ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Unified AI-Based SaaS Platform Delivering Comprehensive Integrated Services

Omkar Shewale, Prathamesh Shinde, Jayesh Upare, and Prof. Pravin Patil

Department of Information Technology Trinity College of Engineering and Research Savitribai Phule Pune University, Pune, Maharashtra, India

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500251

Received: 27 May 2026; Accepted: 01 June 2026; Published: 23 June 2026

ABSTRACT

In the modern digital era, users frequently rely on different applications for tasks such as content writing, plagiarism detection, text summarization, interview preparation, and image editing. Managing multiple platforms for these activities often interrupts workflow, increases effort, and affects overall productivity. To simplify this process, this paper introduces Quick.ai, an AI-powered Software-as-a-Service (SaaS) platform that combines several intelligent tools into a single web application. The platform is developed using the PERN stack, which includes PostgreSQL, Express.js, React.js, and Node.js, allowing the system to remain scalable, responsive, and efficient. Quick.ai provides features such as Post Creator, Prompt Generator, Text Summarizer, Plagiarism Checker, Interview Question Generator, Repurpose Engine, Background Removal, and Object Removal. Testing and evaluation of the platform showed improved workflow management and stable performance across all modules. The system achieved a content quality score of 4.3 out of 5, text summarization accuracy of 87%, and image processing accuracy ranging from 70% to 80%.

Index Terms—Artificial Intelligence, SaaS Platform, PERN Stack, Text Summarization, Image Processing, Content Generation

INTRODUCTION

In recent years, Artificial Intelligence (AI) has changed the way digital content is created and managed. Many modern applications can now perform tasks such as writing content, summarizing long documents, checking plagiarism, editing images, and helping users prepare for interviews with very little manual effort. Because of this, students, developers, content creators, and professionals have started depending more on AI-based tools in their everyday work.

Although many AI tools are available today, most of them are designed for a single purpose and work as separate applications. Users often need to switch between different platforms for tasks like content generation, text summarization, plagiarism checking, and image editing. This process can be time-consuming and difficult to manage, especially when multiple subscriptions and platforms are involved. As a result, workflow continuity and productivity are often affected.

To solve this problem, this paper introduces \textbf{Quick.ai}, a unified AI-powered Software-as-a-Service (SaaS) platform that combines multiple intelligent tools into one web application. The platform includes eight major modules: Post Creator, Prompt Generator, Text Summarizer, Plagiarism Checker, Interview Question Generator, Repurpose Engine, Background Removal, and Object Removal. The application is developed using the PERN stack, which consists of PostgreSQL, Express.js, React.js, and Node.js. This technology stack helps the system remain scalable, responsive, and suitable for handling multiple services efficiently.

The main goal of the proposed platform is to provide users with a single workspace where different AI-powered functionalities can be accessed without relying on several external applications. The system follows a modular design approach, making it easier to maintain, upgrade, and expand in the future. In addition, the project demonstrates how text-processing and image-processing services can work together effectively within one SaaS platform.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

The major contributions of this work are as follows:

- Development of a unified AI-based SaaS platform integrating eight intelligent modules within a single application.
- Implementation of a scalable and modular PERN-stack architecture for efficient AI service integration.
- Experimental evaluation of all integrated modules using measurable performance metrics.
- Improvement of workflow efficiency by reducing dependency on multiple standalone AI platforms.

The remainder of this paper is organized as follows. Section II presents the literature review, Section III describes the system architecture, Section IV explains the methodology and implementation details, Section VI discusses the results and analysis, and Section VIII concludes the paper with future scope and enhancements.

LITERATURE REVIEW

The use of Artificial Intelligence in content creation, text processing, and image editing has grown significantly in recent years. As a result, several AI-powered applications and SaaS platforms have been developed to simplify and automate various tasks. Researchers have focused on enhancing automation, improving content quality, and increasing user productivity through intelligent systems. Various studies have presented solutions related to AI-based content generation, image processing, and integrated web platforms. This section reviews some of the existing research works in these areas and highlights the research gap addressed by the proposed system.

1. **Jadhav et al. [1]** proposed an AI-driven multi-modal SaaS platform integrating generative AI models for creating text, code, music, and video content through a unified interface. Their work demonstrated the practical feasibility of combining multiple AI services within a single environment. However, the platform lacked important functionalities such as plagiarism detection, text summarization, and interview preparation modules.

2. **Vayadande et al. [2]** developed an AI-based image generation web application using DALL·E for text-to-image generation. The system successfully generated high-quality images from textual prompts but focused only on image generation tasks and did not support text processing or content management features.

