Lab2 - HTTP server

1. **Provide Server to List Employees**
- Use a filesystem (Use the **stream** module).
- Both Lab01, Lab02 should use the same JSON file (used as database).
- You should have 4 pages
    - **Page One: Home page localhost:3000/** it's what the users see when they first land on our website. It'll show our employees list that the user entered from yesterday (just Viewing). (don't show the id)
    - **Page Two: Astronomy page: localhost:3000/astronomy**
        - Should display this image along with a few lines of description.
    - **Page Three: Serbal page: localhost:3000/serbal**
        - Should display this image along with a few lines of description.
    - **Page four: 404 page not found (just simple html)**

    - **Add an employee: localhost:3000/employee (using curl or postman)**

- **Style** your page, serve the css file **separately**, from disk.
- Use read **streams**.

Bonus:
- Add a new route to download this image. (trigger the download dialog) **localhost:3000/astronomy/download**

    Hint: look-up for the correct header ;)

- **Page Five: Add an employee, localhost:3000/employee**

Notes:
- **Make sure to try the debugging environment**
- **Make sure that you enable restart on change using nodemon**
- **Make sure to split your logic into modules**
- Please make sure that you separate your logic in functions
- Make the names of your functions and variables are expressive of what they are
- Start by defining each functionality and the steps you should make to achieve it
- Please make sure not to pollute the global scope[refrain from using global variables]
- Before writing a function, ask yourself, is there a function in javascript that can do that, if not, write your own.

Tips:
- If you can't do something, search, then ask for help, this is not an exam.
- Please please please, don't copy someone else's code.

Useful Links:
- npm | build amazing things
- Nodejs Docs
- Fs module docs fs (you will need to use the sync version of the functions)[ends with the word Sync]
- HTTP module to make your http server
- Nodemon for restarting node
- Headers - Web APIs
- Introduction to Node.js

Useful Readings:
- [Build an HTTP Server](#)
- [An introduction to the npm package manager](#)
- [The V8 JavaScript Engine](#)
- [How the Web works - Learn web development](#)
- [What Is a Pure Function in JavaScript?](#)
- [Circular dependency in Node.js and Nest.js](#) (read the Node.js part only)

Must read before Day3:
- [**REST API Best Practices – REST Endpoint Design Examples**](#)