Software Requirements Specification & Test Document

Car Management System (CRMS) – Web Application

BOYCE, Jason A & Manning, Catherine

Contents

[Introduction 1](#_Toc1180489494)

[Purpose 1](#_Toc526632739)

[Intended Audience 1](#_Toc1339884096)

[Project Scope 2](#_Toc1596485855)

[Overall Description 2](#_Toc1966676439)

[Product perspective 2](#_Toc1987320157)

[Constraints 2](#_Toc1484954861)

[Assumptions 2](#_Toc116056699)

[Operating environment 2](#_Toc1064273152)

[Hardware Requirements: 2](#_Toc621437509)

[Software Requirements: 2](#_Toc1457057056)

[System Features 3](#_Toc2033397806)

[1. Functional Requirements: 3](#_Toc1789732824)

[2. Non-Functional Requirements 3](#_Toc1209969659)

[System Models 4](#_Toc366687142)

[Entity Relationship Diagram: 5](#_Toc1732865421)

[Database Schema Diagram: 5](#_Toc578917280)

[Data Dictionaries: 6](#_Toc1506857570)

[Test Plan 8](#_Toc367369030)

[Objectives & Scope of Testing 9](#_Toc594440755)

[Test Strategy 9](#_Toc1229154028)

[Test environment 9](#_Toc284812457)

[Test Schedule 9](#_Toc1859359352)

[Test Deliverables 10](#_Toc1417804209)

[Test team roles and responsibilities 10](#_Toc1901500304)

[Test case design 10](#_Toc242253225)

[Test execution 11](#_Toc2027093216)

[Test data 11](#_Toc24172974)

[Test Metrics 11](#_Toc575513012)

[Defect Management 11](#_Toc820733360)

[Risk Management 11](#_Toc670286790)

[Test completion criteria 11](#_Toc1262950946)

[Sign-off and approval 12](#_Toc530656328)

[Unit Testing 12](#_Toc1769284878)

[Integration Testing 14](#_Toc1722163533)

[System Testing 15](#_Toc155350142)

[User Manual 15](#_Toc1633468394)

[Home Page 15](#_Toc799139071)

[Customer Page 16](#_Toc1636374765)

[Vehicle Information Page 21](#_Toc148734700)

[Rentals Screen 23](#_Toc422395667)

[Rentals Report Page 27](#_Toc1971408127)

[Customer Rentals Page 27](#_Toc1732439279)

[Available Vehicles Page 28](#_Toc1689079561)

[Late Returns Page 29](#_Toc1521059667)

[Turnover Report Page 29](#_Toc701734604)

[Appendix 30](#_Toc580460971)

[Future Additions 30](#_Toc869791111)

# Introduction

This document examines the various user requirements and system requirements (functional and non-functional) for the CRMS. The Non-Functional Requirements contain areas regarding Security, Usability, Performance, Compatibility and Scalability. Regarding the Non-Functional Requirements, there are various constraints being addressed to ensure quality in the system.

## Purpose

To design and implement a fully functioning web application for a Car Rental Management System. The CRMS will automate crucial functions for the automobile rental business, enhancing operational effectiveness. By offering a user-friendly interface for making reservations, maintaining bookings, and tracking rental history, the CRMS will also improve the client experience.

## Intended Audience

The CRMS is designed for use by the client. Additionally, customers of the car rental company will also interact with the system to make reservations, manage bookings, and view rental history.

## Project Scope

The scope of the CRMS project includes the design, development, and testing of a web-based application for managing the car rental company's fleet of vehicles, and customer bookings. The system will provide features such as; a vehicle information screen to help accurately identify and track each vehicle in the car rental company’s fleet, a customer information screen to keep detailed records of each customer who rents a car from the rental company, and a rental/return screen that allows users to manage the rental process and track the return of rental vehicles.

# Overall Description

## Product perspective

The CRMS may also integrate with existing systems used by the car rental company, such as accounting software. The CRMS is a standalone web application that will be used by the client and accessed by customers. It will interact with a database for storing and retrieving data.

## Constraints

1. Time Constraints – The development and implementation of the CRMS is approximately 5 weeks.
2. Technology Constraints – The CRMS is constrained by the selected technologies including the database selected by the client.

## Assumptions

1. The design team counts on the car rental agency to offer up-to-date and correct information for the system, including specifics on the vehicle, the client, and rental pricing.
2. The CRMS anticipates that the CRMS program and database will be given the required permissions and access privileges, including read and write access to pertinent folders, files, and database tables.
3. The CRMS anticipates that the underlying hardware and software meet the recommended operating environment as specified in the “Operating Environment” section of the document.

## Operating environment

### Hardware Requirements:

Server - Dual-core processor, 2.0 GHz or higher, with a minimum of 4 GB RAM.

Storage - Minimum 100 GB free disk space for application and database storage.

Network - Stable internet connection.

### Software Requirements:

Operating System – Windows Server 2016 or later

Database Server – MySQL version 8.0

Web Browser - Google Chrome, Mozilla Firefox, Microsoft Edge, or Opera for accessing the CRMS web interface.

# System Features

## Functional Requirements:

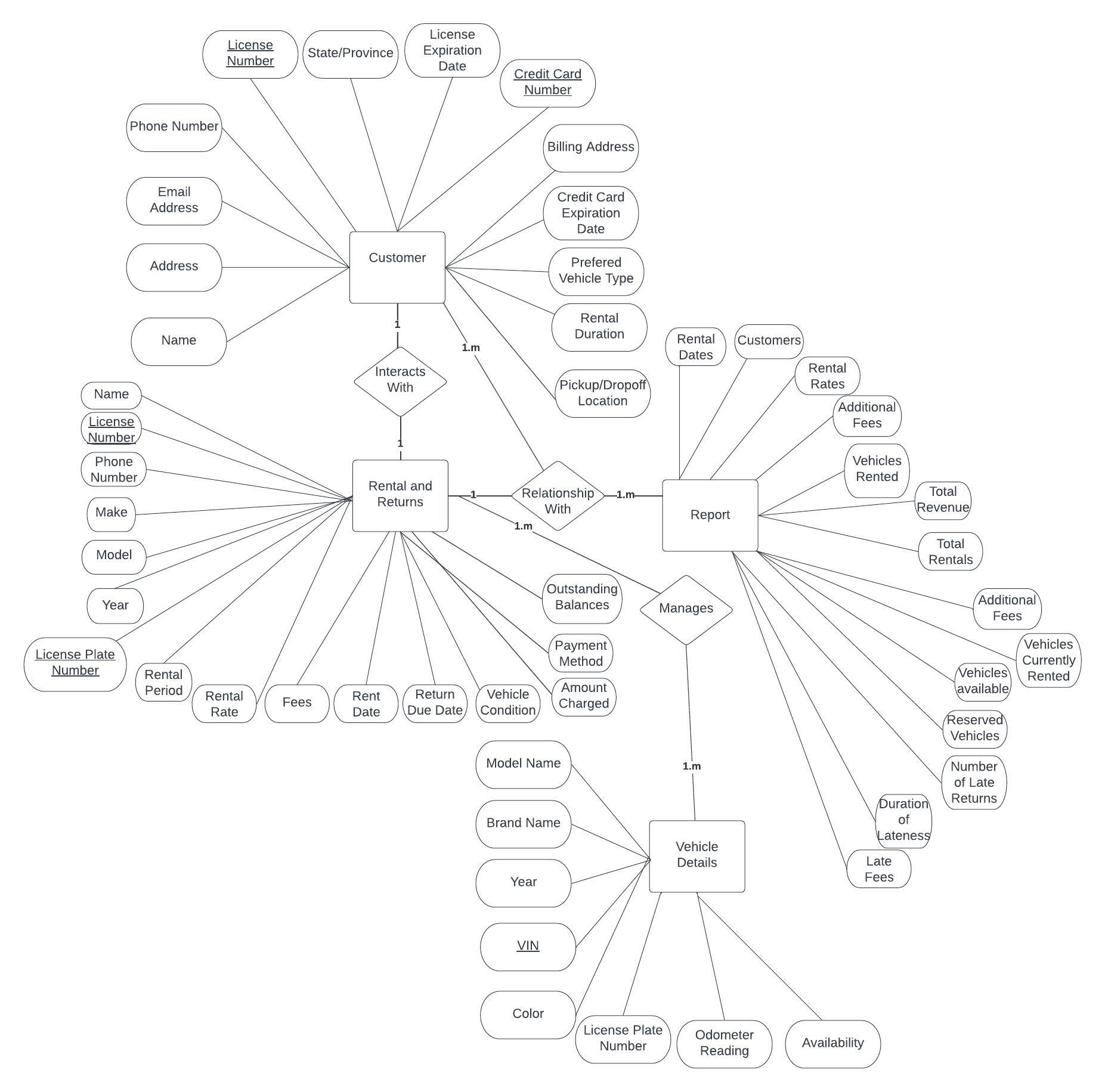
* 1. The system should allow for the client to add, update, and delete vehicles from the inventory.
  2. The system should be able to track vehicle details such as make, model, year, VIN, color, license plate number, odometer reading and availability.
  3. The system should allow customers to rent a vehicle.
  4. The system should allow the customers to cancel a reservation of a vehicle.
  5. The system should generate reports and analytics on various aspects of the car rental operations such as vehicle rental history, customer rental history, rental history, vehicle availability, and late return.
  6. The system should be able to track customer details such as contact information, driver’s license information, payment information, and rental preferences.
  7. The system should be able to track rental/returns information such as
     1. Customer information: Their first and last name, email, phone number, address, driver's license number, license issue location, license expiration date, credit card number, the name on the credit card, the credit card verification number, the expiration date of the card as well as the preferred rental pickup & drop-off locations and their preferred vehicle
     2. Vehicle information: The vehicle’s brand, model, vehicle identification number, odometer reading, interior & exterior color, odometer reading, license plate number and the condition of the vehicle.
     3. Rental information: The information on the associated customer & vehicle, the rental rate, lend date, return date, the lent & return condition, the payment method, and any extra charges.

## Non-Functional Requirements

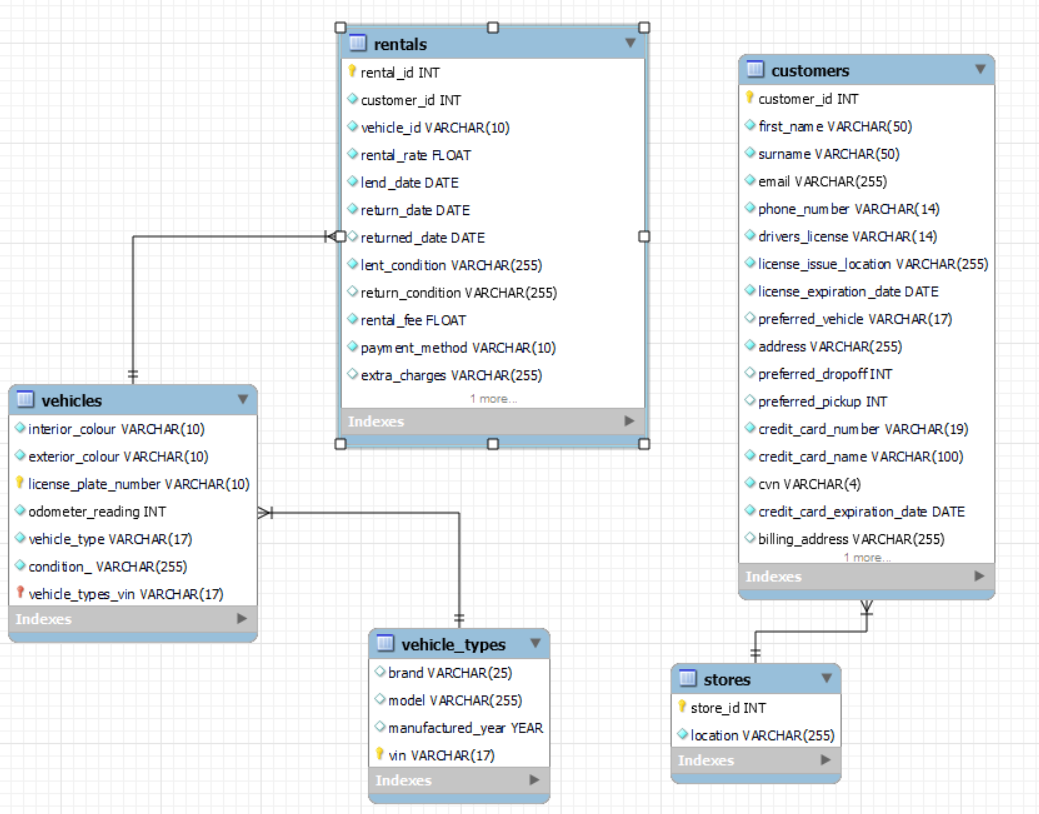
* 1. Performance
     1. The system should provide response times within acceptable limits.
  2. Security
     1. The stem should implement appropriate security measures to protect against data integrity issues.
  3. Usability
     1. The system should have a user-friendly interface with intuitive navigation.
     2. The system should have clear error messages.
     3. The system should have informative prompts.
     4. The system should be accessible to users with disabilities.
  4. Reliability
     1. The system should be available 24/7 with minimal downtime.
  5. Scalability
     1. The system should be designed to handle increasing volumes of data.
  6. Compatibility
     1. The system should be compatible with standard web browsers, operating systems, and devices used by the intended audience.

# System Models

## Entity Relationship Diagram:



## Database Schema Diagram:



## Data Dictionaries:

Vehicles Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Primary/Foreign Key | Description |
| Interior\_colour | Varchar(10) |  | Interior colour of vehicle |
| Exterior\_colour | Varchar(10) |  | Exterior colour of vehicle |
| License\_plate\_number | Varchar(10) | Primary Key | License plate number of the vehicle |
| Vehicle\_type | Varchar(17) | Foreign Key references Vehicle Type’s VIN | The vehicle type |
| Condition\_ | Varchar(255) |  | Condition of the vehicle |
| Odometer\_Reading | int |  | The millage of the vehicle |

Vehicle Types Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Primary/Foreign Key | Description |
| brand | Varchar(25) |  | Brand of the vehicle |
| model | Varchar(255) |  | Model of the vehicle |
| Manufactured\_year | year |  | Year the vehicle was manufactured |
| vin | Varchar(17) | Primary key | Vehicle identification number |

Customers Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Primary/Foreign Key | Description |
| Customer\_id | int |  | ID of the customer |
| First\_name | Varchar(50) |  | First name of the customer |
| surname | Varchar(50) |  | Last name of the customer |
| email | Varchar(255) |  | Customer’s email |
| Phone\_number | Varchar(14) |  | Customer’s contact number |
| Drivers\_license | Varchar(14) |  | Customer’s driver license number |
| License\_issue\_location | Varchar(255) |  | License issue location |
| License\_expiration\_date | date |  | License expiration date |
| Preferred\_Vehicle | Varchar(17) | Foreign key references the vehicle types table’s vin | Customer’s preferred vehicle |
| address | Varchar(255) |  | Customer’s address |
| Preferred\_dropoff | int | Foreign key references the stores table store\_id | Customer’s preferred dropoff |
| Preferred\_pickup | int | Foreign key references the stores table store\_id | Customer’s preferred pick up |
| Credit\_card\_number | Varchar(19) |  | Customer’s credit card number |
| Credit\_card\_name | Varchar(100) |  | Customer’s credit card name |
| cvn | Varchar(4) |  | Customer’s cvn |
| Credit\_card\_expiration\_date | date |  | Customer’s credit card expiration date |
| Billing\_address | Varchar(255) |  | Customer’s billing address |

Stores table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Primary/Foreign Key | Description |
| Store\_id | int | Primary key | Stores ID |
| location | Varchar(255) |  | Stores location |

Rentals table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Primary/Foreign Key | Other |
| Rental\_id | int | Primary key | The rent ID |
| Customer\_id | int | Foreign key references the customer’s table customer\_id | The Customer ID who rented |
| Vehicle\_id | Varchar(10) | Foreign key references the vehicle’s table license plate | The ID of the vehicle that was rented |
| Rental\_rate | float |  | The rental rate |
| Lend\_date | date |  | The lend date of the vehicle |
| Return\_date | date |  | The date the vehicle should be returned. |
| Returned\_Date | date |  | The date the vehicle was returned. |
| Lent\_condition | Varchar(255) |  | The condition of the vehicle when lent out |
| Return\_condition | Varchar(255) |  | The condition of the vehicle when it was returned. |
| Rental\_fee | float |  | The rental fee |
| Payment\_method | Varchar(10) |  | Either the string “cash” or "card” |
| Extra\_charges | Varchar(255) |  | Json string with the keys representing the charge name and the value the associated fee |

# Test Plan

## Objectives & Scope of Testing

The objectives of testing for the CRMS application are as follows:

* Validate that the application meets the functional requirements provided by the client, including the ability to enter and store vehicle information, customer information, and rental preferences.
* Verify that the application is user-friendly and provides a seamless user experience for both the administrative staff and customers.
* Ensure that the application is secure and protects sensitive information, such as customer payment details and driver's license information.

The scope of testing for the CRMS application includes the following areas:

* Vehicle Management: Testing the ability to enter and store vehicle information, including make and model, year, VIN, color, license plate number, odometer reading, and availability status.
* Customer Management: Testing the ability to enter and store customer information, including contact information, driver's license information, payment information, and rental preferences.
* Rental and Returns: Testing the ability to create and manage rental bookings, process rental payments, track vehicle availability, and manage vehicle returns.
* Security: Testing the application's security measures, such as authentication.

## Test Strategy

The following types of testing will be performed during the testing process for the CRMS application:

* Functional Testing: Verify that the application meets the functional requirements provided by the client, including testing the ability to enter and store vehicle information, customer information, and rental bookings. Test cases will cover positive and negative scenarios, boundary conditions, and error handling.
* Usability Testing: Evaluate the application's user-friendliness and overall user experience, including testing the navigation, layout, and ease of use of the application. Usability testing will be performed by real users to gather feedback on the application's usability.
* Security Testing: Validate the application's security measures, including authentication.

## Test environment

The test environment for the CRMS application will consist of the following:

* Operating System: Windows 10
* Web Server: NodeJS HTTP Server
* Database Server: MySQL
* Browser: Google Chrome, Mozilla Firefox, Microsoft Edge

## Test Schedule

The testing schedule for the CRMS application is as follows:

* Test Planning: 2 days
* Test Case Development: 5 days
* Test Data Preparation: 2 days
* Test Execution: 10 days
* Defect Management: Ongoing throughout the testing process
* Test Reporting: Ongoing throughout the testing process

## Test Deliverables

The following deliverables will be produced as part of the testing process:

* Test Plan: This document outlines the testing approach, objectives, scope, test environment, and schedule.
* Test Cases: Detailed test cases that cover all the functional requirements of the application, including positive and negative test scenarios.
* Test Data: Sample test data that will be used during testing to simulate real-world scenarios.
* Test Reports: Regular test reports that provide updates on the testing progress, including test results, defects found, and their status.

