STM JOURNALS

Journal of
Web Engineering & Technology

ISSN: 2455-1880
Volume 12, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoWET

https://journals.stmjournals.com/jowet

Review JoWET

Unified Web Solutions: Video Conferencing, Summarization, Collaboration, and Tech Media

Avadhut S. Pawar[1], Kanchan A. Pujari[2,*], Prerana S. Dhile[3], Pranjali A. More[4]

Abstract

This paper presents the development and implementation of a versatile web-based application suite designed to enhance virtual collaboration and learning experiences. The suite consists of four main applications: uTalkApp, a video conferencing platform; uCoLabCanvas, a collaborative design tool; uTechMedia, a curated video content hub; and Synopsizer, an article summarization tool. Built using modern technologies such as Next.js, TypeScript, and Tailwind CSS, and hosted on Vercel, each application aims to address specific needs within the digital workspace. The project focuses on creating user-friendly interfaces, secure authentication mechanisms, real-time collaboration features, and efficient content management systems. Comprehensive user engagement metrics and feedback have been collected to evaluate the performance and usability of each application. This paper details the design, development process, and deployment strategies, highlighting the integration of advanced cloud deployment technologies to ensure scalability and reliability. The findings indicate significant improvements in user productivity and satisfaction, demonstrating the potential of these tools to facilitate effective remote collaboration and learning. The knowledge acquired from this project offers useful recommendations for creating cohesive web solutions in the future.

Keywords: Web-based application suite, virtual collaboration, next.js, TypeScript, Tailwind CSS, video conferencing, real-time collaboration, cloud deployment, productivity tools, article summarization, collaborative design tools

INTRODUCTION

The rapid shift towards digital solutions for communication, collaboration, and learning has underscored the importance of robust, scalable, and user-friendly web applications. Recent global events, such as the COVID-19 pandemic, have sped up the shift towards remote work and online learning, underscoring the necessity for all-encompassing digital tools. These tools must combine multiple features to boost productivity and ensure smooth interactions [1].

This project addresses this emerging need by developing a web-based application suite composed of four distinct, yet interconnected tools: uTalkApp, uCoLabCanvas, uTechMedia, and Synopsizer. Each application caters to specific requirements ranging from video conferencing and collaborative design to curated video content and article summarization. The integration of these functionalities into a single platform aims to provide users with an efficient and cohesive digital workspace that enhances both productivity and user engagement.

***Author for Correspondence**
Kanchan A. Pujari
E-mail: kapujari.skncoe@sinhgad.edu

[1,3,4]Student, Department of Electronics and Communication Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India
[2]Professor, Department of Electronics and Communication Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India

Received Date: July 03, 2024
Accepted Date: January 09, 2025
Published Date: January 18, 2025

Citation: Avadhut S. Pawar, Kanchan A. Pujari, Prerana S. Dhile, Pranjali A. More. Unified Web Solutions: Video Conferencing, Summarization, Collaboration, and Tech Media. Journal of Web Engineering & Technology. 2025; 12(1): 1–11p.

The choice of technologies, such as Next.js and TypeScript, ensures the creation of reliable, maintainable, and scalable web applications.

The Tailwind CSS was used to ensure that the user interfaces were both responsive and visually attractive. Hosting applications on Vercel leverages its powerful deployment capabilities and global edge network, ensuring optimal performance and accessibility. Additional tools, such as Clerk for authentication, Liveblocks for real-time collaboration, and Fabric.js for canvas operations, were integrated to enhance the overall functionality of the suite.

SYSTEM ARCHITECTURE

The system architecture of the web-based application suite comprises frontend and backend components, each serving specific functions to ensure seamless operation and user interaction [2]. The architecture was designed to be scalable, reliable, and maintainable, leveraging modern technologies and cloud deployment strategies, as shown in Figure 1.

