

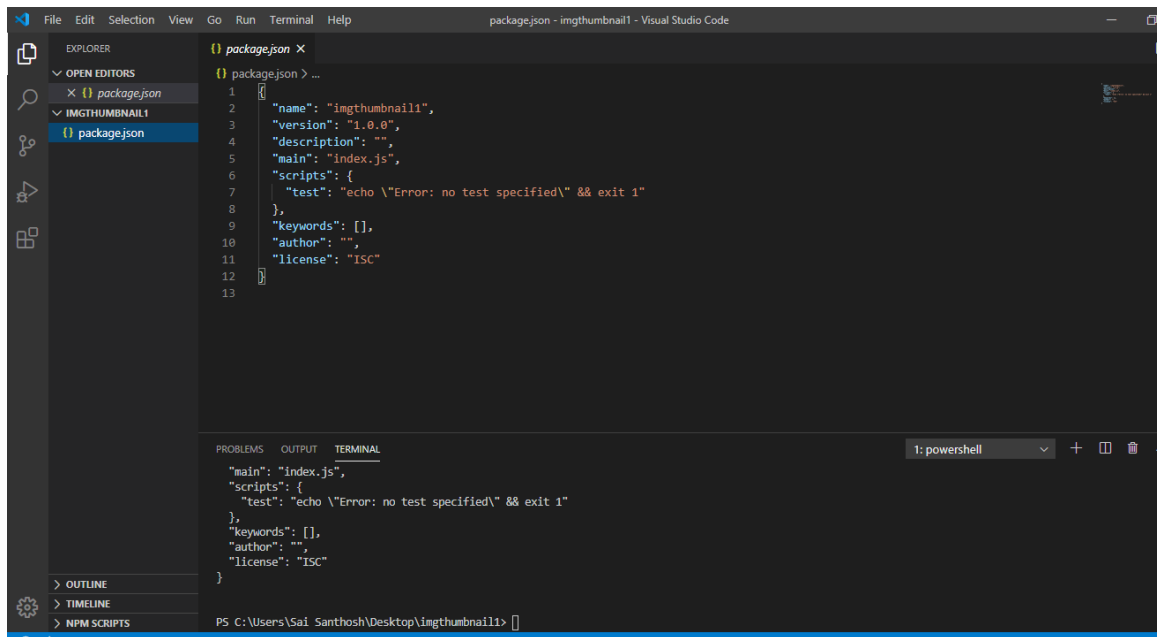
## MODULE 2

### IMAGE THUMBNAIL USING NODEJS

We use gm(Graphic Magick and Image Magick four our Project)

Intially we need a json file so we use

`npm init -y`



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'imgthumbnail1' with a 'package.json' file. The main editor area displays the content of 'package.json' with the following JSON structure:

```
1 {
2   "name": "imgthumbnail1",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC"
12 }
13
```

At the bottom, the TERMINAL panel shows a PowerShell prompt at the directory 'PS C:\Users\Sai Santhosh\Desktop\imgthumbnail1>'. The status bar at the bottom right indicates the active window is '1: powershell'.

The other libraries used for the project are:

Express->Server

Body-parser->To handle user requests

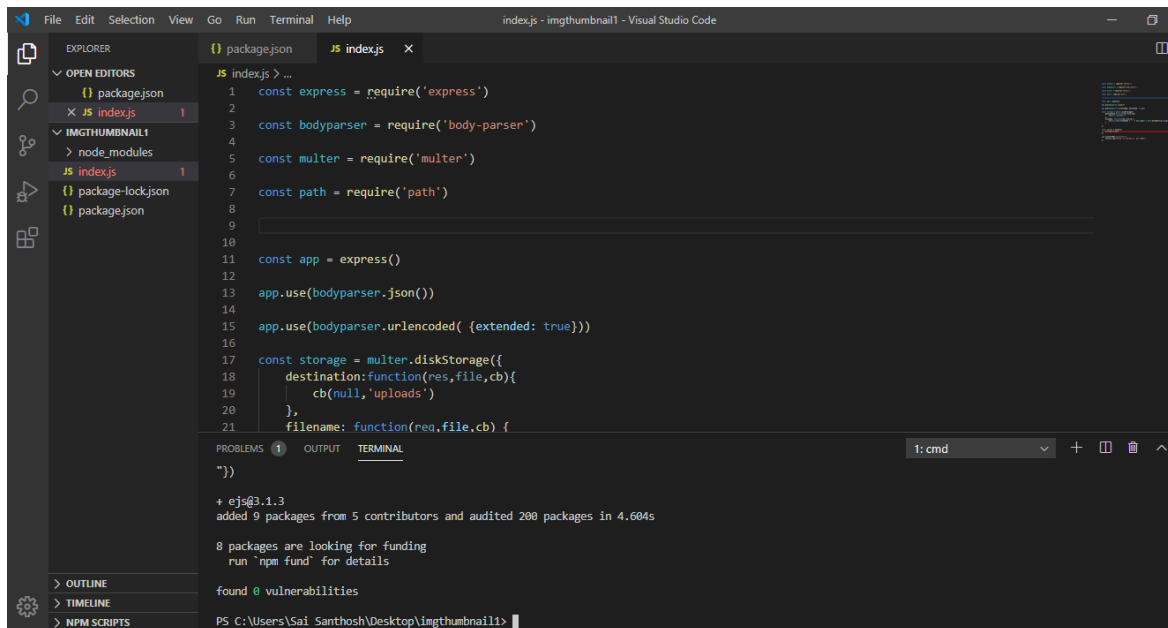
Multer->Image upload

Image-size->to set the dimensions

Nodemon->It's automatically restarts the server

Ejs

Installing required libraries:



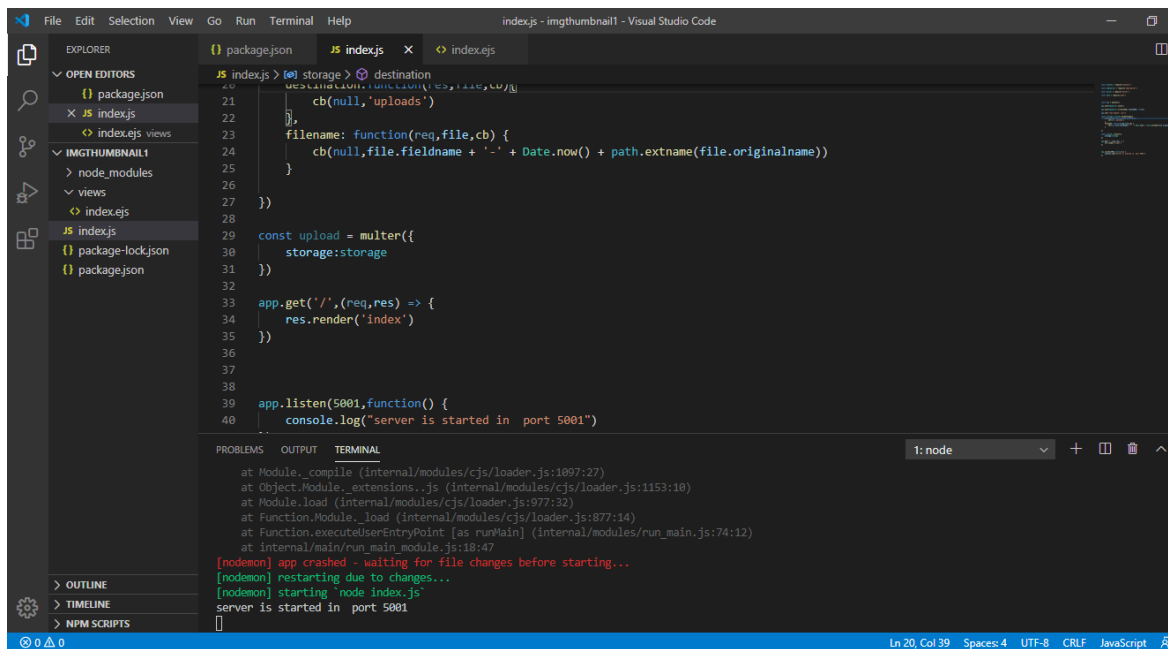
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar shows the file structure: package.json, index.js, node\_modules, package-lock.json, and package.json. The main editor area displays the index.js file with the following code:

```
1 const express = require('express')
2
3 const bodyparser = require('body-parser')
4
5 const multer = require('multer')
6
7 const path = require('path')
8
9
10
11 const app = express()
12
13 app.use(bodyparser.json())
14
15 app.use(bodyparser.urlencoded({ extended: true}))
16
17 const storage = multer.diskStorage({
18   destination: function(res, file, cb) {
19     cb(null, 'uploads')
20   },
21   filename: function(req, file, cb) {
```

