# DropZone - A Dropbox Replica

Saheed Ahmed Abdulrazaq - 3061874

Griffith College Dublin

May 29, 2024

Table of Contents

## Brief

In this assignment I've been tasked to build a simplified replica of Dropbox cloud service using Google Cloud Platform. I've also been required to develop storage available to users where they can create arbitrary directory structure and can upload files and download files from the service. I've also been asked to develop access to basic file sharing between accounts.

*Keywords*: *Buckets, Blobs, Cloud Storage.*

DropZone - A Dropbox Replica

Dropzone is a cloud-based storage solution inspired by dropbox which provides users with file and folders storage. It's utilizes Google Cloud Platform services such as Firebase authentication for user authentication, Firestore database for storing of user data and Google cloud storage for file system like storage.

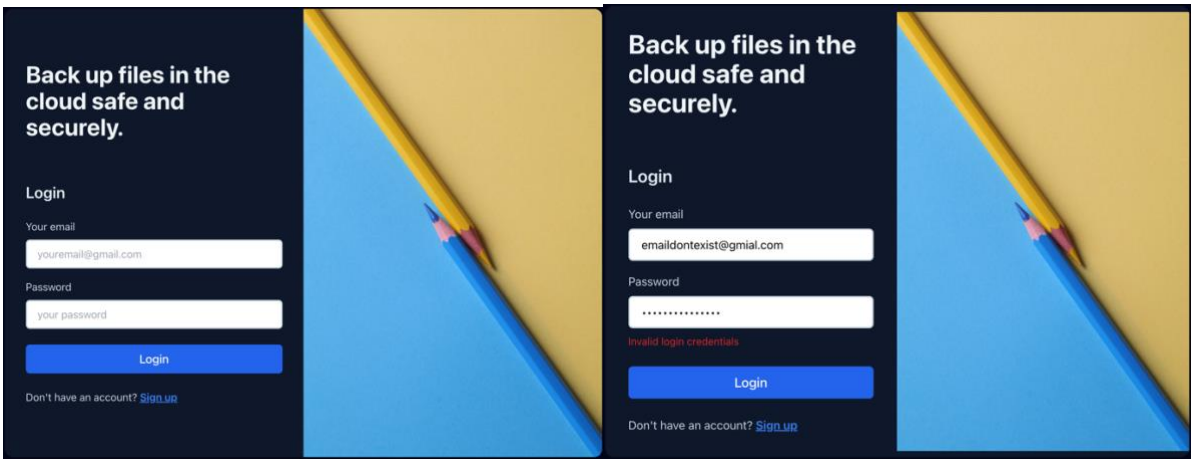
Dropzone closely resembles Dropbox in both user interface and usage semantics. Its components will be discussed in detail in a later section of this documentation.

## User Interface

The design of the user interface for Dropzone was informed by the original Dropbox Application. This choice reflects the established user experience patterns employed by Dropbox, which are well suited to the task of file management and nested navigation. However, due to the focus on the application logic and functionality development, a minimalist user interface approach was adopted. This resulted in a two-view structure, with dynamic switching of view state from login to main view depending on the user state.

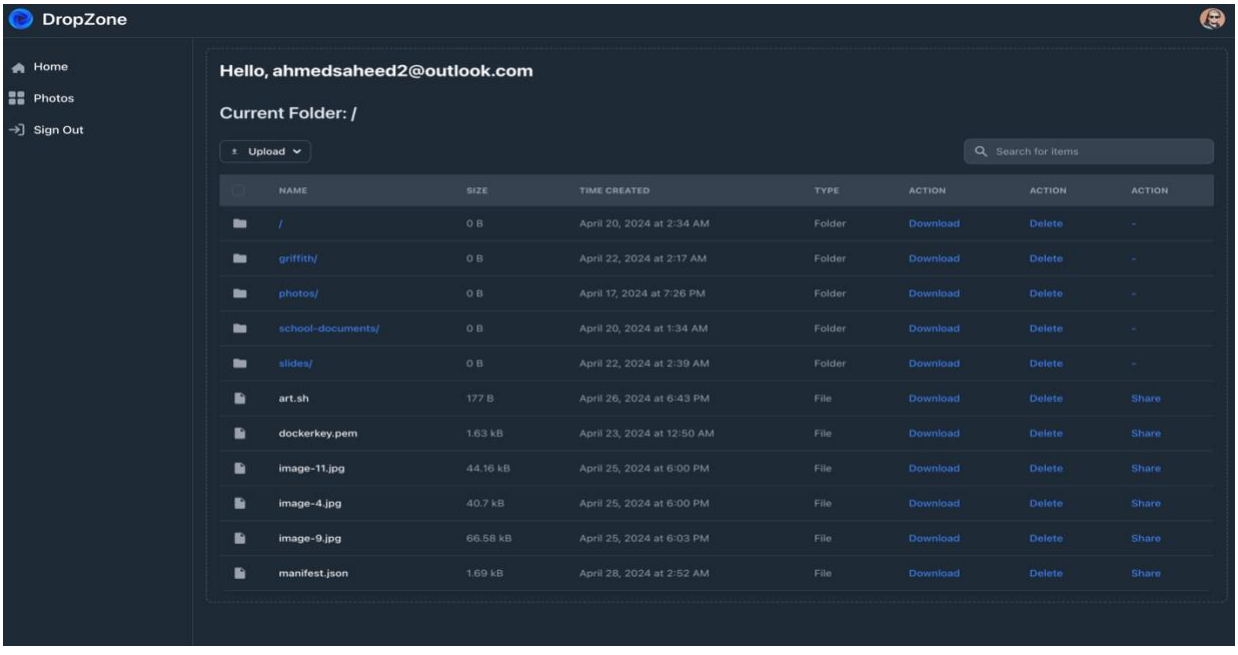
### Login View[1]

The login interface was designed to priorities simplicity and user convenience. It incorporates an email and password login form. Within this login form there a hidden error display element which becomes visible when an error is encountered during either the login in or sign up flow. The view was design with responsiveness in mind, i.e. when viewed on mobile devices or larger screens they easily adopt and become responsive irrespective of the screen size in which its being viewed. This view is displayed to new users upon creating an account or existing users when they need to login to their accounts.




