

File Sharing Application

Project Details

In this project, we had created file sharing application in which we can send and receive files through emails. When we send files via email, the receiver can download the file from the email. In this application, we used Nodejs, express for creating the frontend, and MongoDB Atlas for storing the database.

Team Members:

Student ID	Student Name
C0784327	Gagandeep Kaur
C0784333	Manpreet Kaur

MongoDB Database Information

Database Name:

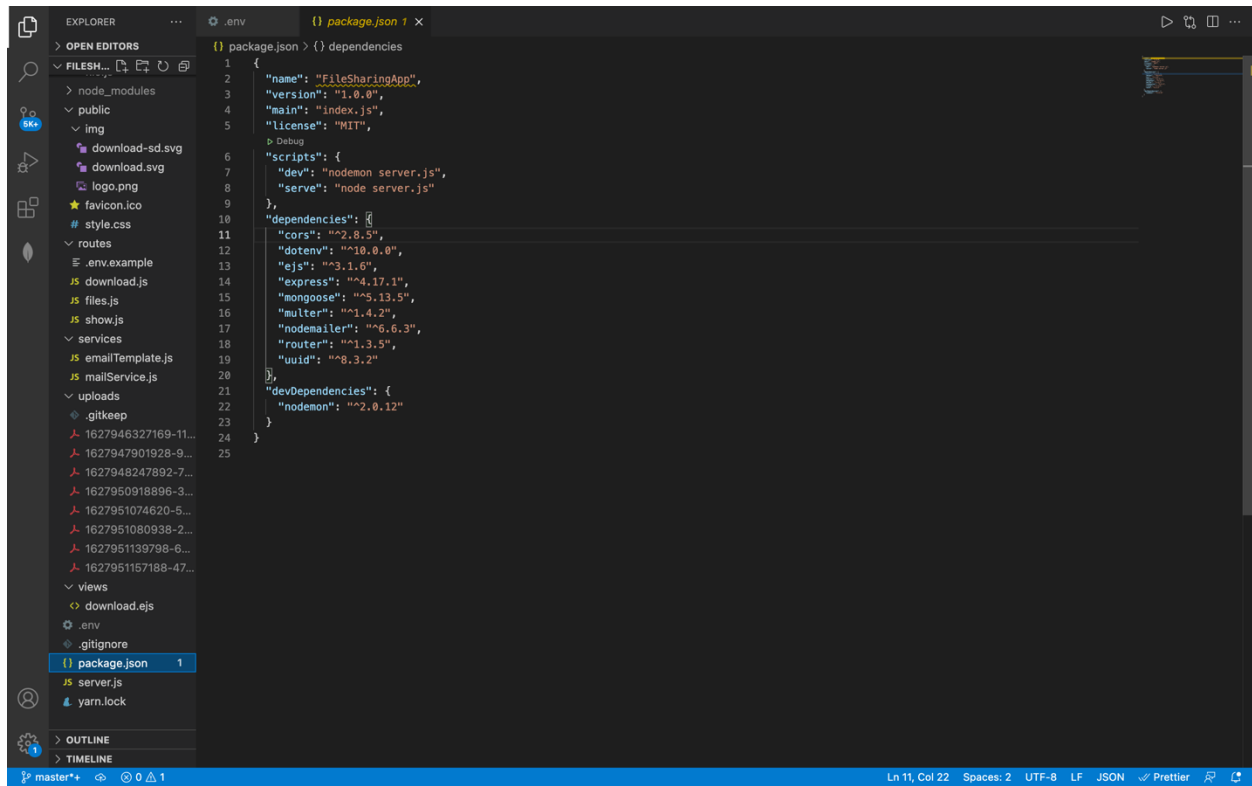
Collection Name	Collection Details and #of Documents available	Team Member(s) worked on it
download	Files which we want to download from email	Gagandeep kaur
upload	These are the files which we want to upload on database	Gagandeep kaur
send	These are the files which we want to send	Manpreet kaur

Database Link of MongoDB Atlas:

<https://cloud.mongodb.com/v2/610872ba62cc7776fd6505b4#clustersNodeJS>

Service Information

Service Codes of NodeJS code package



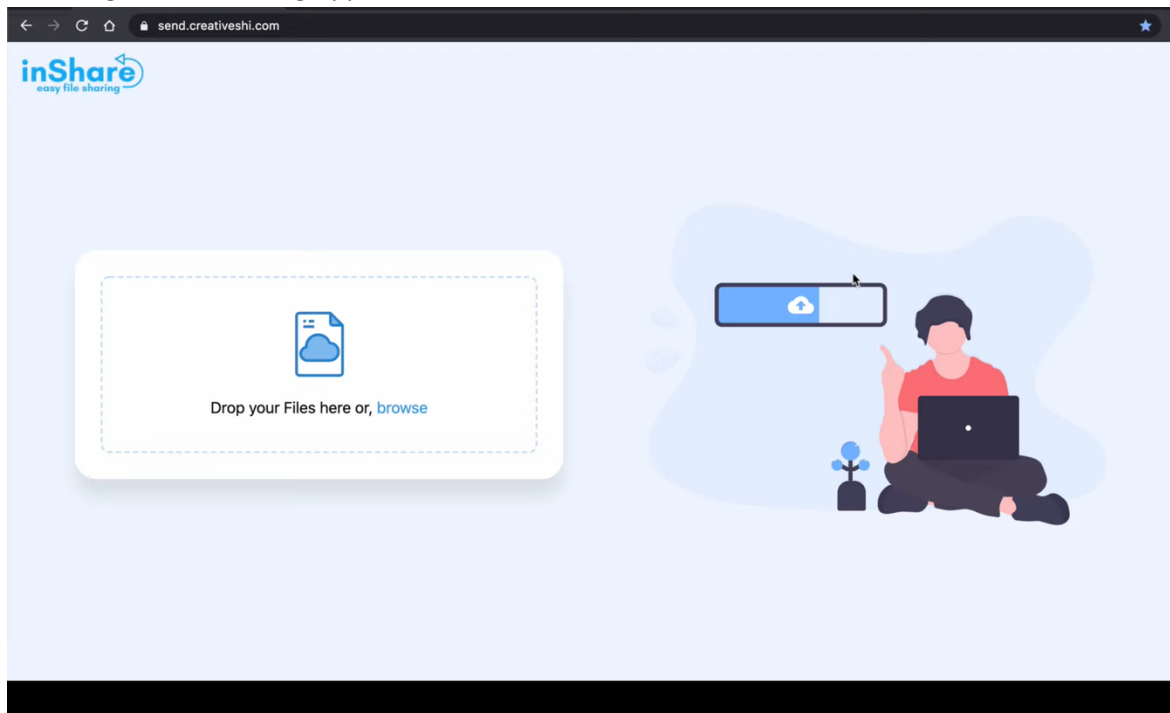
The screenshot shows a VS Code editor with the `package.json` file open. The file contains the following JSON structure:

```
{
  "name": "FileSharingApp",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "dev": "nodemon server.js",
    "serve": "node server.js"
  },
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^10.0.0",
    "ejs": "^3.1.6",
    "express": "^4.17.1",
    "mongoose": "^5.13.5",
    "multer": "^1.4.2",
    "nodemailer": "^6.6.3",
    "router": "^1.3.5",
    "uuid": "^8.3.2"
  },
  "devDependencies": {
    "nodemon": "^2.0.12"
  }
}
```

