**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnanasangama”, Belagavi-590018, Karnataka**

****

**BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V.Puram, Bangalore-560 004**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DATABASE MANAGEMENT SYSTEM LAB WITH MINI PROJECT**

**18CSL58**

**AUTO-SPARE PARTSMANAGEMENTT SYSTEM**

**Submitted By**

**<YATISH S GOWGA>**

**<1BI20CS194>**

**for the academic year 2022-23**

**Department of Computer Science & Engineering**

**Bangalore Institute of Technology**

**K.R. Road, V.V.Puram, Bangalore-560 004**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnanasangama”, Belagavi-590018, Karnataka**

**BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V.Puram, Bangalore-560 004**

****

**Department of Computer Science & Engineering**

***Certificate***

This is to certify that the implementation of **DBMS MINI PROJECT** entitled

“Auto-spare part management system” has been successfully completed by

**USN:1BI20CS194 NAME: YATISH S GOWDA**

of V semester B.E. for the partial fulfillment of the requirements for the Bachelor's

degree in Computer Science & Engineering of the Visvesvaraya Technological

University during the academic year 2022-2023.

**Lab Incharge Faculty :**

##### **Dr. B. N. Shankar Gowda**

Associate Professor

Department of CS&E

Bangalore Institute of Technology

Bangalore -04

Examiners: 1) 2)

**ACKNOWLEDGEMENT**

You must acknowledge to

HOD

Lab In-charges

Department Teaching and Non-Teaching staff

Family and Friends

< Name >

< USN >

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **Contents** | | **Page No.** |
| **1.** |  | **Introduction** | 1 |
| **1.1** | **Scope and Limitation** | **2** |
| **1.2** | **Requirement Analysis** | **3** |
| **1.3** | **Feasibility Study** | **4** |
| **1.4** | **System Requirement** | **6** |
| **2.** |  | **Back end Design** | **7** |
| **2.1** | **Design (ER Diagram)** | **8** |
| **2.2** | **ER to Table Conversion** | **9** |
| **2.3** | **Normalization** | **10** |
| **2.4** | **Schema Diagram(ER Mapping)** | **15** |
| **2.5** | **Coding(DDL,DML,Connectivity)** | **16** |
| **2.6** | **Screenshots** | **32** |
| **2.7** | **Future Scope and Enhancement** | **39** |
| **3.** |  | **Conclusion** |  |
| **4.** |  | **Reference** |  |

**TABLE OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Name** | **Page No** |
| **1.1** | ER Diagram of Auto-Spare Part Management | **8** |
| **1.2** | Schema diagram for Auto-Spare Management | **15** |

**CHAPTER-1**

**INTRODUCTION**

## Introduction

The Auto-Spare Part Management 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

Functions of Auto-Spare Part Management:

* 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. Scope and Limitation

**Scope:**

The main reason to build this project is to provide the seller a in-look of the shop. This project wills the guide owner to the products available in the shop and which are required to order from the sellers. By this, 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 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. 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.

## Limitation:

* + - 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

## 1.2. 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.

## 1.3. Feasibility Study

**1. What Is a 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.

# 1.4. System Requirement

## Hardware Requirement:

* + Core to Dual or above
  + 2 GB RAM
  + 25 GB Free Hard disk space
  + Network interface card or Modem
  + LAN Network

## Software Requirements:

* + Windows XP/7/8/10
  + Linux
  + DBMS
  + NetBeans IDE 8.0.2
  + SQL databases

**CHAPTER-2**

**BACK-END DESIGN**

# Design (ER Diagram)

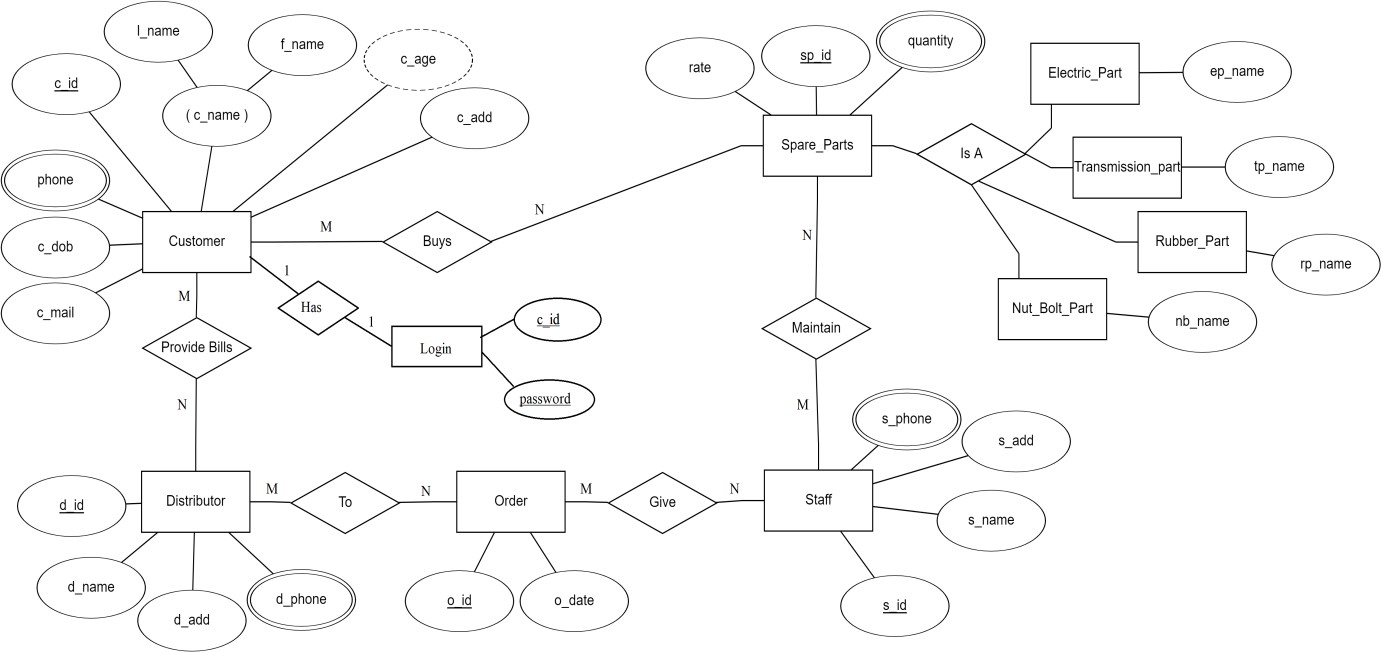


Fig1.1: ER Diagram of Auto-Spare Part Management

# ER to Table Conversion

## Customer Table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| c\_id | c\_dob | phone | c\_mail | l\_name | f\_name | c\_add | c\_age |

**Distributor:**

|  |  |  |  |
| --- | --- | --- | --- |
| d\_id | d\_name | d\_add | d\_phone |

**Staff:**

|  |  |  |  |
| --- | --- | --- | --- |
| s\_id | s\_name | s\_add | s\_phone |

**Spare\_Parts:**

|  |  |  |
| --- | --- | --- |
| sp\_id | rate | quantity |

**Order:**

|  |  |
| --- | --- |
| o\_id | o\_date |

* 1. **Normalization (up to 3NF)**

1. **Customer Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| c\_id | c\_dob | phone | c\_mail | l\_name | f\_name | c\_add | c\_age |

**C\_id->f\_name, l\_name, c\_dob, phone, c\_mail, c\_add, c\_age; The Customer Table is in 1 NF as:**

* + In above table attribute c\_phone can contain multiple values.
  + So it can be further devided into two tables, with separate “Phone” table as below –
  1. **Phone Table**

|  |  |
| --- | --- |
| c\_id | phone |

**c\_id ->phone;**

* 1. **Customer Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| c\_id | c\_dob | c\_mail | l\_name | f\_name | c\_add | c\_age |

**C\_id->f\_name, l\_name, c\_dob, c\_mail, c\_add, c\_age;**

**The Customer Table is in 2 NF as:**

* + - Customer Table is in 1 NF
    - **Candidate key :** c\_id, c\_mail
    - **Non prime attributes :** c\_dob, f\_name, l\_name, c\_add, c\_age
    - No partial dependency here (no change)