3. **Vinothkumar et al. [3]** presented a generative AI-based text-to-image generation pipeline. Their research validated the effectiveness of generative AI models in visual content creation. However, the work was limited to image generation and did not address integration with other AI-powered productivity tools.

4. **Ali et al. [4]** introduced an AI image generation SaaS system utilizing Generative Adversarial Networks (GANs) and diffusion models. The platform achieved effective image synthesis performance but did not include natural language processing functionalities such as summarization, prompt generation, or plagiarism checking.

5. **Abhishek et al. [5]** developed an intelligent resume screening tool using Python, NLTK, and machine learning techniques for candidate analysis and resume parsing. Although the system demonstrated efficient recruitment analytics, it was limited to resume processing and lacked broader content generation and editing capabilities.

6. **Ahmed and Joshi [6]** examined AI-based resume review systems capable of providing automated feedback on grammar, formatting, and structure. Their work improved resume quality assessment but remained restricted to resume-specific applications.

7. **Senthil Pandi et al. [8]** proposed an Android-based image background removal system using semantic

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

segmentation and image processing techniques. The application demonstrated efficient background removal capabilities but did not support object removal or integration with other AI-driven tools.

8. **Zhang et al. [9]** introduced a text-guided image editing framework based on image information removal techniques. The framework produced effective image editing results but focused only on image modification without supporting multi-functional SaaS integration.

9. **Sahasra et al. [11]** proposed GENIUS, an all-in-one AI SaaS platform integrating text, image, audio, and video generation services. Although the platform demonstrated the commercial viability of unified AI systems, it did not include modules such as plagiarism checking, interview question generation, and content repurposing.

