CSE 569: Cloud Computing

Phase-2 Deliverables

13th December 2022

Group Number: 7

Members

Abhimanyu Gupta(2019226)

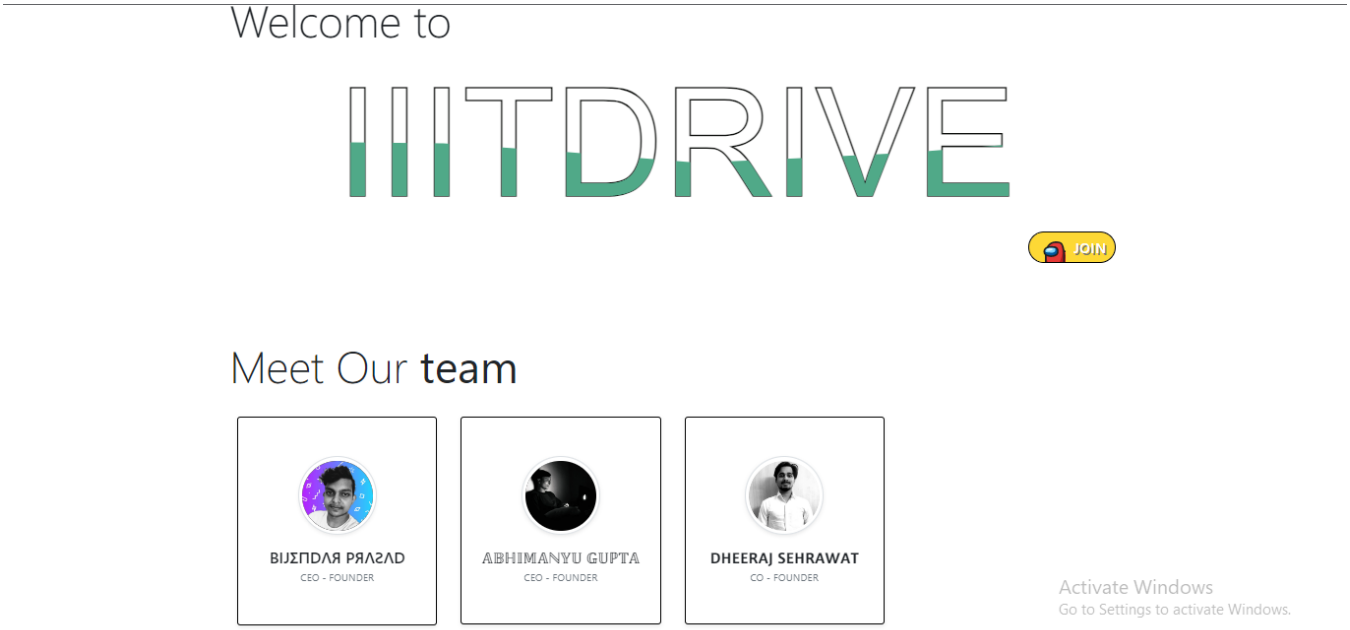
Bijendar Prasad(2019238)

Dheeraj Sehrawat (2019358)

Documentation

Github Repo Link: https://github.com/Findcoding/iiitdrive

We have implemented a Google Drive prototype service. For the front end, we have used HTML, CSS, and javascript; while for the back end, we have used Django. For implementing the cloud features, we are using AWS.



