Web Development and Database Administration Level III

Based on November 2023, Curriculum Version II

Module Title: Integrate Database with a website

Module Code: EIS WDDBA3 M08 0323

Nominal Duration: 80 Hours

Prepared by: Ministry of Labor and Skill

November, 2023
Addis Ababa, Ethiopia

Table of Contents

Acknowledgment

Ministry of Labor and Skills wish to extend thanks and appreciation to the many representatives of TVT instructors and respective industry experts who donated their time and expertise to the development of this Teaching, Training and Learning Materials (TTLM).

Acronym

CGI	Common Gateway Interface
CPU	Central Processing Unit
CSS	Cascade Style Sheet
DBMS	Database Management System
DML	Data Manipulation Language
DQL	Data Query Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Service
PHPH	Ypertext Preprocessor
RAM	Random Access Memory
SEO	Search Engine Optimization
SQL	Structured Query Language
UAC	User Access Control
URL	Uniform Resource Locator
WWW	World Wide Web

Introduction to the Module

This Module. We will see how databases can be connected to the Web, and the most effective way of using the new technology to develop database applications. We will also study the most commonly used approaches for creating Web databases, and discuss related issues such as dynamic updating of Web pages in line with the changes in databases, performance.

The reason is that the Object-oriented model is considered the most suitable for the storage, organization and retrieval of large sets of Web documents.

This module covers the units:

- website to database connection
- Retrieve data from database and display on web pages
- database user input manipulation

Learning Objective of the Module

- Connect website to database
- Retrieve data from database and display on web pages
- Update database data from user input

Module Instruction

For effective use this modules trainees are expected to follow the following module instruction:

1. Read the information written in each unit
2. Accomplish the Self-checks at the end of each unit
3. Perform Operation Sheets which were provided at the end of units
4. Do the “LAP test” giver at the end of each unit and

Unit One: website to database connection

This unit is developed to provide you the necessary information regarding the following content coverage and topics

- web servers, databases and server-side scripting languages
- User Identification and site data technical requirements.
- Web servers Installation.
- Connect database with web application

This unit will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- understand web servers, databases and server-side scripting languages
- Understand and identify user and technical data requirements
- Understand and install web servers.
- Understand and Connect database with web application.

1.1.Web servers, databases and server-side scripting languages

The WWW comprises software (e.g. Web servers and browsers) and data (e.g. Web sites). It simply represents a (huge) set of information resources and services that live on the Internet. Each Web site consists of a set of Web pages, which typically contain multimedia data (e.g. text, images, sound and video). In addition, a Web page can include hyperlinks to other Web pages which allow users (also called net surfers) to navigate through the Web of information pages.

- **There are two types of Web pages: static and dynamic.**

A. Static: An HTML document stored in a file is a typical example of a static Web page. Its contents do not change unless the file itself is changed.

B. Dynamic: For a dynamic Web page, its contents are generated each time it is accessed. As a result, a dynamic Web page can respond to user input from the browser by, for example, returning data requested by the completion of a form or returning the result of a database query. A dynamic page can also be customized by and for each user. Once a user has specified some preferences when accessing a particular site or page, the information can be recorded and appropriate responses can be generated according to those preferences.

From the above, it can be seen that dynamic Web pages are much more powerful and versatile than static Web pages, and will be a focus for developing Web database applications.

When the documents to be published are dynamic, such as those resulting from queries to databases, the appropriate hypertext needs to be generated by the servers. To achieve this, we must write scripts that perform conversions from different data formats into HTML 'on-the-fly'. These scripts also need to recognize and understand the queries performed by clients through HTML forms and the results generated by the DBMS.

In short, a Web database application normally interacts with an existing database, using the Web as a means of connection and having a Web browser or client program on the front end. Typically such applications use HTML forms for collecting user input (from the client); CGI (Common Gateway Interface, to be discussed later) to check and transfer the data from the server; and a script or program which is or calls a database client to submit or retrieve data from the database. The diagram below gives a graphical illustration of such a scenario. More will be discussed in later parts of this chapter.

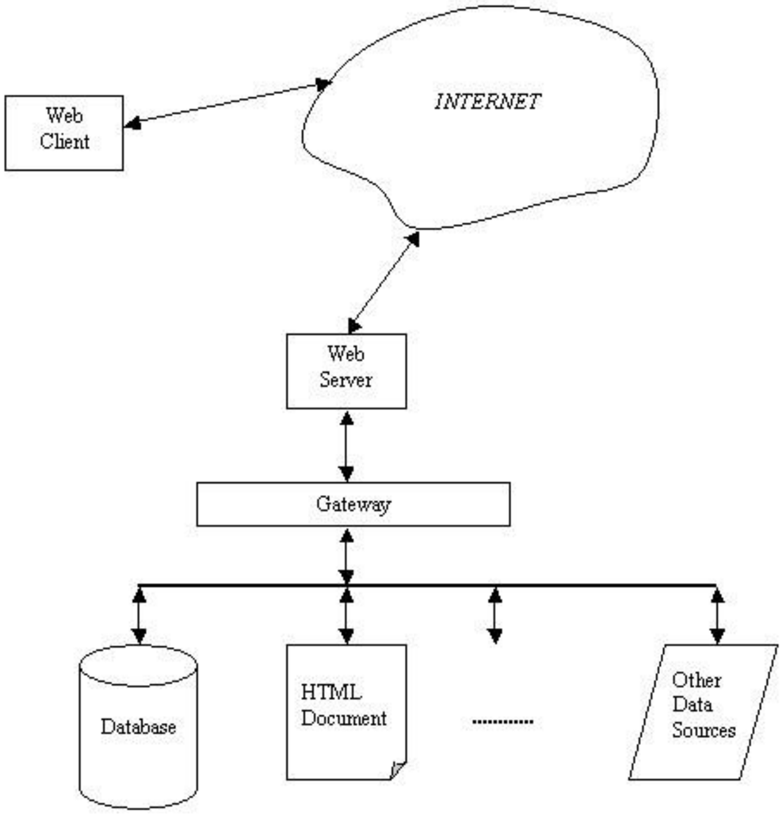


Fig 1.1 Common Gateway Interface

- **Components of a database application**

Web database applications may be created using various approaches. However, there are a number of components that will form essential building blocks for such applications. In other words, a Web database application should comprise the following four layers (i.e. components):

- ✓ Browser layer
- ✓ Application logic layer
- ✓ Database connection layer
- ✓ Database layer

- **Browser layer:**

The browser is the client of a Web database application, and it has two major functions. First, it handles the layout and display of HTML documents. Second, it executes the client-side extension functionality such as Java, JavaScript, and ActiveX (a method to extend a browser's capabilities).

The three most popular browsers at the present are Mozilla Firefox (Firefox for short), Google Chrome and Microsoft Edge.

All three browsers are graphical browsers. During the early days of the Web, a text-based browser, called Lynx, was popular. As loading graphics over the Internet can be a slow and time-consuming process, database performance may be affected. If an application requires a speedy client and does not need to display graphics, then the use of Lynx may be considered.

All browsers implement the HTML standard. The discussion of HTML is beyond this chapter, but you need to know that it is a language used to format data/documents to be displayed on the Web.

Browsers are also responsible for providing forms for the collection of user input, packaging the input, and sending it to the appropriate server for processing. For example, input can include registration for site access, guest books and requests for information. HTML, Java, JavaScript or ActiveX (for IE) may be used to implement forms.

- **Application logic layer**

The application logic layer is the part of a Web database application with which a developer will spend the most time. It is responsible for:

- ✓ Collecting data for a query (e.g. a SQL statement).
- ✓ Preparing and sending the query to the database via the database connection layer.
- ✓ Retrieving the results from the connection layer.
- ✓ Formatting the data for display.

Most of the application’s business rules and functionality will reside in this layer. Whereas the browser client displays data as well as forms for user input, the application logic component compiles the data to be displayed and processes user input as required. In other words, the application logic generates HTML that the browser renders. Also it receives processes and stores user input that the browser sends.

Depending on the implementation methods used for the database application, the application logic layer may have different security responsibilities. If the application uses HTML for the front end, the browser and server can handle data encryption (i.e. a security measure to ensure that data will not be able to be intercepted by unauthorized parties). If the application is a Java applet and uses Java for the front end, then it itself must be responsible for adopting transmission encryption.

- **Database connection layer**

This is the component which actually links a database to the Web server. Because manual Web database programming can be a daunting task, many current Web database building tools offer database connectivity solutions, and they are used to simplify the connection process.

The database connection layer provides a link between the application logic layer and the DBMS. Connection solutions come in many forms, such as DBMS net protocols, API (Application Programming Interface [see note below]) or class libraries, and programs that are themselves database clients. Some of these solutions resulted in tools being specifically designed for developing Web database applications. In Oracle, for example, there are native API libraries for connection and a number of tools, such as Web Publishing Assistant, for developing Oracle applications on the Web.

The connection layer within a Web database application must accomplish a number of goals. It has to provide access to the underlying database, and also needs to be easy to use, efficient, flexible, robust, reliable and secure. Different tools and methods fulfill these goals to different extents.

Note

An API consists of a set of interrelated subroutines that provide the functionality required to develop programs for a target operating environment. For example, Microsoft provides different APIs targeted at the construction of 16- and 32-bit

Windows applications. An API would provide functions for all aspects of system activity, such as memory, file and process management. Specialized APIs are also supplied by software vendors to support the use of their products, such as database and network management systems.

- ✓ **Database layer**

This is the place where the underlying database resides within the Web database application. As we have already learned, the database is responsible for storing, retrieving and updating data based on user requirements, and the DBMS can provide efficiency and security measures.

In many cases, when developing a Web database application, the underlying database has already been in existence. A major task, therefore, is to link the database to the Web (the connection layer) and to develop the application logic layer.

- **Database gateways**

A Web database gateway is a bridge between the Web and a DBMS, and its objective is to provide a Web-based application the ability to manipulate data stored in the database. Web database gateways link stateful systems (i.e. databases) with a stateless, connectionless protocol (i.e. HTTP). HTTP is a stateless protocol in the sense that each connection is closed once the server provides a response. Thus, a Web server will not normally keep any record about previous requests. This results in an important difference between a Web-based client-server application and a traditional client-server application:

- ✓ In a Web-based application, only one transaction can occur on a connection. In other words, the connection is created for a specific request from the client. Once the request has been satisfied, the connection is closed. Thus, every request involving access to the database will have to incur the overhead of making the connection.
- ✓ In a traditional application, multiple transactions can occur on the same connection. The overhead of making the connection will only occur once at the beginning of each database session.

There are a number of different ways to create Web database gateways. Generally, they can be grouped into two categories: client-side solutions and server side solutions,

- **Client-side solutions**

The client-side solutions include two types of approaches for connections: browser extensions and external applications.

Browser extensions are add-ons to the core Web browser that enhance and augment the browser’s original functionality. Specific methods include plug-ins for Firefox, Chrome and IE, and ActiveX controls for IE. Also, all the three types of browsers (Firefox, Chrome and IE) support Java and JavaScript languages (i.e. Java applets and JavaScript can be used to extend browsers’ capabilities).

External applications are helper applications or viewers. They are typically existing database clients that reside on the client machine and are launched by the Web browser in a particular Web application. Using external applications is a quick and easy way to bring legacy database applications online, but the resulting system is neither open nor portable. Legacy database clients do not take advantages of the platform independence and language independence available through many Web solutions. Legacy clients are resistant to change, meaning that any modification to the client program must be propagated via costly manual installations throughout the user base.

- **Server-side solutions**

Server-side solutions are more widely adopted than the client-side solutions. A main reason for this is that the Web database architecture requires the client to be as thin as possible. The Web server should not only host all the documents, but should also be responsible for dealing with all the requests from the client.

In general, the Web server should be responsible for the following:

- ✓ Listening for HTTP requests.
- ✓ Checking the validity of the request.
- ✓ Finding the requested resource.
- ✓ Requesting authentication if necessary.
- ✓ Delivering requested resource.
- ✓ Spawning programs if required.
- ✓ Passing variables to programs.
- ✓ Delivering output of programs to the requester.
- ✓ Displaying error message if necessary.

The client (browser) should be responsible for some of the following:

- ✓ Rendering HTML documents.
- ✓ Allowing users to navigate HTML links.
- ✓ Displaying image.
- ✓ Sending HTML form data to a URL.
- ✓ Interpreting Java applets.
- ✓ Executing plug-ins.
- ✓ Executing external helper applications.
- ✓ Interpreting JavaScript and other scripting language programs.
- ✓ Executing ActiveX controls in the case of IE.

1.2. User Identification and site data technical requirements.

The Data are collected based on the requirements of the website. Every website will have different requirements and the data are collected based on the same. Different ways to retrieve data for a website are via algorithms, data structure and complexity analysis. Gathering data depends on the website, for small websites it takes lesser time; it also depends upon the client requirements. Data should be collected based on some certain criteria

- ✓ Business Objective
- ✓ Website Design
- ✓ Website Features
- ✓ Website Layout
- ✓ SEO / Search Engine Optimization Strategy

Requirements are collected based on the objective what is this project about and how to meet the goal. The basic features and functionality helps in collecting the data required for the project. For example, this may have e-commerce capabilities, support multiple languages, or have a business directory listing. The next factor that helps in the data collection is deliverables this includes how wireframes and mock-ups/structural model will be delivered, timeframe/project plan (including project reporting aspects), how communications will be handled, and quality assurance processes and standards. Smaller projects may not need a wireframe or project plan (a projected deployment date will do), and quality assurance may be as simple as discussing browser and device type support.

