



HOLY ANGEL UNIVERSITY
SCHOOL OF COMPUTING
1st Semester, School Year 2025-2026



6WCSEVER

Project Proposal

Garcia, Charles

Gavino, Ericka Mae

Joves, Carla

Manabat, Alexander

Pinpin, Miguel

WD 302

September 14, 2025

Problem Statement

In today's academic environment, students and researchers frequently encounter challenges related to the flexibility of study materials, inefficient collaboration workflows, and the lack of a structured system for knowledge sharing. Existing solutions for document management and group collaboration often operate independently, forcing users to switch between multiple applications, which leads to the fragmentation of resources and an unconnected user experience. There is a clear need for an integrated platform that centralizes academic resources, facilitates peer-to-peer knowledge sharing, and provides tools for collaborative research and project development.

Several key points characterize current academic workflows:

1. **Fragmented Resources:** Study materials, such as lecture notes and supplementary resources, are often scattered across various devices and applications, making them difficult to organize, access, and share with others.
2. **Inefficient Collaboration:** Group projects, capstones, and thesis writing rely on a mix of tools, which lack a unified system that makes it hard for members to track changes, updates, and feedback from other members. This often results in version conflicts, communication breakdowns, and lost work.
3. **Lack of Quality Control:** The absence of a centralized, community-driven platform means there is no effective mechanism for students to rate, validate, or improve the quality of shared study materials. Reliable notes are hard to find, and valuable knowledge is often lost.

Sharesource is a proposed web application designed to be a comprehensive solution to these problems. It will provide a single, integrated platform for academic collaboration by offering the following core functionalities:

- **Centralized Repository with Quality Control:** A platform for uploading and organizing study materials, which can be tagged by subject and rated by peers to ensure high-quality, reliable resources.
- **GitHub-Inspired Version Control:** A system that allows users to clone and branch from existing study materials. This version history and activity log will track all changes, ensuring transparency, preventing data loss, and encouraging the continuous improvement of content.
- **Integrated Collaborative Workspace:** A unique interface that embeds a live, editable Google Doc on one side of the screen while simultaneously having a dedicated pane for related research materials, links, and real-time comments.

The primary target audience is university students, researchers, and academic groups seeking to improve their productivity and collaboration. While the initial focus is on students, the platform's features apply to a broad range of academic and professional fields that rely on collaborative research and documentation.

Project Objectives

The primary objectives for the development of Sharesource are to:

- Develop a user-friendly platform that consolidates the management of academic materials and collaborative projects.
- Implement a flexible version control system for study materials to facilitate collaborative editing and maintain a comprehensive history of changes.
- Integrate live document editing capabilities with research and feedback tools to create a seamless collaborative workspace.
- Foster a community of peer learning by enabling users to rate, comment on, and contribute to a shared knowledge base.

Target Users

Sharesource is intended for Holy Angel University's School of Computing (SOC) students, especially those who seek dependable and well-structured lecture notes, references, and teamwork tools for their academic endeavors. It also includes students who require a platform to work together, discuss drafts, and monitor changes when working on group projects, capstones, or thesis papers.

Project Scope

Inclusions:

- Centralized repository for study notes and references.
- Tools for rating, commenting, and giving feedback on notes.
- Version history with branching and cloning capabilities.
- Google Docs embed for live collaboration.
- Dashboard and personal library for managing top resources.

List of Features:

- Upload and manage notes (PDF, text, images).

- Tagging system by subject/topic.
- Peer rating system for clarity and usefulness.
- Commenting system for feedback and discussion.
- Version history, cloning, and branching of notes.
- Team collaboration tools for capstones/thesis.
- Google Docs integration (live editing within the platform).
- Dashboard with top-rated notes.
- Personal library for saved notes.

Basic Features:

- User signup/login system.
- Upload and organize notes.
- Tagging and categorization.
- Peer ratings.
- Personal library. Dashboard with top-rated notes.
- Personal library for saved notes.

Proof of Concept: A working prototype where -

- A student uploads a PDF lecture note.
- Other students browse by subject, read the file, and rate it.
- A peer adds a comment with extra references.
- Another student clones the file, adds their version (with diagrams), and uploads it with visible version history.
- A thesis group embeds their Google Doc, edits live within the platform, and discusses ideas in the comments section.

