UNIT-5

Backend Programming with PHP:

1. Linux Installation
2. XAMMP Environment Setup
3. PHP Programming Fundamentals
4. PHP Data Types & Dates
5. Cookies
6. Sessions
7. Objects and Classes
8. Intro to PHP & MySQL
9. creating a MySQL Database
10. Connect & Fetch data from MySQL
11. Insert and Delete MySQL data from PHP

1. Linux Installation

Now that we know what Linux is, it is the time that to learn how we should install it on the computer and choose which Distribution we should use. Let us start by understanding what a Linux Distribution is.
we will learn –

- What is a Linux Distribution?
- How many distributions are out there?
- The Best Linux Distribution!
- Installing Linux using USB stick
- Installing Linux using CD-ROM
- Installing Linux using Virtual Machine

What is a Linux Distribution?
Well, now as you know that **Linux is open-source, free to use kernel**. It is used by programmers, organizations, profit and non-profit companies around the world to **create Operating systems to suit their individual requirements**.
To prevent hacking attempts, many organizations keep their Linux operating systems private. Many others make their variations of Linux available publicly so the whole world can benefit at large.
These versions/ types /kinds of **Linux operating system are called Distributions**.

How many distributions are out there?
There are **hundreds of Linux operating systems or Distributions** available these days. Many of them are designed with a specific purpose in mind. For example, to run a **web server or to run on network switches like routers, modems,** etc.
The latest example of one of the most popular smartphone-based **Linux Distribution is Android!**

Many of these Distributions are built to offer **excellent personal computing**.
Here, are a few popular Linux Distributions (also called Linux Distro) –

Linux Distribution	Name	Description
	Arch	This Linux Distro is popular amongst Developers. It is an independently developed system. It is designed for users who go for a do-it-yourself approach.
	CentOS	It is one of the most used Linux Distribution for enterprise and web servers. It is a free enterprise class Operating system and is based heavily on Red Hat enterprise Distro.
	**Debian**	Debian is a stable and popular non-commercial Linux distribution. It is widely used as a desktop Linux Distro and is user-oriented. It strictly acts within the Linux protocols.
	**Fedora**	Another Linux kernel based Distro, Fedora is supported by the Fedora project, an endeavor by Red Hat. It is popular among desktop users. Its versions are known for their short life cycle.
	**Gentoo**	It is a source based Distribution which means that you need to configure the code on your system before you can install it. It is not for Linux beginners, but it is sure fun for experienced users.
	**LinuxMint**	It is one of the most popular Desktop Distributions available out there. It launched in 2006 and is now considered to be the fourth most used Operating system in the computing world.
	**OpenSUSE**	It is an easy to use and a good alternative to MS Windows. It can be easily set up and can also run on small computers with obsolete configurations.
	**RedHat enterprise**	Another popular enterprise based Linux Distribution is Red Hat Enterprise.It has evolved from Red Hat Linux which was discontinued in 2004. It is a commercial Distro and very popular among its clientele.

Linux Distribution	Name	Description
	Slackware	Slackware is one of the oldest Linux kernel based OS's. It is another easy desktop Distribution. It aims at being a 'Unix like' OS with minimal changes to its kernel.
	Ubuntu	This is the third most popular desktop operating system after Microsoft Windows and Apple Mac OS. It is based on the Debian Linux Distribution, and it is known as its desktop environment.

The Best Linux Distribution!

The term best is **relative. Each Linux distribution is built for a specific purpose-built to meet the demands of its target users.**

The desktop Distributions are **available for free** on their respective websites. You might want to try them one by one till you get to know which Distribution you like the most. Each one of them offers its own unique design **applications**, and **security**.

We will be using Ubuntu for our learning purpose as it's easy for a beginner to understand.

How to Install Linux

Let's look the below Linux installation guide which has various methods we can use to Download Linux(Ubuntu) and install it.

Installing Linux using USB stick

This is one of the easiest methods of installing Ubuntu or any distribution on your computer. Follow the steps to install Ubuntu from USB.

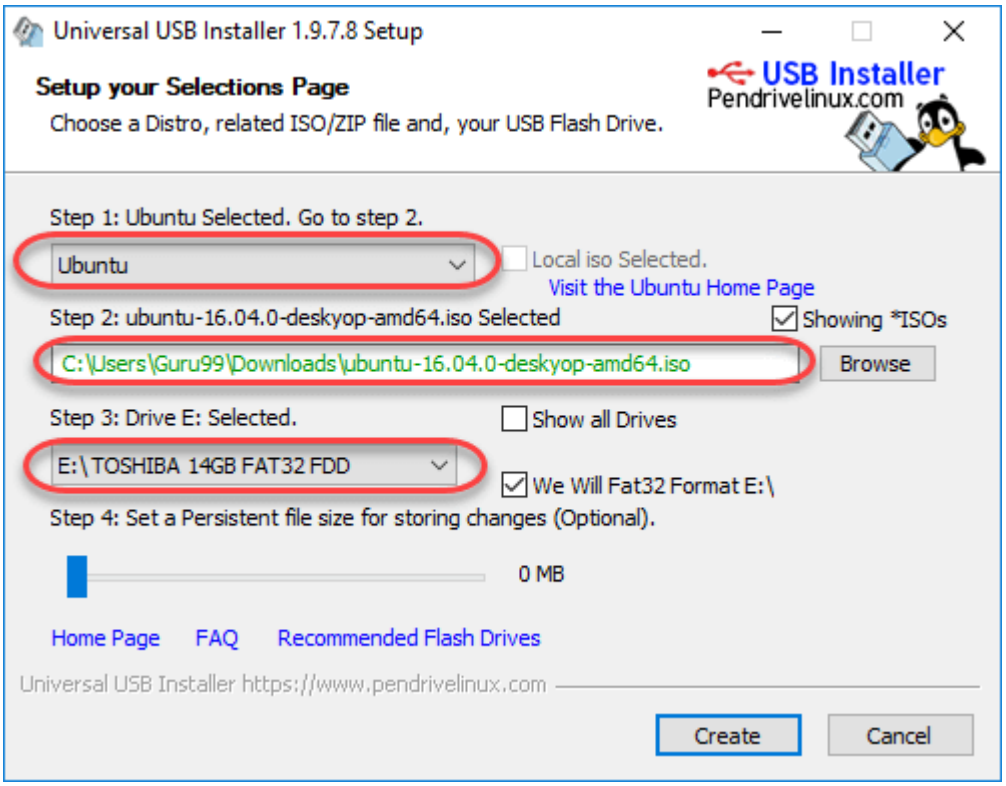
Step 1) Download required files.

Download the .iso or the OS files on your computer from this link.

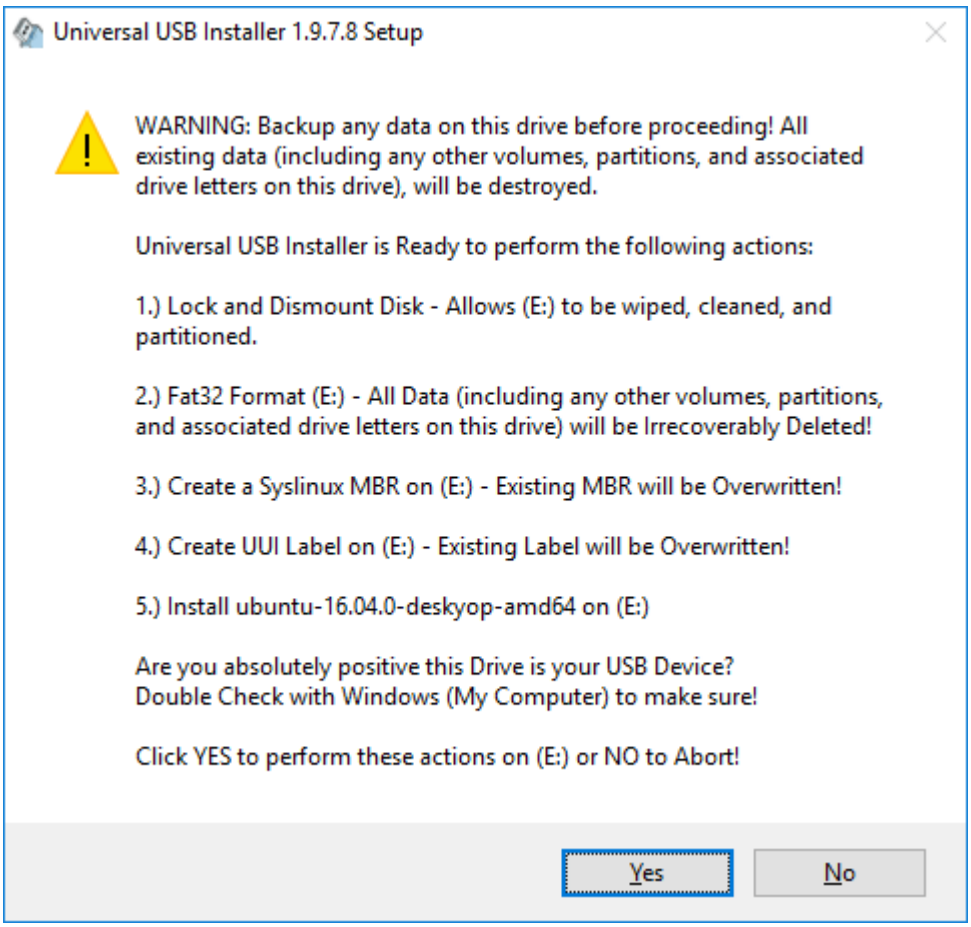
Step 2) Download Universal USB Installer.

Download free software like Universal USB installer to make a bootable USB stick.

Step 3) Select Distribution.
Select an Ubuntu Distribution form the dropdown to put on your USB
Select your Ubuntu iso file download in step 1.
Select the drive letter of USB to install Ubuntu and Press create button.



Step 4) Install Ubuntu.
Click YES to Install Ubuntu in USB.



Step 5) Check your window.
After everything has been installed and configured, a small window will appear
Congratulations! You now have Ubuntu on a USB stick, bootable and ready to go.

Installing Linux using CD-ROM

Those who like the way a CD runs should try using this method.
Step 1) Download the .iso or the OS files onto your computer from this link http://www.ubuntu.com/download/desktop.
Step 2) Burn the files to a CD.
Step 3) Boot your computer from the optical drive and follow the instructions as they come.

Installing Linux using Virtual Machine

This is a popular method to install a Linux operating system. The virtual installation offers you the freedom of running Linux on an existing OS already installed on your computer. This means if you have Windows running, then you can just run Linux with a click of a button. Virtual machine software like Oracle VM can install Linux on Windows in easy steps. Let us look at them.
Here the brief steps

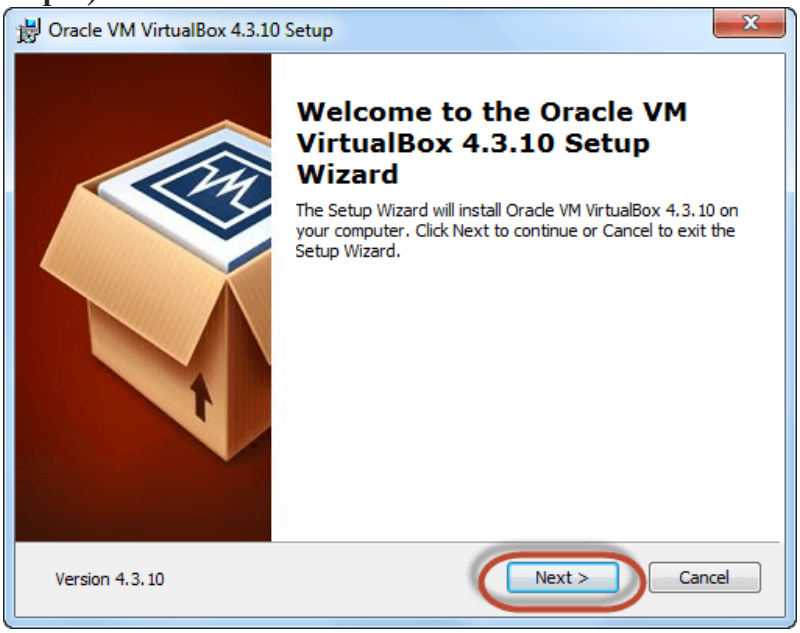
PART A) Download and Install Virtual Box

Download Virtual box using this link

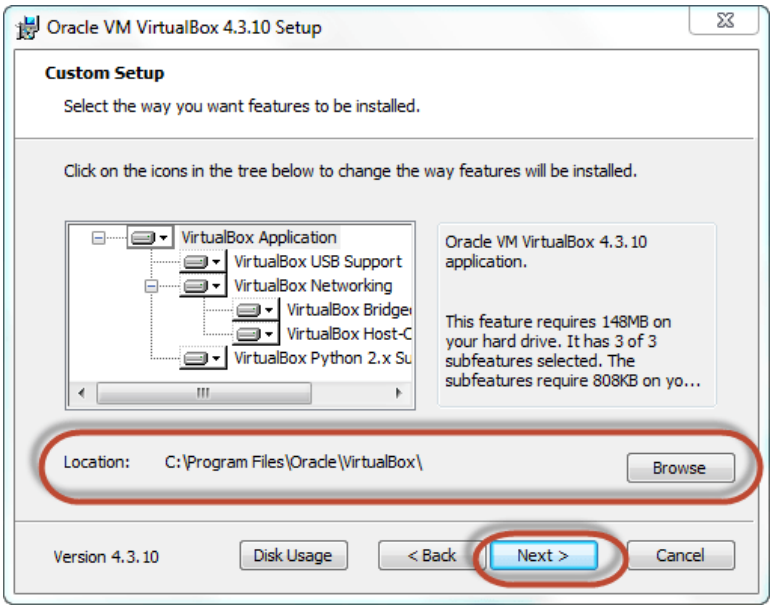
Depending on your processor and OS, select the appropriate package. In our case, we have selected Windows with AMD

Once the download is complete, Open setup file and follow the steps below:

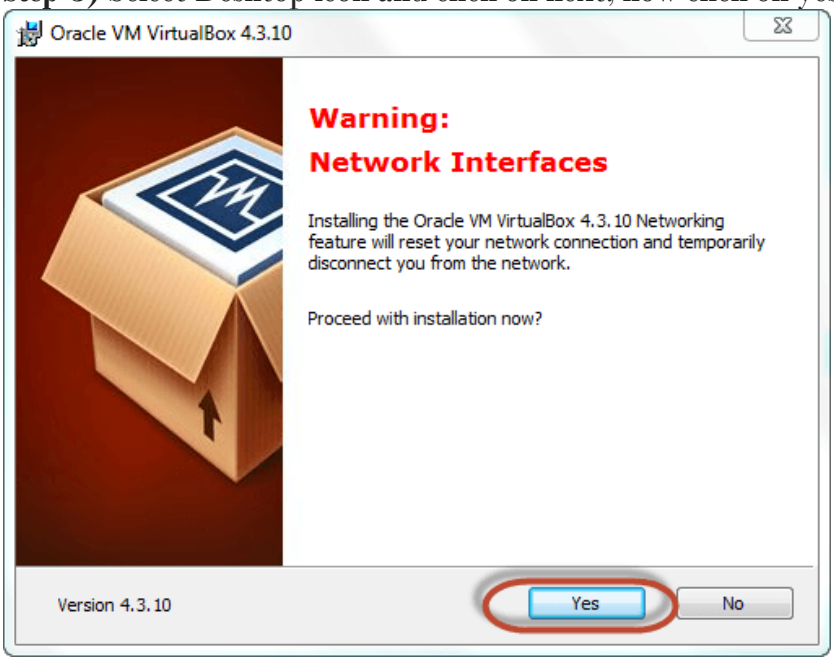
Step-1) Click On next



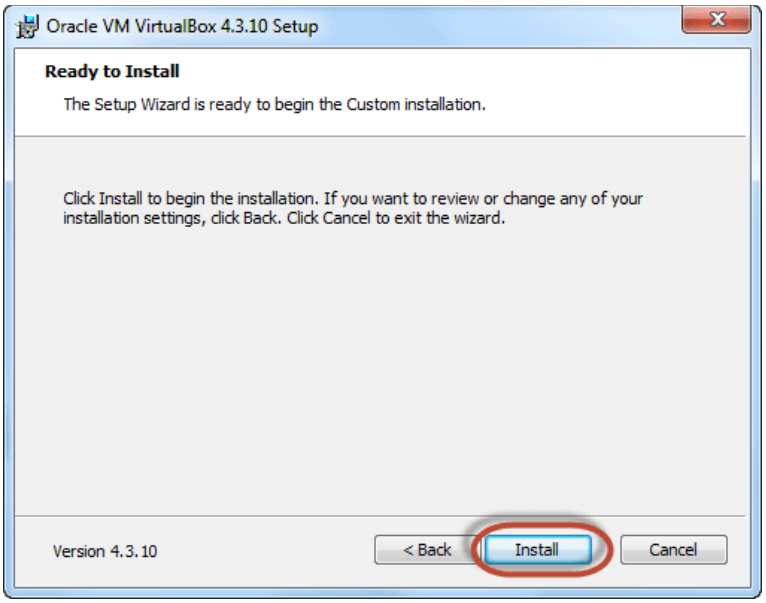
Step-2) Select you're the directory to install VirtualBox and click on next



