



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Formly

Mr. Siddhesh Pravin Patil¹, Mr. Sojal S Patil², Mr. Sainath N Pawar³, Ms. Pallavi Marulkar⁴

¹ Department of Computer Engineering, PHCET, Rasayani

Email: siddheshpp22hcompe@student.mes.ac.in

² Department of Computer Engineering, PHCET, Rasayani

Email: sojalsp22hcompe@student.mes.ac.in

³ Department of Computer Engineering, PHCET, Rasayani

Email: sainathnp22hcompe@student.mes.ac.in

⁴ Department of Computer Engineering, PHCET, Rasayani

Email: Pallavimarulkar@mes.ac.in

ABSTRACT:

FormifyAI is an advanced Next.js-based platform designed to revolutionize web form handling and automation. Utilizing React, Tailwind CSS, and Prisma, it offers a scalable solution for seamless form submissions with minimal manual intervention. The system ensures real-time data validation, reducing errors and improving user experience. Its backend, powered by Prisma, guarantees secure and efficient data storage, while the Next.js framework enhances performance with fast load times. With a dynamic and responsive UI, FormifyAI adapts to various devices, ensuring accessibility and ease of use. Whether for simple contact forms or complex multi-step processes, it optimizes accuracy, efficiency, and productivity, making it an indispensable tool for developers, businesses, and end users.

Keywords: Form automation, Next.js, React, Tailwind CSS, Prisma, form validation, data processing, workflow optimization, digital transformation, database management, containerization, cloud deployment.

INTRODUCTION

In today's fast-paced digital landscape, organizations rely heavily on efficient data collection and management for a variety of operations, including customer feedback, surveys, application processing, and business transactions. However, traditional form management solutions whether paper-based or static digital forms come with several limitations. Paper forms are prone to damage, loss, and labor-intensive manual processing, making them inefficient and error-prone. Meanwhile, static digital forms, such as those offered by platforms like Google Forms, Typeform, and Microsoft Forms, provide only basic data collection functionalities. They often lack automation, intelligent data processing, and seamless integration with external systems, which are essential for organizations handling large volumes of complex data. This results in inefficient workflows, increased operational costs, and limited real-time insights, ultimately hampering decision-making and productivity.

However, traditional form management solutions whether paper-based or static digital forms come with several limitations. Paper forms are prone to damage, loss, and labor-intensive manual processing, making them inefficient and error-prone. Meanwhile, static digital forms, such as those offered by platforms like Google Forms, Typeform, and Microsoft Forms, provide only basic data collection functionalities. They often lack automation, intelligent data processing, and seamless integration with external systems, which are essential for organizations handling large volumes of complex data. This results in inefficient workflows, increased operational costs, and limited real-time insights, ultimately hampering decision-making and productivity.

To address these challenges, **Formly** has been developed as a comprehensive, intelligent form management solution that goes beyond traditional data collection. Designed to meet the evolving needs of modern businesses, Formly enables users to create dynamic, interactive forms with real-time validation, reducing errors and streamlining data input. Additionally, it automates approval processes, integrates effortlessly with third-party applications, and leverages AI-driven insights to enhance decision-making. Security is also a key focus, with robust measures such as end-to-end encryption and multi-factor authentication ensuring the protection of sensitive information. By offering an intuitive, scalable, and highly secure platform, Formly optimizes digital workflows, reduces manual effort, and enhances overall efficiency. This paper delves into the architecture, development methodology, and key advantages of Formly, demonstrating its significance as a next-generation solution for intelligent form management.

RELATED WORK

Several existing solutions have attempted to improve form automation and data management using modern web technologies. Platforms like Google Forms and Typeform offer basic functionalities for creating and collecting form responses, but they primarily focus on ease of use rather than deep automation, advanced validation, or seamless database integration. While these platforms work well for simple surveys or data collection, businesses and enterprises often require more comprehensive solutions that ensure data integrity, minimize manual errors, and integrate effortlessly into existing

workflows. Traditional form management systems lack intelligent validation, real-time data processing, and automation features, which can lead to inefficiencies, inconsistent data, and additional manual effort.

Recent research in form management highlights the significance of automated validation techniques, which help eliminate human errors and improve overall data accuracy. AI-driven solutions have demonstrated their ability to optimize form processing through predictive input suggestions, real-time error detection, and dynamic field adaptation based on user responses. Additionally, modern web frameworks like Next.js are being widely used to enhance performance through server-side rendering (SSR) and static site generation (SSG), enabling faster and more scalable form applications. Database management is also a crucial aspect of form handling, and tools like Prisma ORM have been shown to significantly improve query optimization and structured data handling, making form data retrieval more efficient and structured.

