

# **TRUCKDESK WEB APPLICATION**

## **A PROJECT REPORT**

**Submitted by**

**BALAJI.S  
17CSR014**

**BOOPALAN.M  
17CSR020**

**GOKULNATH.P  
17CSR043**

*In partial fulfilment of the requirements*

*for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES**



**KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI, ERODE - 638060**

**DECEMBER - 2020**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
KONGU ENGINEERING COLLEGE  
(Autonomous)**

**PERUNDURAI, ERODE – 638060**

**DECEMBER – 2020**

**BONAFIDE CERTIFICATE**

This is to notify that Project Report entitled **TRUCKDESK WEB APPLICATION** is the bonafide record work done by **BALAJI.S (17CSR014), BOOPALAN.M (17CSR020), GOKULNATH.P (17CSR043)**, in partial fulfilment of the requirement for the award of the Degree of Bachelor of Engineering in **Computer Science and Engineering** of Anna University, Chennai during the year 2020-2021.

**SUPERVISOR**

**HEAD OF THE DEPARTMENT**

**(Signature with seal)**

**Date:**

Submitted for the end semester viva voice examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING****KONGU ENGINEERING COLLEGE****(Autonomous)****PERUNDURAI, ERODE – 638060****DECEMBER - 2020****DECLARATION**

We affirm that the Project report titled “**TRUCKDESK WEB APPLICATION**” being submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering is the original work carried by us. It has not formed the part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**BALAJI.S**  
**(Reg.No:17CSR014)**

**BOOPALAN.M**  
**(Reg.No:17CSR020)**

**GOKULNATH.P**  
**(Reg.No:17CSR043)**

**Date:**

I certify that the declaration made by the above candidates is true to the best of my knowledge.

**Date:****Name & Signature of the Supervisor with seal**

## ABSTRACT

Transport domain is a domain where owners do their business with the set of vehicles. With vehicles, they maintain lots of record for each trip of each vehicle. In single record, they include distance travelled in a particular trip, rent of the load, diesel details, driver details and other expense details as well. It takes more time to record complete details of that particular trip which is finished and they have to maintain large paper works till the end of the year. At the end of the year, they close the accounts using recorded details and calculate the yearly profit or loss.

The main challenge is to balance both the vehicle physically as well as record it in more detailed manner which is less possible but they managed to do it. Unfortunately if something happens to those records owners will lose lot of money and time to compensate it. The proposed system portrays a web application which is compatible with both computer and mobile that digitally records the details of the successive trips. By doing it digitally, there is no need of taking too much time. The web application gets all the details about the driver and details about each vehicle like vehicle number, engine number, chassis number and etc, which lead them to identify it quickly.

The Truckdesk application record trip details in the form of memo which is in a standard format. Eventually, there is no need to do accounting works on those recorded data as it dynamically does for them by creating an application. The calculations are automatically done once after the data are entered in the particular fields, so these features make it an easy task for the organization.

## ACKNOWLEDGEMENT

We express our sincere thanks and gratitude to our beloved Correspondent **Thiru. P. SACHITHANANDAN**, for giving the opportunity to pursue this course. We are extremely thankful to the Principal **Dr. V. BALUSAMY BE(Hons), MTech., PhD.**, for providing necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department **Dr. N. SHANTHI M.E, PhD.**, for providing necessary facilities.

We extend our thanks to **Dr. S.K. NIVETHA M.E, PhD**, the Project Coordinator for her encouragement and valuable advice that made us to carry out the project work successfully.

We extend our gratitude to our Supervisor **Dr. K. NIRMALA DEVI M.E, PhD.**, for valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department, for their support.

We extend our thanks to **Mr.M. SARAVANAN**, owner of Thirumalaivasa builders for his support to carry out the project work successfully.

## TABLE OF CONTENTS

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
<b>No.</b>		<b>No.</b>
	Abstract	iv
	List of Figures	viii
	List of Abbreviations	ix
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1. Existing System	2
	1.2. Proposed System	3
	1.3. System study	4
	1.4. Objective	5
	1.5. Scope	6
<b>2.</b>	<b>GENERAL DESCRIPTIONS</b>	<b>7</b>
	2.1. Project perspective	7
	2.2. User characteristics	8
	2.3. Design and implementation constraints	9

<b>3.</b>	<b>REQUIREMENTS</b>	<b>11</b>
3.1.	Hardware Requirements	11
3.2.	Software Requirements	11
3.3	Functional Requirements	11
3.3.	Non-functional Requirements	12
<b>4.</b>	<b>DETAILED DESIGN</b>	<b>13</b>
4.1.	Architectural design	13
4.2.	Interface design	16
4.3.	Database design	21
4.4.	Output design	22
<b>5.</b>	<b>RESULTS AND DISCUSSION</b>	<b>26</b>
<b>6.</b>	<b>TESTING</b>	<b>28</b>
<b>7.</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>29</b>
	<b>APPENDIX 1</b>	<b>30</b>
	<b>APPENDIX 2</b>	<b>38</b>
	<b>REFERENCES</b>	<b>42</b>
	<b>CONSULTANCY CERTIFICATE</b>	<b>43</b>

## LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	User characteristics diagram	8
2.2	Front-end design	9
2.3	Back-end design	10
4.1	An overview of the System Architecture	13
4.2	Use case diagram for User scenario	14
4.3	Use case diagram for Memo scenario	15
4.4	Home page	16
4.5	Login page	16
4.6	Choosing account page	17
4.7	Modules	17
4.8	Dashboard	18
4.9	Add driver	18
4.10	View driver details	19
4.11	Edit driver details	19
4.12	Add vehicle	20
4.13	Tracking due	20
4.14	Database design	21
4.15	Section for vehicle information	22
4.16	Load information	23
4.17	Diesel information	23

4.18	Information for other expenses	24
4.19	RTO and PC information	24
4.20	Section for results	25
4.21	Section for overall details	25
5.1	Result 1	27
5.2	Result 2	27
A2.1	Login page	38
A2.2	Dashboard activity	38
A2.3	Driver details page	39
A2.4	Vehicle details page	39
A2.5	Memo entry page	40
A2.6	Memo submission page	40
A2.7	Due tracking page	41

## **LIST OF ABBREVIATIONS**

API	-	Application Program Interface
CDN	-	Content Delivery Network
CSS	-	Cascading Style Sheet
DB	-	Database
FC	-	Fitness Certificate
HTML	-	Hyper Text Markup Language
HTTP	-	Hypertext Transfer Protocol
JS	-	JavaScript
JSON	-	JavaScript Object Notation
JSX	-	JavaScript XML
KM	-	Kilo meter
MERN	-	MongoDB, ExpressJS, ReactJS, NodeJS
PC	-	Police
PDF	-	Portable Document Format
RTO	-	Regional Transport Authority
URL	-	Uniform Resource Locator
XML	-	Extensible Mark-up Language

## CHAPTER 1

### INTRODUCTION

In Transport the owner needs to maintain a large amount of data with the set of vehicles. With vehicles, they maintain lots of record for each trip of each vehicle. In single record, they include distance travelled in a particular trip, rent of the load, diesel details, driver details and other expense details as well. They handle with many vehicle's data manually which is difficult for them. It takes more time to record complete details of that particular trip which is finished and they have to maintain large paper works till the end of the year. At the end of the year, they close the accounts using recorded details and calculate the yearly profit or loss. There may occur some destructive actions to those records like accidental fire. The main challenge is to balance both the vehicle physically as well as record it in more detailed manner which is less possible but they managed to do it.

The Truckdesk application is compatible with both computer and mobile that digitally records the details of the successive trips. By doing it digitally, there is no need of taking too much time. The application helps recording trip details in the form of trip sheet which is in a standard format. Eventually, there is no need to do accounting works on those recorded data as we dynamically do for them by creating an application. The calculations are automatically done once after the data are entered in the particular fields, so these features make it an easy task for the organization.

The web application has been created using ReactJS, MongoDB, ExpressJS and NodeJS. This MERN stack development will make their process easier and gives more data visuals dynamically. The limitations that we will face are there is only 512MB of free MongoDB version. User asked for google authentication service instead of remembering

new credentials. The third-party authentication service that allows only 7000 user logins per month. The major advantage of this application is owners no need to worry about the maintaining large paper works and need only less time to record the details of the successive trips. They can carry information about their vehicles and view wherever they need. Eventually, they can keep track of every year's profit and loss accounts.

## **1.1 EXISTING SYSTEM**

The Organisation uses the hand written manual trip sheet . For every trip the details of the driver and the vehicle details are monitored manually. Each and every calculation from basic level to any extends are performed manually, so human errors may occur during accounts calculation. The organization enters the trip sheet manually. Each truck dues have to monitored, there is a need for maintaining large number of records. It consumes more time and an inefficient one.