## Test team roles and responsibilities

* Test Manager:
  + Develop the overall test strategy, test plan, and test schedule.
  + Define the roles and responsibilities of the test team members.
  + Allocate resources and monitor progress to ensure timely completion of testing activities.
  + Review and approve test deliverables, including test plan, test cases, and test reports.
* Testers
  + Execute test cases as per the test plan and report defects found during testing.
  + Validate that the application meets the functional requirements and other testing objectives.
  + Provide feedback on the usability, performance, and security aspects of the application.
* Client
  + Review deliverables and provide feedback

## Test case design

The following describes the methodology for designing test cases as part of the testing process:

* Identify and understand the functional requirements of the CRMS application.
* Design test cases based on the identified selection criteria, covering positive and negative scenarios, boundary conditions, usability, and security aspects.
* Use a combination of techniques, such as black-box testing, white-box testing, and grey-box testing, to design test cases that validate different aspects of the application.
* Define clear and concise steps for each test case, including inputs, expected results, and actual results.

## Test execution

Test execution will follow the test plan and test schedule, including test case prioritization and sequencing. Test cases will be executed based on the test environment and configuration specified in the test plan. Test results will be recorded and documented, including the actual outcome, observed defects, and any deviations from expected results.

## Test data

Test data will be identified and created to support test scenarios and test cases.

## Test Metrics

Test metrics will be collected and reported to measure the effectiveness of the testing process.

* Metrics such as test coverage, defect density, defect trend, and test execution progress will be captured and analyzed

## Defect Management

The following management techniques will be employed when handling defects:

* Defects discovered during testing will be logged.
* Defects will be assigned to the appropriate team member for resolution and will be tracked throughout the defect lifecycle, from detection to resolution and closure.
* Defects will be prioritized based on severity and priority levels.

## Risk Management

The following risks are identified during the testing process for the CRMS application, along with the corresponding mitigation strategies:

* Lack of test coverage: Mitigation strategy includes thorough test case development, including positive and negative scenarios.
* Time and resource constraints: Mitigation strategy includes effective test planning and scheduling, resource allocation, and prioritization of testing activities to optimize testing efforts within the given timeline and resources.

## Test completion criteria

The following criteria must be met before the testing process can be considered complete:

* All test cases identified in the test plan have been executed.
* All critical defects and high-priority defects have been resolved and retested.
* The application has passed all functional, usability, and security tests with acceptable results.
* Test reports have been reviewed and approved by the client.
* The application has met the defined quality criteria and is ready for production deployment.

## Sign-off and approval

The following procedures and criteria must be met for acceptance:

* Formal sign-off will be obtained from the client once the testing is completed, and the test results are reviewed and accepted.

# Unit Testing

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

The steps involved during Unit testing are as follows:

1. Preparation of the test cases
2. Preparation of the possible test data with all the validation checks
3. Complete code review of the module
4. Actual testing done manually
5. Modifications done for the errors found during testing
6. Prepared the test result scripts

Equivalence Class:

**VEHICLES:**

1. **Year: If the year is equivalent to 2023 (current year) then the car can be rented, and pricing is calculated based on cars’ current year’s rates. If the year is equivalent to 2022 (previous year) then the car can be rented, and pricing is calculated based on cars previous year’s rates. If the year is greater than 2023 (present) then the car cannot be rented, and the appropriate error message is displayed. If the year is lesser than 1985 (historic year) then then car can be rented with the pricing is calculated based on historical year rates.**

BVA values:

1. 2023
2. 2024
3. **VIN: Test if it is equal (good), greater than (error- too many digits) or less than (error- incomplete) 17 digits.**

BVA values:

1. 234-56789-8765-23451 (amount needed-17)
2. 452-76389-0745-2761 (one below 17 digits)
3. 524-17689-0254-326782 (one above 17 digits)
4. **License Plate #: Test if it is equal (good), greater than (error- too many digits) or less than (error- incomplete) 6 digits**

BVA values:

1. BA 6543
2. MA 76432 (one above 6 digits)
3. PT 452 (one below 6 digits)
4. Odometer Reading: Check if it is equal to 200,000 miles (warning- needs to be serviced), greater than 200,000 miles (about 321868.8 km) (error- urgent service needed), less than 200,000 miles by 100,000 miles amount (good to drive) or less than 200,000 miles by 20,000 amount (will need servicing soon)

CUSTOMERS

1. **Payment Information: Test if the year input for expiration is equal to or greater than the present year. If less than give an error, if equal to (based on the month and how long the car is being rented) give a warning and if greater than give the “go ahead.”**

BVA Values:

1. 2022 (less than current year)
2. 2024 (greater than current year)
3. 2023 (current year)(try for February (gone) and December (greater than current date))
4. **Driver’s License Information: Test if the year input for expiration is equal to or greater than the present year. If less than give an error, if equal to (based on the month and how long the car is being rented) give a warning and if greater than give the “go ahead.”**

BVA Values:

1. 2022 (less than current year)
2. 2024 (greater than current year)
3. 2023 (current year)(try for February (gone) and December (greater than current date))
4. **Contact Info (phone number): Check if it is 10 digits (first 3 for the area the # was provided and 7 more for the unique phone #). If greater than 11 (not including the possible 1 people might put), then give an error. If less than 10, give incomplete error. If greater than 11, if error of too much digits being entered.**

BVA Values:

1:

1. Debit card #: Test if it is equal (good), greater than (error- too many digits) or less than (error- incomplete) 16 digits.
2. CVV: Test if it is equal (good), greater than (error- too many digits) or less than (error- incomplete) 3 digits.

RENTALS AND RETURNS

# Integration Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case No. | Test Case Name | Purpose | Precondition | Test Steps | Expected Results |
| 01 | Customer reservation | Enable customer to be entered into the system  (Basic information on the customer renting the vehicle, including their name, contact information, and driver's license number.) | Customer has valid driver’s license |  | customer will be entered into the system |
| 02 | Available Vehicle Information | For customers to view information on the vehicle being rented or returned, including the make, model, year, color, and license plate number. | 1. Vehicle not rented  2. customer logged in | Click to see available vehicles | customer able to view descriptions of available vehicles |
| 03 | Vehicle Condition | For customers to have information on the condition of the vehicle, including any existing damage or issues, and any new damage or issues noted at the time of rental or return | customer logged in |  | Customers able to check all vehicle information (condition of the vehicle, including any existing damage or issues, and any new damage) |

# System Testing

All features will be tested to ensure that all elements of the integrated system function properly.

1. Integration of all the forms/modules in the system
2. Preparation of the test cases
3. Preparation of the possible test data with all the validation checks
4. Modifications done for the errors found during testing

# User Manual

## Home Page

When a user first visits the website they will be taken to the page shown below in figure 1. From this page users can get to any other of the pages; the customer, vehicle, rentals or any of the various report pages by clicking the buttons on the left. The sidebar can be collapsed and reopened as shown in figures 1 & 2 by clicking on the top part by the arrow or the word CRMS. The reports dropdown can also be collapsed by clicking on it as shown in figure 3.

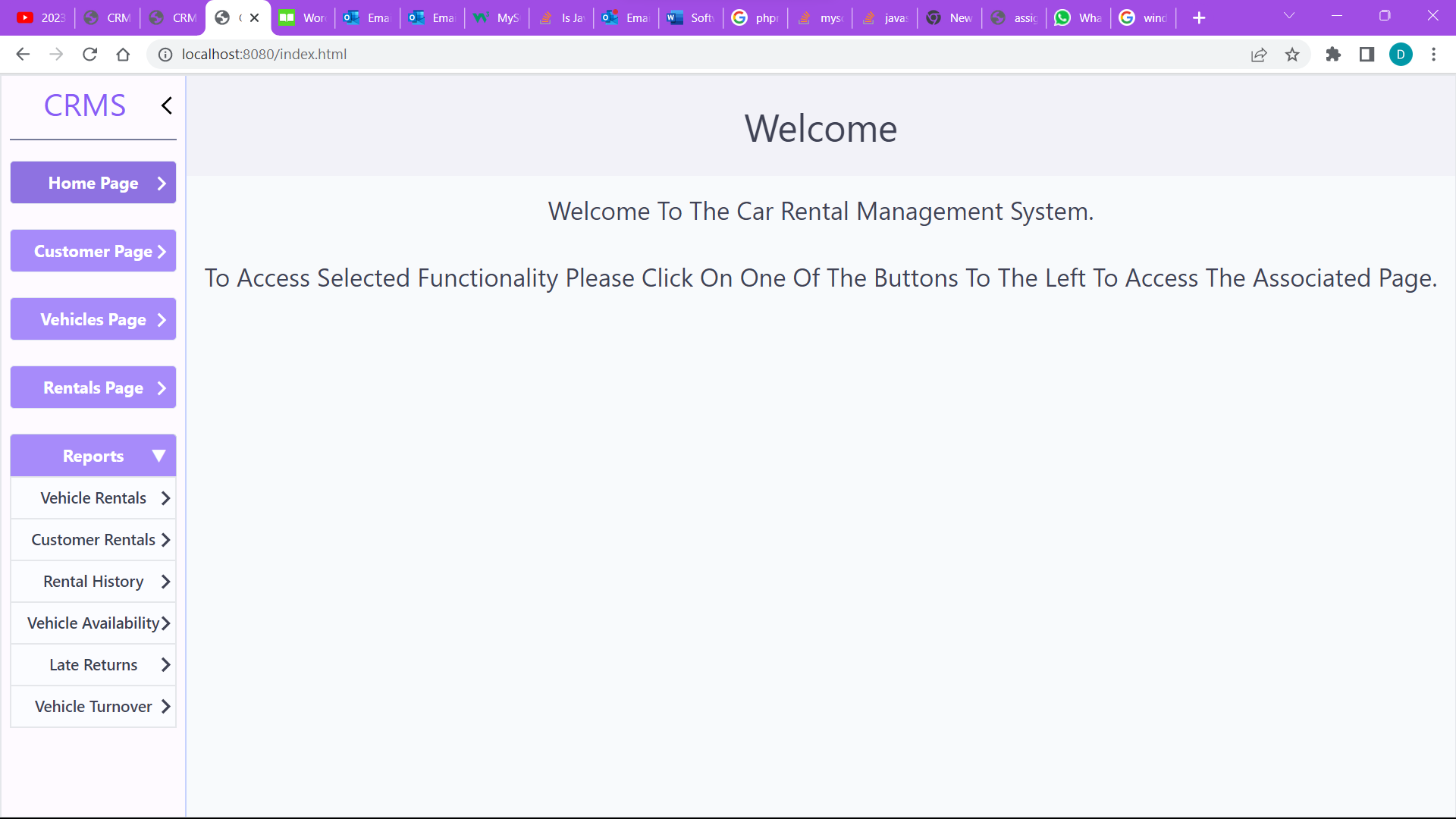


Figure 1 Image of The Home Screen

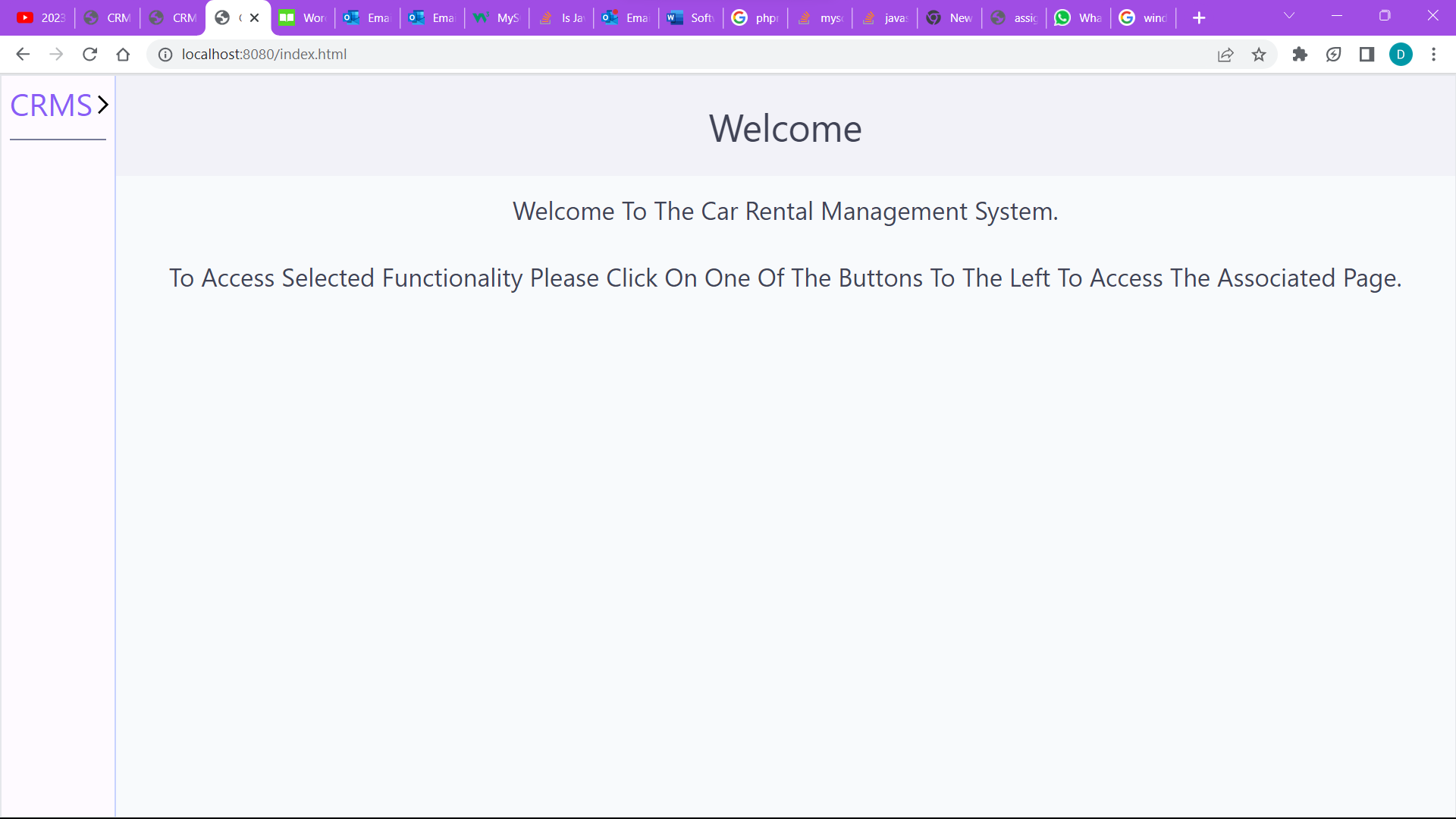


Figure 2 Image Showing the Sidebar Collapsed.

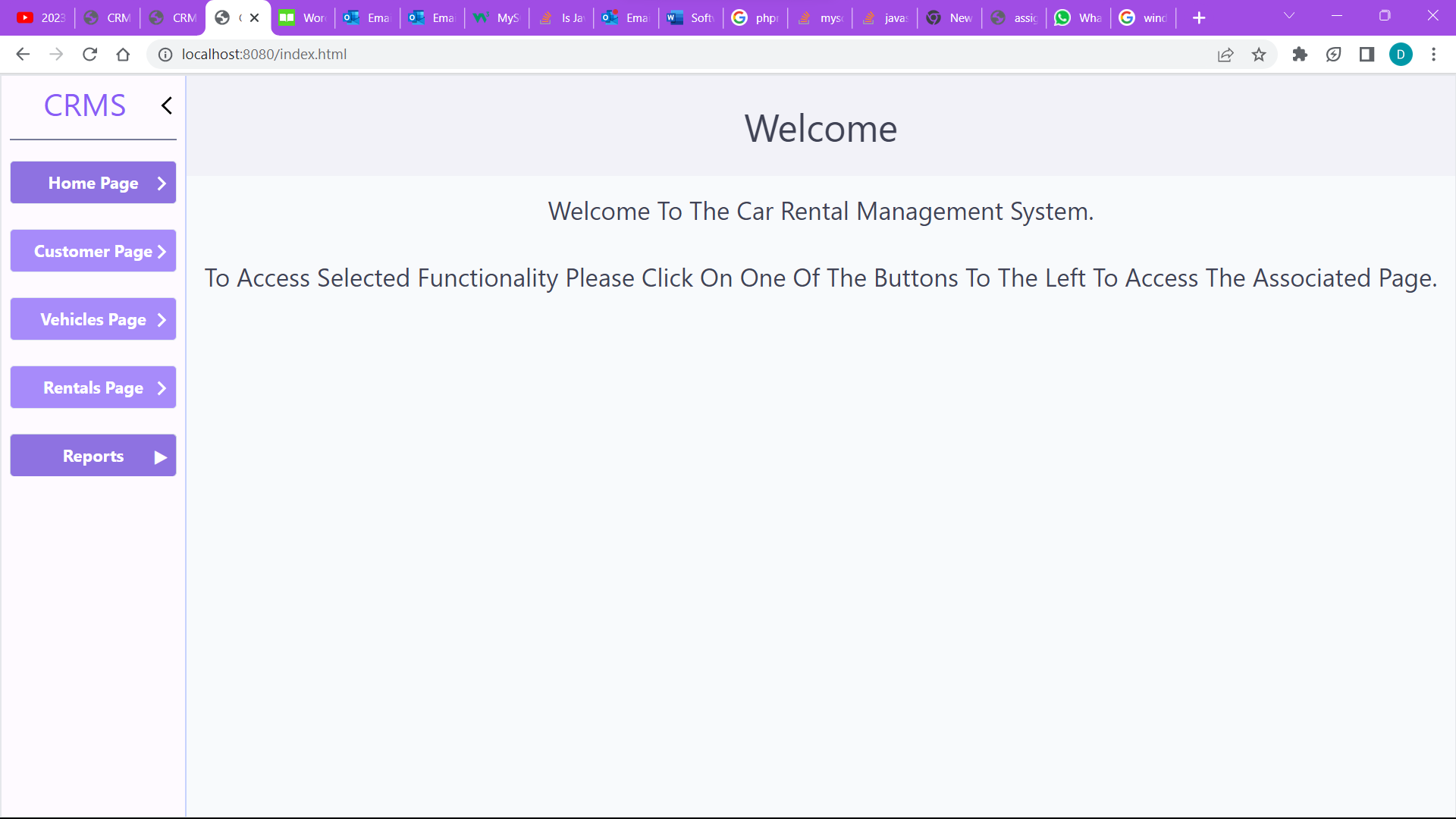


Figure 3 Image Showing the Reports Dropdown Collapsed.

