

ADEKUNLE AJASIN UNIVERSITY, AKUNGBA AKOKO

NAME: OKITI PETER MOYINOLUWA

MATRIC NUMBER: 190404055

DEPARTMENT: COMPUTER SCIENCE

PROJECT TOPIC:

DIGITAL MEMO MANAGEMENT SYSTEM FOR UNIVERSITY OFFICES

CHAPTER 1 & 2

PROJECT SUPERVISOR

MR AKINWUMI A.O.

**CERTIFICATION**

**I, OKITI PETER MOYINOLUWA. Hereby declare that this report has been carried out by me under the supervision of MR AKINWUMI A.O. It has not been presented for award of any degree in any institution. All sources of information are specifically acknowledged by means of reference.**

**Supervisor**

**MR AKINWUMI A.O. -----------------------**

**Signature/Date**

**OKITI PETER MOYINOLUWA** **-----------------------**

**Signature/Date**

# Table of Contents

[Table of Contents ii](#_Toc179313514)

[Chapter One: Introduction 1](#_Toc179313515)

[1.1 Background of Study 1](#_Toc179313516)

[1.2 Statement of Problem 2](#_Toc179313517)

[1.3 Aim & Objectives 3](#_Toc179313518)

[1.4 Methodology 3](#_Toc179313519)

[1.5 Scope of Study 5](#_Toc179313520)

[1.6 Expected Contribution to Knowledge 5](#_Toc179313521)

[1.7 Definition of Terms 5](#_Toc179313522)

[Chapter Two: Literature Review 7](#_Toc179313523)

[2.1 Introduction 7](#_Toc179313524)

[2.2 Analysis of Existing Systems 7](#_Toc179313525)

[2.2.1 Overview of current digital memo management systems 7](#_Toc179313526)

[2.2.2 Comparative analysis of different digital memo management systems 8](#_Toc179313527)

[2.3 Research Concepts 8](#_Toc179313528)

[2.3.1 Discussion on relevant theories in digital communication and document management 8](#_Toc179313529)

[2.3.2 Methodologies for evaluating and improving digital office management systems 9](#_Toc179313530)

[2.4 Technological Advances 10](#_Toc179313531)

[2.4.1 Review of recent advancements in technology applicable to memo management systems 10](#_Toc179313532)

[2.4.2 Emerging trends in user interface design and user experience optimization 10](#_Toc179313533)

[2.5 Case Studies 11](#_Toc179313534)

[2.5.1 Examination of successful implementations of digital memo management systems in various organizations 11](#_Toc179313535)

[2.5.2 Identification of lessons learned and best practices from real-world implementations 11](#_Toc179313536)

[2.6. Conclusion 12](#_Toc179313537)

[2.6.1 Summary of key findings from the literature review 12](#_Toc179313538)

[2.6.2 Implications for the development and implementation of the digital memo management system 12](#_Toc179313539)

[2.6.3 Recommendations for future research areas and potential enhancements to existing systems 12](#_Toc179313540)

[Chapter 3: Methodology 14](#_Toc179313541)

[3.1 Introduction 14](#_Toc179313542)

[3.2 System Requirements and Specifications 14](#_Toc179313543)

[3.2.1 Functional Requirements 14](#_Toc179313544)

[3.2.2 Non-Functional Requirements 15](#_Toc179313545)

[3.3 Frontend Development 16](#_Toc179313546)

[3.3.1 Technologies Used 16](#_Toc179313547)

[3.3.2 Main Pages 17](#_Toc179313548)

[3.3.3 User Interface (UI) Design 18](#_Toc179313549)

[3.4 Backend Development 18](#_Toc179313550)

[3.4.1 Technologies Used 18](#_Toc179313551)

[3.4.2 Key Backend Functionalities 19](#_Toc179313552)

[3.5 Database Design 19](#_Toc179313553)

[3.5.1 Technologies Used 20](#_Toc179313554)

[3.5.2 Database Tables 20](#_Toc179313555)

[3.5.3 MySQL Schema Design 21](#_Toc179313556)

[3.6 Security Measures 22](#_Toc179313557)

[3.7 Email API Integration 23](#_Toc179313558)

[3.7.1 Technology Used 23](#_Toc179313559)

[3.7.2 Email API Setup 23](#_Toc179313560)

[3.7.3 Password Reset Process 23](#_Toc179313561)

[3.8 Hosting and Deployment 24](#_Toc179313562)

[3.8.1 Technology Used 24](#_Toc179313563)

[3.8.2 Deployment Steps 24](#_Toc179313564)

[3.9 Error Handling and Debugging 26](#_Toc179313565)

[3.9.1 Development-Level Debugging 26](#_Toc179313566)

[Chapter 4: System Interface and Functionality 28](#_Toc179313567)

[4.1 Introduction 28](#_Toc179313568)

[4.2 Login Page 28](#_Toc179313569)

[4.3 Sign-Up Page 30](#_Toc179313570)

[4.4 Forgot Password Page 31](#_Toc179313571)

[4.5 Dashboard Page 33](#_Toc179313572)

[4.6 Create Memo Page 34](#_Toc179313573)

[4.7 Success Page 36](#_Toc179313574)

[Chapter 5: Discussion, Conclusion, and Recommendations 37](#_Toc179313575)

[5.1 Discussion 37](#_Toc179313576)

[5.2 Conclusion 38](#_Toc179313577)

[5.3 Recommendations 39](#_Toc179313578)

[References 40](#_Toc179313579)

# Chapter One: Introduction

## 1.1 Background of Study

In many university offices, the traditional paper-based memo process is still prevalent. This method involves creating memos manually, distributing physical copies, and tracking responses and approvals through physical files. Such a process is often slow, prone to errors, and difficult to manage, especially in a large institution. The increasing digitization of administrative tasks presents an opportunity to improve memo management by transitioning to a digital system. A digital memo management system can streamline the creation, routing, and tracking of memos, leading to greater efficiency and productivity.

The reliance on paper-based memos in university offices poses several challenges. Physical documents are susceptible to being misplaced, damaged, or delayed, which can disrupt communication and administrative workflows (Paryasto, 2019). In addition, the manual handling of memos often results in inconsistent tracking and record-keeping, making it difficult to monitor the progress and approval status of important documents (Asogwa, 2013). These inefficiencies are compounded in large institutions where the volume of memos and the number of departments involved further complicate the management process (Cao et al., 2020).

The move towards digital systems in administrative functions is part of a broader trend of digitization aimed at enhancing operational efficiency and reducing reliance on physical paperwork (Heeks, 2006). Digital memo management systems offer a centralized platform where memos can be created, routed, and tracked electronically, eliminating the need for physical copies and manual processes (Garcia & Pinzon, 2020). Such systems integrate features like electronic signatures, automated notifications, and digital archiving, which streamline workflows and ensure that all memos are accounted for and easily retrievable (Bounabat et al., 2018).

Implementing a digital memo management system in university offices can significantly improve the speed and accuracy of memo dissemination. Electronic routing ensures that memos reach their intended recipients promptly and that responses can be tracked in real-time (Chukwuemeka et al., 2019). Moreover, digital systems enhance transparency by providing audit trails that document the entire lifecycle of a memo, from creation to final approval (Tolu, 2021). This level of transparency is crucial for accountability and compliance, particularly in large organizations where oversight is necessary (Oyelere et al., 2020).

The transition to digital memo management aligns with contemporary trends in e-governance and digital transformation, which emphasize the use of technology to improve public administration and service delivery (Eom, 2020). Universities, as complex organizations with diverse administrative needs, stand to benefit greatly from adopting digital solutions that streamline their operations (Hicks & Given, 2022). By leveraging digital tools, universities can reduce administrative burdens, enhance communication efficiency, and foster a more responsive and agile administrative environment (Yadav & Singh, 2021).

## 1.2 Statement of Problem

The traditional paper-based memo process in university offices is marked by significant inefficiencies and operational challenges. One of the primary issues is the time-consuming nature of creating, distributing, and tracking paper memos. Manual handling of memos leads to substantial delays as documents need to be physically transported between departments and individuals, which is particularly problematic in large institutions with extensive administrative structures (Cao et al., 2020). These delays can hinder timely decision-making and disrupt the flow of information, adversely affecting the overall administrative efficiency (Garcia & Pinzon, 2020).

Moreover, paper-based memos are prone to being misplaced or lost, resulting in critical information gaps and potential communication breakdowns (Asogwa, 2013). The physical nature of these documents means they are susceptible to damage and loss, whether through accidental misplacement or external factors such as fire or water damage (Paryasto, 2019). This lack of reliability in document management can lead to repeated efforts and redundancy, further exacerbating inefficiencies.

Tracking the progress and status of paper memos is another significant challenge. With no centralized system to monitor the lifecycle of a memo, it becomes difficult to ascertain its current state, whether it has been received, reviewed, or approved (Chukwuemeka et al., 2019). This lack of transparency can lead to accountability issues, as it is challenging to hold individuals or departments responsible for delays or mishandling without clear records (Tolu, 2021). The absence of a robust tracking mechanism also means that important memos can get stuck at various stages, creating bottlenecks in administrative processes.

This project aims to address these issues by developing a digital memo management system that automates the creation, distribution, and tracking of memos. By leveraging digital tools, the system will streamline these processes, ensuring memos are managed more effectively and efficiently. The proposed system will include features such as electronic routing, automated notifications, and digital archiving, which will eliminate the delays associated with physical transportation and improve the reliability of memo delivery and tracking (Bounabat et al., 2018). Additionally, the digital system will enhance transparency by providing real-time updates and audit trails, making it easier to monitor the progress of each memo and hold relevant parties accountable (Eom, 2020).

## 1.3 Aim & Objectives

**Aim:** The aim of this project is to develop a user-friendly, efficient, and secure digital memo management system specifically designed for university offices.

**Objectives:**

1. To develop and implement a digital memo management system for university offices. Using HTML CSS and JAVASCRIPT for the front end and the backend will be built using PHP.
2. To evaluate the effectiveness of the digital memo management system.
3. To identify and address potential challenges associated with the transition from paper-based to digital memo systems.

