Index	Page No.
1. Introduction	02
1.1 Objectives	02
1.2 Identification of Need	02
1.3 Preliminary Investigation	03
1.4 Problem Domain	03
1.5 Solution Domain	03
1.6 Platform Specification	04
1.7.1 Hardware	04
1.7.2 Software	04
2. System Requirement Analysis	7
2.1 Information Gathering	7
2.1.1 Functional Requirement	7
2.1.2 Non- Functional Requirement	8
2.2 System Feasibility	9
2.2.1 Operational	9
2.2.2 Technical	9
2.2.3 Economical	10
3. System Design	12
3.1 Use Case Diagram	12
3.2 Class Diagram	13
3.3 Sequence Diagram	14

3.5 Activity Diagram	15
4. Design	17
4.1 Data Design	17
4.1.1 ER Diagram	17
5. Implementation	19
5.1 Implementation of Modules	19
5.2 Results	19-27
6. Testing	29
6.1 Testing Objectives	29
6.2 Testing Scope	29
6.3 Testing Principles	29-31
6.4 Testing Method Used	31
6.5 Test Cases	32
6.6 Sample Test Data and Results	33-35
7. Limitation	37
8. Future Scope	39
9. Conclusion	41
10. Bibliography and References	43

LIST OF FIGURES

Figure No.	Title	Page No.
1.	Use Case Diagram(Admin)	12
2.	Use Case Diagram(User)	12
3.	Class Diagram	13
4.	Sequence Diagram(Admin)	14
5.	Sequence Diagram(User)	14
6.	Activity Diagram	15
7.	ER Diagram	17
8.	Home page preview	19
9.	Welcome section	20
10.	Organization Information	20
11.	Event Updates	21
12.	About Us	21
13.	Learner’s Licence Registration	21
14.	Login for Learner’s Licence	22
15.	Category Selection	22
16.	Record Created	23
17.	Exam Details	23
18.	Vehicle Registration	23
19.	Login for Vehicle Registration	24
20.	Vehicle Details	24
21.	Verification	24
22.	Driver’s Licence	25
23.	Login for Driver’s Licence	25
24	Category selection	26
25.	Record Created	26
26.	Complaint	26
27.	Complaint Filed	27
28.	Gallery	27
29.	Inspector Login	33
30.	Correct Username No Password	33
31.	No Email Wrong Password	34

32.	Correct Username Wrong Password	34
33.	No Username No Password	34
34.	Correct username Correct Password	35

LIST OF TABLES

Table No.	Title	Page No.
1	Inspector login form	32

INTRODUCTION

1. INTRODUCTION

1.1 Objective

The main objective of the project on RTO Management System is to manage the details of vehicle, customer, driving licence, registration, vehicle type. It manages all the information about vehicle, vehicle maker, vehicle type, vehicle. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the vehicle, customer, vehicle maker, driving licence. It tracks all the details about the driving licence, registration, vehicle type.

1.2 Identification of Need

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the processing of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in systematic order. There used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there world never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business. For this reason we have provided features present system is partially automated, actually existing system is quite laborious as one has to enter same information at three different places. Points to be considered:-

1. Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.
2. Details of the information needed for each document and report.
3. The required frequency and distribution for each document
4. Probable sources of information for each document in report.
5. With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse.

So the proposed system helps in saving the time in different operations and making information flow, giving valuable reports easy.

1.3 Preliminary Investigation

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

1.4 Problem Domain

The problem with offline or manual RTO system are the following: -

- Time Consuming

The cycle of having to go again and again to the RTO office even for the minimal tasks like applying for learners licence, for getting a test date. It is even more time consuming for the customers living far away from the RTO office.

- Lack of Security of Data

The data stored for this kind of method is sometimes only stored in a databook. The officers has a dedicated databook for a number of citizens. All the data that they gather every day is stored in those databooks.

- More Man Power

It requires many officers to organize people coming for the test and it also requires large volume of pare work and also maual calculations are needed to be done.

1.5 Solution Domain

The solution for manual offline attendance is taking attendance online.

The advantages of online attendance system are: -

- Ensure Data Accuracy

The data collected by the automated system is more accurate than manually collection of data. Each person has a personalized email that they use to access the forms. This makes it impossible for the data to be manipulated by anyone whether it is by mistake or intentionally. Only Admin can give access to anyone required so the data cannot be modified by everyone.

- Security of Data

When it comes to online RTO Management, there are web-based and cloud-based management systems. Each has its own advantages and disadvantages. When it comes to security, though, cloud-based RTO management systems have the upper hand. This is due to the added layers of security upon the development of the system.

- Proper Control of Higher Officials

In this system the admin can easily check whether everyone is doing their work or not just by entering their id's and also it is easy to check every officials actions.

- Minimum Time Required

As online RTO Management System could be accessed by anyone from anywhere, it does not require to visit RTO office again and again which saves a lot of time.

1.6 Platform Specification

1.6.1 Hardware

Server Side:

Operating System: Windows 7/xp, Windows ME

Processor: Pentium 3.0 GHz or higher

RAM: 1 Gb or more

Hard Drive: 10 GB or more

Client side:

Operating System: Windows 7 or above, MAC or UNIX.

Processor: Pentium III or 2.0 GHz or higher.

RAM: 1 Gb or more

1.6.2 Software

Operating system: Windows 7 or above

Editor: VS Code IDE

Technology: PHP Language

Back end: MySQL Database

Scripting Language: HTML, CSS, JS

Browser: Google Chrome, Safari, Firefox

SYSTEM REQUIREMENT ANALYSIS

2. SYSTEM REQUIREMENT ANALYSIS

2.1 Information Gathering

In the very hope that by the end of the implementation of this system many of the problems faced in the by the current system shall be resolved and it will meet all user requirement, user friendly as users were involved in all stages of the development system.

2.1.1 Functional Requirement

- Adding a new consumer:

Function: Sign up a new consumer to the system.

Priority: Top (Required for first release)

Requirements: To add a new user to the system, all of them should have registered in the RTO office before they can register their vehicle or perform any other functionality. The data is inserted by the person and further it is forwarded to admin to add in the database which is only required once.

- Use the system to request learner's licence:

Function: apply for learner's licence

Priority: Top (Required for everyone who drives)

Requirements: When the user wants to make any request, they need to enter login id and password to login to the website to generate request. If the login id or password is incorrect the user need to contact the RTO office using the site then the new password will be sent to their emails which is used by user to login and can be changed later. Applying for learner's licence only requires the user to enter the type of vehicle and their request will be generated.

- Applying for Driver's licence:

Function: The user will request for driver's licence.

Priority: Top (Required for every user)

Requirements: To request for driver's licence the user must already have a learner's licence, if they have learner's licence than only they can apply for driver's licence. they need to enter login id and password to login to the website to generate request. If the login id or password is incorrect the user need to contact the RTO office using the site then the new password will be sent to their emails which is used by user to login and can be changed later. Applying for driver's licence also requires the user to enter the type of vehicle and their request will be generated.

- Vehicle registration:

Function: Vehicle is registered in the RTO's database.

Priority: Top

Requirements: The user can register the vehicle using their login id and password they need to enter vehicle type, maker, and model and they can easily generate a request for the vehicle registration.

- Verification and Test:

Function: User receives a message with the details.

Priority: High (Required for processing request)

Requirements: When the user has to give test for the licence they get a message containing the details of location of RTO office, officer assigned, their request id, date and time of test. Also for verification user gets the message with same details and they have to bring their vehicle for verification.

- Complaining:

Function: Complaint is registered.

Priority: High

Requirements: When a user wants to complaint for any issue they don't need any id or password for login as every complaint is registered only using aadhar. Anyone is allowed to file a complaint as it helps in improving the system.

- Adding new officer:

Function: officers are added as new ones join.

Priority: Medium (Second release, if possible, mandatory for third release)

Requirements: Admin needs to get the details of the officer via website which is further verified and then the new officer is added to the database by the admin. Every officer gets their user id and login details through mail after being verified and added to system.

- Removing a officer

Function: officer is removed as they retire or get transferred

Priority: High

Requirements: Admin need to remove the officer from the database as soon as they retire or get transferred to a location which does not comes under amin region. Because if they are not removed they might get assigned to somone. It can happen in two either admin gets the information of the officers transfer or retirement or else the officer informs admin about the same, so the admin could perform the tasks required for removing the officer.

2.1.2 Non-Functional Requirement

Performance Requirements:

- The program must be able to be run concurrently by multiple users. During peak times of usage.
- Transmission of user's data and request should be under 5 seconds.
- Queries upon the database shall be performed in less than 5 seconds.

- The program shall support taking data from 100 users simultaneously, performance must still confirm to all performance requirements.

Safety Requirements:

System would be protected by a password which would be with the admin. As it is connected via LAN and MAN an antivirus has been installed on system for its safety purpose

Software Quality Attributes:

- Availability: The system shall be available to all officers and users through Information Systems during their normal hours of operation.
- Reliability: Due to the use of a wireless network, reliability of the system always is not guaranteed. However, overall reliability of the system and information shall be achieved through the process of database manipulation.
- Reusability: The system shall be able to be reused for each new year.
- Robustness: If no network connection can be established to receive the information, the officer as well as user shall be allowed to enter roll later.
- Updatability: The system shall allow for addition or deletion of officers, users and offices while incorporating new session.
- Usability: Usability of the system shall be achieved through an online help page.

Security Requirements:

- An officer shall be permitted only view and edit information of the user assigned to them.
- To take perform any task the officers and users are required to insert a password to perform it. So the security of the system is maintained and also same for the admin, if anyone other than admin gets the admin access to the system they can modify the database easily.

