Birat Multiple College

Tribhuvan University

Institute of Science and Technology



**A Project Report On**

###### Online Car Rental System

**Submitted to**

Department of Computer Science and Information Technology

Birat Multiple College

#### Submitted by

Avinash Kumar Sah (11495/073)

Khomraj Singh Urawn (11499/073)

Rabin Gautam (11501/073)

Shrijan Shrestha (11509/073)

April, 2021

**Birat Multiple College**

Tribhuvan University

# Supervisor’s Recommendation

I hereby recommend that this project prepared under my supervision by Avinash Kumar Sah, Khomraj Singh Urawn, Rabin Gautam and Shrijan Shrestha entitled “Online Car Rental System” in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

………………………

SUPERVISOR

**Birat Multiple College**

Tribhuvan University

# LETTER OF APPROVAL

This is to certify that this project prepared by Avinash Kumar Sah, Khomraj Singh Urawn, Rabin Gautam and Shrijan Shrestha entitled “Online Car Rental System” in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| Signature of the Supervisor  ………………………  ……………………… | Signature of the HOD/Coordinator  ………………………  ………………………  Birat Multiple College, Biratnagar |
| Signature of the External Examiner  ………………………  ……………………… | Signature of the Internal Examiner  ………………………  ……………………… |

# Abstract

The world has become smaller through the use of internet and its technologies. In today's world everything is in our fingertips. We can access anything in the world through the internet. Through the use of internet and its technologies any height can be reached and any standard can be maintained. One of the major results of internet is an online car rental system. The project aimed is to book a vehicle online for rent.

Online Car Rental System is aimed to book a vehicle for rent from different places of Nepal. The online car rental system also includes the idea of which vehicles brand are available at what price per day? This system lets you know detail through scroll on pages. One shouldn't have to go manually to branch office to book a car for rent.

The main idea of this system is to go beyond "Geographic barriers". This system is an innovation that helps to bring effective outcomes on vehicles booking through the use of technology.

# ACKNOWLEDGEMENT

At first, we would like to express our sincere thanks and gratitude to our advisor Mr. Ram Babu Chaudhary for his excellent guidance and motivation for doing our project of Online Car Rental System. His kind encouragement, compassionate and invaluable detailed suggestion and supervision throughout the phases of project development, presentation, and paper organization have helped us to successfully accomplish the paper. Secondly, we would like to express our gratitude to Mr. Amit Ghimire for helping us understanding the basic tools and use of programming languages and other components for developing the project and guided us to achieve our desired goal. Finally, we are sincerely grateful to our parents, family, and friends, who give their valuable suggestions and support to complete the project. The whole research work would not be possible without the active encouragement from all of them.

# Report Organization

* + - In chapter 1, why system is built and condition of Online Car Rental System which are not reliable are introduced. And also, objectives and features of the project are explained in detail with scope and limitation of the system.
    - In chapter 2, discussion about the existing system along with requirement as well as feasibility analysis of the system. The data modeling and process modeling technique are used to give the information about the system requirement.
    - In chapter 3, discussion about the system design. The system design can be database schema design, interface design and process design.
    - In chapter 4, discussion is made on which tools are used on our project to make it possible. The testing is also explained in this part with detailed tabular input and output.
    - In chapter 5, the maintenance and support implemented on the project are discussed.
    - In chapter 6, the conclusion and future enhancements of the project are discussed.

**TABLE OF CONTENTS**

[Supervisor’s Recommendation i](#_Toc69667000)

[LETTER OF APPROVAL ii](#_Toc69667001)

[Abstract iii](#_Toc69667002)

[ACKNOWLEDGEMENT iv](#_Toc69667003)

[Report Organization 1](#_Toc69667004)

[Introduction 5](#_Toc69667005)

[1.1 Introduction of the Project 5](#_Toc69667006)

[1.2 Problem Statement 5](#_Toc69667007)

[1.3 Objective 5](#_Toc69667008)

[1.4 Scope of the Project 6](#_Toc69667009)

[1.5 Limitation of the Project 6](#_Toc69667010)

[1.6 Features of the Project 6](#_Toc69667011)

[Requirement Analysis and Feasibility Study 7](#_Toc69667012)

[2.1 Literature Review 7](#_Toc69667013)

[2.2 Requirement Analysis: 8](#_Toc69667014)

[2.2.1 Functional Requirement 8](#_Toc69667015)

[2.2.2 Non – Functional Requirement 11](#_Toc69667016)

[2.2.3 Hardware Requirement 12](#_Toc69667017)

[2.2.4 Software Requirement 12](#_Toc69667018)

[2.3 Feasibility Study 12](#_Toc69667019)

[2.3.1 Economic Feasibility 12](#_Toc69667020)

[2.3.2 Operational Feasibility 13](#_Toc69667021)

[2.3.3 Technical Feasibility 13](#_Toc69667022)

[2.3.4 Schedule Feasibility 13](#_Toc69667023)

[2.4 Data Modeling 14](#_Toc69667024)

[2.5 Process Modeling 16](#_Toc69667025)

[2.5.1 Data Flow Diagram (DFD) 16](#_Toc69667026)

[System Design 17](#_Toc69667027)

[3.1 System Design 17](#_Toc69667028)

[3.2 System Architecture 17](#_Toc69667029)

[3.3 Interface Design 17](#_Toc69667030)

[3.3.1 User Interface Design 17](#_Toc69667031)

[3.1.1 Admin Panel Design 18](#_Toc69667032)

[3.4 Database Schema Design: 19](#_Toc69667033)

[3.5 Sequence Diagram 20](#_Toc69667035)

[3.6 Class Diagram 22](#_Toc69667036)

[3.7 Algorithm and Flowchart 22](#_Toc69667037)

[Implementation and Testing 24](#_Toc69667038)

[4.1 Implementation 24](#_Toc69667039)

[4.2 Tools Used 24](#_Toc69667040)

[4.2.1 Front-End 24](#_Toc69667041)

[a. HTML/CSS/JavaScript 24](#_Toc69667042)

[b. Bootstrap 25](#_Toc69667043)

[4.2.2 Back-End 25](#_Toc69667044)

[a. PHP 25](#_Toc69667045)

[b. MySQL 26](#_Toc69667046)

[c. XAMPP Server 26](#_Toc69667047)

[4.2.3 Documentation Tools 26](#_Toc69667048)

[4.3 Methodology 27](#_Toc69667049)

[4.4 Testing 27](#_Toc69667050)

[4.4.1 Unit Testing 28](#_Toc69667051)

[4.4.2 Integration Testing 31](#_Toc69667052)

[4.4.3 System Testing 31](#_Toc69667053)

[Maintenance and Support 32](#_Toc69667054)

[5.1 Maintenance 32](#_Toc69667055)

[5.2 Support 32](#_Toc69667056)

[Conclusion and Future Enhancement 33](#_Toc69667057)

[6.1 Conclusion 33](#_Toc69667058)

[6.2 Future Enhancement 33](#_Toc69667059)

[Appendices 34](#_Toc69667060)

[Snapshots: 34](#_Toc69667061)

[Source Code: 39](#_Toc69667062)

**LIST OF FIGURES**

[Figure 1: Use Case Diagram 7](#_Toc69667396)

[Figure 2: Gantt Chart 12](#_Toc69667397)

[Figure 3: E-R Diagram 13](#_Toc69667398)

[Figure 4: Data Flow Diagram 14](#_Toc69667399)

[Figure 5: MVC System Architecture 15](#_Toc69667400)

[Figure 6: Front-End User Interface Design 16](#_Toc69667401)

[Figure 7: Admin Panel User Interface Design 17](#_Toc69667402)

[Figure 8: Schema Diagram 18](#_Toc69667403)

[Figure 9: Sequence Diagram 19](#_Toc69667404)

[Figure 10: Class Diagram 20](#_Toc69667405)

[Figure 11: Flowchart of Online Car Rental 21](#_Toc69667406)

[Figure 12: Iterative Model for System Development 25](#_Toc69667407)