## 1.2 PROPOSED SYSTEM

The proposed project provides an automated application for the organization. It provides access to view all the information about the driver and vehicle in that particular organization. The information related to the trip can be viewed only by the user for knowing about the activities. User can add new driver, truck and details related to it in the website, it is done in the memo section which can be stored in the database and retrieved later for further activities. In the dashboard section, the particular vehicle and its activities can be found separately. It makes the process much easier than the existing one. users can only add the required details and authorized users can be able to view it. If needed the trip details can also be generated in PDF format.

This system gives following functionalities,

**Automated calculation:** In this system, when user enters the data the website will calculate automatically.

**Details uploading:** User can upload the details to the database via the website after logging in.

**Report collection:** The authorized users can be able to view the reports and also, they can download it for their reference.

**Form generation:** If needed the trip details including all the expenses can be generated in a PDF format.

**Due Tracking:** Keeps track on the due amount and dates for each and every vehicle.

### 1.3 SYSTEM STUDY

The aim of the project is to minimise and automate the hand written works and paper maintenance. The owners need to login with their Google account. Once they logged in successfully, the owners can view the details of a vehicle, drivers, memo and dues of a vehicle. And they can also download generated memo of a vehicle.

The Truckdesk web application has a menu that contains the following options: Dashboard, add vehicle, add Drivers, memo, due options and Logout. List of vehicles is shown in the dashboard option in which the owners can view the vehicle income and expense details of which ever year they need. Owners can add and view the details of national permit, Insurance Details, FC details, Tax details and payment status.

In Add driver option, they can enter the details of a new driver then can view, edit and delete the details of a driver. In add vehicle option, they can add a vehicle by entering chassis number, Engine number, Vehicle Model, Vehicle number, Due interest and total number of dues for the vehicle.

In memo option, the owner has to enter the data related to their trip and can preview the data in the form of trip sheet and can download as a PDF document. Finally, owner can save trip data in the database.

List of vehicles is shown in Due option. Owners can check out their due that have been paid so far. And can save the same to the database which can be viewed later. Owners can also logout from the web application. The maximum session maintained by the server is 1 hour.

## 1.4 OBJECTIVE

The organisation writes the trip sheets manually, there was a delay in writing calculated amounts in a memo. Owners could not manage their income and expense year wise for a single vehicle. In order to overcome, Truckdesk web application has been established.

The main objectives are:

- To provide Google Authentication service.
- To provide a feature that shows the details of every year profit or loss amount and a graph plotted with amounts per month.
- To provide a feature to maintain the driver details.
- To provide a feature to add a new vehicle to their account.
- To provide a feature to enter the trip details with auto calculation facility.
- To generate a trip sheet as a PDF with the given trip details.
- To download files like generated PDF or yearly graphs to the local storage.
- To track due details for the newly registered vehicles.

## 1.5 SCOPE

- This project traverses the area of truck owners from viewing information to generating and downloading generated trip sheet from the web application.
- This website contains dynamic form generation that should be filled by the owners.
- Owners can able to use the system effectively and it responsive among mobile and computers
- Web platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal and server maintenance.

## CHAPTER 2

### GENERAL DESCRIPTION

#### **2.1 PROJECT PERSPECTIVE**

The proposed system is a web application which is compatible with both computer and mobile that digitally records the details of the successive trips. By doing it digitally, there is no need of taking too much time. Truckdesk collects all the details about the driver and details about each vehicle, which lead them to identify it quickly. It records trip details in the form of trip sheet which is in a standard format. Eventually, there is no need to do accounting works on those recorded data as we dynamically do for them by creating an application.

The proposed system is made up using ReactJS, MongoDB, ExpressJS and NodeJS. This MERN stack development will make their process easier and gives more data visuals dynamically. The limitations that we will face are there is only 512MB of free MongoDB version. User asked for google authentication service instead of remembering new credentials. It uses third party authentication service that allows only 7000 user logins per month. The major advantage of this application is owners no need to worry about the maintaining large paper works and need only less time to record the details of the successive trips. Owners can carry information about their vehicles and view wherever they need.

In trip sheet the form is divided into seven step process in which the details to be entered by the owners are (1) vehicle information, (2) Trip details, (3) Loading details, (4) Diesel information, (5) RTO and PC Details, (6) Other expenses and (7) Overall - which is a final step. In the final step, owners can generate and download the memo in a PDF format. Eventually, they can keep track of every year's profit and loss accounts.

## 2.2 USER CHARACTERISTICS

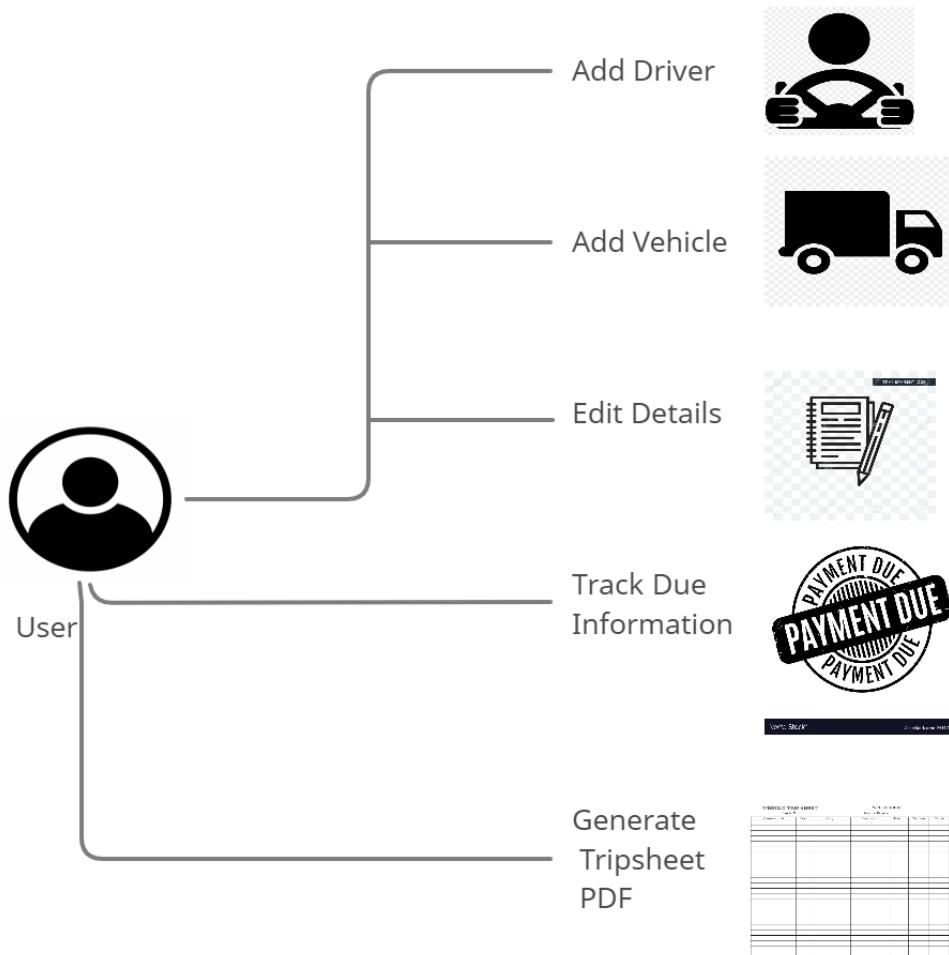


Figure 2.1 User characteristics diagram

Figure 2.1 shows that only the authenticated user is allowed to enter into the web application, so that the user can view, edit and delete the data which is entered by them. The user can also add new driver and vehicle in the application whenever needed.

## 2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

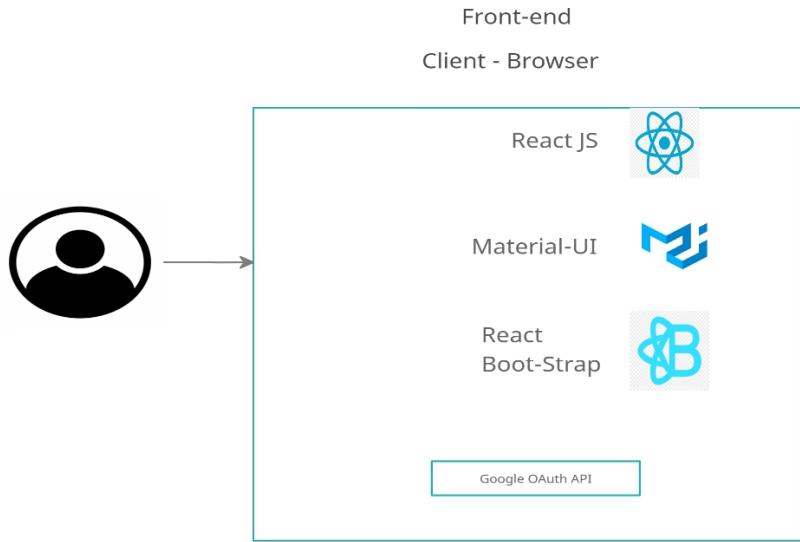


Figure 2.2. Front-end design

Figure 2.2 shows that the Truckdesk is a web-based application and will be implemented with MERN stack development. React JSX (JavaScript XML) which is an alternate to HTML (Hyper Text Mark-up Language), Material UI used for styling, express-JS used as a backend framework for data saving and retrieval from the MongoDB (Mongo Database).

