Student-Initiated Courses

http://sticsumd.com
CMSC 122
Intro to Programming via the Web

Welcome

Welcome to CMSC 122. This course provides an introduction to the internet/web capabilities and trends, and to computer programming in the context of building simple web pages. Intended for students with no previous programming experience who wish to understand the technologies making web sites possible, this course will provide a set of practical problem solving skills necessary for the development of dynamic client-side web content. This class provides non-majors with a basic skill set for leveraging web technologies and limits of such resources.

Topical Content

- History of the Web/Internet
- Internet and Society
- Web/Internet Fundamentals
- HTML/CSS for dynamic web sites
- Web authoring tools
- Design (pseudocode)
- Input/Output/Expressions in JavaScript
- Conditional/Iteration Statements in JavaScript
- Testing/Debugging
- Aggregate types in JavaScript
- Web Page Evaluation
- Research-Quality Web Searching
- Basics of usability and art theory in page design
- Basics of e-commerce and tool integration in the web

Instructor

Fawzi Eranki

Recommended Textbooks

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>ISBN</th>
</tr>
</thead>
</table>
Pilot
House of Representatives

Out: 3/17/17 | Due: 3/31/17 11:59 PM (Deadline 4/1/17 11:59 PM for 10% reduction)

Overview

You have a data.json file that contains much information about most of the US House of Representatives (we removed some representatives from places like Guam, Puerto Rico, etc.).

In this project, you will create a website that allows you to visually explore all of the 50 states and their representatives.

Objectives

This project will give you practice with using handlers as a templating engine. In addition, it will be a refresher for your HTML and CSS understanding.

Grading

Grading is done entirely through visual inspection. The point breakdown of each template is described below. In total, there are 24 points to be earned, with varying numbers coming from each handlers file.

Setup

Clone this project folder and run npm install inside the directory.

You have the following files available to you:

- data.js: this is where you will define all your routes. We have provided a skeleton of a working server that has Express configured and initialized, middleware initialized, and the server listening on port 3000. We have also defined all the routes.
- data.json: this is a json file containing all the representative data you will need. This file should never be modified.
- app.js: these are utility functions available for you. Read them carefully to understand what they are doing. Feel free to modify this file if you want to add your own functions, as you will be turning in the entire project directory.
- views: this is the default template file. You must add a navigation bar to it. More instructions will be provided below and in the file.
- index.js: this is the template file that supports the /index endpoint.
- person.js: this is the template file that will be used for the /person/endpoint.
- states.js: this is the template file that supports the /state/endpoint.
- representatives.js: this is the template file that supports the /representatives/endpoint.

Week 5: Express.js

Why do we need Express.js?

Node.js is a runtime environment that allows us to create applications in JavaScript.

Last week, we explored how to use Node to:

1. Read and write files
2. Make requests
3. Write a http server.

The server we wrote looked like this:

```javascript
var http = require('http');

// Express.js

function handleRequests(req, res) {
  if (req.url === '/') {
    res.write('Hello world!
    return res.end();
  }

  var request = request(req, res);
}

var server = http.createServer(handleRequests);
server.listen(3000, function() {
  console.log('Server listening on http://localhost:3000...'});
```

With this application, we can run a http server (on port 3000) that is capable of sending plain text responses. In addition, it can send different responses based on which route (such as /index) is visited.

However, some of the more advanced features of a web server are not available to us by Node directly. For example, supporting different HTTP verbs (GET, POST, etc.), serving static files, and dynamically creating templates.

Instead of writing code to do all of this from scratch (which would be very hard and tedious), we can use a web framework that has all the work done for us.

By far, the most popular Node.js web framework is Express.js. Other Node frameworks include Hapi.js, Sails.js, and Koa.js.

**Why are we using Express.js and not the others?**

It is the most popular, backed by major companies, and has the most support and active development. It is also very flexible and allows us to mix and match components as we wish.

What can Express.js do?
“I honestly think this class has been the most relevant to the career path I anticipate going into. This class is already on my resume, as I am applying primarily to web development and design positions. I’d love to apply the skills I learned in this class in future projects; in fact, I already did a hackathon project using Node.js since starting this class. I’m very excited to learn more Node.js techniques and further developing skills in this field.”

“This has been one of the most useful classes I have taken at the university.”

“I thought that this class was an amazing experience. The workshop model lends itself to asking a lot of great questions, and I learned a lot from the TAs’ answers and other students’ questions. It was a great way to get hands-on experience as well, which is essential when it comes to topics such as web development.”

“I have always been a hands on learner and actually implementing the material I was learning in class made a huge difference in my understanding of the material. In traditional CS classes, there is usually an instructor who teaches us concepts but we don't implement them until we start the projects. The workshop style teaching worked really well for me and I absorbed a lot. There was also always help available to me through office hours and during class.”
Ishaan! I interviewed with Target just now for next summer.

They asked me to explain something similar to our final project, and I explained what we did in class.

He really liked it.

YESSS
This is the point
That's so awesome
Ishaan! I interviewed with Target just now for next summer.

They asked me to explain something similar to our final project and I explained what we did in class.

He really liked it.

He gave me an offer for next summer. Haha.
Full-Stack Web Development with Node.js
Practical Cloud Computing with AWS
The Coding Interview
Full-Stack Web Development with Node.js
Practical Cloud Computing with AWS
The Coding Interview*
Ethical Hacking
The Intersection of Economy and the Economy
Unity + Game Development*
Digital Logic Design through Minecraft
Simultaneous Localization and Mapping
If you have any questions, need advice, or would like to learn more, please do not hesitate to reach out to us:

Ishaan Parikh: parikh.i.m@gmail.com

STICs: sticsumd@gmail.com

Website: sticsumd.com