CyberX

A critical look at the tech education system for girls, girls of colour, trans girls, and girls+ in the UK

Tags: Girls+ In Tech, Gender Gap In Tech, Secondary Tech Education for Young Girls 11-17

Word Count: 7950

Abstract

Here lies the journey of CyberX, from start to finish, the ups and downs, the journey, the discovery. This was a huge undertaking and I have tried my very best to create what I set out to. Gender equality is a big problem all over the world, even more so within STEM based subjects, teaching, and careers, CyberX was designed to provide a solution for the lack of engagement for young girls+ from a young age within Computer related study and work, and potentially provide outreach in the future.

Contents

Figures & Tables

Acknowledgements

It is important for me to let everyone know who guided and helped me with my university journey, how important and pivotal they were to me getting through this final part with some sort of success.

I want to thank Sarah who has been wonderful, her confidence in me has been inspiring and helped me to acknowledge my own power in completing things, I love her passion for gender equality and teaching, she will always be my main inspiration. Although Sarah is very busy, there's always time for others, I appreciate you being able to support me despite your busy schedule and I really appreciate it. Whenever I feel doubt, imposter syndrome or second guess myself, your words of wisdom and belief in me helped me to stop that negative thought cycle.

Archie Khuman for his creativity in how he sees things, his module allowed me creative freedom to create something of my own, this ultimately helped me to dive right in with my final year project, maybe a bit too bravely, but here we are.

For Jarrad being my go to guy to read over any work, his optimism is infectious and always knows the right thing to say and has a solution based attitude, Also, I never feel stupid saying what I am thinking to him when it comes to any coursework, and that's saying something.

A big thank you to my husband, Pod, for supporting me when I am crazy with stress, whether it's a piece of code that's not working, or exhaustion from trying to study and striving to achieve great marks while dealing with dyslexia, ADD and ADHD, you are always there egging me on, encouraging me, and feeding me (super important). Without your support I would not have been able to quit work so I could focus fulltime on my studies, I definitely would not have carried on going gym, but you were there, guiding me and supporting me, so thank you.

A special thank you and appreciation to my daughter, Charlie, although grown and flown the nest, is always on hand to take the pressure off, whether it's going shopping for me, or completing everyday tasks that frees me up to carry out my uni work. Thank you for being my guinea pig when I want to try something out and thank you for letting me teach you techy stuff, this really helped encourage me in continuing on. You make me very proud and I am so glad you're following in my footsteps.

I have huge gratitude for my fluffy fur baby cats, without Princess, Eilish and Hughie, I would have had no fluffy creatures to destress me and purr on me, whenever I felt frustrated, you were only too willing to provide comfort, for that, I am forever grateful, thank you.

Without all of these beings that I have in my life, I would not have been as successful as I have been with my marks, I am truly grateful for each every one of you.

1. Introduction

As a neurodivergent learner myself, I fully understand the difficulties faced in any learning environment, I have phases of hyperfocus mixed with even longer procrastination phases, with concentration running away from me, I have found on my university journey that one source of information and teaching is never enough and would require a minimum of three different sources on each topic for me to fully grasp any new concepts. With this project, I have aimed to build an online e-learning platform designed to aid neurodivergent and disadvantaged girls+ to learn in a way that covers all bases, provides additional resources and encourages learners to embrace the part of them that seems to rebel against learning new and exciting things. I have attempted to create a platform that has user and admin dashboard functionality, login/logout functionality, and a course functionality. The admin dashboard is one of the biggest features and has the ability to add, edit and delete users and courses. In this report I have detailed my journey, and what a struggle it has been! I have a true appreciation for what it takes to be a good developer, and hope to get there one day.

Figure.1.1 CyberX Logo (Canva, 2022)

2. Executive Summary

CyberX is an innovative online learning platform designed specifically for young girls+ aged 11-17, with a focus on promoting diversity and inclusion in the tech industry. Built using cutting-edge technologies such as Object Oriented PHP programming, HTML, CSS, Twig, and JavaScript, CyberX follows the Model-View-Controller (MVC) architecture and adheres to the SOLID principles, ensuring a robust and scalable platform.

The platform offers a wide range of engaging and interactive courses covering various aspects of technology, including programming, web development, cybersecurity, and data science. Each course is carefully crafted with a user-friendly interface and age-appropriate content to make learning fun and accessible for girls+ at different skill levels. CyberX emphasises hands-on learning through practical projects, coding exercises, and real-world scenarios, allowing girls+ to apply their newly acquired knowledge and skills in a supportive and inclusive environment.

The use of Object Oriented PHP programming ensures efficient code organisation and modularity, making CyberX scalable and easy to maintain. HTML and CSS are used to create responsive and visually appealing user interfaces, providing a seamless experience across different devices and screen sizes. Twig, a popular templating engine for PHP, enables efficient separation of logic and presentation, making the platform highly customisable and extensible. (Patel, 2016)

In addition, CyberX incorporates JavaScript for interactive and dynamic functionalities, enhancing the user experience and allowing for interactive learning elements. When building the platform, the SOLID principles where followed, including Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion, which promote clean and maintainable code architecture.(Madasu et al., 2015)

To conclude the summary, CyberX is a cutting-edge online learning platform that empowers young girls+ to develop their technical skills, foster creativity, and pursue their passion for technology in a supportive and inclusive environment. With its modern look and feel, and adherence to best coding practices, CyberX is poised to revolutionise online learning for girls+ and bridge the gender gap in the tech industry. (“Closing the Childhood Digital Divide,” n.d.)

3 - Development Lifecycle & Journey

For this development project I have opted to utilise SDLC with the agile methodology and have a very open and flowing journey allowing me to change anything if I need to in the first instance and will allow testing that things work at each stage in a non-restrictive way. If there is a component that is not working or I find a better and more productive and secure way of doing things, I want the freedom to change and switch if needs be.

I have found it very difficult to focus for long periods of time with the development side of this project. My working style is chaotic and I have attempted to restrain this side of my working style by implementing time structures, lists, sprints and an organisational app, none seem to work to bring a cohesion to my thought processes and it becomes overwhelming. What I have decided to do is to go with my ‘chaotic’ working style and see where it takes me, so far I have noticed that once I have past the overwhelming pit of despair, something good comes of it.

After completing Secure Web Application Development in the first term of my final year, I wanted to use the skills learnt on this module within my web application. The main components here are to ensure I follow the SOLID Principles, that my web application has a separation of concerns, and a single point of access. According to Oloruntoba (2020), Uncle Bob, as he was fondly known, real name Robert C Martin, established these five principles to lend a structured and secure way of working within an Object Oriented Design (OOD) environment (See Fig). The principles state that classes should have one stand-alone job, meaning each functionality needs to be broken down and given its own class, eg. A class for each input and a class for the output, additionally ‘objects should be open for extension but closed for modification’ Oloruntoba (2020). I must ensure that every class/sub class created is suitable for its parent class/base. Interface segregation will form a part of my design principles to ensure there’s no dependency on unused interfaces. Lastly, the dependency inversion declares ‘entities must depend on abstractions, not concretions’, these principles will help guide me to develop CyberX in a way that brings fewer complications, make it easier to test and spot bugs, and allow future collaboration.

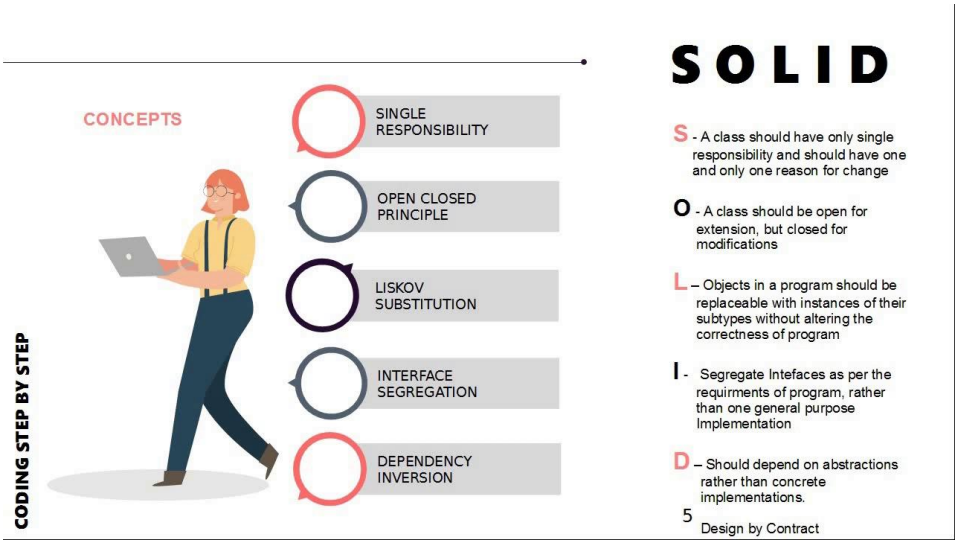


Figure.3.1. – The SOLID Principles. DXKB (2021)

The architecture I am using is the MVC (Model, View, Controller) architecture, this allows my application to follow a separation of concerns and addresses many of the SOLID principles that I am working with.

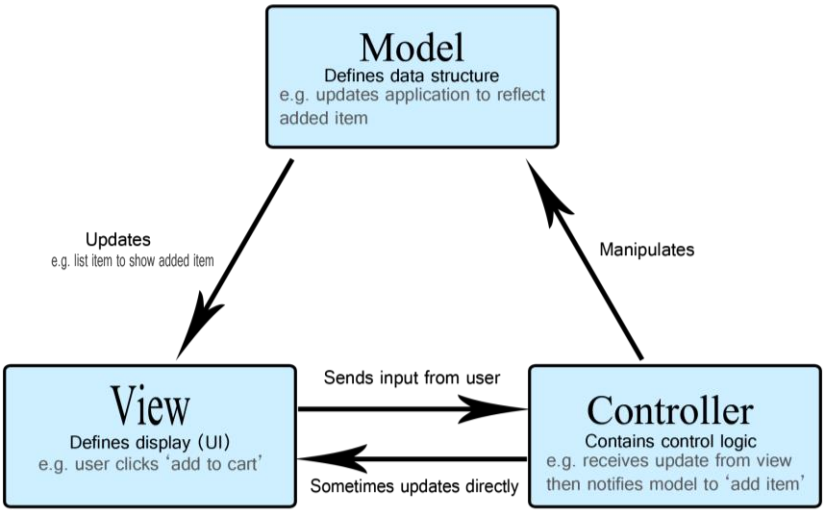


Figure.3.2. – the MVC Architecture. Mozilla (n.d.)

3.1 - Sprints Journey

I opted to incorporate sprints late into my development, I regret not continuing these on sooner, it helped me to separate the functionality of the project into smaller chunks which in turn gave me the greatest feeling of achieving working components of my application, which I thought would never happen.

CyberX Sprint Logs

Completed by:	**Victoria Hektor**
Deadline:	**05/05/2023 12 noon**

Sprint Logs

% done	Sprint No.	Instruction	Date	Due By	Completed?	Notes
100%	1	Frontend	January 13th	February 13th	Y	This month focus on as a minimum creating a working interface demo.
100%	2	Backend	February 14th	March 15th	N	I can't get the backend working, I think I am trying to do too much at once but I can't seem to figure a way around this.
100%	3	Backend	March 16th	April 16th	N	on each individual functionality, this worked, although I haven't completed all of the functionality, I am able to demo some of the backend functionality I wanted to.
100%	4	Finishing up	April 16th	May 5th	Y/N	Partially completed and pleased I actually got some of the backend to work!

Sprint Logs - Functionality

% done	Sprint No.	Function	Date	Due By	Completed?	Notes
100%	1	Login/Logout	January 13th	February 13th	Y	I have a working version, but not all together with the whole project.
100%	2	Database	February 14th	March 15th	Y	I have a working database 'cyberx' with 2 tables, room to add more if time.
100%	3	Dashboard	March 16th	April 16th	Y/N	I have managed to get a working admin dashboard that adds, views and updates users.
100%	4	Finishing up	April 16th	May 5th	Y/N	For this I am trying to bring everything together to work as one whole application as opposed to just individual working functions.

Figure.3.1.1 – My Sprint Logs

4 - Considerations

4.1 - Frontend & Design Considerations

For the design aspect I have made use of HTML and CSS to model my demo and prototype, I will continue to utilise these throughout my development journey and will utilise a Twig Template Engine to allow a separation of concerns with the front end design and will also add an additional layer of security, (Patel, 2016), achieved through using the Twig Library. ("Home - Twig - The flexible, fast, and secure PHP template engine," n.d.)

I have made use of the various open source HTML and CSS libraries available to me like; uiverse.io and Bootstrap, ("Open-Source UI elements - made with CSS and HTML," n.d.), ("JavaScript · Bootstrap," n.d.), which was initially created for Twitter by developers, ("What is a Bootstrap and how does it work?," n.d.). This will reduce coding time, increase productivity and address the pressing time constraints that this project brings with it. Bootstrap will provide support for me to ensure I can effectively create my project while considering time constraints and working more specifically on the more creative side of my project, the backend. Bootstrap also offers amazing responsive UI Components, layouts and tools. Additionally I have found a brilliant template which offered a huge library of utilities, it was hard to separate out what I wanted and didn't want but eventually got everything I needed for the frontend design. ("Histudy Website Template," n.d.).

For all of my logo and design needs I used Canva across the board to ensure a supported design environment, reduced complexity and frees up time to work on the backend development of my project. ("Free Design Tool," n.d.), I created various images, including backgrounds and a favicon for my web application. For some other images I utilised Shutterstock. ("Stock Images, Photos, Vectors, Video, and Music," n.d.)

4.2 - Backend Functionality Considerations

For this project I opted to develop CyberX in PHP, this language is one I am more familiar with and offers the most functionality when considering object oriented programming. The downside being because most web applications are created in PHP, it has become a popular target for XSS (Cross-Site Scripting) attacks, with that in mind, this also has paved the way for many patches to vulnerabilities with new techniques coming out each year to protect a web application from XSS, including using the Twig Templating Engine, Marashdih, Zaaba and Suwais (2018). I have also opted to utilise HTML & CSS for my front end design with support from the various open source resources available to me, eg. Bootstrap and ioverse.