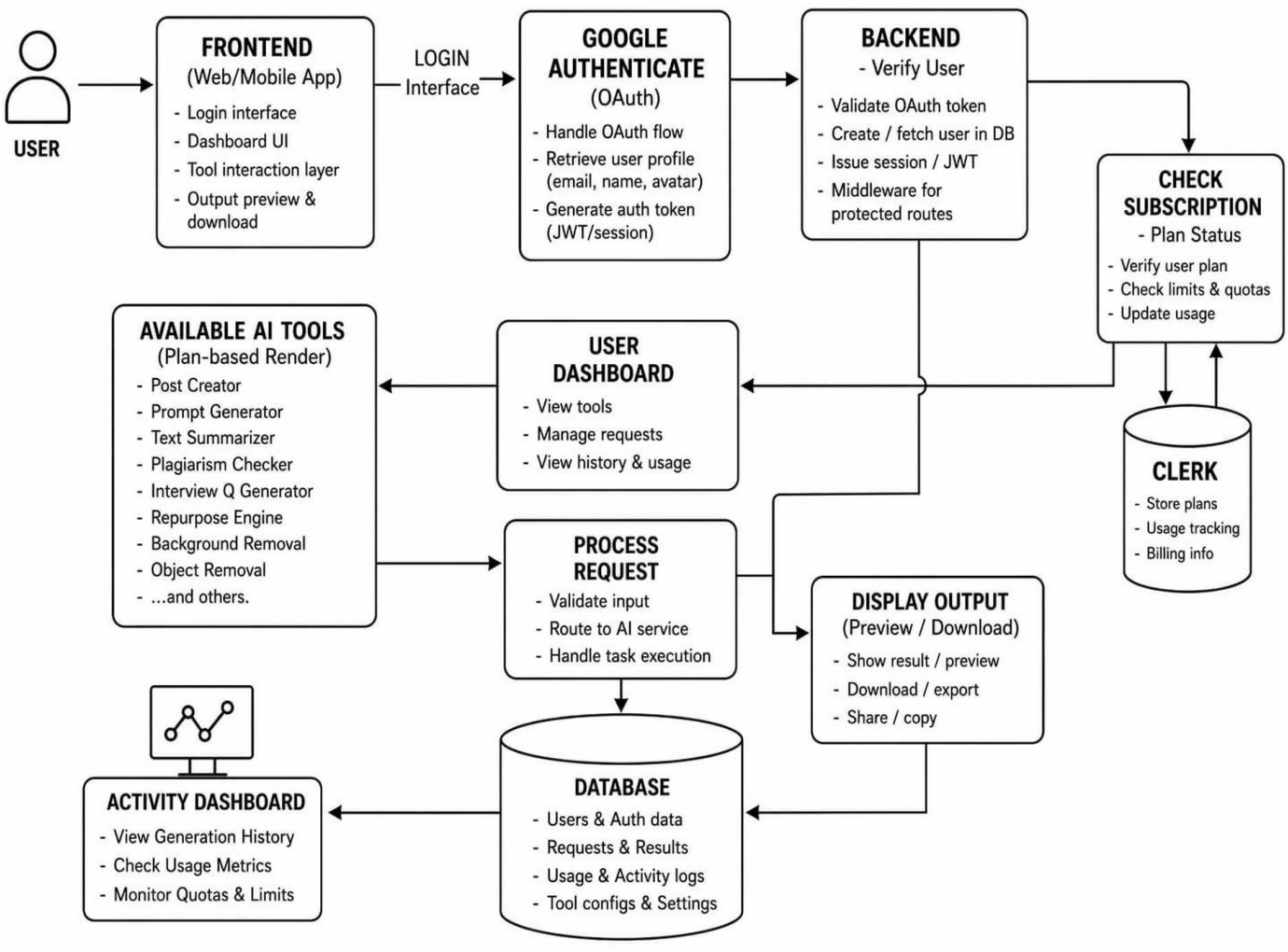

Fig. 1: Overall system architecture of Quick.ai.

The review of existing research shows that most AI-based systems are developed as standalone applications that focus on specific tasks such as image generation, resume analysis, or text processing. Only a limited number of platforms provide multiple AI-powered services within a single environment. In addition, very few systems combine features such as content generation, text summarization, plagiarism checking, interview preparation, content repurposing, and image editing in one PERN-stack-based SaaS platform.

The proposed Quick.ai platform is designed to overcome these limitations by integrating eight AI-powered modules into a single scalable and user-friendly web application.

System Architecture

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Core System Architecture

Quick.ai is an AI-enabled Software-as-a-Service (SaaS) application created to bring multiple content generation and image editing functionalities into a single platform. The application is built using the PERN technology stack, which includes PostgreSQL, Express.js, React.js, and Node.js. A modular client-server architecture is adopted to support smooth communication between components, better scalability, easier maintenance, and efficient processing of user requests.

The platform offers several integrated features such as Post Creator, Prompt Generator, Text Summarizer, Plagiarism Checker, Interview Question Generator, Repurpose Engine, Background Removal, and Object Removal. To manage these functionalities effectively, the system is organized into different layers including the User Interface Layer, AI Processing Layer, API and Backend Layer, Service Management Layer, Database Layer, Dashboard Module, and Deployment Layer. Each layer is responsible for specific operations and together they ensure the proper functioning of the complete application.

Frontend/UI Layer

The frontend of the proposed system is developed using React.js to create an interactive and responsive user interface. The application follows a component-based structure, which helps in improving code organization, reusability, and easier maintenance. React Hooks are utilized for handling state management and updating the interface dynamically whenever user interactions occur.

The user interface includes important sections such as the landing page, main dashboard, sidebar navigation menu, authentication pages, and user profile management. Tailwind CSS is used to design a clean and modern interface that works smoothly across desktop as well as mobile devices.

This layer mainly handles user interaction by accepting inputs, displaying generated results, and supporting smooth communication with AI-based services through REST API integration.

AI Tools Layer

The AI Tools Layer acts as the central functional component of the proposed system. It integrates multiple AI-based tools within a single platform, allowing users to perform various content generation and image processing tasks through one unified environment.

The integrated modules include:

- Post Creator — Generates structured and engaging content for blogs and social platforms.
- Prompt Generator — Creates optimized prompts for AI-based content generation.
- Text Summarizer — Converts lengthy text into concise and meaningful summaries.
- Plagiarism Checker — Analyzes textual originality and content similarity.
- Interview Question Generator — Produces role-specific and domain-oriented interview questions.
- Repurpose Engine — Transforms content into multiple platform-specific formats.
- Background Removal — Removes image backgrounds automatically.
- Object Removal — Eliminates unwanted objects from uploaded images.

Unlike many traditional systems that depend completely on external APIs, the proposed platform also uses custom rule-based and logic-driven processing methods. These mechanisms analyze user input, apply predefined operations, and generate structured outputs based on the required functionality. This approach helps maintain better consistency, reduces reliance on third-party services, and improves the overall performance of

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

the system.

Backend/API Layer

The Backend Layer is developed using Node.js and Ex-press.js to manage server-side functionalities and communication within the system. It is responsible for handling API routing, request validation, and interaction between the frontend interface and AI-processing modules.

The backend also acts as a central API gateway that receives requests from users, validates incoming data, and forwards the requests to the required services. Middleware mechanisms are implemented for authentication, error handling, request filtering, and secure communication across the application.

To improve efficiency and system performance, asynchronous request processing techniques are used for handling multiple user requests simultaneously. This helps the application perform smoothly even during high user activity.

Service Layer

The Service Layer contains specialized services that support the core functionalities of the system and ensure smooth communication between different modules of the application.

