Module Title

Second Year Project

Assessment Weightage & Type

Second-Year Project Proposal Report (5%)

2024/25 Autumn

Members	University ID
Susant Kulung	24046121
Ganga Bajgain	24045880
Manish Bhattarai	24045919
Samir Nepal	24046042

Assignment Due Date: November 24, 2025

Assignment Submission Date: November 23, 2025

Submitted To: Miss Kritika Chaudhary

Project Tittle : SajiloDocs

Word Count : 2839

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table of figures

1 Introduction

The problem of keeping valuable personal documents such as IDs, licenses, and educational records is a common difficulty. They may be hard to maintain, store, and retrieve whenever required.

In this project, we suggest a Digital Locker System, which is a safe web application that serves as a repository of all Important documents. It enables one to store, sort, and share files with trusted people in a safe and easy way, offering a user-friendly yet secure way of handling documents in contemporary times. Not only this, users can also download a notarized version of their document and give role-based access to friends and family for their documents.

2 Problem Statement

Individuals and organizations frequently struggle to store, manage, and retrieve important personal and legal documents in an organized and secure manner. Users often keep documents scattered across emails, mobile galleries, physical files, or cloud drives, making it difficult to locate files quickly, track expiry dates, maintain versions, or share documents safely when needed. Existing solutions, such as Google Drive or Dropbox provide general storage but lack localized features like OCR extraction for Nepali documents, automatic translation, expiry reminders, secure role-based sharing, and structured document categorization tailored to everyday Nepali documentation requirements.

This problem is faced **by** students, professionals, families, and small businesses who routinely handle documents such as citizenship certificates, academic transcripts, IDs, property papers, and legal forms. Current tools fail to provide intelligent metadata handling, expiry notifications, or controlled sharing options, leading to inefficiency, disorganization, privacy risks, and time wasted searching for documents during urgent situations.

Therefore, there is a need for a centralized, secure, and smart document management application that enables users to upload, categorize, translate, share, and track their important documents with ease. The proposed solution aims to reduce document misplacement, automate reminders, enhance accessibility, and improve overall document security

3 Objectives

3.1 General Objective

- To develop a secure, intelligent, and user-friendly document management system that allows users to store, organize, translate, share, and manage personal and legal documents efficiently.

3.2 Specific Objectives

- To create a centralized platform for uploading, organizing, and categorizing documents with relevant metadata.
- To integrate OCR functionality for extracting text and identifying key fields from scanned Nepali documents.
- To implement secure document sharing with customizable permissions such as view, download, and edit access.
- To provide automatic translation of extracted Nepali text into English and recreate formatted, downloadable documents.
- To add an automated reminder system that alerts users about upcoming document expiry dates and renewal deadlines.
- To ensure document privacy and security through encryption, role-based access control, and activity logging.

4 Proposed Solution

The proposed solution is a Digital Locker System, a secure and intelligent web-based application designed to centralize the storage, management, and sharing of important personal and legal documents. The system provides a structured environment where users can upload documents, organize them into categories, extract text using OCR, translate Nepali documents into English, and securely share files with trusted individuals. By integrating automation, metadata handling, and strong security practices, the system aims to eliminate the common issues of document misplacement, disorganization, and privacy risks.

The platform will include smart features such as OCR-based text extraction, expiry-date reminders, metadata indexing, and downloadable notarized versions. Role-based access control ensures private and controlled sharing, while document recreation allows users to obtain clean, translated, and professionally formatted outputs. Through a combination of front-end usability and back-end intelligence, the system provides a highly accessible, secure, and efficient solution tailored to the needs of students, families, and small organizations.

Overall, this solution improves accessibility, enhances security, reduces manual workload, and ensures that important documents are always available, organized, and up to date.

5 Methodology (Scrum Framework)

This project is developed using Scrum Methodology, a widely used Agile framework designed to manage complex work through iterative development. The choice of Scrum supports the dynamic nature of the project, ensuring evolving requirements, stakeholder feedback, and other unexpected obstacles can be effectively handled. Scrum breaks the development process into development time-boxed iterations, known as Sprints, which are usually one to four weeks long. Every Sprint has a systematic workflow that leads the team in its planning phase to

Figure 1: SCRUM

execution and eventually review and reflection. This is a cyclic methodology approach that provides value in a continuous manner and keeps the project in tandem with the needs of the user throughout the project lifecycle.

