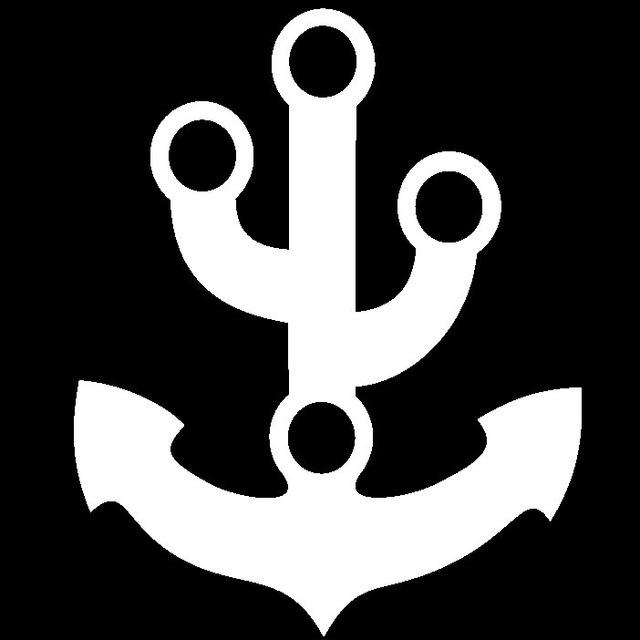
**Sinkd**

Cloud Storage Solution for the Ever Moving Student



**CIS444 Group A**

David Bright, Brandon Tilley, Sam Doroudi

Crystele Dierickx, Chris Banci, Andrew Zhang

Website URL: <http://cis444.cs.csusm.edu/group_A/sinkd/login.php>

December 3, 2017

[ Project Description ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sinkd is the solution to one of the problems most majors at CSUSM face- no centralized storage. Our project gives CSUSM students the ability to access, create folders, upload files, and manage them in a professional way. Users don’t have to worry about losing or wasting time searching for their files. Sinkd allows users to organize and manage files in three easy steps: sign-up, sign-in, and get started. All files are stored in a cloud instead of a local computer. This functionality lets the user access files from anywhere in the world. One of our main focuses was to create a reliable and easy to use application. In today’s world simplicity is the key. Our users don’t have to spend time learning how to use the site. Sinkd is very easy to use, which gives users more power. There are similar platforms like Sinkd, however our product is free, offers limitless storage, and better yet, caters to CSUSM students.

[ Specifications ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Requirements

1. The system shall let the user create an account and login securely to the website.
2. The system shall let the user upload files of any type to the website.
3. The system shall let the user create, rename, and delete folders.
4. The system shall let the user manage their files through the actions of deletion, renaming, viewing and sharing.
5. The system shall let the user manage their account through the actions of changing their display name, changing their password and deleting their account.
6. The system shall let the user sort their files in ascending or descending order.

Alongside with the requirements, users only need access to the following to use Sinkd:

1. Internet
2. Web browser
3. Files to upload (max upload size per file: 2MB)

Technologies/Tools Used

* **TableSorter:** A jquery plugin used to modify and sort data in the table dynamically.
* **Bootstrap**: A front-end framework used to help design our website easily.
* **JQuery:** A javascript library used to make client-side scripting simple and easy.
* **Ajax:** Used to make asynchronous requests to the server to update webpages dynamically without the need of reloading.
* **PHP:** Used to interact with the database and make requests to the server.
* **HTML/CSS/Javascript :** Used for front-end and UI design.
* **MySQL**: Used to manage the user's account information for signup and login system.
* **Server**: Used to store the user's uploaded files.

[ User's Manual ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructions to use website

1. Sign up and create an account. A name, username, and password are required.
2. Login with your new credentials.
3. On top right of the page, click **+Folder** to create a new folder.
4. Click **Upload** to upload files to your desired folder.
5. Awesome! You have now uploaded a file to your account.

File Management

1. Click on the buttons under **Action** to manage files.
   1. To *delete*, click on the red **trashcan** icon.
   2. To *edit* name, click on the green **pencil** icon.
   3. To *view*, click on the blue **download** icon.
   4. To *share*, click on the white **links** icon.
   5. To *save*, click **view**, a new window opens. Right click and select **Save Image As**.

