

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum-590014, KARNATAKA, INDIA



A Mini-Project Report

on

“VEHICLE INVENTORY MANAGEMENT SYSTEM”

*A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belgaum.*

Submitted by:

DHEEKSHA S (1AM20CS062)

ELENA JAMES (1AM20CS066)

INDIRA K V (1AM20CS085)

LEKHANA S (1AM20CS106)

Under the Guidance of:

Mrs. Jaya Karuna B

(Assistant Professor, Department of CSE)



Department of Computer Science and Engineering

AMC Engineering College,

18th K.M, Bannerghatta Main Road, Bangalore-560 083

2022-2023

AMC Engineering College
18th K.M, Bannerghatta Main Road, Bangalore-560 083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the mini-project work entitled “**VEHICLE INVENTORY MANAGEMENT DATABASE SYSTEM**” has been successfully carried out by **LEKHANA.S (1AM20CS106), ELENA JAMES (1AM20CS066), INDIRA KV (1AM20CS085) and DHEEKSHA S (1AM20CS062)**, bonafide users of **AMC Engineering College** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The mini project has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Project Guide

Mrs. Jaya Karuna B
Assistant Professor
Department of CSE

HOD

Dr. Nagaraja R
Professor & Head
Department of CSE

Principal

Dr. Girisha C
Principal,
AMCEC

Externals Examiners:

Signature with Date

1

2

ACKNOWLEDGEMENT

First and foremost I would like to thank GOD, the Almighty for being so merciful on me.

I am grateful to my guide **Mrs. Jaya Karuna B**, Asst. Prof., Department of CSE, AMC Engineering College, Bengaluru for her unfailing and constant motivational & timely help, encouragement and suggestion, given to me in the course of my project work.

I would like to extend my special thanks to **Dr. Nagaraja R** Professor and HOD, Department of CSE, for his support and encouragement and suggestions given to me in the course of my project work.

I express my sincere thanks and gratitude to our Principal **Dr. Girisha C** for providing me all the necessary facilities for successful completion of my project.

I have a great pleasure in expressing my deep sense of gratitude to founder Chairman **Dr. K.R.Paramahamsa** for having provided me with a great infrastructure and well-furnished labs for successful completion of my seminar.

Last but not the least, I wish to thank all the teaching & non-teaching staffs of department of Computer Science and Engineering, for their support, patience and endurance shown during the preparation of this project report.

LEKHANA S : (1AM20CS106)

ELENA JAMES : (1AM20CS066)

INDIRA KV : (1AM19CS085)

DHEEKSHA S : (1AM19CS062)

(i)

ABSTRACT

The entitled project “**VEHICLE INVENTORY MANAGEMENT SYSTEM**”. The main objective is to develop an online web-based application from which admin user can easily manage purchase details, building details, supplier’ details, transaction details, spare parts details, employee details and customer details from the browser. It will be capable of doing all the necessary operations/functions that are done in any vehicle’s provisioning terminal.

For all of the above-mentioned details and sub details web interface has been provided which makes it much more convenient. Hence admin user will be able to track all the information and has rights to edit, add, delete and update a record of sales and orders. Since all the work that is to be done by this software can also be done manually, but this consumes a lot time and labor. So this software will be a relief to those who have to do all this work manually.

The knowledge of computers and programming has become a basic skill needed to survive in the present information based on society. The motive to make this project is to make such kind of software which is very easy to use. There will not be a need for any training to take care of the billing system and the person who does not have much knowledge of computers can also use this. Through this project, the details of the customers can be retrieved whenever needed. All the records of the customers will be kept for further inquiries.

Our objective was to make only contact with eligible person and there is no need to explain the room details on the speak. To develop this project we need basics of **HTML, CSS, JS, SQL** and **PHP**. We have made the most friendly and easy access user interface for our users.

We have made use of HTML, CSS, JS for our Frontend development. The backend is made use of MYSQL and PHP. Hosting is done locally in our own system using a XAMPP Server which contains the MYSQL database.

The users can login to their accounts provided by register and can view and update information about houses. All user accounts will be managed by the admin. This project helps in minimizing the efforts and promoting use of new technology for the wellbeing of the society.

We hope our project will be beneficial for usage and further development by others to take it to next levels of development with their innovative id

(ii)

CONTENTS

Chapters	Title	Page No
1.	INTRODUCTION	01
	1.1 Objectives	02
	1.2 Scope	03
2.	SYSTEM SPECIFICATION	04
	2.1 Hardware requirements	04
	2.2 Software requirements	04
	2.3 Information on Languages used	05
	2.4 Requirement analysis	06
	2.5 Feasibility Study	07
	2.6 Limitations	08
3.	DESIGN	09
	3.1 Description of Vehicle Inventory management system , Schema and ER diagrams	09
	3.2 Normalization	15
	3.3 Procedures	18
4.	IMPLEMENTATION AND CODING	19
	4.1 Source code	19
	4.2 Database Design	21
5.	SCREENSHOTS	35
6.	CONCLUSION	43
7.	REFERENCES	44

LIST OF FIGURES

Figure No	Title	Page No
3.1	Relation between the existing tables in the database	07
3.2	Relational schema of the tables in the database	09
3.3	ER diagram of the tables in the database	10
5.1	Login Home Page	35
5.2	Login Page	35
5.3	Home Page	35
5.4	Customer	36
5.5	Add customer	36
5.6	Edit customer	36
5.7	Delete customer	37
5.8	Employee Page	37
5.9	Add employee	37
5.10	Edit employee	38
5.11	Delete employee	38
5.12	Spare parts page	38
5.13	Add spare parts	39
5.14	Edit spart part	39
5.15	Transaction Page	39
5.16	View the transaction of customer	40
5.17	Supplier page	40
5.18	Add supplier	40
5.19	Edit supplier	41
5.20	Delete supplier	41
5.21	POS page	41
5.22	Adding items	42
5.23	Payments page	42

LIST OF TABLES

Table No.	Title	Page No
3.1	Transaction details table	16
3.2	Customer table	16
3.3	Employee table	16
3.4	User table	16
3.5	Transaction table	16
3.6	Product table	16
3.7	Returns table	16
3.8	Supplier table	16
3.9	Location table	17
3.10	Category table	17
3.11	Job table	17

Chapter 1

INTRODUCTION

The computer plays an important role in our daily life. Anything we want we can get only one mouse click. Speed, reliability and accuracy of the computer make it a powerful tool for different purposes. A very important and basic need of today's modern business world is the quick availability and processing of information using a computer. One can easily get the type of required information within a fraction of a second.

The vehicle inventory provisional system is developed for sale/purchase organization. This system manages customer, product, sales order, and dealer and purchase order. It provides a platform to the trader to automate their record keeping. The purpose of this project is to develop an application program to reduce the manual work for managing inventory, product, customer, dealer and orders. This project is built to help admin in managing his organization. Admin can store customer details in the database. Next time when customer comes in then there is no need to enter his details again. It can selected from the list displaying all customer

The project that I have taken is also in this category which is used in our daily life whenever we want to check availability of spare parts we can easily get them. "VEHICLE INVENTORY MANAGEMENT SYSTEM" has been designed to computerize the following functions that are performed by the system:

- Admin can store all customer and dealer record.
- This application keeps tracks of all the products being sold in the organization.
- This system provides facility of making sales and purchase order.
- All products, customer and dealer record is maintained as master data. User can selected these from display list while making sales order and purchase order.

1.1 OBJECTIVES

The main reason to build this project is to provide the seller a in-look of the shop. This project will guide the owner to the products available in the shop and which are required to order from the sellers. By this, the shopkeeper doesn't have to keep track of the records manually. The more comfortable users of computerized systems were mostly users with a fair accounting knowledge. In some cases business owners left the entire system to the accountant who was more familiar with the system, a situation some said poses a business risk. Deductions from the requirements analysis show that ease of use and knowledge prerequisite are the main reasons why most SMEs are not able to adopt computerized systems fully for tracking and managing their sales processes. This situation makes them stick to old and sometimes wasteful methods of tracking sales. The effect of this is loss of profits due to poor record keeping. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

- Admin will have complete control over the system.
- Only specific permissions are provided to specific parts of the project keeping the work done in order and also avoiding the misuse of info.
- It reduces the work done instead of looking for a suppliers every time. The searching of parts has been made quite simple as all the details of the spare parts can be obtained by simply keying the requirement of the user.
- Similarly, stock maintenance and updating can also be accomplished by the admin. These details are also being automatically updated in the database, thus keeping database up-to-date.
- The admin can update the details of all the customers registered by them.
- The users make payments when they want to buy the product by details.

1.2 SCOPE

Information about the specific subject range, format, or date range a particular specialized database covers is called its scope.

PERFORMANCE:

Manual handling of the record is time consuming and highly prone to error. Hence to improve the performance, computerized and user-friendly system is undertaken.

EFFICIENCY:

Efficiency of the system is a basic need. So, whenever the new user submits his/her details, it has to be updated automatically. It works on all platforms like Mac, Windows, Linux, iOS, Android, etc.

CONTROL:

The complete control is under the admin who has the authority to access, and illegal access is not permitted.

SECURITY:

This is the main criteria of the proposed system. Since illegal access may cause unnecessary actions, so security has been enforced. Here, in this system, we are using 256 Bit AES encryption (256 Bit Advanced Encryption Standard).

Chapter 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE REQUIREMENTS

- A local system (laptop/pc)
- Processor(i3/i5) any gen
- SSD 256 Gb (more than sufficient)
- RAM 8 Gb (more than sufficient)
- Clock speed 1.0 GHz and Cache 6MB .

Recommended Requirements:

- PROCESSOR: i9 10th gen
- RAM: 32GB
- HARD DISK: 4TB

2.2 SOFTWARE REQUIREMENTS

- Windows 10 OS
- XAMPP Server (Localhost)
- Google Chrome Web Browser

2.3 INFORMATION ON LANGUAGES USED:

HTML:

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

CSS:

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

JS:

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just in-time compiled, and multi-paradigm. It has curlybracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

SQL:

It is a relational database whose components (tables, forms, queries) are linked (related). The linkages between database components are created by making relationship links between them. The relationship can be:

- One component and another(one-one relationship)
- One component related to several other components(one-many)
- Several database components(many-many)

Creation of relationships between the database components reduces data redundancies and enhances ease of access of the information.

PHP:

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

2.4 REQUIREMENT ANALYSIS

The Requirements analysis was done to ascertain the type of sales management systems used by the SMEs and users experiences with the existing systems. This process led to the identification of two major types of systems namely; manual systems and computerized systems.