Step-3) Select Desktop icon and click on next, now click on yes

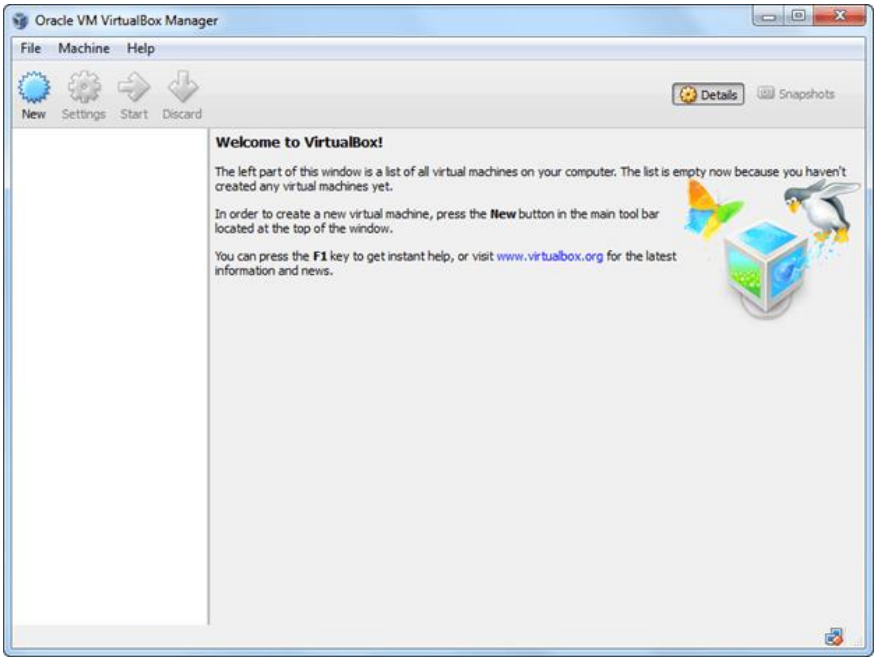


Step-4) Click On install to install Linux on Windows.



Step-5) Now installation of the virtual box will start. Once complete, click on Finish Button to start Virtual Box

The virtual box dashboard looks like this-



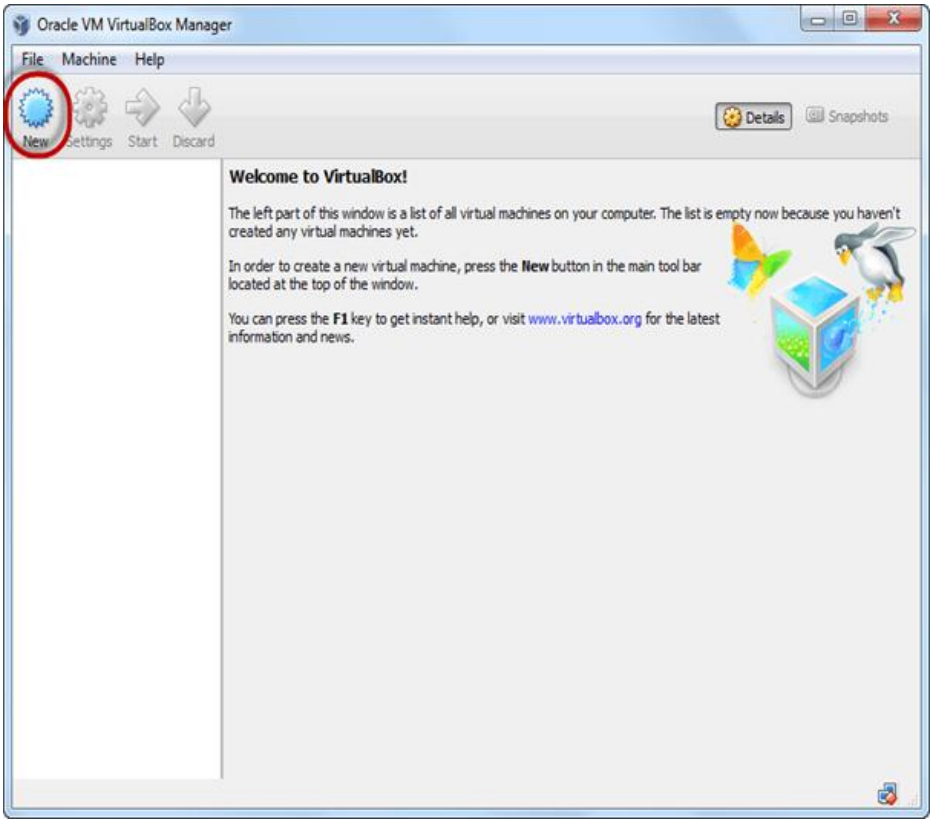
PART B) Download Ubuntu

Visit this link to download Ubuntu.

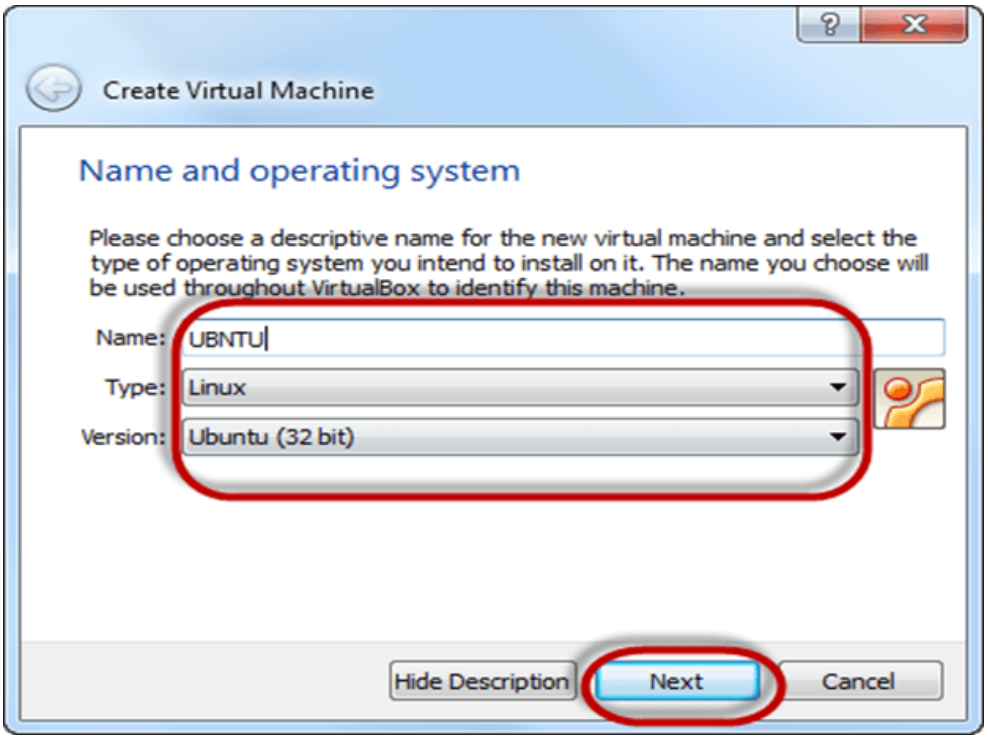
You can select 32/64-bit versions as per your choice.

PART C) Create a Machine in Virtual Box

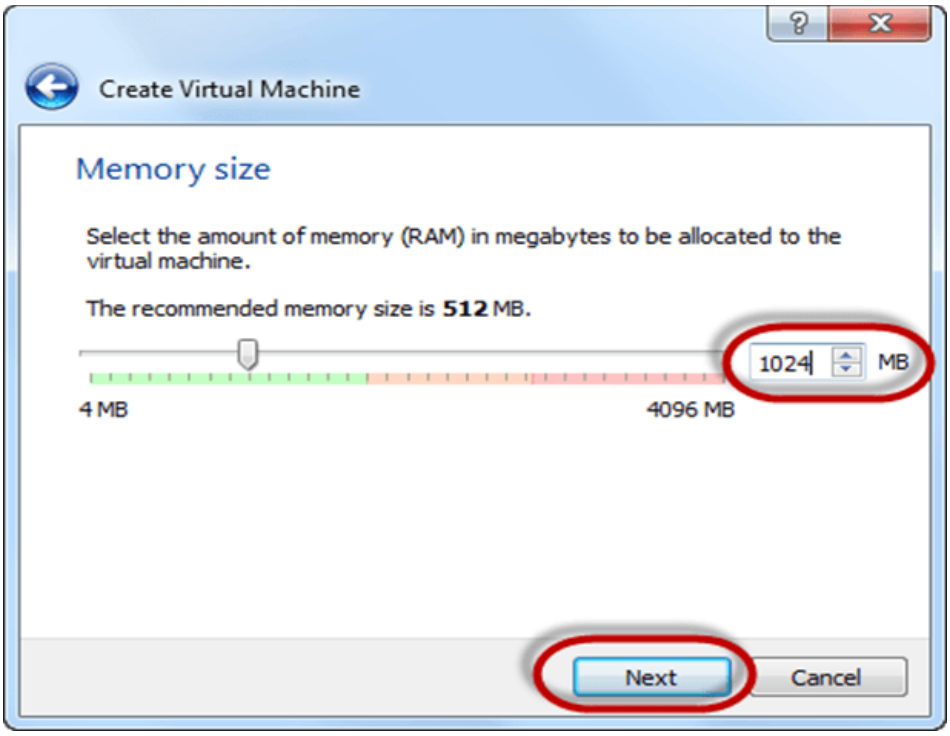
Step-1) Open Virtual box and click on new button



Step-2) In next window, give the name of your OS which you are installing in virtual box. And select OS like Linux and version as Ubuntu 32 bit. And click on next

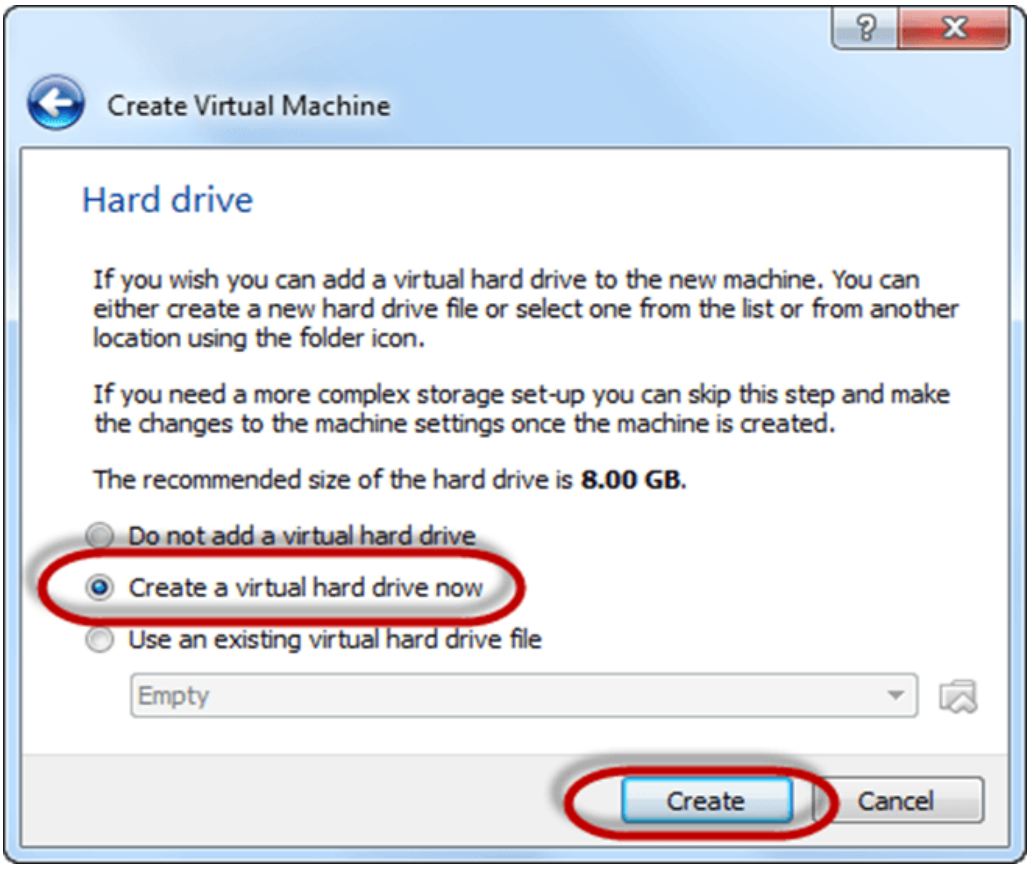


Step-3) Now Allocate Ram Size To your Virtual OS. I recommended keeping 1024mb (1 GB) ram to run Ubuntu better. And click on next.

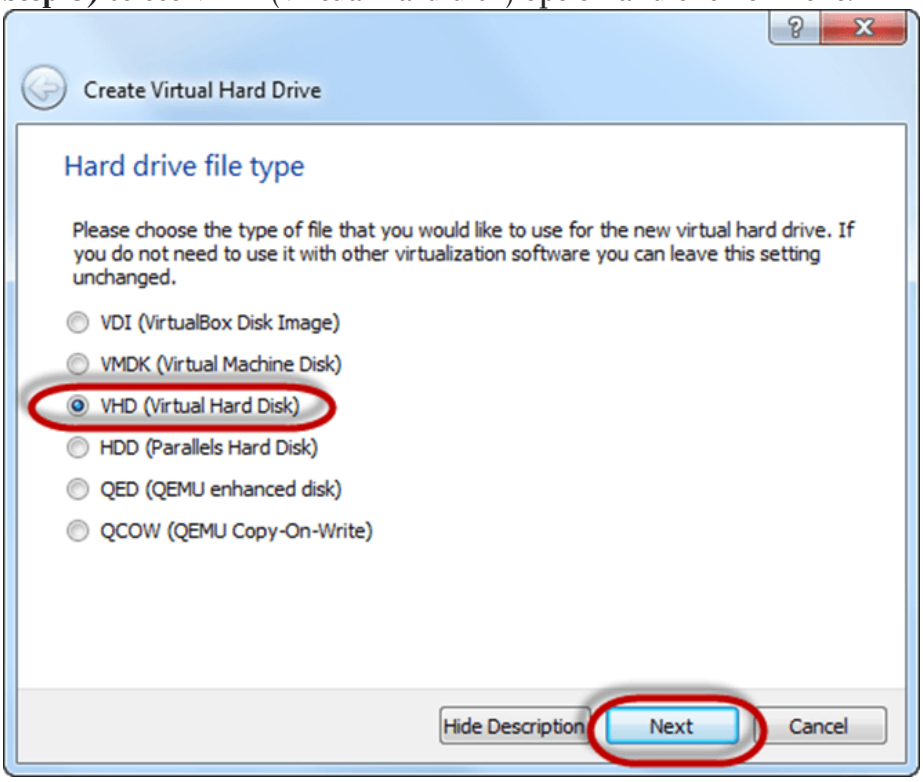


Step-4) Now To run OS in virtual box we have to create virtual hard disk, click on create a virtual hard drive now and click on create button.

