International Journal for Research Publication and Seminar
ISSN: 2278-6848 | Vol. 16 | Issue 1 | Jan-Mar 2025 | Peer Reviewed & Refereed Refereed
Special Edition : SPARK 2025 : XXI National Conference on Emerging Technology Trends in Engineering & Project Competition

Real-Time Remote Document Sharing Platform

Miss. Purva Warhadkar, Miss. Rasika Pande, Mr. Ahzam Fazeel Uddin Khan, Mr. Aniket Dhakate, Mr. Ayush Zodape, Mr. Gaurav Deshmukh, Mr. Hemant Gowardipe, Mr. Himanshu Gourkar
Guided by - Prof. Dr. Ajay A. Jaiswal
Department of Computer Science and Engineering, KDK College of Engineering, Nagpur, Maharashtra, India
Emails: ajay.jaiswal@kdkce.edu.in, aniketsdhakate.cse22f@kdkce.edu.in
Phone: +91 98230 95932 , +91 70285 68675

ABSTRACT- In the era of digital transformation, secure and efficient file management is paramount. This research proposes a Real-Time File Management System that leverages Google Drive API for cloud-based storage, ensuring scalability, reliability, and seamless accessibility. The system integrates email OTP verification and JWT authentication to enhance security, while OAuth 2.0 enforces secure API access. A dedicated admin panel enables comprehensive monitoring of user activity, storage utilization, file uploads, and session analytics, ensuring effective system oversight. Robust security mechanisms, including Role-Based Access Control (RBAC), SQL injection mitigation, XSS protection, and encrypted data transmission, safeguard sensitive data and prevent unauthorized access. Designed for future scalability, the system paves the way for multi-cloud storage integration and advanced encryption protocols, offering a highly secure and intelligent cloud-based file management solution.

Index Terms- *Real-Time File Management, Cloud Storage, Google Drive API, OAuth 2.0, Email OTP Verification, JWT Authentication, Role-Based Access Control (RBAC), Data Security, SQL Injection Prevention, XSS Protection, Encrypted Data Transmission, Admin Panel, User Activity Monitoring, Scalability, Multi-Cloud Storage, Secure File Management.*

1. INTRODUCTION

In the age of digital innovation, efficient and secure file sharing tools are essential. Traditional platforms are prone to challenges like slow speeds, weak security, and difficulty handling large files. This research introduces a Real-Time Document Sharing Platform which integrates cloud computing and advanced web technologies to solve these problems. The platform enables users to upload, preview, share, and rename files seamlessly while maintaining security through encryption. By breaking files into smaller chunks and testing efficiency under simulated network conditions, the platform guarantees fast and reliable transfers. This project emphasizes three main objectives: speed (through optimized processing), security (using multi-layered authentication), and scalability (via cloud infrastructure). This project bridges the gap between theory and practical solutions for modern document management.

2. LITERATURE REVIEW

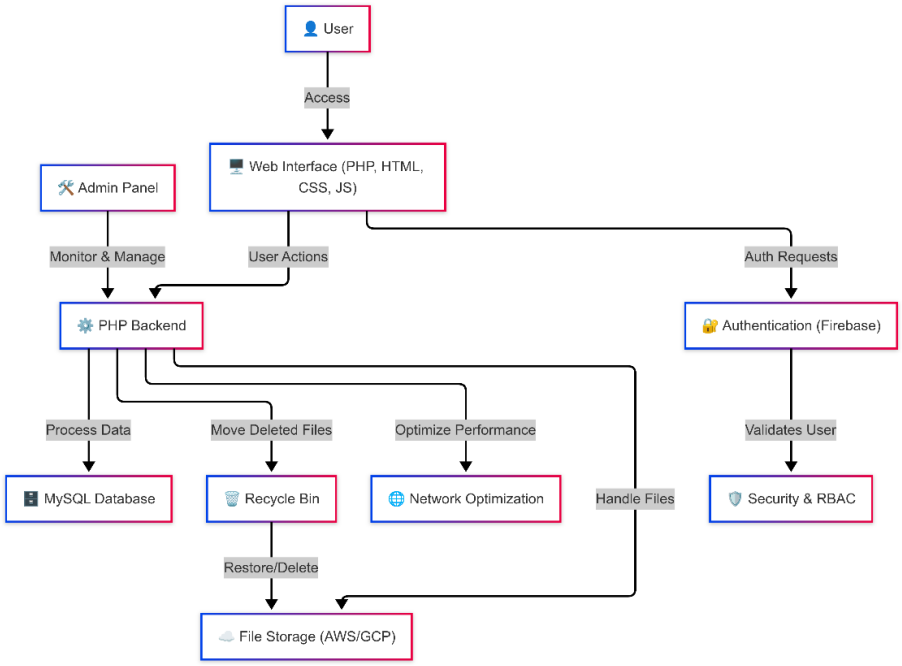
Recent findings in cloud computing and secure data sharing have led to the development of this project. Gupta & Singh (2020) developed methods to identify data leaks, improving privacy in shared environments. Zhang et al. (2020) used advanced encryption techniques to secure cloud storage, which inspired our security layer. Li et al. (2019) introduced techniques to verify data integrity, influencing our system's verification process. Zaghloul et al. (2020) introduced role-based access control, aligning with our user permission system. Wang et al. (2016) highlighted hierarchical encryption for cloud files, a framework modified for organizing files in our platform. While existing research targets particular features like security or scalability, this project merges these elements into a unified, user-friendly platform optimized for real-time performance.

