BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE EDUCATION

DEPARTMENT OF EDUCATIONAL TECHNOLOGY

DESIGNING AN INTELLIGENT DIGITAL LIBRARY SYSTEM.

BY

Recktyse Mushave (B232431B)

SUPERVISOR: Dr T. CHIKEREMA

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF THE BACHELOR OF SCIENCE EDUCATION HONOURS DEGREE IN COMPUTER SCIENCE

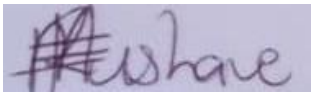
Year SEPTEMBER 2025

RELEASE FORM

Title of the dissertation: **Designing an Intelligent digital library system.**

1. **To be completed by the student**

   I certify that this dissertation is in conformity with the preparation guidelines as presented in the Faculty Guide and Instructions for Typing dissertations.

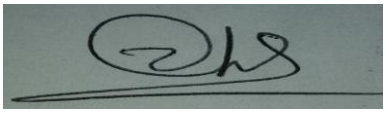


05/09/2025

(Signature of student) (Date)

To be completed by the supervisor

This dissertation is suitable for submission to the Faculty. This dissertation should be checked for conformity with Faculty guidelines.

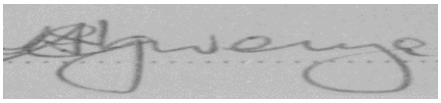


05/09/2025

(Signature of Supervisor) (Date)

2. **To be completed by the chairperson of the department**

   I certify to the best of my knowledge that the required procedures have been followed and the preparation criteria have been met for this dissertation



05/09/2025

(Signature of the chairperson) (Date)

APPROVAL FORM

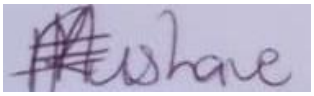
Name of the student: Recktyse Mushave **Registration Number**: B232431B

Title of the dissertation: **Designing an Intelligent digital library system.**

Degree Title: Bachelor of Science Education Honors Degree in Computer Science.

Year of completion: 2025

Permission is hereby granted to Bindura University of Science Education to single copies of this dissertation and to lend and sell such copies for private, scholarly scientific purposes only. The author reserves the rights and neither the dissertation nor extensive extracts from it be granted or otherwise be replicated without the author's consent.



05/09/2025

(Signature of student) (Date)

ACKNOWLEDGEMENT

I am extremely thankful to God for giving me the power, wisdom, and determination to carry out and complete this research project. His guidance has been a constant source of motivation.

I would want to convey my heartfelt gratitude to my supervisor, Dr Chikerema, for her ongoing support, expert advice, and insightful feedback during this project. Your encouragement and guidance were critical in shaping this study. Special appreciation goes to the teachers, administration, and students for their enthusiastic engagement and cooperation. This study would not have been feasible without your willingness to participate.

Thank you, my family and friends for your constant support, patience and encouragement during this academic journey. Your belief in me provided me strength at the most trying circumstances.

Table of Contents

List of Figures

1 Chapter 1: Setting the Scene

1.1 Introduction

This chapter presents the background and motivation, for the need of an intelligent approach to digital libraries in modern educational environments. The problem statement section articulates the shortcomings of current digital library systems, especially regarding search efficiency, personalisation, and classification. The objectives and goals define what the system aims to achieve, while the significance justifies the relevance and importance of the project. The scope and limitations outline what is included and excluded in the project.

1.2 Background and Motivation

The rapid advancement of technology and widespread mobile device usage among students, lecturers, and researchers have significantly expanded access to academic materials in electronic formats. Consequently, this shift has led to a transformation from traditional physical learning methods to e-learning, and from conventional physical libraries to digital repositories and e-book platforms such as ebrary. In today's academic world the volume of digital content is rapidly growing. Digital library enables users to access its resources remotely through the internet or specific applications. This convenience supports distance learning, remote research, and flexible studying, which is particularly beneficial in regions where physical libraries are scarce (Zilahy, 2025). Digital library breaks barriers to information by making resources accessible anytime and anywhere catering to the needs of students, teachers and parents. This digital library is built on advanced technologies such as metadata systems, cloud storage and search algorithms to ensure efficient organisation and retrieval of content. It plays a critical role in preserving educational materials by converting them into digital formats to protect against physical deterioration.

Digital libraries eliminate the need for physical space, printed copies, and multiple editions of the same book. A single digital copy can be accessed simultaneously by many users without additional printing or distribution costs (Singh & Sharma, 2024). Institutions save significantly on storage, maintenance, and staffing. Digital libraries are environmentally friendly. They reduce the need for paper, printing, and shipping, contributing to conservation efforts and lowering the carbon footprint associated with traditional publishing and distribution (Sridhar and Rajesh, 2023). Digital

libraries support efficient information retrieval. Powerful search tools and algorithms enable users to quickly locate specific content by using keywords, topics, or authors. Some systems also use artificial intelligence to personalize search results and recommend related materials based on user behaviour (Prabu & KavithA, 2019). Digital libraries can incorporate multimedia content such as audio recordings, videos, animations, and interactive modules, enriching the learning experience. This is especially useful in education, where diverse content formats cater to different learning styles (Chowdhury, 2010).

Digital libraries have transformed from static into dynamic, user-centric repositories. Siringo, Abighael and Gabriela (2025) asserts that the rapid growth of information technology has enabled libraries to provide round the clock remote access to vast collections of books, articles, and multimedia, effectively breaking down physical barriers to learning. In schools and universities, digital libraries are now common tools that empower students with extensive, credible resources beyond the classroom. Roy, Mallikraj and Moradia (2024) notes that AI can greatly enhance library services improving resource discovery and delivering personalised user experiences. Techniques like Natural Language Processing (NLP) and recommendation algorithms enable smart search interfaces and tailored content suggestions. For example, machine learning driven recommenders can suggest books and articles based on a student's past reading or curriculum, while intelligent retrieval can interpret queries more naturally. By analysing user behaviour and content metadata, AI can automate tedious tasks and make digital library systems more adaptive and user-friendly.

Despite broader access, many existing digital library platforms remain limited. They often rely on basic keyword search and manual cataloguing leave students overwhelmed by the volume of content (Deshpande, 2018). Such systems do not automatically adapt to user needs search results can be irrelevant or incomplete. There is little personalisation or insight into student behaviour. Static interfaces and outdated workflows in school libraries fail to leverage data resulting in missed opportunities for helping learners find the right materials efficiently.

In rural or underdeveloped areas, may lack access to the necessary technology or reliable internet connections. This limits the inclusivity and reach of digital libraries, reinforcing existing educational inequalities (Ashikuzzaman, 2023). Technical challenges also pose significant concerns. System crashes, software bugs, and server downtime can interrupt access to resources. Continuous maintenance, updates, and IT support are essential to keep the system functioning

effectively (Initiative, 2023). Digital libraries are subject to copyright and licensing issues. Content must be legally obtained and shared according to intellectual property laws. This can restrict the availability of certain resources or impose limitations on their usage (Elmasr & Narethe, 2017). Reading long texts on screens can lead to eye strain and discomfort, making it less appealing than reading from printed materials.

This project is motivated by the need to modernise school library systems using intelligent solutions that can cater to the evolving learning needs of students and educators. The development of an Intelligent Digital Library System (IDLS) aims to bridge the gap between the availability of digital resources and their effective utilisation in academic environments.

1.3 Problem Statement

Despite the increasing availability of digital academic resources, most digital libraries lack advanced intelligent classification and user-specific search functionalities. Siringo et al (2025) says the limitation hampers efficient resource discovery, retrieval, and organisation, especially in environments with large and continuously growing collections. Consequently, users face difficulties in accessing relevant materials quickly and effectively, which affects the overall efficiency and usability of digital libraries. Roy et al (2024) postulates that there is a need for an enhanced digital library system that incorporates intelligent algorithms to improve resource classification, personalised search, and retrieval processes, thereby meeting the evolving needs of educational institutions. The lack of intelligent classification and user-adaptive search capabilities in digital libraries limiting efficient resource access. This results in inefficiencies especially in environments with vast and continually growing databases of resources.

1.4 Objectives and Goals

The project goals are:

i. to improve accessibility of academic resources by creating a centralised, easy to navigate digital platform for students and teachers.
ii. To improve content accessibility that ensures students and staff can access the library's digital resources anytime, anywhere, and offline.
iii. build user-profiles and recommendation features so the system tailor's suggestions to each student's history and interests.

1.5 Significance of the Project

The intelligent digital library system can enhance educational outcomes by personalising access to information and improve student engagement. Beel, Langer, Genzmehr and Nürnberger (2013) says when students get tailored content and immediate feedback, they are more motivated and learn more effectively. Ashikuzzaman (2023) indicates that AI-driven adaptive learning increases engagement and academic performance.

Intelligent digital library in schools can give all students, including those in under-resourced settings better access to quality educational content. Over time, this could translate into higher literacy and more equitable learning outcomes.

Usage analytics can help school administrators and educators identify gaps in resource provision and areas needing curriculum support.

Another important contribution of the system is the secure handling of downloaded educational materials. Beel et al (2013) states that to prevent casual copying and unauthorised redistribution of academic content, the system must ensure that downloaded eBooks are stored in secure, app-specific directories on the user's device and these directories are inaccessible to other apps and file browsers, aligning with best practices in digital rights management. This promotes copyright compliance, protects intellectual property, and maintains the integrity of the digital content distributed through the platform.

The system prepares schools to adapt to future educational demands, including remote learning, digital assessment, and AI-assisted instruction.

1.6 Scope and Limitations

The scope of this project includes the design and implementation of an intelligent digital library that supports viewing educational resources, document uploads, intelligent classification, semantic search, and content recommendation. The system will be developed as a web-based platform using Django and Python, with a backend database to manage users and resources.

Certain limitations are acknowledged:

i. The initial training dataset for classification may be limited, affecting the accuracy of early recommendations.

ii. Advanced features such as multilingual support, voice-based search, or real-time collaboration are outside the initial scope.
iii. Internet access is required for full system functionality, which may pose challenges in areas with poor connectivity.
iv. The system will focus primarily on textual documents and may not fully support multimedia formats in the current version.

1.7 Overview of the Report Structure

This report is organised into eight main chapters, each addressing a key aspect of the project:

Chapter 1: Introduction

The foundation of the entire project is being outlined. It begins with the background and motivation, explaining the need for an intelligent approach to digital libraries in modern educational environments. The problem statement section articulates the shortcomings of current digital library systems, especially regarding search efficiency, personalisation, and classification. The objectives and goals define what the system aims to achieve, while the significance justifies the relevance and importance of the project. The scope and limitations outline what is included and excluded in the project.

Chapter 2: Literature Review

Critical analysis of existing research, systems, and technologies relevant to digital libraries and intelligent systems. It explores the evolution and current state of digital library systems, highlighting their architecture, capabilities, and limitations. A review of intelligent classification techniques, such as supervised and unsupervised machine learning models. The role of search and retrieval algorithms, including keyword-based, Boolean, and semantic search approaches. The concept of user personalisation and recommender systems, explaining how systems can adapt to user behaviour. A gap analysis to justify the need for this specific project and to position it within the current research landscape.

Chapter 3: Requirements Specification

The functional and non-functional expectations of the system are systematically documented. It includes stakeholder analysis identifying the key users and beneficiaries that are students,

librarians, teachers, administrators. Functional requirements, detailing what the system should do, such as uploading documents, searching resources, generating recommendations, and managing users. Non-functional requirements, covering aspects such as system performance, scalability, usability, and security. Use cases and user scenarios, presenting narrative descriptions and diagrams to explain user interactions with the system. System constraints, which define the technological or organisational boundaries within which the system is developed.

Chapter 4: System Design

The technical design and architecture of the intelligent digital library system is being outlined. Key areas include the overall system architecture, showing how various components (frontend, backend, database, AI modules) interact. A detailed data model and database design, including entity-relationship diagrams and table structures to manage users, documents, and metadata. A breakdown of the system modules and components, such as the document classifier, search engine, user profiling system, and admin panel, User Interface Design, which includes wireframes and mockups that show how the system will look and function from the user's perspective. A description of the intelligent algorithms used for classification, recommendation, and semantic understanding. Interaction and workflow diagrams, such as sequence and activity diagrams, illustrating how the system operates during various user interactions.

Chapter 5: Implementation Details

This chapter delves into the practical realisation of the system. It describes the development environment and tools used, including programming languages (Python), frameworks (Django), libraries (Scikit-learn, NLTK), and databases (Sqlite3). The implementation of core modules, such as: The classification engine for categorizing uploaded documents, The search and retrieval system with semantic and keyword-based options and the user personalization module that tailors' recommendations and suggestions.

The integration of components, ensuring seamless communication between frontend, backend, database, and AI engines. Challenges encountered during development, such as dataset availability, algorithm tuning, interface issues, and user feedback, and how these were resolved.

Chapter 6: Testing and Evaluation

This chapter presents a comprehensive discussion on the procedures employed to validate and assess the Intelligent Digital Library System. It discusses the testing strategies adopted, the specific test cases and scenarios used, performance metrics applied, the evaluation outcomes, user feedback gathered, and outlines the key limitations of the implemented system.

Chapter 7: Discussion

This chapter analyses the system's performance and its ability to solve the challenges identified in earlier stages. It also compares the system with existing platforms like greenstone and Dspace. Explores its educational and user impact. Examine how well the system performed, especially in handling intelligent search, recommendations, and user interaction. The section explains how the system overcomes previous limitations in traditional digital libraries, such as poor search functionality and lack of personalisation. Also, what the system does well, such as usability and AI integration, and identify areas that need enhancement, such as chatbot capabilities and analytics.

Chapter 8: conclusion and future work

This chapter summarises the major contributions and findings of the project. It also outlines recommendations for future improvements and evaluates the system's readiness for real-world use. Review the key features developed, including AI-driven search and personalised recommendations, and how they enhance digital library systems. Summarises what was discovered through testing and development, particularly regarding usability and performance. Suggest improvements for future versions, such as multilingual support, better chatbot integration, and mobile responsiveness. Evaluate the feasibility of deploying the system in real environments, such as schools.

1.8 Chapter Summary

The chapter introduced the background and motivation for the Intelligent Digital Library System. It outlined the transition from traditional libraries to digital platforms, identified limitations such as poor search efficiency and lack of personalisation, and defined the problem statement. The objectives, significance, scope, and limitations of the project were clearly stated, setting the foundation for the research.

2 Chapter 2: Literature Review

2.1 Introduction

This chapter provides an analysis of existing systems and technologies relevant to digital libraries and intelligent systems. It explores the evolution and current state of digital library systems. A review of intelligent classification techniques, such as supervised and unsupervised machine learning models. The role of search and retrieval algorithms, including keyword-based, Boolean, and semantic search approaches. The concept of user personalisation and recommender systems, explaining how systems can adapt to user behaviour. A gap analysis to justify the need for this specific project and to position it within the current research landscape.

2.2 Overview of Digital Libraries and Their Limitations

Digital libraries are systems that collect, manage, and provide access to digital content such as academic resources, e-books, and multimedia resources. Shiv (2024) defined digital library as a web-based system that provides access to various academic information and services encompassing collections includes e-books, journals, multimedia and user services. Digital libraries are web-based systems that organise and provide access to collections of digitised texts, images, multimedia, and other scholarly resources. They play a critical role in supporting learning, teaching, and research. According to Chowdhury (2010) these libraries use metadata to describe, organise, and retrieve digital objects effectively. They support core functions such as search, retrieval, metadata-driven indexing, preservation, and delivery of academic content to users (Siringo et a, 2025). Digital libraries offer searchable interfaces, metadata catalogues and sometimes user tools such as annotations and alerts.