2.2 System Feasibility

2.2.1 Operational

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken are all self-explanatory even to a layman. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

2.2.2 Technical

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described, and check if everything was possible using different type of front end and backend platforms.

2.2.3 Economical

This is a very important aspects to be considered while developing a project. We decided the technology based on minimum possible cost factor.

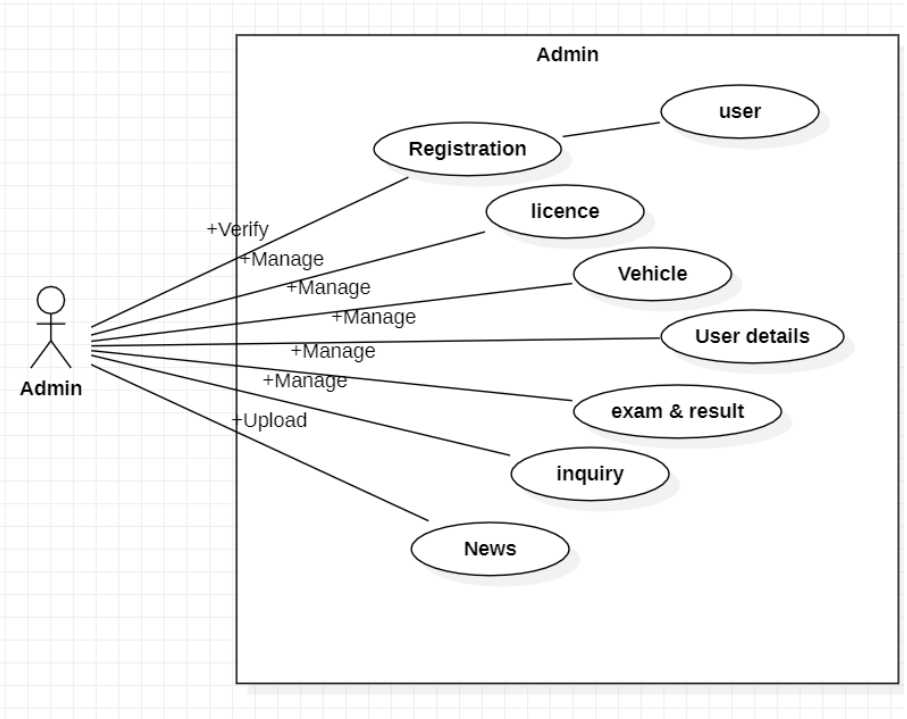
- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial cost and the later on running cost for system.

SYSTEM DESIGN

3. SYSTEM DESIGN

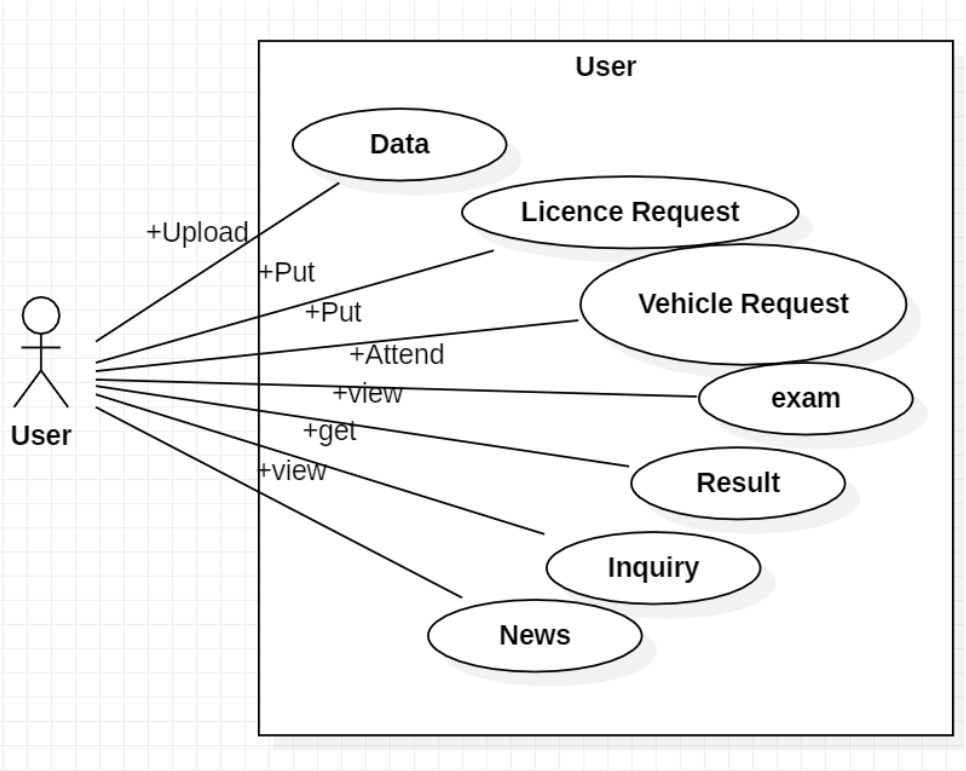
System Design is the process of defining elements of the system like module, architecture, components, and their interface and data for a system bases on specific requirements.