JSX describes the structure of web pages using the markup. JSX elements are the building blocks of React Components. These elements are represented by tags. In the website it is used to develop the front end.

Material UI describes how the JSX elements are to be displayed on screen. Material UI saves a lot of work. It can control the layout of multiple React Components all at once. This can be taken from Material UI CDN.

React-strap is a popular web development framework used for creating websites. The React-strap framework includes styles, JavaScript libraries, and JSX files. React-strap provides a way for developers to easily build responsive websites rather than designing them from scratch.

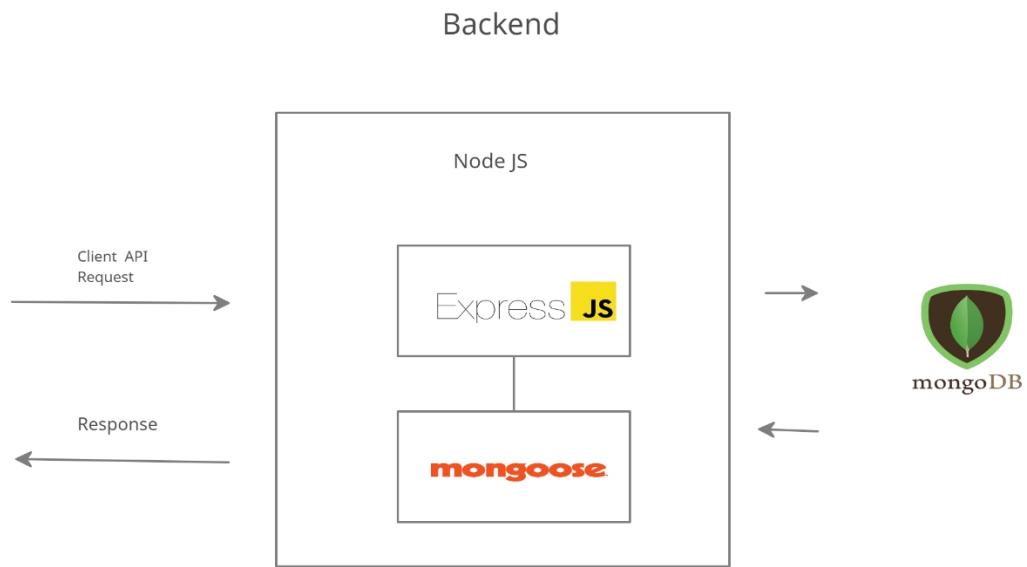


Figure 2.3. Back-end design

Figure 2.3 shows the NodeJS is a JavaScript environment that helps people make back-end servers and front-end frameworks more interactive by allowing them to do more intelligent, complex things.

ExpressJS Handles the client request and store the processed data in mongo dB by using the middleware called mongoose.

## **CHAPTER 3**

### **REQUIREMENTS**

#### **3.1 HARDWARE REQUIREMENTS**

- Processor with 2GHz frequency.
- A minimum of 1GB of RAM.
- Screen resolution: Normal

#### **3.2 SOFTWARE REQUIREMENTS**

- Operating System: Windows 10, Linux
- React-strap
- A browser which supports HTML, CSS, JavaScript

#### **3.3 FUNCTIONAL REQUIREMENTS**

##### **Owners Function**

The owner can update, retrieve the formats and reports and maintain the website.

##### **Authentication**

The owner should login with their google account in google consent screen.

##### **Trip sheet Requirements**

For memo generation, owners are need to be enter details of the trip sheet of a trip.

##### **Driver Requirements**

The owner should enter the name, license number, age, insurance number of a driver to be added to the application.

### **3.4 NON-FUNCTIONAL**

#### **REQUIREMENTS Usability**

The user end of the system must be very easy to understand. The time taken for user to be familiar with the system will be less than 10 minutes. The average task time for a random user will not exceed 20 minutes.

#### **Availability**

The user can access the Truckdesk application from any place at any time. The availability of the website will 99%.

#### **Performance Response Time**

The response time of the Truckdesk website is low and provides the results quickly.

#### **Throughput**

Throughput of the application is considerably adequate to provide a continuous service for the users.

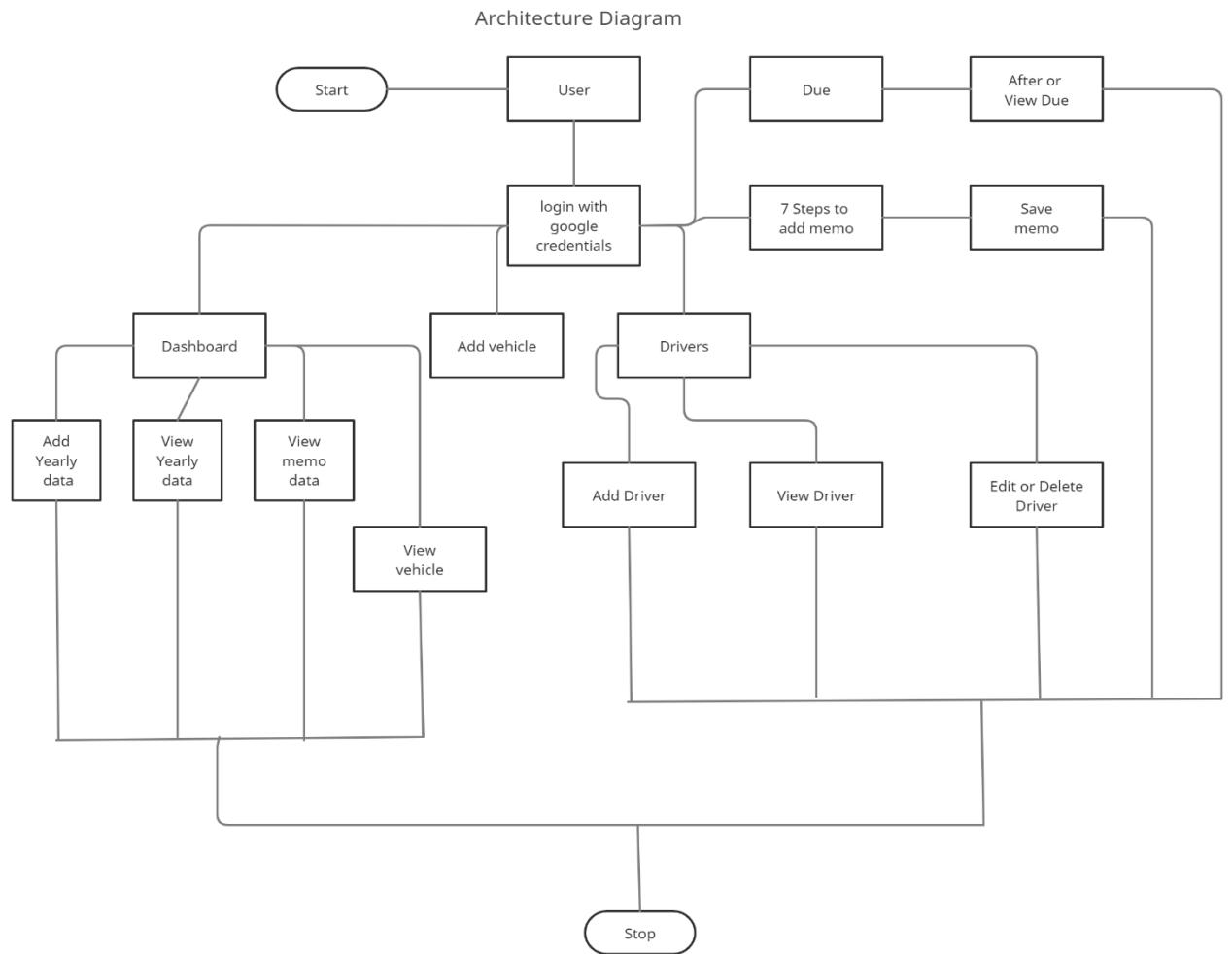
#### **Security**

To enter into the application the user has to go through the google authentication, only the authorized user can enter into the application. Thus, after the authentication process the user is considered as user. The user can enter, view and edit the data in the database

