**Q.1 Create login page using Django**

Create project mysite>cd mysite>create app main>create urls.py in main>

**Form.py**

from django import forms

from django.contrib.auth.forms import UserCreationForm

from django.contrib.auth.models import User

# Create your forms here.

class NewUserForm(UserCreationForm):

    email = forms.EmailField(required=True)

    class Meta:

        model = User

        fields = ("username", "email", "password1", "password2")

    def save(self, commit=True):

        user = super(NewUserForm, self).save(commit=False)

        user.email = self.cleaned\_data['email']

        if commit:

            user.save()

        return user

--------------------------------------------------------------------------

**Q.2 create blog using Django**

**views.py**

from django.shortcuts import  render, redirect

from .forms import NewUserForm

from django.contrib.auth import login, authenticate, logout #add this

from django.contrib import messages

from django.contrib.auth.forms import AuthenticationForm

def register\_request(request):

    if request.method == "POST":

        form = NewUserForm(request.POST)

        if form.is\_valid():

            user = form.save()

            login(request, user)

            messages.success(request, "Registration successful." )

            return redirect("main:homepage")

        messages.error(request, "Unsuccessful registration. Invalid information.")

    form = NewUserForm()

    return render (request=request, template\_name="main/register.html", context={"register\_form":form})

from django.shortcuts import  render, redirect

from .forms import NewUserForm

from django.contrib.auth import login, authenticate #add this

from django.contrib import messages

from django.contrib.auth.forms import AuthenticationForm #add this

def login\_request(request):

    if request.method == "POST":

        form = AuthenticationForm(request, data=request.POST)

        if form.is\_valid():

            username = form.cleaned\_data.get('username')

            password = form.cleaned\_data.get('password')

            user = authenticate(username=username, password=password)

            if user is not None:

                login(request, user)

                messages.info(request, f"You are now logged in as {username}.")

                return redirect("main:homepage")

            else:

                messages.error(request,"Invalid username or password.")

        else:

            messages.error(request,"Invalid username or password.")

    form = AuthenticationForm()

    return render(request=request, template\_name="main/login.html", context={"login\_form":form})

    def logout\_request(request):

    logout(request)

    messages.info(request, "You have successfully logged out.")

    return redirect("main:homepage")

**urls.py**

from django.urls import path

from . import views

app\_name = "main"

urlpatterns = [

    path("", views.homepage, name="homepage"),

    ...

    path("register", views.register\_request, name="register"),

    path("login", views.login\_request, name="login"),

    path("logout", views.logout\_request, name= "logout"),

]

**Register.html**

{% extends "main/header.html" %}

{% block content %}

{% load crispy\_forms\_tags %}

<!--Register-->

<div class="container py-5">

    <h1>Register</h1>

    <form method="POST">

        {% csrf\_token %}

        {{ register\_form|crispy }}

        <button class="btn btn-primary" type="submit">Register</button>

    </form>

    <p class="text-center">If you already have an account, <a href="/login">login</a> instead.</p>

</div>

{% endblock %}

**Login.html**

{% extends "main/header.html" %}

{% block content %}

{% load crispy\_forms\_tags %}

<!--Login-->

<div class="container py-5">

  <h1>Login</h1>

  <form method="POST">

    {% csrf\_token %}

    {{ login\_form|crispy }}

    <button class="btn btn-primary" type="submit">Login</button>

  </form>

  <p class="text-center">Don't have an account? <a href="/register">Create an account</a>.</p>

</div>

{% endblock %}

**Logout.html**

<!--Navbar-->

<nav class="navbar navbar-expand-lg navbar-light bg-light">

    <a class="navbar-brand" href="#">Navbar</a>

    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText" aria-controls="navbarText" aria-expanded="False" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarText">

      <ul class="navbar-nav mr-auto">

        {% if user.is\_authenticated %}

        <li class="nav-item">

          <a class="nav-link" href="/logout">Logout</a>

        </li>

        <li class="nav-item">

          <a class="nav-link" href="#">Welcome, {{user.username}}</a>

        </li>

        {% else %}

        <li class="nav-item">

          <a class="nav-link" href="/login">Login</a>

        </li>

        {% endif %}

      </ul>

    </div>

  </nav>

**Q.2 craete blog using Django**

**Create project mysite>cd mysite>createapp blog>inside it create urls.py file and “templates” fo**

**lder>inside template folder create base.html,index.html,post\_details.html**

**virtualenv django**

**cd Django**

**Scrpts\activate.bat**

**Pip install django’**

**Python manage.py makemigrations**

**Python manage.py migrate**

**Python manage.py createsuperuser**

**Settings.py**

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'blog'

