

How-To: Create a website using HTML5, CSS, PHP, & MySQL

Overview

In this tutorial, you will find step-by-step instructions and resources to help you create a web application from scratch using HTML5, CSS, PHP and MySQL. We will cover file directories, LocalHost, Open-Source design and web server tools we will be using to help us create Front-End and Back-End files that will make up the full web application.



Before we begin, please be aware, what you are about to see is available online written by other authors. We wanted to find one that work, modify it to our needs, and put it back online for you to use. You totally do this on your own if you find the right documentation. It is a good idea though to give the originator of the document or instructions the credit that she or he deserves. We will be adding comments of our own to the original documentation so they fit our fleet of How-To samples. If this sample is not what you were looking for, or this is too basic for your taste and capabilities, keep looking through our website for other/similar samples or different programming methodologies.

For the purpose of this tutorial, we are using the following programming code and design methodologies:

1. **HTML5** — is the newest version of the markup language used for structuring and presenting content on the World Wide Web.
2. **CSS** — describes and enhances HTML elements to be displayed on screens or other media
3. **PHP** — is a highly popular server-side scripting language. It is very flexible and has a lot of innovative features. It is the fastest resource available for creating database-enabled dynamic websites.
4. **MySQL** — is a database management system based on Structured Query Language (SQL) which is used in web applications.

In order to write a successful or efficient web application, it is necessary to find the right tools to work with. And you will need to have the tools that fits the specific type of application you are writing. In this case we downloaded Notepad++ to manage our Front-End and Back-End code and design. We have also downloaded an Open Source web server called XAMPP, a widely-popular tool that most beginner to expert application developers use to run our HTML5, CSS, PHP, and MySQL code; we will go into XAMPP a little more in this tutorial. We are sub-dividing our application development into two parts, the Front-End, controlled by our HTML5, CSS files, and the Back-End, and PHP with MySQL as our Back-End. Please be aware, PHP can also be seen as a Front-End development mean, since it contains some of the design elements for creating a webpage. Without further-a-do, let us go into the process to building our web application.

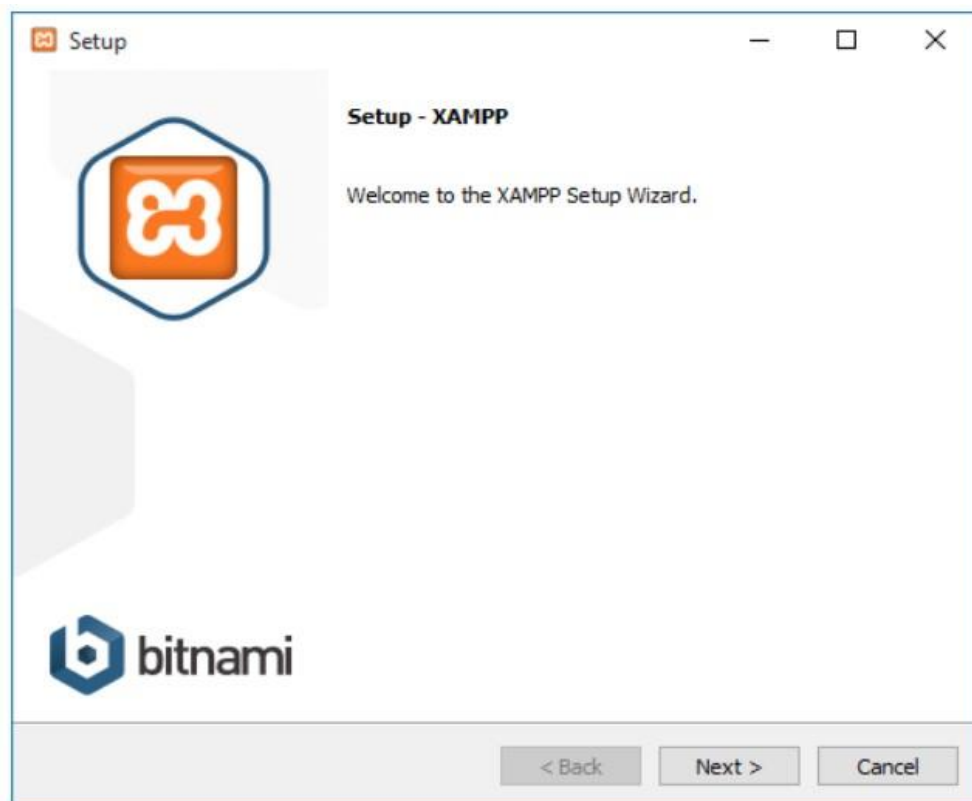
What is XAMPP?

XAMPP is a simple and lightweight solution that allows web application developers to create a local web server. You will need to install XAMPP to create and run the sample application. So go ahead and download the XAMPP Installer. Choose the version that is compatible with your computer; a link to the website to download XAMPP is provided below.