## CHAPTER 4

### DETAILED DESIGN

#### 4.1 ARCHITECTURAL DESIGN

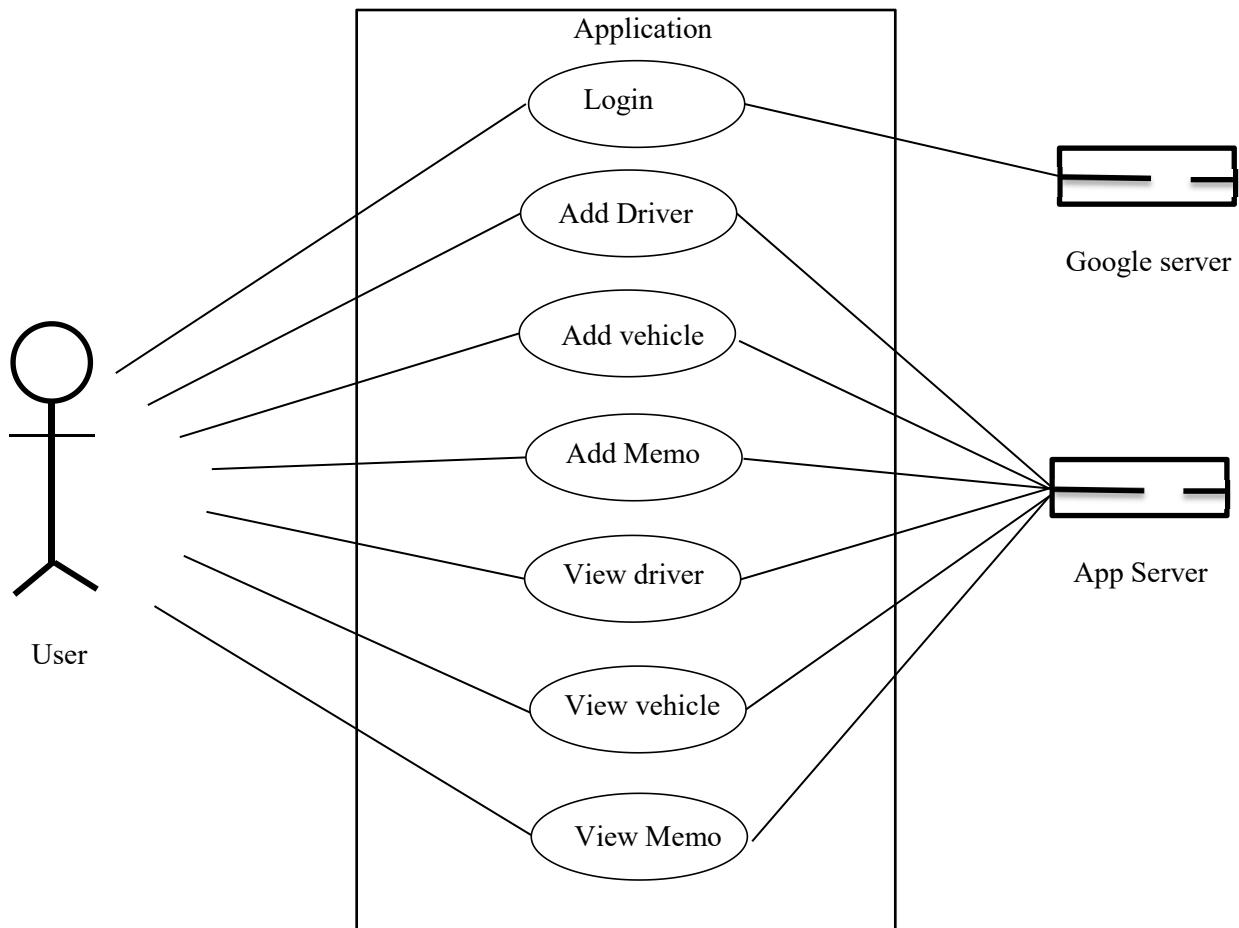


**Figure 4.1. An overview of the System Architecture.**

Figure 4.1 shows the overview of system architecture. In the proposed website, the users can be categorized as user. In the user interface, anyone can create, view and edit the details. In the user interface, the user must have been given authorization from the application to login with a valid google account. The user has the rights to generate the trip sheet in PDF format if needed.

**ACTORS:**

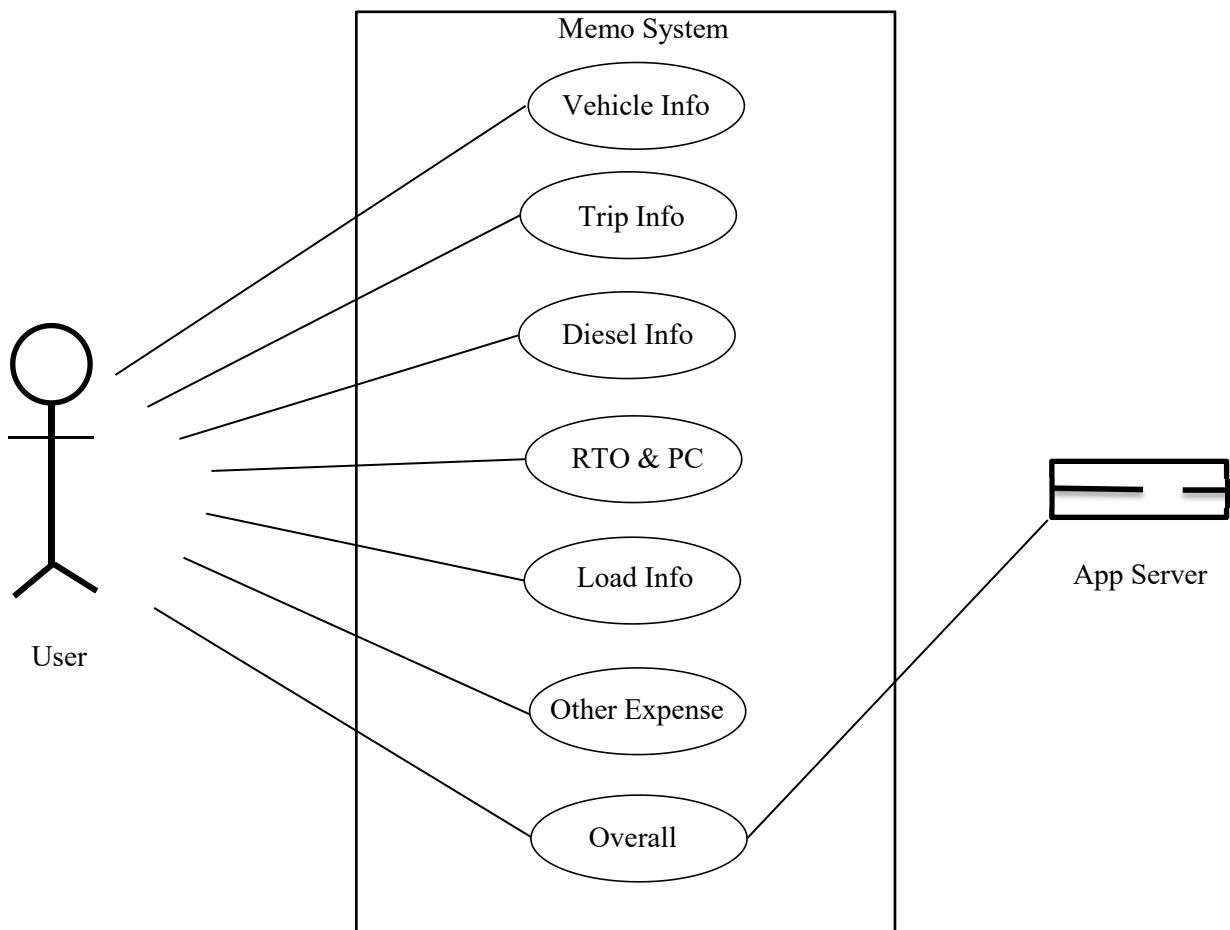
1. Owner or Admin.
2. Memo system.
3. Truckdesk app server



**Figure 4.2 Use Case diagram for user scenario**

### ACTORS:

1. Owner or Admin.
2. Google server.
3. Truckdesk app server



**Figure 4.3 Use Case diagram for memo scenario**

Figure 4.2 and Figure 4.3 shows the Use Case of Truckdesk website respectively.

From this diagram, when a user enters into the website, only the authorized user can enter into the website. Authentication is done using the Google server. Hence the authorized user is considered as User. The User can add, view and edit the driver details, vehicle details and trip sheet. Once the data in the are entered, the application automatically calculates the amount and generates the accurate answer in the respective field, so that it could not be changed after the calculations are made. The entered data are stored in the application's server.