As the use of digital libraries continues to grow, the secure storage and ethical distribution of downloaded eBooks have become increasingly significant challenges. The transition from physical to digital resources introduces risks such as unauthorised sharing, piracy, and violation of copyright licenses (Patel & Gupta, 2024). Secure storage of digital files particularly on user devices is a critical measure to mitigate these risks and preserve intellectual property rights. Modern digital libraries implement sandboxed storage, where downloaded files are saved within app-specific directories that are inaccessible to other applications or direct user access through file explorers (Beel et al ,2023). This approach is widely adopted in mobile environments like Android and iOS,

where internal storage containers are isolated by the operating system, effectively preventing casual copying or unauthorised redistribution of educational materials. According to Shiv (2024) to further enforce content protection, digital libraries may incorporate encryption techniques such as AES-256 to protect files at rest. These encrypted files are decrypted only within authenticated user sessions in the app. According to Martins and Faria (2025), combining encryption with token-based access control significantly reduces the risk of content leakage and enables time-limited, role-restricted downloads, enhancing copyright compliance. Some platforms simulate Digital Rights Management (DRM) mechanisms without using proprietary DRM software. For instance, watermarking downloaded content with user-specific metadata such as a user ID, institution, and timestamp acts as a deterrent against unauthorised sharing (Roy et al., 2024). In the event of a leak, administrators can trace the origin and take corrective measures.

Despite their strengths, many Digital libraries face severe limitations. Zilahy (2025) says the sheer volume of scholarly content leads to information overload users often retrieves thousands of hits for broad queries, many of which are irrelevant. Traditional keyword-based search can return long result lists where most relevant items are suppressed or invisible amid irrelevant ones researchgate.net. Similarly, poor metadata quality incomplete, inconsistent, or erroneous metadata is a common problem that hinders accurate indexing and retrieval. Another challenge is lack of intelligent services. Many academic digital libraries still use only basic cataloguing and search tools, with few employing AI-driven features. Siringo (2025) review notes that applications of AI in digital library are sparse so libraries often rely on manual or legacy systems for routine functions. Finally, limited interoperability among repositories and systems remains a barrier scholarly content is spread across thousands of publishers, repositories and journals, typically without common exchange protocols. Addressing these gaps is a primary motivation for the research.

Secure storage solutions must balance privacy and usability. Overly restrictive DRM systems can discourage use, especially in education. Beel et al (2023) argue for “user-trusting DRM-lite” systems, where ethical sharing is promoted through gentle restrictions and institutional education rather than aggressive enforcement. Emerging AI-powered libraries have begun integrating intelligent access control, where user behaviour analytics help determine whether download

activity is suspicious. This allows dynamic policy enforcement that is disabling downloads for flagged users or limiting offline access in high-risk environments (Siringo et al., 2025).

2.3 Intelligent Classification Techniques in Digital Libraries

Advanced classification and content-organising techniques are increasingly applied to digital library collections. These include both supervised and unsupervised machine learning methods, as well as deep learning and semantic analysis. According to Martins (2024) supervised classifiers that includes SVMs, Naive Bayes and decision trees can be trained on labelled corpora to automatically tag or categorise new documents by categories or subjects. Roy et al (2024) concurs unsupervised methods like clustering or topic modelling that includes Latent Dirichlet Allocation (LDA), k-means and hierarchical clustering group related items. Natural Language Processing (NLP) techniques such as word embedding models or transformer-based encoders are used to represent document semantics for both classification and similarity search. Topic modelling and text mining have been demonstrated to yield valuable insights from library collections. Lamba and Madhusudhan (2022) reviews show topic modelling can reveal hidden themes in library datasets and improve services such as current awareness and recommender alerts while noting that librarians often lack familiarity with these tools. Pirniakan et al. (2025) developed a self-supervised graph convolutional network for keyword-centric topic modelling of digital library publications. By learning semantic keyword relationships, their model greatly improved topic clustering accuracy compared to traditional Latent Dirichlet Allocation (LDA) like approaches. Such studies illustrate how cutting-edge AI graph neural nets and self-supervised learning can bridge the gap between advanced algorithms and practical digital library topic analysis. Other deep learning approaches include using convolutional or recurrent neural networks for content categorisation, and leveraging pretrained language models for semantic tagging. For instance, named-entity recognition or semantic role labelling can annotate documents with standardised concepts. Moreover, semantic analysis using linked data and ontologies is gaining traction the Semantic Web aims to create a global network of data connected through semantic relationships.

2.4 Search and Retrieval Algorithms

Digital libraries traditionally rely on keyword-based retrieval and Boolean query models. Goh (2012) states that a simple search treats documents as bags of words, matching query terms exactly. Boolean search lets users combine keywords precisely. Ranking is often done via the vector space

model where documents and queries are represented as Term Frequency- Inverse Document Frequency (TF–IDF) term vectors and ranked by cosine similarity (Baeza-Yates & Ribeiro-Neto, 2011). Russell and Norvig (2020) states that classical models like TF–IDF or Okapi BM25 still underlie most digital library search engines. However, these techniques have well-known drawbacks poor handling of synonyms, polysemy, and user query ambiguity. As early reviews note, search-algorithms have a dramatic effect on how resources are indexed and will improve the discovery of resources, yet in practice many users issue only simple keyword queries. Studies have found the majority of results in keyword search are irrelevant unless queries are very precise. To address this, modern retrieval methods are emerging in digital library. Semantic search techniques use concept expansion and ontologies to interpret user intent. W3C (2018) states that the semantic web paradigm advocates expressing metadata in Resource Description Framework (RDF) and answering semantic queries so that machines can understand relationships between terms. This enables searches like querying by a concept rather than exact words. In practice, linked-data-enabled search can combine terms with related subjects via Simple Knowledge Organisation System (SKOS) or Web Ontology Language (OWL) links to broaden results intelligently. Neural IR and deep learning have begun to transform scholarly search. Embedding-based retrieval dense vector search uses document and query embeddings e.g. Sentence-BERT and finds nearest neighbours in a continuous space, capturing semantic similarity beyond keywords. Neural ranking models are also being experimented within research. Though still nascent in most library systems, these approaches hold promise to overcome some keyword search limitations.

2.5 User Personalisation in Information Systems

According to Beel et al (2013) user personalisation in an information system refers to the process of tailoring system content, layout, functionality, and recommendations to meet the specific needs, preferences, behaviours, and context of individual users. The goal is to enhance the user experience by making interactions more relevant, efficient, and engaging. In a digital library, user personalisation refers to the system’s ability to deliver content and services tailored to each user's needs, learning goals, research interests, and usage history. Joo and Choi (2015) postulates that as digital libraries grow in size and complexity, personalisation becomes essential to improve user navigation, search efficiency, and resource discovery. Digital libraries increasingly incorporate user modelling and personalisation to tailor content delivery. Russell and Norvig (2020) says user

models capture a person's interests or behaviour via explicit profiles or implicit log analysis and drive adaptive services. For example, adaptive interfaces can adjust visible categories or layout based on a user's role or past usage. Recommender systems are a major personalisation component. These algorithms suggest relevant documents based on the user logs. Traditional approaches include content-based filtering matching items similar to those a user has liked before and collaborative filtering using preferences of like-minded users. Hybrid systems combine both, leveraging item attributes and user-item interactions to improve recommendations. Together these approaches aim to make Intelligent Digital library system more responsive to individual needs and learning contexts.

2.6 Existing Intelligent Digital Library Models

There are various digital models that are used in contemporary libraries and these are described in the succeeding sections.

2.6.1 User-Centred Intelligent Library Model

At the heart of any effective digital library is the user. The User-Centred Intelligent Library Model emphasizes on creating a personalised and adaptive interaction between the user and the system. This model builds user profiles by tracking search history, preferences, learning styles, and usage patterns. Chowdhury (2010) says the system dynamically customises the user interface, recommends relevant materials, and adjusts the presentation of content according to the user's academic or professional background.

This model typically uses machine learning algorithms such as clustering, decision trees, or neural networks to segment users and adapt the system's behaviour. By implementing both collaborative filtering and content-based filtering the system enhances the relevance of its services. It is especially useful in academic environments where learners' needs vary widely.

2.6.2 Intelligent Agent-Based Library Model

The Agent-Based Model introduces autonomous software agents that perform specific functions within the digital library ecosystem. According to Russell and Norvig (2020) each agent operates independently and communicates with others to fulfil complex tasks such as searching for information, organising resources, managing metadata, or even providing user assistance.

In this model, search agents handle user queries by retrieving relevant documents, profile agents maintain and update user profiles, while recommendation agents suggest resources based on user activity. These agents improve efficiency and reduce human intervention by learning from past actions and adapting to new inputs. Multi-agent systems are scalable, and their decentralised nature makes them ideal for large, distributed digital libraries.

2.6.3 Semantic-Based Digital Library Model

Traditional digital libraries primarily rely on keyword-based searches, which are limited in understanding the context or meaning of search queries. Baeza-Yates and Ribeiro-Neto (2011) state that Semantic-Based Digital Library Model addresses this limitation by incorporating semantic web technologies that enable machines to understand and process the meaning of information.

This model uses ontologies formal representations of knowledge in a particular domain to categorise and relate concepts. Technologies like RDF (Resource Description Framework) and OWL (Web Ontology Language) are used to structure metadata and create linked data, facilitating more intuitive and intelligent search results. Instead of simply matching keywords, the system can interpret the user's intent and provide conceptually relevant resources.

This model is particularly powerful in academic and scientific digital libraries, where precise and meaningful retrieval is essential. For instance, the Greenstone Digital Library software has been extended to support semantic search by integrating domain ontologies.

2.6.4 Hybrid Recommendation Model

Recommendation systems are crucial in guiding users to relevant resources, especially in large digital libraries. The Hybrid Recommendation Model combines various recommendation techniques to improve the quality and relevance of suggestions. Beel et al (2013) says this model integrates collaborative filtering (based on user behaviour), content-based filtering (based on item characteristics), and demographic filtering (based on user profiles). Some implementations also consider context-aware filtering, which includes parameters like time, location, and device used.

According to Martins (2024) hybrid model overcomes limitations such as the "cold start" problem where new users or items lack sufficient data and enhances adaptability. Many platforms, such as

Mendeley and ResearchGate, use hybrid recommenders to assist researchers in finding relevant publications, collaborators, and datasets.

2.6.5 Intelligent Metadata Extraction and Classification Model

A significant challenge in digital libraries is the manual creation of metadata for each document. Roy et al (2024) states that Intelligent Metadata Extraction and Classification Model automates this process using AI technologies such as Natural Language Processing (NLP) and Optical Character Recognition (OCR).

This model analyses documents in formats like PDF, Word, and scanned images to extract useful metadata such as titles, authors, abstracts, and keywords. Martins (2024) says classification algorithms then organise the documents into relevant categories or subjects. Tools such as Apache Tika, spaCy, and Scikit-learn are commonly used in these implementations. This model is highly efficient for large-scale digital archives, such as those maintained by universities or research institutions. For example, the CERN Document Server uses ML algorithms to automatically extract and classify metadata for scientific publications.

2.6.6 Intelligent Conversational Library Systems

In recent years, the integration of conversational agents or chatbots has enhanced the interactivity of digital libraries. Russell and Norvig (2020) says Intelligent Conversational Library Model enables users to interact with the library through natural language queries, either through text or voice. These systems use natural language understanding (NLU) and dialogue management to interpret user input and provide relevant responses. They can help users find books, answer FAQs, navigate the library system, or recommend resources based on the conversation.

Such models use frameworks like Google Dialog flow, Microsoft Bot Framework, or OpenAI GPT models. Libraries implementing these tools offer an engaging and accessible experience, particularly for younger users or those unfamiliar with complex search systems.

2.7 Summary of Gaps and Opportunities

The literature converges on several persistent gaps in digital libraries and points to promising research directions. Key challenges include improving metadata creation and quality, enabling smarter search, and delivering user-centred services. Traditional systems store downloaded files in

publicly accessible folders, making them vulnerable to copying, editing, and unauthorised sharing. Xie and Matusiak (2024) highlights that future digital library research should focus on using AI to automate metadata generation and ensure its quality. Since manual cataloguing is a bottleneck, applying machine learning to generate or validate metadata such as auto-tagging subject headings, extracting entities is a ripe opportunity. The integration of AI and deep learning throughout Digital library is another area for growth. Currently, AI use is sparse but promising. Future research should explore embedding such tools into library interfaces, as well as personalised assistants that leverage a library's own collection. Similarly, the adoption of linked data and semantic web technologies is expected to enhance interoperability and discovery. While platforms like CORE aggregate content, wider adoption of shared ontologies and knowledge graphs could let Digital library truly interoperate at the data level. User personalisation also remains under-developed. Cross-library user profile federation is a niche area ripe for innovation. Moreover, evaluation methods for digital library interfaces need further work existing studies suggest using diverse methods logs, interviews to better understand user needs. By combining emerging AI techniques (NLP, deep learning) with the foundational models of digital library architecture, the next generation of academic digital libraries can become far more intelligent, user-focused, and interoperable. Integrating secure storage practices with intelligent library systems ensures that offline access remains compliant with copyright laws while preserving the usability and trustworthiness of the platform. It balances user convenience with publisher and institutional content protection requirements.

2.8 Chapter Summary

To sum up, a critical reviewed existing digital library systems and related research was established. It examined current approaches to classification, search, personalisation, and secure access, highlighting gaps such as inadequate metadata quality, reliance on keyword search, and minimal use of artificial intelligence. The review positioned the proposed system as a response to these shortcomings.

3 Chapter 3: Requirements Specifications

3.1 Introduction

In this chapter, the functional and non-functional expectations of the system are systematically documented. It includes stakeholder analysis identifying the key users and beneficiaries that are students, librarians, teachers, administrators. Use cases and user scenarios, presenting narrative descriptions and diagrams to explain user interactions with the system. System constraints, which define the technological or organisational boundaries within which the system is developed.

3.2 Stakeholder Analysis

Stakeholders are individuals, groups, or entities who interacts with the system directly or indirectly and whose needs and expectations are considered during the system's development (Johnson & Brian, 2018). In the context of the Intelligent Digital Library System, several key stakeholders have been identified, each playing a critical role in the system's operation and value chain.

Foremost among the stakeholders are students, who are the primary users of the system. They rely on the digital library for accessing eBooks, academic resources, performing intelligent searches, and receiving personalised content recommendations to support their learning journey. Librarians serve as custodians of content within the system. They are responsible for uploading academic materials, tagging documents, managing metadata, and monitoring system usage. Their role is crucial in ensuring that the repository remains current, relevant, and accurately classified using both manual input and intelligent algorithms. Teachers are another essential group of stakeholders. They contribute to the system by suggesting materials, guiding students toward appropriate resources, and evaluating the relevance and quality of content. System administrators are responsible for maintaining the security, functionality, and performance of the platform. They manage user roles, system settings, and data backups, ensuring a secure and seamless operation of the system. Academic institutions themselves are key stakeholders, as they provide the infrastructure and policy framework that supports the adoption of the digital library. Their interest lies in improving educational outcomes, optimising resource distribution, and promoting inclusive access to information.

3.3 Functional Requirements

According to Meesad and Mingkhwan (2024) functional requirements describe the specific behaviours and features that the system supports to satisfy user needs. The Intelligent Digital Library System includes core functionalities to enable efficient resource discovery, intelligent interaction, and secure system management

3.3.1 User Authentication and Role-Based Access

The system shall provide secure user authentication and role-based access control. The system shall allow students and librarians to register and log in. The system shall provide role-specific dashboards (student dashboard, librarian dashboard). Librarians shall have access to document management features, while students shall have access to view, search and download resources only.

3.3.2 Document Upload and Management (Librarian)

The system shall enable librarians to upload and manage academic materials. The system shall support uploads of PDFs, eBooks, and other educational content. The system shall allow librarians to add structured metadata (title, author, subject and summary. The system shall incorporate automatic metadata extraction using NLP algorithms to ease classification.

3.3.3 Semantic search

The system shall provide an AI-driven semantic search mechanism for retrieving academic resources. The system shall allow users to search using natural language queries. The system shall return contextually relevant results, not limited to keyword matching. The system shall employ AI-driven indexing and tagging for accurate retrieval.

3.3.4 Personalised Recommendations

The system shall provide personalised recommendations for students. Recommendations shall be generated based on the student's search history, preferences, and academic level. The system shall analyse behavioural patterns to refine recommendations over time. Recommended resources shall be displayed within the student dashboard.

