CSI 2132 - Databases Course Project Deliverable 2

Uottawa
Verena Kantere
March 31st 2023

James Couture - 300076065
Pierre Allard - 3000102131
Robert Djukic - 300197788

1. A report that includes the following:

a. The DBMS and the programming languages that you have used in your implementation of the application.

DBMS: pgAdmin 4,
Programming Languages: postgreSQL, PHP, html, css, javascript

The majority of the project was written in postgreSQL, php, and html. Css and javascript were rarely implemented for small parts of the site. The web app is all one file, and has the .php extension. The DBMS used postgresql v15 using pgadmin to set up the database.

b. Specific steps to guide someone to install your applications

Download and install PHP on your pc from https://www.php.net/downloads.
(x64 nts version for windows.)
Add php to your PATH in order to be able to call it from CMD.
php.ini-production should be renamed to php.ini
Uncomment the line:
extension=pgsql
in the newly renamed php.ini

Open and create a database in pgAdmin 4.
Run provided .sql in Query Tool, refresh.

Copy the php code provided in this document and paste it into a new php file named “HotelsSite.php”.
Open HotelsSite.php in a text editor.
Go to around line 53 where “$cn = pg_connect(…”
Ensure that all variables in the pg_connect function match what appears when you right click the server and go to the connection tab in pgAdmin (making sure the dbname, username and password are correct). Also ensure that the dbname is identical to yours in pgAdmin.

Open CMD Prompt.
Enter the line: “CD ” followed by the path to where you have HotelsSite.php saved. (Make sure HotelsSite.php is on the same drive as your PHP install.)

Run the following line:
“php -S localhost:8000 HotelsSite.php”

Open your browser and type localhost:8000 into the address bar.

You should now see the web app and be able to use its functionality.

c. A list with the DDLs that create your database

The DDL that creates the database is found in the SQL file but here is the list of all the names of DDLs

-DROP VIEW IF EXISTS available_rooms CASCADE;
-DROP VIEW IF EXISTS hotel_capacity CASCADE;
-DROP TABLE IF EXISTS manager CASCADE;
-DROP TABLE IF EXISTS rents CASCADE;
-DROP TABLE IF EXISTS works_for CASCADE;
-DROP TABLE IF EXISTS employee CASCADE;
-DROP TABLE IF EXISTS customer CASCADE;
-DROP TABLE IF EXISTS archived_room CASCADE;
-DROP TABLE IF EXISTS hotel CASCADE;
-DROP TABLE IF EXISTS room CASCADE;
-DROP TABLE IF EXISTS phone_numbers_chain CASCADE;
-DROP TABLE IF EXISTS phone_numbers_hotel CASCADE;
-DROP TABLE IF EXISTS hotel_chain;
-CREATE TABLE hotel_chain
-CREATE TABLE hotel
-CREATE TABLE phone_numbers_chain
-CREATE TABLE phone_numbers_hotel
-CREATE TABLE room
-CREATE TABLE archived_room
-CREATE TABLE customer
-CREATE TABLE employee
-CREATE TABLE works_for
-CREATE TABLE rents
-CREATE TABLE manager
-DROP VIEW IF EXISTS available_room CASCADE;
-DROP VIEW IF EXISTS hotel_capacity CASCADE;
-CREATE VIEW available_room
-CREATE VIEW hotel_room_capacity
-DROP TRIGGER IF EXISTS default_time_archive ON archived_room;
-DROP FUNCTION IF EXISTS set_current_time();
-CREATE FUNCTION set_current_time() RETURNS trigger as $$
-CREATE TRIGGER default_time_archive before insert on archived_room FOR EACH ROW
EXECUTE FUNCTION set_current_time();
-DROP TRIGGER IF EXISTS increment_hotel_chain ON hotel;

```
-DROP FUNCTION IF EXISTS  increment_hotel_chain();
-CREATE FUNCTION increment_hotel_chain()
-CREATE TRIGGER increment_hotel_chain after insert on hotel FOR EACH ROW
EXECUTE FUNCTION increment_hotel_chain();
-DROP TRIGGER IF EXISTS decrement_hotel_chain ON hotel;
-DROP FUNCTION IF EXISTS  decrement_hotel_chain();
-CREATE FUNCTION decrement_hotel_chain() RETURNS trigger as $$
-CREATE TRIGGER decrement_hotel_chain after delete on hotel FOR EACH ROW
EXECUTE FUNCTION decrement_hotel_chain();
-DROP TRIGGER IF EXISTS set_num_hotel_zero ON hotel_chain;
-DROP FUNCTION IF EXISTS set_num_hotel_zero();
-CREATE FUNCTION set_num_hotel_zero() RETURNS trigger as $$
-CREATE TRIGGER set_num_hotel_zero before insert on hotel_chain FOR EACH ROW
EXECUTE FUNCTION set_num_hotel_zero();
-DROP TRIGGER IF EXISTS one_manager ON employee;
-DROP FUNCTION IF EXISTS one_manager();
-CREATE FUNCTION one_manager() RETURNS trigger as $$
-CREATE TRIGGER one_manager before update on employee FOR EACH ROW
EXECUTE FUNCTION one_manager();
-DROP TRIGGER IF EXISTS set_manager ON works_for;
-DROP FUNCTION IF EXISTS set_manager();
-CREATE FUNCTION set_manager() RETURNS trigger as $$
-CREATE TRIGGER set_manager after insert on works_for FOR EACH ROW
EXECUTE FUNCTION set_manager();
-DROP FUNCTION IF EXISTS add_works_for(hotelID int, SSN_SIN int);
-CREATE FUNCTION add_works_for(hotelID int, SSN_SIN int) RETURNS void
LANGUAGE plpgsql as
$$
-DROP FUNCTION IF EXISTS add_rents(hotelID int, SSN_SIN int, room_number int,
date_booked date);
-CREATE FUNCTION add_rents(hotelID int, SSN_SIN int, room_number int,
date_booked date) RETURNS void LANGUAGE plpgsql as
$$
-DROP FUNCTION IF EXISTS add_manager(hotelID int, manager_SIN int,
employee_SSN int);
-CREATE FUNCTION add_manager(chainID int, SSN_SIN int, employee_SSN int)
RETURNS void LANGUAGE plpgsql as
$$
-DROP INDEX IF EXISTS room_capacity_index;
-DROP INDEX IF EXISTS phone_number_info_chain;
-DROP INDEX IF EXISTS phone_number_info_hotel;
-CREATE INDEX room_capacity_index ON room(capacity);
-CREATE INDEX phone_number_info_chain ON
phone_numbers_chain(phone_numberID,phone_number);
```

```
    -CREATE INDEX phone_number_info_hotel ON
    phone_numbers_hotel(phone_numberID,phone_number);
    -DROP INDEX IF EXISTS employee_index;
    -CREATE INDEX employee_index ON employee(first_name,last_name) WHERE
    roles_positions='manager';
    -DROP INDEX IF EXISTS rents_uni;
    -CREATE INDEX rents_uni ON rents(SSN_SIN,hotelID,room_number,date_booked);
```

Section for question 6)

Queries:

1) UPDATE archived_room SET availability = 'rented' WHERE hotelid = $toswap[0] AND room_number = $toswap[1] AND date_booked = '$toswap[3]';
2) UPDATE archived_room SET availability = 'booked' WHERE hotelid = $toswap[0] AND room_number = $toswap[1] AND date_booked = '$toswap[3]'
3) INSERT INTO customer (ssn_sin, first_name, last_name, address) VALUES ($ssn_sin, '$first_name', '$last_name', '$address')
4) DELETE FROM customer WHERE ssn_sin = $ssn_sin

($ variables are php value inserts)
Triggers:
1)

```
--Set manager
DROP TRIGGER IF EXISTS set_manager ON works_for;
DROP FUNCTION IF EXISTS set_manager();

CREATE FUNCTION set_manager() RETURNS trigger as $$
        BEGIN
                INSERT INTO manager(hotelID, manager_SSN, employee_SSN) VALUES
(new.hotelID,

                                                (SELECT SSN_SIN

                                                FROM employee natural join works_for

                                                WHERE works_for.SSN_SIN=NEW.SSN_SIN),

                                                new.SSN_SIN);

                UPDATE manager
                SET manager_SSN = (SELECT SSN_SIN
                                                        FROM employee natural join works_for natural
join
```

```
                                                            (SELECT works_for.hotelID FROM employee
natural join works_for where SSN_SIN=employee.SSN_SIN
                                                                       GROUP BY works_for.hotelID) as x
                                                            WHERE employee.roles_positions = 'manager'
AND works_for.hotelID = new.hotelID)
                    WHERE NEW.hotelID = manager.hotelID;

                    RETURN NEW;
          END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER set_manager after insert on works_for FOR EACH ROW
EXECUTE FUNCTION set_manager();
```

This trigger fills in the manager table automatically. Through the assumption that there is only one manager, it auto completes the manager information for the employees working in a given hotelID.

2)

```
CREATE TRIGGER increment_hotel_chain after insert on hotel FOR EACH ROW
EXECUTE FUNCTION increment_hotel_chain();

--Decrement hotel_chain number on hotel deletion
DROP TRIGGER IF EXISTS decrement_hotel_chain ON hotel;
DROP FUNCTION IF EXISTS  decrement_hotel_chain();

CREATE FUNCTION decrement_hotel_chain() RETURNS trigger as $$
          BEGIN
                    UPDATE hotel_chain
                    SET number_of_hotels = number_of_hotels - 1
                    WHERE old.hotel_chainID = hotel_chain.hotel_chainID;
                    RETURN NEW;
          END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER decrement_hotel_chain after delete on hotel FOR EACH ROW
EXECUTE FUNCTION decrement_hotel_chain();
```

This trigger allows to increment the number of hotels within a hotel_chain. Using this trigger with the help of two other ones in the code (increment_hotel_chain, set_num_hotel_zero), the number of hotels is automatically managed by the DBMS. It does not matter what the initial number is added when creating the hotels as it is set to 0 by default. From there, each time a hotel is inserted, it increments and everytime a hotel is deleted, it will decrement.

Section for question 7)

Note: In PostgreSQL, primary keys and unique() combinations are all automatically indexed. Most queries are done on those elements and therefore, it is hard to get much more index with great effect as most of the important ones are already indexed by default.

1)

```
CREATE INDEX room_capacity_index ON room(capacity);
```

This index is used to help the hotel_capacity view. It was chosen in order to better the efficiency of the system. This is a commonly used query on the employee page. For this, there are no updates on the data, but it is a constant query. Having it load faster just allows me to make sure the app loads quickly on initiation.

2)

```
CREATE INDEX employee_index ON employee(first_name,last_name) WHERE
roles_positions='manager';
```

This is used to help a more expensive trigger. It is a trigger that checks on the employees' managers. The specific trigger is one_manager() which sets the manager for everyone that works at the same hotel to the right SSN_SIN. This is pretty expensive in a larger database. This database is relatively small, so it would be hard to observe much difference, but it is still there in hopes to help scalability for those data-updates that need to find the right information.

3)

```
CREATE INDEX rents_uni ON rents(SSN_SIN,hotelID,room_number,date_booked);
```

This index is used to help that large amount of calling on rents. The entirety of rents is sortable and should be different overall. This has the hopes to load in all the rooms faster in a larger dataset. Even this dataset is somehow too small to see much of a difference. Nonetheless, this query is used for any room related activities.

Section for question 9)
The views can be seen in the employee section on the web app.

Edits From First Deliverable:

- Removed Owns
- Added hotelid
- Rents connected to archived rooms instead of rooms
- Total participation from hotel to hotel chain

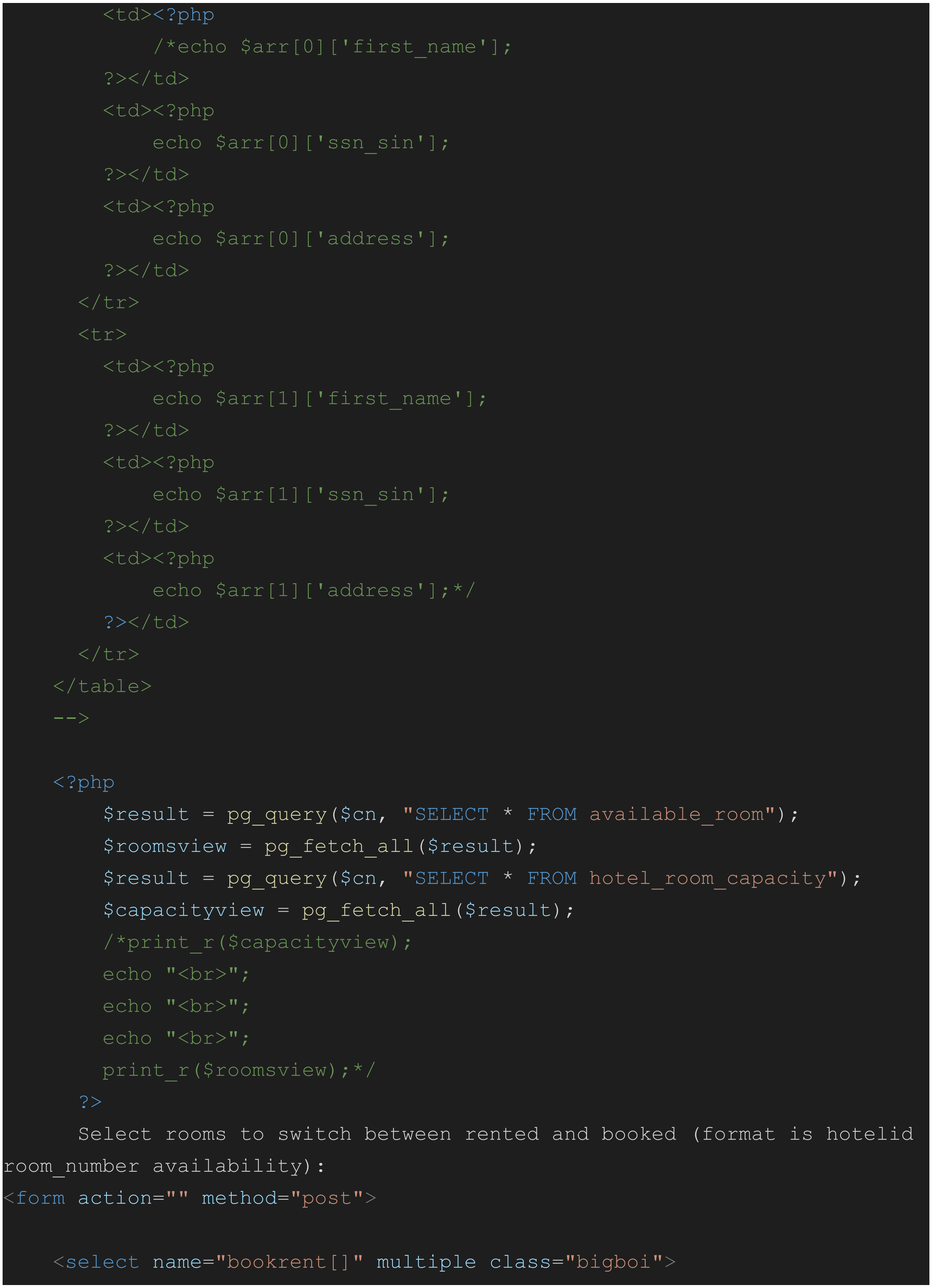
Assumptions:

- If a room is booked for multiple consecutive days with the same customer renting, these are considered to be the start and end dates chronologically.

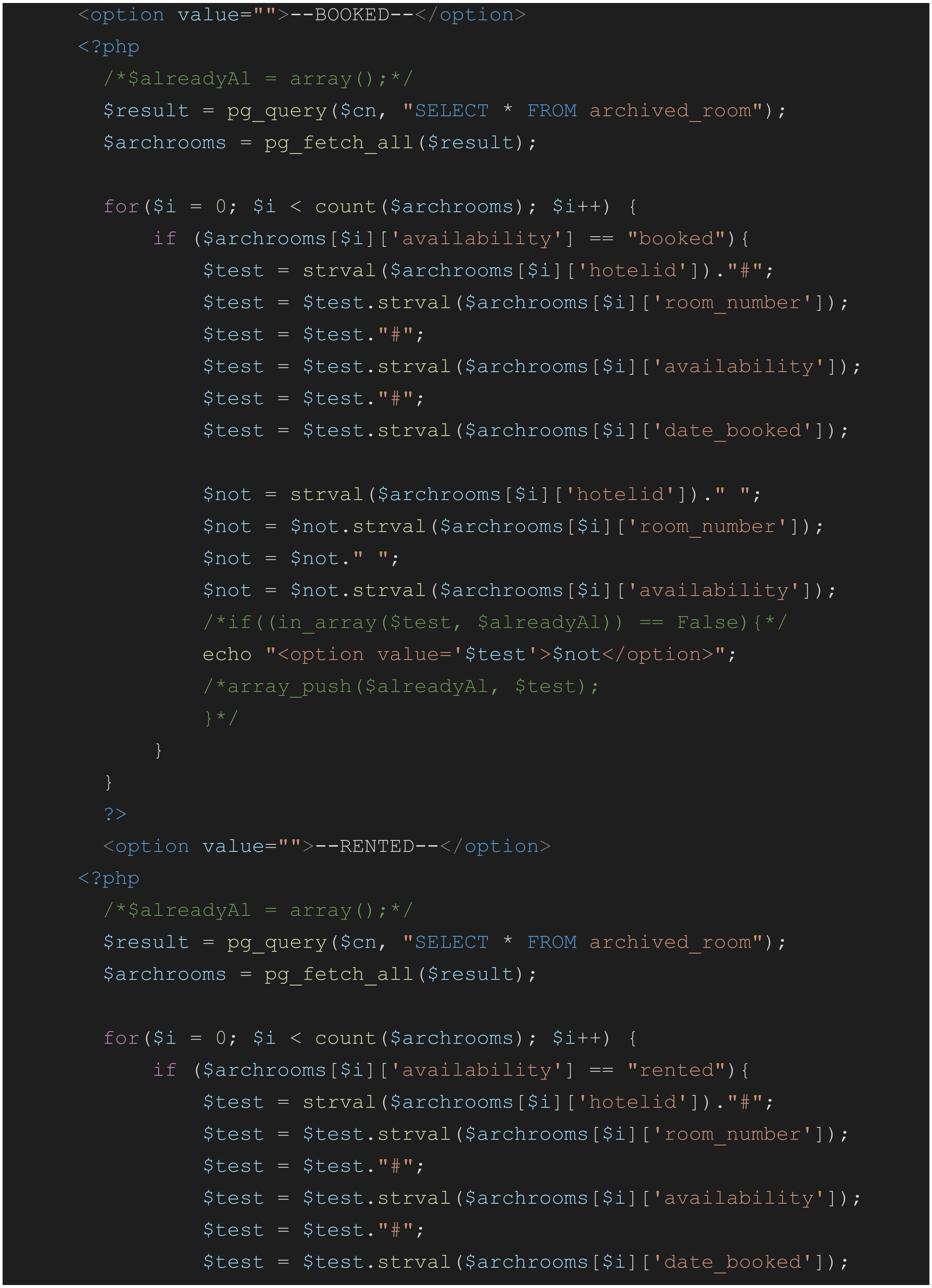
Appendix A:

Github link: https://github.com/Pannvvii/HotelDatabase

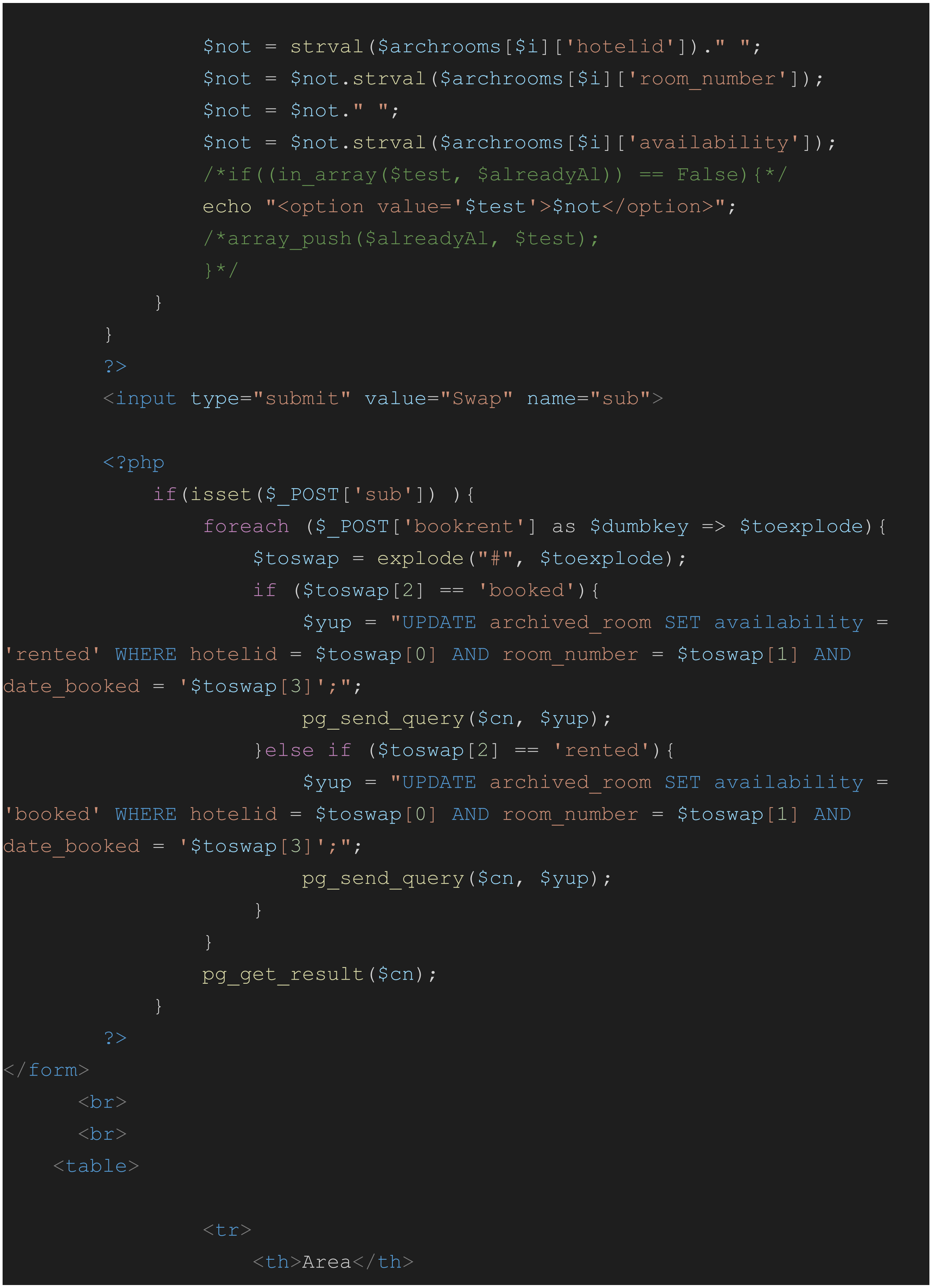
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Hotels</title>
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <meta name="description" content="" />
  <link rel="icon" href="favicon.png">
</head>
<body>
  <h1> Hotels</h1>
  <style>
  /* For best practice, move CSS below to an external CSS file. */
  .accordion {
    background-color: lightgray;
    color: #000000;
    cursor: pointer;
    padding: 18px;
    width: 100%;
    border: none;
    text-align: left;
    outline: none;
    font-size: 15px;
    transition: 0.4s; }
  .accordion-panel {
    padding: 0 18px;
    display: none;
    background-color: white;
    overflow: hidden; }
  .arrow-icon__line {
    width: 20px;
    height: 4px;
    background-color: #000000; }
  .accordion__indicator::after {
    content: "+"; }
  .accordion--open .accordion__indicator::after {
    content: "-"; }
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;}
  .bigboi {
        width: 150px;
        height: 300px;
        margin: 10px;
    }
  </style>
  <button class="accordion">
    Employee<span class="accordion__indicator"></span>
  </button>
  <div class="accordion-panel">
    <br>
      <?php
        $cn = pg_connect("host=localhost port=5432 dbname=v27
user=postgres password=ra2002");
        $result = pg_query($cn, "SELECT * FROM employee");
        $arr = pg_fetch_all($result);
      ?>
      <!--
      <table>
      <tr>
        <th>Name</th>
        <th>SSN</th>
        <th>Address</th>
      </tr>
      <tr>
```



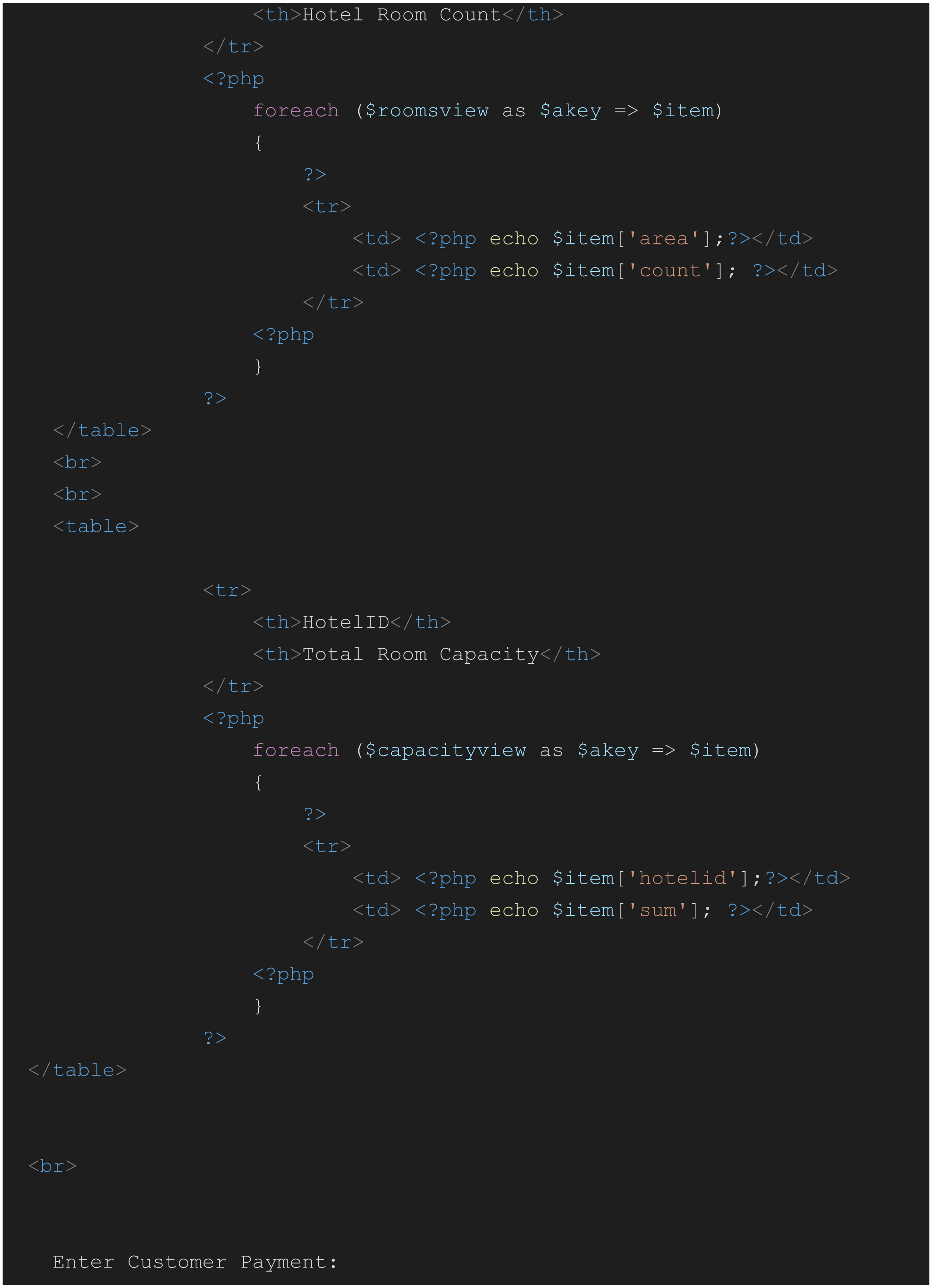
```
        <td><?php
            /*echo $arr[0]['first_name'];
        ?></td>
        <td><?php
            echo $arr[0]['ssn_sin'];
        ?></td>
        <td><?php
            echo $arr[0]['address'];
        ?></td>
      </tr>
      <tr>
        <td><?php
            echo $arr[1]['first_name'];
        ?></td>
        <td><?php
            echo $arr[1]['ssn_sin'];
        ?></td>
        <td><?php
            echo $arr[1]['address'];*/
        ?></td>
      </tr>
    </table>
    -->

    <?php
        $result = pg_query($cn, "SELECT * FROM available_room");
        $roomsview = pg_fetch_all($result);
        $result = pg_query($cn, "SELECT * FROM hotel_room_capacity");
        $capacityview = pg_fetch_all($result);
        /*print_r($capacityview);
        echo "<br>";
        echo "<br>";
        echo "<br>";
        print_r($roomsview);*/
      ?>
      Select rooms to switch between rented and booked (format is hotelid
room_number availability):
<form action="" method="post">

    <select name="bookrent[]" multiple class="bigboi">
```



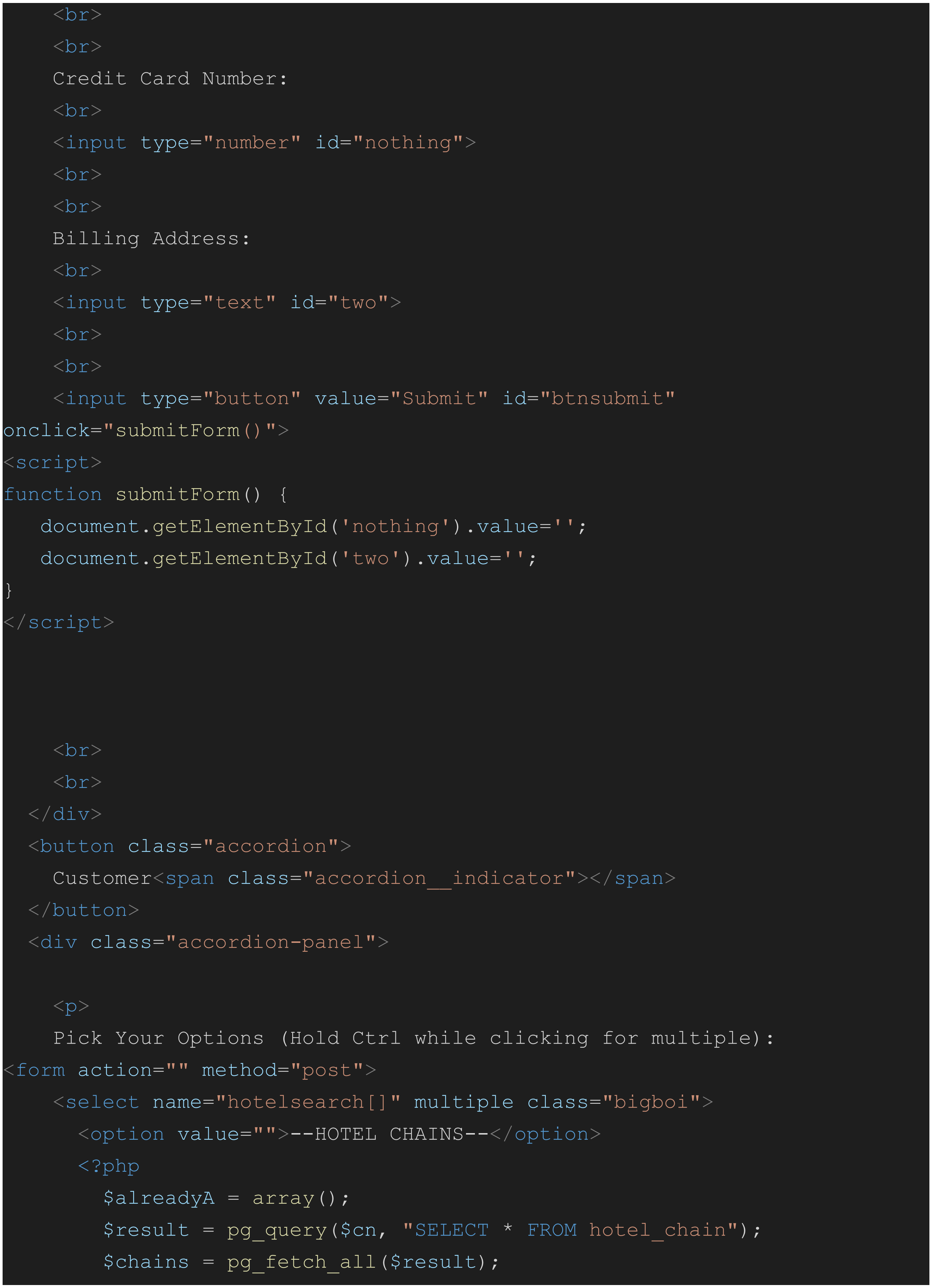
```
<option value="">--BOOKED--</option>
<?php
  /*$alreadyAl = array();*/
  $result = pg_query($cn, "SELECT * FROM archived_room");
  $archrooms = pg_fetch_all($result);

  for($i = 0; $i < count($archrooms); $i++) {
      if ($archrooms[$i]['availability'] == "booked"){
          $test = strval($archrooms[$i]['hotelid'])."#";
          $test = $test.strval($archrooms[$i]['room_number']);
          $test = $test."#";
          $test = $test.strval($archrooms[$i]['availability']);
          $test = $test."#";
          $test = $test.strval($archrooms[$i]['date_booked']);

          $not = strval($archrooms[$i]['hotelid'])." ";
          $not = $not.strval($archrooms[$i]['room_number']);
          $not = $not." ";
          $not = $not.strval($archrooms[$i]['availability']);
          /*if((in_array($test, $alreadyAl)) == False){*/
          echo "<option value='$test'>$not</option>";
          /*array_push($alreadyAl, $test);
          }*/
      }
  }
  ?>
  <option value="">--RENTED--</option>
<?php
  /*$alreadyAl = array();*/
  $result = pg_query($cn, "SELECT * FROM archived_room");
  $archrooms = pg_fetch_all($result);

  for($i = 0; $i < count($archrooms); $i++) {
      if ($archrooms[$i]['availability'] == "rented"){
          $test = strval($archrooms[$i]['hotelid'])."#";
          $test = $test.strval($archrooms[$i]['room_number']);
          $test = $test."#";
          $test = $test.strval($archrooms[$i]['availability']);
          $test = $test."#";
          $test = $test.strval($archrooms[$i]['date_booked']);
```



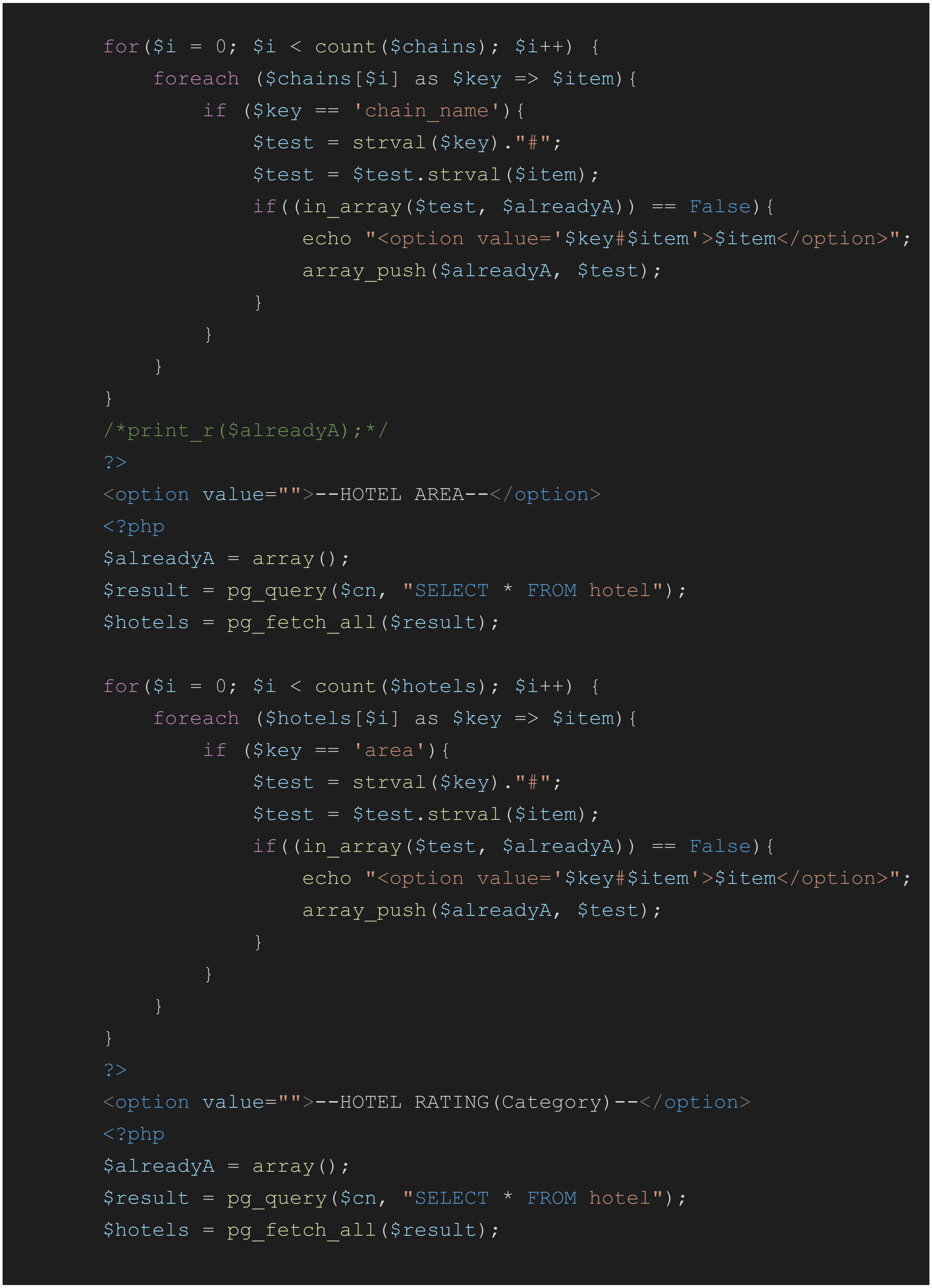
```
                $not = strval($archrooms[$i]['hotelid'])." ";
                $not = $not.strval($archrooms[$i]['room_number']);
                $not = $not." ";
                $not = $not.strval($archrooms[$i]['availability']);
                /*if((in_array($test, $alreadyAl)) == False){*/
                echo "<option value='$test'>$not</option>";
                /*array_push($alreadyAl, $test);
                }*/
            }
        }
        ?>
        <input type="submit" value="Swap" name="sub">

        <?php
            if(isset($_POST['sub']) ){
                foreach ($_POST['bookrent'] as $dumbkey => $toexplode){
                    $toswap = explode("#", $toexplode);
                    if ($toswap[2] == 'booked'){
                        $yup = "UPDATE archived_room SET availability =
'rented' WHERE hotelid = $toswap[0] AND room_number = $toswap[1] AND
date_booked = '$toswap[3]';";
                        pg_send_query($cn, $yup);
                    }else if ($toswap[2] == 'rented'){
                        $yup = "UPDATE archived_room SET availability =
'booked' WHERE hotelid = $toswap[0] AND room_number = $toswap[1] AND
date_booked = '$toswap[3]';";
                        pg_send_query($cn, $yup);
                    }
                }
                pg_get_result($cn);
            }
        ?>
</form>
      <br>
      <br>
    <table>

                <tr>
                    <th>Area</th>
```



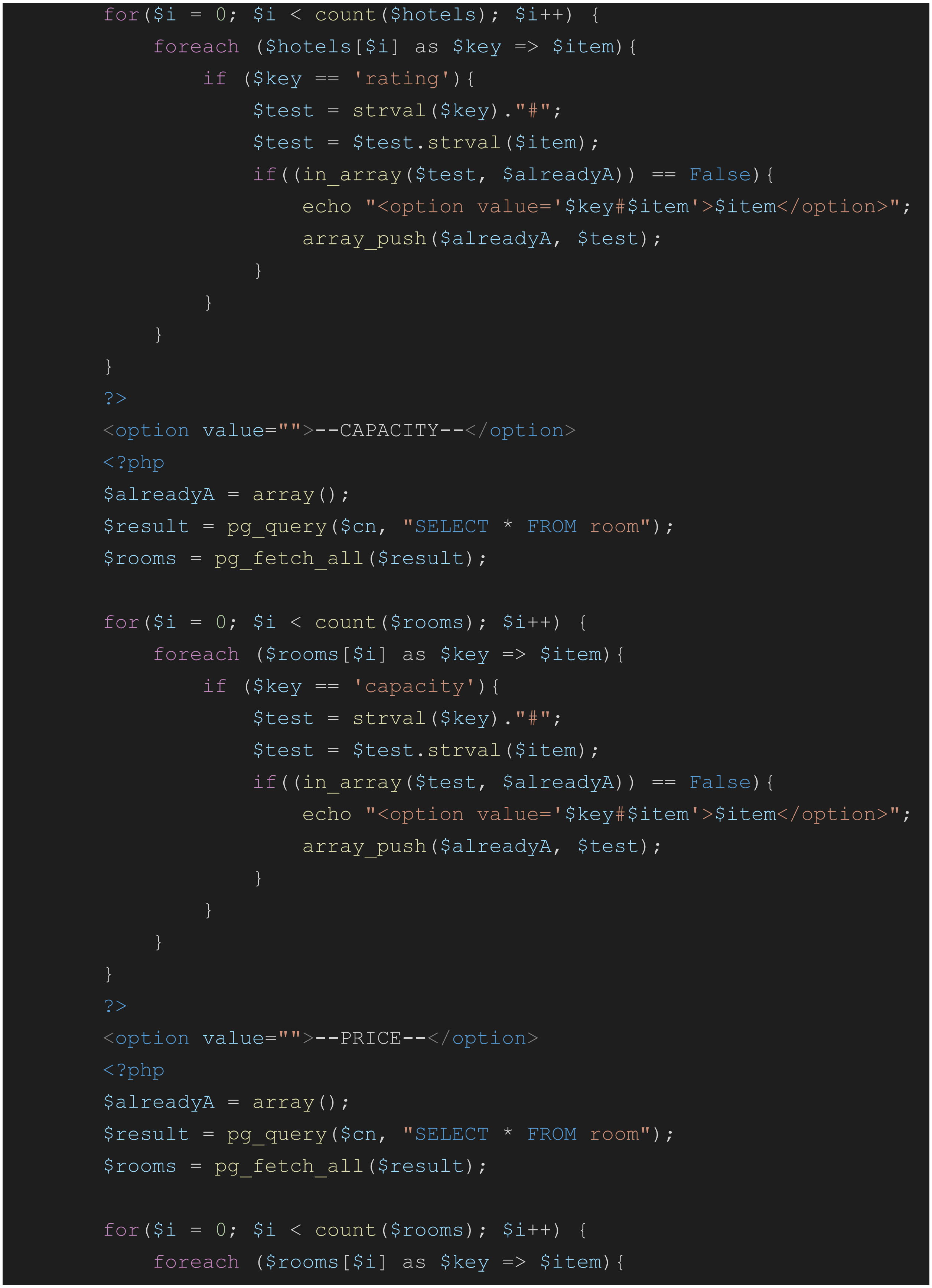
```
                  <th>Hotel Room Count</th>
              </tr>
              <?php
                  foreach ($roomsview as $akey => $item)
                  {
                      ?>
                      <tr>
                          <td> <?php echo $item['area'];?></td>
                          <td> <?php echo $item['count']; ?></td>
                      </tr>
                  <?php
                  }
              ?>
  </table>
  <br>
  <br>
  <table>

              <tr>
                  <th>HotelID</th>
                  <th>Total Room Capacity</th>
              </tr>
              <?php
                  foreach ($capacityview as $akey => $item)
                  {
                      ?>
                      <tr>
                          <td> <?php echo $item['hotelid'];?></td>
                          <td> <?php echo $item['sum']; ?></td>
                      </tr>
                  <?php
                  }
              ?>
</table>


<br>


  Enter Customer Payment:
```



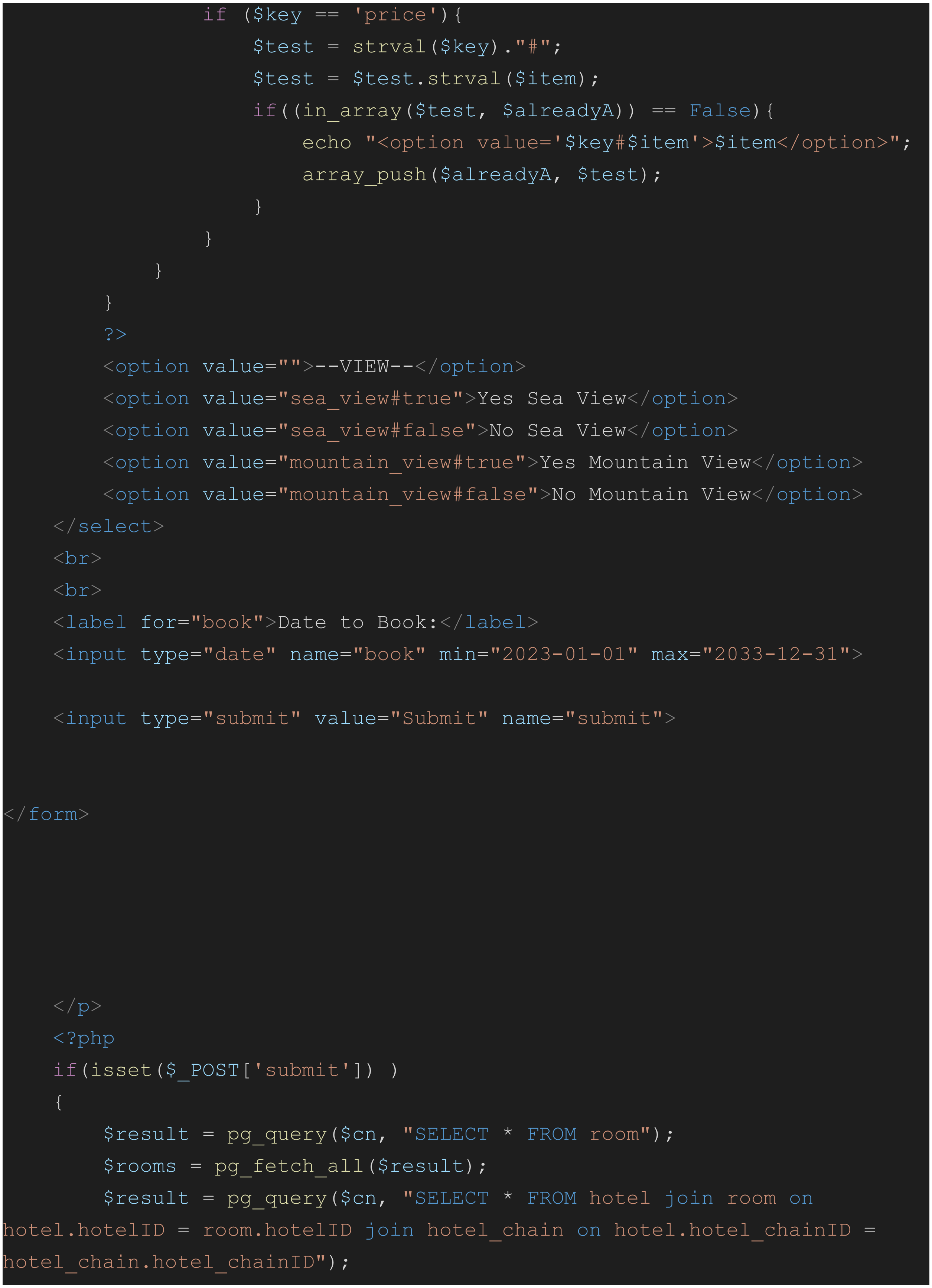
```
    <br>
    <br>
    Credit Card Number:
    <br>
    <input type="number" id="nothing">
    <br>
    <br>
    Billing Address:
    <br>
    <input type="text" id="two">
    <br>
    <br>
    <input type="button" value="Submit" id="btnsubmit"
onclick="submitForm()">
<script>
function submitForm() {
   document.getElementById('nothing').value='';
   document.getElementById('two').value='';
}
</script>



    <br>
    <br>
  </div>
  <button class="accordion">
    Customer<span class="accordion__indicator"></span>
  </button>
  <div class="accordion-panel">

    <p>
    Pick Your Options (Hold Ctrl while clicking for multiple):
<form action="" method="post">
    <select name="hotelsearch[]" multiple class="bigboi">
      <option value="">--HOTEL CHAINS--</option>
      <?php
        $alreadyA = array();
        $result = pg_query($cn, "SELECT * FROM hotel_chain");
        $chains = pg_fetch_all($result);
```



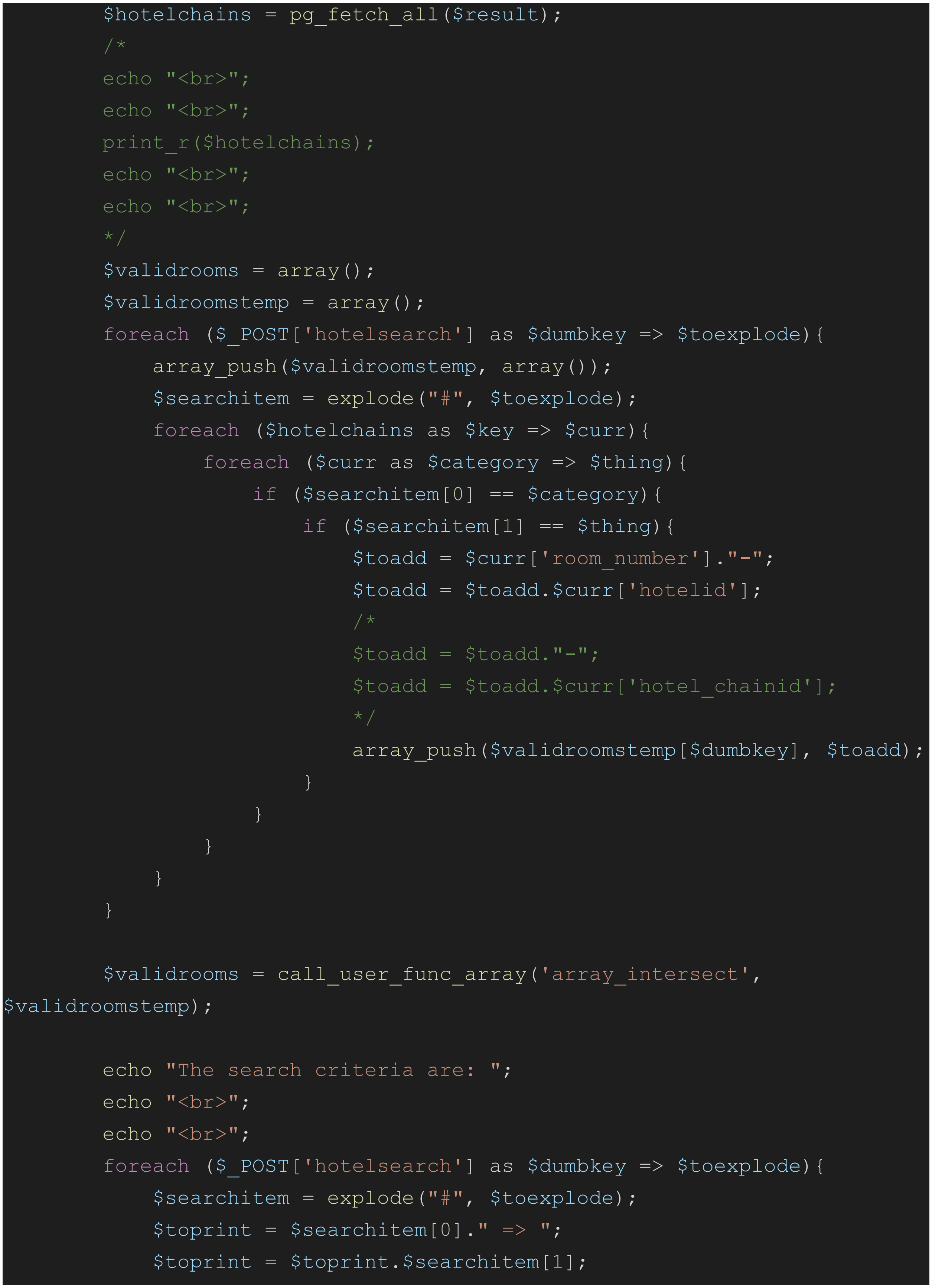
```php
for($i = 0; $i < count($chains); $i++) {
    foreach ($chains[$i] as $key => $item){
        if ($key == 'chain_name'){
            $test = strval($key)."#";
            $test = $test.strval($item);
            if((in_array($test, $alreadyA)) == False){
                echo "<option value='$key#$item'>$item</option>";
                array_push($alreadyA, $test);
            }
        }
    }
}
/*print_r($alreadyA);*/
?>
<option value="">--HOTEL AREA--</option>
<?php
$alreadyA = array();
$result = pg_query($cn, "SELECT * FROM hotel");
$hotels = pg_fetch_all($result);

for($i = 0; $i < count($hotels); $i++) {
    foreach ($hotels[$i] as $key => $item){
        if ($key == 'area'){
            $test = strval($key)."#";
            $test = $test.strval($item);
            if((in_array($test, $alreadyA)) == False){
                echo "<option value='$key#$item'>$item</option>";
                array_push($alreadyA, $test);
            }
        }
    }
}
?>
<option value="">--HOTEL RATING(Category)--</option>
<?php
$alreadyA = array();
$result = pg_query($cn, "SELECT * FROM hotel");
$hotels = pg_fetch_all($result);
```



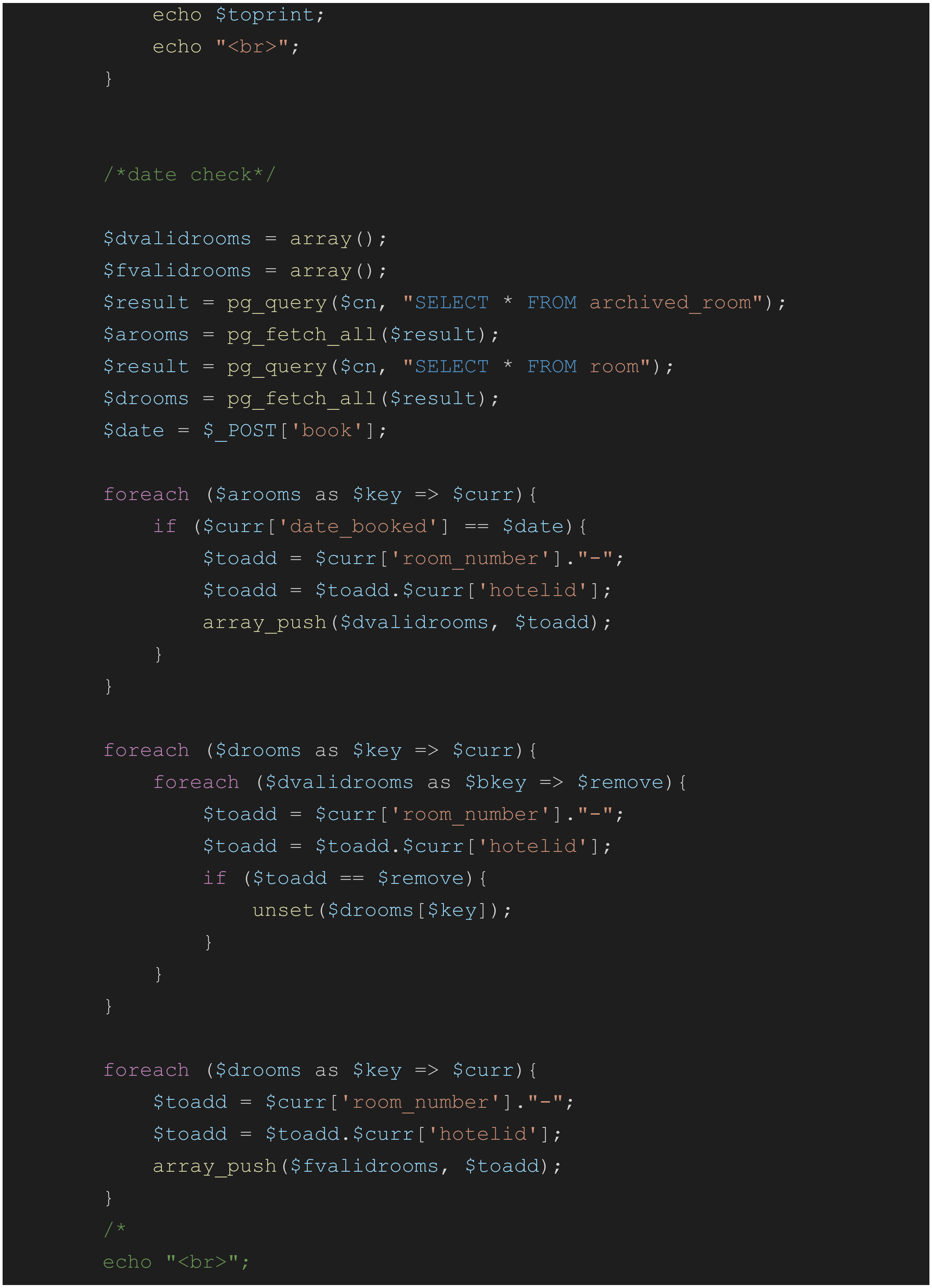
```
for($i = 0; $i < count($hotels); $i++) {
    foreach ($hotels[$i] as $key => $item){
        if ($key == 'rating'){
            $test = strval($key)."#";
            $test = $test.strval($item);
            if((in_array($test, $alreadyA)) == False){
                echo "<option value='$key#$item'>$item</option>";
                array_push($alreadyA, $test);
            }
        }
    }
}
?>
<option value="">--CAPACITY--</option>
<?php
$alreadyA = array();
$result = pg_query($cn, "SELECT * FROM room");
$rooms = pg_fetch_all($result);

for($i = 0; $i < count($rooms); $i++) {
    foreach ($rooms[$i] as $key => $item){
        if ($key == 'capacity'){
            $test = strval($key)."#";
            $test = $test.strval($item);
            if((in_array($test, $alreadyA)) == False){
                echo "<option value='$key#$item'>$item</option>";
                array_push($alreadyA, $test);
            }
        }
    }
}
?>
<option value="">--PRICE--</option>
<?php
$alreadyA = array();
$result = pg_query($cn, "SELECT * FROM room");
$rooms = pg_fetch_all($result);

for($i = 0; $i < count($rooms); $i++) {
    foreach ($rooms[$i] as $key => $item){
```



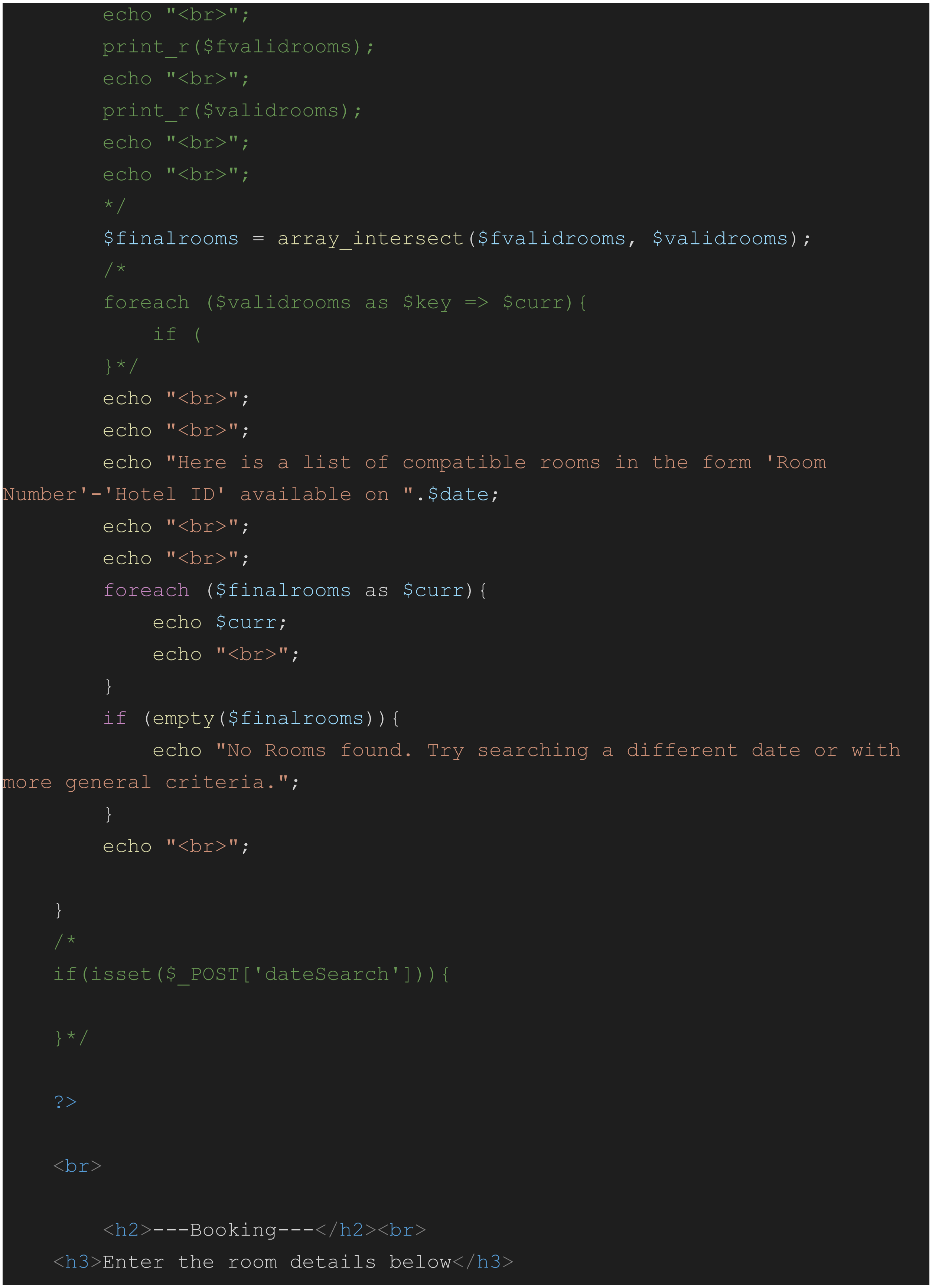
```
                if ($key == 'price'){
                    $test = strval($key)."#";
                    $test = $test.strval($item);
                    if((in_array($test, $alreadyA)) == False){
                        echo "<option value='$key#$item'>$item</option>";
                        array_push($alreadyA, $test);
                    }
                }
            }
        }
        ?>
        <option value="">--VIEW--</option>
        <option value="sea_view#true">Yes Sea View</option>
        <option value="sea_view#false">No Sea View</option>
        <option value="mountain_view#true">Yes Mountain View</option>
        <option value="mountain_view#false">No Mountain View</option>
    </select>
    <br>
    <br>
    <label for="book">Date to Book:</label>
    <input type="date" name="book" min="2023-01-01" max="2033-12-31">

    <input type="submit" value="Submit" name="submit">


</form>






    </p>
    <?php
    if(isset($_POST['submit']) )
    {
        $result = pg_query($cn, "SELECT * FROM room");
        $rooms = pg_fetch_all($result);
        $result = pg_query($cn, "SELECT * FROM hotel join room on
hotel.hotelID = room.hotelID join hotel_chain on hotel.hotel_chainID =
hotel_chain.hotel_chainID");
```



```php
        $hotelchains = pg_fetch_all($result);
        /*
        echo "<br>";
        echo "<br>";
        print_r($hotelchains);
        echo "<br>";
        echo "<br>";
        */
        $validrooms = array();
        $validroomstemp = array();
        foreach ($_POST['hotelsearch'] as $dumbkey => $toexplode){
            array_push($validroomstemp, array());
            $searchitem = explode("#", $toexplode);
            foreach ($hotelchains as $key => $curr){
                foreach ($curr as $category => $thing){
                    if ($searchitem[0] == $category){
                        if ($searchitem[1] == $thing){
                            $toadd = $curr['room_number']."-";
                            $toadd = $toadd.$curr['hotelid'];
                            /*
                            $toadd = $toadd."-";
                            $toadd = $toadd.$curr['hotel_chainid'];
                            */
                            array_push($validroomstemp[$dumbkey], $toadd);
                        }
                    }
                }
            }
        }

        $validrooms = call_user_func_array('array_intersect',
$validroomstemp);

        echo "The search criteria are: ";
        echo "<br>";
        echo "<br>";
        foreach ($_POST['hotelsearch'] as $dumbkey => $toexplode){
            $searchitem = explode("#", $toexplode);
            $toprint = $searchitem[0]." => ";
            $toprint = $toprint.$searchitem[1];
```

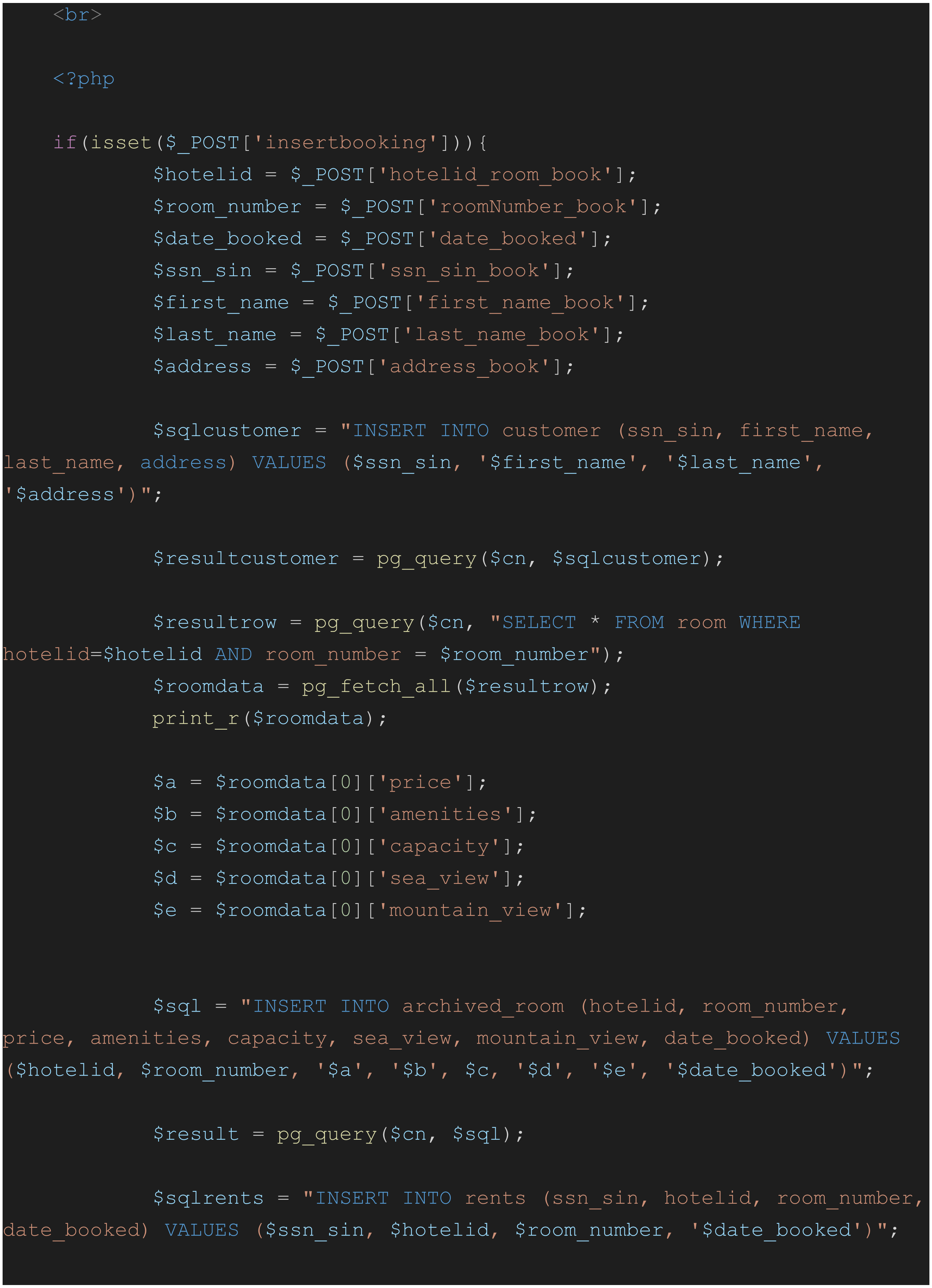


```
        echo $toprint;
        echo "<br>";
    }


    /*date check*/

    $dvalidrooms = array();
    $fvalidrooms = array();
    $result = pg_query($cn, "SELECT * FROM archived_room");
    $arooms = pg_fetch_all($result);
    $result = pg_query($cn, "SELECT * FROM room");
    $drooms = pg_fetch_all($result);
    $date = $_POST['book'];

    foreach ($arooms as $key => $curr){
        if ($curr['date_booked'] == $date){
            $toadd = $curr['room_number']."-";
            $toadd = $toadd.$curr['hotelid'];
            array_push($dvalidrooms, $toadd);
        }
    }

    foreach ($drooms as $key => $curr){
        foreach ($dvalidrooms as $bkey => $remove){
            $toadd = $curr['room_number']."-";
            $toadd = $toadd.$curr['hotelid'];
            if ($toadd == $remove){
                unset($drooms[$key]);
            }
        }
    }

    foreach ($drooms as $key => $curr){
        $toadd = $curr['room_number']."-";
        $toadd = $toadd.$curr['hotelid'];
        array_push($fvalidrooms, $toadd);
    }
    /*
    echo "<br>";
```

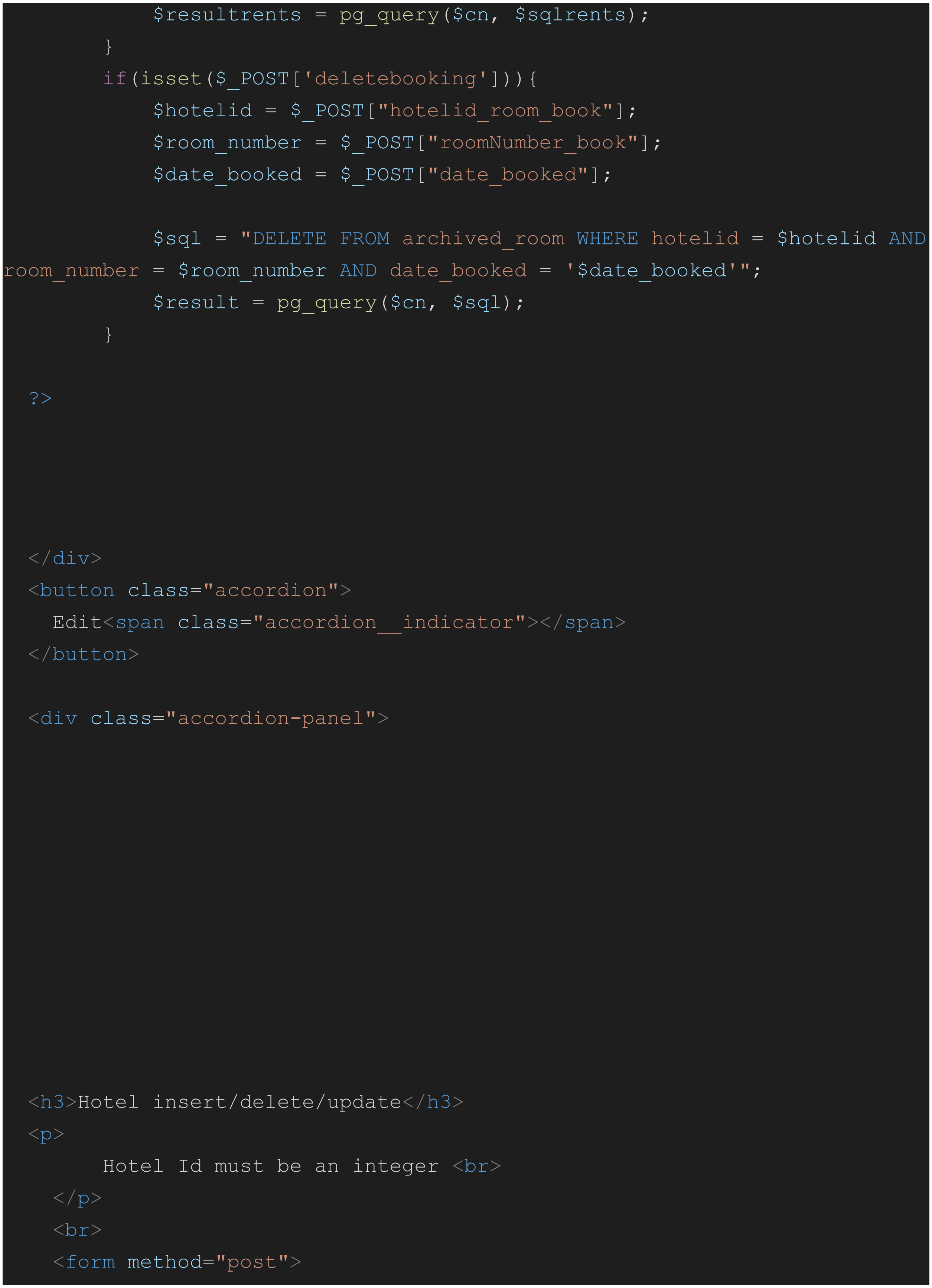


```
        echo "<br>";
        print_r($fvalidrooms);
        echo "<br>";
        print_r($validrooms);
        echo "<br>";
        echo "<br>";
        */
        $finalrooms = array_intersect($fvalidrooms, $validrooms);
        /*
        foreach ($validrooms as $key => $curr){
            if (
        }*/
        echo "<br>";
        echo "<br>";
        echo "Here is a list of compatible rooms in the form 'Room
Number'-'Hotel ID' available on ".$date;
        echo "<br>";
        echo "<br>";
        foreach ($finalrooms as $curr){
            echo $curr;
            echo "<br>";
        }
        if (empty($finalrooms)){
            echo "No Rooms found. Try searching a different date or with
more general criteria.";
        }
        echo "<br>";

    }
    /*
    if(isset($_POST['dateSearch'])){

    }*/

    ?>

    <br>

        <h2>---Booking---</h2><br>
    <h3>Enter the room details below</h3>
```

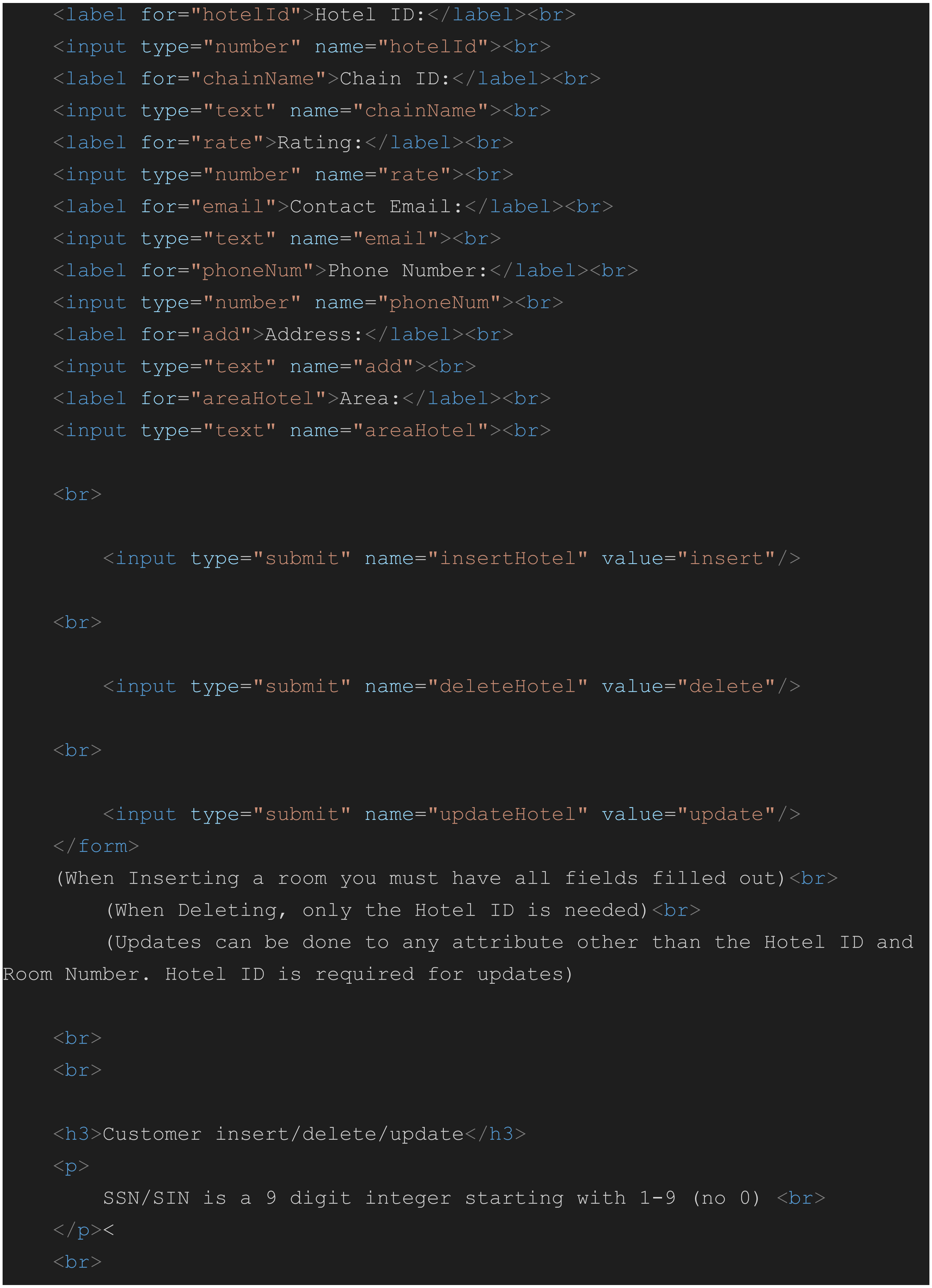
```html
    <p>
        Hotel Id must be an integer <br>
        Room Number must be an integer <br>
        Date Booked must be in the form "year-month-day" including the
dashes (eg: 2023-03-31)
    </p><br>

    <form method="post">
    <label for="hotelid_room_book">Hotel Id:</label><br>
    <input type="number" name="hotelid_room_book"><br>
    <label for="roomNumber_book">Room Number:</label><br>
    <input type="number" name="roomNumber_book"><br>
    <label for="date_booked">Date to Book:</label><br>
    <input type="text" name="date_booked"><br>

    <h3>Enter your personal information</h3><br>
    <p>
        SSN/SIN must be a 9 digit integer that starts with 1-9 (not 0)<br>
    </p><br>

    <label for="ssn_sin_book">SSN/SIN:</label><br>
    <input type="number" name="ssn_sin_book"><br>
    <label for="first_name_book">First Name:</label><br>
    <input type="text" name="first_name_book"><br>
    <label for="last_name_book">Last Name:</label><br>
    <input type="text" name="last_name_book"><br>
    <label for="address_book">Address:</label><br>
    <input type="text" name="address_book"><br>
    <br>

        <input type="submit" name="insertbooking" value="Book"/>

    <br>

        <input type="submit" name="deletebooking" value="Cancel Booking"/>
    </form>
    <p>
        To cancel you need hotel id, room number ,and date booked<br>
    </p><br>
```



```
    <br>

    <?php

    if(isset($_POST['insertbooking'])){
            $hotelid = $_POST['hotelid_room_book'];
            $room_number = $_POST['roomNumber_book'];
            $date_booked = $_POST['date_booked'];
            $ssn_sin = $_POST['ssn_sin_book'];
            $first_name = $_POST['first_name_book'];
            $last_name = $_POST['last_name_book'];
            $address = $_POST['address_book'];

