

Owner Identification:

Name: Jahidul Islam Hiron

Email: namehiron.96@gmail.com

Phone: +8801940660150

Designation: Full-Stack Developer

Country: Bangladesh

Requirement Analysis:

1. This is a multi user system to store file and share file with one user to another
2. A user can do signup using manual information (name, email, address, password) or signup with Google (OAuth)
3. User can do sign in
4. A user is able to do logout from the system
5. User is able to perform all operation after sign in
6. After sign in successful, a user can upload single file or multiple files with file title and description
7. A user can edit file title and description
8. User can delete single file
9. A user can download single file
10. User can share single file using user email address

Technologies are used:

1. Javascript (programming language)
2. NodeJs, ExpressJs (backend)
3. JWT & Google OAuth (authentication)
4. ReactJs (frontend)
5. Redux, Redux Toolkit (state management library)
6. MongoDB (NoSql Database)
7. Mongoose (ORM)
8. Styled-components (Styling)
9. Multer, axios, Bcryptjs, express-validator, http-errors, jsonwebtoken, momentjs, uuid, js-file-download, jwt-decode, react-router-dom(v6) [others third-party library]

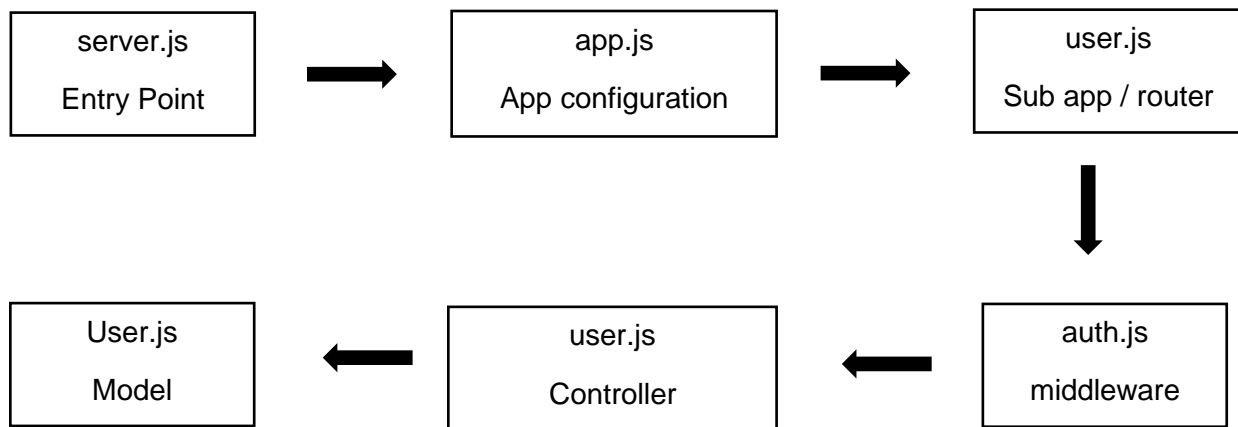
Backend API End Points:

1. Post("/users") → Create user manually
2. Post("/users/signup-with-google") → Signup with google
3. Get("/auth") → Signin user
4. Post("/files") → Upload file create storage and create document to database
5. Patch("/files/:id") → Update file info
6. Delete("/files/:privateKey") → Delete file document and clear storage with privateKey
7. Get("/files") → Get all files information
8. Get("/files/:publicKey") → Download file with publicKey
9. Post("/files/:publicKey") → Share file to other user with publicKey

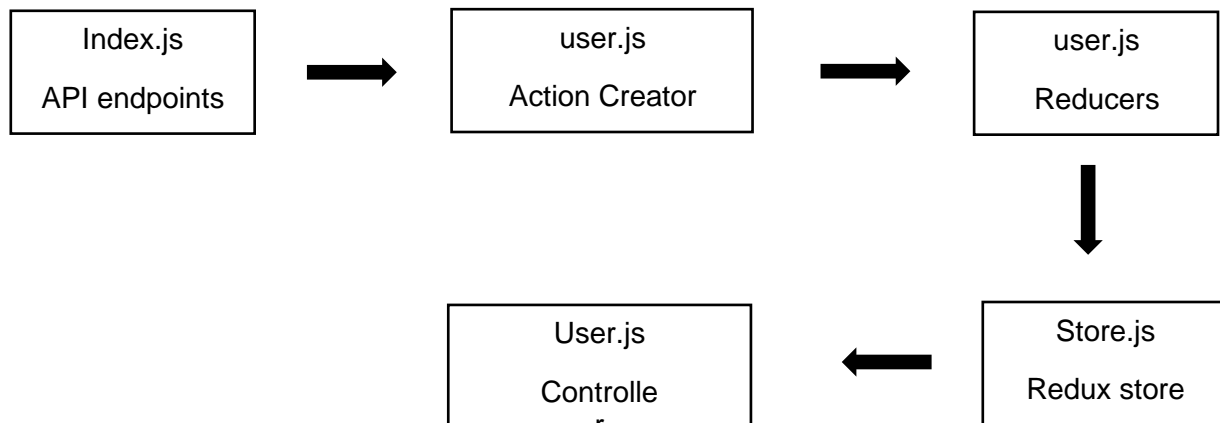
Frontend API End Points:

1. Post("/users") → Create user manually
2. Post("/users/signup-with-google") → Signup with google
3. Get("/auth") → Signin user
4. Post("/file") → Upload file and create document to database
5. Patch("/file/:id") → Update file info with document id
6. Delete("/file/:privateKey") → Delete file document and clear storage with privateKey
7. Get("/file") → Get all files information
8. Get("/file/:publicKey") → Download file with publicKey
9. Post("/file/:publicKey") → Share file to other user with publicKey

