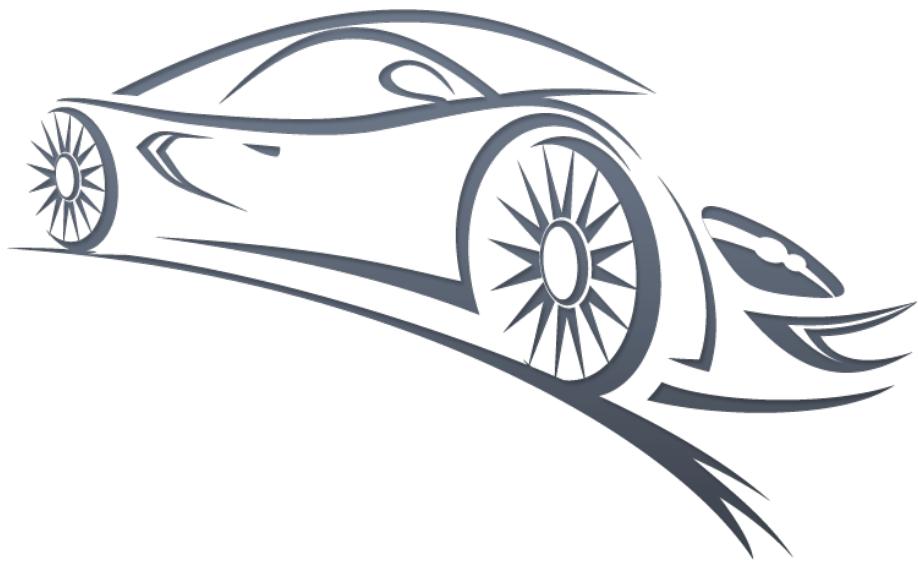


# Database Systems

## Car Rental System

### Final Project

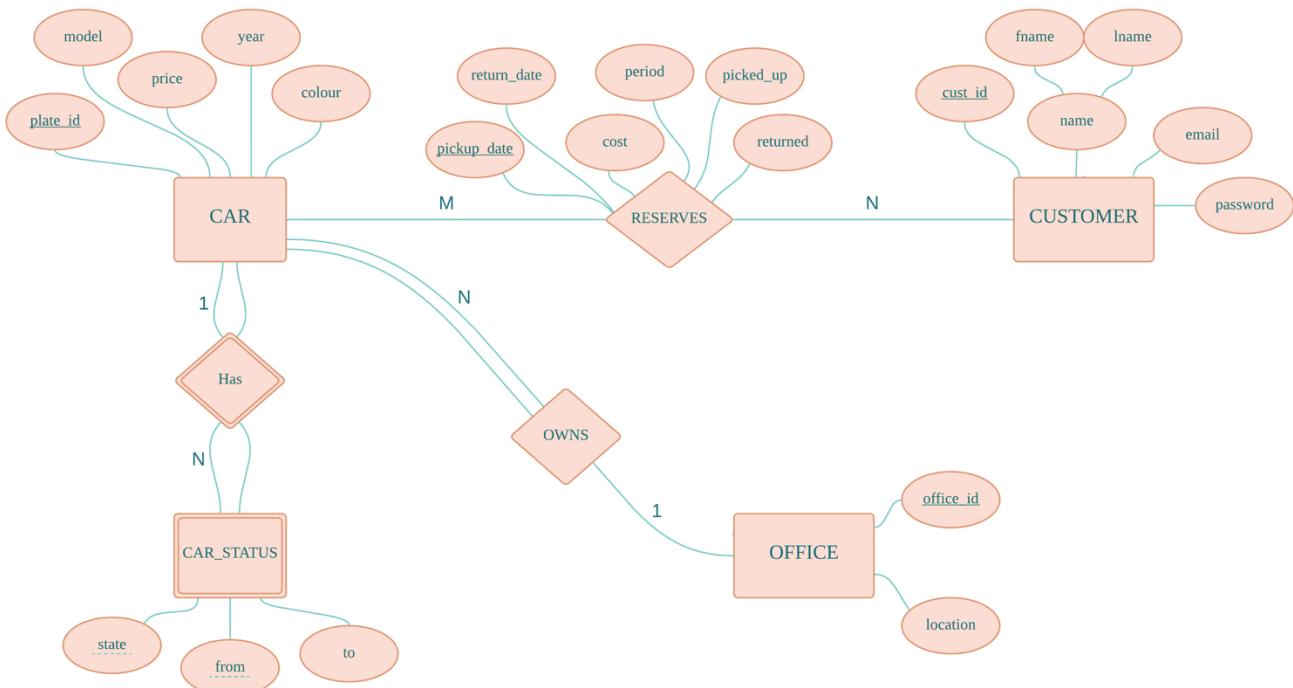


# Overview:

The car rental system developed is web-based system for a company that rents out cars to customers. The system enables the customers to rent a car anywhere in the world where it has an office. The customer is just asked to log in to his account and search for a car that meets his needs for a certain rental period. He will then be shown the cars available meeting his searching criteria and its price for a rental day. The system will then ask him to confirm his payment for the reservation. Upon confirmation, voila, a reservation has just been made. System administrator has full access to all the company's managerial aspects starting from adding a car to the system, updating the rental price of a certain car to printing out reports. The reports may be regarding the cars available, customers of the company, reservations made, etc.

