JOBJUNCTION: BRIDGING THE GAP BETWEEN JOB SEEKERS AND EMPLOYERS

[1]Mohd Sajid, [2]Nibha Kumari, [3]Md Danish, [4]Md Alfaz Khan, [5]Dr. Pooja Kapoor and [6]Dr. Amit Agarwal
[1, 2, 3, 4]Department of (CSE), Mangalmay Institute of Engineering and Technology, Greater Noida, Uttar Pradesh, India
[5]Research Coordinator, Professor and [6]Director, Mangalmay Institute of Engineering and Technology, Greater Noida, Uttar Pradesh, India

ABSTRACT

This research paper introduces the conceptualization, design, and development of ***JobJunction****, an interactive and user-focused web-based job portal that is built with the* ***MERN stack (MongoDB, Express.js, React.js, and Node.js).*** *The main aim of this project is to develop a common platform that enables the smooth interaction between job seekers and recruiters. In a day and age where digital solutions are revolutionizing age-old recruitment methods, JobJunction is a holistic solution to simplify the hiring and job search process.*

The portal features role-based access for two key users: candidates and recruiters. Candidates are able to register, create professional profiles, upload CVs, search for matching jobs, and submit applications with simplicity. Meanwhile, recruiters are able to post jobs, search applicant profiles, and handle candidate selection through a straightforward dashboard. The system is built to ensure responsiveness, usability, and scalability so that it remains flexible to growing recruitment needs.

Security features like data validation, authentication, and secure APIs are put in place to safeguard the data of the users and maintain a stable experience. The use of contemporary web technologies not only optimizes the performance but also offers a secure backend infrastructure for dealing with simultaneous users and live updates.

Keywords: *Job Portal, MERN Stack, Full-Stack Development, Online Recruitment System, Web Application.*

CONTEXT OF THE RESEARCH

The growth of technology and the growing influence from online platforms have shaped how individuals communicate, work and when information. Over the years, the online recruitment system has emerged as a priority focus area because of their impact on corporate, educational and start-up areas. As a new strong tool, cloud -based solutions and data -handled decision makers have made digital platforms an integral part of daily life.

This emphasizes the need for strong, scalable and user-friendly solutions to the growing domains of digital employment and job portals, especially those who can appeal to both technical and non- technical users without compromising on system integrity and performance over time.

PROBLEM STATEMENT

Despite the progress of online recruitment, many job portals still lack efficiency, access and user privatization -especially for new graduates and users of Tier -2 and Tier -3 cities. Unorganized interfaces and irrelevant entries often prevent users from finding suitable jobs, while small businesses struggle to attract the right talent.

A well-organized, user-friendly platform is required that builds over this difference, making the process of searching and hiring jobs more efficient for both job seekers and recruiters.

OBJECTIVE OF THE RESEARCH PAPER

The main objectives of this research are:

1. To identify the boundaries of traditional recruitment methods and define requirements for a skilled job portal.
2. To evaluate corn stack for the manufacture of a modern, scalable and user -friendly job application platforms.
3. To activate job postings, work, start, start and recruit the management and to test.
4. To suggest future promotion as AI-based job matching and integration of mobile apps based on research results.

SCOPE OF THE STUDY

The study focuses on the design and development of an online job portal using Mong Stack (Mongodb, Express.JS, React.JS and Node.JS), which aims to recruit job seekers and primary users such as Aadhaar. The goal is to stream the recruitment process through facilities such as job planning, resume uploading, profile management and secure user authentication. The purpose of this system is to create a centralized, responsible and effective platform for both sides. Research is carried out using a full- stack web development method within a limited development timeline, using open sources educational equipment and assessing the scope of the project environment at the college level.

OVERVIEW OF EXISTING JOBPORTAL SYSTEM

Increasing dependence on digital platforms has changed traditional recruitment processes to a large extent. Various online job portals that work, and LinkedIn has played an important role in connecting job seekers with employers. These platforms provide facilities such as profile creations, resuming submission and job matching based on user input. Many academic and industrial studies have analyzed systems that focus on user experience, scalability and automation of recruitment functions.

Previous research highlights the importance of user -friendly interfaces, real -time data access and secure authentication systems in the production of effective position portals. In addition, studies have emphasized the role of complete-stack development in the creation of responsible and scalable web applications, with a popular alternative due to uniformity in Javascript unit in Marn Stack Frontend and Backend.

GAPS IDENTIFIED IN PREVIOUS RESEARCH

While existing systems are effective, there are many holes:

1. Limited customization options for small businesses and individual recruitment. Inadequate support for two-way communication between candidates and employers.
2. Lack of open sources, job portals at the academic level as modern piles are made from modern piles such as learning and innovation.
3. Limited space for integration with new technologies such as AI-based recommendations in many basic systems.

CONTRIBUTION AND DIFFERENTIATION OF THIS WORK

The aim of the Jobjunction project is to address these intervals by developing a light, full-stack job portal that is particularly suitable for academic performance and implementation of the real world. Unlike large commercial platforms, jobs are designed to be more accessible, adaptation and educational. It provides important features such as roll-based access, secure login, job settings, CV management and a clean user interface.

