Database Management System (IT214)

Date:- 23/11/2022 Team: 3.8

Lab 10 Final Report

Dhananjay Vora
(202001417)

Kalp Pandya
(202001466)

Group No: 1
Section No: 3

Index

Software Requirements Specification

for

District Management System

Version 1.0 approved

Prepared by Dhananjay Vora, Kalp Pandya

DAIICT

29th September 202

Section 1: SRS documentation

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
3.8_Lab4_G1_S3_ V1	25th September 2022	Introduction	V1
3.8_Lab5_G1_S3_ V1	29th September 2022	Fact-Finding Phase: Background reading, Interview, Questionnaire	V1
3.8_Lab6_G1_S3_ Final_SRS	6th October 2022	Fact-Finding Chart, Requirements, User Classes and Privileges, Assumptions and Business Constraints	Final

1. Introduction

1.1 Purpose

The District Management System aims to keep track of user complaints for different districts. The main purpose of this system must be to provide complaint resolution to users as soon as possible keeping the procedure simple. Users can re-issue the complaint, if that was not addressed properly.

System includes streamlined functioning and supervision of the public services provided in the administration of a district like education, health, agriculture, electricity, drinking water and sanitation, transport service and road quality, law and order, etc.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document includes Software developers, All the citizens of a district who are going to use this software to address their issues, Employees at the tehsil administrative level who will go through these issues, resolve it, and update the status of the issue, Employees at the district administrative level who will foresee different tehsils that fall under a district and finally the Law Enforcement Agencies (which falls under the jurisdiction of the tehsil) will be called upon by the tehsils to resolve the issues.

This project is being developed under the guidance of Prof. Minal Bhise and Prof. Rachit Chhaya. This project is useful for the district administrative services and the citizens of the district.

The reader of the document should go through the purpose, product scope and description from the introduction section. Then from the fact-finding phase section read the background readings, requirements gathered, agenda of the interview, questionnaire. Then through fact-finding chart, requirements, user categories and privileges, assumptions and finally business constraints.

1.3 Product Scope

The District Management System project entails the creation of a website that allows the users (citizens of a district) to register a complaint in their respective tehsil's administrative department.

In the current times, the government processes are very complex and it is difficult for any citizen to understand the system and need to visit the administrative offices for the update on their query. But through this system users only need to register a complaint once in the respective category and can receive the update which will be easily accessible via message, mail, or WhatsApp. This system needs to operate on a basic framework so that the user can grasp it.

The system aims to keep track of user complaints for different districts. The administrative dept includes employees that foresee different tehsils that fall under a district. There are multiple tehsils under a district, and every tehsil has its citizens' data.

The citizen can register a complaint in their tehsil's administrative department. If required, the tehsil can employ law enforcement agencies available under their jurisdiction to resolve the issue. The users should be able to track the status of their complaints. Users can re-issue the complaint, if that was not addressed properly.

The system also includes capabilities such as user registration online, information of tehsildars to resolve the issues, information of district level employees who supervise the smooth functioning of the district administration. The respected users can edit their specific details, and the administrator can add, delete, or change users' information and status of their addressed issues.

1.4 Description

The District Management System is a self-contained software product that replaces the current manual and visiting processes for tracking the status of users' registered complaints. The Software Requirements Specification provides a detailed description of the requirements for the project "District Management System". This SRS will allow the readers to understand what to expect from this system and how to approach and create this system step by step. The clear understanding of the system and its functionality will allow the readers to successfully develop the project. From this SRS, the District Management System can be designed, constructed, and finally tested.

This software shall ensure hassle-free registration of a particular complaint with citizens being able to track their complaint. The Tehsildar (Tehsil level administrator) looks into resolving the complaints of the citizens and to update the status of the complaints. The Sub-Division Officer (District level administrator) keeps an eye on the functioning of the tehsil administrative body. The Law Enforcement Agencies (which falls under the jurisdiction of the tehsil) will be called upon by the tehsils to resolve the issues.

The various functions available to the citizens are registering complaints, tracking status of complaints and reopening of the complaints if they are not addressed properly. A place for feedback shall also be built where citizens can give their reviews on their satisfaction with their complaints solved. Quick solving of complaints are to be provided by the tehsildars to the citizens.If required, the tehsil can employ law enforcement agencies available under their jurisdiction to resolve the issue. Citizens are provided with an online help platform which solves their queries and directs them to the right place. In order to ensure security, only authorized people are allowed to make changes in the software. The Sub-Divisional Officer can keep track of the issues undertaken by the tehsils.

1.5 References

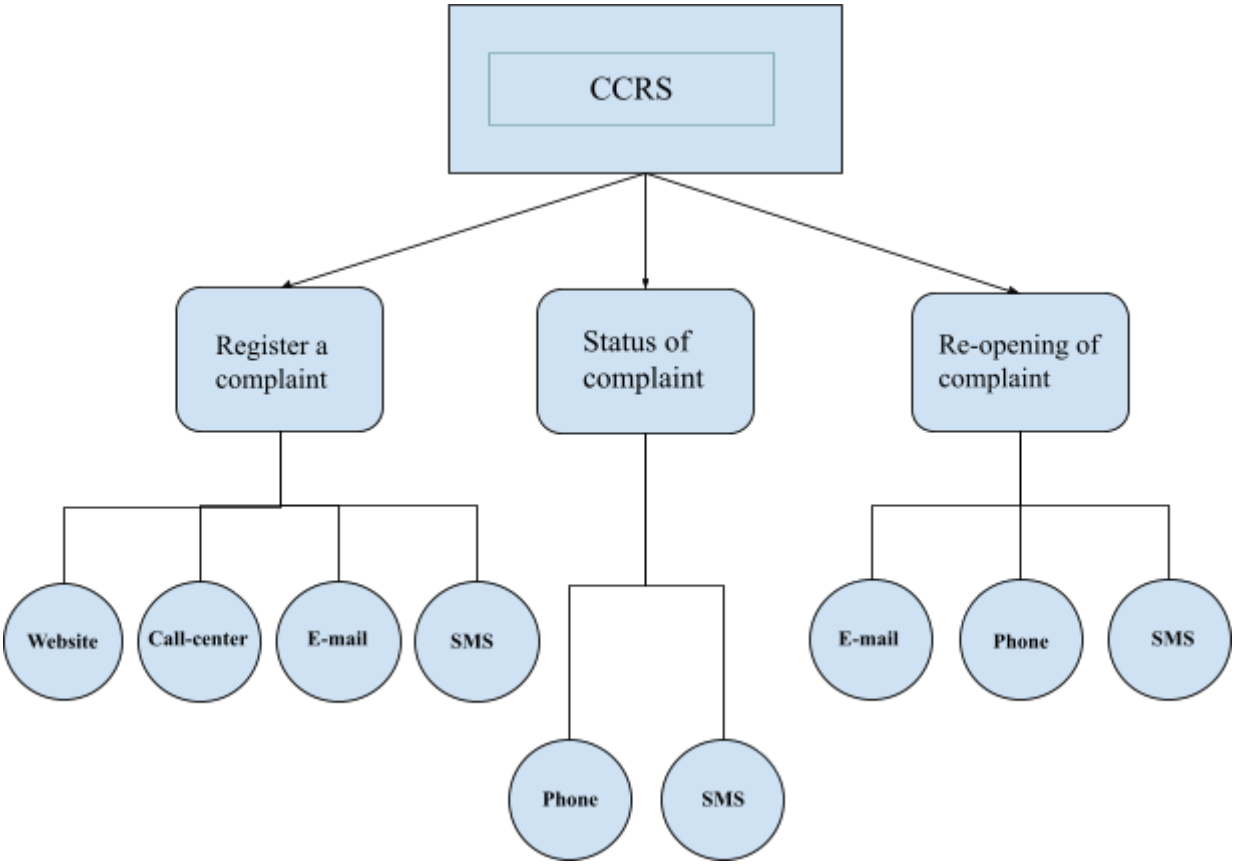
[1] *Administrative Set-up*

2. Fact-Finding Phase

2.1 Background Readings

Case Study 1 - AMC (Ahmedabad Municipal Corporation)

- AMC has kept CCRS department for complaint resolution of citizens of Ahmedabad. CCRS stands for Comprehensive Complaint Redressal System.
- The CCRS is a web-based enterprise solution that allows Municipality/Corporation to enhance citizen satisfaction through comprehensive service management and efficient service delivery.
- Benefits of CCRS:
    - Citizens need not go to Ward offices to register complaint
    - System available round the clock - Call Center, Website, SMS, Email, IVR, Ward Offices, Mobile Application
    - Improved communication by way of SMS / Email alerts
    - CCRS provides data of frequent complaints and average turnaround time for each kind of problem. It helps Municipality/Corporation to focus the areas to improve the services by enhancing the manpower and infrastructure. This also helps Municipality/Corporation to prioritize their services.
- Workflow of CCRS:



- Features provided by CCRS
  - Status of complaint through phone and SMS.
  - Reopening of complaint: Citizens can reopen a complaint within 3 days of closure, if that was not addressed properly.
    - Reopening can be achieved through email, phone or SMS.

References

[AMC CCRS Service]

Requirements Gathered

- Providing information about the tehsils and tehsildars to the citizens for addressing particular issues to the respected tehsils.
- Also, a database to store information about the citizens previous complaints and their
- feedback.
- An effective flowchart should be made at the implementation stage of the district management system.
- Complaint registration should be possible online, and there should be more than one option available.
- One subsection must be kept for providing the updates.
- An individual section should be kept which handles reopening of complaints. So that if a citizen is not satisfied with the problem resolution he can reopen that complaint.

Case Study 2 – BMC (Bhavnagar Municipal Corporation)

- Employ information on basis of their sections
  - BMC has provided well organized employee data grouped by their section.

BMC Official

Sr. No	Name	Designation	Office	Mobile	Email
Commissioner Office					
1	Shri N.V.Upadhyay, IAS	Municipal Commissioner	2439900/2510532		comm-bmc@gujarat.gov.in
2	Shri M.R.Brahmbhatt	Deputy Municipal Commissioner (Admin)	2439292	9925224646	dmca.bmcgujarat@gmail.com
3	Shri J.M.Sompura	I/C Deputy Municipal Commissioner (General)	2439797	9879572323	-
4	Shri F.M.Shah	I/C. Assistant Municipal Commissioner	2424801 TO 10	9979945333	-
5	Shri R.H.Solanki	I/C. PA to Commissioner	2439900/2510532	9825780702	-
Secretary Department					
6	Shri G.J.Patel	Secretary	2422652	9879241493	-
7	Shri J.K.Vegad	Deputy Secretary	2511600	9879547888	-
8	Shri P.D.Patel	Deputy Secretary	2424843	9825412641	-
9	Shri N.H.Pandya	PA to Mayor	2511600	9909048787	-

References

[BMC employee information]

Requirements Gathered

- Details of employees at Tehsil level of different categories must be shared with citizens so that they can easily contact them.
- The website/application must be made in language that all district members can understand. For example, some parts of the BMC website was written in English and some was in Gujarati. Instead, the whole website should be made in one particular language.

2.2 Interview

System:
DistrictManagement System

Participants:

- Kalp Pandya
- Dhananjay Vora

Date: 3rd Oct 2021
Time: 17:00
Duration: 45minutes

Place: DAIICT

Purpose of Interview: Meeting to determine the design and requirements for a District Management System.

Agenda

- Security is of utmost importance to our System.
- The system must only allow privileged users access in order to protect sensitive data.
- Citizens should be able to easily use well-designed, error-free, and simple application.
- The application must be portable, and unauthorized individuals must not use it or have access to it.
- Significant capacity for data storage
- Recurrent maintenance and updating.
- Correct assessment of citizens feedback or complaint reopening.

2.3 Questionnaire

District Management System (Group 3.8)

Survey for creating online complaint management system

* Required

1. Your name

2. State Name *

3. District Name *

4. Select zone of your Tehsil *

Mark only one oval.

- East
- West
- North
- South
- Center

5. Are you a permanent resident of district entered above? *

Mark only one oval.

- Yes
- No
- May be

6. In total How many complaints have you registered in past year? *

7. How many complaints got solved? *

8. How many were solved in satisfied time? *

9. Select level of satisfaction of solution *

Mark only one oval.

	1	2	3	4	5	
Not at all satisfied	◯	◯	◯	◯	◯	Fully satisfied

10. Was the registering complaint's process simple? *

Mark only one oval.

- ◯ Yes
- ◯ No

11. Was there any discrimination while registering your complaint? *

Mark only one oval.

- ◯ Yes
- ◯ No

12. Select categories your complaints *

Check all that apply.

