BSc in Software Development

Year 3

COMP07030 Software Design Project

  Online Learning System

G00253470

Vusumuzi Khumalo

Contents

[Introduction 1](#_Toc449451426)

[Architecture of the solution 2](#_Toc449451427)

[HTTP 3](#_Toc449451428)

[MongoDB Data Model 4](#_Toc449451429)

[Node.js 6](#_Toc449451430)

[HTML 8](#_Toc449451431)

[Express 8](#_Toc449451432)

[System Demonstration 8](#_Toc449451433)

Student Number:  G00253470

Student Name:  Vusumuzi Khumalo

Supervisor:  Brian McGinley

GitHub Link:  https://github.com/vusumuzi/FinalYearProject3-

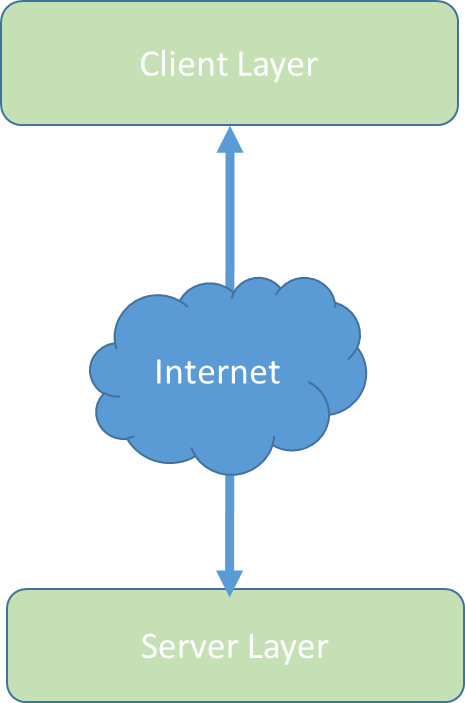
# Introduction

This project presents an online learning application. It has a learning page with classes and links to different classes. It shows class details. Registration is required to access the classes. When registered, details of classes and lessons will be shown when you log in. Lessons can be added or removed.

Technologies used includes Node.js which is suitable for developing server-side Web applications. Some of the modules that are included are: passport and bycrpt for logging in. Mongo and mongoose are used as the database. Express validator and express handlebars would be used as the template engine.

# 

# Architecture of the solution



Technologies used on the Client side include Node.js and has some of the following modules: JavaScript, HTML and CSS and Express. On the server side mongo dB[[1]](#footnote-1) is used for storing data and mongoose for querying the database. The shell provides a convenient way of querying and updating data as well as performing administrative operations. Data in the mongo Db is stored as documents using JSON.

Class diagram and Data Model

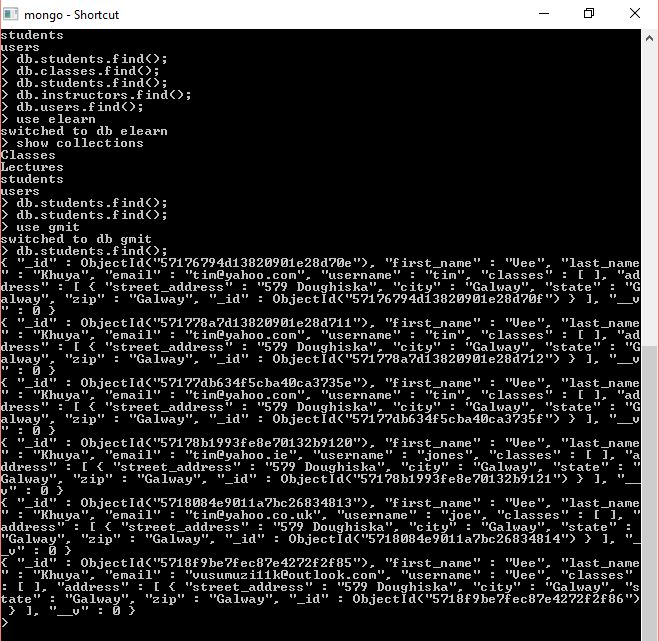
*This section should include a class diagram and data model as well as the reasons why they were designed as they were.*

## 

## MongoDB Data Model

Data as Documents

MongoDB stores data as documents in a binary representation called BSON (Binary JSON). Documents that share a similar structure are typically organized as collections. Think of collections as being analogous to a table in a relational database: documents are similar to rows, and fields are similar to columns.



