WEB APPLICATION SERVICES ON DEBIAN 12 BOOKWORM USING NGINX AND MYSQL

Group 4:

Kheyral Sutan Dumas (2340040301)

Naufal Fauzan Wildani (2320010260)

Charisma Bayu Majestyno (2320010193)

Faculty:

Mr. Tri Agus Riyadi, M.T

Class:

2CS1

CEP CCIT FACULTY OF ENGINEERING

UNIVERSITY OF INDONESIA

2024

PROJECT INFORMATION

Project Title **:** Web Application Service on Debian 12 Bookworm Using Nginx and MySQL

Batch Code **:** 2CS1

Start Date **:** April 28, 2024

End Date **:** May 10, 2024

Name of Faculty **:** Mr. Tri Agus Riyadi, S.Kom, MT

Names of Administrator :

1. Kheyral Sutan Dumas
2. Naufal Fauzan Wildani
3. Charisma Bayu Majestyno

CERTIFICATE OF ORIGINALITY

This is to certify that the project report titled "Web Application Services on Debian 12 Bookworm Using Nginx and MySQL" is an original work completed by Kheyral Sutan Dumas, Naufal Fauzan Wildani, and Charisma Bayu Majestyno. This project has been submitted in partial fulfillment of their course requirement at the National Institute of Information Technology (NIIT).

The project report has been prepared under our guidance and supervision, and it is ensured that the work presented in this report is the result of the individual efforts of the aforementioned students. The contents of this report have not been submitted to any other institution or organization for the award of any degree, diploma, or other similar recognition.

Author acknowledge that the ideas, designs, and implementations presented in this project report are the intellectual properties of the students mentioned above. Any use or reproduction of this work must give proper credit to the original authors.

Author hereby endorse the authenticity and originality of the work presented in this project report and confirm that it meets the academic standards and requirements set forth by the National Institute of Information Technology (NIIT).

ACKNOWLEDGEMENT

The author would like to acknowledge the completion of the insightful paper entitled "Web Application on Debian 12 Bookworm Using Nginx and MySQL." This paper comprehensively discusses the configuration of web server applications within the Debian 12 Bookworm operating system, focusing on services such as Nginx, PHP-FPM, and MySQL, and their role in servicing within the infrastructure.

The contents of this paper provide a detailed overview of configuring web server applications on the Debian 12 platform. The authors have meticulously examined various aspects of web server technology, including configuration and optimization.

Furthermore, the paper explores the challenges associated with implementing web server applications in the Debian 12 environment, offering valuable insights for future research and development in this area. Overall, the paper serves as a significant contribution to the growing body of knowledge on web server applications in the context of the Debian 12 Bookworm operating system.

Depok, 30 April 2024

Authors

SYSTEM ANALYSIS

This paper "Web Server Application on Debian 12 Bookworm" delves into the integration of web server applications. The study aims to research, configuring the web server architecture and resources within this particular environment.

The concept of this configuration is to develop a fully functional demo web application utilizing Nginx services. Users can register an account or log in using MySQL database services. Additionally, PHP-FPM is required to facilitate the connection between the web application and the database services.

To establish all the configurations, the adapters must be configured first. We'll primarily use a NAT adapter and then set up port forwarding from the Debian virtual machine to the host OS with both DHCP IP addresses.

Ultimately, the objective of this configuration is to construct a fully operational demo web application leveraging Nginx services and MySQL database services. Furthermore, PHP-FPM plays a crucial role in establishing seamless communication between the web application and the database services.

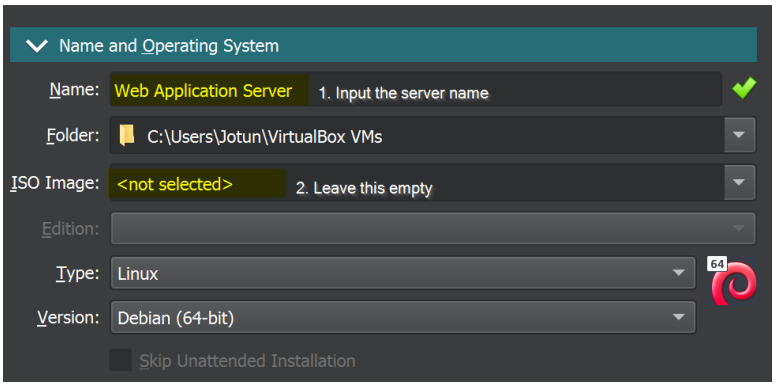
PREPARATION

Before commencing with the configuration process, it is essential to prepare the Debian virtual machine. This can be achieved in two ways: Firstly, by installing Debian using an ISO file, following the **recommended** tutorial provided at https://linuxways.net/debian/how-to-install-debian-12-on-virtualbox/.

