

# CAR RENTAL

## Database

Final Project  
(phase 3)  
2022/05/08

Team Members  
Bijan Saud  
Madison Nguyen  
Trieu Nguyen  
Swagat Pandak

## HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence. I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

## Task 1: Execute the following queries on the CarRental2019 database tables:

### Query 1: [5 points]

Add an extra column ‘Returned’ to the RENTAL table. Values will be 0-for non-returned cars, and 1-for

returned. Then update the ‘Returned’ column with '1' for all records that they have a payment date and with '0' for those that they do not have a payment date.

Query:

ALTER TABLE rental ADD Returned INT;

UPDATE rental SET Returned = IF(PaymentDate = 'NULL',0,1);

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0106 seconds.)

```
CREATE TABLE `rental` ( `CustID` int(11) NOT NULL, `VehicleID` char(17) NOT NULL, `StartDate` varchar(15) NOT NULL, `OrderDate` varchar(15) NOT NULL, `RentalType` int(11) NOT NULL, `Qty` int(11) NOT NULL, `ReturnDate` varchar(15) NOT NULL, `TotalAmount` float NOT NULL, `PaymentDate` varchar(15) DEFAULT NULL, `Returned` int(11) DEFAULT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[Edit inline] [Edit] [Create PHP code]

✓ 23 rows inserted. (Query took 0.0060 seconds.)

```
VALUES (229, '19VDE1F3XEE414842', '2019-05-06', '2019-04-12', 1, 4, '2019-06-10', 400, '2019-05-06', 1, (212, '19VDE1F3XEE414842', '2019-06-10', '2019-04-15', 7, 3, '2019-07-01', 1800, '2019-06-10', 1), (221, '19VDE1F3XEE414842', '2019-07-01', '2019-06-12', 7, 1, '2019-07-08', 600, '2019-07-01', 1), (221, '19VDE1F3XEE414842', '2019-07-09', '2019-06-12', 1, 2, '2019-07-11', 200, '2019-07-01', 1), (210, '19VDE1F3XEE414842', '2019-11-01', '2019-10-28', 7, 2, '2019-11-15', 1200, 'NULL', 0), (221, '19VDE1F3XEE414842', '2020-01-01', '2019-12-15', 7, 4, '2020-01-29', 2400, 'NULL', 0), (216, '1N6BF0KMOEN101134', '2019-08-02', '2019-03-15', 7, 4, '2019-08-30', 2740, '2019-08-02', 1), (216, '1N6BF0KMOEN101134', '2019-08-30', '2019-03-15', 1, 2, '2019-09-01', 230, '2019-08-02', 1), (203, 'JM3KE4DY4F0441471', '2019-09-09', '2019-05-22', 1, 4, [...]
```

[Edit]

The screenshot shows the phpMyAdmin interface for the Car\_Rental database. The left sidebar shows the database structure with tables: Car\_Rental, customers, rate, rental, vehicle. The rental table is selected in the main area. The table structure is shown with columns: CustID, VehicleID, StartDate, OrderDate, RentalType, Qty, ReturnDate, TotalAmount, PaymentDate, and Returned. Below the table, 23 rows of data are listed, each with edit, copy, and delete options. The 'Returned' column is filled with values like 1, 0, or NULL. At the bottom, there is a SQL console with the query `>SELECT * FROM `rental``.

**Create a view vRentalInfo that retrieves all information per rental. The view should have the following attributes:**

Query:

```
SELECT OrderDate, StartDate, ReturnDate, (RentalType*Qty) AS TotalDays,  
rental.VehicleID AS VIN, Description AS Vehicle,  
CASE  
WHEN vehicle.Type = 1 THEN 'Compact'  
WHEN vehicle.Type = 2 THEN 'Medium'  
WHEN vehicle.Type = 3 THEN 'Large'  
WHEN vehicle.Type = 4 THEN 'SUV'  
WHEN vehicle.Type = 5 THEN 'Truck'  
WHEN vehicle.Type = 6 THEN 'VAN'  
END AS Type,  
CASE  
WHEN vehicle.Category = 0 THEN 'BASIC'  
WHEN vehicle.Category = 1 THEN 'LUXURY'  
END AS Category,  
rental.CustID AS CustomerID, customers.Name AS CustomerName, TotalAmount AS  
OrderAmount,  
CASE  
WHEN Returned = 0 THEN TotalAmount  
ELSE 0  
END AS RentalBalance  
FROM customers, rental, vehicle  
WHERE rental.CustID = customers.CustID AND rental.VehicleID = vehicle.VehicleID  
ORDER BY StartDate ASC;
```

localhost / localhost / Car\_Rental

phpMyAdmin

Server: localhost - Database: Car\_Rental - View: rentalinfo

Browse Structure SQL Search Insert Export Privileges Operations Tracking

Show all Number of rows: 25 Filter rows: Search this table

+ Options OrderDate StartDate ReturnDate TotalDays VIN Vehicle Type Category CustomerID CustomerName OrderAmount RentalBalance

✓ Edit Copy Delete 2019-04-15 2019-05-01 2019-05-08 7 JHFF2C26F135BX45 "Lexus IS 250C" Compact LUXURY 210 G. Clarkson 600 0

✓ Edit Copy Delete 2019-04-12 2019-05-06 2019-06-10 4 WAUTFAFH0E010613 "Audi A5" Compact LUXURY 229 D. Kirkpatrick 400 0

✓ Edit Copy Delete 2019-04-12 2019-05-06 2019-06-10 4 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 229 D. Kirkpatrick 400 0

✓ Edit Copy Delete 2019-04-15 2019-06-10 2019-07-01 21 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 212 H. Gallegos 1800 0

✓ Edit Copy Delete 2019-06-12 2019-07-01 2019-07-08 7 WAUTFAFH0E010613 "Audi A5" Compact LUXURY 221 J. Brown 600 0

✓ Edit Copy Delete 2019-06-12 2019-07-01 2019-07-08 7 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 221 J. Brown 600 0

✓ Edit Copy Delete 2019-06-12 2019-07-09 2019-07-11 2 WAUTFAFH0E010613 "Audi A5" Compact LUXURY 221 J. Brown 200 0

✓ Edit Copy Delete 2019-06-12 2019-07-09 2019-07-11 2 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 221 J. Brown 200 0

✓ Edit Copy Delete 2019-03-15 2019-08-02 2019-08-30 28 1N6BF0KMC0EN101134 "Nissan NV" VAN BASIC 216 A. Hess 2740 0

✓ Edit Copy Delete 2019-03-15 2019-08-30 2019-09-01 2 1N6BF0KMC0EN101134 "Nissan NV" VAN BASIC 216 A. Hess 230 0

✓ Edit Copy Delete 2019-05-22 2019-09-09 2019-09-13 4 JM3KE4D1Y4F041471 "Mazda CX5" SUV BASIC 203 A. Hernandez 460 0

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 WAUTFAFH0E010613 "Audi A5" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 WBA3B9C59EP458859 "BMW 3 Series" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 JTHFF2C26F135BX45 "Lexus IS 250C" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 WDCGG0EB0EG188709 "Mercedes\_Benz GLK" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2019-11-01 2019-11-15 14 WBA3A9G51EN73366 "BMW 3 Series" Compact LUXURY 210 G. Clarkson 1200 1200

✓ Edit Copy Delete 2019-10-28 2020-01-01 2020-01-29 28 WBA3B9C59EP458859 "BMW 3 Series" Compact LUXURY 221 J. Brown 2400 2400

✓ Edit Copy Delete 2019-12-15 2020-01-01 2020-01-29 28 WBA3A9G51EN73366 "BMW 3 Series" Compact LUXURY 221 J. Brown 2400 2400

✓ Edit Copy Delete 2019-12-15 2020-01-01 2020-01-29 28 WAUTFAFH0E010613 "Audi A5" Compact LUXURY 221 J. Brown 2400 2400

✓ Edit Copy Delete 2019-12-15 2020-01-01 2020-01-29 28 JTHFF2C26F135BX45 "Lexus IS 250C" Compact LUXURY 221 J. Brown 2400 2400

✓ Edit Copy Delete 2019-12-15 2020-01-01 2020-01-29 28 19VDE1F3XEE414842 "Acura ILX" Compact LUXURY 221 J. Brown 2400 2400

✓ Edit Copy Delete 2019-12-15 2020-01-01 2020-01-29 28 WDCGG0EB0EG188709 "Mercedes\_Benz GLK" Compact LUXURY 221 J. Brown 2400 2400

Show all Number of rows: 25 Filter rows: Search this table

Bookmarks Options History Cies

Console

Press Ctrl+Enter to execute query

```
>SELECT * FROM 'rental'
>SELECT * FROM 'rentalinfo'
>SELECT * FROM 'vrentalinfo'
```

**Task 2: Create a GUI for the CarRental2019 database:**

**Car Rental Application: [20 points]**

**Create a simple and friendly GUI interface that would be able to perform the following tasks. You may**

**use JAVA programming using JDBC, or C/C++/C# programming with ODBC/Oracle or Python**

**programming or PHP/MySQL or other programming languages to develop a GUI interface. You can**

**always create a simple interface like a command prompt! For each query, you need to provide some info**

**to the user about the query purpose. The user will have to type or select the query's input parameters and**

**post the question to your program. The program needs to return all result's rows.**

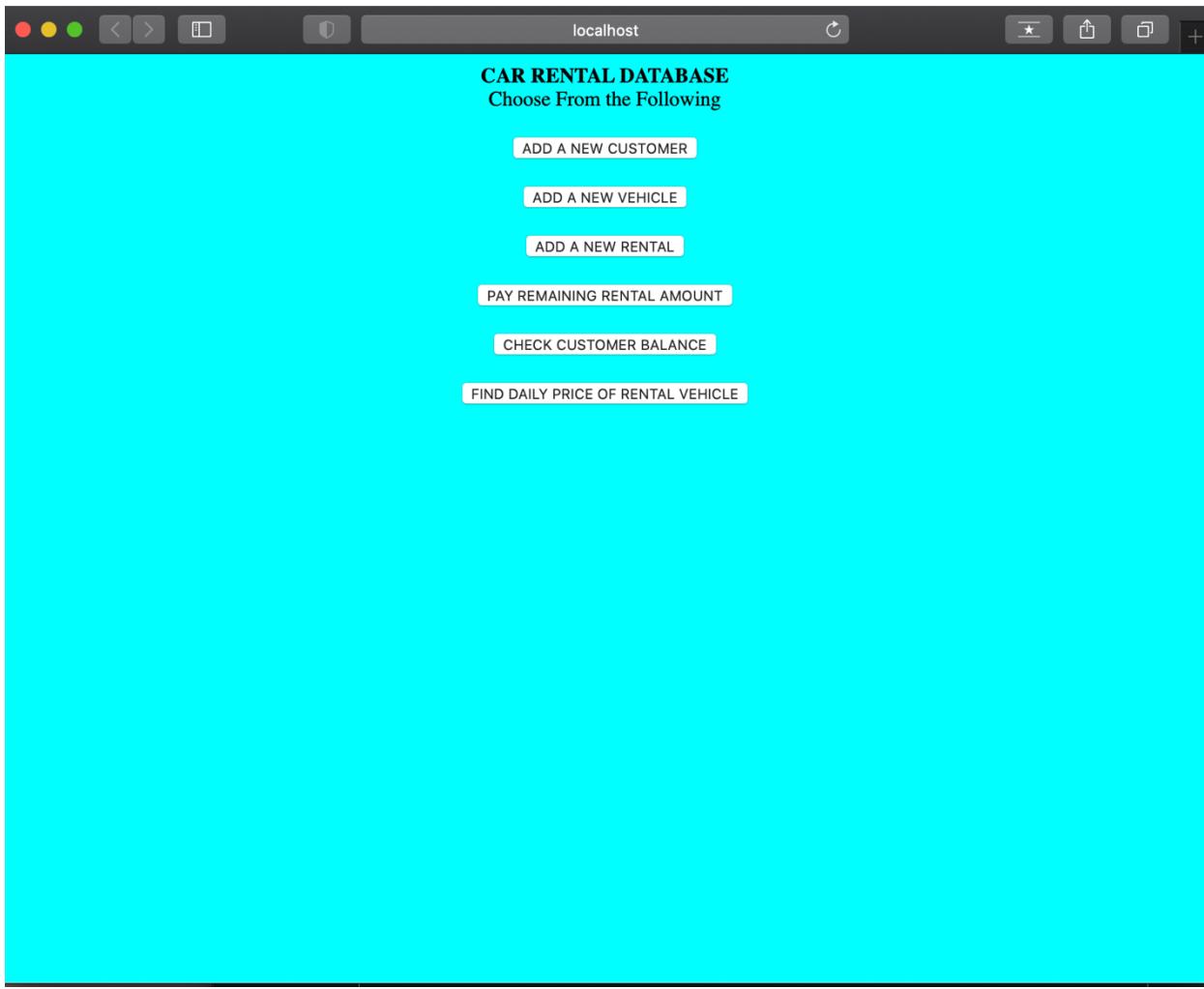
**You also need to print**

**the count of rows returned.**

The file can be run from this link as a local host.

**Link:**

**<http://localhost/Code/main.php>**



**Requirements:**

**1. The first requirement is to add information about a new customer. Do not provide the customer ID in your query. Submit your editable SQL query that your code executes.**  
**[3 points]**

**Query:**

INSERT INTO customers(Name, Phone) VALUES ('\$name', '\$phone\_number')

localhost

Enter customer's NAME and PHONE

Name: Initial AND LastName

B. Saud

Phone Number

(214) 412-5644

Submit

localhost

Success: New Customer Added

The screenshot shows the phpMyAdmin interface for the 'Car\_Rental' database. The left sidebar shows the database structure with the 'customers' table selected under the 'Tables' section. The main area displays the 'customers' table with the following data:

	CustID	Name	Phone
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	226	R. Armstrong	(325) 783-4081
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	227	J. Greenaway	(212) 262-8829
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	228	K. Kaiser Acosta	(228) 576-1557
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	229	D. Kirkpatrick	(773) 696-8009
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	230	A. Odonnell	(439) 536-8929
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	231	K. Kay	(368) 336-5403
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	233	B. Saud	(214) 412-5644

Below the table, there are buttons for 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'. The bottom section shows the query history:

```
Press Ctrl+Enter to execute query
>SELECT * FROM `customers`
>SELECT * FROM `customers`
```

**2. The second requirement is to add all the information about a new vehicle. Submit your editable SQL query that your code executes. [3 points]**

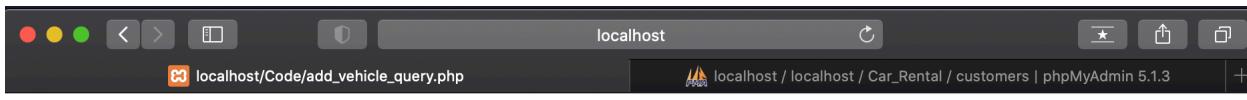
```
INSERT INTO vehicle (VehicleID, Description, Year, Type, Category)
VALUES ('$VIN', CONCAT('\"','$Description','\"'), '$Year', '$Type' , '$Category')
```

The screenshot shows a web browser window titled "CAR\_RENTAL" with the URL "localhost / localhost / Car\_Rental / customers | phpMyAdmin 5.1.3". The main content is a form titled "Enter the Information for Vehicle".

The form fields are:

- Vehicle Identification Number of Length 17**: An input field containing "12345678901234567".
- Vehicle Description**: An input field containing "Test Vehicle".
- Vehicle Year**: An input field containing "2022".
- Type**: A note stating "Type : 'Compact' = 1, 'Medium' = 2, 'Large' = 3, 'SUV' = 4, 'Truck' = 5, 'VAN' = 7". Below it is an input field containing "2".
- Category**: A note stating "Category : 'BASIC' = 0, 'LUXURY' = 1". Below it is an input field containing "1".

A "Submit" button is located at the bottom right of the form.



localhost

localhost/Code/add\_vehicle\_query.php

localhost / localhost / Car\_Rental / vehicle | phpMyAdmin 5.1.3

**phpMyAdmin**

Recent Favorites

Car\_Rental

- Tables
  - New
  - customers
  - rate
  - rental
  - vehicle
- Views
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

VehicleID Description Year Type Category

VehicleID	Description	Year	Type	Category
12345678901234567	"Test Vehicle "	2022	2	1
19VDE1F3XEE414842	"Acura ILX"	2014	1	1
1FDEE3FL6EDA29122	"Ford E 350"	2014	6	0
1FDRF3B61FEA87469	"Ford Super Duty Pickup"	2015	5	0
1FTNF1CF2KE54305	"Ford F Series Pickup"	2014	5	0
1G1JD5SB3E4240835	"Chevrolet Optra"	2014	1	0
1GB3KZCG1EF117132	"Chevrolet Silverado"	2014	5	0
1HGCR2E3XEA305302	"Honda Accord"	2014	2	0
1N4AB7AP2EN855028	"Nissan Sentra"	2014	1	0
1N6BA0EJ9EN516565	"Nissan Titan"	2014	5	0
1N8BF0KM0EN101134	"Nissan NV"	2014	6	0
1VWCH7A3XEC037969	"Volkswagen Passat"	2014	2	1
2HGFB2F94FH501940	"Honda Civic"	2015	1	0
2T3DFREV0FW317743	"Toyota RAV4"	2015	4	0
3MZBM1L74EM109736	"Mazda 3"	2014	1	0
3N1CE2CP0FL409472	"Nissan Versa Note"	2015	1	0
3N1CN7APXEK444458	"Nissan Versa"	2014	1	0
3VW2A7AU1FM012211	"Volkswagen Golf"	2015	1	0
4S4BRCFC1E3203823	"Subaru Outback"	2014	4	0
4S4BSBF3F3261064	"Subaru Outback"	2015	4	0

Console

Press Ctrl+Enter to execute query

>SELECT \* FROM `customers`  
>SELECT \* FROM `customers`  
>SELECT \* FROM `vehicle`

**3. The third requirement is to add all the information about a new rental reservation (this must find a free vehicle of the appropriate type and category for a specific rental period). We assume that the customer has the right either to pay at the order or return date. Submit your editable SQL queries (select available vehicles & insert rental) that your code executes. [8 points]**

```

INSERT INTO rental (CustID, StartDate, RentalType, Qty,
                   ReturnDate, PaymentDate, VehicleID, OrderDate, TotalAmount,
                   Returned)
VALUES ('$CustID', '$StartDate', '$RentalType', '$Qty',
        (
            SELECT DATE_ADD('$StartDate', INTERVAL
(''$RentalType'*$Qty') DAY) AS '$ReturnDate'
        ), '$PaymentDate',
        (SELECT vehicle.VehicleID
         FROM vehicle
         WHERE Type = $Type AND Category = $Category AND
VehicleID IN
        (SELECT vehicle.VehicleID
         FROM vehicle
         LEFT JOIN rental AS RENT ON vehicle.VehicleID =
RENT.VehicleID
         WHERE RENT.VehicleID is NULL
        )
        UNION
        SELECT DISTINCT RENT2.VehicleID
         FROM rental AS RENT2, vehicle
         WHERE RENT2.VehicleID = vehicle.VehicleID AND Type =
$Type AND Category = $Category AND
RENT2.VehicleID NOT IN
        (
            SELECT RENT3.VehicleID
             FROM rental AS RENT3
             WHERE (StartDate >= '$StartDate' AND StartDate <=
'$ReturnDate') OR
'$ReturnDate')
        )
        GROUP BY vehicle.vehicleID
        LIMIT 1
    ), CURDATE(),
    (
        SELECT
        CASE
        WHEN $RentalType = 1 THEN (Daily*$Qty)
        ELSE (Weekly*$Qty)
        END AS TotalAmount
        FROM rate
    )
)

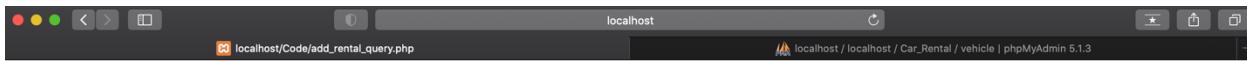
```

```
        WHERE Type = $Type AND Category = $Category
    ) ,
    (
        SELECT
        CASE
        WHEN '$PaymentDate' = 'NULL' THEN 0
        ELSE 1
        END AS Ret
    )
)
```

The screenshot shows a web application interface for entering rental information. The title bar indicates the site is 'CAR\_RENTAL' running on 'localhost'. The main content area has a teal background and displays the following fields:

- Customer ID:** A dropdown menu showing '233'.
- Vehicle Type:** A dropdown menu showing '2'.
- Category:** A dropdown menu showing '1'.
- Rental Start Date:** A dropdown menu showing '2022-05-10'.
- Rental Type:** A dropdown menu showing '7'.
- Rental Type Quantity:** A dropdown menu showing '1'.
- Payment Date:** A dropdown menu showing 'NULL'.

Below the input fields is a 'Submit' button.



A screenshot of the phpMyAdmin interface for the 'Car\_Rental' database. The 'rental' table is selected, showing 23 rows of data. The table has columns: CustID, VehicleID, StartDate, OrderDate, RentalType, Qty, ReturnDate, TotalAmount, PaymentDate, and Returned. A query history at the bottom shows three previous SQL queries related to 'customers' and 'vehicle' tables.

**4. The fourth requirement is to handle the return of a rented car. This transaction should print the total customer payment due for that rental, enter it in the database and update the returned attribute accordingly. You need to be able to retrieve a rental by the return date, customer name (the table needs the id), and vehicle info. Submit your editable SQL queries (retrieve & update rental) that your code executes. [6 points]**

```
SELECT customers.Name,
       CASE
         WHEN PaymentDate = 'NULL' THEN TotalAmount
         ELSE 0
       END AS Balance_Due
  FROM customers, rental
 WHERE rental.ReturnDate = '$Return_Date' AND customers.Name = '$Name'
   AND customers.CustID = rental.CustID AND rental.VehicleID = '$VIN';
```

