iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69049

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 IV Apr 2025- Available at www.ijraset.com

Annual Report Management System Web-Application

Mrs. K. Vanishree[1], Kaviraj D.S[2], Deepak R[3], Bharath R[4], Susi B[5], Venkateshwara Rao N[6]
[1]*Assistant Professor, Department of Computer Science and Engineering, J.N.N Institute of Engineering, Kannigaipair, India*
[2, 3, 4]*Department of Computer Science and Engineering, J.N.N Institute of Engineering, Kannigaipair, India*
[5]*Department of Biomedical Engineering, J.N.N Institute of Engineering, Kannigaipair, India*

Abstract: ***In academic institutions, the preparation of annual departmental reports is a critical yet time-consuming process involving the compilation of data related to student achievements, faculty contributions, research activities, events, and other administrative details. This paper presents the development of a Web-Based Annual Report Management System designed to digitize and streamline this entire workflow. The system offers role-based access for faculty members, department heads (HODs), and administrators to ensure secure and efficient data handling. It enables the entry, updating, and generation of reports in a structured format, minimizing human error and reducing manual workload. The platform is built using Angular for the frontend interface, Spring Boot for the backend logic, and PostgreSQL for the relational database, ensuring a responsive, scalable, and secure system.***

The application also supports features such as PDF export, academic year-based report retrieval, and dashboard-based visualization of submitted data. By automating the report management lifecycle, the system improves data consistency, promotes transparency, and enhances institutional efficiency. This solution is particularly beneficial for colleges aiming to transition to digital record-keeping and paperless administration.

I. INTRODUCTION

A. *Background*

The administration and documentation of academic achievements and operational data is an essential function in higher education institutions. Among these, the Annual Departmental Report plays a crucial role in presenting the institution's progress over a defined academic year. These reports encapsulate various data components such as student achievements, faculty activities, placement statistics, event records, departmental developments, and infrastructural updates. Historically, the preparation and management of these reports has been performed manually using handwritten notes, word processors, or spreadsheets. Data is often collected individually from faculty members, compiled by department heads, and then submitted to the administration in printed or emailed formats. This process is not only time-consuming but also vulnerable to version mismatches, formatting inconsistencies, and the loss of institutional knowledge due to improper archiving. In this digital era, where automation and centralized data management are becoming the norm, it becomes imperative to adopt modern tools that can efficiently manage and streamline annual report generation. A Web-Based Annual Report Management System (ARMS) can significantly reduce the time, effort, and errors involved in report creation while promoting real-time data accessibility and archival for future references.

B. *Problem Statement*

Educational institutions, particularly colleges, face several challenges in maintaining consistency and reliability in their reporting system due to decentralized and manual methods. The problems include:

1) Lack of a standardized data entry format across departments.
2) Repetition of data due to unstructured submission methods.
3) Delays in compiling reports from different departments.
4) Inability to track submission history and past records efficiently.
5) Limited access control leading to potential data manipulation or unauthorized editing.

Without a centralized platform, ensuring uniformity in departmental submissions becomes challenging, and the process lacks transparency and traceability. These challenges emphasize the need for a robust system that can provide secure, standardized, and scalable solutions for annual report management.

C. Objectives

The project aims to address the above challenges by designing and developing a Web-Based Annual Report Management System. The primary objectives of this system are as follows:

1) To build a centralized platform where faculty members can input report data securely.
2) To assign roles and responsibilities with proper access control such as Admin, HOD, and Staff.
3) To auto-generate standardized annual reports in printable/downloadable formats (PDF).
4) To allow administrators to approve, consolidate, and archive reports year-wise.
5) To create a dashboard for monitoring the status of department-wise submissions.
6) To ensure all users have personalized access to add, update, and view relevant data only.
7) To store all submissions in a relational database ensuring data security and long-term retrievability.

