**SAMPLE CODE**

**Store Manager(views.py):**

from django.shortcuts import render , redirect

from django.conf import settings

from django.contrib.auth.models import User

from django.contrib.auth import authenticate, login

from django.contrib.auth.decorators import login\_required

from django.contrib import messages

from Owner.models import Employee\_Enrol

from django.core.files.storage import FileSystemStorage

import cv2

import os

import numpy as np

from skimage.feature import blob\_dog

import matplotlib.pyplot as plt

# Create your views here.

#########################################################################

def StoreManager\_Login(request):

    if request.method =='POST':

        username = request.POST['username']

        password = request.POST['pass']

        try:

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

            if user is not None:

                login(request, user)

                return redirect('StoreManagerHome')

            else:

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

               return render(request, 'StoreManagerLogin.html')

        except User.DoesNotExist:

            messages.error(request, 'something went wrong')

    return render(request , 'StoreManagerLogin.html')

#########################################################################

def StoreManagerHome(request):

    return render(request, 'StoreManager/StoreManagerHome.html')

###########################################################################

import os

import cv2

import numpy as np

import matplotlib.pyplot as plt

from django.conf import settings

from django.core.files.storage import FileSystemStorage

from skimage.feature import blob\_dog

from skimage.util import img\_as\_float  # Import for correct image conversion

def upload\_image(request):

    if request.method == 'POST':

        image = request.FILES['file-upload']

        fs = FileSystemStorage(location=settings.MEDIA\_ROOT)

        fs.save(image.name, image)

        file\_url = fs.url(image.name)

        # Define the base directory for processing images

        base\_dir = os.path.join(settings.MEDIA\_ROOT, 'retail\_images')

        def get\_foreground(img\_path):

            h\_kernel = np.ones((5, 3), np.uint8)

            # Read the image in color format

            color = cv2.imread(img\_path)

            # Check if the image was loaded correctly

            if color is None:

                raise ValueError(f"Error reading the image file: {img\_path}")

            gray = cv2.cvtColor(color, cv2.COLOR\_BGR2GRAY)

            # Ensure the gray image is correctly typed

            gray = gray.astype(np.uint8)

            # Threshold the image using Otsu

            ret3, th3 = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU)

            # Create the minimum and maximum threshold values for Canny Edge detection

            hist\_eq = cv2.equalizeHist(gray)

            min\_thresh = 0.66 \* np.median(hist\_eq)

            max\_thresh = 1.33 \* np.median(hist\_eq)

            # Perform Canny edge detection

            edge\_1 = cv2.Canny(gray, int(min\_thresh), int(max\_thresh))

            # Dilate the edges to make them more visible

            edge\_2 = cv2.dilate(edge\_1, None, iterations=1)

            image\_th\_inv = cv2.bitwise\_not(cv2.dilate(edge\_2, h\_kernel, iterations=2))

            # Ensure the resulting image is the correct data type

            image\_th\_inv = image\_th\_inv.astype(np.uint8)

            return image\_th\_inv, gray, ret3

        def combine\_bounding\_boxes(upp\_th, blob\_list\_sorted):

            i = 0

            while i < len(blob\_list\_sorted):

                j = i + 1

                while j < len(blob\_list\_sorted):

                    # Convert coordinates to float to ensure numeric operations

                    x1, y1, r1 = map(float, blob\_list\_sorted[i])

                    x2, y2, r2 = map(float, blob\_list\_sorted[j])

                    # Check proximity of bounding boxes to combine them

                    if max([abs(x - y) for x, y in zip([x1, y1], [x2, y2])]) < upp\_th:

                        new\_box = [min(x1, x2), min(y1, y2), max(x1 + r1, x2 + r2) - min(x1, x2)]

                        blob\_list\_sorted[i] = new\_box

                        del blob\_list\_sorted[j]

                    else:

                        j += 1

                i += 1

            return blob\_list\_sorted

        def show\_segmentation(img\_path, \_image):

            original = cv2.imread(img\_path)

            if original is None:

                raise ValueError(f"Error reading the image file: {img\_path}")

            image\_th\_inv, gray, th = get\_foreground(img\_path)