# ER-Diagram:

CAR\_RENTAL\_SYSTEM



# DDL used for Implementing the ER-Diagram:

```
CREATE SCHEMA CAR_RENTAL_SYSTEM;
USE CAR_RENTAL_SYSTEM;
CREATE TABLE CAR (
    plate_id varchar(10),
    model varchar(20) not null,
    `year` int(4) not null,
    colour varchar(20),
    price decimal(8,2) not null,
    office_id int not null,
    PRIMARY KEY (plate_id)
);
CREATE TABLE CAR_STATUS(
    plate_id varchar(10),
    state varchar(20),
    `from` date not null,
    `to` date,
    PRIMARY KEY (plate_id,state,`from`)
);
CREATE TABLE OFFICE(
    office_id int AUTO_INCREMENT,
    location varchar(30) not null,
    PRIMARY KEY (office_id)
);
CREATE TABLE CUSTOMER(
    cust_id int AUTO_INCREMENT,
    fname varchar(30) not null,
    lname varchar(30) not null,
    email varchar(50) not null,
    password varchar(256) not null ,
    PRIMARY KEY (cust_id)
);
CREATE TABLE RESERVATION (
    cust_id int,
    plate_id varchar(10),
    pickup_date date,
    return_date date not null,
    cost decimal(12,2) not null,
    period int not null,
    picked_up char(1),
    returned char(1),
    PRIMARY KEY(cust_id,plate_id,pickup_date)
);

/*
-----*/
ALTER TABLE CAR_STATUS ADD CONSTRAINT STATCARFK FOREIGN KEY (plate_id) REFERENCES CAR
(plate_id);
ALTER TABLE CAR ADD CONSTRAINT OFFCARFK FOREIGN KEY (office_id) REFERENCES OFFICE
(office_id);
ALTER TABLE RESERVATION ADD CONSTRAINT RESCUSTFK FOREIGN KEY (cust_id) REFERENCES CUSTOMER
(cust_id);
ALTER TABLE RESERVATION ADD CONSTRAINT RESCARFK FOREIGN KEY (plate_id) REFERENCES CAR
(plate_id);
```

# Login Page

The first page that appears to the system user will be the [login](#) page (Figure 1). The user will be prompted to enter his credentials (Email Address, Password and Role). Verifications using Ajax will occur upon clicking the Login button on the data received to find a matching registered system user. Thereafter, the user is either allowed access to his account or a JavaScript error message would appear stating that no match was found for the credentials entered. If a new customer would like to register, he just has to click on the Register hyperlink provided.

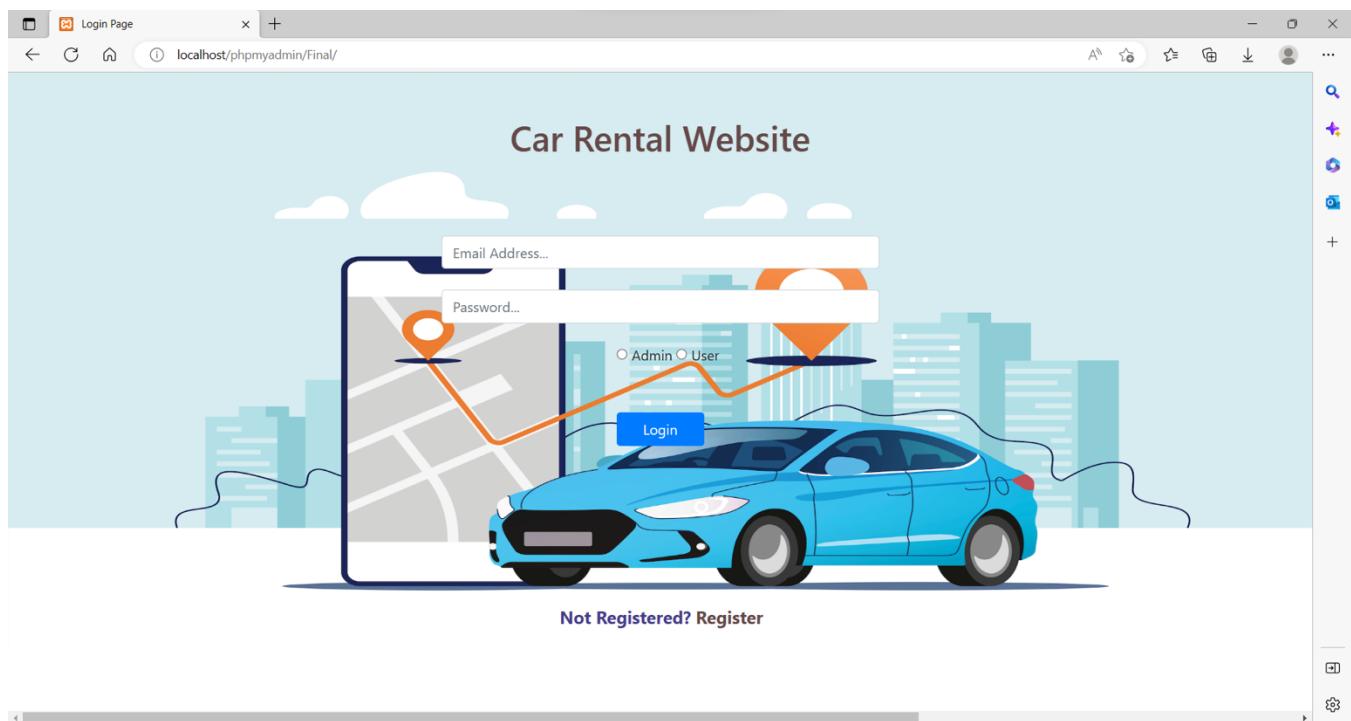


FIGURE 1: LOGIN PAGE

# Registration Page

Upon clicking on the [Register](#) hyperlink, the user will be redirected to an html document having a form for registration (Figure 2).The user will be asked to enter his name (first and last), email address and password for his account. Validations are done using JavaScript to check that an email with valid format has been entered, password and its confirmation field are matching, none of the fields may be left blank, etc. Ajax is used and redirects this html file to another php file to check that the email entered has no match existing in the database already. If no match was found, the data entered by the user would be inserted to the Customer relation in the database.

