A Secure Document Storage and Expiry Reminder System

Dr.R.Amutha
Associate Professor, Department of Computer Science,
PSG College of Arts & Science,
Coimbatore-14.
amutharajan@gmail.com

***Abstract*—The increasing volume of personal and professional documents creates practical difficulties in secure storage, retrieval, expiry tracking, and renewal management. This paper presents PingVault, a web-based document storage and expiry reminder system that centralizes important files and reduces dependence on manual tracking. The system provides a digital vault for uploading, organizing, retrieving, and managing documents with or without expiry dates. Its principal functions are encrypted document storage, expiry-date tracking, automated email reminders, permanent document storage, responsive navigation, theme customization, and an analysis dashboard. A Python backend built with Django and SQLite3 is combined with a frontend implemented in HTML5, CSS3, JavaScript, and Bootstrap. The reminder component uses SMTP-based email delivery and Windows Task Scheduler to issue notifications before documents expire, and ApexCharts visualizes document categories, reminders, uploads, and expiry activity. Qualitative testing covered document upload, expiry management, reminder delivery, dashboard behavior, responsive layouts, and cross-browser compatibility. The platform offers a practical foundation for organized document management; future work should evaluate performance, security controls, usability with representative users, and scalability.**

***Index Terms*—document management; expiry reminders; digital vault; encrypted storage; Django; SMTP; web application; data visualization**

I. INTRODUCTION

Important documents such as identity records, licenses, certificates, insurance papers, and professional files are often spread across local folders, physical storage, calendar applications, and informal reminders. This fragmented practice makes documents hard to locate, metadata inconsistent, and renewal deadlines easy to forget. Missed expiry dates can lead to administrative delays, service interruptions, or avoidable penalties.

Electronic document and records management systems were introduced to replace paper and ad hoc folders with structured repositories [1], [2], and national initiatives such as India's Digital Locker show the same shift at citizen scale [3]. Storage and retrieval alone, however, do not tell a user that a passport, license, or insurance policy is about to lapse. PingVault was developed during an internship at Ether Infotech to close this gap by combining file storage with proactive notification, so that users keep documents in a single digital vault and receive reminders before the relevant expiry dates. Files that never expire can be placed in a permanent storage category.

The main contributions of this work are: (i) a centralized, encrypted document vault with per-document expiry metadata; (ii) an automated reminder service built on SMTP email delivery and a scheduled task; (iii) permanent storage and editable expiry dates, which separate document content from changing renewal information; (iv) an analysis dashboard for document and reminder activity; and (v) a qualitative testing study across browsers and device classes.

II. LITERATURE SURVEY

A. Electronic Document and Records Management

Electronic document management systems (EDMS) and electronic records management systems (ERMS) replace paper and scattered folders with repositories that support storage, indexing, retrieval, and access control. Johnston and Bowen [1] reviewed published and unpublished cases to identify the benefits organizations had actually achieved, and found few case studies that described clear, quantitative benefits. More recent systematic reviews concentrate on adoption. Bizi et al. [2] synthesized 16 studies and identified 12 factors that influence ERMS adoption, including relative advantage, compatibility, management support, and government policy. Anas and Salim [4] reviewed EDMS use in support of knowledge management and reported benefits in storage, indexing, searching, and easy access, together with challenges in software, hardware, and human-resource capability. The works surveyed here focus on storage, retrieval, and organizational adoption; none of them centres on tracking the expiry dates of individual documents.

B. Digital Document Repositories

The Digital Locker Technology Specification [3] defines a national mechanism for issuing government documents to Aadhaar holders. It describes a federated arrangement of repositories rather than a single central store, and lets residents share documents through unique document URIs instead of photocopies, which reduces manual verification effort and the scope for fake documents. Kumar and Baghel [5] give an overview of DigiLocker within the Digital India programme, covering its objectives, features, and the everyday use of e-documents. This work shows how a trusted repository can reduce paper handling and verification overhead. Because the specification centers on government-issued documents, PingVault addresses a complementary need: keeping arbitrary personal and professional files with user-controlled expiry information.

