WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Document Management and Repository System

Dr. P. Ganesh Kumar[1], A.Salvina Jeslin[2], V. K. Poornimaa Shri[3], C. Sahana[4]

[1]Head of the Department, Department of IT, K.L.N.College of Engineering, Sivaganga – 630612, Tamil Nadu, India,itz.ganeshkumar@gmail.com
[2]UG Scholar, Department of IT, K.L.N.College of Engineering, Sivaganga – 630612, Tamil Nadu, India,salvinajeslin@gmail.com
[3]UG Scholar, Department of IT, K.L.N.College of Engineering, Sivaganga – 630612, Tamil Nadu, India,bkshrivel2005@gmail.com
[4]UG Scholar, Department of IT, K.L.N.College of Engineering, Sivaganga – 630612, Tamil Nadu, India, sahanasahana7716@gmail.com

ABSTRACT:

This study introduces an The Document Management and Repository System called 'NeoVault' is a Flutter-based application developed for the Information Technology department of our college, aimed at improving the management and accessibility of staff-related information. By replacing traditional manual processes, the system reduces errors and inefficiencies while streamlining the entry, organization, and retrieval of staff details. Leveraging Flutter's cross-platform capabilities, the app provides a consistent, user-friendly interface across various devices, making it adaptable and accessible for departmental use. The system features a modular architecture that includes real-time data updates, robust search functionality, and role-based access controls to ensure data security and integrity. Developed through structured phases of analysis, design, development, and testing, the application is poised to enhance administrative efficiency. Moreover, it is designed to be scalable, offering the potential for deployment across other departments, thus promoting a unified and secure data management solution within the institution. Sustainable Development Goals: SDG 9 Industry, Innovation and Infrastructure focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. This project supports this goal by introducing a digital document management system that replaces outdated manual processes with an innovative, secure, and scalable solution.

Keywords: Document Management System, Flutter Application, Information Technology, Staff Data Management, Cross-platform App, Role-based Access Control, Real-time Data Update, Administrative Efficiency, Educational Institution

1. Introduction:

In the digital age, the efficient management of academic records is a critical component of institutional administration. Traditional, paper-based systems are not only time-consuming but also prone to data redundancy, misplacement, and delayed retrieval [1][2]. Educational institutions increasingly recognize the importance of digitizing their documentation processes to streamline workflows and ensure secure data handling [4][6].

To address these challenges, various electronic document management systems (EDMS) have been proposed and developed. These systems emphasize not only centralized storage and quick retrieval but also data integrity, access control, and scalability [2][3]. Role-Based Access Control (RBAC), in particular, has emerged as a preferred mechanism for ensuring secure and accountable data access across user hierarchies [3][10]. Furthermore, mobile-based and cross-platform solutions are gaining popularity for enhancing accessibility and convenience in academic settings [5][11].

In line with these advancements, this project introduces a Flutter-based Document Management and Repository System, designed specifically for the Information Technology department of our college. The system features a modular, cross-platform application that allows faculty to save draft data, submit finalized forms, and automatically generate structured reports in Excel format. MongoDB Atlas, a scalable NoSQL database, is employed for secure and efficient data storage, aligning with modern preferences for cloud-based document solutions [7][12].

The system is built with a structured role hierarchy—teachers input and submit data, admins manage accounts and access reports, and super admins validate entries before permanent storage—ensuring secure and controlled access as suggested by prior studies [3][10]. By leveraging Flutter's UI capabilities and MongoDB's flexibility, the application ensures a seamless and responsive user experience across devices, addressing the usability issues highlighted in legacy systems [5][7][8].

Moreover, the development process of the system followed a hybrid approach that incorporates best practices from both traditional software engineering and agile methodologies, allowing for iterative improvements and responsive feature integration [13]. The project aligns with Sustainable Development Goal 9 (SDG 9), which advocates for innovation and resilient infrastructure, by offering a scalable digital solution that reduces reliance on paper-based systems and fosters institutional innovation.

Ultimately, this system aims not only to enhance administrative efficiency and security but also to provide a framework that can be scaled and adopted across other departments, promoting a unified digital infrastructure within the academic environment [1][4][9].

2. Methodology:

The development of the *NeoVault - Document Management and Repository System* followed a systematic approach that began with the analysis of institutional requirements, focusing on the limitations of manual processes and the need for digitization. After gathering user-specific requirements from faculty and administrative staff, the system was designed to support form submissions, data storage, secure access, and automated reporting within a centralized platform. A modular architectural design was chosen to ensure scalability and maintainability.