The Scrum methodology consists of three key roles, namely, the Product Owner, the Scrum Master, and the Development Team. The Product Owner works as the main delegate of the stakeholders and defines, prioritizes and maintains the Product Backlog that contains all the necessary features, enhancements and tasks. The Scrum Master manages the Scrum process by eliminating roadblocks, maintaining compliance to the Scrum principles, and facilitating the appropriate team communication. The Development Team is an interdepartmental team that involves the implementation of the work that has been chosen to be used in each of the Sprints. This role division will provide clarity, responsibility and a smooth process of decisions during the project.

The Product Backlog serves as the starting point of the workflow, which is a dynamic and changing list of all the functional and non-functional requirements. The team reviews high-priority items in the backlog with the Product Owner during Sprint Planning and determines which items can be accomplished during the Sprint. The choice of tasks is the Sprint Backlog that is the plan of action of the development cycle. During the Sprint, the team holds a Daily Scrum (or stand-up meeting) every day to talk about progress, identify possible problems, and coordinate efforts. This regular contact ensures that everything is kept on track, delays are minimized and overall visibility is enhanced (C.C et al., 2024).

The team also holds a Sprint Review at the end of every Sprint in which the stakeholders are shown the completed Increment. This Increment is a possibly shippable and functional part of the product and the stakeholders can determine the progress and give effective feedback. The presence of constant stakeholder consideration will give the project the right course to follow the user expectations and minimize the chances of misunderstanding or rewriting the project. The team conducts a Sprint Retrospective after the review to reflect on processes utilized in the Sprint. In this meeting, the team determines the areas of improvement, the bottlenecks or challenges identified and action steps that may be taken to improve performance in the next Sprint.

The iterative nature of scrum coupled with its focus on transparency, inspection, and adaptation provides it a good fit to projects where requirements can be altered or refined as the project advances. The methodology encourages flexibility, delivery of value frequently and high

collaboration between the development team and the stakeholders. The combination of planning, development, feedback, and improvement in every Sprint makes Scrum a structured and disciplined but flexible model that can aid in ensuring the project comes in lean, incremental, and user-need preparation. This strategy favors the concept of lifelong learning, enhances the quality of products, and helps the team to be responsive to challenges during the development lifecycle (Scrum.org, n.d.).

6 Work Breakdown Structure (WBS)

A Work Breakdown Structure is a hierarchical, deliverable-oriented framework that breaks a project into smaller, manageable components. As explained in the attached paper, a WBS helps define and organize the total scope of a project by decomposing it from the overall objective down to detailed work packages that can be estimated, assigned, scheduled, and monitored. It focuses on planned outcomes—the products, services, or results the project must deliver, rather than the actions or tasks alone. This ensures clarity, accountability, and accurate cost and time estimation.

