iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79720

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 IV Apr 2026- Available at www.ijraset.com

AcademiSync: A Centralized System for Project Managementwith Role-Based Access Control

Sivadharshini A[1], Fathima Asmi Y[2], Vishnupriya P[3], Ms.S.Elakkiya M.E[4]
[1, 2, 3]*B.Tech,* [4]*Assistant Professor, Information Technology, M.I.E.T Engineering College, Tiruchirappalli, Tamil Nadu, India*

Abstract: ***This paper presents an web-based academic project management system designed to streamline the process of final year project submission, evaluation, and monitoring in educational institutions. Traditional project management methods often rely on manual processes and fragmented communication, leading to inefficiencies in tracking, evaluation, and data management. To overcome these limitations, the proposed system introduces a centralized platform that integrates student, staff, and administrative functionalities within a single environment. The system is built using a role-based access control mechanism, where students can upload and manage project submissions, staff members can review and provide feedback, and administrators monitor overall system activities. A key feature of the system is the implementation of automated activity monitoring, which tracks user behavior such as login attempts and data modifications. Suspicious activities, including unauthorized access and abnormal usage patterns, are detected using adaptive threshold mechanisms, and alerts are generated to notify administrators. The system also includes a project gallery that displays approved projects categorized by domain, allowing external users to access and reference academic work. By integrating structured workflow management, real-time monitoring, and secure data handling, the proposed system improves efficiency, enhances transparency, and ensures data integrity. This solution provides a scalable and secure approach for managing academic projects in modern educational environments.***

I. INTRODUCTION

The proposed system provides a unified platform where all stakeholders—Administrators, Staff, and Students—can interact within a single ecosystem. The primary objective is to automate the project workflow from the initial "Project Idea" phase to the final "Review and Grading" stage.Final Year Projects (FYP) play a significant role in undergraduate education by enabling students to apply theoretical knowledge to real-world problems [1]. However, many institutions still rely on traditional project management approaches, which are inefficient and lack proper coordination mechanisms. These systems often fail to provide structured workflows for submission, evaluation, and monitoring. Project allocation and supervision are complex processes that must consider multiple constraints, including student preferences, faculty workload, and domain relevance [2]–[4]. Existing systems often struggle to handle these challenges effectively, leading to inefficiencies and dissatisfaction among users. Traditional systems rely heavily on email communication or manual submissions, limiting realtime interaction and structured feedback [6]. Moreover, these systems lack security mechanisms to monitor user activities and prevent unauthorized access or data manipulation [7].The proposed system addresses these limitations by providing a centralized web-based platform that integrates submission, evaluation, and monitoring. A key feature is automated activity monitoring, where the administrator supervises system behavior instead of manually verifying each user. This improves both security and scalability. The system also includes a project gallery for knowledge sharing, making it more useful for academic environments. outputs.

II. RELATED WORK

A. Traditional Academic Project Management Systems

Early academic project management systems relied on manual processes and basic digital tools such as email communication and document sharing platforms. These systems lacked centralized control and structured workflows, making it difficult to manage project submissions, track progress, and provide systematic feedback. Existing web-based systems improved accessibility but were limited to basic functionalities such as file uploads and simple communication [6]. However, these systems lacked integrated monitoring and security mechanisms, making them vulnerable to unauthorized access and data manipulation. The proposed system addresses these limitations by introducing a centralized and secure workflow with automated monitoring capabilities.

B. Web-Based Project Submission and Evaluation Systems

Several web-based platforms have been developed to support project submission and evaluation processes. These systems allow students to upload project documents and enable staff members to review and provide feedback. While such systems improve communication and reduce manual effort, they often lack structured workflows and real-time tracking features [7]. Additionally, many systems do not support iterative evaluation, where students can update their work based on feedback. The proposed system enhances these approaches by incorporating an iterative feedback mechanism and rolebased access control, ensuring efficient collaboration between students and staff.

C. Role-Based Access Control Systems

Role-Based Access Control (RBAC) has been widely used in web applications to restrict access based on user roles and permissions. In academic systems, RBAC ensures that students, staff, and administrators can only access functionalities relevant to their roles. Existing implementations of RBAC improve system organization but often lack integration with monitoring and security mechanisms [7]. The proposed system extends RBAC by combining it with activity tracking and automated alert generation, ensuring both secure access and continuous system monitoring.

