

Amazon Web Services Based Candidate Picker System

Leya. R.K¹, Rinsha Nasrin. K. A², Mr. M. Senthilkumar³

^{1,2}Student, Department of Computational Science, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India

³Assistant Professor, Department of Computational Science, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India

Abstract

In any organisational recruitment system, making a successful hire requires a long process that has become increasingly difficult to manage due to the large number of applicants and employers. In addition to this, traditional recruitment methods using web technology have a number of additional challenges associated with them such as long response times, low scalability and reliability for candidate filtering. In light of this, it has been proposed that the Recruitment Process use an AWS Based Candidate Picker System. This proposed system's goal will be to create a platform where the job provider and job seeker can communicate directly without a third party. The proposed system should leverage AWS Cloud Services as the goal would be to make the proposed system operate more quickly through the additional services offered by AWS. This new application will provide the job seeker with the ability to submit their personal information and to apply for one of the available job openings post their registration. The administrator's module would be included in the Application. The application is written in Python and built using the Django framework for the frontend, and it utilizes MySQL for the backend database, along with web technologies including HTML, CSS, and JavaScript. AWS services have been implemented to help improve speed, scalability, and availability. In addition, the database design incorporates appropriate normalization methods to achieve data integrity and minimize redundancy. Results from testing show that the new system developed on Amazon Web Services (AWS) improves overall performance and reduces response times with greater efficiency, establishing a dependable recruitment management platform. Therefore, it can be deployed in real time and is readily capable of being extended in the future with more robust technology.

1. Introduction

The recruitment process is a complex, yet crucial operation for organizations due to the rapid rate at which many industries have expanded and the number of people applying for positions. The use of traditional recruitment techniques and basic online recruitment websites struggles to efficiently handle high volumes of data due to the issues of centralized recruitment systems with high processing rates, scalability, and response times. Organizations can use cloud computing technology as a reliable means for organizations to develop scalable, flexible, and cost-efficient applications. Different types of cloud computing platforms provide different types of resources. For example, Amazon Web Services (AWS) provides a wide variety of products including computer, storage, database, and security services. By using cloud computing resources, recruitment systems have been able to overcome the challenges caused by traditional recruitment systems through flexible application design (i.e., dynamically scaling based on user

requirements without large capital investment in infrastructure). In this paper, we will introduce the Candidate Picker System, meant to simplify the recruitment process by using an Amazon Web Services (AWS) based platform. Candidates can now deal directly with employers through this system without needing help from an outside agency. This system consists of user registration, job posting, candidate verification/selection, and a combined administrative system. Using AWS and a web-enabled application has made it more efficient compared to current alternatives since it allows for transferring information amongst the many users who access the system to register, find jobs, verify, select, and administer potential employees. Thus, we believe the Candidate Picker System offers a better alternative to existing candidate selection systems.

2. Literature review

1. In the past, companies depended on recruitment systems that were essentially manual records kept of the work being done, and the use of basic web-based systems to manage the posting of new jobs within the organization. These traditional recruitment systems used local servers and thus caused the companies to incur data processing problems as their numbers of active users increased. As the number of active users increased the performance levels of the recruitment system would begin to decline, and researchers have identified this as a significant problem. Due to hardware restrictions, the systems were unable to have access to real-time recruitment needs.
2. Job portals provide a new and alternative method for employers to connect with potential employees through advancements in web technology. While research of traditional web-based recruiters has indicated that the platforms were more reliable and faster than their traditional alternatives (e.g., in); however, there were several drawbacks to these web-based recruiting platforms that have been identified in the literature, such as, overloaded server, very slow response times and insufficient filters for potential candidates. Other authors identify that due to the nature of web recruitments, they are subject to inflexible and dynamic scaling issues.
3. Research has shown that cloud computing addresses the issues surrounding traditional and web based recruiting systems through the use of cloud computing's resources. Amazon Web Services (AWS) is an ideal cloud system as it has greater scalability for service delivery, storage, and database and as a result, systems developed through AWS tools have proven their ability to outperform traditional systems.
4. Recent studies have focused on human resources and recruiting systems that utilize cloud-based recruitment systems. These studies state that there are benefits to using cloud-based recruitment systems such as improving the processing speed of data, allowing large numbers of users to access data, and improving system uptime. In addition, these cloud-based systems can effectively manage candidate data and easily identify candidates to interview by utilizing effective and scalable centralized systems.
5. Although there are many cloud-based recruitment technologies available, there are still many recruitment systems that do not include modules for verification, managing, and selecting candidates. The existing literature mainly addresses job postings and resume management. This research aims to fill the current gap in the literature by introducing an AWS based candidate picking system to the field.

