Views.py:

from django.shortcuts import render, redirect

from skillapp.forms import ContactForm, CustomerForm, AdminForm, SellForm, AddFeedbackForm, CommentForm,BuyForm

from skillapp.models import Contacts, Customer, Admin, Sell, AddFeedback, Comment,Buy

# Create your views here.

def index(request):

return render(request,"index.html",{})

def about(request):

return render(request,"about.html",{})

def contact(request):

if request.method == "POST":

print("hii")

form = ContactForm(request.POST)

print("hii")

if form.is\_valid():

form.save()

return render(request, "contact.html", {})

return render(request, "contact.html", {})

#Customer Views

def customer(request):

return render(request, "customer.html", {})

def customer\_loginpage(request):

return render(request, "customer\_login.html", {"msg": ""})

def regpage(request):

return render(request, "customer\_reg.html", {})

def customer\_reg(request):

if request.method == "POST":

form = CustomerForm(request.POST)

if form.is\_valid():

try:

form.save()

except Exception as e:

print(e)

return render(request, "customer\_login.html", {"msg": ""})

def customer\_login(request):

if request.method == "POST":

email = request.POST['email']

password = request.POST['password']

print(email, " ", password)

customer = Customer.objects.filter(email=email, password=password)

if customer.exists():

request.session["email"] = email

return render(request, "customer.html", {"msg": email})

else:

return render(request, "customer\_login.html", {"msg": "email or password not exist"})

return render(request, "customer\_login.html", {})

def unknown(request):

return render(request,"customer.html",{})

def is\_customer\_login(request):

if request.session.\_\_contains\_\_("email"):

return True

else:

return False

def customer\_change(request):

if is\_customer\_login(request):

if request.method == "POST":

email = request.session["email"]

password = request.POST["password"]

newpassword = request.POST["newpassword"]

try:

customer = Customer.objects.get(email=email, password=password)

customer.password = newpassword

customer.save()

msg = "password updated successfully"

return render(request, "customer\_login.html", {"msg": msg})

except:

msg = "inavlid data"

return render(request, "customer\_change.html", {"msg": msg})

return render(request, "customer\_change.html", {})

else:

return render(request, "customer\_change.html", {})

def display(request):

email = request.session['email']

customer = Customer.objects.get(email=email)

print("hello")

return render(request,"customer\_display.html",{"customer":customer})

def delete(request, email):

customer = Customer.objects.get(email=email)

customer.delete()

return redirect("/customer\_reg")

def edit(request, email):

customer = Customer.objects.get(email=email)

return render(request, "edit.html", {"customer": customer})

def update(request):

if request.method == "POST":

customeremail = request.POST["email"]

customer = Customer.objects.get(email=customeremail)

customer = CustomerForm(request.POST, instance=customer)

if customer.is\_valid():

customer.save()

return redirect("/display")

return redirect("/display")

def logout(request):

if request.session.has\_key("email"):

email = request.session["email"]

return render(request, "customer\_login.html", {"email": email})

#Admin Views

def administration(request):

return render(request, "administration.html", {})

def admin\_login(request):

if request.method == "POST":

email = request.POST['email']

password = request.POST['password']

print(email, " ", password)

admin = Admin.objects.all()

if admin.exists():

print("hello")

request.session["email"] = email

return render(request, "administration.html", {"msg": email})

else:

return render(request, "admin\_login.html", {"msg": "email or password not exist"})

return render(request, "admin\_login.html", {"msg": ""})

def adminbro(request):

return render(request,"administration.html",{})

def is\_admini\_login(request):

if request.session.\_\_contains\_\_("email"):

return True

else:

return False

def admini\_change(request):

if is\_admini\_login(request):

if request.method == "POST":

email = request.session["email"]

password = request.POST["password"]

newpassword = request.POST["newpassword"]

try:

admin = Admin.objects.get(email=email, password=password)

admin.password = newpassword

admin.save()

msg = "password updated successfully"

return render(request, "admin\_login.html", {"msg": msg})

except:

msg = "inavlid data"

return render(request, "admini\_change.html", {"msg": msg})

return render(request, "admini\_change.html", {})

else:

return render(request, "admini\_change.html", {})

def admini\_display(request):

customer = Customer.objects.all()

# sell = Sell.objects.all()

print("hello")

return render(request, "admini\_display.html", {"customer": customer})

def admini\_mypurchase(request):

# customer = Customer.objects.all()

sell = Sell.objects.all()

print("hello")

return render(request, "admini\_mypurchase.html", {"sell": sell})

def addfeedback(request):

if request.method == "POST":

addfeedback = AddFeedbackForm(request.POST)

if addfeedback.is\_valid():

addfeedback.save()

return render(request, "addfeedback.html", {"msg": "succes"})

return render(request, "addfeedback.html", {})

def viewfeedback(request):

addfeedback = Contacts.objects.all()

return render(request, "viewfeedback.html", {"addfeedbacks": addfeedback})

def viewfeedback\_delete(request, id):

addfeedback = Contacts.objects.get(id=id)

addfeedback.delete()

addfeedback = Contacts.objects.all()

return redirect("/viewfeedback",{"addfeedbacks":addfeedback})

def sell(request):

email = request.session['email']

customer = Customer.objects.get(email=email)

if request.method == "POST":

print("hii")

print(customer.email)

sell = SellForm(request.POST)

print("hii", sell.errors)

if sell.is\_valid():

sell.save()

sell = Sell.objects.filter(customer\_id=customer.id)

return render(request, "sellview\_my.html", {"msg": "success", "id": customer.id,"sells":sell})

return render(request, "sell.html", { "email": customer.id})

def sellview\_my(request):

email = request.session['email']

customer = Customer.objects.get(email=email)

sell = Sell.objects.filter(customer\_id=customer.id)

print("hii")

return render(request, "sellview\_my.html", {"sells": sell})

def sell\_edit(request, id):

sell = Sell.objects.get(id=id)

return render(request, "sell\_edit.html", {"sell": sell})

def sell\_update(request):

global id

email = request.session['email']

customer = Customer.objects.get(email=email)

if request.method == "POST":

sellid = request.POST["id"]

sell = Sell.objects.get(id=sellid)

sell = SellForm(request.POST, instance=sell)

print("erroes",sell.errors)

if sell.is\_valid():

sell.save()

sell = Sell.objects.filter(customer\_id=customer.id)

return redirect("/sellview\_my",{"sells":sell})

return redirect("/sellview\_my")

def sell\_delete(request, id):

sell = Sell.objects.get(id=id)

sell.delete()

sell = Sell.objects.filter(customer\_id=customer.id)

return redirect("/sell",{"sells":sell})

def sellview\_all(request):

email=request.session['email']

customer=Customer.objects.get(email=email)

sell = Sell.objects.exclude(customer\_id=customer.id)

return render(request, "sellview\_all.html", {"sells": sell})

def sell\_buy(request,id):

email = request.session['email']

customer= Customer.objects.get(email=email)

project=Sell.objects.get(id=id)

if request.method == "POST":

buy= BuyForm(request.POST)

if buy.is\_valid():

buy.save()

bookings = Buy.objects.filter(customer\_id=customer.id)

return render(request,"mypurchase.html",{"msg":"success","id":project.id,"cost":project.cost,"description":project.description,"customer":customer.id,"buys":bookings})

return render(request, "sell\_buy.html", {"id": project.id, "cost": project.cost, "description":project.description,"customer":customer.id})

def sell\_likes(request,id):

sell = Sell.objects.get(id=id)

sell.likes = sell.likes+1

sell.save()

sell = Sell.objects.all()

return render(request, "sellview\_all.html", {"sell": sell})

def sell\_dislikes(request,id):

sell = Sell.objects.get(id=id)

sell.dislikes = sell.dislikes+1

sell.save()

sell = Sell.objects.all()

return render(request, "sellview\_all.html", {"sell": sell})

def my\_purchase(request):

email = request.session['email']

customer=Customer.objects.get(email=email)

print("hello1")

bookings = Buy.objects.filter(customer\_id=customer.id)

print("hello")

return render(request,"mypurchase.html",{"buys":bookings})

def comment(request,id):

email = request.session['email']

customer = Customer.objects.get(email=email)

seller = Sell.objects.get(id=id)

if request.method == "POST":

print("hii")

print(customer.email)

comment = CommentForm(request.POST)

print("hii", comment.errors)

if comment.is\_valid():

comment.save()

return render(request, "comment.html", {"msg": "success","id":seller.id, "customer": customer.id})

return render(request, "comment.html", {"id": seller.id,"email": customer.id})

def comment\_update(request):

if request.method == "POST":

commentid = request.POST["id"]

comment = Customer.objects.get(id=commentid)

comment = CustomerForm(request.POST, instance=comment)

if comment.is\_valid():

comment.save()

return redirect("/viewcomment")

return redirect("/viewcomment")

def viewcomment(request):

comment = Comment.objects.all()

return render(request, "viewcomment.html", {"comments": comment})

def admini\_logout(request):

if request.session.has\_key("email"):

email = request.session["email"]

return render(request,"admin\_login.html",{"email":email})

def view\_reviews(request,id):

place=Sell.objects.get(id=id)

comment=Comment.objects.filter(sell\_id=place.id)

return render(request,"customer\_view\_comment.html",{"comments":comment})

def admin\_view\_projects(request):

comment = Sell.objects.all()

return render(request, "admin\_view\_projects.html", {"sells": comment})

def customer\_del(request,id):

customer=Customer.objects.get(id=id)

customer.delete()

customer=Customer.objects.all()

return render(request,"admini\_display.html",{"customers":customer})

def project\_del(request,id):

sells=Sell.objects.get(id=id)

sells.delete()

sell=Sell.objects.all()

return render(request,"admin\_view\_projects.html",{"sells":sell})

def comment\_del(request,id):

sells=Comment.objects.get(id=id)

sells.delete()

sell=Comment.objects.all()

return render(request,"viewcomment.html",{"comments":sell})

def customer\_comment\_del(request,id):

sells=Comment.objects.get(id=id)

sells.delete()

comment=Comment.objects.filter(sell\_id=sells.id)

return render(request,"customer\_view\_comment.html",{"comments":comment})

Models.py:

from django.db import models

# Create your models here.

class Contacts(models.Model):

message = models.TextField()

name = models.CharField(max\_length=100)

email = models.EmailField()

subject = models.CharField(max\_length=100)

class Meta:

db\_table="contacts"

class Customer(models.Model):

fullname = models.CharField(max\_length=100)

email = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

mobile = models.BigIntegerField()

city = models.CharField(max\_length=100)

class Meta:

db\_table="customer"

class Admin(models.Model):

email = models.CharField(max\_length=100)

password = models.CharField(max\_length=100)

class Meta:

db\_table="admin"

class Sell(models.Model):

customer = models.ForeignKey(Customer, on\_delete=models.CASCADE)

title = models.CharField(max\_length=100)

description = models.TextField()

cost = models.BigIntegerField()

technologies = models.CharField(max\_length=100)

domain = models.CharField(max\_length=100)

datetime = models.DateField()

likes = models.IntegerField(default=0)

dislikes = models.IntegerField(default=0)

class Meta:

db\_table="sell"

class Buy(models.Model):

customer = models.ForeignKey(Customer, on\_delete=models.CASCADE)

sell = models.ForeignKey(Sell, on\_delete=models.CASCADE)

description = models.TextField()

cost = models.BigIntegerField()

class AddFeedback(models.Model):

Name = models.CharField(max\_length=100)

Email = models.CharField(max\_length=100)

Message = models.TextField()

class Meta:

db\_table = "addfeedback"

class Comment(models.Model):

sell = models.ForeignKey(Sell, on\_delete=models.CASCADE)

customer = models.ForeignKey(Customer, on\_delete=models.CASCADE)

comment = models.TextField()

rating = models.BigIntegerField()

class Meta:

db\_table = "comment"

Urls.py:

"""skill URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/4.0/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path

from skillapp import views as v

urlpatterns = [

path('admin/', admin.site.urls),

path('', v.index, name="index"),

path('about/', v.about, name="about"),

path('contact/', v.contact, name="contact"),

#Customer Urls

path('customer/', v.customer, name="customer"),

path('customer\_loginpage/', v.customer\_loginpage, name="customer\_loginpage"),

path('regpage/', v.regpage, name="regpage"),

path('customer\_reg/', v.customer\_reg, name="customer\_reg"),

path('customer\_login/', v.customer\_login, name="customer\_login"),

path('customer\_change/', v.customer\_change, name="customer\_change"),

path('display/', v.display, name="display"),

path('edit/<str:email>', v.edit, name="edit"),

path('update/', v.update, name="update"),

path('delete/<str:email>', v.delete, name="delete"),

path('logout/', v.logout, name="logout"),

path('unknown/', v.unknown, name="unknown"),

path('adminbro/', v.adminbro, name="adminbro"),

path('admin\_view\_projects/', v.admin\_view\_projects, name="admin\_view\_projects"),

