Fig.1. UI-server, backend-server and http-server monitoring the uploads folder

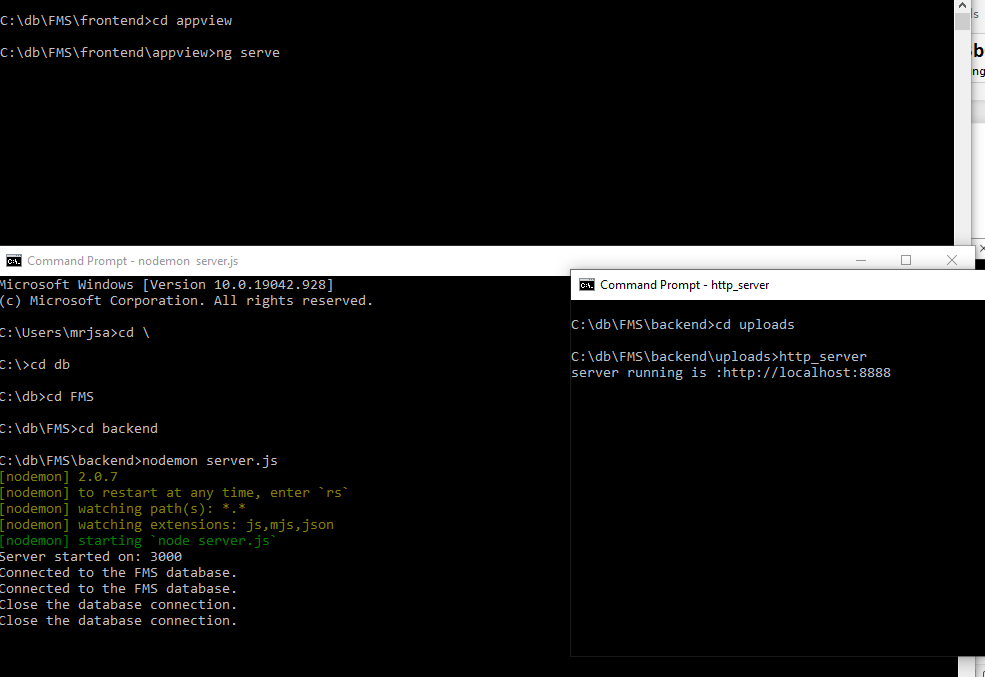


Fig.2.1. Multiple Uploads

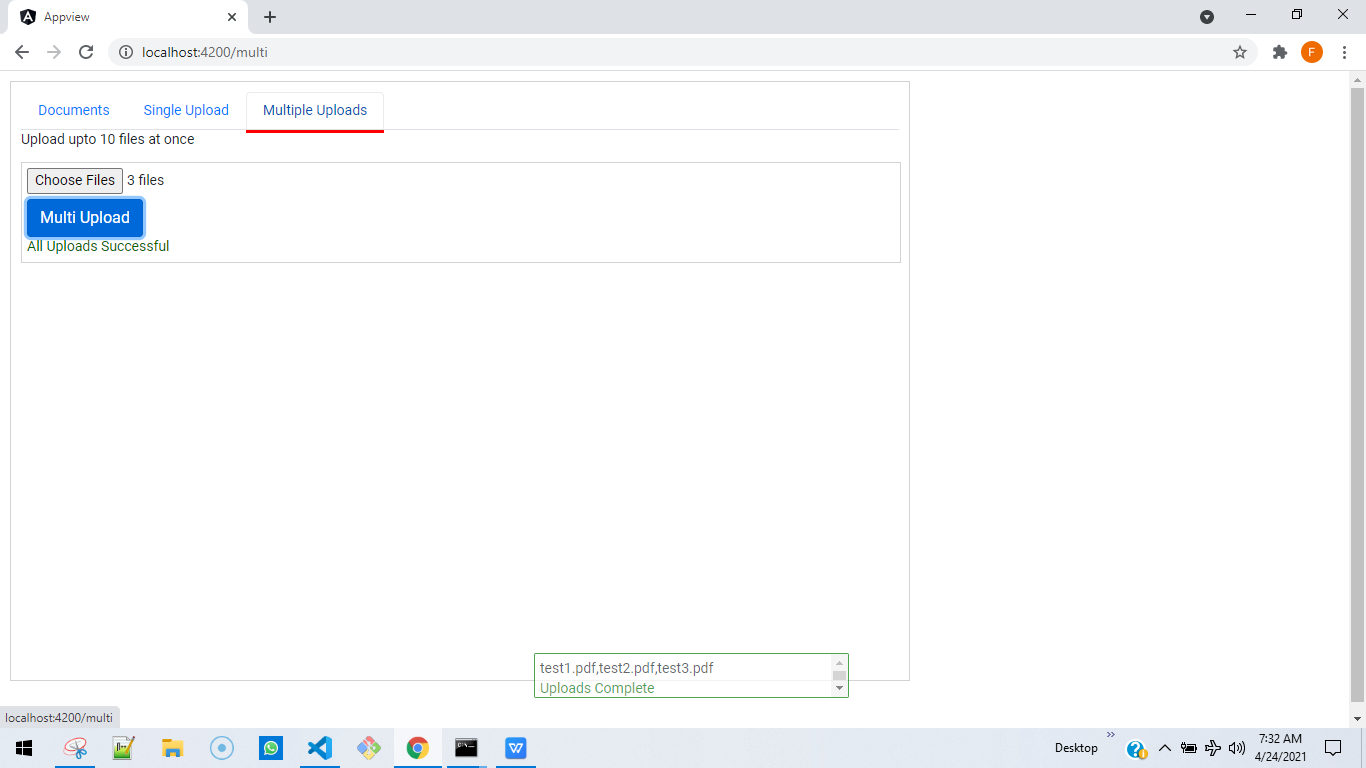


Fig.2.2 Single Upload

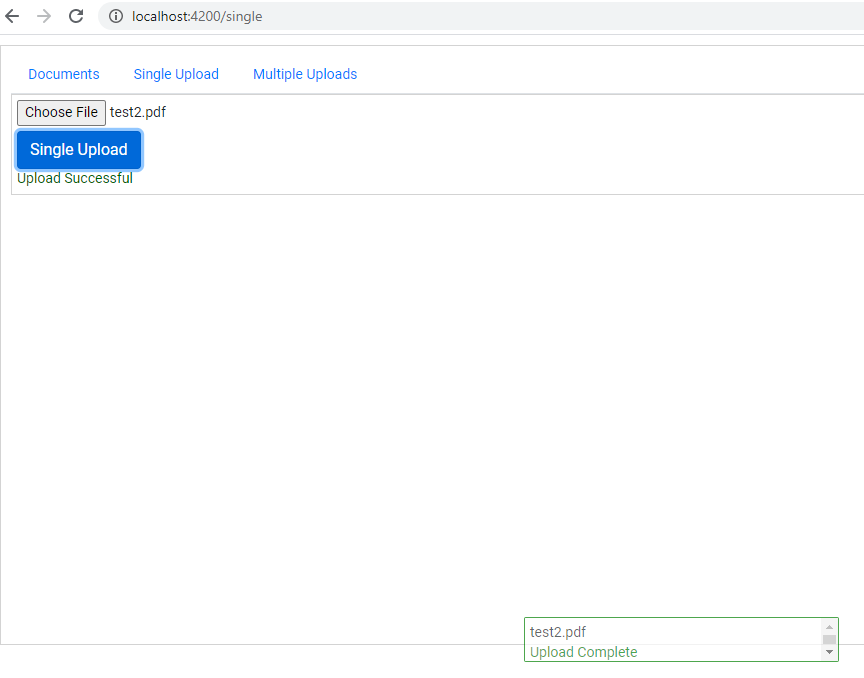


Fig. 2.3 Documents listing. Reads from the uploads/ folder. Download link requires the http\_server to be running on this folder. If http\_server has not been installed use “ npm install http\_server -g “

