Comments

Example SocialMedia

models.py (socialmedia)

```
class Comments(models.Model):
    post = models.ForeignKey(Post ,on_delete=models.CASCADE)
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    content=models.TextField(max_length=160)
    timestamp=models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return '{}-{}'.format(self.post.title, str(self.user.username))
```

Forms.py

```
class CommentForm(forms.ModelForm):
    content = forms.CharField(label="", widget=forms.Textarea(attrs={'class': 'form-control',
'placeholder': 'Text goes here!!!', 'rows':'4', 'cols':'50'}))
    class Meta:
        model = Comments
        fields = (‘content',)
```

Admin.py

```
admin.site.register(Comments)
```

View.py

```
def post_details(request,id,slug):

    # data = Post.objects.get(id=id)

    post = get_object_or_404(Post, id=id, slug=slug)
    comments = Comments.objects.filter(post=post).order_by('-id')
    is_liked = False

    if post.likes.filter(id=request.user.id).exists():
        is_liked = True
    else:
        is_liked = False

    if request.method == 'POST':
        comment_form = CommentForm(request.POST or None)
        if comment_form.is_valid():
            content = request.POST.get('content')
            reply_id = request.POST.get('comment_id')
            comment_qs = None
            if reply_id:
                comment_qs = Comments.objects.get(id=reply_id)
            comment = Comments.objects.create(post=post, user=request.user, content=content)
            comment.save()
            return HttpResponseRedirect(post.get_absolute_url())

    else:
        comment_form = CommentForm()
    context = {
        'post': post,
        'is_liked':is_liked,
        'total_likes':post.total_likes,
        'comments': comments,
        'comment_form': comment_form,

    }

    return render(request, 'blog/post_details.html', context)
```

post_details.py

```
 <div class="container-fluid mt-2">
  <div class="form-group row">
    <form method="post" class="comment-form" action=".">
      {% csrf_token %}
      {{ comment_form.as_p }}
      {% if request.user.is_authenticated %}
        <input type="submit" value="Submit" class="btn btn-outline-success">
      {% else %}
        <input type="submit" value="Submit" class="btn btn-outline-success" disabled>
      {% endif %}
    </form>
  </div>
</div>


    <div class="main-comment-section">

        {{ comments.count }} Comment{{ comments|pluralize }}

        {% for comment in comments %}

            <blockquote class="blockquote">
                <p class="mb-0">{{ comment.content }}</p>
                <footer class="blockquote-footer">by <cite title="Source Title">{{ comment.user|
capfirst }}</cite>
                </footer>
            </blockquote>

        {% endfor %}

    </div>
```

Now For reply on comment (socialMedia)

models.py

```
# by making ForeignKey 'self' you make recursive relationships.They work similar to how One to
#Many relationships. But as the name suggests, the model references itself.

class Comments(models.Model):
    post = models.ForeignKey(Post ,on_delete=models.CASCADE)
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    content=models.TextField(max_length=160)
    reply=models.ForeignKey('self',null=True,related_name='replies',on_delete=models.CASCADE)
    timestamp=models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return '{}-{}'.format(self.post.title, str(self.user.username))
```

Forms.py

```
class CommentForm(forms.ModelForm):
    content = forms.CharField(label="", widget=forms.Textarea(attrs={'class': 'form-control',
'placeholder': 'write your comment here!!!', 'rows':'4', 'cols':'50'}))
    class Meta:
        model = Comments
        fields = (‘content',)
```

Views.py

```
def post_details(request,id,slug):

    # data = Post.objects.get(id=id)

    post = get_object_or_404(Post, id=id, slug=slug)
    comments = Comments.objects.filter(post=post,reply=None).order_by('-id')
    is_liked = False

    if post.likes.filter(id=request.user.id).exists():
        is_liked = True
    else:
        is_liked = False

    if request.method == 'POST':
        comment_form = CommentForm(request.POST or None)
        if comment_form.is_valid():
            content = request.POST.get('content')
            reply_id = request.POST.get('comment_id')
            comment_qs = None
            if reply_id:
                comment_qs = Comments.objects.get(id=reply_id)
            comment = Comments.objects.create(post=post, user=request.user, content=content,
reply=comment_qs)
            comment.save()
            return HttpResponseRedirect(post.get_absolute_url())

    else:
        comment_form = CommentForm()


    context = {
        'post': post,
        'is_liked':is_liked,
        'total_likes':post.total_likes,
        'comments': comments,
        'comment_form': comment_form,

    }

    return render(request, 'blog/post_details.html', context)
```

post_details.html

```
{% include 'blog/comment.html' %}
```

Comment,html

Comments

```
<div class="container-fluid mt-2">
      <div class="form-group row">
         <form method="post" class="comment-form" action=".">
            {% csrf_token %}
            {{ comment_form.as_p }}
            {% if request.user.is_authenticated %}
               <input type="submit" value="Submit" class="btn btn-outline-success">
            {% else %}
               <input type="submit" value="Submit" class="btn btn-outline-success" disabled>
            {% endif %}
         </form>
      </div>
   </div>


   <div class="main-comment-section">

      {{ comments.count }} Comment{{ comments|pluralize }}

      {% for comment in comments %}

         <blockquote class="blockquote">
            <p class="mb-0">{{ comment.content }}</p>
            <footer class="blockquote-footer">by <cite title="Source Title">{{ comment.user|
capfirst }}</cite>
                <button type="button" name="button" class="reply-btn btn btn-outline-dark btn-sm
ml-2">Reply</button>
            </footer>
         </blockquote>


         <div class="replied-comments container mt-2" style="display: none">
            {% for reply in comment.replies.all %}
               <blockquote class="blockquote">
                  <p class="mb-0">
                     <small>{{ reply.content }}</small>
                  </p>
                  <footer class="blockquote-footer">
                     <small>by <cite title="Source Title">{{ reply.user|capfirst }}</cite></small>
                  </footer>
               </blockquote>
            {% endfor %}
            <div class="form-group row">
               <form method="post" class="reply-form" action=".">
                  {% csrf_token %}
                  <input type="hidden" name="comment_id" value="{{ comment.id }}">
                  {{ comment_form.as_p }}
                  {% if request.user.is_authenticated %}
                     <input type="submit" value="Submit" class="btn btn-outline-success">
                  {% else %}
                     <input type="submit" value="Submit" class="btn btn-outline-success"
disabled>
                  {% endif %}
               </form>
            </div>
         </div>

      {% endfor %}
```

```
    </div>
```

Js (base.html)

```
    $('.reply-btn').click(function () {
        $(this).parent().parent().next('.replied-comments').fadeToggle()
    });
```

Comment using Ajax SocialMedia

models.py (socialmedia)

```
class Comments(models.Model):
    post = models.ForeignKey(Post ,on_delete=models.CASCADE)
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    content=models.TextField(max_length=160)
    timestamp=models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return '{}-{}'.format(self.post.title, str(self.user.username))
```

Forms.py

```
class CommentForm(forms.ModelForm):
    content = forms.CharField(label="", widget=forms.Textarea(attrs={'class': 'form-control',
'placeholder': 'Text goes here!!!', 'rows':'4', 'cols':'50'}))
    class Meta:
        model = Comments
        fields = (‘content',)
```

Views.py

```
def post_details(request,id,slug):

    # data = Post.objects.get(id=id)

    post = get_object_or_404(Post, id=id, slug=slug)
    comments = Comments.objects.filter(post=post,reply=None).order_by('-id')
    is_liked = False

    if post.likes.filter(id=request.user.id).exists():
        is_liked = True
    else:
        is_liked = False

    if request.method == 'POST':
        comment_form = CommentForm(request.POST or None)
        if comment_form.is_valid():
            content = request.POST.get('content')
            reply_id = request.POST.get('comment_id')
            comment_qs = None
            if reply_id:
                comment_qs = Comments.objects.get(id=reply_id)
            comment = Comments.objects.create(post=post, user=request.user, content=content,
reply=comment_qs)
            comment.save()
            # return HttpResponseRedirect(post.get_absolute_url())
    else:
        comment_form = CommentForm()


    context = {
        'post': post,
        'is_liked':is_liked,
        'total_likes':post.total_likes,
        'comments': comments,
        'comment_form': comment_form,

    }

    if request.is_ajax():
        html = render_to_string('blog/comment.html', context, request=request)
        return JsonResponse({'form': html})

    return render(request, 'blog/post_details.html', context)
```

post_details.html

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

{% block content %}

    {% include 'blog/alerts.html' %}

    <h3>
        <i>{{ post.title }}</i>
    </h3>
    <p>
        Author : {{ post.author }}
    </p>

