Python Project

On

BLOG WEBSITE PROJECT

Second Year of Engineering in Computer Engineering

By

Class-Batch	Roll No	Name
SE4-D	54	Rajat Savdekar
SE4-D	60	Harshith Shetty
SE4-D	62	Harshbhai Solanki

DEPARTMENT OF COMPUTER ENGINEERING

SHAH AND ANCHOR KUTCHHI ENGINEERING COLLEGE

CHEMBUR, MUMBAI – 400088.

2020 – 2021

TABLE OF CONTENT

CHAPTER 1
PROBLEM DEFINATION

In recent past time blogs were store in the paper files and difficult to search or modify any information, for expanding the blogs infrastructure, awareness of environmental issues or any other issues related to education, health, digital technology, and search for greater safety give to information to all persons in all age groups and a new role within the education system, we chose this project.

We wanted to create a blog website where the administrator can **share** and post a blog online. This blog would be **read** by the visitors of the blogs website. We think that the blog is your best bet for a voice amongst the online crowd and thus wanted to solve this problem with our own solution using a high-level Python Web framework called Django.

CHAPTER 2
MODULES

admin:

Admin modules is used to store the information about the post in database. This module calls the post class in models.py with the help of post class it saves the data of the post in database.

models:

Model modules declare the data to be stored and its types and parameters and its class are called by the admin model to store the data in database.

urls:

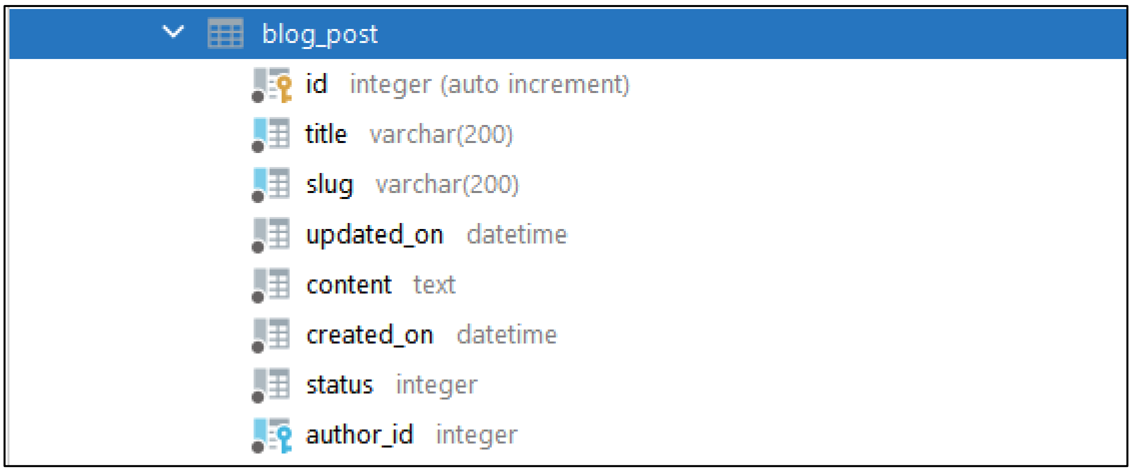
The Url modules is responsible to allocate the url to the main page as well as the detailed single blog page.

view:

View Model is called to display the main page and single blog page it also provide the list of blogs from database to the main page.

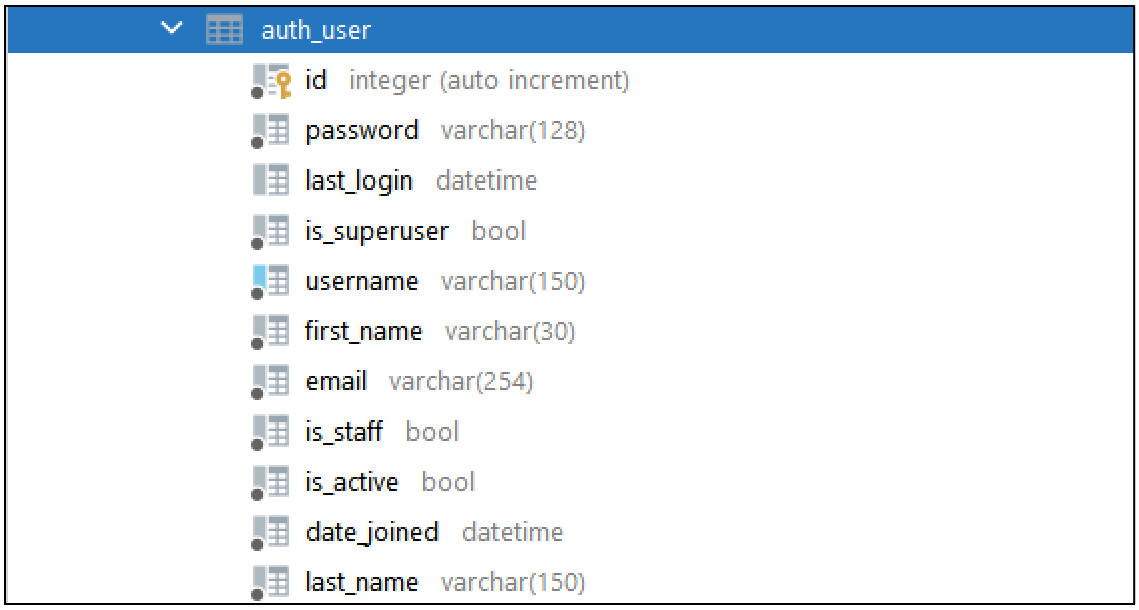
CHAPTER 3
DATABASE

Blog_post Schema



blog_post	
id	integer (auto increment)
title	varchar(200)
slug	varchar(200)
updated_on	datetime
content	text
created_on	datetime
status	integer
author_id	integer

Auth_user Schema



auth_user	
id	integer (auto increment)
password	varchar(128)
last_login	datetime
is_superuser	bool
username	varchar(150)
first_name	varchar(30)
email	varchar(254)
is_staff	bool
is_active	bool
date_joined	datetime
last_name	varchar(150)

CHAPTER 4
IMPLEMENTATION

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)
```

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
```

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'),
]
```

view.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'
```

app.py:

```
from django.apps import AppConfig


class BlogConfig(AppConfig):
    name = 'blog'
```

base.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mind Blogs</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' %}" >Blogs</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>
    </div>
    </nav>


            {% block content %}
          <!-- Content Goes here -->
            {% endblock content %}

    <!-- Footer -->
    <footer class="py-3 bg-grey">
              <p class="m-0 text-dark text-center "></p>
    </footer>

  </body>
</html>
```

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 Mind Blog </h3>
                    <p class="text-light">Mind is an open platform where
readers find dynamic thinking &nbsp
                    </p>
                </div>
                    </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 %}
```

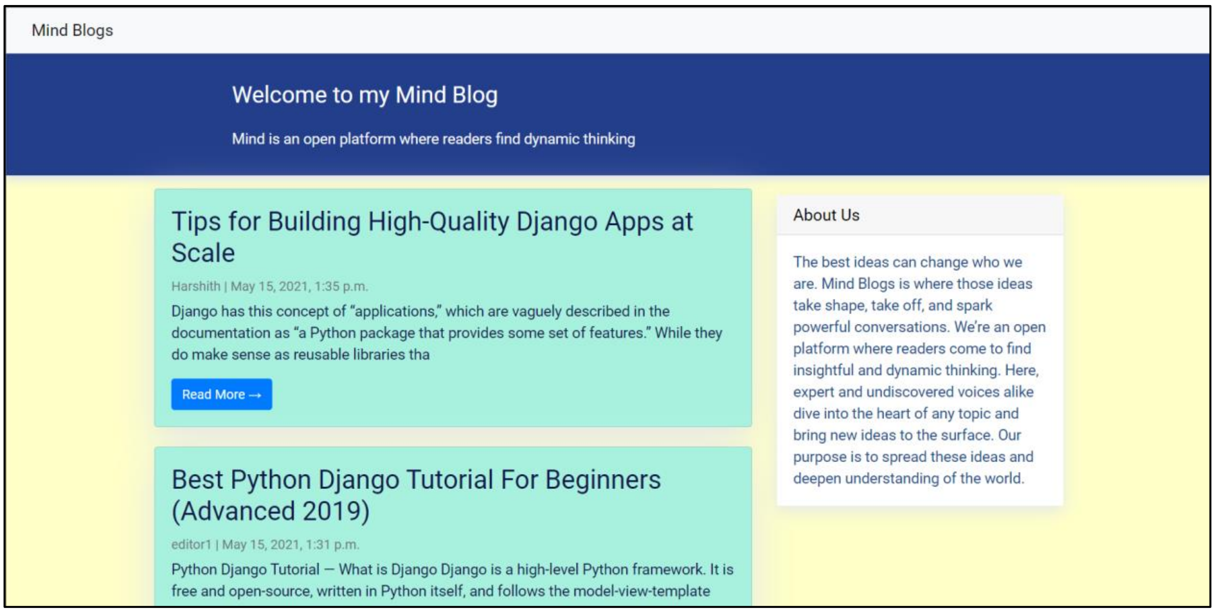
sidebar.html:

```
{% 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"> The best ideas can change who we are. Mind
```

```
Blogs is where those ideas take shape, take off, and spark powerful
conversations. We’re an open platform where  readers come to find
insightful and dynamic thinking. Here, expert and undiscovered voices alike
dive into the heart of any topic and bring new ideas to the surface. Our
purpose is to spread these ideas and deepen understanding of the world.</p>
        <a href="#"
           class="btn btn-danger">Know more!</a>
    </div>
</div>
</div>
{% endblock sidebar %}
```

CHAPTER 5
RESULT

Home Page:



Mind Blogs

Welcome to my Mind Blog

Mind is an open platform where readers find dynamic thinking

Tips for Building High-Quality Django Apps at Scale

Harshith | May 15, 2021, 1:35 p.m.

Django has this concept of "applications," which are vaguely described in the documentation as "a Python package that provides some set of features." While they do make sense as reusable libraries tha

Read More →

Best Python Django Tutorial For Beginners (Advanced 2019)

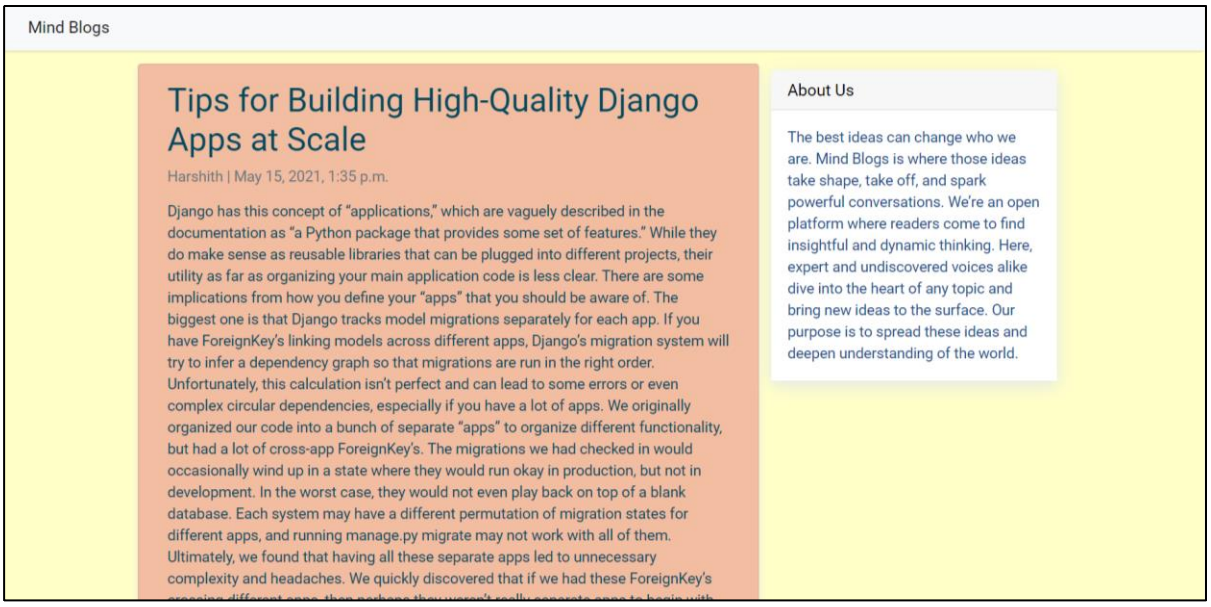
editor1 | May 15, 2021, 1:31 p.m.

Python Django Tutorial — What is Django Django is a high-level Python framework. It is free and open-source, written in Python itself, and follows the model-view-template

About Us

The best ideas can change who we are. Mind Blogs is where those ideas take shape, take off, and spark powerful conversations. We're an open platform where readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Our purpose is to spread these ideas and deepen understanding of the world.

Single Post Page:



Mind Blogs

Tips for Building High-Quality Django Apps at Scale

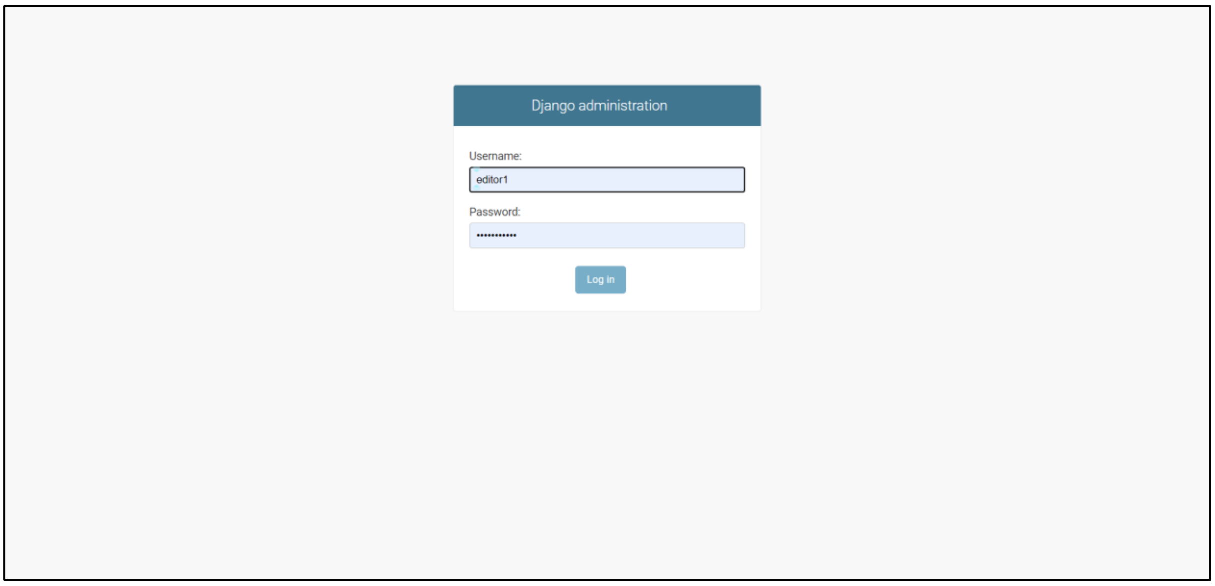
Harshith | May 15, 2021, 1:35 p.m.

Django has this concept of "applications," which are vaguely described in the documentation as "a Python package that provides some set of features." While they do make sense as reusable libraries that can be plugged into different projects, their utility as far as organizing your main application code is less clear. There are some implications from how you define your "apps" that you should be aware of. The biggest one is that Django tracks model migrations separately for each app. If you have ForeignKey's linking models across different apps, Django's migration system will try to infer a dependency graph so that migrations are run in the right order. Unfortunately, this calculation isn't perfect and can lead to some errors or even complex circular dependencies, especially if you have a lot of apps. We originally organized our code into a bunch of separate "apps" to organize different functionality, but had a lot of cross-app ForeignKey's. The migrations we had checked in would occasionally wind up in a state where they would run okay in production, but not in development. In the worst case, they would not even play back on top of a blank database. Each system may have a different permutation of migration states for different apps, and running manage.py migrate may not work with all of them. Ultimately, we found that having all these separate apps led to unnecessary complexity and headaches. We quickly discovered that if we had these ForeignKey's

About Us

The best ideas can change who we are. Mind Blogs is where those ideas take shape, take off, and spark powerful conversations. We're an open platform where readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Our purpose is to spread these ideas and deepen understanding of the world.

Admin Login Page:



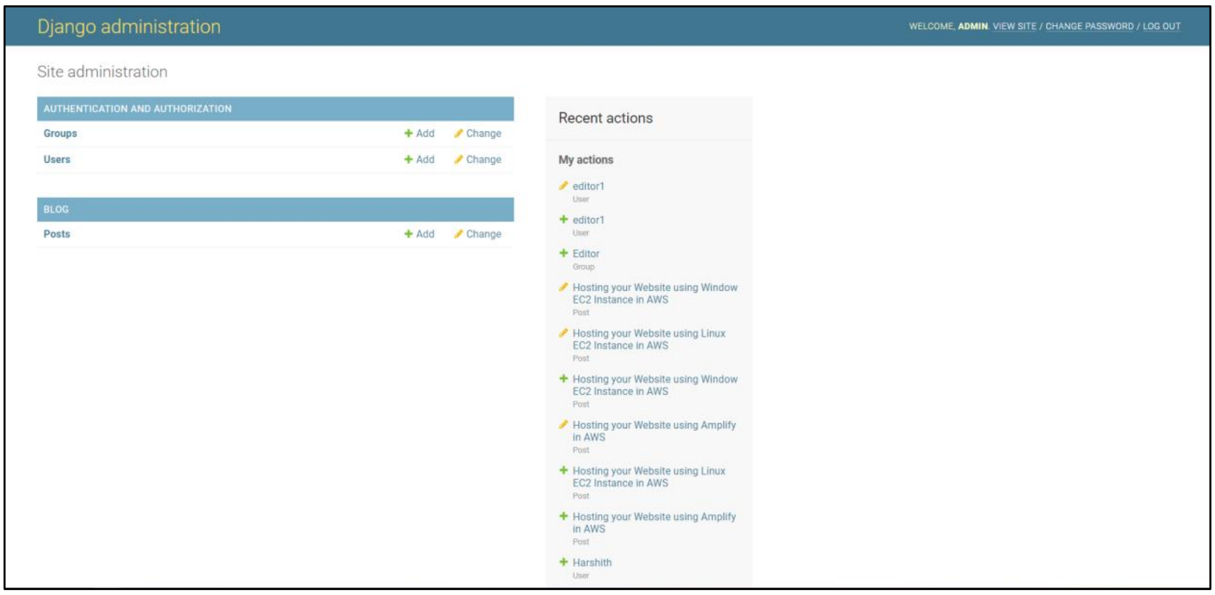
Django administration

Username:
editor1

Password:
•••••••••••

Log in

Admin Home Page:



Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

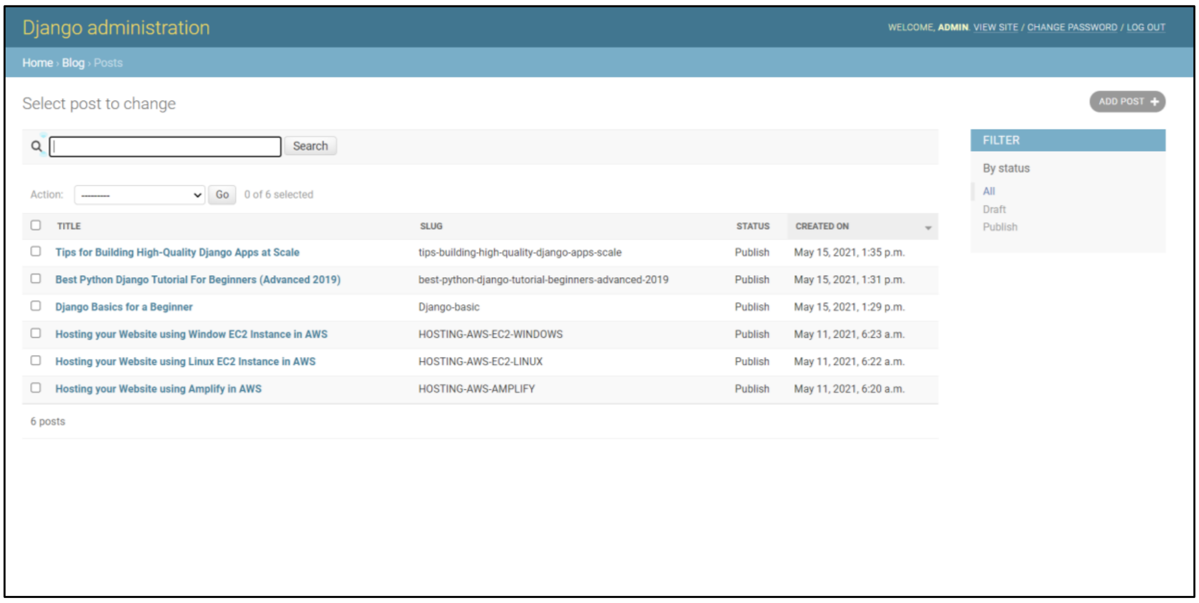
BLOG		
Posts	+ Add	Change

Recent actions

My actions

- editor1
  User
- + editor1
  User
- + Editor
  Group
- Hosting your Website using Window EC2 Instance in AWS
  Post
- Hosting your Website using Linux EC2 Instance in AWS
  Post
- + Hosting your Website using Window EC2 Instance in AWS
  Post
- Hosting your Website using Amplify in AWS
  Post
- + Hosting your Website using Linux EC2 Instance in AWS
  Post
- + Hosting your Website using Amplify in AWS
  Post
- + Harshith
  User

Admin Home Page:



Select post to change

Action: --------- Go 0 of 6 selected

TITLE	SLUG	STATUS	CREATED ON
Tips for Building High-Quality Django Apps at Scale	tips-building-high-quality-django-apps-scale	Publish	May 15, 2021, 1:35 p.m.
Best Python Django Tutorial For Beginners (Advanced 2019)	best-python-django-tutorial-beginners-advanced-2019	Publish	May 15, 2021, 1:31 p.m.
Django Basics for a Beginner	Django-basic	Publish	May 15, 2021, 1:29 p.m.
Hosting your Website using Window EC2 Instance in AWS	HOSTING-AWS-EC2-WINDOWS	Publish	May 11, 2021, 6:23 a.m.
Hosting your Website using Linux EC2 Instance in AWS	HOSTING-AWS-EC2-LINUX	Publish	May 11, 2021, 6:22 a.m.
Hosting your Website using Amplify in AWS	HOSTING-AWS-AMPLIFY	Publish	May 11, 2021, 6:20 a.m.

6 posts

CHAPTER 6
CONCLUSION

Working on this project has been very interesting as we got to know about advancement in the field of Graphical User Interface (GUI) technology. Also giving us an idea of development of a GUI based application using python's Django framework as language.

So, our work aims to develop a blog sharing website which is based on administrator needs and has proved to be useful in the amidst of this pandemic, where not being able to attend university, and missing many events where we can share our thoughts is limited , this can help in search of similar minded people who share the same interest as the administrator of the blog .

REFERENCE

1. Django documentation | Django documentation | Django (djangoproject.com)
2. Getting started with Django | Django (djangoproject.com)
3. https://buildmedia.readthedocs.org/media/pdf/django/latest/django.pdf