To build a login and signup page using PHP, HTML, CSS, and MySQL as specified in your task, follow these steps. I'll provide a detailed guide with code examples for each part of the process.

**Step 1: Setup the Database**

First, you'll need to create a MySQL database to store user credentials. Here's a simple example of how to create a database and a user table.

1. **Create Database**:
2. CREATE DATABASE userDB;
3. **Create User Table**:
4. USE userDB;
5. CREATE TABLE users (
6. id INT(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY,
7. username VARCHAR(50) NOT NULL,
8. password VARCHAR(255) NOT NULL,
9. email VARCHAR(100) NOT NULL
10. );

**Step 2: HTML and CSS for Frontend**

Next, create the HTML for the login and signup forms, and style them using CSS.

**HTML - 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 and Signup</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="form-container">

<form action="login.php" method="post">

<h2>Login</h2>

<input type="text" name="username" placeholder="Username" required>

<input type="password" name="password" placeholder="Password" required>

<button type="submit">Login</button>

</form>

<form action="signup.php" method="post">

<h2>Signup</h2>

<input type="text" name="username" placeholder="Username" required>

<input type="password" name="password" placeholder="Password" required>

<input type="email" name="email" placeholder="Email" required>

<button type="submit">Signup</button>

</form>

</div>

</body>

</html>

**CSS - styles.css**

body {

font-family: Arial, sans-serif;

}

.form-container {

width: 300px;

margin: auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 2px 2px 8px #aaa;

}

form {

display: flex;

flex-direction: column;

}

input, button {

margin-bottom: 10px;

padding: 10px;

border-radius: 5px;

}

button {

cursor: pointer;

background-color: blue;

color: white;

}

**Step 3: PHP for Backend**

Create PHP scripts to handle login and signup functionality.

**Login - login.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "userDB";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$user = $\_POST['username'];

$pass = $\_POST['password'];

$sql = "SELECT \* FROM users WHERE username='$user' AND password='$pass'";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

echo "Login successful";

} else {

echo "Invalid username or password";

}

$conn->close();

?>

**Signup - signup.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "userDB";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$user = $\_POST['username'];

$pass = $\_POST['password'];

$email = $\_POST['email'];

$sql = "INSERT INTO users (username, password, email) VALUES ('$user', '$pass', '$email')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

**Notes:**

* Ensure your server supports PHP and MySQL.
* Replace "root" and "" in the PHP scripts with your actual database username and password.
* Use proper security measures in production, such as using prepared statements with MySQLi to prevent SQL injection and hashing passwords before storing them in the database.

This setup gives you a basic login and signup system. You can extend and improve upon it as needed!

Based on the task description you provided for "Task3: Login Signup Page" built with PHP, HTML, CSS, and MySQL, here's a textual outline of what the output would look like when the forms are processed correctly:

**Login Form Output:**

1. **Successful Login:**
   * If the username and password entered in the login form are correct and match an entry in the database, the output will be:
   * Login successful
2. **Failed Login:**
   * If the username and password do not match any entry in the database, the output will be:
   * Invalid username or password

**Signup Form Output:**

1. **Successful Signup:**
   * If the new user's information (username, password, email) is successfully inserted into the database, the output will be:
   * New record created successfully
2. **Failed Signup:**
   * If there is an error during the insertion process (such as a duplicate username, depending on database constraints), the output will include the error message from the MySQL database. This might look like:
   * Error: [error details from the MySQL server]

These outputs are generated by the PHP scripts login.php and signup.php after processing the form data. Here's a recap of how each script contributes to these outputs:

* **login.php**: Checks the provided username and password against the database. If a match is found, it confirms successful login; otherwise, it reports an invalid login attempt.
* **signup.php**: Attempts to insert the new user's data into the database. If successful, it confirms that a new record was created. If not, it provides an error message detailing the issue.

**Note:**

* For actual production use, it's crucial to handle errors and user feedback appropriately to ensure a good user experience and security. For example, using session management for logged-in users and providing more user-friendly error messages.
* Always use hashed passwords in your database and PHP's built-in functions like password\_hash() for hashing and password\_verify() for checking passwords.

This explanation assumes the backend processing is set up as described and that the database is configured correctly. If you have the scripts running on a server and need specific help with any errors or outputs, you can provide the error messages or further details for more tailored assistance.