

# **CAR RENTAL SYSTEM**

*Project Report Submitted by*

**ARUN BABU**

**Reg. No.: AJC21MCA-2041**

*In Partial fulfillment for the Award of the Degree of*

**MASTER OF COMPUTER APPLICATIONS**

**(MCA TWO YEAR)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2021-2023**

**DEPARTMENT OF COMPUTER APPLICATIONS**  
**AMAL JYOTHI COLLEGE OF ENGINEERING**  
**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**CAR RENTAL SYSTEM**” is the bona fide work of **ARUN BABU (Regno: AJC21MCA-2041)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Ms. Nimmy Francis**

**Internal Guide**

**Ms. Meera Rose Mathew**

**Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**External Examiner**

**// CERTIFICATE ON PLAGIARISM CHECK**

## **DECLARATION**

I hereby declare that the project report “**CAR RENTAL SYSTEM**” is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

**Date:**

**ARUN BABU**

**KANJIRAPPALLY**

**Reg: AJC21MCA-2041**

## ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Nimmy Francis** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ARUN BABU

## **ABSTRACT**

Car Rental System is a large database system which can be used for managing renting cars in a day-to-day business. It allows Renter to give their cars for daily rent and customers to use the cars and store almost all of their information electronically, including information on cars, customer, renter etc. Renters can rent their cars for a short period of time for a few days or week. There is no delay in the availability of any car information, whenever needed, car information can be captured very quickly and easily. This system will helpful to the admin as well as to the customer also.

The core concept of this project is to manage a car rental service. There are 5 users including admin. The admin controls the whole system. He approves and verifies the leased vehicle according to the overall review from car checker. This checker checks the leased cars.

The renter can lease their vehicle if approved and he can add drivers to their own cars, the Customers can hire the vehicle and driver, and opting the driver is an optional choice.

# CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	
1.1	PROJECT OVERVIEW	
1.2	PROJECT SPECIFICATION	
2	SYSTEM STUDY	
2.1	INTRODUCTION	
2.2	EXISTING SYSTEM	
2.3	DRAWBACKS OF EXISTING SYSTEM	
2.4	PROPOSED SYSTEM	
2.5	ADVANTAGES OF PROPOSED SYSTEM	
3	REQUIREMENT ANALYSIS	
3.1	FEASIBILITY STUDY	
3.1.1	ECONOMICAL FEASIBILITY	
3.1.2	TECHNICAL FEASIBILITY	
3.1.3	BEHAVIORAL FEASIBILITY	
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	
3.2	SYSTEM SPECIFICATION	
3.2.1	HARDWARE SPECIFICATION	
3.2.2	SOFTWARE SPECIFICATION	
3.3	SOFTWARE DESCRIPTION	
3.3.1	PHP	
3.3.2	MYSQL	
4	SYSTEM DESIGN	
4.1	INTRODUCTION	
4.2	UML DIAGRAM	
4.2.1	USE CASE DIAGRAM	
4.2.2	SEQUENCE DIAGRAM	
4.2.3	STATE CHART DIAGRAM	
4.2.4	ACTIVITY DIAGRAM	
4.2.5	CLASS DIAGRAM	
4.2.6	OBJECT DIAGRAM	
4.2.7	COMPONENT DIAGRAM	

<b>4.2.8</b>	<b>DEPLOYMENT DIAGRAM</b>	
<b>4.2.9</b>	<b>COLLABORATION DIAGRAM</b>	
<b>4.3</b>	<b>USER INTERFACE DESIGN USING FIGMA</b>	
<b>4.4</b>	<b>DATABASE DESIGN</b>	
<b>5</b>	<b>SYSTEM TESTING</b>	
<b>5.1</b>	<b>INTRODUCTION</b>	
<b>5.2</b>	<b>TEST PLAN</b>	
<b>5.2.1</b>	<b>UNIT TESTING</b>	
<b>5.2.2</b>	<b>INTEGRATION TESTING</b>	
<b>5.2.3</b>	<b>VALIDATION TESTING</b>	
<b>5.2.4</b>	<b>USER ACCEPTANCE TESTING</b>	
<b>5.2.5</b>	<b>AUTOMATION TESTING</b>	
<b>5.2.6</b>	<b>SELENIUM TESTING</b>	
<b>6</b>	<b>IMPLEMENTATION</b>	
<b>6.1</b>	<b>INTRODUCTION</b>	
<b>6.2</b>	<b>IMPLEMENTATION PROCEDURE</b>	
<b>6.2.1</b>	<b>USER TRAINING</b>	
<b>6.2.2</b>	<b>TRAINING ON APPLICATION SOFTWARE</b>	
<b>6.2.3</b>	<b>SYSTEM MAINTENANCE</b>	
<b>6.2.4</b>	<b>HOSTING</b>	
<b>7</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	
<b>7.1</b>	<b>CONCLUSION</b>	
<b>7.2</b>	<b>FUTURE SCOPE</b>	
<b>8</b>	<b>BIBLIOGRAPHY</b>	
<b>9</b>	<b>APPENDIX</b>	
<b>9.1</b>	<b>SAMPLE CODE</b>	
<b>9.2</b>	<b>SCREEN SHOTS</b>	



