MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006

(Approved by AICTE, New Delhi)

UNIVERSITY OF MYSORE

Full Stack Development(21CD71) Assessment Report On:

"COMPLAINT HANDLING SYSTEM "

Under the guidance :
Mr. Karthik M N
Assistant Professor,
Department of Computer Science & Design,
MUSE.

Submitted by:
VARSHINI H S
Reg No : 21SECD54

COMPLAINT HANDLING SYSTEM - DJANGO PROJECT REPORT

OVERVIEW

The Complaint Handling System is a Django-based web application where users can register their complaints online.
The system stores these complaints in a database and allows administrators to manage them through the Django Admin panel.

MAIN FEATURES:

- Complaint form for users
- Database storage for complaint records
- Admin panel for managing complaints
- Light-themed UI with a structured layout

TECHNOLOGY STACK

This project is developed using:

- Backend: Django (Python Framework)
- Frontend: HTML, CSS
- Database: SQLite
- Development Tool: VS Code
- Version Control: GitHub

PROJECT SETUP & FOLDER STRUCTURE

Installation Steps:

Install Django:

pip install django

Create Django Project & App:

django-admin startproject complaint_system

cd complaint_system

python manage.py startapp complaints

Apply Migrations & Run Server:

python manage.py migrate

python manage.py runserver

FOLDER STRUCTURE:

```
complaint_system/
├── complaints/
│   ├── static/complaints/style.css
│   ├── templates/complaints/home.html
│   │              /complaint_file.html
│   │              /success.html
│   ├── models.py
│   ├── views.py
│   ├── urls.py
├── manage.py
├── db.sqlite3
```

DATABASE DESIGN (MODELS)

The system uses the Complaint model to store complaint data.

models.py file:

from django.db import models

```
class Complaint(models.Model):
PRIORITY_LEVELS = [('High', 'High'), ('Medium', 'Medium'), ('Low', 'Low')]
STATUS_OPTIONS = [('Pending', 'Pending'), ('Resolved', 'Resolved')]

name = models.CharField(max_length=100)
email = models.EmailField()
product = models.CharField(max_length=100)
description = models.TextField()
priority = models.CharField(max_length=10, choices=PRIORITY_LEVELS, default='Medium')
status = models.CharField(max_length=10, choices=STATUS_OPTIONS, default='Pending')

def __str__(self):
    return self.product
```

VIEWS & FORMS HANDLING

The **form allows users to submit complaints** and stores the data in the database.

forms.py file:

```
from django import forms
from .models import Complaint
class ComplaintForm(forms.ModelForm):
    class Meta:
        model = Complaint
        fields = ['name', 'email', 'product', 'issue_description', 'priority']
```

views.py file:

```
from django.shortcuts import render, redirect
```

```
from django.urls import reverse_lazy

from .forms import ComplaintForm

def submit_complaint(request):

if request.method == 'POST':

form = ComplaintForm(request.POST)

if form.is_valid():

form.save()

return redirect(reverse_lazy('complaint_success'))

else:

form = ComplaintForm()

return render(request, 'complaints/complaint_file.html', {'form': form})
```

Users are redirected to a success page after submitting the form.

URL MAPPING

The project's urls.py is structured to handle user navigation.

Project URLs (complaint_system/urls.py)

```
from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

path('', include('complaints.urls')),

]
```

App URLs (complaints/urls.py)

```
from django.urls import path

from .views import home, submit_complaint, complaint_success

urlpatterns = [

path('', home, name='home'),
```

path('submit/', submit_complaint, name='submit_complaint'),

path('success/', complaint_success, name='complaint_success'),

]

This ensures that the URLs correctly route to the appropriate views.

USER INTERFACE (TEMPLATES)

- **home.html** : Displays homepage content

home.html

complaints > templates > complaints > home.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Complaint System</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">
</head>
<body>
    <div class="container">
        <h1>Welcome to the Customer Complaint System</h1>
        <p>If you have any issues, feel free to submit a complaint.</p>
        <a href="{% url 'submit_complaint' %}">Submit a Complaint</a>
    </div>
</body>
</html>
```

- **complaint_file.html** : Contains the complaint submission form

home.html | complaint_file.html

complaints > templates > complaints > complaint_file.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Submit Complaint</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">

</head>
<body>
    <h2>Submit Your Complaint</h2>
    <form method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

- **success.html** : Shows confirmation after form submission

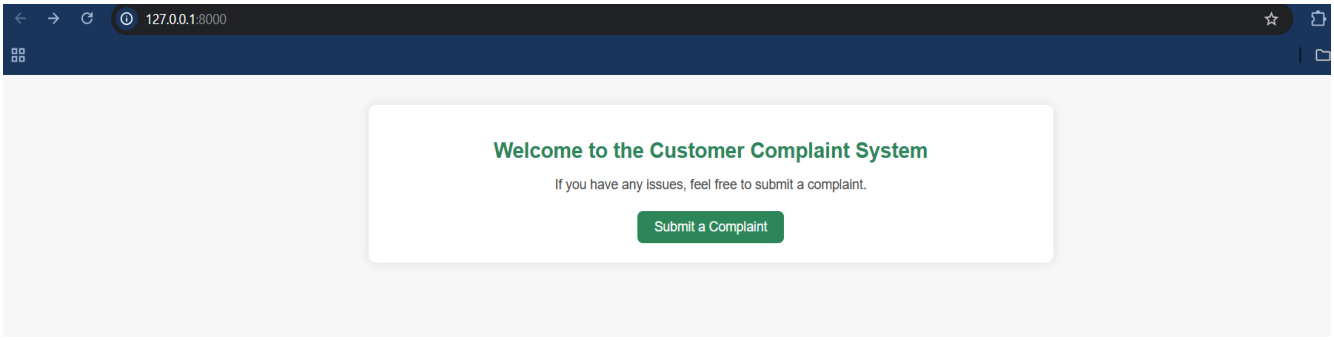
home.html success.html

complaints > templates > complaints > success.html > ...

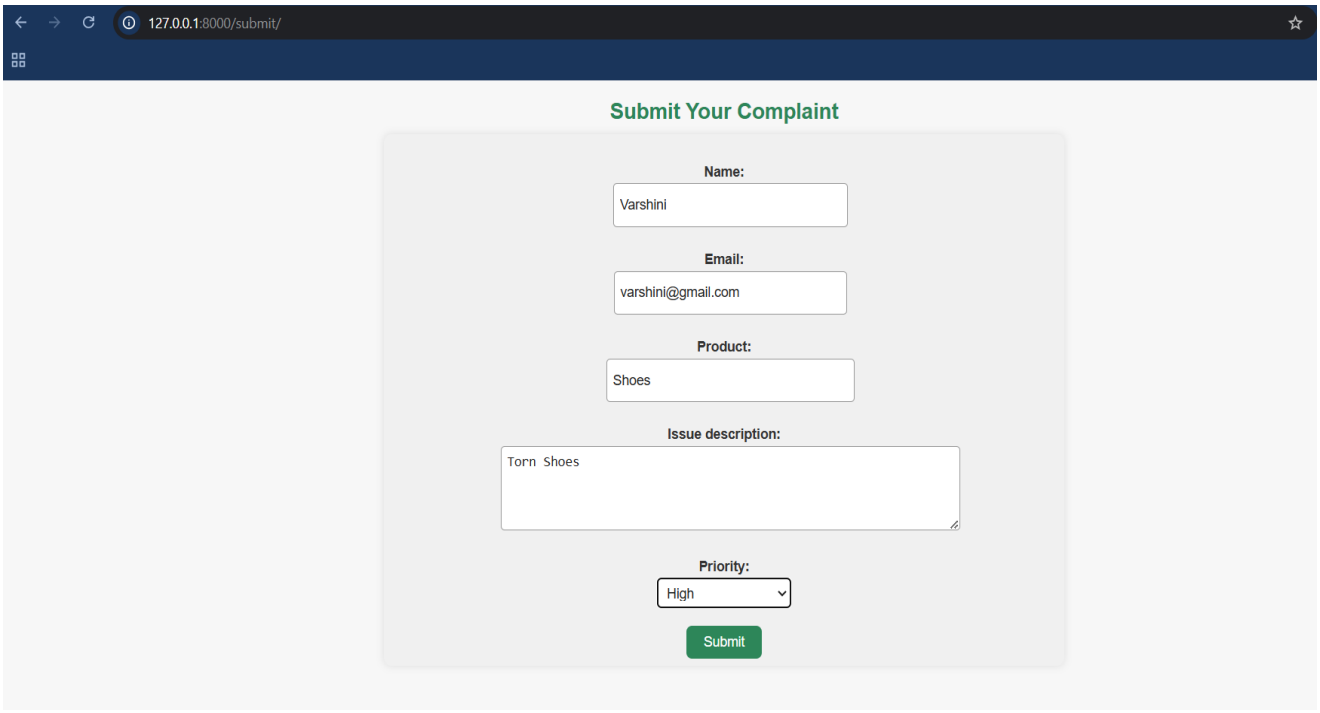
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Success</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">
</head>
<body>
    <h2>Thank you! Your complaint has been submitted.</h2>
    <a href="{% url 'submit_complaint' %}">Submit Another Complaint</a>

</body>
</html>
```

