Greg Harrison – 0100603

Web Dev CMP306 exam 12/1/2021

1. **In a site where you are required to retrieve data from a database, there are many formats in which to deliver the data from the API to the site view. Indicate how you would return JSON rather than a MySQL query and give two significant advantages of using JSON in this situation.**

In a site that is required to retrieve data from a database using JSON is a good option to deliver the data to the view page rather than just sending the result of the mysqli\_query. In an MVC framework we can have api’s in the model that will send the required query to the database and encode it to JSON format it using json\_encode($var). This is a useful function as it can encode result of the query no matter what type of variable or array it is. It is also human readable so is useful during development when checking functions are working an data is being passed correctly (eg the encoded data can be used with var\_dump($vae). On the view page the json that has been returned in the function the JSON string can be decoded using json\_decode($var) and then the data that has been retrieved from the database can be echo’d out to view using $decoded\_var -> databaseField.

The advantages of using JSON over just returning the sqli\_query() are that by using an independent format you can change the way the database is structured and still use the same interface. This can save lot of development time when changes are required. JSON is also easy to use. The -> syntax is simple and quick and makes parsing the data easy. This small simple syntax help JSON excute responses quickly and ensure it is quick no matter what the file format is(eg images or audio files).

1. **SQL injection is regularly cited as a main source of insecurity within a database driven web site. What is meant by SQL Injection and how can this be stopped.**

SQL injection is when a hacker inserts SQL code into a form that they know will be stored in a database. This code, when retrieved will be ran, which will give the hacker the information from the database that they are looking for(ie what the code they injected asks for). Depending what is stored in the database the hacker could retrieve banking information, usernames, passwords etc, all of which would be dangerous to lose. For this reason passwords should always be hashed using a secure algorithm and salts to add an extra layer of security.

One of the main methods of preventing SQL injection is to sanitise input that the user has input into a from that will be destined for the database. The PHP solution to this problem is to use the mysql\_real\_escape\_string() function to ensure that the code is safe when inputted into the database. It does this by removing characters that when ran in a query could be malicious and used to change the query that is intended. The other method to prevent SQL infection attacks is by using prepared statements. Prepared statements add security when putting data into the database as the query and the data is sent to the database separately. The problem with SQL injection attacks is that the code and the data is mixed. By sending these separately this helps to prevent the hacker from getting the information back form the database that they are looking for.

1. **There are many companies who sell IoT devices and as part of the deal they will store the state of these devices for a particular user. Each user will have a different set of IoT devices. How would you store the state of the various devices for each user?**

There are different options that could be used to store the state for internet of things devices. One of the selling points for many IoT devices is that they can all be linked, allowing users information on various things on one interface, for example a website. To allow this to happen the data must be stored and linked together in such a way than it can be retrieved from the database and output to the user, or the user be alerted when a device it in a certain state, for example, a smoke alarm being triggered in their home.

The best option in my opinion to store data for users who have multiple devices which is becoming more common as people move to having “smart homes” so devices such as home security, lights, audio systems, garage doors etc can all be smart and remotely changed and viewed is to have separate tables in the database for each device and a table of ‘users’. There will also need to be a link table that has the users unique id linked to the devices unique table id.

In the table for the device the state would be stored as a JSON string and that will also add flexibility to add additional fields in the future – for example, the security camera gets an upgrade and starts monitoring brightness levels, or another series of data can is stopped being recorded. The state could be a different format depending on what the device is programmed to do, the state could be a Boolean value if it is an on/off switch or the state could be a int if it is a temperature reading. The state would be stored in the JSON string and decoded by an api to the customers view that will allow them to see the state of their devices.

1. Yr.no are a Norwegian weather site that will provide an accurate weather forecast for a city, anywhere in the world in XML format. (<https://www.yr.no/place/United_Kingdom/Scotland/Dundee/forecast.xml>)

How would you display the XML onto your page?

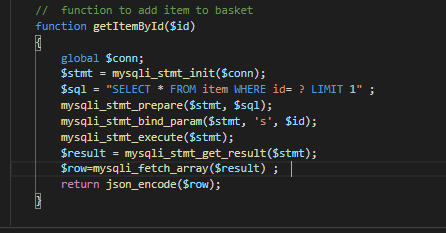
To view the XML weather data on my page I would first create the structure for the XML. After this I would create an XSLT page for the XML data to go into. The XSTL page will contain columns for time, wind direction, wind speed, temp and pressure. This will give users lot of easy to read data and will loop through and fill it for the amount of time slots there are on the XML page using a <xsl: for-each select=time> XSLT give an easy way to merge XML data into a readable format on a HTML webpage. The yr.no…….xlm file will then need to be attached to the XSLT page then the XML.

https://mayar.abertay.ac.uk/~0100603/cmp306/exam

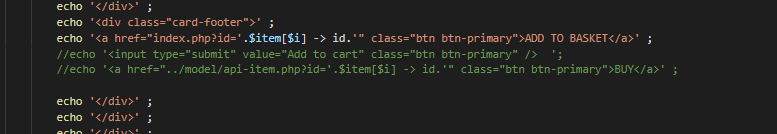
1. For the task there was various different method that I considered using. I thought that either using Sessions or cookies would be a good option as this would store what the user had in their basket and hold in for a set period of time. The other opinion that I considered was when an item was added to the basket to add that to a table in the database then the basket would show those items. In this case, as only one item was allowed to be in the basket at once I decided the best option was to have a function in the model/api-item that gets the item by $id and displays that in the basket. In the real life scenario the BUY button in the basket will transfer the user.



Code that displays the basket item.



Function that gets the item from the item table in the database to add to the basket.



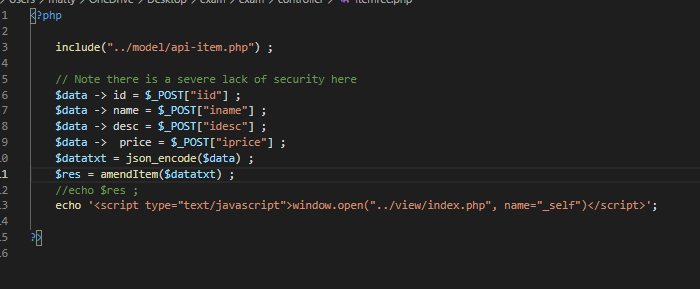
Code that takes user to index page that adds item to basket with passing $id= with the GET method.

The above code works by passing the item id via the get method. When the user click on this the new index page loads up with the id of the item as a parameter. From her the getItemById() function takes the id of the selected item and displays it in the basket. This method would not be possible if more than one item was required to be in the basket at a time but in this scenario the method was the best option. I changed the way the basket was laid out from in the template to have the basket as full width and have added a BUY button.

1. For this task once the user enters the id number that they want to amend and this is passed to the getItemToUpdate.php file in the controller. This passes the id of the number back to index where the fields are populated ready to be altered as shown below. The GetItemToAmend() gets the data to fill the fields and the user can easily amend from here.



This “amended” data is passed from the form into the controller where the data is encoded and passed to the model where is written in an SQL statement to update the database.



****

Another way of doing this would be to have the update form on a different page as when an item update is requested this removes the item from the basket.