Backend Coding Structure:

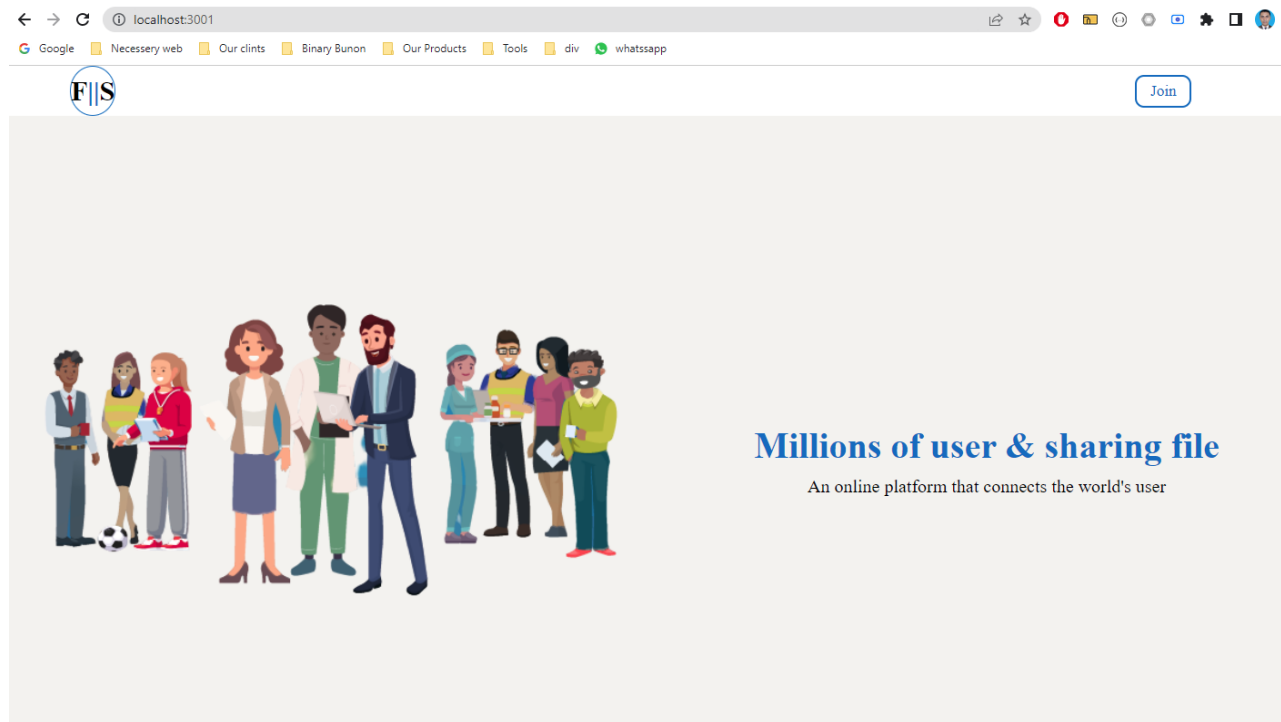


Frontend Coding Structure:



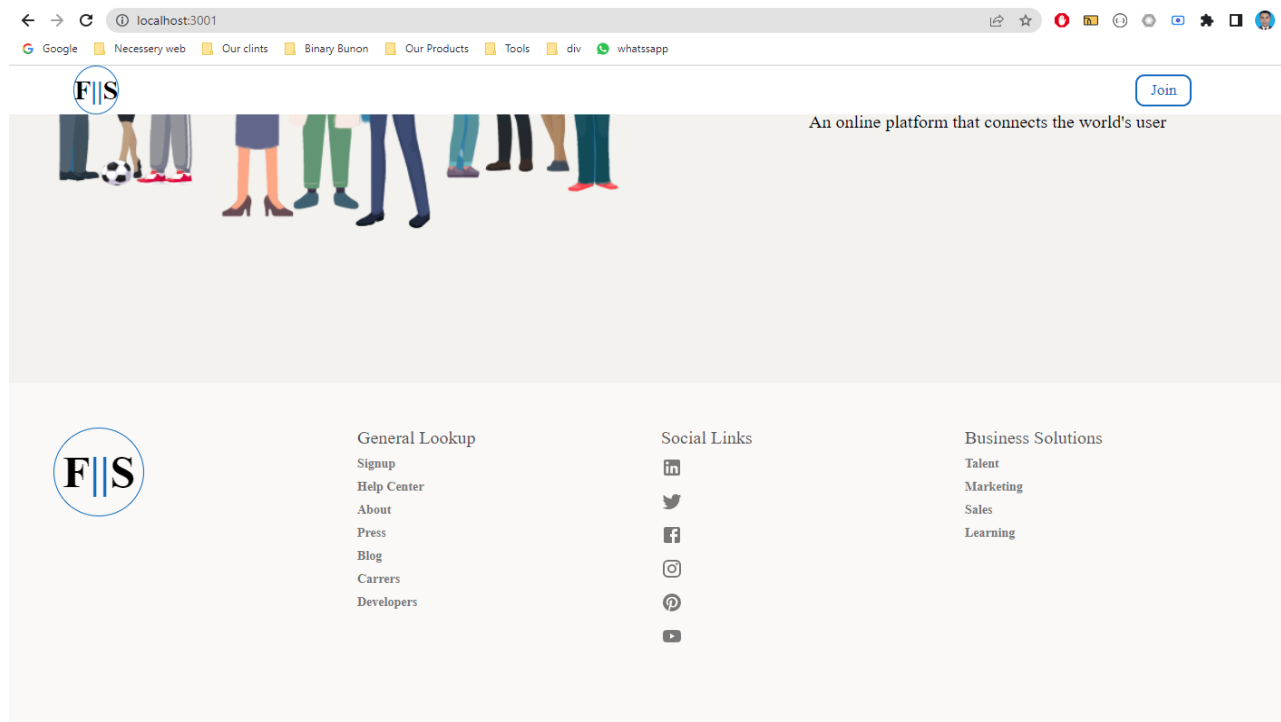
Note: All codes are implemented by following Javascript modularity, clean coding, well documented and commented.