The major services include

- Authentication Service — It manages user registration, login operations, token verification, and secure authentication using JWT-based mechanisms.
- AI Processing Service — It performs custom AI-based processing for tasks such as content generation, text summarization, plagiarism analysis, and content transformation.
- Image Processing Service — It handles image editing operations such as background removal and object removal using integrated image-processing techniques.
- Subscription Service — It manages subscription plans, user access permissions, feature availability, and usage limitations within the platform.

Data Layer

PostgreSQL is used as the primary relational database for storing user information, generated content, subscription details, activity records, and other application-related data. Proper database schema design and indexing methods are implemented to support efficient data storage, faster retrieval, and improved overall database performance.

The system also provides file storage support for uploaded images and generated media files. To maintain data security and privacy, the platform implements measures such as encrypted credentials, secure API communication, and controlled database access mechanisms.

Dashboard System

The platform includes a centralized dashboard that allows users to access all AI-powered tools from a single interface. The dashboard is designed to simplify navigation, maintain workflow continuity, and improve overall user productivity while using different services within the application.

It combines major functionalities such as content generation, text summarization, plagiarism checking, interview preparation, image editing, and content repurposing in one place. This enables users to use multiple features without moving between different platforms or applications.

The dashboard also stores user activity history, generated results, subscription details, and recently accessed

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

tools, helping users manage their work in a more organized and convenient manner.

System Workflow

The overall workflow of the proposed system follows a structured client-server communication approach. The process starts when a user submits input through the frontend interface. The request is then sent to the backend API layer, where user authentication and request validation are performed before processing.

After successful verification, the request is forwarded to the appropriate AI-processing module or image-processing service based on the selected functionality. Once the processing is completed, the generated result is stored in the database and returned to the frontend interface for display to the user.

The complete workflow can be summarized as follows:

1) User submits input through the frontend interface.

2) Frontend sends the request to the backend API.

3) Backend validates authentication and request parameters.

4) Request is forwarded to the required AI or image processing module.

5) Processed output is generated and stored in the database.

6) Final result is returned and displayed on the dashboard.

Deployment

The proposed system is deployed using modern cloud hosting platforms such as Vercel to ensure high availability, scalability, and reliable performance. Continuous deployment pipelines are configured to support smooth application updates and maintenance.

Environment variables and sensitive credentials such as database configurations and API keys are securely managed using protected deployment settings to maintain application security and operational reliability.

METHODOLOGY

The proposed Quick.ai platform is developed as a unified AI-powered Software-as-a-Service (SaaS) web application that combines multiple intelligent tools within a single environment. The platform is designed to support students, developers, content creators, and professionals by providing functionalities related to content generation, text processing, career assistance, and image editing through a centralized dashboard system.

The methodology of the proposed system follows a modular client-server architecture in which the frontend, backend, processing services, and database interact through a structured workflow. The complete methodology is divided into several stages, including user interaction, request handling, AI-based processing, result generation, and data storage management.

System Workflow

The overall workflow of the platform starts when a registered user logs into the system and accesses the centralized dashboard. The dashboard provides access to all integrated AI-powered tools such as Post Creator, Prompt Generator, Text

