**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>

<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/>

<img src="{% static "images\owner.jpg" %}" style="border-radius:40%; margin-left:900px;" alt="Uh Oh, didn't show!" height="300" width="275">

{% 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>

{% 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' %}

{% 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;

}

.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):

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:**

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.

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()

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)),

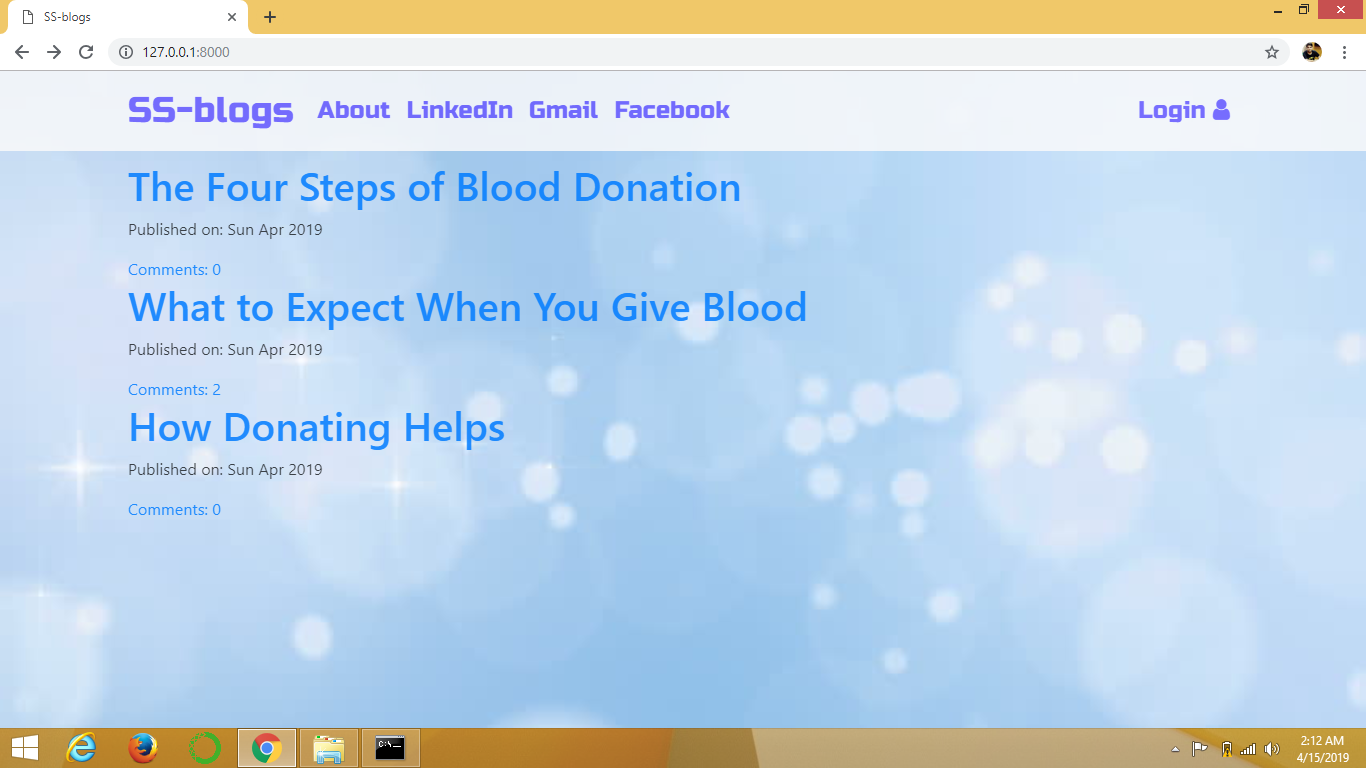
('author', models.ForeignKey(on\_delete=django.db.models.deletion.CASCADE, to=settings.AUTH\_USER\_MODEL)),],),

migrations.AddField( model\_name='comment', name='post',field=models.ForeignKey(on\_delete=django.db.models.deletion.CASCADE, related\_name='comments', to='blog.Post'),),

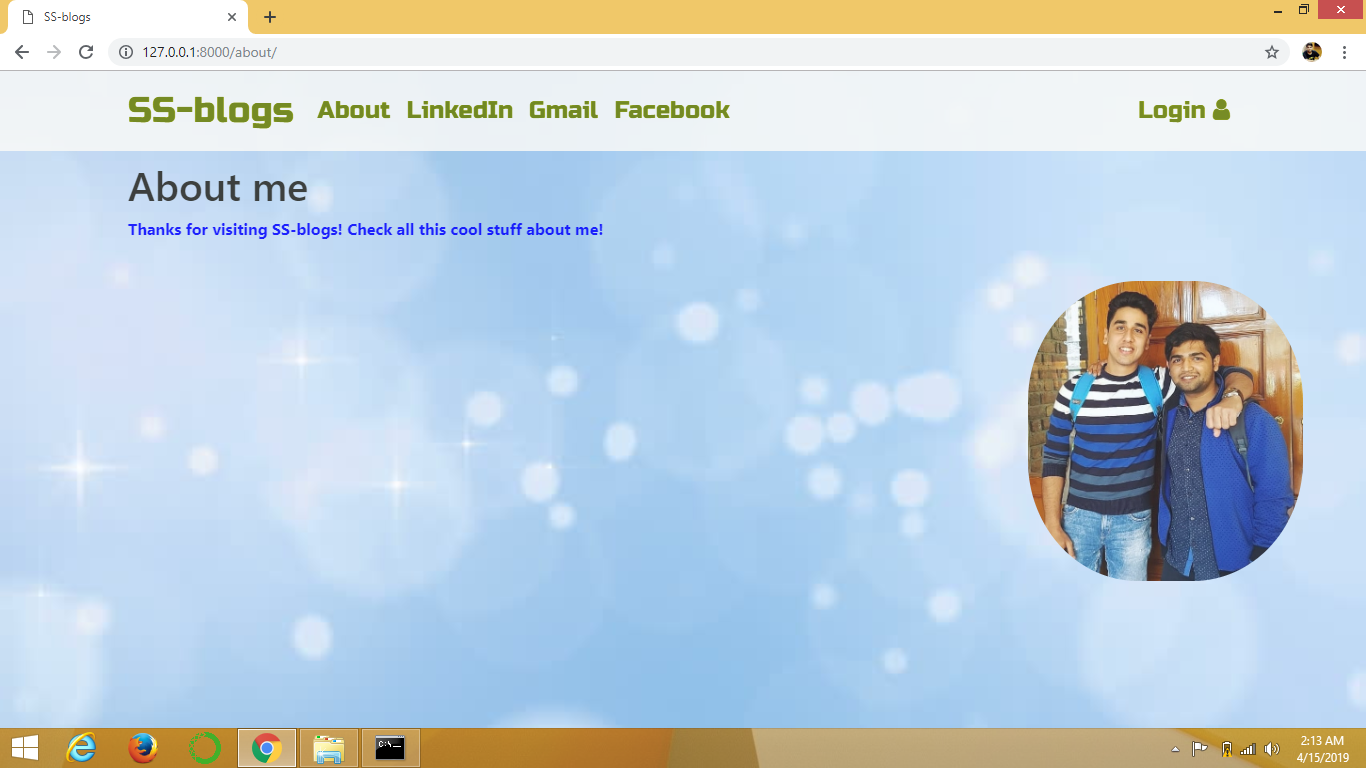
]

**Output :**

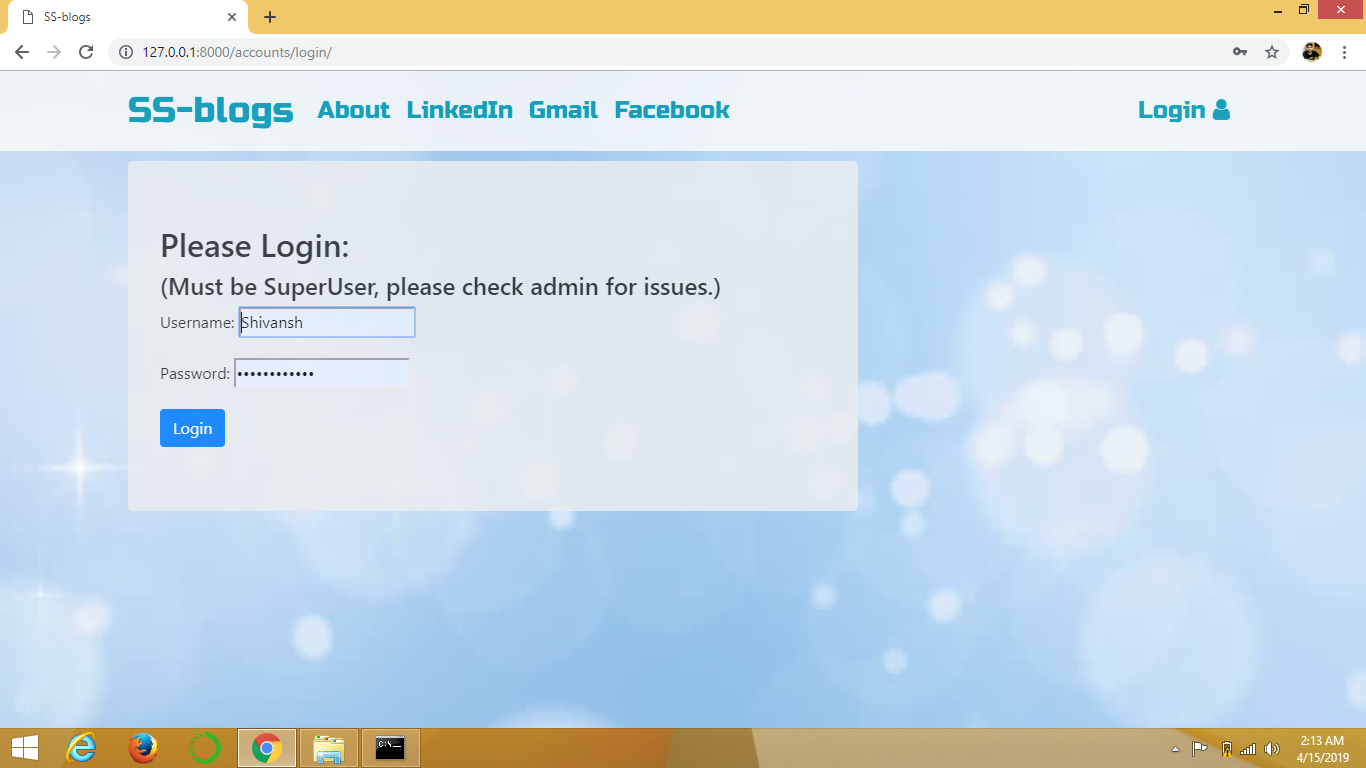
**Home Page**

****

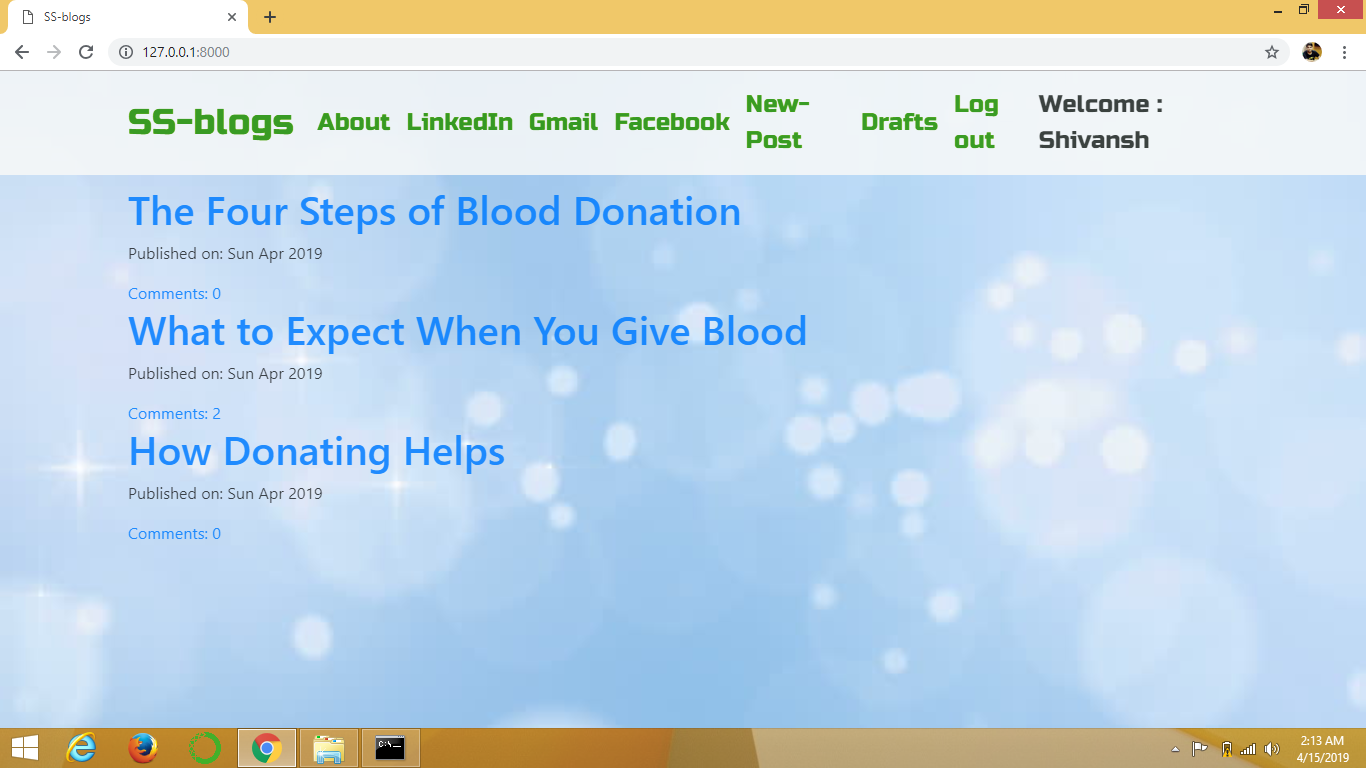
**About me**



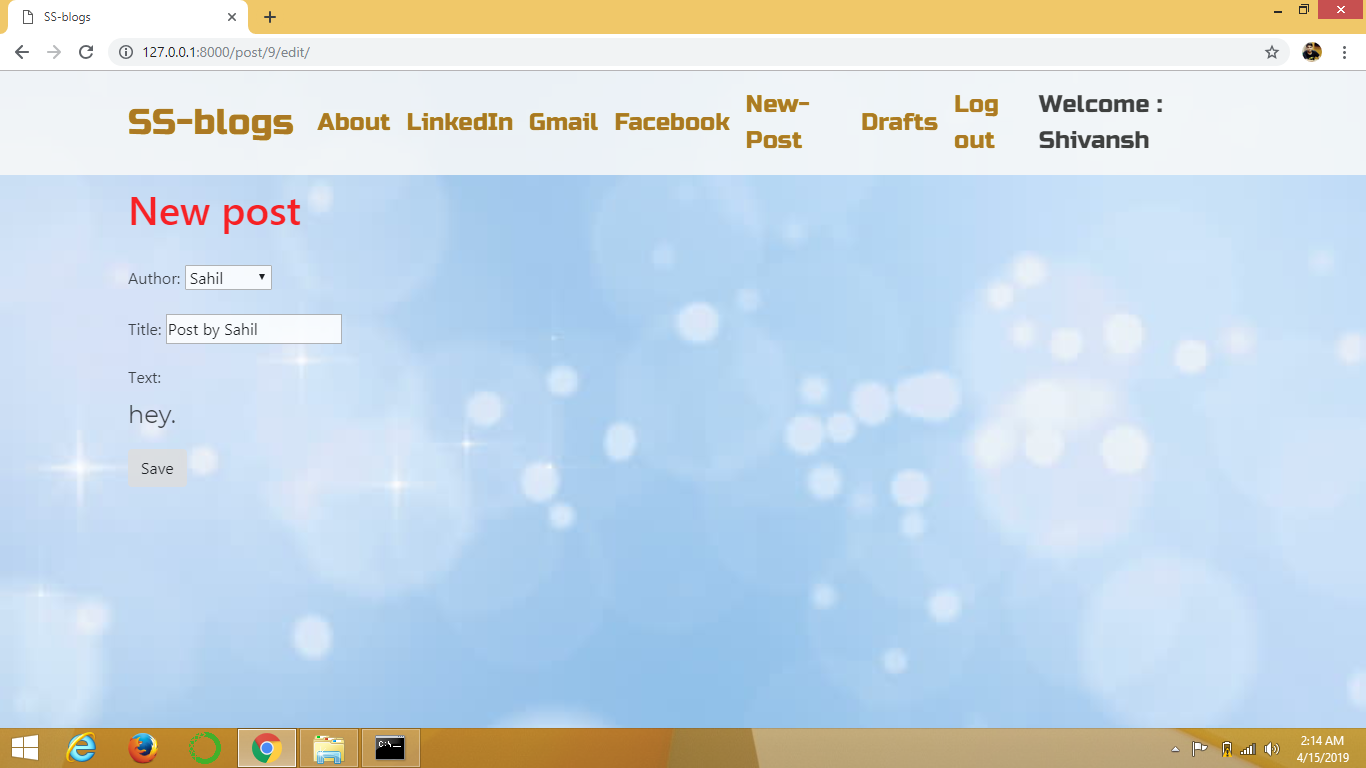
**Login**



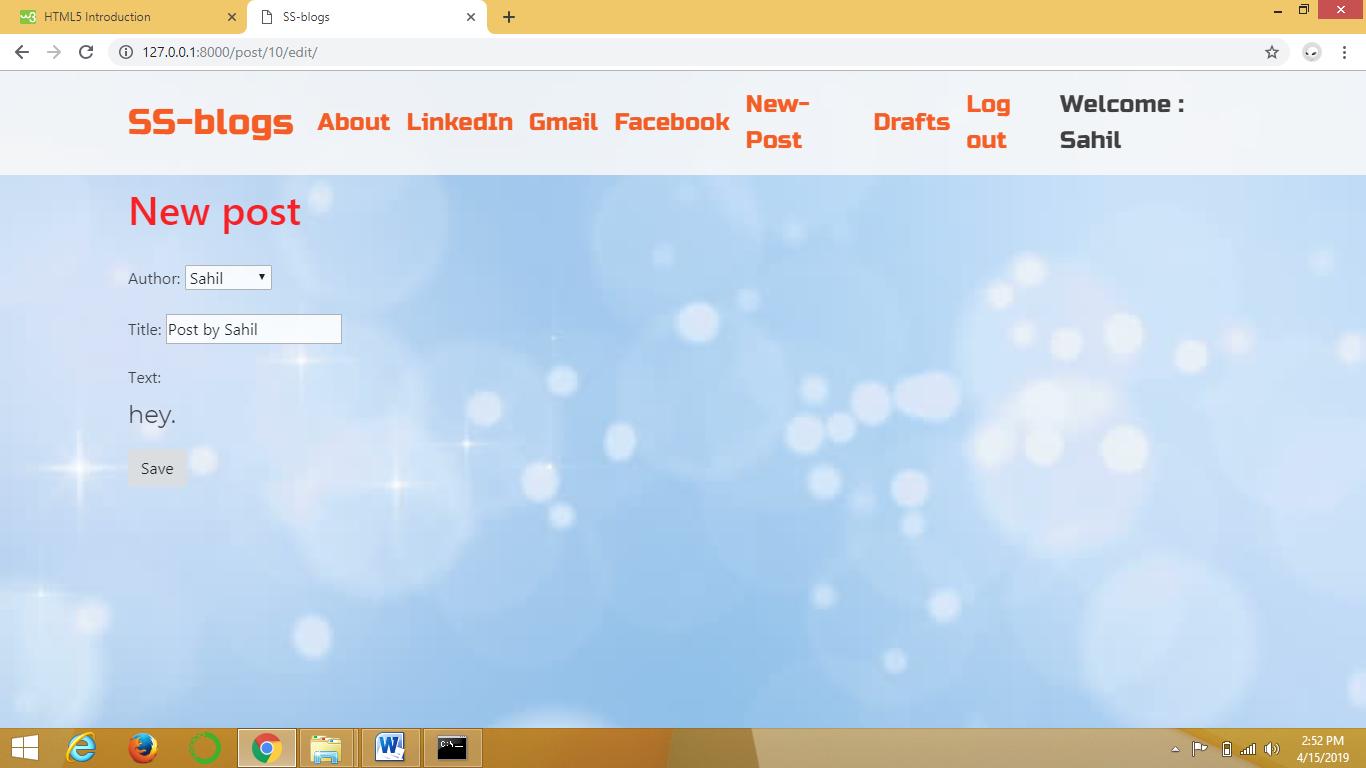
**Welcome Page**

****

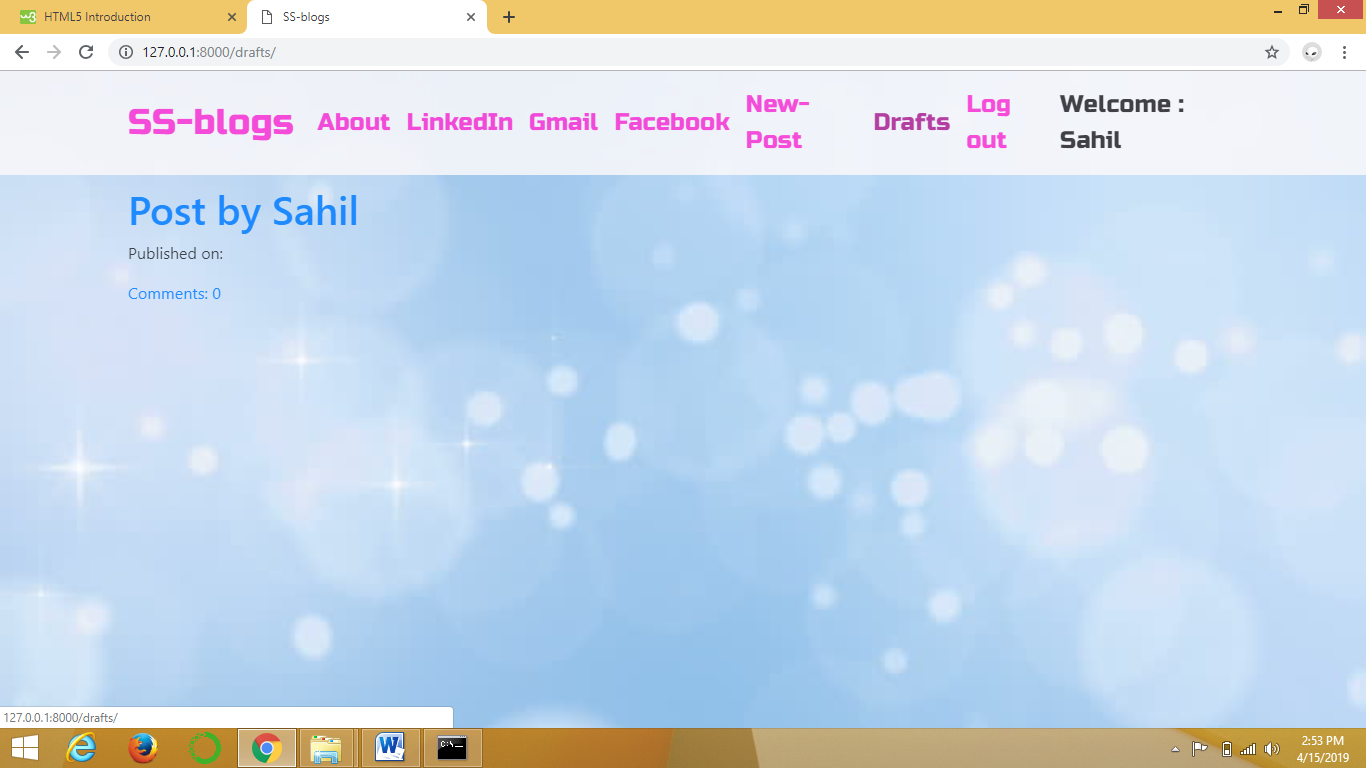
**New Post**

****

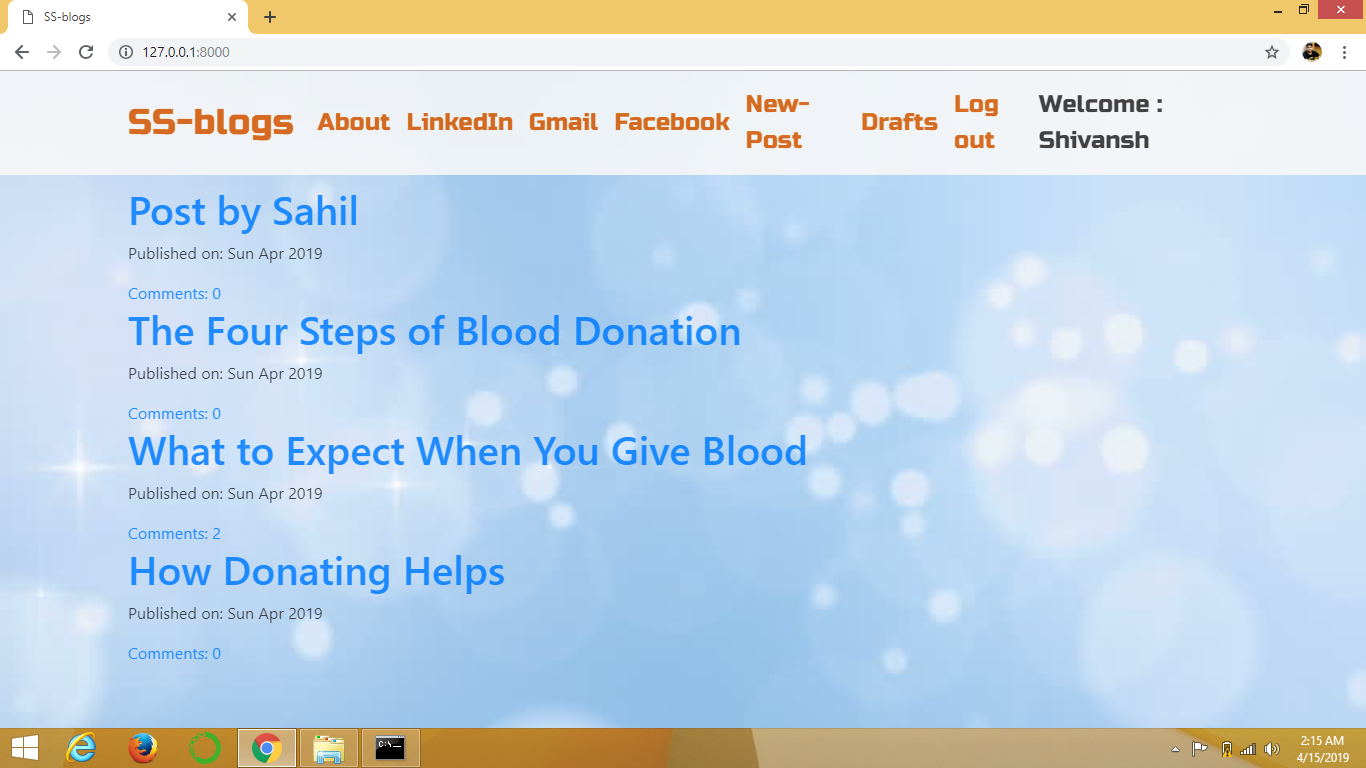
**Writing new post**

****

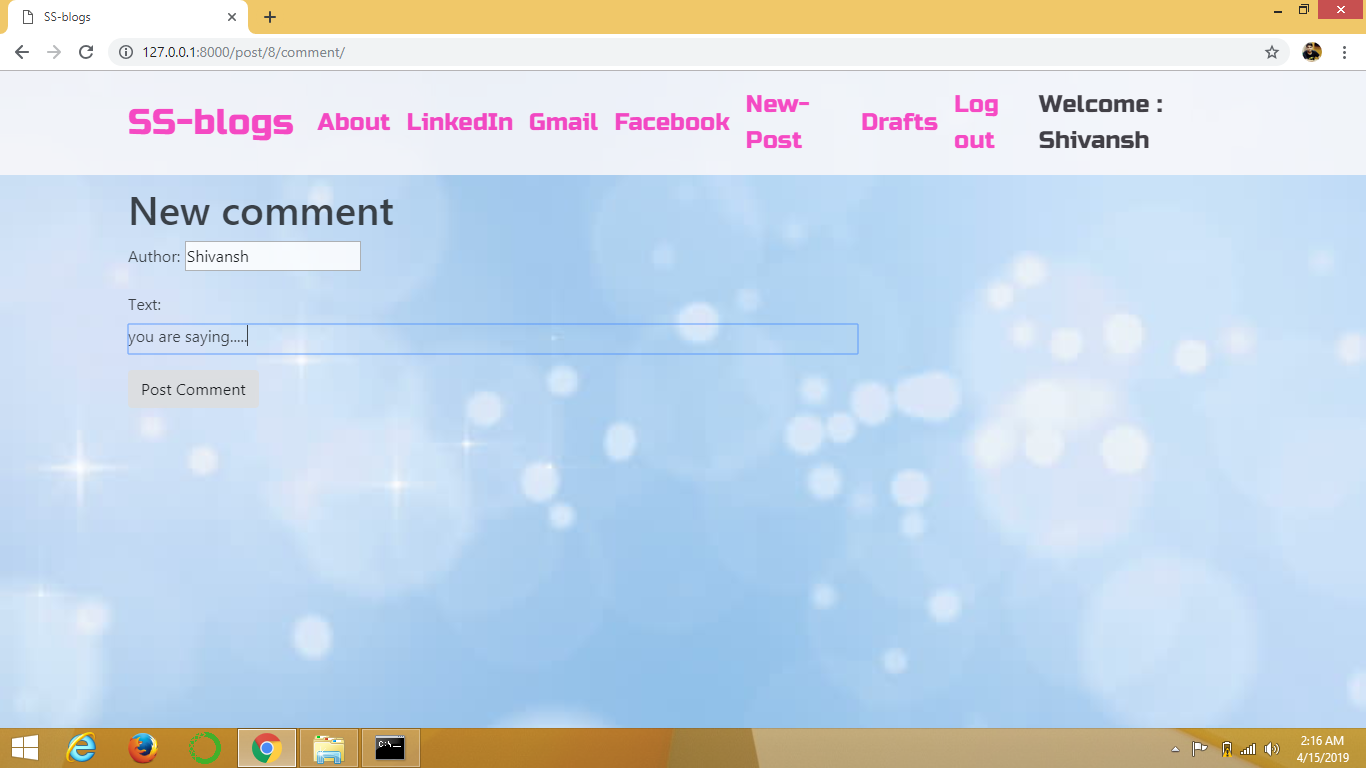
**Showing new post**

****

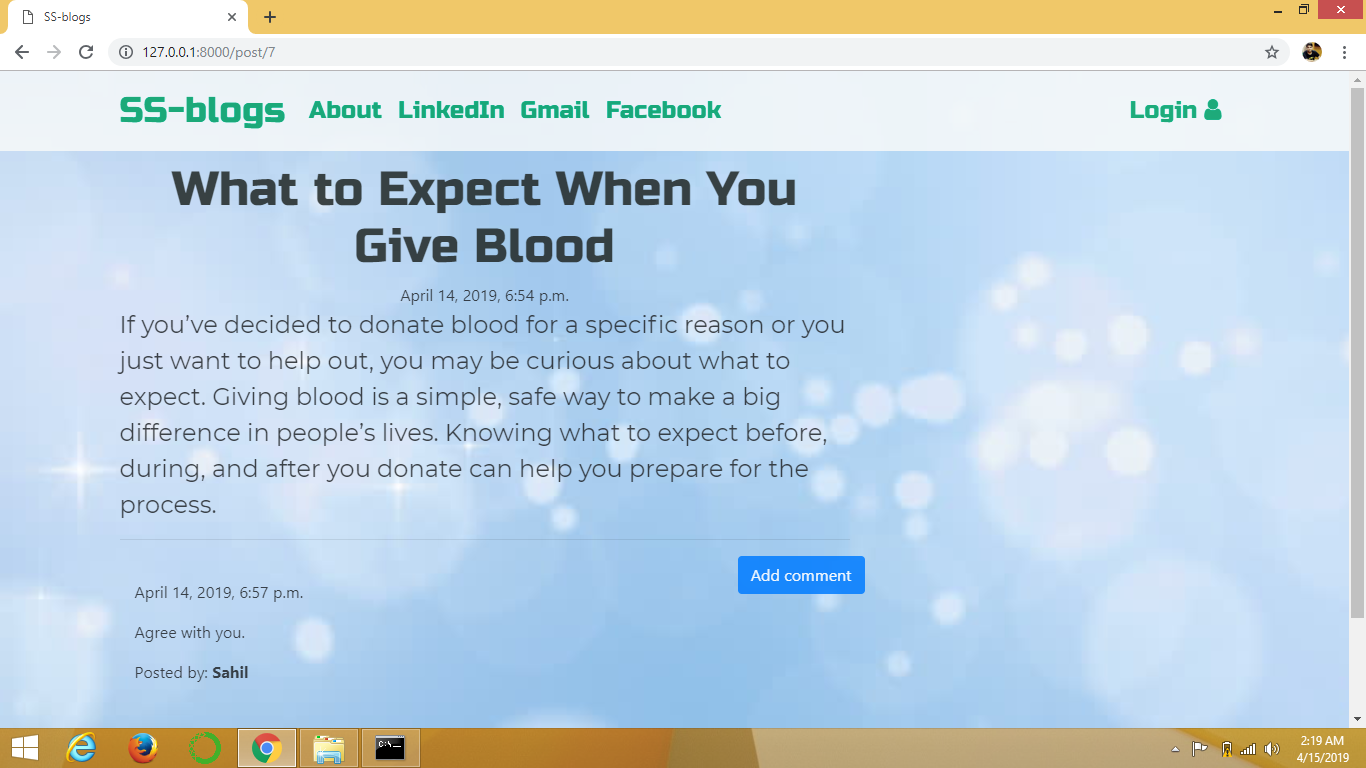
**Showing new post**

****

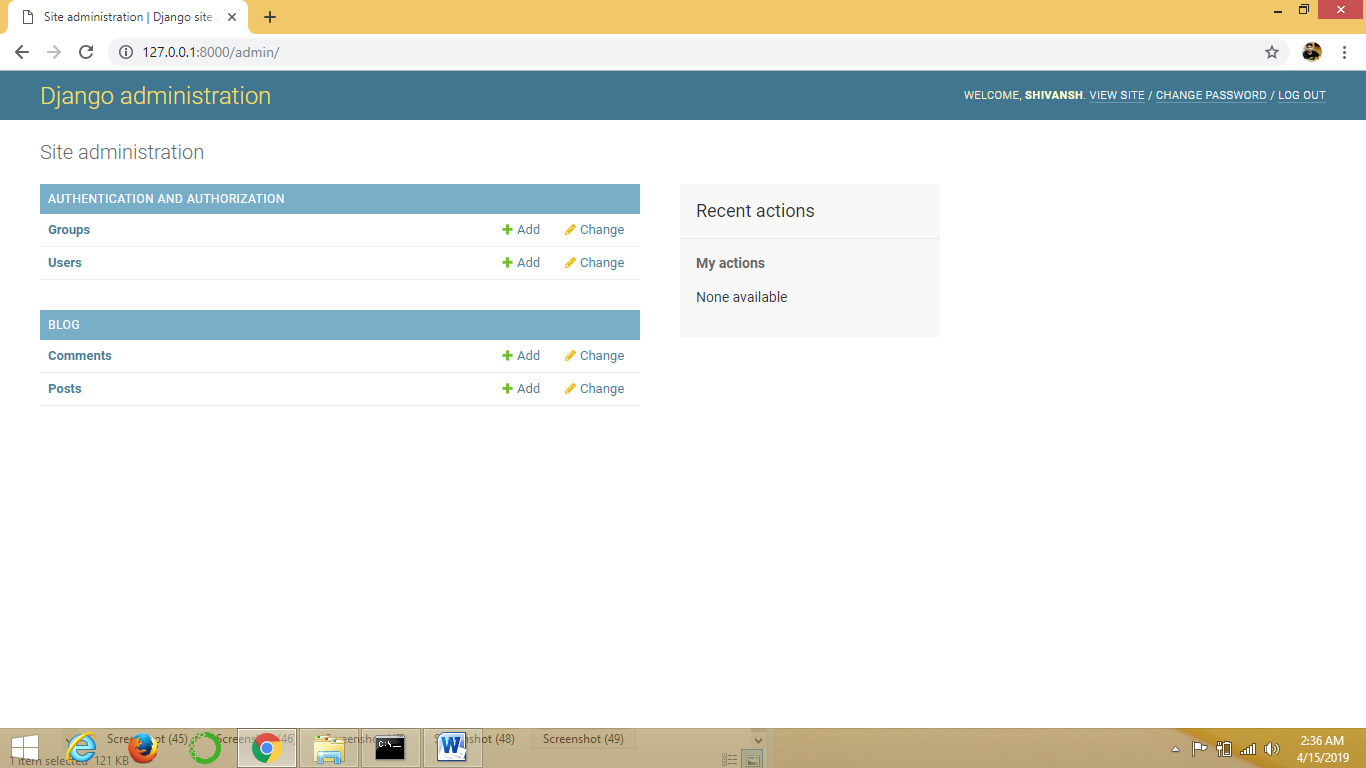