**blog>models.py**

from django.db import models

from django.contrib.auth.models import User

STATUS = (

    (0,"Draft"),

    (1,"Publish")

)

class Post(models.Model):

    title = models.CharField(max\_length=200, unique=True)

    slug = models.SlugField(max\_length=200, unique=True)

    author = models.ForeignKey(User, on\_delete= models.CASCADE,related\_name='blog\_posts')

    updated\_on = models.DateTimeField(auto\_now= True)

    content = models.TextField()

    created\_on = models.DateTimeField(auto\_now\_add=True)

    status = models.IntegerField(choices=STATUS, default=0)

    class Meta:

        ordering = ['-created\_on']

    def \_\_str\_\_(self):

        return self.title

**Blog/admin.py**

from django.contrib import admin

from .models import Post

class PostAdmin(admin.ModelAdmin):

    list\_display = ('title', 'slug', 'status','created\_on')

    list\_filter = ("status",)

    search\_fields = ['title', 'content']

    prepopulated\_fields = {'slug': ('title',)}

admin.site.register(Post, PostAdmin)

**blog/views.py**

from django.views import generic

from .models import Post

class PostList(generic.ListView):

    queryset = Post.objects.filter(status=1).order\_by('-created\_on')

    template\_name = 'index.html'

class PostDetail(generic.DetailView):

    model = Post

    template\_name = 'post\_detail.html'

**blog/urls.py**

from . import views

from django.urls import path

urlpatterns = [

    path('', views.PostList.as\_view(), name='home'),

    path('<slug:slug>/', views.PostDetail.as\_view(), name='post\_detail'),

]

**Mysite/urls.py**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('blog.urls')),

]

**Template/base.html**

<!DOCTYPE html>

<html>

    <head>

        <title>Django Central</title>

        <link href="https://fonts.googleapis.com/css?family=Roboto:400,700" rel="stylesheet">

        <meta name="google" content="notranslate" />

        <meta name="viewport" content="width=device-width, initial-scale=1" />

        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"

            crossorigin="anonymous" />

    </head>

    <body>

        <style>

            body {

            font-family: "Roboto", sans-serif;

            font-size: 17px;

            background-color: #fdfdfd;

        }

        .shadow {

            box-shadow: 0 4px 2px -2px rgba(0, 0, 0, 0.1);

        }

        .btn-danger {

            color: #fff;

            background-color: #f00000;

            border-color: #dc281e;

        }

        .masthead {

            background: #3398E1;

            height: auto;

            padding-bottom: 15px;

            box-shadow: 0 16px 48px #E3E7EB;

            padding-top: 10px;

        }

    </style>

        <!-- Navigation -->

        <nav class="navbar navbar-expand-lg navbar-light bg-light shadow" id="mainNav">

            <div class="container-fluid">

                <a class="navbar-brand" href="{% url 'home' %}">Django central</a>

                <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive"

                    aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                </button>

                <div class="collapse navbar-collapse" id="navbarResponsive">

                    <ul class="navbar-nav ml-auto">

                        <li class="nav-item text-black">

                            <a class="nav-link text-black font-weight-bold" href="#">About</a>

                        </li>

                        <li class="nav-item text-black">

                            <a class="nav-link text-black font-weight-bold" href="#">Policy</a>

                        </li>

                        <li class="nav-item text-black">

                            <a class="nav-link text-black font-weight-bold" href="#">Contact</a>

                        </li>

                    </ul>

                </div>

            </div>

        </nav>

        {% block content %}

        <!-- Content Goes here -->

        {% endblock content %}

        <!-- Footer -->

        <footer class="py-3 bg-grey">

            <p class="m-0 text-dark text-center ">Copyright &copy; Django Central</p>

        </footer>

    </body>

</html>

{% block sidebar %}

<style>

        .card{

            box-shadow: 0 16px 48px #E3E7EB;

        }

</style>

<!-- Sidebar Widgets Column -->

<div class="col-md-4 float-right ">

<div class="card my-4">

        <h5 class="card-header">About Us</h5>

    <div class="card-body">

        <p class="card-text"> This awesome blog is made on the top of our Favourite full stack Framework 'Django', follow up the tutorial to learn how we made it..!</p>

        <a href="https://djangocentral.com/building-a-blog-application-with-django"

           class="btn btn-danger">Know more!</a>

    </div>

</div>

</div>

{% endblock sidebar %}

**Index.html**

{% extends "base.html" %}

{% block content %}

<style>

    body {

        font-family: "Roboto", sans-serif;

        font-size: 18px;

        background-color: #fdfdfd;

    }

    .head\_text {

        color: white;

    }

    .card {

        box-shadow: 0 16px 48px #E3E7EB;

    }

</style>

<header class="masthead">

    <div class="overlay"></div>

    <div class="container">

        <div class="row">

            <div class=" col-md-8 col-md-10 mx-auto">

                <div class="site-heading">

                    <h3 class=" site-heading my-4 mt-3 text-white"> Welcome to my awesome Blog </h3>

                    <p class="text-light">We Love Django As much as you do..! &nbsp

                    </p>

                </div>

            </div>

        </div>

    </div>

</header>

<div class="container">

    <div class="row">

        <!-- Blog Entries Column -->

        <div class="col-md-8 mt-3 left">

            {% for post in post\_list %}

            <div class="card mb-4">

                <div class="card-body">

                    <h2 class="card-title">{{ post.title }}</h2>

                    <p class="card-text text-muted h6">{{ post.author }} | {{ post.created\_on}} </p>

                    <p class="card-text">{{post.content|slice:":200" }}</p>

                    <a href="{% url 'post\_detail' post.slug  %}" class="btn btn-primary">Read More &rarr;</a>

                </div>

            </div>

            {% endfor %}

        </div>

        {% block sidebar %} {% include 'sidebar.html' %} {% endblock sidebar %}

    </div>

</div>

{%endblock%}

**Post\_details.html**

{% extends 'base.html' %} {% block content %}

<div class="container">

  <div class="row">

    <div class="col-md-8 card mb-4  mt-3 left  top">

      <div class="card-body">

        <h1>{% block title %} {{ object.title }} {% endblock title %}</h1>

        <p class=" text-muted">{{ post.author }} | {{ post.created\_on }}</p>

        <p class="card-text ">{{ object.content | safe }}</p>

      </div>

    </div>

    {% block sidebar %} {% include 'sidebar.html' %} {% endblock sidebar %}

  </div>

</div>

{% endblock content %}

**Q.3 create clone “hackerSite”**

**Createporject HackerNews>createapp hnapp>**

**Create templates folder in HackerNews**

**Setting.py**

DIRS': [os.path.join(BASE\_DIR,'HackerNews/templates')],

**>template/Base.html**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<style>

body {

  margin: 0;

  font-family: Arial, Helvetica, sans-serif;

}

.topnav {

  overflow: hidden;

  background-color: #333;

}

.topnav a {

  float: left;

  color: #f2f2f2;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

  font-size: 17px;

}

.topnav a:hover {

  background-color: #ddd;

  color: black;

}

.topnav a.active {

  background-color: #FF0000;

  color: white;

}

.topnav-right {

  float: right;

}

p.ex1 {

  padding-left: 40px;

}

</style>

</head>

<body>

    {% block content %}

    {% endblock %}

</body>

</html>

**Hackerclone/urls.py**

**from** django.contrib **import** admin

**from** django.urls **import** path,include

urlpatterns **=** [

    path('admin/', admin.site.urls),

    path('',include('hnapp.urls')),

]

**Hnapp>Models.py**

**from** django.db **import** models

**from** django.contrib.auth.models **import** User

# Create your models here.

**class** Post(models.Model):

    title **=** models.CharField("HeadLine", max\_length**=**256, unique**=**True)

    creator **=** models.ForeignKey(User, on\_delete**=** models.SET\_NULL, null**=**True)

    created\_on **=** models.DateTimeField(auto\_now\_add**=**True)

    url **=** models.URLField("URL", max\_length**=**256,blank**=**True)

    description **=** models.TextField("Description", blank**=**True)

    votes **=** models.IntegerField(null**=**True)

    comments **=** models.IntegerField(null**=**True)

**def** \_\_unicode\_\_(self):

**return** self.title

**def** count\_votes(self):

        self.votes **=** Vote.objects.filter(post **=** self).count()

**def** count\_comments(self):

        self.comments **=** Comment.objects.filter(post **=** self).count()

**class** Vote(models.Model):

    voter **=** models.ForeignKey(User, on\_delete**=**models.CASCADE)

    post **=** models.ForeignKey(Post, on\_delete**=**models.CASCADE)

**def** \_\_unicode\_\_(self):

**return** f"{self.user.username} upvoted {self.link.title}"

**class** Comment(models.Model):

    creator **=** models.ForeignKey(User, on\_delete**=**models.SET\_NULL, null**=**True)

    post **=** models.ForeignKey(Post, on\_delete**=**models.CASCADE)

    content **=** models.TextField()

    identifier **=** models.IntegerField()

    parent **=** models.ForeignKey('self', on\_delete**=**models.SET\_NULL, null**=**True)

**def** \_\_unicode\_\_(self):

**return** f"Comment by {self.user.username}"

**hnapp>views.py**

from django.shortcuts import render,redirect,get\_object\_or\_404

from django.views.generic import ListView

from .models import Post,Vote,Comment

from .forms import CommentForm,PostForm

from django.contrib.auth.models import User

from django.contrib.auth import authenticate,login,logout

from django.contrib.auth.forms import AuthenticationForm,UserCreationForm

from datetime import datetime,timedelta

from django.utils import timezone

from django.contrib.auth import authenticate,login,logout

from django.contrib.auth.forms import AuthenticationForm,UserCreationForm

# Create your views here.

def PostListView(request):

    posts = Post.objects.all()

    for post in posts:

        post.count\_votes()

        post.count\_comments()

    context = {

        'posts': posts,

    }

    return render(request,'hnapp/postlist.html',context)

def NewPostListView(request):

    posts = Post.objects.all().order\_by('-created\_on')

    for post in posts:

        post.count\_votes()

        post.count\_comments()

    context = {

        'posts': posts,

    }

    return render(request,'hnapp/postlist.html', context)

def PastPostListView(request):

    time = str((datetime.now(tz=timezone.utc) - timedelta(minutes=30)))

    posts = Post.objects.filter(created\_on\_\_lte = time)

    for post in posts:

        post.count\_votes()

        post.count\_comments()

    context={

        'posts': posts,

    }

    return render(request,'hnapp/postlist.html',context)

def UpVoteView(request,id):

    if request.user.is\_authenticated:

        post = Post.objects.get(id=id)

        votes = Vote.objects.filter(post = post)

        v = votes.filter(voter = request.user)

        if len(v) == 0:

            upvote = Vote(voter=request.user,post=post)

            upvote.save()

            return redirect('/')

    return redirect('/signin')

def DownVoteView(request,id):

    if request.user.is\_authenticated:

        post = Post.objects.get(id=id)

        votes = Vote.objects.filter(post = post)

        v = votes.filter(voter = request.user)

        if len(v) != 0:

            v.delete()

            return redirect('/')

    return redirect('/signin')

def UserInfoView(request,username):

    user = User.objects.get(username=username)

    context = {'user':user,}

    return render(request,'hnapp/userinfo.html',context)

def UserSubmissions(request,username):

    user = User.objects.get(username=username)

    posts = Post.objects.filter(creator = user)

    print(len(posts))

    for post in posts:

        post.count\_votes()

        post.count\_comments()

    return render(request,'hnapp/user\_posts.html',{'posts': posts})

def EditListView(request,id):

    post = get\_object\_or\_404(Post,id=id)

    if request.method =='POST':

        form = PostForm(request.POST, instance=post)

        if form.is\_valid():

            form.save()

            return redirect('/')

    form = PostForm(instance =post)

    return render(request,'hnapp/submit.html',{'form':form})

def CommentListView(request,id):

