Student number: 2410516

University of Stirling, computer science

CSCU9N5 assignment

Report

# TABLE OF CONTENTS:

# Product Description

## What is it?

ManxVisual, also abbreviated as MV. ManxVisual is a web application that enforces a different learning style for the basics of programming logic. The idea behind MV is to allow people to learn the basics of how blocks of code such as ‘if statements’ work in a programming language. MV is designed with real life programs in mind, with the terminal being the main area of where the code is executed and displayed.

The twist is that MV displays the terminal similarly in a Mac OS (Operating System) whilst also showing another section that allows the end-user to understand how the blocks of statements are executed. A section of the page is assigned to be the ‘Method to be executed’. This method mimics method from any programming language but it’s displayed visually instead of using text like in an IDE such as Visual Studio.

The components, also known as the blocks of code, are taken from the ‘Components’ section and then dragged into the ‘Method’ section to allow the user to build a custom method. This can then be ‘Run’ and then the output is displayed in the ‘Terminal’.

When I was brainstorming the ideas behind what web application to make, I decided to go for something that would provide an intro to programming without actually writing any code. The problem with teaching programming is that it’s hard to cater for all needs. Some people learn from just deep-diving in; others by learning from the basics with pieces of texts or instructional videos. Then, there’s those that learn visually and need to learn from the nitty-gritty stuff. ManxVisual teaches the very basics of programming logic in a visual but smart way.

## Who is it aimed at?

ManxVisual is aimed at those who want to get into programming but don’t know where to start. It uses visual methods to teach whilst maintaining some text on the screen for explanation purposes.

The age range is any age of 10 and above. Those who love to learn visually are the ideal target due to the dragging and dropping of components involved in the web application.

To be critical, MV could have been used by ages 4 to 10 too but this is only aged 10 and up due to the text potentially containing vocabulary that is out of the standard children’s vocabulary set. Anyone wanting to pursue programming but knows nothing, would find this web application extremely beneficial to grasp the concepts of blocks of code such as ‘if statements’ and ‘for loops’.

## How would it be delivered?

The web application would be delivered via two possible methods. Due to no specifics being outlined in the specification, I will describe two different methods that I could possible use to deliver the web application.

Method one:

This involves compiling the source code and then transferring it into a folder. The source code within the folder would be deployable straight away with the client having to solely copy and paste the source code straight into their web server. The folder would then be zipped up and compressed, finally sent via email.

Method two:

This method involves deploying the web application onto the client’s web server. This would involve signing a contract with the client to handover any of the server credentials. I would then compile the source code and upload the web application via FTP onto the client’s web server.

# Design

## Storyboard:

## Navigation Map:

## Task Analysis:

## Design Decisions:

# Prototype Description

## What does the prototype contain and why?

## How does it differ from the full version?

## What would also be sent along with the prototype to the client?

# Usability Testing

# Web Technology References

Below is a list of web technologies used for this project:

(Format: [INSERT NUMBER]: TECHNOLOGY NAME ([INSERT LINK TO TECHNOLOGY]) – [INSERT REASON FOR USE])

1. Angular 4 (<https://angular.io/>) – Angular4 is the base of the project. This library has helped build the structure of the project by extending HTML's syntax to express my application's components clearly and succinctly. It also provided routing to change pages of the website very easily without having to refresh the page – this used the MVC design pattern to create the web application.
2. Bootstrap (<http://getbootstrap.com/>) – I used Bootstrap to speed up development by using ready-made classes to style components and also lay out the pages of my web application.
3. Angular SortableJS (<https://github.com/SortableJS/angular-sortablejs>) – This library allowed me to create the functionality of easily dragging and dropping the components from one section of the home page onto another section of the page, in turn allowing sorting of elements on the page.
4. JQuery (<https://github.com/jquery/jquery>) – JQuery has been used to allow me to use other libraries such as Angular4 and Bootstrap. It also allowed me to write cleaner and less code.
5. SortableJS (<https://github.com/RubaXa/Sortable>) – SortableJS has been used to support the Angular SortableJS library above.