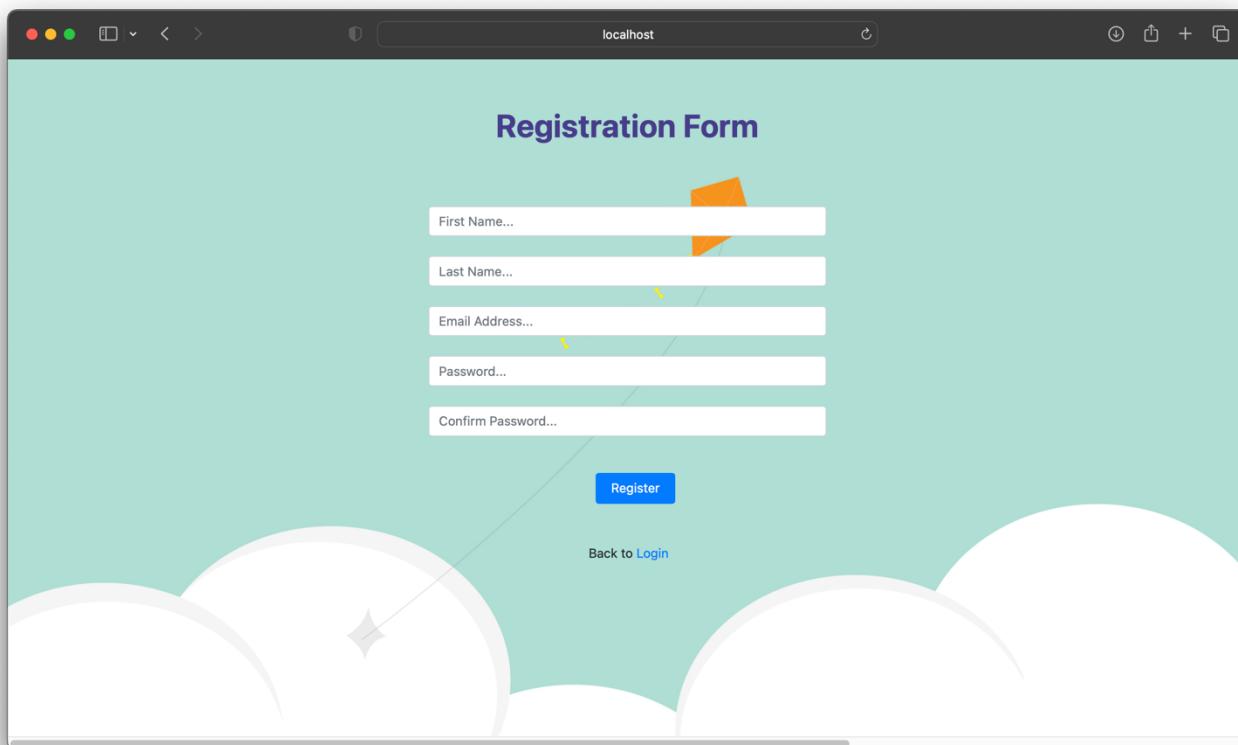


FIGURE 2: REGISTRATION PAGE

# Admin Section

## Admin Home

When the admin logs in, they are redirected to the Admin Home webpage (Figure 3). Containing 13 buttons, in a clear and concise layout, this webpage is the admin's gateway to all the tasks they wish to perform on the system; whether it's an insertion of a new car, an update to a field of an existing one, or a specific type of search, the admin is just one click away from their desired operation.

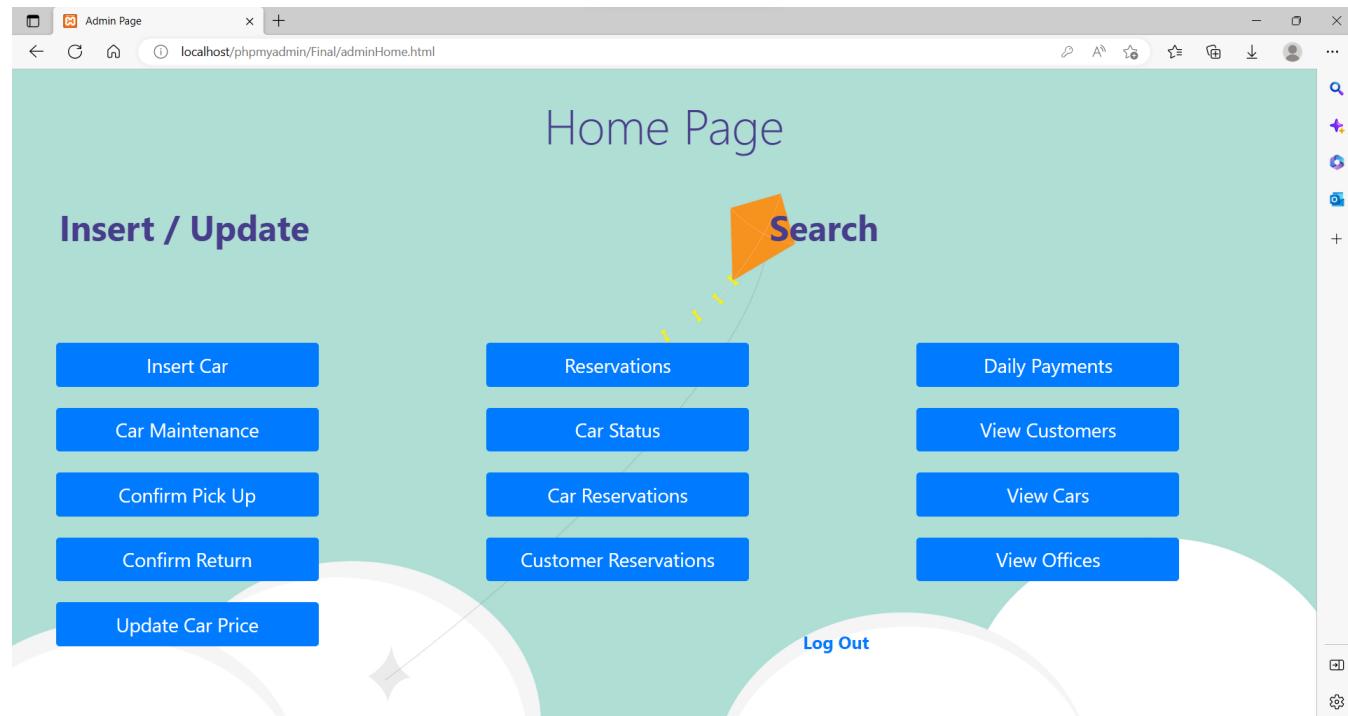


FIGURE 3: ADMIN HOME PAGE

## Insert Car

This button redirects the admin to a new car registration form (Figure 4). Registration cannot occur if any of the form input fields is left empty. Therefore, the validate() function was implemented to notify the admin if they did not fill a required field. When the “Register Car” button is clicked, further checks and queries are run to ensure the admin’s input correctness; that is, that every input conforms to the type of its field (prices are numbers, colours are strings, etc.), and that the car to be inserted has a unique plate ID and belongs to an existing office. If input fails to meet any of these requirements, admin receives a customized alert of the problem behind failure. Otherwise, the new car is successfully inserted into the Car table using the following query: "insert into car (plate\_id, model, year, colour, price, office\_id) values ('".\$plate\_id."', '".\$model."', '".\$year."', '".\$colour."', '".\$price."', '".\$office\_id."') " Admin is notified upon successful registration.