3.1Use Case Diagram



Use Case Diagram For Admin

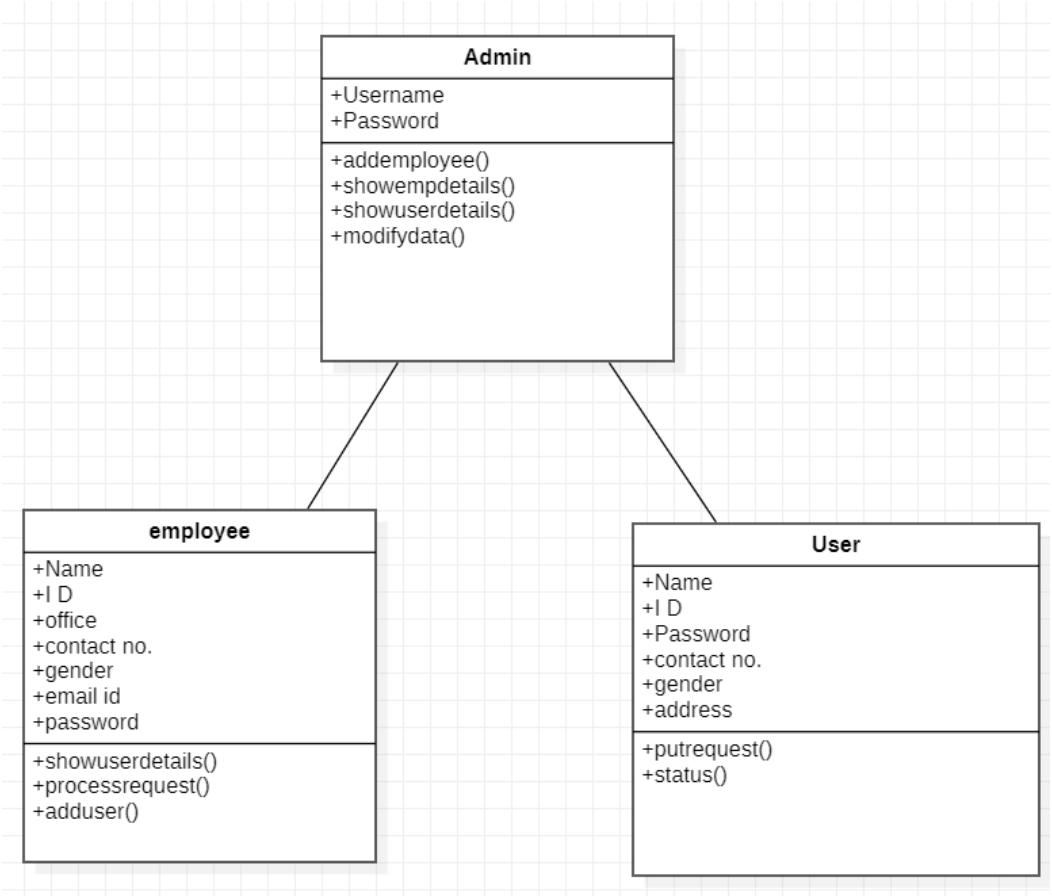
Fig. 1



Use Case Diagram For User

Fig. 2

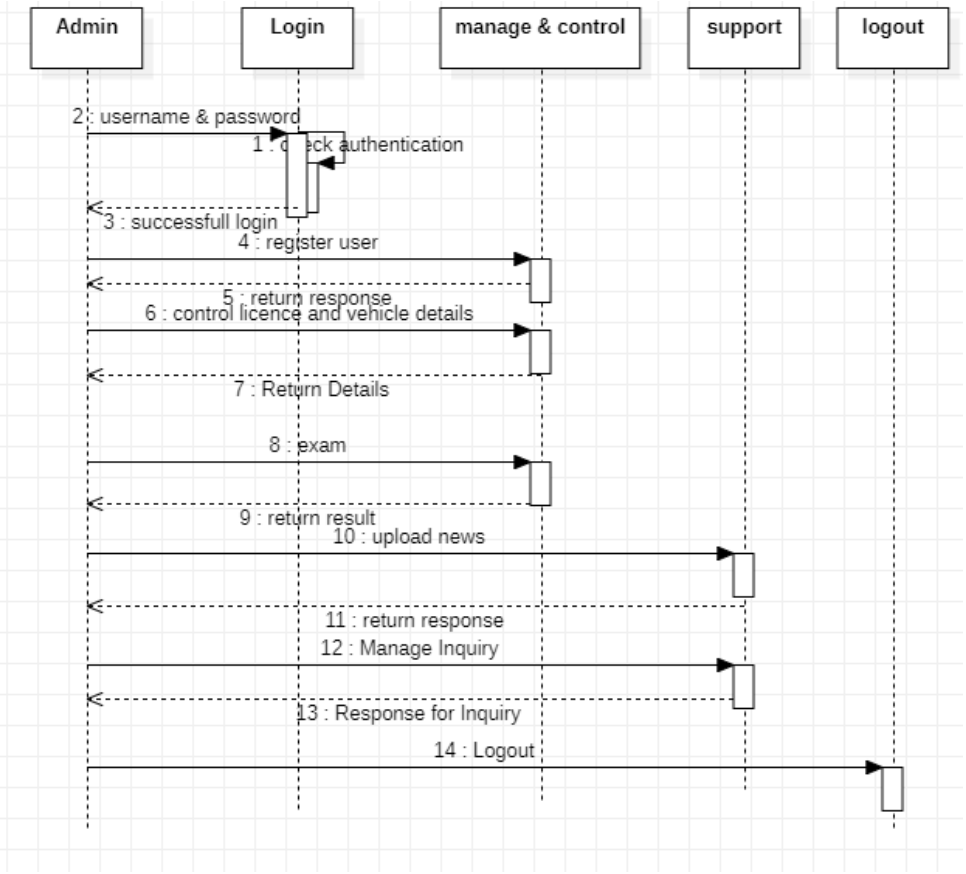
3.2 Class Diagram



Class Diagram for the system

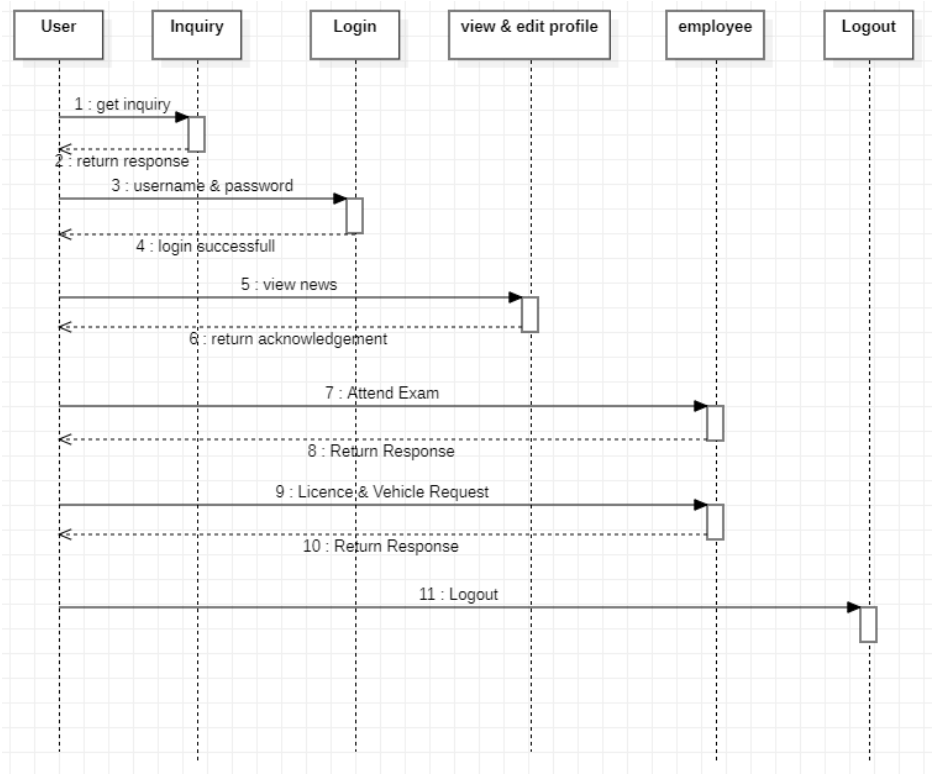
Fig. 3

3.3 Sequence Diagram



Sequence Diagram For Admin

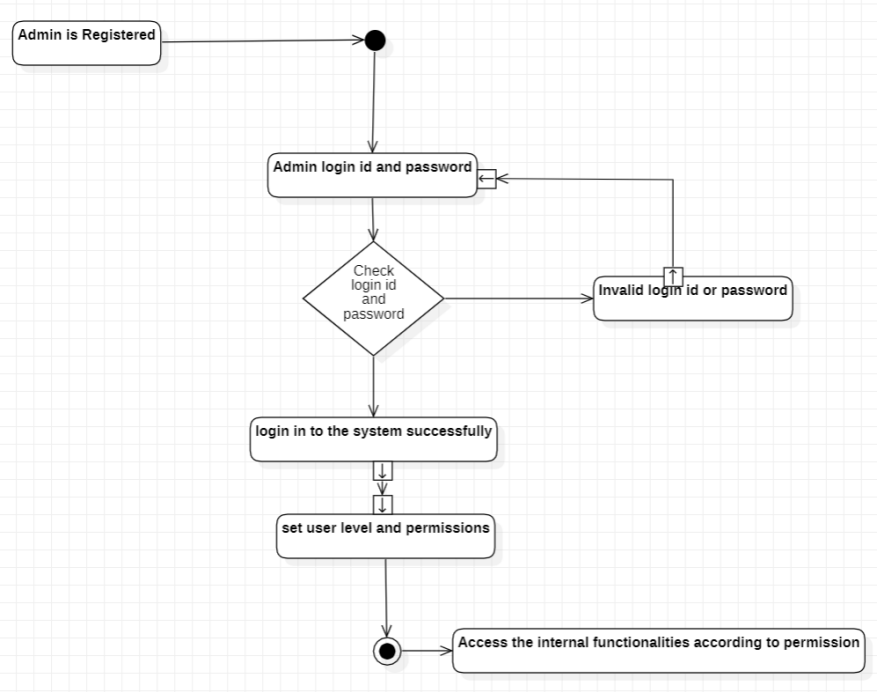
Fig. 4



Sequence Diagram For User

Fig. 5

3.4 Activity Diagram



Activity Diagram for the system

Fig. 6

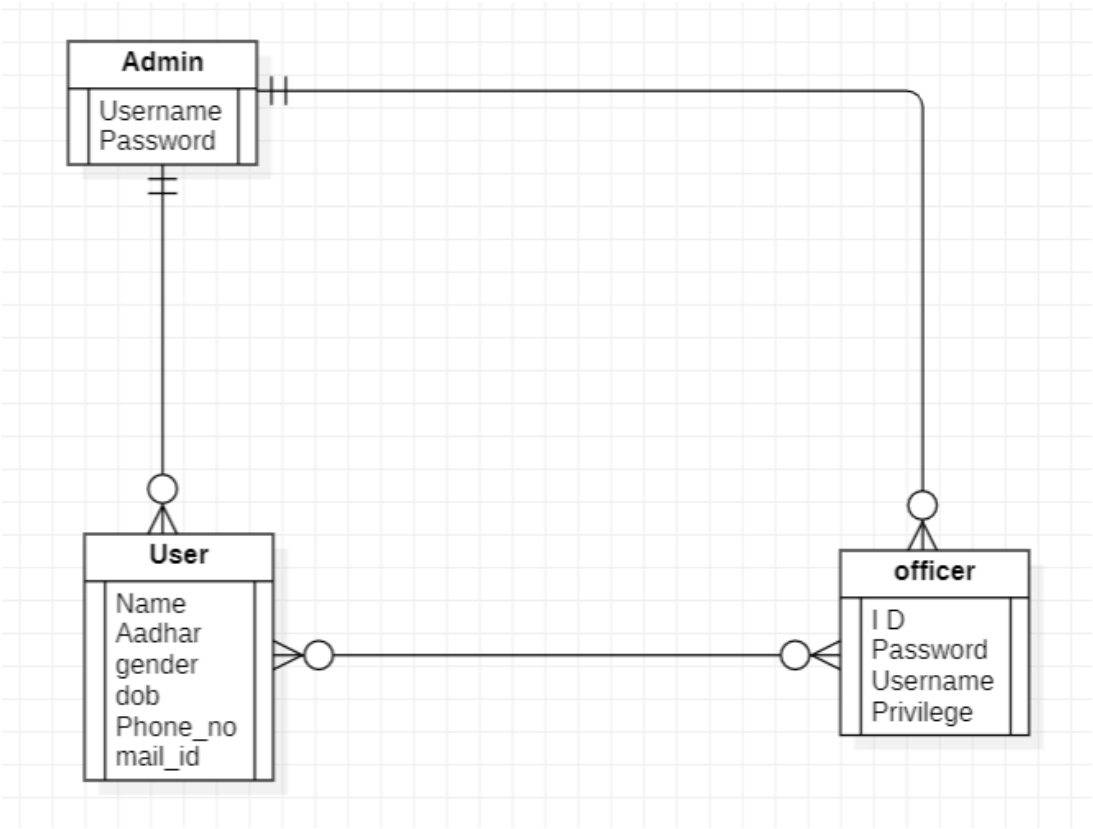
DESIGN

4. DESIGN

Software Design is a process to transform user requirement into some suitable form, which helps the programmer in software coding and implementation.

4.1 Data Design

4.1.1 E-R Diagram



Entity-Relationship Diagram for the system

Fig. 7

IMPLEMENTATION

5. IMPLEMENTATION

The software implementation stage involves the transformation of the software technical data package (TDP) into one or more fabricated, integrated, and tested software configuration items that are ready for software acceptance testing.