D. Activity Monitoring and Security Mechanisms

Modern web applications increasingly incorporate activity monitoring and anomaly detection techniques to enhance system security. These systems track user behavior, including login attempts and data modifications, to identify suspicious activities [7]. However, many academic project management systems do not implement such mechanisms, relying instead on manual verification. The proposed system introduces an adaptive monitoring approach, where user activity is continuously analyzed, and alerts are generated when abnormal behavior is detected. This improves system reliability and reduces administrative workload.

E. Project Repository and Gallery Systems

Project repository systems and digital libraries provide platforms for storing and accessing academic work. These systems allow users to browse and download project documents, promoting knowledge sharing. However, most existing repositories function as static storage systems without integration into the project workflow. They do not support submission, evaluation, or monitoring processes. The proposed system integrates a dynamic project gallery that displays only approved projects, categorized by domain, ensuring both quality control and accessibility.model.

III. PROBLEM STATEMENT

Despite the adoption of digital tools in academic environments, project management in many institutions still faces several practical challenges:

1) Lack of centralized workflow:Traditional systems rely on multiple platforms such as email and manual submissions, leading to fragmented communication and difficulty in tracking project progress.
2) Inefficient submission and evaluation process: Students submit projects without a structured system, and staff members lack a unified platform to review, comment, and manage multiple submissions effectively.
3) Absence of real-time monitoring: Most existing systems do not track user activities such as login attempts or data modifications, making it difficult to detect unauthorized access or misuse.
4) Security vulnerabilities: Without proper activity logging and monitoring, systems are prone to data tampering, unauthorized changes, and potential misuse by invalid users.
5) Lack of structured feedback mechanism: Many systems do not support iterative correction workflows, where students can update projects based on staff feedback, resulting in reduced project quality.
6) Unorganized project storage and accessibility: Existing approaches lack a controlled project repository, where only approved and verified projects are published for external access and academic reference.
7) Manual administrative workload: Administrators often rely on manual verification and supervision, which is timeconsuming and not scalable for large numbers of users.

IV. OBJECTIVES

The specific objectives of this project are:

1) To develop a web-based platform that enables students to upload and manage their academic project submissions through a user-friendly interface.
2) To provide a structured system for staff members to review submitted projects, add comments, and suggest corrections for continuous improvement.
3) To implement a role-based access control mechanism that differentiates functionalities among students, staff, and administrators, ensuring secure and controlled system access.
4) To design an automated monitoring system that tracks user activities such as login attempts, file uploads, and data modifications in real time.
5) To detect suspicious activities, including unauthorized access and abnormal system behavior, using adaptive monitoring techniques and generate alerts for administrative action.
6) To maintain a centralized database for storing project details, evaluation feedback, and activity logs, ensuring data consistency and integrity.
7) To implement a structured workflow where projects are submitted, reviewed, corrected, and approved before final publication.
8) To develop a project gallery that displays only approved projects, categorized by domain, allowing external users to view and reference academic work.
9) To reduce manual workload for administrators by replacing continuous manual verification with automated activity tracking and alert mechanisms.
10) To enhance overall efficiency, transparency, and security in academic project management systems.

V. SYSTEM ARCHITECTURE

A. *Presentation Layer (User Interface)*

The presentation layer provides a web-based interface for all users, including students, staff, and administrators. It is developed using HTML, CSS, and JavaScript to ensure an interactive and userfriendly experience.

Students use this interface to upload project details and track submission status. Staff members access project submissions, review them, and provide feedback. Administrators use a dedicated dashboard to monitor system activities, user behavior, and overall system performance. The project gallery is also part of this layer, allowing external users to view approved projects.

B. *Application Layer (Business Logic and Processing)*

The application layer handles the core functionality of the system. It is implemented using backend technologies such as Node.js and manages request processing, authentication, and workflow control.

This layer enforces role-based access control, ensuring that users can only access functionalities permitted for their role. It manages the complete project lifecycle, including submission, evaluation, correction, and approval.

A key component of this layer is the monitoring module, which continuously tracks user activities such as login attempts, file uploads, and data modifications. The system applies adaptive threshold logic to detect abnormal behavior. If suspicious activity is identified, such as repeated failed login attempts or unauthorized data changes, alerts are generated and sent to the administrator for immediate action.

