

# WEB-ENGINEERING PROJECT

## SS-Blogs

### **Technology Used:**

- 1.HTML 5
- 2.CSS
- 3.Bootstrap v4.1.3
4. Django -2.0

## SS-Blogs

### Source Code :

#### A. Template Files :

##### 1. blog:

###### 1.1. base.html

```
<!DOCTYPE html>
{% load static %}
<html>

<head>
<title>SS-blogs</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet">
{# MEDIUM STYLE EDITOR#}
<script src="//cdn.jsdelivr.net/medium-editor/latest/js/medium-editor.min.js"></script>
<link rel="stylesheet" href="//cdn.jsdelivr.net/medium-editor/latest/css/medium-
editor.min.css" type="text/css" media="screen" charset="utf-8">
{# Custom CSS#}
<link rel="stylesheet" href="{% static 'css/blog.css' %}">
{# Fonts#}
<link href="https://fonts.googleapis.com/css?family=Montserrat|Russo+One"
rel="stylesheet">

</head>
<body>
<div class="loader">
<nav class="navbar navbar-expand-lg bg-light techfont">
<div class="container">
<div class="navbar-nav align-items-center">
<a class="navbar-brand bigbrand" href="{% url 'post_list' %}">SS-blogs</a>
<a class="nav-item nav-link" href="{% url 'about' %}">About</a>
<a class="nav-item nav-link" href="https://www.linkedin.com">LinkedIn</a>
<a class="nav-item nav-link" href="https://www.gmail.com">Gmail</a>
<a class="nav-item nav-link" href="https://www.facebook.com">Facebook</a>
</div>
<div class="navbar-nav align-items-center ml-auto">
{% if user.is_authenticated %}
<a class="nav-item nav-link" href="{% url 'post_new' %}">New-Post</a>
<a class="nav-item nav-link" href="{% url 'post_draft_list' %}">Drafts</a>
<a class="nav-item nav-link" href="{% url 'logout' %}">Log out</a>
```

# WEB ENGINEERING PROJECT

---

```
<a >Welcome : {{ user.username }}</a>
{% else % }
<a class="nav-item nav-link" href="{% url 'login' %}" aria-label="Login">
    Login <span class="fa fa-user" aria-hidden="true"></span></a>
{% endif % }
</div>
</div>
</nav>
</div>

{# The actual blog posts#}
<div class="content container">
    <div class="row">
        <div class="col-md-8">
            <div class="blog_posts">
                {% block content % }
                {% endblock % }
            </div>
        </div>
    </div>
</div>

{# SCRIPTS#}
<script type="text/javascript" src="{% static 'js/blog.js' %}"></script>

</body>
</html>
```

## 1.2. about.html:

```
{% extends 'blog/base.html' % }
{% load static % }
{% block content % }

<h1 class="loader">About me</h1>
<p style="color:blue;"><b>Thanks for visiting SS-blogs! Check all this cool stuff about me!</b></p><br>

{% endblock % }
```

## 1.3. comment\_form.html :

```
{% extends 'blog/base.html' % }
{% block content % }
    <h1>New comment</h1>
    <form method="POST" class="post-form">
        {% csrf_token % }
        {{ form.as_p }}
        <button type="submit" class="save btn btn-default">Post Comment</button>
    </form>
    <script>var editor = new MediumEditor('.editable');</script>
```

```
{% endblock % }
```

## 1.4. post\_confirm\_delete.html :

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

```
{% block content % }
```

```
<form action="" method="post">{% csrf_token % }
  <p>Are you sure you want to delete ?"{{ object }}"?</p>
  <input class='btn btn-danger' type="submit" value="Confirm" />
</form>
```

```
{% endblock % }
```