## Customer Page

When the use clicks the customer page button they will be taken to the page shown below in figure 4. On this page users can view information on all customers that have data stored in the system. Initially some column groups are collapsed and by clicking on the heading (customer details, driver’s license details, payment details or rental preferences) the table columns will be displayed as shown in figure 5.

By entering a user’s first and/or surname separated by a space in the search box with the text “Search Customer Name” and clicking the searhc icon, users can search for specific customer records that match the data entered. All matching records would be displayed in the table as shown in figure 4. To search for more specific customer records the user can click the filter button next to the search box and the dropdown menu as shown in figure 6 will be displayed. From There by clicking on any of the boxes under “Field Names” in the filter box users can select either the customer’s first name, email, phone number or surname (as seen in figure 7) to filter any customer records that match the entered information. After selecting which field to filter on users could enter the associated value in the box to the left of the selected field name and then click apply filters to search the database for any records that match. All associated records would then be displayed in the table. The user can filter any combination of the criteria.

By clicking the view all button all customer records are displayed.

# 

Figure 4. Image Showing the Customer Page

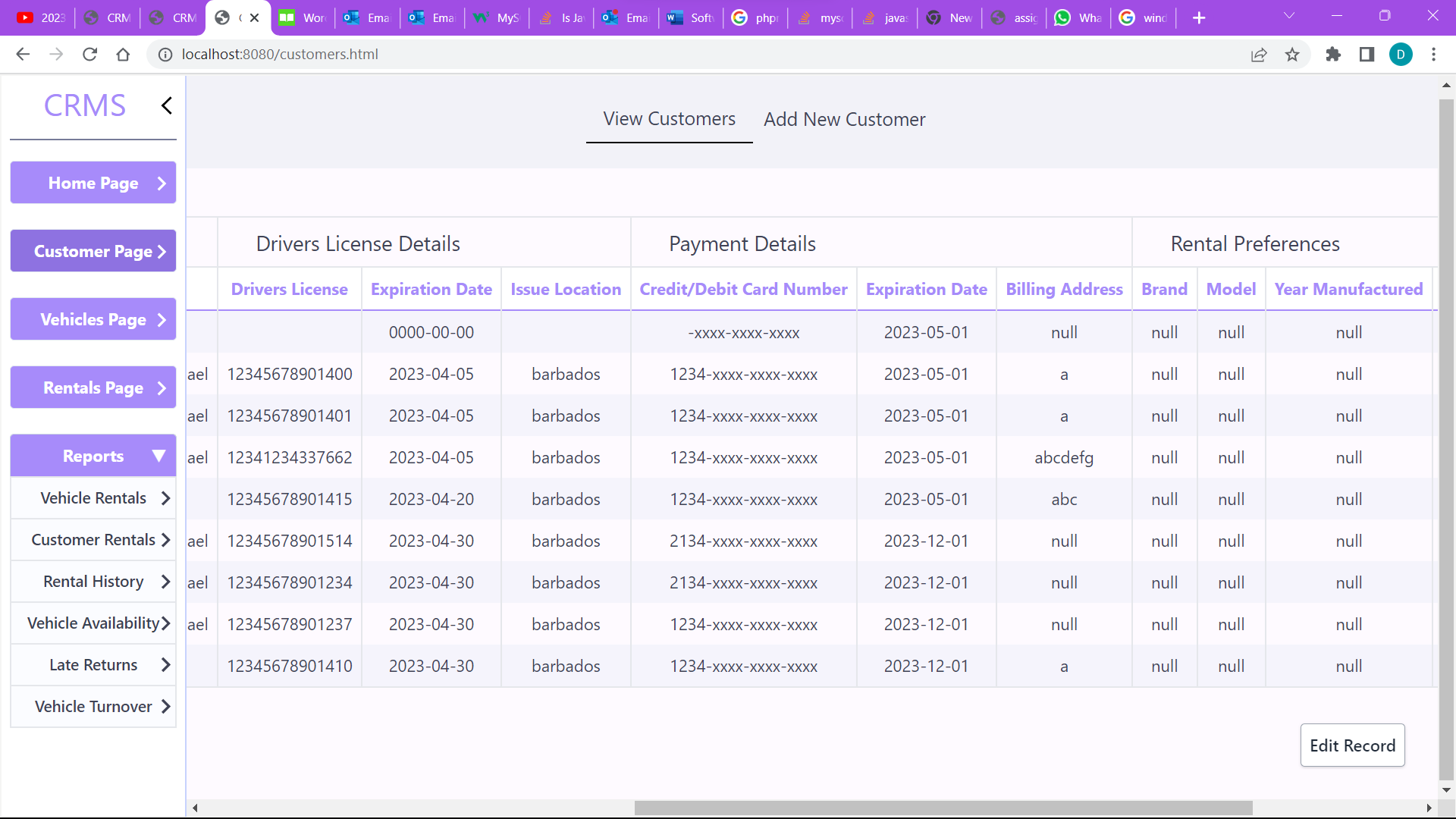


Figure 5. Image Showing The Column Groups Opened.

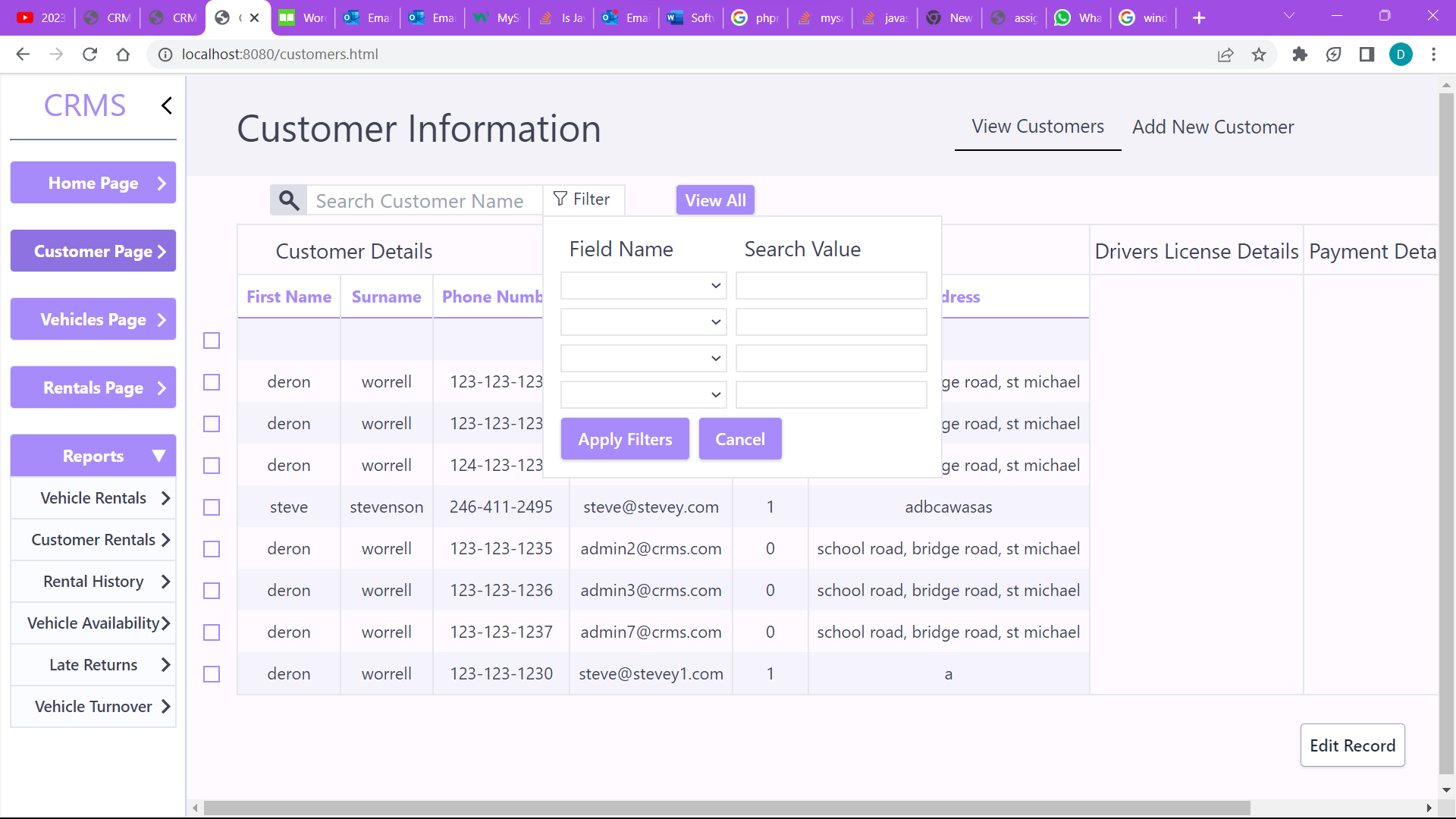


Figure 6. Image Showing The Dropdown Displayed When The Filter Button Is Clicked.

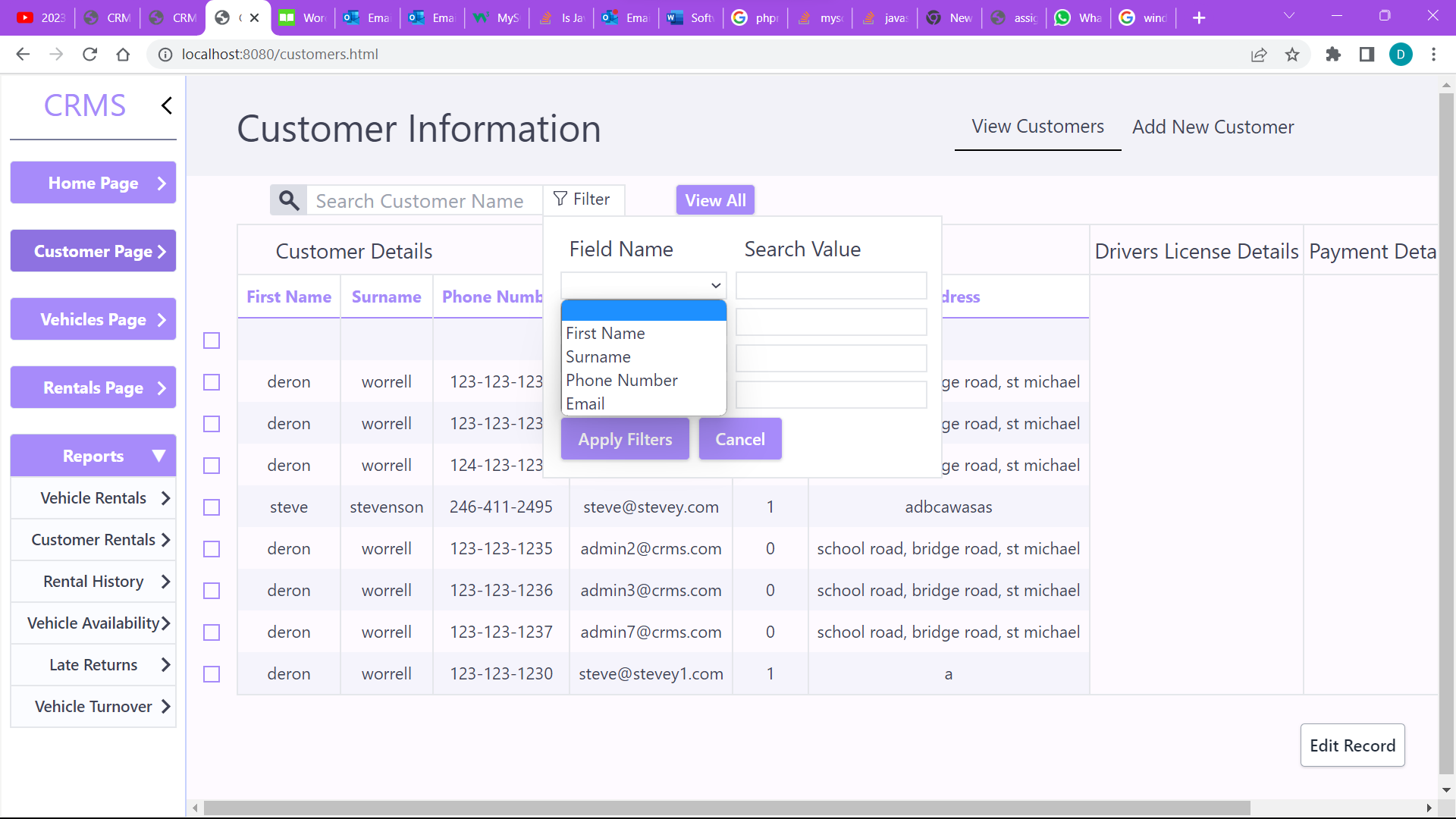


Figure 7: Image Showing The Filter Options When A Field Name Box Is Clicked.

When the add new customer button at the top of the screen is pressed users are taken to the form in figures 8 & 9 where they can enter customer information and save it in the database by clicking the submit button or to reset the form they can click the reset button at the bottom of the form as shown in figure 9. For the valid values for each field please see table 1 in the appendix.