            $sqlcustomer = "INSERT INTO customer (ssn_sin, first_name,
last_name, address) VALUES ($ssn_sin, '$first_name', '$last_name',
'$address')";

            $resultcustomer = pg_query($cn, $sqlcustomer);

            $resultrow = pg_query($cn, "SELECT * FROM room WHERE
hotelid=$hotelid AND room_number = $room_number");
            $roomdata = pg_fetch_all($resultrow);
            print_r($roomdata);

            $a = $roomdata[0]['price'];
            $b = $roomdata[0]['amenities'];
            $c = $roomdata[0]['capacity'];
            $d = $roomdata[0]['sea_view'];
            $e = $roomdata[0]['mountain_view'];


            $sql = "INSERT INTO archived_room (hotelid, room_number,
price, amenities, capacity, sea_view, mountain_view, date_booked) VALUES
($hotelid, $room_number, '$a', '$b', $c, '$d', '$e', '$date_booked')";

            $result = pg_query($cn, $sql);

            $sqlrents = "INSERT INTO rents (ssn_sin, hotelid, room_number,
date_booked) VALUES ($ssn_sin, $hotelid, $room_number, '$date_booked')";
```

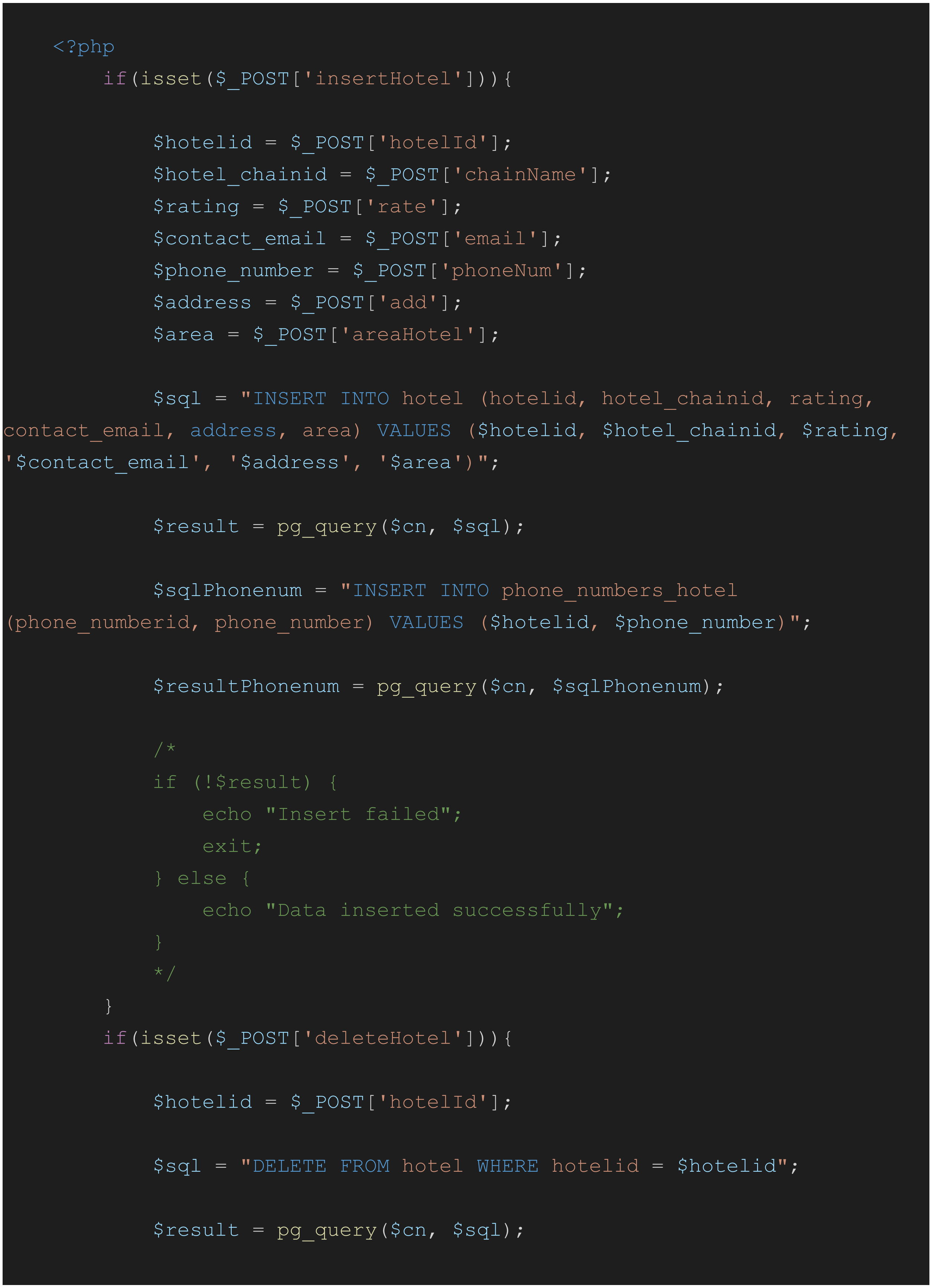


```
            $resultrents = pg_query($cn, $sqlrents);
        }
        if(isset($_POST['deletebooking'])){
            $hotelid = $_POST["hotelid_room_book"];
            $room_number = $_POST["roomNumber_book"];
            $date_booked = $_POST["date_booked"];

            $sql = "DELETE FROM archived_room WHERE hotelid = $hotelid AND
room_number = $room_number AND date_booked = '$date_booked'";
            $result = pg_query($cn, $sql);
        }

  ?>




  </div>
  <button class="accordion">
    Edit<span class="accordion__indicator"></span>
  </button>

  <div class="accordion-panel">











  <h3>Hotel insert/delete/update</h3>
  <p>
        Hotel Id must be an integer <br>
    </p>
    <br>
    <form method="post">
```

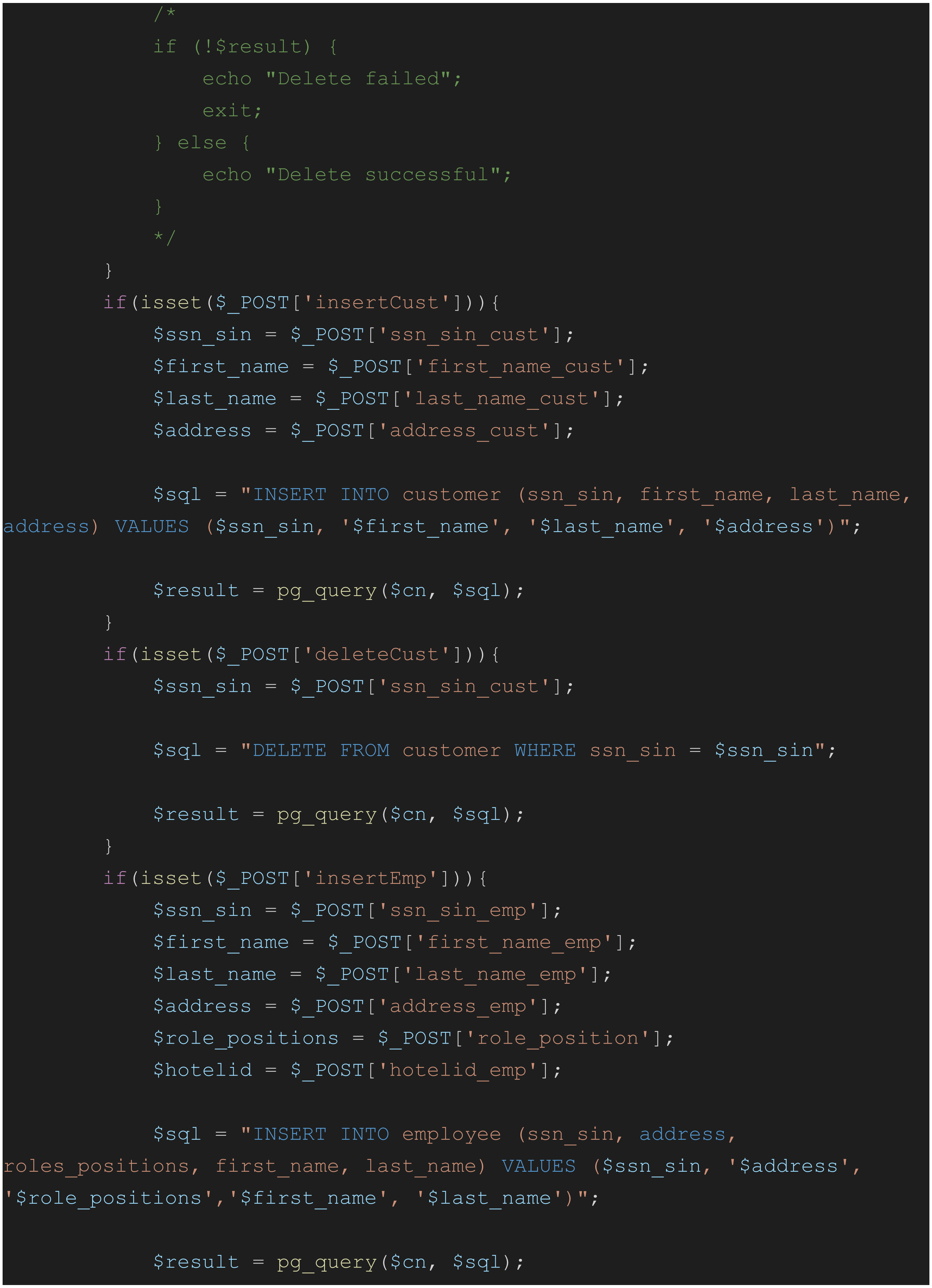


```
    <label for="hotelId">Hotel ID:</label><br>
    <input type="number" name="hotelId"><br>
    <label for="chainName">Chain ID:</label><br>
    <input type="text" name="chainName"><br>
    <label for="rate">Rating:</label><br>
    <input type="number" name="rate"><br>
    <label for="email">Contact Email:</label><br>
    <input type="text" name="email"><br>
    <label for="phoneNum">Phone Number:</label><br>
    <input type="number" name="phoneNum"><br>
    <label for="add">Address:</label><br>
    <input type="text" name="add"><br>
    <label for="areaHotel">Area:</label><br>
    <input type="text" name="areaHotel"><br>

    <br>

        <input type="submit" name="insertHotel" value="insert"/>

    <br>

        <input type="submit" name="deleteHotel" value="delete"/>

    <br>

        <input type="submit" name="updateHotel" value="update"/>
    </form>
    (When Inserting a room you must have all fields filled out)<br>
        (When Deleting, only the Hotel ID is needed)<br>
        (Updates can be done to any attribute other than the Hotel ID and
Room Number. Hotel ID is required for updates)

    <br>
    <br>

    <h3>Customer insert/delete/update</h3>
    <p>
        SSN/SIN is a 9 digit integer starting with 1-9 (no 0) <br>
    </p><
    <br>
    <form method="post">
    <label for="ssn_sin_cust">SSN/SIN:</label><br>
    <input type="number" name="ssn_sin_cust"><br>
    <label for="first_name_cust">First Name:</label><br>
    <input type="text" name="first_name_cust"><br>
    <label for="last_name_cust">Last Name:</label><br>
    <input type="text" name="last_name_cust"><br>
    <label for="address_cust">Address:</label><br>
    <input type="text" name="address_cust"><br>
    <br>

        <input type="submit" name="insertCust" value="insert"/>

    <br>

        <input type="submit" name="deleteCust" value="delete"/>

    <br>

        <input type="submit" name="updateCust" value="update"/>
    </form>
    <p>
        (When Inserting a room you must have all fields filled out)<br>
        (When Deleting, only the SSN/SIN is needed)<br>
        (Updates can be done to any attribute other than the SIN/SSN,
which is required to make any updates)
    </p><br>

    <br>
    <br>

    <h3>Employee insert/delete/update</h3>
    <p>
        SSN/SIN is a 9 digit integer starting with 1-9 (no 0) <br>
        Hotel ID is the hotel id of the hotel the employee is working at
<br>
    </p><br>
    <br>
    <form method="post">
    <label for="ssn_sin_emp">SSN/SIN:</label><br>
```

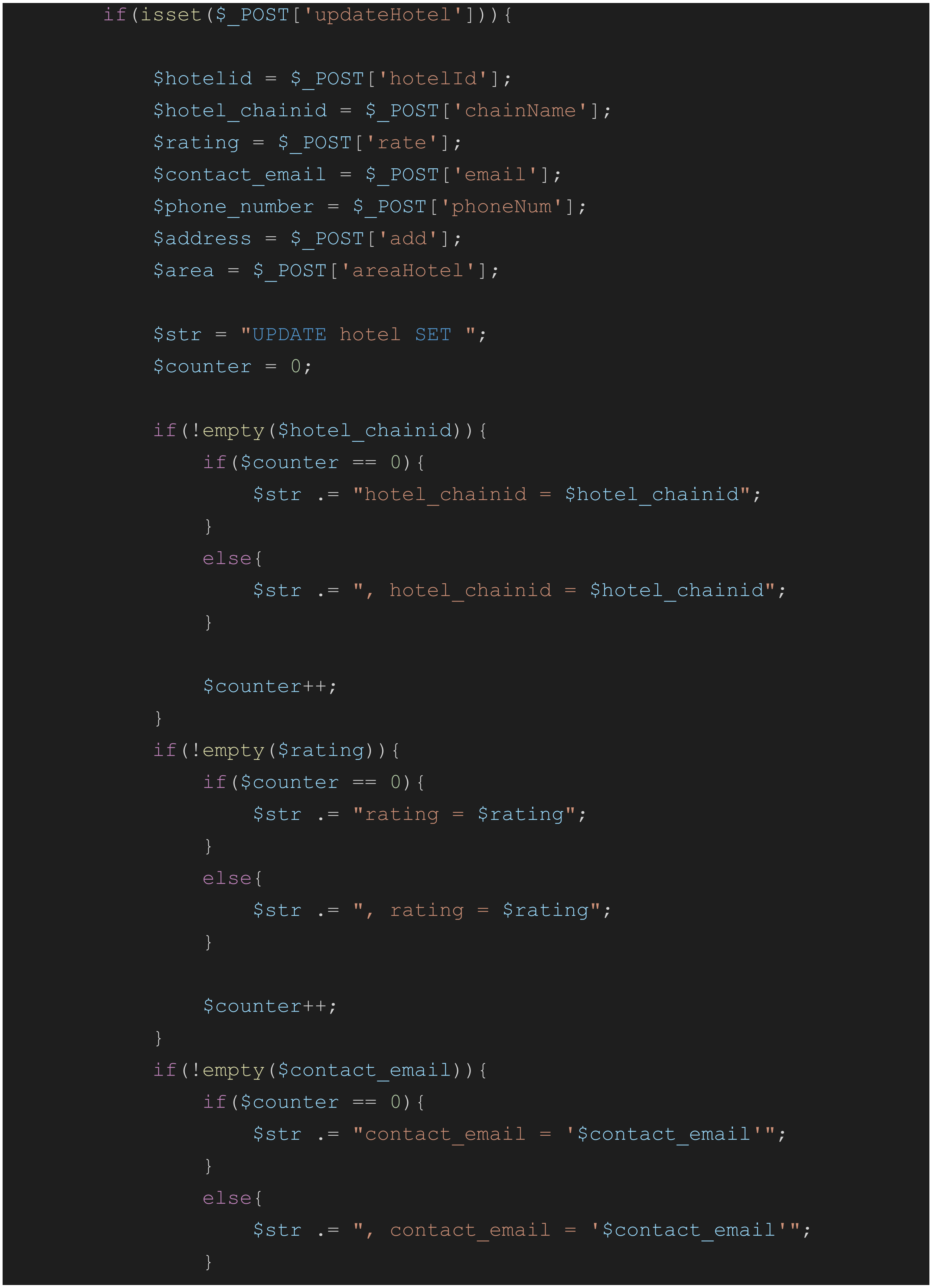
```html
    <input type="number" name="ssn_sin_emp"><br>
    <label for="first_name_emp">First Name:</label><br>
    <input type="text" name="first_name_emp"><br>
    <label for="last_name_emp">Last Name:</label><br>
    <input type="text" name="last_name_emp"><br>
    <label for="address_emp">Address:</label><br>
    <input type="text" name="address_emp"><br>
    <label for="role_position">Role/Position:</label><br>
    <input type="text" name="role_position"><br>
    <label for="hotelid_emp">Hotel Id:</label><br>
    <input type="text" name="hotelid_emp"><br>
    <br>