The terminal at the bottom shows the output of the command `npm install`:

```
PS C:\Users\Sai Santhosh\Desktop\imgthumbnail1> npm install
+ ejs@3.1.3
added 9 packages from 5 contributors and audited 200 packages in 4.604s
8 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
```

We have created a index file and performing required operations on our libraries



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar shows the file structure: package.json, index.js, node\_modules, package-lock.json, and package.json. The main editor area displays the index.js file with the following code:

```
21   cb(null, 'uploads')
22 }
23 filename: function(req, file, cb) {
24   cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname))
25 }
26 })
27
28
29 const upload = multer({
30   storage: storage
31 })
32
33 app.get('/', (req, res) => {
34   res.render('index')
35 })
36
37
38
39 app.listen(5001, function() {
40   console.log("server is started in port 5001")
41 })
```

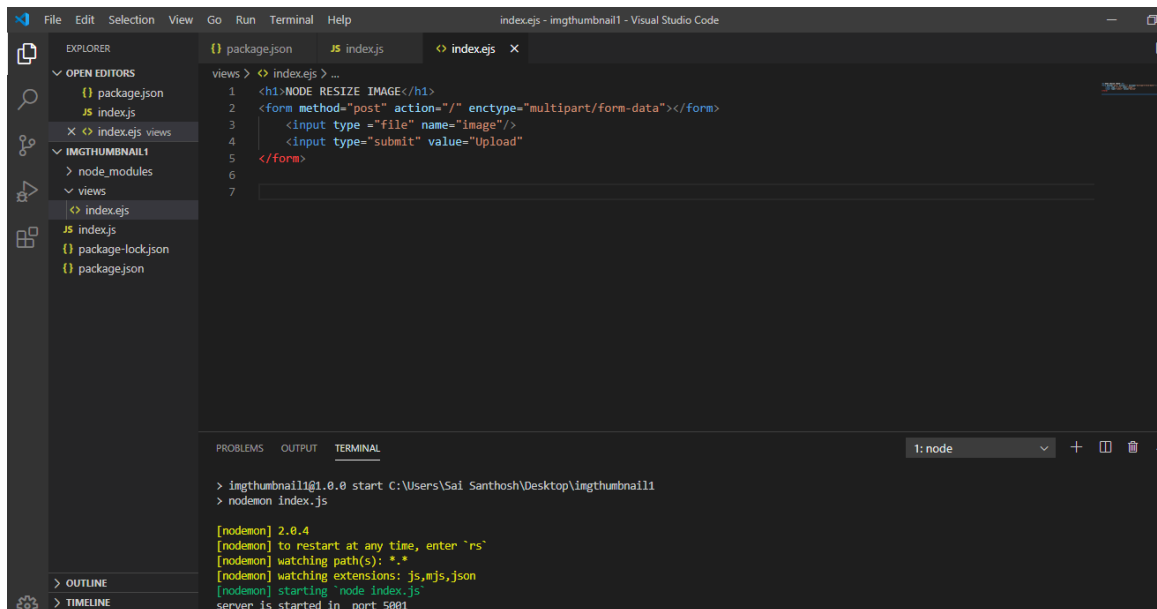
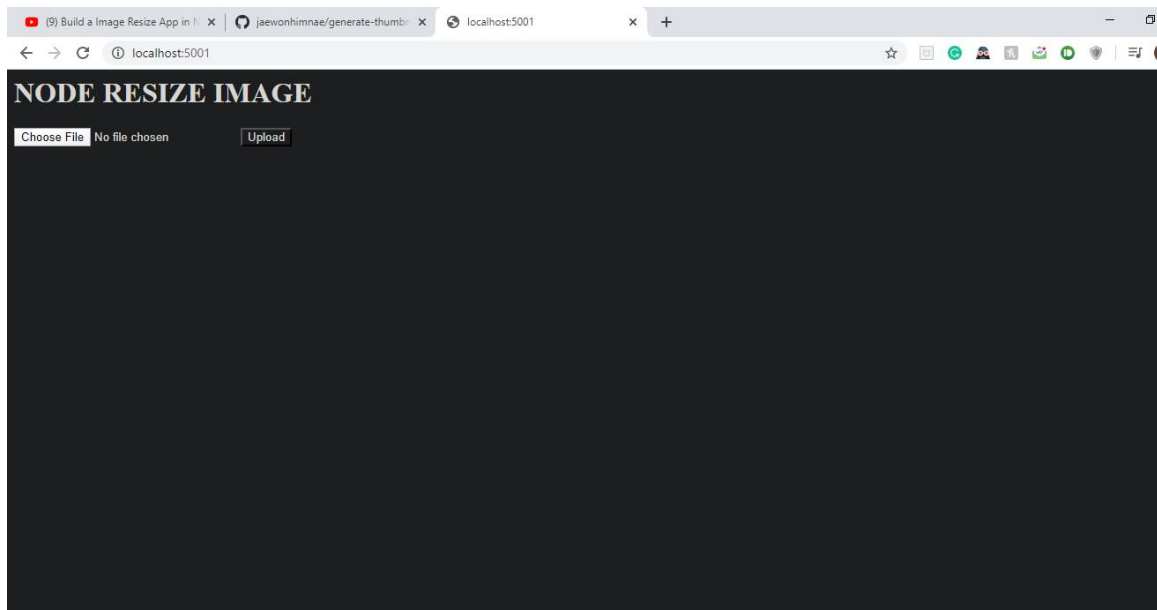
The terminal at the bottom shows the output of the command `node index.js`:

```
at Module._compile (internal/modules/cjs/loader.js:1097:27)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1153:10)
at Module.load (internal/modules/cjs/loader.js:977:32)
at Function.Module._load (internal/modules/cjs/loader.js:877:14)
at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:74:12)
at internal/main/run_main_module.js:18:47
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting "node index.js"
server is started in port 5001
```

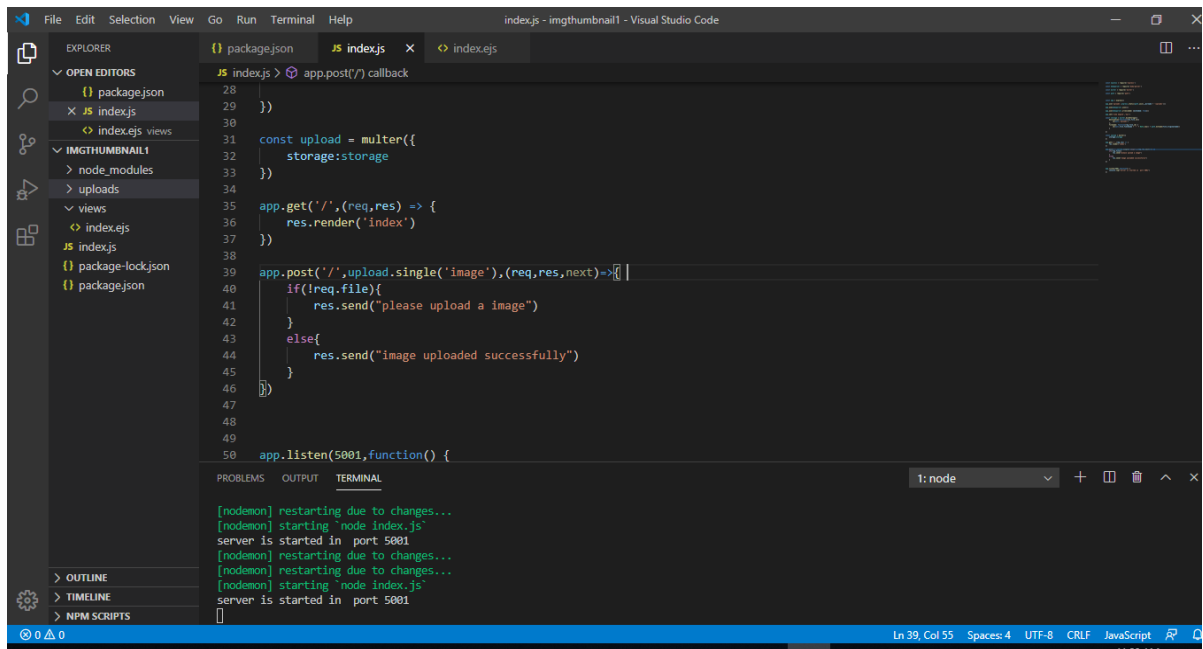
I have also created a view folder and index.ejs file to carry out the HTML files