**The Customer Table is in 3 NF as:**

* + It is in 2 NF
  + c\_id,c\_mail - >c\_name (candidate key)

c\_id,c\_mail - >c\_add (candidate key) c\_id,c\_mail - >c\_dob (candidate key) c\_dob - >c\_age (Transitive Dependency)

* + Here c\_age is transitively dependent on candidate key c\_id,c\_mail.
    - **1.3 Customer\_age Table**

|  |  |
| --- | --- |
| c\_dob | c\_age |

**C\_dob-> c\_add ;**

1. **Distributor**

|  |  |  |  |
| --- | --- | --- | --- |
| d\_id | d\_name | d\_add | d\_phone |

**d\_id->d\_name, d\_phone, d\_add; The Distributor Table in 1 NF as:**

* + In above table attribute c\_phone can contain multiple values.
  + So it can be further devided into two tables, with separate “Phone” table as below –
  1. **Phone Table**

|  |  |
| --- | --- |
| d\_id | phone |

**d\_id ->phone;**

* 1. **Distributor Table**

|  |  |  |
| --- | --- | --- |
| d\_id | d\_name | d\_add |

**d\_id->d\_name, d\_add ;**

**The Distributor Table is in 2 NF as:**

* + It is in 1 NF
  + Here , d\_id- >d\_name (primary key)

d -> d\_add (primary key) d-> phone (primary key)

* + - No partial dependency here (no change)

**The Distributor Table is in 3 NF as:**

* + It is in 2 NF
  + NO transitive dependency here in Distributor table.

1. **Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| s\_id | s\_name | s\_add | s\_phone |

**s\_id->s\_name, s\_phone, s\_add; The Staff Table in 1 NF as:**

* + In above table attribute s\_phone can contain multiple values.
  + So it can be further devided into two tables, with separate “Phone” table as below –
  1. **Phone Table**

|  |  |
| --- | --- |
| s\_id | phone |

**s\_id ->phone;**

* 1. **Staff Table**

|  |  |  |
| --- | --- | --- |
| s\_id | s\_name | s\_add |

**s\_id->s\_name, s\_add;**

**The Staff Table is in 2 NF as:**

* + It is in 1 NF
  + Here , s\_id- >s\_name (primary key)

s -> s\_add (primary key) s-> phone (primary key)

* + - No partial dependency here (no change)

**The Staff Table in 3 NF as:**

* + It is in 2 NF
  + NO transitive dependency Here in staff table.

1. **Spare\_Parts**

|  |  |  |
| --- | --- | --- |
| sp\_id | rate | quantity |

**Sp\_id->rate, quantity;**

**The Spare\_parts Table is in 1 NF as:**

* + All the underlying domains of the relation contain atomic (indivisible) values.
  + No repeating groups in the table.
  + All attributes dependent on primary key.
  + No change here

**The Spare\_parts Table is in 2 NF as:**

* + It is in 1 NF
  + Functiona dependencies here – Sp\_id ->rate (primary key) Sp\_id ->quantity (primary key)
    - No Partial Dependency Here.

**The Spare\_parts Table is in 3 NF as:**

* + It is in 2 NF
  + It includes no transitive dependencies as there are no non key attributes.
  + No change.

1. **Order**

|  |  |
| --- | --- |
| o\_id | o\_date |

**O\_id-> o\_date;**

**The Order Table is in 1 NF as:**

* + All the underlying domains of the relation contain atomic (indivisible) values.
  + No repeating groups in the table.
  + All attributes dependent on primary key.

**The Order Table is in 2 NF as:**

* + It is in 1 NF
  + It includes no partial dependencies as every non key attribute is fully dependent on the primary key.

**The Order Table is in 3 NF as:**

* + It is in 2 NF
  + No transitive Dependency (no change).

# Schema Diagram(ER-Mapping)

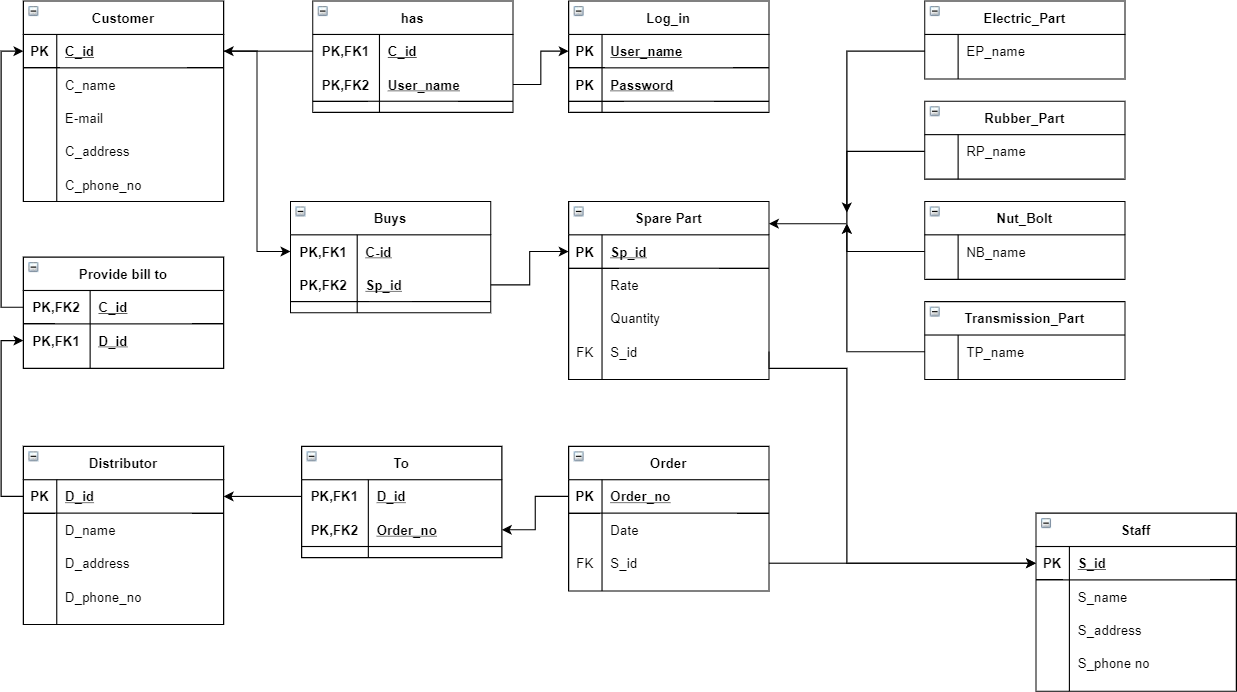


Fig1.2: Schema diagram for Auto-Spare Management

# Coding (DDL,DML,Connectivity)

## DDL and DML commands

-- phpMyAdmin SQL Dump

-- version 5.0.2

-- [https://www.phpmyad](http://www.phpmyadmin.net/)min.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 (17, 'kunal', 'kamra', '9873635464'),

(18, 'gaurav', 'patil', '8934827482'),

(19, 'arnab', 'goswami', '9876478974'),

(20, 'amit', 'shah', '8678947353'),

(21, 'narendra', 'modi', '7865346734');

--

-- 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,

`HIRED\_DATE` varchar(50) 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`, `HIRED\_DATE`) VALUES

(1, 'pavan', 'patil', 'Male', 'pavan@gmail.com', '9876567865', 1, 113, NULL),

(5, 'karan', 'nakum', 'Male', 'karan@gmail.com', '89763546334', 1, 163, NULL),

(6, 'hrushikesh', 'rajankar', 'Male', 'rushi@gmail.com', '9874657834', 2, 164, NULL),

(7, 'gopal', 'paraskar', 'Male', 'gopal@gmail.com', '7894653125', 2, 165, NULL);