#Admin Urls

path('administration/', v.administration, name="administration"),

path('admin\_login/', v.admin\_login, name="admin\_login"),

path('admini\_change/', v.admini\_change, name="admini\_change"),

path('admini\_display/', v.admini\_display, name="admini\_display"),

path('admini\_mypurchase/', v.admini\_mypurchase, name="admini\_mypurchase"),

path('addfeedback/', v.addfeedback, name="addfeedback"),

path('viewfeedback/', v.viewfeedback, name="viewfeedback"),

path('viewfeedback\_delete/<int:id>', v.viewfeedback\_delete, name="viewfeedback\_delete"),

path('admini\_logout/', v.admini\_logout, name="admini\_logout"),

path('sell/', v.sell, name="sell"),

path('sellview\_my/', v.sellview\_my, name="sellview\_my"),

path('sell\_edit/<int:id>', v.sell\_edit, name="sell\_edit"),

path('sell\_update/', v.sell\_update, name="sell\_update"),

path('sell\_delete/<int:id>', v.sell\_delete, name="sell\_delete"),

path('customer\_del/<int:id>', v.customer\_del, name="customer\_del"),

path('sellview\_all/', v.sellview\_all, name="sellview\_all"),

path('sell\_buy/<int:id>', v.sell\_buy, name="sell\_buy"),

path('project\_del/<int:id>', v.project\_del, name="project\_del"),

path('sell\_likes/<int:id>', v.sell\_likes, name="sell\_likes"),

path('sell\_dislikes/<int:id>', v.sell\_dislikes, name="sell\_dislikes"),

path('my\_purchase/', v.my\_purchase, name="my\_purchase"),

path('comment/<int:id>', v.comment, name="comment"),

path('view\_reviews/<int:id>', v.view\_reviews, name="view\_reviews"),

path('comment\_del/<int:id>', v.comment\_del, name="comment\_del"),

path('customer\_comment\_del/<int:id>', v.customer\_comment\_del, name="customer\_comment\_del"),

path('comment\_update/', v.comment\_update, name="comment\_update"),

path('viewcomment/', v.viewcomment, name="viewcomment"),

]

Templates:

{% load static %}

{% include 'header.html' %}

<main>

<!-- slider Area Start-->

<div class="slider-area ">

<!-- Mobile Menu -->

<div class="single-slider slider-height2 d-flex align-items-center" data-background="{% static 'assets/img/hero/contact\_hero.jpg' %}">

<div class="container">

<div class="row">

<div class="col-xl-12">

<div class="hero-cap text-center">

<h2>Customer\_register</h2>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- slider Area End-->

<form method ="POST" class="post-form" action="/customer\_reg/">

{% csrf\_token %}

<section class="h-100">

<div class="container h-100">

<div class="row justify-content-sm-center h-100">

<div class="col-xxl-4 col-xl-5 col-lg-5 col-md-7 col-sm-9">

<div class="text-center my-5">

<!-- <img src="https://getbootstrap.com/docs/5.0/assets/brand/bootstrap-logo.svg" alt="logo" width="100">-->

</div>

<div class="card shadow-lg">

<div class="card-body p-5">

<h1 class="fs-4 card-title fw-bold mb-4">Register Here</h1>

<form method="POST" class="needs-validation" novalidate="" autocomplete="off">

<div class="mb-3">

<label class="mb-2 text-muted" for="fullname">Fullname</label>

<input id="fullname" type="text" class="form-control" name="fullname" value="" placeholder="FullName" required autofocus>

<div class="invalid-feedback">

Email is invalid

</div>

</div>

<div class="mb-3">

<label class="mb-2 text-muted" for="email">Email</label>

<input id="email" type="email" class="form-control" name="email" value="" placeholder="Email" required autofocus>

<div class="invalid-feedback">

Email is invalid

</div>

</div>

<div class="mb-3">

<div class="mb-2 w-100">

<label class="text-muted" for="password">Password</label>

</div>

<input id="password" type="password" class="form-control" name="password" placeholder="Password" required>