As we are using port number 5001

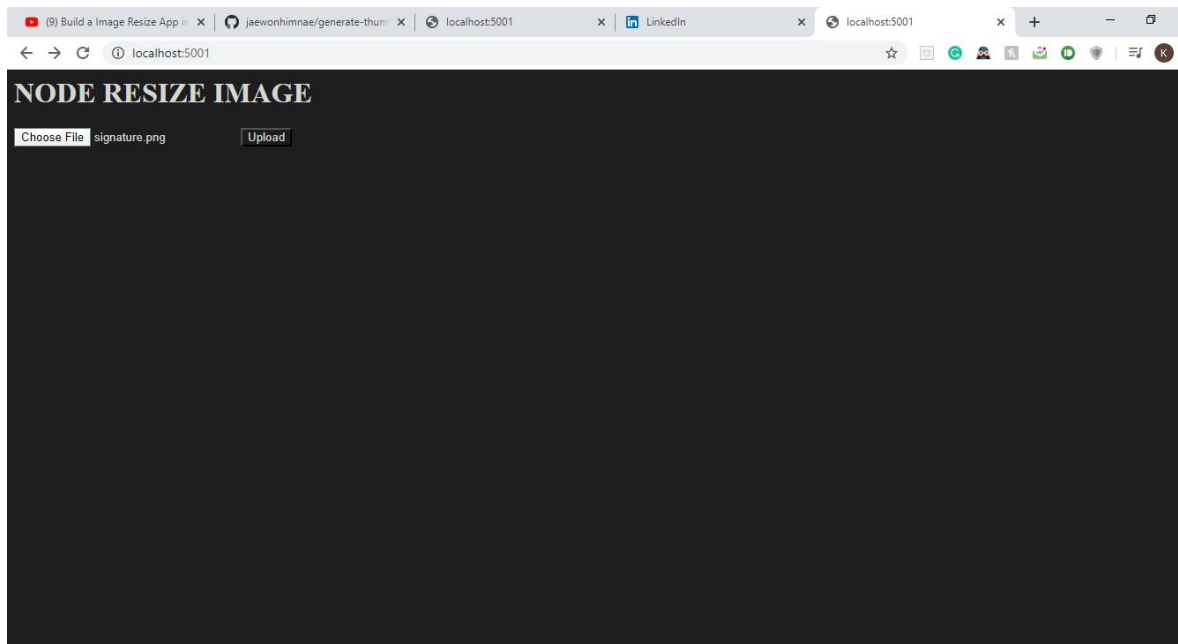
So we should go to localhost:5001 and we get it as