```
UPDATE rental
      SET rental.PaymentDate = CURDATE(), Returned = 1
    WHERE rental.ReturnDate = '$Return_Date' AND rental.VehicleID = '$VIN'
      AND rental.CustID IN
        (SELECT customers.CustID
         FROM customers
        WHERE customers.Name = '$Name') " ;
```

localhost CAR\_RENTAL

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

**ENTER RENTAL INFO:**

Return Date  
2022-06-17

Name: Initial AND LastName  
B. Saud

Vehicle Identification Number. Length = 17  
12345678901234567

localhost Code/pay\_rental\_query.php

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

Customer Name	Total Paid
B. Saud	\$660

The information of Customer is Updated.

localhost

localhost / Code/pay\_rental\_query.php

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

**phpMyAdmin**

Recent Favorites

New Car\_Rental Tables New customers rate rental vehicle Views information\_schema mysql performance\_schema phpmyadmin test

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking

Showing rows 0 - 23 (24 total, Query took 0.0005 seconds.)

SELECT \* FROM `rental`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	CustID	VehicleID	StartDate	OrderDate	RentalType	Qty	ReturnDate	TotalAmount	PaymentDate
<input type="checkbox"/>	233	12345678901234567	2022-05-10	2022-05-08		7	1 2022-05-17	660	2022-05-08
<input type="checkbox"/>	229	19VDE1F3XEE414842	2019-05-06	2019-04-12		1	4 2019-06-10	400	2019-05-06
<input type="checkbox"/>	212	19VDE1F3XEE414842	2019-06-10	2019-04-15		7	3 2019-07-01	1800	2019-06-10
<input type="checkbox"/>	221	19VDE1F3XEE414842	2019-07-01	2019-06-12		7	1 2019-07-08	600	2019-07-01
<input type="checkbox"/>	221	19VDE1F3XEE414842	2019-07-09	2019-06-12		1	2 2019-07-11	200	2019-07-01
<input type="checkbox"/>	210	19VDE1F3XEE414842	2019-11-01	2019-10-28		7	2 2019-11-15	1200	NULL
<input type="checkbox"/>	221	19VDE1F3XEE414842	2020-01-01	2019-12-15		7	4 2020-01-29	2400	NULL
<input type="checkbox"/>	216	1N6BF0KM0EN101134	2019-08-02	2019-03-15		7	4 2019-08-30	2740	2019-08-02
<input type="checkbox"/>	216	1N6BF0KM0EN101134	2019-08-30	2019-03-15		1	2 2019-09-01	230	2019-08-02
<input type="checkbox"/>	203	JM3KE4DY4F0441471	2019-09-09	2019-05-22		1	4 2019-09-13	460	2019-09-09
<input type="checkbox"/>	210	JTHFF2C26F135BX45	2019-05-01	2019-04-15		7	1 2019-05-08	600	2019-05-08
<input type="checkbox"/>	210	JTHFF2C26F135BX45	2019-11-01	2019-10-28		7	2 2019-11-15	1200	NULL
<input type="checkbox"/>	221	JTHFF2C26F135BX45	2020-01-01	2019-12-15		7	4 2020-01-29	2400	NULL
<input type="checkbox"/>	229	WAUTFAFH0E0010613	2019-05-06	2019-04-12		1	4 2019-06-10	400	2019-05-06
<input type="checkbox"/>	221	WAUTFAFH0E0010613	2019-07-01	2019-06-12		7	1 2019-07-08	600	2019-07-01
<input type="checkbox"/>	221	WAUTFAFH0F0010613	2019-07-09	2019-06-12		1	2 2019-07-11	200	2019-07-01

Bookmarks Options History Clear

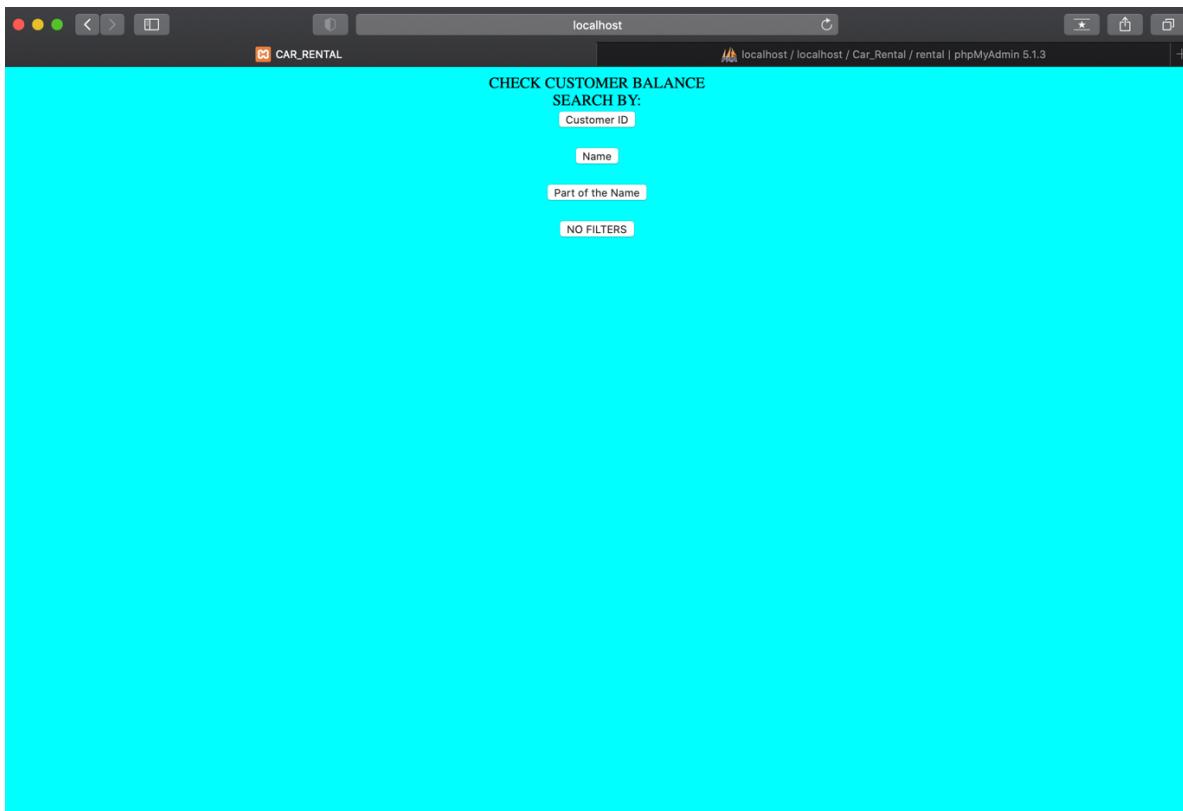
Console

Press Ctrl+Enter to execute query

