

Form Handling in Flask (with Request and Flask-WTF)

1. What is Form Handling in Flask?

Flask allows you to handle user-submitted form data using HTTP methods like GET and POST. You define HTML forms and use Flask's `request` object to access form inputs. For more secure and structured handling, Flask-WTF is used, which integrates Flask with WTForms.

2. Basic HTML Form Example

```
<form action="/login" method="POST">
  <input type="text" name="username" placeholder="Enter username" />
  <input type="password" name="password" placeholder="Enter password" />
  <input type="submit" value="Login" />
</form>
```

3. Handling Forms Using request in Flask

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

```
app = Flask(__name__)
```

```
@app.route("/login", methods=["GET", "POST"])
```

```
def login():
```

```
    if request.method == "POST":
```

```
        username = request.form["username"]
```

```
        password = request.form["password"]
```

```
        return f"Hello, {username}!"
```

```
    return render_template("login.html")
```

4. What is Flask-WTF?

Flask-WTF is a Flask extension that simplifies form creation and handling. It provides automatic CSRF protection, validators, and cleaner syntax for rendering forms.

Form Handling in Flask (with Request and Flask-WTF)

5. Creating a Form Class using FlaskForm

```
from flask_wtf import FlaskForm

from wtforms import StringField, PasswordField, SubmitField

from wtforms.validators import DataRequired, Email


class LoginForm(FlaskForm):
    email = StringField("Email", validators=[DataRequired(), Email()])
    password = PasswordField("Password", validators=[DataRequired()])
    submit = SubmitField("Login")
```

6. Using Flask-WTF Form in Route

```
@app.route("/login", methods=["GET", "POST"])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        return f"Welcome, {form.email.data}"
    return render_template("login.html", form=form)
```

7. Using Flask-WTF Form in HTML Template (Jinja2)

```
<form method="POST" action="{{ url_for('login') }}">
    {{ form.hidden_tag() }}
    {{ form.email.label }}<br>{{ form.email }}<br>
    {{ form.password.label }}<br>{{ form.password }}<br>
    {{ form.submit }}
</form>
```

8. What is CSRF Token and Why We Use It?

CSRF (Cross-Site Request Forgery) is a security threat where unauthorized commands are sent from a user you trust. Flask-WTF includes a CSRF token automatically to prevent these attacks. It must be included in

Form Handling in Flask (with Request and Flask-WTF)

forms using `{{ form.hidden_tag() }}` in templates.

9. Summary

- Use `request.form` to manually handle HTML form inputs.
- Use Flask-WTF for secure, validated forms with CSRF protection.
- Always validate and sanitize user input in production apps.