



CENTRALIZED RESUME ORGANIZER

*A Phase 1 Report on Developing a Centralized Resume Organizer Web Platform using
MERN stack*

¹Mr.Sooraj, ²Dr. Shashidhar Kini K

¹Student, ²Professor & Head

¹Department of Master of Computer Applications,

¹Srinivas Institute of Technology, Valachil Mangaluru, Karnataka, India

Abstract : This project introduces the Centralized Resume Organizer, a web-based platform developed using the MERN stack(MongoDB, Express.js, React.js, Node.js). The system enables users to securely store, manage, and share resumes, cover letters, and portfolios. Key features include password-protected links, real-time resume analytics, expiration settings, and AI-powered feedback via the Gemini API. This report outlines the planning, development tools, and architecture adopted during Phase 1. The platform ensures an efficient and secure digital experience for job seekers and professionals looking to centralize their career documents and improve their online professional branding.

IndexTerms -- Resume Management, MERN Stack, Portfolio Sharing, Secure Document Links, Resume Analytics, Career Tools, Web-based Resume System.

I. INTRODUCTION

In today's dynamic and highly competitive job market, the ability to efficiently manage multiple versions of resumes, cover letters, and professional portfolios has become a crucial aspect of a successful job search. Job seekers often encounter numerous challenges, including maintaining version control across different applications, ensuring consistent and professional formatting, securely sharing sensitive documents, and gaining visibility into how their resumes are being received or viewed by potential employers.

These obstacles can significantly hinder the job application process, leading to missed opportunities and added stress for candidates. To address these common yet critical pain points, the Centralized Resume Organizer has been developed as an innovative digital solution that streamlines the resume management process. This platform allows users to store, organize, and access various versions of their resumes and supporting documents in one centralized location. Built using the MERN stack(MongoDB, Express.js, React.js, and Node.js), the system leverages modern web technologies to deliver a fast, secure, and scalable experience.

One of the standout features of the platform is its ability to generate customizable resume links, which can be tailored for different job roles or industries. These links support advanced sharing capabilities, including password protection, link expiration, and access tracking, ensuring both security and control over sensitive career documents.

The scope of this project is centered on resume and portfolio management for professionals. Built using the MERN stack and integrated with the Gemini API, the system enables secure document storage, customizable sharing, and real-time analytics. This report outlines the Phase 1 development including system design, feature implementation, data integration, and expected outcomes.

II. EASE OF USE

The platform is designed with a user-centric approach, focusing on providing an intuitive and seamless experience for all user categories. It functions both as a standalone web application and as an API-driven system, ensuring flexibility and scalability for future integration with external services such as job boards, applicant tracking systems (ATS), professional networking.

This dual functionality not only enhances usability but also opens pathways for enterprise-level adoption and third-party collaborations. The platform caters to three primary user groups:

Job Seekers: Individuals seeking employment can upload, organize, and manage multiple versions of their resumes and supporting documents. Through the use of secure, customizable sharing links, users can control access to their materials and track engagement metrics.

Recruiters: Hiring professionals can access resumes via branded, secure links, which provide a streamlined viewing experience. They also have the ability to leave feedback or comments, promoting a more collaborative and transparent hiring process.

Professionals: Users looking to maintain an ongoing professional presence can create and manage live portfolios, consolidating resumes, cover letters, project documents, and certifications in one centralized platform.

The frontend of the system, developed using React.js, ensures dynamic, real-time updates and a responsive interface that adapts smoothly across desktop and mobile environments. The backend is built using Node.js and Express.js, responsible for handling core functionalities such as document uploads, resume analytics, and secure user session management. Data persistence is achieved through MongoDB, which stores user profiles and document metadata in a flexible JSON-based schema, allowing for easy scalability and adaptability to evolving data models.

1. Prepare Your Paper before Styling

Before finalizing the structure and formatting of this paper, attention was given to ensuring clarity, completeness, and technical accuracy in the content. The development process began with gathering and simulating user-related data, such as resume uploads, user interactions, viewing activity, and feedback submissions. Sample datasets included resume metadata (titles, timestamps, access permissions) and mock analytics logs to reflect real-world usage scenarios among job seekers and recruiters.

The preprocessing phase involved organizing and standardizing this data. This included cleaning inconsistent file names, removing incomplete records, and labeling variables such as document types, access levels, and feedback scores. Numerical data like view count and click-through rates were normalized to support consistent display and analytics across the platform. These steps were essential for enabling features like secure sharing, performance tracking, and resume rating.

