# 

# ACKNOWLEDGEMENT

I would like to express my deepest gratitude to **Vishal Lunagariya** sir for providing me with the opportunity to undertake my PHP internship at **Genesis Infotech**. Their guidance, support, and mentorship have been invaluable throughout this learning journey. I am also thankful to the entire **Genesis Infotech** team for their assistance and encouragement during my internship.

Their willingness to share knowledge and expertise has enhanced my understanding of PHP development and contributed significantly to my professional growth. Furthermore, I extend my appreciation to my colleagues who have collaborated with me on various projects. Their teamwork and constructive feedback have played a crucial role in my development as a PHP developer.

I am indebted to other team for their assistance with technical issues and for creating a conducive learning environment.

Finally, I am grateful to the entire PHP community for their invaluable resources, forums, and tutorials, which have aided me in expanding my knowledge and skills in PHP programming. This internship has been an enriching experience, and I am thankful to everyone who has contributed to my growth as a PHP developer.

Sincerely,

[Hemal Agravat]

# 

# 

# ABSTRACT

The PHP web development internship is designed to provide participants with comprehensive training in web development using PHP, MySQL, HTML, CSS, JavaScript, and essential tools like phpMyAdmin.

The internship aims to equip participants with practical skills in PHP programming, database management, and frontend development. Through projects and exercises, interns will gain proficiency in writing PHP code, designing and querying databases, and creating dynamic web pages using HTML, CSS, and JavaScript.

By the end of this internship, participants will have gained practical experience and a solid foundation in web development with PHP, MySQL, HTML, CSS, JavaScript, and associated tools. They will be well-equipped to pursue further studies or embark on a career in web development, armed with the skills and knowledge necessary to succeed in today's dynamic digital landscape.

# List of Tables

[ACKNOWLEDGEMENT 1](#_Toc162221542)

[ABSTRACT 2](#_Toc162221543)

[List of Tables 3](#_Toc162221544)

[CHAPTER: 1 OVERVIEW OF INDUSTRY 5](#_Toc162221545)

[1.1 Introduction to HTML 5](#_Toc162221546)

[1.2 History of HTML 6](#_Toc162221547)

[1.3 HTML Tags 6](#_Toc162221547)

[CHAPTER 2: Introduction of CSS 8](#_Toc162221549)

[2.1 Introduction of CSS 8](#_Toc162221550)

[2.2 Basic guide CSS 8](#_Toc162221551)

[2.3 Type of CSS 9](#_Toc162221552)

[CHAPTER 3: INTRODUCTION TO JS 12](#_Toc162221553)

[3.1 What is Java Script 12](#_Toc162221554)

[3.2 Introduction of Java Script 12](#_Toc162221555)

[3.3 Overview of Java Script 13](#_Toc162221556)

[CHAPTER 4: INTRODUCTION TO PHP 14](#_Toc162221557)

[4.1 What is PHP 14](#_Toc162221558)

[4.2 introduction to PHP 14](#_Toc162221560)

[4.3 Key Features of PHP 15](#_Toc162221560)

[4.4 pros and cons of PHP 16](#_Toc162221566)

[CHAPTER 5: INTRODUCTION TO My Sql 17](#_Toc162221567)

[5.1 Introduction to My Sql 17](#_Toc162221568)

[5.2 RDBM 17](#_Toc162221569)

[5.3 Features 17](#_Toc162221570)

[5.4 SQL 18](#_Toc162221571)

[5.5 Client-Server Architecture 18](#_Toc162221572)

[5.6 Compatibility and Platforms 18](#_Toc162221573)

[CHAPTER 6: How to Use XAMPP 19](#_Toc162221574)

[6.1 Introduction of Xampp 19](#_Toc162221575)

[6.2 What is Xampp 19](#_Toc162221576)

[6.3 Installing XAMPP 20](#_Toc162221577)

[Chapter 7: DOMAIN 29](#_Toc162221578)

[7.1 Definition Of Domain: 29](#_Toc162221579)

[7.2 Create Home Page: 30](#_Toc162221580)

[7.3 Create Regestration Page: 32](#_Toc162221581)

[7.4 Create Login Page: 34](#_Toc162221582)

[Chapter 8: Database: 37](#_Toc162221583)

[8.1 Introduction to Database: 37](#_Toc162221584)

[8.2 Create Connection php file to Database: 38](#_Toc162221585)

[8.3 Create Connection php file to Database: 39](#_Toc162221586)

[Create Table in To Database Query: 39](#_Toc162221587)

[Auto Increment Table Query: 39](#_Toc162221588)

[CONCLUSION 40](#_Toc162221589)

# CHAPTER: 1 OVERVIEW OF INDUSTRY

## 1.1 Introduction to HTML

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create web pages. It provides the structure and content of a webpage by using a system of tags and attributes. HTML documents are comprised of a series of elements, each enclosed by tags that define their purpose and formatting.

HTML is a very evolving markup language and has evolved with various versions updating. Long before its revised standards and specifications are carried in, each version has allowed its user to create web pages in a much easier and prettier way and make sites very efficient.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of content** | **HTML 1.2** | **HTML 4.01** | **HTML5** | **Purpose** |
| Heading | Yes | Yes | Yes | Organize page content by adding headings and subheadings to the top of each section of the page |
| Paragraph | Yes | Yes | Yes | Identify paragraphs of text |
| Address | Yes | Yes | Yes | Identify a block of text that contains contact information |
| Anchor | Yes | Yes | Yes | Link to other web content |
| List | Yes | Yes | Yes | Organize items into a list |
| Image | Yes | Yes | Yes | Embed a photograph or drawing into a web page |
| Table | No | Yes | Yes | Organize data into rows and columns |
| Style | No | Yes | Yes | Add CSS to control how objects on a web page are presented |
| Script | No | Yes | Yes | Add Javascript to make pages respond to user behaviors (more interactive) |
| Audio | No | No | Yes | Add audio to a web page with a single tag |
| Video | No | No | Yes | Add video to a web page with a single tag |
| Canvas | No | No | Yes | Add an invisible drawing pad to a web page, on which you can add drawings (animations, games, and other interactive features) using Javascript. |

## 1.2 History of HTML

HTML 1.0 was released in 1993 with the intention of sharing information that can be readable and accessible via web browsers. But not many of the developers were involved in creating websites. So the language was also not growing.

Then comes HTML 2.0, published in 1995, which contains all the features of HTML 1.0 along with a few additional features, which remained the standard markup language for designing and creating websites until January 1997 and refined various core features of HTML.

Then comes HTML 3.0, where Dave Raggett introduced a fresh paper or draft on HTML. It included improved new features of HTML, giving more powerful characteristics for webmasters in designing web pages. But these powerful features of the new HTML slowed down the browser in applying further improvements.

Then comes HTML 4.01, which is widely used and was a successful version of HTML before HTML 5.0, which is currently released and used worldwide. HTML 5 can be said for an extended version of HTML 4.01, which was published in the year 2012.

**1.3 HTML Tags**

Here's a brief overview of the key components and structure of HTML:

**Tags:** Tags are the fundamental building blocks of HTML. They consist of angled brackets < > enclosing a keyword that describes the element. Tags usually come in pairs: an opening tag and a closing tag. The opening tag marks the beginning of an element, while the closing tag marks its end. For example: **<p>This is a paragraph.</p>**

**Attributes:** Tags can also contain attributes, which provide additional information about an element. Attributes are specified within the opening tag and are written as name-value pairs. For example: **<a href="https://www.example.com">Visit Example</a>**

## In this <a> (anchor) tag, href is an attribute that specifies the URL the link points to.

**Elements:** Elements are composed of tags and the content they enclose. They represent the structure and semantics of a document. Some common HTML elements include headings (<h1> to <h6>), paragraphs (<p>), lists (<ul>, <ol>, <li>), links (<a>), images (<img>), and more.

**Document Structure:** An HTML document typically begins with a <!DOCTYPE html> declaration, which specifies the document type and version of HTML being used. This is followed by an <html> element, which serves as the root element of the document. The <html> element contains two main sections: <head> and <body>. The <head> section includes meta-information about the document, such as the title, character encoding, and links to external resources like stylesheets and scripts. The <body> section contains the actual content of the webpage.

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**Comments:** Comments in HTML are enclosed within <!-- and -->. They are not displayed on the webpage and are used for adding notes or explanations within the code.

**<!-- This is a comment -->**

HTML is a versatile and foundational language for web development, serving as the backbone for creating structured and interactive web pages on the Internet. It works in conjunction with other technologies such as CSS (Cascading Style Sheets) for styling and JavaScript for interactivity to create dynamic and visually appealing websites.

.

# CHAPTER 2: Introduction of CSS

## 2.1 Introduction of CSS

CSS stands for Cascading Style Sheets

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS saves a lot of work. It can control the layout of multiple web pages all at once

External stylesheets are stored in CSS files.

