**[Que-45] - Explain how to use Flask-WTF to create and validate forms in a Flask application**

Flask-WTF is an extension for Flask that integrates Flask with the WTForms library, providing a convenient way to create and validate forms in Flask applications. Here’s a step-by-step guide on how to use Flask-WTF to create and validate forms:

### **Step 1: Install Flask-WTF and Flask**

First, make sure you have Flask and Flask-WTF installed. You can install them using pip:

pip install flask flask-wtf

### **Step 2: Set Up the Flask Application**

Create a new directory for your Flask application and within it, create a file named app.py.

### **Step 3: Create a Form Class**

In your Flask application, create a new Python file named forms.py. This is where you define your form classes using WTForms syntax.

#### **Example Form Class (forms.py):**

from flask\_wtf import FlaskForm  
from wtforms import StringField, PasswordField, SubmitField  
from wtforms.validators import DataRequired, Length, Email, EqualTo  
  
class RegistrationForm(FlaskForm):  
 username = StringField('Username', validators=[DataRequired(), Length(min=4, max=20)])  
 email = StringField('Email', validators=[DataRequired(), Email()])  
 password = PasswordField('Password', validators=[DataRequired()])  
 confirm\_password = PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password')])  
 submit = SubmitField('Sign Up')

### **Step 4: Use the Form in Your Flask Application**

#### **Example Flask Application (app.py):**

python

Copy code

from flask import Flask, render\_template, redirect, url\_for  
from forms import RegistrationForm  
from flask\_wtf import FlaskForm  
  
app = Flask(\_\_name\_\_)  
app.config['SECRET\_KEY'] = 'your\_secret\_key'  
  
@app.route('/register', methods=['GET', 'POST'])  
def register():  
 form = RegistrationForm()  
 if form.validate\_on\_submit():  
 # Process form data when it's valid  
 return redirect(url\_for('success'))  
 return render\_template('register.html', form=form)  
  
@app.route('/success')  
def success():  
 return 'Registration successful!'  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

### **Explanation:**

1. **Import Flask and FlaskForm**:
   * Flask: The core Flask class.
   * FlaskForm: Base class for forms in Flask-WTF.
2. **Import Form Class**:
   * RegistrationForm: Form class defined in forms.py for user registration.
3. **Create an Instance of the Flask Class**:
   * app = Flask(\_\_name\_\_): Create an instance of the Flask class.
4. **Configure Secret Key**:
   * app.config['SECRET\_KEY'] = 'your\_secret\_key': Configure a secret key for CSRF protection and form security.
5. **Define Route for Registration Form**:
   * @app.route('/register', methods=['GET', 'POST']): Define a route for /register that accepts GET and POST requests.
   * def register():: Function that handles registration form requests.
6. **Create an Instance of the Form Class**:
   * form = RegistrationForm(): Create an instance of the RegistrationForm class defined in forms.py.
7. **Validate the Form**:
   * if form.validate\_on\_submit():: Check if the form has been submitted and is valid.
   * return redirect(url\_for('success')): Redirect to a success page if the form is valid.
8. **Render the Form Template**:
   * return render\_template('register.html', form=form): Render the register.html template and pass the form object to it.
9. **Define a Success Route**:
   * @app.route('/success'): Define a route for /success.
   * def success():: Function that displays a success message after successful form submission.

### **Step 5: Create HTML Template**

Create an HTML template (register.html) to render the form fields using Flask-WTF's form object.

#### **Example HTML Template (templates/register.html):**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>User Registration</title>  
</head>  
<body>  
 <h1>Register</h1>  
 <form method="POST">  
 {{ form.hidden\_tag() }}  
 <p>  
 {{ form.username.label }}<br>  
 {{ form.username(size=32) }}  
 </p>  
 <p>  
 {{ form.email.label }}<br>  
 {{ form.email(size=32) }}  
 </p>  
 <p>  
 {{ form.password.label }}<br>  
 {{ form.password(size=32) }}  
 </p>  
 <p>  
 {{ form.confirm\_password.label }}<br>  
 {{ form.confirm\_password(size=32) }}  
 </p>  
 <p>{{ form.submit() }}</p>  
 </form>  
</body>  
</html>

### **Step 6: Run the Flask Application**

1. Open a terminal or command prompt.
2. Navigate to the directory where app.py is located.
3. Run the following command:

python app.py

1. Open a web browser and go to <http://127.0.0.1:5000/register>.

You should see the registration form rendered in your browser. Fill out the form and submit it to see the success message.

### **Additional Form Validation**

The RegistrationForm class in the example above uses validators from WTForms (DataRequired, Length, Email, EqualTo) to perform validation on the form fields. These validators ensure that the data submitted meets the specified criteria (e.g., not empty, correct format).

Flask-WTF handles form CSRF protection automatically when the SECRET\_KEY is set in the Flask application configuration. This helps prevent Cross-Site Request Forgery (CSRF) attacks.