**Regional Transport Office(RTO) Management System**

**ABSTRACT**

The RTO, Regional Transport Office, is the governmental organisation responsible for issuing driving licences, maintaining databases of vehicles and sells personalised registration. It has been observed since years that the RTO is not able to deliver quality public services to the citizen without delay. Hence, this project is aimed at developing a computerised system for the functioning of RTO. This system will reduce the manpower required in an RTO and make the existing system fast and efficient.

In short, the website is used for issuing of licence. An individual can apply for learning licence and driving licence online. Accordingly slots and dates for test are generated for the respective test. Moreover, this application sends an alert message for renewal of driving licence to an individual when his driving licence is about to expire. The applications received will be verified and approved the RTO officials. The applicant can monitor the status of their application and download the approved licence. An individual can also apply for vehicle registration and submit any complaints.

The system is implemented as 2-tier approach with a backend database handled by the system administrator and a web browser as the front end client. This document will discuss each of the underlying technologies used to create and implement *Online RTO Management* website. To implement this we have used PHP, which is platform independent and therefore, can be run on all major operating systems. PHP provides support to all major servers like Apache and databases like MySQL. Since it uses its own memory, the loading time is decreased and processing time is increased. Next we have used HTML, JavaScript, CSS and Bootstrap for front-end implementation. They provide a front-end development framework to create fully responsive web pages and define proper styles and presentation of the document. We have used Apache web server, a powerful web server program which is compatible and supports various protocols like FTP, SMTP etc. This website is run on the Local Host for the current time to provide security from being hacked. Lastly, MySQL is used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

1.1Introduction:

This is a web based project which is useful for all those who are looking for a driving licence for their vehicle. This portal provides a complete procedure for acquiring the driving licence. It also provides road safety signs and suggestions to ensure smooth traffic flow without bottlenecks or mishaps. Information about events conducted by state transport ministries are published under regular basis.

There will also be a provision for an online questionnaire where any individual can post their query. All these queries will be answered by some of the esteemed, selected transport officer of the country. Queries can be directed to a particular officer also. Any notifications, circulars, initiation activity or changes in any transportation law will be posted on the website so that every individual is abreast with modern methods and practices of transport.

Pollution control is one of the main factors that needs to be taken care for the cleanliness of the city. The website constantly updates new pollution air norms and rate of acceptable pollution from a vehicle so that each individual can ensure their vehicle accordingly to avoid fine. Information pages should be dynamic in nature so that administrator can change it. There will be different logins provided for administrator and inspector. Based on the type of login, the users operates different functionalities provided to him.

1.2 Project Design

Designing of a website requires a backbone, that is, a relational database. A correct relational database needs to be designed – its structure, various relation schemas, relationship between then and datatypes and constraints that should hold on for the data should be mentioned clearly. This calls for the need of data models. There are two types of data models:

1. High level or Conceptual data model: It gives a concise description of the data requirements of the users and includes detailed descriptions of the entity types, relationships, and constraint. They are usually easier to understand and communicate with non-technical users. Example- Entity Relationship Diagram (ER Diagram). Figure 1 shows the ER diagram of the *Online RTO Management* application.
2. Low-level or Physical data model: It gives a detailed information of how data is processed and stored in a computer.

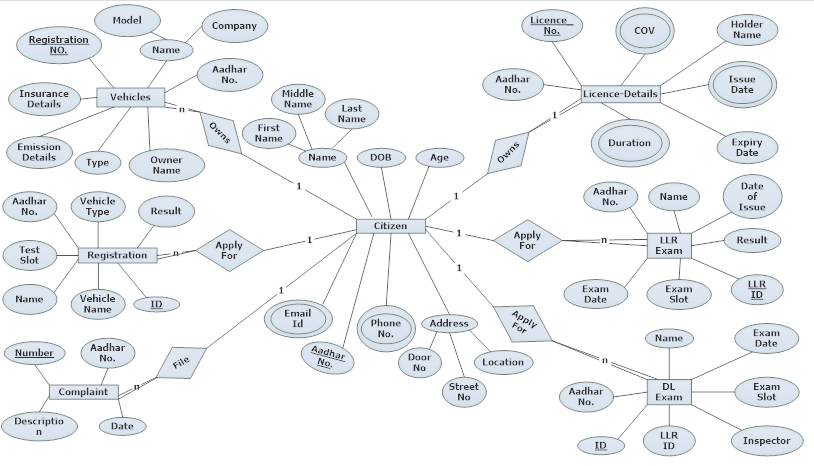


Figure1

1.3 Database Design:

In this section, the basic relational schemas of the RTO database are shown along with the marked primary keys and foreign keys:

**1. Vehicles**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Registration No. | Varchar | Primary key |
| 2 | Type | Varchar |  |
| 3 | Owner\_Name | Varchar |  |
| 4 | Company | Varchar |  |
| 5 | Model | Varchar |  |
| 6 | Insurance­\_details | Varchar |  |
| 7 | Emission\_details | Varchar |  |
| 8 | Aadhar\_No. | Integer | Foreign key |

**2. Licence-details**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Licence\_No. | Varchar | Primary Key |
| 2 | COV | Varchar | Primary Key |
| 3 | Holder\_Name | Varchar |  |
| 4 | Issue\_date | Date |  |
| 5 | Duration | Integer |  |
| 6 | Expiry\_date | Date |  |
| 7 | Aadhar\_No. | Integer | Foreign Key |

**3. Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | ID | Varchar | Primary key |
| 2 | Test\_Slot | Varchar |  |
| 3 | Vehicle\_Name | Varchar |  |
| 4 | Vehicle\_Type | Varchar |  |
| 5 | Name | Varchar |  |
| 6 | Result | Varchar |  |
| 7 | Aadhar\_No. | Integer | Foreign Key |

