PROJECT REPORT

DATABASE MANAGEMENT SYSTEMS (CMPS 460)

University of Louisiana at Lafayette

Project title: DBMS Application for Covid-19 Patients in Lafayette Hospital

Group members: Jamal Ahmadov (C00423516)

Xiaoqing Yan (C00290603)

Submission date: November 20, 2020

Project Requirements

As the Covid-19 pandemic unfolds across the United States, one of the greatest barriers we are encountering is the absence of credible and consistent data records. Tracking daily hospitalization data is a major step forward in quantifying the current impact on local hospital systems, modeling and forecasting future utilization needs, and tracking the rate of change in the disease severity. Doctors, nurses, and other healthcare professionals have recently been overwhelmed with patients who have tested positive for COVID-19, and also with those who are concerned that their symptoms may be indicative of COVID-19. It is necessary to retain these patients' information in a database that is dedicated to COVID-19, rather than just on the general patient database.

The database is designed to keep records of COVID-19 patients, doctors and nurses in the hospital. It includes patients' essential information, visit history, pre-test symptoms and underlying conditions if any. The database also includes hospital staff information, lab test results, and inventory of personal protective equipment (PPE) and Covid-19 medicine.

The database is designed to perform the following interactions:

1. Log of patients' visits and a description of symptoms when a patient was admitted.
2. List of patient's essential information and doctor and nurse information responsible for the patient.
3. Report the diagnosis after the patient was tested, whether or not patient was hospitalized, and the hospital room assigned.
4. List of personal protective equipment (PPE) and Covid-19 drugs, such as remdesivir, or dexamethasone, their quantity and availability.
5. Testing documentation including type of test for asymptomatic (rapid test) and symptomatic patient (standard test), test results, and testing frequency.
6. Insert a new record into or update and delete existing records in the tables containing all the information above.

Data Modeling

We chose Entity Relationship model (i.e. E/R diagram) as a preferred method for high-level conceptual modeling. The finalized E/R diagram is given below.

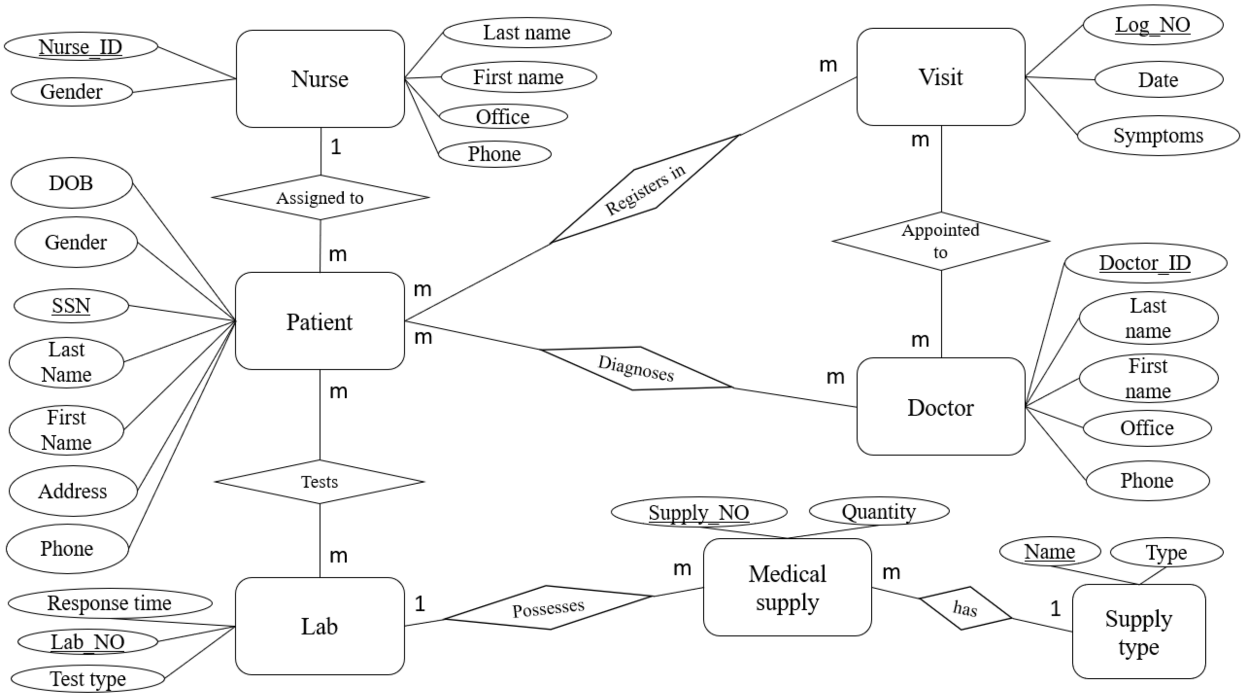


Figure 1. ER Diagram of the Database.

Due to the space restriction, the attributes of relationships were not shown in the E-R diagram. However, they will be given in the low level conceptual model. The E-R diagram was converted to the relational database schema as shown below.

Visit (Log_no [PK], Date, Symptoms)

Patient (SSN [PK], Last name, First name, Gender, DOB, Address, Phone, Nurse_ID [FK])

Doctor (Doctor_ID [PK], Last name, First name, Office, Phone)

Nurse (Nurse_ID [PK], Last name, First name, Gender, Office, Phone)

Lab (Lab_no [PK], Test type, Response time)

Medical supply (Supply_NO [PK], Quantity, Name[FK], Lab_no [FK])

Supply_type (Name[PK], type)

Diagnoses (Diagnosis, Hospitalization, Treatment period, Room assigned, Test frequency, SSN [FK], Doctor_ID [FK])

Registers (SSN [FK], Log_no [FK])

Appointed (Doctor_ID [FK], Log_no [FK])

Tests (Result, Date, Test number, SSN [FK], Lab_no [FK])

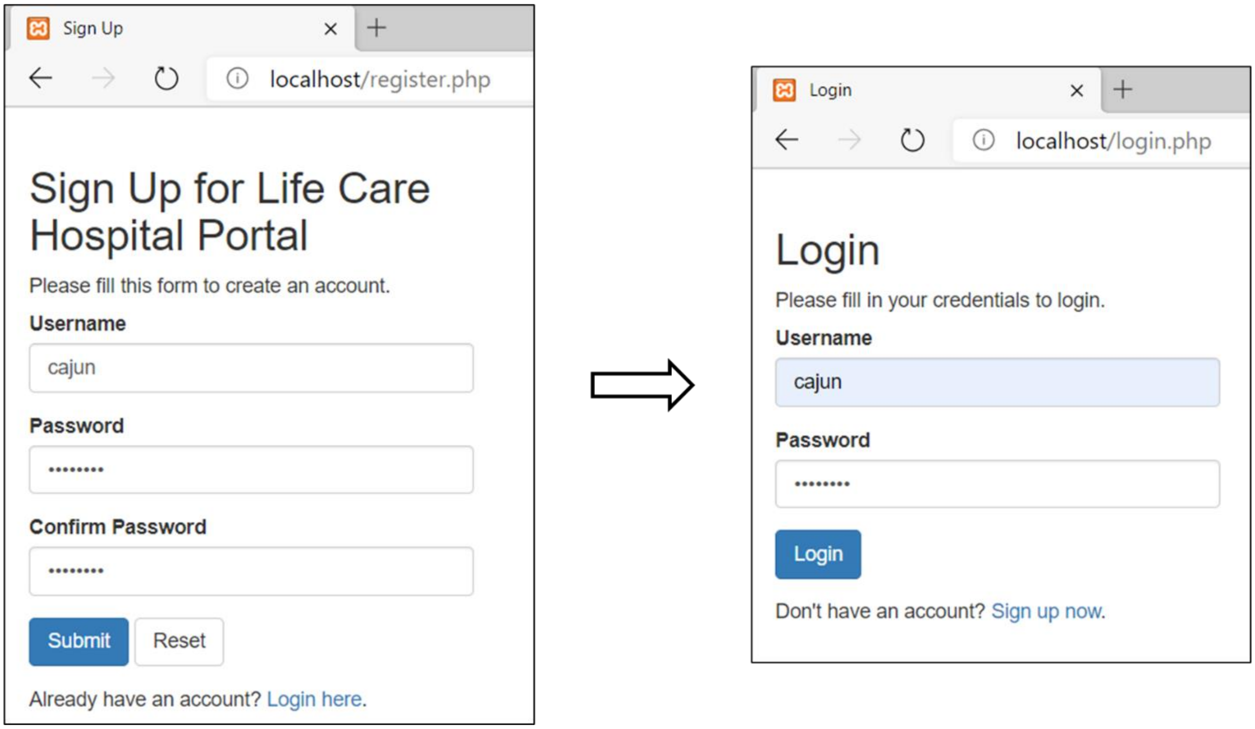
Normalization

In each table, the attributes are atomic. We did not use composite primary keys, since there are no composite primary keys, the primary key is the whole of primary key, therefore the tables are in 2NF. Moreover, there is no functional dependency between non-key attributes, that is, there is no transitive dependency, which makes our tables in 3NF.

Implementation

XAMPP was used to build this application. XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, and it allows to design an application on a local web server on the computer. The two essential components of XAMPP are Apache, which is used to create the local server, and MySQL which is used as a database management system for the application. PHP which is a server scripting language is used to connect to and manipulate the database. The content and structure of the application web pages were designed using HTML and CSS.

The first page of the application is a sign-up page. Once the account is created by setting username and password it will be redirected to the login page. After typing in the correct username and password, it will be redirected to the home page.



Sign Up

localhost/register.php

Sign Up for Life Care Hospital Portal

Please fill this form to create an account.

Username: cajun

Password: ••••••••

Confirm Password: ••••••••

Already have an account? Login here.

Login

localhost/login.php

Login

Please fill in your credentials to login.

Username: cajun

Password: ••••••••

Don't have an account? Sign up now.

Figure 2. a) Sign Up Page, b) Login page.

The home page of the website is given below. This is the page the user will be directed to after signing in.

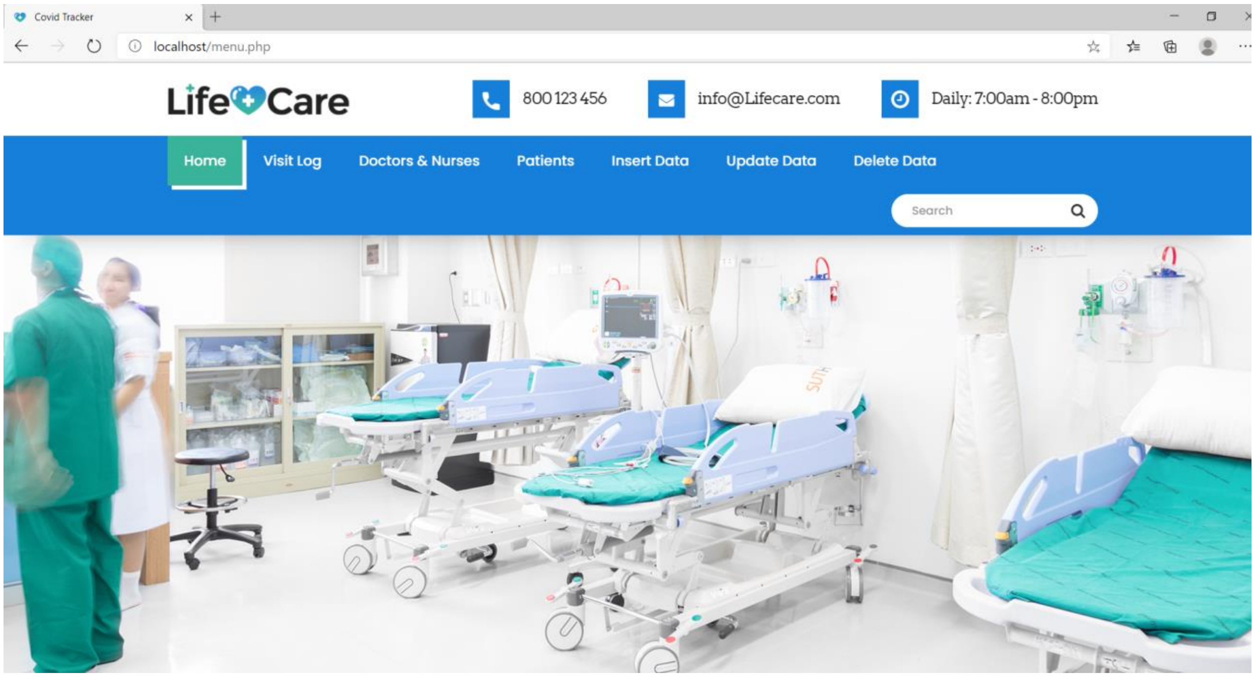
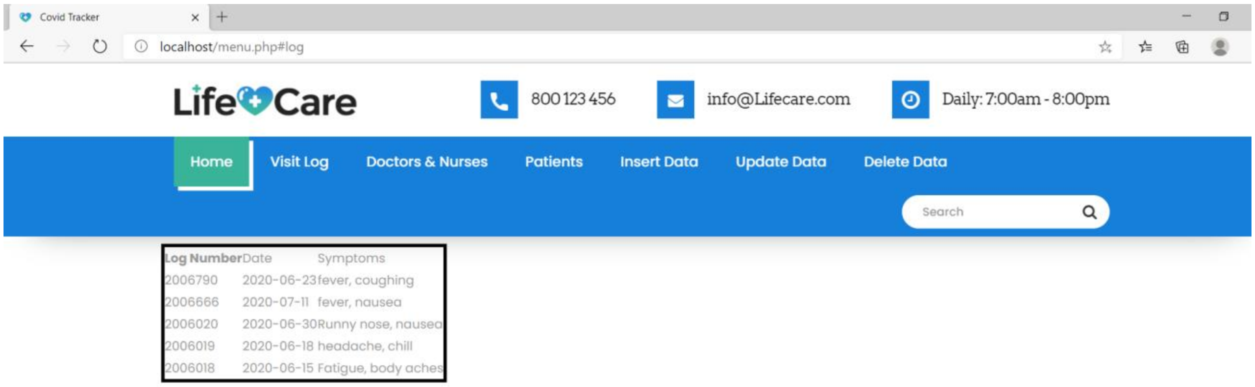


Figure 3. The Home Page.

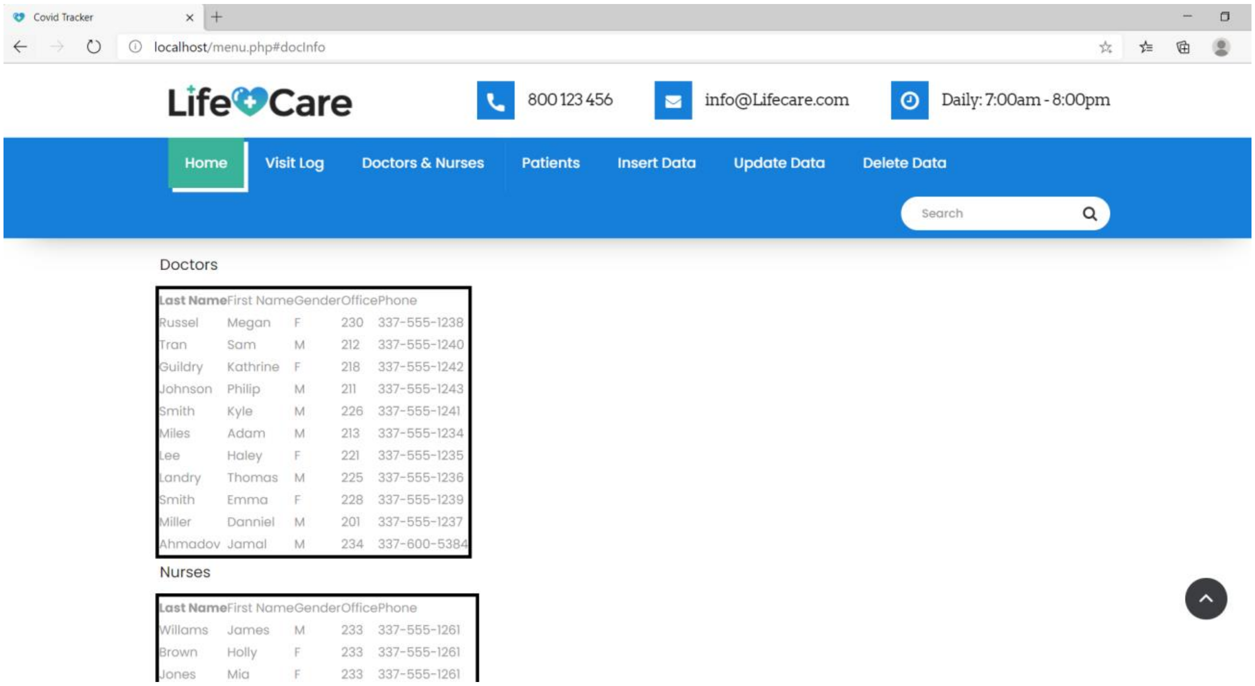
When clicking “Visit Log”, it scrolls down to Log section, which shows the 5 most recent records.



Log Number	Date	Symptoms
2006790	2020-06-23	fever, coughing
2006666	2020-07-11	fever, nausea
2006020	2020-06-30	Runny nose, nausea
2006019	2020-06-18	headache, chill
2006018	2020-06-15	Fatigue, body aches

Figure 4. Visit Log Tab.

When scrolling down or clicking “Doctors & Nurses”, their information is displayed.



Doctors

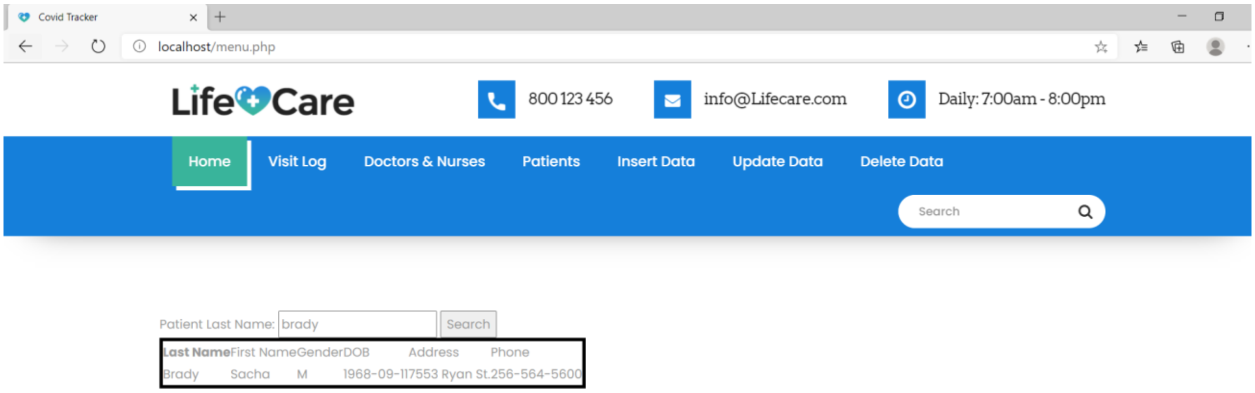
Last Name	First Name	Gender	Office	Phone
Russel	Megan	F	230	337-555-1238
Tran	Sam	M	212	337-555-1240
Guildry	Kathrine	F	218	337-555-1242
Johnson	Philip	M	211	337-555-1243
Smith	Kyle	M	226	337-555-1241
Miles	Adam	M	213	337-555-1234
Lee	Haley	F	221	337-555-1235
Landry	Thomas	M	225	337-555-1236
Smith	Emma	F	228	337-555-1239
Miller	Danniel	M	201	337-555-1237
Ahmadov	Jamal	M	234	337-600-5384

Nurses

Last Name	First Name	Gender	Office	Phone
Willams	James	M	233	337-555-1261
Brown	Holly	F	233	337-555-1261
Jones	Mia	F	233	337-555-1261

Figure 5. Doctors & Nurses Tab.

When clicking “Patients”, it goes to the search section to look up patients by last name.

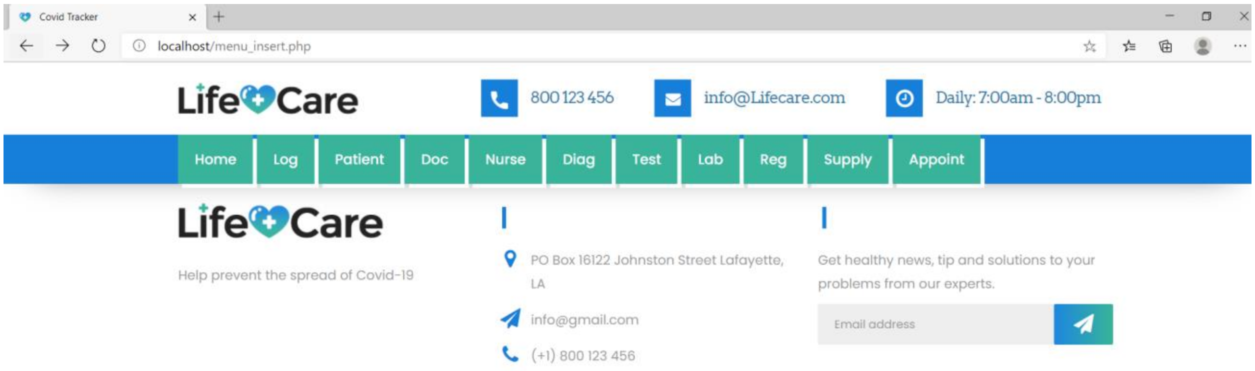


Patient Last Name: brady Search

Last Name	First Name	Gender	DOB	Address	Phone
Brady	Sacha	M	1968-09-11	7553 Ryan St.	256-564-5600

Figure 6. Patients Tab.

When clicking “Insert Data” on Home Page, it is redirected to the insert page.



Help prevent the spread of Covid-19

PO Box 16122 Johnston Street Lafayette, LA

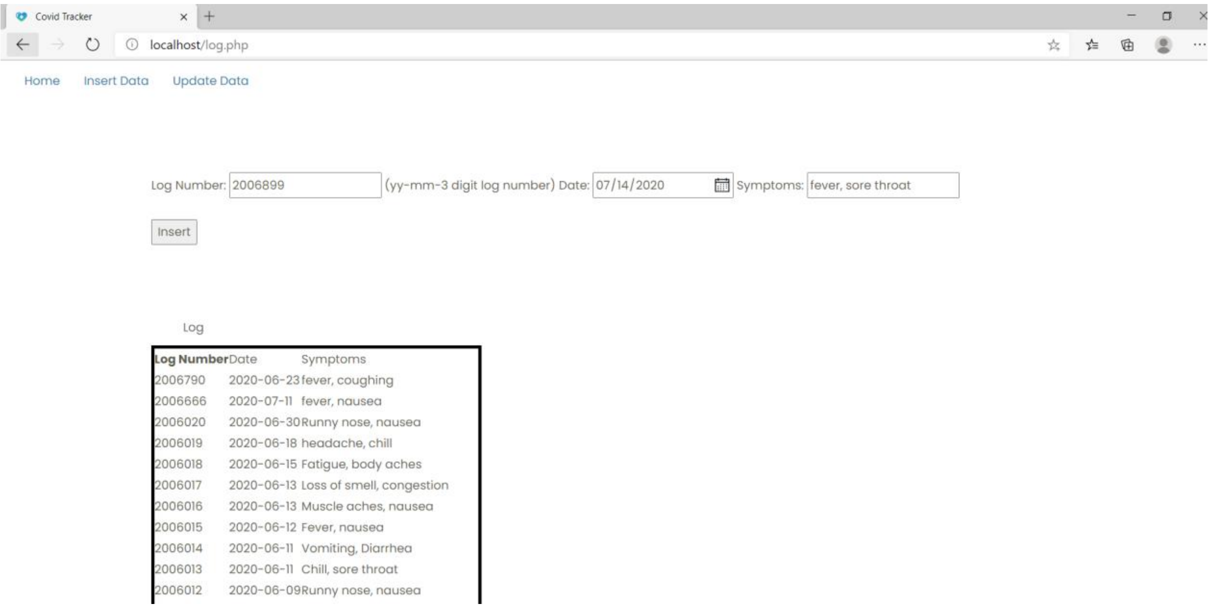
info@gmail.com

(+1) 800 123 456

Get healthy news, tip and solutions to your problems from our experts.

Figure 7. Insert Data Tab.

When clicking “Log” on Insert Page, it is redirected to the Log Insert Page. Input Log Number, Date, and Symptoms and click “Insert” button.



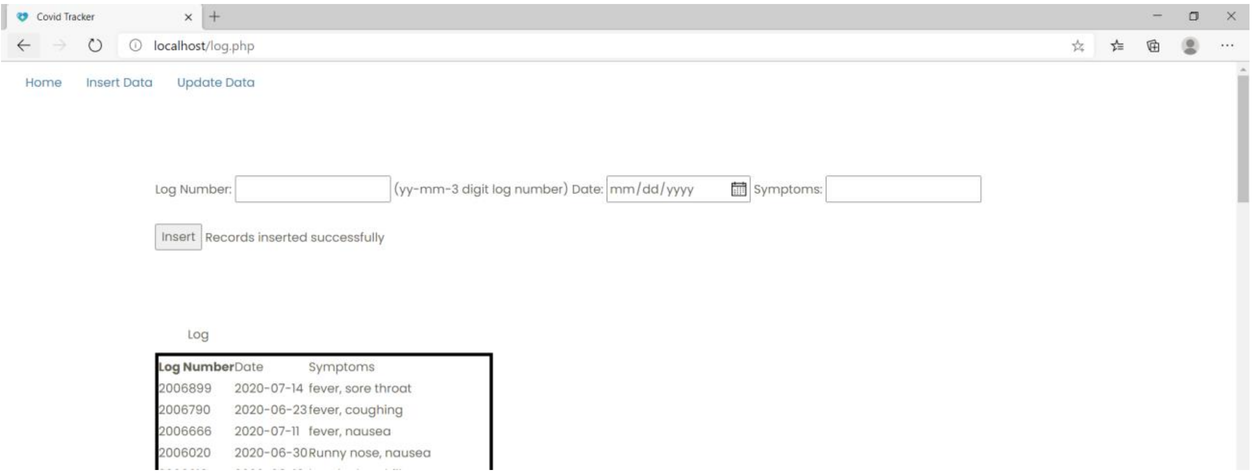
Log Number: 2006899 (yy-mm-3 digit log number) Date: 07/14/2020 Symptoms: fever, sore throat

Log

Log Number	Date	Symptoms
2006790	2020-06-23	fever, coughing
2006666	2020-07-11	fever, nausea
2006020	2020-06-30	Runny nose, nausea
2006019	2020-06-18	headache, chill
2006018	2020-06-15	Fatigue, body aches
2006017	2020-06-13	Loss of smell, congestion
2006016	2020-06-13	Muscle aches, nausea
2006015	2020-06-12	Fever, nausea
2006014	2020-06-11	Vomiting, Diarrhea
2006013	2020-06-11	Chill, sore throat
2006012	2020-06-09	Runny nose, nausea

Figure 8. Insert a Record on Log.

The inserted record is displayed on top of the list.



Log Number: (yy-mm-3 digit log number) Date: mm/dd/yyyy Symptoms:

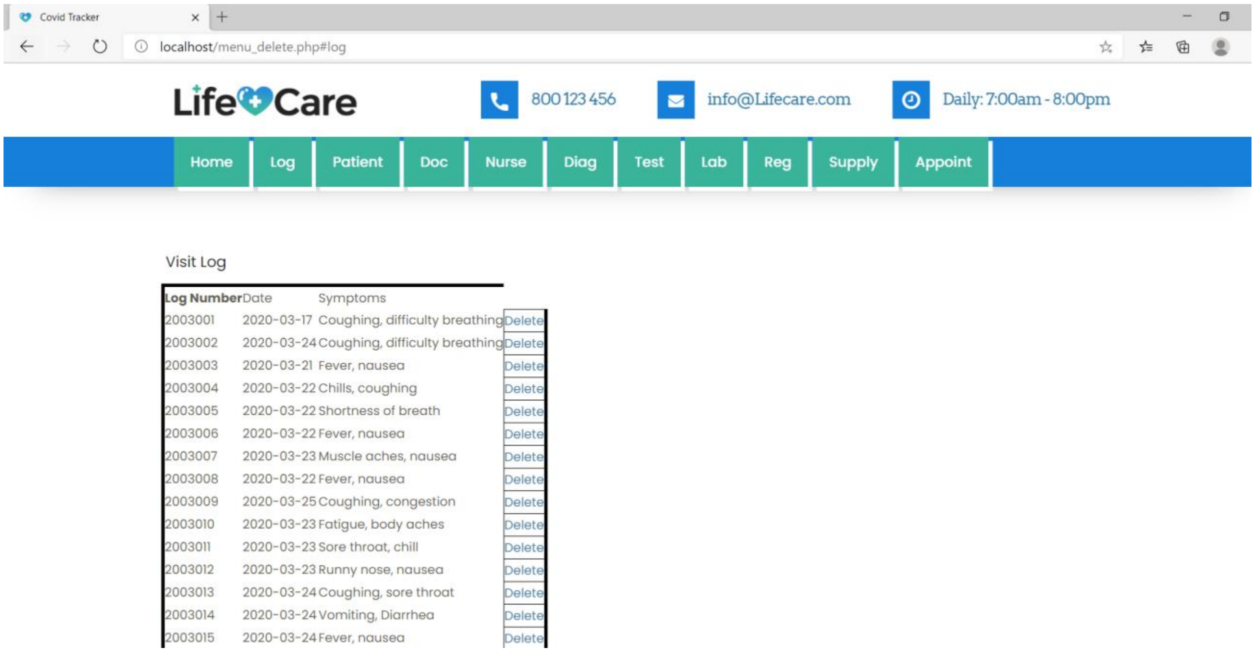
Records inserted successfully

Log

Log Number	Date	Symptoms
2006899	2020-07-14	fever, sore throat
2006790	2020-06-23	fever, coughing
2006666	2020-07-11	fever, nausea
2006020	2020-06-30	Runny nose, nausea

Figure 9. Record Inserted.

From home page, when first clicking “Delete Data” and then “Log”, it moves to the Log table section, with a “Delete” button next to each record. “Delete” should be clicked to delete the corresponding record.

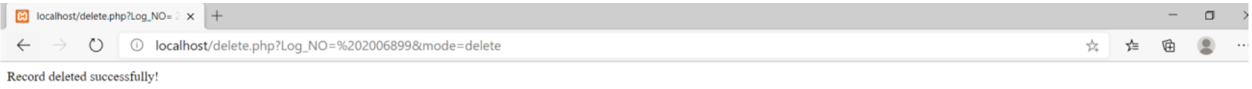


Visit Log

Log Number	Date	Symptoms	
2003001	2020-03-17	Coughing, difficulty breathing	Delete
2003002	2020-03-24	Coughing, difficulty breathing	Delete
2003003	2020-03-21	Fever, nausea	Delete
2003004	2020-03-22	Chills, coughing	Delete
2003005	2020-03-22	Shortness of breath	Delete
2003006	2020-03-22	Fever, nausea	Delete
2003007	2020-03-23	Muscle aches, nausea	Delete
2003008	2020-03-22	Fever, nausea	Delete
2003009	2020-03-25	Coughing, congestion	Delete
2003010	2020-03-23	Fatigue, body aches	Delete
2003011	2020-03-23	Sore throat, chill	Delete
2003012	2020-03-23	Runny nose, nausea	Delete
2003013	2020-03-24	Coughing, sore throat	Delete
2003014	2020-03-24	Vomiting, Diarrhea	Delete
2003015	2020-03-24	Fever, nausea	Delete

Figure 10. Delete Data Page.

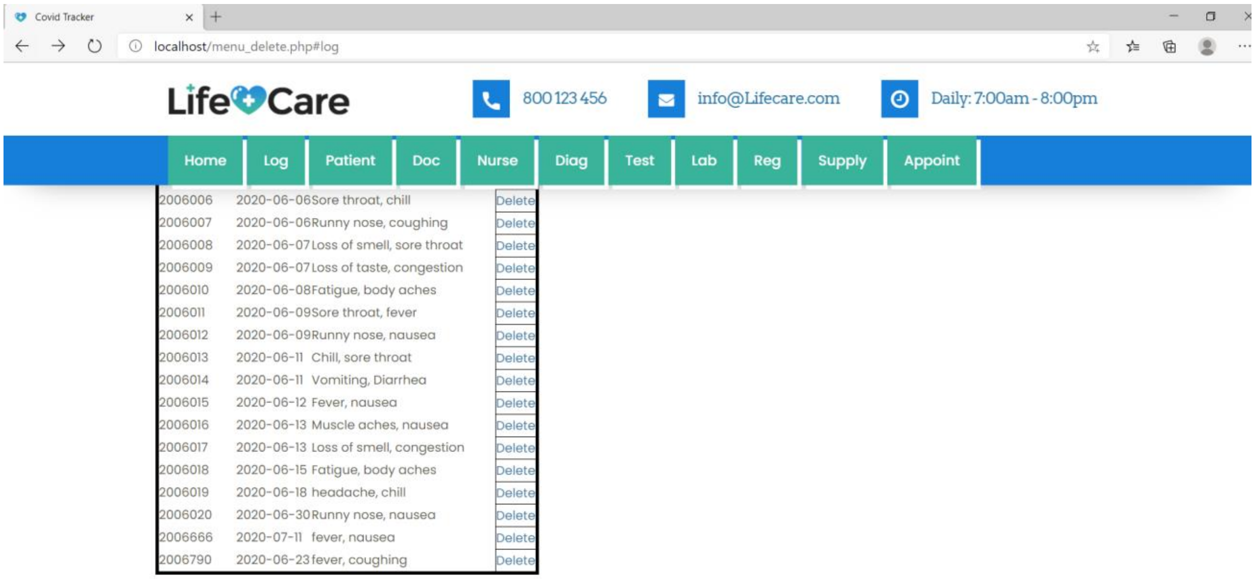
It displayed “Record deleted successfully” message when clicking the delete button on a record that we inserted above (Log Number =2006899).


localhost/delete.php?Log_NO=
localhost/delete.php?Log_NO=%202006899&mode=delete

Record deleted successfully!

Figure 11. Delete Result Display.

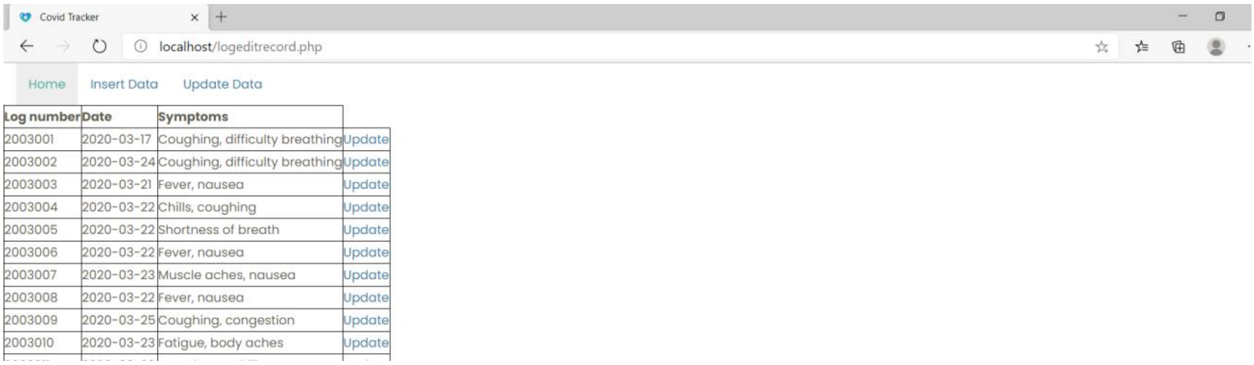
The record for 2006899 has been deleted, and no longer displayed in the list.



2006006	2020-06-06	Sore throat, chill	Delete
2006007	2020-06-06	Runny nose, coughing	Delete
2006008	2020-06-07	Loss of smell, sore throat	Delete
2006009	2020-06-07	Loss of taste, congestion	Delete
2006010	2020-06-08	Fatigue, body aches	Delete
2006011	2020-06-09	Sore throat, fever	Delete
2006012	2020-06-09	Runny nose, nausea	Delete
2006013	2020-06-11	Chill, sore throat	Delete
2006014	2020-06-11	Vomiting, Diarrhea	Delete
2006015	2020-06-12	Fever, nausea	Delete
2006016	2020-06-13	Muscle aches, nausea	Delete
2006017	2020-06-13	Loss of smell, congestion	Delete
2006018	2020-06-15	Fatigue, body aches	Delete
2006019	2020-06-18	headache, chill	Delete
2006020	2020-06-30	Runny nose, nausea	Delete
2006666	2020-07-11	fever, nausea	Delete
2006790	2020-06-23	fever, coughing	Delete

Figure 12. Record Removed from Log.

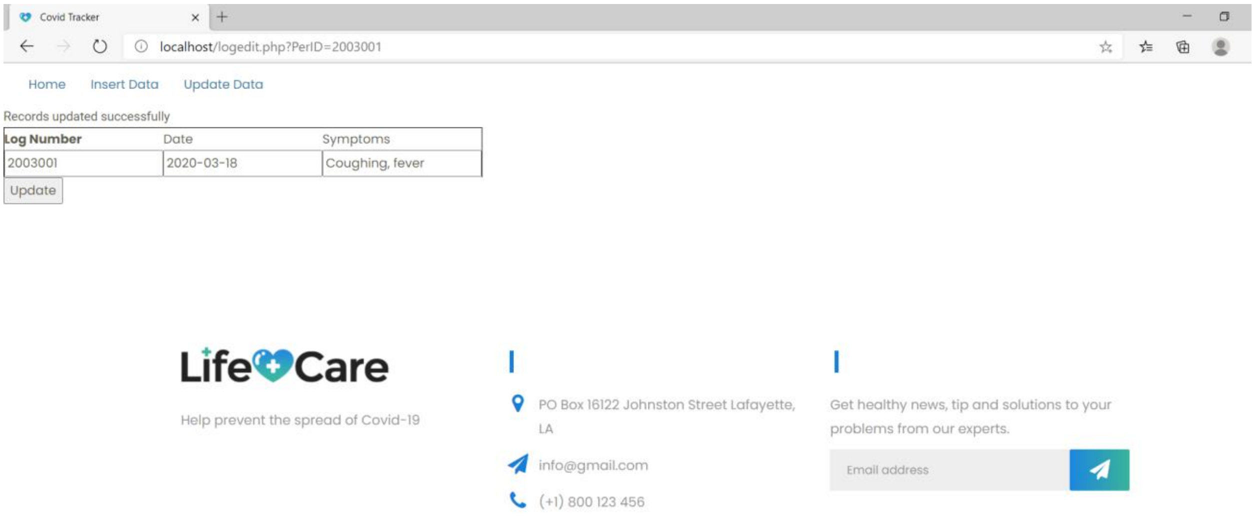
From home page, when first clicking “Update Data” and then “Log”, it moves to the Log table section, with a “Update” button on each row.



Log number	Date	Symptoms	
2003001	2020-03-17	Coughing, difficulty breathing	Update
2003002	2020-03-24	Coughing, difficulty breathing	Update
2003003	2020-03-21	Fever, nausea	Update
2003004	2020-03-22	Chills, coughing	Update
2003005	2020-03-22	Shortness of breath	Update
2003006	2020-03-22	Fever, nausea	Update
2003007	2020-03-23	Muscle aches, nausea	Update
2003008	2020-03-22	Fever, nausea	Update
2003009	2020-03-25	Coughing, congestion	Update
2003010	2020-03-23	Fatigue, body aches	Update

Figure 13. Update Data Page.

When we click “Update” button on the first record, it redirects us to the edit page. We updated the date and symptoms attributes for Log Number=2003001.



Records updated successfully

Log Number	Date	Symptoms
2003001	2020-03-18	Coughing, fever

LifeCare

Help prevent the spread of Covid-19

PO Box 16122 Johnston Street Lafayette, LA

info@gmail.com

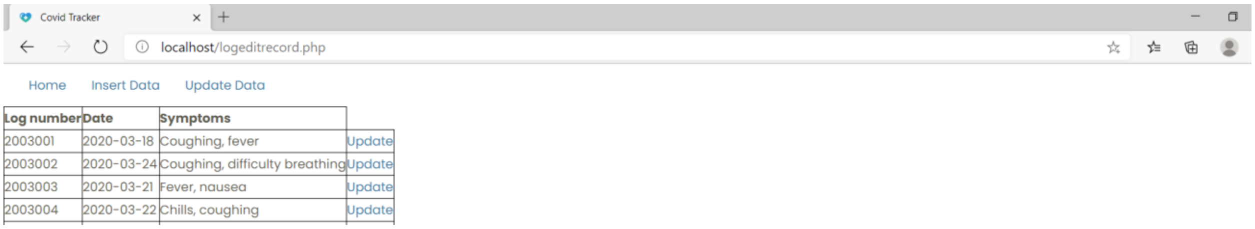
(+1) 800 123 456

Get healthy news, tip and solutions to your problems from our experts.

Email address

Figure 14. Update Data.

When we open the Log table again, we see that the first record is updated.



Log number	Date	Symptoms	
2003001	2020-03-18	Coughing, fever	Update
2003002	2020-03-24	Coughing, difficulty breathing	Update
2003003	2020-03-21	Fever, nausea	Update
2003004	2020-03-22	Chills, coughing	Update

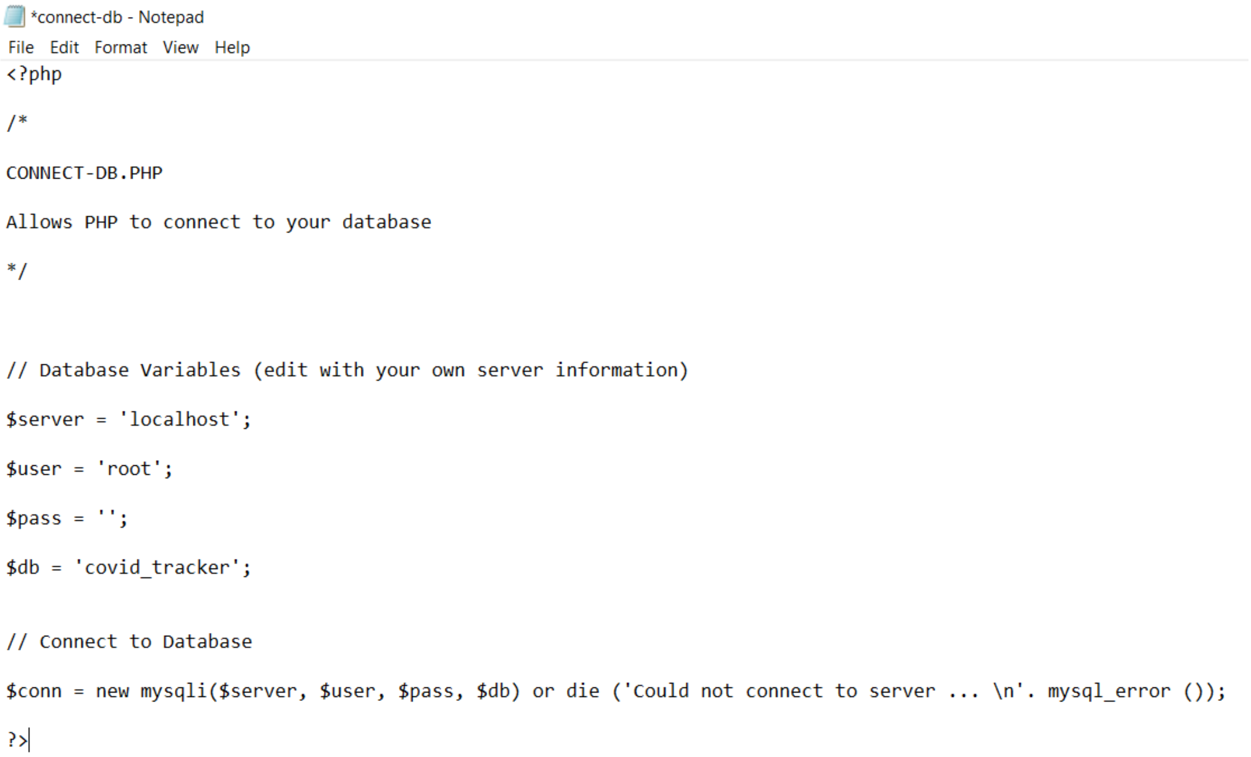
Figure 15. Record Updated.

Code snippets

In this section, we will include snippets of several php codes that were written to connect to and manipulate the database.

1. Connecting to the database

This php code were written to connect to the database and incorporated in the other php codes using php’s “include” function.

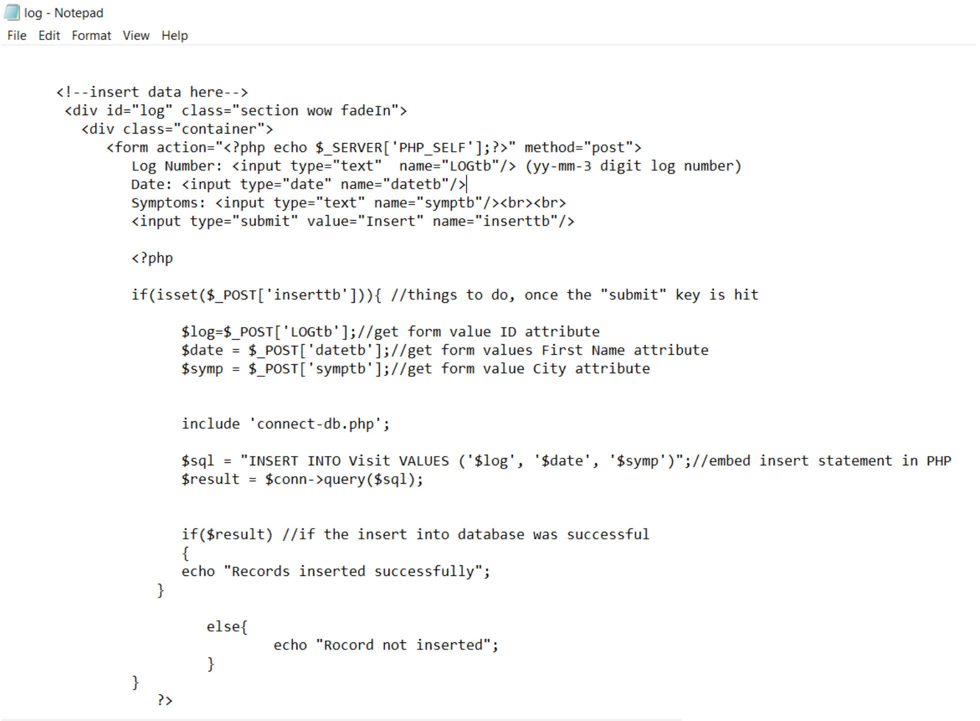


```
<?php

/*

CONNECT-DB.PHP

Allows PHP to connect to your database

*/


// Database Variables (edit with your own server information)

$server = 'localhost';

$user = 'root';

$pass = '';

$db = 'covid_tracker';

// Connect to Database

$conn = new mysqli($server, $user, $pass, $db) or die ('Could not connect to server ... \n'. mysql_error ());

?>
```

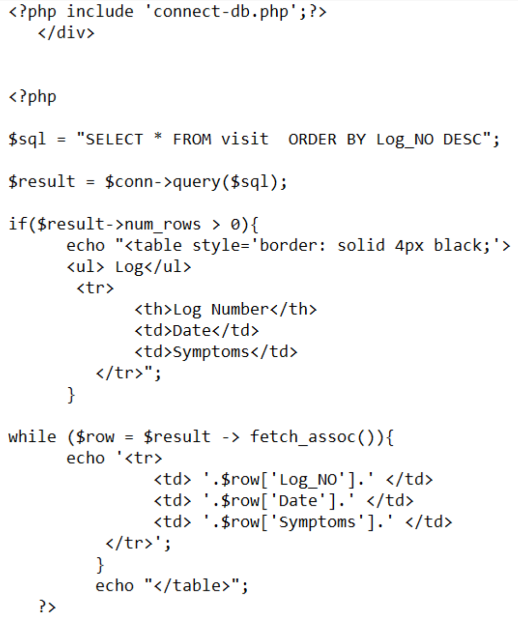
Figure 16. Connecting to Database Code Snippet.

2. Insert new records.

This code shows how the data is inserted and it includes codes to display the table after insertion.



```
      <!--insert data here-->
       <div id="log" class="section wow fadeIn">
         <div class="container">
            <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
               Log Number: <input type="text"  name="LOGtb"/> (yy-mm-3 digit log number)
               Date: <input type="date" name="datetb"/>
               Symptoms: <input type="text" name="symptb"/><br><br>
               <input type="submit" value="Insert" name="inserttb"/>

               <?php

               if(isset($_POST['inserttb'])){ //things to do, once the "submit" key is hit

                     $log=$_POST['LOGtb'];//get form value ID attribute
                     $date = $_POST['datetb'];//get form values First Name attribute
                     $symp = $_POST['symptb'];//get form value City attribute


                     include 'connect-db.php';

                     $sql = "INSERT INTO Visit VALUES ('$log', '$date', '$symp')";//embed insert statement in PHP
                     $result = $conn->query($sql);


                     if($result) //if the insert into database was successful
                     {
                     echo "Records inserted successfully";
                  }

                        else{
                               echo "Rocord not inserted";
                        }
               }
                  ?>
```

