

CAR RENTAL_PART 2



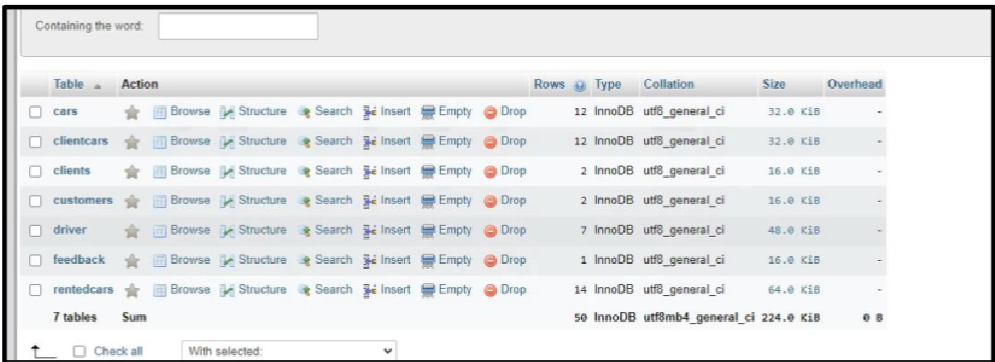
CAR RENTAL SYSTEM

Chapter 4: Result and Implementation

4.1 Introduction

The researcher discusses the outcomes of the directed investigation and provides a top-to-bottom analysis of the findings in this chapter. The implementation of the suggested structure in light of the study objectives outlined in Chapter 1 is the chapter's main focus. The researcher intends to evaluate the suitability and believability of the suggested technique by looking into the findings and going over their recommendations. The study methodology used to obtain and analyze data is featured at the beginning of the chapter. The researcher emphasizes the use of a mixed strategy approach that comprised both objective and quantitative techniques to obtain comprehensive bits of knowledge. Utilizing factual devices and theme analysis techniques, the collected data underwent a detailed investigation. The researcher maintains an impartial and objective tone throughout the entire chapter, providing the findings and implementation specifics in a condensed and understandable manner. The findings and their recommendations serve as the impetus for the succeeding chapters, study further discussions, conclusions, and recommendations in great detail.

4.2 Implementation



The screenshot shows a database management interface with a search bar at the top. Below it is a table listing database tables. The table has columns for Table, Action, Rows, Type, Collation, Size, and Overhead. The tables listed are cars, clientcars, clients, customers, driver, feedback, and rentedcars. Each table has a set of icons for actions like Browse, Structure, Search, Insert, Empty, and Drop. At the bottom, there is a summary row for 7 tables, showing a total of 50 rows, InnoDB engine, utf8mb4_general_ci collation, and a total size of 224.0 KiB with 0.0 KiB overhead.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> cars		12	InnoDB	utf8_general_ci	32.0 KiB	-
<input type="checkbox"/> clientcars		12	InnoDB	utf8_general_ci	32.0 KiB	-
<input type="checkbox"/> clients		2	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> customers		2	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> driver		7	InnoDB	utf8_general_ci	48.0 KiB	-
<input type="checkbox"/> feedback		1	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> rentedcars		14	InnoDB	utf8_general_ci	64.0 KiB	-
7 tables	Sum	50	InnoDB	utf8mb4_general_ci	224.0 KiB	0.0

Figure 4.2 1: Database Creation

(Source: Obtained from Xampp)

The researcher has created the database using different variables such as “cars, Customers, driver, feedback, rented cars” and uploaded the necessary information. The researcher takes numerous steps during the "Database Creation" process. The researcher starts the database's formation, specifies its size, set up the client and customer settings, sets up the search features,

and carries out insert and drop operations. A certain amount of storage space is also allotted, and the structure is established using the InnoDB engine.

```
<?php

function Connect()
{
    $dbhost = "localhost:3307";
    $dbuser = "root";
    $dbpass = "";
    $dbname = "carrental";

    //Create Connection
    $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname) or die($conn->connect_error);

    return $conn;
}
?>
```