## 1.4 Methodology

**I. System Design**

**Front-End:** The front-end of the proposed digital memo management system will be developed using (HTML5, CSS3, and JavaScript).

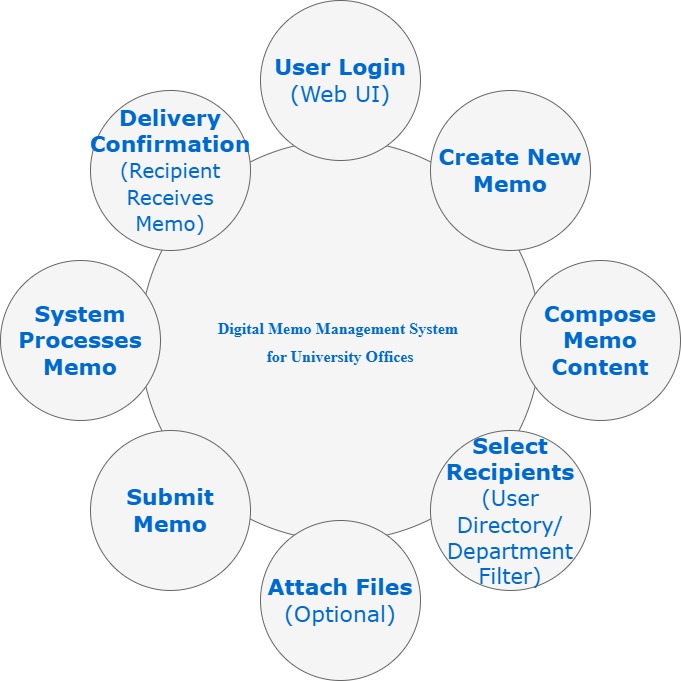
**Back-End:** The back-end will be built using PHP, providing a robust and scalable server-side environment. The system will use RESTful Plugins to facilitate communication between the front-end and back-end components.

**Database:** A relational database management system (RDBMS) such as MySQL or PostgreSQL will be used to store and manage memo data. This choice ensures data consistency, integrity, and security. The database will handle all CRUD (Create, Read, Update, Delete) operations required by the system.

**II. Architecture**

The architecture of the digital memo management system will be based on a client-server model. The client-side application (front-end) will interact with the server-side application (back-end) through RESTful APIs. The back-end will manage business logic, process requests, and interact with the database.

**System Architecture Diagram:**



## 1.5 Scope of Study

The study will focus on developing a digital memo management system specifically for university offices. It will include the design, implementation, and testing phases, but will not cover the integration with other external systems beyond basic university IT infrastructure. The study will also not address potential changes in office workflows or staff training in depth.

## 1.6 Expected Contribution to Knowledge

The expected contribution is the introduction of a digital memo for university office, using Adekunle Ajasin University as a case study. This will transition from a paper-based memo distribution to a digital and automated distribution of memo.

## 1.7 Definition of Terms

**Memo:** A memo, short for memorandum, is a written message commonly used in business or formal communication to convey information, requests, or updates within an organization. Memos are typically concise and focused on a specific topic, serving as an internal communication tool to inform or direct employees and departments (BusinessDictionary, 2024).

**Digital Memo Management System:** A digital memo management system is a software application designed to create, route, and track memos electronically. This system replaces traditional paper-based memo processes, enabling users to generate memos, distribute them to appropriate recipients, and monitor their status and responses through an integrated digital platform. The goal of such a system is to enhance efficiency, accuracy, and accountability in memo management (Garcia & Pinzon, 2020).

**Authentication:** Authentication is the process of verifying the identity of a user or system to ensure that access to resources and information is granted only to legitimate entities. In the context of a digital memo management system, authentication typically involves the use of credentials such as usernames, passwords, biometric data, or security tokens to confirm that users are who they claim to be before allowing them to access the system's features and data (Andress, 2019).

**Data Encryption:** Data encryption is the method of converting data into a coded form to prevent unauthorized access. Encryption ensures that sensitive information remains confidential and secure, even if it is intercepted or accessed by unauthorized individuals. In a digital memo management system, data encryption is used to protect the contents of memos and other critical information both in transit and at rest, ensuring that only authorized users can read and interact with the data (Stallings, 2020).

# Chapter Two: Literature Review

## 2.1 Introduction

In today's digital environment, the importance of efficient office management systems cannot be overstated. Digital office management systems play a crucial role in streamlining administrative processes, enhancing communication, and improving overall productivity within organizations (Garcia & Pinzon, 2020). By leveraging technology to automate tasks, organize information, and facilitate collaboration, these systems enable businesses and institutions to operate more effectively and competitively in the modern era.

The purpose of the literature review in this context is to provide a comprehensive understanding of existing research and related works in the field of digital office management systems. Specifically, the review aims to inform the development of a digital memo management system tailored to the unique needs and challenges of university offices. Through analysing current systems, exploring relevant theories and methodologies, reviewing technological advancements, and examining case studies, the literature review will serve as a foundation for designing and implementing an effective digital memo management solution (Pressman & Maxim, 2020).

## 2.2 Analysis of Existing Systems

### 2.2.1 Overview of current digital memo management systems

Digital memo management systems have emerged as essential tools for organizations seeking to streamline their communication and document management processes. These systems typically offer a range of features and functionalities designed to facilitate the creation, distribution, and tracking of memos within an organization. Common features include the ability to create memos electronically, route them to appropriate recipients, track their status and responses, and archive them for future reference (Garcia & Pinzon, 2020). Additionally, many digital memo systems offer collaboration features such as commenting, version control, and integration with other office productivity tools.

The advantages of digital memo systems over traditional paper-based methods are numerous. One of the primary benefits is the elimination of physical paperwork, which reduces the environmental impact of printing and storing paper memos (Cao et al., 2020). Digital systems also offer greater accessibility, allowing users to access memos from anywhere with an internet connection, rather than being limited to physical copies in the office. Furthermore, digital memo systems often provide improved security features, such as encryption and access controls, to protect sensitive information from unauthorized access (Stallings, 2020).

Despite these advantages, digital memo management systems also face several limitations and challenges. One common challenge is user adoption, as employees may be resistant to change or unfamiliar with the new system's features (Chow & Cao, 2008). Additionally, integrating digital memo systems with existing workflows and IT infrastructure can be complex and time-consuming, requiring careful planning and coordination (Bass, Clements & Kazman, 2012). Furthermore, ensuring data integrity and compliance with privacy regulations is critical, as digital memos may contain sensitive information that needs to be protected (Andress, 2019).

### 2.2.2 Comparative analysis of different digital memo management systems

A comparative analysis of different digital memo management systems reveals a diverse landscape of solutions catering to various organizational needs and preferences. Commercial solutions, such as Microsoft Outlook and Google Workspace, offer comprehensive memo management features integrated into larger office productivity suites. These solutions provide a familiar user experience and seamless integration with other tools, making them popular choices for organizations of all sizes (Pressman & Maxim, 2020).

In contrast, open-source and custom-developed systems provide greater flexibility and customization options but require more technical expertise to deploy and maintain. Open-source solutions like OpenKM and Alfresco offer powerful document management capabilities at a lower cost, but may lack some of the advanced features found in commercial solutions (Fowler, 2018). Custom-developed systems, built in-house or by third-party developers, can be tailored to specific organizational requirements but require significant investment in development and ongoing support (Pohl & Rupp, 2011).

## 2.3 Research Concepts

### 2.3.1 Discussion on relevant theories in digital communication and document management

In digital communication and document management, several theoretical frameworks provide insights into the design and implementation of effective memo management systems. One such theory is information theory, which focuses on the transmission, storage, and processing of information in communication systems (Stallings, 2020). Applied to digital memo systems, information theory offers a framework for understanding how information is encoded, transmitted, and decoded within the system. Concepts such as entropy, redundancy, and channel capacity can inform the design of efficient memo routing algorithms and data compression techniques to optimize system performance and bandwidth utilization (Sommerville, 2016).

Organizational communication models also play a crucial role in shaping memo management practices within institutions. Models such as the transactional model of communication and the communication accommodation theory provide insights into the dynamics of communication within organizations and the factors that influence message interpretation and reception (Paryasto, 2019). By understanding the communication patterns and preferences of different stakeholders, digital memo systems can be designed to accommodate diverse communication styles and facilitate effective message delivery and reception. Additionally, theories of organizational structure and culture can inform the design of memo workflows and approval processes to align with organizational norms and values (Eom, 2020).

### 2.3.2 Methodologies for evaluating and improving digital office management systems

Evaluating and improving digital office management systems requires robust methodologies that address both technical and user-centric aspects of system design and implementation. User-centered design approaches emphasize the importance of involving end-users in the design process to ensure that the system meets their needs and preferences (Brown, 2018). Techniques such as usability testing, user interviews, and personas development can provide valuable insights into user behaviors, preferences, and pain points, which can then be used to inform the design of intuitive and user-friendly memo management interfaces (Myers, Sandler & Badgett, 2011).

Agile and iterative development methodologies offer an alternative approach to traditional waterfall development models, allowing for rapid system evolution and adaptation to changing requirements (Boehm, 1988). By breaking down the development process into small, manageable increments, agile methodologies enable developers to quickly prototype, test, and iterate on system features based on user feedback (Humble & Farley, 2010). This iterative approach is particularly well-suited to the development of memo management systems, where user engagement and feedback are critical for ensuring system usability and effectiveness (Beck et al., 2001).

Incorporating these research concepts and methodologies into the development of digital memo management systems can help ensure that the resulting systems are not only technically robust but also user-friendly, efficient, and aligned with organizational communication practices and goals.

## 2.4 Technological Advances

### 2.4.1 Review of recent advancements in technology applicable to memo management systems

Recent advancements in technology have revolutionized memo management systems, offering new capabilities and opportunities for optimization. One such advancement is cloud computing, which has transformed the way memo management systems are deployed and accessed (Pressman & Maxim, 2020). Cloud-based memo management solutions leverage remote servers hosted on the internet to store data and run applications, offering benefits such as scalability, flexibility, and accessibility. By storing memos and related data in the cloud, organizations can ensure that information is available anytime, anywhere, and can easily scale their memo management systems to accommodate growing user bases and workloads (Pohl & Rupp, 2011).