C. Security of Stored Documents

Keeping sensitive records in centralized or cloud-hosted repositories places confidentiality, integrity, and availability at the center of design. Huang et al. [6] surveyed cloud storage security around exactly this triad, and Yang et al. [7] reviewed data security and privacy challenges, encryption technologies, and protection methods for cloud storage. For file storage in particular, Maitri and Verma [8] proposed a hybrid cryptography algorithm for secure file storage in cloud computing. The Advanced Encryption Standard [9] specifies a symmetric block cipher that is widely used as the basis for encrypting stored data. These works treat encryption as one element alongside access control and other countermeasures [6],

[7]. Accordingly, this paper presents PingVault's security features as design measures that still require formal evaluation.

D. Automated Notification and Reminder Systems

Automated email notification has been studied most rigorously in clinical settings. In a cluster-randomized trial, Dalal et al. [10] found that automated secure email notification of test results pending at discharge improved physicians' self-reported awareness of those results. A later trial by the same group [11] found that notification did not significantly raise the proportion of actionable results with documented follow-up, although follow-up occurred sooner where emails were received. These findings suggest that reminders improve awareness and timeliness more reliably than final completion, which motivates keeping renewal status visible inside the application as well as sending alerts.

In the domain of expiring documents, a recent university project report [12] describes a web-based document-expiration system with a notification engine that delivers reminders by email, Telegram, and in-app alerts, together with role-based access control and calendar integration. Such work confirms the demand for expiry-aware document tools. PingVault adopts a compact Django–SQLite implementation with SMTP reminders and vault-integrated analytics.

E. Web Technologies Used

PingVault builds on widely documented components. Django provides a web framework and email-sending abstractions [13] that wrap Python's smtplib SMTP client [14]. SQLite offers a self-contained, file-based SQL database, and its documentation also discusses appropriate use [15]. Bootstrap supplies responsive, mobile-first layout components [16], in line with the fluid-grid, flexible-image, and media-query principles of responsive web design introduced by Marcotte [17]. ApexCharts provides interactive charting [18]. For the analysis module, Sarikaya et al. [19] reviewed the dashboard literature and characterized dashboards as a distinct visualization context with their own design goals and levels of interaction, which supports presenting document activity as a dedicated dashboard rather than as raw lists.

F. Research Gap

As summarized in Table I, the literature addresses document storage, government repositories, storage security, and notification largely as separate topics. Few of the works reviewed here combine encrypted storage, per-document expiry metadata, automated email reminders, and an analytics dashboard in one lightweight web application. PingVault is positioned at this intersection.

TABLE I
SUMMARY OF RELATED RESEARCH STRANDS

Research strand	Main focus	Relation to PingVault
EDMS / ERMS [1], [2], [4]	Organizational records, retrieval, adoption factors, knowledge management	Applies centralized storage to personal and professional documents with lightweight deployment
Government digital lockers [3], [5]	Issuer-driven, Aadhaar-linked federated repositories for official documents	Stores user-uploaded files and attaches user-managed expiry metadata
Secure cloud storage [6]–[9]	Confidentiality, integrity, availability; encryption and hybrid cryptography	Adopts encryption and authenticated access as core requirements; formal assessment is future work
Automated notification [10]–[12]	Email alerts that improve awareness; expiry-management web systems	Couples reminders directly to each stored document's expiry date
Web technology stack [13]–[19]	Frameworks, database, responsive interface, dashboards	Integrates Django, SQLite3, Bootstrap, and ApexCharts

III. PROBLEM STATEMENT AND OBJECTIVES

A. Limitations of Conventional Document Handling

Fragmented storage. Documents are commonly kept on personal devices or in physical folders, spread over several directories and named inconsistently, so retrieval becomes harder as the collection grows.

Manual expiry tracking. Expiry dates are recorded in calendars, on sticky notes, or from memory. These methods depend on continuous attention and have no dedicated link between a document and its renewal date.

Absence of automated reminders. Individually created alarms can be deleted, forgotten, or separated from the document they concern, so manual processes give limited assurance that a renewal notice will arrive on time.

Weak categorization and retrieval. Without a central classification scheme, searching for a specific document or for all files tied to a particular expiry period is inefficient.

Security and access concerns. Files stored without encryption or access control can be viewed or modified by anyone with access to the device or storage location, which is a concern for identity, financial, and employment records.

Limited analytics and scalability. Traditional storage offers no dashboards for document categories, uploads, reminder history, or upcoming expiries, and manual tracking scales poorly as documents and users increase.

