

# Delhi Technological University



**Database Management System Project**

**“DMV MANAGEMENT  
SYSTEM”**

# **DMV Management System**

A comprehensive project report has been submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Technology**

in

**Computer Engineering**

**Done By:**

**Vijay Kumar Sawhney 2K21/CO/515**

**Tushar Sai Gupta 2K21/CO/495**

Under the supervision of and submitted to

**Mr. Rohit Beniwal**

Delhi Technological University

**Department of Computer Science Engineering**

Delhi Technological University, Shahbad Daultpur, Main Bawana Road,

Delhi-110042, India

# Declaration

"I hereby declare that this submission is my own work conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute and that, to the best of my knowledge and belief, it contains no material previously written by another, neither person nor material (data, theoretical analysis, figures, and text), which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text."

# Acknowledgement

I would like to express my gratitude to my teacher Mr. Rohit Beniwal for his guidance and suggestions for my project as well as for reinforcing the concepts that made it possible for me to bring this project into reality.

I would also like to thank my college, Delhi Technological University, for giving me the golden opportunity to do this project which will help me go a long way in my quest to improve my skills.

## Abstract



The government is the biggest producer and supplier of information in a country. It is split into multiple departments for the distribution of labor. Efficient storage, maintenance, and management of the vast amount of data stored within these government departments are of utmost importance. Due to advancements in information and communication technology (ICT), various e-governance solutions and applications have been implemented to improve government functioning.

However, yet, no proper software for the DMV has been made, at least in my country. Therefore, in this project, I aim to design a DMV management system that mitigates the problems that I and thousands of other applicants in the DMV have faced.

# Table of Contents

<b>Chapter no.</b>	<b>Topics</b>	<b>Page no.</b>
1	Introduction and Components used	6
2	Literature Review	11
3	Software Requirements Specification- Introduction, Overall Description, External Interface Requirements and System Features	12
4	System Analysis and Design	24
5	Implementation	25
6	Results and Analysis	35
7	Conclusion and Future Scope	36
8	References	37

# Chapter 1

## Introduction.



The DMV (Department of Motor Vehicles) is a subdivision of the government that is responsible for governing citizen needs such as driving licenses, vehicle registrations, conducting examinations, laws, amendments, traffic offences, and much more. It plays an important role in the daily lives of millions of people across the country.

However, a lot of important information like license examinee certification, license trial permit, and examination results are all examined, transferred, and stored in papers, which is a highly unreliable and primitive method.

Due to this, my family and I faced a lot of problems over the span of a few weeks in 2018 when I got my driver's license. Owing to unclear communication and bad infrastructure, we had to report to the DMV in person and devote entire days to some small tasks. The current system is inefficient and disregards the time of citizens. Hundreds of people across the nation face these problems daily. I plan to change that.

Hence, in this project, I aim to solve this problem by building a DMV management system that would help users do various tasks related to the DMV from the comfort of their own homes.

The functionalities that I implemented in this project are:

- 1. Register**
- 2. Login and Logout**
- 3. Apply for a license**
- 4. View previous license(s)**
- 5. Locate DMV branches**
- 6. View upcoming license examinations**
- 7. View registered vehicles**
- 8. View exam results**
- 9. Read through laws and amendments**
- 10. Check track record and previous traffic offences**
- 11. Contact authorities**



## Components used

### 1. Django



Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django's primary goal is to ease the creation of complex database-driven websites. Some well-known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket, and Mozilla.

The entire backend is based on this framework. Django handles everything from login authentication to database management. It is a one-stop shop for all the backend tools a developer needs for their web application.

### 2. SQLite



SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

The SQLite file format is stable, cross-platform, and backward compatible and the developers pledge to keep it that way through the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems and as

a long-term archival format for data. There are over 1 trillion SQLite databases in active use.

### 3. HTML



The standard markup language for texts intended to be displayed in a web browser is Hypertext Markup Language. Technologies such as Cascading Style Sheets and programming languages like JavaScript can help.

HTML elements are the components that make up HTML pages. Images and other objects, such as interactive forms, can be embedded in the produced page using HTML techniques. HTML allows you to construct organized documents by indicating structural semantics for text elements like headers, paragraphs, lists, links, quotations, and other elements.

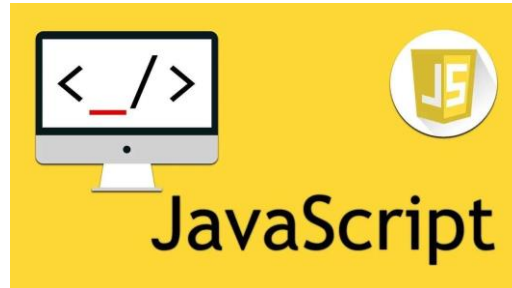
### 4. CSS



Cascading Style Sheets (CSS) is a language for specifying the appearance of a document written in a markup language like HTML. Along with HTML and JavaScript,

CSS is a key component of the World Wide Web. CSS is only utilized for the front-end design in this project, but it is crucial to the user experience.

## 5. JavaScript



JavaScript, sometimes known as JS, is a computer language that follows the ECMAScript standard. JavaScript is a multi-paradigm, high-level programming language that is frequently compiled just in time. Curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions are all features of this language.

JavaScript supports event-driven, functional, and imperative programming approaches as a multi-paradigm language. It contains APIs for working with text, dates, regular expressions, standard data structures, and the Document Object Model, among other things (DOM).

## 6. Bootstrap



Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It includes HTML and CSS-based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many others. Bootstrap is completely free to download and use.

## Chapter 2

### Literature Review

1. **Forcier, Jeff, Paul Bissex, and Wesley J. Chun. Python web development with Django. Addison-Wesley Professional, 2008.**

This book helped me understand the basics of Django while also strengthening my grip on the Python language.

2. **Shyam, Adamy, and Nitin Mukesh. "A Django Based Educational Resource Sharing Website: Shreic." Journal of Scientific Research 64.1 (2020).**

This research paper implemented an educational resource-sharing website using Django. This gave me ideas on how to use the Django framework to my idea and also showed me how a real-life problem can be solved using Web Development and Software Engineering.

3. **Holovaty, Adrian, and Jacob Kaplan-Moss. The definitive guide to Django: Web development done right. Apress, 2009.**

This paper helped me correct some of the mistakes I made when I was first learning and implementing Django.