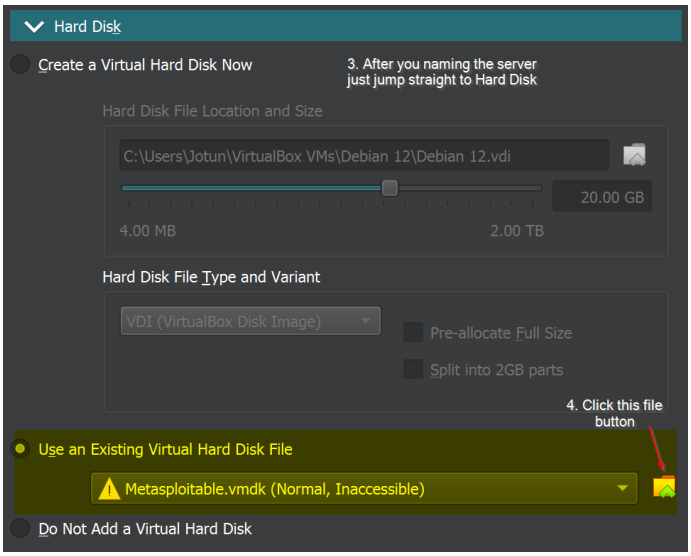
Alternatively, you can utilize the built-in VM machine, which provides a ready-to-use environment without the need for an installation process. You can download it from the following link: https://www.osboxes.org/debian/.

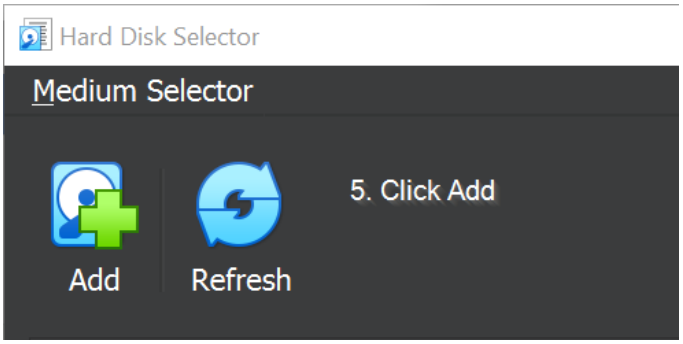
If readers choose the second option, follow these steps:

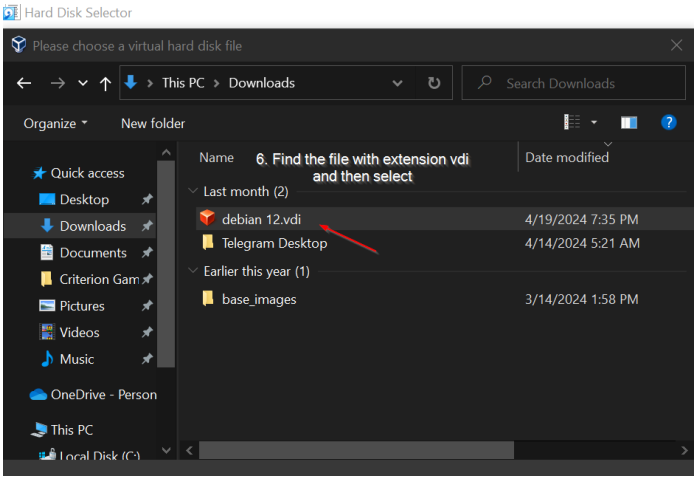
Within VirtualBox, navigate to the "**Machine**" menu and select "**New**." Then, follow the instructions depicted in the image below:



PREPARATION







SERVICES

1. **Secure Shell**

   A tool is used to securely connect to a remote computer over a network. This allows another computer to be logged into and controlled as if it were being operated right in front of it, all while ensuring that data is kept encrypted.

2. **Nginx**

   Free and open-source software, Nginx (pronounced "engine-ex"), primarily functions as a web server, allowing users to host and serve web content efficiently.