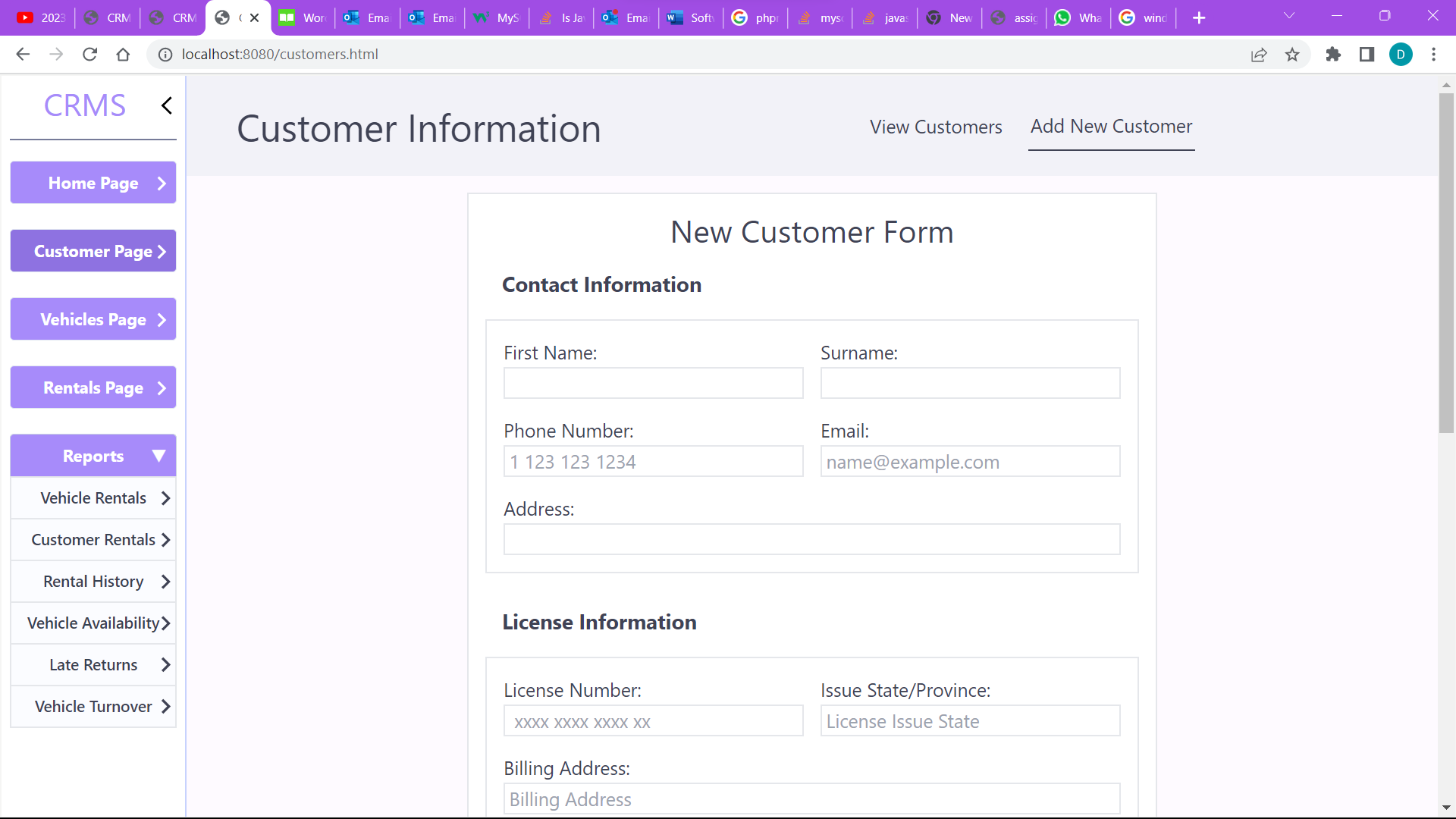


Figure 8: Image Showing The Top Half Of The Customer Form

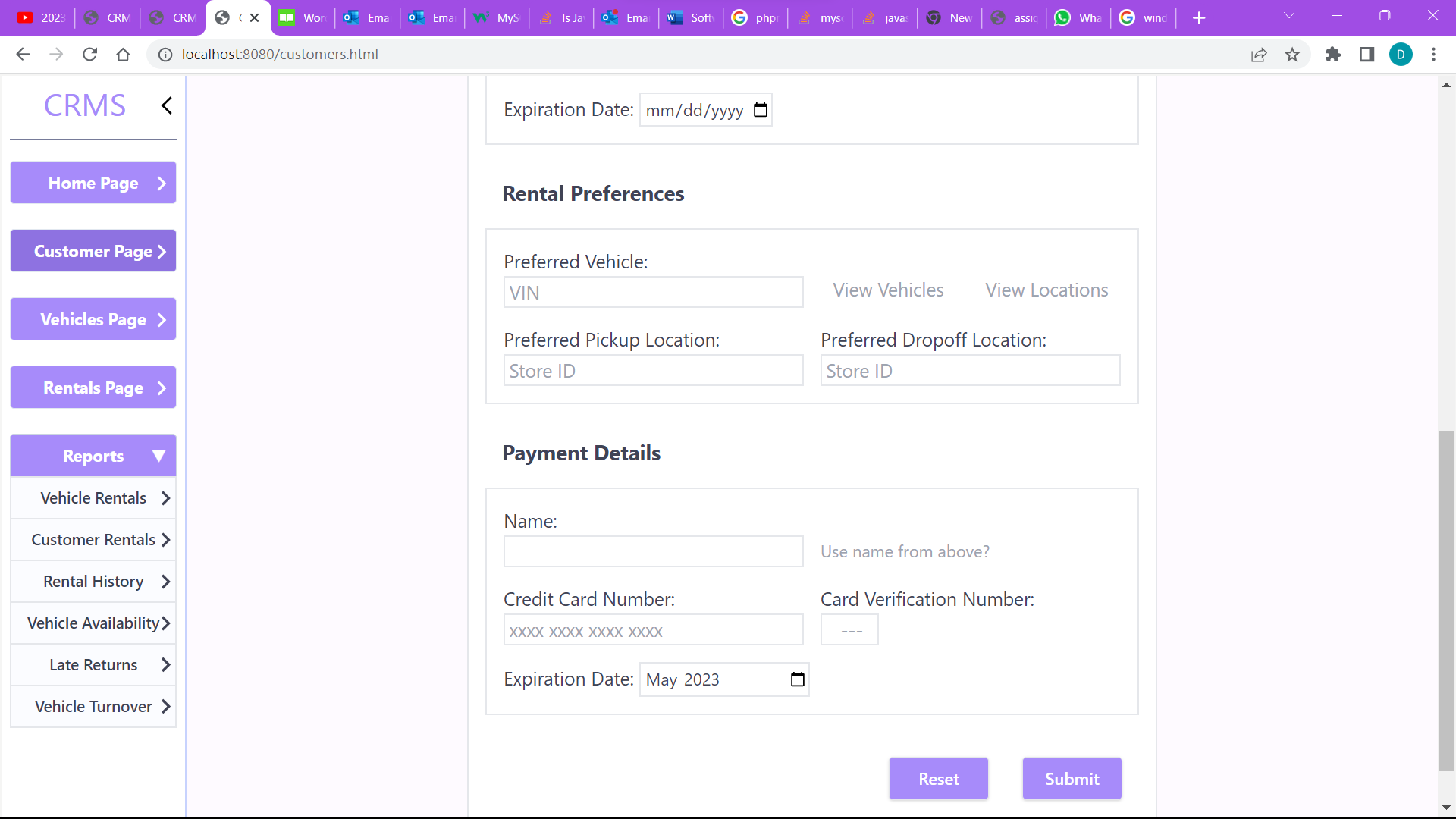


Figure 9: Image Showing The Bottom Half Of The Customer Form

By clicking on one of the checkboxes to the left of the form the user (figure 10) the user can select a row, settings the text colour to purple. After a row the user can press the edit record button and be taken to the page In figure 11 where they can edit customer data. The associated data of the selected row is displayed in the form for users to edit and when they scroll to the bottom of the form they can click on the submit button like in figure 9 above to save all edits or the reset button which reverts all the users changes to the data in the form.

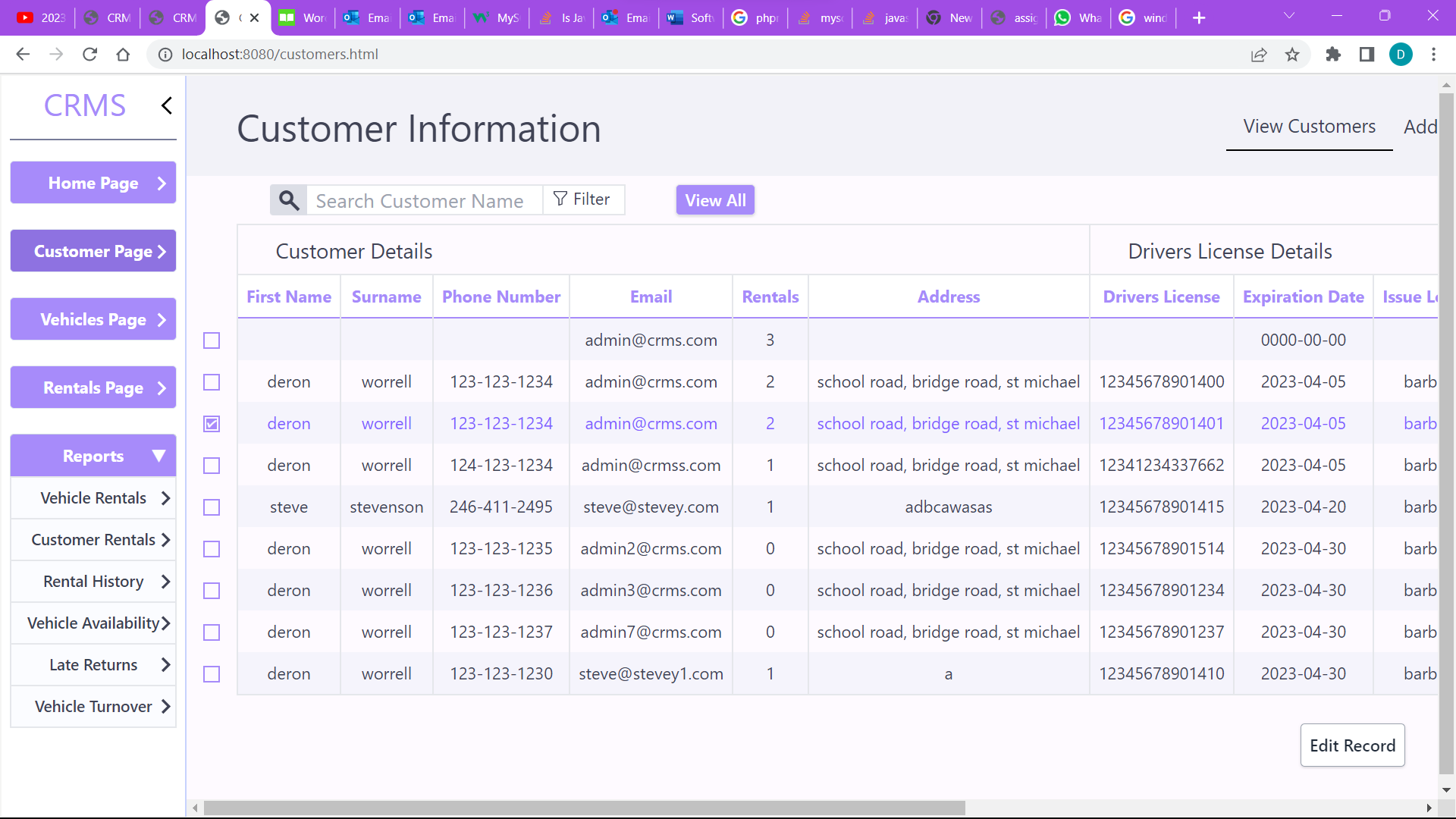


Figure 10: Image Showing A Customer Record Selected

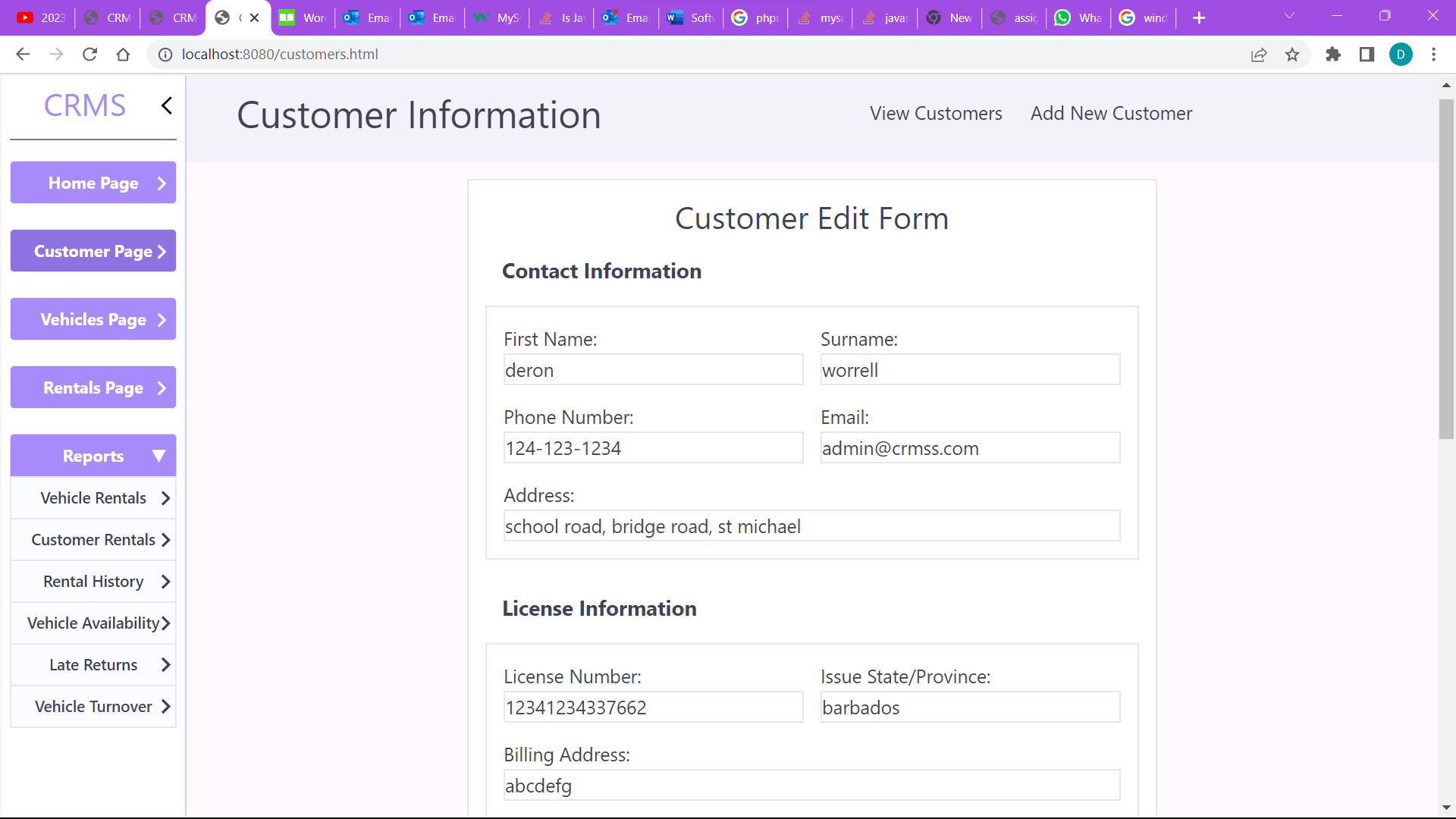


Figure 11: Image Showing The Edit Page

## Vehicle Information Page

By clicking the vehicles page button in the sidebar the users are taken to the vehicles page where they can add new vehicle records, vehicle type records, view the fleets vehicles and the types of vehicles in the fleet as well as edit any of this information. By clicking the add new vehicle text at the top of the screen the user would be taken to the page in figure 13 where they can store new vehicle data. If they click the add new vehicle model, they would be taken to the page in figure 15 where they can add type information on the various vehicles (the brand, model and related information). Lastly, by clicking the view vehicle models text they would be taken to the page in figure 14 where they can view information on the specific vehicle models. By selecting a checkbox on the right as explained above users can edit the related data of the chosen record with the data being filled into the forms below depending on the type. Additionally, the column groups can be collapsed or revealed as explained above by clicking on the title, by default the model details group is closed.

