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 ”

Submitted by:
UTSAV M
21SECD52

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

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 form should includes `name`, `email`, `product`, `issue description`, and `priority level (High, Medium, Low)`.

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
```

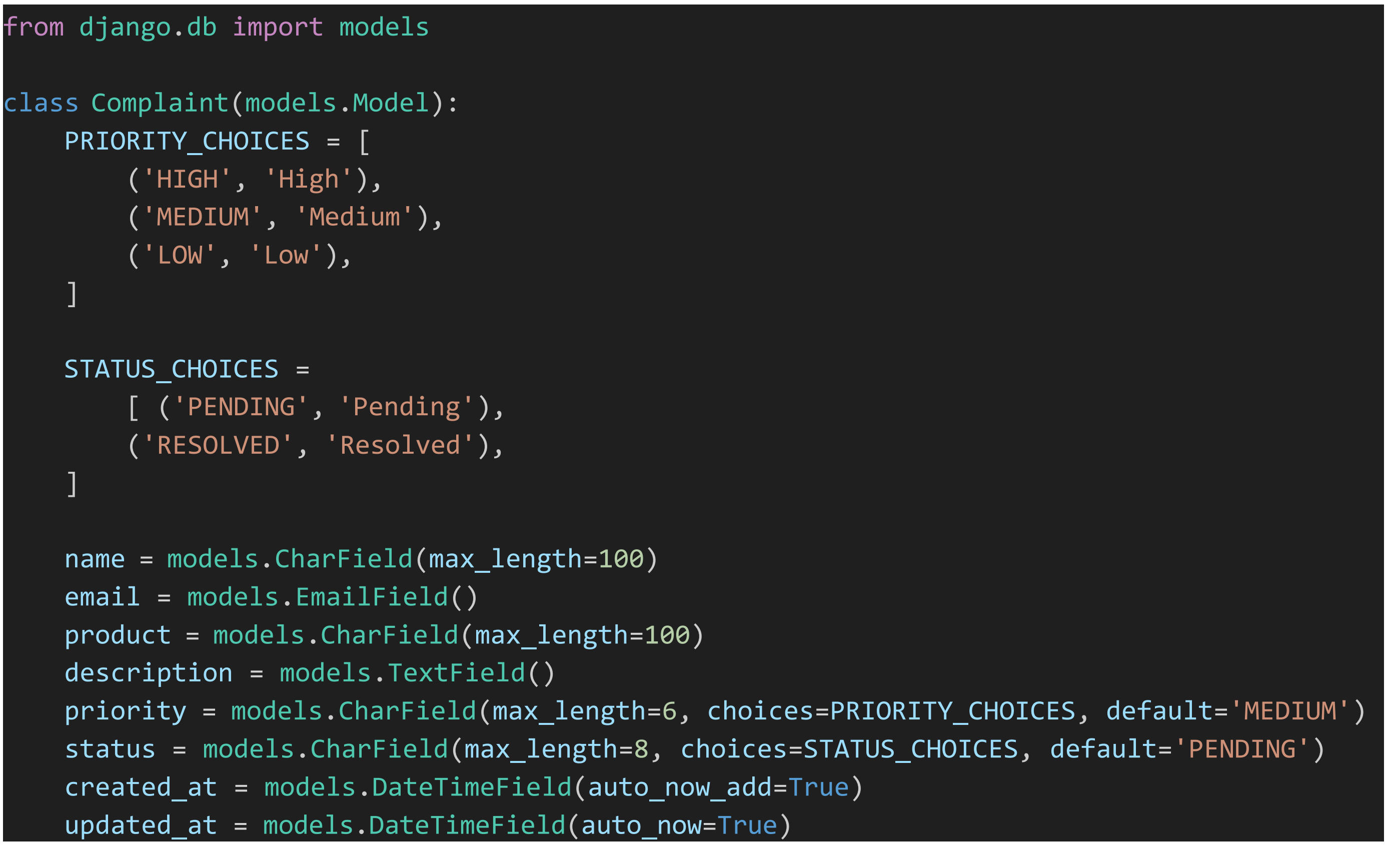
FOLDER STRUCTURE:

```
cutomer_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:

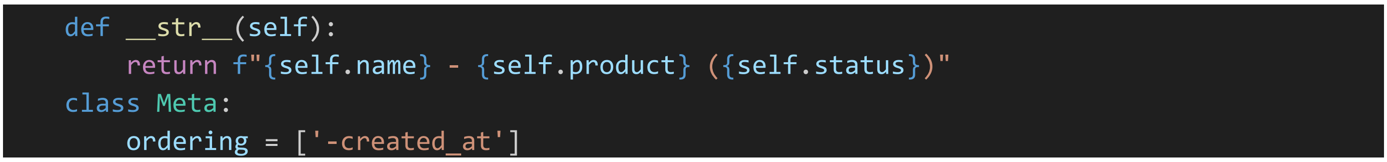


```
from django.db import models

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

# Project URLs (customer_complaints_project/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>
```

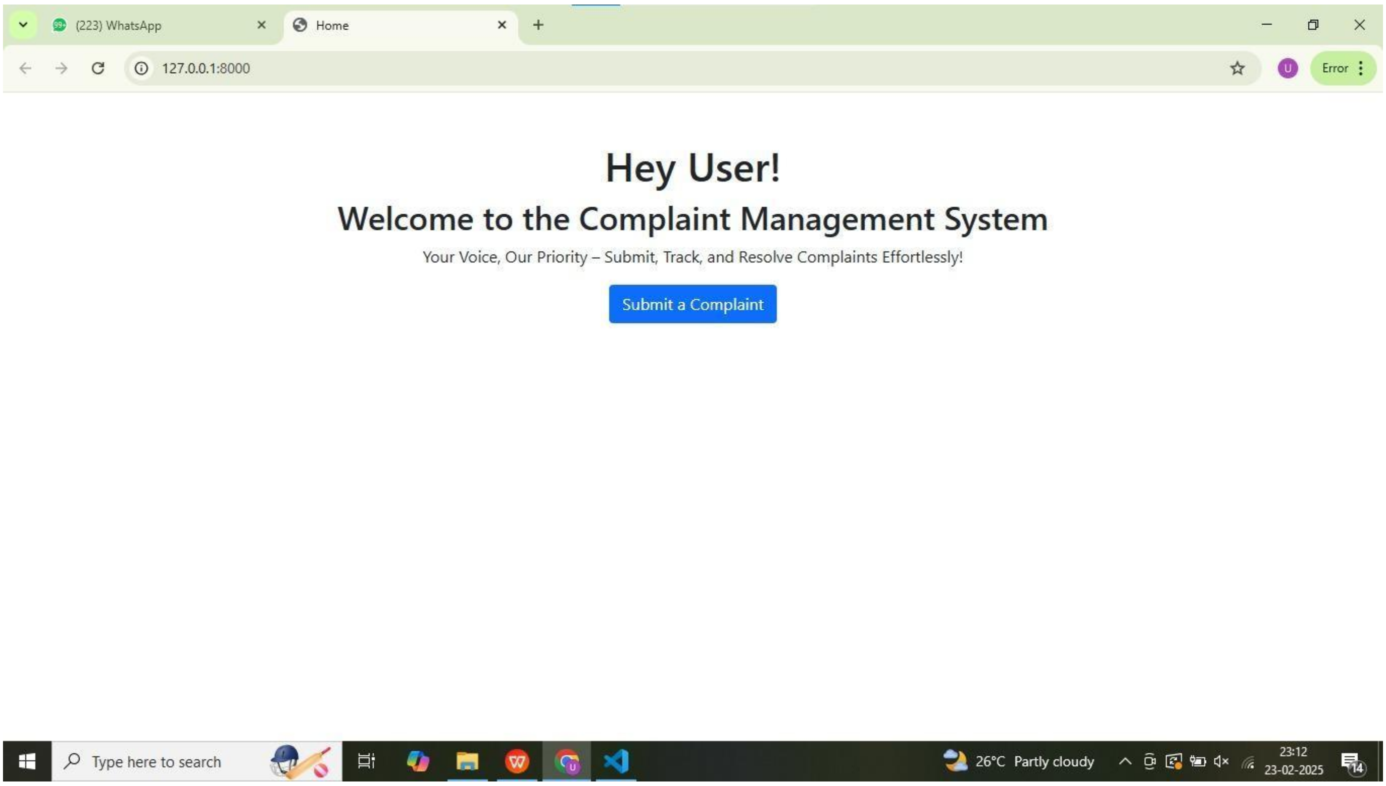
- **complaint_form.html** → Complaint submission form

```
<!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

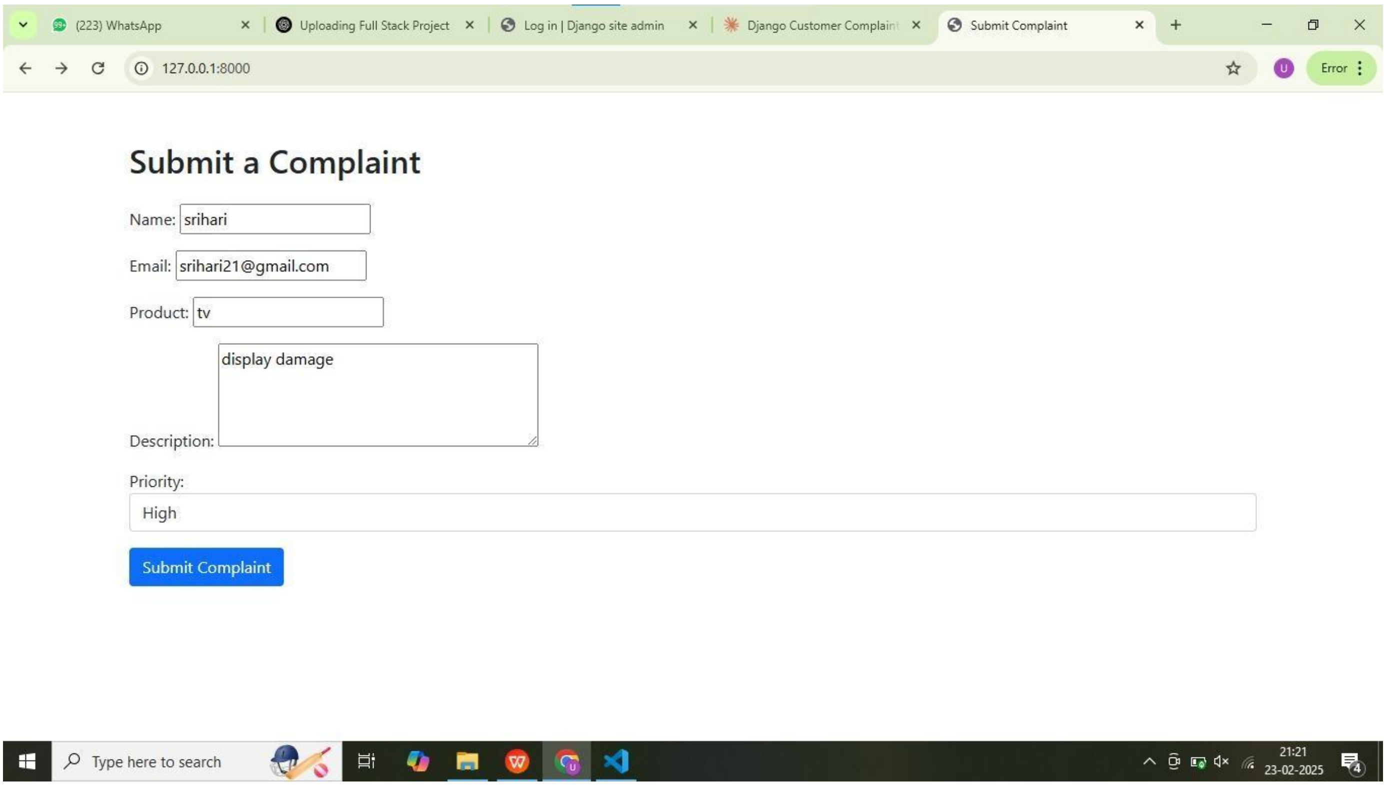


Hey User!