3. **PHP-Fpm**

   PHP FastCGI Process Manager, is a software tool that handles the execution of PHP scripts on a web server.

4. **MariaDB/MySQL**

   Open-source relational databases (RDBMS) that store and manage data in a structured format using tables with rows and columns.

5. **Port Forwarding**

   Port forwarding is a technique used in computer networking to allow external devices or users to access services running on a private network.

SERVICES SETUP

1. Installing Services in Debian 12

First, log onto the server, and it's recommended to **update** the repository before proceeding with any installations. For the reading guide, green indicates system responses, while white indicates user input.

```
============================================================
Debian GNU/Linux 12 webserver tty1
rallserver login: rall
password: (your debian password mine is) “red”
rall@webserver:~$ sudo apt update
============================================================
```

Next, we proceed with installing all the required services, starting from **SSH** and continuing through to **MariaDB**.

```
============================================================
rall@webserver:~$ sudo apt install ssh
The following NEW packages will be installed:
  openssh-server openssh-usermap.
Accept [Y/n]? Y
Downloading... [Download progress bar]
Successfully installed openssh-server openssh-usermap.
============================================================
```

SERVICES SETUP

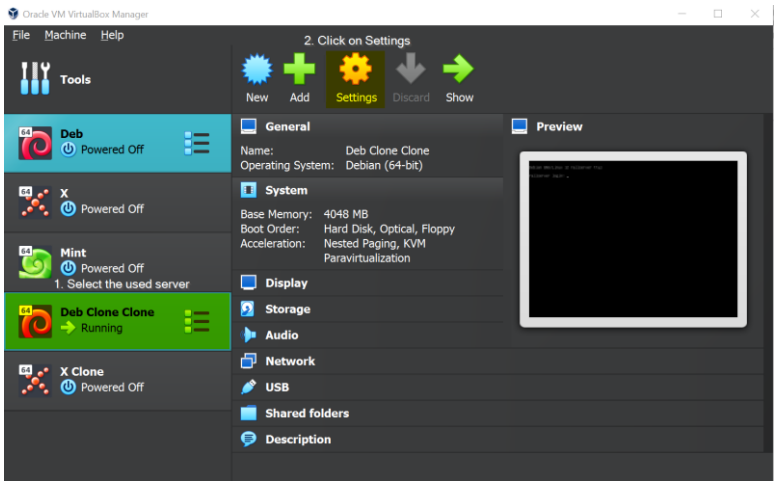
```
============================================================
rall@webserver:~$ sudo apt install nginx
The following NEW packages will be installed:
  nginx nginx-common nginx-extras nginx-full
Accept [Y/n]? Y
Downloading... [Download progress bar]
Successfully installed nginx nginx-common nginx-extras nginx-full.
rall@rallserver:~$ sudo apt install php8.2-fpm
The following NEW packages will be installed:
  php8.2-fpm
Accept [Y/n]? Y
Downloading... [Download progress bar]
Successfully installed php8.2-fpm
rall@rallserver:~$ sudo apt install mariadb
The following NEW packages will be installed:
  mariadb-client mariadb-common mariadb-config mariadb-server
Accept [Y/n]? Y
Downloading... [Download progress bar]
Successfully mariadb-client mariadb-common mariadb-config mariadb-server
============================================================
```

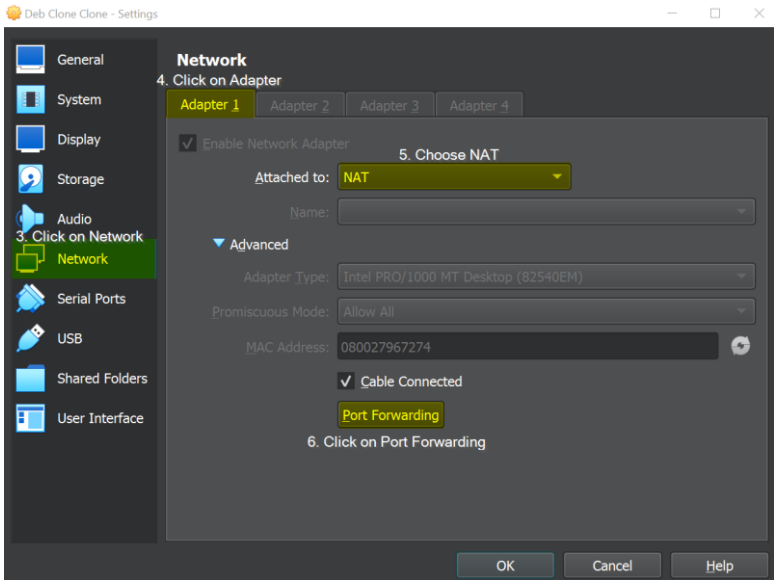
After the installation is completed, All services will be tested. Once all services confirmed to be running, **main configuration** will be performed through **SSH.**

