iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72024

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 13 Issue VI June 2025- Available at www.ijraset.com

Advanced Web-Based Admission Automation: Spot Round Merit Processing with Dynamic Database Creation, Concurrency Handling, and Chatbot Support

Anushree K. Kolte[1], Poorva S. Marne[2], Gauri S. Bhanage[3], Dhanashree A. Thombare[4], Dr. Sharmila K. Wagh[5]
Department ofComputer Engineering, MES Wadia College of Engineering, Pune, India

Abstract: ***Manual academic admission processes are often time-consuming, error-prone, and inefficient, particularly when handling large-scale merit-based evaluations. This paper pro- posesafull-stackautomatedadmissionmanagementsystem that streamlines application handling, merit list generation, and student interaction using modern web technologies. The backend is developed using Node.js and Express, while MongoDB stores structured applicant data dynamically. Uploaded PDF merit lists are parsed using Python scripts and inserted into database collections for filtering and retrieval. The system features a secure admin login with JWT authentication, OTP-based email verification for applicants, and dynamic application form links.It also includes a built-in chatbot powered by a natural language processing (NLP) engine for answering applicant queries in real time. Experimental evaluations demonstrate that the system can efficientlyprocessmeritlistdata,generatecategory-basedfiltered results, and provide seamless user interaction. This work offersa scalable, intelligent, and secure solution suitable for digitizing the admission process in academic institutions***

Keywords: Automated System, MERN Stack, PDF Data Extraction,MongoDB,OTPVerification,NaturalLanguagePro- cessing, Chatbot, Web Application Security

I. INTRODUCTION

Academic admissions in competitive fields such as engineering and postgraduate programmes present significant administrative challenges, requiring institutions to manage extensive applicant datasets, develop merit rankings, and implement equitable selection procedures. Traditional approaches rely heavily on manual processes or rudimentary digital solutions, frequently resulting in data discrepancies, procedural bottlenecks, limited transparency, and suboptimal experiences for prospective students.

The growing applicant pools and increasingly nuanced merit-based selection frameworks necessitate sophisticated, automated admission management infrastructure within educational institutions. Current inefficiencies---including manual document processing, inadequate identity verification protocols, and restricted communication channels---substantially compromise operational effectiveness.

This research introduces a comprehensive web-based admission framework that automates critical procedural elements---spanning document data extraction, merit list compilation, applicant authentication, and interactive query management. The technical architecture employs the MERN stack (MongoDB, Express.js, React.js, Node.js), supplemented by custom Python algorithms for document analysis and data preprocessing. Additional features encompass secure email-based verification systems and a conversational interface utilising natural language processing to support applicant inquiries.

The system's core objectives encompass:

- Automating the generation of merit rankings from official documentation,
- Implementing secure, role-differentiated access mechanisms for applicants and administrative personnel,
- Improving stakeholder engagement through intelligent support systems.

This technological solution seeks to substantially reduce administrative burdens, minimise human error, and provide educational institutions with a scalable platform for admission process digitalization.

II. RELATED WORK

The increasing complexity and scale of academic admissions have driven institutions to adopt digital platforms for streamlining applicant management, document submission, and result processing. However, most conventional systems lack automation in merit evaluation and interactive support for applicants.

Sharma and Gupta [1] proposed a web-based student registration system that digitises form submission workflows using PHP and MySQL. While useful, the system lacked secure access control, automated document processing, and merit list integration. Similarly, Joshi and Shah [9] implemented a cloud-based admission framework, emphasising scalability but not addressing PDF-based merit data ingestion or intelligent interaction.

Several systems have explored the integration of secure authentication techniques, including OTP-based mechanisms and token-based access, to improve user verification. For instance, Kumar et al. [7] and Choudhury and Saha [10] demonstrated OTP implementations in web platforms, highlighting their effectiveness in preventing unauthorised access and fraud.

