



Freesite.me: A Lightweight Static Web Hosting Platform

Assistant Prof. Rakesh Jain

rakesh.jain@indoreinstitute.com

Rituraj Kumar

Department of Computer
Science and Engineering
Indore Institute of Science and
Technology, Indore (M.P.) India
DevRiturajKumar@gmail.com

Shashank Nalkande

Department of Computer Science
and Engineering
Indore Institute of Science and
Technology, Indore (M.P.) India
ShashankNalkande1@gmail.com

ABSTRACT—

The rapid growth of digital content creation has increased the demand for simple and accessible web hosting solutions. However, traditional hosting platforms often require technical expertise in server configuration, deployment procedures, and website management, which creates challenges for beginners and non-technical users. This paper presents the design and development of a lightweight static website hosting platform that enables users to deploy websites quickly and efficiently through a user-friendly web interface.

The proposed platform allows users to upload static web files, including HTML, CSS, and JavaScript resources, without requiring manual server-side configuration. To improve usability and security, the system incorporates features such as password-protected editing access and optional privacy controls for hosted websites. Uploaded content is automatically verified, organized, and published under a unique user-defined domain path, ensuring a streamlined deployment experience.

The architecture of the platform is designed to support efficient file processing, secure content management, and reliable hosting operations. Automated handling of website creation, modification, and deletion minimizes administrative overhead while improving scalability and accessibility. The system aims to provide a cost-effective and simplified hosting environment suitable for students, beginner developers, freelancers, and small organizations.

Experimental evaluation demonstrates that the platform reduces deployment complexity and significantly lowers the technical barriers associated with static website publishing. The proposed solution highlights the potential of automated hosting systems in promoting accessible and scalable web deployment services.

KEYWORDS —

Static Website Hosting, Web Hosting Platform, Automated Deployment, Flask Framework, Secure Site Hosting.

I. INTRODUCTION

[Freesite.me](https://freesite.me) is a lightweight static website hosting platform designed to simplify website deployment for users without technical hosting knowledge. The platform enables secure and instant publishing of HTML, CSS, and JavaScript websites through an automated and user-friendly web interface.

The increasing dependence on digital platforms has made websites an essential medium for communication, business promotion, and information sharing. Despite the growing importance of online presence, many individuals and small organizations still face challenges in deploying websites due to the technical complexity and financial cost associated with conventional web hosting services. Server configuration, domain management, and deployment procedures often require specialized knowledge, making website hosting difficult for beginners and non-technical users.

This project introduces a lightweight and user-friendly web hosting platform designed



specifically for static website deployment. The proposed system enables users to upload HTML, CSS, and JavaScript files through a simplified web interface, allowing websites to be published instantly without requiring manual server administration or advanced hosting knowledge. Each uploaded website is automatically assigned a unique and accessible URL, simplifying the deployment workflow for end users.

To improve reliability and security, the platform incorporates features such as password-protected editing access, optional visitor authentication, automated file validation, and conflict-free file management. These mechanisms ensure secure hosting operations while maintaining a seamless user experience. The system architecture is designed to automate hosting tasks efficiently, reducing operational complexity and improving scalability.

The platform is particularly beneficial for students, beginner developers, educators, freelancers, and small businesses seeking a low-cost and accessible hosting environment. By minimizing technical barriers and automating deployment operations, the proposed solution promotes wider participation in web publishing and contributes toward a more open and accessible internet ecosystem.

II. PROPOSED METHOD

The proposed Static Site Hosting and Management Platform was developed using the Agile Iterative and Incremental development model to ensure flexibility, continuous improvement, secure implementation, and efficient feature integration. The methodology supports repeated testing, refinement, and gradual enhancement of the system throughout the development lifecycle.

1. Requirement Analysis

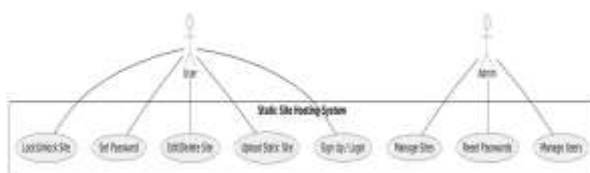


Fig. 1. Use Case Diagram of the Proposed Platform

The initial phase focused on identifying the major functional and non-functional requirements of the platform. Core functionalities such as secure file uploads, instant website deployment, authentication, and website management were analyzed carefully. Additional features including role-based access control, privacy protection, analytics, and upload conflict handling were also considered. Non-functional requirements such as scalability, usability, reliability, maintainability, and security were evaluated through competitor analysis, surveys, prototyping, and user feedback.