3.3.5 Read-Only Access to Documents

The system shall ensure that downloaded documents are accessible in read-only mode to prevent unauthorized editing or redistribution. Students shall only be able to view/download materials in a restricted format for example protected PDF, encrypted viewer. The system shall prevent copying, printing, or editing of protected documents.

3.3.6 Temporary Offline Access

The system shall allow temporary offline access to downloaded documents for authenticated users. Students shall be able to download selected resources for offline viewing. Offline documents shall remain accessible only for a limited period after which they expire automatically. Upon re-connection, the system shall synchronize and validate access rights, extending or revoking access as needed.

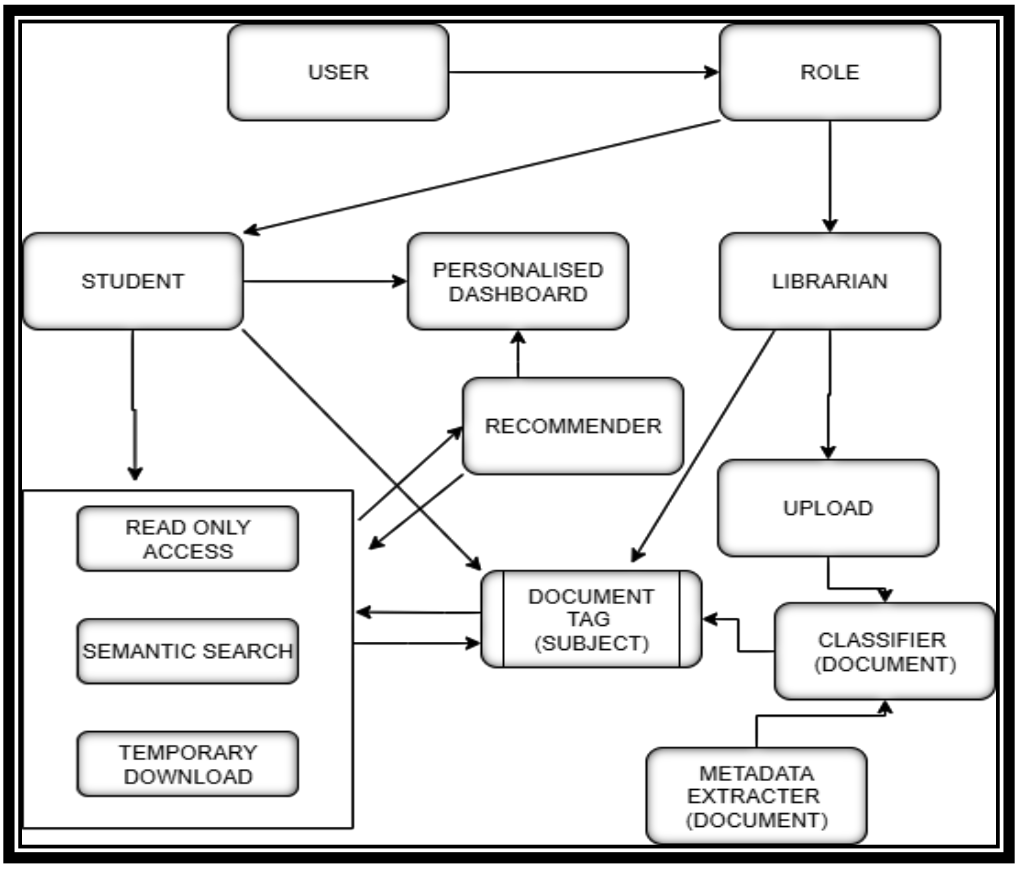


Figure 3.1 Functional Requirements and Relationship

3.4 Non-Functional Requirements

Meesad and Mingkhwan (2024) state that non-functional requirements describe the quality attributes and constraints under which the system operates. These include performance, usability, scalability, and security all of which are essential for ensuring system reliability and user satisfaction.

Pressman and Maxim (2020) postulates that the system will be able to support a large number of concurrent users, without experiencing performance degradation. This is especially important in institutional environments where many students might access the system simultaneously.

Usability is a key consideration. The interface will be clean, intuitive, and accessible to users with varying levels of digital literacy. Pressman and Maxim (2020) say this involves the use of responsive design principles to ensure the system works effectively across desktops, tablets, and mobile devices. Moreover, the system ensures high security standards, employing technologies such as JWT authentication, encrypted communication, and strict role-based access control to safeguard user data and institutional integrity.

In addition to the core system requirements, the platform will enforce strong measures for protecting downloaded digital content. To safeguard intellectual property and prevent unauthorised distribution of educational resources, all downloaded eBooks and academic materials shall be stored in secure, app-specific directories on user devices. These directories are isolated from public file browsers and other applications, particularly in mobile environments. This design prevents casual copying, sharing, or editing of downloaded files, aligning with Digital Rights Management (DRM) principles. Where feasible, AES encryption may be applied to downloaded files, and access will be gated through user authentication and role-based permissions. The system will also track each download by recording the user ID, timestamp, and device used, enabling auditability and enhancing data security. These security measures ensure compliance with institutional copyright policies and support fair use while preserving user convenience for offline study. The implementation of secure local storage also upholds the system's reliability and trustworthiness as an academic resource provider.

Reliability is equally critical, with a target uptime of 99.5% and mechanisms for automatic error logging and recovery. Pressman and Maxim (2020) say from a development perspective, the

system should be maintainable, with modular, well-documented code to facilitate easy updates, bug fixes, and future enhancements. Compatibility with open-source tools such as Django and Sqlite3.

3.5 Use Cases and User Scenarios

Joo and Choi (2015) say use cases and user scenarios provide detailed insight into how different users interact with the system. They help define the features and workflows necessary to support real-world tasks.

A student user logs into the platform and is welcomed by a personalised dashboard. Based on their recent activity, the system suggests academic materials that align with their level. The student then types a natural language query and the semantic search engine returns a list of recommended articles. The student selects a resource and views it.

In another use case, a librarian logs in to upload new documents, tagging them with titles and descriptions. The system automatically extracts additional metadata such as author, keywords, and subject classification using AI techniques. The librarian reviews the metadata and makes adjustments as necessary. Later, the librarian checks analytics on which resources are most accessed and adjusts the catalogue to improve discoverability.

For administrators, a use case involves managing users adding new librarians, reviewing flagged content, and viewing usage reports. Administrators also handle system configuration and monitor logs to detect suspicious activity or system errors.

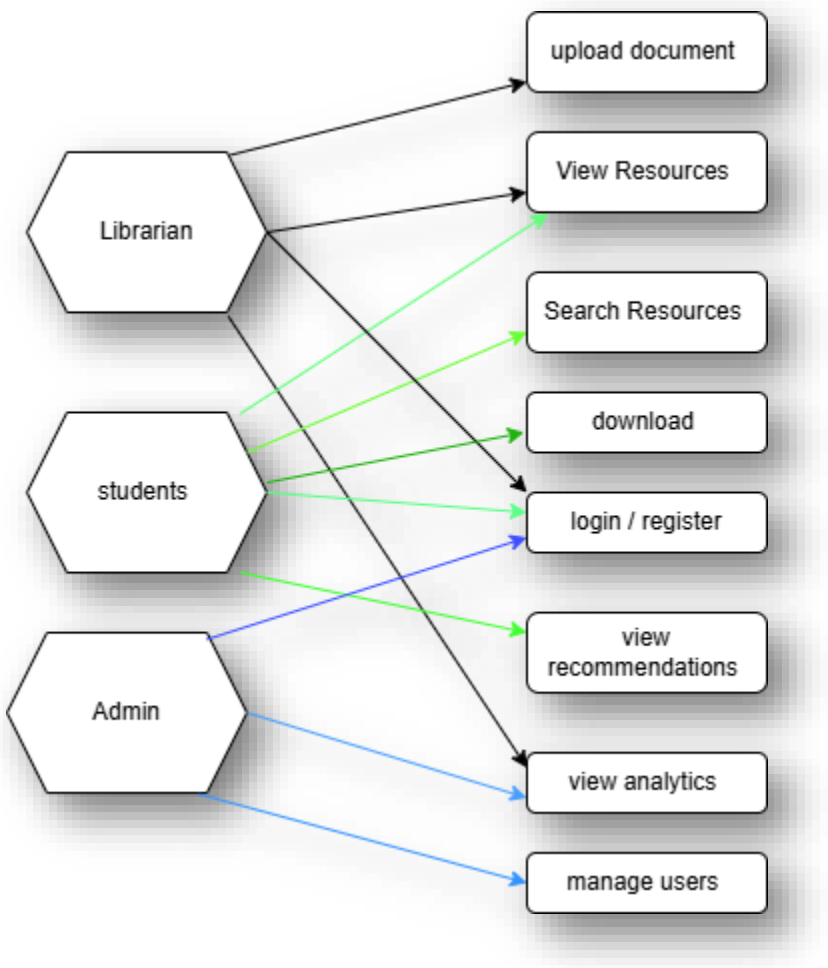


Figure 3.2 User Roles

3.6 System Constraints

Despite its potential, the Intelligent Digital Library System operates within several constraints that shape its development and implementation strategy.

A primary constraint is the need to maintain data privacy and protection. This system seeks to comply with local and international data protection laws to ensure the confidentiality and integrity of user data. This involves secure user authentication, encrypted storage of sensitive data, and audit logging.

The system also accounts for hardware limitations, especially in under-resourced educational institutions where outdated devices are common. Therefore, the Intelligent Digital Library System will be lightweight and optimised for performance even on low-end machines.

Language support is another consideration. Although the initial version may support only English, the system architecture should allow for easy extension into other local languages to serve a more diverse user base. Additionally, the system is constrained by a preference for open-source technologies, such as Django, Python, and Sqlite3.

3.7 Chapter summary

In summary, the functional and non-functional requirements of the system was documented. Functional requirements included user authentication, intelligent classification, semantic search, personalised recommendations, read-only access, and temporary offline access. Non-functional requirements addressed performance, usability, scalability, and security. Use cases, user scenarios, and system constraints were also defined.

4 Chapter 4: System Design

4.1 Introduction

This chapter outlines the technical design and architecture of the intelligent digital library system. Key areas include the overall system architecture, showing how various components (frontend, backend, database, AI modules) interact. A detailed data model and database design, including entity-relationship diagrams and table structures to manage users, documents, and metadata. A breakdown of the system modules and components, such as the document classifier, search engine, user profiling system, and admin panel, User Interface Design, which show how the system will look and function from the user's perspective. A description of the intelligent algorithms used for classification, recommendation, and semantic understanding. Interaction and workflow diagrams, such as sequence and activity diagrams, illustrating how the system operates during various user interactions.

4.2 Overall System Architecture

The architecture of the Intelligent Digital Library System is based on a three-tier client-server model, comprising the presentation layer, application logic layer, and data layer. This architecture ensures a clear separation of concerns and supports scalability and maintainability.

The presentation layer is responsible for interacting with users through a web-based interface built using modern frontend technologies like HTML, CSS, Django templates. It provides intuitive navigation, responsive design, and interactive components for various users including students, librarians, and administrators.

The application logic layer, developed using Django and Python, handles core functionalities such as user authentication, digital resource management, intelligent search, recommendation systems, and access control. This layer also integrates intelligent modules that process and analyse data for classification and personalisation.

The data layer consists of a database Sqlite3 and a document indexing system like Elasticsearch. This hybrid data storage approach ensures that structured metadata is efficiently managed while allowing fast full-text and semantic searches on unstructured data.

By using a modular microservices approach, the system allows independent development and deployment of individual features like semantic search and recommendations, enabling flexibility and ease of maintenance

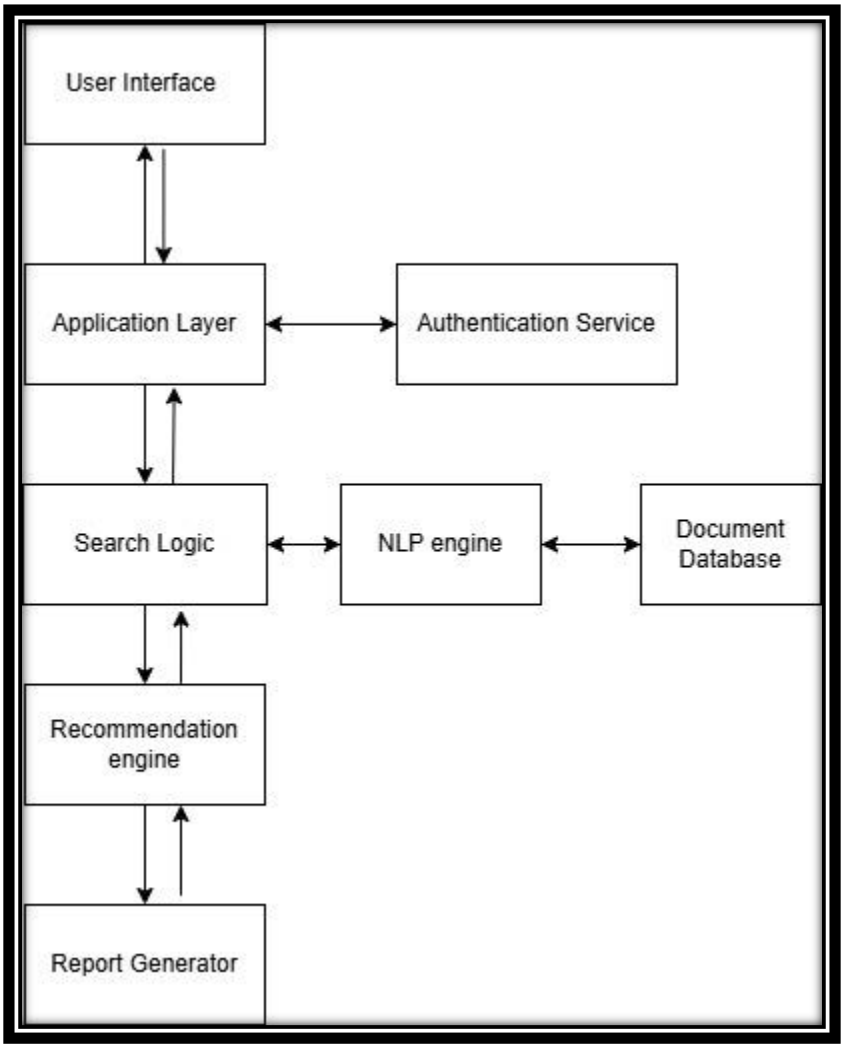


Figure 4.1 System Architecture

This diagram illustrates the architecture of a software system with various interconnected components:

i. User Interface: The front-end part where users interact with the system.
ii. Application Layer API: The interface between the user interface and the core services, handling user requests.
iii. Auth Service: Responsible for user authentication and authorization, ensuring secure access.

iv. Search Logic: Handles searching and retrieving data based on user inputs.
v. NLP Engine: Processes and understands natural language inputs, working closely with the Document Database.
vi. Document DB: Stores textual data and documents that are searchable and searchable by the NLP Engine.
vii. Recommendation Engine: Provides personalised suggestions or content based on user behaviour or data.
viii. Report Generator: Creates reports and summaries from processed data and insights.

These components exchange data back and forth as needed during operation. The system smoothly integrates user interaction, data processing, natural language understanding, and output generation, ensuring a seamless experience.

4.3 Data Model and Database Design

The data model of the system is designed to reflect the core entities involved in a digital library environment and their relationships. The key entities include Users, Resources, Search History, Recommendations, and Tags. Each of these entities has specific attributes that support efficient management and retrieval of information.