            # Ensure image\_th\_inv is correctly thresholded

            image\_th\_inv = np.where(image\_th\_inv > th, 255, 0).astype(np.uint8)

            \_image = cv2.erode(image\_th\_inv, np.ones((3, 3), np.uint8), iterations=1)

            # Display the original and processed images side by side

            fig, axes = plt.subplots(1, 2, figsize=(12, 6))

            ax = axes.ravel()

            ax[0].imshow(cv2.cvtColor(original, cv2.COLOR\_BGR2RGB))

            ax[1].imshow(\_image, cmap='gray')

            plt.show()

            # Save the processed image

            output\_path = os.path.join("CV2\_Segmentation", os.path.basename(img\_path))

            cv2.imwrite(output\_path, \_image)

            return \_image

        def process\_image(image\_filename):

            img\_path = os.path.join(base\_dir, image\_filename)

            original = cv2.imread(img\_path)

            if original is None:

                raise ValueError(f"Error reading the image file: {img\_path}")

            upp\_th = 0.00392

            count = 0

            \_image = cv2.imread(img\_path, 0)

            if \_image is None:

                raise ValueError(f"Error reading the image file: {img\_path}")

            # Convert the grayscale image to float format for blob\_dog

            \_image\_float = img\_as\_float(\_image)

            # Detect blobs using Difference of Gaussians (DoG)

            blobs\_dog = blob\_dog(\_image\_float, max\_sigma=40, threshold=.10)

            # Convert blobs to integer format to avoid type errors

            blob\_list\_sorted = sorted([[int(blob[0]), int(blob[1]), int(blob[2])] for blob in blobs\_dog])

            blob\_list\_sorted = combine\_bounding\_boxes(upp\_th, blob\_list\_sorted)

            results\_cv2 = {"img\_name": [], "bbox": [], "coords": []}

            filtered = []

            contours\_ = []

            # Draw bounding boxes around detected blobs

            for blob in blob\_list\_sorted:

                y, x, r = blob

                if r > 6:  # Filter based on blob size

                    original = cv2.rectangle(original, (int(x), int(y)), (int(x + r), int(y + r)), (255, 0, 0), 2)

                    count += 1

                    results\_cv2["img\_name"].append(str(image\_filename.split(".")[0]))

                    results\_cv2["bbox"].append(count)

                    results\_cv2["coords"].append([x, y, x + r, y + r])

            fig, axes = plt.subplots(1, 2, figsize=(15, 10))

            ax = axes.ravel()

            ax[0].imshow(original)

            # Find contours in the processed image

            contours, \_ = cv2.findContours(\_image, cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)

            objects = np.zeros([original.shape[0], original.shape[1], 3], 'uint8')

            # Draw contours on the original image

            for c in contours:

                if cv2.contourArea(c) < (\_image.shape[0] \* \_image.shape[1]) \* .10:

                    x = [int(val[0][0]) for val in c]

                    y = [int(val[0][1]) for val in c]

                    filtered.append([x, y])

                    contours\_.append(c)

                    cv2.drawContours(original, [c], -1, (255, 0, 0), -1)

            ax[1].imshow(original)

            plt.show()

            # Save the image with identified empty spaces

            output\_path = os.path.join("Identified\_empty\_spaces", os.path.basename(img\_path))

            cv2.imwrite(output\_path, original)

            return results\_cv2, blob\_list\_sorted, filtered, contours\_

        # Example usage of the process\_image function

        image\_filename = image.name

        results\_cv2, blob\_list\_sorted, filtered, contours\_ = process\_image(image\_filename)

    return render(request, 'StoreManager/UploadImage.html')

**Home.html**

<!DOCTYPE html>

<html>

{%load static%}

<head>

  <!-- Basic -->

  <meta charset="utf-8" />

  <meta http-equiv="X-UA-Compatible" content="IE=edge" />

  <!-- Mobile Metas -->

  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

  <!-- Site Metas -->

  <meta name="keywords" content="" />

  <meta name="description" content="" />

  <meta name="author" content="" />

  <title>Efficient Shelf Monitoring System</title>

  <!-- slider stylesheet -->

  <link rel="stylesheet" type="text/css"

    href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- bootstrap core css -->

  <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.css' %}" />

  <!-- fonts style -->

  <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700|Poppins:400,700|Raleway:400,700&display=swap"

    rel="stylesheet" />

  <!-- Custom styles for this template -->

  <link href="{% static 'Css/style.css'%}" rel="stylesheet" />

  <!-- responsive style -->

  <link href="{% static 'Css/responsive.css'%}" rel="stylesheet" />

  <link rel="icon" type="image/png" href="images/icons/favicon.ico" />

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/bootstrap/css/bootstrap.min.css' %}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/fonts/font-awesome-4.7.0/css/font-awesome.min.css' %}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-font.min.css">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/animate/animate.css'%}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/css-hamburgers/hamburgers.min.css'%}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/animsition/css/animsition.min.css'%}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/select2/select2.min.css'%}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/vendor/daterangepicker/daterangepicker.css'%}">

  <!--===============================================================================================-->

  <link rel="stylesheet" type="text/css" href="{% static 'Css/util.css'%}">

  <link rel="stylesheet" type="text/css" href="{% static 'Css/main.css'%}">

</head>

<body>

  <div class="hero\_area">

    <!-- header section strats -->

    <header class="header\_section">

      <div class="container-fluid">

        <nav class="navbar navbar-expand-lg custom\_nav-container">

          <a class="navbar-brand" href="">

            <img src="{% static 'Images/logo.png'%}" alt="" />

            <span style="color:black">

              Efficient

              Shelf

              Monitoring System

            </span>

          </a>

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

            <div class="custom\_menu-btn">

              <button onclick="openNav()">

                <span class="s-1"> </span>

                <span class="s-2"> </span>

                <span class="s-3"> </span>

              </button>

            </div>

            <div id="myNav" class="overlay">

              <div class="overlay-content">

                <a href="{% url 'home' %}">HOME</a>

                <a href="{% url 'StoreManagerLogin' %}">Store Manager</a>

                <a href="{% url 'OwnerLogin' %}">Owner</a>

              </div>

            </div>

          </div>

        </nav>

      </div>

    </header>

    <!-- end header section -->

    <!-- slider section -->

    <section class=" slider\_section position-relative">

      <div class="side\_heading">

        <h5>

          Empty

          Shelf

          Detection

        </h5>

      </div>

      <div class="container-fluid">

        <div class="row">

          <div class="col-lg-3 col-md-4 offset-md-1">

            <div id="carouselExampleIndicators" class="carousel slide " data-ride="carousel">

              <ol class="carousel-indicators">

                <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active">

                  01

                </li>

                <li data-target="#carouselExampleIndicators" data-slide-to="1">

                  02

                </li>

                <li data-target="#carouselExampleIndicators" data-slide-to="2">

                  03

                </li>

                <li data-target="#carouselExampleIndicators" data-slide-to="3">

                  04

                </li>

              </ol>

              <div class="carousel-inner">

                <div class="carousel-item active">

                  <div class="img-box b-1">

                    <img src="{% static 'Images/store.jpeg' %}" alt="" />

                  </div>

                </div>

                <div class="carousel-item">

                  <div class="img-box b-2">

                    <img src="{% static 'Images/store.jpeg' %}" alt="" />

                  </div>

                </div>

                <div class="carousel-item">

                  <div class="img-box b-3">

                    <img src="{% static 'Images/store.jpeg' %}" alt="" />

                  </div>

                </div>

                <div class="carousel-item">

                  <div class="img-box b-4">

                    <img src="{% static 'Images/store.jpeg' %}" alt="" />

                  </div>

                </div>

              </div>

              <div class="carousel\_btn-box">

                <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">

                  <span class="sr-only">Previous</span>

                </a>

                <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">

                  <span class="sr-only">Next</span>

                </a>

              </div>

            </div>

          </div>

          <div class=" col-md-5 offset-md-1">

            <div class="detail-box">

              {% block contents %}

              <h1>

                Efficient <br>

                Shelf Monitoring

              </h1>

              <p>

                In the fast paced world of today where marketing

                and advertisement is at its peak, an effective inventory

                management is essential for success. Remote Inventory

                management is traditionally done through Image processing

                methods in which the images of the shelves are taken and then

                processed to find the missing objects

              </p>

              {% endblock%}

              <div class="btn-box">

                <a href="{% url 'home'%}" class="btn-1">

                  Home

                </a>

                <a href="{% url 'StoreManagerLogin'%}" class="btn-1">

                  Store Manager

                </a>

                <a href="{%  url 'OwnerLogin' %}" class="btn-1">

                  Owner

                </a>

              </div>

            </div>

          </div>

        </div>

      </div>

    </section>

    <!-- end slider section -->

  </div>

  <section class="container-fluid footer\_section">

    <div class="social\_container">

      <h4>

        Follow on

      </h4>

      <div class="social-box">

        <a href="">

          <img src="{% static 'Images/fb.png'%}" alt="">

        </a>

        <a href="">

          <img src="{% static 'Images/twitter.png' %}" alt="">

        </a>

        <a href="">

          <img src="{% static 'Images/linkedin.png' %}" alt="">

        </a>

        <a href="">

          <img src="{% static 'Images/youtube.png' %}" alt="">

        </a>

      </div>

    </div>

    <p>

      &copy; 2025 All Rights Reserved. Design by

      <a href=""></a>

    </p>

  </section>

  <!-- footer section -->

  <script type="text/javascript" src="{% static 'js/jquery-3.4.1.min.js' %}"></script>

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

  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js">

  </script>

  <script>

    function openNav() {

      document.getElementById("myNav").classList.toggle("menu\_width");

      document

        .querySelector(".custom\_menu-btn")

        .classList.toggle("menu\_btn-style");

    }

  </script>

  <!-- owl carousel script -->

  <script type="text/javascript">

    $(".owl-carousel").owlCarousel({

      loop: true,

      margin: 35,

      navText: [],

      autoplay: true,

      autoplayHoverPause: true,

      responsive: {

        0: {

          items: 1

        },

        600: {

          items: 2

        },

        1000: {

          items: 3

        }

      }

    });

  </script>

  <!-- end owl carousel script -->

</body>

</html>

**Owner (Views.py)**

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

from django.contrib import messages

from django.contrib.auth.models import User

from django.contrib.auth import authenticate, login

from django.contrib.auth.decorators import login\_required

from Owner.models import Employee\_Enrol

from Owner.forms import Employee\_EditForm , Employee\_EditForm1

# Create your views here.

def OwnerLogin(request):

    if request.method == 'POST':

        username = request.POST['username']

        password = request.POST['pass']

        if (username == 'admin' and password == 'admin') or (username == 'owner' and password == 'owner'):

            return redirect('OwnerHome')

        else:

            return render(request, 'ownerlogin.html', {'error': 'Invalid username or password'})

    return render(request, 'ownerlogin.html')

def OwnerHome(request):

    return render(request, 'owner/OwnerHome.html')

def Emp\_Register(request):

    if request.method == 'POST':

        Name = request.POST['name']

        username = request.POST['username']

        password = request.POST['pass']

        phone = request.POST['phone']

        email = request.POST['email']

        if User.objects.filter(username=username).exists():

            messages.info(request, 'Username already exists')

            return redirect('EmpRegister')

        elif Employee\_Enrol.objects.filter(phone=phone).exists():

            messages.info(request, 'Phone number already exists')

            return redirect('EmpRegister')

        elif User.objects.filter(email=username).exists():

            messages.info(request, 'Email already exists')

            return redirect('EmpRegister')

        else:

            user = User.objects.create\_user(username=username, password=password)

            user.save()

            if Name and phone:

                emp = Employee\_Enrol.objects.create( name=Name, phone=phone , email=email,)