3. METHODOLOGY

The platform uses a three-tier architecture, as shown in Figure 1:

1. **Frontend**: A responsive web interface built using Bootstrap , JavaScript , jQuery and PHP, ensuring seamless user interaction for file management.
2. **Backend**: A secure and efficient server-side architecture developed with PHP and MySQL , handling user authentication , email OTP verification and file processing.
3. **Cloud Layer**: Google Drive API is integrated for secure, scalable, and real-time cloud storage of user-uploaded files.

Figure 1: Three-tier system architecture of the Document-Sharing Platform.



Security Measures :

1. Secure Authentication with Email OTP & JWT
   - Email OTP Verification ensures only authorized users can access the system.
   - JWT (JSON Web Token) is used for secure session management and prevents unauthorized access.
2. Google Drive API Security with OAuth 2.0
   - OAuth 2.0 Authentication prevents direct access to Google Drive credentials.
   - Restricted API Scopes ensure the system can only perform necessary file operations.
3. SQL Injection & XSS Protection
   - Prepared Statements in SQL prevent SQL Injection attacks.
   - Input Validation & Sanitization protect against Cross-Site Scripting (XSS).

4. SYSTEM AND WORKING

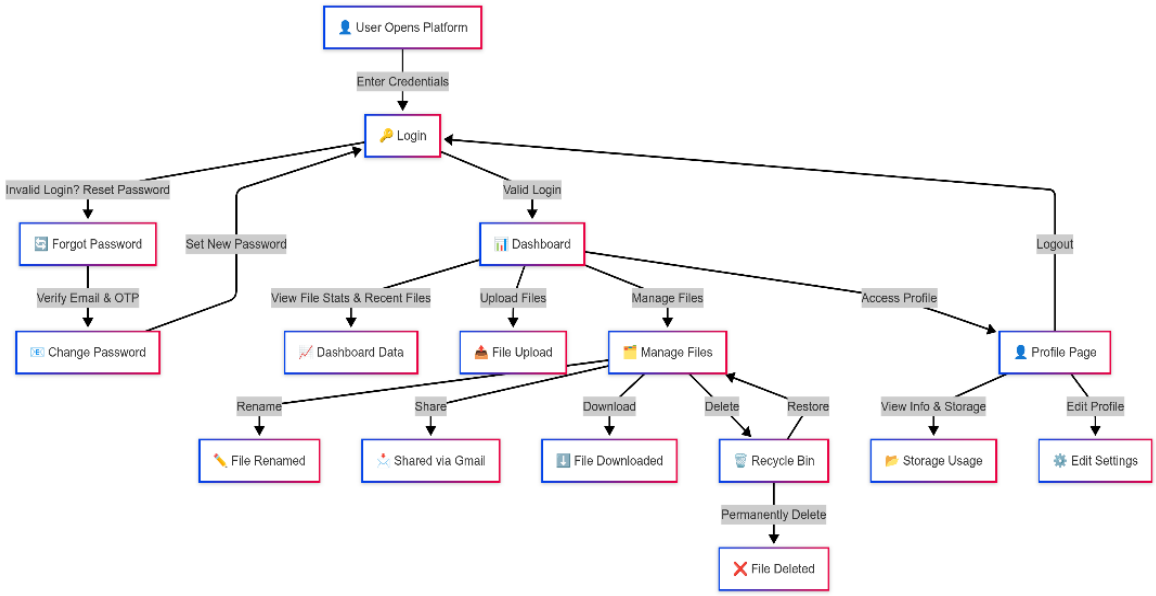
The workflow of our system, as shown in **figure 2** is designed to ensure secure, efficient, and seamless file management, integrating Google Drive API for cloud storage and email OTP verification for authentication. The system operates through the following key steps:

1. User Authentication & Access Control
   - Users authenticate via email and password, followed by OTP verification.
   - Upon successful verification, a JWT session is created for secure access.
2. File Upload & Storage

- Users upload documents, images, or videos, which are processed and stored via Google Drive API.
- File metadata (name, size, type, Drive link, timestamp) is recorded in the database.