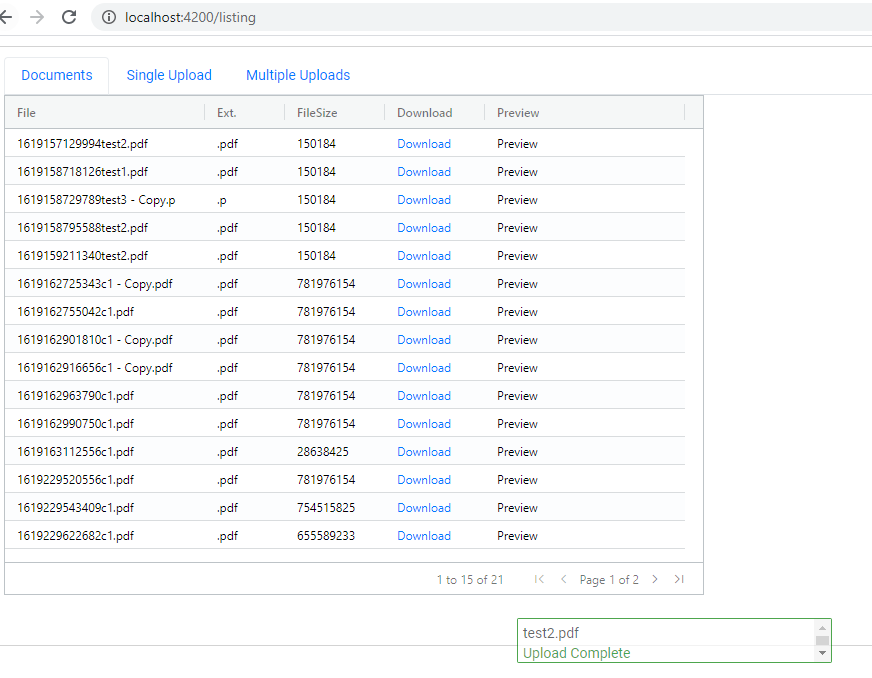
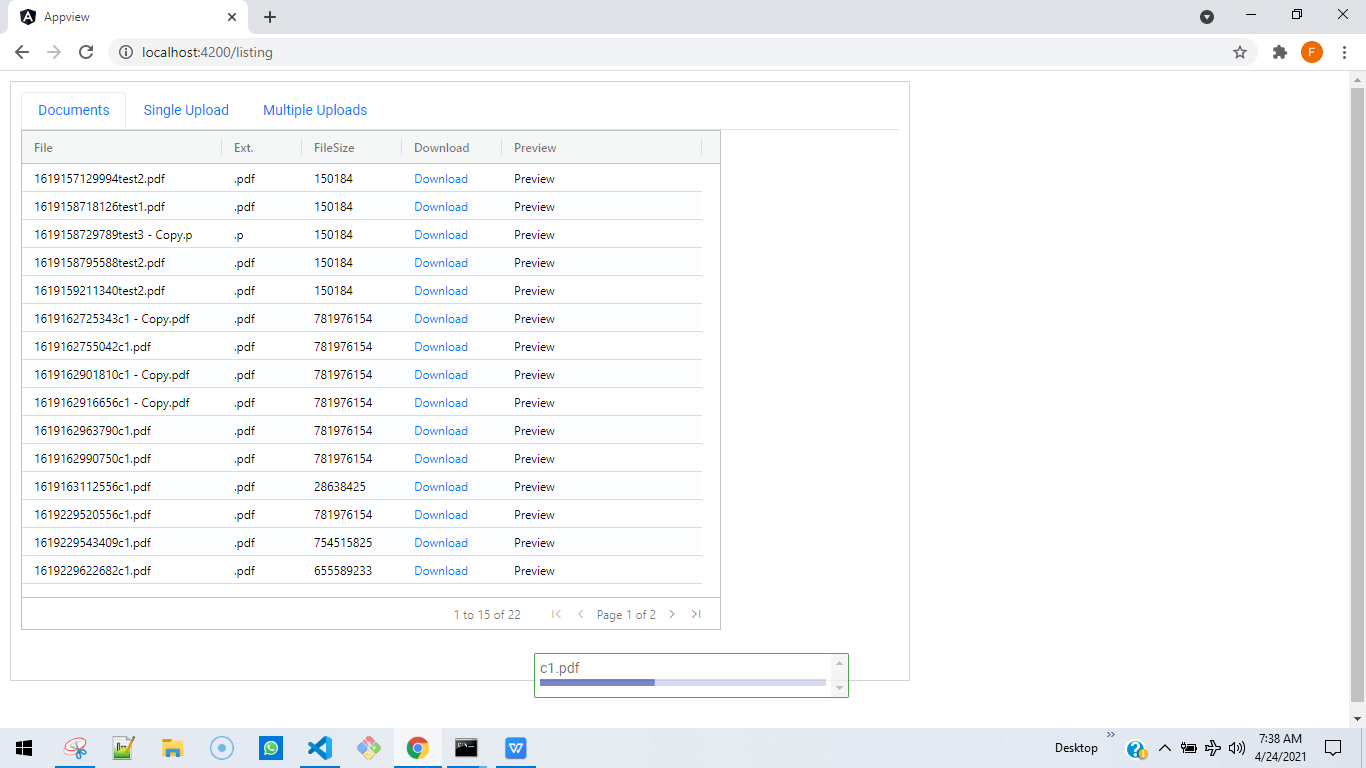
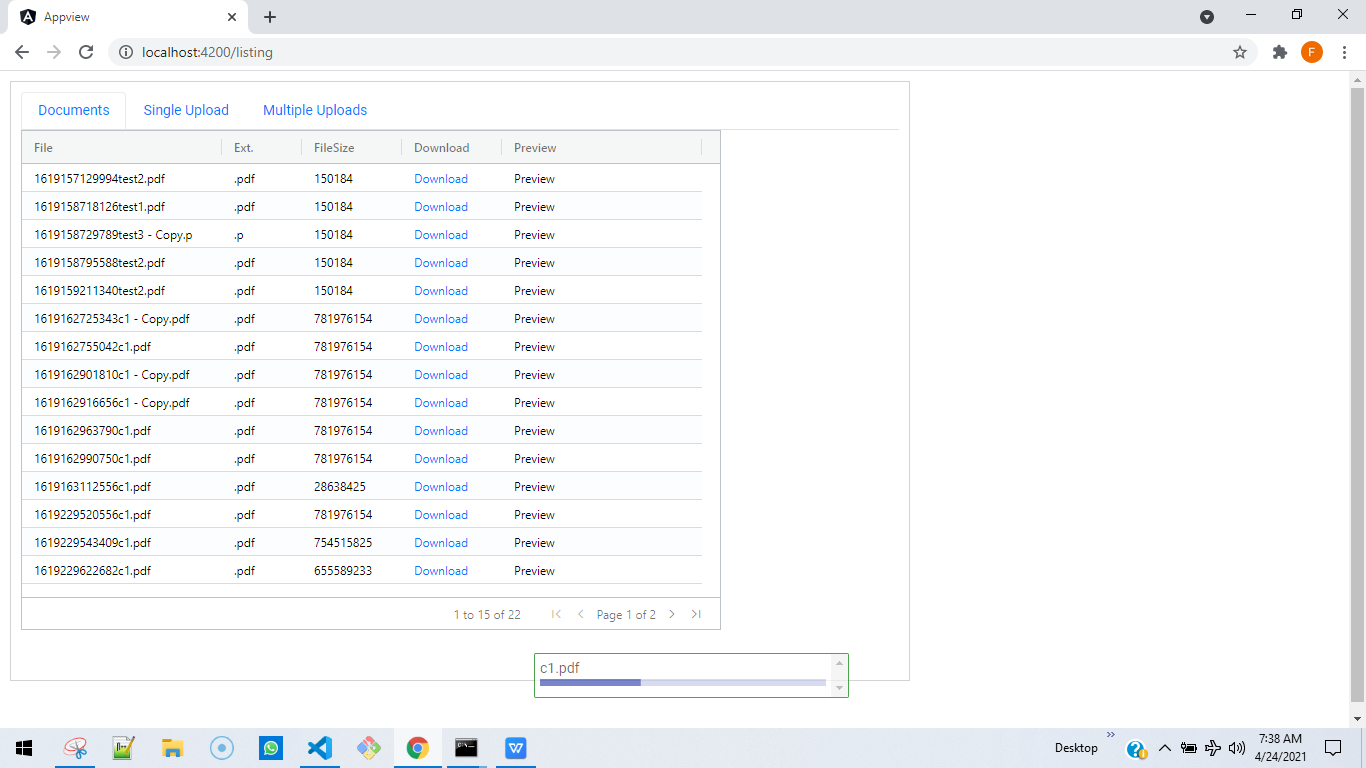


Fig. 2.4 If upload is going on and u still navigate to the Listing page, u should be able to see upload happening on Documents tab.





