FEDERAL UNIVERSITY DUTSE

DEPARTMENT OF SOFTWARE ENGINEERING

CSE 404

(OPEN SOURSE SOFTWARE DEVELOPMENT)

GROUP ASSIGNMENT

PROPOSED PROJECT:

DEPARTMENTAL STUDY MATERIAL DRIVE (THE-ARCHIVE)

By

GROUP TWO(2)

TEAM ARCHIVE

1. Problem Statement

University students often face challenges in accessing comprehensive and high-quality study materials. Resources are scattered across various platforms, are sometimes difficult to find, or are expensive to acquire. There is no centralized, accessible repository for past exam papers, lecture notes, or departmental articles, leading to a fragmented learning experience. This lack of a shared knowledge base hinders collaboration and puts an unnecessary financial burden on students.

2. Project Goal

To create an open-source, collaborative, and centralized web platform for Software Engineering department where students can upload, share, and access study materials freely. The project will foster a culture of knowledge sharing and improve academic performance by making a wide range of resources readily available.

3. Project scope of the MVP (Minimum Viable Product)

The initial version of the platform will focus on the core functionalities of uploading and accessing materials, a MVP(Minimum Viable Product), then future features will be added later based on user feedback.

Core Features:

- **User Authentication:** A basic email and password system to allow registered users to log in.
- **File Uploads:** A user-friendly form for authenticated users to upload PDF documents, including essential metadata like **title, course code, and lecturer name**.
- **Material Display:** A main page that lists all uploaded materials in a clear, organized format.
- **Basic Search/Filter:** A simple search bar to filter materials by title or course code.
- **File Download:** A direct link to download each material.
- **Deletion Policy:** Users can only delete the materials they have uploaded.

Out of Scope for MVP:

- User profiles, ratings, or a commenting system.
- User-to-user messaging.
- Advanced search functionality with multiple filters.
- Version control for documents.

Open-Source angle: The community can enrich the platform by contributing study materials , and by extending and improving the codebase by adding new features like a rating system, notifications, and more advanced organization tools.

4. Tools and Technology

The project will be built using a simplified tech stack to allow all team members to contribute, regardless of their prior coding experience.

- **Frontend: HTML5, CSS3**, and **Vanilla JavaScript**. This allows for a focus on design and basic scripting, which aligns with the team's skillset.
- **Backend & Database: Supabase**, an open-source "Backend-as-a-service" platform. Supabase provides a powerful PostgreSQL database, file storage, and user authentication with a no-code interface.
- **Version Control: Git** and **GitHub**. The entire codebase will be managed on a central GitHub repository, and every team member will be required to use Git commands for their contributions.

5. Open-Source License

The project's source code will utilize two distinct open source licenses to cover both our frontend code and underlying backend technology.

Project License (Frontend code): The project’s source code would be released under the **Mozilla Public License 2.0 (MPL 2.0)**. This license provides an excellent balance for this project, ensuring:

- **Strong Copyleft:** Any modifications made to the project's source code files **must also be**

released under the MPL 2.0. This directly prevents proprietary forks and ensures the source code remains open and accessible to the community.

- **Flexibility:** The MPL allows for the integration of this project's code into larger applications that may have different licensing models, encouraging wider adoption.
- **Attribution:** The license requires anyone using the code to acknowledge the original creators by keeping copyright notices intact.

Supabase Licensing (Backend): The Supabase platform itself is open source, and it's components are licensed under various open-source licenses. The core Supabase platform is licensed unde the MIT license.

This means we can use their backend services freely, and we do not need to license out frontend code under MIT license.

Since our project relies on Supabase, proper attribution would be given to their work. This would be included in our project's documentation, and we would ensure we adhere to their licensing terms.

6. Task Distribution and Workflow

Each task is structured to ensure that every team member directly contributes code to the shared repository using Git. The workflow for each task will be:

1. **Pull** the latest changes from the main branch.
2. **Create** a new branch for the feature.
3. **Code** the feature.
4. **Commit** changes to the new branch.
5. **Push** the branch to the remote repository.
6. Create a **Pull Request (PR)** for review.
7. **Merge** the PR after review.

Task	Team Member(s)	Role & Responsibilities
Project Lead **Project & Git Manager, Supabase(database) set up, and the project setup**	Team Member 1 FCP/CSE/20/1011 Fattahan Taiwo Adeiza	Oversee the project, manage task delegation, and ensure deadlines are met. This member will also serve as the Git lead, helping other members with pull requests and merge conflicts and also start the initial project setup and structure. Also set up the Supabase project, create database tables and storage bucket. **Must push foundational files to Git, including the .gitignore, README.md, LICENSE, index.html, style.css, studyDrive.html and studyDrive.css, script.js,** to establish the initial project structure and documentation.
The landing page	Team Member 2 FCP/CSE/20/1012 Musa Yusuf Galambi	Will code the HTML and CSS for the landing page. This member would work closely with other members working on UI to ensure design consistency. **Must clone repository, pull first, then commit and push into the index.html and style.css files.**
Authentication UI (Sign-up)	Team Member 3 FCP/CSE/20/1013 Muhammad Saidu Hassan	**Code** the HTML and CSS for the user sign-up form. **Must clone repository, pull first, then create the signUp.html and signUp.css files, then commit and push the files into the repository.**

Task	Team Member(s)	Role & Responsibilities
Authentication UI (Login & Logout)	Team Member 4 FCP/CSE/20/1014 Kadijah Ibrahim	**Code** the HTML and CSS for the user login form and the logout button. **Must clone repository, pull first, then create the login.html and login.css files, then commit and push the files into the repository.**
Authentication Logic (Front-end)	Team Member 5 FCP/CSE/20/1016 Abubakar Sadiq Yunusa	**Code** the JavaScript logic for user sign-up and sign-in using the Supabase library. **Must commit and push into the script.js file with this functionality.** working closely with team member 3 & 4
File Upload UI,	Team Member 6 FCP/CSE/20/1017 Adam Abubakar Adam	**Code** the HTML and CSS for the file upload form. **Must Must clone repository, pull first, then commit and push the relevant changes into the studyDrive.html and studyDrive.css files,** also working closely with team member 8, and 10 for the pages layout and structure.
File Upload Logic	Team Member 7 FCP/CSE/20/1018 Mahmud Yahaya Abubakar	**Code** the JavaScript to handle the file upload process, storing files in Supabase Storage. **Must clone repository, pull first, then commit and push this logic into the script.js** also working closely with team member 6.

Task	Team Member(s)	Role & Responsibilities
Materials Display UI	Team Member 8 FCP/CSE/20/1019 Ahmad Muhammad Auwal	**Code** the HTML structure and CSS styling for displaying the list of materials. **Must Must clone repository, pull first, then commit and push the relevant changes into the studyDrive.html and studyDrive.css files,** also working closely with team member 6, and 10 for the page layout and structure.
Materials Display Logic	Team Member 9 FCP/CSE/20/1021 Umar Salihu Faruk	**Code** the JavaScript to fetch materials from the Supabase database and dynamically render them on the page. **Must clone repository, pull first, then commit and push this logic into the script.js** also working closely with team member 8.
Search & Filtering Logic and UI	Team Member 10 FCP/CSE/20/1022 Omokayode Abdulqudus Abdulrahman	**Code** the JavaScript to implement the search functionality on the displayed materials. **Must Must clone repository, pull first, then commit and push the relevant changes into the studyDrive.html, studyDrive.css files, and script.js** also working closely with team member 6, and 8 for the pages layout and structure.