1.3. Web Server Installation.

- **Web Server Definition**
  - ✓ A web server is a computer system capable of delivering web content to end users over the internet via a web browser.
- **How web servers work**

The end user processes a request via a web browser installed on a web server. The communication between a web server or browser and the end user takes place using Hypertext Transfer Protocol (HTTP). The primary role of a web server is to store, process, and deliver requested information or webpages to end users. It uses:

- ✓ **Physical Storage:** All website data is stored on a physical web server to ensure its safety. When an end user enters the URL of your website or searches it using a keyword on a browser, a request is generated and sent to the web server to process the data.

- ✓ **Web browser:** The role of web browsers such as Firefox, Chrome, or Internet Explorer is to find the web server on which your website data is located. Once the browser finds your server, it reads the request and processes the information.

A web server's primary role is to serve web pages for a website. A web page can be rendered from a single HTML file, or a complex assortment of resources fitted together. If you want to host your web application on the internet, in many cases you will need a web server.

One of the most common use cases for web servers is serving files necessary for rendering a website in a browser. When you visit **http://www.google.com**, you begin with entering a URL that starts a request over the internet. This request passes through multiple layers, one or more of which will be a web server. This web server generates a response to your request, which in this case is the Google website, specifically the homepage. Ideally, this happens quickly and with 24/7 availability.

Web servers act as an intermediary between the backend and the frontend, serving up resources. While the term "web server" can refer to either the software itself or the hardware it exists on. A web server handles requests on the internet through HTTP and HTTPS protocol, and is also called an HTTP *server*. A web server is distinct from other types of servers in that it specializes in handling these HTTP and HTTPS requests, differentiating itself from application servers and servers for other protocols

- **Here are some common tasks handled by web servers:**

- ✓ Serves HTML, CSS, and JavaScript files.
- ✓ Serves images and videos.
- ✓ Handles HTTP error messaging.
- ✓ Handles user requests, often concurrently.
- ✓ Directs URL matching and rewriting.
- ✓ Processes and serves dynamic content.
  - ✓ Compresses content for optimized data usage and speed.

- ✓ Enables browser caching for your static content.

In practical terms, here are some personal projects that would involve a web server:

- ✓ You want to make a website.
- ✓ You want to make an app that connects to the internet. This list is by no means comprehensive, and a web server is not strictly limited in the data types it can serve to an end user. For example, a web server that serves web API requests often responds with data in a format such as JSON.

- **Goals of a Web Server**

Web servers cater to an audience with expectations of speed, availability, reliability, and more. They have a shared purpose of serving content on the internet, and in order to be considered a viable web server solution, the following aspects must be considered:

- ✓ **Uptime**: This refers to the time a web server is online and operational. Websites need to be online at all times to serve users, so a high uptime is the goal. This also translates to stability and predictability. When a user enters a URL or clicks a link to your website, the expected page should load every time, and at any given time. The only exceptions should be planned downtimes for updates or maintenance. A web server that is buggy or crashes at random times adversely affects your users' experience.
- ✓ **Speed**: Your web pages should load as fast as possible. Users want their request fulfilled immediately, otherwise you risk losing them. On a slow loading web page, even if the user sits through the first load, every subsequent long load will exponentially decrease their willingness to stay or visit again.
- ✓ **Concurrency**: This refers to the handling of multiple requests coming in at the same time. Having too many people trying to visit your website at once seems like a good thing, but this becomes a real problem when load times slow down to a crawl and your whole server crashes. Your physical or virtual server only has so many resources such as RAM and CPU compute power, and web servers must use these resources efficiently.

- ✓ **Scalability**: Scalability refers to either making your existing servers more powerful through vertical scaling, or adding more servers to your setup through horizontal scaling. As you grow your audience, you may reach a point where you need more than one or two small web servers.
- ✓ **Ease of set up**: Getting a project up and running quickly is key to the iteration of your project. A straightforward and repeatable install process is important for the first web server you set up, and the multiple web servers afterwards when you scale up.
- ✓ **Documentation**: Web servers are complex. The most common setups will get you on your feet quickly, but your needs will grow over time. Oftentimes you will need features that are not as commonly used. When that time comes, good documentation is essential to creating custom solutions for your needs.
- ✓ **Developer support**: If the core developers are not committed to their own project, you shouldn't commit your project to theirs. This includes both plans for long term support for their software, along with immediate short term support they provide in the form of bug fixes and patches.
- ✓ **Community support**: A core development team will handle most of the heavy lifting, but a thriving community contributes to filling in the gaps. With open source projects, this can mean contributions to the actual code base, but a strong community will also answer your questions and help with your specific issues.

While web servers can offer different solutions, the solutions they offer stem from attempts to address these same problems. These problems themselves evolve over time along with the needs and expectations of the end user, making this a living and ever evolving list.

- **What are web servers used for?**

  - ✓ Web servers are primarily used to process and manage HTTP/HTTPS requests and responses from the client system.

A web server can also perform several other functions, such as:

- ✓ **Store and protect website data:** A web server can store and protect critical website data from unauthorized users.

- ✓ **Control bandwidth to regulate network traffic:** A web server can help eliminate the downtime caused by high web traffic. Web hosts can set bandwidth to manage the rate of data transmission over the internet and minimize the excess network traffic.
- ✓ **Server-side web scripting:** The server-side web scripting feature enables users to create dynamic web pages using scripting languages such as Ruby, Python, and PHP.
- ✓ **Virtual hosting:** Web servers can also be used as virtual servers to run multiple applications, websites, data, and other services.

- **Web server software list**

Some of the most common web servers are outlined below:

- ✓ **Apache web server software**: Apache web server or Apache HTTP server is an open-source server that processes user requests and delivers web assets and content via HTTP. This web server uses the MySQL database to store critical information in an easily readable format. With the help of the PHP programming language, Apache can create and serve dynamic web content.
- ✓ **IIS web server software:** Microsoft Internet Information Service (IIS) web server is also known as a Windows web server. It’s one of the most commonly used web servers used on a Windows operating system. It is a versatile and stable web server widely used to host ASP.NET web applications, static websites, and web applications built on PHP. Although it has a built-in authentication option such as Windows, ASP.NET, and Basic, it’s easier for Windows users to sign in to various web applications using their domain account. Other built-in security features include TLS certificate management, request logging, FTP-specific security options, and more.
- ✓ **Linux web server software:** Linux server is built on an open-source Linux operating system that enables you to deliver content, applications, and services to end users. Linux servers are flexible, consistent, and high-performing servers with snapshot capabilities, optimized security, and scalable cloud technologies. These servers help address the increasing requirements of web services, applications, database management, and more.
- ✓ **NGINX web server software:** NGINX is a popular open-source web server that runs and utilizes resources efficiently. It can handle huge volumes of traffic. It offers reverse proxy, HTTP caching services, email proxy, and load balancing. NGINX is a

scalable, lightweight, and powerful web server capable of handling concurrent connections and is ideal for delivering static content.

- **Web server vs. application server differences**

✓ **Web server:** The web server accepts and processes requests from end users for static website content. It handles requests and responses via HTTP only. Web servers are generally helpful in serving static content or static HTML webpages. It consumes fewer resources such as CPU or memory compared to the application server and provides a runtime environment for web applications.

✓ **Application server:** The application server can deliver web content and dynamic content required for displaying decision support, transaction results, or real-time analytics. However, its primary role is to enable interaction between the end user and server-side application code. These servers enhance interactive content or website components depending on the request. Application servers use web containers. These servers use more resources compared to web servers and provide the runtime environment for enterprise applications.

- **Benefits of optimizing a web server**

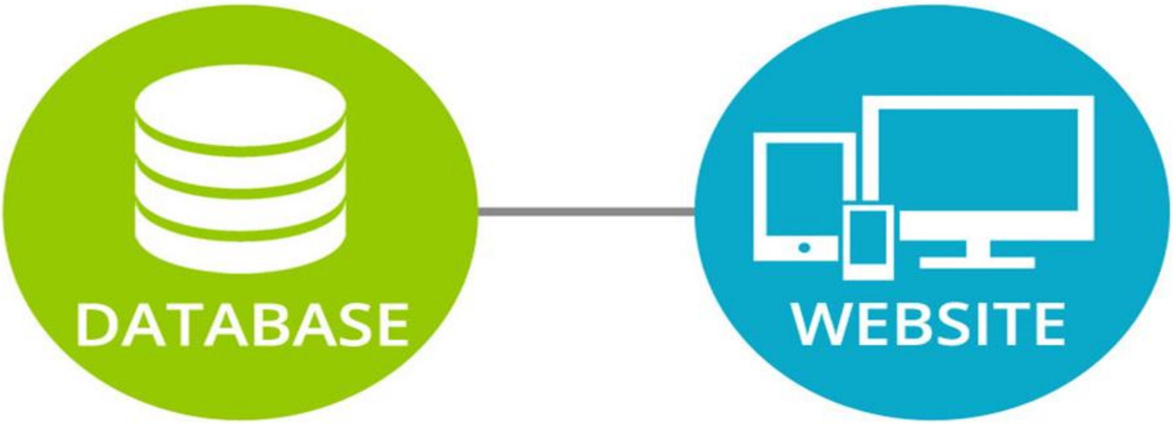
Optimizing a web server requires regular monitoring of web and application servers. Outlined below are the few benefits of monitoring and optimizing your server:

✓ **Helps solve critical problems quickly:** It’s crucial to monitor web and application servers to ensure availability and performance. Monitoring web servers provides vital insights into application pools (worker processes, cache, and requests), connections (current and total connections), websites (network, files), cache (memory usage, file cache).

✓ **Optimize infrastructure resources:** It helps to understand key performance metrics, website load, so you can effectively use infrastructure resources such as CPU utilization, network traffic, disk capacity, and more. It also provides crucial insights, such as client connections, web server traffic and status, server load.

1.4.Connect database with web application

The steps involved in creating and connecting database to web server



- **Prepare your database user account details**.
  - ✓ Database systems use accounts, with specific levels of access to each user. Account details should include a username and password. Locate these details, if necessary copying them into a file.
- **Connect to database.**
  - ✓ One or more server side scripts to connect to your database. The process for making a connection is similar for other database systems and programming languages.
- **Query data**
  - ✓ In most cases scripts use SQL (Structured Query Language) to retrieve specific sets of data from databases. These SQL queries can execute from inside a server side script.
- **Output data:** Once data is retrieved from the database it can be represented in the a structured form in HTML page
- **Test script**

Once you have your database connection script complete, or partially complete, upload it to your server to test it. Once you have your database connection script complete, or partially complete, upload it to your server to test it. If you encounter errors, check your database account details as well as the structure of your tables.

- **Template-driven packages**
  - ✓ A template driven database access package, is another way to link a Web front end to a database back end.
- **The approach**

Template-driven database connectivity packages are offered by database vendors and third-party developers to simplify Web database application programming. Such a package usually consists of the following components:

- ✓ Template consisting of HTML and non-standard tags or directives
- ✓ Template parser
- ✓ Database client daemons

Template-driven packages are very product dependent. Different DBMSs require database access templates in different formats. An application developed for one product will be strongly tied to it. Migrating from one product to another is very difficult and requires a rewrite of all the database access, flow control and output-formatting commands.

An example of a template-driven package is PHP.

- **Benefits of template-driven packages**

The most important benefit from using a template-driven package is speed of development. Assuming an available package has been installed and configured properly, it takes as little time as a few hours to create a Web site that displays information directly from the database.

- **Shortcomings of template-driven packages**

The structures of templates are normally predetermined by vendors or third party developers. As a result, they only offer a limited range of flexibility and customis ability. Package vendors provide what they feel is important functionality, but, as with most off-the-shelf tools, such software packages may not let you create applications requiring complex operations.

Although templates offer a rapid path to prototyping and developing simple
Web database applications, the ease of development is obtained for the cost of speed and efficiency. Because the templates must be processed on demand and require heavy string manipulation (templates are of a large text type or string type; they must be parsed by the parser), using them is slow compared with using direct access such as native database APIs.

The actual performance of an application should be tested and evaluated before the usefulness of such a package is ruled out. The overhead of parsing templates may be negligible if using high-performance machines. Other factors, such as development time or development expertise, may be more important than a higher operational speed.

- **GUI application builders**
    - Visual Web database building tools offer an interesting development environment for creating Web database applications. For developers accustomed to point-and-click application programming, these tools help speed the development process. For instance, Visual Basic and/or Microsoft Access developers should find such a tool intuitive and easy to use.
    - **The approach**

The architectures of visual building tools vary. In general, they include a user friendly GUI (Graphical User Interface), allowing developers to build a Web database application with a series of mouse clicks and some textual input. These tools also offer application management so that a developer no longer needs to juggle multiple HTML documents and other API programs manually. At the end of a building session, the tool package can generate applications using various techniques. Some applications are coded using ODBC; some use native database APIs for the databases they support; and others may use database net protocols.

