Name : Farheen Adam Master.

Roll No: 17CO05

EXP: 01

AIM: Installation and Setting of LAMP/WAMP/XAMP.

- **LAMP INSTALLATION:**

This procedure assumes that Ubuntu is already installed on your machine.

Step 1: Update Package Repository Cache
Before you begin:

1. Open the terminal either by using the CTRL+ALT+T keyboard shortcut or by searching for the word terminal in Ubuntu Dash.
2.Make sure to update the package repository cache to ensure it installs the latest versions of the software. To do so, type in the following command:

sudo apt-get update

each command as it is completed, and so do not necessarily illustrate the entire process.

sudo apt-get upgrade

have to confirm “yes” when running the “upgrade” command.
The first command is going to update the list of packages and their versions on your machine without going through the process of actually installing them. The second command installs the packages. For this reason, you should always run the update command first and then upgrade.

Step 2: Install Apache
1. To install Apache, run the following command in the terminal:

sudo apt-get install apache2

Press y (yes) and hit ENTER to permit the installation.
2. Check if Apache is installed correctly by running the Apache service status. Use the following the command:

sudo service apache2 status

If everything installed correctly, you will receive this output:

Active = active(running)

3. Next, make sure that the UFW firewall has an application profile for Apache by typing in the following command:

sudo ufw app list

In the Apache Full profile, make sure it allows the traffic on ports 80 and 443. Check this by typing the command:

sudo ufw app info “Apache Full”

4. To ensure Apache is running, enter the IP address of your server in the address bar and press ENTER.

To identify the server’s public IP address, run the command:

sudo apt-get install curl

curl http://icanhazip.com

You should now have a running web server on your machine. To check this, go to http://localhost/, where you should see a page saying “It works!”

To install the tool and helper package, type the following command into the Terminal:

sudo apt-get install python-setuptools libapache2-mod-wsgi

Step 3: Install MySQL and Create a Database

To install MySQL, type in the following command:

sudo apt-get install mysql-server

Press y to allow the installation.

During the installation, you will be prompted to set the root user password.

During the process of installing MySQL, you will be prompted to set a root password. This is the password you’ll use for the MySQL install. You can choose to set it now, or you can do so later, within the MySQL shell.

After MySQL has finished installing, it starts automatically. To log into your server, use the following command (you will be prompted for your root password):

mysql -u root -p

From there, you can start to play with MySQL and set up new databases and users.

Step 4: Install PHP

1. To install PHP, run the following command:

sudo apt-get install php libapache2-mod-php php-mysql

Press y and ENTER to allow the installation.

2. Next, you should modify the way Apache serves files when directories are requested. By default, Apache first looks for a file card named index.html. However, we want it to look for the index.php file instead.To change this, open the **dir.conf** file in a text editor with root privileges:

sudo nano /etc/apache2/mods-enabled/dir.conf

In the configuration file, you will see the information .
Then, move the PHP index file to the first position:
3. Press CTRL + X to save and close the file. Press y and ENTER to confirm.

Step 5: Restart Apache
For the changes to take effect, you must restart the Apache service.
Enter the command:

sudo systemctl restart apache2

you've installed an Ubuntu LAMP server. From here, you can start to play with Python, you can see how MySQL store and retrieve data—and you can begin building your very own web applications.

- **GIT INSTALLATION** :

Debian / Ubuntu (apt-get)
Git packages are available via apt:
From your shell, install Git using apt-get:

Follow these steps to install Git on your Ubuntu system:

1.Start by updating the package index:

$ sudo apt-get update

2Run the following command to install Git:

$ sudo apt-get install git

Verify the installation was successful by typing git --version:
verify the installation by typing the following command which will print the Git version:

$ git --version git version 2.9.2

Configuring Git:

Now that you have git installed, it is a good idea to set up your personal information that will be used when you commit your code.

The following commands will set your git commit username and email address:

$git config --global user.name "Your Name"

$git config --global user.email
"youremail@yourdomain.com"

To verify the configuration changes, type:

$git config --list

The output should look something like this:

user.name=Your Name user.email=youremail@yourdomain.com

The configuration settings are stored in the gitconfig file:

If you want to make further changes to your Git configuration, you can either use the git config command or edit the ~/.gitconfig file by hand.

Configure your Git username and email using the commands.

That's it, you have successfully installed Git on your Ubuntu and you can start using it.

- **NODE INSTALLATION :**

Node.js is a JavaScript platform for general-purpose programming that allows users to build network applications quickly.

In order to get this version, we just have to use the apt package manager. We should refresh our local package index first, and then install from the repositories:

$sudo apt-get update

$sudo apt-get install nodejs

If the package in the repositories suits your needs, this is all you need to do to get set up with Node.js. In most cases, you'll also want to also install npm, which is the Node.js package manager. You can do this by typing:

$sudo apt-get install npm

This will allow you to easily install modules and packages to use with Node.js.

Because of a conflict with another package, the executable from the Ubuntu repositories is called nodejs instead of node. Keep this in mind as you are running software.

To check which version of Node.js you have installed after these initial steps, type:

$nodejs -v

- **Live server in VS CODE :**

Snap packages can be installed from either the command-line or via the Ubuntu Software application.

To install the VS Code snap, open your terminal (Ctrl+Alt+T) and run the following command:

$sudo snap install --classic code

That's it. Visual Studio Code has been installed on your Ubuntu machine.

If you prefer using a GUI, open Ubuntu Software, search for "Visual Studio Code" and install the application:

Launch a local development server with live reload feature for static & dynamic pages.

Shortcuts to Start/Stop Server:

Open a project and click to Go Live from the status bar to turn the server on/off.

Right click on a HTML file from Explorer Window and click on Open with Live Server.

Open a HTML file and right-click on the editor and click on Open with Live Server.

Hit (alt+L, alt+O) to Open the Server and (alt+L, alt+C) to Stop the server (You can change the shortcut form keybinding). [On MAC, cmd+L, cmd+O and cmd+L, cmd+C]

Open the Command Pallete by pressing F1 or ctrl+shift+P and type Live Server: Open With Live Server to start a server or type Live Server: Stop Live Server to stop a server.

Open a HTML file and right-click on the editor and click on Open with Live Server . Open the Command Pallete by pressing F1 or ctrl+shift+P and type Live Server: Open WithLive Server to start a server or type Live Server: Stop Live Server to stop a server.