# 

Figure 12

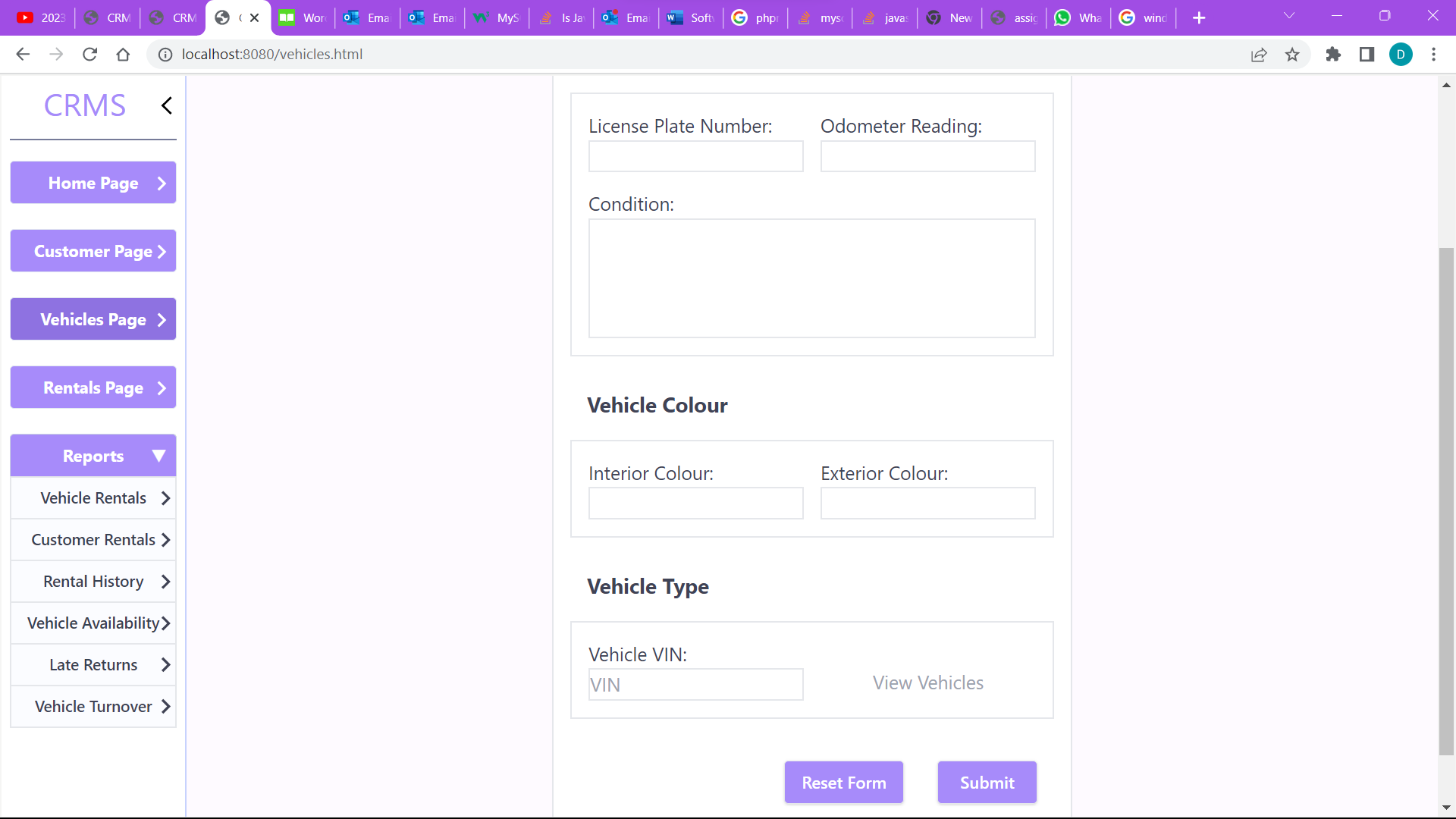


Figure 13

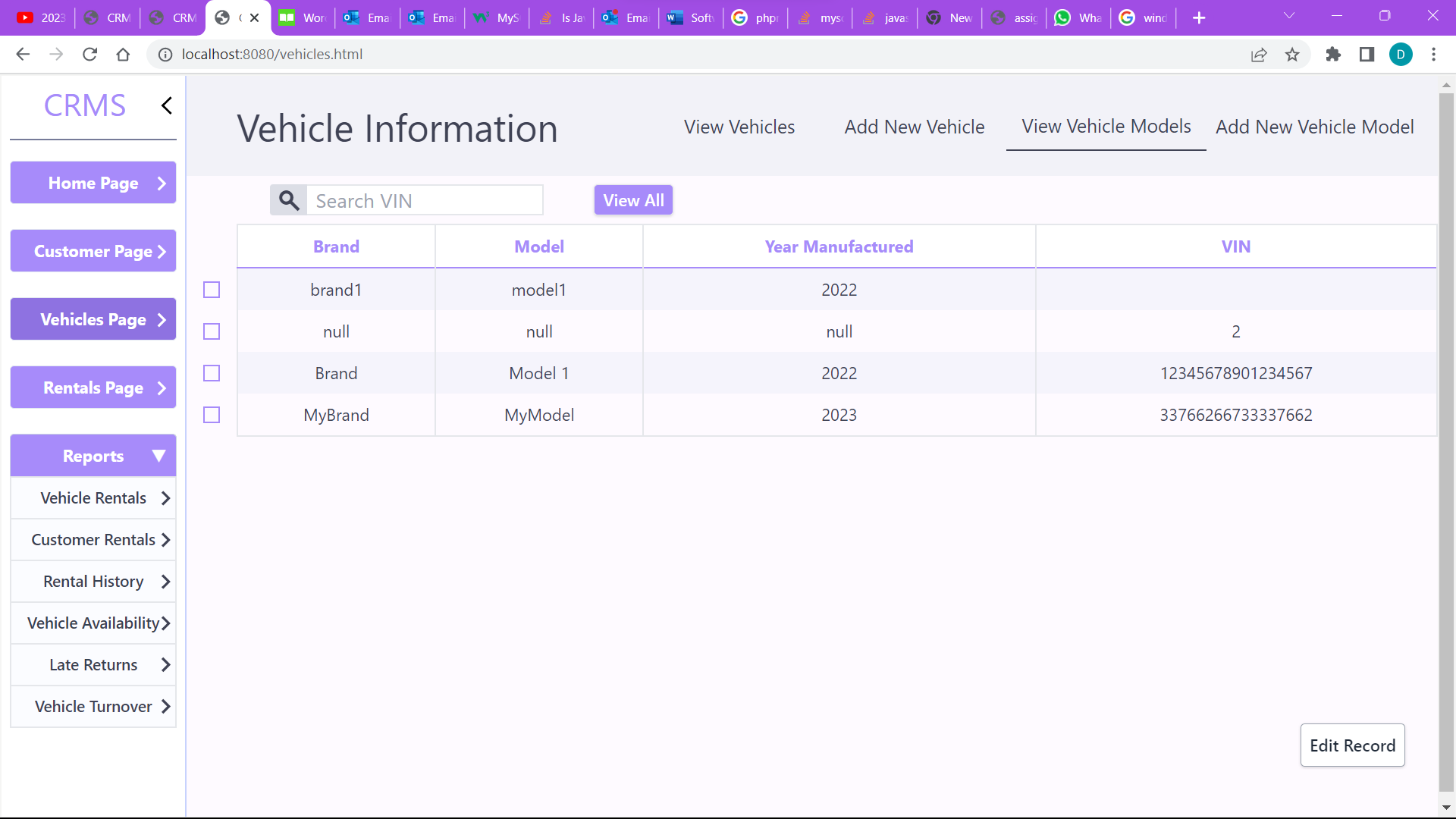


Figure 14

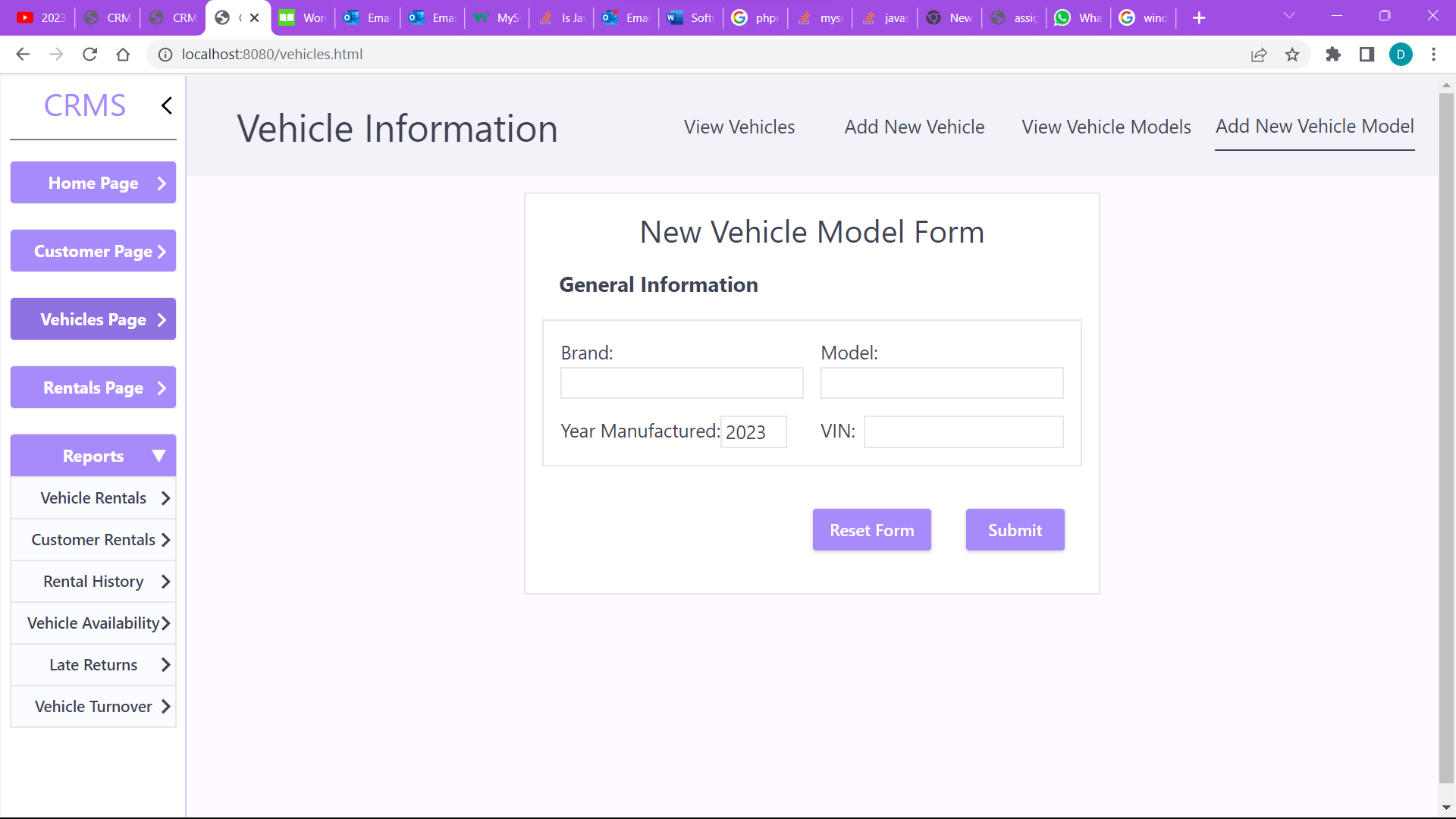


Figure 15

## Rentals Screen

By Clicking on the rentals page button the user would be taken to the rentals screen where they can view or edit rentals records as well as add new rentals to the system as shown in figure 16.

The filter on the rentals page is similar to the one on the customers page allowing users to filter by the lent date, return date or license plate but users can also choose the filter type that is if the filter field’s value is greater than, lesser than or equal to the entered value. When the user selects a field to filter they can then select which filter type to use by default the lent date’s filter type is greater than or equal to, the return date’s is lesser than or equal to and the license plate field is can only be filtered for values equal to it as seen in figures (16,17,18,19). They can be filtered on any permutation of the criteria.

# 

Figure 16: The Rentals Page

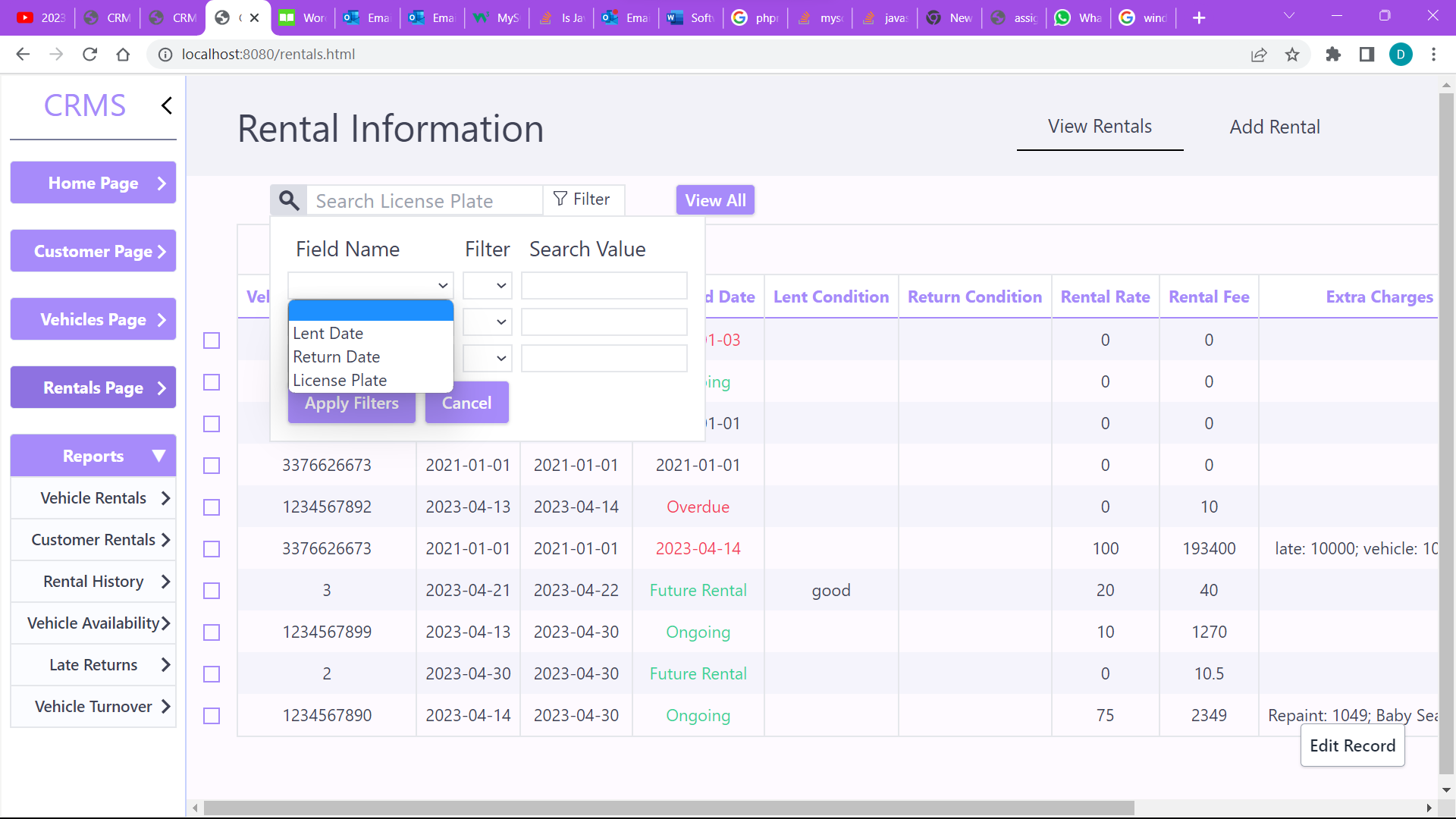


Figure 17: The Filter On The Rentals Page.

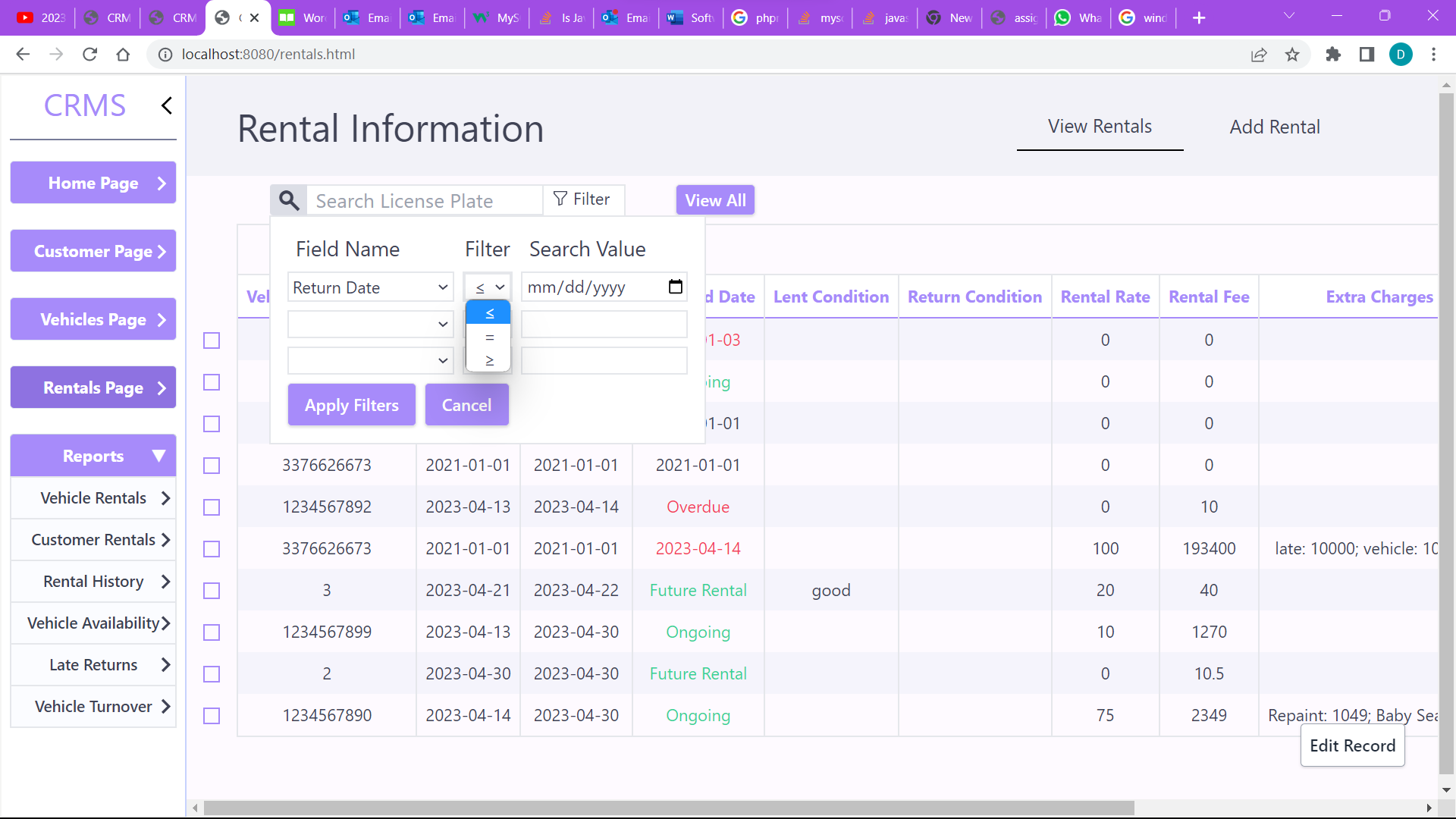


Figure 18: Image Showing The Various Filter Options For A Field.

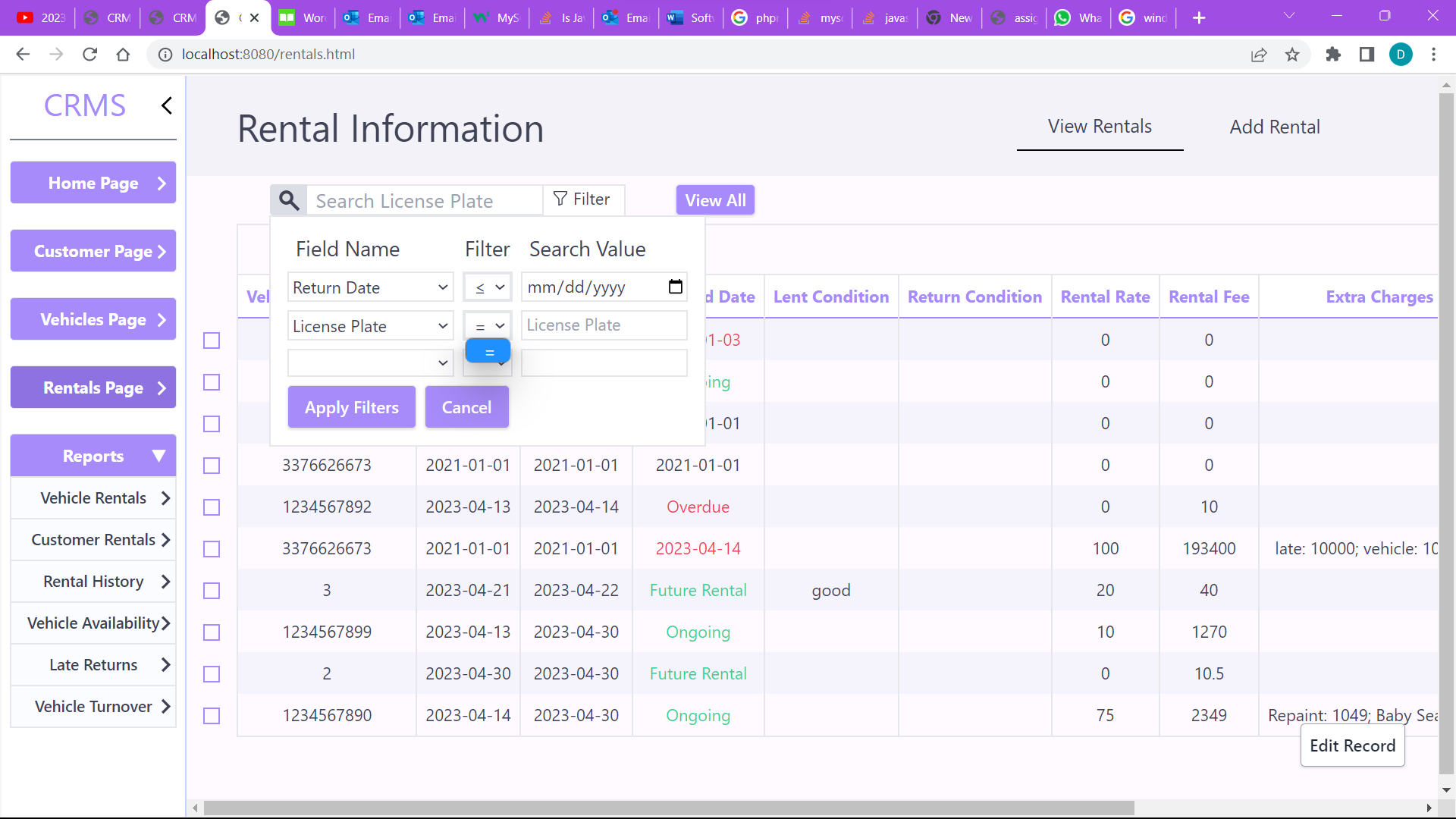


Figure 19: Image Showing The Various Filter Options For A Field.

By clicking on the add rental text in the top right users would be taken to the rental form where they can add new rental data. Users can enter information on new rentals. By clicking the show customers text next to the customer id field the popup in figure 25 will be displayed showing all customer records of customers who aren’t currently renting a vehicle. By clicking on the id of a customer record the customer id field of the form is automatically filled. If a user clicks the “Show Vehicles” text all vehicles that aren’t rented out are shown in a popup as seen in figure 24. From there the user can click on the vehicle\_id to fill the vehicle license plate of the rental.