## 4.2 INTERFACE DESIGN

**Loading Page:**

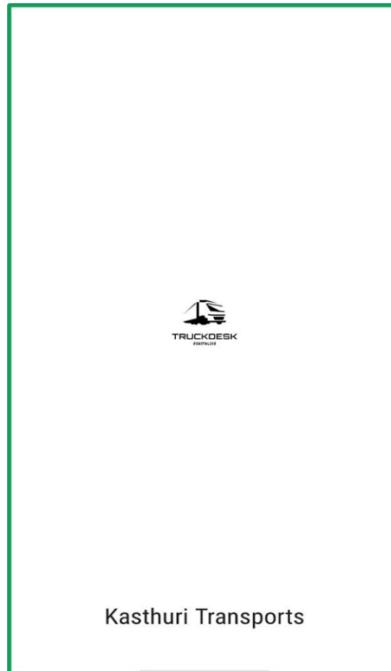


Figure 4.4. Home Page

Figure 4.4 shows the Loading page gives a description about the name of the organization and the application icon.

**Login Page:**



Figure 4.5. Login Page.

Figure 4.5 shows the login page providing the name of the application and the motive of the application google authentication.

### Selecting user:

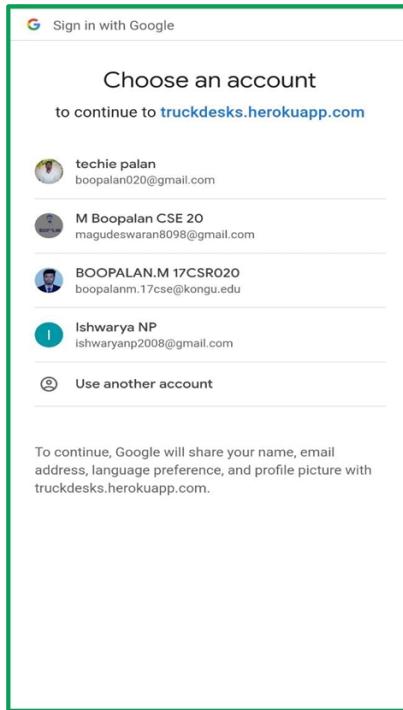


Figure 4.6. Selecting User

Figure 4.6 shows the page that allows the user to choose the account in which they need to log in into the application.

### Modules view page:

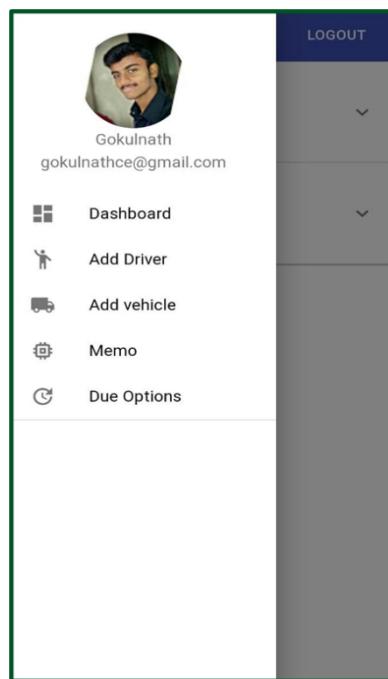


Figure 4.7. Modules

Figure 4.7 shows the different options that are present in the application and clicking the option will direct the user to that particular page

### Dashboard page:

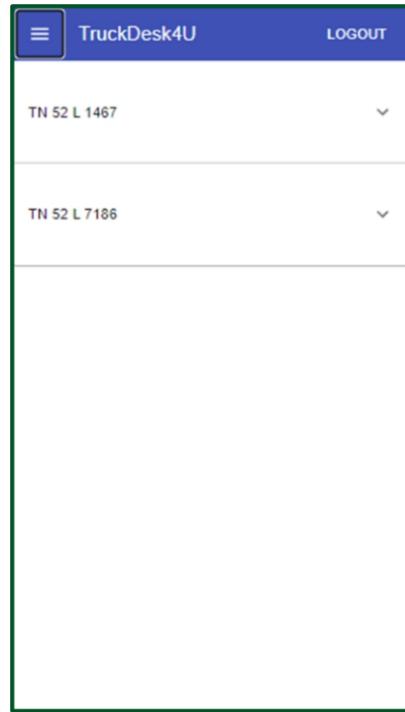


Figure 4.8. Dashboard

Figure 4.8 shows the overall performance of the vehicle in a bar chart view and the cost of the trip details and income from that particular vehicle will also be displayed.

### Add driver page:

The screenshot shows the 'Add Driver' page for 'TruckDesk4U'. The page has a white background with a blue header bar at the top. The header bar includes a back arrow labeled '←', the app name 'TruckDesk4U', and a 'LOGIN' button. Below the header, the page title 'Add Driver' is centered above a series of input fields. Each field has a placeholder text and a value entered: 'Driver Name' (John Doe), 'License.no' (1323fsd23432), 'Blood Group' (B+), 'Age' (34), 'Address' (Salem), 'Phone.no' (9934961994), and 'Driver Insurance' (empty). At the bottom of the form is a blue 'ADD' button with a white plus sign.

Figure 4.9. Add driver

Figure 4.9 shows the page that enables the user to add new driver in the organization, by pressing the add button the driver details are added to the database.

### View driver details:

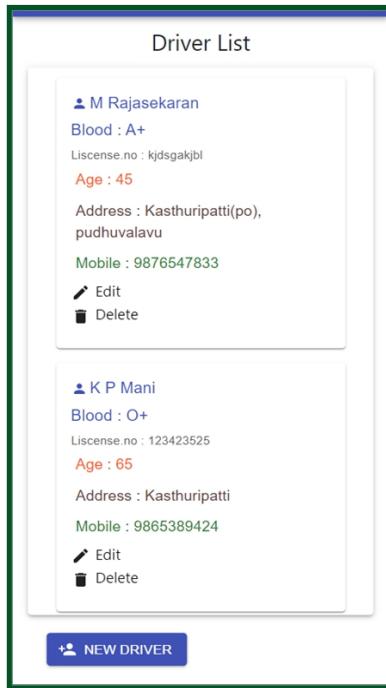


Figure 4.10. View driver details

Figure 4.10 views the details of the drivers which are already added to the database, by clicking on the new driver button will enable the user to add new driver to the database again.

### Edit driver detail:

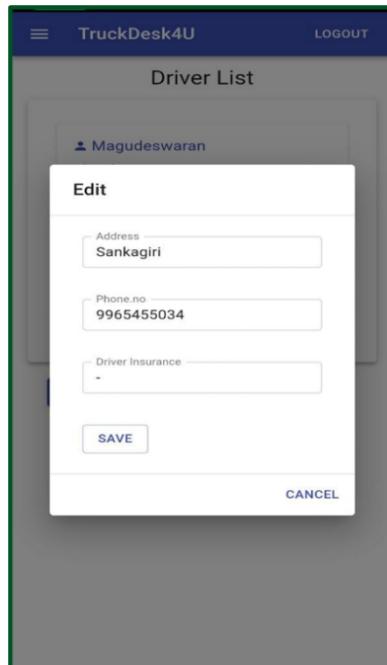


Figure:4.11. Edit driver detail

Figure 4.11 shows the edit option under each driver details allows us to edit the particulars of the driver even after saving it in the database.

### Add vehicle:

The figure consists of two side-by-side screenshots of a mobile application interface. Both screens have a header 'TruckDesk4U' and a 'LOGIN' button in the top right corner.

**Left Screen (Vehicle Details):**

- Vehicle Number: TN 00 AB 1234
- Registration date: 01-11-2020
- Chassis Number: sdjfsgafgd1314re
- Engine Number: Testing engine number 1
- Vehicle Model: Testing Vehicle
- Due detail:**
  - Total Due Amount: 3040565
  - Due Interest: 4.4
  - Total Due Month: 59
  - Completed Due month: 45
- NEXT** button at the bottom.

**Right Screen (Additional Details):**

- National Permit:**
  - Permit Date: 05-11-2020
  - Cost: 2470
- Insurance Detail:**
  - Insurance Date: 14-11-2020
  - Cost: 1540
- Fitness Certificate Detail:**
  - FC Date: 13-11-2020
  - Cost: 2900
- Quarter Tax Detail:**
  - Quarter Tax Date: 09-11-2020
  - Cost: 4000
  - RTO: 3500
- Payment Status:**
  - Paid
  - Unpaid

Figure 4.12. Add vehicle

Figure 4.12 shows add vehicle option is used to add a new vehicle to the organization and the added vehicle is stored in the database

### Tracking due:

The figure is a screenshot of a mobile application showing a list of due payments for a vehicle. The vehicle ID is TN 52 L 7186 and the chassis number is testchasisnumber2.

Date	Current Amount
9/2/2022	914069.17
10/2/2022	812505.94
11/2/2022	710942.71
12/2/2022	609379.48
1/2/2023	507816.25

At the bottom, it shows:  
**Paid Month(s): 22**      **Unpaid Month(s): 8**      **SAVE**

Figure 4.13. Tracking due

Figure 4.13 shows the page that enables user go through the due payments whether the due has been paid or not.