--

-- 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

|  |  |  |
| --- | --- | --- |
| (113, | 'pune', | 'kondhwa'), |
| (159, | 'pune', | 'kondhwa'), |
| (160, | 'pune', | 'kothrud'), |
| (161, | 'nagpur', 'kuhi'), | |
| (162, | 'amravati', 'morshi'), | |
| (163, | 'pune', 'katraj'), | |
| (164, | 'nagpur', 'narkhed'), | |

(165, 'amravati', 'dhamangaon');

--

-- Table structure for table `manager`

--

CREATE TABLE `manager` (

`FIRST\_NAME` varchar(50) DEFAULT NULL,

`LAST\_NAME` varchar(50) DEFAULT NULL,

`LOCATION\_ID` int(11) NOT NULL,

`EMAIL` varchar(50) DEFAULT NULL,

`PHONE\_NUMBER` varchar(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- 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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (28, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (29, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (30, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (31, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (32, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (33, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (34, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (35, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (36, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (37, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (38, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (39, | '1', | 'wires', | '', | 1, | 1, | 300, | | 4, | 19, | ''), | | |
| (40, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (41, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (42, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (43, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (44, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (45, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (46, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (47, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (48, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (49, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 17, | ''), |
| (50, | '2', | 'headlights', | | '', | | 1, | 1, | 2500, | | 1, | 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 (17, 'TATA', 159, '9873647484'),

(18, 'TOYOTA', 160, '9873654783'),

(19, 'Mahindra', 161, '8765904753'),

(20, 'Suziki', 162, '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

(13, 20, '1', '2,500.00', '267.86', '2,232.14', '267.86',

'2,500.00', '2500', '2020-11-11 04:15 am', '111141641'),

(14, 19, '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

(21, '111141641', 'headlights', '1', '2500', 'pavan', 'Manager'),

(22, '111151334', 'wires', '1', '300', 'pavan', 'Manager'),

(23, '111151334', 'headlights', '1', '2500', 'pavan', '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, | 5, | 'admin2', | '315F166C5ACA63A157F7D41007675CB44A948B33', |
| 1), |  |  |  |
| (3, | 6, | 'admin3', | '33AAB3C7F01620CADE108F488CFD285C0E62C1EC', |
| 2), |  |  |  |
| (4, | 7, | '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=10;

--

-- AUTO\_INCREMENT for table `customer`

--

ALTER TABLE `customer`

MODIFY `CUST\_ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=22;

--

-- AUTO\_INCREMENT for table `employee`

--

ALTER TABLE `employee`

MODIFY `EMPLOYEE\_ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT for table `location`

--

ALTER TABLE `location`

MODIFY `LOCATION\_ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=166;

--

-- AUTO\_INCREMENT for table `product`

--

ALTER TABLE `product`

MODIFY `PRODUCT\_ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=665;

--

-- AUTO\_INCREMENT for table `supplier`

--

ALTER TABLE `supplier`

MODIFY `SUPPLIER\_ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=21;

--

-- AUTO\_INCREMENT for table `transaction`

--

ALTER TABLE `transaction`

MODIFY `TRANS\_ID` int(50) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT for table `transaction\_details`

--

ALTER TABLE `transaction\_details`

MODIFY `ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=24;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `ID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=10;

--

-- Constraints for dumped table

--

--

-- 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`),

ADD CONSTRAINT `users\_ibfk\_4` FOREIGN KEY (`EMPLOYEE\_ID`) REFERENCES `employee` (`EMPLOYEE\_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

## Connectivity

<?php

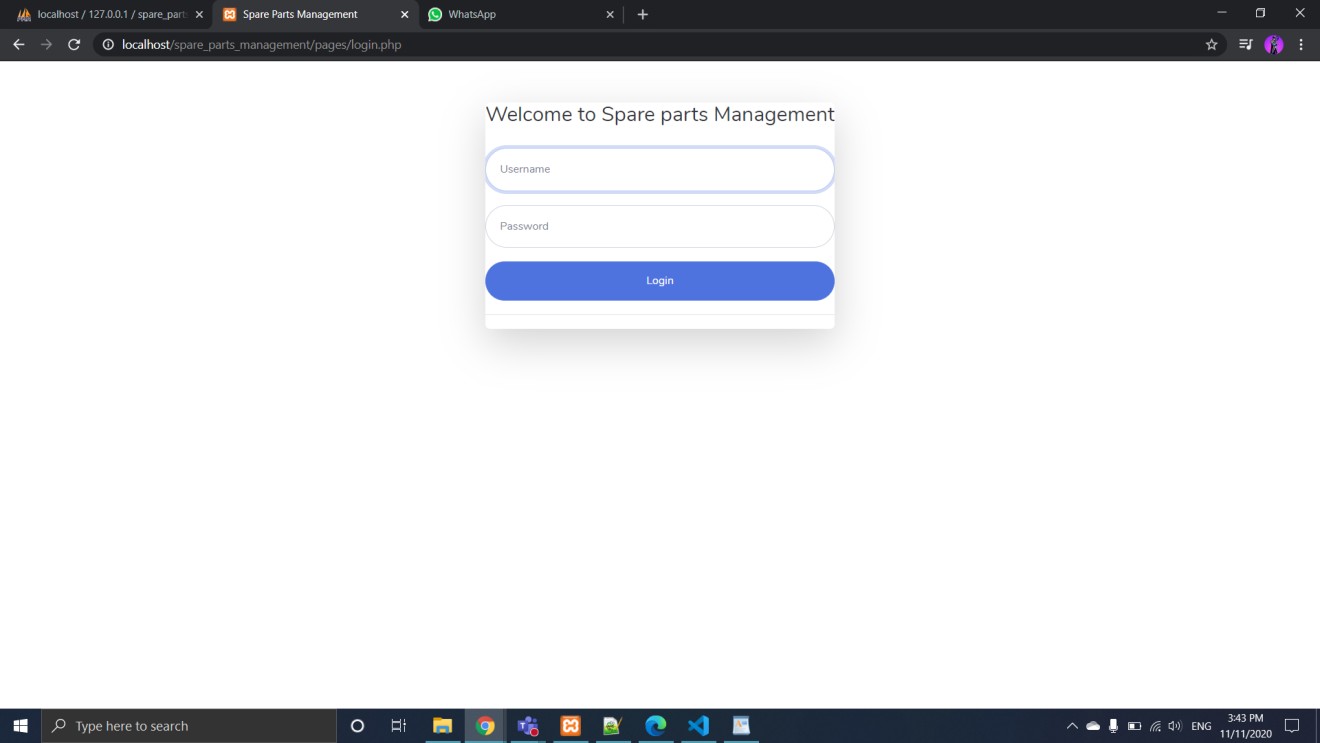
$db = mysqli\_connect('localhost', 'root', '') or

die ('Unable to connect. Check connection.'); mysqli\_select\_db($db, 'Spare\_parts' ) or