CSS (Cascading Style Sheets) is used to style and lay out web pages for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.

This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

## 2.2 Basic guide CSS

**Understand the Basics:** Familiarize yourself with the fundamental concepts of CSS, such as selectors, properties, and values. Learn about the box model, which defines how elements are rendered in CSS, including their padding, borders, margins, and content area.

**Create or Identify HTML Structure:** Start with an HTML document or create one that represents the structure of your webpage. Identify the elements you want to style using CSS. Each HTML element can be targeted and styled using CSS selectors.

**Link CSS File to HTML:** Create a new CSS file or use an existing one to write your styles. Link the CSS file to your HTML document using the <link> element within the <head> section. Example: **<link rel="stylesheet" type="text/css" href="styles.css">**

**Select and Style Elements:** Use CSS selectors to target HTML elements you want to style. Apply styling properties and values to those elements to control their appearance. Experiment with various CSS properties to achieve the desired visual effects.

Example:

/\* Select the <h1> element and change its color and font size \*/

h1 {

color: blue;

font-size: 24px;

}

**Test and Iterate:** Preview your webpage in a web browser to see how the styles are applied. Make adjustments to your CSS code as needed to refine the appearance of your webpage. Test your webpage across different browsers and devices to ensure consistent rendering.

**Learn Advanced Techniques:** Explore more advanced CSS features and techniques, such as: Flexbox and Grid Layout for advanced layout control. CSS animations and transitions for adding dynamic effects. CSS preprocessors like Sass or Less for enhanced productivity. CSS frameworks like Bootstrap or Tailwind CSS for rapid development.

**Stay Updated:** CSS is continuously evolving, with new features and specifications being introduced regularly. Stay updated with the latest trends, best practices, and browser support for CSS features.

## 2.3 Type of CSS

CSS can be categorized into three main types: inline CSS, internal CSS, and external CSS. Let's explore each type with examples:

**1.Inline CSS:** Inline CSS involves styling individual HTML elements directly within the HTML markup using the style attribute. This method is suitable for applying specific styles to individual elements but is not recommended for large-scale styling due to its lack of maintainability.

**Example:**

**<p style="color: blue; font-size: 16px;">This is a paragraph with inline CSS styling.</p>**

**2. Internal CSS:** Internal CSS is defined within the <style> element in the <head> section of an HTML document. Styles defined internally apply to all elements within the HTML document. This method is useful for smaller projects or when you want to keep styles confined to a single HTML document.

**Example:**

<!DOCTYPE html>

<html>

<head>

**<title>Internal CSS Example</title>**

**<style>**

**p {**

**color: blue;**

**font-size: 16px;**

**}**

**</style>**

</head>

<body>

<p>This is a paragraph with internal CSS styling.</p>

</body>

</html>

**3. External CSS:** External CSS involves defining styles in a separate CSS file and linking it to HTML documents using the <link> element. Styles defined externally can be reused across multiple HTML documents, enhancing maintainability and consistency. This method is commonly used in larger projects and promotes a more organized and modular approach to styling.

**Example:Index.Html FIle**

<!-- HTML document: index.html -->

<!DOCTYPE html>

<html>

<head>

**<title>External CSS Example</title>**

**<link rel="stylesheet" type="text/css" href="styles.css">**

</head>

<body>

<p>This is a paragraph with external CSS styling.</p>

</body>

</html>

**CSS file: styles.css File**

p {

color: blue;

font-size: 16px;

}

Each type of CSS has its advantages and use cases, and the choice depends on factors like project size, maintainability, and reusability of styles.

# CHAPTER 3: INTRODUCTION TO JS

## 3.1 What is Java Script

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. Web browsers have a dedicated JavaScript engine that executes the client code.

JavaScript is a scripting language for creating dynamic web page content. It creates elements for improving site visitors' interaction with web pages, such as dropdown menus, animated graphics, and dynamic background colors.

## 3.2 Introduction of Java Script

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.).

There are also more advanced server side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as realtime collaboration between multiple computers).

Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.

Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects.

## 3.3 Overview of Java Script

**1.Client-Side Scripting:** JavaScript is primarily a client-side scripting language, meaning it runs on the user's web browser rather than on a web server. This allows JavaScript to interact with the HTML and CSS of a webpage, making it dynamic and responsive.

**2.Features:** Event Handling: JavaScript can respond to user actions such as clicks, mouse movements, keyboard input, and form submissions.

**DOM Manipulation:** JavaScript can manipulate the Document Object Model (DOM), which represents the structure and content of an HTML document. This allows developers to dynamically change the content and style of web pages.

**Asynchronous Programming:** JavaScript supports asynchronous programming, allowing tasks to run independently without blocking the execution of other code. This is commonly used for handling AJAX requests, timers, and events.

**Cross-Browser Compatibility:** JavaScript is supported by all modern web browsers, making it a versatile language for web development.

**Libraries and Frameworks:** JavaScript has a vast ecosystem of libraries and frameworks like jQuery, React, Angular, and Vue.js, which provide additional functionalities and simplify the development of complex web applications.

**3.Server-Side Development:** While JavaScript is primarily used for client-side scripting, it can also be used for server-side development with platforms like Node.js. Node.js allows developers to use JavaScript to build scalable and high-performance server-side applications.

4.**Security Considerations:** Since JavaScript executes on the client-side, it's important to consider security implications such as cross-site scripting (XSS) vulnerabilities. Developers need to validate and sanitize user input to prevent malicious code injection.

Overall, JavaScript plays a crucial role in modern web development, enabling developers to create dynamic and interactive web applications that provide a rich user experience. Its versatility and widespread adoption make it an essential skill for web developers.

# CHAPTER 4: INTRODUCTION TO PHP

## 4.1 What is PHP

PHP is an acronym for "PHP: Hypertext Preprocessor".

PHP is a widely-used, open source scripting language.

PHP scripts are executed on the server.

PHP is free to download and use.

## What is a PHP File?

PHP files can contain text, HTML, CSS, JavaScript, and PHP code

PHP code is executed on the server, and the result is returned to the browser as plain HTML

PHP files have extension ".php"

## 4.2 introduction to PHP

PHP, which stands for Hypertext Preprocessor, is a server-side scripting language primarily used for web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994 but has since evolved into a powerful and widely-used language for building dynamic websites and web applications.

## What Can PHP Do?

## PHP can generate dynamic page content.

PHP can create, open, read, write, delete, and close files on the server.

PHP can collect form data.

PHP can send and receive cookies.

PHP can add, delete, modify data in your database.

PHP can be used to control user-access.

With PHP you are not limited to output HTML. You can output images or PDF files. You can also output any text, such as XHTML and XML.

## Why PHP?

PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)

PHP is compatible with almost all servers used today (Apache, IIS, etc.)

PHP supports a wide range of databases

PHP is free. Download it from the official PHP resource: [www.php.net](http://www.php.net/).

PHP is easy to learn and runs efficiently on the server side.

## What's new in PHP 7

PHP 7 is much faster than the previous popular stable release (PHP 5.6)

PHP 7 has improved Error Handling

PHP 7 supports stricter Type Declarations for function arguments

PHP 7 supports new operators (like the spaceship operator: <=>)

## 4.3 key features and aspects of PHP

**1.Server-Side Scripting:** PHP is executed on the server side, meaning that the PHP code is processed by the web server before the resulting HTML is sent to the client's browser. This allows PHP to dynamically generate web pages, interact with databases, and perform other server-side tasks.

**2,Embedded within HTML:** PHP code is typically embedded within HTML markup using special delimiters (<?php and ?>). This allows developers to seamlessly mix PHP code with HTML, making it easy to generate dynamic content and perform server-side tasks.

**3.Database Interaction:** PHP provides built-in support for interacting with databases, including MySQL, PostgreSQL, SQLite, and others. This allows developers to perform tasks such as querying databases, inserting data, updating records, and generating dynamic content based on database results.

**4.Versatility:** PHP is a versatile language that can be used for various purposes beyond web development. It can be used for command-line scripting, server-side scripting, and even desktop application development.

**5.Open Source and Cross-Platform:** PHP is an open-source language, meaning that it is freely available and can be used, modified, and distributed by anyone. It is also cross-platform, running on various operating systems such as Windows, macOS, Linux, and Unix.

## 4.4 pros and cons of PHP

**Pros**

1. The most important advantage of PHP is that it’s open-source and free from cost. It can be downloaded anywhere and is readily available to use for events or web applications.
2. It is platform-independent. PHP-based applications can run on any OS like UNIX, Linux, Windows, etc.
3. Applications can easily be loaded which are based on PHP and connected to the database. It’s mainly used due to its faster rate of loading over slow internet speed than other programming language.
4. It has less learning curve because it is simple and straightforward to use.Someone familiar with C programming can easily work on PHP.
5. It is more stable for a few years with the assistance of providing continuous support to various versions.