1. Functionality:

Below are some of the functionalities we have achieved

1) **User Registration and Login**

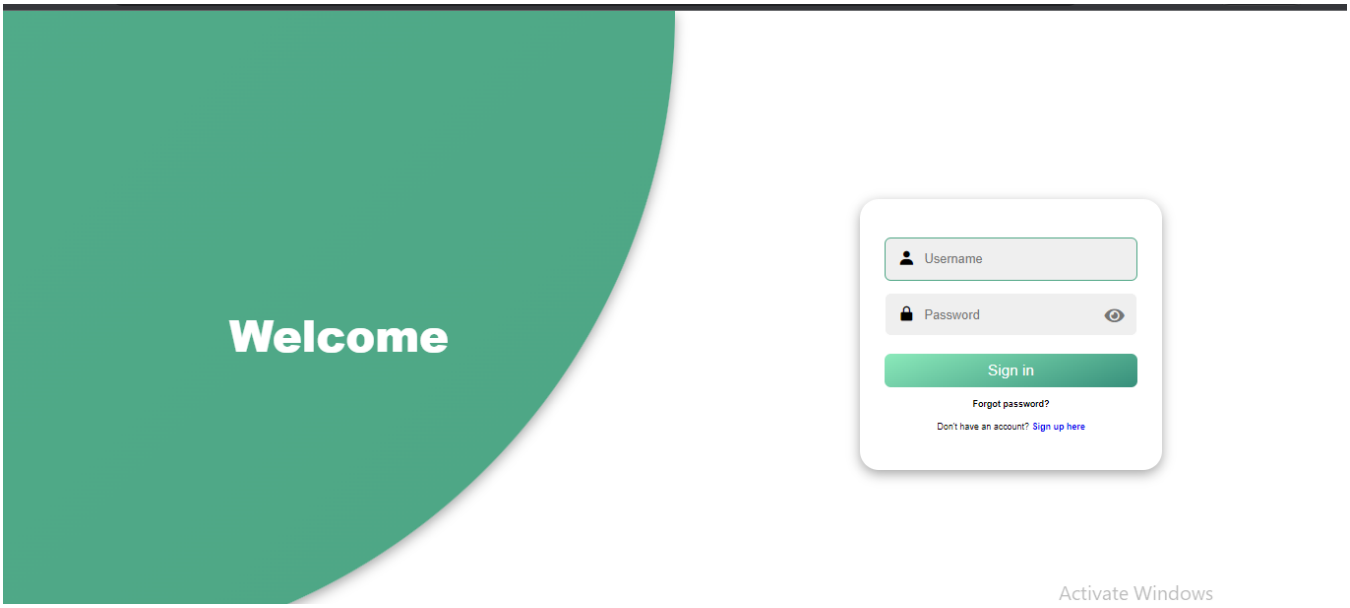
User can signup and login into our platform using their IIITD emails. Users are allowed to log in on the platform only after verifying their identity.

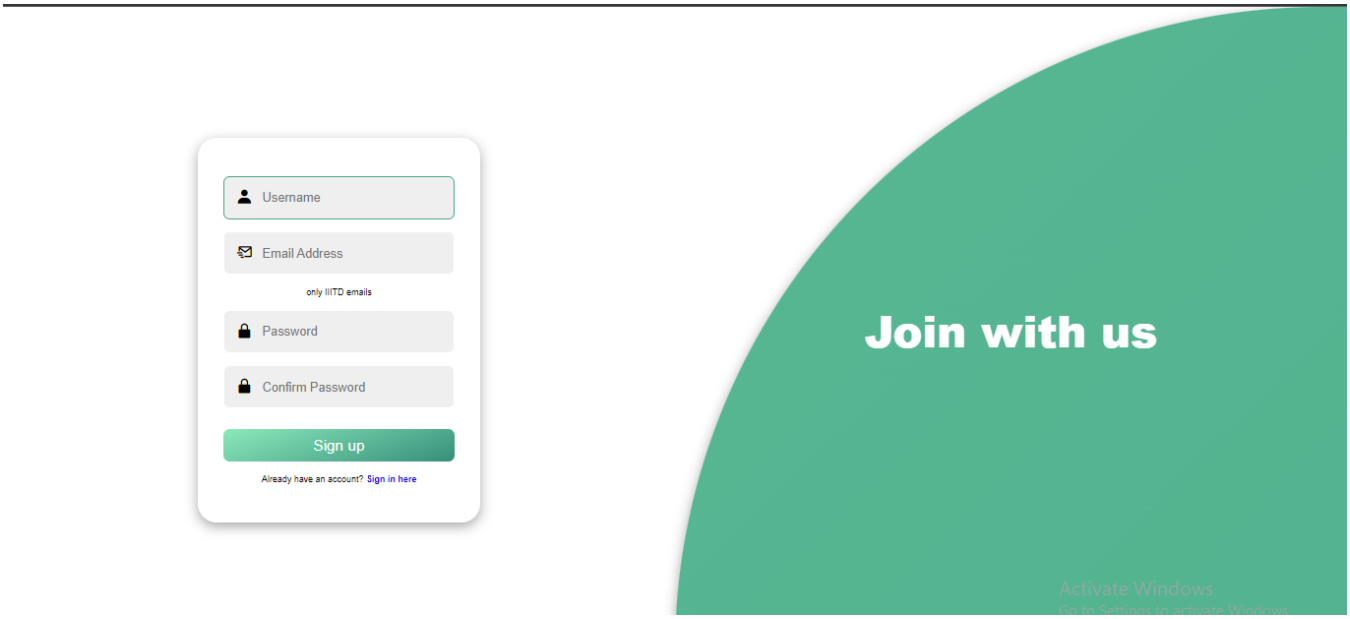
Implementation Details

a) A user table is created to store the user credentials

b) email_verified field is used to ensure that users only access the platform after verification.