[Figure 13: Testing 29](#_Toc69667408)

**LIST OF TABLES**

[Table 1: Use Case Description for Creating Account 8](#_Toc69667434)

[Table 2: Use Case Description for View Vehicles 9](#_Toc69667435)

[Table 3: Use Case Description for Post Testimonial 9](#_Toc69667436)

[Table 4: Use Case Description for Booking Vehicles 10](#_Toc69667437)

[Table 5: Use Case Description for Manage Testimonial 10](#_Toc69667438)

[Table 6: Use Case Description for Manage Booking 10](#_Toc69667439)

[Table 7: Use Case Description for Manage Brands 10](#_Toc69667440)

[Table 8: Use Case Description for View Users 11](#_Toc69667441)

[Table 9: Use Case Description for View Contacts 11](#_Toc69667442)

[Table 10: Use Case Description for Remove Users 11](#_Toc69667443)

[Table 11: Schedule Feasibility 13](#_Toc69667444)

[Table 12: Admin Login 26](#_Toc69667445)

[Table 13: Features of Tested Table 27](#_Toc69667446)

[Table 14: Form Template Table 29](#_Toc69667447)

#### LIST OF ABBREVIATIONS

**Some of the abbreviations used in our report are shown in the table below:**

**HTML** Hyper Text Markup Language

**CSS** Cascading Style Sheet

**JS** JavaScript

**PHP** Personal Home Page

**MySQL** Microsoft Server Structured Query Language

**MVC**  Model View Controller

**Chapter 1**

# Introduction

## Introduction of the Project

Nowadays, there is Online Car Rental System which gives much benefit to user. A rental service is a service which customers arrive to request the hire of a rental unit. It is more convenient than carrying the cost of owning and maintain the unit. A car rental is a company that rent automobiles for short period of time for a fee for few hours or a few days or a week.

It helps to book the cars or vehicles online rather than using the traditional manual system of vehicle reservation. This eliminates the risk of erroneous booking and reduce overall lead time and ensures growth in customer satisfaction. They can book any car according to their brands and price.

## Problem Statement

As people mostly use the manual process to do their rental services, it was not the perfect as they wanted. The clients of Online Car Rental faced these problems:

* In order to offer cars for rent, a car owner normally have to come to the nearest branch office to register car to be rent out.
* Cars that provide difficulties to rent out are normally advertised in local or national newspaper. It involves a lot of paper work and consumes time.
* The system is not effective so we are developing a new web based application called Online Car Rental where user can register, login, view, book a car, post testimonial and can update their profile data.

## Objective

The objective of the project is to automate vehicle rental ad reservation so that the customers do not need to call and spend unnecessary time to reserve a vehicle.

* To transform the manual process of hiring car to a computerize system.
* To validate the Online Car Rental using user satisfaction test.
* To produce the documentation such as Software Requirement Specification (SRS), Software Design Description as system development reference.

## Scope of the Project

The scope of the project includes the following:

* Anyone can view the website and can login to book a car for rent.
* User already register can login and can view, book a car, post testimonial and can update their data.
* All information of the users can be stored systematically using this system.
* Provide dashboard for admin.
* Admin can manage everything using dashboard.

## Limitation of the Project

Some of the limitations are listed as:

* Only user who want to book a car for rent can use the system.
* There need to be reliable super user (Admin) otherwise system may be misused.
* Maintenance of the system is costly.
* Back paper features are not available.

## Features of the Project

The project is targeted to the users for online based. If any user first enters in the system, they are able to view websites. They will first register and login to website and they will see different model of cars for rent, they will choose particular model of car and can book the car for how much day they want by choosing the date from and to.

The main features of the project are described below:

* Responsive Design
* Login/Register
* Easy Bookings
* Manage Brands & Vehicles
* Manage Bookings
* Contact Queries
* Pages

**Chapter 2**

# Requirement Analysis and Feasibility Study

## Literature Review

## Requirement Analysis:

The requirement includes both functional and non-functional requirement**:**

### Functional Requirement

A typical functional requirement will contain a unique name and a number**,** a brief summary and a rationale.

* + - * **User Registration:** User can register so that they can login to view the vehicles and can book a car**.**
      * **Admin Login:** System admin can login with admin user name and password**.** Admin can add car, accept, reject a booking of car and testimonial and can edit, view, update and manage overall data.



Figure 1: Use Case Diagram

The Use Case Description:

Table 1: Use Case Description for Creating Account

|  |  |
| --- | --- |
| Primary Actor | User |
| Description | Creates a user profile |
| Pre-condition | Should be signed in |
| Post-condition | Can book vehicles |
| Fail Scenario | Not registered |

Table 2: Use Case Description for View Vehicles

|  |  |
| --- | --- |
| Primary Actor | User |
| Description | View vehicles |
| Pre-condition | Should be signed in |
| Post-condition | Can book vehicles |
| Fail Scenario | Does not have account |

Table 3: Use Case Description for Post Testimonial

|  |  |
| --- | --- |
| Primary Actor | User |
| Description | View Testimonial |
| Pre-condition | Should be signed in |
| Post-condition | Can post testimonial |
| Fail Scenario | Need of a user account |

Table 4: Use Case Description for Booking Vehicles

|  |  |
| --- | --- |
| Primary Actor | User |
| Description | Get vehicles details from system |
| Pre-condition | Should be signed in |
| Post-condition | Can book vehicles |
| Fail Scenario | Need of a user account |

Table 5: Use Case Description for Manage Testimonial

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can active or inactive testimonial |
| Pre-condition | Should be logged in and registered |
| Post-condition | Cannot book vehicles |
| Fail Scenario | Not registered |

Table 6: Use Case Description for Manage Booking

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can confirm or cancel booking |
| Pre-condition | Should be logged in and registered |
| Post-condition | Cannot book vehicles |
| Fail Scenario | Not registered |

Table 7: Use Case Description for Manage Brands

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can add new brands |
| Pre-condition | Should be logged in and registered |
| Post-condition | View all users who have book a vehicle |
| Fail Scenario | Not registered |

Table 8: Use Case Description for View Users

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can view registered users |
| Pre-condition | Should be logged in |
| Post-condition | Can view user details and remove them if necessary |
| Fail Scenario | There are no users registered |

Table 9: Use Case Description for View Contacts

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can view contacts |
| Pre-condition | Should be logged in |
| Post-condition | Can view user contacts and remove them if necessary |
| Fail Scenario | There are no users contact |

Table 10: Use Case Description for Remove Users

|  |  |
| --- | --- |
| Primary Actor | Admin |
| Description | Can remove users if necessary |
| Pre-condition | View all users’ details |
| Post-condition | Log Out from system |
| Fail Scenario | There are no users registered |

### Non – Functional Requirement

Non – Functional Requirements are**:**

* **Usability:** Every user can use this portal (website) easily.
* **Efficiency:** This web portal provides easy and fast access without consuming more cost**.**
* **Performance:** The performance of the System is good and is user friendly**.** The users easily access without any effort as an organized way and this make the web portal more reliable**.**

### Hardware Requirement

**System**: laptops/desktop with internet.

* Server-Side Requirement
  + Processor 2.0 GHz
  + Ram 2 GB
  + Hard Disk 8 GB
* Client-Side Requirement
  + Processor 1 GB

### Software Requirement

The application is developed using PHP.

* **Operating System:** Window.
* **Frontend:** HTML, CSS, JavaScript and Bootstrap.
* **Backend:** PHP (Scripting Language), MySQL (Database).
* **Web Browser:** Any Compatible Web Browsers.

## Feasibility Study

A feasibility study is conducted once the problem is clearly understood. The purpose of the study is to determine whether the problem is worth solving. It is an analysis and evaluation of a proposed project to determine if it is technically feasible.

We have studied different types of feasibility. Some of them are described below**:**

### Economic Feasibility

The economic feasibility of a system is used to evaluate the benefits achieved from and the costs incurred for the project or system. This is done by a process called cost benefit analysis. It provides tangible and intangible benefits like reduction in cost, more flexibility, faster activities, proper database management, etc.