User Manual:



1.1 Home Page

Whenever you visit this website then you are able to see landing page



1.2 Home page footer section

You can navigate home page to join page by clicking respectively logo(left) & join(right) button

localhost:3001/login

Google Necessary web Our clints Binary Bunon Our Products Tools div whatsapp

F||S Join

Create an Account

Name

Email

Password (8 or more characters)

Confirm Password

Signup & Join

Continue with Google

[Already signup ? Signin](#)

1.3 Signup page

localhost:3001/login

Google Necessary web Our clints Binary Bunon Our Products Tools div whatsapp

F||S Join

Create an Account

Name

You should provide a valid name

Email

You should provide a valid email address

Password (8 or more characters)

You should provide a valid credential

Confirm Password

Signup & Join

Continue with Google

[Already signup ? Signin](#)

1.4 Signup Validation before signup

A user has to pass within some validation before signup. There are some frontend validation like **invalid name**, **password doesn't match** and there are some backend validation like **user already exist**

localhost:3001/login

Google Necessary web Our clints Binary Bunon Our Products Tools div whatsapp

F||S Join

Signin

Email

Password (\$ or more characters)

Signin

Continue with Google

[Don't have an account ? Signup](#)

1.5 Sign in page

localhost:3001/login

Google Necessary web Our clints Binary Bunon Our Products Tools div whatsapp

F||S Join

Signin

Email

You should provide a valid email address

Password (\$ or more characters)

You should provide a valid credential

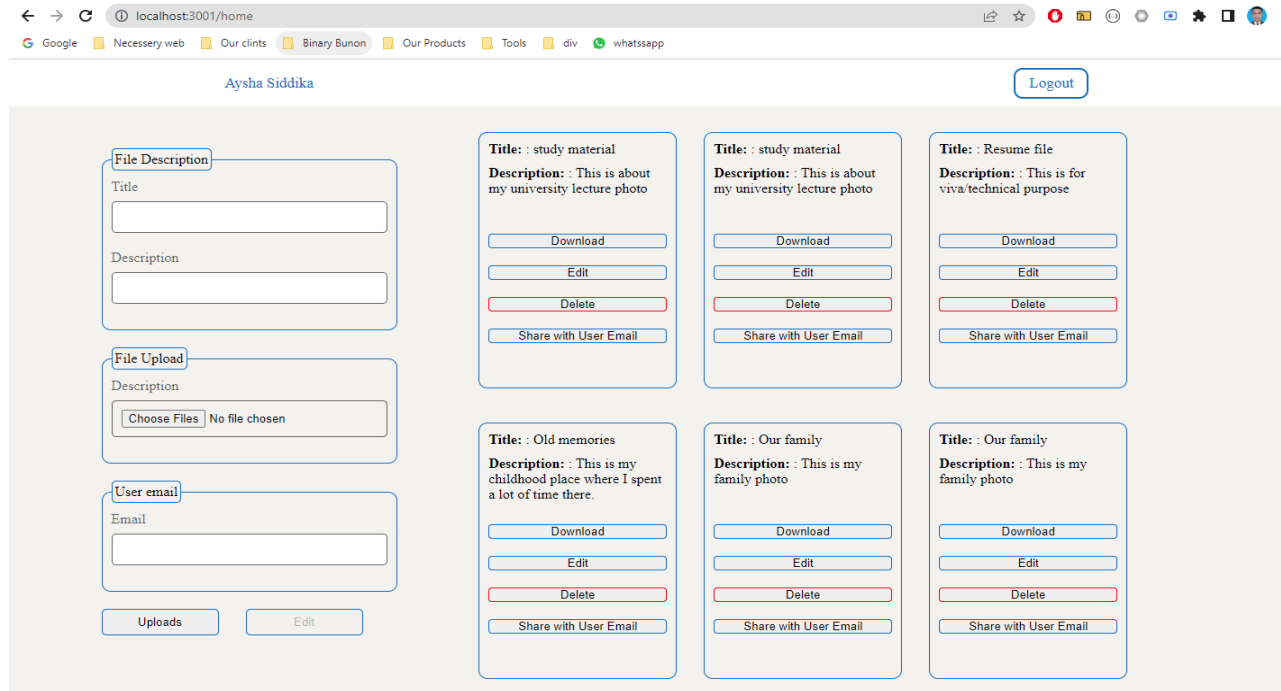
Signin

Continue with Google

[Don't have an account ? Signup](#)