                emp.save()

            messages.info(request, 'Account created successfully')

            return redirect('EmpRegister')

    return render(request, 'owner/Empregister.html')

def Employee\_View(request):

    data = User.objects.all()

    data1= Employee\_Enrol.objects.all()

    return render(request, 'owner/EmployeeView.html' , {'combined': zip(data, data1)})

def Emp\_Edit(request , id , employee\_id):

     user  =get\_object\_or\_404( User,id =id )

     employee=get\_object\_or\_404(Employee\_Enrol,id=employee\_id )

     if request.method == 'POST':

        form = Employee\_EditForm1(request.POST, instance=user)

        form1 = Employee\_EditForm(request.POST, instance=employee)

        if form.is\_valid() and form1.is\_valid():

               user\_instance = form.save(commit=False)

               if form.cleaned\_data['password']:

                 user\_instance.set\_password(form.cleaned\_data['password'])

               user\_instance.save()

               form1.save()

               messages.success(request, 'Profile updated successfully')

               return redirect('Employeeview')

     else:

        form = Employee\_EditForm1(instance=user)

        form1 = Employee\_EditForm(instance=employee)

     return render(request, 'owner/EmployeeEdit.html', {'form': form, 'form1': form1})

def Emp\_Delete(request , id , employee\_id):

    user  = User.objects.filter(id =id )

    user1= Employee\_Enrol.objects.filter(id = employee\_id  )

    user.delete()

    user1.delete()<!DOCTYPE html>

<html>

{%load static%}

<head>

    <!-- Basic -->

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <!-- Mobile Metas -->

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

    <!-- Site Metas -->

    <meta name="keywords" content="" />

    <meta name="description" content="" />

    <meta name="author" content="" />

    <title>Efficient Shelf Monitoring System</title>

    <!-- slider stylesheet -->

    <link rel="stylesheet" type="text/css"

        href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

    <!-- bootstrap core css -->

    <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.css' %}" />

    <!-- fonts style -->

    <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700|Poppins:400,700|Raleway:400,700&display=swap"

        rel="stylesheet" />

    <!-- Custom styles for this template -->

    <link href="{% static 'Css/style.css'%}" rel="stylesheet" />

    <!-- responsive style -->

    <link href="{% static 'Css/responsive.css'%}" rel="stylesheet" />

    <link rel="icon" type="image/png" href="images/icons/favicon.ico" />

    <!--===============================================================================================-->

<body>

    <div class="hero\_area">

        <!-- header section strats -->

        <header class="header\_section">

            <div class="container-fluid">

                <nav class="navbar navbar-expand-lg custom\_nav-container">

                    <a class="navbar-brand" href="">

                        <img src="{% static 'Images/logo.png'%}" alt="" />

                        <span style="color:black">

                            Efficient

                            Shelf

                            Monitoring System

                        </span>

                    </a>

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

                        <div class="custom\_menu-btn">

                            <button onclick="openNav()">

                                <span class="s-1"> </span>

                                <span class="s-2"> </span>

                                <span class="s-3"> </span>

                            </button>

                        </div>

                        <div id="myNav" class="overlay">

                            <div class="overlay-content">

                                <a href="{% url 'OwnerHome'%}">HOME</a>

                                <a href="{%url 'EmpRegister'%}">Emplyoee Enroll</a>

                                <a href="">View Employees</a>

                                <a href="{% url 'home'%}">Logout</a>

                            </div>

                        </div>

                    </div>

                </nav>

            </div>

        </header>

        <!-- end header section -->

        <!-- slider section -->

        <section class=" slider\_section position-relative">

            <div class="side\_heading">

                <h5>

                    Empty

                    Shelf

                    Detection

                </h5>

            </div>

            <div class="container-fluid">

                <div class="row">

                    <div class="col-lg-3 col-md-4 offset-md-1">

                        <div id="carouselExampleIndicators" class="carousel slide " data-ride="carousel">