The application was developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). React.js was used to build a responsive user interface, allowing real-time interactions and analytics updates. MongoDB offered a flexible schema to manage dynamic user data, resumes, and feedback. Node.js and Express.js powered the backend, enabling efficient API communication, secure uploads, and integration with external services like the Gemini API for resume analysis.

Only after verifying the platform's core features through simulated testing was the paper formally organized. The final documentation was structured into standard sections: Introduction, Review of Literature, Methodology, System Architecture, Results, and Conclusion. Supporting elements such as diagrams, tables, and references were incorporated at this stage to ensure coherence, accuracy, and readability.

2. Abbreviations and Acronyms

In this paper, the following abbreviations and acronyms are used:

- MERN – MongoDB, Express.js, React.js, Node.js
- UI – User Interface
- UX – User Experience
- API – Application Programming Interface
- GPS – Global Positioning System
- ODM – Object Data Modeling
- IDE – Integrated Development Environment
- DOM – Document Object Model
- CRUD – Create, Read, Update, Delete
- JSON – JavaScript Object Notation
- CI/CD – Continuous Integration / Continuous Deployment
- HTTP – Hypertext Transfer Protocol
- DBMS – Database Management System
- ETA – Estimated Time of Arrival
- JWT – JSON Web Token
- CSS – Cascading Style Sheets
- JS – JavaScript
- DB – Database

III. RESEARCH METHODOLOGY

This section outlines the methodology adopted to develop a of a web-based application titled Centralized Resume Organizer, which aims to streamline the process of resume management, document sharing, and performance tracking for jobseekers, professionals, and recruiters.

3.1 Population and Sample

The platform targets a diverse user base, including professionals, students, freelancers, and recruiters, each with distinct document management and sharing needs. Sample resumes, cover letters, and portfolios were collected from publicly available online templates and mock users to simulate real-world usage scenarios and test features like document sharing, link-based access control, performance analytics, and centralized document organization. The sample group consisted of 20 participants across various domains, including IT, education, design, and HR, to ensure broad applicability. Participants were asked to perform tasks such as uploading resumes, generating secure links, and reviewing feedback to evaluate the platform's usability. Feedback from these users was instrumental in identifying usability gaps, refining the user interface, and improving system responsiveness. The simulated testing environment also included different devices and browsers to validate cross-platform compatibility.

3.2 Data and Sources of Data

This study utilizes a combination of simulated primary data and secondary data obtained from publicly available resume templates, portfolio samples, and feedback frameworks. These datasets were used to model realistic user interactions and validate the system's core functionalities such as document upload, secure sharing, analytics tracking, and feedback collection. Key data points incorporated into the system include:

- Resume Metadata (e.g., file name, upload date, document type, version number)
- User Information (e.g., name, email, role – student, professional, recruiter)
- Document Access Logs (e.g., number of views, timestamps, IP addresses)
- Feedback Entries (e.g., reviewer comments, ratings, timestamps)
- Sharing Settings (e.g., password protection, link expiration, viewer restrictions)
- Engagement Metrics (e.g., time spent viewing, click-through rates)

Simulated datasets were used to test system responsiveness and ensure consistent feature functionality across varied user roles, document types, and sharing scenarios.

3.3 Theoretical framework

The objective of this system is to streamline and enhance the resume management process by allowing users to securely upload, organize, and share professional documents while gaining insights through real-time analytics. Although the current implementation does not integrate advanced AI models, the system is built using a modular and extensible architecture that allows for future enhancements. Potential future integrations include AI-based resume scoring, keyword optimization, recruiter behavior analysis, and personalized content suggestions based on job role trends. Key variables used in the system include:

- User Role (e.g., Job Seeker, Recruiter, Professional) (categorical)
- Document Type (e.g., Resume, Cover Letter, Portfolio) (categorical)
- Upload Timestamp and File Size (numerical)
- Viewer Engagement Metrics (e.g., views, duration, click-throughs) (numerical)
- Access Controls (e.g., password-protected, expiration-enabled) (categorical)
- User Actions (e.g., share, edit, delete, request feedback) (categorical)

3.4 Statistical tools and econometric models

This section outlines the technological ecosystem, software tools, and development methodologies employed in building the Centralized Resume Organizer, a web-based platform focused on secure document management, intelligent sharing, and performance tracking for career documents.