**Cons**

1. It is not that secure due to its open-source, because the ASCII text file is often easily available.
2. It is not suitable for giant content-based web applications.
3. It has a weak type, which can cause incorrect data and knowledge to users.
4. PHP frameworks got to learn to use PHP built-in functionalities to avoid writing additional code.
5. Using more features of PHP framework and tools cause poor performance of online applications.

# CHAPTER 5: INTRODUCTION TO My Sql

## 5.1 Introduction to My Sql

The full form of MYSQL is My Structured Query Language. It is a popular open-source relational database management system (RDBMS) that is used for storing and managing data.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed.

## 5.2 RDBM

**Relational Database Management System (RDBMS):**

MySQL is an RDBMS, which means it organizes data into tables with rows and columns, following the relational model. It allows for efficient storage, retrieval, and manipulation of data using SQL (Structured Query Language).

## 5.3 Features

1. Scalability: MySQL can handle large volumes of data and scale horizontally or vertically to accommodate growing needs.
2. Performance: It is optimized for speed and performance, making it suitable for high-traffic websites and applications.
3. Security: MySQL provides robust security features such as user authentication, access control, encryption, and auditing.
4. Replication: MySQL supports replication, allowing for the creation of backup copies of databases for redundancy and fault tolerance.
5. Transactions: It supports transactional processing, ensuring data integrity and consistency in multi-user environments.
6. Stored Procedures and Triggers: MySQL supports stored procedures and triggers, enabling developers to execute complex logic and automate tasks within the database.

## 5.4 SQL

**SQL (Structured Query Language):** MySQL uses SQL as its query language for interacting with the database. SQL allows users to perform various operations such as querying data, inserting, updating, and deleting records, creating and modifying tables, and defining relationships between tables.

## 5.5 Client-Server Architecture

**Client-Server Architecture:** MySQL follows a client-server architecture, where client applications communicate with the MySQL server to perform database operations. Clients can connect to the MySQL server using various protocols such as TCP/IP, Unix sockets, or named pipes.

## 5.6 Compatibility and Platforms

**Compatibility and Platforms:** MySQL is cross-platform and runs on various operating systems such as Windows, macOS, Linux, and Unix. It is compatible with many programming languages and platforms, making it a versatile choice for integrating with different software systems.

Overall, MySQL is a powerful and reliable database management system widely used in web development and other software applications. Its features, performance, and scalability make it a popular choice for building and managing databases in a wide range of industries.

# CHAPTER 6: How to Use XAMPP

## 6.1 Introduction of Xampp

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP is an open-source software package that provides a local web server environment for testing and development. It helps you test web applications locally before deployment, ensuring they function correctly on a live server.

## 6.2 What is Xampp

XAMPP stands for:

X – [cross-platform operating systems], which can work on any OS, Mac OX, Windows, Linux, etc.

A – Apache – this is the web server software.

M – MySQL – Database.

P – PHP

P – Perl – scripting language

**Software Components of XAMPP**

Apache The open-source web server Apache is the most widely used server worldwide for web content delivery. The server application is made public as free software by the Apache Software Foundation.

**MySQL/MariaDB**

In MySQL, XAMPP includes one of the world’s most famous relational database management systems. MySQL offers data storage for web services in combination with the web server Apache and the scripting language PHP. Current XAMPP versions have replaced MySQL with MariaDB.

**PHP**

The server-side programming language PHP allows users to make dynamic websites or applications. PHP can be installed on all platforms and supports several various database systems.

**Perl**

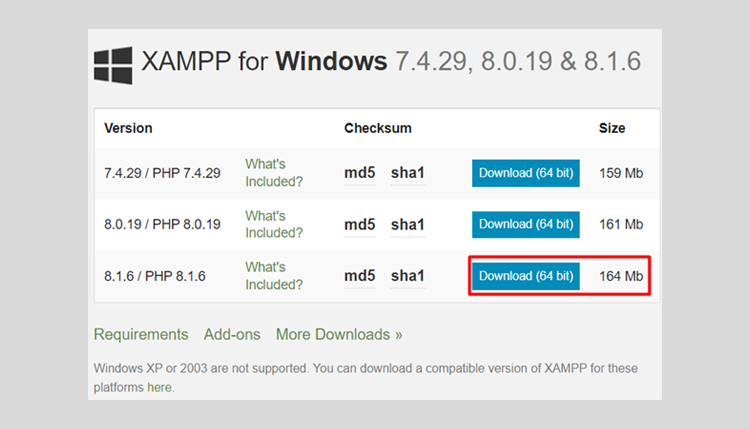
Perl is the scripting language in system administration, web development, and network programming. Like PHP, Perl also enables users to program dynamic web applications.

## 6.3 Installing XAMPP

**Step:1**

**XAMPP is a release made accessible by the non-profit project Apache Friends. There are different versions of PHP, you can install.**

**Click here to download XAMPP for https://www.apachefriends.org/de/download.html**

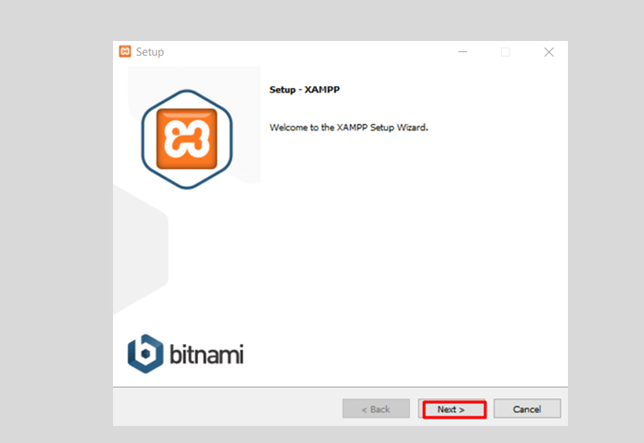


**Run .exe file**

When you download the software package, you can start the XAMPP installation. Double-click on the file ending with the .exe file.

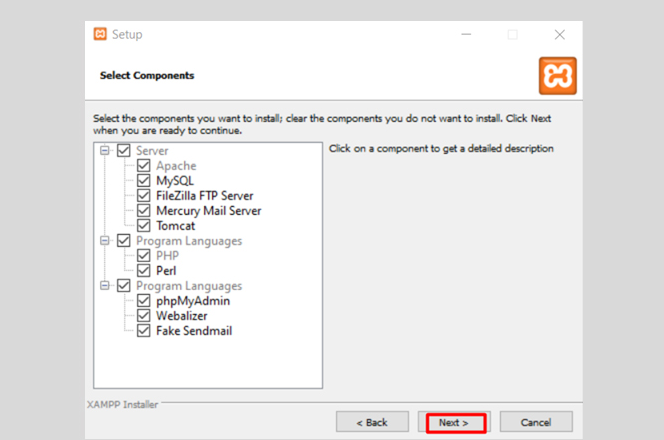
**Start the setup wizard**

After opening the .exe file, the start screen of the XAMPP setup wizard should arise automatically. Click on ‘Next’ to configure the installation settings.



**Select Software Components**

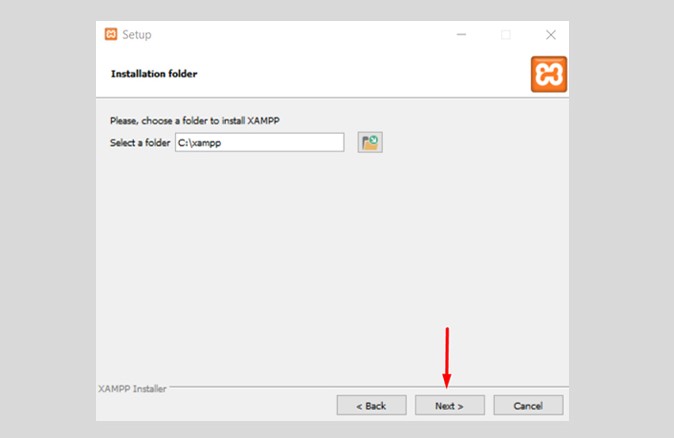
According to the selected components, you have the right to exclude individual components of the XAMPP software bundle from installation. But for a full local test server, we suggest you install using the standard setup and all available features. When you choose any option, click the ‘next’ button.