**Main View2**

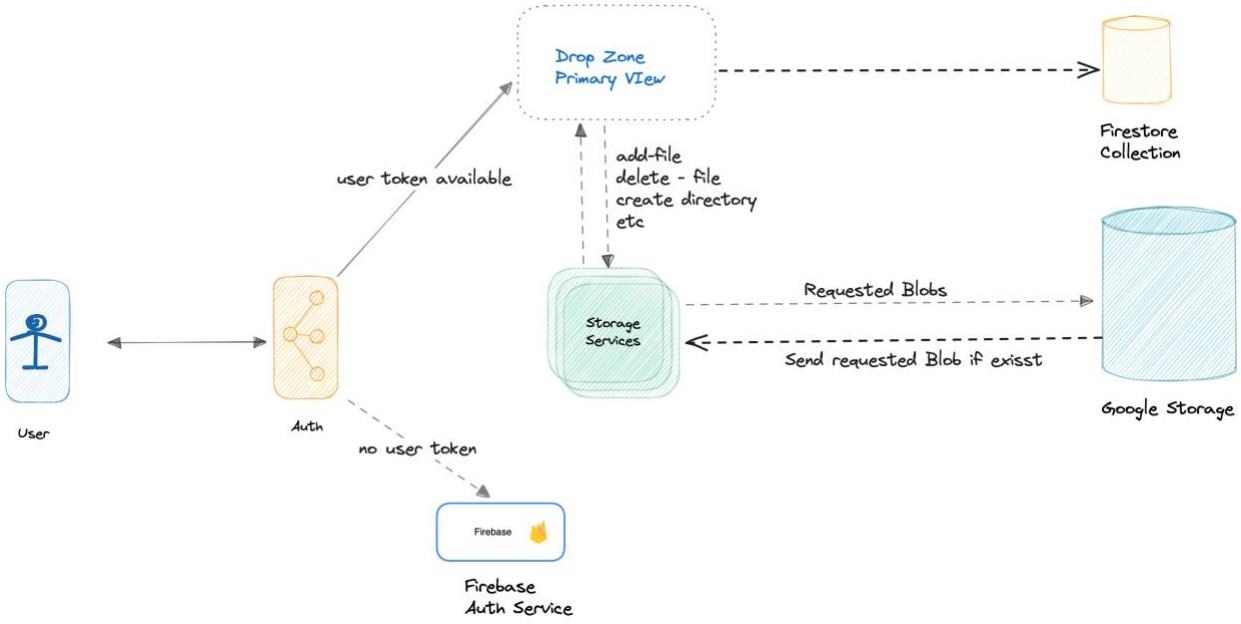
The primary user interface for logged in users centers around a left aligned sidebar menu. The menu is initially focused on the home option. This home option reviles the central content container which presents a table view of all directories and files within the current directory which is typically the root ( / ). An action button positioned at the top of the table which has a dropdown option for either uploading a file or creating directory. Additionally, individual action buttons are provided for each row, enabling users to download, delete and share specific directories or file.



## Architecture Overview

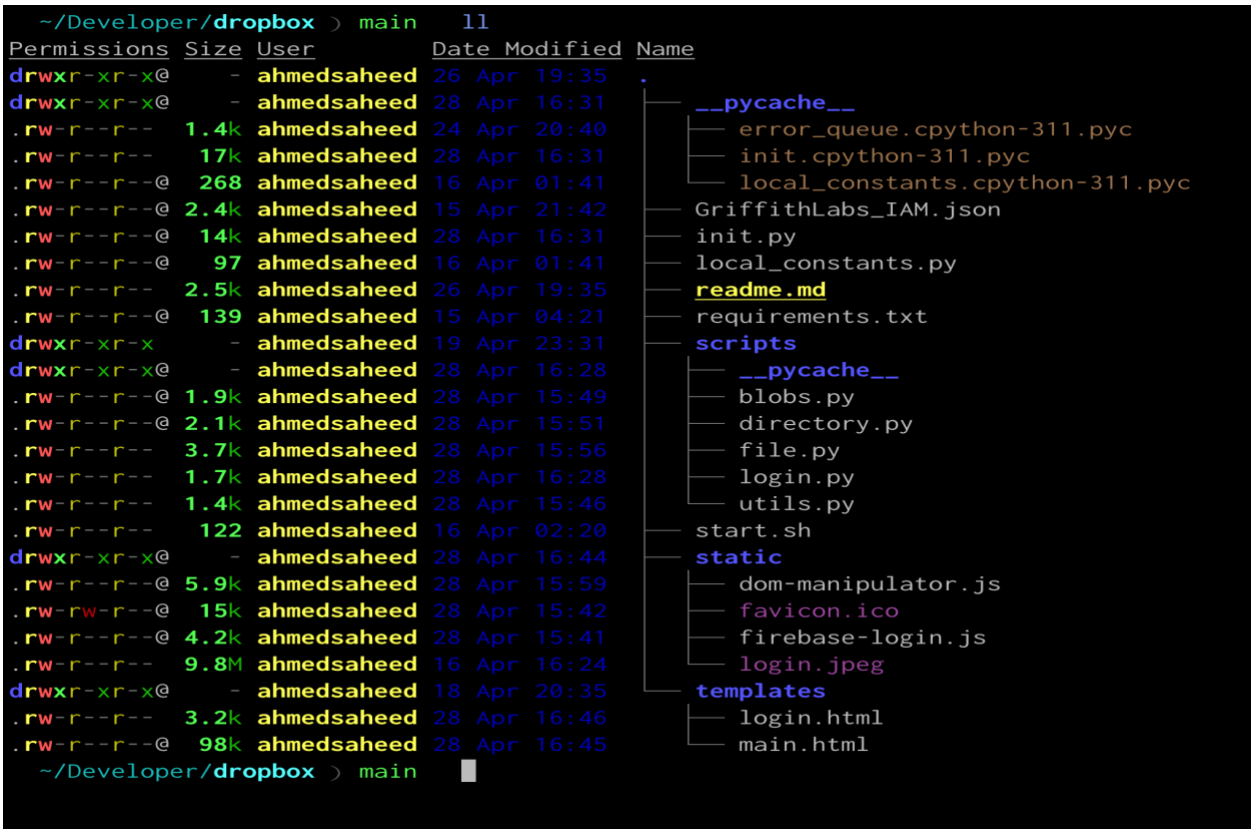
The application uses python FastAPI library which is a fast high performance web framework for building APIs. Within the application, API routes are specified in python code and these routes are usually fetched by the html templates rendered at the moment a request is made . Some routes are static while others are dynamic based on either the data passed via a form or a URL query parameter.