Delimitations

- The platform will not verify or guarantee the accuracy of uploaded notes; reliability depends on the contributors.
- It will not provide plagiarism detectors or enforce academic integrity policies.
- It will not replace official LMS or grading systems.
- Offline functionality is not supported.
- Real-time collaboration is limited to Google Docs embedding and optional WebSocket/Firebase features, not a full in-app editor.
- The system will not include advanced features such as AI-generated summaries or automated tagging in the initial version.

Anticipated Challenges

- Quality Assurance: Preventing misinformation or low-quality content in the uploaded notes.
- Plagiarism Risks: Avoiding misuse of materials without built-in plagiarism detection.
- Version Conflicts: Handling branching and version history without confusion or data loss.
- Scalability: Managing growing amounts of files, comments, and collaboration activity.
- Privacy and Security: Securing Google Docs API integration and protecting sensitive user data.
- Engagement: Motivating students to actively upload, rate, and comment on notes.
- Collaboration Performance: Ensuring smooth, real-time updates when using WebSockets or Firebase.

Methodology

Frontend

- React.js - for building dynamic and modular user interface.
- Tailwind CSS - for responsive, utility-first styling of the interface.

Backend

- Node.js – server-side runtime for handling requests and logic.
- Express.js – framework for creating RESTful APIs to connect frontend and backend.

Database

- MongoDB – NoSQL database for storing notes, comments, user profiles, and version history.

Other Tools & Integrations:

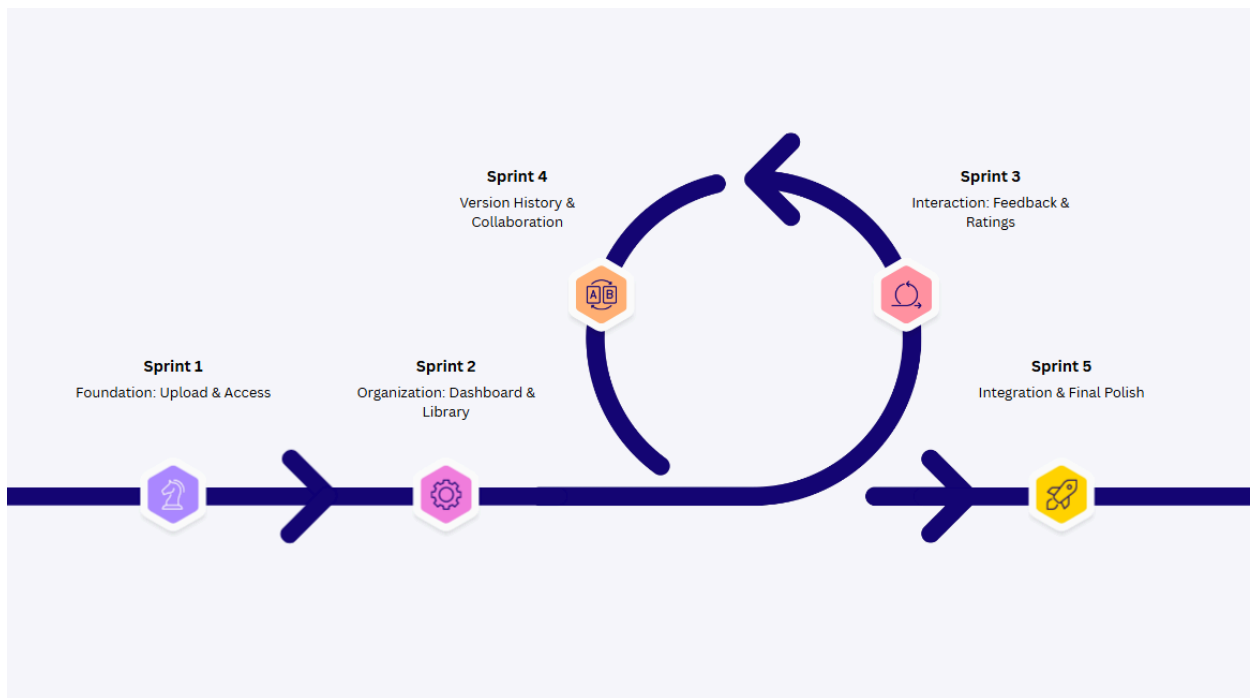
- Figma – for UI/UX prototyping and design
- GitHub – for version control and collaborative development
- Google Docs API – to embed and collaborate on live documents within the platform
- WebSockets/Firebase (optional) – for enabling real-time collaboration and live updates.