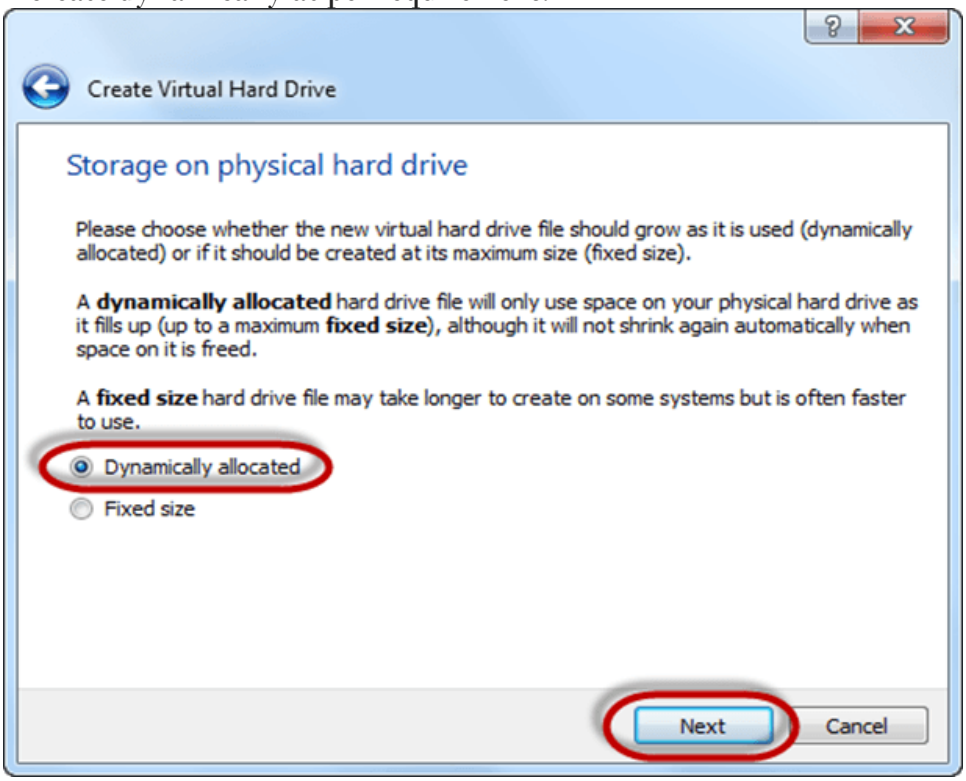
The virtual hard disk is where the OS installation files and data/applications you create/install in this Ubuntu machine will reside



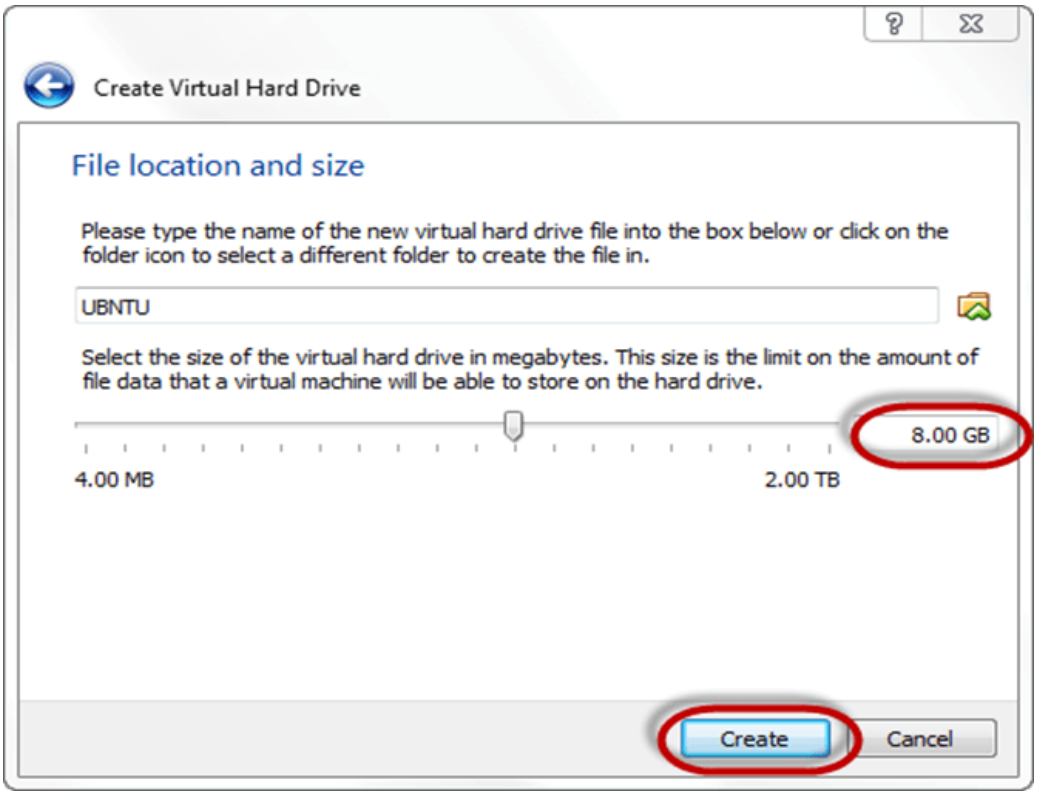
Step-5) select VHD (virtual hard disk) option and click on next.



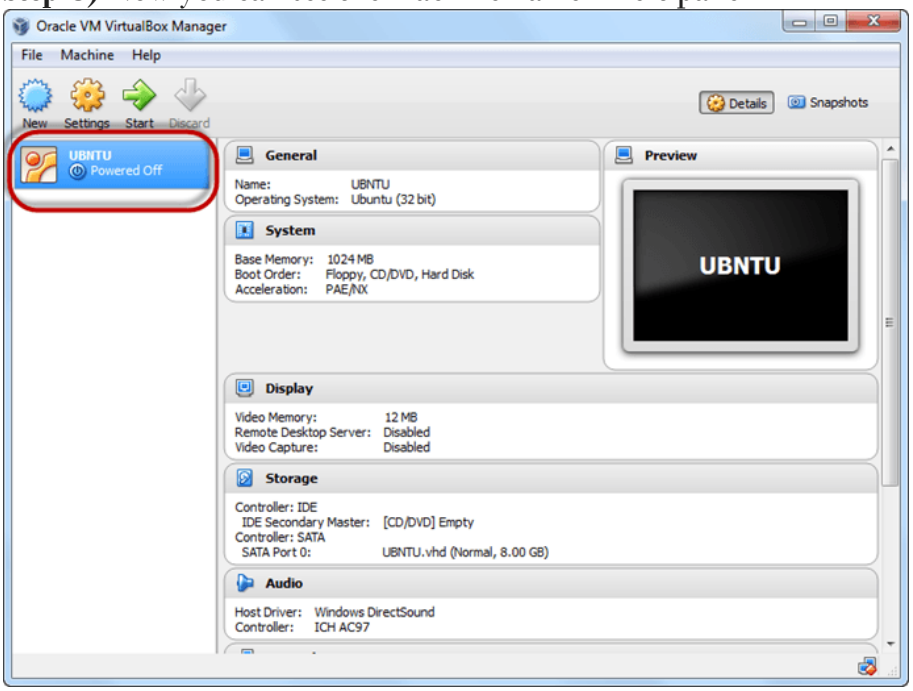
Step-6) Click on dynamic allocated and click on next. This means that the size of the disk will increase dynamically as per requirement.



Step-7) Allocate memory to your virtual hard drive .8GB recommended. Click on create button.



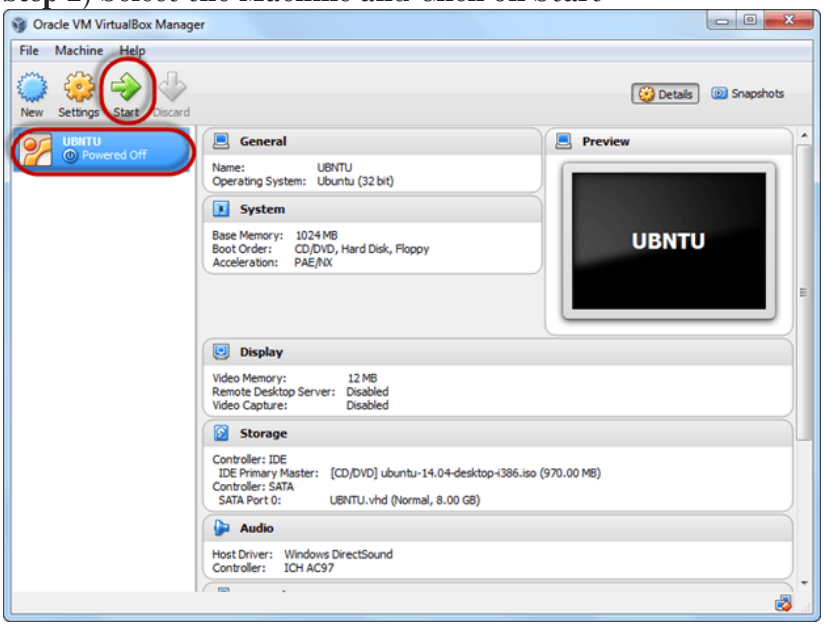
Step-8) Now you can see the machine name in left panel



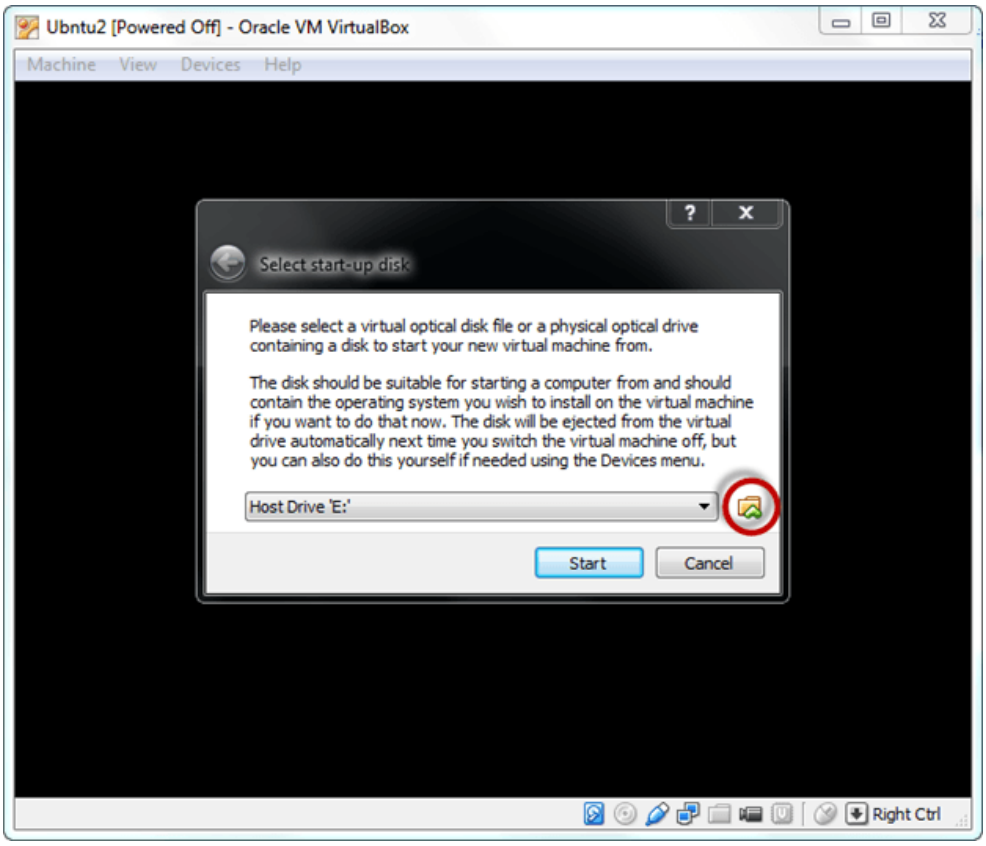
So a Machine (PC) with 8GB Hardisk, 1GB RAM is ready.

PART D) How to Install Ubuntu

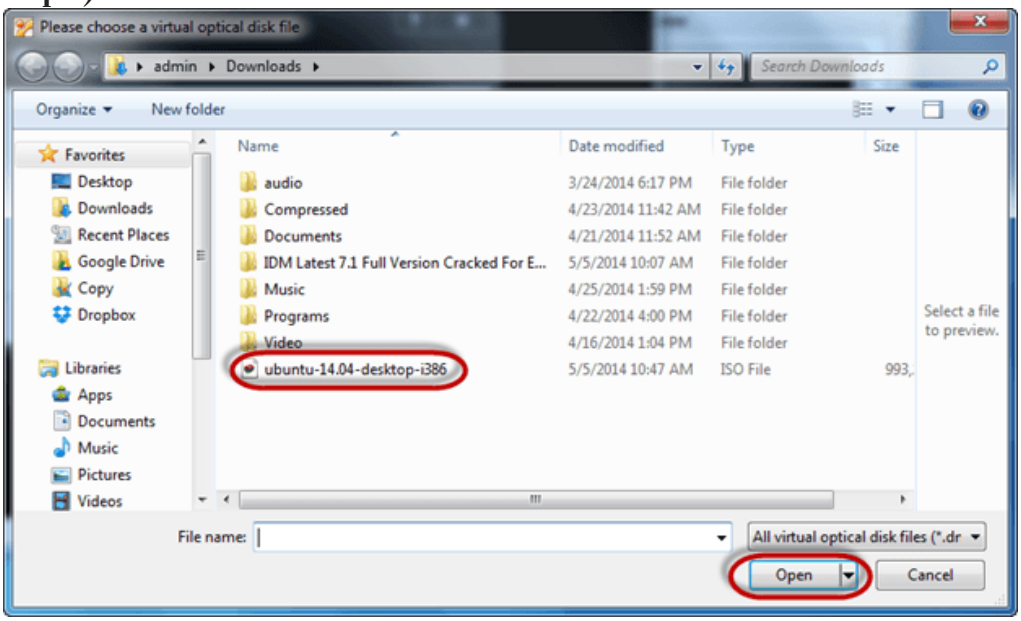
Step 1) Select the Machine and Click on Start



