JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Development and Implementation of a Secure File Sharing Web Application

[1]Kaustav Jyoti Bora, [2]Dipesh Kumar Rajak, [3]Madhusudhan S, [4]Jagannath V, [5]Sumanth Naik G, [6]Dr. G. Vennira Selvi

[1,2,3,4,5]Students, [6]Professor

School of Information Science, Presidency University, Bengaluru, INDIA.

Abstract

The "File Sharing Website" project aims to create a robust, user-friendly platform for securely uploading, storing, sharing, and managing files online. This system is designed to facilitate seamless file sharing, ensuring ease of access and collaboration among users. Key features include user authentication, secure file uploads, intuitive file management, and efficient sharing capabilities. The frontend utilizes React.js for a dynamic user interface, while the backend employs Node.js and Express.js for server-side operations. MongoDB serves as the database, providing a flexible and scalable solution for storing user data and file metadata. Security measures such as JWT-based authentication, data encryption, and rigorous input validation protect user data and privacy. The system architecture is modular, supporting easy maintenance and future enhancements. The project follows agile development methodologies to ensure iterative progress and continuous improvement, with comprehensive testing strategies implemented to guarantee a reliable and bug-free user experience.

Keywords: File Sharing, Web Application, MongoDB, JavaScript, Firebase

Introduction

The purpose of this project is to develop a secure and efficient file-sharing web application that allows users to upload, store, manage, and share files seamlessly. With the increasing need for secure data sharing platforms, this project addresses the critical aspects of security, user experience, and scalability. The application is designed to provide a straightforward and efficient way for users to handle their file-sharing needs, ensuring that data privacy and integrity are maintained.

Scope

The scope of this project encompasses the development of a web-based platform that includes functionalities such as user authentication, file upload and download, file sharing, and file management. The system is designed to be scalable and secure, capable of handling a large number of users and files.

Objectives

- Develop a secure and user-friendly file-sharing platform.
- Implement robust user authentication and authorization mechanisms.
- Ensure secure file storage and transmission.
- Provide an intuitive user interface for efficient file management.
- Facilitate seamless file sharing among users.

Literature Review

Existing Solutions

Several existing solutions provide file sharing and management capabilities, such as Google Drive, Dropbox, and OneDrive. While these platforms are robust, they often come with limitations in terms of customization, control over data, and cost for premium features. This project aims to offer a comparable service with enhanced security features and greater flexibility for users.

Security Concerns

Security is a significant concern in file-sharing applications. According to the OWASP Top Ten, common security risks include injection attacks, broken authentication, and sensitive data exposure. This project incorporates measures to mitigate these risks, including JWT-based authentication, data encryption, and input validation.

System Architecture

The system architecture of the File Sharing Web Application is designed to be modular and scalable. It comprises three main components: the frontend, backend, and database.

Frontend

The frontend of the application is built using React.js, which provides a dynamic and responsive user interface. The design follows modern UI/UX principles to ensure a positive user experience. Key features include a clean layout, easy navigation, and real-time feedback during file operations.

User Interface Design

The user interface is designed to be intuitive and user-friendly. The layout is clean and minimalistic, with clear navigation menus and real-time feedback during file operations. Users can easily upload, download, and share files with just a few clicks.

Backend

The backend is developed using Node.js and Express.js, offering a robust and scalable server-side environment. The backend handles all server-side operations, including user authentication, file processing, and communication with the database. Security is enforced through JWT-based authentication and data encryption.

API Design

The backend provides a RESTful API for the frontend to interact with. This API includes endpoints for user authentication, file upload and download, and file sharing. Each endpoint is secured using JWT tokens to ensure that only authenticated users can access the API.

Database

MongoDB is used as the database for this project, providing a flexible and scalable solution for storing user data and file metadata. MongoDB's document-oriented structure allows for efficient handling of large volumes of data and complex queries.

Data Schema

The data schema is designed to be flexible and scalable. User data and file metadata are stored in separate collections, allowing for efficient querying and indexing. Each file is associated with metadata such as the filename, size, upload date, and owner.

Implementation

Development Methodology

The project employs agile development methodologies, allowing for iterative development, continuous feedback, and early detection of issues. This approach enhances flexibility and responsiveness, ensuring that the project can adapt to changing requirements and deliver incremental improvements over time.

Security Measures

Security is a paramount concern in the development of this application. Several measures are implemented to protect user data and ensure privacy:

- **JWT-based Authentication**: Secure authentication and authorization of users.
- **Data Encryption**: Encryption of files during storage and transmission.
- **Input Validation**: Rigorous validation of user inputs to prevent injection attacks and other security threats.

File Handling

Files uploaded by users are stored securely on the server. Each file is associated with metadata such as the filename, size, upload date, and owner. Files are encrypted during storage and transmission to ensure that they cannot be accessed by unauthorized users.

Testing

Comprehensive testing strategies are employed to ensure the reliability and functionality of the application. These include unit testing, integration testing, and user acceptance testing. Automated testing tools are used to streamline the testing process and ensure coverage of all critical functionalities.

Unit Testing

Unit tests are written for individual components and functions to ensure that they work as expected. These tests are run automatically whenever changes are made to the codebase.

Integration Testing

Integration tests are used to verify that different components of the application work together correctly. These tests simulate real-world usage scenarios to ensure that the application functions as expected.

User Acceptance Testing

User acceptance testing involves testing the application with real users to gather feedback and identify any issues that need to be addressed. This testing phase helps ensure that the application meets the needs and expectations of its users.

Conclusion

The File Sharing Web Application project successfully achieves its objectives by providing a secure, user-friendly platform for file sharing and management. The use of modern technologies and agile development practices ensures that the application is scalable, maintainable, and capable of adapting to future needs. The emphasis on security and user experience positions this application as a reliable solution for secure online file sharing.

Future Work

Future enhancements to the application could include additional security features, such as two-factor authentication, and support for larger file sizes and higher volumes of data. Additionally, the application could be expanded to include collaboration features, such as real-time editing and commenting on shared files.

References

1. OWASP Top 10: The ten most critical web application security risks
2. Mozilla Developer Network - Security: Comprehensive guide to web application security concepts
3. GitHub: Version control platform for hosting and collaborating on code
4. Stack Overflow: Question and answer forum for developers