|  |  |
| --- | --- |
| A picture of a winding road and trees  Journal  WEB601 | Aj Fry |

Contents

[Week 1 1](#_Toc24986163)

[Class 1 1](#_Toc24986164)

[Class 2 2](#_Toc24986165)

[Week 2 2](#_Toc24986166)

[Class 1 2](#_Toc24986167)

[Class 2 3](#_Toc24986168)

[Week 3 4](#_Toc24986169)

[Class 1 4](#_Toc24986170)

[Class 2 4](#_Toc24986171)

[Week 4 5](#_Toc24986172)

[Class 1 5](#_Toc24986173)

[Class2 6](#_Toc24986174)

[Week 5 6](#_Toc24986175)

[Class 1 6](#_Toc24986176)

[Class 2 8](#_Toc24986177)

[Week 6 9](#_Toc24986178)

[Class 1 9](#_Toc24986179)

# Week 1

## Class 1

This week we focused on learning how to work with Git and git hub. Github is a website which can be used to manage code. Using git we can upload our code, divide our code into different sections and merge code branches together. Once we are finished working on our code we can commit and push it to GitHub which stores the code and allows other GitHub users to see and possibly contribute to the code.

Github hub also functions as a sort of social network for programmers. Many programmers use GitHub to network with others in the industry or as a CV for job interviews. Github hub tracks how much a user interacts with the site/ how much they contribute to repositories. This activity tracking allows potential employers to see how the user works. Activity tracking also allows users to form bonds with each other letting them effectively social network.

Git and by extension GitHub let users manage their code through branching. A branch is a copy of a set of code which can be work on without effecting the main piece of code. Branches can be created form a branch and more than one branch can be created at a time. Branching allows multiple programmers to all work on a piece of code at the same time. Multiple people can each have their own branches allowing them to not have to work about other users overriding the work they have completed.

This journal is late as I made a mistake when first trying to commit it and manage to select my documents folder as both the source and destination of my commits this result in Visual studio code and git breaking and me having to install third-party software to uninstall any residual files.

## Class 2

During the first week of this class, we learnt about the various git commands which we will need to complete the git/GitHub portion of this assignment.

* The first step to using git for version control is to add credentials.
* git config --global user.email "hello@example.com" this is an example of the command to do this.
* In my case, I had to do this twice once for username and once for email.
* The next step is to create and navigate to a folder which can be used as a repository.
* Next, we have to use the 'git clone "URL from git hub"' command to get the files from GitHub.
* After the changes were complete we needed to create a new branch which 'git checkout -b "Name Of New branch"'.
* After this, we had to use the 'git add "fileName.extention"' to select the file(s) we wanted to commit.
* Next, we used the 'git commit -m "Message"' TO Commit the changes and add a comment to the commit.
* Finally, we used 'git push origin "Target Branch"' to send the changes to GitHub.

An alternate thing we learnt this week is how to merge branches together. The process work by creating multiple branches navigating to one and them using 'git merge "other branch name"' to merge the two. This function allows us to be able to work on small parts of a project individually and then have them all fuse together once they are finished.

# Week 2

## Class 1

This week we have started learning about web design. As of the first lesson we have look at storyboards, user stories and wireframes. Our teacher walked us through the importance of planning a document and how once we are in the industry we will be required to fulfil multiple roles based on what is needed. This means that it is important to be able to do all of the different roles in web development. Wireframes are an important part of the web development as it allows the developers to have something to show their client and have that client be satisfied. One of the ways that we were shown to do this is by paper prototypes. Paper prototypes are when you hand draw a design, this has many benefits, including not having to create anything meaning that if the client does not like the design the time spent developing it isn't as wasted. Clients are have also been described in a way which makes me think they are 'decision challenged' meaning that they will make bad decision at inappropriate times, for example a client may not feel like they could ask for changes if you have already started with the website by with paper prototypes the client could "screw them up and throw them away" allowing them to make changes as they see fit. Additionally, client asks for additions and changes throughout the project had the distinction that you can change the paper thing but not the work I have already done helps create boundaries for the client.

Storyboards and use-case descriptions allow as to have examples of how we expect our users to interact which the website which can also be explained to the client or used to find oversights.

So good advice we got from this lesson includes. Never start at a start-up, they will expect you to do everything and won't provide mentorship as a large company will. All way start at a job for more than a year or else it will raise flags with future employers. If a client 'has an idea' explain to them how much it would cost to get them to drop that idea.

