# Week 1

**HTML**

HTML is quite intuitive to use, my first real time using a markup language and I can understand it well. HTML is like the foundation / blueprint for your webpage, it outlines what goes where but not how it looks or functions

**CSS**

CSS Is the paint or stylings of how a project looks. I will need to look deeper into best practsises regarding inline CSS and how much to use, otherwise it might contaminate my whole document.

**Java**

Java is the functionality, if CSS is the paint then JS would be the wiring behind the walls, the code that makes things interactive. I like ODD languages so it's nice to use one in this class, it's not C# but it's not that different.

Text

Description automatically generated

**VUE**

VUE kinda turns html code / pages into objects, making them easily interchangeable components or pages. The VUE checkbox was pretty hard to setup, but it helped me gain a much larger understanding of how vue operates.

## Reflections

GIT - <https://github.com/CameronHart1/Responsive-Web-AppsTwo/tree/master/WeeklyPracDocs>

# Week 2

Responsiveness of two similar sites, Garden.org and Gardeningknowhow.com  
  
Garden.org – does seem to have some breakpoints, it is optimized for most mobile devices and screen sizes, although at 280x653 it’s header starts to cover page content. Navigation isn’t the best and neither is presentation

Gardeningknowhow.com – well optimized uses breakpoints with multiple screen layout, the design looks nice and content doesn’t get hidden when resizing (within realistic measures).

##### Epic

As a user I want a site to help me manage when and how to take care of the plants in my garden or around the house. I want tools to help me plan how to layout my garden; giving me an idea of how it might look. I want the ability to ask other gardeners opinions on questions I might have, and I want to join some gardening groups where I can chat with gardeners that share my interests.

##### User Stories

|  |  |
| --- | --- |
| User Story | Sub-stories |
| As a user I want a site to help me manage the care of my plants | * Garden journal to record care given * Recommend care based on plant type, location etc. |
| As a user I want planning tools that also visualize my garden | * Garden planner that can easily sketch a gardens shape and calculate available space. * 2d or 3d models with varying colors based on the selected plants |
| As a user I want to ask other gardeners about questions that are bothering me | * A forum space with the ability to add question threads * Being able to search the forum for similar questions |
| As a user I want to chat with other users that share my gardening interests | * Public and private gardening groups that use the forum style * The ability to search for public groups |

Diagram

Description automatically generated

## Reflections

Using Vue, It was complicated to use the router addon for page navigation, having to set up routes for the different pages, to make them accessible from URL paths took a little to learn. CSS setting up colors was relatively simple too.

I already have some experience with UCD (did the class last tri) so a lot of the content was quite familiar. I made sure to write a big epic, splitting it into stories then functions for my program.

GIT - <https://github.com/CameronHart1/Responsive-Web-AppsTwo/tree/master/WeeklyPracDocs>

Chart

Description automatically generated

Chart

Description automatically generated with medium confidence

Diagram

Description automatically generated

# Week 3

## Reflections

Text

Description automatically generated

Javascript is a lot like C#, with both being OOD languages. This makes getting to grips with JS much easier.

JS does remind me a little of Python, how variables don’t need specified types when being declared (though it can be done in c still there seems to be less hierarchy) and can easily be converted from type to type.

Attaching a foreach function to the array was quite interesting, additionally having access to the index within the foreach loop was interesting.

As I already have experience from other modules using js with html (using “document.getelementbyID()” to get html elements, etc.) I figured I would mess around with some js methods in a console style app, and figure out how to debug js with vscode.

Project progress

* Finalized my concept of the garden-based site. Made some color palette choices.

# Week 4

## Reflections

Text

Description automatically generated

Text

Description automatically generated

Chart

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

It is interesting how vue is smart enough to cache results of methods that depend on computed properties.

Declaritive rendering seems to be quite the umbrella term, as it can include anything that removes or adds html to the .app program, based on some condition. In a way the vue router I use is a big form of declarative rendering.

It is interesting to see that show only changes the visibility of the html object, while if actually removes or adds html to the .app. I had already used v-for in the making of the todo list, it is a neat function. I didn’t know that v-else was a thing either.

Project progress

* Did most of my coding for my site, finishing a basic demo of the profile home page.

# Week 5

Components in Vue are just like classes in C#, although with the added bonus of only being rendered or added to the webpage when they are needed, optimizing speed for user. Components can be nested and data can be sent to and received from children (although ideally you aren’t constantly receiving data from them as that can get confusing to track).

Useful examples:

Two-way binding for user input will be useful in the search bar of my project

Dynamic components for search filters depending on the category of the search.

Handling User input:

In my own programs I prefer to keep the method and mounting onto the component in the same file, as opposed to attaching everything later in the main.js like the example would show. It is interesting how v-on isn’t only on click but on other interactions e.g., v-on: close.

Props – Many of my children components will use props, like my forum entry component. Being able to send data to children to be used in certain places / ways is super useful

Custom events – When saving or selecting certain Journal plans my program will need too send custom bits of data to other components, this can be done through making custom events with the emits option.

Dynamic and async components – I will want to keep certain pieces of data active when tabbed into different areas, especially with the Forum

## Reflections

Graphical user interface, text, application

Description automatically generated

I already had a decent understanding of components, learning about async and dynamic components was very interesting and will be useful in my project.

The two way data binding should also be useful when creating my search bars.

Project progress

* Started looking at alternative color schemes
* Converted my first html project into Vue project

# Week 6

Graphical user interface

Description automatically generated

Text

Description automatically generated

Tasks one and two and three are all summed up in this component, I use v-for to create the three checkboxes, use v-model on Seltags to view what checkboxes have been ticked and the lazy modifier on the v-model so it only updates after a change event, instead of an input event. Checkboxes switching states are counted as a change event, so this could be redundant but the extra insurance it provides is nice.

## Reflections

I had already used v-for when implementing the todo list in my project (this, that) however it was interesting that the input type being Checkbox (instead of checkbox) prevented the v-model from working as it wouldn’t let me set the value. It was also interesting combining the v-for and v-model into a single function, while doing so I learnt more about props and computed values.

Github:

Project Progress

* Added Vue components and routing to the page, made the search bar clickable and interactive, set up a component for the forum blog posts, added an icon and navigation to the page.

# Text Description automatically generatedText Description automatically generatedText Description automatically generatedWeek 7

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedMy website has already been using local registration, this can be seen in the nested components when editing the hosted site. Only the components that are active on the webpage are being loaded by the user.

Again I have been using props, I experimented with default value for props, which I do in the above example. Along with type validation (stating that it needs to be an array).

I use a custom event to emit a message to the parent main page on what the last change to the checklist was. It was interesting learning that you can pass the event, and with it data like value and checked (from a checkbox). Slots are also used to set the name of the list, I did have to use v-slot:xxxx to assign the template to the correct named slot however.

Github:

Project progress

* Login functionality
* V-model for forum search