## 1.5. post\_detail.html :

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

```
{% block content % }
```

```

    <h1 class='posttitle loader'>{{ post.title }}</h1>
    {% if post.published_date % }
        <div class="date postdate">
            {{ post.published_date }}
        </div>
    {% else % }
        <a class="btn btn-outline-dark" href="{% url 'post_publish' pk=post.pk
% }">Publish</a>
    {% endif % }
    <p class='postcontent'>{{ post.text|safe|linebreaksbr }}</p>
    {% if user.is_authenticated % }
        <a class="btn btn-primary" href="{% url 'post_edit' pk=post.pk % }"><span class="fa
fa-pencil"></span></a>
        <a class="btn btn-primary" href="{% url 'post_remove' pk=post.pk % }"><span
class="fa fa-remove"></span></a>
    {% endif % }
    <hr>
    <a class="btn btn-primary btn-comment" href="{% url 'add_comment_to_post' pk=post.pk
% }">Add comment</a>
    <div class="container">
        {% for comment in post.comments.all % }
            <br>
            {% if user.is_authenticated or comment.approved_comment % }
                {{ comment.created_date }}
                {% if not comment.approved_comment % }
                    <a class="btn btn-primary" href="{% url 'comment_remove' pk=comment.pk
% }"><span class="fa fa-remove"></span></a>
                    <a class="btn btn-primary" href="{% url 'comment_approve' pk=comment.pk
% }"><span class="fa fa-check"></span></a>
                {% endif % }
                <p>{{ comment.text|safe|linebreaks }}</p>
                <p>Posted by: <strong>{{ comment.author }}</strong></p>
            {% endif % }
            {% empty % }
                <p>No comments posted.</p>
            {% endfor % }
        </div>
    </div>
</div>
```

```
{% endfor % }
</div>
{% endblock % }
```

## 1.6. post\_draft\_list.html :

```
{% extends 'blog/base.html' % }
{% block content % }
    {% for post in posts % }
        <div class="post">
            <p class="date">Created: {{ post.created_date|date:'d-m-Y' }}</p>
            <h1><a href="{% url 'post_detail' pk=post.pk %}">{{ post.title }}</a></h1>
            <p>{{ post.text|truncatechars:200 }}</p>
        </div>
    {% endfor % }
{% endblock % }
```

## 1.7. post\_form.html:

```
{% extends 'blog/base.html' % }
{% block content % }
    <h1 style="color:red;">New post</h1>
    <br/>
    <form method="POST" class="post-form">
        {% csrf_token % }
        {{ form.as_p }}
        <button type="submit" class="save btn btn-default">Save</button>
    </form>
    <script>var editor = new MediumEditor('.editable');</script>
{% endblock % }
```

## 1.8. post\_list.html :

```
{% extends 'blog/base.html' % }
{% block content % }
<div class="centerstage">
    {% for post in post_list % }
        <div class="post">
            <h1><a href="{% url 'post_detail' pk=post.pk %}">{{ post.title }}</a></h1>
            <div class="date">
                <p>Published on: {{ post.published_date|date:"D M Y" }}</p>
            </div>
            <a href="{% url 'post_detail' pk=post.pk %}">Comments:
            {{ post.approve_comments.count }}</a>
        </div>
    {% endfor % }
</div>
{% endblock % }
```

## 2. registration :

### 1.1. login.html

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

# WEB ENGINEERING PROJECT

---

```
{% block content %}
<div class="jumbotron">
  <h2>Please Login: </h2>
  <h4>(Must be SuperUser, please check admin for issues.)</h4>
  {% if form.errors %}
    <p>Your username and password didn't match. Please try again!</p>
  {% endif %}
  <form method="POST" action="{% url 'login' %}">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" class="btn btn-primary" value="Login" />
    <input type="hidden" name="next" value="{{ next }}" />
  </form>
</div>
{% endblock %}
```

## B. Static files:

### 1. blog.css:

```
.techfont{
  font-family: 'Russo One', sans-serif;
  font-size: 1.5em;
  margin-bottom: 10px;
}

body{
  background: url(https://ak9.picdn.net/shutterstock/videos/1653199/thumb/1.jpg);
  background-size: cover;
  background-repeat: no-repeat;
  opacity:0.85;
}

.postdate{
  text-align: center;
}

.posttitle{
  font-family: 'Russo One', sans-serif;
  font-size: 3em;
  text-align: center;
}

.postcontent{
  font-family: 'Montserrat';
  font-size: 1.5em;
}

.centerstage{
  margin-left: auto;
  margin-right: auto;
```

# WEB ENGINEERING PROJECT

---

```
}

.btn-comment{
  position: absolute;
  right: 0px;
}

.bigbrand{
  font-size: 1.5em;
}

/*COLOR CHANGER*/
/*Credit and Source: http://codepen.io/thapliyalshivam/pen/dvgXVO*/
.loader{
  filter:hue-rotate(0deg);
  color: linear-gradient(45deg,#0f8,#08f);
  animation:hue 5000ms infinite linear;
}

@keyframes spinify {
  0% { transform: translate(0px,0px); }
  33% { transform: translate(0px,24px);
    border-radius:100%;
    width:10px;
    height:10px }
  66% { transform:translate(0px,-16px);}
  88% { transform:translate(0px,4px);}
  100% { transform:translate(0px,0px);}
}

@keyframes hue{
  0% {filter: hue-rotate(0deg);}
  100% {filter:hue-rotate(360deg);}
}
```

**2. Images:** owner.jpeg

## C. .PY Files:

### 1. admin.py:

```
from django.contrib import admin
from .models import Post, Comment
```

```
# Register your models here.
admin.site.register(Post)
admin.site.register(Comment)
```

### 2.apps.py:

```
from django.apps import AppConfig
class BlogConfig(AppConfig):
```

# WEB ENGINEERING PROJECT

---

```
name = 'blog'
```

## 3.models.py

```
from django.db import models
from django.utils import timezone
from django.urls import reverse

# SuperUserInformation

# User: Shivansh
# Email: supercoolshivansh@gmail.com
# Password: password@123

# User: Sahil
# Email: sahilgrover074@gmail.com
# Password: testpassword

# Creating models here.
class Post(models.Model):
    author = models.ForeignKey('auth.User', on_delete=models.CASCADE)
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(default=timezone.now)
    published_date = models.DateTimeField(blank=True, null=True)
    def publish(self):
        self.published_date = timezone.now()
        self.save()
    def approve_comments(self):
        return self.comments.filter(approved_comment=True)
    def get_absolute_url(self):
        return reverse("post_detail", kwargs={'pk':self.pk})
    def __str__(self):
        return self.title

class Comment(models.Model):
    post = models.ForeignKey('blog.Post', related_name='comments', on_delete=models.CASCADE)
    author = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(default=timezone.now)
    approved_comment = models.BooleanField(default=False)
    def approve(self):
        self.approved_comment = True
        self.save()
    def get_absolute_url(self):
        return reverse("post_list")
    def __str__(self):
        return self.text
```

## 4.forms.py:



# WEB ENGINEERING PROJECT

---

```
from django import forms
from .models import Post, Comment

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = ('author', 'title', 'text',)
        widgets = {
            'title': forms.TextInput(attrs={'class': 'textinputclass'}),
            'text': forms.Textarea(attrs={'class': 'editable medium-editor-textarea postcontent'}),
        }

class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ('author', 'text',)
        widgets = {
            'author': forms.TextInput(attrs={'class': 'textinputclass'}),
            'text': forms.Textarea(attrs={'class': 'editable medium-editor-textarea'}),
        }
```

## 5.urls.py

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

urlpatterns = [
    path("", views.PostListView.as_view(), name='post_list'),
    path('about/', views.AboutView.as_view(), name='about'),
    path('post/<int:pk>', views.PostDetailView.as_view(), name='post_detail'),
    path('post/new/', views.CreatePostView.as_view(), name='post_new'),
    path('post/<int:pk>/edit/', views.PostUpdateView.as_view(), name='post_edit'),
    path('drafts/', views.DraftListView.as_view(), name='post_draft_list'),
    path('post/<int:pk>/remove/', views.PostDeleteView.as_view(), name='post_remove'),
    path('post/<int:pk>/publish/', views.post_publish, name='post_publish'),
    path('post/<int:pk>/comment/', views.add_comment_to_post, name='add_comment_to_post'),
    path('comment/<int:pk>/approve/', views.comment_approve, name='comment_approve'),
    path('comment/<int:pk>/remove/', views.comment_remove, name='comment_remove'),]
```

## 6.views.py

```
from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from django.utils import timezone
from .models import Post, Comment
from .forms import PostForm, CommentForm
from django.views.generic import (TemplateView, ListView, DetailView, CreateView,
                                   UpdateView, DeleteView)

from django.urls import reverse_lazy
from django.contrib.auth.mixins import LoginRequiredMixin

# Creating views here.
```

# WEB ENGINEERING PROJECT

---

```
class AboutView(TemplateView):
    template_name = 'blog/about.html'

class PostListView(ListView):
    model = Post
    def get_queryset(self):
        return Post.objects.filter(published_date__lte=timezone.now()).order_by('-published_date')

class PostDetailView(DetailView):
    model = Post

class CreatePostView(LoginRequiredMixin, CreateView):
    login_url = '/login/'
    redirect_field_name = 'blog/post_detail.html'
    form_class = PostForm
    model = Post

class PostUpdateView(LoginRequiredMixin, UpdateView):
    login_url = '/login/'
    redirect_field_name = 'blog/post_detail.html'
    form_class = PostForm
    model = Post

class PostDeleteView(LoginRequiredMixin, DeleteView):
    model = Post
    success_url = reverse_lazy('post_list')

class DraftListView(LoginRequiredMixin, ListView):
    login_url = '/login/'
    redirect_field_name = 'blog/post_draft_list.html'
    model = Post
    def get_queryset(self):
        return Post.objects.filter(published_date__isnull=True).order_by('created_date')

@login_required
def post_publish(request, pk):
    post = get_object_or_404(Post, pk=pk)
    post.publish()
    return redirect('post_detail', pk=pk)

@login_required
def add_comment_to_post(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if request.method == "POST":
        form = CommentForm(request.POST)
        if form.is_valid():
            comment = form.save(commit=False)
            comment.post = post
            comment.save()
```

## WEB ENGINEERING PROJECT

---

```
        return redirect('post_detail', pk=post.pk)
    else:
        form = CommentForm()
    return render(request, 'blog/comment_form.html', {'form': form})
```

```
@login_required
def comment_approve(request, pk):
    comment = get_object_or_404(Comment, pk=pk)
    comment.approve()
    return redirect('post_detail', pk=comment.post.pk)
```

```
@login_required
def comment_remove(request, pk):
    comment = get_object_or_404(Comment, pk=pk)
    post_pk = comment.post.pk
    comment.delete()
    return redirect('post_detail', pk=post_pk)
```

### 7. 0001\_initial.py in Migrations

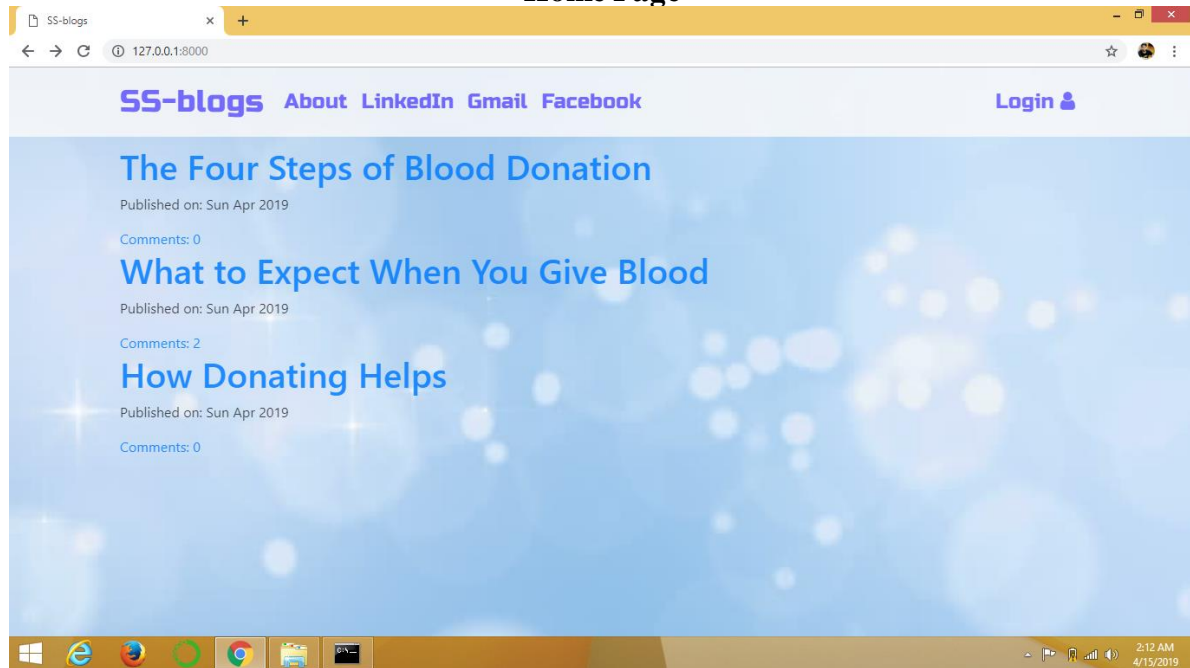
```
from django.conf import settings
from django.db import migrations, models
import django.db.models.deletion
import django.utils.timezone

class Migration(migrations.Migration):
    initial = True
    dependencies = [ migrations.swappable_dependency(settings.AUTH_USER_MODEL), ]
    operations = [ migrations.CreateModel(name='Comment',
        fields=[ ('id', models.AutoField(auto_created=True, primary_key=True,
            serialize=False, verbose_name='ID')),
            ('author', models.CharField(max_length=200)),
            ('text', models.TextField()),
            ('created_date', models.DateTimeField(default=django.utils.timezone.now)),
            ('approved_comment', models.BooleanField(default=False)), ],),
        migrations.CreateModel(
            name='Post',
            fields=[ ('id', models.AutoField(auto_created=True, primary_key=True,
            serialize=False, verbose_name='ID')),
            ('title', models.CharField(max_length=200)),
            ('text', models.TextField()),
            ('created_date', models.DateTimeField(default=django.utils.timezone.now)),
            ('published_date', models.DateTimeField(blank=True, null=True)),
            ('author', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
            to=settings.AUTH_USER_MODEL)), ],),
        migrations.AddField( model_name='comment',
            name='post', field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
            related_name='comments', to='blog.Post')),
    ]
```

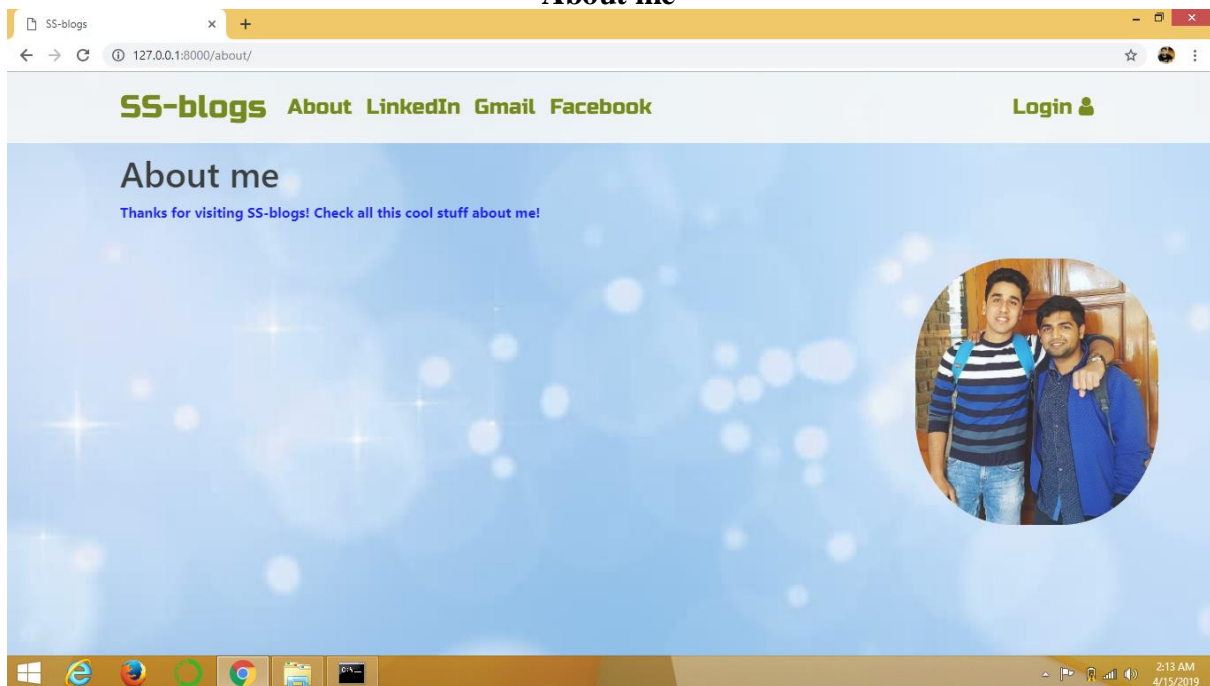
# WEB ENGINEERING PROJECT

Output :

## Home Page



## About me



# WEB ENGINEERING PROJECT

## Login

SS-blogs About LinkedIn Gmail Facebook Login

Please Login:  
(Must be SuperUser, please check admin for issues.)

Username:

Password:

Login

## Welcome Page

SS-blogs About LinkedIn Gmail Facebook New-Post Drafts Log out Welcome : Shivansh

The Four Steps of Blood Donation  
Published on: Sun Apr 2019  
Comments: 0

What to Expect When You Give Blood  
Published on: Sun Apr 2019  
Comments: 2

How Donating Helps  
Published on: Sun Apr 2019  
Comments: 0

## New Post

SS-blogs About LinkedIn Gmail Facebook New-Post Drafts Log out Welcome : Shivansh

New post

Author:

Title:

Text:

Save

# WEB ENGINEERING PROJECT

## Writing new post

Writing new post

SS-blogs About LinkedIn Gmail Facebook New-Post Drafts Log out Welcome : Sahil

New post

Author: Sahil

Title: Post by Sahil

Text:  
hey.

Save

## Showing new post

Showing new post

SS-blogs About LinkedIn Gmail Facebook New-Post Drafts Log out Welcome : Sahil

Post by Sahil

