

ACCA5026 Assignment JMilligan

by Julian Milligan

FILE	16671_JULIAN_MILLIGAN_ACCA5026_ASSIGNMENT_JMILLIGAN_317166_1883337763.DOCX (1.01M)		
TIME SUBMITTED	09-MAY-2021 01:32PM (UTC+0100)	WORD COUNT	2229
SUBMISSION ID	152328586	CHARACTER COUNT	10831

Name – Julian Milligan

Student number – 1904804

Course – Bachelor of Science (Part-time): Computing (Software Engineering)

Module – L5 - Databases and Application Development

Module code - ACCA5026

Date – 07/05/2021

Contents

Introduction	3
Customer Login	3
Register an account.....	4
Shop	4
Staff Login.....	4
Database	5
Customer.....	5
Orders.....	5
Products	5
Staff	5
Staff_login	5
Site Security.....	6
Improvements.....	9
Appendix	9
Appendix 1	9
Bibliography	10
Further reading	10

Introduction

The website designed for this assignment an online store for merchandise based around the popular video game Watchdogs. The site uses the bootstrap framework for the layout aspects navigation, content layout and carousel. On the home page the page shows the top 10 products this is based on a query that gets the top 10 items with the lowest stock item at the top. From the home page there 4 further options

Customer Login

This allows current customers to login to their profile. From there they can update any details held by the company and view their purchase history. When the customer tries to login using the form it asks for their username and password. As a security measure the TYPE for the password field is set to "password" which changes the typed text to *, however this is not secure. On a person device because browsers like Chrome and Edge cache passwords it may show if another user opens the page. Even though at first glance they may not be able to see to that password a simple hack will make display in plain text. If someone right clicked on the webpage and then inspect then they can change some of the code on the site. Changing the TYPE to "text" will display the raw password.

```
<div class="row"> <!--
  <div class="col-lg-8">
    <h2>Please login to update your details and see your order history</h2>
    <h3>Please enter your username and password to login</h3>
    <form action="cust_login_check.php" method="post" pb-autologin="true" as:
      <table style="width:100%">
        <tbody>
          <tr>
            <td>
              <input type="text" id="username" name="username" pb-role="pass
              <input type="password" id="password" name="password" pb-role="pass
            </td>
          </tr>
        </tbody>
      </table>
    </form>
  </div>
```

Please login to update your details and see your history

Please enter your username and password to log

Enter your username

Enter your password

One way to counter act this is stop browser auto filling usernames and passwords in the code of the form.

In order to login both the username and password must match a record in the table customer. Once they have logged in a session cookie is created with their username and a dynamic URL is created to allow them to access their profile. When they click on the link, they are presented with a form that is populated with their held in the database. Below that is a history of any purchases which uses a join to combine three tables' ORDERS, CUSTOMER and PRODUCTS.

Register an account

This allows a new customer to create an account. When an account is created the form and the PHP code to insert the new account are ran on the same page using `<?php print $_SERVER['PHP_SELF'] ?>` in the action on the form tag. In the code there is an IF statement that checks `$_POST` is not empty and then will run the code to insert the user. Each variable that's is passed is cleaned using `mysqli_real_escape_string` to help prevent SQL Injection. This is discussed further on in the report under Security. Then the query is prepared again to prevent SQL Injection [1]. Once the security measures are complete a search is run to see if the username has already been used. There are 2 customers in the database that do have the same username this was to test that both the username and password must match in order to login.

If the username exists a message is passed to the customer and the code stops. If the username is safe to use then the record is inserted and a confirmation message is displayed asking them to login.

Shop

The shop page allows a visitor or guest to browse all the products currently on sale. These are displayed in ascending stock level. It also uses paging which is set to display 10 items on each page. This code is from a previous lecture. To the right of the products is a search facility. The search is based on a wildcard (`%searchTerm%`) against the product name. Again, the form and PHP code run on the same page. Once the search has run the results are shown with the number of results. The search is also in an IF statement that checks if `$_POST` is not empty, this will prevent the search running on page load and with blank data. In the ELSE of the IF statement there is a function that will shown a random quote from the game. [2]

Staff Login

This allows staff to login, depending on if they are an admin or not, they have access to the following features.

1. View current non dispatched orders. (All Staff)
2. View staff login logs (Admin only)
3. Admin Page to add new staff members and view current staff (Admin only)

The staff login works the same as the customer one, the member of staff must match their staff number and password. Once they have a session cookie is created and they are given a list of links depending on if they are an admin. This is checked by an IF statement.

Each time a member of staff logs in to the system an Insert is made in to `staff_logins` table. The only variable that is passed is the staff member's name. The ID is set to auto increment and set as the primary key a time stamp is also added.

If they an admin and view the current list of staff, they have the option to delete a member of staff. They are also able to create a new member of staff. Some of the details are prefilled with defaults and the password is auto generated using javascript the code is a variation of the following code [3]. and is readonly. This could be hidden in future and the staff members email

captured this would then allow for the password and login info to be emailed to the staff member.

Please fill in the details below to register an account.

Staff Number	S100
Name	
Is Admin (1 for yes 0 for no)	0 
Login Password	Sxo6LBT9PP 
Add Staff Member	

© 2017  2021 Julian Milligan, Inc. ·

Please see the readme file for a list of staff logins.

Database

The database consists of 5 tables please see [Appendix 1](#) for the ERD that the design was based on. Some aspects changed when the database were created.

Customer

This table holds all the customer details with the customer ID as the primary key and various other fields to hold Name, Address, Postcode, Telephone, Email, username and password. This is inline 3NF so only one record is needed for each customer, but the ID can be stored in the Orders table against multiple orders.

Orders

This table holds all the orders currently in the system. order_ID is the primary key with foreign keys for customer ID and product ID. The date of the order is also captured and if the order has been processed.

Products

Currently as the company are only starting out their product list is limited. The table holds a product name, description, image name, price, stock level and category.

Staff

This holds the staff data which includes Staff ID, Name, are they an admin and their password.

Staff_login

This table holds the entries for each login for a member of staff. The log_ID is the primary key and is set to auto increment. The login_date is a time stamp so on the insert the only value passed is the staff members name.

Site Security

No matter how well designed a website is if it contains any data that could be personal or business related then steps must be taken to protect it. The method that is widely used is called CIA Triad [2]. As quoted by Josh Fruhlinger

- **“Confidentiality:** Only authorized users and processes should be able to access or modify data
- **Integrity:** Data should be maintained in a correct state and nobody should be able to improperly modify it, either accidentally or maliciously
- **Availability:** Authorized users should be able to access data whenever they need to do so”

The steps taken in the design of the for each of the layers is as follows.

Confidentiality

Because a customer must login to see or update their data this means that another user may not access their account. One way to increase the security as it is a signal factor authentication would be to add multi factor authentication. This would add an extra level of security to the login. There are a number of ways this could be done from entering a simple code that's texted to a mobile number or integrating Microsoft's Office 365 login [3] and using their Authenticator App to generate a code. There are other options such as Google's Authenticator App.

The same would need to be enabled for the staff login.

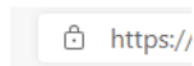
Integrity

Only a customer can update their record currently so only they could change their delivery address, email, and telephone. Allowing staff to alter these details could open a security issues with the possibility of staff altering the address to obtain goods. With in the site only 2 members of staff are allowed to add new staff as they have been set as admins of the system. This again makes sure only a limited number of people can access these pages.

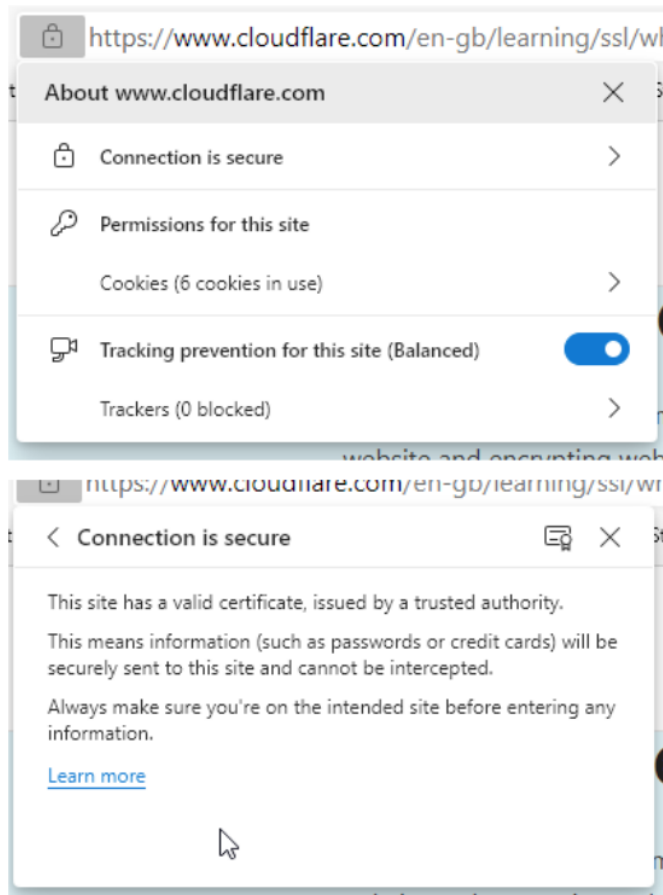
Availability

Anyone who logins to the system can only see the data that they need to see. A customer does not see any data about staff members and the staff can not see any of the data about customers other than the customer ID on the orders page.

To increase the security of the site when it goes live a HTTPS certificate will be needed. This will give a number of benefits. One it will allow the customer to have the confidence that the site is secure because in most browsers a padlock is shown next to the URL.



The main benefit will be that the site will be the site that the customer goes to. The issue with HTTP sites is they can be hijacked. So a customer goes to <http://watchdogsMerch.co.uk> and the index page may look as expected but the rest of the site has been cloned and the details are being captured into another database and payments sent to another person or company. HTTPS will prevent this because the SSL certificate will guarantee the URL of the site is the same as the address on the server. The SSL or Secure Socket Layer holds information on the owner of the site, the date it was issued, the public key and the domain name. They are also encrypted so they can not be tampered or cloned [4]. Clicking on the padlock allows you to view certain information about the certificate.



Another issue which HTTPS may not help with is SQL Injection. This involves a hacker adding an extra SQL query to the query string passed via the POST method when a form is submitted. Once place this could be done on the site is when the customer logs in. The URL to display their order history and profile information is http://localhost/UniYear2/ACCA5022_Assignment/cust_profile.php?id=10001 extra code could be added on. However to get to this page they must first login in so the hacker would need their details first also when the user logs in a Session cookie is created and its existence is checked when this page loads. Another preventive measure that is taken is that the ID is cleaned by using the function 'mysqli_real_escape_string'. This function escapes any special characters from the string. [5]

As part of the design of the site is was run through SQL MAP [6] Below is the results when ran on a Kali Linux virtual machine.

```
root@kali: ~#
root@kali: ~#
[07:12:10] [WARNING] set a git repository. It is recommended to clone the 'sqlmapproject/sqlmap' repository from GitHub (e.g. 'git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap')
do you want to try to fetch the latest 'sqlmap' from repository and extract it (experimental) ? [y/N] y
[07:12:10] [INFO] updated to the latest version '2.5.5.180a'
[*] ending @ 07:12:44 /2021-05-09/

root@kali: ~# sqlmap -u http://192.168.1.193/UniYear2/ACCA5022_Assignment/cust_profile.php?id=10001 --batch

(1.5.3.180a)
http://sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.
[*] starting @ 07:12:47 /2021-05-09/

[07:12:47] [INFO] testing connection to the target URL.
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=7asqkqpydg...8uomamjd'). do you want to see those [Y/n] Y
[07:12:47] [INFO] testing if the target URL content is stable
[07:12:48] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user'
a manual paragraph 'Page comparison'
how do you want to proceed? [[Continue]/[Using]/[Ignore]/[Quit]] c
[07:12:48] [INFO] testing if GET parameter 'id' is dynamic
[07:12:48] [WARNING] GET parameter 'id' does not seem to be dynamic
[07:12:48] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[07:12:48] [INFO] testing for SQL injection on GET parameter 'id'
[07:12:48] [INFO] testing 'AND boolean-based blind - WHEDE or HAVING clause'
[07:12:48] [INFO] testing 'MySQL > 5.1 AND error-based - WHEDE, HAVING, GROUP BY or GROUP BY clause (EXTRACTVALUE)'
[07:12:48] [INFO] testing 'PostgreSQL AND error-based - WHEDE or HAVING clause'
[07:12:48] [INFO] testing 'Microsoft SQL Server/Oracle AND error-based - WHEDE or HAVING clause (MS)'
[07:12:48] [INFO] testing 'Generic inline queries'
[07:12:48] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[07:12:48] [INFO] testing 'Microsoft SQL Server/Oracle stacked queries (comment)'
[07:12:48] [INFO] testing 'Oracle stacked queries (DBMS_PIPE, ADVISE, MAKEVAL, comment)'
[07:12:48] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP) - comment'
[07:12:48] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[07:12:48] [INFO] testing 'MySQL > 5.0.12 AND time-based blind'
[07:12:48] [INFO] testing 'Microsoft SQL Server/Oracle time-based blind (IF)'
[07:12:48] [INFO] testing 'Oracle AND time-based blind'
as it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. do you want to reduce the number of requests? [Y/n] Y
[07:12:48] [INFO] testing 'Generic inline queries (SQL)'
[07:12:48] [INFO] GET parameter 'id' does not seem to be injectable
[07:12:48] [WARNING] all tested parameters do not appear to be injectable. Try to increase values for '--level/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involve
ed (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent'
[*] ending @ 07:13:52 /2021-05-09/

root@kali: ~#
```

It tests form a number of SQL injections vulnerabilities. The results were positive as it was not able to view any data and declared that ID is dynamic. This could be an issue though as the shop part was developed more. As no session cookie would be created and the URL may pass a static value.

Another test was run to see if the database name could be captured by the command 'sqlmap -u http://192.168.1.193/UniYear2/ACCA5022_Assignment/cust_profile.php?id=10001 -dbs'

```
root@kali: ~# sqlmap -u http://192.168.1.193/UniYear2/ACCA5022_Assignment/cust_profile.php?id=10001 -dbs

(1.5.3.180a)
http://sqlmap.org

[!] legal disclaimer: usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.
[*] starting @ 07:17:49 /2021-05-09/

[07:17:49] [INFO] testing connection to the target URL.
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=delrnfjbc...vslapheab'). do you want to see those [Y/n] y
[07:17:50] [INFO] testing if the target URL content is stable
[07:17:50] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user'
a manual paragraph 'Page comparison'
how do you want to proceed? [[Continue]/[Using]/[Ignore]/[Quit]] c
[07:17:50] [INFO] testing if GET parameter 'id' is dynamic
[07:17:50] [WARNING] GET parameter 'id' does not seem to be dynamic
[07:17:50] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[07:17:50] [INFO] testing for SQL injection on GET parameter 'id'
[07:17:50] [INFO] testing 'AND boolean-based blind - WHEDE or HAVING clause'
[07:17:50] [INFO] testing 'MySQL > 5.1 AND error-based - WHEDE, HAVING, GROUP BY or GROUP BY clause (EXTRACTVALUE)'
[07:17:50] [INFO] testing 'PostgreSQL AND error-based - WHEDE or HAVING clause'
[07:17:50] [INFO] testing 'Microsoft SQL Server/Oracle AND error-based - WHEDE or HAVING clause (MS)'
[07:17:50] [INFO] testing 'Generic inline queries'
[07:17:50] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[07:17:50] [INFO] testing 'Microsoft SQL Server/Oracle stacked queries (comment)'
[07:17:50] [INFO] testing 'Oracle stacked queries (DBMS_PIPE, ADVISE, MAKEVAL, comment)'
[07:17:50] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP) - comment'
[07:17:50] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[07:17:50] [INFO] testing 'MySQL > 5.0.12 AND time-based blind'
[07:17:50] [INFO] testing 'Microsoft SQL Server/Oracle time-based blind (IF)'
[07:17:50] [INFO] testing 'Oracle AND time-based blind'
as it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. do you want to reduce the number of requests? [Y/n] y
[07:17:50] [INFO] testing 'Generic inline queries (SQL)'
[07:17:50] [INFO] GET parameter 'id' does not seem to be injectable
[07:17:50] [WARNING] all tested parameters do not appear to be injectable. Try to increase values for '--level/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involve
ed (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=spacecomment') and/or switch '--random-agent'
[*] ending @ 07:18:52 /2021-05-09/

root@kali: ~#
```

Again this information was not captured.

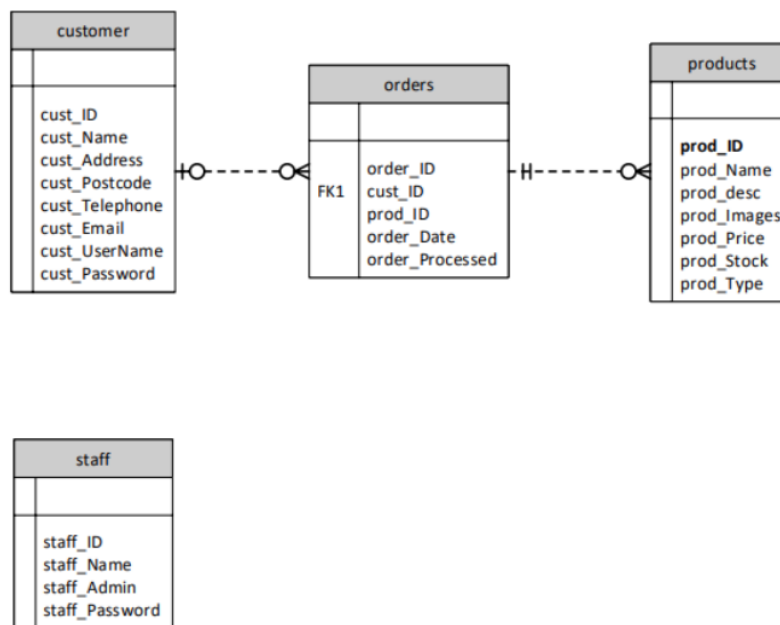
Improvements

These are a list of improvements that could be made to the site in order to make it a fully functioning shop.

- Add a basket so orders could be made.
- On the product search in the shop add a category search.
- On the admin section allow for staff to add new products.
- On the admin section allow for staff to alter products.
- On the customer login add a forgot password option.
- An option to set the order as processed which sends an email to the customer.
- Email alert on products if the stock falls below a set level.

Appendix

Appendix 1



Bibliography

- [1] "The mysqli_stmt class," PHP, [Online]. Available: <https://www.php.net/manual/en/class.mysqli-stmt.php>. [Accessed 05 2021].
- [2] J. Frunhlinger, "The CIA triad: Definition, components and examples," CSO United Kingdom, 10 02 2020. [Online]. Available: <https://www.csoonline.com/article/3519908/the-cia-triad-definition-components-and-examples.html>. [Accessed 05 2021].
- [3] "Secure access to resources with multifactor authentication," Microsoft, [Online]. Available: <https://www.microsoft.com/en-gb/security/business/identity-access-management/mfa-multi-factor-authentication>.
- [4] "What is an SSL certificate?," Cloudflare, [Online]. Available: <https://www.cloudflare.com/en-gb/learning/ssl/what-is-an-ssl-certificate/>. [Accessed 05 2021].
- [5] "mysqli::real_escape_string," PHP, [Online]. Available: <https://www.php.net/manual/en/mysqli.real-escape-string.php>. [Accessed 05 2021].
- [6] B. F. RAMADHAN, "SQL Injection with Kali Linux," 2019. [Online]. Available: <https://linuxhint.com/sql-injection-kali-linux/>. [Accessed 05 2021].

Further reading

Websites

[PHP: How to auto redirect page after form submission to database - WP3 Layouts](#)
[Combining HTML and PHP Code on a Single Page | Working with Forms in PHP | InformIT](#)
[HTML input required Attribute \(w3schools.com\)](#)
[PHP MySQL Insert Data \(w3schools.com\)](#)

Books

Beginning PHP and MySQL Third Edition ISBN – 978-1-59059-862-7
PHP Object-Oriented Solutions ISBN – 978-1-4302-1011-5

ACCA5026 Assignment JMilligan

GRADEMARK REPORT

FINAL GRADE

72/100

GENERAL COMMENTS

Instructor

WEBSITE

Database structure and design 13/15 - Product type would be a good candidate for normalisation as there is a lot of repeated data. Otherwise OK

Range of PHP code demonstrated

29/40 - Search is good and incorporates a wildcard which makes it very usable. It would be nice if more fields were incorporated in the search.

Login is OK, but you need to manage the 2 states on the menu bar, it invites me to log in even when I already am. Also, I cannot find a way to log out?

Register for account works well, but make the feedback to the user a bit bigger. Also, it's a bit confusing that the register page comes up again after the account is successfully registered. Duplicate usernames have been prevented which is good.

Good use of paged results, although there's a slight bug in there on the Top 10 products - sometimes it shows 20 records and sometimes 10.

I'm a bit confused at the orders table as there doesn't seem any way to place an order on the website?

Edit/Delete also used.

Overall integration of website

11/15 - Looks good overall with good use of responsive techniques. It looks best in mobile mode, in the desktop mode there is a lot of waste space. Centering the text area could mitigate this. I never grey text on black is very readable - you might want to consider a more contrasting colour. Good use of menu bar. Login management is a bit confusing as commented on in the above section.

REPORT

Security issues identified I think you've misread the brief somewhat here as the first part of the report was not needed, just the security issues. You have identified some issues well such as SQL injection and HTTPS, but I'm not sure MFA is practicable for a website such as this. There are other issues you could have talked about such as cross-site scripting, preventing spambots etc. 8.5/15 -

Consideration of security issues for website Generally a good appraisal 10/ 15 -

TOTAL: 71.5%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10