4. **Ebrahim, Zakareya, and Zahir Irani. "E-government adoption: architecture and barriers." Business process management journal (2005).**

This paper helped me develop the idea I had about adopting E-governance services instead of physical methods.

5. **Burch, Carl. "Django, a web framework using python: Tutorial presentation." Journal of Computing Sciences in Colleges 25.5 (2010): 154-155.**

This paper helped me apply Django to real world problems because it presented the framework in an easy-to-understand presentation format.

## Chapter 3

### Software Requirements Specification Document

**Team members:** Vijay Kumar Sawhney (2K21/CO/515), Tushar Sai Gupta (2K21/CO/495)

**Project Name:** DMV Management System

#### 1. Introduction

**1.1. Category:** The proposed project is a web application.

**1.2. Purpose:** The purpose of this project is to make a DMV management system. This system will bring its own set of features for users such as applying for a license, viewing the status of many categories, etc. from the comfort of their own homes.

**1.3. Document convention:** To make the document more effective and readable, I used the best font style and font size and headings are bold for easier reader comprehension.

**1.4. Intended audience:** This project was made for teenagers above the age of 16 and young adults who are just considering getting a license. This can also be used by citizens across all age groups who are legally old enough to drive.

**1.5. Product Scope:** This project integrates the benefits of web applications over unreliable real-life interactions and transactions. It is especially beneficial for people who have a lot of responsibilities and are strapped for time.

**1.6. Reference:** The references I used for this project are listed below:

1. Forcier, Jeff, Paul Bissex, and Wesley J. Chun. Python web development with Django. Addison-Wesley Professional, 2008.
2. Shyam, Adamy, and Nitin Mukesh. "A Django Based Educational Resource Sharing Website: Shreic." Journal of Scientific Research 64.1 (2020).

## **2. Overall Description**

**2.1. Product Perspective:** This product will try to change the traditional way of interacting with the government. It will also help remove the stigma of government procedures being lengthy and time-consuming. This product will act as a better and more efficient way of managing various DMV-related tasks.

My approach towards programming was a highly methodical one since a Web application is created using various fields and principles.

**2.2: Product Function:** This product helps users manage various tasks related to the DMV like applying for a license, viewing license(s), viewing registered vehicles, viewing their track record, etc. The system maintains a profile where users can enter their citizenship number, email address, first name, and last name.

Citizenship is used to identify the licenses, vehicles, etc. registered to a particular user.

**2.3. User classes and characteristics:** The system will support two types of user privileges, Users and Administrators.

**Users** of the system should be able to use the following functionalities:

1. **Register**
2. **Login and Logout**
3. **Apply for a license**
4. **View previous license(s)**
5. **Locate DMV branches**
6. **View upcoming license examinations**

7. View registered vehicle(s)
8. View exam results
9. Read through laws and amendments
10. Check track record and previous traffic offences
11. Contact authorities

**Administrators** will have access to all the user functionalities and the special functionalities mentioned below:

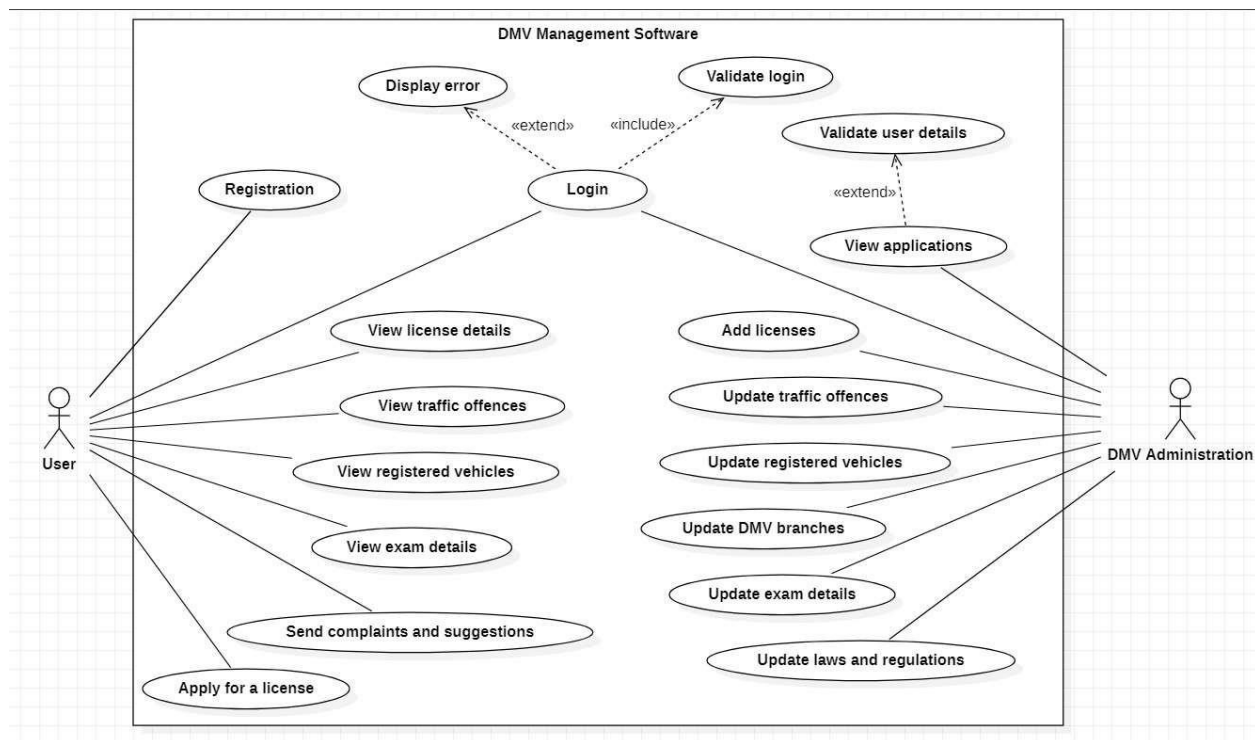
1. Update and add to license(s)
2. Update and add to available DMV branches
3. Update and add to upcoming examinations
4. Update and add to the registered vehicle(s)
5. Update and add to exam results
6. Update and add to laws and amendments
7. Update and add to any particular user's traffic offences and track record
8. Make changes to the database
9. Make changes to the website

**2.4. Operating environment:** The operating environment for the DMV management system is mentioned below:

1. Client/Server System
2. HTML, CSS, JavaScript, and Bootstrap frontend
3. Django backend
4. SQLite database

**2.5. Design and implementation constraints:** The interaction between actors and the system is displayed in the Use Case diagram below:

## Use Case Diagram



The use case diagram has two actors: Users and DMV Administrators. Users will have all the functionalities mentioned in the user classes and characteristics section, and administrators will have the ability to manipulate protected data.