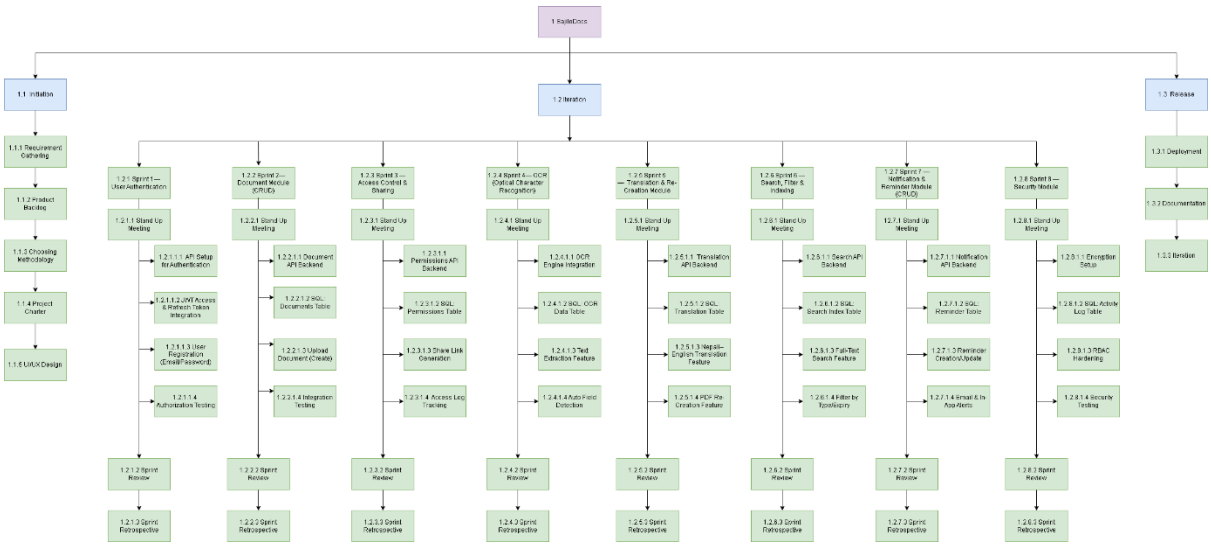

Figure 2: Work Breakdown Structure

The WBS of our SajiloDocs - Document Management System is split into three major phases namely Initiation, Iteration, and Release. The activities during each of the phases are aimed at building the core modules of the system including authentication, document management,

sharing, OCR, translation, search, notifications, and security. The following tasks denote what was done under each branch of the WBS and who was to work on it.

6.1 Initiation Phase (1.1)

This phase established the foundation of the project.

- **Requirement Gathering(1.1.1)**

We collected functional and non-functional requirements such as document upload, OCR extraction, sharing, translation, and security needs.
Assigned to: Samir (Lead), Manish supported with technical feasibility.

- **Product Backlog Creation(1.1.2)**

All features were broken down into backlog items for sprints (Auth, CRUD, OCR, Translation, Notifications, RBAC).
Assigned to: Samir + Sushant.

- **Choosing Methodology(1.1.3)**

Agile with 2-week sprints was selected.
Assigned to: Samir.

- **Project Charter Preparation(1.1.4)**

Project goals, scope, risks, and deliverables documented.
Assigned to: Ganga.

- **UI/UX Design(1.1.5)**

Initial wireframes for login, dashboard, document list, OCR screen, and translation screen were created.
Assigned to: Sushant + Ganga.

6.2 Iteration Phase (1.2)

This phase is divided into **8 sprints**, each implementing a system module.
All sprints included:

- Stand-up meetings
- Sprint review
- Sprint retrospective

Sprint 1 — User Authentication (1.2.1)

Goal: Implement user login, registration, and token management.

Tasks:

- API setup for authentication → **Samir**
- JWT access + refresh token integration → **Manish**
- UI for Login & Register → **Sushant + Ganga**
- Authorization testing → **Ganga**

Sprint 2 — Document Module (CRUD) (1.2.2)

Goal: Create folders, upload files, and store metadata.

Tasks:

- Document API backend → **Samir**
- SQL Documents Table → **Manish**
- Upload Document UI → **Sushant**
- Integration Testing → **Ganga**

Sprint 3 — Access Control & Sharing (1.2.3)

Goal: Share documents with view/download/edit permissions.

Tasks:

- Permissions API backend → **Samir**
- SQL Permissions Table → **Manish**
- Share link generation UI → **Sushant**
- Access log tracking → **Ganga**

Sprint 4 — OCR Module (1.2.4)

Goal: Extract text using OCR and detect fields.

Tasks:

- OCR engine integration (Tesseract) → **Samir**
- SQL OCR Data Table + Text extraction + auto-field detection → **Manish**
- OCR result screen UI → **Sushant**
- Testing accuracy → **Ganga**

Sprint 5 — Translation & Re-Creation Module (1.2.5)

Goal: Translate Nepali → English and generate formatted PDF.

Tasks:

- Translation API backend → **Samir**
- SQL Translation Table, PDF recreation with ReportLab → **Manish**
- UI for reviewing translations → **Sushant**
- Testing translated PDFs → **Ganga**

Sprint 6 — Search, Filter & Indexing (1.2.6)

Goal: Implement full-text search using OCR.

Tasks:

- Search API backend → **Samir**
- SQL Search Index Table → **Manish**
- UI for search & filter by type/expiry → **Sushant**
- Testing search results → **Ganga**

Sprint 7 — Notification & Reminder Module (CRUD) (1.2.7)

Goal: Send notifications for document expiry and sharing events.