Alongside backend optimizations, user interface (UI) and user experience (UX) considerations play a crucial role in ensuring an intuitive and seamless form management system. Modern UI/UX frameworks, such as Tailwind CSS, offer a utility-first approach that accelerates the development process while maintaining a consistent and responsive design. Tailwind allows developers to create highly customizable and visually appealing interfaces with minimal CSS overhead, leading to faster load times and better user engagement. Additionally, large-scale form management solutions require cloud-native deployment strategies to maintain scalability and reliability. The adoption of containerization technologies like Docker ensures uniform performance across different environments, making it easier to deploy and manage applications on cloud platforms such as AWS, Azure, or Google Cloud. By leveraging microservices architecture, form management platforms can be designed to handle high traffic loads while maintaining modularity and ease of maintenance. These advancements in web technologies, database management, and UI/UX frameworks emphasize the importance of integrating scalable, automation-driven, and high-performance solutions into modern form processing systems, addressing the critical shortcomings of traditional approaches.

METHODOLOGY

The development of Formly, an advanced form automation system, was guided by a structured methodology to ensure efficiency, scalability, and security. The system was designed with Next.js for frontend development, Prisma ORM for database management, Clerk for authentication, and Vercel for deployment. This methodology was divided into distinct phases, covering planning, design, implementation, testing, deployment, and maintenance, ensuring a well-optimized and robust solution.

System Planning and Requirement Analysis

The planning phase involved a detailed study of existing form management solutions to identify limitations and key areas for improvement. Extensive research and user feedback helped define critical requirements, including secure authentication, dynamic form validation, real-time data handling, and seamless database integration. Security and access control were key concerns, leading to the selection of Clerk middleware for authentication, providing session management and role-based access control. Additionally, Prisma ORM was chosen to handle structured database interactions, ensuring efficient queries and data consistency. Since scalability and performance were top priorities, Vercel was selected as the deployment platform, offering automated scaling, fast content delivery, and minimal downtime.

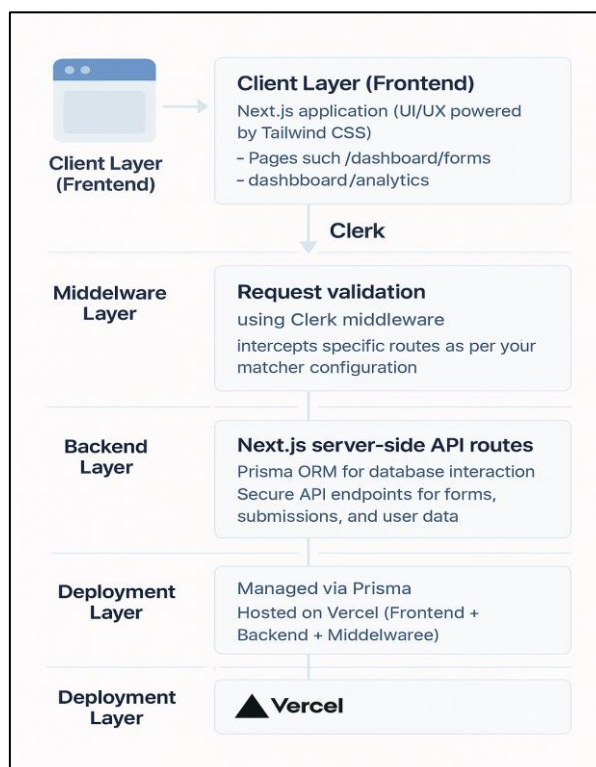


Figure 1: (System Architecture)

System Design and Architecture

The system was designed with a modular and layered architecture to maintain flexibility and ease of future enhancements. The frontend was built using Next.js and Tailwind CSS, providing a responsive, fast, and user-friendly interface while leveraging server-side rendering (SSR) and static site generation (SSG) for optimal performance. The middleware layer, powered by Clerk, acted as a security filter, ensuring that only authenticated users could access specific resources. The backend consisted of Next.js API routes integrated with Prisma ORM, facilitating efficient and secure database interactions. All components were deployed on Vercel, enabling continuous integration and delivery, further streamlining the development workflow. This structured architecture ensures high performance, security, and scalability.