Another significant technological advancement impacting memo management systems is artificial intelligence (AI) and machine learning (ML). AI and ML algorithms can automate memo routing and classification processes, reducing the need for manual intervention and improving system efficiency (Stallings, 2020). For example, AI-powered systems can analyze the content of memos and automatically route them to the appropriate recipients based on predefined rules or machine learning models trained on historical data. Additionally, ML algorithms can be used to classify incoming memos into relevant categories, making it easier for users to search for and retrieve specific documents (Fowler, 2018).

### 2.4.2 Emerging trends in user interface design and user experience optimization

User interface (UI) design and user experience (UX) optimization are critical aspects of memo management systems, as they directly impact user adoption and satisfaction. One emerging trend in UI design is mobile-first design principles, which prioritize the development of user interfaces optimized for mobile devices (Brown, 2018). With the increasing use of smartphones and tablets in the workplace, mobile-friendly memo management interfaces are essential for accommodating diverse user preferences and devices. Mobile-first design principles focus on simplicity, clarity, and responsiveness, ensuring that memos can be accessed and managed efficiently on smaller screens without sacrificing usability (Sommerville, 2016).

Another emerging trend in memo management UI design is the integration of voice recognition and natural language processing (NLP) technologies. Voice recognition allows users to create and interact with memos using spoken commands, while NLP enables the system to understand and process natural language inputs (Chow & Cao, 2008). By integrating these technologies into memo management systems, users can dictate memos, search for information, and perform other tasks using voice commands, making the memo management process more intuitive and efficient (Myers, Sandler & Badgett, 2011).

Incorporating these technological advancements and emerging trends into memo management systems can enhance their functionality, usability, and overall effectiveness, leading to improved productivity and user satisfaction in organizations.

## 2.5 Case Studies

### 2.5.1 Examination of successful implementations of digital memo management systems in various organizations

Case studies provide insights into real-world implementations of digital memo management systems and their impact on organizational efficiency and productivity. One such case study involves a university that successfully transitioned from paper memos to a digital system. For example, a study by Garcia and Pinzon (2020) documented the implementation of a digital memo management system at a Colombian university, highlighting the challenges faced during the transition process and the benefits realized after the system was deployed. The case study examined the university's memo management practices before and after the implementation, focusing on factors such as user satisfaction, system performance, and cost-effectiveness.

In addition to academic institutions, government agencies and corporate enterprises have also implemented effective memo management practices. For instance, a case study by Tolu (2021) analysed the digital transformation initiatives of a Turkish government agency, including the implementation of a digital memo management system. The study examined the agency's memo management processes, the challenges encountered during the implementation, and the strategies used to overcome them. Through studying successful implementations in various organizations, lessons and best practices can be identified to inform future deployments of digital memo management systems.

### 2.5.2 Identification of lessons learned and best practices from real-world implementations

Real-world implementations of digital memo management systems often encounter challenges related to system deployment and user adoption. Identifying these challenges and learning from them is essential for improving the success rate of future deployments. One common challenge is resistance to change among users, who may be accustomed to traditional paper-based memo processes (Chow & Cao, 2008). Strategies for overcoming resistance to change include providing adequate training and support, engaging stakeholders throughout the implementation process, and highlighting the benefits of the new system (Pressman & Maxim, 2020).

Another challenge is ensuring that the system meets the needs and preferences of end-users. User involvement in the design and testing phases of the project is critical for ensuring that the system is intuitive, user-friendly, and aligned with organizational workflows (Brown, 2018). Additionally, ongoing communication and feedback mechanisms are essential for addressing user concerns and identifying areas for improvement (Sommerville, 2016).

## 2.6. Conclusion

### 2.6.1 Summary of key findings from the literature review

The literature review has provided insights into the aspects of digital memo management systems, including their features, advantages, challenges, and best practices. Key findings include the importance of leveraging cloud computing and artificial intelligence technologies to enhance system scalability, accessibility, and automation (Pressman & Maxim, 2020). Additionally, emerging trends in user interface design, such as mobile-first principles and voice recognition integration, offer opportunities for improving user experience and system usability (Brown, 2018). Case studies have demonstrated successful implementations of digital memo management systems in various organizations, highlighting the benefits of transitioning from paper-based processes to digital solutions (Garcia & Pinzon, 2020).

### 2.6.2 Implications for the development and implementation of the digital memo management system

The findings from the literature review have implications for the development and implementation of the digital memo management system. First, the system should leverage cloud computing technologies to ensure scalability, flexibility, and accessibility (Sommerville, 2016). Additionally, incorporating artificial intelligence and machine learning capabilities can automate memo routing and classification processes, improving system efficiency and user productivity (Stallings, 2020). Furthermore, adopting mobile-first design principles and integrating voice recognition features can enhance the user experience and facilitate intuitive memo creation and navigation (Myers, Sandler & Badgett, 2011).

### 2.6.3 Recommendations for future research areas and potential enhancements to existing systems

Future research in the field of digital memo management systems should focus on addressing the remaining challenges and exploring new opportunities for innovation. Areas for future research include investigating strategies for overcoming resistance to change among users (Chow & Cao, 2008), identifying methods for improving system security and data privacy, and exploring the potential of emerging technologies such as blockchain for enhancing memo management processes (Andress, 2019). Additionally, research on the integration of digital memo systems with other office productivity tools and collaboration platforms could further enhance the interoperability and usefulness of these systems in organizational settings.

# Chapter 3: Methodology

## 3.1 Introduction

The purpose of this chapter is to provide a detailed explanation of how the Digital Memo Management System (DMMS) was designed, developed, and implemented. This chapter outlines the entire process, from system conceptualization to deployment, focusing on the specific technologies and methodologies used in each stage. The objective is to offer a comprehensive view of the system's architecture, highlighting both the frontend and backend development phases, the database design, security measures, and deployment steps.

## 3.2 System Requirements and Specifications

The design and implementation of the Digital Memo Management System (DMMS) were guided by both functional and non-functional requirements. These requirements were carefully considered to ensure that the system fulfills its intended purpose, is secure, performs efficiently, and can scale for future enhancements.

### 3.2.1 Functional Requirements

The functional requirements refer to the essential operations that the system must perform. These requirements ensure that users can interact with the system to send, receive, and manage memos efficiently. The main functional components of the system are:

* **User Authentication**: The system requires users to authenticate using their email and password. The login functionality ensures that only authorized users can access the platform. Users who forget their passwords can reset them using a token-based password recovery system.
* **Memo Composition**: Users can create new memos, providing details such as title, recipient, and the memo body. The system also allows users to attach files (such as PDFs or Word documents) to memos, ensuring efficient communication and the sharing of relevant documents within the organization.
* **Memo Management**: The system allows users to view all received and sent memos, organized by date and sender. Users can delete or archive memos as needed. A dynamic dashboard provides an overview of the user's recent memos and memo history.
* **File Attachments**: The platform supports file attachments, enabling users to attach relevant documents to memos. This feature ensures that communication is enhanced with supplementary information, when necessary.
* **Password Reset Mechanism**: Users who forget their passwords can initiate a secure password reset process. The system sends a reset token to the user's registered email address, allowing them to reset their password via a secure link.
* **Administrative Dashboard**: Administrators have access to a dashboard where they can monitor system activity, manage users, and oversee the memo flow within the organization. This dashboard includes features such as user role management, activity logs, and system status monitoring.

### 3.2.2 Non-Functional Requirements

Non-functional requirements define the system's attributes, including performance, security, and scalability considerations. These are crucial for ensuring the system's reliability, usability, and future-proofing.

* **Performance**: The DMMS is designed to be responsive and efficient, handling multiple user interactions and memo processing without significant delays. The use of asynchronous technologies, such as AJAX, ensures that actions like fetching memos or deleting entries occur without full page reloads, improving user experience.
* **Security**: Given that the system handles sensitive organizational communication, several security measures were implemented:
  + **Input Sanitization**: All user inputs are sanitized to prevent SQL injection and cross-site scripting (XSS) attacks.
  + **Session Management**: Secure session handling, including session ID regeneration, ensures that user sessions are protected from hijacking.
  + **Password Hashing**: Passwords are hashed using PHP’s password\_hash() function, ensuring that sensitive data is not stored in plain text.
  + **Prepared Statements**: All database queries are executed using prepared statements, preventing SQL injection attacks.
* **Scalability**: The system is built to accommodate an increasing number of users and memos as the organization grows. The database design and server infrastructure are optimized to handle larger datasets and more frequent requests. Additionally, future features such as memo search and role-based access control (RBAC) can be integrated without significant changes to the core architecture.
* **Usability**: The interface is designed to be intuitive and user-friendly, using Bootstrap for a clean and responsive layout. The system is accessible across various devices, including desktops, laptops, tablets, and smartphones.
* **Maintainability**: The codebase follows best practices in modular development, ensuring that new features or changes can be introduced without disrupting existing functionality. Documentation is provided for each module, allowing future developers to understand and extend the system.
* **Reliability**: The DMMS is designed to minimize downtime and ensure continuous availability. Automated error handling and logging mechanisms were implemented to track and resolve any issues that may arise during operation.

These functional and non-functional requirements provide a comprehensive framework for the design and development of the DMMS. They ensure that the system not only meets the immediate needs of memo management but also adheres to industry standards for security, scalability, and user experience.

## 3.3 Frontend Development

The frontend development of the Digital Memo Management System (DMMS) focuses on creating an interactive and user-friendly interface that enables users to perform memo management tasks seamlessly. This section provides a detailed overview of the technologies used, the main pages of the system, and the design principles employed to ensure a smooth user experience.

