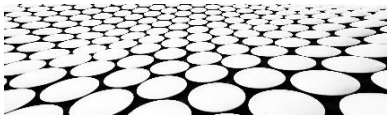
SEG3125 User Interface Design and Analysis



MODULE 6 – TUTORIAL/LAB

Compilation of survey results

GOALS

In laboratory 1, you did a small survey related to the evaluation of the user interface of a website of your choice. It was only the client part. In order to understand client / server interactions, this laboratory asks you to develop an application in Node.js to compile the results of your survey.

In this laboratory, you will:

- Learn a JavaScript framework, Node.js, which allows you to develop the server side of an application, in JavaScript.

We are, of course, in a course on UI design, and therefore, we could bypass the entire server part. But… I think it is important to understand the client/server chain, and also to discover Node.js, because several new client-side JavaScript frameworks (such as React which we will soon discover) are often used in conjunction with Node.js on the server side to develop complete applications.

PLEASE NOTE: Once again, there are 2 levels of requirements. Level 2 requirements are always optional but can stimulate you to learn more.

SUBMISSION DEADLINE

- Tuesday, June 30th, 11:30pm

SUBMISSION METHOD

- In Brightspace, the Module 6 checklist contains a link for your submission.
- As this is a web application running on your local server (localhost), we cannot use GitHub to view the rendering of your pages, as in previous laboratories. However, we can continue to use GitHub to keep the source files for your application.
- Do not submit files. Submit two links:
    - A link to your github folder containing the files for your application. Do not include dependencies. Instead, use a package.json file in your application to contain the dependencies.
    - A link to a video (use a shared link in your Google Drive) which shows:
        - The compilation of the survey
        - The form to fill out
- If you are submitting before the deadline, be sure to enter FINAL SUBMISSION when you are ready to have your submission evaluated.

Here is a link to a video of my project which will give some explanations. This is the kind of video you need to provide for your submission. If the video takes too long to start, you can download it first.

ATTENTION: Any code or even "small piece of code" that you take from a website such as stack overflow or other should be accompanied by a comment that recognizes the source. In your submission text, you must indicate "Code for X inspired by ……. (html link) ".

INSTRUCTIONS / TUTORIALS

Coding environment

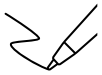
Before even getting started with Node.js. Are you using a good editor for your code? If yes, perfect. But if not, it is time to download a good code editor. I recommend Visual Studio Code, that I really like. There are others (such as the Atom used by the author of the videos which I will tell you about below), but I find Visual Studio Code to be easy to learn.

Node.js

First go to the site https://nodejs.org/en/ and download Node. During the installation, keep the "default settings" that you are offered. No need to download dependencies, we don't need them.

Learning Node.js is not trivial and will take a little time. I point you towards several videos of "The Net Ninja" (yes yes!) which will help you learn Node.js. Each video is very short and well done. These videos are really great, and the author explains well. I highly recommend them (the whole series!). Obviously, as they date from 2016, there will be a few outdated aspects, but still, they give good explanations on Node.js.

1. What is Node.js ? Node JS Tutorial for Beginners #1 – Introduction
2. How to call Node from the command prompt. Node JS Tutorial for Beginners #2 - Installing Node JS. You can create a terminal in Visual Studio Code (just like in Atom as the author does).
3. All about modules which are useful to make code modular. Node JS Tutorial for Beginners #6 – Modules and require()
4. Reading/writing from/to files. Node JS Tutorial for Beginners #9 – Reading & Writing Files (fs)
5. The Node Package Manager (npm) which is very useful. Node JS Tutorial for Beginners #20 – The Node Package Manager. npm will install modules, such as Express, which will help with client/server exchanges.
6. The file package.json explained, which is for keeping track of dependencies Node JS Tutorial for Beginners #21 – The package.json File
7. The Express module which will facilitate routing and client/server exchanges. Be careful, from this video the author uses the command "nodemon" instead of "node" to launch his application, but you can continue to use "node".
    a. Node JS Tutorial for Beginners #23 – Introduction to Express
    b. Node JS Tutorial for Beginners #24 – Express Route Params
8. The Embedded JavaScript templating (EJS) module to display HTML pages with dynamic content Node JS Tutorial for Beginners #25 – Template Engines
9. POST request – Node JS Tutorial for Beginners #30 – Handling POST Requests



DESIGN

In this laboratory, we make a small incursion on the server side to better understand client/server communication. So, we won't do a lot of UI design. However, as you will see in the starting code, I wanted to separate the view from the "survey participant" and the view from the "analyst". These two users of the same application have completely different needs.