Frontend Architecture

- *Landing Page:* The landing page serves as the entry point for users and provides a summary of the applications included in the suite. It was developed using *Next.js, Tailwind CSS,* and *Framer Motion* for enhanced animation.
- *uTalkApp:* This video conferencing application is built using *Next.js* and *TypeScript* for frontend logic. It employs *Clerk* for authentication and authorization features, whereas real-time functionality is powered by *GetStream. Tailwind CSS* ensures a responsive design and visual appeal.
- *uCoLabCanvas:* The collaborative design platform leverages *Next.js, TypeScript,* and *Fabric.js* for the canvas operations. *Liveblocks* enable real-time collaboration, and *Tailwind* CSS maintains a consistent design across the devices.
- *uTechMedia:* This curated video content hub was built using *React.js* and *Material-UI* for frontend components. *React Router DOM* handles navigation within the app, and *Axios* is used to make HTTP requests to fetch video content from external sources.
- *Synopsizer:* The article summarization tool was developed using *React.js* and *Redux Toolkit* for state management. *Material-UI* provides pre-designed components for the user interface, whereas *Axios* is used to fetch articles and generate summaries.

Backend Architecture

- *Authentication Server:* Responsible for managing user authentication and authorization, which is built using Node.js and Express.js. It utilizes Clerk for secure authentication mechanisms and interacts with the MongoDB database for user data storage.
- *Meeting App Server:* This server handles the backend logic for uTalkApp, including creating and managing meetings, recording functionalities, and participant management. It was developed using Node.js and Express.js, with MongoDB as the database for storing meeting data.
- *Media Streaming Server:* Facilitating real-time video streaming functionalities within the uTalkApp, this server was built using technologies such as WebRTC for peer-to-peer communication and Socket.IO for real-time event-based communication.
- *Drawing Board Server:* This server manages the collaborative drawing features of the uCoLabCanvas, handling real-time updates to the canvas and user interactions. It utilizes technologies such as Socket.IO for bidirectional communication, and MongoDB for storing drawing data.
- *API Server for Text Summarizer:* Responsible for processing requests for article summarization, which is built using Node.js and Express.js. It interacts with an external API or a machine learning model to generate article summaries based on user inputs.

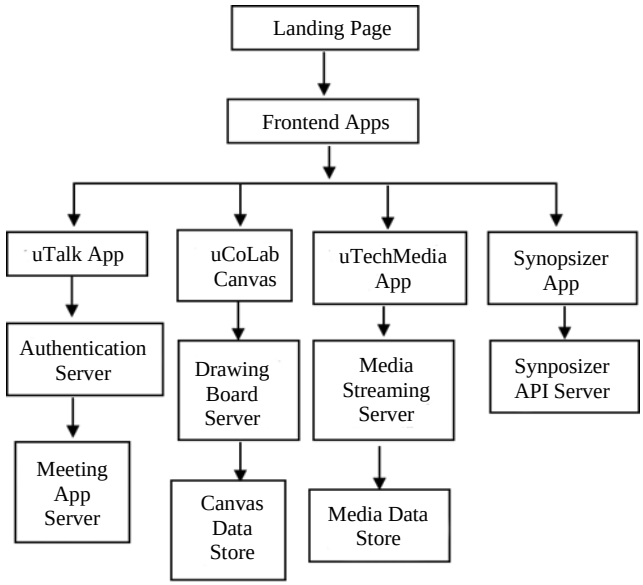

Figure 1. Web-based application architecture.

Frontend Architecture

- *Landing Page:* The landing page functions as the initial access point for users and provides an overview of the applications offered within the suite. It was built using Next.js, Tailwind CSS, and Framer Motion for enhanced animations.
- *uTalkApp:* This video conferencing application was developed using Next.js and TypeScript for frontend logic. It utilizes Clerk for authentication and authorization features, whereas real-time functionality is facilitated by GetStream. The Tailwind CSS ensures a responsive design and aesthetic appeal.
- *uCoLabCanvas:* The collaborative design platform is built using Next.js, TypeScript, and Fabric.js for canvas operations. Liveblocks enable real-time collaboration features, whereas the Tailwind CSS ensures a consistent design across devices.
- *uTechMedia:* This curated video content hub utilizes React.js and Material-UI for frontend components. React Router DOM facilitates navigation within the app, whereas Axios is used to make HTTP requests to fetch video content from external sources.
- *Synopsizer:* The article summarization tool was developed using React.js and the Redux Toolkit for state management. Material-UI provides pre-designed components for the user interface, and Axios is employed for fetching articles and generating summaries.