In response to questions about which of the two types of systems was preferred most of the businesses using the manual systems where daily sales books are maintained chose the computerized systems as most effective and convenient but cited factors such as difficulty of use, cost of acquiring full package and technical functions as reasons for not using them. It was also noted that some of the users of computerized systems also kept daily sales books, explaining that their sales persons needed a more formal training to be able to use the system effectively and for that matter it was convenient to keep a daily sales book to help identify error entries and sales misrepresentations.

The more comfortable users of computerized systems were mostly users with a fair accounting knowledge. In some cases business owners left the entire system to the accountant who was more familiar with the system, a situation some said possess a business risk. Deductions from the requirements analysis show that ease of use and knowledge prerequisite are the main reasons why most SMEs are not able to adopt computerized systems fully for tracking and managing their sales processes. This situation makes them stick to old and sometimes wasteful methods of tracking sales. The effect of this is loss of profits due to poor record keeping. Hence we sought to build a sales management system with friendly and interactive user interfaces which is easy to learn and use. However, the design does not compromise the security or integrity of business processes, and is intended to streamline transactions to eliminate user level errors such as misrepresentation of sales, uncaptured transactions and error entries.

2.5 FEASIBILITY STUDY

- A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.
- Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering blindly into risky businesses.
- Types of Feasibility Study:

- **Technical Feasibility:**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently; this project is not technically feasible.

- **Economic Feasibility:**

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

- **Legal Feasibility:**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project

was not feasible right from the beginning.

▪ **Scheduling Feasibility:**

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

▪ **Operational Feasibility:**

Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

2.6 LIMITATIONS:

- The process of gathering information and the record is time consuming. Since the system, response time is high.
- Only the person who manages it regularly can handle it in proper direction.
- The system allows one participant to register for only one event. A single user can't get registered for more than one event

Chapter 3

DESIGN

3.1 Description of Vehicle Inventory Management System

Vehicle Inventory Management System, is a management system having 1 parts ADMIN. Each customer of a particular requirement is register her with their details .

- The details of customer is stored into the customer tables respective with all tables.
- Each entity (Customer, Employee, Spare parts, Transaction, Suppliers, POS) contains primary key and unique keys.
- The entites like Customer, Employee, POS and Transaction has been binded with primary key.
- There is one-to-one and one-to-many relationships available
- All the entities are normalized and reduce duplicity of records.
- Indexing is implemented on each tables of Vehicle Inventory Management System tables for fast query execution.

Owners is the admin who is in charge of the system. The main objective of this project is that, presently we track of the parts that required based on the customer interest and updating the stock

Since technology is advancing day to day, we thought of an idea to digitalize the vehicle inventory management system since we have all facilities which are required to do so. Databases are perfect for implementing such systems which would be helpful for maintaining the spare parts details.

So, we could use this system to view and store all details about the spare parts ,customers and transaction details in the databases and the admin can login to view their status, customer details, profile info etc.

We have created some webpages where they can enter the details or data wherever required and store them in databases and retrieve it in specified part of your webpage. We can store data by creating various relations/tables. These tables would be interconnected with each other. So, we can write MYSQL queries to perform required actions through the system.

STEP 1:**ENTITY:**

- customer
- category
- employee
- job
- product
- supplier
- transaction
- location

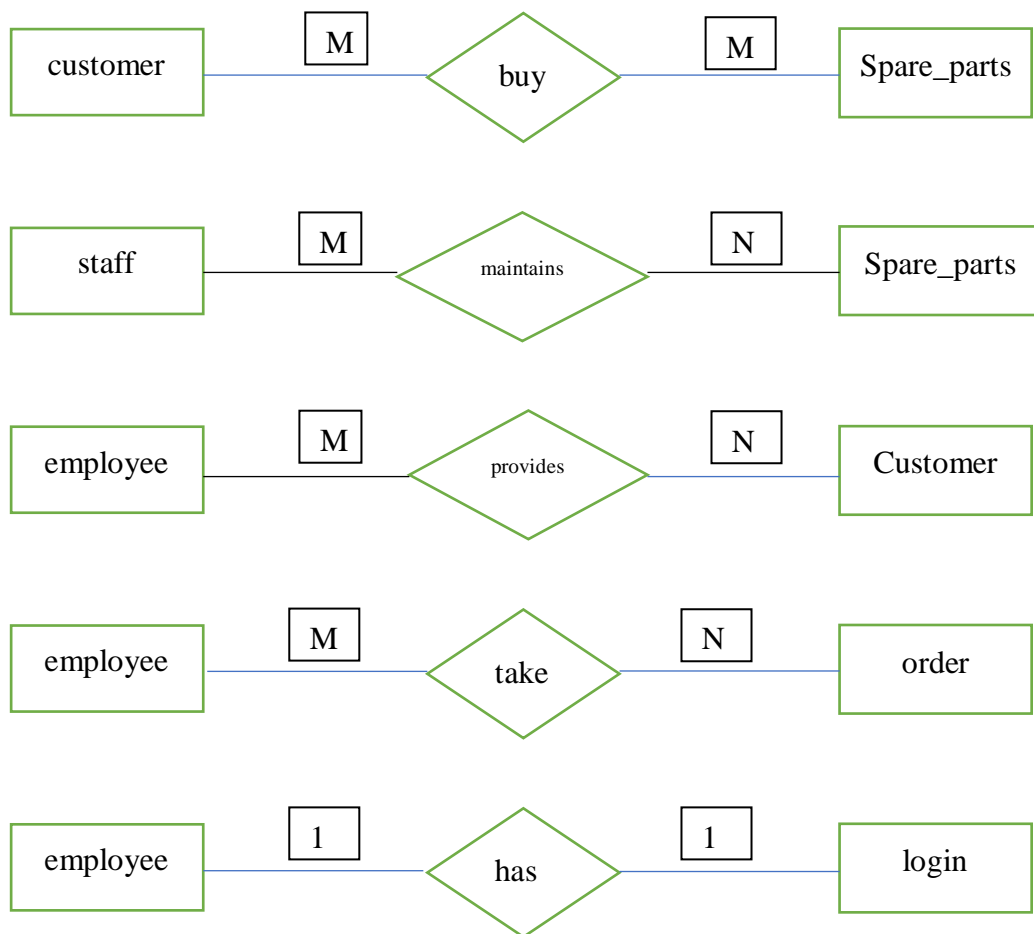
STEP 2:**RELATIONS:**

Figure 3.1: Relation between the existing tables in the database

STEP 3:**KEY ATTRIBUTES:**

- customer(id)
- category (id)
- supplier(id)
- transaction(id)
- employee (id)
- admin(id)
- product(id)

STEP 4:**OTHER ATTRIBUTES:**

- category (category_id , cname)
- customer (cust_id , first_name , last_name , phone_number)
- employee (employee_id , first_name , last_name , gender , email , phone_number , job_id , location_id)
- job (job_id , job_title)
- location (location_id , provinces , city)
- product (product_id , product_code , name , description , qty_stock , on_hand , category_id , supplier_id , date_stock_in)
- supplier (supplier_id , company_name , location_id , phone_number)
- transaction (trans_id , cust_id , numofitems , subtotal , lessvat , netvat , addvat , grandtotal, cash , date , trans_d_id)
- transaction_details (id, trans_d_id , products , qty , price , employee_name , role)
- type (type_id , type)
- user (id , employee_id , username , password , type_id)

STEP 5:**SCHEMA DIAGRAM :**

Category

<u>category_id</u>	cname
--------------------	-------

Customer

<u>cust_id</u>	first_name	last_name	phone_number
----------------	------------	-----------	--------------

Employee

<u>employee_id</u>	first_name	last_name	gender	email	phone_number	job_id	location_id
--------------------	------------	-----------	--------	-------	--------------	--------	-------------

Job

<u>job_id</u>	job_title
---------------	-----------

Location

<u>location_id</u>	provinces	city
--------------------	-----------	------

Product

<u>product_id</u>	<u>product_code</u>	name	qty_stock	on_hand	category_id	supplier_id
-------------------	---------------------	------	-----------	---------	-------------	-------------

Supplier

<u>supplier_id</u>	company_name	location_id	phone_number
--------------------	--------------	-------------	--------------

Transaction

<u>trans_id</u>	<u>cust_id</u>	numofitems	subtotal	grandtotal	cash	date	<u>trans d id</u>
-----------------	----------------	------------	----------	------------	------	------	-------------------

ER DIAGRAM :