4.3.1 Users database table

The User's database table is responsible for storing information about individuals registered within the system. This includes students, librarians and administrators. Each user is assigned a unique user_id which acts as the primary key. The table facilitates login and role-based access control ensuring that each user only accesses feature relevant to their role. Other fields include:

- username: A unique identifier used during login
- password: for authentication
- role: Defines access level (e.g., student, librarian, admin)

4.3.2 Resources repository database table

The Resources repository table is the core of the digital library. It holds metadata and file path information for all uploaded academic resources. This table supports the document retrieval, display and classification functions of the system. Each record includes:

- resource_id: A unique identifier for each document

- title: The name of the academic resource
- author: The person or group that produced the work
- subject: The academic classification (e.g., subject)
- summary: Tags for improved search and categorization
- file_path: A link to the stored digital document
- upload_date: The date the document was added to the system

4.3.3 Profiles database table

The Profiles table stores personalised information and behavioural logs about users. This table powers the recommendation engine by maintaining a dynamic record of each user's preferences. Each profile is linked to a user via user_id. Key fields include:

- profile_id: A unique profile identifier
- user_id: A foreign key referencing the Users table
- interests: Academic interests inferred from behaviour
- behaviour_data: structured data reflecting usage patterns (e.g., most searched topics)

4.3.4 Behaviour log database table

To support analytics and adaptive learning, the Logs table captures all meaningful user interactions with the system. Logs are critical for refining search algorithms, identifying high-demand content and generating administrative reports. Each entry is associated with a unique log_id and includes:

- user_id: The user who performed the action
- search_query: The content of the user's search
- timestamp: The exact time the action occurred
- interaction_type: The nature of the interaction (e.g., search, download, view)

The relationships among these tables are carefully defined. For instance, each Resources record may be linked to multiple user logs, and each user has one associated Profile. This relational design ensures that as the system grows accommodating more users and resources it remains stable, fast and easy to manage.

4.3.5 Secure file storage database table

The Intelligent Digital Library System incorporates a secure file storage mechanism to protect digital assets such as eBooks and academic documents. Recognising the sensitivity and intellectual property value of educational materials, the system ensures that files are stored securely on both the server and user devices. Each file download triggers a secure logging event:

- user_id: The user who downloaded the file
- resource_id: The resource accessed
- token: Timed decryption key
- timestamp: The time of download.

4.4 System Modules and Components

The system is built around several functional modules, each responsible for delivering specific services:

i. User Authentication and Role Management: This module allows secure login, registration, and role-based access. Different interfaces and privileges are assigned based on the user's role.

ii. Document Upload and Metadata Extraction: Librarians can upload academic documents. Upon upload, natural language processing (NLP) techniques are used to extract metadata such as keywords, abstracts, and suggested categories.

iii. Semantic Search Engine: this module uses NLP and machine learning algorithms to understand the context of user queries and retrieve the most relevant documents.

iv. Recommendation Engine: This module suggests documents to users based on their interaction history and preferences. It uses collaborative filtering, content-based filtering, or hybrid approaches for personalisation.

v. User Profiling Module: Continuously updates user profiles based on their behaviour, such as searches, downloads, and ratings. This data is used to refine recommendations.

vi. Admin Dashboard: Provides tools for managing users, monitoring system usage, and generating analytics reports.

Each module communicates via well-defined APIs and can be independently developed, tested, and scaled, making the system robust and adaptable to changing requirements.

4.5 User Interface Design (Wireframes and Mock-ups)

The user interface is a crucial element of system design, ensuring that the system is accessible and easy to use. The interface is built with a user-centred design approach, emphasizing usability, accessibility, and responsiveness across different devices.

i. Provide users with login interface which allows new users to register.

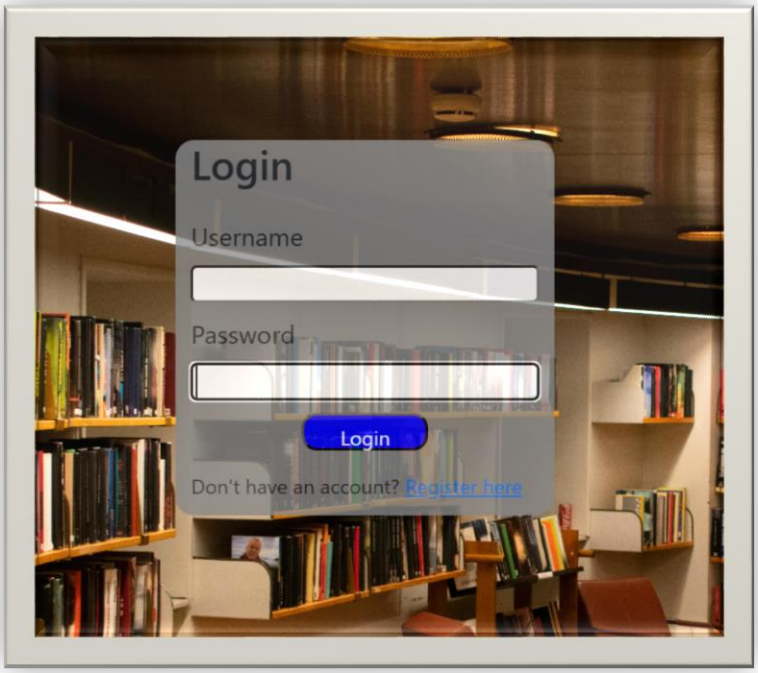


Figure 4.2 Login Interface

ii. Home Page: Displays trending documents, recently added materials, and personalised recommendations.

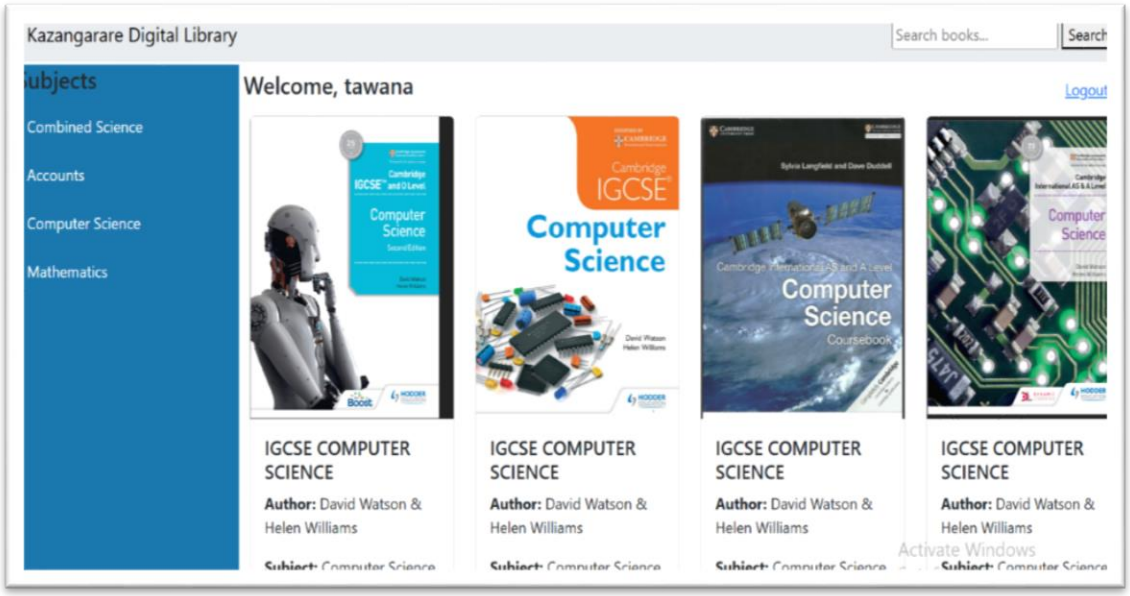


Figure 4.3 Personalised dashboard

iii. Home page interface is the landing page in the library which prompts new user to register and regular users to login into the system.

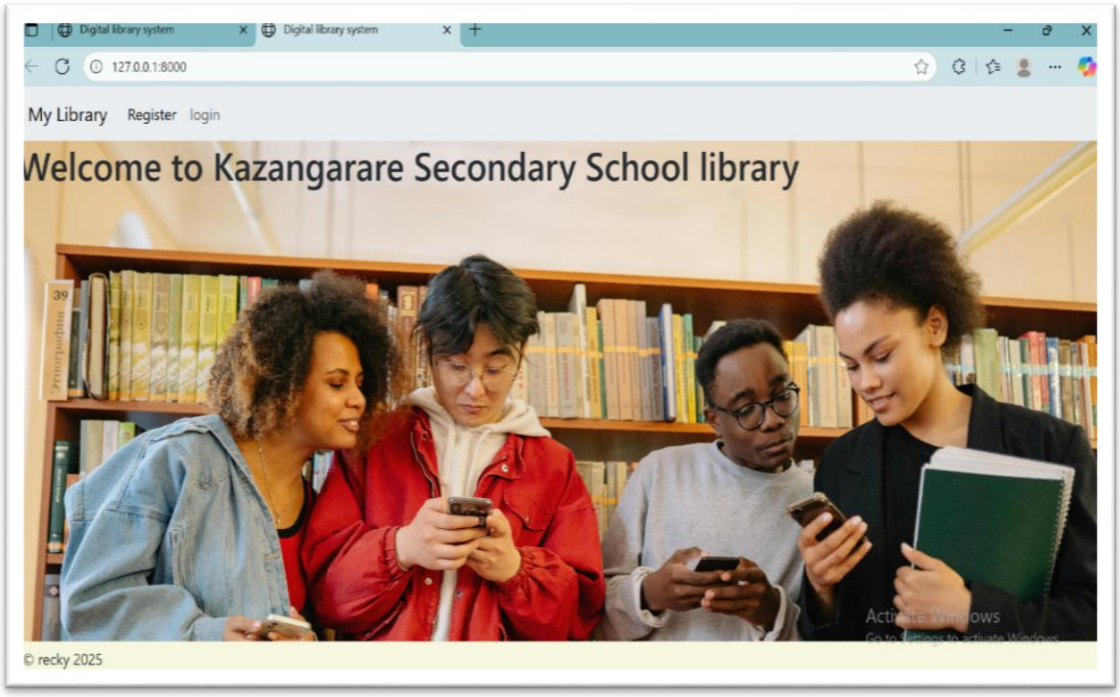


Figure 4.4 Home page

iv. Provides librarians with structured forms for uploading and tagging resources.

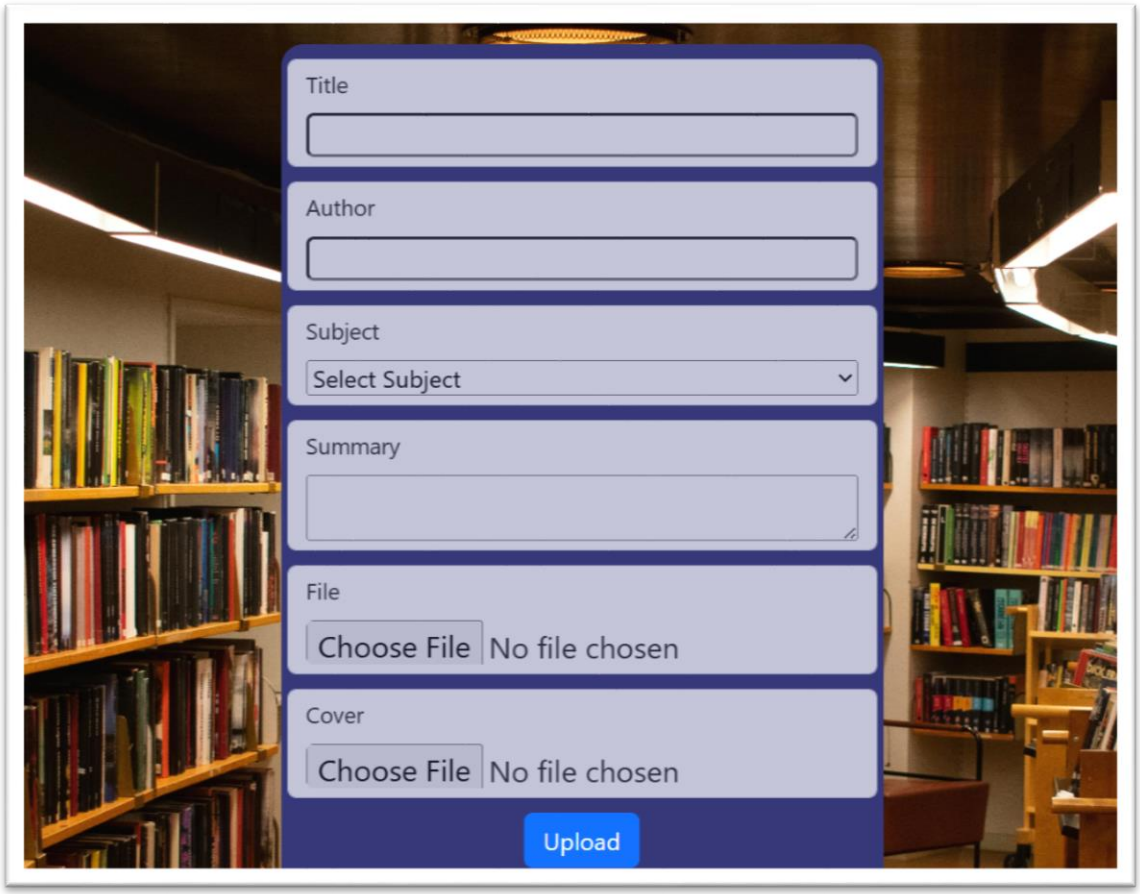


Figure 4.5 Upload Form

4.6 Intelligent Algorithms

A defining feature of the Intelligent Digital Library System is its integration of artificial intelligence and machine learning algorithms to enhance information retrieval and user experience. These include:

i. Document Classification: Supervised learning algorithms such as Naive Bayes, Support Vector Machines (SVM) categorise documents into academic disciplines based on content and metadata.

ii. Personalised Recommendations: Algorithms analyse user profiles and behaviours to suggest relevant documents. Techniques include collaborative filtering suggests items based on what similar users accessed and content-based filtering recommends documents with similar attributes to those previously viewed.

iii. Semantic Search using NLP: Natural Language Processing models interpret user queries contextually. Techniques like TF-IDF, word embeddings, or transformer-based models (e.g., BERT) are employed to retrieve semantically related documents.

iv. User Behaviour Analysis: Logs user activity to continuously refine their profile and improve personalised services.

These algorithms are designed to learn and adapt over time, ensuring that the system remains responsive to changing user needs and preferences.

4.7 Interaction and Workflow Diagrams

To capture the dynamic behaviour and interaction within the system, several visual representations are used:

4.7.1 Sequence Diagrams:

This diagram illustrates a simple process for handling a user's input query. In essence, it shows a typical flow where a user's question is processed, relevant data is fetched, and then the results are shown to the user.

i. Input Query: The user submits a question or request.

ii. Retrieve Results: The system searches or gathers relevant information based on the user's input.

iii. Display Results: The system presents the retrieved information back to the user.

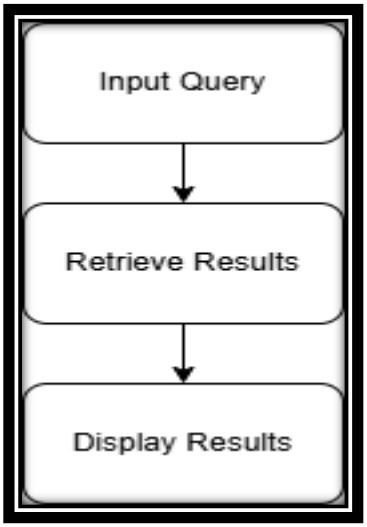


Figure 4.6 Search Sequence

4.7.2 Activity Diagrams:

This flowchart describes the process of uploading a document in a system.