**2.6. User documentation:** The general characteristics of the user of the website will be anyone from the targeted audience like young adults and teenagers old enough to drive, as well as anyone with a license who wants or needs to interact with the DMV.

**2.7. Assumptions and Dependencies:** My approach towards programming was a highly methodical one since a Web application is created using various fields and principles. I focused on developing the backend using Django, its included dependencies, and an integrated SQLite database.



### 3. External Interface Requirements

#### 3.1. User Interface:

Frontend: HTML, CSS, JavaScript, and Bootstrap

Backend: Django and SQLite

**3.2. Hardware Interface:** Windows and any browser which supports CGI, HTML, and JavaScript library.

**3.3. Software Interface:** Following are the software used for the social media web application.

Software Used	Description
Operating System	A web application because it would be cross-platform and lightweight.
Backend	Django was chosen because of its in-built support for user authentication and database integration.
Database	SQLite was chosen because of its quick and lightweight performance and simplistic nature.
Frontend	Bootstrap was used along with simple HTML, CSS, and JavaScript to make the website user-friendly and easy to comprehend.

**3.4. Communication Interface:** This project supports all types of web browsers. I am using simple electronic forms for interacting, browsing, sharing, etc.

## 4. System Features

**4.1. Description:** The DMV plays a crucial role in the government by giving people the authority to drive vehicles. This is required by almost all functioning adults in society for various purposes like going to their jobs, educational institutes, relatives, etc. This application would make the interaction between users and the DMV much more efficient.

**4.2. Priority:** The priority is given to

1. The enclosed environment of communication with the DMV.
2. The interactive and secure dispensing of details.
3. Encapsulating many features of the DMV into one application.

### 4.3. Functional Requirements:

**SQLite:** SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

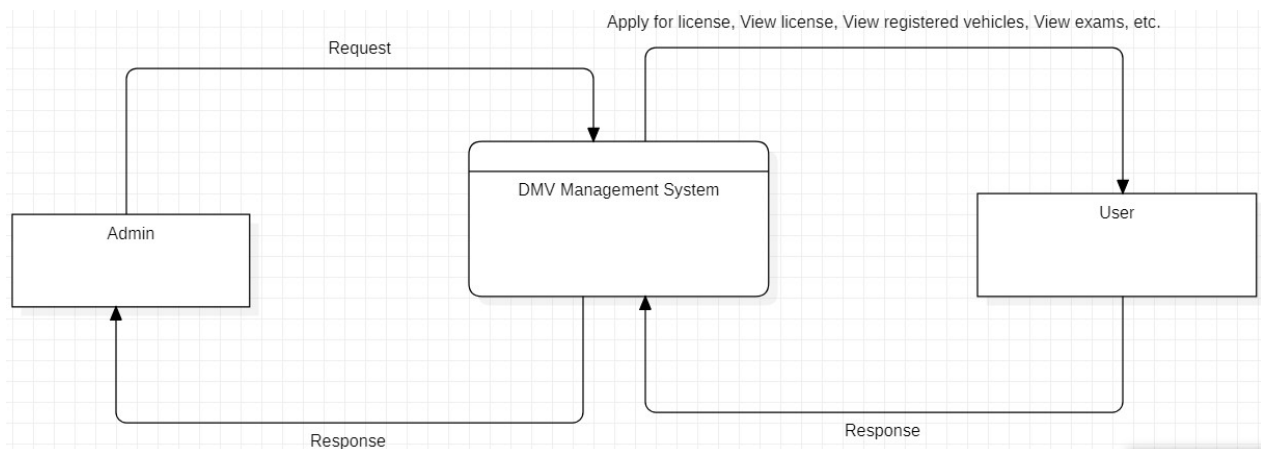
The SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

**Client/Server System:** The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the Django (also known as the back-end).

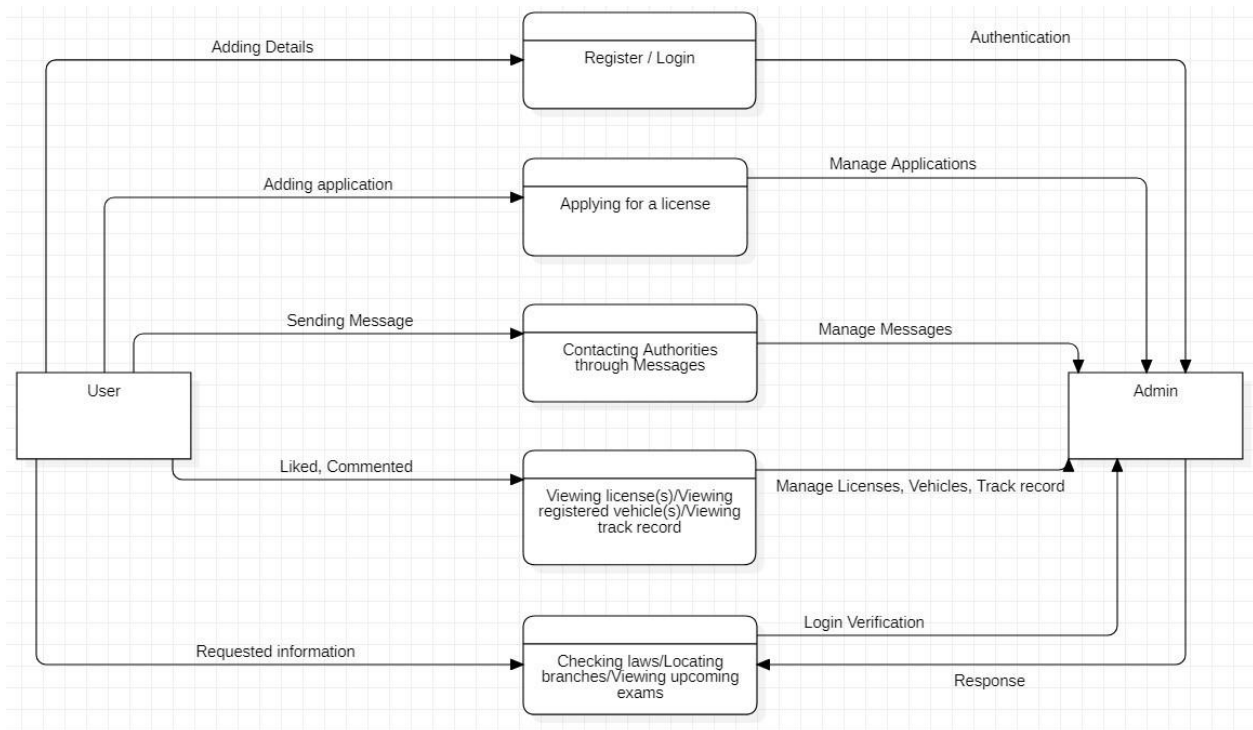
**4.4. Data Flow Diagram:** The Data Flow Diagram is a graphical representation of the flow of data in a system. The DFD I made includes all the functionality and flow of data through these functionalities connecting the entire system. This DFD includes the incoming data, outgoing data as well as stored data.

