Models.py

from django.db import models

# Create your models here.

class Contact(models.Model):

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

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

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

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

Message = models.TextField()

class Meta:

db\_table = "contact"

class Customer(models.Model):

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

email = models.CharField(max\_length=100, primary\_key=True)

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

mobile = models.BigIntegerField()

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

class Meta:

db\_table = "customer"

class Mechanic(models.Model):

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

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

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

mobile = models.BigIntegerField()

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

address = models.TextField()

email = models.CharField(max\_length=100, primary\_key=True)

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

class Meta:

db\_table = "mechanic"

class Admin(models.Model):

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

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

class Meta:

db\_table = "admin"

class Book(models.Model):

mechanic = models.ForeignKey(Mechanic, on\_delete=models.CASCADE)

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

description = models.TextField()

date = models.DateField()

time = models.TimeField()

slot\_status = models.CharField(max\_length=150, default='pending')

class Meta:

db\_table = "book"

class Review(models.Model):

mechanic = models.ForeignKey(Mechanic, on\_delete=models.CASCADE)

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

rating = models.BigIntegerField()

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

class Meta:

db\_table = "review"

views.py:

from django.shortcuts import render,redirect

from logapp.forms import ContactForms, CustomerForms, MechanicForms, AdminForms, BookForms, ReviewForms

from logapp.models import Contact, Customer, Mechanic, Admin, Book, Review

# 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 = ContactForms(request.POST)

print("hii")

if form.is\_valid():

form.save()

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

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

def blog(request):

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

def industries(request):

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

def services(request):

return render(request, "services.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, "reg.html", {})

def reg(request):

if request.method == "POST":

form = CustomerForms(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", {"msg": ""})

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 customer\_display(request):

email = request.session["email"]

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

print("hello")

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

def customer\_delete(request, email):

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

customer.delete()

return redirect("/reg")

def customer\_edit(request, email):

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

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

def customer\_update(request):

if request.method == "POST":

customerid = request.POST["email"]

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

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

print(customer.errors)

if customer.is\_valid():

print(customer.errors)

customer.save()

return redirect("/customer\_display")

return redirect("/customer\_display")

def viewmechanic(request):

mechanic = Mechanic.objects.all()

print("hello")

return render(request, "viewmechanic.html", {"mechanic": mechanic})

def book(request, email):

email1 = request.session['email']

customer = Customer.objects.get(email=email1)

mechanic = Mechanic.objects.get(email=email)

if request.method == "POST":

print("hii")

print(customer.email)

book = BookForms(request.POST)

print("hii", book.errors)

if book.is\_valid():

book.save()

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

return render(request, "book.html", {"mail": mechanic.email, "customer": email1})

def book\_update(request):

if request.method == "POST":

bookemail = request.POST["email"]

book = Book.objects.get(email=bookemail)

book = BookForms(request.POST, instance=book)

if book.is\_valid():

book.save()

return redirect("/viewmechanic")

return redirect("/viewmechanic")

def review(request,email):

email1 = request.session['email']

customer = Customer.objects.get(email=email1)

mechanic = Mechanic.objects.get(email=email)

if request.method == "POST":

print("hii")

print(customer.email)

review = ReviewForms(request.POST)

print("hii", review.errors)

if review.is\_valid():

review.save()

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

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

def review\_update(request):

if request.method == "POST":

reviewemail = request.POST["email"]

review = Review.objects.get(email=reviewemail)

review = ReviewForms(request.POST, instance=review)

if review.is\_valid():

review.save()

return redirect("/viewmechanic")

return redirect("/viewmechanic")

def customer\_logout(request):

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

email = request.session["email"]

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

def mechanic\_booked\_slots(request):

email = request.session["email"]

customer = Book.objects.filter(customer\_id=email)

return render(request, "my\_slots.html", {"customers": customer})

#mechanic views

def mechanic(request):

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

def mechanic\_loginpage(request):

return render(request, "mechanic\_loginpage.html", {"msg": ""})

def mechanic\_regpage(request):

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

def mechanic\_reg(request):

if request.method == "POST":

form = MechanicForms(request.POST)

if form.is\_valid():

try:

form.save()

except Exception as e:

print(e)

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

def mechanic\_login(request):

if request.method == "POST":

email = request.POST['email']

password = request.POST['password']

print(email, " ", password)

mechanic = Mechanic.objects.filter(email=email, password=password)

if mechanic.exists():

request.session["email"] = email

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

else:

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

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

def is\_mechanic\_login(request):

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

return True

else:

return False

def mechanic\_change(request):

if is\_mechanic\_login(request):

if request.method == "POST":

email = request.session["email"]

password = request.POST["password"]

newpassword = request.POST["newpassword"]

try:

mechanic = Mechanic.objects.get(email=email, password=password)

mechanic.password = newpassword

mechanic.save()

msg = "password updated successfully"

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

except:

msg = "inavlid data"

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

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

else:

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

def mechanic\_display(request):

email = request.session["email"]

mechanic = Mechanic.objects.get(email=email)

print("hello")

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

def mechanic\_delete(request, email):

mechanic = Mechanic.objects.get(email=email)

mechanic.delete()

return redirect("/reg")

def mechanic\_edit(request, email):

mechanic = Mechanic.objects.get(email=email)