**Choose the installation folder**

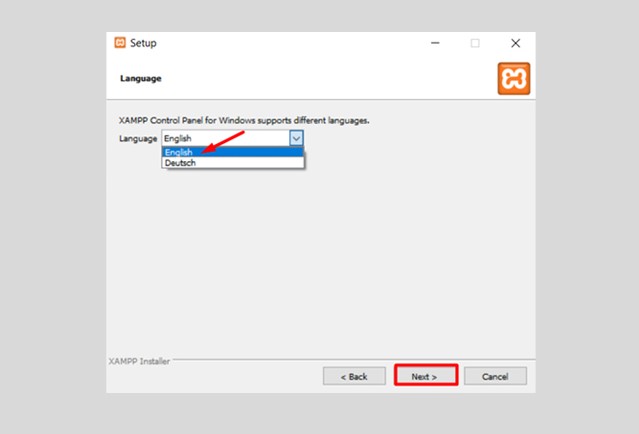
After selecting the software components, you can choose the installing folder. If XAMPP is already installed, then this message will appear on your screen.

But, if you are installing XAMPP for the first time, then the folder with the name XAMPP will be created under C:/ for you. After choosing an installing directory, click on the ‘next’ button.



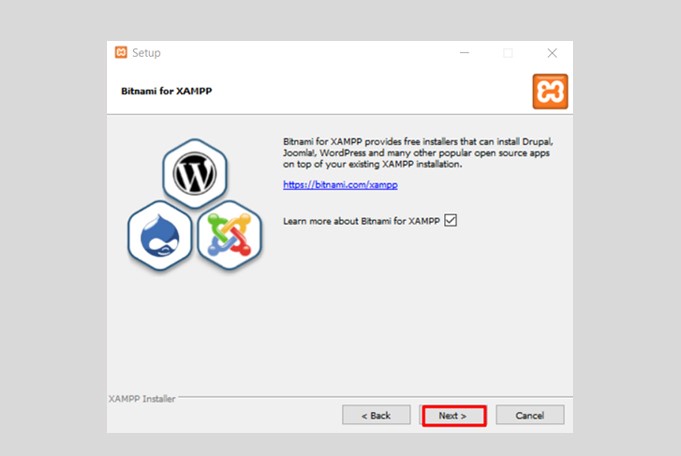
**Choose the language**

After choosing a folder to install XAMPP, you have the option to choose a language. XAMPP control panel for windows supports different languages. Choose your desired language and click on the ‘next’ button.



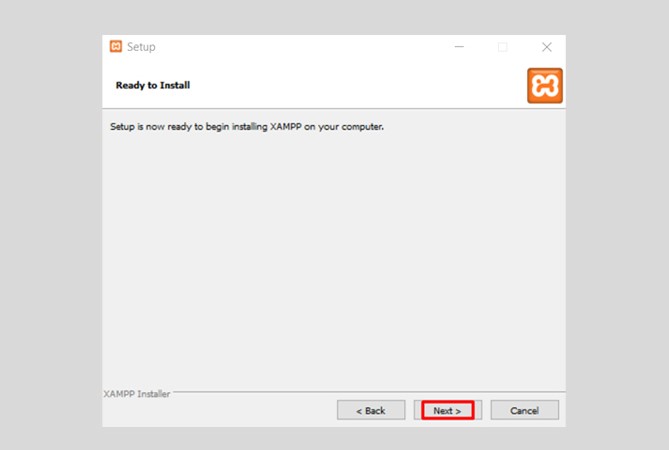
**Learn About Bitnami**

for XAMPP Bitnami for XAMPP provides free installers that can install Drupal, Joomla, WordPress, and many other famous open-source apps on top of your existing XAMPP installation. Click on the ‘next’ button.



**Ready To Install**

XAMPP setup is ready to install on your computers. Click the ‘next’ button.



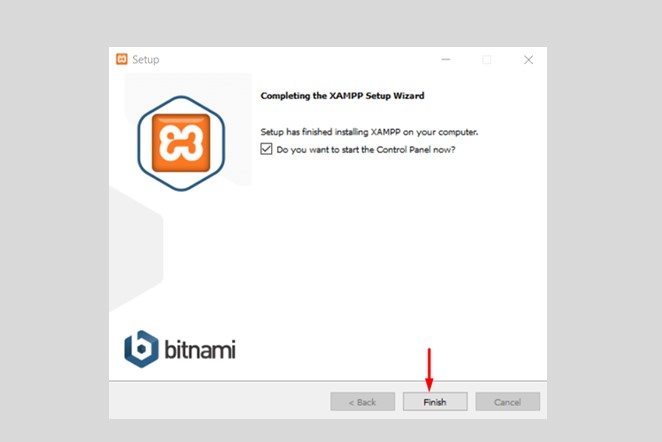
**Start the installation**

process Once all the above choices have been decided, click to begin the installation. The setup wizard will unpack and install the chosen components and save them to the specified directory. This process can take several minutes in total. You can observe the progress of this installation by keeping an eye on the green loading bar in the center of the screen.



**Complete XAMPP Setup**

Once all the components are unpacked and installed, you can close the setup wizard by clicking on the ‘finish’ button. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is completed.



**The XAMPP Control Panel**

Controls for the individual components of your test server can be reached through the XAMPP Control Panel.

The clear user interface logs all actions and lets you start or stop individual modules with a single.

The XAMPP Control Panel also offers you various other buttons

**including:**

**Config:** allows you to configure the XAMPP as well as the individual components

**Netstat:** displays all running processes on the local computer

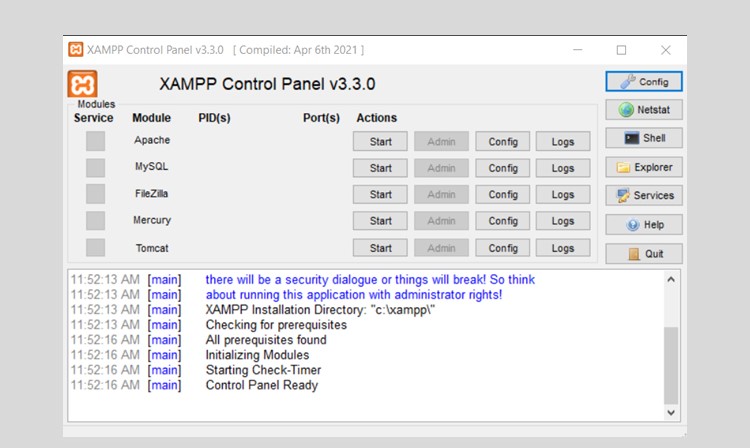
**Shell:** opens a UNIX shell

**Explorer:** opens the XAMPP folder in Windows Explorer

**Services:** shows all services currently running in the background

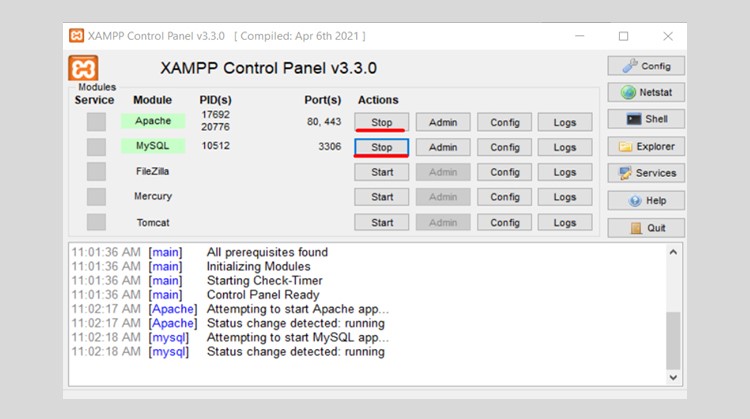
**Help:** offers links to user forums

**Quit:** closes the XAMPP Control Panel



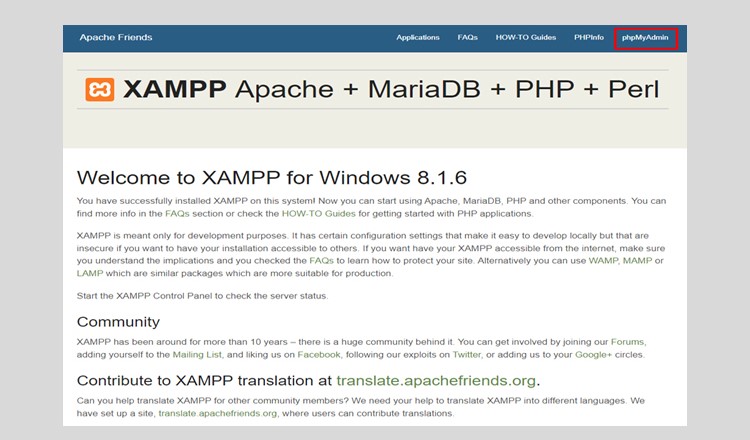
**Start the Modules**

You can start or stop the individual modules on the XAMPP cPanel via action buttons. You can also see the modules created because their names are highlighted. The green color shows that they are active modules.