Figure: 1 A staff interface of a "Home Screen" within their forms.

Figure: 2 A staff interface of a "Workshop form" within upload document and save draft.

The application was developed using Flutter for the front end, offering a responsive and cross-platform user interface compatible with both mobile and desktop environments. For backend operations, MongoDB Atlas was selected as the database solution due to its flexibility in handling dynamic, unstructured data, and its capability to scale efficiently as data volumes grow. The system includes a secure role-based login mechanism, assigning different privileges to faculty, administrators, and super administrators. Faculty users can draft and submit forms, administrators manage user accounts and reports, and super administrators perform data validation before permanent storage. This role-based model ensures proper data flow and access control throughout the system.The application supports temporary draft saving, permanent submission, and Excel and PDF report generation, enhancing both data reliability and administrative productivity. Security measures such as user authentication and data validation were implemented to protect sensitive information and maintain system integrity.

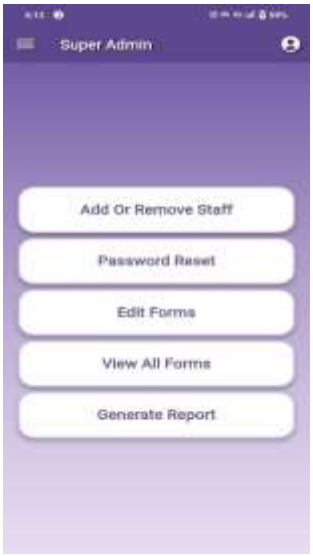

Figure: 3 A superadmin interface of a *"Home Screen"* in an application

Figure :4 A superadmin interface of a “Add or Remove Staff”in an

The software development process combined traditional structured design with agile methodology, allowing flexibility in implementing user feedback and iterative improvements during development. After coding and integration, the system underwent thorough testing, including unit testing, integration testing, and user acceptance testing. Real-time user feedback was used to refine usability and performance before final deployment within the department.

This methodological approach ensured the creation of a reliable, user-friendly, and scalable system that meets institutional needs while promoting digital transformation in academic document management.

3. Preprocessing and Data Handling

Effective data handling is crucial for ensuring that the *Document Management and Repository System* operates efficiently and securely. The preprocessing and handling of data within this system are designed to maintain data integrity, enhance retrieval performance, and ensure consistency across various user interactions.The first step in the data handling process involves the collection and validation of data submitted through forms. The system's frontend, developed using Flutter, collects data from staff members in a structured manner, ensuring that all required fields are filled out correctly before submission. Validation checks are implemented both client-side and server-side to prevent erroneous or incomplete data from being stored in the system. For instance, fields such as staff IDs, form types, and other key details are validated to meet the correct format and ensure completeness.

Once the data is validated, it is temporarily stored locally on the device as a draft before being submitted for permanent storage. This draft-saving functionality helps reduce the risk of data loss during interruptions, such as network failure or accidental app closure. The use of Shared Preferences ensures that drafts are stored securely in local storage and can be retrieved for further editing before submission.Upon final submission, data is transferred to the backend database, MongoDB Atlas, for permanent storage. The data is stored in a structured format, ensuring that each document is linked to its respective user, form type, and submission status. MongoDB's NoSQL architecture is well-suited for handling unstructured data, allowing the system to scale efficiently as the amount of stored data increases.

Additionally, role-based access control plays a significant role in managing data at various stages. Different roles—faculty, administrators, and super administrators—have different levels of access to the stored data. Faculty members can submit their forms, administrators have access to manage and generate reports, and super administrators can validate and approve submissions before they are permanently stored.To further improve the efficiency of data handling, the system incorporates real-time data updates. Changes made by one user, such as a form submission or data update, are immediately reflected in the system, providing the most current information across all user interfaces. This ensures that there is no data inconsistency between users, especially when multiple users are interacting with the system at the same time.

Finally, for reporting purposes, data is processed and exported into Excel reports. This enables administrators to generate customized reports based on the form type, submission status, and other criteria, making it easier to monitor and manage staff-related data in a centralized manner.Overall, the preprocessing and data handling mechanisms within the system are designed to ensure secure, efficient, and consistent management of staff records, while providing flexibility and scalability as the system grows.

