

**END - END VEHICLE MANAGEMENT**

**PROJECT REPORT**

**Submitted by Group J**

**JISHNU KADIYAKOL PARAMESWARAN (20001515)**

**CHARLES LIBIN KANNANKERIL GEORGE (10624280)**

**BASIL KATTUCHIRA ELDHOSE (20001500)**

**GOKUL SURESH (10638111)**

**ADVANCED PROGRAMMING TECHNIQUES**

**(B9CY100-2223-TMD2)**



# **ABSTRACT**

An end-to-end vehicle management system is a web application. For companies that own a fleet of vehicles available for use by their employees, the existing process is like submitting requests, and being assigned a vehicle manually is a time-consuming one. The proposed project is a single centralized framework that would integrate various disparate vehicle management processes such as scheduling, maintenance, sales, insurance, etc. The proposed web-based vehicle management system helps in streamlining and easing the process of managing company vehicles. The solution also provides the client with the means to monitor vehicle and driver performances. The solution required to enable the client to maintain a database of vehicles. Employees need to be provided with the means to submit and track vehicle requests online. Administration managers will be provided with the means to process such requests and assign vehicles. The facility to track vehicle insurance, fuel consumption, service, etc is also required.

- **Benefits**

- Automates the process of vehicle management thereby increasing productivity
- Vehicle insurance, service due, and fuel consumption can be tracked
- Allows for easy scheduling of vehicles

- **Modules**

- 1. Employee management module**

Employees of the company are registered by the super admin. The employees can be for maintenance and can be a driver.

- 2. Employee allocation module**

Vehicles can also be assigned to specific employees for their personal use. Such vehicles can be assigned with or without a driver and will not be available for booking.

### **3. Vehicle requisition module**

Employees can log in to the application using their network username and password. They can submit online vehicle requisitions (to their department manager) detailing the date/time when the vehicle is required, the reasons, and the approximate distance that is to be traveled. Upon approval from the manager, the request is made available to the administration department. Based on the vehicle and driver availability the administration manager can assign a vehicle and driver, from the available pool to the employee. An e-mail notification of the approval is sent to the employee.

### **4. Driver performance module**

Employees can provide feedback regarding their experiences with the driver. This feedback is made available to the administration manager, who in turn, can use this to judge driver performance.

### **5. Running status management**

Drivers can use login into the application and view their assigned jobs. For each job that they undertake the drivers are required to log the start and end mileage reading (this can then be used to determine the kilometers that the vehicle has run and when the next service is due). For the driver and vehicle to be available for new jobs, the previously assigned job is required to be flagged as completed. Based on the jobs assigned, managers can track driver schedules and monitor performances. A record of all accidents that a vehicle has been involved in can be recorded in the system. The accident log is maintained for both the vehicle and the driver assigned to the vehicle at the time of the accident.

### **6. Service module**

The administration manager can define parameters when a vehicle is due for service. The system will generate alerts and e-mail service due to reminders. A vehicle can be sent to any of the listed service stations and will be unavailable for booking till return from the service center. A detailed record for each service such as date sent, charges, work done, etc can be maintained. These service costs and fuel consumption histories allow a manager to monitor the running cost of each vehicle.

## **7. Insurance module**

Vehicle insurance can also be tracked within the system. For each vehicle, the name of the insurance company, the date of insurance and the insurance expiry, and a description of what cover includes can be entered. The system will generate notifications and e-mail alerts when an insurance policy is about to expire. Details of sold vehicles can also be maintained in the system.

# Contents

<b>Abstract</b>	<b>i</b>
<b>1 INTRODUCTION</b>	<b>2</b>
<b>2 SYSTEM CONFIGURATION</b>	<b>3</b>
2.1 SYSTEM REQUIREMENT SPECIFICATION (SRS) . . . . .	3
2.1.1 HARDWARE CONFIGURATION . . . . .	3
2.1.2 SOFTWARE CONFIGURATION . . . . .	3
<b>3 DEVELOPMENT TOOLS</b>	<b>4</b>
3.1 PYTHON . . . . .	4
3.2 MYSQL . . . . .	6
<b>4 SYSTEM ANALYSIS</b>	<b>8</b>
4.1 EXISTING SYSTEM . . . . .	8
4.2 Disadvantages Of Existing System . . . . .	9
4.3 PROPOSED SYSTEM . . . . .	9
4.4 Advantages Of Proposed System . . . . .	9
4.5 FEASIBILITY ANALYSIS . . . . .	9
4.5.1 TECHNICAL FEASIBILITY . . . . .	10
4.5.2 OPERATIONAL FEASIBILITY . . . . .	11
4.5.3 ECONOMIC FEASIBILITY . . . . .	11
4.5.4 TECHNICAL FEASIBILITY . . . . .	11
4.6 DATA FLOW DIAGRAM . . . . .	12
<b>5 SYSTEM DESIGN</b>	<b>16</b>
5.1 MODULE DESCRIPTION . . . . .	16
5.1.1 Employee management module . . . . .	16
5.1.2 Employee allocation module . . . . .	16
5.1.3 Vehicle requisition module . . . . .	16
5.1.4 Driver performance module . . . . .	17
5.1.5 Running status management . . . . .	17
5.1.6 Service module . . . . .	17
5.2 INPUT DESIGN . . . . .	18

5.3	OUTPUT DESIGN . . . . .	19
5.4	DATABASE DESIGN . . . . .	19
5.5	SCHEDULE . . . . .	25
<b>6</b>	<b>TESTING AND IMPLEMENTATION</b>	<b>26</b>
6.1	SYSTEM TESTING . . . . .	26
6.1.1	UNIT TESTING . . . . .	27
6.1.2	INTEGRATION TESTING . . . . .	27
6.1.3	VALIDATION TESTING . . . . .	28
6.1.4	USER ACCEPTANCE TESTING . . . . .	28
6.2	IMPLEMENTATION . . . . .	29
6.2.1	System Description . . . . .	29
6.3	SYSTEM IMPLEMENTATION . . . . .	30
6.4	SYSTEM MAINTENANCE . . . . .	31
<b>7</b>	<b>SCOPE AND FUTURE ENHANCEMENT</b>	<b>33</b>
7.1	SCOPE OF THE PROJECT . . . . .	33
7.2	FUTURE ENHANCEMENT . . . . .	33
<b>8</b>	<b>CONCLUSION</b>	<b>34</b>
<b>9</b>	<b>UI DESIGN</b>	<b>35</b>

# List of Figures

9.1	END-END VEHICLE MANAGEMENT SYSTEM HOME PAGE . . . . .	35
9.2	LOGIN PAGE . . . . .	36
9.3	ADMIN PAGE . . . . .	36
9.4	HR REGISTER PAGE . . . . .	37
9.5	DRIVER REGISTER PAGE . . . . .	37
9.6	ADDING VEHICLE CATEGORY PAGE . . . . .	38
9.7	ADDING VEHICLE NAME PAGE . . . . .	38
9.8	ADDING VEHICLE AMOUNT PAGE . . . . .	39
9.9	SERVICE DETAILS PAGE . . . . .	39
9.10	INSURANCE DETAILS PAGE . . . . .	40
9.11	REQUEST PAGE . . . . .	40
9.12	FEEDBACK PAGE . . . . .	41
9.13	HRs PAGE . . . . .	41
9.14	WORK ALLOCATION PAGE . . . . .	42
9.15	REQUEST FOR VEHICLE PAGE . . . . .	42
9.16	DRIVERS REQUEST PAGE . . . . .	43
9.17	PAYMENT PAGE . . . . .	43
9.18	FEEDBACK PAGE . . . . .	44
9.19	DRIVERS PAGE . . . . .	44
9.20	REQUEST PAGE . . . . .	45
9.21	WORKS PAGE . . . . .	45
9.22	INSURANCE PAGE . . . . .	46
9.23	PAYMENT PAGE . . . . .	46

# **INTRODUCTION**

This project is entitled “End-End Vehicle Management” an online web application, that ensures the safety of its users at any time. Social people can register in this application. It helps employees request vehicles for their personal use too. This system manages all the functions like vehicle management, allocation, registration, etc. It provides services such as daily vehicle allocation and personal allocation. The End-End vehicle management system is implemented in such a way with provides the client with the means to monitor vehicle and driver performances.

# **SYSTEM CONFIGURATION**

## **2.1 SYSTEM REQUIREMENT SPECIFICATION (SRS)**

The selection of hardware configuration is a very important task related to software development. Sufficient random access memory may affect adversely the speed and efficiency of the entire project.

Software configuration is the software environment in which the project was shaped. The programming language used, tools, etc are described here.

### **2.1.1 HARDWARE CONFIGURATION**

- **Processor** : intel i3/ryzen 3
- **RAM** : 2 GB
- **HDD** : 500 MB free space

### **2.1.2 SOFTWARE CONFIGURATION**

- **Operating System** : Windows 10/11
- **Front end** : PYTHON, HTML, CSS
- **Scripting Language** : PYTHON
- **Back end** : MYSQL
- **Browser** : CHROME
- **Web server** : XAMP SERVER

# DEVELOPMENT TOOLS

## 3.1 PYTHON

In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options. Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than in other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers. Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style, and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports an Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

### FEATURES OF PYTHON

- **Easy-to-learn** – Python has few keywords, a simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk library is very portable and cross-platform compatible with UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode that allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries, and Windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, a few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## **3.2 MYSQL**

A database is a collection of information that's related to a particular subject or purpose, such as tracking client orders or maintaining a list of project details. If the database isn't stored on a computer, or only part of it one may be tracking information from a variety of sources that one is having to coordinate and organize himself using MySQL Server, one can manage all information from a single database file.

