CPSC 312 Web Project Proposal

Multi-User & Multi-Modal Notetaking Application

Zac Foteff

October 8, 2021

Abstract

Notetaking applications are rigid in their design by limiting users to only creating text or basic lists in their documents. I propose creating a web app that allows users to not only create notes, but also create sections of notes that can act as a repository notes and related documents like .docx, .md, and .pdf files. Users will be able to create separate accounts with their own segregated notes and documents. Users will be able to login to their account, create new notes, and upload/download files in their repositories. Each user's documents will be encrypted with their own private key so only they will be able to access their own documents

Project Topic and Focus

The focus of the project will be on creating a multi-modal and multi-user notetaking web application. The primary focus will be on creating an application that allows users to store and view their notes in plaintext or markdown form. Additionally, the application will allow users to store and retrieve documents in a variety of other forms. These documents will include .pdf, .md, .pdf, and other file encodings. Documents can be separated into user defined sections to allow grouping of related documents. Each user's data will be encrypted using a private-public key system that allows the user to store sensitive data without fear of other users gaining access to their documents. In browser, the user can scroll through their documents and view files on a window in the application. By triggering a toggle, the user can switch from reader view to writer view, avoiding possible unwanted edits. Some personalization options will be provided to the user, such as changing the theme of the website layout and color-coding document sections for better visual organization.

The primary design issue this project will accomplish will be creating the systems for segregated user accounts along with persistent storage of a variety of filetypes. This issue will likely be resolved by using a combination of relational and document databases to store user data. The risks of this approach include decreases in performance due to querying multiple databases per API call to the application.

Possible Technology Stacks

MEAN Stack

This technology stack is a fast practical stack that emphasizes universal data transmission and using a single programming language for the creation of web applications. This stack is easily scalable and is designed around streamlining the development process. This stack is not dependent on any specific operating system. This stack is growing in popularity and is supported more and more by large companies such as Google.

Pros:

- Uses JavaScript as major underlying language
- All components communicate using JSON objects
- Uses open-source technology

Cons:

- Relative novelty of the stack could lead to sustainability issues in the future if issues are discovered in dependencies
- Usability issues with Internet Explorer

LAMP Stack

This technology is the industry standard web framework for web applications. The stack is efficient, effective, and proven stable through the thousands of web applications today that utilize this stack. The stack is universal to all operating systems by altering the stack into the WAMP and MAMP technology stacks. Many other languages such as Python or Perl can be used in place of PHP in the stack.

Pros:

- Well documented and supported
- Linux is well documented and designed for web hosting
- Easily adaptable to changing business needs
- PHP is very fast

Cons:

- Steep learning curve of PHP can lead to security issues
- Other database options than MySQL have become more efficient and effective

Ruby on Rails

This technology stack is designed primarily around the Ruby on Rails programming language. The server-side component is written in Ruby on Rails and the client-side component is built using HTML, CSS, and JavaScript. The client and server-side components integrate well using JSON or XML to communicate data.

Pros:

- Ruby on Rails has many default structures for database management
- Ruby on Rails has an extensive collection of Gems for quickly extending complicated functionality to your page
- Can be integrated with a wide variety of front-end frameworks

Cons:

- Requires learning Ruby on Rails
- Slow runtime speed

Web Project Technology Stack

I will use the MEAN technology stack to implement this application. I have chosen this stack because the MEAN stack is rapidly growing in popularity in professional settings. The LAMP stack is easy to learn, however it is a standard that is well represented in many web applications that already exists. The Ruby on Rails stack seems interesting to learn about, however the time it would take to learn Ruby on top of all the other frameworks required for this project lead me to forgo this framework.

Learning Resources

- https://www.educba.com/software-development/courses/mean-stack-course/
- https://www.ibm.com/cloud/learn/mean-stack-explained
- https://www.udemy.com/course/learn-mean-stack/

Stack Description

- Front end: HTML, CSS
  - Framework: Angular.js
- Back end: Node.js
  - Framework: Express.js
- Database: MongoDB

Functional Requirements

Front-end	Back-end
• Users must be able to sign in • Users must be able to view all documents associated with their account • Users must be able to toggle between read and write modes for their notes • Users must be able to sign out and switch account from any screen	• Users must be able to retrieve their specific documents from their account • On sign-in, user documents must be retrieved • User documents must be decrypted before being served to user • Users must not be able to access other user's documents

• Users must be able to upload existing documents to the sections • Users must be able to share documents from the webpage to email	• Users must be prompted to commit their changes to a document before the can leave a note • Users must be able to add documents to their database

Possible Hosting Services

I'm focused on selecting a web hosting service with a short learning curve that allows all application resources to be contained to one resource group. It is essential for this hosting service to have the capability to scale with the concurrent number of users. Additionally, the service needs the latency to handle concurrent calls for decryption or document retrieval.

Hosting service	Feature 1	Feature 2	Cost
AWS	Select operating system and virtual environment for hosting web application	Only pay for the resources you use for the web application	~$107 USD / year + Free tier
GoDaddy	Preconfigured hosting environments for easy deployment	Built in project management tools in the hosting service	$4.99 / month