SERVICES SETUP

2. Checking and Testing All Services

Before proceeding with checking and testing, the virtual machine network settings adapter should be configured to **NAT**. Additionally, **port forwarding** from the local virtual network to the local host network needs to be set up.





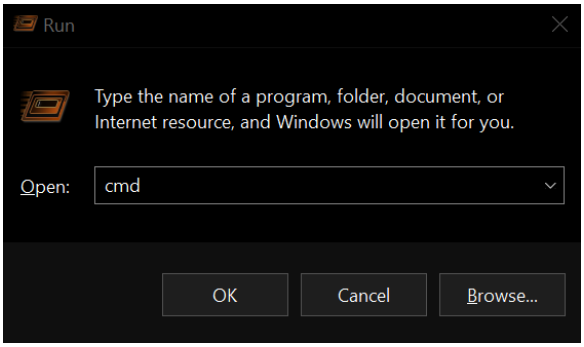
SERVICES SETUP

Port forwarding should be configured as per the following instructions below.

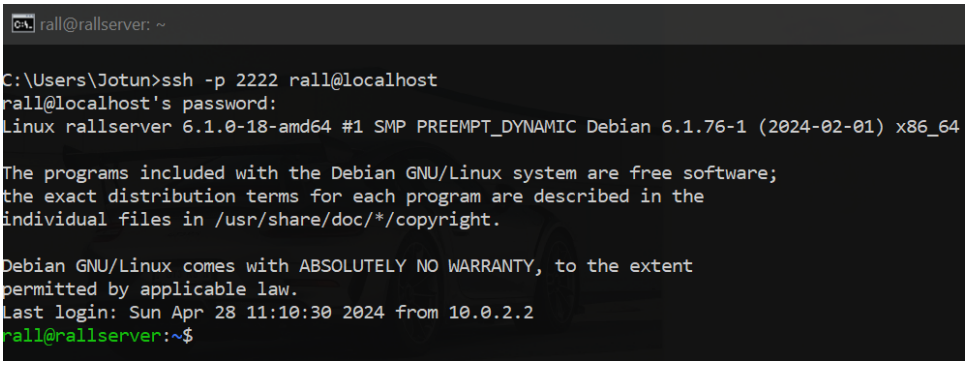
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
NAT SSH	TCP	0.0.0.0	2222	0.0.0.0	22
WEBSERVER	TCP	0.0.0.0	80	0.0.0.0	80

Now, lets test the **Secure Shell**

Secure Shell



Open command line on windows

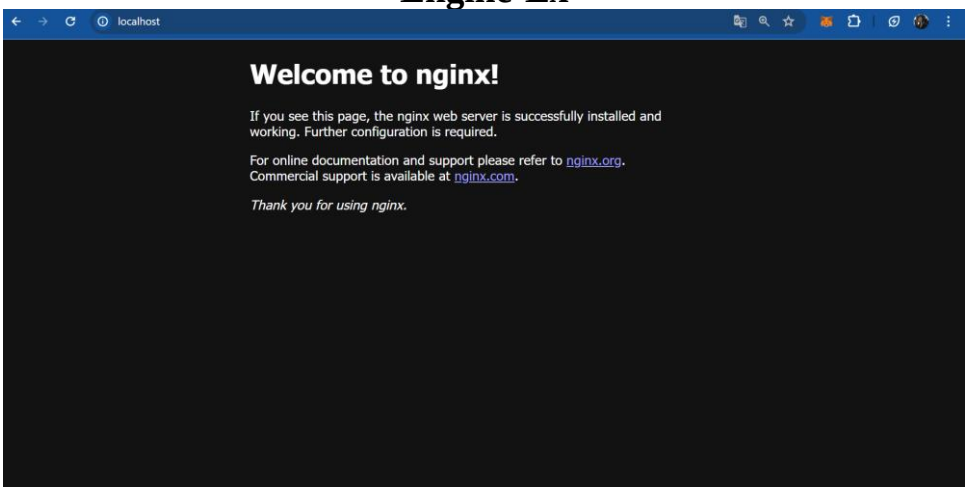


The proof of secure shell is running

SERVICES SETUP

Now, the **Nginx** service will be tested by accessing **localhost** via a web browser.

