# 11/01/23

* I have received my project which is abstract syntax trees to source code generation
* John has sent me some example projects that are like what I will be doing, I have spent most of the day examining what they actually do
* I am not that familiar with node.js and the node packet manager so it took me longer than expected to get the first example up and running, when I did get it running I was able to run a few of the provided test cases and see what happened.
* I decided to dig a bit more into how this worked, but this required me to do some quick research on what an abstract syntax tree actually is, from my understanding it seems to take a piece of code and extract its structure and store it in a more universal format free from the strict syntax of most programming languages hence why it is ‘abstract’.
* I found a tool within one of the example projects that appears to convert from JavaScript to AST, I made a very simple JavaScript of my own to observe the output it produces in the form of an AST in what appears to be JSON
* I feel like this project could be more challenging than I originally thought after looking at the examples.

# 12/01/23

* I started the day by looking into the acorn library in some more detail, it appears to be a parser for JavaScript into AST, it seems to follow a schema called the ESTree spec, it supports the ES5 grammar it seems.
* From the examples I have been sent by John already it seems that conversion from JavaScript to python and ruby has already been done. However what would be good is if you could go backwards and say convert from python to JavaScript or to a lower level language such as Java or even C.
* After some more research I have discovered that there is no universal AST format.
* It seems that different languages have their own AST formats.
* Discovered a useful website here <https://astexplorer.net/> that seems to convert many languages into AST using various different parsers.
* Trying to use the acorn library has been a challenging experience so far, this is due to the fact I thought I knew enough JavaScript to start tinkering with it. This is not the case. I have then spent the remaining part of this day learning the basics of objects in JavaScript before playing around more with the library. Tomorrow, I hope to be able to manipulate the AST objects created by the acorn library more easily.
* I have also ran into issues getting VSCode IntelliSense to recognise the acorn library, I am still having this issue unfortunately
* So far, the project has not started off as smoothly as I expected.

# 13/01/23

* I started off the day by thinking, that surely some tool already existed online for translations between various programming language, what I discovered was quite surprising
* I soon discovered that yes tools exist but they tend to be very specific, for example there is no general tool for converting python to any other language. From what I gathered the tools seem to be much more specialised such as from Python to Java, not a one size fits all approach, as I had previously been thinking of
* I soon came across a good article about transpilers, now from the examples that John sent me it was already using transpilers but I found it hard to understand what it was actually doing. The time I took to understand the basics of JavaScript objects have helped when I took another look over the source code again.
* After doing some more research on transpilers I have discovered become more familiar with the way they work.
* The example project I have been looking at appears to create source code straight from a JavaScript AST, in the ESTree format. Most transpilers seem to convert the AST of the original language to the AST for a different language. This to me makes much more sense than trying to generate source code from the AST of a different programming language.
* I have found that Python appears to have a built in source code to AST library.
* Over the weekend I aim to try and understand the python AST library and see if I can get a very basic JavaScript ESTree to a python AST.

# 15/01/23

* I didn’t have a lot of time to work on the project today
* I started off by investing the python AST grammar and library a bit more
* I discovered that the library has a built in unparse function for turning python ASTs back into python source code
* I was able to create a simple AST my self and successfully turned it back into python source code
* My thinking is that if I map each node in ESTree to a node of a similar functionality in a python AST I can very easily go from JavaScript to Python without the need to mess around with syntax and writing source code myself
* That is my goal for the coming week

# 16/01/23

* I am very happy with the progress that I have made so far
* I have been able to implement the definition of functions when going from JavaScript to python
* Along with this I have also implemented calling of functions with parameters passed to them
* I have also been able to implement console.log in JavaScript into print in python
* It was not easy, but I am pleased that I was able to do it in one day
* I have discovered that GO also has a built in AST library, I am yet to understand if it contains a tool for unparsing
* I have also done some research about how an entire library could be translated into another language, from what I have found it seems that many popular libraries have a specific variation for popular languages. From what I have read transpiling libraries seems to be less favourable than converting libraries by hand, generally because converting code by hand will generally result in more efficient programs than when run through a transpiler, especially for lower level languages such as C
* I cam across a post somewhere (I can’t find it again for the life of me), it was someone asking if there was a tool for converting snippets of code from one language to another. I can see a use for such a tool for someone who is not literate in many languages such as scientists.
* I have my meeting with John tomorrow so I will bring this up there

# Week 2

# 17/01/23

* Just had my first meeting with John today, he provided a better direction for the project to take than I had originally thought of
* He has suggested that I should begin to look at various no-code options and see how they produce source code
* The idea being that I could write code in different languages and then have it converted into source code for different platforms such as an IOS or Android app
* My work I have done on the beginning of translation between JavaScript and python ASTs has not been wasted as it has taught me a lot about ASTs
* I think the best direction to take this is to try and get some sort of small example working where I can create an “application” (in python probably) then have this in line with a current framework
* I gave the project some more thought and have been struggling to see what the big picture is, from what I understand it is a tool that can convert apps between various frameworks, but I am unsure so I will message John on teams to try and clarify these issues before trying to start research tomorrow
* I just got a response on teams and it has helped me to see the bigger picture of the project
* What I was originally thinking of a tool to convert an app between frameworks was correct but it has been made much clearer

# 18/01/23

* I started the day by researching various frameworks that I could use, John suggested electron and I looked into that
* I set up a simple project to understand the structure of an electron program
* It uses HTML for the front end and JavaScript with node.js for the back end
* I did some research to get a deeper understanding of it and it appears to run on a version of chromium, so apps will be created in a similar way to web apps
* After playing around with electron I started to think about how a more abstract framework could be created in a different source language and then converted into an electron project
* The way I see it is that an app will be created in a more abstract way, then this abstraction can be converted into specific elements in different frameworks, effectively allowing developers to write their app once
* I have chosen Python as the language in which the abstract app will be written in as it is very popular but also easier to learn for people new to coding
* Before starting to create an app in this abstract form I think it would be wise to be able to create and set up a blank electron project automatically
* So a few hours later and I have been able to get an electron project automatically set up and created using python

# 19/01/23

* I started off the day by thinking about how python code could be converted to JavaScript, before I had been going from JavaScript to python, but I will need to go from python to JavaScript
* I thought surely this has been done already, so I went back to the AST explorer website I has been using earlier and sure enough it converts python to AST but not only that it converts it to the ESTree format, with a few differences I noticed, such as python standard library functions are not represented properly
* Surely an ESTree to JavaScript converter of some sort exists, and I would be correct
* I have downloaded a library called “estree-util-to-js”
* I started playing around with this and seeing how basic functions are converted
* It doesn’t work with python standard library functions so I think I will need to do some experimenting and see if I can get these to work
* This is a good as it allows me to try and get something working that will be of use to people without having to completely “reinvent the wheel”
* A few hours later and I have found the library the AST explorer website uses for converting python to ESTree
* It is called filbert and I have downloaded it and started to play around with it
* I feel as though as I have a more clear direction it is easier to stay focused on what is important

# 20/01/23

* After giving the project some more thought I have no decide that the abstract app will be written in python and the tool to create the app in different frameworks from the AST will be written in JavaScript, purely because handling ASTs in JavaScript is easier and the parsing and un-parisng libraries I want to use are only available in JavaScript
* So I think I have finally decided on the architecture
* The app will be written in a very abstract form in python, this will then be turned into an AST, I will then manipulate this tree to include parts that are needed specific to the framework that I am converting to
* I am now going to re create the auto creation of an electron program in JavaScript using ASTs
* I want to be able to run this program from the command line with arguments for easier use, so my first task is to figure out how to provide arguments to JavaScript programs
* So I have been able to replicate the functionality the python script I had created a few days ago, it now also uses ASTs to form the basic JavaScript needed for an electron app

# 23/01/23

* Today I started off thinking about how I could insert code into already existing JavaScript, I am trying to insert a simple console.log into the start up method for an electron app using ASTs
* I first need a way to traverse an AST properly, so I had a look online
* I found that it is called traversing an AST and it appears to use a depth first search, which uses preorder traversal, so from my understanding it will go down a “branch” until it reaches the end of the “branch”
* I then found a tutorial online for how to traverse an AST using nodes, it is not exactly the same as what I am doing but I have given it a go and been able to traverse the tree and visit each node exactly once
* It took me longer than expected to get the traversal working as the tutorial was not very good at explaining the things it was doing an the fact that I am having to adapt if for my need
* I still haven’t been able to do any manipulation of the AST
* I was thinking about the how to guide as well, and I think this could be a potential as something to write about as the tutorial is very bad at explaining and it is not for my exact ASTs
* Tomorrow I have my meeting but I also want to try and get some basic manipulation of the AST if possible