Recent studies have also examined the use of modern JavaScript-based web stacks, particularly the MERN (MongoDB, Express.js, React.js, Node.js) architecture, for real-time data processing and dashboard development. Goyal and Verma [16] demonstrated Node.js and MongoDB's suitability for building scalable applications, while Sudhakar and Kumar [11] discussed dynamic data handling using full-stack web technologies.

In the domain of document processing, PDF table extraction remains a critical task for automating merit evaluation. Tools such as Camelot and Tabula have been employed in research for academic data extraction and mining [3], [6]. However, their integration into live web systems for real-time admission processing remains limited.

Conversational agents and chatbots have been increasingly used in e-learning and academic support platforms. Nuruzzaman and Hussain [2] developed an intelligent agent for student guidance, and Bansal et al. [4] utilised NLP techniques to automate student query handling. Winata et al. [14] proposed advanced attention-based models for chatbot design, showcasing real-time responsiveness in user interactions. Despite these advancements, few admission systems have embedded chatbot interfaces into the application process.

To the best of our knowledge, no prior work offers a comprehensive platform that combines real-time PDF merit list ingestion, secure OTP-based access, dynamic data visualisation, and an AI-powered chatbot interface. This paper addresses these gaps by implementing a robust MERN-based solution with integrated Python modules for document parsing and intelligent support.

III. METHODOLOGY

The proposed system is a web-based admission automation platform designed to streamline the entire process of merit-based admissions in academic institutions. It follows a modular architecture consisting of multiple interacting components, as depicted in Figure 1.

A. System Architecture Overview

The system is divided into two main user roles — Admin and Applicant — each interacting with dedicated modules. The core functionalities are handled through a centralised backend connected to a MongoDB database, with frontend interfaces built for each user type. The system supports dynamic data ingestion, secure authentication, application processing, and intelligent query resolution.

B. Admin Functionality

Administrators authenticate via secure login and access a dashboard that provides centralised control over the admission process. Key capabilities include:

- Seat Count Monitoring: Admins can view and configure seat availability per branch or category.
- Link Management: The system allows admins to generate, activate, deactivate, or delete secure application links. Active links are shared with eligible applicants to access the application portal.
- Merit List Processing: Uploaded merit list PDFs are parsed using integrated Python scripts. Extracted tabular data is cleaned and inserted into dynamic MongoDB collections based on examination type (e.g., Exam1, Exam2).
- Data Filtering and Exporting: Admins can filter merit lists by criteria such as category and gender, and download the results in CSV or PDF formats for documentation and decision-making.

C. Applicant Interaction

Applicants access the application portal through system-generated links. The workflow includes:

- OTP-Based Verification: A one-time password is emailed to the applicant for identity verification. Only verified users are allowed to access and submit the application form.
- Application Submission: Verified applicants complete and submit the form, which is then securely stored in the database.
- Chatbot Support: An embedded chatbot enables real-time interaction, answering common queries related to eligibility, form status, or admission procedures using natural language processing techniques.

D. Database Operations

The system utilises a MongoDB database for storing:

- Parsed merit list data per exam type
- Applicant submissions linked to their unique token
- Admin-created links and associated metadata

Dynamic collection creation ensures organised storage, while schema-less design supports flexible data models across various exam formats.