```
>SELECT * FROM `customers`
>SELECT * FROM `customers`
>SELECT * FROM `vehicle`
```

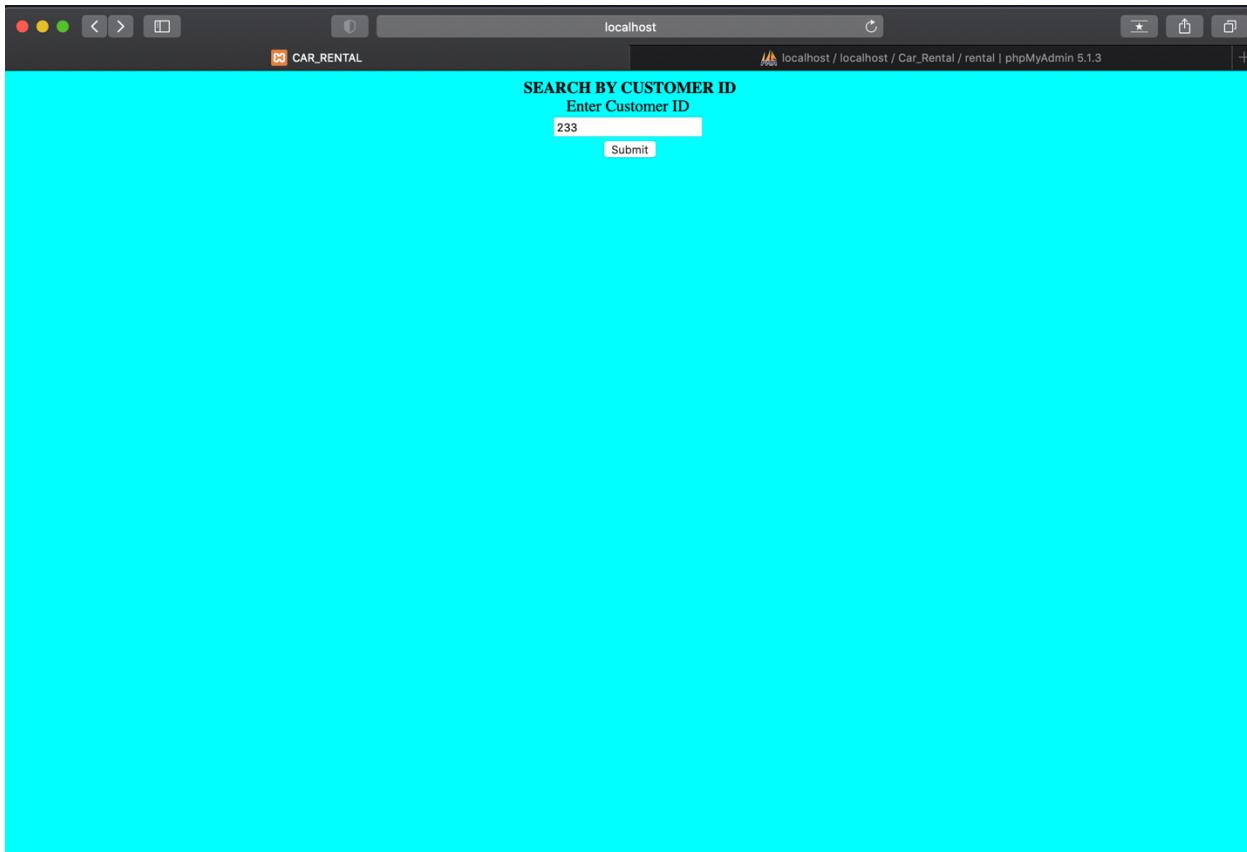
**5. The fifth requirement is to return the view's results by applying the following criteria:**

**a. List for every customer the ID, name, and if there is any remaining balance. The user has the right to search either by a customer's ID, name, part of the name, or to run the query with no filters/criteria. The amount needs to be in US dollars. For customers with zero (0) or NULL balance, you need to return zero dollars (\$0.00). Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the balance amount. Make sure that you return all records. Submit your editable SQL query that your code executes. [10 points]**

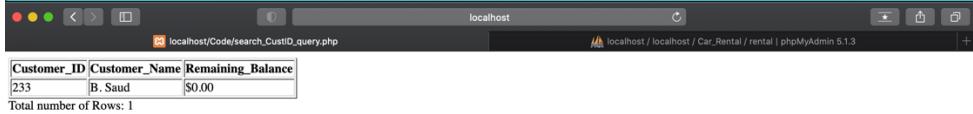


## Search Customer BY Customer ID

```
SELECT CustomerID, CustomerName,  
       SUM(RentalBalance) AS RemainingBalance  
    FROM vrentalinfo  
   WHERE CustomerID = :CustomerID;
```



A screenshot of a web browser window titled "CAR\_RENTAL". The address bar shows "localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3". The main content area displays a search form titled "SEARCH BY CUSTOMER ID". It has a label "Enter Customer ID" and a text input field containing "233". Below the input field is a "Submit" button.



A screenshot of a web browser window titled "localhost". The address bar shows "localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3". The main content area displays a table with the results of the search query. The table has three columns: "Customer\_ID", "Customer\_Name", and "Remaining\_Balance". There is one row with data: Customer\_ID 233, Customer\_Name B. Saud, and Remaining\_Balance \$0.00. Below the table, the text "Total number of Rows: 1" is displayed.

Customer_ID	Customer_Name	Remaining_Balance
233	B. Saud	\$0.00

Total number of Rows: 1

## Search a Payment By Full Name of Customer

```
SELECT CustomerID, CustomerName,  
       SUM(RentalBalance) AS RemainingBalance  
  FROM vrentalinfo  
 WHERE CustomerName = :Name  
 GROUP BY CustomerID
```

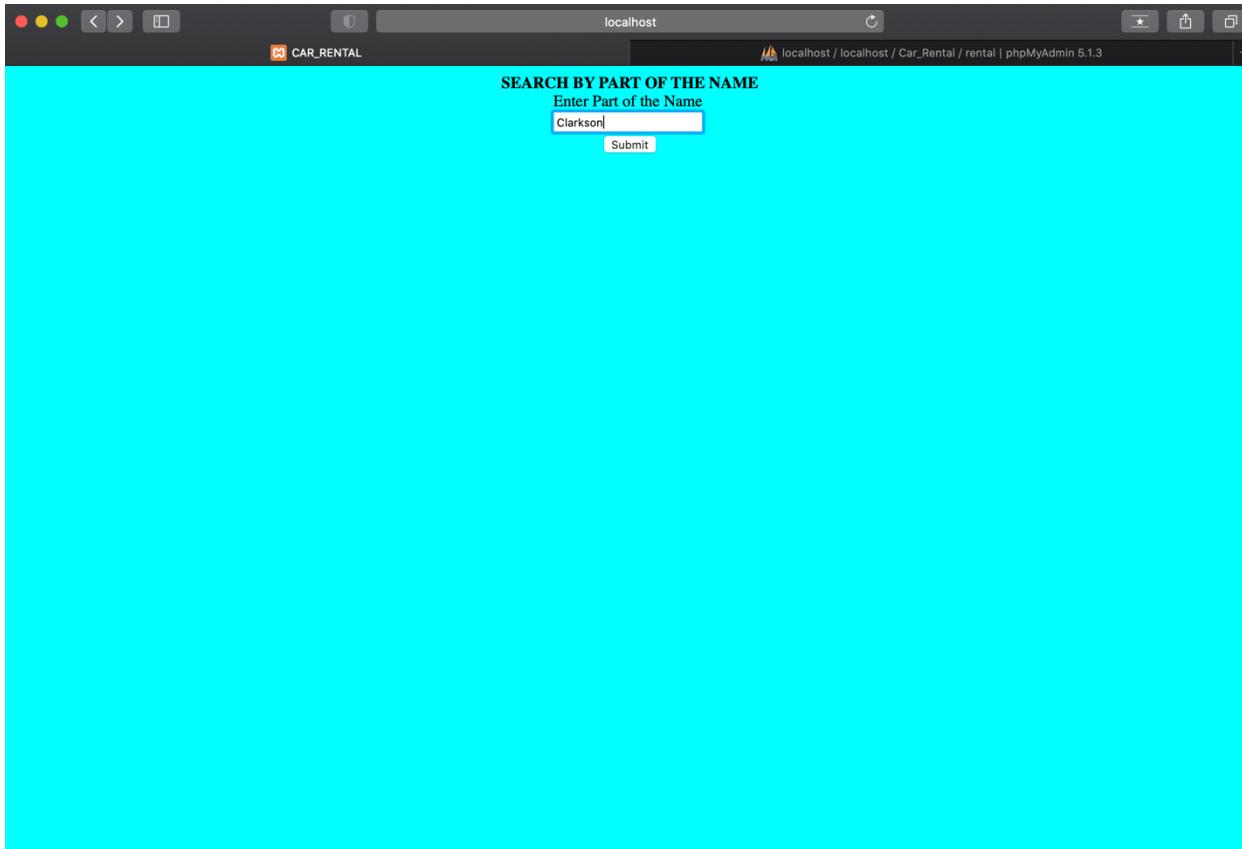
The screenshot shows a web browser window titled "CAR\_RENTAL" with the URL "localhost / Car\_Rental / rental | phpMyAdmin 5.1.3". The page displays a search form titled "SEARCH BY CUSTOMER NAME" with the instruction "Enter First Name Initial and Last Name". A text input field contains "J. Brown" and a "Submit" button is visible below it.

The screenshot shows a web browser window titled "localhost / Code / search\_name\_query.php" with the URL "localhost / Car\_Rental / rental | phpMyAdmin 5.1.3". The page displays a table with three columns: "Customer ID", "Customer Name", and "Remaining Balance". One row is present, showing "221" in the Customer ID column, "J. Brown" in the Customer Name column, and "\$14,400.00" in the Remaining Balance column. Below the table, the text "Total number of Rows: 1" is displayed.

Customer ID	Customer Name	Remaining Balance
221	J. Brown	\$14,400.00

## Search a Payment Due by Part of Name

