GET IT THRIFTED

an ever growing community

Will Ferrante (CA-IDM-5.2)

PHP & Databases - Task 3

Building the Database - How it works?

To build my database I used PHPmyAdmin. This was used because it facilitates database creation and helps to better visualise the data structure.

The database creation process started by creating a detailed plan of all the data I required for my website. This helped me prepare better the tables, fields and keys that I had to create for my website to run smoothly and store data efficiently. By creating a detailed UML Diagram, I was able to visualise what fields all my tables had to have and how they will be linked to each other using multiple foreign keys.

The tables were set up to have the least amount of data repetion as possible. Multiple tables were used to create different codes used throughout the database (eg. Role table to supply roleId to the Users). This helped in keeping the size of the database as small as possible by including more numerical fields rather than text fields (where possible).

> As mentioned above, by creating a *Role Table* with two records, 1 - Seller and 2 - Buyer, in the *Users Table*, each record only included a number, 1 or 2, instead of a repetition of the words.

This might not have made such a big difference in a small structure like the one in my project but it is good practice especially when dealing with large volumes of data and records.

Primary and Foreign Keys in my Database

Every table in the database has 1 primary key which is unique to each record and is used to identify each record specifically. Apart from the Primary Key, each table has multiple foreign keys, used to link fields to other tables in the database. The foreign keys where also used to build paths through different tables.

> Each user in the *Users Table* has a Foreign Key called *streetId*. The numbers in this field match up to one record in the *Street Table*. However, in the *Street Table*, each street is linked to the *Town Table*, through the *townId*. Similarly, in the *Town Table*, each town is linked to one record in *Country Table* through the *countryId*.
>
> Therefore, with the use offoreign keys, when inputting a *streetId* in the *Users Table*, the admin is automatically linking that user to a town and a country.

Manupulating Data through the Website

To Manipulate data through my website, a series of functions was used throughout. A php file called db-function.php was created to gather all the functions in one place together and keep everything orgnaised.

Writing data in the database

Throughout the website, the user has multiple opportunities to fill out different forms. In the user side, the user can sign up and edit his account details and request to become a seller. In the admin side, the user can create new products and edit them. All these forms call different function that write data in different tables of the database. The funtions all working similarly, receive the data input from the user as parameters. The functions initialise a new statment object using the database connection. They prpare SQL statements to instert data into tables and bind the variables to parameters by specifying the datatypes of the variables which is important to be in the same order as the fields in the database. The function will next execute the statemtn and store it in a new variable, check for any errors and if everything works as it's supposed to redirects the user to a different page (could be the same page but updated).

Reading data from the database

The reading functions included an SQL statement to load all fields and records from differnet tables. A new statement object was initalized in every read function, that was later used for pepared statements. A simple if statement checked for errors and if no errors were present the statement was executed. Finally the function stored the result of the statement in a variable which was finally returned back.

Updating Data in the database

Similarly to the create function, the update function also receives data from user input. and uses the SQL Update statement to update data. The function like the ones before, initialises a new statement object using the database connection, checks for errors, binds variables to parameters, if no errors are present, executes the statement and redirects the user to a different page (could be the same page but updated).

Deleting data from the database

Delete functions work similarly to the create functions with the difference of using the SQL Delete rather than SQL Insert. The delete, removes a whole record from the database using the id as identification. The function initialises a statement, checks for errors, binds variables to parameters and if no errors are found executes the statement and redirects the user to a different page (could be the same page but updated).

Setting up the Virtual Server

In order to mimic a live server, a virtual server was set up locally on my device to host the website and database. This process involves replicating on your device the enivironment normally found on a live server. To do this I downloaded and set up the virtual server MAMP.

When successfuly downloaded I set up the server and my project in the following way:

- In the Applications folder, I located the MAMP folder which holds the htdocs folder. Here is where I created different folder for all the different applications I worked on during the semester, including my GET IT THRIFTED project.

- With my folder setup properly, now I could start coding by opening the folder in Visual Studio Code and creating all files and folders as required all in the same folder.

- When having the MAMP running on Apache, I could easily access the server through localhost:8888 and continuiting with the link to my project, in the case of GET IT THRIFTED, localhost:8888/getitthrifted. This gave me access to see the live site.

- The MAMP also gave me access to phpMyAdmin which was used to create and host the database. This was linked to my code to create a connection that could be used through the website.

Building a Dynamic Website

Having a dynamic website means having a website which is responsive and interactive to the user, thus creating a better user experience. A dynamic website is a website that changes as users interact with it. Some of the features of dynamic website include a real time inventory and personalization done by user.