C. *Workflow Management Module*

Within the application layer, a workflow module controls the project lifecycle. When a student uploads a project, it is stored and forwarded to the staff for evaluation. Staff members review the project and either approve it or return it for correction.

This iterative process ensures continuous improvement of project quality. Once approved, the project is locked to prevent further modifications and is automatically published in the project gallery. This structured workflow ensures data consistency and proper project validation.

D. *Data Layer (Database and Storage)*

The data layer is responsible for storing all systemrelated information, including user credentials, project details, feedback, and activity logs. A structured database management system ensures efficient storage, retrieval, and management of data.

Activity logs play a critical role in maintaining system security, as they are used by the monitoring module to analyze user behavior and detect anomalies. File storage mechanisms are used to securely store project documents and associated data.

E. Security and Monitoring Integration

Security is integrated across all layers of the system. Authentication mechanisms ensure secure login, while rolebased access control restricts unauthorized access. The monitoring system continuously analyzes activity logs to identify potential threats.

The implementation of automated alert mechanisms reduces the need for manual supervision and enhances system reliability. This integrated approach ensures that the system remains secure, scalable, and efficient in handling academic project workflows.

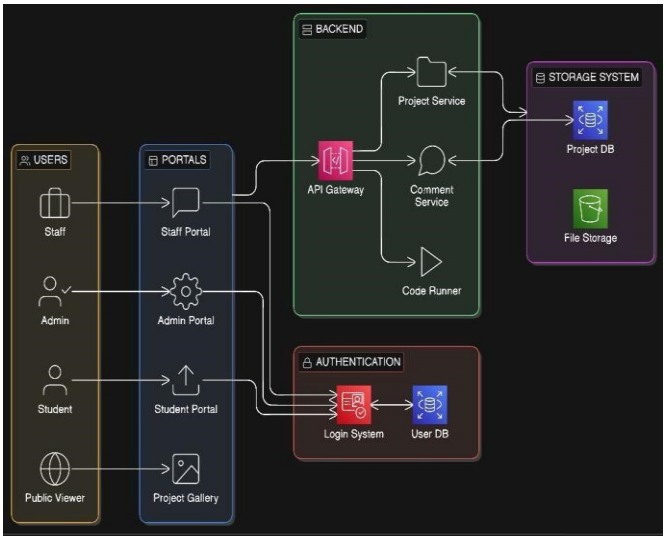


Fig .1 System Architecture

The implementation of automated alert mechanisms reduces the need for manual supervision and enhances system reliability. This integrated approach ensures that the system remains secure, scalable, and efficient in handling academic project workflows.

VI. MODULE DESCRIPTION

A. Module 1: User Authentication and Input Handling

Purpose: To manage secure user login and identify user roles for system access.

Input: User credentials (username, password).

Process: The user logs into the system through the web interface. The system validates credentials using secure authentication mechanisms such as password hashing. Based on the verified credentials, the system assigns a role (student, staff, or administrator) using role-based access control. Unauthorized access attempts are logged for monitoring.

Output:Authenticated user session with role-specific dashboard access.

B. Module 2: Project Submission and Management

Purpose: To allow students to upload and manage academic project submissions.

Input: Project details, documents, and related files uploaded by students.

Process: Students submit project information through the interface. The system validates file type and size before storing it in the database. Each submission is tagged with metadata such as student ID, project title, and submission time. Students can edit or update their projects until the evaluation stage.

Output: Stored project data with associated metadata, ready for staff evaluation.

C. Module 3: Project Evaluation and Feedback

Purpose: To enable staff members to review submitted projects and provide structured feedback.

Input: Project data submitted by students.

Process: Staff members access assigned projects through their dashboard. They review the content, add comments, and suggest corrections. The system supports an iterative workflow where projects can be returned to students for modification. Once the project meets the required standards, it is marked as approved.

Output: Reviewed project with feedback status (approved or correction required).

D. Module 4: Monitoring and Security System

Purpose: To track system activities and detect suspicious behavior.

Input: User activity data such as login attempts, file uploads, and data modifications.

Process: All user actions are recorded in activity logs. The system continuously analyzes these logs using an adaptive threshold mechanism to identify abnormal behavior. Events such as repeated failed login attempts or unauthorized data access trigger alerts. Notifications are sent to the administrator for further action.