```
SELECT CustomerID, CustomerName,  
       SUM(RentalBalance) AS RemainingBalance  
    FROM vrentalinfo  
   WHERE CustomerName LIKE '%$PName%'  
GROUP BY CustomerID;
```



localhost

CAR\_RENTAL

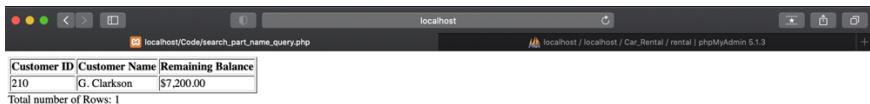
localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

SEARCH BY PART OF THE NAME

Enter Part of the Name

Clarkson

Submit



localhost

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

Customer ID	Customer Name	Remaining Balance
210	G. Clarkson	\$7,200.00

Total number of Rows: 1

Display all customer who have rented and their total due: No Filter

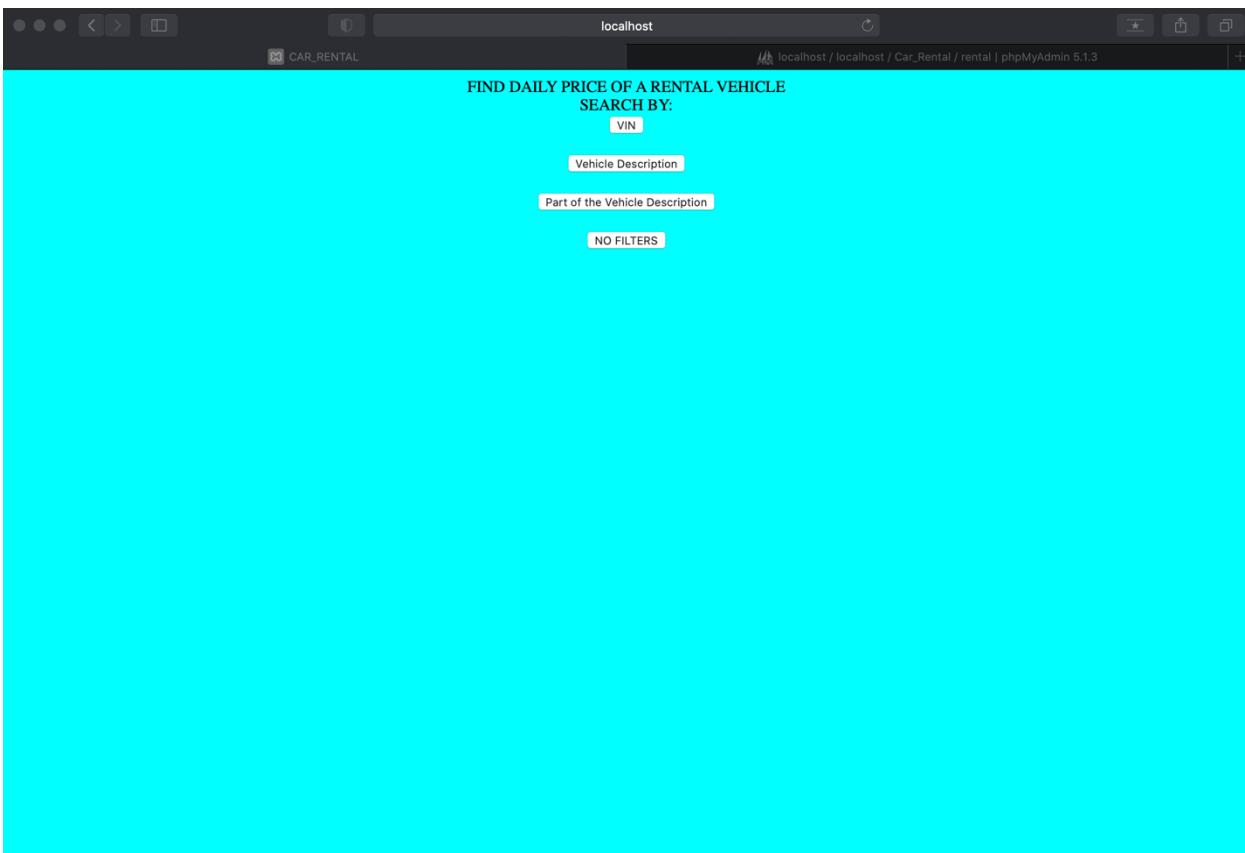
```
SELECT CustomerID, CustomerName,
       SUM(RentalBalance) AS RemainingBalance
    FROM vrentalinfo
   GROUP BY CustomerID
  ORDER BY RemainingBalance ASC;
```

The screenshot shows a web browser window with two tabs. The left tab is titled 'localhost/Code/search\_nofilter\_query.php?' and the right tab is titled 'localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3'. The main content area displays a table with 7 rows of data, representing the results of the SQL query. The table has three columns: Customer ID, Customer Name, and Remaining Balance.

Customer ID	Customer Name	Remaining Balance
233	B. Saud	\$0.00
212	H. Gallegos	\$0.00
203	A. Hernandez	\$0.00
229	D. Kirkpatrick	\$0.00
216	A. Hess	\$0.00
210	G. Clarkson	\$7,200.00
221	J. Brown	\$14,400.00

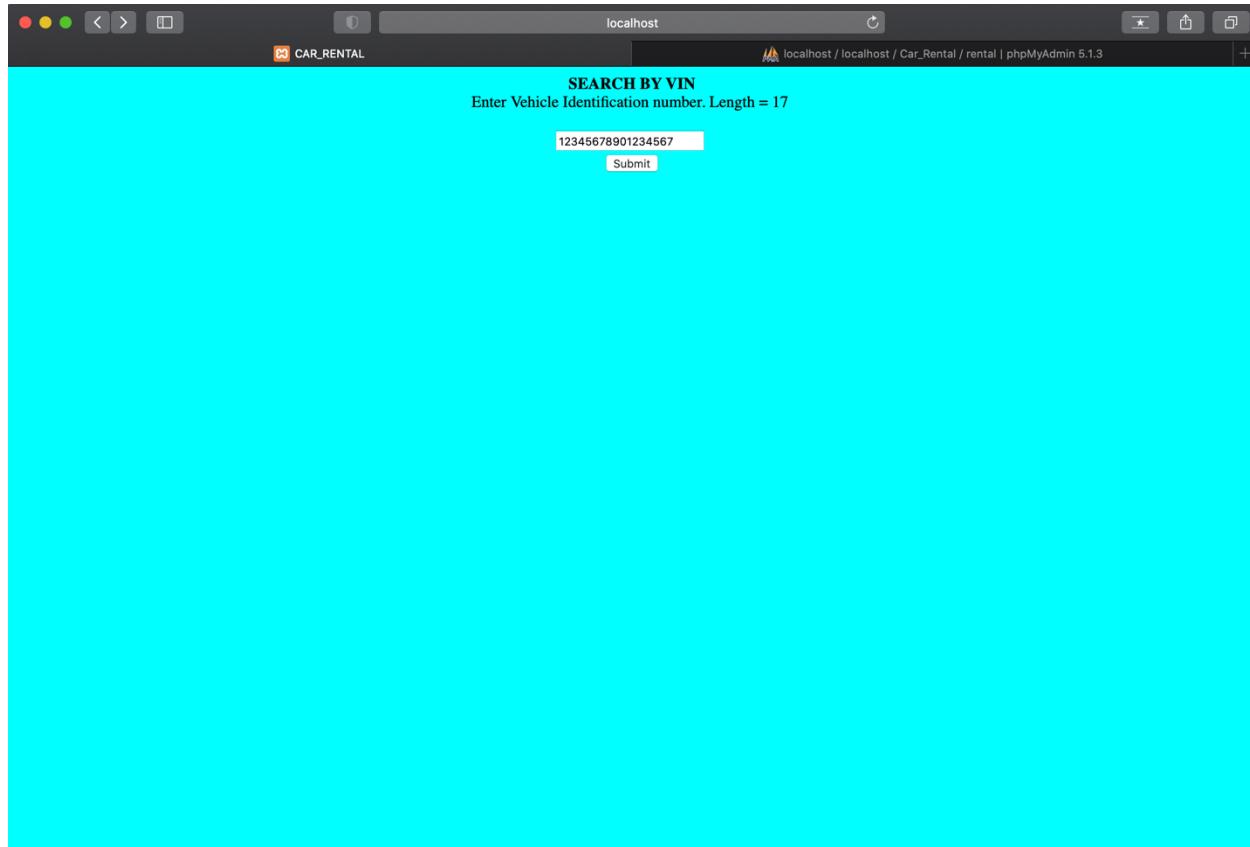
Total number of Rows: 7

**b. List for every vehicle the VIN, the description, and the average DAILY price. The user has the right either to search by the VIN, vehicle's description, part of the description, or to run the query with no filters/criteria. An example criterion would be all 'BMW' vehicles. The amount needs to be in US dollars. The average DAILY price derives from the rental table, and the amount needs to have two decimals as well as the dollar '\$' sign. For vehicles that they do not have any rentals, you need to substitute the NULL value with a 'Non-Applicable' text. Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the average daily price. Submit your editable SQL query that your code executes. [10 points]**



## Search Daily Price By VIN