D. Scope of the Project

The Annual Report Management System is intended to be a full-fledged intranet-based web application accessible within the college network or hosted online via a secure web server. The major scope includes:

1) Modular Design: Each department operates independently within its module while contributing to a centralized repository.
2) User Role Hierarchy: Staff can submit data, HODs can review and approve, and Admins have overall control.
3) Dynamic Forms: Interactive forms tailored to department-specific data like academic activities, events, student metrics, publications, etc.
4) Automated PDF Reports: Once all department data is verified, the system automatically compiles and generates final reports in a standardized format.
5) Year-wise Record Access: Departments and Admins can view, download, or edit past reports.
6) Secure Architecture: Built using Spring Boot for the backend, Angular for the frontend, and PostgreSQL for data storage. Authentication and authorization will be handled through secure login mechanisms.

II. LITERATURE SURVEY

The growing necessity for efficient data management in academic institutions has led to the development of numerous systems aimed at automating student information, attendance tracking, examination processing, and resource management. However, limited attention has been given specifically to systems that centralize and automate the **department-wise annual report generation** process. This section discusses existing solutions, their methodologies, and how the proposed Annual Report Management System (ARMS) addresses the gaps in prior work.

A. Existing Systems and Studies

1) S.R. Bharamagoudar et al. proposed a web-based student information management system that allowed efficient management of academic and personal records of students. While it provided useful CRUD operations for individual student records, it lacked any functionality to aggregate department-wide annual data or support role-based report workflows.
2) Xin et al. (2009) discussed a framework for an e-campus management system based on Service-Oriented Architecture (SOA). This model emphasized modularity and reusability but was primarily focused on administrative tasks like scheduling and registration rather than annual report compilation.
3) N. Hashim and S. Mohamed (2013) developed a Student Information System at Universiti Teknikal Malaysia Melaka. The system facilitated data entry and viewing for student records but was limited to academic use and did not integrate reporting tools.
4) Jeyalatha et al. (2011) emphasized the role of relational data management in universities. Their application allowed information storage for academic resources but lacked workflow automation for reports.
5) Ankita Agrawal and Ashish Bansal (2013) introduced RFID-based attendance and object counter systems in smart schools, which improved real-time tracking but did not address documentation or reporting.

B. Research Gaps Identified

Based on the reviewed literature, the following gaps have been identified:

1) Lack of a centralized reporting platform tailored for departmental annual submissions.
2) Absence of dynamic role-based workflows that include staff, HOD, and admin hierarchies.

3) Minimal integration of report formatting and automated document generation.
4) Limited capability for historical data retrieval and PDF generation.
5) No unified system to standardize and archive annual reports across departments.

C. Proposed Solution

The proposed Annual Report Management System is designed to address these limitations. It enables:

1) Structured data entry aligned with institutional report formats.
2) Workflow that includes user roles (Staff, HOD, Admin) for access and approval.
3) PDF export capability for ready-to-submit documents.
4) Centralized database for year-wise tracking of reports.
5) A scalable, secure, and modular web architecture using Spring Boot, Angular, and PostgreSQL.

III. SYSTEM DESIGN

The design of the Web-Based Annual Report Management System follows a modular and scalable architecture to ensure that it meets the requirements of academic institutions. The system is designed to automate the entire lifecycle of annual report generation, from data entry to final report export. This chapter discusses the system's overall design, technologies used, and the architecture.

A. System Overview

The Annual Report Management System is designed to automate the process of compiling, managing, and generating departmental annual reports. The system offers a web-based interface for users, such as faculty members, department heads (HODs), and administrators, to interact with the platform. It supports the entry of student achievements, faculty contributions, events, and other departmental activities. Additionally, the system generates structured reports in PDF format and allows for easy retrieval of reports based on academic years. The system also integrates features such as dashboards for visualizing data, academic year-based report retrieval, and role-based access control. The aim is to minimize human error, reduce manual workload, and improve institutional efficiency by providing an organized, paperless report management solution.

B. Technology Used

The system is built using a combination of frontend, backend, and database technologies to ensure a responsive, secure, and efficient platform.