A high-level overview of the application structure looks something like this:



## Directory Structure & Method Overview

Here, I'll show what DropZone directory structure looks like. I'd also discuss every method used and how they all fit together on a high-level perspective.




From the above diagram, you'd notice a list of directories and file, below is a breakdown of each directory and single files. The __pycache__ directory can be ignored.

**Local_constants.py File**

This file constants google cloud specific constant variables used across the project scope.

**Requirement.txt File**

This is a file where project specific requirements are listed alongside the version needed.

**GriffithLabs_IAM.json File**

This is a file where all Google Cloud credentials for this project is specified.

**Start.sh File**

A simple bash script used to get the application up and running with minimal effort.

**Init.py File**

This is the entry point to the DropZone application. In this file all the FastAPI routes and their necessary handlers are defined. Below is a list of the routes and how they're used.

| Route | Description | Request Type |
|---|---|---|
| Root ( / ) | Root directory of the application. It's responsible for either displaying the login page or main page depending on the user's state. Here we invoke the blob_list, check_for_duplicate_file functions. | GET |
| /get-subdirectory | Route for navigating into a sub directory. Has same functionality as root, but slightly adjusted for sub directories | [GET, POST] |
| /add-directory | Route for handling add-directory request sent from the client. It invokes the add_directory functions and upload it if the directory doesn't exist already. | POST |
| /upload-file | This route is for handling upload-file request. We check for an overwrite flag sent from the client. If any is available, we overwrite any matching file with the uploaded one otherwise we just upload the file. It invokes the add_file function | POST |
| /download-file | Route for handling download file request. It responds with a StreamingFileResponse which downloads the file to the user's local machine. It invokes the download_file function | POST |
| /delete-file | Route for handling a delete-file request. It deletes the file and redirects to where the client was before. It invokes the delete_file function. | POST |
| /delete-directory | Route responsible for handling delete-directory request. Its invokes the should_delete_directory function which asserts the directory is empty before proceeding to delete. | POST |

| | | |
|---|---|---|
| /share-file | Route for handling a share-file request. It invokes the copy_file function which copies the specified file of the current user to the chosen recipient. | POST |
| /get-photos | Route for handling rh get-photos request. Uses the get_photos method to retrieve all photos the user stored. | POST |

**Scripts Directory**

This is where we house python files which consist of functions used to interact with Google Cloud Storage. The files in this directory includes:

1. **Blob.py**: This file contains simple methods used for interacting with a logged in users bucket. And this is a list of functions within this file.

| Function | Description |
|---|---|
| blob_list(prefix, uid) | Returns all blobs in the root directory of the logged in user using their uid. Has return type of HTTPIterator |
| get_sub_blob_list(uid, sub_dir_path) | Returns all blobs in a specified sub directory of the logged in user. Returns type is of HTTPIterator |
| download_blob(download_path) | Downloads a blob as bytes using the specified download_path. Returns type of bytes |
| get_photos(uid) | Retrieves all blobs with jpeg, jpg and png format in the user's bucket. Makes these files publicly accessible and returns a list containing the blob path and the public accessible url which the client users to display images. |

2. **Directory.py:** Here we have function for directory specific blobs. These are normally identified by blob names ending with a /. The functions in this file includes:

| Function | Description |
|---|---|
| add_directory(dir_name, uid) | Creates a new directory/bucket in the logged in user base bucket. Uses the dir_name to create this new directory |
| dir_exist(dir_name, uid) - Bool | Check whether a directory exist in the user's bucket. Returns a boolean |
| delete_directory(dir_path) | Deletes a directory from a bucket if it exists. |

| | |
|---|---|
| should_delete_dir(dir_path) | Determines whether a directory should be deleted from the logged in user bucket. If it's a non-empty directory returns false otherwise returns true. |
| create_home_dir_if_necessary( Uid, file_blob, directory_blob) | Creates an empty default root directory (/) if it is a new user or if no file or directory exist a user. |