**Writing new comments**

****

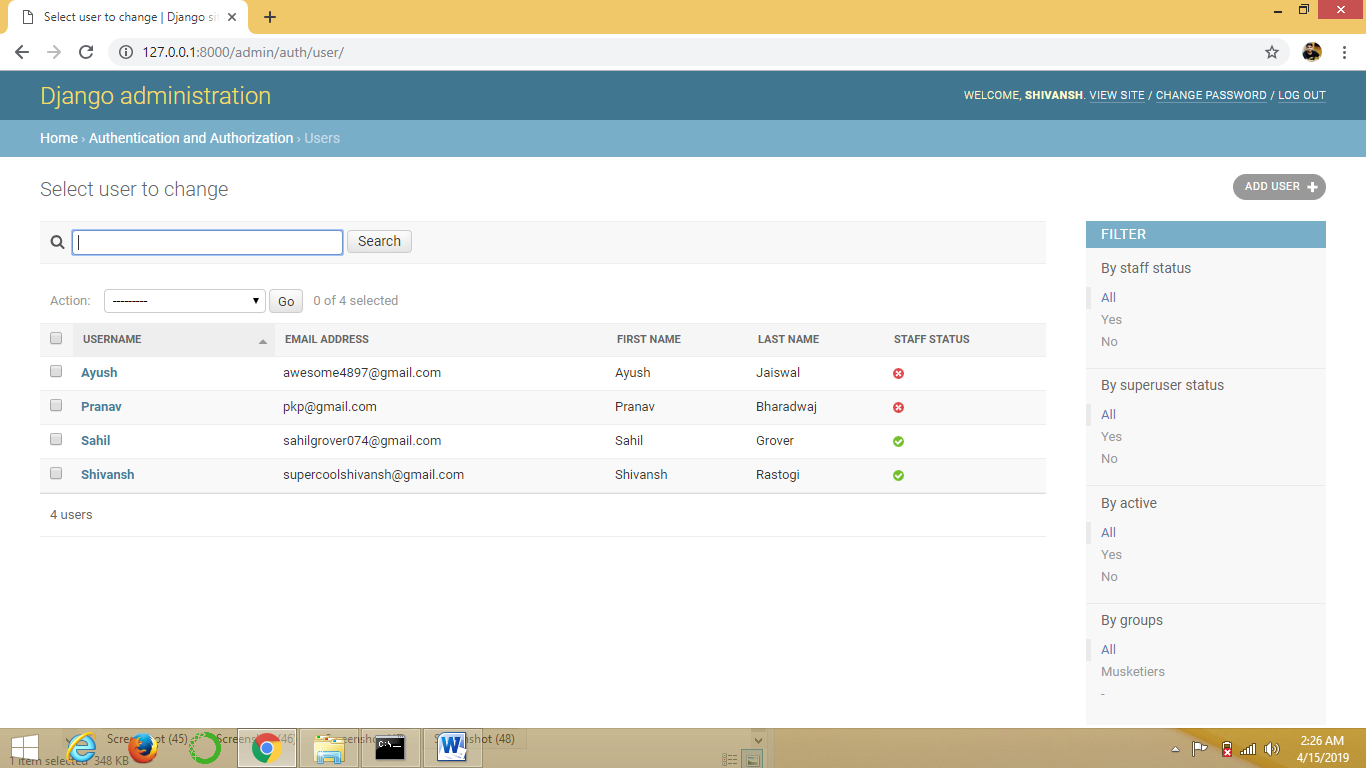
**Showing new comments**

****