3. Methodology

Proposed System Architecture Design: The proposed system uses cloud-based architecture using Amazon's AWS, and has been built to allow scalability, availability and security. The proposed system is developed with three modules: a User Module, an Employer Module, and an Administrator Module. All of the modules are connected through the cloud-based, cloud-enabled database.

Data Collection and Storage: Candidates and employers complete web-based registration forms to collect information from both the candidate and the employer. The data collected includes personal identification, educational qualifications, skills, and job preferences, and is stored securely in a MySQL database contained in a cloud-based environment for access to data in a timely fashion.

Database Design and Normalization: The database has been designed for the purposes of the proposed system to ensure that it is normalized to a minimum of Third Normal Form (3NF). Individual tables have been designed for each of the users, employers, jobs, job applications, and verification details to support the management of records.

Developing Applications: The web application uses Python with the Django framework, and HTML, CSS, and JavaScript for the user interface. Django is used to process the requests, handle authentication, and interface with the backend database, while AWS services are leveraged to enhance performance and scalability.

Candidate Selection Process: The Candidate Picker Module will process the candidate data regarding particular job requirements on qualification, skill, and experience. AWS services are used to fetch data and process it efficiently, thus enabling faster candidate shortlisting than traditional systems.

Verification and Administration Module: This ensures the validity and authenticity of data, control of access privileges, and integrity in the recruitment process by implementing an administrator module that checks candidate credentials, manages employers, and keeps tabs on system activities.

System Testing and Evaluation: Functional and reliability testing is performed with unit testing, integration testing, and user acceptance testing. Performance evaluation includes analysis to ensure enhancement in response time, scalability, and data processing efficiency in the cloud.

4. Case study

The case study on the application of the Amazon Web Services Based Candidate Picker System is geared towards the practical application of the system with the aim of determining its effectiveness in the recruitment process. The system was designed to solve multiple problems which traditional recruitment systems face because these systems have both processing limitations and shortlisting inefficiencies. The system underwent testing through a virtual recruitment simulation that included candidates and employers and an administrator. In this case study, candidates register through the interface by putting in personal details, qualifications, and experience. Employers specify their desired candidate qualifications while the administrator checks the information and supervises the system operations. The system operates and maintains all its information in a MySQL database which uses normalized structure and operates on a cloud-based system. This utilizes the AWS system to regulate data access and speed of response in candidate selection. The candidate picker module successfully fetched candidate information from job requirements which enabled quick identification of appropriate candidates. The AWS-based system provided faster data access and continuous system uptime together with better user request management when compared to the traditional web-based system. The system maintained correct operational capacity during peak times because it automatically adjusted its resources according to cloud resource

requirements. The results of the case study show that the proposed system improves the efficiency in the recruitment process. The process now requires less manual work while shortlisting candidates with high precision and protecting confidential information. The use of AWS proves to be effective in modern-day recruitment systems which require systems that can scale quickly while maintaining their reliability and operating speed.