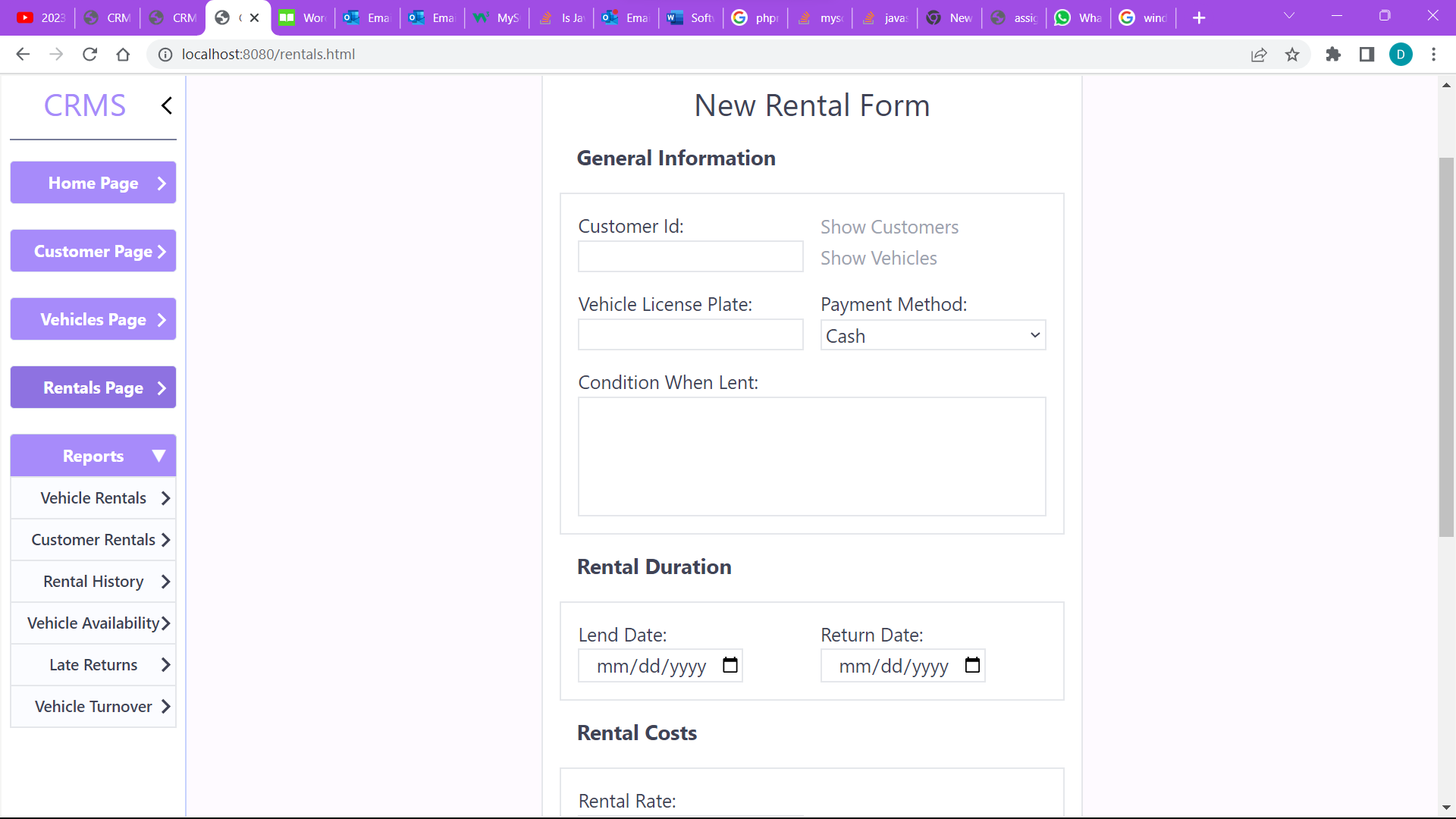


Figure 20. The Rental Form (1)

When the user scrolls down to the page the user can enter any optional extra fees for the rentals. By clicking the add fee button they can add more fees as shown in figure 22 or by clicking the purple x to the right of any fee the user can remove it. As the user updates extra fee information, the rental rate or the rental period (the lend & return dates) the rental fee field is automatically calculated and filled in based on this information.

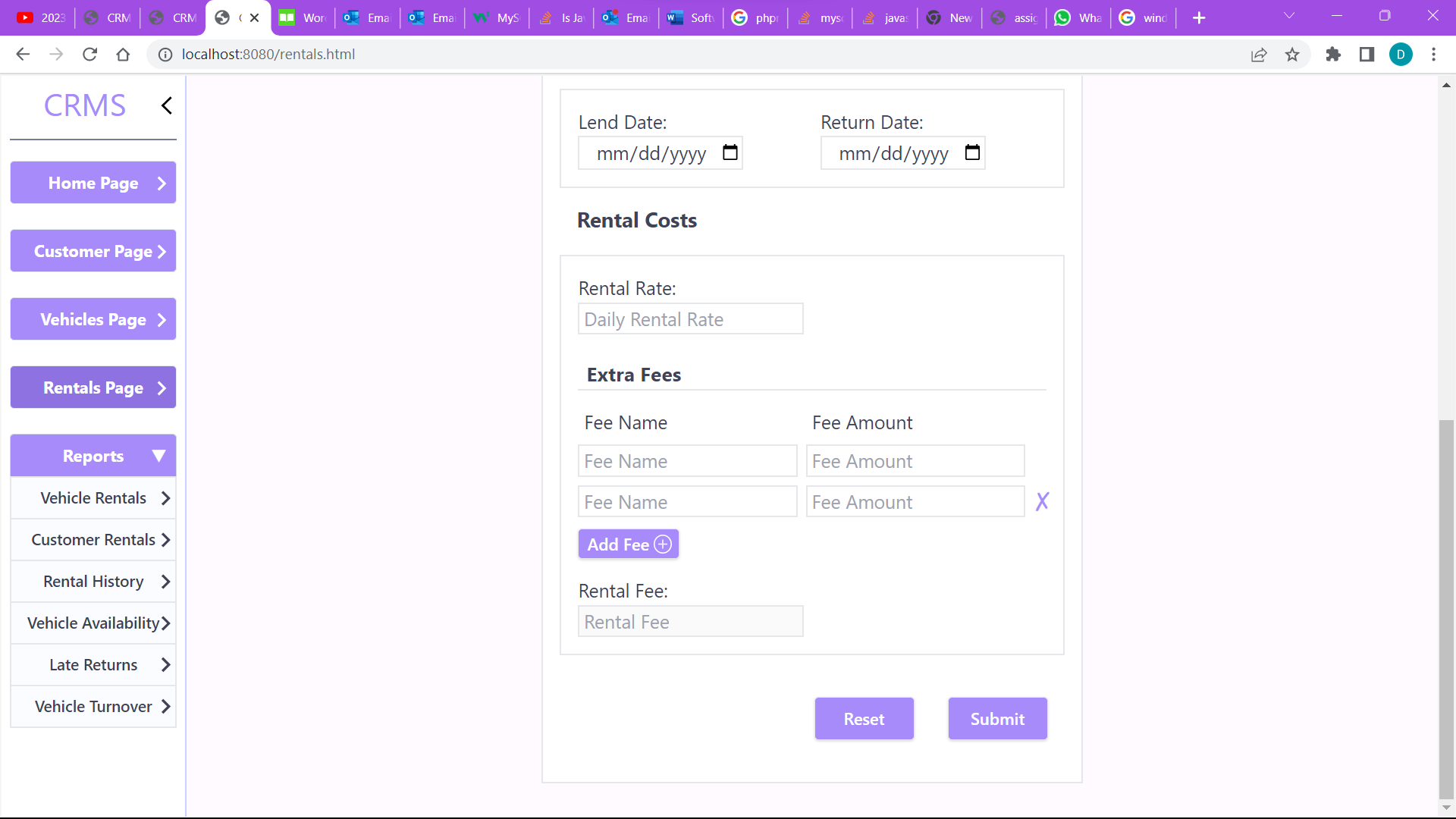


Figure 21 The Rental Form (2)

