Name : Praful Kailas Patil
Roll No : 23110115
Div : B
Sub : Lab on Advance Python Programming

1. Write a Programs related to functions & modules.

Program:-

```
def square(x):
    return x * x
print("Square of 5:", square(5))
```

Output :-

```
Square of 5: 25
```

Modules Program :-

mymodule.py (module file)

```
def add(a, b):
    return a + b
def greet(name):
    return f"Hello, {name}!"
```

main.py (import module & use functions)

```
import mymodule

print("Addition:", mymodule.add(10, 15))
print(mymodule.greet("Praful !"))
```

Output :-

```
 Addition: 25
Hello, Praful !
```

2. Write a program to demonstrate the use of Dictionary& related functions

Program :-

```
student = {"name": "Praful", "age": 21, "course": "Python"}

print("Name:", student["name"])

student["grade"] = "A"

student["age"] = 22


print("Keys:", student.keys())
print("Values:", student.values())
print("Items:", student.items())
student.pop("course")

for key, value in student.items():
    print(key, ":", value)
```

Output :-

```
Name: Praful
Keys: dict_keys(['name', 'age', 'course', 'grade'])
Values: dict_values(['Praful', 22, 'Python', 'A'])
Items: dict_items([('name', 'Praful'), ('age', 22), ('course', 'Python'), ('grade', 'A')])

name : Praful
age : 22
grade :A
```

3. Write a program to demonstrate the working of classes and objects

Program :-

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def
    display(self):
        print("Name:", self.name, "| Age:", self.age)


s1 = Student("Praful",21)
s2 = Student("Om", 22)
s1.display()
s2.display()
```

Output:-

```
Name: Praful | Age: 21
Name: Om | Age: 22
```

4. Write a program to demonstrate the working of Inheritance

Program :-

```
class Person:
    def __init__(self, name):
        self.name = name
    def show(self):
        print("Name:", self.name)

class Student(Person):
    def __init__(self, name, grade):
        super().__init__(name)  # call parent constructor
        self.grade = grade

    def display(self):
        print("Name:", self.name, "| Grade:", self.grade)



p =Person("Praful")
p.show()

s = Student("Om", "A") s.display()
```

Output :-

```
Name: Praful
Name: Om | Grade: A
```

5. Write a program to demonstrate the working of Overloading Methods and Operator Method and Operator Overloading

Program :-

```
class Math:
    def add(self, a, b=0, c=0):
        return a + b + c


class Number:
    def __init__(self, value):
        self.value = value

    def __add__(self, other):  # Overloading '+' operator return
        Number(self.value + other.value)

    def __str__(self):
        return str(self.value)

m = Math()
print("Method Overloading:", m.add(5), m.add(5, 10), m.add(5, 10, 15))

n1 = Number(10)
n2 = Number(20)
print("Operator Overloading:", n1 + n2)
```

Output :-

```
Method Overloading: 5 15 30
Operator Overloading: 30
```

6. Write a program to demonstrate Exception Handling mechanism

Program :-

```
try:
    a = int(input("Enter a number: "))
    b = int(input("Enter another number: "))
    result = a / b
    print("Result:", result)
except ZeroDivisionError:
    print("Error: Division by zero is not allowed!")
except ValueError:
    print("Error: Invalid input, please enter numbers only.")
finally:
    print("Program finished.")
```

Output :-

```
Enter a number: 10
Enter another number: 0
Error: Division by zero is not allowed!
Program finished.
```

7. Create a Django project and a basic app. Display a welcome message on the homepage using a template and view.

Program :-

Bash

```
django-admin startproject myproject
```

bash

```
cd myproject
```

bash

```
python manage.py startapp home myproject
```

→ myproject → settings.py

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

home → views.py

```
from django.shortcuts import render
def home_view(request):
    context = {
        'title': 'Welcome to Django!',
        'message': 'You have successfully set up your first Django app.',
    }
    return render(request, 'home/index.html', context)
```

home → urls.py

```
from django.urls import path from . import views
urlpatterns = [
   path('', views.home_view, name='home'),
]
```

myproject → myproject → urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [     path('admin/',
admin.site.urls),      path('',
include('home.urls')),
]
```

templates│ home │index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>{{ title }}</title>
   <style>       body {          font-
family: Arial, sans-serif;
display: flex;          justify-content:
center;          align-items: center;
height: 100vh; margin: 0;
          background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
      }
      .container {
text-align: center;
background: white;
```

```
padding: 50px;
border-radius: 10px;
box-shadow: 0 10px 30px
rgba(0,0,0,0.3);
      }
h1 {
         color: #667eea;          margin-
bottom: 20px;
      }
p {
         color: #555;
font-size: 18px;
      }
   </style>
</head>
<body>
   <div class="container">
  <h1>Hello, {{ name }}!</h1>
      <p>How are you?</p>
      <p>🎉 Happy coding!</p>
   </div>
</body>
</html>
```

Output :-

8. Write a program to demonstrate Radio button, checkbox, Dialog Boxes using python Django

Program :-

Bash

```
django-admin startproject myproject
cd myproject
python manage.py startapp survey
```

myproject/settings.py

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

survey/forms.py

```
from django import forms
class
SurveyForm(forms.Form):
    name = forms.CharField(max_length=100, label="Name")

    gender = forms.ChoiceField(
        choices=[('M', 'Male'), ('F', 'Female')],
        widget=forms.RadioSelect,
        label="Gender"
    )

    hobbies = forms.MultipleChoiceField(
```

```
        choices=[
            ('reading', 'Reading'),
            ('sports', 'Sports'),
            ('music', 'Music'),
        ],
        widget=forms.CheckboxSelectMultiple,
        label="Hobbies"
    )
```

survey/views.py

```
from django.shortcuts import render, redirect
from django.contrib import messages
from .forms import SurveyForm

def survey_view(request):
    if request.method == "POST":
        form =
        SurveyForm(request.POST) if
        form.is_valid():
            name = form.cleaned_data['name'] gender
            = form.cleaned_data['gender'] hobbies =
            form.cleaned_data['hobbies']

            messages.success(
                request,
                f"Thanks {name}! Gender: {gender}, Hobbies: {', '.join(hobbies)}"
            )
            return redirect('survey')  # same page reload else:
        form = SurveyForm()
```

```
    return render(request, "survey.html", {"form": form})
```

survey/urls.py

```
 from django.urls import path
from . import views


urlpatterns = [
    path('', views.survey_view, name='survey'),
]
```

myproject/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('survey/', include('survey.urls')),
]
```

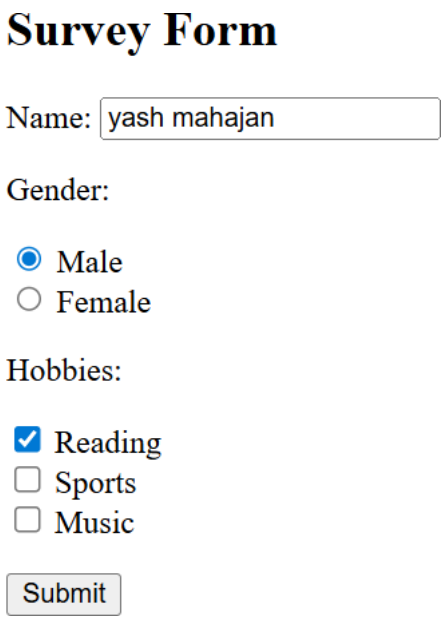
survey/templates/survey.html

```
 <!DOCTYPE html>
<html>
<head>
    <title>Survey Form</title>
</head>
<body>
    <h2>Survey Form</h2>
    <!-- Dialog Box messages -->
    {% if messages %}
        <ul>
            {% for message in messages %}
                <li style="color: green; font-weight: bold;">
                    {{ message }}
                </li>
```

```
{% endfor %}
        </ul>
    {% endif %}

    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
  </body>
  </html>
```

Output :-



9. Write a program to demonstrate to learn GUI programming using Django.

Program :-

bash

```
django-admin startproject myproject
cd myproject
python manage.py startapp guiapp
```

settings.py

```
INSTALLED_APPS = [
    ...,
    'guiapp',
]
```

guiapp/forms.py

```
from django import forms
class UserForm(forms.Form):
    name = forms.CharField(label="Enter your Name", max_length=100)
    age = forms.IntegerField(label="Enter your Age")
```

guiapp/views.py

```
from django.shortcuts import render
from .forms import UserForm
def gui_view(request):
    if request.method == "POST":
        form =
        UserForm(request.POST) if
        form.is_valid():
            name = form.cleaned_data['name']
            age = form.cleaned_data['age']
            return render(request, "result.html", {"name": name, "age": age})
    else:
        form = UserForm()
    return render(request, "gui.html", {"form": form})
```

guiapp/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
   path('', views.gui_view, name='gui'),
]
```

myproject/urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
   path('admin/', admin.site.urls),
   path('gui/', include('guiapp.urls')),
]
```

guiapp/templates/gui.html

```
html
<!DOCTYPE html>
<html>
<head>
   <title>Django GUI</title>
</head>
<body>
   <h2>Django GUI Demo</h2>
   <form method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit">Submit</button>
   </form>
</body>
```

