

```

<input type="hidden" name="dog_app" id="dog_app" value="dog" />
Select Your Dog's Breed <div id="AjaxResponse"></div><br />
<input type="submit" value="Click to create your dog" />
</form>
</div>
<noscript>
<div id="noJS">
<form method="post" action="e5adog_interface.php">
<h2>Please complete ALL fields. Please note the required format of information.</h2>
Enter Your Dog's Name (max 20 characters, alphabetic) <input type="text" pattern="[a-zA-Z ]*"
title="Up to 20 Alphabetic Characters" maxlength="20" name="dog_name" id="dog_name"
required/><br /><br />
Select Your Dog's Color:<br />
<input type="radio" name="dog_color" id="dog_color" value="Brown">Brown<br />
<input type="radio" name="dog_color" id="dog_color" value="Black">Black<br />
<input type="radio" name="dog_color" id="dog_color" value="Yellow">Yellow<br />
<input type="radio" name="dog_color" id="dog_color" value="White">White<br />
<input type="radio" name="dog_color" id="dog_color" value="Mixed" checked >Mixed<br /><br />
Enter Your Dog's Weight (numeric only) <input type="number" min="1" max="120" name="dog_
weight" id="dog_weight" required /><br /><br />
Enter Your Dog's Breed (max 35 characters, alphabetic) <input type="text" pattern="[a-zA-Z ]*"
title="Up to 15 Alphabetic Characters" maxlength="35" name="dog_breed" id="dog_breed"
required /><br />
<input type="hidden" name="dog_app" id="dog_app" value="dog" />
<input type="submit" value="Click to create your dog" />
</form>
</div>
</noscript>
</body>
</html>
<?php
}
?>

```

In Example 7-1, the verification code is placed at the top of the program. The `else` statement must include all the code to execute if the user is signed into the system. Since the code for this program is HTML (and CSS) code, the `if` statement must be wrapped around the existing code. The closing bracket (`s`) of the `else` statement is shifted to the bottom of the code (after all HTML tags).

PHP allows you to close your PHP code (via `?>`) and reopen your PHP code (via `<?php`) as many times as required. In this example, the PHP code is closed at the top of the program (in the `else` statement) just before the closing bracket. The PHP code is then reopened at the bottom of the code to include a single closing bracket, which closes the PHP `else` statement. This wraps the `else` statement around all the existing code. The users can now only access this section of the code if they are logged in.

Since PHP code is now included in the lab program, the file ending must be changed from `.html` to `.php`. Otherwise the server would not execute the PHP code.

Now let's look at how you can populate the session properties by creating a login program. First let's look at the HTML to request information from the users.

```

<form method="post" action="">
Username: <input type="text" pattern=".{8,}" title="Userid must contain eight or more
characters." name="username" id="username" required/><br />

```