DOCUMENT MANAGEMENT SYSTEM

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science and Engineering)

Submitted By:

Golu Kumar (2104099)

Hare Krishna (2104109)

Swagat (2203598)

Submitted To:

Er. *Satinderpal Singh*

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE

LUDHIANA, 141006

May, 2025

ABSTRACT

In the current knowledge-based era, organizations are flooded with enormous amount of documents and data that need to be managed efficiently. A Document Management System (DMS) is a key system that organizes, stores, secures, and retrieves electronic documents in a systematic and efficient way. This project describes the design, development, and implementation of a scalable and robust DMS to suit the requirements of contemporary businesses, educational institutions, and government agencies. The main goal is to remove the inefficiencies and security threats that come with manual or paper document handling by offering an automated, centralized digital repository.

The system suggested supports the orderly storage of documents with metadata tagging, versioning, indexing, and full-text search capabilities. It enables users to upload, modify, and retrieve documents based on user permission and access rights. Role-based access control prevents sensitive documents from being accessed by unauthorized users, thus improving data security and regulatory compliance with requirements like GDPR, HIPAA, or ISO standards. The system also records document histories and keeps audit trails in order to promote transparency and accountability in document usage and alteration.

The DMS architecture follows a modular and service-oriented approach, enabling easy integration with existing enterprise systems such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) platforms. It supports both on-premises deployment and cloud-based hosting to accommodate different organizational needs. A responsive web interface ensures that users can access the system from various devices, including desktops, tablets, and smartphones, promoting remote collaboration and operational flexibility.

Major features of the system are automated document classification, support for OCR (Optical Character Recognition) of scanned documents, batch upload, and real-time alerts for document updates and approvals. A built-in advanced search engine in the system allows users to find documents rapidly based on keywords, metadata fields, or document content. The DMS also includes backup and recovery features to avoid data loss and ensure business continuity.

Technically, the system is created using contemporary web technologies and frameworks for high performance and scalability. Backend services are designed on a secure and highly efficient database model optimized for document storage and retrieval operations. A focus is on user experience, with a neat and simple interface reducing the learning curve for the end-users.

With the application of this Document Management System, organizations are able to achieve valuable benefits such as lower operational expenses, quicker document access, increased collaboration, enhanced security, and increased compliance with legal and regulatory requirements. Not only does the system facilitate the simplification of internal processes but also promotes environmental by eliminating the need for paper-based documents.

This abstract calls out the strategic significance of document management in today's digital age and offers the suggested DMS as a full-fledged solution to the increasing need for secure, efficient, and convenient document handling systems. Future updates could involve AI-powered document categorization, document verification integration using blockchain, and predictive document usage analytics using machine learning algorithms, further enhancing the scope and impact of the system.

ACKNOWLEDGEMENT

We are highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the major project work.

The constant guidance and encouragement received from Dr. Kiran Jyoti H.O.D. CSE Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Project Guide Er.Satinderpal Singh, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of computer science and engineering department of GNDEC for their intellectual support throughout the course of this work.

Finally, we are indebted to all whoseover have contributed in this report work.

Golu Kumar

Hare Krishna

Swagat

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

1.1 INTRODUCTION TO PROJECT

A Document Management System (DMS) is a software solution designed to store, organize, manage, and track digital documents efficiently, helping businesses transition from traditional paper-based filing systems to a fully digital environment. It replaces the need for physical filing cabinets and paper records, offering a centralized, secure, and accessible repository for digital documents. With a DMS, users can seamlessly access, share, edit, and collaborate on documents, regardless of their physical location.

A key feature of DMS is version control, which ensures that multiple versions of a document are tracked and that users always access the most recent updates. This prevents confusion and errors that could arise from working with outdated files. Along with version control, access permissions and role-based access allow organizations to regulate who can view, edit, or share specific documents, safeguarding sensitive information and ensuring that only authorized personnel can make changes.

Automated workflows are another significant benefit of a DMS, enabling organizations to streamline routine processes, such as document approval, review, and distribution. These workflows help reduce delays, minimize errors, and ensure that documents are processed in a timely manner, contributing to overall operational efficiency. Furthermore, the system automatically indexes and categorizes documents based on predefined criteria, such as keywords, metadata, or tags, making document retrieval fast and easy.

In addition to improving operational efficiency, a DMS supports regulatory compliance by ensuring that documents are stored, organized, and maintained according to industry standards and legal requirements. This is particularly important in industries such as healthcare, finance, and legal services, where failure to adhere to regulations can lead to significant penalties and reputational damage. A DMS can provide audit trails that log every action taken on a document, offering transparency and accountability to stakeholders and regulatory bodies.

Security is another cornerstone of a DMS, as it safeguards critical data from unauthorized access, loss, or theft. With encryption, secure cloud storage, and backup capabilities, a DMS helps protect against data breaches and ensures business continuity in the event of technical failures or disasters. This level of security is particularly crucial as businesses increasingly rely on cloud-based solutions for remote work and collaboration.

Beyond improving internal operations, a DMS fosters collaboration by allowing multiple users to work on documents simultaneously, share feedback, and track changes in real-time. This reduces the time spent on manual communication and enhances teamwork, particularly for organizations with distributed teams or international operations. Moreover, the integration of DMS with other enterprise software solutions, such as Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) systems, allows for seamless data flow across business processes, further improving overall productivity.

By streamlining document handling and reducing administrative overhead, a DMS helps organizations save time and money while ensuring that their documents are easily accessible, well-organized, and secure. Additionally, the system provides valuable insights through data analytics, helping organizations identify trends, monitor performance, and make more informed decisions based on document usage and workflow metrics.

In conclusion, adopting a Document Management System is not just about going paperless; it's about transforming the way businesses manage and utilize their information. By leveraging the capabilities of a DMS, organizations can increase productivity, reduce costs, improve compliance, and create a more collaborative, secure, and efficient workplace.

1.2 PROJECT CATEGORY

The Document Management System (DMS), is come under the category of Web-Based Application Development within the domain of Software Engineering and Information Systems. It focuses on the creation of a centralized digital platform for managing, organizing, storing, retrieving, and securing documents in an efficient and systematic way. As organizations increasingly move toward paperless operations, the need for a reliable and scalable document management solution has become critical. This project aims to address this demand by providing a robust, user-friendly system that automates and streamlines traditional document handling processes.

The project leverages modern web technologies, including a front-end interface built with responsive design principles, a secure backend with a relational or NoSQL database, and RESTful APIs for system communication and integration. It incorporates key software development practices such as modular design, object-oriented programming, user authentication, and role-based access control to ensure maintainability, security, and scalability.

Given its web-based nature, the system is accessible across devices and platforms, promoting real-time collaboration and remote document access. This makes it suitable for deployment in various industries,

including education, healthcare, corporate, and government sectors. It also aligns with current trends in cloud computing and enterprise digitalization.

In addition to the technical components, the project integrates important aspects of data governance, such as version control, audit trails, compliance with privacy regulations, and disaster recovery mechanisms. As such, this project not only contributes to the academic understanding of web-based systems but also has strong practical relevance and potential for real-world adoption. The system can be extended in the future with AI-based classification, blockchain integration, or advanced analytics, making it a comprehensive solution in the field of document and content management.

1.3 PROBLEM FORMULATION

In today's fast-paced, information-driven environments, organizations across sectors are challenged by the increasing volume and complexity of documents they must manage daily. Traditional methods of handling documents—such as paper-based filing systems or disorganized digital folders—are inefficient, error-prone, and difficult to scale. These approaches lead to issues like data redundancy, loss of critical documents, unauthorized access, lack of version control, and prolonged document retrieval times. Moreover, compliance with regulatory standards such as GDPR, HIPAA, or ISO requires stringent controls over how documents are accessed, stored, and modified, which is difficult to enforce with outdated or manual systems.

The absence of a centralized and secure system often results in miscommunication, lack of collaboration, data silos, and increased operational costs. Employees waste valuable time searching for documents or verifying outdated versions. Additionally, organizations are increasingly expected to support remote access and digital collaboration, further exposing the limitations of traditional document management.

To address these challenges, there is a clear need for a web-based Document Management System (DMS) that provides a structured, scalable, and secure approach to handling digital documents. The problem lies in designing and implementing a system that not only stores and organizes documents efficiently but also provides features such as metadata tagging, full-text search, version control, role-based access management, and audit trails.

This project formulates the problem as the need to build an integrated, user-friendly, and secure DMS that can meet modern organizational requirements. The system must support multiple users, offer real-time access, and ensure data integrity and security. Furthermore, it should allow seamless integration

with existing systems and be adaptable for future enhancements like AI-driven document classification or blockchain-based verification.

The successful formulation and resolution of this problem will result in a significant improvement in how organizations manage their information assets, ensuring greater efficiency, security, and compliance.

1.4 IDENTIFICATION/RECOGNITION OF NEED

In the modern digital era, the exponential growth of information has made efficient document handling a critical requirement for organizations across all sectors. As businesses, educational institutions, and government agencies generate and manage a vast number of documents daily, the limitations of traditional, manual, or semi-digital document management methods have become increasingly apparent.

Manual systems, including physical file storage or basic digital folder structures, often lead to disorganization, data redundancy, misplaced or lost documents, and unauthorized access. These issues not only compromise operational efficiency but also pose significant security and compliance risks. As organizations seek to adhere to stricter regulations such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and other data governance policies, the lack of a structured and secure document management system becomes a critical concern.

Moreover, the shift toward remote work, digital collaboration, and paperless environments has further amplified the need for a centralized platform that supports easy document access, sharing, and version control in real time. Without such a system, productivity suffers due to delays in document retrieval, poor communication, and lack of accountability in document changes and approvals.

The recognition of these challenges has led to the need for a comprehensive Document Management System (DMS) that can automate document workflows, ensure data security, and facilitate quick and reliable access to information. The system must address the core issues of accessibility, security, scalability, and regulatory compliance, while also offering a user-friendly interface and integration with existing tools.

By implementing such a solution, organizations can not only improve operational efficiency but also ensure that critical documents are managed systematically throughout their lifecycle—creation, review,

approval, storage, and disposal. This need forms the foundation for the development of the proposed DMS project.

1.5 EXISTING SYSTEMS

In most organizations these days, document management is still carried out with traditional or half-automated systems that fail to meet current standards. Such systems are usually based on physical storage, shared network drives, email sharing, or simple cloud computing tools such as Google Drive or Dropbox. Although such approaches possess some sort of convenience and ease of access, they have several drawbacks and threats that compromise productivity, data integrity, and security.

One of the most common existing approaches is the manual, paper-based document management system. Documents are printed, filed in cabinets, and tracked using handwritten or spreadsheet-based logs. While this method is simple and has been traditionally used for decades, it is highly inefficient in today's fast-paced environment. Obtaining documents is time-consuming, space is scarce and costly, and the chances of misplacing, losing, or destroying vital data are high. Besides, working in groups is also limited because multiple users are unable to open or modify documents simultaneously. Manual systems also have low security with papers susceptible to illegal access, theft, or disasters like fire and flood.

Another widespread technique is recourse to simple digital storage repositories, e.g., local computer directories, network shared drives on local networks, or run-of-the-mill cloud storage solutions. These repositories facilitate simple digital document storage and retrieval but do not have the organized foldering, metadata tagging, and version control functions required for efficient management of large numbers of documents. Furthermore, access control is usually restricted within these systems so that it can be challenging to implement user-level permissions or follow document history. Lack of automated workflows, audit trails, and document lifecycle management also limits their capability in commercial settings.

Others also use email for document exchange, which results in several copies of the same document being sent around inboxes with no indication of the latest or approved version. This not only adds redundancy and confusion but also security threats, as sensitive documents can be forwarded or accessed by unintended recipients. Additionally, tracking changes and getting approvals becomes tedious and time-consuming.

Some organizations have implemented simple document management modules integrated within ERP or CRM, but they are usually restrictive in terms of functionality and customization. They can lack

sophisticated features like full-text search, document classification, or smooth integration with other tools such as OCR, e-signatures, or AI-driven analytics.

In short, current systems—manual or half-digital—tend to be disjointed, inefficient, and insecure. They are lacking in sophistication, scalability, and integration features that can facilitate business operations in this day and age. These disadvantages highlight the urgency for a structured and centralized Document Management System with the capability of delivering automation, control, collaboration, and compliance in a safe and user-friendly environment.

1.6 OBJECTIVES

The primary objective of this project is to design and develop a comprehensive, web-based Document Management System (DMS) that enables organizations to efficiently store, organize, manage, and retrieve digital documents in a secure and structured environment. The system aims to eliminate the drawbacks of traditional and semi-digital document handling methods by introducing features such as role-based access control, version control, metadata tagging, full-text search, audit trails, and secure user authentication. It is intended to streamline document workflows, reduce redundancy, enhance collaboration, and ensure compliance with data protection regulations. Additionally, the system is designed to be scalable and user-friendly, supporting remote access and integration with existing enterprise tools to adapt to the evolving needs of modern organizations.

Below are the key objectives of the project:

1. **To streamline document handling through document automation system.**
2. **To fasten document processing and approvals.**
3. **To implement workflow management system to track document status.**

1. To streamline document handling through document automation system.

This objective focuses on improving the efficiency of document-related tasks by implementing a document automation system. In many organizations, handling documents manually—such as creating, sorting, filing, retrieving —consumes significant time and resources. A document automation system helps eliminate repetitive, error-prone tasks by automating processes like document generation, classification, storage, and routing. For example, when a user uploads a file, the system can automatically assign metadata, categorize it into the correct folder, and apply the appropriate access permissions. Automation also includes features like template-based document creation, auto-notifications for approvals or updates, and scheduled archival or deletion based on document lifecycle

policies. As a result, the document handling process becomes faster, more accurate, and less dependent on manual intervention. This not only boosts productivity but also ensures consistency, reduces operational costs, and minimizes the risk of human errors, contributing to better information management across the organization.

2. To fasten document processing and approvals.

This objective aims to accelerate the overall workflow involved in processing and approving documents within an organization. Traditional methods, such as emailing documents for review or printing them for physical signatures, often lead to delays, lost files, and lack of transparency. By implementing a digital Document Management System, the entire approval process can be automated and streamlined. The system can route documents to the appropriate users based on predefined workflows, send real-time notifications for pending approvals, and track the status of each document throughout its lifecycle. Features like electronic signatures, role-based task assignments, and time-stamped approval logs ensure that documents are reviewed and approved quickly and securely. This reduces turnaround time, improves accountability, and enhances overall efficiency. Faster document processing means quicker decision-making and a more agile response, which is critical in today's competitive and fast-paced environment.

3. To implement workflow management system to track document status.

This objective focuses on integrating a workflow management system within the Document Management System to effectively monitor and control the progress of documents throughout their lifecycle. In manual or unstructured systems, it's often difficult to know who has a document, what stage it is in, or if action is pending—leading to delays, confusion, and missed deadlines. By implementing a workflow management system, each document can follow a predefined path with clear steps for creation, review, approval, and archiving. Users can easily track the current status of a document, view its history, and receive notifications about tasks or changes. This enhances visibility, accountability, and transparency in document handling. Ultimately, this system helps streamline operations, reduce manual follow-ups, and improve efficiency by keeping all students informed and aligned throughout the document's workflow.