Some of these tools create their own API, which can be used by other developers. Some generate code that works but can still be modified and customized by developers using various traditional IDEs, compilers and debuggers. A building tool may generate a CGI program or a Web server API program (such as NSAPI and ISAPI). Some sophisticated tools even offer all the options. The developer can choose what he/she wants.

Unlike native database APIs or template-driven database connectivity packages, visual Web database development tools tend to be as open as possible. Many offer development support for the popular databases.

- **Benefits of visual tools**

Visual development tools can be of great assistance to developers who are familiar and comfortable with visual application development techniques. They offer rapid application development and prototyping, and an organized way to manage the application components. Visual tools also shield the developer from low-level details of Web database application development. As a result, a developer can create a useful Web application without the need to know what is happening in the code levels.

- **Shortcomings of visual tools**

Depending on the sophistication of the package used, the resulting programs may be slower to execute than similar programs coded by an experienced programmer. Visual application building tools, particularly Object-oriented ones, tend to generate fat programs with a lot of unnecessary sub-classing.

Another potential drawback is cost. A good visual tool may be too expensive for a small one-off development budget.

HTML are markup languages, basically they are set of tags like <html>, <body>, which is used to present a website using css, and javascript as a whole. All these, happen in the clients system will be browsing the website. Now, Connecting to a database, happens on whole another level. It happens on server, which is where the website is hosted. So, in order to connect to the database and perform various data related actions, you have to use server-side scripts, like php, jsp, asp.net etc.

MySQL is a popular open source database management system commonly used in web applications due to its speed, flexibility and reliability. MySQL employs SQL, for accessing and processing data contained in databases. Simple web applications can be designed using a two-tier architecture, in which a client communicates directly with a server. A web application communicates directly with a MySQL database using the Database Connectivity API. Essentially, it is the MySQL Connector and web application Driver that enables communication between the website code understood by the application server, and any content in SQL.

First of all, you must install any web server or Web Server Applications like **XAMPP or WAMP(Windows) or MAMP (Mac OS)** kind of software on your computer to get a local webserver i.e. Apache, PHP language, and MySQL database.

Steps to connect the web application to the database contains the following steps

Step 1: Design your Database and HTML form requirements for your web page

Step 2: Create a database and a table in MySQL

Step 3: Create HTML form for connecting to database

Step 4: Create a PHP page to save data from HTML form to your MySQL database

Step 5: All done**!**

Self-check 1

Part-I: Fill the blank space

1. ____________________________Website data is stored on a physical web server to ensure its safety.
2. ______________________________________a bridge between the Web and a DBMS, and its objective is to provide a Web-based application the ability to manipulate data stored in the database
3. ______________________refers to the time a web server is online and operational
4. _________________________refers to either making your existing servers more powerful through vertical scaling, or adding more servers to your setup through horizontal scaling.
5. Web servers are primarily used to process and manage _________/___________requests and responses from the client system
6. _____________________can deliver web content and dynamic
7. ______________________content required for displaying decision support, transaction results, or real-time analytics.
8. ______________data is retrieved from the database it can be represented in the a
9. structured form in HTML page

Part-II: Answer the following questions accordingly

1. What is the difference between Static and Dynamic Websites?
2. Are the Internet and WWW (Web) the same concept? Why?
3. Most Web sites have URLs starting with http://Why?
4. What are the major features of a Web-based client-server application?
5. What is a gateway in a Web database application and why is it needed?
6. Where can we implement a gateway for a Web database application?
7. What is Apache web server software?

Operation sheet 1.1. Integrating database with a website

Operation Title: connecting the web application to the database

Purpose: Generate and execute a new database file to test and understand basic database syntax.

Conditions or situations for the operations:

- ✓ Safe working area
- ✓ Properly operated tools and equipment

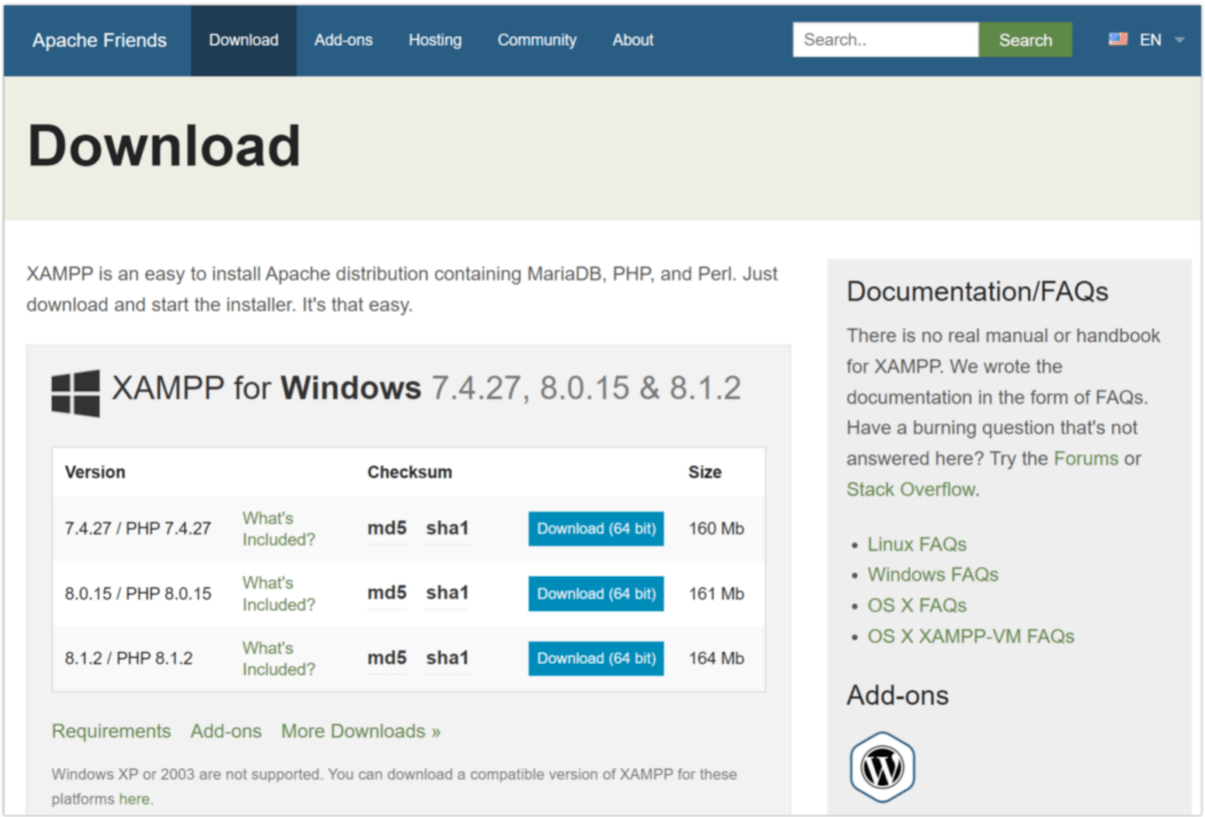
Equipment Tools and Materials:

- ✓ Computer
- ✓ Windows 10 and above.

Steps in doing the task

Step 1: Download XAMPP

Start off by downloading the latest version of XAMPP from Apache Friends website. The current up-to-date version of XAMPP is 8.1.2 / PHP 8.1.2.



Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for **Windows** 7.4.27, 8.0.15 & 8.1.2

Version		Checksum			Size
7.4.27 / PHP 7.4.27	What's Included?	md5	sha1	Download (64 bit)	160 Mb
8.0.15 / PHP 8.0.15	What's Included?	md5	sha1	Download (64 bit)	161 Mb
8.1.2 / PHP 8.1.2	What's Included?	md5	sha1	Download (64 bit)	164 Mb

Requirements Add-ons More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs

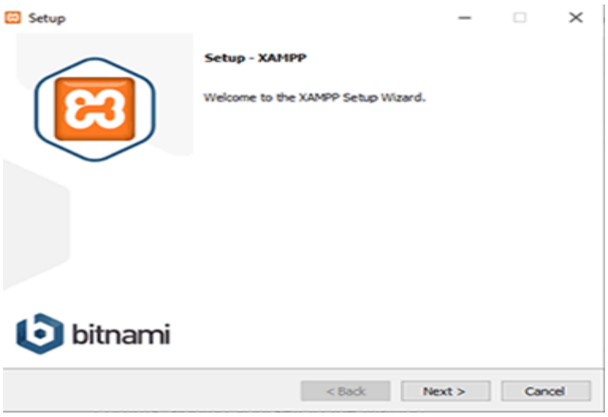
Add-ons

XAMPP & PHP most recent file versions.

Note: If you’re using Linux or OSX, the installation steps are pretty much the same. Just make sure to download the appropriate version of XAMPP for your OS.

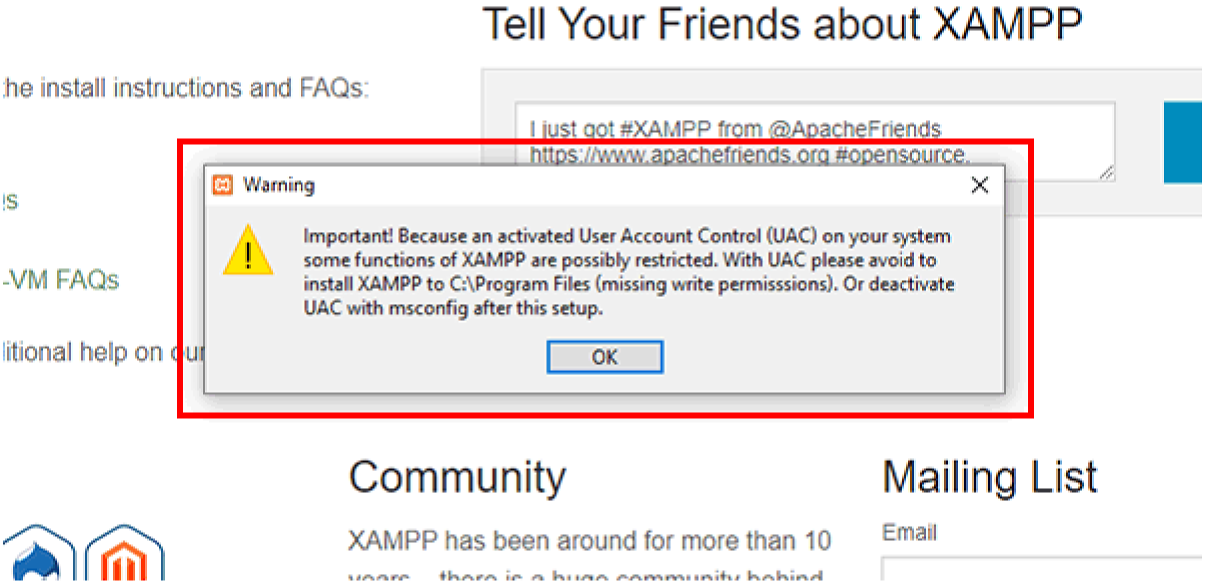
Step 2: Install XAMPP

Once downloaded, run the XAMPP installer file.



Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

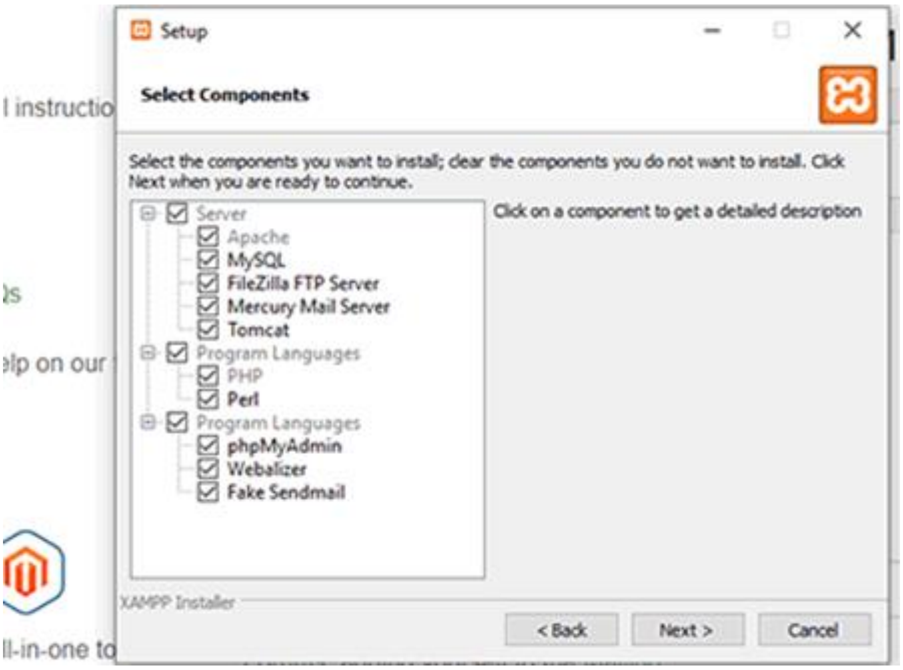


Tell Your Friends about XAMPP
he install instructions and FAQs:
I just got #XAMPP from @ApacheFriends https://www.apachefriends.org #opensource.
s
-VM FAQs
itional help on ou
Warning
OK
Community
XAMPP has been around for more than 10
Mailing List
Email

Important! Because an activated User Account Control (UAC) on your system some functions of XAMPP are possibly restricted. With UAC please avoid to install XAMPP to C:\Program Files (missing write permisssions). Or deactivate UAC with msconfig after this setup.

