~~PHP (Hypertext Preprocessor)~~

~~This was used throughout the website, as the pages needed to read from and insert into our MySQL database hosted on the University’s server, this needed server side code in order to process. An example of a page would be GetItemInfo.php (index…) this is a page which enables the user to view an item’s information and request a loan. This page has a few examples of PHP and the rest is run from an Ajax call, but this page Loads in all the information about an item, this would usually be presented in in a way such as GetItemInfo.php?id=509 this tells the page to call the information related to the item with the ID of 509 as recalled from the database. Other PHP on this page allows the user to select a date, this is done by using the PHP date function, it then uses a while statement, this is done to count days, and adding one day using date to string function in PHP, This was done so it did not need to be manually done by the user or a programmer adding new dates.~~

~~GetItemInfo.php also calls other pages in order to work, these being, CheckBookingDate.php(index …) and InsertLoan.php(index …) these pages are only called when they are needed by using JQuery.~~

~~When using the page GetItemInfo.php and you select a day that you would want to book, the page will then use JavaScript to enable another select box where you can select how many days you would like to book it for, once this has been completed it will run a page called CheckBookingDate.php it uses ajax to send this info from the two select boxes, to run a script to check of the date chosen and the date that is should be returned is between anyone else’s booking, it does this by getting the day that is chosen and again using date to string to add the booking length to the booking day -1, it uses a -1 to count the day that the user books the item to allow them to return it the same day, as some items can only be booked for one day. Once this has been completed it will set information into hidden field that the user cannot see on the page. If the value is set to a specific value it will enable a booking button allowing them to book that item. If the item has already been booked out for the days booked, it will not enable the button and tell the user that it is not available for the days they have chosen .~~

~~Once the user has a booking that they are able to book between days then they can press the book button, this runs code to create a booking after doing some checks, it first checks to see if the user is allowed to book that item, because currently every user can see every item, this allowing them to view and attempt to book items that they should not, of course this could be fixed at a later date if needed be, but for now the checks also occur when the user goes to book an item, this happens within the InsertLoan.php this page gathers the other required PHP files and runs them in a logical order, this being~~

~~getDateInfo.php~~

~~This is a page that gathers the date from the Selection fields this being the date that the user has chosen to book the item as well as the day that user will need to return said item this is a small PHP file that process this information~~

~~CheckLoan.php~~

~~This is code that runs to check if the user is able to book the item, here it checks if the user has a user type to which they are able to book said item. If the user then passes these checks it will then run another PHP from here to insert the loan to the database. This being:~~

~~InsertSQLLoan.php~~

~~This is the page that adds the first part of the loan this uses an SQL statement to do this, once this is done it then runs another PHP page called~~

~~SelectCurrentLoan.php~~

~~This is a page to select the most current loan from the user as we then need this information in order to run another PHP file that finishes adding the loan for the user. It gather information about the first part of the loan that has just been created, the most important thing being the Loan ID as this is needed in the next PHP file~~

~~InsertLoanContent.php~~

~~This is the page that adds to the rest of loan, this gets the information that is gathered from the previous file SelectCurrentLoan.php as it needed to gather the booking ID to add it the Loan Content.~~

~~GetBookingInfo.php~~

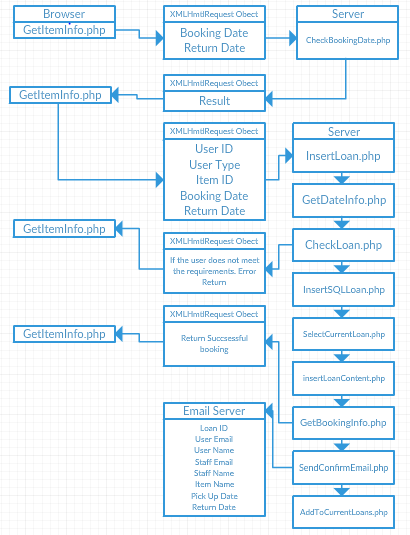
~~This is a simple page to get information about the booking to display it to the user later, as when the booking is a success we will give them a quick prompt to tell them this is information about their booking, it will also give SendConfirmEmail.php information that it can use to send an email~~

~~SendConfirmEmail.php~~

~~This is the page that sends an email to the User as well as the staff member who’s said item is being booked, this letting them know that it has been a success and gives them confirmation about their booking.~~

~~AddToCurrentLoans.php~~

~~The user should know how many items they have booked, this is a small PHP file that adds +1 to the current loans that the user currently has~~



~~Html 5.0 (HyperText Markup Language)~~

~~CSS 3 (Cascading Style Sheets)~~

~~CSS was used to style the webpages, this also allowed us to make responsive web design thanks to the power of CSS 3 and HMTL 5.0. Both of these together allowed the site to use a CSS command @media , this allows us to get the dimensions of the web page which was used to make the site look better on mobile devices with smaller sized monitors. Style is important on a webpage as it helps the user navigate from one page to another with elements of the page being placed in easy to find places, accessibility and ease of use was part of our goal for the system.~~

~~CSS also allowed the site to re-size the navigation bar for smaller screens as it turns in into a drop down menu rather than a list at the top of the screen. This takes up less space on a user’s device screen whist they navigate through the site.~~

~~JQuery (JavaScript library’s)~~

~~JQuery are a set of JavaScript libraries that has been implemented and used to make writing JavaScript easier, as they are exactly the same, and has been designed to simplify html. For this project we used version 3.1.0 from google library [1] [2]. We have used JQuery everywhere on the website, as it handles many things, these being:~~

~~Navigation~~

~~JQuery was used on the navigation rather than using the normal <a href=””> this was used because of how the site works. Instead of going to a new page the page that we want to the user to go to is loaded into an element that is on the page, this means that effectively the entire website is on the same page, of course this has its own benefits, as this allows us to hide pass variables from the Address bar on the page, if the user cannot modify the address they can’t access something they should not be accessing, for example if we choose to hide catalogue items from certain users then they cannot modify the address bar to find items that they are not supposed to, of course this can also be done using PHP, but if the user does not have to option to do it then it stops them from even attempting to do that. [image 1] of course the user can still see the id of the item when looking in the developer tools when looking at the networking traffic [image 2] of course if the user is able to access a page for an item they are not supposed to book, they won’t be able to do it anyway because when booking the item it also checks the users Information before they are able to book it.~~

~~Loading Pages~~

~~Much like the navigation we load every page onto one page, this allows the page to feel like it is loading much faster. This works together with the navigation in order to display the pages and for pretty much the same reasons, this works by finding the element that the user has cocked on, and then loading into another element a page they corresponds to the element clicked. Below is an example of how the code works.~~

~~$(document).ready(function() {~~

~~$(".addi").click(function() {~~

~~$(".holder").show();~~

~~$(".holder").load("ajax/Pages/Inventory/add\_inventoryImage.php");~~

~~}); });~~

~~Calling Ajax~~

~~We also use JQuery for calling AJAX and JQuery’s version of Ajax is much easier to use, it allowed us to gather the information from textboxes and selection boxes on the page and then set them in variables to be sent off to other pages, this allowed information to be transferred to other pages without having to be redirected.~~

~~Disabling/hiding and displaying elements~~

~~Of course using this allows the app to disable buttons and other items that are not needed or should not be used by the user at a certain time, there are a few examples of this, and one being on the administrative pages, and on these pages the website disables the textboxes and hides buttons that are not being used. When the edit button is pressed by the user it will enable the textboxes that are relevant to the one that is pressed, it will also display the save and cancel buttons as they would only be needed once they original edit button has been pressed, this is used as the website does not need to show elements and objects to the user that they don’t currently needed to edit.~~

~~Changing and creating ID’s and classes on HTML elements~~

~~This also works in conjunction with hiding and displaying elements. This allows the website to know what element needs what attributes, as if each element had the same attribute it can cause problems scenically when using ID’s as an ID can only be used once on the page. The website has been made to set the ID Attribute to an element only when it needs it, this allows the website to only collect the correct information.~~

~~AJAX (asynchronous JavaScript and XML)~~

~~Ajax is not a programming language, but rather a combination of JavaScript and HTML, it uses browser built-in XMLHTMLREQUEST to request data from the webserver, this allows the site to enter information into the database as well as gather information from the database, although AJAX is mostly used for XML we can also use it to call a PHP file, the page can see the gathered information in variables and send it as normal to a PHP file.~~

~~This works in 6 steps: [3]~~

~~1: an event occurs on the webpage, such as a click event, or a change event, usually on the webpage it would most likely be a click event, once this happens the information is gathered into;~~

~~2: An XMLHTMLREQUEST object, this would be created by JavaScript this is the information that the website creates in order for it to send the results to a PHP file, once this information has been gathered it will;~~

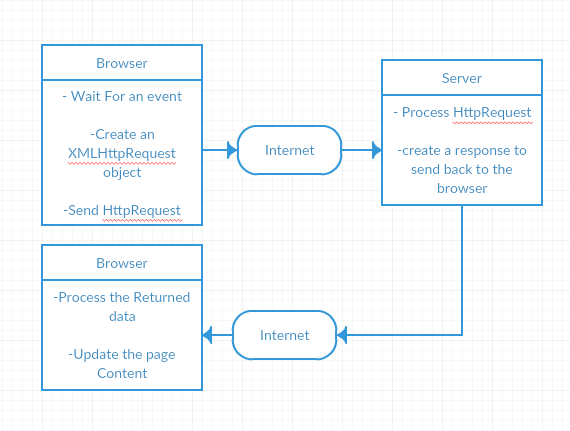
~~3: Send the XMLHTMLREQUEST to the web server, the data that has been sent will then;~~

~~4: Be processed by the server, in the case of BookIT it will use the variables that we have given the PHP file and handle a request with it. For example, it will check all the dates of an item that has been booked in the database, and then check the date that we have sent to it, then it will send a response back.~~

~~5: The response is sent back to the webpage, using Ajax means that the site does not need to navigate between pages, it allows the page to send information to another pages whilst allowing the user to stay where they are.~~

~~6: The response is read by the JavaScript, here the JavaScript decides what to do with it. We have programmed it to display or utilise the information that has been returned to it. For our date example it can disable and enable a button on the webpage, this being the button that allows the user to book the item or not.~~

~~This particular process is explained in further detail later in our report as it is an important process in the booking system.~~

~~~~

~~Ajax was used in the project as this allowed the pages to move data from one page to another without having to load multiple pages. We utilised Ajax all over the website, this was an extremely useful and powerful tool, especially on the page where we needed to check if the user could book the item. If we had done this with normal PHP they would then be redirected to another page or just receive an error when they tried to book an item that has already been booked for the selected days. As we could send this information and get a result back without moving from the page, it made it easier and faster for the user to navigate and see if they could book things or not.~~

~~Ajax is a good way to cut loading times, the site only loads particular data when it's needed, using Ajax means that we don't use a large amount of data, as the site does not need to load an entire page of code that does many things, as the information is only called to the page when it's needed and not it being loaded to the page without it being needed.~~

~~Ajax was a challenge as the team previously had no knowledge of this methodology, one member of the team (James) took it upon himself to learn about Ajax and then spent time with each member helping them to learn and implement it. It was a steep learning curve to adapt what we already knew into this new way of building our pages for some of us as we had already become comfortable with the way PHP worked but it was definitely worth the effort of learning. Our system has benefitted from the sleek way it loads data and our team has benefitted from learning a better way of building a site. Ajax has also benefitted our project in a security aspect; a user cannot ‘bookmark’ a page and return to it at a later time without logging in via SSO, where a php page would display a specific address that can be stored Ajax only ever displays the ‘front’ page of the site, the user is never informed that a page address has changed.~~

MySql (databases)

Mysql was used to

# Bibliography

|  |  |
| --- | --- |
| [1] | "Google Hosted Libraries," Google, 17 January 2017. [Online]. Available: https://developers.google.com/speed/libraries/. [Accessed 2016]. |
| [2] | "Jquery," Jquery, [Online]. Available: https://jquery.com/. |
| [3] | w3schools, "w3schools," w3schools, [Online]. Available: https://www.w3schools.com/xml/ajax\_intro.asp. |

Appendix

Image #1



Image 2

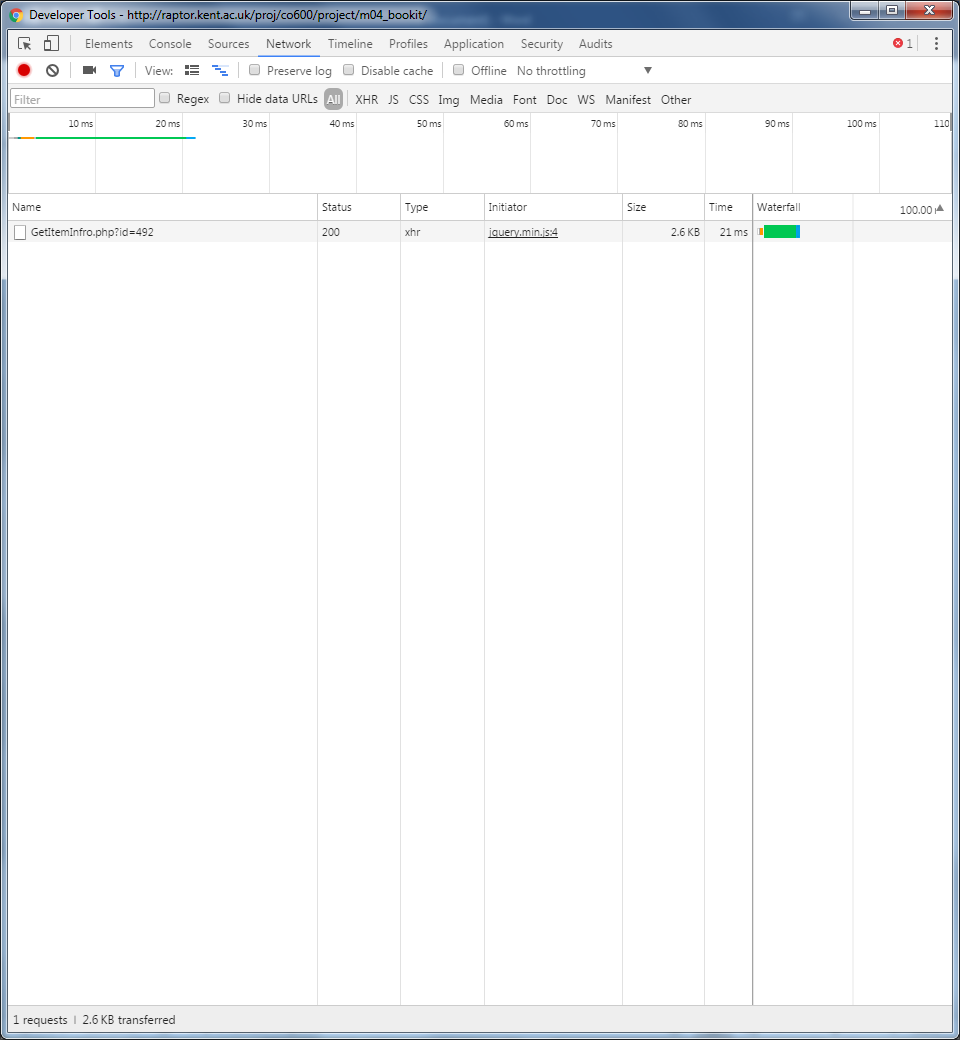


Image 3

