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.

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.

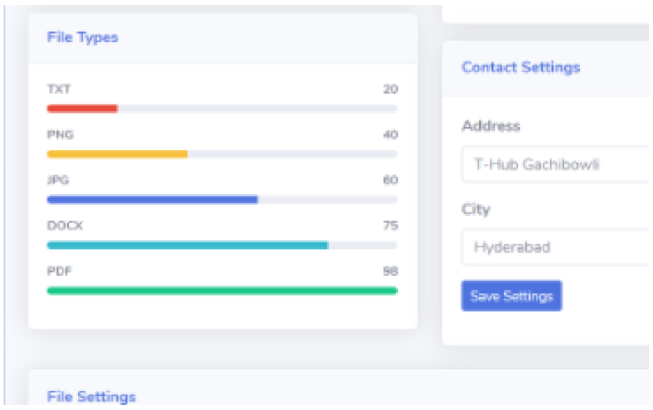
Fetching a document in SecureShare is simple too. If the document is uploaded as a password protected one, then the other end needs the password, this is an optional high security feature otherwise, in case of application forms, when the document is shared in encrypted format in the form, and in the 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.



File Types

TXT	20
PNG	40
JPG	60
DOCX	75
PDF	98

Contact Settings

Address: T-Hub Gachibowli

City: Hyderabad

REQUIREMENTS

User System Requirements:

- Web browser

Developer System Requirements:

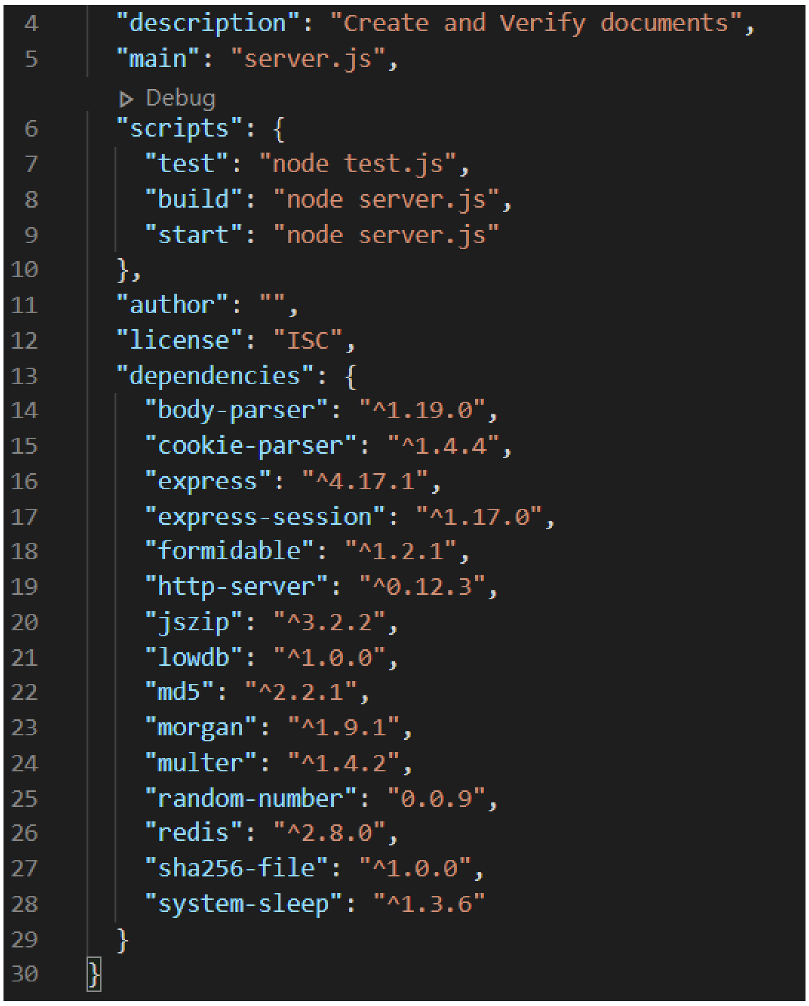
- Node JS
- Express
- Dependencies in package.json need to be supported

Developer Technical Requirements:

- Node JS
- Express
- json DB
- Redis

NPM libraries like:

- jszi,
- lowdb,
- morgan,
- md5,
- sha256-file

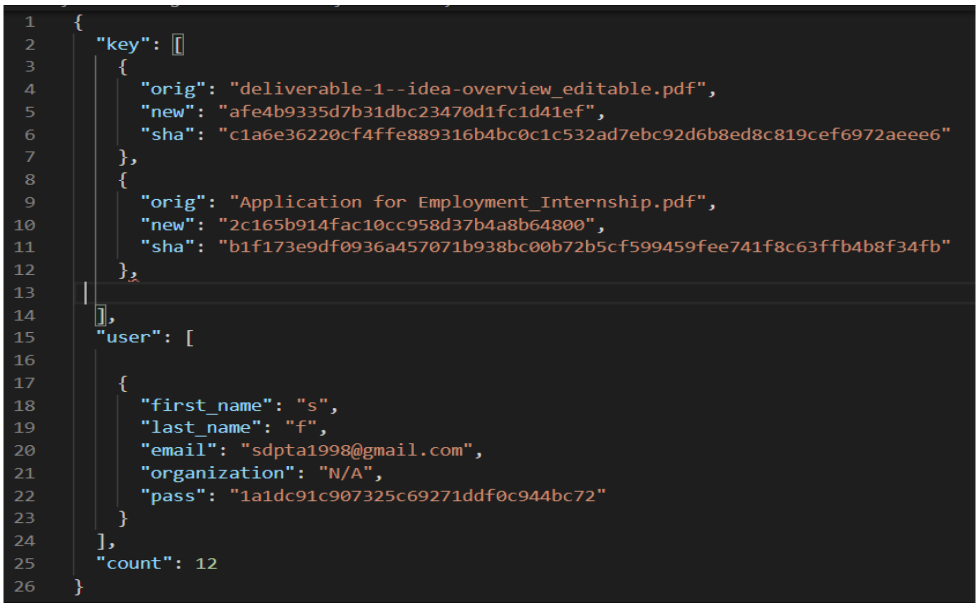


```
  "description": "Create and Verify documents",
  "main": "server.js",
  ▷ Debug
  "scripts": {
    "test": "node test.js",
    "build": "node server.js",
    "start": "node server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "cookie-parser": "^1.4.4",
    "express": "^4.17.1",
    "express-session": "^1.17.0",
    "formidable": "^1.2.1",
    "http-server": "^0.12.3",
    "jszip": "^3.2.2",
    "lowdb": "^1.0.0",
    "md5": "^2.2.1",
    "morgan": "^1.9.1",
    "multer": "^1.4.2",
    "random-number": "0.0.9",
    "redis": "^2.8.0",
    "sha256-file": "^1.0.0",
    "system-sleep": "^1.3.6"
  }
}
```

Developer requirements/dependencies with version

STORAGE

Database JSON View

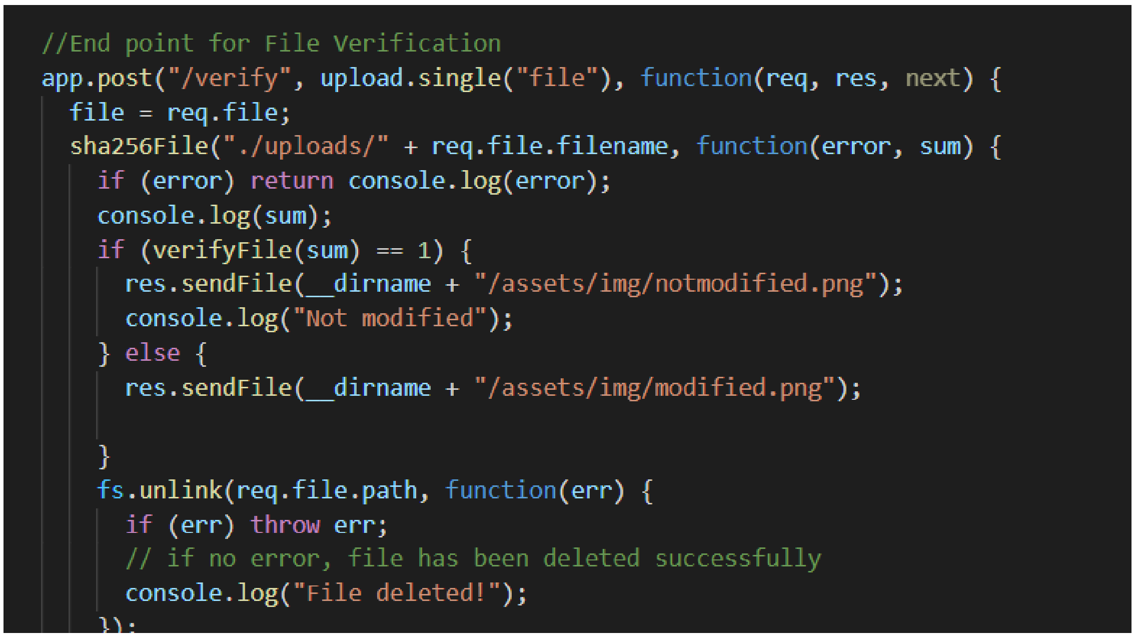


```
{
  "key": [
    {
      "orig": "deliverable-1--idea-overview_editable.pdf",
      "new": "afe4b9335d7b31dbc23470d1fc1d41ef",
      "sha": "c1a6e36220cf4ffe889316b4bc0c1c532ad7ebc92d6b8ed8c819cef6972aeee6"
    },
    {
      "orig": "Application for Employment_Internship.pdf",
      "new": "2c165b914fac10cc958d37b4a8b64800",
      "sha": "b1f173e9df0936a457071b938bc00b72b5cf599459fee741f8c63ffb4b8f34fb"
    },

  ],
  "user": [

    {
      "first_name": "s",
      "last_name": "f",
      "email": "sdpta1998@gmail.com",
      "organization": "N/A",
      "pass": "1a1dc91c907325c69271ddf0c944bc72"
    }
  ],
  "count": 12
}
```

ERROR HANDLING

SecureShare checks if the document uploaded is in one of the supported formats, if not, an error message is displayed. SecureShare also verifies if the document has been modified in any way or not by ‘Verify File’ option.

Function to verify if file has been modified or not and display error message or no error message:



```
//End point for File Verification
app.post("/verify", upload.single("file"), function(req, res, next) {
  file = req.file;
  sha256File("./uploads/" + req.file.filename, function(error, sum) {
    if (error) return console.log(error);
    console.log(sum);
    if (verifyFile(sum) == 1) {
      res.sendFile(__dirname + "/assets/img/notmodified.png");
      console.log("Not modified");
    } else {
      res.sendFile(__dirname + "/assets/img/modified.png");

    }
    fs.unlink(req.file.path, function(err) {
      if (err) throw err;
      // if no error, file has been deleted successfully
      console.log("File deleted!");
    });
```