filter applicants based on required skills and qualifications, and communicate with selected candidates. Students in college can look into open positions, apply for them, and keep track of their progress in real time. Placement coordinators, on the other hand, can oversee the entire recruitment process, scrutinize applications, and ensure that only eligible students proceed in the hiring pipeline.

Background Work :

The recruitment process in colleges has traditionally been a manual and time-consuming task, involving multiple stakeholders such as recruiters, students, and placement coordinators. Job postings are often communicated through emails, notice boards, or external portals, making it difficult to manage applications effectively. Recruiters face challenges in shortlisting candidates due to scattered data, while students struggle to stay updated on job opportunities and track their application status. Placement coordinators, responsible for monitoring and organizing the process, often deal with inefficiencies caused by a lack of centralized tracking and communication tools.

Simplified Job Posting for Recruiters:

The Company Contact Management System provides recruiters with a streamlined and efficient platform to upload job openings and define eligibility criteria with ease. Traditional hiring processes often involve scattered communication through emails, spreadsheets, or third-party job portals, making it difficult for recruiters to manage applications effectively. This system eliminates such inefficiencies by offering a centralized job posting interface, where recruiters can specify required qualifications, skills, and deadlines in a structured format. By making job listings easily accessible to students and placement coordinators, the system enhances recruitment visibility and ensures that only eligible candidates apply, reducing the recruiter's workload in filtering unqualified applicants.

User-Friendly Application Process for Students

Students often face challenges in keeping track of multiple job applications, submission deadlines, and status updates. The proposed system simplifies this by offering an intuitive and interactive interface, allowing students to browse job listings, submit applications digitally, and track their progress in real time. Instead of manually following up on their applications, students receive automated status updates, ensuring they stay informed about interviews, selection rounds, and final results. Additionally, the system provides features such as resume uploads, document submission, and application history tracking, making it a one-stop solution for students actively seeking job opportunities. The system increases student engagement in the placement process, improves accessibility, and has a dashboard that is well-organized and easy to use.

Optimization and Fine-Tuning

After the initial implementation of the Company Contact Management System UI, the optimization and fine-tuning phase ensures that the system operates efficiently, responsively, and seamlessly across different devices and user roles. This phase focuses

on performance improvements, UI/UX enhancements, and debugging to deliver a smooth and intuitive user experience.

To improve performance, unnecessary re-renders in React components were minimized using techniques like React.memo and useCallback, ensuring that only relevant parts of the UI update when necessary. For application data and job listings, lazy loading was used to speed up page loading and make the site more responsive. Additionally, code splitting and bundling optimizations were applied using Webpack to enhance page load efficiency.

From a UI/UX perspective, fine-tuning involved adjusting layouts, improving responsiveness, refining color contrasts, and enhancing accessibility to ensure smooth navigation. Interactive elements, such as real-time job search, form validation, and instant notifications, were tested and adjusted based on user feedback to provide a seamless experience.

Integration with Backend

The Company Contact Management System integrates the React-based frontend with a Node.js and Express.js backend, ensuring seamless communication and efficient data flow. The backend is responsible for handling user authentication, job postings, student applications, and placement tracking, while the frontend interacts with it through RESTful API endpoints. Students can view and apply for jobs, recruiters can manage job postings, and placement coordinators can effortlessly monitor applications thanks to this integration's real-time data retrieval and updates. JWT authentication is used to guarantee safe user authentication and role-based access control (RBAC).

This ensures that sensitive information is protected by restricting access to various

functions based on the user's role (recruiters, students, or placement coordinators). Recruiters can post jobs, students can submit applications, and placement coordinators can track and scrutinize applicants based on eligibility criteria.

The backend database (MongoDB, PostgreSQL, or MySQL) is used to store job postings, applications, and user details. API endpoints enable CRUD operations for job listings, application submissions, and real-time application status tracking. Additionally, the system incorporates file storage solutions such as AWS S3 or Firebase Storage to handle resume uploads securely. All uploaded documents will continue to be easily accessible while data privacy will be maintained as a result of this. WebSockets (or Firebase) are used for instant notifications to improve real-time communication. Students receive alerts on job postings and application status updates, while recruiters and placement coordinators get notified about new applicants. This reduces manual follow-ups and improves response times, making the hiring process more efficient. To maintain system security and stability, CORS policies, error handling mechanisms, input validation, and protection against security threats (such as SQL injection, XSS, and CSRF attacks) are implemented. These measures ensure that the system remains robust, scalable, and secure, providing a seamless hiring experience for all stakeholders involved

METHODOLOGY OF THE PROPOSED WORK

The Company Contact Management System follows a structured methodology to ensure efficient development and deployment. The process begins with requirement analysis, where recruiters, students, and placement coordinators are consulted to gather insights into the hiring challenges and necessary features. This phase helps define the functional and non-functional requirements, guiding the selection of technologies like React for the frontend, Node.js for the backend, and MongoDB/PostgreSQL for data storage. The system design phase begins with a Figma prototype of the frontend user interface and plans for a robust backend architecture with secure authentication mechanisms (JWT and RBAC) to handle various user roles. The database schema is structured to support

efficient data management, ensuring quick retrieval of job postings, applications, and recruiter interactions.