Backend Architecture

- *Authentication Server:* Responsible for managing user authentication and authorization, which is built using Node.js and Express.js. It utilizes Clerk for secure authentication mechanisms and interacts with the MongoDB database for user data storage.
- *Meeting App Server:* This server handles the backend logic for uTalkApp, including creating and managing meetings, recording functionalities, and participant management. It was developed using Node.js and Express.js, with MongoDB as the database for storing meeting data.
- *Media Streaming Server:* Facilitating real-time video streaming functionalities within the uTalkApp, this server was built using technologies such as WebRTC for peer-to-peer communication and Socket.IO for real-time event-based communication.
- *Drawing Board Server:* This server manages the collaborative drawing features of the uCoLabCanvas, handling real-time updates to the canvas and user interactions. It utilizes technologies such as Socket.IO for bidirectional communication, and MongoDB for storing drawing data.

- *API Server for Text Summarizer:* Responsible for processing requests for article summarization, which is built using Node.js and Express.js. It interacts with an external API or a machine learning model to generate article summaries based on user inputs.

PROBLEM STATEMENT

In today's fast-paced digital landscape, users often face the challenge of managing multiple applications to fulfill their communication, collaboration, and content consumption needs [3]. This fragmentation leads to inefficiencies, reduced productivity, and suboptimal user experience. Existing solutions typically operate in silos, which lack the integration needed to streamline workflows. The primary problem addressed by this project is the need for a unified platform that consolidates essential digital interaction tools and provides a seamless and efficient user experience.

By integrating video conferencing, collaborative drawing, media streaming, and text summarization into a single application, this project aimed to eliminate the friction of switching between multiple applications. It seeks to enhance productivity, improve accessibility, and provide a cohesive user experience that supports various digital interactions in both professional and personal contexts [4].

GOALS AND OBJECTIVES

The overarching goal of this project is to develop a comprehensive web application that integrates multiple digital interaction tools into a single user-friendly platform [5]. Specific objectives include:

- *Integration of Multiple Services:* Develop and integrate four distinct services—video conferencing, collaborative drawing, media streaming, and text summarization—into a unified application. This integration was designed to simplify user workflows and deliver a unified user experience.
- *User-friendly Interface:* Create a user interface that is both intuitive and responsive, facilitating smooth navigation and interaction. The design should be accessible to users at all technical skill levels to ensure wide usability.
- *Scalability and Reliability:* Ensure that the application can scale to accommodate a growing user base and maintain reliable performance under varying loads. This involves optimizing the architecture of the application and leveraging cloud services to efficiently handle peak traffic.
- *Security:* Apply strong authentication and data protection strategies to safeguard user information and ensure secure interaction. This includes encryption, secure communication protocols, and routine security audits to defend against potential threats.
- *Real-time Collaboration:* Facilitate real-time collaboration features in the uTalk app and uCoLab Canvas to enhance the interactive experience. This involves the implementation of technologies that support low-latency communication and data synchronization.
- *Cloud Deployment:* Utilizes Vercel for hosting and deploying the application, leveraging its capabilities for scalable and reliable cloud deployment.
- Vercel's serverless functions and automatic scaling ensure that the application remains efficient and accessible.

Achieving these objectives involves a combination of advanced web technologies, meticulous design, and rigorous testing to ensure that an application meets the needs of its users while maintaining high standards of performance, security, and usability [6].

STATEMENT OF SCOPE

This project encompasses the design, development, integration, and deployment of a web application that combines video conferencing, collaborative drawing, media streaming, and text summarization services [7]. The project includes:

- *Frontend Development:* Create responsive and intuitive user interfaces for each service module using Next.js, React, TypeScript, and Tailwind CSS. Frontend development emphasizes providing a seamless user experience through quick load times and fluid interaction.

- *Backend Development:* Development of robust server-side components to handle authentication, real-time communication, media streaming, and AI-driven text summarization. Backend development ensures that the application can efficiently handle multiple simultaneous users and high data loads.
- *Integration:* Guaranteeing smooth interaction between frontend and backend components to ensure fluid data flow and user experience. This includes the integration of various APIs and services to provide a unified platform.
- *Deployment:* Hosting the application on Vercel and leveraging its capabilities for scalable and reliable cloud deployment.
- The deployment process involves configuring continuous integration and deployment (CI/CD) pipelines to automate the release workflow.
- *Testing and Optimization:* Conduct thorough testing to ensure performance, security, and usability, and optimize the application for efficiency and scalability. This includes automated testing, performance monitoring, and collection of user feedback to detect and address issues.

The project scope is designed to cover all aspects of development, from initial design to final deployment, ensuring that the application is robust, reliable, and ready for production use. The integration of multiple services into a single platform requires careful planning and execution; however, the result is a comprehensive solution that meets the needs of modern digital users.