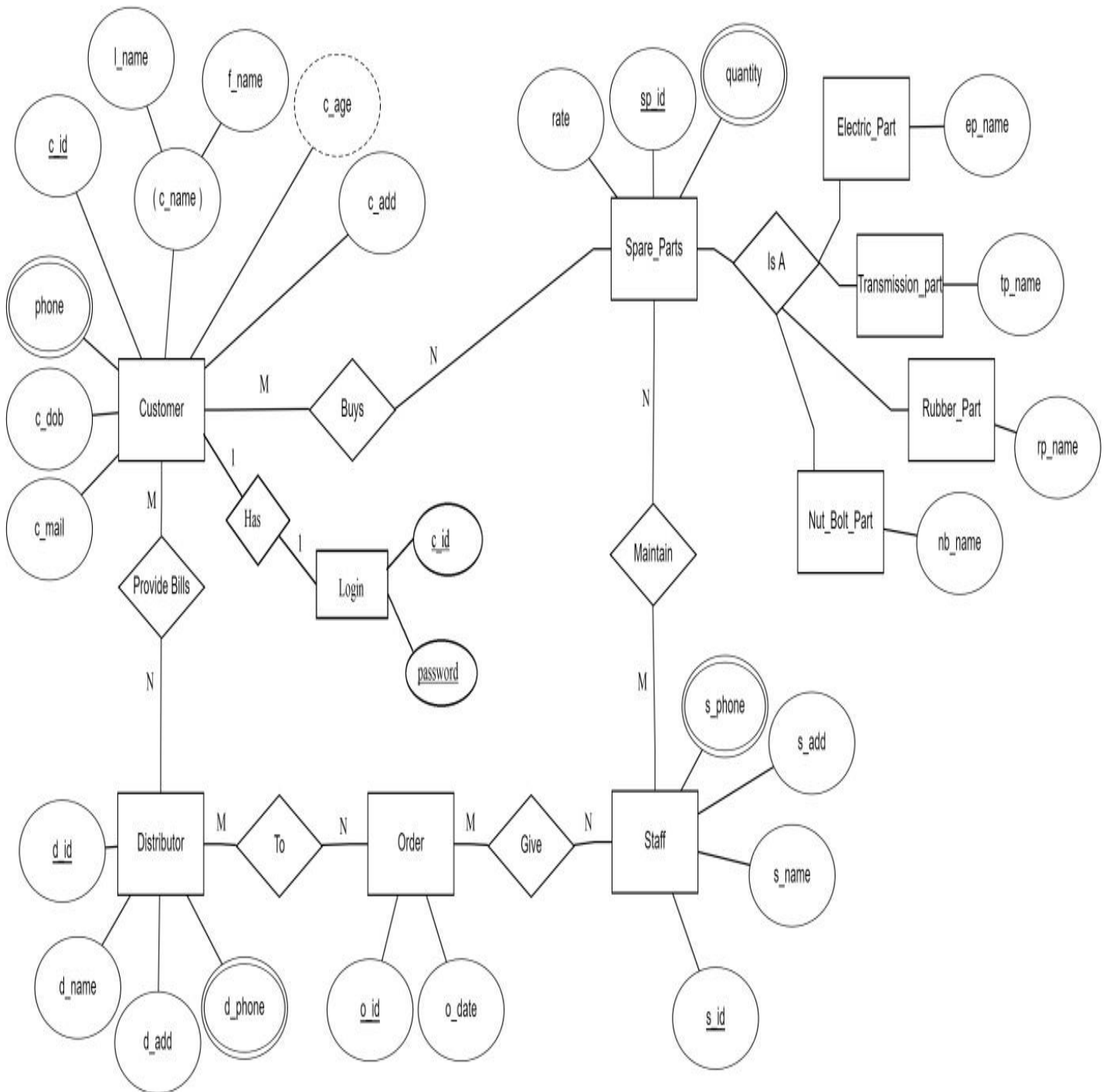


Figure 3.3: ER Diagram for the database

3.2 NORMALISATION

The basic Objectives of normalization is to reduce redundancy, which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of data stored. Relations are normalized so that when a relation in the database is to be altered during the lifetime of the database, information is not. The type of alterations normally needed for relation is:

- Insertion of new data values to relation. This should be possible without being forced to leave blank fields for some attributes.
- Deletion of a tuple, namely, a row of a relation. This should be possible without losing vital information unknowingly.

Functional Dependency:

As the concept of dependency is very important, it is essential that it should be understood first and then proceeded to the idea of normalization. There is no fool-proof algorithmic method of identifying dependency.

Properties of normalized relations:

Ideal relations after normalization should have the following properties:

- No data values should be duplicated in different rows unnecessarily.
- A value must be specified (and required) for every attribute in a row.
- Each relation should be self-contained. In other words, if a row from a relation is deleted, important information should not be accidentally lost.
- When a row is added to a relation, other relations in the database should not be affected.
- A value of an attribute in a tuple may be changed independent of other tuples in the relation and other relations.

Consider the table (refer to the schema diagram on figure 3.2).

- The prime attributes identified are the attributes which is part of candidate key.
- The non-prime attributes are not part of primary key.
- When the table is split into two tables having the following attributes:

Before Normalization-

Attributes rules was present in the event table before the normalization. (see tables below)

Rules attribute stores all the description about the event with multiple points which was containing multiple values.

`SELECT * FROM `type``

TYPE_ID	TYPE
1	Admin
2	User

`SELECT * FROM `transaction_details``

ID	TRANS_D_ID	PRODUCTS	QTY	PRICE	EMPLOYEE	ROLE
21	111141641	headlights	1	2500	pavan	Manager
22	111151334	wires	1	300	pavan	Manager
23	111151334	headlights	1	2500	pavan	Manager

`SELECT * FROM `supplier``

SUPPLIER_ID	COMPANY_NAME	LOCATION_ID	PHONE_NUMBER
17	TATA	159	9873647484
18	TOYOTA	160	9873654783
19	Mahindra	161	8765904753
20	Suzuki	162	8973645632
21	so	166	2345675678
23	DFB	168	2345675678
25	DFB	171	2345675678
26	ww	172	2345675678

`SELECT * FROM `employee``

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	GENDER	EMAIL	PHONE_NUMBER	JOB_ID	LOCATION_ID
5	Elena	James	Female	elenajames@gmail.com	7406751843	1	163
6	Dheekhsa	S	Female	dheekshas@yahoo.com	8050399259	2	164
7	Indira	Dev	Male	indira@gmail.com	6969696969	2	165
8	Katherine	Pierce	Female	katherine@gmail.com	9812345678	2	173

`SELECT * FROM `product``

PRODUCT_ID	PRODUCT_CODE	NAME	DESCRIPTION	QTY_STOCK	ON_HAND	PRICE	CATEGORY_ID	SUPPLIER_ID	DATE_STOCK_IN
1001	1	wires		1	1	300	4	19	
1002	2	headlights		1	1	2500	1	17	
1003	3	starters		1	1	500	1	18	
1004	4	fuel injection pipes		1	1	700	3	17	
1005	5	gears		1	1	3500	4	19	
1006	5	U-J cross		1	1	1500	4	17	
1007	6	Engine mountings		1	1	1200	3	19	
1008	7	Nuts		1	1	150	2	18	
1009	8	T-bolts		1	1	130	2	17	

`SELECT * FROM `users``

ID	EMPLOYEE_ID	USERNAME	PASSWORD	TYPE_ID
1	1	admin1	6C7CA345F63F835CB353FF15BD6C5E052EC08E7A	1
2	5	admin2	315F166C5ACA63A157F7D41007675CB44A948B33	1
3	6	admin3	33AAB3C7F01620CADE108F488CFD285C0E62C1EC	2
4	7	admin4	EA053D11A8AAD1CCF8C18F9241BAEB9EC47E5D64	2

SELECT * FROM `job`		SELECT * FROM `location`		
JOB_ID	JOB_TITLE	LOCATION_ID	PROVINCE	CITY
1	Manager	113	Bangalore	Gotigere
2	Cashier	159	Bangalore	Hulimavu
		160	Bangalore	Whitefield
		161	Bangalore	Domlur
		162	Delhi	KR Puram
		163	Pune	Katraj
		164	Nagpur	Narkhed
		165	Amravati	Dhamangaon
		166	pune	kondhwa
		168	pune	kondhwa
		171	amravati	dharni
		172	nagpur	kuhi
		173	Bangalore	Bannerghatta

Figure 3.4: Table.

1NF

- Each table cell should contain a single value.
- Each record needs to be unique.

attributes is made into a different table by this each table contains single values and each record is unique. Therefore, the given tables are in 1NF (1st Normal Form)

2NF

- Be in 1NF
- Single Column Primary Key

attributes is made into a different table by this each column will be having single column primary key. Therefore, the given tables are in 2NF (2nd Normal Form)

3NF

- Be in 2NF
- Rule 2- Has no transitive functional dependencies

attributes is made into a different table by this the two tables does not have transitive functional dependencies. Therefore, the given tables are in 3NF (3rd Normal Form)

BCNF

- Be in 3NF
- Should not have more than one Candidate Key

Therefore, the given tables are in BCNF (BCNF Normal Form)

3.3 PROCEDURES

Stored Procedures are created to perform one or more DML operations on Database. It is nothing but the group of SQL statements that accepts some input in the form of parameters and performs some task and may or may not returns a value.

The most important part is parameters. Parameters are used to pass values to the Procedure. There are 3 different types of parameters, they are as follows:

IN:

This is the Default Parameter for the procedure. It always receives the values from calling program.

OUT:

This parameter always sends the values to the calling program.

IN OUT:

This parameter performs both the operations. It receives value from as well as sends the values to the calling program.

Chapter 4

IMPLEMENTATION AND CODING

4.1 SOURCE CODE

LOGIN PAGE

```
<?php require('session.php');?>
<?php if(logged_in()){ ?>
    <script type="text/javascript">
        window.location = "index.php";
    </script>
<?php
} ?>
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Vehicle Inventory Management</title>

    <!-- Custom fonts for this template-->
    <link href="../vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="../css/sb-admin-2.min.css" rel="stylesheet">

</head>

<body style="background-image: url('https://media.istockphoto.com/id/1394354711/photo/car-spare-parts.jpg?b=1&s=170667a&w=0&k=20&c=39SN1JSHe2tsTb7RjBprLxGKO4DQ0WEMk7ocjCe7XSs=');">
    <div class="container" style="color:transparent">

        <!-- Outer Row -->
```

```
<div class="row justify-content-center">

  <div class="card o-hidden border-0 shadow-lg my-5 p-2">
    <div class="card-body p-0">
      <!-- Nested Row within Card Body -->

        <h1 class="h4 text-gray-900 mb-4">Welcome to Vehicle Inventory
Management</h1>
      </div>
      <form class="user" role="form" action="processlogin.php" method="post">
        <div class="form-group">
          <input class="form-control form-control-user" placeholder="Username"
name="user" type="text" autofocus>
        </div>
        <div class="form-group">
          <input class="form-control form-control-user" placeholder="Password"
name="password" type="password" value="">
        </div>

        <button class="btn btn-primary btn-user btn-block" type="submit"
name="btnlogin">Login</button>
        <hr>
        <!-- <div class="text-center">
          <a class="small" href="register.php">Create an Account!</a>
        </div> -->
      </form>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- Bootstrap core JavaScript-->
<script src="../../vendor/jquery/jquery.min.js"></script>
<script src="../../vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="../../vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="../../js/sb-admin-2.min.js"></script>

</body>
</html>
```