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

def mechanic\_update(request):

if request.method == "POST":

mechanicid = request.POST["id"]

mechanic = Customer.objects.get(id=mechanicid)

mechanic = CustomerForms(request.POST, instance=mechanic)

if mechanic.is\_valid():

mechanic.save()

return redirect("/mechanic\_display")

return redirect("/mechanic\_display")

def mechanic\_logout(request):

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

email = request.session["email"]

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

def booked\_slots(request):

email = request.session["email"]

booked = Book.objects.filter(mechanic\_id=email)

return render(request, "booked\_slots.html", {"booked": booked})

def slot\_approve(request, slot\_id):

approve = Book.objects.get(id=slot\_id)

approve.slot\_status = 'approved'

approve.save()

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

def slot\_reject(request, slot\_id):

reject = Book.objects.get(id=slot\_id)

reject.job\_status = 'rejected'

reject.save()

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

#admin views

def administration(request):

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

def admin\_login(request):

if request.method == "POST":

username = request.POST['username']

password = request.POST['password']

print(username, " ", password)

admin = Admin.objects.all()

if admin.exists():

print("hello")

request.session["username"] = username

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

else:

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

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

def view\_customer(request):

customer = Customer.objects.all()

print("hello")

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

def view\_mechanic(request):

mechanic = Mechanic.objects.all()

print("hello")

return render(request, "view\_mechanic.html", {"mechanic": mechanic})

def admin\_logout(request):

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

username = request.session["username"]

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

def view\_bookings(request):

book = Book.objects.all()

return render(request, "view\_bookings.html", {"book": book})

def admin\_viewmechanic(request):

mechanic = Mechanic.objects.all()

print("hello")

return render(request, "view\_mechanic.html", {"mechanic": mechanic})

def admin\_customer\_delete(request, email):

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

customer.delete()

return redirect("/administration")

def admin\_mechanic\_delete(request, email):

mechanic = Mechanic.objects.get(email=email)

mechanic.delete()

return redirect("/administration")

def view\_reviews(request):

review = Review.objects.all()

return render(request, "view\_reviews.html", {"review": review})

urls.py:

"""logistics 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 logapp 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"),

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

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

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

#Customer urls

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

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

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

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

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

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

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

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

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

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

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

#Mechanic urls

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

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

path('mechanic\_regpage/', v.mechanic\_regpage, name="mechanic\_regpage"),

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

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

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

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

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

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

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

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

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

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

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

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

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

path('booked\_slots/', v.booked\_slots, name="booked\_slots"),

path('slot\_approve/<int:slot\_id>', v.slot\_approve, name="slot\_approve"),

path('slot\_reject/<int:slot\_id>', v.slot\_reject, name="slot\_reject"),

path('mechanic\_booked\_slots/', v.mechanic\_booked\_slots, name="mechanic\_booked\_slots"),

#Admin urls

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

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

path('view\_customer/', v.view\_customer, name="view\_customer"),

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

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

path('view\_bookings/', v.view\_bookings, name="view\_bookings"),

path('view\_reviews/', v.view\_reviews, name="view\_reviews"),

path('admin\_customer\_delete/<str:email>', v.admin\_customer\_delete, name="admin\_customer\_delete"),

path('admin\_mechanic\_delete/<str:email>', v.admin\_mechanic\_delete, name="admin\_mechanic\_delete"),

]

Templates:

Customer registration

Reg.html:

{% load static %}

{% include 'header.html' %}

<div class="site-blocks-cover inner-page-cover overlay" style="background-image: url({% static 'images/hero\_bg\_1.jpg' %});" data-aos="fade" data-stellar-background-ratio="0.5">

<div class="container">

<div class="row align-items-center justify-content-center text-center">

<div class="col-md-8" data-aos="fade-up" data-aos-delay="400">

<h1 class="text-white font-weight-light text-uppercase font-weight-bold">Customer register</h1>

<p class="breadcrumb-custom"><a href="#">Home</a> <span class="mx-2">&gt;</span> <span>Customer</span></p>

</div>

</div>

</div>

</div>

<form method ="POST" class="post-form" action="/reg/">

{% csrf\_token %}

<div class="container h-100">

<h3 class="fs-4 card-title fw-bold mb-4">Register Here</h3>

<div class="mb-3">

<label class="mb-2 text-muted" for="name">Name</label>

<input id="name" type="text" class="form-control" name="name" required>

</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" required>

</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" 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>

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

</select>

</div>

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

</div>

</form>

{% include 'footer.html' %}

Customer\_login.html:

{% load static %}

{% include 'header.html' %}

<div class="site-blocks-cover inner-page-cover overlay" style="background-image: url({% static 'images/hero\_bg\_1.jpg' %});" data-aos="fade" data-stellar-background-ratio="0.5">

<div class="container">

<div class="row align-items-center justify-content-center text-center">

<div class="col-md-8" data-aos="fade-up" data-aos-delay="400">

<h1 class="text-white font-weight-light text-uppercase font-weight-bold">Customer Login</h1>

</div>

</div>

</div>

</div>

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

{% csrf\_token %}

<div class="container h-100"><br>

<h3 class="fs-4 card-title fw-bold mb-4">Login Here</h3>

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

<label class="col-sm-2 col-form-label" for="Email">Email</label>

<input id="Email" type="email" name="email" required>

</div>

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

<label class="col-sm-2 col-form-label" for="Password">Password</label>

<input id="Password" type="Password" name="password" required>

</div>

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

U are not exist? <a href="/regpage/">Register</a>

</div>

</form>

<!--{{msg}}-->

{% include 'footer.html' %}