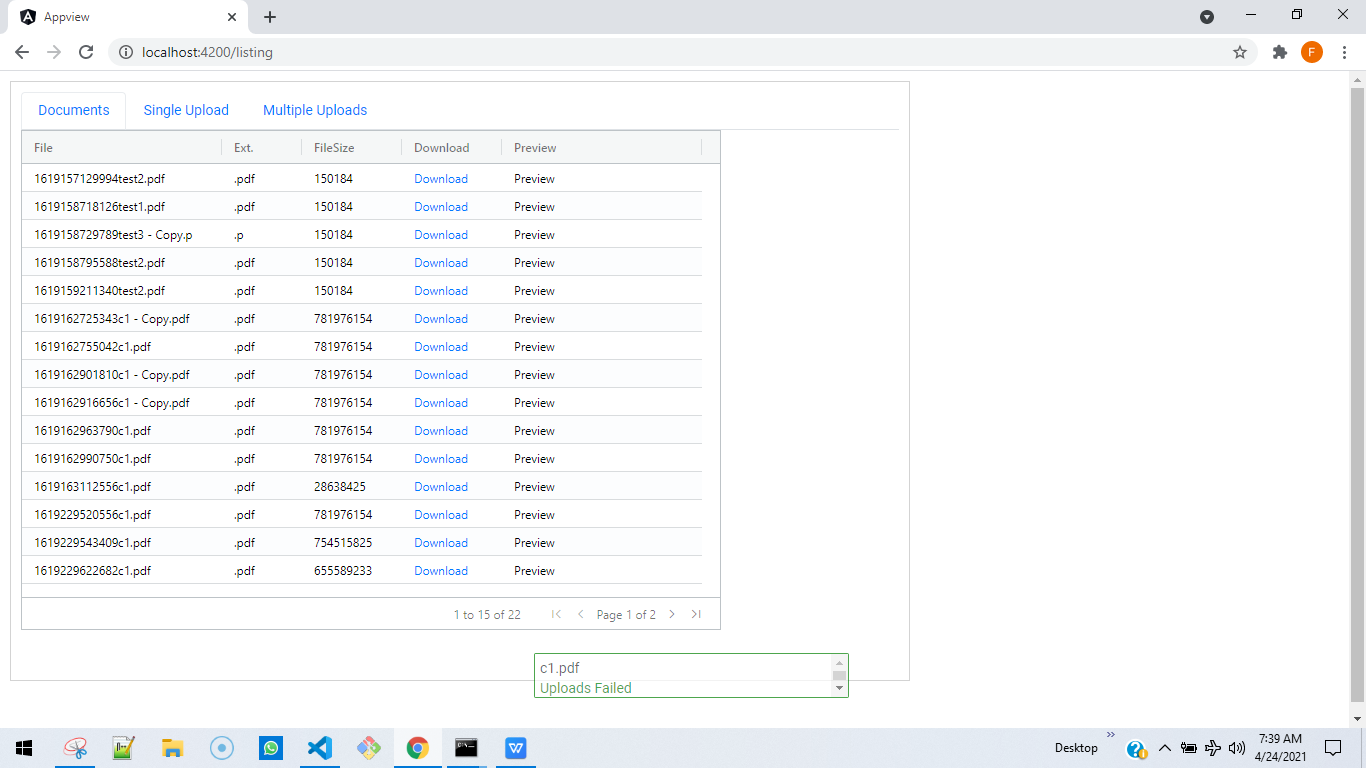
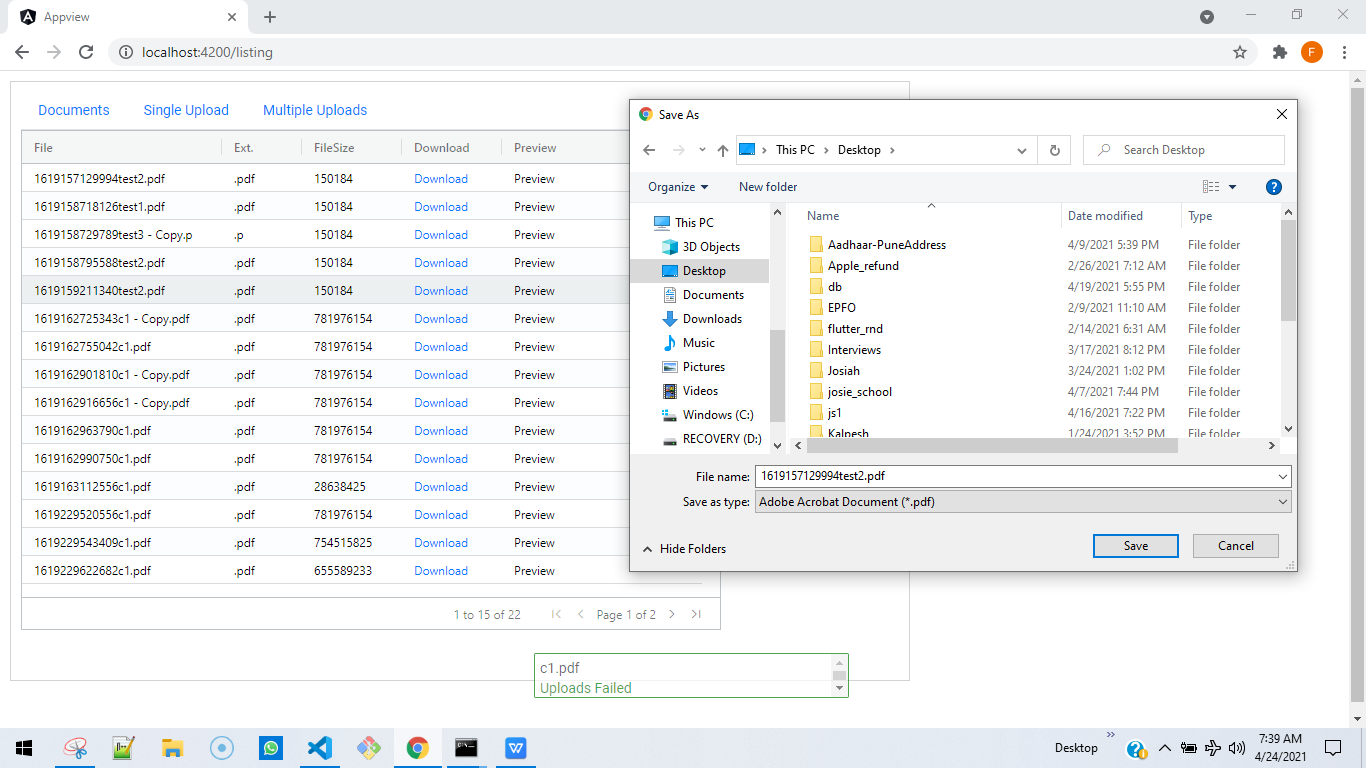


