18/19

Project Report: Interactive Web Content for STEM

VISHAL VENKATRAMAN
3035271749

COMP4801 | Final Year Project

Table of Contents

Introduction

Background

STEM is the umbrella term for Science, Technology, Engineering and Mathematics. When coupled with "education", it is a reference to school curricula that focus primarily on training students to apply concepts derived from the aforementioned fields to tackle practical real-world challenges. In line with this phenomenon, there are several websites that attempt to involve youngsters in engaging academic discourse. Examples include ***Code.org****:* a non-profit organisation aiming to teach the disparaged of the world about Computer Science *[1]*; the ***National Aeronautics and Space Administration (NASA)*****:** an agency of the United States government that provides lessons on the many facets of space travel *[2]*; and ***"Playground.tensorflow.org"***: an interactive simulation showcasing the capabilities of Neural Networks *[3]*.

However, in the provided examples, either the notion of education is secondary to effective interaction, or the topics covered are too niche and specialized to be considered an encompassment or celebration of STEM. As a response to this, I propose the following project: a **web application** that thoroughly educates its users on a **single topic** through the use of **interactive media** and a **sandbox** environment.

Objective

The primary objective of this project is to create a product that would serve potential users as a guide as they explore everything from the fundamentals to the specialized aspects of a particular topic. This way, the user would have a more complete experience than what is offered on the discussed web platforms. The topic to be discussed will be the computer processor: users would start from an abstract overview of how data is processed within one, and will gradually progress to building their own programs within a provided sandbox environment. Concurrently, the platform would encourage teenagers and young adults to pursue a career in STEM-centric fields.

In line with this, the proposed web application would be targeted towards higher secondary school students, and all materials will be designed to cater to the aforementioned group.

Additional Information

The topic discussed on the proposed web application will be the history and evolution of computer processors. To accomplish this, the work of key historical figures (such as John von Neumann and Alan Turing) will be recreated as interactive and digestible simulations.

One significant simulation would be that of the Turing Machine *[4]*: a hypothetical device proposed by Alan Turing in 1936 that is capable of solving any conceivable problem, granted that the problem is expressed using binary digits, and running any algorithm. In many ways, the Turing Machine is the ideal computer processor, and will be discussed for this very reason.

Another concept to be discussed would be that of the von Neumann Model *[5]*: the architectural model that is employed within most modern computers. Conceived in 1945, this model allows for a processor comprised of an arithmetic component, a processing component, and some memory space dedicated to the aforementioned components (now called registers). A simulation of this model would attempt to explain to users what role(s) each component has within the model.

Quantum computing, being the subject of another simulation, is a computing model that deviates from the current binary systems *[6].* In binary systems, a bit of data is encoded as either a 1 or a 0, but the quantum computing model allows for the bit to be 1 and 0 simultaneously, as an example. This allows for the creation of more complex computer systems than what is available, and represents the future of computing. For this reason, simulations on this model will be included in the web application.

To supplement these simulations, a sandbox environment will be provided. It can be used by users to test concepts demonstrated by the simulations, which would then enable them to grasp the concepts better.

Deliverables

The completion of this project will, firstly, produce a web application will allow users to interact using both the standard array of peripherals (keyboards and mice) and touch-based actions (to cater to mobile platform users). This application will be able to respond asynchronously to any input provided by the user, thus allowing for a more seamless user experience.

Moreover, in terms of the content of the web application, a sandbox environment coupled with educative simulations that can be interacted with will be produced. The concept of a computer processor will be studied from the most abstract level down to the hardware level, after which users would be provided access to a sandbox environment. Within it, they could test and consolidate their knowledge, and the idea behind this is to provide a more holistic learning experience than that provided on other platforms, and therefore promote the viability of a STEM-oriented career.

Furthermore, the adherence to the Progressive Web Application user experience will allow the product to exhibit native mobile application-like behavior: this would grant the application access to features such as offline access and push notifications on mobile devices. This would allow the product to demonstrate cross-platform capabilities, and therefore be available for a larger user base.

Technologies Employed

The implementation of the proposed web application will be accomplished using the MERN stack: this software stack is comprised of MongoDB, Express, React and Node.js. A point to note is that the entirety of the MERN stack is built upon JavaScript, thereby making it a fundamental technology for this project. To facilitate the creation of the interactive materials and sandbox, Java will be used. With regard to Progressive Web Applications, Lighthouse will be used.