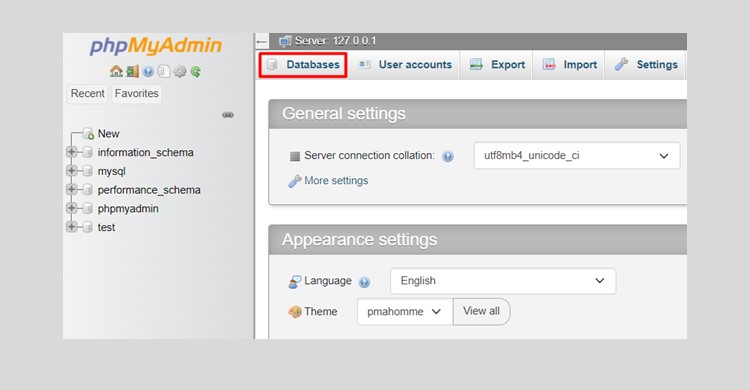


**Module administration**

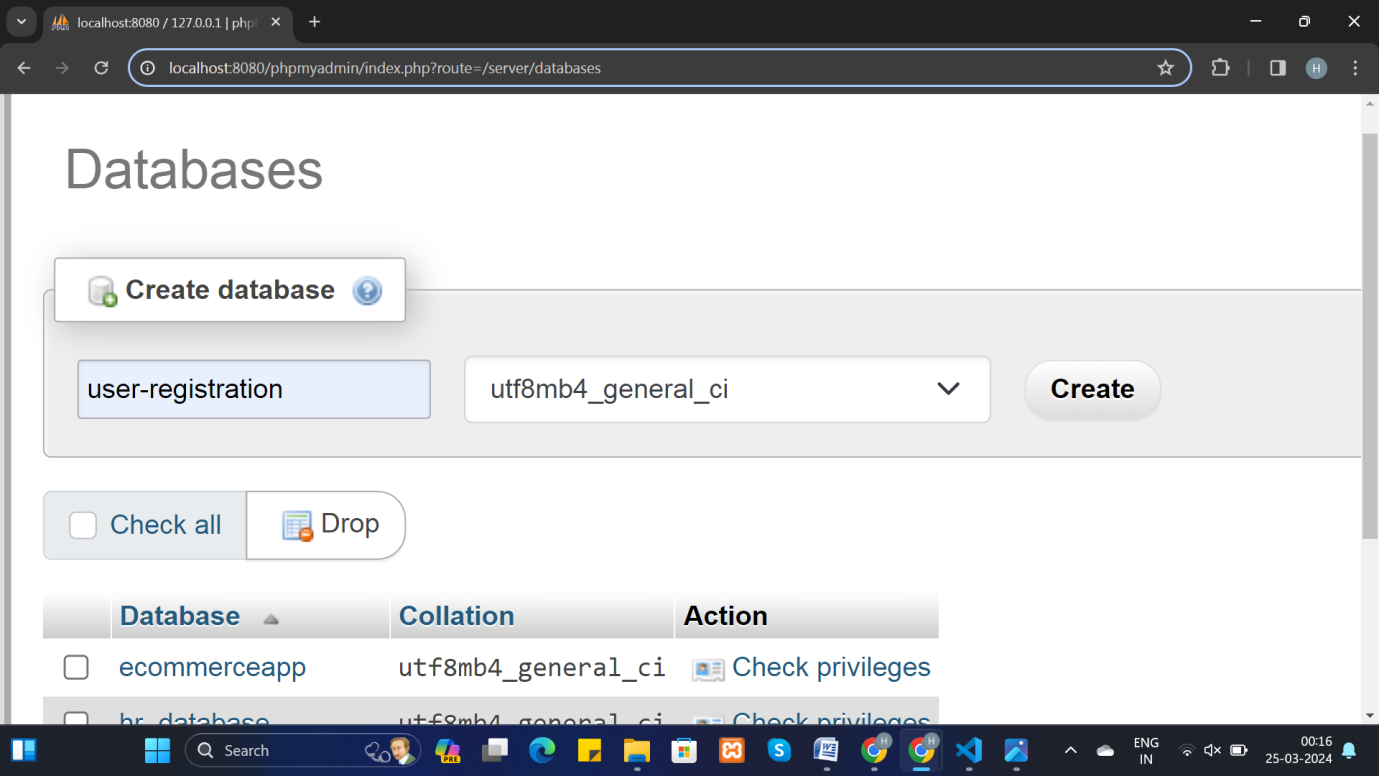
You have an ‘Admin’ option on the for every module in your XAMPP. Click on the Admin button of your Apache server to go to the web address of your web server.now start in your standard browser, and you will be led to the dashboard of your XAMPP’s local host. The dashboard features multiple links to websites for valuable information and the open source open-sourceNami, which shows you many other applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can get the dashboard through localhost/dashboard/.



Use the admin button of your database to open phpMyAdmin. You can also handle the databases of your web projects. You can also get access to the administration section of your MySQL database through localhost/PHPMyAdmin/.



Open the ‘databases’ and create a database with the name.



# Chapter 7: DOMAIN

## 7.1 Definition Of Domain:

Developing a static e-commerce website involves a similar process to developing a regular static website, but with additional considerations for managing products, and ensuring security.

To develop a static website, you'll need several tools to create, design, code, test, and deploy your website efficiently.

Here's a list of essential tools:

**Text Editor or Integrated Development Environment (IDE):** Choose a text editor or IDE for writing HTML, CSS, JavaScript, and other code. Popular options

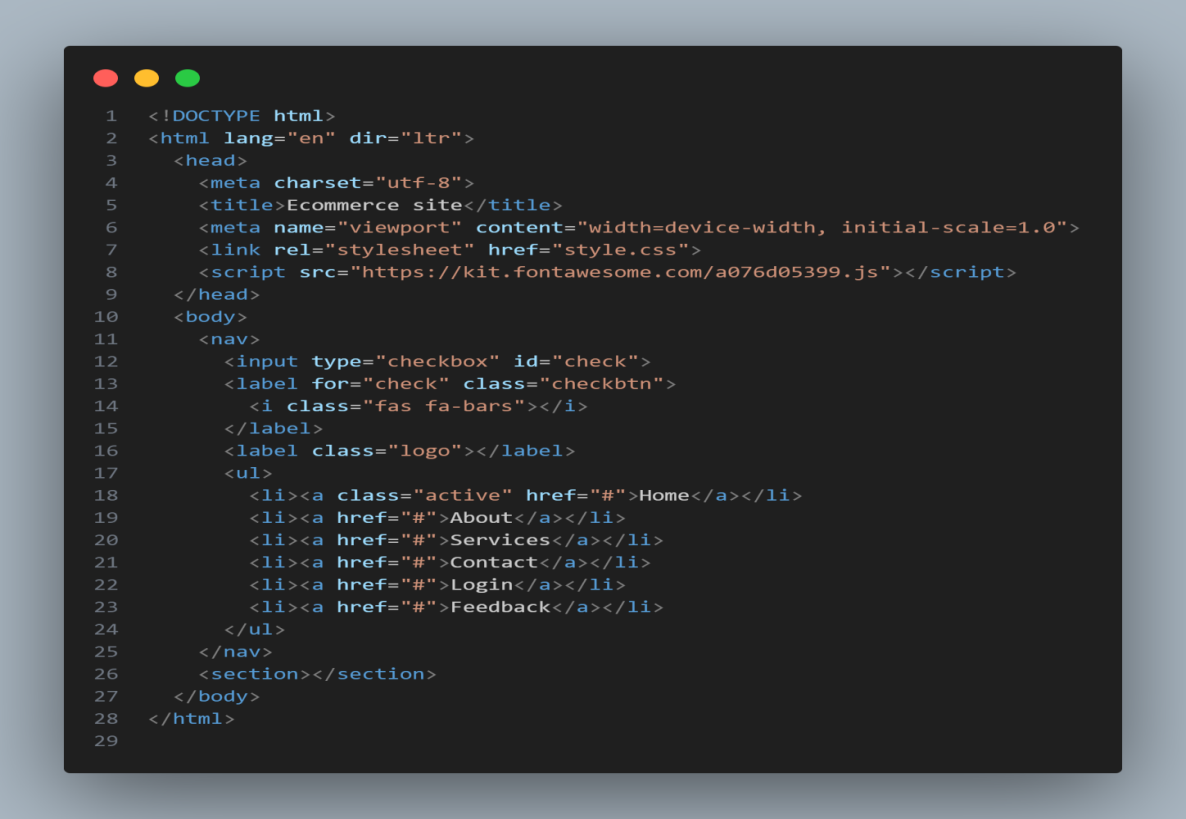
include: Visual Studio Code, Sublime Text ,Notepad++ ETC..

**Web Browser:** Use web browsers like Google Chrome, Mozilla Firefox, or Safari to preview and test your website during development.