Implementation Phase

The development process followed an Agile methodology, allowing for incremental improvements and user feedback integration at every stage. The frontend was designed using React-based components, enabling dynamic form creation, real-time validation, and seamless user interaction. Next.js was utilized for server-side processing, improving performance and enabling efficient rendering of form pages. Clerk was integrated for user authentication, ensuring secure login and role-based access control. Backend implementation focused on creating secure API endpoints using Next.js server-side routes, while Prisma ORM handled database operations, ensuring data integrity and optimized queries. Special attention was given to automated form validation mechanisms, minimizing user input errors and enhancing accuracy.

Testing and Quality Assurance

To ensure the system's reliability, security, and performance, a comprehensive testing strategy was employed. Unit testing was conducted on individual React components to verify user interface functionality, while integration tests checked the seamless communication between frontend, middleware, and backend services. Stress testing was performed to evaluate system stability under high user loads, ensuring it could handle large-scale deployments without performance degradation. Security audits were carried out on Clerk authentication, testing for vulnerabilities and unauthorized access attempts. Additionally, cross-browser and cross-device compatibility tests ensured that the system performed consistently across different platforms.

Deployment and Maintenance

Once testing was completed, the platform was deployed on Vercel, taking advantage of its serverless infrastructure, automated scaling, and CI/CD pipeline. Monitoring tools were set up to track system performance, detect issues, and analyze user activity for potential improvements. Regular updates and patches were planned based on performance insights and emerging security threats. The combination of automated deployment, real-time monitoring, and periodic maintenance ensures that Formly remains a highly efficient, scalable, and secure solution for form automation.

RESULTS

Formly, developed using Next.js, React, Tailwind CSS, Firebase, and Vercel, is an advanced form automation and data management system designed for seamless user interaction and efficient backend processing. It leverages modern web technologies to provide real-time validation, secure data storage, and dynamic form generation while ensuring an intuitive and responsive user experience. By integrating server-side rendering (SSR) and static site generation (SSG) with Firebase's authentication and database services, Formly optimizes performance, enhances security, and supports scalable deployment. This system addresses the limitations of traditional form management solutions by offering deep automation, structured data handling, and seamless integration with enterprise workflows, making it a versatile and reliable platform for diverse applications.

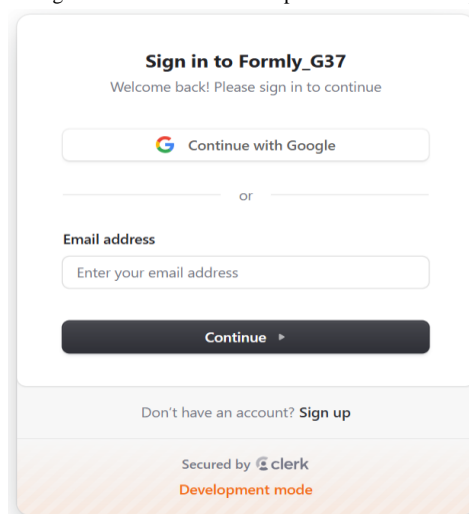


Figure 2: (Login Page)

Figure 2 (Login Page) showcases a login interface prompting users to sign in with Google, ensuring a secure authentication process powered by Clerk. Its minimalistic design prioritizes ease of use, enabling quick access without complex sign-up steps. By integrating Google authentication, the system reinforces data security through trusted verification protocols.

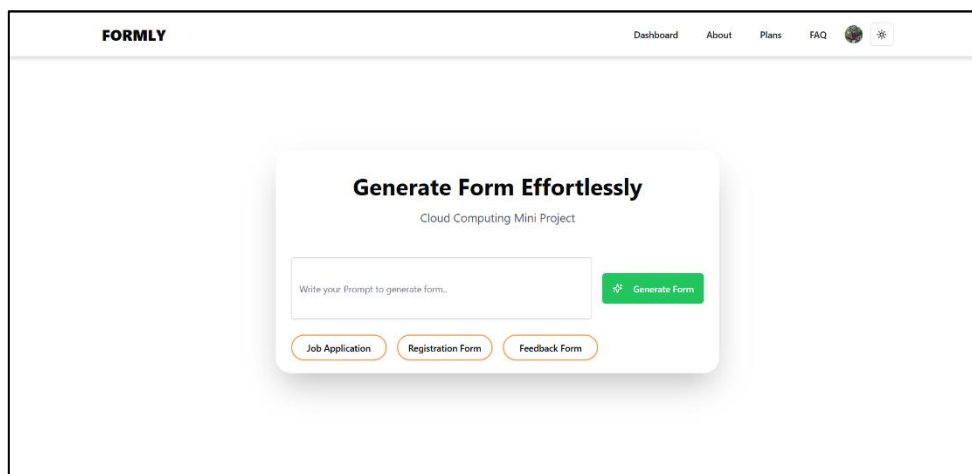


Figure 3: (Home Page)