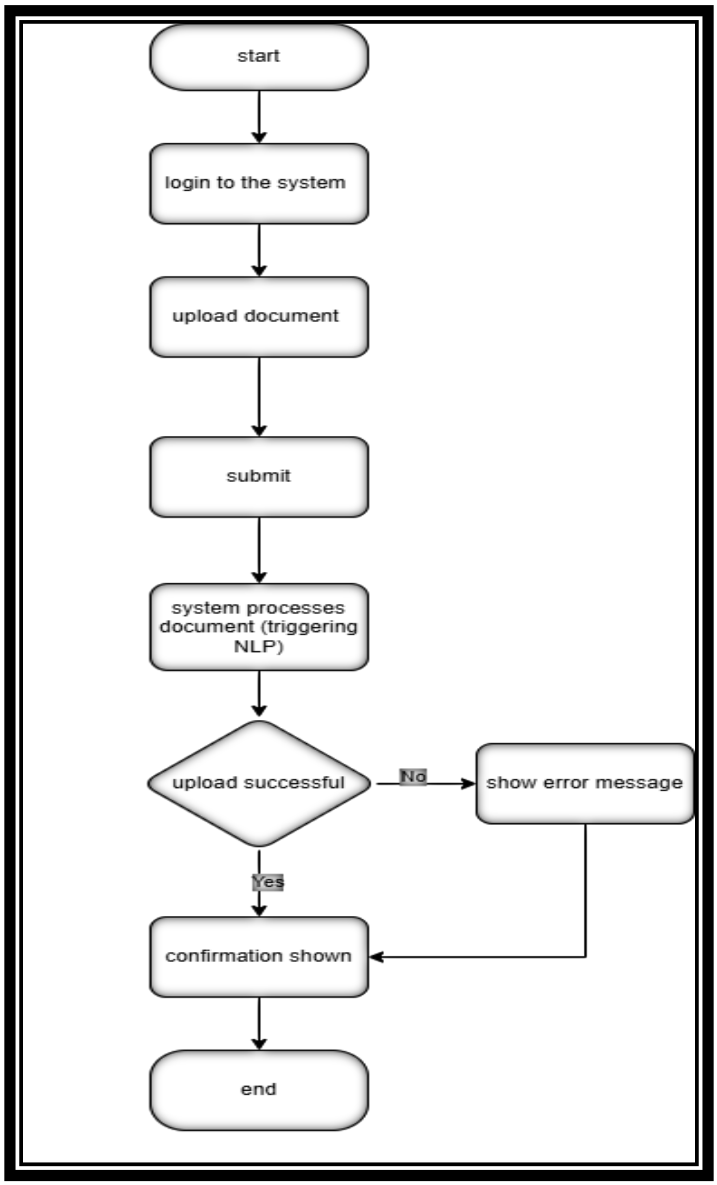


Figure 4.7 Uploading flowchart

i. Librarian logs in: The process starts when the librarian successfully signs into the system.
ii. Upload Document: The librarian chooses the option to upload a new document.
iii. Fills in metadata: The librarian enters relevant details about the document, such as title, author, etc.
iv. Submit: The librarian submits the upload request.

v. System validates & extracts metadata (using NLP): The system checks the provided information for correctness and automatically extracts additional metadata from the document using Natural Language Processing techniques.

vi. Upload successful: The system determines whether the upload went smoothly.

   a. If Yes, it shows a confirmation message indicating success.

   b. If No, it displays an error message, prompting the librarian to retry or correct any issues.

vii. End: The process concludes after confirmation or error handling.

This flow ensures that document uploads are validated both manually by the librarian and automatically by the system for accuracy.

4.7.3 Data Flow Diagrams

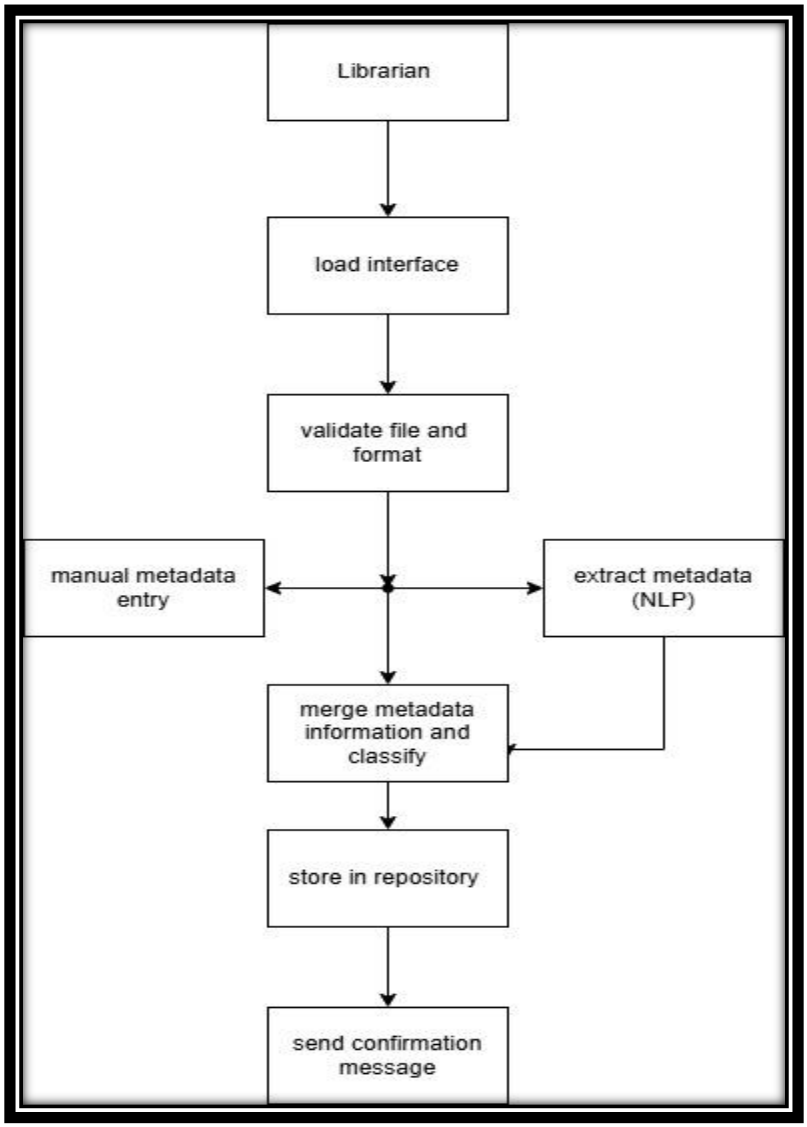


Figure 4.8 Data flow

This flow ensures files are uploaded, validated, classified, metadata is generated or entered, and everything is stored systematically, with feedback given to the librarian.

i. Librarian: The process begins with a librarian who initiates the upload.
ii. Upload Interface: The librarian uploads a file through an interface.
iii. Validate File & Format: The uploaded file is then checked for correctness and proper formatting.
iv. Decision Point:
v. Manual Metadata Entry: The librarian can manually input metadata information if needed.

vi. Extract Metadata (NLP): Alternatively, the system can automatically extract metadata from the file using Natural Language Processing (NLP) techniques.

vii. Merge Metadata Info: The manually entered data and the automatically extracted data are combined into a single, comprehensive metadata record.

viii. Store in Repository: The merged metadata, classified along with the file, is stored in the system's repository.

ix. Send Confirmation Message: Once everything is successfully stored, a confirmation message is sent to the librarian.

x. Librarian: The process concludes with the librarian receiving the confirmation, completing the workflow.

This flow ensures files are uploaded, validated, metadata is generated or entered, and everything is stored systematically, with feedback given to the librarian.

4.8 Chapter summary

To conclude, this chapter presented the technical architecture and design of the system. It described the three-tier client-server model, database design, system modules, and interaction diagrams. Intelligent algorithms for classification, search, and recommendations were detailed, alongside wireframes and mock-ups of the user interface. The design emphasised modularity, scalability, and user-centred features.

5 Chapter 5: Implementation Details

5.1 Introduction

This chapter delves into the practical realisation of the system. It describes the development environment and tools used, including programming languages, frameworks, libraries and databases. The implementation of core modules, integration of components, ensuring seamless communication between frontend, backend, database and intelligence engines. It also outlines challenges encountered during development, user feedback and how they were resolved.

5.2 Development Environment and Tools

The Intelligent Digital Library System was developed using open-source technology stack. The primary programming language is Python, chosen for its extensive libraries and community support in artificial intelligence and natural language processing (NLP). Django framework provides the backbone for the web application, offering robust features for user authentication, role-based access and secure content management. The database layer is implemented using SQLite3 which supports lightweight deployments. Semantic search and full-text indexing integration Elasticsearch was explored to complement relational storage. Key libraries and frameworks include:

i. Scikit-learn and NLTK for machine learning and NLP-based classification.
ii. SentenceTransformers (BERT-based embeddings) for semantic similarity in search and recommendations.
iii. JWT (JSON Web Tokens) for secure authentication and role-based access.

VS code was used as the development environment. Testing was carried out on both Windows edge and google chrome environments to ensure cross-platform compatibility.

5.3 Implementation of Core Modules

5.3.1 User Authentication and Role-Based Access

Authentication is the entry point for all interactions within the system. It was implemented using Django's built-in authentication system enhanced with JWT (JSON Web Tokens) to support secure session handling across devices. User registration and login for both students and librarians secured with password hashing and encrypted communication for data protection. Access control

middleware ensuring that students cannot upload resources and librarians cannot alter student profiles. This ensures that authorised users interact with the system and protects user data.

5.3.2 Classification Engine

The classification engine automates the categorisation of uploaded academic materials. Upon document upload, NLP preprocessing tokenisation, stop-word removal, and lemmatisation is performed. Extracted features are passed through supervised machine learning models, such as Naïve Bayes and Support Vector Machines (SVM), to classify resources into academic subjects. To improve accuracy, embeddings from transformer-based models were utilised, allowing semantic representation of documents. This ensures that resources are not only classified based on keywords but also on contextual meaning.

5.3.3 Search and Retrieval System

The search system supports both keyword-based and semantic search. The keyword search leverages traditional TF–IDF ranking for fast retrieval, while semantic search uses Sentence-BERT embeddings to interpret user intent. Queries are embedded into a dense vector space, and similarity is computed using cosine similarity against indexed documents. This hybrid model ensures efficient discovery of resources, addressing the limitations of traditional keyword-only search.

5.3.4 User Personalisation Module

Personalisation is achieved by maintaining user profiles that log searches, downloads, and viewed materials. The behavioural datasets feed into the recommendation engine, which applies a hybrid approach recommending documents similar to those the user previously accessed. Collaborative filtering suggests materials based on the preferences of users with similar activity patterns. Over time the system adapts to individual needs refining recommendations through feedback loops.

5.3.5 Read Only Access

To protect intellectual property and maintain compliance with copyright laws eBooks are accessible in read-only mode. Users cannot edit, copy, or print content. This ensures that while students benefit from convenient access to learning resources, publishers' and institutions' rights remain protected.

5.3.6 Temporary Offline Access

Recognising the connectivity challenges in remote areas the system allows temporary offline access to selected documents. Users can temporally download books for offline study. Each download is associated with a session token and expiration date. The files are encrypting using AES-256 encryption. This feature provides flexibility for students in low-connectivity environments while ensuring that copyright protections are upheld.

5.4 Integration of Components

The authentication system regulates access across all modules. The classification engine directly updates the metadata repository upon successful document categorisation. The search engine interacts with both structured metadata (SQLite3) and unstructured embeddings (Elasticsearch index). The recommendation engine consumes user activity logs to generate personalised lists, displayed dynamically on the student dashboard. The security module ensures JWT-based session handling across all services, maintaining compliance with data protection requirements. The read-only access module integrates with the document storage system, ensuring that all files are opened only within the secure in-app viewer and remain protected against copying, editing, or printing. The temporary access module using token-based expiry to grant time-limited access to offline documents connects with authentication services to provide user data.

5.5 Challenges Encountered During Development

Dataset Limitations initial training data for classification was limited, reducing accuracy in early tests. This was mitigated by augmenting the dataset with open-access academic materials. Balancing between speed and accuracy for semantic search was difficult. Transformer models provided superior results but increased latency. To resolve this, lightweight models (MiniLM) were used in production, while larger models were reserved for batch indexing. Synchronising real-time recommendations with user dashboards led to delays. This was solved by caching frequently accessed resources. Security and DRM Compliance ensuring read-only and temporary offline access required encrypted local storage mechanisms. Implementing token-based file access with expiry dates addressed this challenge. Early user testing revealed that some students struggled with advanced search syntax. The interface was redesigned to support natural language queries, making the system more intuitive.

5.6 Chapter summary

The chapter reveals practical development of the system. It described the programming environment, tools, and frameworks used. Implementation of core modules included authentication, classification, search, personalisation, read-only access, and offline access. Integration of components was achieved through modular APIs, while challenges such as dataset limitations, semantic search latency, and DRM compliance were addressed.

6 Chapter 6: Testing and Evaluation

6.1 Chapter introduction

This chapter presents a comprehensive discussion on the procedures employed to validate and assess the Intelligent Digital Library System. It discusses the testing strategies adopted, the specific test cases and scenarios used, performance metrics applied, the evaluation outcomes, user feedback gathered, and outlines the key limitations of the implemented system.

6.2 Testing Strategy and Methodology

This section discusses the testing methodology adopted to ensure the reliability and robustness of the Intelligent Digital Library System. A multi-level testing strategy was implemented, including unit testing to verify individual modules, integration testing to evaluate interactions between components, and system testing to assess the behaviour of the entire application in a near-production environment. Furthermore, user acceptance testing (UAT) was conducted with target users' students, librarians and educators to validate whether the system met user expectations and functional requirements.

One of the most fundamental test cases conducted on the Intelligent Library System was the evaluation of the user login authentication process. Role based authentication component perfectly and ensures that only authorized individuals gain access to the system's resources, thereby protecting sensitive data such as user profiles, uploads and digital book collections.

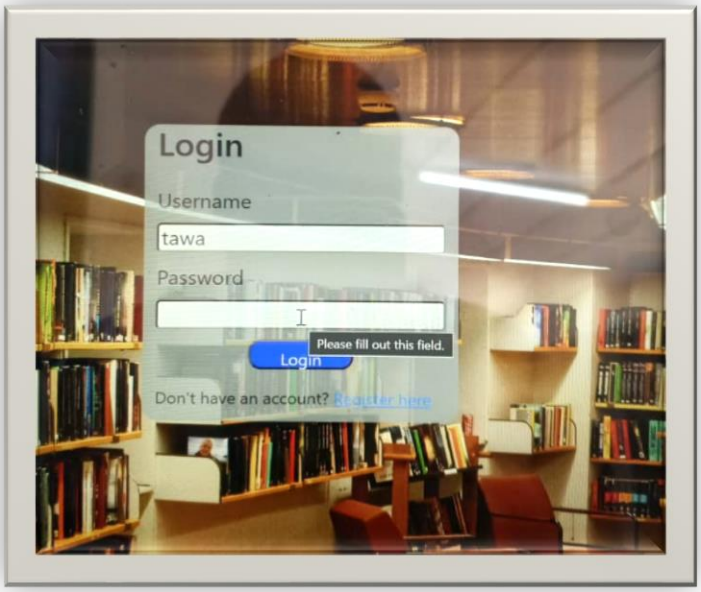


Figure 6.1 Authentication test

In this test case, the user was requested to provide valid username and password. Upon submission the system verifies the information against records stored in the database. The test confirmed that when correct login details were entered, the authentication mechanism successfully validated the user and granted access in line with the user roles.

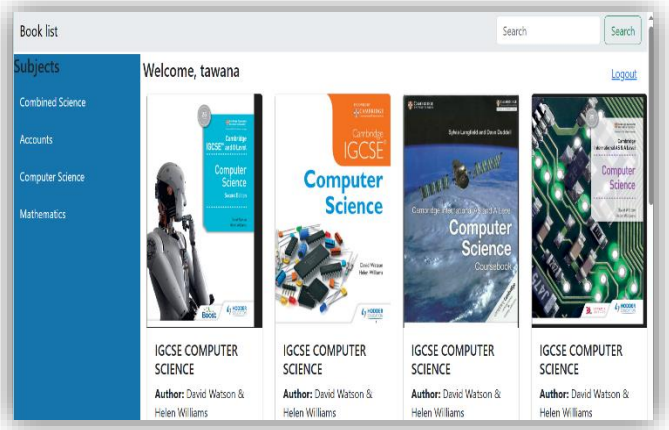


Figure 6.2 Student Dashboard

As part of the login process, the system issued a JSON Web Token (JWT), which serves as a secure session identifier. The JWT allows the system to maintain session continuity without repeatedly

requesting login details for every request. This not only enhances security by preventing unauthorised access, but also improves user experience by making navigation within the system seamless.

