www.ijarp.com ISSN 2456-9992 Page: 01-07
International Journal Advanced Research Publications
Research Article
Volume: 02
Issue: 05

MINI DRIVE: A CLOUD-BASED FILE STORAGE AND SECURE SHARING SYSTEM

***Ashutosh Pradhan, Harsh Sharma, Deepanshu**

Department of Computer Science and Engineering RD Engineering College, Ghaziabad, India.

Article Received: 21 March 2026, Article Revised: 11 April 2026, Published on: 01 May 2026

***Corresponding Author: Ashutosh Pradhan**
Department of Computer Science and Engineering RD Engineering College, Ghaziabad, India.
DOI: https://doi-doi.org/101555/ijarp.9211

ABSTRACT

In the modern digital era, the exponential growth of data has created a significant demand for efficient, scalable, and secure storage systems. Cloud computing has emerged as a dominant paradigm that provides on-demand access to computing resources and storage infrastructure. However, most commercial cloud storage platforms such as Google Drive and Dropbox require paid subscriptions and complex configurations, making them less suitable for academic and small-scale applications.

This research presents *Mini Drive*, a lightweight cloud-based file storage and sharing system developed using Node.js and Supabase. The system enables users to upload, store, retrieve, and download files securely through a web-based interface integrated with cloud infrastructure. The backend is implemented using RESTful APIs, while Supabase provides storage and database services with built-in security mechanisms such as Row Level Security (RLS).

The system follows a modular architecture to ensure scalability, flexibility, and efficient data handling. The primary objective is to demonstrate that a fully functional cloud storage system can be developed using free-tier technologies without compromising performance and security. The implementation focuses on core functionalities including file upload, file listing, and file download, along with a user-friendly interface.

KEYWORDS: Cloud Storage, File Sharing System, Supa base, Node.js, Web Application, Data Management.

1. INTRODUCTION

The rapid advancement of digital technologies has resulted in a massive increase in data generation across various domains. Organizations and individuals require efficient storage systems to manage, retrieve, and share data seamlessly. Cloud storage platforms such as Google Drive, Dropbox, and Microsoft OneDrive have become essential tools in modern computing environments.

However, these platforms often require paid subscriptions and complex infrastructure, which makes them less accessible for students and small-scale developers. Therefore, there is a need for a simplified and cost-effective cloud storage system.

Mini Drive is designed as a lightweight alternative that provides essential functionalities such as file upload, file listing, and file download using free-tier cloud services. The system integrates a frontend interface with a Node.js backend and Supa base cloud storage, ensuring secure and efficient file management.

Furthermore, the increasing dependency on cloud-based systems highlights the need for lightweight and customizable storage solutions. Mini Drive addresses this gap by focusing on simplicity, usability, and scalability.

2. LITERATURE REVIEW

Cloud storage technologies have been extensively studied in recent years. Armbrust et al. described cloud computing as a model that enables on-demand access to shared computing resources. Similarly, Mell and Grance defined cloud computing as a flexible and scalable computing model.

Modern platforms such as Google Drive and Dropbox rely on distributed storage systems and advanced security mechanisms to ensure data integrity and availability.

Recent advancements in Backend-as-a-Service (BaaS) platforms such as Firebase and Supabase have simplified backend development by providing integrated services like authentication, database, and storage. However, Firebase often requires billing activation, which limits its use for academic projects.

Supabase provides an open-source alternative with free-tier capabilities, making it suitable for building lightweight applications such as Mini Drive.

3. PROBLEM STATEMENT

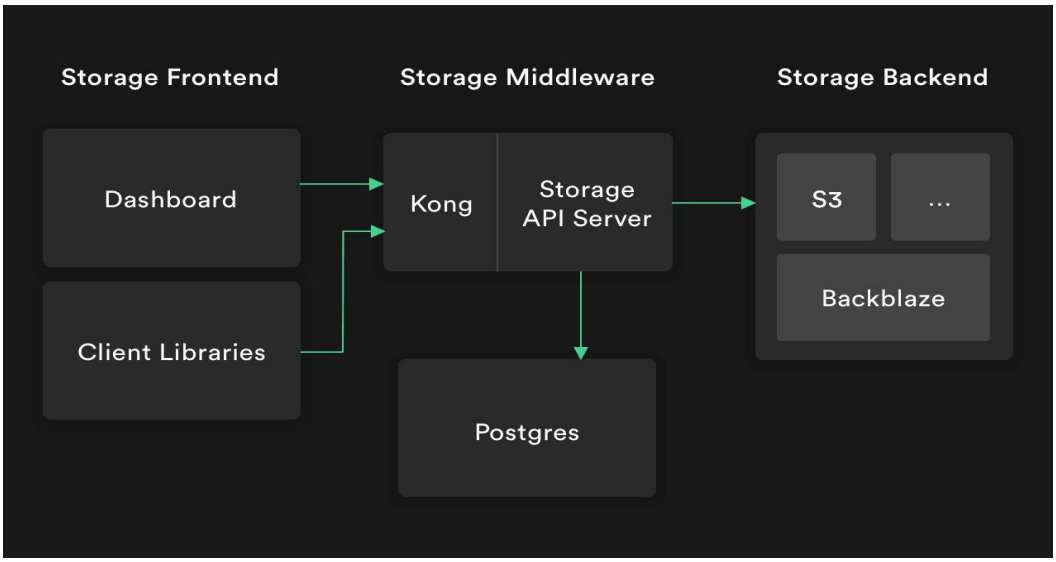
Despite the availability of advanced cloud storage platforms, many users face challenges due to high costs and complex configurations. Students and small-scale developers often require a simple and cost-effective solution for managing digital files.