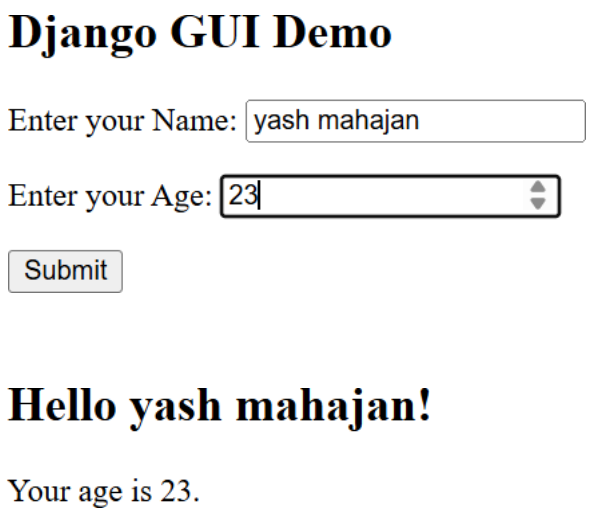
```
</html>
```

guiapp/templates/result.html

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Result</title>
</head>
<body>
    <h2>Hello {{ name }}!</h2>
    <p>Your age is {{ age }}.</p>
</body>
</html>
```

Output :-



10. Write a program to create a database application for insert, update and delete in a table using MySQL.

Program :-

Bash

```
pip install mysql-connector-python
```

MySQL

```
-- Create database
CREATE DATABASE IF NOT EXISTS testdb;

-- Use database
USE testdb;

-- Create table
CREATE TABLE IF NOT EXISTS students (
    roll_no INT PRIMARY KEY,
    name VARCHAR(100),
    age INT
);
```

mysql_app.py

```
import mysql.connector

def connect():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="Praful@2023",
        database="testdb"
```

```
    )

def insert_student(roll_no, name, age): db =
    connect()
    cursor = db.cursor()
    query = "INSERT INTO students (roll_no, name, age) VALUES (%s, %s, %s)" values =
    (roll_no, name, age)
    cursor.execute(query, values) db.commit()
    print("✅ Student inserted successfully!") db.close()

def update_student(roll_no, name, age): db =
    connect()
    cursor = db.cursor()
    query = "UPDATE students SET name=%s, age=%s WHERE roll_no=%s" values
    = (name, age, roll_no)
    cursor.execute(query, values) db.commit()
    print("✅ Student updated successfully!") db.close()

def delete_student(roll_no): db =
    connect()
```

```
cursor = db.cursor()
    query = "DELETE FROM students WHERE roll_no=%s" values =
    (roll_no,)
    cursor.execute(query, values) db.commit()
    print("✅ Student deleted successfully!") db.close()


 def view_students(): db =
    connect() cursor =
    db.cursor()
    cursor.execute("SELECT * FROM students") rows =
    cursor.fetchall()
    print("\n🎒 Current Students:")
    print("Roll No | Name | Age") for row
    in rows:
        print(row[0], "|", row[1], "|", row[2]) db.close()


 def main():
    while True:
        print("\n===== Student Database Application =====")
        print("1. Insert Student")
        print("2. Update Student")
        print("3. Delete Student")
```

```
    print("4. View Students")
    print("5. Exit")
    choice = input("Enter your choice: ")

    if choice == "1":
        roll_no = int(input("Enter Roll No: ")) name =
        input("Enter Name: ")
        age = int(input("Enter Age: "))
        insert_student(roll_no, name, age) elif
    choice == "2":
        roll_no = int(input("Enter Roll No to Update: ")) name =
        input("Enter New Name: ")
        age = int(input("Enter New Age: "))
        update_student(roll_no, name, age)
    elif choice == "3":
        roll_no = int(input("Enter Roll No to Delete: "))
        delete_student(roll_no)
         elif choice == "4":
       view_students() elif
            choice == "5":
        print("👋 Exiting program...") break
    else:
        print("❌ Invalid choice. Try again.") if
____name____== "__main___":
```

main()

Output :-

===== Student Database Application =====

1. Insert Student

2. Update Student

3. Delete Student

4. View Students

5. Exit

Enter your choice: 1 Enter

Roll No: 23110048

Enter Name: yash anil mahajan Enter

Age: 21

✅ Student inserted successfully!

===== Student Database Application =====

1. Insert Student

2. Update Student

3. Delete Student

4. View Students

5. Exit

Enter your choice: 2

Enter Roll No to Update: 23110023

Enter New Name: nayan sanjay narkhede Enter

New Age: 23

✅ Student updated successfully!

===== Student Database Application =====

1. Insert Student

2. Update Student

3. Delete Student

4. View Students

5. Exit

Enter your choice: 3

Enter Roll No to Delete: 23110023

✅ Student deleted successfully!

===== Student Database Application =====

1. Insert Student

2. Update Student

3. Delete Student

4. View Students

5. Exit

Enter your choice: 4

🛍️ Current Students:

Roll No | Name | Age 23 |

yash | 23

23110048 | yash anil mahajan | 21

===== Student Database Application =====

1. Insert Student

2. Update Student

3. Delete Student

4. View Students

5. Exit

Enter your choice: 5

👋 Exiting program..

11. Create a basic form using Django's forms.Form class with fields like name, email, and message. Display the form in a template and handle form 2 submission.

Program :-

Cantact/template/contact_form.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Contact Form</title>
</head>
<body>
  <h1>Contact Us</h1>
  {% if submitted %}
    <p>Thank you for your message!</p>
  {% else %}
    <form method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit">Send</button>
    </form>
  {% endif %}
</body>
</html>
```