E. Data Flow

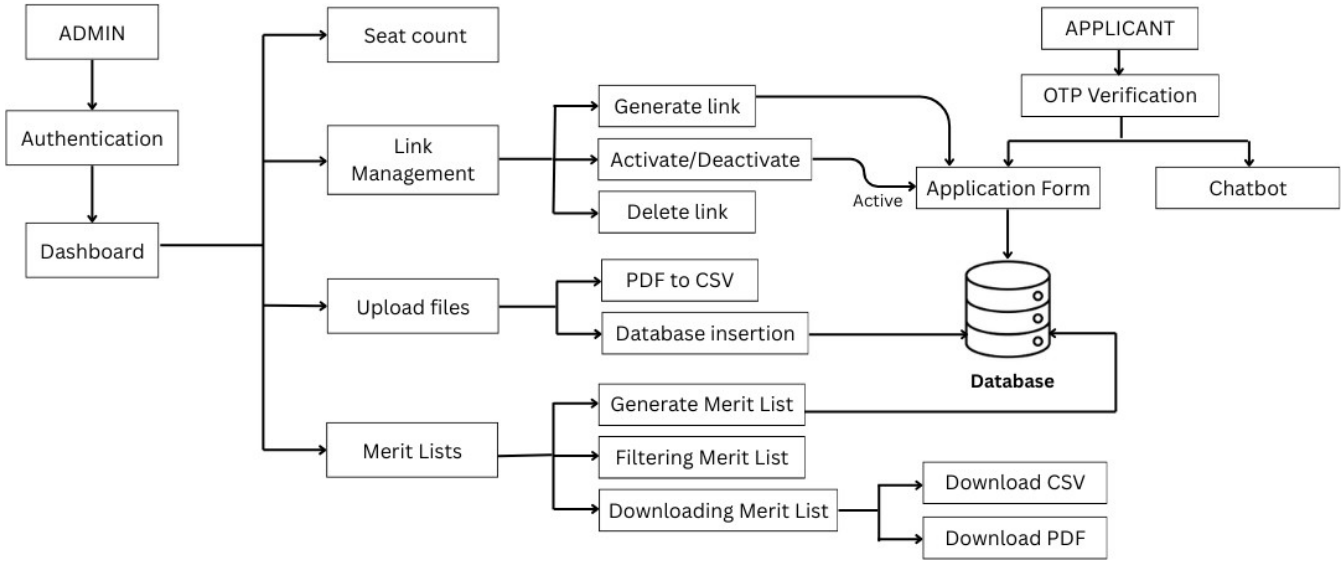


FIG.1.SYSTEMARCHITECTUREDIAGRAM

As illustrated in Figure 1, the workflow is initiated by the admin and flows through merit data ingestion, application distribution, user verification, and application submission. All interactions are tightly coupled with database operations to ensure data integrity and real-time updates.

IV. IMPLEMENTATION

The implementation of the proposed admission management system is realised using the MERN stack architecture, integrated with Python-based tools for data extraction and natural language processing. The development follows a modular and service-oriented approach to facilitate scalability, maintainability, and future extensibility.

A. Backend Architecture

The backend is developed using Node.js with Express.js to handle HTTP requests, authentication, data processing, and API management. MongoDB serves as the NoSQL database for storing applicant data, merit lists, and system metadata. The application follows RESTful principles, ensuring smooth integration with client-side components.

Key backend modules include:

- *Authentication and Authorisation*: Secure admin login using bcrypt for password hashing and JWT for session management.
- *OTP Verification*: Email-based one-time password generation using the nodemailer package, ensuring verified access for applicants.
- *File Upload and Parsing:* Admins upload PDF files via multer. The files are then passed to Python scripts (extract_tables.py and preprocess_csv.py) using child processes to convert tabular data into structured CSV, followed by JSON conversion and insertion into MongoDB.
- *Dynamic Merit List Handling*: Collections are dynamically created based on the type of exam (e.g., Exam1, Exam2), allowing for isolated storage and flexible retrieval.
- *Merit List Filtering and Export*: Admins can filter merit data based on gender, category, or exam type and export the filtered data as CSV or PDF using json2csv.

B. Frontend Interface

The front end is implemented using React.js, offering role-based navigation and responsive layouts. It provides:

- *Admin Dashboard*: Features for file uploads, link management, real-time data metrics, and merit list generation.
- *Applicant Interface*: A guided form submission flow protected by OTP verification, ensuring data authenticity.
- *Chatbot Window*: A real-time chatbot interface integrated with backend Python NLP processing, enabling applicants to receive instant responses to common queries.

The UI incorporates asynchronous operations and client-side validations for enhanced responsiveness and user experience.