To make my website dynamic, I made use of various different techniques

- **Having a Server side & User Side**
  My website has both a user side, which was created mainly using HTML, CSS & Javascript. This side of the website is what the user sees when visiting the website. It shows the user the products available, his cart and other useful information that the user might require.

  My website also has a Server side which is where the administrators of the website can manage their online shop. From this side, which is inaccessible to the user, admins can create, edit and delete products. They can approve new sellers and manage the requests. They can also track orders.

  PHP was used to link both side together with the database.

- **Having multiple ways to interact with the Database**
  Throughout my website, both the user and the admin have multiple ways of interacting with the database. The user can create new profiles and request to sell on the website, while the admin can edit the product list and the user roles.

- **User Login**
  The login feature differentiates between the two roles of users, buyers and sellers. When logging in, the website checks the role assigned to your user record and decides where to redirect you. If your role is buyer, when logging in you are left in the user-side, where you can view products and buy, while if your role is seller, logging in will redirect you to the server side where you have more options to manage the online shop.

- **Using Responsive Design**
  By using Bootstrap and a CSS Framework, my website is accessible and functional on a number of different devices, and can adapt itself to different screen sizes.

- **Version Control**
  By using Github and saving my website in a repository, I am able to track changes in my code and go back to older version of my website if required to.

Test Cases - Client Side

Input	Expected Output	Actual Output	Pass / Fail	Notes
Site URL (http://localhost:8888/getitthrifted/)	Load index page of Website	Index Page	☑	
Click on 'Products' in the Navbar	Load Products Page	Products Page	☑	
Click the 'View Details' Button	Show the Product details Page	Product Details Page	☑	
Click the 'Add to Cart' Button	Add item to Cart	Redirect to Cart Page with the item	☑	
Click on 'Gallery' in the Navbar	Load Gallery Page	Gallery Page	☑	
Click on 'Contact' in the Navbar	Load Contact Page	Contact Page	☑	
Input First Name	Show input in field	Input in field	☑	
Input Last Name	Show input in field	Input in field	☑	
Input Email Address	Show input in field	Input in field	☑	
Input Message	Show input in field	Input in field	☑	

Click the Send Button	Opens mail to send message	Opens Mail	☑	
Click on the Cancel Button	Empties the fields	Empties the fields	☑	
Click the ‘Shopping Cart’ Button	Show the Shopping Cart	Load the Shopping Cart	☑	
Click ‘Remove from Cart’ Button	Removes item from the cart	Removes the item from the cart	☑	Only visable if user has items in cart
‘View Products’ Button	Load Products Page	Load Products Page	☑	Only visable if user has no items in the cart
Click on ‘Continue Shopping’	Load Products Page	Load Products Page	☑	
Click the ‘Checkout’ Button	Show Login Screen	Load Login Screen	☑	Only if user is not logged in
Click the ‘Checkout’ Button	Show Checkout Screen	Checkout Screen with card details and order summary	☑	Only if user is logged in
Click the ‘Place Order’ Button	Show Success Message	Error	☒	Only if user is logged in
Click on ‘Wrong Card?’	Show Account Details	Account Details screen	☑	Only if user is logged in
Hover on the ‘Profile’ Icon	Show Menu	Show Menu	☑	
Click on ‘My Account’	Show Login Menu	Login Menu	☑	Only if user is not logged in

Input username	Show input in field	Input In field	☑	In the Login page when user is not logged in
Input Password	Show input (*) in field	Input (*) in field	☑	In the Login page when user is not logged in
Click the Log in Button	Log in user if user exists in database	Log in user if user exists in database	☑	In the Login page when user is not logged in. If user has roleId = 2, the page will redirect the user to the index. If user has roleId = 1, he will be redirected to the Admin page
Click on ‘Sign Up Now’	Load empty sign up form	Load empty sign up form	☑	In the Login page when user is not logged in
Input First Name	Show input in field	Show input in field	☑	
Input Surname	Show input in field	Show input in field	☑	
Input Date of Birth	Show input in field	Show input in field	☑	
Input Email Address	Show input in field	Show input in field	☑	
Input Username	Show input in field	Show input in field	☑	
Input Password	Show input (*) in field	Show input (*) in field	☑	

Input Confirm Passowrd	Show input (*) in field	Show input (*) in field	☑	
Input House Number and House Name	Show input in field	Show input in field	☑	
Choose Street from list	Show choice in field	Show choice in field	☑	
Choose Town from list	Show choice in field	Show choice in field	☑	
Choose Country from list	Show choice in field	Show choice in field	☑	
Input Card Number	Show input in field	Show input in field	☑	
Input Account Holder	Show input in field	Show input in field	☑	
Input CVV	Show input in field	Show input in field	☑	
Input Expiration Date	Show input in field	Show input in field	☑	
Choose Bank from list	Show choice in field	Show choice in field	☑	
Click on the ‘Signup’ Button	Enter new user in database with roleId = 2	Enter new user in database with roleId = 2	☑	
Click on ‘My Account’	Show Account Details	Show Account Details	☑	Only if user is logged in