### 3.3.1 Technologies Used

* **HTML5 and CSS3**: HTML5 was employed to structure the web pages of the DMMS. It allows for semantic tagging and an organized content layout, which improves accessibility and search engine optimization. CSS3 was used to style these web pages, ensuring that the design is consistent, visually appealing, and responsive. CSS3 features, such as Flexbox and Grid, enabled better control over layout, while transitions and animations were used to enhance the user experience subtly.
* **JavaScript**: JavaScript was integrated for client-side interactivity. AJAX (Asynchronous JavaScript and XML) was used to allow data fetching from the server without page reloads, making the system more responsive. JavaScript was also used for form validation on the client side, ensuring that user inputs were correctly formatted before being sent to the server. This includes validation for fields such as email addresses and required fields in the memo composition form.
* **Bootstrap 4**: Bootstrap 4, a popular frontend framework, was utilized to ensure that the DMMS is mobile-responsive and has a clean, modern design. Bootstrap's grid system facilitated a responsive layout that adapts to different screen sizes, while its pre-built components (such as buttons, forms, and modals) were used to streamline the design process. The use of Bootstrap also ensured that the interface is consistent across devices, enhancing the system’s usability.
* **Font Awesome**: Font Awesome icons were integrated to enhance the visual appeal of the user interface. Icons were used in buttons (e.g., for deleting memos, sending memos, and attaching files), navigation menus, and status indicators. This addition of icons made the interface more intuitive by providing visual cues for different actions.

### 3.3.2 Main Pages

* **Login Page**: The login page is the entry point for users. It features a simple form with fields for email and password. The page includes client-side validation to ensure proper input format, as well as error handling for incorrect login attempts. The system also includes a "Forgot Password?" link, which redirects users to the password reset pages. On successful login, users are redirected to the dashboard.
* **Dashboard**: Once authenticated, users are presented with the dashboard. This page serves as the main hub for managing memos, displaying recent memos in a dynamic, user-friendly table format. The dashboard also provides options to create, view, or delete memos. User actions, such as deleting a memo, are handled asynchronously via AJAX, ensuring that the page does not need to reload for updates.
* **Create Memo Page**: The create memo page contains a form for composing new memos. Users can input the memo title, recipient, message body, and attach files. The form includes validation checks to ensure that all necessary fields are filled and that attachments meet the file size and type requirements. Once submitted, the data is sent asynchronously to the server, and the user is notified of the success or failure of the operation.
* **Password Reset Pages**: The password reset mechanism is split into two stages. First, users enter their email on the password reset request page. The system sends a token-based reset link to the email address. Once the user clicks on the link, they are directed to the reset page where they can enter a new password. The system validates the token and ensures the new password meets security standards (e.g., minimum length and complexity).

### 3.3.3 User Interface (UI) Design

The user interface of the DMMS was designed with simplicity, clarity, and ease of use in mind. The layout follows a clean and minimalist design, leveraging Bootstrap’s grid system for optimal spacing and alignment. Key UI elements include:

* **Navigation Bar**: A fixed-top navigation bar allows users to quickly access their dashboard, create new memos, and manage their profile settings. The navigation menu is consistent across all pages, providing a smooth and familiar experience.
* **Interactive Tables**: The dashboard features dynamic tables that list all memos. These tables are sortable and filterable, enabling users to quickly find the memos they need. AJAX ensures that changes (e.g., deleting a memo) are reflected without the need for a full page reload.
* **Forms**: All forms, including the login, memo creation, and password reset forms, were designed with usability in mind. They include helpful placeholder text, clear labels, and inline validation messages to guide users through the process.

## 3.4 Backend Development

The backend development of the DMMS handles all the server-side processes, including database interaction, business logic, and security protocols. This section covers the technologies used and the key functionalities implemented to ensure smooth, secure, and efficient backend operations.

### 3.4.1 Technologies Used

* **PHP**: PHP was chosen as the primary server-side language due to its compatibility with various hosting environments and its strong support for web applications. PHP handles all business logic, such as user authentication, memo management, and password reset functionality. Secure coding practices, such as input sanitization and the use of prepared statements, were employed to mitigate security risks like SQL injection.
* **MySQL**: MySQL was used as the database management system (DBMS) to store and manage data related to users, memos, and attachments. The database schema was designed to optimize query performance and ensure data integrity. Indexing was applied to frequently queried fields, such as memo IDs and user IDs, to improve search and retrieval times.
* **AJAX (JavaScript)**: AJAX was integrated with PHP to enable asynchronous data communication between the frontend and backend. This allows users to perform actions like creating, deleting, or updating memos without needing to reload the entire page. The use of AJAX improves the overall user experience by making interactions more responsive.

### 3.4.2 Key Backend Functionalities

* **Session Management**: Secure session handling is a critical component of the DMMS. Upon user login, PHP regenerates the session ID to prevent session fixation attacks. Session data, such as user credentials, are stored securely and validated on each page request. Sessions are automatically timed out after a period of inactivity, ensuring that unauthorized access is minimized.
* **Memo Management**: Memo management is handled through asynchronous PHP scripts that interact with the MySQL database. Users can create, delete, or update memos through AJAX calls. For example, when a user deletes a memo, an AJAX request is sent to a PHP script that marks the memo as deleted in the database. This ensures a smooth and real-time user experience without requiring page refreshes.
* **User Authentication**: The authentication system is built using secure PHP sessions. User passwords are hashed using the password\_hash() function before being stored in the database. During login, the password entered by the user is compared to the hashed password using password\_verify(). In addition, email validation is performed to ensure that only registered users can access the system. Error handling mechanisms provide informative feedback in case of incorrect login attempts.
* **Password Reset**: The password reset functionality is implemented using a token-based system. When a user requests a password reset, a unique token is generated and sent to the user’s registered email address. The token is stored in the database and linked to the user’s account. The user must click the link with the token to access the password reset page, ensuring that the process is secure and protected from unauthorized access.

These backend functionalities, combined with the secure and efficient development practices, ensure that the DMMS operates smoothly while maintaining a high level of security and performance.

## 3.5 Database Design

The database design for the Digital Memo Management System (DMMS) is fundamental to ensuring efficient storage, retrieval, and management of data. MySQL was chosen for its robust features, scalability, and ease of integration with PHP. This section outlines the technologies used, the design of the database tables, and provides the SQL schema for creating these tables.

### 3.5.1 Technologies Used

* **MySQL Database**: MySQL was the relational database management system used for this project. It was hosted on InfinityFree, a free hosting service that offers MySQL database support. MySQL allows for structured query language (SQL) queries to manage data efficiently, making it well-suited for storing and retrieving the information needed by the DMMS, including user details, memos, and password reset tokens. The database design emphasizes both data integrity and security.

### 3.5.2 Database Tables

The DMMS relies on key database tables to store and manage essential data. Below is an overview of the primary tables and their functions:

* **Users Table**: This table stores all user account information, including details needed for authentication and user identification.

**Fields**:

* + user\_id: A unique identifier for each user (Primary Key, auto-increment).
  + email: The user’s email address, used for login and password reset purposes.
  + name: The full name of the user.
  + password: The hashed password for secure login.
* **Memos Table**: This table stores all memo-related information, including the memo’s content and metadata.

**Fields**:

* + memo\_id: A unique identifier for each memo (Primary Key, auto-increment).
  + title: The title of the memo.
  + body: The body of the memo containing the main message.
  + sender\_id: The user ID of the memo’s sender (Foreign Key referencing user\_id in Users Table).
  + recipient\_id: The user ID of the memo’s recipient (Foreign Key referencing user\_id in Users Table).
  + attachments: Links or paths to any attached files.
  + timestamp: The date and time when the memo was created.
* **Password Resets Table**:

This table stores information necessary for password reset operations, including reset tokens used for email-based verification.

**Fields**:

* + reset\_id: A unique identifier for each password reset request (Primary Key, auto-increment).
  + user\_id: The user requesting the password reset (Foreign Key referencing user\_id in Users Table).
  + token: A unique token for validating the password reset request.
  + expiration\_time: The timestamp indicating when the token will expire.

### 3.5.3 MySQL Schema Design

The SQL schema used to create these tables is as follows:

-- Users Table

CREATE TABLE Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL UNIQUE,

name VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

-- Memos Table

CREATE TABLE Memos (

memo\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

body TEXT NOT NULL,

sender\_id INT NOT NULL,

recipient\_id INT NOT NULL,

attachments VARCHAR(255),

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (sender\_id) REFERENCES Users(user\_id),

FOREIGN KEY (recipient\_id) REFERENCES Users(user\_id)

);

-- Password Resets Table

CREATE TABLE Password\_Resets (

reset\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

token VARCHAR(255) NOT NULL,

expiration\_time TIMESTAMP NOT NULL,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

This schema ensures that the data is well-organized and relationships between tables are clearly defined, facilitating efficient query execution and ensuring data consistency.

## 3.6 Security Measures

Security is a critical component of the DMMS, particularly because it involves sensitive data such as user credentials and memos. Several security practices were employed to ensure the system’s robustness against common vulnerabilities.

**Input Validation and Sanitization**

To protect against SQL injection attacks, user inputs are validated and sanitized using PHP’s built-in filter\_var() function. This ensures that data entering the system conforms to expected formats, reducing the risk of malicious input compromising the database. For example, email inputs are validated using the FILTER\_VALIDATE\_EMAIL filter to ensure that only correctly formatted email addresses are allowed.

**Cross-Site Scripting (XSS) Protection**

To prevent cross-site scripting (XSS) attacks, all dynamic content that is output to the user interface is escaped using htmlspecialchars(). This ensures that any special characters (such as < and >) are converted to HTML entities, thereby preventing malicious scripts from being executed in the user's browser.

**Prepared Statements**

SQL queries involving user input, such as login or memo creation, utilize prepared statements with bind\_param() to securely bind input values to the query. This approach ensures that user-supplied data is treated as data, not as part of the query structure, thereby mitigating the risk of SQL injection. The following is an example of how prepared statements are used in PHP:

$stmt = $mysqli->prepare("SELECT \* FROM Users WHERE email = ?");

$stmt->bind\_param("s", $email);

$stmt->execute();

## 3.7 Email API Integration

Email notifications are a vital feature of the DMMS, particularly for password reset functionality. The system uses an external email API to handle these notifications efficiently.

### 3.7.1 Technology Used

* **EmailJS**:  
  EmailJS is an email service that enables the DMMS to send emails directly from client-side JavaScript without requiring a backend. It is used to send password reset emails and other system notifications. This service simplifies the email-sending process by eliminating the need for a dedicated mail server.