Welcome to the Complaint Management System

Your Voice, Our Priority – Submit, Track, and Resolve Complaints Effortlessly!

Complaint Form Screenshot



Submit a Complaint

Name: srihari

Email: srihari21@gmail.com

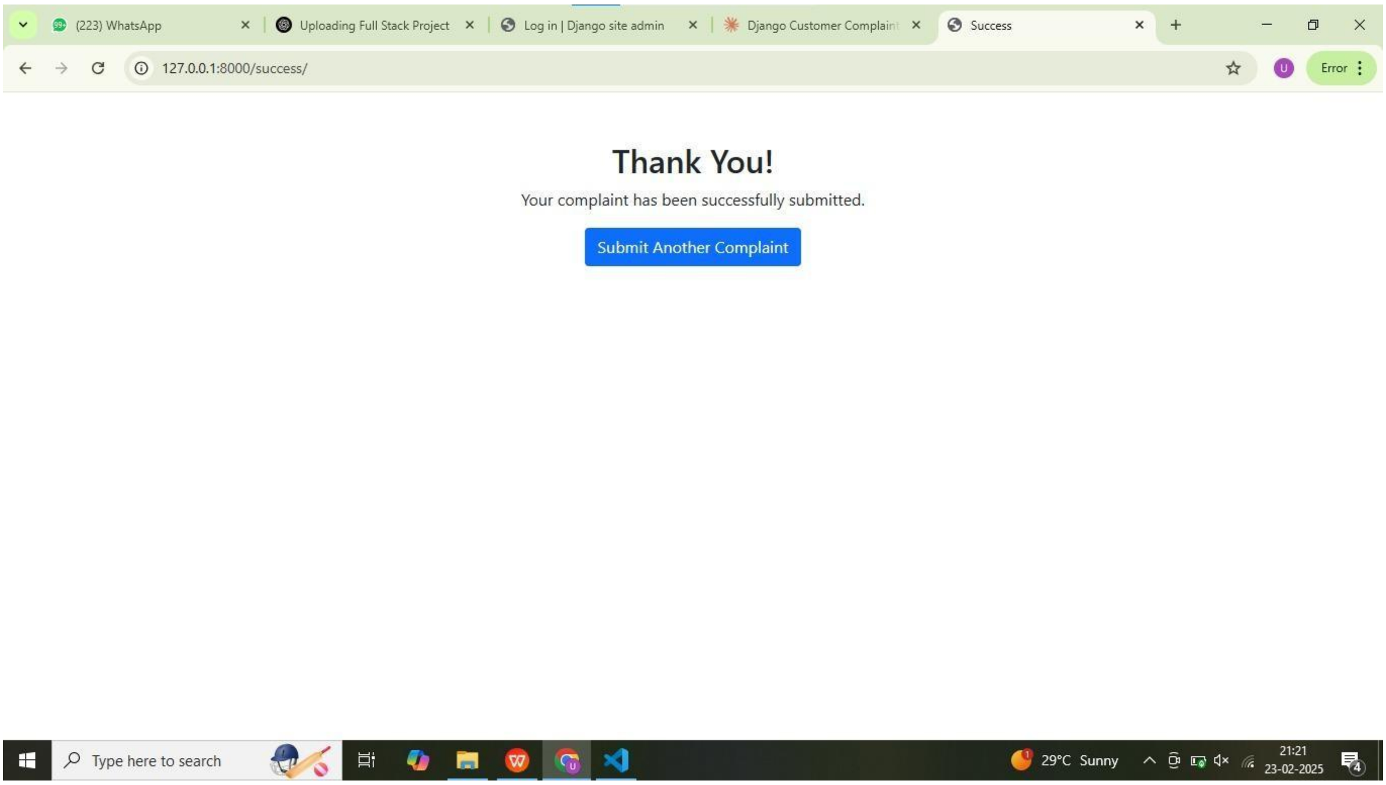
Product: tv

Description: display damage

Priority: High

Submit Complaint

Success Page Screenshot

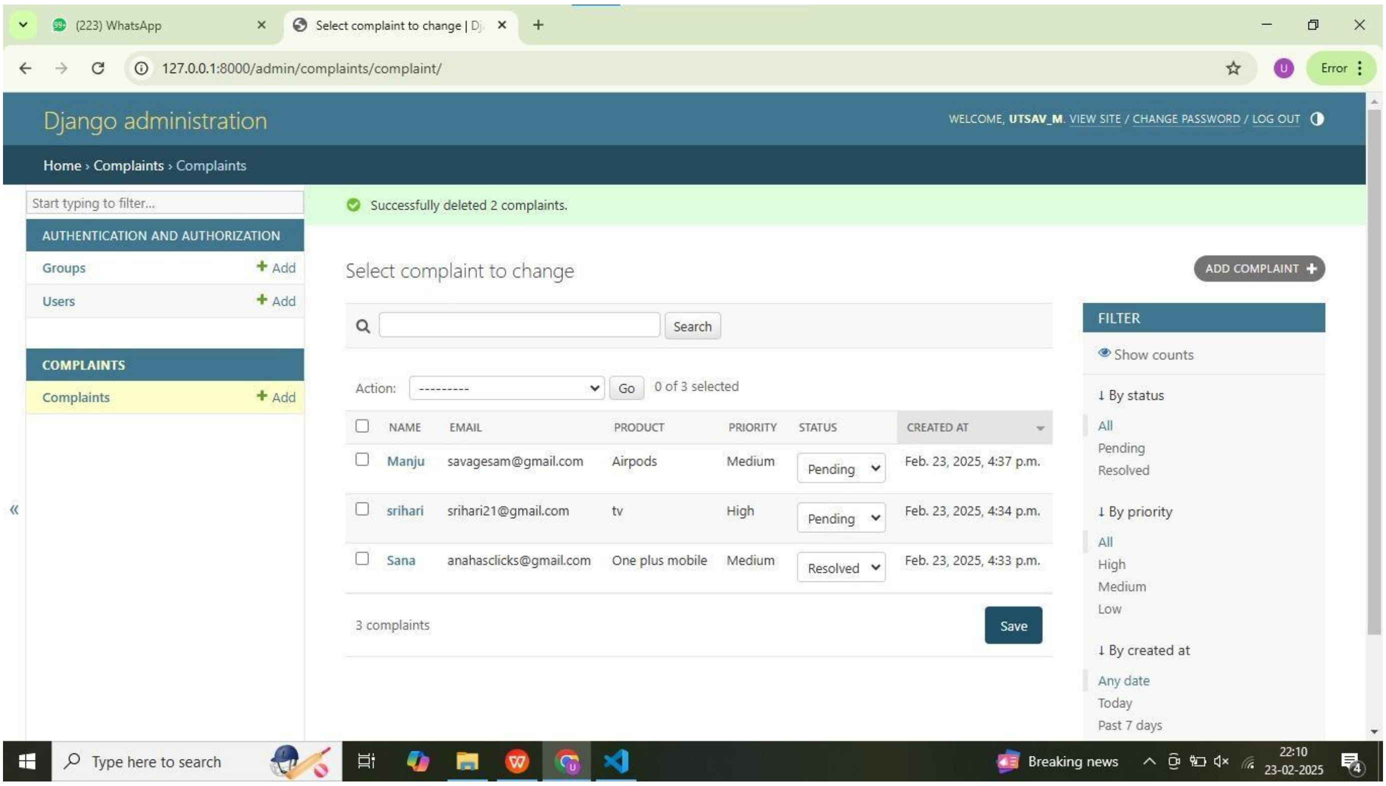


Thank You!

Your complaint has been successfully submitted.

Submit Another Complaint

Admin Panel Screenshot



(223) WhatsApp
Select complaint to change | Dj
127.0.0.1:8000/admin/complaints/complaint/
Error

Django administration

WELCOME, **UTSAV_M**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Complaints › Complaints

Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
COMPLAINTS
Complaints + Add

Successfully deleted 2 complaints.

Select complaint to change

ADD COMPLAINT +
Search
Action: ---------
Go

0 of 3 selected

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

3 complaints

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

```python
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')
```