DATABASE DESIGN

```
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 11, 2020 at 06:43 AM
-- Server version: 10.4.14-MariaDB
-- PHP Version: 7.4.9

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `spare_parts`
--
-- -----
--
-- Table structure for table `category`
--
CREATE TABLE `category` (
  `CATEGORY_ID` int(11) NOT NULL,
  `CNAME` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `category`
--
```

```
INSERT INTO `category` (`CATEGORY_ID`, `CNAME`) VALUES
```

```
(1, 'Electric_parts'),
```

```
(2, 'Metal_parts'),
```

```
(3, 'Rubber_parts'),
```

```
(4, 'Tranmission_parts');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `customer`
```

```
--
```

```
CREATE TABLE `customer` (
```

```
  `CUST_ID` int(11) NOT NULL,
```

```
  `FIRST_NAME` varchar(50) DEFAULT NULL,
```

```
  `LAST_NAME` varchar(50) DEFAULT NULL,
```

```
  `PHONE_NUMBER` varchar(11) DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `customer`
```

```
--
```

```
INSERT INTO `customer` (`CUST_ID`, `FIRST_NAME`, `LAST_NAME`,  
`PHONE_NUMBER`) VALUES
```

```
(1, 'Vaishnavi', 'G', '7406751843'),
```

```
(2, 'Ram', 'S', '8050399259'),
```

```
(3, 'Gagan', 'S', '9980631823'),
```

```
(4, 'Indu', 'K V', '8088072712'),
```

```
(5, 'Mr', 'Stark', '9060842578');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `employee`
```

```
--
```

```
CREATE TABLE `employee` (
```

```
  `EMPLOYEE_ID` int(11) NOT NULL,
```

```
  `FIRST_NAME` varchar(50) DEFAULT NULL,
```

```
  `LAST_NAME` varchar(50) DEFAULT NULL,
```

```
  `GENDER` varchar(50) DEFAULT NULL,
```

```
`EMAIL` varchar(100) DEFAULT NULL,
`PHONE_NUMBER` varchar(11) DEFAULT NULL,
`JOB_ID` int(11) DEFAULT NULL,
`LOCATION_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `employee`
--

INSERT INTO `employee` (`EMPLOYEE_ID`, `FIRST_NAME`, `LAST_NAME`,
`GENDER`, `EMAIL`, `PHONE_NUMBER`, `JOB_ID`, `LOCATION_ID`) VALUES
(1, 'Admin', '', 'Male', 'admin@spareparts.in', '9060842578', 1, 2),
(2, 'Elena', 'James', 'Female', 'elenajames@gmail.com', '7406751843', 1, 1),
(3, 'Dheekhsa', 'S', 'Female', 'dheekshas@yahooemail.com', '8050399259', 2, 4),
(4, 'Indira', 'Dev', 'Male', 'indira@gmail.com', '6969696969', 2, 5);

-----

--
-- Table structure for table `job`
--

CREATE TABLE `job` (
  `JOB_ID` int(11) NOT NULL,
  `JOB_TITLE` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `job`
--

INSERT INTO `job` (`JOB_ID`, `JOB_TITLE`) VALUES
(1, 'Manager'),
(2, 'Cashier');

-----

--
-- Table structure for table `location`
--

CREATE TABLE `location` (
  `LOCATION_ID` int(11) NOT NULL,
  `PROVINCE` varchar(100) DEFAULT NULL,
```

```
`CITY` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `location`
--
INSERT INTO `location` (`LOCATION_ID`, `PROVINCE`, `CITY`) VALUES
(1, 'Bangalore', 'Gottigere'),
(2, 'Bangalore', 'Arekere'),
(3, 'Bangalore', 'Koramangala'),
(4, 'Bangalore', 'MG Road'),
(5, 'Ramanagara', 'Kanakapura'),
(6, 'Ramanagara', 'Satnur'),
(7, 'Mysore', 'Nanjangud'),
(8, 'Mysore', 'Nanjangud');
-----
--
-- Table structure for table `product`
--
CREATE TABLE `product` (
  `PRODUCT_ID` int(11) NOT NULL,
  `PRODUCT_CODE` varchar(20) NOT NULL,
  `NAME` varchar(50) DEFAULT NULL,
  `DESCRIPTION` varchar(250) NOT NULL,
  `QTY_STOCK` int(50) DEFAULT NULL,
  `ON_HAND` int(250) NOT NULL,
  `PRICE` int(50) DEFAULT NULL,
  `CATEGORY_ID` int(11) DEFAULT NULL,
  `SUPPLIER_ID` int(11) DEFAULT NULL,
  `DATE_STOCK_IN` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `product`
--
```



```
INSERT INTO `product` (`PRODUCT_ID`, `PRODUCT_CODE`, `NAME`,  
`DESCRIPTION`, `QTY_STOCK`, `ON_HAND`, `PRICE`, `CATEGORY_ID`,  
`SUPPLIER_ID`, `DATE_STOCK_IN`) VALUES
```

```
(1001, '1', 'wires', '', 1, 1, 300, 4, 19, ''),  
(1002, '2', 'headlights', '', 1, 1, 2500, 1, 17, ''),  
(1003, '2', 'headlights', '', 1, 1, 2500, 1, 17, ''),  
(1004, '3', 'starters', '', 1, 1, 500, 1, 18, ''),  
(1005, '3', 'starters', '', 1, 1, 500, 1, 18, ''),  
(1006, '3', 'starters', '', 1, 1, 500, 1, 18, ''),  
(1007, '4', 'fuel injection pipes', '', 1, 1, 700, 3, 17, ''),  
(1008, '4', 'fuel injection pipes', '', 1, 1, 700, 3, 17, ''),  
(1009, '4', 'fuel injection pipes', '', 1, 1, 700, 3, 17, ''),  
(1010, '4', 'fuel injection pipes', '', 1, 1, 700, 3, 17, ''),  
(1011, '5', 'gears', '', 1, 1, 3500, 4, 19, ''),  
(1012, '5', 'gears', '', 1, 1, 3500, 4, 19, ''),  
(1013, '5', 'gears', '', 1, 1, 3500, 4, 19, ''),  
(1014, '5', 'gears', '', 1, 1, 3500, 4, 19, ''),  
(1015, '5', 'gears', '', 1, 1, 3500, 4, 19, ''),  
(1016, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1017, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1018, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1019, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1020, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1021, '6', 'U-J cross', '', 1, 1, 1500, 4, 17, ''),  
(1022, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1023, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1024, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1025, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1026, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1027, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1028, '7', 'Engine mountings', '', 1, 1, 1200, 3, 19, ''),  
(1029, '8', 'Nuts', '', 1, 1, 150, 2, 18, ''),  
(1030, '8', 'Nuts', '', 1, 1, 150, 2, 18, ''),  
(1031, '8', 'Nuts', '', 1, 1, 150, 2, 18, ''),  
(1032, '8', 'Nuts', '', 1, 1, 150, 2, 18, ''),
```

```
(1033, '8', 'Nuts', "", 1, 1, 150, 2, 18, ""),
(1034, '8', 'Nuts', "", 1, 1, 150, 2, 18, ""),
(1035, '8', 'Nuts', "", 1, 1, 150, 2, 18, ""),
(1036, '8', 'Nuts', "", 1, 1, 150, 2, 18, ""),
(1037, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1038, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1039, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1040, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1041, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1042, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1043, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1044, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1045, '9', 'T-bolts', "", 1, 1, 130, 2, 17, ""),
(1046, '9', 'T-bolts', "", 1, 1, 130, 2, 17, "");
```

```
-- -----
```

```
--
```

```
-- Table structure for table `supplier`
```

```
--
```

```
CREATE TABLE `supplier` (
  `SUPPLIER_ID` int(11) NOT NULL,
  `COMPANY_NAME` varchar(50) DEFAULT NULL,
  `LOCATION_ID` int(11) NOT NULL,
  `PHONE_NUMBER` varchar(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `supplier`
```

```
--
```

```
INSERT INTO `supplier` (`SUPPLIER_ID`, `COMPANY_NAME`, `LOCATION_ID`,
`PHONE_NUMBER`) VALUES
(1, 'TATA', 1, '9873647484'),
(2, 'TOYOTA', 3, '9873654783'),
(3, 'Mahindra', 6, '8765904753'),
(4, 'Suzuki', 1, '8973645632');
```

```
-- -----
```

```
--
```

-- Table structure for table `transaction`

--

```
CREATE TABLE `transaction` (  
  `TRANS_ID` int(50) NOT NULL,  
  `CUST_ID` int(11) DEFAULT NULL,  
  `NUMOFITEMS` varchar(250) NOT NULL,  
  `SUBTOTAL` varchar(50) NOT NULL,  
  `LESSVAT` varchar(50) NOT NULL,  
  `NETVAT` varchar(50) NOT NULL,  
  `ADDVAT` varchar(50) NOT NULL,  
  `GRANDTOTAL` varchar(250) NOT NULL,  
  `CASH` varchar(250) NOT NULL,  
  `DATE` varchar(50) NOT NULL,  
  `TRANS_D_ID` varchar(250) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Dumping data for table `transaction`

--

```
INSERT INTO `transaction` (`TRANS_ID`, `CUST_ID`, `NUMOFITEMS`, `SUBTOTAL`,  
`LESSVAT`, `NETVAT`, `ADDVAT`, `GRANDTOTAL`, `CASH`, `DATE`,  
`TRANS_D_ID`) VALUES  
(1, 2, '1', '2,500.00', '267.86', '2,232.14', '267.86', '2,500.00', '2500', '2020-11-11 04:15 am',  
'111141641'),  
(2, 1, '2', ' ', ' ', ' ', ' ', '2,800.00', '2800', '<br />\r\n<b>Notice</b>: Undefined variable: today ',  
'111151334');
```

-- -----

--

-- Table structure for table `transaction_details`

--

```
CREATE TABLE `transaction_details` (  
  `ID` int(11) NOT NULL,  
  `TRANS_D_ID` varchar(250) NOT NULL,  
  `PRODUCTS` varchar(250) NOT NULL,  
  `QTY` varchar(250) NOT NULL,  
  `PRICE` varchar(250) NOT NULL,
```

```
`EMPLOYEE` varchar(250) NOT NULL,
`ROLE` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `transaction_details`
--
INSERT INTO `transaction_details` (`ID`, `TRANS_D_ID`, `PRODUCTS`, `QTY`, `PRICE`,
`EMPLOYEE`, `ROLE`) VALUES
(1, '111141641', 'headlights', '1', '2500', 'Lekhana', 'Manager'),
(2, '111151334', 'wires', '1', '300', 'Lekhana', 'Manager'),
(3, '111151334', 'headlights', '1', '2500', 'Lekhana', 'Manager');

-----
--
-- Table structure for table `type`
--
CREATE TABLE `type` (
  `TYPE_ID` int(11) NOT NULL,
  `TYPE` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `type`
--
INSERT INTO `type` (`TYPE_ID`, `TYPE`) VALUES
(1, 'Admin'),
(2, 'User');

-----
--
-- Table structure for table `users`
--
CREATE TABLE `users` (
  `ID` int(11) NOT NULL,
  `EMPLOYEE_ID` int(11) DEFAULT NULL,
  `USERNAME` varchar(50) DEFAULT NULL,
  `PASSWORD` varchar(50) DEFAULT NULL,
  `TYPE_ID` int(11) DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`ID`, `EMPLOYEE_ID`, `USERNAME`, `PASSWORD`, `TYPE_ID`)
VALUES
(1, 1, 'admin1', '6C7CA345F63F835CB353FF15BD6C5E052EC08E7A', 1),
(2, 2, 'admin2', '315F166C5ACA63A157F7D41007675CB44A948B33', 1),
(3, 3, 'admin3', '33AAB3C7F01620CADE108F488CFD285C0E62C1EC', 2),
(4, 4, 'admin4', 'EA053D11A8AAD1CCF8C18F9241BAEB9EC47E5D64', 2);