The MERN Stack

MongoDB is a NoSQL database *[7]*, and it will be used in this application. It was chosen over SQL-based databases for its greater flexibility with regard to storing data. Using SQL databases, data can be stored only in tables, whereas NoSQL databases allow for document-based storage and key-value based storage, among other flavours, in addition to tables *[7].* Their lack of strongly-defined schema would allow for more flexibility with regard to design, and should permit better database scalability.

React is a JavaScript-based library specifically used to create fast and lightweight user interfaces *[8]*. It embodies the "View" aspect of the Model-View-Controller model: a web design approach that compartmentalises aspects of a web application into the 3 aforementioned aspects. It was chosen over Angular (with which the stack would have been called the MEAN stack) due to its less steep learning curve and its use of the so-called "Virtual Document Object Model (DOM)": a significantly-abstracted version of the structure of an HTML document *[9]*.

Node.js is the JavaScript-based asynchronous runtime environment that will be used to handle the back-end of the proposed web application *[10].* Express will run on top of this environment: it is a robust framework for the development of web and mobile applications *[11].*

Java

Java is a powerful general-purpose programming language that is supported by a large community of libraries and extensions *[12]*. It supports object-oriented programming, which will be the language model employed for this project. Moreover, it can be used to produce relatively high-performance, due to its use of a Just-In-Time (JIT) compiler (source): a runtime compiler *[13].*

Lighthouse

Lighthouse is an open-source tool developed by Google Among other things, it is used to gauge the viability of a web application as a Progressive Web Application *[14].* It will be used towards the end of the project development cycle.

Risks and Mitigation

The most significant risk involved in this project is whether it would be possible to optimize the assets in accordance with Lighthouse standards. A key requirement for a web application to be deemed a PWA is the quality of its performance on low-specification mobile devices *[14]*. The simulations are expected to employ a large volume of visual media assets, which may adversely affect the application's performance on the aforementioned devices. This can be mitigated by arranging for means of data compression within the application, and this workaround will be applied during the testing stage of the process. As an extension of this, the process of optimising the application to meet PWA standards will be fairly time-consuming, and it can only be mitigated by remaining on or ahead of schedule.

Another point to consider involves the hosting of the web application online. A solution that does not involve employing the virtual budget provided by the faculty is the use of web hosting services such as GitHub and GitLab. For the time being, GitHub will remain the solution of choice, for its free web hosting and extensive support community.

Progress

The User Interface

Material UI is a JavaScript implementation of Google's Material Design aesthetic. It is being used to compose the user interface of the product, as it allows for a clean and distraction-free user experience. Given that it has implementations of most commonly-used UI elements (such as sidebars, loading spirals, and so on), it is being used predominantly throughout the UI of the product.

Being among the most widely-used JavaScript UI frameworks on GitHub, it is well-maintained and relatively bug-free. In essence, the only modifications required for the purposes of this project is the modification of its colours: this is done by including the following line of code.

```
import {MuiThemeProvider, createMuiTheme } from '@material-ui/core/styles/';
```

A 'createMuiTheme' object is created, which contains properties that can be modified to change the colour scheme of the final product. At the moment, this is a work in progress, as more time is being spent getting familiar with general workings of the Material UI components.

An optional feature being considered is support for "Dark Mode": this feature would simply darken the UI by replacing all light-shaded colours with darker equivalents. This can be accomplished by designating a "Switch" component to set and modify the state of each component that needs to be darkened. The Material UI website contains a robust implementation of "Dark Mode" that does not require much overhead to operate, and it is being taken as an example to work with.

The Code View

As discussed earlier, a major goal of this project is to facilitate users' learning by providing a Sandbox space, where they would be able to experiment with the code of the simulations provided. To facilitate this, a JavaScript plugin named Ace Editor will be used: it is a lightweight plugin that provides basic text editor functionality such as syntax highlighting and auto-completion, and supports a significant number of languages and themes. In order to minimize the

scope of the project, only JavaScript, Java, HTML and CSS will be the enabled by default. This can be accomplished by simply limiting the number of options available for users to choose from.

Moreover, the number of themes available for selection will be limited to two: one for when Dark Mode is not enabled, and one for when it is enabled. At the moment, the themes ‘terminal’ and ‘tomorrow’ are being used as presets, but if time allows for it, some custom themes that blend better with the product’s design language may be designed.