    form = CommentForm()

    post = Post.objects.get(id =id)

    post.count\_votes()

    post.count\_comments()

    comments = []

    def func(i,parent):

        children = Comment.objects.filter(post =post).filter(identifier =i).filter(parent=parent)

        for child in children:

            gchildren = Comment.objects.filter(post =post).filter(identifier = i+1).filter(parent=child)

            if len(gchildren)==0:

                comments.append(child)

            else:

                func(i+1,child)

                comments.append(child)

    func(0,None)

    if request.method == "POST":

        if request.user.is\_authenticated:

            form = CommentForm(request.POST)

            if form.is\_valid():

                content = form.cleaned\_data['content']

                comment = Comment(creator = request.user,post = post,content = content,identifier =0)

                comment.save()

                return redirect(f'/post/{id}')

        return redirect('/signin')

    context ={

        'form': form,

        'post': post,

        'comments': list(reversed(comments)),

    }

    return render(request,'hnapp/post.html', context)

def CommentReplyView(request,id1,id2):

    form = CommentForm()

    comment = Comment.objects.get(id = id2)

    post = Post.objects.get(id=id1)

    if request.method == "POST":

        if request.user.is\_authenticated:

            form = CommentForm(request.POST)

            if form.is\_valid():

                reply\_comment\_content = form.cleaned\_data['content']

                identifier = int(comment.identifier + 1)

                reply\_comment = Comment(creator = request.user, post = post, content = reply\_comment\_content, parent=comment, identifier= identifier)

                reply\_comment.save()

                return redirect(f'/post/{id1}')

        return redirect('/signin')

    context ={

        'form': form,

        'post': post,

        'comment': comment,

    }

    return render(request,'hnapp/reply\_post.html', context)

def SubmitPostView(request):

    if request.user.is\_authenticated:

        form = PostForm()

        if request.method == "POST":

            form = PostForm(request.POST)

            if form.is\_valid():

                title = form.cleaned\_data['title']

                url = form.cleaned\_data['url']

                description = form.cleaned\_data['description']

                creator = request.user

                created\_on = datetime.now()

                post = Post(title=title, url=url, description=description, creator = creator, created\_on=created\_on)

                post.save()

                return redirect('/')

        return render(request,'hnapp/submit.html',{'form':form})

    return redirect('/signin')

def signup(request):

    if request.user.is\_authenticated:

        return redirect('/')

    if request.method == 'POST':

        form = UserCreationForm(request.POST)

        if form.is\_valid():

            form.save()

            username = form.cleaned\_data['username']

            password = form.cleaned\_data['password1']

            user = authenticate(username = username,password = password)

            login(request, user)

            return redirect('/')

        else:

            return render(request,'hnapp/auth\_signup.html',{'form':form})

    else:

        form = UserCreationForm()

        return render(request,'hnapp/auth\_signup.html',{'form':form})

def signin(request):

    if request.user.is\_authenticated:

        return redirect('/')

    if request.method == 'POST':

        username = request.POST['username']

        password = request.POST['password']

        user = authenticate(request, username =username, password = password)

        if user is not None:

            login(request,user)

            return redirect('/')

        else:

            form = AuthenticationForm()

            return render(request,'hnapp/auth\_signin.html',{'form':form})

    else:

        form = AuthenticationForm()

        return render(request, 'hnapp/auth\_signin.html', {'form':form})

def signout(request):

    logout(request)

    return redirect('/')

**urls.py**

**from** django.contrib **import** admin

**from** django.urls **import** path

**from** .views **import** **\***

urlpatterns **=** [

    path('',PostListView, name='home'),

    path('new',NewPostListView, name**=**'new\_home'),

    path('past',PastPostListView, name**=**'past\_home'),

    path('user/<username>', UserInfoView, name**=**'user\_info'),

    path('posts/<username>',UserSubmissions, name**=**'user\_posts'),

    path('post/<int:id>',CommentListView, name**=**'post'),

    path('submit',SubmitPostView, name**=**'submit'),

    path('signin',signin, name**=**'signin'),

    path('signup',signup, name**=**'signup'),

    path('signout',signout, name**=**'signout'),

    path('vote/<int:id>',UpVoteView,name**=**'vote'),

    path('downvote/<int:id>',DownVoteView,name**=**'dvote'),

    path('edit/<int:id>',EditListView, name**=**'edit'),

    path('post/<int:id1>/comment/<int:id2>',CommentReplyView,name**=**'reply'),

]