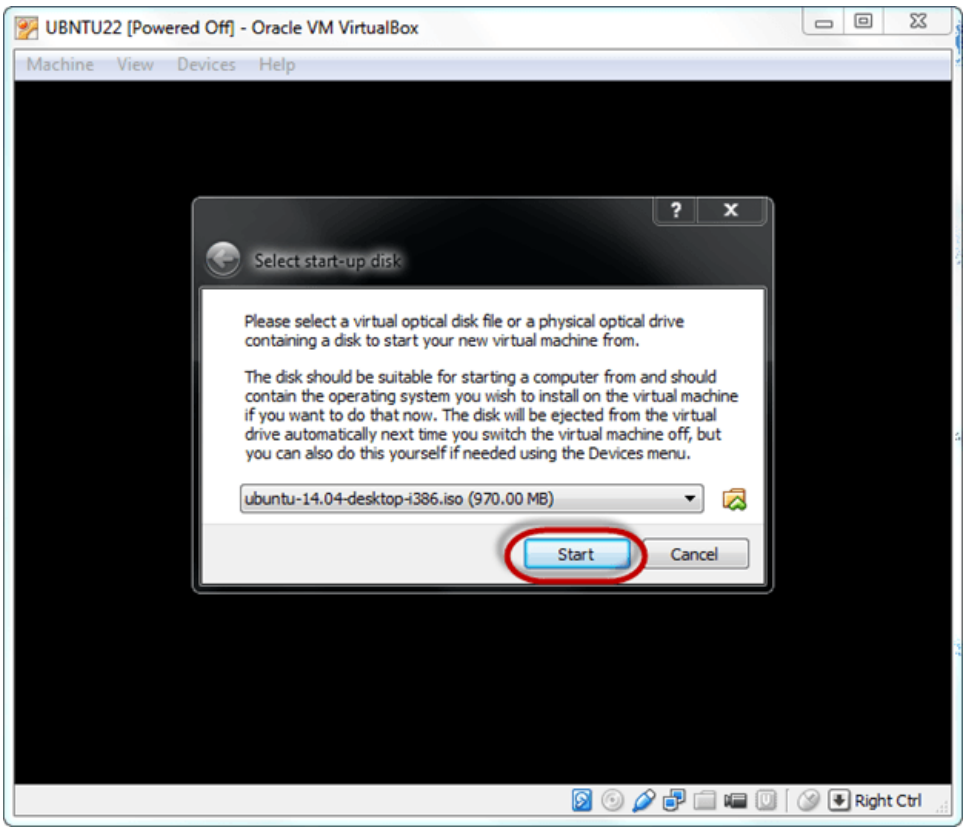
Step 2) Select the Folder Option



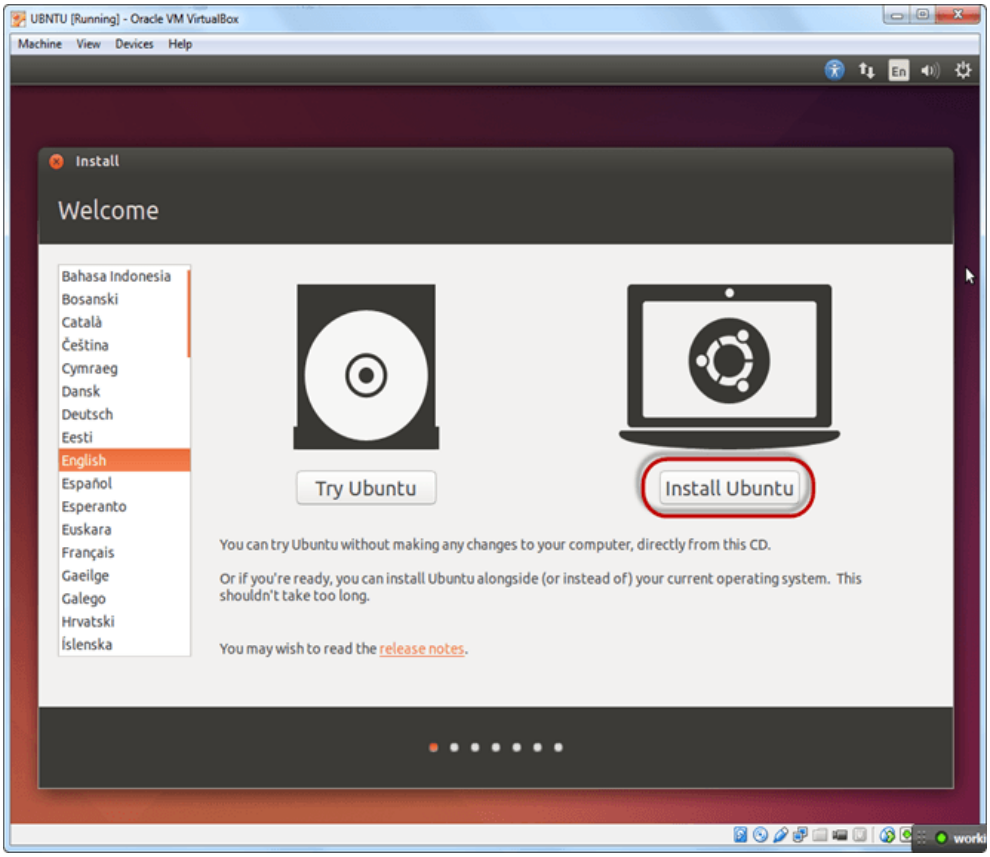
Step 3) Select the Ubuntu iso file



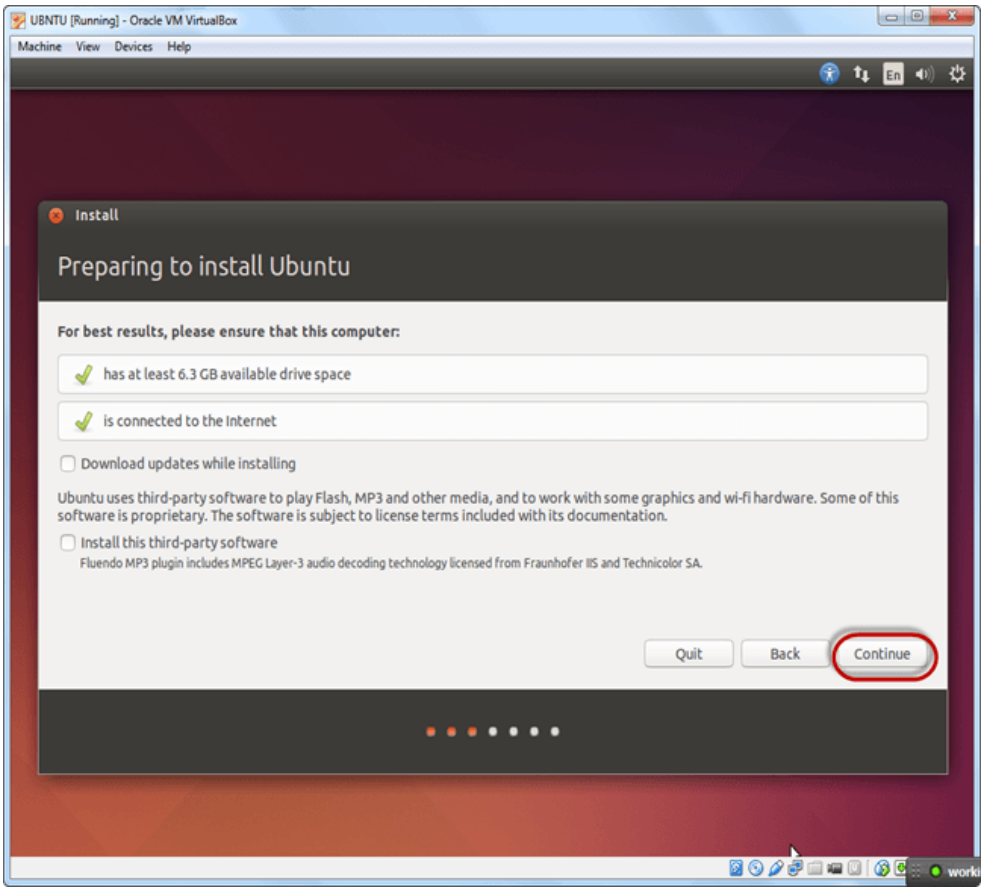
Step 4) Click Start



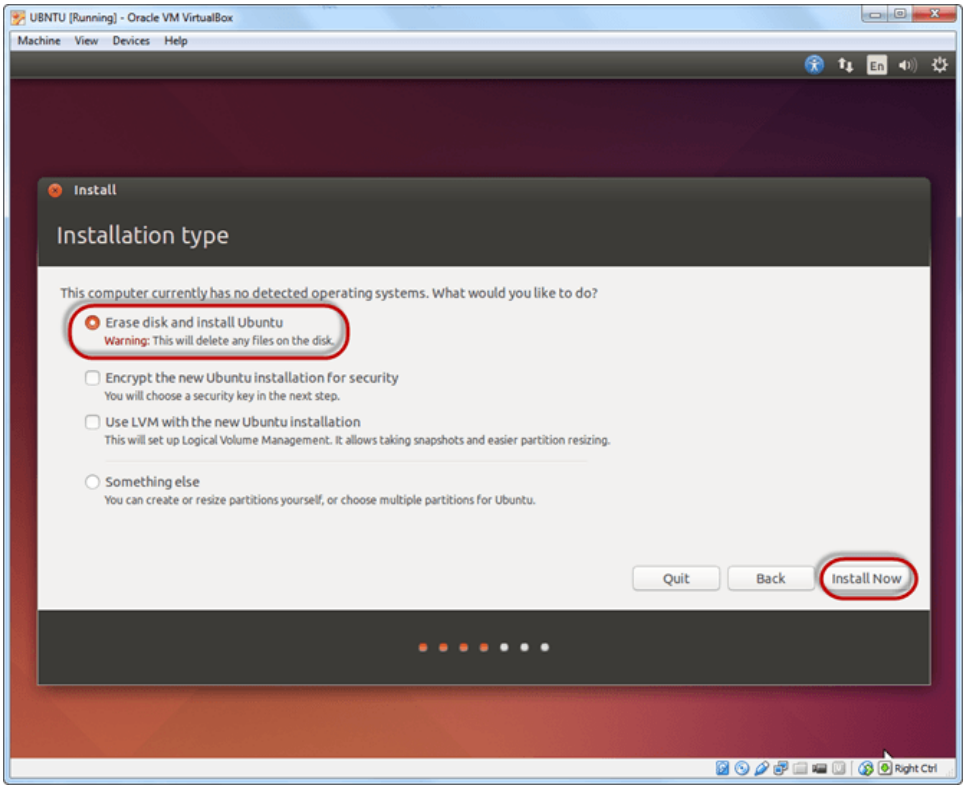
Step-5) You have an option to Run Ubuntu WITHOUT installing. In this tutorial will install Ubuntu



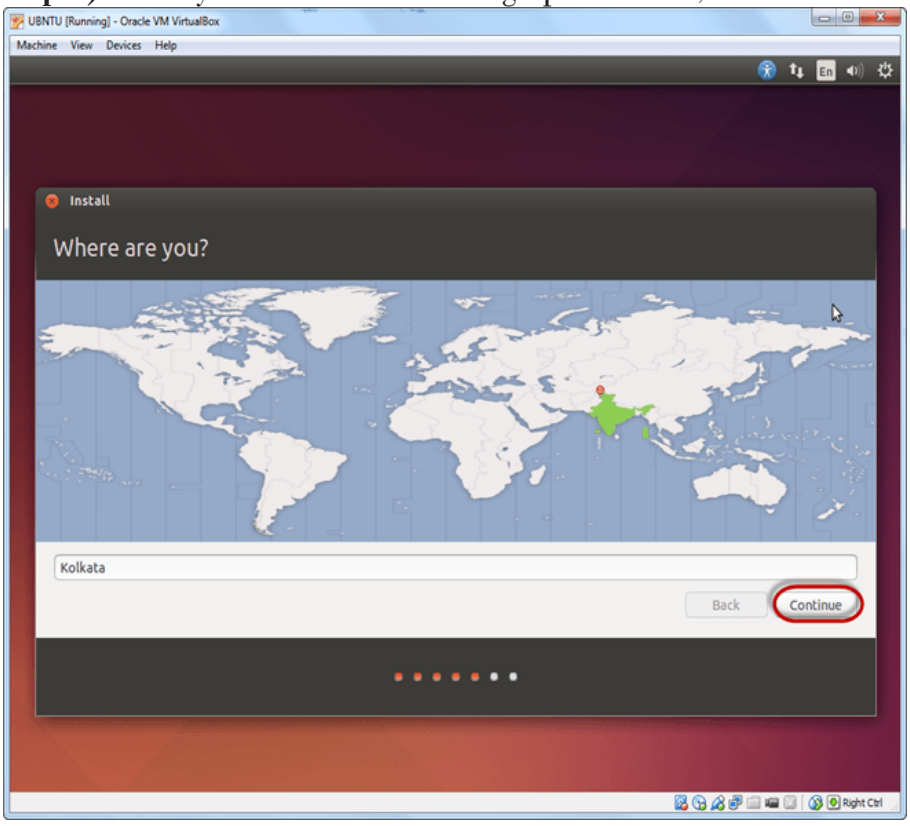
Step-6) Click continue.



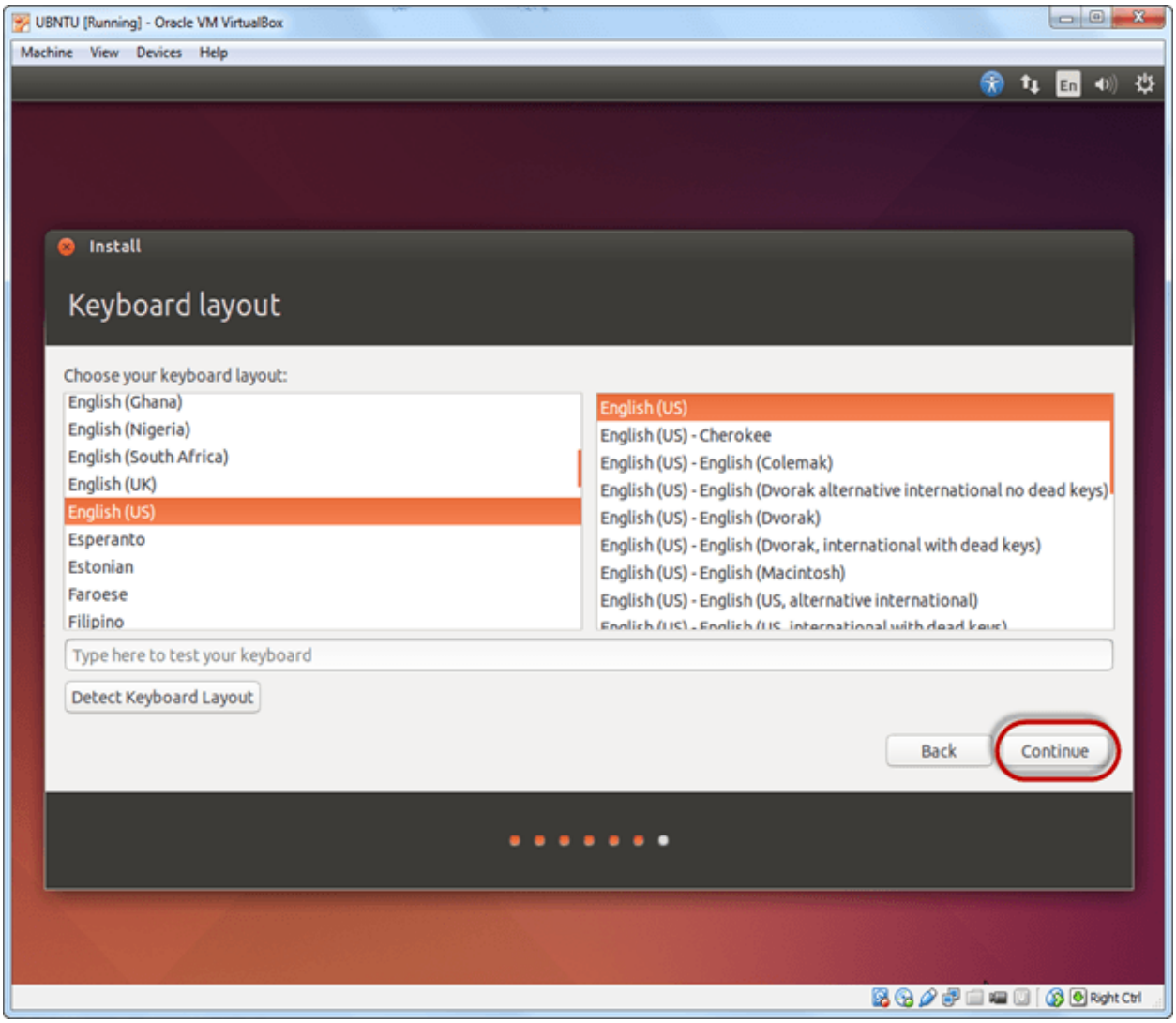
Step-7) Select option to erase the disk and install Ubuntu and click on install now. This option installs Ubuntu into our virtual hard drive which is we made earlier. It will not harm your PC or Windows installation