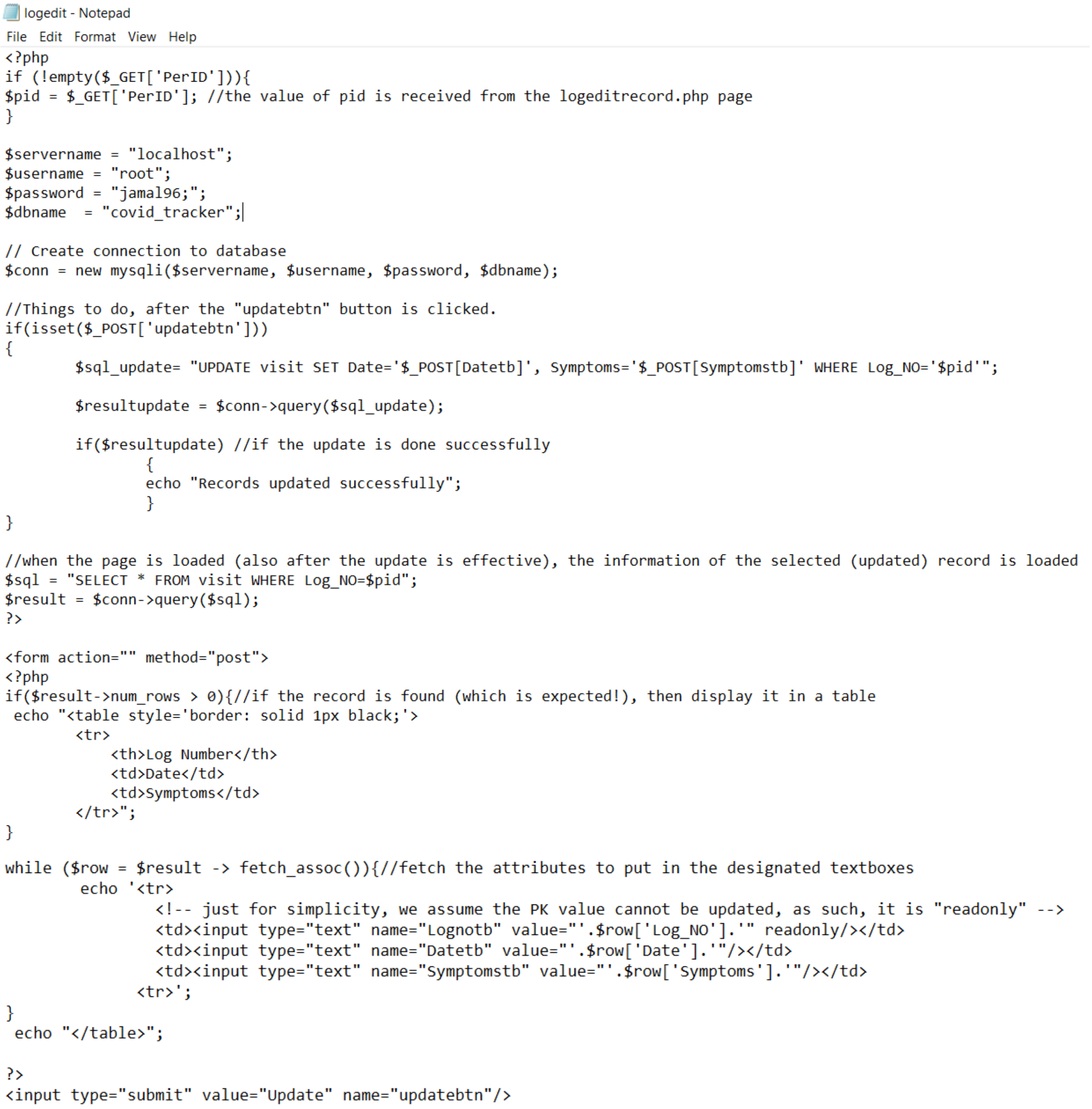


```
   <?php include 'connect-db.php';?>
      </div>


   <?php

   $sql = "SELECT * FROM visit  ORDER BY Log_NO DESC";

   $result = $conn->query($sql);

   if($result->num_rows > 0){
         echo "<table style='border: solid 4px black;'>
         <ul> Log</ul>
          <tr>
                <th>Log Number</th>
                <td>Date</td>
                <td>Symptoms</td>
            </tr>";
         }

   while ($row = $result -> fetch_assoc()){
         echo '<tr>
                  <td> '.$row['Log_NO'].' </td>
                  <td> '.$row['Date'].' </td>
                  <td> '.$row['Symptoms'].' </td>
             </tr>';
            }
            echo "</table>";
      ?>
```

Figure 17. Insert Record Code Snippet.

3. Update the records.

This code shows how the form to update a record.

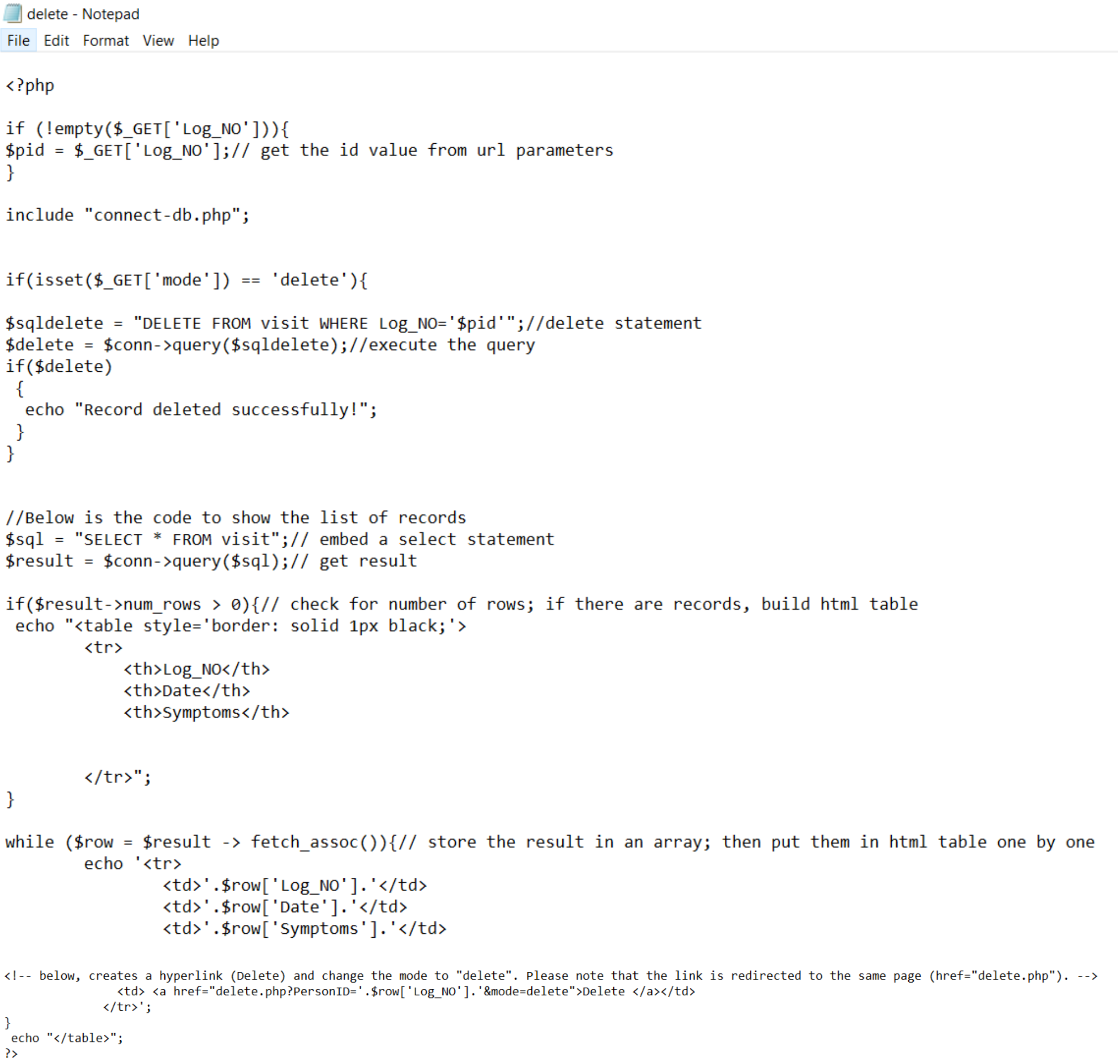


```
<?php
if (!empty($_GET['PerID'])){
$pid = $_GET['PerID']; //the value of pid is received from the logeditrecord.php page
}

$servername = "localhost";
$username = "root";
$password = "jamal96;";
$dbname  = "covid_tracker";

// Create connection to database
$conn = new mysqli($servername, $username, $password, $dbname);

//Things to do, after the "updatebtn" button is clicked.
if(isset($_POST['updatebtn']))
{
        $sql_update= "UPDATE visit SET Date='$_POST[Datetb]', Symptoms='$_POST[Symptomstb]' WHERE Log_NO='$pid'";

        $resultupdate = $conn->query($sql_update);

        if($resultupdate) //if the update is done successfully
                {
                echo "Records updated successfully";
                }
}

//when the page is loaded (also after the update is effective), the information of the selected (updated) record is loaded
$sql = "SELECT * FROM visit WHERE Log_NO=$pid";
$result = $conn->query($sql);
?>

<form action="" method="post">
<?php
if($result->num_rows > 0){//if the record is found (which is expected!), then display it in a table
 echo "<table style='border: solid 1px black;'>
        <tr>
            <th>Log Number</th>
            <td>Date</td>
            <td>Symptoms</td>
        </tr>";
}

while ($row = $result -> fetch_assoc()){//fetch the attributes to put in the designated textboxes
        echo '<tr>
                <!-- just for simplicity, we assume the PK value cannot be updated, as such, it is "readonly" -->
                <td><input type="text" name="Lognotb" value="'.$row['Log_NO'].'" readonly/></td>
                <td><input type="text" name="Datetb" value="'.$row['Date'].'"/></td>
                <td><input type="text" name="Symptomstb" value="'.$row['Symptoms'].'"/></td>
              <tr>';
}
 echo "</table>";

?>
<input type="submit" value="Update" name="updatebtn"/>
```

Figure 18. Update Data Code Snippet.