Output: Activity logs and real-time alerts for suspicious system behavior.

E. Module 5: Workflow Management and Approval System

Purpose: To manage the lifecycle of project submissions from upload to final approval.

Input: Project status updates from student and staff modules.

Process: The system controls the workflow where projects move from submission to evaluation and then to approval. If corrections are required, the project is sent back to the student. Once approved, the project is locked to prevent further modifications and marked as finalized.

Output: Approved and finalized project ready for publication.

F. Module 6: Project Gallery and Output Display

Purpose: To display approved projects for external access and academic reference.

Input: Approved project data from the workflow module.

Process: The system categorizes approved projects based on domain or topic and publishes them in the project gallery. External users can view project details through a public interface. Only verified and approved projects are displayed to ensure quality and reliability.

Output: Categorized project gallery accessible to external users.

VII. ALGORITHMS

A. User Authentication and Access Control

User authentication ensures secure access to the system by validating user credentials. The input consists of login details (username and password), which are verified using secure hashing mechanisms. Role-Based Access Control (RBAC) assigns permissions based on user roles (student, staff, administrator).Let U represent the set of users and R represent roles. Access is granted based on: $Access(U)=f(Role(U))$

Unauthorized login attempts are logged, and repeated failures trigger alert mechanisms. This ensures controlled system access and prevents unauthorized usage.

B. Project Submission and Data Handling

Students submit project data through the web interface, including files and metadata. Each submission is validated for file type, size, and completeness before storage.

Let P represent a project submission:P={Title, Description ,File, StudentID, Timestamp}

The system stores project data in a structured format and associates it with the respective user. This ensures organized data handling and easy retrieval for further processing.

C. Project Evaluation and Feedback Mechanism

Staff members evaluate submitted projects and provide feedback. The evaluation process follows an iterative model where projects can be corrected and resubmitted.

Let S represent submission status:

$S \in \{Pending, Reviewed, Correction, Approved\}$

The system updates the project status based on staff input. Feedback is stored and linked to the project, enabling continuous improvement and structured evaluation.

D. Activity Monitoring and Anomaly Detection

All user activities, including login attempts, file uploads, and data modifications, are recorded in activity logs. The system applies an adaptive threshold mechanism to detect abnormal behavior.

Suspicious activities such as repeated login failures or unauthorized data changes trigger alerts. These alerts are sent to the administrator for further action, improving system security and reliability.

E. Workflow Processing and Approval Logic

The system follows a structured workflow for project management. After submission, projects are reviewed by staff and either approved or returned for correction.

Workflowstages:

Submission→Review→Correction→ApprovalOnce approved, the project is locked to prevent further modification. This ensures data integrity and maintains consistency in project records.

F. Project Gallery and Output Presentation

Approved projects are displayed in a categorized project gallery for external access. The system organizes projects based on domain or category, enabling easy navigation.

Only verified and approved projects are published:

Gallery={P|Status(P)=Approved}

This ensures that only high-quality and validated academic work is accessible to users, promoting knowledge sharing and transparency.

VIII. IMPLEMENTATION

A. *Technology Stack Frontend:*

HTML5, CSS3, JavaScript (Vanilla) are used to design a responsive and user-friendly interface. The frontend includes dashboards for students, staff, and administrators, along with forms for project upload, review, and monitoring.

1) Backend:

Node.js with Express.js framework is used to handle server-side logic, routing, and API development. It manages authentication, project workflows, and system monitoring functionalities.

2) Database:

MongoDB / MySQL is used to store user data, project details, feedback, and activity logs. Structured schemas ensure efficient data retrieval and consistency.

3) Authentication & Security:

Password hashing using bcrypt and session/JWT-based authentication are implemented. Role-Based Access Control (RBAC) ensures restricted access for students, staff, and administrators.

4) Monitoring System:

Custom logging mechanisms are implemented to track user activities such as login attempts, file uploads, and data modifications. Alerts are generated for suspicious activities.

B. *Authentication and Role Management*

User authentication is initialized during login, where credentials are validated against stored hashed passwords.

```
const user = await User.findOne({ email }); if
      (user      &&      bcrypt.compare(password,
user.password)) {
generateToken(user.role);
}
```

Once authenticated, the system assigns access permissions based on user roles:

Student → Upload & manage projects
Staff → Review & comment
Admin → Monitor & control system
This ensures secure and controlled system interaction.