MAJOR CONSTRAINTS

The development of this comprehensive web application faces several constraints that must be addressed to ensure its success [8]:

- *Performance Optimization:* Ensuring smooth performance, particularly for real-time collaboration and media streaming, requires efficient resource management and optimization techniques. This involves enhancing the architecture of the application, reducing latency, and ensuring efficient resource utilization.
- *Security:* Protecting user data and maintaining secure communication channels is paramount, necessitating robust authentication and encryption mechanisms. Security measures should be embedded in every layer of the application, spanning from the frontend to the backend.
- *Scalability:* Designing an architecture to handle high user traffic and data loads efficiently, ensuring that the application can scale with growing demands. Scalability includes both horizontal and vertical scaling approaches to handle higher loads while maintaining performance.
- *Integration Complexity:* Seamless integration of multiple services into a single platform poses significant technical challenges, requiring careful planning and execution. Each service must be able to communicate and interact with others, without introducing bottlenecks or inconsistencies.
- *User Experience:* Delivering a cohesive and intuitive user experience across diverse functionalities and ensuring accessibility to various devices and screen sizes. The user interface must be designed to be aesthetically pleasing and functionally efficient, accommodating different user preferences and needs.

These constraints highlight the complexities involved in developing a multi-functional web application. Overcoming these challenges demands a mix of sophisticated technical solutions, thorough testing, and continuous optimization to ensure that the application achieves its objectives.

METHODOLOGIES OF PROBLEM-SOLVING AND EFFICIENCY ISSUES

To address these challenges and ensure efficiency, the project employed a range of modern web development methodologies and technologies.

- *Component-based Architecture:* Utilizing React's component-based architecture enhances code reusability and maintainability, thereby facilitating efficient development and testing. The components are modular, allowing for easier management and updates of the individual parts of the application without impacting the entire system.

- *Server-side Rendering:* Next.js enables server-side rendering, improves page load times, and enhances overall performance. Server-side rendering guarantees a quick initial page load, enhancing the user experience, particularly for those with slower internet connections.
- *Static Typing:* TypeScript introduces static typing into JavaScript, allowing errors to be detected at compile-time and boosting code quality and developer efficiency. Its type of system helps avoid many typical errors and enhances the maintainability and scalability of the codebase.
- *Responsive Design:* The Tailwind CSS guarantees a responsive design, offering a uniform user experience across different devices and screen sizes. The utility-first methodology enables quick prototyping and maintains consistent styling throughout the application.
- *Real-time Collaboration:* Libraries such as Liveblocks and GetStream enable real-time collaboration features in the uTalk app and uCoLab Canvas, ensuring smooth and interactive user experiences. These libraries provide the necessary infrastructure for low-latency and high-reliability communication.
- *Authentication and Security:* Clerk handles authentication and authorization, providing secure access control, and safeguarding data protection. Clerk's comprehensive authentication solutions simplify the implementation of secure login, registration, and user management.
- *Cloud Deployment:* Vercel provides a scalable and reliable cloud deployment platform that simplifies the deployment process and ensures its high availability. Vercel's serverless functions and automatic scaling features enable effective management of fluctuating loads.

These methodologies and technologies were chosen to address the specific challenges in developing a multi-functional web application. They provide a robust framework for building, deploying, and maintaining an application, ensuring that it meets the performance, security, and usability requirements.

RESOURCE REQUIREMENTS

Hardware

- Development and testing were conducted using standard development laptops with sufficient processing power and memory. These laptops are equipped with modern processors, ample RAM, and SSD storage to handle the demands of development and testing.
- Internet connectivity for accessing cloud services and deployment applications. Reliable Internet access is essential for continuous integration, deployment, and collaboration among team members.

Software

- *Frontend Technologies:* Next.js, React, TypeScript, Tailwind CSS. These technologies provide a powerful and flexible framework for developing responsive dynamic user interfaces.
- *Backend Technologies:* Node.js, Express.js, and associated libraries for managing real-time communication and data processing. These technologies support the development of scalable and efficient server-side devices.
- *Authentication:* Clerks for user authentication and authorization. Clerk simplifies the implementation of secure authentication mechanisms and reduces the burden on the developers.
- *Real-time Collaboration:* Liveblocks and GetStream enable real-time features. These libraries provide the necessary infrastructure for low-latency and high-reliability communication.
- *Deployment:* Vercel is used for hosting and deploying applications and offers both scalability and reliability. Its serverless functions and automatic scaling features ensure efficient handling of fluctuating loads.
- *Development Tools:* VSCode for coding, GitHub for version control, and various testing tools to ensure functionality and performance. These tools support an efficient development workflow, version control, and automated testing.