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | First Name | Last Name | email | username | classes | Street address | city | state | zip |
|  | Brian | McGinley | brian@gmail.ie | brian@1 | Software  Development | Gmit  Dublin Road | Galway | Galway | Gal |
|  | Gerry | Agnew | gerry@gmit.ie | gerry@5 | Server Side  Rad | Church  Street Cork | Cork | Cork | Cok |
|  | James | Ngondo | jamie@yahoo.ie | jamie@90 | Business Studies | Station Road | Dublin | Dublin | Dub |

Fast, Iterative Development.

It is a flexible data model with dynamic schema and makes it fast for developers to build their applications.

Flexible Data Model.

MongoDB's document data model makes it easy to store and combine data of any structure. It is easy to modify the schema in a short space of time. It takes less time to prepare the schema and more time to insert data

Scalability. MongoDB can be scaled within and across geographically distributed data centres. MongoDB scales easily with no downtime, and without changing your application.

MongoDB documents tend to have all data for a given record in a single document.

Technologies used

## HTTP[[2]](#footnote-2)

HTTP is an abbreviation for Hyper-Text Transfer Protocol, which is the underlying protocol used by the World Wide Web. It outlines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

**Main Features of HTTP:**

Below are three main features of HTTP:

HTTP is connectionless, meaning that HTTP client which is a browser will send a request and after a request is made, the client will disconnect from the server and waits for a server response. After the server has processed the request it re-establishes the connection with the client to send a response back.

The other feature of HTTP is that it is media independent which means that any type of data can be sent using HTTP as long as both the client and the server know follow the right protocol of handling the data content. It is a requirement for both the client and server to specify the content type when sending data.

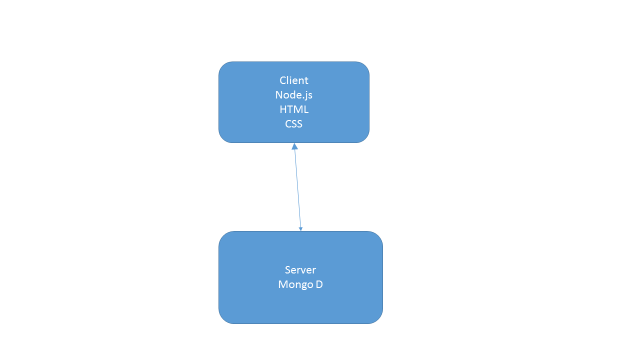
Finally, one of the main features of HTTP is that it is stateless. The server and client are aware of each other only during a current request. Afterward disconnection both of them forget about each other. This has the advantage of that neither the client nor the browser can retain information between different requests across the web pages.

**GET Method:**

A GET is the main method used to retrieve data from a web server by specifying parameters in the URL portion of the sent request.

**POST Method:**

To send data to the server, The POST method is used especially when you want to update the file.



## Node.js

 I installed Node.js[[3]](#footnote-3) so that l could use it to easily build fast and scalable network applications. Node.js 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.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications

Features of Node.js[[4]](#footnote-4)

The following are some of the important features that make Node.js the first choice of software architects.

**Asynchronous and Event Driven:**

All APIs of Node.js library are asynchronous that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

**Very Fast:**

Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

**Single Threaded but Highly Scalable:**

Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

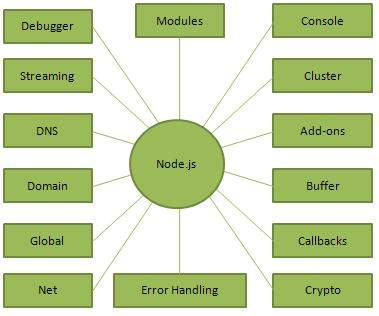
**No Buffering:**

Node.js applications never buffer any data. These applications simply output the data in chunks.

Many companies use Node.js and some examples includes eBay, General Electric, Microsoft, PayPal, Uber, Yahoo!, and Yammer to name a few.