C. *Project Handling and Workflow Processing*

Projects submitted by students are processed through a structured workflow. Each project is stored with metadata including title, description, and submission timestamp.
Example structure:

```
{
 "title": "Project Title",
 "studentId": "123",
 "status": "Pending",
 "file": "project.pdf"
}
```

The workflow follows:

Upload → Stored in database
Review → Staff evaluates
Correction → Sent back if needed
Approval → Finalized project
This ensures proper lifecycle management and data consistency.

D. .*Monitoring and Activity Logging*

All system activities are logged for security and analysis. Events such as login attempts, file uploads, and data modifications are recorded.
Example log structure:

```
{
 "userId": "123",
 "action": "LOGIN_ATTEMPT",
 "timestamp": "2026-04-07",
 "status": "FAILED"
}
```

The system evaluates activity frequency and detects anomalies:

Multiple failed logins
Unauthorized data modification
Suspicious access patterns

If abnormal behavior is detected, alerts are generated and sent to the administrator dashboard

IX. EXPERIMENTAL RESULT

A. Test Setup

All experiments were conducted on a system with Intel Core i5 processor, 8 GB RAM, and a standard web browser environment. The application was deployed locally using a Node.js server and tested under real-time conditions.
The test dataset included sample users from three roles: students, staff, and administrators. A total of 50 project submissions were used for evaluation, including different types of project files and metadata. System performance was evaluated based on usability, response time, workflow efficiency, and security monitoring capabilities.

B. System Performance Evaluation

Table I presents the performance of the system based on workflow efficiency and user interaction.

TABLE
System Performance Metrics

Metric	Value
Average Login Time	1.2 s
Project Upload Time	2.8 s
Review Processing Time	3.5 s
Alert Detection Time	< 1 s
Overall System Response	2.5 s

The results indicate that the system performs efficiently under normal usage conditions. The monitoring module detects suspicious activities in real time, ensuring quick administrative response. The structured workflow significantly reduces delays in project evaluation.

C. Workflow Efficiency

The system was tested for project lifecycle management across multiple iterations. On average, each project required 1– 2 review cycles before approval.

Compared to traditional manual systems, the proposed system reduced processing time by approximately 35–40%. Students were able to track project status in real time, while staff provided structured feedback through the system interface.

D. Monitoring and Security Evaluation

The monitoring system was evaluated by simulating abnormal activities such as repeated failed login attempts and unauthorized data modification.

The system successfully detected:

- Multiple failed login attempts
- Unauthorized access to restricted modules
- Suspicious data modification attempts

Alerts were generated instantly and displayed in the admin dashboard. This demonstrates the effectiveness of the adaptive monitoring mechanism in enhancing system security.

E. Output and Usability Evaluation

The system interface and project gallery were tested with multiple users. Approved projects were correctly displayed in categorized formats, and access was restricted to verified content only.

User feedback indicated:

- Improved ease of use
- Better transparency in project tracking
- Faster communication between students and staff

X. DISCUSSION

A. What Worked Well

The role-based system architecture proved to be the most effective design decision. By implementing Role-Based Access Control (RBAC), the system clearly separates functionalities between students, staff, and administrators, ensuring secure and organized access. This approach improves workflow efficiency and prevents unauthorized operations within the system.

The structured project workflow significantly enhances academic project management. Students can submit projects, track progress, and receive feedback in a systematic manner, while staff members can review and provide corrections efficiently. This reduces communication gaps and ensures better coordination compared to traditional methods.

The monitoring and alert mechanism is another key strength of the system. By tracking user activities such as login attempts and data modifications, the system is able to detect suspicious behavior in real time. This reduces the need for manual verification and improves overall system security.

Additionally, the project gallery feature provides a centralized repository of approved projects. This enables knowledge sharing and allows external users to access validated academic work, which is not commonly available in traditional systems.

B. Limitations

One of the main limitations of the system is its dependency on manual evaluation by staff members. Although the system provides a structured workflow, the quality of project evaluation still depends on human input, which may vary between reviewers.