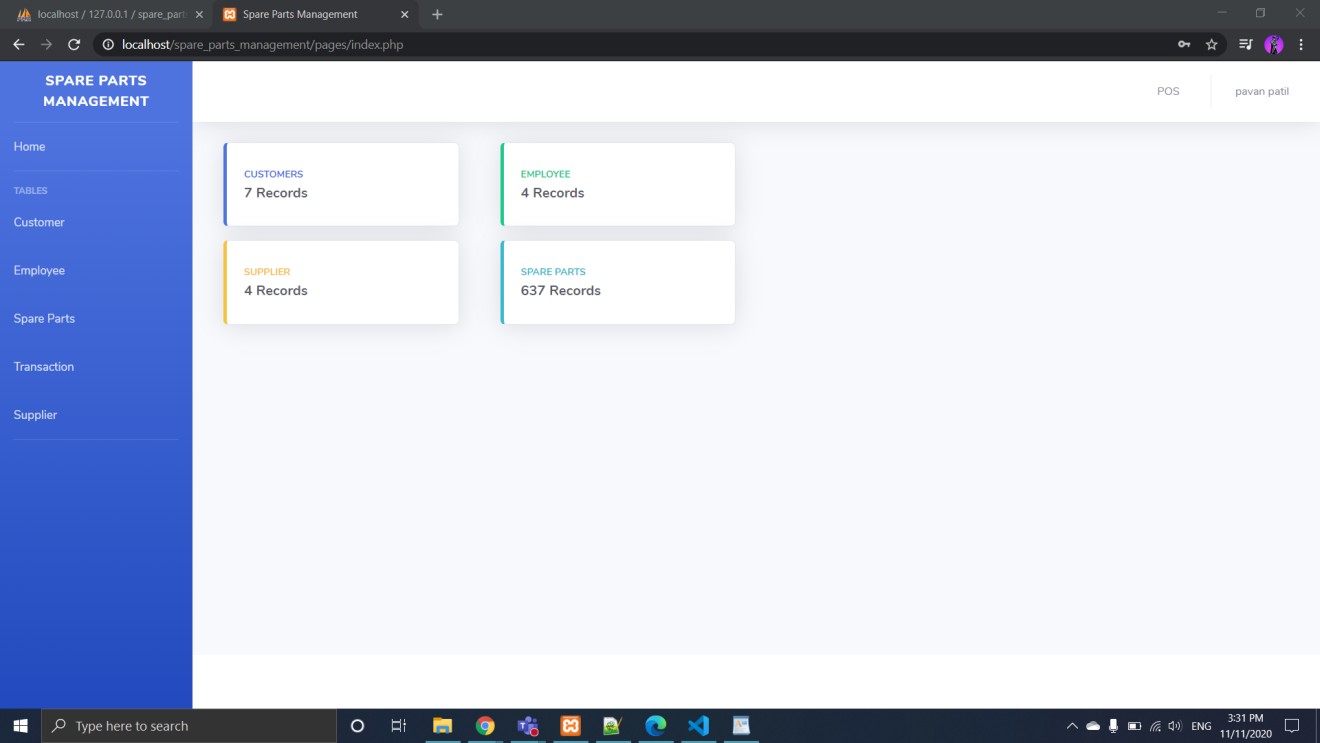
die(mysqli\_error($db));

?>

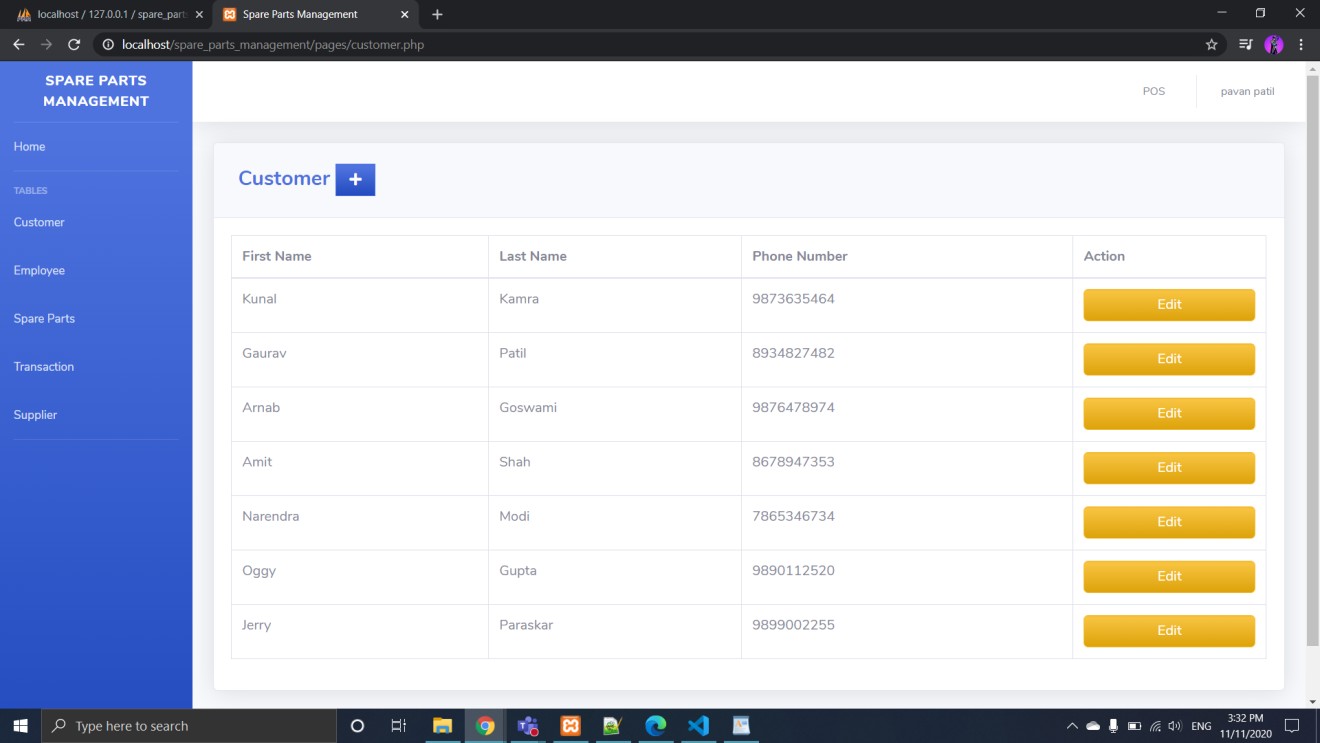
# Screenshots



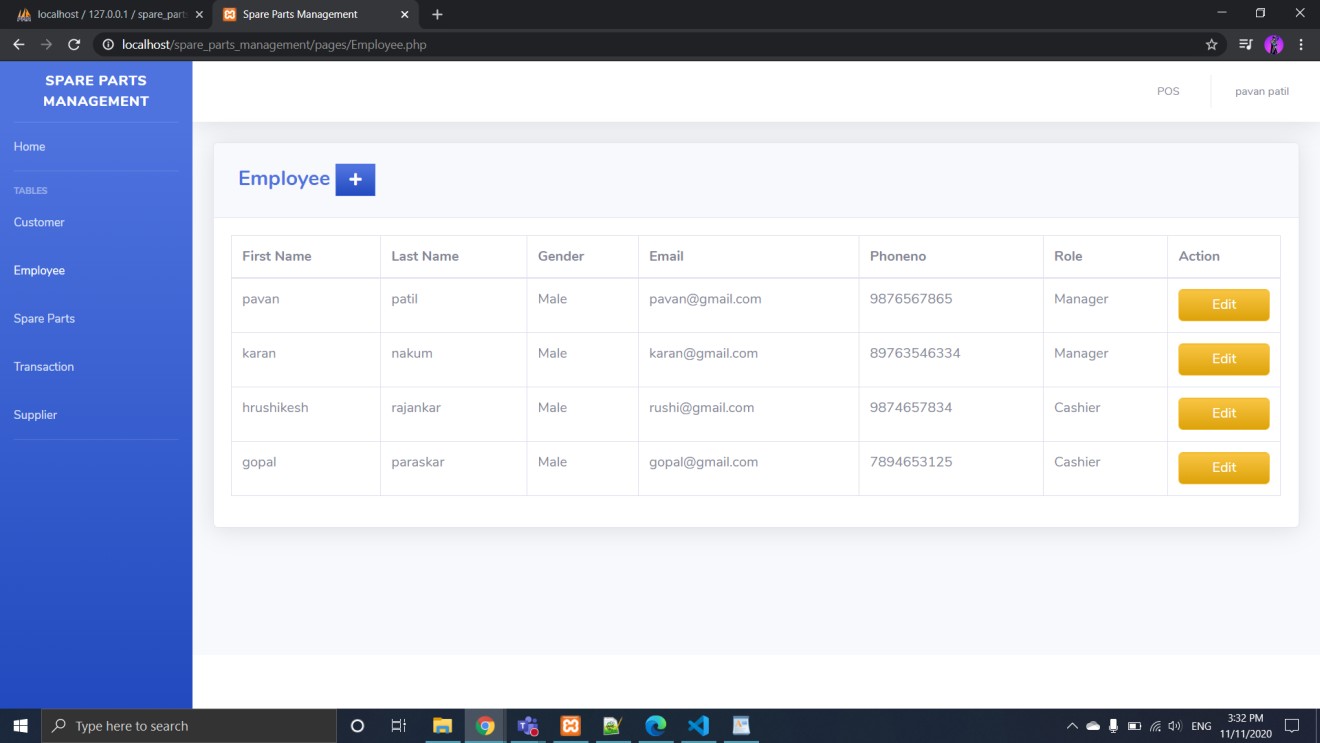
This is over login page. There are two types of login one for manager and one for cashier. The following screenshots is for manager login.



