```
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&p
id=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&p
id=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&p
id=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;
```

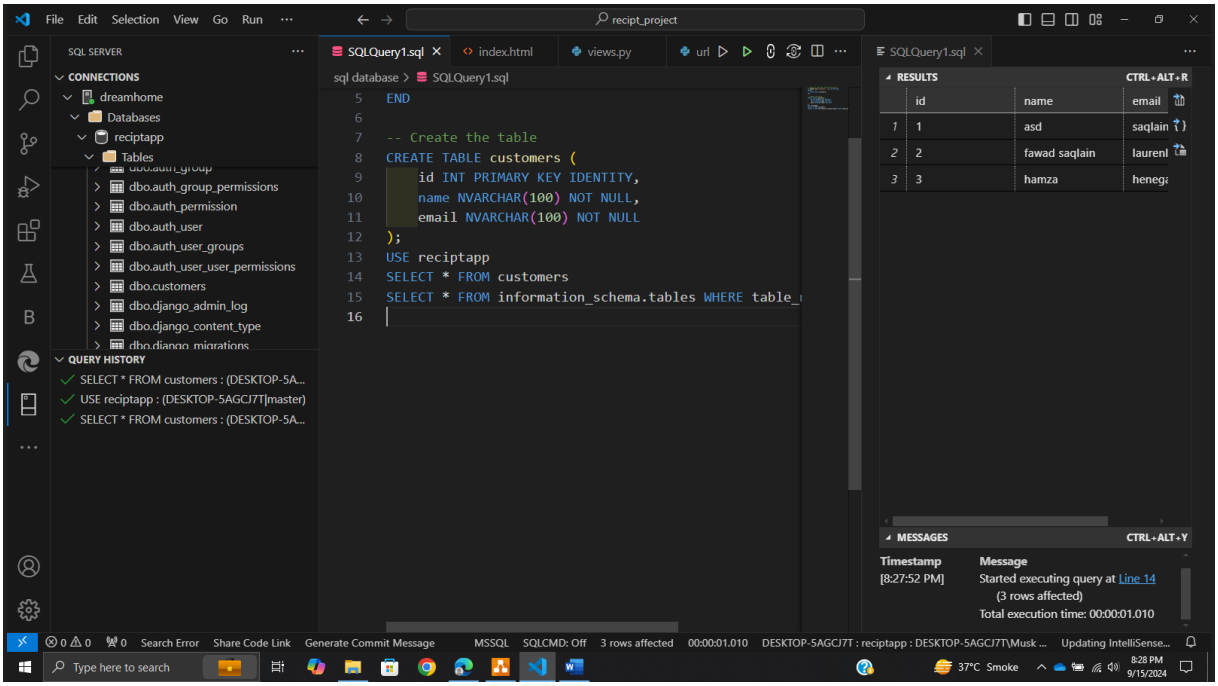
```
            border-radius: 8px;
            padding: 20px;
            background-color: #fff;
            box-shadow: 0 4px 8px rgba(0,0,0,0.1);
        }
        .message {
            font-size: 18px;
            margin-bottom: 20px;
        }
        .error {
            color: red;
            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.UserAttributeSimilarityVali
dator',
    },
    {
        '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:
```


File Edit Selection View Go Run
recipt_project
SQL SERVER
CONNECTIONS
dreamhome
Databases
reciptapp
Tables
dbo.auth_group_permissions
dbo.auth_permission
dbo.auth_user
dbo.auth_user_groups
dbo.auth_user_user_permissions
dbo.customers
dbo.django_admin_log
dbo.django_content_type
QUERY HISTORY
SELECT * FROM customers : (DESKTOP-5A...
USE reciptapp : (DESKTOP-5AGCJ7T\master)
SELECT * FROM customers : (DESKTOP-5A...
SQLQuery1.sql
index.html
views.py
url
sql database > SQLQuery1.sql
5 END
6
7 -- Create the table
8 CREATE TABLE customers (
9 id INT PRIMARY KEY IDENTITY,
10 name NVARCHAR(100) NOT NULL,
11 email NVARCHAR(100) NOT NULL
12);
13 USE reciptapp
14 SELECT * FROM customers
15 SELECT * FROM information_schema.tables WHERE table_
16
RESULTS CTRL+ALT+R
id name email
1 1 asd saqlain
2 2 fawad saqlain laurenl
3 3 hamza heneg
MESSAGES CTRL+ALT+Y
Timestamp Message
[8:27:52 PM] Started executing query at Line 14
(3 rows affected)
Total execution time: 00:00:01.010
Search Error Share Code Link Generate Commit Message
MSSQL SQLCMD: Off 3 rows affected 00:00:01.010
DESKTOP-5AGCJ7T : reciptapp : DESKTOP-5AGCJ7T\Musk ...
Updating IntelliSense...
Type here to search
37°C Smoke
8:28 PM
9/15/2024