The application is medium scale application and is economically feasible for us to accomplish it. This involves cost benefits analysis. Thus, there is no problem of high cost and cost benefits analysis.

### Operational Feasibility

The system is operational feasible as the system can be operate by normal users with basic computer skills without any additional trainings. We have developed this system with the willingness and ability to create, manage and operate the system which is easy for the end users to operate it.

### Technical Feasibility

To develop the application, web fundamentals like HTML, CSS, JavaScript for Front-end are used and for back-end PHP language has been used, whereas for database MySQL and XAMPP and Apache server has been used. Hence, the project seems feasible in terms of every technical aspect.

### Schedule Feasibility

Schedule Feasibility is the process of assessing the degree to which the potential time frame and completion dates for all major activities within a project meet deadlines and constraints for affecting change.

Table 11: Schedule Feasibility

|  |  |  |  |
| --- | --- | --- | --- |
| Task ID | TASK | Start Date | Durations |
| 1 | Requirement  Specification | 17-July | 10 days |
| 2 | Proposal Submission | 27-July | 10 days |
| 3 | Proposal Defense | 06-August | 10 days |
| 4 | User Interface design | 16-August | 10 days |
| 5 | Database Design | 26-August | 10 days |
| 6 | Coding | 26-August | 30 days |
| 7 | Testing | 05-October | 10 days |
| 8 | Documentation | 15-October | 10 days |

Figure 2: Gantt Chart

## Data Modeling

Entity-Relationship (E-R) diagram are commonly used in data modeling.

ER-Diagram is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities.



Figure 3: E-R Diagram

Each of the entities have various and unique attributes that are essential for the existence of the entities and give the entities their meaning.

## Process Modeling

The process modeling of the application is done through Data Flow Diagram and schema diagram.

### Data Flow Diagram (DFD)



Figure 4: Data Flow Diagram

Above Data Flow Diagram, explains the overall structure of the system. It shows how and what types of services the client chooses and the amount of admin interaction in it.

**Chapter 3**

# System Design

## System Design

System design is a process of defining the components, modules, interfaces and data for a system in order to satisfy specified requirements. It can also be defined as a process of creating or altering systems along with the processes, practices, models and methodologies that can be used to develop them. The main objective of the detailed system design is to prepare a blueprint of a system that meets the goals of the conceptual system design requirements. The system designs used for building this project include system architecture, database schema, input output design, class diagram.

## System Architecture

The proposed system is based on MVC (Model View Controller) framework. The model is responsible for getting data from a database, packaging it in data objects that can be understood by other components, and delivering those objects, most of which will happen in response to input from the controller. The controller is how the user interacts with the application. The view is where the data provided by the model is presented to the user.

View (car booking form)

Controller (accepts requests and transfer data to database)

Model/Database (response to server queries)

User

Figure 5: MVC System Architecture

## Interface Design

### User Interface Design

The UI design for both of the front-end and back-end is shown below:



Figure 6: Front-End User Interface Design

### Admin Panel Design

Admin Panel is the interface where administrations can manage overall information of the website and also helps to confirm, cancel, active, inactive the booking of car and testimonial. The admin can:

* + - * Can manage the user details.
      * Can post and manage the brand and vehicles.
      * Can manage booking of car and testimonial.
      * Can manage pages and update contact info.

Back-End

Admin

Can manage pages and update contact information



Can manage the booking of car and testimonial

Can manage the brand and vehicles

Can manage the user details

Dashboard

System

Figure 7: Admin Panel User Interface Design

## Database Schema Design:

A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of formulas known as integrity constraints that govern a database. Those formulas are expressed in a data definition language, such as SQL. As part of a data dictionary, a database schema indicates how the entities that make up the database relate to one another, including tables, views, stored procedures, and more.

## 

Figure 8: Schema Diagram

## Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. A sequence diagram specifically focuses on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.



Figure 9: Sequence Diagram

## Class Diagram

A class diagram in the [Unified Modeling Language (UML)](https://en.wikipedia.org/wiki/Unified_Modeling_Language) is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



Figure 10: Class Diagram

## Algorithm and Flowchart



Figure 11: Flowchart of Online Car Rental

**Chapter 4**

# Implementation and Testing

## Implementation

Implementation phase is one of the important phases of project development. In this phase we implement our conceptual design into the working program by using various tools. The successful implementation of project is nearer steps towards the project completion. Project implementation was not an easy step to us as we encountered various issues related to the programming logic as challenges.

It is the final phase of moving the solutions from development status to the production status. In this phase the project developers begin building and developing the software for system. The phase is followed by software development life cycle model used ,tools used and description of major classes/methods

## Tools Used

### Front-End

A "front-end" application is one that application users interact with directly. The front end is an interface between the user and the back end. In simple words we can understand the concept of front and back end with the help of below given definition. In client/server applications, the client part of the program is often called the front end and the server part is called the back end. While creating a front end, different components relating to the software development were used. They are listed below:

### a. HTML/CSS/JavaScript

HTML stands for hypertext markup language. It is not a programming language but a markup language that specifies the layout and style of a document. HTML documents describe web pages, which is read by web browser to display the web pages.

CSS stands for cascading style sheets. It is used for styling the layout of HTML document and allows developers to separate content from design and layout so that HTML could perform more of the function without worry about the design and layout. It is used to separate style from content.

JavaScript is a scripting language designed to add interactivity to HTML pages. It helps to make html pages dynamic meaning it can react to events and can be set to execute when something happens, like when a page has finished loading or when a user clicks on an html element.

### b. Bootstrap

Bootstrap is the most popular framework used to create a responsive and beautiful web pages. Bootstrap is a free and open-source collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications.

Bootstrap is a front-end framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

### Back-End

A "back-end" application or program serves indirectly in support of the front- end services, usually by being closer to the required resource or having the capability to communicate with the required resource. The back-end application may interact directly with the frontend or, perhaps more typically, is a program called from an intermediate program that mediates front-end and back-end activities.

It is the part of the system that normal users don’t want to know about it. System administrator manages it. The following parts are contained in the back end of our system:

### PHP

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server code is embedded into the HTML source document. PHP code in a requested file is executed by the PHP runtime, enabling our web application to display dynamic web content.

### MySQL

MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database.

### XAMPP Server

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standards. The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, OpenSSL, and SQLite.

### Documentation Tools

* + - * **MS Word 2019:** MS Word is used for making the documentation of the project.
      * **Visio 2019:** Visio has been used for drawing different diagrams like Use case, ER, Sequence, Class Diagram, Flowchart etc.
      * **Visual Studio Code**: Visual Studio Code is a code editor where codes for designing frontend and backend are written an edited.

## Methodology

There are several methods which can support developing process of any project such as:

* Waterfall Prototyping
* Incremental development
* Iterative model, etc.

But according to the development of the project Incremental Development Model is used.

Initialization

Requirements

Design

Planning

Deployment

Implementation

Verification

Figure 12: Iterative Model for System Development

## Testing

Software Testing has a dual function; it is used to identify the defects in program and it is used to help judge whether or not program is usable in practice. Thus, software testing is used for validation and verification, which ensure that software conforms to its specification and meets need of the software customer**.**

The testing phase can be carried out manually or by using automated testing tools to ensure each component works fine. After the project is ready we tested its various components in terms of quality, performance to make it error free and remove any sort of technical jargons. The testing phase can be carried out manually or by using automated. The testing phase can be carried out manually or by using automated testing tools to ensure each component works fine.

### Unit Testing

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews

Table 12: Admin Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Test Inputs** | **Expected Outcome** | **Actual Outcome** | **Results** |
| 1 | Admin: admin  Password: Test@12345 | Successfully Login | Login Successful | Test Successful |
| 2 | Admin: admin  Password: admin1 | Incorrect ID  or Password | Incorrect ID  or Password | Test Unsuccessful |

Input:

* Unit Test Specifications
* Code to be tested

Testing Process:

* Checking for availability of Code Walk-through reports which have documented the existence of and conformance to coding standards.
* Verify the Unit Test Specifications conform to the program specifications.
* Verify that all boundary and null data conditions are included.