3. **File.py:** Here we have function for file specific blobs. These are normally identified by blob names not ending with a /. The functions in this file includes:

| Function | Description |
|---|---|
| add_file(file, prefix, uid) | Uploads a file to a logged in user base bucket if prefix is empty. Otherwise upload to the sub directory matching the provided prefix |
| delete_file(file_path) | Deletes a blob from the user's bucket matching the specified file_path if it exist |
| check_for_duplicate_file(file_list) | Detect if there's a duplicate file in the user's current bucket. It iterates through the file_list of provided checking for matching md5hash properties in each file |
| copy_file(user_token, source_path, recipient_email) | Copies the specified file from the logged in user to another user base bucket which matches the recipient_email. |
| file_exist(file, prefix, uid) | Determines whether a file exist in the prefix storage bucket. |

4. **Login.py:** This file contains users and login specific logic.

| Function | Description |
|---|---|
| get_user(user_token) | Returns the current users data from the firestore database using to token stored on the client cookie. |
| get_all_users(user_token) | Returns a list of all users available in the firestore database. It's mainly used with copy_file function to share files with other users. |
| validate_firebase_token(id_token) | Asserts the validity of the firebase token. |

5. **Utils.py:** This is a helper file where helper functions are defined. Below is a list of all helper function and their description.

| Function | Description |
|---|---|
| extract_relative_path(path) | Extract the relative path from a full path |

| | |
|---|---|
| should_add_to_list(blob_name) | Determines whether a blob should be added to the base list of blobs. This returns false for nested directories and files |
| extract_file_name(file_path) | Extracts the file name from the files full path |

**Static Directory**

This directory contains static files that are used infrequently. These files include:

- Firebase-login.js: This JavaScript file is used for interacting with Firebase Authentication to log in, sign up, and sign out users.
- Dom-manipulator.js: This file handles some client-side DOM manipulation tasks, such as converting timestamps into formats that are easier for humans to read and converting byte-based file sizes into more user-friendly formats.
- Favicon.ico: This file contains the small icon that appears in the browser tab or window title bar.
- Login.png: This image is used on the login screen.

**Template Directory**

This directory contains html template files which are used by Jinja2 and FastAPI to render the app. These files include:

- Login.html: The template displayed when a user token is available to the app and the user need to sign in to generate one
- Main.html: This is the main app view which contains all the logged in users files and folders

  Theses files are style using tailwind-css so there are no style.css file in this app.

# References

Google. (n.d.-a). *Copy, rename, and move objects | cloud storage | google cloud*. Google. https://cloud.google.com/storage/docs/copying-renaming-moving-objects

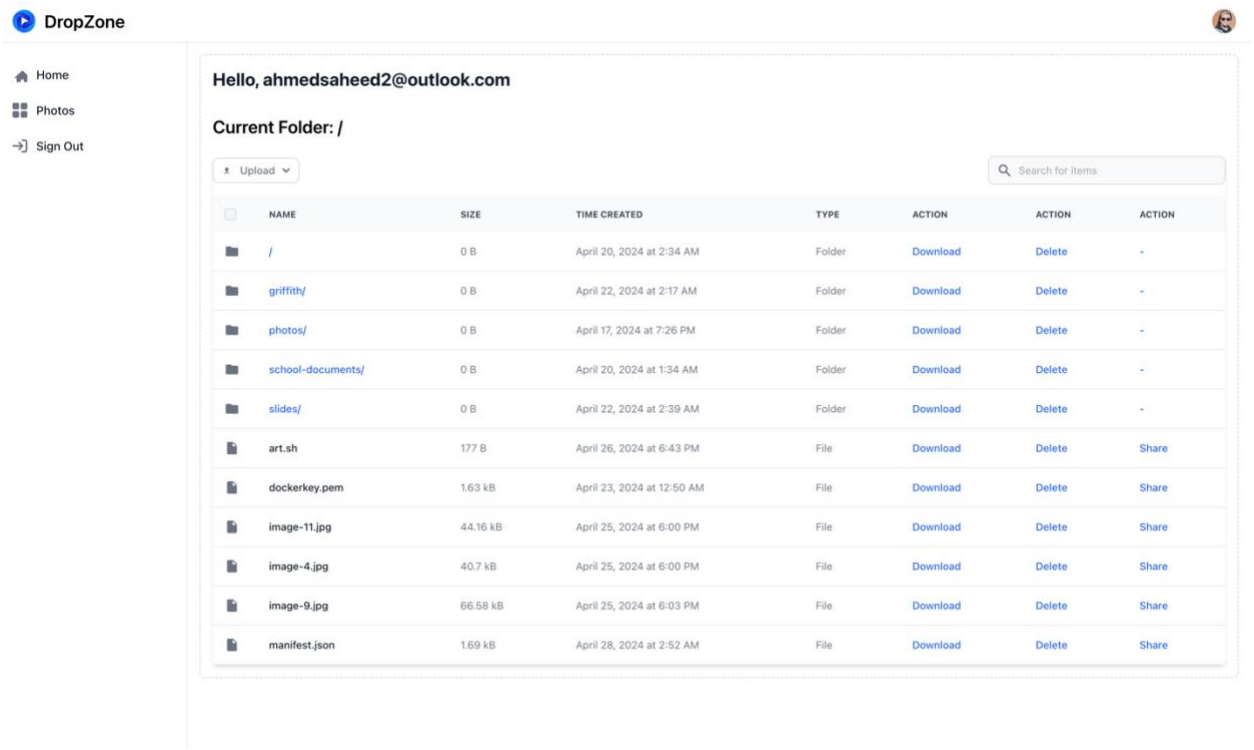
Google. (n.d.-b). *Get started with Firebase Authentication on websites*. Google. https://firebase.google.com/docs/auth/web/start

