MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006

(Approved by AICTE, New Delhi)

UNIVERSITY OF MYSORE

A Seminar(21CD71, Full Stack Development) Report

On

"Customer Complaints"

Submitted By

Syeda Aliya

21SECD51

7th Semester,

Department of CS&D

MUSE.

Under Faculty Incharge

1) Dr. M. S. Govinde Gowda, Director.
2) Mr. Karthik M N
   Asst. Professor,
   Dept. of CS&D
   MUSE.

Building a Full-Stack Django Web Applications for Customer Complaints - Introduction

i. **Objective :** Develop a web application using Django where customers can submit complaints, and administrators can manage and resolve them.

ii. **User Roles :**

Customers : Submit complaints via a form.

Admin : View, manage, and update complaint statuses.

iii. **Complaint Submission Form :**

Fields : Name, Email, Product, Issue Description, Priority Level (High, Medium, Low). Uses Django forms for validation and submission.

iv. **Database Management :**

Complaints are stored in Django's database (SQLite/PostgreSQL/MySQL). Managed through Django Admin Panel.

v. **Admin Functionality :**

View all complaints in a structured table format and Change the status of a complaint.

vi. **User Redirection with reverse_lazy():**

After submission, customers are redirected to a success page. Implements Django's **reverse_lazy()** for efficient URL redirection.

vii. **Technology Stack:**

Frontend : HTML, CSS, Bootstrap (for styling).

Backend : Django (Python-based web framework).

Database : SQLite (default) or PostgreSQL/MySQL

viii. **Outcome :**

A structured system for handling customer complaints efficiently. Easy management for
\ admins with a user-friendly dashboard.

Detailed Steps Implementation :

Step 1 : Install Django and Create a Virtual Environment

# Create a virtual environment

python -m venv venv

# Activate the virtual environment

# On Windows:

venv\Scripts\activate

# Install Django

pip install Django

Step 2 : Create a Django Project

Run the following command to create a Django project:

django-admin startproject customer_complaints

cd customer_complaints

Step 3 : Create a Django App

python manage.py startapp students

Step 4 : Configure settings.py

Open customer_complaints/settings.py and add 'students' to INSTALLED_APPS

Step 5 : Create the Complaint Model

Run migrations to apply the model:

python manage.py makemigrations

python manage.py migrate

Step 6 : Register the Model in Django Admin

In complaints/*admin.py:*

Step 7 : Create Views for customer_complaints

In complaints/*views.py:*

Step 8 : Configure URLs

Create complaints/*urls.py*

Link the complaints app to the project's main urls.py in *customer_complaints*/urls.py

Step 9 : Create HTML Templates

Inside students/templates/students/ create files naming:

1. *complaints_form.html*
2. *success_form.html*

Step 10: Create a Superuser for Admin Panel:

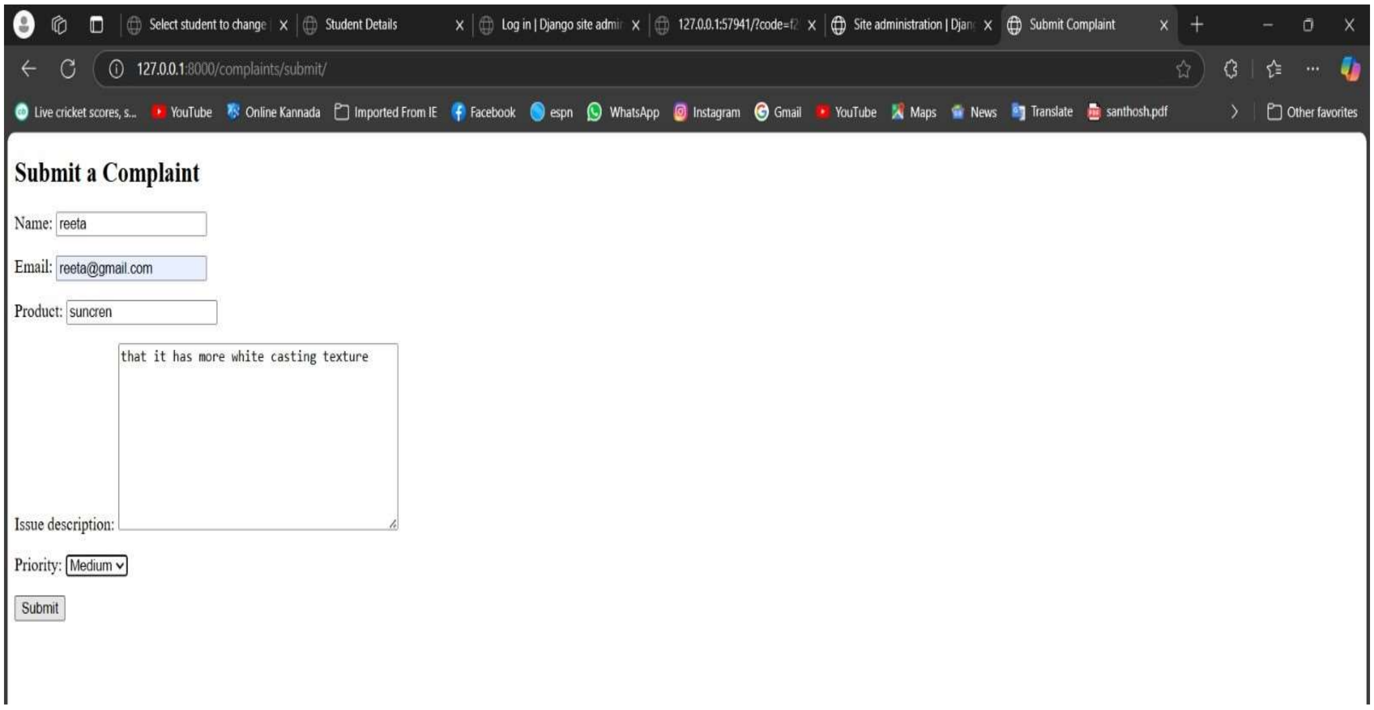
python manage.py createsuperuser

Step 11: Run the Django Development Server

python manage.py runserver

Step-by-Step Output Results :

i. Customers can submit complaints through a form –



Submit a Complaint

Name: reeta

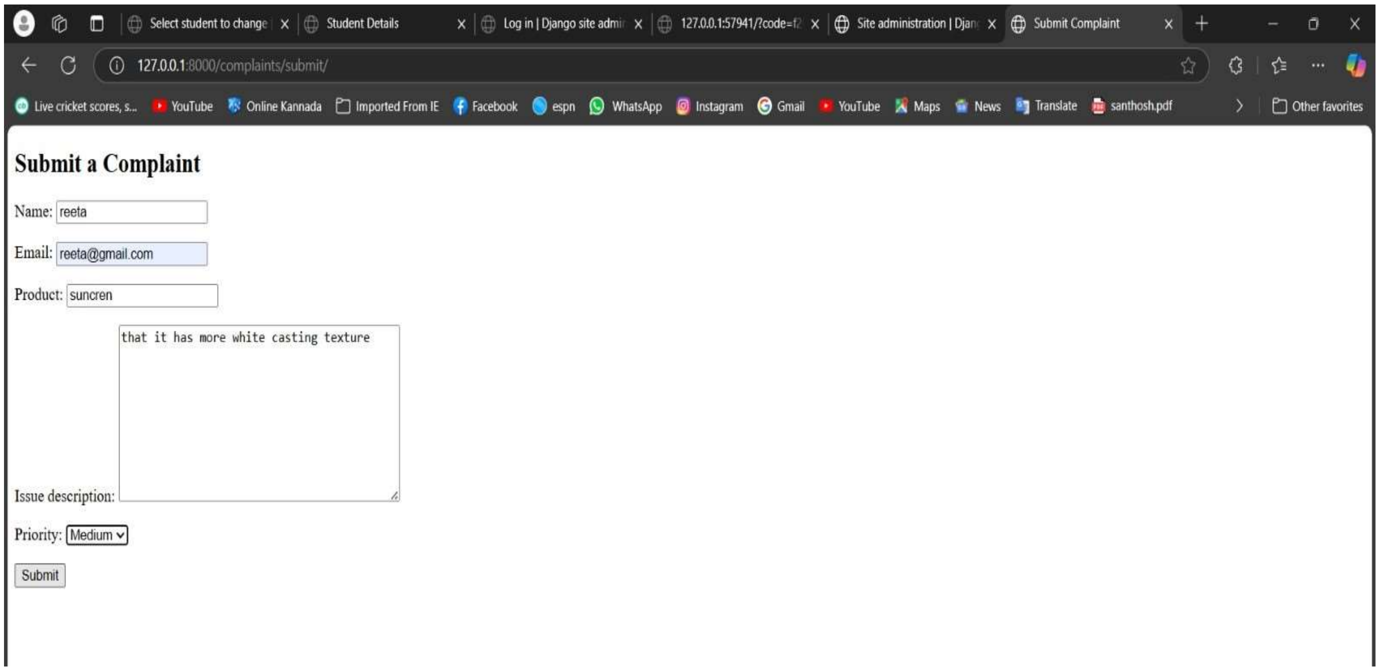
Email: reeta@gmail.com

Product: suncren

Issue description: that it has more white casting texture

Priority: Medium

ii. **The form should include name, email, product, issue description, and priority level (High, Medium, Low) –**


Select student to change | Student Details | Log in | Django site admin | 127.0.0.1:57941/?code=f2 | Site administration | Djan | Submit Complaint

127.0.0.1:8000/complaints/submit/

Live cricket scores, s... YouTube Online Kannada Imported From IE Facebook espn WhatsApp Instagram Gmail YouTube Maps News Translate santhosh.pdf Other favorites

Submit a Complaint

Name: reeta

Email: reeta@gmail.com

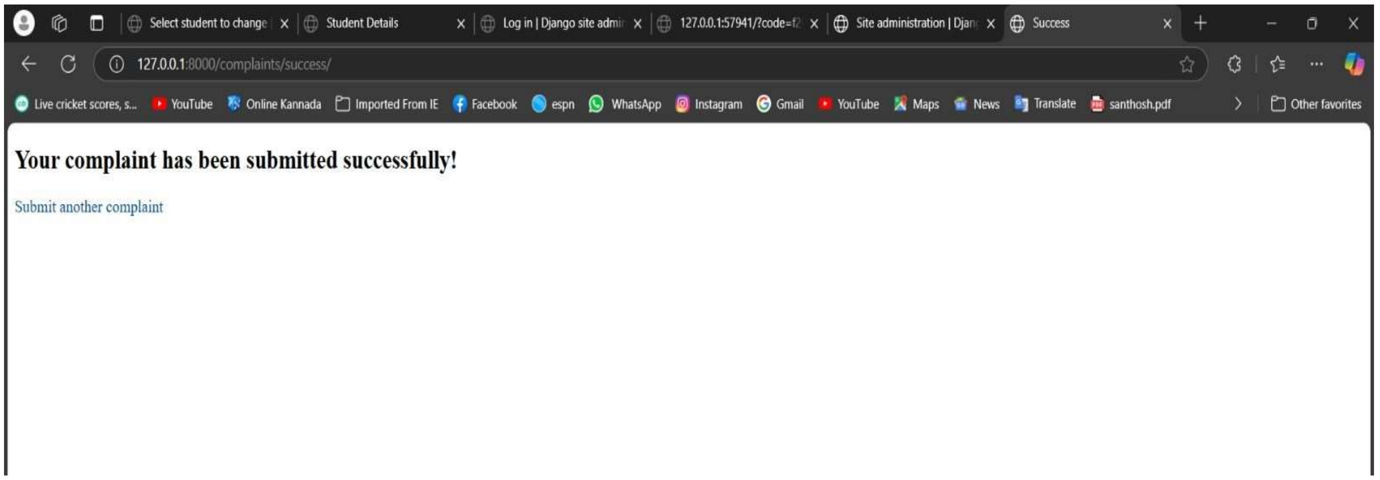
Product: suncren

Issue description: that it has more white casting texture

Priority: Medium

Submit

iii. **The data should be stored in a database and managed via Django Admin –**


Select student to change | Student Details | Log in | Django site admin | 127.0.0.1:57941/?code=f2 | Site administration | Djan | Success

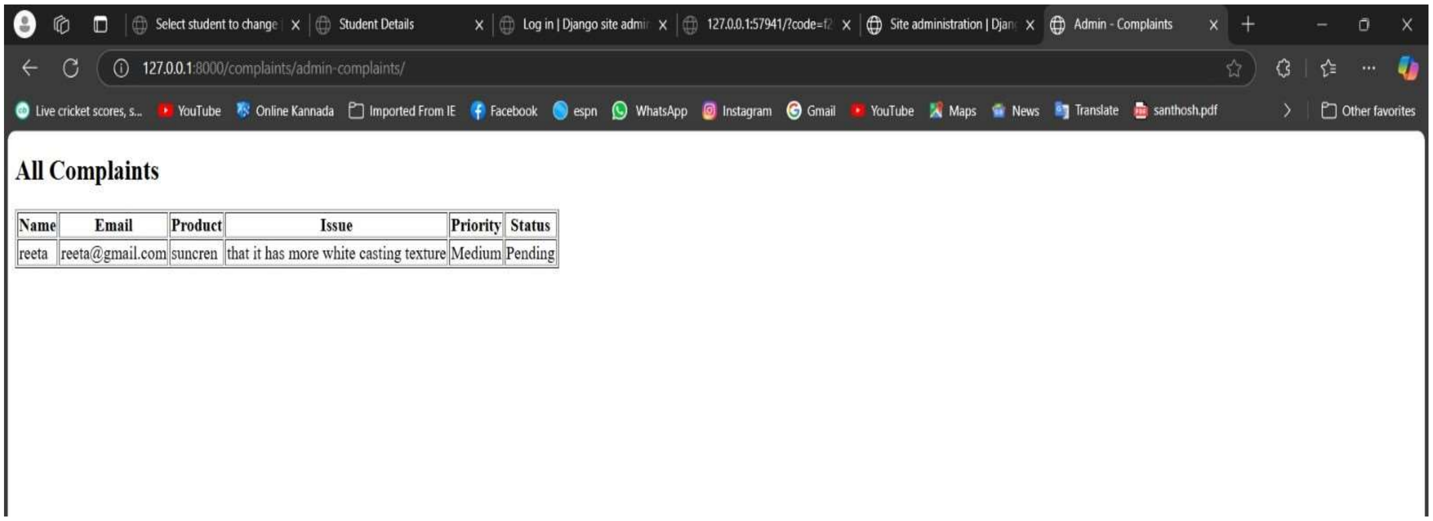
127.0.0.1:8000/complaints/success/

Live cricket scores, s... YouTube Online Kannada Imported From IE Facebook espn WhatsApp Instagram Gmail YouTube Maps News Translate santhosh.pdf Other favorites

Your complaint has been submitted successfully!

Submit another complaint

iv. **Display all complaints on an admin page, allowing the admin to change the status (Pending, Resolved) -**