3.4.1 Descriptive Statistics

Preliminary data analysis ensured the accuracy and consistency of document metadata and user interactions. Categorical fields were standardized for reliable filtering, while numerical metrics like views and file size were validated to reflect realistic usage patterns.

3.4.2 System Architecture and Technology Stack

The system is developed using the MERN stack—MongoDB, Express.js, React.js, and Node.js—for full-stack web development:

- Frontend (React.js): Creates a dynamic, responsive interface for users to upload, view, and manage resumes, while customizing sharing and tracking features.
- Backend (Node.js & Express.js): Handles API routes, user authentication, document uploads, and real-time analytics, along with managing document sharing and feedback.
- Database (MongoDB): Stores user profiles, resume data, sharing settings, and analytics in a flexible JSON format, enabling efficient querying and scalability.

3.4.3 Feature Implementation and Evaluation

The following key features were implemented and manually tested to ensure reliability and performance:

- Secure Upload & Sharing: Allows users to upload documents and generate password-protected, time-limited sharing links.
- Analytics Dashboard: Displays views, access times, and engagement metrics for each document.
- Feedback Mechanism: Enables structured comments from recruiters or peers.
- User Role Management: Provides tailored access and features for job seekers, recruiters, and professionals.

Performance and usability were evaluated based on:

- Analytics accuracy (correct tracking of document views and interactions)
- Filter efficiency (accurate document searches and sorting)
- System responsiveness (quick updates for document sharing and feedback)

IV. RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Table 4.1 Descriptive Statics

User ID	ResumeID	Title	Timestamp	Views	Action Taken
101	R101	Fullstack_2024	2025-04-28 09:22	33	Shared via Custom Link
102	R102	Designer_Portfolio	2025-04-14 09:10	25	Downloaded PDF

Table 4.1 displays simulated user interactions, tracking actions like viewing, downloading, sharing, and feedback. This helps users understand resume engagement and popularity.

V. ACKNOWLEDGMENT

The author would like to express sincere gratitude to the Project Guide and Head of the Department of MCA, Dr. Shashidhar Kini K, for his invaluable guidance, continuous support, and encouragement throughout the development of this project on the Centralized Resume Organizer. Special thanks are extended to the Principal, Dr. Shrinivasa Mayya D, for providing institutional support and creating an environment conducive to the successful completion of this project. The author also appreciates the management of Srinivas Institute of Technology for their assistance, both directly and indirectly. Heartfelt thanks go to all faculty members and non-teaching staff of the MCA department for their consistent help and technical insights. Finally, the author is deeply grateful to their parents and friends for their unwavering support, patience, and belief during this academic endeavor.

REFERENCES

- [1] Kumar, R., & Sharma, N. (2021). Digital Resume Management Systems Using Cloud-Based Architectures. *International Journal of Computer Applications*, 178(30), 15–22. <https://doi.org/10.5120/ijca2021921331>
- [2] Patel, A., & Das, S. (2020). MERN Stack for Scalable Web Application Development. *Journal of Software Engineering and Applications*, 13(7), 325–334.
- [3] Mehra, V., & Reddy, K. (2022). Role-Based Access and Security in Document Sharing Platforms. *IEEE Conference on Secure Computing*, 98–104. <https://doi.org/10.1109/SecureComp.2022.9832765>
- [4] Thomas, L., & Banerjee, P. (2019). User Engagement Analytics in Online Portfolios and Resume Platforms. *Journal of Web-Based Applications*, 7(4), 215–222.
- [5] Agarwal, S., & Prasad, R. (2021). Frontend Optimization Techniques with React.js in Dynamic User Interfaces. *International Conference on Modern Web Development*, 67–73.
- [6] Singh, D., & Kapoor, A. (2020). MongoDB for Flexible Schema Management in Resume Databases. *International Journal of Data Management Systems*, 5(2), 89–95.
- [7] Roy, M., & Bansal, T. (2022). Real-Time Feedback Systems for Career Development Platforms. *Journal of Career Technology and Innovation*, 10(3), 112–118.
- [8] Joshi, R., & Menon, P. (2023). Future Trends in Resume Screening Using AI and Machine Learning. *Journal of Artificial Intelligence in Recruitment*, 8(1), 45–52. <https://doi.org/10.1016/j.jair.2023.101234>