- ☐ Transport service and road quality
- ☐ Social Discrimination
- ☐ Hospital service
- ☐ Police related
- ☐ Electricity
- ☐ Loud noise at inappropriate time
- ☐ Water supply
- ☐ Educational system
- ☐ Garbage and Cleanliness
- ☐ Other: _____

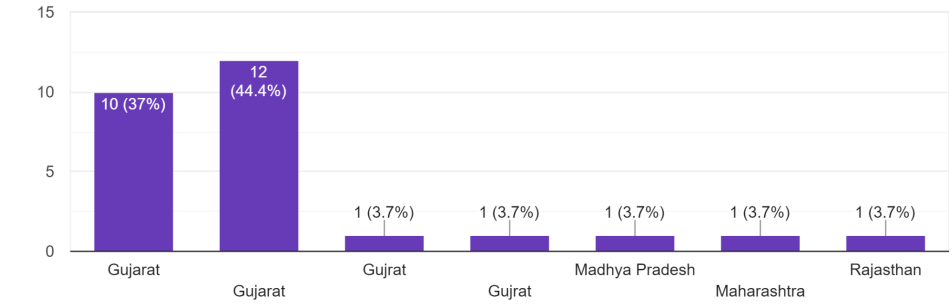
13. Thank you for filling this form and helping in this survey. Your input will be great help in designing the online district complaint management system.

Any suggestion or feature which should be added in project to make system more simpler and fast.

- **Summary**

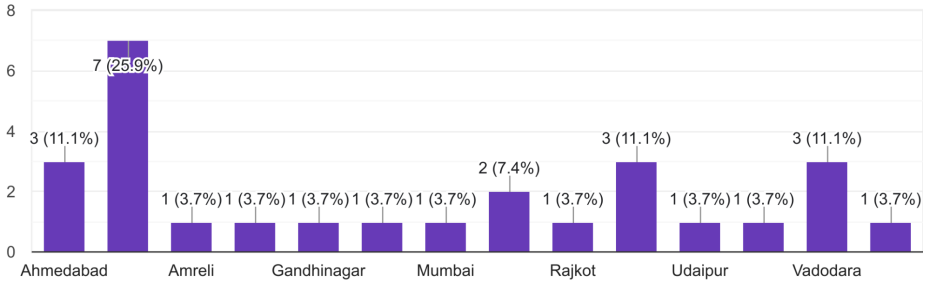
State Name

27 responses



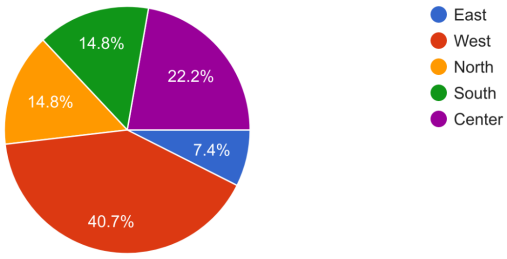
District Name

27 responses



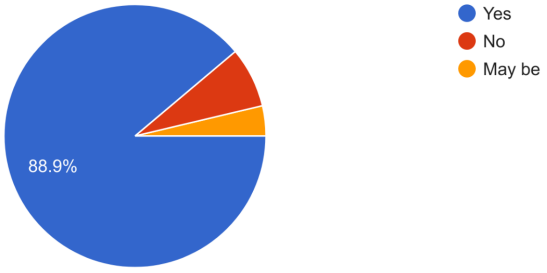
Select zone of your Tehsil

27 responses



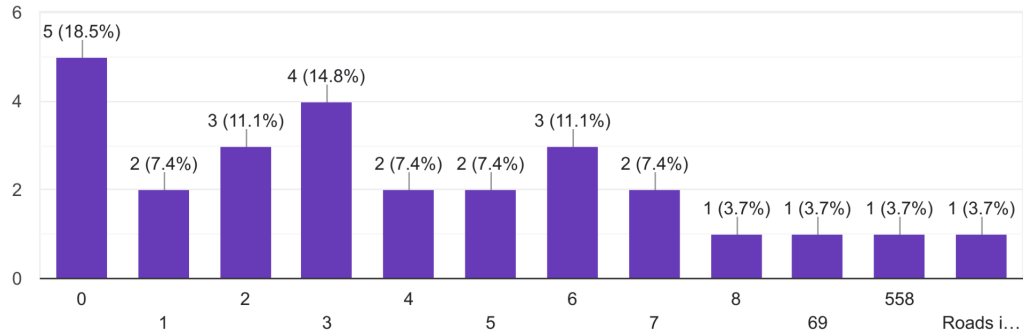
Are you a permanent resident of district entered above?

27 responses



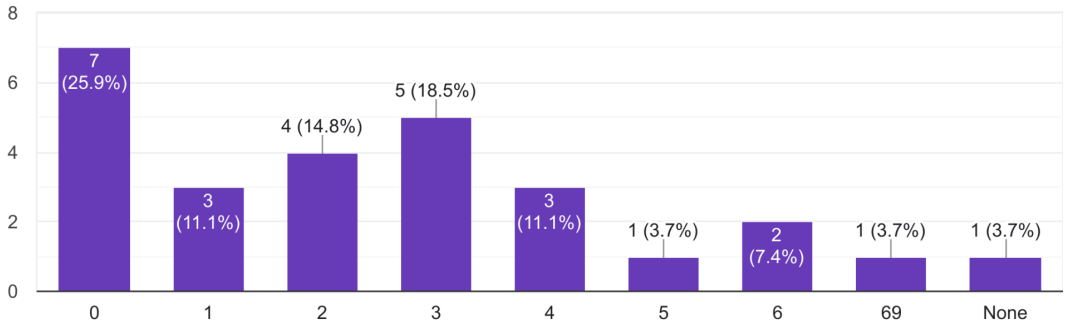
In total How many complaints have you registered in past year?

27 responses



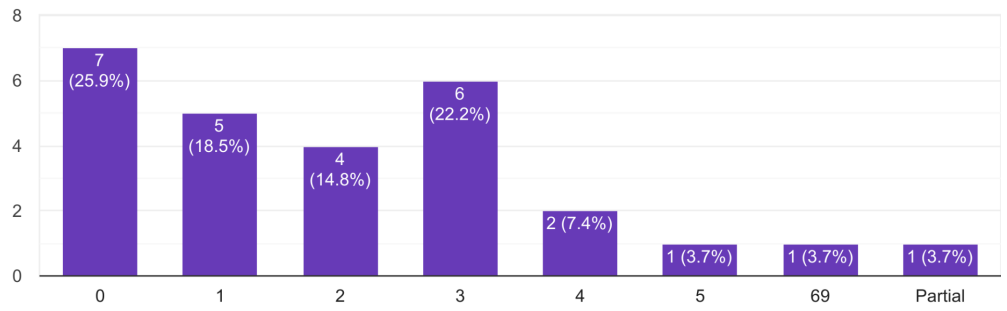
How many complaints got solved?

27 responses



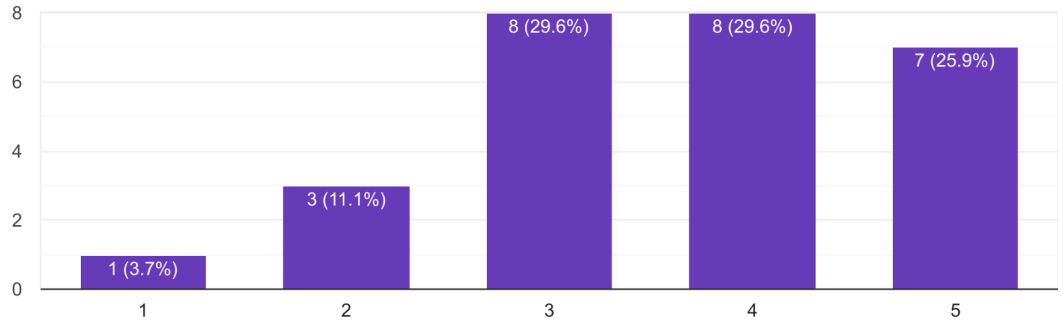
How many were solved in satisfied time?

27 responses



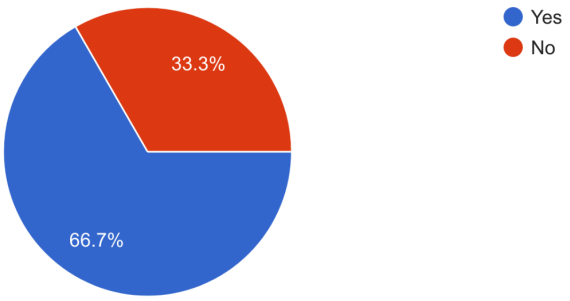
Select level of satisfaction of solution

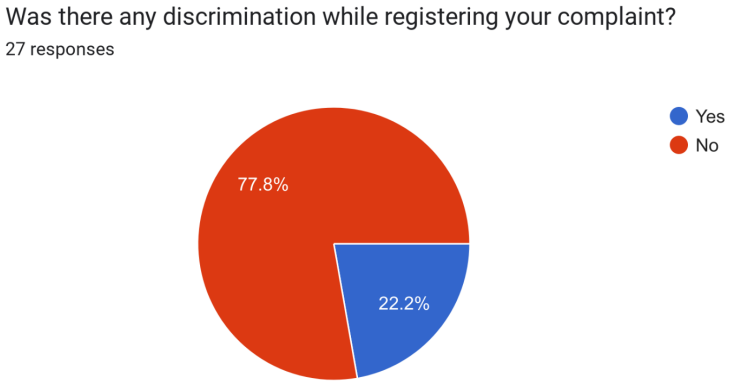
27 responses

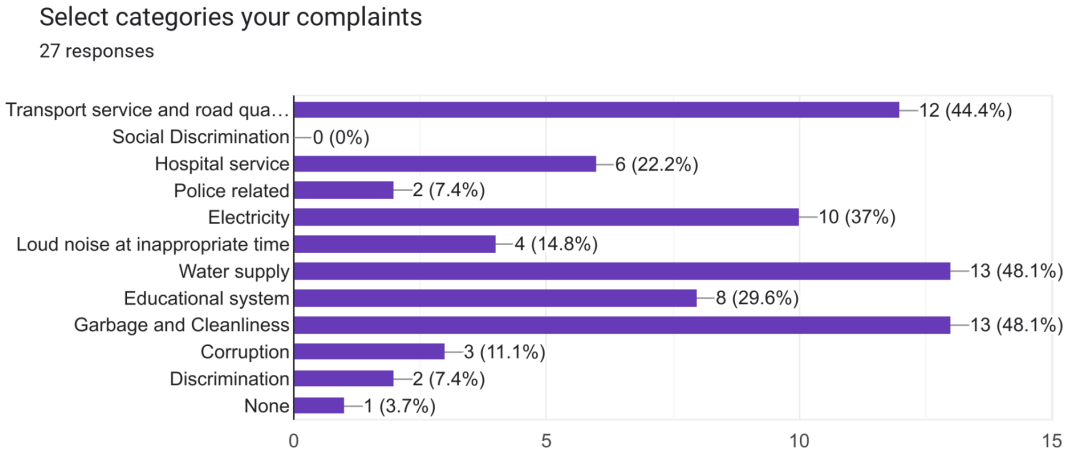


Was the registering complaint's process simple?

27 responses







3. Fact Finding Chart

Objective	Technique	Resources	Time Commitment
To get the background of the existing district management system of Ahmedabad	Background Reading	Ahmedabad Municipal Corporation website (CCRS AMC)	2 hours
To get the background of the existing district management system of Bhavnagar	Background Reading	Bhavnagar Municipal Corporation website(BMC system)	1 hour

To gain the understanding of the requirements of the district management.	Interview	Batchmates	2 hours
To gather public opinions about their suggestions on the system.	Questionnaire	Batchmates and Family members	5 hours
Updating and maintenance on a regular basis	Background Reading, Interview	Verified online resources and batchmates.	3 hours
To gain an understanding of the security of the system.	Interview		1.5 hours

4. Requirements

- The system must be very secure.
- Access to the system needs to be privileged so that only essential information is exposed.
- Significant data storage capacity
- Updating and maintenance on a regular basis
- The system needs to have the ability to handle multiple services
- The system should be reliable and maintainable which ensures that errors are solved using limited amount of time
- Authorized users can only access certain sections

5. User Classes and Privileges

5.1 User Classes and Characteristics

- **Citizen:** The person who utilizes the system or for whom it was designed.
- **Tehsildar:** A Tehsildar main work to hear the disputes. He also ensures that records are properly kept and land revenue is collected.
- **Sub divisional Officer:** The Sub-divisional Officer(Civil) is the chief civil officer of the Sub-Division. He exercises direct control over the Tehsildars and their staff.
- **Law Enforcement Agencies:** They are agencies hired for complaint resolution by Tehsildar. Their main function is to resolve complaints in a speedy process with satisfactory resolution.

5.2 Privileges

- **Citizen Module:** Basic three privileges are provided to a user. Registration of complaints in the simplest way available. Updating the status of complaint on a daily basis to the citizen and providing an option to reopen the complaint on not getting a satisfied resolution.
- **Tehsildar Module:** In this module Tehsildar has all the information about citizens and complaints of his district. For example, how many complaints are pending and how many got resolved, Which category has more complaints and how many staff members are assigned there. He is also provided with all the staff members and their respective categories. Tehsildar is also given the employed law agencies list, so that s/he can distribute complaints among them.
- **SDO(Sub- Divisional Officers) Module:** All the Tehsildars' and their respective areas' list is provided to SDO, so that he knows which Tehsil is assigned for which area. He is also provided with reopened complaints in all the Tehsils, so that he knows which Tehsil are working efficiently and which ones are not, and he can take respective steps.
- **Law Enforcement Agencies Module:** All the law agencies are provided with the list of complaints allotted to them by Tehsildar. They need to provide the status on a daily basis to Tehsil so that they can forward it to citizens.

6. Assumptions

- Users can read and write English.
- Users have devices that support internet service.
- Users have a decent internet connection.
- Users know how websites work.
- Users must have a Windows operating system PC to access the software.
- The overall speed of the system will depend on the speed of the network used to communicate between the database PC and the user PC.

7. Business Constraints

- It will be difficult to manage such a huge database for a single person and this will lead to hiring of manpower. This leads to additional expenses which the owner must incur.
- Funding such a massive system will also be a concern.

Section 2: Noun and verb analysis

Final Description

Purpose

The District Management System aims to keep track of user complaints for different districts. The main purpose of this system must be to provide complaint resolution to users as soon as possible keeping the procedure simple. Users can re-issue the complaint, if that was not addressed properly.

System includes streamlined functioning and supervision of the public services provided in the administration of a district like education, health, agriculture, electricity, drinking water and sanitation, transport service and road quality, law and order, etc.

Intended Audience and Reading Suggestions

The intended audience for this document includes Software developers, All the citizens of a district who are going to use this software to address their issues, Employees at the tehsil administrative level who will go through these issues, resolve it, and update the status of the issue, Employees at the district administrative level who will foresee different tehsils that fall under a district and finally the Law Enforcement Agencies (which falls under the jurisdiction of the tehsil) will be called upon by the tehsils to resolve the issues.

This project is being developed under the guidance of Prof. Minal Bhise and Prof. Rachit Chhaya. This project is useful for the district administrative services and the citizens of the district.

The reader of the document should go through the purpose, product scope and description from the introduction section. Then from the fact-finding phase section read the background readings, requirements gathered, agenda of the interview, questionnaire. Then through fact-finding chart, requirements, user categories and privileges, assumptions and finally business constraints.

Product Scope

The District Management System project entails the creation of a website that allows the users (citizens of a district) to register a complaint in their respective tehsil's administrative department.

In the current times, the government processes are very complex and it is difficult for any citizen to understand the system and need to visit the administrative offices for the update on their query. But through this system users only need to register a complaint once in the respective category and can

receive the update which will be easily accessible via message, mail, or WhatsApp. This system needs to operate on a basic framework so that the user can grasp it.

The system aims to keep track of user complaints for different districts. The administrative dept includes employees that foresee different tehsils that fall under a district. There are multiple tehsils under a district, and every tehsil has its citizens' data.

The citizen can register a complaint in their tehsil's administrative department. If required, the tehsil can employ law enforcement agencies available under their jurisdiction to resolve the issue. The users should be able to track the status of their complaints. Users can re-issue the complaint, if that was not addressed properly.

The system also includes capabilities such as user registration online, information of tehsildars to resolve the issues, information of district level employees who supervise the smooth functioning of the district administration. The respected users can edit their specific details, and the administrator can add, delete, or change users' information and status of their addressed issues.

Description

The District Management System is a self-contained software product that replaces the current manual and visiting processes for tracking the status of users' registered complaints. The Software Requirements Specification provides a detailed description of the requirements for the project "District Management System". This SRS will allow the readers to understand what to expect from this system and how to approach and create this system step by step. The clear understanding of the system and its functionality will allow the readers to successfully develop the project. From this SRS, the District Management System can be designed, constructed, and finally tested.

This software shall ensure hassle-free registration of a particular complaint with citizens being able to track their complaint. The Tehsildar (Tehsil level administrator) looks into resolving the complaints of the citizens and to update the status of the complaints. The Sub-Division Officer (District level administrator) keeps an eye on the functioning of the tehsil administrative body. The Law Enforcement Agencies (which falls under the jurisdiction of the tehsil) will be called upon by the tehsils to resolve the issues.

The various functions available to the citizens are registering complaints, tracking status of complaints and reopening of the complaints if they are not addressed properly. A place for feedback shall also be built where citizens can give their reviews on their satisfaction with their complaints solved. Quick solving of complaints are to be provided by the tehsildars to the citizens.If required, the tehsil can employ law enforcement agencies available under their jurisdiction to resolve the issue. Citizens are provided with an online help platform which solves their queries and directs them to the right place. In order to ensure security, only authorized people are allowed to make

changes in the software. The Sub-Divisional Officer can keep track of the issues undertaken by the tehsils.

Perform Noun Analysis and build the ER Diagram for your project

1. Noun and Verb Analysis

a. All Extracted Nouns and Verbs from Problem Description

Noun	Verb
District	Track
Management	Provide
System	Create
User	
Complaints	
Resolution	
Issue	
Administration	Functioning
Supervision	
Education	
Health	
Agriculture	
Electricity	
Water	
Sanitation	
Transport	

Road	
Law	
Order	
Audience	
Document	
Software	
Developers	
Citizens	
Data	
Employees	Resolve
Tehsil	Update
Status	Foresee
Administrative	Employ
Level	
Law	
Enforcement	
Agencies	
Jurisdiction	
Project	Developed
Minal	
Bhise	
Rachit	
Chhaya	
District	Creation
Management	

System	
Website	Allow
User	
Citizen	Register
Complaint	
Department	
Government	
Processes	
Offices	
Query	Solving
Category	
Message	
Mail	
WhatsApp	
Re-issue	
Online	
Information	Add
Tehsildars	Delete
Users	Edit
Details	Reopening
Software	Designed
Manual	Constructed
Requirements	Tested
Specification	
Description	

Hassle	
Sub-Division	
Officer	
Feedback	
Built	
Reviews	
Satisfaction	
Platform	
Security	

b. Accepted Noun and Verbs List

Candidate Entity Set	Candidate Attribute Set	Candidate Relationship Set
Citizen	C_id C_name C_DOB C_age C_address C_email C_contact_no T_id	Complaint Management
Tehsildar	Td_id Td_name T_id	List of Tehsildar
Sub-Division Officer	S_id S_name S_zone S_email	Head of Tehsildar
Tehsil	T_id T_name T_office_address	Tehsils under zone

	T_email T_contactno S_id	
Law Enforcement Agency	L_id L_name L_helpline_no T_id	Different Agencies
Complaints	Cp_id Cp_issue Cp_location Cp_registration_date Cp_duration Ts_id T_id C_id	List of complaints
Feedback	F_id F_satisfaction F_review Cp_id	Feedbacks by public
Reopen Complaints	R_id R_reason Cp_id	Reopening option
Track status	Ts_id Ts_status Ts_current_authority	Status of company

c. Rejected Noun and Verbs List

Noun	Reject Reason
resolution	general
time	general
next	irrelevant
system	general
record	attributes
name	attributes

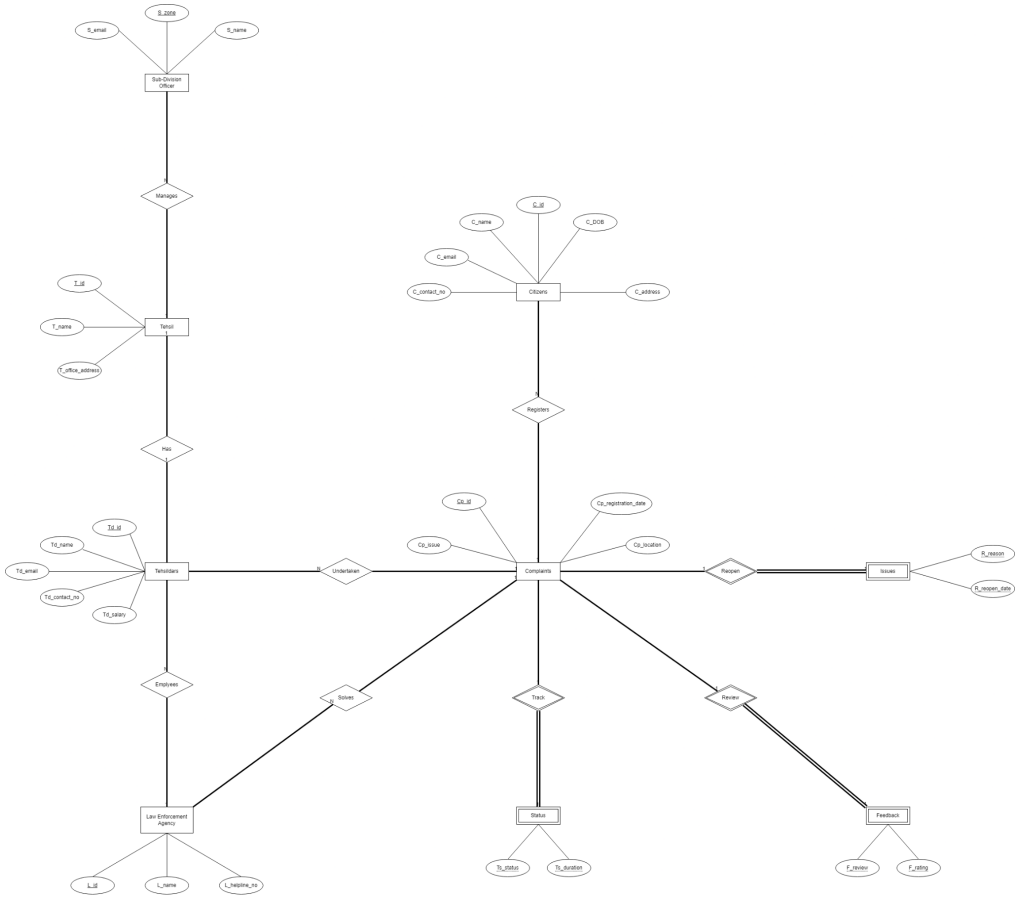
Aadhar ID	attributes
phone number	attributes
address	attributes
information	vague
numbers	Irrelevant
employee ID	attributes
complaint ID	attributes
complaint type ID	attributes
order	Irrelevant
checks	Irrelevant
tables	duplicate
data	Irrelevant
authorities	duplicate
action	attributes
activities	vague
level	vague
details	vague
Nodal officer	Duplicate
input	Irrelevant
query	duplicate
deadline	attributes
action	general
status	attributes
facilities	duplicate
personnel	attributes
grievance	duplicate
portal	general
timeframe	duplicate
activities	Irrelevant
ratings	attribute

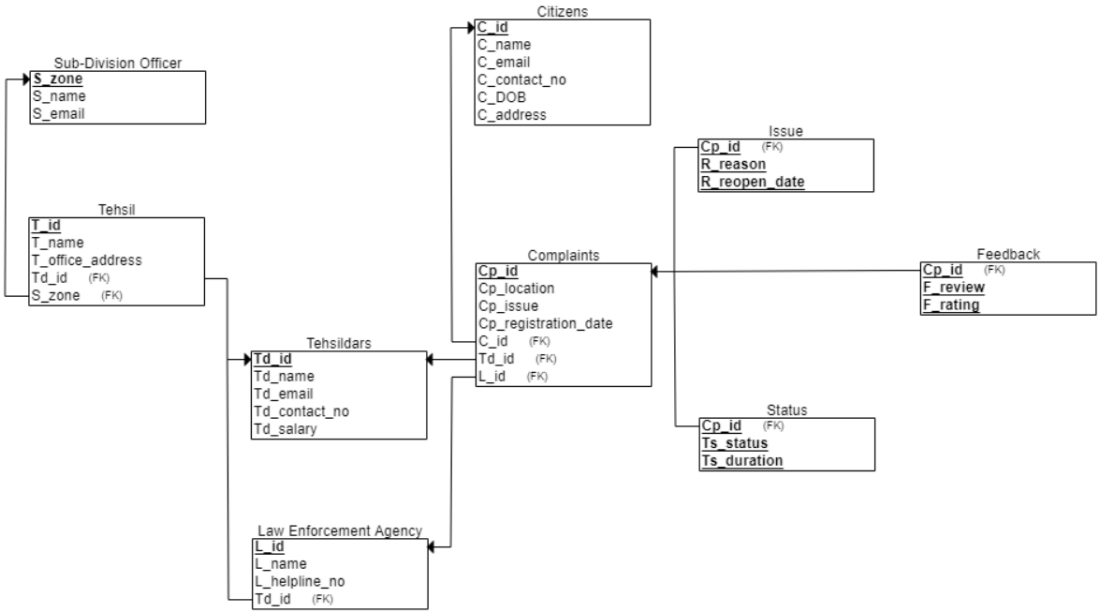
Date	attribute

Verb	Reject Reason
seek	duplicate
improve	duplicate
keep	general
functioning	irrelevant
take	vague
required	duplicate
lodge	duplicate
reside	general
addressed	duplicate
checked	duplicate
provided	general
complain	attribute
according	vague
get	vague
update	duplicate
solved	duplicate
have	vague
update	duplicate
devoted	general
shows	duplicate
resolved	duplicate
redressed	duplicate
done	vague
resolve	duplicate
update	duplicate
stipulated	attribute
add	duplicate
remove	duplicate

see	general
pending	general
resolved	duplicate
functioning	general
seek	duplicate
improve	duplicate
keep	general
functioning	irrelevant
take	vague

Section 3: ER diagram and Relational Model





Section 4: Normalization and DDL script

Normalization and Schema Refinement:

1. Sub-Division Officer

Schema: Sub-Division Officer (**S_zone**, S_name, S_email)

PK: S_zone

Functional Dependencies:

S_zone → S_name, S_email

S_email → S_zone

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key. Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

2. Tehsil

Schema: Tehsil (**T_id**, T_name, T_office_address, Td_id, S_zone)

PK: T_id

FK: Td_id, S_zone

Functional Dependencies:

T_id → T_name, T_office_address, Td_id, S_zone

Td_id → T_id

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key. Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

3. Tehsildar

Schema: Tehsildar (**Td_id**, Td_name, Td_email, Td_contact_no, Td_salary)

PK: Td_id

Functional Dependencies:

Td_id → Td_name, Td_email, Td_contact_no, Td_salary

Td_email → Td_id, Td_name, Td_contact_no, Td_salary

Td_contact_no → Td_id, Td_name, Td_email, Td_salary

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key. Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

4. Law Enforcement Agency

Schema: Law Enforcement Agency (**L_id**, L_name, L_helpline_no, Td_id)

PK: L_id

FK: Td_id

Functional Dependencies:

L_id → L_name, L_helpline_no, Td_id

L_helpline_no → L_id, L_name, Td_id

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key. Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

5. Citizens

Schema: Citizens (**C_id**, C_name, C_email, C_contact_no, C_DOB, C_address)

PK: C_id

Functional Dependencies:

C_id → C_name, C_email, C_contact_no, C_DOB, C_address

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key. Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

6. Complaints

Schema: Complaints (**Cp_id**, Cp_issue, Cp_registration_date, Cp_location, C_id, Td_id, L_id)

PK: Cp_id

FK: C_id, Td_id, L_id

Functional Dependencies:

Cp_id → Cp_issue, Cp_registration_date, Cp_location, C_id, Td_id, L_id

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key.

Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

7. Status

Schema: Status (**Cp_id, Ts_status, Ts_duration**)

PK: (Cp_id, Ts_status, Ts_duration)
FK: Cp_id
Partial /Unique Key: Ts_status, Ts_duration

Functional Dependencies:

Cp_id, Ts_status, Ts_duration → Cp_id, Ts_status, Ts_duration

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key.
Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

8. Feedback

Schema: Feedback (**Cp_id, F_review, F_rating**)

PK: (Cp_id, F_review, F_rating)
FK: Cp_id
Partial /Unique Key: F_review, F_rating

Functional Dependencies:

Cp_id, F_review, F_rating → Cp_id, F_review, F_rating

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key.
Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

9. Issue

Schema: Issue (**Cp_id, R_reason, R_reopen_date**)

PK: (Cp_id, R_reason, R_reopen_date)
FK: Cp_id
Partial /Unique Key: R_reason, R_reopen_date

Functional Dependencies:

Cp_id, R_reason, R_reopen_date → Cp_id, R_reason, R_reopen_date

Normal Form and Anomalies:

This table is already in BCNF as all attributes are dependent on super key or candidate key.

Since it is in BCNF, it guarantees that there exists no anomalies in the relation.

DDL Script:

```
CREATE SCHEMA district_ms;
SET SEARCH_PATH TO district_ms;

CREATE TABLE IF NOT EXISTS Citizens
(
  C_id integer NOT NULL,
  C_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
  C_email character varying(60) COLLATE pg_catalog."default" NOT NULL,
  C_contact_no character varying(50) COLLATE pg_catalog."default" NOT NULL,
  C_DOB date NOT NULL,
  C_address character varying(100) COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT C_pkey PRIMARY KEY (C_id)
);

CREATE TABLE IF NOT EXISTS Tehsildars
(
  Td_id integer NOT NULL,
  Td_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
  Td_email character varying(60) COLLATE pg_catalog."default" NOT NULL,
  Td_contact_no character varying(50) COLLATE pg_catalog."default" NOT NULL,
  Td_salary integer NOT NULL,
  CONSTRAINT Td_pkey PRIMARY KEY (Td_id)
);

CREATE TABLE IF NOT EXISTS Sub_Division_Officer
(
  S_zone character varying(10) COLLATE pg_catalog."default" NOT NULL,
  S_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
  S_email character varying(60) COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT S_pkey PRIMARY KEY (S_zone)
);

CREATE TABLE IF NOT EXISTS Law_Enforcement_Agency
(
  L_id integer NOT NULL,
```

```
  L_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
  L_helpline_no character varying(50) COLLATE pg_catalog."default" NOT NULL,
  Td_id integer NOT NULL,
  CONSTRAINT L_pkey PRIMARY KEY (L_id),
  CONSTRAINT Td_id FOREIGN KEY (Td_id) REFERENCES Tehsildars(Td_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS Complaints
(
  Cp_id integer NOT NULL,
  Cp_issue character varying(100) COLLATE pg_catalog."default" NOT NULL,
  Cp_registration_date date NOT NULL,
  Cp_location character varying(50) COLLATE pg_catalog."default" NOT NULL,
  C_id integer NOT NULL,
  Td_id integer NOT NULL,
  L_id integer NOT NULL,
  CONSTRAINT Cp_pkey PRIMARY KEY (Cp_id),
  CONSTRAINT C_id FOREIGN KEY (C_id) REFERENCES Citizens(C_id) MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
  NOT VALID,
  CONSTRAINT Td_id FOREIGN KEY (Td_id) REFERENCES Tehsildars(Td_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
  NOT VALID,
  CONSTRAINT L_id FOREIGN KEY (L_id) REFERENCES Law_Enforcement_Agency(L_id)
MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
  NOT VALID
);

CREATE TABLE IF NOT EXISTS Tehsil
(
  T_id integer NOT NULL,
  T_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
```

```
  T_office_address character varying(100) COLLATE pg_catalog."default" NOT
NULL,
  Td_id integer NOT NULL,
  S_zone character varying(10) COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT t_pkey PRIMARY KEY (T_id),
  CONSTRAINT Td_id FOREIGN KEY (Td_id) REFERENCES Tehsildars(Td_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION,
  CONSTRAINT S_zone FOREIGN KEY (S_zone) REFERENCES
Sub_Division_Officer(S_zone) MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS Status
(
  Ts_status character varying(50) COLLATE pg_catalog."default" NOT NULL,
  Ts_duration integer NOT NULL,
  Cp_id integer NOT NULL,
  CONSTRAINT Ts_pkey PRIMARY KEY (Ts_status, Ts_duration, Cp_id),
  CONSTRAINT Cp_id FOREIGN KEY (Cp_id) REFERENCES Complaints(Cp_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS Feedback
(
  F_rating integer NOT NULL,
  F_review character varying(100) COLLATE pg_catalog."default" NOT NULL,
  Cp_id integer NOT NULL,
  CONSTRAINT F_pkey PRIMARY KEY (F_rating, F_review, Cp_id),
  CONSTRAINT Cp_id FOREIGN KEY (Cp_id) REFERENCES Complaints(Cp_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS Issue
```

```
(
  R_reason character varying(100) COLLATE pg_catalog."default" NOT NULL,
  R_reopen_date date NOT NULL,
  Cp_id integer NOT NULL,
  CONSTRAINT R_pkey PRIMARY KEY (R_reason, R_reopen_date, Cp_id),
  CONSTRAINT Cp_id FOREIGN KEY (Cp_id) REFERENCES Complaints(Cp_id) MATCH
SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

---
COPY district_ms.citizens(c_id,c_name,c_email,c_contact_no,c_DOB,c_address)
FROM 'C:\Users\Public\District management system\Citizens_data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.tehsildars(td_id,td_name,td_email,td_contact_no,td_salary)
FROM 'C:\Users\Public\District management system\Tehsildar_Data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.sub_division_officer(s_zone,s_name,s_email)
FROM 'C:\Users\Public\District management
system\Sub_Division_Officer_Data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.law_enforcement_agency(l_id,l_name,l_helpline_no,td_id)
FROM 'C:\Users\Public\District management
system\LawEnforcementAgency_data.csv'
DELIMITER ',' CSV HEADER;

COPY
district_ms.complaints(cp_id,cp_issue,cp_registration_date,cp_location,c_id,
td_id,l_id)
FROM 'C:\Users\Public\District management system\Complaints_data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.tehsil(t_id,t_name,t_office_address,td_id,s_zone)
FROM 'C:\Users\Public\District management system\Tehsil.csv'
DELIMITER ',' CSV HEADER;
```

```
COPY district_ms.status(cp_id,ts_status,ts_duration)
FROM 'C:\Users\Public\District management system\Status_data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.feedback(cp_id,f_review,f_rating)
FROM 'C:\Users\Public\District management system\Feedback_data.csv'
DELIMITER ',' CSV HEADER;

COPY district_ms.issue(cp_id,r_reason,r_reopen_date)
FROM 'C:\Users\Public\District management system\Issue_data.csv'
DELIMITER ',' CSV HEADER;
```

Section 5: Queries

```sql
SET SEARCH_PATH TO district_ms;

--Que 1
select *
from complaints;

--Que 2
SELECT cp_id, cp_issue
FROM complaints;

--Que 3
SELECT *
FROM complaints
WHERE c_id=171;

--Que 4
SELECT Td_id, T_name
FROM Tehsil;

-- Que 5
SELECT *
FROM citizens
ORDER BY c_id ASC ;

--Que 6
SELECT L_id, L_name, Td_id
FROM law_enforcement_agency;

--Ques7
SELECT c_dob
FROM citizens
where c_dob > '1990-01-01'

--Que 8
select *
from issue;
```

```sql
--Que 9
SELECT *
From status
where ts_duration > 7
order by ts_duration asc;

-- Que 10
select f_rating, f_review
from feedback
where f_rating < 7
order by f_rating;

-- Que 11 using function
CREATE OR REPLACE FUNCTION Issues()
    -- declaring return type
    RETURNS integer
    LANGUAGE 'plpgsql'

    AS $BODY$

    DECLARE
    entries integer;
    BEGIN
    select count(*)
    into entries
    from complaints
    inner join status
    on complaints.cp_id = status.cp_id
    where ts_duration > 7;

    return entries;
    END;
    $BODY$;

select * from Issues();

-- Que 12
select *
from complaints
where td_id = 2 or td_id = 5;
```

```sql
-- Que 13
SELECT t_id, t_name, tehsildars.td_id, td_name, td_contact_no, s_zone
FROM tehsildars
INNER JOIN tehsil
ON tehsildars.td_id = tehsil.td_id;

-- Que 14
SELECT citizens.c_id, c_name, c_contact_no, c_address, c_dob, cp_id,
cp_issue
FROM complaints
INNER JOIN citizens
ON complaints.c_id = citizens.c_id
order by c_dob;

-- Que 15
SELECT cp_id, c_id, complaints.td_id, cp_issue, l_name, l_helpline_no
FROM complaints
INNER JOIN law_enforcement_agency
ON complaints.l_id = law_enforcement_agency.l_id;

-- Que 16
create view view_complaints as
select cp_issue, cp_location, td_id, l_id, cp_registration_date
from complaints
where cp_registration_date < '2022-10-01';

select *
from view_complaints;

-- Que 17
create view view_issue as
select r_reason, r_reopen_date, cp_id
from issue
where r_reopen_date < '2022-10-01';

SELECT cp_issue, cp_location, td_id, l_id, cp_registration_date, r_reason,
r_reopen_date
FROM complaints
INNER JOIN view_issue
```

```
ON complaints.cp_id = view_issue.cp_id
order by r_reopen_date;

-- Que 18
create view view_feedback as
select f_rating, f_review, cp_id
from feedback
where f_rating < 7;

SELECT cp_issue, cp_location, td_id, l_id, cp_registration_date, f_rating,
f_review
FROM complaints
INNER JOIN view_feedback
ON complaints.cp_id = view_feedback.cp_id
order by f_rating;

-- Que 19
create view view_status as
select ts_status, ts_duration, cp_id
from status
where ts_status = 'Work In Progress';

SELECT cp_issue, cp_location, td_id, l_id, cp_registration_date, ts_status,
ts_duration
FROM complaints
INNER JOIN view_status
ON complaints.cp_id = view_status.cp_id;

-- Que 20 using trigger
create or replace function "trig_func1"()
returns TRIGGER
LANGUAGE 'plpgsql'
as $body$
declare cit_id integer;
declare issue text;
BEGIN
select c_id, cp_issue into cit_id, issue
from complaints
where c_id = new.c_id and cp_issue = new.cp_issue;
if(cit_id = new.c_id and issue = new.cp_issue) then
```

```
raise notice 'complaint already exists, if you wish you can reopen the
complaint.';
ELSE
raise notice 'complaint does not exist.';
return new;
end if;
end
$body$;

CREATE TRIGGER "check_duplication"
before insert
ON complaints
FOR EACH ROW
EXECUTE PROCEDURE trig_func1();

INSERT INTO complaints values(1041, 'Water leakage from the pipes',
'2022-11-23', 'No.4, Brooks Villa, Nebraska', 194, 2, 8);
```

Section 6: Full-stack web development

Front-Hand related documents

- **Models**

```python
# This is an auto-generated Django model module.
# You'll have to do the following manually to clean this up:
#   * Rearrange models' order
#   * Make sure each model has one field with primary_key=True
#   * Make sure each ForeignKey and OneToOneField has `on_delete` set to the desired behavior
#   * Remove `managed = False` lines if you wish to allow Django to create, modify, and delete the table
# Feel free to rename the models, but don't rename db_table values or field names.
from django.db import models


class Citizens(models.Model):
    c_id = models.IntegerField(primary_key=True)
    c_name = models.CharField(max_length=50)
    c_email = models.CharField(max_length=60)
    c_contact_no = models.CharField(max_length=50)
    c_dob = models.DateField()
    c_address = models.CharField(max_length=100)

    class Meta:
        managed = False
        db_table = 'citizens'


class Complaints(models.Model):
    cp_id = models.IntegerField(primary_key=True)
    cp_issue = models.CharField(max_length=100)
    cp_registration_date = models.DateField()
    cp_location = models.CharField(max_length=50)
    c = models.ForeignKey(Citizens, models.DO_NOTHING)
```

```
    td = models.ForeignKey('Tehsildars', models.DO_NOTHING)
    l = models.ForeignKey('LawEnforcementAgency', models.DO_NOTHING)

    class Meta:
        managed = False
        db_table = 'complaints'


class Feedback(models.Model):
    f_rating = models.IntegerField(primary_key=True)
    f_review = models.CharField(max_length=100)
    cp = models.ForeignKey(Complaints, models.DO_NOTHING)

    class Meta:
        managed = False
        db_table = 'feedback'
        unique_together = (('f_rating', 'f_review', 'cp'),)


class Issue(models.Model):
    r_reason = models.CharField(primary_key=True, max_length=100)
    r_reopen_date = models.DateField()
    cp = models.ForeignKey(Complaints, models.DO_NOTHING)

    class Meta:
        managed = False
        db_table = 'issue'
        unique_together = (('r_reason', 'r_reopen_date', 'cp'),)


class LawEnforcementAgency(models.Model):
    l_id = models.IntegerField(primary_key=True)
    l_name = models.CharField(max_length=50)
    l_helpline_no = models.CharField(max_length=50)
    td = models.ForeignKey('Tehsildars', models.DO_NOTHING)

    class Meta:
        managed = False
        db_table = 'law_enforcement_agency'
```

```python
class Status(models.Model):
    ts_status = models.CharField(primary_key=True, max_length=50)
    ts_duration = models.IntegerField()
    cp = models.ForeignKey(Complaints, models.DO_NOTHING)

    class Meta:
        managed = False
        db_table = 'status'
        unique_together = (('ts_status', 'ts_duration', 'cp'),)


class SubDivisionOfficer(models.Model):
    s_zone = models.CharField(primary_key=True, max_length=10)
    s_name = models.CharField(max_length=50)
    s_email = models.CharField(max_length=60)

    class Meta:
        managed = False
        db_table = 'sub_division_officer'


class Tehsil(models.Model):
    t_id = models.IntegerField(primary_key=True)
    t_name = models.CharField(max_length=50)
    t_office_address = models.CharField(max_length=100)
    td = models.ForeignKey('Tehsildars', models.DO_NOTHING)
    s_zone = models.ForeignKey(SubDivisionOfficer, models.DO_NOTHING,
db_column='s_zone')

    class Meta:
        managed = False
        db_table = 'tehsil'


class Tehsildars(models.Model):
    td_id = models.IntegerField(primary_key=True)
    td_name = models.CharField(max_length=50)
    td_email = models.CharField(max_length=60)
    td_contact_no = models.CharField(max_length=50)
```

```
    td_salary = models.IntegerField()

    class Meta:
        managed = False
        db_table = 'tehsildars'
```

- **Views**

```
from django.shortcuts import render
from django.contrib import messages
from django.http import HttpResponse
from django.db import connection
from District.models import *

def homepage(request):
    return render(request,'Homepage.html')

def adminloginportal(request):
    return render(request,'AdminLoginPortal.html')

def adminportal(request):
    return render(request,'AdminPortal.html')

def citizenloginportal(request):
    return render(request,'CitizenLoginPortal.html')

def citizenportal(request):
    return render(request,'CitizenPortal.html')

def citizencomplaintportal(request):
    return render(request,'CitizenComplaintPortal.html')

def tehsildarloginportal(request):
    return render(request,'TehsildarLoginPortal.html')

def tehsildarportal(request):
    return render(request,'TehsildarPortal.html')
```

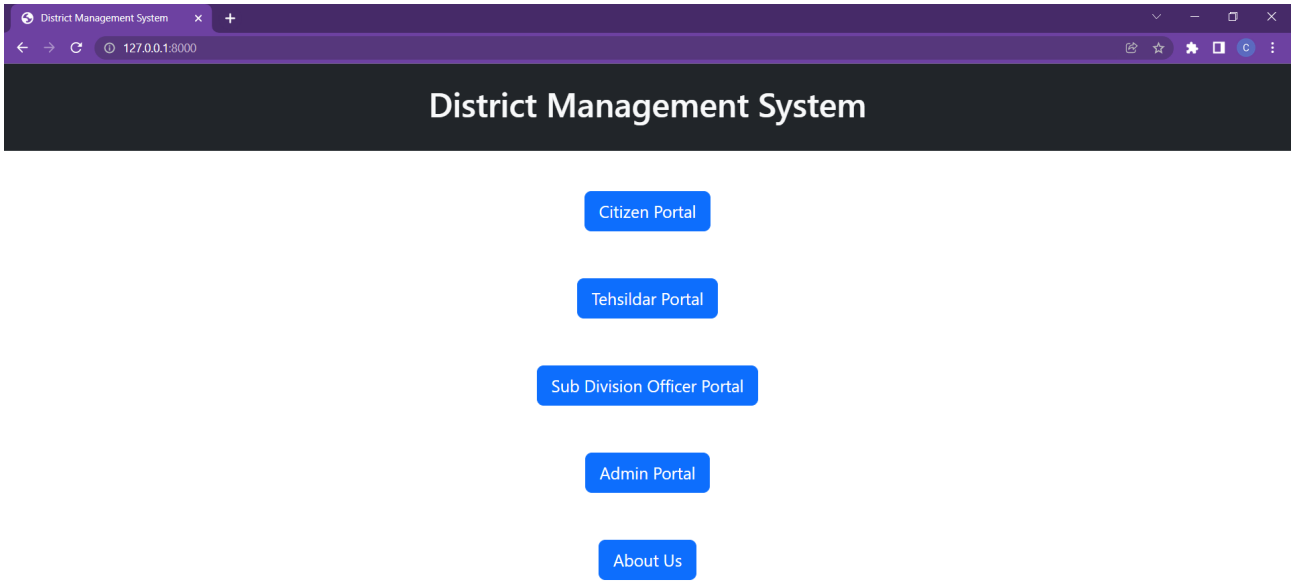
```
def tehsildarcomplaintsportal(request):
    return render(request,'TehsildarComplaintsPortal.html')

def sdologinportal(request):
    return render(request,'SDOLoginPortal.html')

def sdoportal(request):
    return render(request,'SDOPortal.html')

def aboutus(request):
    return render(request,'AboutUs.html')

def showallcitizen(request):
    showall = Citizens.objects.all()
    return render(request, 'CitizenDetails.html', {"data":showall})

def InsertCitizen(request):
    if request.method=="POST":
        saverecord=Citizens()
        saverecord.c_id=request.POST.get('c_id')
        saverecord.c_name=request.POST.get('c_name')
        saverecord.c_email=request.POST.get('c_email')
        saverecord.c_contact_no=request.POST.get('c_contact_no')
        saverecord.c_dob=request.POST.get('c_dob')
        saverecord.c_address=request.POST.get('c_address')
        saverecord.save()

        messages.success(request,'Citizens '+saverecord.c_id+' is saved
successfully!')
        return render(request,'InsertCitizen.html')
    else:
            return render(request,'InsertCitizen.html')

def runQuerySolvedComplaints(request):
    raw_query = "select cp1.cp_id, cp1.cp_issue, cp1.cp_location,
ts1.ts_status, ts1.ts_duration from public.complaints cp1 join public.status
ts1 on cp1.cp_id = ts1.cp_id where ts1.ts_status='Resolved' order by cp_id
asc;"
    cursor = connection.cursor()
    cursor.execute(raw_query)
```

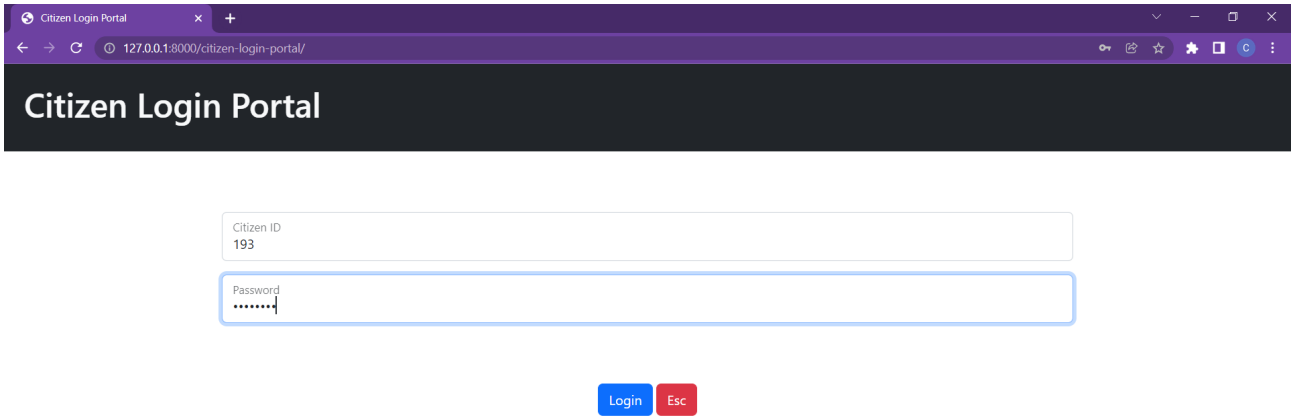
```
    alldata=cursor.fetchall()
    return render(request,'SolvedComplaints.html',{'data':alldata})

def runQueryStatusComplaints(request):
    raw_query = "select cp1.cp_id, cp1.cp_issue, cp1.cp_location,
ts1.ts_status, ts1.ts_duration from public.complaints cp1 join public.status
ts1 on ts1.cp_id = cp1.cp_id where cp1.cp_id='1010'"
    cursor = connection.cursor()
    cursor.execute(raw_query)
    alldata=cursor.fetchall()
    return render(request,'StatusComplaints.html',{'data':alldata})

def runQueryTehsilTehsildar(request):
    raw_query = "select t1.t_id, t1.t_name, t1.t_office_address, t1.s_zone,
td1.td_id, td1.td_name, td1.td_email, td1.td_contact_no, td1.td_salary from
public.tehsil t1 join public.tehsildars td1 on td1.td_id = t1.td_id;"
    cursor = connection.cursor()
    cursor.execute(raw_query)
    alldata=cursor.fetchall()
    return render(request,'TehsilTehsildarDetails.html',{'data':alldata})
```

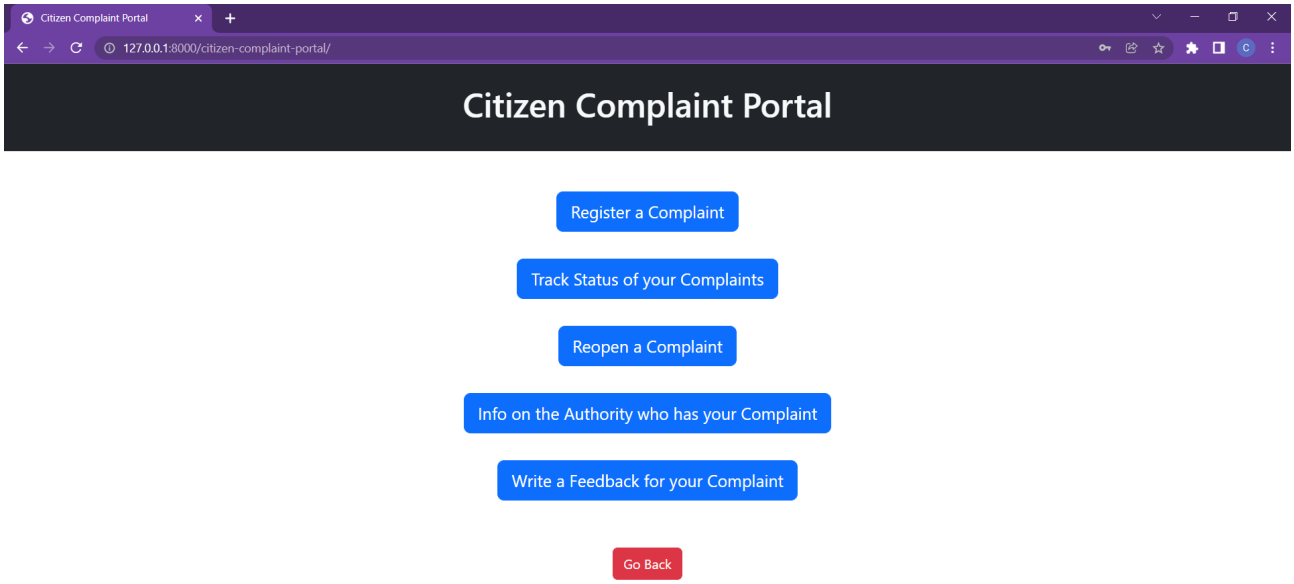
- **URLs**

```
"""District URL Configuration

The `urlpatterns` list routes URLs to views. For more information please
see:
    https://docs.djangoproject.com/en/4.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
```

```
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.homepage,name="Homepage"),
    path('admin-login-portal/',
views.adminloginportal,name="AdminLoginPortal"),
    path('admin-portal/', views.adminportal,name="AdminPortal"),
    path('insert-citizen/', views.InsertCitizen,name="InsertCitizen"),
    path('citizen-details/', views.showallcitizen,name="CitizenDetails"),
    path('citizen-login-portal/',
views.citizenloginportal,name="CitizenLoginPortal"),
    path('citizen-portal/', views.citizenportal,name="CitizenPortal"),
    path('citizen-complaint-portal/',
views.citizencomplaintportal,name="CitizenComplaintPortal"),
    path('tehsildar-login-portal/',
views.tehsildarloginportal,name="TehsildarLoginPortal"),
    path('tehsildar-portal/', views.tehsildarportal,name="TehsildarPortal"),
    path('tehsildar-complaints-portal/',
views.tehsildarcomplaintsportal,name="TehsildarComplaintsPortal"),
    path('solved-complaints/',
views.runQuerySolvedComplaints,name="SolvedComplaints"),
    path('status-complaints/',
views.runQueryStatusComplaints,name="StatusComplaints"),
    path('sdo-login-portal/', views.sdologinportal,name="SDOLoginPortal"),
    path('sdo-portal/', views.sdoportal,name="SDOPortal"),
    path('tehsil-tehsildar-details/',
views.runQueryTehsilTehsildar,name="TehsilTehsildarDetails"),
    path('about-us/', views.aboutus,name="AboutUs"),
]
```


District Management System
127.0.0.1:8000
District Management System
Citizen Portal
Tehsildar Portal
Sub Division Officer Portal
Admin Portal
About Us

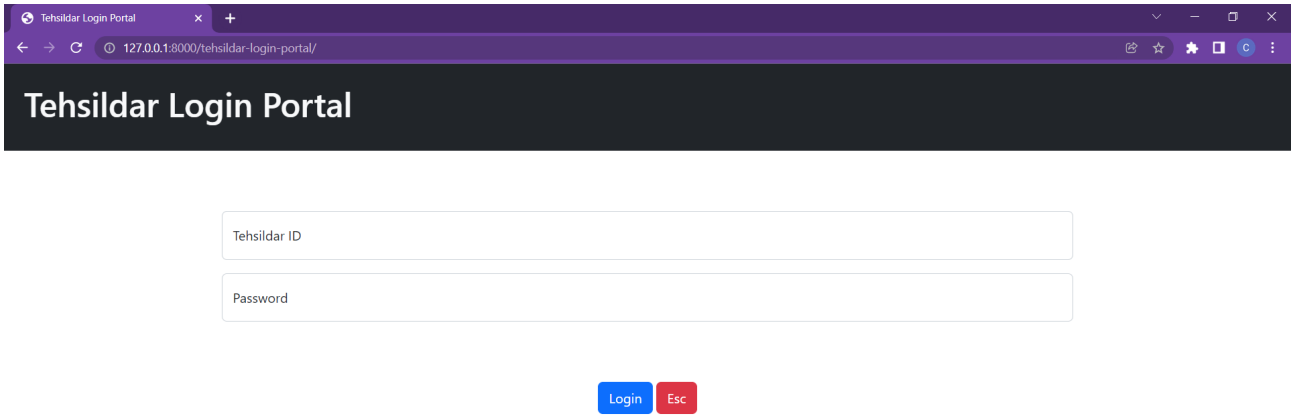

Citizen Login Portal
127.0.0.1:8000/citizen-login-portal/
Citizen Login Portal
Citizen ID
193
Password
Login
Esc

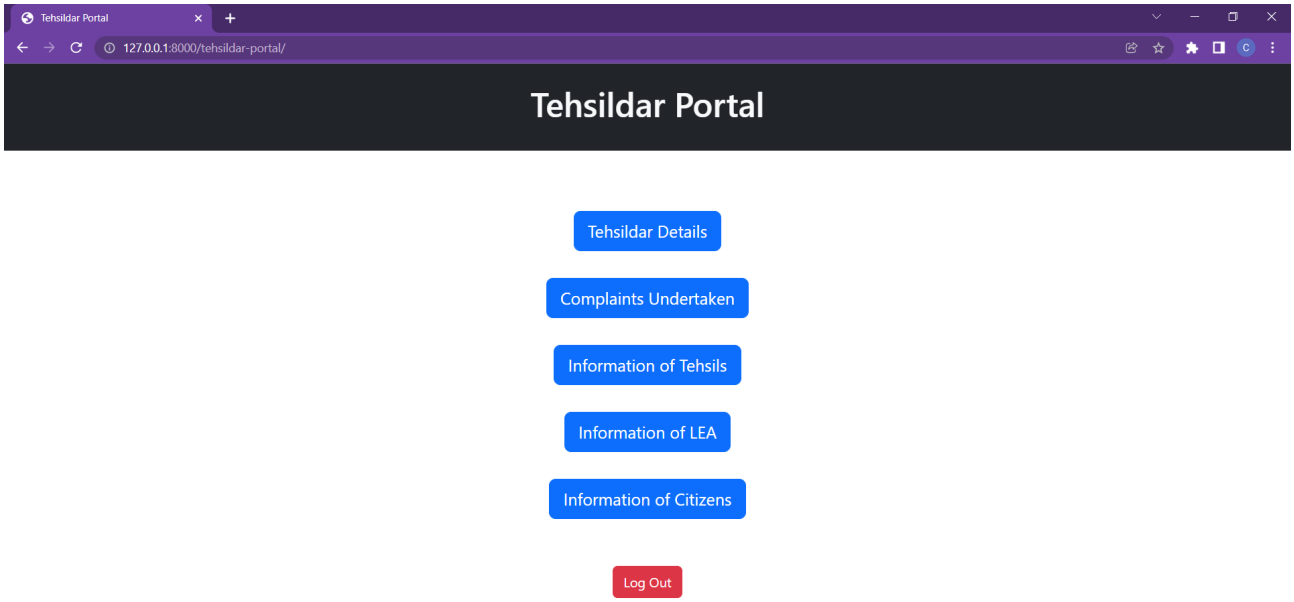


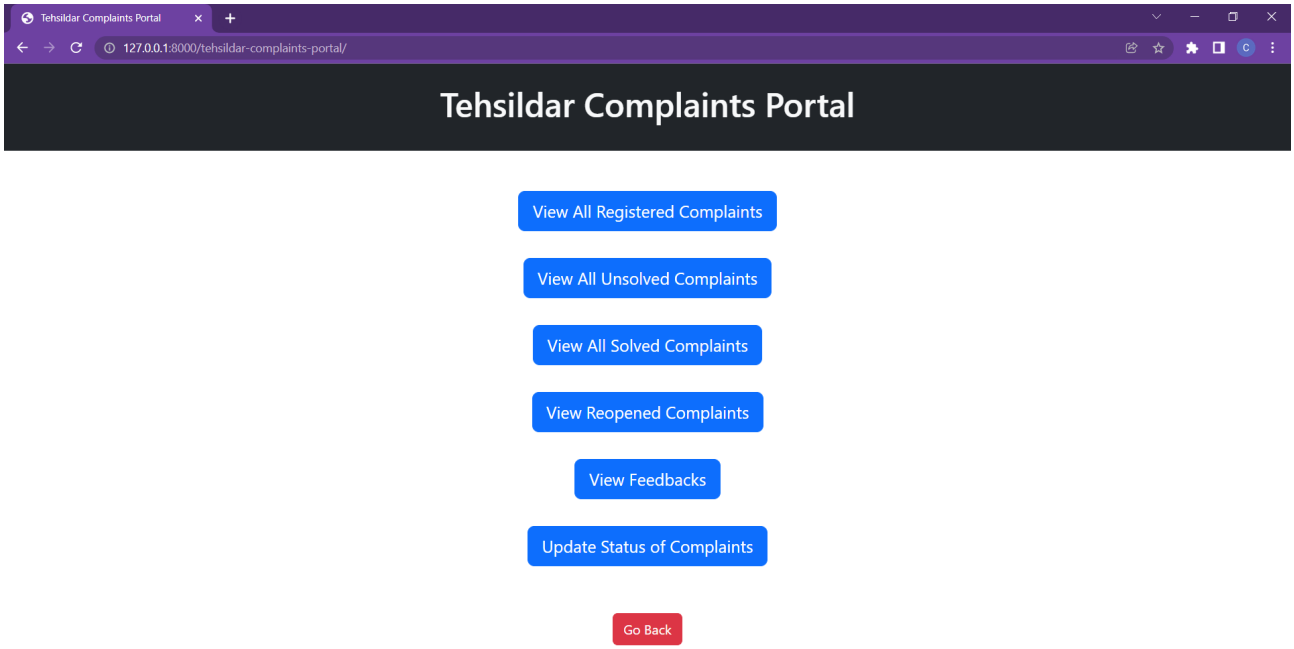
Go Back

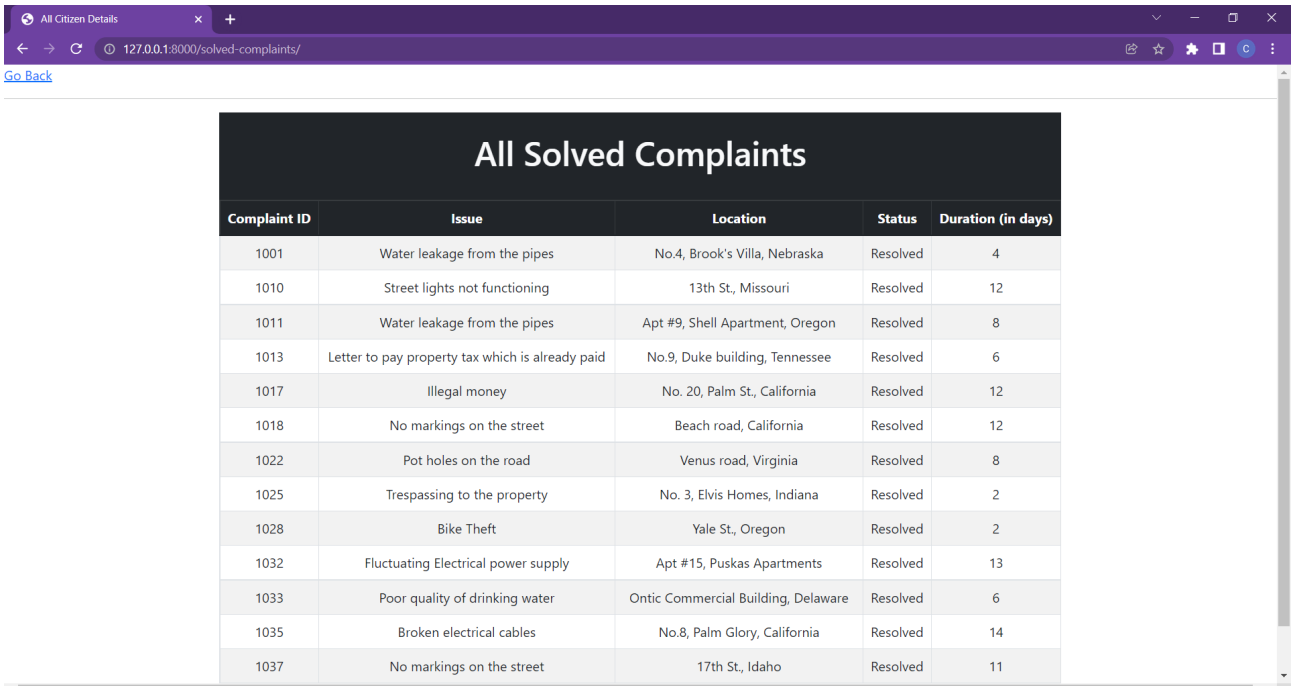
Complaint Status

Complaint ID	Issue	Location	Status	Duration (in days)
1010	Street lights not functioning	13th St., Missouri	Resolved	12