Within the file, data is divided into separate storage containers called tables; view, add, and update data by using online forms; find and retrieve just the data wanted for reports. MySQL Server allows the user to view, update or analyze the database from the Internet of an intranet by creating data access pages. MySQL Server as a relational database stores data in many related tables.

A table is a collection of data about a specific topic such as projects or clients. Using a separate table for each topic means that, store that data only once. This makes the database more efficient and reduces data-entry errors. Tables organize data into columns (called fields) and rows (called records). A common field relates two tables so that MySQL Server can bring together the data from the two tables for viewing, editing, or printing. In the table Design view one can create an entire table from scratch or add, delete, or customize the fields in an existing table.

The user can also display records from tables that are related to the current table by displaying sub-datasheets within the main data sheet. With some restrictions, the user can work with the data in sub-datasheets in many of the same ways that they work with data in the main datasheet. To store data, create one table for each type of information that is to be tracked. To bring the data from multiple tables together in a query, form, report, or data MySQL Server page, define relationships between the tables

System design's main aim is to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design, all the major data structures, file formats, and major modules in the system and their specification are decided.

# **SYSTEM ANALYSIS**

System analysis is a general term that refers to an orderly structured process for identifying and solving problems. System analysis is concerned with investigating, analyzing, and evaluating an information system under consideration in the organization. It is the first stage of any project. The term system is derived from the Greek word “Systema” which means an organized relationship among functioning units or components. The system exists because it is designed to achieve one or more objectives.

Analysis can be defined as breaking up a whole so as to find out its nature, function, etc. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture, and an intellectual orientation. System analysis involves the study of an application area to fully understand the problem being posed. Activities are focused on developing a comprehensive knowledge of the existing system, its strength and weakness, and the reasons for the need to restructure, replace or automate the existing system.

## **4.1 EXISTING SYSTEM**

The study phase reveals that there is no proper website or application that administers all the vehicle management services. The needy man himself should find a solution for their personal use vehicle. There are a lot of websites around us, but there is no other website with all functions on the same platform.

## **4.2 Disadvantages Of Existing System**

- The existing system does not handle personal and daily allocation in the same platform.
- Time-consuming.
- Not enough responsible.
- It is not user Friendly

## **4.3 PROPOSED SYSTEM**

The proposed system is designed to eliminate the disadvantages of the existing system and should satisfy user needs. End-End vehicle management ensures users at any time and in any situation. This website is protected and it is so user-friendly. has. Automates the process of vehicle management thereby increasing productivity. Vehicle registration, service due and fuel consumption can be tracked. Allows for easy scheduling of vehicles

## **4.4 Advantages Of Proposed System**

- Easy to retrieve data available in the database.
- Quick implementation
- Very user friendly
- Does not require a large amount of memory
- Very cost-effective

## **4.5 FEASIBILITY ANALYSIS**

The feasibility of designing the system is determined by evaluating alternate methods of converting available data into the required outputs to fulfill the system objectives. Each of these alternate methods is termed a candidate system. The constraints unique to each

candidate are stored Candidate systems are evaluated by identifying factors that significantly affect system cost and performance by ranking each candidate in terms of these factors. Typical factors are development cost, operating cost, response time, development time, accuracy; reliability. It is a study to determine whether the proposed system is technically, economically, and behaviorally feasible in all respects. The objective of the feasibility study is not to solve the problem but to acquire a sense of its scope.

Feasibility analysis involves 8 steps:

1. Form a project team and appoint a project leader.
2. Prepare system flow charts.
3. Enumerate potential candidate system.
4. Describe and identify characteristics of candidate systems.
5. Determine and evaluate the performance and cost-effectiveness of each candidate system.
6. Weigh system performance and cost data.
7. Select the best candidate system.
8. Repair and report final project directive to management.

Three key considerations are involved in the feasibility analysis: economic, technical, operational, and behavioral.

#### **4.5.1 TECHNICAL FEASIBILITY**

Technical feasibility centers around the existing computer system and to what extent it can support the proposed addition. A study of function, performance, and constraint that may affect the ability to achieve an acceptable system is done.

In the proposed system data can be easily stored and managed using database management system software. The result of various queries can be generated easily. Therefore the system is technically feasible.

#### **4.5.2 OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned into information systems that will meet the organization's operating requirements. This test of feasibility asks if the system will work when it is developed and installed.

There was no difficulty in implementing the system and the proposed system is so effective, user-friendly, and functionally reliable that the user will find that the new system reduces their hard steps.

#### **4.5.3 ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. The cost-benefit analysis is done, a procedure to determine the benefits and savings those is expected from a candidate system and compare them with cost. The selected candidate system is economically feasible.

Proposed system is developed with the available resources. Since the cost input for the software is almost nil the output of the software is always a profit. Hence this proposed system, the number of employees to be evolved is reduced drastically. So, the proposed system is said to be feasible.

#### **4.5.4 TECHNICAL FEASIBILITY**

Technical feasibility centers around the existing computer system and to what extend it can support the proposed addition. A study of function, performance and constraint that may effect to the ability to achieve an acceptable system is done.

In the proposed system data can be easily stored and managed using database management system software. The result of various queries can be generated easily. Therefore the system is technically feasible.

## 4.6 DATA FLOW DIAGRAM

A Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by using a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows that go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labeled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. A candidate system, describes the input (source), outputs (destination), database (files), and procedures (data flow), all in a format that meets the user's requirements.

The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the result flows.

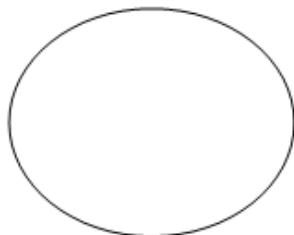
### Basic DFD Symbols



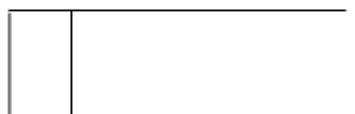
**Data source or Destination**



**A flow of data or data stream**

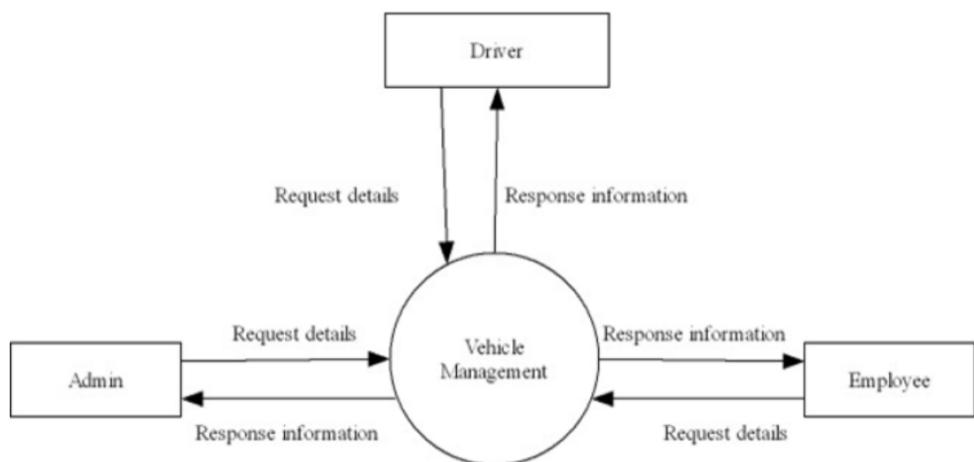


**Process that transforms data stream**

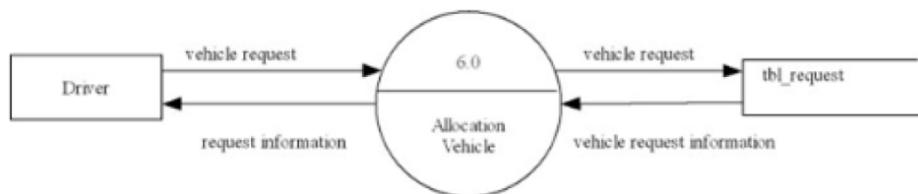
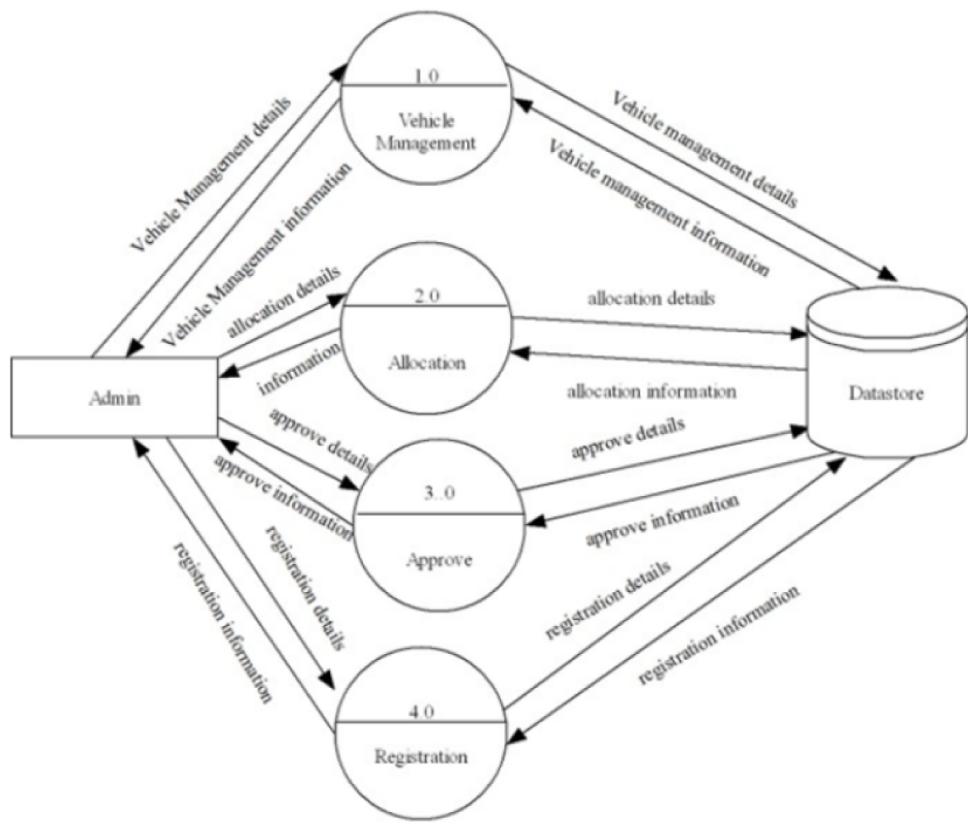