1.7 PROPOSED SYSTEM

The Document Management System (DMS) is an web-based application that is intended to overcome the shortcomings of current manual and semi-digital document processing practices. It provides a centralized, secure, and easy-to-use platform for digital document storage, management, and retrieval across their lifecycle. The system is intended to automate document processes, decrease processing time, facilitate collaboration, and increase compliance with organizational and legal requirements.

The basis of the suggested system is a centralized database that enables users to upload, categorize, and retrieve documents in an efficient manner. Each document can be tagged by metadata, sorted into folders, and referenced to particular projects or departments. There are various file types supported by the system, such as PDFs, Word documents, spreadsheets, photographs, and scanned papers. Full-text search support will be enabled for scanned documents by implementing Optical Character Recognition (OCR), which will extract text from the scanned documents and render it searchable.

User administration is carried out using role-based access control. This provides users with the ability to only see or modify documents based on their respective rights. For instance, an employee can upload and view documents, whereas a manager can approve, delete, or archive them. The system keeps audit trails for document activities like uploading, modifying, approving, and deleting, providing better accountability and traceability.

One of the standout aspects of the DMS under proposal is its workflow management system. The system enables documents to progress through stages like draft, review, approval, and finalization. In each stage, automatic reminders and notifications are made to the concerned users to ensure timely response and avoid bottlenecks. The approval process can also be customized to accommodate a single or multiple level based on the organization's structure.

To further accelerate document processing, the system has document templates and automatic routing. Templates automate frequently occurring documents to save time on redundant work, and automatic r outing sends documents to the right people according to metadata or workflow rules.

Security is also a high concern in the proposed system. Data is encrypted in transit and in storage. User authentication is performed through secure login processes, and optional two-factor authentication (2FA) can be used for extra security. The system has a backup and disaster recovery module to ensure that data is not lost.

User interfaces have been crafted in a user-friendly and interactive mode that grants desktop, tablet, or smartphone-based access. In doing this, the access granted provides end users the freedom of operating from wherever, offering flexibilities as well as distant-level collaborations.

Features that can accommodate integrations to communicate effectively with other business systems like CRM, ERP, or email packages have been included to ensure business functions around the handling of documents become integral in entire enterprise workflow plans.

In short, the suggested system provides an extensive solution that replaces manual document management with an automated, efficient, and secure electronic process. The system increases productivity, enhances transparency, and allows documents to be managed in line with internal regulations and external guidelines.

1.8 UNIQUE FEATURES OF THE PROPOSED SYSTEM

The Document Management System project is designed to improve the academic support system within the Organization through a document automation, user-friendly design, and seamless website integration. It offers a modern, interactive platform that caters to the real-time needs of students. Below are the unique features that set this system apart:

- Centralized Document Repository for Academic and Administrative Documents :
  The system provides a centralized location to store and manage academic papers, student records, syllabi, course materials, faculty documents, and administrative files. This streamlines access and reduces dependency on physical storage or scattered digital files.

- Role-Based Access Control (RBAC) for Faculty, Students, and Admin :
  Different users (faculty, students, and administrative staff) have distinct access levels, ensuring that sensitive information such as grades or personal data is protected, while faculty can easily access course materials and students can view lecture notes and assignments.

- Automated Workflow for Document Submission and Grading :
  The DMS automates document submission processes for assignments, thesis, and project work. It ensures students' submissions are routed directly to instructors, with an integrated grading system and automatic feedback.

- Real-Time Document Status Tracking :

  Students can track the status of their submitted documents (e.g., pending review, graded, feedback provided), and instructors can monitor the progress of assignments and projects, reducing delays in processing and feedback.

- Responsive Interface for Students and Faculty :

  The DMS is accessible on all devices, whether for browsing class notes, submitting assignments, or accessing faculty feedback, ensuring that users have a seamless experience on desktops, tablets, or smartphones.

CHAPTER 2 - REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

2.1 FEASIBILITY STUDY

Technical Feasibility:-

The technical feasibility of implementing a Document Management System (DMS) is highly promising due to the availability of mature, scalable, and secure technologies designed to manage, store, retrieve, and organize digital documents efficiently. A DMS provides centralized access to academic and administrative resources, streamlining the workflow for institutions like the CSE department.

Modern DMS platforms support modular architecture and are compatible with a wide range of third-party tools, making integration with existing institutional portals and applications seamless. Features such as version control, access permissions, metadata tagging, and full-text search ensure efficient and secure handling of documents across various categories including course materials, student records, announcements, and departmental reports.

Deployment of the DMS can be done via cloud-based platforms or on-premises servers, depending on institutional requirements. Cloud deployment, supported by platforms like Render or AWS, offers high availability, automatic backups, and scalability, allowing users to access the system 24/7 from any device with proper authentication.

A DMS also supports advanced features such as audit trails, workflow automation, and document sharing with role-based access control. This ensures compliance with data security standards and improves collaboration among faculty, staff, and students.

Given the wide adoption of DMS technologies, there is strong community and vendor support, rich documentation, and an abundance of training resources. A skilled technical team can efficiently implement and maintain the system with minimal risk. Overall, the implementation of a Document Management System is technically feasible, offering a reliable, secure, and scalable solution tailored to meet the growing digital needs of academic departments.

Economic Feasibility:-

The economic feasibility of implementing a Document Management System (DMS) is highly favorable, especially when considering the long-term cost savings and operational efficiencies it provides to academic departments like Computer Science and Engineering (CSE). While there is an initial investment associated with acquiring or subscribing to a DMS, the benefits quickly outweigh the costs due to reductions in manual work, paper-based processes, and document storage requirements.

A wide range of DMS options are available, including open-source solutions such as OpenKM, which significantly reduce licensing costs. Even commercial platforms offer flexible pricing models—such as subscription-based or one-time licenses—allowing institutions to choose solutions that fit their budget. Additionally, cloud-based deployment on platforms like Render or AWS eliminates the need for costly in-house server infrastructure and ongoing maintenance expenses.

By automating document handling, search, retrieval, and storage processes, a DMS greatly reduces the administrative workload and the associated labor costs. It also minimizes the risk of document loss or duplication, which can be financially damaging over time. The system ensures quicker access to information, improving productivity among staff and students, and leading to better decision-making and resource utilization.

Moreover, implementing a DMS supports compliance with data management regulations and avoids potential fines related to information mishandling or unauthorized access. As a result, institutions benefit from improved governance and lower legal risks.

Operational Feasibility:-

The operational feasibility of implementing a Document Management System (DMS) in an academic setting, such as a Computer Science and Engineering (CSE) department, is strong due to its ability to seamlessly integrate into existing workflows while significantly improving day-to-day operations. A DMS provides centralized access to academic and administrative documents, streamlining operations such as course management, faculty coordination, student support, and record-keeping.

The user-friendly interface of most modern DMS platforms ensures that faculty, staff, and students can adapt quickly with minimal training. Features such as advanced search, version control, document tagging, and access permissions help users manage, retrieve, and share documents efficiently. These

capabilities enhance productivity by reducing time spent on locating and organizing physical or unstructured digital documents.

The system supports role-based access control, ensuring that only authorized users can view, edit, or upload sensitive documents, thereby improving security and compliance with institutional policies. Audit trails and automated backups also ensure accountability and data integrity. These functionalities contribute to smoother operations, especially during accreditation processes, audits, and student evaluations, where quick access to accurate documents is essential. Overall, a DMS offers a practical, reliable, and effective operational solution that aligns with the department's current practices while enhancing efficiency, transparency, and service quality.

2.2 SOFTWARE REQUIREMENT SPECIFICATION

The Software Requirement Specification (SRS) for the Document Management System (DMS) serves as a foundational document that outlines both the functional and non-functional requirements essential for the successful development, deployment, and operation of the system. It defines critical system behaviours such as the DMS's ability to store, organize, retrieve, and manage documents efficiently, support role-based access control, maintain version history, and enable real-time collaboration among users. The SRS ensures the system is capable of delivering fast and reliable document access, supporting essential workflows for academic departments, such as uploading syllabi, distributing notices, managing student records, and storing official communications.

The specification also covers non-functional requirements including usability, scalability, performance, reliability, and security. These ensure that the DMS offers a user-friendly interface accessible to faculty, students, and administrators, while also being scalable to accommodate increasing data volumes and concurrent users. Security features such as access control, encryption, and audit trails are specified to protect sensitive academic information and ensure compliance with data protection standards.

By establishing a clear and unified understanding among all stakeholders, the SRS serves as a blueprint for the development team, facilitating structured implementation, rigorous testing, and ongoing maintenance. It ensures the DMS aligns with the department's operational goals, enhances document accessibility and collaboration, and supports efficient academic and administrative communication within the institution.

Data Requirements:

1. User Data

   The system must manage data related to all users who interact with the DMS:

   - User ID
   - Name
   - Email Address
   - Role (Student, Faculty, Admin)
   - Authentication Credentials (hashed passwords or Single Sign-On tokens)
   - Access Permissions

2. Document Content

   - Actual file contents (stored securely in the file system or cloud storage)
   - Support for multiple file types: PDF, DOCX, etc.

3. Access and Activity Logs

   To maintain security and compliance, the system must log:

   - Document upload, download, edit, and deletion events
   - User login and logout activity

4. Version Control Data

   The system must track:

   - Document revisions and change history
   - Who made changes and when
   - Ability to restore previous versions

5. Backup and Recovery Data

   - Periodic backups of all documents and metadata
   - Disaster recovery snapshots and rollback data

6. Search Index Data

- Indexed metadata and document contents (if supported) to facilitate fast search
- Support for full-text indexing and keyword-based queries

Functional Requirements:

1. Document Storage and Organization

- The system must allow users to upload and store documents in various formats (e.g., PDF, DOCX, PPT, XLSX).
- The system must support the creation of folders and categories to organize documents based on department, project, or subject.
- The system must allow users to move or reorganize documents within the file structure.

2. Metadata Management

- The system must allow the user to add and edit metadata for each document, including title, author, date, category, keywords, and description.
- The system must automatically generate metadata based on document properties (e.g., date of upload, file type).
- The system must allow metadata to be searched, filtered, and sorted.

3. Access Control and Permissions

- The system must support role-based access control, where users are assigned roles (e.g., Administrator, Faculty, Student) with specific permissions (view, edit, delete, upload).
- The system must allow document-level permissions, restricting access based on document sensitivity or user role.
- The system must support user authentication and authorization to ensure secure access to documents.

4. Version Control

- The system must maintain version history for each document, storing all previous versions and allowing users to restore earlier versions if necessary.
- The system must track and display the user responsible for each version update.

- The system must allow users to view version differences (if applicable) between documents.

5. Search and Retrieval

   - The system must provide a search functionality allowing users to search documents by metadata (e.g., title, author, date) and full-text content.
   - The system must support advanced search features, including filters (e.g., document type, category, keywords).
   - The system must return relevant results within a reasonable time frame, even with large document repositories.

6. Audit Trails and Activity Logs

   - The system must record and maintain an audit trail for each document, logging actions such as upload, edit, view, and delete.
   - The system must provide administrators with reports of user activities and document access.

7. Document Collaboration

   - The system must allow users to comment on or annotate documents in a collaborative manner.
   - The system must allow multiple users to view a document simultaneously (read-only or collaborative editing).

8. Backup and Recovery

   - The system must automatically back up documents and metadata on a scheduled basis (e.g., daily, weekly).
   - The system must allow administrators to restore documents and metadata from previous backups in case of system failure.

9. Document Sharing and Export

   - The system must allow users to share documents with other users through links, email, or integrated communication tools.
   - The system must allow documents to be exported in different formats (e.g., PDF, DOCX, CSV).

10. Document Security

- The system must provide encryption for documents stored within the system to protect sensitive data.
- The system must ensure secure transmission of documents (using HTTPS and SSL/TLS protocols).
- The system must support multi-factor authentication (MFA) for added security.

Performance Requirements :

1. Document Upload and Storage

- The system must be able to upload and store a document.

2. Search and Retrieval Performance

- The system must return search results within 2 seconds for queries involving document metadata (e.g., title, author, tags).

3. Document Rendering and Access Speed

- The system must open and render documents (e.g., PDF, DOCX) for viewing or editing.

4. Data Backup and Restoration

- The system should perform daily backups with minimal impact on system performance, with backup processes.
- The system must restore documents from backups.

5. Security and Encryption

- Authentication and authorization checks (e.g., login, role verification).

Dependability Requirements:

Dependability in a Document Management System (DMS) is the reliability, security, and consistency under different conditions under which it works. In academic settings, where the integrity and

availability of documents like student records, academic texts, and administration files are extremely important, the following dependability requirements should be fulfilled:-

1. Reliability

   The DMS must perform its intended functions accurately and without failure over time. Users—students, faculty, and administrators—should be able to store, retrieve, and manage documents without errors or system crashes. Reliability ensures that documents are not lost or corrupted and that all operations (e.g., upload, access, approvals) execute correctly.

2. Maintainability

   The DMS must be easy to maintain, update, and troubleshoot. Any bugs, errors, or performance issues should be easily identifiable and fixable without long downtimes. Modular design and clear system documentation will support quick updates and the addition of new features.

3. Security

   Given the sensitivity of academic and personal data, the system must enforce robust security policies. This includes encryption of stored and transmitted data, user authentication, role-based access control, and protection against unauthorized access or data breaches. The system should also comply with data privacy standards relevant to educational institutions.

4. Data Integrity

   The system must ensure that documents are not altered or tampered with without proper authorization. Version control can help maintain the integrity of documents throughout their lifecycle.

5. Fault Tolerance

   In the event of hardware failures, network interruptions, or other unexpected issues, the system should continue to operate or recover gracefully. Failover mechanisms, redundant backups, and auto-restart features can help achieve fault tolerance.

6. Auditability

   The DMS should maintain detailed logs of all user activities and system actions, such as uploads, edits, approvals, and deletions. This not only helps in tracking usage but also supports accountability and forensic investigations in case of security incidents or disputes.

7. Scalability and Flexibility

   As the institution grows or as more users and documents are added to the system, the DMS should scale without performance degradation. The architecture should allow flexibility in modifying workflows, adding departments, or integrating new tools.

2.3 SDLC MODEL TO BE USED

The Software Development Life Cycle (SDLC) provides a structured framework for planning, creating, testing, and deploying a software system. For the development of a Document Management System (DMS) in a college environment, the Waterfall Model is an appropriate choice due to its sequential, well-defined phases, which align well with the academic project's scope and timeline.

