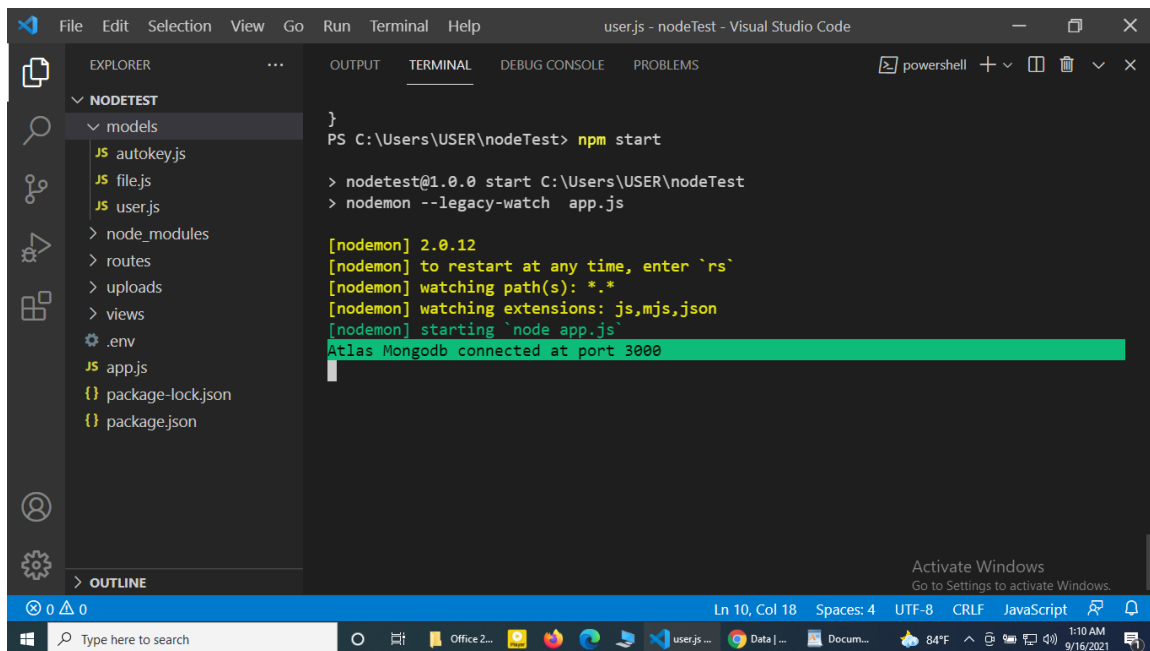


# PROJECT TEST INSTRUCTIONS

While working on it, I thought it will be a good idea to make a detailed instruction. So I took screen shots of different parts of execution and elaborated the steps.

This file consists of screenshots of the file system application, and logs of VS code console and mogodb