Published on:

Comments: 0

## Showing new post

Showing new post

SS-blogs About LinkedIn Gmail Facebook New-Post Drafts Log out Welcome : Shivansh

Post by Sahil

Published on: Sun Apr 2019

Comments: 0

The Four Steps of Blood Donation

Published on: Sun Apr 2019

Comments: 0

What to Expect When You Give Blood

Published on: Sun Apr 2019

Comments: 2

How Donating Helps

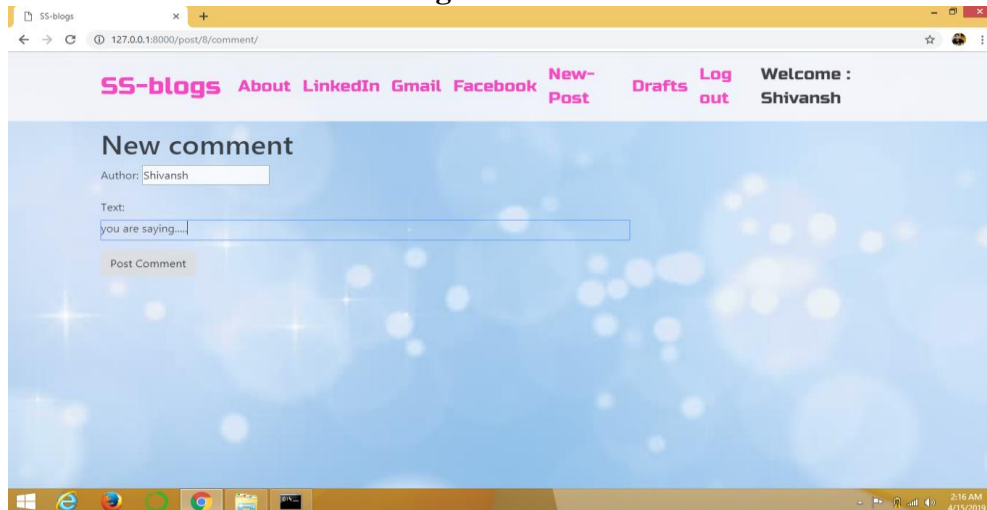
Published on: Sun Apr 2019

Comments: 0

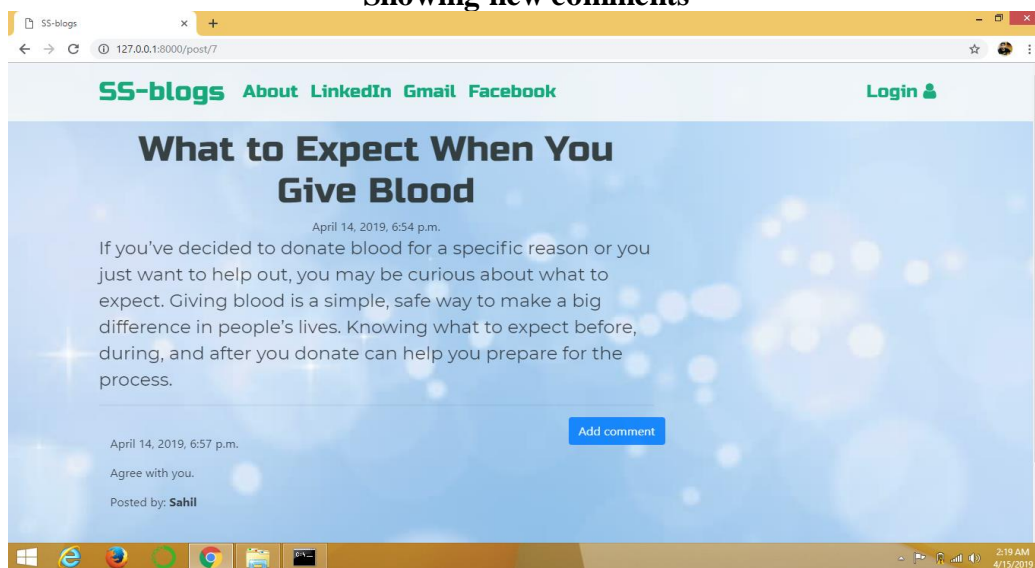


# WEB ENGINEERING PROJECT

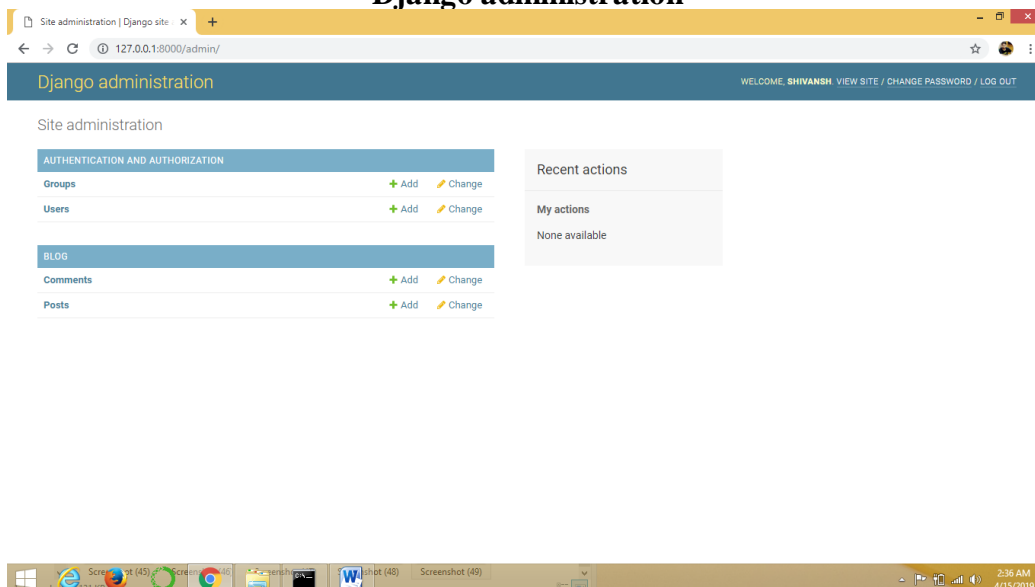
## Writing new comments



## Showing new comments

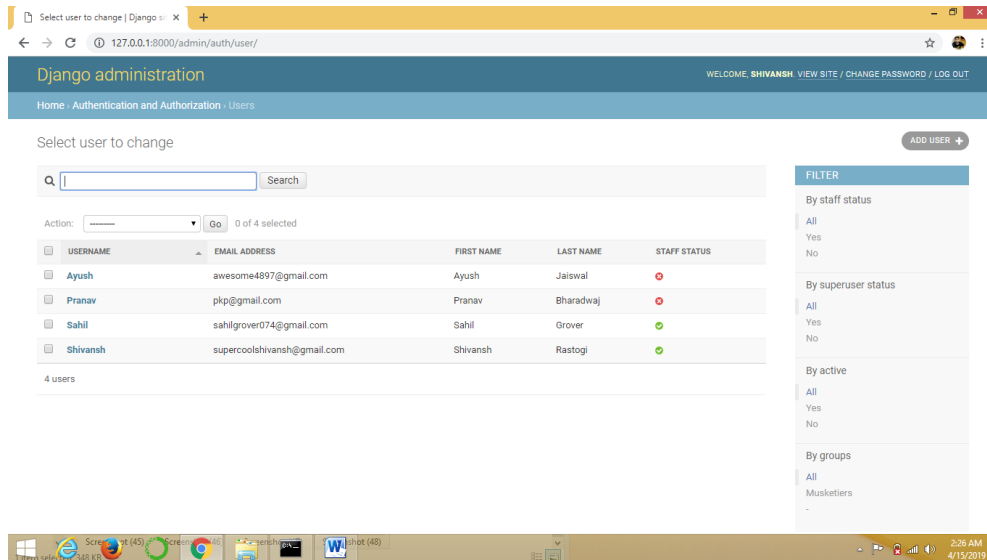


## Django administration



# WEB ENGINEERING PROJECT

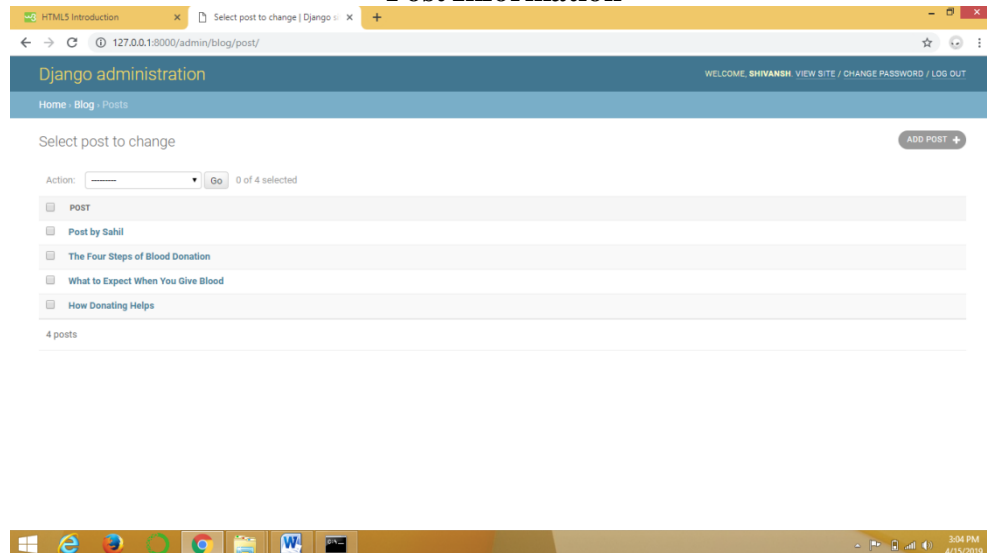
## User Information



The screenshot shows the Django administration interface for the 'Users' section. The page title is 'Select user to change'. There is a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. There are 4 users listed. On the right, there is a 'FILTER' sidebar with options for staff status, superuser status, and active status. The bottom of the screenshot shows a Windows taskbar with various icons and the system clock showing 2:25 AM on 4/15/2019.