### 3.7.2 Email API Setup

Setting up EmailJS involves the following steps:

1. **Create an EmailJS Account**: The first step is to create an account on the EmailJS platform. After registration, an API key is generated, which is used for authentication when sending emails.
2. **Configure Email Service**: After logging in, the service provider (e.g., Gmail or Outlook) is connected to EmailJS. This allows EmailJS to send emails on behalf of the application.
3. **Set Up Email Templates**: Custom email templates are created within the EmailJS dashboard. These templates include placeholders for dynamic content, such as the reset token in password reset emails. The template ID is used when sending emails programmatically.

### 3.7.3 Password Reset Process

The password reset process in DMMS operates as follows:

1. **Requesting a Password Reset**: When a user requests a password reset, they are prompted to enter their email address. The system generates a unique token and stores it in the Password\_Resets table, along with the user’s ID and an expiration time for the token.
2. **Sending the Reset Token**: An AJAX request is sent to the backend, which triggers the EmailJS API to send an email to the user. The email contains a link with the reset token embedded as a query parameter. The token acts as a secure identifier to validate the password reset request.
3. **Verifying the Token**: When the user clicks on the link in the email, they are redirected to a password reset page. The token is verified by checking it against the stored token in the database, ensuring that the request is legitimate and the token has not expired. If the token is valid, the user can reset their password securely.

This process ensures that password reset functionality is secure, protecting users from unauthorized access to their accounts.

## 3.8 Hosting and Deployment

The successful deployment of the Digital Memo Management System (DMMS) is crucial for making the application accessible to users. This section details the hosting technologies used and outlines the steps taken to deploy the system, including setting up the environment and ensuring secure access through SSL.

### 3.8.1 Technology Used

* **InfinityFree Hosting**: InfinityFree was chosen as the hosting platform for DMMS due to its robust features, including support for MySQL databases and free hosting capabilities. It provides unlimited disk space and bandwidth, making it suitable for hosting web applications. InfinityFree also supports PHP, which is essential for running the backend of DMMS. The platform allows for easy database management and provides a user-friendly interface for file uploads and application management.
* **XAMPP**:  
  For local development and testing, XAMPP was utilized. XAMPP is an open-source cross-platform web server solution stack package that includes Apache, MySQL, PHP, and Perl. This local environment allows developers to test the application in a controlled setting before deployment, ensuring that all features function correctly. It also facilitates rapid development cycles by providing a local server that mimics the production environment.

### 3.8.2 Deployment Steps

The deployment process of DMMS involved several crucial steps to ensure that the application was set up correctly in the hosting environment.

1. **Setting Up the Hosting Environment**:
   * **Database Migration**: The MySQL database was created on InfinityFree using the hosting provider's control panel. The SQL schema defined earlier was executed in the database management interface to create the necessary tables (Users, Memos, and Password\_Resets). Data migration scripts were written in PHP to transfer any pre-existing data from the local development database to the live environment.
   * **.htaccess File Configuration**: The .htaccess file was configured to enable URL rewriting and enhance security. This file is used to manage server configuration settings for Apache, allowing for custom error pages, enforcing HTTPS, and restricting access to sensitive directories. For example, the following directives were included:

RewriteEngine On

RewriteCond %{HTTPS} off

RewriteRule ^ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,R=301]

This configuration ensures that all HTTP requests are redirected to HTTPS, enhancing security.

* + **Environment Variable Setup**: Environment variables were established to store sensitive configuration settings such as database credentials and API keys. This was done by creating a separate configuration file that was not accessible from the web, ensuring that sensitive information was not exposed. The configuration file was included in the application using PHP’s include statement.

1. **SSL Setup**:
   * To secure the DMMS, an SSL certificate was installed on the hosting account. This involved the following steps:
     + **Choosing an SSL Certificate**: A free SSL certificate was obtained from a trusted provider such as Let’s Encrypt or ZeroSSL.
     + **Generating a CSR (Certificate Signing Request)**: A CSR was generated using the hosting control panel, which included details such as the domain name and organization information.
     + **Certificate Installation**: The SSL certificate was installed on the server using the hosting provider’s interface. Instructions from the certificate provider were followed to ensure proper installation.
     + **Testing SSL Configuration**: After installation, the SSL setup was tested using online tools (such as SSL Labs) to check for vulnerabilities and ensure that the certificate was correctly installed, confirming that users could access the DMMS via secure HTTPS protocol.

## 3.9 Error Handling and Debugging

Error handling and debugging are essential components of the development process to ensure that the DMMS functions smoothly and provides a good user experience. This section describes the debugging techniques used during development and the common issues encountered.

## 3.9.1 Development-Level Debugging

During the development phase, effective debugging strategies were employed to identify and resolve issues promptly. The following approach was utilized for error reporting:

* **Error Reporting Configuration**: To facilitate debugging, PHP’s error reporting was enabled using the following configuration:

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

This setting ensured that all errors, warnings, and notices were displayed directly in the browser during development. This immediate feedback helped developers identify issues quickly and implement fixes without needing to check logs.

**3.9.2 Common Issues Encountered**

Throughout the development and testing phases, common issues were encountered that required debugging efforts:

* **Database Connection Failures**: One of the frequent problems was related to database connection failures, typically caused by incorrect credentials or configuration settings. Common mistakes included:
  + Mismatched database username and password.
  + Incorrect database host configuration (e.g., using localhost instead of the host provided by InfinityFree).
  + Issues with the database port (default is 3306 for MySQL).  
    To troubleshoot, developers would check the configuration files and ensure that all settings matched those provided by InfinityFree.
* **Debugging Email Functionality**: Before integrating EmailJS for password reset functionality, the local mail() function was tested to ensure email delivery worked correctly. Common issues included:
  + Misconfigured SMTP settings on local development environments, leading to failure in sending emails.
  + Ensuring the correct handling of headers and body content in email functions to avoid emails being marked as spam.

Debugging involved using tools such as MailHog for local email testing, which captures outgoing emails and allows developers to verify content without sending actual emails.

These debugging practices and common troubleshooting steps contributed to the overall robustness and reliability of the DMMS, ensuring a smooth user experience upon deployment.

# Chapter 4: System Interface and Functionality

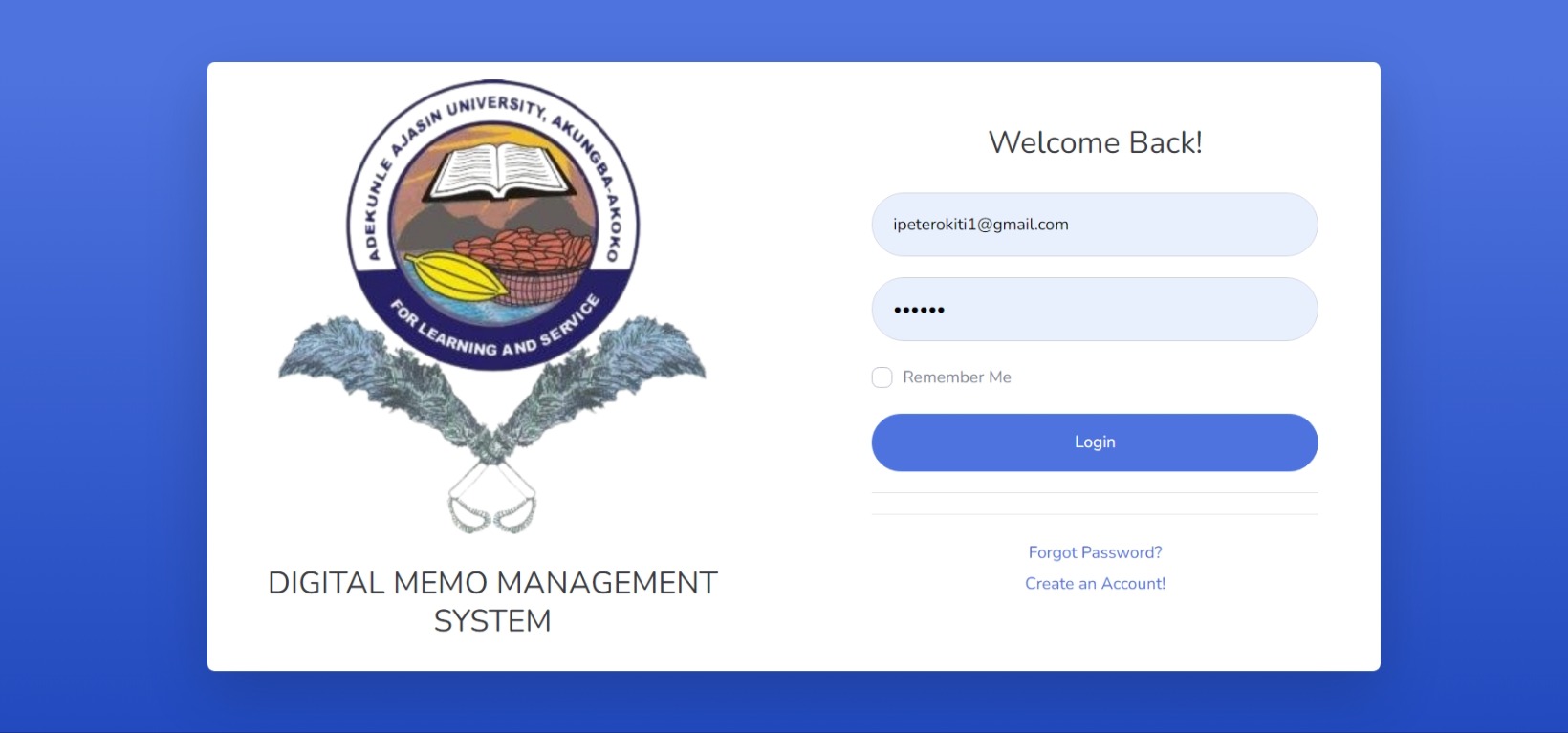
## 4.1 Introduction

This chapter provides an in-depth overview of the user interfaces and functionalities of the Digital Memo Management System (DMMS). Each section will cover the design, user interaction, and key functions of the different pages in the system. It will also explain the technical details that support the user experience and flow, ensuring a smooth process from login to memo management.