Figure 3 (Home Page) displays FORMLY's main interface, designed for effortless form creation. Users can enter their form requirements in a simple input field and generate forms instantly. Pre-defined templates like Job Application, Registration, and Feedback Forms offer quick access. A clean Tailwind CSS-powered layout enhances usability, while the top navigation bar provides access to the Dashboard, Plans, and FAQ for a seamless experience.

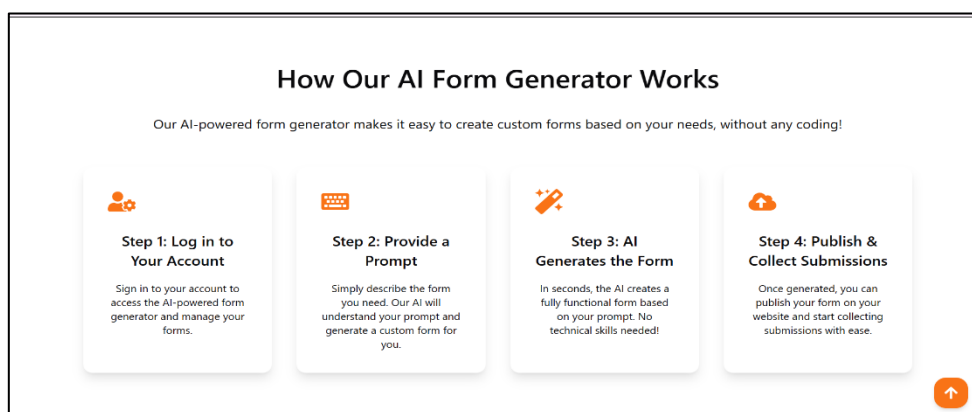


Figure 4: (About Section)

Figure 4 (About Section) outlines AI Form Generator's four-step process. It starts with logging in, followed by providing a prompt to define form requirements. AI generates form, and finally, users can publish it and collect submissions. Each step includes brief description and an icon for clarity, ensuring simple and user-friendly experience.

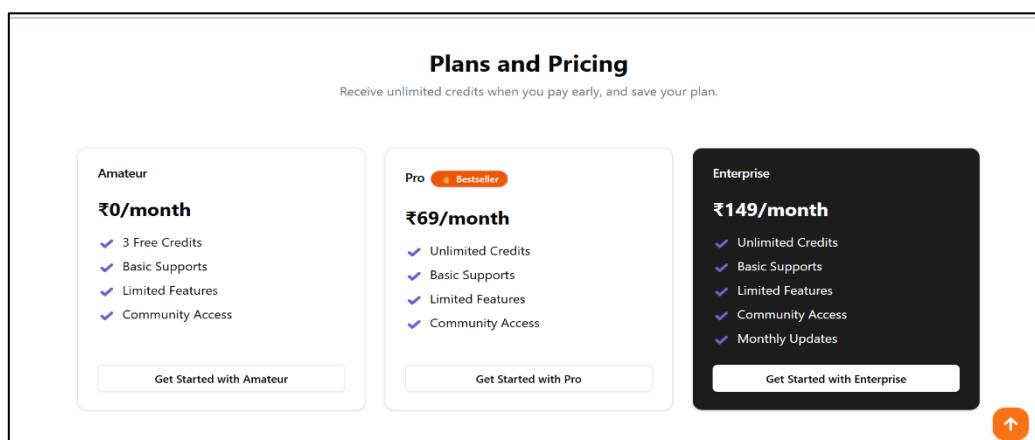


Figure 5: (Plans and Pricing)

Figure 5: (Plans and Pricing) segment showcases the "Plans and Pricing" section with three tiers: Amateur (₹0/month), Pro (₹69/month), and Enterprise (₹149/month). The Amateur plan offers basic features, while the Pro plan (bestseller) includes unlimited credits. The Enterprise plan adds monthly updates. A clean, organized layout highlights pricing, features, and "Get Started" buttons for easy comparison.