**Data store**

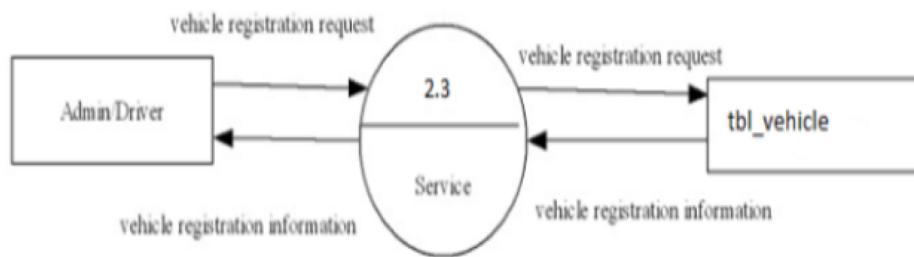
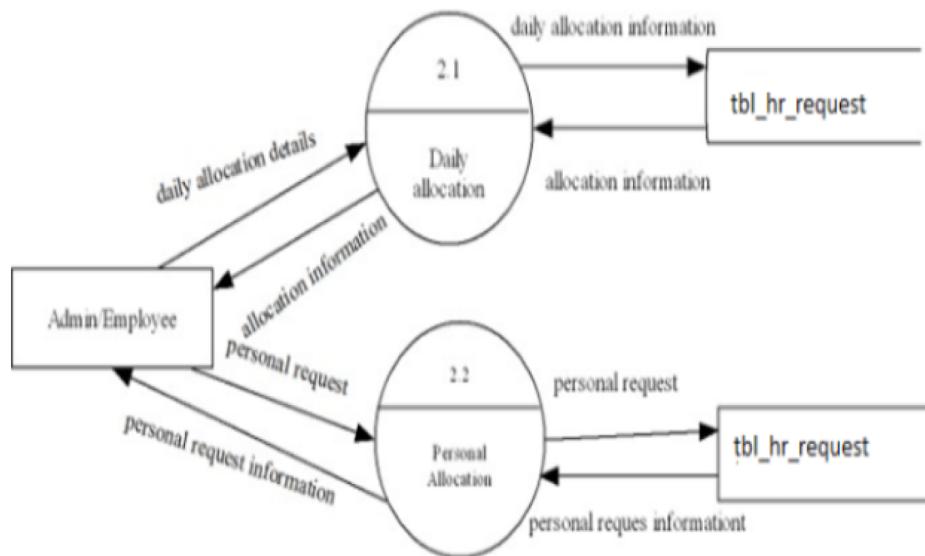
**Level:0**



## Level:1



## Level:2



# **SYSTEM DESIGN**

System design's main aim is to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design, all the major data structures, file formats, and major modules in the system and their specification are decided.

## **5.1 MODULE DESCRIPTION**

### **5.1.1 Employee management module**

Employees of the company are registered by the super admin. The employees can be for maintenance and can be a driver.

### **5.1.2 Employee allocation module**

Vehicles can also be assigned to specific employees for their personal use. Such vehicles can be assigned with or without a driver and will not be available for booking.

### **5.1.3 Vehicle requisition module**

Employees can log in to the application using their network username and password. They can submit online vehicle requisitions (to their department manager) detailing the date/time when the vehicle is required, the reasons, and the approximate distance that is to be traveled. Upon approval from the manager, the request is made available to the administration department. Based on the vehicle and driver availability the administration manager can assign a vehicle and driver, from the available pool to the employee. An

e-mail notification of the approval is sent to the employee.

### **5.1.4 Driver performance module**

Employees can provide feedback regarding their experiences with the driver. This feedback is made available to the administration manager, who in turn, can use this to judge driver performance.

### **5.1.5 Running status management**

Drivers can use login into the application and view their assigned jobs. For each job that they undertake the drivers are required to log the start and end mileage reading (this can then be used to determine the kilometers that the vehicle has run and when the next service is due). For the driver and vehicle to be available for new jobs, the previously assigned job is required to be flagged as completed. Based on the jobs assigned, managers can track driver schedules and monitor performances. A record of all accidents that a vehicle has been involved in can be recorded in the system. The accident log is maintained for both the vehicle and the driver assigned to the vehicle at the time of the accident.

### **5.1.6 Service module**

The administration manager can define parameters when a vehicle is due for service. The system will generate alerts and e-mail service due to reminders. A vehicle can be sent to any of the listed service stations and will be unavailable for booking till return from the service center. A detailed record for each service such as date sent, charges, work done, etc can be maintained. These service costs and fuel consumption histories allow a

manager to monitor the running cost of each vehicle.

## 5.2 INPUT DESIGN

Inaccurate input data is the most cause of errors in the data processing. Errors by data entry operators can be controlled by input validation. Input design is the process of converting user-oriented inputs to a computer-based format. Input design is part of an overall system design, which makes the system user-friendly. The input design is the link between the information system and users. It comprises developing specifications and those steps that are necessary to put transaction data into a usable form processing for data entry. The activity of putting data into the computer for processing can be achieved by instructing the computer to read data from a written or printed document or it can occur by having people key data directly into the system. The design of input focuses on controlling the number of inputs required, controlling errors, avoiding delay, and avoiding extra steps and the system analyst decides the following input design detail.

- What data to input.
- What medium to use.
- How the data should be arranged or coded.
- The dialogue guides the users in providing input.

Methods for performing input validations and steps to follow when errors occur. The input screen is displayed in a way that the user can understand entries very easily. Proper validation and control of input data will reduce the chance of errors. All major operations are done only after checking the results of confirmation messages. This makes the system more reliable and user-friendly.

## **5.3 OUTPUT DESIGN**

Efficient transport output design improves the system's relationship with the user and helps him to understand faster. A complicated output design contributes nothing to the system but leaves the user confused. For a successful system, it is a must that it maintains a clear output design. The users of the output its purpose and the sequence of details to be printed are all considered. The output form of is system is the justification for its existence. If the outputs are inadequate in a way the system itself is inadequate. The basic requirements of content medium and layout for its intended purpose. Hence it is necessary to design output in the form of reports. While designing the output, the following steps are to be considered:

- Determine what information to be presented to the user.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to the intended recipients.

## **5.4 DATABASE DESIGN**

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information systems and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at a low cost

- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data.

Usually, the centralized data managing software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has direct access to stores devices. This is the difference between logical and physical data.

### **TABLE DESIGN**

Table Name: login

Table Description: details of login

FIELD	DATA TYPES	CONSTRAINTS	DESCRIPTION
id	int(10)	Not null	id
Username	Varchar	Not null	Username
Password	Varchar(50)	Not null	Passwords
Utype	Varchar(50)	Not null	type of user

Table 5.1: TABLE DESIGN

Table Name: tbl\_amount

Table Description: details of vehicle amount

FIELD	DATA TYPES	CONSTRAINTS	DESCRIPTION
aid	int(10)	Not null	amount id
Vehicle name	Varchar	Not null	Vehicle name
amount	Varchar(50)	Not null	amount

Table Name: tbl\_card

Table Description: details of card

FIELD	DATA TYPES	CONSTRAINTS	DESCRIPTION
id	int(10)	Not null	id
cnum	int(26)	Not null	card number
exdate	date	Not null	expiry date
cvv	int(10)	Not null	cvv number

Table Name: tbl\_category

Table Description: category of vehicle

FIELD	DATA TYPES	CONSTRAINTS	DESCRIPTION
catid	int(10)	Not null	category id
catname	Varchar(50)	Not null	category name

Table Name: tbl\_driverreg

Table Description: driver register

FIELD	DATA TYPES	CONSTRAINTS	DESCRIPTION
id	int(10)	Not null	id
name	Varchar(50)	Not null	driver name
address	Varchar(110)	Not null	driver address
phone	int(50)	Not null	phone number
email	Varchar(50)	Not null	email
password	Varchar(50)	Not null	login password