## Class 2

For the second lesson we were meant to create a site based on the framework we designed in the first lesson of the week. I was assigned to the CSS of the website which was a struggle as one of the other members of the team added a CSS template which somehow still overnight my changes after I started making changes that actually work I manage to loses them once I started merging. I managed to merge a few times with some success but after I tried to merge back to the master, my changes were removed even though I thought I had told It not to over right that part of the code.

One of the main things that I learned was how to pull reports after I had already downloaded it once. Once selecting the correct branch The pull command will allow me to gain any changes to that report. I also learnt how to approve merges on Github as it was required to work with the website. As I ran into difficulties near the end of the lesson I might need to look into and practice merging document so I won’t make the mistake I made again.

Week 3

## Class 1

For the first class of this week were discussed the different types of frameworks and markup languages there are and how popular they are. This is both interesting and important as I might want to pick languages which are popular or in high demand. We also discussed basic JS functions, they are similar to other programming languages, just like other languages JS has it's own differences for example number is less rigid than in other languages. This sort of thing is important as all languages have their own syntax and understanding that is key to understanding the language. Were also covered equal and not equal statements in JS we use === or !== for javascript which is different from other programming languages which just use == and !=.

## Class 2

For the second lesson we talked about the differences between if statements and switch statements. The operator || is JS version of or which is useful to know as I have a tendance to create large complex and or statements. It is important to combine statements like this or you would need to create many nested statements which is one of the first things I learnt I wasn't meant to do when starting this course. I don't really understand error functions but I will look into it as it seems important to be able to prevent errors in my code so I Should look into it latter.

Functions in JS are very similar to what I am used to and are called similarly to MySQL functions. I should look for some differences SO That I don't make error or incorrect assumptions.

# Week 4

## Class 1

I have decided on my project. For this class, I will make a generic product web site which allows small businesses to advertise their products. The website will allow people who don't have websites of their own can easily advertise online as well as people who want to find the product they want.

We started this class with a high-level description of a basic description of the product. We went over a basic diagram and with basic terminology for the use of a web app. We have covered this in a previous lesson by this description made it made more sense to me. Being able to discuss a project in the most basic terms is important when it comes to talking to clients as client might get lost if they don't understand what you are trying to communicate. This is an important skill for me to learn as I struggle to clearly explain my point and to give the necessary details.

## Class2

I struggling to understand what express.js is. I believe that express is a code library built on node.js, it's most important feature is how it is customisable. Node.js is a library which allows for web communications. The node.js is allowed for connections between servers and other computers by creating connections called sockets, which is a concept that I am struggling to understand but I think that this might be a little advanced for where I currently am at. How I understand is that Javascript can communicate with the hardware of a server so node.js allows for translation between the two.

Note: I called node, js and express.js libraries they are actually frameworks.

While using code libraries it is important to make sure that they are up to date before they become an important part of your app. This can be done by using GitHub to check how active the repository is when how many issues there are and when it was last updated. The reason only using currently maintained code is important is that if the code is not being maintained error and issues will occur and won't be resolved. It is also important to find workarounds for issues as the people updating the repository might not resolve the issue straight away.

# Week 5

## Class 1

Just to some technical issues, I was late for class. I entered class and was encountered by the discussion of using express.js to code we were going over the next() command which lets a program move onto the next piece of code which is useful after something like an if statement. We also covered use, req and res which is what I missed so I will need to figure out what this means at some other time. How I believe that these works are that when the code needs to take a response from the internet we use req to get the result and when we want to respond to it we use res. We also covered the term middle were which is used to describe the action which takes place between the requests of the user and the responses of the system, I think I'm not really sure.

The next function moves to the next action among the middleware. We also talked about app.use and app.get which are used to load the functions such as req and res from express.js. The different from app.use and app.get is that the app.use doesn't need to querry/receive information. Where app.get goes and retrieves information.

We also covered routers which I do not understand but they seem important. I think that this will make more sense once I try and use it, I should look up some tutorial to try it out.

app.listin allows us to specify what to listen for to run the code. app.listen opens a port allowing the user to tap into it to access the code which has been set up for the user to run.

Remember:

Require statement

middleware

route

## Class 2

During this lesson we covered the differents between javascript classes, functions and props. These are called components. Functions are basic they allow for code to quickly run and don't clutter your code like a class will. Functions consider stateless as they only have the ability to run their code and don't remember any of the other functions. Classes are more complicated than Functions have additional features such as storing variables. As they store variables they are considered stateful. Although some people might recommend not using classes out teach does as it is extremely useful to be able to take advantages of classes Props are components which have the ability to store data and that is it. They can be passed to classes and used to more around data.