The screenshot shows a web browser window titled "Insert Car". The URL in the address bar is "localhost/phpmyadmin/Final/insertcar.html". The main content is a "Car Registration Form" with the following fields:

- Plate ID...
- Model...
- Year...
- Colour...
- Price...
- Office ID...

A large orange play button icon is overlaid on the top-left area of the form. At the bottom center is a blue "Register Car" button. Below the form, there is a link "Back to Admin Home". The browser interface includes standard navigation buttons (back, forward, search, etc.) and a toolbar on the right.

FIGURE 4: CAR REGISTRATION FORM

## Car Maintenance

Admin must know when a car is available for renting, and when it is out of service for maintenance reasons and must be able to update the system with the car's current state. Clicking this button displays a table of all cars and their states in specific periods and gives the admin the option to change these states through the "Available" and "Out of Service" buttons. If an initially available car is to be declared out of service through the "Out of Service" button, the following changes are made to ensure the database system holds: First, the sql query "delete from car\_status where `from` >= '".\$from."' and plate\_id = '".\$plate\_id.'"'" is run to remove any previous information about the car's state from the date of declaring it out of service and onward. Then, the Car\_status table is updated using the query "insert into car\_status (plate\_id, state, `from`) values ('".\$plate\_id."', '".\$out."', '".\$from."')" which declares the car out of service to an indefinite period, until maintenance period finishes.

Previous reservations of an out of service car in this specific period must be deleted from the Reservation table for the database to hold; this is done by the following query "delete from reservation where (pickup\_date >= '".\$from."' or return\_date >= '".\$from."') and plate\_id = '".\$plate\_id.'"'" and an alert is displayed to inform the admin of the number of cancelled reservations. The webpage is then updated, showing all the previously available dates of the car and that it's currently out of service.

If maintenance of an Out of Service car was completed, and the car is to be re-declared "Available" through the button, all tuples with the state "Available" until the current date are deleted from the Car\_status table, because they are no longer of use. This specification has a goal of not losing previous rental records. This is accomplished through the query "delete from car\_status where ((state != 'Rented') or state != 'Not Available') and plate\_id= '".\$plate\_id.'" ". The Car\_status is later updated through the following query "insert into car\_status values ('".\$plate\_id."', '".\$av."', '".\$today."', NULL)" which declares the car available for rental from the current date indefinitely. The "Update Car State" webpage is updated accordingly.

Update Car State

Plate Id	State	From	To	Button
56BHN9	Available	2022-01-01	2022-05-09	OutofService
56BHN9	Available	2022-05-16		OutofService
56BHN9	Out of Service	2022-05-10	2022-05-15	Available
5YZ3F2	Available	2022-01-01		OutofService
ABC5DY	Available	2022-01-01		OutofService
FY4321	Available	2022-01-01		OutofService
H3F2N7	Available	2022-01-01	2022-03-01	OutofService
H3F2N7	Available	2022-03-16		OutofService
KLM09M	Available	2022-01-01		OutofService
MS2011	Available	2022-01-01		OutofService

FIGURE 5: CAR MAINTENANCE PAGE

## Confirm Pickup

This button redirects the admin to a table of all reserved cars that are yet to be picked up (Figure 6), this table is retrieved through the following query: "select \* from reservation where picked\_up='n'" When a customer picks up their rented car from the office, admin confirms that the car is now in the customer's possession by clicking the "Pickup" button, which updates the database, specifying that the rented car has been picked up through this query: "update reservation set picked\_up = ".\$val." where cust\_id =".\$cust\_id." and plate\_id = ".\$plate\_id." and pickup\_date = ".\$pickup\_date.". The webpage is then reloaded, with the picked-up car removed.

Customer Id	Plate Id	Pick Up Date	Return Date	Cost	Period	Picked Up	Returned	Pick Up
1	H3F2N7	2022-03-02	2022-03-05	8000.00	4	n	n	<button>Pick Up</button>
1	H3F2N7	2022-03-06	2022-03-10	10000.00	5	n	y	<button>Pick Up</button>
2	H3F2N7	2022-03-11	2022-03-12	4000.00	2	n	y	<button>Pick Up</button>
3	H3F2N7	2022-03-13	2022-03-15	6000.00	3	n	y	<button>Pick Up</button>

FIGURE 6: PICKUP CAR PAGE

## Confirm Return

This button redirects the admin to a table of all reserved, picked up cars that are yet to be returned (Figure 7), this table is retrieved through the following query: "select \* from reservation where reservation.picked\_up = 'y' AND reservation.returned='n'" When a customer returns their rented car to the office, admin confirms that the car is now back in the office's possession by clicking the "Return" button, which updates the database, specifying that the rented car has been returned successfully through this query "update reservation set reservation.returned = "'.\$yesvalue.'" where reservation.plate\_id='".\$plate\_id.'" AND reservation.cust\_id='".\$cust\_id.'" AND reservation.pickup\_date='".\$pickup\_date.'" ". The webpage is then reloaded, with the returned car removed.

Customer Id	Plate Id	Pick Up Date	Return Date	Cost	Period	Picked Up	Returned	Return
1	H3F2N7	2022-03-02	2022-03-05	8000.00	4	y	n	<button>Return</button>
1	H3F2N7	2022-03-06	2022-03-10	10000.00	5	y	n	<button>Return</button>

[Back To Admin Page.](#)

FIGURE 7: RETURN CAR PAGE

## Update Car Price

This button redirects the admin to a form where they need to specify the car's plate ID and new price (Figure 8) . Admin's input correctness is inspected using validation checks and queries, as it must conform with the input field's type and the plate ID must already exist on the system. Admin is notified if input fails to meet the requirements, otherwise, price of the selected car is updated using the query "update car set price = ".\$price." where plate\_id = ".\$plate\_id."".