Figure 4.2.2: Connect the Frontend To the database

(Source: Obtained from VS Code)

In the above code image, the researcher completes the following actions, A connection to a MySQL database is made possible by the researcher's definition of the PHP function "Connect" in this code. Variables like "\$dbhost" (the hostname and port of the database server), "\$dbuser" (the username for accessing the database), "\$dbpass" (the password for the database user), and "\$dbnaae" (the database name) are set to certain values by the researcher. By supplying the values of the database host, username, password, and database name as inputs to the "mysqli" function, the researcher creates a new connection object "\$conn". When a connection is established successfully, the researcher returns the connection object "\$conn". The researcher uses the "die" function to stop the script from running if a connection error happens, and then they display an error message using the data they've pulled from "\$conn- y-connect_error" in it. The researcher's overall goal with this snippet of code is to connect the PHP script to the MySQL database, making sure that the required credentials are supplied and resolving any potential errors that may arise during the connection process.

```

session_client.php
1  <?php
2  // mysqli_connect() function opens a new connection to the MySQL server.
3  require 'connection.php';
4  $conn = Connect();
5
6  session_start();// Starting Session
7
8  // Storing Session
9  $user_check=$_SESSION['login_client'];
10
11 // SQL Query To Fetch Complete Information Of User
12 $query = "SELECT client_username FROM clients WHERE client_username = '$user_check'";
13 $ses_sql = mysqli_query($conn, $query);
14 $row = mysqli_fetch_assoc($ses_sql);
15 $login_session =$row['client_username'];
16 ?>

```

Figure 4.2 3: Session connection on the database

(Source: Obtained from VS Code)

The researcher has established the connection to start the session which is necessary to store details of clients who are logged in. The researcher runs the following commands in the provided snippet of code. They entail including the necessary connection file, connecting to the SQL server using the `mysqli_connect()` function, starting a session, storing the session data, and running a SQL query to retrieve all of the user's information from the "clients" table based on the supplied username.

```

session_start(); // Starting Session
$error=''; // Variable To Store Error Message

if (isset($_POST['submit'])) {
    if (empty($_POST['customer_username']) || empty($_POST['customer_password'])) {
        $error = "Username or Password is invalid";
    }
    else
    {
        // Define $username and $password
        $customer_username=$_POST['customer_username'];
        $customer_password=$_POST['customer_password'];
        // Establishing Connection with Server by passing server_name, user_id and password as a parameter
        require 'connection.php';
        $conn = Connect();

        // SQL query to fetch information of registered users and finds user match.
        $query = "SELECT customer_username, customer_password FROM customers WHERE customer_username=? AND customer_pa

        // To protect MySQL injection for Security purpose
        $stmt = $conn->prepare($query);
        $stmt -> bind_param("ss", $customer_username, $customer_password);
        $stmt -> execute();
        $stmt -> bind_result($customer_username, $customer_password);
        $stmt -> store_result();

        if ($stmt->fetch()) //fetching the contents of the row
        {
            $_SESSION['login_customer']=$customer_username; // Initializing Session
            header("location: index.php"); // Redirecting To Other Page
        } else {
            $error = "Username or Password is invalid";
        }
    }
}

```

**Figure 4.2.4: Connection of Log in
(Customer)**

(Source: Obtained from VS Code)

The researcher runs the following code snippet to complete the tasks. They begin a session, initialize an empty variable to store error messages, perform input validation to see if the user's username and password are provided, define variables to hold the user name and password values, establish a connection with the server by including the connection file, execute a SQL query to fetch information of registered users that match the user name and password provided, protect against SQL injection using prepared statements, fetch the user's username and password values, and finally, they fetch the user's username and password values.