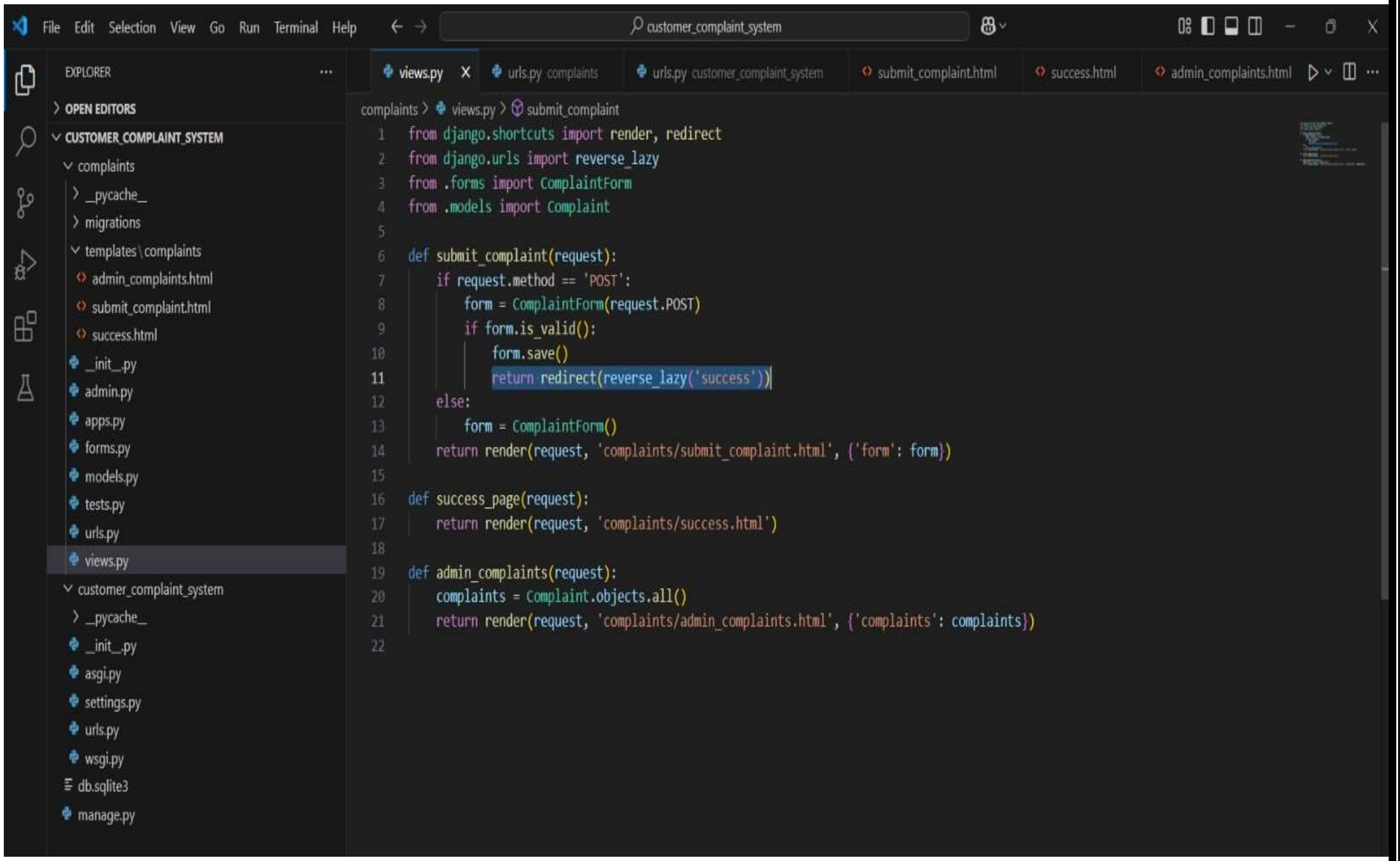


127.0.0.1:8000/complaints/admin-complaints/

All Complaints

Name	Email	Product	Issue	Priority	Status
reeta	reeta@gmail.com	suncren	that it has more white casting texture	Medium	Pending

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



```python
from django.shortcuts import render, redirect
from django.urls import reverse_lazy
from .forms import ComplaintForm
from .models import Complaint

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

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

def admin_complaints(request):
    complaints = Complaint.objects.all()
    return render(request, 'complaints/admin_complaints.html', {'complaints': complaints})
```

Conclusion

The Full-Stack Django Web Application for Customer Complaints is designed to streamline the process of handling customer grievances effectively. It provides a structured approach where customers can submit complaints through a user-friendly form, including essential details like name, email, product, issue description, and priority level. The submitted complaints are stored in a database and managed through the Django Admin panel, allowing administrators to track, update, and resolve issues efficiently. The system enables admins to change the status of complaints (Pending, Resolved), ensuring transparency and better customer service management.A key feature of the application is the use of reverse_lazy(), which ensures smooth redirection of users to a success page after submission. The project utilizes Django's built-in features for form validation, database management, and admin control, making it a robust and scalable solution. The frontend is designed using HTML, CSS, and Bootstrap for a responsive and visually appealing interface. The backend, powered by Django, ensures secure and efficient data handling. By implementing this system, businesses can enhance customer satisfaction by addressing complaints in a systematic manner. The structured workflow improves communication between customers and administrators, reducing response time and increasing efficiency. Additionally, the Django framework provides scalability, making it suitable for future enhancements like email notifications, user authentication, and automated complaint tracking. Overall, this project demonstrates how Django can be leveraged to build a practical, real-world web application that simplifies complaint management and improves customer support operations.