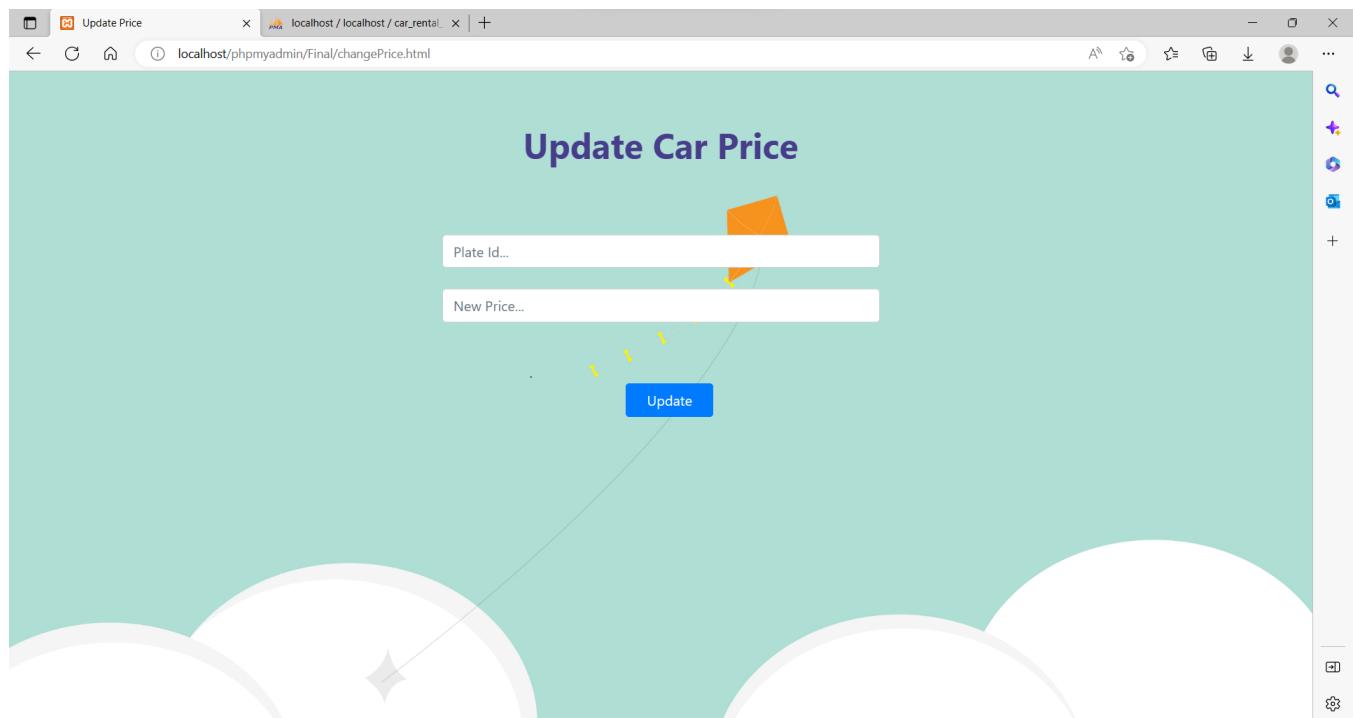


FIGURE 8: UPDATE CAR PRICE

## Reservations

The admin is allowed to see all the reservations that have been made in the system. So once, the admin clicks the reservations bottom, he is redirected into another page containing a form (Figure 9). To get the reservations made during a specific period, the admin is asked to enter a start date and an end date. By click the search button, finally another webpage appears showing a table containing all the reservations made in the specified period. All the reservation information is shown in the table involving the customer and the car reserved (Figure 10).

To get the information shown in this table, a SQL query is used to get the data from the RESERVATION table, which is joined with two other tables (CAR, CUSTOMER), to get the full information.

The query used:

```
"SELECT cust_id, fname, lname, email, plate_id, pickup_date, return_date, cost, period,  
picked_up, returned, model, year, colour, price, office_id from RESERVATION Natural Join  
CAR Natural Join CUSTOMER where pickup_date>='".$start."' and  
return_date<'".$end."'"
```

It has to be noted that if the admin did not specify the start date or the end date or both, a table is also shown depending on the input. If start date and end date are both empty, all reservations are brought from database. Else, if the admin only specified the start date, any reservations after that date appears. Finally, if he only entered an end date, reservations before that date appear. So, the previous query is varied for each of the previous conditions.

Get reservations within specific period

Start Date...

End Date...

Search

FIGURE 9: FORM PAGE FOR RETRIEVING RESERVATIONS

Customer Id	First Name	Last Name	Email	Plate Id	Pickup Date	Return Date	Cost	Period	PickedUp	Returned	Model	Year	Color	Price	Office Id
1	Adham	Amr	adham@gmail.com	H3F2N7	2022-03-02	2022-03-05	8000.00	4	y	y	KIA	2019	BLACK	2000.00	2
1	Adham	Amr	adham@gmail.com	H3F2N7	2022-03-06	2022-03-10	10000.00	5	y	y	KIA	2019	BLACK	2000.00	2
2	Arsany	Mousa	arsany66@gmail.com	H3F2N7	2022-03-11	2022-03-12	4000.00	2	y	y	KIA	2019	BLACK	2000.00	2
3	Youssef	Amr	joe@gmail.com	H3F2N7	2022-03-13	2022-03-15	6000.00	3	y	y	KIA	2019	BLACK	2000.00	2

FIGURE 10: RESERVATIONS PAGE

## Car Status

This button provides the user with information about the state of each car on a specific day. A car can be either: “available”, “rented”, “out of service” or “not available”.

By clicking the car status button, the admin is given a form asking for the day he wants to see the state of each car (Figure 11). By entering the date, a table appears containing the state of each car on that day (Figure 12).

This information is brought from the database by a query that just access on table which is the CAR\_STATUS table and get the state of each car on that day by checking that the (from) field of the record is before the date entered by the user, and the (to) field is after it.