Timeline

- Week 1: Planning and Foundation:

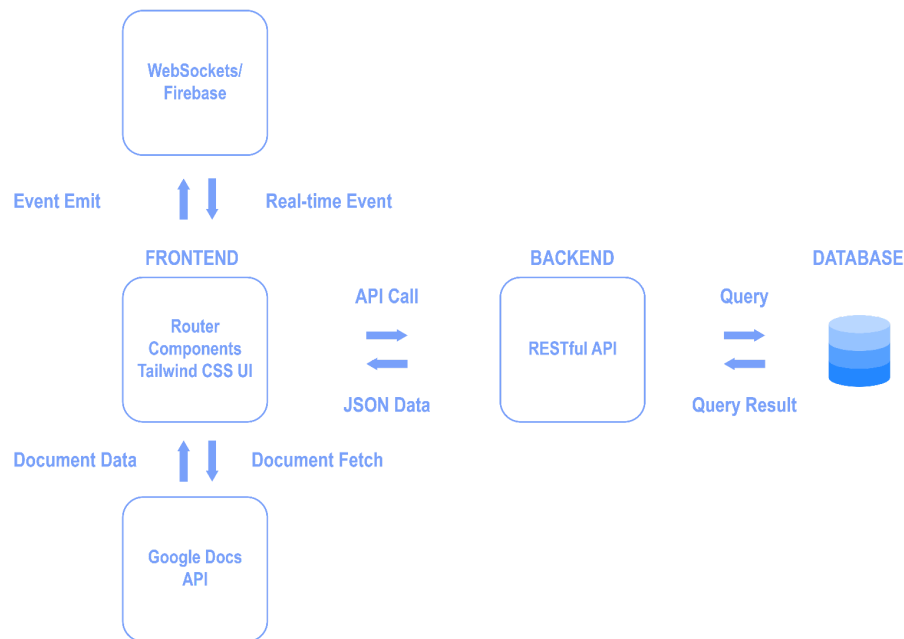
- Week 2: Backend Development and Authentication
- Week 3: Frontend, Core Functionality
- Week 4: Community and Versions
- Week 5: Collaboration and Proof of Concept

Development



Sharesource will follow the Agile Scrum methodology, dividing development into five sprints to deliver features step by step. The first sprint builds the foundation with user login and uploading notes, followed by a dashboard and personal library in the second sprint for better organization. The third sprint adds interaction through ratings and comments, while the fourth focuses on version history and group collaboration. Finally, the fifth sprint integrates Google Docs for live editing and polishes the platform. This incremental approach ensures that each sprint produces a usable feature, making the system evolve smoothly with continuous feedback and improvement.

Software Architecture



Sharesource uses a client-server architecture where the frontend provides the user interface and communicates with the backend through RESTful APIs. The backend manages requests and connects to MongoDB for storing notes, comments, and version histories. External tools like the Google Docs API enable live document collaboration, while WebSockets/Firebase can support real-time updates. This setup ensures smooth interaction, scalability, and collaboration across the platform.

Entity Relationship Diagram

Users

- UserID (PK)
- Username (Unique)
- PasswordHash
- Email (Unique)
- FirstName
- LastName

- Bio
- DateJoined
- ProfileImageUrl (optional)

Notes

- NoteID (PK)
- Title
- Content (Text, PDF, image links)
- OwnerUserID (FK to Users)
- SubjectID (FK to Subjects)
- DateCreated
- LastEditedDate

Subjects

- SubjectID (PK)
- SubjectName

Comments

- CommentID (PK)
- NoteID (FK to Notes)
- UserID (FK to Users)
- CommentText
- DatePosted
- ParentCommentID (FK to comments – replies)

Ratings

- RatingID (PK)
- NoteID (FK to Notes)
- UserID (FK to Users)
- RatingValue
- DateRated

Tags

- TagID (PK)
- TagName (ex. “Midterm”, “Chapter 5”)

