CSE2252/CSE2242 (Software Development Lab II)

(Mock Exam)

Prerequisite: Make sure you have installed following software on you machine

1. MongoDB Community Edition (configure if needed)
2. Node.js (check version)
3. npm (check version)
4. Postman (for testing rest api)

**Optional Tools**

1. Express
2. mongoose (elegant mongodb object modeling for node.js)
3. dotenv (pull in environment variables from a .env file)
4. nodemon (automatically restarting the node application when file changes)

Create a Node.js web application “campus” where you will be building a Node.js RESTful API for creating, updating, retrieving or deleting users. **For these operations HTTP already has the adequate toolset: POST, PUT, GET, PATCH or DELETE.** Use mongoDB as a database to store all the data for users.

As a best practice, your **API routes should always use nouns as resource identifiers**. Speaking of the user's resources, the routing can look like this:

* POST campus/user or PUT campus/user:/id to create a new user,
* GET campus/user to retrieve a list of users,
* GET campus/user/:id to retrieve a user,
* PATCH campus/user/:id to modify an existing user record,
* DELETE campus/user/:id to remove a user.

**Use HTTP Status Codes Correctly**

If something goes wrong while serving a request, you must set the correct status code for that in the response:

* 2xx, if everything was okay,
* 3xx, if the resource was moved,
* 4xx, if the request cannot be fulfilled because of a client error *(like requesting a resource that does not exist)*,
* 5xx, if something went wrong on the API side *(like an exception happened)*.

**Test your API**

Test your API endpoint using Postman or curl and make sure each and every method (POST, PUT, GET, PATCH or DELETE) are working properly.

**Create a Proper API Documentation**

You write APIs so others can use them, benefit from them. Providing an API documentation for your Node.js REST APIs are crucial.

The following open-source projects can help you with creating documentation for your APIs:

* API Blueprint https://apiblueprint.org/
* [Swagger](http://swagger.io/) http://swagger.io/

Alternatively, if you want to use a hosted product, you can go for [Apiary](https://apiary.io/) https://apiary.io/

**Amazing REST APIs for Inspiration**

Here are four real-life examples that are worth checking out:

* [GitHub API](https://developer.github.com/v3/) https://developer.github.com/v3/
* [Twilio API](https://www.twilio.com/docs/api/rest) https://www.twilio.com/docs/api/rest
* [Stripe API](https://stripe.com/docs/api) https://stripe.com/docs/api
* [DigitalOcean API](https://developers.digitalocean.com/documentation/v2/#introduction) https://developers.digitalocean.com/documentation/v2/#introduction