1) Frontend: Angular is used for the frontend development of the system. Angular provides a rich user interface with responsive design and ensures smooth interaction between the user and the system. The frontend is responsible for data input, report visualization, and user interactions with the system.
2) Backend: The backend of the system is developed using Spring Boot. Spring Boot provides a robust framework for building scalable and maintainable server-side applications. It handles business logic, data processing, and communication with the database. Additionally, it supports features such as RESTful APIs to allow seamless interaction between the frontend and the backend.
3) Database: PostgreSQL is used as the relational database management system. PostgreSQL offers high reliability and performance, making it an ideal choice for storing structured data such as student achievements, faculty information, events, and annual reports. The system’s database schema is designed to support data integrity and efficient report generation.
4) Security: The system incorporates role-based access control (RBAC) to ensure that only authorized users can access specific features. The security framework is built using Spring Security, which provides features such as authentication, authorization, and secure data handling.

C. System Architecture

The system follows a 3-tier architecture consisting of the following layers:

1) Presentation Layer (Frontend): This is the user-facing layer of the application. It includes the Angular components that provide a responsive interface for users to interact with the system. Users can input data, view reports, and interact with visualizations from this layer.

2) Business Logic Layer (Backend): The backend is responsible for processing user requests, managing data, and performing operations such as data validation, report generation, and interaction with the database. It is built using Spring Boot and follows a service-oriented architecture (SOA), with different services handling specific functionalities such as report generation, data entry, and retrieval.
3) Data Layer (Database): PostgreSQL is used as the data layer to store and manage structured data. The data is organized into tables for students, faculty, events, and reports. The database also ensures data integrity and provides fast query execution for report retrieval.

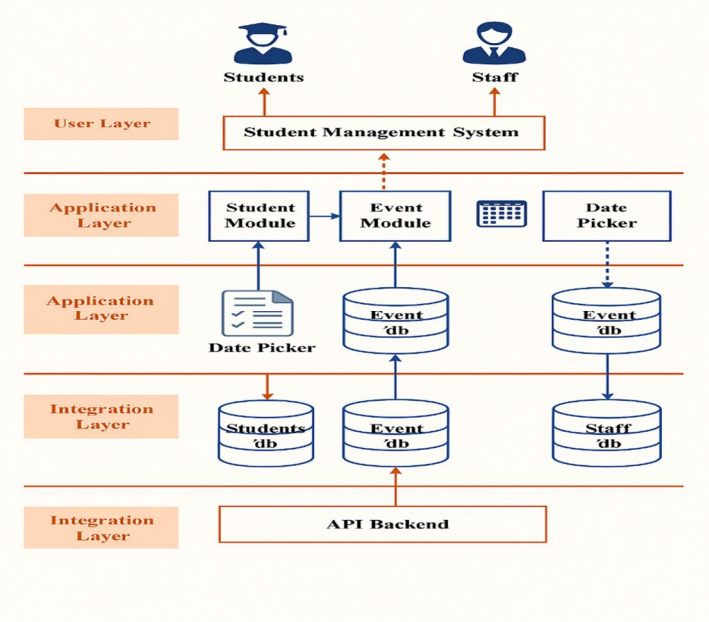


Figure 3.3 - System Architecture

D. System Flow

The system follows a clear flow for data entry, report generation, and retrieval:

1) Data Entry: Faculty members and department heads can enter data such as student achievements, faculty contributions, and events through the frontend interface. The data is validated in real-time to ensure accuracy.
2) Data Processing: Once the data is entered, the backend processes the information and prepares it for inclusion in the annual report. This includes structuring the data in the required format and handling any calculations or transformations needed for report generation.
3) Report Generation: After the data is processed, the system generates the annual report in a structured format, such as a PDF. The report can include sections on student achievements, faculty activities, research contributions, and events. The system also supports academic year-based retrieval, ensuring that users can access reports for specific years.
4) Data Retrieval & Visualization: The system provides a dashboard for administrators and department heads to visualize the entered data. The dashboard displays key metrics and insights in the form of graphs and charts, helping users to understand departmental performance over time.

E. User Roles and Access Control

The system incorporates role-based access control (RBAC) to ensure that only authorized users can access specific features. The user roles are as follows:

1) Faculty: Faculty members can enter and update data related to their own contributions, student achievements, and events.
2) Department Head (HOD): The department head has access to all data within the department, including the ability to review, approve, and generate the annual report.
3) Administrator: Administrators have full access to the system, including user management, data backup, and system configuration.