Fig. 2.5. Files can be saved using the Download -> Save As option



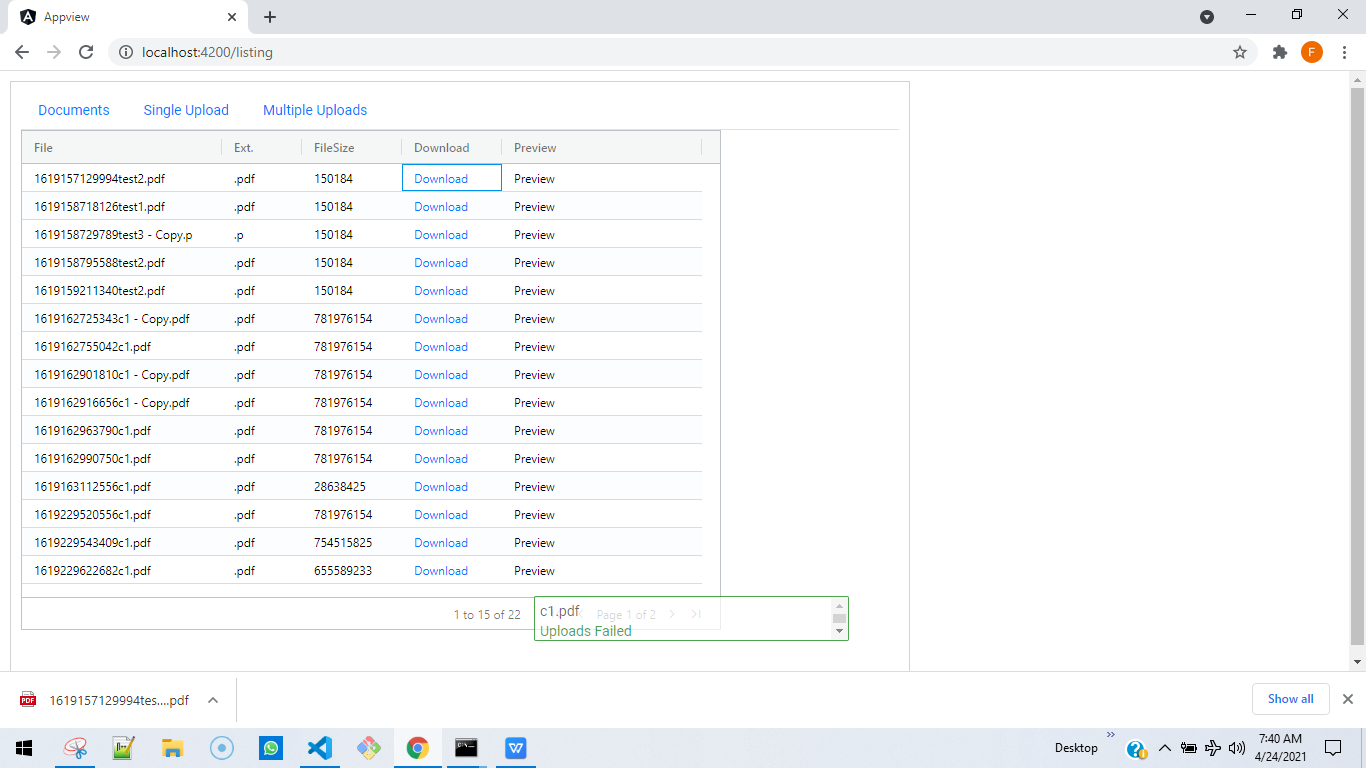


Fig. 2.6.