You might get a UAC warning before **installation. Click OK and continue**.

Select the components you want to install. If you’re planning to install a WordPress site with XAMPP, you only need Apache, MySQL, PHP, **and phpMyAdmin**. I’ll check all the components as I’d like to experiment with them later.



Setup

Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

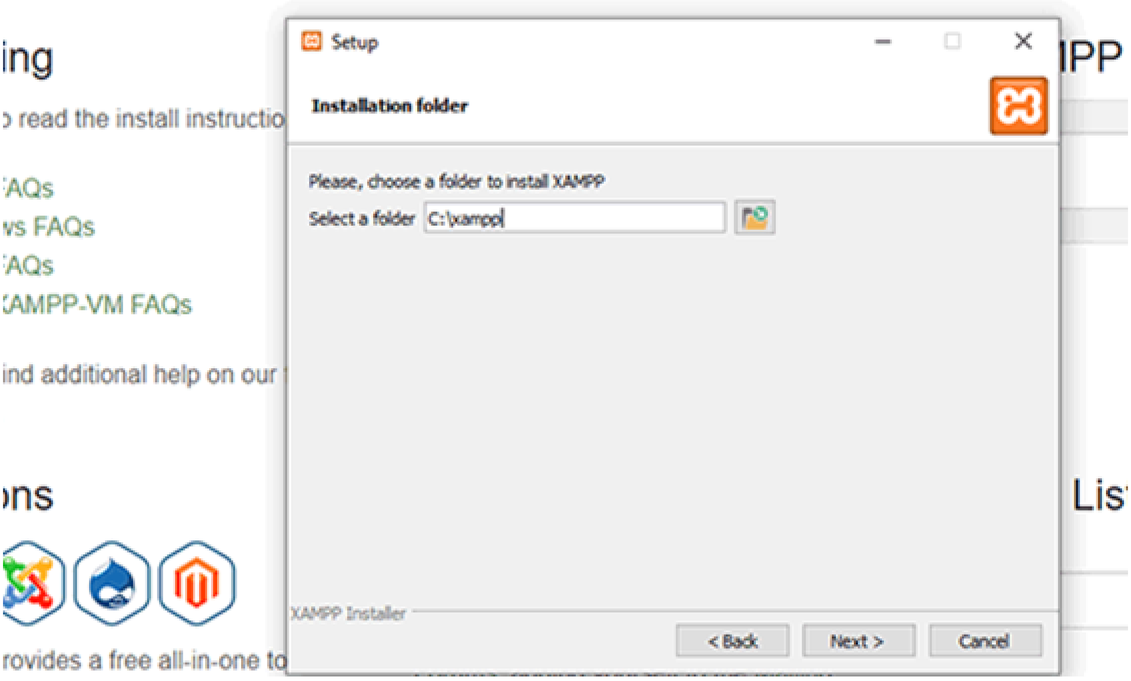
- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - ☑ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - ☑ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - ☑ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - ☑ Webalizer
  - ☑ Fake Sendmail

Click on a component to get a detailed description

XAMPP Installer

Choose the installation directory for XAMPP (default recommended).

Choose the installation directory for XAMPP (default recommended).



Setup

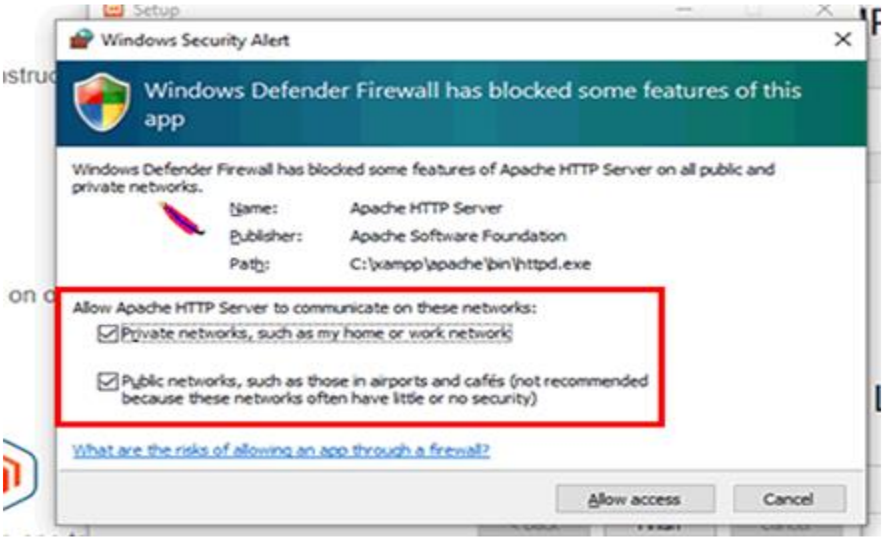
Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

XAMPP Installer

Uncheck Learn more about Bitnami option. Bitnami provides all-in-one tools to install popular open source apps on top of XAMPP. This includes add-on modules for installing Word Press too. However, we’ll be installing it manually here.



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of Apache HTTP Server on all public and private networks.

Name: Apache HTTP Server
Publisher: Apache Software Foundation
Path: C:\xampp\apache\bin\httpd.exe

Allow Apache HTTP Server to communicate on these networks:

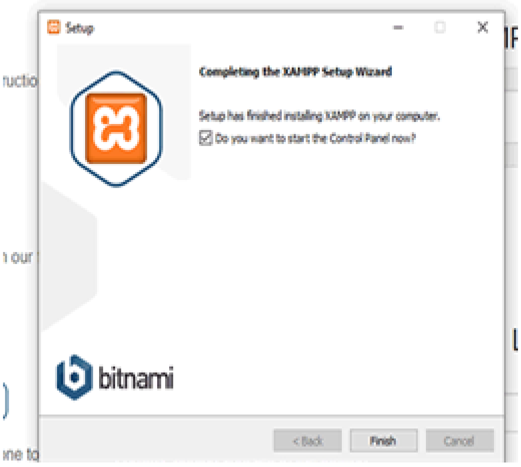
☑ Private networks, such as my home or work network

☑ Public networks, such as those in airports and cafés (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

You may be presented with a Windows Security Alert at the end of the installation. You need to whitelist Apache HTTP Server from your Windows Defender Firewall by clicking the Allow access button. Make sure to check the “Private networks, such as my home or work network” option. This is very important. XAMPP won’t work if you don’t check this and click Allow access.

Complete the setup and run XAMPP Control Panel.



Setup

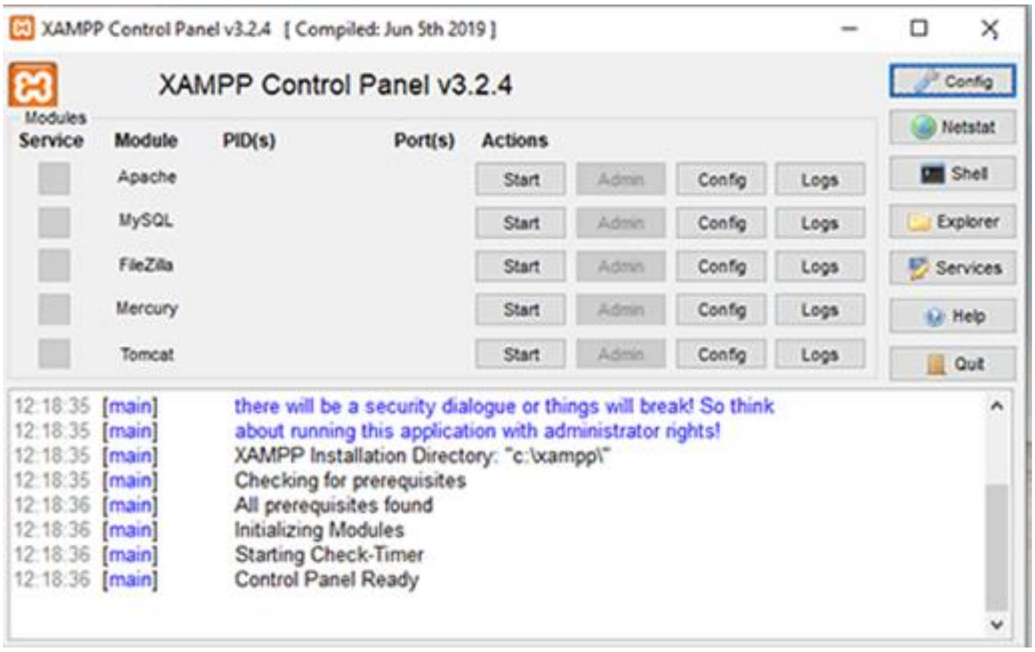
Completing the XAMPP Setup Wizard

Setup has finished installing XAMPP on your computer.

☑ Do you want to start the Control Panel now?

Step 3: Open the XAMPP Control Panel

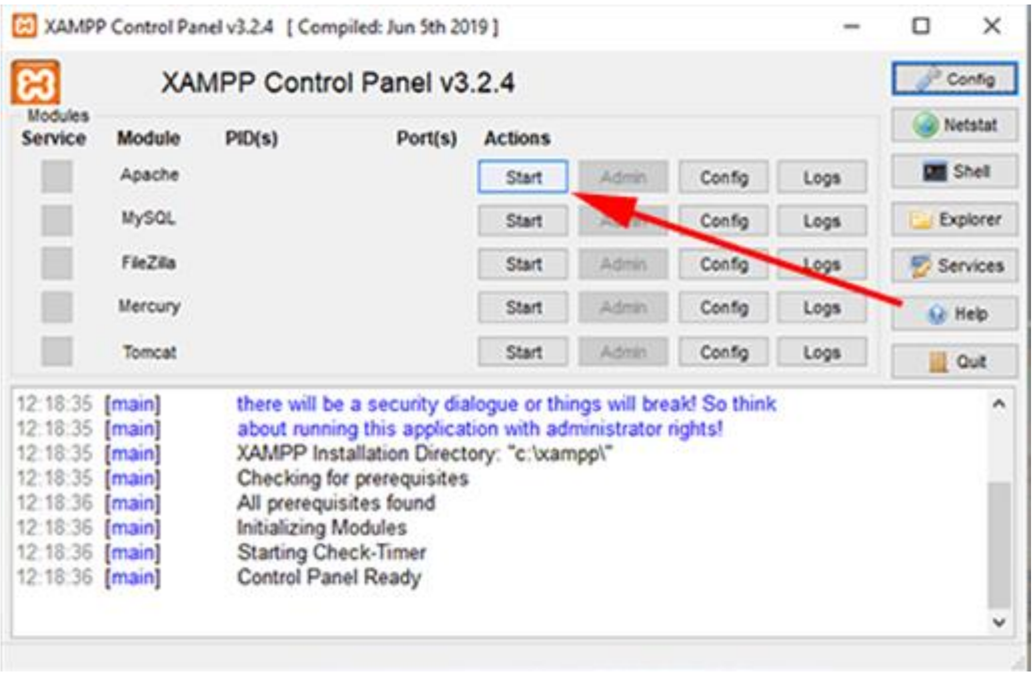
The XAMPP Control Panel sports a simple user interface that lists all the modules of your local server. It allows you to Start/Stop individual modules, access their Admin area, Config files, and Logs with just a single click. Its bottom section also displays all your actions and errors (if any).



```
12:18:35  [main]   there will be a security dialogue or things will break! So think
12:18:35  [main]   about running this application with administrator rights!
12:18:35  [main]   XAMPP Installation Directory: "c:\xampp\"
12:18:35  [main]   Checking for prerequisites
12:18:36  [main]   All prerequisites found
12:18:36  [main]   Initializing Modules
12:18:36  [main]   Starting Check-Timer
12:18:36  [main]   Control Panel Ready
```

