Introduction

My project is an online examination to test year nine and ten students for technology subjects. I created this for my client, Terry Hawkings, a technology teacher at my school. This report critically analyses the technological practice I undertook to develop my solution.

I analysed by project and techniques in undertaking the project throughout the year, with weekly entries in my visual diary and ongoing planning. I made two PowerPoint presentations regarding my project: one at the start of the year to my class, the other once I had finished my project. The latter presentation was given at parliament to industry professionals, other technology students and The Honourable Marion Hobbs (who hosted the event). I have included with my submission a CD with the presentation in Mpeg format, as well as the original PowerPoint document. Many aspects of the presentation will overlap with this report, but the presentation was limited to five minutes, so I did not have the time to go into depth about the finer points of my project and techniques.

I have also included all my PHP code and MySQL database information on the CD for further evidence of my project.

The Issue

My client wanted an online testing solution to avoid the large amounts of photocopying he has to do in order to test a class. He was concerned with students cheating, which was very easy to do because all tests given out were exactly the same. Another issue was the time it takes for teachers to mark tests and the delay for students to view their results. In other solutions he looked into before I undertook the project, the correct answers were often visible in the source code of the page (craftier students can cheat very easily), so he needed something personalised for his needs.

My solution is made to be restricted to Wellington High School (WHS) technology students, so expansion was not a key issue. This was kept in mind, but the nature of my coding allows quick and easy change. Essentially, it could be used to test anyone for anything, but I kept my focus on the direct needs of my client.

Page design was not a critical aspect of the project. The only specifications given by the client are that it should be very simple and involve bright, positive colours. Other specifications I made for the project are as follows: made for 1024x768 resolution as that is the default on WHS workstations, compatible with Internet Explorer and Mozilla Firefox, file sizes do not need to be minimized as it is all on a LAN which provides a very fast and reliable transfer of data.

Copyrights are not needed. All the data is kept on a secure server so the only thing that can be 'stolen' is the HTML coding which the host can view, which would not provide anybody with a working product as all the 'working parts' are in the server-side PHP code and SQL database. Also this is strictly non-profit and will most likely not be seen on any screens outside WHS.

For this technological practice I am developing this through to a final release, where it will be available for use for the technology faculty of WHS.

What is a database?

A database is a file or set of files used to hold vast amount of information in an organised manner. They range from simple flat-file databases (single table – eg. Basic MS Access databases) to normalised relational databases (many tables which interact). My database is relational, but it is not fully normalised.

Normalising is the process of splitting a database into more tables to avoid data duplication. As seen in page 23 and page 24 of my visual diary, I did change my database throughout my project. However complete normalisation, as I have been told by my teacher, requires third year university experience in database design.

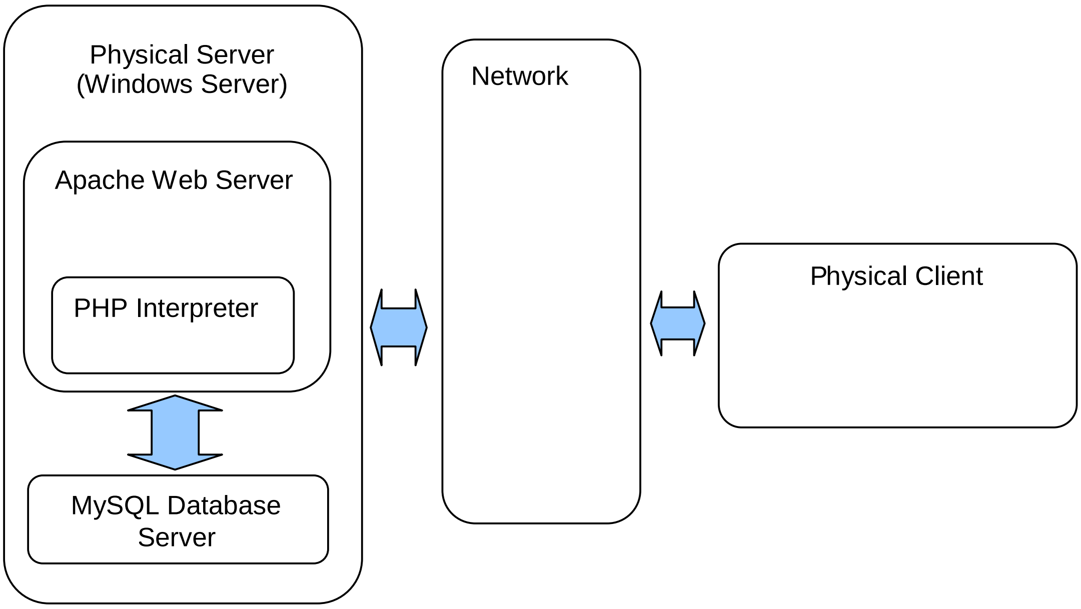
There are many different database solutions available. The most common databasing language for exchange over the Internet is SQL (Structured Query Language), it has been standardised by ANSI (American National Standards Institute) and ISO (International Standardisation Organisation). SQL was originally designed as an object-oriented relational database language in 1987, but over time it has evolved to much more. Different vendors have released their own SQL solutions, such as: MySQL, PostgreSQL and MicrosoftSQL. I chose to use MySQL because it is open-source and free to use and distribute. It is also arguably the most common SQL solution available and it has been packaged to be used with other free distribution server software, such as Apache server, FileZilla, PHP, etc. It also incorporates a web-based solution to simplify future modifications of the database, such as expansion, in PHPMyAdmin. I used PHPMyAdmin very frequently throughout the testing phase of my project, because I can quickly see if data has been inserted into the database.

Some limitations of SQL are:

- The language is often complex when using long strings of commands, it is often compared to lower level languages
- Does not *fully* meet the standards of relational theorists to be considered a relational database
- <MySQL only> - does not meet the ISO standards for default values (treats them differently)

Despite these limitations, there are too many benefits to list which far outweigh trivial downsides. Some well known users of MySQL are Yahoo, Google, NASA, Nokia, Wikipedia.org, Amazon.com and Apple Corporation.