Examples:

Funtion: preform a spicific equation.

Classes: preform an equation based on its state

Props: pass a value into classes to be added.

We covered how to take advantage of reacting parent-child relationships. This was something that I took advantage of for milestone one.A react component can be placed within another component and then that component can be called from the root. This allows us to be able to render multiple components at the same time. I currently am not sure how to properly load a component after a piece of child code. For example I would like to have my header and footer on every page and then content in between but I don't know how to load the footer without loading it as a child as well. I will need to look into this in the future, I believe that it has some relation to boilerplates so that might be a place to look into.

We covered different types of styling. Inline-styles are bad and I will not consider using them. I am currently using a CSS file but there is an additional type of file which I might want to look into doing. Style components allow us to download style templates and applies them to each component, this is similar to an inline style but doesn't have a lot of the downsides to inline styles. I might look into this as I am not good at web designs so this could help me make my website more polished looking. It also prevents me from forgetting an interfering tag which could coarse issues, which is a common problem I have.

Another example is CSS in javascript which lets us write CSS in javascript. This isn't popular as it can add a large amount of code to the project, but it allows for a large amount of work to be performed at once.

# Week 6

## Non-contact week

During this week I didn’t do much in the way of WEB work.

Most of the work I have done is based on the codacadimy express tutorials. I thought this might help me understand express because I’m still not getting it, but I am struggling to do these tutorials.

I can almost get the functions, but I am unsure about how the different components work together. I normally get the syntax for the examples slightly off and then I am unsure on how to work with what I have created. I also have no idea how to go about implementing this for my project.

Note from future A.J.

At some point in this week, I completed a tutorial which was clearer then code academy. I was still a little confused, but I made express not as scary, I, however, forgot to write this at the time so I don’t have a link.

I think I need to find some other form of tutorial because I don’t understand the work I have done this week. I feel like I have spent too much time on reach and I’m not able to move on with my project.

# Week 7

## Class 1

During this class, we looked at the different types of JavaScript objects which we will use with React.

One of the most important things we covered today is ReactDOM. ReactDOM is the line of code that we use to render javascript and react components. Generally, it is placed at the end of the main file you a running. Once called it will not only render what the component that is called has in it but also all of the other components which are nested inside of it.

This is extremely useful as it would allow me to set up a page using react components like normal HTML tags, I think this makes React more appreciable as it is similar to what I have already learnt.

We also discussed the difference between functions and classes. Before I start I would like to mention that classes in JS aren’t the same as classes in other languages and can’t be called in the same way because JS is not an object-based language. A function is just code so it can’t do more than perform an action, for example, a function could perform basic maths this is called being stateless. A class, on the other hand, can store information as well as functionality. The ability to store information is called state.

Components in react can be passed props which are similar to how in other languages different aspects of the code are parsed information when they are called. Although the design behind props are more dynamic and the user doesn’t have to specify what information is needed. Meaning if a user doesn’t need a specific value it’s not hardcoded in so no errors will occur. I am not sure however if not parsing in a required variable will cause the website to crash.

I feel that I need to practice with parsing props through an app as I’m am unsure about how they should work. I also don’t know if you can parse components through props which I feel like could be useful when it comes to rendering my pages as I plan on using containers to hold other components.

## Class 2

This week we started with git classroom. During this we used the new tool to learn about the React life cycle. The react life cycle contains 5 different mainly used methods.

1. Constructor
2. render
3. componentDidMount
4. componentWillumount
5. componentDidUpdate

There are more lifecycles that this except these are the only one which is commonly used. Each component runs at a different type during the lifecycle of a component.

The constructor is run before the component is mounted and lets the user set up things such as the state of objects and event binds.

The constructor is limited as too what they can do as their code takes place before the component is mount and therefore doesn't have a state yet.

Render is the only composite part of a class. Render is the main part of the code and render what the company is meant to render. Render can not be used to differently influent the state of the component or to directly interact with the browser.

componentDidMount will be run directly after the code was mounted and therefore can be used to set up things like subscriptions.

compoentDidUpdate is run when the component is updated in some way. By using this we can invoke code after a change is made allowing

us to be able to re-render the component after anything is changed.

commpentWillUnMount lets us run any code that is needed before the competed is unmounted and destroyed.

Knowledge of the life cycle react allows us to fully take advantage of how react compoents render code allowing us to be able to have certain bits of code rendered at certain times.

# Week 8

## Class 1

For this class, we finally started to create react apps from scratch.