--

-- Indexes for dumped tables

--

-- Indexes for table `category`

--

ALTER TABLE `category`
  ADD PRIMARY KEY (`CATEGORY_ID`);

--

-- Indexes for table `customer`

--

ALTER TABLE `customer`
  ADD PRIMARY KEY (`CUST_ID`);

--

-- Indexes for table `employee`

--

ALTER TABLE `employee`
  ADD PRIMARY KEY (`EMPLOYEE_ID`),
  ADD UNIQUE KEY `EMPLOYEE_ID` (`EMPLOYEE_ID`),
  ADD UNIQUE KEY `PHONE_NUMBER` (`PHONE_NUMBER`),
  ADD KEY `LOCATION_ID` (`LOCATION_ID`),
  ADD KEY `JOB_ID` (`JOB_ID`);

--

-- Indexes for table `job`

--
```

```
ALTER TABLE `job`
  ADD PRIMARY KEY (`JOB_ID`);
--
-- Indexes for table `location`
--
ALTER TABLE `location`
  ADD PRIMARY KEY (`LOCATION_ID`);
--
-- Indexes for table `manager`
--
ALTER TABLE `manager`
  ADD UNIQUE KEY `PHONE_NUMBER` (`PHONE_NUMBER`),
  ADD KEY `LOCATION_ID` (`LOCATION_ID`);
--
-- Indexes for table `product`
--
ALTER TABLE `product`
  ADD PRIMARY KEY (`PRODUCT_ID`),
  ADD KEY `CATEGORY_ID` (`CATEGORY_ID`),
  ADD KEY `SUPPLIER_ID` (`SUPPLIER_ID`);
--
-- Indexes for table `supplier`
--
ALTER TABLE `supplier`
  ADD PRIMARY KEY (`SUPPLIER_ID`),
  ADD KEY `LOCATION_ID` (`LOCATION_ID`);
--
-- Indexes for table `transaction`
--
ALTER TABLE `transaction`
  ADD PRIMARY KEY (`TRANS_ID`),
  ADD KEY `TRANS_DETAIL_ID` (`TRANS_D_ID`),
  ADD KEY `CUST_ID` (`CUST_ID`);
--
```

```
-- Indexes for table `transaction_details`
--
ALTER TABLE `transaction_details`
  ADD PRIMARY KEY (`ID`),
  ADD KEY `TRANS_D_ID` (`TRANS_D_ID`) USING BTREE;
--
-- Indexes for table `type`
--
ALTER TABLE `type`
  ADD PRIMARY KEY (`TYPE_ID`);
--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`ID`),
  ADD KEY `TYPE_ID` (`TYPE_ID`),
  ADD KEY `EMPLOYEE_ID` (`EMPLOYEE_ID`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `category`
--
ALTER TABLE `category`
  MODIFY `CATEGORY_ID` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `customer`
--
ALTER TABLE `customer`
  MODIFY `CUST_ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
--
-- AUTO_INCREMENT for table `employee`
--
ALTER TABLE `employee`
```

```
MODIFY `EMPLOYEE_ID` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=5;  
  
--  
-- AUTO_INCREMENT for table `location`  
  
--  
ALTER TABLE `location`  
MODIFY `LOCATION_ID` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=9;  
  
--  
-- AUTO_INCREMENT for table `product`  
  
--  
ALTER TABLE `product`  
MODIFY `PRODUCT_ID` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=1047;  
  
--  
-- AUTO_INCREMENT for table `supplier`  
  
--  
ALTER TABLE `supplier`  
MODIFY `SUPPLIER_ID` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=5;  
  
--  
-- AUTO_INCREMENT for table `transaction`  
  
--  
ALTER TABLE `transaction`  
MODIFY `TRANS_ID` int(50) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=3;  
  
--  
-- AUTO_INCREMENT for table `transaction_details`  
  
--  
ALTER TABLE `transaction_details`  
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;  
  
--  
-- AUTO_INCREMENT for table `users`  
  
--  
ALTER TABLE `users`  
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
```



```
--  
-- Constraints for dumped tables  
--  
--  
-- Constraints for table `employee`  
--  
ALTER TABLE `employee`  
  ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`),  
  ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`JOB_ID`) REFERENCES `job` (`JOB_ID`);  
--  
-- Constraints for table `manager`  
--  
ALTER TABLE `manager`  
  ADD CONSTRAINT `manager_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`);  
--  
-- Constraints for table `product`  
--  
ALTER TABLE `product`  
  ADD CONSTRAINT `product_ibfk_1` FOREIGN KEY (`CATEGORY_ID`) REFERENCES `category` (`CATEGORY_ID`),  
  ADD CONSTRAINT `product_ibfk_2` FOREIGN KEY (`SUPPLIER_ID`) REFERENCES `supplier` (`SUPPLIER_ID`);  
--  
-- Constraints for table `supplier`  
--  
ALTER TABLE `supplier`  
  ADD CONSTRAINT `supplier_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`);  
--  
-- Constraints for table `transaction`  
--  
ALTER TABLE `transaction`
```

```
ADD CONSTRAINT `transaction_ibfk_3` FOREIGN KEY (`CUST_ID`) REFERENCES
`customer` (`CUST_ID`),
```

```
ADD CONSTRAINT `transaction_ibfk_4` FOREIGN KEY (`TRANS_D_ID`)
REFERENCES `transaction_details` (`TRANS_D_ID`);
```

```
--
```

```
-- Constraints for table `users`
```

```
--
```

```
ALTER TABLE `users`
```

```
ADD CONSTRAINT `users_ibfk_3` FOREIGN KEY (`TYPE_ID`) REFERENCES `type`
(`TYPE_ID`);
```

```
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Chapter 5

SCREENSHOTS

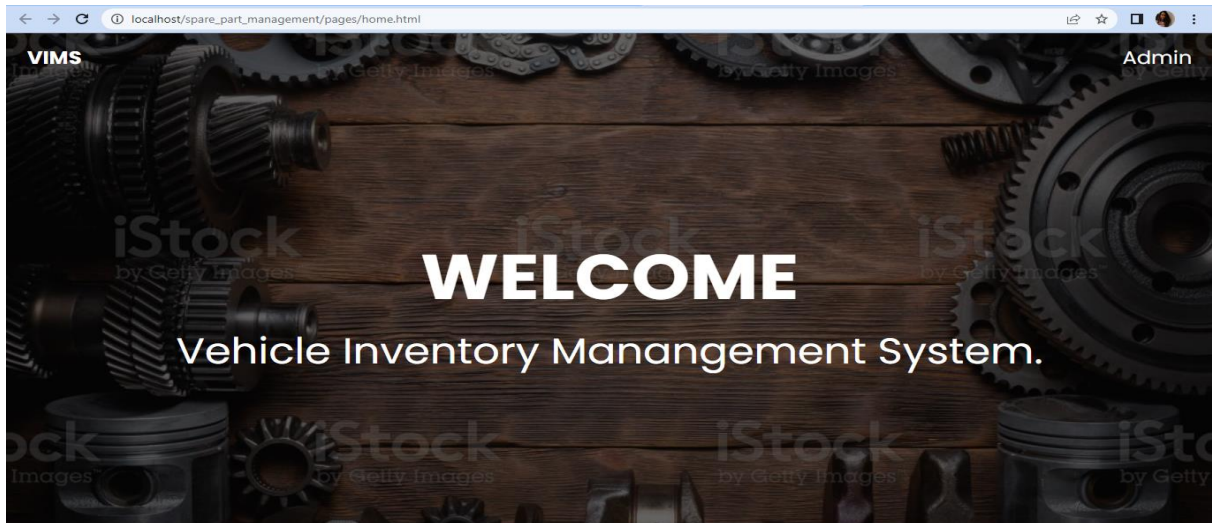


Fig. 5.1: Login Home Page

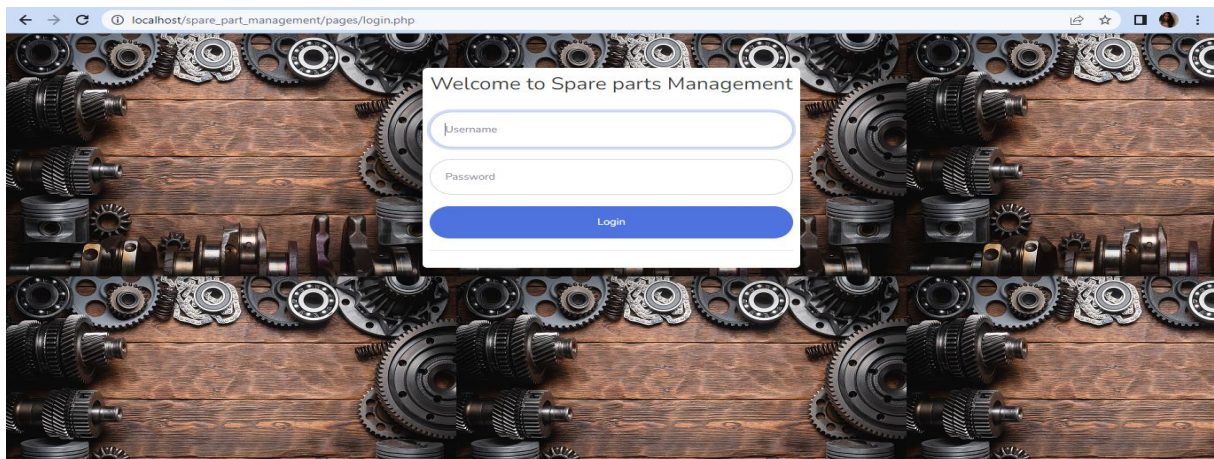


Fig. 5.2: Login Page

This is over login page. There are two types of login one for manager and one for cashier. The following screenshots is for manager login.