2. System Design

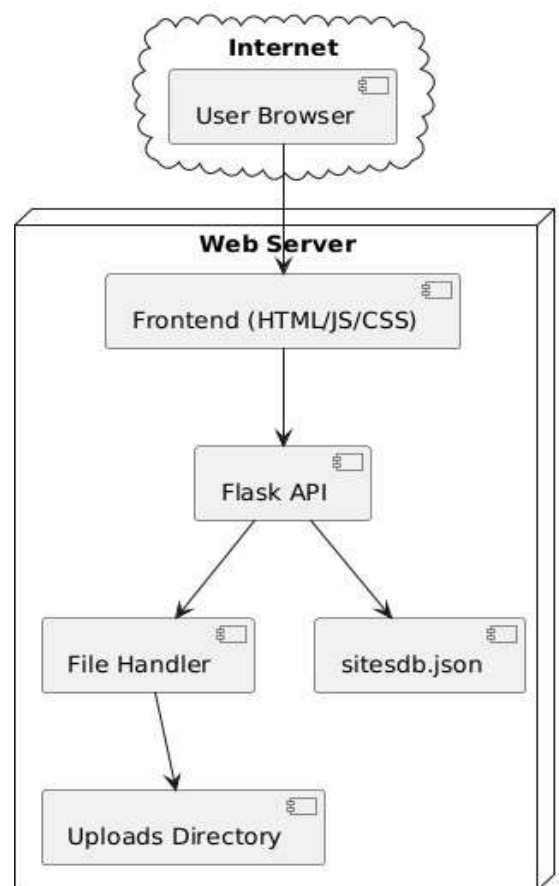


Fig. 2. System Architecture of the Proposed Platform

The system was designed using a modular architecture consisting of a Flask-based backend, responsive frontend interface, and structured file management workflow. JSON-based metadata storage was implemented for efficient website and configuration management. Security mechanisms such as session-based authentication, SHA-256 password hashing, ZIP validation, and path traversal prevention were integrated to ensure



secure hosting operations. UML diagrams and workflow models were also prepared to represent system processes, API interactions, and user workflows.

3. Incremental Development

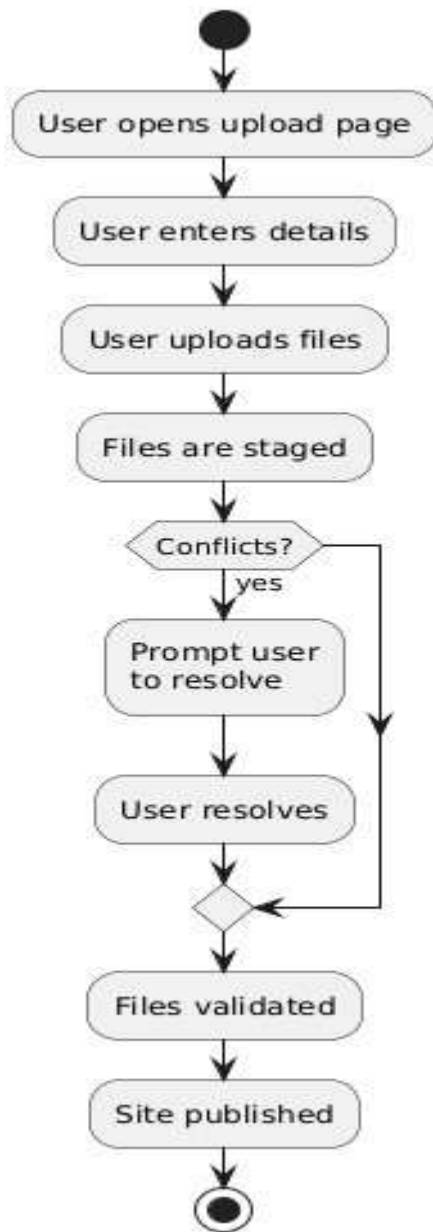


Fig. 3. Activity Diagram of the Proposed Platform

The platform was developed incrementally through multiple Agile iterations to ensure stable and manageable implementation.

A. Iteration 1 – Core Hosting Engine



Fig. 4. Homepage of the Proposed Platform

The first iteration focused on implementing the basic Flask server setup, file upload handling, ZIP extraction, and static website serving functionalities.

B. Iteration 2 – Security and Conflict Handling

The second iteration introduced staging directory workflows for upload conflict resolution along with ZIP-slip prevention, path traversal protection, and secure password validation mechanisms.

C. Iteration 3 – Management Features



Fig. 5. Management Panel of the Proposed Platform



The third iteration focused on developing the Admin Dashboard and Site Owner Dashboard. APIs for website editing, renaming, deletion, visibility control, and authentication management were also implemented.

D. Iteration 4 – Analytics and Optimization

The final iteration integrated visitor analytics, hit logging, 404 suggestion mechanisms using similarity matching, and optimization of deployment and settings management workflows.

4. Testing

Comprehensive testing procedures were conducted throughout the development process to ensure system reliability and security. Unit testing, integration testing, and user acceptance testing were performed to validate authentication workflows, API endpoints, file handling operations, and deployment processes. Additional security testing was conducted to identify vulnerabilities such as unauthorized access attempts and directory traversal attacks. Both automated and manual testing techniques were used to ensure responsive behaviour and cross-browser compatibility.

5. Deployment and Maintenance

The system was deployed using WSGI-based web servers and modern hosting workflows to support reliable static website publishing. Monitoring systems, backup mechanisms, and error logging strategies were implemented to maintain system stability and performance. Future enhancements such as custom domain support, advanced analytics, and improved dashboard management are planned to further increase platform scalability and usability.

III. ADVANTAGES

1. Simplified Website Deployment

The platform enables users to deploy static websites instantly without requiring technical knowledge of server configuration or hosting management.

2. User-Friendly Interface

A clean and responsive web interface allows users to upload, manage, edit, and publish websites easily through browser-based operations.

3. Secure Hosting Environment

The system incorporates security features such as password-protected access, SHA-256 password hashing, ZIP validation, and path traversal prevention to ensure safe hosting operations.

4. Automated Hosting Operations

Website creation, file extraction, deployment, and management processes are automated, reducing manual intervention and improving efficiency.

5. Cost-Effective Solution

The platform provides free static website hosting, making it suitable for students, beginners, freelancers, and small organizations with limited resources.

6. Scalable Architecture

The modular backend architecture supports future scalability, feature expansion, and integration of advanced hosting functionalities.

7. Efficient File Management

The system handles upload conflicts, metadata management, and structured file organization efficiently to maintain reliable hosting performance.

8. Cross-Platform Accessibility

The platform is accessible through modern web browsers and supports responsive operation across desktops, tablets, and mobile devices.

IV. SPECIFICATIONS

1. Hardware Requirements

A. Minimum Hardware Requirements

Users accessing the platform require a basic system configuration consisting of an Intel Core i3 processor or equivalent, a minimum of 4 GB RAM, and approximately 20 GB of available storage space for website files and browser operations. A stable broadband internet connection is necessary for smooth website uploading, management, and deployment activities.



B. Recommended Hardware Requirements

For improved performance and faster website deployment, users are recommended to use systems equipped with an Intel Core i5/i7 processor or equivalent, 8 GB RAM or higher, and SSD-based storage. These configurations provide better responsiveness during large file uploads, dashboard operations, and concurrent hosting management tasks.

2. Software Requirements

A. Operating System

The platform is accessible on multiple operating systems including Windows 10/11 and Linux-based distributions such as Ubuntu and CentOS. Since the system operates through a browser-based interface, users can access the platform without requiring specialized operating system configurations.

B. Backend Technologies

The hosting platform is powered by Python and the Flask framework, which manage backend operations such as authentication, website deployment, file processing, and secure hosting workflows.

C. Frontend Technologies

The user interface is developed using HTML5, CSS3, and JavaScript (ES6) to provide a responsive and interactive hosting experience. AJAX-based communication is implemented to support smooth and real-time interaction between the client and server.

D. Database and Storage

The system uses lightweight JSON-based metadata storage for managing hosted website information and user configurations. Uploaded website files are maintained through organized local file system management.

E. Web Server and Deployment

Website deployment and hosting operations are supported using web servers such as Nginx or Apache combined with Gunicorn/uWSGI for efficient application handling and static website serving.

F. Development Tools

The development and maintenance of the platform utilize tools such as Visual Studio Code, Postman, Git, and GitHub for coding, API testing, debugging, and version control management.

3. Security Specifications

A. Authentication Security

The platform implements SHA-256 password hashing and session-based authentication mechanisms to secure user credentials and protect access to website management features.

B. File Security

Advanced file validation techniques including ZIP-slip prevention and path traversal protection are integrated to ensure secure website uploads and prevent malicious file execution.

C. Access Control

The system provides access control and privacy management features that allow users to restrict website visibility and secure editing operations through authentication-based protection.

4. Performance Specifications

A. Hosting Performance

The platform is optimized for efficient static website hosting, fast file uploads, instant deployment, and responsive dashboard operations to improve overall user experience.

B. Scalability and Reliability

The modular architecture supports concurrent user access, scalable hosting workflows, and reliable error recovery mechanisms to maintain stable platform performance under varying workloads.

V. FUTURE SCOPE

The proposed static website hosting platform has significant potential for future enhancement and scalability. Several advanced features and optimizations can be integrated to improve functionality, security, and user experience.

1. Custom Domain Support

Future versions of the platform can allow users to connect and manage custom domain names for professional website deployment.

2. Cloud-Based Storage Integration

Integration with cloud storage services can improve scalability, backup management, and distributed hosting performance.

3. Advanced Analytics Dashboard

The platform can include detailed analytics features such as visitor tracking, traffic



visualization, performance monitoring, and geographic usage statistics.

4. SSL Certificate Automation

Automatic SSL certificate generation and renewal can be integrated to provide secure HTTPS-based website hosting for all users.

5. Multi-User Collaboration

Future development may support collaborative website management where multiple users can manage and edit the same hosted project.

6. Database Hosting Support

The system can be extended to support dynamic websites and database-driven applications in addition to static website hosting.

7. AI-Based Security Monitoring

Artificial intelligence techniques can be implemented to detect malicious uploads, suspicious activities, and abnormal traffic patterns automatically.

8. Mobile Application Support

A dedicated mobile application can be developed to allow users to upload, manage, and monitor websites directly from smartphones and tablets.

9. Automated CI/CD Integration

Future versions may integrate GitHub-based CI/CD pipelines for automatic deployment and continuous website updates.

10. Improved Scalability and Load Balancing

Advanced server clustering and load balancing mechanisms can be implemented to support large-scale concurrent hosting operations and high traffic workloads.

VI. CONCLUSION

The platform has been successfully deployed and tested through the [Freesite.me](https://freesite.me) hosting environment.

The proposed Static Site Hosting and Management Platform provides a simplified, secure, and cost-effective solution for deploying static websites without requiring advanced technical knowledge. The system successfully automates important hosting operations such as file uploading,

validation, deployment, and website management through a user-friendly interface.

The platform incorporates security mechanisms including authentication management, password protection, ZIP validation, and path traversal prevention to ensure safe and reliable hosting operations. Its modular architecture and scalable design support efficient file management, responsive performance, and concurrent user handling.

The project demonstrates that automated static hosting platforms can significantly reduce the complexity associated with traditional website deployment methods. By improving accessibility and minimizing technical barriers, the proposed solution creates new opportunities for students, beginner developers, freelancers, educators, and small businesses to publish and manage websites efficiently.

REFERENCES

- [1] GitHub Pages Documentation, 2025. Available: <https://docs.github.com/en/pages>
- [2] Firebase Hosting Documentation, 2025. Available: <https://firebase.google.com/docs/hosting>
- [3] Flask Documentation, 2025. Available: <https://flask.palletsprojects.com/>
- [4] Python Documentation, 2025. Available: <https://docs.python.org/3/>
- [5] Mozilla Developer Network (MDN), "Web Technologies Documentation," 2025. Available: <https://developer.mozilla.org/>
- [6] M. Fowler, *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.
- [7] R. C. Martin, *Clean Code*. Prentice Hall, 2008.
- [8] AWS Amplify Hosting Documentation, 2025. Available: <https://docs.amplify.aws/>