



Portfolio_1

EPLORATION OF NODE.JS

Hosam Abdeltawab | Computer Science 319 | February 11, 2017



Table of Contents.

Introduction	Page 2
What is Node.js?	2.0
What can be built using Node.js?	2.1
Experience	2.2
Overview	Page 3
New Things Learned	3.0
Complexities Faced	3.1
Blooms Taxonomy	Page 4,5
Creation	4.0
Analysis	4.1
Evaluation	5.0
Denouement	Page 6
Conclusion	6.0
Works Cited	Page 7



Introduction.

WHAT IS NODE.JS – 2.0.

Node.js (Node) is an open source, cross-platform runtime environment for developing server-side and networking applications that are written in JavaScript. Node uses an event-driven non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices

WHAT CAN BE BUILT USING NODE.JS – 2.1.

Node can be programmed in various ways to satisfy any need of the project. In addition, some of the projects that can be built using Nodes are:

- A chat room.
- Remote control for a car.
- Drawing tools on a website.
- Cross-platform applications.

EXPERIENCE – 2.2.

In the first time working with Node.js, I have managed to learn new and interesting concepts on how to better implement, organize, and break down big problems into small parts.

The concepts that I have learned in doing this project were: object-references, data-query, prototypes, object-factory, modules, shared-modules, core-modules, EJS (Embedded JavaScript), and JSON (JavaScript Object Notation) file format.

An EJS is a JavaScript template Library that is used to build HTML string from data received from the JSON files, while the JSON is used to transmit data between a server and a web application, it is known as a format for structuring data.

In the upcoming sections a detailed explanation is provided about the process of what was the process of making this project and how it was a beneficial experience for me as a programmer to encounter such an extraordinary language. In addition, an explanation about what obstacles were faced in building this project, and how these obstacles were conquered.



Overview.

NEW THINGS LEARNED – 3.0.

Before starting with this project, I had to read about Node.js due to the fact that I did not have any previous experience with the language. First, I started reading on what Node was, and what is the best environment (compiler) that it can be compiled on. Second, I looked at online tutorials on how to best start learning how to code Node. Finally, I implemented every concept I learned in a separate project file to help me remember a specific concept whenever needed to.

When I started reading, I first went on Node's website (www.nodejs.org) to read on how to start working with Node. It turned out that Node is very similar to JavaScript; however, Node has new and interesting concepts that are advanced than JavaScript. Some new concepts I learned were: object-referencing, data-querying, and modeling functions.

One of the best intriguing concepts that I learned was “Core-Modules.” Core-modules are similar to Java collections, it has predefined functions that are ready to use such as: “connect” and “express”; core-modules can be used to set up servers and to “inherit” functions from other files.

Another interesting application was structuring the data in a JSON in which I learned how that by using this type of formatting I can send data over to be used on the server-side file (HTML).

COMPLEXITIES FACED – 3.1.

In doing this project, I have faced multiple problems. A problem I faced was with Modules, a module is a reusable piece of information such as a function or a variable, I couldn't understand how to make a module and pass it to another file.

Another issue I faced was with the EJS (Embedded JavaScript) object referencing, in learning how to code EJS the problem I faced was with the local variables, how to set and call them in other files. The third problem was with the JSON (JavaScript Object Notation), faced a problem with transmitting data over to other files.

Finally, I faced a dilemma with the JSON file formatting, I couldn't understand how did the structure of the file worked together which resulted in more research and tests that were concluded by using “Pretty Print” to inspect JSON and using “Linearize” when writing JSON;



Blooms Taxonomy.

CREATION – 4.0.

When I started to learn Node.js, all I thought about was “what is Node.js” and “what are the concepts and advantages that I can receive from it.” The idea of the project is not an application that has multiple lines of code and accomplishes a specific task. The Idea behind my project is to teach.

When asked how does this project succeeds in teaching? I say that idea in mind was to break up the most known (defined) notions within Node.js and to break them up into smaller different projects with each project implements one or more notions in a simple and a direct way that could be easily comprehended by any reader.

ANALYSIS – 4.1.

The project is divided between two folders “NodeJS” and “express”. In the NodeJS folder there are ten subfolders “Applicaon”, “Connect_Server_Framework”, “coreModules”, “Creating_an_HTTP_Server”, “dataQuerring”, “Modules”, “objectFactory”, “objectReferences”, “prototyping”, “sharedModules.” Each one of the projects (subfolders) explains a concept about Node.js.


The “Application” project is the introduction to Node.js folder, it explains the notion of setting variables, setting functions, and printing out values to the console.

The “Modules” project defines what is a module and explains how to use modules to send value to a different file.

The “objectReferences” project shows the notion of referencing and comparing objects together and what types of errors a user can get if he/she did not reference an object correctly.

The “Modules” project show the notion of Modules. What are modules, and how to send values from a different file using the method exports().

The “objectFactory” project shows the notion of implementing modules with one function and two different values with objects. How a user can send and receive a value from a different file by using the method require().



The “dataQuerring” project shows the concept of nested functions by simulating a restaurant food ordering system. How to call functions within a function and hoe to use the setTimeout() method.

The “sharedModules” project shows the notion of how to use Modules together. How do modules link together?

The “coreModules” project shows the notion of how the predefined FileSystem and Path modules work. How to call them and how to implement some functions of both modules.

The “protoyping” project shows the notion of how to add methods or properties to the defined objects in the file.

The “Creating_an_HTTP_Server” project shows the notion of creating a server, receiving requests, sending responses, and handling I/O by using the core-module “http.”

The “Connect_Server_Framework” project shows the notion of how to use the connect core-module when a client is connected to the server.

The “express” project shows how the express core-module works to build a web-page.

EVALUATION – 5.0.

As a software engineering major, I am constantly learning new languages and new concepts. The functionality of Node.js is amazing, when building a server the developer can use many functionalities from modules to core-modules.

The web is a changing technology landscape. When choosing the perfect language the developer can be bewildered by all the options available. In exploring Node.js, I came across other scripting languages such as PHP an Angular.js. Angular is an open source framework to create powerful single page applications, while PHP is a server-side scripting language. I can say with no doubt that Node is the right choice.

Node is very similar to JavaScript, when used correctly it can be a powerful tool in the hands of the programmer. The reason I choose Node is because Node’s simple structure and its easiness to learn. If the developers know JavaScript then they are good on Node.js.



Denouement.

CONCLUSION – 6.o

In a nutshell, this was one of the greatest experiences I had in programming. I learned most of Node's functions and concepts. I learned how to use modules and core-modules, and I learned how to write and use JSON files; in addition I learned how to EJS. Towards the end, I encourage every web developer to learn Node.js, it is a very useful language.



Works Cited

Foundation, Node.js. "About Node.js®." About / Node.js. N.p., n.d. Web. 12 Feb. 2017.

PHP 5 Tutorial." PHP 5 Tutorial. N.p., n.d. Web. 12 Feb. 2017.

Tutorialspoint.com. "Node.js Introduction." Www.tutorialspoint.com. N.p., n.d. Web. 12 Feb. 2017.

The Node Beginner Book." The Node Beginner Book » A Comprehensive Node.js Tutorial. N.p., n.d. Web. 12 Feb. 2017

7 Awesome Things You Can Build with Node.js." Treehouse Blog. N.p., 14 July 2015. Web. 12 Feb. 2017.