Table 13: Features of Tested Table

|  |  |
| --- | --- |
| **Test Specification** | **Description** |
| GUI and general Tests | Screen consistency with respect to project specific standards and checklist. |
| Menu commands are executed at least once | Functionality with respect to description in menu or image in tool bar. |
| Functionality Test | All possible scenarios to test the functionality of the component are listed here. This list is made very exhaustive to cover all the expected functionality described in the Software  Requirement Specifications and Design document completely. |
| ‘Null Data’ cases are covered | Handling of Null values. |

Unit Test Specification

A sample Unit Test Specification is as follows:

Table 14: Form Template Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Action** | **Expected Result** | **Observed Result** | **Verified** |
| 1 | On pressing book button | Should add the new records to the database. | As Expected | Yes |
| 2 | On pressing confirm button | Record should be inserted in database state of other. | As Expected | Yes |
| 3 | On pressing update button | Record should be updated. State of other  Buttons in Toolbar Add= Disable Save= Enable  Query= Disable Fetch= Disable Delete= Disable Cancel= Enable | As Expected | Yes |
| 4 | On pressing Delete button | Should delete the selected records from the database. | As Expected | Yes |

### Integration Testing

After the individual modules were tested out this project, proceed to the integration testing to create a complete system. This integration process involves building the system and testing the resultant system for problems that arise from component interactions.

In this project, applied top-down strategy to validate high-level components of a system before design and implementations have been completed. In this development process started with high-level components and we worked down the component hierarchy.

### System Testing

System testing is actually a series of tests whose purpose is to fully exercise the Web- based system. It verifies that system elements have been properly integrated and perform allocated functions. It checks whether the system as a whole works as per requirement. We have used Performance testing. Performance testing-designed to test the run-time performance of software, especially real-time software.

Acceptance Testing

System Testing

Integration Testing

Unit Testing

Figure 13: Testing

**Chapter 5**

# Maintenance and Support

## Maintenance

Online Car Rental System is implemented in the XAMPP server. For now, our system is implemented with browser. XAMPP Server uses MySQL for database storage server. Also, admin can edit, add, delete update all the brands, vehicles, testimonial, pages and contact information.

XAMPP server is used as a connection between the PHP code and database where database has been created.

Database consist of many tables each table has some attributes and entities. Tables field are also set as primary key and foreign key.

## Support

The system is an online based system that provides services 24\*7 for users and admin where they can fetch the site whenever they want. Database support is provided by XAMPP server which provides proper connectivity between the entities of the sites and database.

Here, all the information regarding users is saved securely in the database by creating different tables. Tables includes primary key and unique key that reduces duplication of data.

The system can be supported by any web browsers. Hence, user-friendly and flexible.

**Chapter-6**

# Conclusion and Future Enhancement

## 6.1 Conclusion

* + - Online Car Rental System is a web-based system that helps the user to view the vehicles. Not only the vehicles but also the user can login and can book vehicles according to their choice.
    - The application has been developed using command–line tools, visual studio code as defining layout and backend part using the PHP.
    - All the requirement and the objectives of the project have been fulfilled by the system.
    - The users can access all functionalities provided by the system and he/she can view other information on website.
    - This provides easy, accurate unambiguous and faster data access.

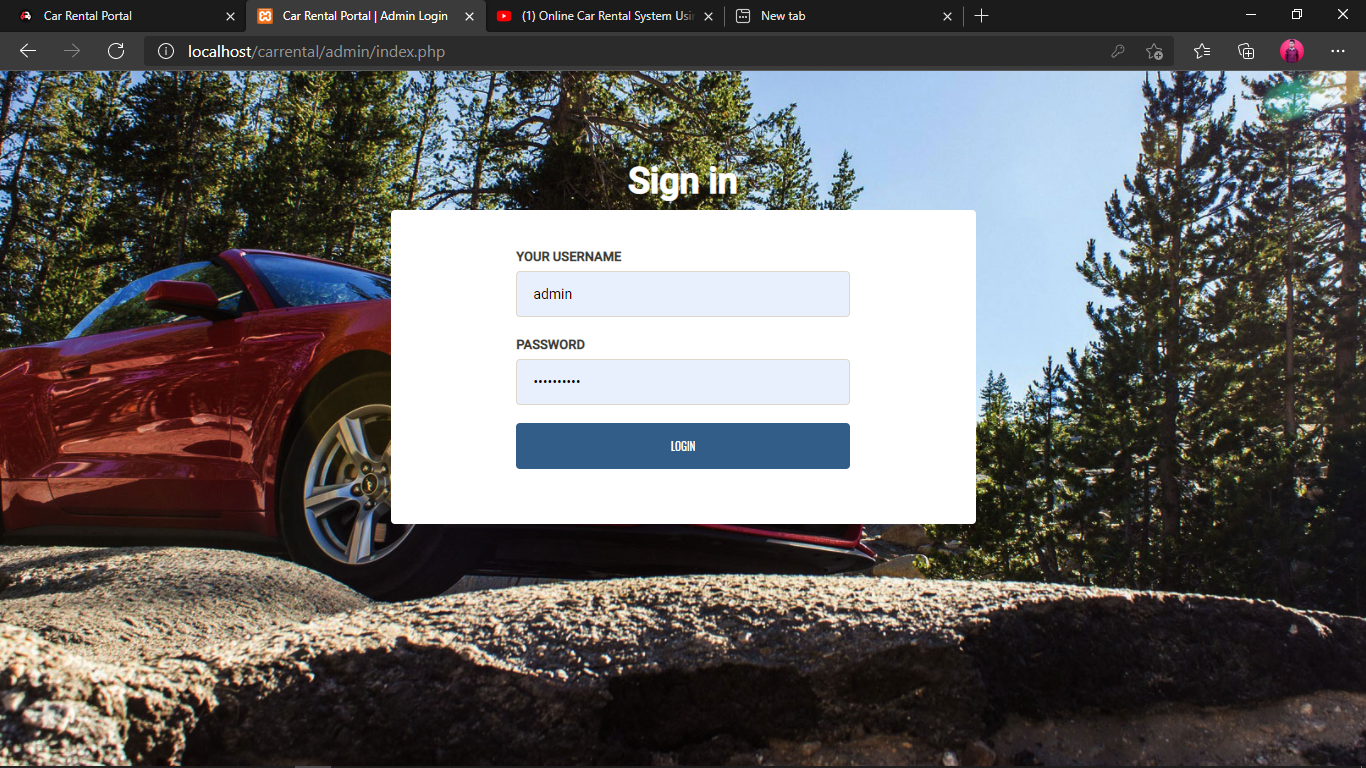
## Future Enhancement

Currently the application has the features which allows user to login in the website to book a car. The user can search for cars easily and book. For bookings, the user has to provide information such as booking dates and text message. All car details are provided and it also includes car’s feature and overview. The user can also post their testimonials and the user can update their profile as well as passwords anytime they want from the site. Admin can add/manage car brands, manage vehicles, bookings, testimonial, pages and many more. It’s easy to operate and understand by users. This site makes customers/users easy for car rental. The design is pretty simple and the user won’t find it difficult to understand, use and navigate.