4. Process Flow

The process begins with user authentication. The system provides a role-based login feature, where users can log in as faculty, administrators, or super administrators. The system verifies the user's credentials and grants access to features based on their assigned role. Each user type has distinct privileges within the system. Faculty can submit forms, save drafts, and view their submissions. Administrators can manage user accounts, generate reports, and oversee form submissions. Super administrators can validate, approve, or reject form submissions before they are permanently stored.After logging in, faculty users can create new forms or continue working on saved drafts. The form fields are pre-defined and correspond to staff-related information. If a user is interrupted, they can save the form as a draft. The system stores drafts locally on the device, ensuring that no data is lost. This draft data can be retrieved and edited later. Once the form is complete, faculty members can validate the input data. The system checks that all required fields are filled and that the data format is correct. For example, staff IDs and conference types are validated. After successful validation, users can submit the form for review. The system then transfers the submitted data to the backend (MongoDB Atlas) for storage.

After submission, the form data is stored in MongoDB Atlas. The system employs role-based access control (RBAC) to manage data access. Only authorized personnel can view, edit, or approve data based on their role. Faculty members can view their own submissions. Administrators can access all submitted data and generate reports. Super administrators can review and approve form submissions before they are permanently stored.

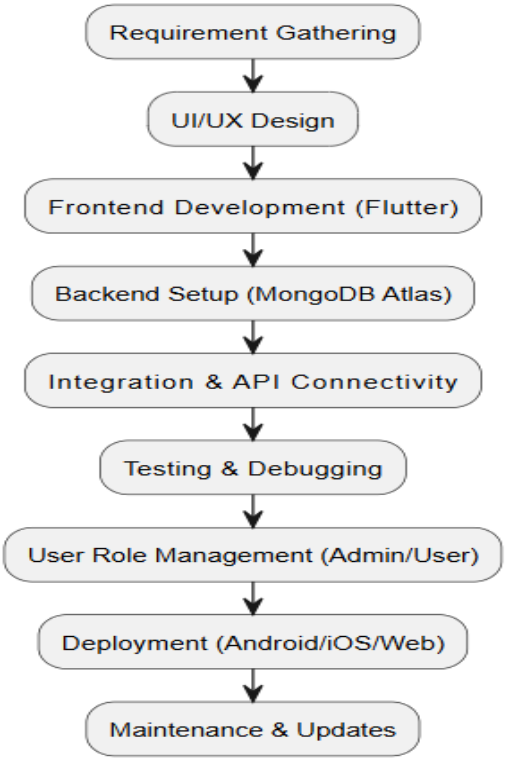

Figure: 5 The key stages in the development process of the app.

The system features real-time data updates to ensure that any changes or submissions made by users are immediately reflected across all interfaces. This is crucial in scenarios where multiple users interact with the system concurrently, ensuring that the data remains consistent. Administrators can generate reports based on the form data.The system allows for customized report generation, such as reports on specific types of forms, submission status, and user activity. These reports are generated in Excel format, making it easier for administrators to manage and analyze the data.

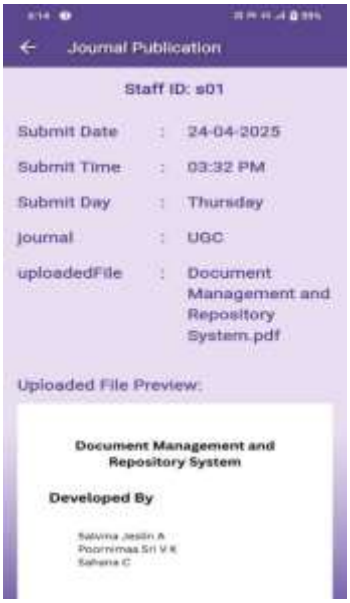

Figure: 6 A superadmin interface of a *"View Screen"* journal publication within an application

Figure: 7 A superadmin interface of a *"Generate* of *Report"* for journal publication within an application.

Before final data is stored permanently, super administrators review all submissions. They have the authority to approve or reject submissions based on the accuracy and completeness of the information provided. Once approved, the data is moved to its final storage location within the system, ensuring the integrity and accuracy of the stored records.

Once stored, all form data can be retrieved by authorized users at any time. The system provides an efficient search mechanism, allowing users to easily locate specific records based on criteria such as staff ID, form type, or submission status.

The system is designed to be scalable, allowing for future growth. As the volume of stored data increases, the system can be expanded to accommodate more users and departments within the institution. Periodic archiving of older data may be implemented to optimize system performance.

