- Lab3 - simple Express server

- Create an express server
- 1 - with the following endpoints

HTTP Method	route	Description	
GET	/todos	Return the list of todos	
GET	/todos/:username	Return list of todos for certain username	bonus
POST	/todos	Accepts title, username If username exists, and logged in, create it else return an descriptive error message Return ({message: "todo created successfully"},	//blue is bonus
DELETE	/todos/:id	Delete the todo with selected id Enhancements He has to own the todo, and has to be logged in	
PATCH	/todos/:id	- Edit the todo with the selected id Accepts {title:"",status:""} - Return ({message:"todo was edited successfully"}) if success - Return ({error:"invalid attributes"}) if user tries to edit a non existing attribute Enhancements he has to own the todo and logged in	
post	/register	- Register a user with the following **required** attributes Username,password , firstName Notes: - Write the user to a file asynchronous - Return ({message:"user was registered successfully"}) if success - Return ({error:"{Attribute name} is required" if there is a validation error with 422 status code	bonus
Post	/login	Return ({message: "logged in successfully", profile:{name:"username"}}) if the user exist in your file Change property LoggedIn: true Else Return ({error:"invalid credentials" }) with 401 status code	bonus

- 2 - Create a middleware that logs the request url, method, and current time

- 3 - Create a global error handler that logs the error and return {“error”:”internal server error”} with status code 500
- Use async methods. writeFile,
- Modularize your code into files.
- Test your api through postman

BONUS

After building your middleware, use a templating engine to view the data.

Notes:

- **Read about express static**
- **Read about express route params (you’ll find it in the routing page at express docs).**
- **Make sure to try the debugging environment**
- **Make sure that you enable restart on change using nodemon**
- **You can access the url from within the req param**
- Please make sure that you seperate 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
- Template Engines
- Express framework module to make your http server
- Postman for testing your APIS

Please send the labs to one of us and cc the other two, please write your name and ITI_ISMAILIA_41, lab03 in the title of the email, also the name of the folder and I'll reply to each e-mail with the notes and comments.
OR email me the github link, make sure to ignore the node modules folder
-- mohammedessehemy@gmail.com
-- ahmad.kholaif2@gmail.com
-- eng.loai.alaa.eg@gmail.com

-