Future Plans

The Turing Machine

An immediate priority is to fully implement the Turing Machine simulation. Owing to being behind schedule, additional time has been allocated for the completion of this element of the project. The core is being written using Java, and a JavaScript plugin named Paper.js is being used to provide step-by-step visual representations of the Turing Machine.

In addition, given the complex nature of the concept of a Turing Machine, some text and animations have to be provided to help users comprehend the workings of the Turing Machine.

User Authentication and Incentives

Initially, the product was meant to provide an experience that did not require the users to make an account and establish credentials. However, upon considering some of the comments given after the presentation of the project, it became apparent that further scoping is required. At the moment, the plan involves using cookies and sessions to store user data temporarily on the client side, so that the server overhead is reduced.

However, if this project is developed beyond the scope of university, a simple user authentication and data collection system can be implemented: users can use emails to sign-up, and their accounts can store progress information (such as amount of time spent on the site, and the number of actions taken).

PWA Compliance

As mentioned earlier, an important aspect of this project is its optimization for mobile device users, and limited offline functionality. This is an aspect that will be worked on towards the end of the project development cycle, as there are few meaningful features that can be offered offline at the moment. Thus, this aspect of the project is declared an optional feature, and will be implemented only if time allows for it.

Amended Schedule

Proposed Deadline	Corresponding Deliverables
October 20, 2018	~~• Preliminary Research~~ ~~• Implementation Solutions~~ • ~~Solution for hosting of web application~~
January 31, 2018	• Front-end user interface
February 14, 2018	• Turing Machine implementation
February 28, 2019	• von Neumann Architecture implementation
March 15, 2019	• Quantum computing simulations implementation
March 31, 2018	• Turing Machine testing • von Neumann Architecture testing • Quantum Computing testing • *OPTIONAL: PWA Optimization*
April 5, 2019	• Final testing • Final debugging

References

1. Code.org. 2018. *About Us*. [ONLINE] Available at: https://code.org/about. [Accessed 27 September 2018].
2. NASA. 2018. *About Us*. [ONLINE] Available at: https://www.nasa.gov/about/index.html. [Accessed 30 September 2018].
3. Playground.tensorflow.org. 2018. *A Neural Network Playground*. [ONLINE] Available at: https://playground.tensorflow.org/. [Accessed 30 September 2018].
4. Robert Mullins, University of Cambridge. 2012. *Computer Laboratory - What is a Turing Machine?* [ONLINE] Available at: https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html. [Accessed 26 September 2018]
5. Godfrey, M.D., 2013. *Introduction to "The First Draft Report on the EDVAC" by John von Neumann*. Available at: https://sites.google.com/site/michaeldgodfrey/vonneumann/vnedvac.pdf?attredirects=0 . [Accessed 27 September, 2018]
6. Gershenfeld, N, Chuang, I.L, 1998. Quantum Computing with Molecules. *Scientific American*, 66. Available at: http://cba.mit.edu/docs/papers/98.06.sciqc.pdf. [Accessed 30 September, 2018]
7. MongoDB. 2018. *What is MongoDB?*. [ONLINE] Available at: https://www.mongodb.com/what-is-mongodb. [Accessed 30 September 2018].
8. React. 2018. *A JavaScript library for building user interfaces*. [ONLINE] Available at: https://reactjs.org/. [Accessed 30 September 2018].
9. MVC: Model, View, Controller. 2018. *App Organization Explained*. [ONLINE] Available at: https://www.codecademy.com/articles/mvc. [Accessed 30 September 2018].
10. Node.js. 2018. *About Node.js*. [ONLINE] Available at: https://nodejs.org/en/about/. [Accessed 30 September 2018].
11. Express. 2018. *Fast, unopinionated, minimalist web framework for Node.js*. [ONLINE] Available at: https://expressjs.com/. [Accessed 30 September 2018].

12. Java. 2018. *About Java*. [ONLINE] Available at: https://go.java/index.html. [Accessed 30 September 2018].
13. IBM. 2018. *The JIT Compiler*. [ONLINE] Available at: https://www.ibm.com/support/knowledgecenter/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_overview.html. [Accessed 30 September 2018].
14. Google Developers. 2018. *Lighthouse - Tools for Web Developers*. [ONLINE] Available at: https://developers.google.com/web/tools/lighthouse/. [Accessed 30 September 2018].