The Waterfall Model is a traditional software development methodology that follows a linear and sequential approach, where the development process is divided into distinct phases such as requirement analysis, system design, implementation, testing, deployment, and maintenance. Each phase must be completed before the next one begins, and there is typically no overlapping between phases. In the context of a Document Management System (DMS) for a college, the Waterfall Model provides a clear and structured path to follow, starting with a thorough understanding of user requirements—such as document storage, access control, and approval workflows—and proceeding step-by-step through design and development. This model emphasizes comprehensive documentation and well-defined deliverables at each stage, making it easier to manage and track progress. While it lacks flexibility for accommodating changing requirements later in the process, its simplicity and predictability make it ideal for academic projects where the scope is clear from the beginning. Overall, the Waterfall Model offers a disciplined development process that helps ensure the system is built efficiently and in alignment with the initial specifications.

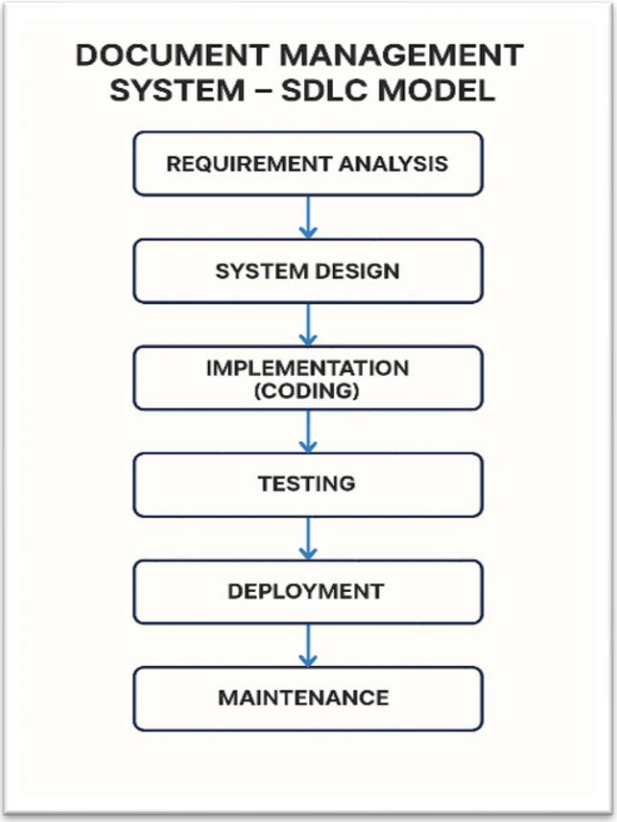

Figure 2.1 SDLC Model

Preparation and Planning

Preparation and planning for a Document Management System (DMS) involve a thorough assessment of organizational needs, document workflows, compliance requirements, and existing infrastructure. It begins with identifying key stakeholders, defining the scope, and setting clear objectives such as improving document retrieval, enhancing security, or enabling collaboration. A comprehensive audit of current document handling practices helps determine gaps and inefficiencies. The planning phase includes selecting the right platform (cloud-based or on-premise), setting access control policies, designing metadata schemas for effective indexing, and establishing a migration plan for existing documents. Budget estimation, risk analysis, and timeline development are also crucial to ensure a smooth and scalable implementation.

System Design

Design and prototyping of a Document Management System (DMS) are crucial phases that translate requirements into a visual and functional blueprint. The design phase focuses on system architecture, including database design for storing documents and metadata, user interface (UI) layouts, role-based

access control structures, and workflow diagrams for document lifecycle management. It also includes designing the integration points with email systems, cloud storage, or ERP tools. Prototyping involves creating interactive mockups or clickable wireframes using tools like Figma or Adobe XD to simulate the user experience. These prototypes allow stakeholders to visualize how users will upload, retrieve, and manage documents, helping to gather early feedback and make adjustments before full development begins. This phase ensures the system is user-friendly, secure, and aligns with organizational processes.

Implementation

Development of a Document Management System (DMS) involves the actual coding and construction of the system based on the approved design and prototype. This phase typically starts with setting up the development environment, including selecting the appropriate technology stack—such as a backend framework (e.g., PHP), a frontend framework (e.g., HTML, CSS, Javascript), and a database (e.g., MySQL). Developers implement core functionalities such as document upload/download, version control, search and indexing, user authentication, and access management. APIs are built for integration with other services like email notifications or cloud storage. Code is version-controlled using systems like Git, and modular programming is adopted for scalability and maintainability. Throughout this phase, developers perform unit testing to ensure individual components work as expected and prepare the system for integration testing in the next stage.

Testing

Testing in a Document Management System (DMS) ensures that all functionalities work as intended, are secure, and meet the defined requirements. This phase includes multiple levels of testing: unit testing checks individual components like document upload or user authentication; integration testing ensures that modules such as access control; and system testing validates the entire DMS under simulated real-world conditions. User Acceptance Testing (UAT) is performed with end-users to confirm the system is intuitive and meets business needs. Additionally, security testing checks for vulnerabilities such as unauthorized access, file injection, or data leaks. Performance testing is conducted to verify the system can handle large volumes of documents and concurrent users efficiently. Identified bugs and issues are logged, prioritized, and fixed before deployment.

Maintainence

Maintenance of a Document Management System (DMS) is the ongoing process of ensuring the system remains secure, functional, and aligned with evolving organizational needs. This phase includes routine

activities such as fixing bugs, applying security patches, and updating software dependencies. Regular backups and system monitoring are critical to prevent data loss and identify performance issues. Maintenance also involves updating user permissions, managing storage, and ensuring compliance with regulatory standards like GDPR or HIPAA. As users provide new feedback or business processes change, enhancements may be required—such as adding features, improving search capabilities, or integrating with other tools. Proper documentation and user support are also key parts of maintenance to ensure the system remains reliable and user-friendly over time.

CHAPTER 3 - SYSTEM DESIGN

3.1 DESIGN APPROACH

The design approach for the Document Management System (DMS) in a college setting focuses on building a modular, scalable, and user-friendly application that streamlines document handling for students, faculty, and administrative staff. The system is designed using a modular architecture where core functionalities—such as user management, document upload/download, workflow automation, and notifications—are developed as independent yet integrated components. A key element of the design is role-based access control, ensuring that users only have access to features and documents relevant to their roles; for example, students can submit documents, faculty can review and approve them, while administrators oversee the entire process. A relational database will be used to manage user data and document metadata, while actual files are stored securely using an integrated file system. The user interface will be intuitive and responsive, designed to cater to the varying technical proficiency of its users, with dashboards and clearly defined actions for each role. Automated workflows will be implemented to reduce manual effort in document routing and status tracking. Security is a critical aspect of the design, with features like encrypted communication, secure logins, and audit trails. The system is also designed with future growth in mind, ensuring it can scale to handle additional users, documents, and new functionalities over time. Overall, this design approach ensures a reliable, secure, and efficient system tailored to the academic environment.

Function-Oriented Approach:

The function-oriented approach to designing the Document Management System (DMS) emphasizes breaking down the system into a set of distinct, well-defined functions or processes that perform specific tasks. Instead of focusing on objects or data structures, this approach views the system as a collection of interrelated functions that transform inputs into outputs. For a college-based DMS, the primary functions include user login and authentication, document upload and storage, document retrieval and search, approval workflow, and status tracking. Each function is designed to operate independently but in coordination with others, forming a structured and logical flow of operations. For example, when a student uploads a document, the system captures metadata and passes it to the workflow module, which routes the document to the assigned faculty member for review.

This function-based design makes it easier to create Data Flow Diagrams (DFDs) and functional decomposition diagrams, which visually map the flow of data between processes and data stores. It

also simplifies development and maintenance, as each function can be coded, tested, and modified independently. Security and access control can also be integrated into individual functions, ensuring each process performs validations before proceeding. Additionally, the function-oriented approach enhances clarity and reduces complexity, which is particularly beneficial in academic projects where structured and modular design is preferred. By organizing the DMS around its core functions, the development process becomes more manageable, scalable, and easier to document—making it well-suited for both educational environments and real-world deployment.

Object-Oriented Approach:

The object-oriented approach focuses on designing the system around real-world entities and their interactions rather than functions alone. In this approach, the Document Management System (DMS) is modeled as a collection of objects, each representing entities such as User, Document, Workflow, Folder, or Notification. Each object encapsulates both data (attributes) and behavior (methods) related to that entity. For example, a Document object may have attributes like title, uploadedBy, status, and timestamp, while its methods could include upload(), approve(), and trackStatus(). Similarly, a User object may have roles like Student, Faculty, or Admin, each with specific access permissions and associated actions. This modeling closely mirrors the real-world use of the system, making it intuitive, modular, and easier to understand.

Using object-oriented principles such as encapsulation, inheritance, and polymorphism, the system becomes highly reusable, maintainable, and scalable. For instance, the User class could be a base class, and Student, Faculty, and Admin can be derived classes with role-specific extensions. The encapsulation of data within objects helps enforce data security and access control, ensuring only authorized actions can be performed on documents. The object-oriented approach also facilitates code reuse, as common functionality across different objects can be abstracted and reused, reducing redundancy. Moreover, changes in one part of the system have minimal impact on others, promoting better maintainability—an important factor for evolving systems like a college DMS.

In a college setting, this approach offers additional benefits, such as the ability to easily integrate new features like version control or document analytics by adding or extending classes without major structural changes. It also allows the system to grow as more departments, users, or workflows are added. Overall, the object-oriented approach provides a robust and future-proof framework for building a DMS that is aligned with both academic needs and real-world software engineering practices.

3.2 DETAILED DESIGN

The detailed design of a Document Management System (DMS) focuses on translating functional and non-functional requirements into a complete system blueprint, covering architecture, user interaction, security, and workflow processes. The system is structured to support the storage, retrieval, sharing, and approval of documents across different user roles—mainly students, faculty, and administrators.

1. System Architecture

The system adopts a **three-tier architecture** consisting of:

1. ***Presentation Layer***: The front-end interface built using HTML5, CSS3, JavaScript. It provides role-specific dashboards for users, such as document upload interfaces for students, approval panels for faculty, and administrative control panels for staff.
2. ***Application* Layer**: This layer handles the logic, including user authentication, file validation, workflow processing, and access control. It is developed using a backend technology such as PHP.
3. ***Data Layer***: Manages structured data in a relational database (e.g., MySQL) and unstructured data (actual documents) in a secure file storage system.

2. Key Components

1.User Management Module

- Handles registration, login, and role-based access control.
- Defines roles: Student, Faculty, Admin.
- Controls what each role can see or do.

2. Document Module

- Allows users to upload, download, view, or delete documents based on permissions.
- Metadata such as title, subject, author, date, and status are stored with each document.

3. Workflow Engine

- Manages document lifecycle: submission → review → approval/rejection.
- Notifies users of pending tasks or completed actions via alerts.

3. Database Design

- **Users**: user_id, name, email, password_hash, role
- **Documents**: doc_id, title, uploaded_by, file_path, status, upload_date

4. Security Features

- **Authentication**: Password hashing, secure login sessions.
- **Authorization**: Role-based access to features and documents.

5. System Interfaces

- **Student Interface**: Upload documents, track status, view feedback.
- **Faculty Interface**: Review submissions, approve/reject, add comments.
- **Admin Interface**: Manage users, view reports, access all documents, manage system settings.

3.2.1 SYSTEM DESIGN

The system design of the Document Management System (DMS) for a college is centered around a three-tier architecture comprising the presentation layer, application layer, and data layer. The presentation layer serves as the front end, where users such as students, faculty, and administrators interact with the system through user-friendly web interfaces. This layer enables role-based access to features like document uploads, status tracking, and approval actions. The application layer, acting as the core of the system, handles business logic, including authentication, document workflow processing, and user role management. The backend is responsible for validating inputs, managing sessions, and coordinating the document approval lifecycle. Technologies such as PHP can be used in this layer to ensure smooth operation and system responsiveness.

The data layer includes a relational database (such as MySQL) that stores structured information like user accounts, document metadata, and workflow status. Actual files are stored securely either on a server file system. The design also incorporates modules for notifications, audits, and logs to maintain transparency and accountability. Security is enforced through role-based access control, encrypted communication (HTTPS), and secure password management. This modular design ensures that each function—such as uploading, reviewing, and tracking documents—is clearly defined, making the system easier to maintain and scale. Overall, the system is designed to automate and streamline document handling processes in a college environment, reducing manual effort and improving efficiency for both students and staff.

Design Features:

1. Modular Architecture

The system is built using a modular design, separating functionalities into independent components such as user management, document handling, workflow processing, and notifications. This makes the system easier to develop, test, and maintain.

2. Role-Based Access Control (RBAC)

Different user roles (Student, Faculty, Admin) are assigned specific permissions. For example, students can upload and view documents, faculty can approve or reject them, and administrators have full access to manage users and monitor workflows.

3. Document Storage

Uploaded documents are stored on the system. Metadata is stored in a relational database, allowing efficient search, retrieval, and classification.

4. Workflow Automation

The system supports automated document routing and approval workflows. When a document is uploaded, it is automatically forwarded to the assigned reviewer (e.g., faculty), and status updates are triggered as it moves through the approval process.

5. Responsive User Interface

The user interface is designed to be clean, intuitive, and responsive across devices. Role-specific dashboards simplify navigation and display relevant tasks, document statuses, and notifications.

6. Notification System

Users receive real-time alerts about important events, such as document approvals, rejections, or comments from reviewers.

7. Scalability and Maintainability

The system is designed to scale with the institution’s needs, supporting increasing users and

documents while remaining maintainable due to its clean architecture and modular codebase.

Data Flow Diagram

A **Data Flow Diagram (DFD)** is a visual representation that illustrates how data moves through a system, highlighting the flow of information between processes, external entities, data stores, and the system itself. In the context of a Document Management System (DMS) for a college, a DFD helps map out how documents and related data are created, processed, and stored by various users such as students, faculty, and administrators. For example, when a student uploads a document, the DFD shows the data flowing from the student (external entity) to the document submission process, which then stores the document in a file repository and updates its metadata in the database. The diagram also depicts how the document is routed to faculty for review, and how notifications are triggered throughout the process. By breaking the system down into logical processes and data interactions, the DFD serves as a useful tool for understanding, analyzing, and designing the internal workflow of the DMS. It ensures that all functional requirements are accounted for and helps developers and stakeholders visualize how the system operates as a whole.