Creating an app wasn’t as hard as I thought it would be as I have already made a couple trying to better understand the classwork.

However, I made a few mistakes and result in having to restart, I also had an issue where I forgot to install one of the required packages.

I think that I need to practice command line a bit more to try and improve my skills with it. I think the best way to do this would be to repeat this excise. One of the biggest mistakes that I made was not CDing into the correct directory making my app non-functional.

## Class 2

This class was a field trip of sorts. For today's lesson, we meet at the Kofi lounge to present presentations on full-stack development that we had researched. For the first half of the class we attempted to do some open source contribution. I struggled with this as my confidence with web development is not very high. Most of the task with I found that I could help with had forum post with someone asking if it had already been solved making it hard for me to find something relevant to do.

In the end, I found a ‘your first contribution’ repository and did that. In the future when my skill advances a little more and I have more confidence then I would like to do this again.

For the presentations, I didn’t realise that my group was unsure of what to do and therefore only did about a third of the presentation not realising they were waiting for me to tell them something to do. For the most part the presentation we came up with was ok. We as a group could have used some more practice. Our group topic overlaps with another group but we talked about different thing making the two presentations complement each other.

In the future I need to better commutate with people I think once my commination skills improve I will also be better at open source development.

# Week 9

## Class 1

For this class we created played with SQL, I am very final with SQL as I have been using it for the last two year due to course work. For this class we created and populated a couple of tables for use in a future activity I found this easy as I have created many tables before and am familiar with the syntax. I like always to struggle to think of test data as it is always hard to think of new random information on the spot.

We also briefly looked at nodemon which is a library which lets us create and run our back-end API. It seems like an interesting tool. I hope that the practice I put into command line will help me to better implement it.

## Class 2

This session fell to pieces on me. The express stuff that was some to use still didn’t make sense to me. I understood the basics of how we are creating functions in the code to represent standard CRUD functionality. I understood that the problem is I don’t understand how you translate express routes into database calls.

It is possible that the main reason that I struggled with this is I couldn’t get the example to run. It seems to be a problem with nodemon/my laptop. According to Ali, I need to reinstall visual studio.

Another part of the lesson I didn’t understand was what all of the different commands we had available me, there are the standard ones like getting and post, but then there are other like patch and push which I don’t know what they do, there are also more options than I am used to with SQL. Because of this I need to look into what all of these mean and what would be useful for my project.

Update: reinstalling didn’t work I’m going to need to bring my spear computer which seems to be working fine. I think the problem has something to do with a conflict between visual studio and visual studio code which I find ironic.

# Week 10

## Class 1

For this class, we went over how to connect the back end to the front end with all the different pieces. Due to my switch in computer I was a little behind as I had to quickly make my database again. I once again am not closer to understanding express.

I was recommended to do the code academy Java script basic to try and better understand express. I have started these and am still no closer to understanding express.

I got lost trying to follow Ali in class. His code however is well commented, and I should have no problem going over it in my own pace.

Update: It is a little less easy to understand than I originally thought as it has a lot of files which I don’t understand the purpose. I understand the basic that you create a route, the route is associated with a specific function which alters the table. I don’t, however, understand which files are relevant why the routes connect through a separate Js file. Also, I don’t understand where the middleware which was discussed in previous classes comes into this.

## Class 2

I created an API area in my project which is like its own app. I not sure if this is correct but I have it set up with its own readme and the like. This file has a lot of different files in it which were created with the command that I ran. I have no idea how to work with any of this.

I have started setting up some basic routes which can be called from insomnia I don’t know how I can connect any of these to my application.

As part of these classes, I created a basic table showing what tables I will need to complete my project.

From here I don’t know if I am meant to create a set of functions for each of my table that was planned or if I am meant to be setting up routes first.

# Week 11

## Class 1

I forgot to write about this class. Honestly, I have no idea what happened at this point. I think I was desperately trying to understand my project, which resulted in me forgetting about this blog, sorry.

## Class 2

For this class, we did peer coding in which a switch between a few people who were farther along than I was. This allowed me to catch up with the work I was struggling with.

First, someone went through and help me to create routes with the correct middleware attached and they connected them with the functionality.

As part of this, they help me with the file structure with the restful.API containing the routes which were connected to a file named after the table I was trying to edit and was connected through a router. We also briefly went over, not enough for me to write my own but enough to stop it from breaking my program.

I stay late a class and had someone help me set up my first working database calls I have got all 4 of my required functions working in insomnia. I have also got the very basics of the get working with my site.

Now that I understand what I’m doing it should be a matter of time before I get it all working. However, I am running out of time which could be a problem.