Table Name: tbl\_driver\_request

Table Description: for requesting driver

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
drid	int(10)	Not null	driver id
vehid	int(10)	Not null	vehicle id
status	Varchar(50)	Not null	vehicle status

Table Name: `tbl_feedback`

Table Description : for feedback

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
drivername	varchar(50)	Not null	driver name
feedback	varchar(500)	Not null	feedback

Table Name: `tbl_hrreg`

Table Description: for HR Registration

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
name	Varchar(50)	Not null	driver name
address	Varchar(110)	Not null	driver address
phone	int(50)	Not null	phone number
email	Varchar(50)	Not null	email
password	Varchar(50)	Not null	login password

Table Name: `tbl_hr_request`

Table Description: for HR Request

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
hrid	int(10)	Not null	HR id
vehid	int(10)	Not null	Vehicle id
reqdate	date	Not null	Request date
retdate	date	Not null	Return date
status	int(50)	Not null	Status

Table Name: tbl\_insurance

Table Description: for Insurance Details

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
regno	varchar(50)	Not Null	Register Number
exdate	date	Not null	Expiry date

Table Name: tbl\_payment

Table Description: for Payment Details

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
driverid	int(10)	Not null	Driver id
paydate	date	Not null	Pay date
status	varchar(50)	Not Null	Status
tamnt	bigint(50)	Not null	Total Amount

Table Name: tbl\_service

Table Description: for Payment Details

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
regno	varchar(50)	Not Null	Register Number
servicedate	date	Not null	Service date

Table Name: tbl\_vehicle

Table Description: for Vehicle Details

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
vehid	int(10)	Not null	Vehicle id
catid	int(10)	Not null	category id
vehiclename	varchar(50)	Not Null	Vehicle Name
regno	varchar(50)	Not Null	Register Number

Table Name: tbl\_work

Table Description: for Work Details

<b>FIELD</b>	<b>DATA TYPES</b>	<b>CONSTRAINTS</b>	<b>DESCRIPTION</b>
id	int(10)	Not null	id
did	int(10)	Not null	Driver id
work	varchar(50)	Not Null	Work
wdate	date	Not null	Work date
estimatedkm	int(10)	Not null	Estimated Km
estimateddays	int(10)	Not null	Estimated Days
wrkload	varchar(50)	Not Null	Work Load
status	varchar(50)	Not Null	Status

## **5.5 SCHEDULE**

<b>Description</b>	<b>Duration</b>
System analysis	3 weeks
Requirement gathering	2 weeks
Analyze the modules of the project	2 weeks
Table design	1 weeks
Development	4 weeks
Document submission	23/03/2020

Table 5.2: SCHEULE

# **TESTING AND IMPLEMENTATION**

## **6.1 SYSTEM TESTING**

Testing is one of the major hurdles in the development of the system. A good test of course has a high probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. System test makes a logical assumption that if all parts of the system are correct, then the goal will be successfully achieved. A series of tests are performed before the system is already for user acceptance testing. During the testing, we tried to make sure that the product does exactly what is supposed to do.

Testing is the final verification and validation activity within the organization itself. The rules that are served as well as testing objectives are,

- Testing is a process of executing a program with the intent of finding errors.
- A successful test is one that uncovers an as-yet-undiscovered error.
- Tests should be planned long before testing begins.
- Exhaustive testing is not possible.

**TEST CASES** The objective of the system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified, and to ensure that the computer system and the associated clerical and other procedures work together. The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produced a schedule of expected results, runs the tests, and compares the computer produced with the expected results. The analyst may also be involved in procedure testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing.

Testing is the process of correcting a program with the intent of finding an error. Differ-

ent types of testing are,

- Unit Testing
- Integrated Testing
- Validation Testing
- User Acceptance Testing

### **6.1.1 UNIT TESTING**

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In this testing, we test each module individually and integrated the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during the programming stage itself.

### **6.1.2 INTEGRATION TESTING**

Integration testing is a logical extension of unit testing. In its simplest form, two units have already been combined into a component and the interface between them is tested. The component, in this sense, refers to an integrated aggregate of more than one unit. Integration testing identifies problem that occurs when units are combined. This method reduces the number of possibilities to a far simpler level of analysis.

### **6.1.3 VALIDATION TESTING**

At the culmination of integration testing, the software is completely assembled as a package. Interfacing errors have been uncovered and corrected and the final series of software validation testing begins.

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software function in a manner that is reasonably expected by the user. Software validation is achieved through a series of tests that demonstrate conformity with requirements. After the validation test has been conducted, one of two conditions exists:

- The function or performance characteristics conform to specifications and are accepted.
- A validation from specifications is uncovered and a deficiency is created.

Deviation or error discovered at this step in this project is corrected prior to completion of the project with the help of the user.

### **6.1.4 USER ACCEPTANCE TESTING**

Acceptance testing is a level of the software testing process where a system is for acceptability. User Acceptance Testing is the software testing process where the system is tested for acceptability and validation of the end-to-end business flow.

User acceptance testing is also known as Customer acceptance testing (CAT) if the system is being built or developed by an external supplier. The CAT or UAT is the final confirmation from the client before the system is ready for production. These tests are created by business customers and articulated in business domain language. It consists of test suites that involve multiple test cases and each test case contains input data as well as the expected output.

## **6.2 IMPLEMENTATION**

Implementation is the stage in the project where the theoretical is designed is turned into a working system and is giving confidence in the new system for the users which will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, and evaluation of changeover methods. Apart from planning major tasks of preparing the implementation are education and training of users.

### **6.2.1 System Description**

#### **Coding Standards**

Coding conventions are a set of guidelines for a specific programming language that recommended programming style, practices, and methods for each aspect of a piece program written in this language.

#### **Purpose**

Code conventions are important to programmers and web content developers for a number of reasons:

- They improve the readability of software artifacts
- They reduce training management and effort

#### **Major Coding Standards**

Major coding standards used in our project are;

- Naming convention

textbf{Naming Convention}

Applying naming conventions makes your web components elements easier to identify,

classify and coordinate in projects. In this section, we will look at these conventions specific to JSP technology.

## 6.3 SYSTEM IMPLEMENTATION

The implementation is the final stage and it is an important phase. It involves individual programming: system testing, user training, and the operational running of the developed proposed system that constitute the application subsystem. One major task of preparing for implementation is the education of users, which should relay have taken place much earlier in the project when they were being involved in the investigation and design work. During the implementation phase system actually takes physical shape. In order to develop a system implemented planning is very essential. The user tests the developed system and the changes are according to the needs.

Implementation is the stage of the project where the theoretical design is turned into a working system. It involves careful planning, investigation of the current system and its constraints on the implementation, design of method to achieve the changeover, training of staff in the changeover procedures, and evolution changeover method.

The implementation phase is an important one in which the source code is put into operation. Before implementing the software careful testing and documentation are necessary. During the implementation and testing phase configuration management and quality assurance of requirements, design specification, and source code are performed. Implementation should be provided with well-defined software requirements and architectural design specifications.

The implemented system has the following features:

- Reduced data redundancy
- Ease of use
- Controlled flow.

- Simplifies the management activities.

The following are the steps involved in the implementation plan:

- Test system with sample data
- Detection and correction of errors
- Make the necessary changes in the system
- Check the existing system.
- Installation of hardware and software utilities

Implementation includes all those activities that take place to convert from the old system to the new one. The process of putting the developed system into actual use is called system implementation. The system is implemented only after thorough checking is done and if it is found working according to the specifications.

## 6.4 SYSTEM MAINTENANCE

Maintenance is the enigma of software development. It holds the software industry captive, tying up programming resources. It is perceived as requiring neither skill nor experience. Maintenance covers a wide range of activities, including correcting, coding, and designing errors, updating documentation and test data, and upgrading user support. Maintenance is making adaptations of software for external changes (enhancements) and internal changes (fixing bugs). When made during the maintenance phase all preceding steps of the model must be revisited.

There are 3 types of maintenance,

- Corrective (fixing bugs/errors)
- Adaptive (update due to the environment changes)
- Perfective (enhancements, requirement changes)

### 1. CORRECTIVE MAINTENANCE

Corrective maintenance is the most commonly used maintenance approach, but it is easy

to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. The reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Therefore, corrective maintenance is carried out on all items where the consequences of failure or wearing out are not significant and the cost of this maintenance is not greater than preventive maintenance.

**2. PERFECTIVE MAINTENANCE** This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece of software tends to be subjected to a succession of changes resulting in an increase in its requirements. This is based on the premise that as the software becomes useful, the user experiment with new cases beyond the scope for which it was initially developed. Expansion in requirements can take the form of enhancement of existing system functionality and improvement in computational efficiency.

### **3. ADAPTIVE MAINTENANCE**

Adaptive maintenance includes any work initiated as a consequence of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of a software system. The environment in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will warrant a corresponding modification of the software.

# **SCOPE AND FUTURE ENHANCEMENT**

## **7.1 SCOPE OF THE PROJECT**

The Scope of the project is the system on which the software is installed. i.e. the project was developed as a web application, and it will work for any user who is in need of a vehicle. But later, the project can be modified to be more user-friendly.

## **7.2 FUTURE ENHANCEMENT**

The scope of the project is still expandable. The application can be enhanced in the future to the needs of the users. The database and the information can be updated to the latest coming versions.