        <input type="submit" name="insertEmp" value="insert"/>

    <br>

        <input type="submit" name="deleteEmp" value="delete"/>

    <br>

        <input type="submit" name="updateEmp" value="update"/>
    </form>
    <p>
        (When Inserting a room you must have all fields filled out)<br>
        (When Deleting, only the SSN/SIN is needed)<br>
        (Updates can be done to any attribute other than the SIN/SSN, which is required to make any updates)
    </p><br>

    <br>
    <br>

    <h3>Room insert/delete/update</h3>
    <p>
        Hotel Id must be an integer <br>
        Room Number must be an integer <br>
    </p>
    <br>
    <form method="post">
    <label for="hotelid_room">Hotel Id:</label><br>
    <input type="number" name="hotelid_room"><br>
    <label for="roomNumber">Room Number:</label><br>
    <input type="number" name="roomNumber"><br>
    <label for="price">Price:</label><br>
    <input type="double" name="price"><br>
    <label for="amenities">Amenities:</label><br>
    <input type="text" name="amenities"><br>
    <label for="capacity">Capacity:</label><br>
    <input type="number" name="capacity"><br>
    <label for="seaView">Sea View:</label><br>
    <select name="seaView">
        <option value="true">True</option>
        <option value="false">False</option>
    </select><br>
    <label for="mountView">Mountain View:</label><br>
    <select name="mountView">
        <option value="true">True</option>
        <option value="false">False</option>
    </select>
    <br>
    <br>

        <input type="submit" name="insertRoom" value="insert"/>

    <br>

        <input type="submit" name="deleteRoom" value="delete"/>

    <br>

        <input type="submit" name="updateRoom" value="update"/>
    </form>
    <p>
        (When Inserting a room you must have all fields filled out)<br>
        (When Deleting, only the Hotel ID and Room Number are needed)<br>
        (Updates can be done to any attribute other than the Hotel ID and Room Number. Hotel ID and Room Number are also required to make any updates)
    </p><br>
```



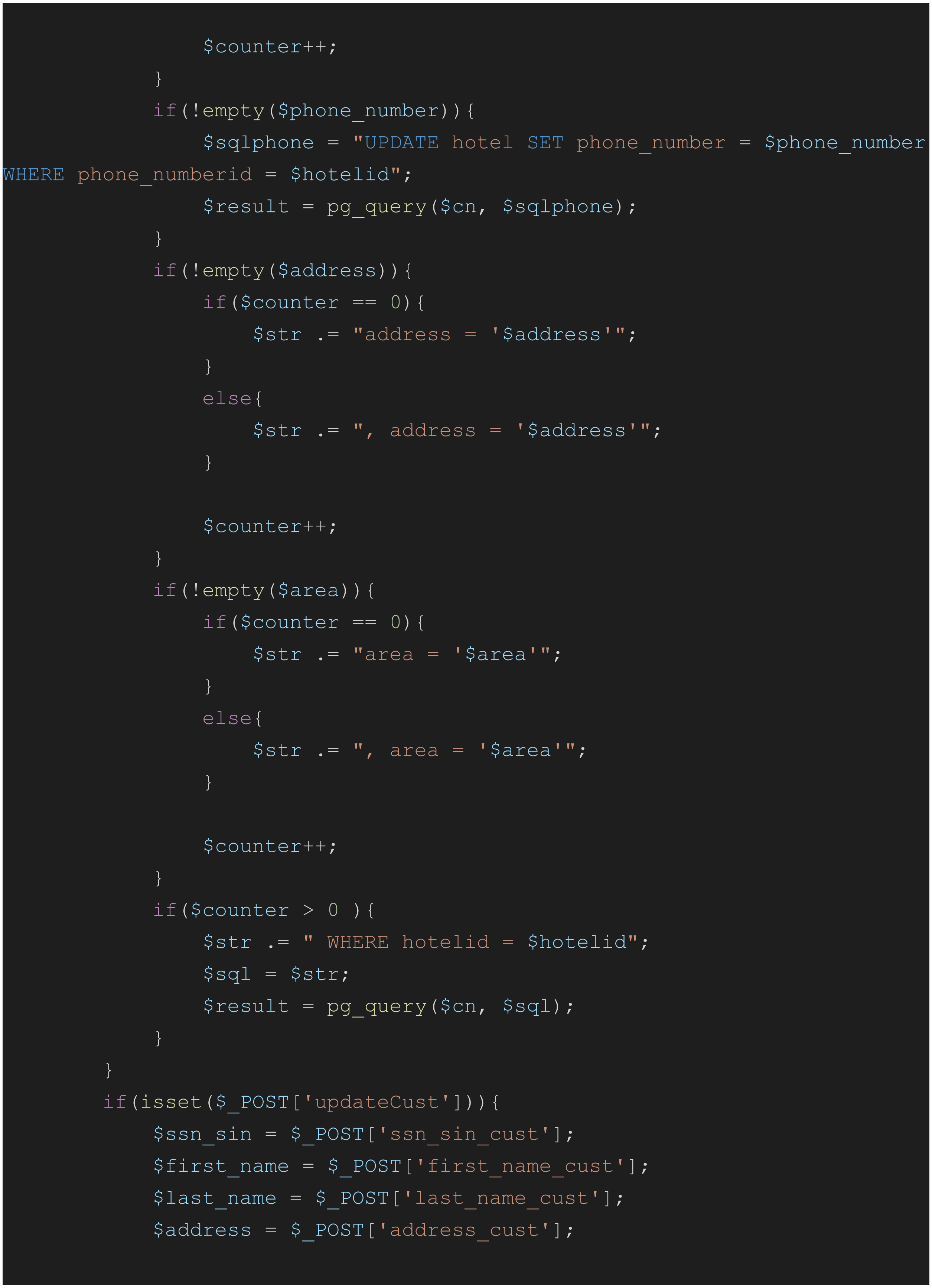
```
    <?php
        if(isset($_POST['insertHotel'])){

            $hotelid = $_POST['hotelId'];
            $hotel_chainid = $_POST['chainName'];
            $rating = $_POST['rate'];
            $contact_email = $_POST['email'];
            $phone_number = $_POST['phoneNum'];
            $address = $_POST['add'];
            $area = $_POST['areaHotel'];

            $sql = "INSERT INTO hotel (hotelid, hotel_chainid, rating,
contact_email, address, area) VALUES ($hotelid, $hotel_chainid, $rating,
'$contact_email', '$address', '$area')";

            $result = pg_query($cn, $sql);

            $sqlPhonenum = "INSERT INTO phone_numbers_hotel
(phone_numberid, phone_number) VALUES ($hotelid, $phone_number)";

            $resultPhonenum = pg_query($cn, $sqlPhonenum);

            /*
            if (!$result) {
                echo "Insert failed";
                exit;
            } else {
                echo "Data inserted successfully";
            }
            */
        }
        if(isset($_POST['deleteHotel'])){

            $hotelid = $_POST['hotelId'];

            $sql = "DELETE FROM hotel WHERE hotelid = $hotelid";

            $result = pg_query($cn, $sql);
```



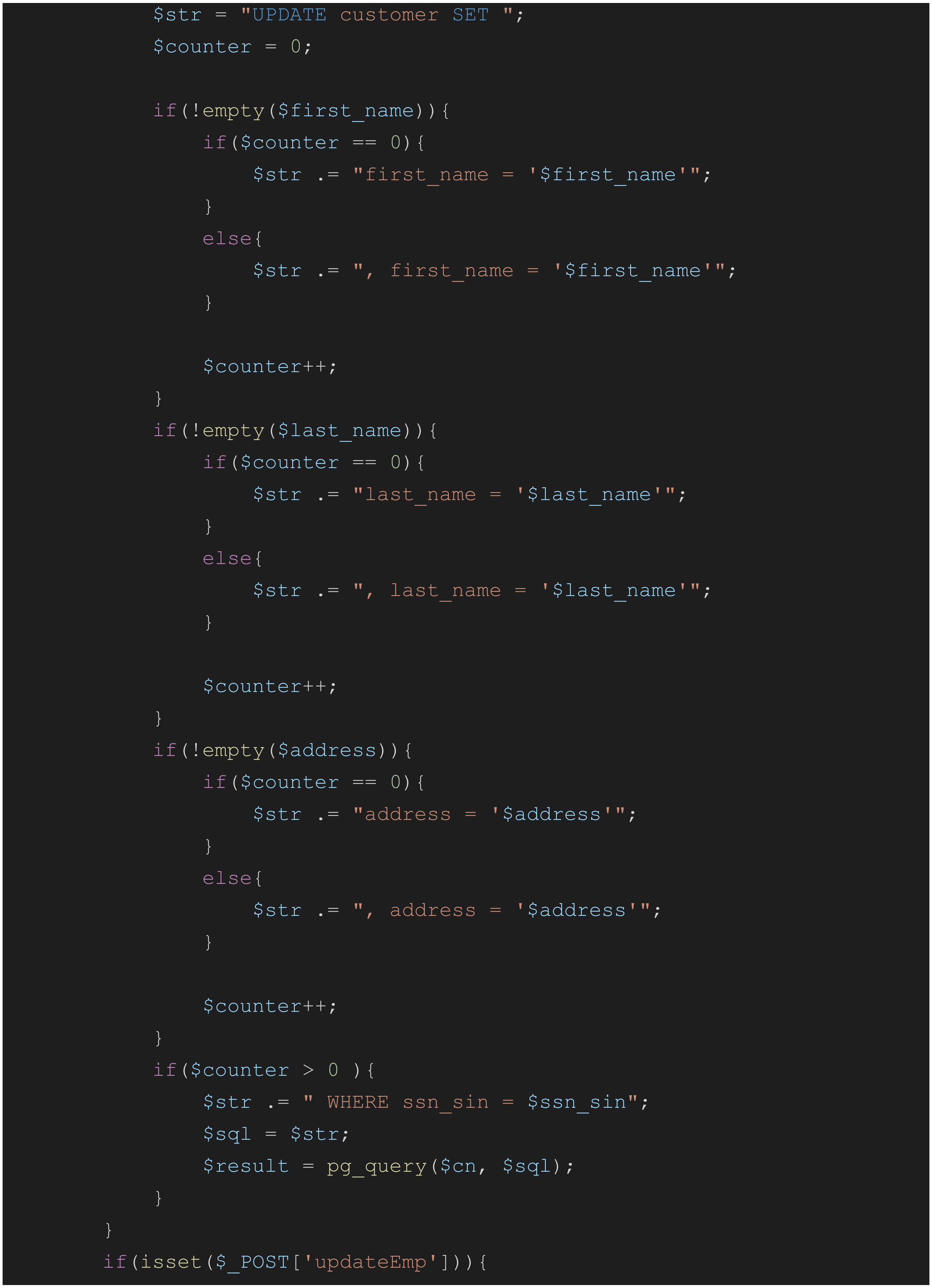
```
            /*
            if (!$result) {
                echo "Delete failed";
                exit;
            } else {
                echo "Delete successful";
            }
            */
        }
        if(isset($_POST['insertCust'])){
            $ssn_sin = $_POST['ssn_sin_cust'];
            $first_name = $_POST['first_name_cust'];
            $last_name = $_POST['last_name_cust'];
            $address = $_POST['address_cust'];

            $sql = "INSERT INTO customer (ssn_sin, first_name, last_name,
address) VALUES ($ssn_sin, '$first_name', '$last_name', '$address')";

            $result = pg_query($cn, $sql);
        }
        if(isset($_POST['deleteCust'])){
            $ssn_sin = $_POST['ssn_sin_cust'];

            $sql = "DELETE FROM customer WHERE ssn_sin = $ssn_sin";

            $result = pg_query($cn, $sql);
        }
        if(isset($_POST['insertEmp'])){
            $ssn_sin = $_POST['ssn_sin_emp'];
            $first_name = $_POST['first_name_emp'];
            $last_name = $_POST['last_name_emp'];
            $address = $_POST['address_emp'];
            $role_positions = $_POST['role_position'];
            $hotelid = $_POST['hotelid_emp'];

            $sql = "INSERT INTO employee (ssn_sin, address,
roles_positions, first_name, last_name) VALUES ($ssn_sin, '$address',
'$role_positions','$first_name', '$last_name')";

            $result = pg_query($cn, $sql);
            $sqlworks = "INSERT INTO works_for (hotelid, ssn_sin) VALUES
($hotelid, $ssn_sin)";

            $resultworks = pg_query($cn, $sqlworks);
        }
        if(isset($_POST['deleteEmp'])){
            $ssn_sin = $_POST['ssn_sin_emp'];

            $sql = "DELETE FROM employee WHERE ssn_sin = $ssn_sin";

            $result = pg_query($cn, $sql);
        }
        if(isset($_POST['insertRoom'])){
            $hotelid = $_POST['hotelid_room'];
            $room_number = $_POST['roomNumber'];
            $price = $_POST['price'];
            $amenities = $_POST['amenities'];
            $capacity = $_POST['capacity'];
            $sea_view = $_POST['seaView'];
            $mountain_view = $_POST['mountView'];

            $sql = "INSERT INTO room (hotelid, room_number, price,
amenities, capacity, sea_view, mountain_view) VALUES ($hotelid,
$room_number, $price, '$amenities', $capacity, '$sea_view',
'$mountain_view')";

            $result = pg_query($cn, $sql);
        }
        if(isset($_POST['deleteRoom'])){
            $hotelid = $_POST['hotelid_room'];
            $room_number = $_POST['roomNumber'];

            $sql = "DELETE FROM room WHERE hotelid = $hotelid AND
room_number = $room_number";

            $result = pg_query($cn, $sql);
        }

        //Update button functionality
```



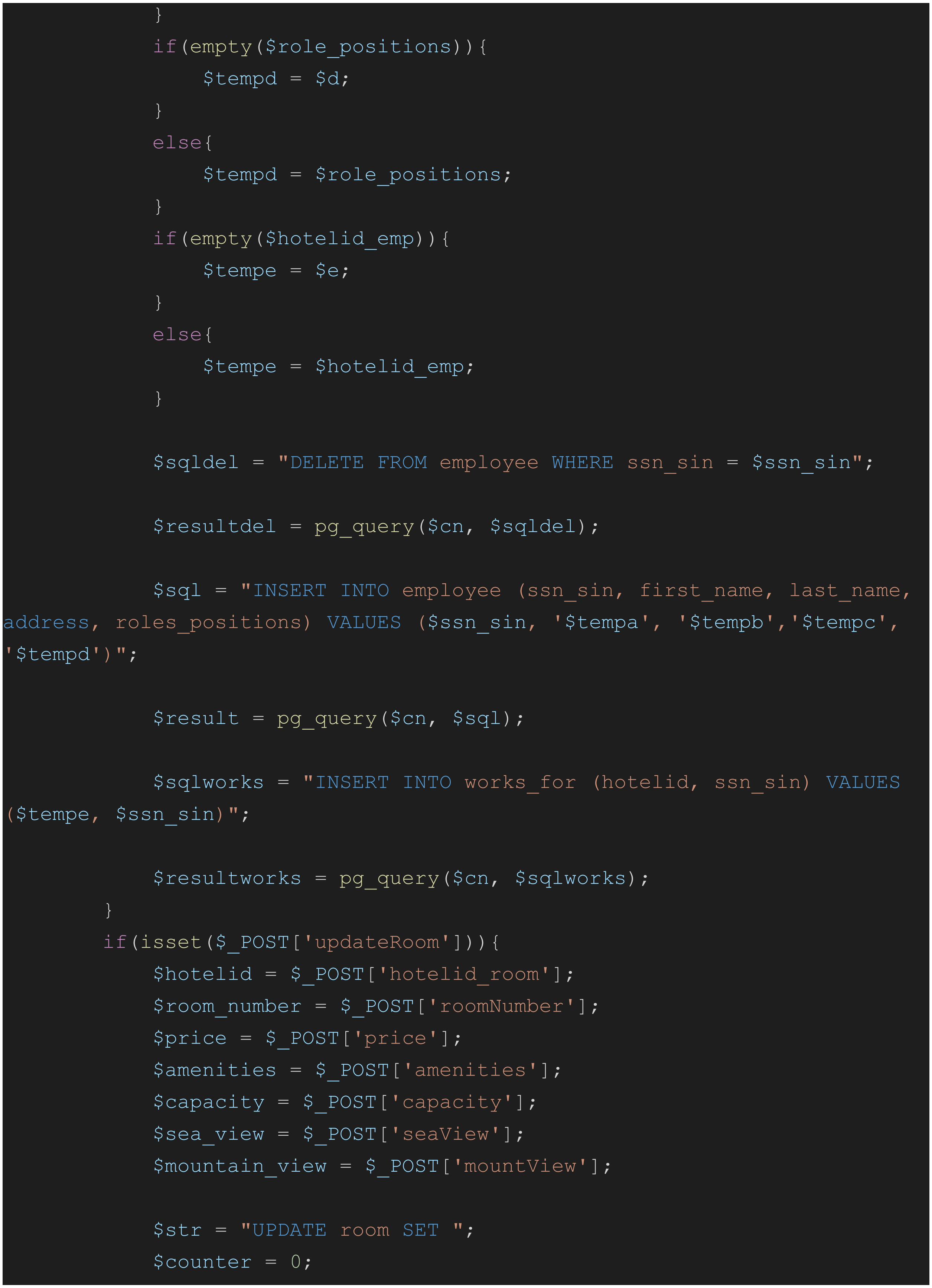
```
        if(isset($_POST['updateHotel'])){

            $hotelid = $_POST['hotelId'];
            $hotel_chainid = $_POST['chainName'];
            $rating = $_POST['rate'];
            $contact_email = $_POST['email'];
            $phone_number = $_POST['phoneNum'];
            $address = $_POST['add'];
            $area = $_POST['areaHotel'];

            $str = "UPDATE hotel SET ";
            $counter = 0;

            if(!empty($hotel_chainid)){
                if($counter == 0){
                    $str .= "hotel_chainid = $hotel_chainid";
                }
                else{
                    $str .= ", hotel_chainid = $hotel_chainid";
                }

                $counter++;
            }
            if(!empty($rating)){
                if($counter == 0){
                    $str .= "rating = $rating";
                }
                else{
                    $str .= ", rating = $rating";
                }

                $counter++;
            }
            if(!empty($contact_email)){
                if($counter == 0){
                    $str .= "contact_email = '$contact_email'";
                }
                else{
                    $str .= ", contact_email = '$contact_email'";
                }
```



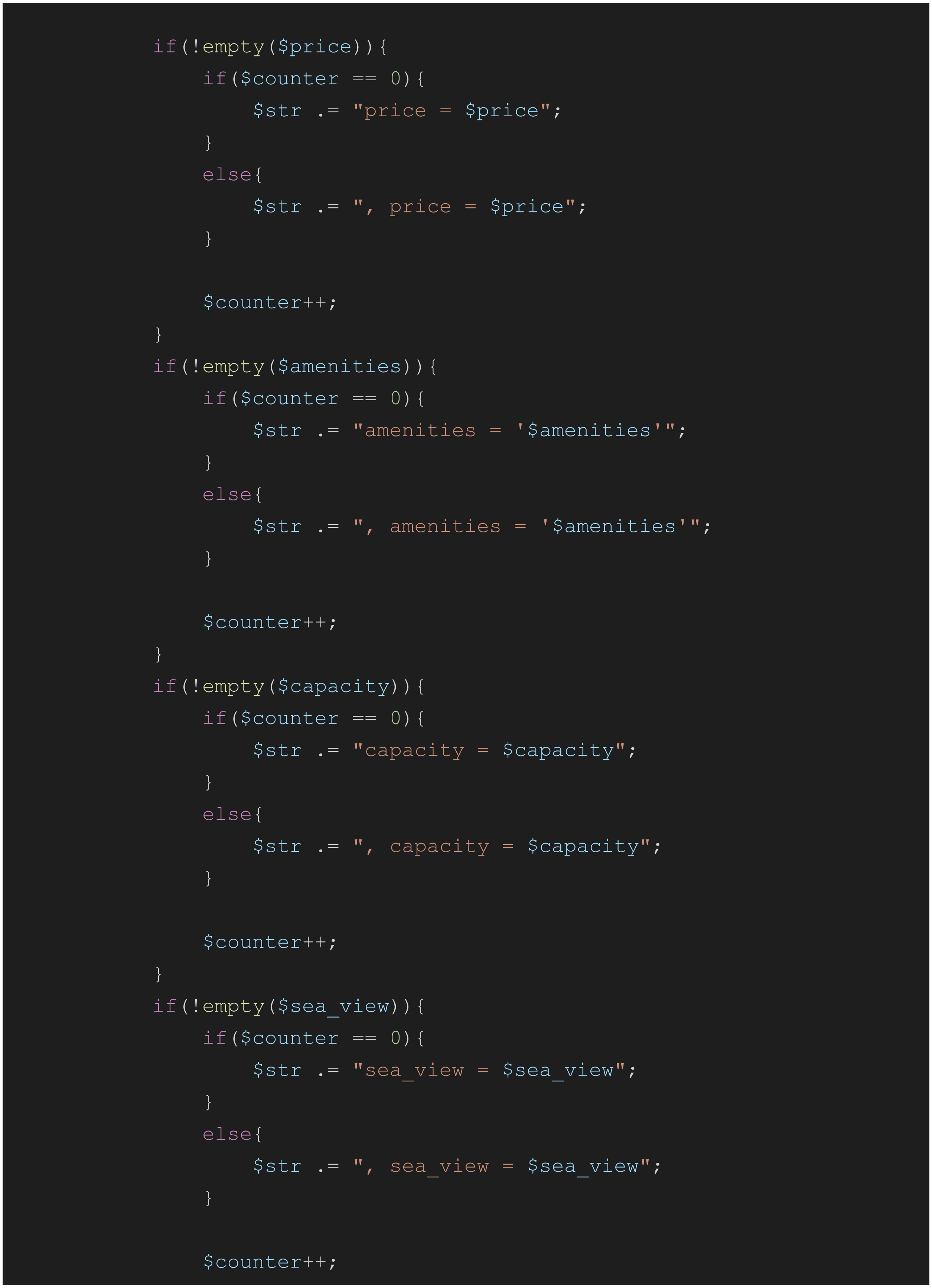
```
                $counter++;
            }
            if(!empty($phone_number)){
                $sqlphone = "UPDATE hotel SET phone_number = $phone_number
WHERE phone_numberid = $hotelid";
                $result = pg_query($cn, $sqlphone);
            }
            if(!empty($address)){
                if($counter == 0){
                    $str .= "address = '$address'";
                }
                else{
                    $str .= ", address = '$address'";
                }

                $counter++;
            }
            if(!empty($area)){
                if($counter == 0){
                    $str .= "area = '$area'";
                }
                else{
                    $str .= ", area = '$area'";
                }

                $counter++;
            }
            if($counter > 0 ){
                $str .= " WHERE hotelid = $hotelid";
                $sql = $str;
                $result = pg_query($cn, $sql);
            }
        }
        if(isset($_POST['updateCust'])){
            $ssn_sin = $_POST['ssn_sin_cust'];
            $first_name = $_POST['first_name_cust'];
            $last_name = $_POST['last_name_cust'];
            $address = $_POST['address_cust'];
```



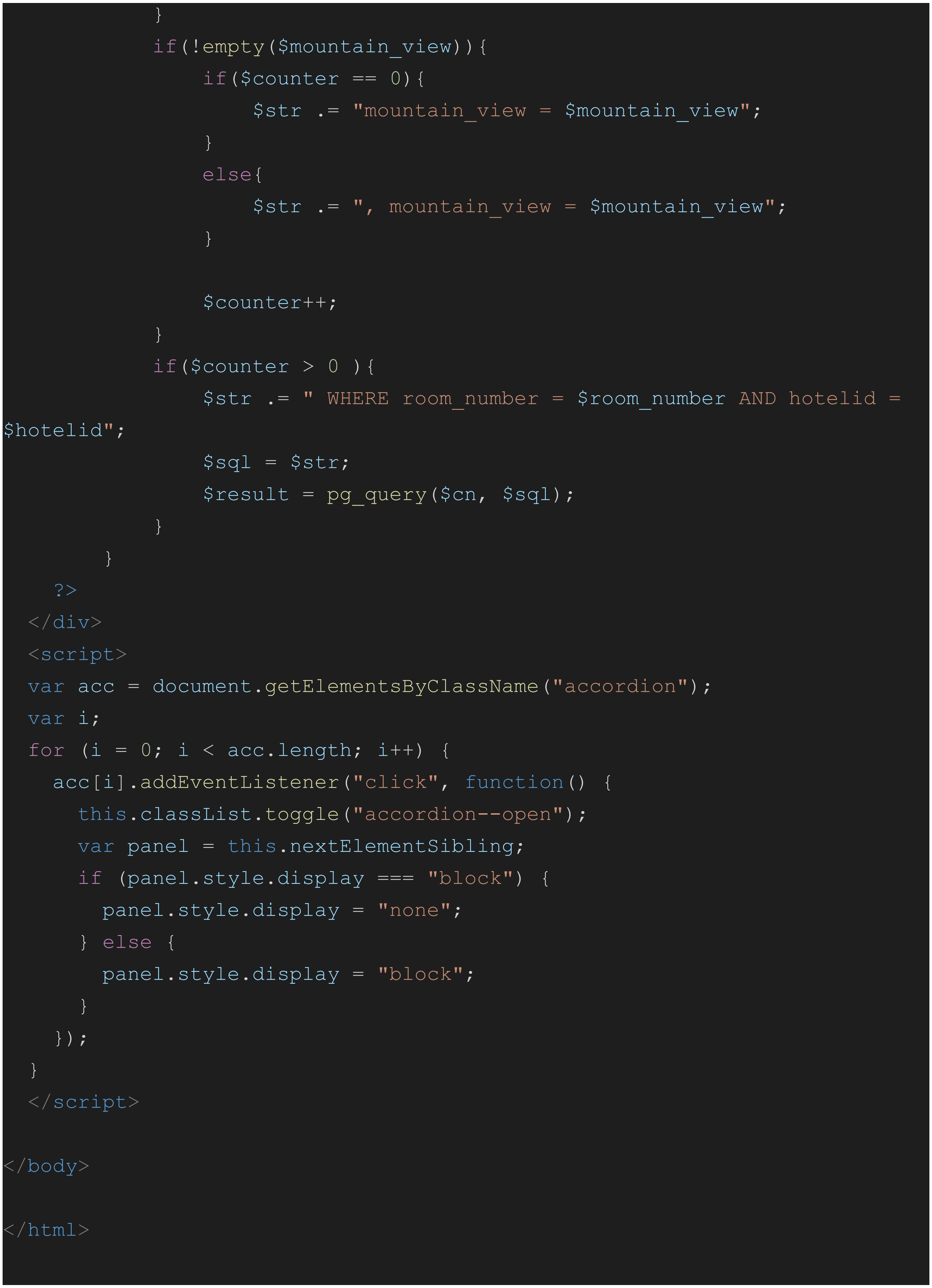
```
            $str = "UPDATE customer SET ";
            $counter = 0;

            if(!empty($first_name)){
                if($counter == 0){
                    $str .= "first_name = '$first_name'";
                }
                else{
                    $str .= ", first_name = '$first_name'";
                }

                $counter++;
            }
            if(!empty($last_name)){
                if($counter == 0){
                    $str .= "last_name = '$last_name'";
                }
                else{
                    $str .= ", last_name = '$last_name'";
                }

                $counter++;
            }
            if(!empty($address)){
                if($counter == 0){
                    $str .= "address = '$address'";
                }
                else{
                    $str .= ", address = '$address'";
                }

                $counter++;
            }
            if($counter > 0 ){
                $str .= " WHERE ssn_sin = $ssn_sin";
                $sql = $str;
                $result = pg_query($cn, $sql);
            }
        }
        if(isset($_POST['updateEmp'])){
            $ssn_sin = $_POST['ssn_sin_emp'];
            $first_name = $_POST['first_name_emp'];
            $last_name = $_POST['last_name_emp'];
            $address = $_POST['address_emp'];
            $role_positions = $_POST['role_position'];
            $hotelid_emp = $_POST['hotelid_emp'];

            $resultrow = pg_query($cn, "SELECT * FROM employee WHERE
ssn_sin=$ssn_sin");
            $empdata = pg_fetch_all($resultrow);

            $resultid = pg_query($cn, "SELECT * FROM works_for WHERE
ssn_sin=$ssn_sin");
            $id = pg_fetch_all($resultid);

            $a = $empdata[0]['first_name'];
            $b = $empdata[0]['last_name'];
            $c = $empdata[0]['address'];
            $d = $empdata[0]['roles_positions'];
            $e = $id[0]['hotelid'];



            if(empty($first_name)){
                $tempa = $a;
            }
            else{
                $tempa = $first_name;
            }
            if(empty($last_name)){
                $tempb = $b;
            }
            else{
                $tempb = $last_name;
            }
            if(empty($address)){
                $tempc = $c;
            }
            else{
                $tempc = $address;
```



```
            }
            if(empty($role_positions)){
                $tempd = $d;
            }
            else{
                $tempd = $role_positions;
            }
            if(empty($hotelid_emp)){
                $tempe = $e;
            }
            else{
                $tempe = $hotelid_emp;
            }

            $sqldel = "DELETE FROM employee WHERE ssn_sin = $ssn_sin";

            $resultdel = pg_query($cn, $sqldel);

            $sql = "INSERT INTO employee (ssn_sin, first_name, last_name,
address, roles_positions) VALUES ($ssn_sin, '$tempa', '$tempb','$tempc',
'$tempd')";

            $result = pg_query($cn, $sql);

            $sqlworks = "INSERT INTO works_for (hotelid, ssn_sin) VALUES
($tempe, $ssn_sin)";

            $resultworks = pg_query($cn, $sqlworks);
        }
        if(isset($_POST['updateRoom'])){
            $hotelid = $_POST['hotelid_room'];
            $room_number = $_POST['roomNumber'];
            $price = $_POST['price'];
            $amenities = $_POST['amenities'];
            $capacity = $_POST['capacity'];
            $sea_view = $_POST['seaView'];
            $mountain_view = $_POST['mountView'];

            $str = "UPDATE room SET ";
            $counter = 0;
```



```
        if(!empty($price)){
            if($counter == 0){
                $str .= "price = $price";
            }
            else{
                $str .= ", price = $price";
            }

            $counter++;
        }
        if(!empty($amenities)){
            if($counter == 0){
                $str .= "amenities = '$amenities'";
            }
            else{
                $str .= ", amenities = '$amenities'";
            }

            $counter++;
        }
        if(!empty($capacity)){
            if($counter == 0){
                $str .= "capacity = $capacity";
            }
            else{
                $str .= ", capacity = $capacity";
            }

            $counter++;
        }
        if(!empty($sea_view)){
            if($counter == 0){
                $str .= "sea_view = $sea_view";
            }
            else{
                $str .= ", sea_view = $sea_view";
            }

            $counter++;
```



```
            }
            if(!empty($mountain_view)){
                if($counter == 0){
                    $str .= "mountain_view = $mountain_view";
                }
                else{
                    $str .= ", mountain_view = $mountain_view";
                }

                $counter++;
            }
            if($counter > 0 ){
                $str .= " WHERE room_number = $room_number AND hotelid =
$hotelid";
                $sql = $str;
                $result = pg_query($cn, $sql);
            }
        }
    ?>
  </div>
  <script>
  var acc = document.getElementsByClassName("accordion");
  var i;
  for (i = 0; i < acc.length; i++) {
    acc[i].addEventListener("click", function() {
      this.classList.toggle("accordion--open");
      var panel = this.nextElementSibling;
      if (panel.style.display === "block") {
        panel.style.display = "none";
      } else {
        panel.style.display = "block";
      }
    });
  }
  </script>

</body>

</html>
```