



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:

Yashaswini P
Reg No : 21SECD57

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 X  
complaints > templates > complaints > home.html > ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4 <meta charset="UTF-8">  
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6 <title>Home - Complaint System</title>  
7 {% load static %}  
8 <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">  
9 </head>  
10 <body>  
11 <div class="container">  
12 <h1>Welcome to the Customer Complaint System</h1>  
13 <p>If you have any issues, feel free to submit a complaint.</p>  
14 <a href="{% url 'submit_complaint' %}">Submit a Complaint</a>  
15 </div>  
16 </body>  
17 </html>  
18 |
```

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

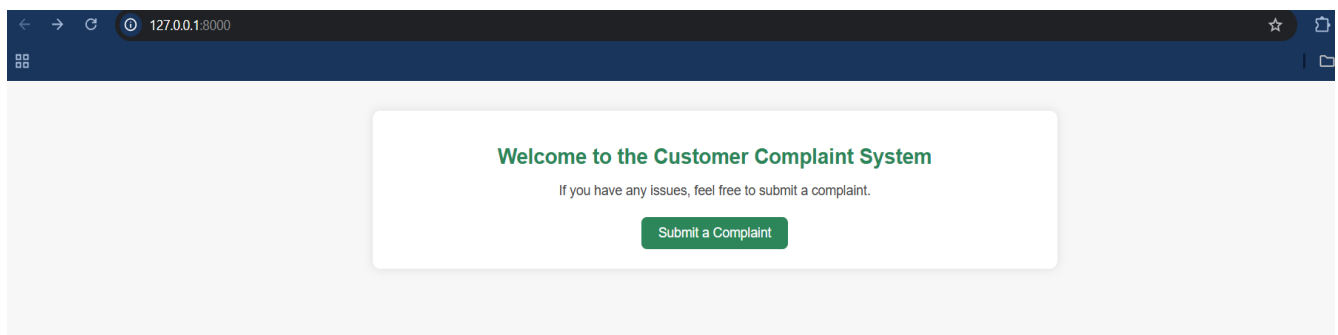
```
home.html complaint_file.html X  
complaints > templates > complaints > complaint_file.html > ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4 <meta charset="UTF-8">  
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6 <title>Submit Complaint</title>  
7 {% load static %}  
8 <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">  
9 </head>  
10 <body>  
11 <h2>Submit Your Complaint</h2>  
12 <form method="POST">  
13 <div class="form-group">  
14 <input type="text" value="{% csrf_token %}">  
15 <input type="text" value="{% form.as_p %}">  
16 <button type="submit">Submit</button>  
17 </form>  
18 </body>  
19 </html>  
20 |
```

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

```
home.html success.html X
complaints > templates > complaints > success.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Success</title>
7   {% load static %}
8   <link rel="stylesheet" type="text/css" href="{% static 'complaints/style.css' %}">
9 </head>
10 <body>
11   <h2>Thank you! Your complaint has been submitted.</h2>
12   <a href="{% url 'submit_complaint' %}">Submit Another Complaint</a>
13
14 </body>
15 </html>
16
```

SCREENSHOTS

Home Page Screenshot



Complaint Form Screenshot

Submit Your Complaint

Name:

Email:

Product:

Issue description:

Priority:

Success Page Screenshot

Thank you! Your complaint has been submitted.

Admin Panel Screenshot

Django administration

WELCOME, VARSHINI. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Complaints > Complaints

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups
- Users

COMPLAINTS

- Complaints

Select complaint to change

Search

Action: Go 0 of 4 selected

<input type="checkbox"/>	NAME	PRODUCT	PRIORITY	STATUS
<input type="checkbox"/>	lav	bag	High	Pending
<input type="checkbox"/>	yash	chain	High	Pending
<input type="checkbox"/>	varsh	dres	Medium	Pending
<input type="checkbox"/>	kim	shoe	Medium	Resolved

4 complaints

FILTER

- Show counts
- By priority
 - All
 - High
 - Medium
 - Low
- By status
 - All
 - Pending
 - Resolved

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

views.py

complaints > views.py > ...

```
1  from django.shortcuts import render, redirect # type: ignore
2  from django.urls import reverse_lazy # type: ignore
3  from .forms import ComplaintForm
4
5  def home(request):
6      return render(request, 'complaints/home.html')
7
8  def submit_complaint(request):
9      if request.method == "POST":
10         form = ComplaintForm(request.POST)
11         if form.is_valid():
12             form.save()
13             return redirect(reverse_lazy('complaint_success'))
14         else:
15             form = ComplaintForm()
16             return render(request, 'complaints/complaint_file.html', {'form': form})
17
18  def complaint_success(request):
19      return render(request, 'complaints/success.html')
20
```