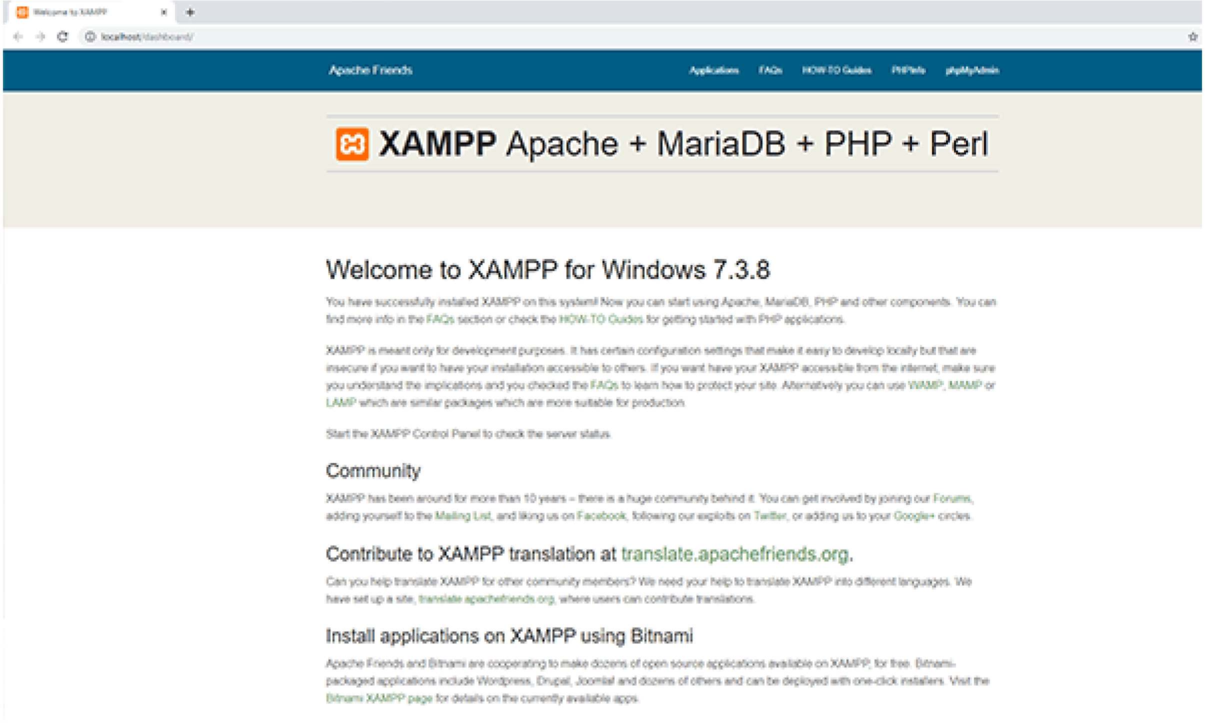
Step 4: Start Apache and MySQL Modules

Click the Start button beside Apache module. If everything's set correctly, your Apache server should start successfully under ports 83 and 443.



```
12:18:35  [main]   there will be a security dialogue or things will break! So think
12:18:35  [main]   about running this application with administrator rights!
12:18:35  [main]   XAMPP Installation Directory: "c:\xampp\"
12:18:35  [main]   Checking for prerequisites
12:18:36  [main]   All prerequisites found
12:18:36  [main]   Initializing Modules
12:18:36  [main]   Starting Check-Timer
12:18:36  [main]   Control Panel Ready
```

You can access your Apache server's dashboard by clicking the Admin button beside it. Alternatively, you can also reach it via http://localhost/dashboard/ URL in your browser.



Welcome to XAMPP for Windows 7.3.8

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

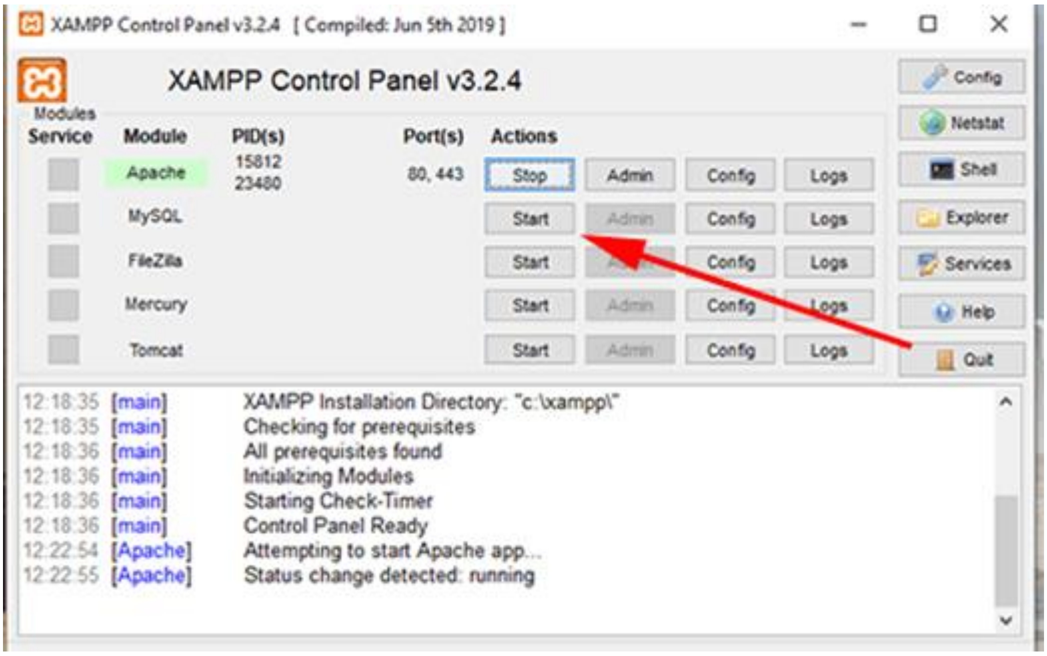
Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We have set up a site, translate.apachefriends.org, where users can contribute translations.

Install applications on XAMPP using Bitnami

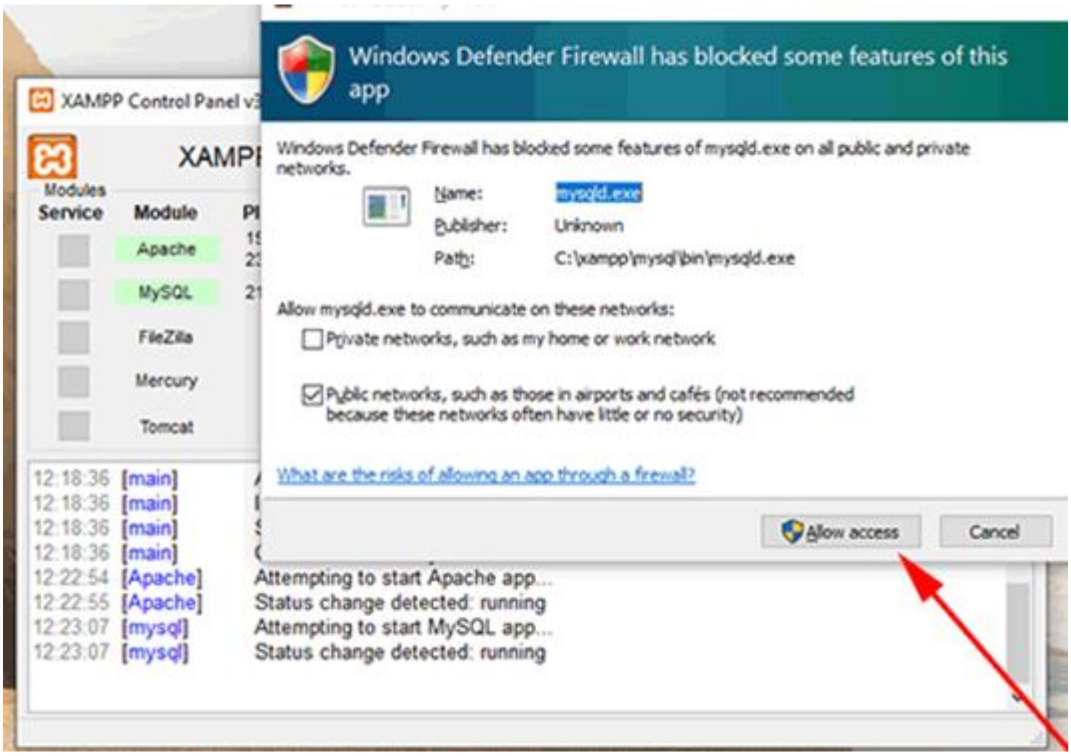
Apache Friends and Bitnami are cooperating to make dozens of open source applications available on XAMPP, for free. Bitnami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the Bitnami XAMPP page for details on the currently available apps.

Then Start the MySQL module. If you’re presented with a Windows Security Alert to whitelist mysqld.exe, click Allow access. Like before, make sure that you’ve ticked the “Private networks…” option.



Module	PID(s)	Port(s)
Apache	15812 23480	80, 443
MySQL		
FileZilla		
Mercury		
Tomcat		

```
12:18:35  [main]     XAMPP Installation Directory: "c:\xampp\"
12:18:35  [main]     Checking for prerequisites
12:18:36  [main]     All prerequisites found
12:18:36  [main]     Initializing Modules
12:18:36  [main]     Starting Check-Timer
12:18:36  [main]     Control Panel Ready
12:22:54  [Apache]   Attempting to start Apache app...
12:22:55  [Apache]   Status change detected: running
```



Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of mysqld.exe on all public and private networks.

Name: mysqld.exe
Publisher: Unknown
Path: C:\xampp\mysql\bin\mysqld.exe

Allow mysqld.exe to communicate on these networks:

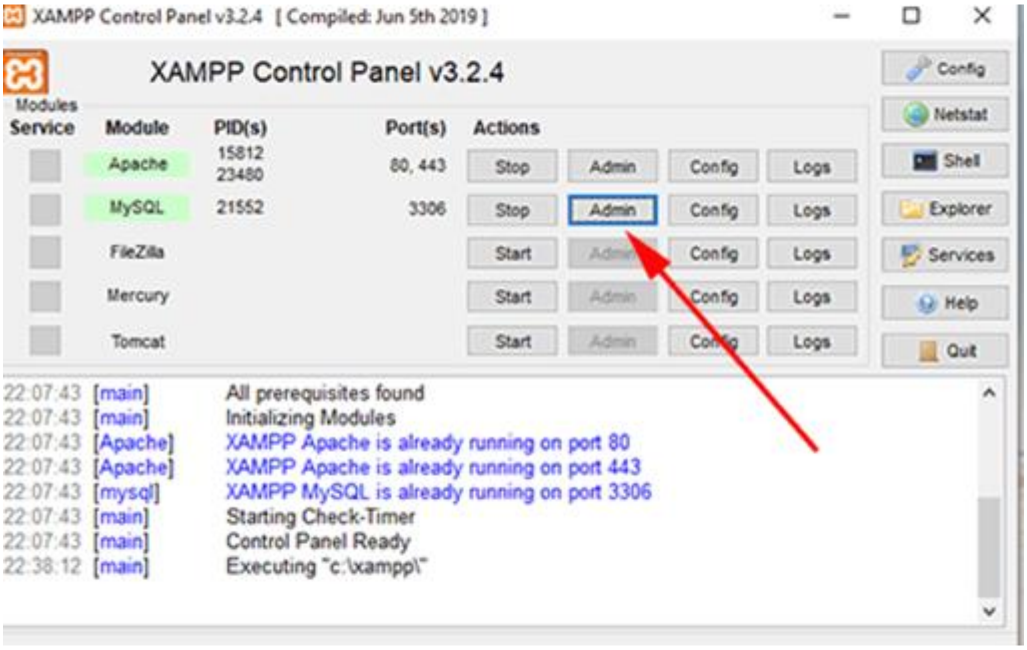
□ Private networks, such as my home or work network

☑ Public networks, such as those in airports and cafés (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

```
12:18:36  [main]
12:18:36  [main]
12:18:36  [main]
12:18:36  [main]
12:22:54  [Apache]    Attempting to start Apache app...
12:22:55  [Apache]    Status change detected: running
12:23:07  [mysql]     Attempting to start MySQL app...
12:23:07  [mysql]     Status change detected: running
```

Don’t forget to tick the “Private networks” option. You can access your phpMyAdmin dashboard by clicking the Admin button beside MySQL module. Or you can simply go to http://localhost/phpmyadmin/ in your browser. Here, you can manage the MariaDB (or MySQL) databases of your web projects.