Click on Preview to Preview the document

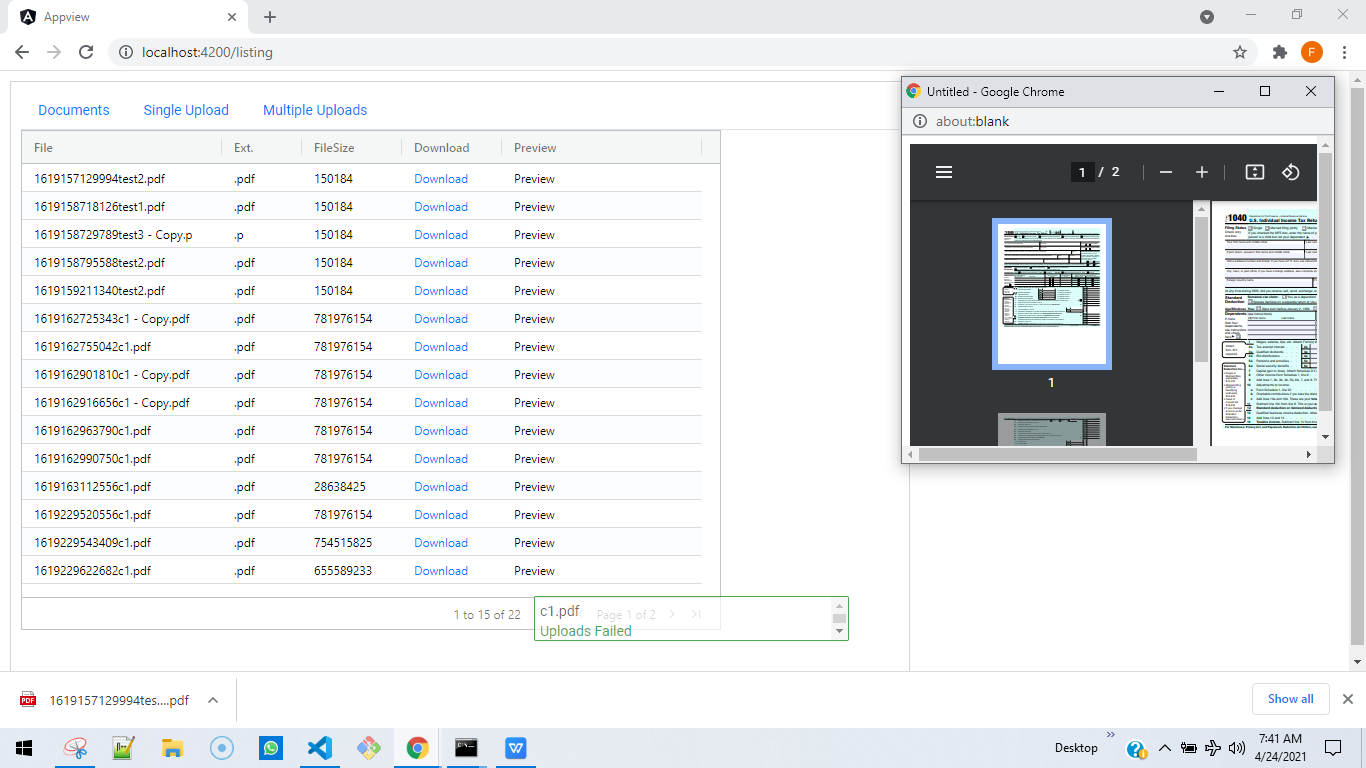


Fig. 2.7. Files more than 200MB are not allowed.

