Century Mazda Website User Manual

Century Mazda is your premier outlet for Mazda vehicles. As such our state of the art database is available for use and perusal by our customers and employees alike. Our company only used one program, not because we are cheap, but to pass the savings on to you. Here we have laid out all the functionality and features of this masterpiece of program design.

1. Homepage: When you first access out site you will be a customer. You have the option to search our in-stock vehicles or search our employee database relations for phone numbers and email. If you are an employee, you can log in.
2. Vehicle searches. We allow our users to search our vehicle database by a car’s make model and max price. Once it returns the table, please click on the vin to see more detailed information of that car.
3. Employee. To search employees, look up by department to see a list with contact info.
4. Login. Use a known login ID and password to access the employee section, this turns into logout after logging in.

(Employee empIDs: 10001 - 10014, Password 12345)

(Manager empIDs: 50001 - 50002, Password 12345)

1. Employee: We have the same functionality as a customer but our searches return more fields. We also add the ability to sell a car: you search for and find a car, click the VIN, then give it a sold price, financing info and click sell.
2. Manager: They have all the same abilities as employees and customers. They also have the power to add and remove both vehicles and employees.
3. Employee performance: This return relevant data about how well employees are selling cars
4. Add vehicle. This allows the manager to add new stock to the lot.
5. Add employee, when a new employee is hired on we add their data
6. Also if you navigate to a car or employee through the search tabs you can delete them

Finally we logout and make some sales. We hope you enjoy buying form us.

Century Mazda

Our project SQL

Search a department for all employees POC, this is for customers, employees just returns more fields

$stmt = "SELECT name, phone, email, dname FROM $employeeTable, $worksinTable, $departmentTable WHERE $departmentTable.dname = $department AND $departmentTable.did = $worksinTable.did AND $worksinTable.empID = $employeeTable.empID";

These add a new vehicle to our inventory

$stmt = "INSERT INTO $vehiclesTable $th VALUES $vals";

$stmt = "INSERT INTO $onlotTable $th VALUES $vals";

$stmt = "INSERT INTO $installedTable $th VALUES $vals"; (This is repeated for all installed options)

These display the detailed info of a vehicle and its options when selected by vin, again for customers. More fields are returned if you were an employee or a manager

$stmt = "SELECT DISTINCT $vehicleTable.Vin, make, model, level, year, color, listedPrice FROM $vehicleTable, $installedTable WHERE $vehicleTable.Vin=$installedTable.Vin AND $vehicleTable.Vin=$vin";

$stmt = "SELECT DISTINCT Oname FROM $vehicleTable, $installedTable WHERE $vehicleTable.Vin=$installedTable.Vin AND $vehicleTable.Vin=$vin";

These search for a vehicle by price or model, More fields are returned if you were an employee or a manager

$stmt = "SELECT DISTINCT $vehicleTable.Vin, make, model, level, year, color, listedPrice FROM $vehicleTable, $installedTable WHERE $vehicleTable.model=$model AND $vehicleTable.Vin=$installedTable.Vin AND $vehicleTable.make=$make AND $vehicleTable.Type='On\_Lot' ORDER BY listedPrice DESC";

$stmt = "SELECT DISTINCT $vehicleTable.Vin, make, model, level, year, color, listedPrice FROM $vehicleTable, $installedTable WHERE $vehicleTable.model=$model AND $vehicleTable.Vin=$installedTable.Vin AND $vehicleTable.make=$make AND $vehicleTable.listedPrice<=$max AND $vehicleTable.Type='On\_Lot' ORDER BY listedPrice DESC";

These are how an employee or manager is deleted when they are fired

$stmt = "DELETE FROM $worksinTable WHERE $worksinTable.empID=$id";

$stmt = "DELETE FROM $managerTable WHERE $managerTable.empID=$id";

$stmt = "DELETE FROM $employeeTable WHERE $employeeTable.empID=$id";

This is how a vehicle is deleted if for example a mistake was made on its input or it was destroyed

$stmt = "DELETE FROM $onlotTable WHERE $onlotTable.Vin=$vin";

$stmt = "DELETE FROM $installedTable WHERE $installedTable.Vin=$vin";

$stmt = "DELETE FROM $vehicleTable WHERE $vehicleTable.Vin=$vin";

To sell a car as employee or manager

$stmt = "INSERT INTO sold (Vin, empid, DateSold, invoiceNum, daysOnLot, priceSold) VALUES ('" . $\_SESSION['carArray'][0]['Vin'] . "', '" . $\_SESSION['user'] . "', '" . $date . "', '" . $invoiceNumber . "', '" . $daysOnLot . "', '" . $\_POST['sellPrice'] . "')";

$stmt = "INSERT INTO sold\_by VALUES ('" . $\_SESSION['user'] . "', '" . $invoiceNumber . "', '" . $loanNumber . "')";

$stmt = "UPDATE Vehicle SET Type = 'Sold' WHERE Vin = '" . $\_SESSION['carArray'][0]['Vin'] . "'";

The trigger to delete from on\_lot when added to sold

CREATE TRIGGER `sellVehicle` BEFORE INSERT ON `sold` FOR EACH ROW BEGIN

DELETE FROM on\_lot

WHERE on\_lot.Vin=NEW.Vin;

END

To validate login credentials

$stmt = "SELECT \* FROM $table WHERE empID='$user' AND password='$password'";

To see who has been selling how many cars

$stmt = "SELECT COUNT(empID) FROM $soldTable WHERE `empID` = $id";

To add employees as a manager

These add a new vehicle to our inventory

$stmt = "INSERT INTO $employeeTable $th VALUES $vals";

$stmt = "INSERT INTO $worksinTable $th VALUES $vals";

$stmt = "INSERT INTO $installedTable $th VALUES $vals"; (This is repeated for all installed options)

In addition to these we use SELECT DISTINCT to get us the list of models, makes, or departments to populate in the drop down to make searching easier.