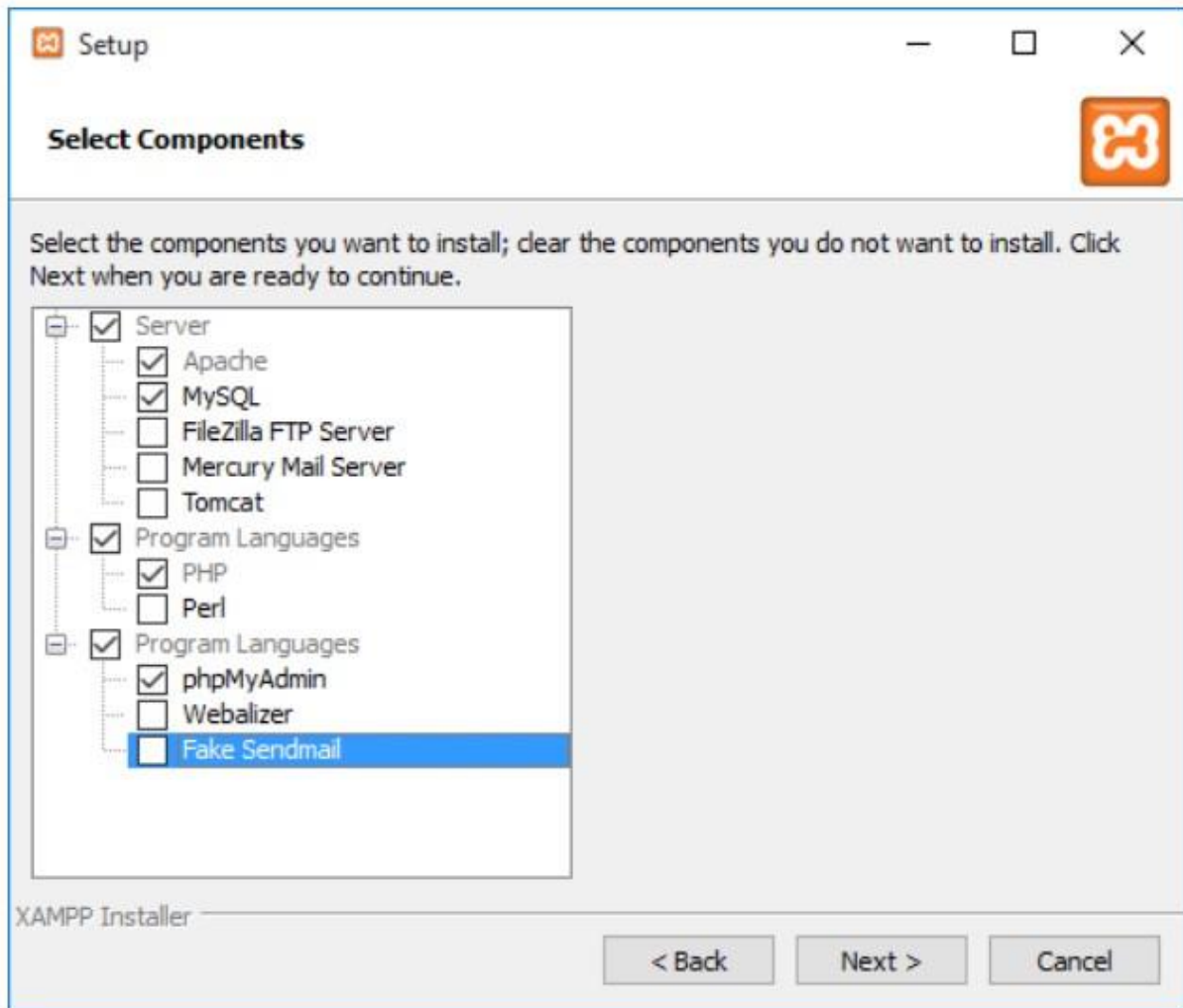
<https://www.apachefriends.org/index.html>