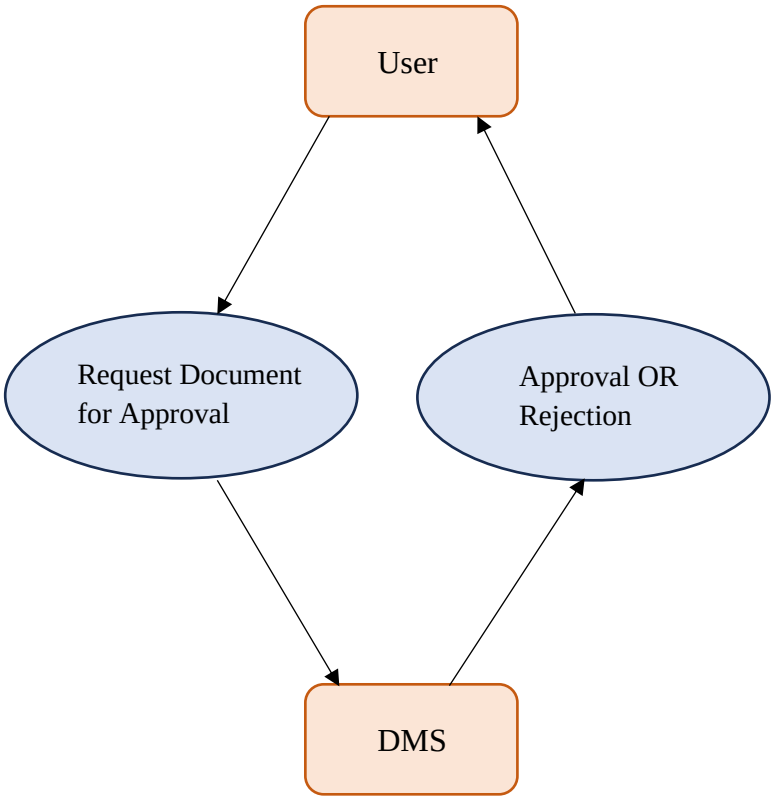

Figure 3.1 Data Flow Diagram

Block Diagram

A block diagram is a high-level graphical representation of a system that shows the major components and their interactions in a simplified, easy-to-understand format. For a Document Management System (DMS) in a college environment, the block diagram outlines the core modules such as the User Interface, Authentication System, Document Storage, Workflow Engine, Database, and Notification System. Each block represents a major function or subsystem, and arrows between them illustrate the flow of control and data. For example, when a student logs in through the User Interface, the request is passed to the Authentication System, which validates the credentials before granting access to the Document Upload module. Once uploaded, the document passes through the Workflow Engine for faculty review, while related data is stored and managed in the Database. The Notification System operates alongside to alert users about status changes. This kind of diagram is especially useful in planning and communication, as it provides a clear overview of system architecture without getting into technical details, making it ideal for presentations, documentation, and initial design discussions.

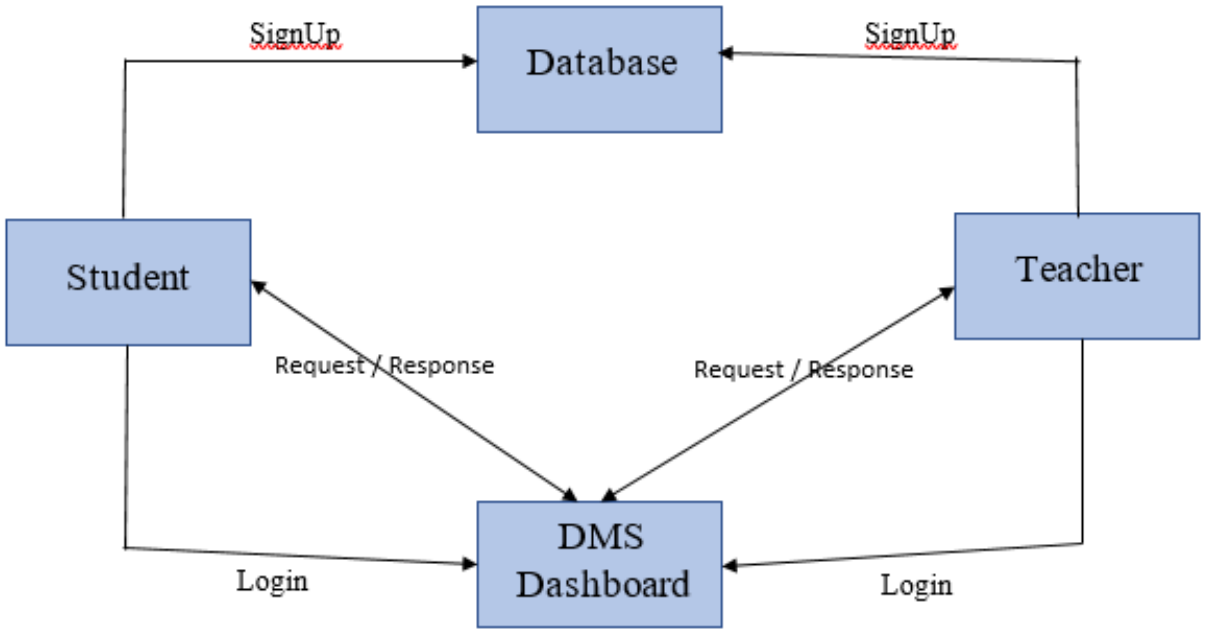

Figure 3.2 Block Diagram

Below is the explanation of the block diagram :-

1. Student:
   - **Attributes:**
     i. URN: University Roll Number of the student.
     ii. Name: Full name of the student.
     iii. Password / Confirm Password: Used for authentication and account creation.

2. Faculty:
   - **Attributes:**
     i. Name, Department, Course, Email: Faculty member's personal and academic details.
     ii. Username, Password / Confirm Password: Used for login and registration.
     iii. Role: Designation (e.g., Student, admin).

3. Sign Up:
   - A shared process for both students and faculty to register in the system.
   - Stores the user data in the Database after validation.

4. Login:
   - Common login entry point for both students and faculty.
   - Validates user credentials and routes them to their respective dashboards.

5. Document Management System:
   - Core system or dashboard where all user interactions take place after login.
   - Handles:
     i.Document submission
     ii.Status tracking
     iii.Review and approval workflows
     iv.Notifications
     v.User-specific dashboards based on roles

6. Database:
   - Central data store for all user information, credentials, and document metadata.
   - Used during both Sign Up and system operations like login and request handling.

7. Request/Response:
   - Represent the communication between users (student/faculty) and the DMS system.
   - Include actions like uploading documents, submitting requests, and receiving system responses or updates.

Flow Chart

A flowchart is a visual representation of a workflow or process that outlines the sequence of steps using standardized symbols such as rectangles (processes), diamonds (decisions), and arrows (flow direction). In the context of a Document Management System, the flowchart illustrates the logical progression from user actions—such as registration and login—to various document-related operations like uploading, retrieving, or forwarding files. It serves as a roadmap that visualizes how the DMS handles user interactions.

The process typically begins with user authentication, where users such as students or faculty sign up or log in. Once authenticated, the system grants access to the dashboard, where users can perform actions based on their role (e.g., upload documents, send requests, or forward files). Decision points determine the flow based on conditions like user role or request type. The system also interfaces with a central database to store, retrieve, and update documents and user data. Each block in the flowchart represents a specific function or decision point, and arrows indicate the flow of actions.

This visual tool is especially valuable in software design and analysis, as it simplifies complex processes and helps stakeholders understand the operational logic of the DMS. It ensures clarity in development, promotes consistency in process execution, and supports better communication between technical teams and end-users.

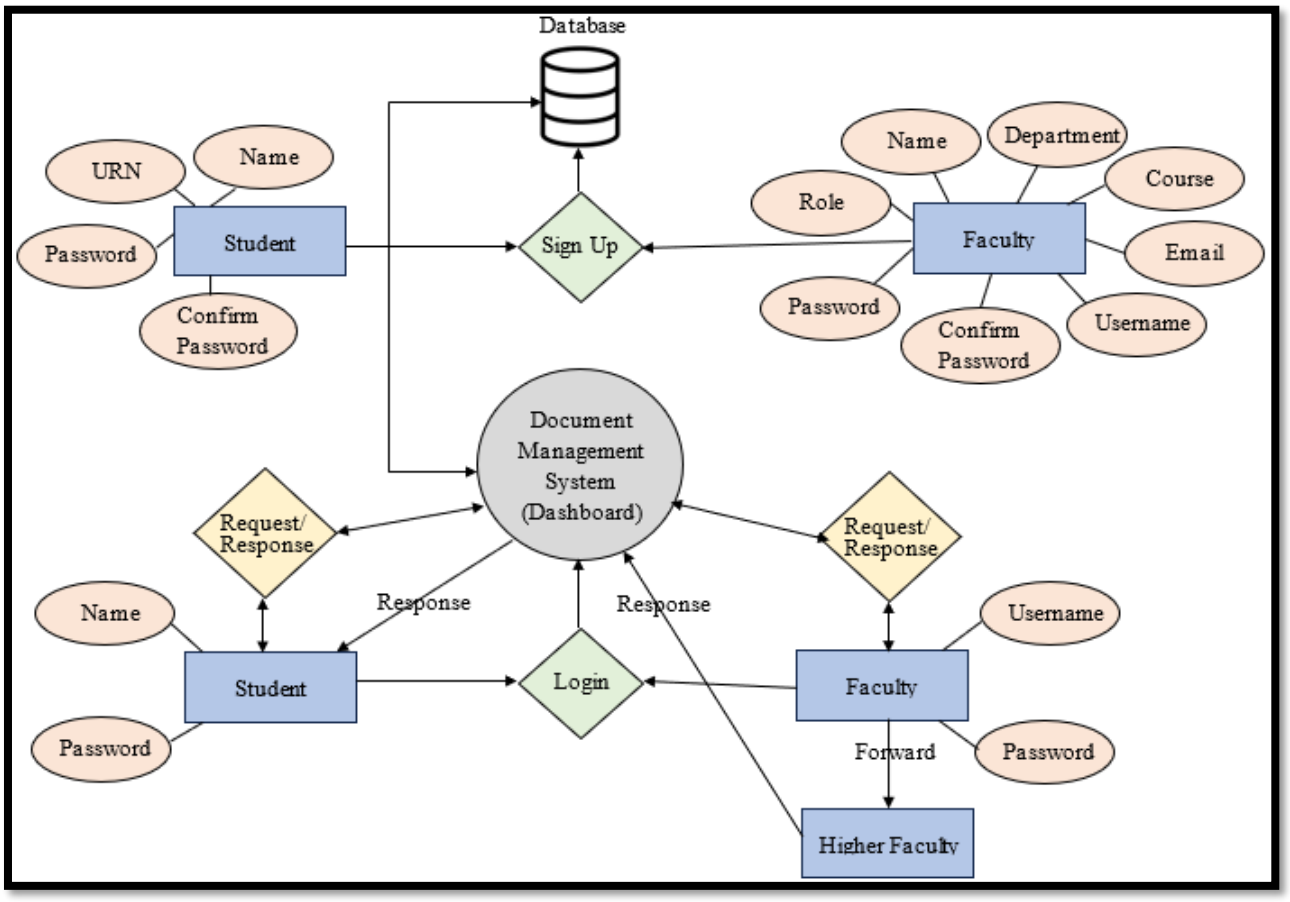

Figure 3.3 Flow Chart

The flowchart visually outlines the working process of the Document Management System (DMS), which handles user interactions for uploading, retrieving, and managing digital documents based on role-based access and workflow rules. The process begins when a user (student or faculty) provides login credentials to access the system. Upon successful authentication, the user is directed to the dashboard where they can perform different actions based on their role.

If the user chooses to upload a document, the system checks whether the required metadata (e.g., title, category, permissions) is provided. If all required information is complete, the document is stored in the central database/repository and a success response is returned. If any metadata is missing, the system prompts the user to complete the form before proceeding.

For retrieving or viewing a document, the user submits a request that is checked against access control rules. If the user has the necessary permissions, the document is retrieved from the repository and presented; otherwise, the system returns an access denied message.

In workflows involving faculty approvals, a student's request may be forwarded to a higher faculty level based on predefined rules. The faculty can then approve, reject, or request modifications, and the system responds accordingly.

Finally, the result of each operation—upload, retrieve, forward, or approve—is communicated back to the user through the dashboard interface. This structured flow ensures efficiency, security, and compliance in document handling, while also allowing escalation and review for controlled processes.

Use Case Diagram

A use case diagram is a visual representation that shows how users interact with the document management system to achieve specific goals or perform tasks. It identifies different types of users, known as "actors"—such as students, staff, or administrators—and the various actions or "use cases" they can perform, such as login, signup, request an application, approvals, workflows. This diagram helps outline how the document management system responds to each user input, clarifying its functional requirements and interaction points.

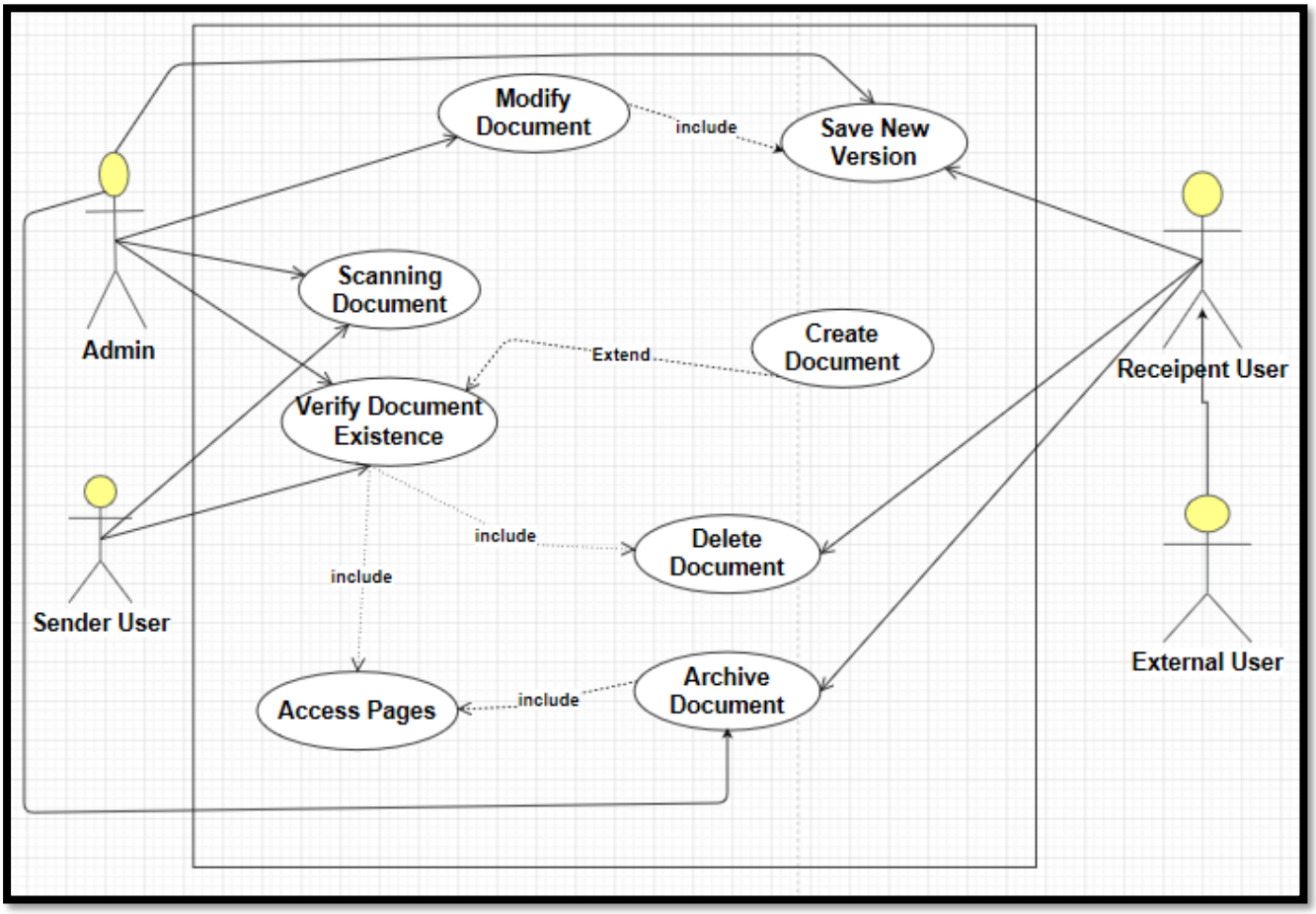

Figure 3.4 Use Case Diagram

The use case diagram illustrates the interaction flow between the user and the Document Management System (DMS), highlighting the core functionalities involved in handling digital documents. The user initiates the process by logging into the system and can perform several main use cases such as “Upload Document,” “Search Document,” “View Document,” and “Download Document.” These core interactions represent the primary activities users engage in during daily operations.