4. Delete the records.

This code shows how the record is deleted.



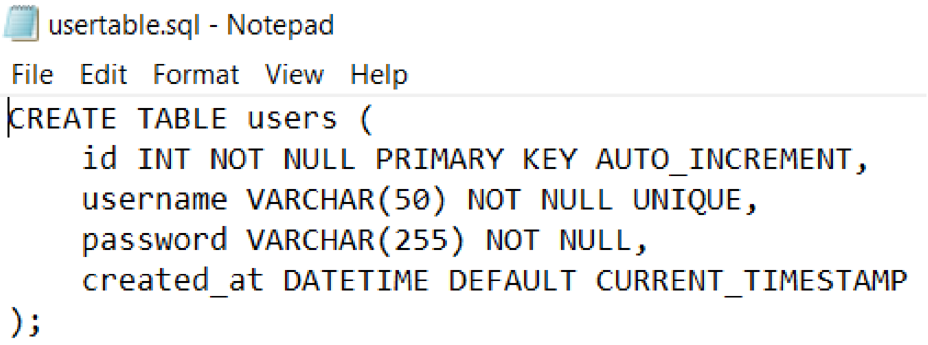
```
<?php

if (!empty($_GET['Log_NO'])){
$pid = $_GET['Log_NO'];// get the id value from url parameters
}

include "connect-db.php";


if(isset($_GET['mode']) == 'delete'){

$sqldelete = "DELETE FROM visit WHERE Log_NO='$pid'";//delete statement
$delete = $conn->query($sqldelete);//execute the query
if($delete)
 {
  echo "Record deleted successfully!";
 }
}


//Below is the code to show the list of records
$sql = "SELECT * FROM visit";// embed a select statement
$result = $conn->query($sql);// get result

if($result->num_rows > 0){// check for number of rows; if there are records, build html table
 echo "<table style='border: solid 1px black;'>
        <tr>
            <th>Log_NO</th>
            <th>Date</th>
            <th>Symptoms</th>


        </tr>";
}

while ($row = $result -> fetch_assoc()){// store the result in an array; then put them in html table one by one
        echo '<tr>
                <td>'.$row['Log_NO'].'</td>
                <td>'.$row['Date'].'</td>
                <td>'.$row['Symptoms'].'</td>

<!-- below, creates a hyperlink (Delete) and change the mode to "delete". Please note that the link is redirected to the same page (href="delete.php"). -->
                <td> <a href="delete.php?PersonID='.$row['Log_NO'].'&mode=delete">Delete </a></td>
               </tr>';
}
 echo "</table>";
?>
```

Figure 19. Delete Record Code Snippet.

Security features

Database Management System should include a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access. For our application, we implemented "Access control"; a security mechanism for restricting access to a system's objects (the database) as a whole. Access control is defined as creating user accounts and passwords to control the login process.

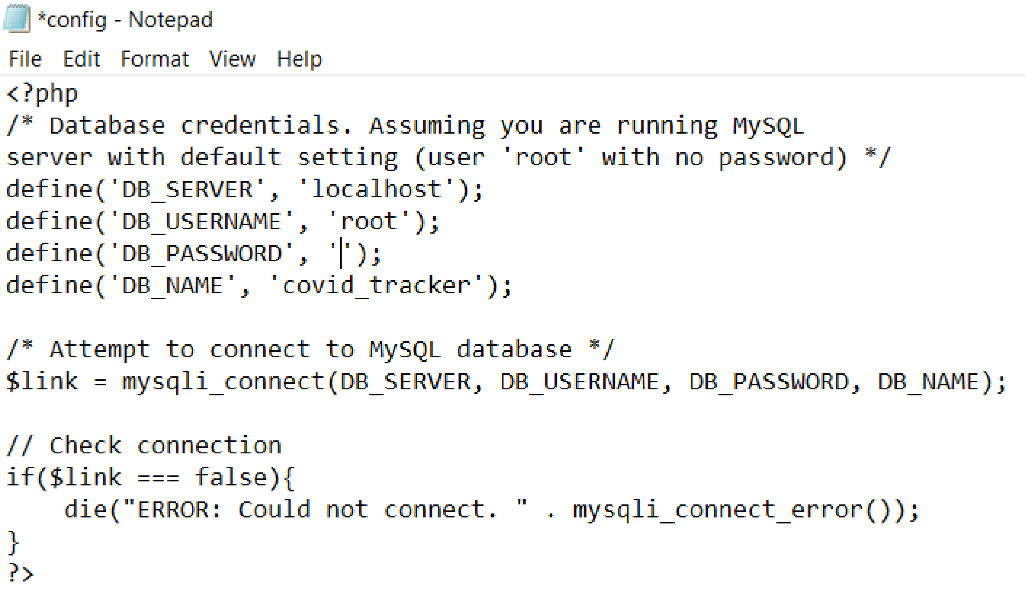
First, we create a table that will hold all the user data.



```
CREATE TABLE users (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Figure 20. Code Snippet for Creating a User.

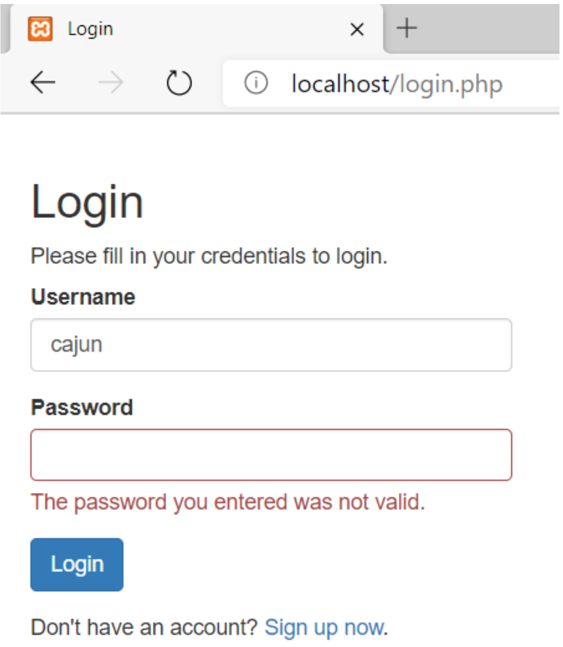
After creating the table, we create a PHP script in order to connect to the MySQL database server.



```
<?php
/* Database credentials. Assuming you are running MySQL
server with default setting (user 'root' with no password) */
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'covid_tracker');

/* Attempt to connect to MySQL database */
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```

Figure 21. Code Snippet to Connect to the Server.

Next, we build a registration system that allows users to create a new account by filling out a web form (Figure 2a). The system also generates errors if a user tries to submit the form without entering any value, or if username entered by the user is already taken by another user. After that, we create a login form where user can enter their username and password (Figure 2b). When user submit the form these inputs will be verified against the credentials stored in the database, if the username and password match, the user is authorized and granted access to the site, otherwise the login attempt will be rejected.



Login

Please fill in your credentials to login.

Username: cajun

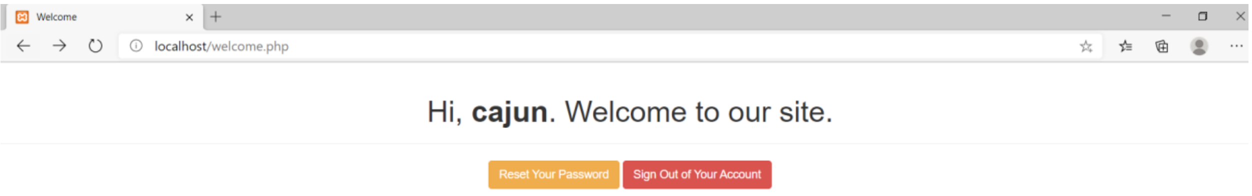
Password

The password you entered was not valid.

Don't have an account? Sign up now.

Figure 22. Login Error Message.

In addition, if you run "login.php" on the server while you are logged in you will be redirected to the welcome page. Here, you have options such as signing out of your account or resetting your password.



Hi, **cajun**. Welcome to our site.

Figure 23. Welcome Message after User Login.