Start the installation

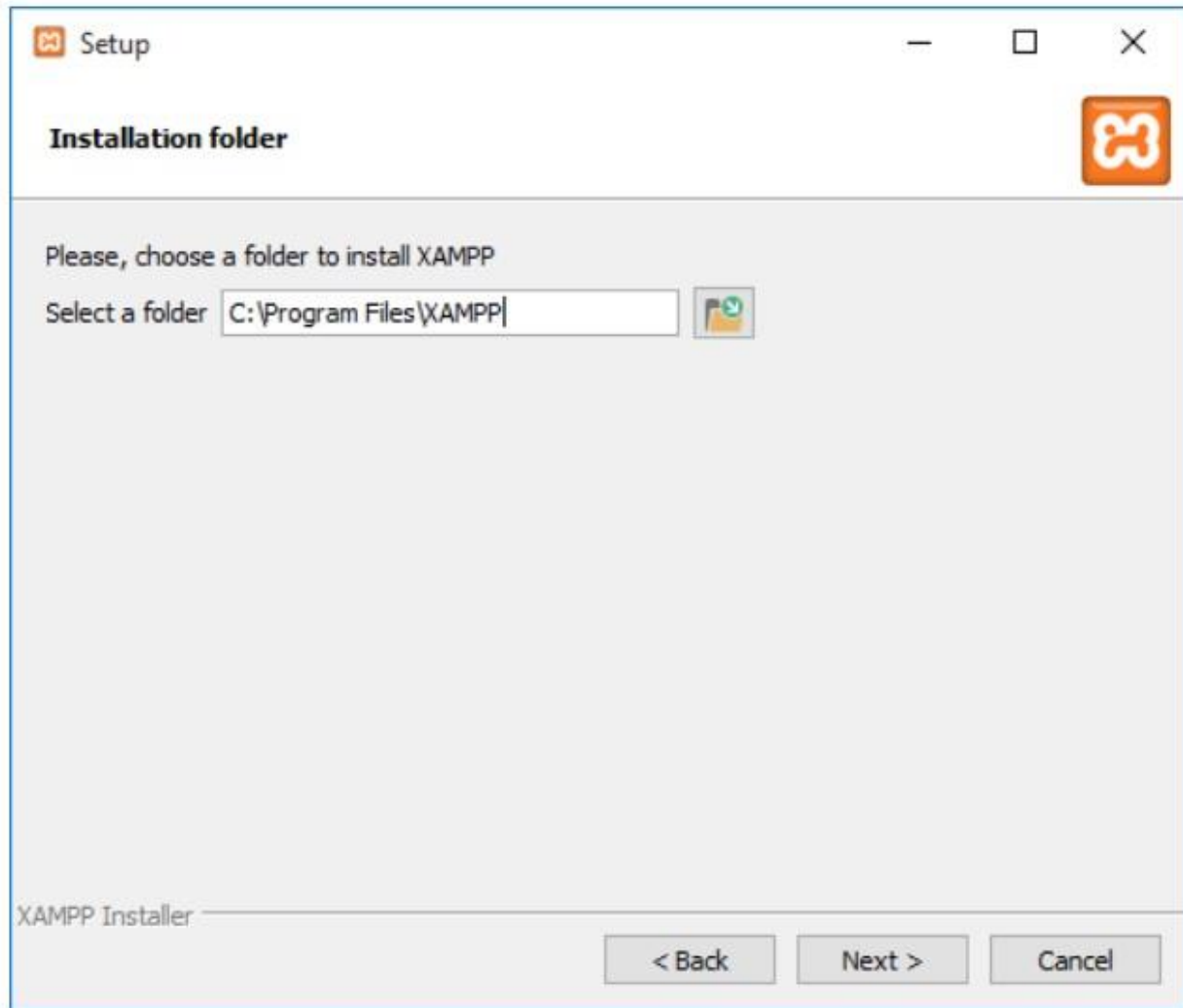
Next, just run the installer file and continue the installation. I've attached a screenshot so you can follow the configuration.



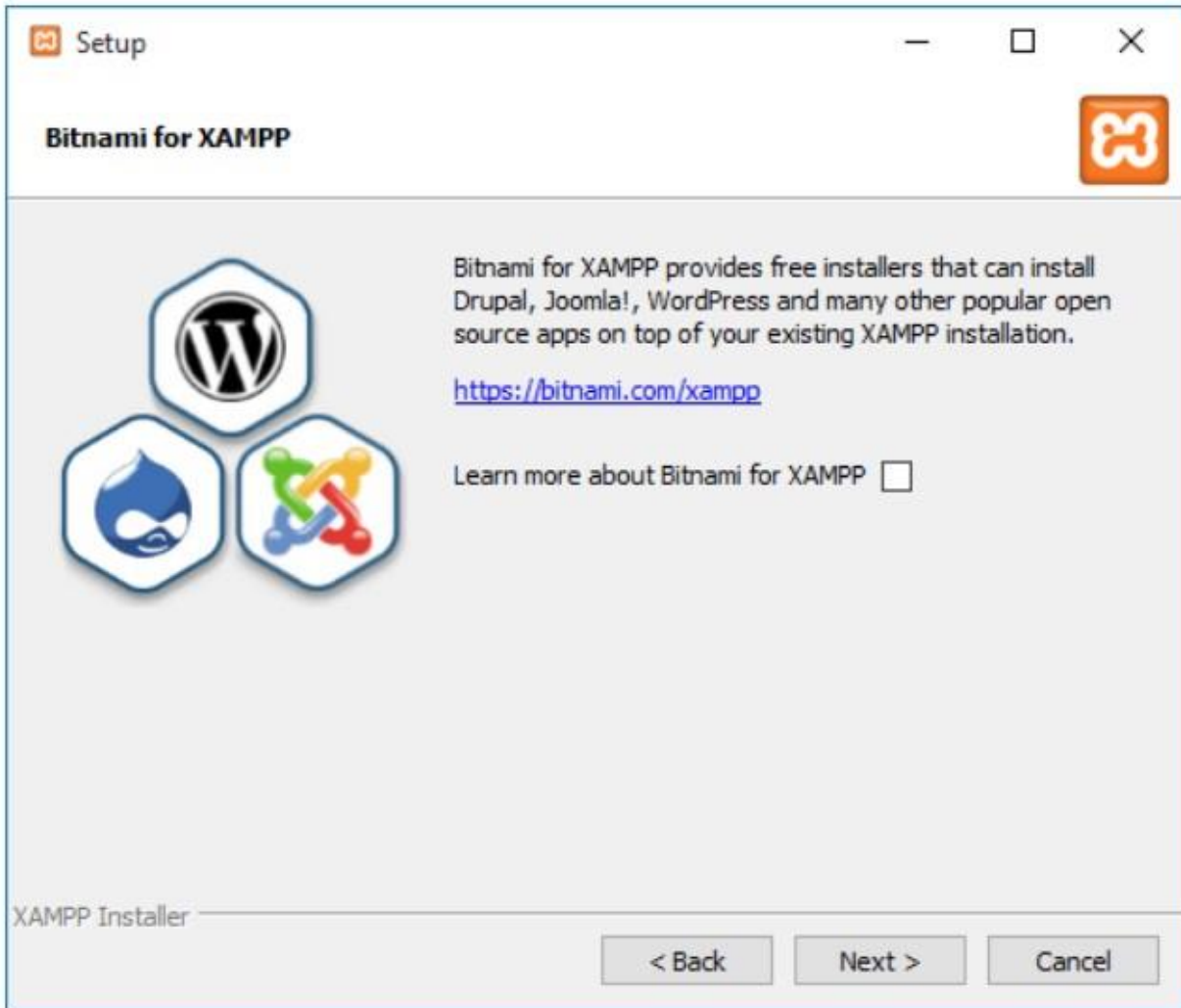
Click Next to continue the installation.