C. Python Integration

Python scripts are invoked from the Node.js backend using child processes. These scripts leverage libraries such as Camelot or Tabula (assumed) for PDF table extraction and Pandas for data cleaning and formatting. This hybrid approach ensures accurate extraction of semi-structured data and seamless pipeline integration.

Additionally, a lightweight NLP module powers the chatbot, which interprets user queries and generates responses using rule-based logic or pre-trained models.

D. Security and Session Handling

Security is implemented at multiple levels:

- Token-based access using JWT ensures secure communication between the frontend and backend.
- OTP mechanisms prevent unauthorised access during application submission.
- Data validation and schema enforcement mitigate injection or corruption risks during file ingestion.

E. Deployment and Testing

The application was deployed in a local environment for testing and validation. MongoDB runs as a local instance with dynamic schema-less collections for flexible storage. Backend and frontend services run on separate ports and communicate via HTTP.

System performance and reliability were validated using simulated datasets and realistic admission scenarios, ensuring end-to-end correctness.

V. RESULT

The proposed system was evaluated based on its functional correctness, performance efficiency, and user interaction capabilities. Several key modules including document processing, secure authentication, applicant data submission, merit list filtering, and chatbot interaction were individually and collectively tested to validate their effectiveness in real-world scenarios.

A. Functional Validation

The system was subjected to a series of functional tests simulating the typical workflow of an academic institution during the admission cycle. Merit list documents in PDF format were successfully processed and converted into structured tabular data. These datasets were dynamically stored in the database, and collection-specific queries confirmed the correct ingestion and retrieval of information.

The application form module, secured via one-time password (OTP) verification and token-based access, consistently allowed valid applicants to submit their data. Additionally, the chatbot interface was able to interpret and respond to standard user queries, thereby demonstrating its role in reducing the human workload for repetitive informational requests.

B. Performance Metrics

Performance testing was conducted to assess processing efficiency and system responsiveness. Table 1 summarises the performance of core modules:

TABLEI
PERFORMANCEMETRICSOFCORESYSTEMMODULES

Module	AverageExecutionTime
PDFParsingandIngestion	3–6secondsperdocument
OTP Email DeliveryMerit List	Less than 5 secondsLess than 1 second per query
FilteringChatbotQueryResponse	Approximately1.5secondsperrequest

The results indicate that the system is capable of handling real-time workloads typical in an institutional context. Tests involving bulk data insertion (200,000+ entries) confirmed that database performance remained stable under increased load.

C. Reliability and Usability

System reliability was assessed by observing its behaviour under edge cases such as invalid OTPs, expired tokens, incomplete form submissions, and malformed PDF files. In all cases, the system exhibited appropriate fail-safe responses and feedback mechanisms, thereby enhancing overall robustness.

From a usability standpoint, the interface was designed to support responsive layouts, clear validation prompts, and user-friendly navigation for both administrators and applicants. Administrative functions, such as link generation, merit list download, and data filtering, were performed without functional inconsistencies.

VI. CONCLUSION

This investigation has documented the architectural framework and practical implementation of a computerised admission management solution that remedies significant constraints inherent in conventional merit-driven admission procedures. The engineered system incorporates contemporary web frameworks with computational document analysis and encrypted user verification protocols to establish a dependable and extensible infrastructure for educational institutions.

Constructed utilising the MongoDB, Express.js, React.js, and Node.js technological ecosystem, supplemented by Python-derived PDF extraction utilities, the framework automates the acquisition of structured information from merit catalogues, facilitates applicant authentication through one-time password verification methodologies, and enables configurable merit list formulation with multi-parameter filtering capabilities. Furthermore, the incorporation of an interactive dialogue mechanism enhances participant engagement by facilitating instantaneous query resolution.

Quantitative assessment verified the system's capacity to process substantial data volumes with minimal processing delays and considerable operational reliability. The structural design accommodates modular deployment, rendering it adaptable to diverse institutional requirements. These findings substantiate the system's capacity to markedly improve the efficiency, transparency, and accessibility of digitalised admissions frameworks.