```
SELECT vehicle.VehicleID, vehicle.Description,
       (
         CASE
           WHEN EXISTS (SELECT NULL FROM vrentalinfo WHERE VIN = :VIN)
           THEN CONCAT( '$', FORMAT( (AVG(vrentalinfo.OrderAmount/vrentalinfo.TotalDays)), 2 ) )
           ELSE 'Non-Applicable'
         END)
       AS Average_Daily_Price
  FROM vehicle, vrentalinfo
 WHERE vehicle.VehicleID = :VIN AND vrentalinfo.VIN = :VIN;
```



localhost

localhost/Code/search\_VIN\_query.php

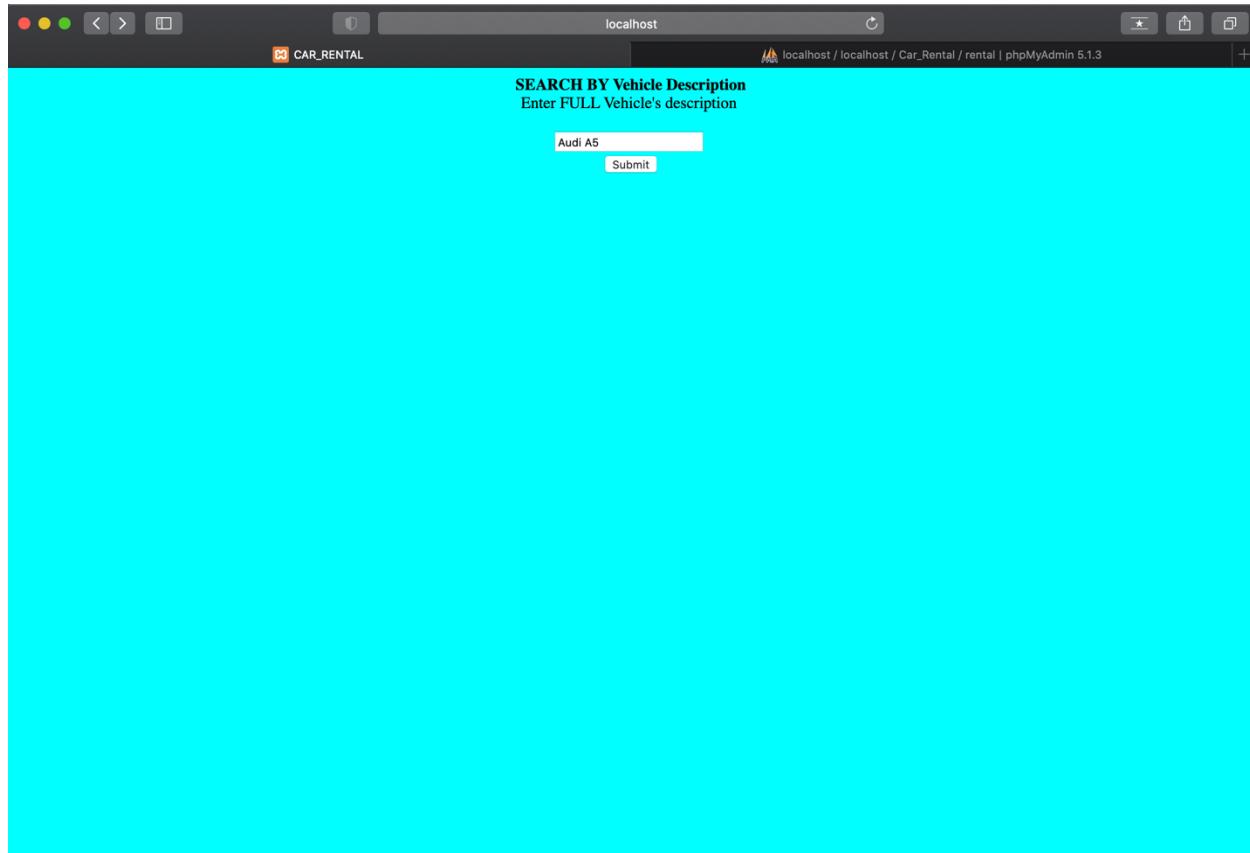
VIN	Vehicle Description	Average Daily Price
12345678901234567	"Test Vehicle "	\$94.29

Total number of Rows: 1

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

## Search Daily Price By full description of Vehicle