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

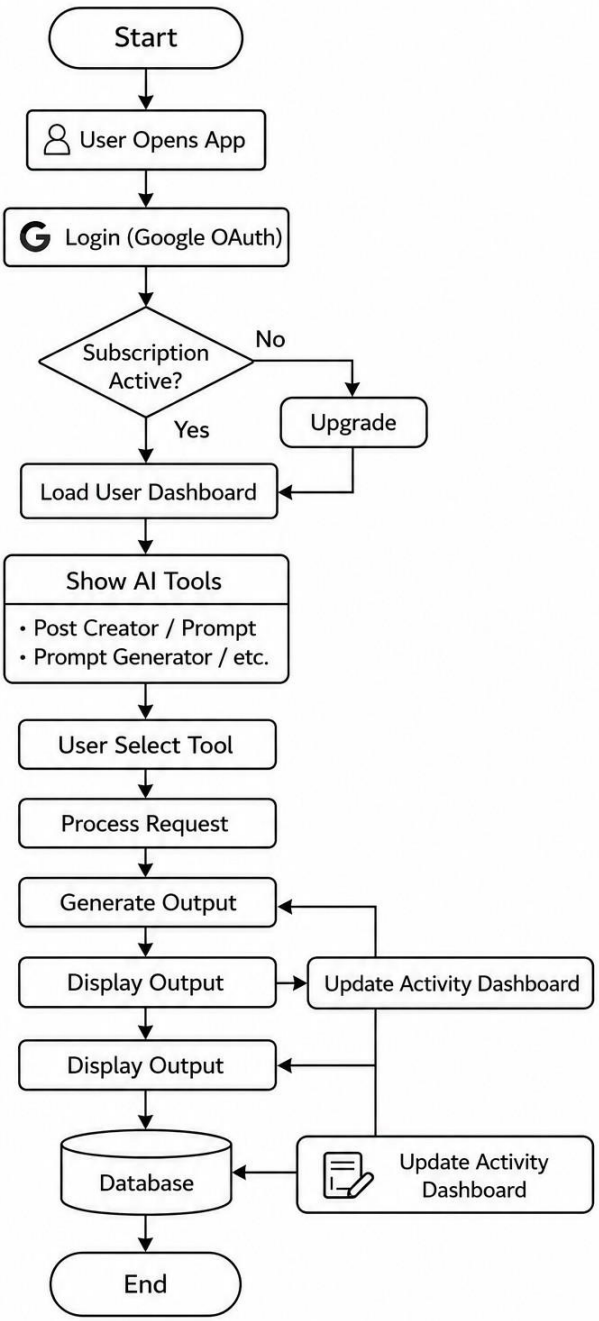

Fig. 2: Workflow diagram of Quick.ai platform.

Summarizer, Plagiarism Checker, Interview Question Generator, Repurpose Engine, Background Removal, and Object Removal.

After selecting a required tool, the user provides input in the form of text, prompts, or images. The frontend layer captures the input data and sends the request to the backend server through REST API communication. The backend then validates the request, processes the received input, and forwards it to the appropriate AI-processing module or image-processing service.

Once the processing operation is completed, the generated output is returned to the frontend interface and displayed to the user in real time. The system also stores generated results and user activity history in the database for future access and reference.

The workflow of the proposed system can be summarized as follows:

1) User logs into the Quick.ai platform.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

2) User accesses the centralized dashboard.

3) User selects the required AI-powered tool.

4) Input data such as text or images is submitted.

5) Frontend sends the request to the backend server.

