

Permbajtja Dinamike e Web Inxhinieringut

Tema: Online Rent a Car

Profesor:

Prof.Ass.Dr Zirije Hasani Punuan:

Ardit Berisha: 200307054

Bleon Berisha: 200307034

Eren Pllava: 200307081

Contents

[1. Introduction 3](#_Toc136225122)

[Project Architecture 4](#_Toc136225123)

[Technologies Used 5](#_Toc136225124)

[Files Structure 6](#_Toc136225125)

[Functionality 7](#_Toc136225126)

[Key Features and Functionality 8](#_Toc136225127)

[Known issues, bugs and limitations 11](#_Toc136225128)

# 1. Introduction

The project aims to develop a web-based car rental management system called "DriveNow Rent a Car." The purpose of this system is to provide an efficient and user-friendly platform for customers to rent cars and for administrators to manage the rental process.

The primary problem this project addresses is the need for a convenient and streamlined car rental process. Traditional car rental systems often involve lengthy paperwork, manual booking procedures, and limited availability of car options. DriveNow Rent a Car seeks to overcome these challenges by offering an online platform where customers can easily browse available cars, make reservations, and manage their bookings.

The motivation behind this project stems from the increasing demand for car rentals and the desire to enhance the overall customer experience. By providing an intuitive and efficient system, customers can enjoy a hassle-free car rental experience, while rental agencies can streamline their operations and maximize their revenue.

The key goals of the project include:

Developing a user-friendly website for customers to search and book rental cars.

Implementing a secure and reliable database to store car and customer information.

Creating an intuitive administration panel for staff members to manage bookings, inventory, and customer requests.

Integrating payment processing functionality for seamless online transactions.

Ensuring the system is scalable and can accommodate future growth and additional features.

Through this project, we aim to revolutionize the car rental industry by providing a modern and efficient solution that simplifies the rental process for both customers and administrators.

# Project Architecture

The architecture of the "DriveNow Rent a Car" project follows a layered approach, consisting of various components and modules that work together to achieve the system's functionality. The architecture can be divided into three main layers: The Presentation Layer, the Business Logic Layer, and the Data Layer.

Presentation Layer:

This layer handles the user interface and interaction with the system.

It includes the web pages, forms, and visual elements that users interact with when browsing the website and making car reservations.

The presentation layer communicates with the business logic layer to retrieve and display data to the users.

Examples of components in this layer include HTML/CSS templates for client-side interactivity, and PHP scripts for server-side processing.

Business Logic Layer:

This layer contains the core logic and functionality of the car rental system.

It processes user requests, performs validations, and handles business rules.

It interacts with the data layer to retrieve and store data.

Examples of components in this layer include PHP scripts that handle user authentication, car booking etc.

Data Layer:

This layer is responsible for managing the system's data storage and retrieval.

It includes the database where information such as car details, customer records, and booking history is stored.

The data layer interacts with the business logic layer to provide the necessary data for processing user requests.

Examples of components in this layer include a relational database management system (e.g., MySQL) and SQL queries for data manipulation.

# Technologies Used

The "DriveNow Rent a Car" project utilizes a combination of technologies, and tools to create a robust and functional car rental system. Here is a list of the key technologies used in the project:

HTML5 and CSS3:

HTML5 is used for structuring the web pages and defining their content.

CSS3 is used for styling the web pages and enhancing the user interface.

PHP:

PHP is used as the server-side scripting language for dynamic web page generation and backend processing.

It handles form submissions, interacts with the database, and performs business logic operations.

MySQL:

MySQL is used as the relational database management system to store and manage data.

It stores information such as car details, customer records, reservations, and booking history.

SQL queries are used to interact with the database and retrieve or manipulate data.

XAMPP (Version 3.3.0):

XAMPP is a cross-platform software package that includes Apache web server, MySQL database, PHP, and Perl.

It provides a local development environment for running the project on the local machine during the development phase

Code editor:

Visual Studio Code

Web browser:

Google Chrome, Mozilla Firefox:

Web browsers are used to view and test the web pages of the project.

Compatibility and responsiveness of the website are checked using different browsers.

# Files Structure

The project follows a specific file structure to organize the various files and directories. Here is an overview of the file structure:

**Root Folder:**

This is the main folder that contains all the project files and directories.

It is typically named after the project or application.

**index.php:**

This file serves as the entry point of the web application.

It handles the initial routing and loading of the necessary dependencies.

CSS Folder:

This folder contains CSS files used for styling the web pages.

It includes files such as "style.css" for general styling and "contact.css" for the contact page styling.

**Pics Folder:**

This folder contains images used throughout the website.

It includes subfolders for organizing different types of images, such as "car\_pictures" for car images.

PHP Files:

**The root folder** may contain various PHP files that handle different functionalities of the project.

These files include "header.php" and "footer.php" for header and footer sections that are included in multiple pages.

**Admin Folder (Located inside the root folder):**

This folder contains files related to the administration panel of the car rental system.

It includes PHP files for managing cars, reservations, and other administrative tasks.

## Functionality

The project aims to provide the following key functionalities:

* Car Listing and Details
* Users can view a list of available cars for rent.
* Each car has detailed information such as its make, model and price.

**Car Reservation:**

Users can select a car and make a reservation by specifying the pickup and return dates.

The system checks if we have the car and confirms the reservation if it is available.

**User Registration and Authentication:**

Users can create an account and log in to access additional features.

**Contact Form:**

A contact form is available for users to send inquiries or messages to the website administrators.

Submitted messages are stored in the database for further processing.

Administration Panel:

**Admin panel:**

Allows authorized users to manage cars, reservations, and other system settings.

Administrative functionalities include adding new cars, updating prices, and approving reservations.