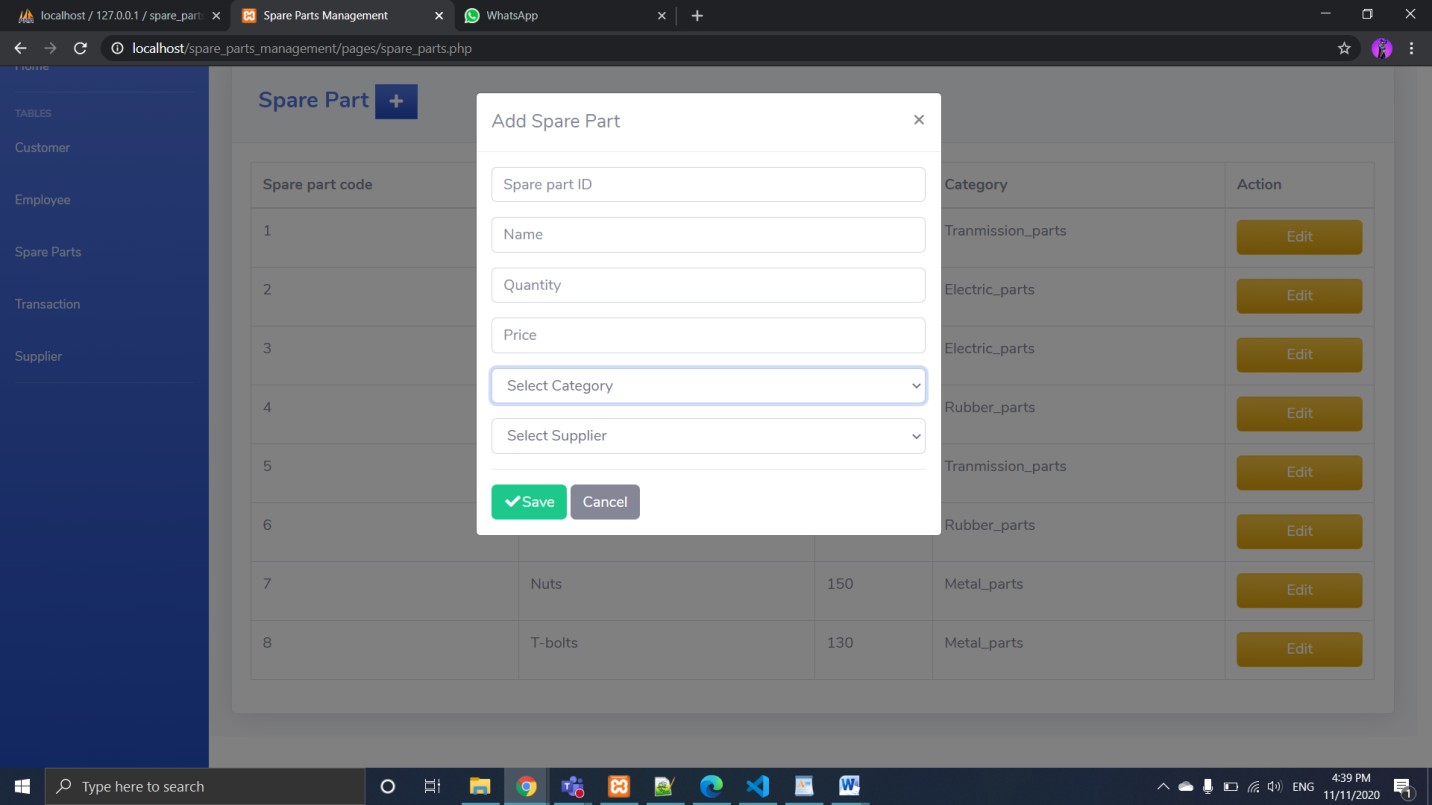
This is our home page. After entering correct username and password the manager will redirect to this page.



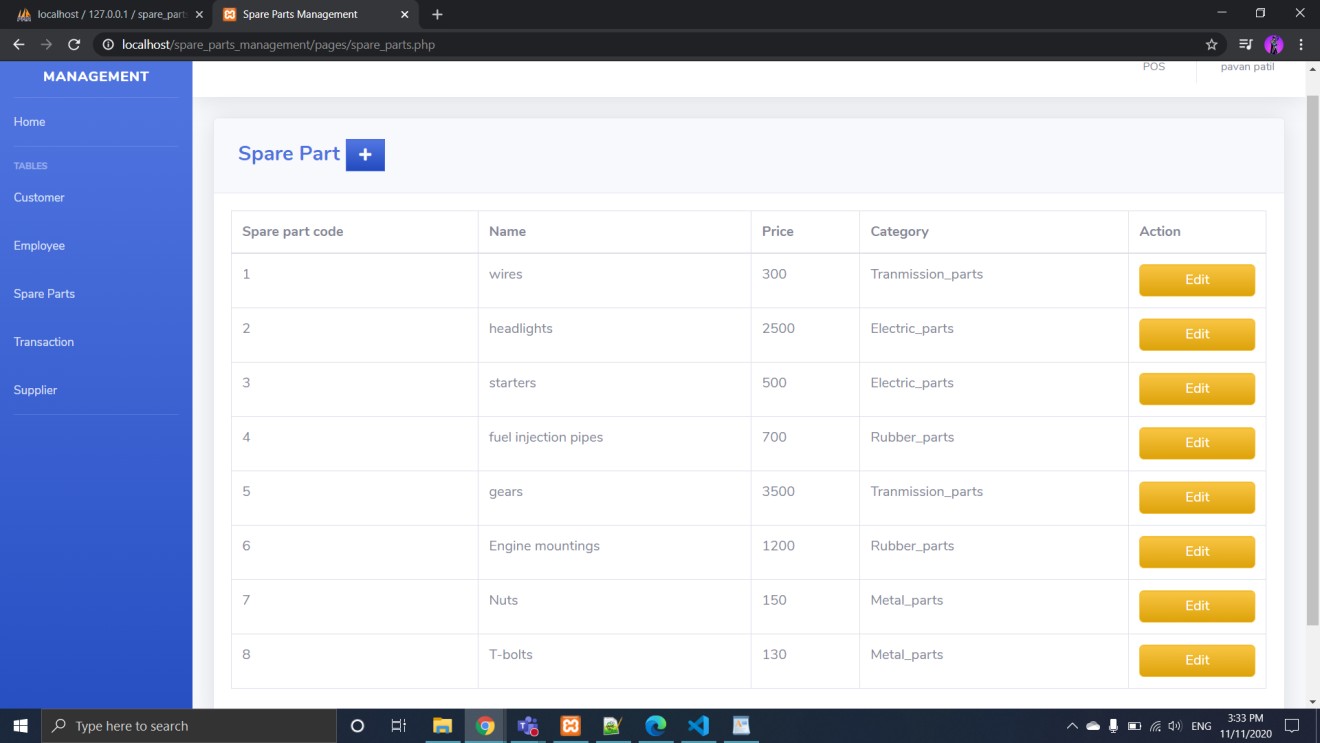
This is customer page. Here you will information about the entire customer who has buy product from you. When you click on edit, you can change the information about the customer.