- 1) To start running the app using npm start



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure for 'nodetest'. The file explorer shows a 'models' folder and several files including 'autokey.js', 'file.js', 'user.js', 'node\_modules', 'routes', 'uploads', 'views', '.env', 'app.js', 'package-lock.json', and 'package.json'. The Terminal view on the right shows the command prompt output for running 'npm start'. The output indicates that the application is starting using nodemon, watching for file changes, and connecting to Atlas MongoDB at port 3000. The terminal text is as follows:

```
PS C:\Users\USER\nodeTest> npm start

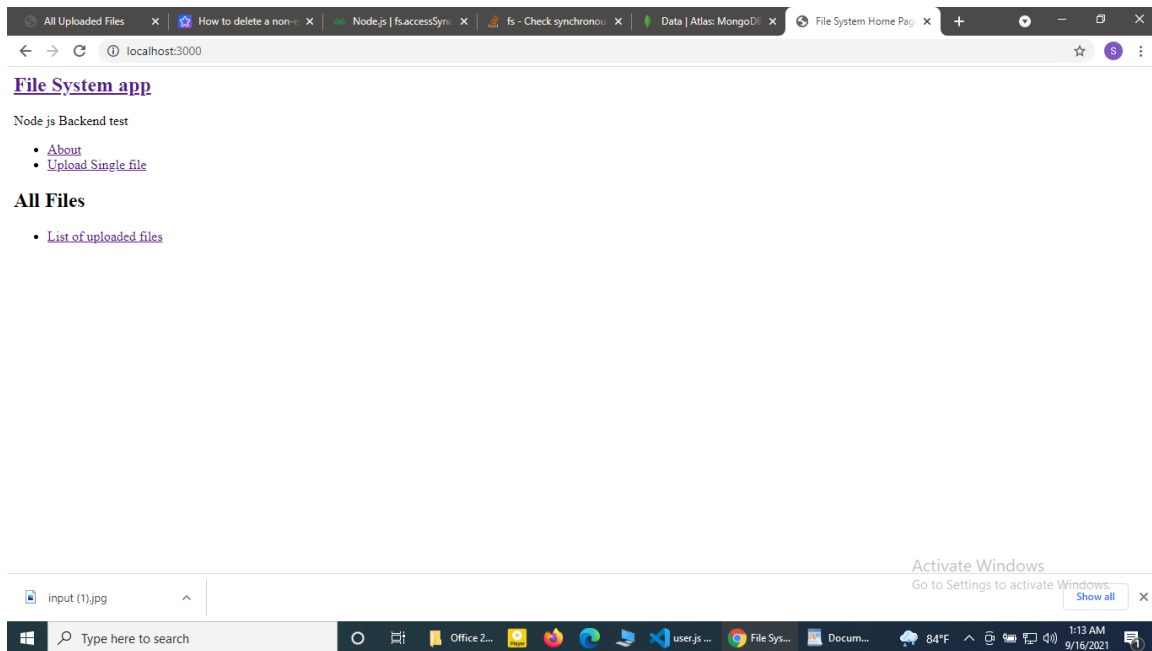
> nodetest@1.0.0 start C:\Users\USER\nodeTest
> nodemon --legacy-watch app.js

[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Atlas MongoDB connected at port 3000
```

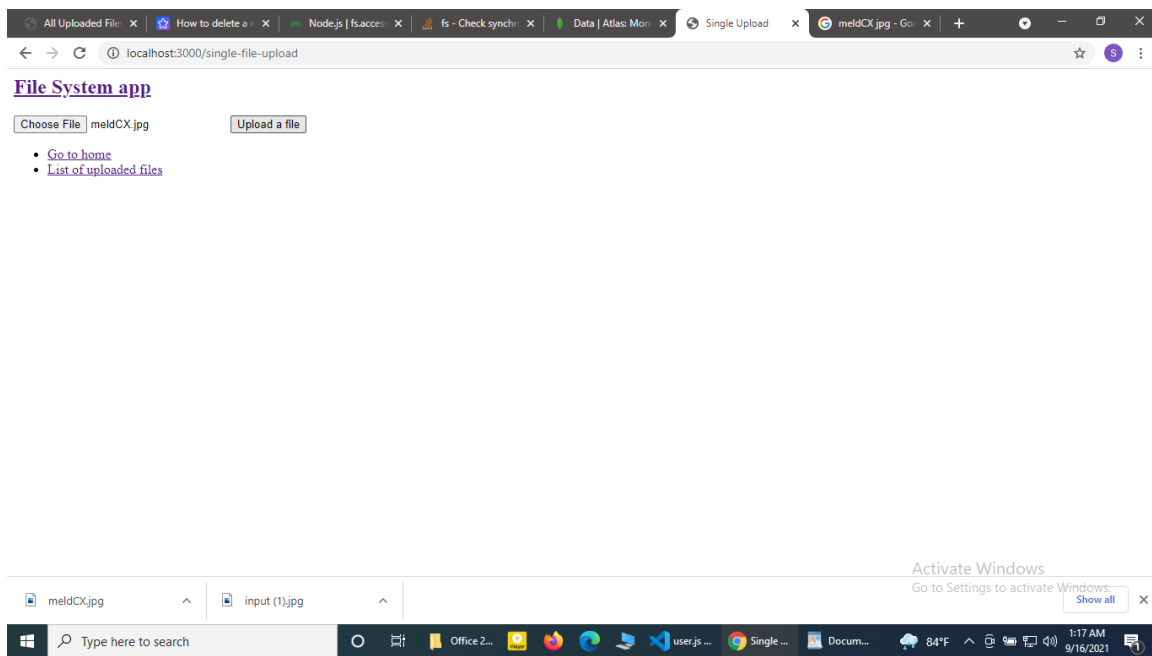
- 2) Now I will try to run the application as client from google chrome at localhost:3000