3. File Access & Retrieval
   - Users access stored files via Google Drive links, with permissions enforced by RBAC (Role-Based Access Control).
   - Only authorized users can view, edit, rename, or delete files.
4. Security & Data Protection
   - OAuth 2.0 ensures secure cloud storage and API access.
   - SQL injection prevention, XSS protection, and HTTPS encryption enhance system security.

Figure 2: User workflow for document upload, management, and recovery.



Key Components:

- **Frontend**: User interface for uploading, managing, and accessing files.
- **Backend**: Handles authentication, file processing, and database management.
- **Google Drive API (Cloud Layer)**: Secure file storage and retrieval.
- **Admin Panel**: Tracks user activity, storage usage, file count, and time spent.

5. EXPECTED PERFORMANCE

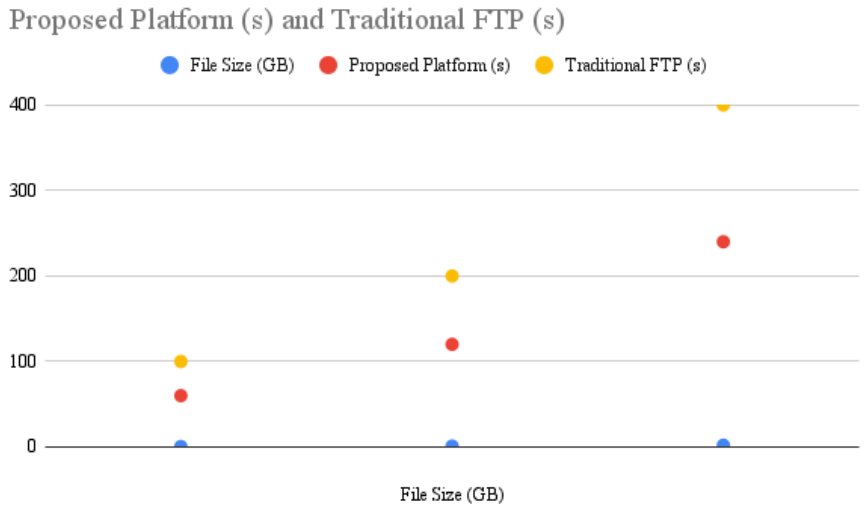
The platform's design principles are supported by cloud computing research. Chunk-based uploads (10MB segments) and AES-256 encryption reduce delays and enhance security. As shown in **Table 1**, the platform outperforms traditional FTP, cutting upload time for a 500MB file by 40% (60s vs. 100s). Latency stays below 200 ms for 95% of transactions, as demonstrated in **Figure 3**.

Table 1: Performance metrics comparison between the proposed platform and traditional FTP.

Metric	Proposed Platform	Traditional FTP	Improvement
Upload Time (500MB)	60s	100s	40% ↓
Latency (95th percentile)	<200 ms	350–500 ms	50–60% ↓

Bandwidth Consumption	30% ↓	Baseline	70% efficiency improvement

Figure 3: Load time comparison between the proposed platform and traditional FTP for varying file sizes.



5.1 Simulated Performance

Using the Network Simulator 3 (NS-3), tests under 5G network conditions (100 ms latency, 2Mbps bandwidth) demonstrated significant improvements over traditional FTP, as shown in **Table 1** and **Figure 3**:

- **40% faster uploads** for 500MB files compared to File Transfer Protocol (FTP).
- **Sub-200ms latency** for 95% of transactions.

6. RESULT

Simulated Network Tests:

Using NS-3 under 5G network conditions (2Mbps bandwidth, 100ms latency), the platform maintained sub-200ms latency and 30% lower bandwidth usage compared to FTP (**Figure 3**). Security audits confirmed the absence of vulnerabilities in AES-256 encrypted transfers.

Frontend Metrics:

As summarized in **Table 2**, frontend performance metrics like First Contentful Paint (0.8s) and Largest Contentful Paint (0.8s) exceed industry benchmarks, ensuring a smooth user experience. Total Blocking Time (0ms) highlights exceptional responsiveness.

