iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78289

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

A WebApp AI Software as a Service Platform of Creative Tools Using PERN Stack

Prof. Mr. Shrikant A. Deshmukh[1], Vishwajit S. Deshmukh[2], Sameer S. Ahate[3], Shrivesh R. Chafekar[4], Vedant S. Bakshi[5]

Department of Computer Science & Engineering,Prof Ram Meghe College Of Engineering & Management, Badnera – Amravati
Sant Gadge Baba Amravati University, Amravati, Maharashtra, India

Abstract: ***Artificial intelligence technologies are increasingly integrated into digital platforms to support automated content generation, visual design, and analytical tasks. However, most AI-based tools are distributed across independent applications, which forces users to manage multiple platforms and interrupts workflow efficiency. The absence of unified AI environments also creates challenges related to accessibility, cost management, and system organization.***

This research introduces NexusAI, a web-based Artificial Intelligence Software as a Service (AI SaaS) system designed to combine multiple creative AI tools within a single platform. The system architecture is implemented using the PERN technology stack, which includes PostgreSQL, Express.js, React.js, and Node.js. These technologies provide a scalable infrastructure capable of handling high user interaction and structured data processing.The platform integrates multiple AI-driven modules such as text generation, image synthesis, and document evaluation tools. To maintain computational efficiency and prevent excessive API usage, the system employs a credit-based resource management mechanism that regulates user access to AI services.The proposed system demonstrates that consolidating multiple AI capabilities within a unified SaaS architecture can enhance usability, improve workflow continuity, and support sustainable computational resource management. The results indicate that centralized AI service aggregation can significantly reduce platform fragmentation while providing a scalable foundation for future AI-powered applications.

Keywords: Artificial Intelligence (AI), AI Software as a Service (AI SaaS), PERN Stack Architecture, Generative AI Systems, Web Application Development, Multi-Model AI Platform, Computational Resource Management, Scalable AI Platforms.

I. INTRODUCTION

Artificial intelligence has rapidly evolved into one of the most influential technologies in modern computing environments. With continuous advancements in machine learning algorithms, natural language processing, and generative models, AI systems are now capable of performing complex tasks such as automated content generation, visual media creation, and intelligent document analysis. These capabilities have significantly improved productivity in areas including digital marketing, education, content development, and software engineering. As organizations and individuals increasingly rely on AI-powered tools, the demand for accessible and efficient AI platforms has grown considerably.

Despite the growing popularity of AI technologies, most existing AI tools are distributed across separate platforms and applications. For example, some systems specialize in generating written content, while others focus on image generation, design assistance, or document evaluation. Users who require multiple AI functionalities must interact with different tools, each with its own interface, workflow, and subscription model. This fragmented ecosystem creates several challenges, including increased operational complexity, inconsistent user experiences, and reduced workflow efficiency. In many cases, switching between multiple platforms interrupts the creative process and requires users to repeatedly manage inputs, outputs, and configurations.

Another important limitation associated with existing AI services is the cost and management of computational resources. Advanced AI models require substantial processing power and often depend on external APIs or cloud-based machine learning infrastructures. Without appropriate resource management strategies, excessive usage of these services may lead to increased operational costs and system performance issues. As the number of users increases, maintaining a balanced distribution of computational resources becomes essential for ensuring platform sustainability and service reliability.

Software as a Service (SaaS) architecture provides an effective solution for delivering scalable digital services through web-based platforms. SaaS applications allow users to access software tools through internet-enabled interfaces without the need for complex installations or hardware requirements.

©IJRASET: All Rights are Reserved

This model offers several advantages, including centralized system management, simplified updates, improved accessibility, and efficient resource allocation. Integrating artificial intelligence capabilities within SaaS platforms enables developers to deliver powerful AI functionalities to users through a unified and user-friendly environment.

To address the challenges associated with fragmented AI tools and inefficient resource management, this research proposes NexusAI, a web-based Artificial Intelligence Software as a Service platform designed to integrate multiple creative AI utilities within a single application. The system aggregates diverse AI-driven tools such as text generation, image creation, document analysis, and other creative assistance services into a unified framework. By consolidating these capabilities into one platform, users can perform multiple AI-assisted tasks without the need to switch between different applications.

The proposed platform is developed using the PERN technology stack, which consists of PostgreSQL for database management, Express.js and Node.js for backend server operations, and React.js for building the user interface. This technology stack provides a scalable and efficient architecture capable of handling high volumes of user requests while maintaining structured data processing and system reliability. The modular design of the system also enables the integration of additional AI services in future updates.

In addition to integrating multiple AI tools, the system introduces a credit-based computational resource management mechanism. This mechanism regulates user access to AI services by assigning a limited number of credits for performing AI-related tasks. Each operation consumes a predefined number of credits depending on the computational resources required. This approach helps prevent excessive API consumption, ensures fair resource allocation among users, and maintains system stability.

The objective of this research is to design and implement a unified AI SaaS platform that improves workflow efficiency, enhances accessibility to AI tools, and provides sustainable resource management. By combining scalable web technologies with integrated AI services, the proposed system demonstrates how centralized AI platforms can simplify user interaction while supporting efficient computational resource utilization.

II. LITERATURE REVIEW

The integration of artificial intelligence into web-based systems has attracted significant attention in recent research. Several studies have focused on the development of generative AI models capable of producing natural language text, visual content, and automated analytical outputs. These models have contributed to the advancement of intelligent digital tools used in content creation, data processing, and decision support systems.

Software as a Service (SaaS) architecture has emerged as a widely adopted approach for delivering applications through cloud-based platforms. SaaS systems allow users to access software services through web interfaces without requiring local installation. This architecture provides advantages such as scalability, centralized management, and simplified software maintenance.