**Concepts**

The diagram below[[5]](#footnote-5) shows some important parts of Node.js



## HTML[[6]](#footnote-6)

Hyper-Text-Mark-up Language, commonly abbreviated as HTML, is the standard mark-up language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically and, before the advent of Cascading Style Sheets (CSS), included cues for the presentation or appearance of the document (web page), making it a mark-up language, rather than a programming language.

## Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web applications. With a lot of HTTP utility methods and middleware available, creating a robust API is quick and easy. Express provides a thin layer of fundamental web application features, without obscuring Node.js features. Many popular frameworks are based on Express.

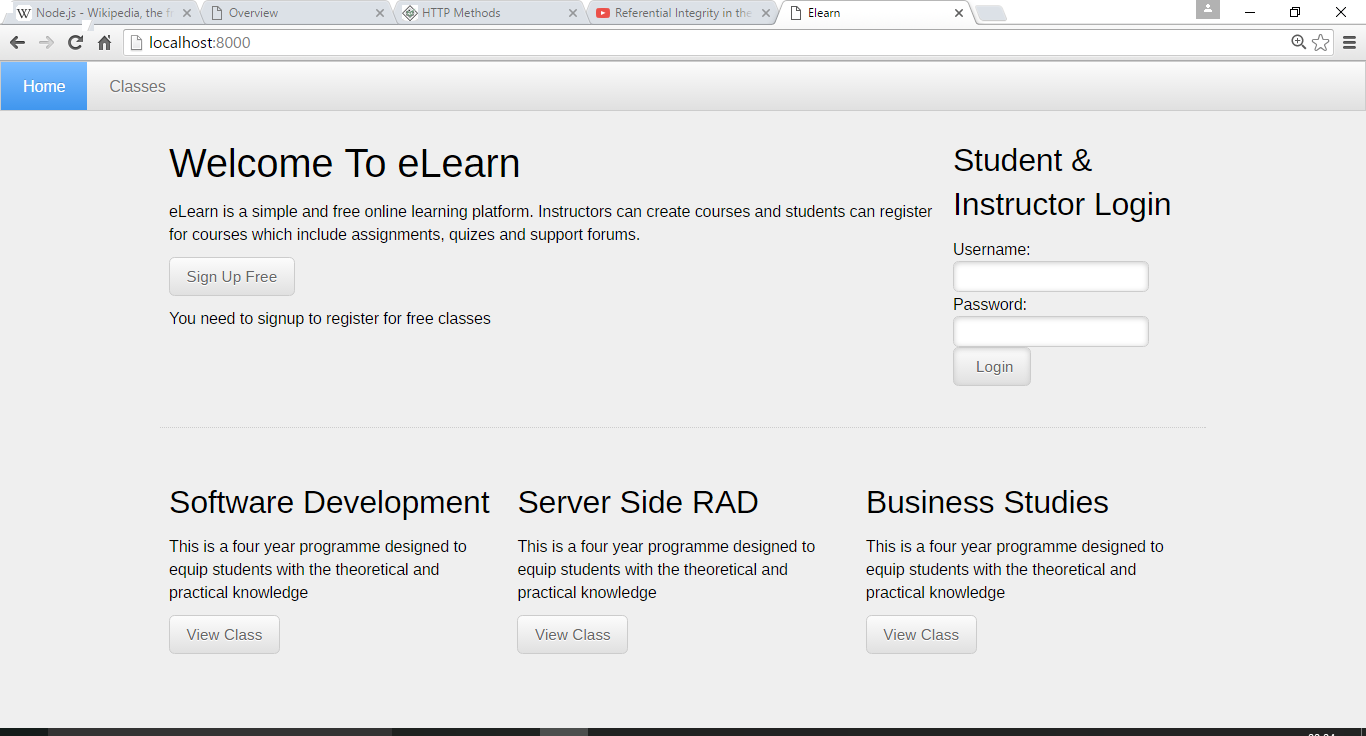
Express-validation is a middleware that validates the body, params, query, headers and cookies of a request and returns a response with errors; if any of the configured validation rules fail. Express-validation supports validating the following: body, params, query, headers and cookies.