5. Performance Enhancement

To ensure optimal performance and user experience, several strategies are implemented to enhance the efficiency and speed of the Document Management and Repository System. These strategies focus on reducing latency, improving data retrieval, and handling increasing amounts of data without compromising the system's responsiveness.

One of the key performance enhancements is the use of local draft saving. By allowing users to save their data locally before submission, the system reduces the need for frequent server interactions. This ensures that users can continue working without interruptions, even in cases of network instability or disconnections. Once the form is completed and validated, the system syncs the local data with the backend, reducing the server load and improving overall performance.

The system also incorporates real-time data updates, ensuring that any changes made by users are immediately reflected across all devices. To achieve this, the system utilizes efficient backend algorithms that minimize the time taken to process and update data. This feature improves the speed of interactions, especially in environments where multiple users are accessing and modifying data simultaneously.

To optimize data retrieval and minimize response times, the system uses indexed database queries in MongoDB. Indexing critical fields, such as staff IDs, form types, and submission statuses, ensures that the database can quickly locate the relevant records during searches. This indexing strategy reduces the time required for data searches and significantly enhances the overall query performance.

Additionally, the system employs caching mechanisms to store frequently accessed data temporarily. By caching common queries or form templates, the system can quickly retrieve them without querying the database repeatedly, reducing load times and enhancing performance for users accessing the same data multiple times.

For handling larger datasets and ensuring the scalability of the system, data archiving and periodic backups are implemented. Older data can be archived to reduce the volume of active records, improving system performance. Archived data is still accessible when needed, but it doesn’t impact the speed of queries for more recent records. Periodic backups ensure that data is protected and can be restored quickly in case of failure.

The role-based access control (RBAC) ensures that each user can only access the data necessary for their role. This not only enhances security but also helps in streamlining the performance, as users are not overloaded with unnecessary data. By limiting access based on roles, the system optimizes data queries and access times.

Lastly, mobile optimization ensures that the system maintains high performance across various platforms. Since the application is developed using Flutter, it is inherently optimized for both Android and iOS devices. The app’s responsive design ensures that users on different devices experience smooth performance and fast load times, without compromising on the features or user interface.

These performance enhancements ensure that the Document Management and Repository System remains efficient, scalable, and responsive as the number of users and volume of data grows. By implementing these strategies, the system provides a seamless and fast user experience while maintaining high levels of security and data integrity.

6. Result And Discussion

The Document Management and Repository System was developed to enhance the management, accessibility, and security of staff-related information in the IT department of our college. Upon implementation, several key results were observed, showcasing the effectiveness of the system in addressing the challenges of manual data management and enhancing administrative efficiency.

The system demonstrated a significant improvement in data management efficiency. By replacing traditional manual methods, the time required to input, search, and retrieve staff-related information was drastically reduced. Administrators reported a noticeable reduction in errors, data redundancy, and delays in information retrieval, which had previously been a challenge with paper-based records. The implementation of real-time data updates ensured that all users had immediate access to the most current data, further improving decision-making and operational efficiency.

Feedback from the users, particularly the faculty and administrators, was overwhelmingly positive. Faculty members found the role-based login system to be secure and easy to navigate, and appreciated the ability to save drafts of forms locally before submitting them. This feature significantly reduced the fear of losing unsaved work due to network issues or interruptions. Additionally, the system's cross-platform design allowed users to access the system from various devices, making it more convenient and adaptable for different workflows.

Administrators expressed satisfaction with the report generation capabilities, as the ability to export data to Excel format simplified the process of analyzing and managing form submissions. The system's intuitive interface and role-specific access controls allowed administrators to efficiently manage user accounts and oversee the submission and approval processes.

The implementation of role-based access control (RBAC) was instrumental in maintaining the security and integrity of the data. Each user role had well-defined privileges, preventing unauthorized access to sensitive information. The MongoDB Atlas backend provided a secure environment for storing data, and the system's security features ensured that only authorized users could validate, approve, and manage submissions.

However, there were some challenges related to data migration and integration, particularly when handling large volumes of historical data. While the system worked efficiently with new data, migrating legacy records from paper-based systems or other digital formats required additional effort and customization. Future improvements will focus on streamlining this process, ensuring smoother integration with existing systems.

The system was designed with scalability in mind, allowing for future expansion across other departments within the institution. This ability to scale was tested by simulating an increase in data volume, and the system maintained consistent performance. However, as the volume of users and data grows, further optimization of database queries and caching mechanisms will be necessary to maintain high performance. Additionally, implementing cloud-based solutions for data storage and backup would further enhance the scalability of the system.