CollaborativeSpaces

- spaceID (PK)
- spaceName
- description
- OwnerUserID (FK to Users)
- Members[userID, role, joinedAt]
- sharedNotesIds
- sharedDocIds
- createdAt
- updatedAt

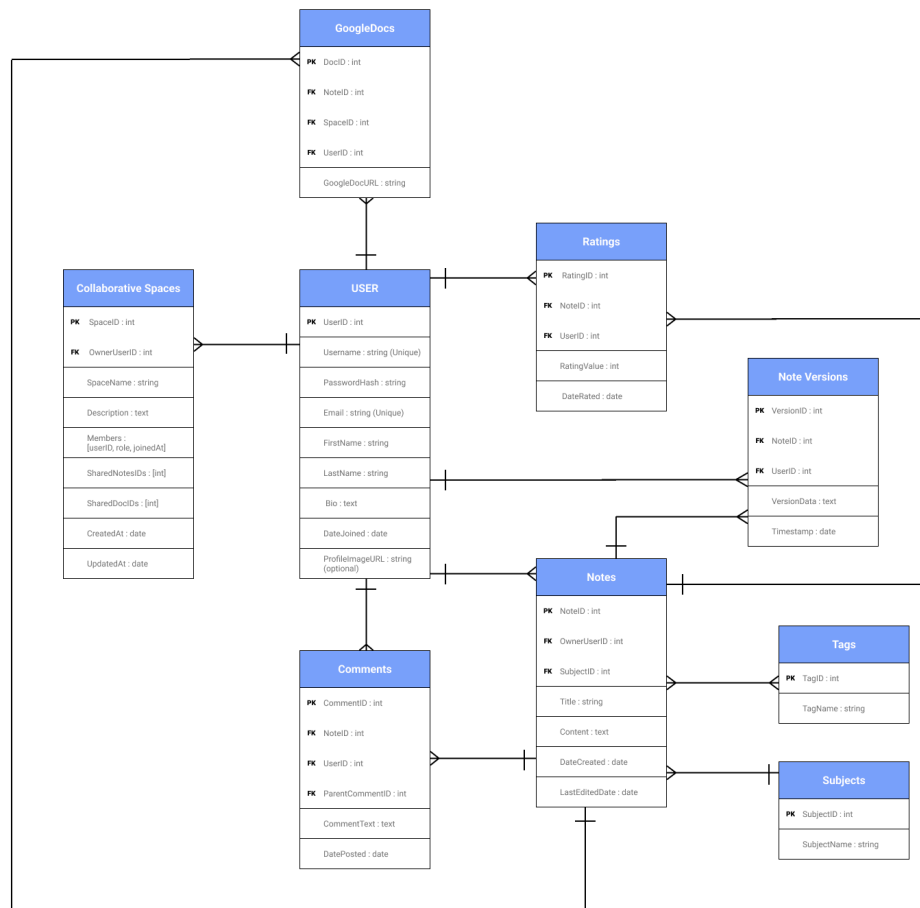
GoogleDocs

- DocID (Primary Key)
- GoogleDocURL
- NoteID (FK to Notes)
- SpaceID (FK to Collaborativespaces)

- UserID (Fk to Users)

NoteVersions

- VersionID (Primary Key)
- NoteID (Foreign Key to Notes)
- VersionData (The snapshot of the note's content)
- UserID (Foreign Key to Users - who made the change)
- Timestamp