Research on AI integration frameworks has explored methods for combining multiple AI services within unified platforms. Multi-model architectures allow different AI models to operate within a single system, enabling platforms to support diverse functionalities such as text generation, image processing, and document analysis. Such integrated systems improve user experience by reducing the need to switch between separate applications.

Another important research area focuses on computational resource management in AI systems. AI services frequently rely on external APIs and high-performance computing resources, which can result in increased operational costs. To address this issue, several studies have proposed token-based or credit-based mechanisms that regulate user access to AI functionalities and prevent excessive resource consumption.

Although these studies highlight the advantages of AI integration and SaaS architectures, limited research has examined platforms that combine multi-model AI aggregation, scalable web technologies, and structured resource management mechanisms within a single framework. The NexusAI platform extends these concepts by integrating multiple AI services within a PERN-based SaaS architecture while implementing a credit-based resource management model.

Recent studies have also focused on the integration of artificial intelligence services within cloud-based environments to improve scalability and accessibility. Cloud computing platforms allow AI models to be deployed as on-demand services that can be accessed through web interfaces. This approach enables developers to deliver AI functionalities to a large number of users without requiring high computational resources on local devices. Researchers have highlighted that combining AI models with cloud infrastructure improves processing efficiency, reduces hardware dependency, and allows applications to scale dynamically based on user demand. Such architectures have become increasingly important for modern AI-driven platforms that need to support continuous user interaction and real-time data processing.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Another area of research emphasizes the importance of modular system design in AI-enabled web applications. Modular architectures allow different components of a system, such as user interfaces, backend services, and AI processing modules, to operate independently while remaining interconnected. This design approach improves system maintainability and allows developers to easily integrate new AI models or services without modifying the entire platform. Studies have shown that modular frameworks are particularly beneficial for AI SaaS systems, where different AI services may be updated or expanded over time. By adopting a modular architecture, platforms can achieve better flexibility, improved performance, and easier scalability when introducing new intelligent features.

III. PROPOSED SYSTEM

The proposed system introduces NexusAI, a web-based Artificial Intelligence Software as a Service (AI SaaS) platform that integrates multiple AI-powered creative tools within a unified environment. The primary objective of the system is to provide users with a centralized platform where various AI functionalities such as text generation, image creation, and document analysis can be accessed through a single interface. By consolidating multiple AI services into one application, the platform reduces workflow fragmentation and improves user productivity.

The system is implemented using the PERN technology stack, which consists of PostgreSQL, Express.js, React.js (or Next.js), and Node.js. This architecture enables efficient communication between the frontend interface, backend services, and database layer. The React-based frontend provides an interactive user interface where users can access AI tools, submit input prompts, and view generated outputs. The backend server, developed using Node.js and Express.js, handles API requests, processes user inputs, and manages communication with external AI services. PostgreSQL is used as the database system to store structured data such as user accounts, credit balances, and generated outputs.

The workflow of the proposed system begins with user authentication and account management. Users register on the platform and log in through a secure authentication mechanism. After successful login, users are directed to the application dashboard, where various AI tools are available. Each tool is designed to perform a specific task, such as generating articles, creating images, or analyzing documents. Users provide input prompts through the web interface, which are then transmitted to the backend server for processing.

To maintain efficient resource utilization, the system incorporates a credit-based computational resource management mechanism. Each user is assigned a certain number of credits that represent their ability to access AI services. When a user requests an AI operation, the system verifies whether sufficient credits are available. If the request is approved, the backend server forwards the input data to the appropriate AI service. After the AI model generates the output, the corresponding number of credits is deducted from the user's account. This mechanism ensures fair usage of computational resources while preventing excessive API consumption.

The platform also implements modular AI service integration, which allows different AI models to be connected through APIs. This modular approach enables the system to support multiple AI functionalities while maintaining a flexible architecture. New AI tools can be added in future updates without requiring major changes to the existing system infrastructure.

Overall, the proposed system provides a scalable and efficient AI SaaS platform that integrates multiple AI-driven services within a single web application. The combination of PERN stack architecture, modular AI integration, and credit-based resource management enables the platform to deliver intelligent services while maintaining system stability and cost efficiency.

IV. PROPOSEDARCHITECTURE

The proposed system architecture is designed to support a scalable and efficient AI Software as a Service platform capable of integrating multiple artificial intelligence tools within a unified web-based environment. The architecture follows a layered structure that separates the user interface, backend processing, artificial intelligence services, and database management components. This layered approach improves system organization, enhances scalability, and simplifies future integration of additional AI modules.

The User Interface Layer represents the interaction point between the user and the platform. This layer is developed using modern web technologies such as React or Next.js and provides a dashboard where users can access various AI-powered tools. Through this interface, users can submit prompts, upload input data, and view the results generated by the AI services. The interface is designed to provide a simple and intuitive user experience while ensuring smooth interaction with the backend system.

The Backend Processing Layer is implemented using Node.js and Express.js. This layer acts as the central controller of the system and manages communication between the frontend interface, AI services, and the database.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

When a user submits a request, the backend server processes the input, validates user credentials, checks available computational credits, and forwards the request to the appropriate AI service. The backend also manages API routing, request validation, and error handling to ensure reliable system operation.