SCREENSHOTS

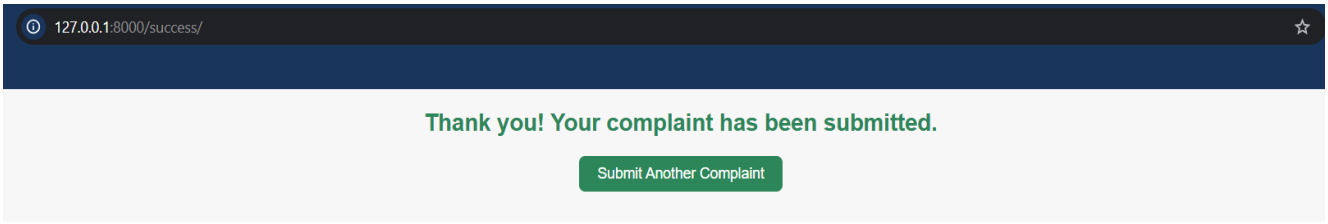
Home Page Screenshot



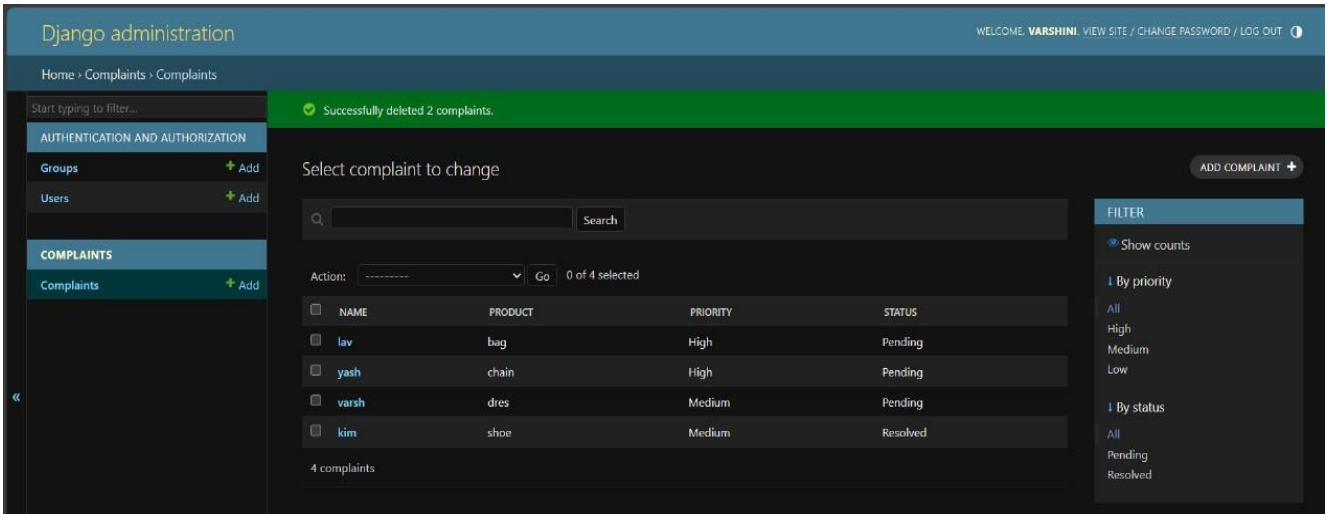
Complaint Form Screenshot



Success Page Screenshot



Admin Panel Screenshot



Implementing reverse_lazy() to redirect users to a success page after submission.

views.py

complaints > views.py > ...

```python
from django.shortcuts import render, redirect # type: ignore
from django.urls import reverse_lazy # type: ignore
from .forms import ComplaintForm

def home(request):
    return render(request, 'complaints/home.html')

def submit_complaint(request):
    if request.method == "POST":
        form = ComplaintForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(reverse_lazy('complaint_success'))
    else:
        form = ComplaintForm()
    return render(request, 'complaints/complaint_file.html', {'form': form})

def complaint_success(request):
    return render(request, 'complaints/success.html')

```