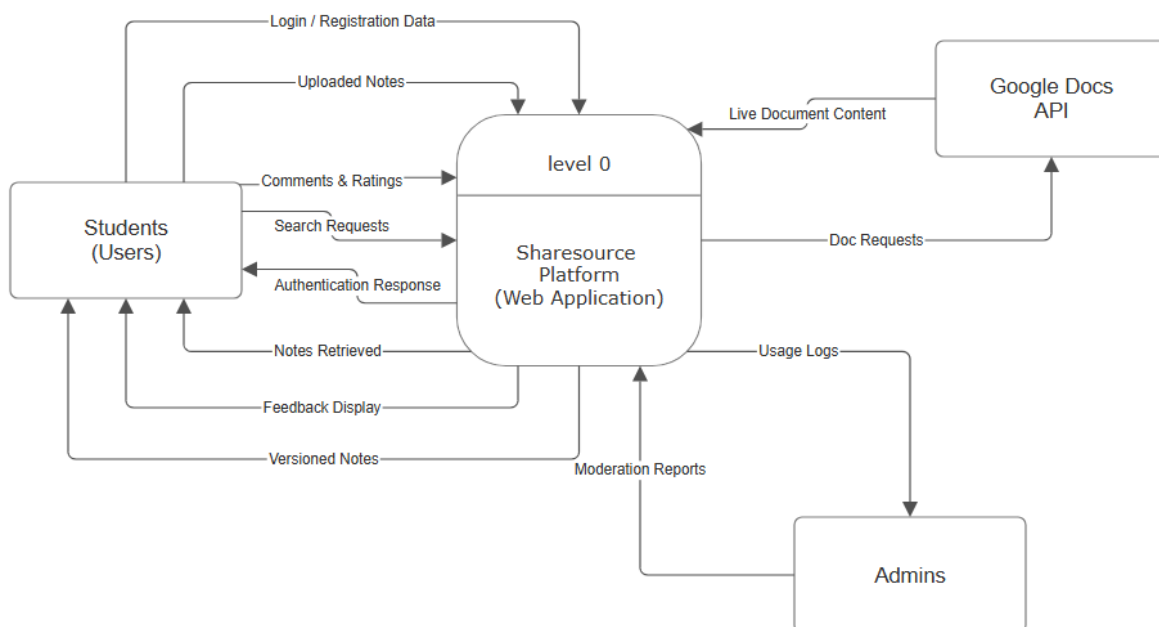


ERD Link:

<https://www.figma.com/design/SkSQgnMpcpMLxE7QoEoZa0/ERD-MANABAT-FINALS?node-id=0-1&t=tgET6ZGjLZkQ0Dmq-1>

A user can also create many notes, comments, ratings, collaborative spaces, Google Docs, and note versions, each of which belong to a specific user. Users can view their own and other users' notes. Each note is owned by one user and associated with a subject; notes may also have many comments, many ratings, and many versions. Notes can also be associated with many Google Docs and have many tags, which is a many-to-many relationship. The subjects are the way notes are organized into topics, where a subject can be associated with many notes. Comments are associated to both notes and users, and can, but do not have to, reference other comments (which are represented through the self-referencing one-to-many relationship), allowing for threaded replies. Ratings allow users to rate notes, and this creates a relationship between notes and users. Tags also allow flexible organization, where one note can have many tags, and one tag can belong to many notes. Users create collaborative spaces to work with others, and each collaborative space can have many members (table joining) and share many notes and docs. Google Docs lets users create documents that can be associated with notes and shared in collaborative spaces. Finally, note versions keep track of edits, with each version tied to the note it belongs to and the user who made the change.