Folder Management

1. Click on the buttons under **Action** to manage folders.
   1. To *delete*, click on the red **trashcan** icon.
   2. To *edit* name, click on the green **pencil** icon.

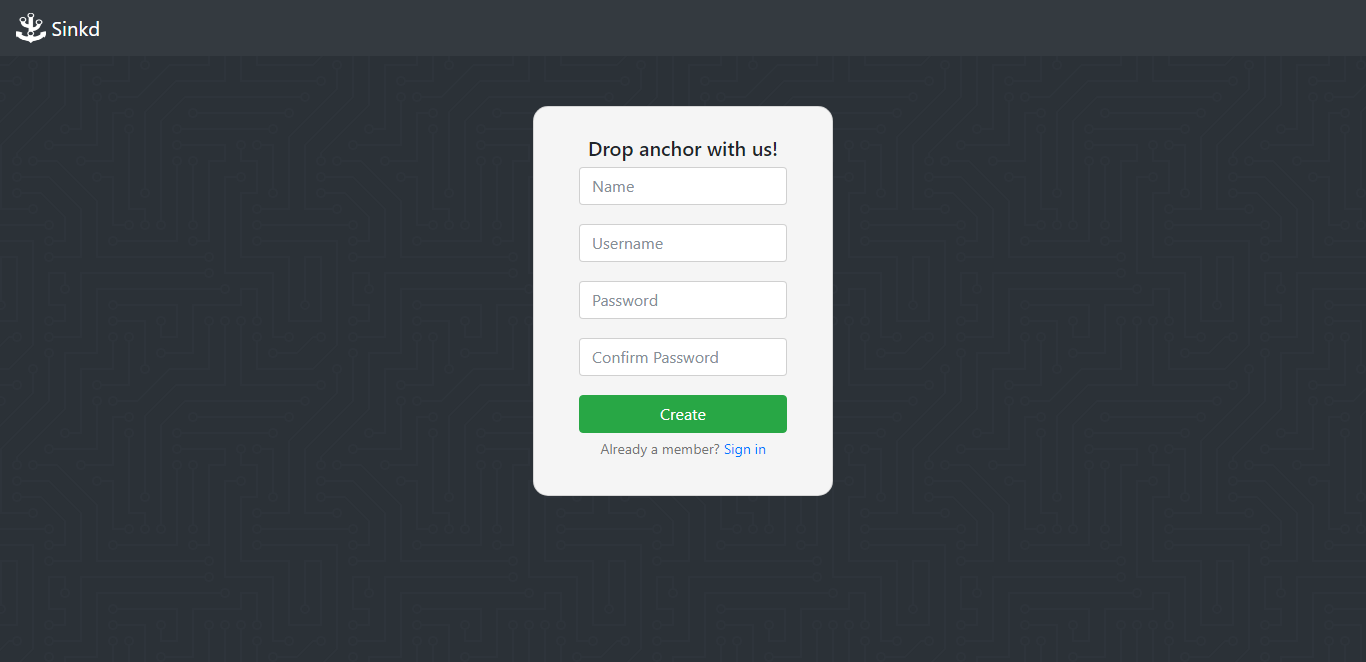
Account Management

1. On top right of the page, click on the button with the gear icon to manage account.
   1. To *edit* password, click on the green button named **Edit Password**.
   2. To *edit* display name, click on the green button named **Edit Name**.
   3. To *delete* account, click on the red button named **Delete Account**.
   4. To *logout*, click on the blue button named **Logout**.

[ Web Pages & Interrelationships ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Page [1] - Account Creation (signup.php)