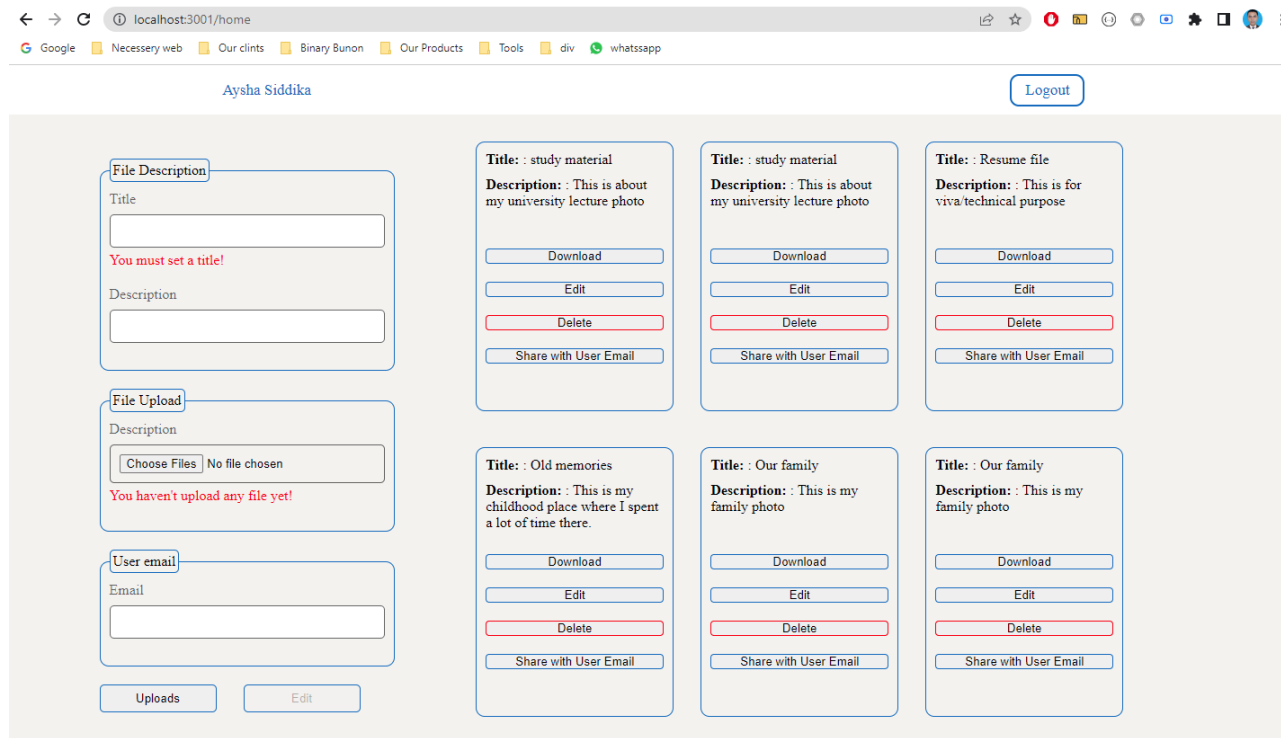
1.6 Sign in validation before sign in

A user has to pass within some validation before sign in. There are some frontend validation like “**you should provide valid email address**” or backend validation like “**invalid credential**”.



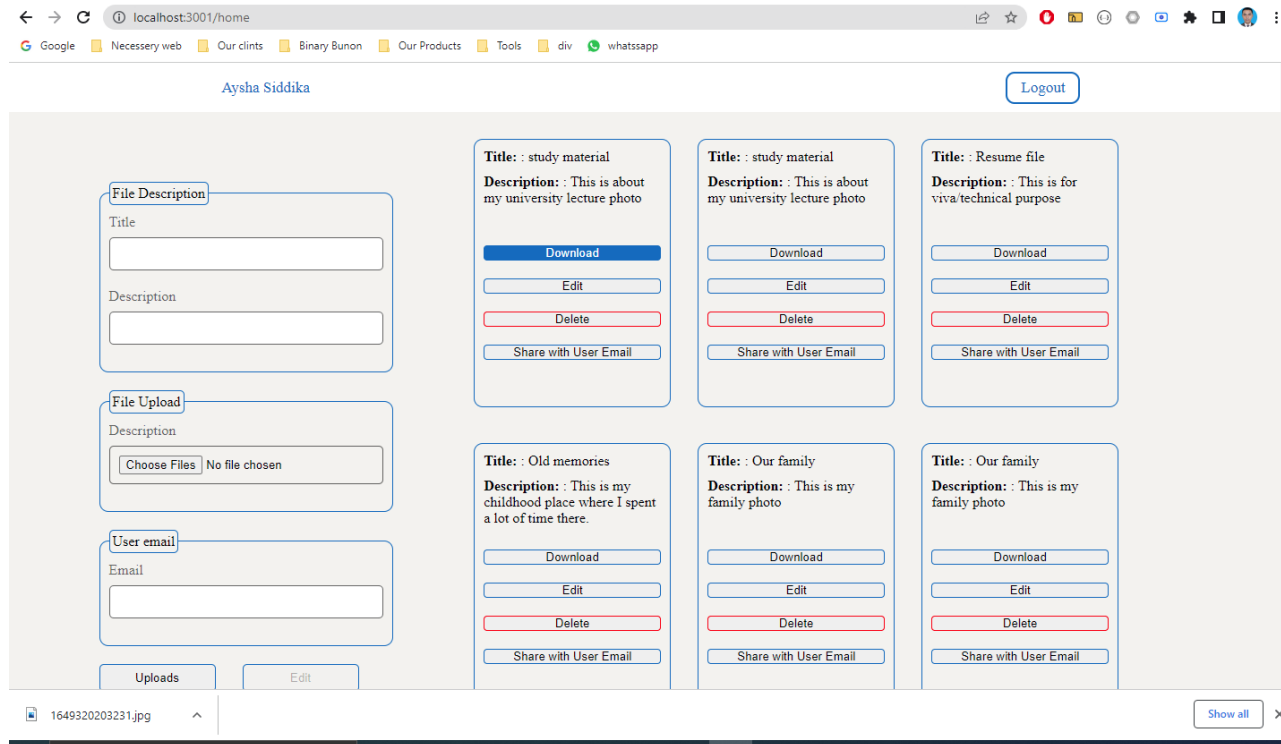
2.1 Home page after login

If a user successfully do login to the system then user will see file list to right side that he uploaded and someone who shared only and in the left side, there is option to create new file