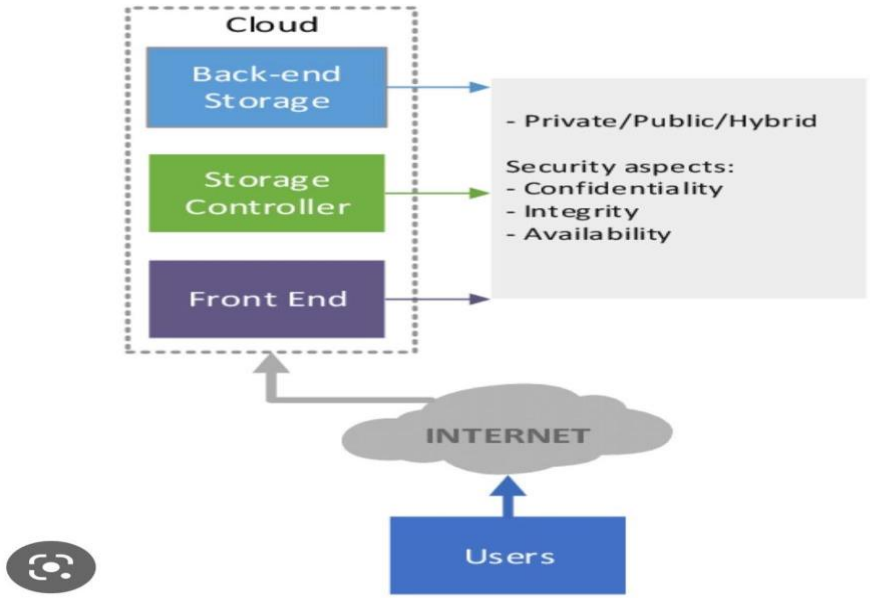
The problem addressed in this research is:

How to develop a secure, scalable, and cost-effective cloud file storage system using free-tier technologies?

4. SYSTEM ARCHITECTURE

The Mini Drive system follows a three-layer architecture consisting of the frontend, backend, and cloud storage layers.





