

WP LAB 8– Form processing using Django – Part 2

Name: Pranamya G Kulal

Class: CSE A

Reg no: 220905018

Roll no: 8

Q1) Create a website with two pages. First page contains: RadioButton with HP, Nokia, Samsung, Motorola, Apple as options. CheckBox with Mobile and Laptop as items. TextBox to enter quantity. There is a button with text as "Produce Bill". On Clicking Produce Bill button, item should be displayed with total amount on another page.

i) settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'billing',  
]
```

ii) BillGenerator/urls.py

```
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", include('billing.urls')), # Include app URLs  
]
```

iii) billing/urls.py

```
from django.urls import path  
from . import views  
  
urlpatterns = [  
    path("", views.input_page, name="input_page"), # Home Page  
    path('bill/', views.bill_page, name="bill_page"), # Bill Page  
]
```

iv) views.py

```
from django.shortcuts import render, redirect
```

```
# Define prices for Mobile and Laptop
```

```
PRICES = {  
    "Mobile": 10000, # Base price  
    "Laptop": 50000 # Base price  
}
```

```
def input_page(request):
```

```

if request.method == "POST":
    brand = request.POST.get("brand")
    items = request.POST.getlist("items") # Get multiple values
    quantity = int(request.POST.get("quantity", 0))

    # Store data in session
    request.session["brand"] = brand
    request.session["items"] = items
    request.session["quantity"] = quantity

    return redirect("bill_page") # Redirect to bill page

return render(request, "billing/input_page.html")

def bill_page(request):
    brand = request.session.get("brand", "Unknown")
    items = request.session.get("items", [])
    quantity = request.session.get("quantity", 0)

    total_amount = sum(PRICES[item] for item in items) * quantity # Calculate total price

    return render(request, "billing/bill_page.html", {
        "brand": brand,
        "items": items,
        "quantity": quantity,
        "total_amount": total_amount
    })

```

v) templates/billing/bill_page.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bill Summary</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <h2>Bill Summary</h2>
        <p><strong>Brand:</strong> {{ brand }}</p>
        <p><strong>Items:</strong> {{ items|join:", " }}</p>
        <p><strong>Quantity:</strong> {{ quantity }}</p>
        <h4><strong>Total Amount:</strong> ₹{{ total_amount }}</h4>
        <a href="{% url 'input_page' %}" class="btn btn-success">Generate Another Bill</a>
    </div>
</body>
</html>

```

vi) templates/billing/input_page.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bill Generator</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h2>Bill Generator</h2>
    <form method="post">
      {% csrf_token %}

      <label><strong>Select Brand:</strong></label><br>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="brand" value="HP" required> HP
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="brand" value="Nokia"> Nokia
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="brand" value="Samsung">
Samsung
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="brand" value="Motorola">
Motorola
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="brand" value="Apple"> Apple
      </div>

      <br>
      <label><strong>Select Items:</strong></label><br>
      <div class="form-check">
        <input class="form-check-input" type="checkbox" name="items" value="Mobile">
Mobile (₹10,000)
      </div>
      <div class="form-check">
        <input class="form-check-input" type="checkbox" name="items" value="Laptop">
Laptop (₹50,000)
      </div>

      <br>
      <label for="quantity" class="form-label"><strong>Enter Quantity:</strong></label>
      <input type="number" class="form-control" id="quantity" name="quantity" required>

      <br>
      <button type="submit" class="btn btn-primary">Produce Bill</button>
    </form>
  </div>

```

```
</form>
</div>
</body>
</html>
```

vii) Screenshots

Bill Generator

Select Brand:

- ☐ HP
☐ Nokia
☒ Samsung
☐ Motorola
☐ Apple

Select Items:

- ☒ Mobile (₹10,000)
☒ Laptop (₹50,000)

Enter Quantity:

3

Produce Bill

Bill Summary

Brand: Samsung

Items: Mobile, Laptop

Quantity: 3

Total Amount: ₹180000

Generate Another Bill

Q2) Write a program to display the following feedback form. The different options for the list box must be ASP-XML, DotNET, JavaPro and Unix,C,C++. When the Submit Form button is clicked after entering the data, a message as seen in the last line of the above figure must be displayed.

i) settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
```

```

'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'feedback',
]

```

ii) FeedbackApp/urls.py

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('feedback.urls')), # Include app URLs
]

```

iii) feedback/urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path("", views.feedback_form, name="feedback_form"),
]

```

iv) views.py

```

from django.shortcuts import render

def feedback_form(request):
    message = ""
    if request.method == "POST":
        name = request.POST.get("name")
        email = request.POST.get("email")
        course = request.POST.get("course")
        comments = request.POST.get("comments")

        message = f"Thank you {name} for your feedback on {course}!"

    return render(request, "feedback/form.html", {"message": message})

```

v) templates/feedback/form.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Feedback Form</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container mt-5">

```

```

<h2>Feedback Form</h2>
<form method="post">
  {% csrf_token %}

  <div class="mb-3">
    <label for="name" class="form-label">Name:</label>
    <input type="text" class="form-control" id="name" name="name" required>
  </div>

  <div class="mb-3">
    <label for="email" class="form-label">Email:</label>
    <input type="email" class="form-control" id="email" name="email" required>
  </div>

  <div class="mb-3">
    <label for="course" class="form-label">Course:</label>
    <select class="form-select" id="course" name="course">
      <option value="ASP-XML">ASP-XML</option>
      <option value="DotNET">DotNET</option>
      <option value="JavaPro">JavaPro</option>
      <option value="Unix">Unix</option>
      <option value="C">C</option>
      <option value="C++">C++</option>
    </select>
  </div>

  <div class="mb-3">
    <label for="comments" class="form-label">Comments:</label>
    <textarea class="form-control" id="comments" name="comments"
rows="3"></textarea>
  </div>

  <button type="submit" class="btn btn-primary">Submit Form</button>
</form>

{% if message %}
<div class="alert alert-success mt-3">
  {{ message }}
</div>
{% endif %}
</div>
</body>
</html>

```

vi) Screenshots

Feedback Form

Name:

Peter Parker

Email:

peter@friendly.neighbor.com

Course:

JavaPro



Comments:

The worst course ever.

Submit Form

Feedback Form

Name:

Email:

Course:

ASP-XML



Comments:

Submit Form

Thank you Peter Parker for your feedback on JavaPro!