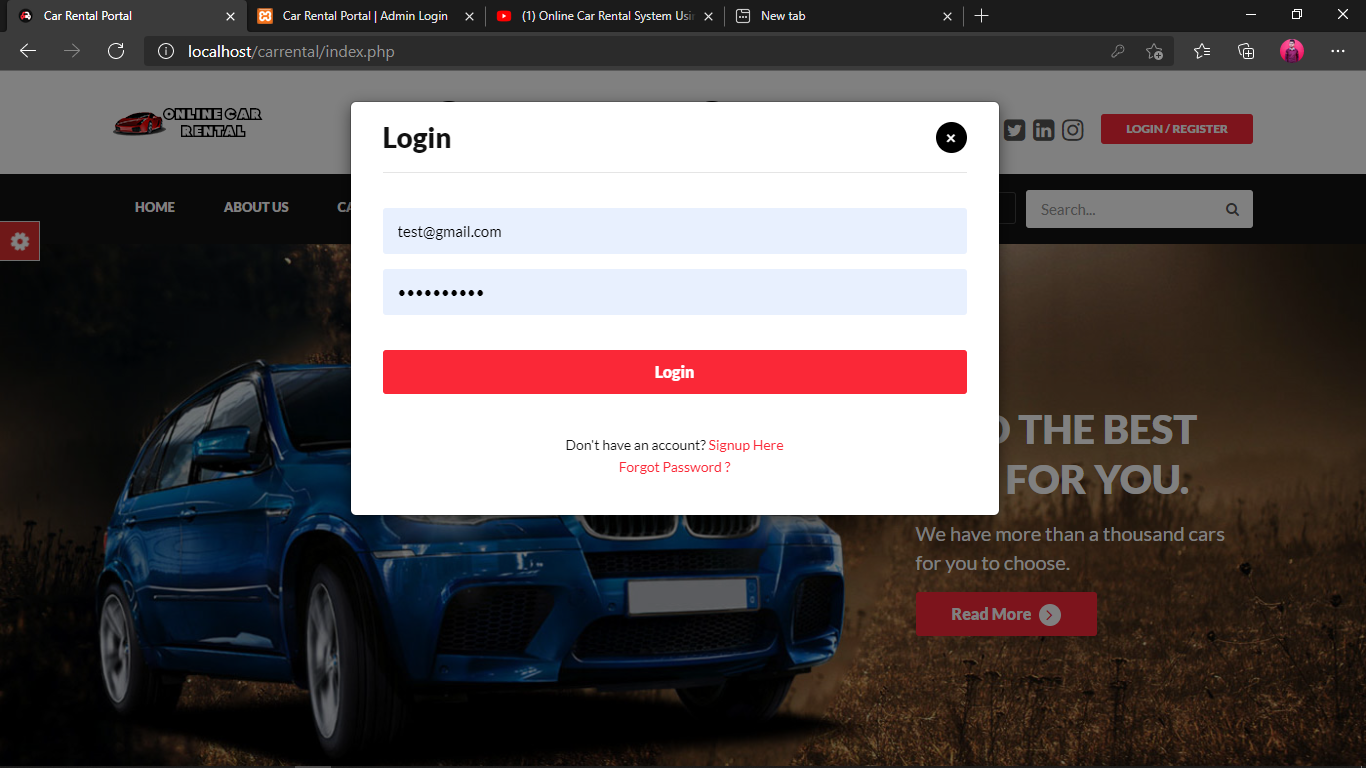
* + - Application will provide the user to subscribe so that they can get latest news about our website.
    - Currently we are focusing only on user who need car for rent but in future we will add feature like apartment for rent.

# Appendices

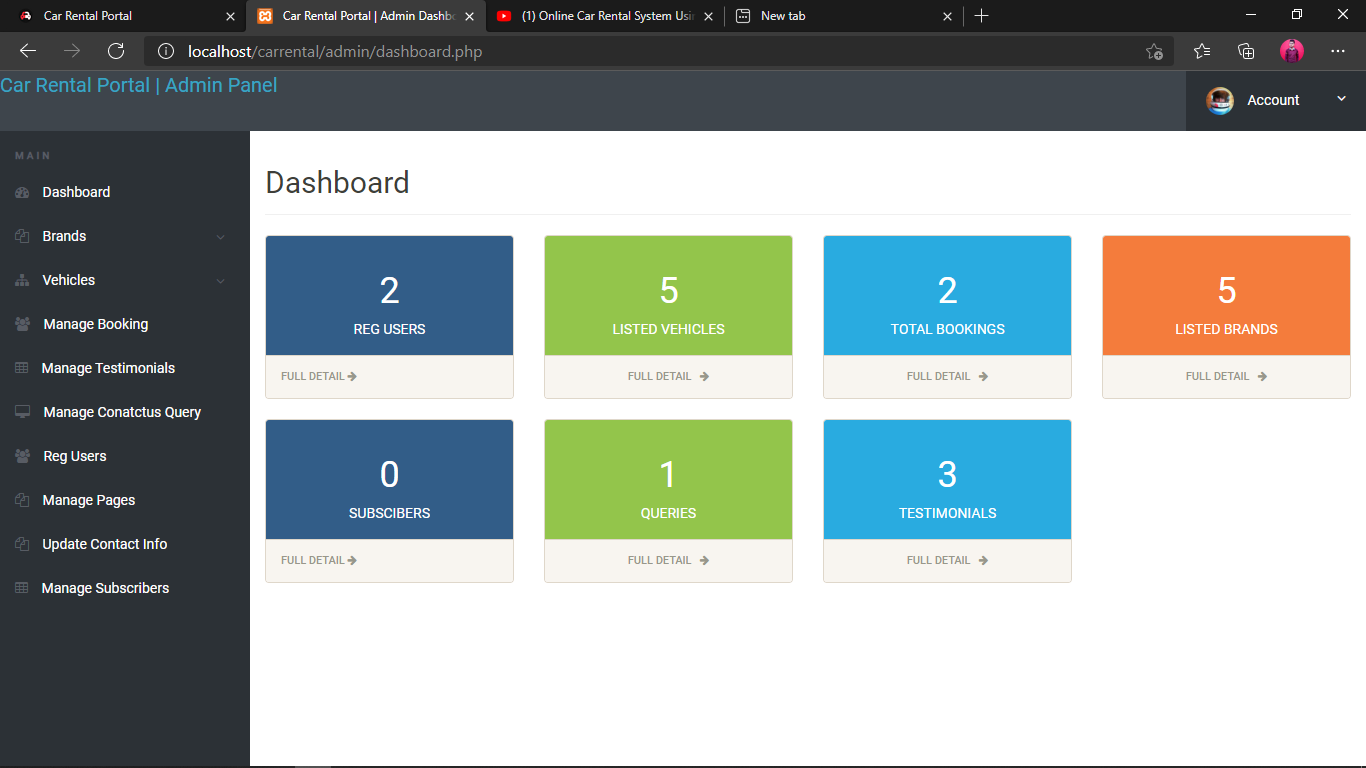
## Snapshots:



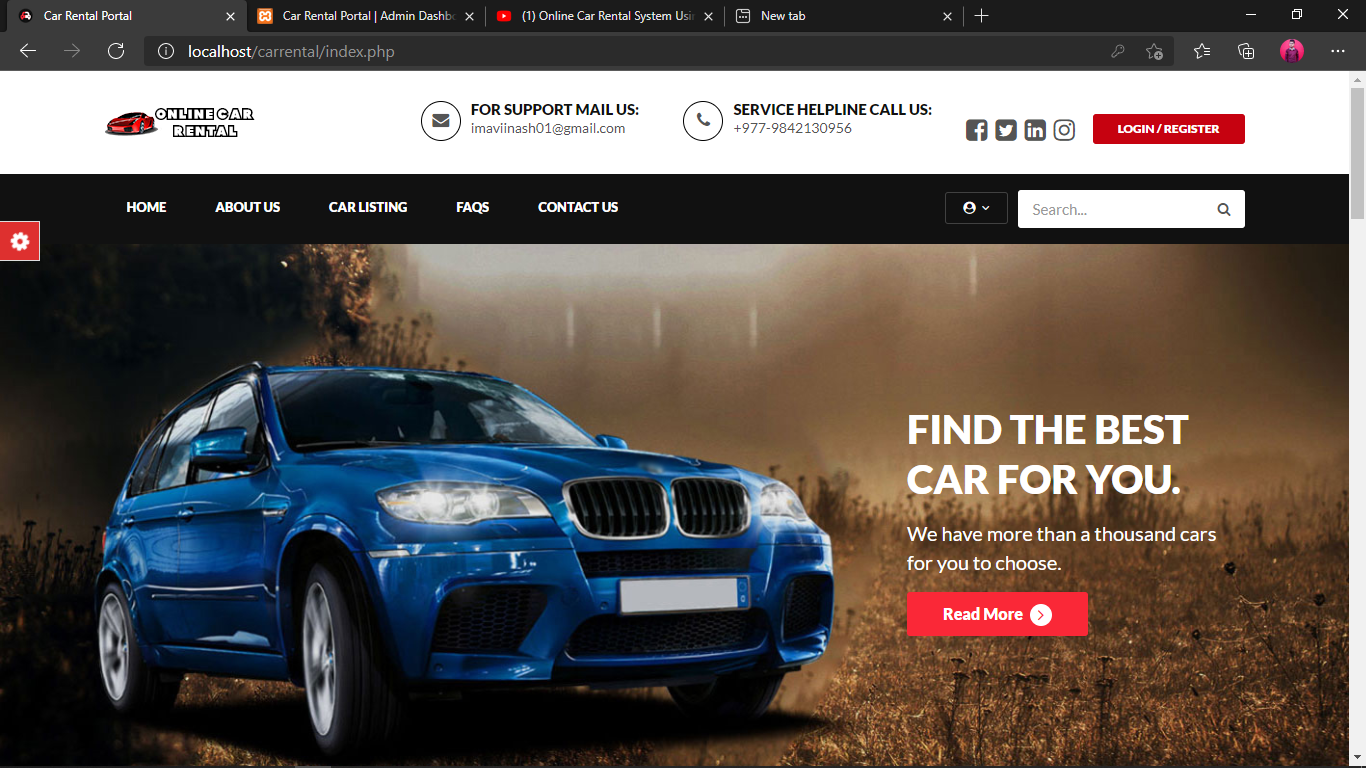
**Snapshot of Admin Login form**



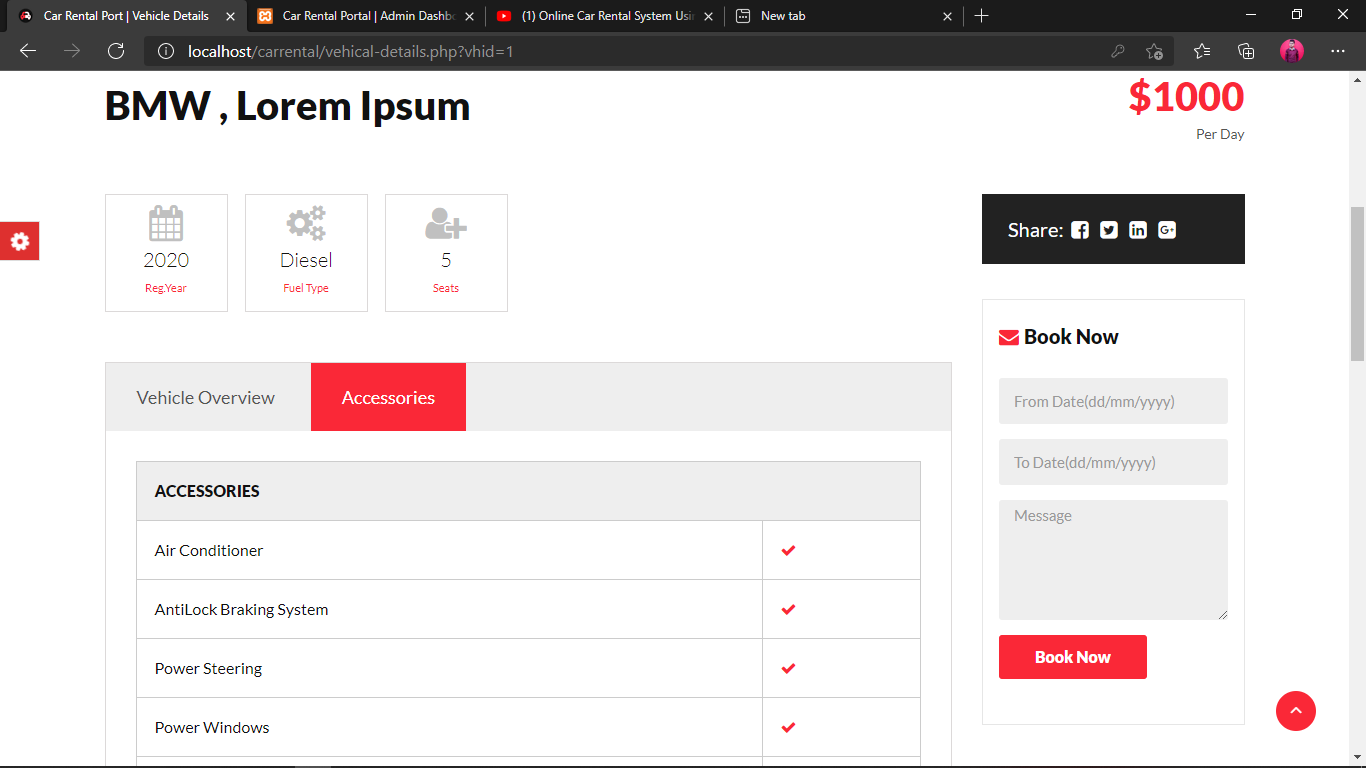
**Snapshot of User Login Form**



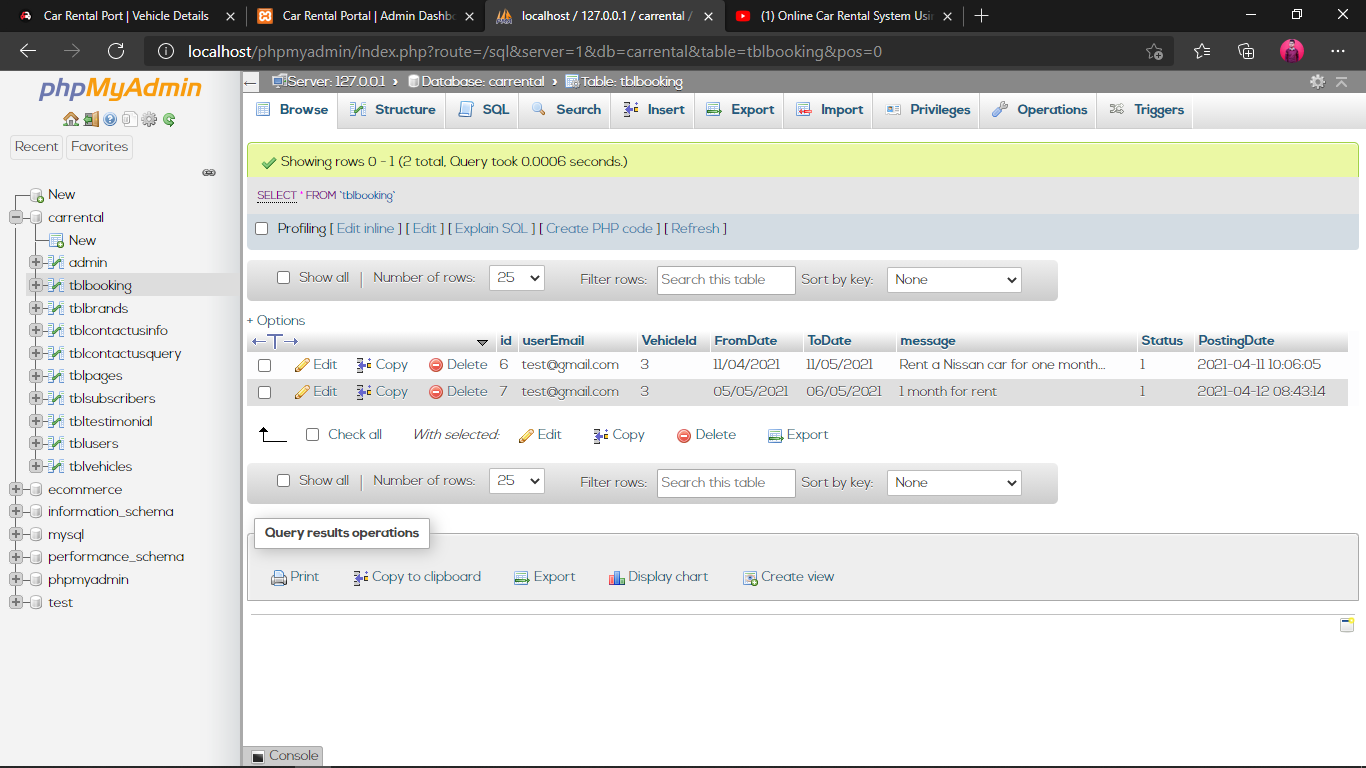
**Snapshot of Admin Dashboard**



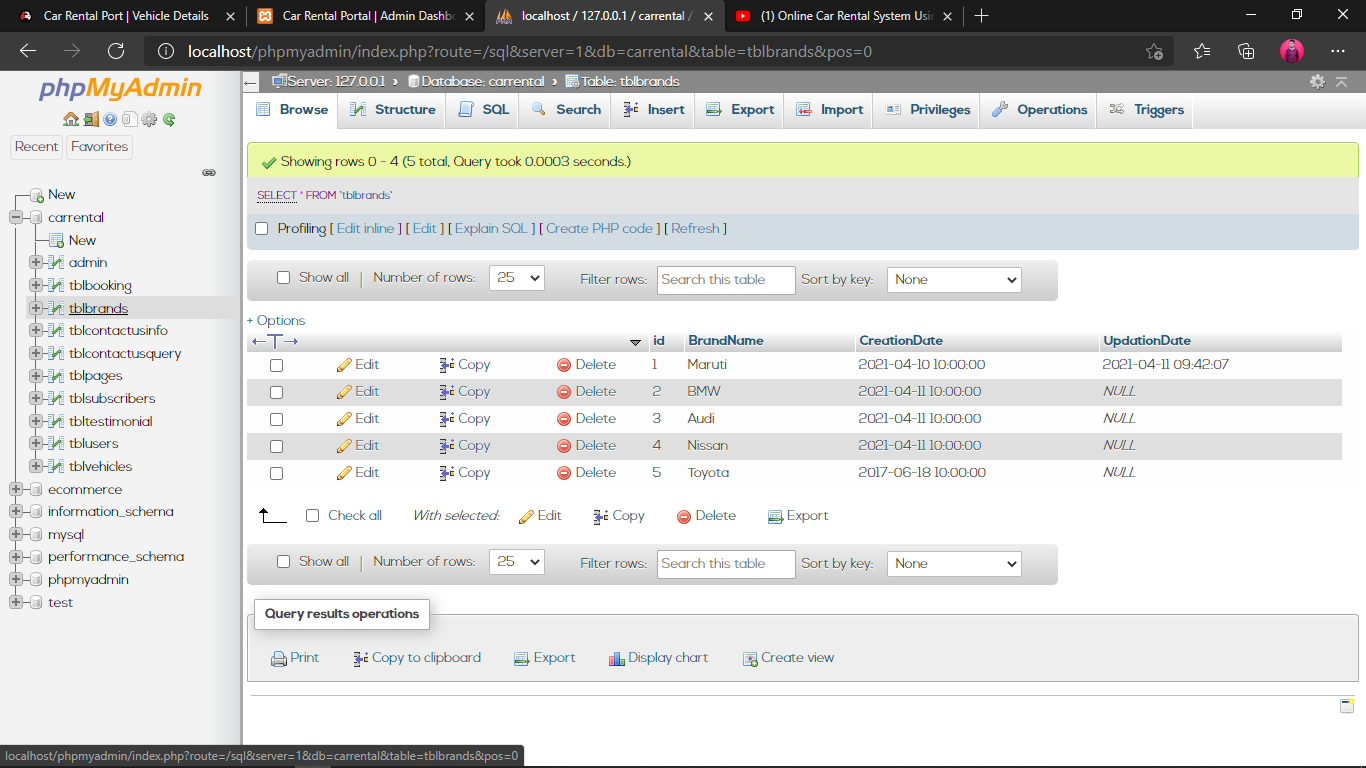
**Snapshot of Home page**



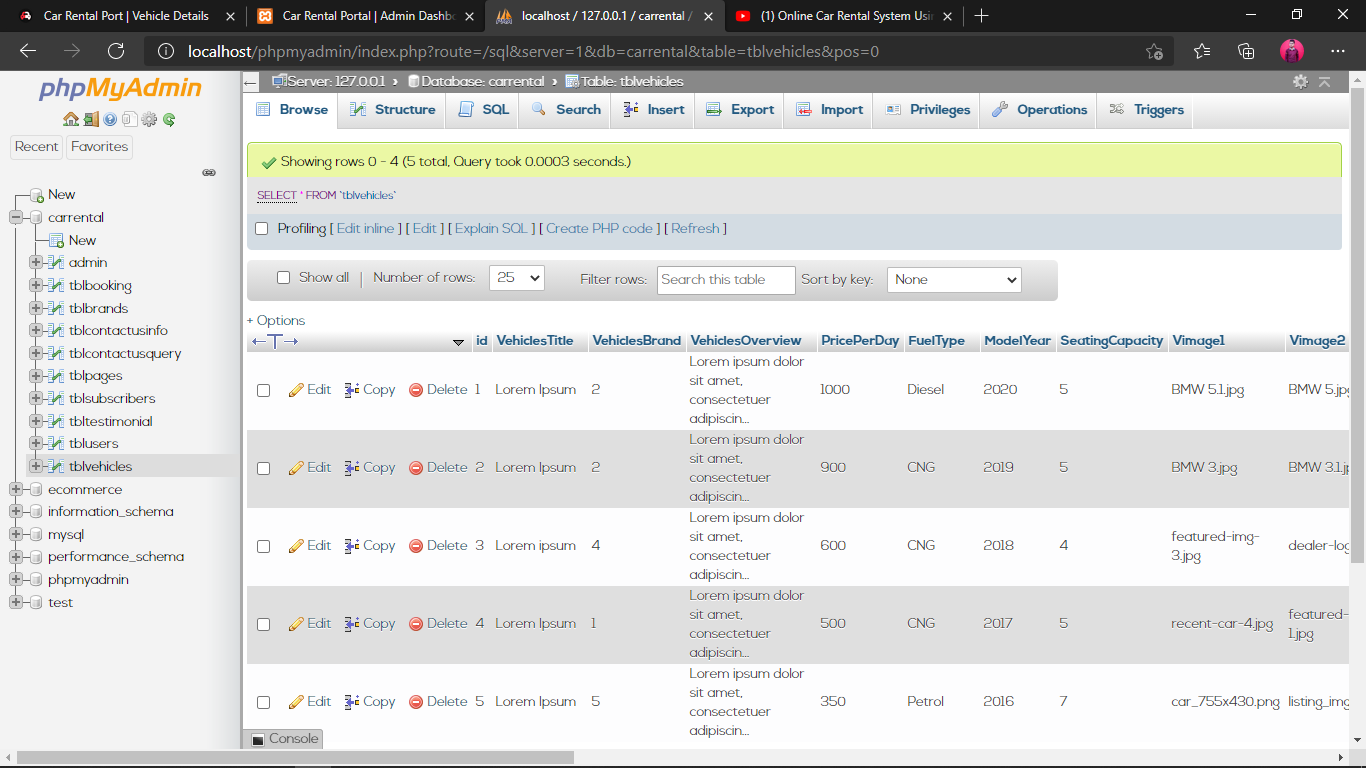
**Snapshot of Booking Form**



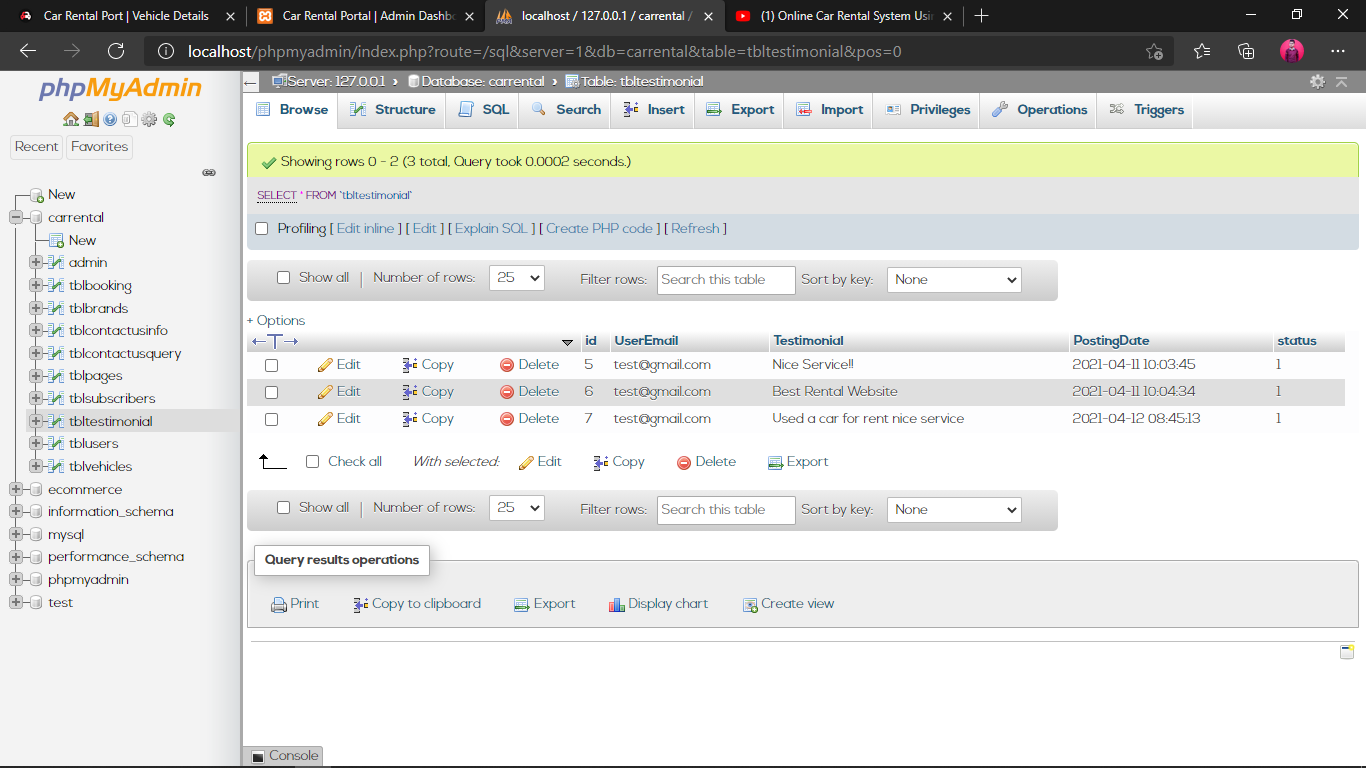
**Snapshot of Booking Table in Database**



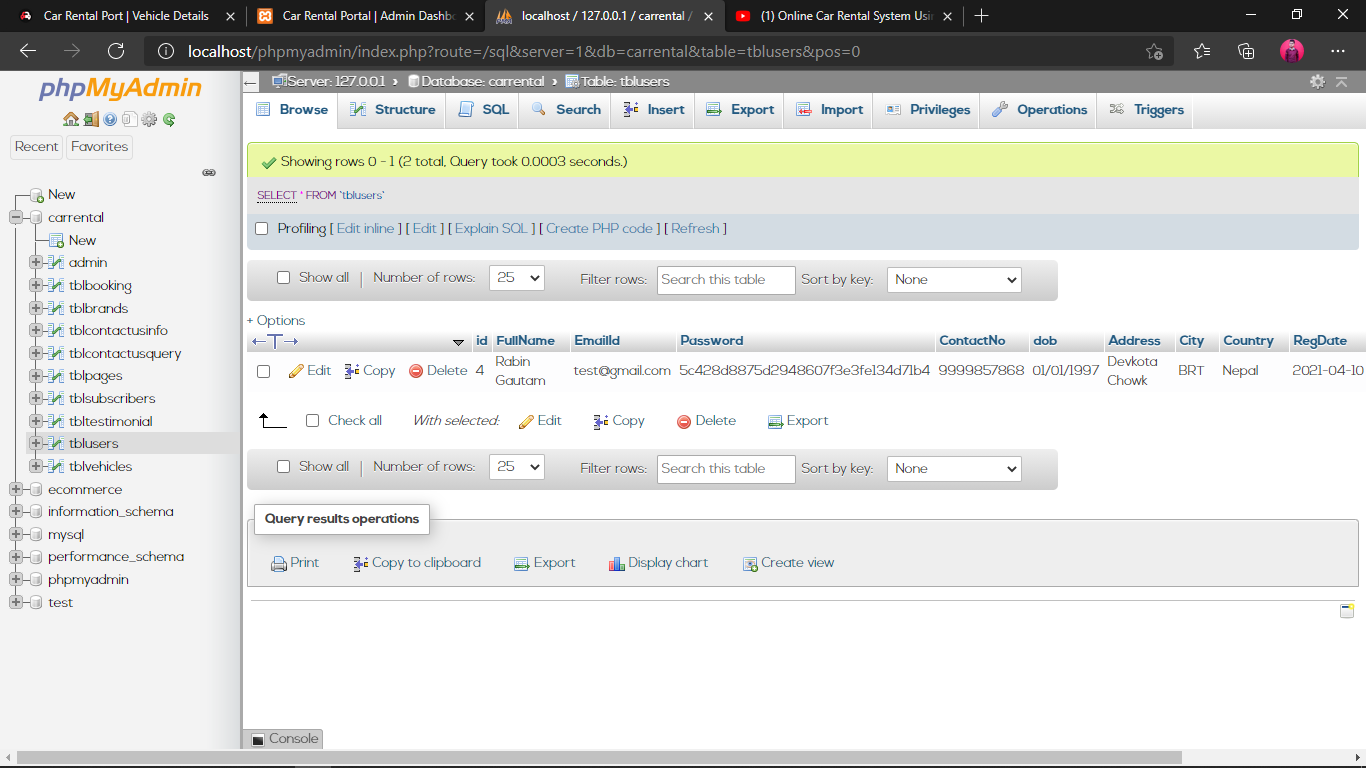
**Snapshot of Brand Table in Database**



**Snapshot of Vehicles Table in Database**