### 4.3 DATABASE DESIGN

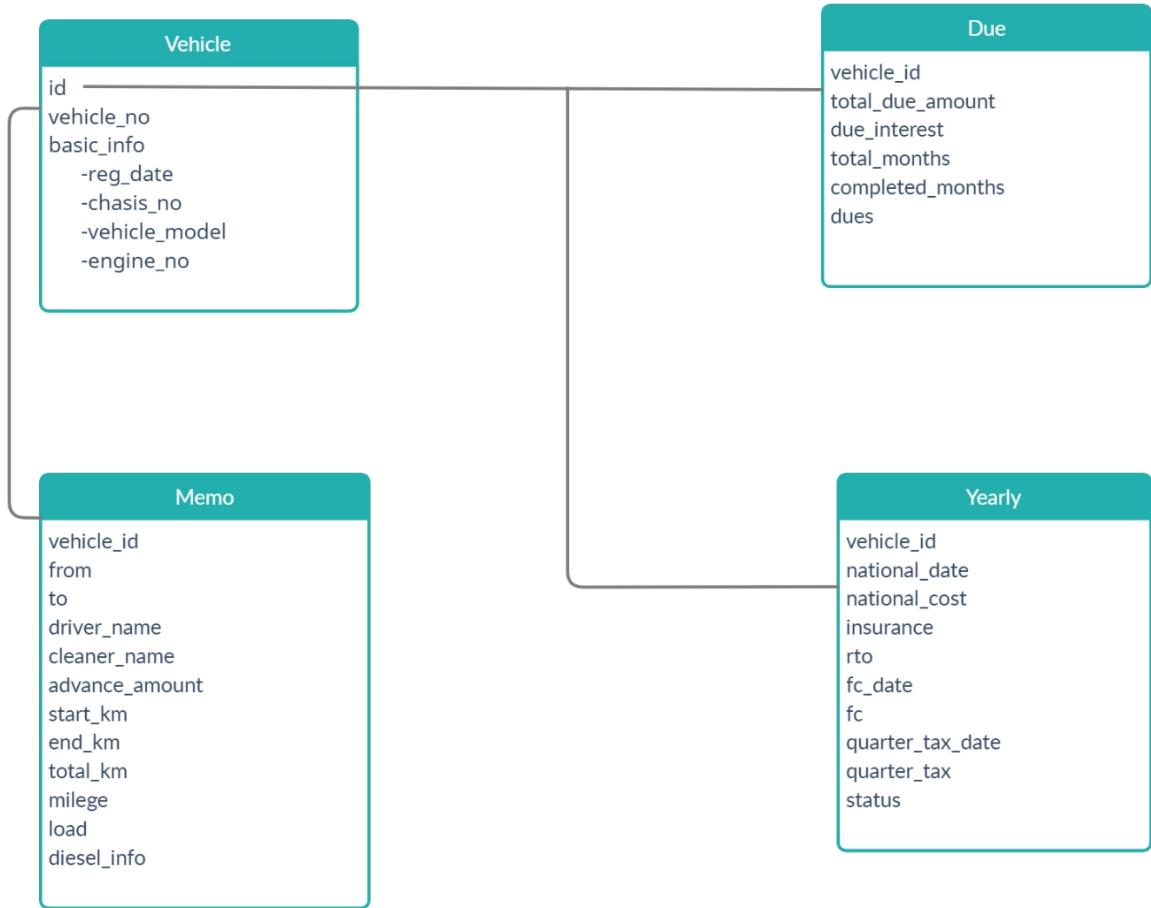


Figure 4.14 Database design

Figure 4.14 shows the database design of a module in which there are several JSON objects like vehicle, driver, memo, yearly and due for storing the vehicle details, driver details, memo details, yearly details and due details uploaded by the user respectively. The JSON objects are designed according to the fields used in the user interface. The details given by the user are validated and stored in the separate objects. Owner can add, edit, view and delete data in the database. Owners can generate PDF and can download the generated PDF to their local storage.

## 4.4 OUTPUT DESIGN

The Truckdesk web application display the information about the trip sheet year wise, due details month wise and existing driver details. This provide google authentication with google consent screen for owner's login purpose. Furthermore, The owners can enter the details of a trip sheet corresponding to a particular trip, then they can generate, view and download the PDF to the local storage. Can add a new arriving driver details and can edit the same for the later use. This application will respond across multiple devices like mobile and tablets. This application provides details periodically about the vehicle, driver, memo and due of the vehicle to the owner

### Steps for Output :

லாரி நெ.....	அட்வான்ஸ்.....	மைலேஜ்.....	தேதி.....					
ஆரம்ப கி.மீ.....	முடிவு கி.மீ.....	மொத்த கி.மீ.....	மூசல்.....					
டிரைவர் பெயர் 1. ....	2. ....	கிளீனர் பெயர்.....	தேதி.....To.....					
தேதி	புறப்படும் இடம் To சேருமிடம்	லோடு டன் வரவு	கமிஷன்	ஏற்றுக்கூலி	இறக்குக்கூலி	இதர செலவுகள்	ரூ.	பை.
தேதி	மூசல்	ரூ.	பை.	ரூ.	பை	மொத்தம்	ரூ.	பை.
						மூசல்		
						கமிஷன்		
						ஏற்றுக்கூலி		
						இறக்குக்கூலி		
						இதர செலவுகள்		
						டிரைவர் படி		
						கிளீனர் படி		
						பட்டறை செலவு		
						RTO & PC செலவு		
						பாலம்		
						நாக்கா		
						பில்படி		
						பத்தாயம்		
						மொத்த செலவு		
						மொத்த வரவு		
						மீதி இருப்பு		

Figure 4.15 Section for vehicle information

Figure 4.15 shows that the data like driver name, starting and ending km are given in step 1 will be appeared on the above highlighted columns.

லாரி நெ.....	அட்வான்ஸ்.....	மைலேஜ்.....	தேதி.....							
ஆரம்ப கிழி.....	முடிவு கிழி.....	மொத்த கிழி.....	மூசல்.....							
டிரைவர் பெயர் 1.....	2.....	கீளைர் பெயர்.....	தேதி.....To.....							
தேதி	புறப்படும் இடம் To சேருமிடம்	வோடு	டன்	வரவு	கமிஷன்	ஏற்றுக்கூலி	இரக்குக்கூலி	இதர செலவுகள்	ரூ.	பை.
தேதி	மூசல்	ரூ.	பை.		ரூ.	பை	மைத்தம்		ரூ.	பை.
							மூசல்			
							கமிஷன்			
							ஏற்றுக்கூலி			
							இரக்குக்கூலி			
							இதர செலவுகள்			
							டிரைவர் படி			
							கீளைர் படி			
							பட்டறை செலவு			
							RTO & PC செலவு			
							பாலம்			
							நாக்கா			
							பிள்படி			
							பத்தாயம்			
							மைத்த செலவு			
							மொத்த வரவு			
							நீதி இருப்பு			

Figure 4.16 Load information

Figure 4.16 shows that the data like load, load type, weight, commission are given in step 2 will be appeared on the above highlighted columns.

லாரி நெ.....	அட்வான்ஸ்.....	மைலேஜ்.....	தேதி.....							
ஆரம்ப கிழி.....	முடிவு கிழி.....	மொத்த கிழி.....	மூசல்.....							
டிரைவர் பெயர் 1.....	2.....	கீளைர் பெயர்.....	தேதி.....To.....							
தேதி	புறப்படும் இடம் To சேருமிடம்	வோடு	டன்	வரவு	கமிஷன்	ஏற்றுக்கூலி	இரக்குக்கூலி	இதர செலவுகள்	ரூ.	பை.
தேதி	மூசல்	ரூ.	பை.		ரூ.	பை	மைத்தம்		ரூ.	பை.
							மூசல்			
							கமிஷன்			
							ஏற்றுக்கூலி			
							இரக்குக்கூலி			
							இதர செலவுகள்			
							டிரைவர் படி			
							கீளைர் படி			
							பட்டறை செலவு			
							RTO & PC செலவு			
							பாலம்			
							நாக்கா			
							பிள்படி			
							பத்தாயம்			
							மைத்த செலவு			
							மொத்த வரவு			
							நீதி இருப்பு			

Figure 4.17 Diesel information

Figure 4.17 shows that the data like diesel, litre, amount are given in the step 3 will be appeared on the above highlighted columns.