Software implementation is the process of integrating an application into an organization's workflow. The process typically begins with choosing a vendor and establishing a budget. The next steps may involve installing the application, migrating data, and testing various features.

5.1 Implementation of Modules

Home module: It contains few details about the website and has navigation bar with options.

LLR module: It is used for getting details for registering to get learner's licence.

Registration module: It is used for getting vehicle details from the user for registration of vehicle.

DL module: It is used for getting details of user when they want to register request for driver's licence.

Complaint module: It is used to register a complaint using aadhar number.

Gallery module: It contains few pictures of RTO events.

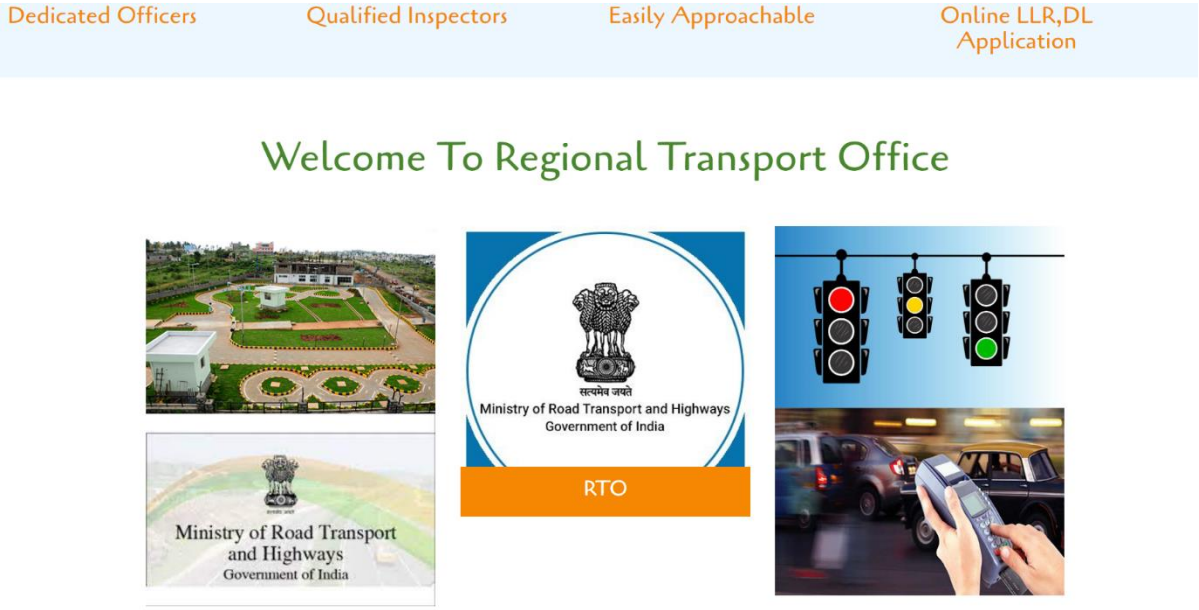
5.2 Results

During the implementation we have tested each function. Even though we tested the code that we wrote, a new implementation of a function might lead to that a previously written function would be affected and if we did not test all the various input alternatives we could think of, this might lead to a problem later. We have had a few problems like this but since we started to test and refactor our code early in this project the overall time spent fixing problems was reduced.