    <small style="margin-top: -10px;float: left">{{ post.created }}</small>
    <div style="height: 10px;"></div>
    <hr>
    <p>{{ post.body }}</p>


    <div class="row gallery">
        {% for p in post.images_set.all %}
            {% if p.image %}
                <div class="col-md-3">
                    <img src="{{ p.image.url }}" alt="Image Not Found" class="img-thumbnail">
                </div>
            {% endif %}
        {% endfor %}
    </div>


    <div id="like_section">
        {% include 'blog/like_section.html' %}
    </div>

    {% if post.author == request.user %}

        <div class="section-1" style="float:right;">

            <a href="{% url 'post_edit' id=post.id %}">
            <button type="button" class="btn btn-primary">Edit</button>

          </a>

          <a href="{% url 'post_delete' id=post.id %}">

            <button type="button" id="delete" class="btn btn-danger">Delete</button>

          </a>

        </div>

    {% endif %}
    <br>
    <h2>add your comment</h2>
    <br>

    <div class="container-fluid mt-2">
        <div class="form-group row">
            <form method="post" class="comment-form" action=".">
                {% csrf_token %}
                {{ comment_form.as_p }}
                {% if request.user.is_authenticated %}
                    <input type="submit" value="Submit" class="btn btn-outline-success">
                {% else %}
                    <input type="submit" value="Submit" class="btn btn-outline-success" disabled>
                {% endif %}
            </form>
        </div>
    </div>

    {% include 'blog/comment.html' %}




{% endblock %}
```

comment.html

```
    <div class="main-comment-section">

        {{ comments.count }} Comment{{ comments|pluralize }}

        {% for comment in comments %}

            <blockquote class="blockquote">
                <p class="mb-0">{{ comment.content }}</p>
                <footer class="blockquote-footer">by <cite title="Source Title">{{ comment.user|
capfirst }}</cite>
                    <button type="button" name="button" class="reply-btn btn btn-outline-dark btn-sm
ml-2">Reply</button>
                </footer>
            </blockquote>


            <div class="replied-comments container mt-2" style="display: none">
                {% for reply in comment.replies.all %}
                <blockquote class="blockquote">
                    <p class="mb-0">
                        <small>{{ reply.content }}</small>
                    </p>
                    <footer class="blockquote-footer">
                        <small>by <cite title="Source Title">{{ reply.user|capfirst }}</cite></small>
                    </footer>
                </blockquote>
            {% endfor %}
            <div class="form-group row">
                <form method="post" class="reply-form" action=".">
                    {% csrf_token %}
                    <input type="hidden" name="comment_id" value="{{ comment.id }}">
                    {{ comment_form.as_p }}
                    {% if request.user.is_authenticated %}
                        <input type="submit" value="Submit" class="btn btn-outline-success">
                    {% else %}
                        <input type="submit" value="Submit" class="btn btn-outline-success"
disabled>
                    {% endif %}
                </form>
            </div>
        </div>

    {% endfor %}

  </div>
```

Js>base.html

```
  $(document).on('submit', '.comment-form', function(event){
        event.preventDefault();
        console.log($(this).serialize());
        $.ajax({
          type: 'POST',
          url: $(this).attr('action'),
          data: $(this).serialize(),
          dataType: 'json',
          success: function(response) {
            $('.main-comment-section').html(response['form']);
            $('textarea').val('');
            $('.reply-btn').click(function() {
              $(this).parent().parent().next('.replied-comments').fadeToggle();
              $('textarea').val('');
            });
          },
          error: function(rs, e) {
            console.log(rs.responseText);
          },
        });
      });

    $(document).on('submit', '.reply-form', function(event){
        event.preventDefault();
        console.log($(this).serialize());
        $.ajax({
          type: 'POST',
          url: $(this).attr('action'),
          data: $(this).serialize(),
          dataType: 'json',
          success: function(response) {
            $('.main-comment-section').html(response['form']);
            $('textarea').val('');
            $('.reply-btn').click(function() {
              $(this).parent().parent().next('.replied-comments').fadeToggle();
              $('textarea').val('');
            });
          },
          error: function(rs, e) {
            console.log(rs.responseText);
          },
        });
      });
```