

Table of Contents

1. SRS

1.1. Introduction	3
1.1.1 Purpose	3
1.1.2 Scope	3
1.1.3 Definitions, Acronyms and Abbreviations	3
1.1.4 Overview	3
1.2. Positioning	4
1.2.1 Business Opportunity	4
1.2.2 Problem Statement	4
1.2.3 Product Position Statement	4
1.3. Stakeholder and User Descriptions	5
1.3.1 Market Demographics	5
1.3.2 Stakeholder Summary	5
1.3.3 User environment	5
1.3.4 Key Stakeholder / User Needs	5
1.4. Product Overview	6
1.4.1 Product Perspective	6
1.4.2 Summary of Capabilities	6
1.4.3 Assumptions and Dependencies	6
1.5. Product Features	7
1.5.1 Functional Requirements	7
1.5.2 Non-Functional Requirements	8
1.6. Constraints	9
1.7. Quality Ranges	9
1.8. Other Product Requirements	10
1.8.1 System Requirements	10
1.8.2 Environmental Requirements	10
1.9. Documentation Requirements	11
1.9.1 User Manual	11
1.9.2 Online Help	11
1.9.3 Installation Guides, Configuration, Read Me File	11
1.9.4 Labeling and Packaging	11
1.10. Appendix 1 - Feature Attributes	12
1.10.1 Status	12
1.10.2 Effort	12
1.10.3 Risk	12
1.10.4 Stability	12
1.10.5 Target Release	12
1.10.6 Assigned To	13
1.10.7 Reason	13

2. UML Diagrams

2.1 Use Case Diagram	14
2.2 Activity Diagram	15
2.3 Sequence Diagram	16
2.4 Class Diagram	16
2.5 State Diagram	17

3. Wireframe Screens of the project

3.1 User Side	18
3.2 Admin Side	22

4. Code Screenshots 23

Vision

1.1 Introduction

The main objective of this document is to collect the needs of users and accordingly analyze the features of the Complaint Management System. It focuses on capabilities needed by accountants, employees, management team, account section, and directors, and also focuses on the Reports section. The system is built so that it saves time by storing the information in the database reducing the use of paperwork. The system helps the users to directly register their complaints through the use of a Complaint management website or application.

1.1.1 Purpose

Main purpose of this system is to provide a product that is far better than the current complaint system. It helps management by easy maintenance of all information as we can store all the information into the database instead of storing it in the papers as was done before. This system is easy to use as before complaints were taken at some place where users have to go to register their complaints. So by using this system they can directly register their complaint by using the respective app or website.

1.1.2 Scope

Complaint management system is developed for a society which has keen interest in getting feedback from the members of the society. It provides a mechanism to get complaints / feedback from the members residing in the society by avoiding the paperwork and storing the data dynamically to databases so as a result this reduces the time of fetching the information from paperwork. The system focuses on managing customers' complaints and passes that to the particular department.

1.1.3 Definitions, Acronyms and Abbreviations

IEEE	The Institute of Electrical and Electronics Engineers, Inc.
CMS	Complaint Management System
CMSS	Complaint Management System Software
SRS	Software Requirement Specification

1.1.4 Overview

This document has been prepared in accordance with the IEEE Std. 830, 1998. This document covers requirements, technical information and user manual of the Complaint Management System. It provides the information of Product perspective, Product functions, User characteristics, Constraints, Assumptions and dependencies and specific requirements.

1.2 Positioning

1.2.1 Business Opportunity

The business opportunity for this product is huge in societies where there is feedback and complaints are accepted from the members of the society.

1.2.2 Problem Statement

Regular Complaint management system has lots of drawbacks and problems. For instance, if it is needed to find out a complaint of a customer, then it becomes quite a time consuming job to find out the paper from the physical paper file. If a user files a complaint on paper, the user has to wait a little longer for a reply from the particular authority. And also maintaining the data in the papers is very difficult. So this software will help to store data in the database and can be easily accessed within seconds. Another problem is that to file a complaint within the society members have to go to an authorised person in order to file their respective complaint. So such problems can be solved by providing an online complaint management system.

1.2.3 Product Position Statement

The complaint management system is stand-alone software that manages complaints of the societies. The key points of the Complaint management system are the Complaint registering page and if the complaint is not answered timely, users can again raise a ticket in which higher authority will be informed and action will be taken based on the complaint number and that particular department will be consulted regarding the issue. Another function is to do the analysis for common problems and to provide appropriate solutions.

1.3 Stakeholder and User Descriptions

1.3.1 Market Demographics

Our primary focus is on the users of our products, which are the members of the society which uses any complaint management system software as a platform to take complaints from users. We can say that it can be for the age group like 15-65. Considering the complaints the society committee members can improve their society facilities which would be beneficial for the members of the society.

1.3.2 Stakeholder Summary

CEO/Product Owner	CEO/Product Owner are the ones which are the main stakeholders of the company and they think for the betterment of the company or the product by taking best suitable decisions for the benefit of the users as well as company. Product owners keep keen interest in the process of work and designing and also in engineering to make sure that the system requirements are met according to the needs of the users. He will also make sure that the technology used is the latest one and the flow of product development and its process is also good.
Category of Users	The users are the ones who will use the system of Online complaint management system. In the complaint management system the main users are members of the society.
Admin	The admin user of the complaint management system can manage the user data and manage the solution of the complaints of the users. At the admin side of the complaint management system admin can see how many complaints are registered and how many users are using the system. The list of departments can also be handled from the admin side. So we can say that admin is also the main stakeholder of the complaint management system.

1.3.3 User Environment

The first time the user uses the complaint management system, they will need to register themselves by filling up a sign-up form. Then they will be asked to login by entering to the login page. After logging in to our website User will be able to register its complaint about the particular department. While filing a complaint, users will be asked for the department they are willing to complain to. Then they can select the topic of complaint then the user can file their complaint over there by filling the form. Users can check their old complaints and responses at the dashboard.

1.3.4 Key Stakeholder / User Needs

Nowadays there are too many companies providing too many services but not everyone gets the perfect use of the service as they might have a slight issue / problem with the society's particular service. Then the user will want to complain about that particular problem. The user will also want a seamless system where they can complain about their problem easily. The societies will also want this type of system to improve themselves and resolve their members' problems.

1.4 Product Overview

1.4.1 Product Perspective

The complaint management system is stand-alone software that manages complaints of the society. When any user files a complaint about a particular problem the complaint first goes to the admin department and then goes to the particular department. If they can resolve their complaint over the message they will resolve the complaints over the complaint system by replying to that complaint.

1.4.2 Summary of Capabilities

- Users can complain about any problem any time.
- User will have tracks of their past complaints in the dashboard

1.4.3 Assumptions and Dependencies

The complaint management system is used by the members of the society who can file their complaint by using the complaint management system. The user must understand the system properly in order to file a complaint. They must know the department of their complaint for filing the complaint properly.

1.5 Product Features

1.5.1 Functional Requirements:

1. **Functions:** User Login
User: Any
Input: Username and Password.
Output: If the user name or password is incorrect an error generates. If the user name and password are correct then the user is able to access the functionalities according to their access level.
Results: The user is granted access to functionality according to type of user.
2. **Functions:** Change Password
User: Any
Input: Old password, new password and re-enter new password.
Output: If old password is incorrect or new password and re-enter new password doesn't match or new password doesn't follow password protocol an error generates. If all requirements for a full fill password are set to a new password.
Results: Old password replaced with new password. So, the user has to use a changed password for login next time.
3. **Functions:** Register a User
User: Any
Input: User's name, Phone number, Email, Password, Re-type Password.
Output: If inserted information is valid then user registers to system else generates errors.
Results: Registered user reflects in other modules related to his/her.
4. **Functions:** Raise a complaint
User: User
Input: Username, UID, Complaint category, Date, Complaint title, Complaint Description
Output: If inserted information is valid then a complaint is submitted to the system else it generates errors.
Results: This information is reflected in the complaints module.
5. **Functions:** User dashboard
User: User
Output: User can see their previous complaints and its responses
6. **Functions:** Admin dashboard
User: Admin
Output: Admin can see all complaints and its responses

1.5.2 Non Functional Requirements

1.5.2.1 Security

Priority: High

- The profile details of the users are kept secure in the database and the databases are not allowed to be viewed to everyone
- The password which the user entered is not stored in the database.
- The Admin modules are kept different from the main modules and its accessibilities are given to very limited developers

1.5.2.2 Reliability

Priority: High

- Fault tolerance - The data are stored in multiple places to protect the data in case of any loss of any data from a particular from particular from a particular database. So there is no fear of losing the user data.
- Non-deficiency – degree to which SW does not contain undetected defects

1.5.2.3 Maintainability

Priority: High

- Finding and fixing the bugs if any
- According to the bugs or any default in functionality there patch update will be given
- Change in hardware for improvements
- Addition of new functionality and new modules according to the requirements
- After the changes updating the documentation and again maintaining it

1.5.2.4 Portability

Priority: Medium

- Hardware Independence – Software of this system does not rely on any specific hardware requirements
- Software Independence – Software of this system does not rely on any specific software environment

1.5.2.5 Reusability

Priority: Medium

- Generality – This system can be reuse across another application

1.6 Constraints

- Live tracking of the complaint is not possible here as once the complaint is reached to particular department and they start to work on the problem user can not see the progress regarding their complain
- As a disadvantage due to the online source of complaining there is no source of interacting with the department handler so that the complaint can be expressed properly.

1.7 Quality Ranges

- This system is so robust that with use of this system users can fill up complaints at any time from anywhere and as many as they want. There are lesser chances of system faults.
- There is a security feature about the user that the name of the complainant does not appear to the main authority.

1.8 Other Product Requirements

1.8.1 System Requirements

Minimum Hardware Requirements :

- Laptop / Desktop
- Processor - 1GHz or Above
- Ram - 2GB or Above
- Free Disk Space - 30GB or Above
- Internet Connection

Minimum Software Requirements :

- Eclipse IDE 2021-12 4.0 or higher
- Apache Tomcat Server v10.0 or higher
- JDK 12.0 or higher
- JavaSE-14 or higher
- JRE 6.0.1
- MySQL Workbench 8.0 or higher
- Bootstrap
- MDBootstrap

1.8.2 Environmental Requirements

A periodic backup of the database is suggested so in case one of it crashes or any data loss happens from one database we can have backup in another so there can be no reliability on one database.

1.9 Documentation Requirements

1.9.1 User Manual

User Manual Contents:

- a. How to use the system
- b. How to use each page with screenshots
- c. How to use backup and restoration (in case of database crash and recovery)

1.9.2 Online Help

N/A

1.9.3 Installation Guides, Configuration, Read Me File

As this is a web application there is no installation guide required and also there is no readme file. The interface of the web application is very easy to use and configurations are mentioned above.

1.9.4 Labeling and Packaging

N/A

1.10 Appendix 1 - Feature Attributes

1.10.1 Status

Proposed	Live track of complaint, user verification
Approved	User registration,user login,forget password, raise a complaint, \User Dashboard
Incorporated	User registration,user login,forget password, raise a complaint, live track of complaint, user verification,User Dashboard

1.10.2 Effort

The development of the project has been done by the joint effort of all the team members Jinay Shah and Purav Shah. Without the team it would not be possible to complete the project. The work done together with equal efforts of each one for example system analysis, document analysis, use case study, software coding, database designing, software testing etc.

1.10.3 Risk

- cost is also increased as increased hardware or/and software cost is increased due to licence version.
- It may take time to complete a project. So delay will be generated to release a software
- Some bugs will not be solved

1.10.4 Stability

In the process of development of a project if the client asks about the change in information or any addition in the project the changes may be done accordingly but not in between the ongoing sprints but after the work with higher priority is completed.

1.10.5 Target Release

The final release will allow users to signup, login and raise complaints. It will allow the users to signup,login in order to raise a complaint about a particular department. Users will get the response from the responsible department over here only. Users can check their previous complaints and their responses in the dashboard area. The Complaint management system's final launch will be approximately after 4 months of developing stage and it will get updated further by regular updates.

1.10.6 Assigned To

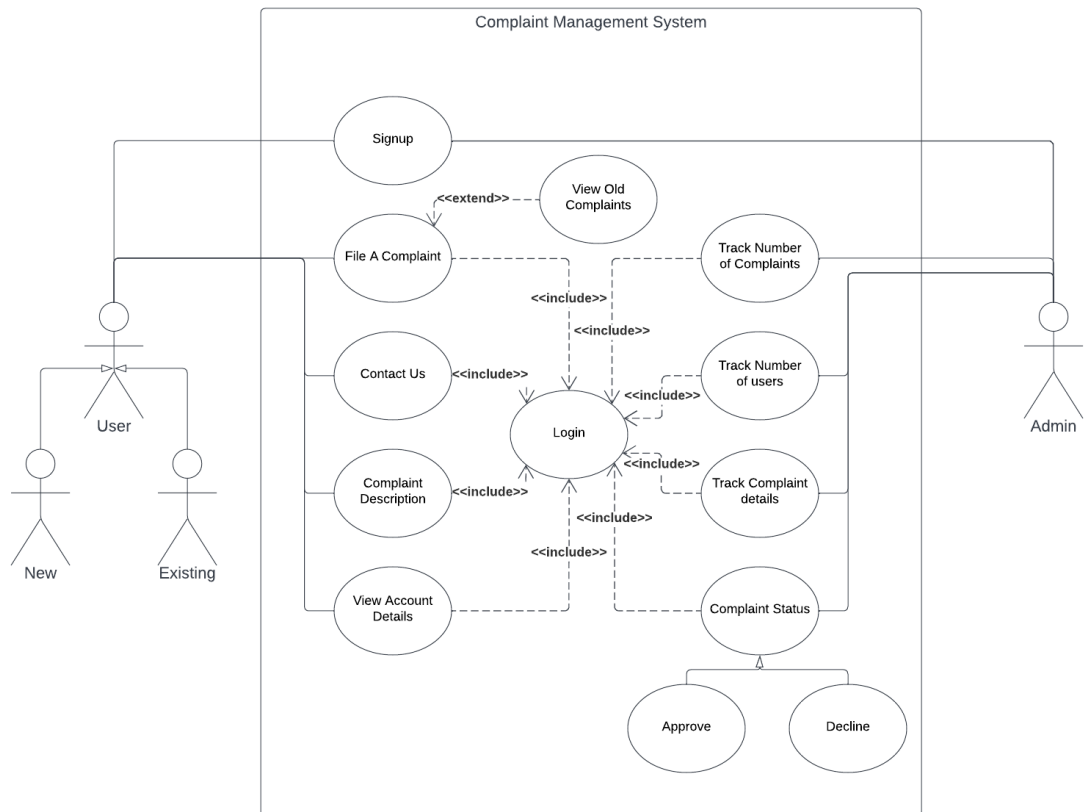
Project Management Team	This team will ensure proper flow of the project based on time and work
Research & Analytics Team	This team will do the user research and requirement analysis
Frontend Designer Team	They will design the UI/UX of project for better experience
Frontend Developer Team	They will develop the frontend based on the designs of the Frontend Designer Team.
Backend Designer Team	They will construct & design the system architecture and database designs
Backend Developer Team	They will develop the backend on the basis of the system requirements given by the Backend Designer Team
Software Testing Team	They will ensure the testing of the product prior to launching. They will make sure that no potential bugs are present
DevOps Team	They will deploy the final product to the servers and maintain the integrity of the product.

1.10.7 Reason

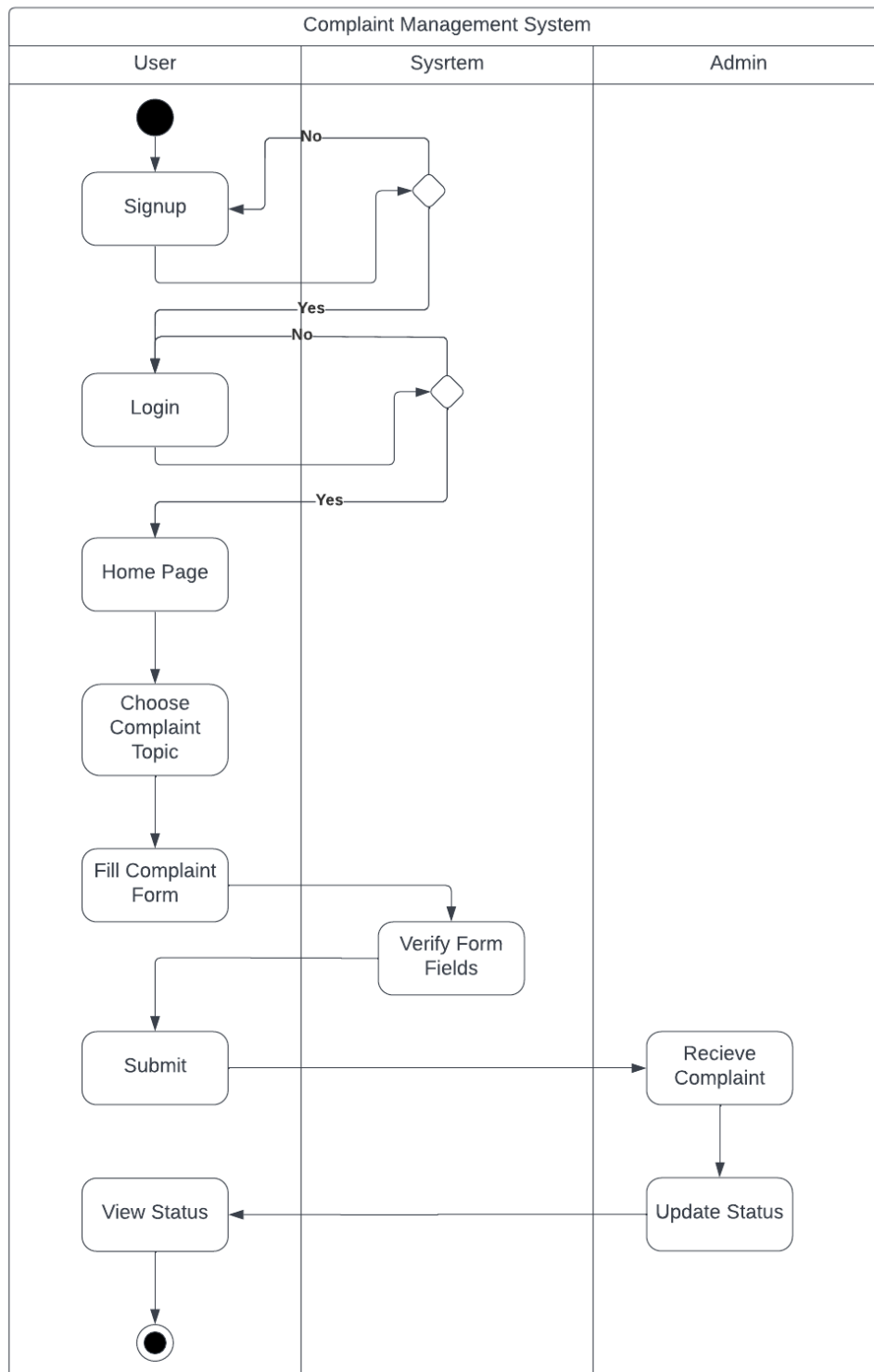
The main reason for this document is to understand the needs of clients and tools and technology and basic understanding of the project. This document will give the flow of how the project will be implemented, how the project will be and what is the main purpose of the project.

2. UML Diagrams:

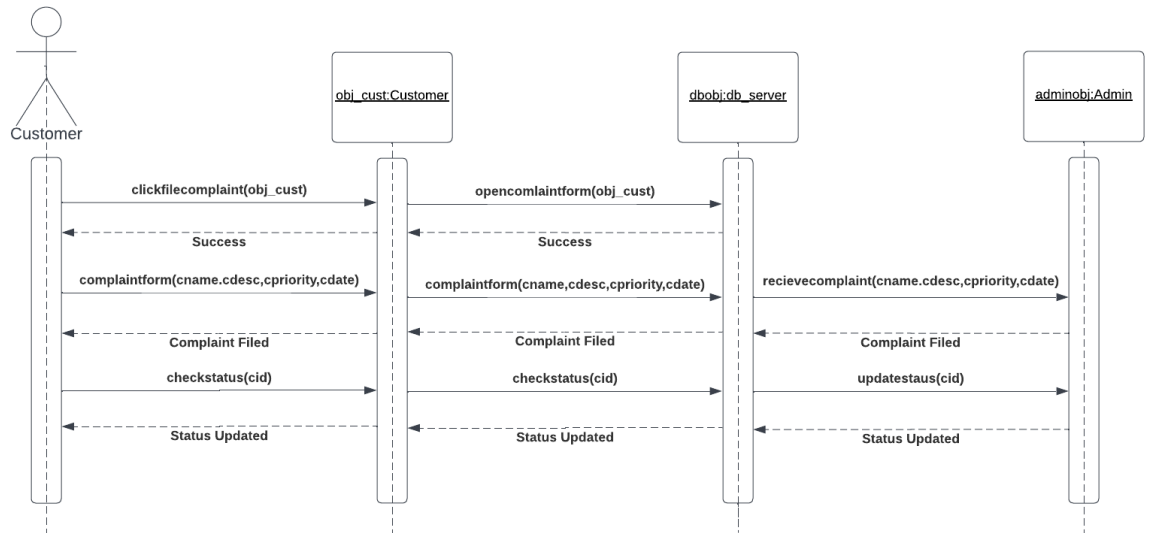
2.1 Use Case Diagram



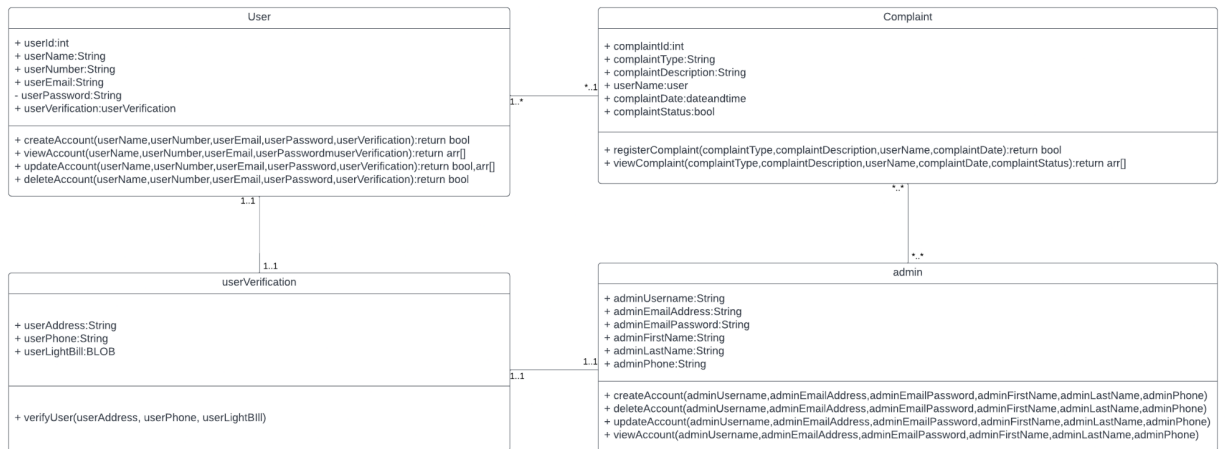
2.2 Activity Diagram



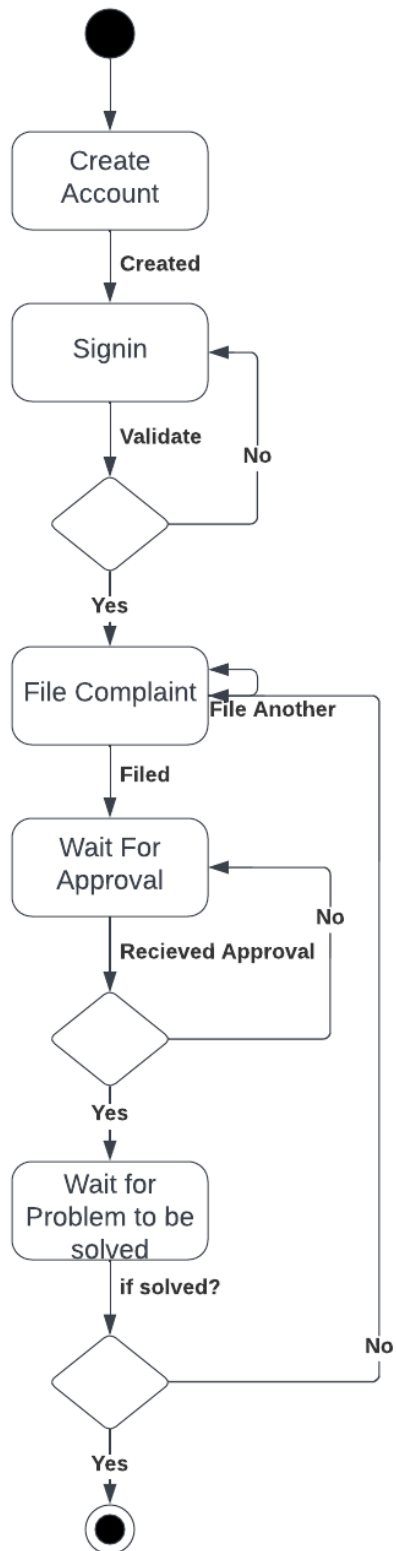
2.3 Sequence Diagram



2.4 Class Diagram



2.5 State Diagram



3. Wireframe Screens of the project

3.1 User Side:



Login To Continue

Forgot Password?

Don't have an account? [Create new](#)



Sign up

Already User? [Login](#)

Complaint Management System[Home](#)[Dashboard](#)[Contact us](#)[Account](#)[Logout](#)

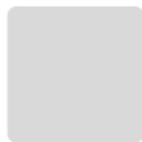
Handling customer complaints is easy now.

We helps you receive, respond to, and resolve customer grievances with simple solutions, and leaves you with more time for tasks that matter.

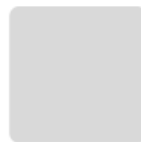
Hello Jinay, do you have a complaint? Raise your ticket by filling the respective form.



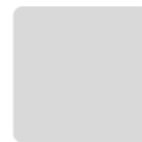
Water



Light



Cleanliness



Other

Complaint Management System[Home](#)[Dashboard](#)[Contact us](#)[Account](#)[Logout](#)

File a Complaint

Complaint Name

Complaint Description

Priority Of Complaint
☐ High ☐ Moderate ☐ Low

Enter Date

File complaint

Complaint Management System[Home](#)[Dashboard](#)[Contact us](#)[Account](#)[Logout](#)

Dashboard

Track of your Complaints



Complaint Management System[Home](#)[Dashboard](#)[Contact us](#)[Account](#)[Logout](#)

Complaint Description

Sr No	Complaint Type	Comaplain Description	Date	Status
1	Water	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	18-09-2022	Completed
2	Light	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	5-10-2022	Pending
3	Other	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	6-10-2022	Pending

Complaint Management System	Home	Dashboard	Contact us	Account	Logout
-----------------------------	------	-----------	------------	---------	--------

Contact - Us

Your name

Your Phone number

Your Email

Your message

Submit

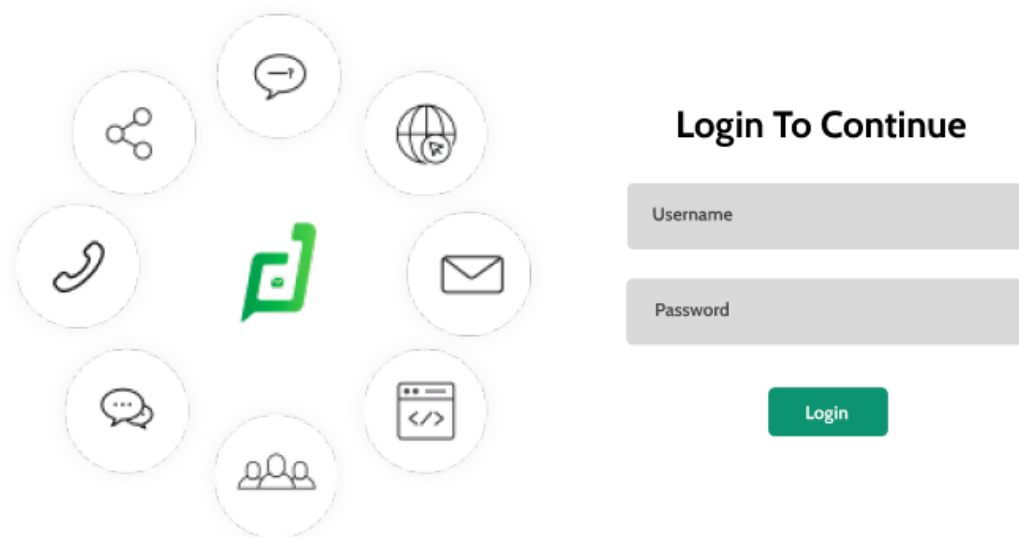
Complaint Management System	Home	Dashboard	Contact us	Account	Logout
-----------------------------	------	-----------	------------	---------	--------

User Account



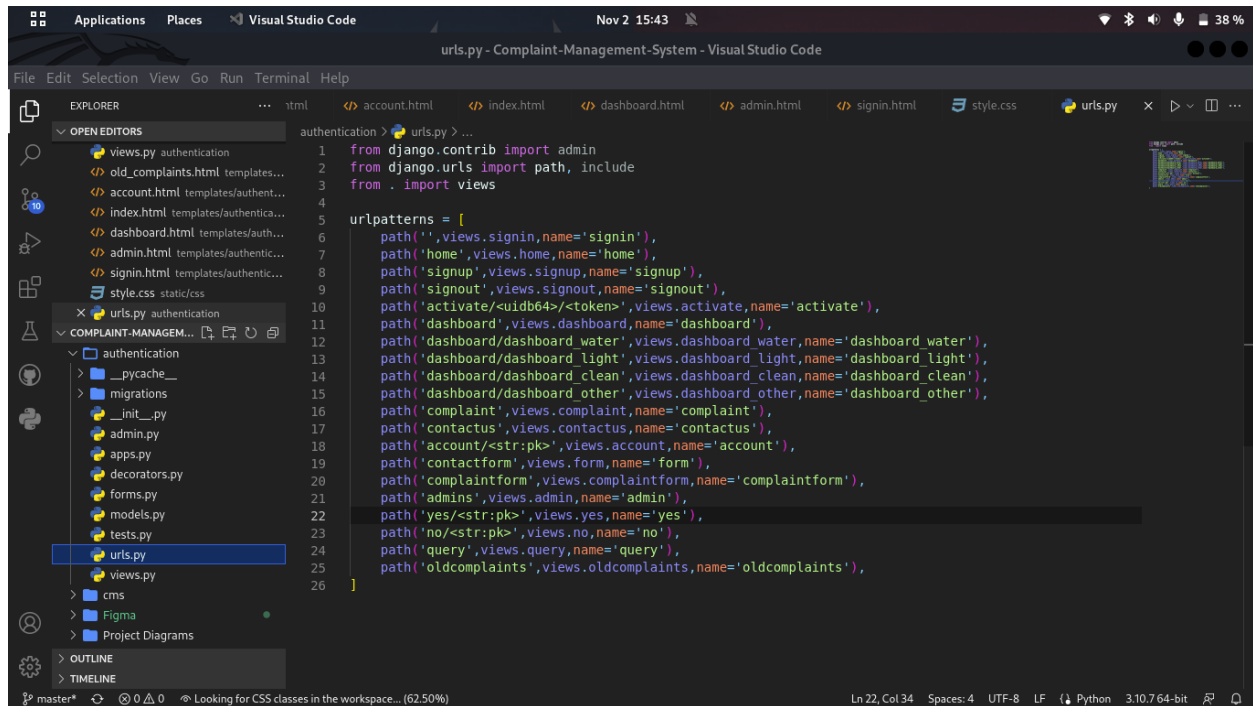
Full Name	Jignesh
Email	jignesh01@gmail.com
Phone	1234567890
House No	A-101

3.2 Admin Side:



Complaint Management System					Logout	
Total Users		Total Complaints		Complaints Completed		Complaints Pending
56		103		73		30
Sr No	Username	Complaint Type	Complaint Description	Date		
1	Jignesh	Water	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	18-09-2022	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Jigo	Light	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	5-10-2022	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Jigi	Other	Lorem Ipsum is simply dummy text of the printing and typesetting industry.	7-10-2022	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4. Code Screenshots

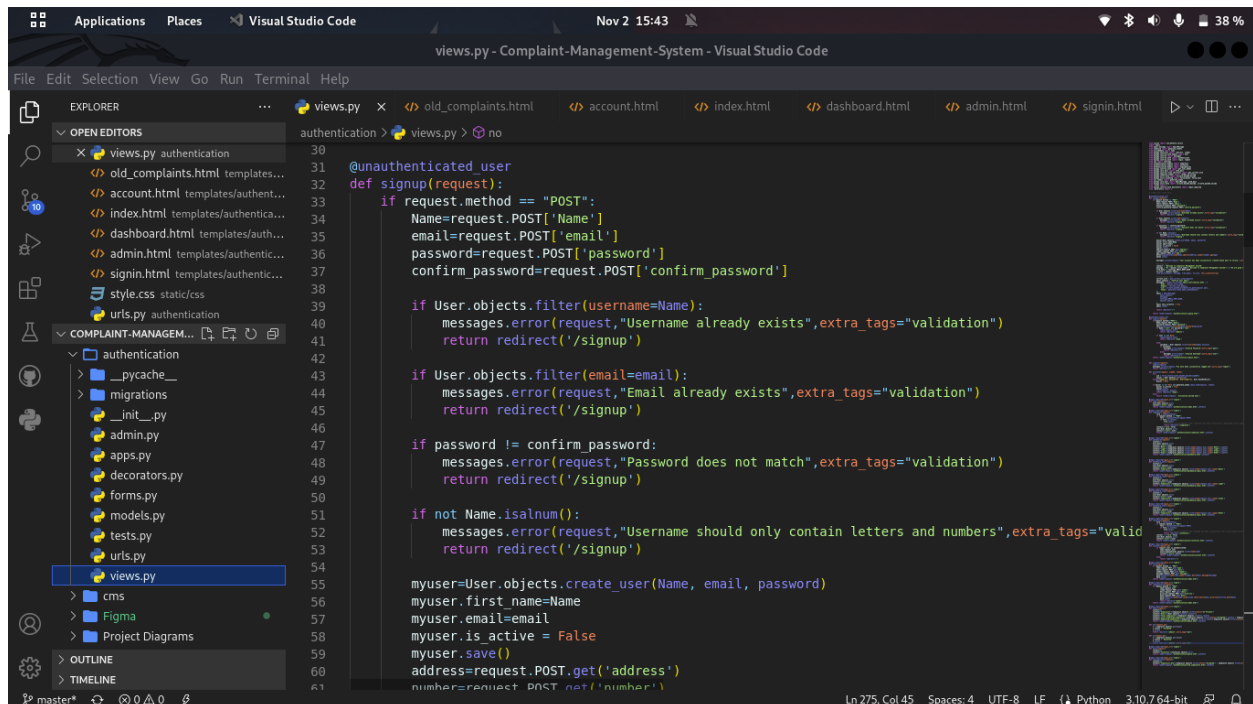


```

urls.py - Complaint-Management-System - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    authentication > urls.py > ...
  COMPLAINT-MANAGEM...
    authentication
      __pycache__
      migrations
      _init_.py
      admin.py
      apps.py
      decorators.py
      forms.py
      models.py
      tests.py
      urls.py
      views.py
    cms
    Figma
    Project Diagrams
  OUTLINE
  TIMELINE
  master* 0 0 0 Looking for CSS classes in the workspace... (62.50%) Ln 22, Col 34 Spaces: 4 UTF-8 LF Python 3.10.7 64-bit

1 from django.contrib import admin
2 from django.urls import path, include
3 from . import views
4
5 urlpatterns = [
6     path('', views.signin, name='signin'),
7     path('home', views.home, name='home'),
8     path('signup', views.signup, name='signup'),
9     path('signout', views.signout, name='signout'),
10    path('activate/<uidb64>/<token>', views.activate, name='activate'),
11    path('dashboard', views.dashboard, name='dashboard'),
12    path('dashboard/dashboard_water', views.dashboard_water, name='dashboard_water'),
13    path('dashboard/dashboard_light', views.dashboard_light, name='dashboard_light'),
14    path('dashboard/dashboard_clean', views.dashboard_clean, name='dashboard_clean'),
15    path('dashboard/dashboard_other', views.dashboard_other, name='dashboard_other'),
16    path('complaint', views.complaint, name='complaint'),
17    path('contactus', views.contactus, name='contactus'),
18    path('account/<str:pk>', views.account, name='account'),
19    path('contactform', views.form, name='form'),
20    path('complaintform', views.complaintform, name='complaintform'),
21    path('admins', views.admin, name='admin'),
22    path('yes/<str:pk>', views.yes, name='yes'),
23    path('no/<str:pk>', views.no, name='no'),
24    path('query', views.query, name='query'),
25    path('oldcomplaints', views.oldcomplaints, name='oldcomplaints'),
26 ]

```



```

views.py - Complaint-Management-System - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    authentication > views.py > no
  COMPLAINT-MANAGEM...
    authentication
      __pycache__
      migrations
      _init_.py
      admin.py
      apps.py
      decorators.py
      forms.py
      models.py
      tests.py
      urls.py
      views.py
    cms
    Figma
    Project Diagrams
  OUTLINE
  TIMELINE
  master* 0 0 0 0 Ln 275, Col 45 Spaces: 4 UTF-8 LF Python 3.10.7 64-bit

30
31 @unauthenticated_user
32 def signup(request):
33     if request.method == "POST":
34         Name=request.POST['Name']
35         email=request.POST['email']
36         password=request.POST['password']
37         confirm_password=request.POST['confirm_password']
38
39         if User.objects.filter(username=Name):
40             messages.error(request,"Username already exists",extra_tags="validation")
41             return redirect('/signup')
42
43         if User.objects.filter(email=email):
44             messages.error(request,"Email already exists",extra_tags="validation")
45             return redirect('/signup')
46
47         if password != confirm_password:
48             messages.error(request,"Password does not match",extra_tags="validation")
49             return redirect('/signup')
50
51         if not Name.isalnum():
52             messages.error(request,"Username should only contain letters and numbers",extra_tags="validation")
53             return redirect('/signup')
54
55         myuser=User.objects.create_user(Name, email, password)
56         myuser.first_name=Name
57         myuser.email=email
58         myuser.is_active = False
59         myuser.save()
60         address=request.POST.get('address')
61         number=request.POST.get('number')

```

```

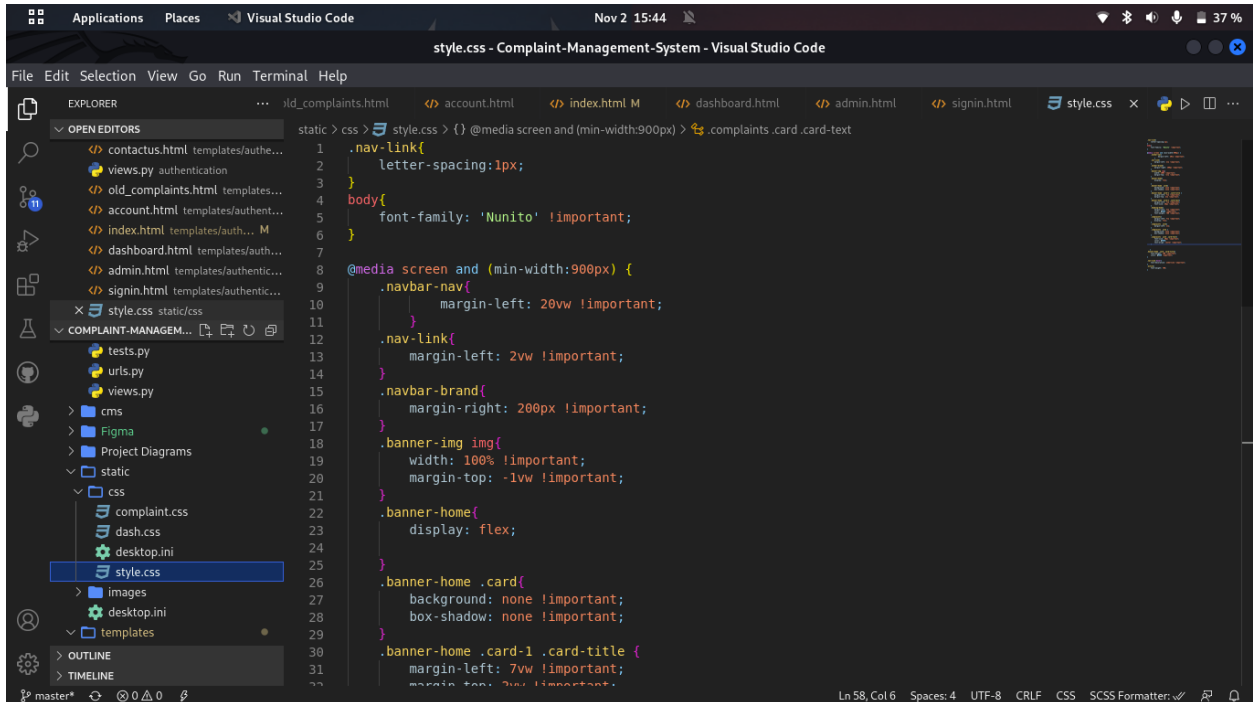
10
11 class Complaint(models.Model):
12     name = models.CharField(max_length=100,null=True)
13     cname = models.CharField(max_length=100)
14     cdescription = models.TextField()
15     CHOICES=[('High','High'),('Moderate','Moderate'),('Low','Low')]
16     priority = models.CharField(max_length=10,choices=CHOICES)
17     date = models.DateField(auto_now_add=True)
18     status=models.CharField(default='In Process',max_length=10)
19
20     def __str__(self):
21         return self.name,self.cname,self.cdescription,self.priority,self.status
22
23 class Contactus(models.Model):
24     name = models.CharField(max_length=100,null=True)
25     number = models.CharField(max_length=10,null=True)
26     mail = models.EmailField(max_length = 254,null=True)
27     message = models.TextField(null=True)
28
29     def __str__(self):
30         return self.name,self.number,self.mail,self.message
31
32 class SignupFields(models.Model):
33     name = models.CharField(max_length=100,null=True)
34     address=models.CharField(max_length=100,null=True)
35     number=models.CharField(max_length=10,null=True)
36     age=models.IntegerField(null=True)
37
38     def __str__(self):
39         return self.name,self.address,self.number,self.age
40
41

```

```

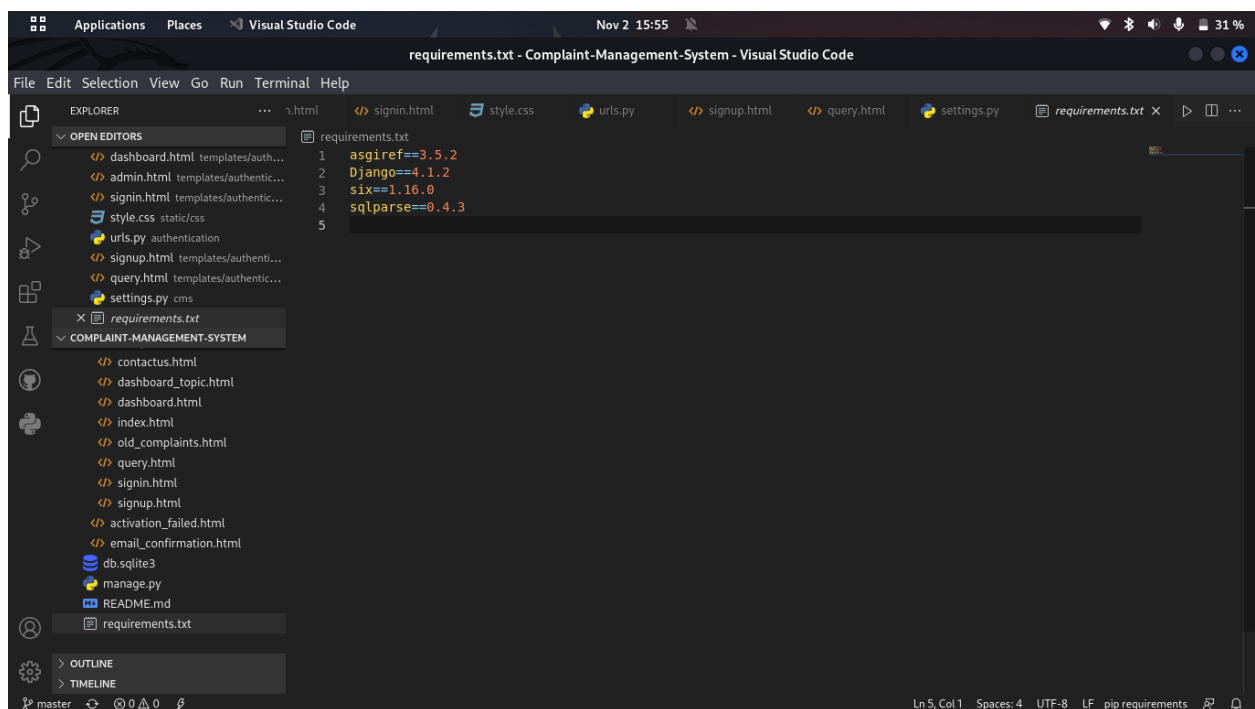
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8" />
6     <meta name="viewport" content="width=device-width" , initial-scale="1.0" />
7     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
8     <title>Home - Complaint Management</title>
9     <script src="https://kit.fontawesome.com/36a2be57d1.js" crossorigin="anonymous"></script>
10    <link href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/5.0.0/mdb.min.css" rel="stylesheet" /
11    <link rel="stylesheet" type="text/css" href="../../static/css/style.css">
12    <link href="https://fonts.googleapis.com/css2?family=Nunito:ital,wght@0,200;0,300;0,400;0,500;0,600
13    <link href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@400;600;700;800;900&display=s
14
15    <link rel="icon" href="../../static/images/favicon.png">
16
17 </head>
18
19 <body>
20     {% for i in users %}
21
22     <nav class="navbar fixed-top navbar-expand-lg navbar-light bg-light shadow-0">
23         <div class="container-fluid">
24             <button class="navbar-toggler" type="button" data-mdb-toggle="collapse"
25                 data-mdb-target="#navbarTogglerDemo01" aria-controls="navbarTogglerDemo01" aria-expande
26                 aria-label="Toggle navigation">
27                 <i class="fas fa-bars"></i>
28             </button> <a class="navbar-brand" href="/home"
29                 style="letter-spacing:0px;margin-left:100px !important;">
32

```

The screenshot shows the Visual Studio Code interface with the 'style.css' file open. The Explorer sidebar on the left shows the project structure, including folders like 'static', 'css', 'images', and 'templates'. The 'style.css' file is selected in the 'css' folder. The main editor displays the following CSS code:

```
static > css > style.css > {} @media screen and (min-width:900px) > .complaints_card .card-text
1  .nav-link{
2    letter-spacing:1px;
3  }
4  body{
5    font-family: 'Nunito' !important;
6  }
7
8  @media screen and (min-width:900px) {
9    .navbar-nav{
10     margin-left: 20vw !important;
11   }
12   .nav-link{
13     margin-left: 2vw !important;
14   }
15   .navbar-brand{
16     margin-right: 200px !important;
17   }
18   .banner-img img{
19     width: 100% !important;
20     margin-top: -1vw !important;
21   }
22   .banner-home{
23     display: flex;
24   }
25   .banner-home .card{
26     background: none !important;
27     box-shadow: none !important;
28   }
29   .banner-home .card-1 .card-title {
30     margin-left: 7vw !important;
31     margin-top: 2vw !important;
32   }
```



The screenshot shows the Visual Studio Code interface with the 'requirements.txt' file open. The Explorer sidebar on the left shows the project structure, including folders like 'static', 'css', 'images', and 'templates'. The 'requirements.txt' file is selected in the 'static' folder. The main editor displays the following requirements:

```
requirements.txt
1 asgiref==3.5.2
2 Django==4.1.2
3 six==1.16.0
4 sqlparse==0.4.3
5
```