c) Further, the is_active field is present, which allows for the soft deletion of users if required.



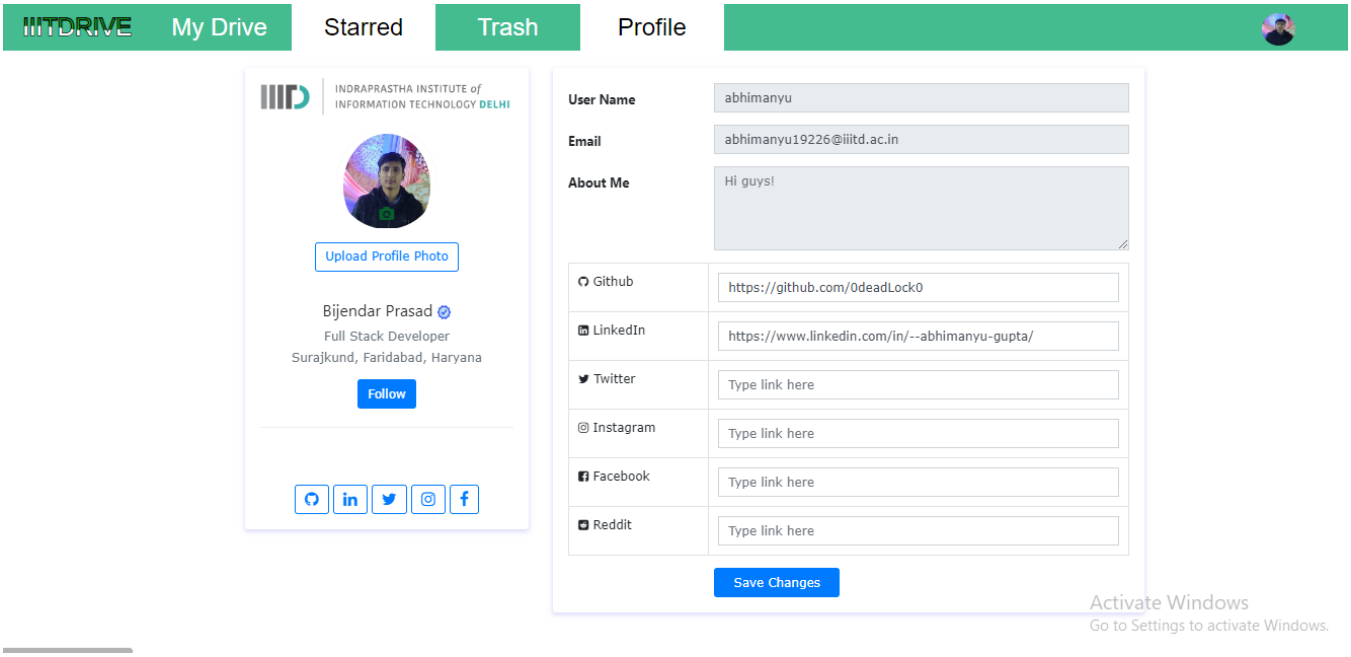


2) User Profile

User can customize their appearance on the platform by fine tunning their profiles. Links to their various socials can be provided by the user to be seen by other users.

Implementation Details

a) User information like a profile picture and about me is stored with other user details in a table. Further, social media links of platforms like GitHub and LinkedIn are also stored in a Social table.