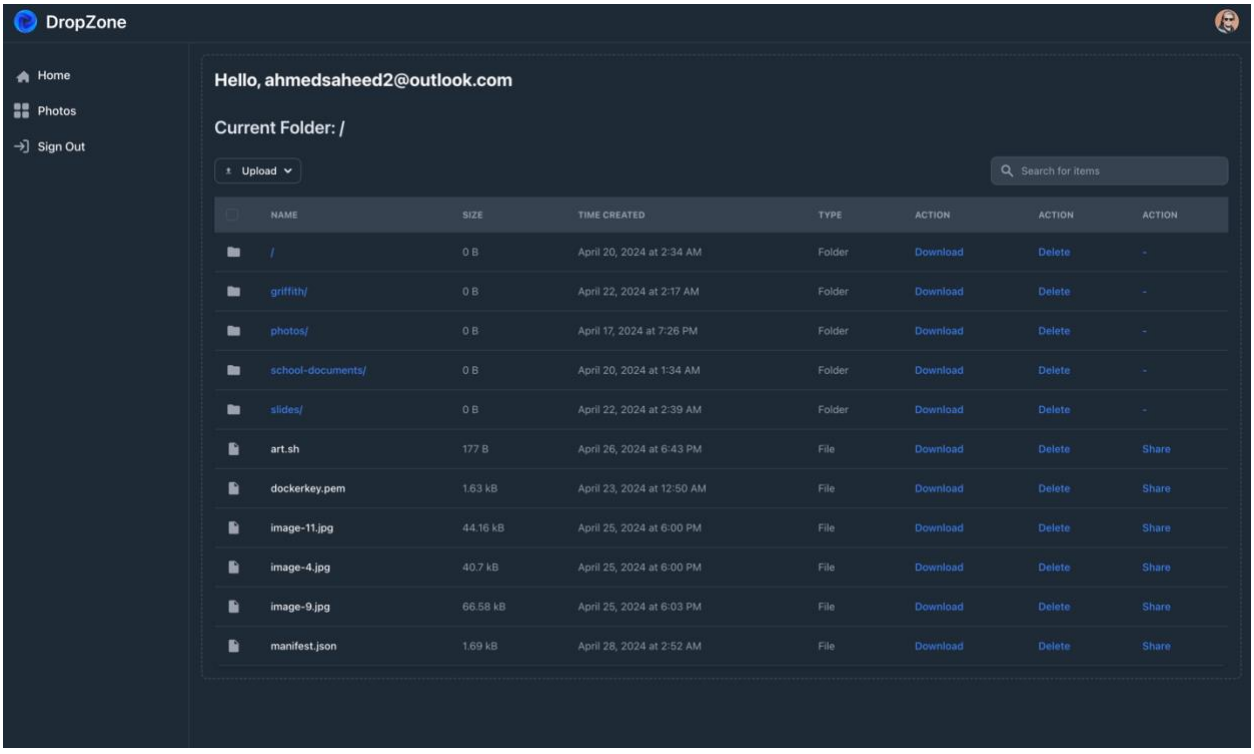
Google. (n.d.-c). *Product overview of cloud storage | google cloud*. Google. https://cloud.google.com/storage/docs/introduction

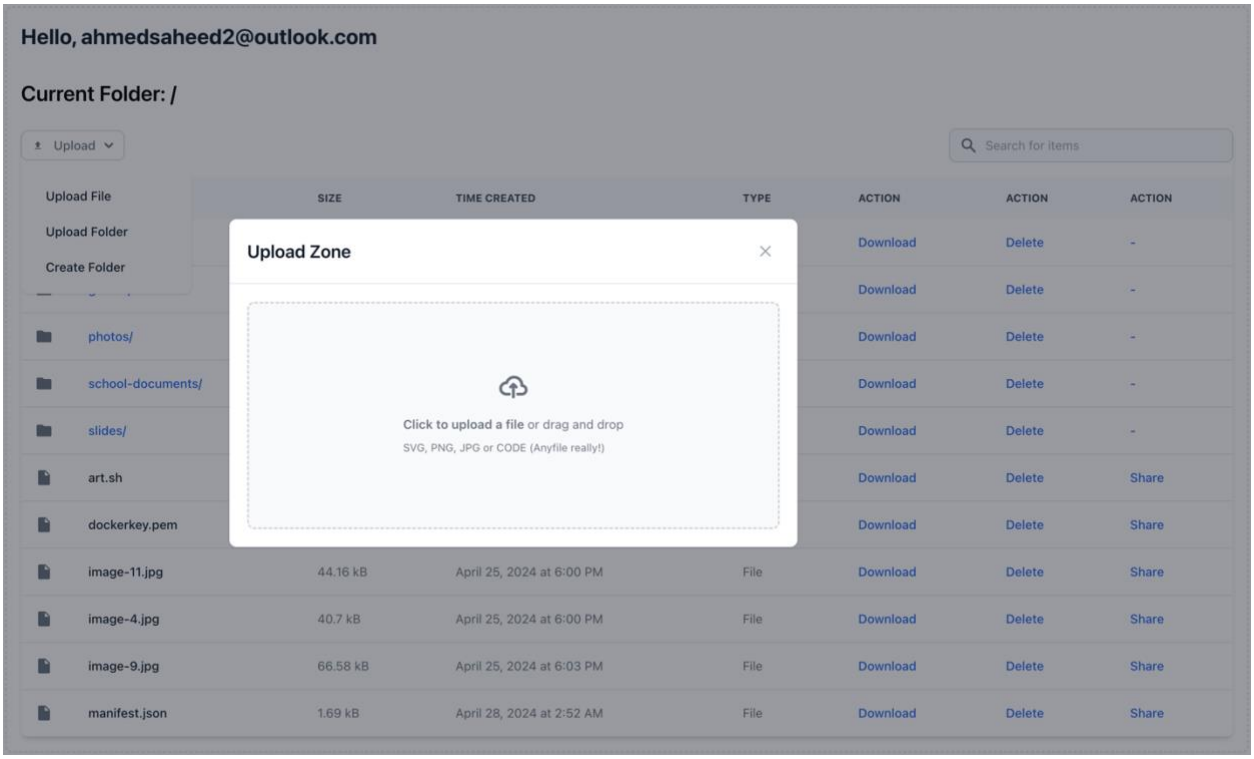
Google. (n.d.-d). *Python client library | google cloud*. Google. https://cloud.google.com/python/docs/reference/firestore/latest