Edit Name	Show edited input in field	Show edited input in field	☑	
Edit Surname	Show edited input in field	Show edited input in field	☑	
Edit Date of Birth	Show edited input in field	Show edited input in field	☑	
Edit Email Address	Show edited input in field	Show edited input in field	☑	
Edit Username	Show edited input in field	Show edited input in field	☑	
Edit Password	Show edited input (*) in field	Show edited input (*) in field	☑	
Edit Confirm Password	Show edited input (*) in field	Show edited input (*) in field	☑	
Click on ‘Update Personal Details’ Button	Save changes to users database	Save changes to users database	☑	
Edit Address	Show edited input in field	Show edited input in field	☑	
Edit Street Choice	Show edited choice in field	Show edited choice in field	☑	
Edit Town Choice	Show edited choice in field	Show edited choice in field	☑	
Edit Country Choice	Show edited choice in field	Show edited choice in field	☑	

Click on ‘Update Residence Details’ Button	Save changes to database	Save changes to databse	☒	
Edit Card Number	Show edited input in field	Show edited input in field	☑	
Edit Account Holder	Show edited input in field	Show edited input in field	☑	
Edit CVV	Show edited input in field	Show edited input in field	☑	
Edit Expiration Date	Show edited input in field	Show edited input in field	☑	
Edit Bank Choice	Show edited choice in field	Show edited choice in field	☑	
Click on ‘ Update Payment Details’ Button	Save Changes to database	Save Changes to database	☑	
Click on ‘Become a Seller’	Show Become a Seller Form	Show Become a Seller Form	☑	
Input username	Show input in field	Show input in field	☑	
Input Passoword	Show input (*) in field	Show input (*) in field	☑	
Input ‘Why do you want to sell with us?’	Show input in field	Show input in field	☑	
Input ‘Name of product you want to sell’	Show input in field	Show input in field	☑	

Input Description	Show input in field	Show input in field	☑	
Input Selling Price	Show input in field	Show input in field	☑	
Choose Size from list	Show choice in field	Show choice in field	☑	
Input Image Link	Show input in field	Show input in field	☑	
Click ‘Send Request’ Button	Send request and create new record in database	Send request and create new record in database	☑	

Test Cases - Server Side

Input	Expected Output	Actual Output	Pass / Fail	Notes
Click on Dashboard	Load Dashboard with the charts	Load Dashboard with the charts	☑	
Click on Products	Load Products Page	Products Page	☑	
Click on Page Roles	Load Page Users	Load Page Users	☑	
Click on Logout	Redirect to Login Page	Redirect to Login Page	☑	
Click on ‘Create New Product’ Button	Load Gallery Page	Gallery Page	☑	
Input Product Name	Show input in field	Input in field	☑	
Input Product Price	Show input in field	Input in field	☑	
Choose size from list	Show choice in field	Choice in field	☑	
Input Stock Quantity	Show input in field	Input in field	☑	
Input image Link	Show input in field	Input in field	☑	

Choose Seller from List	Show choice in field	Show choice in field	☑	
Click ‘Create Product’ Button	Add product to database	Add product to Database	☑	
Click the ‘Edit Details’ Button	Show Product Detials editable form	Show Product Detials editable form	☑	
Edit Name	Show edited input in field	Show edited input in field	☑	
Edit Price	Show edited input in field	Show edited input in field	☑	
Edit Size choice	Show edited choice in field	Show edited choice in field	☑	
Edit description	Show edited input in field	Show edited input in field	☑	
Edit Stock quantity	Show edited input in field	Show edited input in field	☑	
Edit Image Link	Show edited input in field	Show edited input in field	☑	
Edit Seller Choice	Show edited choice in field	Show edited choice in field	☑	
Click on ‘Update Product Details’ Button	Update product in database	Update product in database	☑	
Click on ‘Delete Product’ Button	Remove Product from database	Remove Product from database	☑	

Click ‘Remove Seller’ Button	Change user roleId to 2	Change user roleId to 2	☑	
Click ‘Approve’ Button	Remove request and change user roleId to 1	Remove request and change user roleId to 1	☑	Only available when there are pending requests
Click the ‘Deny’ Button	Remove request but do not change user roleId	Remove request but do not change user roleId	☑	Only available when there are pending requests