Engine-Ex



Nginx now running on localhost

Next, **PHP** and **MySQL** services will be tested to confirm their activation status.

```
==========================================================
rall@webserver:~$ sudo systemctl status php8.2-fpm | grep active
● Active: active (running) since Sun 2024-04-28 15:10:59 +07; 12min ago
   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
rall@webserver:~$ sudo systemctl status nginx | grep active
● Active: active (running) since Sun 2024-04-28 15:10:59 +07; 12min ago
==========================================================
```

Now that we've confirmed all services are running fine, let's proceed with some **main configurations**.

MAIN CONFIGURATION

3. Creating Databases in MySQL for Nginx Webservices

Login to the **MySQL** user **root** with an **empty password** by pressing enter. Once inside the service, proceed to change the root password and create a database for the webserver along with the tables.

```
============================================================
rall@webserver:~$ sudo mysql -u root -p
Enter password:
MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0.011 sec)
MariaDB [(none)]> CREATE DATABASE webserver;
Query OK, 1 row affected (0.000 sec)
MariaDB [(none)]> SHOW DATABASES;
+--------------------------+
|        Database          |
+--------------------------+
| information_schema       |
| mysql                    |
| performance_schema       |
| sys                      |
| webserver                |
+--------------------------+
5 rows in set (0.001 sec)
MariaDB [(none)]> USE webserver;
Database changed
MariaDB [webserver]> CREATE TABLE username (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> username VARCHAR (50) NOT NULL,
    -> password VARCHAR (100) NOT NULL);
Query OK, 0 rows affected (0.007 sec)
============================================================
```

MAIN CONFIGURATION

4. Configuring Nginx to Connect to Database

Configuration within default files in **Nginx** is crucial. Let's proceed with it.

```
rall@webserver:~$ sudo mkdir /var/www/webserver
rall@webserver:~$ sudo nano /etc/nginx/sites-available/default
```

```
root /var/www/webserver;    < Change from html to webserver

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#                       Remove # and  adjust to image
location ~ \.php$ {
        include snippets/fastcgi-php.conf;
#
#       # With php-fpm (or other unix sockets):
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
#       # With php-cgi (or other tcp sockets):
#       fastcgi_pass 127.0.0.1:9000;      Change PHP version to
}                                          the installed version.
```

```
//CTRL + X, Y, ENTER TO SAVE
```

Next, a **PHP** script will be created for establishing a connection between **MySQL** and **Nginx** services.

MAIN CONFIGURATION

Keep in mind that the script to be created is sourced from a Google reference and is not the original work of the author. If the reader has their own version, they are welcome to use it.

```
rall@webserver:~$ cd /var/www/webserver
rall@webserver:~$ sudo nano login.php
```

```php
<?php
// Database connection
$servername = "localhost"; // Change this to your MySQL server
$username = "root"; // Change this to your MySQL username
$password = "123"; // Change this to your MySQL password
$database = "webserver"; // Change this to your MySQL database name

$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Form submission handling
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = $_POST["password"];

    // SQL query to check if the username and password match
    $sql = "SELECT * FROM username WHERE username='$username' AND password='$password'";
    $result = $conn->query($sql);

    if ($result->num_rows == 1) {
        // Login successful
        echo "Login successful!";
    } else {
        // Login failed
        echo "Invalid username or password!";
    }
}

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

read the comment for instruction

change to username

#Creating login connection to database#

MAIN CONFIGURATION

rall@webserver:~$ sudo nano register.php

```php
<?php
// Database connection
$servername = "localhost"; // Change this to your MySQL server
$username = "root"; // Change this to your MySQL username
$password = "123"; // Change this to your MySQL password
$database = "webserver"; // Change this to your MySQL database name

$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Form submission handling
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = $_POST["password"];

    // SQL query to insert new user into the database
    $sql = "INSERT INTO username (username, password) VALUES ('$username', '$password')";

    if ($conn->query($sql) === TRUE) {
        echo "Registration successful!";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}

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

#Creating register connection to database#

Now that configuration for connecting the web server and the database has been completed, the next step is to create an **HTML** page for it.