Here, we will see mainly 3 levels in the data flow diagram: -

1. **0-level DFD:** It is also known as a context diagram. It is designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

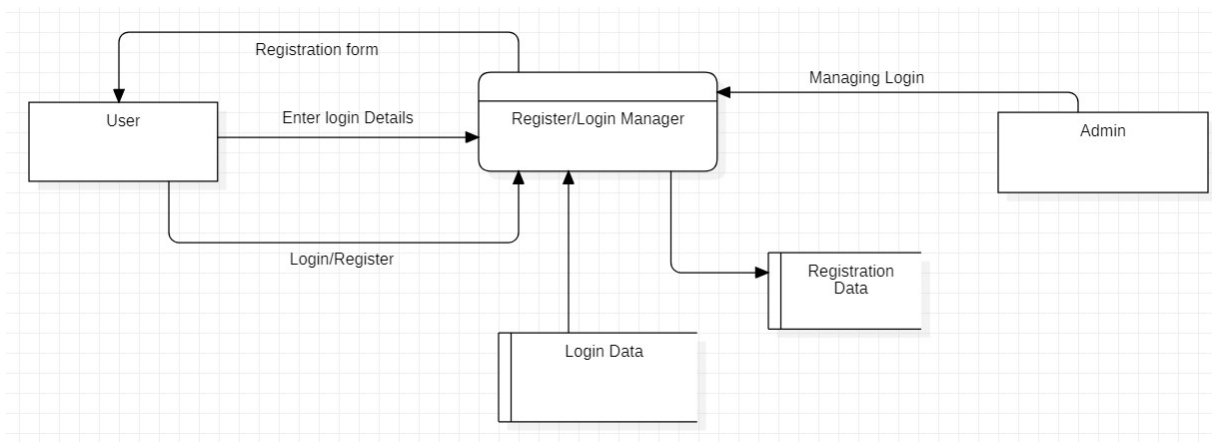


2. **1-level DFD:** In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and break down the high-level process of 0-level DFD into subprocesses.

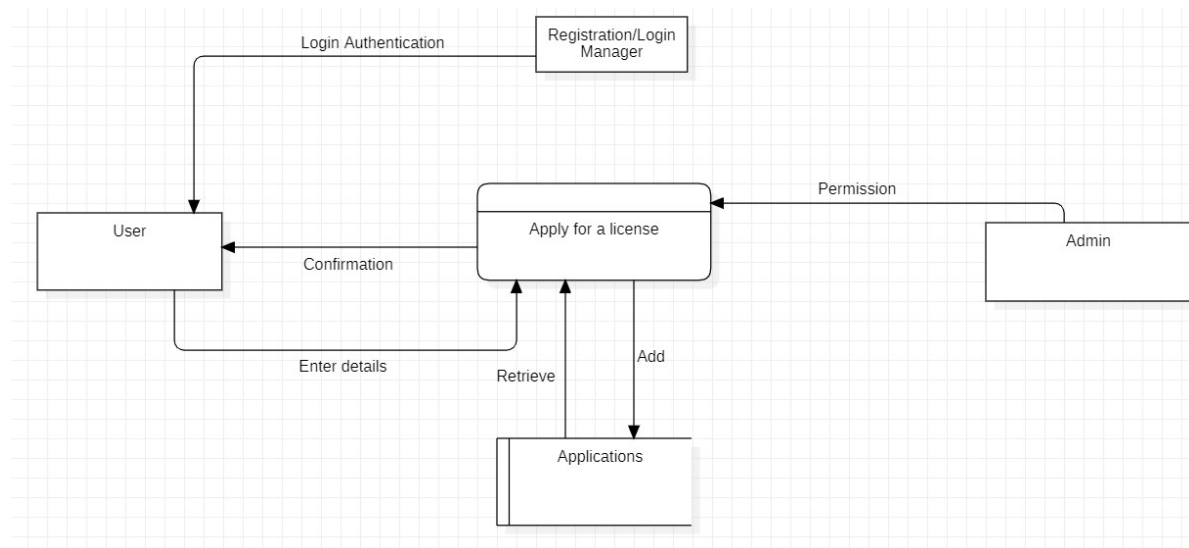


3. **2-level DFD:** 2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning. Some functionalities have been grouped into one because they share similar characteristics in terms of data flow. This was done for more clarity and simplification of the following data flow diagrams.