```
SELECT vehicle.VehicleID, vehicle.Description ,  
      (  
        CASE  
          WHEN EXISTS (SELECT NULL FROM vrentalinfo WHERE vrentalinfo.Vehicle =  
CONCAT('\"",$description,"'))  
            THEN CONCAT( '$', FORMAT( (AVG(vrentalinfo.OrderAmount/vrentalinfo.TotalDays)), 2 ) )  
          ELSE 'Non-Applicable'  
        END)  
        AS Average_Daily_Price  
      FROM vehicle, vrentalinfo  
      WHERE vehicle.Description = CONCAT('\"",$description,"') AND vrentalinfo.Vehicle =  
CONCAT('\"",$description,"')  
      GROUP BY vehicle.VehicleID;
```



localhost

localhost/Code/search\_description\_query.php

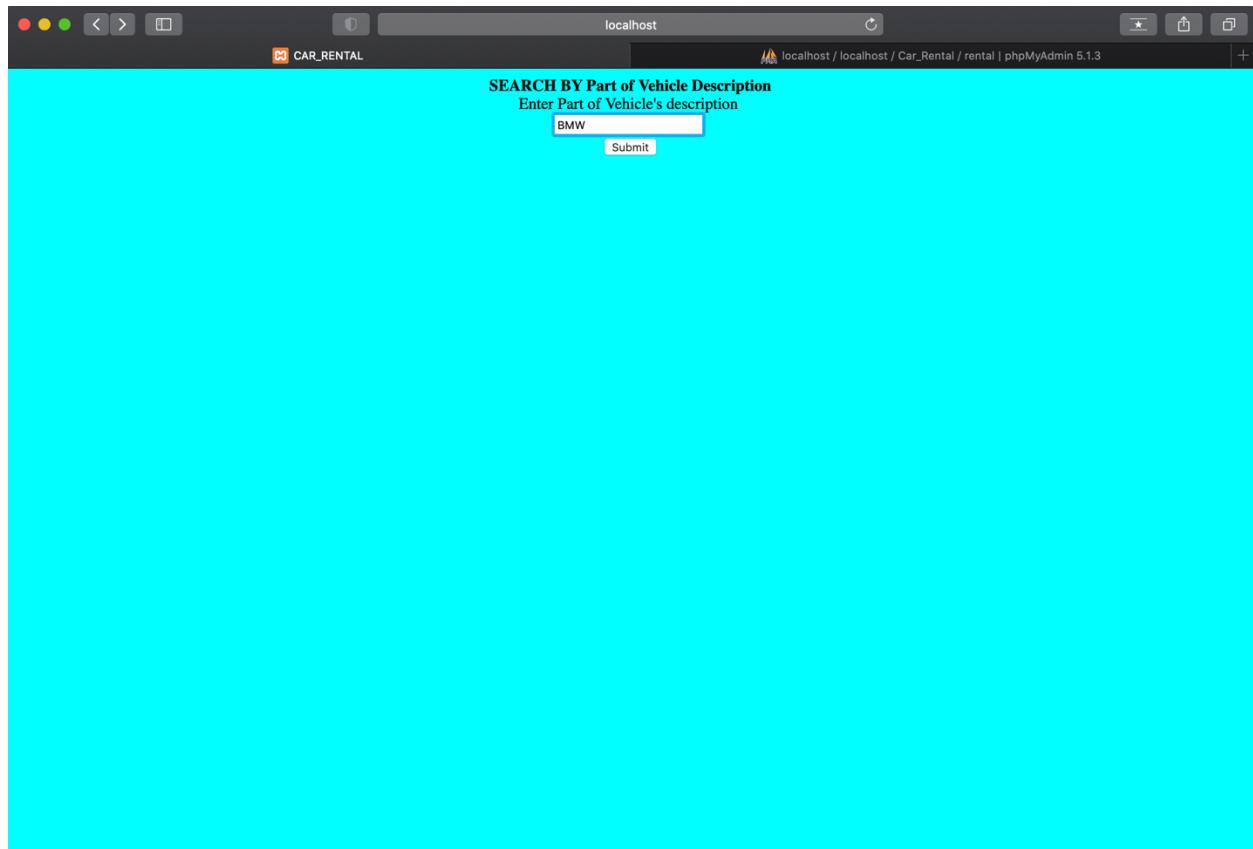
VIN	Vehicle Description	Average Daily Price
WAUTFAFH0E0010613	"Audi A5"	\$91.43

Total number of Rows: 1

localhost / localhost / Car\_Rental / rental | phpMyAdmin 5.1.3

## Search Daily Price By part of Vehicle Description

```
SELECT vehicle.VehicleID AS VIN, vehicle.Description AS Vehicle_Description , 'Non-Applicable' AS Average_Daily_Price
    FROM vehicle
   WHERE vehicle.Description LIKE '%$Pdescription%' AND VehicleID IN
        (SELECT vehicle.VehicleID
         FROM vehicle
        LEFT JOIN vrentalinfo ON vehicle.VehicleID = vrentalinfo.VIN
        WHERE vrentalinfo.VIN is NULL)
UNION
SELECT VIN, Vehicle AS Vehicle_Description ,
CONCAT( '$', FORMAT( (AVG(OrderAmount/TotalDays)), 2 ) )
     AS Average_Daily_Price
    FROM vrentalinfo
   WHERE Vehicle LIKE '%$Pdescription%'
  GROUP BY VIN;
```



A screenshot of a web browser window. The address bar shows "localhost/Code/search\_part\_description\_query.php". The main content area displays a table with three rows of vehicle data. Below the table, a message indicates the total number of rows found.

VIN	Vehicle Description	Average Daily Price
WBAVL1C57EVR93286	"BMW X1"	Non-Applicable
WBA3A9G51ENN73366	"BMW 3 Series"	\$85.71
WBA3B9C59EP458859	"BMW 3 Series"	\$85.71

Total number of Rows: 3

To Find the daily prices with for all the vehicles: no Filtering

```
SELECT vehicle.VehicleID AS VIN, vehicle.Description AS Vehicle_Description , 'Non-Applicable' AS Average_Daily_Price
    FROM vehicle
   WHERE VehicleID IN
        (SELECT vehicle.VehicleID
         FROM vehicle
        LEFT JOIN vrentalinfo ON vehicle.VehicleID = vrentalinfo.VIN
        WHERE vrentalinfo.VIN is NULL)
UNION
SELECT VIN, Vehicle AS Vehicle_Description ,
CONCAT( '$', FORMAT( (AVG(OrderAmount/TotalDays)), 2 ) )
     AS Average_Daily_Price
  FROM vrentalinfo
 GROUP BY VIN
 ORDER BY CAST( REPLACE(REPLACE(REPLACE( Average_Daily_Price ,':', ''), '$', ''), 'Non-Applicable', '000') AS SIGNED ) ASC;
```

VIN	Vehicle Description	Average Daily Price
IHGCR2E3XEA305302	"Honda Accord"	Non-Applicable
SNPDH4AE2FH565275	"Hyundai Elantra"	Non-Applicable
JTHCE1BL3F151DE04	"Lexus GS 350"	Non-Applicable
IFTNF1CF2EKE54305	"Ford F Series Pickup"	Non-Applicable
WBAVL1C57EVR93286	"BMW X1"	Non-Applicable
SFNRL6H58KB133711	"Honda Odyssey"	Non-Applicable
JN8AS5MV0FW760408	"Nissan Rogue Select"	Non-Applicable
KNDPCCA65F7791085	"KIA Sportage"	Non-Applicable
4S4BRCFC1E3203823	"Subaru Outback"	Non-Applicable
JH4KC1F56EC000095	"Acura RLX"	Non-Applicable
KNAGN4AD2F5084324	"Kia Optima Hybrid"	Non-Applicable
3N1CE2CP0FL409472	"Nissan Versa Note"	Non-Applicable
KMHTC6AD8EU998631	"Hyundai Veloster"	Non-Applicable
2HGF82F94FH501940	"Honda Civic"	Non-Applicable
5XYKUDA77EG449709	"Kia Sorento"	Non-Applicable
JTMBFREV1FJ019885	"Toyota RAV4"	Non-Applicable
1N4AB7AP2EN855026	"Nissan Sentra"	Non-Applicable
STDBKRHF4ES26D590	"Toyota Highlander"	Non-Applicable
JTHDL5EF9F5007221	"Lexus LS 460"	Non-Applicable
1G1JD5SB3E4240835	"Chevrolet Optra"	Non-Applicable
YV440MDD6F2617077	"Volvo XC60"	Non-Applicable
SJ6RM4H90FL028629	"Honda CR-V"	Non-Applicable
WDCGG0EB0EG188709	"Mercedes_Benz GLK"	\$85.71
WBA3A9G51ENN73366	"BMW 3 Series"	\$85.71
WBA3B9C59EP458859	"BMW 3 Series"	\$85.71
JTHFF2C26F135BX45	"Lexus IS 250C"	\$85.71
19VDE1F3XEE414842	"Acura ILX"	\$90.48
WAUTFAFH0E0010613	"Audi A5"	\$91.43
12345678901234567	"Test Vehicle "	\$94.29
1N6BF0KM0EN101134	"Nissan NV"	\$106.43
JM3KE4DY4F0441471	"Mazda CX5"	\$115.00

Total number of Rows: 62