If the user requires assistance, they can access the “Help and Support” feature, which offers guidance on system usage and troubleshooting tips. Once a document is uploaded, it is stored securely in the system's repository, and metadata is captured to support indexing and future retrieval.

The system backend processes these operations by interacting with the database and enforcing role-based access control, ensuring only authorized users can access, edit, or delete specific documents. The “Version Control” use case allows the system to manage document updates while preserving previous versions for audit and rollback purposes. In case of administrative needs, an Admin actor can perform additional use cases like “Manage Users”, “Audit Logs”, and “Assign Roles.”

3.3 USER INTERFACE DESIGN

The user interface design of the Document Management System (DMS) is structured to provide a clean, intuitive, and role-based experience tailored for a college environment. The system begins with a login page where users—students, faculty, or administrators—can securely enter their credentials. A role selection dropdown ensures users are directed to the appropriate dashboard upon successful login.

For new users, the sign-up page features separate forms: students provide their URN, name, and password, while faculty input additional details such as department, course, email, and role. Once authenticated, users access a personalized dashboard. Students are presented with options to upload documents, view submission status, and receive notifications, all within a user-friendly layout featuring upload forms and status tables. Faculty members, on the other hand, have access to a review interface where they can evaluate, approve, reject, or forward documents to higher authorities, supported by action buttons and feedback modals.

Administrators have broader control, with tools to manage users, view system logs, and oversee document flow. The design maintains a consistent layout across all roles with a modern color palette, legible fonts, and a responsive structure that works across devices. A notification system integrated with the dashboard ensures that users remain updated on document statuses and system alerts in real time, enhancing communication and overall usability.

3.4 METHODOLOGY

The methodology adopted for developing the Document Management System (DMS) in a college setting is rooted in a structured and systematic approach, aimed at ensuring efficient document processing, user satisfaction, and system scalability. The Waterfall Model of Software Development Life Cycle (SDLC) has been chosen due to its linear and sequential structure, which aligns well with the requirements and predictability of the project. This model divides the development process into distinct phases: Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. The project initiates with a thorough Requirement Analysis phase, where the needs of different user types—students, faculty, and administrators—are carefully identified. This includes understanding how users interact with documents, what actions they perform (upload, review, approve, track), and what data is necessary to support these actions. Detailed documentation of these requirements helps avoid ambiguity and sets a clear path for design and implementation.

Following this, the System Design phase translates requirements into a technical architecture. This includes designing data flow diagrams, block diagrams, database schemas, and user interface mockups. Both functional and non-functional requirements are considered, ensuring the system not only meets its goals but also performs efficiently and securely. The Implementation phase involves actual coding using suitable technologies such as web frameworks for the front end (e.g., HTML, CSS, JavaScript) and backend technologies (e.g., PHP, Python, or Node.js) along with database systems like MySQL or MongoDB. This phase follows coding standards and best practices to maintain readability, reusability, and scalability.

The Testing phase is critical to ensure that each module functions as intended. Unit testing is carried out for individual components, followed by integration testing to validate the flow of data between modules. System testing and user acceptance testing are conducted to verify the system's performance under real-world conditions. Feedback gathered during this phase is used to make improvements before deployment. Once tested, the system enters the Deployment phase where it is made available for use within the college environment. This includes installing the software on the server, configuring access for users, and initializing the database.

Lastly, the Maintenance phase ensures that the system remains functional and up to date after deployment. This includes handling bug fixes, implementing feature updates based on user feedback, and ensuring compatibility with evolving technologies. Documentation is maintained throughout the development process to support future modifications and scalability.

Throughout this methodology, user involvement is considered at key stages to ensure the final product aligns with actual needs. Emphasis is placed on data security, user-friendliness, and role-based access, which are essential for a document management system in an academic context. By following this structured methodology, the DMS is expected to improve document handling, reduce processing time, and enhance transparency and accountability across departments, contributing to a more efficient and paperless campus environment.

CHAPTER 4 - IMPLEMENTATION AND TESTING

4.1 INTRODUCTION TO LANGUAGES, TOOLS AND TECHNOLOGIES USED FOR PROJECT WORK

The development of the Document Management System (DMS) relies on a combination of programming languages, tools, and technologies that collectively support efficient web-based application design and deployment. At the core of the project, HTML, CSS, and JavaScript are used to build the front-end interface, offering a user-friendly and responsive design for both students and faculty. HTML structures the content, CSS styles the visual elements, and JavaScript adds interactivity to the web pages, such as form validations and dynamic content updates. For server-side development, PHP is employed to handle backend logic, manage sessions, process data, and ensure secure communication between the user interface and the database. The MySQL database system is used for storing user credentials, document metadata, submission statuses, and logs, offering reliable data integrity and support for complex queries.

To enhance productivity and maintain code quality, Visual Studio Code (VS Code) serves as the primary code editor, providing powerful extensions, syntax highlighting, and version control integration. Additionally, XAMPP is used as a local development server for PHP-based implementations, offering an easy environment to test database and server functionality during development. For project management and version tracking, Git and GitHub are utilized, allowing collaborative development, change history tracking, and seamless deployment. Moreover, if the application includes asynchronous operations or needs to improve user experience, technologies like AJAX can be integrated to enable real-time data communication without reloading the page.

In terms of security and performance, built-in hashing functions are used to secure passwords, and form validation is implemented both on client and server sides to prevent SQL injection and invalid inputs. Overall, the chosen languages, tools, and technologies provide a robust and scalable foundation for building a reliable, secure, and user-friendly document management system tailored to meet the academic and administrative needs of a college environment.

4.1.1 LANGUAGES USED IN THE PROJECT

1. HTML (HyperText Markup Language):

HTML is the core language used to create and structure content on the web. It defines the layout of a webpage by using various tags like <h1>, <p>, <a>, and . These tags help organize content such as headings, paragraphs, links, and sections. HTML doesn't manage the look or interactivity of the content; it simply structures it. For example, an image might be added using the <img> tag, and a hyperlink is created with the <a> tag. HTML forms the skeleton of a webpage, ensuring browsers know what content to display and how to organize it on the page.

2. CSS (Cascading Style Sheets):

CSS is used to style the HTML content by determining its appearance. While HTML structures the page, CSS makes it visually appealing. CSS controls elements such as colors, fonts, spacing, and layout. It can also make a website responsive, ensuring that it adjusts well to different screen sizes. CSS allows for advanced styling like animations, transitions, and grid systems. By separating content (HTML) from design (CSS), web developers can easily maintain and update the look of websites without altering the underlying structure.

3. JavaScript:

JavaScript is a programming language that adds interactivity and dynamic behavior to websites. It runs in the browser and allows for real-time updates, such as form validation, pop-up windows, and animations. JavaScript reacts to user actions, like clicks or keypresses, enabling features such as dropdown menus, interactive maps, and live content updates without reloading the page. It works seamlessly with HTML and CSS, manipulating the document structure and appearance in response to user interactions. JavaScript is essential for building modern, interactive, and dynamic websites and applications.

4. PHP (Hypertext Preprocessor):

PHP is a server-side scripting language used to create dynamic web pages and interact with databases. Unlike HTML, CSS, and JavaScript, which operate in the browser, PHP runs on the server, processing requests and generating content before it is sent to the browser. It is commonly used to manage forms, send emails, and fetch data from databases. For example, PHP can display personalized user content based on login details or retrieve product information from a database

for an e-commerce site. PHP is essential for building interactive websites that need to perform complex tasks behind the scenes.

4.1.2 IDE USED IN THE PROJECT

Visual Studio Code

Visual Studio Code (VS Code) is a free, open-source source code editor developed by Microsoft. It has quickly become one of the most popular code editors among developers due to its versatility, extensive feature set, and strong community support. Here's a detailed explanation of what Visual Studio Code is and its key features:

1. Code Editing: Visual Studio Code offers a powerful code editor with essential features like syntax highlighting, code completion, bracket matching, and code snippets. These tools enhance coding efficiency and help reduce errors.

2. IntelliSense: VS Code includes IntelliSense, providing context-aware code suggestions, auto-completions, and function signatures. This boosts productivity by making coding faster and more accurate.

3. Integrated Terminal: The editor features an integrated terminal, allowing developers to execute command-line tasks, run build scripts, and interact with servers directly from the editor, streamlining workflows.

4. Version Control: VS Code has built-in support for Git, offering tools for file comparison, viewing commit history, managing branches, and integrating with GitHub and other repositories.

5. Extensions: The VS Code marketplace offers a wide range of extensions to enhance its functionality. Developers can customize the editor with language support, debugging tools, themes, and more.

6. Debugging Support: The editor provides built-in debugging capabilities for various languages, including setting breakpoints, inspecting variables, and stepping through code, which simplifies bug detection and fixes.

7. Task Automation: VS Code supports task automation with tools like Grunt, Gulp, or npm scripts. Developers can define custom tasks for compiling code, testing, and deploying applications, reducing manual work.

8. Cross-Platform Accessibility: VS Code is available on Windows, macOS, and Linux, ensuring a consistent experience across different platforms, while also supporting an active developer community that contributes extensions and updates.

4.1.3 TOOLS USED IN THE PROJECT

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It stands for:

- **X** – Cross-platform (works on Windows, Linux, Mac)
- **A** – Apache (the web server)
- **M** – MySQL or MariaDB (the database system)
- **P** – PHP (server-side scripting language)
- **P** – Perl (another scripting language)

Purpose:

XAMPP allows developers to set up a **local web server environment** on their computer to build and test web applications before deploying them to a live server.

Key Features:

- **Apache**: Handles requests and serves web pages.
- **MySQL/MariaDB**: Stores data for your web applications (e.g., user info, articles).
- **PHP**: Runs server-side code to generate dynamic content.
- **phpMyAdmin**: A browser-based tool included with XAMPP for managing MySQL databases easily.

Use Case:

XAMPP is commonly used for learning web development, testing websites locally, or developing PHP/MySQL-based applications like WordPress.

4.2 ALGORITHM USED

1. **Authentication & Security :**

   Hashing Algorithms (e.g., SHA-256, bcrypt) Used for securely storing user passwords in the database.

2. **Request and Workflow Management**
   - Queue Data Structure (FIFO) For handling request-response cycles in order.
   - Workflow Algorithms (State Machine or DAG) Model the flow of document requests, approvals, and responses.
3. **Access Control**

   Role-Based Access Control (RBAC) Algorithm Grants permissions based on roles (student, faculty, higher faculty).

4.3 TESTING TECHNIQUES: IN CONTEXT OF PROJECT WORK

In a software project such as a Document Management System (DMS), the use of proper testing methodology is important to realize reliability, security, and effortless user experience.

Unit testing is among the most basic techniques in which individual functions or modules like the sign-up form, sign-up process, and request processing are tested. This ensures bugs are detected early on in the development stage.

Integration testing comes next, verifying how different elements (such as the login module and database) play off one another. As an example, it validates that user credentials inputted on the front end properly authenticate against the database.

System testing is finally employed to verify the fully and complete application, making sure all features — from document upload to request forwarding — play together as they should.

Another critical process is User Acceptance Testing (UAT), wherein real users like students or faculty use the system in real-world situations to determine if it suits their requirements.

Security testing is also critical in this kind of project to confirm that user information is safeguarded, passwords are stored safely, and there is no unauthorized access. Moreover, performance testing may also be utilized to assess how the system responds to multiple users or the uploading of huge documents. All these methods guarantee the system is working, is safe, and is deployable.

1. Functional Testing:

Functional testing that validates whether every functionality of a system works as per the specifications. In the scenario of a Document Management System (DMS) project, functional testing validates if all the facilities like user registration, login, document upload, request submission, and approval flows work as required. This testing targets user interactions and system responses, ensuring that inputs (such as form data) yield the intended outputs. For instance, when a student logs in with correct credentials, the system ought to provide access to the dashboard; upon incorrect credentials, it should present a suitable error message. In the same way, when a faculty member sends a document through forwarding, functional testing confirms that it gets to the intended higher authority. This kind of testing usually includes black-box testing methods, where the tester does not have to be familiar with the internal code but analyzes the application based on the user's point of view.

2. Performance Testing:

Performance testing is a type testing that evaluates how well a system performs under specific conditions, particularly in terms of speed, responsiveness, and stability. In a Document Management System (DMS) project, performance testing ensures that the application can handle multiple users simultaneously accessing features like login, document upload, request submission, and dashboard interactions without slowing down or crashing. It checks how the system behaves under normal load, as well as during peak usage, identifying any bottlenecks or

delays. For instance, if many students and faculty members are logged in at the same time, performance testing verifies that the system continues to respond efficiently and processes requests within an acceptable time frame. It also helps determine the maximum capacity the system can handle before performance degrades. This type of testing is crucial for maintaining a smooth user experience, especially in environments like colleges where many users may access the system at once.

3. User Acceptance Testing (UAT):

User Acceptance Testing (UAT) is the final phase of testing, where the end users (such as students, faculty, or administrative staff in the case of a Document Management System) test the system to ensure it meets their real-world needs and requirements. Unlike other forms of testing that focus on technical correctness or performance, UAT is concerned with whether the software is useful and usable from the user's perspective. During UAT, users perform typical tasks like registering, logging in, submitting document requests, reviewing approvals, and navigating the dashboard. The goal is to identify any gaps between the system's functionality and user expectations. If users encounter issues or find the system inefficient, those concerns are reported for fixes before the final deployment. UAT is crucial because it confirms that the system is ready for actual use and supports all intended workflows in a practical, user-friendly manner.

4. Security Testing:

Security testing in a Document Management System (DMS) is vital to protect sensitive documents and ensure data integrity. It begins with authentication and authorization testing, ensuring that only verified users can access the system and that their access is properly controlled based on roles and permissions. Input validation is also critical to prevent attacks like SQL injection or cross-site scripting (XSS). Data encryption should be tested both at rest and in transit, safeguarding documents from unauthorized access. Audit logging is essential for tracking user activities, such as document uploads or modifications, ensuring all actions are logged and secure for future auditing.