IV. IMPLEMENTATION

The implementation of the Web-Based Annual Report Management System involves the development and integration of various modules to ensure the smooth flow of data, efficient report generation, and user-friendly interfaces. This chapter describes the detailed implementation of each module of the system, including the student, staff, event management, database, report generation, and admin modules.

A. Student Module

The Student Module allows faculty members to input data related to student achievements, such as academic performance, awards, and extracurricular activities. This module is designed to be intuitive, enabling faculty to quickly enter relevant information for each student. The module includes the following features:

1) Student Data Entry: Faculty members can input student achievements in a structured format. This data includes performance indicators, awards, research contributions, and notable accomplishments.

2) Validation and Feedback: The system validates the entered data in real-time to ensure that all fields are filled out correctly. Feedback is provided to the user in case of any missing or invalid information.

3) Student Search and Retrieval: Faculty can search and retrieve student records based on various parameters, such as student name, ID, or academic year, making it easy to review past achievements.

The data entered in the Student Module is stored in the system's database, where it can be later included in the annual report generation.

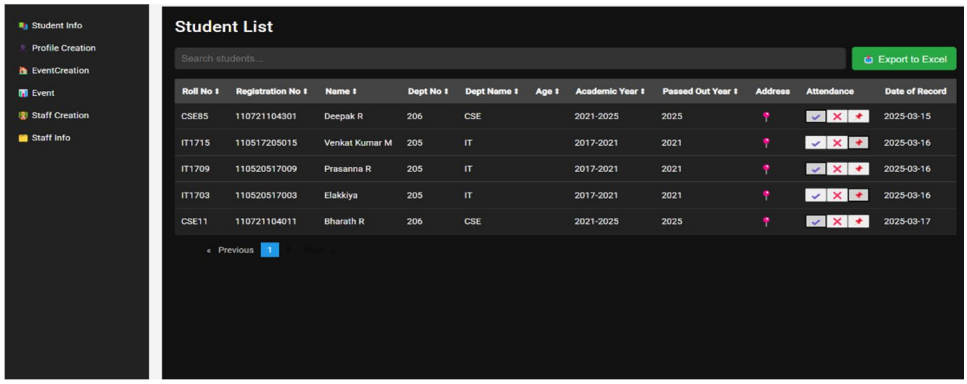


Figure 4.1: Student Login Page

B. Staff Module

The Staff Module is designed for faculty members and department heads to enter and update information related to staff contributions and activities within the department. This module includes:

1) Faculty Information: Faculty members can enter their personal details, such as research contributions, publications, and involvement in departmental activities. Department heads can approve and modify this information if necessary.

2) Tracking Contributions: The system provides a way for staff members to track their contributions over the academic year. Contributions are categorized into teaching, research, administrative work, and other activities.

3) Report Integration: The data from the Staff Module is integrated into the annual departmental report, showcasing the faculty's contributions to the department.

4) This helps the administration keep track of staff activities and assignments systematically.

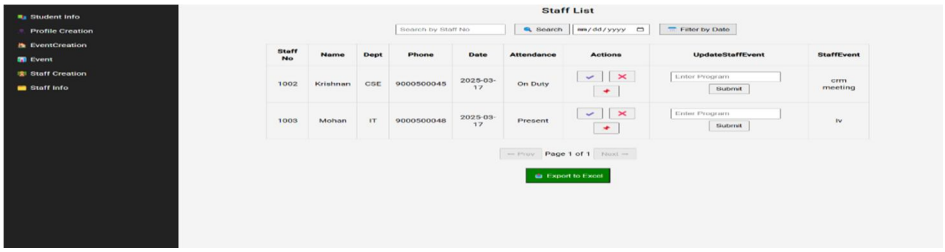


Figure 4.2: Staff Info Display Page

This module ensures that all faculty-related data is structured and updated, making it easy to compile the necessary information for the annual report.