Choose a preferred installation directory for XAMPP. Click Next to continue.

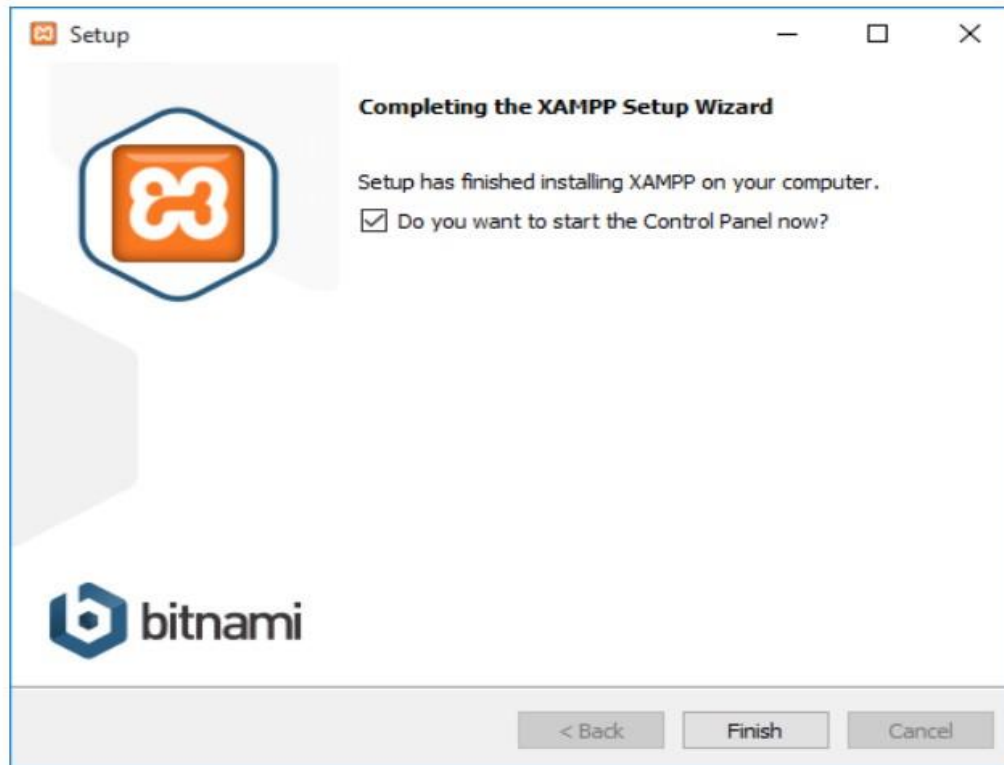


XAMPP will begin the initial installation. Click Next to continue.