**Snapshot of Testimonial Table in Database**



**Snapshot of User Table in Database**

**References:**

1. Stellman, Andrew; Greene, Jennifer (2005). *Applied Software Project Management*. O'Reilly Media. p. 113. ISBN 978-0-596-00948-9.
2. Jennifer (2005).[*http://www.slideshare.net/olmsgal/collecting-research-data-with-*](http://www.slideshare.net/olmsgal/collecting-research-data-with-) *questionnaires-and- interviews, March*-12, 2019.
3. Bruza, P. D., Van der Weide, Th. P., "*The Semantics of Data Flow Diagrams*", March- 12, 2019.

[4] Stellman, Andrew; Greene, Jennifer (2005). S. Gündüz-Ögüdücü, "*Web Page Recommendation Model*," Morgan & Claypool, 2011.

1. Andrew (2009). S. Gündüz-Ögüdücü, “Recommendation *Model*," Morgan & Claypool, 2019.
2. Jennifer(2007)“*https:/*[*/w*](http://www.lucidchart.com/documents%23docs?&browser=icon&sort)*w*[*w.lucidchart.com/documents#docs?&browser=icon&sort=*](http://www.lucidchart.com/documents%23docs?&browser=icon&sort) *saved-desc*, 12th August, 2019.
3. Thomas, Builtinmethods using JavaScript Available at*: https://*[*www.codecademy.com/lea*](http://www.codecademy.com/lea) *rn/introduction-to-javascript*, 7th April, 2019.