The query used:

```
"select plate_id, state from car_status where ('$date' between car_status.from AND car_status.to) OR ('$date' >= car_status.from AND car_status.to IS NULL)"
```

The screenshot shows a web browser window with the title bar "Car Status". The address bar displays "localhost/phpmyadmin/Final/status.html". The main content area has a light green background with white clouds at the bottom. A central heading says "Car Status At Specific Day". Below it is a search input field with the placeholder "Day's date...". To the right of the input field is a blue button labeled "Search". A yellow dashed arrow points from the text "Status" in the question to the "Search" button.

FIGURE 11: FORM PAGE FOR RETRIEVING STATUS ON A SPECIFIC DAY

The screenshot shows a web browser window with the title bar "localhost/phpmyadmin/Final/status.php". The address bar displays "localhost/phpmyadmin/Final/status.php". The main content area has a light green background with white clouds at the bottom. A central heading says "Car Status". Below it is a table with two columns: "Plate ID" and "State". The table lists eight rows of data. A yellow dashed arrow points from the text "Status" in the question to the "Available" entry in the second row.

Plate ID	State
56BHN9	Out of Service
5YZ3F2	Available
ABC5DY	Available
FY4321	Available
H3F2N7	Available
KLM09M	Available
MS2011	Available
MS7BA8	Available

FIGURE 12: CAR STATUS

## Car Reservations

Clicking this button allow the admin to get the reservations made on a specific car in the system. The admin specifies the car he wants and a period to check the reservations of the car within it as required by the form shown in Figure 13. It is very similar to customer reservations button; the admin can specify any of the attributes of a car and will get the corresponding result and display them in a table as shown in Figure 14.

The query here joins the CAR table with RESERVATION table, then filter the tuples to select those that meet the conditions.

The screenshot shows a web browser window titled "Car Reservations". The URL in the address bar is "localhost/phpmyadmin/Final/reservations\_car\_info.html". The main content area has a teal background and features a title "Reservations (Search by Car)" at the top. Below the title are seven input fields arranged vertically: "Start Date...", "End Date...", "Plate Id...", "Model...", "Price...", "Year...", and "Colour...". A blue "Search" button is located at the bottom right of these fields. The browser interface includes standard navigation buttons (back, forward, home) and a toolbar with various icons on the right side.

FIGURE 13: FORM PAGE FOR RETRIEVING RESERVATIONS OF A CERTAIN CAR

localhost/phpmyadmin/Final/res... x +

localhost/phpmyadmin/Final/reservations\_car\_info.php

### Reservations

Plate Id	Model	Year	Colour	Price	Office Id	Customer Id	Pickup Date	Return Date	Cost	Period	PickedUp	Returned
H3F2N7	KIA	2019	BLACK	2000.00	2	1	2022-03-02	2022-03-05	8000.00	4	y	y
H3F2N7	KIA	2019	BLACK	2000.00	2	1	2022-03-06	2022-03-10	10000.00	5	y	y
H3F2N7	KIA	2019	BLACK	2000.00	2	2	2022-03-11	2022-03-12	4000.00	2	y	y
H3F2N7	KIA	2019	BLACK	2000.00	2	3	2022-03-13	2022-03-15	6000.00	3	y	y

FIGURE 14: RESERVATIONS OF A CERTAIN CAR

## Customer Reservations

The admin also can get all the reservations made by a specific customer. By clicking the button, a form appears asking the admin to enter some information about the customer required (Figure 15). If the admin did not enter any information, the table resulted will contain all the customers' reservations. By putting any in specifications, the resulted table will become more specific as shown in Figure 16. For example, if the admin entered a name in the 'Last Name' field of the form, so the table will contain the reservations of all customers with this last name. So, the admin is free to leave any field in the form empty.

To get the information from the database, the query used just join the customer table with reservations table, and only select the tuples that meet the specifications entered in the form.

To form that query every field in the form is checked whether it is empty or not. So, the query begins first by joining the two tables, then step by step the conditions are concatenated to the query to select the correct tuples.

Customer Reservations

localhost/phpmyadmin/Final/reservations\_cust\_info.html

## Reservations (Search by Customer)

Customer Id...

First Name...

Last Name...

Email...

Search

FIGURE 15: FORM PAGE FOR RETRIEVING RESERVATIONS OF A CERTAIN CUSTOMER

localhost/phpmyadmin/Final/reservations\_cust\_info.php

## Reservations

Customer Id	Plate Id	Car Model	Pickup Date	Return Date	Cost	Period	PickedUp	Returned	First Name	Last Name	Email
1	H3F2N7	KIA	2022-03-02	2022-03-05	8000.00	4	n	n	Adham	Amr	adham@gmail.com
1	H3F2N7	KIA	2022-03-06	2022-03-10	10000.00	5	n	y	Adham	Amr	adham@gmail.com
2	H3F2N7	KIA	2022-03-11	2022-03-12	4000.00	2	n	y	Arsany	Mousa	arsany66@gmail.com
3	H3F2N7	KIA	2022-03-13	2022-03-15	6000.00	3	n	y	Youssef	Amr	joe@gmail.com

FIGURE 16: RESERVATIONS OF A CERTAIN CUSTOMER

## Daily Payments

This button redirects the admin to a form that retrieves a table of total revenue per day within a specific period. Form input must be in the correct form to be accepted. Form is shown in Figure 17. If both start and end date are not specified, all payments are displayed using the query "select pickup\_date, sum(cost) from reservation group by pickup\_date". If only a start date is specified, all payments from that date onward are displayed as instructed by the query "select pickup\_date, sum(cost) from reservation where pickup\_date>='".\$start.'" group by pickup\_date". If only an end date is specified, all payments until that date are displayed as instructed by the query "select pickup\_date, sum(cost) from reservation where return\_date<='".\$end.'" group by pickup\_date". If both start and end dates are specified, all payments are displayed as instructed by the query "select pickup\_date, sum(cost) from reservation where pickup\_date>='".\$start.'" and pickup\_date<='".\$end.'" group by pickup\_date". If no payments were made in the specified period, "No payments are made in this period." is printed.

Get Daily Payments within specific period

Start Date...

End Date...