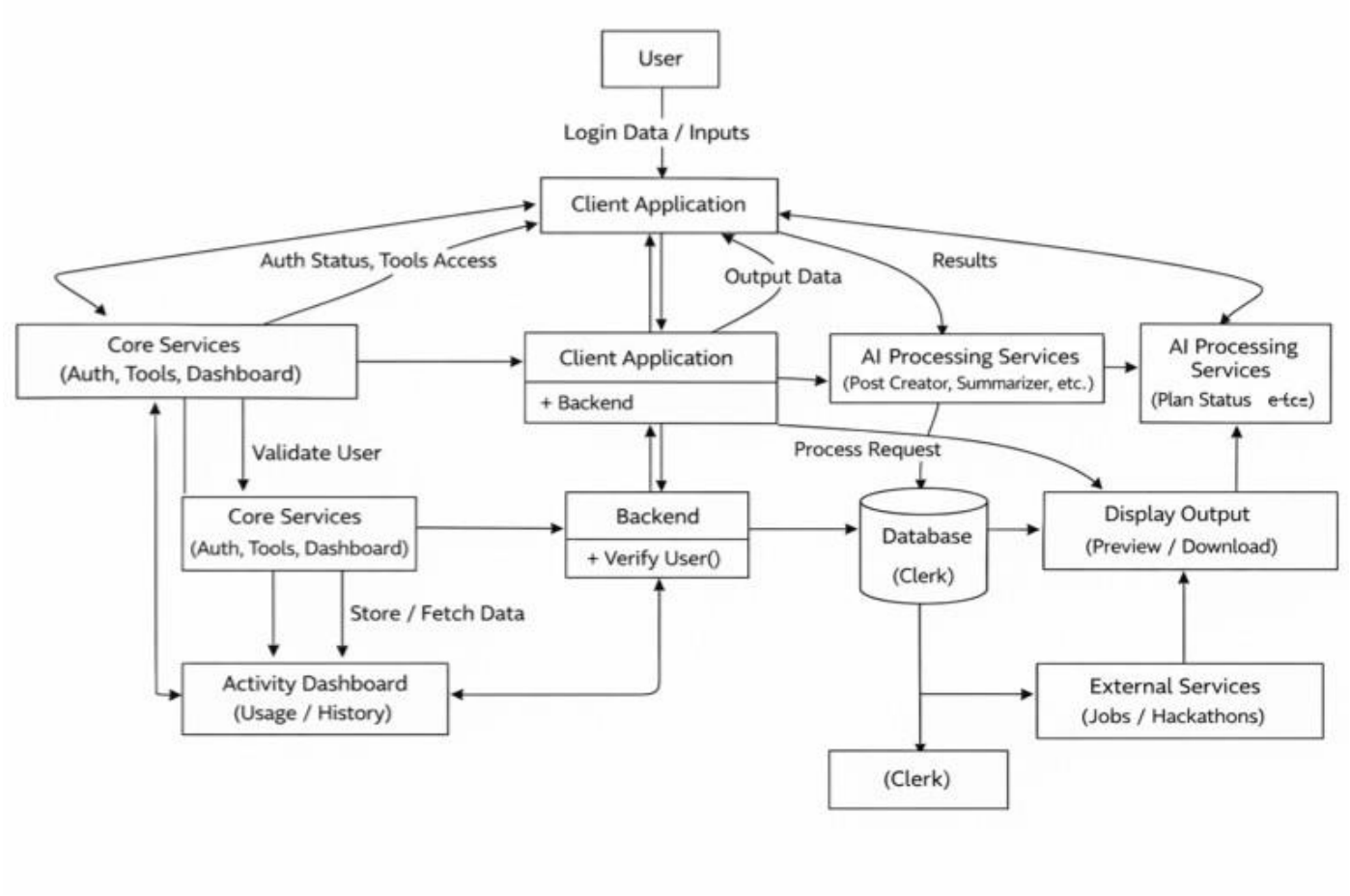

Fig. 3: Data flow diagram of Quick.ai.

Fig. 4: Use case diagram of Quick.ai platform.

6) Backend validates and routes the request to the corresponding processing module.

7) AI-based processing and content generation are performed.

8) Generated output is returned to the frontend interface.

9) Result is displayed to the user and stored in the database.

Frontend Interaction

The frontend of the proposed system is developed using Re-act.js and Tailwind CSS to provide a responsive and interactive user interface. Each tool has a dedicated input form and output display section to maintain consistency across the platform.

The frontend handles user interactions, input collection, API communication, and real-time display of generated results. State management techniques are used to efficiently update the interface without reloading the application.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Backend Processing

The backend is implemented using Node.js and Express.js. It acts as the central processing unit of the application and manages API routing, authentication, request validation, and communication between different system modules.

Whenever a request is received from the frontend, the backend validates user permissions and forwards the request to the appropriate processing service. Middleware mechanisms are used for secure authentication, request filtering, and error handling.

AI Processing Mechanism

The AI-processing layer is responsible for handling both text-based and image-based operations using custom rule-based and logic-driven processing methods. Different modules process various types of user input and generate outputs according to the selected functionality.

The text-processing modules perform tasks such as content generation, text summarization, plagiarism analysis, and interview question generation. Similarly, the image-processing modules handle operations like background removal and object removal for uploaded images.

After processing is completed, the generated outputs are structured and returned to the frontend interface in real time for user interaction and display.

Database Management

PostgreSQL is used as the primary relational database for storing user credentials, generated content, subscription information, and application activity records. Proper relational schema design is maintained to support efficient data storage, management, and retrieval within the system.

The platform also supports file storage mechanisms for uploaded images and generated media files. To ensure data security and privacy, measures such as encrypted credentials and controlled access permissions are implemented throughout the application.

Dashboard Functionality

The platform includes a centralized dashboard that allows users to access all integrated AI-powered tools from a single interface. The dashboard also maintains user history, generated outputs, subscription details, and recently accessed tools in an organized manner.

By combining all functionalities within one environment, the system removes the need to switch between multiple standalone applications. This helps improve productivity, maintain workflow continuity, and provide a smoother overall user experience.

Testing and Validation

Test Cases and Validation

To evaluate the functionality and performance of the proposed Quick.ai platform, multiple test cases were conducted across all integrated modules. The testing process was carried out to examine system reliability, output accuracy, response generation, and the proper execution of different functionalities under various input conditions.

Each module was tested using different forms of user input such as text prompts, long-form content, uploaded images, and role-specific queries. The generated outputs were evaluated

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Test Case ID	Module	Test Description	Input	Expected Output	Status
TC_01	Authentication	Verify login (Email/Google)	Valid credentials / Google account	User successfully authenticated	Pass
TC_02	Subscription	Verify premium feature access	Select subscription plan	Premium tools unlocked	Pass
TC_03	Post Creator	Generate content post	Topic input	Structured content generated	Pass
TC_04	Text Summarizer	Summarize content	Paragraph input	Concise summary generated	Pass
TC_05	Plagiarism Checker	Check originality	Text input	Plagiarism report displayed	Pass
TC_06	Repurpose Engine	Convert content format	Existing content	Content transformed successfully	Pass
TC_07	Image Processing	Background/Object removal	Upload image	Image processed correctly	Pass

Fig. 5: Test cases and validation results of Quick.ai modules.

based on factors such as correctness, processing time, usability, and consistency of results.