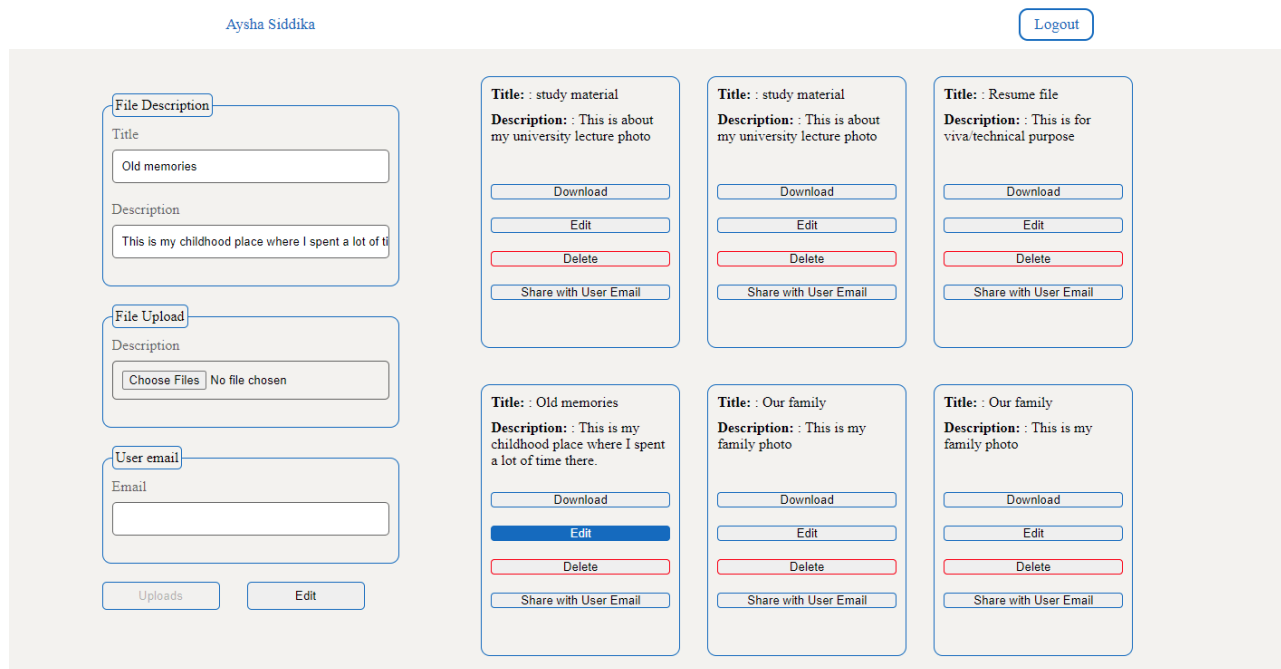
2.2 Add new file

A user must provide file title and at least one file need to upload before storing & making document to database



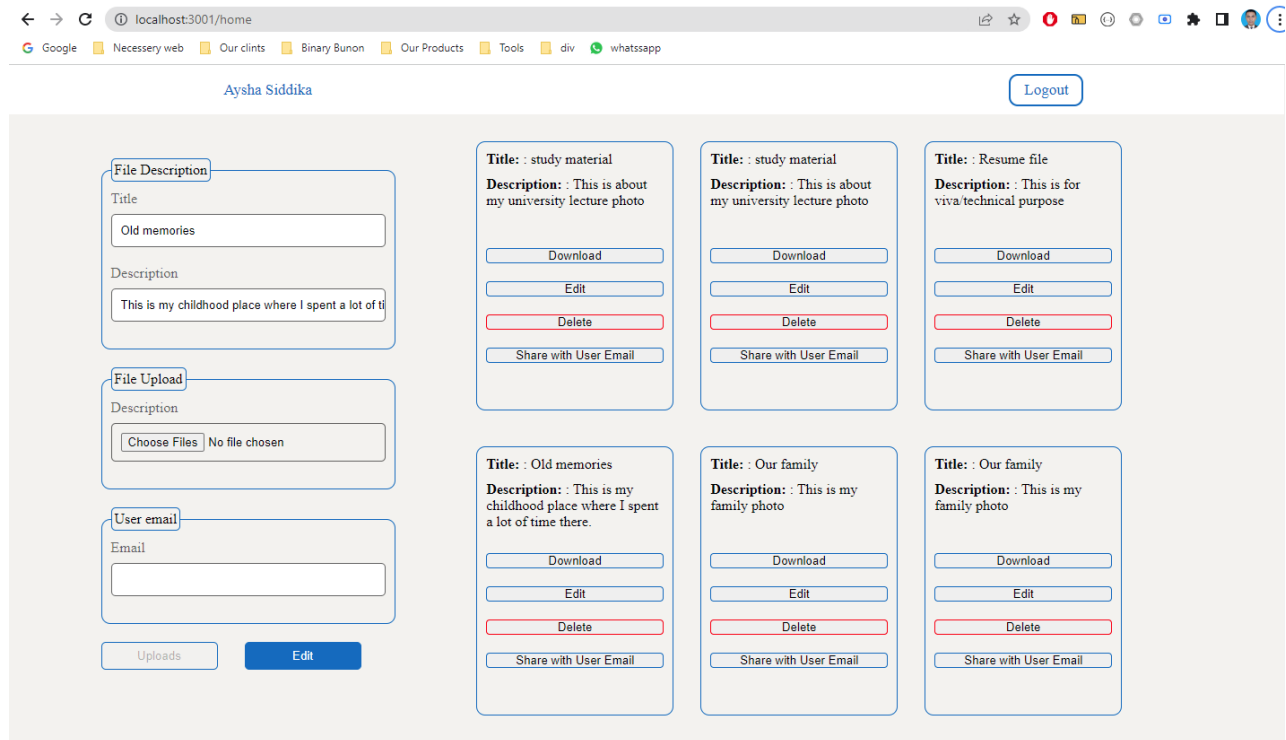
2.3 Download file

A user can download his file by clicking download button attached to particular file with publicKey.



