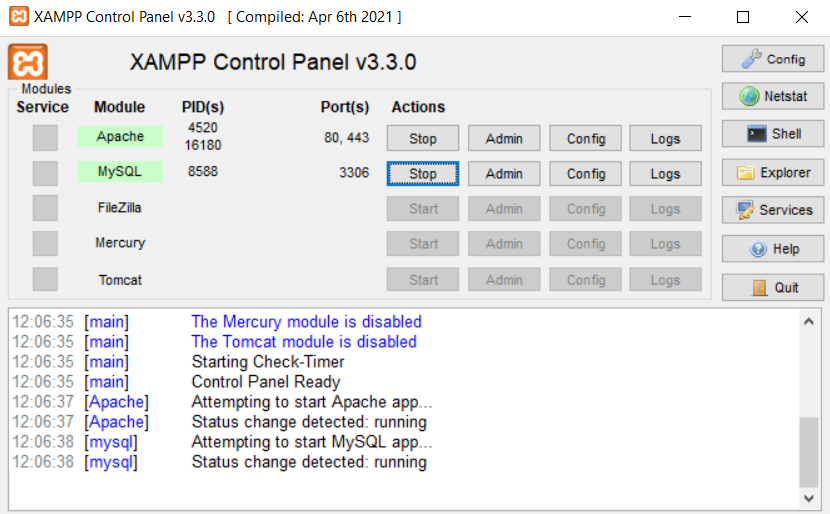
USER MANUAL

The application has been developed using html, css and php. The application is connected to a database built in sql using the free and open-source tool called phpMyAdmin. To run the application, I used XAMPP, a free software that allows you to build sites offline, on a local web server on your computer.

# XAMPP INSTRUCTIONS

1. Download Xampp at this link <https://www.apachefriends.org/it/index.html>
2. Run the .exe file and follow the procedures. To run our application, we only need to select the Apache, MySQL, PHP and PhpMyAdmin components.
3. Once Xampp is installed we must go to the xampp/htdocs directory (often it is C:\xampp\htdocs), create a new folder called Forum and paste all the files of the application.
4. Open the XAMPP Control Panel and start Apache and MySql (like in the foto)



1. To run the application just open the browser and go to localhost/directory\_of\_your\_folder/file.php, so in our case we go to localhost/Forum/login.php
2. The app will work only if a database is created. To do so, we go to localhost/phpmyadmin and import the forum.sql file

# STRUCTURE OF THE DATABASE

The database is composed of three tables:

* authentication(username, password)
* questions(question, username)
* answers(question, username, answer)

In the authentication table we save all the login data of the users, that is username and password where username is the primary and unique key.

In the questions table we save the text of the question and the username of the user who posted it.

In the answers table we save the text of the answer, the username of the user who responded and the reference question that is answered. In this way, when we want to access to the answer to a specific question, we just need to do a JOIN query of the two tables where question is the reference value.

# STRUCTURE OF THE CODE

The code is structured in 5 classes.

* Login.php
* Register.php
* Authentication.php
* Home.php
* Answer.php

For the style we have two files:

* Style.css
* loginStyle.css

Finally, we have one file for the database

* forum.sql

## LOGIN.PHP

In this file, there is the Html code to display the login form. The form action is performed in the authentication.php file which is included in the login.php file. If the user is not yet a member there is the link to register.php to sign up.

## REGISTER.PHP

This file is almost the same as login.php. There is the Html code to display the sign-up form. The form action is performed in the authentication.php file which is included in the login.php file. If the user is already a member there is the link to login.php to sign in.

## AUTHENTICATION.PHP

This file is mainly coded in php, and its aim is to configure the connection with the database. In addition, the sign-up method is defined, which takes the data from the form and inserts it into the database. The user is then logged in and directed to the file home.php

In the same way, the login takes the data from the form and checks if it matches the data in the database. If so, the user is directed to the home page.

## HOME.PHP

In the home.php file it is outputted the content of the database and it represents the home of the application, that is, the page where all main actions take place. We have some php code to maintain the authentication of the user and for the logout option. This is followed by some HTML code to design a form where a user can enter some new question. The methods implemented in php are: 1) the insertion of the question in the database 2) the display of the content of the database. To retrieve the data, we run a query where we join the questions and answers tables in order to list of all the questions and under each there is the related answer. Every time a new question is inserted, the page is loaded so the addition is in real time.

It is also possible to update or insert an answer by clicking the apposite button that will redirect you to the answer.php file. To do so, an <a> tag is used, and the question is attached as parameter.

## ANSWER.PHP

In the answer.php file, we retrieve the question passed as parameter and we display it, followed by a form to insert the answer and a button. Once the button is pressed the answer is inserted in the database by running the update query. The user is then redirected to the home page.

For the secure version of the application, the code is based on the code of the insecure app. The only changes are methods used to prevent attacks. These are explained in the demonstration\_secure\_app file.