The monitoring system currently uses rule-based detection for identifying suspicious activities. While effective for basic security, it may not detect complex or advanced threats. Incorporating intelligent anomaly detection techniques could improve accuracy.

The system performance may also be affected when handling a large number of users or project uploads simultaneously. Without optimization techniques such as load balancing or cloud scaling, response time may increase under heavy usage.

Another limitation is the lack of advanced analytics, such as automated project quality assessment or plagiarism detection, which could further enhance the system's capabilities.

C. Future Work

Several improvements can be made in future versions of the system. First, integrating machine learning-based anomaly detection can enhance the monitoring system by identifying complex patterns of suspicious behavior.

Second, implementing cloud-based deployment and scalable infrastructure can improve system performance and handle largescale usage efficiently.

Third, adding plagiarism detection and automated project evaluation tools can support staff members and improve the quality of project assessment.

Fourth, introducing notification systems such as email or mobile alerts can enhance communication between students, staff, and administrators.

Finally, incorporating analytics dashboards for administrators can provide insights into system usage, project trends, and performance metrics, enabling better decisionmaking.

D. END-TO-END WORKFLOW

To illustrate the system functionality, consider a realworld scenario where a student submits a final year project for evaluation.

Step 1 — Login:

The student logs into the web application using valid credentials. The system authenticates the user and grants access to the student dashboard based on role.

Step 2 — Project Submission:

The student uploads project details and documents through the interface. The system validates the input and stores the data with a "Pending" status.

Step 3 — Staff Evaluation:

The staff member accesses the submitted project through their dashboard. The project is reviewed, and feedback is provided. If corrections are required, the project is returned to the student.

Step 4 — Correction and Resubmission:

The student updates the project based on feedback and resubmits it. This iterative process continues until the project meets the required standards.

Step 5 — Approval:

Once the project is approved by the staff, its status is updated to "Approved." The system locks the project to prevent further modifications.

Step 6 — Monitoring and Security Check:

Throughout the process, the system continuously monitors user activities. Any suspicious behavior, such as unauthorized access or abnormal data changes, triggers alerts for the administrator.

Step 7 — Publication in Project Gallery:

The approved project is published in the project gallery, where it becomes accessible to external users for reference and knowledge sharing.

Step 8 — Admin Oversight:

The administrator monitors system performance, user activity logs, and security alerts through a dedicated dashboard, ensuring smooth and secure operation.

XI. CONCLUSION

This work presented the design and implementation of a webbased academic project management system aimed at improving efficiency, transparency, and security in handling final year projects. The proposed system integrates project submission, staff evaluation, and administrative monitoring into a unified platform, eliminating the limitations of traditional manual and partially digital systems.

The implementation of role-based access control ensures that students, staff, and administrators have clearly defined functionalities, enhancing both usability and system security. The structured workflow model enables systematic project submission, review, correction, and approval, resulting in improved coordination and reduced delays. The integrated monitoring mechanism plays a significant role in enhancing system reliability by tracking user activities and detecting unauthorized access or suspicious data modifications in real time.

The project gallery feature further extends the system's usefulness by providing a centralized repository of approved academic work, promoting knowledge sharing and accessibility. Overall, the proposed system offers a scalable and practical solution suitable for educational institutions.