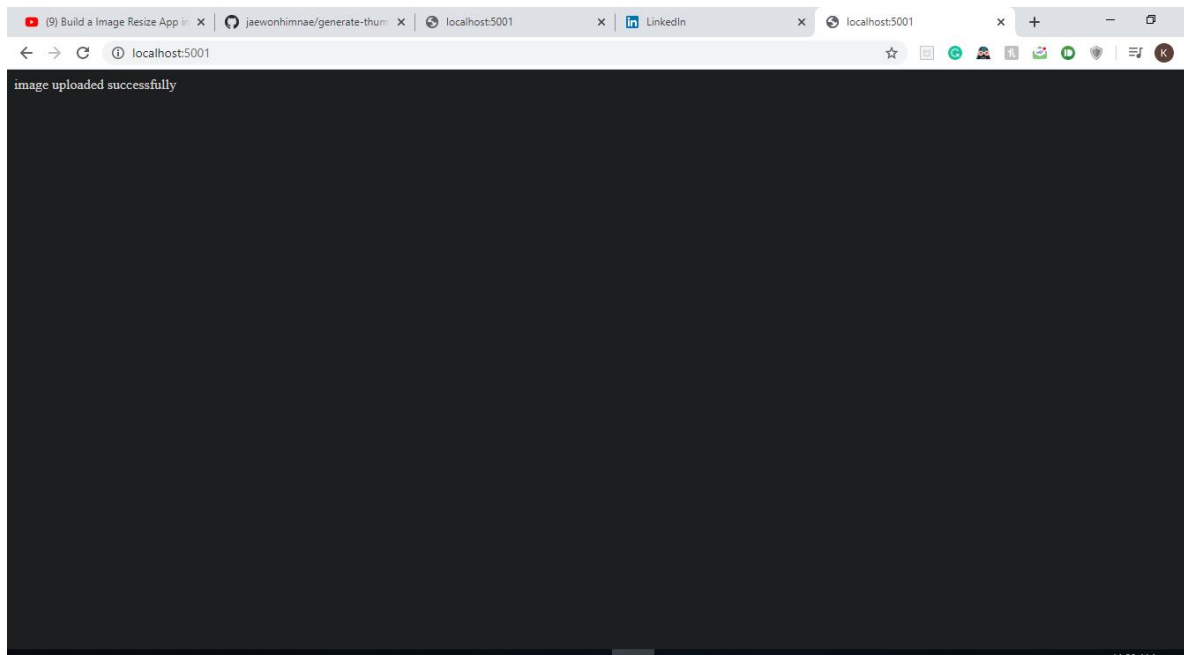


I have also created a Static Folder in which all the files which are uploaded is placed