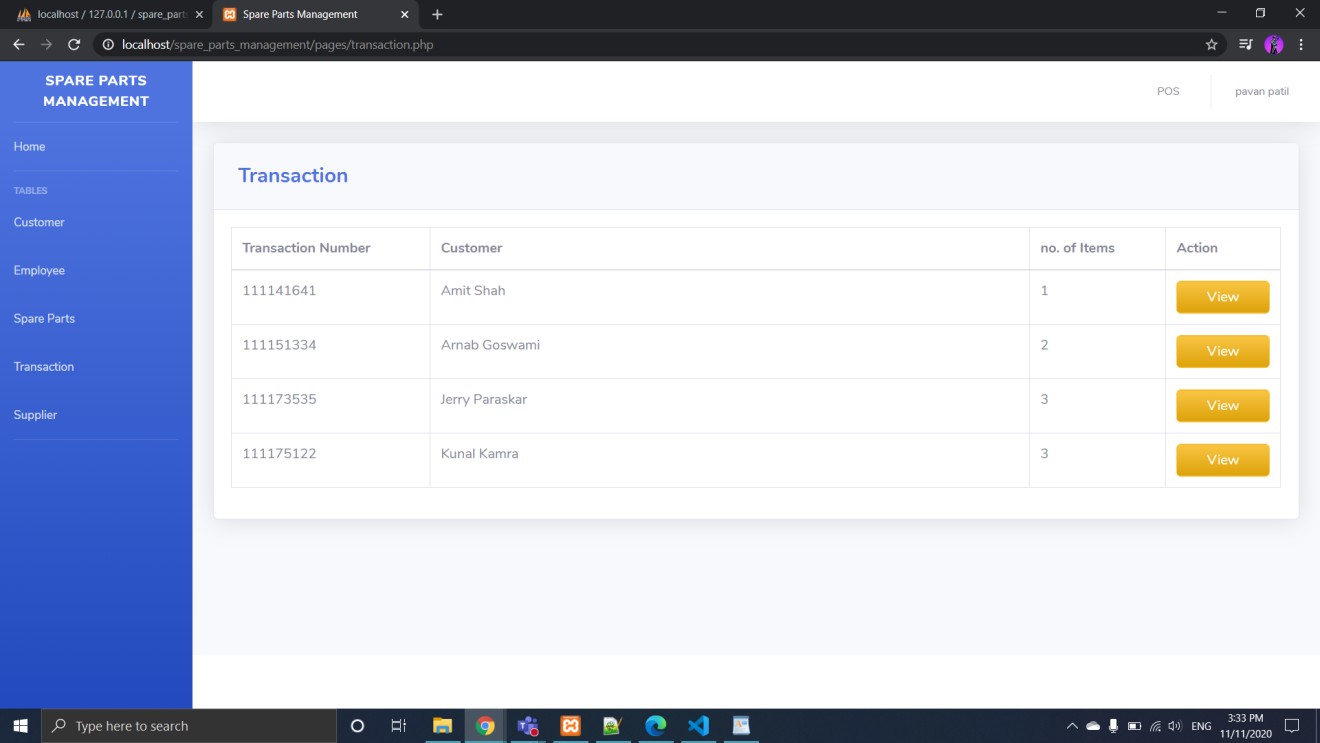
This is our employee page here you can add or remove employee name who are working with you or who was working with you.



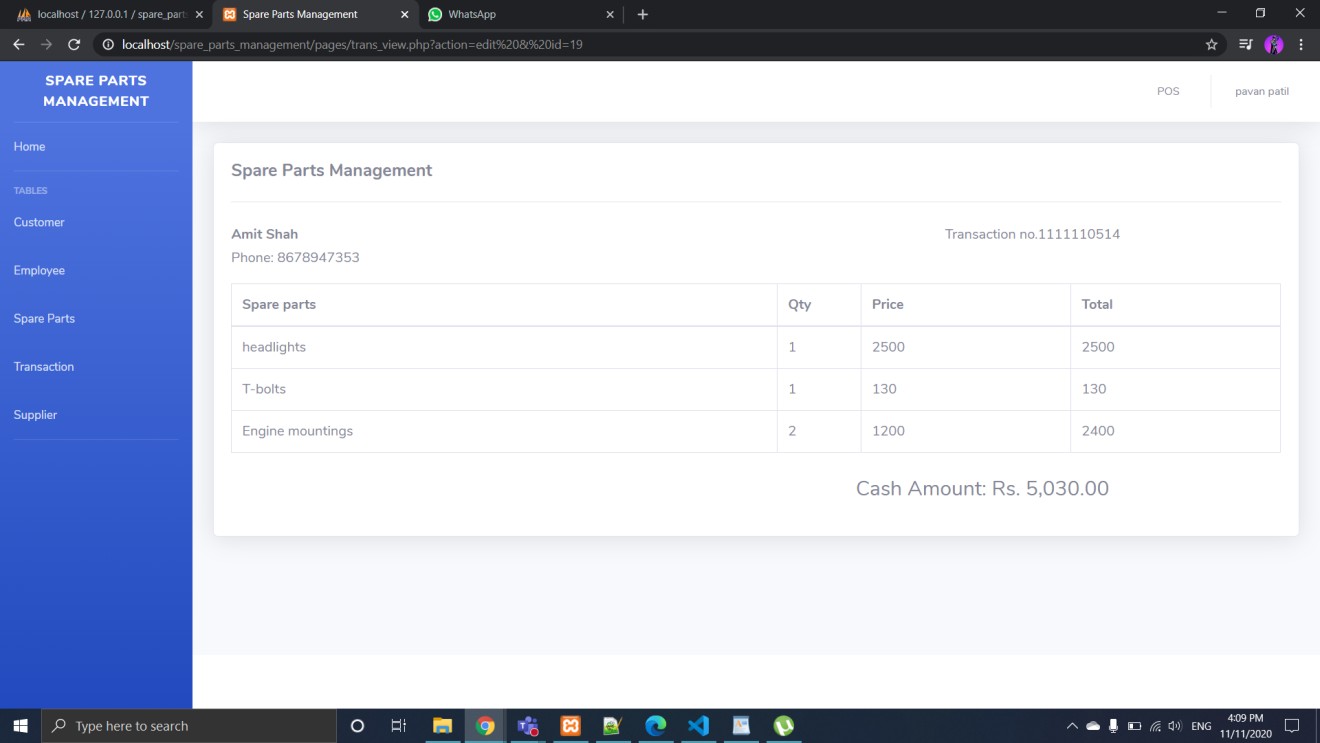
Here you add name of new spare part with their details.