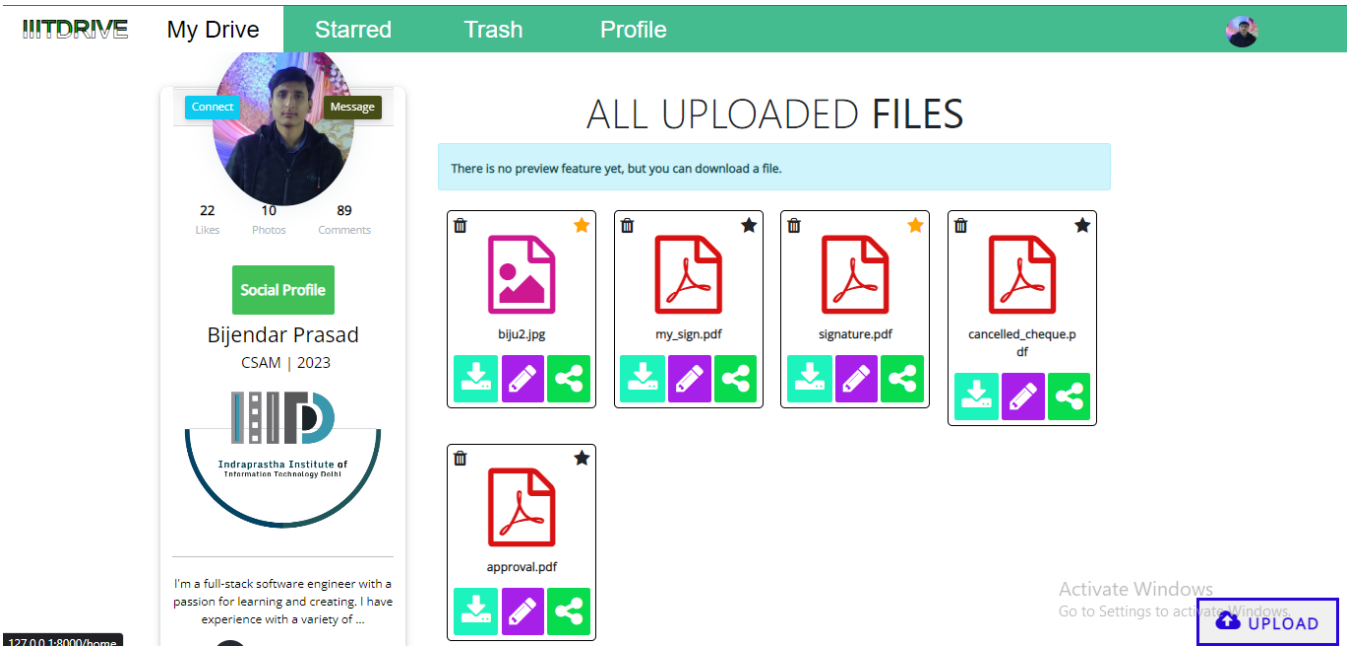


3) File Upload

Users can upload files with any extension and access them later from anywhere having an internet connection.

Implementation Details

a) A table dedicated to storing files is created with a foreign key relationship with the owner of that file. Further, metadata such as display file names and shared users are stored.

