frameboxx 2.0®
animation | visual effects | programming | coding
Premier Academy for IT Development

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

Lecture 4: Forms and User Input

Theory

1. HTML Forms with Flask

HTML forms allow users to send data to the server (for example: submitting feedback, registration, or login).

In Flask, forms are handled by defining a route that processes form data using **GET** or **POST** methods.

Example HTML Form:

```
<form method="POST" action="/submit">
  <label>Name:</label>
  <input type="text" name="username">
  <input type="submit" value="Submit">
</form>
```

---

2. Handling GET and POST Requests

By default, Flask routes respond to **GET** requests.
If you want a route to handle form submissions, you must enable **POST**.

Example:

```
@app.route('/submit', methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        return "Data received using POST"
    else:
        return "Send data using POST method"
```

Method	Description	Example Use

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

GET	Used to retrieve data	Display a form
POST	Used to send data to the server	Submit form data

---

3. Accessing Form Data using `request.form`

When a form is submitted via POST, Flask stores the form input in the **`request.form`** dictionary.

Example:

```
from flask import request

@app.route('/feedback', methods=['POST'])
def feedback():
    name = request.form['name']
    message = request.form['message']
    return f"Thank you {name}, your feedback: {message}"
```

---

4. Flash Messages

Flash messages provide user-friendly notifications (like “Form submitted successfully!”). They require a **secret key** in the Flask app for session management.

Example:

```
from flask import Flask, render_template, request, flash

app = Flask(__name__)
app.secret_key = 'secret123'

@app.route('/submit', methods=['POST'])
def submit():
    flash("Form submitted successfully!")
    return render_template('form.html')
```

In your template, display flash messages:

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

```
{% with messages = get_flashed_messages() %}
  {% if messages %}
    {% for msg in messages %}
      <p>{{ msg }}</p>
    {% endfor %}
  {% endif %}
{% endwith %}
```

Practical

Objective

Create a simple **Feedback Form** using Flask that accepts user input and displays it on submission.

Step 1: Folder Structure

```
flask_forms_app/
│
├── app.py
│
├── templates/
│   ├── base.html
│   └── feedback.html
│
└── static/
    └── style.css
```

Step 2: Files and Code

File 1: app.py

```
from flask import Flask, render_template, request, flash

app = Flask(__name__)
app.secret_key = 'secretkey123'
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

```
@app.route('/')
def index():
    return render_template('feedback.html')

@app.route('/submit', methods=['POST'])
def submit():
    name = request.form['name']
    message = request.form['message']
    flash(f"Thank you {name}! Your feedback has been received.")
    return render_template('feedback.html', name=name,
message=message)

if __name__ == '__main__':
    app.run(debug=True)
```

File 2: templates/base.html

```
<!DOCTYPE html>
<html>
<head>
    <title>{% block title %}Flask Feedback Form{% endblock
%}</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
    <header>
        <h2>Flask Form Example</h2>
        <hr>
    </header>

    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for msg in messages %}
          <p class="flash">{{ msg }}</p>
        {% endfor %}
      {% endif %}
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

```
    {% endwith %}

    {% block content %}{% endblock %}
</body>
</html>
```

File 3: templates/feedback.html

```
{% extends "base.html" %}
{% block title %}Feedback Page{% endblock %}
{% block content %}
<h3>Submit Your Feedback</h3>

<form method="POST" action="/submit">
  <label for="name">Name:</label><br>
  <input type="text" name="name" required><br><br>

  <label for="message">Feedback:</label><br>
  <textarea name="message" rows="4" cols="40"
required></textarea><br><br>

  <input type="submit" value="Submit">
</form>

{% if name and message %}
  <hr>
  <h4>Feedback Summary</h4>
  <p><strong>Name:</strong> {{ name }}</p>
  <p><strong>Message:</strong> {{ message }}</p>
{% endif %}
{% endblock %}
```

File 4: static/style.css

```
body {
    font-family: Arial, sans-serif;
    margin: 30px;
}
```

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

```
label {
    font-weight: bold;
}
.flash {
    background-color: #e8f5e9;
    padding: 10px;
    border: 1px solid #81c784;
    color: #2e7d32;
    margin-bottom: 10px;
}
```

Step 3: Run the Application

In terminal:

```
python app.py
```

Open your browser and visit:

```
http://127.0.0.1:5000/
```

Fill out the form → Submit → Your feedback and flash message should appear.

Practice Task

Task 1:

Add a new input field for **Email** and display it with the feedback.

Task 2:

Modify the app to handle both **GET** and **POST** methods in the same route (`/feedback`).

Task 3:

Apply CSS styling to improve the form layout and flash message appearance.

+91 96622 16697
jay@frameboxxers.com
F-201, Shilp Square-B, Nr. Shreeji Tower,
Drive-in Road, Vastrapur, Ahmedabad - 380015

Task 4:

Add validation — if the user leaves a field blank, flash an error message.