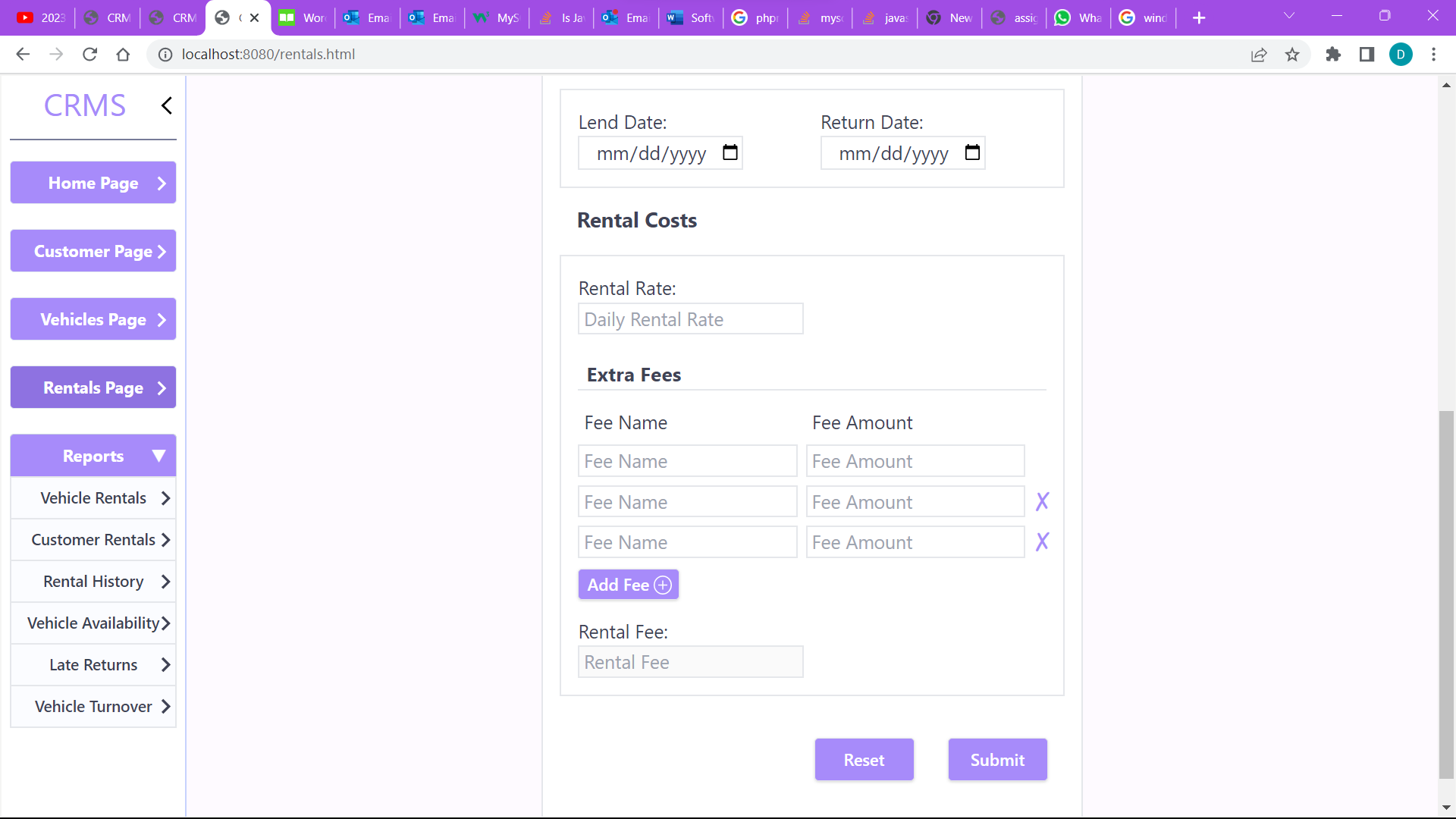


Figure 22

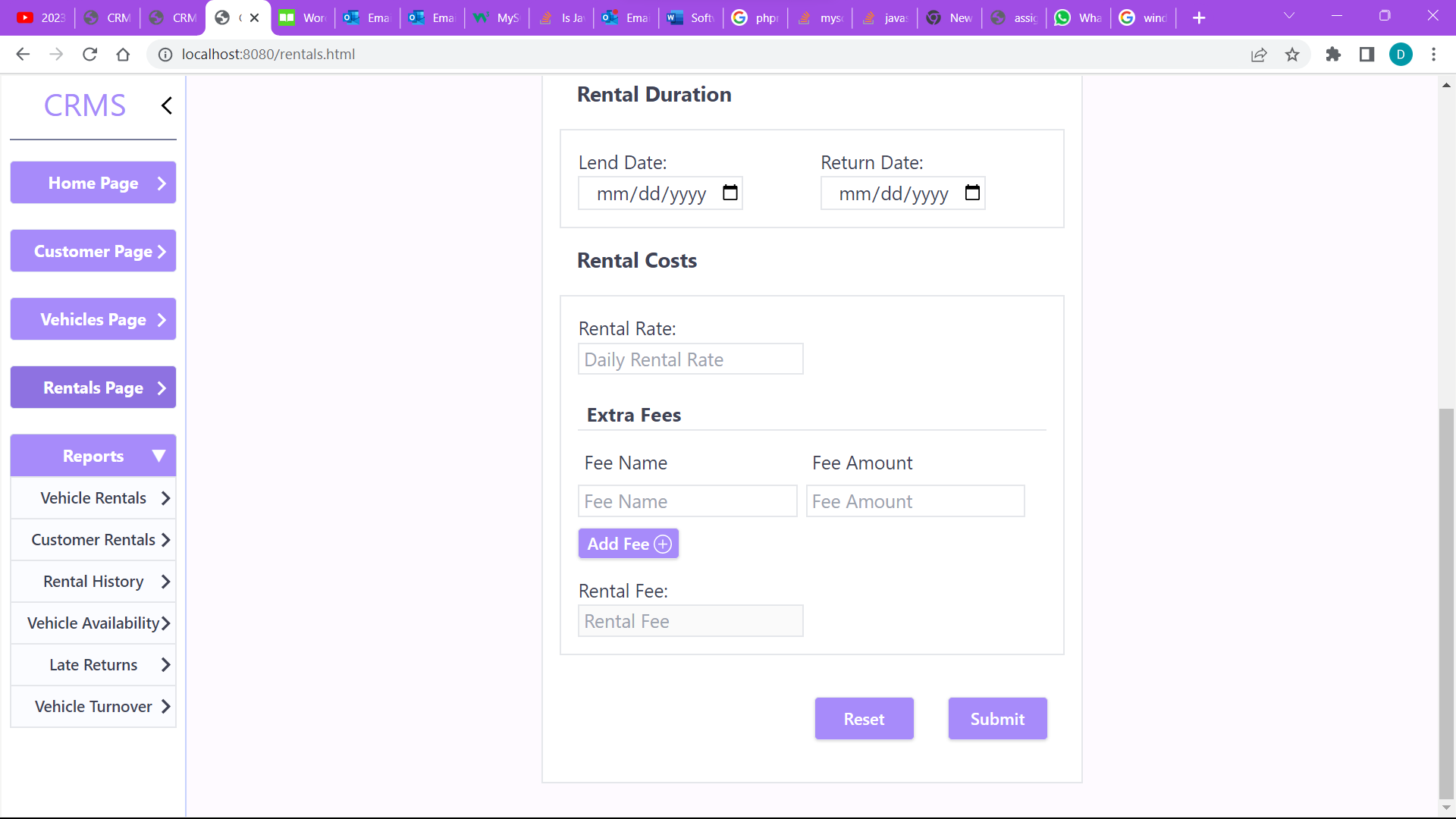


Figure 23

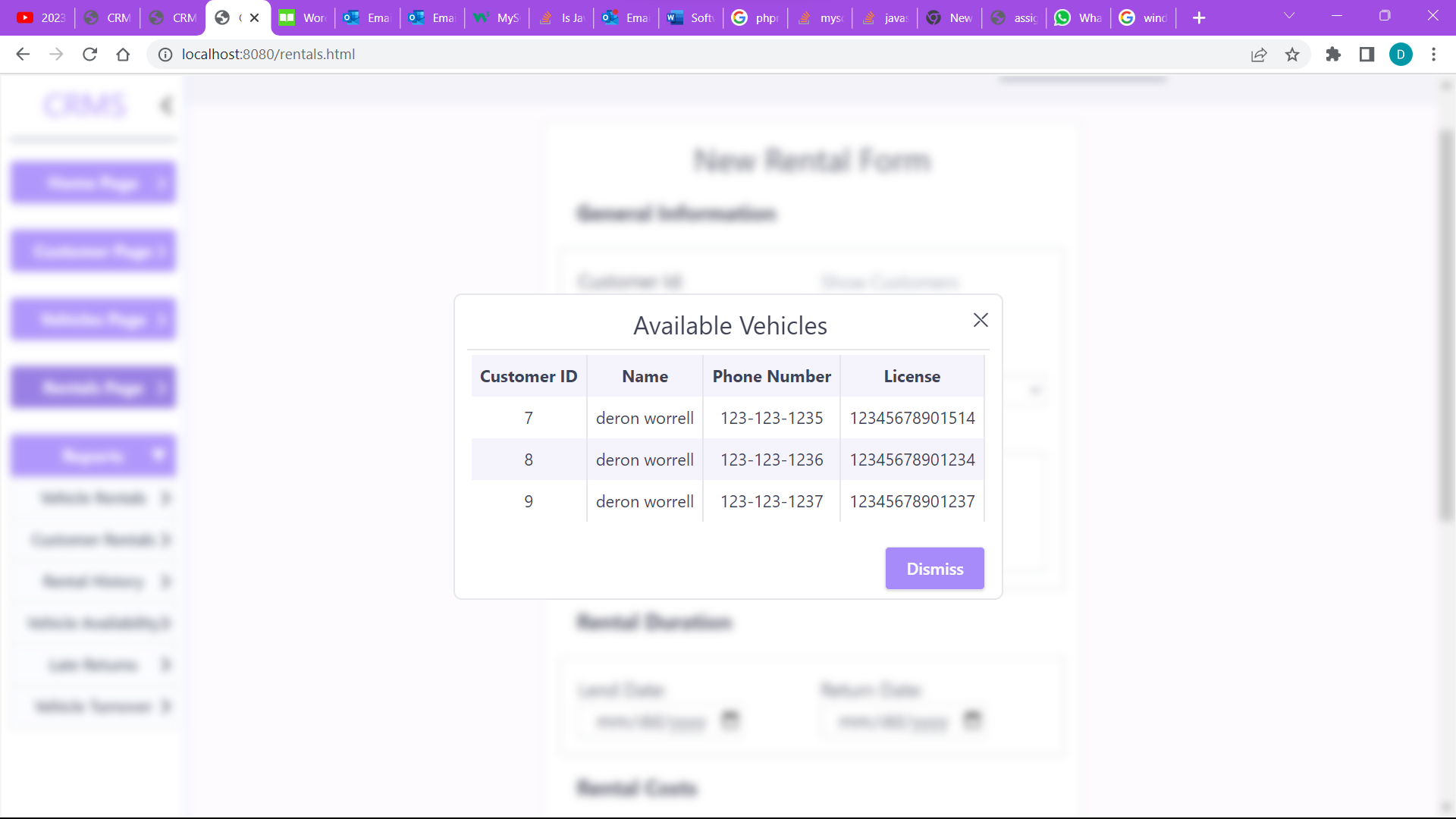


Figure 24

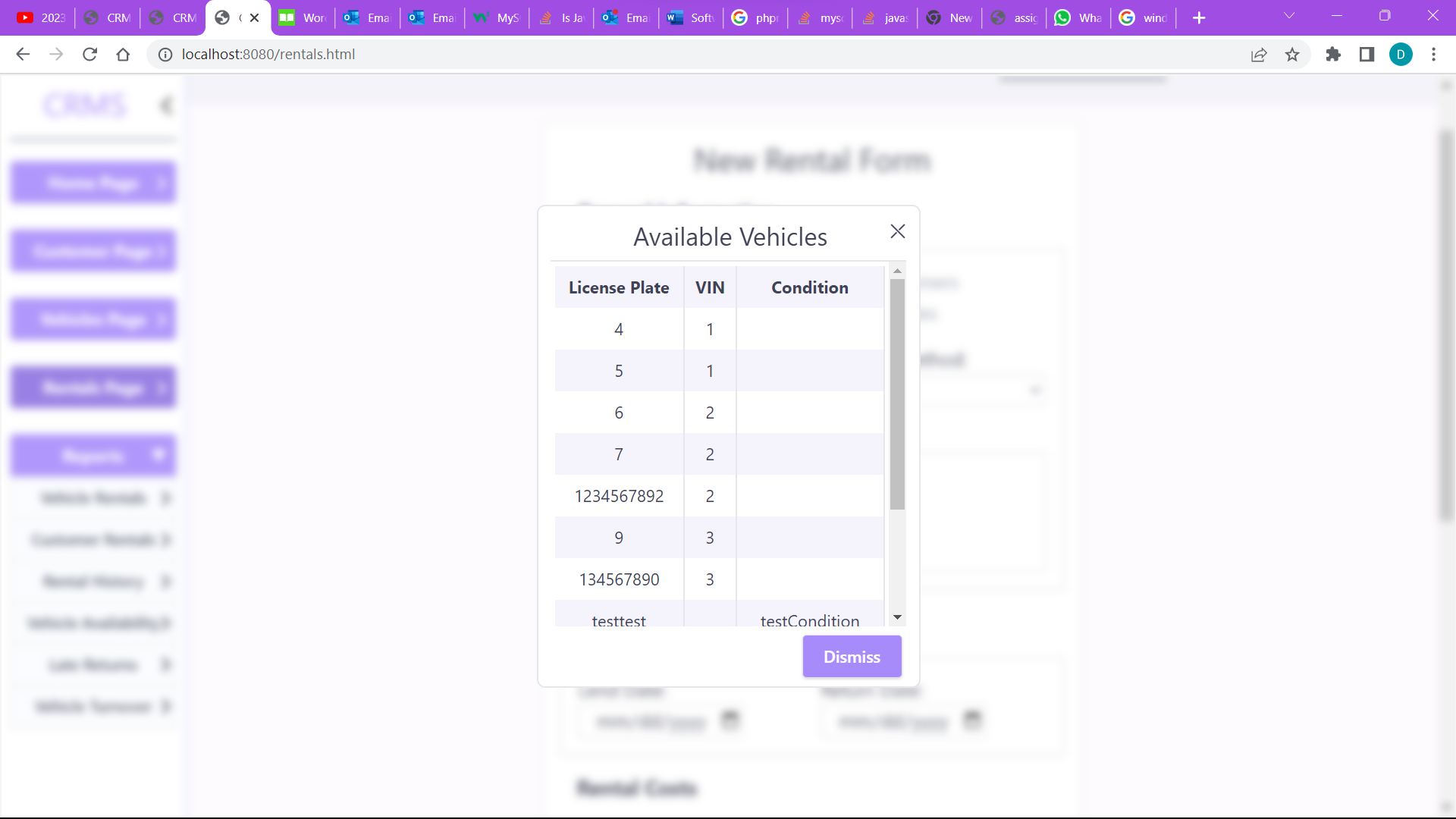


Figure 25

## Rentals Report Page

(Vehicle Rentals & Rental History)

By clicking on the vehicle rentals or the rental history page the user can view detailed information on the rentals for specific vehicles or for all, showing the average daily revneue, the total revenue, total rental period and the number of rentals. By default, it shows the data for all rentals. When a vehicle is selected the associated information is displayed in the box to the top left of the screen in figure 26. The user can search the license plate using the search box to the right of the screen as the user enters a license plate all matching license plates are displayed in a popup as shown in figure 27. From there the user can click and select a license which will be stored and displayed in the Vehicle License Box underneath the search box. The User can then proceed to select a lent and/or return date and then click the view button to see all data for the associated query. Clicking the clear button resets the selected vehicle license and clears the lent & return date values. Clicking View without entering any query data or selecting a license shows data for all rental records.

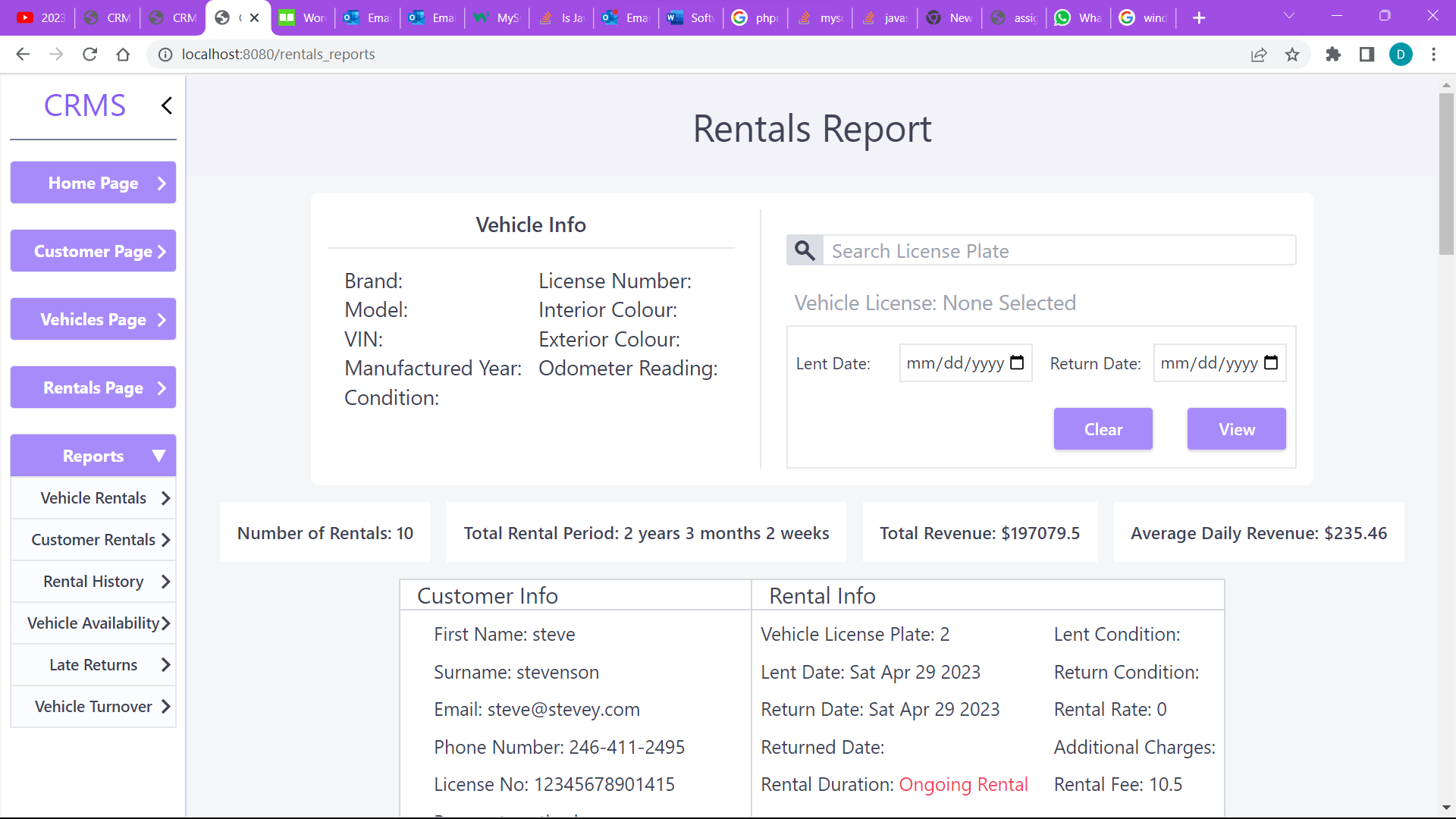


Figure 26

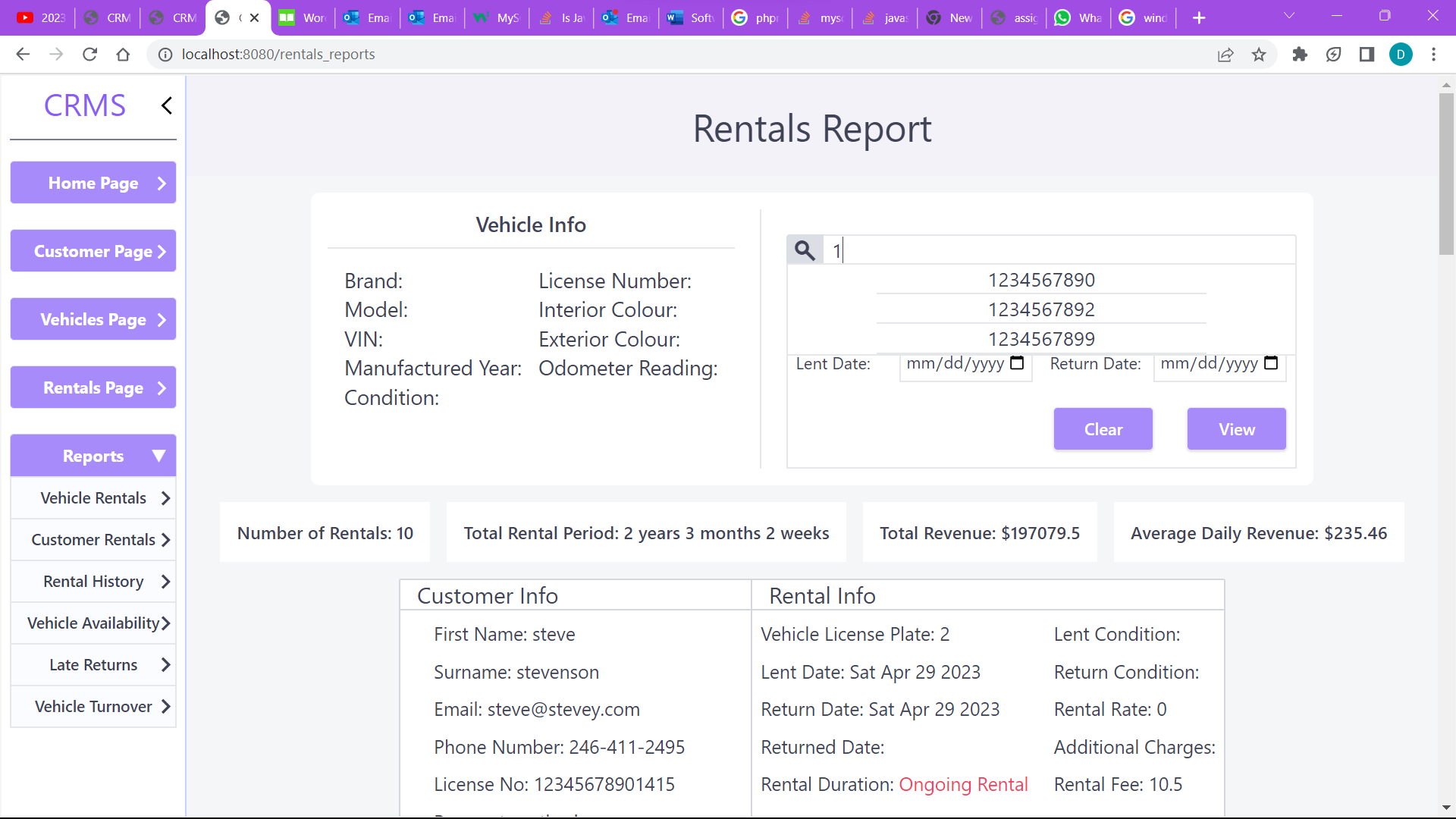


Figure 27

When the user clicks the customer rentals button, they will be taken to the page below in figure 28 where they can view the rentals of a specific customer either by searching the customer's name or making a more specific query by filtering the data as shown in Figure 28. The user can filter based on any combination of the customer’s first name, surname, email or phone number as explained above for the customer page.