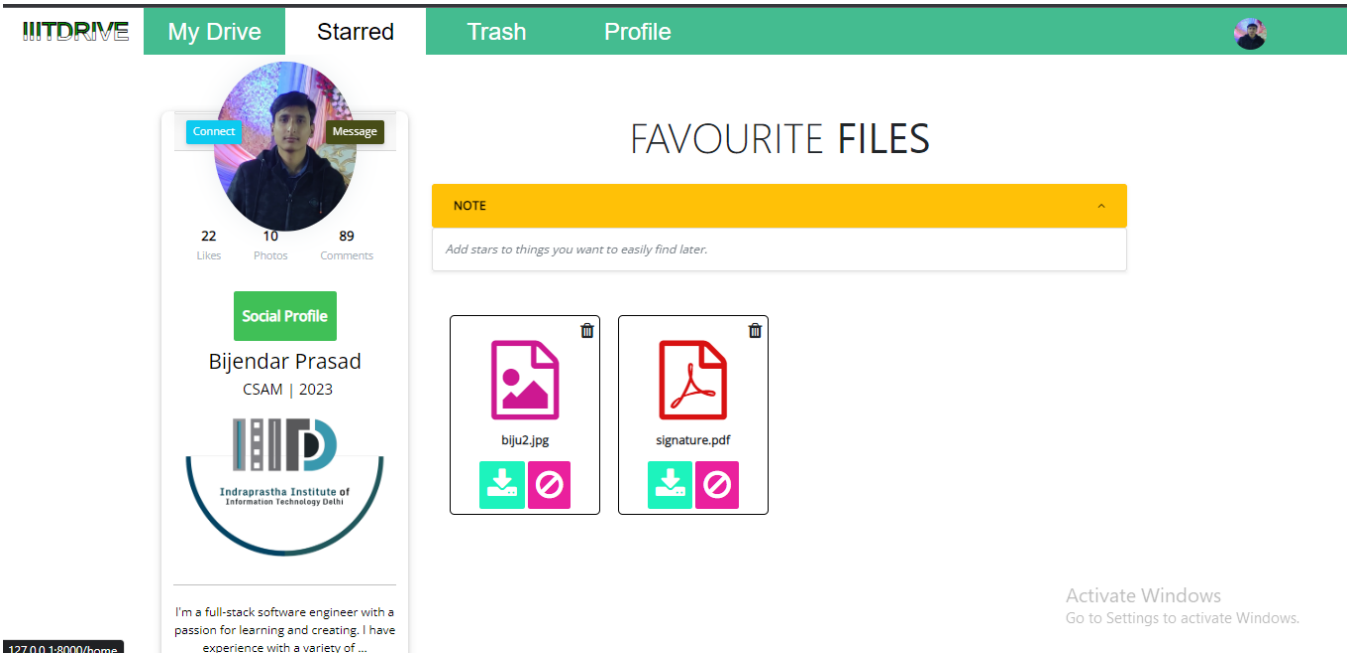


4) Star files

Important files can be starred by the user for quick look-ups.

Implementation Details

a) A is_starred value is tracked with each of the uploaded Files to allow the user to star/unstar files at will.



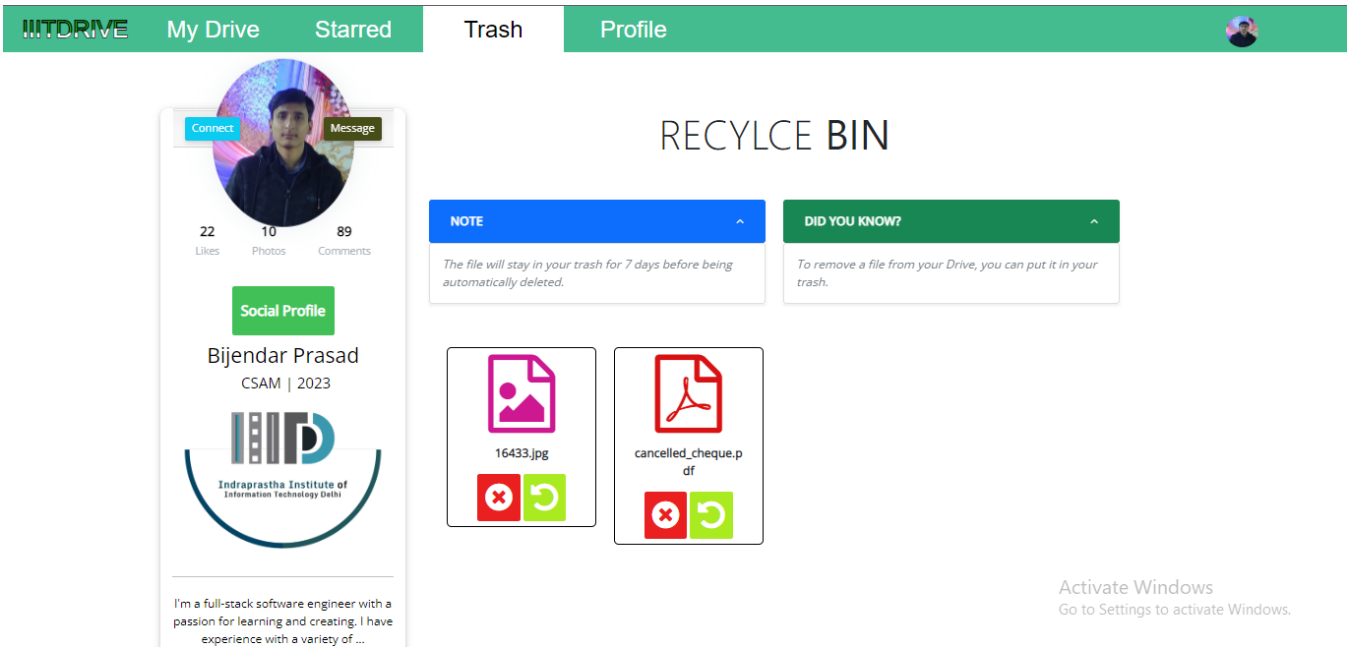
5) Put File in Trash

Unnesceray files can be deleted by the user, which would be first moved to the trash from where its deletes automatically within 7 days of inactivity.