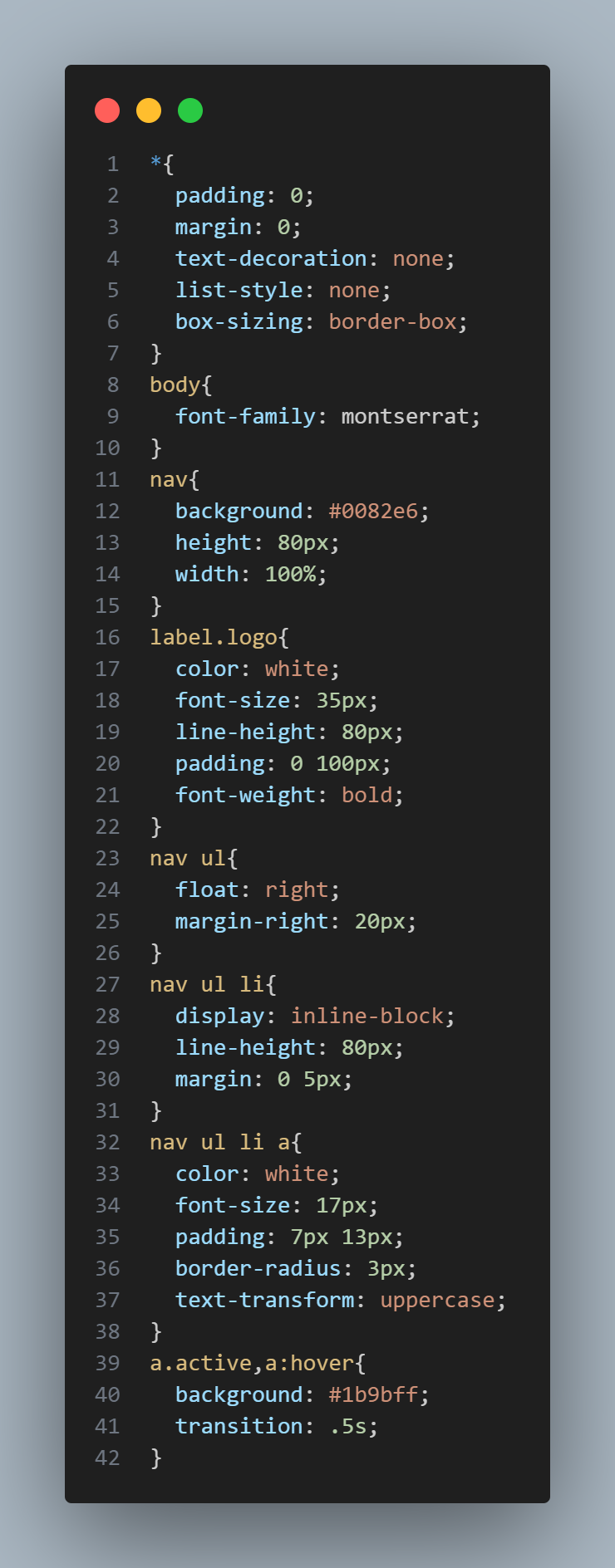
## 

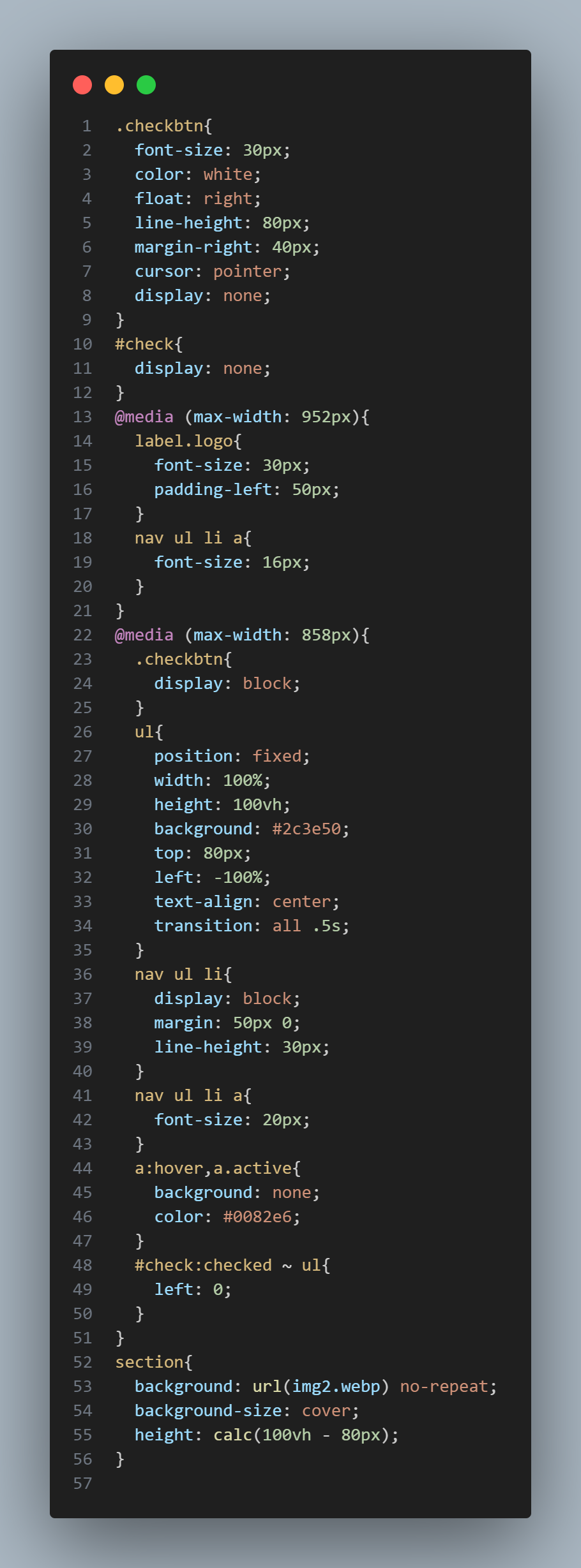
## 7.2 Create Home Page:

**Index.html**

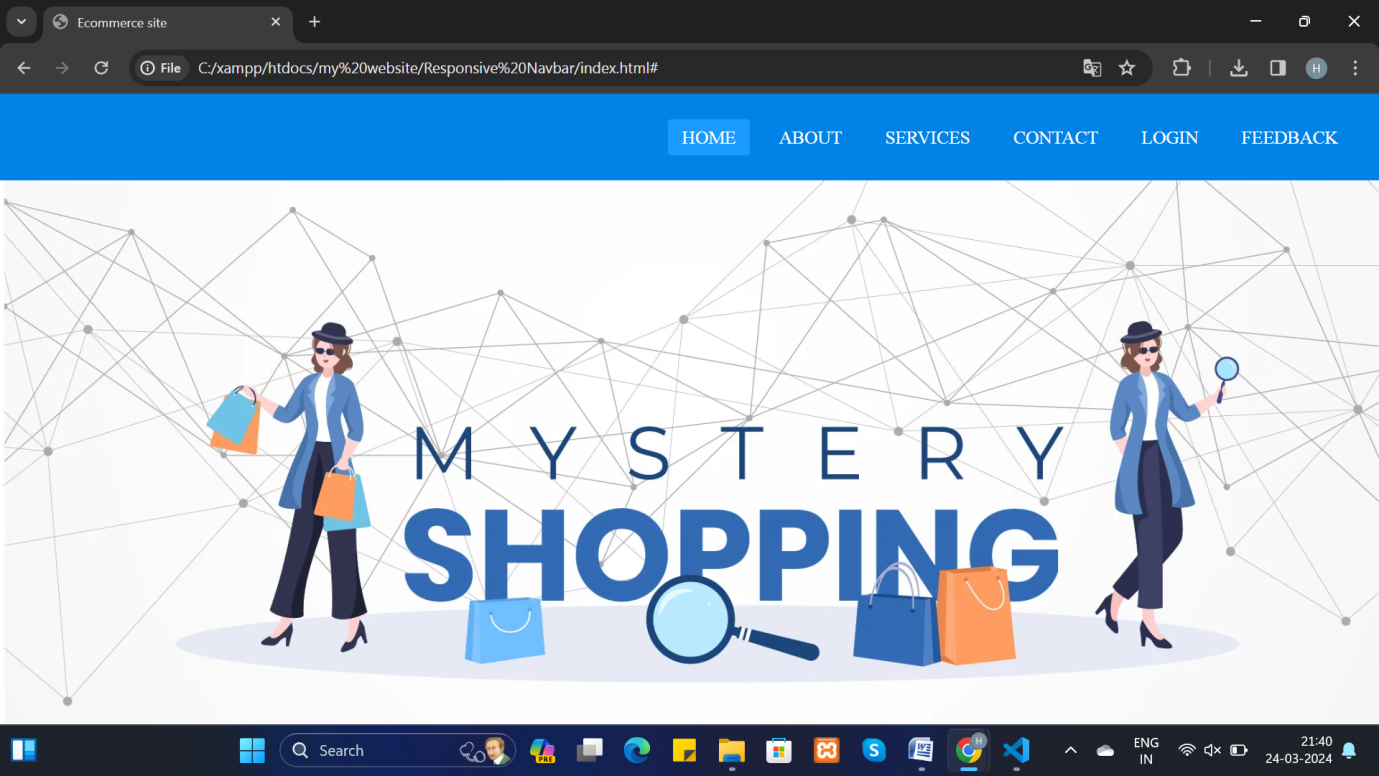


**Style.css**

****



**View of Home page**

****

## 7.3 Create Regestration Page:

This code defines a simple HTML form for user registration.