Access control tests ensure users can only interact with the documents they are authorized to view, while testing file uploads and downloads ensures that harmful files can't be introduced into the system. Additionally, backup and recovery protocols must be tested to prevent data loss, and

disaster recovery procedures should be evaluated to ensure system resilience. Penetration testing is another key aspect, simulating real-world attacks to identify vulnerabilities and weaknesses.

5. Error Handling & Exception Testing:

Error handling and exception testing are essential for ensuring that a Document Management System (DMS) remains stable, secure, and user-friendly under both normal and unexpected conditions. Effective error handling allows the system to gracefully manage failures without crashing, providing meaningful feedback to users with clear and informative error messages instead of vague system alerts. Additionally, errors should be logged securely, capturing details such as the error type and user actions, to help developers diagnose and fix issues. Exception testing focuses on verifying how the system handles exceptional conditions, such as invalid inputs, database failures, or network issues. The system should manage edge cases, such as large file uploads or database timeouts, without crashing and ensure that operations like file uploads or document access are not disrupted by permission issues or resource limitations. Testing should also include failover and recovery mechanisms, ensuring that the system can recover from failures such as network outages or service crashes, and that transactions can be rolled back to maintain data consistency. Negative testing ensures the system can handle invalid inputs or actions without exposing vulnerabilities, while API exception handling ensures that proper HTTP status codes and clear error messages are returned when the system encounters issues. Overall, thorough error handling and exception testing ensure the DMS remains resilient, secure, and reliable, providing a seamless experience even in the face of unexpected conditions.

CHAPTER 5 - RESULTS AND DISCUSSIONS

5.1 USER INTERFACE REPRESENTATION

The user interface (UI) of a Document Management System (DMS) is designed to provide an intuitive, efficient, and user-friendly experience, allowing users to easily manage, access, and collaborate on documents. The main dashboard is typically the first point of interaction, displaying an organized view of the user's documents, with options to upload, download, search, and manage files. Navigation is often streamlined with a side menu or top navigation bar, providing quick access to key sections such as document libraries, shared files, user settings, and notifications. The document viewing area is designed to allow users to open, preview, and edit documents directly within the interface, offering features like annotation tools, version control, and access to metadata.

The search function is usually prominent, enabling users to quickly locate documents through keywords, filters (such as file type, date, or author), and advanced options. The upload process is simplified, often featuring drag-and-drop functionality for adding files, with progress indicators to show upload status. User permissions are clearly reflected, ensuring that access controls are visible, and users can easily identify what actions they can perform (e.g., view, edit, delete). A robust error-handling system provides clear messages when something goes wrong, while maintaining the integrity of the UI. Overall, the design is clean, responsive, and customizable, ensuring that users can navigate and perform tasks with minimal friction, regardless of their technical expertise.

5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM

The project can be divided into several modules, each responsible for distinct functionalities. Below is a breakdown of the modules along with brief descriptions:

1. **Document Storage**

   The document storage and organization module is responsible for storing documents in an efficient and structured way. It includes features for creating folders or categories to group related documents, making it easier for users to organize their files logically. This module typically supports metadata tagging, allowing users to add custom tags or attributes (such as

author, document type, or department) to help with the classification and retrieval of documents. It also ensures that documents are stored securely, with support for encryption at rest to protect sensitive information.

2. **User Access and Permissions Management**

The user access and permissions management module controls who can view, edit, or delete documents within the system. It typically incorporates role-based access control (RBAC), allowing administrators to define roles (e.g., administrator, editor, viewer) and assign specific permissions to each role. This ensures that users can only perform actions on documents that they are authorized to access, providing a secure and controlled environment. The module also tracks user activities to maintain an audit trail, which is critical for compliance with data protection regulations.

3. **Document Security**

The document security and encryption module focuses on safeguarding sensitive information within the DMS. It ensures that documents are protected against unauthorized access both during transmission and when stored. This module includes features such as user authentication (e.g., password-based), role-based permissions, and activity logs, which record every action taken within the system for auditing purposes. Security measures may also extend to the physical infrastructure, such as secure data centers, to further protect the integrity and confidentiality of documents.

4. **Workflow and Document Approval**

The workflow and document approval module automates and streamlines the process of document creation, review, and approval. It enables organizations to define specific workflows that documents must follow before they are finalized. For example, a document may need to be reviewed by multiple people in a set order, and each person may be required to provide feedback or approval before it moves to the next stage. This module helps eliminate bottlenecks in document processing and ensures that the right stakeholders are involved in

decision-making. It can also generate notifications and reminders to keep the process moving efficiently.

5. Database

The database in a Document Management System (DMS) serves as the central repository where all documents, metadata, user data, and system configurations are stored and managed. It ensures that documents are organized, easily accessible, and secure, allowing users to efficiently search for and retrieve files based on various criteria such as document type, metadata, or content. The database stores metadata, such as document titles, authors, creation dates, and access permissions, which helps in classifying and indexing documents for quick retrieval. A well-designed database also plays a key role in managing version control, keeping track of document changes, and ensuring that users can access previous versions when necessary. Security is a critical aspect of the database, with encryption often implemented to protect sensitive data both at rest and in transit. Additionally, the database should be optimized for performance to handle large volumes of documents and simultaneous user queries without significant delays. It is also important for the database to integrate seamlessly with other components of the DMS, such as workflow automation, user access control, and audit trails, ensuring that all document-related actions are recorded and compliant with relevant regulatory requirements. Proper backup and disaster recovery mechanisms must also be in place to ensure that the database can be restored.

