Team Activity 1 Layout

Udemy layout:

* Create a brand new project(folder)

1. Write node JS code to spin up a server on port 3000
2. Handle 2 Routes: “/” and “/users”
   1. Return some greeting text on “/”
   2. Return a list of dummy users (e.g. <ul><li>User 1</li></ul>)
3. Add a form with a “username: <input> to the “/” page and submit a POST request to “/create-user” upon a button click.
4. Add the “/create-user” route and parse the incoming data (i.e. the username) and simply log it to the console.

My layout

For the team activity this week, you will be using what you learned from the Udemy course to create a Node.JS server that will expose several web endpoints in your local environment (i.e., accessed through localhost:3000 and not on the web).

While not required, the following organization will make your life a lot easier:

* (optional) Create a new folder in your class GitHub repository named “teamActivities” or another name of your choosing. This is a good way to organize your work and have the code available to others on your team and serve as a possible backup.
* (optional) Within the “teamActivities” folder, create a new folder named “ta01” or something similar.
* (optional) Within the ta01 folder, create an app.js file. This is the file you will run locally with the “node” or “nodemon” commands.

Requirements:

Once you have your environment set up, work with your team to spin up a local server that will:

* Listen on port 3000.
* Have 3 web endpoints:
  + “/” – Create an html page with a <title> and <body> content of your choosing.
  + “/activities” – Creates an html page which contains:
    - A list of 3+ activity names (“soccer”, “basketball”, etc) specified in hardcoded JavaScript array,
    - a <form> containing a single text <input> and submit <button> which when clicked submits a post request to “/add-activity”
  + “/add-activity” – Handle the body of the POST request and console.log() the text input from the form in “/activities”, and redirect to “/activities”.

If your team ever gets stuck, it is advised that you review this week’s reading and take to Google and StackOverflow to research solutions to any errors.

Hints:

* Use the back ticks (``) for templating. (e.g. `Hello, ${myVariableName}`)
* So far, html page creation is done through the .write() method in the response object.
* Make sure to return properly in your request handler to avoid setting the header twice.
* The req.on() event listener is good for handling POST request data: <https://stackoverflow.com/questions/12892717/in-node-js-request-on-what-is-it-this-on/12900268> [in-depth explanation]

Once you have finished the core challenges above, you’ll be qualified for a 93% grade on the team activity. It’s encouraged use any extra time to experiment with your work and research/test features not covered so far. In order to get 100%, it’s encouraged to complete at least one of these stretch challenges:

* Create and link an external CSS page for styling one or more of your pages. (link: https://stackoverflow.com/questions/44943676/link-css-to-html-for-nodejs-server)
* Write your form’s text input to a file using the ‘fs’ package.
* Read two number inputs from a form and outputs the sum to the console.