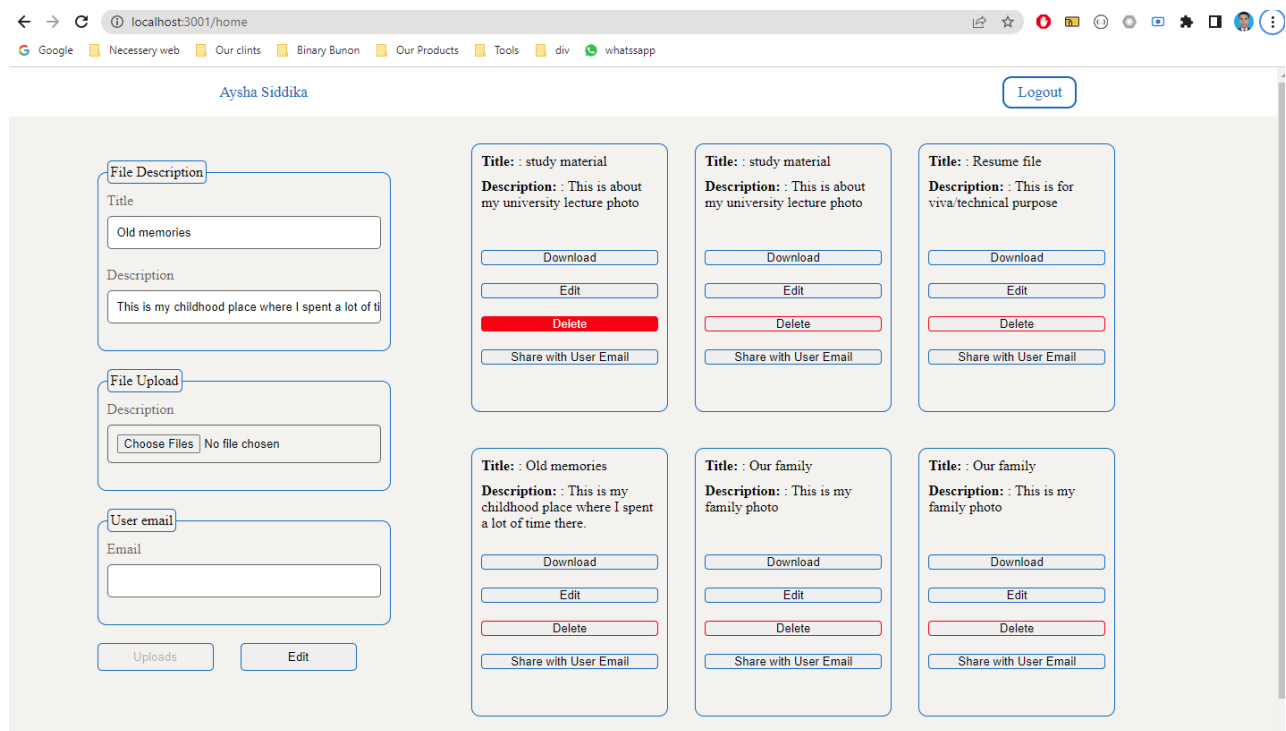
2.4 Load file information to edit

A user can load file information to left side form to edit.