Search

FIGURE 17: FORM PAGE FOR RETRIEVING DAILY PAYMENTS

Day	Total Payment
2022-03-02	8000.00
2022-03-06	10000.00
2022-03-11	4000.00
2022-03-13	6000.00

FIGURE 18: DAILY PAYMENTS IN A CERTAIN PERIOD

## View Customers

Admin can also see all the customers registered on the system. So, clicking this button leads to a table containing the information of the customers. Also, a simple query is used to select the customers from the CUSTOMER table: "select \* from customer"

Customer Id	First Name	Last Name	Email
1	Adham	Amr	adham@gmail.com
2	Arsany	Mousa	arsany66@gmail.com
3	Youssef	Amr	joe@gmail.com
4	Youssef	Samuel	samo@gmail.com
5	Adham	ElQazzaz	adham22@gmail.com

FIGURE 19: CUSTOMERS OF THE COMPANY

## View Cars

This button allows the admin to see all the cars that are present in the system. A simple query is used that selects all the cars in the CAR table: “select \* from car”

Plate ID	Model	Year	Colour	Price	Office ID
56BHN9	MERCEDES	2021	WHITE	5000.00	1
5YZ3F2	BMW	2022	RED	3000.00	1
ABC5DY	Opel	2020	Gray	1000.00	1
FY4321	FIAT	2018	SILVER	1000.00	4
H3F2N7	KIA	2019	BLACK	2000.00	2
KLM09M	TOYOTA	2018	RED	2500.00	1
MS2011	ALFA ROMEO	2022	BLACK	2022.00	3
MS7BA8	HONDA	2021	BROWN	3000.00	3

FIGURE 20: CARS OF THE COMPANY

## View Offices

This button redirects the admin to a table of information about the company's offices (Figure 21). It contains each office's ID, location, and the number of cars belonging to this office. The following query joins the Office and Car tables to retrieve the desired information:

"select office\_id, location, count(\*) from office Natural Join car group by office\_id"

If the company has no offices, no table appears and "No offices" is printed for the admin.

Office ID	Location	Number Of Cars
1	London	4
2	Cairo	1
3	Berlin	2
4	Los Angeles	1

FIGURE 21: OFFICES OF THE COMPANY

# Customer Section

## Customer Home

The customer\_start webpage acts as the homepage for the customer. It displays a table on the left for the cars linked to that customer. Whether it's a future rent reservation or a car that he has rented before. On the right, a form exists if the customer would like to make a new car reservation. The form has fields so the customer can state the specific criteria he would like in the car. The pickup date and the return date are a must to be entered, as based on it, the available cars will be displayed. Advanced searching fields exist as to state the office he would like to pick up from, the desired price for the car, its model, make year, etc. All these fields combined will select the appropriate car(s) existing to be displayed in the next webpage available\_cars (if existing).

The screenshot shows a web browser window titled "Customer Page" with the URL "localhost/phpmyadmin/Final/customer\_start.php". The page is divided into two main sections. On the left, under the heading "Your Reservations", there is a table with the following data:

Plate ID	PickUp Date	Return Date	Cost	Period	PickedUp?	Returned?
H3F2N7	2022-03-02	2022-03-05	8000.00	4	y	y
H3F2N7	2022-03-06	2022-03-10	10000.00	5	y	y

On the right, there is a search form titled "Search for a car" with the sub-instruction "Enter the required period". It includes fields for "Pickup Date (YYYY-MM-DD)" and "Return Date (YYYY-MM-DD)". Below this, under "Enter car specifications", are fields for "Office Location...", "Model...", "Year...", "Color...", and "Search" button. At the bottom, there is a section titled "Enter the price range" with fields for "Minimum Price..." and "Maximum Price...". A "Back to Login" link is also present.

FIGURE 22: CUSTOMER HOME PAGE

## Available Cars

It's the webpage displaying the cars available matching the searching criteria. The user will be prompted to choose a car from the ones displayed by clicking on the reserve button beside it. Upon clicking on the button, the corresponding car becomes unavailable in the system for rentals until payment confirmation or cancellation is made. If another user tries to search for the availability for that specific car in that time period, it will not be displayed to him. Next page that will be loaded when a reserve button has been clicked is the payment page.