B. System Objectives

The primary objective of PingVault is to provide a secure, reliable, and user-friendly platform for managing important documents in a centralized digital vault. The specific objectives are:

1. To centralize documents in a structured digital vault.
2. To associate documents with expiry dates and renewal information.
3. To automate email notifications before document expiry.
4. To protect sensitive files and metadata through encryption and access controls.
5. To support permanent storage for documents that do not expire.
6. To provide rapid retrieval and editable expiry information.
7. To visualize document and reminder activity through an analysis dashboard.

8. To deliver a responsive interface across common browsers and device sizes.
9. To provide a modular foundation for future users and analytics requirements.

IV. PROPOSED SYSTEM: PINGVAULT

A. Centralized Digital Vault

PingVault provides a single location for uploading and managing important documents. Each file can carry metadata, including its expiry date where applicable. Centralization improves organization and gives a consistent workflow for uploading, viewing, editing, and retrieving documents.

B. Expiry Reminder Service

The reminder service identifies documents approaching their expiry dates and sends notification emails before the deadline. It is implemented as a Python automation component that uses SMTP email delivery and Windows Task Scheduler, an arrangement consistent with the email facilities provided by Django and Python [13], [14].

C. Secure and Encrypted Storage

PingVault is designed to protect stored documents and sensitive metadata through encryption and authenticated access; document encryption and encrypted user authentication are included as security features. These controls should be configured and evaluated according to the sensitivity of the stored records. A formal threat model and measured security assessment are not part of the present work.

D. Permanent Storage, Retrieval, and Expiry Editing

Not every document needs expiry monitoring, so PingVault includes a storage category for permanent documents that generates no reminders. Users can search for, view, and retrieve files, and can update an expiry date when a document is renewed without uploading the file again. This separates document content from changing renewal metadata.

E. Dashboard and Interface

The Analysis module summarizes document activity through interactive charts covering file types such as PDF, JPG, and PNG, uploaded documents, reminders sent, permanent documents, and documents nearing expiry; ApexCharts supplies the charting layer [18]. The interface offers light and dark themes and structured navigation (My Vault, Reminders, Storage, Analysis, About, and Logout), and relies on Bootstrap's responsive, mobile-first components [16].

V. IMPLEMENTATION AND TECHNOLOGY STACK

A. Application Architecture

PingVault follows a full-stack web application model (Fig. 1). The backend manages authentication, document metadata, storage operations, expiry logic, and reminder processing. The frontend provides forms, navigation, document lists, theme controls, and dashboard visualizations. A relational database stores records for users, files, expiry dates, and reminder status.

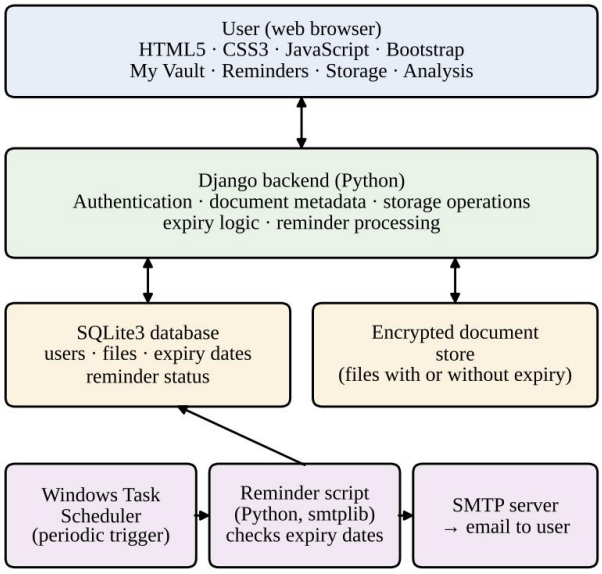


Fig. 1. System architecture of PingVault.

B. Backend and Database

The backend uses Python and Django, which provide a structured web framework and an email abstraction that can route messages through development or SMTP delivery [13]. SQLite3 serves as a lightweight database for application data; its documentation covers the SQL language, interfaces, file-based operation, and appropriate-use considerations [15].