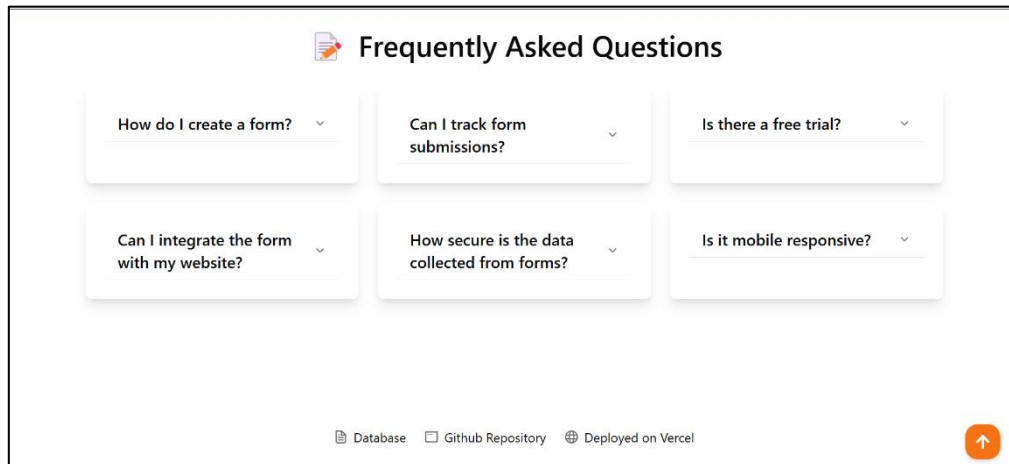


Figure 6: (FaQ)

Figure 6, FAQ section addresses common queries in a six-card grid layout. Each card poses a question with a downward arrow for expandable answers. A document icon accompanies the title, and at the bottom, small icons highlight the app's database, GitHub repository, and Vercel deployment. The design is clean and user-friendly for easy navigation.

Figure 7: (Publishing Page)

Figure 7 displays a "Job Application Form" in the FORMLY app's development view (localhost:3000). It includes input fields for applicant details like name, email, phone, experience, and education level. A "Publish" button suggests readiness for live use. The left sidebar features "My Forms" and "Analytics," indicating form management and data analysis tools.

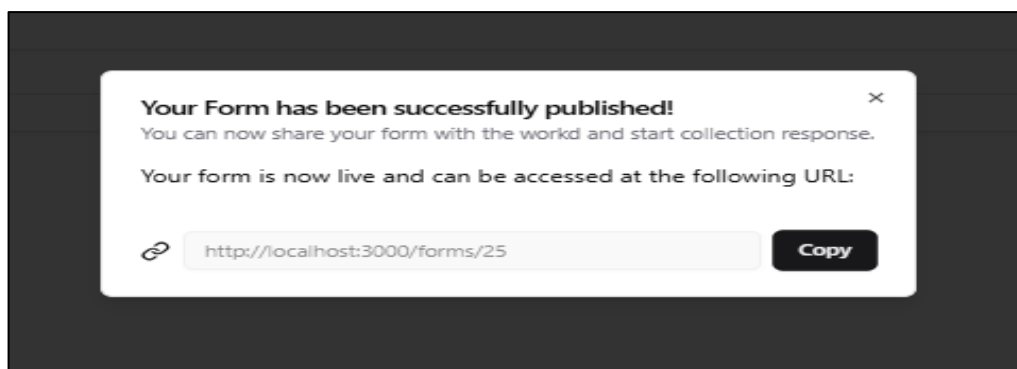


Figure 8: (Publish Notification)

Figure 8 shows a success notification dialog after publishing a form. It displays the message "Your Form has been successfully published!" along with the URL "<http://localhost:3000/forms/25>" for sharing the form. A "Copy" button is provided to easily copy the URL. The dialog has a dark theme with a white overlay, and an "x" icon in the top right corner to close the notification.

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/forms/25'. The page title is 'Job Application Form'. The form fields and their values are as follows:

| Field | Value |
|------------------------|--|
| First Name | Siddhesh |
| Last Name | Patil |
| Email | siddheshpatil54@gmail.com |
| Phone Number | 07715954900 |
| Desired Job Title | ABC |
| Years of Experience | 2 |
| Education Level | <input type="checkbox"/> High School <input type="checkbox"/> Associate's Degree <input checked="" type="checkbox"/> Bachelor's Degree <input type="checkbox"/> Master's Degree <input type="checkbox"/> Doctorate |
| Gender | <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other |
| Additional Information | XYZ |

A 'Submit' button is located at the bottom of the form.