Tasks:

- Notification API → **Samir**
- SQL Reminder Table → **Manish**
- Reminder creation/update UI → **Sushant**
- Email & in-app alerts testing → **Ganga**

Sprint 8 — Security Module (1.2.8)

Goal: System security hardening and RBAC.

Tasks:

- Encryption setup → **Sushant**
- SQL Activity Log Table → **Manish**
- RBAC hardening → **Samir**
- Security testing → **Ganga**

6.3 Release Phase (1.3)

Final polishing and deployment.

- **Deployment(1.3.1)**

Backend + frontend deployment setup.
Assigned to: Samir.

- **Documentation(1.3.2)**

User manual, API documentation, and system design.
Assigned to: Sushant + Ganga.

- **Iteration & Final Fixes(1.3.3)**

Performance enhancement, bug fixing, and final review.
Assigned to: Manish (Backend/OCR), Sushant (Frontend), Ganga (Testing).

7 Technology Stack

- **Frontend:** React

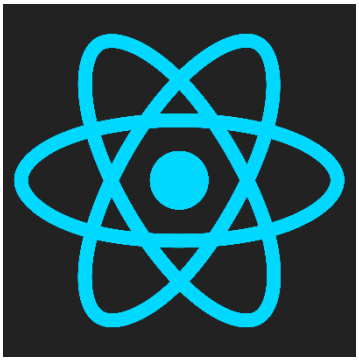

Figure 3: React

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library that aims to make building user interfaces based on components more "seamless". It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. According to the Stack Overflow Developer Survey, React is one of the most commonly used web technologies. (Wikipedia, n.d.)

React is the frontend of our Document Management Web App due to the efficient and fast structure and component-based structure which perfectly fits the features of our project. Our system has custom dashboards, document previews, sharing features, OCR outputs and live updates, and due to React, these dynamic interfaces are easily built and do not require page reloads. It is also simple to integrate with our backend APIs, allow reusable components of the UI and is compatible with Tailwind to form a clean and responsive design. In general, React will enable us to create an up-to-date, scalable, and friendly application.

- **Backend:** Django

Figure 4: Django Logo

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. (Django Software Foundation, 2025)

We chose Django as the backend, due to the fact that it is secure, reliable and is scalable, which best suits our document management application. The in-built functionality of Django such as authentication, ORM and form handling allows dealing with users, metadata, permissions and documents more easily. It is also compatible with MySQL and compatible with our OCR, translation and file-handling processes. In general, Django enables us to create a stable server-side infrastructure which has better security and development speed.

- **Database:** SQL

  Structured Query Language (SQL) is a domain-specific, standardized, programming language used to interact with relational database management systems (RDBMS) such as MySQL, SQL Server, IBM Db2, PostgreSQL and Oracle Database. (Jonker & Mucci, n.d.)

  MySQL is our preferred database, as it is fast, reliable and is appropriate in handling structured data like user information, document details, metadata and access permissions. MySQL offers high levels of data integrity, query efficiency and easy integration with Django ORM which aids us to handle data easily across the system. In general, MySQL provides a safe and scalable database within the framework of which the main features of our application can be properly supported.

Figure 5: MySQL Logo

- **Version Control & Collaboration:** GitHub

GitHub is a proprietary developer platform that allows developers to create, store, manage, and share their code. It uses Git to provide distributed version control and GitHub itself provides access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. GitHub has been a subsidiary of Microsoft since 2018 and its headquarters are located in San Francisco. (Wikipedia, n.d.)

Figure 6: Git and Github

GitHub was our choice of version control and collaboration tool in the project since it enables us to effectively manage our code and work collaboratively without any friction. GitHub enables us to follow the process of change, resolving, and keeping the development process clean with its features such as branches, pull requests, and commit history. It is a safe storage of our project as well and it is easier to distribute our code with teammates or instructors. In general, GitHub assists in maintaining an efficient, transparent, and structured development.

- **Project Management:** MS Teams.

  Microsoft Teams is a team collaboration platform developed by Microsoft as part of the Microsoft 365 suite. It offers features such as workspace chat, video conferencing, file storage, and integration with both Microsoft and third-party applications and services. (Wikipedia, n.d.)