C. Frontend

The frontend uses HTML5, CSS3, JavaScript, and Bootstrap. Bootstrap's responsive, mobile-first approach suits the desktop, tablet, and smartphone layouts required [16], and JavaScript supports navigation, forms, theme switching, and chart rendering.

D. Reminder Automation

A scheduled process checks expiry metadata and starts email notifications for relevant documents. SMTP provides the protocol-level client capability for sending mail to an SMTP or ESMTP listener [14], and Windows Task Scheduler triggers the process at scheduled intervals. Table II lists the components of the stack.

TABLE II
TECHNOLOGY STACK OF PINGVAULT

Layer	Technology	Role
Backend	Python, Django [13]	Authentication, metadata, storage operations, expiry logic, reminder processing
Database	SQLite3 [15]	Records for users, files, expiry dates, and reminder status
Frontend	HTML5, CSS3, JavaScript, Bootstrap [16]	Forms, navigation, themes, responsive layout
Email	SMTP via Python smtplib [14]	Delivery of reminder messages
Scheduling	Windows Task Scheduler	Periodic trigger of the reminder process
Visualization	ApexCharts [18]	Interactive dashboard charts

E. Data Visualization

ApexCharts represents document and reminder activity as interactive charts, helping users identify file-type distributions, upload activity, reminder counts, and expiry patterns [18], [19].

VI. SYSTEM TESTING AND EVALUATION

A. Testing Scope

System testing assessed the main PingVault modules: document upload, expiry tracking, email reminders, analytics, navigation, and responsive presentation. The objective was to confirm functional stability, usability, and compatibility across commonly used platforms, and to verify that the workflow from storage to notification operates as a connected system rather than as isolated features. Table III summarizes the scope.

TABLE III
SCOPE OF QUALITATIVE TESTING

Test area	Scope examined
Functional	Uploading documents; assigning or editing expiry dates; storing documents without expiry; retrieving files; generating reminders; displaying analytical summaries
Cross-browser	Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari: upload, reminder-related interactions, navigation, consistent HTML, CSS, and JavaScript behavior; fallback considerations for older browsers
Responsive / cross-device	Desktops, laptops, tablets, and smartphones: layout scaling, navigation menus, document listings, action buttons, upload forms, dashboard components, chart readability

B. Responsive Behavior

Particular attention was given to collapsible menus, touch-friendly controls, scrollable document tables, and the behavior of charts on smaller screens. Bootstrap's grid and responsive design approach [16], [17] supported this testing strategy.

C. Evaluation Limitations

The testing reported here is qualitative. Numerical measurements such as response time, notification latency, test-case counts, failure rates, storage capacity, user sample size, and security-test results were not recorded, so this paper makes no quantitative claims about performance or usability. A future evaluation should define reproducible test cases and report measurable results.

VII. DISCUSSION

PingVault addresses a practical gap between passive document storage and active document lifecycle management. A conventional repository preserves a document but does not necessarily connect it to its expiry date, renewal status, or notification workflow. By combining a vault, metadata, automated reminders, and analytics, PingVault treats document management as an ongoing process rather than a one-time upload. Evidence from automated email notification studies [10], [11] suggests that alerts raise awareness and speed up follow-up but do not by themselves guarantee action, which supports pairing reminders with a dashboard that keeps documents nearing expiry visible.

The system also shows how common web technologies can be combined into a focused administrative application. Django and SQLite provide a lightweight backend, while HTML5, CSS3, JavaScript, Bootstrap, and ApexCharts support the interface and analytical layer [13], [15], [16], [18], and SMTP-based automation connects stored metadata with external communication [14].

However, the security and scalability claims require validation before production use with highly sensitive records. Encryption key management, authorization boundaries, audit logging, backup procedures, malware scanning, secure deletion, and protection against unauthorized file access should be specified and tested, in line with the cloud storage security literature [6], [7]. Migration from SQLite to a client/server database may also become appropriate if concurrent use or data volume grows.

VIII. CONCLUSION AND FUTURE SCOPE

This paper presented PingVault, a secure document storage and expiry reminder system that improves the organization and lifecycle management of important files. The platform centralizes documents, associates files with expiry metadata, supports permanent storage, sends automated email reminders, and visualizes document activity through an analysis dashboard. Its implementation combines Python, Django, SQLite3, HTML5, CSS3, JavaScript, Bootstrap, SMTP, Windows Task Scheduler, and ApexCharts [13]–[16], [18]. Qualitative testing covered core functionality, major browsers, and multiple device classes; because numerical performance and security measurements are not yet available, further empirical evaluation is needed to establish production readiness.