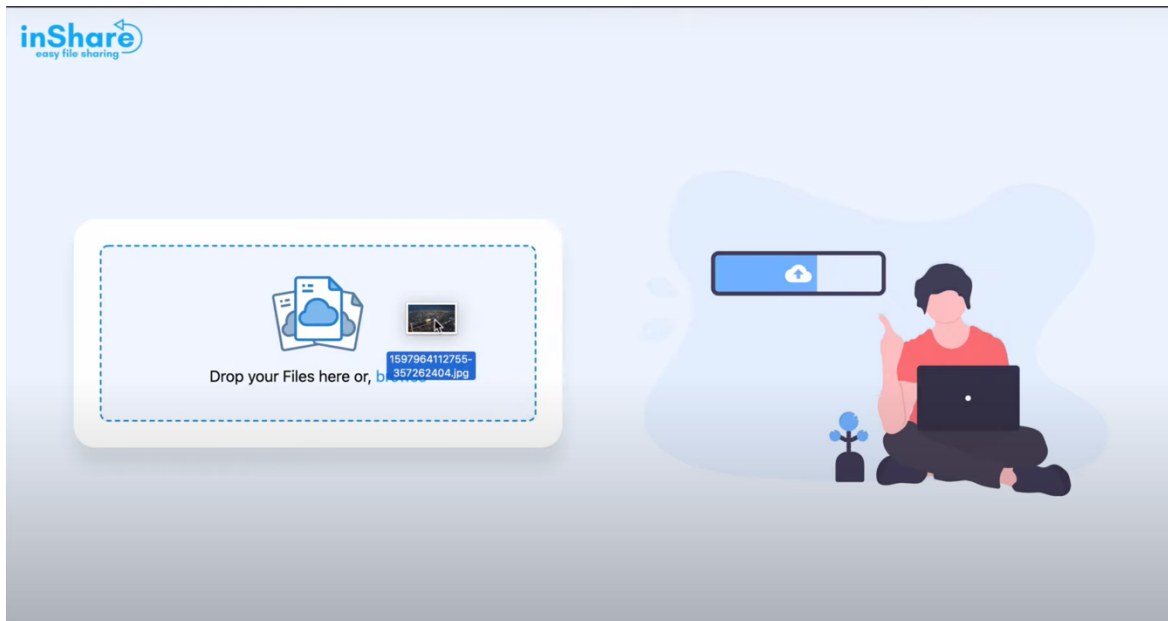
The Explorer sidebar on the left shows the project structure, including files like `download-sd.svg`, `download.svg`, `logo.png`, `favicon.ico`, `style.css`, `routes`, `.env.example`, `download.js`, `files.js`, `show.js`, `services`, `emailTemplate.js`, `mailService.js`, `uploads`, `.gitkeep`, `.gitignore`, `package.json` (selected), `server.js`, and `yarn.lock`.

Application Details

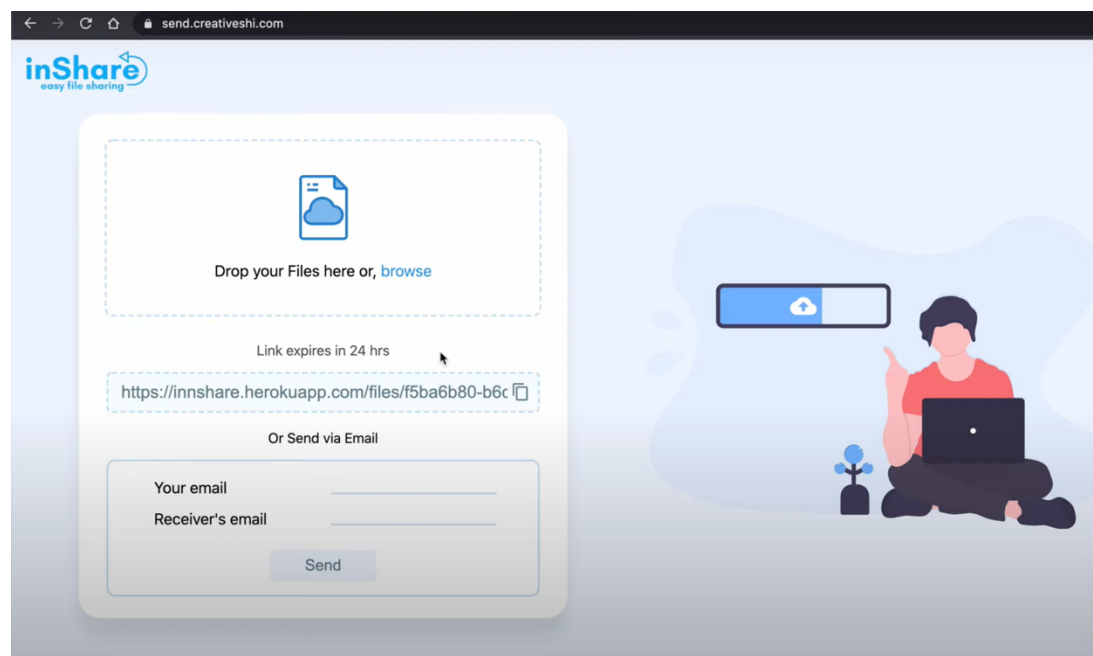
1.) Main Page of File Sharing Application