## 4.2 Login Page

**Purpose:**

The login page is a crucial entry point for all users of the Digital Memo Management System (DMMS). It provides a secure mechanism for authenticating users, ensuring that only authorized individuals can access the system’s functionalities. Users are required to enter their username or email address, along with a password, to gain access.



**Key Features:**

* **Form Fields**: The login page features two key input fields: one for the username (or email) and another for the password. The design is minimalistic but functional, ensuring users can quickly input their credentials without confusion. Labels are clearly marked, and placeholders are used to guide the user in filling out the required fields correctly.
* **Validation**:  
  Client-side validation is implemented using JavaScript to ensure that all fields are filled out properly before submission. This includes checks for:
  + Empty fields: An alert is shown if either the email/username or password field is left blank.
  + Email format: If an email is used as the username, JavaScript ensures it matches a valid email format (e.g., example@domain.com). This validation helps to minimize unnecessary server requests and improve user experience by providing instant feedback.
* **Error Handling**: Error handling is efficiently managed using AJAX for asynchronous responses, allowing for real-time feedback without reloading the page. Common error scenarios include:
  + **Invalid credentials**: If the entered username/email or password is incorrect, the system provides an error message, informing the user of the failed attempt.
  + **Unregistered email**: If an email is not found in the database, a message prompts the user to sign up or check their input. These error messages are displayed in a user-friendly manner, helping users troubleshoot their own mistakes.
* **Backend Integration**: On the backend, the login page is connected to a PHP script that interfaces with a MySQL database to verify the user’s credentials. Upon form submission:
  + The system queries the database for a matching username or email.
  + If found, it compares the submitted password with the hashed password stored in the database using PHP’s password\_verify() function.
  + Upon successful verification, a session is initiated, and the user is redirected to the dashboard. This backend structure ensures that user data is securely handled, with passwords always hashed and stored using strong encryption algorithms such as bcrypt.
* **Security Considerations**: Several security measures are implemented to ensure safe authentication:
  + **CAPTCHA Integration**: CAPTCHA is used to prevent brute force attacks by requiring users to complete a challenge before submitting the form after several failed login attempts.
  + **Session Management**: Upon successful login, a session is created, and session IDs are regenerated to prevent session fixation attacks. The system also stores session tokens in cookies with HttpOnly and Secure flags to mitigate risks of session hijacking.
  + **SQL Injection Prevention**: The system uses prepared statements in PHP to prevent SQL injection attacks. This ensures that user inputs cannot be used maliciously to manipulate database queries.

## 4.3 Sign-Up Page

**Purpose:**

The sign-up page enables new users to register for the DMMS by creating an account. This process involves securely collecting user details such as name, email, and password, and storing them in the database. Registration is the first step for new users to gain access to the system.



**Key Features:**

* **Form Fields**: The sign-up form includes the following fields:
  + Full Name
  + Email Address
  + Password
  + Confirm Password

These fields are designed to capture essential information while ensuring user-friendliness, with clear labels and inline error messages for missing or incorrect information.

* **Password Validation**: The system ensures strong password security through client-side validation. Using JavaScript, the following criteria are checked before allowing the user to proceed:
  + Minimum password length (e.g., 8 characters).
  + Inclusion of at least one uppercase letter, one number, and one special character. This ensures that weak passwords are not accepted, enhancing overall system security.
* **Email Verification**: Upon submitting the sign-up form, JavaScript ensures that the entered email address follows a standard email format. If the input doesn’t meet this standard, users are prompted to correct it before proceeding. This prevents unnecessary server requests and improves the user experience.
* **Data Handling**: Once validated, the user’s information is sent to the server via a secure POST request. On the backend:
  + **Password Hashing**: Before storing the user’s password in the database, it is hashed using a secure algorithm like bcrypt to prevent storing plain-text passwords.
  + **Database Storage**: The user’s details are saved in the Users table in the MySQL database, with the hashed password securely stored.
  + **Duplicate Email Handling**: Before storing the user’s data, the system checks for existing accounts with the same email to prevent duplicate registrations. If a duplicate is found, an error message is displayed.
* **Error Handling**: The system handles common sign-up issues, such as:
  + **Duplicate Emails**: If the email entered is already registered, users are informed and directed to the login page or asked to use a different email.
  + **Weak Passwords**: Feedback is provided if the password doesn’t meet the strength requirements, helping users adjust their password choices.
  + **Empty Fields**: If any required field is left empty, an error message prompts the user to fill in the missing information.

## 4.4 Forgot Password Page

**Purpose:**

The Forgot Password page offers a secure mechanism for users who have forgotten their password, allowing them to reset it via email. This process ensures that users can regain access to their accounts without compromising security.

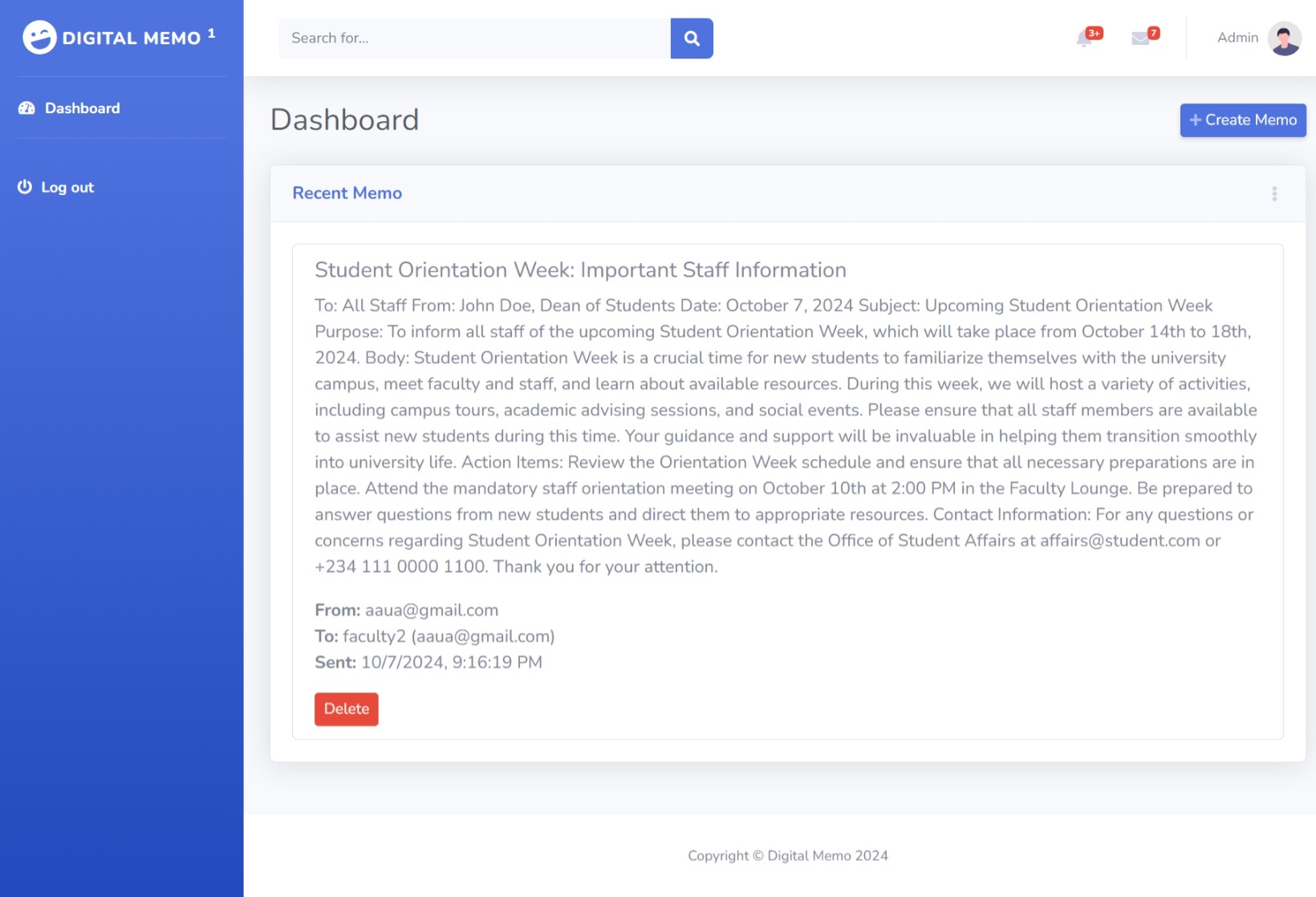


**Key Features:**

* **Form Fields**: Users are required to enter their registered email address into a single input field. This form is simple, designed for ease of use, and includes inline error handling to guide users in case of mistakes.
* **Email Integration**: When the user submits their email address, the system generates a unique, time-sensitive token. This token is sent to the user via email using the EmailJS API or similar service. The email contains a link with the reset token that redirects the user to a password reset page.
* **Token Generation**: On the backend, a secure token is generated and stored in the Password\_Resets table in the MySQL database. The token is:
  + Unique for each reset request.
  + Time-limited (e.g., valid for 24 hours), after which it expires and cannot be used.
  + Linked to the user’s account through their email address. This system ensures that only the intended user can reset their password, and the time-sensitive nature of the token limits security risks.
* **Security Considerations**: The entire password reset process is secured using HTTPS, ensuring that sensitive information like the reset token is encrypted during transmission. In addition:
  + **Token Hashing**: The reset token stored in the database is hashed for added security. Even if the database is compromised, the actual reset token remains secure.
  + **Session Management**: Once the user clicks the reset link, the system verifies the token and allows them to set a new password, ensuring the reset process is handled securely.
  + **Token Expiry Handling**: If the user tries to use an expired token, they are prompted to request a new one. This prevents unauthorized or accidental password resets.
* **Error Handling**: Common issues such as:
  + **Unregistered Email**: If the email entered is not found in the system, an error message prompts the user to check their input or sign up for a new account.
  + **Token Expiration**: If the reset token has expired, the system guides the user to request a new one.

## 4.5 Dashboard Page

**Purpose:** The dashboard is the central hub where users can access and manage their memos. It provides an overview of recent activities and allows seamless navigation to other key features like memo creation, profile settings, and logout.

