



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 MANAGEMENT SYSTEM ”

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

Submitted by:
UTSAV M
21SECD52

COMPLAINT MANAGEMENT SYSTEM - DJANGO PROJECT

REPORT

INTRODUCTION

The **Complaint Management System** is a web-based application built using Django that allows users to submit complaints through a form.

The submitted complaints are stored in a database and managed via the Django Admin panel.

After submission, users are redirected to a **success page**.

KEY FEATURES:

- ✓ User-friendly complaint submission form
- ✓ Admin panel for complaint management
- ✓ Success page after submission
- ✓ Light-themed UI for better user experience

TECHNOLOGIES USED

The project is developed using the following technologies:

- **Backend:** Django (Python Web Framework)
- **Frontend:** HTML, CSS
- **Database:** SQLite3
- **Development Environment:** VS Code
- **Version Control:** GitHub

PROJECT SETUP & FOLDER STRUCTURE

Installation Steps:

Install Django:

```
pip install django
```

Create Django Project & App:

```
django-admin startproject customer_complaints_project
```

```
cd customer_complaints_project
```

```
python manage.py startapp complaints
```

Apply Migrations & Run Server:

```
python manage.py migrate
```

```
python manage.py runserver
```

FOLDER STRUCTURE:

Customer_complaints_project/

| — complaints/

| | — __python__/

| | — migrations/

| | — templates/

| | | — complaints/complaint_form.html

| | | — complaints/home.html

| | | — complaints/success.html

| | — models.py

| | — views.py

| | — urls.py

| — complaint_system/

| — db.sqlite3

| — manage.py

MODELS (DATABASE STRUCTURE)

The **Complaint model** is used to store user complaints in the database.

models.py file:

```
from django.db import models

class Complaint(models.Model):
    PRIORITY_CHOICES = [
        ('HIGH', 'High'),
        ('MEDIUM', 'Medium'),
        ('LOW', 'Low'),
    ]

    STATUS_CHOICES = [
        ('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=6, choices=PRIORITY_CHOICES, default='MEDIUM')
    status = models.CharField(max_length=8, choices=STATUS_CHOICES, default='PENDING')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f"{self.name} - {self.product} ({self.status})"

    class Meta:
        ordering = ['-created_at']
```

FORMS & VIEWS

The **complaint form** collects user input and saves it to 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', 'description', 'priority']
        widgets = {
            'description': forms.Textarea(attrs={'rows': 4}),
            'priority': forms.Select(attrs={'class': 'form-control'}),
        }
```

views.py file:

```
from django.views.generic.edit import CreateView
from django.urls import reverse_lazy
from django.shortcuts import render
from .models import Complaint
from .forms import ComplaintForm

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

```
class ComplaintCreateView(CreateView):
    model = Complaint
    form_class = ComplaintForm
    template_name = 'complaints/complaint_form.html'
    success_url = reverse_lazy('complaint_success')
```

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

URL CONFIGURATION

Project-level (complaint_system/urls.py) and app-level (complaints/urls.py) URL patterns are defined as follows:

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_view, ComplaintCreateView
from django.views.generic import TemplateView

urlpatterns = [
    path('', home_view, name='home'),
    path('submit/', ComplaintCreateView.as_view(), name='submit_complaint'),
    path('success/', TemplateView.as_view(
        template_name='complaints/success.html'),
        name='complaint_success'
    ),
]
```

TEMPLATES

Templates (HTML Files):

- home.html → Landing page