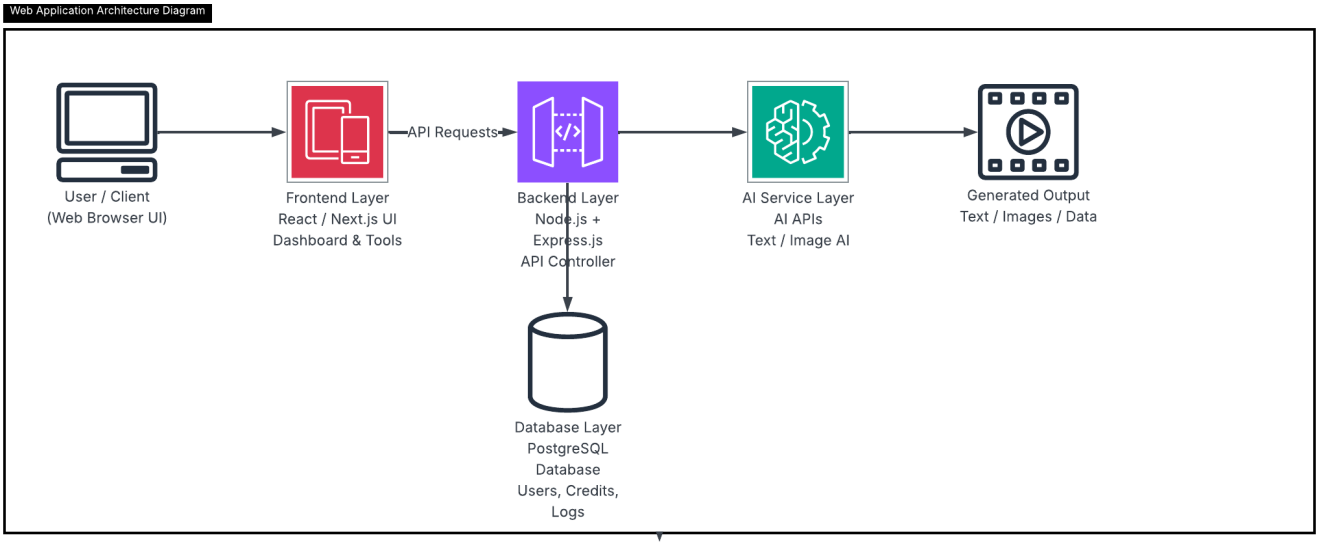


Fig1- Architecture

The AI Service Integration Layer connects the application with external artificial intelligence services. These services perform tasks such as text generation, image synthesis, and document analysis. By using API-based integration, the system can access advanced AI models without requiring complex local computation. This modular integration allows new AI services to be added to the platform in the future without modifying the overall system structure.

The Database Layer is responsible for storing and managing structured system data. PostgreSQL is used as the primary database management system because of its reliability and strong support for relational data structures. The database stores information related to user accounts, authentication data, credit balances, and records of AI-generated outputs. Maintaining structured records allows the system to track user activities and manage resource usage effectively.

In addition to these components, the platform implements a credit-based resource management mechanism that controls the usage of AI services. Each AI operation consumes a certain number of credits depending on the complexity of the task. This mechanism ensures fair distribution of computational resources among users while preventing excessive API consumption.

Overall, the proposed architecture enables efficient integration of AI services within a scalable SaaS platform. By combining modern web technologies, modular AI integration, and structured database management, the system provides a flexible environment capable of supporting multiple AI-powered applications.

V. METHODOLOGY

The methodology of the proposed system explains the process used to design, develop, and implement the AI Software as a Service platform. The system integrates web technologies, artificial intelligence services, and database management to provide users with multiple AI-powered creative tools through a unified web application. The methodology focuses on system development stages, workflow execution, and resource management mechanisms that ensure efficient operation of the platform.

1) System Development Approach

The development of the NexusAI platform follows a modular and layered architecture to ensure scalability and maintainability. The system is built using the PERN stack, which consists of PostgreSQL, Express.js, React.js, and Node.js. This technology stack provides a reliable framework for developing modern web applications capable of handling large numbers of user interactions.The frontend component of the system is developed using React.js or Next.js, which provides a responsive and dynamic user interface. The backend services are implemented using Node.js with the Express.js framework, which manages application logic and API communication. PostgreSQL is used as the database system to maintain structured records of user accounts, system activities, and AI-generated outputs.

This modular development approach allows different components of the system to operate independently while maintaining efficient communication between layers.

2) User Authentication and Account Management

The first stage of the system operation involves user authentication and account management. Users are required to register on the platform by providing necessary information such as username, email address, and password. Once the registration process is completed, users can log into the system through a secure authentication mechanism.Authentication ensures that only authorized users can access the AI tools available on the platform. After successful login, the system creates a secure session that allows users to interact with the application dashboard. This process protects the platform from unauthorized access and ensures that each user operates within their assigned account environment.

3) AI Tool Selection and User Interaction

After authentication, users are directed to the main dashboard of the application where multiple AI tools are available. These tools provide functionalities such as text generation, image creation, content assistance, and other AI-driven creative operations.Users interact with the platform by entering prompts or input data into the system interface. The user interface collects these inputs and sends them to the backend server through API requests. The system then processes the input data and determines the appropriate AI service required to perform the requested task.The interactive dashboard allows users to easily navigate between different AI tools without switching to external platforms.

4) Backend Processing and Request Handling

The backend server plays a central role in the system workflow. When the server receives a request from the frontend interface, it performs several operations before sending the request to an AI service.First, the server validates the user session and verifies that the request is generated by an authenticated user. Next, the system checks the user's available computational credits through the credit management module. This verification ensures that the user has sufficient credits to perform the requested AI operation.Once these validations are completed, the backend server prepares the input data and forwards it to the relevant AI service through API communication. The backend also manages error handling, request routing, and system logging to maintain reliable system operation.

5) AI Service Integration

The platform integrates multiple artificial intelligence services using external APIs. These AI services are responsible for performing tasks such as generating text content, producing images, or analyzing documents based on user inputs.When a request is received from the backend server, the AI service processes the input using machine learning models and returns the generated output. The system then forwards the results to the user interface where they are displayed to the user.This modular AI integration approach allows the system to support multiple AI functionalities without requiring complex local model deployment. Additionally, new AI services can be integrated into the system in future updates.