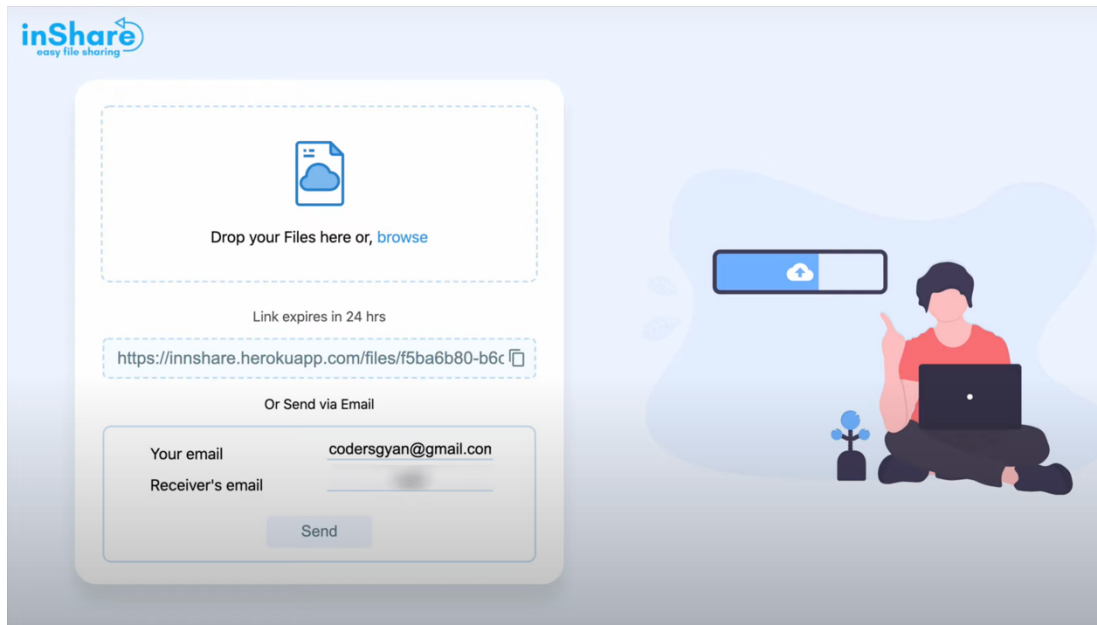
2.) We can drag file from via browse



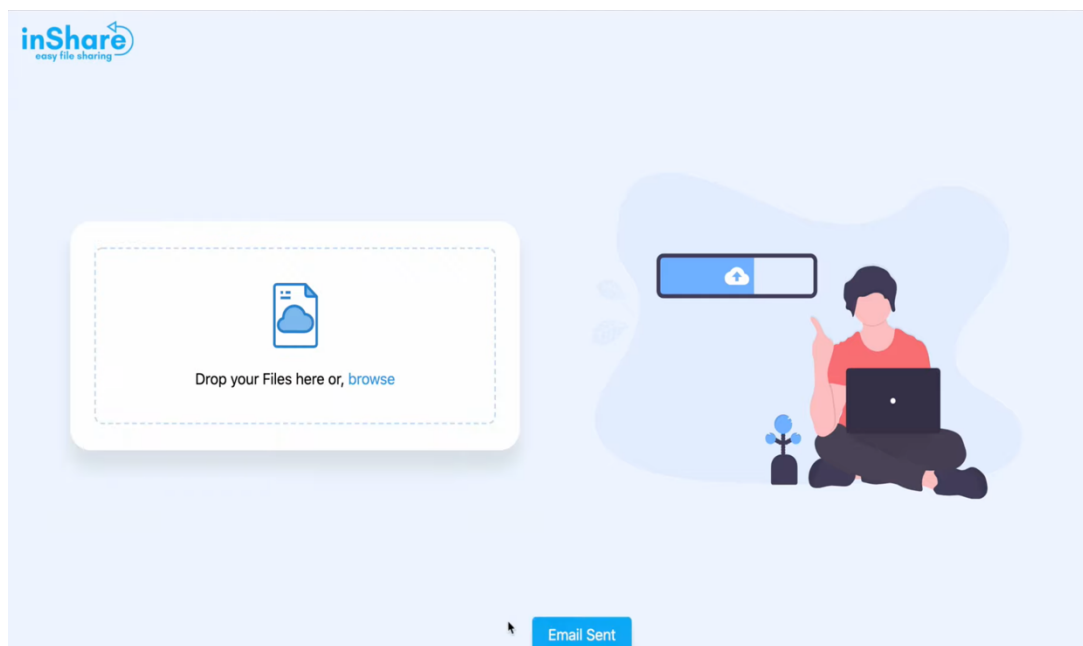
3.) When you drag file to browse, then file is uploaded