I have a focus on attempting dependency injection within my application as it allows for much better application flexibility, easy maintainability and simpler testability of my code. Dependency injection also allows me to keep a separation of concerns within my application enabling a more modular view with ease of understanding, I will be afforded the ability to reuse code throughout my application which will save me time and reduce duplication of code. One such tool for this is utilising composer which will support me in building an application that integrates easily into frameworks and uses the autoloading function. Hofmann (2013)

Additionally there are many frameworks that I can utilise within my PHP project, including Slim, CodeIgniter, Symfony, Laravel, and Yii, just to name a few. Code Igniter offers a lightweight and straightforward framework, with a small footprint for modest web applications, releases are irregular so it is not the best choice if high security is needed. Symfony and Laravel seem to front runners with all operating within the MVC architecture that I'll be using. Slim offers a simpler approach and I have worked with this framework before making this the easier option to go with.

There were multiple tutorials available which could guide me through creating an application using CRUD (Create, Read, Update, Delete) operations and PHP PDO("PHP PDO CRUD," n.d.), (Kumar, 2017), not all followed the SOLID Principles, a SOC or a single point of access which made it difficult for me to create the desired web application. ("PHP PDO CRUD," n.d.)

4.3 - Database Considerations

For the database portion of my web application, I have many options to choose from, as I am coming from an object oriented perspective, I need to take this into consideration. From my research, I know that there are

differences in security between PDO and MySQLi, they both make use of prepared statements, and the differences are that PDO supports various databases, around 12, whereas MySQLi only supports MySQL with the addition of being slightly quicker but also classified as deprecated, making it a security risk. For this point I seriously need to consider the impact this will have on my web application and the databases I use, initially PDO or Doctrine were my go to but I am now considering other options. A valid point to note here; it is the programmer that defines the security of any application, not the language, library or driver a programmer uses.

It all comes down to which I am more familiar with, from this view, I am not overly keen or familiar with either and will look at both options and see which flows better. From the security perspective, I have to ensure that I sanitise any data from all kinds of input, even if a user doesn't necessarily input, if there is any kind of incoming data, it must be sanitised, and validated before it is stored, and prepared statements made use of.

When it came to creating a table for the users I opted to keep it as anonymous as possible by asking for minimal information. I considered expanding and asking for contact numbers, location etc but felt this may be too intrusive and open up different aspects of worldwide data laws. I am still considering the pros and cons of each and will circle back to this conundrum. More will be included in the major components section of my report.

4.4 - Version Control

For keeping track of my development journey, I have used GitHub as its version control offers a robust and popular platform which enables me to carry out code reviews, track bugs and, for the future, offers great collaboration tools. Due to the immense popularity of GitHub I am able to seek out support within the GitHub Community which offers me support in a way that not many version controls have, helping me to find solutions to problems quickly should I need to. Additionally, I have used GitHub in previous projects and group work through university assignments which helps me get a head start on working right away rather than having to learn any new platform. GitHub also offers a great planning tool which allows me to plan each step of my coding journey, and it helps me to keep track of what I have completed, what needs doing and future tasks coming up.

For the most part, I got on much better using PHPStorm("PhpStorm," n.d.), which the university provided a product key for learning use, and Notepad++("Downloads | Notepad++," n.d.) for development, Notepad has a more pleasing interface that I understand and offers a functionality for developers in that whichever file you save it as, it knows the indentations and colourises the file for ease of use. I had multiple copies stored in various places for fear of technical issues, this did make it difficult to keep track and often caused confusion for me.

4.5 - Learning Platform

For this section I have carried out much research on how I could implement this. So far I have discovered multiple existing libraries to support this functionality that utilises Javascript which sits nicely in the theme of an object oriented programming perspective. There were many ways to implement the e-learning aspect and there are security considerations for wanting an online compiler, which could leave CyberX vulnerable.

I have discovered multiple online tutorials which will guide me in creating an e-learning platform and provide the basics, allowing me to build on what I create in the tutorials any additional functionality I require in my project. From this perspective it is important that I consider their secure coding style and ensure I am security conscious and implement additional security as and when it is needed along the way. I noticed that with the tutorials and existing projects, the coding style wasn't secure and had the architecture combined with php within the html files. This caused difficulty in applying this to my web application.

5 - System Functionality – Major Components Journey

5.1 - Database Functionality, Security & Design Desires

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives CyberX_1"

NB: Please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

My database has many files relating to it, including the settings file, sql file and the DatabaseWrapper, the wrapper is there to provide a middle ground for database interaction allowing for secure communications, there are public and private methods and using monolog logging for logging of any errors, it makes use of prepared statements which in turn go some way to prevent SQL injection style attacks.

My original Database.sql file creates a database for 'cyberx' and adds an admin user (me), I wanted my database to have at least two tables for users and courses, with the potential to add more for users to test the knowledge they have learnt. For the user table I have the 'mem_id' as the PK with additional columns for the username, password and email. My courses table has the 'course_id' as the PK, along with the course name, description and topic. As I started on this journey I hadn't thought about how a user could test their skills after completing on of the courses and I wanted to add in a quiz time and question functionality, so I hope to be able to add in two more tables to facilitate this.

At this point in the database journey I have only managed to add in the user table and struggling to get everything to work, I keep attempting this aspect and hopefully this comes to fruition. I am utilising PHPMyAdmin and xampp to manage my database as this is the option that makes more sense to me, I have attempted via other methods but to no avail. With PHPMyAdmin I am able to import my sql commands and also carry out sql commands in the console, which allows me to run my code for the database as I go along and test it out.

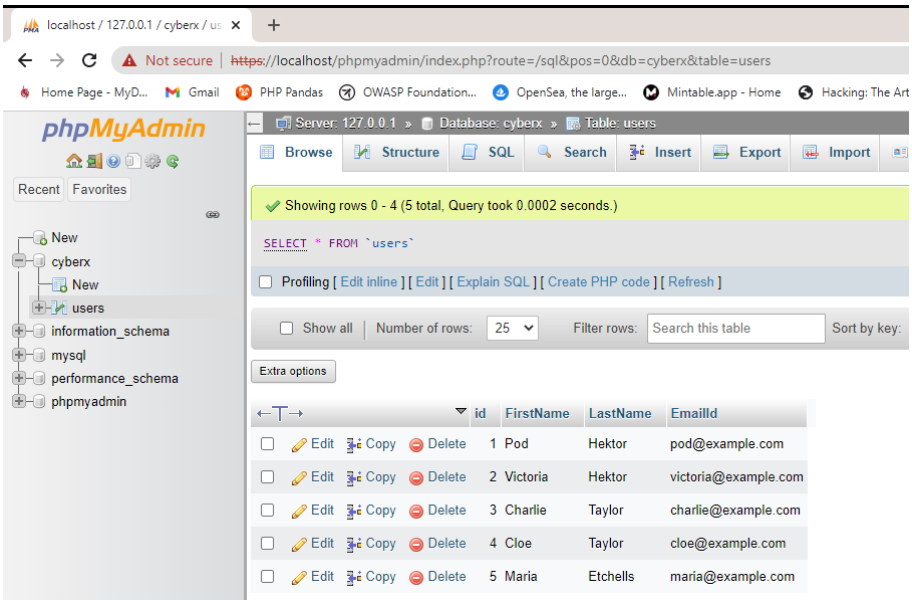


Figure.5.1.1 – PHPMyAdmin – cyberx users' table

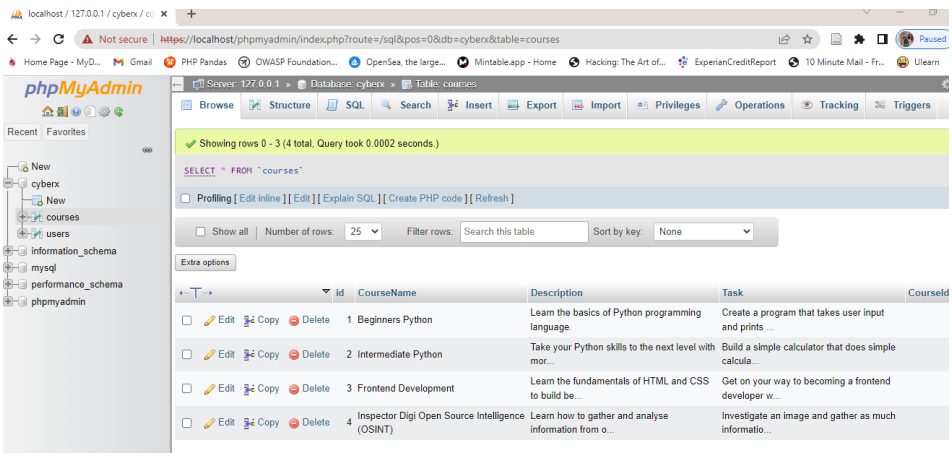


Figure.5.1.2 – PHPMyAdmin – cyberx courses table

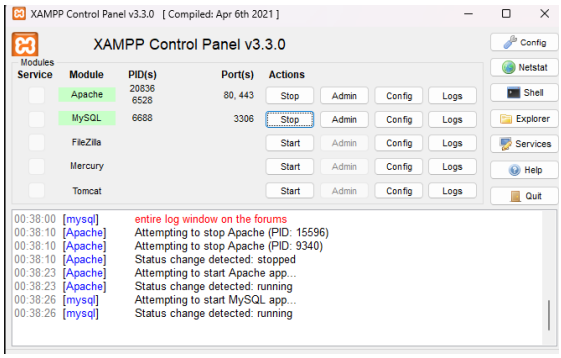


Figure.5.1.3 – xampp

5.2 - My Desired PHP Settings – from a secure perspective:

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives CyberX_1"

As it stands, I cannot get my web application to work the way I want it to, the following are my settings for if I should ever get the web application to work. I will place non-working code in a separate folder within the code directory, having an insecure web application isn't something I desired, although it is common for vulnerabilities to be present within working applications. (Doyle and Walden, 2011)

Constants are important in this process and will support me achieving the above mentioned points and ensures I can set variables that will not change.("PHP: Constants - Manual," n.d.) I first concatenate the variables, then define them all within the settings file. Firstly, I set a directory separator constant, ("PHP: Predefined Constants - Manual," n.d.), and set the value as 'DIRECTORY_SEPARATOR', this will allow my code to function a way that doesn't make it platform dependent and making it operationally versatile across various operating systems.

The next constant is for my web application name, to ensure I can use this throughout my application lowering the risk of any accidental changes. Moving forward, I just call the variable 'APP_NAME' whenever I need to call upon the name for a title within headers and wherever else I might need my app name to appear.

I am utilising BCRYPT to ensure a good balance between security and performance for my web application password hashing. With BCRYPT I am able to define what amount of computational resources I wish to use, the recommended amount is between 10 and 13, I have it set to 12. This allows for a good amount hashing which ultimately makes it harder to brute force my web application. Setting my BCRYPT cost as a constant allows me to easily adjust the cost in the future should my resources and/or hardware change. ("Hashing in Action," n.d.).

To ensure a cohesive and smooth browsing journey, I set up the superglobal variable, '$_SERVER', set $SCRIPT_NAME as the key in the array which stores the current script's path relative to the server document root, then passing the value to the 'dirname' function, resulting in the final value being assigned to the $app_url variable. This ultimately allows the application to work fluidly regardless of any server configuration. ("Understanding PHP Superglobals," n.d.)

To ensure my web application can generate and locate the correct css directory, I have added $css_path to my settings, I concatenate the $app_url variable with /css/ assigning this to $css_path, this allows CyberX to find the correct css files regardless of the current location of scripts, and despite any configurations within the server, it will still find the file path it needs. Additionally, as I am using bootstrap for some of my design, I layout the settings for this too, concatenating $css_path with the bootstrap location. Using bootstrap.min.css within my design framework allows me to minimize the amount of files needed, reduces the file sizes and ultimately helps with page loading times. ("Concatenation," 2022)

To ensure proper and legal logging, I am making use of the built-in logging features, including monolog logging. I ensure that current settings updated within my settings file for logging for all necessities, which include logging for when there are any changes to the databases, adding a layer of security should any breaches occur, logging for when a user logs in and also logging of sessions in a secure way to ensure a legal, user friendly, and secure web application that complies with data protection regulations. (Mendez, 2018)

Additionally I have pre-set the logout path and dashboard path within my settings so I am able to easily reference this within my codebase and keep consistency with my logout URL.

The settings portion of this file is to set out multiple configurations and database settings for CyberX. This covers things like ensuring error details are displayed to the user, I have also set 'addContentLengthHeader to false to prevent potential issues due to some web servers and/or proxies possibly removing the header, causing delays in response times. For now CyberX's mode is set to development as it is still in its development phase, as well as having debug set to true during this phase. The view settings are set out to configure the Twig related settings with cache set to false to avoid the templates being cached, and auto_reload set to true so any modifications will be reloaded into the view, the template path directs the application to the twig templates, this again uses concatenation via _DIR_ constant and adds the current directory with the '/TwigTemplates/ directory.

For my database I am using PDO (PHP Data Objects) and have set out the connection settings within my settings file, I have set the 'rdbms' to run MySQL, the host is set to localhost for now during development, the port is blank for now, and I have set the 'db_name', 'user_name', and 'user_password' to all represent my current databases. I have set my database collation to utf8_unicode_ci as is standard and the charset to utf8. For my PDO options, I have it set to throw exceptions when an error occurs rather than just spitting an error message, I have set it to 'PDO::ATTR_DEFAULT_FETCH_MODE' to fetch as an associative array and instructed it to emulate prepared statements which will provide a layer of security within my web application.("PHP: PDO::prepare - Manual," n.d.) . It is important for me to consider permissions and ensure that only the admin has full permissions within the application. (Laverdière et al., 2021)

5.3 - Login/Register/Logout Functionality, Validation & Sanitisation

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives/ CyberX_1"

NB: Please see shared folder: "Victoria_Etchells/Development/Code/cyberx-login-logout"

Validation is one of the most important parts of getting right within any development project to ensure safety, security and prevention of cyber-attacks, such as XSS. As it stands I cannot get the right validation, nor implement the bcrypt function, I have a basic PDO database that is only md5 hashed which isn't the most secure, I have found online tutorials abundant in older methods of creation in PHP but not many on current methods, the following is how I would set up this functionality from a secure perspective.

For most of the validating I am making use of the Controller part of the MVC Architecture (Model, View, Controller). With the login controller, I needed to validate input and output and ensure there is correct data and session management. The LoginController has a function called loginOrRegister which will be what receives object arguments, the validateInput function will then validate and clean the received objects (login credentials etc), it will check against the database and create a session value which is returned if the checks are successful, this function also is responsible for hashing the password using the bcryptWrapper. Once all of this is done, it creates an output to the user which is appropriate based on whether or not the user and/or session exists, or the credentials are incorrect, essentially a success or failure output. Should the session value correspond to the admin user, it will direct the view to the adminDashboardController view. I need to implement a security addition here as I don't think this right now will prevent back button displaying the old session, I will test this further and come back to it if needed, I understand I can make use of header(nocache) which may solve this issue. I do feel here that the LoginOrRegister function does a lot and doesn't have a clear separation of concerns, if there is time I will circle back and refactor this.

Additionally there is a separate file to function solely as the logging out facilitator, after checking everything out (user & session info etc) it will destroy the session and stores the session data, should no session value be set, it displays the homepage rather than the logout page.

My FormValidator provides my application with basic input validation and sanitisation of user input, here I have made use of FILTER_SANITIZE_FULL_SPECIAL_CHARS which will prevent potentially harmful characters being used within the username section of the form.

The ValidateUser file is there to specifically validate users login credentials, it makes use of the protective 'private variables for the sensitive user information such as the username, password (hashed & raw), and various database information. This file also checks the user credentials against those stored in the database and will create a new user if the user does not exist, it will also return success or failure message when comparing passwords to a user account.

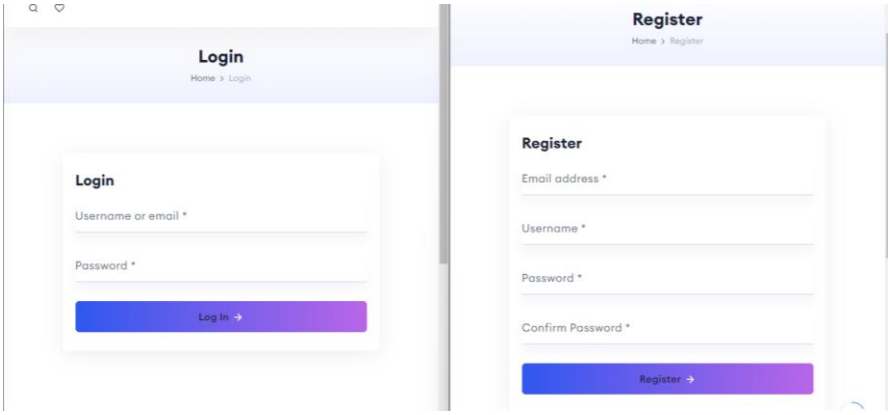


Figure.5.3.1. – Login/Register Function

5.4 - Session Management

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives/CyberX_1"

Incorrect session management could leave CyberX vulnerable to a myriad of attacks, for example; give attackers access to sensitive information. My session wrapper will go some way in taking care of providing a secure session for CyberX users. It sets a session logger with a value, and uses if statements to verify the value parameters, it also has the functionality to unset sessions and to also get existing sessions. The SessionModel then extends the session management and manages session data relating to the users, it allows for storage, retrieval and logging. I am also looking into how I can check for session hijacking and session fixating attacks Contrast Security (n.d.) which sees an attacker either hijacking sessions or using an already accessed session to carry out an attack. If time constraints allow I will look at implementing a session expiration time to combat the risk of session hijacking, and possibly use regeneration of session id's to combat the risk of session fixating, the downside to the regeneration is that sessions could be lost as it doesn't work well all of the time (PHP, n.d).

For proper session management I have added a session model, a session wrapper and session logger. The wrapper adds a layer of abstraction within the session management section. Although this doesn't work, I have the code for this and place the non-working code in a separate folder FYI.

5.5 – Desired Password Policy, Management & Security

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives/CyberX_1"

I will be making use of a PasswordWrapper file to facilitate password hashing and authentication, coupled with the bcrypt algorithm to provide a secure password journey. This files will check the password, hash it and then return a true or false view.

Password security is a critical aspect of any web application, and I have taken several steps to ensure that my platform provides users with the highest level of protection for their passwords. One of my top priorities is to prevent password reuse and encourage users to choose stronger passwords. To achieve this, I have implemented policies that require users to choose a new password that has not been used before and enforce password complexity rules that require a mix of uppercase and lowercase letters, numbers, and symbols.

On the back end, I have implemented the bcrypt algorithm for password hashing, which is considered one of the most secure hashing algorithms available. Bcrypt is designed to make it more difficult for attackers to crack passwords by adding a layer of complexity that slows down the password hashing process. This ensures that even if an attacker gains access to the password hashes, it would take them significantly longer to crack them.

In addition to bcrypt, there are other steps I could take in the back end to make my password policy more secure. For instance, I could implement two-factor authentication to provide an extra layer of security to the login process. This would require users to provide a second form of identification, such as a one-time code sent to their mobile device or email address, in addition to their password. This is something I plan to implement in the future.

Another option would be to implement rate limiting, which would limit the number of login attempts a user can make within a certain period. This would prevent brute force attacks, where attackers try multiple combinations of usernames and passwords to gain access to the platform. By limiting the number of login attempts, I can reduce the risk of a successful attack and provide users with greater peace of mind when it comes to the security of their accounts.

5.6 - Privacy Policy, Cookie Policy, Data Usage, & GDPR

NB: Please see shared folder: "Victoria_Etchells/Development/Code/Archives/CyberX_1"

I have opted to use an external generator for any cookie, data, and privacy policies, I have found Termly (2023) to offer the most comprehensive and it takes into account national regulations as well as international laws around data protection. I feel this is the best move as it stands to ensure I can focus my time and energy on the main components.

5.7 - Learning Platform

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: For course management, please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

The development of my learning platform was a challenging but rewarding process that required careful planning and attention to detail, although I would like more time on this area. One of the key components of the platform was the creation of four courses: a Beginners Python course, an Intermediate Python Course, an Open Source Intelligence Course (OSINT), and a Front End Web Development Course. Each of these courses had its own interface page and was designed to provide users with a comprehensive and engaging learning experience.

To create the courses, I conducted extensive research and drew on my own experience and knowledge in these fields. I decided to start with the Beginners Python course as it was the most fundamental course, and then progressed to the Intermediate Python course, which built upon the concepts covered in the previous course. The Open Source Intelligence Course (OSINT) and Front End Web Development Course were more advanced courses, aimed at users who had some interests in these areas.

Each course was designed to be engaging and interactive, with guided write-ups and interactive exercises to reinforce the concepts covered in the course. While I had initially thought of creating videos for each course, I decided to focus on creating guided write-ups first due to the time constraints of this project. However, I plan to create videos for the courses in the future to enhance the user's learning experience.

In addition to creating the courses, I also developed an admin functionality that enabled the deletion, editing, or addition of new courses. This functionality was essential in ensuring that the platform remained up-to-date and

relevant to users. The admin functionality was designed to be intuitive and user-friendly, with clear and concise instructions on how to perform each task.

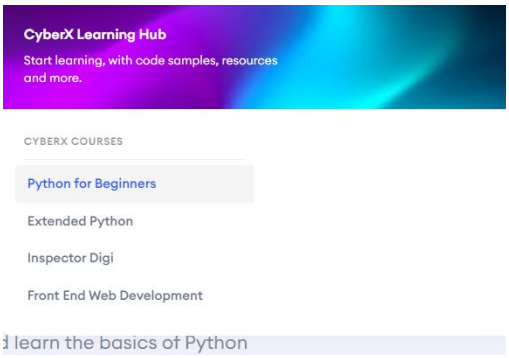


Figure.5.7.1. – CyberX Courses Menu

5.7.1 - CyberX Courses

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: For course management, please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

In this section I will be exploring the course content that I will implement and integrate within my web app. At this stage I am unsure how it will work with integrating a responsive online course and plan to undertake further learning online to supplement my knowledge on this. The architecture will have a twig file for each course, further facilitated by individual files located in the Routes, View and Controller portions of my existing architecture. There will also be an admin management section for the courses.

I would like to create a database to hold the courses which will allow for proper content management via an admin login, this will be visualised as an e-learning website and the content will be created by me. I have added links to further learning on the subjects and will also aid the user learning journey in a positive way and help to condense learning in a way that helps all learners from diverse backgrounds.

5.7.2 - Course one - Python for Beginners

For this initially I was going to implement an online compiler, but on reflection I now know that this would make my website vulnerable to XSS attacks if it is done incorrectly, building something like this is better done with experience and know-how, instead I opted for a written style guided coding session with links to external online compilers for users to check their code, ensuring the security of CyberX while still allowing students to see their work in real time. Python is considered an excellent programming language for beginners due to the following reasons:

Simple and Readable Syntax:	Python has a clean and straightforward syntax that is easy to understand, making it an ideal language for beginners. It uses indentation to denote blocks of code, which enforces good coding practices and promotes readability.
Beginner-Friendly Development Environment:	Python has a simple and beginner-friendly development environment with a wide range of Integrated Development Environments (IDEs) and online code editors, such as PyCharm, VSCode, and Jupyter Notebook, and online-python.com, ("Online Python - IDE, Editor, Compiler, Interpreter," n.d.), that provide features like auto-completion, error checking, and debugging, which helps beginners in writing and debugging code more effectively.
Immediate Feedback and Quick Results:	Python is an interpreted language, which means that code can be executed directly without the need for compiling, providing immediate feedback on the results. This allows beginners to see the outcomes of their code quickly, making it a motivating experience and helping them to learn faster.
Emphasis on Code Readability and Best Practices:	Python's syntax and coding style emphasize code readability and follow best practices, such as using indentation to denote blocks of code and enforcing proper naming conventions. This helps beginners develop good coding habits from the beginning and understand fundamental programming concepts.
Extensive Documentation and Community Support:	Python has extensive documentation, tutorials, forums, mailing lists, and online communities that provide support and resources for beginners. The large and active Python community is always willing to help newcomers, making it easy for beginners to seek assistance and learn from experienced developers.

Table.5.7.2.1. – Python benefits.

Python's simple syntax, large standard library, rich ecosystem of third-party libraries, immediate feedback, and extensive community support make it an excellent choice for beginners who are starting their coding journey and want to learn programming in a user-friendly and effective manner.

5.7.3 - Course two - Extended Python Exercise

For this exercise I have opted to extend the Python lesson and teach more in depth code, this will be an exercise to encourage creativity and show students that coding isn't boring, but artistic, and has many options for post-education employment. I chose Python due to its versatility and popularity and is a popular programming language for several reasons as detailed above in the course one section.

5.7.4 - Course three - Inspector Digi

For this course it will follow an OSINT style exercise and provide users with an image and they will need to answer questions about the metadata within the image as well as information about the location of the image. For this I have added some images to the CyberX Twitter Account which can be accessed without having to sign-up, a guide has been created for users to follow with some questions asked of them. I have included links to resources which will help them on their quest. I feel this will foster a passion for digital forensics within students who try their hand at this task.

5.7.5 - Course four - Basic Frontend Design

The world wants more full stack developers, and this course is designed to show users there's more to coding than just back end stuff. I have opted not to go for the basic Hello World and opted to encourage a creative input from the user. I will provide guidance on how to include images as links so everything can be done online without having to need a backend architecture. Creating the step by step guide for this was difficult as I needed to be able to write

the html code but this is then recognised within the web page as html code for my site, For this I may need to create and add images to depict the steps for the users to follow.

5.8 – Dashboard

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: Please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

I have a member dashboard for users login/register and I have a separate dashboard for admin logins. I wanted to keep these separate keeping in line with a separation of concerns. For this I needed to be able to link this with the database so that when an admin logs in, it shows the admin dashboard and not a user dashboard. I tried a separate approach for the logins and split these as admin login and member login. Eventually I discovered a template that fit the colour scheme I was looking for; it took a while to go through and take out what I didn't like and add in what I wanted.

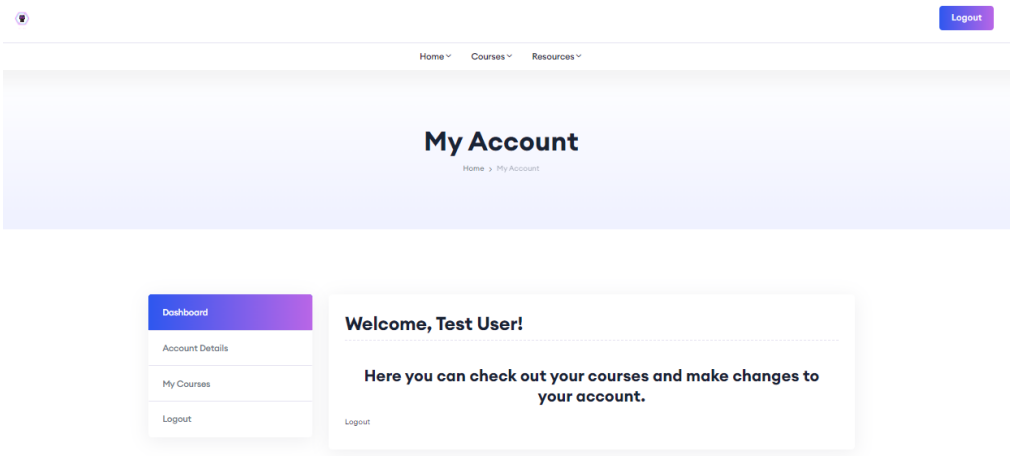


Figure.5.8.1. – Member Dashboard

The admin dashboard offers more functionality by way of being able to manage the users and the courses, despite the backend code not working for this, the demo was created to mimic the actions if my backend code would have worked. I am hoping to have this functional but possibly not in the way I had initially envisioned.

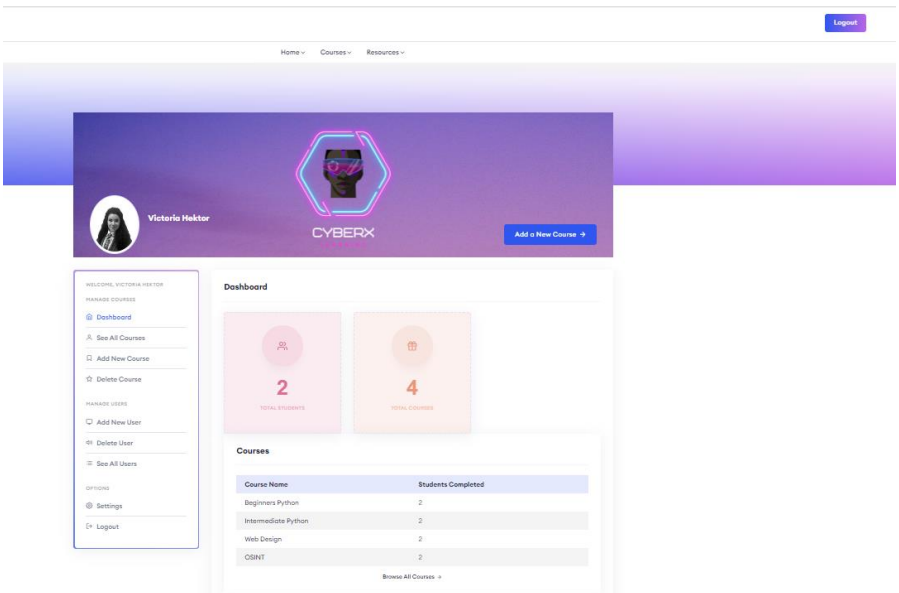


Figure.5.8.2 – Admin Dashboard

5.9 - Resources Functionality

For this part of the web application I wanted to make sure that young learners had access to free resources outside of CyberX, there is so much available online to encourage and foster tech based learning and I know that I would have loved to know about these types of resources when I was younger. If this impacts one young person then this has filled its functionality quota.

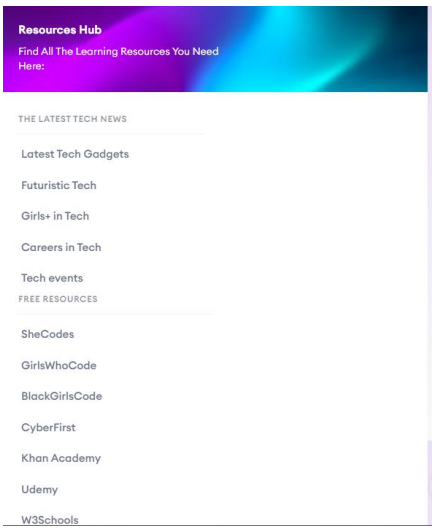


Figure.5.9.1. – Resources Menu

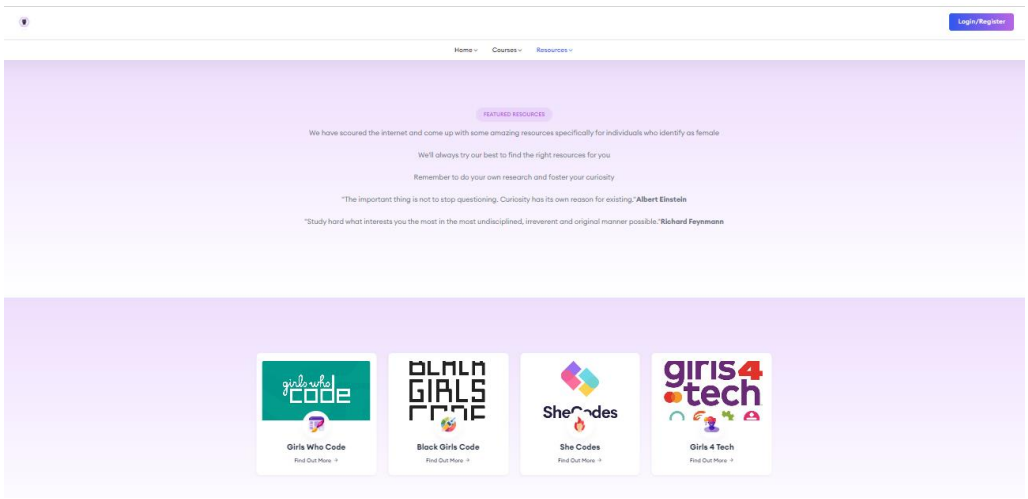
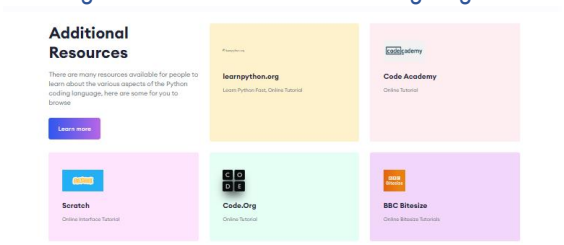


Figure.5.9.2 – Resources Landing Page 1



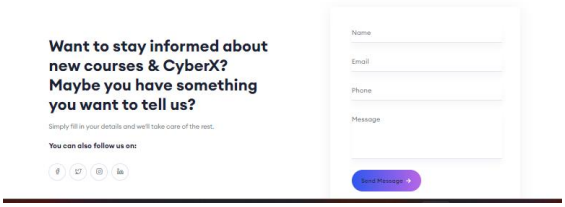


Figure.5.9.3 – Resources Landing Page 2

5.10 – Use Case Journey

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: For user management, please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

The journey for the users has changed quite a lot, with my weakness in development hindering progress, I opted to recreate the Use Case diagrams. The member doesn't have much functionality as it stands, and the Admin Use Case has a lot more functionality with being able to do everything that a user does plus additional user and course management.

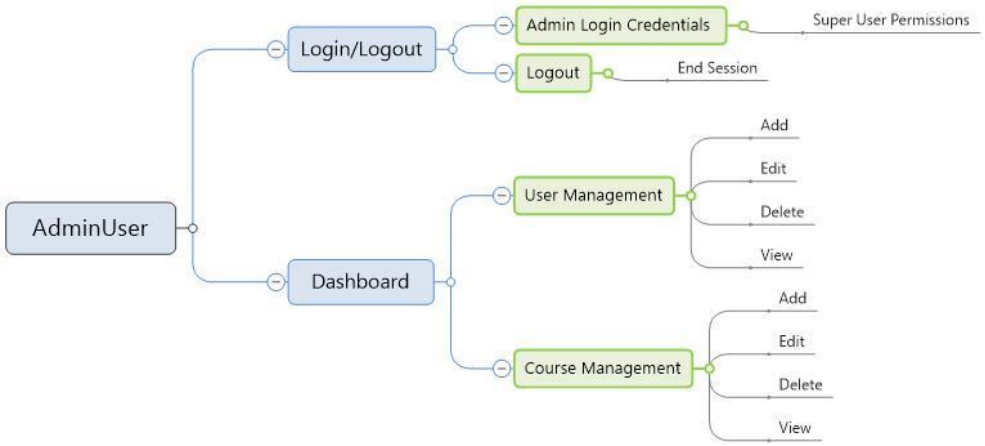


Figure.5.9.4 – Admin Use Case

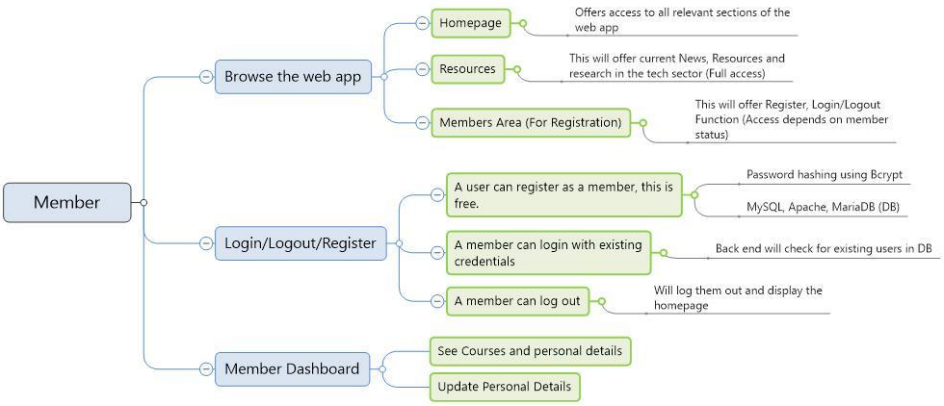


Figure.5.9.5 – Member Use Case

6 – Testing

NB: Please see shared folder: "Victoria_Etchells/Development/Testing/CyberX_Testing"

For all of the tests I carried out, I documented this in my shared folder in an excel spreadsheet. I wasn't able to test everything I had planned to due to non-working aspects of the application. Please *see appendix 12.3 for the tables.*

6.1 - Frontend

Testing a web application involves evaluating both the front end and back end to ensure that it performs as intended. In my experience, front-end testing involved checking the HTML functionality using w3 and evaluating the interface for navigation and usability. While testing the HTML functionality by browsing the website myself and using w3, I discovered broken links due to incorrect file names, and incorrect indents, which I promptly resolved to ensure the website's usability. I also evaluated the interface across various pages, including the home page, courses page, resources page, and login and register page, to ensure the links were working correctly, and users could navigate seamlessly between the different sections. By conducting comprehensive front-end testing, I was able to create a user-friendly and effective interface for CyberX.

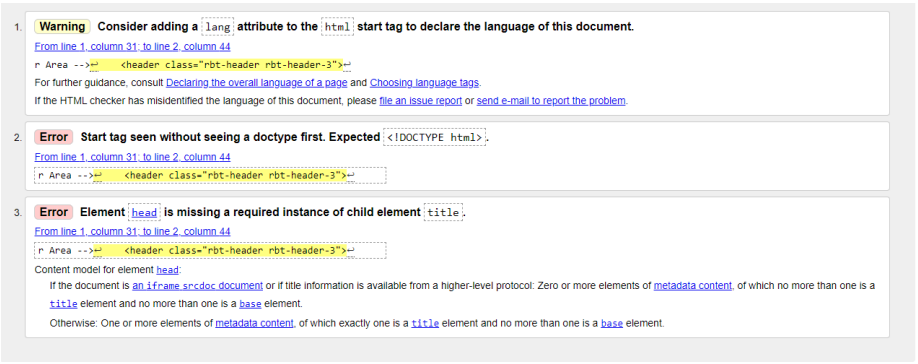


Figure.6.1.1 – Header Interface W3 Validation Check

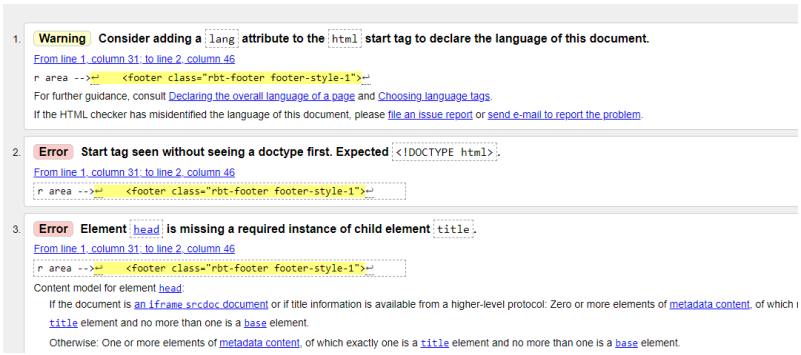


Figure.6.1.2 – Footer Interface W3 Validation Check

6.2 – Backend

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: Please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

Although I encountered some challenges with creating the back end of the web application, I feel that I am close to making it work correctly. However, due to time constraints, I was not able to complete the back-end testing and implementation fully. Despite this setback, I wanted to ensure that I could still demonstrate the navigation and usability of the web application to showcase its potential. As a result, I created a demo using HTML to showcase the application's functionality and allow others to navigate through its various pages. Although the back end was not fully implemented, I felt that the demo still provided a useful way to present the application's potential and highlight the various features that users could expect. In conclusion, while the back-end implementation remains a work in progress, I remain optimistic that with further testing and development, the application could, given more time, meet all of its objectives and provide users with a seamless and enjoyable experience.

After doing a sprint focusing solely on the database, I have managed to create the admin dashboard functionality with both back and front end working. It isn't built like I want it to be built but I am just grateful I have a working prototype of this functionality. Please note, I have had to zoom out on the webpage to get everything in the screenshot.

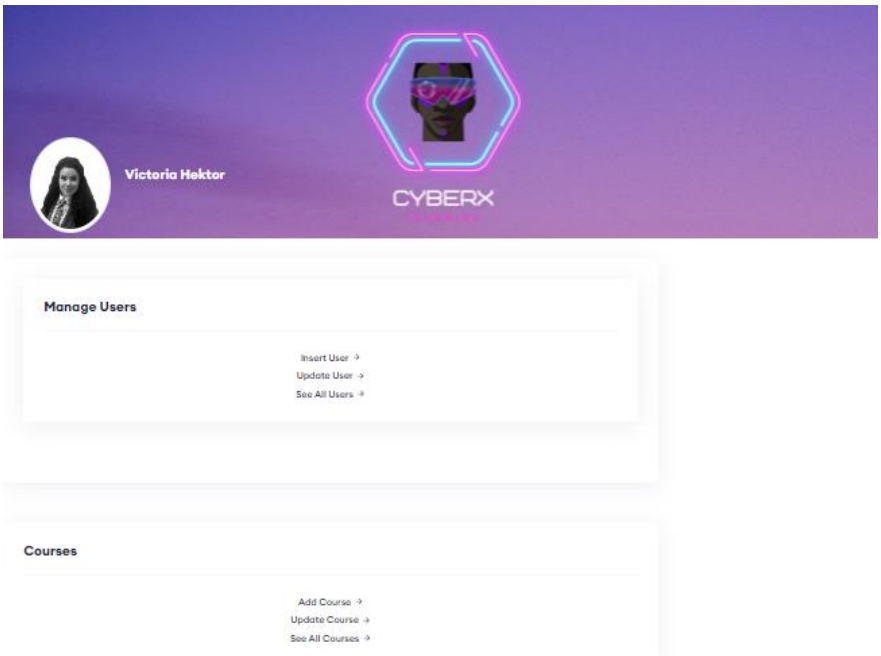


Figure.6.2.1 – Fully Functional Admin Dashboard

I have tested adding a user, which works perfect, and I am able to check within PHPMyAdmin and the 'See All Users' function to check that a user has been added, as well being able to go back to the admin dashboard.

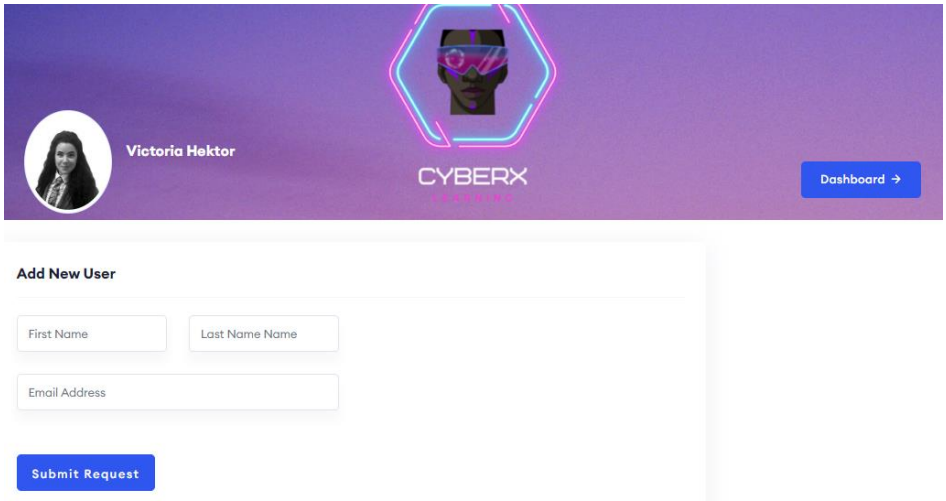


Figure.6.2.2 – Add User Functionality

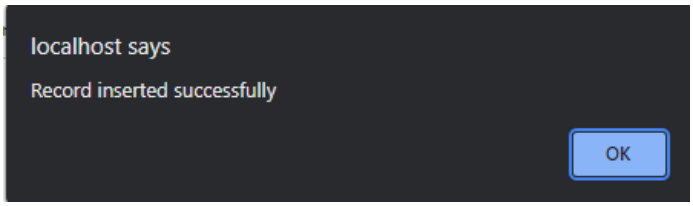


Figure.6.2.3 – Add User Functionality Confirmation

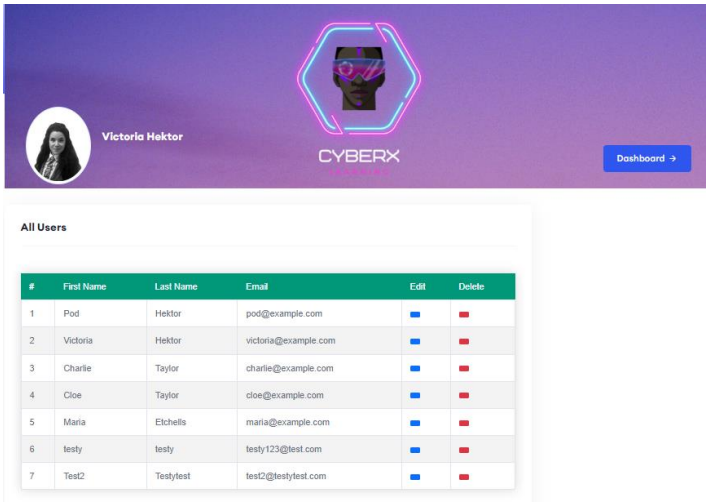


Figure.6.2.4 – Add User Functionality Test Check via interface

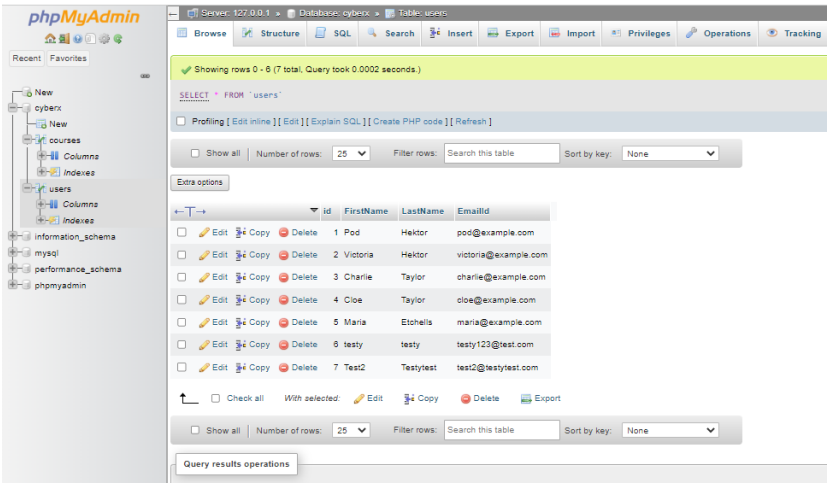


Figure.6.2.5 – Second Add User Functionality Test Check via PHPMyAdmin

#	Course Name	Description	Task	Edit	Delete
1	Beginners Python	Learn the basics of Python programming language.	Create a program that takes user input and prints the output.		
2	Intermediate Python	Take your Python skills to the next level with more advanced concepts.	Build a simple calculator that does simple calculations and takes user input.		
3	Frontend Development	Learn the fundamentals of HTML and CSS to build beautiful and functional websites.	Get on your way to becoming a frontend developer with this course, we will teach you some basics, the rest is up to you.		
4	Inspector Digi Open Source Intelligence (OSINT)	Learn how to gather and analyse information from open sources on the internet.	Investigate an image and gather as much information as possible about the image using freely available tools and techniques.		
5	Test	Test Course	To test the add course function		

Figure.6.2.6 – Add course functionality, followed the same steps as adding a user

7 - Implementation Report

NB: Please see shared folder: "Victoria_Etchells/Development/Code/CyberX_Demo"

NB: Please see shared folder: "Victoria_Etchells/Development/Code/cyberx"

As part of the development process for my web application, CyberX, I created an interface demo to demonstrate the application's various functionalities and navigation features. This interface demo was crucial in showcasing my vision for the application and enabled me to test the application's user interface and overall user experience. However, due to time constraints presented by the project, I was not able to fully implement a working back end behind the interface and have some of the Administration functionality on the dashboard; user and course management.

The interface demo allowed me to showcase the application's various features that I planned to implement in the backend, including the navigation across the home page, courses page, resources page, login, and register page. Additionally, I tested the application's HTML functionality using the w3 tool and made necessary changes to the file names for links that were initially wrong. Despite the limitations that the project presented, I am pleased with the overall outcome of the interface demo and believe that it accurately reflects my vision for CyberX. I have identified areas for further development and improvement and plan to continue working on the project in the future. As a separate demonstration, I have individual working functionalities of my application, the admin dashboard has the functionality to add a new user, update users, and view all users.

8 - Critical Reflection

This whole project turned out to be a larger undertaking than I had previously thought it to be, that being said, despite all of the frustrations and mental blocks that happened, I would choose to do this again. In doing this, I have become more resilient, understanding and adaptable in terms of software development. I have found it easier and easier to overcome hurdles and even my debugging methods are becoming more refined, I find that I enjoy trying to solve a problem despite the overwhelming fruastration of the problem/bug.

Even though I have code, some of the backend portion doesn't work, which frustrates me, it would take a solid six additional months for me to gain the knowledge needed to create a backend for CyberX, which I accept. The positive is, my front end development has come on leaps and bounds, for this I am grateful. I haven't managed to fulfil what I set out to do, and have created the frontend interface, which I really like, but the backend is inelegant, I am confident that I can walk away from this project knowing what I need to work on to refine my development skills.

A project like this in the world of work would generally have a whole team helping to create this and it took me a long while to stop being frustrated with myself for lack of progress and understanding in the different aspects of my project. This has made me realise that development is tough, and I am by no means a developer yet, with time and practice I hope to become a great one.

On reflection, my organisational style in the beginning was extremely chaotic and this in turn overwhelmed me, I'd delete whole sections and files of code because of my frustrations. It was a very tough journey of self-discovery and reflection, but ultimately, this has helped me to understand myself better in ways I thought I never would. Due to my neurodiversity I have always struggled with organisation and I always get overwhelmed with big projects and ideas, I feel that completing this project has helped me in so many ways that enable me to work alongside my ADHD and dyslexia and utilising these in a positive way to support my ideas and development, and future endeavours.

I will certainly complete further learning to ensure I can code in a more secure style as I have only managed get working code in a very chaotic style. I feel this has matched my personality and organisational style and is something I am battling with to try and change to a more calm and organised approach to projects.

9 - Further Research & Development

I specifically chose this project as it helps to provide a solution for a desperate need, gender disparity in tech, it is also something I wish to carry on after graduating university. Engaging more women in tech is a passion of mine and something I am trying to dive deeper into, creating this platform is the first step, for future development I will make this live as a more refined web application as well as looking into how I can provide outreach, specifically to disadvantaged schools.

There will be scope for adding web application functionality along with the outreach I desire to do, could also focus on building confidence in young girls+, if my research has taught me anything, it's that people who identify as female do not have much confidence in their own abilities and potential, I would love to provide something that supports and nurtures this aspect of a young girls+ life.

Within the Fuzzy Logic module that I completed in the first term of year three, I created a fuzzy inference system that would rate a schools performance in their delivery of tech education, I want to be able to incorporate this into my web application which will enable schools to seek out support in providing information which are focused on quantity and quality of the technology based subjects being taught, as well as taking into account the students journey, this will then provide a rating based on the answers provided. Additionally, I would use this information to put together a plan of action and recommendations, along with some resources and ideas for improving tech education engagement within a school setting.

I would love for this project to be something that makes an impact and that can interact with schools on an educational level that supports young girls+ learning journeys. This is something that could have a huge impact on the way tech education is delivered and will provide schools with the knowledge and resources needed to proceed in supplying a holistic technology education system for all. I wish for the future of CyberX to foster a collaborative learning community, where girls+ can connect with peers, mentors, and industry experts to seek guidance, share ideas, and collaborate on projects.

10 - Conclusion

Overall, I have been out of my depth but tried my best. I feel a more learned individual but still not quite understanding everything. I have created a working interface demo of CyberX which mimics what I would have wanted my backend to do. In hindsight I wish I had opted to use the python programming language as it is more modern with abundant, easier to understand tutorials. But ultimately, I have enjoyed the chaos that has been this project, despite the hurdles, and there were many, I am proud of what I have accomplished with this project and feel everything that I have experienced will make me a better person in

11 – References

CyberX_Learn (@CyberxLearn) / Twitter [WWW Document], 2023. . Twitter. URL https://twitter.com/CyberxLearn (accessed 4.29.23).

3.11.3 Documentation [WWW Document], n.d. URL https://docs.python.org/3/ (accessed 4.25.23).

Closing the Childhood Digital Divide: And end to digital exclusion for children and young people in the UK in the UK [WWW Document], n.d. . UNICEF UK. URL https://www.unicef.org.uk/policy/closing-the-digital-divide-uk/ (accessed 4.25.23).

Concatenation, 2022. . Wikipedia.

Curiosity Quotes (1226 quotes) [WWW Document], n.d. URL https://www.goodreads.com/quotes/tag/curiosity (accessed 4.29.23).

(@cyberx_learn) • Instagram photos and videos [WWW Document], n.d. URL https://www.instagram.com/cyberx_learn/ (accessed 5.5.23).

Downloads | Notepad++ [WWW Document], n.d. URL https://notepad-plus-plus.org/downloads/ (accessed 5.5.23).

Doyle, M., Walden, J., 2011. An Empirical Study of the Evolution of PHP Web Application Security. Presented at the Proceedings - 2011 3rd International Workshop on Security Measurements and Metrics, Metrisec 2011, pp. 11–20. https://doi.org/10.1109/Metrisec.2011.18

EXIF Data Viewer [WWW Document], n.d. URL https://exifdata.com/ (accessed 4.29.23).

exifmeta.com, n.d. Online EXIF & Metadata viewer [WWW Document]. Online EXIF Metadata Viewer. URL https://www.exifmeta.com/ (accessed 4.29.23).

Free Design Tool: Presentations, Video, Social Media [WWW Document], n.d. . Canva. URL https://www.canva.com (accessed 4.25.23).

Girls4Tech | Mastercard's STEM Education program for Middle School Girls [WWW Document], n.d. . Girls4Tech. URL https://www.girls4tech.com (accessed 4.29.23).

Hashing in Action: Understanding bcrypt [WWW Document], n.d. . Auth0 - Blog. URL https://auth0.com/blog/hashing-in-action-understanding-bcrypt/ (accessed 5.5.23).

Histudy Website Template [WWW Document], n.d. . ThemeForest. URL https://themeforest.net/search/histudy (accessed 5.5.23).

Home - Twig - The flexible, fast, and secure PHP template engine [WWW Document], n.d. URL https://twig.symfony.com/ (accessed 4.25.23).

How do I add a tab space instead of multiple non-breaking spaces ("nbsp")? [WWW Document], 2023. . Sentry. URL https://sentry.io/answers/how-do-i-add-a-tab-space-instead-of-multiple-non-breaking-spaces/ (accessed 4.30.23).

How To Make a Calculator Program in Python 3 | DigitalOcean [WWW Document], n.d. URL https://www.digitalocean.com/community/tutorials/how-to-make-a-calculator-program-in-python-3 (accessed 4.26.23).

HTML & CSS - W3C [WWW Document], n.d. URL https://www.w3.org/standards/webdesign/htmlcss (accessed 4.26.23).

HTML Styles CSS [WWW Document], n.d. URL https://www.w3schools.com/html/html_css.asp (accessed 4.26.23).

Is there a demand for front-end developers in 2022? [WWW Document], n.d. URL https://www.frontendplanet.com/are-front-end-developers-in-demand/ (accessed 4.26.23).

JavaScript · Bootstrap [WWW Document], n.d. URL https://getbootstrap.com/docs/3.4/javascript/ (accessed 4.25.23).

Kumar, A., 2017. PHP CRUD Operation using PDO Extension | PHP CRUD Operation. PHPGurukul. URL http://phpgurukul.com/php-crud-operation-using-pdo-extension/ (accessed 5.5.23).

Laverdière, M.-A., Julien, K., Merlo, E., 2021. RBAC protection-impacting changes identification: A case study of the security evolution of two PHP applications. Inf. Softw. Technol. 139, 106630. https://doi.org/10.1016/j.infsof.2021.106630

Learn computer science. Change the world. [WWW Document], n.d. . Code.org. URL https://code.org/ (accessed 4.29.23).

Learn HTML [WWW Document], n.d. . Codecademy. URL https://www.codecademy.com/learn/learn-html (accessed 4.29.23).
Madasu, V., Venna, T., Eltaeib, T., 2015. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. J. Eng. Sci. Technol. 2, 3159–40.
Mendez, J., 2018. PHP Monolog Tutorial: A Step by Step Guide [WWW Document]. Stackify. URL https://stackify.com/php-monolog-tutorial/ (accessed 5.5.23).
Online photo metadata and EXIF data viewer | Jimpl [WWW Document], n.d. URL https://jimpl.com (accessed 4.29.23).
Online Python - IDE, Editor, Compiler, Interpreter [WWW Document], n.d. URL https://www.online-python.com/ (accessed 4.25.23).
Open-Source UI elements - made with CSS and HTML [WWW Document], n.d. URL https://uiverse.io (accessed 4.25.23).
Patel, P., 2016. TWIG:Integrating Markup Languages and PHP for Web Development. Open Source You.
PHP: Constants - Manual [WWW Document], n.d. URL https://www.php.net/manual/en/language.constants.php (accessed 5.5.23).
PHP PDO CRUD [WWW Document], n.d. . Phppot. URL https://phppot.com/php/php-pdo-crud/ (accessed 5.5.23).
PHP: PDO::prepare - Manual [WWW Document], n.d. URL https://www.php.net/manual/en/pdo.prepare.php (accessed 5.5.23).
PHP: Predefined Constants - Manual [WWW Document], n.d. URL https://www.php.net/manual/en/dir.constants.php (accessed 5.5.23).
PhpStorm: PHP IDE and Code Editor from JetBrains [WWW Document], n.d. . JetBrains. URL https://www.jetbrains.com/phpstorm/ (accessed 5.5.23).
Python Program to Make a Simple Calculator [WWW Document], n.d. URL https://www.programiz.com/python-programming/examples/calculator (accessed 4.26.23).
Scratch - About [WWW Document], n.d. URL https://scratch.mit.edu/ (accessed 4.29.23).
Stock Images, Photos, Vectors, Video, and Music [WWW Document], n.d. . Shutterstock. URL https://www.shutterstock.com/discover/stock-assets-uk-0220 (accessed 5.5.23).
TinEye Reverse Image Search [WWW Document], n.d. URL https://tineye.com/ (accessed 4.29.23).
Understanding PHP Superglobals [WWW Document], n.d. . Eng. Educ. EngEd Program Sect. URL https://www.section.io/engineering-education/php-superglobals/ (accessed 5.5.23).
View Metadata [WWW Document], n.d. . metadata2go.com. URL https://www.metadata2go.com/view-metadata (accessed 4.29.23).
W3Schools online HTML editor [WWW Document], n.d. URL https://www.w3schools.com/tryit/tryit.asp?filename=tryhtml_hello (accessed 4.29.23).
Weggo! Create HTML [WWW Document], n.d. URL https://www.weggo.net/create.html (accessed 4.29.23).
What is a Bootstrap and how does it work? [WWW Document], n.d. . WhatIs.com. URL https://www.techtarget.com/whatis/definition/bootstrap (accessed 4.25.23).

12 - Appendices

12.1 – Shared Folders

12.1.1 - Shared folder for the testing notes: “Victoria_Etchells/Writing/TestPlanDocuments/CyberX_Testing.xlsx”

12.1.2 – Shared folder for the full code: “Victoria_Etchells/Development/Code/cyberx”

12.1.3 – Shared folder for the demo: “Victoria_Etchells/Development/Code/CyberX_Demo”

12.1.4 – Shared Folder for non-working code: “Victoria_Etchells/Development/Code/Archive”

12.1.5 – Shared Folder for Course Designs: “Victoria_Etchells/Writing/Reports/CourseDesigns”

12.2 – Course Design Notes

12.2.1 – Beginners Python

Title: Hello World in Python - A Fun Coding Adventure for Girls!

Website Message/Intro:

Welcome to Hello World in Python, a beginner-friendly online coding course designed just for young girls aged 11-17! Are you ready to embark on an exciting coding adventure and learn the basics of Python programming? Let's get started!

Course Overview:

Introduction to Python: Learn what Python is, why it's a popular programming language, and how to set up your coding environment.

Basic Python Syntax: Dive into the fundamental building blocks of Python programming, including variables, data types, operators, and basic input/output operations.

Fun with Functions: Explore how to create and use functions, which are like mini programs that can perform specific tasks. Discover how functions can make your code more organized and efficient.

Showcasing Your Work: Learn how to share and showcase your coding projects online and celebrate your accomplishments with family and friends!

What Makes Our Course Special:

Designed for Girls: Our course is specifically designed to empower young girls to explore the world of coding and build their confidence in STEM fields.

Beginner-Friendly: No prior coding experience is required. Our course starts from scratch and gently guides you through the concepts at your own pace.

Interactive Learning: Our lessons are designed to be engaging and interactive, with plenty of hands-on coding exercises and projects that encourage creativity and critical thinking.

Fun and Creativity: Our course emphasizes the creative aspect of coding, encouraging you to express yourself through coding and create projects that reflect your interests and personality.

Join Us Today!

Are you ready to embark on an exciting coding adventure? Enrol in our ‘Hello World’ in beginners Python course and unlock the world of coding with creativity, fun, and empowerment! Sign up now and let's code together!

Step By Step Guide

Step 1: Welcome message

Welcome and congratulations on starting your first steps into learning Python, here we will explore a little about programming and give you some starter tips on how to create in Python. This is a short course and may take between 1-2 hours. I highly recommend doing your own research on everything you are taught here to help you understand what everything is doing, this will also help to condense your learning.

Pro Tips:

Python files are always saved as .py, so for this lesson it will be saved as HelloWorld.py

There are many ways to test your code that you have written, just google “Online Python compilers”, and choose the one you like. We have provided a link to one for this course.

Resources needed:

An online python compiler to check your code in real-time. A text editor where you can write and save your code. Our recommendation: https://www.online-python.com/

Step 1. Creating your first python script

Open the text editor of your choice and save it as hello.py – be sure to save this in a folder that you’ll remember, you could even create your own folder where you can save all of your scripts. In the file, type:

```
print("Welcome to Hello World in Python!")
```

```
print("This is your first step in learning Python programming.")
```

Step 2: User input for name

Now let’s use the input() function add in your name, we will store it in a variable for personalised greetings.

```
name = input("What's your name? ")
```

Step 3: Greeting message

Display a personalised greeting message using the user's name.

```
print(f"Hello, {name}!")
```

```
print("Let's start our coding adventure together!")
```

Step 4: Explanation of print statement

Explain the concept of the print() function in Python, which is used to display messages or outputs to the console. Provide a brief overview of its usage.

```
print("In Python, 'print' is a function that displays messages or outputs to the console.")
```

```
print("You can use it to show text, numbers, or the results of calculations.")
```

Step 5: Example of a print statement

Show an example of a print() statement to reinforce the concept. Display a simple "Hello, world!" message as a common programming tradition.

```
print("Example: ")
```

```
print("Hello, world!")
```

Step 6: Prompt for user input to continue

Pause the program and prompt the user to press Enter to continue to the next step.

```
input("Press Enter to continue...")
```

Step 7: Conclusion

Congratulations completing your first Python code, we hope you continue coding and exploring the world of programming.

```
print("Congratulations!")

print("You've successfully written your first Python code.")

print("Keep coding and exploring the world of programming!")
```

12.2.2 – Intermediate Python

Title: Intermediate Python Methods

The following code allows the user to input two numbers and choose an operation to perform (addition, subtraction, multiplication, or division). It then performs the corresponding calculation and displays the result. You can run this code in any Python environment, such as an online compiler or a local Python installation, to create a simple calculator that can perform basic arithmetic operations. Feel free to customise and extend the code to add more functionalities or improve the user experience!

It is important to note that there are many ways in which to create a calculator and additional resources are provided at the end so you can explore this.

```
# Step 1: Getting Input from the User

# Ask the user for the first number

num1 = float(input("Enter the first number: "))


# Ask the user for the second number

num2 = float(input("Enter the second number: "))


# Step 2: Performing Calculations

# Ask the user for the operation to perform

operation = input("Enter the operation (+, -, *, /): ")

# Perform the corresponding calculation based on the user's input

if operation == "+":

    result = num1 + num2

elif operation == "-":

    result = num1 - num2

elif operation == "*":

    result = num1 * num2

elif operation == "/":

    result = num1 / num2

else:
```

```
    print("Invalid operation!")
    exit()


# Step 3: Displaying the Result
# Print the result to the console
print("Result: ", result)
```

Step by Step Guide

Step 1: Getting Input from the User

```
# Ask the user for the first number
num1 = float(input("Enter the first number: "))
# Ask the user for the second number
num2 = float(input("Enter the second number: "))
```

In this step, the code uses the input() function to get input from the user. The input() function displays a prompt to the user and waits for them to enter a value. The entered value is then stored in the variables num1 and num2 after converting it to a float using the float() function. The float() function is used to convert the input to a floating-point number, which allows for decimal inputs.

Step 2: Performing Calculations

```
# Ask the user for the operation to perform
operation = input("Enter the operation (+, -, *, /): ")


# Perform the corresponding calculation based on the user's input
if operation == "+":
    result = num1 + num2
elif operation == "-":
    result = num1 - num2
elif operation == "*":
    result = num1 * num2
elif operation == "/":
    result = num1 / num2
else:
    print("Invalid operation!")
    exit()
```

In this step, the code asks the user for the operation they want to perform using the input() function and stores it in the variable operation. The code then uses conditional statements (if, elif, else) to determine the corresponding calculation to perform based on the value of operation. If the input is valid, the result of the

calculation is stored in the variable result. If the input is not valid, the code displays an error message and exits the program using the print() and exit() functions.

Step 3: Displaying the Result

```
# Print the result to the console

print("Result: ", result)
```

In this step, the code uses the print() function to display the result of the calculation to the console, along with a label "Result:". The result variable contains the calculated result from the previous step, and it is displayed as the output of the calculator.

Additional Resources:
https://www.programiz.com/python-programming/examples/calculator

https://www.digitalocean.com/community/tutorials/how-to-make-a-calculator-program-in-python-3

12.2.3 – Basic Frontend Development

Title: Beginners' Guide to Creating a Front-End Webpage Using HTML and CSS

In this tutorial, you will learn how to design a front-end webpage using HTML and CSS. This tutorial is perfect for beginners who are new to web development and want to learn how to create a simple webpage with basic styling.

The tutorial will cover the following steps:
Setting up the HTML file: You will learn how to create a new HTML file and set up the basic structure of an HTML page, including the <!DOCTYPE html> declaration, the <html>, <head>, and <body> tags.

Adding the HTML structure: You will learn how to add various HTML elements, such as headings, paragraphs, images, and links, to create the content of your webpage.

Adding CSS for styling: You will learn how to create a separate CSS file, link it to your HTML file, and use CSS rules to style your HTML elements. Topics covered will include applying colours, font sizes, and text decorations, as well as using selectors and classes.

Previewing your webpage: You will learn how to open your HTML file in a web browser to see the preview of your webpage, and how to make changes to your HTML and CSS files and see the updates in the browser.

By the end of this tutorial, you will have a basic understanding of how to create a front-end webpage using HTML and CSS and be able to design a simple webpage with basic styling. You can then continue to expand your web development skills by learning more advanced HTML and CSS concepts, as well as integrating JavaScript to create interactive webpages. Happy coding!

Step 1: Set Up the HTML File
Open a text editor or an Integrated Development Environment (IDE) such as Notepad++, Visual Studio Code, Sublime Text, or Atom.

Create a new file and save it with a ".html" extension, e.g., "index.html".

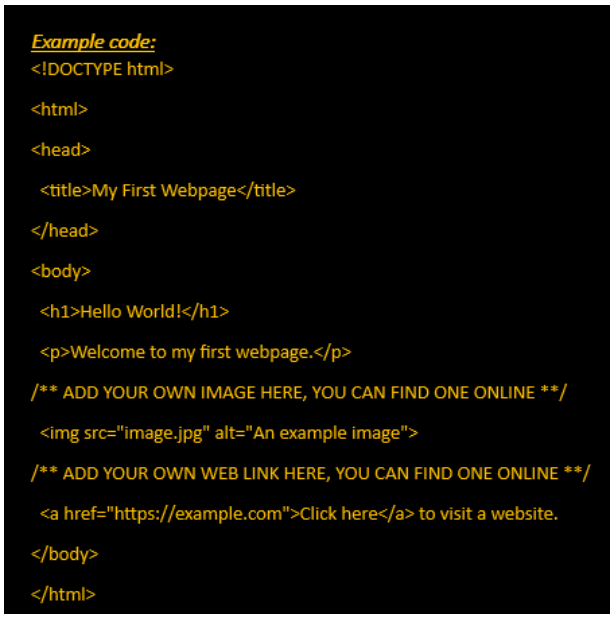
Step 2: Add the HTML Structure
Start with the basic structure of an HTML page by adding the <!DOCTYPE html> declaration at the top of your file. This tells the browser that you are using HTML5.

Create the HTML document structure using the <html>, <head>, and <body> tags. The <head> section is used to define meta-information about the page, such as the title, which appears in the browser's title bar or tab.

Inside the <body> tag, you can add various HTML elements such as headings, paragraphs, images, links, and more to create the content of your webpage.

Example code:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <h1>Hello World!</h1>
  <p>Welcome to my first webpage.</p>
/** ADD YOUR OWN IMAGE HERE, YOU CAN FIND ONE ONLINE **/
  <img src="image.jpg" alt="An example image">
/** ADD YOUR OWN WEB LINK HERE, YOU CAN FIND ONE ONLINE **/
  <a href="https://example.com">Click here</a> to visit a website.
</body>
</html>
```



Step 3: Add CSS for Styling

Create a new file and save it with a ".css" extension, e.g., "style.css".

Add CSS rules in your CSS file to style your HTML elements. CSS allows you to control the layout, colour, font, and other visual aspects of your webpage.

CSS Example Code

```
/* style.css */
h1 {
  colour: blue;
```

```
  font-size: 24px;
}
p {
  colour: #333;
  font-size: 16px;
}
img {
  width: 100%;
}
a {
  colour: green;
  text-decoration: none;
}
```

CSS Example Code

```
/* style.css */
h1 {
  colour: blue;
  font-size: 24px;
}
p {
  colour: #333;
  font-size: 16px;
}
img {
  width: 100%;
}
a {
  colour: green;
  text-decoration: none;
}
```

Link the CSS file to your HTML file by adding a <link> tag inside the <head> section of your HTML file. Set the href attribute to the file path of your CSS file.

Example code:

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>My First Webpage</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Hello World!</h1>
  <p>Welcome to my first webpage.</p>
  <img src="image.jpg" alt="An example image">
  <a href="https://example.com">Click here</a> to visit a website.
</body>
</html>
```

Step 4: Preview Your Webpage

Open your HTML file in a web browser to see the preview of your webpage. You can do this by right clicking the file and selecting 'open with' then choose a web browser.

Make changes to your HTML and CSS files and refresh the browser to see the updates.

Congratulations! You've created your first front-end webpage using HTML and CSS. You can now continue to learn more advanced HTML and CSS concepts, such as responsive design, CSS frameworks, and JavaScript integration, to build more complex webpages. Happy coding!

Additional Learning Resources:
https://www.w3.org/standards/webdesign/htmlcss

https://www.w3schools.com/html/html_css.asp

12.2.4 - Inspector Digi

Question/s?

What date was this image taken?

Can you find out the device on which the image was taken?

What famous location is in the image?

Can you find out what iconic painting is this place famous for?

What other interesting metadata can you find?

Guide/s:

Welcome to the beginner's guide to Open Source Intelligence (OSINT) activities! OSINT is the process of gathering information from publicly available sources to gain insights and intelligence for various purposes, such as research, investigation, or decision-making. In this guide, we will cover the basics of OSINT activities with a focus on the legal and ethical considerations specific to the UK and provide you with a step-by-step approach to get started.

The Plan:

Step 1: Define Your Objective

The first step in any OSINT activity is to clearly define your objective. What information are you looking for? What is the purpose of your investigation or research? Clearly define your goals and objectives to guide your OSINT activities, while keeping in mind the legal and ethical framework in the UK.

Step 2: Choose Your Sources

In the UK, there are several sources of publicly available information that can be used for OSINT activities, including:

* Search engines:* Popular search engines like Google and Bing can be used to search for information on websites, forums, social media, news articles, and more.

* Social media:* Social media platforms such as Facebook, Twitter, LinkedIn, Instagram, and others can provide valuable information about individuals, organizations, events, and activities. However, it's important to adhere to the privacy settings and terms of service of these platforms.

* Public records:* Publicly available records such as property records, court records, business registrations, and government databases can provide valuable information.

* News and media:* News websites, online publications, and media archives can provide information about recent events, stories, and trends.

* Forums and discussion boards:* Online forums and discussion boards related to your topic of interest can provide insights, opinions, and information from individuals with relevant knowledge.

* Websites and blogs:* Websites, blogs, and online forums related to your topic can provide valuable information and insights.

* Publicly available documents:* Reports, whitepapers, research papers, and other publicly available documents can provide in-depth information and insights.

Step 3: Use Search Techniques

When using search engines, it's important to use effective search techniques to get relevant results. Here are some tips:

* Use advanced search operators:* Search operators like site:, filetype:, inurl:, and intext: can help you refine your search and find specific types of information.

Use quotation marks: Placing search terms in quotation marks ("") will search for an exact phrase, which can be helpful when looking for specific information.

Try different search engines: Different search engines may have different results, so it's worth trying multiple search engines to get a comprehensive search result.

Step 4: Verify Information

Once you find information, it's crucial to verify its accuracy and reliability. Always cross-check information from multiple sources to ensure its validity. Be cautious of misinformation, fake news, and biased sources.

Step 5: Organise and Analyse Information

As you gather information, it's important to organise and analyse it effectively. Use tools like spreadsheets, note-taking apps, or specialised OSINT tools to keep track of your findings. Analyse the information to identify patterns, trends, and relationships that can provide insights.

Step 6: Respect Privacy and Legal Guidelines

During your OSINT activities, it's essential to respect privacy and legal guidelines. Avoid accessing restricted information, hacking, or engaging in illegal activities. Always follow ethical guidelines and laws related to OSINT in the UK.

Step 7: Stay Safe and Secure

Finally, remember to prioritise your safety and security during OSINT activities. Be cautious of sharing personal information online, and use tools like VPNs, proxy servers, or anonymisation techniques to protect your identity and privacy.

Step by Step Guide:

Step 1: Choose an Online Metadata Extraction Tool

There are several online tools available that can extract metadata from images. Some popular options include: (see html file)

Choose a tool that suits your needs and has good user reviews.

Step 2: Upload Your Image

Once you have selected a metadata extraction tool, navigate to the website and look for the option to upload an image. This may be labelled as "Choose File" or "Upload Image" depending on the tool you are using. Click on the button and select the image you want to extract metadata from from your local device.

Step 3: Extract Metadata

After uploading the image, the online tool will process the image and extract the metadata embedded within it. This may include information such as camera make and model, exposure settings, date and time the photo was taken, GPS coordinates, and more. The extracted metadata will be displayed on the screen, allowing you to view and analyse the information.

Step 4: Analyse and Interpret the Metadata

Once the metadata is extracted, it's important to analyse and interpret the information. This can provide valuable insights about the image, such as the camera used to capture it, the location where it was taken, and other details that may be relevant to your investigation or research.

Step 5: Save or Export Metadata

Depending on the online tool you are using, you may have the option to save or export the extracted metadata. This can be useful for documentation purposes or further analysis. Follow the instructions provided by the tool to save or export the metadata in a suitable format, such as CSV or XML.

Step 6: Consider Privacy and Legal Considerations

It's important to be mindful of privacy and legal considerations when extracting metadata from images. Some images may contain sensitive information, such as geolocation data, that may violate privacy rights or local laws.

Always ensure that you have proper authorisation and legal rights to access and use the metadata extracted from images.

Step 7: Stay Safe and Secure

Lastly, always prioritize your safety and security when using online resources. Use reputable and trustworthy online tools and be cautious of sharing personal information or uploading images that may contain sensitive data.

Remember, metadata extraction from images should always be conducted in an ethical and legal manner, respecting privacy rights, and adhering to applicable laws and regulations.

Social Links to Pictures to look for metadata:

Instagram: https://www.instagram.com/cyberx_learn/

Twitter: https://twitter.com/CyberxLearn

12.3 – Testing Screenshots

CyberX Testing

To be completed by:	Victoria Hektor	
Deadline:	05/05/2023	

Functionality Testing (HTML)

% done	Test Name	Actual Test	Expected Result	Actual Result	Working?	Notes
100%	Homepage (HP) — Links & buttons, View & Navigation	Clicking all Buttons on the home Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Learning Platform (LP) - Links & buttons, View & Navigation	Clicking all Buttons on the LP, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Resources Page (R.) - Links & buttons, View & Navigation (Include subsequent buttons/links (Resources)	Clicking all Buttons on the R Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Login/Logout/Register Page (LL & R) - Links, buttons and view	Clicking all Buttons on the LL & R Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Members Area (MA) - Links & buttons, View & Navigation	Clicking all Buttons on the MA Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Course Pages (C) - Links & buttons, View & Navigation (x4 pages)	Clicking all Buttons on the C Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	About Us Page (AU) - Links & buttons, View & Navigation	Clicking all Buttons on the AU Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Contact Us Page (CU) - Links & buttons, View & Navigation	Clicking all Buttons on the CU Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Some didn't navigate which highlighted some name link issues	No	Went through and changed the names to the right ones, Although I am still not happy about my logo placement and size or the header
100%	Follow-up notes:	All had the same issue, went through the application again and retested, results below.				

Figure.12.3.1. – Frontend Testing

Functionality Testing (HTML) - Retest						
% done	Test Name	Actual Test	Expected Result	Actual Result	Working?	Notes
100%	Homepage (HP) — Links & buttons, View & Navigation	Clicking all Buttons on the home Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Learning Platform (LP) - Links & buttons, View & Navigation	Clicking all Buttons on the LP, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Resources Page (R.) - Links & buttons, View & Navigation (Include subsequent buttons/links (Resources)	Clicking all Buttons on the R Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Login/Logout/Register Page (LL & R) - Links, buttons and view	Clicking all Buttons on the LL & R Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Members Area (MA) - Links & buttons, View & Navigation	Clicking all Buttons on the MA Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Course Pages (C) - Links & buttons, View & Navigation (x4 pages)	Clicking all Buttons on the C Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	About Us Page (AU) - Links & buttons, View & Navigation	Clicking all Buttons on the AU Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Contact Us Page (CU) - Links & buttons, View & Navigation	Clicking all Buttons on the CU Page, ensuring the view is as expected	Navigates to expected page and the view is as expected	Navigated Correctly	Yes	All Navigates Correctly.
100%	Follow-up notes:	I am still unhappy regarding the logo placement and size and will work on this.				

Figure.12.3.2. – Frontend Retest

Processes Testing (Admin Dashboard etc)						
% done	Test Name	Actual Test	Expected Result	Actual Result	Working?	Notes
100%	Add User	add a user via the dashboard	Adds new user	Correctly adds a user	Y	Checked via interface and via PHPMyAdmin
100%	Delete User	Delete selected user	Deletes User	Asks if I am sure, press ok, then deletes	Y	Checked via interface and via PHPMyAdmin
100%	Edit User	Edit user details	Correctly changes details	It doesn't display anything	N	Worked on trying to debug but couldn't figure out.
100%	Add Course	Add a course via the admin dashboard	Adds a course	Correctly adds course		Checked via interface and via PHPMyAdmin
100%	Edit Course	Edit a course via the admin dashboard	Correctly edits the course details of choice	It doesn't display anything	N	Worked on trying to debug but couldn't figure out.
100%	Delete Course	Delete a course via the admin dashboard	Deletes chosen course	Gives correct interface but doesn't delete	N	I have tried separating this function and used javascript in the html code.
	View All Courses	Click 'view all courses'	correctly shows all courses	correctly shows all courses	Y	No issues
100%	View All Users	Click 'view all users'	Correctly shows all users	Correctly shows all users	Y	No issues

12.3.3 – Admin Dashboard Testing

Processes Testing (Admin Dashboard retest)						
% done	Test Name	Actual Test	Expected Result	Actual Result	Working?	Notes
100%	Edit User	Edit user details	Correctly changes details	It doesn't display anything	N	Worked on trying to debug but couldn't figure out.
100%	Edit Course	Edit a course via the admin dashboard	Correctly edits the course details of choice	It doesn't display anything	N	Worked on trying to debug but couldn't figure out.
100%	Delete Course	Delete a course via the admin dashboard	Deletes chosen course	Gives correct interface but doesn't delete	N	I have tried separating this function and used javascript in the html code.

12.3.3 – Admin Dashboard Retesting

12.4 – First Deliverable Information (For Reference)

12.4.1 – First Deliverable System Design & Documentation

I wanted to keep a copy of my system design documentation and feel it is relevant in looking at what I have managed to do and what I have not in regard to the development.

Figure.12.1. Front End System Design

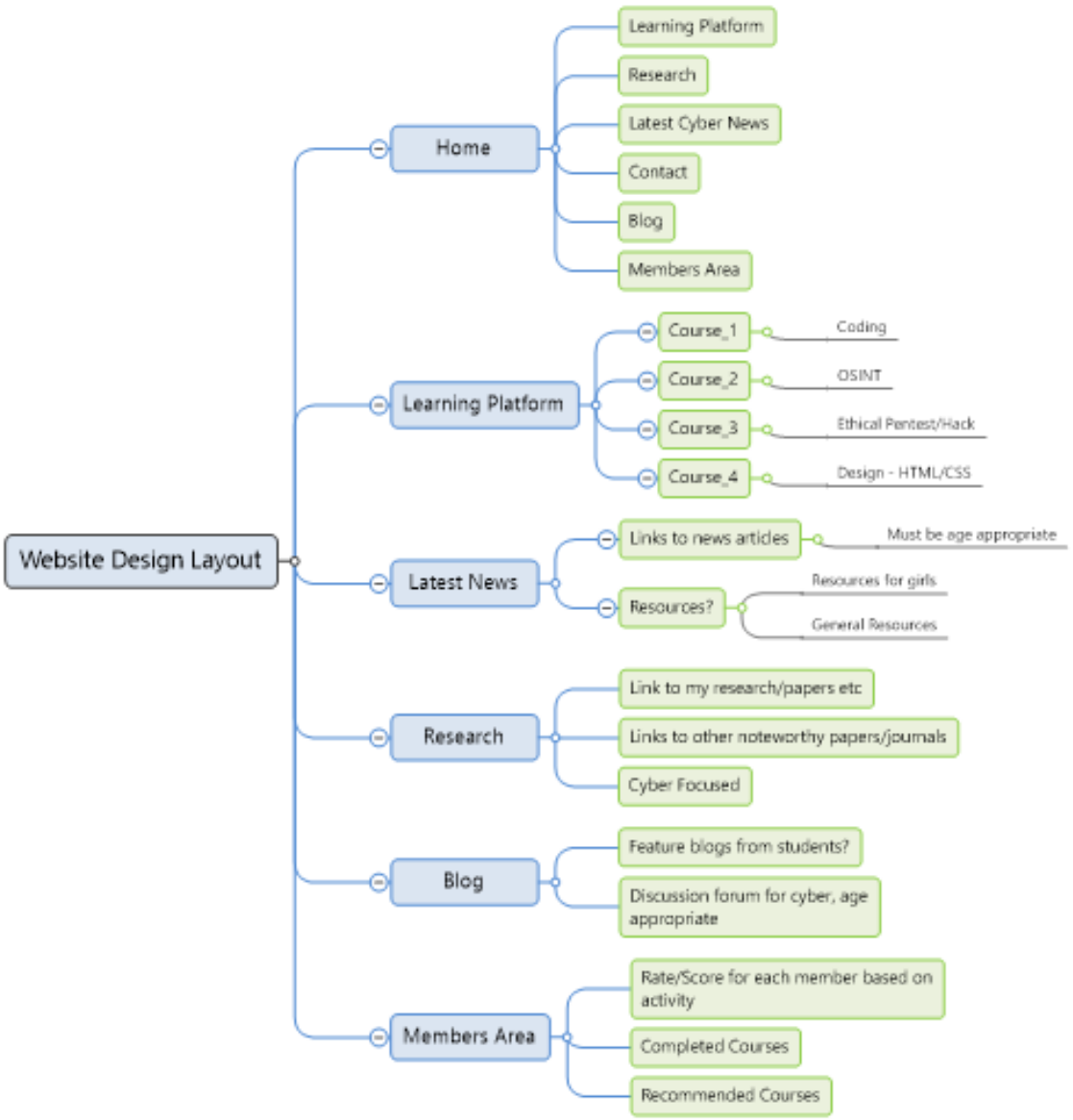


Figure.12.2. Database ERD Examples

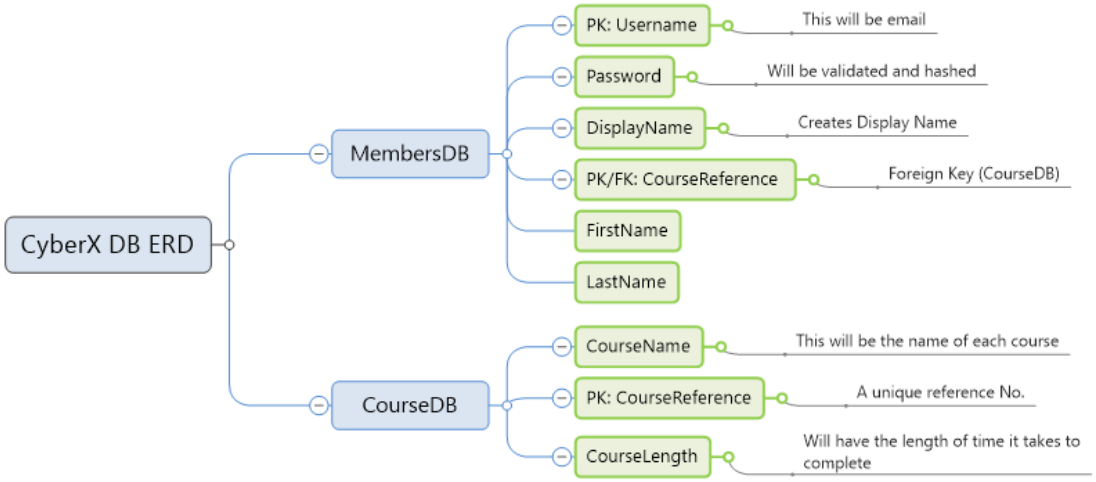


Figure.12.3. Back End System Architecture

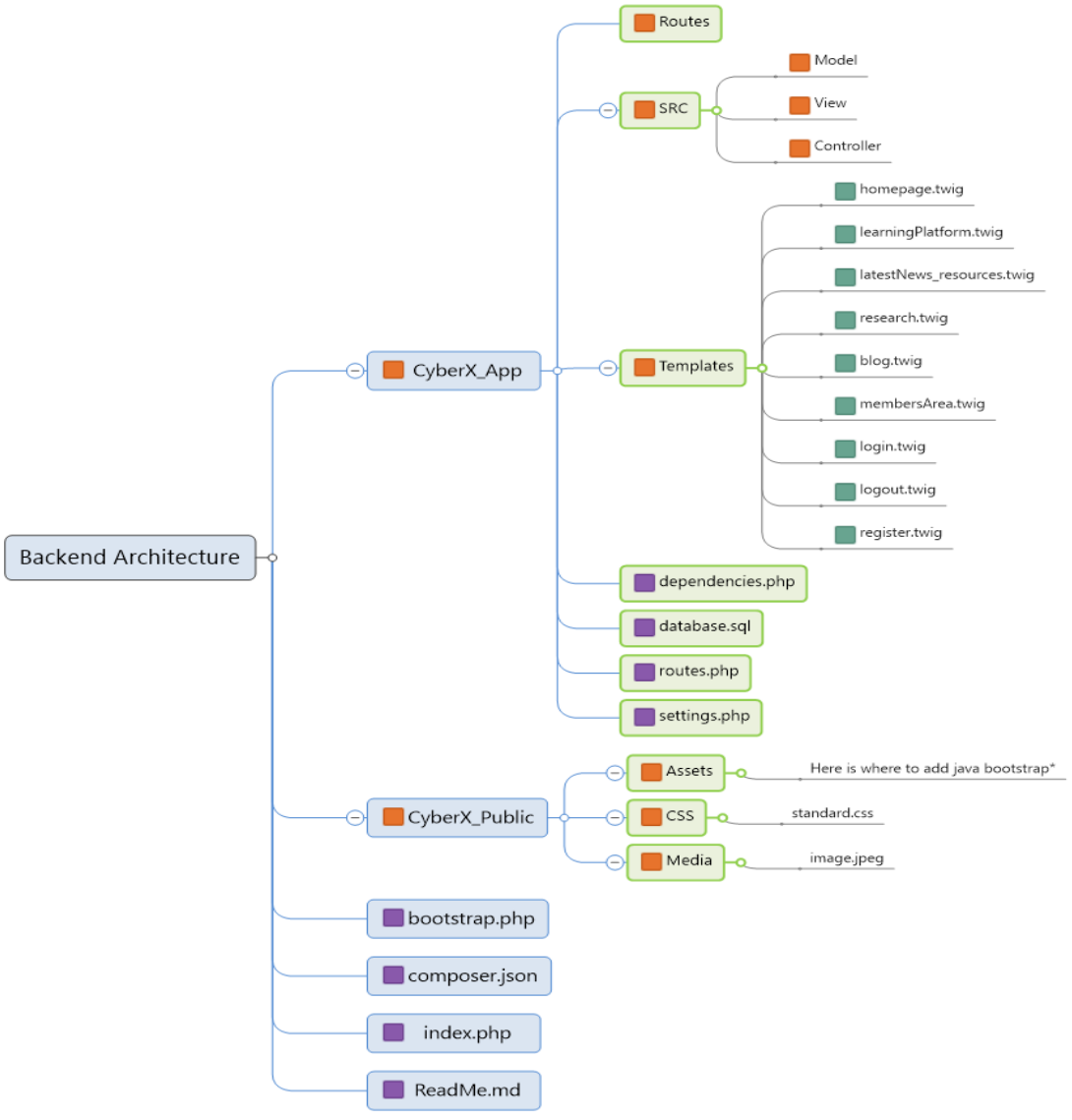


Figure.12.4. Visitor Journey

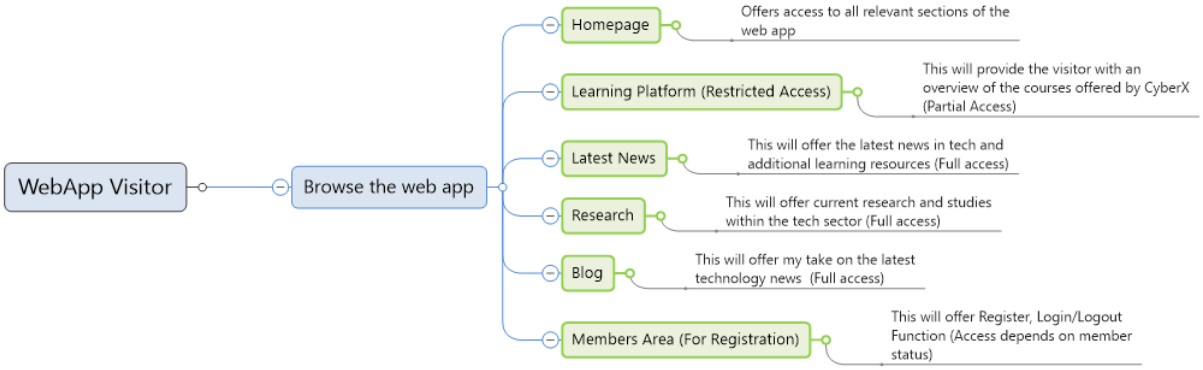


Figure.12.5. Member Journey

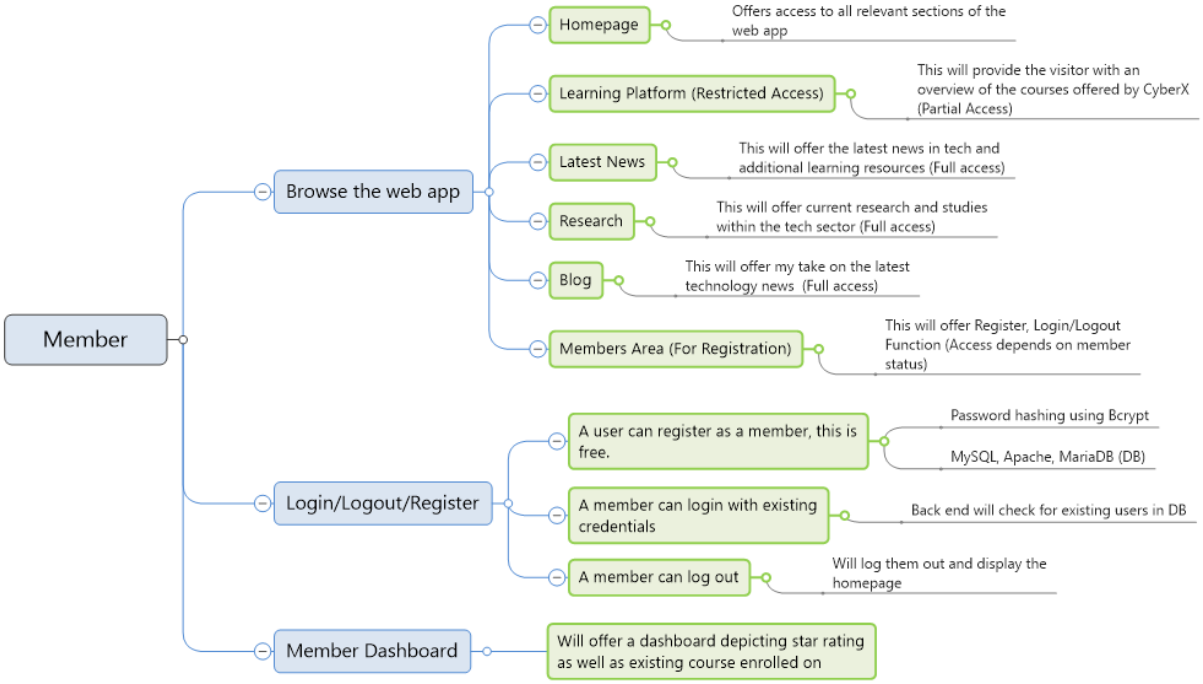


Figure.12.6. Admin User

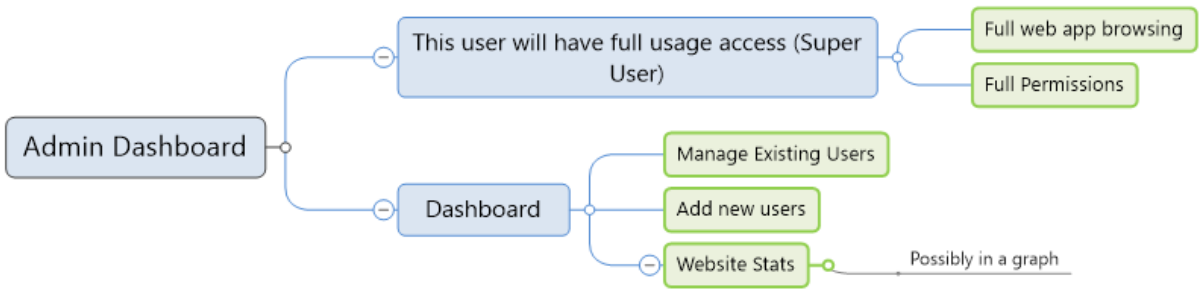


Figure.12.7. Diagram showing front end and back end

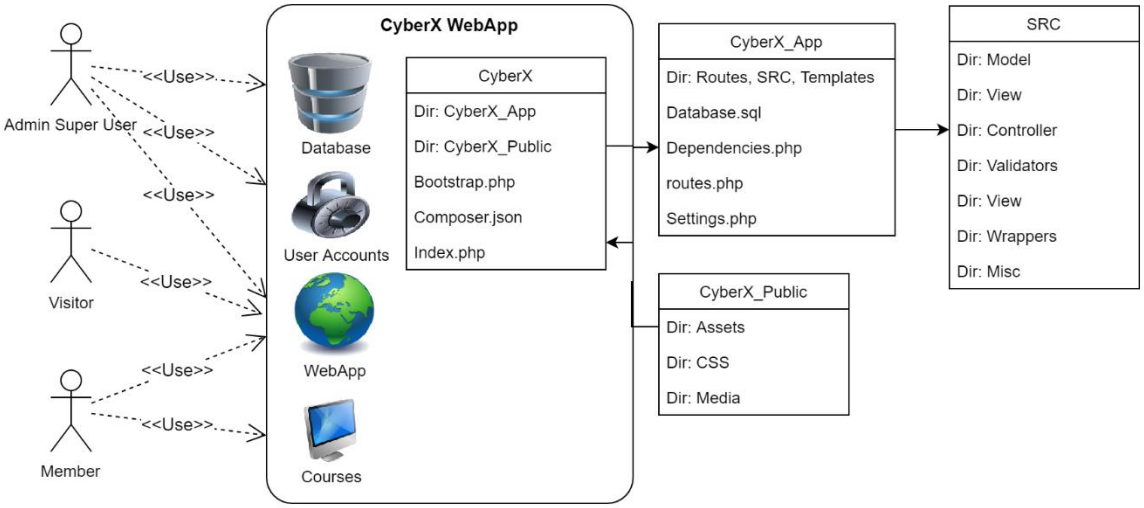


Figure.12.8. UML Depicting Access for CyberX Users