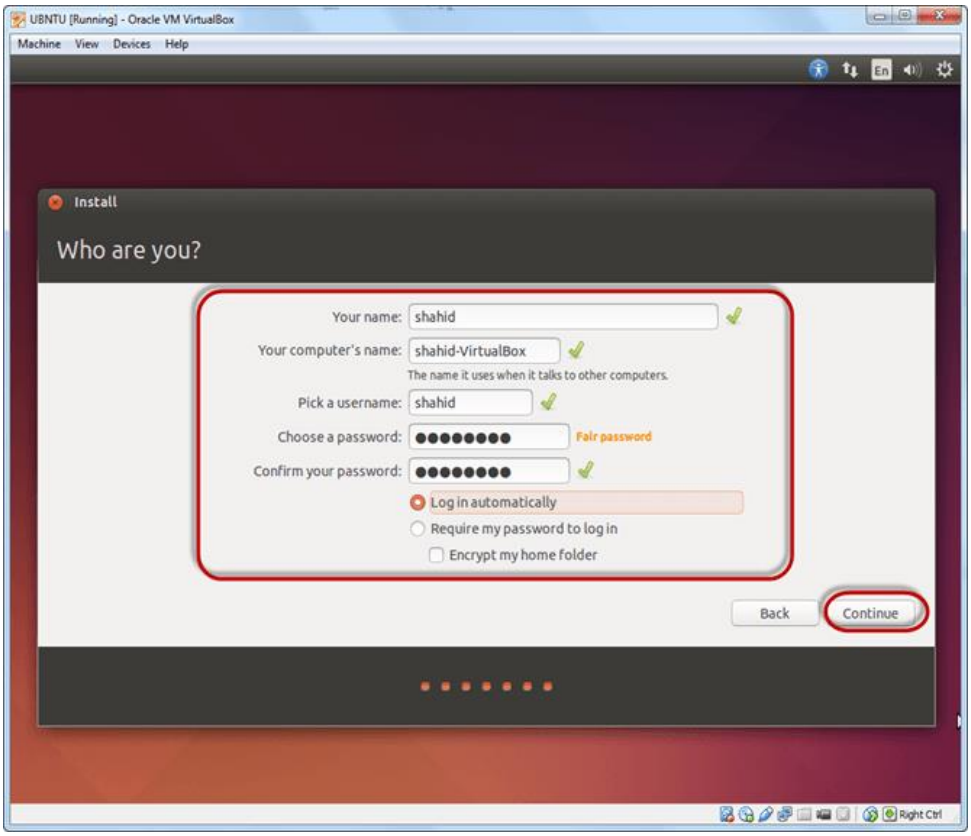
Step-8) Select your location for setting up time zone, and click on continue



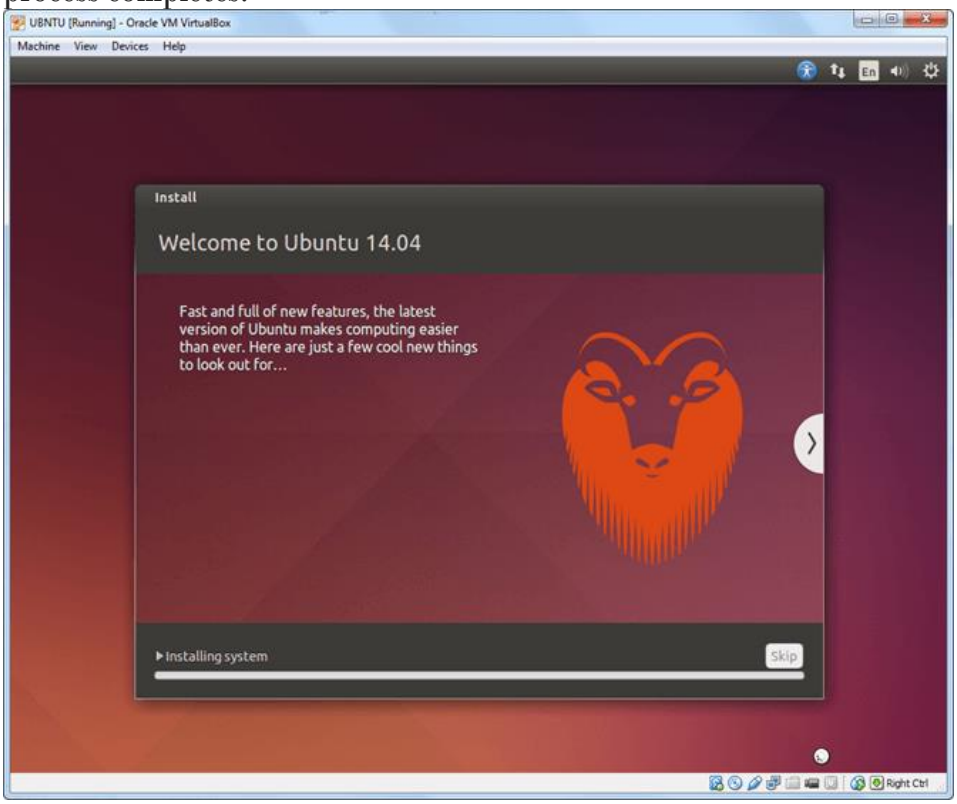
Step-9) Select your keyboard layout, by default English (US) is selected but if you want to change then, you can select in the list. And click on continue



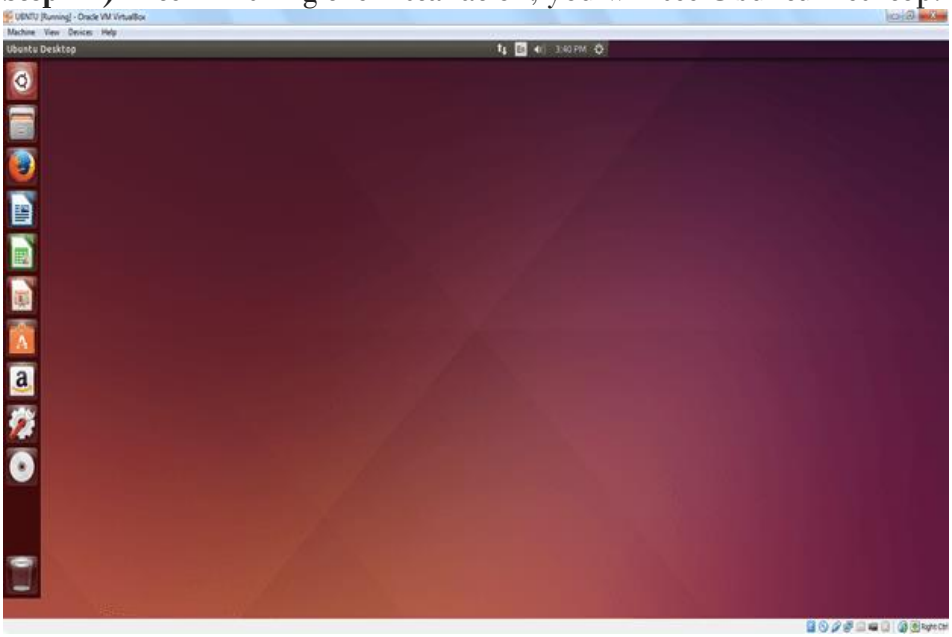
Step-10) Select your username and password for your Ubuntu admin account. This information has been needed for installing any software package into Ubuntu and also for login to your OS. Fill up your details and tick on login automatically to ignore login attempt and click on continue



Step-11) Installation process starts. May take up to 30 minutes. Please wait until installation process completes.



Step-12) After finishing the installation, you will see Ubuntu Desktop.



- An operating system based on the Linux kernel is called a Distribution or Distro
- There are hundreds of Distributions available, some of which are designed to accomplish a sole purpose like running servers, act as network switches, etc.
- Naming the best Linux Distribution is difficult as they are made for different.
- Linux can be installed on your system via the below-mentioned methods:
    - USB stick
    - Live CD
    - Virtual Installation

2. XAMMP Environment Setup

What is XAMPP?

XAMPP is an open-source, cross-platform web server that consists of a web server, MySQL database engine, and PHP and Perl programming packages. It is compiled and maintained by Apache. It allows users to create WordPress websites online using a local web server on their computer. It supports Windows, Linux, and Mac.

It is compiled and maintained by apache. The acronym XAMPP stands for;

- X – [cross platform operating systems] meaning it can run 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

Why use XAMPP?

XAMPP provides an easy-to-use control panel to manage Apache, MySQL, and other programs without using commands. To use PHP, we need to install Apache and MySQL. It's not easy to install Apache and configure it as it needs to be set up and integrated with PHP and Perl, among other things. XAMPP deals with all the complexity to set up and integrate Apache with PHP and Perl.

Unlike Java that runs with the Java SDK only, PHP requires a web-server to work.

In this XAMPP Tutorial, you will learn-

How to Install XAMPP

We look into step by step process to install XAMPP for Windows. For Other Operating Systems, XAMPP installation steps are similar.

Step 1) Download XAMPP

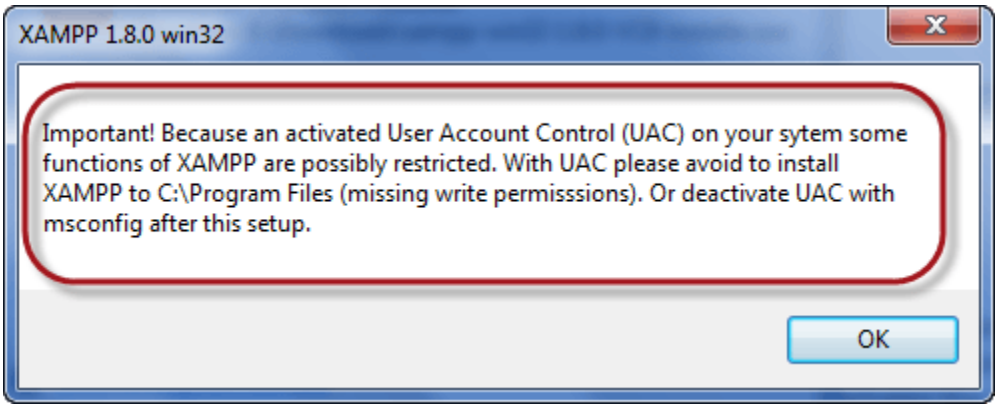
Click here to XAMPP download for Windows: https://www.apachefriends.org/download.html

Step 2) Start Installation

XAMPP Installation is just like installing any other windows program. There are however, a few things that we must note.

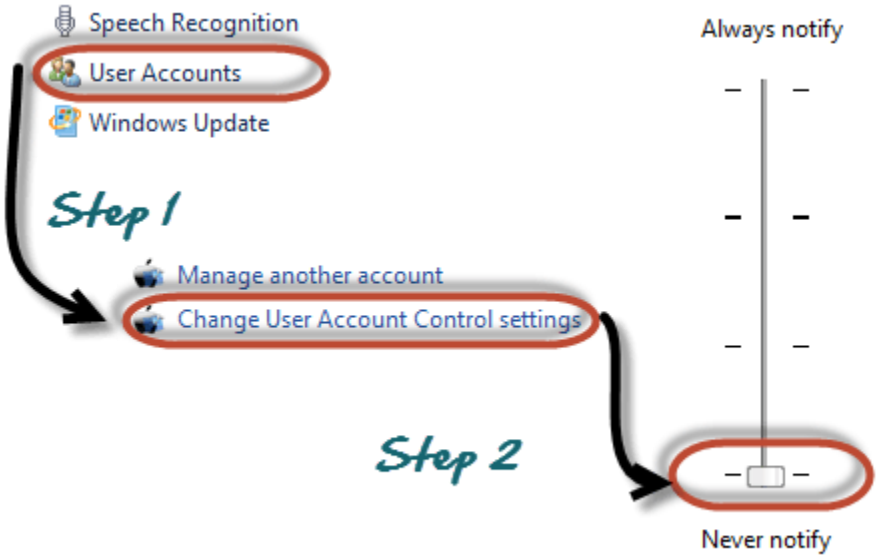
Step 3) Run the Setup

After you have downloaded XAMPP, run the setup. The warning message dialog window shown below appears.



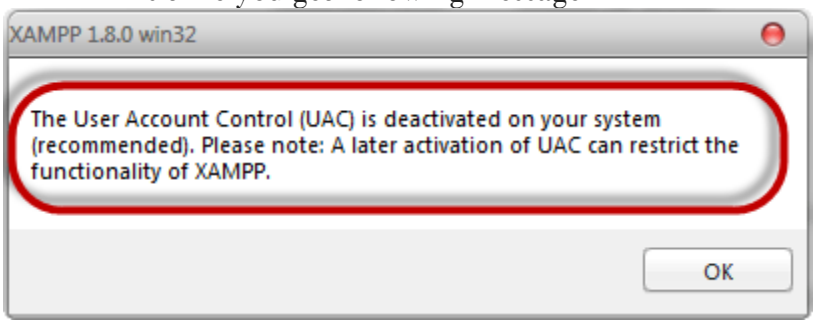
Step 4) Change User Control Settings

If you are using Windows Vista or Windows 7, make sure that you deactivate the User Account Control feature. To do this, Select Control Panel > User Accounts > Change User Access Control settings. The diagram below illustrates the main steps.