4.3 Result

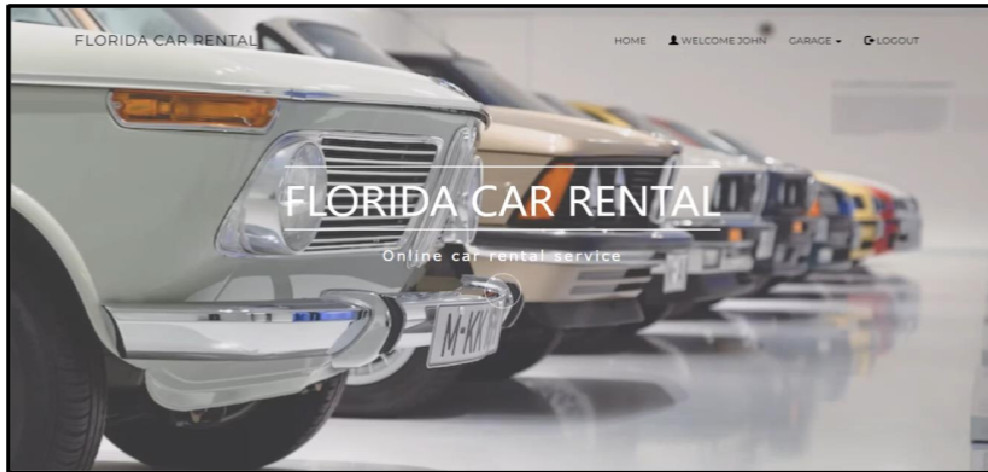


Figure 4.3.1: Landing Page

(Source: Self-created)

The developer carries out the following tasks on the "Landing Page" of Florida Car Rental. The Developer uses HTML and CSS to design and develop the layout of the landing page, add appealing visuals and branding components, optimize the page for responsive design, guarantee easy navigation and a positive user experience, incorporate pertinent content and promotional offers, and test the page's functionality and browser compatibility.

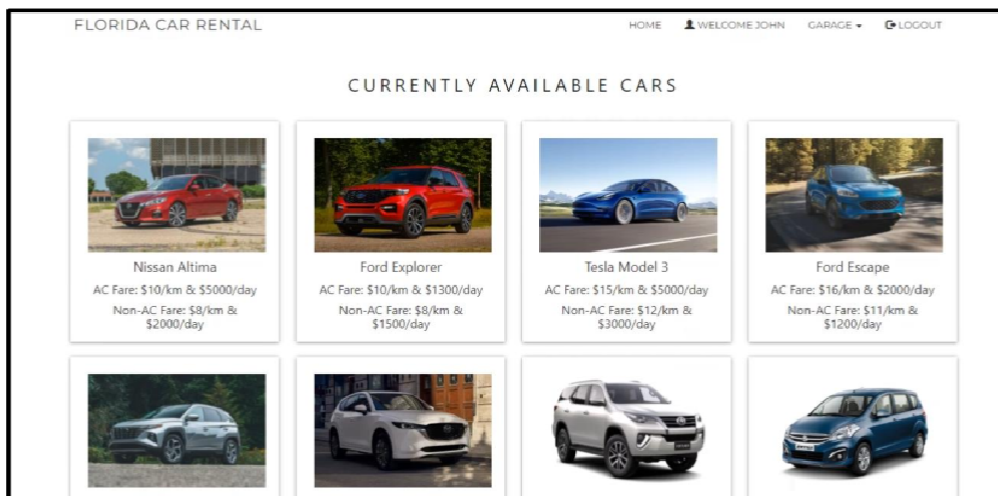
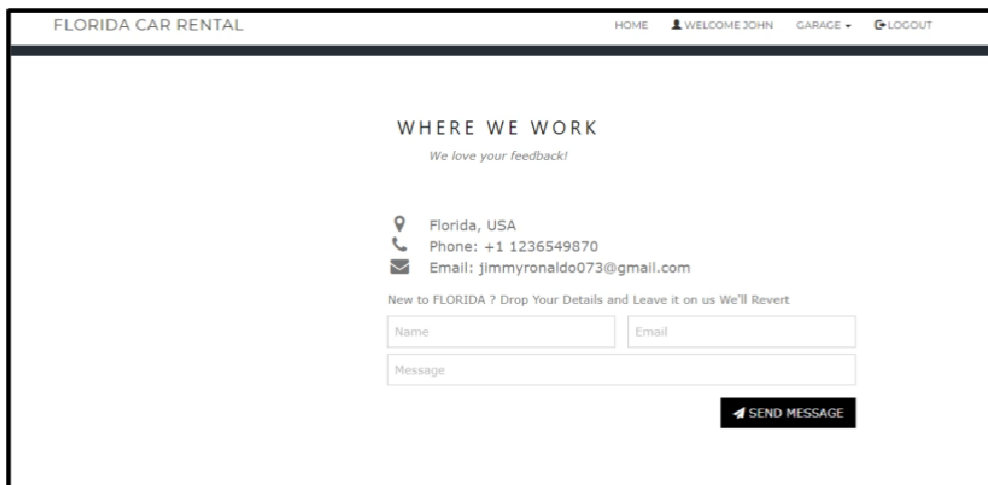