This is the home page of the application. It has 3 links:

- About
- Upload a single file
- List of uploaded files

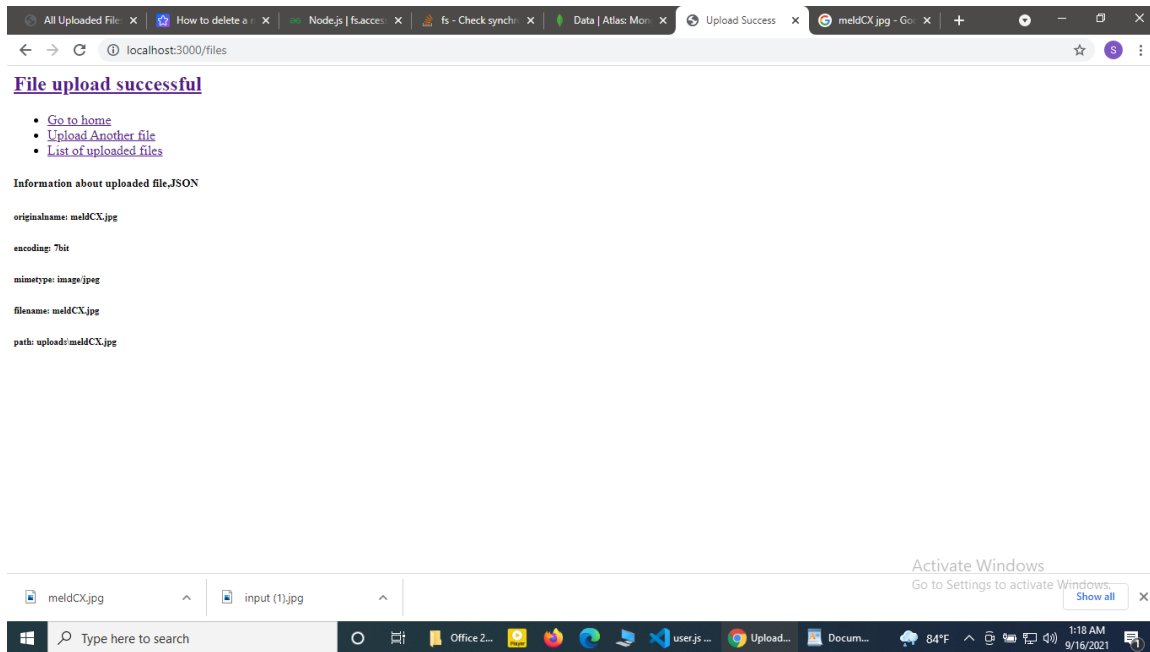


3) Now I will be uploading a file to this app by going to the link "Upload Single File". Here I select a jpg and hit the upload file button.