                            <ol class="carousel-indicators">

                                <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active">

                                    01

                                </li>

                                <li data-target="#carouselExampleIndicators" data-slide-to="1">

                                    02

                                </li>

                                <li data-target="#carouselExampleIndicators" data-slide-to="2">

                                    03

                                </li>

                                <li data-target="#carouselExampleIndicators" data-slide-to="3">

                                    04

                                </li>

                            </ol>

                            <div class="carousel-inner">

                                <div class="carousel-item active">

                                    <div class="img-box b-1">

                                        <img src="{% static 'Images/store.jpeg' %}" alt="" />

                                    </div>

                                </div>

                                <div class="carousel-item">

                                    <div class="img-box b-2">

                                        <img src="{% static 'Images/store.jpeg' %}" alt="" />

                                    </div>

                                </div>

                                <div class="carousel-item">

                                    <div class="img-box b-3">

                                        <img src="{% static 'Images/store.jpeg' %}" alt="" />

                                    </div>

                                </div>

                                <div class="carousel-item">

                                    <div class="img-box b-4">

                                        <img src="{% static 'Images/store.jpeg' %}" alt="" />

                                    </div>

                                </div>

                            </div>

                            <div class="carousel\_btn-box">

                                <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"

                                    data-slide="prev">

                                    <span class="sr-only">Previous</span>

                                </a>

                                <a class="carousel-control-next" href="#carouselExampleIndicators" role="button"

                                    data-slide="next">

                                    <span class="sr-only">Next</span>

                                </a>

                            </div>

                        </div>

                    </div>

                    <div class=" col-md-5 offset-md-1">

                        <div class="detail-box">

                            {% block contents %}

                            <h1>

                                Welcome <br>

                                Owner

                            </h1>

                            <p>

                                In the fast paced world of today where marketing

                                and advertisement is at its peak, an effective inventory

                                management is essential for success. Remote Inventory

                                management is traditionally done through Image processing

                                methods in which the images of the shelves are taken and then

                                processed to find the missing objects

                            </p>

                            {% endblock%}

                            <div class="btn-box">

                                <a href="{% url 'OwnerHome'%}" class="btn-1">

                                    Home

                                </a>

                                <a href="{% url 'EmpRegister' %}" class="btn-1">

                                    Employe Enroll

                                </a><br>

                                <a href="{% url 'Employeeview'%}" class="btn-1">View Emplyees

                                </a>

                                <a href="{%  url 'home' %}" class="btn-1">

                                    Logout

                                </a>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </section>

        <!-- end slider section -->

    </div>

    <section class="container-fluid footer\_section">

        <div class="social\_container">

            <h4>

                Follow on

            </h4>

            <div class="social-box">

                <a href="">

                    <img src="{% static 'Images/fb.png'%}" alt="">

                </a>

                <a href="">

                    <img src="{% static 'Images/twitter.png' %}" alt="">

                </a>

                <a href="">

                    <img src="{% static 'Images/linkedin.png' %}" alt="">

                </a>

                <a href="">

                    <img src="{% static 'Images/youtube.png' %}" alt="">

                </a>

            </div>

        </div>

        <p>

            &copy; 2025 All Rights Reserved. Design by

            <a href=""></a>

        </p>

    </section>

    <!-- footer section -->

    <script type="text/javascript" src="{% static 'js/jquery-3.4.1.min.js' %}"></script>

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

    <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js">

    </script>

    <script>

        function openNav() {

            document.getElementById("myNav").classList.toggle("menu\_width");

            document

                .querySelector(".custom\_menu-btn")

                .classList.toggle("menu\_btn-style");

        }

    </script>

    <!-- owl carousel script -->

    <script type="text/javascript">

        $(".owl-carousel").owlCarousel({

            loop: true,

            margin: 35,

            navText: [],

            autoplay: true,

            autoplayHoverPause: true,

            responsive: {

                0: {

                    items: 1

                },

                600: {

                    items: 2

                },

                1000: {

                    items: 3

                }

            }

        });

    </script>

    <!-- end owl carousel script -->

</body>

</html>

    return redirect('Employeeview')

**OwnerHome.html**