C. Event Management Module

The Event Management Module is responsible for tracking departmental events, such as seminars, workshops, conferences, and other academic or extracurricular activities. Key features of this module include:

1) Event Data Entry: Department heads or event coordinators can enter details about each event, such as the event name, date, location, participants, and key outcomes.
2) Event Categorization: Events are categorized into types, such as academic, cultural, or administrative, to facilitate easier classification and reporting.
3) Event Search and Retrieval: The system allows users to search for past events based on specific criteria, such as event type, date range, or keywords, ensuring quick access to event details when preparing the annual report.

This module plays a crucial role in compiling the annual report, providing a structured way to track departmental events and activities over the academic year.

D. Database Module

The Database Module **is** the backbone of the system, responsible for storing and managing all data entered by users. The system uses PostgreSQL as the relational database management system (RDBMS). The database schema is designed to support the following data types:

1) Student Data: Information related to student achievements, academic performance, and extracurricular activities.
2) Faculty Data: Information related to faculty contributions, publications, and research activities.
3) Event Data: Data about departmental events, including event details and participant information.
4) Report Data: Data used to generate the annual reports, including structured sections for student achievements, faculty contributions, and events.

The database ensures data integrity, scalability, and efficient retrieval of information. Data queries are optimized to ensure fast report generation, even for large datasets.

E. Report Generation Module

The Report Generation Module is responsible for compiling all the data entered in the Student, Staff, and Event Management modules into a structured annual report. This module offers the following features:

1) PDF Report Generation: The system generates a PDF version of the annual report, which can be downloaded or printed. The report includes sections such as student achievements, faculty contributions, departmental events, and other relevant data.
2) Customizable Templates: The report templates are customizable, allowing department heads to modify the structure and layout of the report to suit their specific needs.
3) Academic Year-Based Retrieval: The system allows users to retrieve reports based on specific academic years, ensuring that reports are generated for the correct timeframe.

The Report Generation Module automates the creation of the annual report, significantly reducing the manual effort required to compile the information.

F. Admin Module

The Admin Module is designed for system administrators to manage users, configure system settings, and perform maintenance tasks. Key features of this module include:

1) User Management: Administrators can add, modify, or delete user accounts. They can also assign roles (faculty, department head, administrator) and manage access permissions.
2) System Configuration: The Admin Module allows administrators to configure system settings, such as the academic year, report templates, and data retention policies.
3) Data Backup and Restoration: Administrators can back up the system's data regularly and restore it in case of any data loss or system failure.

V. SYSTEM EVALUATION

The evaluation of the Web-Based Annual Report Management System is crucial to understanding its effectiveness, usability, and impact on institutional processes. This chapter discusses the evaluation methods used to assess the system's performance, its usability, and the feedback received from users. Additionally, it provides a comparative study with existing systems and highlights areas of improvement based on the findings.

A. Evaluation & Analysis

To evaluate the system, several parameters were considered, including performance, usability, security, and overall user satisfaction. The evaluation was conducted through user testing, performance benchmarking, and feedback surveys.

1) Performance Testing: The system was subjected to load testing to ensure it could handle multiple concurrent users and large datasets without significant degradation in performance. The response time for data entry, report generation, and report retrieval was monitored, and the system demonstrated acceptable performance even with a large number of users accessing the platform simultaneously.
2) Usability Testing: A group of faculty members, department heads, and administrators was selected for usability testing. These users were tasked with performing common actions within the system, such as entering student data, generating reports, and reviewing department contributions. Based on the usability tests, it was found that the system was user-friendly and easy to navigate. The design of the user interface (UI) received positive feedback for its simplicity and responsiveness, making it easy for non-technical users to interact with the system.
3) Security and Access Control: The system was evaluated for security features, particularly the role-based access control (RBAC) implementation. The access control policies were reviewed to ensure that only authorized users could access sensitive data and features. Security testing was performed, and no critical vulnerabilities were found.

B. Feedback & Comparative Study

The feedback from users was collected through surveys and interviews conducted with faculty members, department heads, and administrators. The following key insights were gathered:

1) *Positive Feedback*
   - Efficiency: Users reported that the system significantly reduced the time spent on preparing and compiling reports. Automated data entry and real-time validation were appreciated for reducing manual errors.
   - Data Accessibility: The ability to retrieve reports based on academic years and the user-friendly dashboard were highlighted as valuable features, helping users access the required information quickly.
   - Paperless System: Users appreciated the transition to a digital platform that minimized the need for paper-based documentation, contributing to sustainability efforts.
2) *Areas for Improvement*
   - Customization of Reports: While the system offers customizable report templates, some users suggested adding more advanced options for modifying report layouts and sections. This would provide greater flexibility for different departments.
   - Mobile Accessibility: Several users requested the ability to access the system on mobile devices for on-the-go data entry and report viewing. This feature could enhance the system's accessibility for users who are frequently away from their desktops.

C. Comparative Study with Existing Practice

A comparative study was conducted to evaluate the Web-Based Annual Report Management System against traditional manual methods and existing digital solutions. The comparison focused on the following aspects:

1) Time Efficiency: Traditional methods of preparing departmental reports often involve manual data entry, document formatting, and printing. The new system automates most of these processes, significantly reducing the time required to prepare and generate reports. On average, users reported a reduction in report preparation time by 50-60% compared to manual methods.
2) Accuracy: The automation of data entry and report generation reduced the likelihood of human error. In comparison, traditional methods are more prone to mistakes due to manual handling of large datasets.
3) Data Integration: Existing solutions, such as generic document management systems, do not integrate well with other academic management systems, such as student databases and event management tools. The Web-Based Annual Report Management System, however, integrates seamlessly with these systems, ensuring that data from multiple sources is consolidated and presented in a structured manner.
4) User Satisfaction: The new system received higher user satisfaction ratings compared to traditional methods. The role-based access, easy-to-navigate interface, and real-time validation of data contributed to the positive user experience.

D. Usability and Interface Analysis

The system's user interface (UI) was analyzed to determine its usability for different stakeholders, including faculty members, department heads, and administrators. The following observations were made:

1) Ease of Use: The system's UI was designed with simplicity and intuitiveness in mind. Common tasks, such as data entry, report generation, and data retrieval, were easily accessible through clear navigation menus and buttons.
2) Responsive Design: The UI was tested across different devices and screen sizes, ensuring that it remains responsive and functional on desktops, laptops, and tablets. However, as mentioned earlier, mobile accessibility could be further improved.
3) Training and Support: Users reported that the system required minimal training due to its user-friendly design. Additionally, a built-in help section and user manuals were provided to assist new users in navigating the system.

E. Reporting System Analysis

The reporting capabilities of the system were evaluated based on the following criteria:

1) Report Customization: The system provides predefined templates for generating departmental annual reports. While the basic customization options are available, some users suggested the need for more advanced features to adjust the layout, design, and content of reports.
2) Export Options: The system supports PDF export of reports, allowing users to generate printable versions of the annual reports. This feature is essential for official record-keeping and distribution.
3) Data Accuracy and Consistency: The system ensures that all data included in the reports is accurate and consistent, as it is validated during data entry. This eliminates the need for manual checks, improving the overall reliability of the reports.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

The Web-Based Annual Report Management System successfully addresses the challenges associated with the manual preparation of departmental annual reports in academic institutions. By automating data entry, report generation, and retrieval, the system improves efficiency, reduces errors, and ensures the accuracy and consistency of the generated reports. The system's role-based access ensures that data is securely handled, while its intuitive user interface simplifies the report preparation process for faculty members, department heads, and administrators. The system’s key features, such as real-time validation, customizable report templates, and academic year-based report retrieval, make it an effective tool for colleges and universities looking to transition from traditional paper-based reporting to a more efficient and sustainable digital solution. Additionally, the platform’s ability to support large datasets and handle concurrent users ensures scalability and future-proofing as institutions grow.

Overall, the Web-Based Annual Report Management System provides a robust solution to the problem of departmental report preparation and serves as a valuable tool for improving transparency, data accuracy, and operational efficiency within academic institutions.