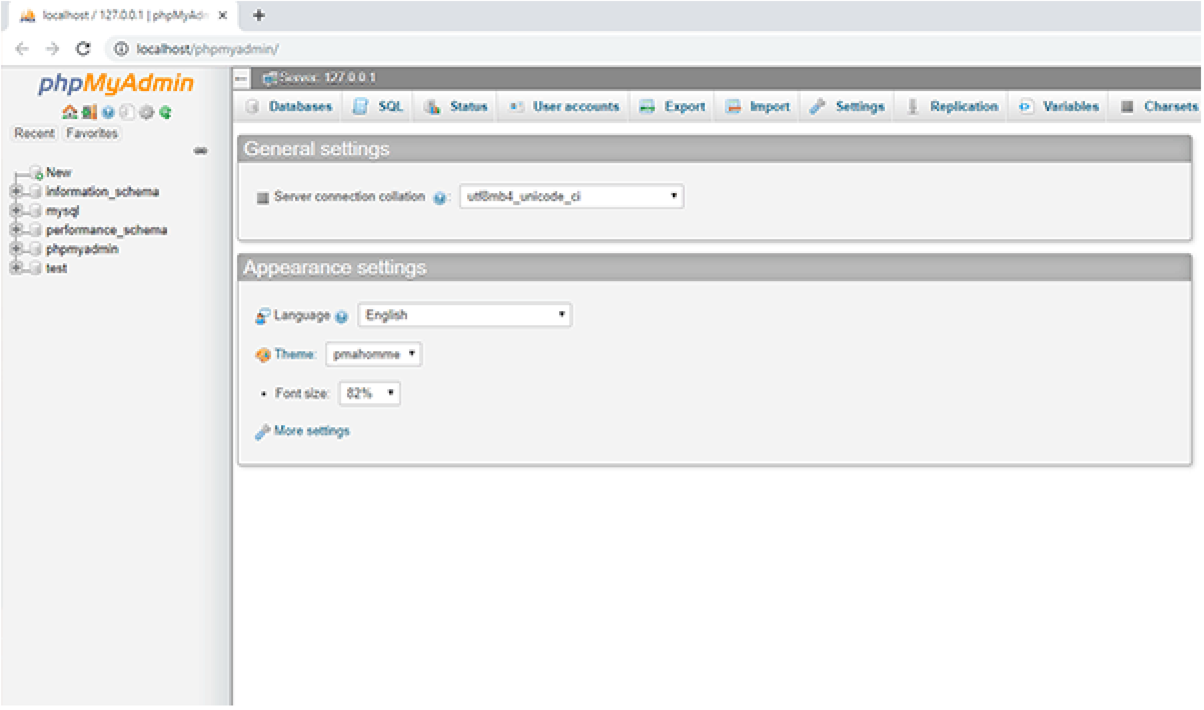


XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions			
	Apache	15812 23480	80, 443	Stop	Admin	Config	Logs
	MySQL	21552	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
22:07:43  [main]      All prerequisites found
22:07:43  [main]      Initializing Modules
22:07:43  [Apache]    XAMPP Apache is already running on port 80
22:07:43  [Apache]    XAMPP Apache is already running on port 443
22:07:43  [mysql]     XAMPP MySQL is already running on port 3306
22:07:43  [main]      Starting Check-Timer
22:07:43  [main]      Control Panel Ready
22:38:12  [main]      Executing "c:\xampp\"
```



General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

Font size: 82%

More settings

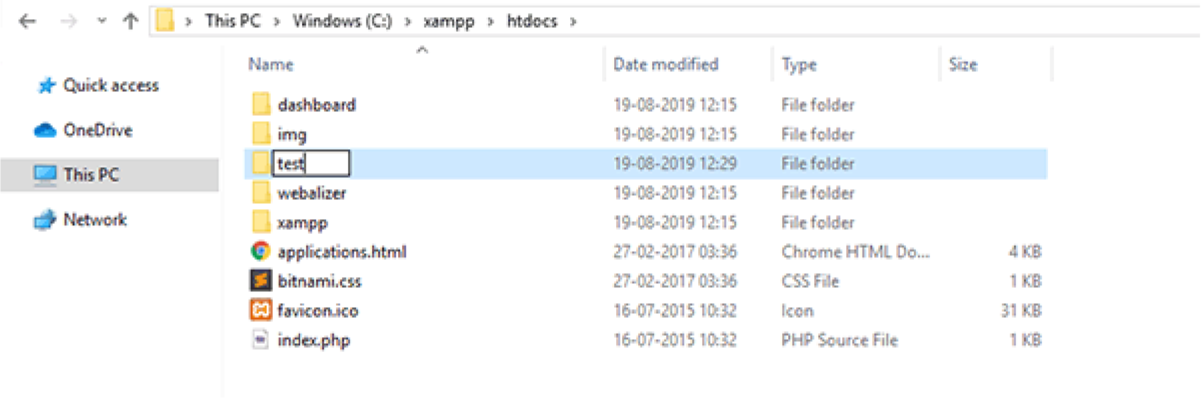
The phpMyAdmin Dashboard

This concludes the setup of XAMPP as your local development environment. It's now ready to host any PHP-based software (e.g. Word Press).

Testing Your XAMPP Installation

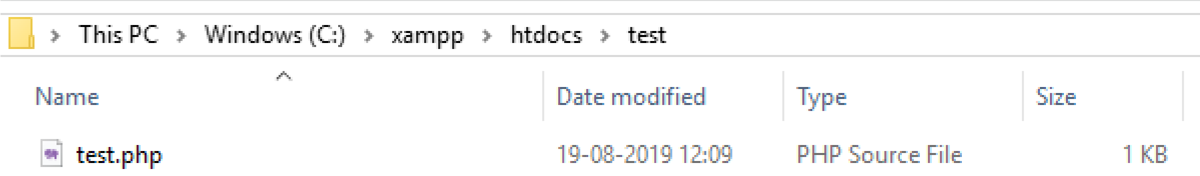
The best way to check whether your local server has been installed and configured correctly is to create a PHP test page, place it in XAMPP's local host folder, and then try accessing it via your browser.

Let's do that now. Create a new folder called test in your C:\xampp\htdocs\ directory. This directory can also be accessed easily by clicking the Explorer button in XAMPP Control Panel and then going to htdocs folder.

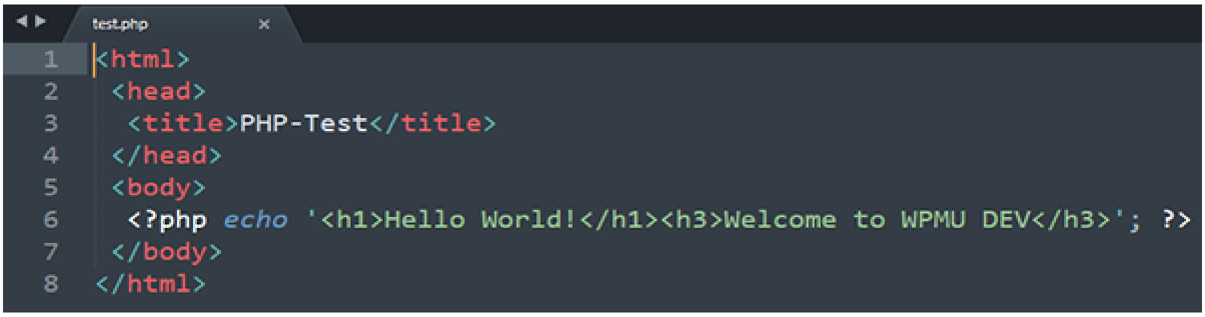


Name	Date modified	Type	Size
dashboard	19-08-2019 12:15	File folder	
img	19-08-2019 12:15	File folder	
test	19-08-2019 12:29	File folder	
webalizer	19-08-2019 12:15	File folder	
xampp	19-08-2019 12:15	File folder	
applications.html	27-02-2017 03:36	Chrome HTML Do...	4 KB
bitnami.css	27-02-2017 03:36	CSS File	1 KB
favicon.ico	16-07-2015 10:32	Icon	31 KB
index.php	16-07-2015 10:32	PHP Source File	1 KB

Create a file called test.php with the code below and place it in C:\xampp\htdocs\test folder.



Name	Date modified	Type	Size
test.php	19-08-2019 12:09	PHP Source File	1 KB



```
<html>
 <head>
  <title>PHP-Test</title>
 </head>
 <body>
  <?php echo '<h1>Hello World!</h1><h3>Welcome to WPMU DEV</h3>'; ?>
 </body>
</html>
```

<html>

<head>

<title>PHP-Test</title>

</head>

<body>

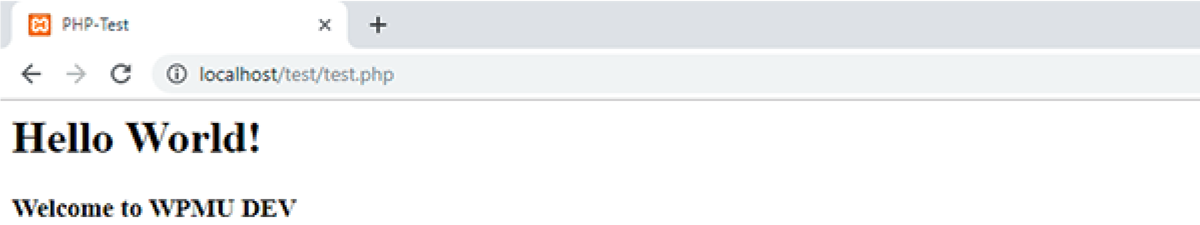
<?php echo '<h1>Hello World!</h1><h3>Welcome to WPMU DEV</h3>'; ?>

</body>

</html>

view raw test.php hosted with ❤ by GitHub

Try visiting http://localhost/test/test.php in your browser. If it displays the words "Hello World! Welcome to WPMU DEV", then XAMPP is successfully installed and configured on your system.



Hello World!

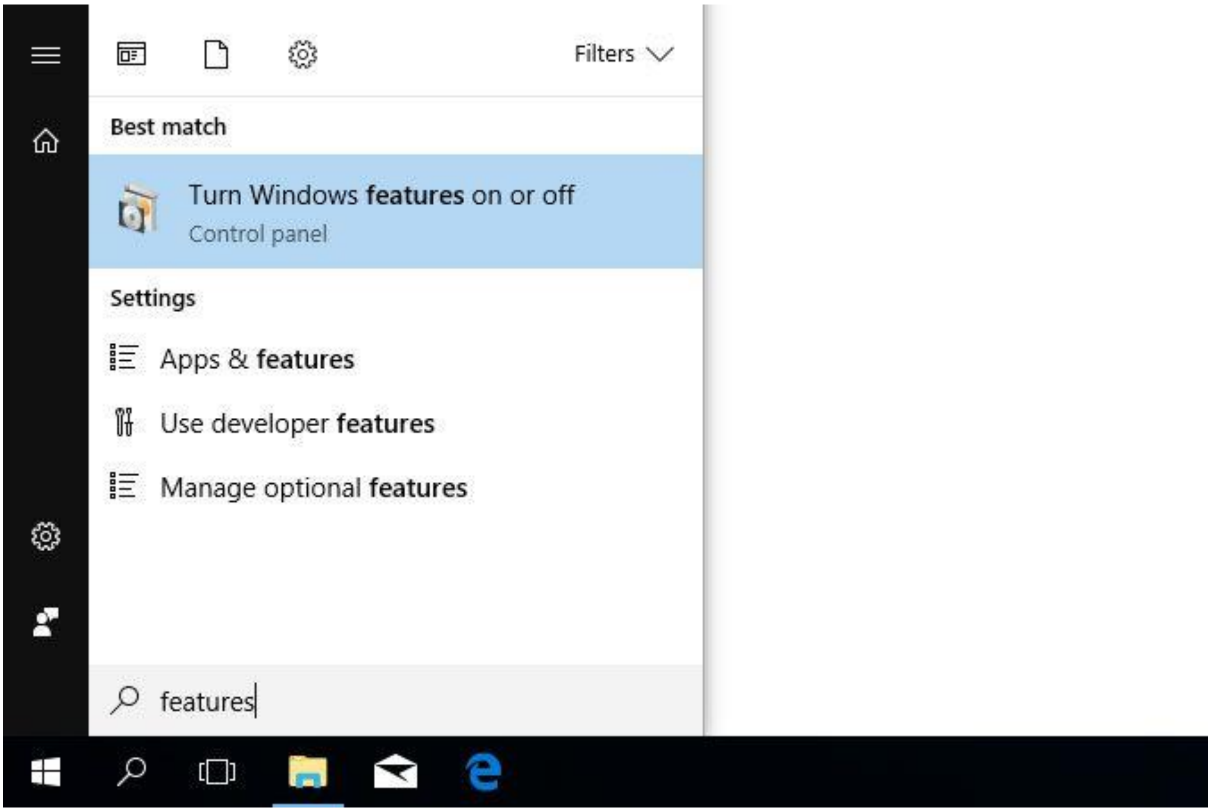
Welcome to WPMU DEV

If IIS is not installed by default, you can download and install the latest version of IIS Express for your Windows 10 or Windows 11 OS from the Microsoft website. IIS versions later than 7.0 include the new IIS Manager User interface.

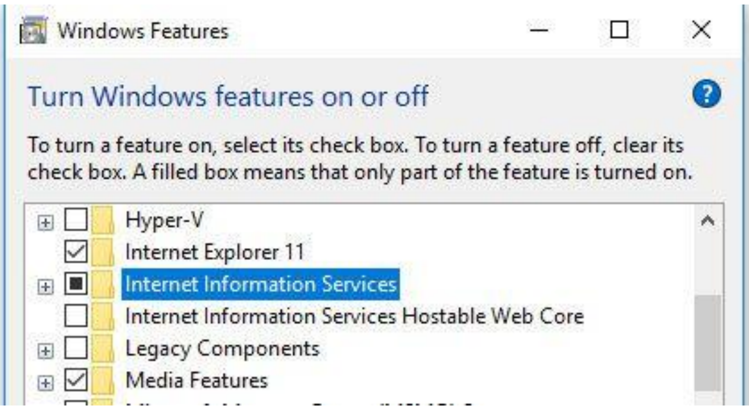
Enable Internet Information Services (IIS) on a Windows computer

1. Open the Start menu.

2. Type "features" and select Turn Windows features on or off



3. Tick the Internet Information Services checkbox and select OK.

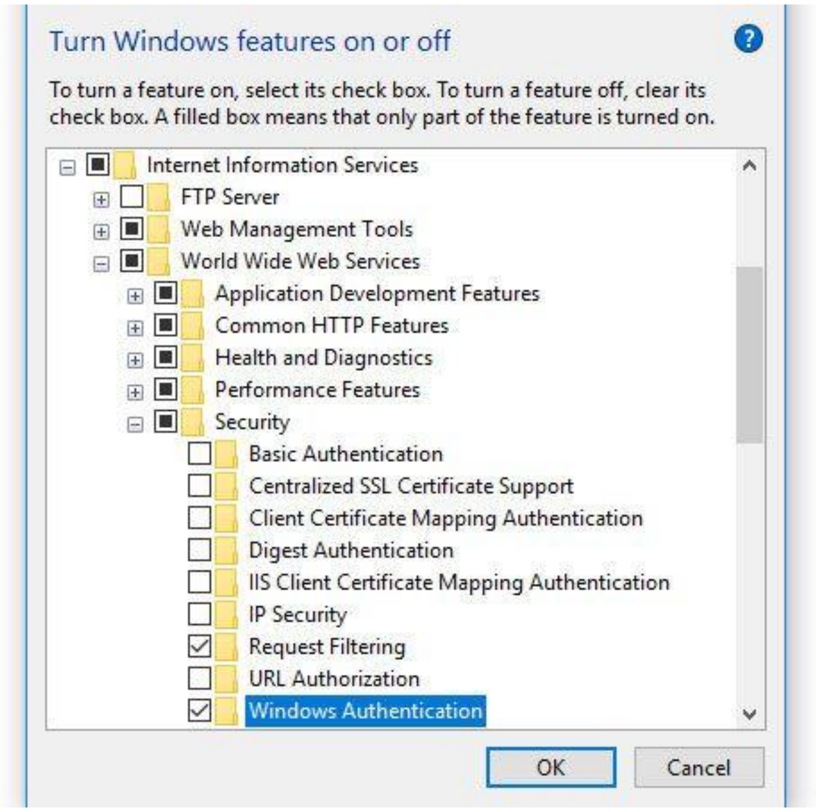


To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊞ □ Hyper-V
- ☑ Internet Explorer 11
- ⊞ ■ Internet Information Services
- □ Internet Information Services Hostable Web Core
- ⊞ □ Legacy Components
- ⊞ ☑ Media Features

4. Wait for the installation to complete and select Close.

5. If you plan on using integrated Windows authentication in website hosted in IIS, tick the Windows Authentication option under Internet Information Services > **World Wide Web Services > Security as well and select OK.**



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊟ ■ Internet Information Services
  - ⊞ □ FTP Server
  - ⊞ ■ Web Management Tools
  - ⊟ ■ World Wide Web Services
    - ⊞ ■ Application Development Features
    - ⊞ ■ Common HTTP Features
    - ⊞ ■ Health and Diagnostics
    - ⊞ ■ Performance Features
    - ⊟ ■ Security
      - □ Basic Authentication
      - □ Centralized SSL Certificate Support
      - □ Client Certificate Mapping Authentication
      - □ Digest Authentication
      - □ IIS Client Certificate Mapping Authentication
      - □ IP Security
      - ☑ Request Filtering
      - □ URL Authorization
      - ☑ Windows Authentication

1. You can now install and test your website on IIS. After installation, you will see your web console listed in IIS Manager (inetmgr), along with a default website generated when you enabled IIS.

Connecting the web application to the database

Step 1: Design your Database and HTML form requirements for your web page

Filter your HTML form requirements for your contact us web page

Step 2: Create a database and a table in MySQL

Step 3: Create HTML form for connecting to database

Step 1: Filter your HTML form requirements for your contact us web page

Suppose you selected the form field Name (text input), Email (email input), Phone (number input), and message (multi-line text). The form submit button also necessary for submitting the form. You will get the complete form in HTML coding in step 3.

Name	E-mail	Phone	Message

Step 2: Create a database and a table in MySQL

2a. you can use SQL Commands to create your database

