**Roadmap for Backend**

**//Complete Glorifire training on Glorifire WebApp**

**Setup git and Bitbucket**

1. Install git with `sudo apt-get install git`
2. Follow these tutorials to setup bitbucket:

* [Git Notes](https://docs.google.com/document/d/1K--eDRiilkFZE6NuAcblYeL2XhbXAjHVAaGz8pkqGtc/edit?usp=sharing)
* <https://docs.google.com/document/d/11hKNrccnzWdHMSbcS1_D7owUaT56fyMq0tgGZzVz-Bs/edit>

1. More resources on learning git:

* <https://git-scm.com/>
* <https://docs.google.com/presentation/d/0B0ryzF27BotlNXVCVW5iUWpoUEk/edit?resourcekey=0-ml-X5TWArHHo_q3tm3BSYQ#slide=id.p1>
* <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

**Setup Node**

1. Install nvm first using: <https://github.com/nvm-sh/nvm#install--update-script>
2. Update your bashrc: `source ~/.bashrc`
3. Install node version 14 using: `nvm install 14`
4. Now you can use node: `nvm use 14`
5. Check for correct installation with `node -v` and `npm -v`

**Note:** Node will be used to run javascript code.

**Start Learning Javascript**

1. Follow the resources listed below to learn and practice javascript:

<https://javascript.info/>

<https://developer.mozilla.org/en-US/>

1. Use the following document to get yourself familiarized with code practices:

<http://docs.mypat.org.s3-website.ap-south-1.amazonaws.com/backend/code-practices.html>

**#Task 1:**

Create a calculator with js that can handle multiple variables and operations.

E.g input: “12\*2/12+23-12”, output: 13

**#Task 2:**

Create a deep object cloner with js.

E.g input:

input = {

name: "Bhupesh",

id: "1",

task: {

first: "task1",

second: "task2"

},

getName() {

return this.name;

}

}

Output: Cloned object with no reference to the original object.

**#Task 3:**

Design a microblogging message-based social network like Twitter.

Features - 1. Login/Logout (DON'T DO)

2. User should be able to follow and unfollow

3. Users should be able to tweet and tweet should be visible to all the followers

Tasks:

Design DB schema, mention collection and fields

Create a function to get the list of tweets posted by my friends of location ‘X’ (easy skip)

or

Top 3 users country wise who have posted highest no of tweets / get highest likes / get highest retweets

**Start Learning about Promises, async/await** (from the same resources)

**#Task 4:**

You are given an array x operations, you have to process them y promises at a time.

e.g Array of names you have to save, each saving to db is one operation and you have to perform 5000 operations.

You have limited resources and and you can parallelly run only "limit" promises at a time.

Write code to perform all promises. (take setTimeout as your asynchronous function)

**Setup node and start learning how to use node and express**

**#Task 5:**

Create a basic express app that can handle the following http methods: get, post, put, delete, patch.

**Setup MongoDB on your local machine**

Read about MongoDB from: <https://www.mongodb.com/docs/manual/tutorial/getting-started/>

Read about mongoose (npm module to communicate with mongodb): <https://mongoosejs.com/docs/guide.html>

Run the following commands:

1. *sudo apt install -y software-properties-common gnupg apt-transport-https ca-certificates*
2. *wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -*
3. *echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list*
4. *sudo apt update*
5. *wget* [*http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1\_1.1.1f-1ubuntu2\_amd64.deb*](http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2_amd64.deb)
6. *sudo dpkg -i libssl1.1\_1.1.1f-1ubuntu2\_amd64.deb*
7. *sudo apt-get install -y mongodb-org*
8. *sudo systemctl start mongod*
9. *sudo systemctl enable mongod*
10. Check for correct installation and running status with

*sudo systemctl status mongod*

*sudo ss -pnltu | grep 27017*

**Setup Robo3t on your local machine for MongoDB GUI**

1. Run:

*sudo snap install robo3t-snap*

1. Go to the gdm custom.conf file and uncomment `WaylandEnable=false`and then run:

*sudo nano /etc/gdm3/custom.conf*

1. Reboot your system and run Robo3t

**Setup Redis on your local machine**

1. Run `*sudo apt install redis-tools`*
2. Run `redis-cli*`* to access redis shell

**Setup ESLint for your project**

ESLint is used to verify correct code practices in code.

1. Install ESLint using: `*npm init @eslint/config`*
2. Configure ESLint according to your requirements
3. Run ESLint in your code directory using: `*npx eslint*`

**Setup swagger for your project**

Swagger is used to document your api. One can read about all the different requests and responses, try your api and understand the working without reading any code. We will use a yaml file to create the documentation and css to style it.

Install swagger-ui-express, swagger-jsdoc and yamljs using:

*`npm i swagger-ui-express swagger-jsdoc yamljs`*

Create swagger.yaml and swagger.css files in your project folder.

You can get swagger.css from here: <https://github.com/necolas/normalize.css/blob/master/normalize.css>

Example swagger.yaml file: <https://github.com/swagger-api/swagger-samples/blob/master/java/inflector-dropwizard/src/main/swagger/swagger.yaml>