

Technical manual

To implement this database the queries in the database.txt file must be run.

The database contains 7 tables. The people table contains the Drivers in the database. The vehicle table contains the vehicles owned by the drivers and is linked to the drivers using the ownership table. The Offence table contains all possible offences. The incident table contains the incident report for the drivers with a foreign key for the offence. The fines table contains the fines for those incidents that do have a fine with a foreign key for the Incident it is relating to. The user table contains a column called Admin which is set to 1 for admin users.

There are 16 PHP files in total. The way they work for this website is that the website will contain links that redirect to a different PHP file using PHPs header() function. There is a single database connection file called db_connection.php which contains the SQL connection details and is called in every single file that requires a sql connection. For the links in the tables returned from people, vehicle and incident search they redirect to the PHP file but also set the GET parameters so that the relevant rows pop up.

Files include:

- login.php
- mainpage.php
- People.php
- Vehicle.php
- db_connection.php
- session.php
- static.php
- passwordchange.php
- logout.php
- incident.php
- edit_incident.php
- view_incident.php
- delete.php
- create_user.php
- adddriver.php
- add_vehicle.php

Logging into the site works using PHP sessions. Once a user is logged in session variables are set which includes whether the user is an admin. There is a session.php file which checks whether you are currently in a PHP session for every file. If a session has not been started (not logged in) it will redirect to login.php. If you try to get to the login page but there is a session then it will redirect to mainpage.php. The logout.php file unsets the session variables, destroys the session and redirects to the login page.

The php and sql works using the GET and POST parameters in the html form. The search pages work by setting a variable usually called \$sql to the relevant sql query which has the GET parameter in a where clause. And then calling either mysqli_query or mysqli_multi_query. While loops are used to display all the information required.

To post items to the database this works in a similar way, a `$sql` variable is set to a query which has the POST parameter in a where clause and then the query is run using either `mysqli_query` or `mysqli_multi_query`. For some of the sql queries in the php files you may see `echo mysqli_error($conn)` which is used for testing purposes to see if the query hasn't worked and why not.

There is a `delete.php` file which contains delete queries. The query that is run depends on the GET parameter 'type' and the row deleted depends on the parameter 'id'.

There is a little bit of inline javascript used for this website, all it is doing is asking for confirmation whenever something is deleted.

The frontend was designed using the bootstrap framework.

There is a file called `static.php` which is called in most of the files which just puts the log out; main page and change password buttons at the bottom of the page.