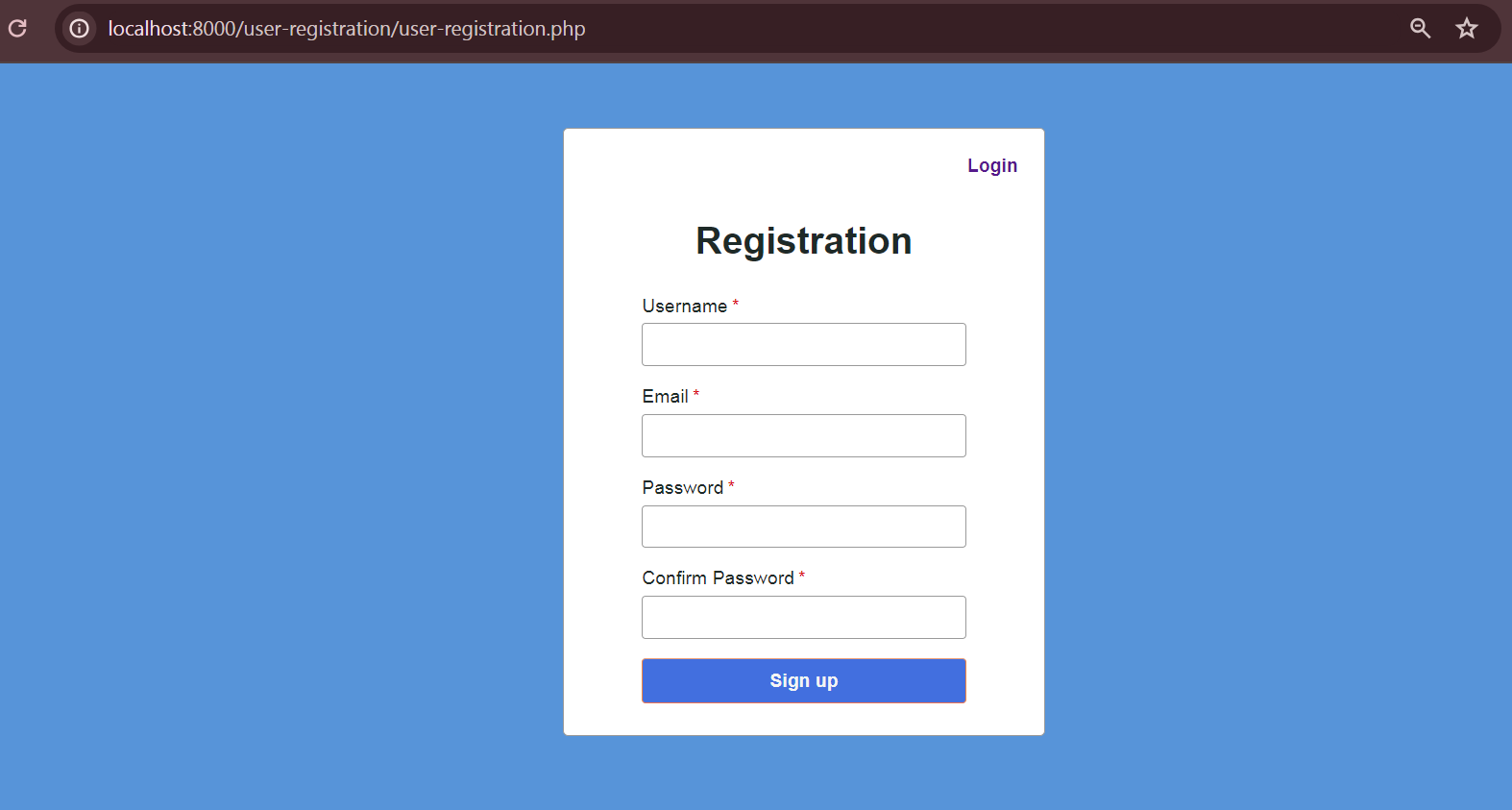
The form's action attribute is set to "register.php", meaning that when the form is submitted, the data will be sent to a PHP script named "register.php" for processing. The form uses the POST method to send data securely.

The required attribute in the input fields ensures that the fields cannot be left blank.

Basic CSS styling is applied for aesthetics.



**View of Regestration Page**

****

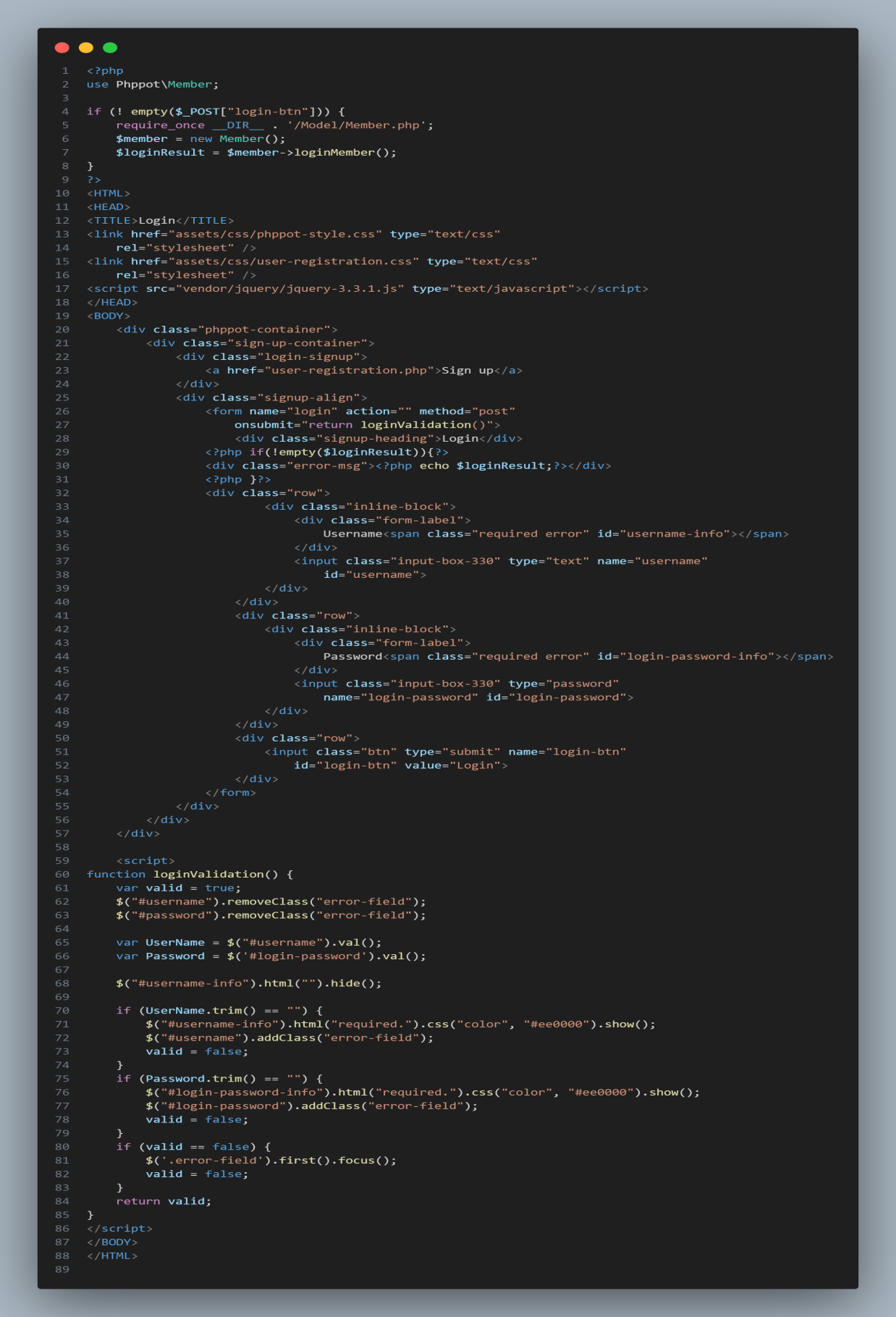
## 7.4 Create Login Page:

This code defines a simple HTML form that collects a username and password.