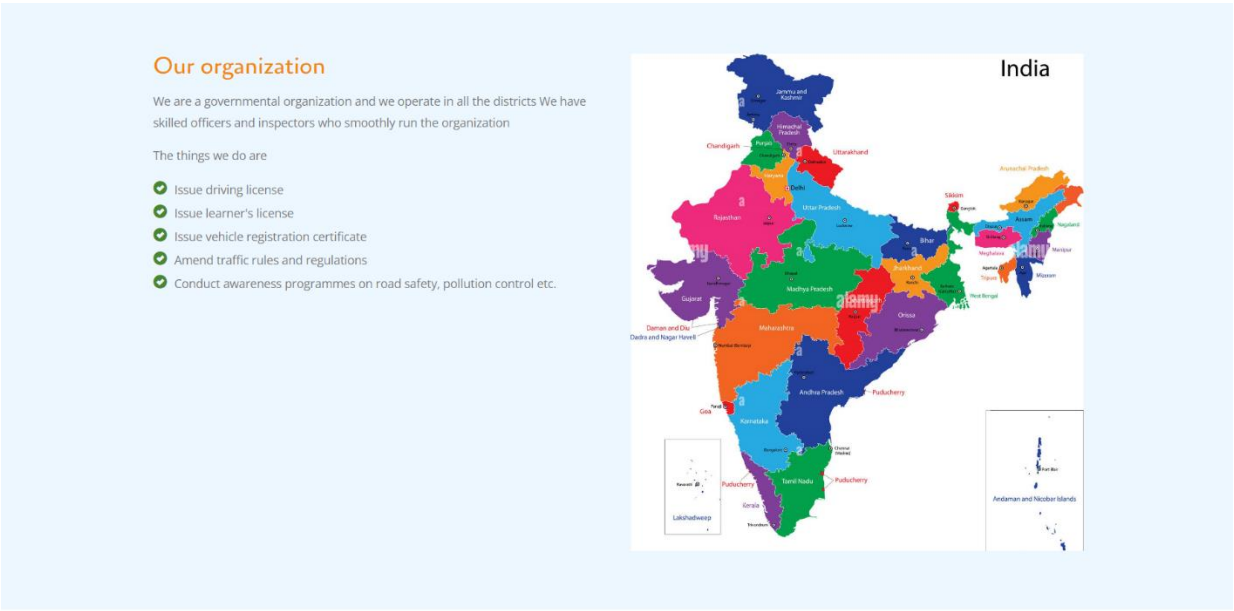
- **Home**

First Preview of Home Page with navigation bar

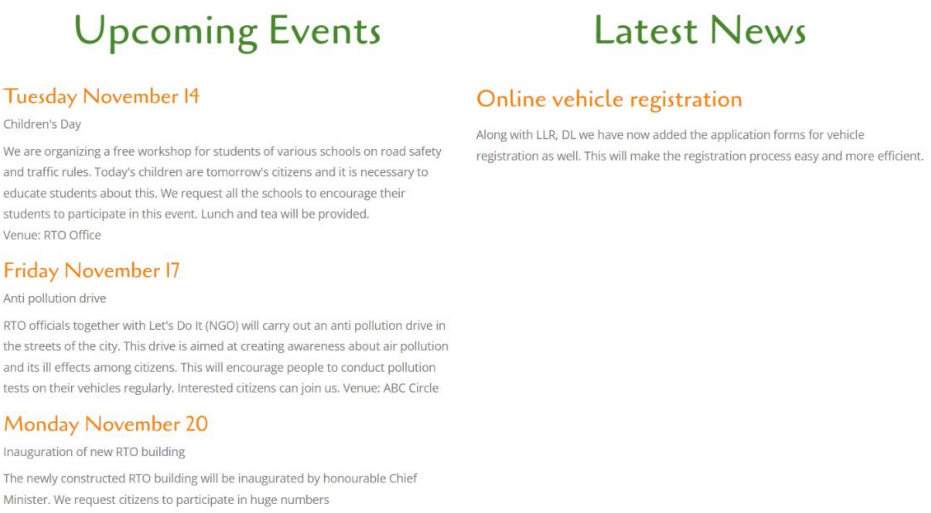
Fig. 8



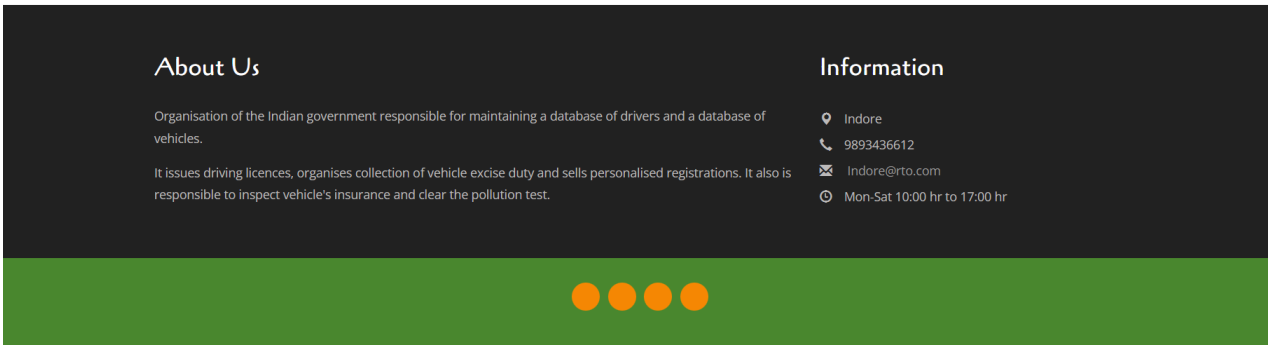
Welcome section on Home Page
Fig. 9



Organization Information section on Home Page
Fig. 10

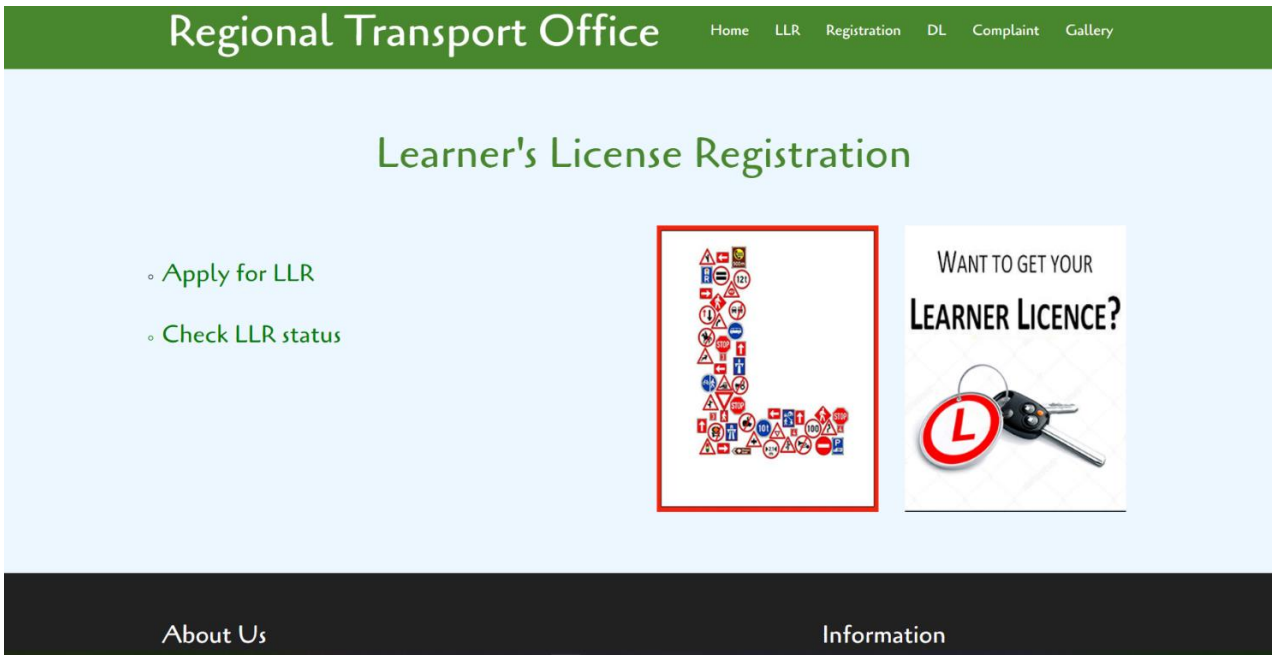


Event updates section on Home Page
Fig. 11

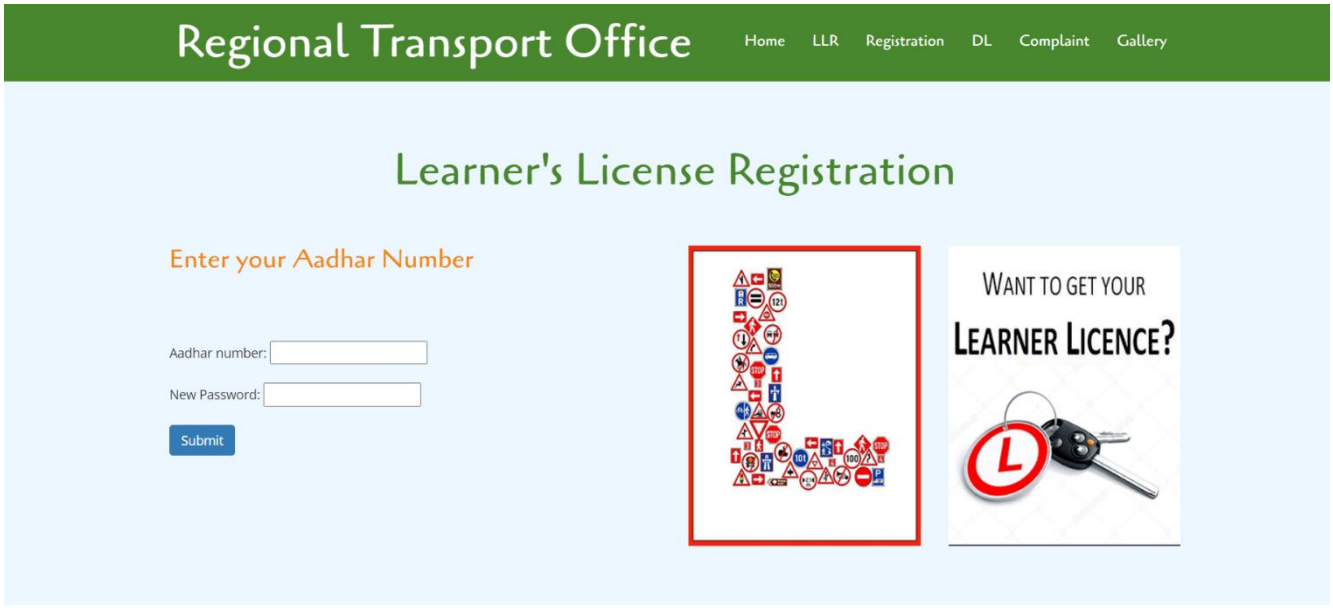


About Us section on Home Page
Fig. 12

- **Learner's Licence Registration**



Learner's Licence Registration Page
Fig. 13

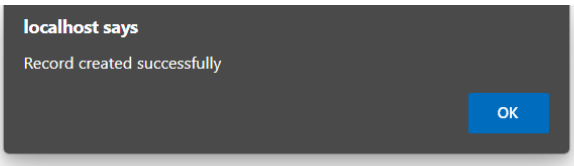


Learner’s License Registration Page asks for aadhar and password for Login

Fig. 14

LLR asks for category of vehicle for License Registration

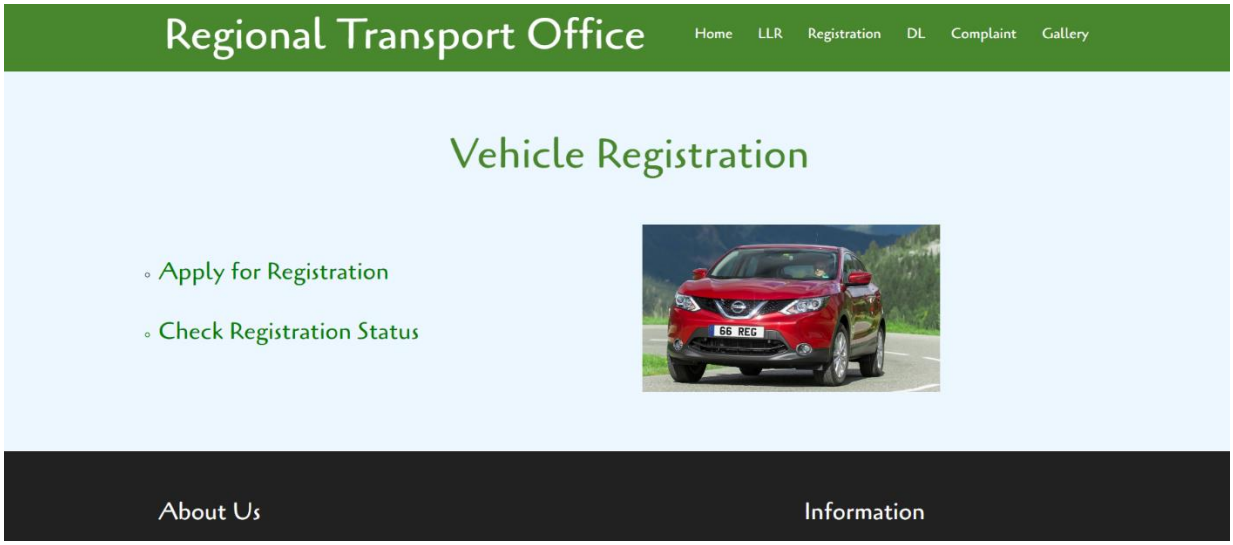
Fig. 15



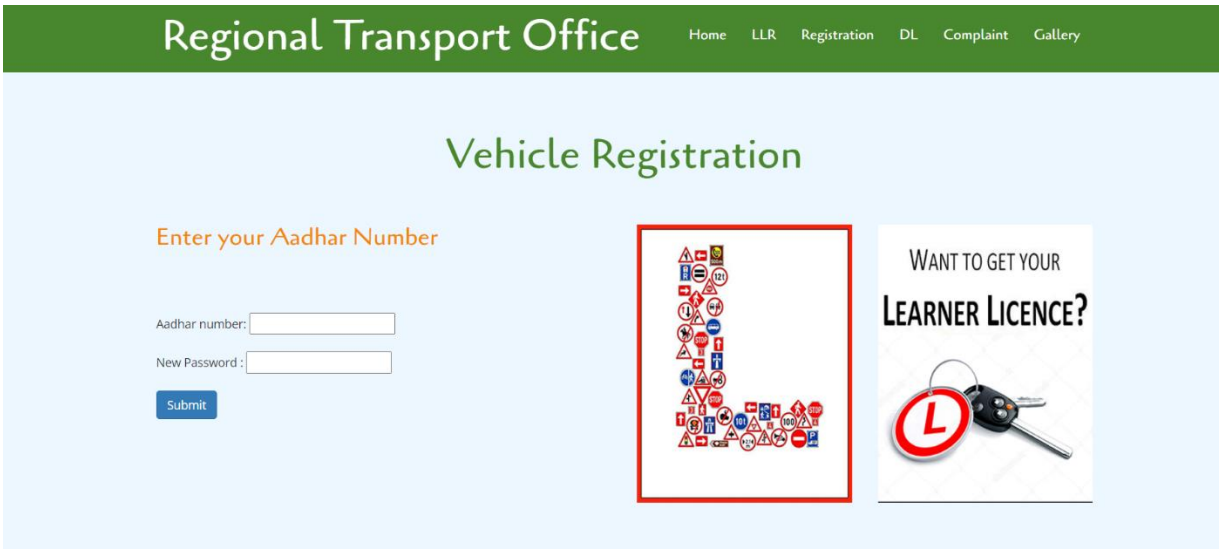
As category is choosen it shows “Record created successfully” message
Fig. 16