Figure 4.3.2: Currently Available Cars Page

(Source: Self-created)

The developer works on the "Currently Available Cars Page" of the Florida Car Rental website. They develop a dynamic website that pulls information about cars from the database, displays car models along with their associated prices, and groups them according to availability and amenities like AC or non-AC. The webpage's aesthetic appeal and user-friendliness are ensured by the web designer, who also ensures that it provides prospective customers with correct and up-to-date information.



FLORIDA CAR RENTAL

HOME WELCOME JOHN GARAGE LOGOUT

WHERE WE WORK

We love your feedback!

📍 Florida, USA
📞 Phone: +1 1236549870
✉ Email: jimmyronaldo073@gmail.com

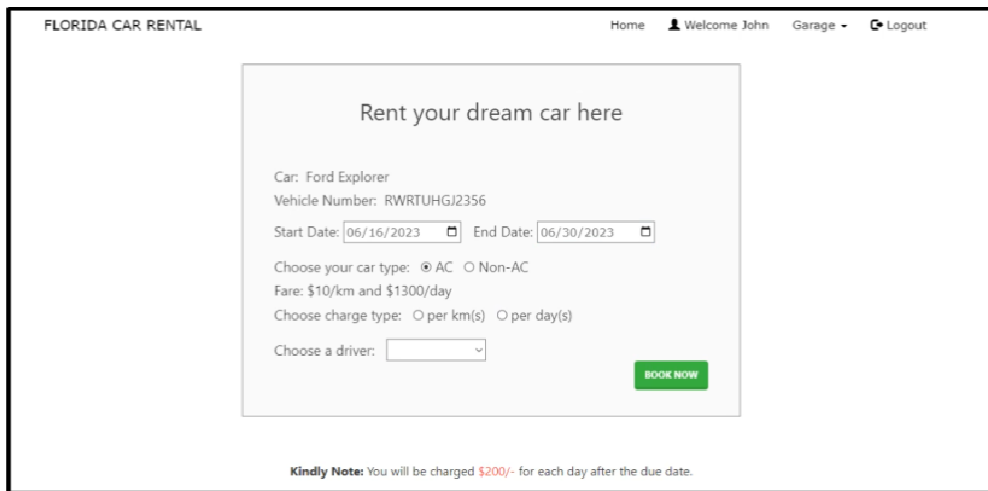
New to FLORIDA ? Drop Your Details and Leave it on us We'll Revert

[SEND MESSAGE](#)

Figure 4.3.3: A page for Information about the company

(Source: Self-created)

The following duties are finished by the developer on the "Information about the company" part of the Florida Car Rental website. They make a page with a list of all the company's locations, including Florida, and pertinent contact information. The page's author ensures sure a space where users can express their ideas is present. The developer offers a form on which customers may enter their information and request assistance, and they guarantee a speedy response.