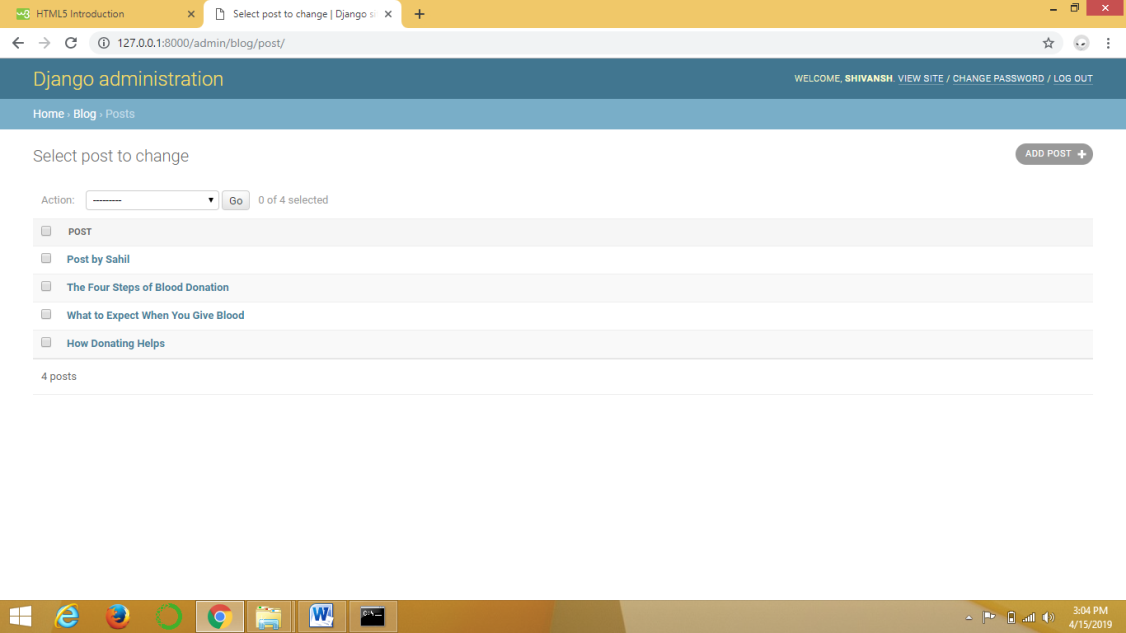
**Django administration**

****

**User Information**

****

**Post Information**

****

**Comment Information**