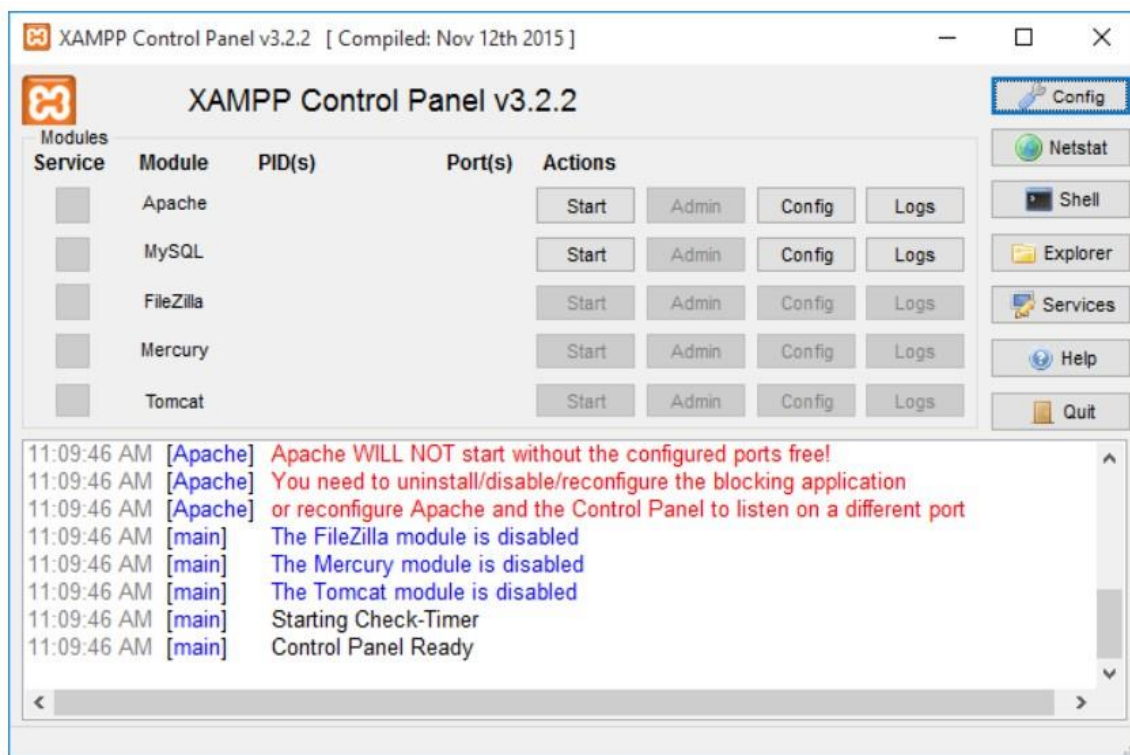




Once all of the files have been installed. Click Finish to complete the installation.



Next, you need to start Apache & MySQL manually. Just click the Start button.



Once the configuration is done, go to <http://localhost/> in your web browser. Now, you have successfully installed your local server.

Front-End Coding

First, you will need a text editor to create and edit your web application files. There are many options on the Internet. We downloaded Notepad++, it is an Open-Source tool and it is very easy to use. We are including a link to the website where you can download it.

<https://notepad-plus-plus.org/download/>

Step 1

Assuming you already have the editor installed:

1. Go to the directory where you installed your version of XAMPP
2. Find this folder (htdocs) in the following directory *C:\Program Files\XAMPP\htdocs*
3. Inside the htdocs folder, create a main web application folder, call it *my_folder*
4. Your directory should look as such *C:\Program Files\XAMPP\htdocs\my_folder*
5. Now, launch Notepad++ and create a new php called index.php
6. Save the new file in your main application folder “my_folder”
7. Your directory should now match this: *C:\Program Files\XAMPP\htdocs\my_folder\index.php*

Technical Notes: If you do not know how to create a file in Notepad++ there are tutorials online to help you.

8. Inside the index.php file. Write the following chunk of code, then save the file. Now, test the link.
9. You’ve successfully created your first php page for your web application. Now Launch it!

```
1  <html>
2      <head>
3          <title>My First Project</title>
4      </head>
5      <body>
6          <h1>Homepage</h1>
7      </body>
8  </html>
```

Provided that XAMPP is currently running, open your preferred browser and add the following web address in the address bar http://localhost/my_folder/index.php, then hit your enter key. If XAMPP is configured properly, you see the text Homepage in your browser

Step 2

Now, this is getting more interesting! You will need to create other php files that will be linked to the index.php file/page. Those new php files or pages are independent from one another but they carry design patterns that the index.php page will need to be a complete design; you can pretty call those pages templates, as they can be called from other pages beyond index.php.

1. First, we are going to create some additional folders, css, images, templates
2. Create a Cascading Stylesheets(CSS) file in Notepad++ called main.css
3. Create or save an image as logo.png inside the images folder to be called into index.php
4. Inside the templates pages, create a header.php, menu.php, and footer.php
5. Use the same steps above when we created my_folder and config.php

TIP: Screenshots below should give you an idea how your directory should look

Name	Date modified	Type	Size
css	4/27/2019 9:43 PM	File folder	
images	9/24/2018 10:03 A...	File folder	
templates	9/24/2018 9:19 AM	File folder	
index	4/27/2019 9:44 PM	PHP File	1 KB

Name	Date modified	Type	Size
footer.php	9/24/2018 9:19 AM	PHP File	0 KB
header.php	9/24/2018 9:18 AM	PHP File	0 KB
menu.php	9/24/2018 9:19 AM	PHP File	0 KB

Now, open the index.php file then write the following code:

```
1 <html>
2   <head>
3     <title>My First Project</title>
4     <link href="css/main.css" rel="stylesheet" type="text/css">
5   </head>
6   <body>
7     <!-- HEADER -->
8     <?php include('templates/header.php'); ?>
9
10    <!-- MENU -->
11    <?php include('templates/menu.php'); ?>
12
13    <!-- CONTENT -->
14    <h1>Title</h1>
15    <p>Sample content here</p>
16
17    <!-- FOOTER -->
18    <?php include('templates/footer.php'); ?>
19  </body>
20 </html>
```

Then, open your header.php and put this code:

```
1 <header>
2   
3   <h1>Website Name</h1>
4 </header>
```

Now, launch the webpage (index.php) in your browser. Go to http://localhost/my_folder/index.php, you should see the image on the page.