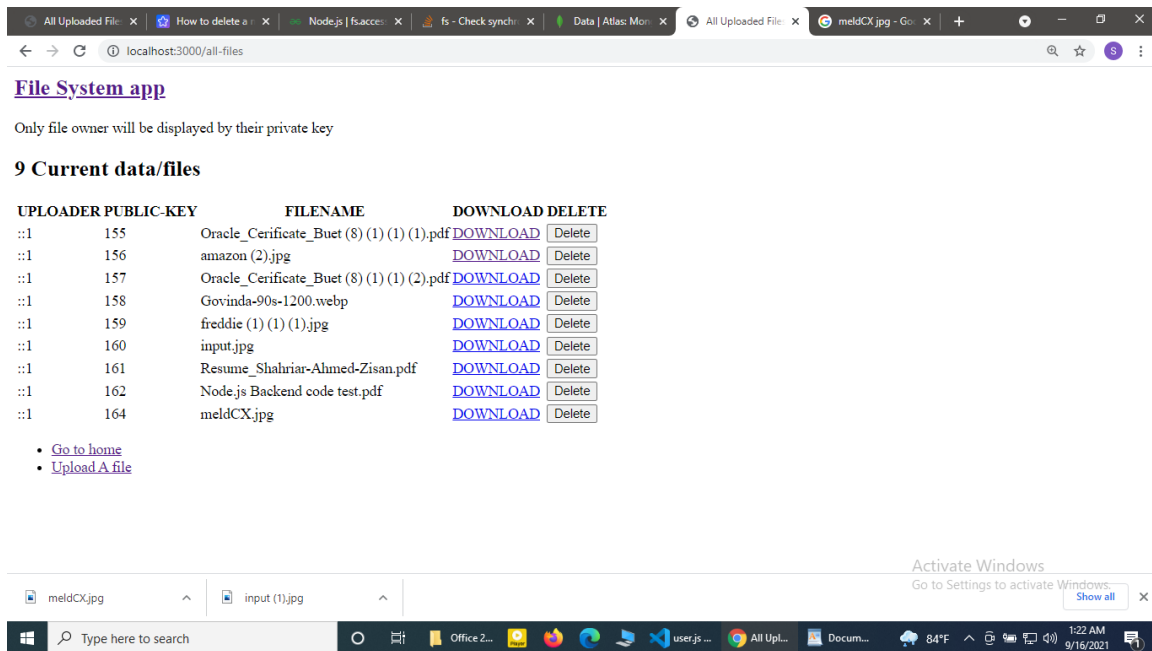


As shown below, the meldCX.jpg has been successfully uploaded, the page renders the information of the file as json object and some navigation links,

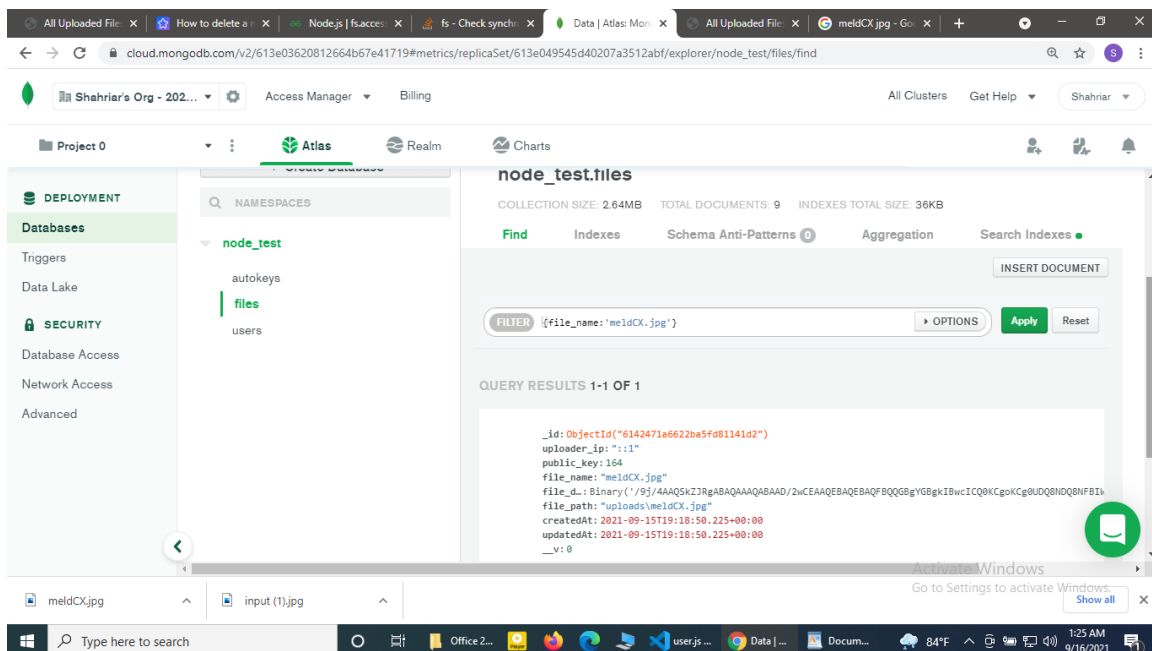
- 4) Now I will go to "List of uploaded files" link to see all the files stored in the database



