Here’s a comprehensive step-by-step guide to create a login and registration page using Python on two Ubuntu machines in your local environment.

**Project Overview**

1. **Machine 1**: Hosts the frontend and backend (Flask application).
2. **Machine 2**: Hosts the MySQL database.

**Step 1: Set Up Machine 2 (MySQL Database Server)**

**1.1 Install MySQL Server**

1. **Open a terminal on Machine 2** and update the package list:

bash

Copy code

sudo apt update

1. **Install MySQL server**:

bash

Copy code

sudo apt install mysql-server -y

**1.2 Secure MySQL Installation**

Run the MySQL security script to configure security settings:

bash

Copy code

sudo mysql\_secure\_installation

* **Set a root password** and follow the prompts to configure security settings (remove anonymous users, disallow remote root login, etc.).

**1.3 Configure MySQL for Remote Access**

1. **Edit the MySQL configuration** file:

bash

Copy code

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

1. **Update the bind address**:
   * Find the line:

plaintext

Copy code

bind-address = 127.0.0.1

* + Change 127.0.0.1 to 0.0.0.0 to allow connections from any IP address:

plaintext

Copy code

bind-address = 0.0.0.0

1. **Save and close** the file (press CTRL+X, then Y, and Enter).

**1.4 Restart MySQL**

Restart MySQL for changes to take effect:

bash

Copy code

sudo systemctl restart mysql

**1.5 Create the Database and User**

1. **Log in to MySQL**:

bash

Copy code

sudo mysql -u root -p

1. **Create the database**:

sql

Copy code

CREATE DATABASE login\_app;

1. **Create a user and grant privileges**:

sql

Copy code

CREATE USER 'app\_user'@'%' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON login\_app.\* TO 'app\_user'@'%';

FLUSH PRIVILEGES;

1. **Exit the MySQL shell**:

sql

Copy code

EXIT;

**1.6 Allow MySQL Through the Firewall**

bash

Copy code

sudo ufw allow 3306/tcp

**1.7 Verify Remote Connection (From Machine 1)**

1. On Machine 1, check if you can connect to MySQL:

bash

Copy code

mysql -u app\_user -h <Machine2\_IP> -p login\_app

1. Enter the password (password). You should see the MySQL prompt if the connection is successful.

**Step 2: Set Up Machine 1 (Flask Application with Frontend and Backend)**

**2.1 Install Python and Required Packages**

1. **Update the package list**:

bash

Copy code

sudo apt update

1. **Install Python and pip**:

bash

Copy code

sudo apt install python3 python3-pip -y

1. **Install Flask and MySQL Connector**:

bash

Copy code

pip3 install flask mysql-connector-python

**2.2 Set Up Project Directory**

1. **Create a directory for the project**:

bash

Copy code

mkdir ~/login\_app

cd ~/login\_app

**2.3 Create the Flask Application**

1. Create a new file called app.py inside ~/login\_app:

bash

Copy code

nano app.py

1. Copy and paste the following code into app.py:

python

Copy code

from flask import Flask, render\_template, request, redirect, url\_for

import mysql.connector

app = Flask(\_\_name\_\_)

db\_config = {

'user': 'app\_user',

'password': 'password',

'host': '<Machine2\_IP>',

'database': 'login\_app'

}

def get\_db\_connection():

return mysql.connector.connect(\*\*db\_config)

@app.route('/register', methods=['GET', 'POST'])

def register():

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute('INSERT INTO users (username, password) VALUES (%s, %s)', (username, password))

conn.commit()

cursor.close()

conn.close()

return redirect(url\_for('login'))

return render\_template('register.html')

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute('SELECT \* FROM users WHERE username = %s AND password = %s', (username, password))

user = cursor.fetchone()

cursor.close()

conn.close()

if user:

return 'Login successful'

else:

return 'Invalid credentials'

return render\_template('login.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

1. **Save and close** the file (press CTRL+X, then Y, and Enter).

**2.4 Create HTML Templates**

1. **Create a folder for HTML templates**:

bash

Copy code

mkdir templates

1. **Create register.html**:

bash

Copy code

nano templates/register.html

1. Paste the following HTML code into register.html:

html

Copy code

<h2>Register</h2>

<form method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<br>

<input type="submit" value="Register">

</form>

1. **Create login.html**:

bash

Copy code

nano templates/login.html

1. Paste the following HTML code into login.html:

html

Copy code

<h2>Login</h2>

<form method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<br>

<input type="submit" value="Login">

</form>

**2.5 Create the Database Table**

1. Connect to MySQL on Machine 2 from Machine 1:

bash

Copy code

mysql -u app\_user -h <Machine2\_IP> -p login\_app

1. Enter the password password and create the users table:

sql

Copy code

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL

);

1. **Exit** the MySQL shell:

sql

Copy code

EXIT;

**2.6 Run the Flask Application**

1. On Machine 1, start the Flask application:

bash

Copy code

python3 app.py

**Step 3: Access and Test the Application**

1. **Open a browser** on any device connected to the local network.
2. **Access the Registration Page**:
   * Go to http://<Machine1\_IP>:5000/register
   * Register a new user by entering a username and password.
3. **Access the Login Page**:
   * Go to http://<Machine1\_IP>:5000/login
   * Log in with the credentials you just registered.
4. **Verify Data**:
   * On Machine 2, log in to MySQL and check the users table to confirm that data is saved correctly:

sql

Copy code

SELECT \* FROM users;

With these steps, you’ve successfully set up a basic login and registration application using Python, Flask, and MySQL on two machines!

4o