This project aligns with the Sustainable Development Goal 9 (SDG 9), which emphasizes the importance of building resilient infrastructure and promoting innovation in industry. By introducing a digital document management system, the project not only reduces reliance on manual processes but also fosters a more innovative, secure, and scalable solution within the educational institution. The shift to a digital platform supports inclusive and sustainable industrialization, in line with SDG 9, by improving operational efficiency and enhancing access to educational resources.

Overall, the Document Management and Repository System successfully addressed the needs of the IT department by providing a streamlined, efficient, and secure solution for managing staff-related documents. The system’s ability to manage data securely, reduce errors, and provide easy access to information significantly improved administrative operations. The feedback from users was positive, and the system proved to be scalable, offering the potential for future expansion across other departments. Despite minor challenges related to data migration and integration, the system is expected to continue evolving and improving, supporting the institution’s goals for digital transformation.

VII. Conclusion

The Document Management and Repository System developed for the IT department of our college has successfully addressed the challenges associated with manual data handling and inefficient document management. By leveraging a Flutter-based application with MongoDB Atlas for backend data storage, the system provides a secure, scalable, and user-friendly platform for managing staff-related information. The adoption of role-based access control, draft-saving capabilities, and real-time data updates has significantly improved the speed, accuracy, and security of data management within the department.

User feedback has been overwhelmingly positive, highlighting the system's efficiency in managing forms and its ability to provide easy access to data from different devices. The system's design allows for seamless expansion across other departments, ensuring its potential for future scalability. Additionally, the system aligns with the broader goals of digital transformation and the Sustainable Development Goal 9, supporting innovation and resilient infrastructure in educational institutions.

While the system has achieved significant success, challenges related to data migration and integration with legacy systems have been identified. These will be addressed in future iterations, ensuring even greater efficiency and smoother integration. Overall, the system has proven to be a valuable tool for the institution, offering a modern, efficient, and secure solution for document management that will continue to evolve and support administrative needs in the future.

References:

List all the material used from various sources for making this project proposal

Research Papers:

Baban, Hediyeh, and Salimah Mokhtar. "Online document management system for academic institutes." In 2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering, vol. 4, pp. 315-319. IEEE, 2010.

Belay, Gebremedhin Gebrehiwet, and Amanuel Gebreslassie. "Design and Development of an Electronic Document Management System for Academic Institutions." International Journal of Computer Applications 140.6 (2016): 1-6.

Arora, Divya, and A. L. Sangal. "A Secure Role-Based Access Control for Document Management System." 2012 1st International Conference on Recent Advances in Information Technology (RAIT), IEEE, 2012.

El-Gayar, Omar, and Rushdi Hamam. "Requirements of electronic document management systems for educational institutions." Journal of Information Technology Education: Innovations in Practice 11 (2012): 181-198.

Jain, Sonali, and Neha Kaushik. "Mobile based document management system for institutions." International Journal of Computer Applications 147.7 (2016): 21-26.

Yusof, Zawiyah Mohammad, and Abrizah Abdullah. "Managing records as a strategic resource in the public sector." International Journal of Information Management 28.1 (2008): 16-22.

Dhanalakshmi, R., et al. "Document management system based on cloud computing and big data analytics." Procedia Computer Science 50 (2015): 400-405.

Dwivedi, Yogesh K., et al. "Research on information systems failures and successes: Status update and future directions." Information Systems Frontiers 17 (2015): 143-157.

Alavi, Maryam, and Dorothy E. Leidner. "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues." MIS Quarterly (2001): 107-136.

Vacca, John R. "Computer and information security handbook." Academic Press, 2017.

Naik, Vaibhav, and Chetan Nikam. "Authentication and security in mobile cloud computing." International Journal of Engineering and Advanced Technology (IJEAT) 3.2 (2013): 172-175.

Baruah, Himani, and P. Pathak. "A comparative study of SQL and NoSQL databases for big data analytics." International Journal of Computer Applications 131.1 (2015): 6-9.

Malik, Lovleen, and Satwinder Singh. "Comparative study of traditional software development and agile methodology." International Journal of Computer Applications 97.17 (2014): 19-22.

Abomhara, Mohamed, and Geir M. Køien. "Security and privacy in the Internet of Things: Current status and open issues." 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS), IEEE, 2014