FLORIDA CAR RENTAL

Home Welcome John Garage Logout

Rent your dream car here

Car: Ford Explorer
Vehicle Number: RWRUHGJ2356

Start Date: 06/16/2023 End Date: 06/30/2023

Choose your car type: ☒ AC ☐ Non-AC
Fare: \$10/km and \$1300/day

Choose charge type: ☐ per km(s) ☒ per day(s)

Choose a driver:

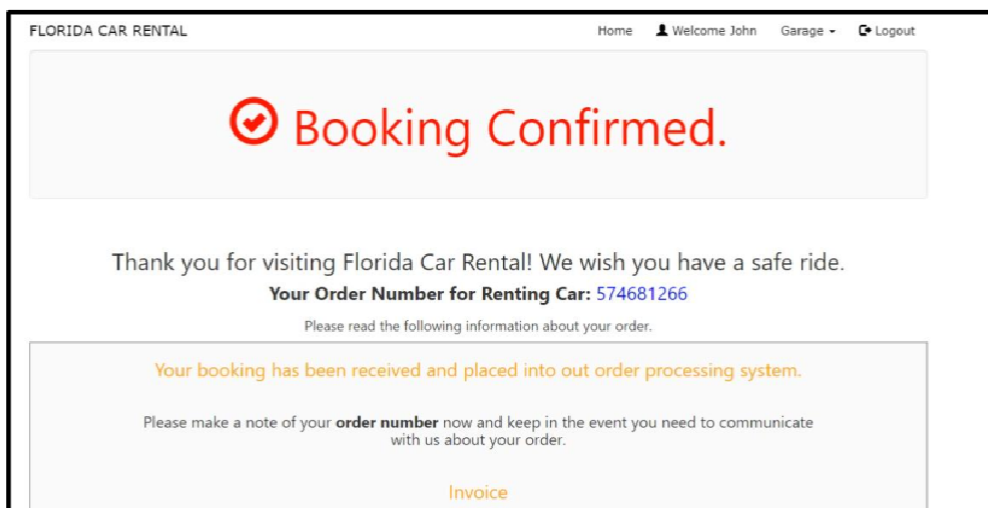
BOOK NOW

Kindly Note: You will be charged \$200/- for each day after the due date.

Figure 4.3.4: This Page for Rent a Car

(Source: Self-created)

The "Rent a Car" segment of the Florida Car Rental site is taken care of by the designer. The maker gives a user-friendly interface that produces it basic for guests to sign up and rent the leading car. The creator provides details with respect to a particular car, such as the Ford Explorer, in conjunction with the vehicle's permit plate. Users can choose between AC and non-AC automobiles as well as the fare and charge type in addition. Additionally, the developer refers to the availability of a driver and warns clients about late return fines.



FLORIDA CAR RENTAL

Home Welcome John Garage Logout

📍 Booking Confirmed.

Thank you for visiting Florida Car Rental! We wish you have a safe ride.

Your Order Number for Renting Car: 574681266

Please read the following information about your order.

Your booking has been received and placed into out order processing system.

Please make a note of your **order number** now and keep in the event you need to communicate with us about your order.

[Invoice](#)

Figure 4.3.5: Booking Confirmation

(Source: Self-created)

The developer posts a note on the website thanking the visitor for coming and wishing them a safe journey for the Florida Car Rental website. For the car rental, the developer creates a special order number and gives details about the booking's receipt and entry into the order processing system. Additionally, they advise the user to remember the order for future correspondence and emphasize that an invoice is available.