Figure 7: Microsoft Teams

We used Microsoft Teams for communication and coordination throughout the project. It helped us easily share updates, discuss ideas, and stay connected with team members through chats, meetings, and file sharing. Teams also allowed us to organize our tasks and maintain smooth collaboration, ensuring everyone stayed informed and aligned

during the development process. Overall, Microsoft Teams improved our teamwork and made project communication more efficient.

8 Features & Functionalities

8.1 User Authentication and Profile Management

- Secure login and registration using email/password or Google sign-in.
- Personal dashboard to view uploaded and shared documents.

8.2 Document Storage and Organization

- Upload multiple types of documents (PDF, JPG, PNG, DOCX).
- Organize files into custom folders (e.g., ID, Academic, Property, Legal).
- Add metadata such as document name, issue date, expiry date, and description.

8.3 Access Control and Sharing

- Share documents or folders with others through:
    - View-only access
    - View and Download access
    - Edit access
- Generate a unique link or token to grant access.
- Manage access permissions and revoke at any time.

8.4 OCR (Optical Character Recognition) Processing

- Use **Python Tesseract OCR** to extract text from scanned documents.
- Detect important fields automatically, such as:
    - Document type
    - Document number
    - Date of issue
    - Expiry date

8.5 Translation and Document Re-Creation

- Automatically translate extracted Nepali text to English using Google Cloud Translate API or DeepL API.

- Recreate the translated text in a formatted document layout using ReportLab (PDF).
- Users can review and edit the translation before finalizing.
- Option to **download the translated version** as:
    - PDF (formatted document)
- Append a digital notarization footer or stamp for authenticity.

8.6 Reminder and Notification System

- Automatic reminders for documents nearing expiry.
- Customizable alert intervals (e.g., 7, 15, or 30 days before expiry).
- Notifications via email or in-app alerts.

8.7 Search, Filter, and OCR Text Indexing

- Full-text search using OCR results.
- Filter documents by type, expiry date, or shared access.

8.8 Security and Privacy

- Encrypted file storage and secure database connections.
- Role-Based Access Control (RBAC).
- Access logs for document viewing and editing history.

8.9 Downloadable Notarized Versions

- Option to download digitally notarized English versions of Nepali documents.
- Add signatures, date, and a watermark or digital stamp(Optional).

9 Conclusion

This project aims to provide a secure, organized, and intelligent document management system that simplifies the way users store, access, translate, and share important personal and legal documents. By integrating features such as OCR extraction, automated translation, expiry reminders, metadata-based organization, and controlled sharing, the system contributes a practical and reliable solution to the challenges faced by individuals and small organizations in managing their documents.

The proposed solution is fully achievable within the project timeline, supported by a clear architecture, a suitable tech stack, and well-defined features. Its focus on automation, security, and user convenience makes it not only practical but also highly relevant in real-world scenarios where document handling is often inefficient and disorganized.

Throughout the development process, this project provides valuable learning outcomes, including technical experience in full-stack development using React, Django, MySQL, Tailwind, and OCR/translation APIs. In addition, it enhances professional skills such as problem-solving, requirement analysis, project planning, and building user-centered solutions. Overall, the project demonstrates both technical growth and the ability to create a meaningful, user-focused application.

10 References

C.C, E., C.D, O. & H.E, A. (2024) Enhancing agile product development with scrum methodologies: A detailed exploration of implementation practices and benefits. *Engineering Science & Technology Journal*, 5(5), pp.1542-70.

Django Software Foundation. (2025) *Django* [Online]. Available from: https://www.djangoproject.com/ [Accessed 22 November 2025].

Jonker, A. & Mucci, T. (n.d.) *What Is Structured Query Language (SQL)?* [Online]. Available from: https://www.ibm.com/think/topics/structured-query-language [Accessed 22 November 2025].

Scrum.org. (n.d.) *What is Scrum?* [Online]. Available from: https://www.scrum.org/learning-series/what-is-scrum/ [Accessed 19 November 2025].

Wikipedia. (n.d.) *GitHub* [Online]. Available from: https://en.wikipedia.org/wiki/GitHub [Accessed 22 November 2025].

Wikipedia. (n.d.) *Microsoft Teams* [Online]. Available from: https://en.wikipedia.org/wiki/Microsoft_Teams [Accessed 22 November 2025].

Wikipedia. (n.d.) *React (software)* [Online]. Available from: https://en.wikipedia.org/wiki/React_(software) [Accessed 22 November 2025].