6) Credit-Based Resource Management

To ensure efficient utilization of computational resources, the platform implements a **credit-based usage control mechanism**. Each AI operation requires a certain number of credits depending on the complexity of the requested task.Before processing any request, the system checks whether the user has sufficient credits available in their account. If the user has enough credits, the system allows the request to proceed and deducts the appropriate number of credits after the operation is completed.This mechanism helps prevent excessive API usage and ensures fair distribution of computational resources among users. It also helps maintain the long-term sustainability of the platform by controlling operational costs associated with AI services.

7) Data Storage and Result Management

All system data is stored within the PostgreSQL database. The database maintains structured records including user information, authentication data, credit balances, and AI-generated outputs.Storing generated outputs allows users to access previously generated results and maintain a history of their activities. The database system also helps administrators monitor system performance and manage user accounts effectively.

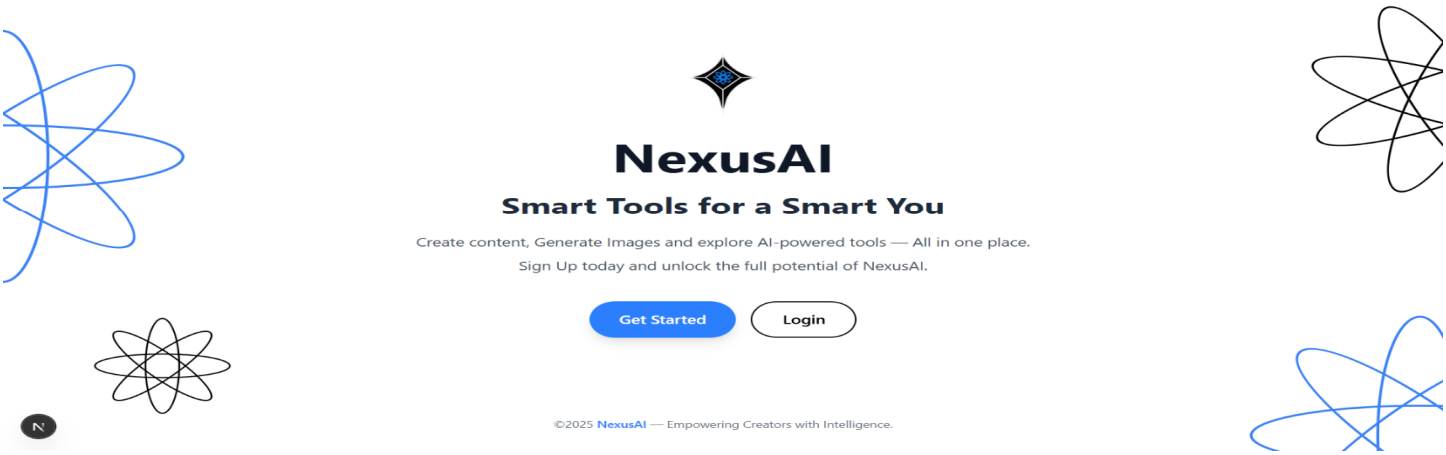
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

8) System Workflow Integration

The complete workflow of the NexusAI platform begins with user authentication, followed by AI tool selection, backend request processing, AI model execution, and result delivery. Each component of the system works together to provide a smooth and efficient user experience.By combining modern web technologies with AI service integration and structured resource management, the methodology ensures that the platform operates in a reliable and scalable manner. The proposed system demonstrates how artificial intelligence capabilities can be delivered through a centralized SaaS architecture while maintaining efficient control of computational resources.

9) System Monitoring and Performance Management

To maintain stable operation of the platform, the system includes a monitoring and performance management mechanism. This component tracks system activity such as user requests, AI service usage, credit consumption, and response times. Monitoring helps identify potential performance issues and ensures that the platform continues to operate efficiently as the number of users increases.



1- Landing Page Interface

Figure 2.The landing page of the NexusAI platform acts as the entry point for users who visit the application. It provides an overview of the system and highlights the main purpose of the platform, which is to deliver multiple artificial intelligence tools through a single web-based environment. The interface is designed with a simple layout that introduces the platform name, its tagline, and the primary features offered by the system.The page includes call-to-action buttons such as Get Started and Login, which allow users to either create a new account or access their existing profile. The design emphasizes clarity and accessibility, ensuring that users can easily understand the purpose of the application before interacting with its services. This landing page plays an important role in guiding new users and encouraging them to explore the available AI-powered tools.

