

SecureShare Documentation

OVERVIEW

SecureShare is a powerful web application that allows for secure upload and sharing of documents. Our seamless UI integrations allow for a hassle-free document management experience. SecureShare's scalable infrastructure allows for safe and secure document upload and sharing – with unique features like document verification, certificate generation and password encryption – all in one place.

Team LegionX

Members:

1) Soumyadeepta Das:

email: sdpta1998@gmail.com

phone: 9079765022

linkedin: <https://www.linkedin.com/in/soumyadeepta-das/>

2) Ananya Sajwan:

email: anasajwan2407@gmail.com

phone: 6383311802

linkedin: <https://www.linkedin.com/in/ananya-sajwan-2407/>

GETTING STARTED

A technical and non-technical guide to the features and ecosystem of SecureShare.

Working of SecureShare

A user/organization can easily register to create an account on SecureShare. The user can upload a document at the 'Upload Files' tab which can be optionally password protected. After uploading, a unique certificate of a minimal size of 1KB is generated which can be uploaded elsewhere on third party applications to provide access to the secured document.

The user also has the option to 'Verify Files' which informs the user whether or not the document was modified by an external source. The tool is useful to verify the authenticity of a document.

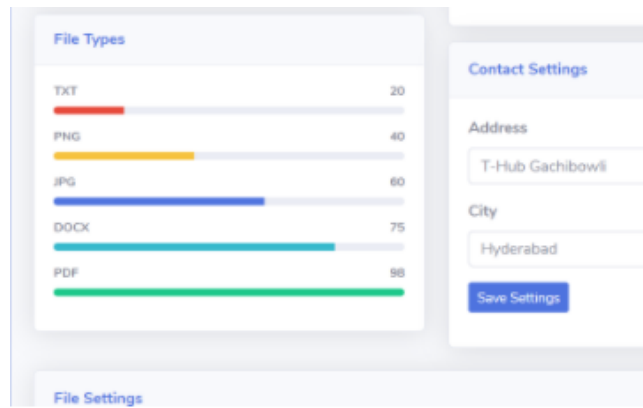
Fetching a document in SecureShare is simple too. If the document is uploaded as a password protected one, then the other end needs the password, this is an optional high security feature otherwise, in case of application forms, when the document is shared in encrypted format in the form, and in the receiving end an authorized user can just fetch the document locally, without any password required.

USER INTERFACE

The user interface will have a single navigation bar for hassle free experience of the user. The UI will show the user how many files of each type are there and the percentage in the form of bar charts for better understanding. (Eg. 20% pdf)

The user has a central profile and a central locker for all files along with the details and tags. The user can keep track of the type of files they have uploaded. They get a dashboard of all files along with the related metadata

The analytics dashboard visible to the user depicting the percentage of each file type uploaded by the user.



REQUIREMENTS

Dependencies:

Main dependencies for this project is NodeJs installed on the system.

There are other packages used which should be installed after `npm install` command.

There should be internet connectivity for running the webapp.

OS built on: Windows 10

Browser: Google Chrome

User System Requirements:

- Web browser

Developer System Requirements:

- Node JS
- Express
- Dependencies in package.json need to be supported

- package.json

```
4  "description": "Create and Verify documents",
5  "main": "server.js",
   ▶ Debug
6  "scripts": {
7    "test": "node test.js",
8    "build": "node server.js",
9    "start": "node server.js"
10 },
11 "author": "",
12 "license": "ISC",
13 "dependencies": {
14   "body-parser": "^1.19.0",
15   "cookie-parser": "^1.4.4",
16   "express": "^4.17.1",
17   "express-session": "^1.17.0",
18   "formidable": "^1.2.1",
19   "http-server": "^0.12.3",
20   "jszip": "^3.2.2",
21   "lowdb": "^1.0.0",
22   "md5": "^2.2.1",
23   "morgan": "^1.9.1",
24   "multer": "^1.4.2",
25   "random-number": "0.0.9",
26   "redis": "^2.8.0",
27   "sha256-file": "^1.0.0",
28   "system-sleep": "^1.3.6"
29 }
30 }
```

Developer requirements/dependencies with version

Installation:

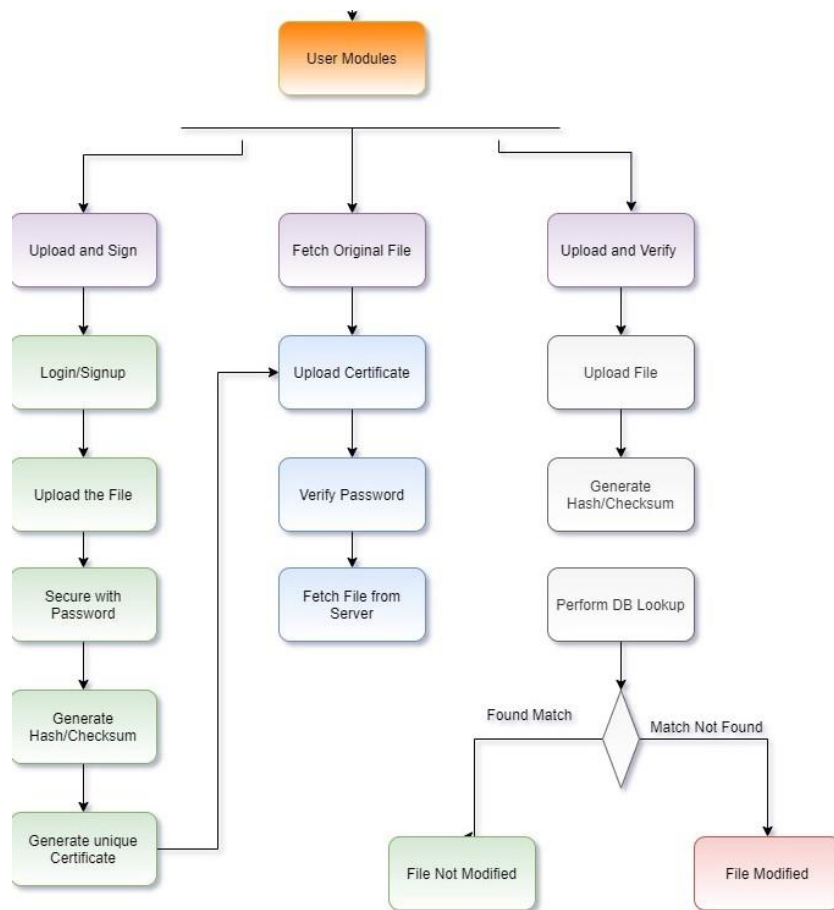
To install the project locally on the system,
the platform it was built on is Windows 10.

Steps:

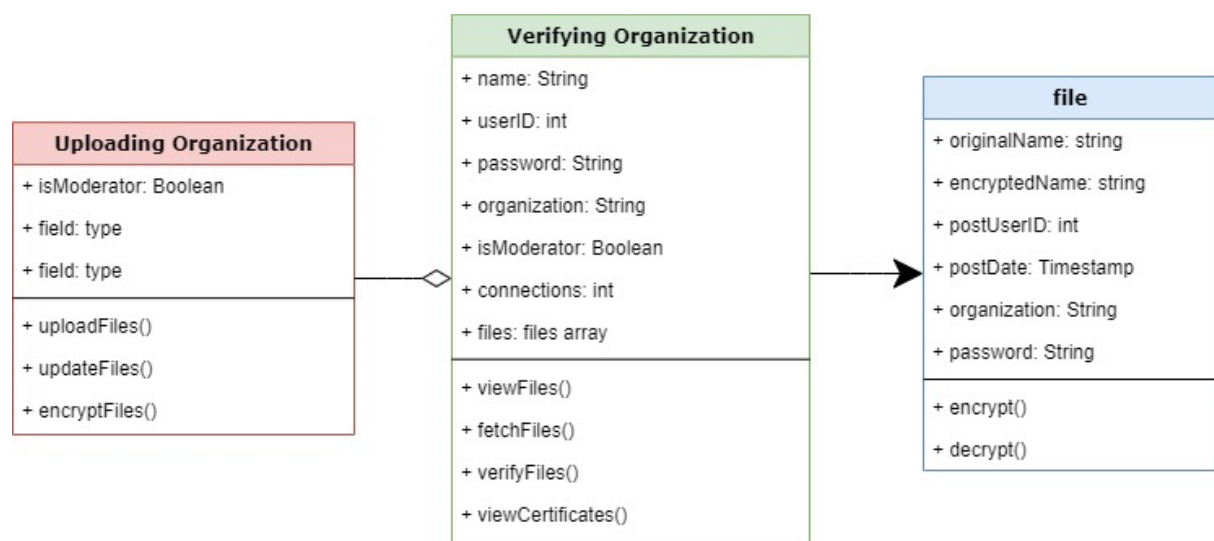
- 1) Start Command Prompt
- 2) cd into project folder
- 3) run `npm install` or `npm install --save`
- 4) run `npm start`

The webapp should start on the specified port

SecureShare Activity Diagram:



SecureShare Class Diagram:



An already created user is available for use after locally running it.

Directly logging in to the webapp using:

email: sdpta1998@gmail.com

password: pass

SecureShare Project Structure:

assets:

- bootstrap
- css
- fonts
- img
- js

certificates:

contains certificates generated for each uploaded documents.

uploads:

stores uploaded documents in encrypted format

server.js:

server file for the project which is run to start project and contains the various functions.

db.json:

stores uploaded filenames, generated filenames, and their generated encrypted passwords as objects, along with the user data, their username, password.

HTML files: corresponding to diff views

- index.html
- index-dynamic.html
- profile.html
- register.html
- upload.html
- verify.html
- fetch.html
- files.html
- 404.html
- stats.html