```
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container mt-5 text-center">
        <h1>Hey User!</h1>
        <h2>Welcome to the Complaint Management System</h2>
        <p>Your Voice, Our Priority - Submit, Track, and Resolve Complaints
Effortlessly!</p>
        <a href="{% url 'submit_complaint' %}" class="btn btn-primary">Submit a
Complaint</a>
    </div>
</body>
</html>
```

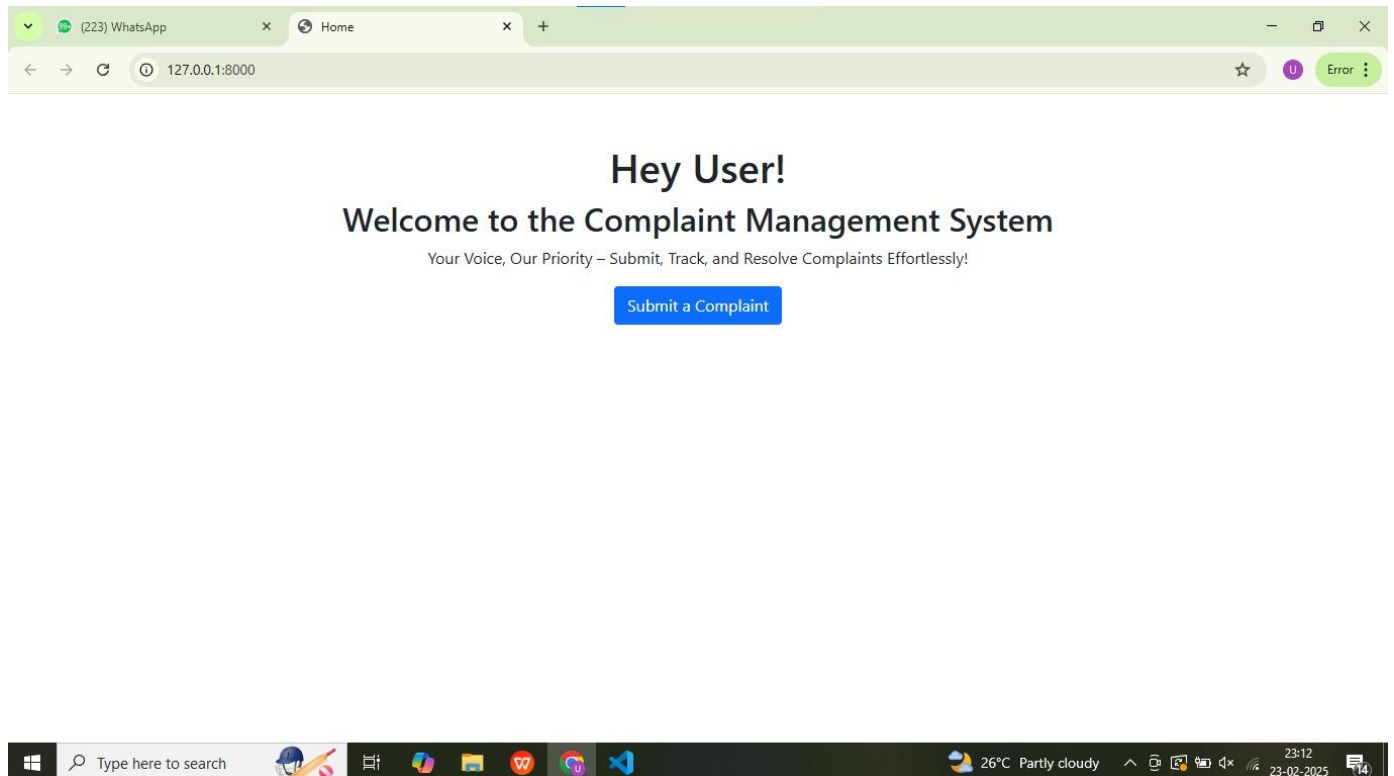
- **complaint_form.html** → Complaint submission form

```
<!DOCTYPE html>
<html>
<head>
  <title>Submit Complaint</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h2>Submit a Complaint</h2>
    <form method="post" class="mt-4">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit" class="btn btn-primary">Submit Complaint</button>
    </form>
  </div>
</body>
</html>
```

- **success.html** → Complaint successfully submitted message

SCREENSHOTS

Home Page Screenshot



Complaint Form Screenshot

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000'. The browser has several tabs open: '(223) WhatsApp', 'Uploading Full Stack Project', 'Log in | Django site admin', 'Django Customer Complaint', and 'Submit Complaint'. The 'Submit Complaint' tab is active. The page title is 'Submit a Complaint'. The form contains the following fields:

- Name:
- Email:
- Product:
- Description:
- Priority:

A blue button labeled 'Submit Complaint' is located at the bottom of the form.

Success Page Screenshot

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/success/'. The browser has several tabs open: '(223) WhatsApp', 'Uploading Full Stack Project', 'Log in | Django site admin', 'Django Customer Complaint', and 'Success'. The 'Success' tab is active. The page title is 'Thank You!'. The page content includes the text 'Your complaint has been successfully submitted.' and a blue button labeled 'Submit Another Complaint'.

Admin Panel Screenshot

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin/complaints/complaint/`. The page title is "Django administration". The navigation bar includes "Home > Complaints > Complaints" and a welcome message for "UTSAV.M" with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

On the left, there is a sidebar menu with sections: "AUTHENTICATION AND AUTHORIZATION" (Groups, Users) and "COMPLAINTS" (Complaints). The "Complaints" section is highlighted.

The main content area shows a message "Successfully deleted 2 complaints." and a heading "Select complaint to change". Below this is a search bar and a table of complaints. The table has columns: NAME, EMAIL, PRODUCT, PRIORITY, STATUS, and CREATED AT. There are three rows of data:

	NAME	EMAIL	PRODUCT	PRIORITY	STATUS	CREATED AT
<input type="checkbox"/>	Manju	savagesam@gmail.com	Airpods	Medium	Pending	Feb. 23, 2025, 4:37 p.m.
<input type="checkbox"/>	srihari	srihari21@gmail.com	tv	High	Pending	Feb. 23, 2025, 4:34 p.m.
<input type="checkbox"/>	Sana	anahasclicks@gmail.com	One plus mobile	Medium	Resolved	Feb. 23, 2025, 4:33 p.m.

Below the table, it says "3 complaints" and there is a "Save" button. On the right, there is a "FILTER" sidebar with options to "Show counts", "By status" (All, Pending, Resolved), "By priority" (All, High, Medium, Low), and "By created at" (Any date, Today, Past 7 days). There is also an "ADD COMPLAINT" button.

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

```
views.py 2 X
complaints > views.py > submit_complaint
1 from django.shortcuts import render, redirect
2 from django.urls import reverse_lazy
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
```