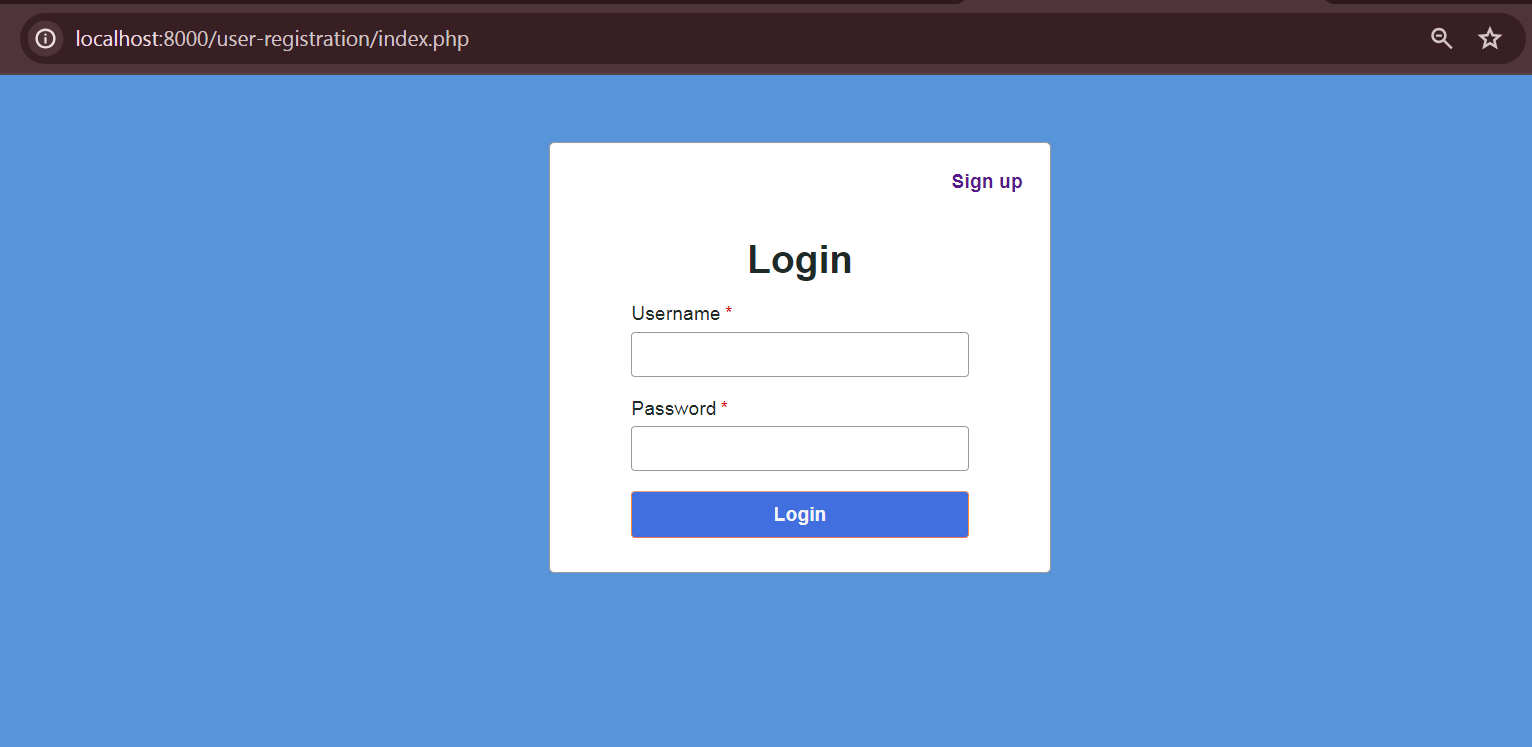
The form's action attribute is set to "login.php", meaning that when the form is submitted, the data will be sent to a PHP script named "login.php" for processing. The form uses the POST method to send data securely.

The required attribute in the input fields ensures that the fields cannot be left blank.

Basic CSS styling is applied for aesthetics.



**View Login Page**

****

# Chapter 8: Database:

## 8.1 Introduction to Database:

**What is a Database?**

A database is a structured collection of data. It provides an organized way to store, retrieve, and manage information. Databases are crucial for web applications to store and retrieve data efficiently.

**Types of Databases:**

**Relational Databases:** These databases organize data into tables with rows and columns. Examples include MySQL, PostgreSQL, SQLite.

**NoSQL Databases:** These databases store and retrieve data in a format other than the tabular relations used in relational databases. Examples include MongoDB, Couchbase.

**Using Databases in PHP:**

PHP provides built-in functions and extensions to interact with databases. The most commonly used extension for relational databases in PHP is the MySQLi (MySQL Improved) extension, which supports MySQL databases. Another popular option is PDO (PHP Data Objects), which is more versatile as it supports multiple database systems.

## 8.2 Create Connection php file to Database:



## 8.3 Create Connection php file to Database:

## Create Table in To Database Query:

CREATE TABLE `tbl\_member` (

`id` int(11) NOT NULL,

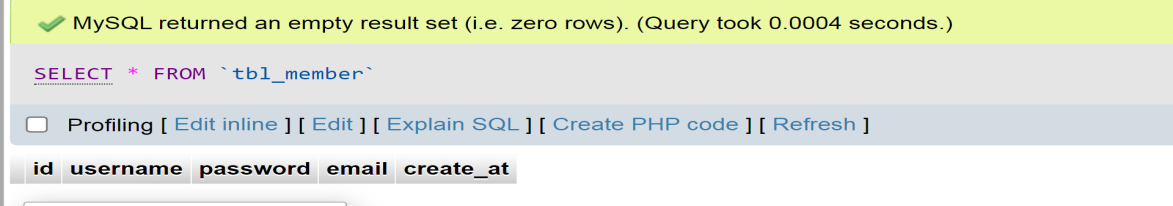
`username` varchar(255) NOT NULL,

`password` varchar(200) NOT NULL,

`email` varchar(255) NOT NULL,

`create\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=latin1;



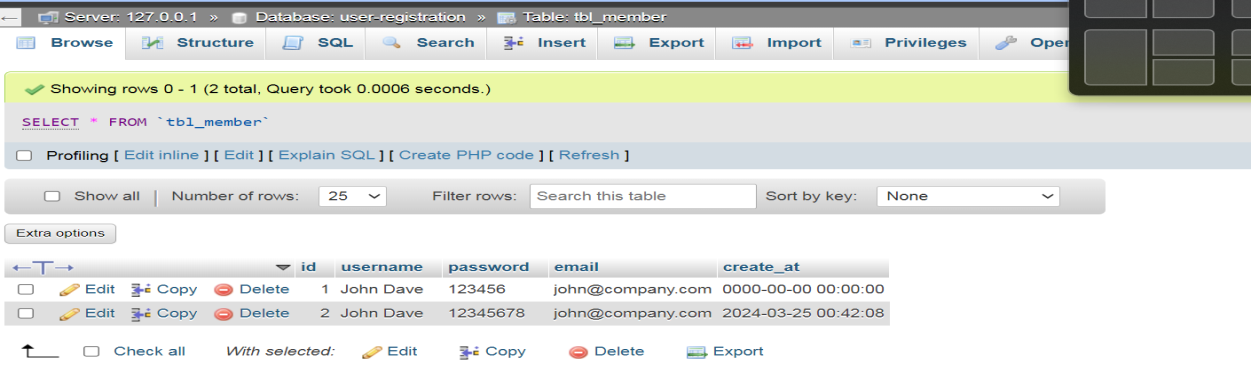
## Auto Increment Table Query:

ALTER TABLE ``tbl\_member `

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

**Insert Data into database Query**

INSERT INTO tbl\_member (username, password,email) VALUES ('John Dave', 12345678,'john@company.com');



# CONCLUSION

My internship experience has been immensely rewarding, particularly in the development of the static project assigned to me. Over the course of the internship, I have gained valuable insights into various aspects of web development and project management.

Through this project, I have enhanced my proficiency in HTML, CSS, and JavaScript. I have successfully implemented responsive design principles to ensure the website's compatibility across various devices and screen sizes.

While working on the project, I encountered several challenges such as cross-browser compatibility issues and optimizing page load speed. Through systematic troubleshooting and research, I was able to overcome these obstacles, thereby honing my problem-solving skills.

Collaborating with team members and receiving constructive feedback has been instrumental in refining the project. Regular communication ensured that everyone was aligned with project goals and timelines, facilitating smoother progress.

This internship has emphasized the importance of continuous learning in the rapidly evolving field of web development. I am committed to staying updated with the latest technologies and best practices to further enhance my skills and contribute meaningfully to future projects.

In conclusion, my internship experience has been incredibly enriching, and the development of the static project has been a pivotal aspect of my learning journey. I am grateful for the guidance and support provided by my mentors and colleagues throughout this experience. Moving forward, I am excited to apply the knowledge and skills gained during this internship to future endeavors in the field of web development.