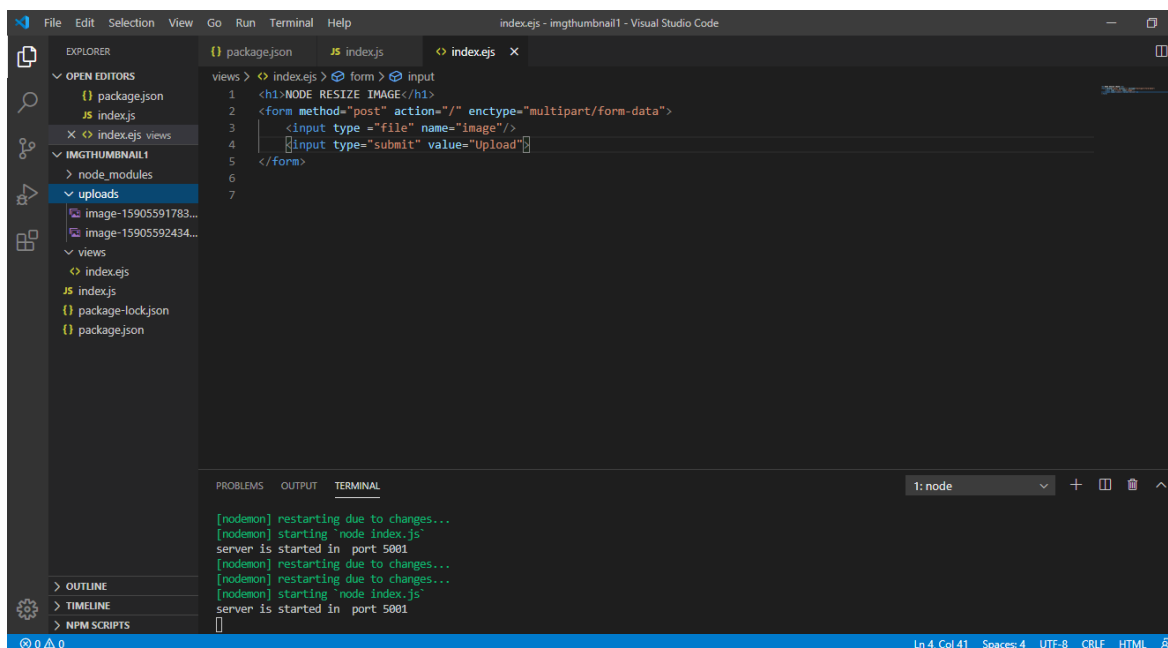


We can upload any kind of image like .png,.jpg etc such as:





The files that we have uploaded are store in local static folder named uploads

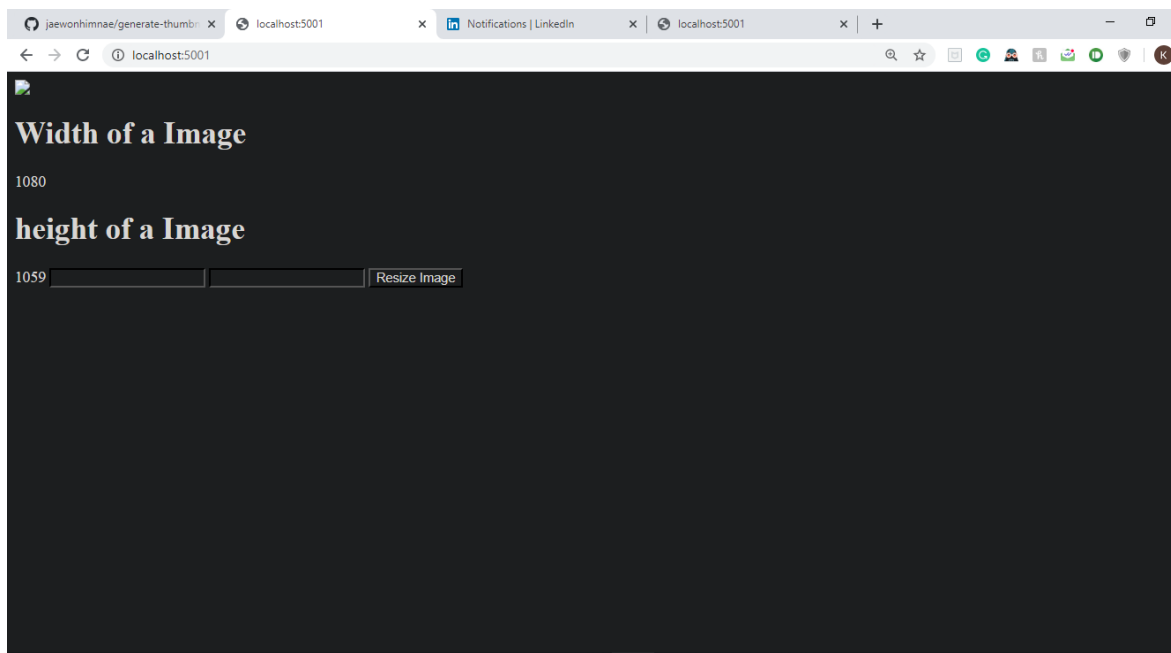


Till now we have uploaded only the pictures but now we can able to see the images.

We have taken the required dimensions and we can resize it as we required

The home would be as shown:

After uploading the file it will ask for the dimensions



The screenshot shows a web browser window with the address bar displaying 'localhost:5001'. The page has a dark background and contains the following elements:

- A small logo in the top left corner.
- The text 'Width of a Image' in a large, bold, serif font.
- The number '1080' below the width text.
- The text 'height of a Image' in a large, bold, serif font.
- The number '1059' followed by an empty text input field.
- A 'Resize Image' button to the right of the input field.