MAIN CONFIGURATION

5. Creating HTML Page for Web Application

Once again, it's important to note that the script is not original from the author, it's sourced from a Google reference. The reader can create the HTML page as they wish.

```
rall@webserver:~$ sudo nano index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login or Register</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f7f7f7;
    }
    .container {
      max-width: 600px;
      margin: 50px auto;
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
    h1 {
      text-align: center;
      color: #333;
    }
    p {
      text-align: center;
      color: #666;
      margin-bottom: 20px;
    }
```

#Creating index pages#

MAIN CONFIGURATION

```
============================================================

ul {
            list-style-type: none;
            padding: 0;
            text-align: center;
        }
        li {
            margin-bottom: 10px;
        }
        a {
            display: inline-block;
            padding: 10px 20px;
            background-color: #007bff;
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            transition: background-color 0.3s;
        }
        a:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to Our Website</h1>
        <p>Please select an option:</p>
        <ul>
            <li><a href="login.html">Login</a></li>
            <li><a href="register.html">Register</a></li>
        </ul>
    </div>
</body>
</html>
============================================================
```

#Creating index pages#

MAIN CONFIGURATION

```
==========================================================

rall@webserver:~$ sudo nano login.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="styles.css"> <!-- Optional: Add your CSS file here -->
</head>
<body>
    <div class="container">
        <h2>Login</h2>
        <form action="login.php" method="post">
            <div class="form-group">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Login</button>
        </form>
    </div>
</body>
</html>
==========================================================

                    #Creating login pages#
```

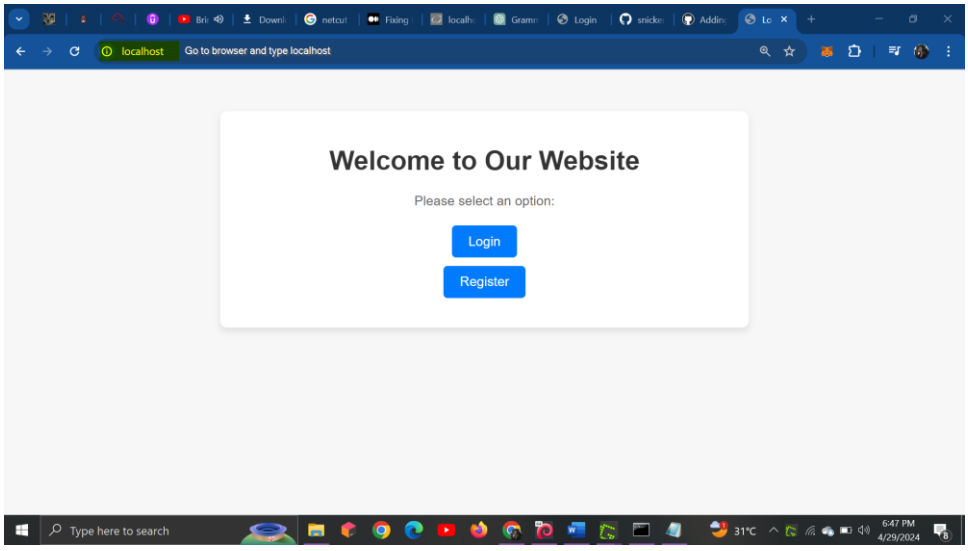
MAIN CONFIGURATION

```
============================================================

rall@webserver:~$ sudo nano register.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
    <link rel="stylesheet" href="styles.css"> <!-- Optional: Add your CSS file here -->
</head>
<body>
    <div class="container">
        <h2>Register</h2>
        <form action="register.php" method="post">
            <div class="form-group">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Register</button>
        </form>
    </div>
</body>
</html>
============================================================

                    #Creating register pages#
```

SIMULATION

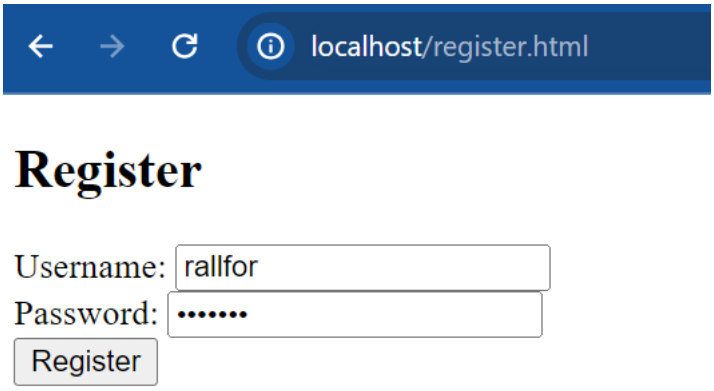
6. Simulating the Web Application

Finally, it's time to thoroughly test the web application that has been both created and configured. To begin, open your browser and type "**localhost**" into the address bar.