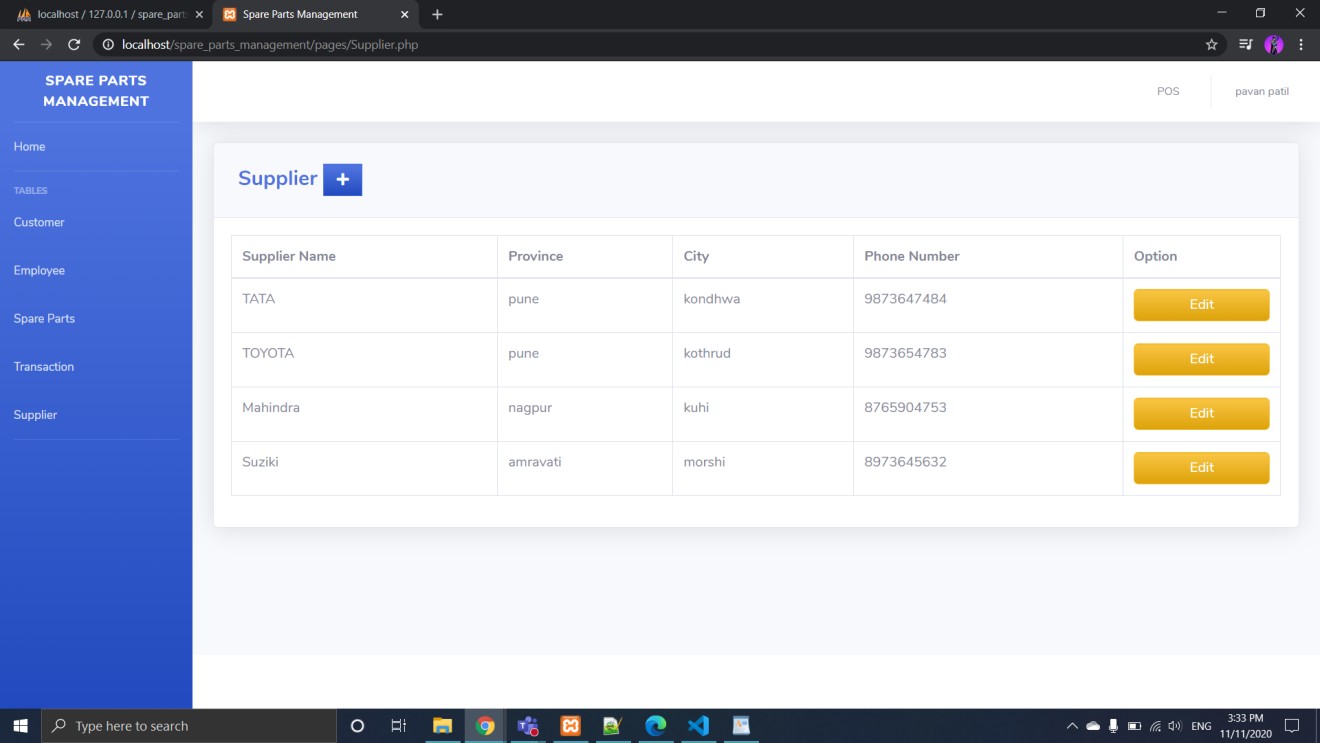


This is the page where you get all information about product you have in your shop.

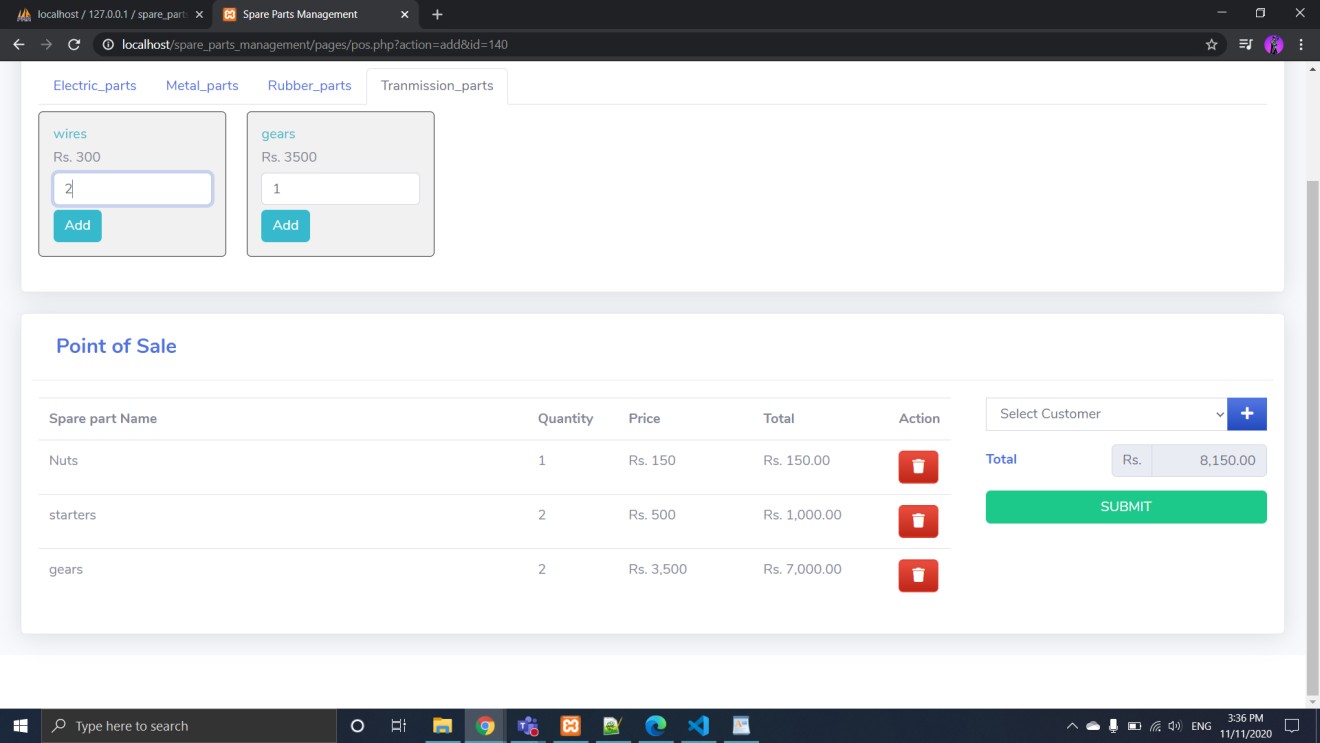


Here you will get transaction details of the entire customer. This will also show how much of their money are remaining to pay by clicking on view as below.

