Add.html: {% extends "recipt/layout.html" %}

{% block body %}

<div>

    <form action="{% if is\_editing %}{% url 'recipt:edit\_product' id %}{% else %}{% url 'recipt:add' %}{% endif %}" method="post">

        {% csrf\_token %}

        <!-- Dynamic heading based on whether you are adding or editing an item -->

        <h1>

            {% if is\_editing %}

                Please edit details

            {% else %}

                Please add details

            {% endif %}

        </h1>

        {% if not request.session.customer\_name %}

            <!-- Show the customer name field only if it is not already set -->

            {% if form\_customer %}

                {{ form\_customer.as\_p }}

            {% endif %}

        {% endif %}

        {% if form %}

            {{ form.as\_p }}

        {% endif %}

        <input type="submit" style="width: fit-content; height: fit-content; background-color: brown; color: white; padding: 5px; margin-top: 7px; border-radius: 2px; border: 1px solid rgb(0, 0, 0);">

    </form>

    <a href="{% url 'recipt:index' %}">

        <button style="width: fit-content; height: fit-content; background-color: brown; color: white; padding: 5px; margin-top: -25px; border-radius: 2px; border: 1px solid rgb(0, 0, 0); float: right;">

            Your receipt

        </button>

    </a>

</div>

<script>

    document.addEventListener("DOMContentLoaded", function() {

        // Focus on the product name field if the customer name field is hidden

        const productNameField = document.getElementById("id\_name");

        const customerNameField = document.getElementById("id\_customer\_name");

        if (customerNameField && !customerNameField.closest("form").querySelector(".hidden")) {

            // Focus on the customer name field if it's visible

            customerNameField.focus();

        } else if (productNameField) {

            // Focus on the product name field otherwise

            productNameField.focus();

        }

    });

</script>

{% endblock %}

Edit\_customer.html: {% extends "recipt/layout.html" %}

{% block body %}

<div>

    <form action="{% url 'recipt:edit\_customer' customer\_name customer\_email %}" method="post">

        {% csrf\_token %}

        <h1>Edit Customer Details</h1>

        {% if customer\_form %}

        {{ customer\_form }}

        {% endif %}

        <input type="submit" value="Update Details" style="background-color: brown; color: white; padding: 5px; margin-top: 30px;">

    </form>

    <a href="{% url 'recipt:index' %}">

        <button style="background-color: brown; color: white; padding: 5px; float: right; margin-top: -25px;">

            Back to Receipt

        </button>

    </a>

</div>

{% endblock %}

Index.html: {% extends "recipt/layout.html" %}

{% block body %}

<div>

    <div id="receipt-content">

        <div style="display: flex; justify-content: space-between;">

            <div>

                {% if not customer\_name %}

                    {% if request.user.first\_name and request.user.last\_name %}

                        <h3>Here is your receipt by {{ request.user.first\_name }} {{ request.user.last\_name }}</h3>

                    {% else %}

                        <h3>Here is your receipt by {{ request.user.username }}</h3>

                    {% endif %}

                {% else %}

                    {% if request.user.first\_name and request.user.last\_name %}

                        <h3>Here is {{ customer\_name }}'s receipt by {{ request.user.first\_name }} {{ request.user.last\_name }}

                            {% if customer\_name %}

                                <a href="{% url 'recipt:edit\_customer' customer\_name customer\_email %}" class="edit-button">

                                    <img src="https://th.bing.com/th/id/OIP.P6trHImFdjGgSXQBU7\_sagHaHa?rs=1&pid=ImgDetMain" alt="EDIT"

                                         style="width: 15px; height: 20px; margin-left: 10px;">

                                </a>

                            {% endif %}

                        </h3>

                    {% else %}

                        <h3>Here is {{ customer\_name }}'s receipt by {{ request.user.username }}

                            {% if customer\_name %}

                                <a href="{% url 'recipt:edit\_customer' customer\_name customer\_email %}" class="edit-button">

                                    <img src="https://th.bing.com/th/id/OIP.P6trHImFdjGgSXQBU7\_sagHaHa?rs=1&pid=ImgDetMain" alt="EDIT"

                                         style="width: 15px; height: 20px; margin-left: 10px;">

                                </a>

                            {% endif %}

                        </h3>

                    {% endif %}

                {% endif %}

            </div>

            <h6 class="receipt-date">{{ now }}</h6>

        </div>

        <div id="index\_outer\_div" class="receipt-details">

            <div>

                <ol>

                    <br>

                    {% if products %}

                        <h3 style="margin-top: 8px;">Product Name</h3>

                        {% for product in products %}

                            <li style="margin-left: 15px;">{{ product.0 }}</li>

                            <br>

                        {% endfor %}

                    {% else %}

                        <h1>No items added</h1>

                    {% endif %}

                </ol>

            </div>

            <div class="receipt-section">

                <br>

                {% if products %}

                    <h3>Quantity</h3>

                    {% for product in products %}

                        {{ product.1 }}

                        <br><br>

                    {% endfor %}

                {% endif %}

            </div>

            <div class="receipt-section">

                <br>

                {% if products %}

                    <h3>Price</h3>

                    {% for product in products %}

                        {{ product.2 }}

                        <br><br>

                    {% endfor %}

                {% endif %}

            </div>

            <div class="receipt-section">

                <br>

                {% if products %}

                    <h3>Total for Each Item</h3>

                    {% for product in products %}

                        {{ product.3 }}

                        <br><br>

                    {% endfor %}

                {% endif %}

            </div>

            <div class="receipt-actions">

                <br><br>

                {% for id in range\_5 %}

                    <a href="{% url 'recipt:dele' id %}" class="delete-button">

                        <img src="https://icon-library.com/images/icon-remove/icon-remove-18.jpg" alt="DEL"

                             style="width: 15px; height: 20px; margin-bottom: 10px; margin-right: 10px;">

                    </a>

                    <a href="{% url 'recipt:edit\_product' id %}" class="edit-button">

                        <img src="https://th.bing.com/th/id/OIP.P6trHImFdjGgSXQBU7\_sagHaHa?rs=1&pid=ImgDetMain" alt="EDIT"

                             style="width: 15px; height: 20px; margin-bottom: 10px; margin-right: 10px;">

                    </a>

                    <br>

                {% endfor %}

            </div>

        </div>

        <div>

            <h4>Total Price: {{ total\_price }}</h4>

        </div>

        <br>

    </div>

        <a href="{% url 'recipt:new\_receipt' %}" class="action-button">New Receipt</a>

        <a href="{% url 'recipt:sendmail' %}" class="action-button">E.recipt</a>

        <a href="{% url 'recipt:add' %}" class="action-button">Add New Item</a>

        <a href="{% url 'recipt:save\_customer' %}" class="action-button">Save E.recipt</a>

        <button onclick="printReceipt()" class="action-button print-button">Print Receipt</button>

    </div>

    <script>

        function printReceipt() {

            // Hide edit and delete buttons for printing

            document.querySelectorAll('.delete-button').forEach(el => el.style.display = 'none');

            document.querySelectorAll('.edit-button').forEach(el => el.style.display = 'none');

            // Print the page

            window.print();

            // Restore the visibility of the hidden elements

            document.querySelectorAll('.delete-button').forEach(el => el.style.display = 'inline');

            document.querySelectorAll('.edit-button').forEach(el => el.style.display = 'inline');

        }

    </script>

</div>

{% endblock %}

Layout.html: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    {% load static %}

    <link rel="stylesheet" type="text/css" href="{% static 'recipt\_css/style.css' %}">

    <link rel="icon" type="image/x-icon"

          href="https://th.bing.com/th/id/OIG1.y\_2amRAytZhrAjPsZA6Y?w=173&h=173&c=6&r=0&o=5&pid=ImgGn"

          alt="BUGS">

    <title>B U G S</title>

</head>

<body>

    <header>

        <h1>Bhatti Union General Store</h1>

        <a href="{% url 'recipt:logout' %}">

            <button type="button" class="logout-button"style="float: right;width: 100px;height: 30px; color: brown; "><strong>Log out</strong></button>

        </a>

    </header>

    <div id="outerContainer">

        <div id="container">

            {% block body %}

            {% endblock %}

        </div>

    </div>

</body>

</html>

LOGIN.HTML: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    {% load static %}

    <link rel="stylesheet" type="text/css" href="{% static 'recipt\_css/style.css' %}" />

    <link rel="icon" type="image/x-icon"

        href="https://th.bing.com/th/id/OIG1.y\_2amRAytZhrAjPsZA6Y?w=173&h=173&c=6&r=0&o=5&pid=ImgGn"

        alt="https://th.bing.com/th/id/OIG1.y\_2amRAytZhrAjPsZA6Y?w=173&h=173&c=6&r=0&o=5&pid=ImgGn">

    <title>B U G S</title>

</head>

<body>

    <header>

        <h1>Bhatti Union General Store</h1>

    </header>

    <br>

    <div id="outerContainer" >

        <div id="container">

            {% if message %}

            <p>{{ message }}</p>

            {% endif %}

            <form method="POST">

                {% csrf\_token %}

                <label for="username">Username:</label>

                <input type="text" name="username" id="username" value="{{ username }}">

                <br>

                <br>

                <label for="password">Password:</label>

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

                <br>

                <br>

                <br>

                <button style="margin-left: 100px; padding: 10px;" type="submit"><strong>Login</strong></button>

            </form>

    </div>

    </div>

</body>

</html>

Model.html: from django.db import models

class Receipt(models.Model):

    customer\_name = models.CharField(max\_length=255)

    products = models.JSONField()  # Store products as a JSON object

    total\_price = models.DecimalField(max\_digits=10, decimal\_places=2)

    date = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return f"Receipt for {self.customer\_name} on {self.date}"

views.html: from django.contrib.auth import authenticate, login, logout

from django import forms

from django.shortcuts import render, redirect

from django.http import HttpResponseRedirect , HttpResponse

from django.urls import reverse

from datetime import datetime

from .sendmail import viewsdata

# from .models import insert\_customer\_data

class NewDataForm(forms.Form):

    def for\_edit\_product(self, nam, pric, quant, \*args, \*\*kwargs):

        super().\_\_init\_\_(\*args, \*\*kwargs)

        self.fields['name'].initial = nam

        self.fields['price'].initial = pric

        self.fields['quantity'].initial = quant

    name = forms.CharField(

        widget=forms.TextInput(attrs={

            'id': 'id\_name',

            'placeholder': 'Enter product name',

            'class': 'form-control',

            'style': 'width: 100%; padding: 10px; margin-bottom: 10px;'

        })

    )

    price = forms.IntegerField(

        widget=forms.NumberInput(attrs={

            'placeholder': 'Enter product price',

            'class': 'form-control',

            'style': 'width: 100%; padding: 10px; margin-bottom: 10px;'

        })

    )

    quantity = forms.IntegerField(

        widget=forms.NumberInput(attrs={

            'placeholder': 'Enter product quantity',

            'class': 'form-control',

            'style': 'width: 100%; padding: 10px; margin-bottom: 10px;'

        })

    )

class CustomerForm(forms.Form):

    def for\_edit\_customer(self, customer\_name, customer\_email, \*args, \*\*kwargs):

        super().\_\_init\_\_(\*args, \*\*kwargs)

        self.fields['customer\_name'].initial = customer\_name

        self.fields['customer\_email'].initial = customer\_email

    customer\_name = forms.CharField(

        required=False,

        widget=forms.TextInput(attrs={

            'placeholder': 'Customer name',

            'class': 'form-control',

            'style': 'width: 100%; padding: 10px; margin-bottom: 10px;',

        })

    )

    customer\_email = forms.EmailField(

        required=False,

        widget=forms.EmailInput(attrs={

            'placeholder': 'Customer name',

            'class': 'form-control',

            'style': 'width: 100%; padding: 10px; margin-bottom: 10px;',

        })

    )

def index(request):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    if "products" not in request.session:

        request.session["products"] = []  # 2D list to store [name, quantity, price, quantity\_price]

    if "total\_price" not in request.session:

        request.session["total\_price"] = 0

    if "customer\_name" not in request.session:

        request.session["customer\_name"] = None

    if "customer\_email" not in request.session:

        request.session["customer\_email"] = None

    return render(request, 'recipt/index.html', {

        "products": request.session["products"],

        "total\_price": request.session["total\_price"],

        "customer\_name": request.session["customer\_name"],

        "customer\_email": request.session["customer\_email"],

        'range\_5': range(len(request.session["products"])),

        'now': datetime.now()

    })

def sendmail(request):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    user\_data = {

        'username': request.user.username,

        'first\_name': request.user.first\_name,

        'last\_name': request.user.last\_name,

        'products': request.session.get("products"),

        'total\_price': request.session.get("total\_price"),

        'customer\_name': request.session.get("customer\_name"),

        'customer\_email': request.session.get("customer\_email"),

        'now': datetime.now()

    }

    result = viewsdata(user\_data)

    print(f"Result from viewsdata: {result}")  # Log the result for debugging

    if result.startswith("Error"):

        # Redirect to edit\_customer with the customer's name and email as URL parameters

        return redirect('recipt:edit\_customer', customer\_name=user\_data['customer\_name'], customer\_email=user\_data['customer\_email'])

    elif result == "Success":

        # Render the redirect\_popup.html template with customer details

        return render(request, 'recipt/redirect\_popup.html', {

            'customer\_name': user\_data['customer\_name'],

            'customer\_email': user\_data['customer\_email']

        })

    else:

        # Render the redirect\_popup.html template with an error message

        return render(request, 'recipt/redirect\_popup.html', {

            'customer\_name': user\_data['customer\_name'],

            'customer\_email': user\_data['customer\_email'],

            'error': 'An unexpected issue occurred while sending the email.'

        })

def add(request):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    if request.method == 'POST':

        form = NewDataForm(request.POST)

        form\_customer = CustomerForm(request.POST)

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

            name = form.cleaned\_data['name']

            price = form.cleaned\_data['price']

            quantity = form.cleaned\_data['quantity']

            quantity\_price = price \* quantity

            request.session["products"].append([name, quantity, price, quantity\_price])

            request.session['total\_price'] += quantity\_price

            customer\_name = form\_customer.cleaned\_data['customer\_name']

            customer\_email = form\_customer.cleaned\_data['customer\_email']

            if customer\_name:

                request.session["customer\_name"] = customer\_name

            if customer\_email:

                request.session["customer\_email"] = customer\_email

        else:

            return render(request, 'recipt/add.html', {'form': form, 'form\_customer': form\_customer})

    return render(request, 'recipt/add.html', {"form": NewDataForm(), 'form\_customer': CustomerForm()})

def new\_receipt(request):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    # Reset session data for a new receipt

    request.session["products"] = []

    request.session["total\_price"] = 0

    request.session["customer\_name"] = None

    request.session["customer\_email"] = None

    return redirect('recipt:add')

def dele(request, id):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    try:

        product = request.session["products"].pop(id)

        request.session['total\_price'] -= product[3]  # Subtract the quantity\_price

    except IndexError:

        pass  # Handle index errors if necessary

    return redirect('recipt:index')

def edit\_customer(request, customer\_name, customer\_email):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    if request.method == 'POST':

        customer\_form = CustomerForm(request.POST)

        if customer\_form.is\_valid():

            customer\_name = customer\_form.cleaned\_data['customer\_name']

            customer\_email = customer\_form.cleaned\_data['customer\_email']

            request.session['customer\_name'] = customer\_name

            request.session['customer\_email'] = customer\_email

            return redirect('recipt:index')

    else:

        customer\_form = CustomerForm(initial={'customer\_name': customer\_name, 'customer\_email': customer\_email})

    return render(request, 'recipt/edit\_customer.html', {"customer\_form": customer\_form, "customer\_name": customer\_name, "customer\_email": customer\_email})

def edit\_product(request, id):

    if not request.user.is\_authenticated:

        return HttpResponseRedirect(reverse("recipt:login"))

    try:

        product = request.session["products"][id]

        name, quantity, price, quantity\_price = product

    except IndexError:

        return redirect('recipt:index')  # Redirect if invalid ID

    if request.method == 'POST':

        form = NewDataForm(request.POST)

        if form.is\_valid():