Back-End Coding

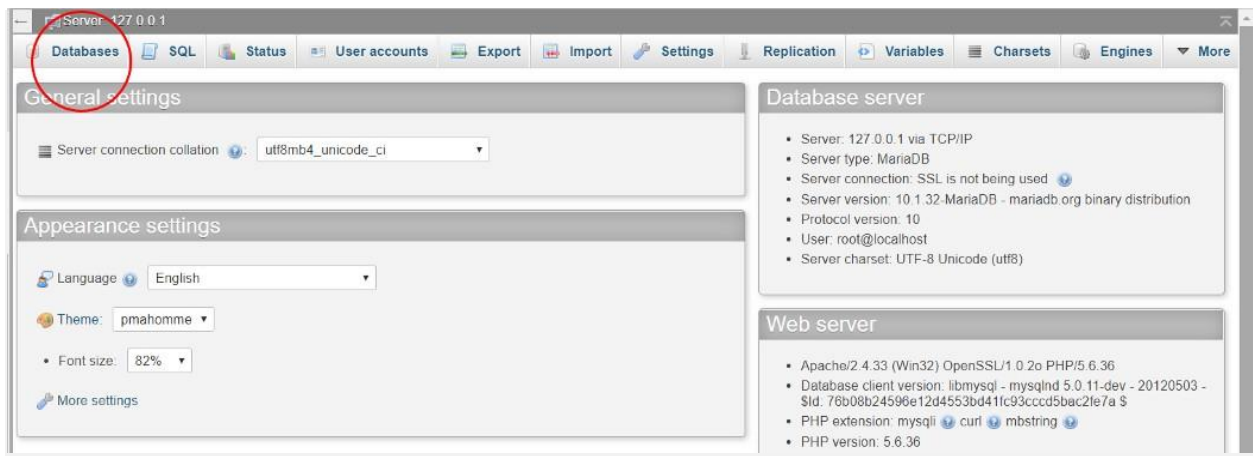
So far so good! We have tested our Front-End files and they work as intended. We are going to dive into our Back-End programming to allow our Front-End (HTML5, CSS, PHP) to communicate with our Back-End (MySQL database).

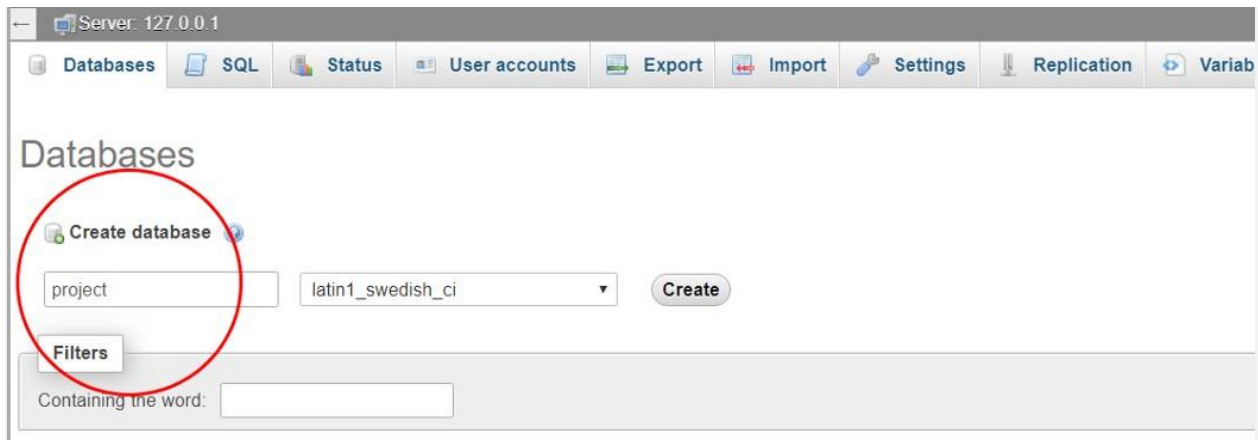
Step 3

Let us set up a MySQL database. For this tutorial, we are going to continue using XAMPP since it comes complete with a MySQL Database. We can administer and manage database files using phpMyAdmin. You can read about phpMyAdmin by going on the Internet but basically, it is a built-in XAMPP tool that allows developers to create and maintain databases to manage stored server (XAMPP) data.

You can access phpMyAdmin the following way in your browser:

1. Go to <http://localhost/phpmyadmin/>, to begin database creation
2. From the menu click Databases
3. In the database field, type the database name.
4. Then click Create button, as seen below

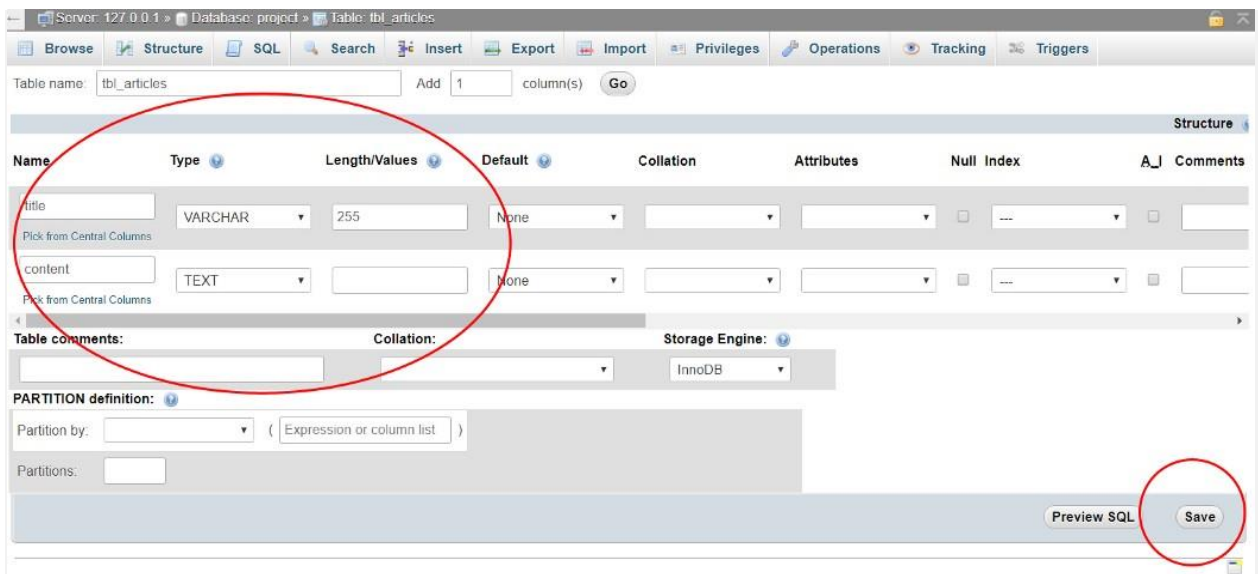




- Now that you have successfully created the database
- create a new table and name it tbl_articles
- Then, edit the number of columns to "2"