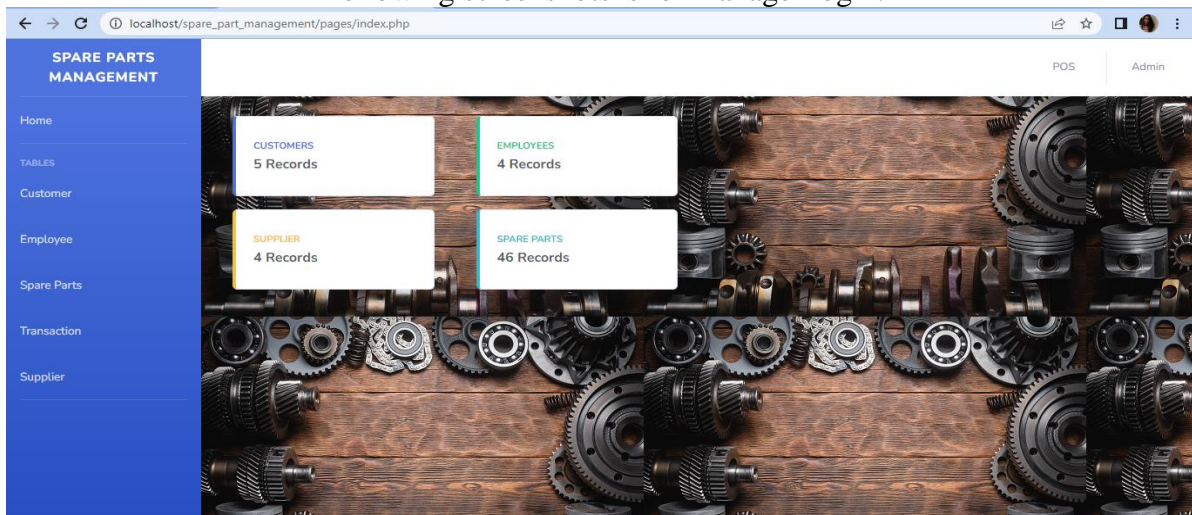
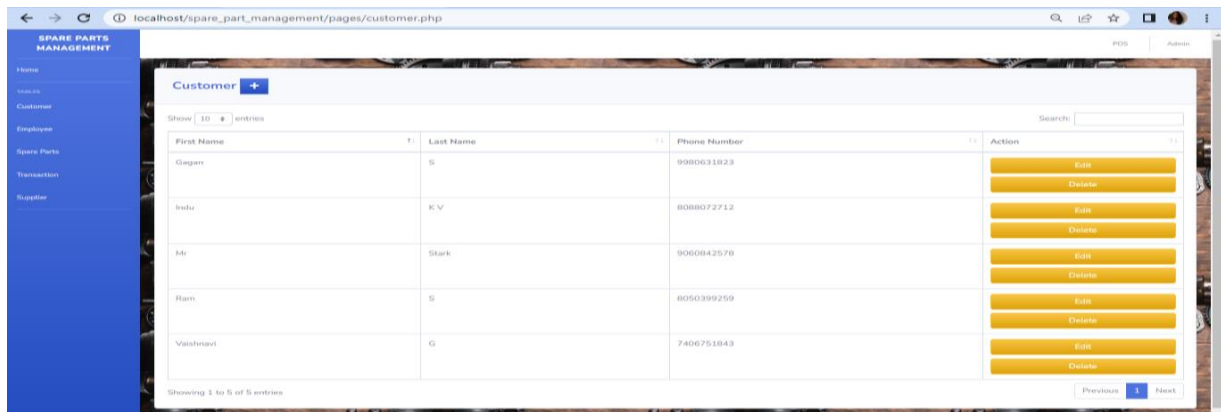


Fig. 5.3: Home page

This is our home page. After entering correct username and password the manager will redirect to this page

**Fig. 5.3: Customer page**

This is customer page. Here you will information about the entire customer who has buy product from you. When you click on edit, you can change the information about the customer.

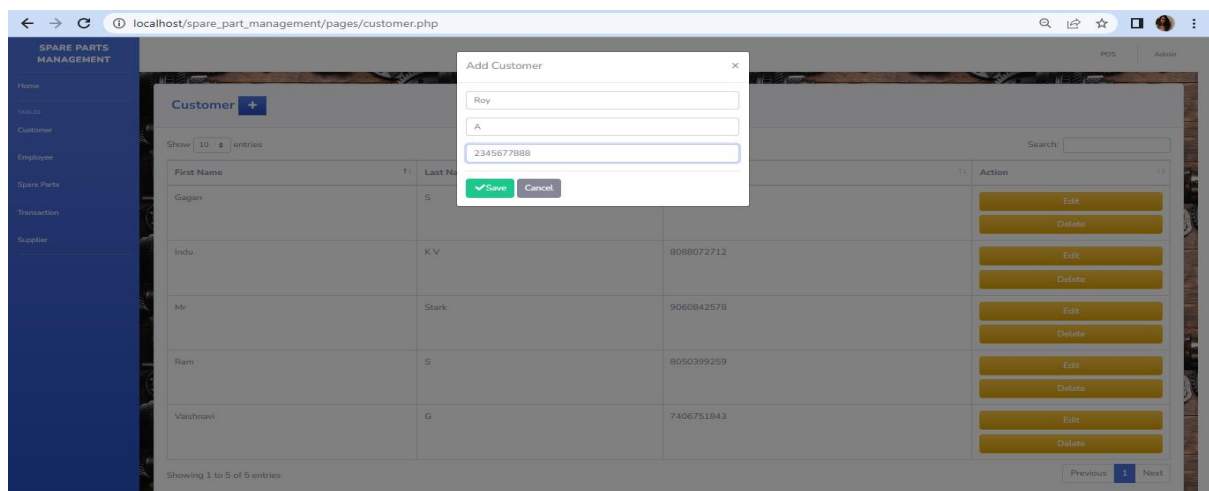
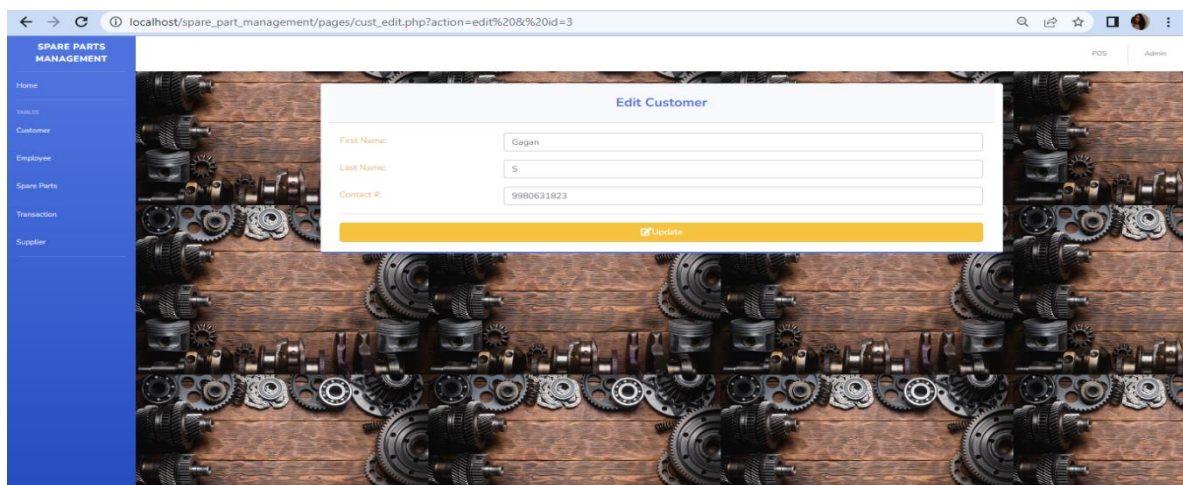
**Fig. 5.4: Add customer**

Fig. 5.5: Edit customer

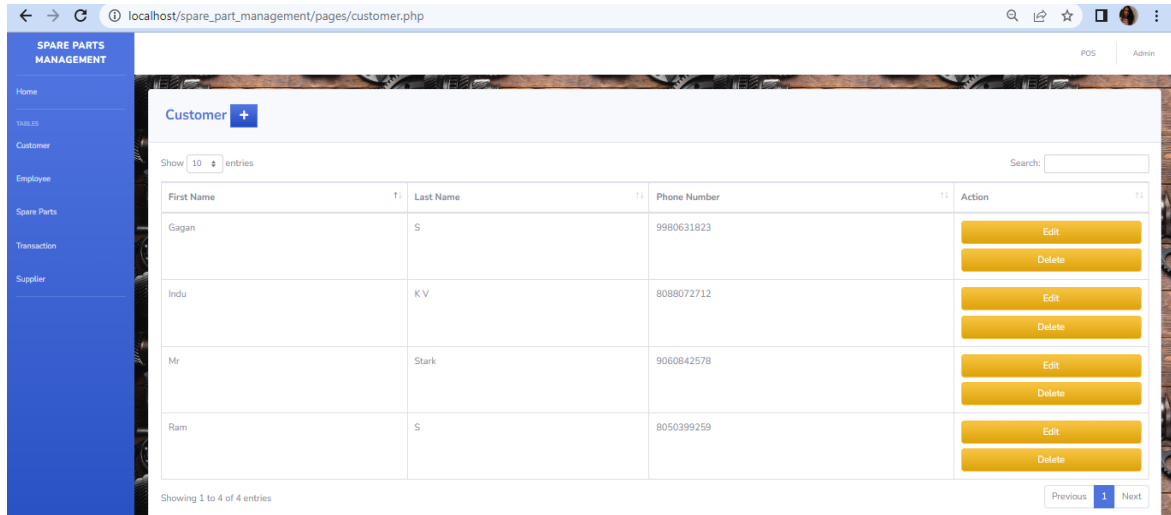


Fig. 5.6: Delete customer

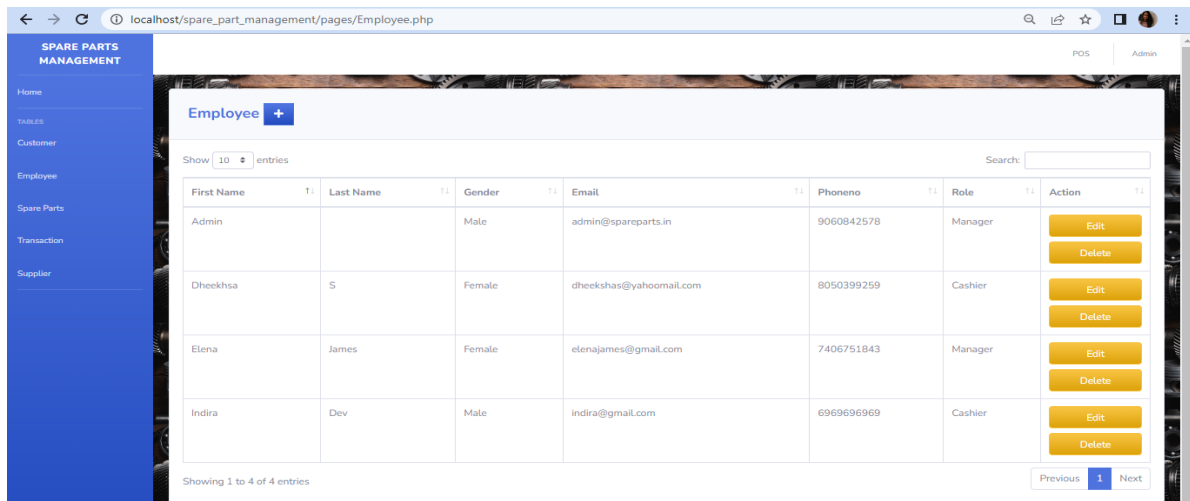


Fig. 5.7: Employee page

This is our employee page here you can add or remove employee name who are working with you or who was working with you.