Exam Details are shown as soon as records are created
Fig. 17

- **Vehicle Registration**



Vehicle Registration Page
Fig.18

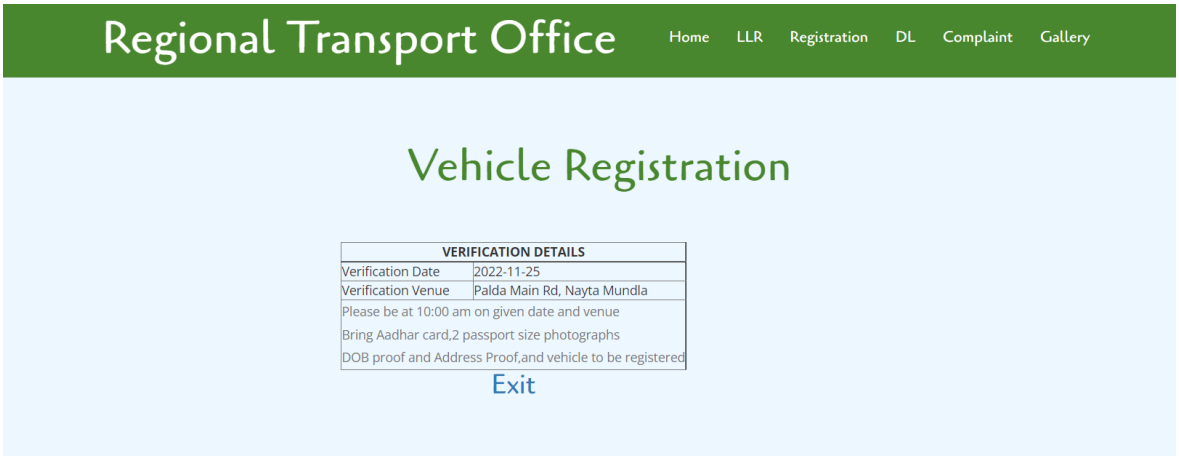


Vehicle Registration Page asks for aadhar and password for Login

Fig. 19

Registration page asks vehicle details to proceed with Registration

Fig.20



Registration page displays date and venue for Vehicle Verification

Fig. 21

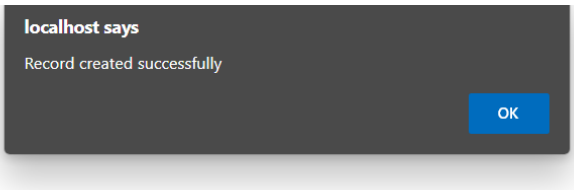
- **Driver's License**

Main page of Driver's licence
Fig. 22

Driver's license Page asks for aadhar and password for Login
Fig. 23

Driver's license page checks the learner's license Registration status and displays status if it is not approved else it asks for the vehicle type to Proceed

Fig. 24



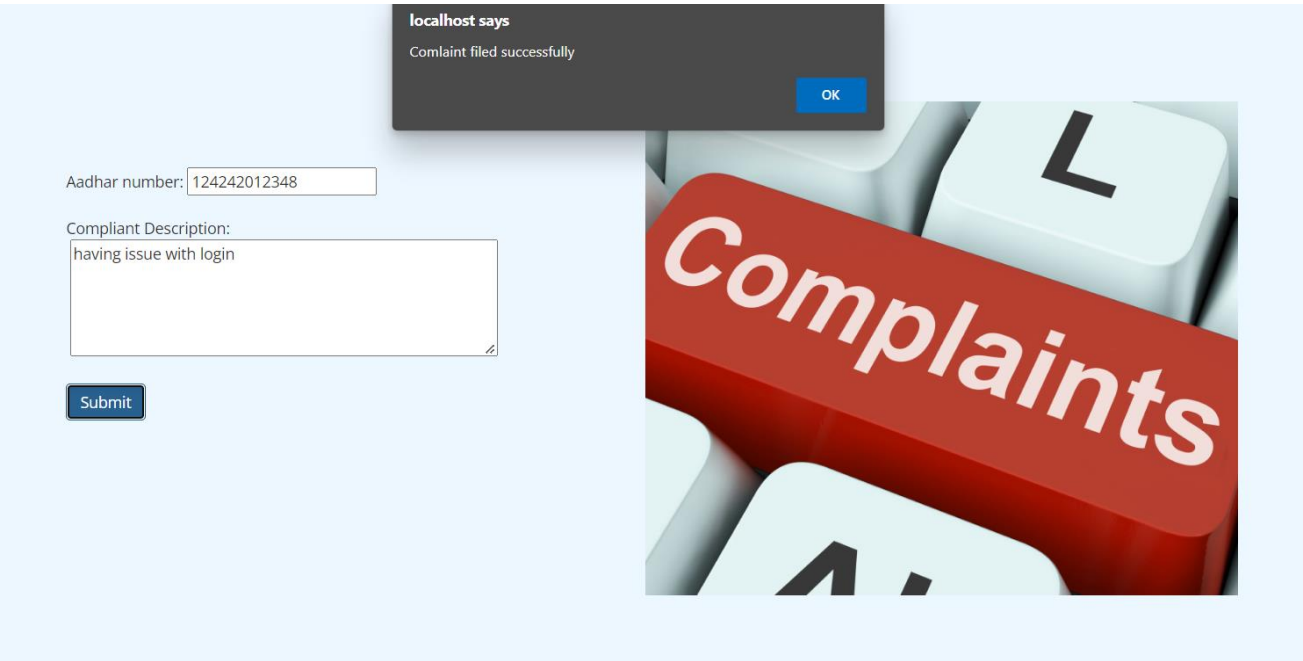
As category is choosen it shows "Record created successfully" message

Fig. 25

- **Complaint**

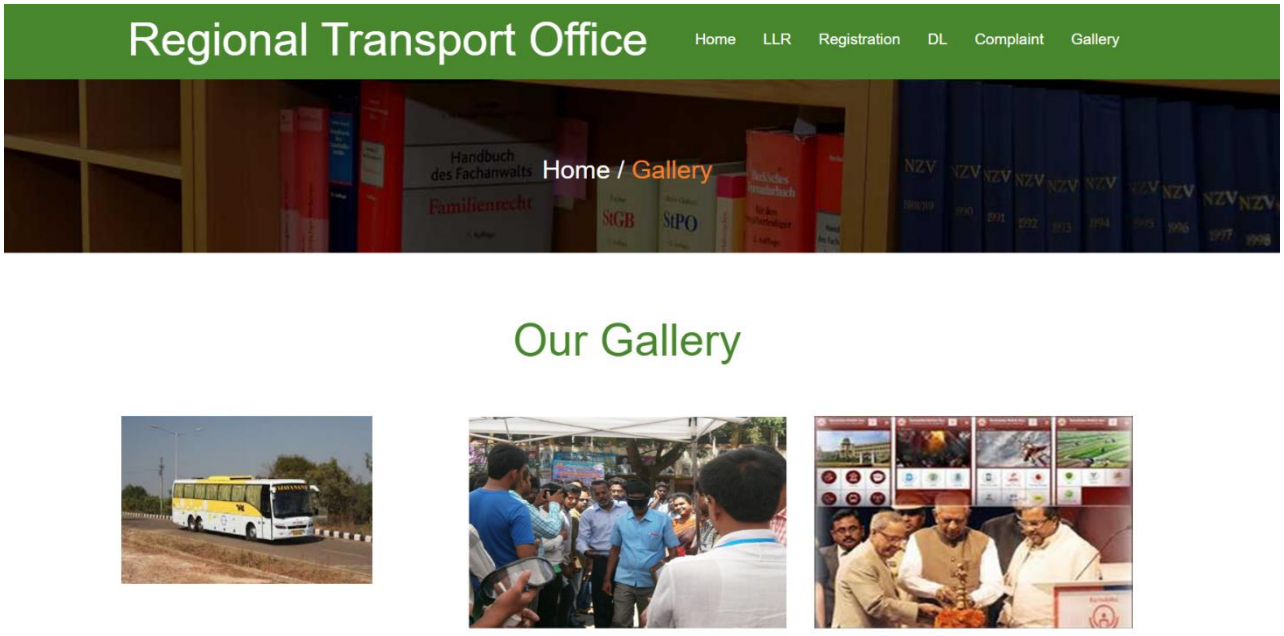
Complaint page asks for aadhar and Complaint Description for complaint Registration

Fig. 26



Once the aadhar and description is submitted, page shows “Complaint filed successfully” message
Fig. 27

- **Gallery**



Gallery page contains images of events with a small description
Fig. 28

TESTING

6. TESTING

6.1 Testing Objectives

The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

6.2 Testing Scope

Technically, Software Testing is an investigation conducted to provide stakeholders with information about the quality of a particular product or service under test. In other words, software testing is a process of verification and validation of a software. Testing scope is always good for development. A software or a product is good when we do the testing of the product or software just like our RTO management system. It contains the software components hardware, software, middleware to be tested, and also those that will not be tested of the banking software system. It gives the details of the objectives of the banking project. Also, it details user scenarios to be used in tests. If necessary, the scope can specify what scenarios or issues the banking project will not cover and what types of issues it can cover easily. A testing plan in software testing is the backbone on which the entire project is built. Without a sufficiently extensive and well-crafted plan, Q & A's are bound to get confused with vague, undefined goals and deadlines. This hinders fast and accurate testing unnecessarily, slowing down results, and delaying release cycles.

