Regularity (2)	Content (4)	Viva-voce (2)	Timely Submission (2)	Total (10)	Dated Sign of Subject Teacher

---

Assignment No. 07 Date of Performance: ………………….………………

Title: Install, configure XMPP server and deployed an application on Raspberry Pi/Beagle board/Arduino. Write client applications to get services from the server application.

Objective: To study installation & configuration of XMPP server for client-server application on Raspberry Pi/Beagle board/Arduino.

Theory:

Why a Raspberry Pi as a web server?

First, from an economic point of view, you should know that web hosting services are not free and that you have to pay every month / year. Unlike the Raspberry who just need to a connection. In addition, by choosing Raspberry, you have the possibility to modify your services like you want (examples: the size of the disk, the hosting of Database, etc.), which is generally not the case with specialized hosts, which often sell shared hosting with low configuration capacity. However, to support more users, you should use a Raspberry Pi 3 (the Pi 3 can be found here), the Raspberry Pi with 1 GB of RAM, rather than the Raspberry Type B + (512 MB of RAM).

What is Apache?

Apache is the most used web server, with about 60% market share. Apache has its own license, used by many other projects. In addition, the massive use of Apache (which has become the standard for web servers), coupled with its high popularity, has led to a tremendous abundance of documentation, courses, and other books dealing with its use, and his security.

Apache installation

Before installing the server, make sure we have an up-to-date machine. To do this we must have administrator rights, either because of the sudo command.

```
sudo apt update
sudo apt upgrade
sudo apt update
```

Once the Raspberry Pi is up to date, we will install the Apache server.

```
sudo apt install apache2
```

By the way, we’ll take advantage of it to give rights to the apache file that you can easily manage your sites. To do this, run the following commands:

```
sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
```

Check if Apache is working

Once the installation completed, we can test that Apache is working properly by going to the Raspberry address. To do this, it’s necessary to try to access to the Raspberry from port 80 (this port not being opened from the outside, it will have to do since the Raspberry itself). Simply open the Raspberry web browser, and go to “http://127.0.0.1”. You should then get a page with a message like “It works! “And plenty of other text. If you do not already have a GUI on your Raspbian, or you use SSH to connect to your Raspberry, you can use the following command:

```
wget -O check_apache.html http://127.0.0.1
```

This command will save the HTML code of the page in the file “check_apache.html” in the current directory. So you only have to read the file with the command

```
cat ./check_apache.html
```

If you see marked at a location in the code “It works!” is that Apache is working. Apache uses the directory “/var/www/html” as the root for your site. This means that when you call your Raspberry on port 80 (http), Apache looks for the file in “/var/www/html”. For example, if you call the address “http://127.0.0.1/example”, Apache will look for the “example” file in the “/var/www/html” directory. To add new files, sites, etc., you will need to add them to this directory. You can now use your Raspberry to make a site in HTML, CSS and JavaScript, internally. However, you may want to quickly allow interactions between the site and the user. For example, to allow the user to register, etc. For this, you are going to need PHP.

PHP installation on your Raspberry Pi

We will again use the administrator to install PHP with the command line.

```
sudo apt install php php-mbstring
```

Control if PHP is working

To know if PHP is working properly, it’s not very complicated, and the method is quite similar to the one used for Apache. You will first delete the file “index.html” in the directory “/var/www/html”.

```
sudo rm /var/www/html/index.html
```

Then create an “index.php” file in this directory, with this command line

```
echo "" > /var/www/html/index.php
```

From there, the operation is the same as for the Apache check.

A MySQL database for server

Now that we have set up PHP, you will probably want to store information for use in your sites. For this purpose, databases are most often used. We will therefore set up a DBMS (Database Management System), namely MySQL. MySQL is a free, powerful, massively used DBMS (about 56% market share of free DBMS). Here again, MySQL is so essential to development, whatever the language, that you must absolutely learn and master it.

How to install MySQL

To do this, we will install mysql-server **and** php-mysql (which will serve as link between php and mysql)

```
sudo apt install mysql-server php-mysql
```

Verify that MySQL is working correctly

To check the operation of MySQL, this time we will only use the command line. To do this, we will simply connect via the command:

```
sudo mysql --user=root
```

We will no delete the default mysql root user and create a new mysql root user, because the default one can only be used with Linux root account, and so not available for the webserver and php scripts. To do so, once you connect to MySQL, simply run those commands (replace password with the password you want):

```
DROP USER 'root'@'localhost';
CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost'
```

So you now have a web server, connected to PHP and MySQL. That's all it takes.

Client Application:

The client application will be measuring environmental condition like humidity and temperature using DHT22 sensor.

Server Application:

On server side, the data generated on client side will be stored into MySQL database using XMPP server. Here client will be going to use different services of server.

Outcome: Thus we have studied how to install and configure XMPP server for data exchange between client application and server application.