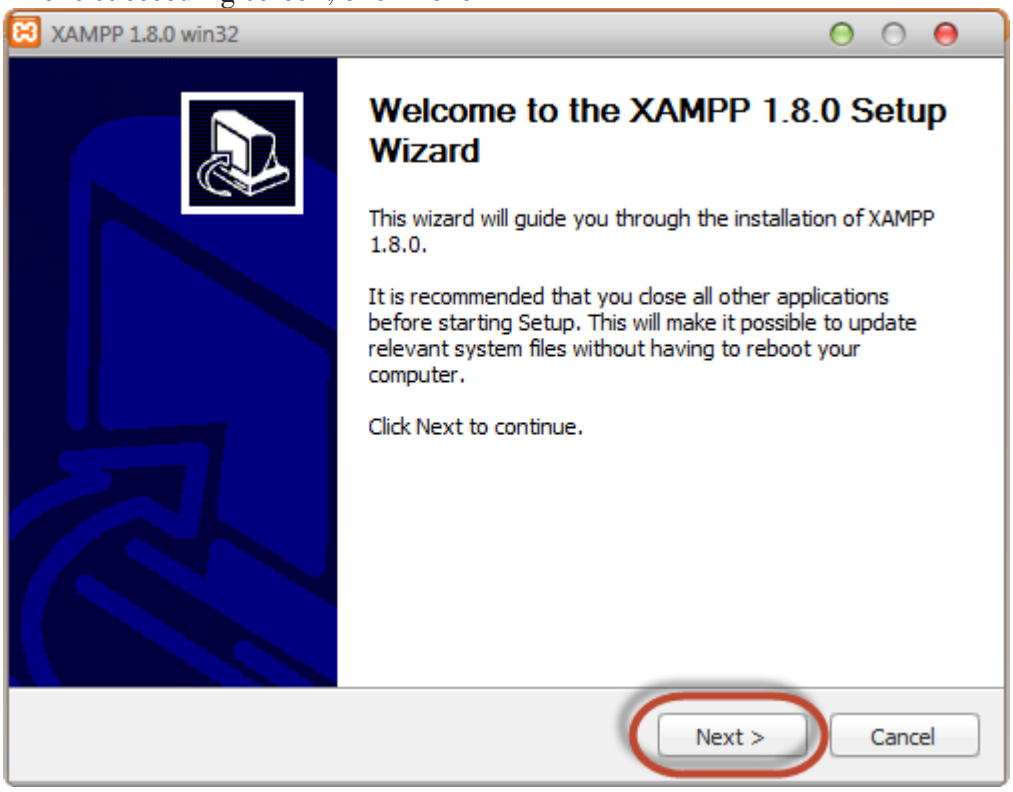
Step 5) Save the settings

- After you have deactivated the User Account Control, click on OK button on the warning message box.
- This time you get following message



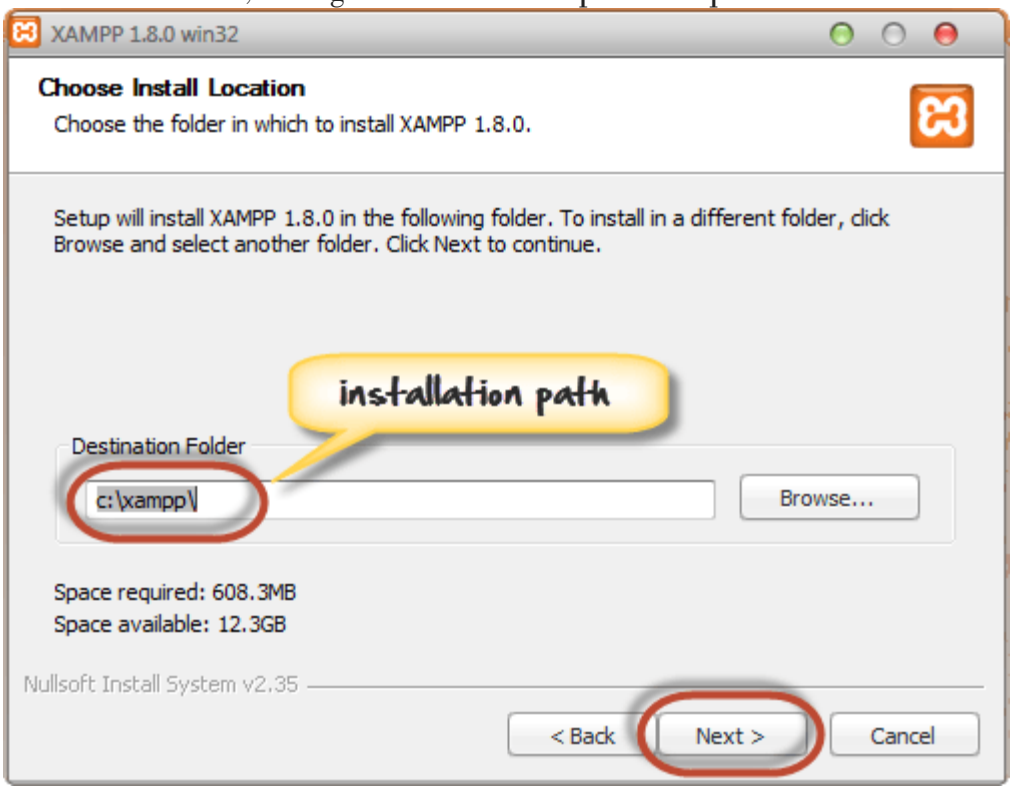
Step 6) Click Next

In the succeeding screen, click next



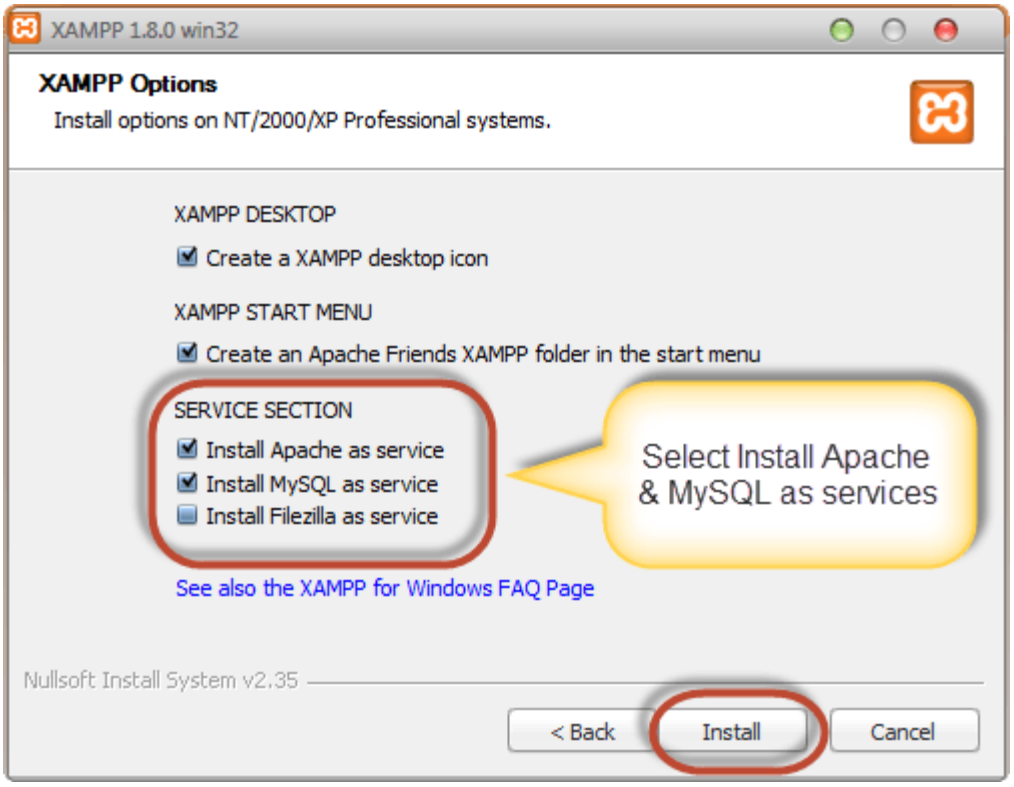
Step 7) Choose the Insatllation path

In the next screen, Change the installation path if required. Click Next



Step 8) Check the necessary services

In the next screen select Apache and MySQL. You may optionally select FileZilla (FTP Client) if needed. Click Install



***Note** a service is a long-running program in windows that does not require user intervention. Services can be set to run automatically whenever the windows operating system is started. **For you to use Apache and MySQL, they are supposed to be running in the background. Installing them as services runs both Apache and MySQL automatically in the background whenever you power up your computer**. If you have not installed Apache and MySQL as services, then you have to manually start them every time that you want to use them. You will have to do this from the XAMPP control panel.PHP and*

Step 9) Finish the installation

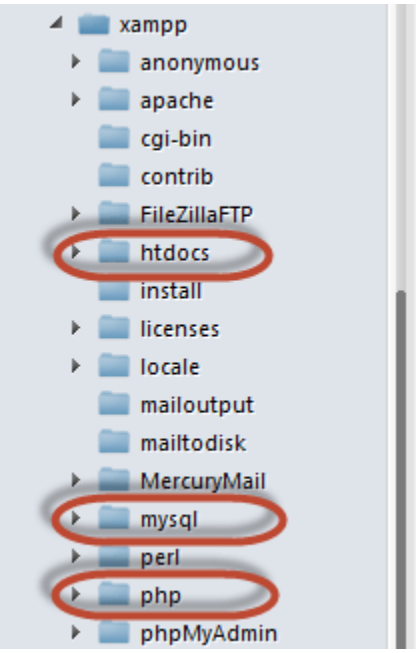
On successful completion of installation, you will see following window

- Click on Finish button

Before we test our XAMPP installation, let's first look at the basic directories that we will be working with.

Basic XAMPP Web Server Configuration

This XAMPP Tutorial assumes that you have **installed XAMPP on drive C in Windows using the steps mentioned above.** The following is a list of the basic directories that you are supposed to be aware of.



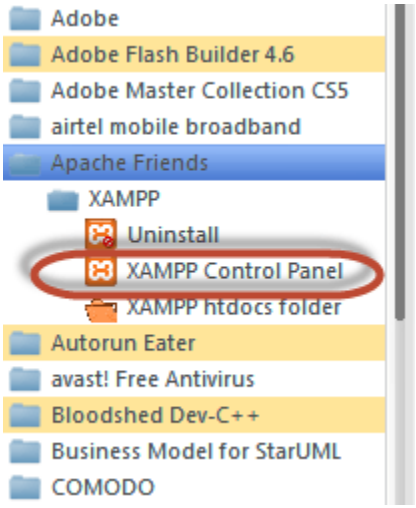
- **htdocs**; this is the web root directory. All of our PHP codes will be placed in this directory.
- **mysql** – this directory contains all the information related to MySQL database engine, by default it runs on port 3306.
- **php** – this directory contains PHP installation files. It contains an important file named php.ini. This directory is used to configure how PHP behaves on your server.

By default, the Apache web server runs on **port 80**. If port 80 is taken by another web server, you can use a different port number. For this tutorial we will assume we are using port 80. Note,If you use SKYPE , it uses the same port. Close Skype if you want to use XAMPP for PHP on port 80

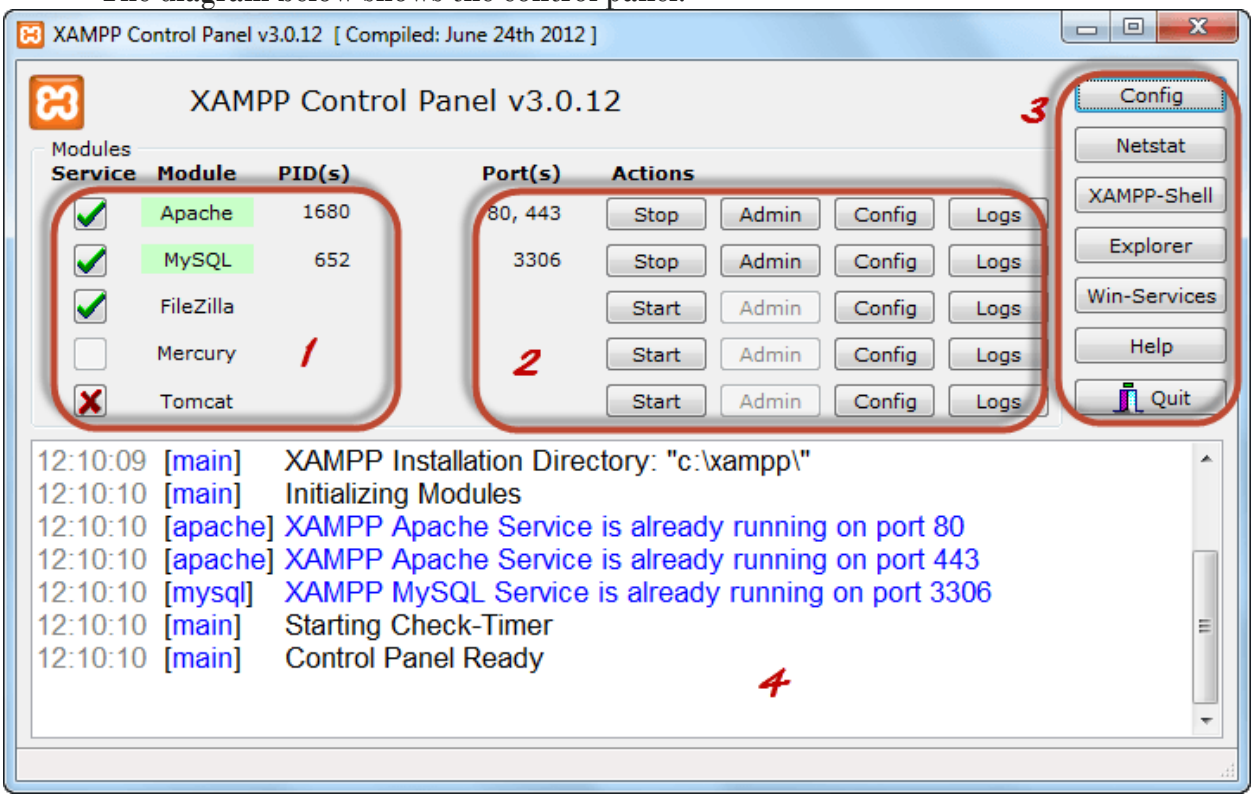
XAMPP Control Panel

The control panel is used to manage programs installed via XAMPP. To open the XAMPP Server control panel,

- Click on start menu
- Explore the programs directory and locate Apace Friends then XAMPP as shown in the diagram below



- The diagram below shows the control panel.



1) This section lists the installed services, modules and the process IDs PID(s). A green tick means the module has been installed as a service. The red mark means it has not been installed as a service. To install a service, click on the red mark. If the button shows a green tick and you click on it, the control panel will ask you if you want to uninstall the system.
2) This section shows Port(s) associated with the modules. The actions section is for;

1. starting and stopping modules
2. Open the administrative windows for Apache and MySQL
3. Open configuration files for Apache, MySQL etc. to make changes

4. View log files for the modules

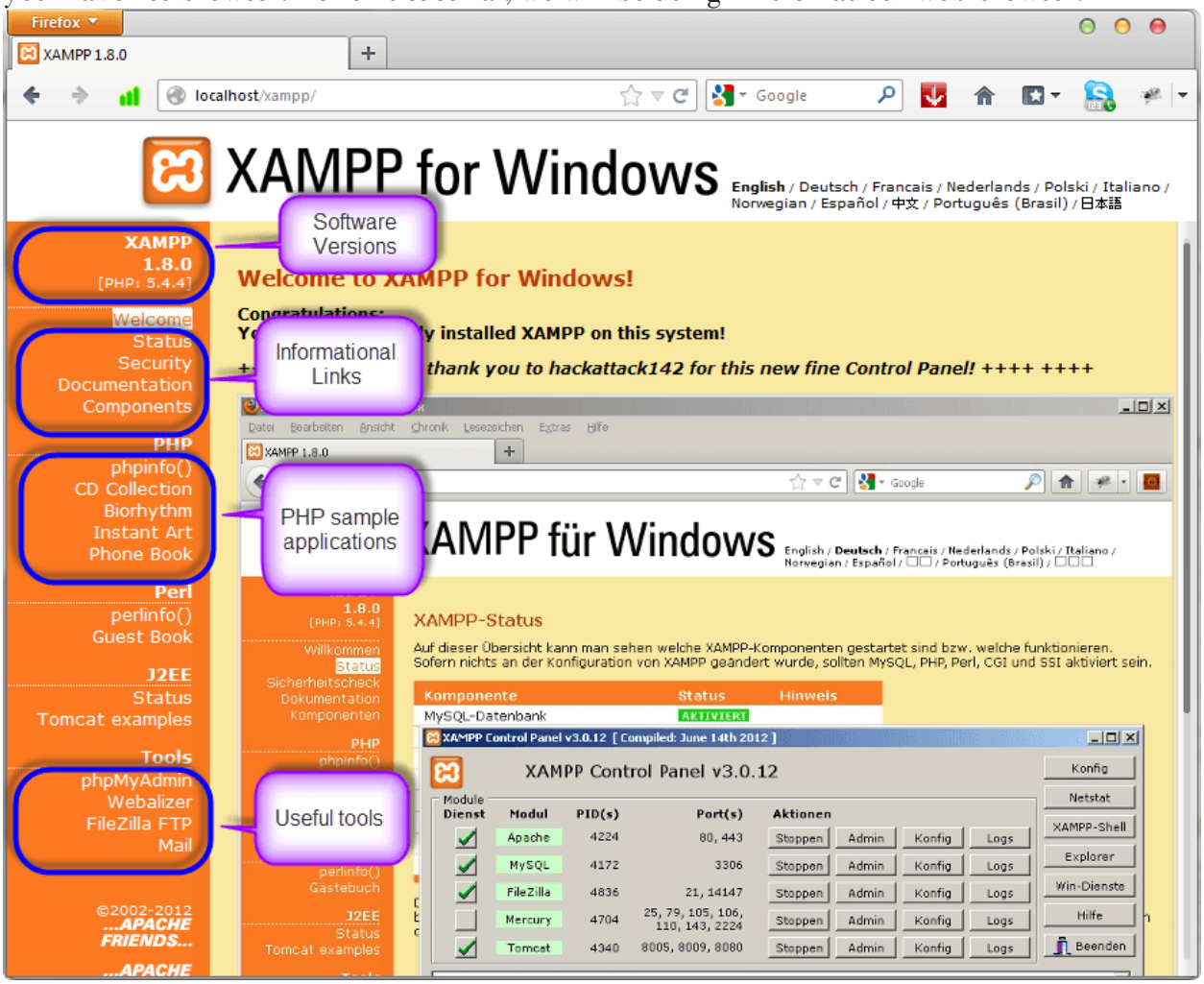
3) This section contains useful utilities such as Netsat, windows services short cuts etc.