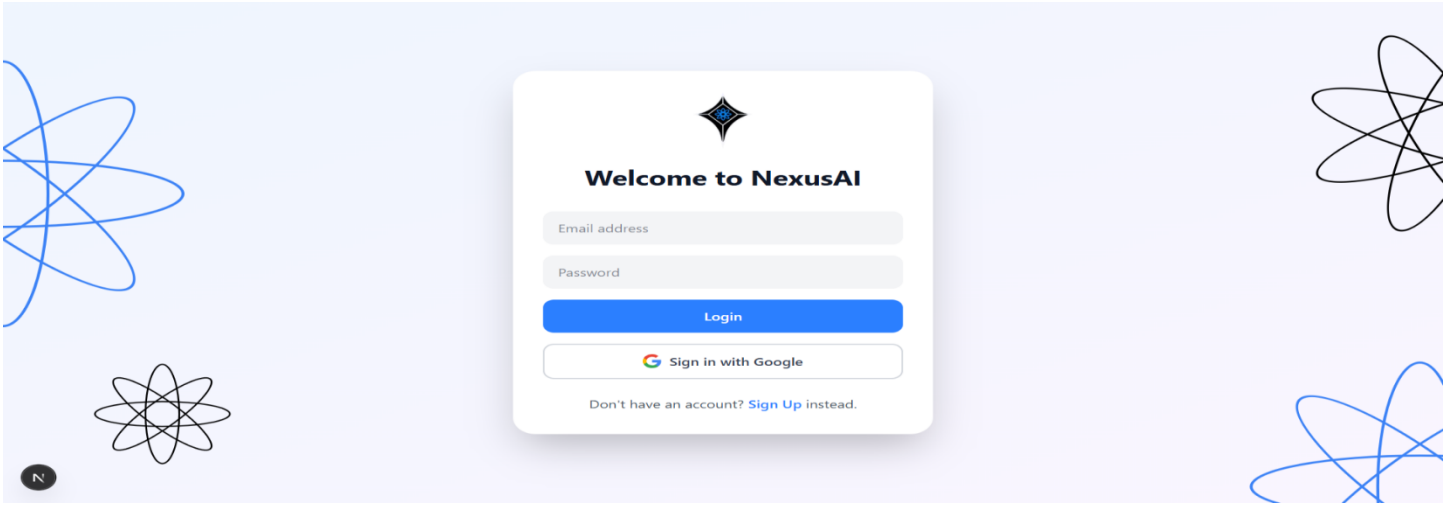


Fig2.2-User Authentication Interface

Figure 2.The login page is responsible for authenticating users before granting access to the application dashboard. It contains input fields for email and password, allowing registered users to securely log into the system. In addition to the standard login method, the platform also supports **Google authentication**, which simplifies the login process and improves user convenience.Authentication ensures that only authorized users can access the AI tools provided by the platform. Once the user credentials are verified, the system creates a secure session and redirects the user to the main dashboard. This mechanism helps protect the application from unauthorized access while maintaining secure user management.

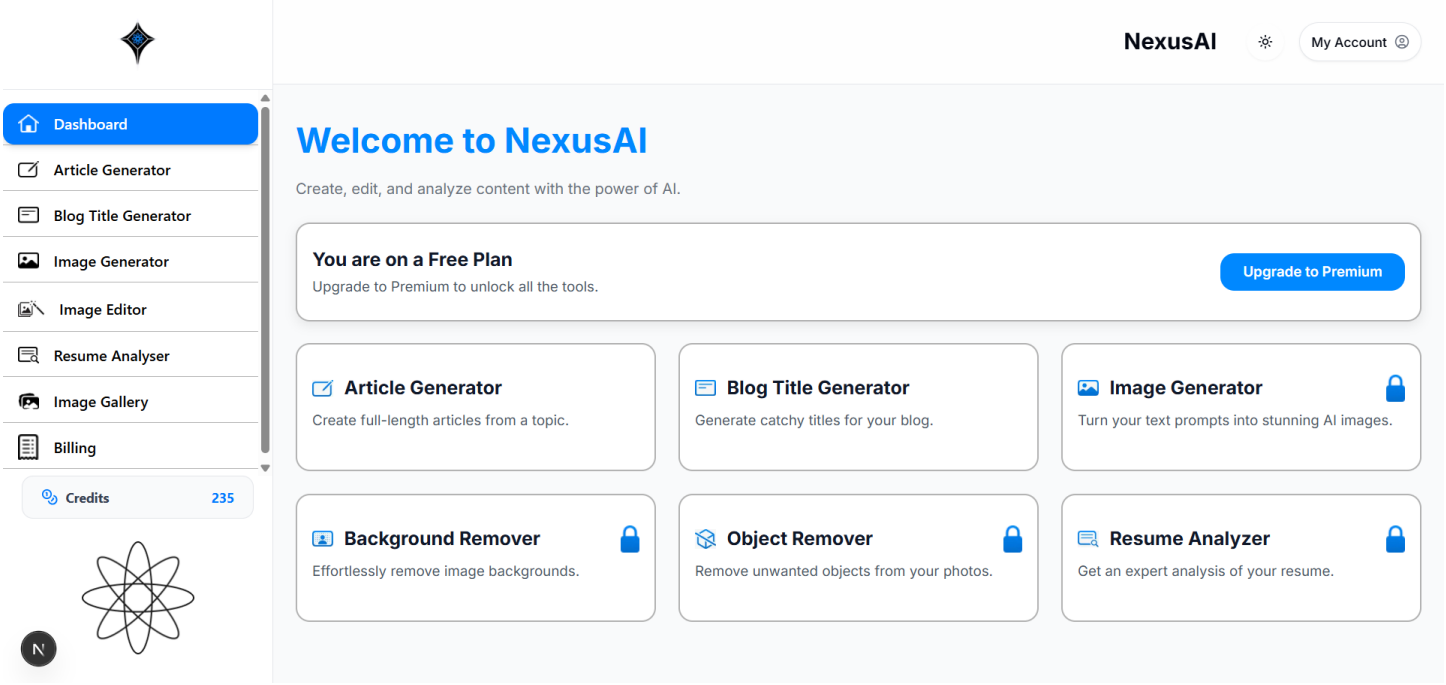


Fig2.3-Dashboard Interface

Figure 2.3 After successful login, users are redirected to the main dashboard of the NexusAI platform. The dashboard serves as the central workspace where users can access different AI tools available within the system. The interface displays various modules such as Article Generator, Blog Title Generator, Image Generator, Image Editor, Resume Analyzer, and Image Gallery.Each module represents a specific AI functionality designed to assist users in performing creative or analytical tasks. The dashboard also displays the user's current subscription status and available credits. By presenting all tools in a unified interface, the dashboard reduces the need for users to navigate multiple platforms and improves overall workflow efficiency.