# Week 12

## Class 1

My site was not complete to stand today, I ran out of time as I was only able to understand what I was doing on Friday. My site had a lot of components thrown onto the page to get the functionality wired up, the CSS was basically non exist due to layout changes and the delete function was not implemented as I ran into some trouble last minute.

I have been given an extension. For this extension, I need to return to my planning to better choose styling including colours and spacing. I need to move the components to the places they are meant to go. I should also start implement react strap as this will allow me to better lay my page out.

## Class 2

So redux seems to be great. Redux is like an overhead filing cabinet which other components can access. The information is stored in a store and is like the state which components have had until this point. With redux all components can have access to the same information.

Redux can be directly accessed on its own it is instead accessed by actions and reducers. I am unsure how these work currently so I will need to look into them later on.

Redux can use either switch or if statement to decide what action should be preform based on the request type. I don’t fully understand this yet but it seems to be intuitive.

# Week 13

## Class 1

I spent half this lesson catching up on homework I didn’t do.

After catching up we looked at flux. MVC vs Flux

Model view controller has a large number of branches which it has to manage many views and models per controller, and they are all linked.

With flux, other files manage the links reducing the complexity of the file.

Currently, I don’t understand how this relates to redux, but the terminology such as Action Dispatcher Store View makes senses to me with what I understand about redux.

## Class 2

This lesson I was sick.

According to the class we when over a tutorial about how to implement Redux. Many of my classmates had issues getting the tutorial to work but there is an example provided by the tutor which apparently works better. I will need to go over this tutorial in my own time.

# Week 14

## Class 1

Today in class we compared the Redux cycle to an insurance company.

A person goes to AA with a for which

Redux cycle:

Action Creation => Client

Action => Form

Dispatch => Front Desk

Reducers

State

When designing redux we have to focus on Action creation and action as these will decide how we handle the creation.

When were use redux we are updating a single array this is important to remember as it is commonly believed that the redux functions are constantly creating new array. When creating the array we must use '[]' so the reducer knows what we are trying to create.

After creating the reducer we then perform an action based on what type of requests is being passed.When naming action type the standard is to have the title being all caps with an underscore and not a dash.

We have the choice between a switch and an if statements, however major developers avoid switching the reason for this is unknown. I may need to look into the reason for this as my inclination is to use switch statements but if they are a specific reason to avoid this it would be best that I do.

'.filter' can take an error function and can take out an item from an array and check the rest of the array. After this point the filtered name is gone from the array. This may be useful for my project to remove invalid entries in my tables such as if a menu is deleted. I should look into more about this as I am unsure how it works fully and I only a have a ruffe understanding of why this would be useful. After creating all of the reducers we then combine them.

Redux allows us to remover much of the state from our react components. This is extremely important as the parsing of information is what is creating a lot of my issues as well as limiting what I can currently implement. The ability to be able to manage information in the way redux does will allow me to manage things such as how to select and edit and item as in my design these are managed by different components which are barely related.

## Class 2

This class we went over two examples for a redux based login system.

The first example let us log in with test and the second let us register and delete accounts.

This is similar to what I wish to do with my web app. I wish to have a login screen which changes which components are displayed where only a login user can see and edit menus for the business they manage this is a good starting point.

While we were trying to get the examples too work we had a few errors: We needed to use npm i -g react-scripts@latest to install the appropriate packages for the program to run.

The program had used a config file which was empty, to fix this we had to remove any mention of it from the files in the program.

I think on my site it would best to place the log in place of any educational panels with the registration button taking the user to a new page. The code in the examples is similar to that which is used with our restful API.But it is slightly different and in other places it doesn't seem the same. One large part of this project is it uses a false back end which is something I don't understand and will probably need to look into to properly understand my program.

# Week 15

## Class 1

For these classes, we went over how to connect redux to react. The work we did in class looks nothing like what I was trying to implement into my project. Also, I had a slightly more complex version with an extra file for my constraints.

I need to go over the example given in class to better understand where these differences are coming from. Luckily we have a repository to go through.

I should be able to understand this as we went through the repository in class and I managed to follow along an get it mostly working. The error I had were most likely due to me not loading a gitignore in the correct place.

## Class 2

We went over thunk which allows functions to be parsed like variables.

This is an interesting concept I can see being useful in another project as I have need a similar thing in SDV before. However I don’t understand how I should implement this in my project.

I haven’t been planning around this functionality so I have nowhere to use it.

Hopefully after going over it a little more, I can see how I should implement it.