# Week 3

# 24/01/23

* I had my second weekly meeting with John this morning, and he seems happy with what I have been doing
* He suggested that instead of trying to insert directly into the ASTs for somethings I could just insert keywords into files and then just do a text replace on these keywords with code blocks or something similar
* This is a good idea as I think it will allow me to more easily insert into files, however I am still going to work on inserting code through the use of ASTs as I think that will also be necessary for transforming functions.
* Later this evening I have been able to insert into the startup function of an electron app
* I have to admit I struggled with locating the correct node and where abouts to insert the code
* I realised I was over complicating the process.
* For now I am just using a lot of comparisons and manually checking the children of nodes, however I think a search function could greatly help
* Anyway, I was able to insert a console.log into the start up method, but as it is just inserting an AST into the body of a function I was also able insert a loop to see if it worked and it did
* I have also created a nice function to convert any string of JavaScript into its AST representation, this means I don’t have to form AST objects from scratch
* Overall I am happy with the progress that has been made
* Tomorrow I want to try and see if making a search function is possible, and then if I have time see about parsing some python and seeing if I can insert it into the JavaScript

# 25/01/23

* I started off by thinking about a how a search function could be implemented, I tried a few ways but after experimenting and giving it a bit more thought I don’t think creating a generic search function is really needed as I will just be inserting into positions that will never change
* I am instead working on inserting code from an abstract function called start\_up in my python “app” into the start up method of an electron app
* After some hours of work I have decided to use a different approach than I originally intended, I was originally going to parse python into ESTree pick out the key functions I defined and then insert this into the correct place in in the AST of the electron app js file
* This worked fine but the issue was too many python stdlib functions were missing when I was using the parser I was using
* I did some googling and discovered Rapyscript
* RapydScript is just python but it is turned into JavaScript and allows JavaScript to be written in python, so I think this will help with DOM manipulation later
* So now what I am doing is writing the app in RapydScript which is basically just python, this is then transpiled into JavaScript, then parsed into an AST
* The same parsing of the python AST I was doing earlier worked with this, but it works better with python stdlib functions
* However I am not able to get the python stdlib functions working in JavaScript yet, the RapydScript compiler says just to do import stdlib and it will include the correct lib, this work when compiling on the command line, but for some unknown reason it does not work when using the compiler as a library
* This is not a huge deal as with RapydScript you can just use JavaScript stdlib functions but that defeats the purpose of writing the app in python
* I will take another look tomorrow

# 26/01/23

* I started off the day pretty frustrated that I couldn’t get the RapydScript stdlib to work for the python built in functions, as this sort of defeated the purpose of using python if I couldn’t use the built in functions, at that stage you are just writing JavaScript with pythonic syntax
* I looked over the RapydScript github repository again to try and make sense of how the standard library functions are to be used
* Doing import stdlib as mentioned yesterday was not working with the compiler included in the library
* I have been trying to get this to work for about an hour now, I will come back to it later
* Ok some time later and I have made some progress
* The documentation for RapydScript is not really documentation but rather the creator showing off there creation
* It tells me to include the stdlib.js but this file just doesn’t exist anywhere
* I have found a file called baselib.js which seems to contain function definitions for what I need, I have tried to include it into my electron index.js but it does not have imports set up
* So what I have done is ripped the baselib.js file from the github repository from RapydScript and added it to my project, then when I run the program over the abstract app the baselib.js is converted into an AST and is quite literally bolted on to the end of the index.js file
* This works for now and allows me to use python functions in JavaScript, however I don’t know how elegant this is, but I don’t want to spend all my time trying to get this to work so it will serve it purpose for now
* I have realised that my how to guide is due next Friday so I will need to start and plan what I am going to be writing about
* I just created a github and pushed my code, I was getting paranoid I would lose my code if my laptop died, the repository is bare for now I will have to come back and tidy it up later

# 27/01/23

* I was busy with lectures today but I still managed to get a bit done
* I started thinking about my how to guide, and more specifically how to create it as a webpage
* I was originally going to use an online tool but, then I thought it would be more fun to do it in HTML myself, but obviously I am not going to write all the CSS myself
* So I am using Bootstrap with HTML as I have a bit of experience using that before and it will allow me to have a simple but professional looking website
* I spent about two hours in total creating a template that includes images and code snippets so when I come to write my guide it will be much easier to and I can even use it for the blog post later, etc
* I also decided on the theme of my guide, it will be how to manipulate JavaScript ASTs
* I think this is a good area because there is not many good tutorials showing how to use ASTs from start to finish. I will start with a section on what are ASTs, then a section showing how to parse existing JavaScript using acorn into an AST and show how to turn it into JSON. I then want to explain some basic nodes. Then show how to write a traversal algorithm to go through and change some characteristic of certain nodes and then finally describe how to turn this new AST back into source code again.
* I didn’t get to work on the actual project today, but I hope to be able to do some more work on it tomorrow as well as on the how to guide

# 29/01/23

* Today I continued on with my How To Guide, I decided to add a heading called introduction to the guide, I was able to write a good introduction I think and also completed the what is an AST section.
* To avoid stupid spelling mistakes and grammar errors I am writing the paragraphs in word first then copying them across to the html, I think spending the time to create a basic HTML template really helped as I am just able to paste the text into the HTML and not have to mess around with trying to position things.
* I added a few code snippets showing how some basic JavaScript can be turned into an AST, I then explained the concepts of nodes and walked through a tree explaining the various different nodes within it.
* I am using images in it that are not my own but I have linked to where I got them, I think this will be ok but I will check with John at the next meeting
* I however haven’t been able to work on with the core project as I have started too late today, I think it may be wise to focus on the how to guide for the next few days then go back to the core project after
* I was paranoid about losing the files for the how to guide so I made another gihtub repo for the deliverables that I will just keep private

# 30/01/23

* I was able to get some progress done with the how to guide today
* I was able to complete the section on parsing code into an AST
* I am trying to keep the guide as simple as possible as if I was reading it for the first time trying to understand ASTs and how to use them
* I have added more code snippets which I think makes it really clear what to do and most code tutorials I have ever done have included good code examples and snippets to make it clearer.
* I am aiming to have the guide finished for Thursday, as my other modules are starting to get more intensive, I want to leave enough time for unforeseen circumstances

# Week 4

# 31/01/23

* I had my third weekly meeting with John today
* He seemed pleased with the progress
* The next step with the project seems to be getting a basic button working in the abstract app, but that will wait till next week until after I have finished the how to guide
* He pointed out that in my diary I should be talking about any materials that I am using in my how to guide that are not my own
* I haven’t been doing that enough I think, so I will try and do that now
* I found a good photo on Wikipedia of an AST that I am using but I have added a link to the page I took it from, though it is in the public domain this is not needed but I have done it anyway
* For my main section on manipulating ASTs I am basing my algorithm on one I found in an article, however the article was very hard to understand and in my opinion overcomplicated it, so I am creating a guide on how to create the algorithm but explaining it better in my opinion and keeping it simpler
* I have created the code I want to show in the code beforehand and checked it works well, and I have started writing the guide
* The part about parsing and compiling to and from source code I have created those parts myself, using the documentation on the appropriate githubs, I have linked to the parser and will link to the compiler when I get to that part
* Today I continued on with the part about manipulating the ASTs, I found a good GIF on Wikipedia again showing how a depth first search works, as that is what the algorithm is, this image requires accreditation, so I have linked appropriately to the original of it
* Tomorrow I wish to continue on with the manipulation section of the guide

# 01/02/23

* I started the day by continuing with the how to guide
* I finished the manipulation section of the guide
* I decided to add a small example showing how to manipulate a simple function with the traversal algorithm I created, this differs from the tutorial I had been following originally as it doesn’t go into that much detail, I also feel like I have explained the code better than the original tutorial I was following.
* When I finished the manipulation section I was debating whether to continue on tonight or wait till tomorrow
* I opted to just push ahead and finish the guide today as I usually don’t get much time on a Thursday or Friday to work on this
* I originally intended to just do one section on converting back to source code but I changed my mind and decided to add another final section just showing a full program
* For my conversion back to source code I use the estree-util-to-js library, this section of the guide I created entirely myself, just using the github documentation for this library
* I show how to use it turn the modified AST back into source code
* The final section is really just a combination of all the sections showing a full program than can be copied and run
* I also had an issue with the code block and resizing but I seem to have fixed that
* I think I have finished the guide, but I will review it tomorrow and possibly try and show it to someone else to get their opinion on the looks

# 02/02/23

* As expected I didn’t have that much time today
* I gave the entire guide a review and fixed small spelling mistakes I missed
* I added a picture of some code output which I hadn’t originally, but I thought it might be a good idea
* I added a small section on the uses of ASTs in a bullet point form
* I also added a conclusion giving my final thoughts and suggested what the reader could move onto next, specifically I suggested they look into writing a transpilers as ASTs play a big role in that, I then linked to a good article talking about them
* I have also added a very simple contents section to the top of the page, in case readers where just interested in a certain part they could more easily just jump to that
* I was thinking about changing the styling such as the background look but I think the minimalist look with the white background and dark code blocks looks good as it helps to distinguish between the code and the normal text
* I think the guide is done, I was unable to show it to anyone, it is a bit late now
* I will give it another quick check over tomorrow before submitting it but I don’t plan on adding anything else

# 03/02/23

* I just finished giving the guide a quick check over before I submit it and all seems to be good, all I need to do now is to upload the guide and the diary to canvas.