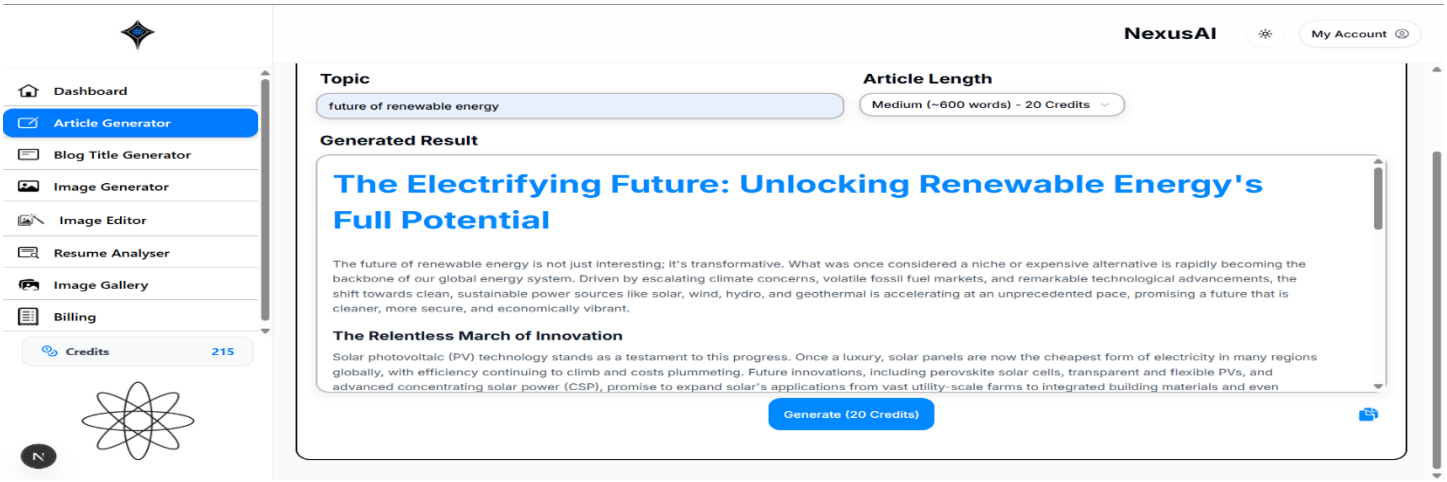


Fig2.4-Article Generator Tool

Fig 2.4 The article generator module allows users to automatically create written content based on a specific topic. Users enter a topic or prompt into the input field and select the desired article length. After submitting the request, the system processes the prompt using an integrated artificial intelligence model.The AI model analyzes the input topic and generates structured written content related to the requested subject. The generated result is displayed within the interface, allowing users to review, copy, or modify the content as required. This feature is particularly useful for content creators, bloggers, and researchers who need assistance in generating textual material quickly.

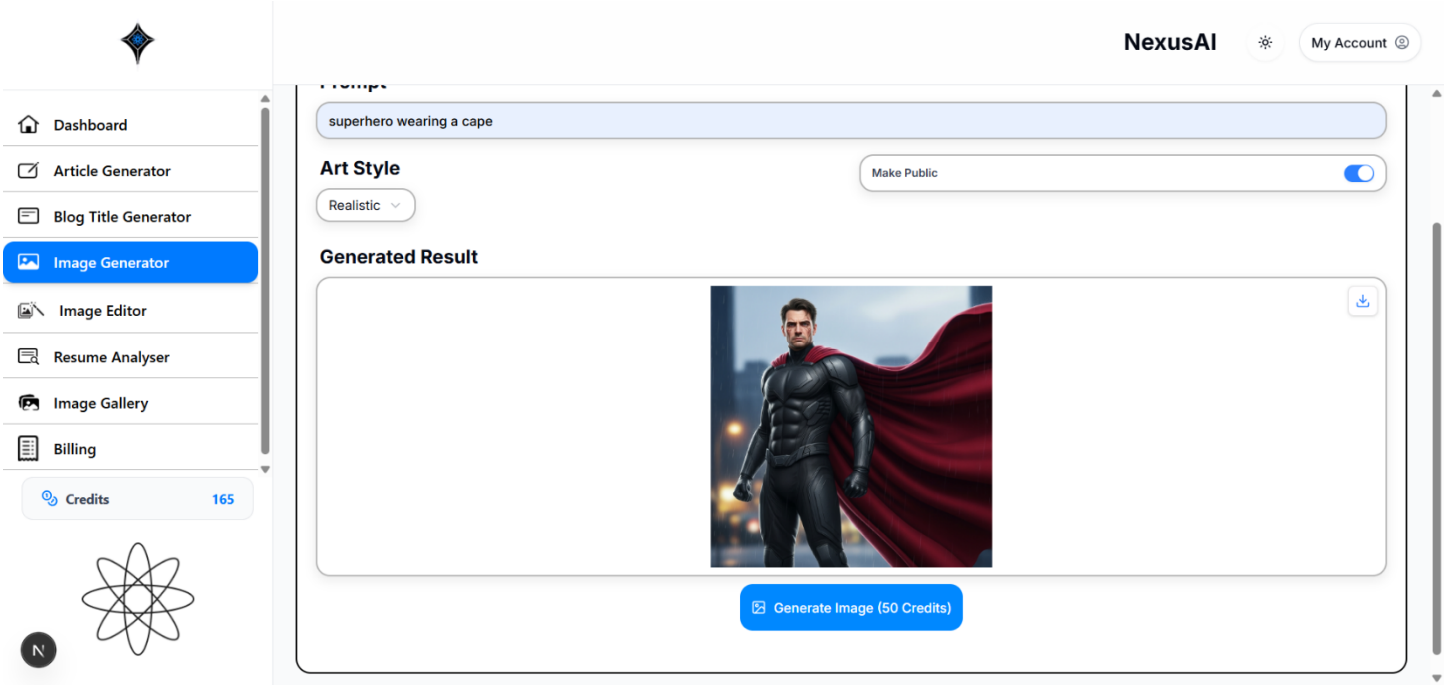


Fig2.5-Image Generation Tool

Fig 2.5 The image generation module enables users to create images using text-based prompts. In this feature, users describe the type of image they want to generate, and the AI system converts the textual description into a visual output. Users can also choose different artistic styles before generating the image.After processing the prompt, the system displays the generated image within the application interface. The user can then download or share the generated result. This functionality demonstrates the integration of artificial intelligence models capable of converting textual inputs into visual representations, which is a significant advancement in generative AI technologies.

