

Continuous Assessment: ECM1417

blue20.ex.ac.uk/as1200/Coursework/index.php

March 27, 2019

0.1 Database

Schema: The database has three tables, a users table, a stock information table and a stock entries table. These tables allow for logging of stock entries and ease of user management. The schema is as follows:

Users:	StockInfo:	StockEntries:
<ul style="list-style-type: none">• UserID INT Primary key	<ul style="list-style-type: none">• StockID INT Primary key	<ul style="list-style-type: none">• EntryID INT Primary key
<ul style="list-style-type: none">• username VARCHAR(20) UNIQUE	<ul style="list-style-type: none">• StockName VARCHAR(255)	<ul style="list-style-type: none">• StockID INT
<ul style="list-style-type: none">• Name VARCHAR(30)	<ul style="list-style-type: none">• StockPrice INT	<ul style="list-style-type: none">• ChangeType INT
<ul style="list-style-type: none">• HashedPW BINARY(60)	<ul style="list-style-type: none">• CurrentQuantity INT	<ul style="list-style-type: none">• Amount INT
<ul style="list-style-type: none">• Permissions INT	<ul style="list-style-type: none">• LastModified TIMESTAMP	<ul style="list-style-type: none">• TimeCreated TIMESTAMP
	<ul style="list-style-type: none">• LastModifiedBy INT	<ul style="list-style-type: none">• CreatedBy INT
	<ul style="list-style-type: none">• UserID INT	
	<ul style="list-style-type: none">• TimeCreated TIMESTAMP	

There are four relationships between the tables:

- Users.UserID \leftarrow one to many \rightarrow StockInfo.UserID.
- Users.UserID \leftarrow one to many \rightarrow StockInfo.LastModifiedBy.
- Users.UserID \leftarrow one to many \rightarrow StockEntries.CreatedBy.
- StockInfo.StockID \leftarrow one to many \rightarrow StockEntries.StockID.

Reasons This method of storing the stock data is useful as it allows an admin to get a record of what has changed and manage users. A trigger on the StockInfo table updates the LastModified and LastModifiedBy columns

with the values of the latest entry in StockEntries with the same StockID. The permissions column allows a user to be restricted in what they are allowed to do changing the pages they visit and blocking them from submitting some forms. There are two ways of changing the Stock, "change" and "set". They are intended for different purposes; a "change" stock is when a new shipment arrives or some stock is sold, the stock changes by some amount. A "set" stock on the other hand is intended for when something like a stock count is done, if there is some discrepancy between the current stock amount and the counted amount it can be resolved by simply setting the current stock count to one that was just found.

0.2 UI

0.2.1 Index

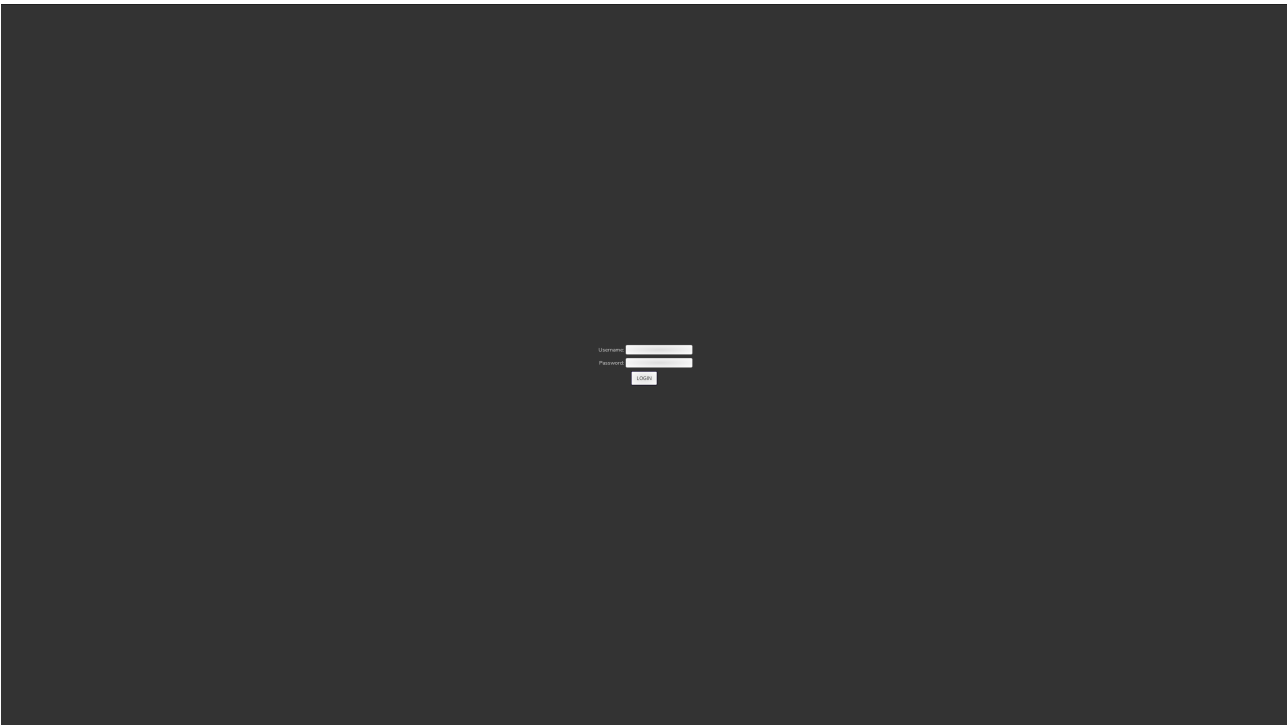


Figure 1: index.php

The index page has a simple layout, it's just a small form for the username and password and a login button. The form always appears in the centre of

the screen horizontally and vertically. If the user incorrectly logs in an error message appears in red as can be seen in Figure 2.

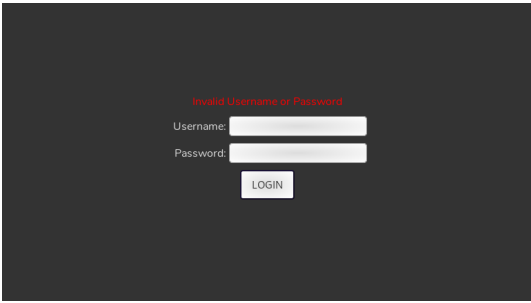


Figure 2: index.php error message

The dark background with the lighter boxes brings the users attention to the form making it clear and easy to see what to do.

0.2.2 Stock

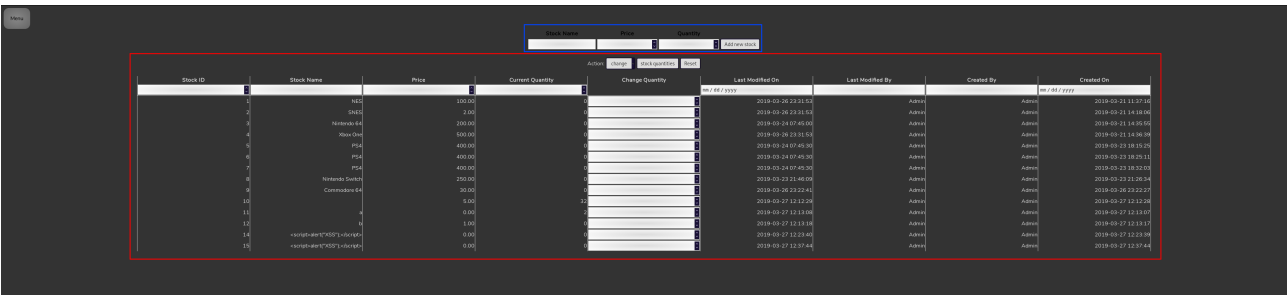
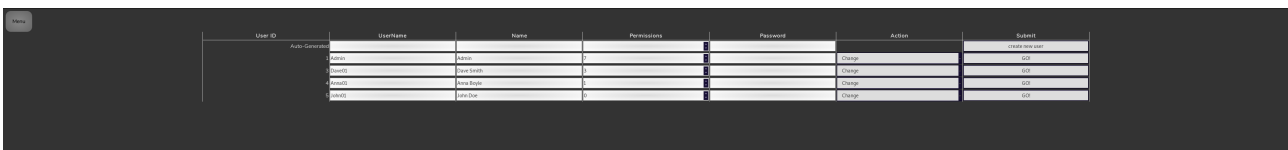


Figure 3: stock.php

The stock page as seen in Figure 3 has three main components, the add stock form (shown in red), the change stock form (shown in blue), and the sidebar (not shown). The add stock form and the change stock form both have permission requirements to see or send. This means that if a logged in user does not have the adequate privileges the forms will not be produced by the php. There is also a check on each of the post request handlers to see if the user have permission to make the request. A session token is also generated for these forms to stop Cross Site Request Forgery. Care is also taken to stop Cross Site Scripting, you can see in the Figure that there are some attempts at XSS in the name column but these have been correctly rendered and the XSS didn't work. Like the index page the darker

background highlights the forms allowing a user to see where they should put in any input. The lone button in the corner is for opening sidebar. The dark minimalist layout makes the website match the trend in design leading to modern feel. The table has three components, in descending order: the submit options, the search terms, and the quantity form. The submit options allow a user to submit their changes as "change" or "set" requests, it also allows the user to reset all the fields in the table. The search terms allow a user to filter the table to only include the results they want, this makes a request to getStock.php which in turn returns a table. The quantity form allows the user to set the quantities they want to change. Only the fields with values will be sent so the post request is smaller and takes less time to process. The sidebar on this page does not have a link to the admin page if the user does not have the correct privileges for it.

0.2.3 Admin



User ID	Username	Name	Permissions	Password	Action	Submit
Admin	Admin	Admin	1		Change	100
Admin	Admin	Admin	1		Change	100
Admin	Admin	Admin	1		Change	100
Admin	Admin	Admin	1		Change	100
Admin	Admin	Admin	1		Change	100

Figure 4: admin.php

The admin page is only accessible to users who have admin privileges, if they do not they are redirected to the stock page. The page has a table which allows an admin to manage user accounts. The admin can add an account with with the first row and change any of the user's name, username, permissions and password. The password is hashed using bcrypt allowing it to be stored in one column of data in the database. The admin can also remove any account by changing the action field. This page allows for quick creation of new accounts for new employees or removal of accounts if they become compromised. Again this page fits with the theme with a dark background and light form inputs to bring attention to where the user can make changes. This look keeps within the modern and simple feel of the site. The sidebar, like on the stock page, is hidden until the menu button is clicked at which it appears on the right hand side.

0.3 Security

Cross site request forgery Cross site request forgery (CSRF) is a form of attack that allows an attacker to perform action for the victim without their knowledge. A common mitigation for this is to generate a token when the logs in. This token is then inserted into all requests that the user makes. The token is checked against the one stored on the server and the request is only allowed if the tokens match. I have implemented this on all my forms that change data on the server so CSRF is not possible. Unfortunately, all CSRF protections are pointless if a the site is vulnerable to Cross site scripting.

Cross site scripting Cross site scripting (XSS) can be thought of as a more powerful version of CSRF, where CSRF can only make a user unknowingly perform a predefined action, XSS allows an attacker to execute arbitrary commands on the victim giving them no limit on what they can do. To protect against XSS, any place where users can input data must be sanitized and escaped properly. In PHP this can be done with the *htmlspecialchars* function.

Password storage One of the most popular words of wisdom when it comes to anything security related is: “don’t attempt to make your own encryption, other people have already done it better”. This lead me to using bcrypt, which is easy to use in PHP. The *password_hash* function has an option for bcrypt making it as simple as changing a variable. the hashes (which also contain a salt) can be stored in a 60 byte string and verified with *password_verify*. My application is written in php7 so I used these functions as they are recommended. It wasn’t until I went to put the source code on the University’s computers that I realised that they use php5.4. The functions I’ve used were introduced in php5.5 so I can’t use them. However Anthony Ferrara on Github has made a library that forward ports these functions to older functions of php so I have used it in this project because I refuse to compromise the user’s security.