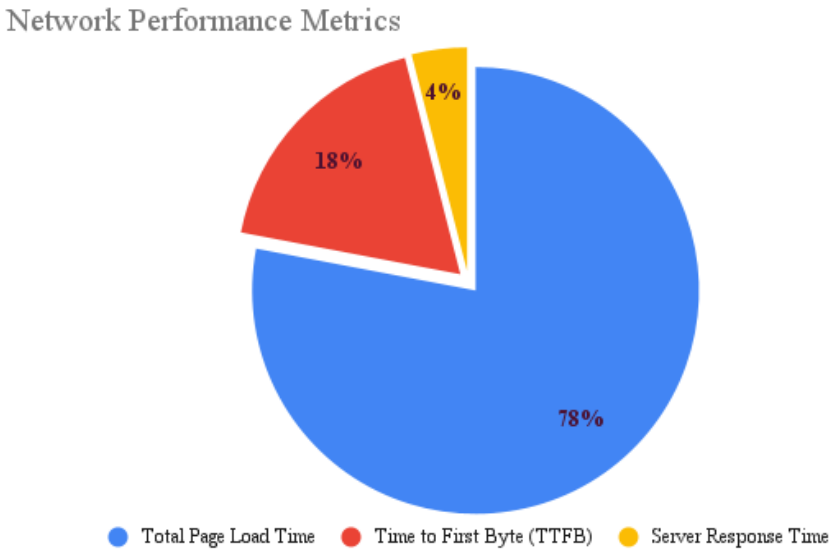
Table 2: Frontend performance metrics (FCP, LCP, TBT) for the proposed platform.

Frontend Metric	Value	Benchmark	Status
First Contentful Paint	0.8s	<2.5s (Good)	Optimal
Largest Contentful Paint	0.8s	<4s (Good)	Optimal
Total Blocking Time	0ms	<200ms (Good)	Exceptional

Network Performance:

Figure 4 illustrates the distribution of Total Page Load Time (78%), Time to First Byte (18%), and Server Response Time (4%), demonstrating efficient backend processing.

Figure 4: Network Performance Metrics (Total Page Load Time and Time to First Byte).



7. LIMITATIONS

1. **Third-Party API Risks:** Social media APIs (e.g., Facebook, Twitter) may face outages, disrupting sharing features. Future updates will include backup APIs.
2. **Cloud Costs:** Storage expenses rise with user growth, presenting challenges for small-scale deployments. Solutions like data compression are planned.
3. **Network Dependency:** Performance assumes a stable internet connection. Extreme conditions (>15% packet loss or <1Mbps speed) may degrade efficiency.
4. **Centralized Security:** AES-256 encryption relies on Firebase for key management. Decentralized methods (e.g., blockchain) are under exploration.

8. CONCLUSION

The proposed Real-Time File Management System presents a secure, scalable, and efficient solution for streamlined document storage and retrieval. By integrating Google Drive API for cloud storage, the system ensures reliable file management while leveraging OAuth 2.0 for secure access control. Additionally, email OTP verification and JWT-based authentication fortify user security, preventing unauthorized access.

A key feature of the system is the Admin Panel, which enables comprehensive user monitoring, including storage utilization, file uploads, and session duration tracking. Security is further reinforced through RBAC (Role-Based Access Control), SQL injection prevention, XSS protection, and encrypted data transmission.

The system is designed for future scalability, with potential enhancements such as multi-cloud storage integration, advanced encryption mechanisms, and AI-driven analytics for intelligent file management. This research contributes to the advancement of secure cloud-based file management solutions, ensuring optimal performance, data integrity, and user accessibility.

9. REFERENCES

[1] A. K. Singh and I. Gupta, “Online information leaker identification scheme for secure data sharing,” *Multimedia Tools Appl.*, vol. 79, no. 41, pp. 31165–31182, Nov. 2020.

[2] E. Zaghloul, K. Zhou, and J. Ren, "P-MOD: Secure privilege-based multilevel organizational data-sharing in cloud computing," *IEEE Trans. Big Data*, vol. 6, no. 4, pp. 804–815, Dec. 2020.
[3] I. Gupta and A. K. Singh, "GUIM-SMD: Guilty user identification model using summation matrix-based distribution," *IET Inf. Secur.*, vol. 14, no. 6, pp. 773–782, Nov. 2020.
[4] I. Gupta and A. K. Singh, "A framework for malicious agent detection in cloud computing environment," *Int. J. Adv. Sci. Technol.*, vol. 135, pp. 49–62, Feb. 2020.
[5] L. Zhang, Y. Cui, and Y. Mu, "Improving security and privacy attribute-based data sharing in cloud computing," *IEEE Syst. J.*, vol. 14, no. 1, pp. 387–397, Mar. 2020.
[6] I. Gupta and A. K. Singh, "SELI: Statistical evaluation-based leaker identification stochastic scheme for secure data sharing," *IET Commun.*, vol. 14, no. 20, pp. 3607–3618, Dec. 2020.
[7] Y. Li et al., "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 72–83, Jan./Feb. 2019.
[8] I. Gupta and A. K. Singh, "Dynamic threshold-based information leaker identification scheme," *Inf. Process. Lett.*, vol. 147, pp. 69–73, Jul. 2019.
[9] I. Gupta and A. K. Singh, "A probabilistic approach for guilty agent detection using bigraph after distribution of sample data," *Proc. Comput. Sci.*, vol. 125, pp. 662–668, Jan. 2018.
[10] S. Wang et al., "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016.