# System Demonstration

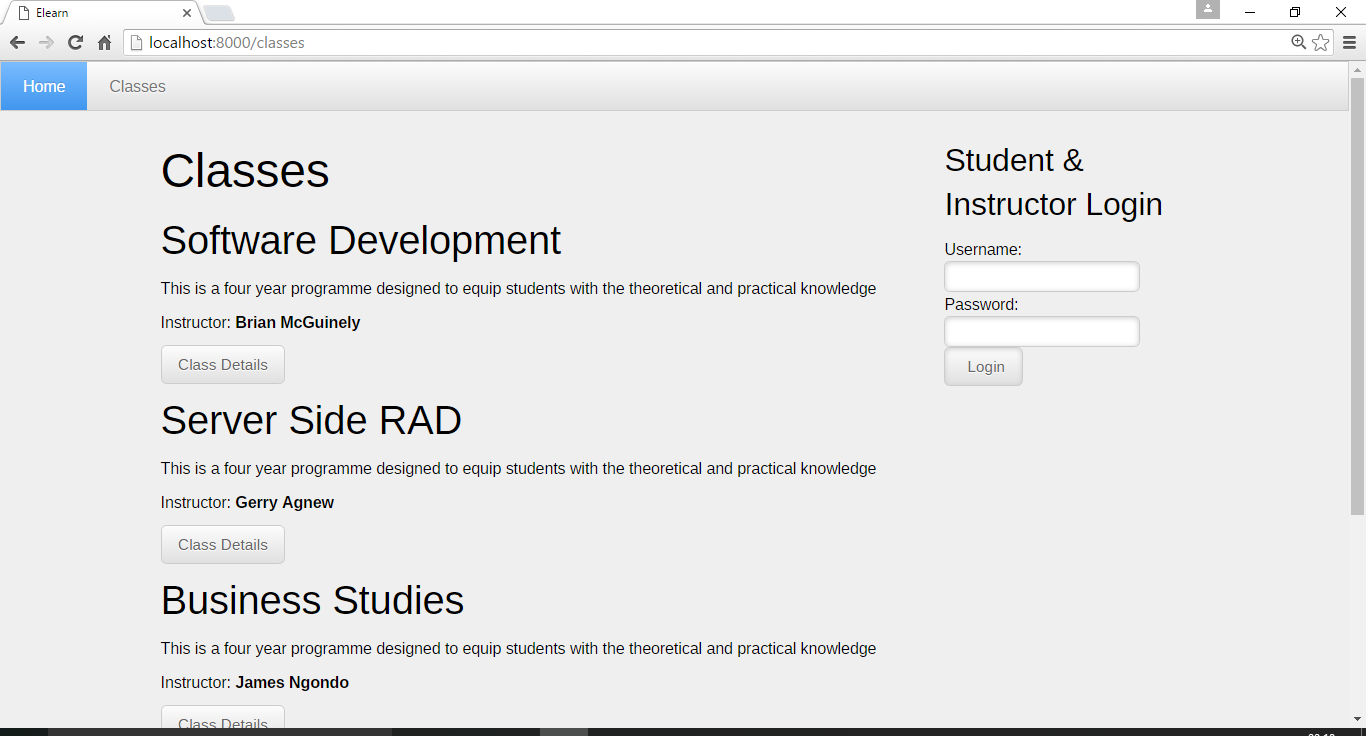
Screen Shot 1

Home page of an online Learning System

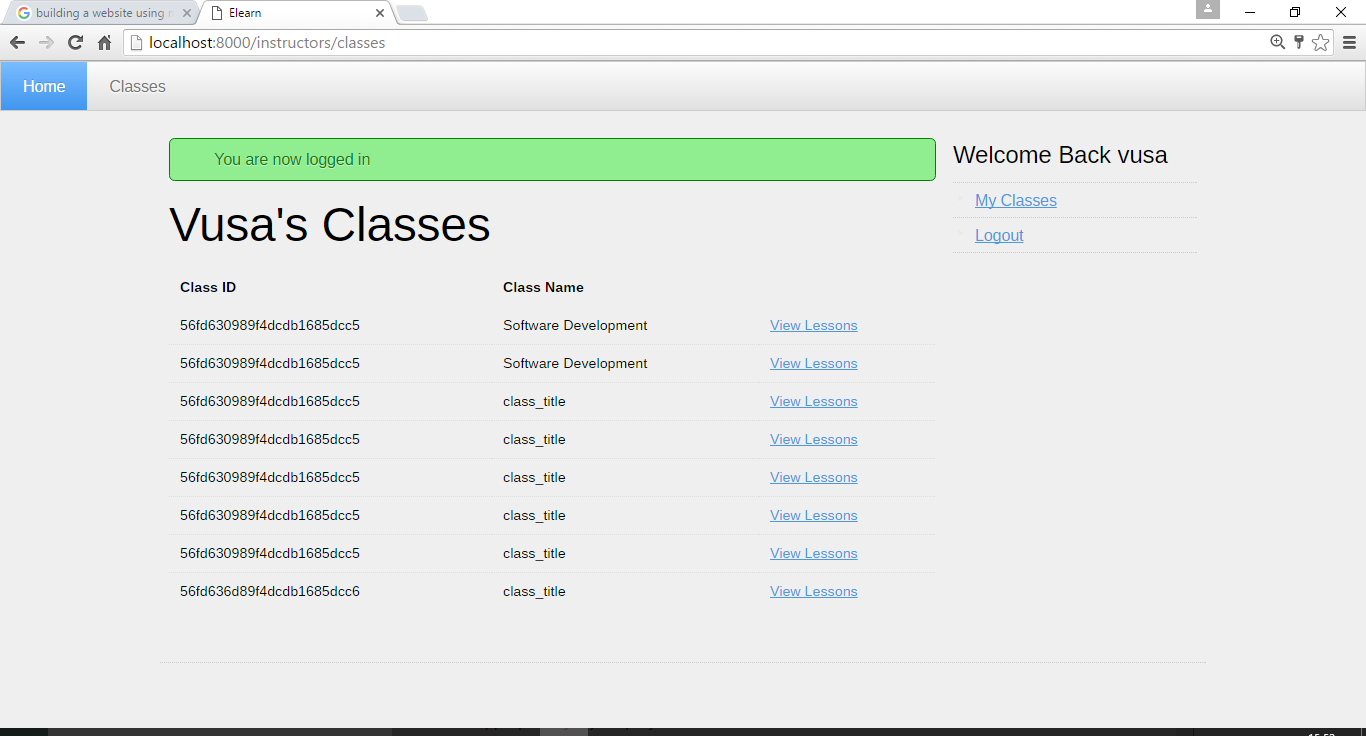
When you open up the browser, it opens up in the home page. In the home page there is a welcome section, Classes button link, sign up link and student and lecturer log in. You can view classes and class details.



