JaneAldridge / DjangoForensics Public

Watch 1 · Fork 0 · Star 0

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights

main · DjangoForensics / Group2 / internetForensics / views.py / <> Jump to · Go to file · ···

MUTEGIbeatrice UPDATES 2 ... Latest commit 0343527 2 days ago · History

2 contributors

105 lines (64 sloc) | 4.22 KB · Raw · Blame

```python

#DJANGO IMPORTS

from django.contrib import admin
from django.conf import settings
from django.shortcuts import render, redirect #to allow redirection from one page to another pages
from django.http import HttpResponse #FOR HTTP RESPONSE
from django.contrib.auth.models import User #FOR IMPORTING USERS DATABASE
from django.contrib.auth import authenticate, login, logout #FOR AUTH AND LOGOUT
from django.contrib.auth.models import Group #FOR IMPORTING USERS GROUP MODELS
from django.contrib import messages #FOR ERROR AND INFORMATION/STATUS MESSAGES
from django.contrib.auth.decorators import login_required #FOR ENABLING VIEW RESTRICTIONS UNTIL LOGGED IN
from datetime import date, timedelta



# Create your views here.
from .models import * #FOR IMPORTING ALL MODELS
from .decorators import unauthenticated_user, allowed_users #FOR IMPORTING THE USER ACCESS CONTROL DECORATORS
from django.contrib.auth import views as auth_views



 # LOGIN  REQUEST, AUTHENTICATION AND RESTRICTION TO AN ALREADY AUTHENTICATED USER

@unauthenticated_user # only accessed to unauthenticated users and restriction put on validated users who try to go back to login page via the URL

def Login(request): #DEHINING THE LOGIN AUTHENTICATION

    if request.method == 'POST': #authentication
        username = request.POST.get("username")
        password = request.POST.get("password")

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

        if user is not None:
            messages.success(request, 'Login Successful')
            login(request,user)
            return redirect('mainpage')
        else:
            messages.info(request, 'Invalid Username or Password')# Return an 'invalid login' error message.

    return render(request, 'internetForensics/login.html')

# PASSWORD CHANGING
class PasswordChangeView(auth_views.PasswordChangeView):
    template_name = 'internetForensics/changepassword.html'



# LOG OUT REQUEST
def logoutUser(request):
    logout(request)
    return redirect('login')




# VIEW REQUESTS FOR THE REST OF THE PAGES AND ACCESS RESTRICTIONS

@login_required(login_url='login')   # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
def mainpage(request):
    return render(request, 'internetForensics/mainpage.html')



@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
@allowed_users(allowed_roles=['admin','managers']) # ACCESS CONTROL AS PER THE USER ROLES
def manage_reports(request): #to view manage_report page
    return render(request, 'internetForensics/manage_reports.html')


@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
def create_reports(request): #to view create_report page
    return render(request, 'internetForensics/create_reports.html')


@login_required(login_url='login')   # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
def criminalActivity(request): #to view search and locate criminal activities page
    return render(request, 'internetForensics/criminalactivity.html')
```

```python



@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
@allowed_users(allowed_roles=['admin','managers','senior_security_analysts']) # ACCESS CONTROL AS PER THE USER ROLES
def audits(request): #to view audits page
    return render(request, 'internetForensics/audits.html')



@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
def cases(request): #to view cases page
    return render(request, 'internetForensics/cases.html')



@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
@allowed_users(allowed_roles=['admin','managers']) # ACCESS CONTROL AS PER THE USER ROLES
def userslogs(request): #to view userslogs page
    return render(request, 'internetForensics/userslogs.html')


@login_required(login_url='login')    # RESTRICTION ON UNAUTHENTICATED USERS AND DERIVED FROM 'IMPORT LOGIN_REQUIRED'
def changepassword(request): #to view userslogs page
    return render(request, 'internetForensics/changepassword.html')
```