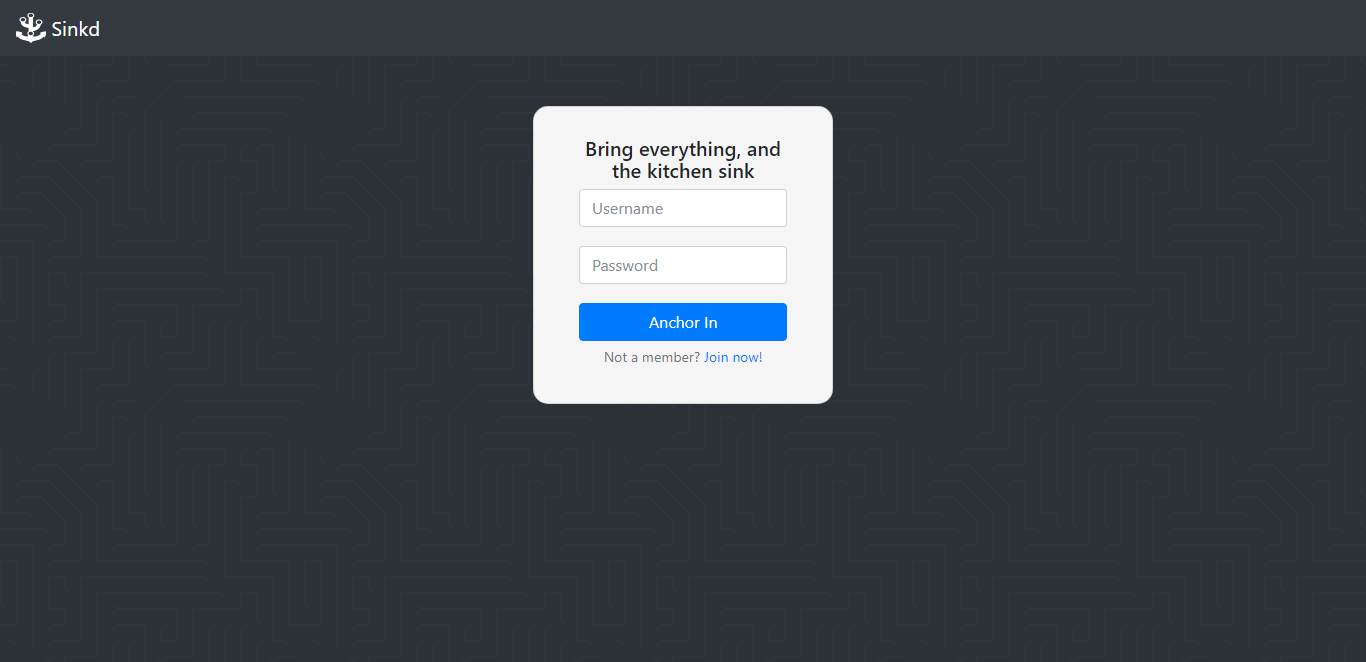
* Web URL: <http://cis444.cs.csusm.edu/group_A/sinkd/signup.php>



This page is used to create an account in the website along with the functionality to be redirected to login to the website. The information that the user must enter to create a new account is a name, username and password. Once the user confirms their password and creates their account, they are redirected to the main page. If the user tries to create an account that already exists, the website displays an error message.

Page [2] - Login (login.php)

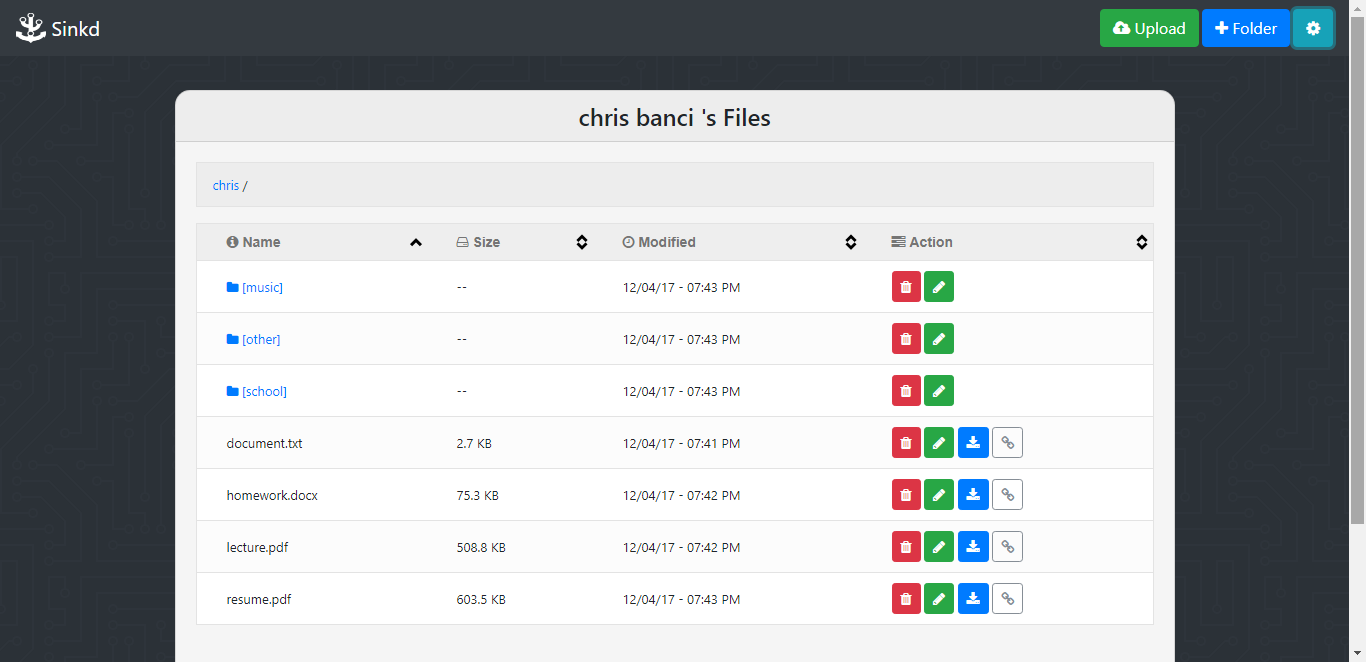
* Web URL: <http://cis444.cs.csusm.edu/group_A/sinkd/login.php>