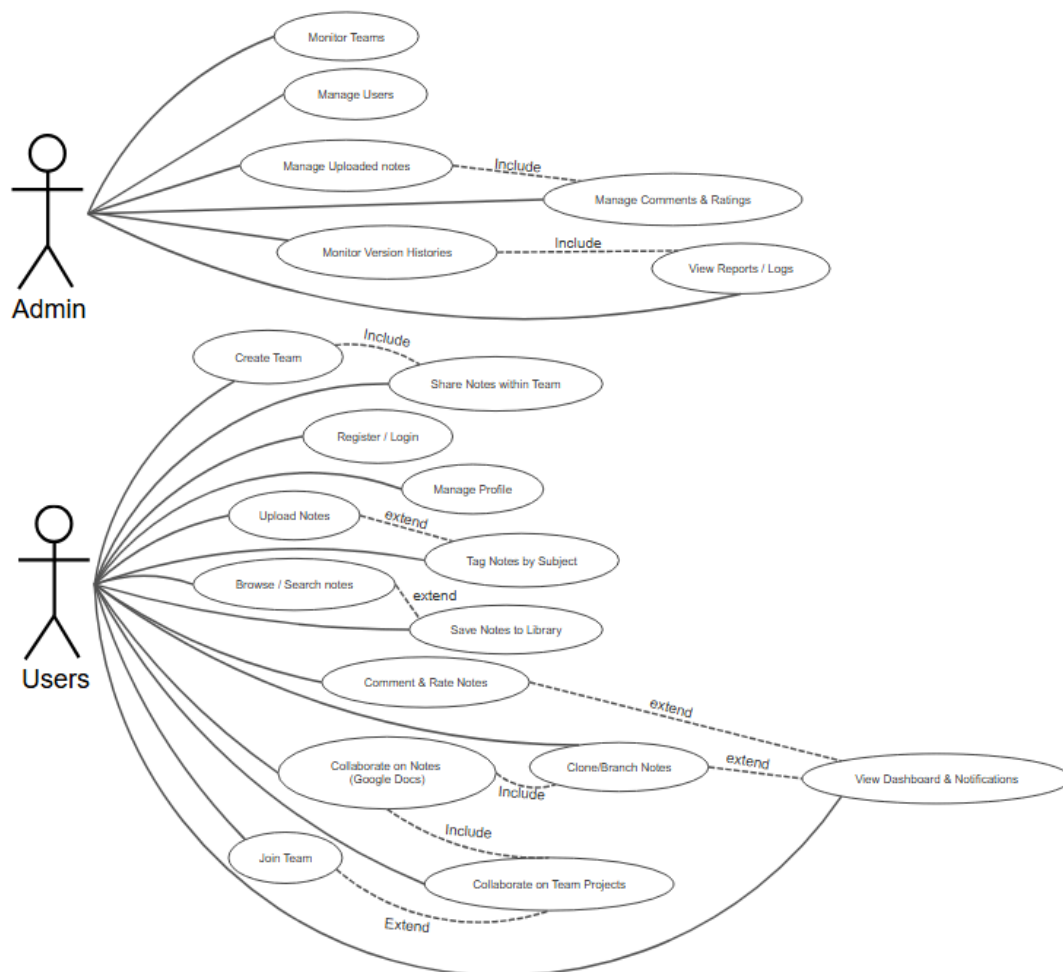
Data Flow Diagram



The Level 0 Data Flow Diagram illustrates the overall flow of information within Sharesource. Students interact with the system by submitting login details, uploading notes, posting comments, and sending search requests. The application processes these inputs and returns responses

such as authentication results, retrieved notes, feedback, and versioned files. Administrators exchange moderation reports and usage logs with the system to ensure proper oversight. Additionally, Sharesource communicates with the Google Docs API to fetch or update live documents, which are then delivered back to the students. This diagram provides a high-level view of how data moves between users, the system, external services, and administrators.

Use Case Diagram



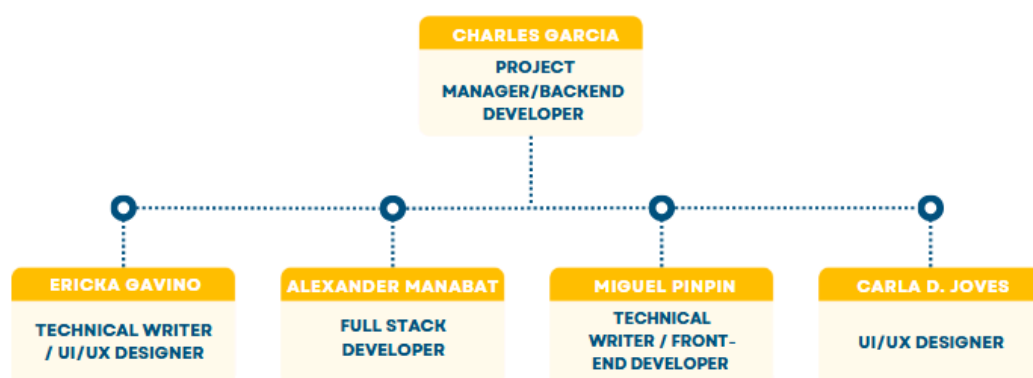
The Use Case Diagram for Sharesource shows the main interactions between users and the system. Students can register, log in, upload and share notes, create or join teams, search for materials, give feedback, and collaborate on documents through Google Docs. Administrators oversee the platform by moderating content, managing users, and monitoring activity logs. The diagram highlights how both students and admins interact with the system's core features, ensuring collaboration, accessibility, and proper management of shared academic resources.

