

FILES

MyMusicGear.php

This is the main file of the dynamic site, all other files should work through here

functions.php

This file contains all the classes and functions necessary for the site, this was done in the hopes of more cleanly separating the php and html where I could.

login.php

The interface for a user to log in

register.php

The interface for a user to register, the user can switch between client and admin types

home.php

The home page for admin type accounts

clientHome.php

The home page for client type accounts

addProduct.php

Interface for admins to insert products

listProducts.php

Interface for all users to view products. Displays differently depending on user type

search.php

Interface for users to search for products. Same for both user types but results display slightly differently

rentProduct.php

Interface for client to select their rental length and commit

returnProduct.php

Interface for client user to return products with a notice of costs

style.css

CSS file for styling

CLASSES

Entry

Functions for entry into the website

Validate

Functions for the validation of user information

ViewingProducts

Contains the Functions for the viewing of SQL database

Change Status

Functions for the altering SQL database

Calculations

Functions for calculations relating to rent costs

Errors

Error Function

DATABASE

user

Stores the data of user accounts

product

Stores the data of products

Notes

1. The Characteristics column stores data on the condition and colour of a product. Normally I would put these into their own columns as this is easier to understand and deal with, however the generic description of "characteristics" confused me in the assignment spec and I went with the thought that I had to have a column named characteristics specifically
2. If Admin changes the status of a product, this does not set a due date in rented or create a rentalrecords row as I considered this different to a "rental". I figured an admin may be making something unavailable to due damage to the product or whatever other reason.

rented

Stores the data of currently rented products

rentalrecords

Stores a record every time a product is rented