This page is used to login to the website securely along with the functionality to be redirected to a create new account if the user does already have one. After the user logs in with the correct credentials, they are redirected to the main page. If the user tries to login to the website with an invalid username or password, the website displays an error message. Furthermore, if the user is already logged in and tries to enter this page, it redirects the user back to the main page.

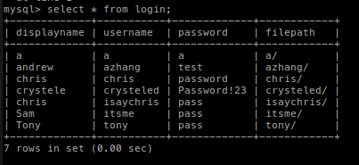
Page [3] - Main (main.php)

* Web URL: <http://cis444.cs.csusm.edu/group_A/sinkd/main.php>



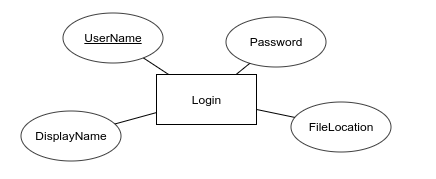
This page is used to display the user's files in a sortable table. The page refreshes dynamically upon uploading, renaming and deletion of files. We value user experience, and the best way to express this is through the use of a single page. Each file/folder listed in the table has the actions of deletion, editing filename, downloading, and sharing. When a user performs any of these action, an alert message is displayed on top of the table. The alert varies from success, warning, and error messages. On top of the table, is the user's current location within the filesystem hierarchy. As the user traverses through the filesystem, it gets updated accordingly. The user can use this to traverse back to a previous directory. If the user renames a file or folder to that of an existing one, it is renamed with an iterator. Likewise, the same thing occurs when the user tries to upload a file to that of an existing file name. Error handling is implemented to ensure the users a smooth experience. Besides the management of files, the user can also manage their account. The user has the options of being able to change their display name, change their password, delete their account, and logout from the website.

[Database Design & Implementation ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

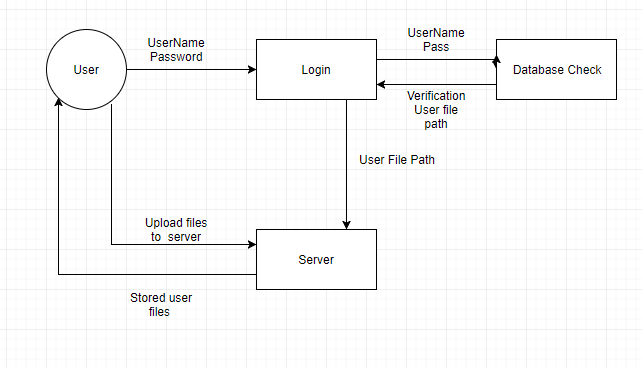


The database for our project is primarily used for creating new accounts and logging into the website. Our database contains a single table that has the attributes of an account's username, password, display name and its file path in the server. Other interactions with the database includes changing the display name and password of the user and the deletion of their account within the account settings of our website.