Server



This diagram shows the client/server relationship involved with my project. The server runs Windows and can only be accessed from inside Wellington High Schools network. The server uses XAMPP which has an Apache web server with a PHP interpreter inside it, which runs along-side MySQL server.

All requests for information from the client run through the network to the server. The server then processes any requests and sends relevant information back to the client through the network.

PHP

What is PHP?

PHP is a hypertext pre-processor, which means that it dynamically creates web pages based on variables. It is an open-source server-side scripting language and is free to use (it is also a recurring acronym, as the first P stands for PHP (some would argue it stands for Perl). It is designed around a CGI (Common Gateway Interface) which allows it to pass information from a web browser to an application for processing. Essentially, when a client requests a PHP document the server executes the code in the document, and then sends the results to the client. It has in-built support for SQL queries and is especially designed to run along side it. Much like SQL, it is object-oriented and relational. PHP allows writers to use both static, pre-defined tags and dynamic tags which can be declared at any time. This makes it a very flexible language and explains why over 23 million domains use PHP(source).

Below I will explain how I met requirements of the brief through use of PHP.

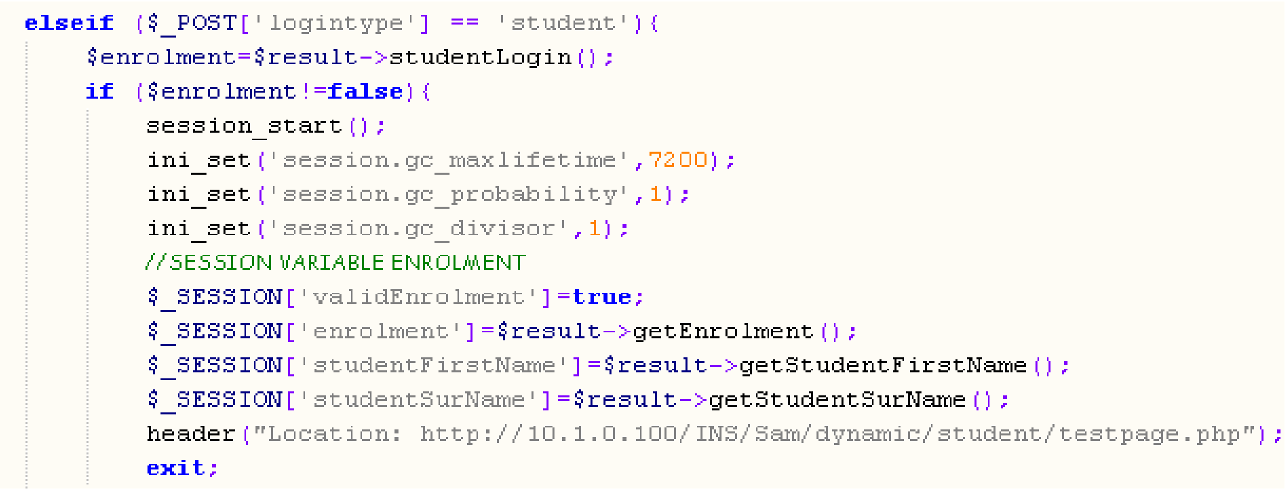
QuoteSmart

The quoteSmart function is a security feature of PHP. It is used when text fields are being stored in a SQL database. SQL can be exploited by entering SQL commands into the text fields, when the data is stored it is put into an SQL command, so any commands inside it would usually be executed. For example "TRUNCATE table[information.schema][sam] 'results'", if entered could empty all results stored in my database. The quoteSmart function searches through forms and strips out any SQL commands before posting it to the database.

Login

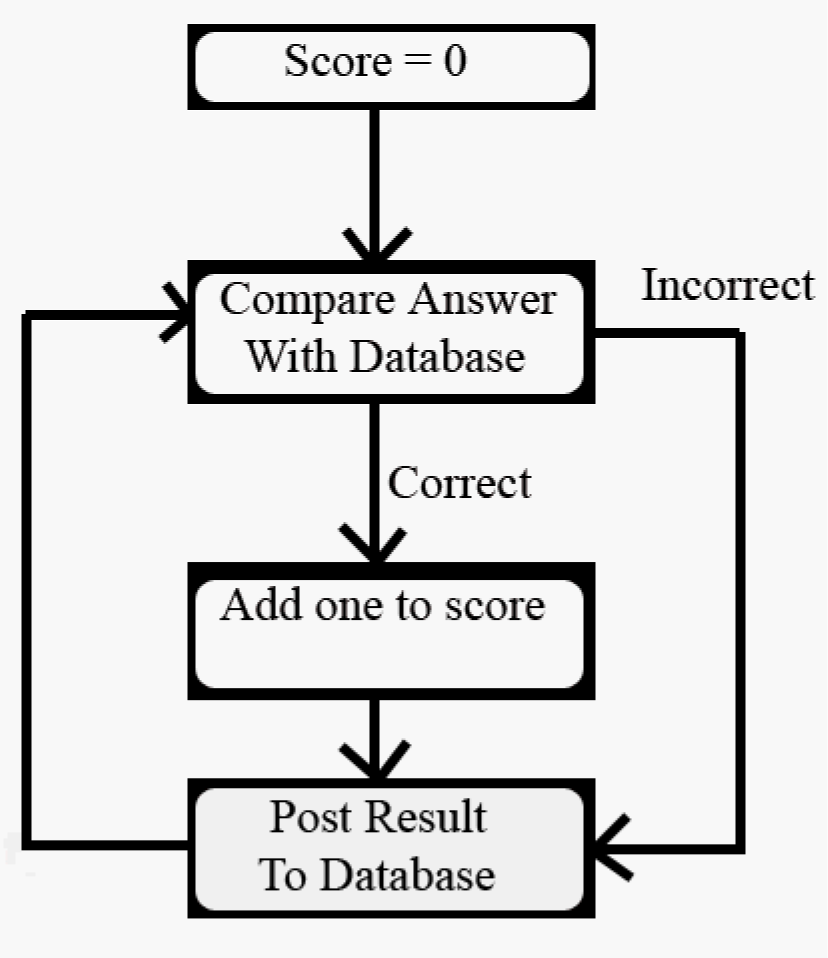
The client selects "Teacher" or "Student" from the pull-down menu and enters their login name and password, which, once posted, are queried on the corresponding table of the database. If the number of results that come back is equal to one, the login is successful. Session variables are added, declaring them valid users and they are re-directed to their corresponding section. If the number is not equal to one, the client is re-directed to a page explaining that they used an incorrect login and offers the same login form so they can retry.

If the user is a student, more session variables are added: enrolment number for authenticity and for use as a unique identifier on the results table, and first name and surname for authenticity. The first name and surname are displayed with the enrolment number on every page they view, so an examiner can check it against the students school ID (authenticity in this case means trying to stop cheats). A session lifetime is also set, defining a maximum time to sit the exam. After the session lifetime expires, any activity will re-direct the student to the login page.



```
elseif ($_POST['logintype'] == 'student'){
    $enrolment=$result->studentLogin();
    if ($enrolment!=false){
        session_start();
        ini_set('session.gc_maxlifetime',7200);
        ini_set('session.gc_probability',1);
        ini_set('session.gc_divisor',1);
        //SESSION VARIABLE ENROLMENT
        $_SESSION['validEnrolment']=true;
        $_SESSION['enrolment']=$result->getEnrolment();
        $_SESSION['studentFirstName']=$result->getStudentFirstName();
        $_SESSION['studentSurName']=$result->getStudentSurName();
        header("Location: http://10.1.0.100/INS/Sam/dynamic/student/testpage.php");
        exit;
```

My Marking Mechanism



This loop was created using the 'foreach' command. To explain my code in English: The score equals zero. For each question, compare the value (answer selected) to the 'correct' column in the 'qa' table. If the are the same, increment the score by one and post '1' along with the questionID and enrolment number to the 'results' table. If they are not the same, post '0' along with the questionID and enrolment number to the 'results' table.

So the score is how many questions were answered correctly, which is displayed instantly to the student. A percentage score is also included (score/total questions*100).

Sessions

At the beginning of my coding, I basically had three options of how to store data: I could store them on the server as session variables, store them on the client as cookies, or store them in the web page as a hidden element. A savvy user could look at the source code and find any hidden data, so that idea was over almost before it began. Cookies are unreliable in cases where privacy conscious users turn them off, however they take some load of the server so they are preferable. In the end, I weighed up the pros and cons of server-side vs. client-side sessions, and decided that server-side sessions were the most ideal because of the zero-risk factor of dropped logins. Below I will list and explain the session variables I used.

- ValidTeacher – When a teacher logs in, the value is set to “true” - any 'teacher pages' require the value to be set “true”. If it is not, the user is re-directed to the login page. This is to restrict access to sensitive areas of the site. If an unauthorised user gained access to certain pages, they could ruin the integrity of the database and basically create a headache for the person who maintains the project.

- ValidEnrolment – Performs the same task as validTeacher, but for the student section
- Enrolment – The students' enrolment number is requested from the database and saved as a session variable. The enrolment number is a unique identifier, so when the results are posted to the database, there is no confusion. Also used for authenticity purposes (see presentation).
- StudentFirstName, StudentSurName – There was a problem when I tried to put both fields into the same session variable, so I overcame the complexity by requesting them one at a time with different functions and saving them as different session variables. Then I simply fed them out on the page side-by-side. This is used for authenticity purposes (see presentation).

Classes

I discovered inefficiency in my early design. I was using all my PHP files as single entities in that they did not interact with each other. This meant that data was being duplicated across PHP files. I solved this problem by creating class files:

- result.php contains all MySQL queries, stored as functions
- page.php contains the HTML tags, the header and title image, and CCS and DTD references (all of which are used on every page). Anything that needed to be displayed that was unique to a certain page could simply be wrapped in <h1> or <h2> tags etc, this made my page production more efficient.
- form.php contains all input forms (add a student, add a question)

By creating a modular design, I have allowed future changes to be simplified. My code is quicker and easier to interpret, and I have eliminated unnecessary repetition of my code.

Encryption

I had a few different options for encryption. Encryption would be used for sending of login information because this is the most sensitive information sent in my project. When data is sent over a network, packet sniffers can be used to read whatever is sent, it is made especially easier if it is sent in plain-text. In the earlier days of web development, encryption was a very difficult mechanism to employ. This was partly due to a lack of secure encryption algorithms and the complexity of early languages.

Message-Digest 5 (MD5) is an algorithm that securely converts any plain text into a 128-bit 'hash value'. The algorithm has *no known reversal*, so essentially it is 'hack-proof' i.e. any computer which reads the hash value could never turn it back into its' original form. This makes for a very secure password system. This kind of encryption is commonly used to verify files and store passwords. The way in which I have written my PHP code allows a developer to simply place 'md5' in front of the data fields (which lie between brackets) that they want to be sent as MD5 encrypted hash values, rather than plain text. As mentioned earlier, all database transactions are stored in 'result.php', so the fields can be quickly found and changed as needed.

HTTPS is similar to HTTP, except it uses TCP port 8080 rather than 80. This port number indicates to hosts that SSL 3.0 (Secure Socket Layer) encryption is in use, so all data transferred has 128-bit encryption. This type of encryption is not as secure as MD5, as it can be reversed to its original form, given a unique key. It is commonly used for secure logins, such as online banking, and is detected by all reputable browsers. However it results in noticeably slow connections, even on an internal network.

I decided not to use any encryption, as it was not necessary for the purpose of the solution. The solution is made for use on WHS's internal network. No users have administration rights to install files such as packet sniffers or key loggers so there is no threat. Although I have designed my code so that it will be very easy to include encryption if the site was moved onto the public internet for wider use.

Usability

Usability was a very important aspect of my project. This project was made for information science, rather than digital media; therefore I did not allocate much of my time to the design of the interface. Rather I concentrated on the engineering side of the project, for example my database, as this is what is expected for that subject. However the client expressed certain needs concerning the design of the page (refer to visual diary).

Scott Abernathy pointed out the importance of usability in his visit to our class. The concerns of my client and advice of a registered engineer prompted me to look further into usability. I stumbled upon Dr Jakob Nielsen's website and found an abundance of resources. Nielsen is a world leading expert in the field of heuristics and one common element he suggests throughout his documentation is simplicity. I also kept in mind the success and exponential growth of Google, which has the simplest design out of any major 'dot coms'.

I used Nielsen's heuristic evaluation model while designing my pages. I later used it as a testing mechanism, but I will discuss this in further depth in the “Testing” section of this report. Below is an extract from useit.com explaining heuristic evaluation.

> “Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the usability “heuristics").
>
> In general, heuristic evaluation is difficult for a single individual therefore it is recommended to use three to five evaluators since one does not gain that much additional information by using larger numbers.
>
> Heuristic evaluation is performed by having each individual evaluator inspect the interface alone.
>
> The evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognized usability principles (the heuristics). A sheet is provided so the evaluator can make notes.
>
> The evaluators decide on their own how they want to proceed with evaluating the interface. A general recommendation would be that they go through the interface at least twice. The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.
>
> Since the evaluators are not using the system as such (to perform a real task), it is possible to perform heuristic evaluation of user interfaces that exist on paper only and have not yet been implemented (Nielsen 1990).
>
> The designer can collate the results and make adjustments to the interface for revaluation.”

CSS

Cascading Style Sheets are used to format HTML in order to add colour, font, effects, etc. They can be embedded internally into an HTML document, or they can be external .css files, referenced in the header of the HTML document. I chose to use an external CSS document, rather than embedding the code in the HTML for a few reasons:

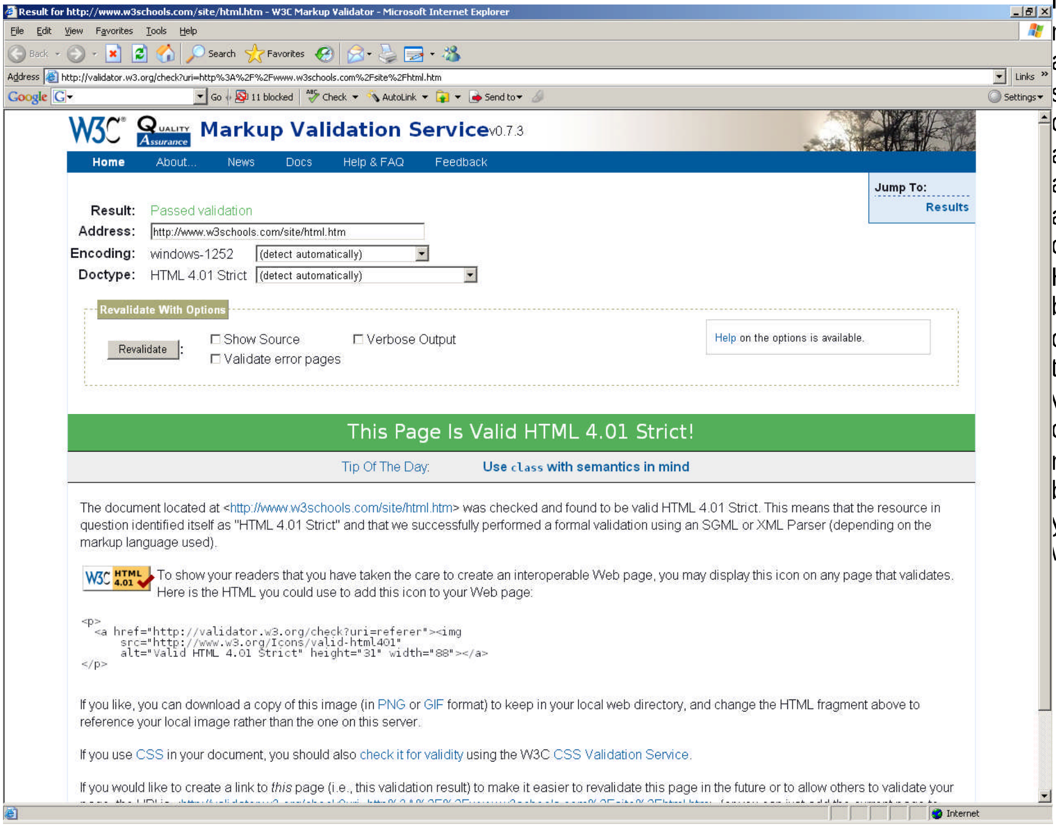
- My page class file could reference to it easily
- Less duplication of code
- The server can produce it quicker, as it doesn't have to scan it for PHP commands before sending it to the client
- It was easier to validate[v] the two codes separately
- A digital media student, or web page designer could come in at a latter date and edit the CSS file without touching the PHP files (future improvements)

Validation

When developing my prototype, I used the W3 schools validators to ensure valid code. I used essentially the same HTML in my PHP code, with minor additions where needed. Later in my project I did not validate my code as there was simply too much of it to strip out of the PHP, however I did make sure that all code was compatible with Microsoft Internet Explorer and Mozilla Firefox as these are the most common browsers in the world today. They are also the only browsers available within the WHS network.

Validators work by checking a file against its DTD (Document Type Definition) file to make sure there are no errors. The most common errors are in the syntax of code and in the use of tags. There are strict guidelines for each mark-up language as to how the code should be written. Reference to a DTD file should be contained in the header of an HTML document.

Below is an example of a validated HTML document:


Result for http://www.w3schools.com/site/html.htm - W3C Markup Validator - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Links
Go 11 blocked Check AutoLink Send to Settings
Markup Validation Service v0.7.3
Home About... News Docs Help & FAQ Feedback
Jump To: Results
Revalidate With Options
Revalidate
Help on the options is available.
Internet

Address: http://validator.w3.org/check?uri=http%3A%2F%2Fwww.w3schools.com%2Fsite%2Fhtml.htm

Result: Passed validation
Address: http://www.w3schools.com/site/html.htm
Encoding: windows-1252 (detect automatically)
Doctype: HTML 4.01 Strict (detect automatically)

☐ Show Source ☐ Verbose Output
☐ Validate error pages

This Page Is Valid HTML 4.01 Strict!

Tip Of The Day: Use class with semantics in mind

The document located at <http://www.w3schools.com/site/html.htm> was checked and found to be valid HTML 4.01 Strict. This means that the resource in question identified itself as "HTML 4.01 Strict" and that we successfully performed a formal validation using an SGML or XML Parser (depending on the markup language used).

To show your readers that you have taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the HTML you could use to add this icon to your Web page:

```
<p>
  <a href="http://validator.w3.org/check?uri=referer"><img
      src="http://www.w3.org/Icons/valid-html401"
      alt="Valid HTML 4.01 Strict" height="31" width="88"></a>
</p>
```

If you like, you can download a copy of this image (in PNG or GIF format) to keep in your local web directory, and change the HTML fragment above to reference your local image rather than the one on this server.

If you use CSS in your document, you should also check it for validity using the W3C CSS Validation Service.

If you would like to create a link to *this* page (i.e., this validation result) to make it easier to revalidate this page in the future or to allow others to validate your

I wanted to validate my HTML to create a standards-based site that will be compatible with any browser. It took a long time to learn all the strict conventions of HTML and CSS, but I got the hang of it. The good thing about validators is you can learn from your mistakes quickly, because they tell you where you went wrong.

The Technological Process

Finding a client

Initially, I had a choice of four clients who each had a need. Obviously I could only choose one of these needs to undertake for my project.

Matt Easterbrook wanted a solution to improve students understanding of biology using interactive multimedia. It could have been either a website or a program. However I chose not to undertake this project because he wanted the project to be used by year 12 and 13 students, which suggests a prerequisite understanding of biology. As I have never done biology beyond year 11 core science (and was never any good at it anyway), I decided that I am not the right person to be teaching the subject.

Calum McGonigle was worried about increasing concern from parents regarding their student's academic progress. He needed a solution to keep parents up-to-date with student and class progress, showing absences and milestones that the student as met, as well as test results.

Tony Hickmore wanted a database to store client contact information as well as keeping records for orders and payments. He found over time that his customers were not always being followed up, and that payments due were not recorded properly. Although this would require a complex database, I could not see it taking up a whole term of solid work due to the simplicity of the overall project.

I decided to undertake the project outlined on page 2 of this report because it involved database design as well as dynamic web design, which I was interested in gaining more experience in. Also I did not have to understand the curriculum as I could provide a method in which teachers could add questions easily.

Researching Existing Solutions

One of the courses I do at WHS, CCNA, involves online testing. It is run by Cisco Systems, which are a very large and reputable company (83rd largest corporation in USA). I learned a lot in the way they do things, although I took a slightly different approach. They use JavaScript/JSP and each exam is identical. I use PHP and each exam is random. Although I did follow their suit in using all multiple choice answers, as well as returning a mark instantly to the student (which was also requested by the client).

WintecActive teach and test high-school level computing skills online. My teacher, Vince Brannigan, recommended that I look into this. I did not use much of what I learned from this site. They use flash for much of their activities and I chose not to go down that road. There are long loading times associated with flash files as well as waiting around for load and play, especially on some of the lower-end PCs at our school.

Hotpotatoes is a different way of approaching the issue. They have programs in which you can create tests, cross-words, word-finds and all sorts of other activities. Although it is quick and easy to use, there are many vulnerabilities in their methods. Answers are clearly visible in the source code of a document, which can be read from within a browser. Also the results can not be posted to a database, unlike PHP and JSP.

Initial Plan

I made my initial plan based on several factors: the time I had available, the amount of research required and the amount of time expected for the project to be complete. I used a Gannt-style chart (but made in excel) to allocate my time efficiently throughout the year in order to maximise my level of output in regard to the project.

I managed my time well and met all of my targets on time. Because of this, my revised plan hardly differs from my initial plan. The only thing I did not expect is the amount of client interaction required. The timeline in my visual diary shows the extra meetings shaded in with pen.

Stakeholders

I separated my stakeholders into two groups:

Primary stakeholders are the people who the project is designed for – the people who will be making use of it. My client is the most crucial primary stakeholder as all decisions for the project come directly from him. If my client doesn't like the project, it has failed. The WHS technology faculty and students are also primary stakeholders as they will be making use of the project. The primary stakeholders are paramount as they provide opinions, key factors, needs and opportunities which shape what the resulting solution should be.

Secondary stakeholders are those interested in the project, but will not use the project. I deemed it unimportant to enquire secondary stakeholders regarding my project, but I acknowledge that their values and ideals do have an effect on what I would include or omit in the solution (but only at extreme ends of the continuum). For example I was careful not to conflict with cultural and social values of parents, and indeed the general public. In this case, the main secondary stakeholders are the parents of WHS students using the exam.

Key Factors

I prioritised and justified my key factors as seen in my visual diary. This allowed me to allocate resources more effectively. The key factors I initially chose ensured an *ethical* practice that, most importantly, appealed to stakeholders and was ready on time.

Time was a major issue as I only had 11 weeks available for implementing the project. As time goes on, key factors are bound to change – I had to be flexible to any changes, and work around them. During the first few weeks of implementation, the server my class was using was down. Even before that, it took a long time for the school to dig into the budget to purchase a server, which slowed our learning of PHP and SQL as we could not do practical work of our own. Once it arrived, Vince (teacher) planned to use Linux as it is well known as a more stable server environment than Windows. However it is very complex and experience is required to set Linux up properly, this is one thing Vince lacked. After a while of down-time, the server was changed to a Windows system and worked fine – work commenced. This was a major blow to the time available, but I managed to work around it and complete my project on the date I specified.

After I had finished the teacher section of the page, it was time to add questions. This is where copyright issues come into play – I was given a year 11 technology textbook by Vince. I called up the 0800 number on the inside cover and asked if I could use their information for my project. The person I spoke to, Mark, told me I could only use a few pages and that I would have to e-mail him with a list of what I wanted to use – with references to page numbers etc. He would then consider it and reply to me. This was also a major blow to time, as I barely had enough time to add enough questions (with them given to me), let alone search through an unfamiliar textbook for things I can ask about - then asking for permission to write questions about them (the initial brief specified 200 questions). I spoke to the client about this, he agreed to meet with the technology faculty and speak to them about this issue. He responded telling me that they will put the questions in, as they already have the knowledge base for what should be asked.

Aesthetics were also an important key factor. The client specified a simple design with bright colours. This led me to conduct a heuristic evaluation with the help of technology students.

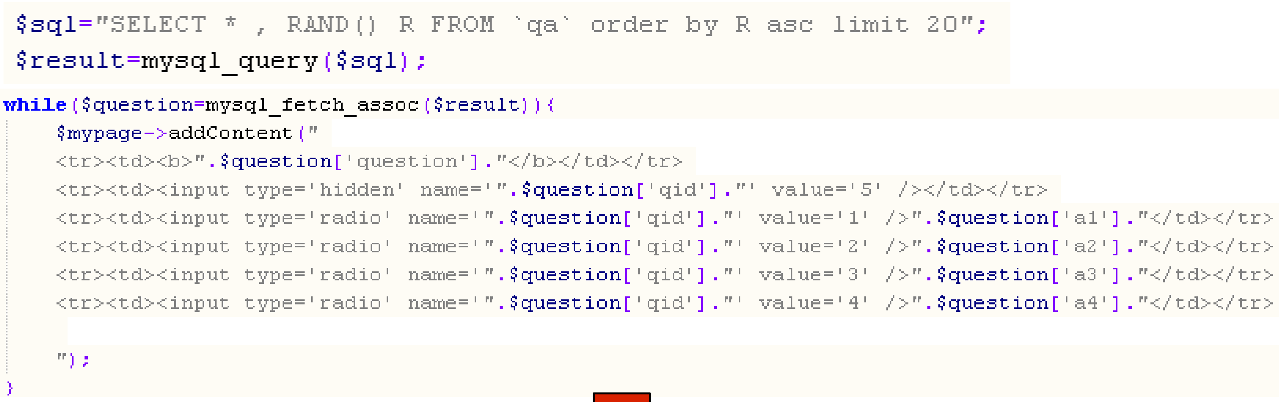
The specifications I made for my solution are largely based on my prioritised key factors. Without taking this step I would have been unable to ensure that the end product met all expectations.

Implementation – Prototyping

My first stage of implementation was my prototype. Based on the specifications of the client, I sketched what I wanted the page to look like on scrap paper. Then I wrote HTML and CSS to model it on a web page (included with submission). I showed it to the client to get feedback and he liked it. My prototype then became the basis for the rest of my project. My final solution still has the original HTML and CSS mixed throughout the class files and page files.

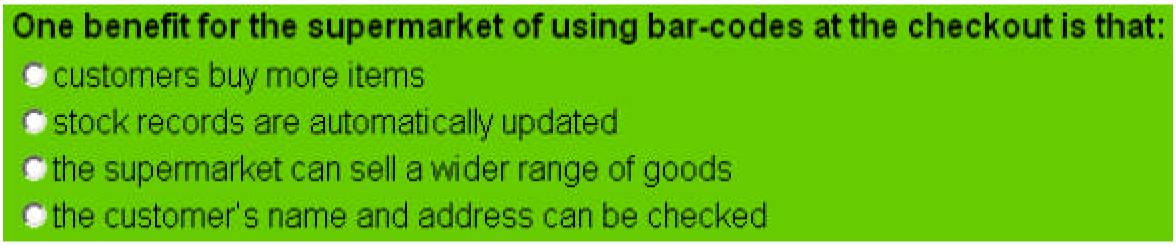
Everything after this stage was full of problem-solving as I was in uncharted territory. I will try not to repeat what I have already stated in this report, but I will go into more detail about my techniques in my code.

Implementation – Question Loop



```
$sql="SELECT * , RAND() R FROM `qa` order by R asc limit 20";
$result=mysql_query($sql);
while($question=mysql_fetch_assoc($result)){
    $mypage->addContent("
    <tr><td><b>".$question['question']."</b></td></tr>
    <tr><td><input type='hidden' name='".$question['qid']."' value='5' /></td></tr>
    <tr><td><input type='radio' name='".$question['qid']."' value='1' />".$question['a1']."</td></tr>
    <tr><td><input type='radio' name='".$question['qid']."' value='2' />".$question['a2']."</td></tr>
    <tr><td><input type='radio' name='".$question['qid']."' value='3' />".$question['a3']."</td></tr>
    <tr><td><input type='radio' name='".$question['qid']."' value='4' />".$question['a4']."</td></tr>

    ");
}
```



One benefit for the supermarket of using bar-codes at the checkout is that:

○ customers buy more items

○ stock records are automatically updated

○ the supermarket can sell a wider range of goods

○ the customer's name and address can be checked

This is the function used to populate the questions on the test page. '$sql' is the name of the query used to get the questions from the database. Originally I thought about using a formula or algorithm in my PHP code to randomly order the questions. After hunting around the internet for useful information to no avail, my student mentor advised me that I can do it all in a MySQL command if I read through the reference manual carefully enough.

The SQL function selects everything from the table, creates a column called R, which is all random numbers from zero to one (makes a random number correspond to each questionID). It then orders the table by the random numbers and defines a limit of how many questions will be selected. Each time the request is executed; the R column is re-calculated and gives a different order of questions.

While there is a question in the SQL result, the question is added. The corresponding qid and answers are added to the HTML. For every question that comes out of the database, the function is repeated.

The 'hidden' input with value '5' is not seen by the student (but can be seen in the source code) and is selected by default. Therefore if a student does not answer a question, it is wrong because there is no answer corresponding to value '5' in the database. This solved an ongoing problem that I was having trouble with; once I had figured out how to post to the database, I noticed that some numbers didn't add up – it didn't take me long to figure out why. I tried setting the default value in the database to wrong (0), which didn't really achieve anything. It took take me a while to click on that I could use a hidden input in to make them wrong.

This allowed results to be displayed and calculated correctly, which satisfied one of the requirements of the brief.

Implementation – JavaScript Issues

I noticed in the Cisco Systems exams they snap the browser to take up the full screen, including the task bar. The window has no address or navigational toolbars, and is not resizable. This is a good deterrent to cheaters. I tried to implement this in my solution using pre-made scripts, which are free to distribute, from dynamic drive. I found out that it is called a 'modeless window'. I successfully implemented the script in my code; however I decided to remove it because the annoyances outweighed the benefits. The default security settings of the school computers caused the web browser to pop-up security alerts about running the scripts. This interrupted the look and feel of the page, which hindered my aesthetics key factor.

```
<script>
  function modelesswin(url,mwidth,mheight){
  if (document.all&&window.print) //if ie5

eval('window.showModelessDialog(url,"","help:0;resizable:1;dialogWidth:'+mwidth+'px;dialogHeight:'+mheigh
t+'px")')
 else
 eval('window.open(url,"","width='+mwidth+'px,height='+mheight+'px,resizable=1,scrollbars=1")')
 }
</script>
```

Implementation – Software Used In Site Design

1. Notepad++
2. Microsoft Internet Explorer 6
3. Mozilla Firefox 1.0
4. XAMPP -> Apache web server with PHP module, MySQL database server (has more features, but were not used).

Notepad++ was used for writing all my code. All code used in my final design is 100% original and written from scratch. I used Notepad++ because it is free under the GNU General Public Licence and it has tabs for going from one file to the next. This meant that I could edit my class files while editing my page files and just flick back and fourth between them quickly. Also, as seen in some of the diagrams, it colour-codes values based on what kind of value they are, this makes editing code quicker and easier.

Testing was implemented with IE6 and Firefox 1.0, to make sure any changes worked as expected. Often Firefox interpreted tables differently to IE6, either pushing a column out too far or causing them to overlap. I solved this problem by just using rows without columns. Firefox is more standards based, so in theory it is a benchmark for all other browsers. Whether that is a good thing, I am not sure – Firefox was always the browser to cause a problem that I had to work through.

To upload files to the FTP server, I just used IE6. Usually IE6 is harshly unreliable for FTP, but as this was over a LAN, rather than the Internet, it worked fine all of the time. I used IE6 because it was on any computer I would be working on at school and no FTP programs were available anyway.

Implementation – Testing

Testing was conducted throughout my entire project. Every time I made a major change to my code, I would upload it to the server and test it. In doing this I could remedy any problems before moving on to other feature of my project. The login script required weeks of testing to get right, I had to make many minor adjustments to get all the nitty-gritty bits working. Sessions were the most difficult initially, but after adding the first two session variables I had a better understanding of them and adding more only took minutes.

Heuristic testing was also conducted from beginning to end. I based my prototype around as many of Dr Nielsen's 'rules' as I found applicable to my project. Then my client toured the prototype, told me he liked my layout and gave the thumbs up to move on to the final response to the brief. Throughout my PHP coding I kept the same ideals in mind. This was all to appeal to my key factor – aesthetics, and to follow the advice of Scott Abernathy.

Once my project was finished, I got four year 12 information science students to do a heuristic evaluation of the student section of my site, using Dr Nielsen's criteria. These evaluations can be found with my visual diary notes. To summarise: all of the evaluations returned good results. Some common comments were: “straight forward”, “simple interface/design”, “no errors”.

I made use of the heuristic evaluation to prove that the project meets international standards for usability. This provides proof that I have met the brief in terms of usability.

Skill Development

I had no experience in any of the skills I displayed in this project before the start of this year. On the timeline in my visual diary one can see when I learned HTML & CSS, PHP, and MySQL. This project required extensive use of a database, while delivering dynamic pages to the client.

In class we covered HTML and CSS fairly quickly. We were given 40-odd pages of notes and some examples. From there I learned more from W3 Schools, after that point I started getting really interested in code. If I saw something that I didn't know how to do on a site, I would view the source code and retain the information. I practiced at home, creating a static intranet for my home network.

MySQL was quite a step up from mark-up languages, but I understood it quickly. We were given notes in class to read through, and once the server was finally brought up, I had already drafted my initial database design. I practiced the language by creating the tables for my database. I learned more SQL throughout the project, by using the reference manual whenever I got stuck on something.

I began learning PHP from simple “Hello World” examples in class. Vince introduced us to new ways of using PHP throughout the time specified in the timeline. My student mentor, Hayden, was very experienced with PHP and I could always bounce ideas off him for advice. I learned a lot from him about how to structure my code the most efficiently. I found that there were 1001 ways to do anything with PHP, but usually only a few ways to do it quickly and efficiently. Most of my later learning of PHP was from browsing online for what I needed, whether it was the reference manual, php.net, or PHP freaks. There is a plethora of knowledge to be found on PHP.

Apart from W3 Schools, all the resources I used were commonly used by developers, so I never ran out of new information. PHP.net is run by The PHP Group who are the creators of PHP, not only do they know the language back-to-front and inside-out, but they know the best practice way to do everything. It is very community based, so I found a lot of links to brother/sister sites which provided more information.

Future Implications

Personally

Before I undertook this project, I had no experience of the technological process. Upon completion of this project I realised the importance of planning. Before I began my project, milestones were a foreign concept to me; milestones provided me with a structured method of how to complete my project. It was necessary for me to do things in a sequential order. Without the database, the PHP code would not achieve the desired functions, so it made sense for me to create the database first. I now understand the importance of planning when undertaking a large project.

I also realised that when undertaking a technological solution, it is very useful to look at existing solutions. It is much easier to criticize something in front of you than in your head. Studying existing solutions allows project managers to select which aspects are important, and should be included in the design of the solution; and what could possibly hold it back from being a polished product. It also gives hints on what kind of interfaces work well.

The skills I learned while undertaking this project will provide me with a foundation for further study at University. The use of classes, loops, database design, etc, is all covered in first year computer science. This will give me a head-start to further study.

Locally

Vince Brannigan and Terry Hawkings both want to use this for testing their junior students, which was the original purpose of the brief. Using my database techniques, students should be getting better feedback from their tests. With the amount of available PCs in WHS, over 100 students could all sit a test at the same time. In theory, the project could be used to test anything, or even provide different tests at the same time. This would require an extra column in the 'qa' table such as 'exam name', so it could test for all kinds of subjects. Time constraints meant that I could not expand my project beyond the brief, but because of the flexible design of my code, changes would be rather simple.

Nationally

Schools are always looking at new ways to test students. Although the format of my test is not compliant with NCEA, it is a very useful tool for testing students' ability to retain information. The test would be especially useful for testing subject-specific vocabulary. The most effective use for this solution, in a national sense, would be a formative test. It could give an indication to the student of their level of understanding of a specific subject; before they sit the NCEA based summative testing.

Roger Joyce, web designer for techlink, informed me of a limitation in my usability design. My code does not provide a way for blind people to access the pages; therefore it does not meet Government 'e-learning standards' of usability. This is one thing I did not consider, but it does affect the level of exposure my project could receive on a national level.

Globally

Similar style multiple-choice questions have been used by very reputable firms for revision and testing purposes. In England, the BBC has provided resources for schools involving multiple choice questions, although these are provided in PDF format. In USA, Cisco Systems uses multiple choice questions for their academy programme. This shows that it is feasible to make this style of online test run in a world-wide environment in the future, of course the database structure would have to be much more complex to handle those kind of expectations.

Critical Reflection

I developed my technological process to a high standard, although there is still much more room for improvement. I produced an elegant product because it can be easily modified and upgraded. A project like this could be worked on for years and constantly expanded to provide more and more testing functions. However time was my most limited resource, and I had to use it very wisely. Therefore I set myself tasks that I expected to achieve and did them. I barely scraped by with enough time to write the report, but I managed to make sacrifices to ensure I completed everything to a standard I was happy with.

I was told by Vince that my methods in dealing with my client were quite unconventional in a school environment. He called it 'rapid prototyping'. Basically I asked the client what he wanted, drew up a prototype and brought it straight back to him for feedback. Then I fixed up anything he didn't like and took it straight back to him again until he was happy with it. I found that this was a very convenient way of speeding up the design process, as Terry was usually only two minutes away from my class I could just run over and show him what I had done and quickly get feedback.

I was very loose with security because I knew that everything was on an internal network which was protected. If it was going to be released for use over public network areas, such as the Internet, more precautionary measures would need to be taken. HTTPS would be used by making an absolute reference to the login page, and MD5 would be used for all passwords stored in the database. That way if a teacher login is compromised, the unauthorised user cannot see the students' passwords in the database. All they would see is a MD5 hash value. Also packet sniffers would not be able to view any data in transit, with SSL enabled.

Using class files allowed me to meet the requirements of the client. Throughout the course of my project, my client discovered new features he would like to be added to the site. By using class files, I could quickly add or remove relevant pieces of information where necessary.

One of my early ideas was to have preset code to create an exam which could be activated by a teacher at testing time. If there were multiple tests on the database (example: ICT, fabrics, etc), the teacher would choose a test, choose a class to sit the test (class lists would be required in student table), then choose to activate the test. If the student logged in when an exam was not activated, the PHP script would inform the student that there are no exams currently active. If there was an exam activated, it would have provided a description of the exam and a link. This was featured in my early prototypes. However I only just finished my working exam page and mark processor at the end of development phase, and did not have time to explore functionality to that extent as I had to move on to my write-up and final presentation, as planned.

With my design, a student can log on at any time (inside WHS) and sit the exam. For integrity of data, we would be counting on the fact that students don't know it exists before they sit the exam. It is a very inconspicuous URL though (http://10.1.0.100/INS/Sam/dynamic/index.php), the odds of a year 10 technology student guessing that address without being told that an online exam exists on the network are slim to none. However I must admit that this is a vulnerability.

Although all my code was tested as working for Firefox and IE, realistically I could have validated it all as I went. I didn't bother stripping my PHP down to its HTML components for validation. This could cause a bug in later releases of browsers. With IE7 and Firefox2.0 just around the corner, I can not be certain that they will react as well as their predecessors to my HTML code.

This experience has taught me many valuable lessons as well as several new languages. I now understand that time is a very precious resource, and if it is used efficiently… great things can happen.