Index Pages

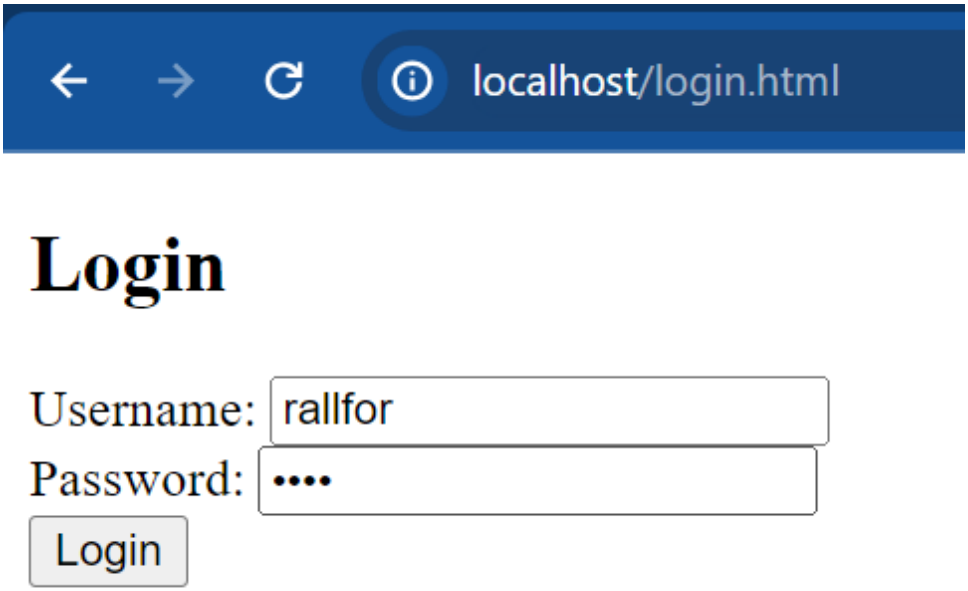
Following that, attempt to register a new user.



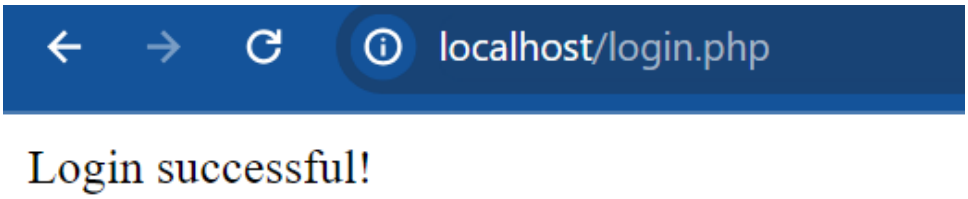
Register Pages

SIMULATION

After registering, proceed to log on to the web app.



Login Pages



Login Success

SIMULATION

For confirmation, let's check the backend database in MySQL for the user "rallfor."

```
MariaDB [webserver]> SELECT * FROM username;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  1 | rallfor  | real     |
+----+----------+----------+
1 row in set (0.001 sec)

MariaDB [webserver]>
```

End Result

In conclusion, after all the configuration carried out by the author, the web application is now ready. However, it's essential to bear in mind that this is merely a demonstration of a real web application. Real-world applications are typically more complicated and extensive than this.

REQUIREMENTS

Hardware :

1. Lenovo V14 G2

Operating System :

1. Windows 10 64-bit
2. Debian 12 Bookworm

Software :

1. Virtual Box VMs
2. Ms. Word
3. Google Chrome

PROJECT FILE DETAILS

No	Filename	Remarks
1	2CS1 Project 2.pdf	Microsoft Words contain research paper about the project
2	Deb Clone.vdi	Server file contains the webapp configuration
3	Project 2 Presentation.pptx	Presentation file