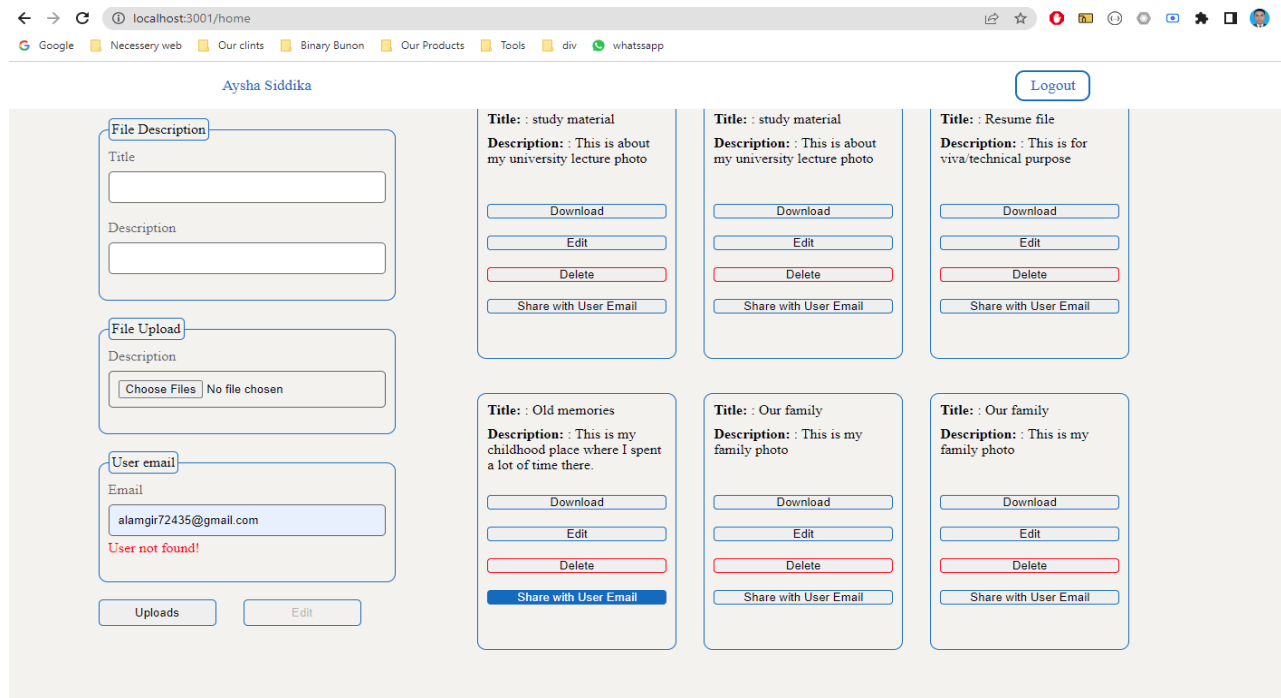
2.5 Edit file information

Finally by clicking left side edit button, a user can update his particular file information by file id.



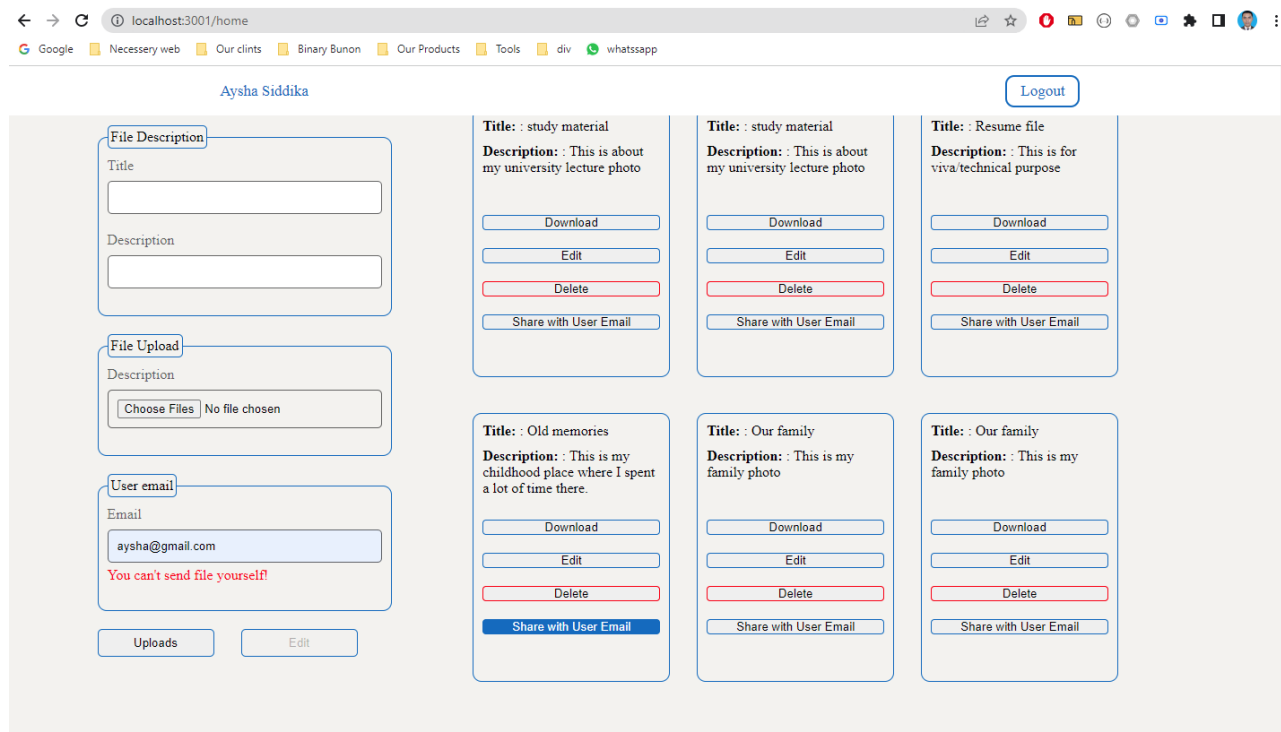
2.6 Delete file

User can delete file and clean storage by privateKey. File created user can only clean storage.