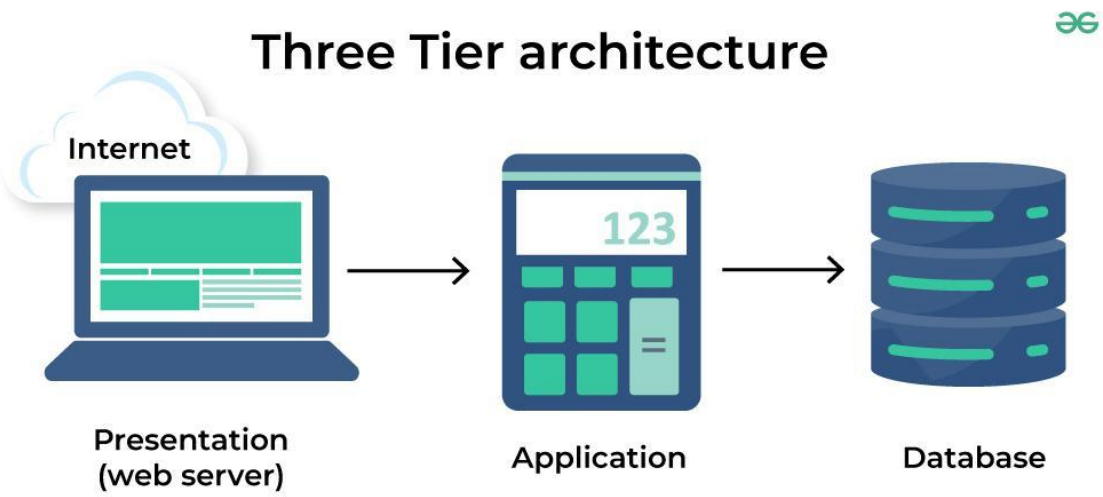
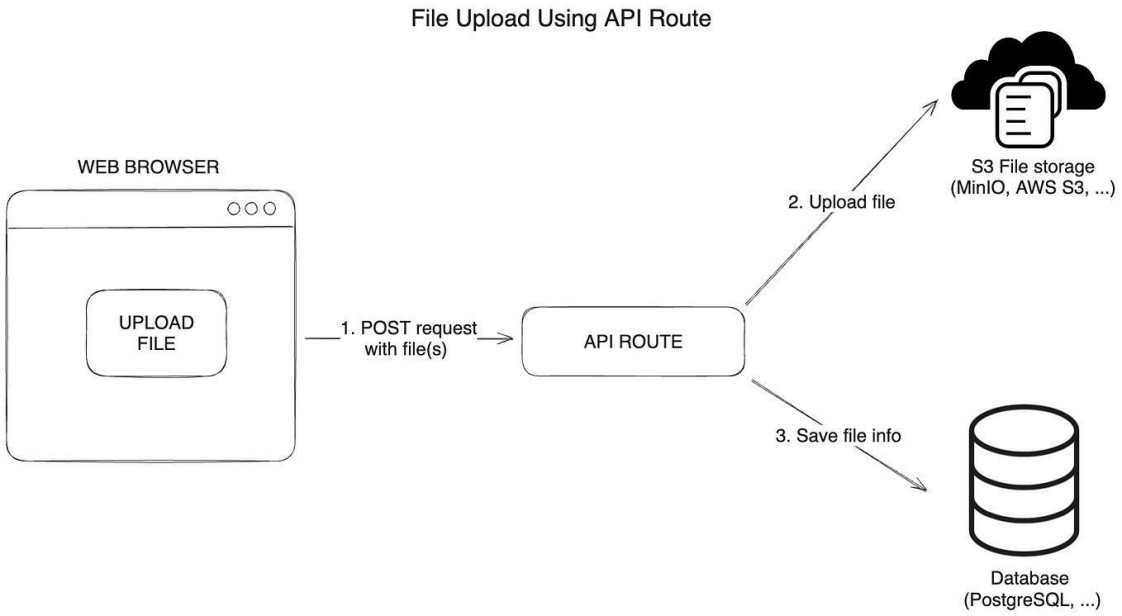


Fig 1: System Architecture of Mini Drive.

The frontend provides the user interface for file operations. The backend processes requests using Node.js and Express APIs. Supabase serves as the cloud storage and database layer.

5. METHODOLOGY

The system follows a modular approach based on RESTful API architecture. File uploads are handled using Multer middleware, which processes incoming files and stores them in Supabase storage.



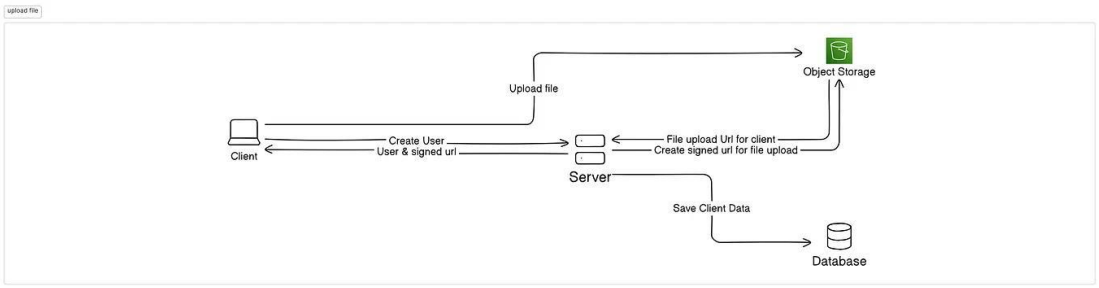


Fig 2: File Upload Workflow.

The system uses asynchronous operations to ensure efficient data handling. Unique identifiers (UUID) are used to prevent file naming conflicts.

6. IMPLEMENTATION

The implementation of Mini Drive consists of multiple modules including frontend interface and backend APIs.