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
Ayush	awesome4897@gmail.com	Ayush	Jaiswal	✗
Pranav	pkp@gmail.com	Pranav	Bharadwaj	✗
Sahil	sahilgrover074@gmail.com	Sahil	Grover	✓
Shivansh	supercoolshivansh@gmail.com	Shivansh	Rastogi	✓

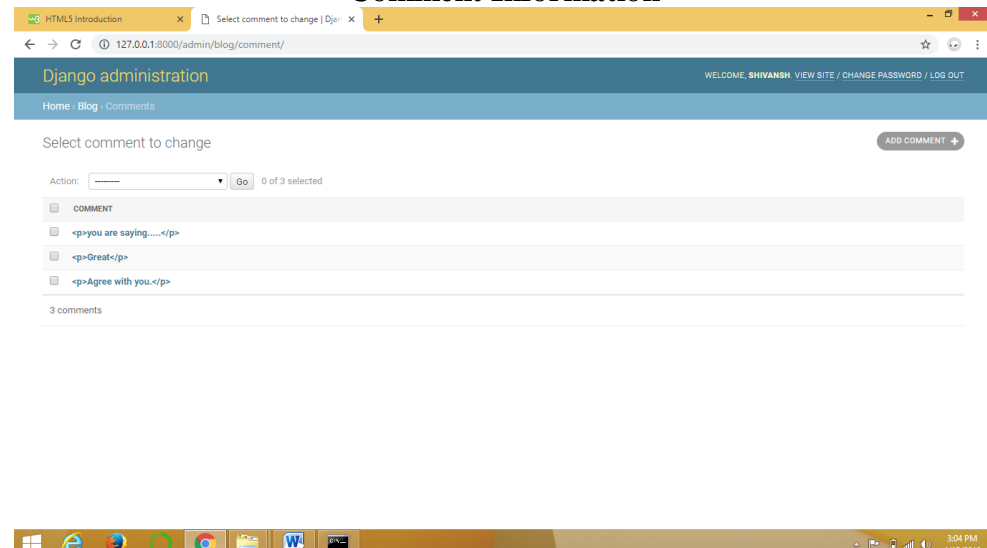
## Post Information



The screenshot shows the Django administration interface for the 'Posts' section. The page title is 'Select post to change'. There is a search bar and a list of posts. The list includes a 'POST' button and several post titles. The bottom of the screenshot shows a Windows taskbar with various icons and the system clock showing 3:04 PM on 4/15/2019.

POST
Post by Sahil
The Four Steps of Blood Donation
What to Expect When You Give Blood
How Donating Helps

## Comment Information



The screenshot shows the Django administration interface for the 'Comments' section. The page title is 'Select comment to change'. There is a search bar and a list of comments. The list includes a 'COMMENT' button and several comment texts. The bottom of the screenshot shows a Windows taskbar with various icons and the system clock showing 3:04 PM on 4/15/2019.

COMMENT
<p>you are saying.....</p>
<p>Great</p>
<p>Agree with you.</p>