Contact/forms.py

```
from django import forms

class ContactForm(forms.Form):
  name = forms.CharField(max_length=100, label='Name')
  email = forms.EmailField(label='Email')
```

```
    message = forms.CharField(widget=forms.Textarea, label='Message')
```

contact/views.py

```
from django.shortcuts import render
from .forms import ContactForm


def contact_view(request):
    submitted = False
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            submitted = True
    else:
        form = ContactForm()
    return render(request, 'contact/contact_form.html', {'form': form, 'submitted': submitted})
```

myproject/settings.py

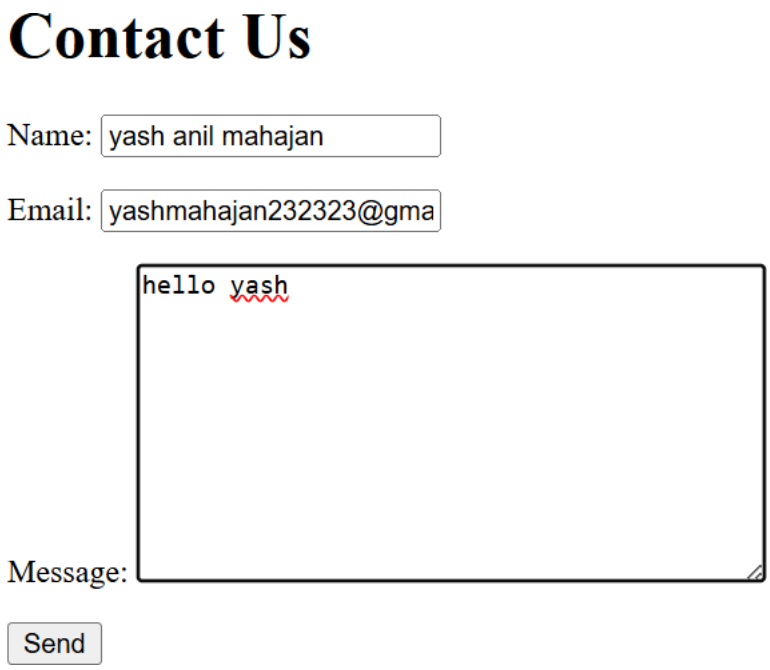
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'contact',
]
```

Myproject/urls.py

```
from django.contrib import admin
from django.urls import path
from contact.views import contact_view
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('contact/', contact_view, name='contact'),
]
```

Output :-



12.Style your form using Bootstrap for a responsive layout.

Program :-

bash

```
django-admin startproject myproject
cd myproject
python manage.py startapp guiapp
```

myproject/settings.py

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

bash

```
pip install django-widget-tweaks
```

guiapp/forms.py

```
from django import forms
class UserForm(forms.Form):
    name = forms.CharField(label="Enter your Name", max_length=100)
    age = forms.IntegerField(label="Enter your Age")
```

guiapp/views.py

```
from django.shortcuts import render
from .forms import UserForm
def gui_view(request):
```

```
    if request.method == "POST":
        form =
        UserForm(request.POST) if
        form.is_valid():
            name = form.cleaned_data['name']
            age = form.cleaned_data['age']
            return render(request, "result.html", {"name": name, "age": age})
    else:
        form = UserForm()
    return render(request, "gui.html", {"form": form})
```

guiapp/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.gui_view, name='gui'),
]
```

myproject/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('gui/', include('guiapp.urls')),
]
```

guiapp/templates/gui.html

```
{% load widget_tweaks %}
<!DOCTYPE html>
<html>
<head>
```

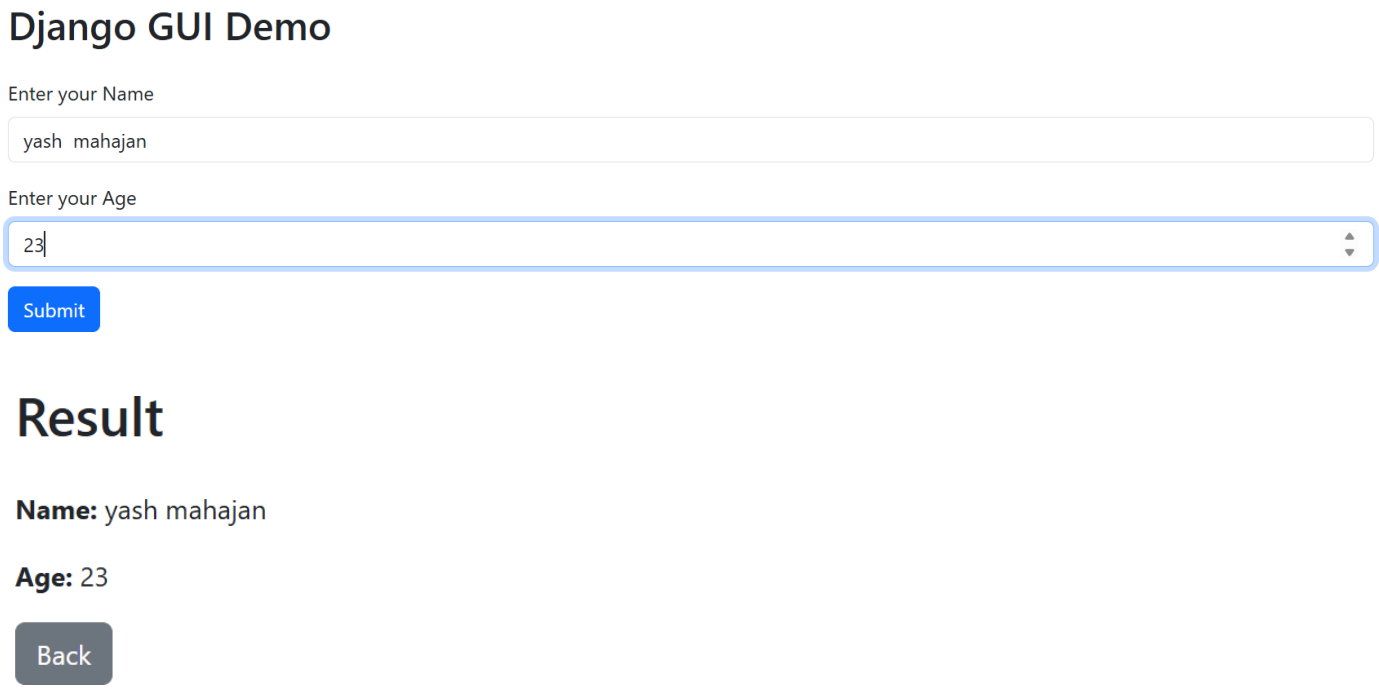
```
    <title>Django GUI</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2 class="mb-4">Django GUI Demo</h2>
    <form method="post" class="row g-3">
        {% csrf_token %}
        {% for field in form %}
            <div class="col-12">
                <label class="form-label">{{ field.label }}</label>
                {{ field|add_class:"form-control" }}
                {% if field.errors %}
                    <div class="text-danger">{{ field.errors }}</div>
                {% endif %}
            </div>
        {% endfor %}
        <div class="col-12">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </form>
</div>
</body>
</html>
```

guiapp/templates/result.html

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Result</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
  <div class="alert alert-success">
    <h4 class="alert-heading">Hello {{ name }}!</h4>
    <p>Your age is {{ age }}.</p>
  </div>
  <a href="{% url 'gui' %}" class="btn btn-secondary">Back</a>
</div>
</body>
</html>
```

Output :-



13.Add a gender radio button and terms checkbox to the form. Bash

Program :-

```
django-admin startproject myproject
cd myproject
python manage.py startapp guiapp
```

bash

```
pip install django-widget-tweaks
```

myproject/settings.py

```
INSTALLED_APPS = [
    'guiapp'
    'widget_tweaks',
]
```

guiapp/forms.py

```
from django import forms
class UserForm(forms.Form):
    name = forms.CharField(label="Enter your Name", max_length=100)
    age = forms.IntegerField(label="Enter your Age")
    # Gender radio button gender
    = forms.ChoiceField(
        choices=[('M', 'Male'), ('F', 'Female')],
        widget=forms.RadioSelect,
        label="Gender"
    )
    # Terms & Conditions checkbox
    terms = forms.BooleanField(
        label="I agree to the Terms & Conditions",
        required=True
    )
```

guiapp/views.py

```
from django.shortcuts import render
from .forms import UserForm

def gui_view(request):
    if request.method == "POST":
        form =
        UserForm(request.POST) if
        form.is_valid():
            name = form.cleaned_data['name']
            age = form.cleaned_data['age']
            gender = form.cleaned_data['gender']
            return render(request, "result.html", {"name": name, "age": age, "gender": gender}) else:
        form = UserForm()
    return render(request, "gui.html", {"form": form})
```

guiapp/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.gui_view, name='gui'),
]
```

myproject/urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('gui/', include('guiapp.urls')),
]
```

guiapp/templates/gui.html

```
{% load widget_tweaks %}
<!DOCTYPE html>
<html>
<head>
   <title>Django GUI</title>
   <!-- Bootstrap CSS -->
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
   <script>
   document.addEventListener('DOMContentLoaded', function() {
      var radios = document.querySelectorAll('input[type="radio"]');
      radios.forEach(function(radio) {
         radio.disabled = false;
      });
   });
   </script>
</head>
<body>
<div class="container mt-5">
   <h2 class="mb-4">Django GUI Demo</h2>
   <form method="post" class="row g-3">
      {% csrf_token %}
      {% for field in form %}
         <div class="col-12">
            {% if field.name == "gender" %}
               <label class="form-label">{{ field.label }}</label><br>
               {% for radio in field %}
                  <div class="form-check form-check-inline">
```

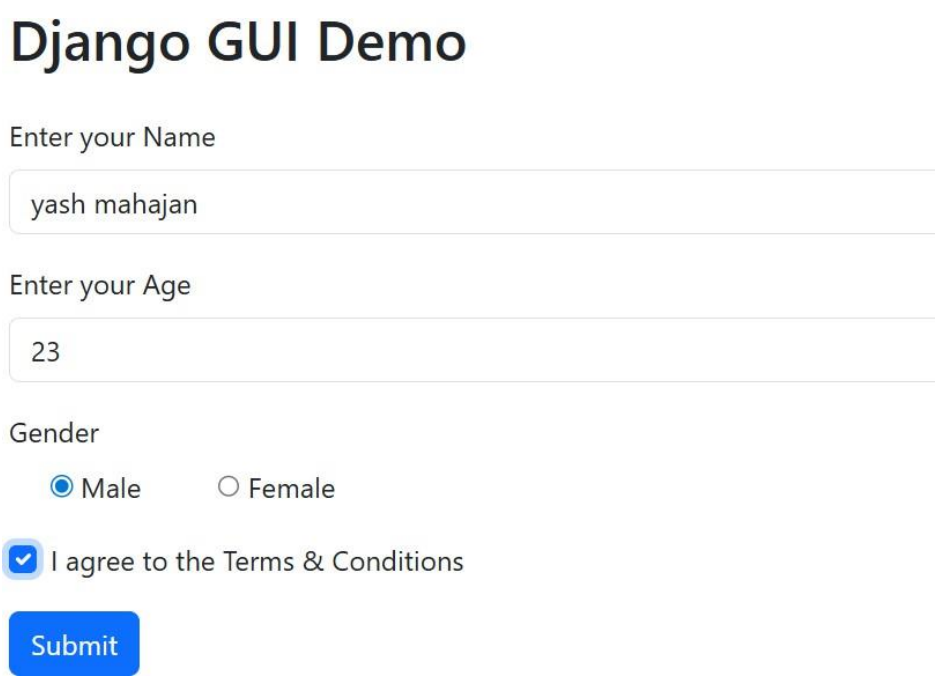
```
                {{ radio.tag }}
                <label class="form-check-label" for="{{ radio.id_for_label }}">{{ radio.choice_label }}</label>
              </div>
            {% endfor %}
          {% elif field.name == "terms" %}
            <div class="form-check">
              {{ field|add_class:"form-check-input" }}
              <label class="form-check-label">{{ field.label }}</label>
            </div>
          {% else %}
            <label class="form-label">{{ field.label }}</label>
            {{ field|add_class:"form-control" }}
          {% endif %}
          {% if field.errors %}
            <div class="text-danger">{{ field.errors }}</div>
          {% endif %}
        </div>
      {% endfor %}
      <div class="col-12">
        <button type="submit" class="btn btn-primary">Submit</button>
      </div>
    </form>
</div>
</body>
</html>
```

guiapp/templates/result.html

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Result</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <div class="alert alert-success">
        <h4 class="alert-heading">Hello {{ name }}!</h4>
        <p>Age: {{ age }}</p>
        <p>Gender: {{ gender }}</p>
        <p>Terms Accepted: Yes</p>
    </div>
    <a href="{% url 'gui' %}" class="btn btn-secondary">Back</a>
</div>
</body>
</html>
```

Output :-



Hello yash mahajan!

Age: 23

Gender: M

Terms Accepted: Yes

Back

14.Use Django's csrf token in a template to protect your form against CSRF attacks.

Program :-

bash

```
django-admin startproject myproject
cd myproject
python manage.py startapp myapp
```

project29/settings.py

```
INSTALLED_APPS = [
   'myapp',
]
MIDDLEWARE = [
   'django.middleware.csrf.CsrfViewMiddleware',
]
```

Myapp/template/contact_form.html

```
<!-- templates/contact_form.html -->
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Contact Form - CSRF Protected</title>
   <style>
      * {
         margin: 0;
         padding: 0;
         box-sizing: border-box;
      }
```

```
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 20px;
}

.container {
    background: white;
    padding: 40px;
    border-radius: 10px;
    box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
    max-width: 500px;
    width: 100%;
}

h1 {
    color: #333;
    margin-bottom: 10px;
    text-align: center;
}

.csrf-info {
    background: #e8f5e9;
```

```
    padding: 15px;
    border-radius: 5px;
    margin-bottom: 25px;
    border-left: 4px solid #4caf50;
}

.csrf-info h3 {
    color: #2e7d32;
    font-size: 14px;
    margin-bottom: 5px;
}

.csrf-info p {
    color: #1b5e20;
    font-size: 12px;
}

.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 8px;
    color: #555;
    font-weight: 500;
}
```

```
input, textarea {
    width: 100%;
    padding: 12px;
    border: 2px solid #ddd;
    border-radius: 5px;
    font-size: 14px;
    transition: border-color 0.3s;
}

input:focus, textarea:focus {
    outline: none;
    border-color: #667eea;
}

textarea {
    resize: vertical;
    min-height: 120px;
}

button {
    width: 100%;
    padding: 14px;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    font-weight: 600;
```

```
        cursor: pointer;
        transition: transform 0.2s;
    }

    button:hover {
        transform: translateY(-2px);
    }

    button:active {
        transform: translateY(0);
    }

    .messages {
        margin-bottom: 20px;
    }

    .alert {
        padding: 12px;
        border-radius: 5px;
        margin-bottom: 10px;
    }

    .alert-success {
        background: #d4edda;
        color: #155724;
        border: 1px solid #c3e6cb;
    }
</style>
```

```
</head>
<body>
  <div class="container">
    <h1>🔒 Secure Contact Form</h1>

    <div class="csrf-info">
      <h3>✓ CSRF Protection Enabled</h3>
      <p>This form is protected against Cross-Site Request Forgery attacks using Django's {% csrf_token %} tag.</p>
    </div>

    {% if messages %}
    <div class="messages">
      {% for message in messages %}
      <div class="alert alert-success">
        {{ message }}
      </div>
      {% endfor %}
    </div>
    {% endif %}

    <form method="POST" action="{% url 'contact_form' %}">
      {% csrf_token %}

      <div class="form-group">
        <label for="name">Name *</label>
        <input type="text" id="name" name="name" required>
      </div>
```

```
        <div class="form-group">
            <label for="email">Email *</label>
            <input type="email" id="email" name="email" required>
        </div>

        <div class="form-group">
            <label for="message">Message *</label>
            <textarea id="message" name="message" required></textarea>
        </div>

        <button type="submit">Send Message</button>
    </form>
  </div>
</body>
</html>
```

Myapp/views.py

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

def contact_form(request):
  if request.method == 'POST':
    name = request.POST.get('name')
    email = request.POST.get('email')
    message = request.POST.get('message')

    # Process the form data
    messages.success(request, f'Thank you {name}! Your message has been received.')
```

```
        return redirect('contact_form')

    return render(request, 'contact_form.html')
```

myapp/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.contact_form, name='contact_form'),
]
```

Project29/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('myapp.urls')),
]
```

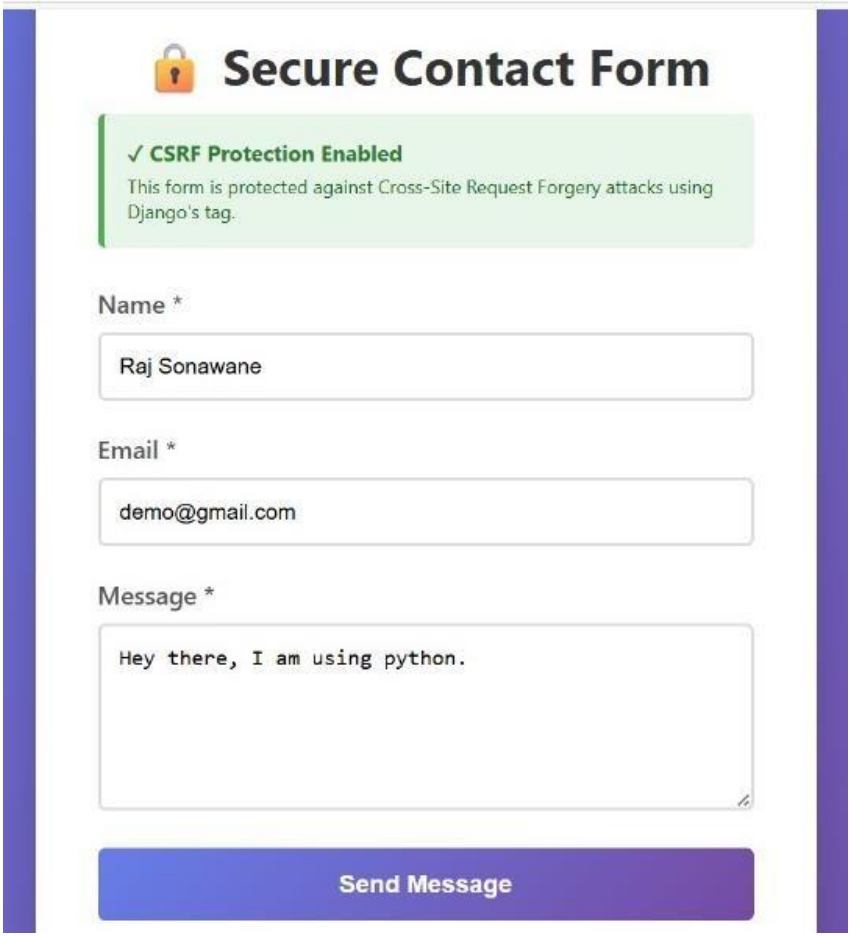
Bash

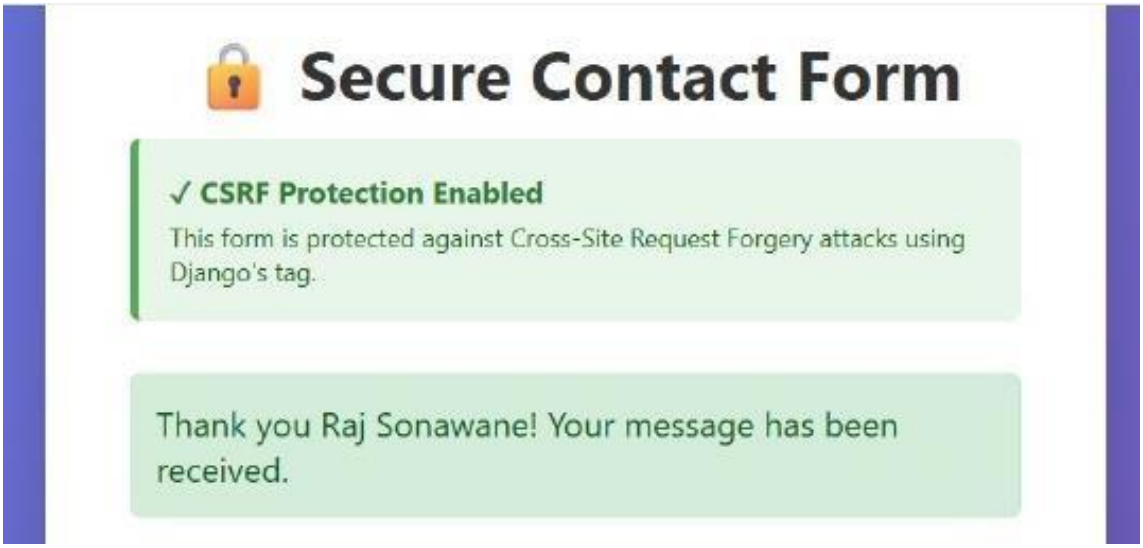
```
Python manage.py mairate
```

Bash

```
Python manage.py runserver
```

<u>Output</u> :-


Secure Contact Form
✓ CSRF Protection Enabled
This form is protected against Cross-Site Request Forgery attacks using Django's tag.
Name *
Raj Sonawane
Email *
demo@gmail.com
Message *
Hey there, I am using python.
Send Message


Secure Contact Form
✓ CSRF Protection Enabled
This form is protected against Cross-Site Request Forgery attacks using Django's tag.
Thank you Raj Sonawane! Your message has been received.

15.Show a success message after form submission using Django messages.

Program :-

Bash

```
django-admin startproject myproject cd
myproject
python manage.py startapp guiapp
```

myproject/settings.py

```
INSTALLED_APPS = [
    'guiapp',
]
```

Guiapp/templates/guiapp/my_form.html

```
<!DOCTYPE html>
<html>
<head>
    <title>My Form</title>
</head>
<body>
    <h1>Submit the Form</h1>
    {% if messages %}
        <ul>
            {% for message in messages %}
                <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>
            {% endfor %}
        </ul>
    {% endif %}
    <form method="post">
        {% csrf_token %}
```

```
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

Guiapp/urls.py

```
from django.urls import path from .
import views urlpatterns = [
    path('', views.my_form_view, name='my_form'),
]
```

Guiapp/views.py

```
from django.shortcuts import render, redirect from django
import forms
from django.contrib import messages #
Simple form example
class MyForm(forms.Form):
    name = forms.CharField(label='Your Name', max_length=100) def
my_form_view(request):
    if request.method == 'POST':
        form = MyForm(request.POST) if
        form.is_valid():
            # Process form data here (e.g., save to DB)
            messages.success(request, 'Form submitted successfully!') return
            redirect('my_form')
    else:
```

```
        form = MyForm()
    return render(request, 'guiapp/my_form.html', {'form': form})
```

myproject/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('guiapp.urls')),
]
```

Output :-

Submit the Form

Your Name: yash mahajan

Submit

Submit the Form

- Form submitted successfully!

Your Name:

Submit