```
CREATE DATABASE `db_Name`

USE `db_Name`;

CREATE TABLE `tbl_Name’()

DROP TABLE IF EXISTS `tbl_Name`;

ALTER TABLE

ADD/Delete PRIMARY KEY (`id`);
```

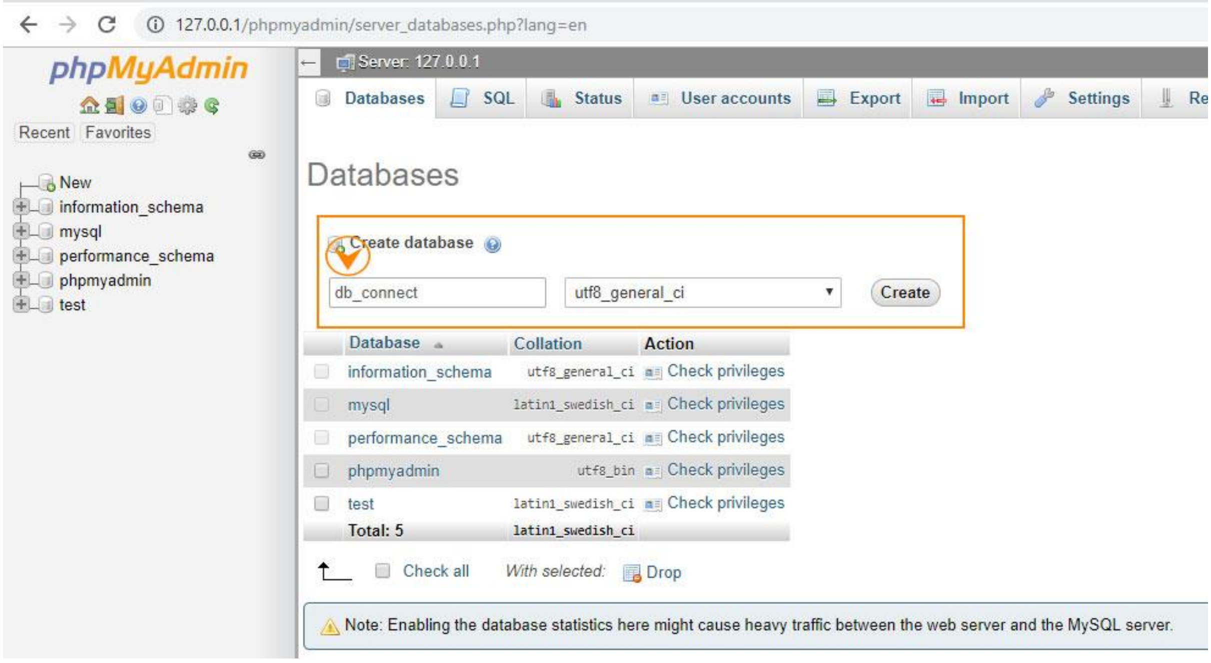
Example

```
CREATE DATABASE IF NOT EXISTS `db_contact`

USE `db_contact`;

CREATE TABLE IF NOT EXISTS `tbl_contact` (

`id` int(10) PRIMARY KEY NOT NULL,

  `fldName` varchar(50) NOT NULL,

  `fldEmail` varchar(45) NOT NULL,

`fldPhone` varchar(15) NOT NULL,

`fldMessage` text NOT NULL

)
```

2b. you can use your web server applications to create the Database

Open a web browser (chrome, Firefox, edge, etc.,) and type this http://localhost/phpmyadmin/ or http://127.0.0.1/phpmyadmin/ for open GUI for managing DB on your computer. See the xampp screen below how it is coming.

Click on the databases link and create your db by clicking"new"



Databases

Create database

db_connect | utf8_general_ci | Create

	Database	Collation	Action
□	information_schema	utf8_general_ci	Check privileges
□	mysql	latin1_swedish_ci	Check privileges
□	performance_schema	utf8_general_ci	Check privileges
□	phpmyadmin	utf8_bin	Check privileges
□	test	latin1_swedish_ci	Check privileges
	Total: 5	latin1_swedish_ci	

□ Check all *With selected:* Drop

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

E.g. click on the databases new link and create your db by the name "db_contact".

Step 3: Create HTML form for connecting to database

Now you have to create an HTML form. For this, you need to create a working folder first and then create a web page with the name "contact.html". If you install xampp your working folder is in folder this "E:\xampp\htdocs". You can create a new folder "contact" on your local host working folder. Create a "contact.html" file and paste the following code.

```
<html>

<head>

<title>Contact Form - PHP/MySQL Demo Code</title>

</head>

<body>

<fieldset>
```

```
<legend>Contact Form</legend>

<form name="frmContact" method="post" action="contact.php">

<p>

<label for="Name">Name </label>

<input type="text" name="txtName" id="txtName">

</p>

<p>

<label for="email">Email</label>

<input type="text" name="txtEmail" id="txtEmail">

</p>

<p>

<label for="phone">Phone</label>

<input type="text" name="txtPhone" id="txtPhone">

</p>

<p>

<label for="message">Message</label>

<textarea name="txtMessage" id="txtMessage"></textarea>

</p>

<p>&nbsp;</p>

<p>

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

</p>

</form>

</fieldset>
```

```
</body>

</html>
```

Now your form is ready. You may test it in your localhost link http://localhost/contact/contact.html

Connect webpage with the DB

For storing data in MySQL as records, you have to first connect with the DB. Connecting the code is very simple. The mysql_connect in PHP is deprecated for the latest version therefore I used it here mysqli_connect.

```
$con = mysqli_connect ("localhost", "your_localhost_database_user" ,"your_localhost_database_password", "your_localhost_database_db");
```

You need to place value for your localhost username and password. Normally localhost MySQL database username is root and password blank or root. For example, the code is as below

```
$con = mysqli_connect ('localhost', 'root', '',’db_contact’);
```

The “db_contact” is our database name that we created before. Then you will save the file with “contact.php”

Quality Criteria: The web page should display the output of your database.

Lap Tests

Task 1: Create database table

Task 2: create HTML form.

Task: Connect webpage with the DB

Unit Two: Retrieve data from database and display on web pages

This unit is developed to provide you the necessary information regarding the following content coverage and topics

- Data retrieval using structured query language (SQL)
- Data display stored in database
- Display format data in database

This unit will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- Perform Retrieve data stored in database
- display data stored in database
- display format data stored in database

2.1. Retrieve data using structured query language (SQL)

Structured Query Language (SQL) is a database language designed for managing data held in a relational database management system. SQL was initially developed by IBM in the early 1970s (Date 1986). The initial version, called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's quasi-relational database management system, System R. Then in the late 1970s, Relational Software Inc., which is now Oracle Corporation, introduced the first commercially available implementation of SQL, Oracle V2 for VAX computers.

SQL or structured query language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).It is based on upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. The SQL language is subdivided into several language elements, including:

- **Clauses**, which are constituent components of statements and queries.
- **Expressions**, which can produce either scalar values, or tables consisting of columns and rows of data.
- **Predicates**, which specify conditions that can be evaluated to SQL three-valued logic (true/false/unknown) or Boolean truth values and are used to limit the effects of statements and queries, or to change program flow.
- **Queries**, which retrieve the data based on specific criteria. This is an important element of SQL.
- **Statements**, which may have a persistent effect on schemata and data, or may control transactions, program flow, connections, sessions, or diagnostics.

SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.

Step 1: Create a PHP page to save data from HTML form to your MySQL database

Step 2: All done! Browse your webpage.

Step 1: Create a PHP page to save data from HTML form to your MySQL database

The contact HTML form action is on “contact.php” page. On this page, you can continue to write code on the contact.php which was created for connecting the database with website, now we insert records into the database.

Inserting data into your database from a form

HTML form: First we create an HTML form that need to take user input from keyboard. HTML form is a document which stores information of a user on a web server using interactive controls. An HTML form contains different kind of information such as username, password, contact number, email id etc.

The elements that are used in an HTML form are check box, input box, and radio buttons, submit buttons etc. With the help of these elements, the information of a user is submitted on the web server. The form tag is used to create an HTML form.

Syntax:

```
<form> Form Elements... </form>
```

or

To pass the values to next page, we use the page name with the following syntax. We can use either GET or POST method to sent data to server.