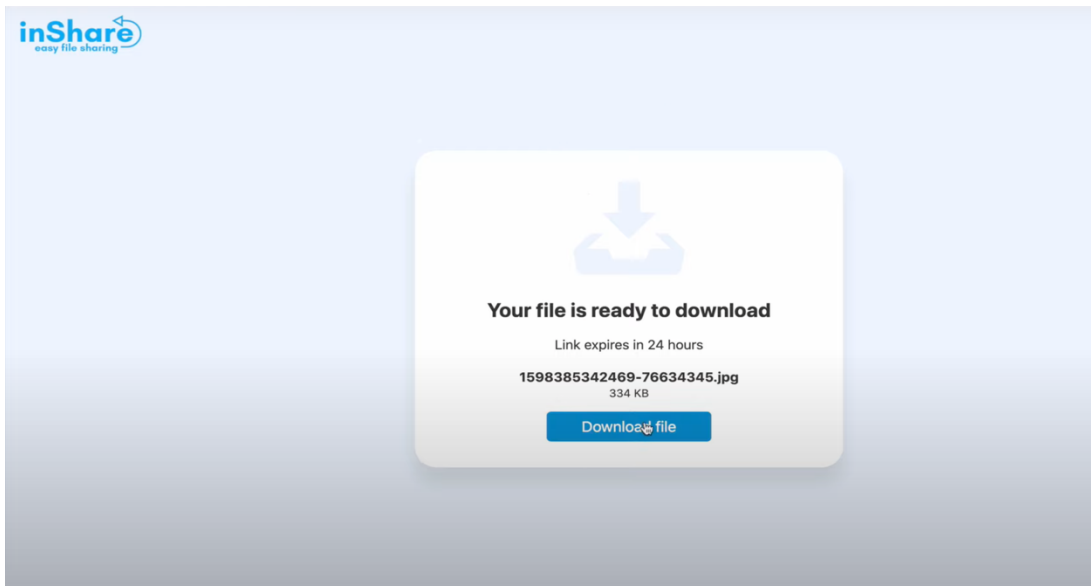
4.) Then we have to fill email address of sender and receiver



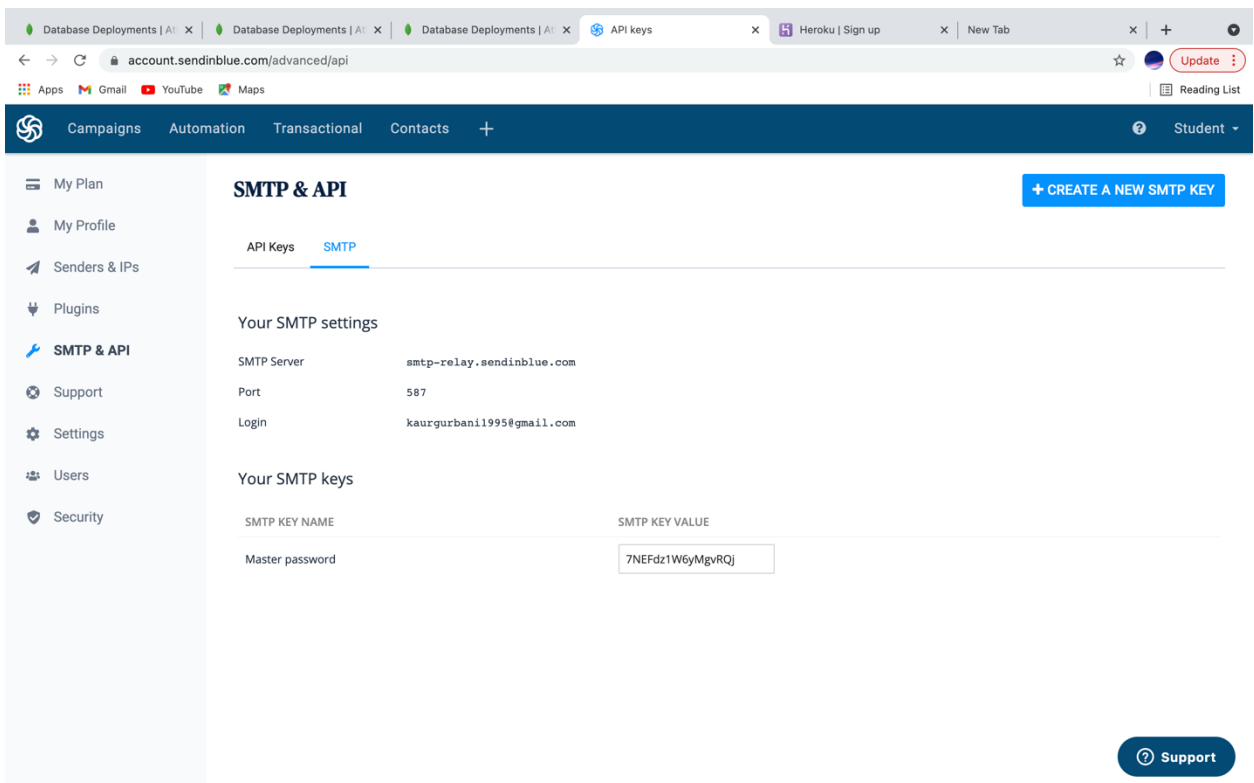
5.) When we click on Send Button, it will show you email sent popup