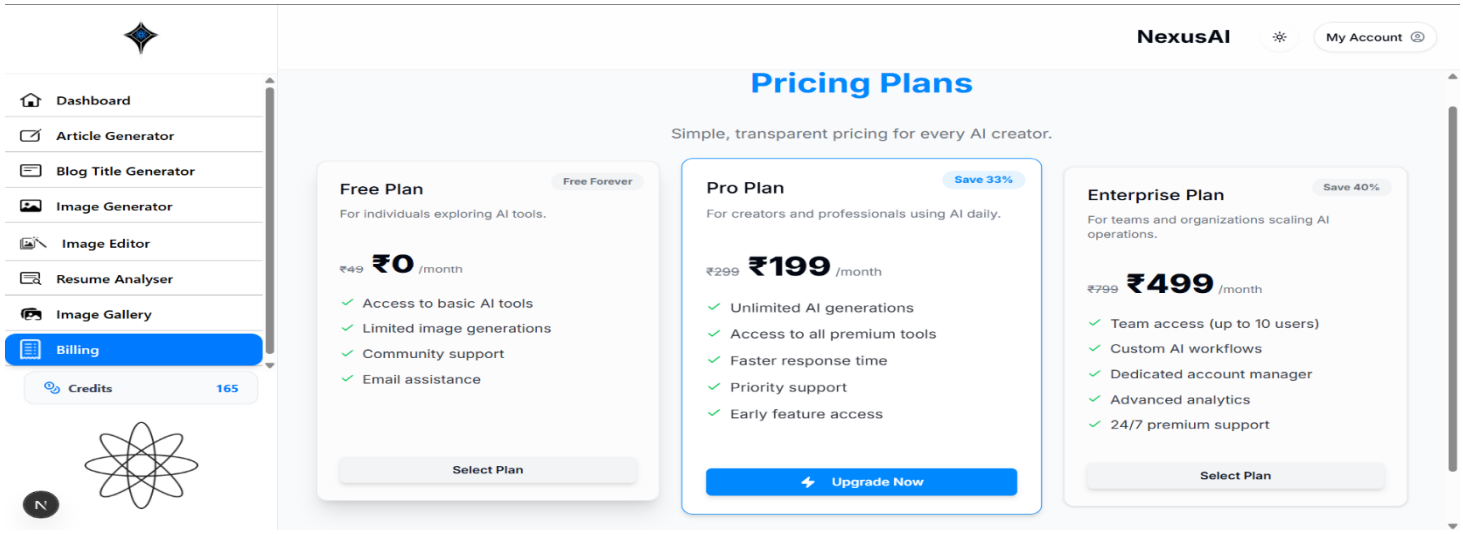


Fig2.6-Billing and Subscription Management

The billing section of the NexusAI platform manages user subscription plans and credit allocation. The system provides multiple pricing options, including Free, Pro, and Enterprise plans, each offering different levels of access to AI services. These plans allow users to select a subscription based on their requirements and usage frequency.The billing interface clearly displays pricing details, available features, and the benefits associated with each plan. Users can upgrade their subscription directly through the interface to unlock additional AI tools or increased usage limits. This subscription-based model supports the sustainable operation of the platform by regulating access to computational resources.

VI. CONCLUSION

This research presents the design and development of NexusAI, a web-based Artificial Intelligence Software as a Service platform that integrates multiple AI-powered creative tools within a unified system. The objective of the proposed platform is to simplify user interaction with artificial intelligence technologies by combining various AI functionalities such as article generation, blog title creation, image generation, image editing, and resume analysis into a single application. By providing these services through a centralized interface, the system reduces the need for users to access multiple independent platforms and improves overall workflow efficiency.

The implementation of the platform using the PERN technology stack, which includes PostgreSQL, Express.js, React.js, and Node.js, ensures a scalable and efficient system architecture. The modular design of the application allows seamless communication between the user interface, backend services, and database management system. This architecture supports reliable request processing, structured data storage, and smooth integration of external artificial intelligence services.

An important contribution of the proposed system is the introduction of a credit-based resource management mechanism. This mechanism controls the usage of AI services by assigning credits to users and deducting them whenever AI operations are performed. Such an approach helps maintain fair resource distribution, prevents excessive API consumption, and supports the sustainable operation of the platform.

The developed system demonstrates how multiple AI tools can be successfully integrated into a unified SaaS platform to enhance accessibility and usability. By combining modern web technologies with artificial intelligence services, NexusAI provides an efficient environment where users can perform different AI-assisted tasks through a single interface.

The results of the implementation indicate that centralized AI platforms can significantly improve productivity and simplify the interaction between users and AI technologies.

Overall, the proposed system highlights the potential of AI SaaS platforms in delivering scalable, accessible, and efficient intelligent services. The architecture and methodology presented in this research can serve as a foundation for the development of future AI-driven web applications.

VII. FUTURE SCOPE

The NexusAI platform provides a foundation for integrating multiple artificial intelligence tools within a single Software as a Service environment. In the future, the system can be enhanced by incorporating additional AI capabilities such as video generation, voice synthesis, and advanced language translation tools. These features would expand the functionality of the platform and make it useful for a wider range of creative and professional applications.

Another potential improvement is the implementation of intelligent recommendation systems. By analyzing user behavior and interaction patterns, the platform could provide personalized suggestions for AI tools and content generation. This would help users obtain more relevant results while improving the overall user experience and efficiency of the system.