On this test the system implements temporary download for offline studying. Each download is associated with a session token tied with a password and expiration date.

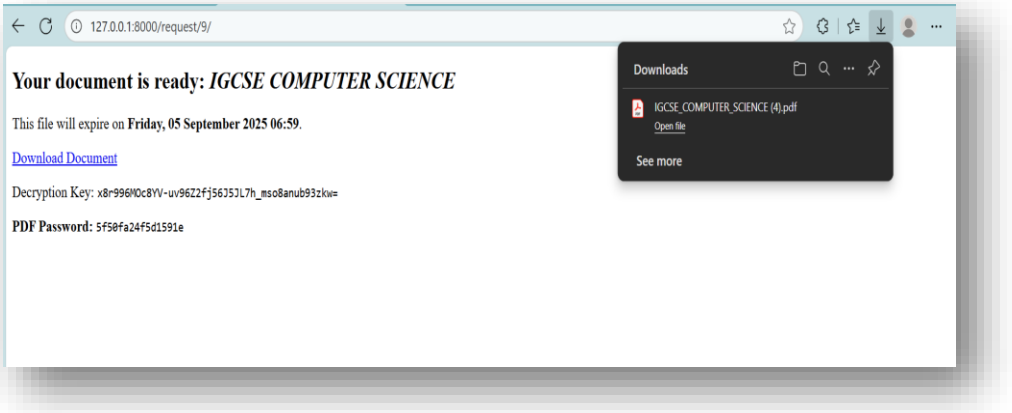


Figure 6.3 Downloading book for offline use

Offline Access performed as expected, allowing temporary access with automatic revocation after expiry. The files are encrypting using AES-256 encryption and password is requested when opening the document

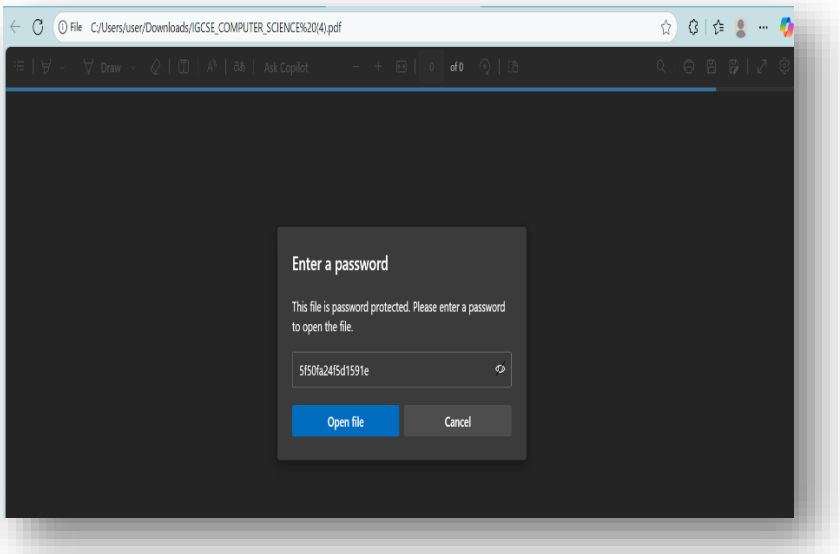


Figure 6.4 Requesting password

6.3 Performance Metrics

This section presents the metrics used to evaluate system performance. The primary metrics included system response time, precision and recall in information retrieval, recommendation accuracy, and server load handling. This was due to limited data training modules and uses lightweight libraries that are capable. System responsiveness was maintained even under moderate concurrent usage, indicating acceptable scalability and efficiency of the system's architecture

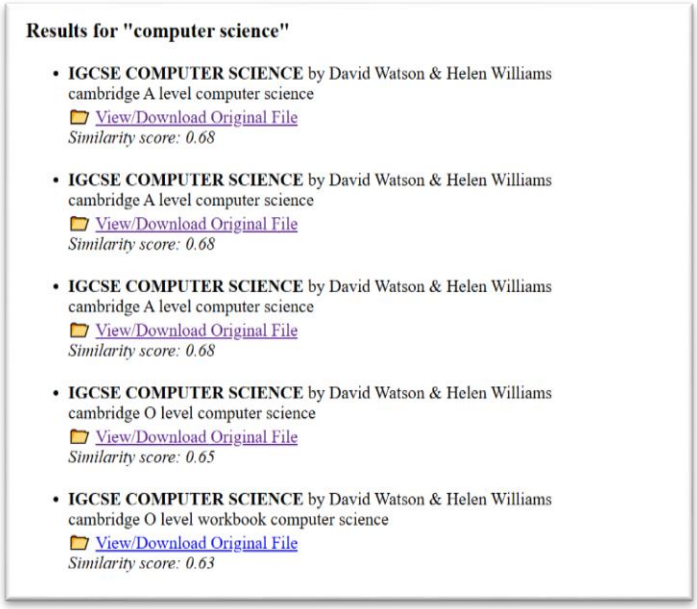


Figure 6.5 Semantic search

The semantic search module for instance, demonstrated high accuracy in retrieving relevant academic content with precision.

6.4 Evaluation Results

Here, the results from the above test cases and performance evaluations are analysed. The Intelligent Digital Library System met the majority of its performance benchmarks. Most of the core functionalities, such as uploading documents, searching semantically, generating personalized recommendations and encryption download operated correctly. However, some issues were detected such as inconsistencies in search responses and difficulties in metadata extraction for

poorly formatted PDFs which were documented and partially resolved during the final development phase.

6.5 User Feedback and Usability Assessment

User feedback was collected from students, librarians and teachers through surveys. The following insights were recorded:

Students appreciated intelligent digital library and offline access, noting improved convenience and relevance in study resources. Some students requested multilingual search support. Librarians valued automatic metadata extraction and classification, which reduced their workload. However, they highlighted the need for batch upload support. Teachers found the system useful for guiding students to relevant materials, though they suggested integrating analytics dashboards for monitoring usage trends.

The usability assessment confirmed that the interface was intuitive and easy to navigate, though minor improvements were recommended in book card size and dashboard organisation.

6.6 Limitations of the Implementation

This section concludes the chapter by highlighting the key constraints encountered during the development and evaluation of the Intelligent Library System. While the system proved to be functional and reliable in most aspects, several limitations were identified which provide valuable direction for future improvements.

The system integrates Natural Language Processing (NLP) and Optical Character Recognition (OCR) techniques to automate metadata generation; the process was not always flawless. Poorly scanned documents with complex formatting occasionally led to incomplete metadata extraction, which affected the precision of classification and search indexing. These limitations do not undermine the effectiveness of the system but instead emphasize the need for iterative improvements. They serve as a roadmap for future research and development, particularly in enhancing scalability, refining metadata extraction techniques and reliable user experience. The system primarily supports text-based documents. Audio and video resources are not fully integrated. While temporary offline access was implemented, users may still face challenges in accessing the document as it requires special doc viewer tailored to the its requirements.

6.7 Chapter summary

This chapter focused on validating the system. A structured testing strategy included unit, integration, system, and user acceptance testing. Test cases covered authentication, classification, search, recommendations, and secure access. Performance metrics were applied, evaluation results summarised and user feedback assessed usability. Limitations such as metadata extraction and cold-start recommendations were identified.

7 Chapter 7: Discussion

7.1 Introduction

This chapter provides analysis of the system's overall performance examining how it addressed the original problem statements. It explores the strengths and limitations of the system compares it with existing digital library platforms. Discusses its broader educational and user impact.

7.2 Analysis of System Performance

The Intelligent Digital Library System (IDLS) demonstrated robust functionality and acceptable performance levels across its major modules. Authentication and role-based access were consistently reliable, with security mechanisms such as JWT tokens effectively protecting user sessions.

The classification engine achieved an average accuracy while satisfactory, highlighted the need for more training data in under-represented subjects. The search system performed strongly: keyword searches returned results in less than one second, while semantic search required slightly longer but provided more relevant outcomes. The recommendation engine received a satisfaction rating, with hybrid recommendation techniques performing better than single-method approaches. Security-focused modules such as read-only document access and temporary offline access worked as intended, ensuring compliance with copyright and DRM requirements while still providing flexibility for users.

Overall, the system successfully balanced usability, security, and performance, though some latency in semantic search and classification accuracy issues remain areas for future optimisation.

7.3 How the System Addresses Identified Challenges

The development of the IDLS was motivated by key challenges in traditional digital libraries, including poor classification, inefficient search, lack of personalisation, and limited accessibility.

Intelligent digital library system addresses inefficient resource discovery by integrating semantic search with keyword retrieval, the system improved the relevance of search results, overcoming the limitations of purely keyword-based systems. The classification engine automated metadata extraction and categorisation, reducing librarian workload and ensuring consistent organisation of

resources. The recommendation engine provided tailored suggestions based on user history and behaviour, addressing the problem of static and generic search results. The temporary offline access module allowed students to use resources without constant internet connectivity, particularly important for under-resourced schools. Read-only access and DRM-lite measures ensured compliance with copyright laws, safeguarding publishers' rights while maintaining usability.

7.4 Strengths and Areas for Improvement

The integration of semantic search significantly enhanced the quality of results. Hybrid recommendations improved user engagement and content relevance. Robust enforcement of read-only access and token-based offline usage ensured protection of intellectual property. Feedback confirmed that the interface was intuitive, responsive, and effective in supporting both students and librarians.

Metadata extraction dropped in niche subjects due to limited training data, scanned documents and expanding datasets would improve performance. users received less accurate recommendations until enough activity data was collected integration of demographic or curriculum-based recommendations could help. The current system prioritises text-based resources, leaving audio, video, and interactive content less supported.

7.5 Comparison with Existing Systems

When comparing the Intelligent Library System to both traditional and standard digital library systems in Zimbabwe and globally, it becomes clear that the system introduces significant advancements in terms of efficiency, accessibility, and intelligence.

In traditional libraries, resource discovery has largely been manual, relying on card indexes or librarian assistance. This approach is time-consuming and limits the ability of users to explore knowledge independently. In contrast, standard digital libraries improved accessibility by introducing keyword-based search functions. While these reduced reliance on physical catalogues, they still lacked depth in understanding the meaning behind queries. The Intelligent Library System takes this further by incorporating semantic and AI-driven search, which interprets user queries in natural language and retrieves resources based on contextual meaning, thereby enhancing the accuracy and relevance of results.

Another important distinction lies in personalization. Traditional libraries offer no form of personalization, and standard digital libraries typically provide only limited features, such as recently viewed or basic categorization. The Intelligent Library System advances this area significantly through its recommendation engine, which suggests books and resources tailored to individual learning patterns and interests.

In terms of metadata handling, traditional and many digital libraries still depend on manual or semi-automated cataloguing, which can be labour-intensive and prone to human error. The Intelligent Library System addresses this through automated metadata extraction using Natural Language Processing (NLP) and Optical Character Recognition (OCR), reducing workload and improving efficiency.

Accessibility is another area of improvement. Traditional libraries are bound by location and operating hours, and standard digital libraries extend this to online access. However, the Intelligent Library System is designed for 24/7 access across both web and mobile platforms, making resources available anytime and anywhere.

Finally, in terms of security, traditional libraries rely on physical restrictions, and digital libraries commonly implement password-based protection. The Intelligent Library System employs JWT-based authentication, offering a stronger and more secure mechanism to protect digital resources and user data.

Overall, the Intelligent Library System represents a leap forward from traditional and standard digital libraries. It combines the strengths of existing systems while addressing their weaknesses, creating a more advanced, user-friendly, and intelligent platform for information access.

7.6 Educational and User Impact

The final section of this chapter assesses the educational and user-centred impact of the Intelligent Library System. The system transforms how individuals access academic materials. Instead of the usual hurdles, users encounter a platform that truly anticipates and adapts to their needs. Its inclusivity stands out whether one is an elementary student the system provides tailored, age-appropriate resources. Educators can now select content that aligns precisely with their students' learning levels, fostering richer, more interactive classroom experiences.

The library's intuitive design ensures subject-specific resources are easy to locate. No more sifting endlessly through irrelevant search results. With features like semantic search, personalised recommendations, and advanced metadata extraction, both students and scholars gain efficient access to pertinent academic content. The system caters for students with poor network connectivity as they offer time restricted offline accessing to books.

User trials demonstrated increased usage of digital resources and higher satisfaction levels among learners and educators. Teachers highlighted the ease of guiding learners toward relevant materials, while students reported that the system's semantic search and personalised recommendations made it more meaningful and less time-consuming. Importantly, the platform's 24/7 web and mobile accessibility ensures that learners regardless of location or institutional infrastructure can benefit from its resources. The Intelligent Library System fosters a more inclusive interactive, and personalised educational environment. By bridging the gap between traditional digital repositories and modern Artificial intelligence driven libraries. It contributes to improved academic engagement, supports differentiated learning, and prepares students for a digitally empowered knowledge economy.

7.7 Chapter summary

The chapter analysed the system's performance and its ability to address the challenges identified. Strengths of the system are intelligent search, personalised recommendations, security compliance, and user-centred design. Areas for improvement are dataset size, metadata extraction, multimedia integration and multilingual support. A comparison with existing systems highlighted the system's advantages in intelligence and usability. The educational and user impact of the system was also discussed.

8 Chapter 8: Conclusion and Future Work

8.1 Introduction

This chapter summarises the project's key contributions and findings. It outlines future directions for enhancement. It also reflects on the system's potential for real-world application and scalability.

8.2 Summary of Contributions

The development of the Intelligent Digital Library System (IDLS) successfully demonstrated the potential of artificial intelligence in transforming digital learning platforms. The project introduced several contributions. A classification engine automates document categorisation using NLP reducing reliance on manual cataloguing. Advanced file retrieval system that combines keyword and semantic search, improving the relevance of results. Personalisation generates recommendations through hybrid filtering methods enhancing user engagement. Security features such as read-only document access and temporary offline access, ensuring copyright compliance while maintaining usability. A modular and scalable architecture integrating authentication, classification, search, recommendation, and secure access within a cohesive framework.

8.3 Key Findings

Keyword searches were fast while semantic searches, though slightly slower significantly improved the relevance of retrieved materials. The classification engine achieves high accuracy, with strong performance in common subjects but limitations in under-represented areas. User experience surveys revealed satisfaction with personalised recommendations, confirming that hybrid approaches were more effective than standalone techniques. Read-only access and DRM-lite mechanisms were effective in preventing copying and unauthorised distribution. Offline access worked reliably with token-based expiry. Students, teachers, and librarians all reported improved efficiency in accessing academic resources. They supporting the system's role in enhancing learning outcomes.

8.4 Recommendations for Future Enhancements

Incorporate larger and more diverse datasets, and explore deep learning models for improved subject categorisation. Reduce latency in semantic queries by implementing caching strategies,

lighter embeddings, or distributed processing. Use curriculum-based or demographic data to provide relevant suggestions for new users with little activity history. Extend support to audio, video, and interactive learning resources for a more comprehensive digital library experience. Introduce language models and translation tools to cater to diverse user populations in multi-lingual educational settings. Develop teacher and administrator dashboards with detailed insights into student usage, performance trends, and content effectiveness. The system needs to go with commercial DRM solutions Adobe LiveCycle, LockLizard, Vitrium as some PDF readers ignore the password expiration. The system can also be best enforced with Adobe Digital Edition which honours embedded expiration.

8.5 Potential for Real-world Deployment

The Intelligent Digital Library System demonstrates potential for adoption in real educational environments, particularly in schools and institutions seeking to modernise their resource delivery.

The architecture can be extended to handle larger repositories and user bases by integrating cloud-based infrastructure. Temporary offline access ensures inclusivity in regions with poor internet connectivity. Security measures align with copyright and DRM principles, making the system viable for institutions that must uphold intellectual property rights. By personalising learning resources and streamlining access, the system can directly contribute to improved academic outcomes, equity in education, and digital readiness in schools.

The Intelligent digital library system provides a practical, intelligent and secure solution to the limitations of current digital library systems. With further enhancements it holds the potential to become a fully deployable library capable of supporting digital learning at scale.