These functionalities provide users with the ability to browse and rent cars, while the administration panel enables system administrators to manage the car inventory and monitor reservations

# Key Features and Functionality

**Car Listing and Details:**

Users can view a list of available cars for rent.

Each car is displayed with relevant information such as make, model, price, and availability.

Users can click on a car to view detailed information and images.

Car Search and Filtering:

**Car Reservation:**

Users can select a car and make a reservation by specifying the pickup and return dates.

The system checks the availability of the selected car during the specified dates.

If the car is available, the user can proceed with the reservation and provide necessary details.

**User Registration and Authentication:**

Users can create an account and log in to access additional features.

Registration requires providing personal information and creating a unique username and password.

Authentication ensures that only registered users can access certain functionalities.

**Contact Form:**

A contact form is available for users to send inquiries or messages to the website administrators.

Users can provide their name, email address, and message.

The submitted messages are stored in the database and can be accessed by administrators.

**Administration Panel:**

An admin panel allows authorized users to manage cars, reservations, and other system settings.

Administrators can add new cars, update car information, and delete cars from the system.

They can view and manage reservations, approve or reject bookings, and generate reports.

Integration of Project Components

The different parts of the project work together to achieve the desired functionality. Here's how they integrate:

**User Interface (HTML, CSS):**

The user interface presents the website's layout, design, and interactivity.

HTML markup structures the content, CSS styles provide visual presentation, and multiple forms handle client-side interactivity.

**Backend (PHP):**

PHP scripts handle server-side processing, database operations, and dynamic content generation.

It interacts with the database to retrieve or store data, validate user inputs, and execute business logic.

**Database (MySQL):**

The database stores and manages the project's data, such as car information, user details, and reservations.

It provides a structured storage system with tables, columns, and relationships.

PHP scripts communicate with the database using SQL queries to retrieve or update data.

**Admin Panel:**

The admin panel is a separate part of the project that allows administrators to manage the system.

It has its own set of PHP scripts and database tables to handle administrative functionalities.

The admin panel integrates with the main project components to access and modify data.

Code snippet for how the admin can edit a car in the database:  
 <div class="main">

                <?php

                // Check if car ID is provided in the query string

                if (isset($\_GET['car\_id'])) {

                    $carId = $\_GET['car\_id'];

                    // Retrieve the car data from the database based on car ID

                    include "../connect.php";

                    $sql = "SELECT \* FROM cars WHERE id = $carId";

                    $result = mysqli\_query($conn, $sql);

                    if (mysqli\_num\_rows($result) > 0) {

                        $car = mysqli\_fetch\_assoc($result);

                        ?>

                        <form action="update\_car.php" method="POST" enctype="multipart/form-data">

                            <input type="hidden" name="car\_id" value="<?php echo $car['id']; ?>">

                            <label for="make">Make:</label>

                            <input type="text" name="make" id="make" value="<?php echo $car['make']; ?>"><br><br>

                            <label for="model">Model:</label>

                            <input type="text" name="model" id="model" value="<?php echo $car['model']; ?>"><br><br>

                            <label for="year">Year:</label>

                            <input type="number" name="year" id="year" value="<?php echo $car['year']; ?>"><br><br>

                            <label for="color">Color:</label>

                            <input type="text" name="color" id="color" value="<?php echo $car['color']; ?>"><br><br>

                            <label for="rental\_price">Rental Price:</label>

                            <input type="number" name="rental\_price" id="rental\_price"

                                value="<?php echo $car['rental\_price']; ?>"><br><br>

                            <label for="image">Car Image:</label>

                            <?php if (!empty($car['image\_path'])): ?>

                                <img src="../car\_images/Car Pictures/<?php echo $car['image\_path']; ?>" alt="Car Image"><br><br>

                                <input type="checkbox" name="remove\_image" id="remove\_image">

                                <label for="remove\_image">Remove Picture</label><br><br>

                            <?php else: ?>

                                <p>No image available.</p>

                            <?php endif; ?>

                            <input type="file" name="image" id="image"><br><br>

                            <input type="submit" value="Update">

                        </form>

                        <?php

                    } else {

                        echo '<p>Car not found.</p>';

                    }

                    // Close the database connection

                    mysqli\_close($conn);

                } else {

                    echo '<p>Invalid car ID.</p>';

                }

                ?>

# Known issues, bugs and limitations

**Email Sending:** Currently, the email sending functionality is not working due to server configuration issues or missing SMTP settings. As a result, users may not receive email notifications for their reservations or contact form submissions. To address this issue, you can consider implementing a third-party email service or consult with your hosting provider to resolve the email sending problem.

**Security Vulnerabilities:** The project may have potential security vulnerabilities, such as SQL injection or cross-site scripting (XSS) attacks. It is important to conduct a thorough security audit and implement proper input validation, parameterized queries, and data sanitization to prevent these vulnerabilities. Regularly updating your project's dependencies and frameworks is also crucial to address any security patches or bug fixes.

**Responsive Design:** The current project may lack responsive design, making it less user-friendly on different screen sizes and devices. It is recommended to implement responsive design techniques using CSS media queries and flexible layouts to ensure optimal viewing and usability across various devices.

**Error Handling and Validation:** The project may have limited error handling and validation in certain areas, such as user input validation, form submissions, and database operations. Enhance the error handling by implementing proper validation checks, error messages, and logging mechanisms to provide a more user-friendly and robust system.

**Performance Optimization:** Depending on the scale of the project and the number of concurrent users, there might be potential performance issues. Common areas that can be optimized include database indexing, query optimization, caching mechanisms, and server-side optimizations. Conduct performance testing and profiling to identify bottlenecks and implement appropriate optimizations.