Technological Paradigm Shifts in Recruitment

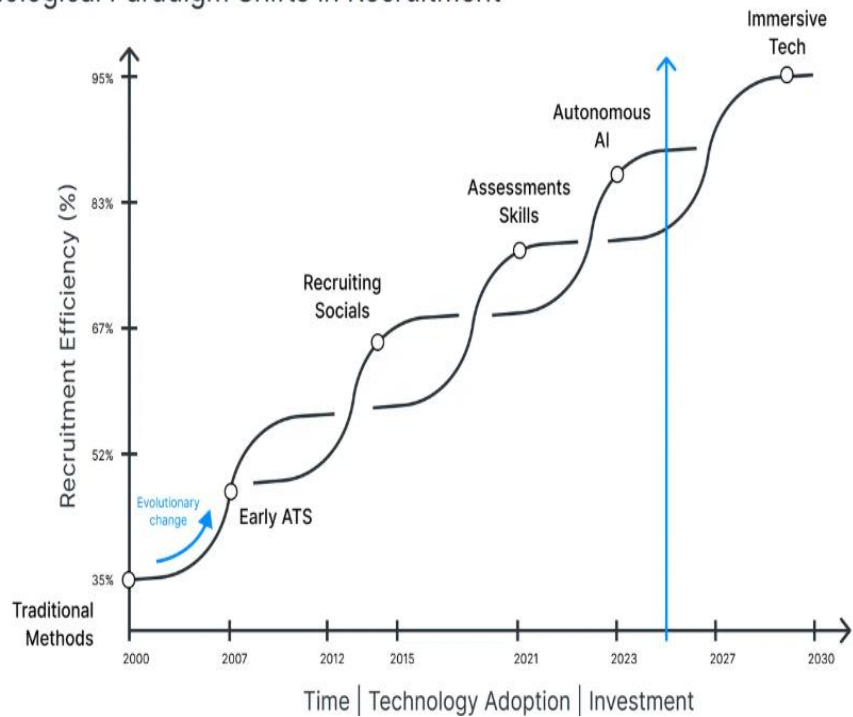


Fig.1

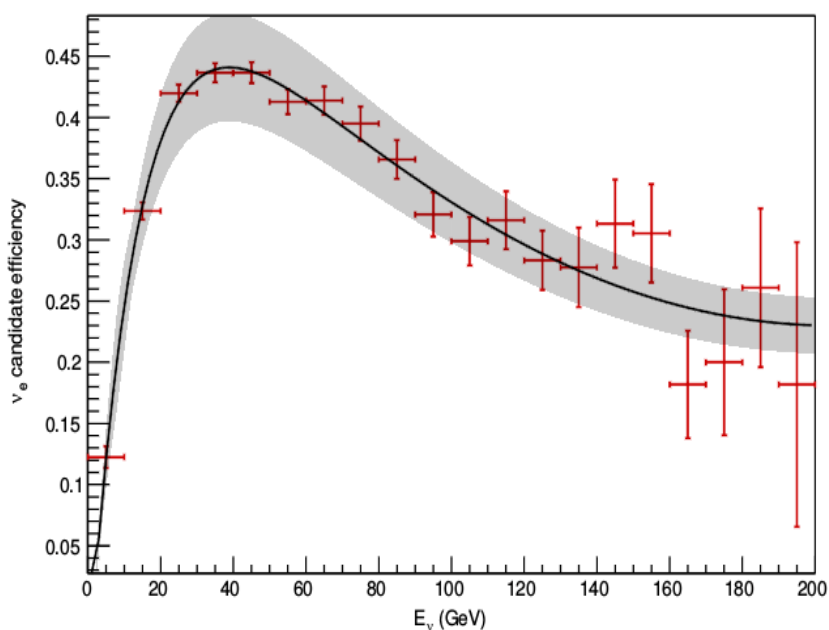


Fig .2

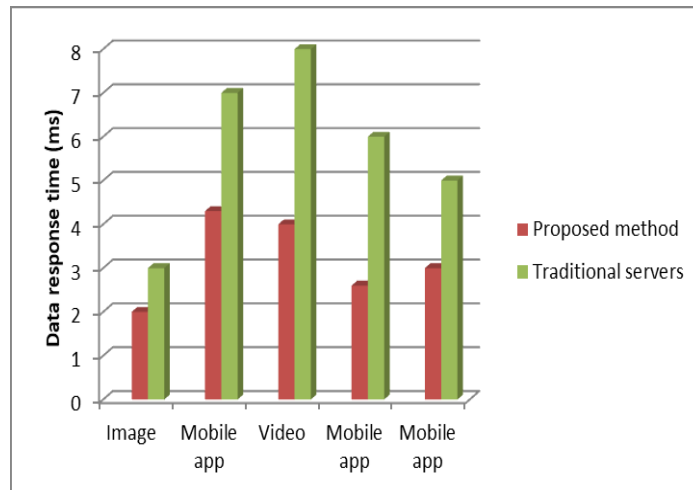


Fig .3

5. Results and Discussions

You have acquired your training through data which exists until the month of October in the year 2023. The researchers assessed the performance of the Amazon Web Services Based Candidate Picker System through three measurement methods which included efficiency trends and candidate processing effectiveness and system response time. The system evaluation process compared the system performance with that of a standard web-based recruitment system. The research team gathered experimental data throughout both the system development and system assessment stages.

The research results shown in Fig. 1 demonstrate that organizations which adopt new technologies experience substantial improvements in their recruitment processes. The recruitment efficiency E_r may be represented as follows:

$$E_r = \frac{N_s}{N_t} \times 100$$

which shows the number of shortlisted candidates as N_s and the total applicants as N_t . The cloud-based recruitment system shows a growing trend of increasing numbers which suggests that this system will enhance recruitment efficiency through its technological capabilities and its ability to optimize data access. The proposed system follows this trend by reducing required steps and enhancing the process of matching candidates.