Entity Relationship Diagram



Information Flow Chart



[Web Design Issues & Future Improvements]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problems & Difficulties

Some issues/difficulties that we encountered and resolved had to do with creating folders. As part of testing, a user found a problem. After creating and naming a folder, the system alerted of successful folder creation, but the folder didn’t show up. Same user reported some buttons weren’t working properly and not responding. Another user reported the share feature had some difficulties, and wouldn’t load. These problems were resolved shortly after through constant debugging of our website. An issue that we could not resolve is changing the max upload file size limit through php. Due to limitations set by the server, the max upload file size is 2MB. This cannot be changed without having the correct privileges on the server. Another issue we could not resolve was displaying files recursively in all directories onto a tree structure using a jquery plugin called jsTree. Once we figured it was too difficult to accomplish, we figured that using a sortable table would be a much better option than using a tree structure for displaying files. If we used the jsTree, the file name would be the only information that would have been show to the user. Using sortable table however, besides just the filename, the modified time and size could be shown. On top of that, the filenames could be sorted in ascending or descending order. In a sense, we have resolved this issue with a better solution.

Possible Enhancements

Some possible future improvements and enhancements we can consider when maintaining the website will be making the site look more professional, and changing the look based on seasons but besides aesthetics, the site does what it's supposed to do and it does it well. Very user friendly and we will keep maintaining it to stay that way and make sure no issues appear. Maybe later on we could add a blog on our site where users can give feedback and post their messages on our main page. Could also add a search bar to search for any specific file or folder. Sorting options to sort files by format or add account verification to verify accounts upon registering via email. Another possible enhancement would be adding favorites. This way a user could mark a file or folder as their favorite to have quick access to them. Have an admin user with strong permissions to maintain and modify site instead of connecting to the database each time. Lastly we could implement a stronger security system. Some implementations are use of strong passwords. One example is to hash the passwords in the database. This way in case of a hack, passwords are not shown in plain text. Another example is to require certain length on each password as well as some numbers or special characters.

[ Web Implementations ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

When the user creates an account, the information is inserted in the the database and session data is set. The user is then redirected to the main page where they can manage their files. If the user has already has created an account and wants to login, the login credentials are compared with database. If they match, session variable is set and the user is redirected to the main page; if they don't, they are restricted from entering.

When the user enters the main page, the system first checks to the user has session data set. If the user does not have session data set, they are redirected to the main page. This ensures that the user cannot enter the main page if they are not logged in. As the main page is loading, the system calls a function that makes an ajax call to the server to returns a list of the user's files and its information in their root directory within an associative array. Then the associative array is parsed into rows of table data and buttons with events are then appended to them. These buttons have unique data attributes that contain the file paths of the files. The rows of data are then added dynamically onto the sortable table. The table will only show the files of the user's current directory.

When the user changes directories, the same process is performed as described above to display their files. Alongside that, to keep track of the current user's location within the file system system hierarchy, a file path navigator is displayed on top of the the table. When a user goes up a directory, it pushes the directory path into a stack. When a user goes down a directory, it pops directory paths from the stack. This ensures that the user can traverse through their directories smoothly.

File management actions such as upload files, creating folders, deleting, renaming, and downloading all are done through ajax calls to the server. Through ajax calls, it is possible to send requests to the server with php code and get a response back without having to refresh the page. All of these actions have specific php server commands that allows our website to manage the users files. Besides the management of the user's files within the server, changes to the user's account settings interact with the database using the same principle of using ajax calls.

Once the user accesses their cloud files and selects them to upload into their desired folder or adjust them as they please, they then can proceed to access them from any other device when using our webpage.

[ Individual Contributions ]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Andrew Zhang

* Account settings, Database design, Design, Debugging

Brandon Tilley

* Documentation,Web Organization, Maintenance, Testing

Chris Banci

* Frontend developer of the Login/Signup page, Javascript/UI on Main page, UI Design, Debugging

Crystele Dierickx

* Documentation, Presentation, Design, Testing

David Bright

* Project Manager, Project Idea, Concept Design, Logo/Slogan Designer, Established Navbar commonality across pages, Backend developer of the Main/Login/Signup page, Database Design, PHP Debugging

Sam Doroudi

* Documentation, Presentation, Programming front-end and logout