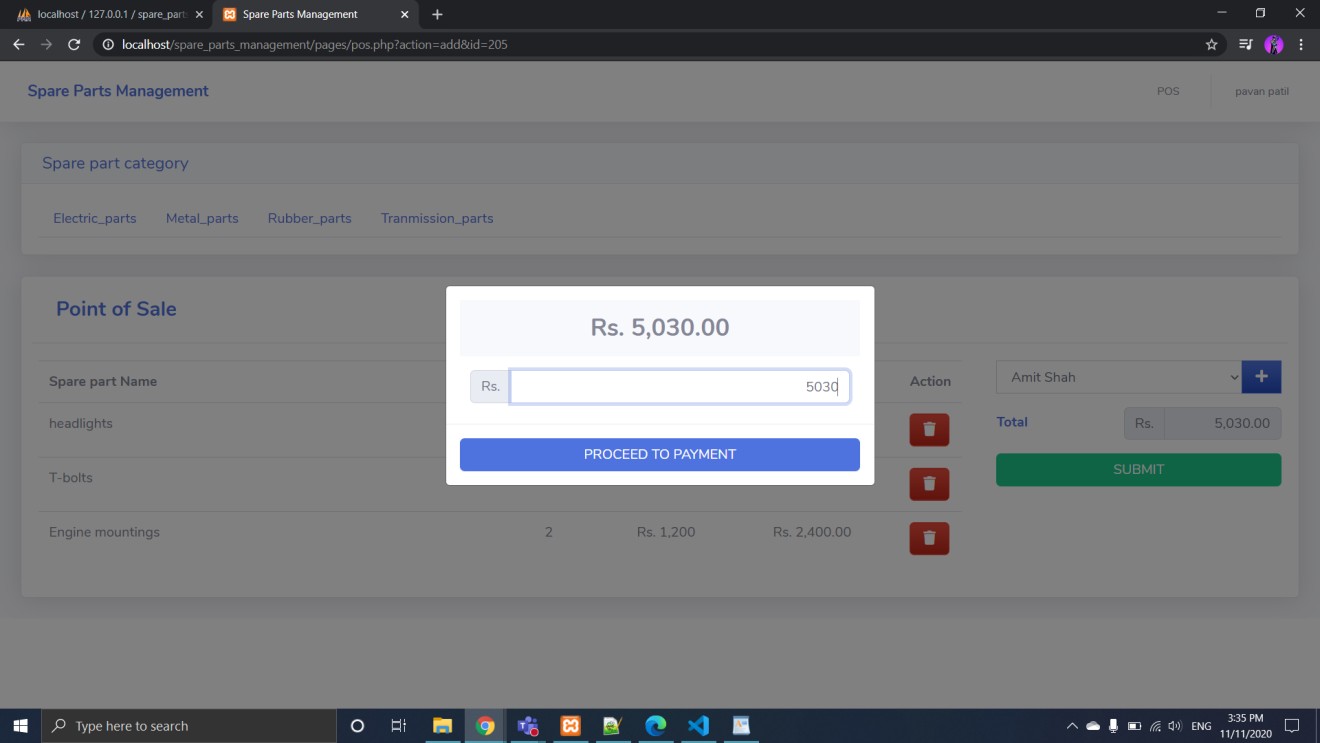




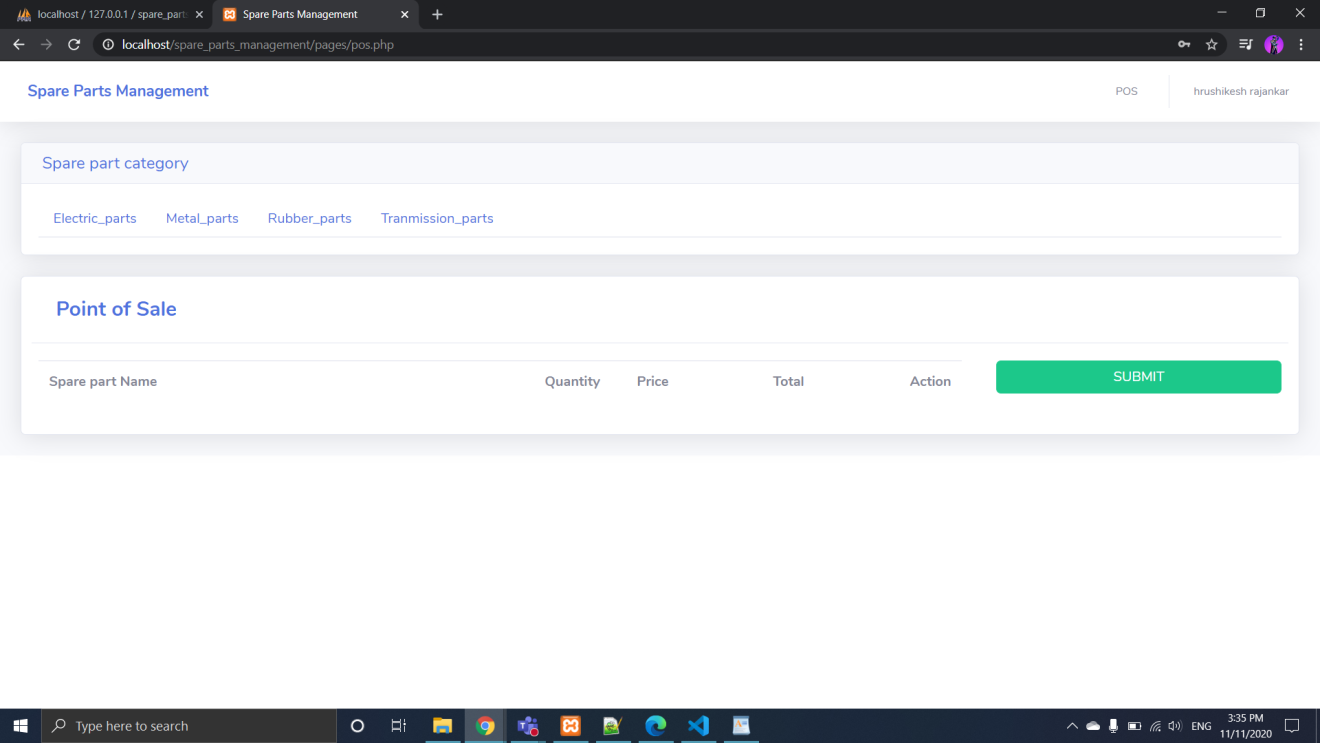
This page will show all the details of the supplier from whom we buys product to sell.



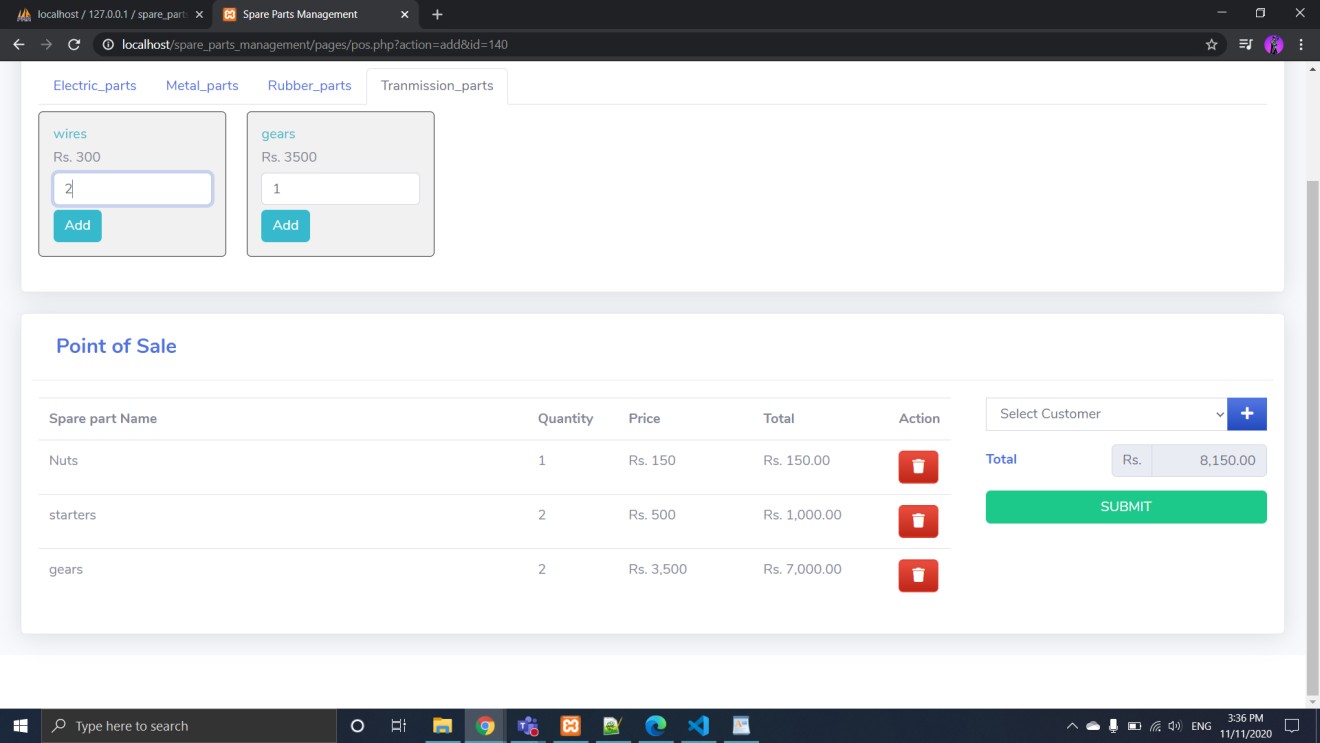
On right hand-side top you will find POS(point of sale) where we have created four categories. Here you add the product how much you want. If you have new user you can add on plus and add the user and create bill.



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.



Here, cashier add the customer and make the bill.

# Future Scope/Enhancement

As you see, IT industry is booming the industry from last few decades. It will keep on booming as you go on and on. As you see our project is limited to one user\owner. But, we can further expand this to multi-user to multi-customer and there is way we can do that. We all know that there and various online shopping application and website available for their basic need. But, there is no application or website available for the Automobile Spare Parts. These were we can enhance this project further. We can make an application like Amazon were you get every single piece of the spare part on your application and need to go anywhere to find it. By providing the best product at best prices everyone going to come and visit the site. We can show the picture of the product with the offers on the particular occasion.

We can also add the feature that how amount of work has been done by the employee at your shop and how leaves taken by him/her. We can also add online payment methods with the help of Google Pay, PhonePe etc.

**CHAPTER-3**

**CONCLUSION**

# Conclusion

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.

**CHAPTER-4**

**REFERNCES**

# Reference

# http://docs.oracle.com/cd/B12037\_01/server.101/b10739/create.htm

# http://docs.oracle.com/javafx/index.html

# http://stackoverflow.com/questions/15482623/javafx-1-3-listview

# http://stackoverflow.com/questions/8392119/javafx-1-3-crud-using-listview

# http://docs.oracle.com/cd/E17802\_01/javafx/javafx/1.3/docs/api

# https://www.tutorialspoint.com/php/index.htm

# https://www.w3schools.com/php/DEFAULT.asp

# “Database System Concepts” by Henry Korth.

# “Mysql” by Paul Dubois.