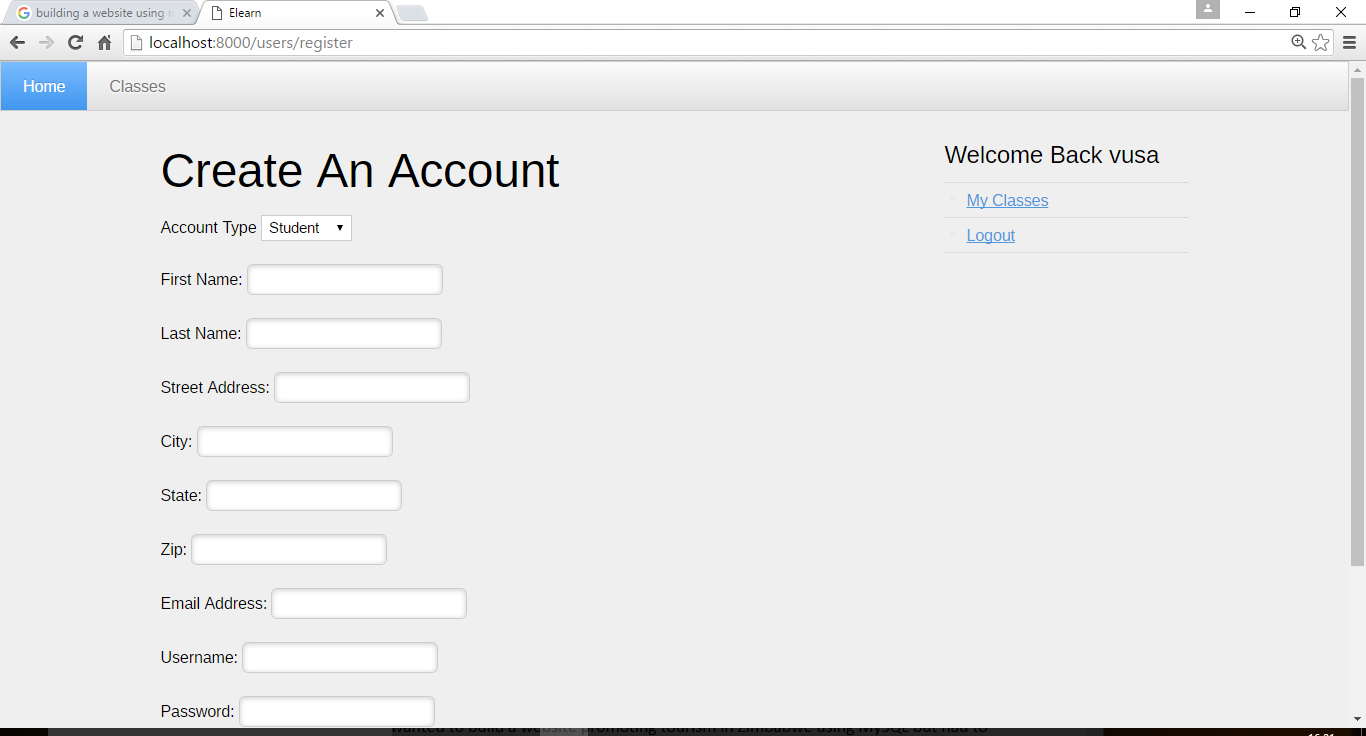
The screen short below is Classes Page. It shows the classes in each department and a brief description. It has a summary of the classes and the lecturers.



The screen shot shows the user logged in and the classes the user is enrolled in.



The screen shot below is link page where you can register to have access to the classes.



Problems Encountered/Solved

Initially l wanted to build a website promoting tourism in Zimbabwe using MySQL but had to change the idea and do an online learning system using mongo db. and Node.js. Setting up the database in MySQL gave me a lot of problems and I could not pursue the project due to the limited amount of time l had to do the project. After researching and studying about mongo db. I decided to go on and use it to do my project.

Most of the problems I had were very minor in that l was careless in writing the code by omitting commas and quotation marks. Researching and getting help online helped me in building my project as shown by the references I consulted. The Mongo Website proved to be the main site where I got most of the data about the layout of the database.

Conclusions

The online learning system was built using Node.js and Mongo db. One can open the website and register if you are not registered and can log in and view your classes and do assignments and tutorials. The website is built such that is checks for validation. When registering all fields must be filled with correct details. It also checks for correct passwords and usernames using Express Validation.

 Using mongo db. and Node.js in building my website helped me to achieve my goals.

Recommendations

I had some challenges when I was doing my project but I would happily recommend any developer to use Node.js and Mongo db. because of the ease of using the two technologies combined.

One of the features that need to be included in my project is that when users are taking the exam they should not have access to the internet.

The appearance of the website can be improved to be more dynamic and appealing in terms of colours and the general layout.

More classes can be added in the database so the users can have a wider choice.

1. Mongo DB Website: [↑](#footnote-ref-1)
2. http://www.tuto rialspoint.com/http/http\_methods.htm [↑](#footnote-ref-2)
3. https://nodejs.org/ [↑](#footnote-ref-3)
4. http://www.tutorialspoint.com/nodejs/nodejs\_introduction.htm [↑](#footnote-ref-4)
5. From: https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js [↑](#footnote-ref-5)
6. <https://www.npmjs.com/package/express-validation>

   8 HTML Kick-Start HTML Elements & Documentation [↑](#footnote-ref-6)