            # Update session data

            new\_name = form.cleaned\_data['name']

            new\_price = form.cleaned\_data['price']

            new\_quantity = form.cleaned\_data['quantity']

            new\_quantity\_price = new\_price \* new\_quantity

            # Update the product

            request.session["products"][id] = [new\_name, new\_quantity, new\_price, new\_quantity\_price]

            # Recalculate total price

            request.session['total\_price'] = sum(p[3] for p in request.session["products"])

            return redirect('recipt:index')

    else:

        form = NewDataForm(initial={'name': name, 'price': price, 'quantity': quantity})

    return render(request, 'recipt/add.html', {"form": form, 'is\_editing': True, 'id': id})

def login\_view(request):

    if request.method == "POST":

        username = request.POST["username"]

        password = request.POST["password"]

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

        if user is not None:

            login(request, user)

            return HttpResponseRedirect(reverse("recipt:new\_receipt"))

        else:

            return render(request, "recipt/login.html", {

                "message": "Invalid credentials.",

                "username": username  # Retain the entered username

            })

    return render(request, "recipt/login.html")

def logout\_view(request):

    logout(request)

    return HttpResponseRedirect(reverse("recipt:login"))

from django.shortcuts import render

from django.http import HttpResponse

from django.views.decorators.csrf import csrf\_exempt

from django.db import connection

@csrf\_exempt  # Use this if you want to bypass CSRF for testing; remove in production

def save\_customer(request):

    customer\_name = request.session.get("customer\_name")

    customer\_email = request.session.get("customer\_email")

    with connection.cursor() as cursor:

        cursor.execute("""

            INSERT INTO customers (name, email) VALUES (%s, %s)

        """, [customer\_name, customer\_email])

    return HttpResponse("Customer data inserted successfully.")

urls.html: from django.urls import path

from . import views

app\_name = 'recipt'

urlpatterns = [

    path('', views.index, name='index'),

    path('add/', views.add, name='add'),

    path('new/', views.new\_receipt, name='new\_receipt'),

    path('delete/<int:id>/', views.dele, name='dele'),

    path('edit\_product/<int:id>/', views.edit\_product, name='edit\_product'),

    path('edit\_customer/<str:customer\_name>/<str:customer\_email>/', views.edit\_customer, name='edit\_customer'),

    path('sendmail/', views.sendmail, name='sendmail'),

    path('login/', views.login\_view, name='login'),

    path('logout/', views.logout\_view, name='logout'),

    path('save-customer/', views.save\_customer, name='save\_customer'),

]

Sendmail.py: import smtplib

from email.mime.text import MIMEText

from datetime import datetime

class EmailSupportAgent:

    def \_\_init\_\_(self, smtp\_server, smtp\_port, smtp\_user, smtp\_password):

        self.smtp\_server = smtp\_server

        self.smtp\_port = smtp\_port

        self.smtp\_user = smtp\_user

        self.smtp\_password = smtp\_password

    def create\_receipt\_body(self, email\_data):

        body = f"""

        -----------------------------------

        E-RECEIPT

        -----------------------------------

        Customer Details:

        -------------------

        Name: {email\_data['customer\_name']}

        Email: {email\_data['customer\_email']}

        Products Purchased:

        -------------------

        """

        for product in email\_data['products']:

            name, quantity, price, quantity\_price = product

            body += f"""

            - Product: {name}

            Quantity: {quantity}

            Price per Unit: ${price:.2f}

            Total: ${quantity\_price:.2f}

            """

        body += f"""

        -------------------

        Total Price: ${email\_data['total\_price']:.2f}

        User Information:

        -------------------

        First Name: {email\_data['first\_name']}

        Last Name: {email\_data['last\_name']}

        Username: {email\_data['username']}

        Date of Purchase: {email\_data['now'].strftime('%B %d, %Y at %I:%M %p')}

        -----------------------------------

        Thank you for shopping with us!

