Software Engineering - Cloud Drive Project

**LEMP Stack:**

[How To Install Linux, Nginx, MySQL, PHP (LEMP stack) on Ubuntu 18.04 | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-ubuntu-18-04)

**The Assignment Page:**

[CSC445 Software Engineering (blackboardcdn.com)](https://learn-us-east-1-prod-fleet01-xythos.content.blackboardcdn.com/blackboard.learn.xythos.prod/57857cf84a2e5/5373453?X-Blackboard-Expiration=1613703600000&X-Blackboard-Signature=g%2BcI1BM9LQgTszJIn%2Fq%2BXmz3Bl6fi10wtUBTDnP63ZQ%3D&X-Blackboard-Client-Id=200051&response-cache-control=private%2C%20max-age%3D21600&response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27CSC445Project_Spring2021.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20210218T210000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=AKIAYDKQORRYTKBSBE4S%2F20210218%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=2a7e4b818fba3e70217777bc57d1882038fb58681840aaeeb45769fb76ffca42)

**Function priority and assignment**

**display\_table() -1 Jae**

**allocateSpace() (create user folder when an account is made with 3 subfolders for privacy) –1 Abdulsalam**

**downloadFile() –1 Kevin/Abdulsalam**

**isUploadable(File file) –1 Jae/Will**

**signOut() (understand user login) – 1 Abdulsalam**

**GetStorage() –1 Will**

**uploadFile(File file) -- 1 Kevin/Abdulsalam**

**Beautify code w/CSS – Jae/Will**

**setPermissionsFile(int FileID, int UserID, + int number) - 0 is private, 1 is shared with all contacts, and 2 is public. All other values will be invalid. -- 2**

**setPermissionsFolder(int FolderID, int UserID, int number) - 0 is private, 1 is shared with all contacts, and 2 is public. All other values will be invalid. -- 2 Jae**

**StorageBar () -- 2 Will**

**updateStorageBar(int newStorageValue) --2 Jae/Will**

**updateFilePath(int fileID, String filePathway) --2**

**addCaption(File file) / or (int FileID) – 2 Will**

**addContact(int UserID, int otherUserID) –2 Kevin**

**addFolder(String directory, String folderName, int UserID, boolean deletable) -- 2**

**isContact(int UserID) – 2 Kevin**

**deleteFile(File file) --2 Will**

**lookupUser(String username) – 2**

**copyFile(File file) or (int FileID) -- 3**

**deleteFolder(String pathway) -- 3**

**renameFile(File file) or (int FileID) –3 Jae**

**deleteContact(int UserID, int otherUserID) -- 3**

**Presentation Script:**

**Slide 1:**

**Slide 2:**

**The cloud drive we developed is a file hosting service that allows account holders to save files on a cloud and access them from any computer with an internet connection. Users will be able to make accounts that allow them to sign in, sign out, add friends, set privacy permissions to files, and share files with other users along with the main operations of storing, downloading, viewing, deleting, and organizing files. We also have the added purpose of anonymity so that users don't need to input any personal information and we will not ask for any or track user activity.**

**Slide 3:**

**The main goals of our software project were to create a working server to host a cloud drive, develop a working user interface along with working connections to a MySQL database. We also wanted to make the website visually appealing, even if we weren’t able to complete some of the lower priority operations, such as the storage bar.**

**Slide 4:**

**Slide 5:**

**The Server is constructed on an old macbook that is no longer used.**

The website is hosted from a home network to the internet without the help of any third party companies besides the one used for the domain name.

The project was created in a LEMP stack

LEMP stack includes:

Linux OS

Web Server

Database Server

Dynamic Processor

**Slide 6:**

A LEMP stack is a group of software that can be used to serve dynamic web pages written in PHP. The acronym describes a Linux operating system, ubuntu server in this case, an Nginx (pronounced like “Engine-X”) web server. A MySQL database and lastly, a PHP processor

**Slide 7:**

In order to serve our web pages to site visitors, we used Nginx, a modern, efficient web server. Nginx is a high performance server that is the host of the online interface when users need to access the website, Ngnix has many capabilities but for the scope of this project Nginx manages the html, php, and css files we create in addition it handles the user load scaling. Nginx also maintains the directives that declare what domain names and ports the website responds for. When users are accessing certain web pages associated with the cloud drive, they will be directly communicating with this server.