VII. FUTURE WORK

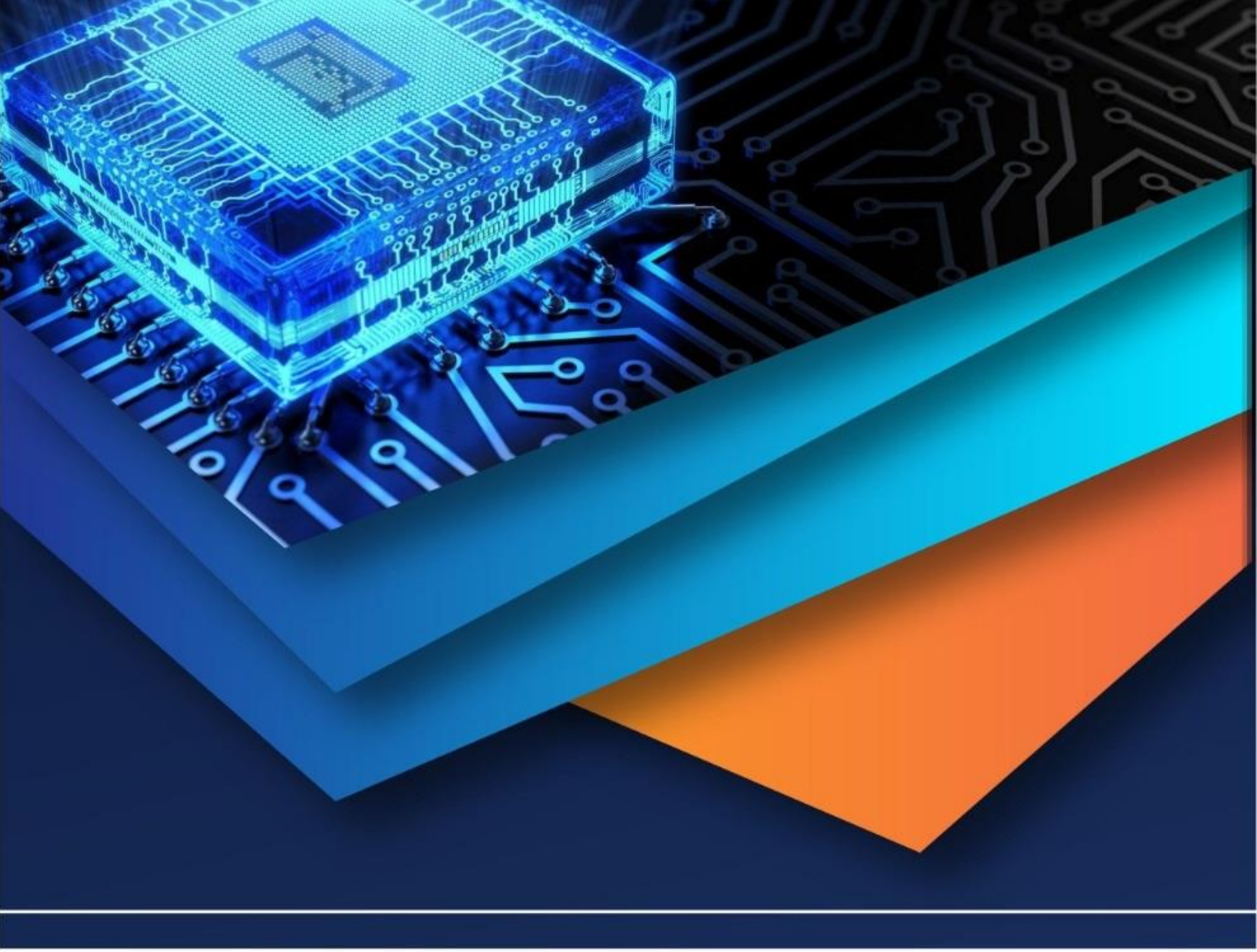
While the current implementation effectively automates fundamental academic admission functionalities, several prospective enhancements may further augment its applicability and operational resilience:

1) *Role-based Access Control (RBAC)*: Subsequent iterations might incorporate granular access management for administrative hierarchies (e.g., departmental, institutional authorisation levels).
2) *Mobile Application Integration:* Development of native or cross-platform mobile software would enhance accessibility for candidates in regions with limited connectivity.