        -----------------------------------

        """

        return body

    def send\_email(self, subject, body, to\_email):

        msg = MIMEText(body, \_charset='utf-8')

        msg['Subject'] = subject

        msg['From'] = self.smtp\_user

        msg['To'] = to\_email

        try:

            with smtplib.SMTP(self.smtp\_server, self.smtp\_port) as server:

                server.starttls()

                server.login(self.smtp\_user, self.smtp\_password)

                server.send\_message(msg)

            return "Success"

        except smtplib.SMTPAuthenticationError as e:

            return f"Authentication Error: {str(e)}"

        except smtplib.SMTPConnectError as e:

            return f"Connection Error: {str(e)}"

        except smtplib.SMTPDataError as e:

            return f"Data Error: {str(e)}"

        except smtplib.SMTPException as e:

            return f"SMTP Error: {str(e)}"

        except Exception as e:

            return f"Unexpected Error: {str(e)}"

    def handle\_incoming\_email(self, email\_data):

        subject = f"{email\_data['customer\_name']}'s Purchase Receipt from BUGS at {email\_data['now'].strftime('%B %d, %Y at %I:%M %p')}"

        body = self.create\_receipt\_body(email\_data)

        return self.send\_email(subject, body, email\_data['customer\_email'])

# Global initialization of support\_agent

support\_agent = EmailSupportAgent(

    'smtp.gmail.com', 587,

    'saqlainfawad@gmail.com', 'jtpqvszrodmcarlt'  # Replace with your actual app password

)

def viewsdata(email\_data):

    return support\_agent.handle\_incoming\_email(email\_data)

redirect\_popup.html: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Redirecting</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            height: 100vh;

            margin: 0;

            background-color: #f8f9fa;

            color: #333;

        }

        .container {

            text-align: center;

            border: 1px solid #ddd;

            border-radius: 8px;

            padding: 20px;

            background-color: #fff;

            box-shadow: 0 4px 8px rgba(0,0,0,0.1);

        }

        .message {

            font-size: 18px;

            margin-bottom: 20px;

        }

        .error {

            color: red;

            margin-top: 10px;

        }

        h1 {

            font-size: 24px;

            margin: 20px 0;

        }

    </style>

    <script type="text/javascript">

        window.onload = function() {

            // Redirect after 3 seconds

            setTimeout(function() {

                window.location.href = "{% url 'recipt:new\_receipt' %}";

            }, 3000); // 3000 milliseconds = 3 seconds

        };

    </script>

</head>

<body>

    <div class="container">

        <h2 class="message">E-Receipt sent to {{ customer\_name }} at {{ customer\_email }}</h2>

        {% if error %}

        <p class="error">{{ error }}</p>

        {% endif %}

        <h1>Redirecting...</h1>

    </div>

</body>

</html>

Settings.py: """

Django settings for recipt\_project project.

Generated by 'django-admin startproject' using Django 5.0.8.

For more information on this file, see

https://docs.djangoproject.com/en/5.0/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/5.0/ref/settings/

"""

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-2j%1h14$s3e@t&-ut@m#g\*n-\_(9$gsd^#6l3ppv+!\*ty658t\*t'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'recipt\_app'

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'recipt\_project.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'recipt\_project.wsgi.application'

# Database

# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'mssql',

        'NAME': 'reciptapp',  # Your database name

        'USER': '',  # Leave empty for Windows Authentication

        'PASSWORD': '',  # Leave empty for Windows Authentication

        'HOST': 'localhost',  # Or your server name

        'PORT': '',  # Leave blank for default

        'OPTIONS': {

            'driver': 'ODBC Driver 17 for SQL Server',  # Make sure the driver is installed

            'extra\_params': 'Trusted\_Connection=yes;',  # This enables Windows Authentication

        },

    }

}

print('Database Connected')

# Password validation

# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC\_URL = 'static/'

# Default primary key field type

# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

Models.py: from django.db import models

# Create your models here.

from django.db import connection

from django.http import HttpResponse

def insert\_customer\_data(customer\_name, customer\_email):

    with connection.cursor() as cursor:

        cursor.execute("""

            INSERT INTO customers (name, email) VALUES (?, ?)

        """, [customer\_name, customer\_email])

    return HttpResponse("Customer data inserted successfully.")

SQL:

