

Host Genius-Smart Deployment Platform

A.Alagar¹, A. Hameetha Jasra², R.Dharshini³

¹ Assistant Professor, Dept of CSE

^{2,3}Dept of CSE

^{1,2,3} K.L.N. College of Engineering, Sivagangai, India

Abstract- Deploying web applications often requires developers to perform multiple manual tasks, including project configuration, dependency installation, build validation, and cloud hosting. These processes can be challenging for beginners and time-consuming for developers. To address these limitations, this paper proposes Host Genius-Smart Deployment Platform, a web-based system that automates the deployment workflow. The system uses React.js for the frontend and Node.js with Express.js for backend processing. It allows users to upload project files in ZIP format, analyze project structures, validate configurations, and build supported web applications. The Platform integrates Firebase Hosting and Netlify to support cloud deployment and provides deployment status, live URLs, and deployment history. The system makes deployment easier, reduces manual work, and helps developers deploy applications quickly.

I. INTRODUCTION

Web application deployment is an essential stage in software development, enabling developers to make their applications accessible to users through the internet. Traditional deployment processes often require manual configuration, dependency management, build execution, and hosting setup. These tasks may create difficulties for beginners and increase development time. Although several cloud hosting platforms provide deployment services, developers still need to prepare their projects and configure the deployment process correctly.

Host Genius-Smart Deployment Platform introduces a centralized web-based solution that simplifies these activities. The system analyzes uploaded project files, validates project configurations, performs build operations, and supports deployment through Firebase Hosting and Netlify.

II. LITERATURE SURVEY

Existing Cloud deployment platforms provide services for hosting and publishing web applications. Platforms such as Firebase Hosting and Netlify simplify deployment by providing hosting infrastructure and deployment tools. Modern web development frameworks, including React.js, support the creation of interactive user

interfaces, while Node.js and Express.js enable backend processing and API integration.

However users may still need to understand project Structures, configure build settings, and resolve deployment errors manually. Automated build systems help identify configuration issues and prepare applications for deployment, while authentication and database services support secure access and project information management. These technologies provide the foundation for Host Genius-smart Deployment Platform, which combines project upload, automated analysis, validation, build processing, and cloud deployment within a single interface.

III. PROPOSED METHODOLOGY

The proposed system integrates multiple modules to simplify the web application deployment process. The frontend, developed using React.js and Vite, provides an interactive interface for authentication, project uploads, deployment configuration, and status monitoring. The backend, developed using Node.js and Express.js, manages API requests, ZIP extraction, project analysis, validation, and deployment operations. The system identifies supported project structure, including React/Vite applications and static HTML projects. The build module checks project configuration and executes the required build process. Firebase Hosting and Netlify integrations enable users to deploy supported projects to their selected hosting provider. Firebase services support authentication and project information storage. The platform also records deployment status, live URLs, and deployment history, enabling users to track their deployment activities.

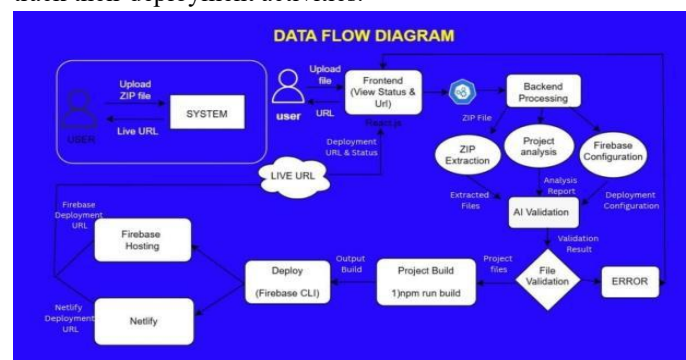


Figure 1 : Dataflow diagram of our proposed module

IV. SYSTEM PROCESS FLOW

1. Project Upload:

Users upload their web application source code as a ZIP file through the platform.

2. Project Extraction and Analysis:

The uploaded ZIP file is extracted, and the system analyzes its structure to identify supported frameworks, configuration files, and entry points.

3. Project Validation:

The system checks essential project files and configurations to identify potential issues before the build process.

4. Build Processing:

The backend installs the required dependencies and executes the appropriate build process for supported projects.

5. Cloud Deployment:

User select firebase hosting or Netlify, and the system processes the deployment through the corresponding integration.

6. Result Visualization:

The platform displays deployment status, the generated live URL when available, and deployment history through the user interface.

V. RESULT AND DISCUSSIONS

The Host Genius- Smart Deployment Platform prototype integrates project upload, automated analysis, validation, build processing, and cloud deployment into a single workflow. The system supports React/Vite and static HTML project structures and provides Firebase Hosting and Netlify deployment options. The user interface displays project information, deployment progress, and deployment history to help users monitor their applications. Integration with cloud hosting services reduces the need to perform every deployment step manually. The platform also stores deployment information, allowing users to access previously recorded deployment details. These features demonstrate how automated project processing and cloud hosting integration

can simplify web application deployment, particularly for beginners and developers seeking a centralized deployment workflow.

VI. CONCLUSION

This paper presents HostGenius- Smart Deployment Platform, a web- based solution designed to simplify and automate web application deployment. By combining React.js, Node.js, Express.js, Firebase services, and Netlify integration, the system supports project analysis, validation, build processing, cloud deployment, and deployment history management. Unlike workflows that require developers to perform each step independently, the proposed platform provides a centralized interface for managing the deployment process. Future enhancement include supporting additional programming frameworks, improving deployment error diagnostics, strengthening security validation, and expanding deployment monitoring capabilities. The system makes web application deployment easier and more efficient.

REFERENCES

- [1] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
- [2] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to build high-performance network programs", IEEE Internet Computing, vol. 14, no. 6, pp. 80-83, Nov.-Dec. 2010.
- [3] I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson.
- [4] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison- Wesley, 2002.
- [5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns :Elements of Reusable Object-Oriented Software. Reading, MA, USA: Addison-Wesley, 1994.
- [6] R.S. Pressman and B.R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill, 2019.