This research not only creates a practical solution, but also contributes to educational understanding that shows how modern equipment can be used to develop scalable and functional web applications.

TECHNOLOGICAL ADVANCEMENTS AND TRENDS IN JOB PORTALS

This section can discuss the fact that modern technologies (such as Mern Stack, Cloud Integration or AI) form the future of job portals and how they are adopted in the new system. This emphasizes how these innovations increase scalability, real -time performance and user engagement. This determines that the technology you are not only relevant, but also able to develop with the rapidly changing requirements of the recruitment industry, is also adaptable, future for the future.

RESEARCH DESIGN

This research adopts a design with mixed method, and combines both qualitative and quantitative attitudes to ensure a comprehensive understanding of the problem and the site of solution. The qualitative aspect focuses on gathering user expectations, interface preferences and speechless insights through interview and observational response. It helps shape the user interface and functionality to match the needs of the real world.

On the other hand, quantitative components include structured tests and measurement of the performance of the platform, functionality and user satisfaction using matrix. These include response time, error rate and degree of completion for main functions (eg registration, job settings and application submission).

DATA COLLECTION METHODS

To collect relevant data, the study used the following methods:

1. **User interview and response:** Understanding the general challenges and desired functions in a job portal so that job seekers and recruitment are organized.
2. **Questionnaire:** Correct, user satisfaction and interface are informally distributed to evaluate intuitivity.
3. **Overview test:** The real -time user was performed during the development phase to assess user interactions and identify targeted problems.
4. **Bencmarking:** Compared to the features of the job to identify existing platforms working and really improving areas.
5. **System's performance log:** Backnd Log and System Response Time Analysis to evaluate platform reliability under different user loads.
6. **Promotional test session:** Concentrated sessions where participants tried main tasks (eg registration, job application) to assess the efficiency of user friendship and completion of the work.

TOOLS AND TECHNOLOGIES USED

The project was built using Mern Stack, chosen for its efficiency, flexibility and scale:

1. **Mongodb -** a NOSQL database to save job listing, user profiles and applications.
2. Express.JS A minimal and flexible framework for handling logic and Serversid and API router.
3. **React.JS** A powerful front-standing library to create interactive, responsible and component- based user interfaces.
4. **Node.Jss** ET Javascript Runtime environment is used to create servers side logic and connect front and backnd. Further equipment includes:
5. **Postman -** for API testing and troubleshooting.
6. **Git & Github -** for version control and collaboration.
7. **Visual Studio Code -** as the main development environment.

A test group of 10-15 users participated in the test of the Jobjunction Platform. The group consisted of students, fresh graduates and recruitment with small businesses, representing the primary user base of the system. These participants were chosen to provide different approaches and realistic reactions based on their different needs and experiences. User interactions were carefully monitored under major activities such as registration, job postings, job search, profile update and application submissions. Their answers helped to identify both functional and non-functional problems related to purposeful, navigation and system performance.

PROCEDURE FOR ANALYSIS

The system was evaluated:

1. **Functional testing:** According to the intention to ensure all functions such as registration, login, job application and profile management.
2. **Proper tests:** Navigation, interface design and simple user satisfaction through reactions.
3. **Result tests:** Overview of side load time and backnd responsibility under several user interactions.
4. **Comparative analysis:** Measurement of how jobs vary or improve existing platforms depending on the most important right matrix and response.

RESULTS AND DISCUSSION

PRESENTATION OF RESULT

The results of the test and feedback from the user were collected and analyzed to evaluate the efficiency of the Job Junctional Portal. Big comments include:

1. **Registration and login:** 100% users were able to register and log in without meeting important problems.
2. **Position settings and applications:** More than 90% of users could easily post or search for the job.
3. **Navigation and spokesman:** 80% of users provided the platform status as user -friendly on the basis of access and clean interface design.
4. **Performance:** The average side load time was less than 2 seconds during testing with contemporary users.

INTERPRETATION OF RESULTS

The results suggest that the job will successfully meet their primary goals: to offer a smooth user experience, offer effective positions and application features and ensure a reliable backnand system. The use of the Mern Stack platform contributed to the spontaneous integration between the front and the backnd components, which enabled rapid growth, stability and continuous performance. High satisfaction score reflects the efficiency of user-centered design options, intuitive workflows and a responsible interface.

Compared to existing job portals:

1. Jobjunction provides more personal experience with an intuitive interface, making it easier for new users to navigate compared to many existing platforms.
2. Unlike large commercial job portals, the job feature is fully designed using Open-Source techniques, making it very cost-effective and scalable-specifically suitable for educational institutions and startups.
3. It builds together the communication interval by integrating real -time updates and information, a feature that often reduces the basic job tablet system.
4. The platform clearly distinguishes between recruiters and candidates through roller -based access, and improves clarity in the workflow and user -specific functionality.
5. As a project developed in an academic setting, it acts as a functional solution and performance of full-stack development both, which rarely focuses on mainstream portals.