**Hnapp>Forms.py**

**from** django **import** forms

**from** .models **import** Comment,Post

**class** CommentForm(forms.ModelForm):

**class** Meta:

        model **=** Comment

        fields **=** ('content',)

**class** PostForm(forms.ModelForm):

**class** Meta:

        model **=** Post

        fields **=** ('title','url','description')

**hnapp>admin.py**

**from** django.contrib **import** admin

**from** .models **import** **\***

# Register your models here.

admin.site.register(Post)

admin.site.register(Vote)

admin.site.register(Comment)

#admin.site.register(UserInfo)

**Templates>Postlist.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

  {% if request.user.is\_authenticated %}

    <div class="topnav-right">

      <a href="{% url 'signout' %}">Sign Out </a>

    </div>

  {% else %}

    <div class="topnav-right">

      <a href="{% url 'signin' %}">Sign In </a>

    </div>

  {% endif %}

</div>

<div class="w3-panel w3-light-grey w3-leftbar w3-border-grey">

  <ol>

{% for post in posts %}

  <li><p><a href = "{{post.url}}"><strong>{{post.title}}</strong></a> - <a href = "{% url 'vote' post.id %}">Upvote</a> - <a href = "{% url 'dvote' post.id %}">Downvote</a></p>

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

    <p>{{post.votes}} votes | Created {{post.created\_on}}| <a href = "{% url 'user\_info' post.creator.username %}">{{post.creator.username}}</a> | <a href="{% url 'post' post.id %}"> {{post.comments}} Comments</a> | <a href="{% url 'edit' post.id %}"> Edit</a></p></li>

  {%else %}

    <p>{{post.votes}} votes | Created {{post.created\_on}}| <a href = "{% url 'user\_info' post.creator.username %}">{{post.creator.username}}</a> | <a href="{% url 'post' post.id %}"> {{post.comments}} Comments</a></p></li>

  {%endif%}

{% endfor %}

</ol>

</div>

{% endblock %}

**User\_info.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

  {% if request.user.is\_authenticated %}

    <div class="topnav-right">

      <a href="{% url 'signout' %}">Sign Out </a>

    </div>

  {% else %}

    <div class="topnav-right">

      <a href="{% url 'signin' %}">Sign In </a>

    </div>

  {% endif %}

</div>

<div class="w3-panel w3-light-grey w3-leftbar w3-border-grey">

<p><strong>User: </strong>{{user.username}}</p>

<p><strong>Created: </strong>{{user.date\_joined}}</p>

</div>

<a href="{% url 'user\_posts' user.username %}">Submissions</a>

{% endblock %}

**User\_post.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

  {% if request.user.is\_authenticated %}

    <div class="topnav-right">

      <a href="{% url 'signout' %}">Sign Out </a>

    </div>

  {% else %}

    <div class="topnav-right">

      <a href="{% url 'signin' %}">Sign In </a>

    </div>

  {% endif %}

</div>

<ol>

{%for post in posts%}

  <div class="w3-panel w3-light-grey w3-leftbar w3-border-grey">

  <li><p><a href = "{{post.url}}">{{post.title}}</a></p>

  <p>{{post.votes}} | Created {{post.created\_on}}| <a href = "{% url 'user\_info' post.creator.username %}">{{post.creator.username}}</a> | <a href="{% url 'post' post.id %}"> {{post.comments}} Comments</a></p></li>