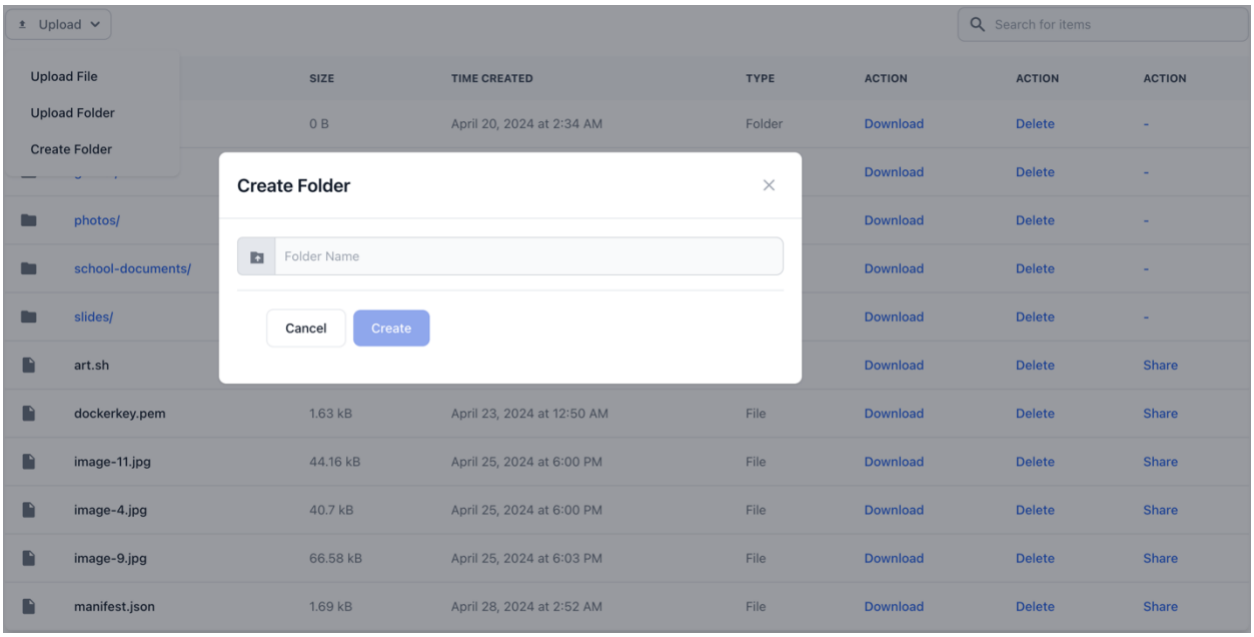
*Tailwind Styling Documentation*. Tailwind UI - Official Tailwind CSS Components & Templates. (n.d.). https://tailwindui.com/documentation