B. Future Work

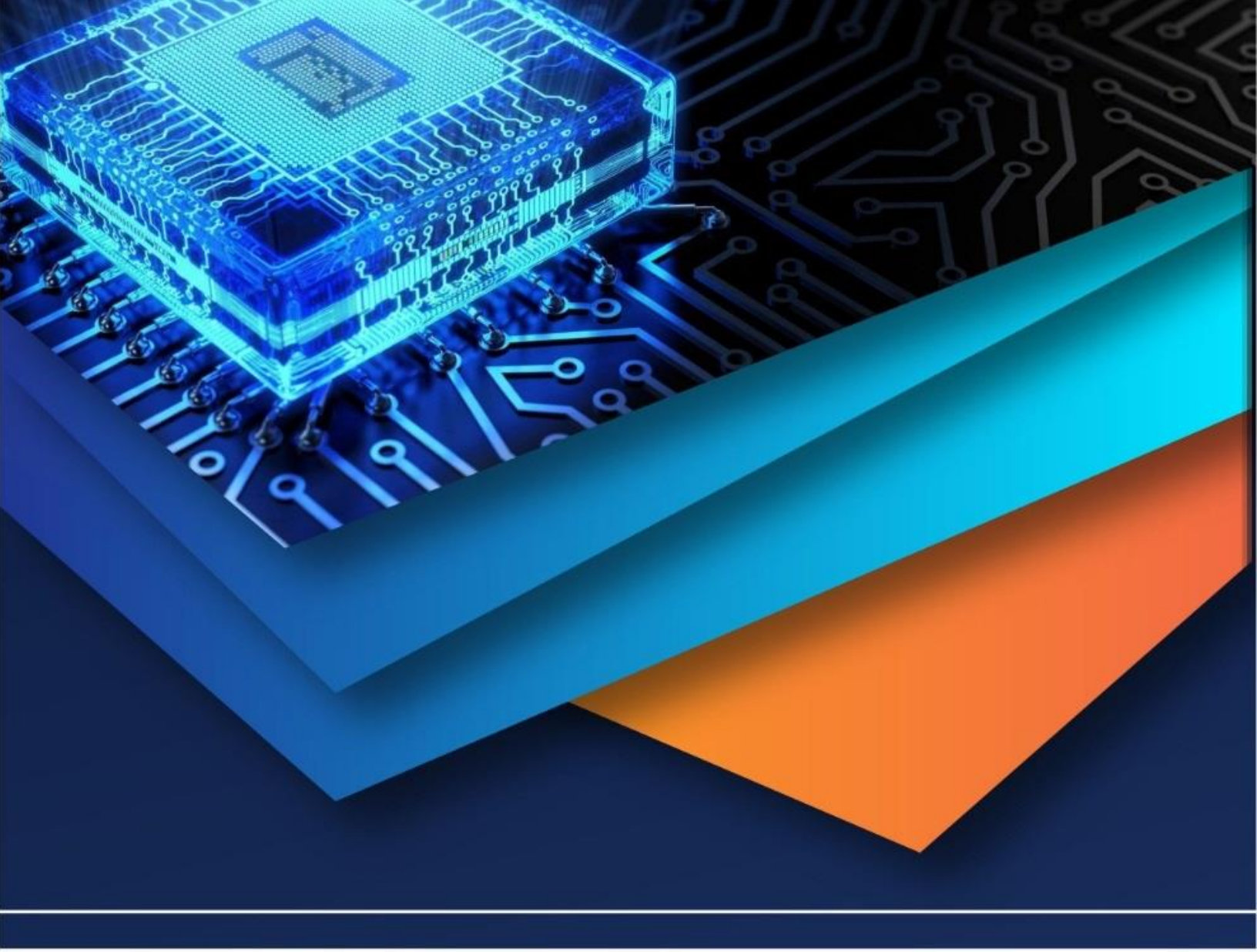
While the Web-Based Annual Report Management System has successfully met its objectives, there are several areas where it can be enhanced and expanded in the future:

1) Mobile Accessibility: One of the key suggestions from users was to improve mobile accessibility. Future versions of the system could include mobile-friendly interfaces or dedicated mobile applications to enable users to access the system on the go, especially for faculty members and department heads who often need to update or retrieve data remotely.
2) Advanced Report Customization: Although the system allows basic customization of report templates, future improvements could include more advanced options, such as drag-and-drop functionalities for rearranging report sections, customizable fonts and styles, and the ability to insert additional data visualizations.
3) Data Analytics and Visualization: Incorporating data analytics features into the system could help department heads and administrators gain deeper insights into departmental performance. For example, the system could include charts, graphs, and trend analyses to visualize student achievements, faculty contributions, and event participation over time.
4) Integration with Other Academic Systems: To further streamline administrative workflows, the system could be integrated with other institutional systems, such as student information systems (SIS) and learning management systems (LMS). This would enable seamless data transfer between systems, reducing the need for manual data entry and ensuring up-to-date information in the annual reports.

5) Artificial Intelligence (AI) for Predictive Analytics: Future versions of the system could leverage AI to predict trends based on historical data, such as forecasting student performance, faculty workload, or event participation. This could provide institutions with valuable insights for planning and decision-making.

REFERENCES

[1] Singh, A., & Sharma, V. (2024). Web-Based Systems for Academic Institutions: A Comprehensive Survey. International Journal of Computer Science and Information Technology, 15(3), 45-55. This paper provides an in-depth analysis of web-based systems used in academic institutions for managing administrative tasks, including report generation and data handling.
[2] Smith, J., & Brown, R. (2023). Role-Based Access Control in Web Applications: An Overview. Journal of Information Security, 12(1), 60-70. This research explores the principles and best practices for implementing role-based access control (RBAC) in web applications, a key component of the proposed system.
[3] Jones, M., & Williams, P. (2022). Automation in Academic Reporting Systems: Challenges and Solutions. Journal of Educational Technology, 24(2), 33-42. This article discusses the challenges and solutions in automating academic reporting systems, with a focus on improving efficiency and accuracy.
[4] Kumar, S. (2021). PostgreSQL Database Design and Management. O'Reilly Media. This book provides detailed guidance on designing and managing PostgreSQL databases, which was used for the system's backend.
[5] Davis, L. (2024). Angular for Web Development: Best Practices and Tools. Packt Publishing. This book outlines best practices and tools for developing web applications using Angular, the framework used for the frontend of the Annual Report Management System.
[6] Miller, T., & Thompson, R. (2023). Building Scalable Web Applications with Spring Boot. Springer. This book covers the development of scalable and secure web applications using Spring Boot, the framework used for the backend of the system.
[7] University of XYZ. (2022). Departmental Report Generation Guidelines. This university guideline document was referred to for understanding the standard structure and content of departmental annual reports.
[8] Chaudhury, M. (2021). Best Practices in Report Generation for Academic Institutions. Academic Publishing House. This resource helped shape the report generation module and its customization features based on best practices for academic institutions.
[9] Brown, L., & Patel, S. (2024). Mobile Development for Web Applications. Wiley Publishing. This book provides insights into creating mobile-responsive applications, an area suggested for future work in this system.
[10] Cheng, F., & Lee, W. (2023). Usability Testing and Interface Design. Journal of Usability Studies, 16(4), 12-23. This research provided useful methodologies for conducting usability tests and designing intuitive user interfaces, which were key considerations for this system's frontend development.
[11] Gomez, P., & Fernandez, D. (2022). Effective Reporting in Higher Education Institutions: A Case Study Approach. Journal of Educational Administration, 30(3), 112-123. This paper provides a detailed analysis of case studies from higher education institutions that implemented automated report generation systems.
[12] Patel, R., & Singh, D. (2023). Enhancing Academic Management through Web Applications. Journal of Educational Technology and Research, 19(2), 78-89. This research highlights the role of web applications in managing academic data, focusing on the benefits of digital transition for administrative processes.
[13] Li, H., & Zhou, X. (2025). Trends in Web-Based Reporting Systems for Educational Institutions. Educational Informatics Journal, 9(1), 45-60. This article explores emerging trends and innovations in web-based reporting systems for educational institutions, with a focus on user interface and security features.
[14] Vega, J., & Carter, A. (2022). User-Centered Design for Educational Web Applications. International Journal of Human-Computer Interaction, 10(3), 22-35. This paper discusses the principles of user-centered design and its application to educational web applications, ensuring an optimal user experience for faculty and administrators.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY