CHAPTER-I

INTRODUCTION

In the current scenario, the key issue faced by public while travelling is to deal with the inefficiency and discrepancy in the traffic system and the people involved in managing the same. On being caught by the police, the commuter must submit his driving license, RC Book and/or other vehicle documents for verification. In case the commuter forgets/misplaces the documents, he is unnecessarily fined. In case a vehicle is booked for any traffic violations, the details are uploaded onto a website. The owner is not updated with this information. If he fails to check this website regularly, he may not be aware of any pending traffic violations on his vehicles. In case a vehicle is stolen, the owner must contact the nearest police station. The process to lodge a complaint and subsequent response is slow and inefficient. In this age of science and technology, where the internet is in the palm of everyone's hands, a better system for managing the above process can be set up which would greatly reduce the burden on the daily commuters as we as the traffic police.

1.1 OVERVIEW OF THE PROJECT

Now a day's population has become a major factor to be considered as a result the number of vehicle's are growing by increasing problems of vehicle registration, license registration, emission testing and insurance validity for RTO departments and vehicle related documents verification by traffic police. RTO employees having lot of work burden of making registration, license issue, transfer etc., which requires lot of paper work.

Police can view and check that details like (License, Insurance and RC Book) and they can enter their punishment also. These databases are handled by RTO admin. Once police enter public punishment those details are updated to license. Police can only enter punishment details and check Insurance Expiry date and RC book validation details. So, it will be reduced time consuming for police.

1.2 OBJECTIVE OF THE PROJECT

The proposed system overcomes these issues in the current scenario by implementing a web server which uses a database to store, update and access the above-mentioned documents with a user-friendly front-end web application, tailored to the needs of the appropriate users. The application also allows users to report a stolen vehicle and check unpaid offences on his vehicle, all in the click of a button.

The application for the police allows him to review earlier driving offences by the rider and also report any current offence committed by him. The police's application generates a digital receipt avoiding any kind of unwanted tampering with the process. The system also allows integration with emission test centers and insurance companies who can upload verified emission test documents and insurance details directly on to the database.

1.3 ORGANIZATION PROFILE

ORGANIZATION PROFILE:

Nitroware is a software development company in India started with business acumen in providing a user-friendly feature and customized solutions for small, medium, and large business. The company provides full-cycle services in the areas of customized software development, web designing, web development, SEO and software solutions and services.

MISSION:

The company mission is to "Enhancing the business growth of our customers with creative design and development to deliver market-defining high-quality solutions that create value and reliable competitive advantage for our clients around the world".

VISION:

The company vision is to "develop in a constant manner and grow as a major IT service provider to become a leading performer, in a providing quality web and software development solutions in the competitive global marketplace.

CHAPTER-II

SYSTEM ANALYSIS

2.1 BACKGROUND STUDY

2.1.1 EXISTING SYSTEM

In the existing level, due to the population the number of vehicle’s are growing by increasing problems of vehicle registration, license registration, emission testing and insurance validity for RTO departments and vehicle related documents verification by traffic police. RTO employees having lot of work burden of making registration, license issue, transfer etc., which requires lot of paper work.

Also, the existing system mainly focuses on providing the information only to the traffic police officers. It consists of vehicle information and license information's and It also generates the fine.

DISADVANTAGES

Drawbacks of the existing system are as follows

- Manual work will not give accuracy.
- It will take more time for traffic police as well as public people to complete the process.

2.1.2 PROPOSED SYSTEM

The proposed system overcomes these issues in the current scenario by implementing a web server which uses a database to store, update and access the above-mentioned documents with a user-friendly front-end web application, tailored to the needs of the appropriate users. The application also allows users to report a stolen vehicle and check unpaid offences on his vehicle, all in the click of a button. The application for the police allows him to review earlier driving offences by the rider and also report any current offence committed by him.

The main theme of this proposed system is to allows users to report a stolen vehicle details to the crime police. This can help notify the police in the nearby locality faster and help track the vehicle in a more efficient manner. It provides a web page to users to update the stolen status of vehicle for investigation purpose. Push notification can be used to send stolen vehicle reports to the police.

ADVANTAGES:

- Police can view and check the details like License, and RC Book and also enter their punishment easily.
- It will reduce time consuming for police as well as public people.
- At anywhere anytime police can view public documents.
- This application also allows users to report a stolen vehicle and check unpaid offences on his vehicle.
- Public can able to check the status of their complaints, if any theft occurs on their vehicle.

2.2 SYSTEM SPECIFICATION

2.2.1 HARDWARE SPECIFICATION

Processor	: Core i3
RAM Capacity	: 1GB
Hard Disk	: 160 GB
Mouse	: Logical Optical Mouse
Keyboard	: 104 Keys
Monitor	: 16 inch
Mother Board	: Intel

Speed	: 2GHZ
Floppy Disk Drive	: 100 MB

2.2.2 SOFTWARE SPECIFICATION

Operating System	: Windows 10
Front End	: HTML, CSS
Backend	: MySQL

2.2.3 APPLICATION SPECIFICATION

FRONT END: HTML, CSS

HTML

HTML was first created by Tim Berners-Lee, Robert Caillou, and others starting in 1989. It stands for Hyper Text Markup Language. Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether. The latest version is known as HTML5. A Markup Language is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and attributes.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML.

BENEFITS

The following are the advantages of HTML5

- Web pages development. HTML is heavily used for creating pages that are displayed on the world wide web.
- Web document Creation.
- Internet navigation.
- Cutting edge feature.
- Responsive images on web pages.
- Client-side storage.
- Offline capabilities usage.
- Data Entry support with HTML.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

CSS Selectors

CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

PYTHON

Python is an integrated, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

DJANGO (WEB FRAMEWORK)

Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

The model layers

Django provides an abstraction layer (the "models") for structuring and manipulating the data of your Web application. Learn more about it below:

- Models: Introduction to models, Field types, Indexes, Meta options, Model class
- Query Sets: Making queries, Query Set method reference, Lookup expressions
- Model instances: Instance methods, Accessing related objects
- Migrations: Introduction to Migrations, Operations reference, Schema Editor, Writing migrations

BACKEND: MySQL

MySQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.

Design

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

Develop

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.

Database Migration

MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL, and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL both on Windows and other platforms. Migration also supports migrating from earlier versions of MySQL to the latest releases.

Insert Query

Now that you have a table in your database, let us see how you can insert values into that table. You might have used insert query many times, but have you used an insert query that not only inserts data but can also updates data if need be? This is achieved by using the keyword "on DUPLICATE KEY."

JOINS

Joins are supposed to be complicated because there is so much variety in them. Joins are known by different names in different databases: self-join, outer join, inner join, natural join, and so many others. We will use a join between two tables, which is generally used to pull data.

Adding Indexing and Search Queries

Creating indexes for a column undoubtedly increases incoming queries to the database, and indexes are created on a column basis. For example, in our sample table "emp", if most of the queries fetch data using the column id, then it would be a wise decision to create an index for the id column.

2.3 COST ESTIMATION AND SCHEDULING

Software cost is related to many variables such as Human, Technical, Environment, and Effort applied to develop it. The estimates of cost depend, on our ability to estimate and evaluate several factors, given below.

2.3.1 NUMBER OF USER INPUTS

- Number of user output
- Experience and ability of the project personnel
- The quality of the software development environment
- The degree to which software components can be reused

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

Let us see how to compute the manpower estimation and the cost estimation of the project "**VEHICLE DOCUMENT AUTHENTICATION AND VERIFICATION SYSTEM**" is described briefly below.

COCOMO, Constructive Cost Model, is a good measure for estimating the number of person-months required to develop software. COCOMO consists of a hierarchy of three increasingly detailed and accurate forms.

The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for the difference in project attributes (Cost Drivers).

DESCRIPTION OF TASK	NO OF DAYS
Abstract	5
Problem Statement	3
System Requirements	2
Design	15
Coding	80
Implementation	10
Testing	15
Reports	10
Deployment	5
Scope	5
Total	150

2.3.2 THE COCOMO COST ESTIMATION FORMULA IS

E = c*size k

Where, E = effort in person-months.

The effort measure helps to make estimates like the number of person-months that will take for the project to execute.

The size estimate is converted into effort estimates.

c = 3.0 for semidetached mode k= 1.12 in semidetached mode Size= (SLOC)/1000=3.0 Thus the effort for making VEHICLE DOCUMENT AUTHENTICATION AND VERIFICATION SYSTEM

Subscription D = a*Eh

Where,

D= Development time in chronological months a = 2.5 in semi-detached mode h =

0.38 in semi-detached mode No of days worked= 150 days

1-day work =5 Hours

Total no of hours = 750 Hours Cost for 1 hour = ₹50

The total number of costs=Total number of hours*cost for 1 hour = 750*50 = ₹37,500

Therefore the total cost of the product is ₹45,000.

CHAPTER III

SYSTEM DESIGN AND DEVELOPMENT

3.1 DESIGN PROCESS

3.1.1 DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

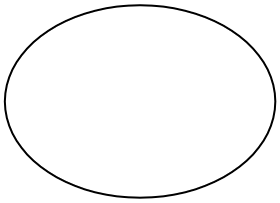
A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design.

CONSTRUCTING A DFD

Several rules of thumb are used in drawing DFD’S

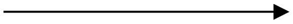
- Process should be named and numbered for an easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal.
- When a process is exploded into lower level details, they are numbered.
- The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

NOTATIONS

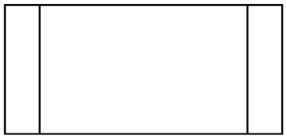


Process

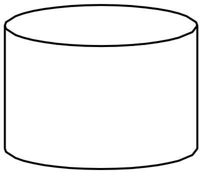
Source or Destination of data



Data Flow



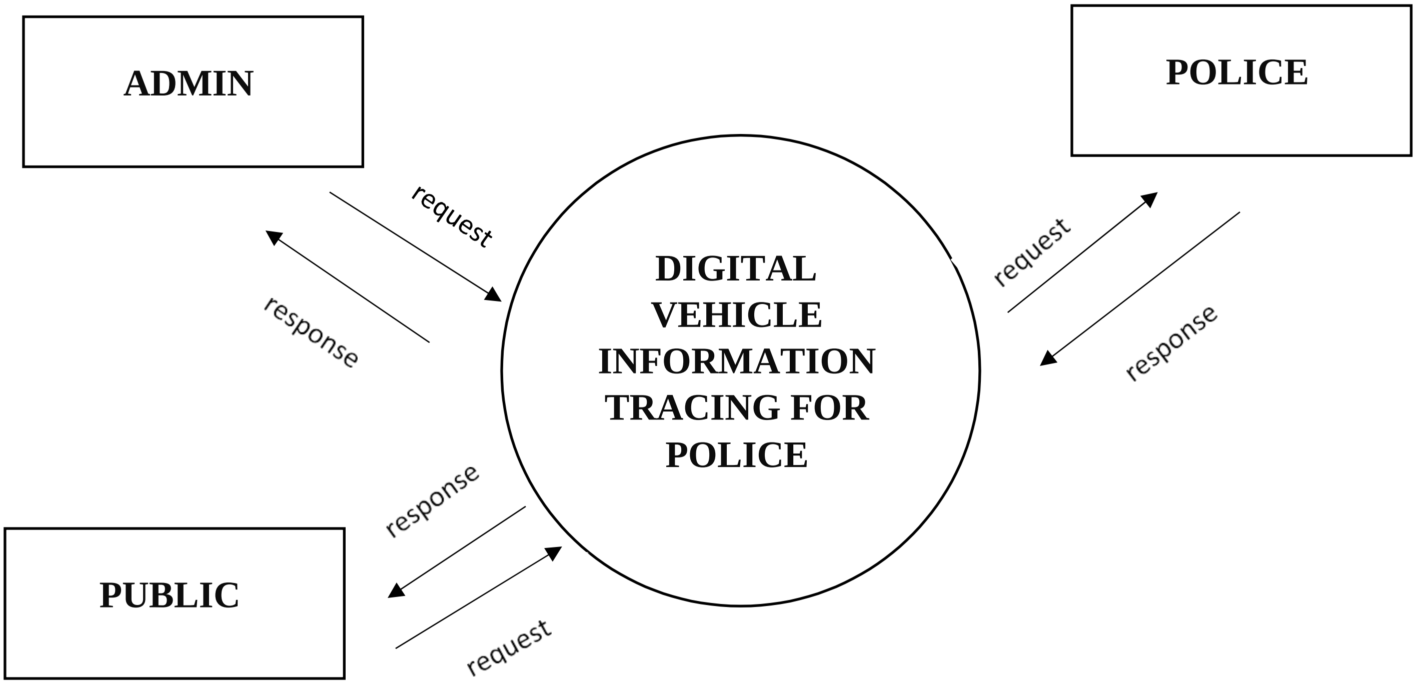
Data Table



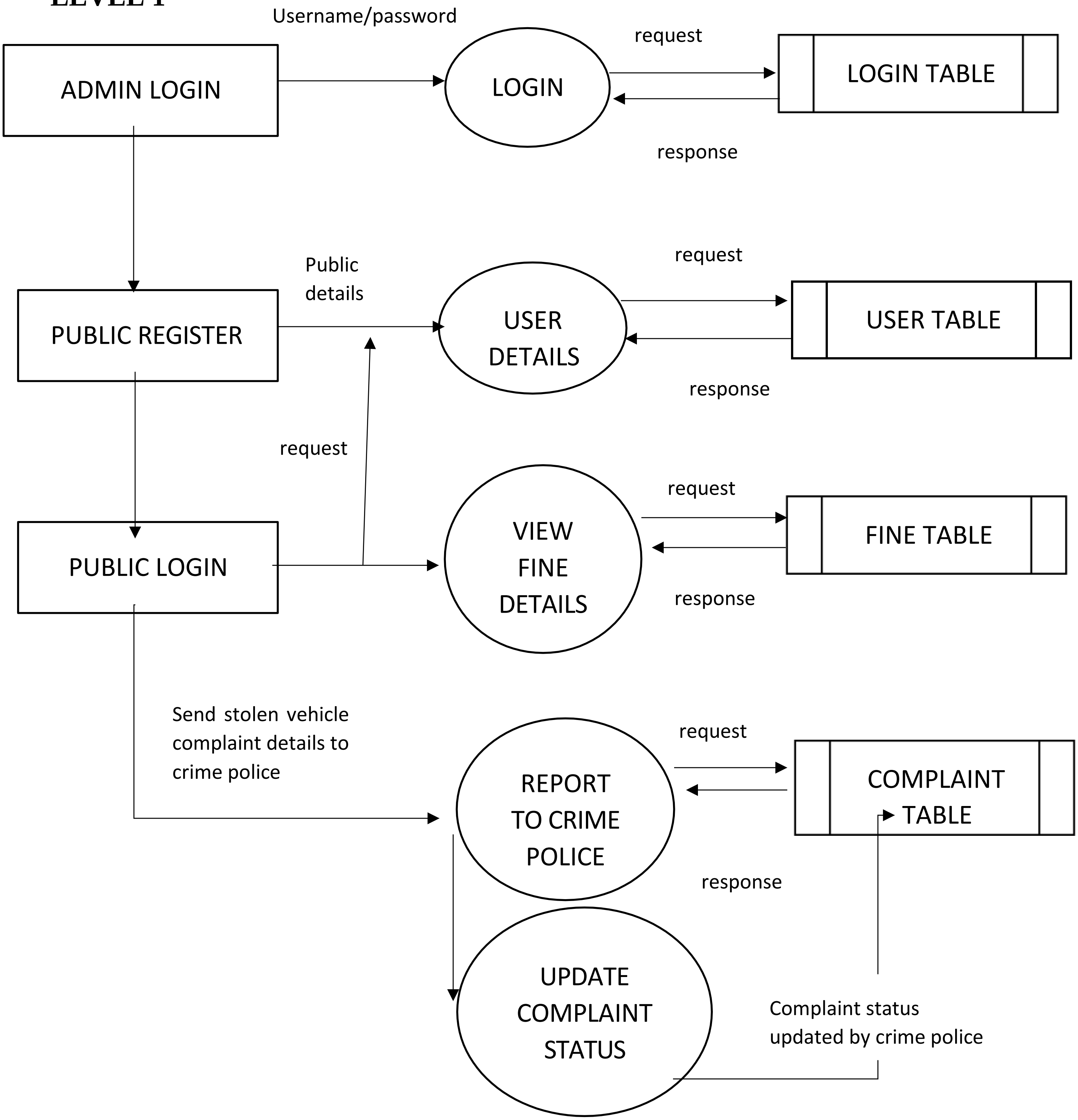
Database

CONTEXT DIAGRAM

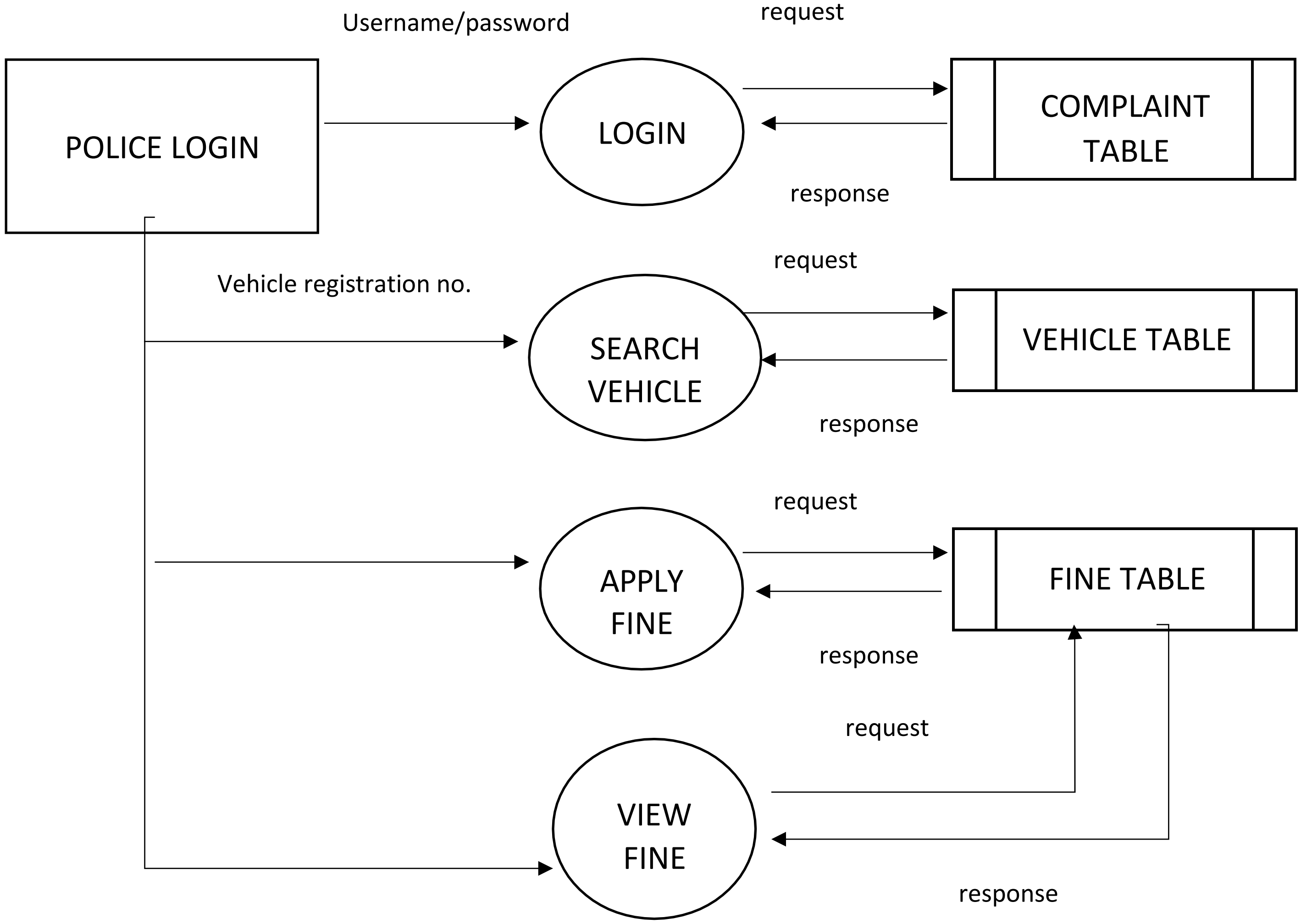
LEVEL 0



LEVEL 1


Username/password
ADMIN LOGIN
LOGIN
request
LOGIN TABLE
response
Public details
PUBLIC REGISTER
USER DETAILS
request
USER TABLE
response
request
PUBLIC LOGIN
VIEW FINE DETAILS
request
FINE TABLE
response
Send stolen vehicle complaint details to crime police
REPORT TO CRIME POLICE
request
COMPLAINT TABLE
response
UPDATE COMPLAINT STATUS
Complaint status updated by crime police

LEVEL 2


Username/password
request
POLICE LOGIN
LOGIN
COMPLAINT TABLE
response
request
Vehicle registration no.
SEARCH VEHICLE
VEHICLE TABLE
response
request
APPLY FINE
FINE TABLE
response
request
VIEW FINE
response

3.2 TABLE STRUCTURE

LOGIN TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	Username	Char Field	User ID	Primary Key, Not null
2	Password	Char Field	Password	Not null

USER TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	User name	Char Field	User ID	Unique
2	Password	Char Field	Password	Not Null
3	Email	Email Field	Email Id	Not Null
4	Phone No	Integer Field	Mobile Number	Not Null
5	Country	Char Field	Country	Not Null
6	State	Char Field	State	Not Null
7	City	Char Field	City	Not Null
8	Address	Char Field	Address	Not Null
9	Lic no	Char Field	License Number	Not Null, Unique
10	Upload image	File Field	Upload the owner’s image	Null
11	Upload RC Book	File Field	Upload the RC Book	Null

VEHICLE TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	Vehicle.Num.Plate	Char Field	Vehicle Number Plate	Unique
2	Owner Name	Char Field	Vehicle’s Owner Name	Not Null
3	Country	Char Field	Country	Not Null
4	State	Char Field	State	Not Null
5	City	Char Field	City	Not Null
6	Address	Char Field	Address	Not Null
7	Lic.No	Char Field	License Number	Not Null, Unique
8	Lic.Expire.Date	Date Field	License Expire Date	Null
9	Vehicle.Reg.No	Char Field	License Registration Number	Null
10	Upload Owner image	File Field	Upload the owner’s image	Null
11	Upload RC Book	File Field	Upload the RC Book	Null
12	Upload Vehicle image	File Field	Upload the Vehicle’s Image	Null

POLICE TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	User name	Char Field	User ID	Unique
2	Password	Char Field	Password	Not Null
3	Email	Email Field	Email Id	Not Null
4	Phone No	Integer Field	Mobile Number	Not Null
5	Country	Char Field	Country	Not Null
6	State	Char Field	State	Not Null
7	City	Char Field	City	Not Null
8	Address	Char Field	Address	Not Null
9	Police Id	Char Field	Police Identification Number	Not Null, Unique

CRIME POLICE TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	Police name	Char Field	Police name	Unique
2	Designation	Char Field	Designation	Not Null
3	Email	Email Field	Email Id	Not Null
4	Phone No	Integer Field	Mobile Number	Not Null
5	Address	Char Field	Address	Not Null
6	State	Char Field	State	Not Null
7	City	Char Field	City	Not Null
8	Duty area	Char Field	Duty area	Unique, Not Null

FINE TABLE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	Fine Name	Char Field	Fine Name	Not null
2	Amount	Float Field	Amount	Not null

APPLY FINE

S. NO.	FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
1	Fine Name	Char Field	Fine Name	Foreign Key
2	Amount	Float Field	Amount	Foreign Key
3	Due Date	Date Field	Due Date	Not null
4	Status	Char Field	Status of the fine	Not null

3.3 FILE DESIGN

The file design is one of the important features which mainly depend on the performance of the system. The file design deals with the two important elements. They are the size of the files and the redundancy of the data. At the same time all the files are design to incorporate all relevant information regarding each module. A single database with information about all the modules will make the system complex. The design base files are the most important of the system. The performance of the system depends on how the system is designed. It has been given at most attention to reduce the size of files and redundancy. At the same time all the files are design to incorporate all relevant information regarding each entity.

Home

It is used to set an index page.

Dashboard

It is used to set a template in the web page

Public Register

It is used to register into this application for the public people through the registration form. It is used to save public details such as name, password, city, phone number, country, email ID, address, license number, upload rc book, upload image.

Public Login

Public has to give valid username and password for proper login.

Police Login

Police has to give valid username and password for proper login.

Admin Login

Admin has to give valid username and password for proper login.

3.4 INPUT DESIGN

Input is any data or instructions entered into the memory of a computer. Two types of input are data and instructions. Data is a collection of unorganized items that can include words, numbers, pictures, sounds, and video. A computer processes data into information, which is organized, meaningful, and useful. Instructions can be in the form of programs, commands, or user responses.

Public_Register

It is used to save public details such as name, password, city, phone number, country, email ID, address, license number, upload rc book, upload image.

Search_Vehicle

Search_Vehicle is a function, which takes vehicle registration number as an input and it will produce the vehicle details.

Apply_Fine

Police can update fine amount, if there is any issue occurs.

Rise_Complaint

It is used to rise complaint to the police, about the stolen vehicle. The complaint details consist of owner name, phone number, address, vehicle type, brand, registration number, engine number, chassis number, date and place (where the vehicle get lost).

3.5 OUTPUT DESIGN

Output design will be done through reports in the system. The system's output will be generated either through reports or through system prompted messages. The system will be designed in such a way that output forms will be designed for displaying outputs to the user. Output will be generally displayed to the user through monitor as visual display. Output can also be an attached email or bill from the printer.

View_fine

Police and public can able to view the fine amount on the vehicle from their dashboard.

View_complaint

Public people can able to view the complaint status of their vehicle, which is updated by the crime police.

View_vehicle_details

Police can able to view the vehicle details like license and rc book easily.

Send_mail

The information of the stolen vehicle will be sent to the crime police through the mail.

3.6 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing the input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy, quick, inexpensive and flexible for other users.

Login Table

Login table consists of username and password fields for authentication. User_ID is considered as primary key in this table.

User Table

User table consists of username, password, email, phone number, country, state, city, address, license number(unique), upload image, upload rc book.

Vehicle Table

Vehicle table consists of vehicle number plate, owner name, country, state, city, address, license number(unique) , license expire date, vehicle registration number, upload rc book, upload vehicle image, upload owner image.

Police Table

Police table consists of username, password, email, phone number, address, country, state, city, police id. Police id is taken as a unique one.

Crime Police Table

Crime police table consists of police name, designation, email, phone number, address, state, city, duty area. In this table duty area is taken as a unique one.

Fine Table

Fine table consists of fine name and amount.

Apply Fine Table

Apply Fine table consists of fine name (foreign key), amount (foreign key), duty area, and status.

3.7 SYSTEM DEVELOPMENT

3.7.1 DESCRIPTION OF MODULES

Proposed project has various modules. They are as follows:

- ➢ Admin Module
- ➢ Police Module
  - Search Vehicle
  - Apply Fine
  - View Fine
- ➢ Public module(user)
  - Fine Details
  - View Fine
  - Rise Complaint
  - View Complaint status

ADMIN MODULE

Admin module allows system administrator to set up back end systems and basic system configurations and it maintains all the module information and valid registration process. Admin module has two sub process. They are as follows,

1. **Admin Login**

In log in page, admin can manage all information. They can update or edit any information.

2. **Upload public documents and Maintain database**

Admin Can upload the xlsx data base of all digital vehicle.

POLICE MODULE

Police Module is one of the main module in this application. Here, the police can only able to login in to this module using their respective username and password. There are several sub process can be used by the police based on their requirements. The sub processes are Police login , Search vehicle details, apply fine, check the vehicle documents like license and rc book, and View the fine details of the public people.

Firstly, Using username and password police can able to login to this application. After the login process police can able to see the dashboard which consist of several options like search vehicle and view fine. Based on the police wish, he can able to choose their option and do the process.

In this module, there is an option called ‘search vehicle’ , by clicking that, the police can able to search the vehicle details of the public people using vehicle registration number. It will produce the details of the respective vehicle, which the police want to check. After the checking process, if there are any issues occurs means, the police can able to apply fine on the respective vehicle, using the button called ‘apply fine’.

1. **Police Login**

In log in page, police can only view the shared information.

2. **search vehicle**

police can search and view the vehicle details like RC Book and license expiry date details

3. **Update punishment**.

Police can update fine details for public.

4. **View punishment**

Police can view the fine details and also checks the status of applied fine, whether the fine amount is paid or not.

PUBLIC MODULE

Public module is the main module of this application. Using the username and password public can able to login in to this application. Before the public login process public should register their details with this application using the option called ‘Public Register’. After the registration only the public can able to get an access to use this application in an essential manner.

There are various options are available in the public module. They are Fine details, view fine, report, complaint. If the public select the File details option, it will show the list of fine names and their amounts, which is used to know the general fine details followed by the government. Then the next option called, ‘view fine’, by clicking that the public can able to check, whether there is any fine or offense applied on his vehicle or not. Then the next option is called ‘report’ ,through this option public can able to report to the crime police about the stolen vehicle. The next option is called ‘complaint’ through that public can able to check the status of the applied complaint. The status of the stolen vehicle report will be updated by the crime police.

1. Login

Using the user id, the user login to the application.

2. fine details

User can view all general fine details on the vehicle and also their own vehicle’s fine detail.

3. Report

User can Report stolen vehicle to notify the nearest police.

4. Check complaints

User can check any unpaid complaints on his vehicle.

CHAPTER IV

TESTING AND IMPLEMENTATION

The common view of testing held by users is that it is performed to prove that there are no errors in a program. This is extremely difficult since designer cannot prove to be one hundred percent accurate. Therefore, the most useful and practical approach is with the understanding that testing is the process of executing a program with explicit intention of finding errors.

Testing has its own cycle. The testing process begins with the product requirements phase and from there parallels the entire development process. In other words, for each phase of the development process there is an important testing activity. Successful testing requires a methodical approach. It requires focusing on basic critical factors:

- Planning
- Project and process control
- Risk management
- Inspections
- Measurement tools
- Country and professionalism

TEST PLAN

Before going for testing, first we have to decide upon the type of testing to be carried out. The following factors are taken into consideration:

- To ensure that information properly flows into and out of program

- To find out whether the local data structures maintains its integrity during all steps in an algorithm execution

- To ensure that the module operate properly at boundaries established to limit or restrict processing

- To find out whether error - handling paths are working correctly or not

4.1 UNIT TESTING

Unit or module testing is the process of testing the individual components (subprograms or procedures) of a program. The purpose is to discover discrepancies between the modules interface specification and its actual behavior. In our system each module namely is tested independently for validation.

4.2 VALIDATION TESTING

Validation testing provides the final assurance that software meets all functional, behavioral and performance requirement. The software once validated must be combined with other system elements. System testing verifies that as elements combine properly and that overall system function and performance is achieved.

4.3 INTEGRATION TESTING

Integration testing is the process of combining and testing multiple components together. The primary objective of integration testing is to discover errors in the interfaces between the components. In our system each of the modules mentioned above, are tested for checking the integration between them, after each of them are tested individually.

4.4 TEST CASES

LOGIN TABLE

TEST ID	TEST DATA	DESCRIPTION	ACTUAL VALUE	EXPECTED VALUE	RESULT
TC-01	Username	To test the field username	Enter the values in character number and symbols.	Enter the values in character number and symbols.	True
TC-02	Username	To test the field username	No values entered	Enter the values in character number and symbols.	False
TC-03	Password	To test the field password	Enter the values in character number and symbols.	Enter the values in character number and symbols.	True

TC-04	Password	To test the field password	No values entered	Enter the values in character number and symbols.	False
TC-05	Login(button)	To test the Login button	Click enter	Click enter	True

USER TABLE

TEST ID	TEST DATA	DESCRIPTION	ACTUAL VALUE	EXPECTED VALUE	RESULT
TC-01	User name	To test the field username	Enter the values in character number and symbols.	Enter the values in character number and symbols.	True
TC-02	User name	To test the field username	No values entered	Enter the values in character number and symbols.	False
TC-03	Password	To test the field password	Enter the values in character number and symbols.	Enter the values in character number and symbols	True
TC-04	Password	To test the field password	No values entered	Enter the values in character number and symbols	False

TC-05	Email	To test the field email	Enter the values in character, number, @symbol	Enter the values in character, number, @symbol	True
TC-06	Phone No	To test the field phone number	Enter the number only.	Enter the values in number	True
TC-07	Phone No	To test the field phone number	Enter the characters	Enter the values in number	False
TC-08	Country	To test the field country	Enter character with numbers	Enter only Characters	False
TC-09	State	To test the field State	Enter Characters only	Enter only Characters	True
TC-10	State	To test the field State	Enter character with numbers	Enter only Characters	False
TC-11	Address	To test the field address	Enter character with number	Enter character with number	True
TC-12	Address	To test the field address	Enter number only	Enter character with number	False
TC-13	Lic no	To test the field License number	Enter Alphanumeric character	Enter Alphanumeric character	True
TC-14	Lic no	To test the field License number	No values entered	Enter Alphanumeric character	False
TC-15	Upload image	To test the upload image field	Select the image file from the source	Select the image file from the source	True

TC-16	Upload image	To test the upload image field	Enter the character, number	Select the image file from the source	False
TC-17	Upload RC Book	To test the upload RC Book field	Select the image file from the source	Select the image file from the source	True
TC-18	Public Register Button	To test the Public register button	Address field is empty	All the Field should be filled	False

POLICE TABLE

TEST ID	TEST DATA	DESCRIPTION	ACTUAL VALUE	EXPECTED VALUE	RESULT
TC-01	User name	To test the field username	Enter the values in character number and symbols.	Enter the values in character number and symbols.	True
TC-02	User name	To test the field username	No values entered	Enter the values in character number and symbols.	False
TC-03	Password	To test the field password	Enter the values in character number and symbols.	Enter the values in character number and symbols	True

TC-04	Password	To test the field password	No values entered	Enter the values in character number and symbols	False
TC-05	Phone No	To test the field phone number	Enter the number only.	Enter the values in number	True
TC-06	Phone No	To test the field phone number	Enter the characters	Enter the values in number	False
TC-07	Country	To test the field country	Enter character with numbers	Enter only Characters	False
TC-08	State	To test the field State	Enter Characters only	Enter only Characters	True
TC-09	State	To test the field State	Enter character with numbers	Enter only Characters	False
TC-10	Address	To test the field address	Enter character with number	Enter character with number	True
TC-11	Address	To test the field address	Enter number only	Enter character with number	False
TC-12	Police Id	To test the police Id field	No values are entered	Enter the alphanumeric values	False
TC-13	Police login (button)	To test the login button	Invalid password	Enter valid username and password	False

VEHICLE TABLE

TEST ID	TEST DATA	DESCRIPTION	ACTUAL VALUE	EXPECTED VALUE	RESULT
TC-01	User name	To test the field username	Enter the values in character number and symbols.	Enter the values in character number and symbols.	True
TC-02	User name	To test the field username	No values entered	Enter the values in character number and symbols.	False
TC-03	Password	To test the field password	Enter the values in character number and symbols.	Enter the values in character number and symbols	True
TC-04	Password	To test the field password	No values entered	Enter the values in character number and symbols	False
TC-05	Phone No	To test the field phone number	Enter the number only.	Enter the values in number	True
TC-06	Phone No	To test the field phone number	Enter the characters	Enter the values in number	False
TC-07	Country	To test the field country	Enter character with numbers	Enter only Characters	False
TC-08	State	To test the field State	Enter Characters only	Enter only Characters	True

TC-09	State	To test the field State	Enter character with numbers	Enter only Characters	False
TC-10	Address	To test the field address	Enter character with number	Enter character with number	True
TC-11	Address	To test the field address	Enter number only	Enter character with number	False
TC-12	Police Id	To test the police Id field	No values are entered	Enter the alphanumeric values	False
TC-13	Police login (button)	To test the login button	Invalid password	Enter valid username and password	False

4.5 SYSTEM IMPLEMENTATION

System implementation is the important stage of project when the theoretical design is tunes into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover planning

Planning is the first task in the system implementation. Planning is deciding on the method and the time scale to be adapted. At the time of implementation of any system people from different departments and system analysis involve. The line manager controlled through an implementation co-ordinate committee.

The committee consists of ideas, Problems and complaints of user department. It must also consider,

- The implementation of system environment.
- Self-selection and allocation for implementation tasks.
- Consultation with unions and resources available.
- Standby facilities and channels of communication.

CHAPTER V

CONCLUSION

The first important outcome of is to improve the efficiency in traffic policing. The work and burden on the police are greatly reduced as they can now check validated vehicle and driver documents with just the click of a button on the application. It also paves way for an easier and better system for handling traffic related violations as the police can now book the vehicles on the road that are breaking the rules from within the application. Also, as the owner of the vehicle is updated with this information in real time, they can ensure that no fines are due for the vehicles they own.

The system also enables a simplified system for commuters to handle various vehicle documents as they now no longer have to worry about managing the hard copies of each. With every important document present as a soft copy in their mobiles, the commuters can now just show these to the police for verification. Along with all of the above-mentioned benefits, the system now greatly helps vehicle owners lodge a stolen vehicle report from within the app. This can help notify the police in the nearby locality faster and help track the vehicle in a more efficient manner. This integrated system greatly improves efficiency and makes way for a better approach to manage the work of the traffic department. Thus, the major expected outcome is to improve public welfare using the latest technologies.

CHAPTER VI

SCOPE FOR FURTHER ENHANCEMENT

The proposed system can be further enhanced and greatly improved by adding new functionalities and in-app services. By integrating this system with the Google Cloud Messaging (GCM) service, we could enable push notifications which would help notify or send important updates to the police as well as the general users. Push notifications can be used to send stolen vehicle reports to the police as well as bookings or offences on a vehicle to its owner. Also, location tracking (GPS) can be enabled and integrated into the system which would open up a plethora of possibilities to expand the application, with traffic updates and location-based vehicle tracking being a few. Addition services may be added such as in-app service for ambulance assistance where accidents can be quickly reported which would allow both the police and ambulance drivers to respond to the emergency more efficiently. Emergency contact alerts could be added where a user in distress could notify his near and dear ones about an emergency from within the application.

BIBLIOGRAPHY

REFERENCE BOOKS

[1] Garima Pandey, Diksha Dani, “Android Mobile Application Build on Eclipse”, International Journal of Scientific and Research Publications, Vol-4, Issue 2, Feb-2014.

[2] Manjunath S. Patil, Basavaraj K. Madagouda, Vinod C. Desai, “E-RTO Management System”, International Journal of Engineering Research & Technology (IJERT),Vol-2 Issue 7, July-2013.

[3] Suhas Holla, Mahima M. Katti, “Android Based Mobile Application Development and Its Security”, International Journal of Computer Trends and Technology, Vol-3 Issue 3, 2012.

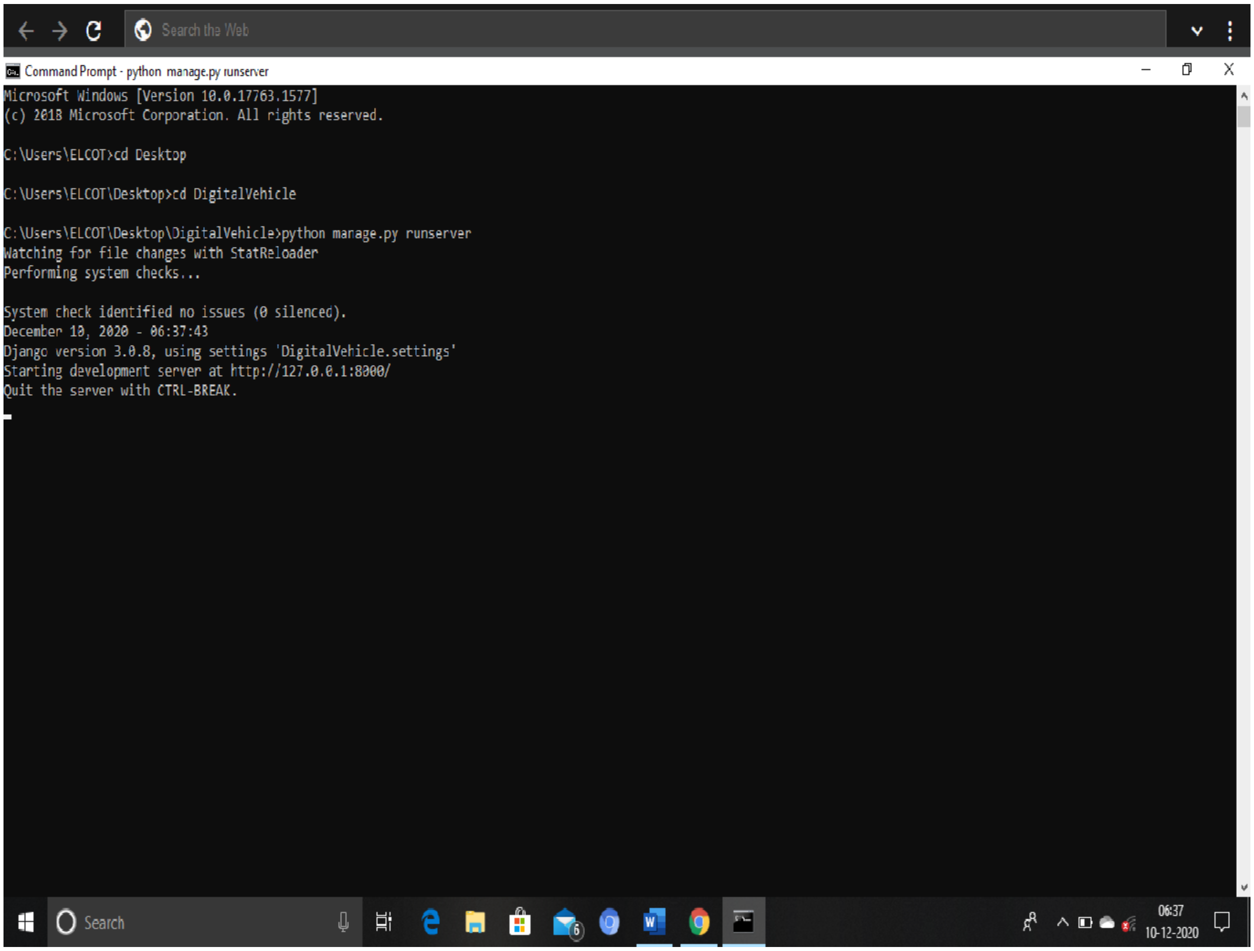
[4] Robert W. Sebesta, “Programming the World Wide Web”, 4th Edition, Pearson Education, 2008.

REFERENCE WEBSITES

- www.tutorialspoint.com/index.html
- www.javatpoint.com
- www.w3schools.com
- www.seminarsonly.com

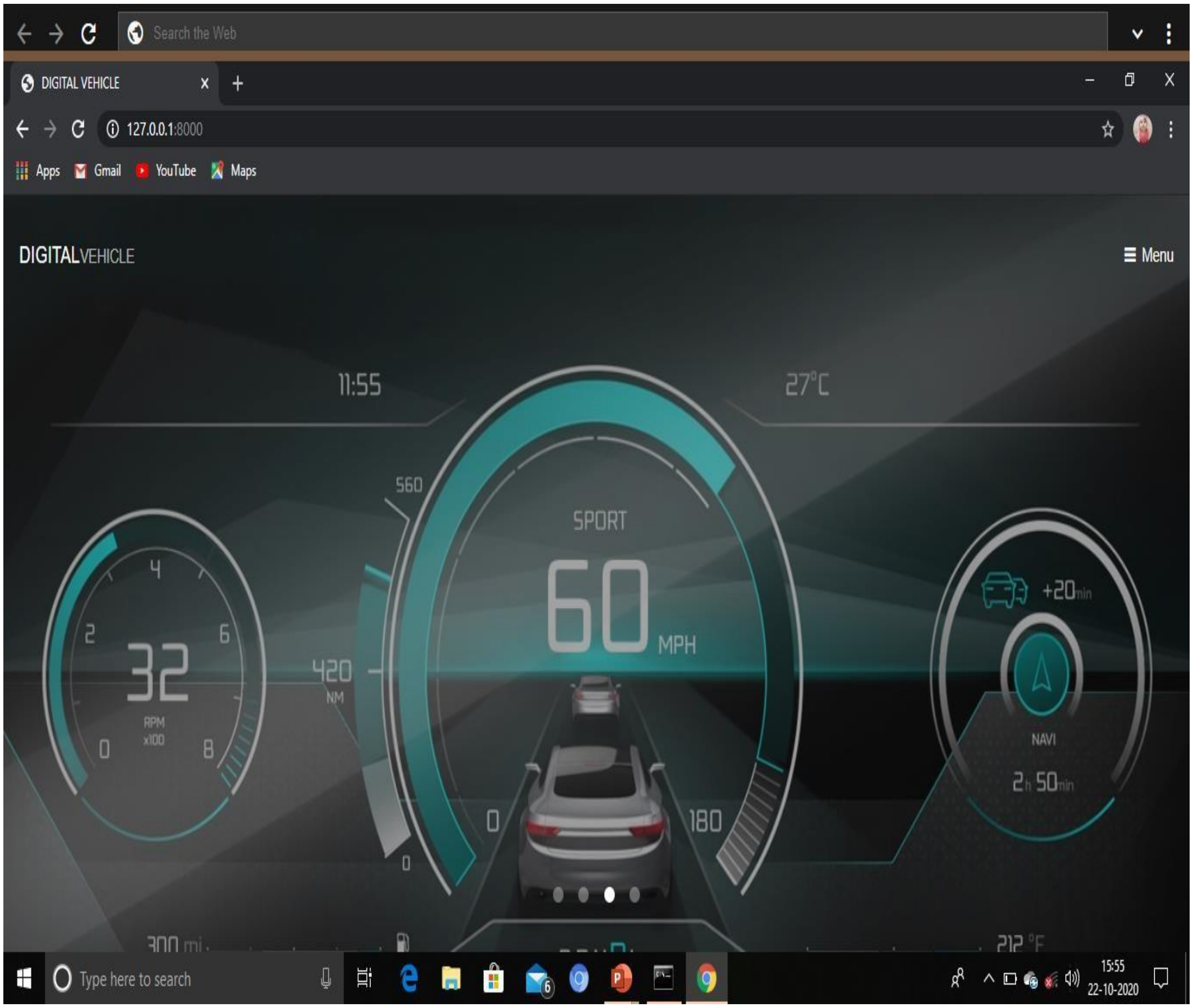
ANNEXURES

A. SAMPLE INPUT - INPUT DESIGN



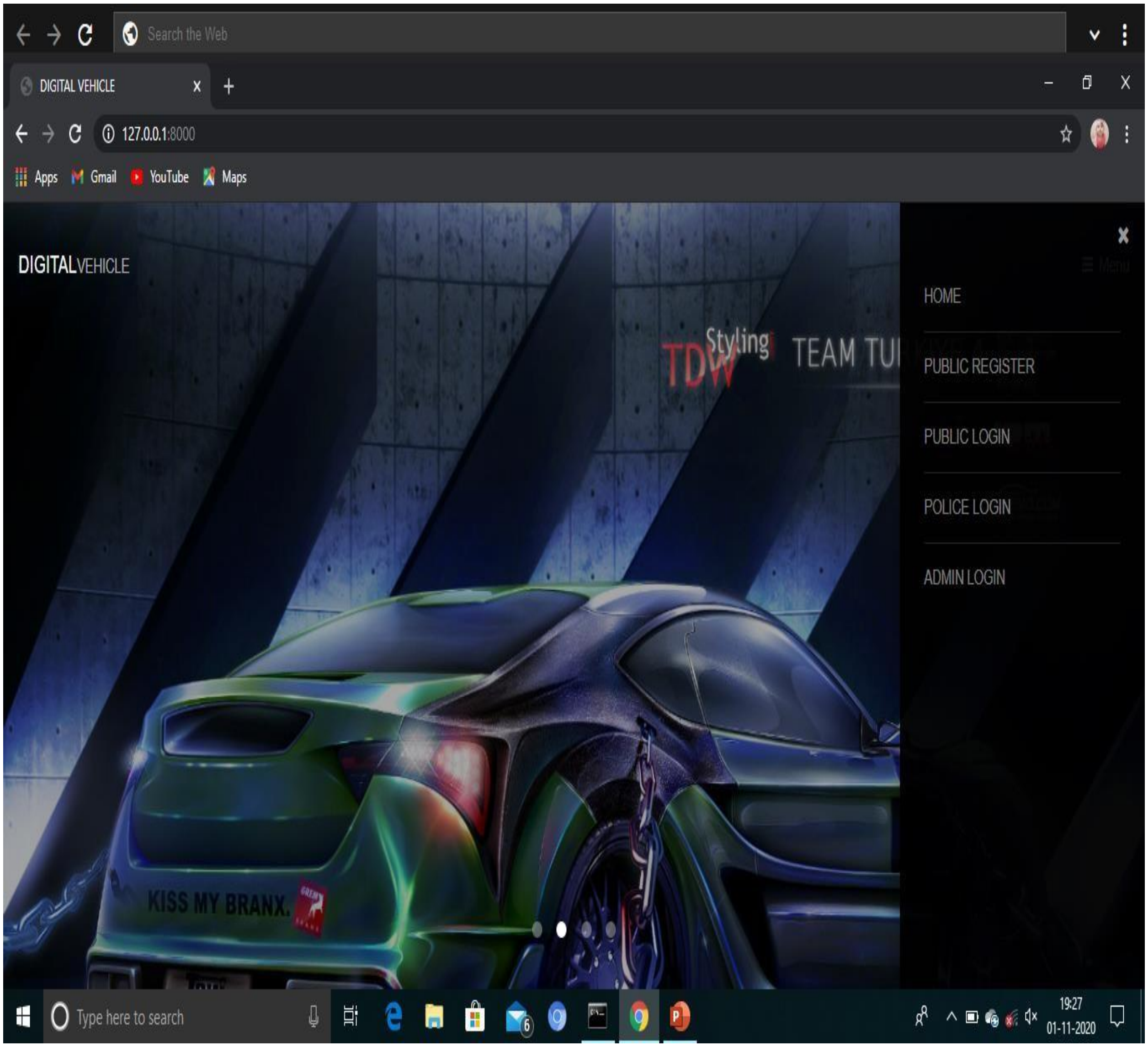
```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ELCOT>cd Desktop

C:\Users\ELCOT\Desktop>cd DigitalVehicle

C:\Users\ELCOT\Desktop\DigitalVehicle>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 10, 2020 - 06:37:43
Django version 3.0.8, using settings 'DigitalVehicle.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

SCREEN DESIGN



Screen design of an application

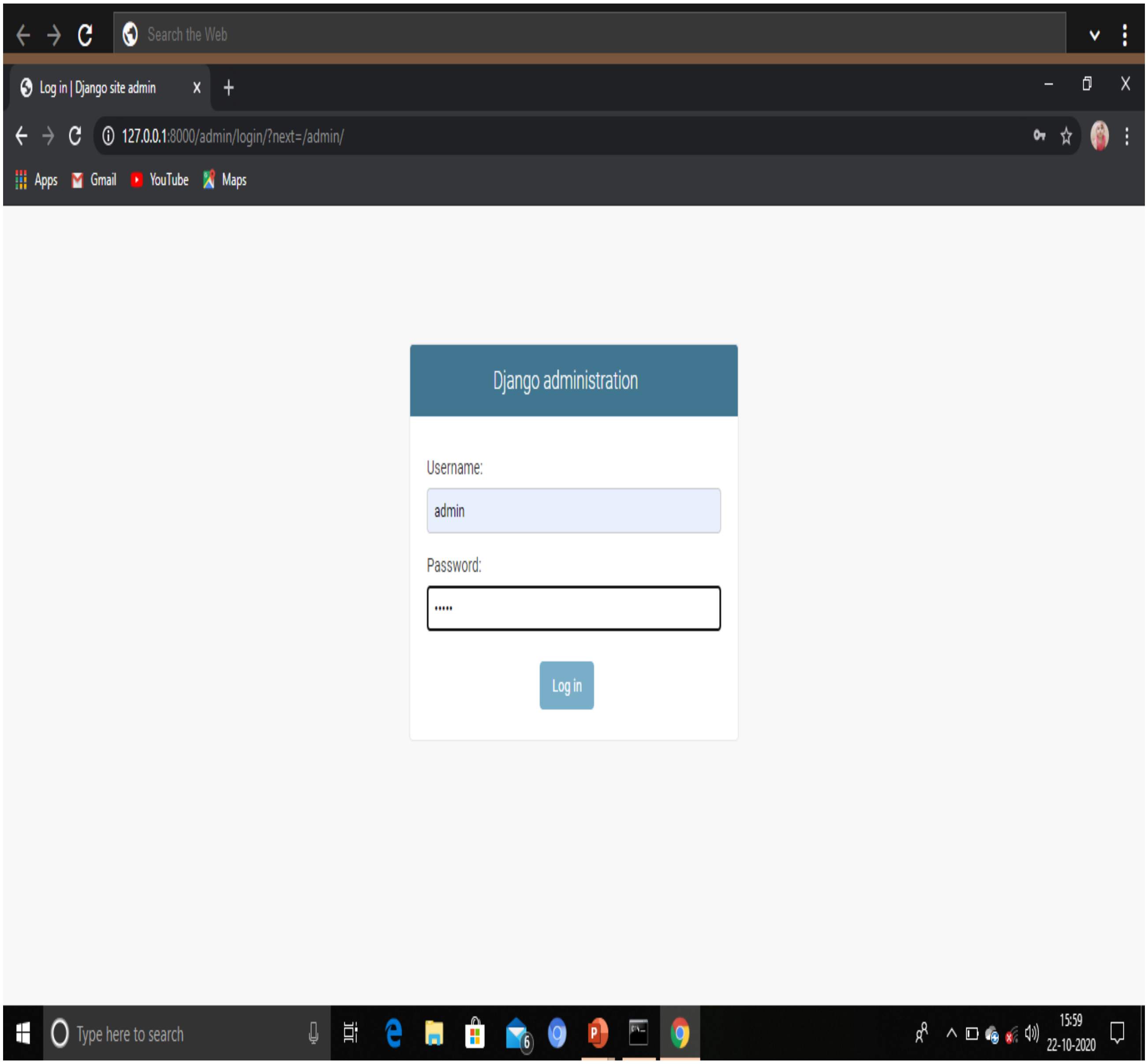
MENU OPTIONS



Menu Options

ADMIN MODULE

LOGIN PAGE:



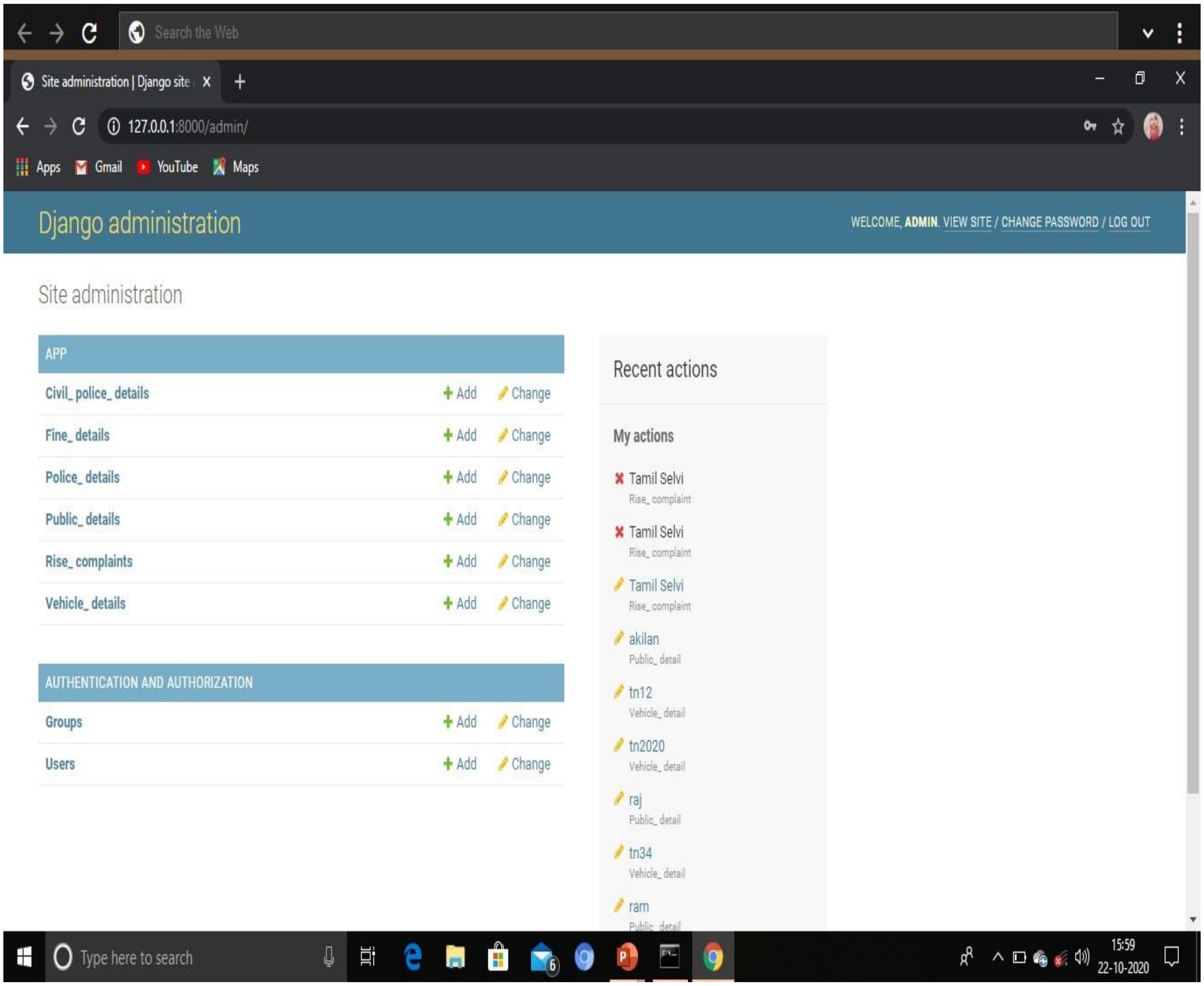
Django administration

Username: admin

Password: •••••

Login page for Admin Module

ADMIN MODULE:



Django administration

Site administration

APP		
Civil_ police_ details	+ Add	Change
Fine_ details	+ Add	Change
Police_ details	+ Add	Change
Public_ details	+ Add	Change
Rise_ complaints	+ Add	Change
Vehicle_ details	+ Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

Recent actions

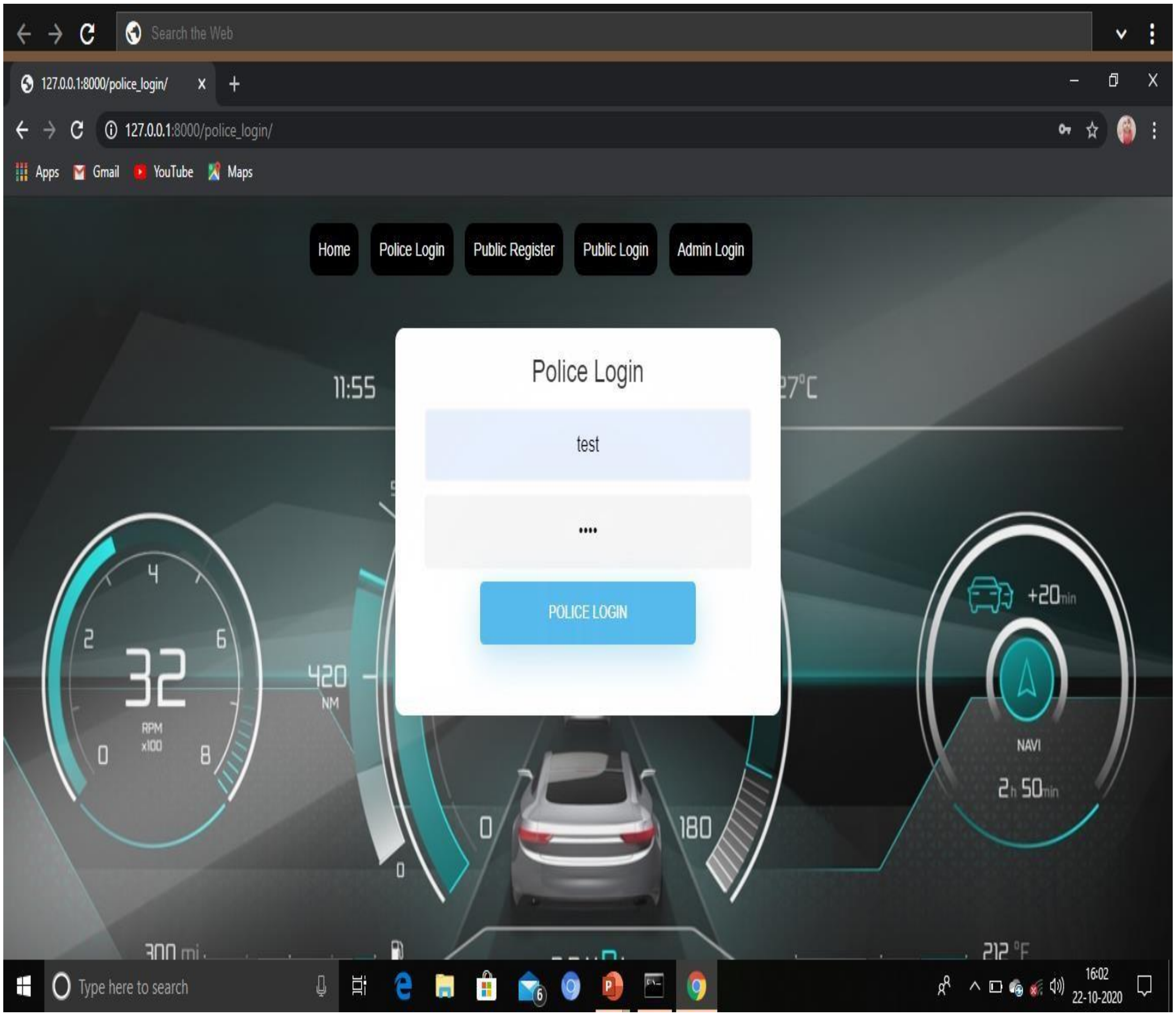
My actions

- ✖ Tamil Selvi
  Rise_ complaint
- ✖ Tamil Selvi
  Rise_ complaint
- Tamil Selvi
  Rise_ complaint
- akilan
  Public_ detail
- tn12
  Vehicle_ detail
- tn2020
  Vehicle_ detail
- raj
  Public_ detail
- tn34
  Vehicle_ detail
- ram
  Public_ detail

After admin login, every tables of the application are displayed

POLICE MODULE

LOGIN PAGE:



Police Login

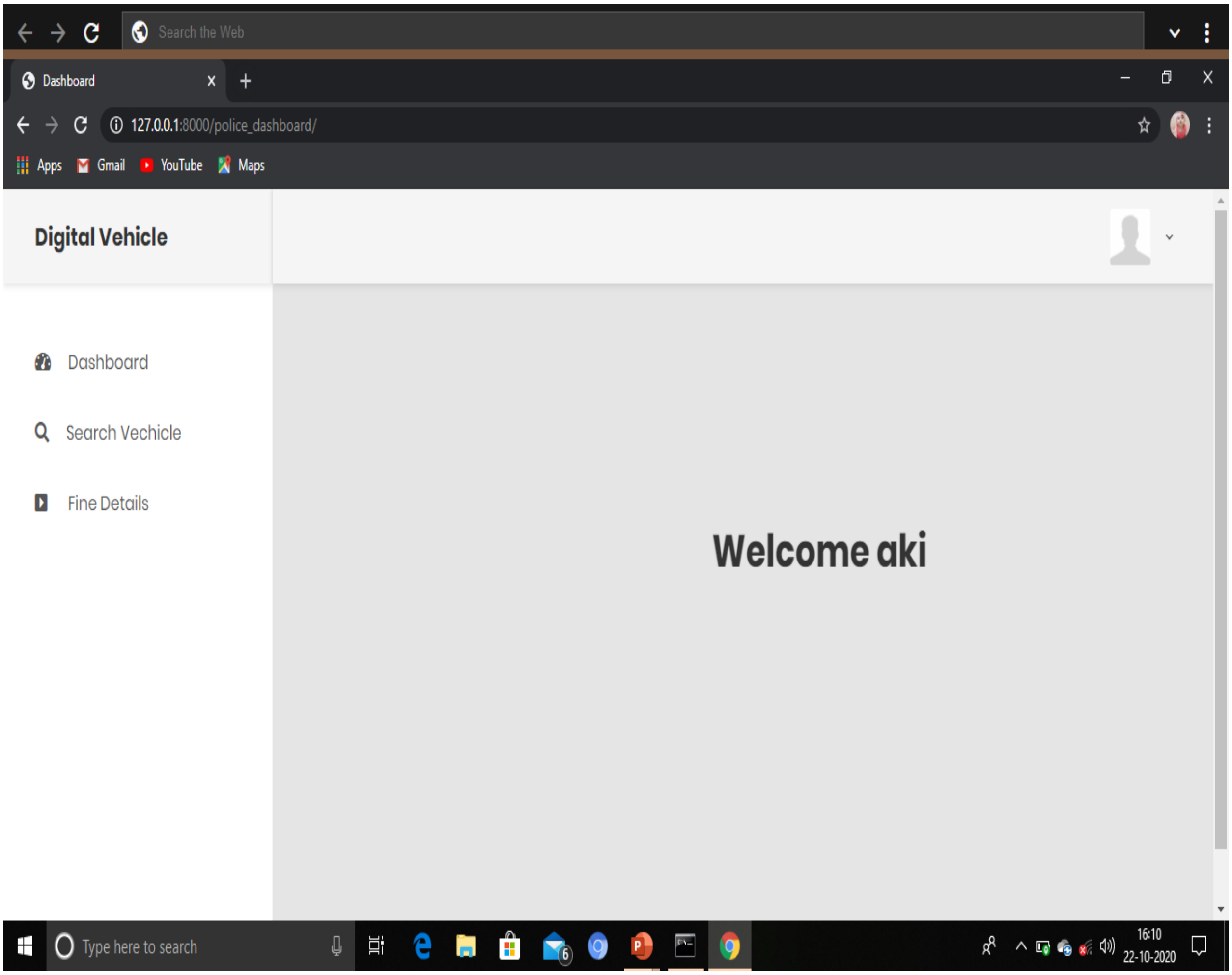
test

••••

POLICE LOGIN

Login page for Police Module

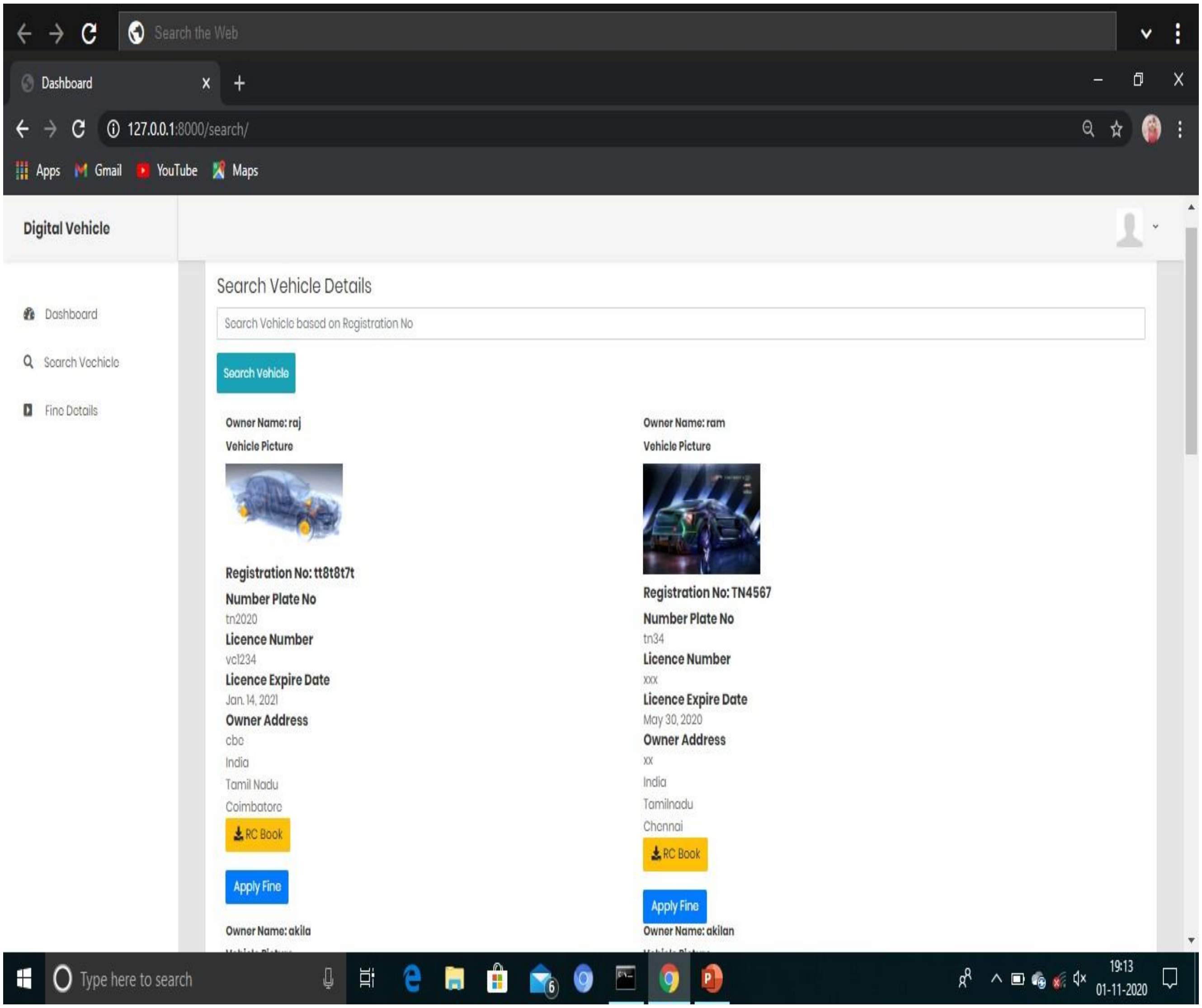
DASHBOARD



Welcome aki

Dashboard of Police Module

SEARCH VEHICLE



Owner Name: raj

Vehicle Picture

Registration No: tt8t8t7t

Number Plate No

tn2020

Licence Number

vc1234

Licence Expire Date

Jan. 14, 2021

Owner Address

cbe

India

Tamil Nadu

Coimbatore

Owner Name: ram

Vehicle Picture

Registration No: TN4567

Number Plate No

tn34

Licence Number

xxx

Licence Expire Date

May 30, 2020

Owner Address

xx

India

Tamilnadu

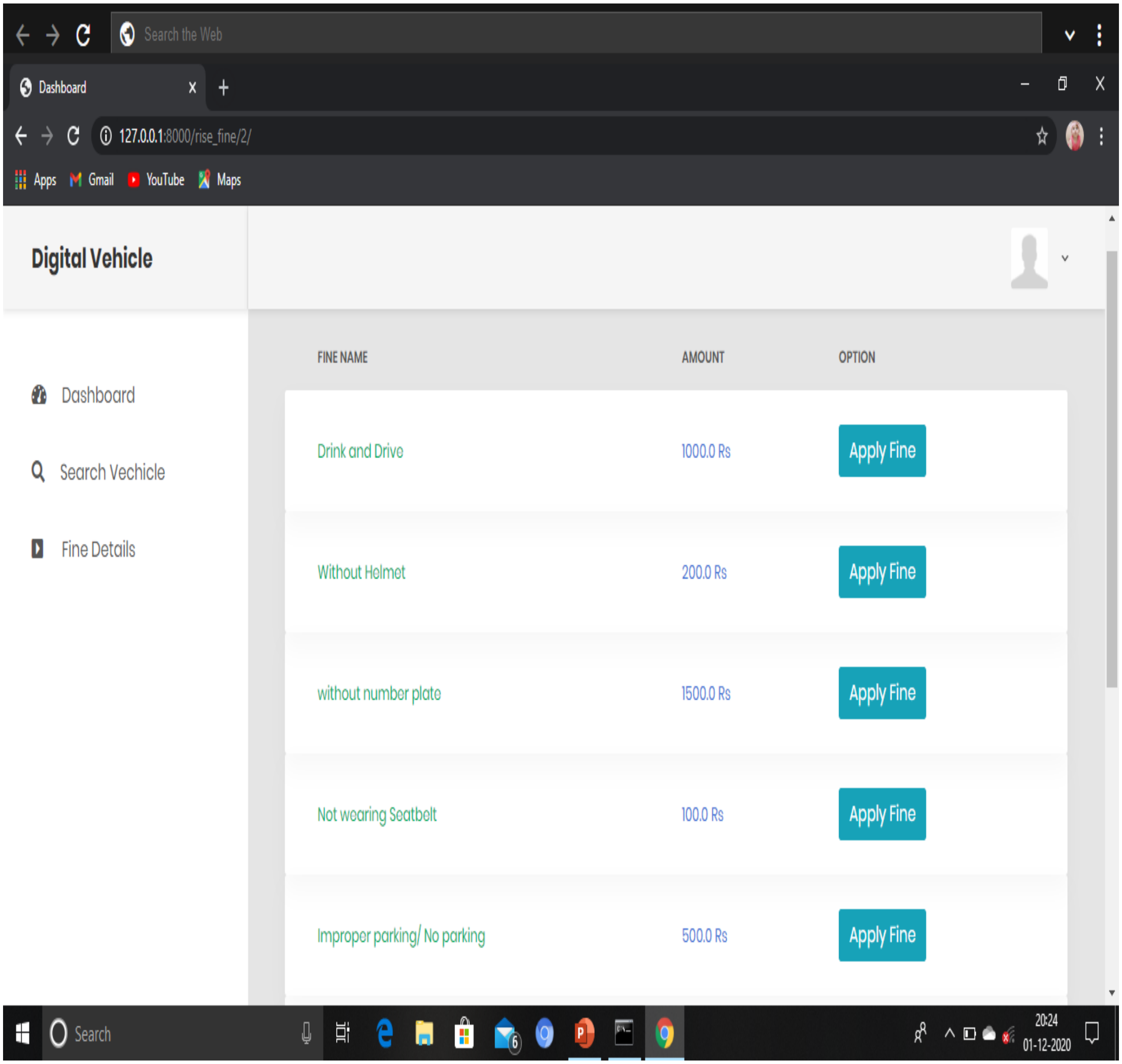
Chennai

Owner Name: akila

Owner Name: akilan

Search vehicle is used to search the vehicle details of the user, based on vehicle registration number.

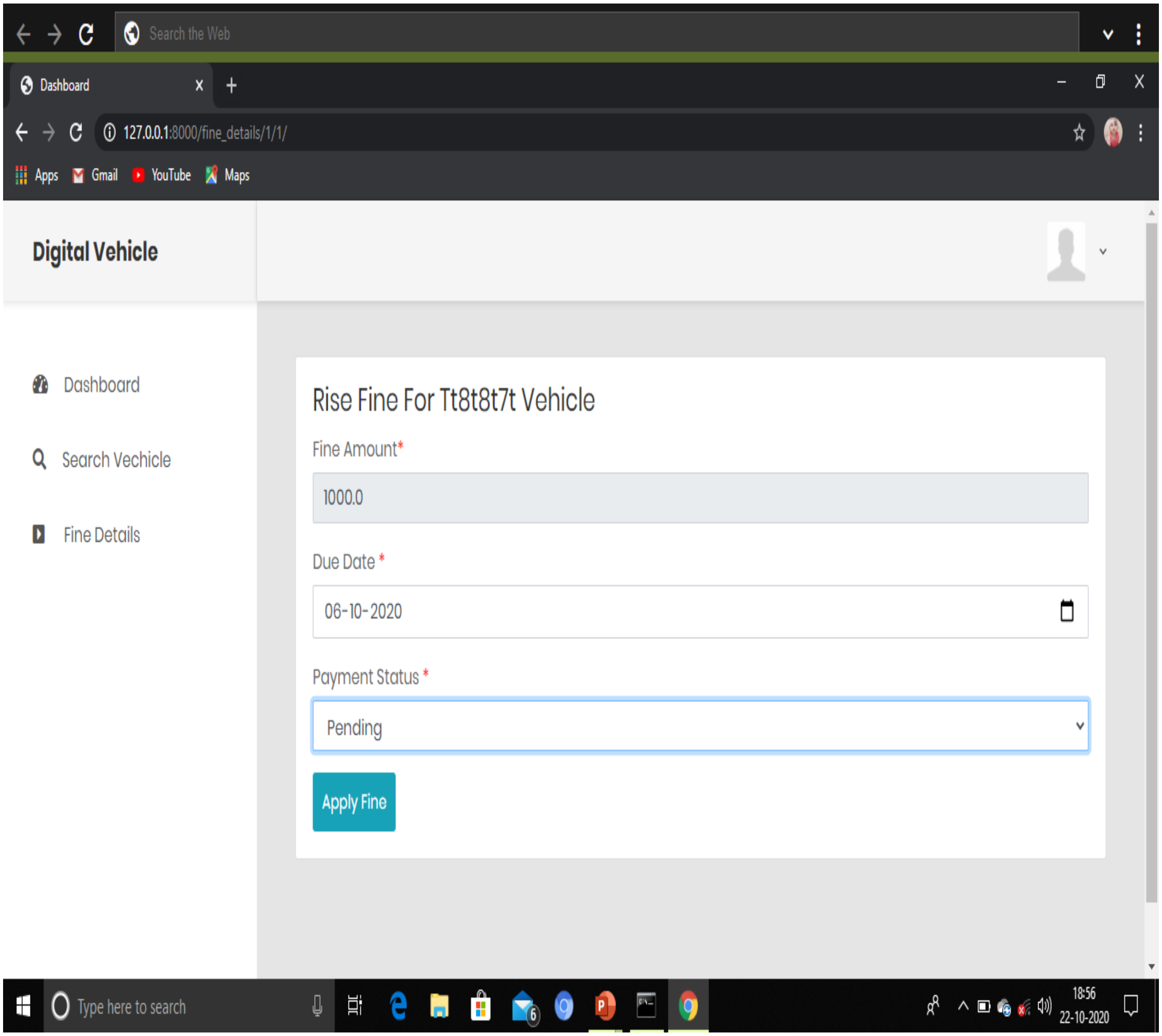
APPLY FINE



FINE NAME	AMOUNT	OPTION
Drink and Drive	1000.0 Rs	Apply Fine
Without Helmet	200.0 Rs	Apply Fine
without number plate	1500.0 Rs	Apply Fine
Not wearing Seatbelt	100.0 Rs	Apply Fine
Improper parking/ No parking	500.0 Rs	Apply Fine

Fine Details process consist of list of fine details and their amounts

PAYMENT STATUS



Rise Fine For Tt8t8t7t Vehicle

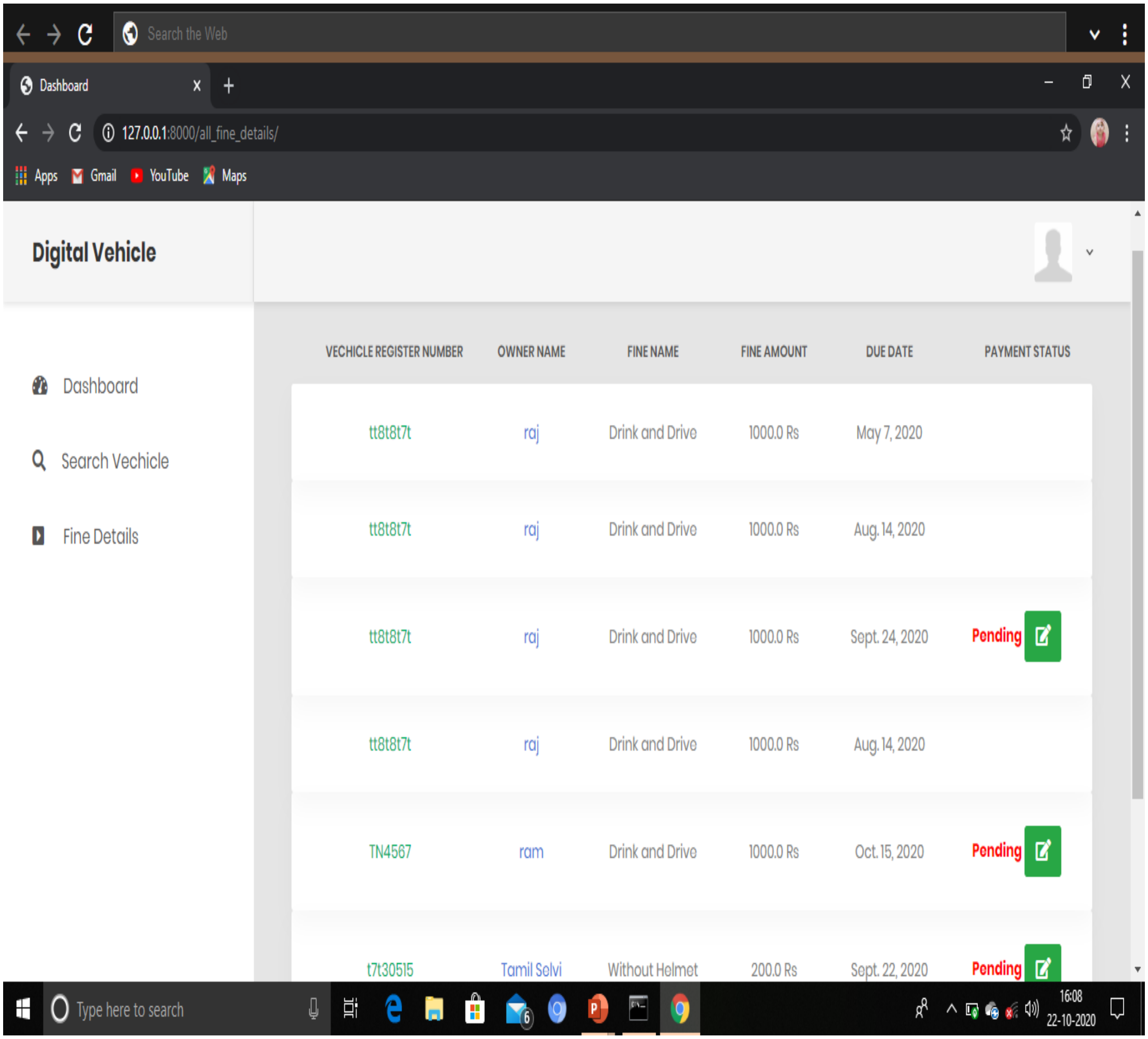
Fine Amount*
1000.0

Due Date *
06-10-2020

Payment Status *
Pending

Payment status of the Fine applied on public

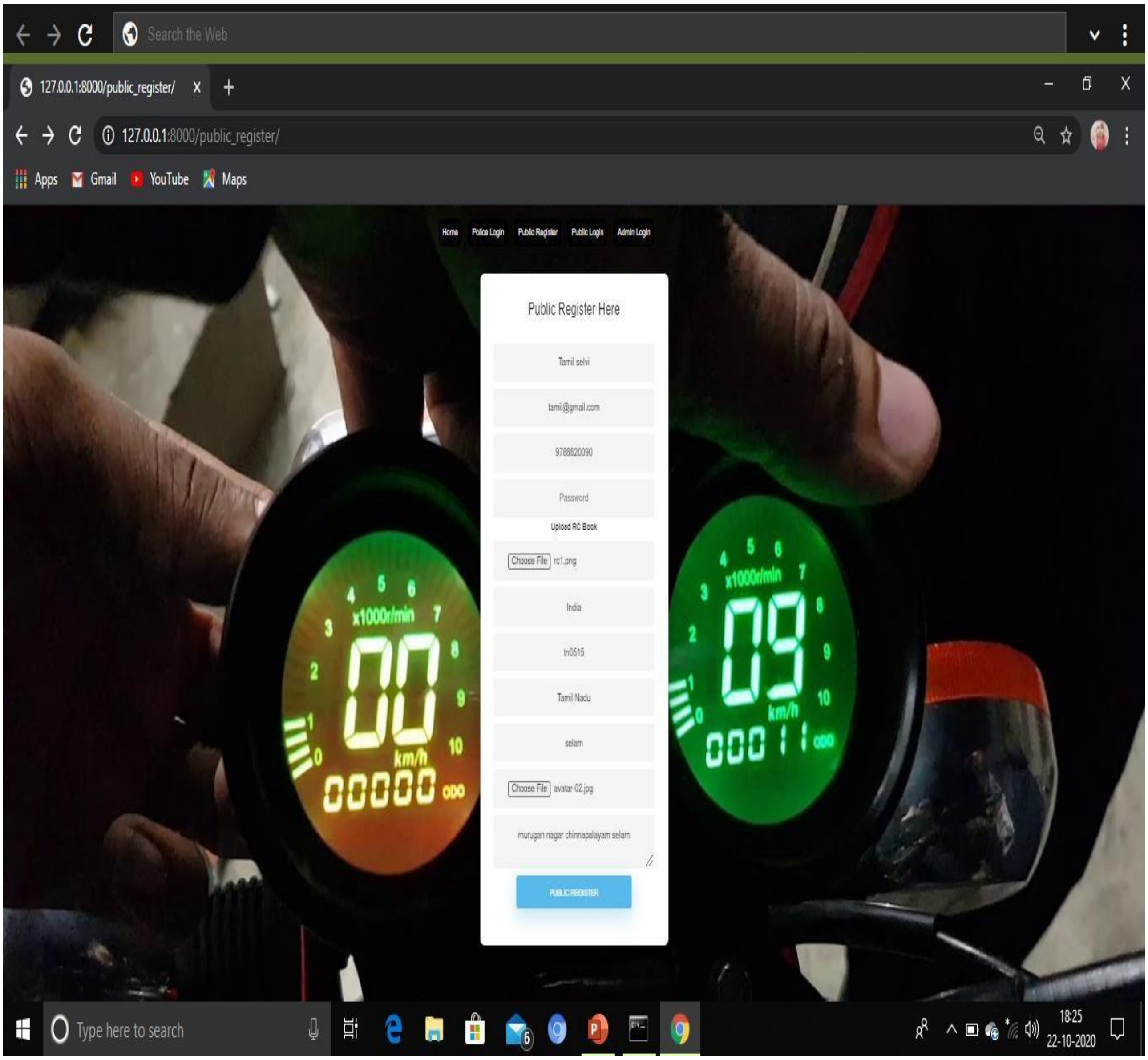
FINE DETAILS



VECHICLE REGISTER NUMBER	OWNER NAME	FINE NAME	FINE AMOUNT	DUE DATE	PAYMENT STATUS
tt8t8t7t	raj	Drink and Drive	1000.0 Rs	May 7, 2020	
tt8t8t7t	raj	Drink and Drive	1000.0 Rs	Aug. 14, 2020	
tt8t8t7t	raj	Drink and Drive	1000.0 Rs	Sept. 24, 2020	Pending
tt8t8t7t	raj	Drink and Drive	1000.0 Rs	Aug. 14, 2020	
TN4567	ram	Drink and Drive	1000.0 Rs	Oct. 15, 2020	Pending
t7t30515	Tamil Selvi	Without Helmet	200.0 Rs	Sept. 22, 2020	Pending

List of Fine details, applied on people

PUBLIC REGISTRATION FORM



Public Register Here

Tamil selvi

tamil@gmail.com

9788820090

Password

Upload RC Book

Choose File rc1.png

India

tn0515

Tamil Nadu

selam

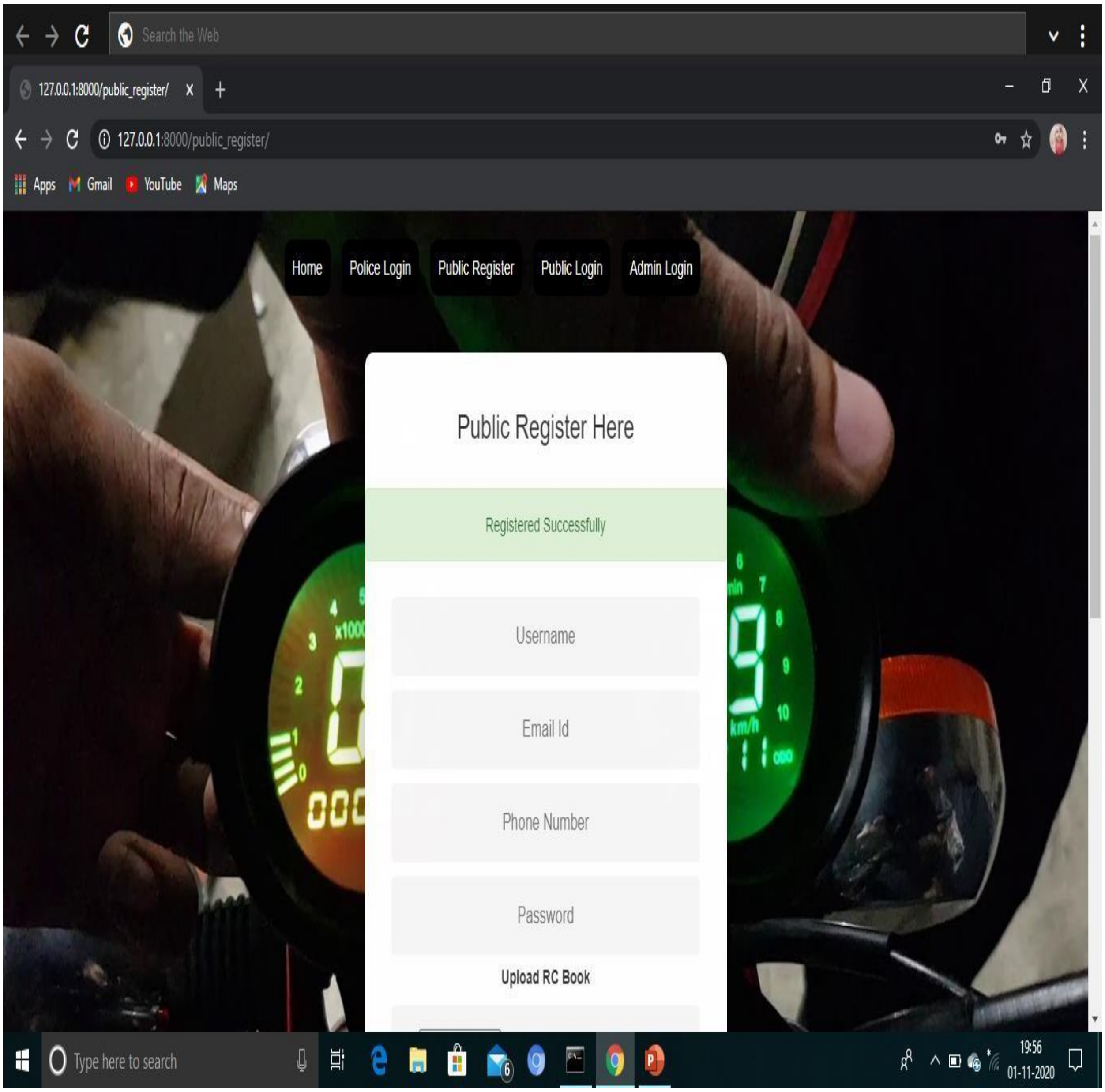
Choose File avatar-02.jpg

murugan nagar chinnapalayam selam

PUBLIC REGISTER

Public registration form consists of public details

AFTER REGISTRATION



Public Register Here

Registered Successfully

Username

Email Id

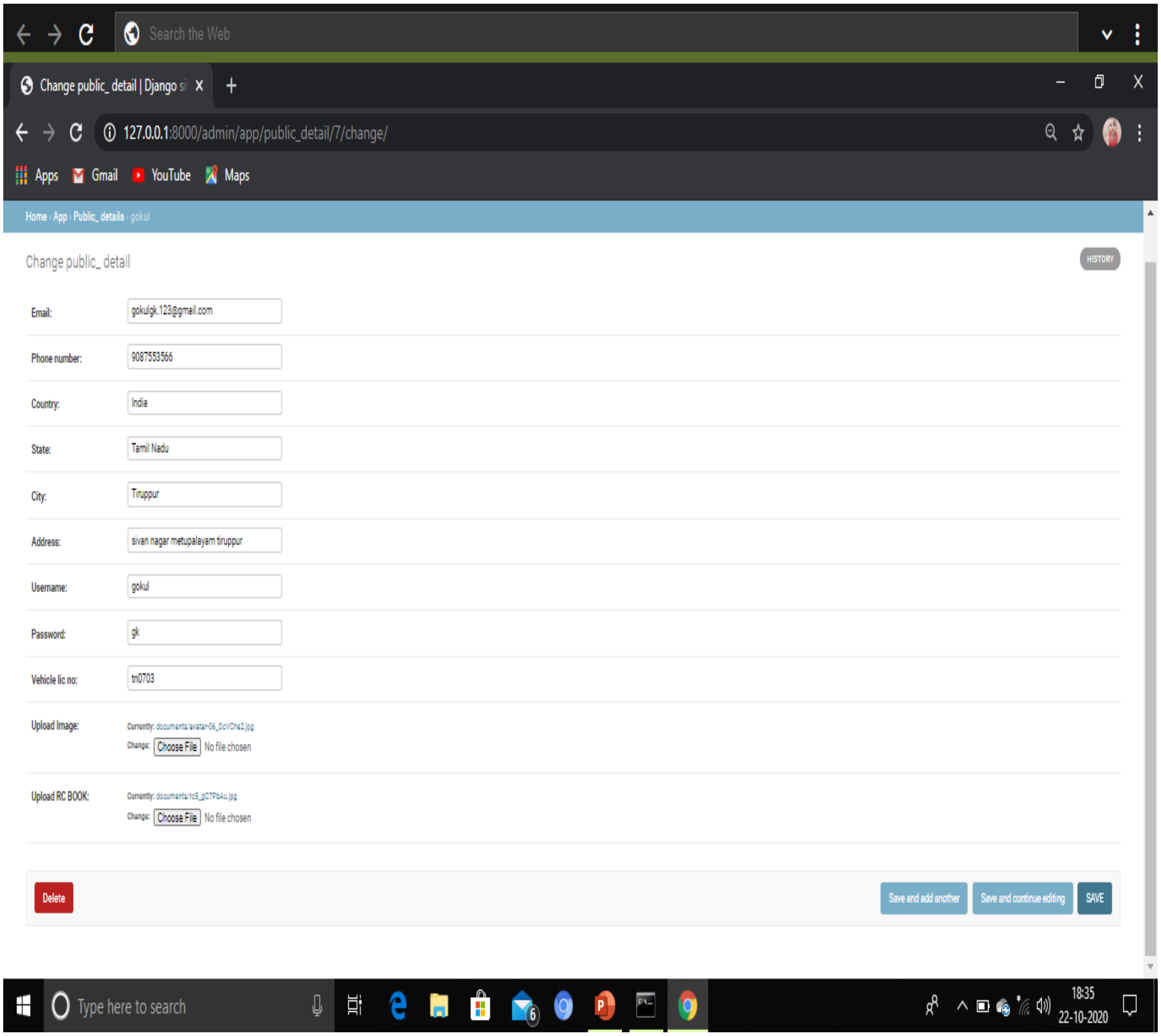
Phone Number

Password

Upload RC Book

After registration, the message “Registered successfully” will be displayed

AFTER REGISTRATION THE DETAILS CAN BE VIWED FROM THE ADMIN PANEL

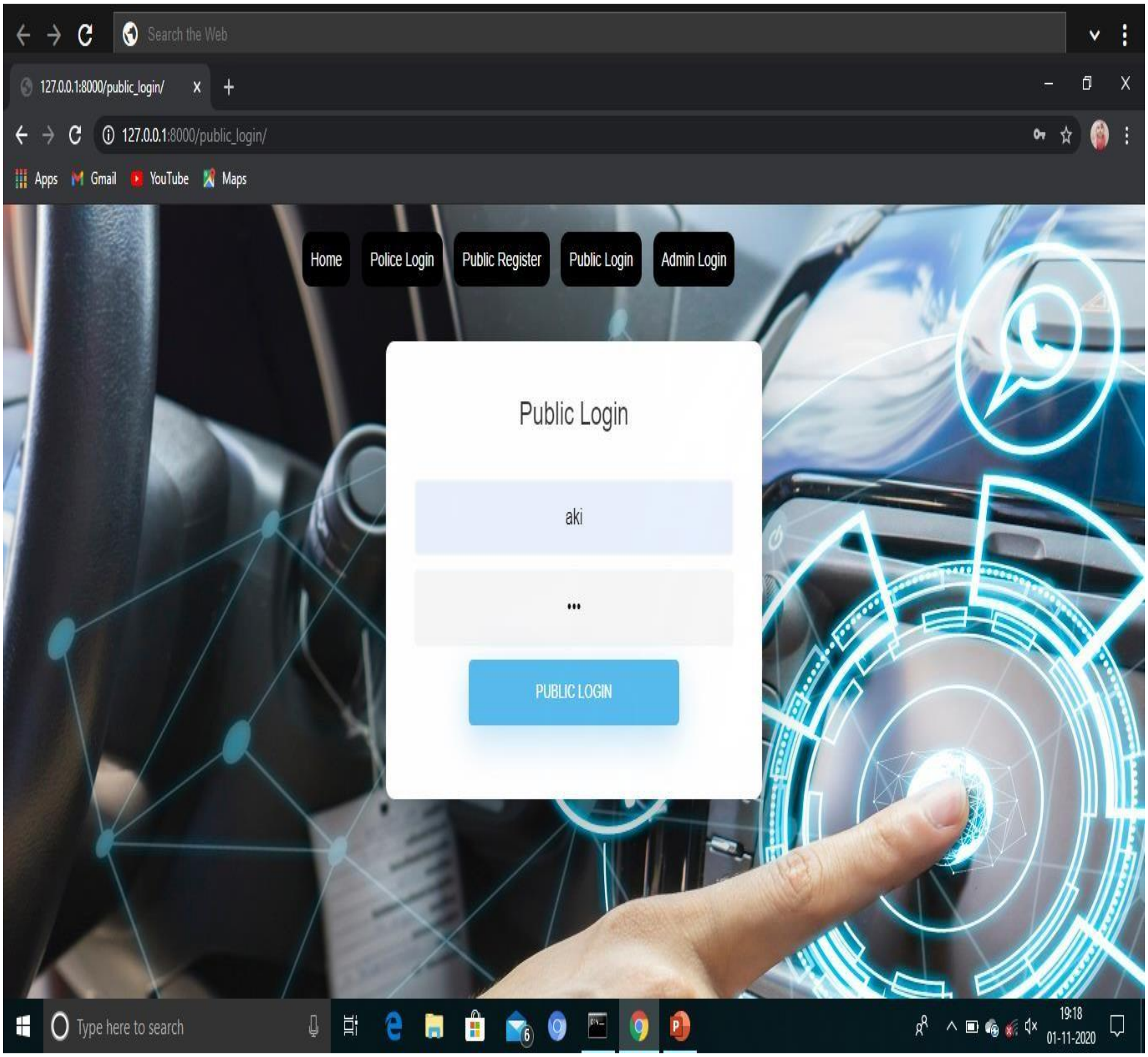

Search the Web
Change public_ detail | Django si
127.0.0.1:8000/admin/app/public_detail/7/change/
Apps
Gmail
YouTube
Maps
Home › App › Public_ details › gokul
HISTORY
Delete
Save and add another
Save and continue editing
SAVE
Type here to search
18:35
22-10-2020

Change public_ detail

Email:	gokulgk.123@gmail.com
Phone number:	9087553566
Country:	India
State:	Tamil Nadu
City:	Tiruppur
Address:	sivan nagar metupalayam tiruppur
Username:	gokul
Password:	gk
Vehicle lic no:	tn0703
Upload Image:	Currently: documents/avatar-06_OcVChe2.jpg Change: Choose File No file chosen
Upload RC BOOK:	Currently: documents/rc5_gC7PbAu.jpg Change: Choose File No file chosen

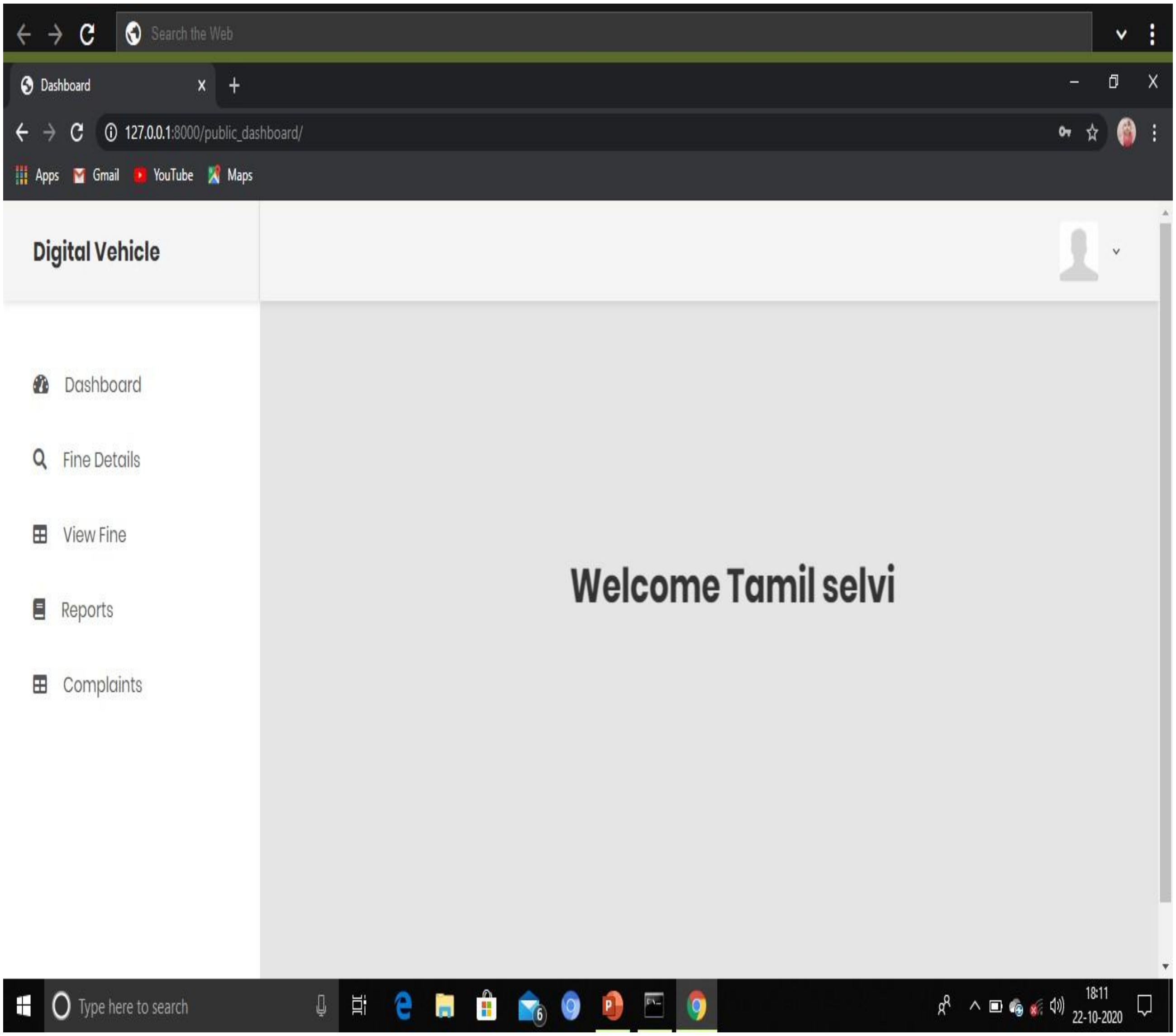
After registration the details can be viewed from the admin panel

PUBLIC LOGIN PAGE



Public login page

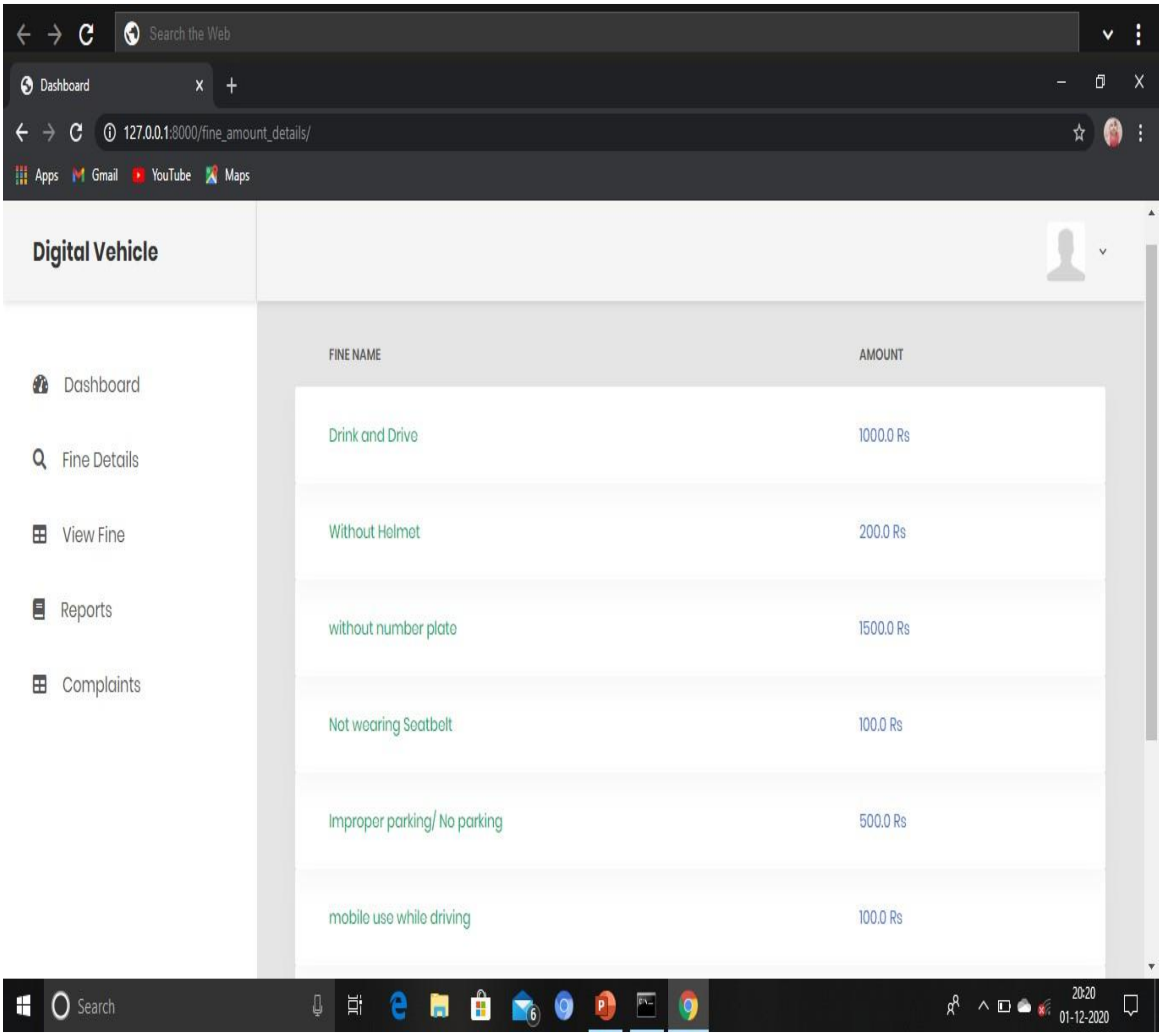
PUBLIC DASHBOARD



Welcome Tamil selvi

Dashboard of Public Module

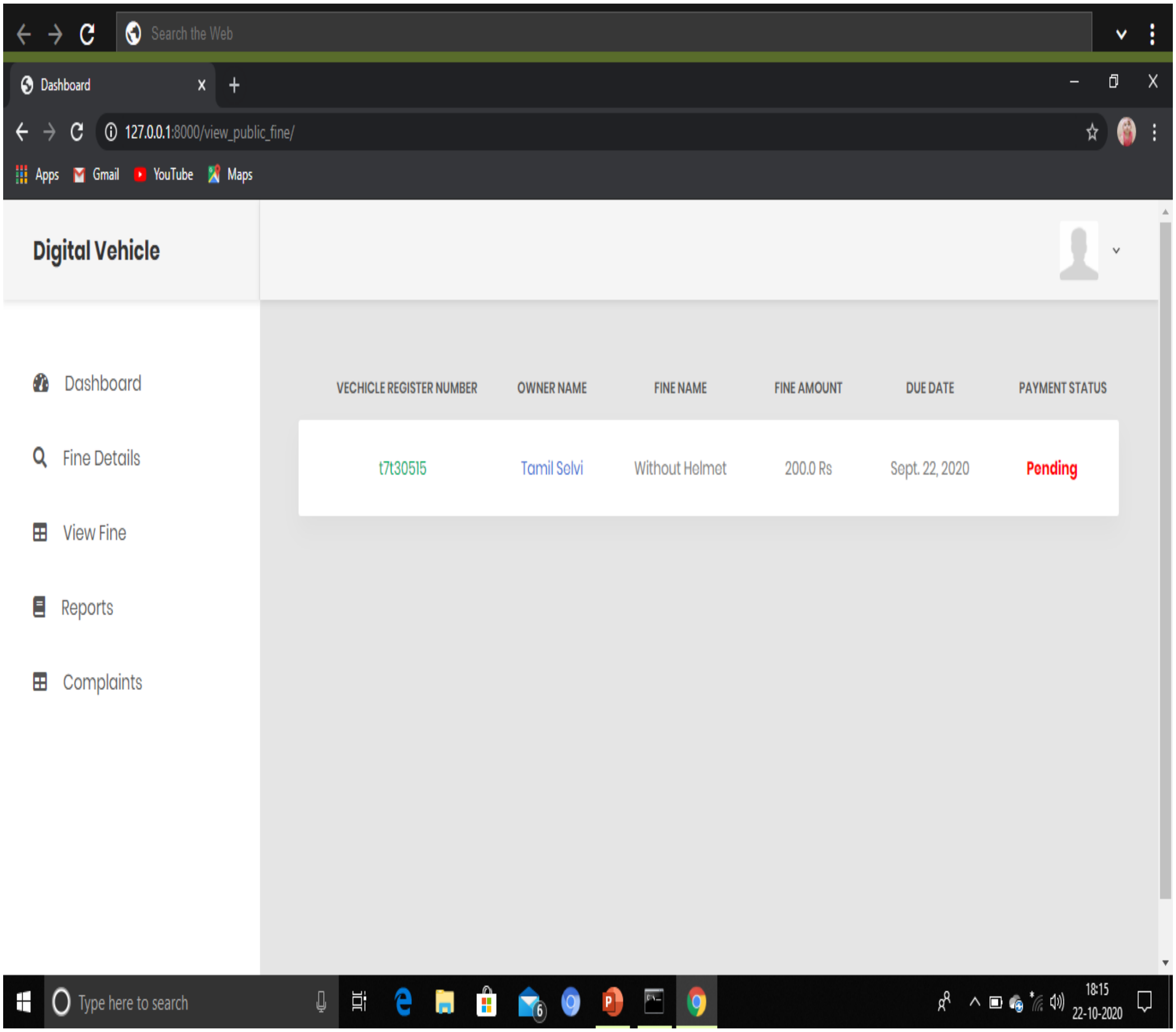
FINE DETAILS



FINE NAME	AMOUNT
Drink and Drive	1000.0 Rs
Without Helmet	200.0 Rs
without number plate	1500.0 Rs
Not wearing Seatbelt	100.0 Rs
Improper parking/ No parking	500.0 Rs
mobile use while driving	100.0 Rs

Fine Details process consist of list of fine details and their amounts

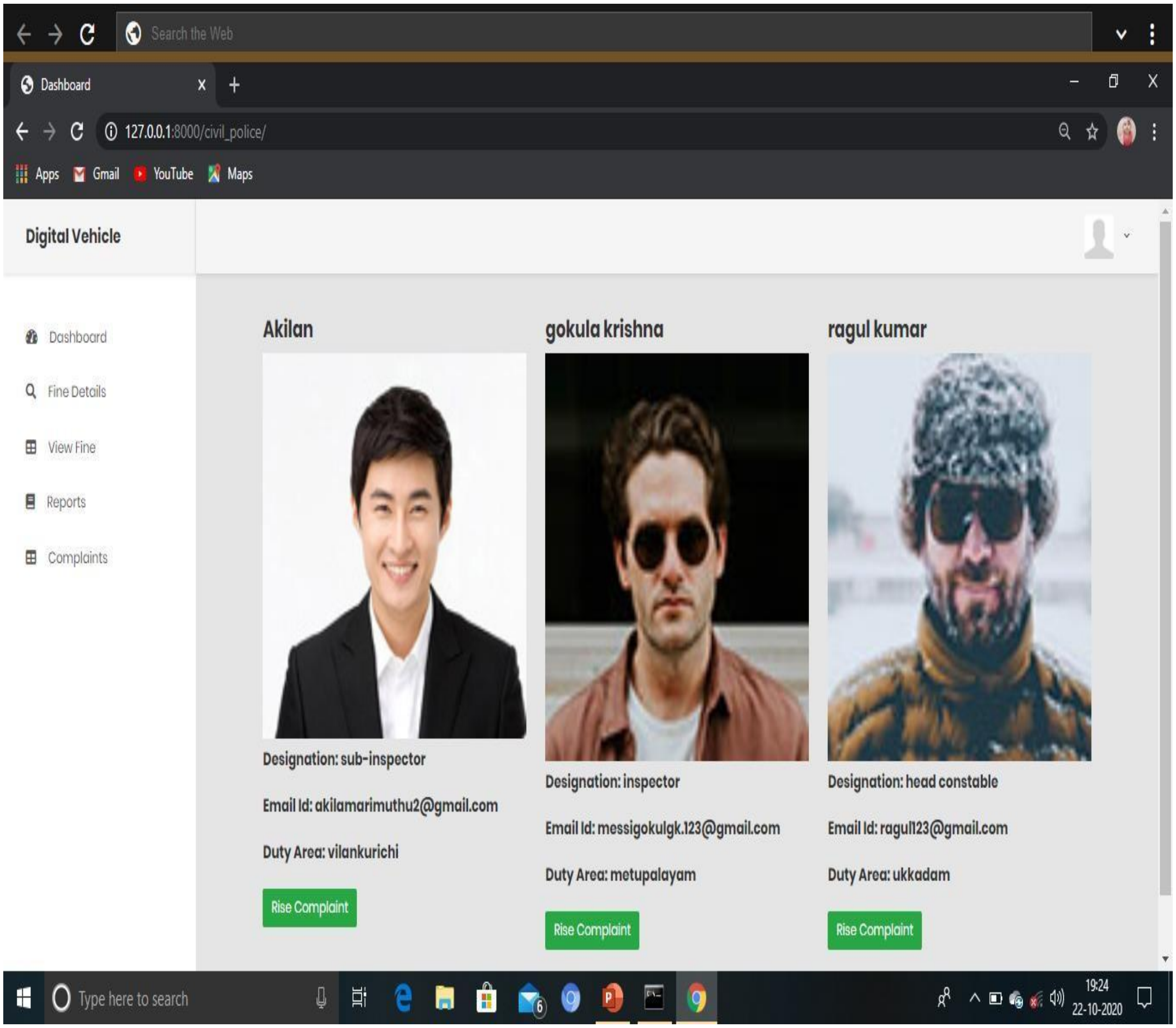
VIEW FINE



VECHICLE REGISTER NUMBER	OWNER NAME	FINE NAME	FINE AMOUNT	DUE DATE	PAYMENT STATUS
t7t30515	Tamil Selvi	Without Helmet	200.0 Rs	Sept. 22, 2020	Pending

View Fine shows the fine amount applied on vehicle of the respective public

REPORT – USED TO REPORT A COMPLAINT ON STOLEN VEHICLE



Akilan

Designation: sub-inspector

Email Id: akilamarimuthu2@gmail.com

Duty Area: vilankurichi

gokula krishna

Designation: inspector

Email Id: messigokulgk.123@gmail.com

Duty Area: metupalayam

ragul kumar

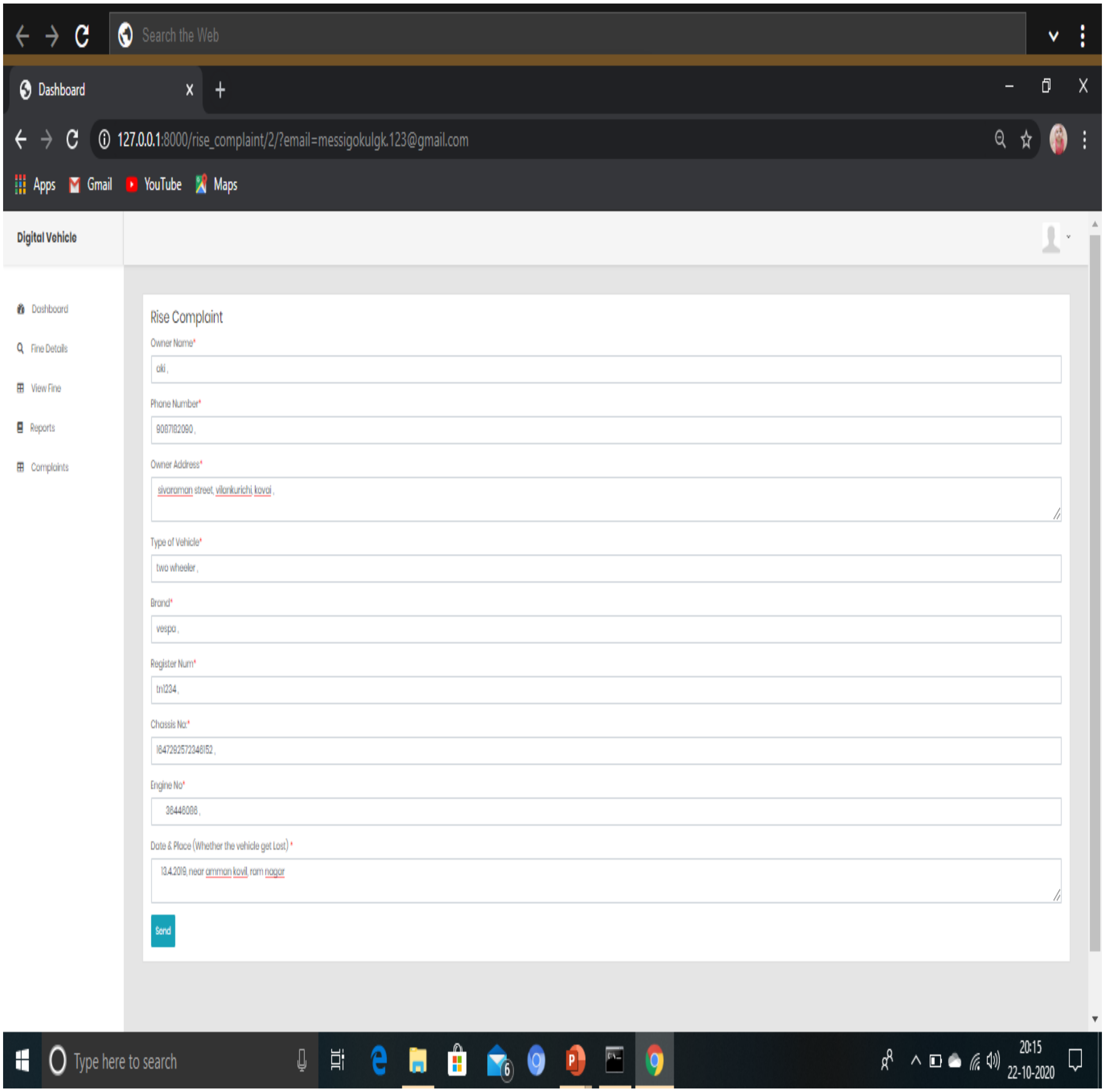
Designation: head constable

Email Id: ragul123@gmail.com

Duty Area: ukkadam

Report – used to report a complaint on stolen vehicle

RISE COMPLAINT USING THE COMPLAINT FORM



Rise Complaint

Owner Name*

oki ,

Phone Number*

9087182090 ,

Owner Address*

sivaraman street, vilankurichi, kovai ,

Type of Vehicle*

two wheeler ,

Brand*

vespa ,

Register Num*

tn1234 ,

Chassis No:*

1647292572346152 ,

Engine No*

38446088 ,

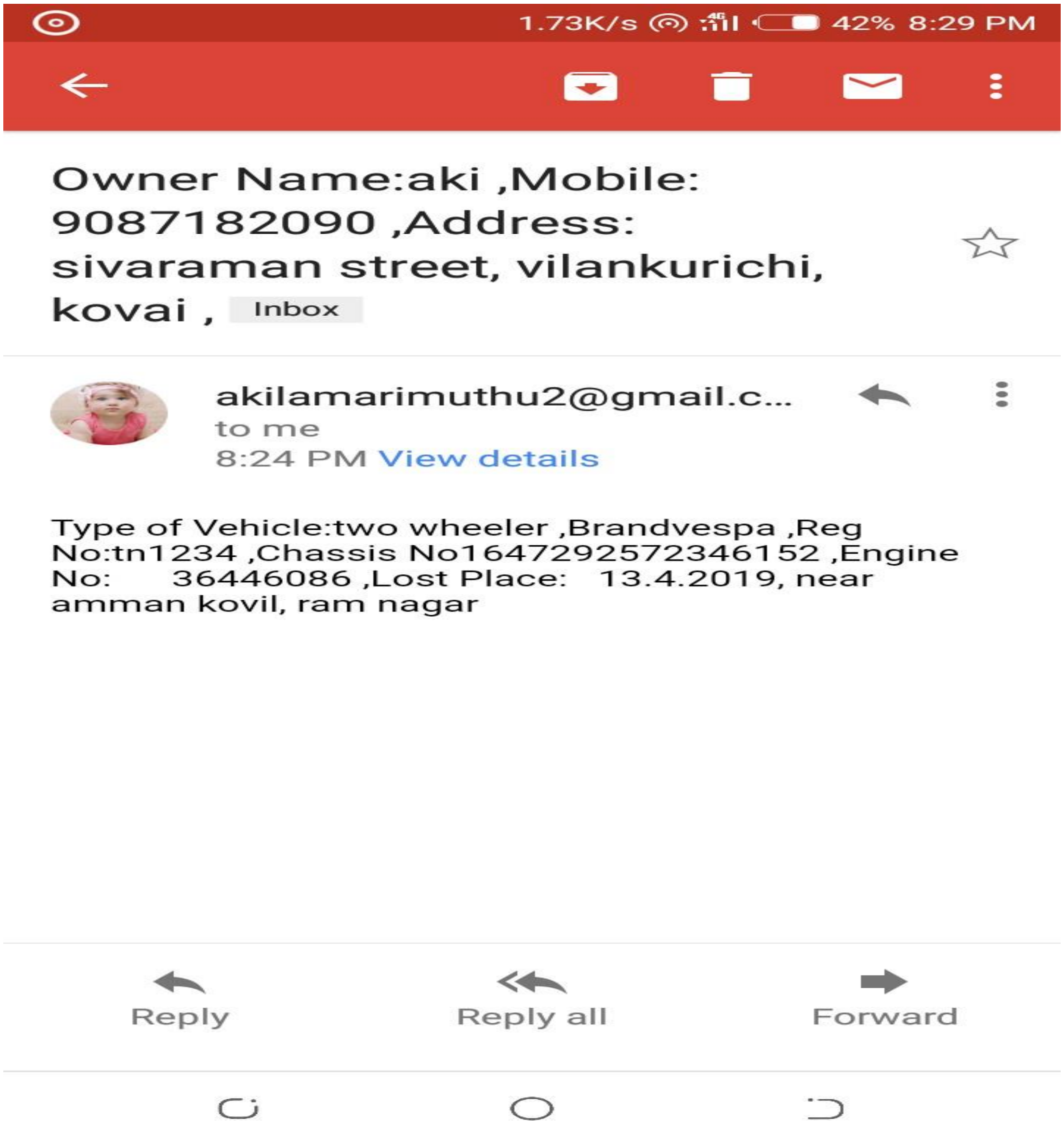
Date & Place (Whether the vehicle get Lost) *

13.4.2019, near amman kovil, ram nagar

Send

Rise complaint is used to apply complaint on stolen vehicle if there is any issue occurs

EMAIL NOTIFICATION FROM PUBLIC TO CRIME POLICE



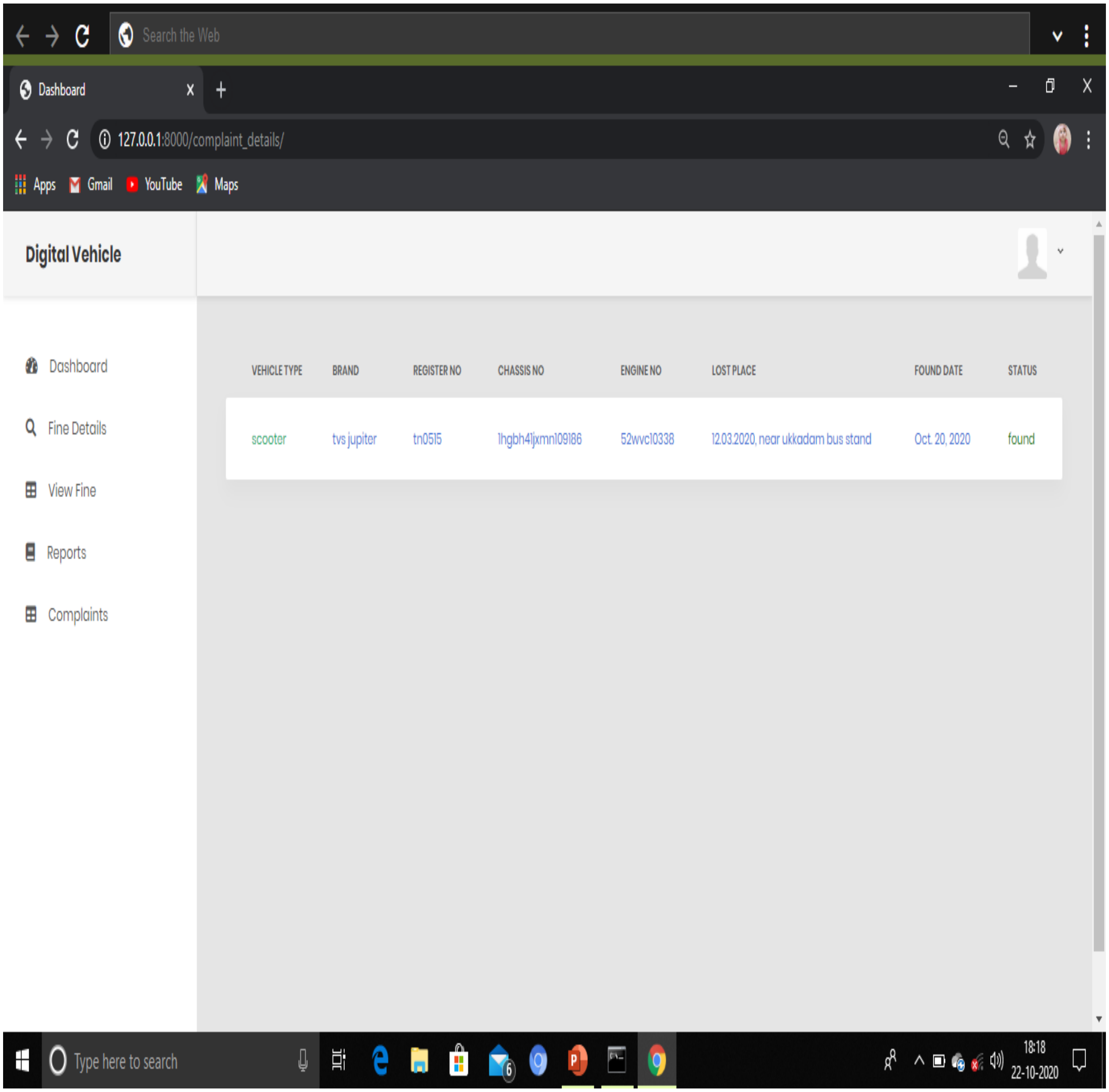
Owner Name:aki ,Mobile: 9087182090 ,Address: sivaraman street, vilankurichi, kovai ,

akilamarimuthu2@gmail.c...
to me
8:24 PM View details

Type of Vehicle:two wheeler ,Brandvespa ,Reg No:tn1234 ,Chassis No1647292572346152 ,Engine No: 36446086 ,Lost Place: 13.4.2019, near amman kovil, ram nagar

Email notification send to crime police by the public about stolen vehicle details

VIEW COMPLAINT – TO CHECK THE STATUS OF THE COMPLAINT



VEHICLE TYPE	BRAND	REGISTER NO	CHASSIS NO	ENGINE NO	LOST PLACE	FOUND DATE	STATUS
scooter	tvs jupiter	tn0515	1hgbh41jxmn109186	52wvc10338	12.03.2020, near ukkadam bus stand	Oct. 20, 2020	found

View complaint – to check the status of the complaint

B. SAMPLE CODING

1. DATABASE CONNECTIVITY

```
# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

2. URLS.PY

```
"""DigitalVehicle URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.2/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,include
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('app.urls')),
]
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

3. MANAGE.PY

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'DigitalVehicle.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

```
if__name__== '_main_':
    main()
```

4. VIEW_PUBLIC_FINE

```
{% include 'public_header.html' %}
<style type="text/css">
    tr,td
    {
        text-align: center !important;
    }
</style>
<div class="table-responsive table-responsive-data2">
    <table class="table table-data2">
        <thead>
            <tr>
                <th>Vechicle Register Number</th>
                <th>Owner Name</th>
                <th>Fine Name</th>
                <th>Fine Amount</th>
                <th>Due Date</th>
                <th>Payment Status</th>
            </tr>
        </thead>
        <tbody>
        {% for i in ids %}
            <tr class="tr-shadow">
                <td>
                    <span class= " status--
process"> {{ i.vehicle_id.vehicle_register_no }} </span> <br></td>
```

```
                <td class="desc">{{ i.vehicle_id.owner_name }} </td>
                <td>{{ i.fine_id.fine_name }}</td>
                <td>{{ i.fine_id.amount }} Rs</td>
                <td>{{ i.due_date }}</td>
                <td>{% if i.status == 'pending' %} <span style="color: red;font-weight: bold;">Pending</span>
                {% elif i.status == 'paid' %} <span style="color: green;font-weight: bold;">Paid</span>
               {% endif %}</td>
            </tr>
            {% empty %}
            <tr class=" alert alert-danger "> No Fine Details </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
{% include 'footer.html' %}
```

5. RISE_COMPLAINT

```
{% include 'public_header.html' %}
<div class="card">
    <div class="card-body">
        <div class="card-title">
            <h3 class=" title-2">Rise Complaint</h3>
        </div>
               {% if messages %}
            {% for i in messages %}
                <div class="alert alert-success">{{i}}</div>
            {% endfor %}
        {% endif %}
    <form action="" method="post" enctype="multipart/form-data"> {% csrf_token %}
         <div class="form-group ">
        <label class="control-label mb-
1">Owner Name<span style="color: red;">*</span></label>
        <input name="owner_name" type="text" class="form-control " required="required">
        </div>
      <div class="form-group">
        <label class="control-label mb-
1">Phone Number<span style="color: red;">*</span></label>
        <input type="text" name="phone_number" class="form-
control " required="required" >
        </div>
      <div class="form-group">
        <label class="control-label mb-
1">Owner Address<span style="color: red;">*</span></label>
        <textarea name="address" class="form-control " required="required" >  </textarea>
              </div>
      <div class="form-group">
        <label class="control-label mb-
1">Type of Vehicle<span style="color: red;">*</span></label>
        <input type="text" name="type_of_vehicle" class="form-
control " required="required" >
      </div>
      <div class="form-group">
        <label class="control-label mb-1">Brand<span style="color: red;">*</span></label>
        <input type="text" name="brand" class="form-control " required="required" >
              </div>
      <div class="form-group">
        <label class="control-label mb-
1">Register Num<span style="color: red;">*</span></label>
        <input type="text" name="reg_no" class="form-control " required="required" >
      </div>
```

```
<div class="form-group">
                <label class="control-label mb-
1">Chassis No:<span style="color: red;">*</span></label>
                <input type="text" name="chassis_no" class="form-control " required="required" >
            </div>
            <div class="form-group">
                <label class="control-label mb-
1">Engine No<span style="color: red;">*</span></label>
                <input type="text" name="engine_no" class="form-control " required="required" >
            </div>
            <div class="form-group">
                <label class="control-label mb-
1">Date & Place (Whether the vehicle get Lost) <span style="color: red;">*</span></label>
                <textarea name="place" class="form-control " required="required" >  </textarea>
            </div>
            <div>
               <input type="submit" value="Send" class="btn btn-info " style="padding: 10px;font-
size: 15px;">
            </div>
        </form>
    </div>
</div>
{% include 'footer.html'%}
```

6. RISE_FINE

```
{% include 'scientific_header.html' %}


<div class="table-responsive table-responsive-data2">
    <table class="table table-data2">
        <thead>
            <tr>
                <th>Fine Name</th>
                <th>Amount</th>
                <th>Option</th>
            </tr>
        </thead>
        <tbody>
        {% for form in ids %}
            <tr class="tr-shadow">
                <td>
                    <span class= " status--process"> {{ form.fine_name }} </span> <br>
                </td>
                <td class="desc">{{ form.amount }} Rs</td>
                <td><a href="{% url 'fine_details' pk=form.id v_id=vehicle_id %}" class="btn btn-
info">Apply Fine</a></td>
            </tr>
            {% empty %}
            <tr class=" alert alert-danger "> No Fine Details </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
{% include 'footer.html' %}
```

7. FINE_AMOUNT_DETAILS

```
{% include 'public_header.html' %}

<div class="table-responsive table-responsive-data2">
    <table class="table table-data2">
        <thead>
            <tr>
                <th>Fine Name</th>
                <th>Amount</th>
            </tr>
        </thead>
        <tbody>
        {% for form in ids %}
            <tr class="tr-shadow">
                <td>
                    <span class= " status--process"> {{ form.fine_name }} </span> <br>
                </td>
                <td class="desc">{{ form.amount }} Rs</td>
            </tr>
            {% empty %}
            <tr class=" alert alert-danger "> No Fine Details </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
{% include 'footer.html' %}
```

8. COMPLAINT_DETAILS

```
{% include 'public_header.html' %}
<div class="table-responsive table-responsive-data2">
    <table class="table table-data2">
        <thead>
            <tr>
                <th>Vehicle Type</th>
                <th>Brand</th>
                <th>Register No</th>
                <th>Chassis No</th>
                <th>Engine No </th>
                <th>Lost Place</th>
                <th>Found Date</th>
                <th>Status </th>
            </tr>
        </thead>
        <tbody>
        {% for i in ids %}
            <tr class="tr-shadow">
                <td>
                    <span class= " status--process"> {{ i.type_of_vehicle }} </span> <br>
                </td>
                <td class="desc">{{ i.brand }} </td>
                <td class="desc">{{ i.reg_no }} </td>
                <td class="desc">{{ i.chassis_no }} </td>
                <td class="desc">{{ i.engine_no }} </td>
                <td class="desc">{{ i.place }} </td>
                <td class="desc">{{ i.date }} </td>
                <td class="desc" style="color: green;">{{ i.status }} </td>
            </tr>
            {% empty %}
            <tr class=" alert alert-danger "> No Complaint Details </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
{% include 'footer.html' %}
```