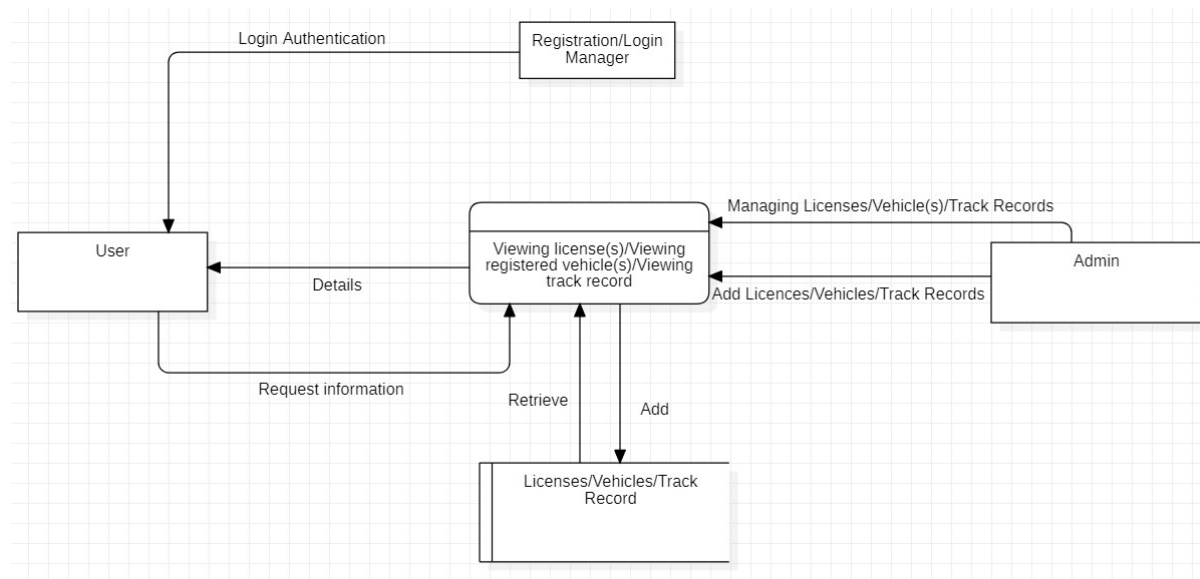
### 3.1. Register/Login Manager



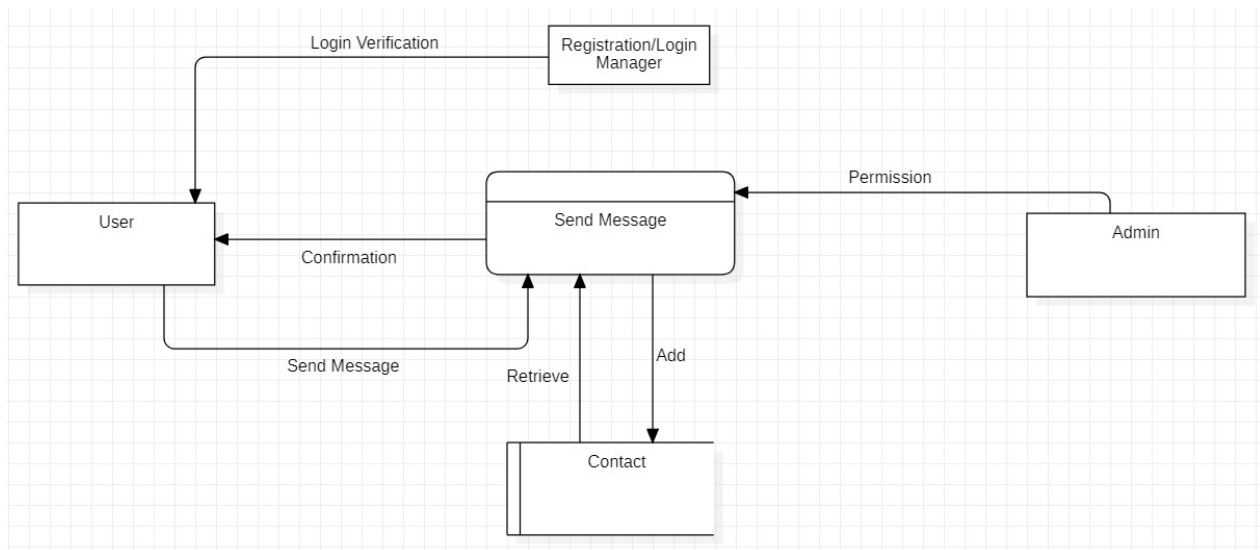
### 3.2. Applying for a license



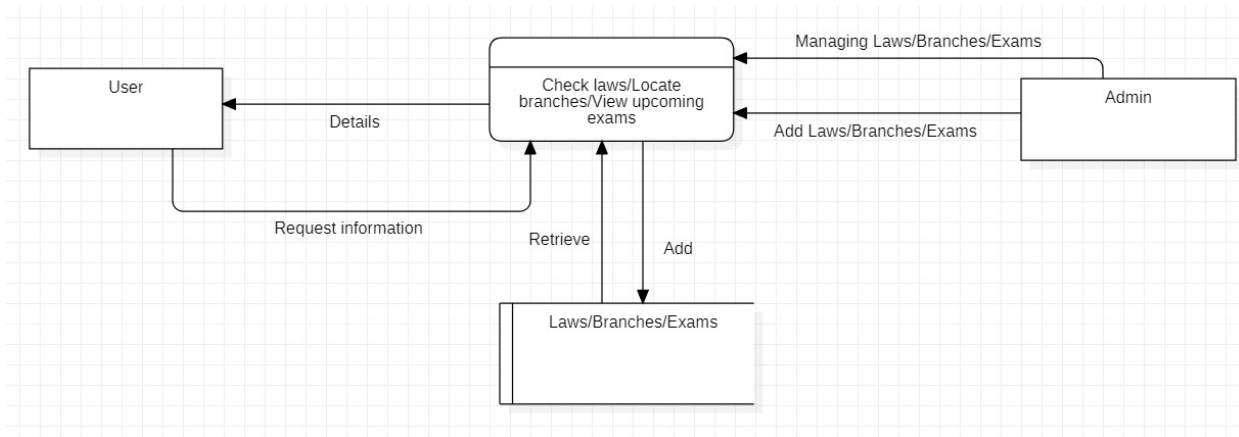
### 3.3. View License(s)/Vehicle(s)/Track record