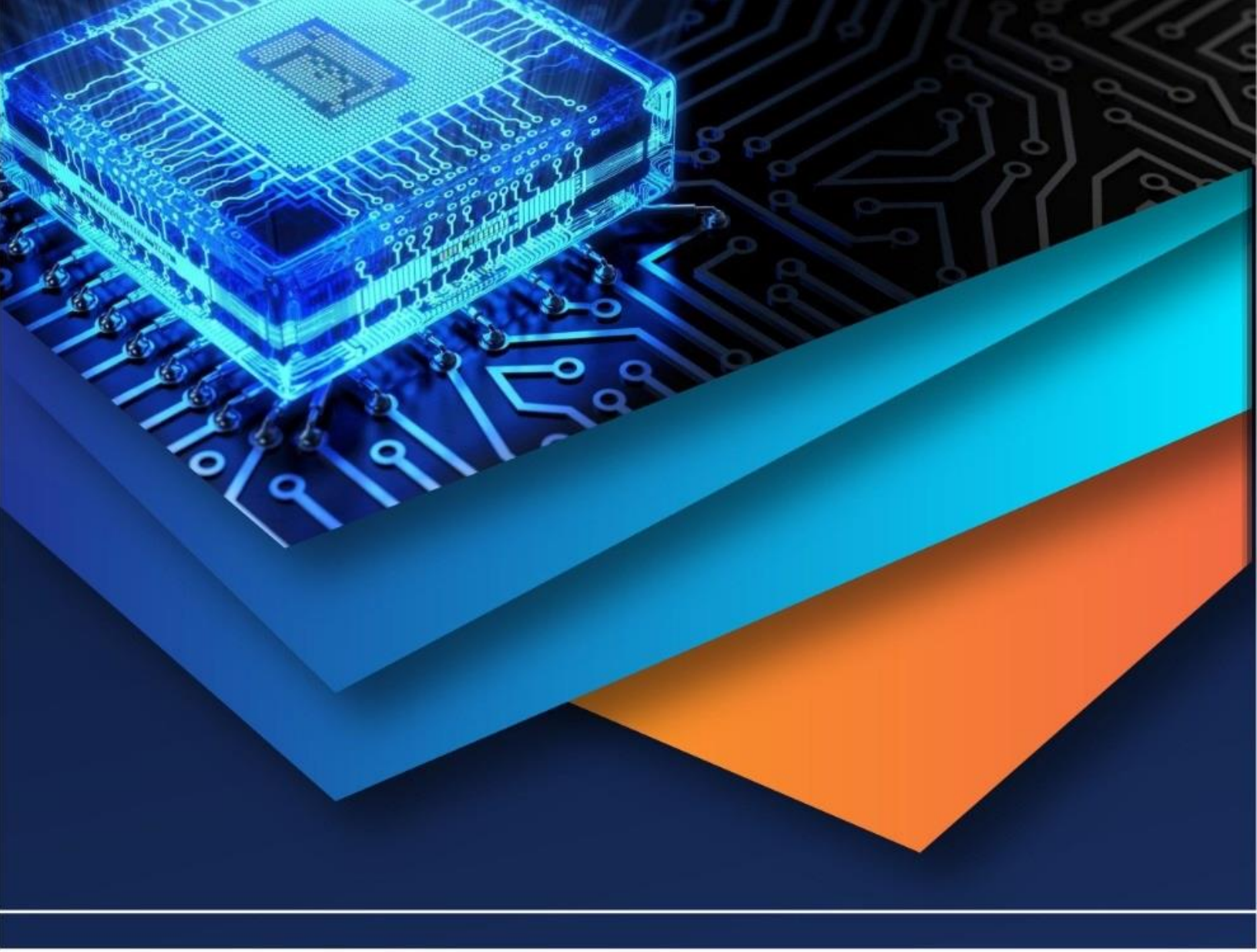
XII. ACKNOWLEDGMENT

We express our sincere gratitude to the Department of Information Technology of our institution for providing the necessary infrastructure and support to successfully complete this project. We would like to thank our project guide for their continuous guidance, valuable suggestions, and encouragement throughout the development process.

Their insights into system design, workflow structuring, and security considerations played a crucial role in shaping this project. We also extend our appreciation to all faculty members and peers who contributed directly or indirectly to the successful completion of this work.

REFERENCES

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts, 10th ed. Wiley, 2018.
[2] I. Sommerville, Software Engineering, 10th ed. Pearson, 2016.
[3] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, 1999.
[4] D. Flanagan, JavaScript: The Definitive Guide, 7th ed. O'Reilly Media, 2020.
[5] M. Richards and N. Ford, Fundamentals of Software Architecture: An Engineering Approach. O'Reilly, 2020.
[6] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," 2021.
[7] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. McGraw-Hill, 2015.
[8] N. MongoDB Inc., "MongoDB Documentation," [Online]. Available: https://www.mongodb.com
[9] Node.js Foundation, "Node.js Documentation," [Online]. Available: https://nodejs.org
[10] J. Nielsen, "Usability Engineering in Web Applications," Morgan Kaufmann, 1994.
[11] S. Pearson, "Taking Account of Privacy when Designing Cloud Computing Services," in Proc. CLOUD, 2009.
[12] A. Greenberg et al., "The Cost of a Cloud: Research Problems in Data Center Networks," ACM SIGCOMM, 2008.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY