PHP:

PHP stands for Hypertext Preprocessor. PHP is a server side scripting language. It is used to develop dynamic websites. PHP is an object-oriented language. PHP scripts are executed on the server. PHP file extension .php. PHP is an Open-Source scripting language. It was created by Ramsum Lerdorf in 1994. PHP runs on various platform likes Windows, Linux, Unix, Mac OS etc.

Basic PHP Syntax:

A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document.

```
<?php
    // code statements
?>
```

For example:

```
<?php
        echo "Hello World";
?>
```

Variables:

Variable is the name of memory location. In other word we can say it is user defined name which is given by the user. Variable is a container that store any types of values. In PHP, a variable is declared using a $ sign, followed by the variable name.

Syntax:

```
$variablename=value;
```

For example:

```
<?php
        $str="Hello world!";
        $a=5;
        $b=9.5;
        echo "String is: $str <br/>";
        echo "Integer is: $x <br/>";
        echo "Float is: $y <br/>";
?>
```

Rules to declare variable in PHP:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name can't start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($str and $STR both are two different)

Data Types:

PHP Data types specify the different types of data that are supported in PHP language. They are

- String
- Integer
- Float
- Boolean
- Array
- Object
- NULL

String

String data type in PHP and in general, is a sequence of characters (or anything, it can be numbers and special characters too) enclosed within quotes. You can use single or double quotes.

For example:

```
<?php
        $str1 = "Hello";
        $str2 = "Word";
        $str3 = "123";
        var_dump($str1);
        echo "Str1 :".$str1."<br>"."Str2: ".$str2."<br>"."Str3: ".$str3;
?>
```

Integer:

An Integer data type is used to store any non-decimal numeric value within the range -2,147,483,648 to 2,147,483,647. An integer value can be negative or positive, but it cannot have a decimal

For example:

```
<?php
  $a=5;
  $b=10;
  $sum=$a+$b;
  echo "Sum : ".$sum;
?>
```

Float

Float data type is used to store any decimal numeric value. A float value can also be either negative or positive.

For example:

```
<?php
  $a=5.90;
  $b=10.55;
  $sum=$a+$b;
  echo "Sum : ".$sum;
?>
```

Boolean:

A Boolean represents two possible states: TRUE or FALSE.

For example:

```
$a = true;
echo $a."<br>";
var_dump($a);
```

Array:

An array stores multiple values in one single variable.

For example:

```
<?php
        $studentName=array("Nawaraj","Abhiraj","Rahul","Bipin","Pramod");
        echo "Length of Array:".count($studentName)."<br>";
        echo "Student Names are
        :".$studentName[0].",".$studentName[1].",".$studentName[2].",".$studentName[
        3].",".$studentName[4]."<br>";
        $arrayLength=count($studentName);
        for($i=0;$i<$arrayLength;$i++){
          echo $studentName[$i]."<br>";
        }
?>
```

Constants:

Constants are variables whose value cannot be changed. In other words, once you set a value for a constant, you cannot change it.

- Using the define() method.
- Using the const keyword.

For example:

defind() method:

```
<?php
  define("cons", "Hello Word!");
  echo cons;
?>
```

const keyword:

```
<?php
  const cons = "Hello Word!";
  echo cons;
?>
```

Operators:

Operators is a symbol which is used to perform operations on operands. In PHP there are total 7 types of operators, they are:

1. Arithmetic Operators
2. Assignment Operators
3. Comparison Operators
4. Increment/Decrement Operators
5. Logical Operators
6. String Operators
7. Array Operators

There are a few additional operators as well like, Type operator, Bitwise operator, Execution operators etc.

Based on how these operators are used, they are categorized into 3 categories:

1. Unary Operators: Works on a single operand (variable or value).
2. Binary Operators: Works on two operands (variables or values).
3. Ternary Operators: Works on three operands.

Flow Controls:

There are three types of non-looping conditionals, the if statement, else statement and switch statement.

If statement:

The if statement is used to check some given condition. If condition is true then the statement inside the body of if is executed. Otherwise if condition is false then the statement inside the body of if is skipped.