Our meldCX.jpg file has been uploaded to the database and it's also showing all the files which were previously uploaded



MONGODB screenshot, showing the file was successfully uploaded in collection "files"



5) Now let's try to download the meldCX.jpg file. We need to click the DOWNLOAD link

**File System app**

Only file owner will be displayed by their private key

**9 Current data/files**

UPLOADER	PUBLIC-KEY	FILENAME	DOWNLOAD	DELETE
::1	155	Oracle_Cerificate_Buet (8) (1) (1) (1).pdf	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	156	amazon (2).jpg	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	157	Oracle_Cerificate_Buet (8) (1) (1) (2).pdf	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	158	Govinda-90s-1200.webp	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	159	freddie (1) (1) (1).jpg	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	160	input.jpg	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	161	Resume_Shahriar-Ahmed-Zisan.pdf	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	162	Node.js Backend code test.pdf	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>
::1	164	meldCX.jpg	<a href="#">DOWNLOAD</a>	<a href="#">Delete</a>

- [Go to home](#)
- [Upload A file](#)

localhost:3000/files/164

meldCX (1).jpg

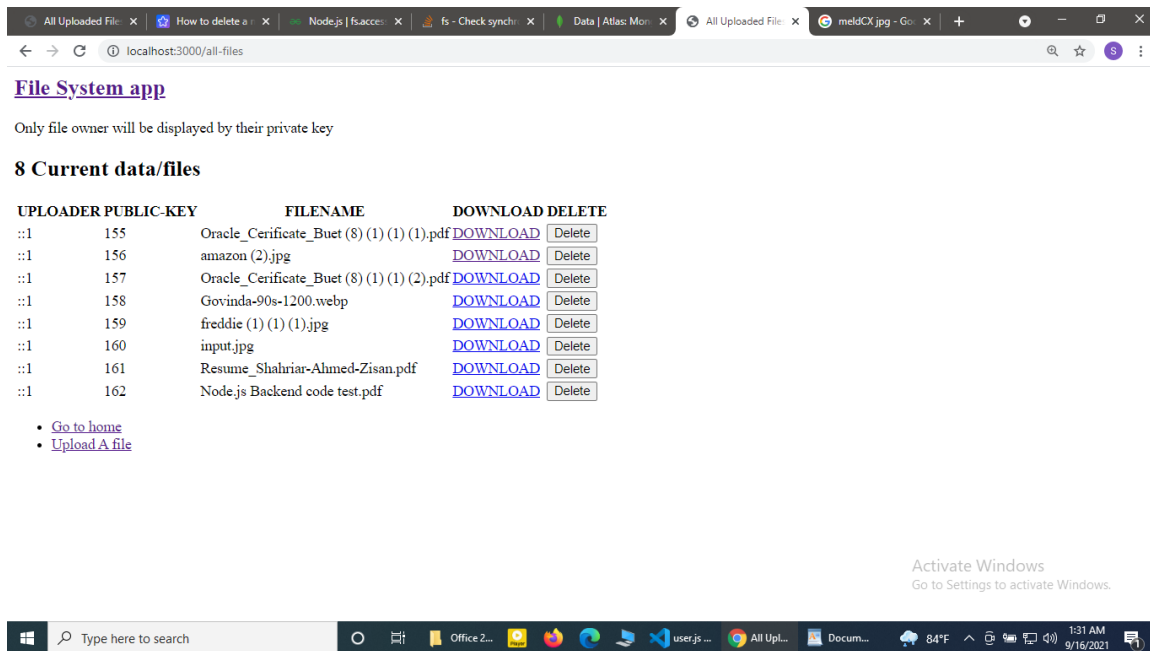
Activate Windows  
Go to Settings to activate Windows.  
[Show all](#)