# **CONCLUSION**

The project entitled with End-End vehicle management System is an online web application, that ensures that its users are at any time. The solution required to enable the client to maintain a database of vehicles. Employees need to be provided with the means to submit and track vehicle requests online. Administration managers will be provided with the means to process such requests and assign vehicles. The facility to track vehicle service, fuel consumption, service, etc is also required. End-End vehicle management ensures users at any time and in any situation. This website is protected and it is so user-friendly. has. Automates the process of vehicle management thereby increasing productivity. Vehicle registration, service due and fuel consumption can be tracked. Allows for easy scheduling of vehicles.

# UI DESIGN

Figure 9.1: END-END VEHICLE MANAGEMENT SYSTEM HOME PAGE

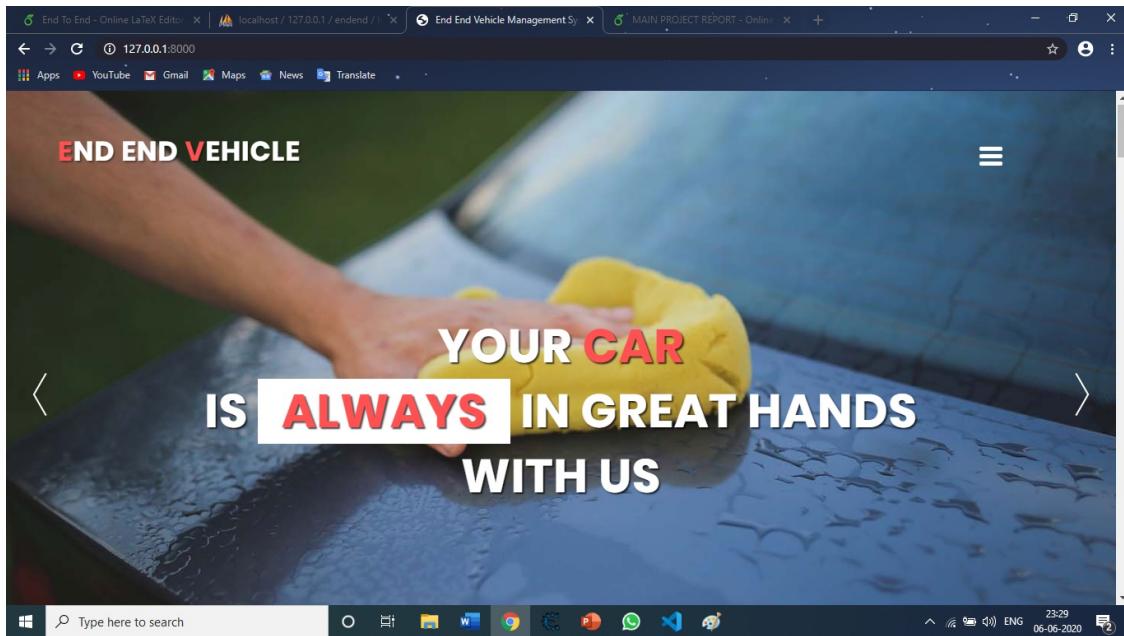


Figure 9.2: LOGIN PAGE

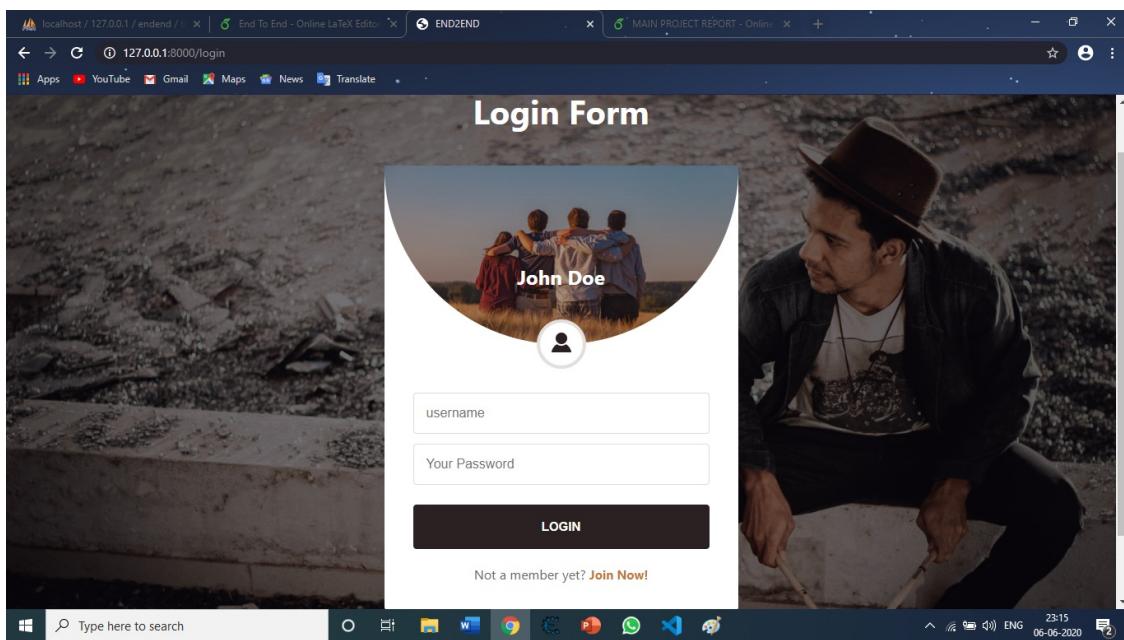


Figure 9.3: ADMIN PAGE

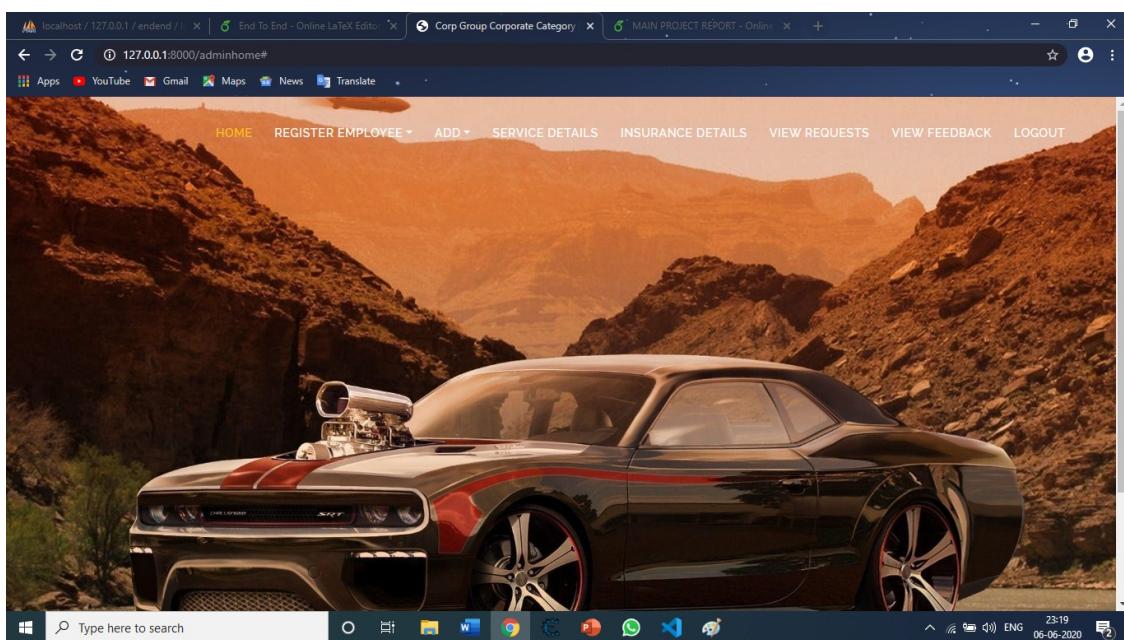


Figure 9.4: HR REGISTER PAGE

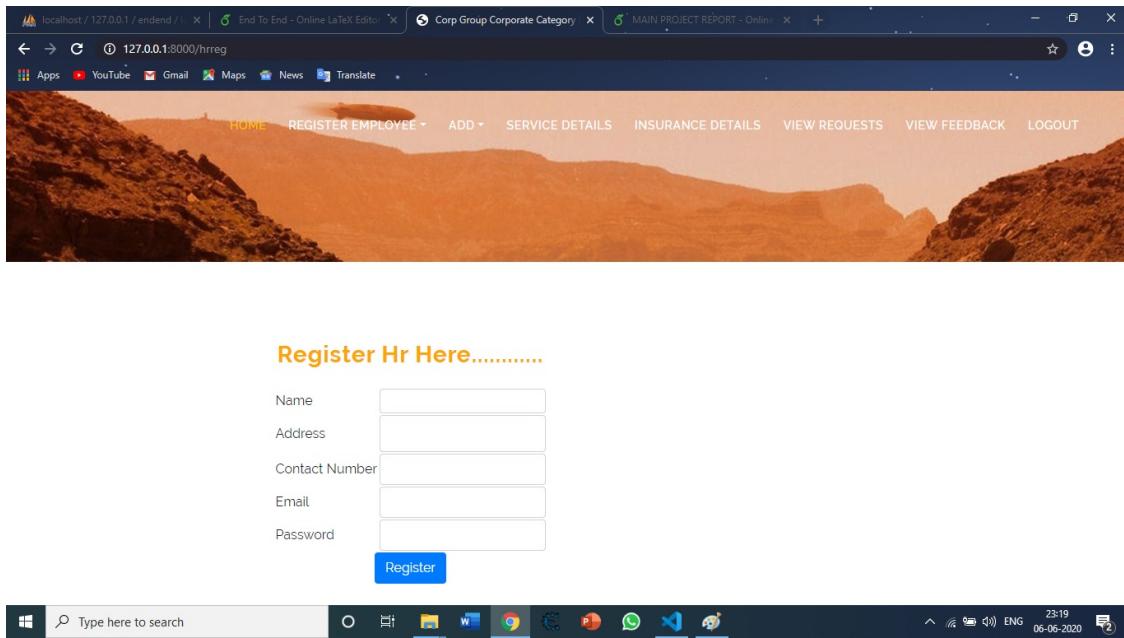


Figure 9.5: DRIVER REGISTER PAGE

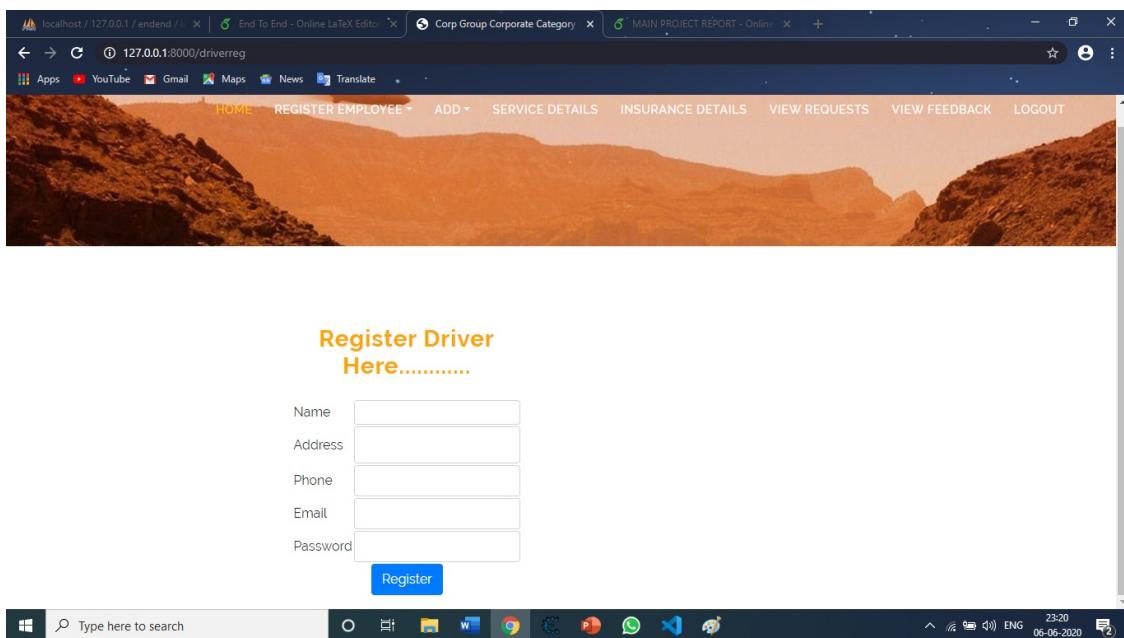


Figure 9.6: ADDING VEHICLE CATEGORY PAGE

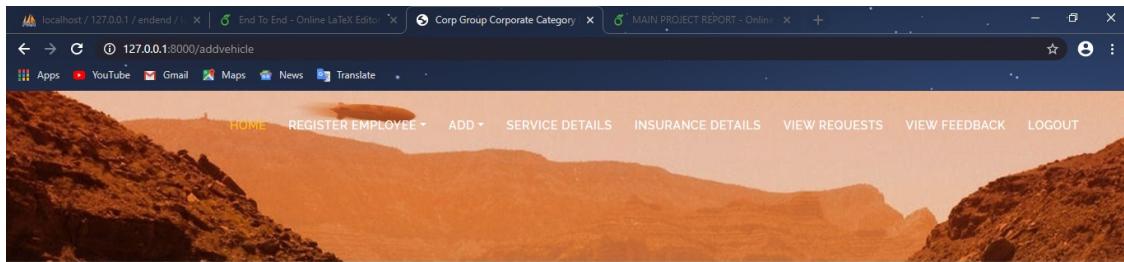


Figure 9.7: ADDING VEHICLE NAME PAGE

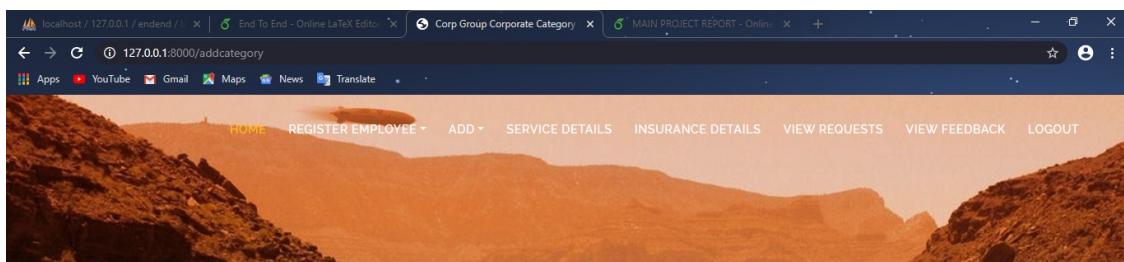
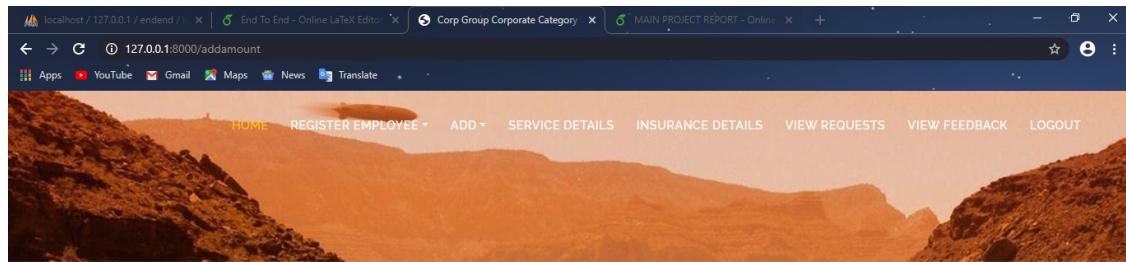