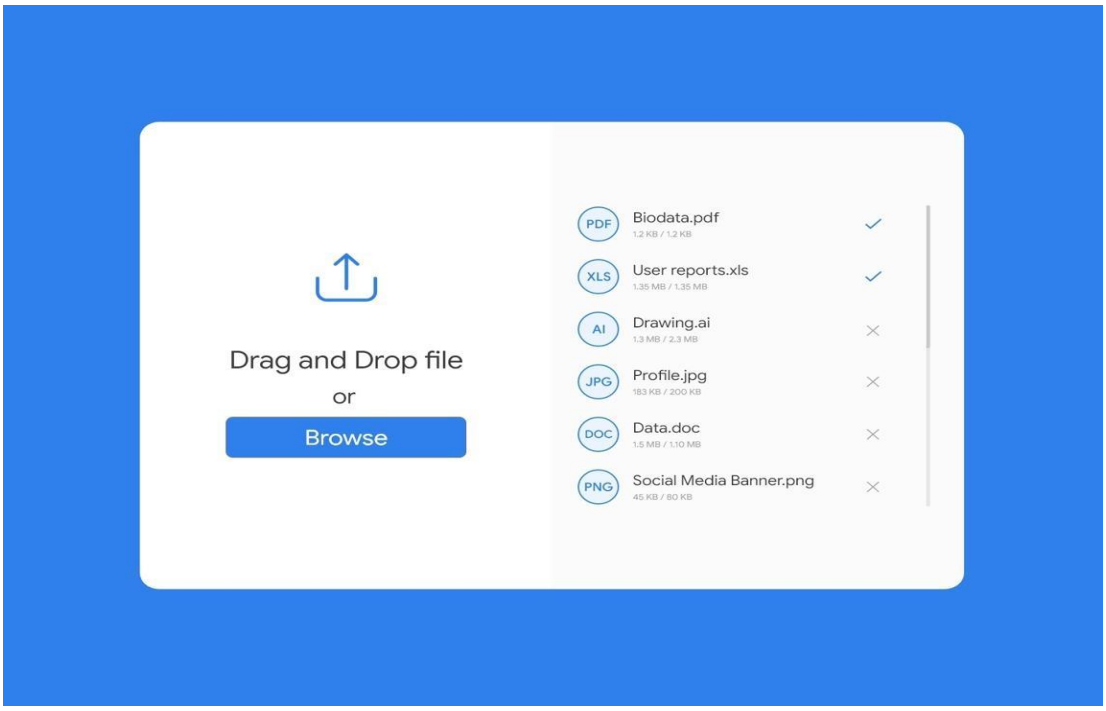


Fig 4: Mini Drive User Interface.

The frontend is developed using HTML, CSS, and JavaScript, while the backend uses Node.js and Express. Supabase is used for storing and retrieving files securely.

7. RESULTS AND DISCUSSION

The Mini Drive system was implemented and tested in a local development environment to evaluate its performance, reliability, and functionality. Various test cases were conducted

using different file formats such as images, documents, and PDF files to ensure system compatibility and stability.

The file upload functionality was successfully tested with multiple file sizes, and the system was able to store files in the Supabase cloud storage without any data loss. The use of Multer middleware ensured efficient handling of file data, while Supabase storage provided reliable and secure data persistence.

The file listing feature accurately retrieved all uploaded files from the cloud storage and displayed them in the user interface. The response time for fetching file data was observed to be minimal, indicating efficient communication between the backend server and cloud database.

The file download functionality was also tested and produced reliable results. Users were able to download files using generated links without any errors. The system ensured that file URLs were generated securely, maintaining data integrity and access control.

Performance analysis indicated that the system operates with low latency and stable response times under normal usage conditions. The modular architecture of the system allows easy scalability and future expansion.

Overall, the results demonstrate that Mini Drive successfully achieves its objective of providing a simple, efficient, and cost-effective cloud storage solution suitable for academic and small-scale applications.

8. CONCLUSION

The Mini Drive system successfully demonstrates the design and implementation of a cloud-based file storage and sharing platform using free-tier technologies. The integration of Node.js and Supabase provides a robust backend and reliable cloud storage infrastructure, enabling efficient file management operations.

The system achieves its primary objectives by implementing core functionalities such as file upload, file listing, and file download through a user-friendly interface. The use of RESTful APIs ensures seamless communication between frontend and backend components, while Supabase provides secure and scalable data storage.

One of the key strengths of the system is its simplicity and cost-effectiveness. Unlike traditional cloud storage platforms that require paid subscriptions, Mini Drive utilizes free-tier services, making it accessible for students and small-scale developers.

The modular architecture of the system allows easy integration of additional features such as user authentication, expiring share links, and access control mechanisms. These enhancements can further improve system security and usability.

In conclusion, Mini Drive serves as a practical implementation of cloud computing concepts and demonstrates that a scalable and efficient storage system can be developed without complex infrastructure. The project provides a strong foundation for future development and real-world applications.

9. REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
2. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology (NIST), 2011.
3. K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An Analysis of Security Issues for Cloud Computing," Journal of Internet Services and Applications, 2013.
4. I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, 2008.
5. Supabase Inc., "Supabase Documentation," Available: https://supabase.com/docs
6. Node.js Foundation, "Node.js Official Documentation," Available: https://nodejs.org
7. Express.js, "Express Web Framework Documentation," Available: https://expressjs.com
8. Amazon Web Services (AWS), Overview of Cloud Storage Services," Available: https://aws.amazon.com
9. Google Cloud, "Cloud Storage Overview," Available: https://cloud.google.com/storage