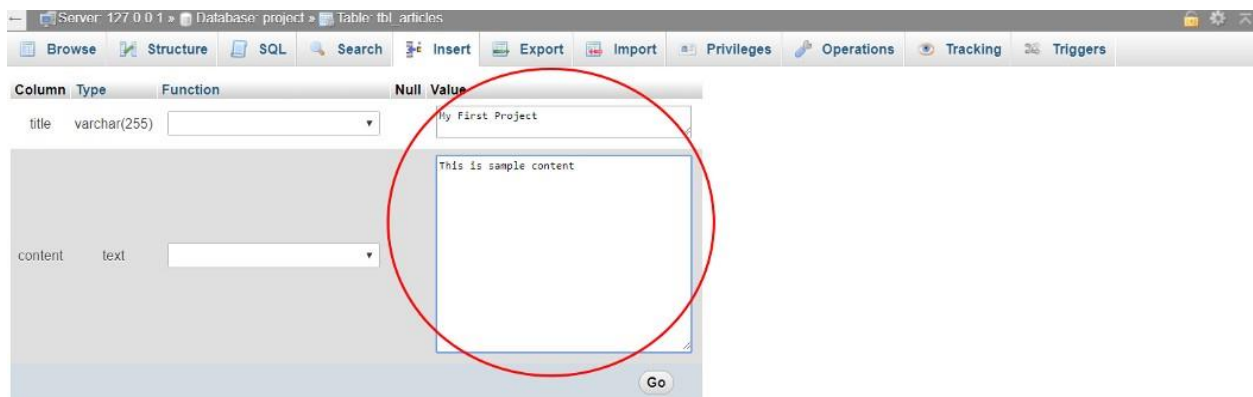
- Enter the following information for each field in the table... as seen in attached screenshot-



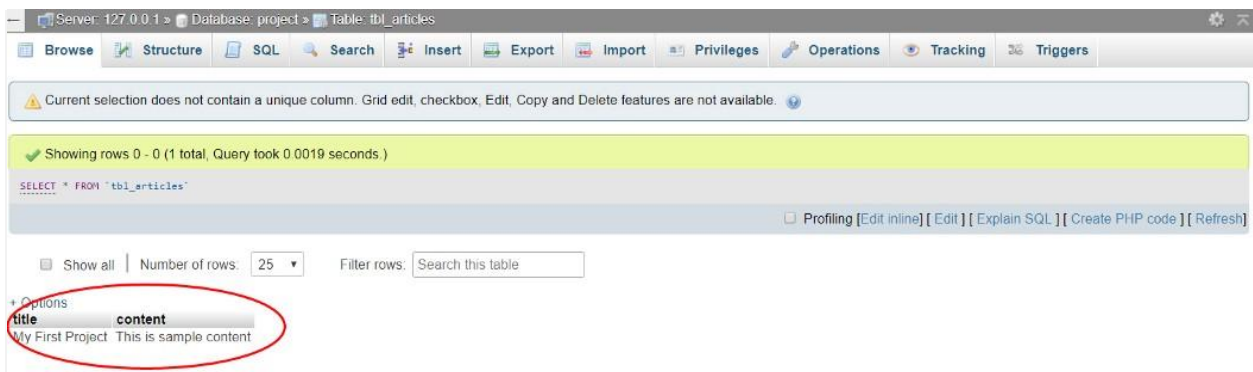
Next, we need to insert data to the fields. From the menu click Insert



Then click Go.



We've got Data! Now, let us see how we can retrieve it from our main index.php file



Step 4

Next, create a folder for the database configuration section for our web application. We will need a folder to collect all configuration files, one being config.php, a file that will contain database connection parameters and SQL statements. The config.php file can then be called from its folder/location within our main index.php file.

The way this works is that your main folder for the application, "my_folder", should have a subfolder inside of it called "include", which will then contain the config.php file.

The directory and file structure should look like the following:

- *C:\Program Files\XAMPP\htdocs\my_folder\include*
- *C:\Program Files\XAMPP\htdocs\my_folder\include\config.php*

Inside the config.php file, write the following snippet:

```
1 <?php
2 //DATABASE CONNECTION
3 $host = "localhost";
4 $username = "root";
5 $password = "";
6 $database = "project";
7
8 $sql = mysqli_connect($host, $username, $password, $database) or die('Could not connect');
9
10 ?>
```

Next, call config.php from within our index.php using the include() function.