Conclusion

- **Result**
 - Sharesource was built around a simple idea: make studying and group work easier by giving students one place to share and improve their notes together. The platform solves common problems like scattered resources, messy collaboration, and unclear version control. By combining familiar tools—such as a central library for notes, GitHub-style branching, and Google Docs for real-time editing, Sharesource creates a space where students can learn from each other more effectively. Our prototype shows that this approach is practical and has strong potential to support both individual learning and group projects.
- **Reflection**
 - Working on Sharesource has been a meaningful experience for our team. We didn't just design a system, we learned how to work together, divide responsibilities, and solve real challenges that came up along the way. Some of us focused on design, others on technical structure or documentation, but every contribution helped shape the project. We also saw how important communication and adaptability are, especially when following an Agile process. More than just a class requirement, this project showed us how technology can directly respond to the everyday struggles of students like us.
- **Recommendations**
 - Sharesource is only the beginning. In the future, it could include tools like automatic summarization, plagiarism detection, or even built-in real-time editing without needing Google Docs. Stronger moderation features would also help make sure notes are accurate and trustworthy. On the technical side, improvements like better security and scalable cloud hosting could prepare it for larger use. If developed further, Sharesource has the potential to grow into more than a school project, it could become a trusted study companion across different schools and communities.

Appendices

Organizational Chart



Name: Alexander Manabat

Role: Full Stack Developer

Responsibilities: Responsible for building the front-end using React.js and Tailwind CSS, developing and maintaining back-end APIs, managing and optimizing databases, ensuring smooth integration between client and server, and assisting in deployment and overall system performance.

Name: Charles Garcia

Role: Project Manager/Backend Developer

Responsibilities: Responsible for leading the team by planning the project scope, assigning tasks, monitoring progress, and maintaining documentation, while also handling backend development by designing the system architecture, building APIs, managing the database, implementing authentication and business logic, and ensuring proper testing and deployment.

Name: Carla D. Joves

Role: UI/UX Designer

Responsibilities: Responsible for designing user interfaces, ensuring good user experience, contributing to documentation, and assisting in front-end programming.

Name: Ericka Gavino

Role: Technical Writer / UI/UX Designer

Responsibilities: Responsible for creating and maintaining documentation, ensuring clarity in technical content, designing user interfaces, and supporting user experience improvements.

Name: Miguel Pinpin

Role: Technical Writer / Front-End Developer

Responsibilities: Responsible for producing clear documentation, maintaining technical accuracy, supporting user-facing materials, and contributing to front-end development.

Github Repository

The screenshot shows a GitHub repository page for '6WCSEVER-Final-Project', which is a public repository. At the top, there are buttons for 'Watch' (0), 'Fork' (0), and 'Star' (0). Below this, there's a navigation bar with 'main' selected, a 'Go to file' button, and a 'Code' button. The main content area shows a commit by 'duuuyyyyyy' titled 'Update README.md' with commit hash '71a5dc6' from 'last month'. Below the commit, there's a file named 'README.md' with the same commit hash and date. The README content is displayed below, featuring the repository name '6WCSEVER-Final-Project' and a description: 'Repo for 6WCSEVER'. The description text reads: 'Group Name: MECCA Charles Garcia was here Alexander Manabat was here Ericka Mae Gavino was here miguel pinipn was here Carla jove was here'. On the right side, there's an 'About' section with links to 'Readme', 'Activity', '0 stars', '0 watching', and '0 forks'. Below this, there's a 'Releases' section stating 'No releases published' with a link to 'Create a new release'. The 'Packages' section also states 'No packages published' with a link to 'Publish your first package'. At the bottom, there's a 'Contributors' section showing 5 contributors with their profile pictures.

6WCSEVER-Final-Project Public

Watch 0 Fork 0 Star 0

main Go to file + Code

duuuyyyyyy Update README.md 71a5dc6 · last month

README.md Update README.md last month

README

6WCSEVER-Final-Project

Repo for 6WCSEVER

Group Name: MECCA Charles Garcia was here Alexander Manabat was here Ericka Mae Gavino was here miguel pinipn was here Carla jove was here

About

Repo for 6WCSEVER

Readme

Activity

0 stars

0 watching

0 forks

Report repository

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 5

<https://github.com/GR4C3FR/6WCSEVER-Final-Project.git>