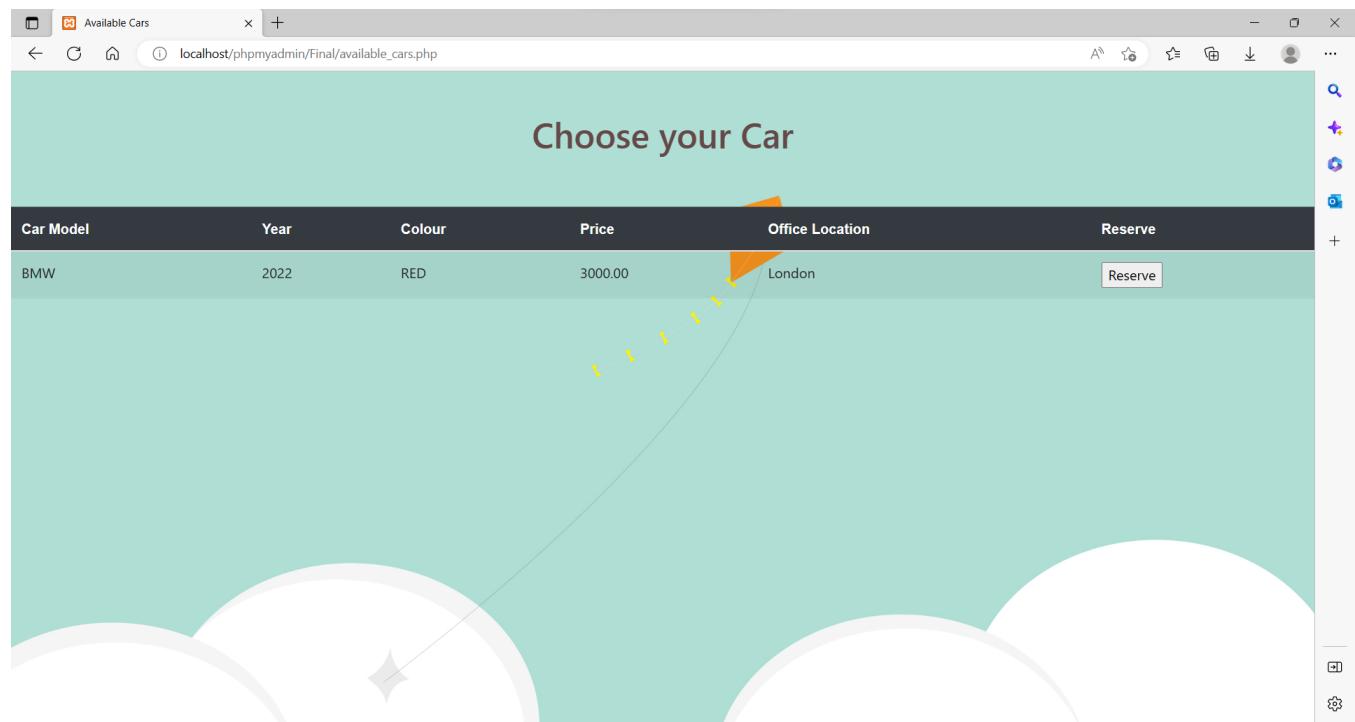


FIGURE 23: AVAILABLE CAR MATCHING THE SEARCHING CRITERIA

# Payment

The payment webpage acts as the confirmation page for the reservation. Details for the reservation will be displayed and then prompting the user for confirmation or cancellation (Figure 24). If the cancel button is pressed, rollback would occur to all the car status changing queries made so far. By that, the car becomes available once again as if nothing had happened on the database. If the confirm button is pressed, reservation would be confirmed and added to the Reservation relation on the database affirming it. Both buttons will redirect the user to the homepage (customer\_start) displaying an appropriate message regarding whether a reservation has been made or not. Case the reservation was confirmed, the customer will be headed to the page shown in Figure 25.

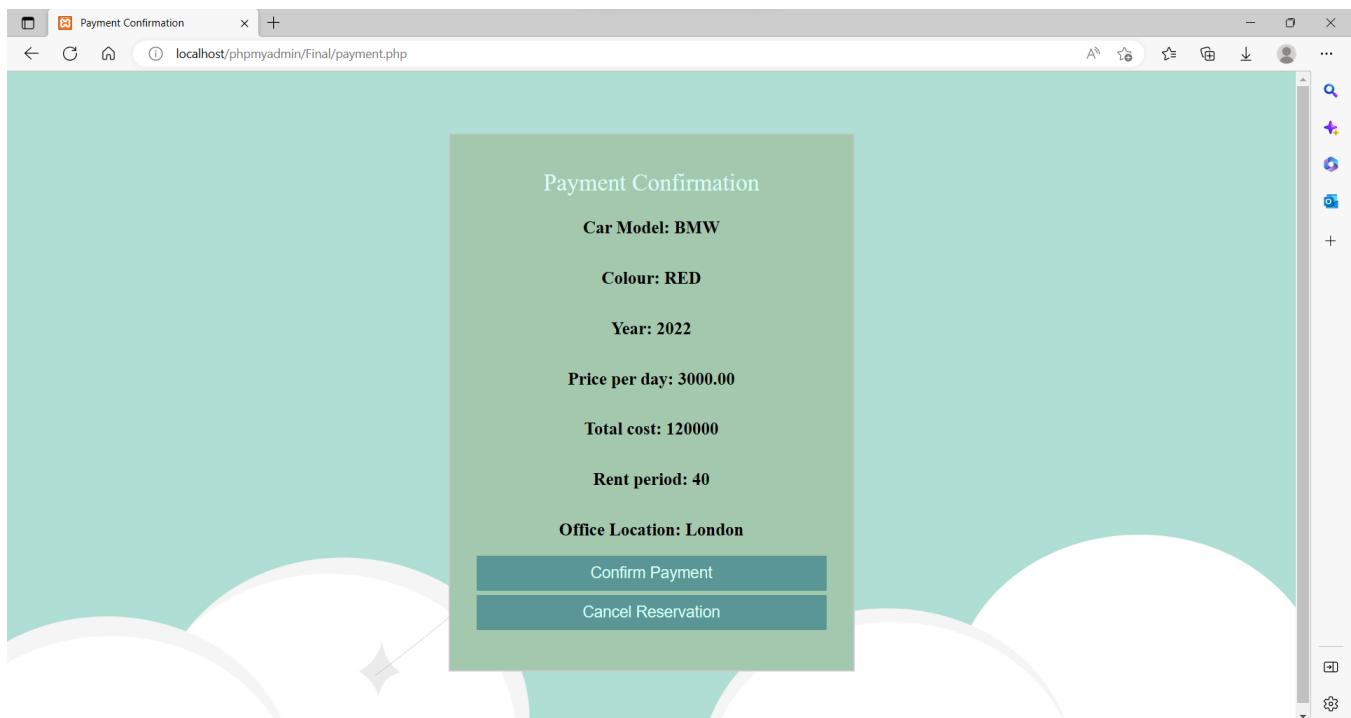


FIGURE 24: PAYMENT CONFIRMATION PAGE

The screenshot shows a web browser window titled "Customer Page" with the URL "localhost/phpmyadmin/Final/customer\_start.php". The main content is a confirmation message: "Welcome Adham ! Your reservation is confirmed!". Below this, there's a section titled "Your Reservations" containing a table:

Plate ID	PickUp Date	Return Date	Cost	Period	PickedUp?	Returned?
5YZ3F2	2022-01-01	2022-02-09	120000.00	40	n	n
H3F2N7	2022-03-02	2022-03-05	8000.00	4	y	y
H3F2N7	2022-03-06	2022-03-10	10000.00	5	y	y

To the right of the reservations table is a search interface with several input fields and sections:

- Search for a car**: "Enter the required period" with "Pickup Date (YYYY-MM-DD)" and "Return Date (YYYY-MM-DD)" fields.
- Enter car specifications**: "Office Location...", "Model...", "Year...", "Color..." input fields.
- Enter the price range**: "Minimum Price..." and "Maximum Price..." input fields.

At the bottom right are "Search" and "Back to Login" buttons.

FIGURE 25: CUSTOMER HOME PAGE AFTER A CONFIRMED RESERVATION