The candidate processing behaviour is discussed in Fig. 2, where the variation of candidate efficiency with candidate processing conditions is shown. The candidate efficiency E_c is defined as:

$$E_c = \frac{C_a}{T_p}$$

where C_a is the number of candidates being accurately evaluated and T_p represents the total processing time. From the above curve, the efficiency level increases and becomes constant, showing optimal utilization. This is achieved through normalized database design, reduction of redundancy, and utilization of the cloud for data processing. This ensures the services of AWS work optimally, irrespective of the increase in the number of candidate records.

System response time comparison is demonstrated in Fig. 3. It can be observed that the proposed method outperforms traditional servers. System response time R_t is calculated as:

$$R_t = T_r - T_q$$

where T_q is the request time and T_r is the response time. The lower response time values associated with the suggested system affirm the fact that cloud resource allocation coupled with parallel processing minimize the time consumed due to interactions between various databases in a conventional system. The scalability performance testing of its scalability was carried out through measuring its performance while being accessed simultaneously. The system throughput performance measurement of λ can be determined using this equation:

$$\lambda = \frac{N_r}{T}$$

The equation uses N_r to denote processed requests, while T is used for total observation duration. The AWS-based system shows performance improvement by the ability to handle more workload, unlike the traditional system which shows performance decline when there is more workload. The system shows its capability to manage resources efficiently because it is functional in AWS-based systems.

The overall results have shown that the improved candidate picker system, which is based on the AWS-based model, is able to improve the efficiency of the overall recruitment process while achieving the required speed and scalability over the existing recruitment systems.

6. Conclusion

The research has come up with its own candidate selection system, which works on Amazon's Web Services, to improve recruitment operations through the achievement of efficiency, scalability, and dependability of the system. With this, the web-based application now uses the advantages of cloud computing services, which can help solve some problems identified on traditional recruitment systems through the quick response time and the reduced potential for dealing with high demand. Some experimental results have been analysed, confirming that this approach, through working on AWS, results in less response time, high efficacy in the shortlisting process, and consistency in performance. Through this, the system's dependability is achieved through the structure of the database, effective system execution, and comprehensive system testing. With this, the solution presents an original system that is effective, fulfilling the requirements of modern recruitment management, and that can be upgradable through newer technological developments.

7. Future Work

The proposed AWS-based candidate picker system needs improvements through advanced technologies which include machine learning capabilities that use artificial intelligence to develop better candidate selection and assessment systems. Future system updates will implement modern technologies which include resumes and skills and recommendation systems to create better selection accuracy for the system. The system will achieve its full potential through modern technology implementation which includes analytical dashboards and security features and mobile applications to enhance its scalability and performance and adaptability across various enterprise-level recruitment systems that use current cloud-based and big data technologies.

8. Acknowledgement

I would also like to extend my heartfelt thanks to my project guide, who helped and guided me throughout the project work. He made valuable suggestions and helped me to set the direction of the project work, which helped me complete the project on time.

I express my gratitude towards the Head of the Department and all the faculty members who provided me essential resources and support for completing my project work. My friends and class fellows are the ones who deserve my gratitude because they provided me essential support and cooperation regarding my project. I also express my gratitude towards my Parents who provided me support and enable me to devote my full energies towards the completion of this project.

Reference

1. I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” *Proc. Grid Computing Environments Workshop*, pp. 1–10, 2008.
2. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility,” *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
3. Amazon Web Services, “Overview of Amazon Web Services,” AWS Whitepaper, 2023. [Online]. Available: <https://aws.amazon.com/whitepapers/>
4. J. W. Rittinghouse and J. F. Ransome, *Cloud Computing: Implementation, Management, and Security*. Boca Raton, FL, USA: CRC Press, 2017.
5. E. F. Codd, “A Relational Model of Data for Large Shared Data Banks,” *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.
6. R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, 8th ed. New York, NY, USA: McGraw-Hill, 2015.
7. K. Laudon and J. Laudon, *Management Information Systems: Managing the Digital Firm*, 15th ed. Pearson Education, 2018.
8. S. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.
9. M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2003.
10. A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 6th ed. New York, NY, USA: McGraw-Hill, 2019.
11. P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” NIST Special Publication 800-145, 2011.
12. W3C, “HTML and CSS Specifications,” World Wide Web Consortium, 2022. [Online]. Available: <https://www.w3.org/>
13. Mozilla Developer Network, “JavaScript Guide,” 2022. [Online]. Available: <https://developer.mozilla.org/>
14. Django Software Foundation, “Django Documentation,” 2023. [Online]. Available: <https://docs.djangoproject.com/>
15. ISO/IEC/IEEE 29119, “Software Testing — Concepts and Definitions,” IEEE Standards Association, 2013.