# 05/02/23

* I sat down today to do some work on the actual project
* I was thinking about how to implement buttons, but I realised I forgot to implement anything outside of the start\_up function being included in the main js file of the electron app
* This was not an issue to get this to work as my understanding of ASTs has greatly increased
* However the rapydscript stdlib is still causing problems
* As a quick hack last week I just stuffed the baselib.js file I found on the git hub to the end of my electron app js file
* However I realised today that when compile the pyj into javascript, Rapydscript actually appends only parts of the stdlib to the front of the python, file
* Yet again the documentation fails to mention this, and worse the fact that it only appends part means it is hard to get to work
* But worse than that it has some of the python functions hidden away in some pyj files that it won’t append
* That was a bit of a rant about the terrible design and documentation of this library
* However to make sure these external libraries work I have copied them to my project directory and manually compile them to javascript then put them into the electron file directly
* After doing this it appears that more python functions work, such as the .append() function for lists
* However my issue is not getting these libraries to reliably be placed in the electron file, as they are very hard to access within the ASTs, because they are nested in unnecessary “ExpressionStatements”
* I will take another look tomorrow, but I think I can compile the stdlib.pyj file by itself and then just include this in the electron app js file

# 06/02/22

* So I went back to the demonic rapydscript, trying to get it to correctly generate its own libraries
* I tried making a script to convert the stdlib.pyj into JavaScript and seeing what that got, it worked and I got a library I could use but when actually using it in a electron app it just wouldn’t work
* I think this is partially just because it old and not maintained anymore
* I am considering just swapping and using JavaScript as the abstract app language
* So after some research I have found an alternate library called JavaScripthon
* I had originally looked at this when looking for something lime rapydscript but I didn’t consider it enough because it was written in python and not JavaScript
* So this is just a transpiler from python to JavaScript with no stupid extra libraries, it does seem to lack some python standard library functions, but it covers more core ones such as lists, which I just could get to work with rapydscript
* However this library is only available on the command line or through a python script
* So after some experimenting and seeing what is produced I have been able to get good clean JavaScript
* What I have done is created a python script that will transpile the app and print it out to the console
* Using another JavaScript library called PythonScript I can run this from my JavaScript and collect the transpiled code
* I can now just put this into my existing AST code and it now works to convert to the app to JavaScript much more reliably than with rapydscript
* Now that this is working I can start thinking about buttons
* I have also just noticed how much of a mess my code is at the minute, so I will end the day by trying to tidy it up a bit

# Week 5

* I had another weekly meeting today
* A few things were brought up, John mentioned I should possibly look into getting capacitor up and running and trying to get the basic functionality working with it, I don’t think that will be too difficult as I know how to more easily insert code and convert it, it is just a matter of locating the correct methods to insert into
* He also mentioned that trying to get CRUD (create, read, update, delete) functionality, this will mean that it will provide some sort of usable thing for other people
* Finally it was talked about how to actually create the front end, he suggested using webflow to create the html and css and then having some sort of naming conventions to relate to button presses etc.
* He suggested to take a look at drapcode as it produces source code for its no code tools, to see how it creates it
* I have looked at drapcode before and do recall that you need to pay to get the source code, but I will take another look tomorrow
* In the meantime I wanted to try and get buttons defined in some sort of manner in my app, as without this lower level plumbing all the stuff talked about day won’t really make sense on its own
* I did a bit of research into how electron handles buttons
* I found that buttons and the button functions are handled in the html scripts just like a web app
* But to send data back to the electron app itself you have to use their inter process communication library, this seems to work similar to opening web-sockets on a webserver and client
* I was able to then get a simple synchronous example working in electron, they offer asynchronous as well, but I feel to keep the tool simple at the start I will just stick with synchronous methods
* So now that I understand how basic button presses work I need to think about how they will be implemented in my “abstract app”
* I didn’t get to do as much as I would like to have done today as my other modules are starting to get quite busy with tests next week

# 08/02/23

* Before I jumped into any coding today I wanted to investigate how buttons are handled in electron, as I said above electron apps can handle button presses within html just like a normal client side app, but it can also send data to the “back-end” running on node.js
* I thought if capacitor is to be used that I should look into how this works before coding any buttons
* Capacitor is more just a runtime for mobile apps for client side web apps
* My initial thinking was having all button presses go to backend of electron, but if I did this it would make it very confusing and complicated to try and get this to run on a client side app, thus not making it easy for capacitor to run it
* So my thinking at the minute is to just have the electron apps all created client side, this allows me to more easily port it to capacitor in the future
* I think this is the right choice as having the tool work reliably for multiple platforms is probably better than having it work on just one platform
* I did some more digging and found someone who has ported the web app version of 2048 the game to electron, from looking at their code, the electron backend is very minimal, and all the work seems to be done in the front end html javascript
* So I think my goal is to try and get client side buttons working with electron, in my app
* My biggest problem is I need a way to specify what functions relate to what buttons
* One solution is to have a prefix on the end of function names, such as getData\_btnClick\_idOfBtn, this could work I think as it is just a matter of locating functions that have this name structure within the AST
* Another solution is to use JSON to define what methods relate to what button ids
* I personally think the first approach would be simpler for someone using this tool
* It means they can get a button working quickly without the need for knowledge of JSON
* For the HTML as John suggested yesterday I think using something such as webflow that exports its HTML will work as the user will just have to ensure their ids match
* For now I will just write my own HTML, to get the buttons to work
* So an hour or so later I have been able to get events added to buttons using my system of defining the event and id in the function definition
* To differentiate between normal functions and event functions I am using regex to split the names
* Ok the fact that I have got this working now means I can work close to CRUD
* I think I will look over this again tomorrow as I am surprised how easy it was to implement the buttons, things don’t usually work within the first couple of tries, maybe all the heavy lifting I did in the first couple of weeks have helped

# 09/02/23

* I started off by trying to add events to other DOM elements, I did this by adding a click event to an input tag and just getting it to print out to the console that it had been clicked
* So I think that I can now add events to any DOM element using my system of methodName\_event\_idOfElement
* For now it is very limited, as I can’t get any data from the DOM or send any data to it, so I think that is the next priority
* The generated code needs to be human readable, so if someone using the tool is looking at it they can easily understand what is happening
* At current the compiler I am using to go from AST to source code doesn’t include any spacing and there is no way have comments or spaces within an AST
* So I spent an hour or two trying to get comments to be inserted into the generated JavaScript so that it adds spacing and allows the user to see what is going on
* I had some issues with new lines but I think it is working to an acceptable level for now
* At the minute comments can only be added through the tool and putting comments in the python code will be ignored at current
* I am not sure if having the ability to keep the comments from the python to JavaScript is a necessary feature
* To make it work I would need to replace comments with some special characters that could be read by the AST parser, it might be a lot of work for little gain so I might leave it for now and just have comments inserted through the tool
* I was not sure if doing the events my way of names in the function definitions would be ok, but after looking at JavaScripthon again they seem to do something similar by specific special functions that can be defined to do certain things

# 10/02/23

* Didn’t get a chance to work on the project further today as I have sort of neglecting my other modules and have tests next week, hopefully I will be able to work further tomorrow or the next day

# 12/02/23

* Before going any further with my way of defining events with the function names, I wanted to look into drapcode again as John Suggested
* After some investigation as I had already known the code export ability is a paid feature of drapcode
* I looked if they offered a student plan or something similar that was free but they do not seem to, the only free plan allows you to create an app but it must be hosted om their cloud and there is no way to export the code
* So I started to look into webflow for the html definition and how it creates its code
* However it requires a paid subscription, I can get a paid subscription using a student account
* After some consideration I was questioning why I actually need to use a specific website builder, as I am going to be using the ids of the elements to assign them events, so surely the user can just create the html in any way they like
* I did come across a tool called grapejs, which seems to be a framework for building your own html drag and drop editors, I tried to get a simple editor up and running, but I couldn’t get past using the templates they provide, as it is very involved to create a working editor
* I think the best course of action at the minute is to just try and get data from html such as from inputs
* I think using a method similar for the events will be good such as getValue\_attribute\_idOfElement or similar