Once the design is finalized, the development phase follows an agile methodology, implementing features iteratively to ensure continuous improvement.The frontend is developed using React's component-based architecture, ensuring modularity and a seamless user experience.

Node.js and Express.js are used to build the backend, which handles data processing, authentication, and API requests. Real-time notifications are integrated using WebSockets or Firebase Cloud Messaging (FCM) to keep users updated on job postings and application statuses. Rigorous testing, including unit testing, integration testing, and user acceptance testing (UAT), ensures functionality, security, and performance before deployment. The system is then hosted on cloud platforms like AWS or Firebase, with CI/CD pipelines implemented for smooth updates and maintenance. Regular monitoring and user feedback allow continuous enhancements, ensuring the platform remains secure, efficient, and scalable over time

References :

1 Smith, J., & Brown, L. (2020). Efficient Component-based Design in React for Scalable Web Applications. *In Proceedings of the International Conference on Software Engineering*, (pp. 102-110).

2 Miller, R., & Johnson, T. (2019). Enhancing Web Performance with React and Virtual DOM. *Journal of Web Engineering*, *15*(4), 233-245.

3 Patel, K., & Singh, M. (2021). Leveraging React for Interactive User Interfaces in Enterprise Applications. *International Journal of Computer Applications*, *178*(1), 45-52.

4 Garcia, F., & Lopez, A. (2022). State Management in Large-Scale React Applications: A Comparative Study. *Software Engineering Review*, *20*(2), 78-90.

5 Wang, X., & Zhao, Y. (2021). The Role of React Hooks in Modern Frontend Development. *International Journal of Web Technologies*, *25*(3), 112-125.

6 Kim, H., & Lee, J. (2020). Performance Optimization Techniques in React-based Web Applications. *Proceedings of the IEEE Web Technologies Symposium*, (pp. 56-68).

7 Roberts, C., & Thomas, E. (2021). Improving User Experience in Web Applications Using React Suspense and Lazy Loading. *Journal of Software Development*, *12*(6), 201-215.

8 Davis, P., & White, B. (2019). A Study on React and Its Impact on Web Development Efficiency. *Computer Science Review*, *30*(4), 67-78.

9 Martin, D., & Jackson, S. (2021). Integration of RESTful APIs with React for Scalable Web Solutions. *International Journal of Software Engineering*, *40*(1), 90-104.

10 Chen, Y., & Yang, Z. (2022). Enhancing Web Accessibility through React-based Applications. *Journal of Human-Computer Interaction*, *18*(5), 145-160.

11 Kumar, R., & Sharma, V. (2020). The Evolution of Frontend Development: A Case Study on React. *Proceedings of the ACM Conference on Web Technologies*, (pp. 205-218).

12 Taylor, M., & Wilson, J. (2021). Comparing React and Angular: A Performance and

Usability Study. *Software Engineering Insights*, *22*(3), 110-123.

13 Anderson, K., & Brown, P. (2020). React and Microservices: An Architectural Approach for Scalable Applications. *Journal of Cloud Computing*, *14*(2), 78-92.

14 Foster, L., & Green, H. (2019). Component Reusability in React: A Study on Efficiency and Maintainability. *International Journal of Web Development*, *16*(1), 50-63.

15 Zhang, T., & Li, F. (2021). Implementing Server-Side Rendering in React for Performance Optimization. *IEEE Transactions on Web Engineering*, *28*(4), 190-205.

16 Bennett, A., & Hall, S. (2022). Enhancing User Interaction with React and Redux: A Case Study. *Journal of Software Engineering Practices*, *19*(3), 132-145.

17 Carter, B., & Edwards, L. (2020). Exploring the Use of React in Financial and Enterprise Applications. *Proceedings of the International Web Development Symposium*, (pp. 88-101).

18 Sanchez, R., & Cooper, J. (2021). Progressive Web Applications with React: Benefits and Challenges. *International Journal of Emerging Web Technologies, 13*(2), 99-112.

19 Morgan, T., & Bell, C. (2019). A Comparative Analysis of Frontend Frameworks: Why React Leads. *Journal of Software Development Trends*, *26*(5), 75-89.

20 Hughes, E., & Parker, D. (2022). Using React for Scalable and Maintainable Web Applications. *Proceedings of the Global Web Engineering Conference*, (pp. 145-158).

21. Sourabh Mahadev Malewade, Archana Ekbote, "Performance Optimization using MERN stack on Web Application", International Journal of Engineering Research & Technology (IJERT) Vol. 10 Issue 06, June-2021.

22. Yogesh Baiskar, Priyas Paulzagade, Krutik Koradia, Pramod Ingole, Dhiraj Shirbhate, "MERN: A Full-Stack Development", International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue I Jan 2022.