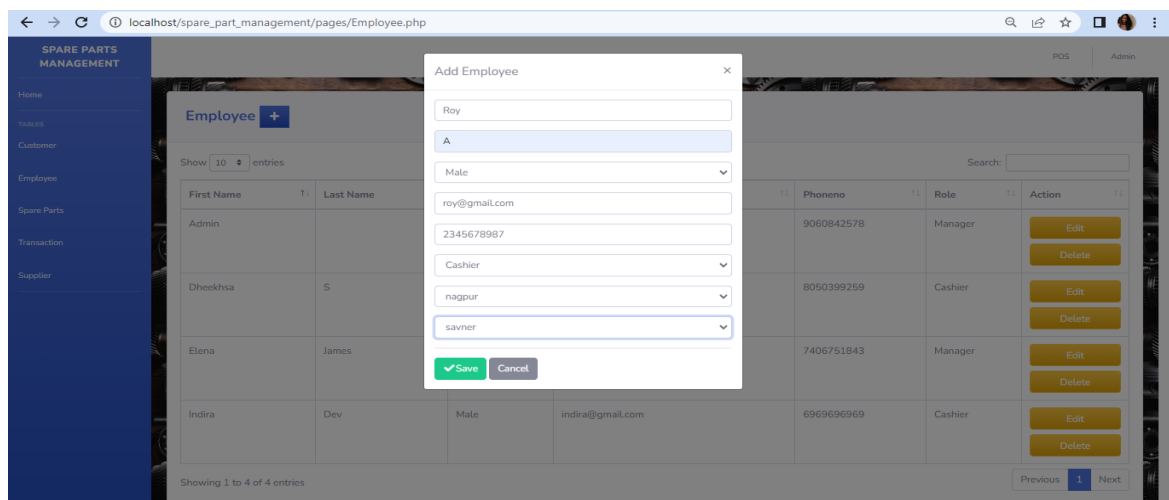


Fig. 5.8: Add employee

Edit Employee

First Name:

Last Name:

Gender:

Email:

Contact no.:

Role:

Province:

City:

Fig. 5.9: edit employee

Employee +

Show 10 entries

Search:

First Name	Last Name	Gender	Email	Phoneno	Role	Action
Admin		Male	admin@spareparts.in	9060842578	Manager	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Dheekhsa	S	Female	dheekshas@yahooemail.com	8050399259	Cashier	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Elena	James	Female	elenajames@gmail.com	7406751843	Manager	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Fig. 5.10: delete employee

Spare Part +

Show 10 entries

Search:

Spare part code	Name	Price	Category	Action
1	wires	300	Transmission_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	headlights	2500	Electric_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	starters	500	Electric_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	fuel injection pipes	700	Rubber_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	gears	3500	Transmission_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	U-j cross	1500	Transmission_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	Engine mountings	1200	Rubber_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
8	Nuts	150	Metal_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
9	T bolts	110	Metal_parts	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Fig. 5.11: Spare parts page

Here you add name of new spare part with their details or remove

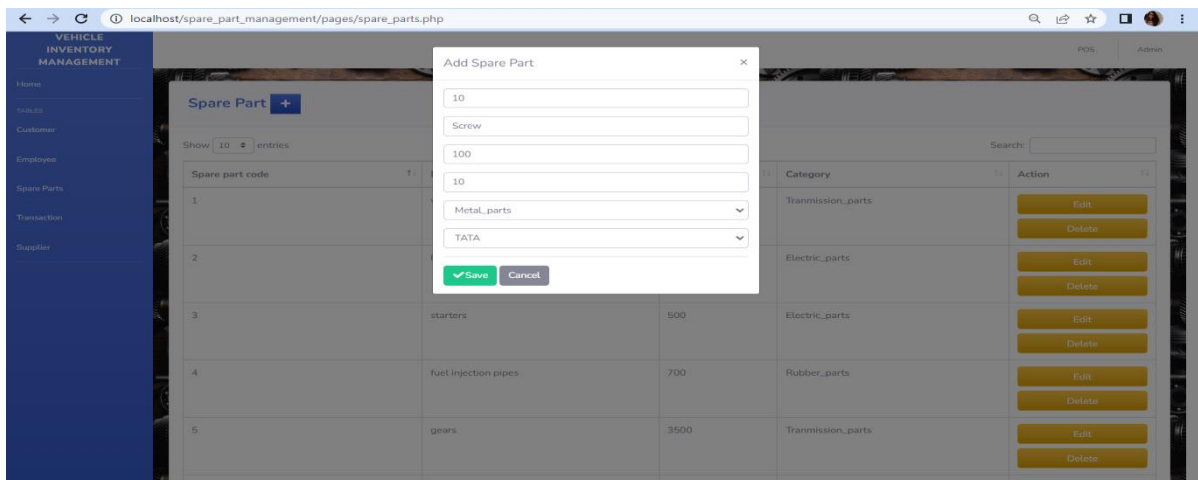


Fig.5.12: add spare part page

This is the page where you get all information about product you have in your shop.

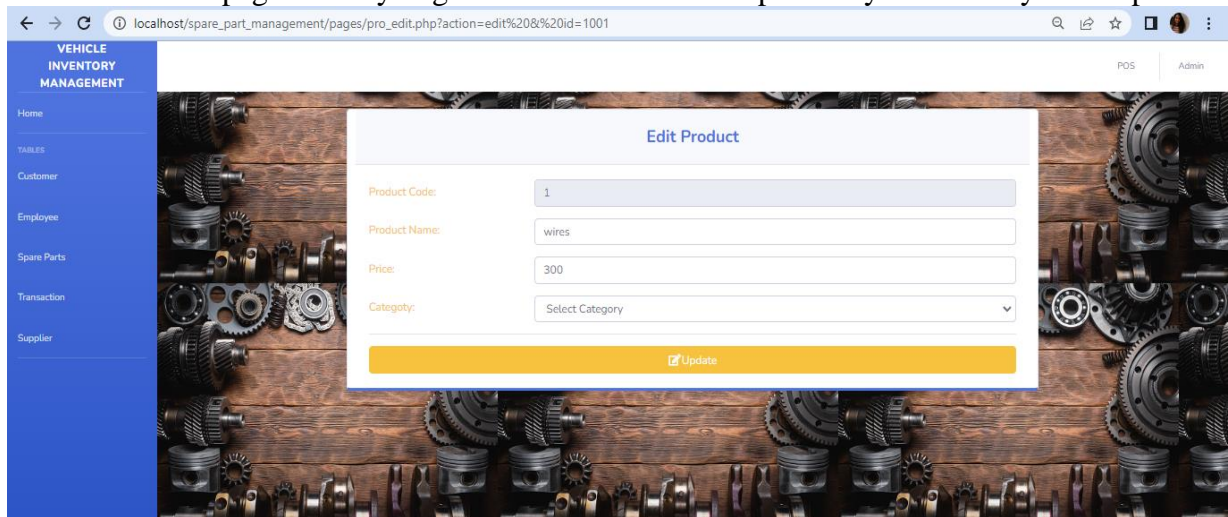


Fig.5.13: edit spare part page

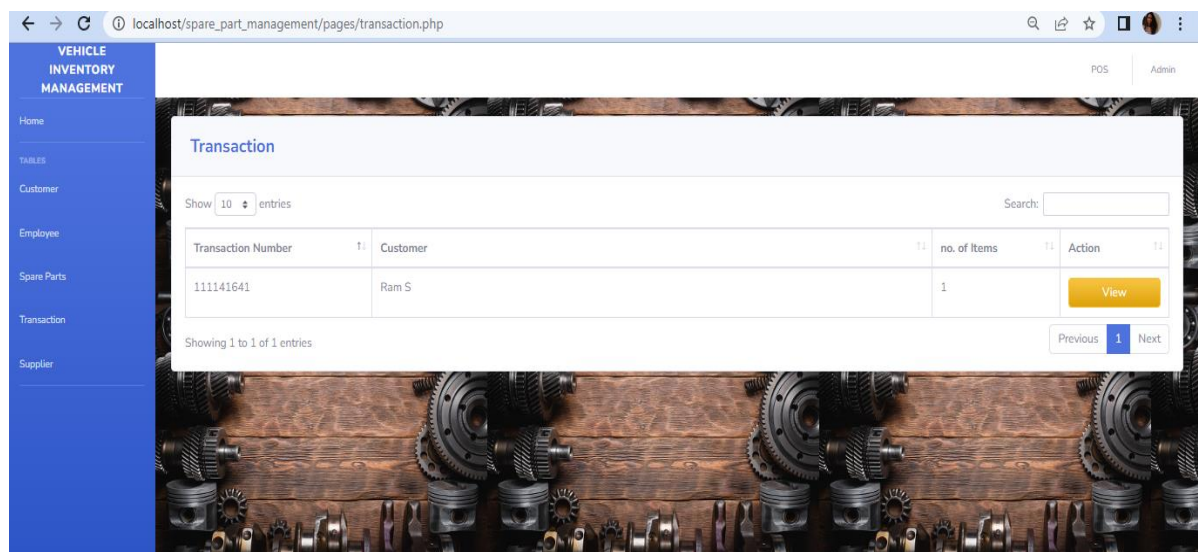


Fig.5.14: Transaction page

Here you will get transaction details of the entire customer. This will also show how much of their money are remaining to pay by clicking on view as below.

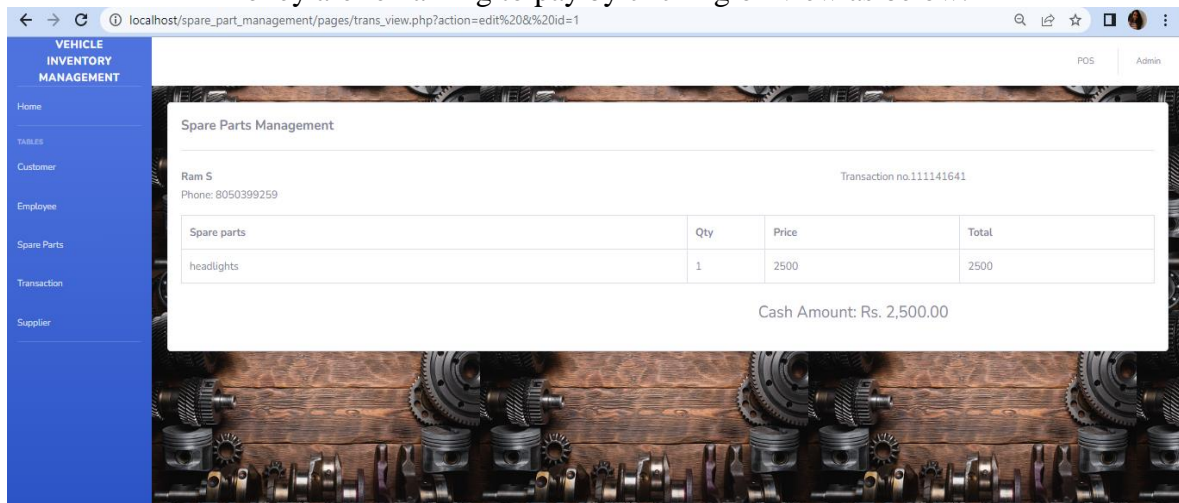


Fig.5.15 :View the transaction of customer

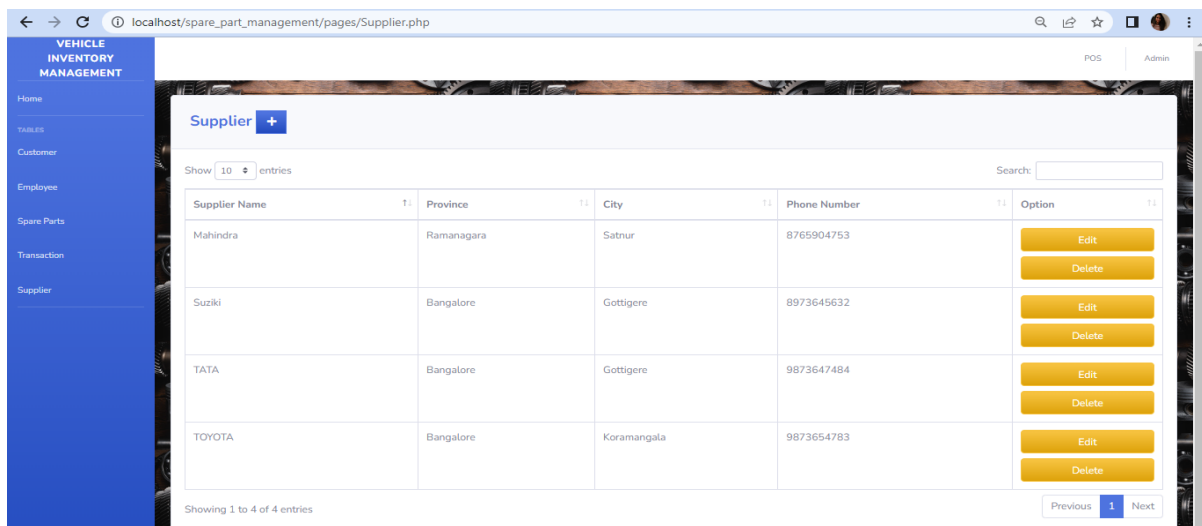


Fig.5.16: Supplier Page

This page will show all the details of the supplier from whom we buys product to sell.

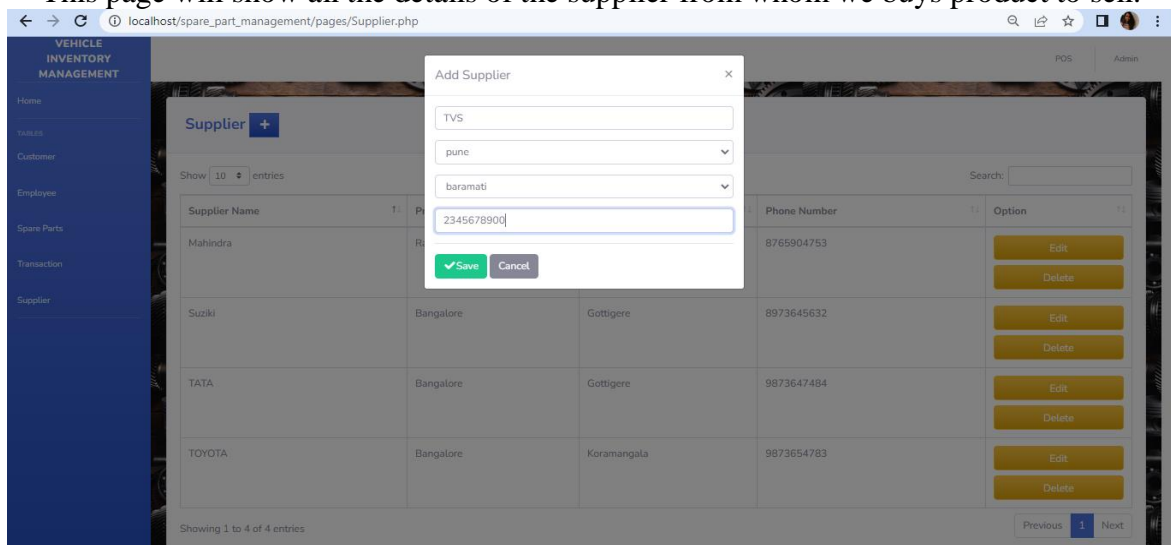


Fig.5.16: add Supplier

localhost/spare_part_management/pages/sup_edit.php?action=edit%20&%20id=3

VEHICLE INVENTORY MANAGEMENT

Home

TABLES

Customer

Employee

Spare Parts

Transaction

Supplier

POS Admin

Edit Supplier

Company Name: Mahindra

Province: Ramanagara

City: Satnur

Phone Number: 8765904753

Update

Fig.5.17:edit Supplier

localhost/spare_part_management/pages/Supplier.php

VEHICLE INVENTORY MANAGEMENT

Home

TABLES

Customer

Employee

Spare Parts

Transaction

Supplier

POS Admin

Supplier +

Show 10 entries Search:

Supplier Name	Province	City	Phone Number	Option
Mahindra	Ramanagara	Satnur	8765904753	Edit Delete
Suzuki	Bangalore	Gottigere	8973645632	Edit Delete
TATA	Bangalore	Gottigere	9873647484	Edit Delete

Fig.5.18:delete Supplier

localhost/spare_part_management/pages/pos.php

Vehicle Inventory Management

POS Admin

Spare part category

Electric_parts Metal_parts Rubber_parts Tranmission_parts

Point of Sale

Spare part Name	Quantity	Price	Total	Action
				SUBMIT

Fig. 5.17: POS page

On right hand-side top you will find POS(point of sale) where we have created four categories. Here you add the product how much you want. If you have new user you can add on plus and add the user and create bill.

Vehicle Inventory Management

POS Admin

Spare part category

Electric_parts Metal_parts Rubber_parts Transmission_parts

headlights Rs. 2500
1
Add

starters Rs. 500
1
Add

Point of Sale

Spare part Name	Quantity	Price	Total	Action
SUBMIT				

localhost/spare_part_management/pages/pos.php?action=add&id=1022

Vehicle Inventory Management

POS Admin

Spare part category

Electric_parts Metal_parts Rubber_parts Transmission_parts

Point of Sale

Spare part Name	Quantity	Price	Total	Action
starters	1	Rs. 500	Rs. 500.00	
Engine mountings	1	Rs. 1,200	Rs. 1,200.00	

Ram S +

Total Rs. 1,700.00

SUBMIT

Fig. 5.18: adding items

This page is for only cashier. Whenever, cashier login he will redirect to this page. Access of the cashier is limit to this page.

localhost/spare_part_management/pages/pos.php?action=add&id=1022

Vehicle Inventory Management

POS Admin

Spare part category

Electric_parts Metal_parts Rubber_parts Transmission_parts

Point of Sale

Spare part Name	Quantity	Price	Total	Action
starters	1	Rs. 500	Rs. 500.00	
Engine mountings	1	Rs. 1,200	Rs. 1,200.00	

Ram S +

Total Rs. 1,700.00

SUBMIT

Rs. 1,700.00

Rs. 1700

PROCEED TO PAYMENT

Fig.5.19: Payment page

Here you enter amount to pay to the owner.

This page is for only cashier. Whenever, cashier login he will redirect to this page. Access of the cashier is limit to this page.

Chapter 6

CONCLUSION

This project is completely for our education usage and would definitely help in .maintaining a good record of all info of parts as the users required and minimize the efforts with the help of modern tech usage.

The software and hardware requirements are quite easily available for the implementation of this project and all individuals can easily understand how to develop this project and implement their own ideas and can be used in large scale applications.

Developing this project was so much fun and informative, helped us learn a lot of new stuff about new languages, databases, and new ideas of how to implement things in various ways.

This project is developed using complete basics of html, css, php, mysql. We like to keep it simple and satisfying all necessary requirements as per objectives of this Mini Project.

At last we want to conclude that this application will decrease the overhead of the owner of the shop. It will be easily able to monitor the available stock present in the shop. It will also helpful for the owner that whose money is pending or how much he has to pay to the supplier. The main motivation behind this application is to reduce tension and easily monitor the current condition of the shop. Now there is no need of books and paper in the shop and manually keep track of each record

We hope our project has been satisfactory as per the Mini project Needs and is beneficial in any ways possible. Thank you all for this opportunity and has helped to progress in our career building.

Chapter 7

REFERENCES

1. Fundamentals of Database Systems, Seventh Edition, Navathe, Pearson.
2. The Complete Reference J2EE, Seventh Edition, Mc-Graw Hill
3. Github.com
4. www.w3schools.com
5. <https://www.mysql.com/>
6. www.google.com
7. www.youtube.com
8. http://docs.oracle.com/cd/B12037_01/server.101/b10739/create.htm
9. <http://docs.oracle.com/javafx/index.html>
10. <http://stackoverflow.com/questions/15482623/javafx-1-3-listview>
11. <http://stackoverflow.com/questions/8392119/javafx-1-3-crud-using-listview>
12. http://docs.oracle.com/cd/E17802_01/javafx/javafx/1.3/docs/api
13. <https://www.tutorialspoint.com/php/index.htm>
14. <https://www.w3schools.com/php/DEFAULT.asp>
15. “Database System Concepts” by Henry Korth.
16. “Mysql” by Paul Dubois
17. <https://getbootstrap.com/>