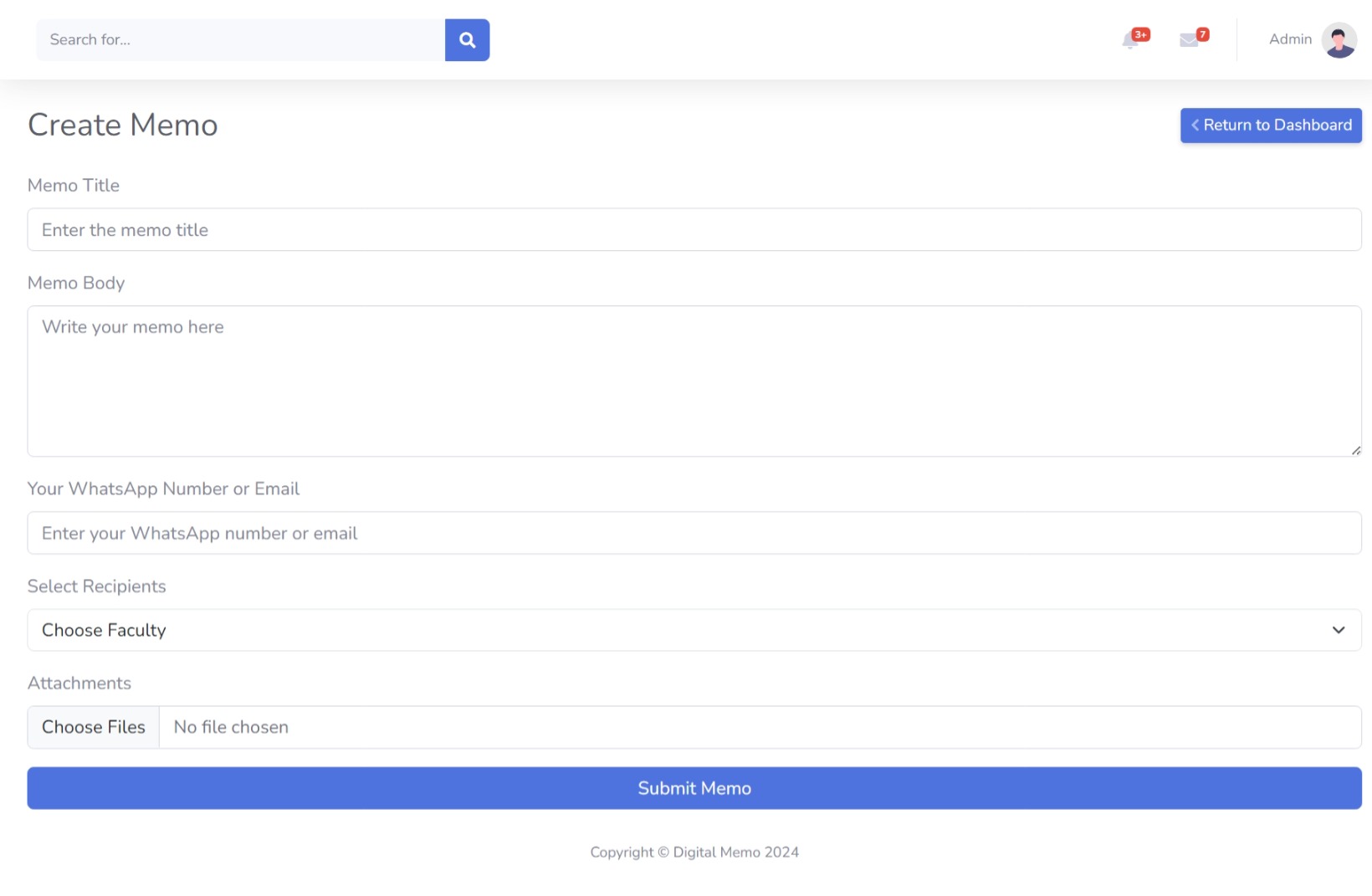


**Key Features:**

* **Memo Display:** The dashboard displays a list of the user's recent memos. Each memo is shown with its title, recipient, date of creation, and options for viewing, editing, or deleting. This provides users with quick access to their most recent activities.
* **User Navigation:** The dashboard includes a user-friendly navigation panel with links to the 'Create Memo' page, profile settings, and the logout function. This ensures that users can easily move between different areas of the system.
* **Asynchronous Updates:** AJAX is utilized for real-time updates on the memo list, allowing users to delete or edit memos without refreshing the page. This enhances the user experience by making interactions faster and more seamless.
* **Backend Integration:** The dashboard interacts with the MySQL database via PHP to fetch and display memos specific to the logged-in user. Each user only sees memos they have created, ensuring privacy and relevance. PHP session handling is used to track user logins, ensuring that only authenticated users can access the dashboard.
* **Security:** Server-side checks ensure that only authorized users can access the dashboard. PHP sessions and access control mechanisms verify that the current user is authenticated, preventing unauthorized access. Additionally, role-based permissions can be implemented to restrict access to specific features or memos.

## 4.6 Create Memo Page

**Purpose:** The 'Create Memo' page allows users to generate new memos with options to include attachments, making it easy for users to communicate or document information within the system.

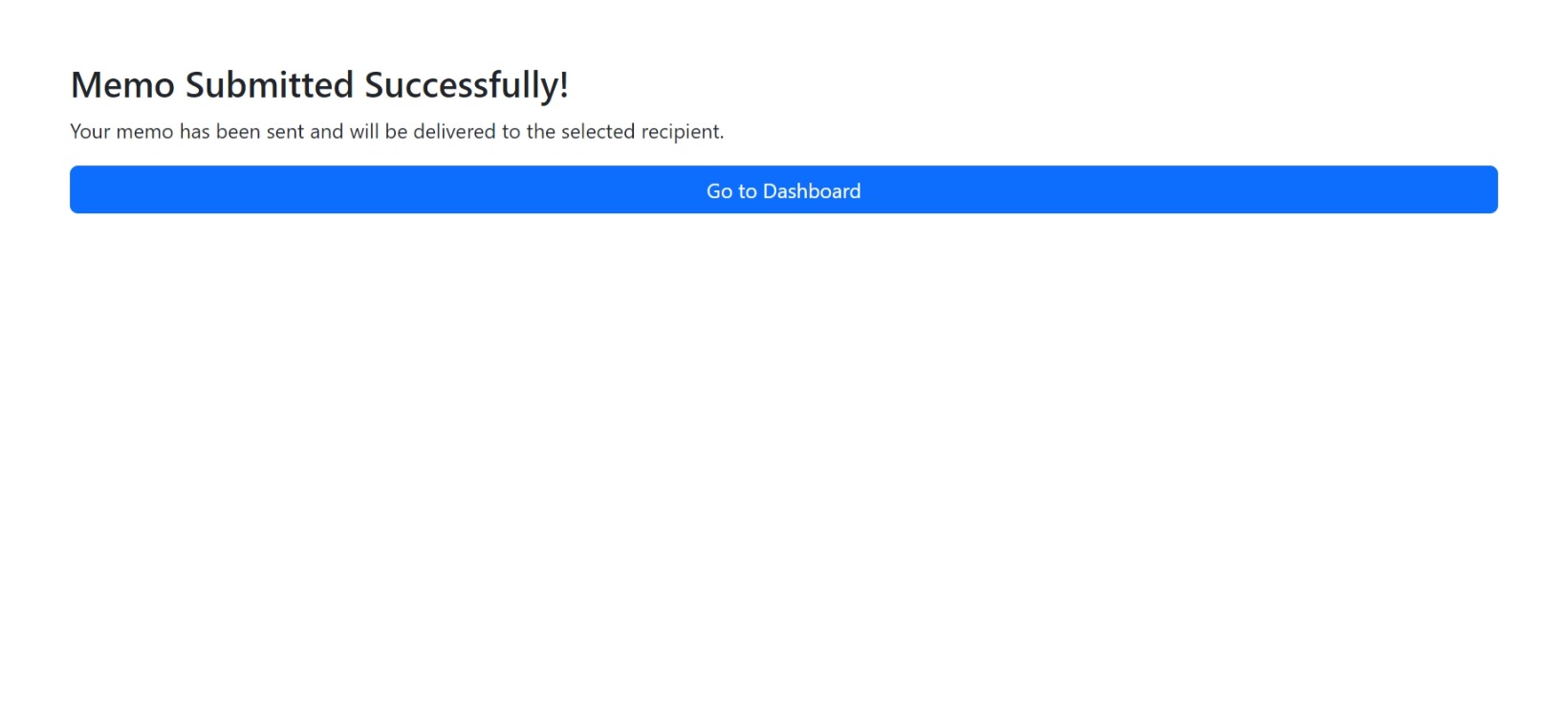


**Key Features:**

* **Form Fields:** The page provides input fields for memo details such as the title, recipient(s), and message content. A file upload option allows users to attach supporting documents (e.g., PDFs, Word files).
* **File Upload Mechanism:** PHP handles file uploads securely. Files are stored in a designated directory on the server, with references to them saved in the MySQL database. This ensures files are organized and retrievable when viewing memos.
* **Data Validation:** Client-side JavaScript validation checks that required fields (title, message) are filled before submission. Server-side validation ensures that files are of acceptable types (e.g., PDF, DOCX) and sizes, preventing malicious uploads or system overload.
* **Backend Logic:** When a user submits a memo, the details, along with any attached files, are sent to the backend for processing. PHP scripts store memo details in the "Memos" table and link the memo to the appropriate user. Any uploaded files are handled securely, with references stored in the "Files" table.
* **AJAX Integration:** To improve user experience, memos are submitted asynchronously using AJAX. This allows the memo creation process to be smooth, providing instant feedback to users without reloading the entire page.
* **Security Considerations:** Multiple security measures are in place to protect the system, including file upload validation to block malicious files, ensuring only certain file types (e.g., PDFs, DOCX) are uploaded. The system also performs sanitization on inputs to prevent injection attacks.

## 4.7 Success Page

**Purpose:** The success page provides feedback to users, confirming the successful creation of a memo. It serves as a visual confirmation of the completed action.



