Lab No: Project Part 4

Lab Title: Program and deploy a database-driven website

Objective: At the end of the lesson, students will be able to develop and deploy a min database-driven web application

Resources:

1. Development Computer.
2. Apache server, MySQL, and phpmyadmin
3. Code editor.

Procedures:

4. Launch the Apache server and MySQL on your development computer. Place the pess project folder and place it inside htdocs. The pess folder already contains pess_style.css and images folder that are provided to you for styling the pess web interfaces.
5. Accourding to the application requirements specification, there are a total of three tasks to be served by three web pages respectively.

Task No	Task Name	Task Description
1	Input Call by Operator	The system shall allow the operator to input information pertaining to an emergency call.
2	Dispatch Patrol Cars	The system shall display a list of available patrol cars for the operator to select and dispatch.
3	Update Patrol Car Status	The system shall allow the operator to update the status of each patrol car in the database.

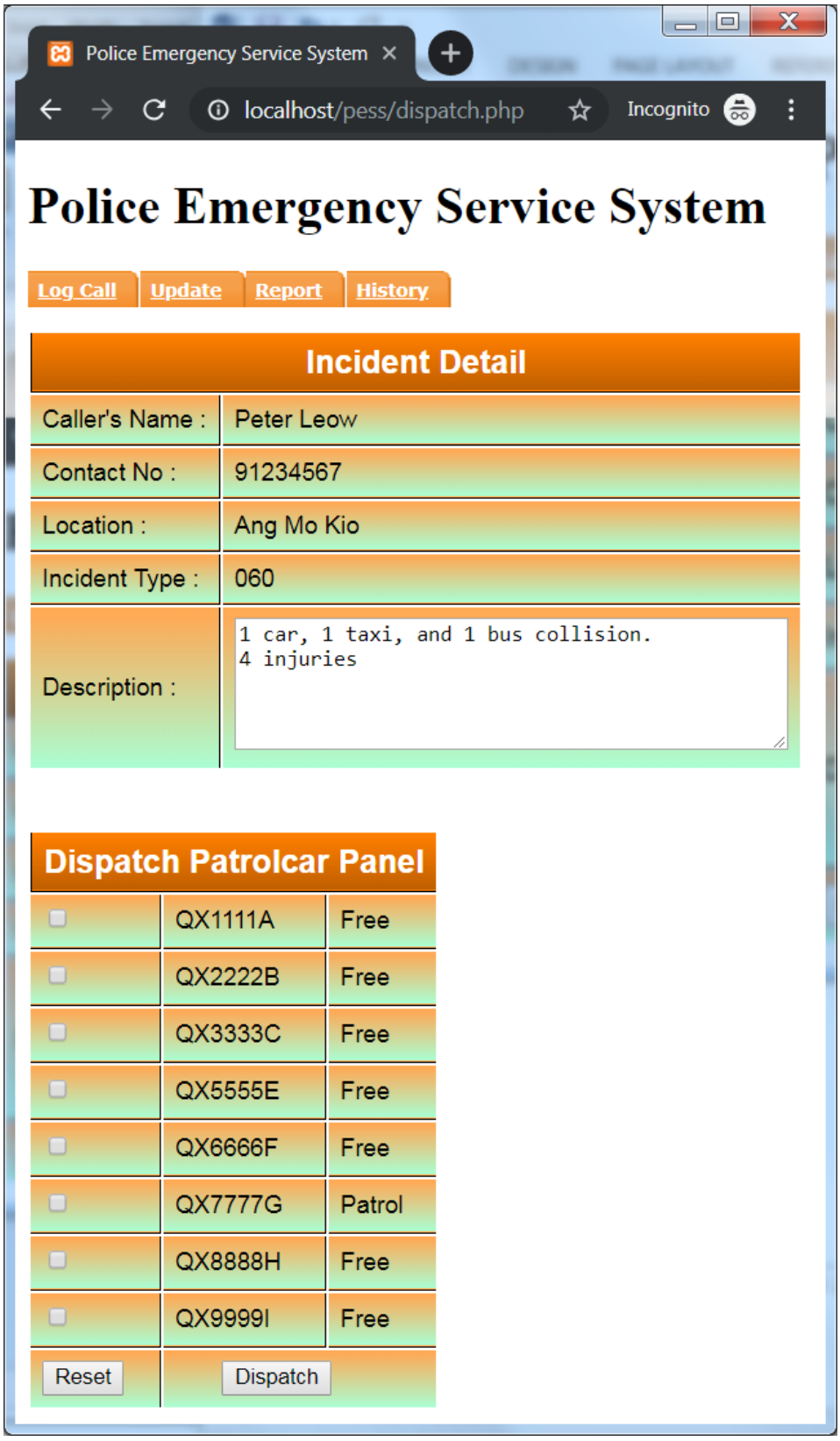
6. Tasks 1, 2, and 3 will be served by web page called **logcall.php**, **dispatch.php**, and **update.php** respectively as shown overleaf:

logcall.php

7. "logcall.php" page provides a form for the operator to enter details of each call received. Upon receiving a call, the operator will enter the call details via this **Log Call Panel** screen. Clicking the **Process Call…** button will forward the form data to the next page "dispatch.php" for dispatching of patrol cars.

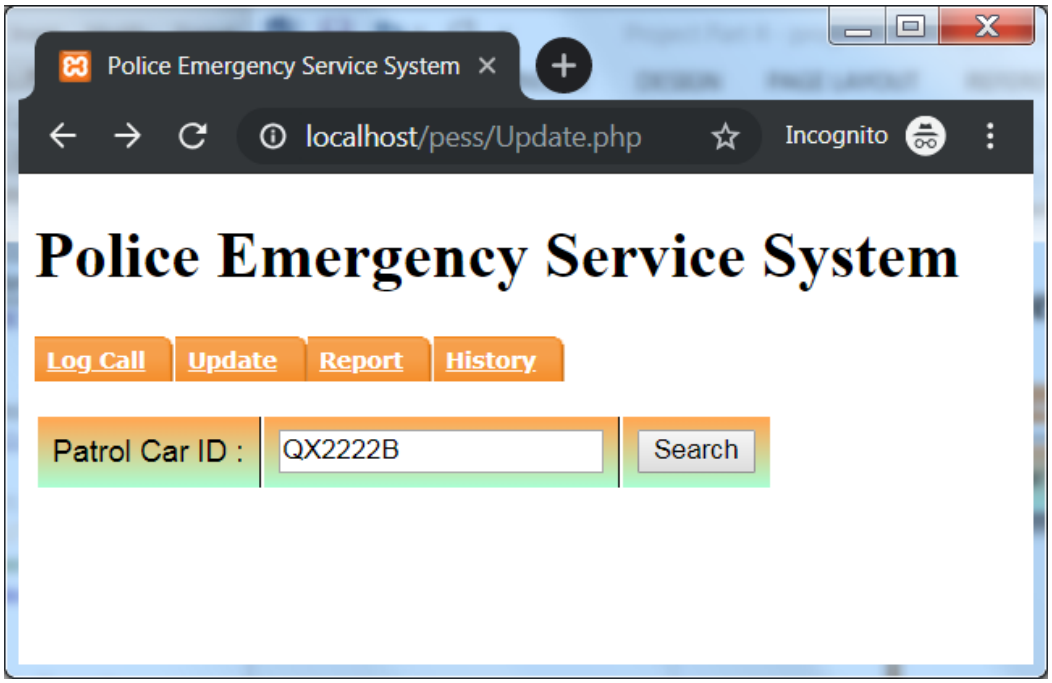
dispatch.php

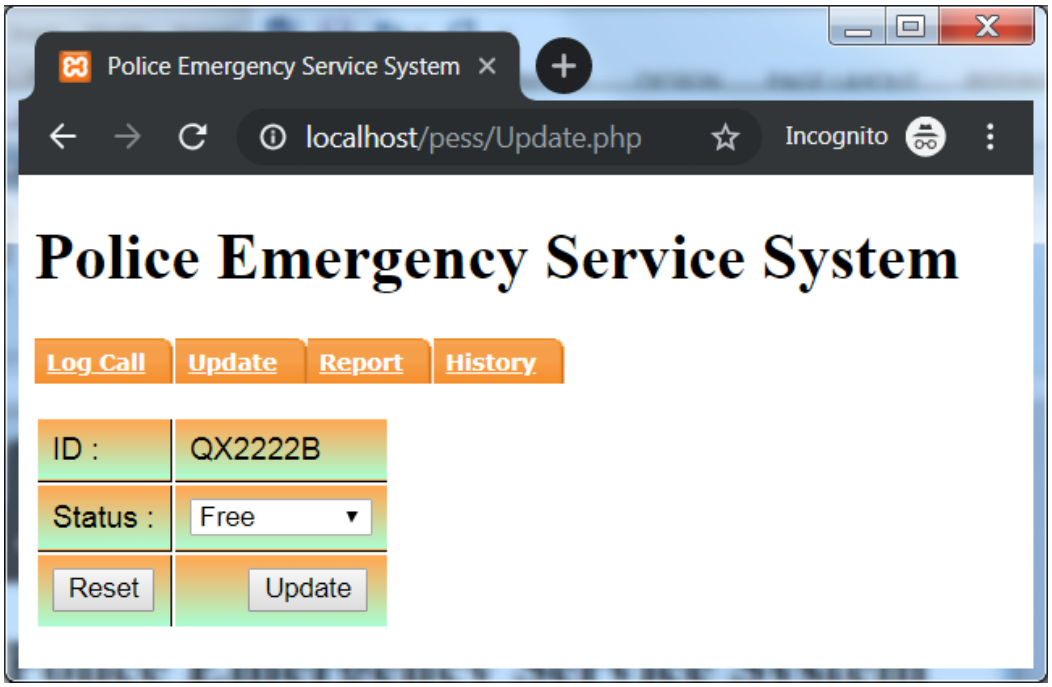
8. This page will list those patrol cars that are available for dispatch to deal with the incident logged in logcall.php.



update.php

9. This page allows the operator to update the status of a patrol car.

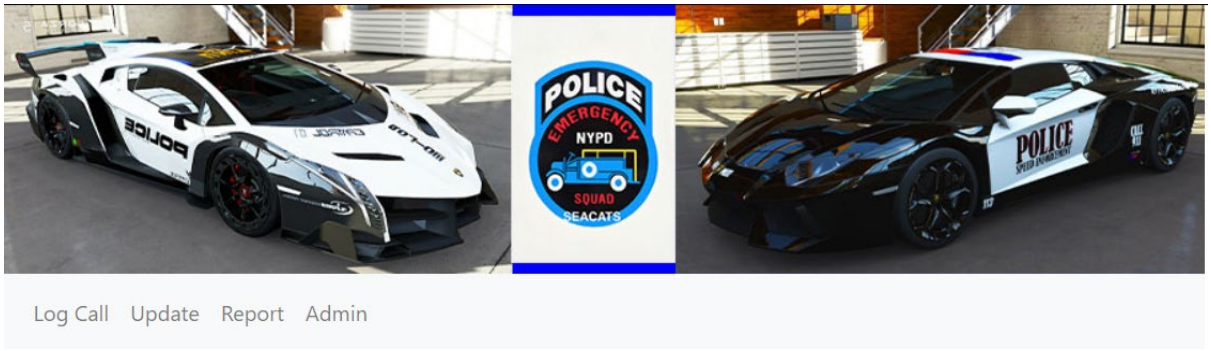




10. Notice that these three web pages share the same heading and navigation menu. In fact, they also share the same database connection parameters. Instead of repeating the code for creating such common features in the three web pages, what would you suggest? What are the merits of your suggestion? You may discuss with your classmates. Write them down:

Suggested answer: code the common features in separate files and include them in the three web pages using php **require_once** reusability ease of maintenance

11. You will write the code to create the navigation menu as shown:



12. The **log Call** link will navigate to **logcall.php** and the **Update** link to **update.php**.

13. You should save this code in a php file called **nav.php**.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item">
        <a class="nav-link" href="logcall.php">Log Call</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="search.php">Update</a>
```

```
        </li>
                <li class="nav-item">
          <a class="nav-link" href="report.php">Report</a>
        </li>
                <li class="nav-item">
          <a class="nav-link" href="admin.php">Admin</a>
        </li>
       </ul>


     </div>
    </nav>
```

14. Continue to write the code to define the database connection parameters in a new php file and save it as **db.php**.

```
 <?php // define database connection variables as constant
 define('DB_USER', "root"); // db user
 define('DB_PASSWORD', ""); // db password
 define('DB_DATABASE', "pessdb"); // database name
 define('DB_SERVER', "localhost"); // db server
?>
```

15. By the end of this lesson, you would have create the **nav.php** and **db.php**. You will continue to work on **logcall.php** next time.