</div>

{% endfor %}

</ol>

{% endblock %}

**Submit.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

</div>

<form method ='post'>

    {% csrf\_token %}

    {{form.as\_p}}

    <input type="submit" value = "Submit">

</form>

<br>

<h3>Already Have an Account??</h3>

<a href = "{% url 'signin' %}">Sign In Here</a>

{% endblock %}

**Auth\_signin.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

</div>

<form method ='post'>

    {% csrf\_token %}

    {{form.as\_p}}

    <input type="submit" value = "Submit">

</form>

<br>

<h3>Dont have an account??</h3>

<a href = "{% url 'signup' %}">SignUp Here</a>

{% endblock %}

**Commentpost.html**

{% extends 'base.html' %}

{% block content %}

<div class="topnav">

  <a class="active" href="{% url 'home'%}">Hacker News</a>

  <a href="{% url 'new\_home'%}">New</a>

  <a href="{% url 'past\_home'%}">Past</a>

  <a href="{% url 'submit'%}">Submit</a>

  {% if request.user.is\_authenticated %}

    <div class="topnav-right">

      <a href="{% url 'signout' %}">Sign Out </a>

    </div>

  {% else %}

    <div class="topnav-right">

      <a href="{% url 'signin' %}">Sign In </a>

    </div>

  {% endif %}

</div>

<div class="w3-panel w3-light-grey w3-leftbar w3-border-grey">

<p><a href = "{{post.url}}"><strong>Title: {{post.title}}</strong></a></p>

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

<p>{{post.votes}} votes | Created {{post.created\_on}}| <a href = "{% url 'user\_info' post.creator.username %}">{{post.creator.username}}</a> | {{post.comments}} Comments | <a href="{% url 'edit' post.id %}"> Edit</a></p>

{%else %}

<p>{{post.votes}} votes | Created {{post.created\_on}}| <a href = "{% url 'user\_info' post.creator.username %}">{{post.creator.username}}</a> | {{post.comments}} Comments</p>

{% endif %}

<p><strong>Description: </strong>{{post.description}}</p>

<form method ='post'>

    {% csrf\_token %}

    {{form.as\_p}}

    <input type="submit" value = "Submit">

</form>

<br>

</div>

{% for comment in comments %}

{% if comment.identifier %}

<div class="w3-panel w3-orange w3-leftbar w3-border-red">

<p class="ex1" style="font-family:helvetica;" style="color:black"><a href = "{% url 'user\_info' comment.creator.username %}">Comment by: {{comment.creator.username}}</a> | Thread Level: {{comment.identifier}}</p>

<p class="ex1" style="font-family:helvetica;" style="color:black"><strong>{{comment.content}}</strong></p>

<p class="ex1" style="font-family:helvetica;" style="color:black"><a href = "{% url 'reply' id1=post.id id2=comment.id %}">reply</a></p>

</div>

{% else %}

<div class="w3-panel w3-red w3-leftbar w3-border-orange">

<p style="font-family:helvetica;" style="color:black"><a href = "{% url 'user\_info' comment.creator.username %}">Comment by: {{comment.creator.username}}</a> | Thread Level: {{comment.identifier}}</p>

<p style="font-family:helvetica;" style="color:black"><strong>{{comment.content}}</strong></p>

<p style="font-family:helvetica;" style="color:black"><a href = "{% url 'reply' id1=post.id id2=comment.id %}">reply</a></p>

</div>

{% endif %}

{% endfor %}

**Replay\_post.html**

{% extends 'base.html' %}

{% block content %}

<**div** class="topnav">

  <**a** class="active" href="{% url 'home'%}">Hacker News</**a**>

  <**a** href="{% url 'new\_home'%}">New</**a**>

  <**a** href="{% url 'past\_home'%}">Past</**a**>

  <**a** href="{% url 'submit'%}">Submit</**a**>

  {% if request.user.is\_authenticated %}

    <**div** class="topnav-right">

      <**a** href="{% url 'signout' %}">Sign Out </**a**>

    </**div**>

  {% else %}

    <**div** class="topnav-right">

      <**a** href="{% url 'signin' %}">Sign In </**a**>

    </**div**>

  {% endif %}

</**div**>

<**div** class="w3-panel w3-light-grey w3-leftbar w3-border-grey">

<**p**> <**h5**><**a** href = "{% url 'user\_info' comment.creator.username %}">{{comment.creator.username}}</**a**> | On : <**a** href = "{% url 'post' post.id %}">{{post.title}}</**a**></**h5**></**p**>

<**p**>{{comment.content}}</**p**>

<**form** method ='post'>

    {% csrf\_token %}

    {{form.as\_p}}

    <**input** type="submit" value = "Submit">

</**form**>

</**div**>

{% endblock %}