LIMITATIONS OF THE STUDY

1. **Sample size:** The test was performed on a small group (10-15 users), who could not fully represent wider user behavior or expectations in different industries or age groups.
2. **Lack of AI integration:** Although it was originally proposed, advanced facilities such as AI-AI- based job recommendations, re-start screening or chatbot assistance were not implemented due to time and lack of resources.
3. **Limited load test:** System performance was only tested under moderate traffic conditions. This evaluation requires further testing of how the platform handles high versions of contemporary users in real world scenarios.
4. **Mobile responsibility:** While basic responsibility was assessed, the forum was mainly tested on desktop units. A fully adapted mobile version is still developing.
5. **Security measures:** Basic certification and increased handling were implemented, but advanced security practices such as encryption, two -factor authentication and data security audit were not considered deeply as deeply ascertained.

SUGGESTIONS FOR IMPROVEMENT

1. Use AI-based features for individual job recommendations, restart the screening and automated candidate-kari-matching.
2. Develop a fully responsible mobile version to ensure steady access and purposes on all devices.
3. Integrates advanced safety measures such as encryption, two -factor authentication and regular safety audit.
4. Add real -time communication equipment such as live chat or messages between recruiters and job seekers.
5. Conduct wide loads and stress tests to improve platform scaling for a high traffic environment.
6. Introduce a response and ranking system to increase transparency and user confidence.
7. Increase the Administration Table with better user management tools, analysis and system monitoring.
8. Provide multilingual assistance to access a comprehensive user base in different fields.

Problem Statement Addressed / Inspiration: The project was inspired by a user -friendly, scalable and the need to create an open-source job portal that creates a difference between traditional recruitment platforms and modern digital employment requirements. It focuses on improving user interactions, efficiency and access to both job seekers and recruitment.

Method used: The portal was developed using Mern Stack (Mongodb, Express.JS, React.JS and Node.JS) after a full-stack web development approach. Practical test methods including feedback from users, evaluation of benefits and benchmarking of systems were used throughout the development cycle.

Central findings: The platform meets both functional and non-functional needs, and offers spontaneous navigation, effective job application processes and rapid load time. Profit tests showed high responsibility, and users reported a positive experience with design and purpose. This system is particularly suitable for its simplicity and scalability for educational institutions, startups and small businesses.

LIMITS AND FUTURE WORK

While the current version provides a stable and functional basis, AI-based recommendations for future development, integration of mobile apps and advanced analysis. This improvement will improve personalization, access and general system intelligence, and ensure the continuous increase and relevance of the platform.

REFRENCES

1. MongoDB. (n.d.). *The Developer Data Platform*. Retrieved from https://www.mongodb.com
2. Express.js. (n.d.). *Fast, unopinionated, minimalist web framework for Node.js*. Retrieved from https://expressjs.com
3. React.js. (n.d.). *A JavaScript library for building user interfaces*. Retrieved from https://reactjs.org
4. Node.js. (n.d.). *JavaScript runtime built on Chrome's V8 JavaScript engine*. Retrieved from https://nodejs.org
5. Postman. (n.d.). *API Platform for Building and Using APIs*. Retrieved from https://www.postman.com
6. GitHub. (n.d.). *GitHub: Where the world builds software*. Retrieved from https://github.com
7. Visual Studio Code. (n.d.). *Code Editing. Redefined.* Retrieved from https://code.visualstudio.com
8. W3Schools. (n.d.). *Web Development Tutorials*. Retrieved from https://www.w3schools.com
9. Mozilla Developer Network (MDN). (n.d.). *Web Docs*. Retrieved from https://developer.mozilla.org
10. Stack Overflow. (n.d.). *Where Developers Learn, Share, & Build Careers*. Retrieved from https://stackoverflow.com
11. Zhang, X., Zhang, X., & Han, L. (2019). An energy efficient Internet of Things network using restart artificial bee colony and wireless power transfer. *IEEE Access*, 7, 12686–12695. Retrieved from https://doi.org/10.1109/ACCESS.2019.2892798
12. Zhong, X., Zhang, L., & Wei, Y. (2019). Dynamic load-balancing vertical control for a large-scale software-defined Internet of Things. *IEEE Access*, 7, 140769–140780. Retrieved from https://doi.org/10.1109/ACCESS.2019.2943173.
13. Kaur chitranjanjit, kapoor pooja, kaur Gurjeet(2023), “image recognition(soil feature extraction)using Metaheuristic technique and artificial neural network to find optimal output.Eur. Chem. Bull.2023(special issue 6).
14. Maheshwari Chanana shalu, Kapoor pooja,kaur chitranjanjit(2023),”Data mining techniques adopted by google: A study.: Empirical Economics Letters,22(special issue 2).
15. Malik, M., Dutta, M., & Granjal, J. (2019). A survey of key bootstrapping protocols based on public key cryptography in the Internet of Things. *IEEE Access*, 7, 27443–27464. Retrieved from https://doi.org/10.1109/ACCESS.2019.2900957
16. Netlify. (n.d.). *Deploy modern web projects in seconds*. Retrieved from https://www.netlify.com