3) *Multi-language Support*: Implementation of linguistic localisation capabilities would improve usability for applicants from varied linguistic contexts.
4) *Analytical Recommendation Mechanisms:* Incorporation of statistical modelling to propose suitable academic pathways based on applicant characteristics and historical matriculation patterns could constitute a valuable supplementary feature.
5) *Distributed Computing Implementation and Workload Distribution*:Deployment on distributed computing infrastructures with automatic scaling and request distribution mechanisms will enhance system availability and computational performance during periods of elevated user activity.

REFERENCES

[1] S. Sharma and P. Gupta, "A Web-Based Student Registration System Using PHP and MySQL," International Journal of Computer Applications, vol. 65, no. 21, pp. 1–5, 2013.
[2] M. Nuruzzaman and O. Hussain, "Intelligent Agent-Based Student Support through Chatbot," in Proc. IEEE 15th Int. Conf. e-Business Engineering (ICEBE), Xi'an, China, 2018, pp. 87–94.
[3] A. Kumar and S. Gupta, "Automated Admission Management System Using Web Technologies," International Journal of Computer Science and Mobile Computing, vol. 9, no. 5, pp. 10–16, 2020.
[4] A. Bansal, D. Batra, and P. Khurana, "Implementation of a Chatbot for Student Query Handling Using NLP and Machine Learning," in Proc. 5th Int. Conf. Comput. Commun. Autom. (ICCCA), 2019, pp. 1–6.
[5] V. K. Jain and P. Tiwari, "Merit-Based Admission Process Automation Using MongoDB and Node.js," International Journal of Advanced Research in Computer Science, vol. 11, no. 6, pp. 121–125, 2020.
[6] J. S. Sethi and A. S. Arora, "PDF Table Extraction Using Camelot and Tabula," International Journal of Computer Applications, vol. 183, no. 30, pp. 18–23, 2021.
[7] A. Kumar, P. Yadav, and M. Singh, "Enhancing Security in Web Applications Using OTP-Based Verification," International Journal of Scientific and Research Publications, vol. 9, no. 3, pp. 15–20, 2019.
[8] R. S. Kalbande and K. K. Wagh, "Automated Admission System Using MERN Stack for Higher Education," International Journal of Innovative Research in Computer and Communication Engineering, vol. 8, no. 7, pp. 180–186, 2020.
[9] M. Joshi and A. Shah, "Cloud-Based Online Admission System for Higher Education," in Proc. Int. Conf. Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 139–144.
[10] M. F. Choudhury and S. Saha, "Building Secure Web Applications: A Case Study of OTP and JWT Implementation," in Proc. Int. Conf. Emerging Trends in Computing and Communication Technologies (ICETCCT), 2022, pp. 42–47.
[11] D. Sudhakar and N. K. S. Kumar, "Dynamic Data Handling and Visualization Using MERN Stack," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 6, pp. 5012–5016, 2019.
[12] S. Patel and H. Mistry, "Use of MongoDB for Real-Time Web Applications," International Journal of Advanced Research in Computer Science, vol. 9, no. 3, pp. 212–215, 2018.
[13] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.
[14] T. Winata et al., "Attention-Based Recurrent Neural Networks for Automatic Chatbot Design," in Proc. Conf. Neural Information Processing (ICONIP), 2017, pp. 208–217.
[15] P. Jain and R. Sharma, "A Smart Admission System for Higher Education Using Modern Web Technologies," International Journal of Computer Sciences and Engineering, vol. 7, no. 5, pp. 45–50, 2019.
[16] S. Goyal and A. Verma, "Real-Time Web Application Development Using Node.js and MongoDB," International Journal of Computer Applications, vol. 172, no. 10, pp. 22–27, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY