My name is Kemal Osmanovic, and

my student number is c0013667. This assignment is for the undergraduate course for Bsc (Honours) Computing at the Sheffield Hallam University. This individual assignment is for my PHP integrated computing module.

Contents

[Introduction 2](#_Toc67577287)

[Create statements 3](#_Toc67577288)

[Login page 5](#_Toc67577289)

[Home page 6](#_Toc67577290)

[Products page 7](#_Toc67577291)

[Contact us page 8](#_Toc67577292)

[Basket page 9](#_Toc67577293)

[About us page 11](#_Toc67577294)

[Conclusion 11](#_Toc67577295)

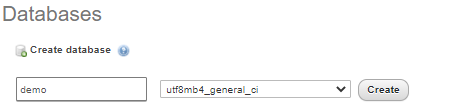
[References 11](#_Toc67577296)

# Introduction

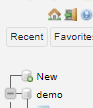
This assignment was to develop an e-commerce prototype website which links to a MySQL database where data is stored and can be written to and fetched from by my website. I didn’t have a theme for my website, but I just chose a different range of products which had different prices. My website only promotes three products which are; an apple, a drone, and a toothbrush. This can be seen by the screenshots taken from my website which are below. The screenshots will also show the prices for each of these products.

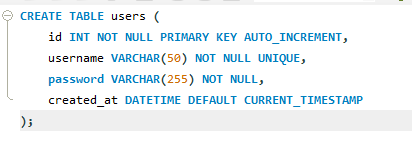
In this report I will include all the php statements with concise descriptions of what each do as well as all sql statements.

# Create statements



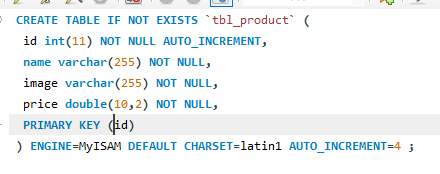
The screenshot above shows me creating a database called demo which was the name I went for. This was a first try at creating a database hence the shortened name for ‘demonstration.’ However, I didn’t need to retry making a new database as it worked first time. Once creating it the result from this was that a new database was displayed on the left-hand side of PhpMyAdmin called ‘demo.’ This can be seen by the screenshot below.



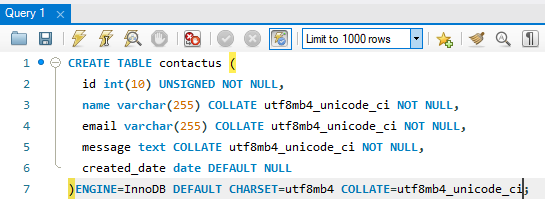


This creates statement creates a table called users with 4 different attributes. The first attribute is ‘id’ which is the primary key integer that cannot be null and auto increments which mean a value is automatically assignment to the first row and each following row has a value incremented by 1 based on the value of the id of the previous record. The second attribute is the username which can’t be null and must be unique. The datatype assigned to this attribute is a varchar which means the value can be a variable length string which can be a combination of letters, numbers, and special characters. The parameter of this attribute is 50 which means a maximum of 50 characters can be entered. (W3.css, 2021) The third attribute is the password which doesn’t have to be unique but must not be null and is a varchar with the parameter of 255 characters. The final attribute created for this table is called created\_at which has a datatype ‘DATETIME’ which assigns the exact date and time that the user was created. This table will be used for the login page and the result of running this create statement can be seen below.





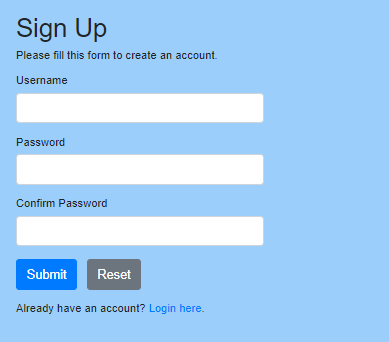
The above screenshot shows the create statemet I ran to create the table for products called “tbl\_product.” This creates a table with 4 attributes. The first attribute is called id which is an integer primary key which cant be null and autoincrements. The second attribute is called name which is a varchar with a paremeter of 255 characters and is not null. The third attribite is called image which is a varchar with a parameter of 255 characters and is not null. The final attribute is called price which is not null and has a datatype called double which is a float that can have parameters to handle decimals. In this case, the price can have 10 values before the decimal place and 2 after the decimal place. Below is a screenshow of the result of running this create statement. 

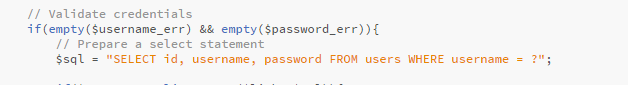


The above screenshot shows the create statement that will create the table “tblcontact” that will be used for the contact form that is used on the website. This will create the table with 5 attributes. The first attribute is the id which is the primary key which is an integer and not null with a maximum of 10 characters. The second attribute is the name which is a varchar with a maximum of 255 characters and is not null. The third attribute is the email which is a varchar with a maximum of 255 characters and is not null. The fourth attribute is the message which has a datatype called text and is not null. The final attribute is the created\_date which has a date datatype and is not null. Below is a screenshot of the result of running this create statement.

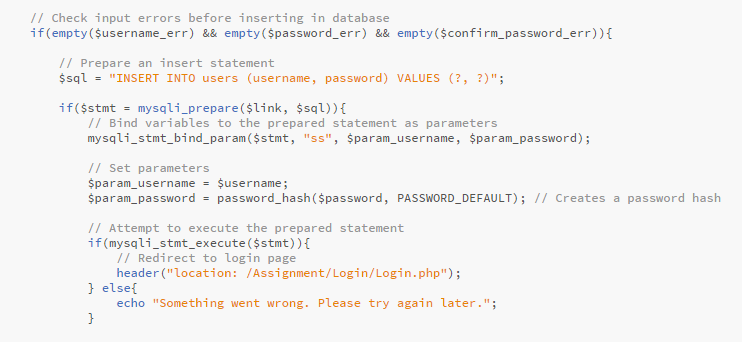


# Login page

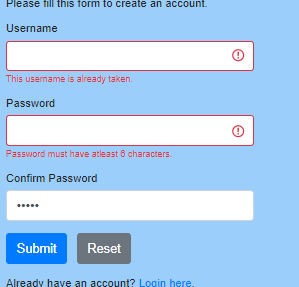
This is what the login page looks like. The user can type in their username and password to login by entering values in the corresponding fields then clicking the button ‘login.’ However, if they don’t have an account, they can click sign up now which redirects them to a page which they can sign up. The sign-up page can be seen in the screenshot below.



The above screenshot shows how I validate the credentials. I use a sql query in php which selects the credentials which correspond to the username which is entered. If the password and username both match with what is in the database, then the user can login.



The above statement is linked to the sign-up page. This php and sql query code in summary, checks if the username is taken as it must be unique. If it’s not taken, then it can be stored in the database. However, if it’s already taken then an output message is outputted for the user so they can try again. If the query to store the information is executed, then the user is redirected to the login page where they can use them credentials to sign in and gain access to the main website.

The screenshot to the left shows the result of entering a username which has already been taken as well as the output message for both the username and password as the password didn’t meet the password requirements.



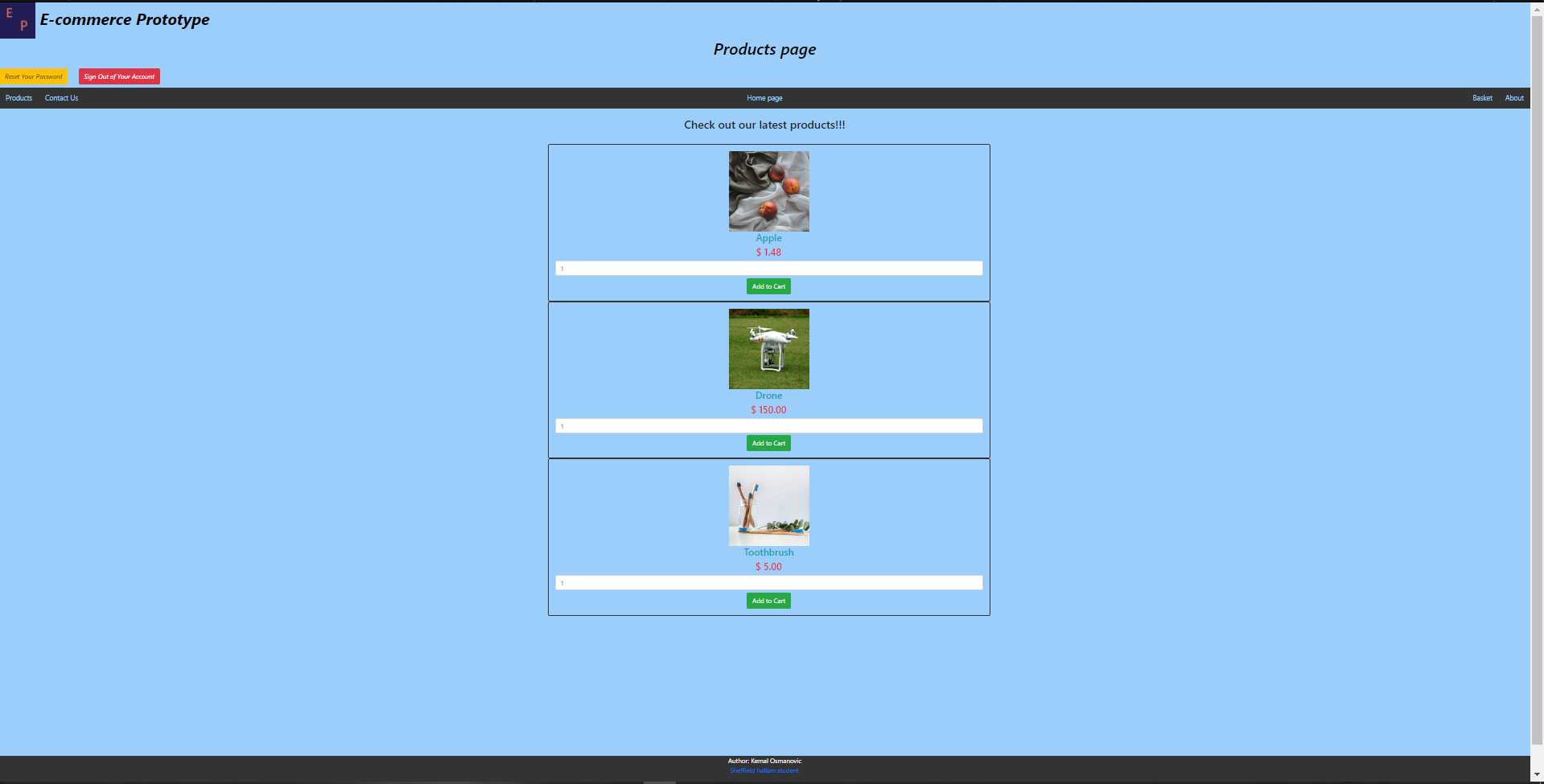
The screenshot above is taken from the homepage when the user signs in. The username is taken and combined with an output message to let the user know that the database recognises which user has just signed in. This was done using the code seen in the screenshot below which takes a variable that was assigned when the user signed in.

# Home page

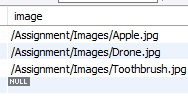


The screenshot above shows the homepage of my website. There are two buttons at the top which can allow the user to sign out and reset their password and this was duplicated and placed on every page. Also, the homepage has a slideshow which shows each different product.

# Products page

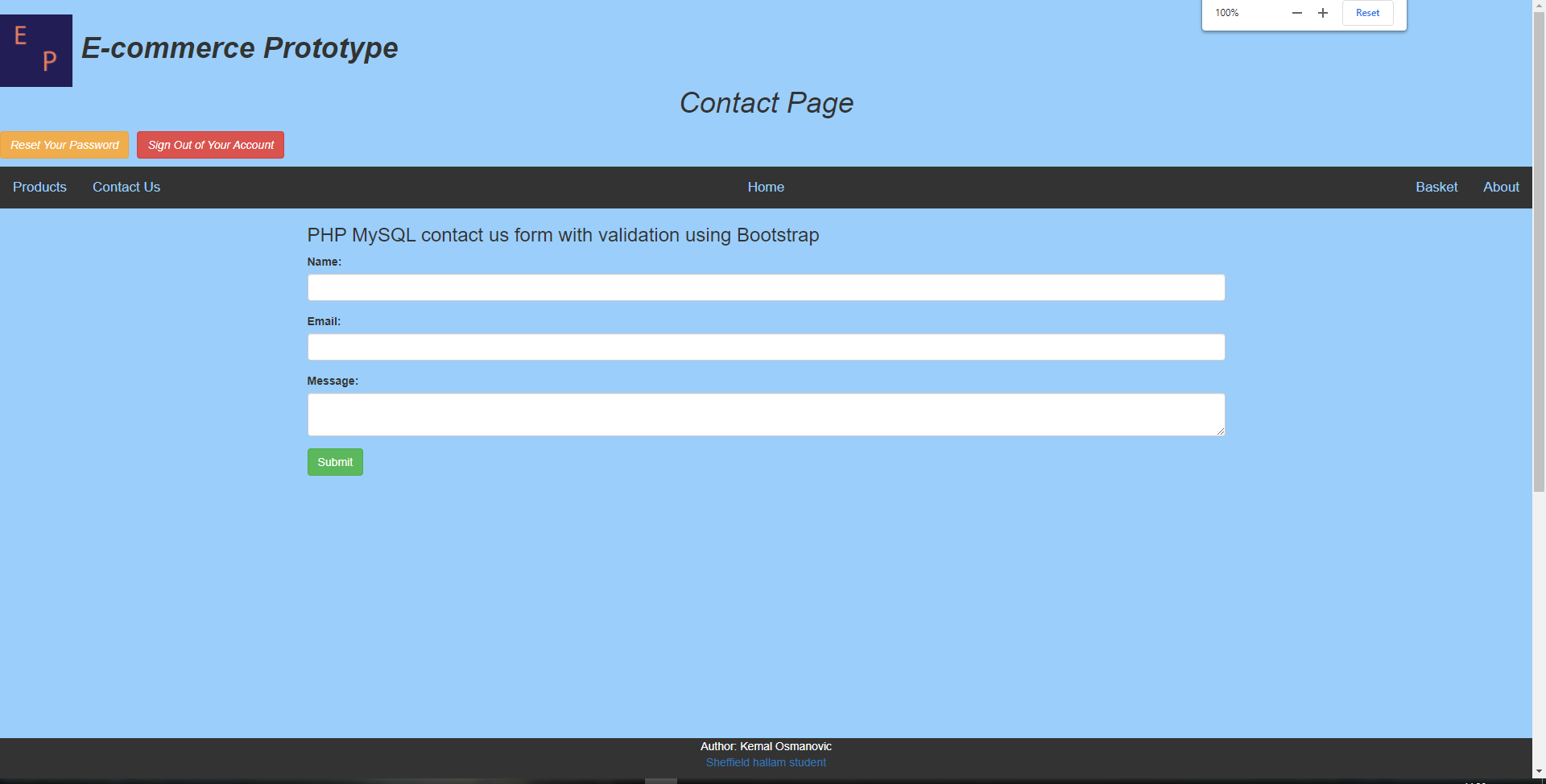


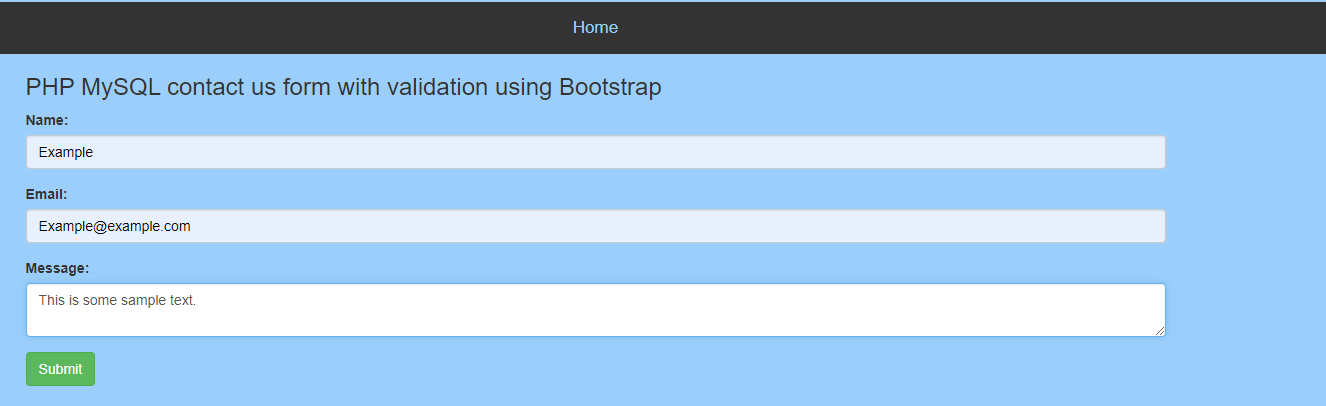
The screenshot above shows the products page which displays all the products for the user. The user can then enter the quantity of the item they want and click the button to add it to the cart. The product is displayed using the echo function in php which connects to the database and displays the value that was assigned to the corresponding image. This can be seen by the screenshots below.

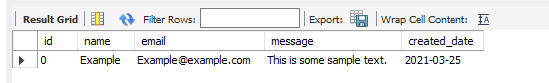


The screenshot to the left shows how the image is displayed as it located the file directory for the image.

# Contact us page



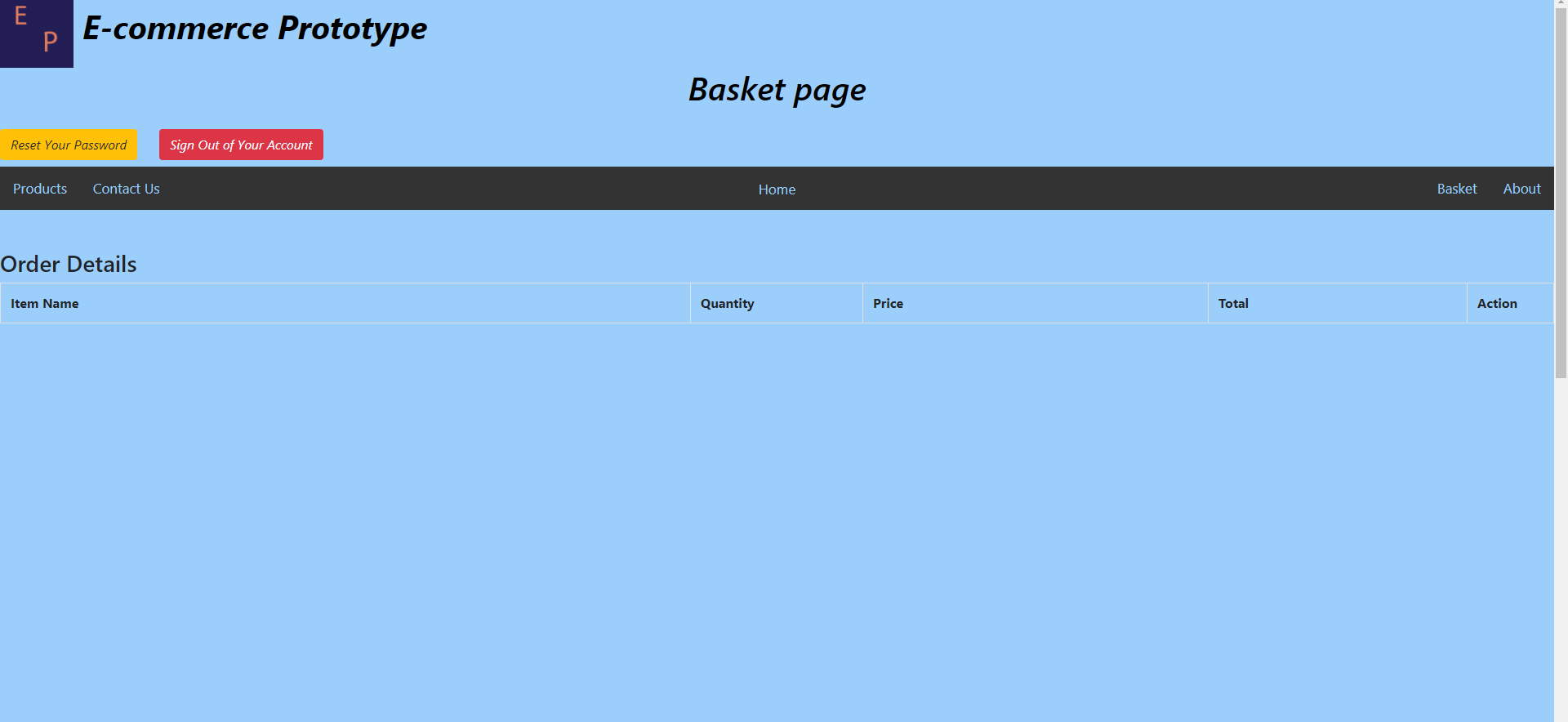
The above screenshot shows the contact page which the user can enter information in the fields that when submitted will be sent to the database. For example, below is a screenshot of before I enter some sample text before hitting submit. 

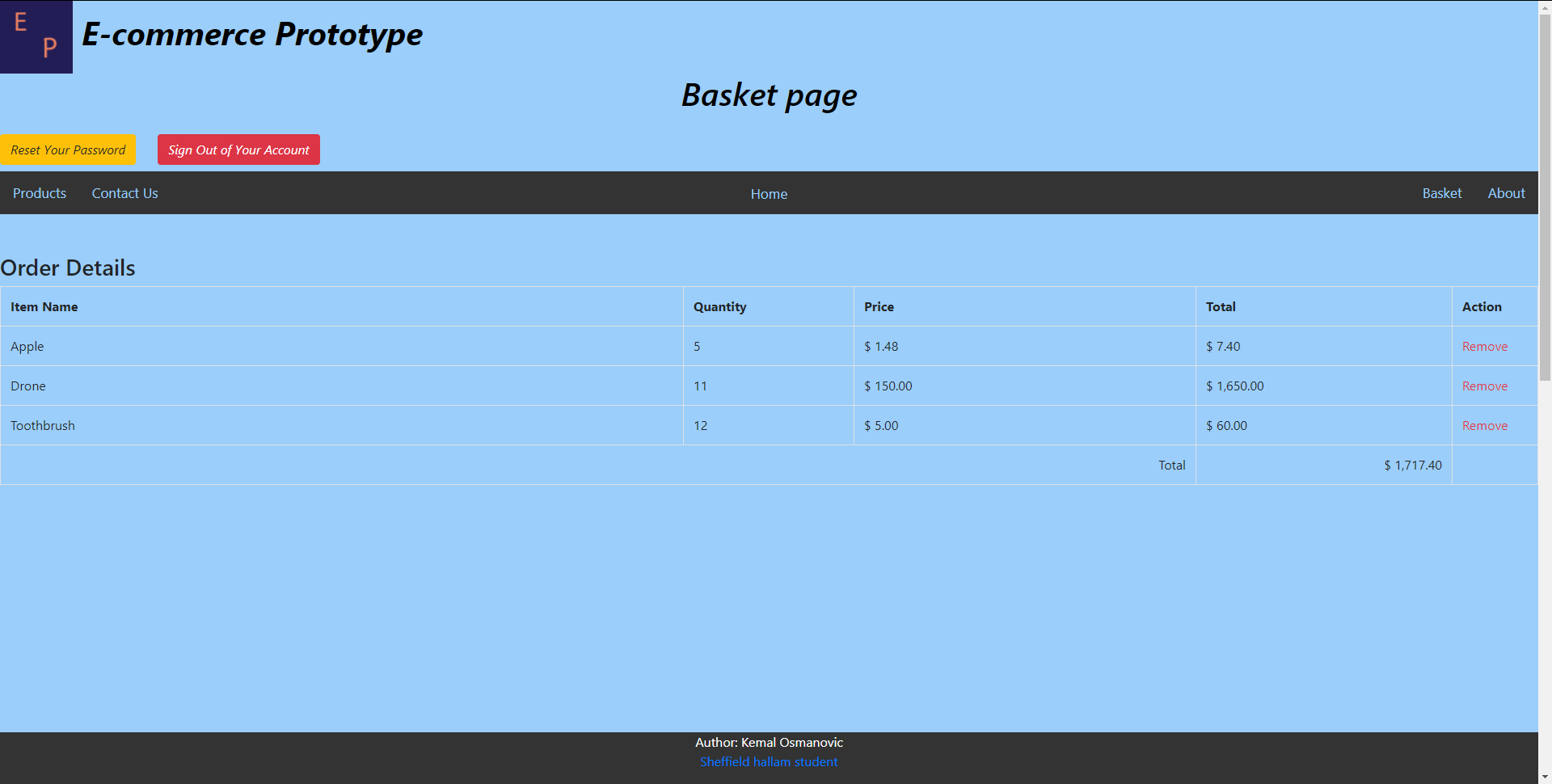
After the user clicks submit, this is then written to the database as can be seen by the screenshot below.



The screenshot above shows the php and sql query which is done by assigning the values of the entered fields to “$name”, “$email”, and “$message”. Also, the date is automatically generated when the data is submitted.

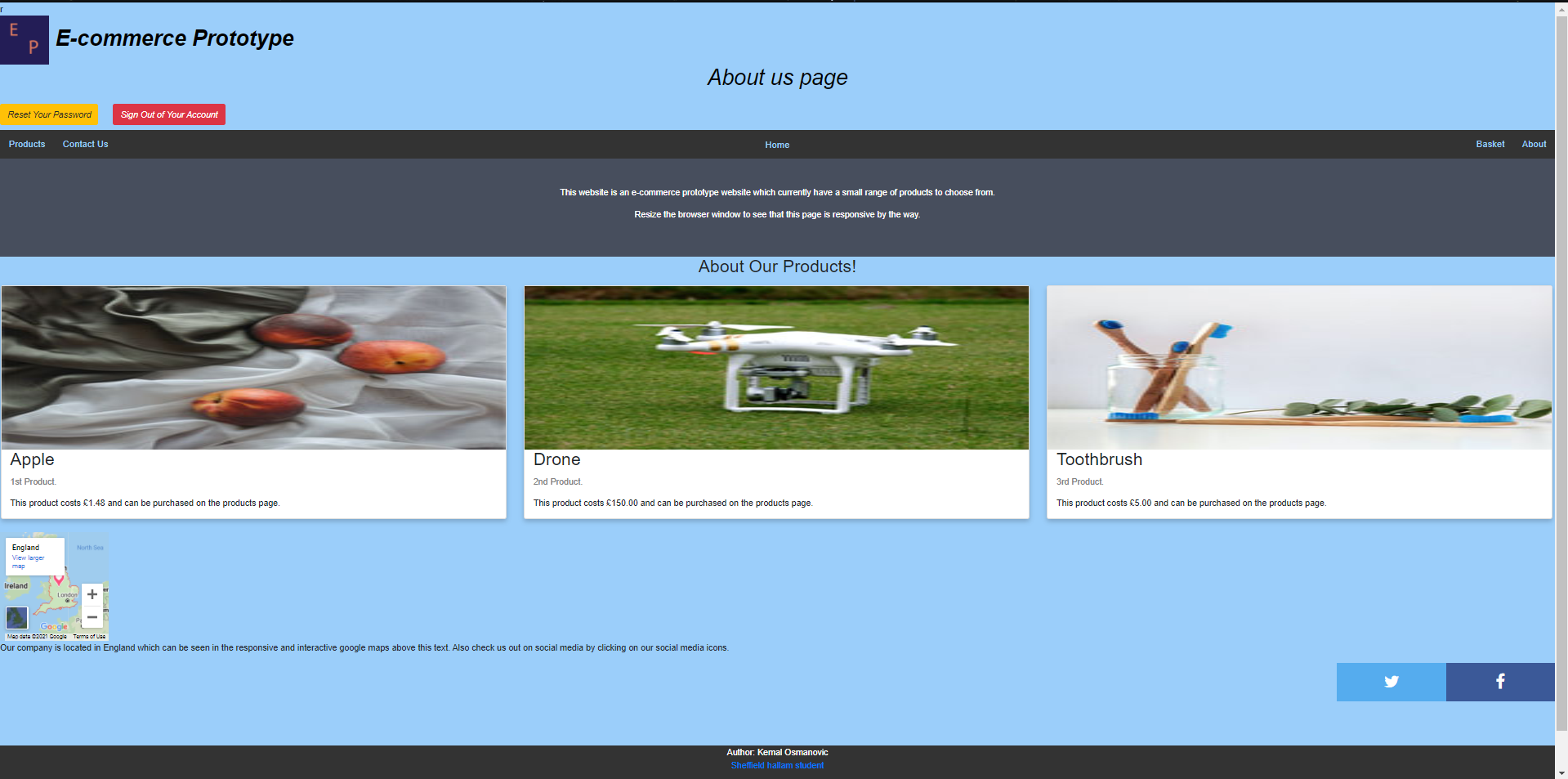
# Basket page



The above screenshot shows the basket page when the user hasn’t selected any product. Once the user has selected a product from the products page, it will look like the screenshot below. 

As you can see by the screeshot above, they have the option to remove the item. The total for each product is calculated by the formula “quantity x price” and then the overall total is calculated by adding up the total of each product. This can be seen by the screenshot below. Variables were pre-set for each product in the database and is called back using the php statement.

# About us page



The above screenshot is the about page. This has a title, 3 divisions which display each product and descriptions on each. It also has a google map of England and two social media icons which can be clicked which will redirect the user to the social media websites.

# Conclusion

In conclusion I believe I had created a good e-commerce prototype website using html, CSS, JavaScript, php and MySQL. I followed a few tutorials to be able to do the login page, the shopping cart, and the contact form. The links will be provided in the references section below. If I were to do this again, I would improve my naming conventions and make sure they are all similar and I would attempt to find a way to make the cart work by highlighting products based on attributes.

# References

“PHP MySQL contact us form with validation using Bootstrap.” (Hardik Savani, July 16, 2017).

ItSolutionStuff.com (2021). Used in the contact page and retrieved from: <https://www.itsolutionstuff.com/post/php-mysql-contact-us-form-with-validation-using-bootstrapexample.html>

“PHP MySQL Login System.” (tutorialrepublic.com, March 25th, 2021).

Tutorialrepublic.com (2021). Used in the login page, logout page and reset password page and retrieved from: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>

“Create Simple Shopping Cart using PHP and MySQL.” (web lesson, 3rd August 2021).

Weblesson.info (2021). Used in products and basket page. Retrieved from: <https://www.webslesson.info/2016/08/simple-php-mysql-shopping-cart.html>