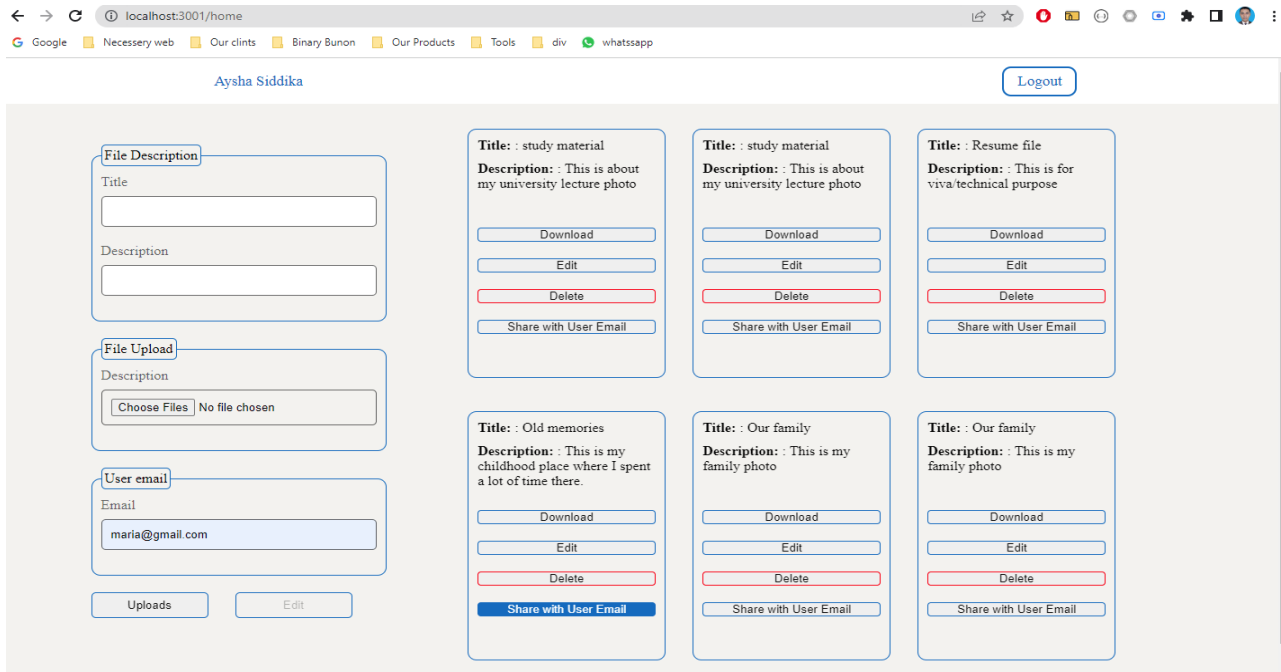
2.7 Share file with user email

During sharing file by user email address, our system will look up that user exist or not. If user doesn't exist then it will us an error message “**User doesn't exist**”



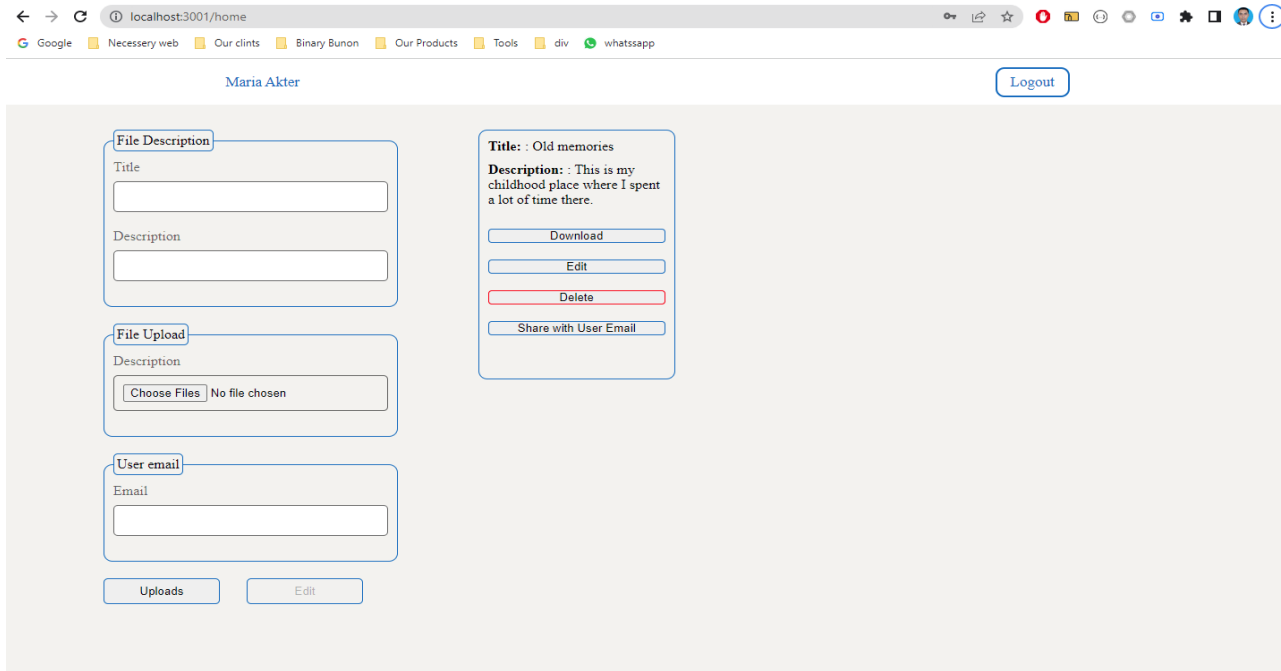
2.8 Share file yourself

If you send file yourself then, it will give us an error that “**you can't send file yourself**”



2.9 Share file yourself

If everything is okay then user is able to share his file to other through user email and publicKey.



3.1 Share file from other user

A user can do everything with shared file like edit, download, and delete. But when user will delete shared file it will delete from database instead cleaning storage. As we need privateKey to clean storage

Project Setup:

Backend setup [Run bellow command]

```
git clone git@github.com:jahidhiron/meldCX-technical-test.git  
cd server  
yarn  
yarn prod
```

Frontend setup [Open another terminal to project folder]

```
cd client  
  
yarn  
  
yarn start
```

Note: For security purpose I have remove CONNECTION_URL value. Please put your MongoDB database link before install this project. Make sure that your database link is online version otherwise you may miss some features.

Server/.env

PORT = 8080

CONNECTION_URL=Please_setup_you_own_mongoDB_connection_string_before_install_this_project___make_sure_that_your_connection_string_is_online_virson___otherwise_you_may_mis
s_some_features

PASSWORD_SALT=12

TOKEN_SECRET_KEY=5585921B461068B67FFFF7170E9364548FA0A6EF

ALLOWED_ORIGIN_1=<http://127.0.0.1:3000>

ALLOWED_ORIGIN_2=<https://127.0.0.1:3000>

FOLDER=uploads