**4. LLR\_Exam**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | LLR\_ID | Varchar | Primary Key |
| 2 | Name | Varchar |  |
| 3 | Exam\_Date | Date |  |
| 4 | Exam\_Slot | Varchar |  |
| 5 | Result | Varchar |  |
| 6 | Date\_of\_Issue | Date |  |
| 7 | Aadhar\_No. | Integer | Foreign Key |

**5. DL\_Exam**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | ID | Varchar | Primary key |
| 2 | Name | Varchar |  |
| 3 | Exam\_Date | Date |  |
| 4 | Exam\_Slot | Varchar |  |
| 5 | Inspector | Varchar |  |
| 6 | Aadhar\_No. | Integer | Foreign Key |
| 7 | LLR\_Id | Varchar | Foreign Key |

**6. Complaint**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Number | Integer | Primary key |
| 2 | Description | Varchar |  |
| 3 | Date | Date |  |
| 4 | Aadhar\_No. | Integer | Foreign Key |

**7. Citizen**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Aadhar\_No. | Integer | Primary Key |
| 2 | First\_Name | Varchar |  |
| 3 | Middle\_Name | Varchar |  |
| 4 | Last\_Name | Varchar |  |
| 5 | DOB | Varchar |  |
| 6 | Door\_No. | Varchar |  |
| 7 | Street\_No. | Varchar |  |
| 8 | Location | Integer |  |
| 9 | Age | Varchar |  |

**8. Contact**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Aadhar\_No. | Integer | Primary key |
| 2 | Phone\_No. | Integer | Primary key |

**9. Mail**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Aadhar\_No. | Integer | Primary key |
| 2 | Email-Id. | Varchar | Primary key |

1.4 Non Functional Requirements:

1. Performance requirement: The system should respond quickly to the user input. Loading of application forms, submission, test result generation should be fast. Even retrieval of data should be fast.
2. Security: Only the admin should be able to access the databases and perform certain jobs.
3. Fault tolerance: The system should be fault tolerant in order avoid any inconsistency.
4. Maintainability: The system should have high maintainability in terms of correctness, adding new functionalities and should be highly adaptive.
5. Portability: The system should work independently of available hardware. It should work on all PCs in the same way.
6. Extensibility: We should be able to add new features and update the application frequently.
7. Usability: The application must be user friendly and unambiguous.

1.5 Functionality:

This project is an attempt for alleviating the delay in the working of RTO offices and thereby enabling RTO office to provide quality of service. The project consists of three modules:

1. Citizen, who aspires to get a driving licence or register his vehicle.
2. RTO Inspector
3. Administrator

Apply for LLR

Citizen who craves for a driving licence, starts by applying for a *learning licence registration* (LLR). The system is expected to allow only those citizens who are 18 years old or above for registration. All the minor applicants will be denied of further processing. Once his age is verified, he is asked for the category of vehicle (COV) for which he wants to apply. After selecting COV, the system will generate an appropriate exam date, exam-id and password. On the exam day, the administrator will inform the citizen his exam id and password. There is a time limit allotted for the exam and on submitting, the result will be displayed instantly and stored in a database.

Apply for DL

Once the citizen has received his LLR, he will apply for driving licence. The system will check whether the LLR issued period is more one month and less than six months or not. If the LLR issued period is less than one month, the citizen will be asked to wait till one month completes. If the LLR issued period is more than six months, the system will ask the citizen to re-apply for LLR. On a legitimate time period of the LLR, the citizen will be asked for the COV to match whether LLR has been issued for the same COV for which the citizen is applying driving licence for. On a correct match, the system will generate test date, test-id and assign inspector via inspector-id and the citizen will appear for DL exam.

Vehicle Registration:

The citizen will be asked for COV, Vehicle Company and model. Accordingly, the system will generate the test date, test-id and inspector for the vehicle to be examined.

Inspector:

An RTO inspector mainly has two duties:

1. Entering the results of the driving licence (DL) tests into its corresponding database.
2. Entering the results of vehicle registration tests into its corresponding database.

Based on the result given by the inspector, if the DL aspirant has passed the test his details will automatically get stored in the database. Vehicle registration also undergoes a similar process.

Administrator:

The administrator has a login page. He can access the database of driving licence holders and the registered vehicles. Prior to one month of the expiry of driving licence and RCs book of the vehicle, the administrator will send an alert message to the respective individual. He has a scope of information- aadhar number, exam-id and password of all candidates who are going to appear for the exam on that given day. He can add new inspectors into the database or remove some of the existing one from the same. When any aspirant has got the driving licence successfully, then the administrator will remove his record from the database of LLR or DL in order to maintain a fresh copy of both database.

1.6 Conclusion:

This project *Online RTO Management* brings out an improvement over the existing RTO system by reducing the processing delay and allowing RTO to provide quality of service to the citizen. It overall increases the efficiency of the RTO office. People need not stand in long queues for application. All this pre-registration task can be done online through this application. Any doubts or queries will be answered properly. This project even eliminates the presence of middle man from the whole process and thereby decreasing the degree of corruption in the state.

1.7 References:

1. https://www.tutorialspoint.com/dbms/er\_diagram
2. <https://en.wikipedia.org/wiki/Functional_dependency>
3. <http://transport.karnataka.gov.in/>
4. [www.w3schools.com/](http://www.w3schools.com/)
5. <https://httpd.apache.org/>
6. http://localhost/xampp/splash.php

INDEX:

1.1 Introduction

1.2 Project Design

1.3 Database Design

1.4 Non-Functional Requirements

1.5 Functional Requirements

1.6 Conclusion

1.7 References