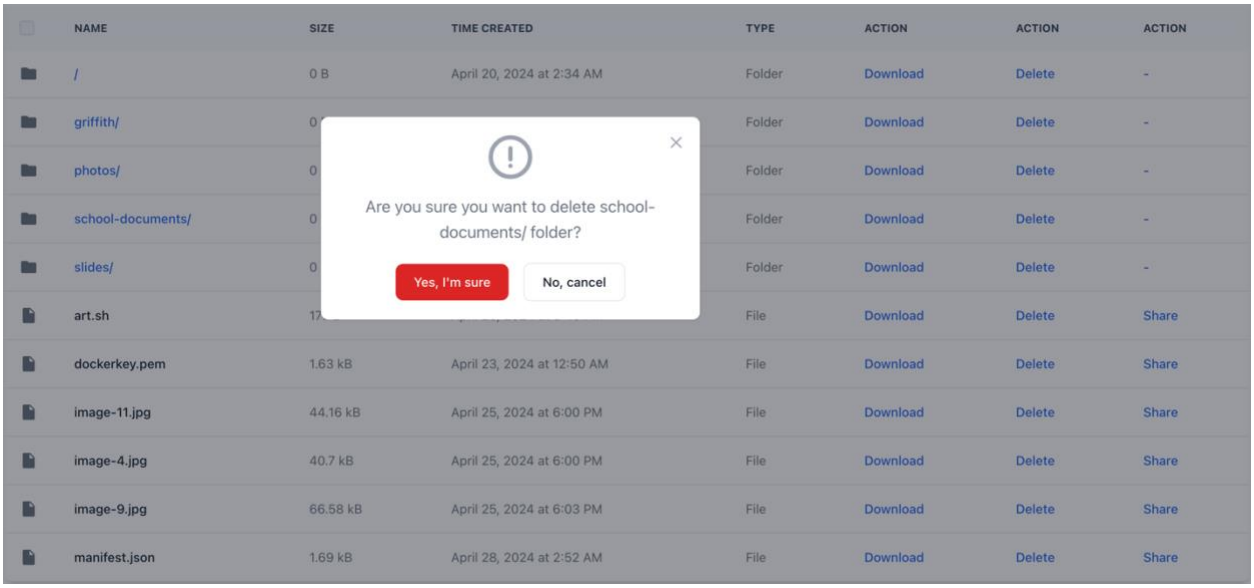
Figures

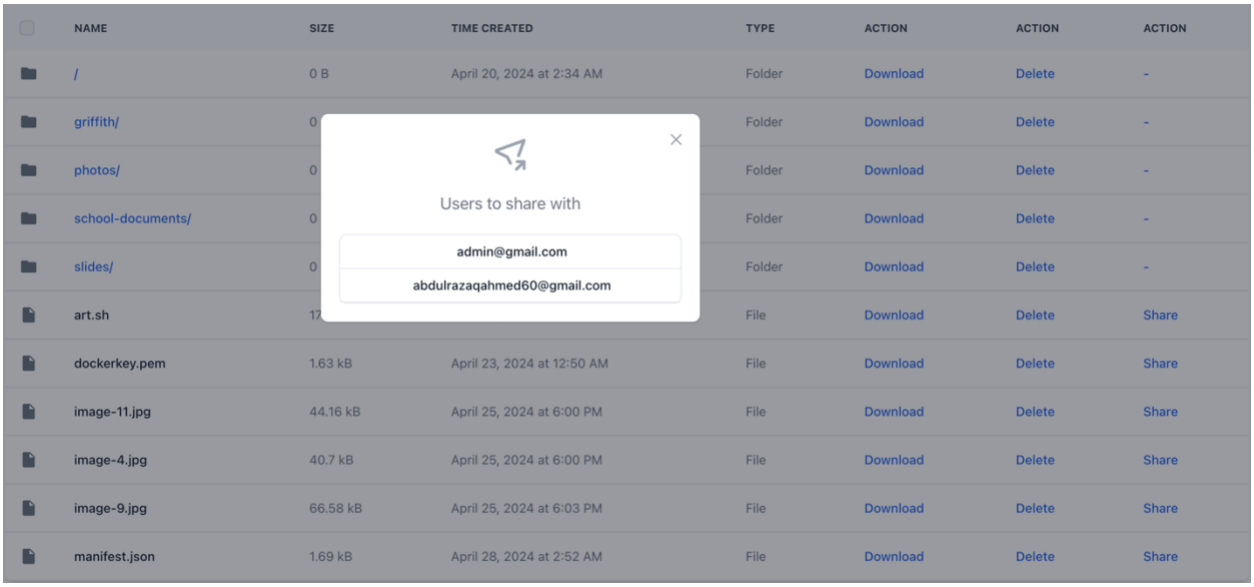

DropZone
Home
Photos
Sign Out
Hello, ahmedsaheed2@outlook.com
Current Folder: /
Upload
Search for items
NAME SIZE TIME CREATED TYPE ACTION ACTION ACTION
/ 0 B April 20, 2024 at 2:34 AM Folder Download Delete -
griffith/ 0 B April 22, 2024 at 2:17 AM Folder Download Delete -
photos/ 0 B April 17, 2024 at 7:26 PM Folder Download Delete -
school-documents/ 0 B April 20, 2024 at 1:34 AM Folder Download Delete -
slides/ 0 B April 22, 2024 at 2:39 AM Folder Download Delete -
art.sh 177 B April 26, 2024 at 6:43 PM File Download Delete Share
dockerkey.pem 1.63 kB April 23, 2024 at 12:50 AM File Download Delete Share
image-11.jpg 44.16 kB April 25, 2024 at 6:00 PM File Download Delete Share
image-4.jpg 40.7 kB April 25, 2024 at 6:00 PM File Download Delete Share
image-9.jpg 66.58 kB April 25, 2024 at 6:03 PM File Download Delete Share
manifest.json 1.69 kB April 28, 2024 at 2:52 AM File Download Delete Share

