

Github links:

Backend: <https://github.com/R4K0/exif-backend>

Frontend: <https://github.com/R4K0/exif-stripper> (front end **includes** the back-end as a sub-module)

G00376658 GMIT

Purpose of project:

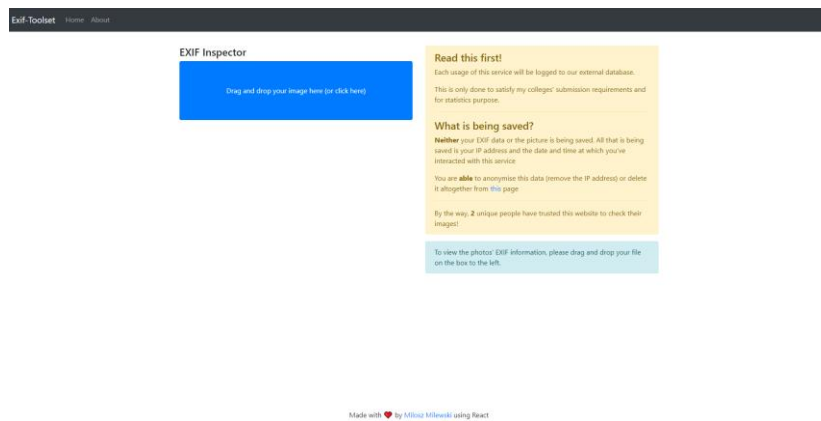
React app built on the MERN (Mongo, Express, React, Node) stack.

The purpose of this app is to let users safely explore EXIF data of their jpeg files and if possible, remove it.

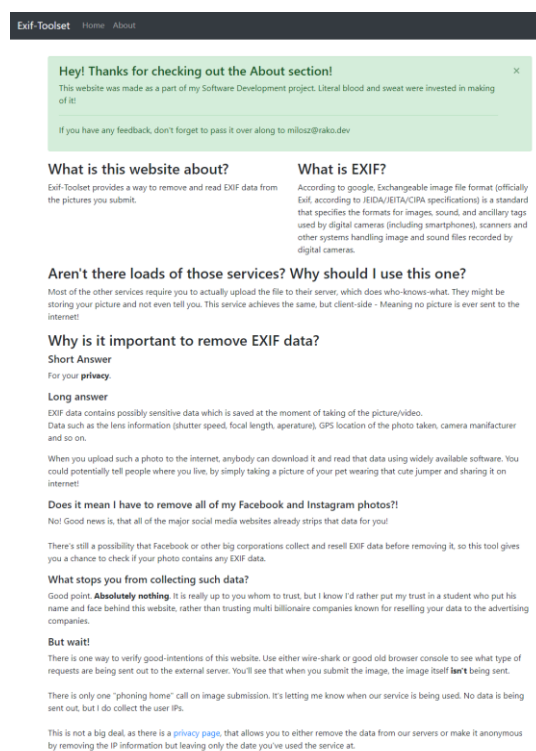
The user first drag and drops (or clicks the button that opens a file upload prompt) to select which file do they want to load. The site then displays all of the detected EXIF data to the user, and lets him download a new image file, without the EXIF data.

Additionally, major dependencies used: Bootstrap; exif-parser and piexifjs for manipulating exif data

Home page:



About page:

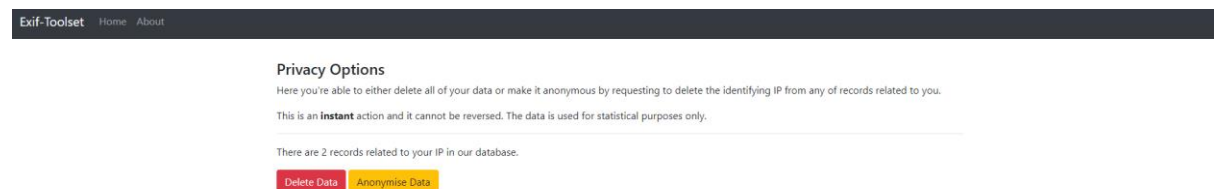


EXIF data for an example picture:

< 1 2 3 4 >

Data Header	Data Value
Make	FUJIFILM
Model	FinePix6900ZOOM
Orientation	1
XResolution	72
YResolution	72
ResolutionUnit	2
Software	GIMP 2.4.5
ModifyDate	1217524676
YCbCrPositioning	2
Copyright	

Privacy page, where you can clear any information collected about you, the user



So, summarizing:

There are 5 components.

- About page
- Privacy page
- Upload page
- Navbar
- EXIF Display table

Rest of the elements either come from Bootstrap or are defined within the pages themselves.

About the back end:

It is included in the project as a sub-module.

For the API endpoints, it uses rate limiting to minimize abuse, there are 5 endpoints in the total:

GET /statistics/count - Takes the requester IP and counts the records in the database with such an IP, returns how many records there are.

GET /statistics/distinct - Gets the count of all the unique IPs there are in the database.

POST /statistics/add – Doesn't take any params, it gets the requester IP and adds a new document to the database with the IP of the requester and the date the request came in for

POST /statistics/anonymize – Doesn't take any params, it looks for all records done by this requester IP and unsets the IP from those records, making those records anonymous.

DELETE /statistics/delete – Doesn't take any params, it looks for all the records done by this requester IP and DELETES those records.

The back-end serves REACT files from one folder up (../build), so it is expected that exif-database is a folder within exif-stripper, maintaining the folder structure as the project was pulled from the GitHub. Because of this, the installation procedure is quite complicated.

As per GitHub readme, to deploy this application you must follow those steps:

Deployment:

1. CD into the exif-stripper folder
2. `npm install` to install all the dependencies
3. Run `npm run build` to build the project. The project should have created the build folder with the production-ready app in it.
4. CD into the exif-backend folder and run `npm install` to download dependencies for the backend server.
5. While still in the exif-backend, run the `node main.js`
 - The backend server expects the "build" folder to be one folder up the hierarchy. Example: backend is in `exif/backend/` then it looks for the React build folder in `exif/build/`

Configuration:

The server will listen on `process.env.PORT` or will default out to 3000

Mongo details can be set in: `config/database.json`, but if `process.env.mongo_connecturl` is defined, then it'll use that over json to connect