## Customer Rentals Page

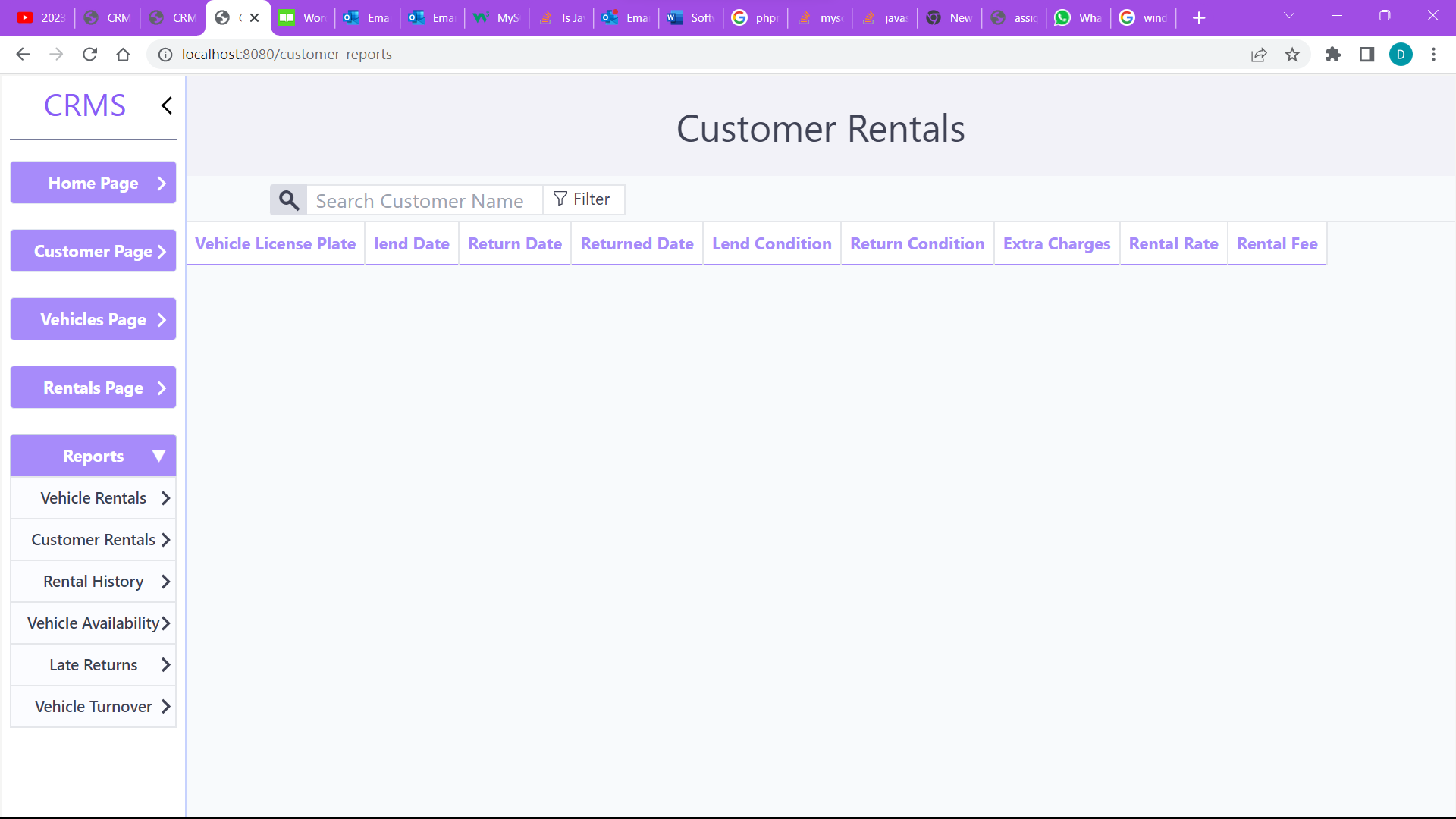


Figure 28

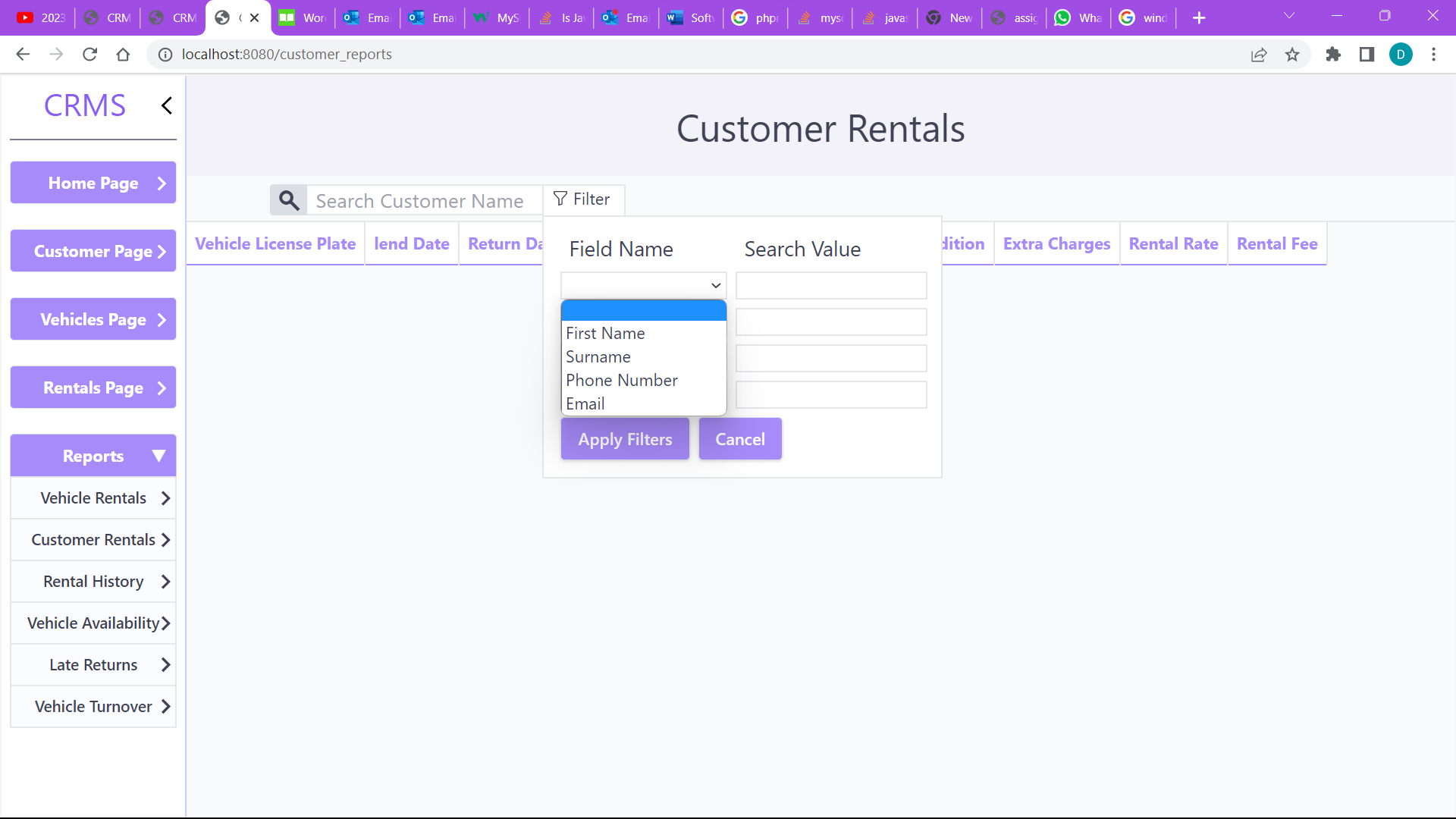
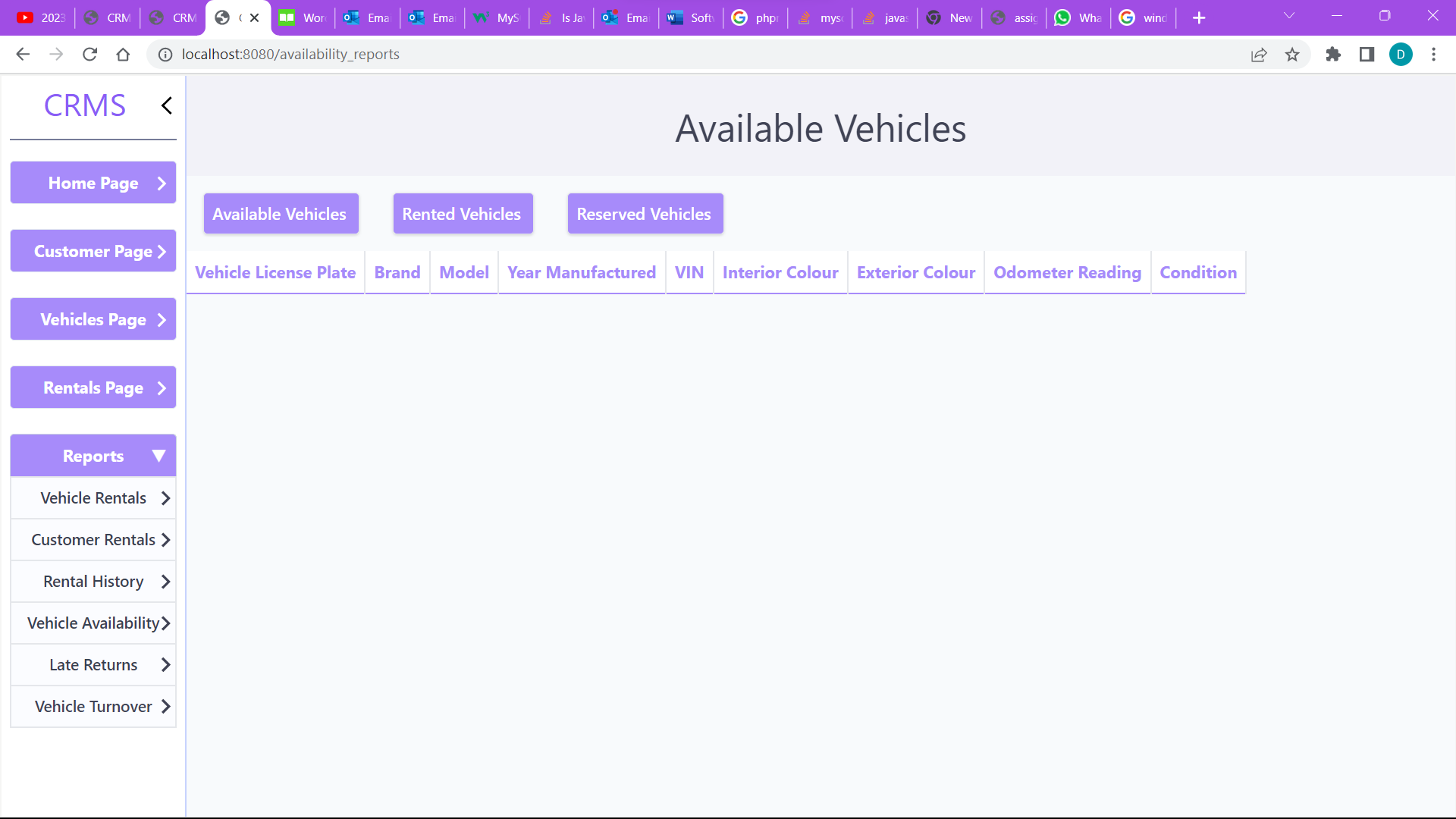


Figure 292

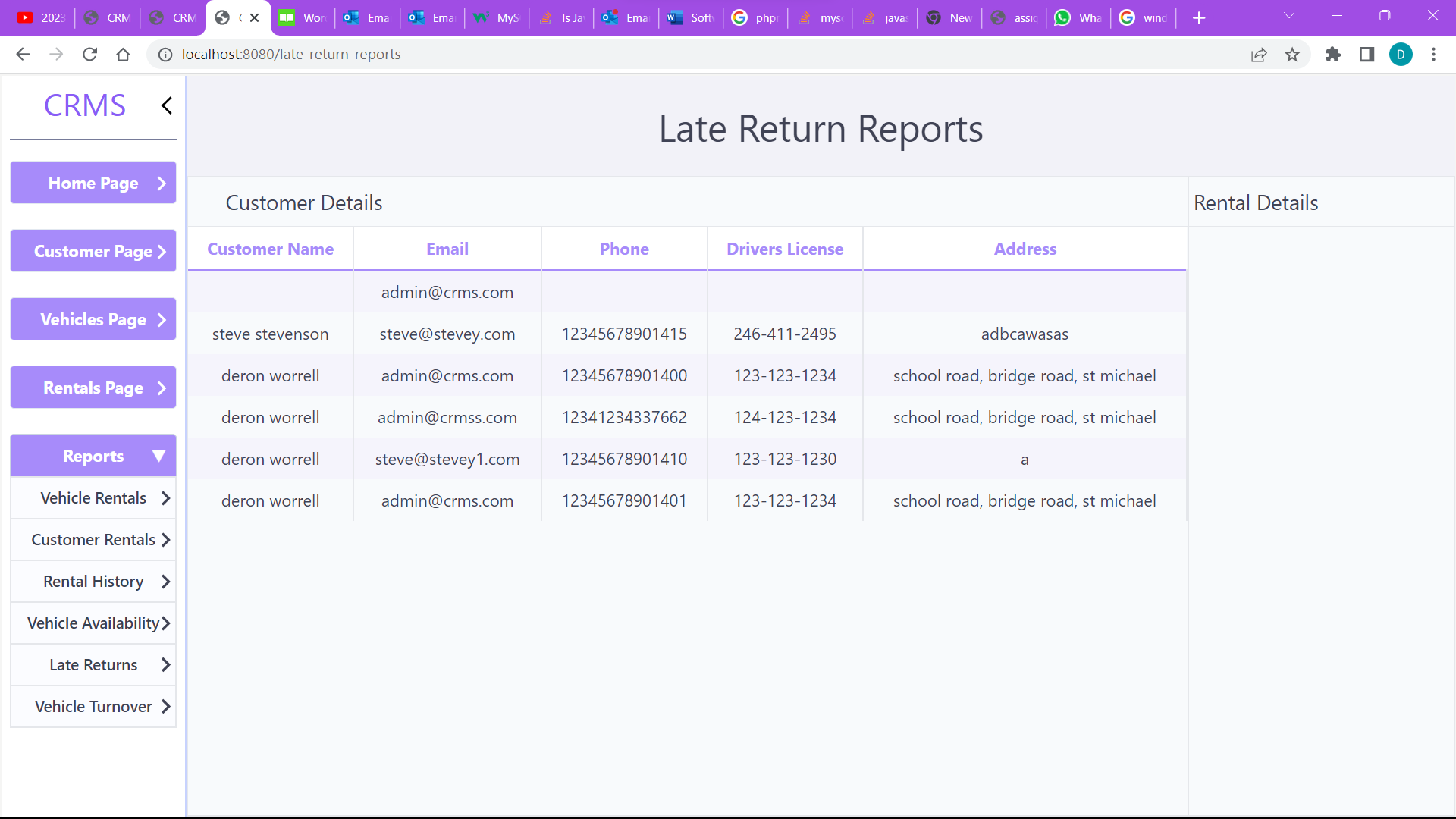
By clicking the vehicle availability button the user would be taken to the page below where they can view the availability of all vehicles in the fleet. By clicking on the available vehicles button information on all vehicles which aren’t rented out would be displayed in the table below. If they click on the rented vehicles button information on all vehicles that are currently rented out will be displayed. Lastly, if the user clicks on the reserved vehicles button information on all vehicles that have future rentals planned and stored in the system will be displayed.

## Available Vehicles Page



By clicking on the late returns button the user will be taken to a page which displays all associated customer and rental information on any rentals that have not been returned although the agreeded upon return date has passed.

## Late Returns Page



By clicking the vehicle turnover button, the page below in figure 30 will be displayed where users can view information on the total rentals, rentals this month, rentals this year, avg monthly rentals and average rental time of a specific vehicle. Users can search for a specific vehicle by using the search box at the top of the page or they can click the previous or next vehicle buttons to go through each vehicle. They can also click the rentals details button and be redirected to the rentals page where they can see detailed information on each rental. In the rentals this month box the percentage represents the increase or decrease in rentals when compared to the previous month in the data shown below there was a 50% decrease in rentals for the current month. The graph to the right shows the rentals for each month of the current year including future rentals.

## Turnover Report Page

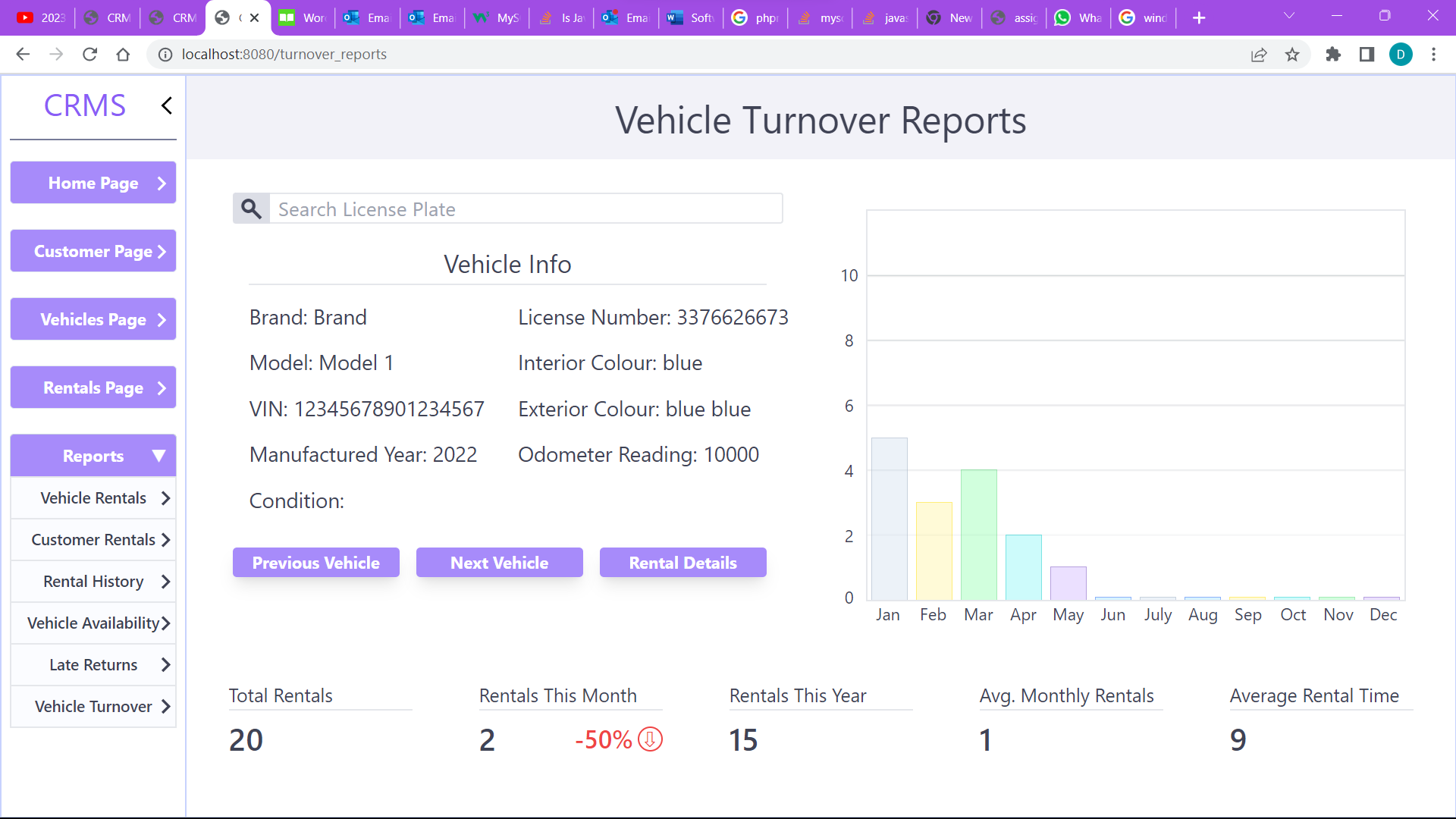


Figure 30

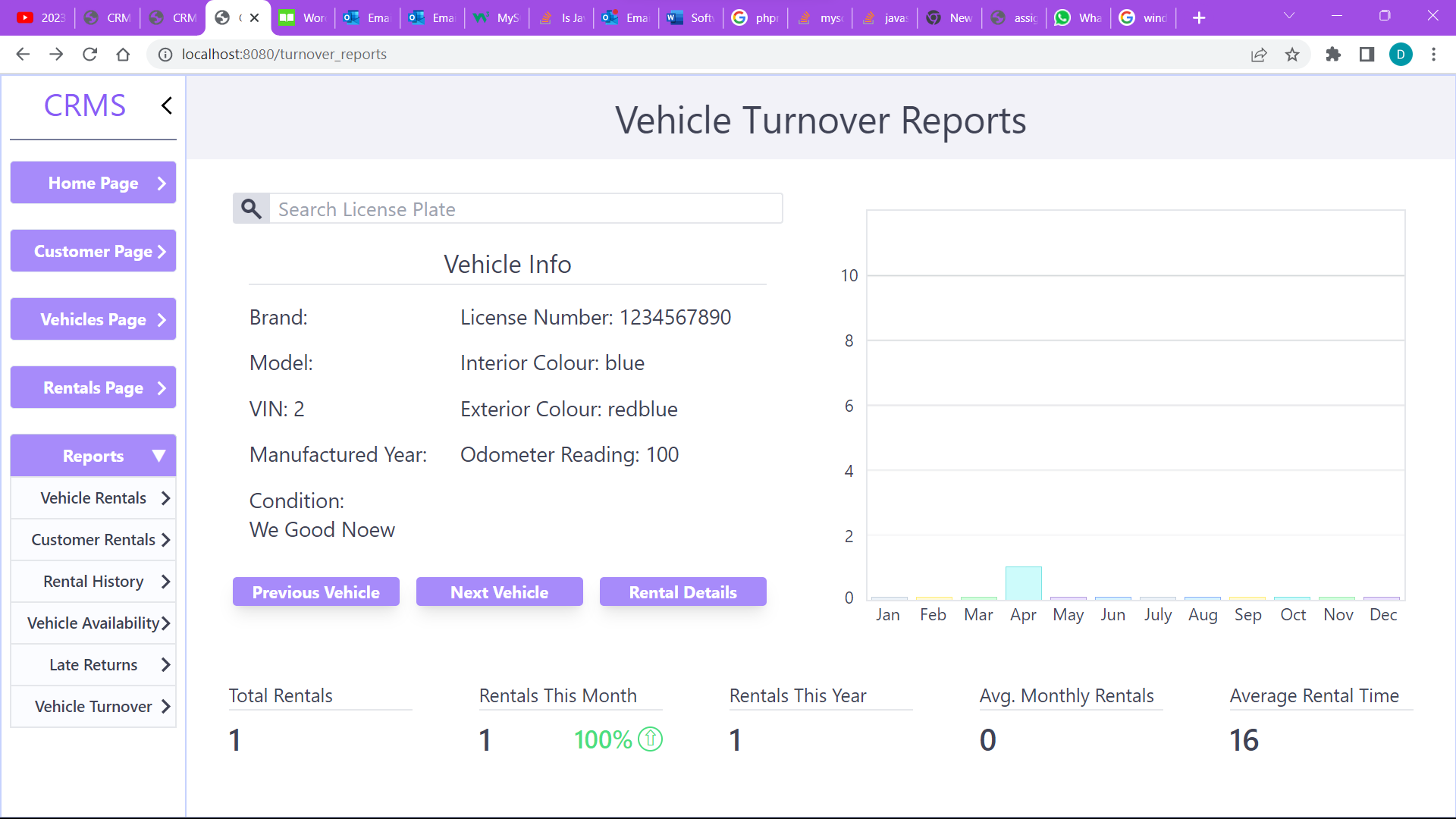


Figure 31

# Future Additions

As the CRMS application evolves over time and new features are added, there may be additional requirements and enhancements that need to be considered in the test plan. The following are potential areas for future additions to the application:

* Updated User Interface – An updated user interface could provide customers with additional information related to the rental of a car including a visual representation of the Car
* Mobile application: A mobile app can provide a convenient way for customers to make reservations, view rental history, and manage their accounts from their mobile devices.
* Loyalty program: Implementing a loyalty program can incentivize repeat business and provide customers with rewards for their loyalty, such as discounts on future rentals, free upgrades, and other perks.
* GPS tracking: GPS tracking can allow customers to locate and navigate to their rental vehicle, as well as help rental companies track the location and condition of their fleet.
* Predictive maintenance: Implementing predictive maintenance technology can help rental companies identify potential vehicle issues before they become major problems, which can improve vehicle reliability and reduce downtime.