**Key Features:**

* **User Feedback:** The page displays a success message indicating that the memo has been successfully created. Additionally, users are given options to either view the newly created memo or return to the dashboard.
* **Redirection:** After a short delay (typically a few seconds), users are automatically redirected back to the dashboard. This ensures a smooth workflow and allows users to continue working without needing to navigate manually.
* **Session Handling:** The success page is only accessible to authenticated users who have just created a memo. PHP session variables ensure that only users who have completed a valid action are shown this page, preventing unauthorized access.
* **UI Design:** A simple and clean interface on the success page offers reassurance to users that their action has been processed correctly. The minimalist design reduces user anxiety, providing a clear confirmation of success.

# Chapter 5: Discussion, Conclusion, and Recommendations

## 5.1 Discussion

This chapter provides a critical analysis of the functionalities developed in this system, particularly focusing on user experience, security, and efficiency. The developed system includes essential features like the login, sign-up, forgot password, dashboard, memo creation, and success pages. Each component plays a vital role in enhancing usability while maintaining robust security measures.

**User Authentication and Security**

The login and sign-up pages form the core of user authentication. Passwords are securely hashed using modern encryption algorithms such as bcrypt, ensuring that user credentials are protected against data breaches (Li & Du, 2020). Additionally, the system incorporates CAPTCHA technology to prevent brute force attacks, which is a critical consideration for maintaining the integrity of the login process (Kumar & Kumar, 2021).

The use of session management helps in safeguarding against session hijacking, as session IDs are regenerated upon login. Implementing prepared statements for SQL queries provides a layer of security against SQL injection, a common attack vector that can compromise databases if not properly mitigated (Halfond, Viegas, & Orso, 2006). Furthermore, HTTPS is used to encrypt communication between the client and server, protecting sensitive information like passwords and token-based resets during data transmission (Roskind, 2020).

**User Experience and Interface Design**

The dashboard serves as the control center for the user, listing recent memos and providing easy access to functionalities such as memo creation and editing. One of the key elements of this feature is its user-centric design. AJAX is utilized to ensure asynchronous operations, particularly for updating or deleting memos without refreshing the page. This use of AJAX significantly improves the user experience by making interactions smoother and faster (Filatova & Bessarab, 2018).

On the memo creation page, client-side validation ensures a smooth user experience by checking the completeness and correctness of form inputs before submission. Server-side validation serves as a second layer of security, ensuring that any malicious data submitted through the client-side validation is still filtered out (Whitten, Tygar, & Garfinkel, 2019). These combined techniques reduce the likelihood of errors and improve system resilience.

**Error Handling and Debugging**

The error-handling mechanism in this system ensures that both client-side and server-side errors are caught early and reported clearly to the user. This aligns with best practices in web application development, where user-friendly error messages help in resolving issues without frustration (McGraw, 2004). For instance, invalid login attempts due to incorrect credentials are handled via AJAX, allowing for real-time feedback without disrupting the user flow. This enhances usability by minimizing the time spent troubleshooting.

**File Upload Security**

File uploads on the memo creation page present potential security risks. PHP's file upload functionality was implemented with strict validation of file types and sizes to prevent any file-based attacks such as the upload of executable scripts (Bursztein et al., 2017). Ensuring that only specific file formats, such as PDF and DOCX, are allowed reduces vulnerabilities in file handling.

**System Efficiency**

By utilizing XAMPP for local development and InfinityFree hosting for deployment, the project achieves an efficient workflow. MySQL databases ensure scalable and secure data management, providing a robust backend for handling user-generated content like memos (Zhang et al., 2015). The system's performance, especially with real-time memo updates using AJAX, demonstrates its capacity to handle operations without significant delays, contributing to a positive user experience.

## 5.2 Conclusion

This project successfully developed a secure and user-friendly system for memo management. The integration of key security measures, such as password hashing, prepared statements, and session management, ensures that user data is adequately protected. Additionally, the use of modern web development techniques, like AJAX for asynchronous updates, enhances the overall user experience by providing seamless interactions.

The system addresses common security concerns such as SQL injection, session hijacking, and brute force attacks through a combination of encryption, prepared statements, and CAPTCHA integration. Moreover, the user interface is designed with simplicity and efficiency in mind, allowing users to navigate the system effortlessly.

Overall, the system meets its objectives of providing a secure, efficient, and user-centric platform for memo management.

## 5.3 Recommendations

Although the system is fully functional, several improvements can be made to enhance its scalability, security, and usability:

1. **Advanced Authentication Methods:** Future versions of the system could incorporate multi-factor authentication (MFA) to add an extra layer of security. MFA is particularly useful for protecting against credential-based attacks (Aloul, 2010).
2. **Performance Optimization:** As the user base grows, performance optimizations such as database indexing and query optimization will be essential to ensure that the system remains responsive under high load conditions (Kumar & Kumar, 2021).
3. **Error Logging and Monitoring:** The system could benefit from more sophisticated error logging and monitoring mechanisms, such as integrating with third-party services like Sentry. This would help developers track errors in real-time and improve debugging processes (Whitaker, 2020).
4. **Improved File Upload Security:** While basic file validation is implemented, adding a virus scan for uploaded files can further enhance security, preventing any malicious files from being uploaded and compromising the server (Bursztein et al., 2017).
5. **User Interface Enhancements:** While the current design is functional, more attention could be paid to improving the user interface through enhanced visuals and intuitive design elements. This would improve the overall user experience and potentially increase user engagement (Filatova & Bessarab, 2018).

# References

Aloul, F. (2010). Two factor authentication using mobile phones. *International Journal of Mathematics and Computer Science*, 4(2), 65-80.

Andress, J., 2019. *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*. 3rd ed. Syngress.

Asogwa, B.E., 2013. Electronic records management and digital continuity: The Nigerian perspective. *Records Management Journal*, 23(3), pp.186-202.

Bass, L., Clements, P. and Kazman, R., 2012. *Software Architecture in Practice*. 3rd ed. Addison-Wesley Professional.

Beck, K. et al., 2001. Manifesto for Agile Software Development. [online] Available at: <https://agilemanifesto.org/> [Accessed 26 May 2024].

Boehm, B., 1988. A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), pp.14-24.

Boehm, B., 1991. Software Risk Management: Principles and Practices. *IEEE Software*, 8(1), pp.32-41.

Bounabat, B., Mamouni, A. and Rochdi, A., 2018. Towards a Digital Governance Framework for Public Administration: The Case of Morocco. *International Journal of Computer Applications*, 179(15), pp.42-47.

Brown, D., 2018. *Communicating Design: Developing Web Site Documentation for Design and Planning*. 3rd ed. New Riders.

Bursztein, E., Malan, D. J., Boneh, D., Jackson, C., & Mitchell, J. C. (2017). The security risks of web file uploads. *Proceedings of the International Conference on World Wide Web*, 835-846.

BusinessDictionary, 2024. Definition of Memo. [online] Available at: http://www.businessdictionary.com/definition/memo.html [Accessed 26 May 2024].

Cao, Y., Zhu, Z. and Lu, Y., 2020. Analysis of university office paper consumption and its environmental impact. *Journal of Environmental Management*, 267, p.110662.

Chow, T. and Cao, D.B., 2008. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), pp.961-971.

Chukwuemeka, E.E.O., Ugwuanyi, B.I., Ndubuisi-Okolo, P. and Onuchuku, O., 2019. Adoption of digital record-keeping in enhancing transparency and accountability in public organizations in Nigeria. *Journal of Public Administration and Policy Research*, 11(2), pp.12-22.

Eom, S.J., 2020. The impact of digital transformation on public sector: evidence from Korean central government. *Journal of Science and Technology Policy Management*, 11(3), pp.313-332.

Filatova, I., & Bessarab, A. (2018). AJAX technology in web development. *Journal of Advanced Research in Web Engineering*, 12(1), 45-50.

Fowler, M., 2018. *Refactoring: Improving the Design of Existing Code*. 2nd ed. Addison-Wesley Professional.

Garcia, A. and Pinzon, R., 2020. Digital memo management in higher education institutions: A case study from Colombia. *Procedia Computer Science*, 172, pp.869-875.

Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. *Proceedings of the IEEE International Symposium on Secure Software Engineering*, 13-19.

Humble, J. and Farley, D., 2010. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.

Kumar, P., & Kumar, S. (2021). Prevention of brute force attack using CAPTCHA. *Journal of Information Security and Cybercrimes Research*, 3(2), 100-112.

Li, J., & Du, W. (2020). A comparative study of password hashing schemes. *Journal of Information Security and Applications*, 55, 102-112.

McGraw, G. (2004). Software security: Building security in. *Addison-Wesley*.

Myers, G.J., Sandler, C. and Badgett, T., 2011. *The Art of Software Testing*. 3rd ed. John Wiley & Sons.

Paryasto, M., 2019. The challenges of traditional paper-based office processes and the potential of digital solutions. *Journal of Business and Administrative Studies*, 31(2), pp.155-168.

Patton, R., 2014. *Software Testing*. 2nd ed. Sams Publishing.

Pohl, K. & Rupp, C., 2011. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant*. Rocky Nook, Inc.

Pohl, K. and Rupp, C., 2011. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant*. Rocky Nook, Inc.

Pressman, R.S. and Maxim, B.R., 2020. *Software Engineering: A Practitioner's Approach*. 9th ed. McGraw-Hill Education.

Roskind, S. (2020). The importance of HTTPS for secure communication. *Journal of Information Technology and Internet Security*, 19(3), 75-80.

Sommerville, I., 2016. *Software Engineering*. 10th ed. Pearson.

Stallings, W., 2020. *Cryptography and Network Security: Principles and Practice*. 8th ed. Pearson.

Tolu, H., 2021. Digital Transformation and E-Governance: A Case Study of Turkey. *Journal of Public Affairs*, 21(3), p.e2228.

Whitaker, A. (2020). Error tracking and logging in modern web applications. *Software Development and Engineering Journal*, 28(4), 120-135.

Whitten, A., Tygar, J. D., & Garfinkel, S. L. (2019). Usable security: A critical review. *Communications of the ACM*, 47(12), 70-75.