Future work will proceed in several directions: role-based access control and document-sharing permissions; audit logs for uploads, downloads, edits, deletions, and reminder events; improved encryption-key management and document-level security policies; configurable reminder schedules with multiple notification channels; search by document type, owner, tags, expiry period, and status; backup, recovery, and secure deletion workflows; automated, load, and security testing with published metrics; usability evaluation with representative users against accessibility guidelines; migration to a production database and scalable file storage; and trend analysis with configurable dashboard reports.

ACKNOWLEDGMENT

The authors thank Ether Infotech for the internship during which PingVault was developed.

REFERENCES

[1] G. P. Johnston and D. V. Bowen, "The benefits of electronic records management systems: A general review of published and some unpublished cases," *Rec. Manag. J.*, vol. 15, no. 3, pp. 131–140, 2005.

[2] M. K. Bizi, A. Mohammed Ghazali, and M. N. Yunus, "The adoption of electronic records management system in an organization: A systematic literature review approach," *J. Inf. Knowl. Manag.*, vol. 15, no. SI2, pp. 280–292, Aug. 2025.

[3] Government of India, "Digital Locker technology specification (DLTS)," ver. 2.3, Mar. 2015.

[4] A. Anas and T. A. Salim, "Tinjauan literatur sistematis pemanfaatan electronic document management system bagi organisasi dalam menunjang manajemen pengetahuan," *Berkala Ilmu Perpustakaan dan Informasi*, Dec. 2022.

[5] R. Kumar and A. Baghel, "DigiLocker (Digital Locker – A better India for tomorrow)," *Int. J. Adv. Res.*, Jun. 2019, doi: 10.21474/IJAR01/9265.

[6] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C.-C. J. Kuo, "Survey on securing data storage in the cloud," *APSIPA Trans. Signal Inf. Process.*, vol. 3, p. e7, 2014.

[7] P. Yang, N. N. Xiong, and J. Ren, "Data security and privacy protection for cloud storage: A survey," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3009876.

[8] P. V. Maitri and A. Verma, "Secure file storage in cloud computing using hybrid cryptography algorithm," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, 2016, pp. 1635–1638.

[9] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Federal Inf. Process. Stand. Publ. 197, 2001.

[10] A. K. Dalal *et al.*, "Impact of an automated email notification system for results of tests pending at discharge: A cluster-randomized controlled

trial," *J. Am. Med. Inform. Assoc.*, vol. 21, no. 3, pp. 473–480, 2014, doi: 10.1136/amiajnl-2013-002030.

[11] A. K. Dalal *et al.*, "The impact of automated notification on follow-up of actionable tests pending at discharge: A cluster-randomized controlled trial," *J. Gen. Intern. Med.*, vol. 33, no. 7, pp. 1043–1051, Jul. 2018, doi: 10.1007/s11606-018-4393-y.

[12] K. Murat, I. Koshim, N. Kelessov, and A. Ormanov, "Automated notification system for document expiration management," CSCI 409 Senior Project II Final Rep., Nazarbayev Univ., Spring 2025.

[13] Django Software Foundation, "Sending email," *Django Documentation*, ver. 6.1. [Online]. Available: https://docs.djangoproject.com/

[14] Python Software Foundation, "smtplib — SMTP protocol client," *Python 3.15.2 Documentation*. [Online]. Available: https://docs.python.org/3/library/smtplib.html

[15] SQLite Consortium, "SQLite documentation." [Online]. Available: https://www.sqlite.org/docs.html

[16] The Bootstrap Authors, "Get started with Bootstrap," *Bootstrap Documentation*, ver. 5.3.8. [Online]. Available: https://getbootstrap.com/docs/5.3/getting-started/introduction/

[17] E. Marcotte, "Responsive web design," *A List Apart*, no. 306, May 25, 2010.

[18] ApexCharts, "Documentation." [Online]. Available: https://apexcharts.com/docs/

[19] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, Jan. 2019, doi: 10.1109/TVCG.2018.2864903.