### 3.4. Contact



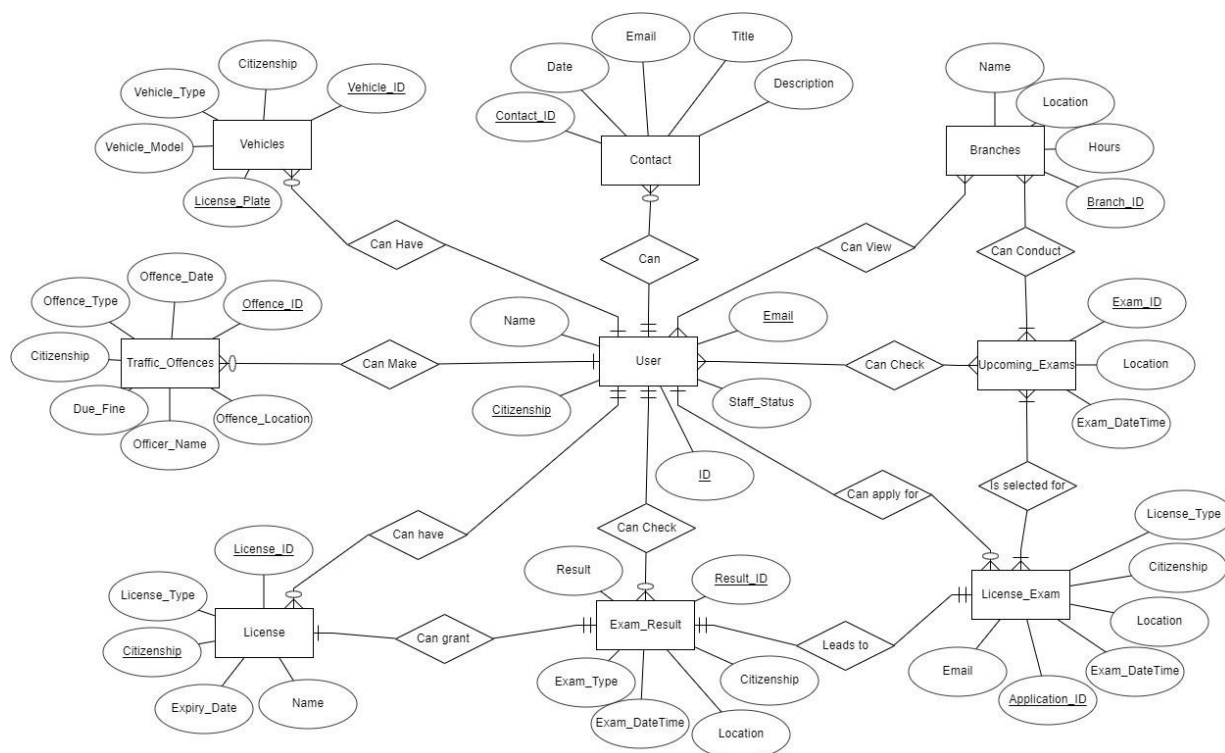
### 3.5. Check laws/Locate branches/View upcoming exams



## 5. Non- Functional Requirements

**5.1. Performance Requirements:** The processes involved in performing the implementation of the DMV management system are displayed in the **Entity-Relationship Diagram** below:

## Entity-Relationship Diagram



The entity-relationship diagram has 9 entities: User, Branches, Upcoming\_Exams, License\_Exam, Exam\_Result, License, Traffic\_Offences, and Vehicles (some entities were not included in the entity-relationship diagram for clarity purposes) and their respective attributes. Each entity has at least one unique attribute. The relationship between entities is also described.

**5.2. Safety Requirements:** Users can only register for accounts after entering their citizenship number. They can only see features tied to their own citizenship number like license(s), vehicle(s) etc.

**5.3. Security Requirements:** Security can be maintained by the website administration (DMV Officials). Administrators can manage users' profiles, license(s), vehicle(s), application(s), etc.

#### 5.4. Software Quality Attributes:

**Availability:** The project is aimed to be used by the target audience, especially by young adults who have many responsibilities and are strapped for time. Including these, teenagers old enough to drive and people that have any sort of interaction with the DMV will be able to use the website. The website will be launched by the government, named 'Transportation Management' in English or 'Yatayat Vyavastha' in Nepali for more accessibility. The application can be used as a blueprint for other government departments as well.

**Correctness:** The main component of my project, Database Management, was carried out in SQLite. The database is fast, reliable, lightweight, and has good error detection and management tools. The Django backend was responsible for populating this database based on input form fields by users or through the admin portal by the website administrators.

**Maintainability:** The implementation of the website is done on a local host server, but the application is complete and can be deployed to a hosting website such as Heroku or Firebase. Even in the relatively small scope of local hosting, Django performs various integrity and correctness checks so that the application runs perfectly and interactions between the frontend, backend, and database are uninterrupted.

**Usability:** The application I have presented gives the user a simple interface to register, log in, apply for a license, view license(s), locate DMV branches, view upcoming examination details, view registered vehicles, view exam results, read through laws and amendments, check their track record and contact authorities with suggestions, complaints or help requests. Additional features for administrators have been added. These features will increase the usability of my web application.



## Chapter 4

### System Analysis and Design

In this project, I am going to be making a DMV Management System using HTML, CSS, JavaScript, and Bootstrap for the frontend, Django for the backend, and SQLite for the database.

SQLite is a great database choice instead of traditional MySQL because this lightweight application makes scaling and managing the database much easier. This is why SQLite is the most used database management framework in the world.

Necessary features such as login authentication are done by Django itself. It also supports a base user class on which we can improve and add features upon.