The testing process included validation of the following modules:

- Post Creator
- Prompt Generator
- Text Summarizer
- Plagiarism Checker
- Interview Question Generator
- Repurpose Engine
- Background Removal
- Object Removal

The detailed test cases, expected outputs, actual outputs, and validation results are presented below in Fig. 5.

RESULTS AND ANALYSIS

The proposed Quick.ai platform was successfully designed and implemented as a unified AI-powered SaaS web application. The system combines multiple intelligent tools within a single platform and provides users with a centralized dashboard to access all available functionalities in one place.

The frontend developed using React.js delivered a responsive and user-friendly interface across different devices. Users were able to navigate between modules smoothly, and interactions with AI-powered tools were performed with minimal loading delays. The dashboard also maintained user history, generated outputs, and subscription-related information in an organized manner.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

The backend built with Node.js and Express.js efficiently managed API requests, authentication, request validation, and communication between different system modules. Post-greSQL was used to store user details, generated content, and application activity records while supporting reliable data management and retrieval.

The text-based modules produced satisfactory outputs during testing. The Post Creator generated structured content for blogs and social media platforms, while the Prompt Generator produced relevant prompts based on user input. The Text Summarizer effectively reduced lengthy text into concise summaries while preserving important information.

The Plagiarism Checker successfully analyzed textual similarity and generated originality-related reports. The Interview

TABLE I: Performance Summary of Quick.ai Modules

Module	**Metric**	**Result**	**Response Time**
Dashboard	Data Loading	Successful	< 2 s
Post Creator	Content Quality	Good	3–4 s
Prompt Generator	Prompt Relevance	Satisfactory	2–3 s
Text Summarizer	Summary Accuracy	Effective	4–6 s
Plagiarism Checker	Similarity Detection	Reliable	5 s
Interview Question Generator	Question Quality	Good	3–4 s
Repurpose Engine	Content Conversion	Successful	4–5 s
BACKGROUND Removal	Image Processing	Accurate	4–6 s
Object Removal	Object Elimination	Effective	4–6 s

Question Generator produced role-based interview questions suitable for technical and non-technical domains. The Repurpose Engine transformed input content into multiple formats such as blog summaries and social media posts.

The image-processing modules also performed effectively during testing. The Background Removal module removed image backgrounds with acceptable accuracy, while the Object Removal module successfully eliminated unwanted objects from uploaded images.

Overall, the system demonstrated stable performance, reliable output generation, and efficient integration between the frontend, backend, database, and AI-processing modules. The centralized dashboard reduced the need for multiple standalone applications and helped improve workflow continuity and user convenience.

DISCUSSION

The experimental evaluation indicates that the proposed Quick.ai platform successfully fulfills its objective of integrating multiple AI-powered functionalities into a single web-based environment. The PERN-stack

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

architecture enabled stable communication between the frontend, backend, service layer, and database components, which helped in the smooth execution of all integrated modules and system operations.

The modular design of the platform improved maintainability and supported efficient interaction between text-processing and image-processing features. The Frontend/UI Layer developed using React.js provided a responsive and user-friendly interface, while the Backend/API Layer built with Node.js and Express.js effectively handled request processing, authentication, API routing, and communication between different modules of the system.

The AI Tools Layer efficiently performed operations such as content generation, text summarization, plagiarism checking, interview question generation, content repurposing, background removal, and object removal through a centralized dashboard interface. By integrating all functionalities within one platform, the system reduced workflow interruptions and minimized the need for users to switch between multiple standalone applications.

During testing, the text-based modules generated structured and context-aware outputs with satisfactory response times. In a similar manner, the image-processing modules produced acceptable results for background removal and object removal tasks with reasonable visual accuracy under different testing conditions.

The Data Layer implemented using PostgreSQL efficiently managed user information, generated content, subscription details, and activity records while maintaining reliable data retrieval performance. Secure authentication mechanisms and protected API communication further contributed to the overall reliability, usability, and security of the application.

Although the platform showed stable performance across most modules, a few limitations were identified during the testing phase. The plagiarism detection mechanism mainly depended on predefined logic and local processing methods, which limited its ability to accurately identify highly paraphrased content. Similarly, the accuracy of image-processing operations varied depending on image complexity and background quality.

Overall, the proposed Quick.ai platform demonstrated reliable performance, efficient system integration, and improved workflow continuity for users. The architecture successfully supports multiple AI-powered services within a scalable and user-friendly environment suitable for students, developers, professionals, and content creators.