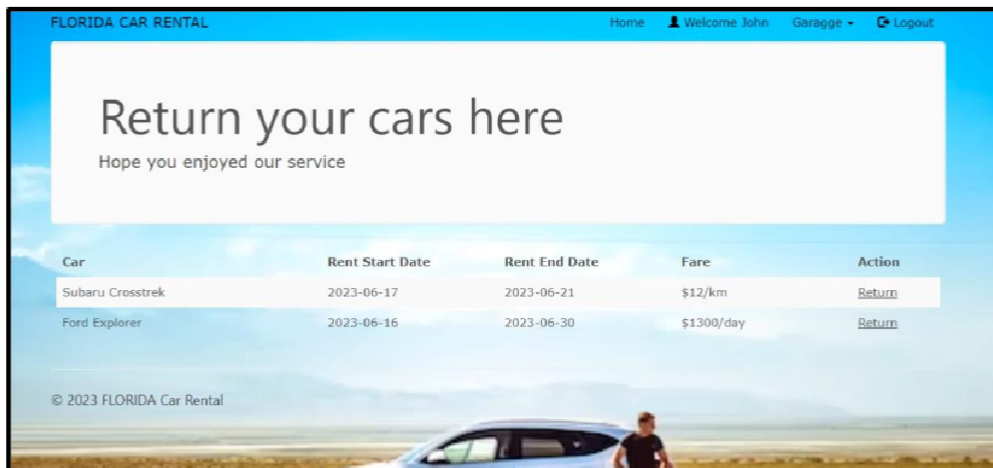


Figure 4.3.6: Car return Page

(Source: Self-created)

The developer performs the following things on the "Return Car" page of the Florida Car Rental website. They ask the user to return their autos to a certain area by displaying a message. The creator wishes the user had a pleasant experience using their service. For the start date, finish date, and fare information, they offer entry fields. Additionally, they indicate that you can rent a Ford Crosstree vehicle.

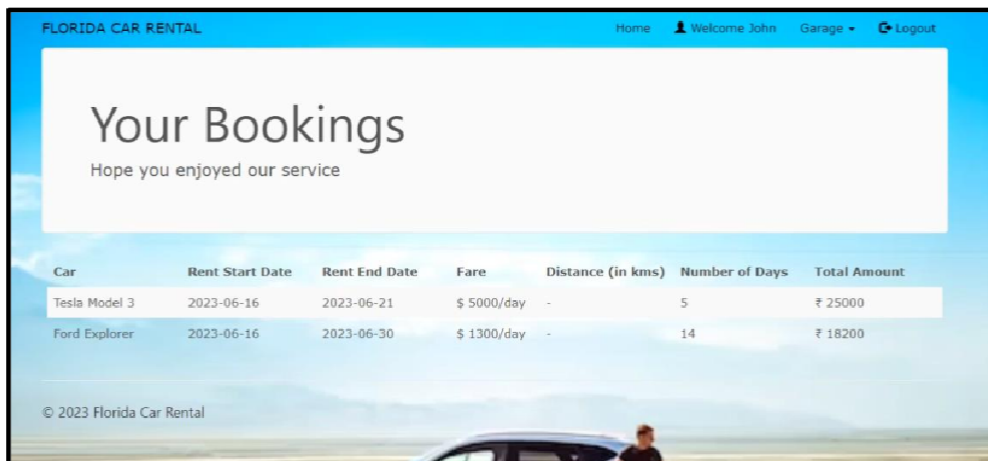


Figure 4.3.7: Booking Section of the website

(Source: Self-created)

The developer provides an overview of the customer's bookings in the "Your Bookings" area. They pull the booking information from the database and present details like the type of rental car (Tesla Model 3 or Ford Explorer), the start and finish dates of the rental, the number of days rented, the daily rate, the number of kilometers driven, and the total price charged.