லாரி நெ.....	அடவாளர்ஸ்.....	மைவேல்.....	தேதி.....							
ஆரம்ப கிடம்.....	முடிவு கிடம்.....	மொத்த கிடம்.....	மசல்.....							
டிரைவர் பெயர் 1.....	2.....	கிளீனர் பெயர்.....	தேதி.....							
தேதி	புறப்படும் இடம் To சேருமிடம்	வோடு	டன்	வரவு	கமிஷன்	ஏற்றுக்கூலி	இரக்குக்கூலி	இதர செலவுகள்	ஏ.	பை.
<b>STEP - 4</b>										
தேதி	மசல்	ஏ.	பை.	ஏ.	பை.	மொத்தம்	ஏ.	பை.		
						மசல்				
						கமிஷன்				
						ஏற்றுக்கூலி				
						இரக்குக்கூலி				
						இதர செலவுகள்				
						டிரைவர் படி				
						கிளீனர் படி				
						பட்டறை செலவு				
						RTO & PC செலவு				
						பாலம்				
						நாக்கா				
						பில்படி				
						பத்தாயம்				
						மொத்த செலவு				
						மொத்த வரவு				
						மீதி இருப்பு				

Figure 4.18 Information for other expenses

Figure 4.18 shows that the data like reason and amount spent are given in the step 4 will be appeared on the above highlighted columns.

RTO	ஏ.	ஏ.	PC	ஏ.	விபரம்	டிரைவர் வசம்	ஒனர் விசம்
					<b>STEP - 5</b>		

Figure 4.19 RTO and PC information

Figure 4.19 shows that the data like place and amount are given in the step 5 will be appeared in the above highlighted columns.



## CHAPTER 5

### RESULTS AND DISCUSSION

The user who is entering into the web application is considered as user because, only authenticated users are allowed to enter into the web application. The user can register a new vehicle, new driver and trip details, to do so user is required to collect the information regarding vehicle and the driver and the. This web application is mainly created to minimize the time and increase the efficiency of all the calculations done. To overcome the human errors and maintenance of large number of records will be avoided using this application. This web application also keeps track on the due payments of each and every truck. The main advantage is Truckdesk is a web application, so that it is compatible with both personal computer and mobile devices.

Once the data are entered it is stored in the database. This application is also enabled for providing a pdf format for further purposes. This website will be able to undergo evolution in a much simpler way and will be more adaptable to the changing systems. The upcoming changes in the near future will be able to give a popup notification for reminding the user to keep a track on the due payments and the website will be designed in a way to adapt to the changes in the future.



## CHAPTER 6

### TESTING

Validation testing is done when user attempts to login. The user name and password entered by the user and faculty is validated against the user name and password in the program. The login google account should match with the account that have been already authorized. The user can test whether the data entered are stored successfully are not. The unauthorized registered login can also be validated.

Unit testing is testing driver module by which source code are tested to determine if they are ready to use. The main idea here is to test each part of the program and show that the individual modules are correct. This testing will ensure the correctness of the program. It can be performed at any time and tests only the functionality of unit themselves and may not catch integration errors and other system wide issues.

Integration testing is a testing where individual units of a program are combined and tested as a group. The main idea behind integration testing is to combine the modules in the application and test as a group to see that they are working fine. It may detect errors when modules are integrated to build the overall system.

Database testing is to test if any errors are shown while executing the queries. Data integrity is maintained while creating, updating or deleting data in the database. It checks the response time of queries and fine tune there if necessary. Test data retrieved from the database is shown accurately in the web application.

## CHAPTER 7

### CONCLUSION AND FUTURE WORK

#### CONCLUSION

The main objective of Truckdesk web application is to provide a trip sheet in a consistent way, and flaunt less user interface which will make the user to feel easy and user-friendly for performing the operations. This is promoting the usability of the application. The basic functionalities of the Truckdesk web application to view the objectives and outcomes of Truckdesk and describes the outline of Truckdesk. And also describes the functionalities of the application such as viewing vehicle and outcome, uploading and generating the memo as a PDF, adding and deleting the details of vehicles, drivers, memos and dues, viewing the same in dashboard was successfully implemented.

#### FUTURE WORK

The future works will include addition of multiple users such ad driver login, etc. and also support to multiple languages, so that it could be used by the people who are comfortable with their own language. The generated pdf can also be generated in the language that is selected by the user and continuous maintenance will be provided to the application. These features will make the web application much easier to use.

## APPENDIX 1

### CODING

#### **App.js:**

```

import React, { useState } from 'react'

import { Provider } from 'react-redux'

import store from './redux/store'

import MainComponent from './components/MainComponent'

import 'toasted-notes/src/styles.css'

import HeaderComponent from './components/homepage/HeaderComponent'

import { makeStyles } from '@material-ui/core/styles'

import { CssBaseline } from '@material-ui/core'

const useStyles = makeStyles(() => ({
  root: {
    minHeight: '90vh',
    backgroundImage: `url(${process.env.PUBLIC_URL + '/assets/bg.jpg'})`,
    backgroundRepeat: 'no-repeat',
    backgroundSize: 'cover',
    backgroundPosition: 'right',
  },
}));

function App() {
  const classes = useStyles()

  const [loggedIn, setLoggedIn] = useState(false)

  return (
    <Provider store={store}>

```

```

<div>

{
  loggedIn ?

  <MainComponent setloggedIn = { setLoggedIn } /> :

  <div className={classes.root} >

    <CssBaseline />

    <HeaderComponent setLoggedIn = { setLoggedIn } />

  </div>

}

</div>

</Provider>

);

}

export default App;

```

### MainComponent.js:

```

import React from 'react'

import { BrowserRouter as Router, Route, Switch } from "react-router-dom"

import NavbarComponent from './NavbarComponent'

import DriverComponent from './driver/DriverComponent'

import MultistepComponent from './vehicle/MultistepComponent'

import MemoStepComponent from './memo/MemoStepComponent'

import DueComponent from './due/DueComponent'

import DashboardComponent from './dashboard/DashboardComponent'

function MainComponent(props) {

  return (
    <div>

```

```
<Router>

  <NavbarComponent setloggedIn = { props.setloggedIn } />

  <Switch>

    <Route exact path='/' component = { DashboardComponent } ></Route>

    <Route path='/viewdrivers' component = { DriverComponent } ></Route>

    <Route path='/viewvehicle' component =
      { MultistepComponent } ></Route>

    <Route path='/memo' component = { MemoStepComponent } ></Route>

    <Route path='/dueOption' component = { DueComponent } ></Route>

  </Switch>

</Router>

</div>

)

}

export default MainCompone
```

### Server.js:

```

const express = require('express')

const app = express()

const morgan = require('morgan')

const mongoose = require('mongoose')

const cors = require('cors')

const driver = require('./routes/drivers')

const vehicle = require('./routes/vehicles')

const memo = require('./routes/memo')

const dashboard = require('./routes/dashboard')

const path = require('path')

const keys = require('./config/key')

const port = process.env.PORT || 3001

// Mongoose connection

mongoose.connect(keys.mongodb.dbURI, {useNewUrlParser: true, useUnifiedTopology:
true }, (err, result) => {

  if(err)

    console.log('Database server not connected Successfully' + err)

  else

    console.log('Database server connected Successfully' )

})

app.use(express.static(path.join(__dirname, './build'))) //Important line for serving

build files

app.use(morgan('dev'))

app.use(cors({


  // origin : "http://localhost:3000"

  // origin : "http://localhost:3001"
})

```

```

origin : "https://truckdesks.herokuapp.com/"

}))  

app.use('/dashboard', dashboard)  

app.use('/drivers', driver)  

app.use('/vehicle', vehicle)  

app.use('/memo', memo)  

app.get('/*', (req, res) => {  

  res.sendFile(path.join(__dirname, './build', 'index.html'))  

})  

app.listen(port, () => console.log("Server is running at PORT ===" + port))

```

### **Driver\_model.js:**

```

const mongoose = require('mongoose');  

const driverSchema = new mongoose.Schema({  

  drivername : String,  

  license : String,  

  blood : String,  

  address : String,  

  age : String,  

  phone : String,  

  insure_no : String,  

})  

const driver = mongoose.model("drivers", driverSchema)  

module.exports = driver

```

**Due\_model.js:**

```

const mongoose = require('mongoose')

const vehicleDueSchema = new mongoose.Schema({
    vehicle_id : String,
    total_due_amount : String,
    due_interest : String,
    total_months : String,
    completed_month : String,
    dues : [],
});

const vehicleDue = mongoose.model("due", vehicleDueSchema)

module.exports = vehicleDue

```

**Memo\_model.js:**

```

const mongoose = require('mongoose');

const memoSchema = new mongoose.Schema({
    vehicle_id : String,
    Memo : [
        {
            calc_date : String,
            from : String,
            to : String,
            driver_name : String,
            cleaner_name : String,
            advance_amount : String,
            start_km : Number,
            end_km : Number,
        }
    ]
});

const memo = mongoose.model("memo", memoSchema)

module.exports = memo

```

```

total_km : Number,
milege : String ,
loads : [],
new_tyre : String,
old_tyre : String,
expense_details : [
{
  reason : String,
  amount : String
},
],
diesel : [
{
  filled_date : String,
  litre : Number,
  rate : Number,
  rate_on_day : String,
  place : String
},
],
total_diesel_litre : String,
rto_details :
[
{
  place : String,
  amount : String
}
]