Syntax of if statement:

```
jf(condition)
{
        Block of statements;
        ……..
}
```

For example:

```
<?php
        $a = 5;
        if($a <= 25)
        {
          echo "Value of a is greater than 25";
        }
?>
```

else statement:

The else statement is used to perform two operations for a single condition. The else statement evaluates the specified condition. If it is true, it executes a block of statements. If the condition is false, it executes another block of statements. We must notice that if and else block cannot be executed simultaneously.

Syntax of if else statement:

```
if(condition)
{
        true block of statements;
}
else
{
        false block of statements;
}
```

For example:

```
<?php
        $a = 5;
        if($a %2==0)
        {
          echo "A is event number";
        }else{
          echo "A is odd number";
        }
?>
```

if-else-if statement:

The if-else-if conditional Statement is used to execute one code from multiple conditions. It is also known as a multi-path decision statement. It is a sequence of if-else statements where every if statement is associated with else if Statement and last would be an else statement.

Syntax of if-else-if statement:

```
if(condition1)
{
        true block of statements1;
}
else if(condition2)
{
        false block of condition1;
and
        True block of condition2
}
```

For example:

```
<?php
        $a = 5;
        if($a==10)
        {
          echo "A is equeal to 10";
        }else if($a==15)
        {
          echo "A is equeal to 15";
        }else if($a==50)
        {
          echo "A is equeal to 50";
        }else{
          echo "A is not equal to 10, 15 or 50";
        }
?>
```

switch...case Statement:

Switch statement is used to compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is executed. Each case in a block of a switch has a different choice which is referred to as an identifier. When the user enters a value, then the value provided by the user is compared with all the cases inside the switch block until the exact match is found. If a case match is not found, then the default statement is executed, and the control goes out of the switch block.

Syntax of switch...case:

```
switch (expression)
{
        case value1:
                //Statements
        break;
        case value 2:
                //Statements
        break;
        case value n:
                //Statements
        break;
        Default:
           //Statements
}
```

For example:

```
<?php
        $a = 3;
        switch($a){
          case 1:
            echo "Sunday";
            break;
          case 2:
            echo "Monday";
            break;
          case 3:
            echo "Tuesday";
            break;
          case 4:
            echo "Wednesday";
            break;
            case 5:
```

```
        echo "Thursday";
        break;
    case 6:
            echo "Firday";
            break;
    default:
    echo("Default case executed");
    }
?>
```

Looping Statements:

Looping statements repeatedly execute a statement or a block of statements.

- for loop
- while loop
- do-while loop
- for-in loop

for loop statement:

The for loop statement is used to execute a single statement or a block of statements repeatedly as long as the given condition is true.

Syntax of for loop statement:

```
for(initialization; condition; increment/decrement)
{
        statement-block;
}
```

- **Initialization:** First, a value is assigned to the variable.
- **Condition:** The following step is to check the condition; if it returns false, the for loop is ended. However, if the condition returns true, the statements inside the for loop's body are executed.
- **Increment/decrement:** In the last step, variable is incremented or decremented. Again the condition, statement inside body of loop and increment/decrement operation repeated until the condition expression is false and the loop terminates for the final time

For example:

```
<?php
        for($i=1;$i<=5;$i++)
        {
                echo $i."<br>";
        }
?>
```

while Loop statements:

When the condition is true, then the while loop's statements will be executed, otherwise not.

Syntax of while Loop statements:

```
variable initialization;
while(condition)
{
        statements;
        variable increment/decrement;
}
```

For example:

```
<?php
      $a = 1;
      while($a <= 5)
      {
        echo $a."<br>";
        $a++;
      }
?>
```

do while loop statement:

do while loop will be executed at least once, even though the condition is false.

Syntax of do while loop:

```
do {
        // code to be executed
} while (condition);
```

For example:

```
<?php
      $a = 5;
      do
      {
        echo $a."<br>";
        $a++;
      } while($a <= 10)
      ?>
```

foreach loop:

The for-in loop is a special type of a loop that iterates over the properties of an object, or the elements of an array.

Syntax of foreach loop:

```
foreach($variable as $object)
{
            // Code to be executed
}
```

For example:

```
<?php
        $studentName = array("Kabi", "Raj", "Abhiraj", "Annaya");
        foreach($studentName as $list)
        {
           echo "$list <br/>";
        }
?>
```

Function:

A Function is nothing but a 'block of statements' which generally performs a specific task and can be used repeatedly in our program. In PHP there are thousands of built-in functions which we can directly use in our program/script. PHP also supports user defined functions, where we can define our own functions. A function doesn't execute when its defined, it executed when it is called.

Types of function:

- Built-in Functions
- User Defined Functions

User Defined Functions:

We can define our own functions in our program and use those functions.

Syntax:

```
<?php
        function function_name()
        {
           // function code statements
        }
?>
```

For example:

```
<?php
        function display()
        {
         echo "Hello.<br>";
        }
        function show()
        {
         echo "Welcome to User Defind Function.<br>";
        }
        display();
        show();

?>
```

Parameters to Functions:

We can call function by passing arguments.

Syntax of parameters of function:

```
        function functionName($parameter1, $parameter2)
        {
          // Code to be executed
        }
```

For example:

```
<?php
        function display($name)
        {
         echo "Hello : $name.<br>";
        }
        display("Abhiraj");

        function display1($fname,$lname)
        {
         echo "Hello $fname $lname.<br>";
        }
        display1("Nawaraj","Prajapati");

        function sum($a,$b)
        {
         echo $a + $b;
        }
        sum(5,10.99);
?>
```

Function Returning Values:

```
<?php
        function add($a, $b)
        {
          $sum = $a + $b;
          return $sum;
        }
        echo "Sum= " . add(5, 10) ;
?>
```

Class and Object:

A class is a user-defined data type which includes local variables and local methods. While an **object** is an instance of the class which holds the local variables with values assigned and on which we can call the local methods defined in the class.

Syntax:

The syntax for defining a class in PHP is very simple, we use the keyword class followed by the name of the class and then we enclose the code for the class within curly braces { } just like for a function/method.

```
<?php
        class MyClass
         {
           // Class properties and methods go here
         }
?>
```

Syntax of an object:

```
  $obj = new MyClass;
```

For example:

```
<?php
        class Student
        {
                var $name = "Abhiraj";
                function display()
                {
                        echo "Welcome to PHP";
                }
         }
         $obj = new Student();
         echo $obj->name . "<br/>";

        $obj->display();
?>
```

PHP Forms:

When we develop a website or a web application. We have to create forms to take input from users, like a login form or a registration form. Creating a form on the webpage is accomplished using HTML, while PHP servers as a transport for those values from the webpage to the server and then in further processing those values. PHP provides two methods $_GET and $_POST for collecting form data for processing.

For example:

formsExample.html

```
<html>
<head>

   <title>Page Title</title>

</head>
<body>
   <form action="formsExample.php" method="post">
      <table align="center">
         <tr><td>User Name:</td><td><input type="text" name="uname"></td></tr>
         <tr><td>Address:</td><td><input type="text" name="address"></td></tr>
         <tr><td>Age:</td><td><input type="text" name="age"></td></tr>
         <tr><td><input type="submit" value="Save"></td></tr>
      </table>

   </form>
</body>
</html>
```

formsExample.php

```
<?php
         $nname=$_POST['uname'];
         $aaddress=$_POST['address'];
         $aage=$_POST['age'];

         echo "User Name:$nname","<br>","Address : $aaddress","<br>","Age : $aage";

?>
```

State Management in PHP:

PHP provides for two different techniques for state management of your web application, they are:

1. Server Side State Management
2. Client Side Server Management

Server Side State Management

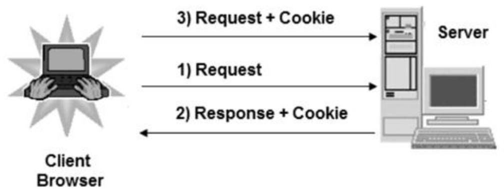
In server side state management we store user specific information required to identify the user on the server. And this information is available on every webpage. In PHP we have Sessions for server side state management. PHP session variable is used to store user **session** information like username, userid etc and the same can be retrieved by accessing the session variable on any webpage of the web application until the session variable is destroyed.

Client Side State Management

In client side state management the user specific information is stored at the client side i.e. in the bowser. Again, this information is available on all the webpages of the web application. In PHP we have Cookies for client side state management. Cookies are saved in the browser with some data and expiry date(till when the cookie is valid).One drawback of using **cookie** for state management is the user can easily access the cookie stored in their browser and can even delete it.

Cookies:

Cookie is a small piece of information stored as a file in the user's browser by the web server. Once created, cookie is sent to the web server as header information with every HTTP request. You can use cookie to save any data but it should not exceed 1K(1024 bytes) in size.



Types of Cookies

There are two types of cookies, they are:

1. Session Cookie
2. Persistent Cookie

Session Cookie: This type of cookies are temporary and are expire as soon as the session ends or the browser is closed.

Persistent Cookie: To make a cookie persistent we must provide it with an expiration time. Then the cookie will only expire after the given expiration time, until then it will be a valid cookie.

For example:

```
//Create cookies
<?php
        setcookie("name","Nawaraj");
        setcookie("id", "101");
?>
//Retrieve cookies in PHP

<?php
        if(isset($_COOKIE["name"]) && isset($_COOKIE["id"]))
        {
        echo " The name = " .$_COOKIE["name"]. "<br/>";
        echo "The id = ". $_COOKIE["id"];
        }
        else
        {
        echo "Sorry !! cookies is not set.";
        }

?>
```

Delete a Cookie:

```
        <?php
                setcookie("user", "", time() - 3600);
        ?>
        <?php
                echo "Cookie 'user' is deleted.";
        ?>
```

Example of CRUD Operation:

dbConnection.php

```
<?php
        $host="localhost";
        $user="root";
        $pass="admin";
        $db="db_php_student";
        $con=mysqli_connect($host,$user,$pass,$db);
        if($con){
          echo "Database Connection Created SuccessFully";
        }
        else
        {
          echo "Database Not Connected";
        }
?>
```

insertData.php

```
<?php include 'dbConnection.php'; ?>
<html>
<head>
  <title>Insert Page</title>
</head>
<body bgcolor="skyblue">

<form action="" method="post" name="formName">
  <table align="center">
    <tr><td>FirstName</td><td><input type="text" name="fname" required></td></tr>
    <tr><td>LastName</td><td><input type="text" name="lname" required></td></tr>
    <tr><td>Address</td><td><input type="text" name="address" required></td></tr>
    <tr><td>Age</td><td><input type="text" name="age" required></td></tr>
    <tr><td>Nationality:</td><td>
    <select name="nationality" size="1">
      <option selected value="Select nationality">Select nationality</option>
      <option value="Nepal">Nepal</option>
      <option value="Indian">Indian</option>
      <option value="US">US</option>
      <option value="UK">UK</option>
      <option value="UAE">UAE</option>
      <option value="Others">Others</option>
    </select>
</td></tr>
    <tr><td><input type="submit" value="Save" name="btn_save"></td><td><button><a
href="view.php">View</a></button></td></tr>
  </table>
</form>
<?php
if(isset($_POST['btn_save'])){
   $fname=$_POST['fname'];
   $lname=$_POST['lname'];
   $address=$_POST['address'];
   $age=$_POST['age'];
   $nationality=$_POST['nationality'];
   $query="insert into
tb_insertdata(firstname,lastname,address,age,nationality)values('$fname','$lname','$address','$age','$natio
nality')";
   $data=mysqli_query($con,$query);

   if($data){
      ?>
      <script>
         alert("Data has been Saved Successfully");
      </script>
      <?php
   }
   else
   {
      ?>
      <script>
         alert("Please Try Again!!!");
      </script>
      <?php
   }

}
?>
</body>
</html>
```

View.php

```
<?php include 'dbConnection.php'; ?>
<html>
<body bgcolor="skyblue">
        <center>
                <a href="insertData.php">Home</a>
        </center>
        <table border="1px" cellpadding="10px" cellspacing="0px" align="center">
                <tr>
                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>Address</th>
                        <th>Age</th>
                        <th>Nationality</th>
                        <th colspan="2">Action</th>
                </tr>
                <?php
   $query="select *from tb_insertdata";
   $data=mysqli_query($con,$query);
```

```
$result=mysqli_num_rows($data);
if($result){
   while($row=mysqli_fetch_array($data)){
      ?>
                <tr>
                        <td>
                                <?php echo $row['firstname'];?>
                        </td>
                        <td>
                                <?php echo $row['lastname'];?>
                        </td>
                        <td>
                                <?php echo $row['address'];?>
                        </td>
                        <td>
                                <?php echo $row['age'];?>
                        </td>
                        <td>
                                <?php echo $row['nationality'];?>
                        </td>
                        <td><a href="update.php?id=<?php echo $row['id']; ?>">Update</a>
                        </td>
                        <td><a
                                onclick="return confirm('Are You Sure, You want to Delete?')"
                                href="delete.php?id=<?php echo $row['id']; ?>">Delete</a>
                        </td>
                </tr>

                <?php

   }
}

else
{
   ?>
                <tr>
                        <td>No Record Found!!!</td>
                </tr>
                <?php
}

?>

        </table>
</body>
</html>
```

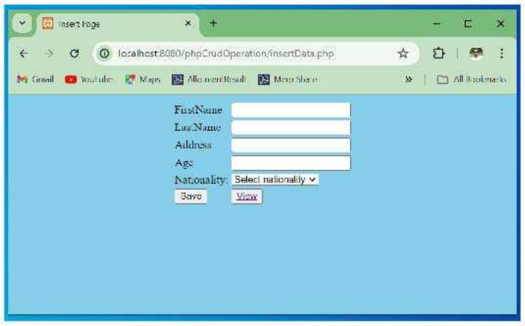
delete.php

```
<?php include 'dbConnection.php'; ?>
<?php
$id=$_GET['id'];
$query="delete from tb_insertdata where id='$id'";
$data=mysqli_query($con,$query);
if($data){
   ?>
<script>
      alert("Data Deleted Successfully");
      window.open("http://localhost:8080/phpCrudOperation/view.php","_self");
   </script>
<?php
   }
   else
   {
      ?>
<script>
         alert("Please Try Again!!!");
      </script>
<?php
   }
   ?>
```

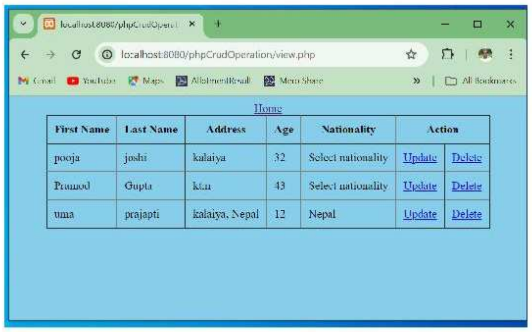
update.php

```
<?php include 'dbConnection.php'; ?>
<?php
   $id=$_GET['id'];
   $query="select *from tb_insertdata where id='$id'";
   $data=mysqli_query($con,$query);
   $row=mysqli_fetch_array($data);
?>
<body bgcolor="skyblue">
         <form action="" method="post">
                  <table align="center">
                           <tr>
                                    <td>FirstName</td>
                                    <td><input type="text" value="<?php echo $row['firstname'] ?>"
                                             name="fname">
                                    </td>
                           </tr>
                           <tr>
                                    <td>LastName</td>
                                    <td><input type="text" value="<?php echo $row['lastname'] ?>"
                                             name="lname">
                                    </td>
                           </tr>
                           <tr>
                                    <td>Address</td>
                                    <td><input type="text" value="<?php echo $row['address'] ?>"
```

```
                                        name="address">
                                </td>
                        </tr>
                        <tr>
                                <td>Age</td>
                                <td><input type="text" value="<?php echo $row['age'] ?>"
                                        name="age">
                                </td>
                        </tr>
                        <tr>
                                <td>Nationality:</td>
                                <td><select name="nationality" size="1">
                                                <option selected value="Select nationality">Select
                                                        nationality</option>
                                                <option value="Nepal">Nepal</option>
                                                <option value="Indian">Indian</option>
                                                <option value="US">US</option>
                                                <option value="UK">UK</option>
                                                <option value="UAE">UAE</option>
                                                <option value="Others">Others</option>
                                </select></td>
                        </tr>

                        <tr>
                                <td><input type="submit" value="Update" name="btn_update">
                                </td>
                                <td><button>
                                                <a href="view.php">Back</a>
                                        </button>
                                </td>
                        </tr>
                </table>
        </form>
</body>
<?php
if(isset($_POST['btn_update'])){
   $fname=$_POST['fname'];
   $lname=$_POST['lname'];
   $address=$_POST['address'];
   $age=$_POST['age'];
   $nationality=$_POST['nationality'];
   $query="update tb_insaertdata set
firstname='$fname',lastname='$lname',address='$address',age='$age',nationality='$nationality' where
id='$id'";
   $data=mysqli_query($con,$query);

   if($data){
      ?>
<script>
        alert("Data has been Updated Successfully");
        window.open("http://localhost:8080/phpCrudOperation/view.php", "_self");
</script>
<?php
    }
    else
    {
      ?>
<script>
          alert("Please Try Again!!!");
</script>
<?php
    }

}
?>
```