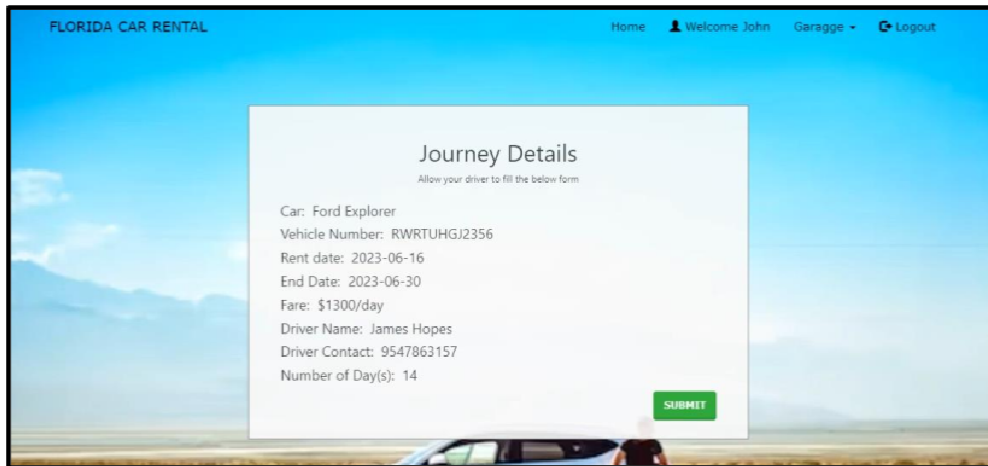
The screenshot shows a web application for "FLORIDA CAR RENTAL". The top navigation bar includes links for "Home", "Welcome John", "Garage", and "Logout". The main content area features a "Journey Details" form with the subtitle "Allow your driver to fill the below form". The form contains the following information: Car: Ford Explorer, Vehicle Number: RWRTUHGJ2356, Rent date: 2023-06-16, End Date: 2023-06-30, Fare: \$1300/day, Driver Name: James Hopes, Driver Contact: 9547863157, and Number of Day(s): 14. A green "SUBMIT" button is located at the bottom right of the form. The background of the page shows a car parked in a desert landscape under a blue sky.

Figure 4.3.8: Journey Details

(Source: Self-created)

The journey's precise information is retrieved by the developer. The model of the rented car (Ford Explorer), the vehicle number, the start and finish dates of the rental, the daily rate, the name of the designated driver (James Hopes), the driver's phone number, and the total number of rental days (14 days) are all displayed.

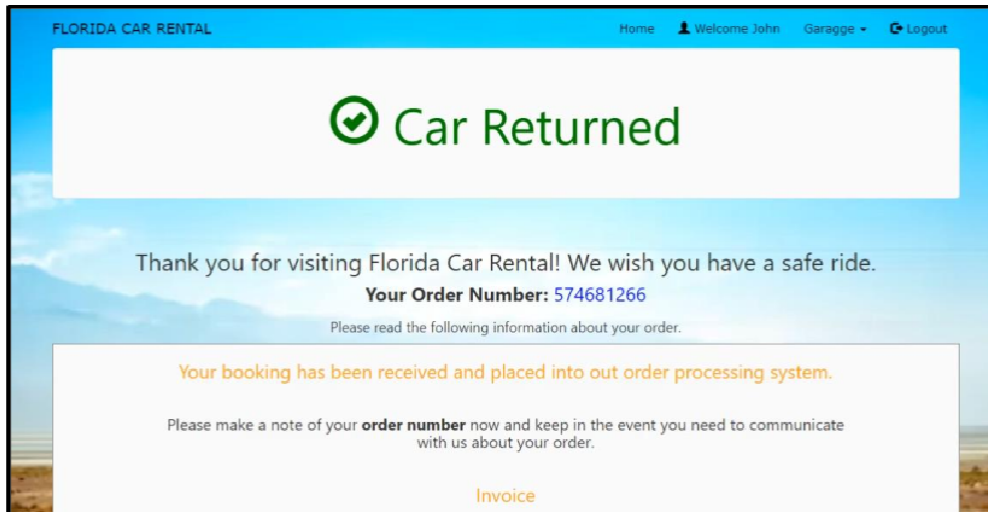


Figure 4.3.9: Car return of the page

(Source: Self-created)

To confirm the rental car's return, the developer generates a confirmation message. A thank-you message is displayed along with a reference to the order number (574681266). As soon as the booking is accepted and completed, the developer additionally suggests that the consumer examine the details they have already submitted. They also provide a link to check the invoice and stress how crucial it is to keep track of the order information for any upcoming correspondence over it.

Chapter 5: Conclusion and Recommendation

The conclusion and recommendation section provides a summary of the study conducted in the earlier sections, highlighting the significant findings and nuggets of information gained from nurturing a Basic Car Rental System using MySQL, HTML, and CSS. This part offers suggestions for further advancements and lines of inquiry for future study.

5.1 Conclusion

Using MySQL, HTML, and CSS, the research conducted for this paper aimed to develop a simple Car Rental System for academic use. The research system used positivism as the basis for its logical research strategy and descriptive research plan. Optional subjective data collection strategies were employed, such as a comprehensive writing assessment of excellent scholarly writing, business reports, and web resources. The purpose of the data analysis method was to rate the effectiveness and efficiency of the developed automobile rental system. The findings of this study include a few important points of view. First and foremost, the creation of the Car Rental System was accomplished using MySQL, HTML, and CSS, resulting in a useful and user-friendly system. The front-end development phase took into account the creation of an entertaining Because a user-friendly point of engagement was planned using HTML wireframes and CSS styles, the visual experience was enhanced. The addition of JavaScript enhanced user experience by empowering intelligence.

The necessary conditions were set up during the back-end development phase, including the use of XAMPP as a web server environment. The executives' system for managing appointments, accessibility, customer information, and other crucial data used MySQL as an open-source social database. To manage HTTP requests, connect to the MySQL database, and run the Car Rental System's business logic, PHP prearranging was used. Together, these tools and technology made it possible to create a comprehensive and practical system. The investigation also covered the drawbacks of the established car rental system. The approach may not apply to real car rental companies because of its scholarly nature, which has additional complicated criteria and functional considerations. The scope of this study did not include sophisticated features such as coordinated installation doors, complex booking computations, and broad executive capabilities. Additionally, by focusing on MySQL, HTML, and CSS technologies, the system's adaptability and flexibility may be limited to newer frameworks or technologies that can provide more advanced user experiences or abilities.

5.2 Recommendations

A few suggestions for further advancements and areas for future research can be offered in light of the findings and limitations of this study:

Integration of Advanced Features:

It is advised to combine innovative features like integrated installation doors, sophisticated booking calculations, and thorough throughout-the-board abilities to increase the utility and handiness of the car rental system. With these increases, the system would be adjusted even more closely to the needs of real automobile rental companies.

Exploration of New Technologies and Frameworks:

MySQL, HTML, and CSS were successfully employed in this study, it is advised to look into newer technologies and frameworks to improve the system's adaptability and flexibility. Front-end frameworks like Respond or Precise, as well as other emerging technologies, can provide improved user experiences and capabilities by taking into account the use of diverse databases.

User Testing and Validation:

To ensure the system's usability, effectiveness, and viability, it is crucial to oversee widespread user testing and validation. Testing with a larger sample size and involving various user groups can provide useful insights for further improvements and amplifications.

Security Enhancements:

Given the importance of data security in automobile rental systems, it is advised to implement strict security procedures to protect user data from unauthorized access, breaks, and malicious activities. This includes ensuring compliance with data assurance laws, using encryption techniques, and engaging in secure coding exercises.

Accessibility Considerations:

Planning the car rental system because of accessibility can make it usable and open to individuals with handicaps. Keeping accessibility rules and standards can work on the inclusivity and ease of use of the system.

Performance Optimization:

For the system to deliver a consistent user experience, regular monitoring and performance optimization are urgently needed. Performance bottlenecks can be located and fixed with the aid of techniques like reserving, code optimization, and load testing.

Comparative Analysis:

Leading a comparative analysis of various vehicle rental system implementations, technologies, and frameworks can provide a more comprehensive perspective on the benefits

and drawbacks of alternative techniques. In making decisions for system development or enhancement in the future, this can be helpful.

In conclusion, the creation of a car rental system using HTML, CSS, and MySQL demonstrated the appropriate application of web development technology. The technology used in this study served as a foundation for a practical and approachable connection point, demonstrating the possibility for further improvements. The suggested changes and research in the area of automobile rental systems are greatly aided by the recommendations made, which focus on overcoming obstacles and examining novel features and technologies to satisfy the expanding needs of the industry.