Implementation Details

a) A is_trashed value is tracked with each of the uploaded Files to allow the user to delete/undelete files at will.

b) Further, no hard delete is made even on files deleted from the trash. Instead, a soft delete is done using the is_deleted field.



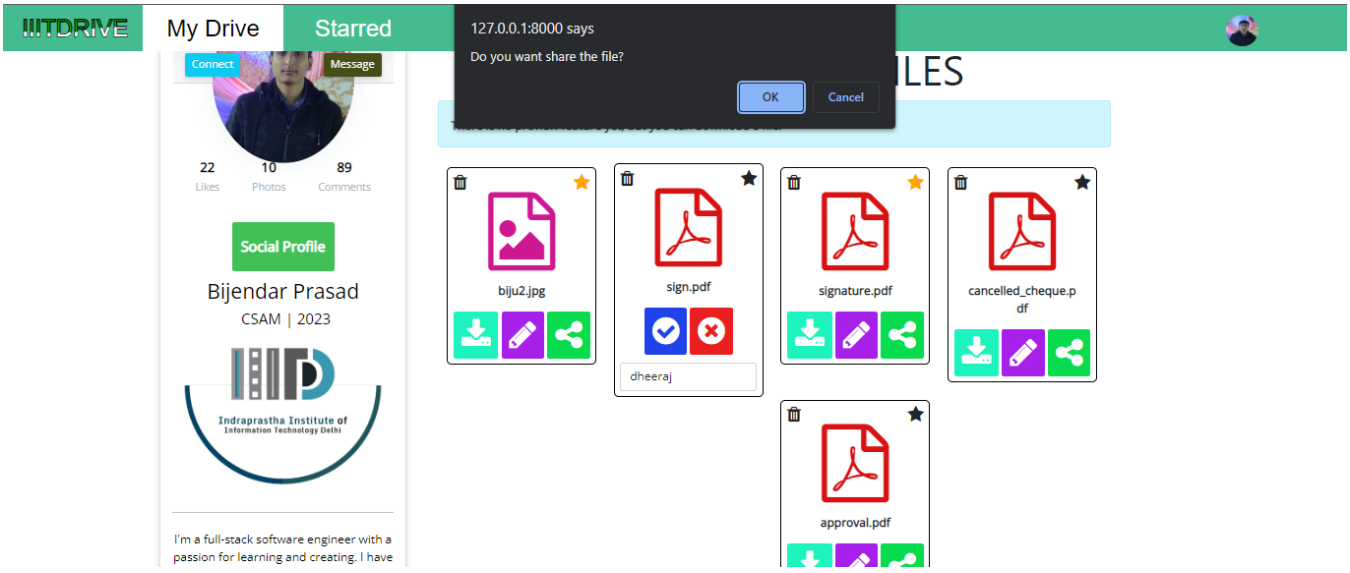
6) Sharing files

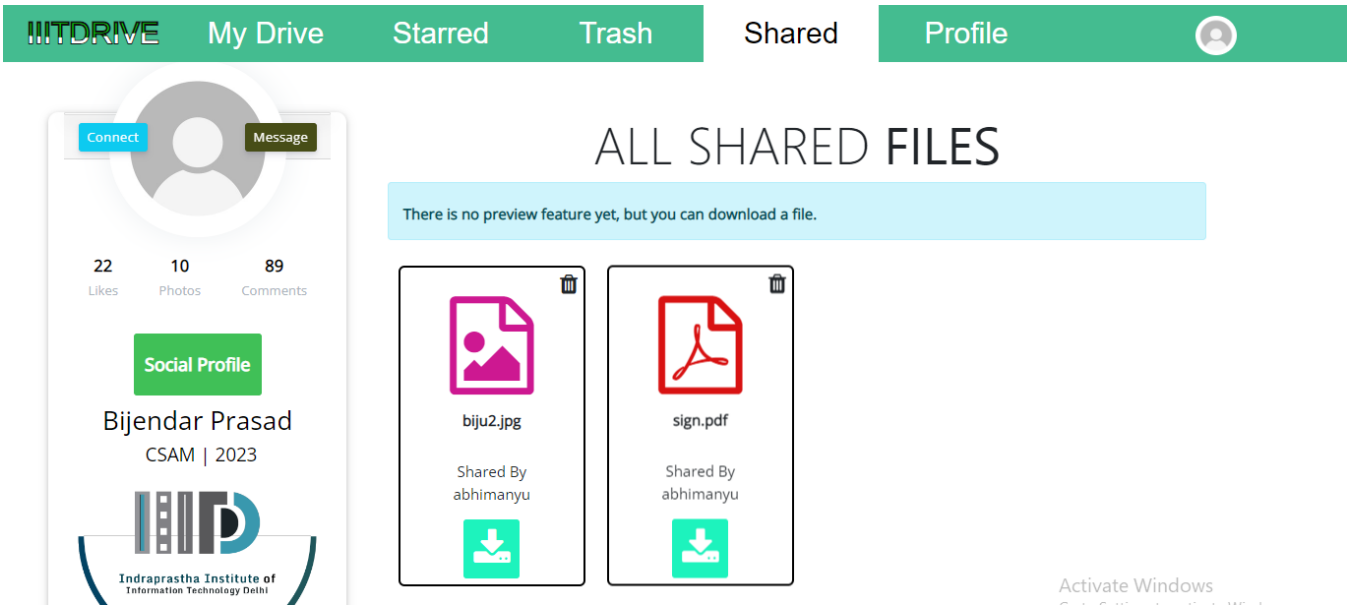
Any user can share the files with all other users through a registered username on the platform.

Implementation Details

a) Each user has corresponding table of all the files which are shared with them.

b) A entry in this table is made based on a new share done by the user of the platform