**Slide 8:**

**Slide 9:**

**MySQL is an** [**open-source**](https://en.wikipedia.org/wiki/Open-source_software)[**database management system**](https://en.wikipedia.org/wiki/Relational_database_management_system) **(RDBMS). MySQL works with our site’s php code to implement a relational database in a server's storage system to manage our users and the files that get uploaded. It allows for user access to the network and can be used to create backups of the data we store.**

**Slide 10:**

**This is the table that holds the entity for user\_info and FILE\_INFO. User info table is used whenever a user signs in or creates a new account.**

**Slide 11:**

**In the picture shown, these are the entities that SQL Database contains data for each folder and contacts. The Folder has the primary key FOLDER\_ID and a foreign Key FUSER\_ID.**

**Folder table is associated with an owner which is referenced in the foreign key. It keeps track of each users folders.**

**Contacts table keeps track of who becomes friends with who**

**Slide 12:**

**PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Our** PHP processor handles the backend of the website, sets the filesize download limit, time to live for a function call, etc.

**Slide 13:**

**Slide 14:**

**Set up host machine firewall settings to accept for port 443 and 80 (HTTPS and HTTP)**

**Obtained the domain name “Stuffmyfiles.cf”**

**Altered the DNS management settings → Type A forwarded to Router’s Public IP Address**

**Port forwarding was set up in router to point incoming connections for port 80 and port 443 to my webserver**

**TLS Certificate was installed on the host machine for HTTPS encryption**

**Slide 15:**

**Slide 16:**

**Account Registration is quite simple but hard. When a new user enters the website, they click on the button. The button brings them to the Registration page, where they will input a username and password. The inputted username and password are sent to SQL database.**

**Slide 17:**

**When the user inputs in a new Username and password, Registration.php will check the SQL database to see if the username is taken. If taken it will show an alert. If not taken, the host computer will create a new Folder with the username and saves the data in USER\_INFO in SQL database.**

**Slide 18:**

**This part is the HTML code for the registration page. When a user goes into the Registration page, they will see input username, input password and confirm password. This html code checks if the username is valid, and checks if the password they have inputted is same as confirm password.**

**Slide 19:**

**When a User wants to login to the site, they will proceed to input the username and password. The inputted Username and Password will be checked through the Database to see if there are any matching. If there are matching, then the user is allowed access to its home page.**

**Slide 20:**

**The image shown is the login screen a user is directed to when they punch in the URL to our website.**

**Slide 21:**

**Slide 22:**

**The biggest and the most important element in Login Page is the User Session. User session prevents users from accessing the main homepage without logging in, and checks the current activity of the user. Whenever a user is uploading their new file, User Session prevents the user from automatically logging off, whenever the page is refreshed.**

**Slide 23:**

**The home page is very simple. It is the page the user will see after they successfully log in to their account. The homepage will be consisted of the basic functions of a cloud drive. There will be a display table consisting of the uploaded files, upload and download button, and finally a user search bar.**

**Slide 24:**

**This was our initial design of the welcome page when the user successfully logs in.**

**Slide 25:**

**Due to time constraint, we were only able to finish the basic necessities for the project. The upload, user search, display table, and download.**

**Slide 26:**

**Slide 27:**

**The front hand functions are the functions that are viewable to the user. These include, style.css, upload, download, logout. Organization by Name, Date, Size, and Type. User Contacts and finally, the display table that shows all the uploaded files.**

**Slide 28:**

**Slide 29:**

**Config.php was a function that worked on connecting our codes to the SQL database. It tries to connect to a specific user’s table in the database, and returns an error message when it was not able to connect.**

**Slide 30:**

**Slide 30:**

**Slide 32:**

**The way we went about coding this is by each installing a program called PuTTY. This software allowed us to log in to access the host server and code on the linux terminal simultaneously. Setting this up was a simple task of adding the IP, the port, and checking off that we would be using an SSH connection. SSH (or secure shell protocol), is a cryptographic network protocol that can be used to operate network services over an unsecure network.**

**Slide 33:**

**To actually code in the terminal there were a few commands that we needed particularly more than others. The first one was sudo su which allowed us to gain root privileges so we could edit files and server information. The next useful command we had to use was cd. This directly stand for change directory and was necessary to get to where we were working. The command to edit the contents of a file was nano followed by the file name. Other useful commands were cat> which was used for creating files, ls which showed everything in a file directory, and mv/rm, which were used to move and delta files.**

**Slide 35:**

**During this project, we had some difficulties. Our first major setback was that the host computer refused to cooperate with us and the wi-fi adapter could not be recognized. The second main setback was our lack of knowledge on PHP and the linux terminal. Another huge setback was that only 1 person was able to edit a program at any given time in the linux terminal. We could all be logged in at the same time, but if we wanted to work on the same file, only one could edit and save. Another issue was with the hard drive meant for storing the files. We were unable to get them to store in that hard drive due a permissions issue and we ended up simply using a folder within the directory we were writing code in. The last issue was a time constraint. We were unable to get every function that we wanted working, but we are happy with what we have.**