6.3 Testing Principles

1. Testing shows presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

- Testing shows the presence of defects:

Suppose the user is testing the e-learning software to identify problems so that the developer can fix them before they deploy the software to production environments. However, this process doesn't mean that there aren't any bugs in the product. It just means that there may be bugs, but you didn't find them because in the e-learning software while performing a specific operation there can be some minor bugs present in the software which can identify through the deeply testing of this software. There could be any number of reasons that you didn't uncover every bug, including the fact that the test cases didn't cover every scenario.

- Exhaustive testing is not possible:

The truth is that you cannot test everything, i.e., every combination of preconditions and inputs which is incorporated and used in this system software. And if you try to do so you'll waste time and money, but it won't affect the overall quality of this software. What you need to do is assess risk and plan your tests around these risks while using this system so you can be sure you're testing the key functions and necessary operations. Careful planning and assessment ensure your test coverage is good so you can have confidence in your final product before the deployment of the banking software and you don't even have to test every individual line of code of the software.

- Early Testing:

When it comes to the SDLC, early testing of the software is the key to identifying any defects in the requirements or design phase as soon as possible which is present in your software because it will be going to be much easier and less expensive to fix bugs in the early stages of testing of the software than at the end of the software lifecycle as then you might have to rewrite entire areas of functionality. And that likely means missed deadlines and cost overruns.

- Defect clustering:

In a project, a small number of modules can contain most of the defects. Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.

- Pesticide paradox:

Repeating the same test cases, again and again, will not find new bugs. So, it is necessary to review the test cases and add or update test cases to find new bugs. Likewise, if the banking software got continuously run under the same tests or operations, eventually it would fail into finding of the new defects, even though they'll probably confirm the e-Learning software is working properly.

- Testing is context-dependent:

Software testing is all about the context, which means that no one strategy will fit every scenario. The types of testing and the methods you use totally depend on the context of the systems or the software, e.g., the testing of an iOS application is different from the testing of an e-commerce website. Put simply, what you're testing will always affect the approach you use. Likewise, the e-Learning software testing is also going to be different than the software application because both of them have their own respective approaches and techniques of testing.

- Absence of errors fallacy:

If the e-learning software is 99% bug-free but it does not follow the user's requirement, then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customer requirements. As such, you should also test that software with the users. You can test against early prototypes at the usability testing phase so you can get feedback from the users that you can use to ensure the banking software is usable and they are getting less or absolute zero issues in using the banking system. Even though the e-Learning software might get few issues, doesn't mean it is ready to ship / deploy; it also has to meet the customer's requirements and expectations.

6.4 Testing Methods used

1.BLACK BOX TESTING - In the black-box approach, test cases are designed using only the functional specification of the software. That is, test cases are designed solely based on an analysis of the input/out behavior (that is, functional behavior) and does not require any knowledge of the internal structure of a program. For this reason, black-box testing is also known as functional testing. On the other hand, designing white-box test cases requires a thorough knowledge of the internal structure of a program, and therefore white-box testing is also called structural testing. Black- box test cases are designed solely based on the input-output behavior of a program. In contrast, white-box test cases are based on an analysis of the code. These two approaches to test case design are complementary. That is, a program has to be tested using the test cases designed by both the approaches, and one testing using one approach does not substitute testing using the other In black-box testing, test cases are designed from an examination of the input/output values only and no knowledge of design or code is required. The following are the two main approaches available to design black box test cases:

• Equivalence class partitioning

• Boundary value analysis The important steps in the black-box test suite design approach:

• Examine the input and output values of the program.

• Identify the equivalence classes.

• Design equivalence class test cases by picking one representative value from each equivalence class.

• Design the boundary value test cases as follows. Examine if any equivalence class is a range of values. Include the values at the boundaries of such equivalence classes in the test suite.

The strategy for black-box testing is intuitive and simple. For black-box testing, the most important step is the identification of the equivalence classes. Often, the identification of the equivalence classes is not straightforward. However, with little practice one would be able to identify all equivalence classes in the input data domain. Without practice, one may overlook many equivalence classes in the input data set. Once the equivalence classes are identified, the equivalence class and boundary value test cases can be selected almost mechanically.

2. WHITE BOX TESTING - White-box testing is an important type of unit testing. A large number of white-box testing strategies exist. Each testing strategy essentially designs test cases based on analysis of some aspect of source code and is based on some heuristic. We first discuss some basic concepts associated with white-box testing, and follow it up with a discussion on specific testing strategies.

A white-box testing strategy can either be coverage-based or fault-based.

Fault-based testing: A fault-based testing strategy targets to detect certain types of faults. These faults that a test strategy focuses on constitute the fault model of the strategy. An example of a fault-based strategy is mutation testing, which is discussed later in this section.

Coverage-based testing: A coverage-based testing strategy attempts to execute (or cover) certain elements of a program. Popular examples of coverage-based testing strategies are statement coverage, branch coverage, multiple condition coverage, and path coverage-based testing.

A white-box testing strategy can either be coverage-based or fault-based. Fault-based testing a fault-based testing strategy targets to detect certain types of faults These faults that a test strategy focuses on constitutes the fault model of the strategy. An example of a fault-based strategy is mutation testing, which is discussed later in this section. Coverage-based testing A coverage-based testing strategy attempts to execute (or cover) certain of a program. Popular examples of coverage-based testing strategies are statement coverage, branch coverage, multiple condition coverage, and path coverage-based testing elements

6.5 Test Cases

- Inspector login form

Test Cases-	Desired Output	Actual Output	Result
Correct username no password	Error	Please fill out this field	Pass
No email wrong password	Error	Please fill out this field	Pass

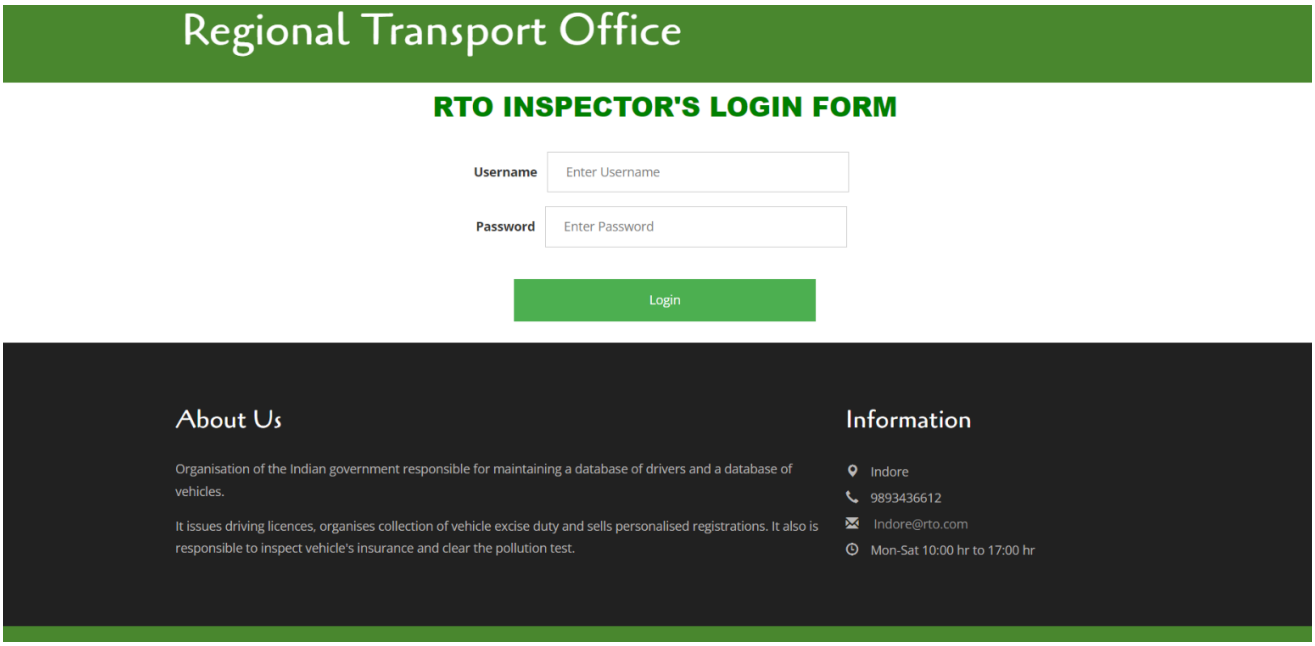
Correct username wrong password	Error	Invalid Credentials	Pass
No username no password	Error	Please fill out this field	Pass
Correct username and correct password	No Error	Login Successful	Pass