5.2 SNAPSHOTS OF SYSTEM

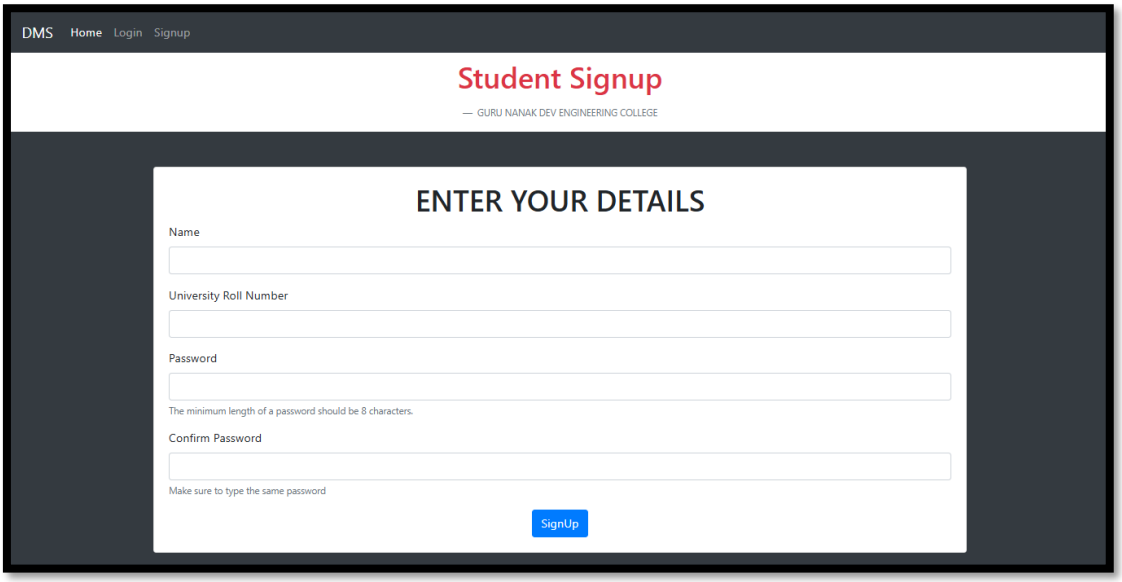

Figure 5.1 Student Signup Page

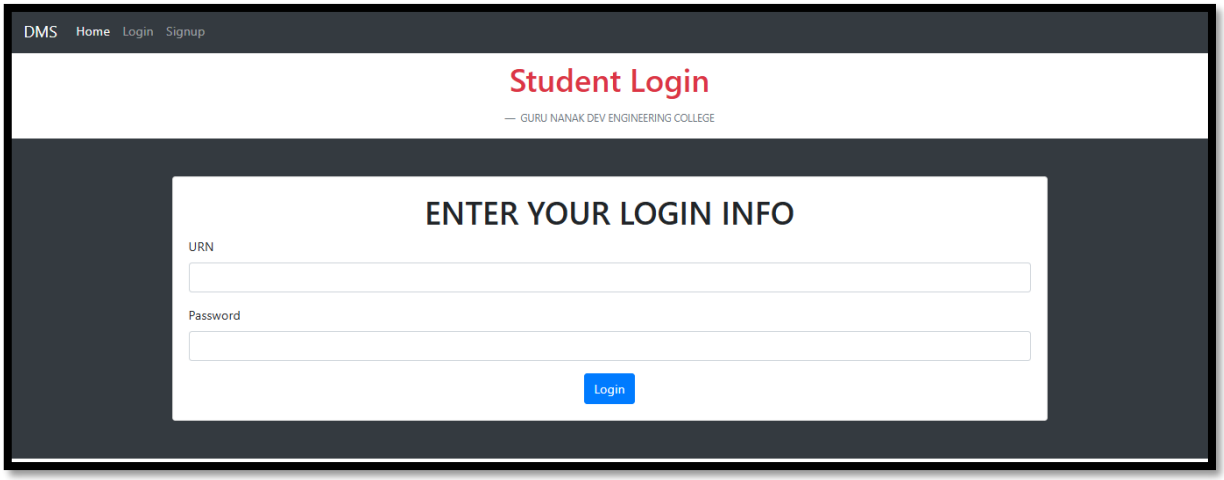

Figure 5.2 Student Login Page

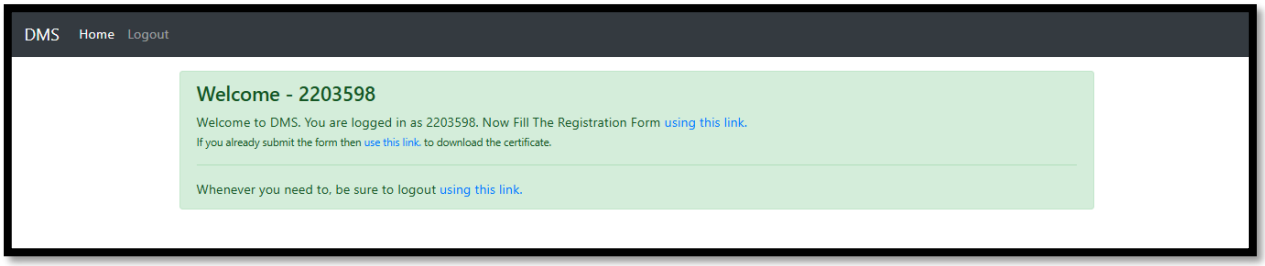

Figure 5.3 Student Dashboard

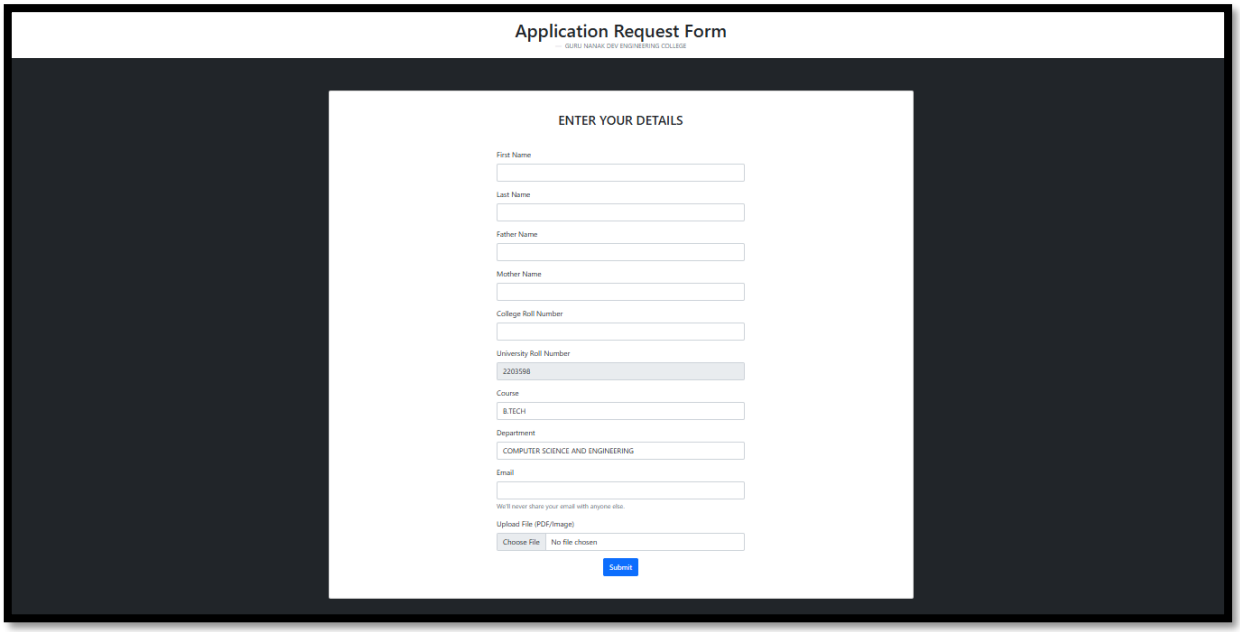

Figure 5.4 Student Application Form

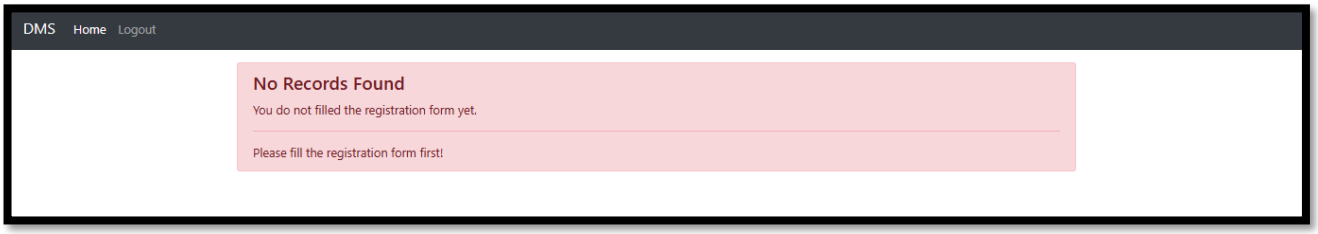

Figure 5.5 Application Status Page

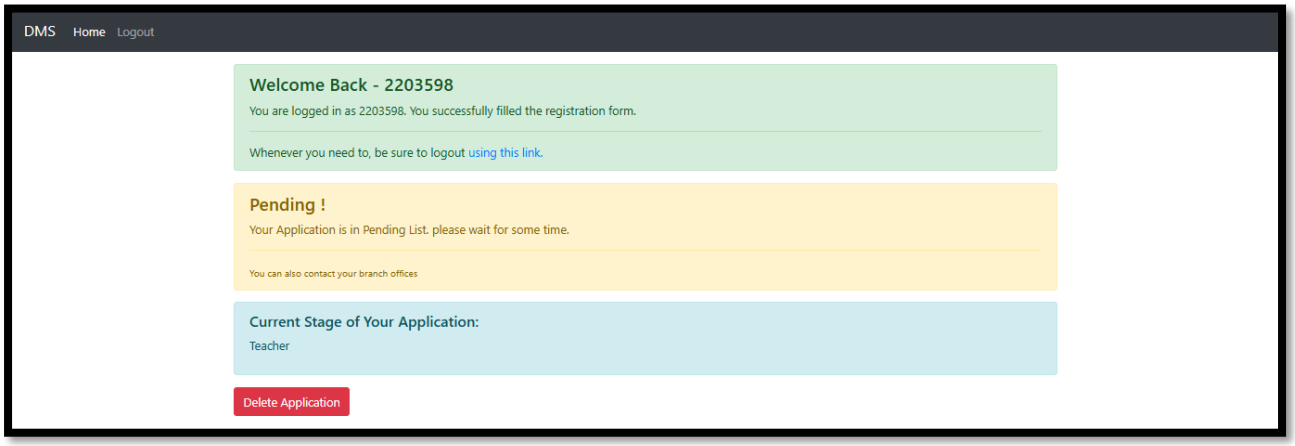

Figure 5.6 Application Pending Status

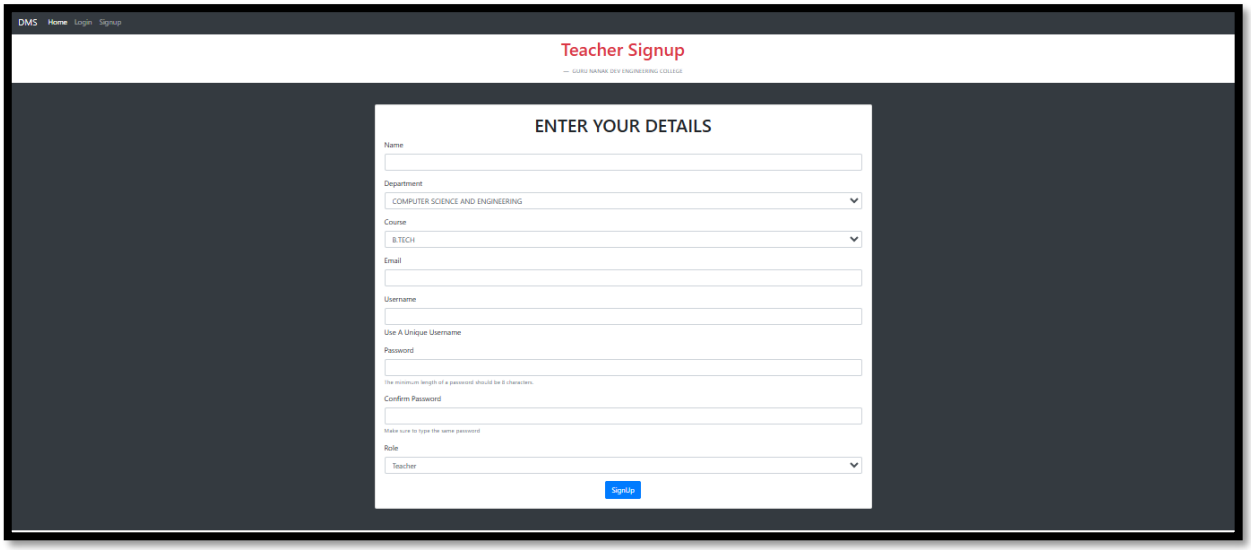

Figure 5.7 Teacher Signup Page

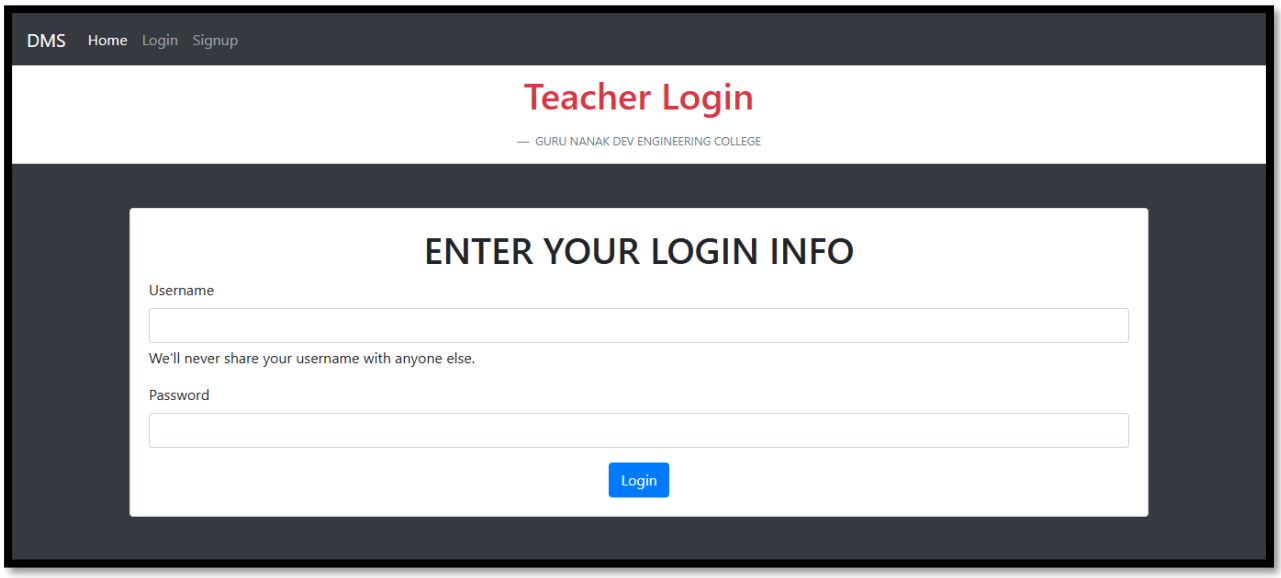

Figure 5.8 Teacher Login Page

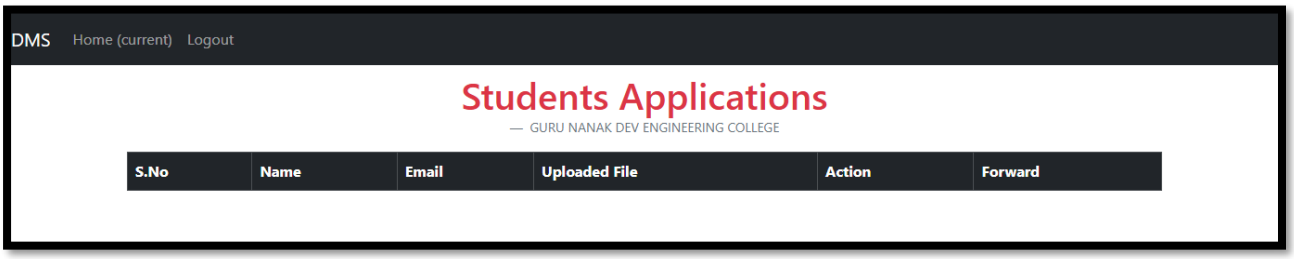

Figure 5.9 Student Applications Page

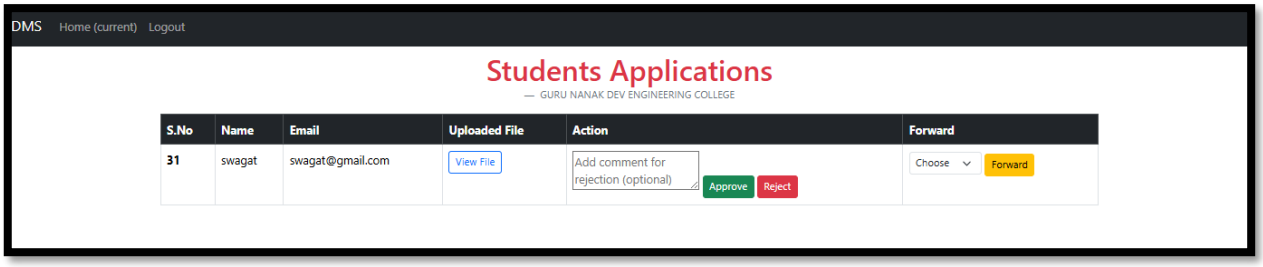

Figure 5.10 Applied Applications list

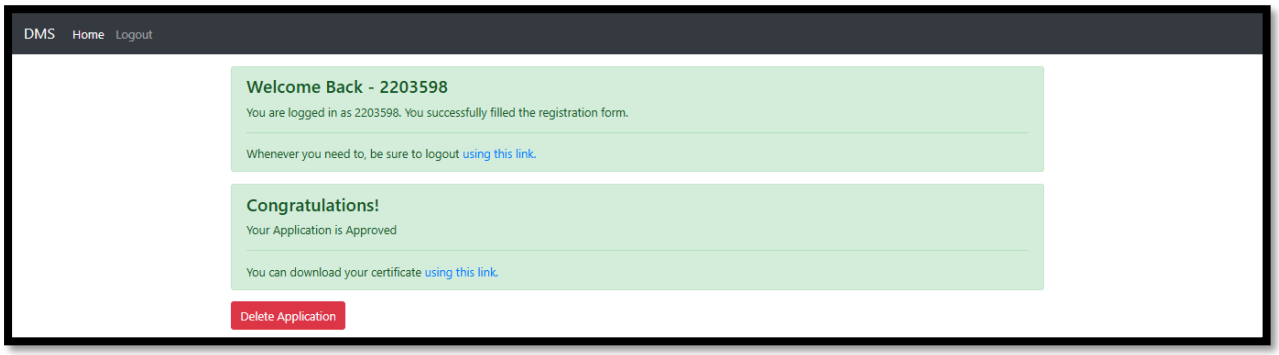

Figure 5.11 Application Approved

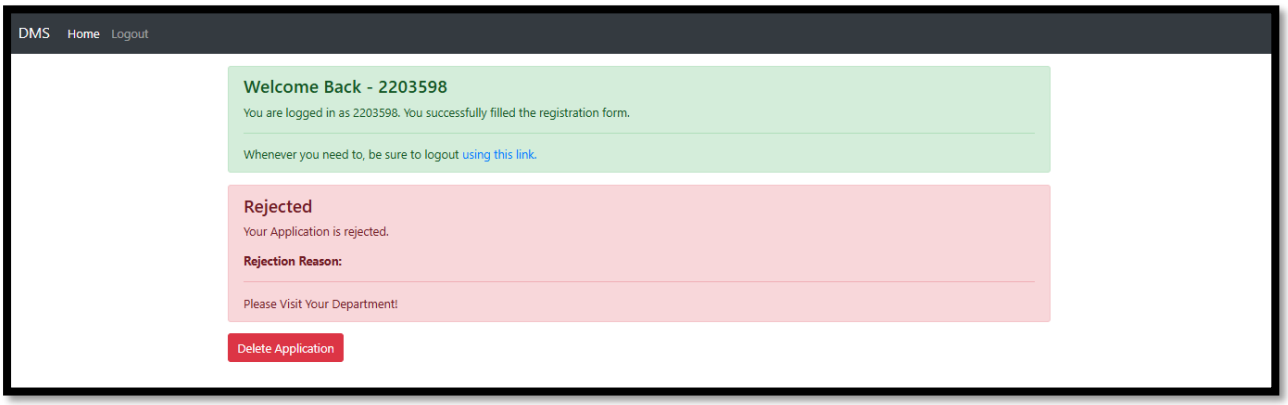

Figure 5.12 Application Rejected

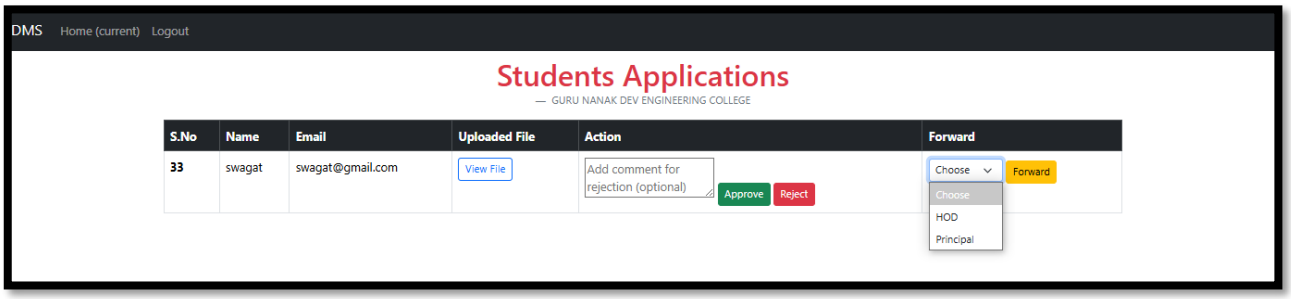

Figure 5.13 Application Forward Options

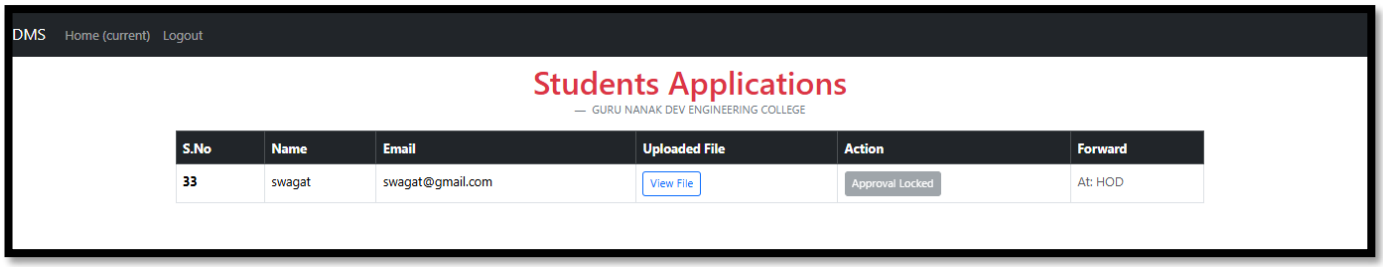

Figure 5.14 Application Forwarded to HOD

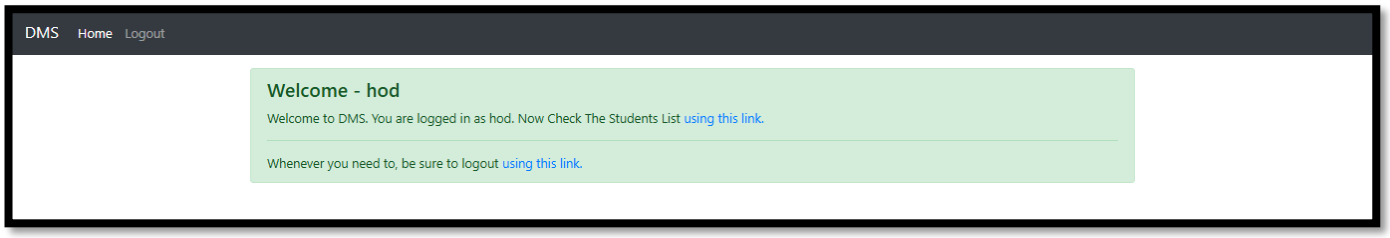

Figure 5.15 HOD Dashboard

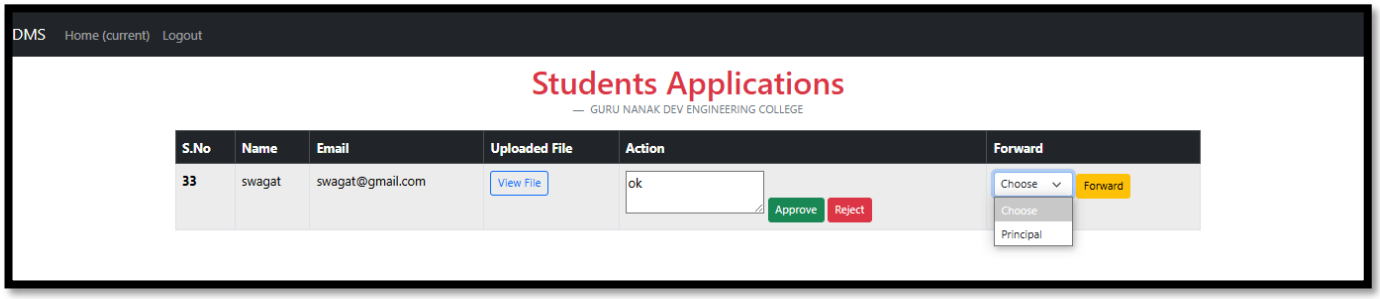

Figure 5.16 Forward Option At HOD Dashboard

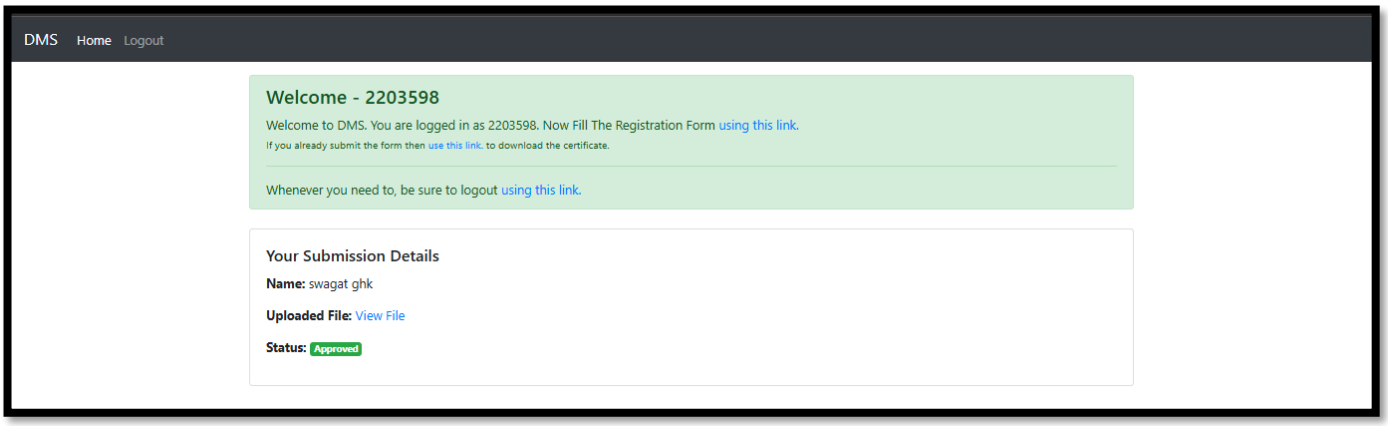

Figure 5.17 File Status At Student Dashboard

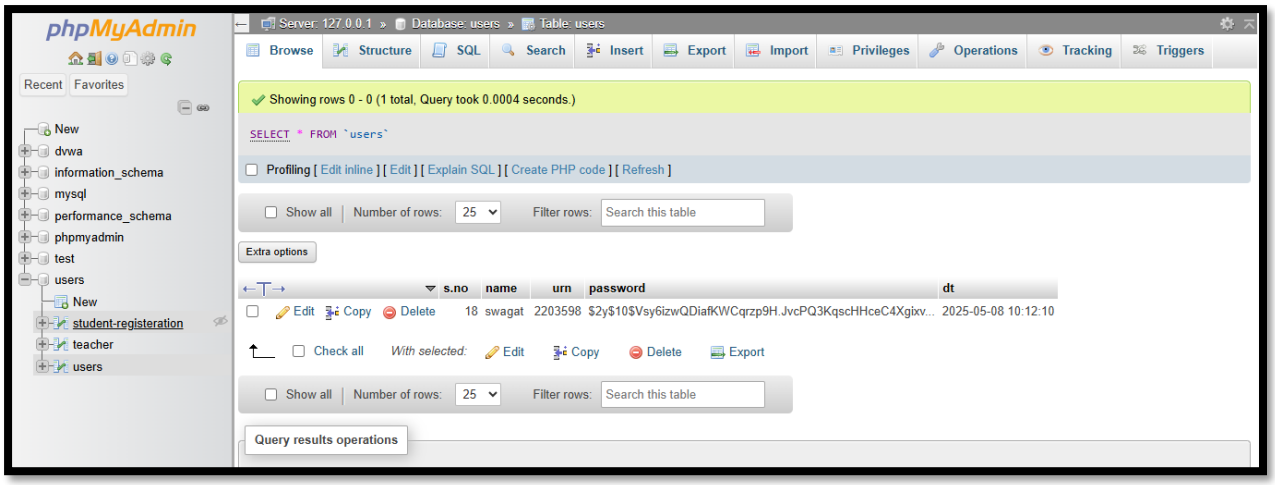

Figure 5.18 User Database

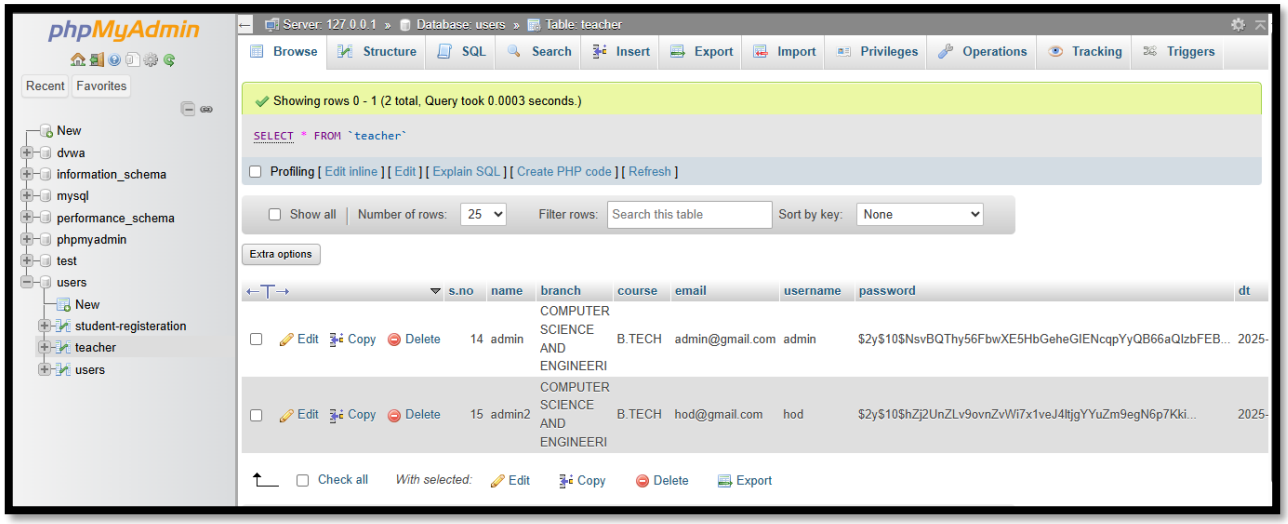

Figure 5.129 Teacher Database

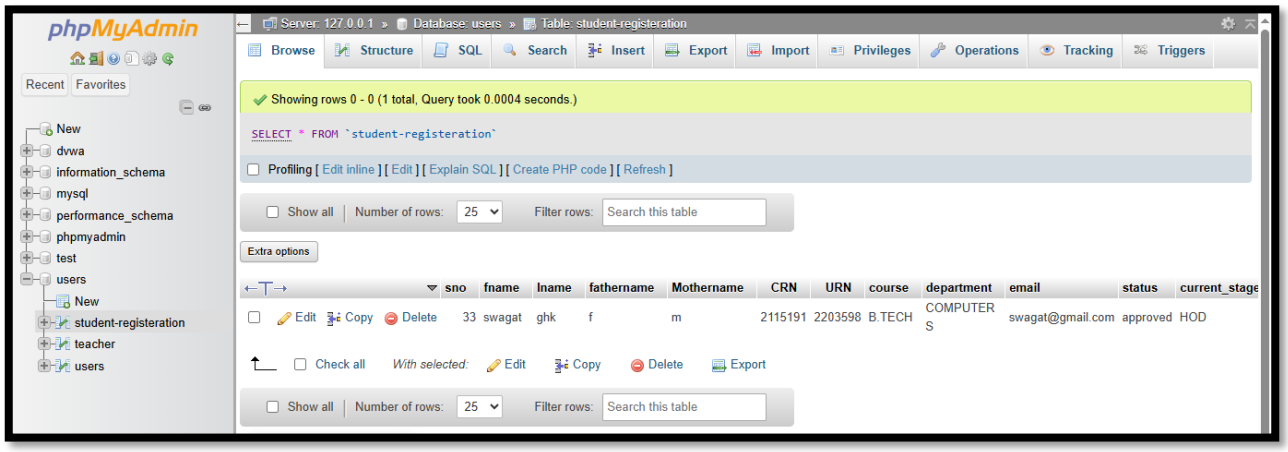

Figure 5.20 Student Applications Database

CHAPTER 6 - CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In conclusion, a Document Management System (DMS) plays a vital role in modern organizations by streamlining the storage, management, and retrieval of documents, ultimately improving efficiency, collaboration, and security. As organizations continue to generate and manage vast amounts of digital information, having a robust and well-designed DMS becomes increasingly essential for ensuring that documents are not only securely stored but also easily accessible when needed. By integrating various modules such as document storage, retrieval, version control, access management, and collaboration, a DMS enhances workflow efficiency and enables teams to collaborate seamlessly, regardless of their location.

One of the key strengths of a DMS is its ability to ensure document security and compliance. With strict access controls, encryption, and auditing features, a DMS safeguards sensitive information from unauthorized access and ensures that documents are handled in accordance with legal and regulatory requirements. Additionally, version control and document retention policies within a DMS ensure that organizations can track document changes over time, maintain historical records, and easily recover previous versions if necessary.

The user interface (UI) of a DMS is another critical aspect that significantly influences user adoption. A clean, intuitive, and responsive UI helps users navigate the system with ease, reducing the learning curve and improving overall productivity. A well-designed UI, combined with powerful search and retrieval capabilities, ensures that users can quickly locate and access the documents they need, further enhancing operational efficiency.

Furthermore, backup and disaster recovery mechanisms integrated within a DMS provide peace of mind, ensuring that documents are protected against data loss due to system failures, accidental deletion, or cyber-attacks. By automatically backing up documents and providing a reliable recovery process, organizations can maintain business continuity and mitigate the impact of potential data disasters.

Finally, as organizations increasingly adopt cloud-based solutions, the scalability, integration, and accessibility of a DMS become even more important. A cloud-based DMS offers the flexibility of remote access, real-time collaboration, and seamless integration with other enterprise systems, making it easier for teams to work together across locations and time zones.

As technology continues to evolve, the capabilities of DMS platforms will continue to expand, offering even more advanced features such as artificial intelligence-driven document categorization and automated workflows.