</div>

<div class="mb-3">

<div class="mb-2 w-100">

<label class="text" for="mobile">Mobile</label>

</div>

<input id="mobile" type="number" class="form-control" name="mobile" required placeholder="Mobile">

</div>

<div class="mb-3">

<div class="mb-2 w-100">

<label class="text" for="city">City</label>

</div>

<select name="city">

<option value="Hyd">Hyd</option>

<option value="Pune">Pune</option>

<option value="Chennai">Chennai</option>

<option value="Mumbai">Mumbai</option>

</select>

</div><br>

<br>

<button type="submit" class="btn btn-primary ms-auto">Register</button><br>

</form>

</div>

</div>

</div>

</div>

</div>

</section>

</form>

</main>

{% include 'footer.html' %}

{% load static %}

{% include 'customer\_header.html' %}

<main>

<!-- slider Area Start-->

<div class="slider-area ">

<!-- Mobile Menu -->

<div class="single-slider slider-height2 d-flex align-items-center" data-background="{% static 'assets/img/hero/contact\_hero.jpg' %}">

<div class="container">

<div class="row">

<div class="col-xl-12">

<div class="hero-cap text-center">

<h2>Details</h2>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- slider Area End--><br>

<body>

<table align="center" border="2" cellpadding="10" cellspacing="0" width="80%">

<!-- <table id="employeeList" class="table table-bordered table-striped">-->

<thead>

<tr>

<th>id</th>

<th>fullname</th>

<th>email</th>

<th>mobile</th>

<th>city</th>

</tr>

</thead>

<tbody>

<tr>

<td>{{customer.id}}</td>

<td>{{customer.fullname}}</td>

<td>{{customer.email}}</td>

<td>{{customer.mobile}}</td>

<td>{{customer.city}}</td>

<!-- <td><a href="/edit/{{customer.email}}">edit</a></td>-->

<td><a href="/edit/{{customer.email}}" class="btn-primary">edit</a> </td>

<td><a href="/delete/{{customer.email}}" class="btn-danger" onclick="return confirm('really want delete?');">delete</a></td>

</tr>

</tbody>

</table>

<footer>

<!-- Footer Start-->

<!-- Footer End-->

</footer>

<!-- JS here -->

<!-- All JS Custom Plugins Link Here here -->

<script src="{% static './assets/js/vendor/modernizr-3.5.0.min.js' %}"></script>

<!-- Jquery, Popper, Bootstrap -->

<script src="{% static './assets/js/vendor/jquery-1.12.4.min.js' %}"></script>

<script src="{% static './assets/js/popper.min.js' %}"></script>

<script src="{% static './assets/js/bootstrap.min.js' %}"></script>

<!-- Jquery Mobile Menu -->

<script src="{% static './assets/js/jquery.slicknav.min.js' %}"></script>

<!-- Jquery Slick , Owl-Carousel Plugins -->

<script src="{% static './assets/js/owl.carousel.min.js' %}"></script>

<script src="{% static './assets/js/slick.min.js' %}"></script>

<!-- One Page, Animated-HeadLin -->

<script src="{% static './assets/js/wow.min.js' %}"></script>

<script src="{% static './assets/js/animated.headline.js' %}"></script>

<script src="{% static './assets/js/jquery.magnific-popup.js' %}"></script>

<!-- Scrollup, nice-select, sticky -->

<script src="{% static './assets/js/jquery.scrollUp.min.js' %}"></script>

<script src="{% static './assets/js/jquery.nice-select.min.js' %}"></script>

<script src="{% static './assets/js/jquery.sticky.js' %}"></script>

<!-- contact js -->

<script src="{% static './assets/js/contact.js' %}"></script>

<script src="{% static './assets/js/jquery.form.js' %}"></script>

<script src="{% static './assets/js/jquery.validate.min.js' %}"></script>

<script src="{% static './assets/js/mail-script.js' %}"></script>

<script src="{% static './assets/js/jquery.ajaxchimp.min.js' %}"></script>

<!-- Jquery Plugins, main Jquery -->

<script src="{% static './assets/js/plugins.js' %}"></script>

<script src="{% static './assets/js/main.js' %}"></script>

</main>

</body>

{% include 'footer.html' %}

</html>