FirstName	
LastName	
Address	
Age	
Nationality:	Select nationality
Save	View



localhost:8080/phpCrudOperat
localhost:8080/phpCrudOperation/view.php
Home

First Name	Last Name	Address	Age	Nationality	Action	
pooja	joshi	kalaiya	32	Select nationality	Update	Delete
Pramod	Gupta	ktm	43	Select nationality	Update	Delete
uma	prajapti	kalaiya, Nepal	12	Nepal	Update	Delete

CodeIgniter:

CodeIgniter is an open-source PHP web application framework. It provides a compact and quick method for creating dynamic web pages and web apps. It is easy to install CodeIgniter, and its directory structure is intuitive. It has a vibrant network for support and is well-documented.

Installation Instructions:

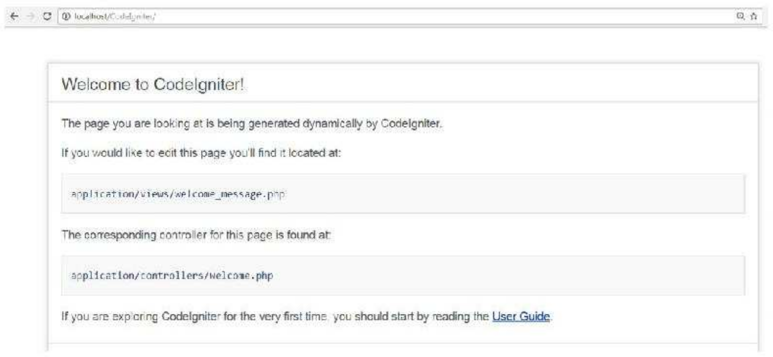
Follow given steps to install CodeIgniter:

1) Download CodeIgniter from its official website.

Download current version of CodeIgniter from its official website

https://www.codeigniter.com

2) Unzip CodeIgniter package.
Downloaded CodeIgniter will be in zip format. Copy it and place it in your htdocs folder. Unzip and rename it. We are naming it as CodeIgniter.
3) CodeIgniter user guide


Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
application/views/welcome_message.php
```

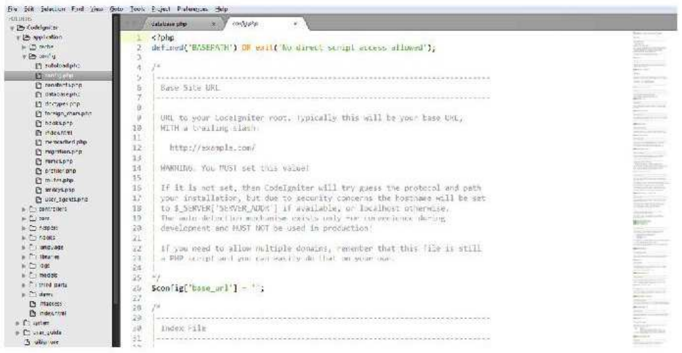
The corresponding controller for this page is found at:

```
application/controllers/Welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the User Guide.

On browser type **localhost/CodeIgniter/** (after localhost type name of your unzipped folder). If the above snapshot page appears then it means your file is successfully installed.

4) Set the base URL in application/config/config.php file with any text editor.



```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
|--------------------------------------------------------------------------
| Base Site URL
|--------------------------------------------------------------------------
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|   http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
*/
$config['base_url'] = '';

/*
|--------------------------------------------------------------------------
| Index File
|--------------------------------------------------------------------------
```

5) You need to establish the connectivity to your database. Go to the path application/config/database.php file

Laravel:

Laravel is a free open-source PHP framework. It provides web developers tools and resources for building modern PHP web applications.