4) This section displays status information on the modules. The control panel can be used to;

- Install and uninstall services such as Apache, MySQL etc. that are installed via XAMPP
- Start and stop services.
- Open configure files etc.

Configure XAMPP

Let's now look at the basic configurations required before we start using our XAMPP installation for developing PHP powered web sites. Type the URL **http://localhost/xampp/** in your favorite browser. For this tutorial, we will be using Firefox as our web browser.



If you are able to see the above screen then you have installed XAMPP successfully. The panel on the left hand side contains links to useful information such as;

- The version of PHP installed
- Security settings of XAMPP

- Access to utilities such as phpMyAdmin etc.

3.php variables and data types:[10 Marks]

Variables:variables are classified into different types.

a)local variables

b)global variables

c)static variables

d)reference variable

e)variable variables

f)super global variable

a)local variable:

local variable is a variable and we can't call the local values out side of the function.when we call local variable with in the function only.

Ex:<?php

function fun1()

{

$a=100;

Echo $a;

}

Fum1();

Echo $a;//error

?>

b)global variable:

global variable is a variable and when we call global variable,it access entire script. But php can not access global variable directly.By using "$GLOBAL"we can call the global variable.

->Here,we have to re-declare the global variable with in the function using global keywords.

Ex:<?php

$gb="scott";

Function fun1()

{

Echo $GLOBAL['gb'];

}

Fun();

?>

Ex2:<?php

Error-reporting(E-ALL)

$a=10;

Function fun1()

```
{
Echo $GLOBALS['$a'];
}
Fun1();
?>
```

c)static variable:

static is a keyword.static variable is used to maintain the state of an application.

We can assign values into the static variables only 'one' time.

Ex:

```
<?php
Function fun1()
{
Static $1=10;
Echo $a;
$a++;
}
Fun1();
Fun1();
Fun1();
?>
```

o/p:10,11,12

ex2:

```
<?php
function fun1()
{
$a=10;
Echo $a;
$a++;}
Fun1();
Fun1();
Fun1();
?>
```

o/p:10,10,10

d)Reference variable:

reference variable is a variable and we can create the reference variable by using "&" symbol.In this variable refers the sane address location what actual variable is referring.Reference variable is an alias name of actual variable.

Ex:

```
<?php
$a=100;
$b=&$a;
//echo $b;->here o/p=100;
```

$b=123;
Echo $a;->here o/p=123
?>

e)variable variable:

<?php
$a=10;
$b= 'a';
Echo $$b;
?>
o/p:10

f)super global variable:

php providing number of super global variables to access the values from different locations.the data type super global variables is an "Array".
->the scope of accessibility is entire application and out side the application.
1)$_GET:this variable is used to access "get" method posted values.
2)$_POST:To access post method posted values.
3)$_REQUEST:used to access get ,post values ,query string values and cookies values.
4)$_ENN(environment variables):
By using this super global variable we can the "o.s"variable.
5)$_COOKIE:To get the values of cookies.
6)$_SESSION:To create session variables and to read the values of session variables.
(7)$_FILE:this super global variable return the information about upload file.

3. PHP DATA TYPES:[10 marks]

A data type is used to specify the of data what variable holds.
Ex:int,string,etc.
->in php we have three basic types there are
1)scalar data type
2)compound data
3)specia data type

1)scalar data type: all primitive typea comes under this catogery.

Boolean: this data type represents either true (or) false. In php,value of true is "1"and false is nothing.

a)is-bool: by using this function we can check wether the input value is Boolean (or) not.

Ex:<?php

$x=true;
Echo $x;
?>
Ex2:<?php
$x=false;
Echo is_bool($x);
?>
b)(bool)variable,(Boolean)variable: to convert a variable into boolean data type.
Ex:<?php
$x= "false";
$x=(bool)$x;
Echo is_bool($x);
?>
Integer:this data type represents numerical value.
(a)is-int:this function checks wether the input value is integer (or) not.
Ex:<?php
$x=123;
Echo is_int($x);
?>
(b)(int)variable,(integer) variable,intval(variable):
To convert variable into integer data type.
Ex:<?php
$x= "123";
$x= intval($x);
Echo is_int($x);
?>
Float:this data type represents floating point numbers(decimal value).
(a)is_flaot:to check wether the input variable is float (or)not.
(b)float variable:to convert a variable into floating point number.
String: a string is a collection of chatacters.in php we can declare string in three ways.
(a)using single quotations(' ')
(b)using double quotations(" ")
(c)using heredoc syntax
Ex: <?php
$uname= "scott";
$add= 'hyd';
Echo $uname;
Echo $add;
?>

o/p:scotthyd

Ex:

```
<?php
$uname="scott";
$add='hyd';
$str= "$uname city is $add";
$str1= '$uname city is $add';
Echo $str;
Echo $str1;
?>
```

o/p:scott city is hyd

$uname city is $add.

Note :if we place a variable with in the double quotation gets the value of the variable.

(c)heredoc syntax: by using heredoc syntax we can avoid the conflict form ' '(or) " "quotation.

->heredoc syntax mostly used to display html script throw o/p function.

Ex:

```
<?php
Echo <<< mystr;
<input type= "button" value= "login">
Mystr;
?>
```

(2)compound data type:

Array: Array is a collection of hetrogenious data types.php is loosely type language that's why we can store different data types in array variable.

$arr= array(10,20,30);

->In an array ew can store no.of elements and each element is the combination of elements keys and element values.

->the key of first element start with '0' and last element is n-1;

Ex:

```
<?php
$arr=array(10,20,30);
Print_r($arr);
?>
```

Objects: An object is instance of a class we can access the class properties with help of objects.

(3)special data type:

(a)Resource data type: a resourse variable refers the external like database connections,filepointers,FTP connections etc.

Ex:

```
<?php
$con=MYSQL_connect("localhost",root "  ");
?>
```

(b)Null data type:In php null is not a value we can consider a variable has null based on three conditions.

->If the variable not assigning with any value

->If the variable is assign with null

->If the value of variable is delete using unset function.

(c)is-null:by using this function we check the input variable is null (or) not.

Ex:<?php

$x=123;

Unset($x);

Echo is_null($x);

?>

5 COOKIES:[10 marks]

A cookie is a piece of information sent by a webserver to a web browser and saved by the browser,and sent back to the server later.cookies are transmitted inside the HTTP header cookies move from server to browser and back to server as shown in the following diagram.

Web server	Web browser	Local system	Web browser	Web server
Send cookies	Receive cookies	Save cookies	Send back cookies	Receive cookies

From the diagram; cookies are actually saved by the local system in memory.

And cookies are mainly used to pass information from one php script to the next script.

--> one browser created cookies is not allowed to access by another browser.

-->we need to provide expiry time for cookie in hard disk other wise we don't provide expire time then it stores in RAM.

-->so,every time cookies store in browser memory location only.

Cookies are divided into two types there are:

(1)Inmemory cookie

(2)persistence cookie.

1.Inmemory cookie:

A cookie without explicit expiry time is called an Inmemory cookie.

-->Inmemory cookie destroys it self when the browser is closed.

Example-1

```
<?php
 Setcookie("city","kadapa");
 Echo $_cookie['city'];
 Setcookie("uname","scott");
 Echo $_cookie('uname');
 ?>
```

Example-2

```
<?php
Setcookie("sno","1001");
```

```
Echo $_cookie['sno'];
?>
```

Output:1001

2.persistence cookie: A cookie with explicit expiry time is called as persistent cookie.

-->It stores in clients hard disk and destroys it self when expiry time is reached.

NOTE: In php time is a function it returns time and date information(time()).

```
<?php
Setcookie("city","hyd",time()+3600);
Echo "cookie is created";
?>
```

Setcookie():- By using this function we can create the cookies.the syntax of calling stecookie() is

Setcookie(stringname,stringvalue,intexpire)

-->"name" is name of cookie.

-->"value" is the value of cookie.here value is optional.if"**value"**is not provide,this cookie will be set in the HTTP response without any value.

--> "Expire" is the time when this cookie should be expire.expire is optional if expire is not provided, this cookie will saved in browser memory only.if expire is provided,it represents a time in number of seconds since the epoch.

→if the provided time is a future time this cookie will be saved on the hard disk of browser system.

$_COOKIE :- By using this super global variable we can read the values of cookies.

- The best way to set expire is use the time function,which represents the current time in number of seconds.example, 30 days from today can expressed as
- "time()+60*60*24*30".the expire is not given ,a temporary cookie will be created.
- To shows you hoe to create persistent cookie.
  ->Get the date and time information when user is accessing cookie.
  ->Add life time():-to the current date and time information to get the expire time.
  ->create the cookie with the expire time.

NOTE:- All persistence cookie will store in a file.

```
<?php
    $cookiename="user";
    $cookievalue="herong young";
    $expiration=time()+60*60*24*30;
    Setcookie($cookiename,$cookievalue,$expiration);
    Print("<pre>\n");
    Print("cookies added by the server:\n");
    Print("$cookiename:$cookievalue\n");
    Print("expires at:$expiration\n");
    Print "</pre>\n";
    ?>
```

Output: user : herong young
Expires at : 1134531525.

Example:-

```
<?php
Setcookie("os","win",time()+3600);
Echo $_cookie['os'];
?>
```

Output: win 1

Once browser loads,It checks the cookie expire time and checks it with os time

NOTE:-

-->By using –ve time value we can delete the cookie before expire time

Time()-1

(or)

-->we can destroy the cookie before expiretime by re-creating the cookie with negative time.

```
<?php
Setcookie("john","jouny",time()-1);
?>
```

Disadvantages of cookie:-

cookies are unsecured why because user can see the information of cookie and user can delete the cookie information.

-->cookies are store only "test data".

-->cookies can store limited amount of DATA.

6 SESSIONS:-[5 Marks]

A session is an abstract concept to represent a series of HTTP request and responses exchanged between a specific web browser and a specific web server.

The session concept is very useful for web based applications to pass and share information from one web page to another web page. And session concept is used by web server side application.since the current desigen of HTTP protocol does not support session concept.

-->all web server side scripting technologies, including php, have designed their own ways to support the session concept.

-->how to identify a session with an ID and how to maintain the session ID.one common way to maintain the session ID is use the cookie technology .

```
                Browser                                    Server
                   |  <----- -------  request   # 1   -------  -----> | ID created
ID kept as choice  |  <-------------- response  # 1   -------------->  |
ID send back to server  |  <-------------- request   # 2   -------------->  |
                   |  <-------------- response  # 2   -------------->  |
ID send back to server|  <-------------  request   #3   ------------>  |
```

-->The session concept should be managed by the server when first request comes from a browser on a client host.

-->the server should created a new session, and assigns a new session ID.

-->the session ID will be sent back to the same browser as a cookie.

-->the browser will remember this ID,and send the ID back to the server in subsequent requests.

-->when the server receives a request containing the same session ID,it knows that this request is a continuation of an existing session.
-->session ID's are passed as cookies (or) GET/POST variable session-start() is the built-in function to start the session.$_SESSION is the built-in array to manage session data.

In this session concept we have to implement different types of functions.

(1)session_start():- By using this we have to creates the new session,with a session ID and records

Syntax:

Boolean session_start()

In this function is called more than once in the same session and the function retrives the $_SESSION array.this array stores all the session variables with their values.these values to the $_SESSION

Example:

```
<?php
Session_start();//code          ?>
```

Example:

```
<?php
Session_start();
$_SESSION['uname']="scott";
echo $_SESSION['uname'];
?>
```

In php.ini 'session.auto_start' is directive to initialize session on request startup.the default value is zero that's why we can't access session from the one page to another page.

Access the session change the session_auto_start value '1' declare as a session_start on webpage.

```
<?php
Session_start();
$_SESSION['uname']="scott";
//echo $_SESSION['uname'];
?>
<form method='POST'action="sess1.php">
<input type="submit"name="submit"value="send">
</form>
```

Sess1.php

```
<?php
Session_start();
echo $_SESSION['uname'];
echo "hi";
?>
```

(2)Session_id():-

By using this function for every user webserver will generate an unique 'id' is called as 'session id'.it is 32-characters length and alphanumeric string.and it's return the session id of current user.

```
<?php
Session_start();
```

```
echo session_id();
?>
<form method="post"action="session1.php">
<input type="submit">
</form>
```

Session1.php:

```
<?php
Session_start();
echo session_id();
?>
```

(3)session_unset:-

This function delets all the session variables stored in the current session.however it does not completely delete the session from the storage mec hanism

Syntax:

Void session_unset()

Example:

```
<?php
Session_start();//create session variables.
$_SESSION['login']="ysr";
$_SESSiON['password']='secret';
Session_unset();//delete all session variables
?>
```

(4)session_destroy:-

This function deletes the current session completely from the storage mechanism.in other words,after a call to this function,the current session becomes invalid.however ,it does not delete the cookies that are stored on the browser

Syntax:

Boolean session_destroy()

Example:

```
<?php
Session_start();//creating a session variable
$_SESSION["page_number"]=1;
//accessing session variable
$ count=$_SESSION["page_number"];
Print "this page visited $ count times<br/>";
//increament page no
Session_unset();
Session_destroy();
?>
```

Storing simple data types in sessions:

------>Integer

------>String

------>floating-point value

Example:

```
<?php
Session_start();
()$_SESSION['string_value']="welcome";
(float)$_SESSION['float_value']="98.3";
(int)$_SESSION['integer_value']="16";
Function outputsession()
{
echo $_SESSION['integer_value']"<br/>";
echo $_SESSION['string_value']"<br/>";
echo $_SESSION['float_value']"<br/>";
}
Outputsession();
?>
```

$_SESSION:-By using this super global variable we can create the session and read session
-->SESSION is a state management object used to maintain the state of the application.

DIFFERENCE BETWEEN COOKIES AND SESSION:- [2 marks]

1.Cookies resides in client system 2.cookies are used to store limited amount of data 3.cookies are unsecured 4.cookies are burden to the client system because cookies occupy client memory	1.session resides in webserver 2.session can store huze amount of data 3.sessions are highly secured 4.sessions occupy server memory that's why they are overread to the web servers

7 php class and object[5 marks]

Class: - class is a collection of objects and it is a template to implement software logic. Class contains properties and methods.
Object: - object is instance f a class and we can access class properties with help of object.
Method: - A function within the class called as method.
Property: - variable with in the class is called properly.
Ex:-

```
<? Php
Class clasdemo
{
   Function funcls ( )
      {
           Echo "from class";
       }
```

```
}
    Function fun1 ( )
      {
         Echo "from function 1 ";
      }
Fuc1 ( );
$obj = new clsdemo ( );
$obj funcls ();
?>
```

NOTE: - "new "keyword we can create the new memory location load the class contents.

- CONSTRUCTOR:-

A constructor is special type of method with in the class. Which contain class name as method name.

- Constructor can accept parameters, which are assigned to specific object properties at creation time.
- Constructor can call class methods (or) other functions.
- Class constructors can call on other constructors;
- Constructors are evaluated in two ways.