Table-I

6.6 Sample Test Data and Results

Inspector home page-

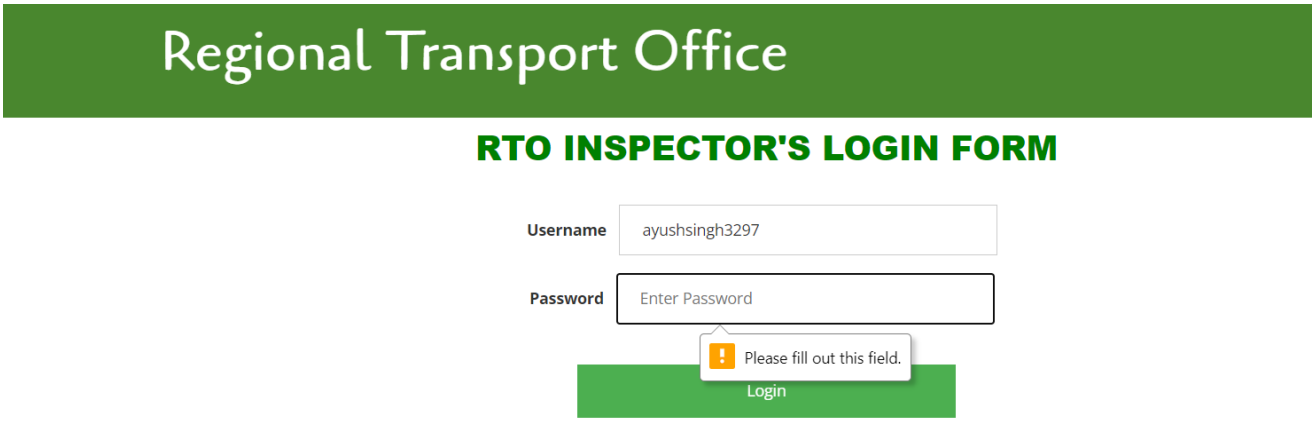
Before:



Inspector's Login Page, It takes Username and Password as Input

Fig. 29

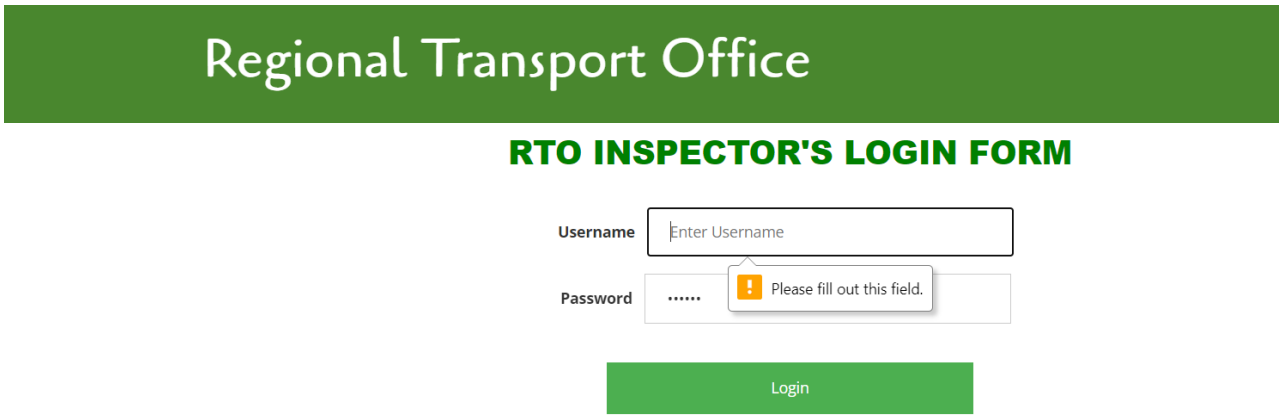
Correct username no Password:



when password is not entered it does not continues any further

Fig. 30

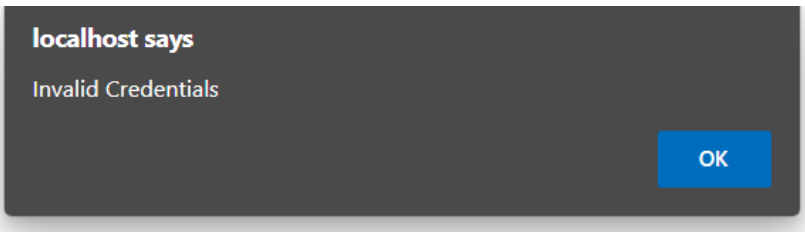
No email Wrong password:



When username is not entered it does not continues any further

Fig. 31

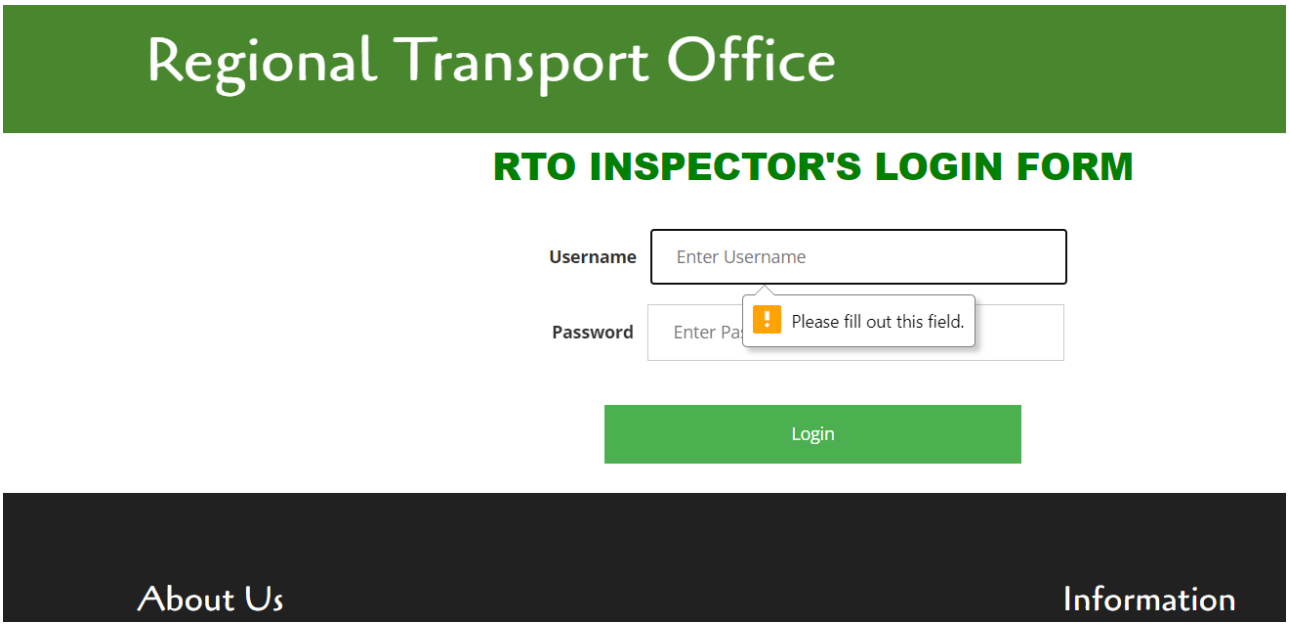
Correct username wrong Password:



When the password it wrong page shows a message “Invalid Credentials”

Fig.32

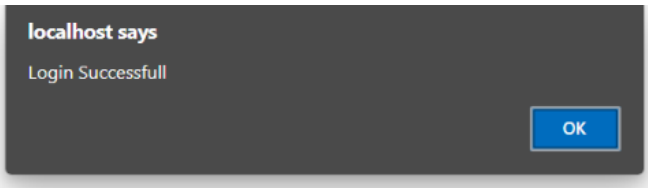
No username no password:



Without username and password the page does not continues further

Fig.33

Correct Username and Correct Password:



When the username and password both are correct page shows message “Login Successful” and proceeds

Fig.34

LIMITATIONS

7. LIMITATIONS

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out. Though the software presents a broad range of options to its users some intricate options could not be covered into it; partly because of logistic and partly due to lack of sophistication. Paucity of time was also major constraint, thus it was not possible to make the software fulproof and dynamic. Lack of time also compelled to ignore some part such as storing old result of the candidate etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at first instance. The user is provided help at each step for his convenience in working with the software.

The limitations of RTO management system are:

- Increased equipment expense

Organization must spend more money on devices and training of officials, but in the manual way of maintaining records there is not much expenses involved. All they require are registers and pens to maintain records.

- Difficult to maintain and repair

As it is a software, there is a need for proper maintenance. It is difficult to maintain and repair a software. Though it might be rare, once it is damaged, there would be expense for the sake of repairing.

- System is ineffective if there is no power supply

Software becomes useless if there is no power supply. The whole system works on power. But this is not the case with the manual way of RTO's working.

- Browser Dependency

The RTO Management System cannot work without a browser, it needs a browser to run and be accessed by the user, whereas the manual system does not depend on any kind of the system to run or for any other functionality.

Only thing that can be done is to keep in mind these limitations and manage it well for efficient management of the Record in the system.

FUTURE SCOPE

8. FUTURE SCOPE

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printer in the future.
- We can give more advance software for RTO Management System including more facilities.
- We will host the platform on online servers to make it accessible worldwide.
- Integrate multiple load balancers to distribute the loads of the system.
- Create the master and slave database structure to reduce the overload of the database queries.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of vehicle and customer. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the RTO management system. Enhancements can be done to maintain all the vehicle, customer, driving licence, registration, vehicle type.

CONCLUSION

9. CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a framework that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points:

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of purpose, scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the problem on which we are working in the project.
- We describe the requirement specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally the system is implemented and tested according to test cases.

BIBLIOGRAPHY AND REFRENCES

10. BIBLIOGRAPHY AND REFRENCES

https://www.tutorialspoint.com/software_engineering/software_project_management.htm

https://www.slideshare.net/rhspcte/software-engineering-ebook-roger-s-pressman

www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf

https://www.google.com