```
<form action=other_page.php  method= POST/GET>
   Form Elements...
</form>
```

2.2. Display data with web page

All done! Browse your webpage.

Now the coding part is done. If you would like to check then you can fill the form http://localhost/contact/contact.html and see the result in the database. You may check via phpmyadmin your inserted record and also you can format styles or reorganize your webpages if the outlooks are not full enough.

Other template based webpages can show a list of open datasets and their various feature types. These components are displayed as a hierarchy of levels that can be turned on or off, from entire datasets down to individual feature types. Data List and Repeater controls are used to alter the layout of data source records in the data List, common master/details scenarios, ways to edit and delete data, how to page through records, and so on.

The steps involved maintaining the data list and repeater controls are

- Add them to the web page
- Display information with web page
- Improve the appearance to the data list
- Exploring the data list and other templates
- Rendering specific markup with repeater control
- Improving appearance of repeater

2.3. Display format data

How data should be formatted depends on the data itself. For example, when listing products we might want to display the product information in a light gray font color if it is discontinued, or we may want to highlight the Units In Stock value if it is zero. The Grid View, Details View, and Form View offer distinct ways to format their appearance based on their data, one of which is Formatting Functions in Templates when using Template Fields in the details View or Grid View controls, or a template in the Form View control, we can add a formatting function to the ASP.NET page s code-behind class, the Business Logic Layer, or any other class library that is accessible from the web application. This formatting function can accept an arbitrary number of input parameters, but must return the HTML to render in the template.

Self-check 2

Part-I: Fill the blank space

1. ______________________________ a database language designed for managing data held in a relational database management system.
2. ____________________________retrieve the data based on specific criteria
3. The elements that are used in an HTML form are ________________, ____________, ____________etc.
4. DDL stands for __________________
5. DQL Stands for_________________

Part-II: Answer the following questions accordingly

1. List at least four different browsers that you might use to test a web application
2. Explain the different types of web programming concepts;

   Hypertext Transfer Protocol (HTTP)

   Authentication and Web Security
3. Describe what is a database structure?
4. What does SQL stand for?

Operation sheet 2.1: PHP Database connection

Operation Title: connect Database with PHP

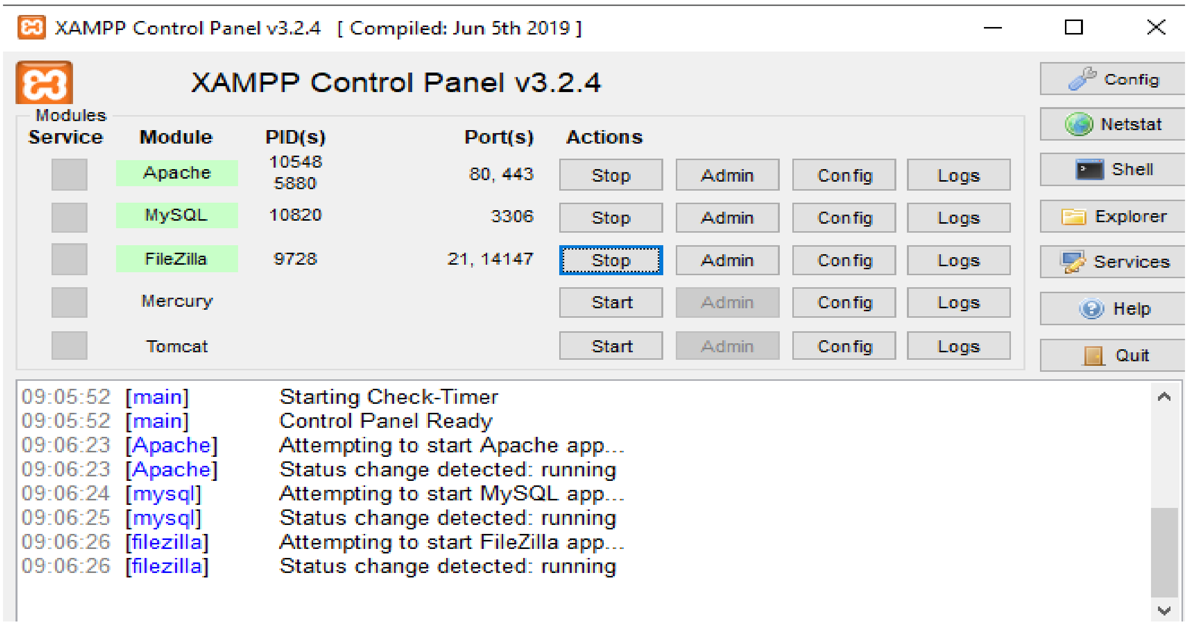
Purpose: establish a connection and interact with the database.

Requirements: XAMPP web server procedure:

- ✓ Computer
- ✓ Internet
- ✓ database

Steps in Detail:

1. Open XAMPP and start running Apache, MySQL and FileZilla



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions			
	Apache	10548 5880	80, 443	Stop	Admin	Config	Logs
	MySQL	10820	3306	Stop	Admin	Config	Logs
	FileZilla	9728	21, 14147	Stop	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
09:05:52  [main]      Starting Check-Timer
09:05:52  [main]      Control Panel Ready
09:06:23  [Apache]    Attempting to start Apache app...
09:06:23  [Apache]    Status change detected: running
09:06:24  [mysql]     Attempting to start MySQL app...
09:06:25  [mysql]     Status change detected: running
09:06:26  [filezilla] Attempting to start FileZilla app...
09:06:26  [filezilla] Status change detected: running
```

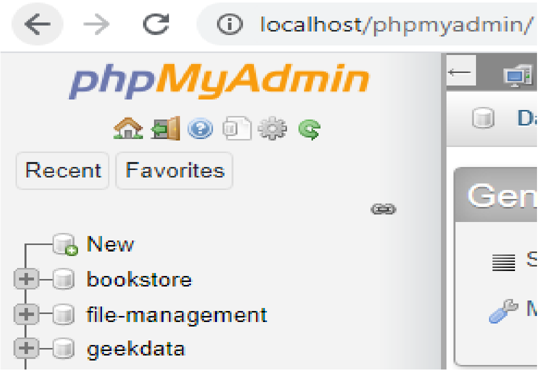
Now open your PHP file and write your PHP code to create database and a table in your database.

PHP code to create a database:

```
<?php

// Server name must be localhost

$servername = "localhost";

// In my case, user name will be root

$username = "root";
```

```
// Password is empty

$password = "";

// Creating a connection

$conn = new mysqli($servername,

            $username, $password);

// Check connection

if ($conn->connect_error) {

    die("Connection failure: "

        . $conn->connect_error);

}

// creating a database named geekdata

$sql = "CREATE DATABASE geekdata";

if ($conn->query($sql) === TRUE) {

    echo "Database with name geekdata";

} else {

    echo "Error: " . $conn->error;

}

// Closing connection

$conn->close ();

?>
```

Finally the database is created and connected to PHP.



If you want to see your database, just type localhost/phpmyadmin in the web browser and the database can be found.

After connection database you need to take post variable from the form. See the below code

$txtName = $_POST['txtName'];

$txtEmail = $_POST['txtEmail'];

$txtPhone = $_POST['txtPhone'];

$txtMessage = $_POST['txtMessage'];

When you will get the post variable then you need to write the following SQL command.

$sql = "INSERT INTO `tbl_contact` (`Id`, `fldName`, `fldEmail`, `fldPhone`, `fldMessage`) VALUES ('0', '$txtName', '$txtEmail', '$txtPhone', '$txtMessage');"

For fire query over the database, you need to write the following line

$rs = mysqli_query($con, $sql);

Here is PHP code for inserting data into your database from a form.

<? Php

// database connection code

// $con = mysqli_connect('localhost', 'database_user', 'database_password','database');

$con = mysqli_connect('localhost', 'root', '','db_contact');

```
// get the post records

$txtName = $_POST['txtName'];

$txtEmail = $_POST['txtEmail'];

$txtPhone = $_POST['txtPhone'];

$txtMessage = $_POST['txtMessage'];

// database insert SQL code

$sql = "INSERT INTO `tbl_contact` (`Id`, `fldName`, `fldEmail`, `fldPhone`, `fldMessage`) VALUES ('0', '$txtName', '$txtEmail', '$txtPhone', '$txtMessage')";

// insert in database

$rs = mysqli_query($con, $sql);

if($rs)

{

        echo "Contact Records Inserted";

}

?>
```

Quality Criteria: The web page should display database data and information about variables of different data types.

Lap Tests

Task one: Create a database for the cosmetics website**.**

Task two: Use data list and repeater to maintain the data in the cosmetics webpage database

Task three: Format the data the cosmetics webpage database to display the data in a good appearance

Unit Three: Database user input manipulation

This unit is developed to provide you the necessary information regarding the following content coverage and topics

- Updating existing data stored in the database
- Insertion of data in the database
- Deletion of data in the database
- Check error and validation of data

This unit will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- Understand and update existing data stored in the database
- Insert data in the database.
- Delete data in the database
- Check error and validate of data

3.1 Update existing data stored in the database

The SQL data manipulation language (DML) is used to query and modify database data.

How to use the SELECT, INSERT, UPDATE, and DELETE SQL DML command statements, defined below.

SELECT: to query data in the database

INSERT: to insert data into a table

UPDATE: to update data in a table

DELETE: to delete data from a table

In the SQL DML statement:

- Each clause in a statement should begin on a new line.
- The beginning of each clause should line up with the beginning of other clauses.
- If a clause has several parts, they should appear on separate lines and be indented under the start of the clause to show the relationship.
- Upper case letters are used to represent reserved words.
- Lower case letters are used to represent user-defined words.

The **UPDATE** statement lets the database system know that you wish to update the records for the table specified in the table_name parameter

The columns that you want to modify are listed after the **SET** statement and are equated to their new updated values. Commas separate these columns

The condition in the **WHERE clause** dictates which rows from the mentioned columns will be updated.

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information. In computer programming, create, read, update, and delete are the four basic functions of persistent storage

Most application programs need to be connected to the database in order to do some basic operations like saving and retrieving the user details / data.

```
<? php
            $conn = mysql_connect('localhost','root',");
           $db   = mysql_select_db ('database_name', $conn);
?>
```

The code above will establish the connection to MySQL Sever. The UPDATE statement is used to update existing records in a table, syntax for the same is

The Syntax for SQL UPDATE Command

```
     UPDATE table_name
     SET column1=value, column2=value2...
WHERE some_column=some_value
```

In the syntax shown below we see that the UPDATE procedure requires the **SET** and **WHERE** definitions to pinpoint the changes.

An example program to implement the update query is as below:

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


// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE contact SET phone='091234' WHERE name='ABC';
 if ($conn->query($sql) === TRUE) {     echo "Record updated successfully";
} else {
   echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

3.2 Insertion of data in the database.

After a database and a table have been created, we can start adding data in them. The **INSERT INTO** statement is used to add new records to a MySQL table.

- **The syntax to insert data in the database**

Insert Into table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)

If a column is AUTO_INCREMENT (like the "id" column) or TIMESTAMP (like the "reg_date" column), it is no need to be specified in the SQL query; MySQL will automatically add the value.

- **An example program to input data using a PHP language is as below**:

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
//    Check    connection    if
($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if    ($conn->query($sql)    ===    TRUE)    {
echo "New record created successfully";
} else {
   echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

Multiple SQL statements must be executed with the mysqli_multi_query() function.

Example for the same is below:

```
<? php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection if
($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname,
email) VALUES ('Julie', 'Dooley', 'julie@example.com')"; if
($conn->multi_query($sql) === TRUE) {
   echo "New records created successfully";
} else {    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close ();
?>
```

3.3 Deletion of data in the database

The DELETE statement is used to delete records from a table.

Syntax is Delete the data

DELETE FROM table_name WHERE some_column = some_value

Example:

```
<? php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
//    Check    connection    if
($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if  ($conn->query($sql)   ===   TRUE)   {
echo "Record deleted successfully";
} else {
   echo "Error deleting record: " . $conn->error;
}
$conn->close();
?>
```

3.4 check error and validation

PHP is a great scripting language that allows many dynamic functions in your site.

Consider the below example:

```
$query = mysql_query("SELECT username FROM Users WHERE username=$username",
$con);
  if (mysql_num_rows($query) != 0)
  {
     echo "Username already exists";
  }
  else
  {
    ...
  }
```

The table name is 'Users' and the column to be searched is 'username'. The input from form was read into $username and I verified that it was read in properly using echo. $con contains the connection to the server.

Self-check 3

Part-I: Answer the following questions accordingly

1. How to select unique records from a table?
2. What is PHP most used for?
3. What are the data types present in PHP?
4. What is join in SQL?

Part-II: Fill the blank space

1. _______________________command to query data in the database
2. ________________________ command to insert data into a table
3. _________________________ command to insert data into a table
4. _________________________ command to delete data from a table

Reference

Books

Learning PHP, MySQL & JavaScript: With jury, CSS & HTML5 Robin Nixon, 2015, 5th Edition

Learning Web Design Jennifer Niederst Robbins, Fourth Edition,2012

Learning PHP, MySQL, JavaScript, and CSS By Robin Nixon • 2012

PHP and MySQL Web Development: Second Edition By Luke Welling, Laura Thomson

Learning PHP 7 by Antonio Lopez • 2016

URL

https://www.geeksforgeeks.org/what-is-database/

https://www.javatpoint.com/what-is-databasehttps://www.techtarget.com/searchdatamanagement/definition/database

https://www.britannica.com/technology/database

https://opentextbc.ca/dbdesign01/chapter/chapter-sql-dml/

https://medium.com/@actualiyke/manipulation-of-user-input-851dabdac33e

http://localhost/contact/contact.html

Developer’s Profile

No	Name	Qualification	Field of Study	Organization/ Institution	Mobile number	E-mail
1	Frew Atkilt	M-Tech	Network & Information Security	Bishoftu Polytechnic College	0911787374	frew.frikii@gmail.com
2	Gari Lencha	MSc	ICT Managment	Gimbi Polytechnic	0917819599	Garilencha12@gmail.com
3	Kalkidan Daniel	BSc	Computer Science	Entoto Polytechnic	0978336988	kalkidaniel08@gmail.com
4	Solomon Melese	M-Tech	Computer Engineering	M/G /M /Polytechnic College	0918578631	solomonmelese6@gmail.com
5	Tewodros Girma	MSc	Information system	Sheno Polytechnic College	0912068479	girmatewodiros @gmail.com
6	Yohannes Gebeyehu	BSc	Computer Science	Entoto Polytechnic College	0923221273	yohannesgebeyehu73@gmail.com