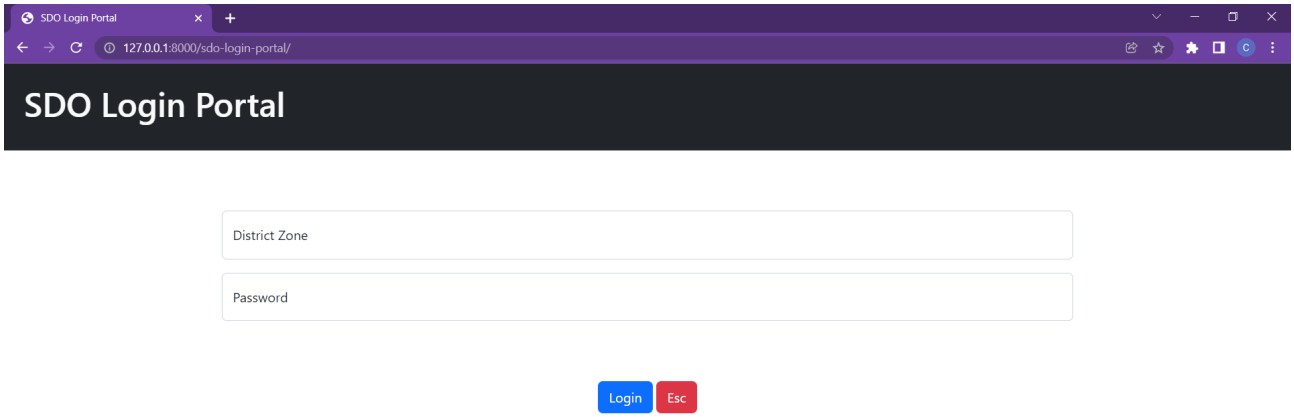

Tehsildar Login Portal
127.0.0.1:8000/tehsildar-login-portal/
Tehsildar Login Portal
Tehsildar ID
Password
Login
Esc

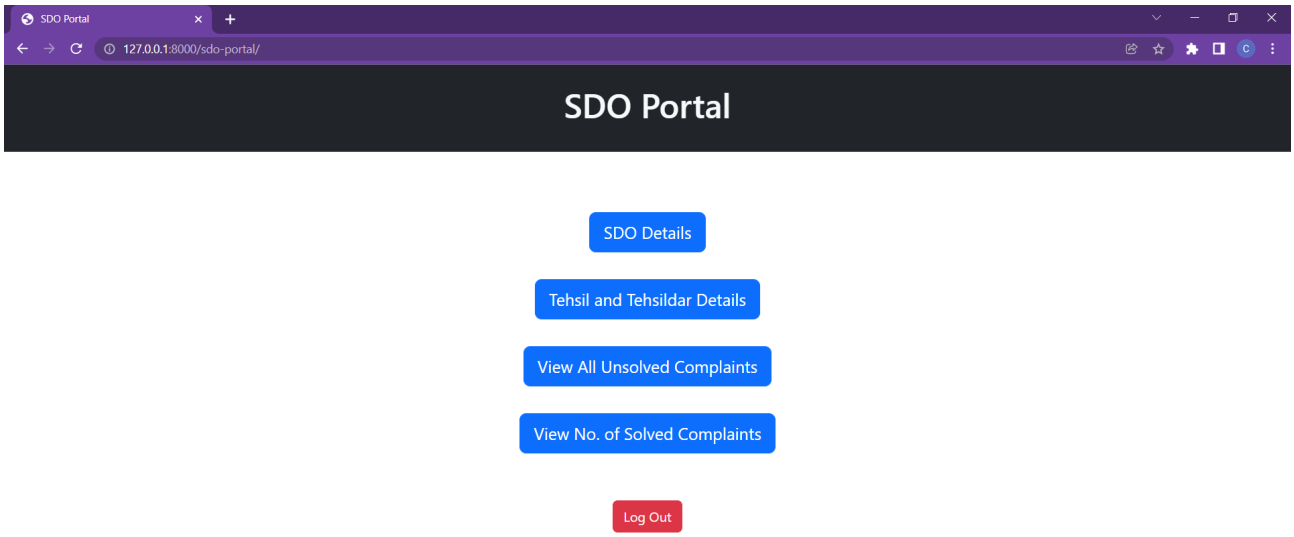

Tehsildar Portal
127.0.0.1:8000/tehsildar-portal/
Tehsildar Portal
Tehsildar Details
Complaints Undertaken
Information of Tehsils
Information of LEA
Information of Citizens
Log Out

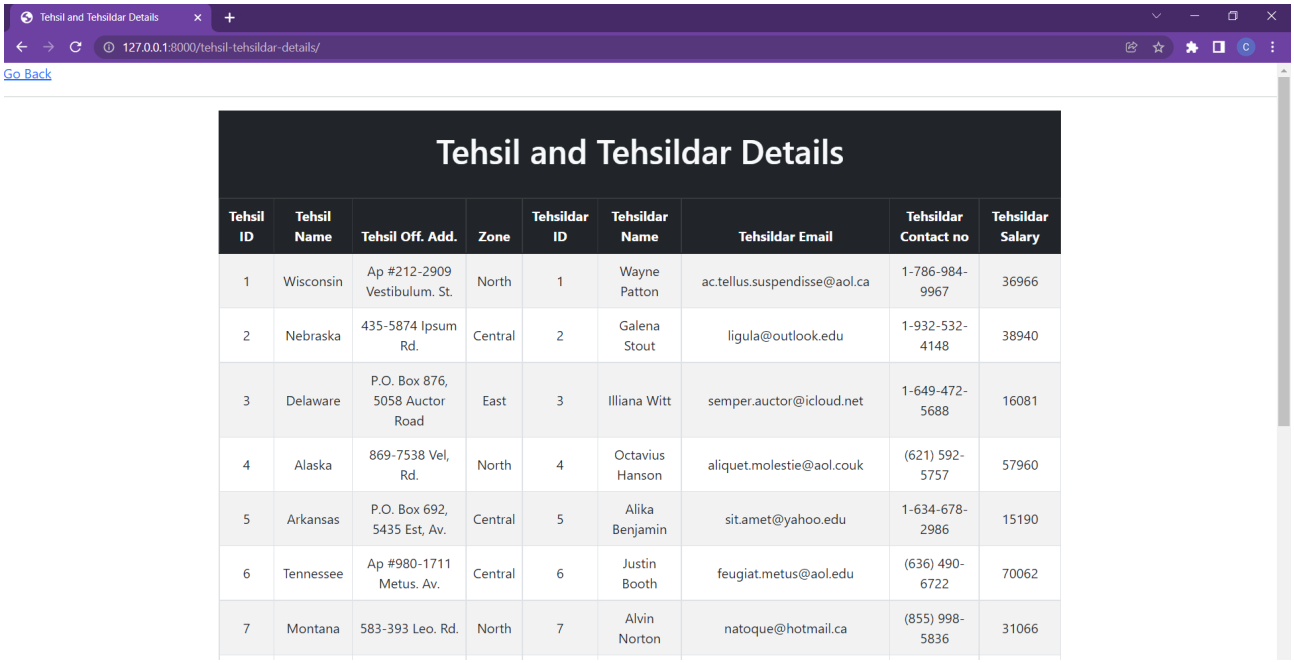




Complaint ID	Issue	Location	Status	Duration (in days)
1001	Water leakage from the pipes	No.4, Brook's Villa, Nebraska	Resolved	4
1010	Street lights not functioning	13th St., Missouri	Resolved	12
1011	Water leakage from the pipes	Apt #9, Shell Apartment, Oregon	Resolved	8
1013	Letter to pay property tax which is already paid	No.9, Duke building, Tennessee	Resolved	6
1017	Illegal money	No. 20, Palm St., California	Resolved	12
1018	No markings on the street	Beach road, California	Resolved	12
1022	Pot holes on the road	Venus road, Virginia	Resolved	8
1025	Trespassing to the property	No. 3, Elvis Homes, Indiana	Resolved	2
1028	Bike Theft	Yale St., Oregon	Resolved	2
1032	Fluctuating Electrical power supply	Apt #15, Puskas Apartments	Resolved	13
1033	Poor quality of drinking water	Ontic Commercial Building, Delaware	Resolved	6
1035	Broken electrical cables	No.8, Palm Glory, California	Resolved	14
1037	No markings on the street	17th St., Idaho	Resolved	11

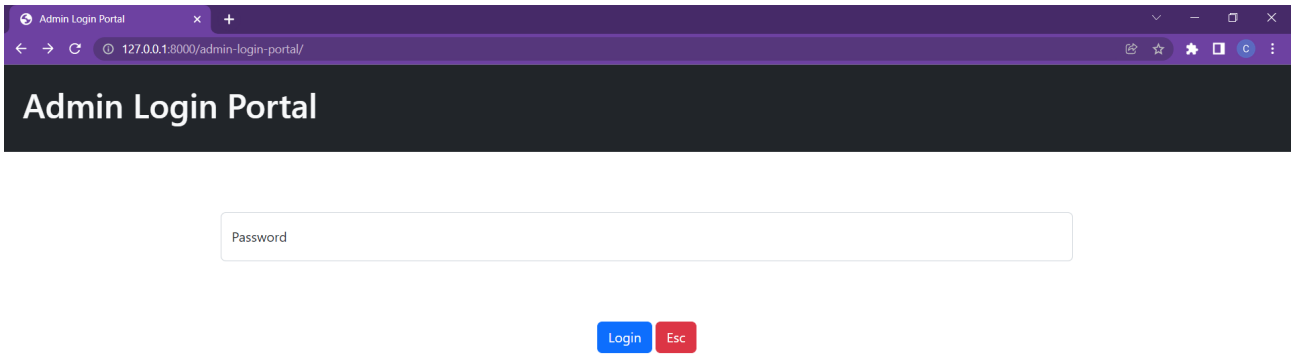

SDO Login Portal
127.0.0.1:8000/sdo-login-portal/
SDO Login Portal
District Zone
Password
Login
Esc

