IT338 - Web Development - Week 1

Spring 2019 - J. deBettencourt PhD

Discussion Notes

1. Course extends understanding of Web Applications starting from Full-Stack Server-Side Rendered Sites, progressing to various Client-Side Rendered JAM-stack (JAM: Javascript, APIs, Markup) applications using XHR/Ajax, REST, and GraphQL backend services, and culminating in the use of stored client-side serverless Progressive Web Applications.
2. The prototypical Full-Stack application discussed is a Node.js, Express.JS, and MongoDB (or other database) type application, with a UI built with HTML5, CSS3, and JS/ES. This is also a server-side configuration (basically less any UI component) that can provide XHR, REST, and even support GraphQL services.
3. Development and testing of class-related applications will involve use of standard Javascript, Javascript libraries, and Javascript application frameworks (e.g. React), and requiring familiarity with HTML5 and CSS3 (and related responsive design styling frameworks).
4. Class activities will require use of various toolchains, generally requiring a local and/or online IDE (e.g. local: Visual Studio Code; online: Codeanywhere.com), Github support, Node.js integration for development and testing; and access to production server platforms (e.g. Heroku, AWS, Google, Surge, Netlify, etc.).
5. Various local tools and sandbox environments can also be used productively in this course; these include your own code-runners to test JavaScript and HTML/CSS snippets, and online tools like JSFiddle (<https://jsfiddle.net/>), CodePen (<https://codepen.io/>) and others (Google: ‘jsfiddle alternatives’ for more options).
6. Use of Chrome, Firefox, or Edge developer tools is required, especially in regards to PWA (Progressive Web Applications).
7. In addition to React CLI, we will use the GatsbyJS CLI to create example Web Applications/PWAs; GatsbyJS also requires understanding of GraphQL and the use of Gatsby plug-ins for data access and acquisition (similar to REST APIs).
8. All class activities with a development or implementation component must be supported with a public Github repository managed by the student.
9. Expect to encounter coding errors, typos, misconfigurations and sometimes package compatibility problems within assignments; resolving these types of problems and breaking changes is part of the course experience.
10. A local web server option for local testing is the Web Server for Chrome extension. The Github source is at: <https://github.com/kzahel/web-server-chrome> and the webstore link: <https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhemlocgigb?hl=en>

IT338 - Web Development - Week 1

Assignments:

1. Development toolchains and services; personal development workflow

* Choose your own set of tools, systems
* Document installation and configuration; maybe try a Gist

1. JavaScript ‘script runners’ and developer ‘sandboxes’

* Test the simple script runners presented to class
* Test one or more of the online JS tools (JSfiddle, Codepen, etc.)
* Document use and example programs (Word.doc & screenshots)

1. JavaScript reintroduction: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript>

* Go through reintroduction tutorial to understand new JS capabilities
* Get the Fetch Web API working; Information in MDN Client-Side web APIs: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs>
* Document what it takes

1. Github Account; Github pages web-site; Github pages repository

* Create (Use) Github Account
* Create Github pages web-site
* Use a repository workflow approach to updating/modifying
* Document your workflow (continue to use throughout the quarter)

1. Localhost Web Server instance(s)

* Install one or more local web-servers capable of serving static pages
* One of easiest is Web Server for Chrome
* Could also have local installs of Apache or Nginx, for example
* Might also have a server instance installed through an IDE
* Document installation, configuration, and use

IT338 - Web Development - Week 1

References

1. Learn Web Development - Mozilla Developer Network : <https://developer.mozilla.org/en-US/docs/Learn>
2. Node.js Simple Server - Getting started - Nodejs.org : <https://nodejs.org/en/docs/guides/getting-started-guide/>
3. Node.js Documentation - Nodejs.org : <https://nodejs.org/en/docs/>
4. W3Schools.com Node.js Tutorials : <https://www.w3schools.com/nodejs/default.asp> (W3Schools Tutorials are sometimes more elementary than similar tutorials at MDN)
5. Github Pages: <https://pages.github.com/>
6. Current EcmaScript Standard: <https://tc39.github.io/ecma262/>
7. Platform (Browser) Compatibility: <https://caniuse.com/> and <https://kangax.github.io>
8. Web Standards Groups/Resources: Web Hypertext Application Technology Working Group (WHATWG): <https://whatwg.org/>

HTML Living Standard: <https://html.spec.whatwg.org/multipage/>

Some Additional Tech/Tech Tracking/Trending/Review/Discussion Sites :

**BuiltWith:** [**https://trends.builtwith.com**](https://trends.builtwith.com)

**W3Techs:** [**https://w3techs.com/**](https://w3techs.com/)

**Google Trends:** [**https://trends.google.com/trends/?geo=US**](https://trends.google.com/trends/?geo=US)

**The State Of JS:** [**https://stateofjs.com/**](https://stateofjs.com/)

**Medium:** [**https://medium.com/**](https://medium.com/)

**Google Developers:** [**https://developers.google.com/**](https://developers.google.com/)

**Amazon Web Services:** [**https://aws.amazon.com**](https://aws.amazon.com)

**Codeanywhere:** [**https://codeanywhere.com**](https://codeanywhere.com)

**Github:** [**https://github.com**](https://github.com)

**Visual Studio Code:** [**https://code.visualstudio.com/**](https://code.visualstudio.com/)

**Jamstack:** [**https://jamstack.org/**](https://jamstack.org/)

**Gatsby:** [**https://www.gatsbyjs.org/**](https://www.gatsbyjs.org/)

**GraphQL:** [**https://graphql.org/**](https://graphql.org/)

**Nginx:** [**http://nginx.org/en/**](http://nginx.org/en/)

**React:** [**https://reactjs.org/**](https://reactjs.org/)

**Progressive Web Apps:** [**https://developer.mozilla.org/en-US/docs/Web/Progressive\_web\_apps**](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps)

**Node/ExpressJS:** [**https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs**](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs)

**NodeJS(without a framework example):** [**https://developer.mozilla.org/en-US/docs/Learn/Server-side/Node\_server\_without\_framework**](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Node_server_without_framework)