## Chapter 5

### Implementation

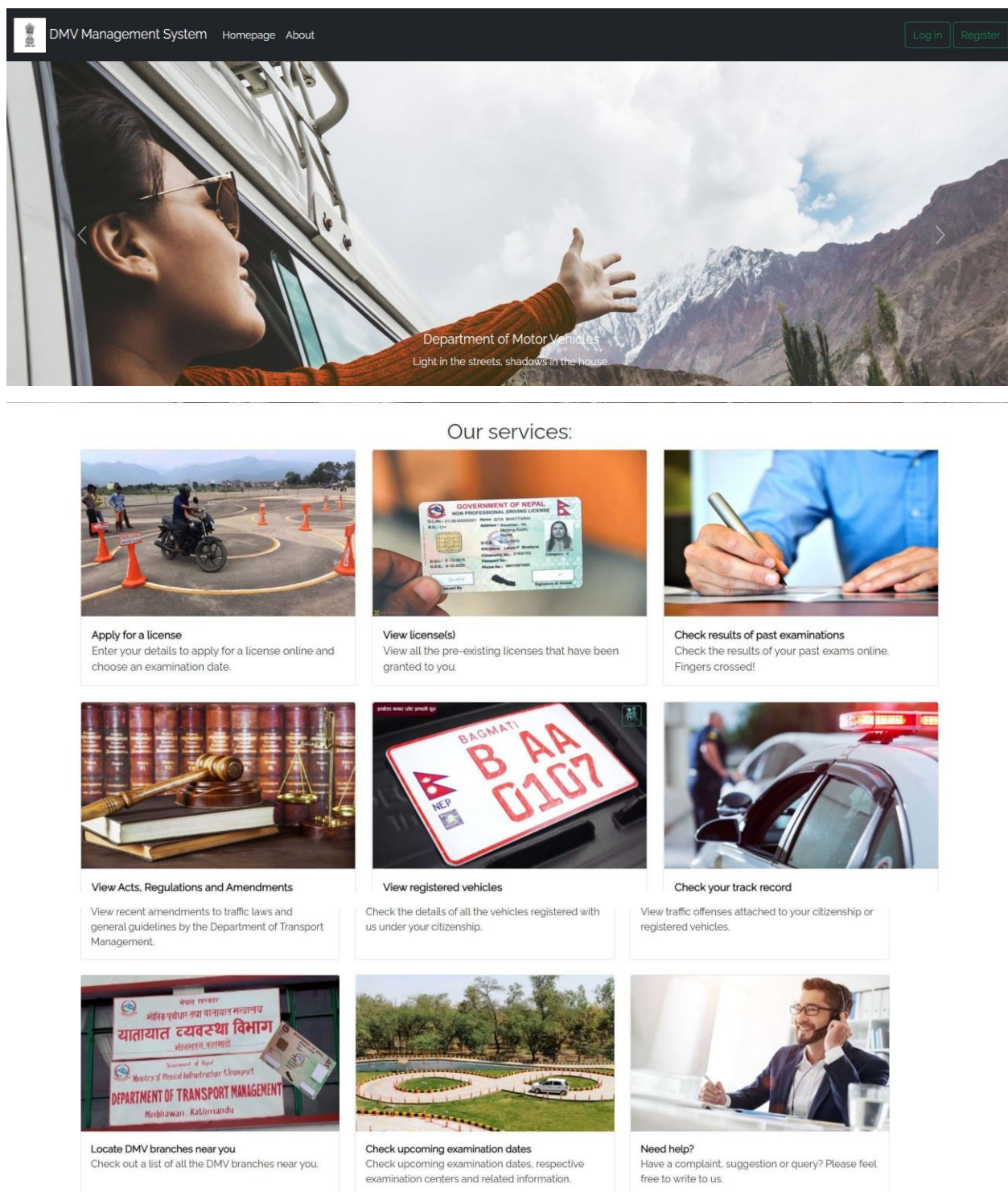
This project turned out to be large in scale. As stated, multiple times before in this report, this application was built using HTML, CSS, JavaScript, and Bootstrap for the frontend, Django for the backend, and SQLite for the database.

The code was written entirely from scratch by me in Visual Studio Code based on the vision for the application that I developed due to the planning above.

The entire project took over two months to complete from its inception which included learning about the different phases in software engineering, proposal, planning, implementation, Use Case Diagram, Entity-Relationship Diagram, Data Flow Diagram, and report.

The final application is being run on my localhost, but is ready to be deployed online.

Briefly, this is what the website looks like:



## User functionalities

### 1. Register

Register

×

---

First Name

Last Name

Citizenship No.

Email address

Password

Register

Users can register to this website using their First name, Last name, Email address, Citizenship number, and Password. Additional security measures like citizenship number verification and email verification can be added as well.

## 2. Login and Logout

Log In

✕

---


Email Address

Password

Log in

Users can log in and subsequently log out when they wish. For this, email and password are required.

### 3. Apply for a license


DMV Management System
[Homepage](#)
[About](#)
Welcome, Vijay Kumar!
[Log out](#)

## Apply for a license

Select the type of license you want to apply for:

Learner

Select date and time of examination:

May 29, 2023 2:00 PM

Select location of examination:

Chabahil

Submit

Users that are logged in can apply for a license trial examination. They must enter the type of license, date of examination, and location of the examination. Important details like name and citizenship are fetched from the user's account, and application date and time are taken by the backend. This is stored as an object in the `apply_license_db` table in the database.

### 4. View previous license(s)

View all the license(s) registered under your citizenship number:

Name	Citizenship number	Liscence type	Expiry Date
Siddhant Bikram Shah	3000	Four-Wheeler	Sept. 1, 2023
Siddhant Bikram Shah	3000	Two-Wheeler	Nov. 1, 2023
Siddhant Bikram Shah	3000	Learner	Nov. 16, 2021

All licenses registered to a user's citizenship number can be viewed by the user. They cannot view the details of other users. Important details such as the Expiry Date can also be checked. These licenses are updated by the administrator.

## 5. Locate DMV branches

### Official DMV Branches:

Branch Name	Location	Working hours
Bike/Scooter Trial Center	Chabahil	11:00-15:00
Easy Way Trial Center	Araniko Highway, Thimi	5:00-19:00
Vehicle Academy & Research	Chandragiri	6:00-18:00
Suzuki Driving School	Ring Road, Lalitpur	6:00-18:00
Namaste Nepal Driving Centre	Tokha Road	6:00-18:00

Official DMV branches can be viewed by any user of the website. Logging in is not required here. Branch Name, Location, and Working hours can be seen here.

## 6. View upcoming license examinations

### Upcoming license examination dates:

Branch Name	Location	Test date	Test time
Bike/Scooter Trial Center	Chabahil	June 1, 2023	8 a.m.
Easy Way Trial Center	Araniko Highway, Thimi	June 1, 2023	noon
Vehicle Academy & Research	Chandragiri	June 1, 2023	3 p.m.
Suzuki Driving School	Ring Road, Lalitpur	June 1, 2023	2 p.m.
Namaste Nepal Driving Centre	Tokha Road	June 1, 2023	1 p.m.
Easy Way Trial Center	Araniko Highway, Thimi	June 1, 2023	11 a.m.

Upcoming license examinations can be viewed by any user of the website. Logging in is not a necessity for this page. The details of upcoming examinations can be viewed which include Branch name, Location, Test date, and time.