Overall, a well-implemented Document Management System is not just a tool for organizing files but a critical asset that drives operational efficiency, security, compliance, and collaboration within an organization.

6.2 FUTURE SCOPE

The future scope of Document Management Systems (DMS) is expansive and evolving, driven by advancements in technology, increasing organizational needs for collaboration, security, and compliance, as well as the growing demand for automation and efficiency. As digital transformation accelerates across industries, the role of DMS platforms will continue to evolve, integrating new capabilities that align with these changes and address emerging challenges.

One significant trend is the integration of artificial intelligence (AI) and machine learning (ML) in DMS platforms. These technologies will automate document classification, metadata tagging, and content analysis, making it easier for users to locate relevant documents. AI algorithms will enable systems to understand the context of a document and intelligently categorize it based on its content, improving search functionality and user experience. Additionally, AI can help automate the extraction of key data points from documents, such as invoices, contracts, or forms, reducing manual intervention and enhancing operational efficiency.

Another area of growth is cloud-based DMS solutions. Cloud computing has already begun to transform how organizations manage their documents by providing remote access, scalability, and cost-effectiveness. The future of cloud-based DMS platforms lies in increased integration with other cloud services such as project management tools, customer relationship management (CRM) systems, and enterprise resource planning (ERP) platforms. This will create a more unified digital ecosystem, allowing organizations to centralize their data, collaborate across functions, and streamline business processes. Furthermore, with the growing adoption of hybrid and multi-cloud environments, DMS platforms will likely offer more flexibility in terms of data storage and access, providing organizations with more control over where and how their documents are stored.

Blockchain technology also presents a promising future for DMS, particularly in enhancing security and compliance. Blockchain can be used to create immutable records of document

transactions, ensuring the integrity and authenticity of documents, which is particularly valuable for industries that require high levels of data integrity, such as legal, financial, and healthcare sectors. Blockchain's decentralized nature can also facilitate more secure document sharing, where stakeholders have greater control and transparency over document exchanges.

As organizations increasingly focus on data security and privacy, the future of DMS will also involve more sophisticated encryption techniques, enhanced user authentication, and compliance features. DMS platforms will need to stay up to date with global regulations such as GDPR, HIPAA, and others, implementing stronger compliance tracking and reporting functionalities.

Lastly, collaborative features will become even more advanced. Real-time editing, seamless version control, and better integration with communication tools will enhance teamwork and collaboration, allowing users to work on documents from anywhere, on any device, and in any environment. This is particularly important as the future of work increasingly leans toward remote and distributed teams.

In conclusion, the future of Document Management Systems will be shaped by technological innovations such as AI, blockchain, and cloud computing, leading to more intelligent, secure, and collaborative platforms that streamline document processes and drive business efficiency. These advancements will provide organizations with greater flexibility, control, and scalability, transforming the way documents are managed, accessed, and shared across industries.

BIBLIOGRAPHY

1. https://www.google.com/
2. https://chatgpt.com/
3. https://www.geeksforgeeks.org/
4. https://getbootstrap.com/
5. https://www.apachefriends.org/