

```
name="password" id="password" required /><br />
<input type="submit" value="Login">
</form>
```

In addition to the code mentioned, Example 7-2 also includes a try catch block to catch the exceptions thrown and a call to a user log file to record successful logs in to the system.

## JSON Data

To use JSON data instead of XML data, Example 7-2 would only need to include the changes shown in Chapter 6. The userid and password JSON data would also need to be formatted as shown.

```
{"user":
[
{"userid": "Fredfred", "password": "$2y$10$VosI32FejL.b0MaCjGbBp.Jre6Ipa.tLYQrVqj9kiVpef5zZ25qQK"},
{"userid": "Petepete", "password": "$2y$10$FdbXxIVXmVOHtaBNxB8vzupRBjFCqUyOTJXr1pNdrLOHKQ\U.jFHO"}
] }
```

## MySQL Data

It is more common and usually more secure to store user ID and password information in a database. Databases can be secured using user IDs and passwords, which can also include levels of access (read only, read, and write) to the information. Even with this level of security, the password should still be encrypted.

Very few minor changes are needed to the MySQL constructor example from Chapter 6 to accomplish authentication.

```
$mysqli =mysqli_connect($server, $db_username, $db_password, $database);

if (mysqli_connect_errno())
{
    throw new Exception("MySQL connection error: " . mysqli_connect_error());
}

$sql="SELECT * FROM Users"; // Change the table used
$result=mysqli_query($con,$sql);

If($result===null)
{
    throw new Exception("No records retrieved from Database");
}

$valid_useridpasswords = mysqli_fetch_assoc($result); // change the array used

mysqli_free_result($result);

mysqli_close($con);
```