6.) On the other side, receiver can download the file via link



7.) SMTP & API for server



8.) MongoDB Database with collections

The screenshot displays the MongoDB Atlas web interface. The browser address bar shows the URL: `cloud.mongodb.com/v2/610872ba62cc7776fd6505b4#metrics/replicaSet/61087379c644b3430941556c/explorer/FileShare/files/find`. The interface includes a top navigation bar with tabs for 'Database Deployments | Atlas', 'Data | Atlas: MongoDB Atlas', and 'API keys'. Below this is a sidebar with navigation options: 'FilesShare', 'Atlas', 'Realm', and 'Charts'. The main content area is titled 'Cluster0' and shows the 'FileShare.files' collection. The collection details include 'COLLECTION SIZE: 2.29KB', 'TOTAL DOCUMENTS: 11', and 'INDEXES TOTAL SIZE: 36KB'. A 'Find' button is visible, and a query filter is entered: `{ "filter": "example" }`. The query results show 11 documents, with the first document displayed as a JSON object:

```
{
  "_id": ObjectId("61087d572ad4bc9ec0d5ab10"),
  "filename": "1627946327169-11056536.pdf",
  "uuid": "319fed32-ca56-4002-bd7e-ad4298121318",
  "path": "uploads/1627946327169-11056536.pdf",
  "size": 99929,
  "createdAt": 2021-08-02T23:18:47.181+00:00,
  "updatedAt": 2021-08-02T23:18:47.181+00:00,
  "__v": 0
}
```

. The interface also includes a 'Get Started' button and a 'Feature Requests' link.

9.) POST data via Insomnia tool

The screenshot shows the Insomnia API client interface. The top bar indicates the environment is 'No Environment' and the current view is 'File Sharing'. The main panel displays a POST request to 'https://api.myproduct.com/v1/users'. The request body is a JSON object with the following content:

```
1- {"uuid": "a6030be9-8a8f-4336-94d6-d042431c5767",  
2  "emailTo": "kaurgurbaan1995@gmail.com",  
3  "emailFrom": "birthdaygagan5991@gmail.com"  
4  }
```

The response status is 'Error' with a message '0 ms' and '0 B'. The response body is empty. The left sidebar shows a list of requests: 'POST Send Email', 'GET download page', and 'POST Upload file'. The right sidebar shows the 'Timeline' tab with a log of the request execution, including details about the request preparation, current time, and various library versions.

10.) GET data

The screenshot shows the Insomnia API client interface. The top bar indicates the environment is 'No Environment' and the current view is 'File Sharing'. The main panel displays a GET request to 'http://localhost:6000/files/a6030be9-8a8f-4336-94d6-d042431c5767'. The response status is '200 OK' with a message '54.1 ms' and '964 B'. The response body is a PDF file named 'filesshare-download.pdf'. The left sidebar shows a list of requests: 'POST Send Email', 'GET download page', and 'POST Upload file'. The right sidebar shows the 'Body' tab with the content of the PDF file, which includes a 'Ready to download' message and a link to download the file.

11.) Upload file via Rest API client Insomnia

