

### Step 1: Document Declaration

```
<!DOCTYPE html>
```

```
<html lang="en">
```

- Declares this as an **HTML5 document**.
  - `lang="en"` specifies the language as English for accessibility and SEO.
- 

### Step 2: Head Section

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Admin Login</title>
```

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
```

```
</head>
```

1. `<meta charset="utf-8">` → ensures support for all characters and symbols.
  2. `<title>` → sets the browser tab title as **Admin Login**.
  3. `<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">`
    - Links the **CSS file** for styling.
    - `{{ url_for('static', filename='style.css') }}` → Flask template function that generates the correct URL to `style.css`.
- 

### Step 3: Body Section

```
<body class="bg">
```

```
<div class="auth-container">
```

- `<body>` contains all visible content.
  - `class="bg"` → applies background styling from CSS.
  - `<div class="auth-container">` → container for the login form (centered box styling in CSS).
- 

### Step 4: Header

```
<h2>Admin Sign In</h2>
```

- Displays the heading for the page.
  - Clearly indicates it's an **Admin login form**.
-

### Step 5: Flash Messages (Flask)

```
{% with messages = get_flashed_messages(with_categories=true) %}

{% if messages %}

{% for cat,msg in messages %}

    <div class="flash {{ cat }}">{{ msg }}</div>

{% endfor %}

{% endif %}

{% endwith %}
```

- Flask feature to display **temporary messages** (success/error).
  - `get_flashed_messages(with_categories=true)` → fetches messages stored on the server.
  - Loops through each message and displays it in a `<div>` with a class for styling (e.g., flash error, flash success).
- 

### Step 6: Login Form

```
<form method="post">

<input name="username" placeholder="Admin username" required>

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

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

</form>
```

1. `<form method="post">` → submits form data securely to the Flask server.
  2. **Username Input:**
    - `<input name="username" placeholder="Admin username" required>`
    - Placeholder gives a hint, and required ensures field isn't empty.
  3. **Password Input:**
    - `<input name="password" type="password" placeholder="Admin password" required>`
    - `type="password"` hides the entered text.
  4. **Submit Button:**
    - `<button type="submit">Login</button>` → submits the form.
- 

### Step 7: Closing Tags

```
</div>
```

</body>

</html>

- Closes the container, body, and HTML document.
- 

### Summary of Flow

1. Admin opens the login page → sees a **centered login form**.
2. Admin enters username and password.
3. Admin clicks **Login** → form sends a **POST request** to the Flask server.
4. Flask server validates credentials:
  - If valid → logs in admin.
  - If invalid → sends a **flash message**, displayed above the form.