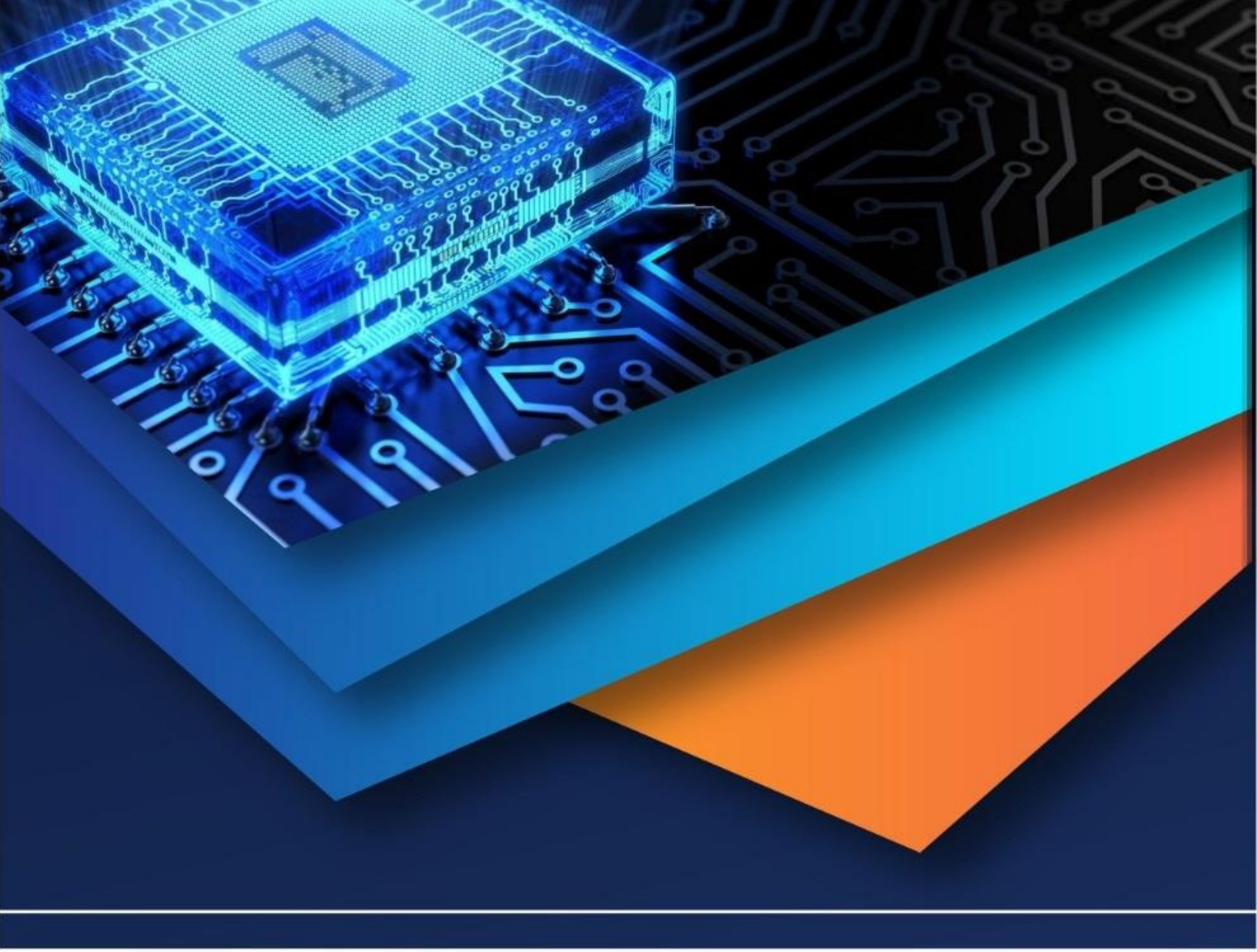
Future development may also focus on improving system scalability and security. Deploying the platform on cloud infrastructure with load balancing mechanisms can support a larger number of users without affecting performance. In addition, stronger security techniques such as advanced authentication and encrypted data communication can be implemented to ensure safe and reliable operation of the platform.

REFERENCES

[1] S. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[2] T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877–1901.
[3] J. Smith and R. Kumar, “Generative Artificial Intelligence Systems in Web-Based Platforms,” Journal of Artificial Intelligence Systems, vol. 12, no. 3, pp. 45–

58, 2024.
[4] P. Gupta, “Software as a Service Architectures for Artificial Intelligence Applications,” International Journal of Cloud Computing, vol. 9, no. 2, pp. 101–115, 2023.
[5] L. Zhao, Y. Chen, and M. Wang, “Multi-Model Artificial Intelligence Integration Frameworks for Web Applications,” IEEE Access, vol. 11, pp. 22345–22358, 2023.
[6] R. Fielding, “Architectural Styles and the Design of Network-Based Software Architectures,” ACM Transactions on Internet Technology, 2020.
[7] M. Stonebraker and L. Rowe, “The PostgreSQL Database Management System,” Communications of the ACM, vol. 64, no. 2, pp. 112–120, 2021.
[8] D. Taibi, V. Lenarduzzi, and C. Pahl, “Architectural Patterns for Microservices: A Systematic Mapping Study,” Journal of Systems and Software, vol. 150, pp. 77–97, 2019.
[9] A. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux. Sebastopol, CA, USA: O’Reilly Media, 2020.
[10] E. Casalicchio and V. Perciballi, “Auto-Scaling of Web Applications in Cloud Computing Environments,” Future Generation Computer Systems, vol. 27, no. 5, pp. 633–642, 2019.
[11] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2018.
[12] S. Newman, Building Microservices: Designing Fine-Grained Systems. Sebastopol, CA, USA: O’Reilly Media, 2021.
[13] D. Chappell, Enterprise Service Bus. Sebastopol, CA, USA: O’Reilly Media, 2019.
[14] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, NIST Special Publication 800-145, 2011.
[15] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Proc. Advances in Neural Information Processing Systems, 2012, pp. 1097–110.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)