Create a .env file in the same directory as your PHP script with the content provided

Replace the placeholder values in the .env file with your actual database credentials.

Ensure the .env file is not accessible from the web (you can use .htaccess rules to prevent access).

Make sure your web server has read permissions for the .env file.

This approach offers several advantages:

- Improved security: Sensitive information is kept out of the main code file.
- Better maintainability: You can easily update configuration without touching the main code.
- Environment-specific configuration: You can have different .env files for development, staging, and production environments.

The rest of the functionality remains the same as in the previous version:

- Role-based access (admin vs user)
- Secure login and session management
- Least privilege principle implementation
- Pop-up messages for successful logins

Remember to keep your .env file secure and never commit it to version control systems. It's common practice to include a .env.example file in your project with dummy values as a template.

// .env Configuration File

```
DB_HOST=localhost
DB_NAME=your_database
DB_USER=your_db_user
DB_PASS=your_db_password
```

//index.php

```php
<?php
session_start();

// Load environment variables
function loadEnv($path) {
    if (file_exists($path)) {
        $lines = file($path, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
        foreach ($lines as $line) {
            if (strpos($line, '=') !== false) {
                list($key, $value) = explode('=', $line, 2);
                $key = trim($key);
                $value = trim($value);
                putenv("$key=$value");
                $_ENV[$key] = $value;
            }
        }
    }
}

// Load .env file
loadEnv(__DIR__ . '/.env');

// Database connection function
function connectDB() {
    try {
        $pdo = new PDO("mysql:host=" . getenv('DB_HOST') . ";dbname=" . getenv('DB_NAME'),
                    getenv('DB_USER'),
                    getenv('DB_PASS'));
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $pdo;
    } catch (PDOException $e) {
        die("Database connection failed: " . $e->getMessage());
    }
}

// User login function
function loginUser($username, $password) {
    $pdo = connectDB();

    $stmt = $pdo->prepare("SELECT id, username, password, role FROM users WHERE
username = :username LIMIT 1");
    $stmt->bindParam(':username', $username, PDO::PARAM_STR);
    $stmt->execute();
```

```
    if ($user = $stmt->fetch(PDO::FETCH_ASSOC)) {
        if (password_verify($password, $user['password'])) {
            return $user;
        }
    }

    return false;
}

// Process login
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);
    $password = $_POST['password'];

    $user = loginUser($username, $password);

    if ($user) {
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];
        $_SESSION['role'] = $user['role'];

        $message = ($user['role'] === 'admin') ? "You are logged in as admin" : "You are logged in
as user";
        echo "<script>alert('$message');</script>";
    } else {
        echo "<script>alert('Invalid username or password');</script>";
    }
}

// Check if user is logged in
function isLoggedIn() {
    return isset($_SESSION['user_id']);
}

// Check if user is admin
function isAdmin() {
    return isset($_SESSION['role']) && $_SESSION['role'] === 'admin';
}

// Logout function
function logout() {
    session_unset();
    session_destroy();
    header("Location: login.php");
```

```
    exit();
}

// Handle logout
if (isset($_GET['logout'])) {
    logout();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>User Login</title>
</head>
<body>
    <?php if (isLoggedIn()): ?>
        <h1>Welcome, <?php echo htmlspecialchars($_SESSION['username']); ?></h1>
        <p>You are logged in as: <?php echo htmlspecialchars($_SESSION['role']); ?></p>
        <?php if (isAdmin()): ?>
            <h2>Admin Panel</h2>
            <!-- Add admin-specific content here -->
        <?php else: ?>
            <h2>User Dashboard</h2>
            <!-- Add user-specific content here -->
        <?php endif; ?>
        <a href="?logout">Logout</a>
    <?php else: ?>
        <h1>Login</h1>
        <form method="post" action="">
            <input type="text" name="username" required placeholder="Username">
            <input type="password" name="password" required placeholder="Password">
            <input type="submit" value="Login">
        </form>
    <?php endif; ?>
</body>
</html>
```