Figure 9: (Submission Form)

Figure 9 shows a "Job Application Form" in a web browser at "localhost:3000/forms/25," indicating a development environment. The form is partially filled with example data, including names, email, phone number, job title, education level, and gender. The "Additional Information" field is being edited, and a "Submit" button is visible. The browser interface, including the address bar and extensions, confirms this is a local development version.

The screenshot shows the 'FORMLY' dashboard at 'localhost:3000/dashboard/forms'. The 'Generated Forms' section displays a card for the 'Job Application Form' with the subtitle 'Deploy your new project in one-click.' Below the card, there is a 'Submission - 1' link and two buttons: 'Edit' and 'Delete'. A '+ Create New Form' button is located in the top right corner. The sidebar on the left contains 'My Forms' and 'Analytics' links.

Figure 10: (Form List)

Form list section shows the "Generated Forms" section in the "FORMLY" dashboard at "localhost:3000/dashboard/forms." It displays a "Job Application Form" with options to edit, delete, and view submissions. A "+" button allows creating new forms, and the sidebar offers navigation for "My Forms" and "Analytics." The browser interface indicates a local environment.

The screenshot shows the 'FORMLY' dashboard at 'localhost:3000/dashboard/forms/25/submissions'. The 'SUBMISSIONS' section displays a card for 'RESPONSE : 1' with the following data:

| Field | Value |
|----------------|---------------------------|
| email | siddheshpatil54@gmail.com |
| lastName | Patil |
| firstName | Siddhesh |
| desiredJob | ABC |
| experience | 2 |
| phoneNumber | 07715954900 |
| additionalInfo | XYZ |

Figure 11: (Submission Responses)

Responses' interface shows the "Submissions" page for form 25 in "FORMLY," displaying a single response with key-value pairs for fields like email, name, job title, experience, and more. The sidebar offers navigation for "My Forms" and "Analytics," confirming this is a local development environment. Overall, the "FORMLY" project provides a powerful form creation and management solution, leveraging the capabilities of React, Tailwind CSS, Firebase, and Next.js. Its core features—such as easy form creation, submission tracking, user-friendly editing, and a responsive dashboard—address the complexities of form management and data collection. With a clean interface and robust backend, "FORMLY" offers a scalable and efficient solution for creating, managing, and analyzing forms in the cloud.

VI. CONCLUSION

This project presents "FORMLY," a dynamic form creation and management platform built using modern web development tools such as Tailwind CSS, React, Firebase, and Next.js. It efficiently handles form creation, submission tracking, and data management, offering a secure and scalable solution for users.

The frontend of the application is developed with React and styled using Tailwind CSS, ensuring a smooth, responsive, and intuitive user interface. The design simplifies form creation, submission, and management, making it user-friendly for both developers and non-technical users.

Firebase is utilized for the backend, enabling real-time data synchronization, robust authentication, and reliable form data storage. Next.js is used for server-side rendering, improving dynamic content delivery and reducing load times, while Vercel hosts the application to ensure scalability and high availability.

In summary, "FORMLY" integrates modern web development technologies to deliver a comprehensive, secure, and scalable form management system that meets the needs of users and developers alike. Future enhancements, such as improved analytics, better user customization, and additional form integrations, can further improve its capabilities in the ever-evolving digital landscape.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology (NIST), Special Publication 800-145, 2011.
- [2] J. W. Rittinghouse and J. F. Ransome, *Cloud Computing: Implementation, Management, and Security*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2016.
- [3] Google Cloud, "Overview of Cloud Storage Solutions," 2022. [Online]. Available: <https://cloud.google.com/storage>.
- [4] Dropbox Inc., "Understanding Dropbox: Secure File Storage and Sharing," 2022. [Online]. Available: <https://www.dropbox.com>.
- [5] Microsoft, "Features and Capabilities of OneDrive Cloud Storage," 2022. [Online]. Available: <https://www.microsoft.com/en-us/microsoft-365/onedrive/online-cloud-storage>.
- [6] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, Burlington, MA, USA: Morgan Kaufmann, 2013.
- [7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, et al., "An Analytical Perspective on Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, 2010.
- [8] Q. Zhang, L. Cheng, and R. Boutaba, "Current Trends and Research Challenges in Cloud Computing," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-18, 2010.
- [9] Amazon Web Services, "Comprehensive AWS Cloud Storage Solutions," 2022. [Online]. Available: <https://aws.amazon.com/storage>.
- [10] Cisco Systems, "Best Practices for Cloud Security," 2021. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/cloud.html>.