The combination of hardware and software resources provides a solid foundation for the development, testing, and deployment of web applications. The choice of technologies and tools was designed to support the project's performance, scalability, and usability goals.

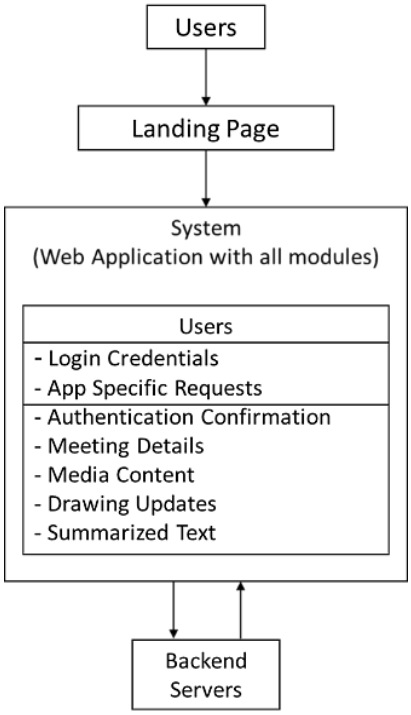

Figure 2. Context-level data flow diagram (DFD Level 0).

DATA FLOW DIAGRAM

Context-level Data Flow Diagram (DFD Level 0): A high-level diagram illustrating the system's interactions with external entities and the primary data flow within the system, as shown in Figure 2.

Entities

- *Users:* The primary external entity interacting with the system.
- *Authentication Server:* Responsible for authenticating users and managing access control.
- *Meeting App Server:* Hosts the video conferencing application.
- *Media Streaming Server:* Facilitates streaming of media content.
- *Drawing Board Server:* Manages collaborative drawing sessions.
- *Text Summarizer API Server:* Provides text summarization services.

Processes

- *User Interaction with Frontend:* Represents the interaction between users and the frontend applications.
- *Authentication:* Manages user authentication and access control.
- *Data Processing and Storage:* Handles the processing and storage of data within the system.
- *API Requests and Responses:* Facilitates communication between the frontend applications and backend servers.

DataStores

- *User Data Store:* Stores user-related information, including authentication credentials and profile data.
- *Meeting Data Store:* Stores information related to scheduled meetings and conference sessions.
- *Media Data Store:* Stores media content such as video recordings and streaming data.
- *Canvas Data Store:* Stores data related to collaborative drawing sessions.
- *Summarizer Data Store:* Stores text data and summarizes content generated by the text summarization service.

- This diagram offers a broad overview of the system architecture and highlights key components and their interactions.

Figure 2 presents a high-level view of the entire system, illustrating the interactions between the external entities and core components. It demonstrates how users engage with the system via frontend applications and how their requests are managed by backend servers [9]. The Authentication Server is responsible for user authentication and the generation of access tokens, which are subsequently used to access different services. The Meeting App Server, Media Streaming Server, Drawing Board Server, and Text Summarizer API Server handle specific functionalities of the system such as hosting meetings, streaming media content, managing collaborative drawing sessions, and generating text summaries.

Level 1 Data Flow Diagram Specification (DFDS) for User Interaction with Frontend

Figure 3 shows how users interact with the frontend applications of the system [10]. It shows the flow of data and user requests between the users and the frontend applications, including the Landing Page, uTalk app, uCoLab Canvas, uTechMedia App, and Synopsizer App. Users input data such as login credentials, meeting configurations, drawing inputs, and text content, which are then processed by the respective frontend applications.

ACTIVITY DIAGRAM

Figure 4 shows the *process of scheduling and conducting a meeting* in a video conferencing application. The following is an explanation of each step:

Explanation

- The process starts when the user chooses the "Schedule Meeting" option from the uTalk app's dashboard.
- The user enters the meeting details including the title, date, time, and participants.
- The system checks the provided information to confirm it is complete and accurate.
- Upon successful validation, the system saves meeting details, generates a unique meeting link, and stores it in a database.
- The system notifies participants about the scheduled meeting via email or in-app notifications.
- At the scheduled time, the participants joined the meeting using the provided link, and the meeting began.
- During the meeting, participants interacted using video conferencing, screen sharing, chats, etc.
- The meeting concludes when all participants exit or the host decides to end it.
- The process concludes, and the activity diagram ends.