  One is b using "- - construct" keyword and second one is by using "class name".
- When we create " -- construct" keyword we called as default constructor

Example:-

```
<? Php
Class clsdemo
 {
    Echo "constructor with class name";
  }
Function - - construct ( )
{
Function funcls ( )
 {
   Echo "from class";
  }
}
    Function   fun1 ( )
      {
         Echo "from function 1 ";
      }
Fun1 ( );
   $obj = new clsdemo ( );
?>
```

- DESTRUCTOR: - Destructor is a special kind of method executes when we destroy the object of a class.
- If we want to execute any statement like uninitialized the variable, close the data base.
- In php, when the script reaches last statement the object will destroy itself otherwise we can destroy the object of a class using "unset" method at the middle of the script.

- By using “--destruct” keyword we can create the destructor.

Example:-

```
<? Php
Class mydesclass
 {
   Function—construct ( )
      {
        Print “In construction”;
       $this. name = “my desclass”;
      }
  Function --destruct ( )
     {
         Print “Destroying”. $this name. “\n”;
       }
  }
$obj = new mydesclass;
?>
```

- **ABSTRACT CLASSES AND METHODS:-**

Abstract class is a class to maintain the abstract methods. Abstract methods we are using to provide a structured to the application.

- What method user must implement in derive class they should be abstract method.
- Abstract classes can contain abstract methods & concrete methods.
- By using “abstract” keyword we have to create the abstract methods.
- Abstract methods are declared with the abstract keywords.
- Abstract methods are only created in an abstract class.

EXAMPLE:-

```
<? Php
Abstract    class    abclass
 {
   Abstract function absfun ( );
Function fun1 ( )
  {
    Echo ‘This is concrete method”;
  }
 Class   clschild extends   abclass
   {
      Function   absfun ( )
       {
        Echo “Implementation from derived class”;
        }
      $obj = new clschild ( );
      $obj .absfun ( );
    // $obj.fun ( )   we can not call
?>
```

Interface:-

Interface is same as abstract classes. But interface cannot maintain non-abstract method.

1. Interface comes under fully abstraction and in an interface we cannot declare the variables.
2. By using interface we can implement multiple inheritance.
3. Interface methods are implementing in derived classes.
4. In an interface we can use only public access specifier.
5. By using "implements "keyword, we have to implements interface methods.

Example:-

```
<? Php
Interface iface1
{
 Function fun1 ();
}
Interface iface2
{
 Function fun2 ();
}
Class cls1 implements iface1, iface2
{
Function fun1 ()
{
Echo "implements from derived class ";
}
Function fun2 ()
{
Echo "implements from derived class";
}
}
?>
```

- *Diff Between Abstract class and Interface*

Abstract	Interface
1. Abstract classes can maintain the abstract methods and non abstract methods. 2. Abstract classes comes under partial abstraction 3. In abstract class we can declare variable 4. By using abstract classes we cannot implement multiple inheritance concept. 5. We can extend the members of abstract class into derived class.	1. Interface cannot maintain non-abstract method. 2. Interface comes under fully abstraction 3. In an interface we cannot declare the variables 4. By using interface we can implement multiple inheritance

6. Abstract classes we can use any type of access specifier	5. Interface methods we are implementing in derived classes. 6. In interface we can use only public access specifier

8 PHP MySQL Create Database[5 marks]

Since PHP 4.3, **mysql_create_db()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::__query()**

PHP MySQLi Create Database Example

Example

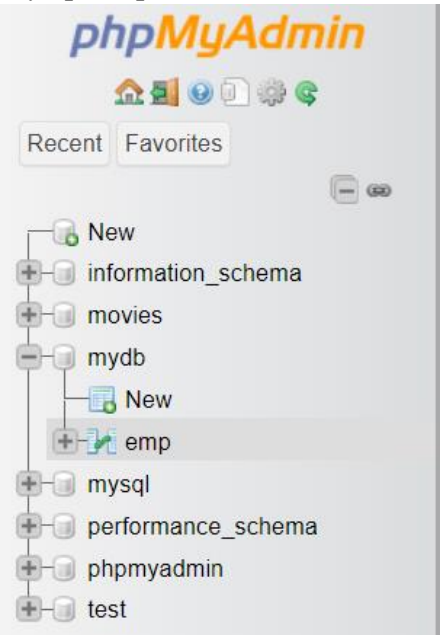
```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$conn = mysqli_connect($host, $user, $pass);
if(! $conn )
{
  die('Could not connect: ' . mysqli_connect_error());
}
echo 'Connected successfully<br/>';


$sql = 'CREATE Database mydb';
if(mysqli_query( $conn,$sql)){
  echo "Database mydb created successfully.";
}else{
echo "Sorry, database creation failed ".mysqli_error($conn);
```

```
}
mysqli_close($conn);
?>
```

Output:

```
Connected successfully
Database mydb created successfully.
```

Mysql output:



9 PHP MySQL Create Table[5 marks]

PHP mysql_query() function is used to create table. Since PHP 5.5, **mysql_query()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::__query()**

PHP MySQLi Create Table Example

Example

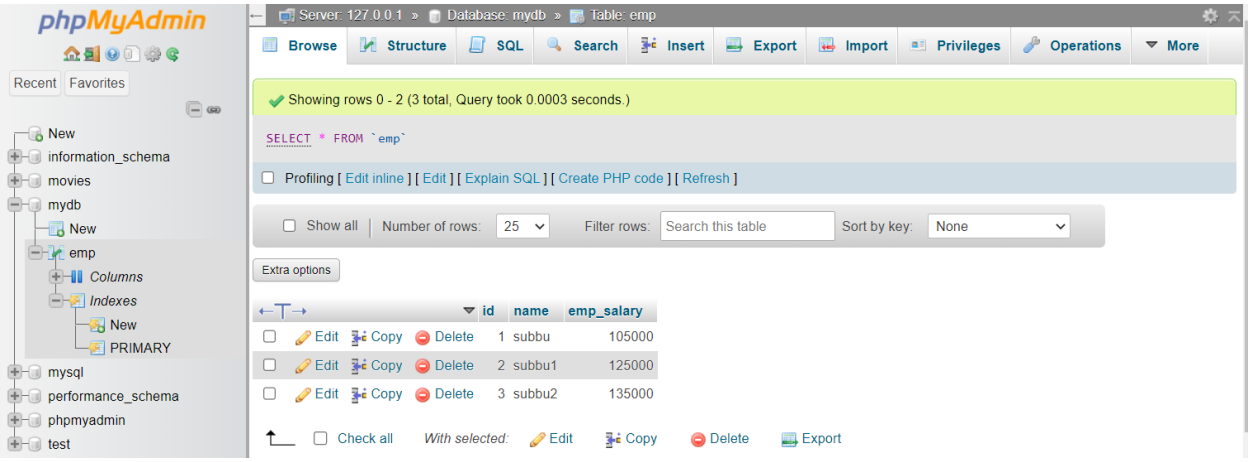
```
<?php
$host = 'localhost:3306';
$user = 'root';
$pass = '';
```

```
$dbname = 'studentsa';


$conn = mysqli_connect($host, $user, $pass,$dbname);
if(!$conn){
  die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';


$sql = "create table emp(id INT AUTO_INCREMENT,
                    name VARCHAR(20) NOT NULL,
                     emp_salary INT NOT NULL,primary key (id))";
if(mysqli_query($conn, $sql)){
 echo "Table emp created successfully";
}
else
{
echo "Could not create table: ". mysqli_error($conn);
}
  mysqli_close($conn);
?>
```

Output:

```
Connected successfully
Table emp5 created successfully
```

Mysql output:



10 PHP MySQL Insert Record[5 marks]

PHP mysql_query() function is used to insert record in a table. Since PHP 5.5, **mysql_query()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::__query()**

PHP MySQLi Insert Record Example

Example

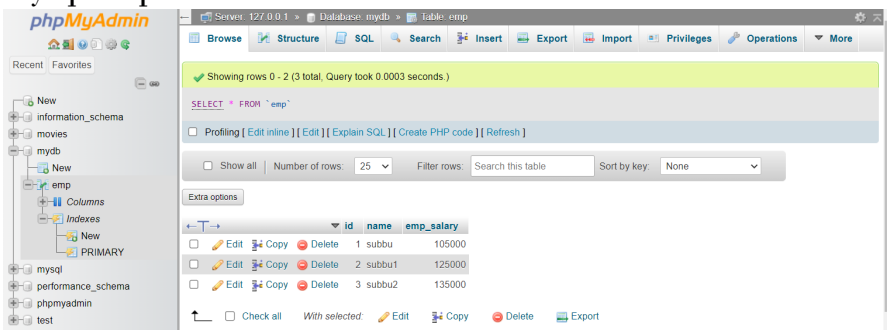
```
<?php
$host = 'localhost:3306';
$user = 'root';
$pass = '';
$dbname = 'mydb';

$conn = mysqli_connect($host, $user, $pass,$dbname);
if(!$conn){
  die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';

$sql = 'INSERT INTO emp(name,emp_salary) VALUES ("subbu", 105000)';

if(mysqli_query($conn, $sql))
{
 echo "Record inserted successfully";
}
else
{
```

```
echo "Could not insert record: ". mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Output:

```
Connected successfully
Record inserted successfully
```

Mysql output:



11 PHP MySQL Select Query[2 marks]

PHP mysql_query() function is used to execute select query. Since PHP 5.5, **mysql_query()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::__query()**

There are two other MySQLi functions used in select query.

- **mysqli_num_rows(mysqli_result $result)**: returns number of rows.
- **mysqli_fetch_assoc(mysqli_result $result)**: returns row as an associative array. Each key of the array represents the column name of the table. It return NULL if there are no more rows.

12 Fetching data from MySQL using PHP

Example

```
<?php
$host = 'localhost:3306';
$user = 'root';
$pass = '';
```

```
$dbname = 'mydb';
$conn = mysqli_connect($host, $user, $pass,$dbname);
if(!$conn)
{
  die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';

$sql = 'SELECT * FROM emp';
$retval=mysqli_query($conn, $sql);

if(mysqli_num_rows($retval) > 0)
{
 while($row = mysqli_fetch_assoc($retval))
{
    echo "EMP ID :{$row['id']}  <br> ".
        "EMP NAME : {$row['name']} <br> ".
        "EMP SALARY : {$row['emp_salary']} <br> ".
        "--------------------------------<br>";
 } //end of while
}
else
{
echo "0 results";
}
mysqli_close($conn);
?>
```

Output:

```
Connected successfully
EMP ID :1
EMP NAME : subbu
EMP SALARY : 105000
--------------------------------
EMP ID :2
EMP NAME : subbu1
EMP SALARY : 125000
--------------------------------
EMP ID :3
EMP NAME : subbu2
```

EMP SALARY : 135000
---

13 Delete Data From a MySQL Table Using php

The DELETE statement is used to delete records from a table:
DELETE FROM table_name
WHERE some_column = some_value
Notice the WHERE clause in the DELETE syntax: The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

To learn more about SQL, please visit our SQL tutorial.

Let's look at the "MyGuests" table:

id	firstname	lastname	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23

The following examples delete the record with id=3 in the "MyGuests" table:
Example (MySQLi Object-oriented)Get your own PHP Server

```
<?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();
?>
```

Example (MySQLi Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
  echo "Record deleted successfully";
} else {
  echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

After the record is deleted, the table will look like this:

id	firstname	lastname	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30

PHP DATE [5 MARKS]

In this , we will see how to get the date & time using the date() & time() function in PHP, we will also see the various formatting options available with these functions & understand their implementation through the examples.

Date and time are some of the most frequently used operations in PHP while executing SQL queries or designing a website etc. PHP serves us with predefined functions for these tasks. Some of the predefined functions in PHP for date and time are discussed below.

PHP date() Function: The PHP date() function converts timestamp to a more readable date and time format.

Why do we need the date() function?

The computer stores dates and times in a format called UNIX Timestamp, which measures time as a number of seconds since the beginning of the Unix epoch (midnight Greenwich Mean Time on January 1, 1970, i.e. January 1, 1970, 00:00:00 GMT). Since this is an impractical format for humans to read, PHP converts timestamp to a format that is readable and more understandable to humans.

Syntax:

date(format, timestamp)

Explanation:

- The format parameter in the date() function specifies the format of returned date and time.
- The timestamp is an optional parameter, if it is not included then the current date and time will be used.

Example: The below program explains the usage of the date() function in PHP.

- PHP

```
<?php
  echo "Today's date is :";
  $today = date("d/m/Y");
  echo $today;
?>
```

Output:

Today's date is :05/12/2017

Formatting options available in date() function: The format parameter of the date() function is a string that can contain multiple characters allowing to generate the dates in various formats. Date-related formatting characters that are commonly used in the format string:

- d: Represents day of the month; two digits with leading zeros (01 or 31).
- D: Represents day of the week in the text as an abbreviation (Mon to Sun).
- m: Represents month in numbers with leading zeros (01 or 12).
- M: Represents month in text, abbreviated (Jan to Dec).

- y: Represents year in two digits (08 or 14).
- Y: Represents year in four digits (2008 or 2014).

The parts of the date can be separated by inserting other characters, like hyphens (-), dots (.), slashes (/), or spaces to add additional visual formatting.

Example: The below example explains the usage of the date() function in PHP.

- PHP

```php
<?php
  echo "Today's date in various formats:" . "\n";
  echo date("d/m/Y") . "\n";
  echo date("d-m-Y") . "\n";
  echo date("d.m.Y") . "\n";
  echo date("d.M.Y/D");
?>
```

Output:

```
Today's date in various formats:

05/12/2017

05-12-2017

05.12.2017

05.Dec.2017/Tue
```

The following characters can be used along with the date() function to format the time string:

- h: Represents hour in 12-hour format with leading zeros (01 to 12).
- H: Represents hour in 24-hour format with leading zeros (00 to 23).
- i: Represents minutes with leading zeros (00 to 59).
- s: Represents seconds with leading zeros (00 to 59).
- a: Represents lowercase antemeridian and post meridian (am or pm).
- A: Represents uppercase antemeridian and post meridian (AM or PM).

Example: The below example explains the usage of the date() function in PHP.

- PHP

```php
<?php
  echo date("h:i:s") . "\n";
  echo date("M,d,Y h:i:s A") . "\n";
  echo date("h:i a");
?>
```

Output:

```
03:04:17

Dec,05,2017 03:04:17 PM

03:04 pm
```

PHP time() Function: The time() function is used to get the current time as a Unix timestamp (the number of seconds since the beginning of the Unix epoch: January 1, 1970, 00:00:00 GMT).
The following characters can be used to format the time string:

- h: Represents hour in 12-hour format with leading zeros (01 to 12).
- H: Represents hour in 24-hour format with leading zeros (00 to 23).
- i: Represents minutes with leading zeros (00 to 59).
- s: Represents seconds with leading zeros (00 to 59).
- a: Represents lowercase antemeridian and post meridian (am or pm).
- A: Represents uppercase antemeridian and post meridian (AM or PM).

Example: The below example explains the usage of the time() function in PHP.

- PHP

```php
<?php
  $timestamp = time();
  echo($timestamp);
  echo "\n";
  echo(date("F d, Y h:i:s A", $timestamp));
?>
```

Output:

```
1512486297

December 05, 2017 03:04:57 PM
```

PHP mktime() Function: The mktime() function is used to create the timestamp for a specific date and time. If no date and time are provided, the timestamp for the current date and time is returned.
Syntax:

```
mktime(hour, minute, second, month, day, year)
```

Example: The below example explains the usage of the mktime() function in PHP.

- PHP

```php
<?php
  echo mktime(23, 21, 50, 11, 25, 2017);
?>
```

Output:

```
1511652110
```

The above code creates a time stamp for 25th Nov 2017,23 hrs 21mins 50secs.