Type here to search

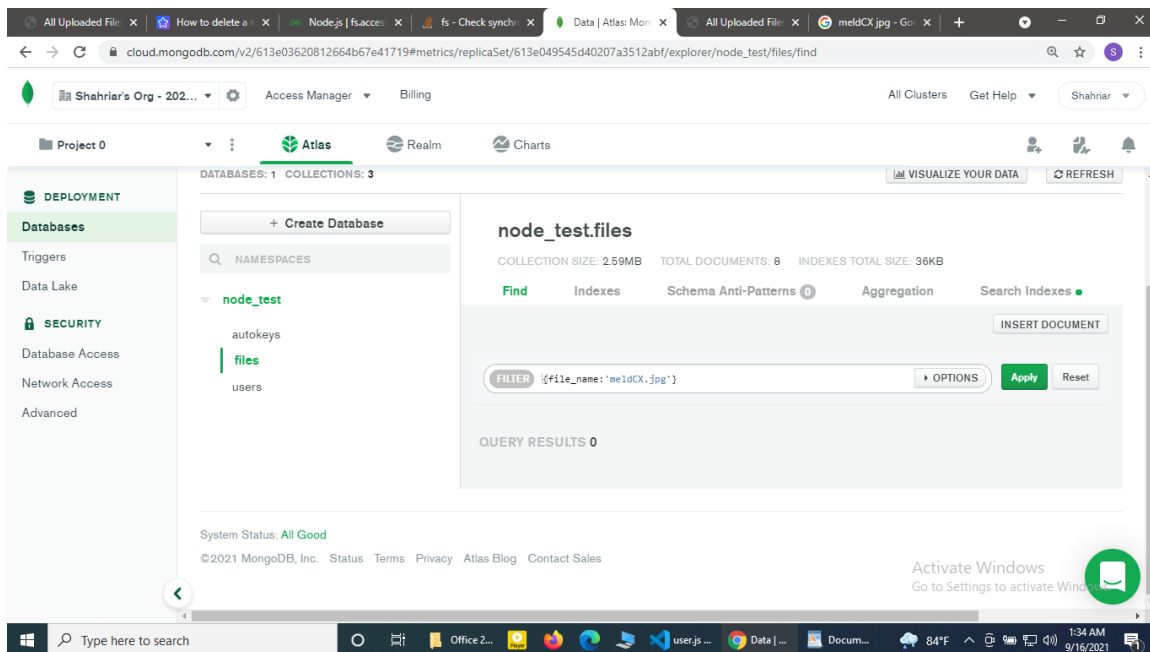
Office 2... user.js ... All Up... Docum... 84°F 1:27 AM 9/16/2021

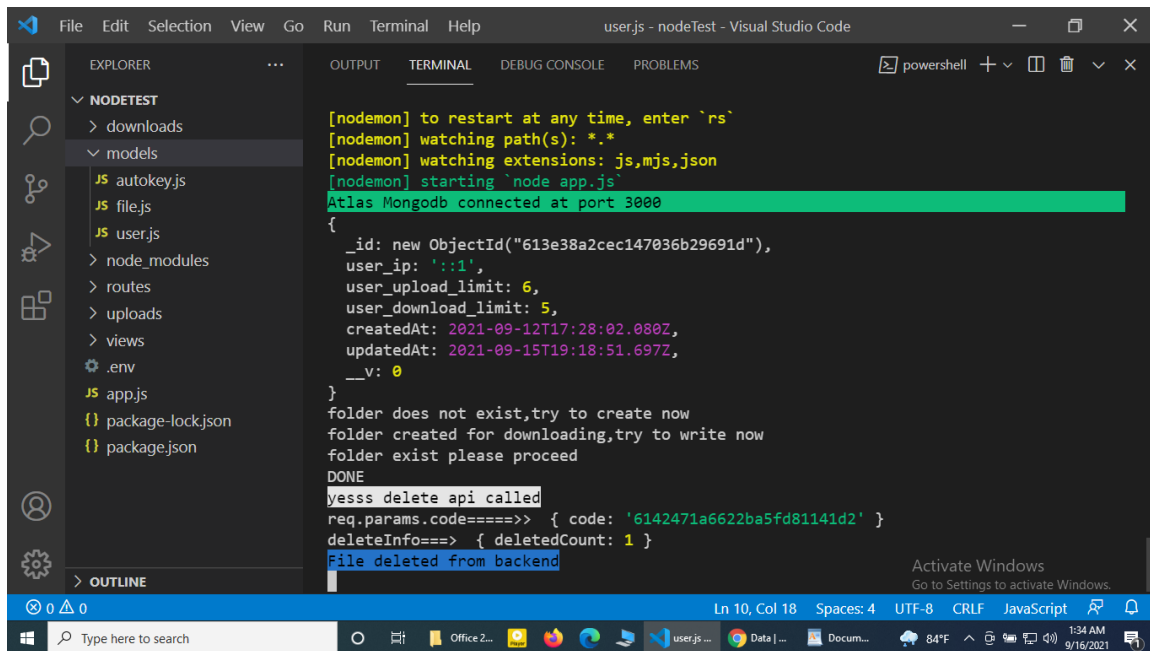
Notice that meldCX has been downloaded and OS has confirmed the download

6) Now I will be deleting "meldCX.jpg", for that I need to click the DELETE button.



Note that "meldCX.jpg" has been deleted. For confirmation, I have provided the mongoDB screenshot and VSCODE console.log



The image is a screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named 'NODETEST' with a 'models' folder containing 'autokey.js', 'file.js', and 'user.js'. The main editor area displays a terminal window with the following output:

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Atlas MongoDB connected at port 3000
{
  _id: new ObjectId("613e38a2cec147036b29691d"),
  user_ip: ':::1',
  user_upload_limit: 6,
  user_download_limit: 5,
  createdAt: 2021-09-12T17:28:02.080Z,
  updatedAt: 2021-09-15T19:18:51.697Z,
  __v: 0
}
folder does not exist,try to create now
folder created for downloading,try to write now
folder exist please proceed
DONE
yesss delete api called
req.params.code====>> { code: '6142471a6622ba5fd81141d2' }
deleteInfo====> { deletedCount: 1 }
File deleted from backend
```

The status bar at the bottom indicates 'Ln 10, Col 18' and 'Spaces: 4'. The Windows taskbar is visible at the very bottom.

### Some notes to consider:

- I could not manage the feature to reset "daily upload and download" for lack of time. But It could be easily done by writing a mongodb scheduled function by some sort of cron to reset the schema field 'user\_download\_limit' and 'user\_upload\_limit' after some predefined time or interval.
- Another key feature is "clean internal storage". It has been implemented but it will execute if we uncomment the line from which it is calling the async function runMyCron().The cron regex is hardcoded for now and if modification is required, I could use the regex as an environment variable to set interval for internal storage.

Thanks,

Shahriar Ahmed Zisan