## 7. View registered vehicle(s)

View all the vehicles registered with us under citizenship number:

Name	Citizenship number	Vehicle Type	Vehicle Model	License Plate Number
Siddhant Bikram Shah	3000	Four-Wheeler	Mercedes Benz Maybach S560	BA.11XA 8826
Siddhant Bikram Shah	3000	Many-Wheeler	Bharat Benz 1923C Tipper	BA.23CHA 7086
Siddhant Bikram Shah	3000	Two-Wheeler	Vespa Piaggio	BA.15PA 1085

Vehicles registered with the DMV under a particular user's citizenship number can be viewed by that user. This includes details like License Plate Number, Vehicle Model, Vehicle Type, etc.

## 8. View exam results

Check your exam result(s):

Name	Citizenship number	Exam location	Exam date and time	Exam type	Result
Siddhant Bikram Shah	3000	Araniko Highway, Thimi	Nov. 17, 2021, 8 a.m.	Written	Passed
Siddhant Bikram Shah	3000	Araniko Highway, Thimi	Nov. 17, 2021, 10 a.m.	Practical	Failed
Siddhant Bikram Shah	3000	Araniko Highway, Thimi	Nov. 19, 2021, 10 a.m.	Practical	Passed

The results of past examinations can be viewed here. All previous records are displayed along with the result, exam location, exam date and time, exam time, etc.

## 9. Read through laws and amendments

### Official Traffic Laws and Amendments

Law Name	Date of issue	PDF Link
Motor Transport Rules, 2054	June 26, 1997, 7:31 a.m.	<a href="#">View</a>
Traffic Signs Review, 2067	May 17, 2010, 7:39 a.m.	<a href="#">View</a>
Urban road standards, 2019	May 11, 2019, 7:36 a.m.	<a href="#">View</a>

Laws and amendments can be viewed by any user of the website. Logging in is not required here. Users can view the Law's Name, date of issue and also go through an online .pdf file of their issue. These laws can be updated by administrators.

## 10. Check track record and previous traffic offences

Your previous traffic offence(s):


Name	Citizenship number	Offence type	Due fine Fee status	Offence Date and Time	Offence Location	Officer Name
Vijay	3000	Hit and Run	5000	Jan 15, 2021, 4:15p.m.	Delhi	Tushar
Tushar	2000	Speeding	10000	Oct 02, 2021, 11:30p.m.	U.P	Vijay
Nish	8000	Speeding	10000	Oct 02, 2021, 11:30p.m.	U.P	Sid

Previous traffic offences tied to the citizenship number of a user can be viewed. This includes Offence type, Due fine, Fee status, Offence date and time, Offence location, and the name of the officer who filed the offence. These are again updated by the administrator.

## 11. Contact authorities

Contact Us

Select the type of request:
 

Help
 

Describe the situation. How can we help?

Users can contact DMV authorities through the contact us page. Users need to enter details such as their email address, the title of message, content, and type of request to be made to the authorities. This is stored as an object in the `contact_dbtable` in the database.



## **Administrator functionalities**

### **1. Login and Logout**

The administrator portal is different from the normal user portal. Here, administrators and superusers that are already registered with the site can log in.

### **2. Manage users**

Administrators can view the users table and see normal users, staff users and super-users of the database. The administrators are super-users and have the most control of the database according to the user hierarchy of the Django user interface. Administrators can view non-sensitive data such as email address, username and full name but cannot view passwords. Passwords are encrypted and stored by Django automatically and cannot be seen by anyone.

### **3. Manage other database tables**

Administrators are free to check and manage all the tables in the database. They can check data from any table in case a discrepancy happens.

### **4. Manipulate database tables**

Administrators are responsible for updating data like new license(s) and newly registered vehicle(s) to the database. So, they can freely create database objects here and update the database freely through an easy form with a simple interface. Administrators can also view and manipulate previous database entries.

## Chapter 6

### Results

The results that I got from just a few months of development were very promising. Adding on to that, I worked on this whole project myself. The application is basic in this stage, but easily scalable and shows a lot of potential. Many features have not been implemented yet, but they have all been polished. I hope that these kinds of applications and proposals are seen more frequently in developing countries in India and Nepal for their modernization.

### Analysis

After analyzing what I accomplished in this project, I am pretty confident that I can now work on bigger projects or even add more features to this very project. This project helped me reinforce my Software Engineering concepts as well as strengthened my grip on Python and frameworks like Django.

Some features that could be added are the ability to apply for vehicle registration, study material for the written examinations, resources, or locations for practicing the practical examination, etc. Additionally, a stronger document verification system can be implemented in the registration phase so that users cannot enter wrong citizenship and name details.

## Chapter 7

### Conclusion

Through this project, I learned a lot about web development and building simple, lightweight, and scalable applications for a large target audience. It reignited my passion for web development and inspired me to explore more frameworks and libraries in the future to make my applications even better.

Even more importantly, it helped me learn how to apply software engineering to a real-life problem and try to solve it. If ideas like this are developed on a large scale, it could lead to great developments in my country.

### Future Scope

Some features I could implement in this project after devoting more time to it and becoming more skilled in web development are:

1. Making the Website more user-friendly
2. Creating a better design for the website
3. Making different designs and layouts for different types of functionalities.
4. Using more frontend libraries is more advanced than Bootstrap.
5. More security through citizenship verification.
6. Reduce the vulnerability of the website through email or OTP verification.
7. Better user interface for administrators.

I hope ideas like these are seriously considered by the government because it could help thousands of people make their daily lives much easier and more efficient. Research and funding into developments like this are key in leading us into the digital future.

## Chapter 8

### References

1. Forcier, Jeff, Paul Bissex, and Wesley J. Chun. Python web development with Django. Addison-Wesley Professional, 2008.
2. Shyam, Adamya, and Nitin Mukesh. "A Django Based Educational Resource Sharing Website: Shreic." Journal of Scientific Research 64.1 (2020).
3. Holovaty, Adrian, and Jacob Kaplan-Moss. The definitive guide to Django: Web development done right. Apress, 2009.
4. Ebrahim, Zakareya, and Zahir Irani. "E-government adoption: architecture and barriers." Business process management journal (2005).
5. Burch, Carl. "Django, a web framework using python: Tutorial presentation." Journal of Computing Sciences in Colleges 25.5 (2010): 154-155.
6. Django documentation: <https://docs.djangoproject.com/en/3.2/>
7. SQLite documentation: <https://www.sqlite.org/docs.html>
8. Bootstrap documentation: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
9. Django tutorial: <https://www.youtube.com/watch?v=JxzZxdht-XY>