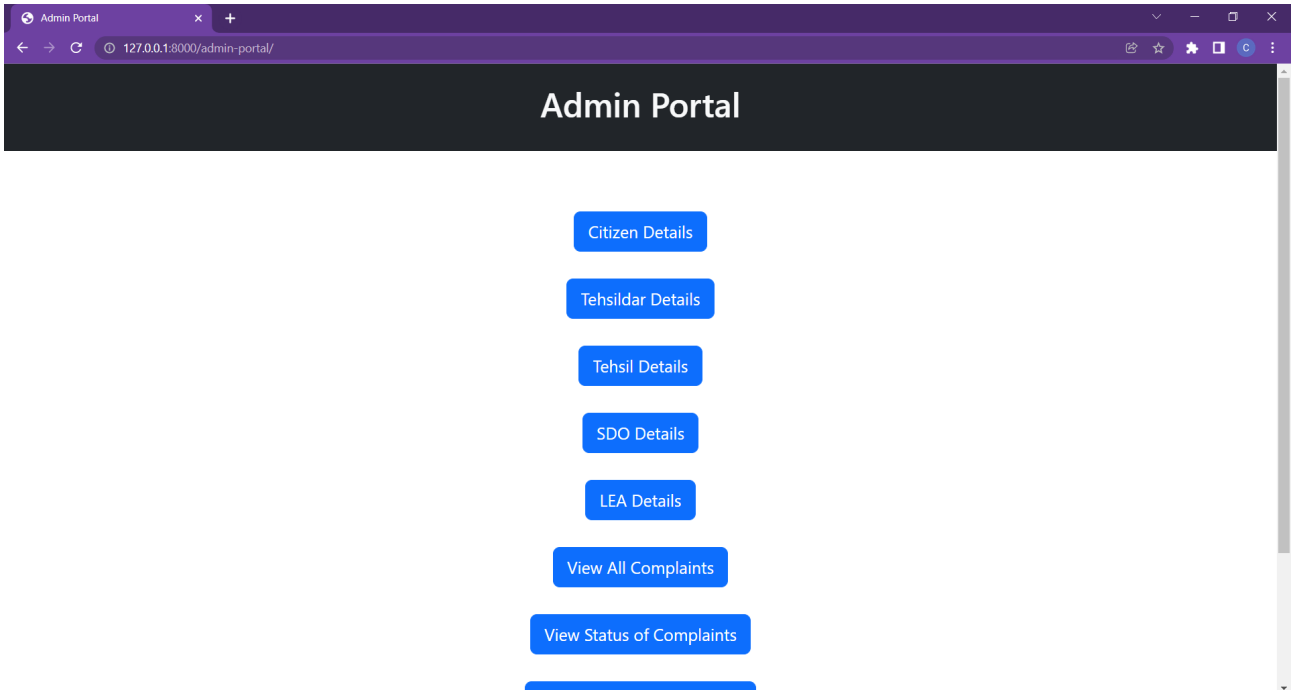

SDO Portal
127.0.0.1:8000/sdo-portal/
SDO Portal
SDO Details
Tehsil and Tehsildar Details
View All Unsolved Complaints
View No. of Solved Complaints
Log Out

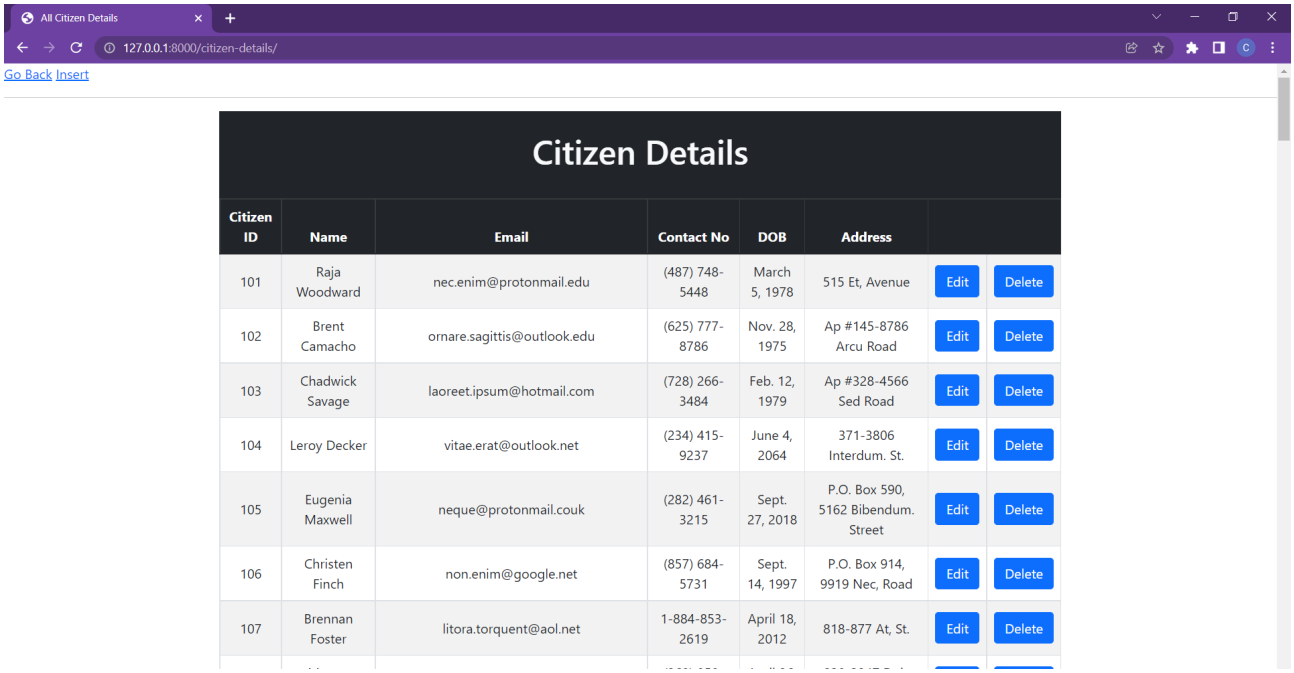


Tehsil and Tehsildar Details

Tehsil ID	Tehsil Name	Tehsil Off. Add.	Zone	Tehsildar ID	Tehsildar Name	Tehsildar Email	Tehsildar Contact no	Tehsildar Salary
1	Wisconsin	Ap #212-2909 Vestibulum. St.	North	1	Wayne Patton	ac.tellus.suspendisse@aol.ca	1-786-984-9967	36966
2	Nebraska	435-5874 Ipsum Rd.	Central	2	Galena Stout	ligula@outlook.edu	1-932-532-4148	38940
3	Delaware	P.O. Box 876, 5058 Auctor Road	East	3	Illiana Witt	semper.auctor@icloud.net	1-649-472-5688	16081
4	Alaska	869-7538 Vel, Rd.	North	4	Octavius Hanson	aliquet.molestie@aol.couk	(621) 592-5757	57960
5	Arkansas	P.O. Box 692, 5435 Est, Av.	Central	5	Alika Benjamin	sit.amet@yahoo.edu	1-634-678-2986	15190
6	Tennessee	Ap #980-1711 Metus. Av.	Central	6	Justin Booth	feugiat.metus@aol.edu	(636) 490-6722	70062
7	Montana	583-393 Leo. Rd.	North	7	Alvin Norton	natoque@hotmail.ca	(855) 998-5836	31066



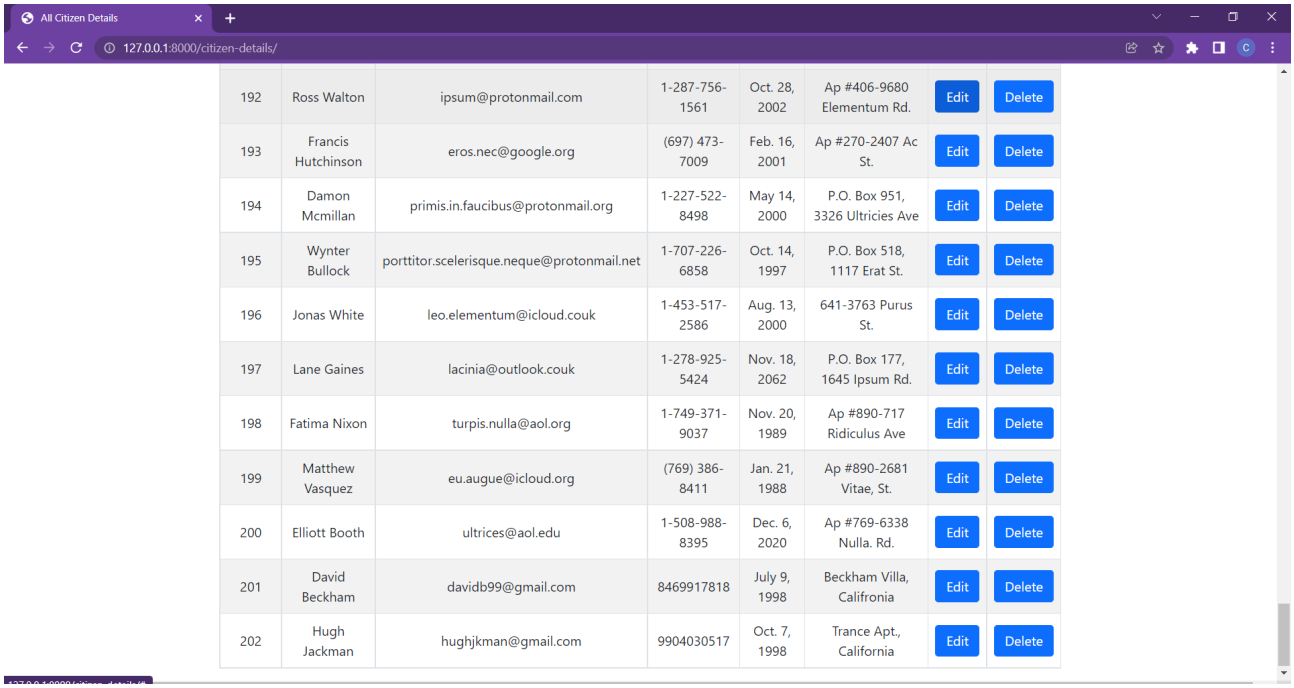




Citizen ID	Name	Email	Contact No	DOB	Address		
101	Raja Woodward	nec.enim@protonmail.edu	(487) 748-5448	March 5, 1978	515 Et, Avenue	Edit	Delete
102	Brent Camacho	ornare.sagittis@outlook.edu	(625) 777-8786	Nov. 28, 1975	Ap #145-8786 Arcu Road	Edit	Delete
103	Chadwick Savage	laoreet.ipsum@hotmail.com	(728) 266-3484	Feb. 12, 1979	Ap #328-4566 Sed Road	Edit	Delete
104	Leroy Decker	vitae.erat@outlook.net	(234) 415-9237	June 4, 2064	371-3806 Interdum. St.	Edit	Delete
105	Eugenia Maxwell	neque@protonmail.couk	(282) 461-3215	Sept. 27, 2018	P.O. Box 590, 5162 Bibendum. Street	Edit	Delete
106	Christen Finch	non.enim@google.net	(857) 684-5731	Sept. 14, 1997	P.O. Box 914, 9919 Nec, Road	Edit	Delete
107	Brennan Foster	litora.torquent@aol.net	1-884-853-2619	April 18, 2012	818-877 At, St.	Edit	Delete

Citizen ID	202
Name	Hugh Jackman
Email	hughjkman@gmail.com
Contact no	9904030517
Date of Birth	07-10-1998
Address	Trance Apt., California
Insert Record	

Citizen ID	Enter Citizen ID
Name	Enter Name
Email	Enter email
Contact no	Enter Contact no
Date of Birth	dd-mm-yyyy
Address	Enter Address
Insert Record	**Citizens 202 is saved successfully!**



192	Ross Walton	ipsum@protonmail.com	1-287-756-1561	Oct. 28, 2002	Ap #406-9680 Elementum Rd.	Edit	Delete
193	Francis Hutchinson	eros.nec@google.org	(697) 473-7009	Feb. 16, 2001	Ap #270-2407 Ac St.	Edit	Delete
194	Damon Mcmillan	primis.in.faucibus@protonmail.org	1-227-522-8498	May 14, 2000	P.O. Box 951, 3326 Ultricies Ave	Edit	Delete
195	Wynter Bullock	porttitor.scelerisque.neque@protonmail.net	1-707-226-6858	Oct. 14, 1997	P.O. Box 518, 1117 Erat St.	Edit	Delete
196	Jonas White	leo.elementum@icloud.couk	1-453-517-2586	Aug. 13, 2000	641-3763 Purus St.	Edit	Delete
197	Lane Gaines	lacinia@outlook.couk	1-278-925-5424	Nov. 18, 2062	P.O. Box 177, 1645 Ipsum Rd.	Edit	Delete
198	Fatima Nixon	turpis.nulla@aol.org	1-749-371-9037	Nov. 20, 1989	Ap #890-717 Ridiculus Ave	Edit	Delete
199	Matthew Vasquez	eu.augue@icloud.org	(769) 386-8411	Jan. 21, 1988	Ap #890-2681 Vitae, St.	Edit	Delete
200	Elliott Booth	ultrices@aol.edu	1-508-988-8395	Dec. 6, 2020	Ap #769-6338 Nulla. Rd.	Edit	Delete
201	David Beckham	davidb99@gmail.com	8469917818	July 9, 1998	Beckham Villa, Califronia	Edit	Delete
202	Hugh Jackman	hughjkman@gmail.com	9904030517	Oct. 7, 1998	Trance Apt., California	Edit	Delete