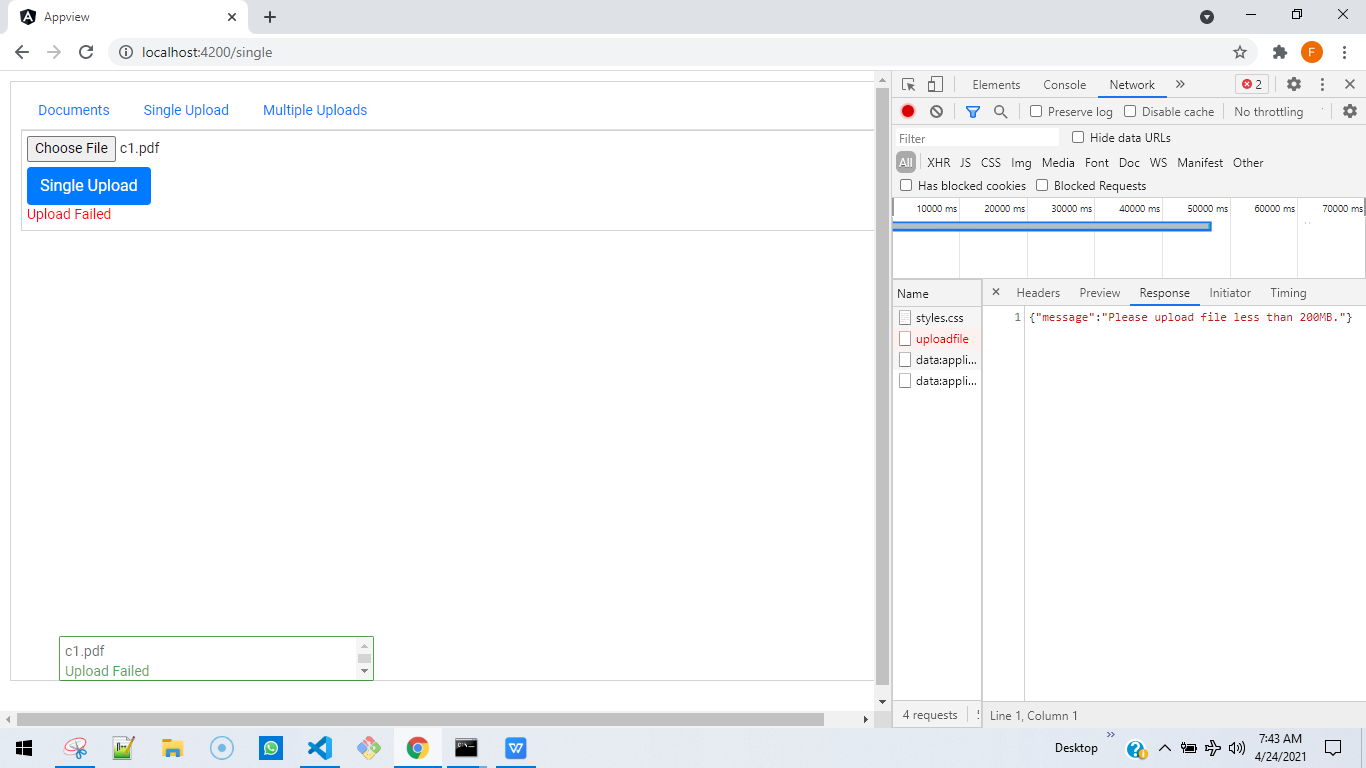


Fig.2.8 Backend upload APIs are unit-tested using Jest.

cd C:\db\FMS\backend

npm test

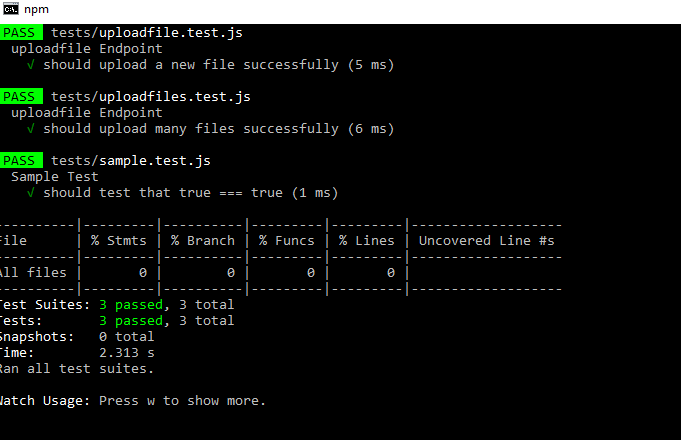
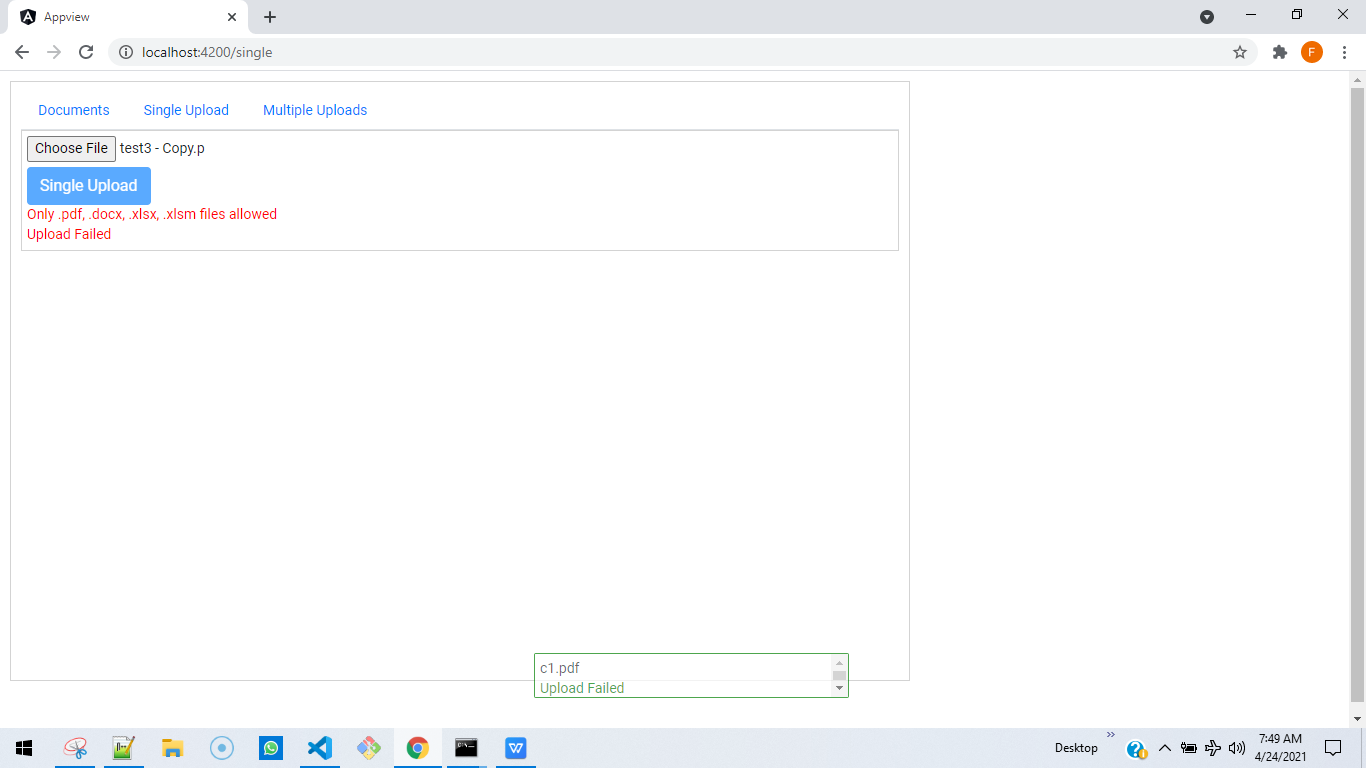