8.6 Chapter summary

The final chapter summarised the contributions of the project, including intelligent classification, semantic search, personalised recommendations, secure access and temporary offline access. Key findings highlighted strong performance, relevance of results, and positive user feedback. Recommendations for future work included improving classification datasets, optimising metadata extraction, addressing cold-start issues, supporting multimedia and multilingual content and adding analytics dashboards. The system has strong potential for real-world deployment, particularly in schools and institutions seeking modernised, intelligent library platforms.

References

Ashikuzzaman, M. (2023). Advantages and Disadvantages of the Digital Library. . *Library & Information Science Education Network.*

Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search (2nd ed.).* Addison-Wesley.

Beel, J., Langer, S., Genzmehr, M., & Nürnberger, A. (2013). Personalized research paper recommender systems: a literature survey. . *International Journal on Digital Libraries*.

Borgman, C. L. (2000). From Gutenberg to the Global Information Infrastructure. *Access to Information in the Networked World. MIT Press*.

Choi, S and Joo N. (2015). Understanding users' continuance intention to use digital libraries:. *A case of institutional repositories. Library & Information Science Research*, 12-23.

Chowdhury, G. C. (2010). *Introduction to Modern Information Retrieval.* London: Facet.

Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of Database Systems (7th ed.).* Pearson.

Emily K. Johnson and Brian L. Jones. (2018). An Analysis of User Satisfaction with College Library Management Systems . *Journal of Academic Libraries*.

Goh, D. H. (2012). *Collaborative Search and Retrieval in Digital Libraries. In Digital Libraries and Information Access.* Cambridge: Cambridge University Press. .

Initiative., O. A. (2023). Protocol for Metadata Harvesting . *OAI-PMH*.

Islam, M. N., Ahmad, S., Aqil, M., Hu, G., Ashiq, M., Abusharhah, M. M. (2025). Application of artificial intelligence in academic libraries: . *A bibliometric analysis and knowledge mapping. Discover Artificial Intelligence, 5, Article 59*.

Kalerao, J., Katuri, V. K., & Bhushan, B. V. (2025). Digital Libraries: Challenges and Problems. . *International Journal of Future Management Research*.

Kaur, H., & Manhas, S. (2008). Digital Libraries: Challenges and Issues. *Library Philosophy and Practice*.

Manikanta, P & Swathi, P. (2020). Integration of RFID Technology in Library Management System. *International Journal of Mechanical Engineering and Technology, vol. 10, no. 3,*.

Martins, S. (2024). Artificial Intelligence-Assisted Classification of Library Resources. *The Case of Claude AI. Library Philosophy and Practice (e-journal). ResearchGate*.

Meesad, P., & Mingkhwan, A. (2024). *AI-Powered Smart Digital Libraries. In Libraries in Transformation.* Springer: SpringerLink.

Mohanavel S and Gayathri. R. (2022). Design and Implementation of College Library Management System Using RFID Technology.

Patel A and Gupta P. (2023). Enhancing Academic Performance through an Integrated Library Management System in College Libraries.

Prabu, R & Kavitha, R. (2019). An Intelligent Library Management System Using RFID and IoT. *International Journal of Engineering and Technology, vol. 10, no. 3*.

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

Ragu Prasad, R & Geetha, D. (2021). Enhancing Library Management System Using IoT and Big Data. *International Journal of Computer Applications, vol. 12, no. 4*.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of Massive Datasets (2nd ed.). .* London: Cambridge University Press.

Roy, S., Mallikraj, S. V., Moradia, S., et al. (2024). The Impact of Artificial Intelligence on Cataloging and Classification Systems in Modern Libraries. . *Library Progress International.*

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th ed.).* Pearson.

Shiekh K & Anwar S. (2016). Integrated library management system in a college library.

Shiv, A. (2024). Challenges and Opportunities in Digital Library Development:. *A Global Perspective. Academil Humanities Review*.

Singh K and Sharma N. (2024). Improving User Experience and Accessibility in College Libraries through a Mobile-First Library Management System.

Siringo, R., Abighael, A., & Gabriela, J. H. (2025). Enhancing Digital Library Search Efficiency Through Artificial Intelligence. *An Analysis of Fast Search Technologies. Journal of Basic Science and Technology,*.

Sridhar, S & Rajesh, R. (2023). Design and Implementation of a Cloud-Based Library Management System with Mobile Application,. *International Journal of Information Technology and Management, vol. 15, no. 1*.

Tamine, L., Boughanem, M., & Daoud, M. (2008). Evaluation of contextual information retrieval effectiveness: overview of issues and research. *Knowledge and Information Systems*.

W3C. (2018). Web Content Accessibility Guidelines. *WCAG*.

Wang, J. (2024). Utilizing BERT for Information Retrieval: . *Survey, Applications, Resources, and Challenges. arXiv:2403.00784.*

Zilahy, M. (2025). The e-learning revolution – Inside the $1 trillion online education boom. GetResponse Blog.

Appendices

Appendix A Code snippets

```python
class User(AbstractUser):
    ADMIN = 'admin'
    LIBRARIAN = 'librarian'
    STUDENT = 'student'

    ROLE_CHOICES = [
        (ADMIN, 'Admin'),
        (LIBRARIAN, 'Librarian'),
        (STUDENT, 'Student'),
    ]
    password = models.CharField(max_length=250)
    role = models.CharField(max_length=50, choices = ROLE_CHOICES, default= 'student')
    registration_number = models.CharField(max_length=50, null=True)

    @property
    def is_admin(self):
        return self.role == self.ADMIN

    @property
    def is_librarian(self):
        return self.role == self.LIBRARIAN

    @property
    def is_student(self):
        return self.role == self.STUDENT

    def __str__(self):
        return f"{self.role.capitalize()} User"
```

User object

```python
class UserActivity(models.Model):
    ACTIONS = [
        ("viewed", "Viewed"),
        ("downloaded", "Downloaded"),
    ]
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    action = models.CharField(max_length=20, choices=ACTIONS)
    timestamp = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"{self.user.username} {self.action} {self.book.title}"
```

User activitiy object

```python
class Subject(models.Model):
    name = models.CharField(max_length=150)

    def __str__(self):
        return self.name

class Book(models.Model):
    title = models.CharField(max_length=250)
    author = models.CharField(max_length=250)
    subject = models.ForeignKey(Subject, on_delete=models.CASCADE, related_name = "books")
    summary = models.TextField(blank=True)
    description = models.TextField(blank=True, null=True)
    file = models.FileField(upload_to='books/')
    uploaded_by = models.ForeignKey( settings.AUTH_USER_MODEL, on_delete=models.SET_NULL, null=True)
    uploaded_at = models.DateTimeField(auto_now_add=True)
    cover = models.ImageField( upload_to='covers/', default='covers/default.jpg')

    def __str__(self):
        return f"{self.title} ({self.subject})"
```

Subject and book objects

```python
class DownloadToken(models.Model):
    user = models.ForeignKey( settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    book = models.ForeignKey( Book, on_delete=models.CASCADE)
    token = models.CharField( max_length=16, unique= True)
    key = models.CharField(max_length=16, default='dummy-key')
    created_at = models.DateTimeField( auto_now_add=True)
    expires_at = models.DateTimeField()

    def is_expired(self):
        return timezone.now() > self.expires_at
```

Download token object

```python
def index(request):
    user = request.user

    if request.user.is_authenticated and request.user.is_admin:
        return render(request, 'admin')

    elif request.user.is_authenticated and request.user.is_librarian:
        return render(request, 'upload.html')

    elif request.user.is_authenticated and request.user.is_student:
        return render(request, 'home.html')

    else:
        return redirect ('login')
```

Role selection

```python
@login_required
def home(request):
    subjects = Subject.objects.all()
    user = request.user
    books = []

    if user.is_authenticated:
        activities = UserActivity.objects.filter(user=user).select_related("book")

        if activities.exists():
            personalized_books = recommend_books_for_user(user)

            other_books = Book.objects.exclude(id__in=[b.id for b in personalized_books])

            books = list(personalized_books) + list(other_books)
        else:
            trending_books = get_trending_books()
            other_books = Book.objects.exclude(id__in=[b.id for b in trending_books])
            books = list(trending_books) + list(other_books)
    else:
        trending_books = get_trending_books()
        other_books = Book.objects.exclude(id__in=[b.id for b in trending_books])
        books = list(trending_books) + list(other_books)

    return render(request, 'home.html', { 'subjects': subjects,'books': books,})
```

Personalised book display

```python
from transformers import pipeline
from transformers import AutoModel, AutoTokenizer

model_name = "facebook/bart-large-mnli"
AutoTokenizer.from_pretrained(model_name)
AutoModel.from_pretrained(model_name)

classifier = pipeline("zero-shot-classification", model=model_name)

def classify_book(text, labels=None):
    # Validate inputs
    if not text or not text.strip():
        raise ValueError("Book summary text is empty.")

    if not labels:
        labels = ["Combined Science", "Mathematics", "Physics", "Computer Science", "Accounts"]

    result = classifier(text, candidate_labels=labels)
    return result["labels"][0]
```

Classifier

```python
from PyPDF2 import PdfReader
from PIL import Image
import pytesseract

def extract_pdf_text_and_meta(path):
    reader = PdfReader(path)
    meta = reader.metadata
    pages_text = []
    for p in reader.pages:
        try:
            pages_text.append(p.extract_text() or "")
        except:
            pages_text.append("")
    text = "\n".join(pages_text).strip()
    return text, dict(meta or {})

def ocr_image(path):
    img = Image.open(path)
    return pytesseract.image_to_string(img)

```

Metadata extractor

```python

def recommend_books_for_user(user, top_n=10):
    activities = UserActivity.objects.filter(user=user).select_related("book").order_by("-timestamp")
    if not activities.exists():
        return []

    last_book = activities.first().book
    if not last_book or not last_book.description:
        return []

    last_embedding = model.encode(last_book.description, convert_to_tensor=True)

    all_books = Book.objects.exclude(id=last_book.id)
    if not all_books.exists():
        return []

    book_texts = [b.description if b.description else "" for b in all_books]
    book_embeddings = model.encode(book_texts, convert_to_tensor=True)

    similarities = util.pytorch_cos_sim(last_embedding, book_embeddings)[0]
    ranked_indices = torch.argsort(similarities, descending=True).tolist()
    recommended = [all_books[i] for i in ranked_indices[:top_n]]

    return recommended
```

Recommender

```python
def secure_download(request, token):
    token_obj = get_object_or_404(DownloadToken, token=token)
    expiry = token_obj.expires_at if isinstance(token_obj.expires_at, datetime) else datetime.combine(token_obj.expir

    if timezone.now() > expiry:
        logger.info(f"Token expired: {token}")
        return render(request, 'download_expired.html', {'book': token_obj.book})

    encrypted_path = token_obj.book.file.path + '.enc'
    if not os.path.exists(encrypted_path):
        logger.error(f"Encrypted file not found: {encrypted_path}")
        raise Http404("Encrypted file missing")

    key = token_obj.key.encode()
    try:
        fernet = Fernet(key)
        with open(encrypted_path, 'rb') as f:
            encrypted_data = f.read()
        decrypted_data = fernet.decrypt(encrypted_data)
    except Exception as e:
        logger.exception(f"Decryption failed for token {token}: {str(e)}")
        raise Http404("Decryption failed")

    pdf_reader = PdfReader(io.BytesIO(decrypted_data))
    pdf_writer = PdfWriter()
    for page in pdf_reader.pages:
        pdf_writer.add_page(page)
```



```python
    pdf_writer = PdfWriter()
    for page in pdf_reader.pages:
        pdf_writer.add_page(page)

    expiry_str = expiry.strftime('%Y%m%d%H%M%S')
    password_source = f"{token_obj.key}-{expiry_str}"
    password = sha256(password_source.encode()).hexdigest()[:16]
    pdf_writer.encrypt(password)

    pdf_stream = io.BytesIO()
    pdf_writer.write(pdf_stream)
    pdf_stream.seek(0)

    UserActivity.objects.create(
        user=token_obj.user,
        book=token_obj.book,
        action='download',
        timestamp=timezone.now()
    )

    filename = token_obj.book.title.replace(' ', '_') + '.pdf'
    return FileResponse(pdf_stream, as_attachment=True, filename=filename)
```

Securing download

```python
def encrypt_file(file_path, key):
    fernet = Fernet(key)
    with open(file_path, 'rb') as f:
        data = f.read()
    encrypted = fernet.encrypt(data)
    encrypted_path = file_path + '.enc'
    with open(encrypted_path, 'wb') as f:
        f.write(encrypted)
    return encrypted_path
def generate_download_token(user, book, duration_hours=4):
    token = secrets.token_urlsafe(16)
    key = Fernet.generate_key()
    expires_at = timezone.now() + timedelta(hours=duration_hours)
    encrypted_path = encrypt_file(book.file.path, key)
    return DownloadToken.objects.create(
        user=user,
        book=book,
        token=token,
        key=key.decode(),
        expires_at=expires_at
    )
def decrypt_file(encrypted_path, key, output_path):
    fernet = Fernet(key.encode())
    with open(encrypted_path, 'rb') as f:
        encrypted_data = f.read()
    decrypted_data = fernet.decrypt(encrypted_data)
    with open(output_path, 'wb') as f:
        f.write(decrypted_data)
```

Encryption of downloading files

```python
_model = SentenceTransformer("all-MiniLM-L6-v2")
_faiss_index = None
_index = []

def build_index():
    """Build FAISS index from all Book objects."""
    global _faiss_index, _index
    books = list(Book.objects.all())
    _index = books
    corpus = [" ".join([book.title, book.summary or ""]) for book in books]

    if not corpus:
        _faiss_index = None
        return

    embeddings = _model.encode(corpus, convert_to_numpy=True, show_progress_bar=True)
    embeddings = embeddings.astype("float32")
    faiss.normalize_L2(embeddings)

    dim = embeddings.shape[1]
    _faiss_index = faiss.IndexFlatIP(dim)
    _faiss_index.add(embeddings)
```

```python
def semantic_search(query, top_k=5):
    global _faiss_index, _index
    if _faiss_index is None or not _index or not query.strip():
        return []

    query_emb = _model.encode([query], convert_to_numpy=True).astype("float32")
    faiss.normalize_L2(query_emb)
    sims, indices = _faiss_index.search(query_emb, top_k)

    results = []
    for idx, score in zip(indices[0], sims[0]):
        if 0 <= idx < len(_index):
            results.append({"book": _index[idx], "score": float(score)})
    return results
```

```python
@api_view(["GET", "POST"])
def search_view(request):
    query = request.GET.get("q", "")
    all_books = Book.objects.all()

    global _faiss_index
    if _faiss_index is None:
        build_index()

    results = []
    if query.strip():
        results = semantic_search(query, top_k=5)

    context = { "query": query, "results": results,"all_books": all_books,}
    return render(request, "search_view.html", context)
```