Hint: It is a good idea at this point to test the connection to the database to ensure you can reach it. This way, when you run SQL statements and you are getting errors, you can be certain that at least the connection is proper, limiting your troubleshooting steps.

```
1 <?php include('include/config.php'); ?>
2 <html>
3     <head>
4         <title>My First Project</title>
5         <link href="css/main.css" rel="stylesheet" type="text/css">
6     </head>
7     <body>
8         <!-- HEADER -->
9         <?php include('templates/header.php'); ?>
10
11     <!-- MENU -->
12     <?php include('templates/menu.php'); ?>
13
14     <!-- CONTENT -->
15     <h1>Title</h1>
16     <p>Sample content here</p>
17
18     <!-- FOOTER -->
19     <?php include('templates/footer.php'); ?>
20     </body>
21 </html>
```

Let us write a simple SQL statement to retrieve the data submitted to the back-end earlier...

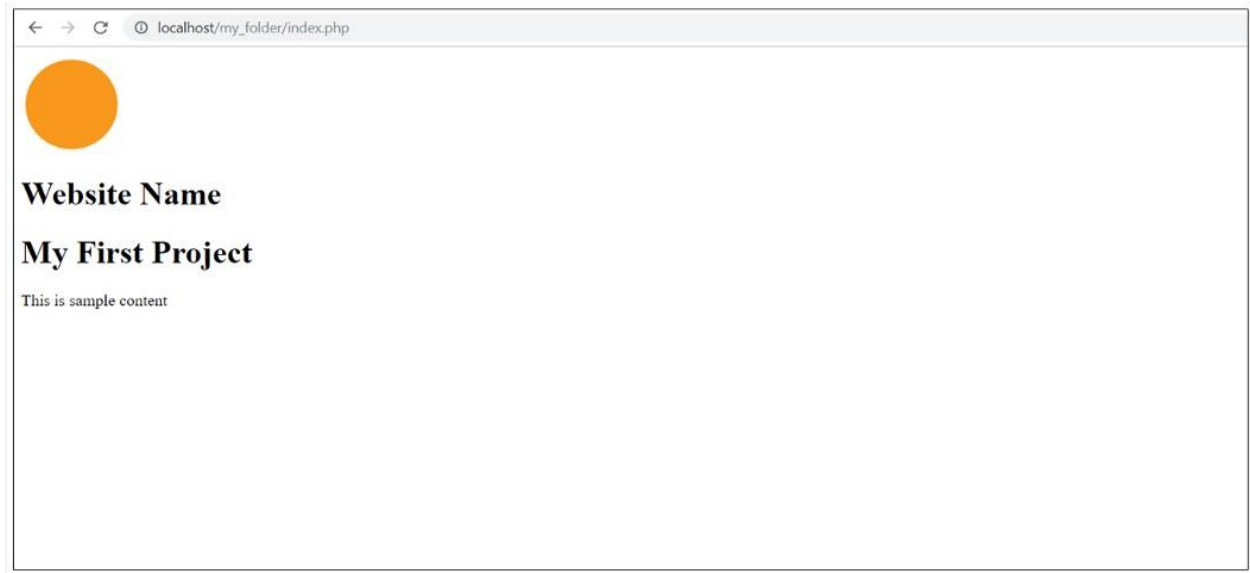
Technical Notes: As of this writing, there are better and safer ways, even more efficient ways to write SQL Statements to read MySQL database values. One can use PDO Query to access database items. The current documentation is modified work, that we are happy and privileged to be able to use for teaching purposes. We will pick up a sample PDO Query to show you how you can convert below `mysqli_query` to PDO.

Moving on! Remove the content on the `<h1>` and `<p>` tags and put this new code inside the section...

```
1  <?php include('include/config.php'); ?>
2  <html>
3      <head>
4          <title>My First Project</title>
5          <link href="css/main.css" rel="stylesheet" type="text/css">
6      </head>
7      <body>
8          <!-- HEADER -->
9          <?php include('templates/header.php'); ?>
10
11         <!-- MENU -->
12         <?php include('templates/menu.php'); ?>
13
14         <!-- CONTENT -->
15         <?php
16             $query = mysqli_query($sql, "SELECT * FROM tbl_articles");
17             while($row = mysqli_fetch_assoc($query))
18             {
19                 $title = $row['title'];
20                 $content = $row['content'];
21             }?>
22         <h1><?php echo $title; ?></h1>
23         <p><?php echo $content; ?></p>
24
25         <!-- FOOTER -->
26         <?php include('templates/footer.php'); ?>
27     </body>
28 </html>
```

As you can see above, the query is selecting all records from the database, which in this case is minimal. Since we only made one entry to the back-end. Let us see what that will look like in the browser.

Go to your browser, then open the URL http://localhost/my_folder/index.php



Bingo! Values from the MySQL Database loaded straight away in the browser...

Courtesy: <https://medium.com/@sapphdek/learn-how-to-build-a-website-using-html-css-php-mysql-af385524a5d6>

Alternative:

We discussed an option to querying MySQL databases that is safer and widely used currently. Get a look at this PDO query and compare it against the above mysqli_query. This also means that the database connection would need to change to conform with PDO requirements.

Here is what the PDO Query database connections look like:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username,
$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

Now, have a look an actual SQL statement:

```
<?php

$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach(new TableRows(new RecursiveArrayIterator($stmt-
>fetchAll())) as $k=>$v) {
        echo $v;
    }
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

Courtesy: https://www.w3schools.com/php/php_mysql_select.asp

Modified: 2021.10.05.11.16.PM
Dököll Solutions, Inc.