**Bootstrap 5**:

Install Bootstrap 5 with this command:

```
pip install django-bootstrap-v5
```

Which will give you a result like this:

```
Collecting django-bootstrap-v5
Downloading django_bootstrap_v5-1.0.11-py3-none-any.whl (24 kB)
Requirement already satisfied: django<5.0,>=2.2 in c:\users\your
name\myworld\lib\site-packages (from django-bootstrap-v5) (4.1.4)
Collecting beautifulsoup4<5.0.0,>=4.8.0
  Downloading beautifulsoup4-4.11.1-py3-none-any.whl (128 kB)
     |████████████████████████████████| 128 kB 6.4 MB/s
Requirement already satisfied: tzdata; sys_platform == "win32" in
c:\users\your name\myworld\lib\site-packages (from django<5.0,>=2.2-
>django-bootstrap-v5) (2022.7)
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\your
name\myworld\lib\site-packages (from django<5.0,>=2.2->django-bootstrap-
v5) (3.5.2)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\your
name\myworld\lib\site-packages (from django<5.0,>=2.2->django-bootstrap-
v5) (0.4.3)
Collecting soupsieve>1.2
  Downloading soupsieve-2.3.2.post1-py3-none-any.whl (37 kB)
Installing collected packages: soupsieve, beautifulsoup4, django-
bootstrap-v5
Successfully installed beautifulsoup4-4.11.1 django-bootstrap-v5-1.0.11
soupsieve-2.3.2.post1
```

---

# Update Settings

Next step is to include the bootstrap module in the `INSTALLED_APPS` list in `settings.py`:

`my_tennis_club/my_tennis_club/settings.py`:

```
INSTALLED_APPS = [

    'django.contrib.admin',
    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'bootstrap5',

]
```

Bootstrap 5 is now ready to use in your project!

To make your login page look more beautiful using Bootstrap, you can utilize Bootstrap's predefined CSS classes and components. Here's how you can modify your template to incorporate Bootstrap styling:

**Login.html**

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
  <div class="row justify-content-center mt-5">
    <div class="col-md-6">
      <div class="card">
        <div class="card-header">
          <h2 class="text-center">Login Page</h2>
        </div>
        <div class="card-body">
          <form method="POST" novalidate>
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit" class="btn btn-primary btn-block">Login</button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock content %}
```

In this modified version:

1. I wrapped the form inside a Bootstrap card component (`<div class="card">`). This gives it a nice bordered appearance.
2. The card header (`<div class="card-header">`) contains the "Login Page" heading.
3. The form is placed inside the card body (`<div class="card-body">`), making it stand out from the background.
4. The "Login" button is styled using Bootstrap's button classes (`btn btn-primary btn-block`) to give it a consistent look and feel.

Remember to include Bootstrap CSS and JavaScript in your base.html file, as you've already done. This ensures that Bootstrap styles and components are applied correctly to your login page.

**Registraion.html:**

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <form method="POST" novalidate class="mt-5">
        {% csrf_token %}
        <h2 class="mb-4">Register Page</h2>
        {{ form.non_field_errors }}
        <div class="mb-3">
            {{ form.username.label_tag }}
            {{ form.username }}
            <small class="text-danger">{{ form.username.errors }}</small>
        </div>
        <div class="mb-3">
            {{ form.email.label_tag }}
            {{ form.email }}
            <small class="text-danger">{{ form.email.errors }}</small>
        </div>
        <div class="mb-3">
            {{ form.password1.label_tag }}
            {{ form.password1 }}
            <small class="text-danger">{{ form.password1.errors }}</small>
        </div>
        <div class="mb-3">
            {{ form.password2.label_tag }}
            {{ form.password2 }}
            <small class="text-danger">{{ form.password2.errors }}</small>
        </div>
        <button type="submit" class="btn btn-primary">Register</button>
    </form>
</div>
{% endblock %}
```

1. Each form field and its associated label are wrapped in a `<div>` with the Bootstrap class `mb-3` to add margin-bottom for spacing between form elements.
2. Error messages for each field are displayed below the input fields with red text color (Bootstrap's `text-danger` class) for better visibility.
3. The submit button is styled as a Bootstrap primary button (`btn btn-primary`).
4. The form itself is wrapped in a container (`<div class="container">`) to constrain the content width and provide padding.
5. The form is positioned with some margin from the top of the page using Bootstrap's `mt-5` class.

**Base.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% if title %} {{ title }} {% else %} Blog {% endif %}</title>
    {% load bootstrap5 %}

    <!-- Bootstrap CSS -->
    {% bootstrap_css %}

    <!-- Custom CSS -->
    <link rel="stylesheet" href="{% static 'css/style.css' %}">
    <style>
        .header-right {
            float: right;
        }
    </style>

    <!-- Bootstrap JavaScript -->
    {% bootstrap_javascript %}

    <!-- Custom JavaScript -->
    <script src="{% static 'js/app.js' %}" defer></script>
</head>
<body>
```

```
<header>
    <div class="header-right">
        {% if request.user.is_authenticated %}
            <span>Hi, {{ request.user.username|title }}</span>
            <form action="{% url 'logout' %}" method="post" class="d-inline">
                {% csrf_token %}
                <button type="submit" class="btn btn-danger btn-sm">Logout</button>
            </form>
        {% else %}
            <a href="{% url 'login' %}" class="btn btn-primary btn-sm">Login</a>
            <a href="{% url 'register' %}" class="btn btn-secondary btn-sm">Register</a>
        {% endif %}
    </div>
</header>

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

<div class="container">
    {% block content %}
    {% endblock %}
</div>
</body>
</html>
```

1. For the logout button, I replaced the anchor tag with a form that contains a logout button. This form is submitted via POST method when the logout button is clicked.
2. Both login and register buttons are styled as Bootstrap buttons using the appropriate classes (`btn btn-primary` for login and `btn btn-secondary` for register).
3. Added `btn-sm` class to make the buttons smaller
4. I added a `header-right` class with CSS to float the content to the right.
5. I placed the logout button inside a `<div>` with the `header-right` class, so it will be positioned on the top right corner of the header.

## Home.html

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <h1 class="mb-4">My Posts</h1>
    <p class="mb-4">Session Value: {{ request.session.greeting }}</p>
    {% for post in posts %}
    <div class="card mb-3">
        <div class="card-body">
            <h2 class="card-title">{{ post.title }}</h2>
            <p class="card-text"><small class="text-muted">Published on {{ post.published_at }} by {{ post.author }}</small></p>
            <p class="card-text">{{ post.content }}</p>
            <div class="btn-group" role="group" aria-label="Post Actions">
                <a href="{% url 'edit-post' post.id %}" class="btn btn-primary">Edit</a>
                <a href="{% url 'delete-post' post.id %}" class="btn btn-danger">Delete</a>
            </div>
        </div>
    </div>
    {% endfor %}
</div>
{% endblock %}
```

1. I wrapped each post in a Bootstrap card (`<div class="card">`) to provide a consistent container for each post.
2. Each post's title is styled as a card title (`<h2 class="card-title">`) and content as card text (`<p class="card-text">`).
3. The buttons for editing and deleting are wrapped in a button group (`<div class="btn-group">`) for better alignment and spacing.
4. I used Bootstrap button classes (`btn btn-primary` for editing and `btn btn-danger` for deleting) to style the buttons.

**post_confirm_delete.html:**
To add Bootstrap styling to your delete form, you can use Bootstrap's CSS classes to enhance the appearance of your HTML elements. Here's your code with Bootstrap classes added:

~~~
```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="card">
        <div class="card-body">
            <h2 class="card-title">Delete Form</h2>
            <form method="POST">
                {% csrf_token %}
                <p class="card-text">Are you sure that you want to delete the post "{{ post.title }}"?</p>
                <div class="d-flex justify-content-between align-items-center">
                    <button type="submit" class="btn btn-danger">Yes, Delete</button>
                    <a href="{% url 'posts' %}" class="btn btn-secondary">Cancel</a>
                </div>
            </form>
        </div>
    </div>
</div>
{% endblock %}

```
~~~

In this version:

1. I wrapped the content of the delete form in a Bootstrap card (`<div class="card">`) to provide a card-like appearance.
2. The title "Delete Form" is styled as a card title (`<h2 class="card-title">`).
3. The text "Are you sure that you want to delete the post..." is styled as card text (`<p class="card-text">`).
4. The buttons "Yes, Delete" and "Cancel" are styled as Bootstrap buttons (`btn btn-danger` for delete and `btn btn-secondary` for cancel).
5. The buttons are aligned horizontally using flexbox utilities (`d-flex justify-content-between align-items-center`).

This should give your delete form a cleaner and more visually appealing appearance using Bootstrap styling. Adjustments can be made as per your specific design preferences.

## Post_form.html

```
{% extends 'base.html' %}

{% block content %}
```

```
<div class="container">
    <h2>{% if id %}Edit{% else %}New{% endif %} Post</h2>
    <form method="post" class="needs-validation" novalidate>
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock %}
```

In this version:

1. I wrapped the content of the form in a Bootstrap container (`<div class="container">`) to constrain it to a specific width and provide some padding.
2. The title "Edit" or "New" Post is displayed using an `<h2>` tag.
3. The form itself is given the class `needs-validation`, which is a Bootstrap class for form validation.
4. The submit button is styled as a Bootstrap primary button using the `btn btn-primary` classes.

**Pagination.html**

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <title>Pagination in Django</title>
</head>
<body>
  <div class="container">
    <h1 class="text-center">List of Contacts</h1>
    <hr>

    <ul class="list-group">
      {% for contact in page_obj %}
        <li class="list-group-item">{{ contact }}</li>
      {% endfor %}
    </ul>

    <br><hr>
    <div class="d-flex justify-content-between align-items-center">
      <span>
        {% if page_obj.has_previous %}
          <a href="?page={{ page_obj.previous_page_number }}" class="btn btn-primary">Previous</a>
        {% endif %}
      </span>
      <span>
        Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}.
      </span>
      <span>
        {% if page_obj.has_next %}
          <a href="?page={{ page_obj.next_page_number }}" class="btn btn-primary">Next</a>
        {% endif %}
      </span>
    </div>

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

1. I've added the Bootstrap CSS file from a CDN link.
2. Adjusted the classes for the list group (`list-group`) and buttons (`btn btn-primary`) to align with Bootstrap's styling conventions.
3. Wrapped the previous and next buttons in a `<div>` with the classes `d-flex justify-content-between align-items-center` to make them align horizontally with equal spacing.

About.html

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <h1>About</h1>
    <div class="card">
        <div class="card-body">
            {{ contacts }}
        </div>
    </div>
    <p class="mt-3">Session Value: {{ request.session.greeting }}</p>
    <p>Username Value: {{ request.session.username }}</p>
</div>
```

```
{% endblock %}
```

In this version:

1. Wrapped the content of the About page in a Bootstrap container (`<div class="container">`) to provide padding and constrain the content width.
2. Wrapped the `{{ contacts }}` variable in a Bootstrap card (`<div class="card">`) to give it a bordered appearance.
3. Used Bootstrap's spacing classes (`mt-3`) to add margin-top to the "Session Value" paragraph for better spacing.