Semantic search

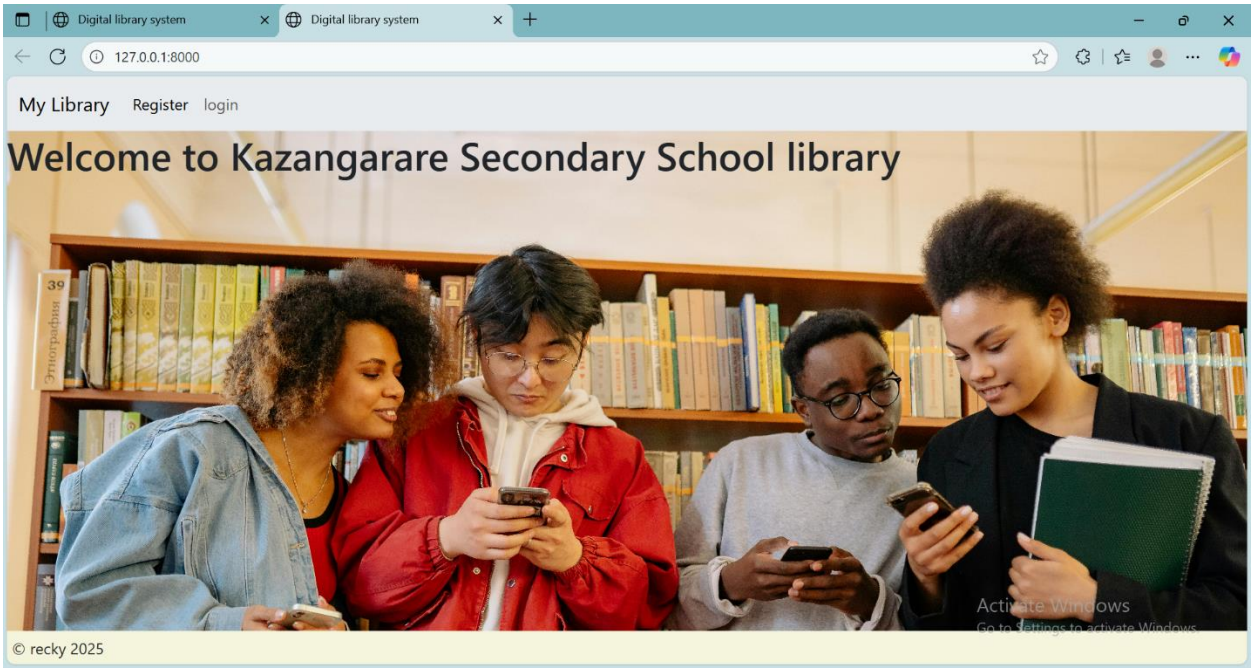
```
<div class="login">
    <div class="container mt-5">
    <h2 class="mb-4 position-center">Login</h2>

    {% if form.errors %}
        <div class="alert alert-danger">Invalid username or password.</div>
    {% endif %}
    <form method="post">
        {% csrf_token %}
        <input type="hidden" name="next" value="{{ next }}">
        <div class="login-control">
                <label class="form-label">Username</label>
                {{ form.username }}
            </div>
            <div class="login-control">
                <label class="form-label">Password</label>
                {{ form.password }}
            </div>
        <div class="button">
        <button type="submit" >Login</button>
        </div>
    </form>
    <p class="mt-3">Don't have an account? <a href="{% url 'register' %}">Register here</a></p>
    </div>
</div>
```

Login form

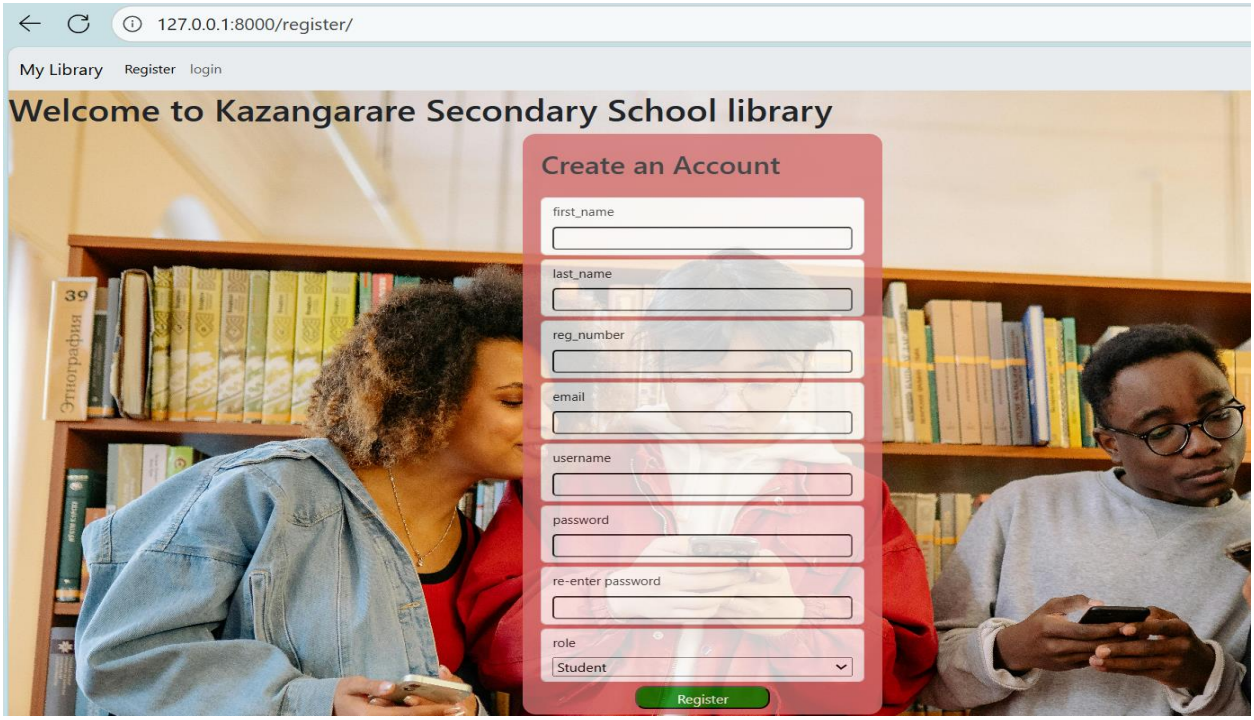
```
<!DOCTYPE html>
<html>
<head>
    <title>{{ book.title }} Download Ready</title>
</head>
<body>
    <h2>Your document is ready: <em>{{ book.title }}</em></h2>
    <p>This file will expire on <strong>{{ expires_at|date:"l, d F Y H:i" }}</strong>.</p>
    <a href="{{ download_url }}">Download Document</a>
    <p>Decryption Key: <code>{{ decryption_key }}</code></p>

</body>
</html>
```

Download page

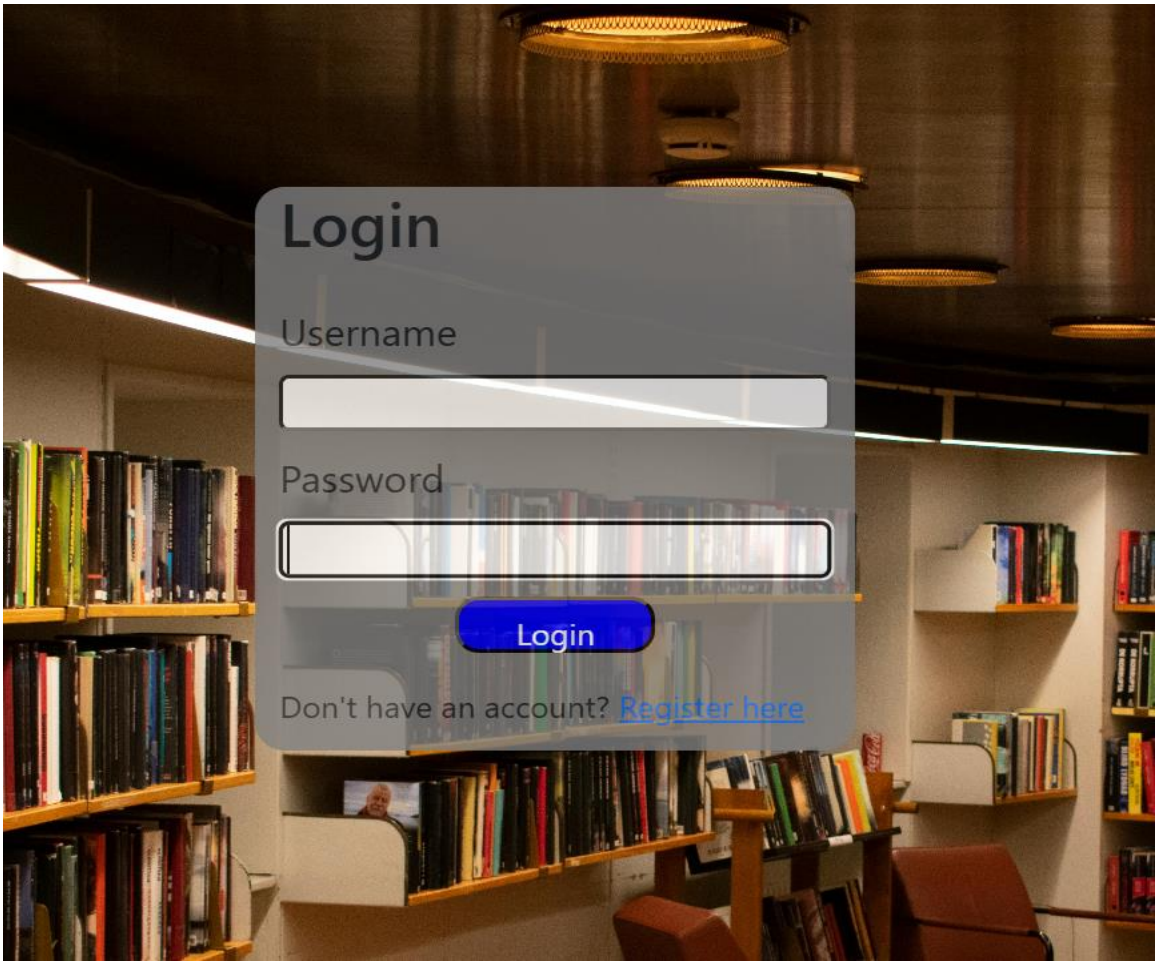
Appendix B User interface Mockups



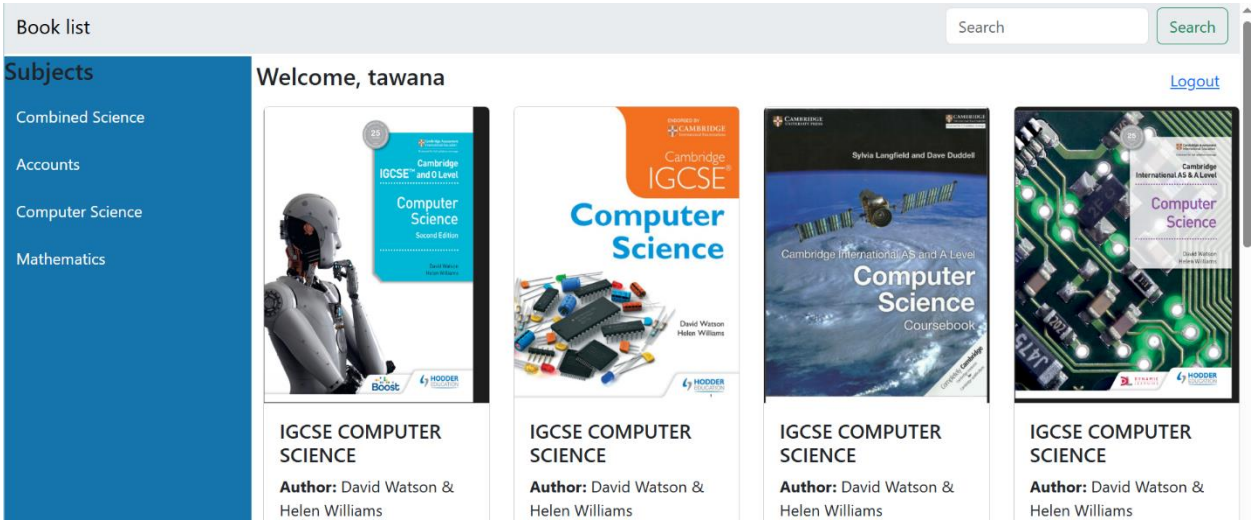
Home page



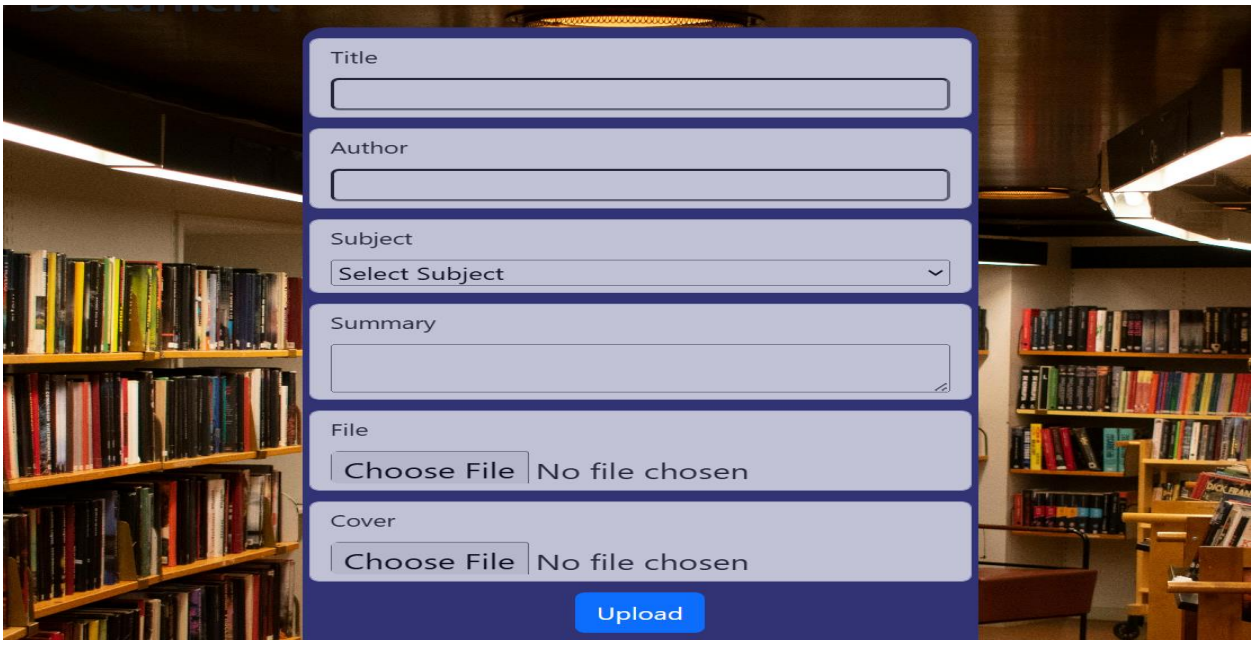
Registration form



Login interface



Personalised dashboard



Upload interface

Results for "computer science"

- **IGCSE COMPUTER SCIENCE** by David Watson & Helen Williams
  cambridge A level computer science
  View/Download Original File
  Similarity score: 0.68

- **IGCSE COMPUTER SCIENCE** by David Watson & Helen Williams
  cambridge A level computer science
  View/Download Original File
  Similarity score: 0.68

- **IGCSE COMPUTER SCIENCE** by David Watson & Helen Williams
  cambridge A level computer science
  View/Download Original File
  Similarity score: 0.68

- **IGCSE COMPUTER SCIENCE** by David Watson & Helen Williams
  cambridge O level computer science
  View/Download Original File
  Similarity score: 0.65

- **IGCSE COMPUTER SCIENCE** by David Watson & Helen Williams
  cambridge O level workbook computer science
  View/Download Original File
  Similarity score: 0.63

Search result

Appendix C Evaluation Data

Evaluation Data Form – Intelligent Digital Library System

Section A: User Information

Name (Optional): _____

Role (tick one):

- Student
- Teacher
- Librarian
- Other: _____

Frequency of Library Use:

- Daily
- Weekly
- Monthly
- Rarely

Section B: System Usability

Please rate the following on a scale of 1 (Strongly Disagree) – 5 (Strongly Agree):

Statement	**1**	**2**	**3**	**4**	**5**
The system is easy to navigate.	☐	☐	☐	☐	☐
Search results are relevant and useful.	☐	☐	☐	☐	☐
The recommendation system is helpful.	☐	☐	☐	☐	☐
The encryption download is helpful.	☐	☐	☐	☐	☐
Uploading and accessing books is smooth.	☐	☐	☐	☐	☐
Overall satisfaction with the system.	☐	☐	☐	☐	☐

Section C: System Performance

How would you rate the speed of the system?

- Very Slow
- Slow
- Average

- Fast
- Very Fast

Did you encounter any technical issues?

- Yes
- No
- If yes, please describe: _____

Section D: Impact and Usefulness

How would the system improve your learning/teaching/research?

……
……
……
.

Which feature do you find most useful?

Semantic Search

…….

Recommendations

……

Offline access

……

Book Upload & Metadata Extraction

……..

Others:

……..

What improvements would you like to see in future versions?

……
……
……