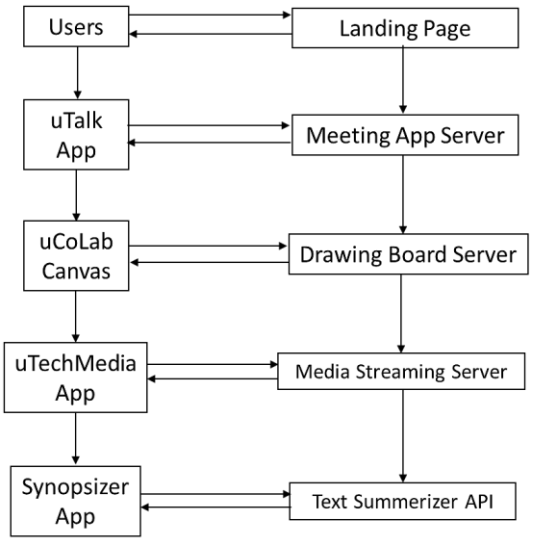

Figure 3. Level 1 Data Flow Diagram Specification (DFDS) for user interaction with frontend.

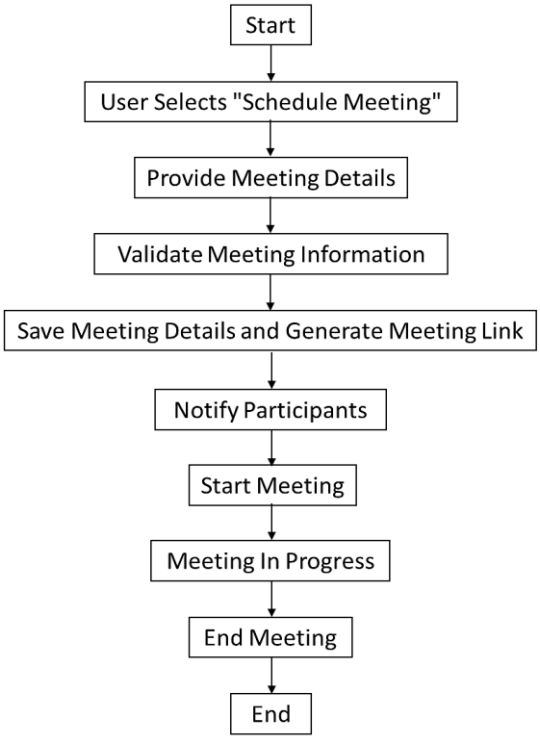

Figure 4. uTalk App activity diagram.

RESULTS AND DISCUSSION

The application was tested extensively to ensure functionality, performance, and user satisfaction. The testing process involves both automated tests and user feedback to identify and resolve these issues.

Performance Testing

- *Load Testing:* The application was subjected to load testing to assess its performance under different scenarios. This included simulating heavy user traffic and evaluating the response times, resource consumption, and overall system stability. The findings indicate that the application can manage a high volume of concurrent users without a notable performance decline.
- *Stress Testing:* Stress testing was conducted to determine the breaking point of the application and identify potential performance bottlenecks. The system was tested beyond its expected capacity to observe how it handled extreme conditions and ensure that it could recover completely from failures.

Security Testing

- *Penetration Testing:* Security experts performed penetration testing to uncover vulnerabilities and confirm that the application was safeguarded against common threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The results indicated that the application had robust security measures.
- *Authentication Testing:* Authentication mechanisms were tested to ensure that user data were protected and that unauthorized access was prevented. This included testing Clerk integration for secure login, registration, and user management.

Functional Testing

- *Unit Testing:* Each component was tested independently to verify that it operated correctly. This included testing frontend components, backend services, and integrations.
- *Integration Testing:* Integration testing was performed to confirm that the various modules of the application interacted smoothly with each other. This involved testing the interactions between

the frontend interfaces and backend servers, as well as the communication between different services.

DISCUSSION

The results of the testing process indicated that the application met its performance, security, and usability goals.