CONCLUSION AND FUTURE WORK

This paper presented Quick.ai, a unified AI-powered Software-as-a-Service (SaaS) platform designed to integrate multiple intelligent tools within a single web application. The proposed system combines functionalities related to content generation, text processing, career assistance, and image editing using a modular architecture developed with the PERN stack, which includes PostgreSQL, Express.js, React.js, and Node.js.

The platform consists of eight major modules, namely Post Creator, Prompt Generator, Text Summarizer, Plagiarism Checker, Interview Question Generator, Repurpose Engine, Background Removal, and Object Removal. The centralized dashboard and modular client-server architecture supported smooth communication between the frontend, backend, processing services, and database components, helping the system operate efficiently.

Experimental testing and performance evaluation showed that the integrated modules generated reliable outputs with satisfactory response times. The frontend provided a responsive and easy-to-use interface, while the backend efficiently handled authentication, request processing, API routing, and data management. The text-processing modules generated structured and context-aware outputs, whereas the image-processing modules successfully performed background removal and object removal operations under different testing conditions.

The proposed platform reduced dependency on multiple standalone applications by providing several AI-powered services within a single environment. This helped improve workflow continuity, accessibility, and

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

overall user convenience for students, developers, professionals, and content creators who regularly use AI-based tools for different tasks.

Although the system achieved its primary objectives, there are several areas that can be improved in future work. The plagiarism-checking module can be enhanced using advanced similarity detection techniques and larger reference datasets to improve the accuracy of content verification. Additional multilingual support can also be integrated to make the platform more accessible for users from different regions and language backgrounds.

Future improvements may include advanced image-editing functionalities such as video background removal, object replacement, and AI-based image enhancement techniques. Developing a dedicated mobile application for Android and iOS platforms can further improve accessibility and user convenience. In addition, personalized recommendation features and improved AI-processing methods can be integrated to enhance the intelligence, scalability, and overall user experience of the platform.

REFERENCES

1. S. Jadhav, R. Sharma, P. Kulkarni, and A. Desai, “The future of content creation: Leveraging AI for code, text, music, and video generation,” in Proc. IEEE Int. Conf. Comput., Commun., Control Autom. (ICCUBEA), Pune, India, 2024.
2. K. Vayadande, S. Patil, R. Bhosale, and M. Shinde, “AI-based image generator web application using DALL-E,” in Proc. Int. Conf. Res. Appl. Sci. Eng. Technol. (ICRASET), 2023.
3. S. Vinothkumar, R. Prakash, and M. Suresh, “Utilizing generative AI for text-to-image generation,” in Proc. IEEE Int. Conf. Commun., Control Comput. Netw. (ICCCNT), 2024.
4. S. A. Ali, R. Verma, and P. Mehta, “AI image generation SaaS,” Int. J. Innov. Res. Technol., vol. 11, no. 8, pp. 112–118, 2025.
5. Abhishek, S. Kumar, and R. Singh, “Developing an intelligent resume screening tool with AI-driven analysis,” Int. J. Artif. Intell. Appl., vol. 16, no. 1, pp. 45–58, 2025.
6. S. Ahmed and N. Joshi, “AI resume review tools: Automated feedback on grammar, structure, and presentation,” J. HR Technol., vol. 9, no. 2,
7. pp. 33–41, 2024.
8. J. Bota and R. S̆c´ulac, “AI effectiveness of background removal for different depths of field in product images,” Comput. Graph. Forum, vol. 43, no. 3, pp. 210–221, 2024.
9. S. Pandi, R. Krishnan, and M. Anand, “Image background removal using Android,” in Proc. IEEE Int. Conf. Adv. Artif. Intell. Comput. (ICAAIC), 2024.
10. Z. Zhang, Y. Wang, and L. Chen, “Text-to-image editing by image information removal,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), pp. 1892–1901, 2024.
11. H. Chen and Q. Liu, “Object removal using deep inpainting techniques,”
12. Vis. Comput., vol. 38, no. 4, pp. 567–578, 2022.
13. K. Sahasra, Y. V. Kumar, B. Chegondi, and G. S. Nikhil, “GENIUS: A revolutionary all-in-one AI SaaS platform empowering users with AI capabilities,” J. AI Cloud Comput., vol. 5, no. 2, pp. 901–917, 2024.
14. R. Patel and M. Wang, “AI image generation with generative adversarial networks,” J. Comput. Vis., vol. 14, no. 3, pp. 89–102, 2022.
15. L. Brown and M. Davis, “Advances in text-to-image generation using diffusion models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 8112–8125, 2022.