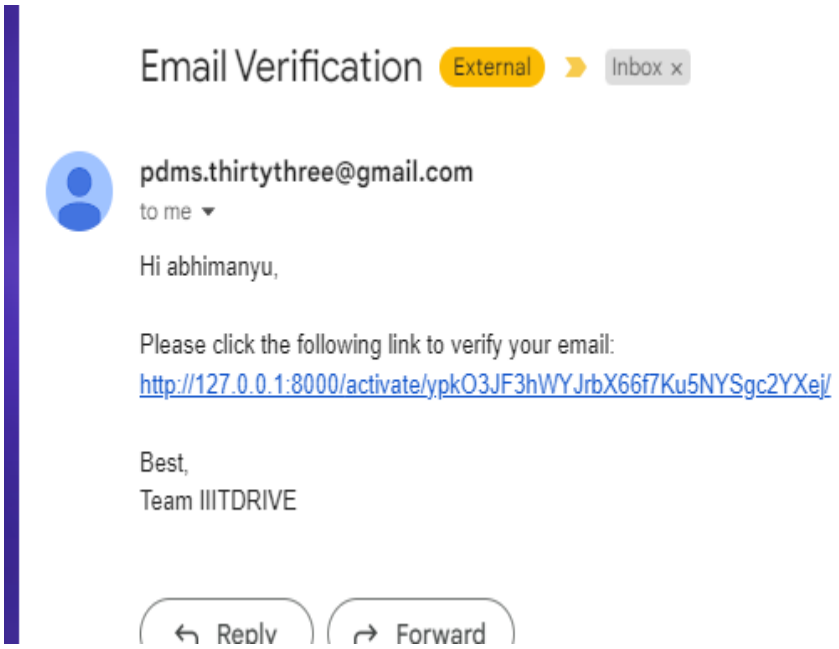


7) Email Notification

The receiver user will get a notification every time the file is shared with them.

Implementation Details

a) When a new share is done by the user, an email to the shared user is sent notifying him of this action.



8) Admin Dashboard

The super admin of the platform can access all the data stored on the platform. The admin can then step in as a moderator whenever required to keep check on platform activity.

Implementation Details

a) Django comes with an automatic admin interface.

b) It reads metadata from all the models (aka tables) to list all the data present in them.