The successful integration of multiple services into a unified platform provides a smooth and effective user experience. The adoption of modern web technologies and best practices has ensured that the application is resilient, scalable, and secure.

The feedback from users has been positive, highlighting the ease of use, intuitive interface, and value of having multiple services integrated into one platform. Real-time collaboration features, in particular, have been well received, with users appreciating their ability to work together seamlessly.

The successful deployment of the application on Vercel highlighted the advantages of utilizing a contemporary cloud platform for hosting and scaling web applications. The serverless functions and automatic scaling provided by Vercel ensured that the application remained stable and available under varying load conditions.

The project has achieved its goal of creating a comprehensive web application that integrates video conferencing, collaborative drawing, media streaming, and text summarization. This application provides a valuable tool for users in various sectors to enhance productivity and user engagement.

CONCLUSION

This study presents the development of a comprehensive web application that integrates video conferencing, collaborative drawing, media streaming, and text summarization into a single platform. The application utilizes contemporary web technologies and best practices to deliver a smooth, effective, and secure user experience.

The successful integration of these services addresses the fragmentation of digital tools and provides streamlined solutions for users in various sectors. The architecture, methodologies, and testing results of the application demonstrate their robustness, scalability, and usability.

Future work will focus on further enhancing the application’s features, improving its performance, and expanding its capabilities to meet the evolving needs of users. Ongoing developments and user feedback will guide the future direction of the project, ensuring that it continues to provide valuable and innovative solutions for digital interaction.

By offering a unified platform for communication, collaboration, and content consumption, this project has the potential to revolutionize digital workflows and improve productivity across various domains. The integration of multiple services into a cohesive application represents a significant advancement in web application development and user experience.

Acknowledgment

We extend our heartfelt gratitude to Ms. K.A. Pujari for her invaluable guidance and support during this project. We also thank the faculty of the Department of Electronics and Telecommunication Engineering at Smt. Kashibai Navale College of Engineering, Pune for their insightful suggestions and encouragement. Special thanks go to Mr. P.S. Kokare and Ms. M.M. Sonkhaskar for their coordination and assistance. Lastly, we are grateful to Dr. S.K. Jagtap, Dr. A.V. Deshpande, Dr. K.R. Borole, and Mr. R.H. Borhade for their continuous support, and to our families and friends for their unwavering encouragement.

REFERENCES

1. Zhao Y, Luo C, Tang C, Chen D, Codella N, Zha ZJ. Streaming video model. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada. 2023. pp. 14602-12. DOI: 10.1109/CVPR52729.2023.01403.
2. Jartarghar HA, Salanke GR, Ar AK, Sharvani GS, Dalali S. React apps with server-side rendering: Next.js. J Telecommun Electron Comput Eng. 2022 Jan 22;14(4):25-9. DOI: https://doi.org/10.54554/jtec.2022.14.04.005.
3. Srivastava AK, Laxmi V, Singh P, Pratima K, Kirti V. React JS (open source JavaScript library). Int J Innov Res Technol. 2022;8:342-6.
4. Singh G, Javed M, Dhaliwal BK. Full stack web development: Vision challenges and future scope. Int Res J Eng Technol. 2022;9:3043-89.
5. Lazuardy MF, Anggraini D. Modern front end web architectures with react.js and next.js. Res J Adv Eng Sci. 2022;7:132-41.
6. Warkar A, Deshmukh V. Pathocare Website. J Emerg Technol Innov Res. 2024;11:124-8.
7. Falcão AX, Udupa JK, Samarasekera S, Sharma S, Hirsch BE, Lotufo RA. User-steered image segmentation paradigms: Live wire and live lane. Graph Models Image Process. 1998;60:233-60. DOI: 10.1006/gmip.1998.0475.
8. Miorandi D, Sicari S, De Pellegrini F, Chlamtac I. Internet of things: Vision, applications and research challenges. Ad Hoc Netw. 2012;10:1497-516. DOI: 10.1016/j.adhoc.2012.02.016.
9. Goh HA, Ho CK, Abas FS. Front-end deep learning web apps development and deployment: A review. Appl Intell. 2023;53:15923-45. DOI: 10.1007/s10489-022-04278-6. PubMed: 36466774.
10. Syafariani RF, Sitanggang AS, Maseleno A. Application of backend and fronted systems on go-baby application in Bandung City. Int J Recent Technol Eng. 2019;7:113-8.