Fig. 2.9

Only .pdf, .docx. .xlsx, .xlsm files are allowed.

If any other file is attempted to be uploaded then the upload-button gets disabled and error is shown as below.



Retry and select a valid file extension and it works as before.

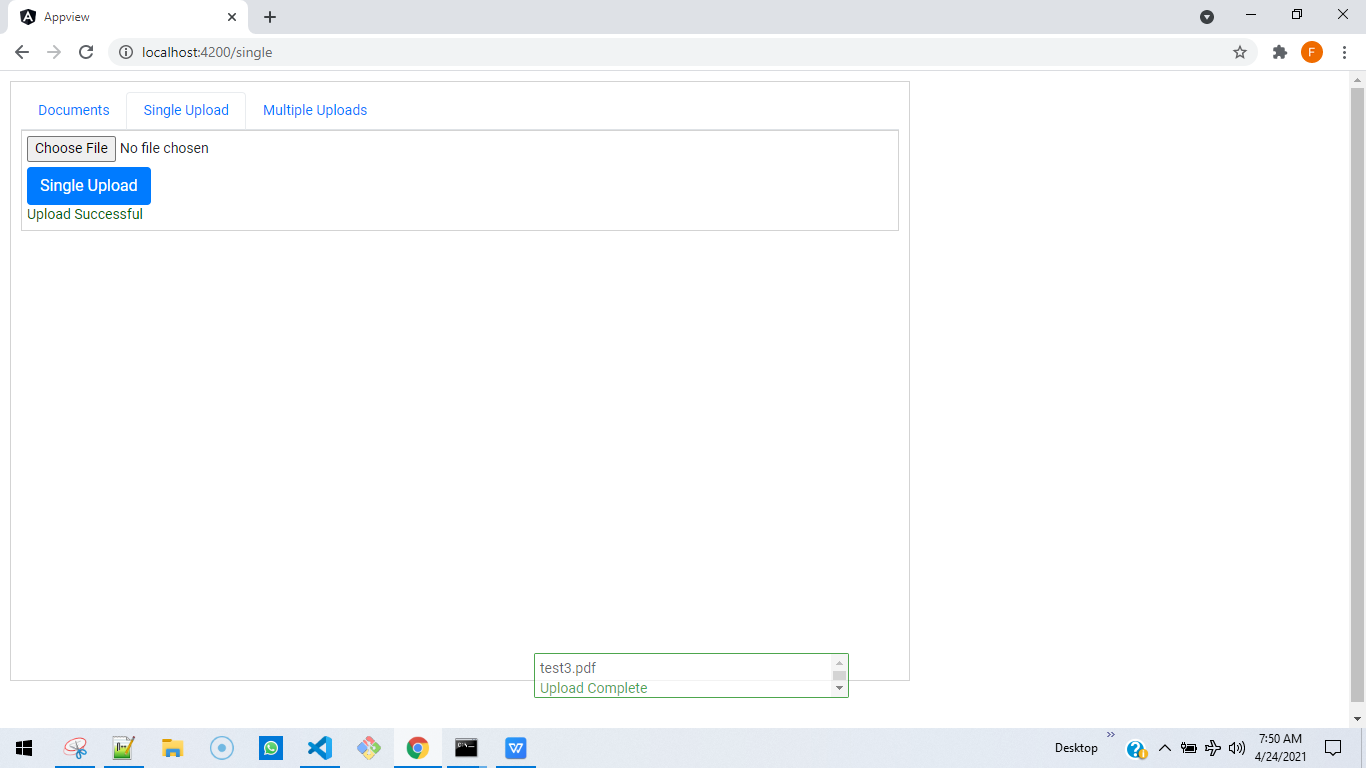


Fig. 3.

Backend Details:

1. Multer is being used for uploading the files to the uploads/ (c:/db/FMS/backend/uploads/) folder.
2. http\_server provides a service over this folder
3. Sqllite 3 is used as an offline db and is available in FMS.db file. Select.js queries and can additional data from it, but I didn’t have enough time to merge these additional data to the API response
4. 3 APIs
   1. /uploadfile See uploadfile.test.js POST API
   2. /uploadfiles See uploadfiles.test.js POST API
   3. /getAllfiles

Fig.4.

Frontend Details:

1. Ag-grid
2. Bootstrap
3. Angular Material Progress bar
4. Angular 11