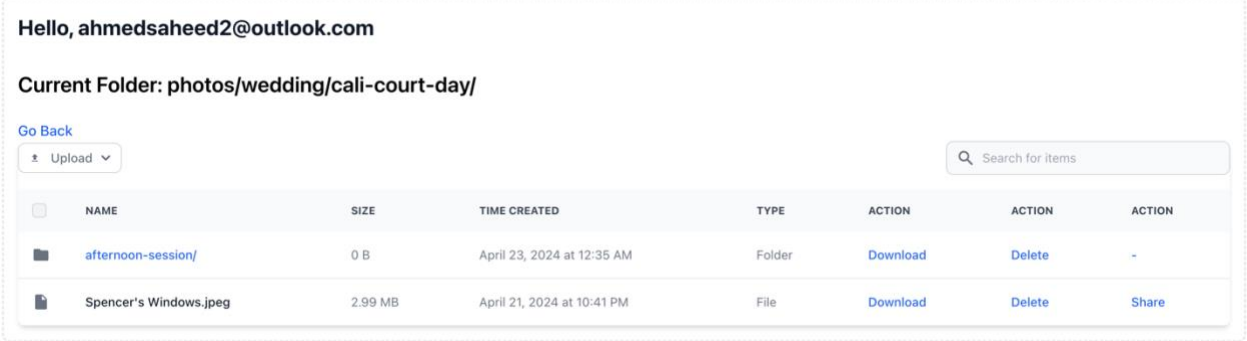

DropZone
Home
Photos
Sign Out
Hello, ahmedsaheed2@outlook.com
Current Folder: /
Upload
Search for items
NAME SIZE TIME CREATED TYPE ACTION ACTION ACTION
/ 0 B April 20, 2024 at 2:34 AM Folder Download Delete -
griffith/ 0 B April 22, 2024 at 2:17 AM Folder Download Delete -
photos/ 0 B April 17, 2024 at 7:26 PM Folder Download Delete -
school-documents/ 0 B April 20, 2024 at 1:34 AM Folder Download Delete -
slides/ 0 B April 22, 2024 at 2:39 AM Folder Download Delete -
art.sh 177 B April 26, 2024 at 6:43 PM File Download Delete Share
dockerkey.pem 1.63 kB April 23, 2024 at 12:50 AM File Download Delete Share
image-11.jpg 44.16 kB April 25, 2024 at 6:00 PM File Download Delete Share
image-4.jpg 40.7 kB April 25, 2024 at 6:00 PM File Download Delete Share
image-9.jpg 66.58 kB April 25, 2024 at 6:03 PM File Download Delete Share
manifest.json 1.69 kB April 28, 2024 at 2:52 AM File Download Delete Share


Hello, ahmedsaheed2@outlook.com
Current Folder: /
Upload
Search for items
Upload File
Upload Folder
Create Folder
Upload Zone
Click to upload a file or drag and drop
SVG, PNG, JPG or CODE (Anyfile really!)
SIZE
TIME CREATED
TYPE
ACTION
photos/
school-documents/
slides/
art.sh
dockerkey.pem
image-11.jpg 44.16 kB April 25, 2024 at 6:00 PM File
image-4.jpg 40.7 kB April 25, 2024 at 6:00 PM File
image-9.jpg 66.58 kB April 25, 2024 at 6:03 PM File
manifest.json 1.69 kB April 28, 2024 at 2:52 AM File
Download
Delete
Share


Upload
Search for items
Upload File
Upload Folder
Create Folder
Create Folder
Folder Name
Cancel
Create
SIZE
TIME CREATED
TYPE
ACTION
0 B April 20, 2024 at 2:34 AM Folder
photos/
school-documents/
slides/
art.sh
dockerkey.pem 1.63 kB April 23, 2024 at 12:50 AM File
image-11.jpg 44.16 kB April 25, 2024 at 6:00 PM File
image-4.jpg 40.7 kB April 25, 2024 at 6:00 PM File
image-9.jpg 66.58 kB April 25, 2024 at 6:03 PM File
manifest.json 1.69 kB April 28, 2024 at 2:52 AM File
Download
Delete
Share


NAME
SIZE
TIME CREATED
TYPE
ACTION
ACTION
ACTION
/
0 B
April 20, 2024 at 2:34 AM
Folder
Download
Delete
-
griffith/
Folder
Download
Delete
-
photos/
Folder
Download
Delete
-
school-documents/
Folder
Download
Delete
-
slides/
Folder
Download
Delete
-
art.sh
File
Download
Delete
Share
dockerkey.pem
1.63 kB
April 23, 2024 at 12:50 AM
File
Download
Delete
Share
image-11.jpg
44.16 kB
April 25, 2024 at 6:00 PM
File
Download
Delete
Share
image-4.jpg
40.7 kB
April 25, 2024 at 6:00 PM
File
Download
Delete
Share
image-9.jpg
66.58 kB
April 25, 2024 at 6:03 PM
File
Download
Delete
Share
manifest.json
1.69 kB
April 28, 2024 at 2:52 AM
File
Download
Delete
Share
Are you sure you want to delete school-documents/ folder?
Yes, I'm sure
No, cancel


NAME
SIZE
TIME CREATED
TYPE
ACTION
ACTION
ACTION
/
0 B
April 20, 2024 at 2:34 AM
Folder
Download
Delete
-
griffith/
Folder
Download
Delete
-
photos/
Folder
Download
Delete
-
school-documents/
Folder
Download
Delete
-
slides/
Folder
Download
Delete
-
art.sh
File
Download
Delete
Share
dockerkey.pem
1.63 kB
April 23, 2024 at 12:50 AM
File
Download
Delete
Share
image-11.jpg
44.16 kB
April 25, 2024 at 6:00 PM
File
Download
Delete
Share
image-4.jpg
40.7 kB
April 25, 2024 at 6:00 PM
File
Download
Delete
Share
image-9.jpg
66.58 kB
April 25, 2024 at 6:03 PM
File
Download
Delete
Share
manifest.json
1.69 kB
April 28, 2024 at 2:52 AM
File
Download
Delete
Share
Users to share with
admin@gmail.com
abdulrazaqahmed60@gmail.com


Hello, ahmedsaheed2@outlook.com
Current Folder: photos/wedding/cali-court-day/
Go Back
Upload
Search for items
NAME
SIZE
TIME CREATED
TYPE
ACTION
ACTION
ACTION
afternoon-session/
0 B
April 23, 2024 at 12:35 AM
Folder
Download
Delete
-
Spencer's Windows.jpeg
2.99 MB
April 21, 2024 at 10:41 PM
File
Download
Delete
Share