Figure 9.8: ADDING VEHICLE AMOUNT PAGE



### Add Amount for vehicle Here.....

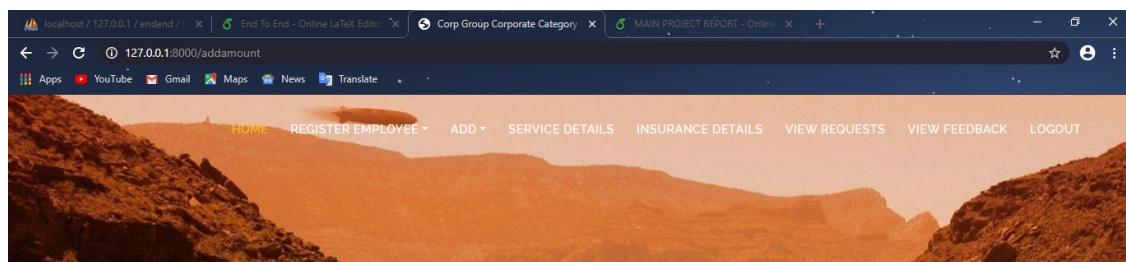
Select Vehicle

Vehicle Amount Per Day

**ADD**



Figure 9.9: SERVICE DETAILS PAGE



### Add Amount for vehicle Here.....

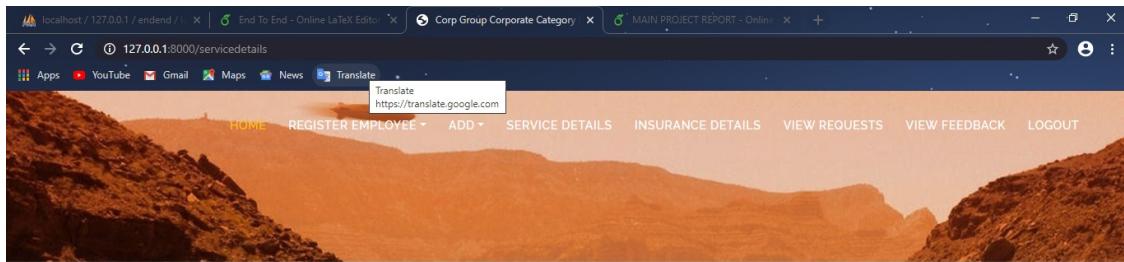
Select Vehicle

Vehicle Amount Per Day

**ADD**



Figure 9.10: INSURANCE DETAILS PAGE



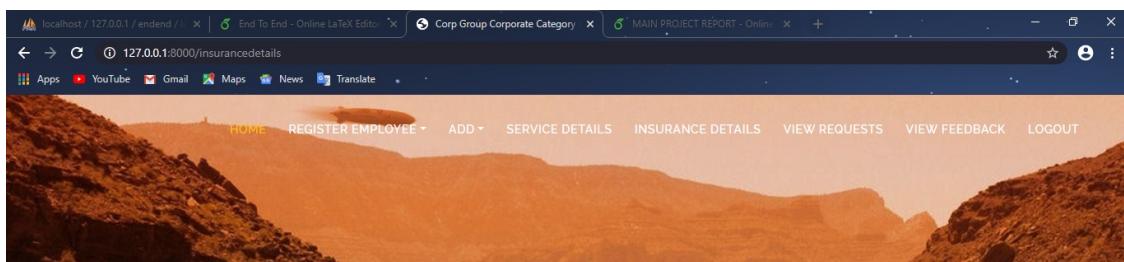
**servicedetails .....**

Select Register Number

Service Date



Figure 9.11: REQUEST PAGE



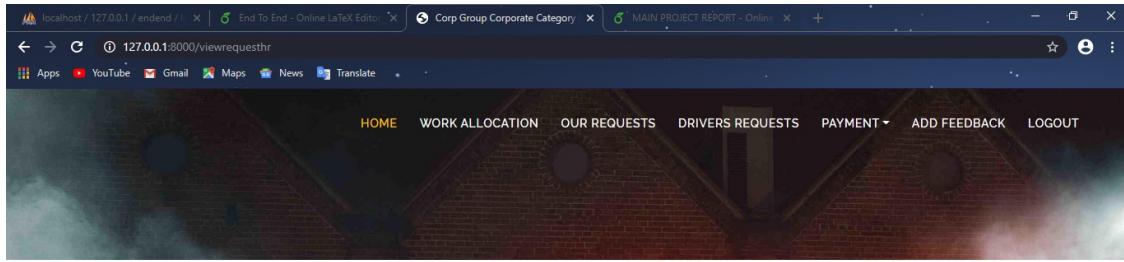
**Insurance Detailse Here.....**

Select Register Number

Expiry Date



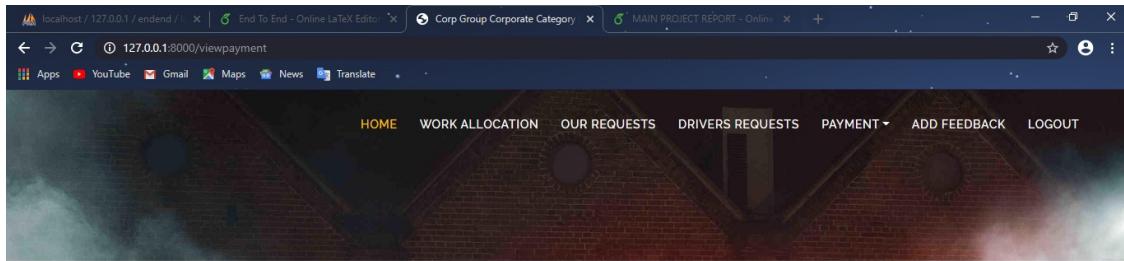
Figure 9.12: FEEDBACK PAGE



### Vehicle Requests.....

Username	Vehicle Name	Register Number	Category Name	Action
vishak@gmail.com	maruthy	kl72323	car	Accept

Figure 9.13: HRs PAGE



### Payments.....

Driver Username	Payment Date
manu	March 3, 2020



Figure 9.14: WORK ALLOCATION PAGE

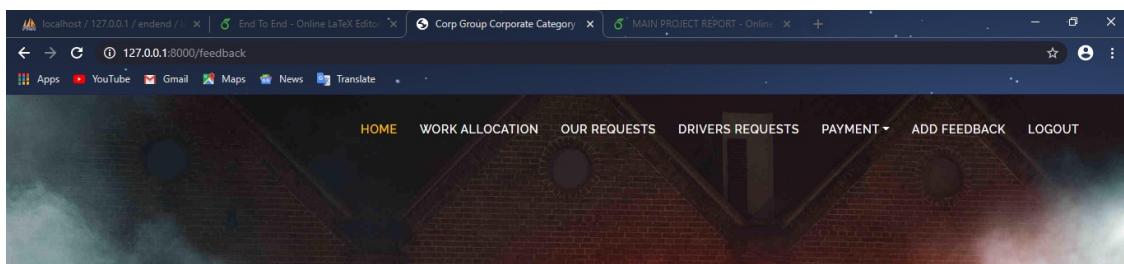


Figure 9.15: REQUEST FOR VEHICLE PAGE

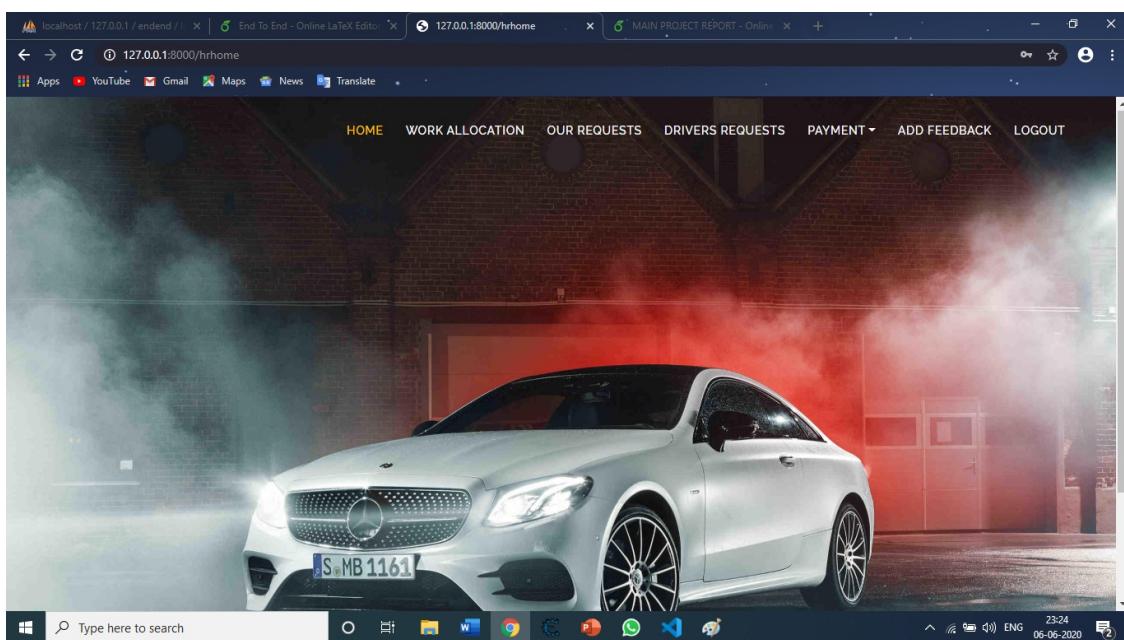


Figure 9.16: DRIVERS REQUEST PAGE

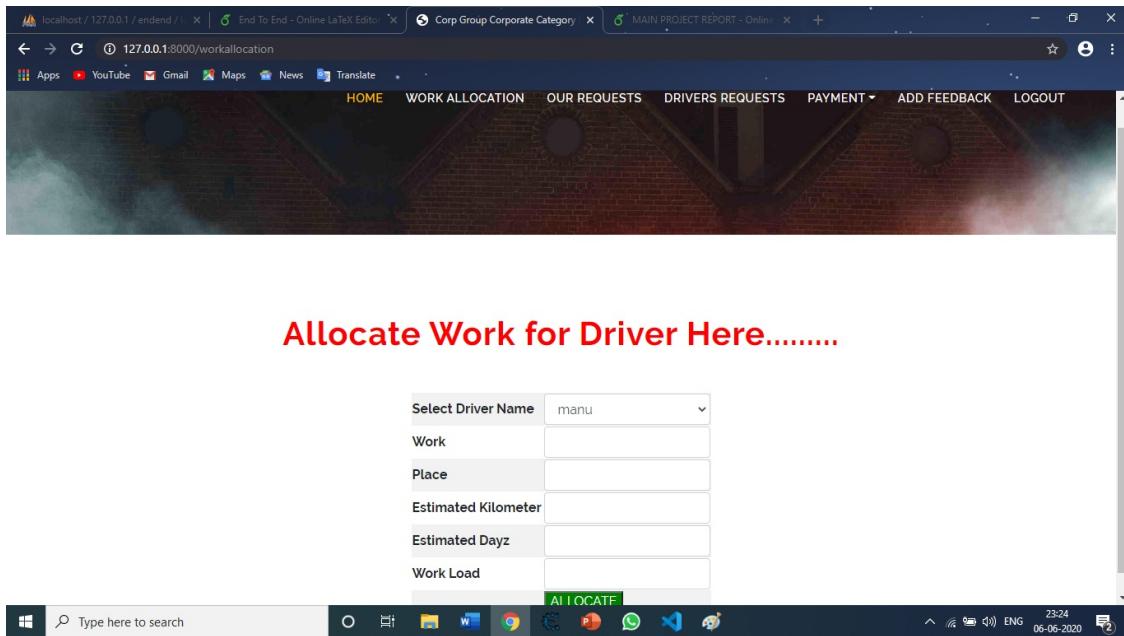
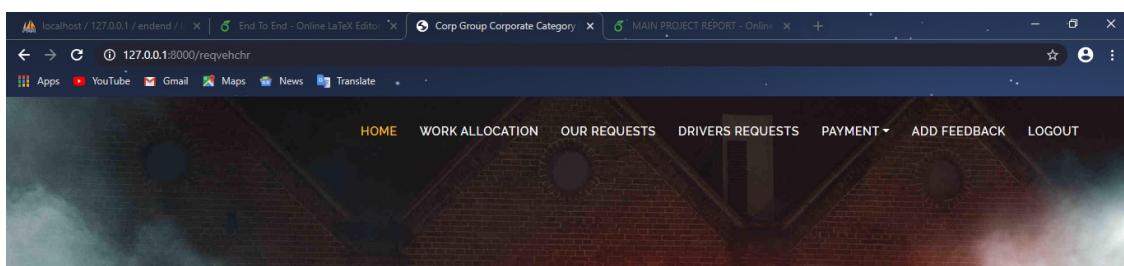


