# Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

## Guidelines

Provided below is a template you must use to write your report for each of the technologies you use in your project.

Here are some things to note when working on your report, specifically about the **General Information & Licensing** section for each technology.

- **Code Repository**: Please link the code and not the documentation. If you'd like to refer to the documentation in the **Magic** section, you're more than welcome to, but we'd like to see the code you're referring to as well.
- **License Type**: Three letter acronym is fine.
- **License Description**: No need for the entire license here, just what separates it from the rest.
- **License Restrictions**: What can you *not* do as a result of using this technology in your project? Some licenses prevent you from using the project for commercial use, for example.
- **Who worked with this?**: It's not necessary for the entire team to work with every technology used, but we'd like to know who worked with what.

Also, feel free to extend the cell of any section if you feel you need more room.

If there's anything we can clarify, please don't hesitate to reach out! You can reach us using the methods outlined on the course website or see us during our office hours.

## Multer

### General Information & Licensing

| | |
|---|---|
| Code Repository | https://github.com/expressjs/multer |
| License Type | MIT |
| License Description | ● For commercial/personal use<br>● The software can be modified in any way shape or form once held by the user<br>● The software can be distributed or used in any way once in the users hands. |

| | |
|---|---|
| License Restrictions | • The software is under no warranty<br>• The creators of the software are not liable for any damages incurred by the misuse of the software |
| Who worked with this? | Refat Ullah |

*Use as many of the sections below as needed, or create more, to explain every function, method, class, or object type you used from this library/framework.*

# require (“multer”)

## Purpose

- Multer is the middle-ware we used to handle all the image uploading functionalities
- It was used to set the destination where the images will be uploaded to.
- We set a limit of 10MB to each of the profile pictures using multer.
- It was also used to verify the file type of the uploaded file to make sure the uploaded file is JPG, JPEG, or PNG.

## Magic

For the multer to store the image, firstly we have to define the storage destination. In the following code, we are creating a diskStorage with the destination "./media/" and the file name as the original final name. So this code will upload the pics into the media directory with the original file name of the pics:

```
11    const imageStorage = multer.diskStorage({
12        destination: './media/',
13        filename: (req, file, cb) => {
14            cb(null, file.originalname)
15        }
16    });
```

Now we have to declare a multer function with the imageStorage and the file size limits. The following code also implements a file filter which will check if the uploaded file is having JPG, JPEG, or PNG extension:

```
18    const imageUpload = multer({
19        storage: imageStorage,
20        limits: {
21            fileSize: 1024 * 1024 * 10
22        },
23        fileFilter(req, file, cb) {
24            if (!file.originalname.match(/\.(jpg|jpeg|png)$/)) {
25                return cb(new Error('Only image files are allowed!'), false);
26            }
27            cb(undefined, true)
28        }
29    })
```

Finally, the imageUpload function is called to trigger the upload. Notice the ".single" in the function call. This makes sure that only a single file will be uploaded to the server.

```
33    router.post('/image-upload', imageUpload.single('image'), (req, res) => {
```