The only requirement in terms of UI is to minimally use Bootstrap to make your survey more attractive on the "participant" side. I'm not asking you to improve the "analyst" side because it is HTML with dynamic content, and it is more complicated. I prefer that you focus on the participant's view.

STARTING POINT

Before looking at the base code that I provide for this Lab 6, I recommend that you listen to 4 other Net Ninja videos. Those will make it easier for you to understand the structure I used, and what to install to run my application. I purposely followed the structure defined in the Net Ninja videos which shows how to make a To-do list application. I use the same packages (express, ejs, body-parser). So it would be a great practice for you to follow the tutorials for the TO DO list, and then you will be better prepared to understand and adapt my code.

- Project structure: Node JS Tutorial for Beginners #31 – Making a To-do App (part 1)
- Explaining the controller: Node JS Tutorial for Beginners #32 – Making a To-do App (part 2)
- Using EJS: Node JS Tutorial for Beginners #33 – Making a To-do App (part 3)
- Dealing with POST request with jQuery (no need to listen to the delete request) Node JS Tutorial for Beginners #34 – Making a To-do App (part 4)

I provide a repertory with Node.js code to compile survey results. Here is the structure :

- app.js (entry point)
- surveyController.js (controller in an MVC pattern)
- data (data repertory to contain the statistics on questions, on the server side)
    - fruit.json
    - animal.json
    - color.json
- public/assets (in order to be compatible with the instructions in the video, I used this same folder)
- action.js (jQuery code which will do the POST when the survey is submitted)
- views
    - niceSurvey.html (survey, static view for the participant)
    - showResults.ejs (results, dynamic view for the analysis)
- package.json (package dependency manager)

CODING

Level 1 requirements (Mandatory)

Data

Normally, the data would be in a server-side database, but in this lab, we're not going to talk about server/database links. We will limit ourselves to the server reading/writing files (json files).

1. Create the data file(s) corresponding to your survey questions.

I created 3 data files in the base code. These files are fruit.json, animal.json and color.json. These are files that contain the accumulated survey results. You can keep the information in a single file. I just wanted to show you that it is possible to access several files if desired which could be useful for a good organization of the data in a large project.

Client side / View of the participant

This view would be accessible (in my code) at http://localhost:3000/niceSurvey. You can use the route you want.

2. Improve the design of your survey
    a. Now that you know Bootstrap, you can use Bootstrap to enhance the UI of your Lab 1 survey.
3. Make sure the structure of your survey is adequate
    b. You should have 6 questions of various types. In my starting code, I show you 3 types (text, list, checkbox).
    c. You had to use the <form> tag in lab 1, but perhaps without really knowing why. This tag takes on its meaning when the form is being submitted to the server, which can analyze it. The "submit" button is used to POST the survey, meaning to send the information.
4. Make sure the survey is visible (available at http://localhost:3000/niceSurvey). It is the surveyController.js which takes care of putting the form online by responding to a GET request.

Transfer and saving of the information (server side)

5. Make sure that the form is correctly sent to the server
    a. The action.js file of the base code takes care of doing the POST.
    b. The surveyController.js file takes care of recovering the POST data
6. Update the data files according to the responses to the questionnaire
    c. Modify the surveyController code according to your data

Client side / Analysis view

This view would be available (in my code) at http://localhost:3000/analysis. You can use the route you want.

7. Show the results in a different view of the survey itself
    a. Adapt the start code to show the content of your data files

Level 2 Requirements (Optional)

You already know Node.js? You can add:

- Make the visualization of the results much prettier with graphics (pie-chart or other)
- Modify the survey to make it "multi-form", the user answering only 1 or 2 questions per page.
- Put a database in the back-end rather than files.

☆☆☆ EVALUATION

- This laboratory is worth 3.5%.
- Any student who has met the 7 requirements (points 1 to 7 above) will be granted 10/10. Each missing requirement will be penalized by one point.
- Failure to provide sources of code inspired online will result in a score of zero.
- Any delay beyond the deadline will have a penalty of 10% per day.

QUESTIONS

- You can ask your questions in the Module 6 discussion forum on Brightspace.
- There is a consultation schedule for the laboratories in the organization section of the course on Brightspace. That will tell you when a teaching assistant is available on the forum or by zoom.
- You can also send your questions directly to the TA you are assigned to. Refer to the lab consultation schedule to see which TA you are assigned to.