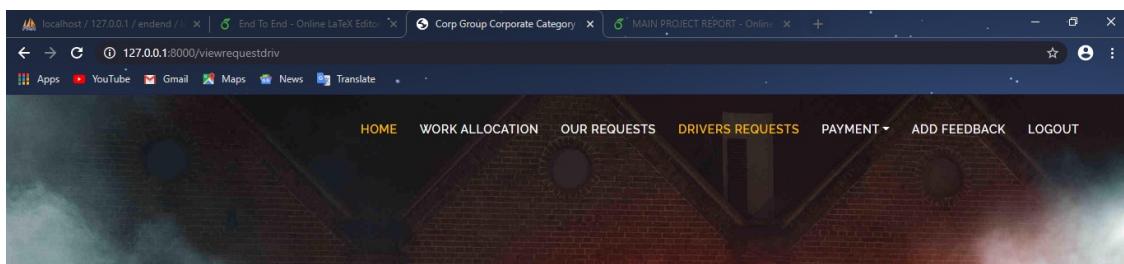
Figure 9.17: PAYMENT PAGE



Request For Vehicle.....

Vehicle ID	Vehicle Name	Register Number	Category Name	Action
1	maruthy	kl72323	car	Request
2	max	kl32d5050	jeep	Request

Figure 9.18: FEEDBACK PAGE



### Vehicle Requests.....

Username|Vehicle Name|Register Number|Category Name



Figure 9.19: DRIVERS PAGE

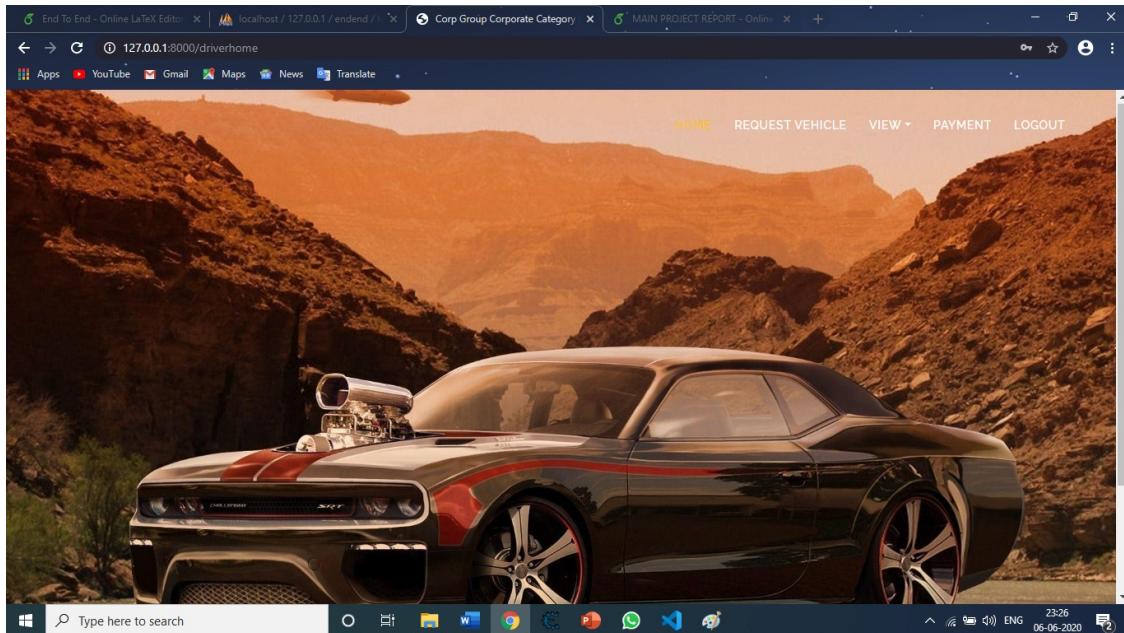
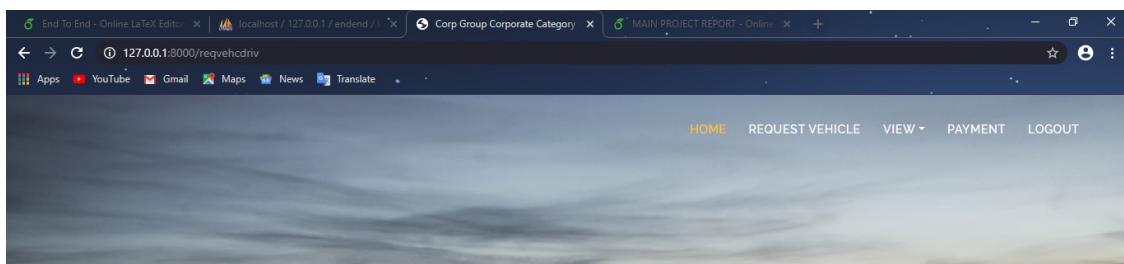


Figure 9.20: REQUEST PAGE

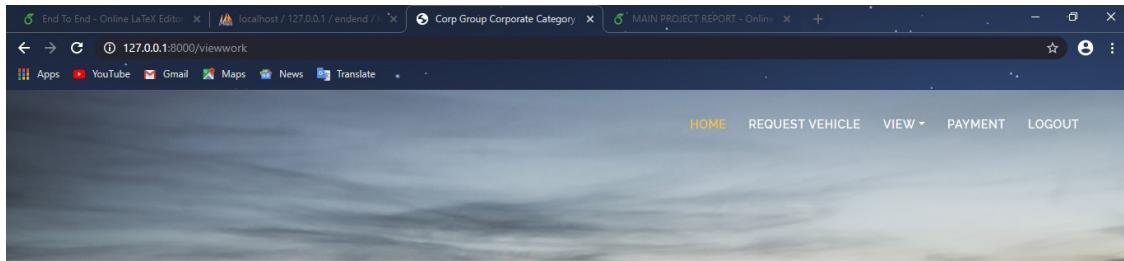


### Request For Vehicle.....

Vehicle ID	Vehicle Name	Register Number	Category Name	Action
1	maruthy	kl72323	car	Request
2	max	kl32d5050	jeep	Request



Figure 9.21: WORKS PAGE

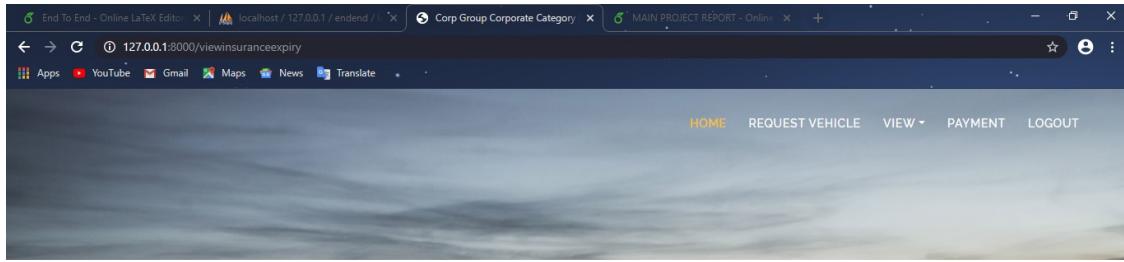


### View Works.....

Driver Id	Work	Work Date	Estimated Kilometer	Estimated dayzs	Work Load



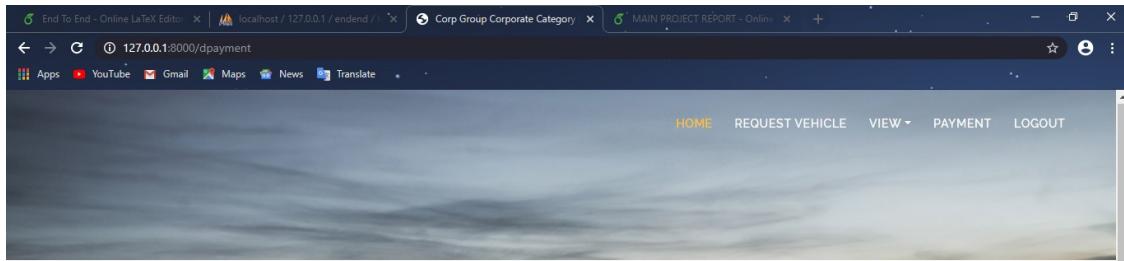
Figure 9.22: INSURANCE PAGE



### Expiry Dates .....

Expiry Date	Vehicle Name	Vehicle Register Number
Oct 18, 2019	maruthy	KL72323

Figure 9.23: PAYMENT PAGE



### Payment.....

User Name	manu@gmail.com
Category	jeep
Vehicle Name	max
Register Number	KL32D5050
Amount	2000
No Of Days Needed	
Required Date	dd-mm-yyyy <input type="button" value="Calendar"/>