```

```

    ],
    trip_duration : String,
    total_diesel_amount : String,
    total_commission : String,
    total_loading : String,
    total_unloading : String,
    total_expense : String,
    driver_salary : String,
    cleaner_salary : String,
    pathayam : String,
    total_rto : String,
    toll_gate : String,
    total_rent : String,
    bill_padi : String,
    trip_expense : String,
    final_balance :
    {
        hands_on : Number,
        income_day : Number,
        income_km : Number,
        expense_km : String
    }
},
],
})

const memo = mongoose.model("memo", memoSchema)

module.exports = memo

```

## APPENDIX 2

### SNAPSHOTS

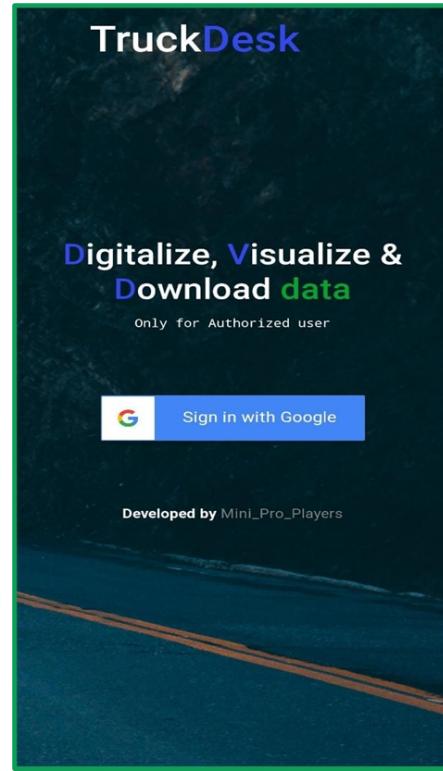


Figure A2.1 Login page

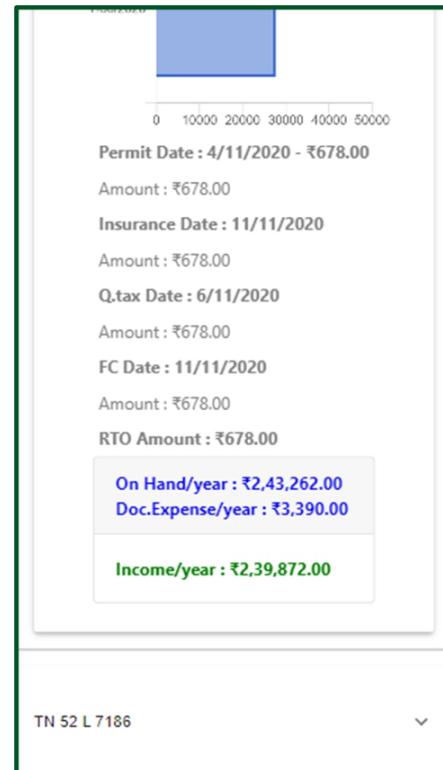
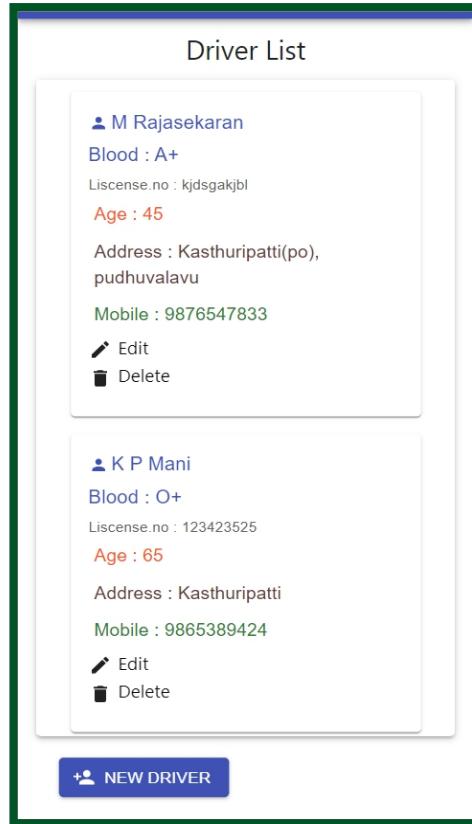


Figure A2.2 Dashboard Activity



**Figure A2.3 Drivers detail page**



**Figure A2.4 Vehicle details page**

TruckDesk4U

LOGIN

Vehicle Info

Vehicel Number  
TN 00 AB 1234

From  
01-11-2020

To  
08-11-2020

Driver Name  
John Doe

Cleaner Name  
-

TRIP DETAILS →

**Figure A2.5 Memo entry page**

Toll gate  
₹ 1900

Bill Padi  
₹ 0

Collection  
₹ 48700

Trip Expense : ₹ 33636

Total Collection : ₹ 48700

Trip Duration : 5 Day(s)

**On Hand Amount : ₹ 15064**

Income/day : ₹ 3012.80

Income/km : ₹ 14.81

Expense/km : ₹ 33.07

**FINISH**

**PREVIEW MEMO**

← RTO & PC INFO    SAVE MEMO

**Figure A2.6 Memo submission page**

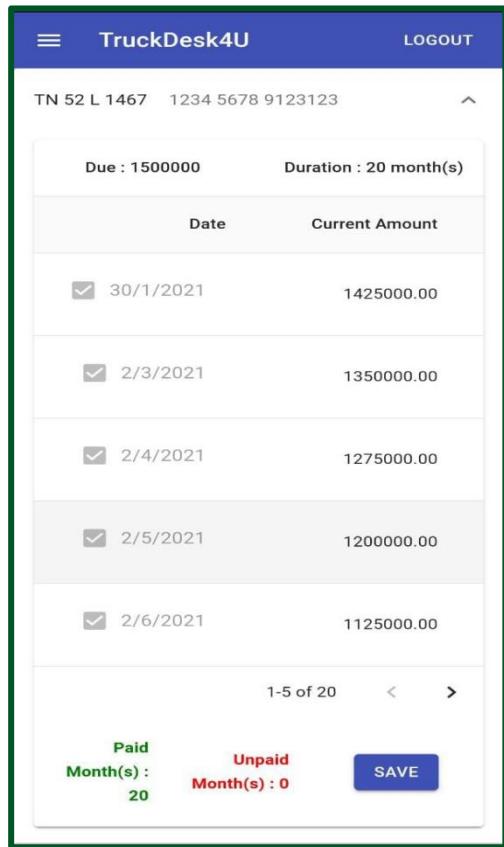
≡ **TruckDesk4U** LOGOUT

TN 52 L 1467 1234 5678 9123123 ^

	Date	Current Amount
<input checked="" type="checkbox"/>	30/1/2021	1425000.00
<input checked="" type="checkbox"/>	2/3/2021	1350000.00
<input checked="" type="checkbox"/>	2/4/2021	1275000.00
<input checked="" type="checkbox"/>	2/5/2021	1200000.00
<input checked="" type="checkbox"/>	2/6/2021	1125000.00

1-5 of 20 < >

**Paid**  
Month(s) : 20      **Unpaid**  
Month(s) : 0      **SAVE**



**Figure A2.7 Due Tracking page**

## REFERENCES

1. ReactJS - <http://reactjs.org/docs>
2. React-Redux - <https://react-redux.js.org/>
3. React-formik - <https://formik.org/>
4. MaterialUI - <https://material-ui.com/>
5. ReactBootstrap - <https://react-bootstrap.github.io/>
6. NodeJs - <https://nodejs.org/en/>
7. ExpressJs - <https://expressjs.com/>
8. Mongoose- <https://mongoosejs.com/>
9. Google console - <https://console.cloud.google.com/>
10. PDF-lib - Andrew Dillon - Senior software engineer at QDivision. created and published PDF-lib works in any JavaScript environment.  
[https://github.com/Hopding/pdf-lib.git.](https://github.com/Hopding/pdf-lib.git)



GSTIN : 33CACPK3571N1ZJ

Sri Murugan Thurai

98427 74902  
98656 55119

# SRI THIRUMALAIVASA LABOUR BODY BUILDERS

Bhavani Main Road, Opp.  
ATC Tippo, SANKAGIRI - 537 301.  
Salem Dt. TamilNadu.

Date: 29.12.2020

To

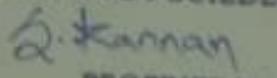
Dr. K.Nirmala Devi and Team,  
Department od Computer Science and Engineering,  
Kongu Engineering College, Perudurai, Erode.

Respected Madam,

We would like to thank you for successfully developing our Progressive Web Application for our transports as a consultancy work. We are really pleased with the work done by you. We would like to extend our thanks to the team members who worked towards completing this Web Application. We also request you to extend your support in future.

Thanking you,

For SRI THIRUMALAIVASA  
LABOUR BODY BUILDERS

  
A. Kannan  
PROPRIETOR