# 13/02/23

* I started today wanting to make some progress
* It was still bothering me that I didn’t have the ability to pass comment from the python file to the source code, as I think this is a core feature of such a tool
* I initially thought that it would be a lot of work to get this to work
* I thought about it again and realised I can just do something similar to the event functions, but this time just using a function I am using \_\_COMMENT\_\_(comment)
* There is some limitations in that they can only be placed in the bodies of functions, and placing them anywhere else I don’t think will work
* It is a good place to stop the minute
* I wanted to try and move further to CRUD so getting data from an input box is important
* I am initially just going to try and get text from an input tag in HTML
* I am using a method with a name of getValue\_idOfElement, this method will not actually be run, all this is, is for the tool to be able to locate it and replace it with a document.getElementById(idOfElement).value
* I have got this partially working, in that it will replace the function name correctly if it is just an expression, I can’t assign it to a variable yet, or have it as a parameter, but I think I know how to get that to work

# Week 6

# 14/02/23

* I wanted to try and get the accessing of input working
* I went back to it again and looked in more detail again, it turns out that all is needed for the document.getElementById(“ID”).value is a root member expression, if I can change all nodes that use my predefined pattern then it should work, as previously what I have been doing is setting the nodes to an Expression Statement, this means on compilation it will be treated as a single line statement and add a semicolon, this can cause syntax errors in the JavaScript if the value is passed as a parameter
* I fixed this issue by just extracting the member expression statement from the expression statement, assigning this member expression to the correct nodes
* I had issues at first as I was not assigning all the correct keys to the node object, but once I figured this out it appears to be working
* At this stage I am still limited to outputting to the console, but it does appear that entering data into an input allows it to be passed to the generated JavaScript
* To test this I created a very simple Celsius to Fahrenheit calculator using the input and buttons, it appears to work correctly
* During this I added some comments using my system from earlier, but found that having strings with concatenation in them cause issues, so for now I have just marked this as a syntax error, the system will throw an exception if it finds a comment block with the wrong format, informing the user what the issue is
* I think next is to try and get update working, I will need to do some research as how to do this, I have found now that the issue is not getting the source code to generate but to define all the methods for accessing elements in an abstract way so that it can be adapted for different systems

# 16/02/23

* I was quite tight for time today as I have a big test tomorrow, so I didn’t actually get to do any coding
* I had previously just put the creation of HTML to the side for a while whilst I got CRUD working
* John emailed me some links to HTML generators, which I had a look at, the main problem I see is the way the tool works at the minute is it just uses the IDs of HTML elements to assign the events and get data from them using relevant JavaScript
* It was suggested that I try and just adapt the existing editors rather than creating my own UI editor
* John also sent over a link with projects that seem to be similar to what I am trying to do
* I found a good one in which someone has converted webflow sites into a react app
* I wasn’t aware but you can add attributes to some webflow components, it seems all they do is give an attribute to certain components and set the name as the value, this allows them to manipulate certain components
* I can see how this might work for me
* To keep it simpler I could just use the name attribute and have the type of the element in there, this would mean I could more easily control what events can be assigned to each element, it also means that different elements could be ignored
* I will need to think about the idea a bit more
* Tomorrow my tests are over for a while so I can put some more time into this
* I think I will try and experiment with webflow a bit more as I was a bit too fast to dismiss it when I hit the paywall.

# 17/02/23

* So I believe I made some good progress today
* I looked into webflow further and got the student free plan working, I thought this provided the ability to export code, but it does not
* I was starting to lose motivation at this point as I did not know what to do
* I then realised that you can publish your webflow site, and this gives access to all the HTML
* I then just used the fetch API in JavaScript to get the HTML from this
* I also realised that webflow does allow you to add IDs to elements, so my current system of adding events and getting data will work through webflow, I am not sure about other platforms, but for now I will just use webflow
* I was just trying to get a simple button and input field to work with the example temperature converter I have already written
* I then hit another roadblock, webflow only allows input fields to be part of a form, this is obviously not ideal
* I then discovered that it is just a custom webflow class that controls the forms, so I think if users just add a custom attribute called fake-form, I can parse the HTML and convert it to just a normal Input
* For the next day I want to try and
* Get the HTML of a webflow app integrated into my current app
* Get the temperature converter working with the webflow HTML
* I was too quick to dismiss webflow and these other low code, no code platforms as I have experiences with them being too restrictive and am used to creating things myself for understanding, but I think for a tool like this for someone with less coding knowledge I can see it being useful

# 19/02/23

* Today I was able to get the HTML from my webflow site into my electron app, using the fetch api in JavaScript, as you can enter ids to elements in webflow, I was able to use the approach of using the ids in the python abstract app
* The issue I outlined last day about inputs only being able to be used in forms was still present
* This is not ideal as you don’t want to have a from every time you want a single input element
* I did some research and found that in older versions of webflow it was possible to have single inputs, but that was removed
* My thinking as outlined last day was to add a custom attribute to a form element and then parse the HTML and do some rearranging
* After some further research I found that just removing a single class from a webflow div that contains a form will stop it from using the default webflow code
* It still is a form, but no data will be sent to webflow
* All I have done is just parsed the HTML and look for a custom attribute I created called fake-form, any div that has this I will remove the webflow class from
* After doing this I was able to get a pretty good app working
* I still only have output to the console for now, but I will need to work on changing some sort of element in HTML to show output, but I will need to think about it a bit more

# 20/02/23

* Before moving on to trying to get output working, I noticed that my input tags are still acting as forms, such as when enter is pressed on them, the form will post and the page is reloaded, this is not ideal
* I found online that to fix this I just need to add onsubmit=”return false” and this will stop the form from submitting
* I did this so that any form elements with a fake-form div will have this attribute added
* All I did was using the same node-html-parser library I started using I was able to just locate all the form elements and add the appropriate attribute
* After this I also added some error checking to the HTML fetch which I had forgotten about earlier
* I was also thinking how the command is quite clunky to use at the minute, you have to enter the name of the file, the name of the app, the type of project to create and the web address of the published webflow site
* I will need to think about how to make this simpler, as at the minute it would be quite complex for someone else to use, as one of the main goals is to make developing an app easier and simpler

# Week 7

# 22/02/23

* I had my next weekly meeting yesterday, but didn’t get a chance to note down what was talked about
* John seemed to be happy with the progress and basically we just discussed moving towards getting a CRUD system working, I suggested what I was thinking about just using DIV elements, he pointed out that I should not try to generalise it yet and just keep it really specific at the beginning to get it to work
* So today I started investigating how webflow handles text, there appears to be something called a text block, but when looking at the HTML it produces it is literally just a DIV with text in it
* I also created a quick markdown file containing some information about the current methods/commands that can be used in the python file to access HTML elements, I have only covered things that I have verified that they work
* I was looking through some code generated through webflow and saw that they have the date and time that the code was generated at, at the top of the file, so using my comments system I added that in about 5 minutes just so I can know when the JavaScript I am looking at was last created
* I gave some through to the output and it seems that most elements allow you to access the inner text attribute and change it, so I think the best thing to do is keep the system I have using the ids elements and just have a method called setOutput\_id(data)
* I will just look for this and set the inner text of the element to the data supplied as the argument

# 23/02/23

* Ok so I was able to get a very basic out system working
* All I did was changes the innerText property of an element
* I access the elements by doing setOutput\_id(data) where the data is the string to set the innerText attribute to
* I changed my approach with this and decided to form the node myself, this allowed me more control over what the data could be
* This is in contrast to what I have already done which is create a code string and parse it into an AST then place it in the correct place in the parent tree
* Now that I have a greater understanding of the ASTs I can form them myself more easily
* The output is very limited at the minute but I will need to test it a bit more to find its limitations
* I think tomorrow I will make a start on the blog post due next week, I will probably just use Bootstrap again to keep the site have a similar feel to my last How To Guide

# 24/02/23

* I made a start today on the blog post
* I am using bootstrap again to make it, as I have a working template for it, and this will give my sites a consistent feel
* I thought about what I will need to include to allow someone else to be able to understand my project within a week
* I think I will first have a section talking about what the project is in a very high level
* I will then have a section talking about the motivation of the project, this will include why someone would want to use it and who it is targeted at
* Explaining this should show people what they are looking for if they decide to extend the source
* I will then have a section on how to download the source from my github and run some examples
* I will then have a section talking about how it works, and possibly how to use it, but as I have already written a simple guide on my github I can probably just link to that, but I will consider it more when I get there
* Next I will have a section that will aim to explain to other developers how they can extend the project if they wish, I think this will include functionality that needs to be added, and what frameworks should be targeted next
* I started today and did a first draft of the what is this project and the motivation sections
* After looking over my source code I realised it was such as mess and only I can understand it, so I took some time and split the source into multiple files to make it easier to understand to an external observers
* I then took some time to create a very barebones README.md on my github, this will be a very slimed down version of my blog post, I think.
* Next day I will get started on the how to use this section

# 26/02/23

* I was able to get a lot done today
* I wanted to write a section in my blog about how to compile the example temperature converter app I have written in my abstract syntax
* But before I did this I realised that it was very convoluted trying to run the tool as it is
* So I spent some time creating a new system for running the commands
* I am using the yargs command library to more easily control the commands
* I have limited it to two commands gen-web-app which just creates the HTML and JavaScript on their own and gen-electron-app which takes the generated code and slots into an electron app, the electron app is not compiled at this stage but I will look to do that in the future, for now users just have to run the app manually
* Once I finished this I moved on to creating a section on prerequisites, which includes downloading python, node and installing a python library
* I then have a section explaining how to clone the project from my github and install the necessary node packages
* I then have a section showing how to run each commands and to show what is to be expected
* I think taking the time do this is good for people wanting to use my tool, as many other blogs/guides I see out there don’t explain things in a lot of detail, which can put off people, especially beginners
* I want tomorrow to create a section under the try it yourself heading about how to recreate the temperature converter from scratch and add some extra features, this will show people how to use webflow basics and also how to understand my framework

# 27/02/23

* I was able to finish my try it yourself section today and I think I am ready to move onto the how it works section explaining how the tool works and why I have done what I have done, I think this will be quite a big section and will probably take tomorrow and the next day
* I did what I said and added a section explaining how to recreate the temperature app from the example, this includes how to use webflow and setup a basic website
* I have linked to my METHODS.md file on my git hub as there is too much to cover in the blog post and this explains how to use the things that I didn’t cover and will contain any future things added
* Feedback I received on the first how to guide was to put more emphasis on why someone would want to use my product. I have tried to include this in my motivation section, as well as saying who it is targeted at, which is anyone who wants to make apps quicker and people who don’t want to learn multiple frameworks.
* I have tried to link to relevant guides such as one on python and JavaScript if the reader is not familiar with these
* Tomorrow I want to start how it works section to explain how the tool works in the code, there is too much code to cover everything so I will just identify key methods that are used and explain how the overall thing works

# Week 8

# 28/02/23

* I had my weekly meeting today
* John talked about possibly using firebase to get some sort of database working, I will look into this but the priority at the minute is getting the blog post
* I was able to get the how it works section 80% finished, I have chosen to explain in a high level how the thing works, and only really show specific bits of code in the custom parser of the AST of the python app. My code is almost 800 lines long, it is not feasible to explain every bit of it. So I have just chosen what I believe is the most important parts to be looked at if someone wants to understand and contribute to the project
* I have linked my GitHub so if people want to go and dig deeper into the code they can
* I realised that my code is not very well commented so I spent some time going through and commenting bits I had forgot about. I commented in line with JSDocs so that vscode and other IDEs can understand them. This makes it easier for someone looking at the source code to understand it better. I still have to do a section on the fake-form attribute.
* After that I will do a section explaining how someone could add to the project. I think I will have a current list of limitations and a list of things which should be added first, obviously this is a static page that is relevant to the time of writing so I will copy the limitations and things which should be added to my github, that way I can update it as time goes on

# 01/03/23

* I started by finishing the how it works section, this included a section how the HTML is processed
* I included a small code example but found it better to just link to the appropriate file in my github as the code is quite large.
* I have tried to be as clear as possible when explaining how the tool works
* I decided to just push on to try and finish the blog post today, this includes the how to contribute section. I basically explained the philosophy someone should follow if they want to add features and the goals for the project. I did this so that people will not stray too far from what the project is meant to be when adding features
* I have done what I said and added list of limitations, stating what I think is acceptable and what needs to be addressed for the tool to be successful. This should allow people understand what features need to be added.
* I have also added a section with some guidelines to follow when developing the tool, these include some important libraries that need to be used and shouldn’t be changed, these include acorn, estree-to-js-util and yargs, all these libraries are probably what someone would encounter if they were developing
* I have also stated that it should be written in ES6 JavaScript as that is the modern standard now
* I have done this so when people are developing they can understand the code base. It also means the code base won’t become like the wild west with people using whatever library they want.
* I have a full list of the libraries used in my github which I have linked to
* I have also made my github public as I have links in my blog linking to it
* I have finished the blog post for now but I will review it tomorrow and then get it uploaded on Friday.

# 02/03/23

* Ok so I got the blog post finished
* I read over the whole thing and made some minor changes, and fixed any stupid spelling mistakes that I missed
* When making some modifications I just tried to clarify some things I thought were a bit confusing
* I also made a change to my How To Guide, some of the feedback from the first time I created it was that I should consider other guides and websites to see what they do well and what people like.
* One of the major concerns I had with my How to Guide the first time was people had to download node.JS and try and get it working, this could be hard especially if they have never done it before, or perhaps they were looking at it on a device such as a tablet that doesn’t allow them to download node.js
* After looking at some other sites in particular w3schools.com which I personally use quite a bit myself they have runnable code examples built in to the site
* I think this is a good idea to have, so I look for a solution
* I end up using code pen
* This is really designed for showing off websites but I just write a small piece of HTML to show the same as the console output of the code before and after manipulation.
* The reason I use code pen is because it is free and allows me to embed the example straight into my website. I have turned on code edit so people can edit the code and see their changes all without having to download Node.JS
* I have also added a small section explaining that people don’t have to download Node.JS and can just use my code pen, but can still use Node.JS if they want
* I personally think this is a huge addition as it means the guide becomes more accessible
* In order to get some feedback I asked my Brother to take a look at the blog post and the updated How To Guide
* He said that he was able to understand the goal and what the tool is supposed to do from the first couple of paragraphs. He didn’t have much time so couldn’t follow the try it yourself section
* I also asked what he thought about the improvements to the How To Guide and he said that being able to run the code on the browser is useful.
* Tomorrow I will give the whole thing a quick check over then submit it.

# 03/03/23

* I am just going to give everything a quick check over then submit it
* For my source code I have just cloned it from my GitHub repo so it won’t have any of the libraries downloaded, I explain in my blog post how to download the libraries

# 06/03/23

* I started by fixing some of the limitations I outlined in my blog. I wanted to be able to use any python file not just App.py and also to use local HTML as well as webflow, merely for testing purposes
* In order to allow any python file to be used as the App I just had to create a custom object and pass it to Python-Shell to use as an argument. I had to add another required argument of app\_path to allow this to work
* To allow local HTML, for now I am just trying to get from the supplied address, if that fails it will try and load a local file from the same address. It works for now but might not be the best solution
* I finally wanted to start and look into databases, as suggested by John last week I investigated firebase
* It seems to be more than a database, but a whole backend for apps
* I watched a few videos explaining it
* Its main attraction seems to be that apps can be updated in real time from a database as it uses a no SQL database
* I will need to look into it further, I will try and get an app up and running just in plain javascript, then look at what I can abstract to my app framework

# Week 9

# 08/03/23

* I had my weekly meeting yesterday
* We spoke about the next steps which is getting some sort of database integration, as I talked about in Monday I already looked into firebase, but it had a number of issues such as having to create an account and a paywall past a certain point
* I didn’t think it would be ideal
* I conveyed this to John and he suggested I look at MongoDB
* So I looked into it a bit more today
* It is a no-SQL database which seems to use JSON to store its data
* It does have a free cloud hosting systems similar to firebase but it also allows to you to download MongoDB and host it locally, this means it is much easier to set up
* I downloaded it and discovered it also comes with a program called Compass, which seems to allow you to edit and view the records in the database, it is very useful and easy to use
* I just set up a simple database and got connected to it and played around with some sample data
* I just used Node.js to connect to it
* It is incredibly easy to use
* I decided to use mongoose instead of the official MongoDB library as that is what people online say it better to use
* I think I will need to implement some backend python app also, because doing database queries in the frontend is probably not a good idea
* I already have the electron backend working
* But for the web app I will need a solution, I am thinking express.js but I am not sure and will need to research it a bit more

# 09/03/23

* I had originally wanted to try and start thinking about how to get he backend app working and the mongodb working but I didn’t have enough time to start that today
* Instead I tried to get the app working as an npm package so people don’t have to clone the entire repo from git hub to get it working
* I got it working so that you can just type abstract-app now and you don’t have to do in the repo directory. This allows me to test the tool easier as it means I don’t clutter up the working directory with generated files.
* I will need to figure out how to make the package public, I think it involves making an npm account and putting the code up on the website

# 10/03/23

* I didn’t have much time today, but I took some time to think about how the connection with the database will work
* I originally wanted to have a file for the backend and a file for the frontend by I think that will be troublesome and over complicated for what I need
* I think the best way to do it is to just have the database functions also in the normal app and just create the backend without the user knowing, this way the whole process is just easier for the user
* I will try and get started on this tomorrow

# 12/03/23

* I looked into getting mongodb working
* I originally wanted to use mongoose to connect to mongodb but I think the library from mongodb is better as it will be easier to abstract
* I have started with electron because a backend already exists for it
* I have it so when someone types connectDB(address, dbName) it will add the appropriate code at the top of an electron app to connect
* This took longer than expected due to having to use asynchronous which some of the parsers don’t like
* I need to think about how to pass data back and forth from the front and backends
* I think using ipc protocol included with electron will work but I will need to look into it further

# 13/03/23

* I started today by trying to get some ipc code generating on the backend of an electron app, I was able to do so easily as I had created some example code a few weeks ago
* I went with a naming scheme of DBLoadData(“collection name”, searchDict), this will load data from the collection in the database and filter it by the contents of the searchDict
* This was working but for some reason when using this method name, javascripthon thought it was a class and tried to make it such in javascript
* I just changed the method name to dbLoadData instead and it worked, I think what was happening was as there is no method definition for DBLoadData it just thinks it is a class, but I will need to look into this further
* The backend ipc code was not to hard to get working
* I also got the front end ipc code working in some sort of manner, it just connects to an ipc on the backend with a unique id which is just a counter that increments by one when a new load data is found
* I had trouble sending the object over ipc that is produced from mongodb, but a good fix I found online is to just turn the object into a json string and send that, then when it is received turn it back into an object

# Week 10

# 14/03/23

* In my weekly meeting today john suggested that I try and get mongodb to work similar to firebase such as adding listeners for when data is changed
* I did some research on this and found that mongodb does have a built in method for such a thing
* I think I just need a custom method in my python app to set up a connection between the two
* I will look into this tomorrow
* I wanted to test my load data method more thoroughly so I just made a simple app in python that takes a search term and finds a song with that name in the database, it seemed to work pretty well, there is some things that may not be normal to python, but I will note them in my METHODS.md file at some point

# 15/03/23

* I did some research today about how to detect changes made to data in a mongodb database, I found that it does contain something similar to firebase, however to run it on a local database it requires changing some options on the database launch. I did as it said but I couldn't get mine to work properly
* I also think that the changes are quite extensive for someone with little knowledge to do on their own, as I think the tool won’t be able to make such specific changes
* Another solution is to just have a timer that checks for changes manually, but this might be a bit hacky
* Instead I decided to get inserting working
* It didn’t take very long as I already have most of the ipc code working for electron
* I got it to work in a similar way to finding data, dbInsertData(“collection name”, {data}). It will return a string indicating if the insert succeeded or not

# 16/03/23

* I was going to start looking into express.js today but remembered that I still have the update and delete of CRUD to do, so I got to work on an update statement
* I first tried it out outside of electron, the way it works is it takes some JSON to select the record to update and then some different JSON to update the record to
* There is two main methods that can be used updateOne and updateMany, I am electing to use updateOne for now for the ease of use it provides, I am may go ad add support for the other method in the future
* The updateOne follows a similar structure to the other methods dbUpdateData(“collection name”, {filter data}, {data}).
* Again it just creates ipc methods when it finds the relevant method names in the front and back ends

# 19/03/23

* I was looking through the code and realised that if the app couldn’t connect to a database then it would crash
* So I added some error checking code to be generated when a database is connecting so that it doesn’t break
* I also went through and made all the current database methods return null if it cannot access it
* This way the app will still open and run but the database parts won’t break the entire system if they can’t access the data
* I think to finish off CRUD I just need to implement delete next

# 20/03/23

* I wanted today to try and add deleting of records
* I decided on the syntax dbDeleteData(“collection name”, {filter data}), like the other database methods as it is easy to understand
* I was able to get the deleting working quite quickly as it is similar to the other methods
* I decided to change the update method to be more like the delete where it returns the number of records affected, as I am using deleteOne and updateOne this will always be 1 or 0
* After this I started experimenting with some logic that will keep trying to connect to the database even if the first attempt failed and the app is open
* I got this working but when trying to use the connectionClosed event to try and reconnect if the connection is closed it would not work, for some reason I couldn't get the connectionClosed event to fire
* I will have another look tomorrow, but I am yet to implement this new logic into the tool

# Week 11

# 21/03/23

* I had my last meeting with John today
* He suggested that in order to sync up the databases that I use a “master” database on a server, then all the desktop clients will connect to that
* This is different from the idea I had of users setting up there own database, but know the tool will download it and provide it to the user in some sort of way
* He suggested that I try and make some sort of discord style app as the end product
* I think this is doable but only through trying will I be able to find out
* My initial thinking is just to use an express.js server that holds the database, this will also allow the front end web client to connect and send and receive data
* Then each electron client will have a mini database that they will use to store changes and send them to the server database
* This way data can be seen by all the connections and who it is from
* For sending data between the server and electron apps I will need to use sockets, I will use socket io as I have some experience using it and it works well on the front end when I get to that
* As I only have about 5 weeks left I will try and spend the next 2 working on the code, then get it into a stable state and spend some time doing the testing and other required parts
* I will start tomorrow to look into the server in more detail

# 22/03/23

* I started to look into having a master database
* I did some experimenting and found the best way is to have a script that connects to the database and uses sockets to communicate with clients
* I got the database script up and running quickly, it just spawns the mongodb in a separate process
* I then set up a socket using socket.io
* The electron app and web apps will send data to the sockets and it is then placed in the master mongodb
* I got some very basic sockets set up and working between electron and the database script, however it took some time, as I was making a stupid mistake when trying to receive events on the server, as my previous experience with socket.io on the server is with flask not JavaScript so I was unaware of the difference, but I managed to get it working in the end

# 23/03/23

* I continued on with my socket code today to get data from the apps to the server
* I got it working to a good state where data can be sent to the backend server for insert, update, and delete, I haven’t got load working yet as I need to think about it a bit more
* It took a bit longer than expected as I had to change ipc code slightly, but after that it is working
* I then updated the tool itself with the new code and a new template for the electron main.js to allow it work correctly
* Next I need to look into the load data as I am not sure if I should keep mongodbs for each app that sync with the main one as John suggested or if I could keep the entire database in memory when it is running and load it every-time the app starts
* The problem with this approach is if the database is very large it won’t be able to be loaded into memory, in that case the separate mongodb approach is better but I have read that syncing local and server databases is no easy task
* I will need to research it a bit more before making a decision
* I may move on to getting the server auto generated by the tool and the express.js web app set up also

# 24/03/23

* I did some thinking today before starting to implement a local database
* After looking into it would be very difficult to have multiple mongodb instances running at the same time as they would all need a different port to run on, I see no way to do this, as mongodb is not designed to be used in the way I want to use it
* I then thought about just using JSON to store the database locally, this would work, after thinking about it a bit more, the amount of code needed to make the “databases” sync up would probably take me a long time to write
* I have decided to put the local database idea to the side for now
* I instead want to focus on getting some sort of firebase like functionality working, where when a change is made the clients can be told about it
* I decided to do this just by having an event emitted to all clients when an update is made to the database, this is then received by the backend of the apps and sent via ipc back to the front end, it works well
* I have just made it so that when the socket between the backend and the server is not connected it will refuse to do any database operations, I may change this in the future to contain some sort of buffer that allows data to be queued up to be sent to the server, but for now I want to keep it simple
* I also fixed my issue of loading data just by sending an event to the server and it returns the data through the socket callback

# 27/03/23

* Today I wanted to try and get the insert and delete events working, the code for these is very similar to the code for an update event
* That did not take me as long as I expected as it was merely an exercise in copy and paste with a few changes
* So now when someone makes an update, insert or deletion from the database, the server will inform all the connected clients that one of these happened
* The app can run a callback function defined in the python app as updateChange that will run when an update is detected, similar functions exist for insert and delete
* It may be possible to pass the data that was changed to the clients, but for now I have kept it simple and not implemented that
* As this didn’t take me as long as expected I started to implement the functionality of creating the server
* I have decided to have this as a separate command for now, just for simplicity
* I have the server.js file in my project and the command simply copies it to the relevant directory supplied to the user
* It also copies the mongodb.exe to the directory
* It also contains a bat file generated by the command that allows users to more easily run the servers
* The next goal is to get an express server working for the web app to allow it to connect to the db

# Week 12

# 28/03/23

* I wanted to see if it was possible to download the mongodb.exe directly from the web in my JavaScript and send it to the server directory, I got the zip downloading but unzipping it proved to be harder than is worth, so I ended up just going back to my original idea of having the exe in my project and copying it across, this is faster and more reliable
* I then looked into getting the express.js server working
* My idea for this is to have the express.js server, this will communicate with the database server.js through socket.io
* The frontend web app will use http to send data to the express server and it will send it on to the database
* To get the change events I will just have to use socket.io on the front end
* I started to get a simple express.js server working today
* I will just need to do that then integrate it into my tool
* That will be a good place to stop I believe
* I can then make a simple discord style messaging app with my framework, this part should take a day at most as I will keep the app simple

# 30/03/23

* I was looking into getting my web app version up and working, I have gotten the code working for loading data from the database
* I was going to implement it into my tool, but the issue was as the post requests work asynchronously I can’t easily implement it into my synchronous code
* To keep the framework simple and easy to use I didn’t want to add asynchronous programming to it yet, as that would be difficult for some people
* I then considered my options and saw 3 options, run the post request synchronously, this is possible but jQuery recently removed support this as it is dangerous as it can cause applications to hang
* The other two options were to nest the callbacks of the post requests to give the impression of synchronous application, this had issues as constructing the correct callbacks would be difficult and to error prone
* The last options which I went with is to convert every function to an asynchronous function and use await with every function call
* This is the easiest to get working and will give the impression of the programming running synchronously
* I spent some time implementing this into my tool and got it working pretty quickly once I knew what I was doing
* The next step is to get update, delete and insert working, then the event listeners, this should not be too hard as I have changed the way the code is constructed on the web app to use helper functions, this means that when the dbLoadData() method is called in the python it will be directly calling a function found in the JavaScript that will be added on.

# 31/03/23

* I continued on with trying to my web app working with the databases
* I have changed my approach with this
* Instead of iterating through and changing specific methods based on there name I am just appending on the methods as JavaScript, I talked about this yesterday, but I decided to go full on with it
* This would probably have been the better way to go with the electron app also but it is too late to change it now
* Using this new approach I was very quickly able to get the insert, update and delete working
* Again these all use POST requests to the express server and it sends the data to the databases via sockets
* In order to get the change events working like in electron I have to use sockets again
* However I did not want to give the front end JavaScript direct socket access to the server that holds the database, so I decided to make another socket.io server that runs on the express server, that just emits events to the connected web clients when a change is detected
* This is a better approach in my opinion as it keeps it similar to the electron app
* I think I now have all that is needed to make a discord style messaging app, I will need to do some more testing tomorrow but once I do that I can get onto the discord style app
* After I have done that I will look into testing and getting an updated blog post etc

# 02/04/23

* I was looking into getting testing started today, so I did a bit of research about what a test plan is
* After some short research what I found was that as my testing is going to be quite small and only really covering the project as a whole, I would be doing system testing
* What I have found is that most of testing plans start with needing a specification to test against
* As I don’t really have one of those the next best thing is to use my docs that I have created
* These have not been updated in a while, so I took some time to update these as needed
* It took longer than expected update these, but I should hopefully should be able to start a test plan tomorrow
* I think the best way to do it is to just state what each test should produce, and manually check the output, automated testing would be nice but I don’t feel I have time to start diving into that
* What I am thinking is just having many different python apps, each testing different aspects, I can then create a small tool that will run these files. I can then check the output against what should be produced
* This is the easiest way, I will need to create a small API so that the tool can be used without the command line.

# 03/04/23

* I am trying to get ahead slightly as I am not available to work a lot on the project for the next 2 – 3 days.
* I started a test plan, I have decided to test in two different ways
* The first way it to run automated tests to check if the custom methods I have produce the correct code
* These can just be a list of different code files that are converted to JavaScript and the output can be checked against a string automatically
* The second part is to test the functionality of the code produced
* This will involve running manual tests, these will consist of small tests to check database operations and the manipulation of the HTML
* I will try to keep the tests small and simple to understand, so that another can more easily expand upon them
* In my test plan I intend to outline each test and give brief descriptions for the automated tests
* For the manual tests I will provide more details as to what the output should be
* It may be a bit early to start this, but I am trying to manage my time especially with me knowing I will be busy the next few days
* I will try and get as much testing done this week and next as possible
* I can then use the remaining time to complete the updated blog post, the how to guide and the social media post

# Week 13

# 04/04/23

* I got started on my tests today
* I first spent some time creating a small test program, it consists of a class to hold test data and a custom method for writing to a log, as the standard out is cluttered with debug information that I do not want to remove at current
* I had to make some small adjustments to the actual code to allow testing, I have just exposed some parts externally to allow me to test without having to use the CLI every time
* To check the output against the expected I am using the AST versions of the code created, this makes it much easier to compare them
* I can just stringify the JSON and compare them as needed
* I was able to test the event method and the start up method, I am trying to include test cases for valid and invalid cases
* As I have said in my Test Plan it is obviously impossible to test every combination, that is why I am testing the parts of the tool separately
* In my test plan I have a table listing the test cases carried out to current and the status (pass/fail)
* At current 3 tests are failing, one I don’t know why
* The other two are because of the way reg ex is used to check the events
* They are generating events when they should not be
* I will fix the bugs after all testing is complete
* I thought testing would be difficult, but it so far has been helpful as I would not have seen the bugs present without it
* I will try and get the other custom methods (excluding the database) tested tomorrow

# 05/04/23

* I was able to get all the custom methods tested today, I did it automated as expected
* Most of the test were passing, except some that were failing
* The ones that are failing all seem to have the same issue
* I think it may be a bug in one of the libraries I am using, but I will look into it after all testing is completed
* I have tried to test everything I deem necessary, such as different data types for a setOutput
* I have tried to include some tests that are wrong, such as ones that will throw an exception
* For example the setOutput will throw an exception to the user if there is no parameter passed to it, I am testing to see if the correct exception is thrown
* I will next try and move onto the testing of the html, that should only be about 5 tests in total
* I will then move onto testing the database methods, this will need to be done twice I believe as the web app and electron handle them both differently

# 06/04/23

* I didn’t get as much time as I would have liked today, but I was able to complete the testing for the comments and start the HTML testing
* For the comments I have tried like before to include tests that will generate the exceptions I have and tests that pass
* And for the HTML I have tried to include webflow and local HTML tests
* I should get the HTML tests down tomorrow then I can move on to some database tests
* I am still considering if these should be automated or not as there is a lot of different elements that need to work together
* I am still leaning towards automated tests at the moment as that would greatly benefit others, and it should not take too long to do these as I have the test harness already working
* I am aiming to have testing done by mid to end of next week, this allows me to do the updated blog post and social media post on the last week

# 07/04/23

* I continued with testing today and was able to complete tests for loading local HTML and electron dbLoadData tests
* I was going to try and automate the webflow HTML tests, but that proved more difficult than expected so I will just do these manually
* During my testing I found an error were an exception is not being generated as expected when a searchDict is not an object, I think this is an easy fix, but again I will wait till the end to fix the bugs
* Next I will do the insert tests, these test cases are very similar to the update so it is really just a matter of copy and paste these with some minor changes
* I think my original plan to get testing done by mid – end will probably be ok
* I feel as if may be doing too much testing, but in order to prove that it works well to someone else looking to use it I believe it is necessary to spend the time doing this

# 09/04/23

* I continued on with my tests today
* I was able to get all the electron tests completed today
* It did not take as long as I expected as the tests for the database methods are all practically identical, so it was just a case of copy and paste the tests and then make some changes to the input and output
* I also completed the electron change event methods, I just tested for these if the correct code is generated and then if the correct exception is thrown if there is more than one of these change methods
* I didn’t get any failures, I think this is because I wrote the database code last, and by then I had become more capable of JavaScript and more aware of the different aspects that should raise an exception
* I may be able to finish all the automatic tests tomorrow as it is just the web database tests to complete, I will just use the same tests as I used for the electron database tests, but change the input and output, so it should no take long at all

# 10/04/23

* I was able to get the majority of the web database test cases done today
* It did not take long as I expected as I was just copy and pasting them as the actual test cases were the same for electron just the output was slightly different so had to be checked, and they both use a different method so I wanted to test them separately
* I found that it was very repetitive to copy and paste and change a few numbers, so I thought I would see if I could speed up using ChatGPT
* I have been sceptical of using is so far as in my experience you spend more time trying to debug and figure out what its code does but I thought I would try it for this
* I just gave it the example test case code I had and told it to generate test cases for x – y and it did it very well, so, this sped me up a lot
* I also asked it to generate the code needed to execute the test case using an example it gave me and it was able to do it
* Using ChatGPT for this type of repetitive thing really helped speed up this pretty boring task
* Enough about the AI
* So I got all web database methods tested apart from the change events
* I kept getting the same failed test where the database web methods were not throwing an exception when they should be for incorrect syntax
* After so digging I found that I had forgot to add the validation methods to the web section of the parse database file, after adding these all the test cases were passing as expected
* Those are just 6 test cases so I can do those quickly tomorrow
* After that I think I will have finished my automated tests
* I will then write a few tests in my abstract-app framework and have these as manual tests, for example testing if the database is passing data correctly that would be too time consuming to automate
* After that I can start on the updated blog post, social media post and creating the simple discord style messaging app as John suggested
* I think the discord app could be a test also but I will think about it when I get to it

# Week 14

# 11/04/23

* I took today to finish my automated tests
* I just finished the 6 from yesterday about the change events for a web app
* These all passed as expected
* To finish up I wrote a bit about the test cases in my test plan and also explained the structure of the test input and output data
* I also added the test program to my NPM package so a user can just run npm test in the repo and all the tests will be carried out
* I also added a few sentences explaining how to use this
* Tomorrow I will make my manual test cases
* I was originally going to do manual tests to test the functionality of the html methods but I don’t think this is needed as if the correct code is generated then these will work, it is the database methods that require a bit more checking
* So for manual tests I will have one to check the server.js is created correctly
* A few to test webflow HTML downloading
* One or two to test the python app selection
* And a couple to test the database methods
* I will make these more as apps that can be compiled and used as I think that will be the best way
* In total I think I will have about 10 – 20 manual tests
* I should be able to finish this tomorrow then

# 12/04/23

* So I think I have finally finished testing
* I only ended up with 9 manual tests, as I bundled all the database testing into one app which I believed to be better
* For these I have gone with the format of describing how to carry them out and then the output that should be expected
* I have manual tests, to test the webflow HTML, I just supply the webflow address from a test site I made and then compare it to the app generated and the site when viewed through webflow
* I have two tests to check if the path for an app works correctly and if the correct error is thrown when a syntax error is found
* To test if the database methods are working correctly I just made a small song app that shows off all the different parts of the database connection to ensure it is working correctly, I believe it work correctly
* I just have a few automated failed tests to look at, but that should not take long
* I can take tomorrow to fix those bugs and also create the discord style app that John suggested, I also need to tidy up the test plan to reflect these new test cases and just explain to any future testers how they work
* If I have more time I will start to plan what I will add to the blog post and how to guide

# 13/04/23

* I was short on time yesterday but I managed to get my test plan finished
* I have explained how the test are to be run and why I have chosen certain tests as I cant test everything
* As the goal of this project is for others to expand upon it I added a section at the end explain how to add a new automated test case, it just shows a UML diagram of the TestCase class and how to use it with some examples
* I also added a few sentences on how to add a manual test but it is much simpler than an automated test case
* Tomorrow I will look into the failed test cases and create the very simple discord style messaging app

# 14/04/23

* I started by trying to fix the failed test cases
* I was able to fix two of them easily because it was just a quick change to some regex, however some keep failing, with the same error
* It seems that when an underscore is added to the end of a method name it is being removed when it shouldn’t
* This is not an error in my program but an error in one of the libraries I am using, I have left the test cases failing for now but have noted in the test cases document why they are failing
* Next I finally made the very simple discord style app, it is very basic in that you can enter your name and send messages, however this would have taken hours to write normally but using my tool I was able to get a prototype version up and running for electron and web with a database backend in about 1 hour, that includes the time it took for me to create the HTML in webflow which is not a great skill of mine
* I tested it and it works well just I can’t get a scrolling bar to appear through webflow when too many messages appear, I have left it for now as it is not a critical thing but I might be nice to have it
* I have also been planning out what I will change in the blog post
* I believe I can keep most of what I have, however I will change the try it yourself section to use this new discord app instead of the old example, in the previous blog post I showed how to use webflow but I have discovered that read only links can be shared to webflow projects
* I think a better way to do this is to share the link and just explain key parts such as the fake-form attribute, this allows me to spend more time explain the code and how the new database parts of the tool work
* I will then have a section explain how the new database code is generated
* Finally I will have a section explaining how testing works
* For my updated how to guide to keep it simple I think I will just add a section on advanced manipulation of ASTs as I know how to do that know
* I will hopefully be able to get all this done before the deadline next week, if I don’t think I will I will cut down the content slightly

# 15/04/23

* I started by trying to get the try it yourself section of the blog post updated
* I removed everything that was there and replaced it with the discord style app I have made to se as the example. I have changed the approach, in that I want it read more like a commentary of the code rather than a tutorial
* As I said yesterday I have provided a read only link to the webflow project which I used, I then just highlight key parts of it rather than explaining how to create the app in webflow, as that is not the focus of the post in my opinion
* For the code I have again just highlighted key parts rather than explain what each method does as I have already done that in greater detail in my docs that I have linked, I feel that repeating the same information again is not a good use of time
* I then showed how to create the apps and a server using the updated syntax
* Finally I added a small 15 second video showing the app working
* Just need to explain how this new database stuff works on the blog and discuss testing

# 16/04/23

* I continued on with the updates to the blog post
* I added a section talking about the database integration in general
* I talked about the server that you can generate
* I have decided to talk more about how the server code is generated rather than the code that is creates as that is more the focus of the tool
* I have linked the template for the server I wrote to allow others to view it, but as it is not part of the functioning of the tool I have not explained it
* I have rather explained the order for generating certain files and the function of the server as a whole
* Next I will talk about the database methods and how they are generated in electron and the express server

# 17/04/23

* I was able to complete the database section of my updated blog
* I have explained how the tool parses the database methods, in a similar way to how I have done it before with the original HTML methods
* I have tried to focus on explaining the most relevant parts of the code rather than every line as then I would be here all day writing this blog
* I have linked the relevant git hub files on the post as needed
* I think for the blog all I have left is a section on testing and to update the current limitations section as I have resolved some of those

# Week 15

# 18/04/23

* I continued with the blog post today and I think I may have finished it
* I need to read over it in more detail tomorrow but for now it is done
* For my final additions I have talked about how external libraries are added to the HTML for the web apps
* I also updated the limitations page with some information about the newest limitations that includes stuff to do with the database
* I lastly added more information to the tips for development section, mostly focusing on the database integration
* As John suggested in the last hand in that my post had quite a lot of spelling mistakes, I installed a spell checker onto my vscode to check this for me and eliminate such a stupid mistake. Something such as spelling mistakes can really hurt the professionalism and credibility of a piece of writing
* I also started using GitHub co-pilot, it really helped with some useful text suggestions, this sped me up quite a bit when writing and I really should have been using it before now
* Finally I wanted to check my tool works on an external machine, so on my desktop computer I don’t use for development I downloaded all the necessary tools and then installed the package through npm, I then used the discord app source in the github to test if the tool was working correctly and it does appear to be

# 19/04/23

* I started today by reading over my blog post, I made a few minor changes but nothing major, so it is finally finished
* I moved on to the updated how to guide
* I wanted to do some advanced manipulation, so I thought that showing how I do my comments being included in an AST was a good idea. It is not too complex and it did not take very long to complete
* I followed the format I had in the previous how to guide, with showing how the code is created in a step by step basis
* I also included another code pen, the same as the first example for people to run the code who don’t have access to a desktop environment
* The final thing is just the social media post, that will just require some research tomorrow

# 20/04/23

* I started by researching the social media post I will make.
* I think I will post to r/programming on reddit. This is the biggest programming subreddit and is the most general one. It has 5.4 million members and is the 38th biggest on reddit
* I thought about using a discord server, however these are too closed off in my opinion and a lot of them don’t allow self-promotion
* When posting to r/programming you can only have a title and url, to have extra information it must be added as a comment, which is what I will do
* The best thing to do is to have a catchy title, for the URL I will link my blog post as I feel it is most relevant to people
* In order to have the link accessible to the public I had to set up a github pages site for my repository, it can be found at <https://dogracer78.github.io/Abstract-App-To-Source-Code/>
* From reading similar posts, it seems the best ones use a conversational tone, when they talk about their projects. I think this is a good approach rather than having it sound like an advertisement.
* Generally most posts are not very long, and simply state, this is what I made, here is how to check it out, there is no point in trying to explain all the features, that is what the blog post is for
* Shorter is better in my opinion as sometimes if I see a large chunk of text I won’t read it very thoroughly
* I will make this post to reddit, but I will wait till my project is complete, then I will publish it and add a link to it in the post.docx
* I have also started to add the deliverables to github, they are under the folder /docs as that is the only way to keep it organised and for github pages to work
* I have written the post, I tried to keep it short and sweet, I have stated what the project is, who it is aimed at, linked the relevant blog and git hub and finally suggested that others can leave feedback or suggestions if they want. This is another thing I see a lot of posts doing

# 21/04/23

* I added any comments that I had neglected to during the development of the project
* I also updated my READMEN and docs on my github to be relevant to the most up to date version of the project
* I think the entire project is pretty much done
* I was checking the submission and I think I may need to add a section to my blog post about a roadmap as it is not that clear what needs to be added
* I can do that tomorrow

# 22/04/23

* I was checking over the project and decided to add the section an a roadmap to the blog, it did not take long, I just listed various things that I believe should be worked on next
* I also updated the npm package to the latest version, I am going to try it on my other pc later just to check I broke nothing, I was not changing any code just comments and I ran the auto tests that I know should pass were passing, so I believe it is working ok
* I was going to look over everything tomorrow but I have decided to just upload it today to get everything over with
* I did a last check on my other pc after uploading the new NPM package and everything seems to be working ok
* I will put this diary and the updated post document when I make the post, into the /docs folder on the github
* I am happy with the way the project has turned and, but am also relieved that it is over at the same time, I may continue to work on the project in my free time, but I will have to see how that turns out