## **List of Abbreviation**

IDE - Integrated Development

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SASS - Syntactically Awesome Style Sheets

SQL - Structured Query

UML - Unified Modeling Language

PHP - Hypertext Preprocessor

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 PROJECT OVERVIEW

‘Car Rental System’ is a web application to rent different types of cars that are available in different parts of Kerala. This is intended to help customers to get a variety of cars for rent. The customer can reduce the time, effort and cost for searching rental cars. Customers can easily find the kind of car they need and can book it. If the car comes with a driver, there is an option to book the driver as well. The proposed system includes five users which include administrator, car checker, renter, driver and customer. Registered renters can log in to the site and add their car for renting. They can also add drivers to their cars.

Customers can also log in to the site and select cars from these renters for their use. Admin has an overall management power. The car checker checks the cars added by the renter and it’s a manual process and gives reviews about the cars. Based on the review given by checker admin approve the cars to the website.

## 1.2 PROJECT SPECIFICATION

The system includes 3 modules. They are:

### ➤ Admin Module

Admin must have a login into this system. He has overall control of the system. He manages the users and cars, and check the credibility. Functionalities

- Login
- Add checker
- Manage Renter
- Manage customer
- Manage cars.
- Manage driver
- Add models

### ➤ Renter Module

Renters can register and log in to the website. They can add cars for renting that are available with them. They have an option to edit their profile and car details. Renters can choose the

customers for their cars according to the customer requests. Functionalities

- Registration
- Login
- Add car details
- Add driver
- View/Approve booking
- Assign driver
- Add/view payment
- Add status
- View feedback
- View status
- Update profile

#### ➤ **Customer Module**

Customers can register and log in to take car for rent from the website.

They can book available cars for their need. If the car comes with a driver, there is an option to book the driver as well. They also have an option to edit their profile. Functionalities

- Registration
- Login
- View cars
- Car booking
- Add/view payment
- Add feedback
- Update profile

#### ➤ **Driver Module**

Driver can login to the website. They have an option to view their booking details and view

payment details too. Functionalities

- Login
- View booking
- View payment
- Update profile

#### ➤ **Checker Module**

Checker can login to the website. They have an option to accept the cars assigned to them and add their reviews or comments about that car. Based on this review admin approve the cars to the website. Functionalities

- Login.
- View/accept assigned cars.
- Add reviews about car.
- Update profile

## **CHAPTER 2**

### **SYSTEM STUDY**

## **2.1 INTRODUCTION**

System analysis is a review of a technological system, like a software package, for troubleshooting, development, or improvement purposes. Through in-depth analysis, analysts can uncover errors in code, accessibility issues for end-users or design incompatibilities. Performing an effective systems analysis often requires experts to have knowledge of a software product's or package's requirements so that they can approach their analysis effectively. Unlike systems administrators, who focus on day-to-day system maintenance, systems analysts consider the viability and effectiveness of a product overall. This allows them to suggest changes or make fixes that improve the system.

With a systems analysis, considering the goals of the system is important for solving problems and creating efficiencies. From there, dividing a system into components can make it easier to perform individual analyses that influence the complete system.

## **2.2 EXISTING SYSTEM**

### **2.2.1 NATURAL SYSTEM STUDIED**

In natural system the user or customer will directly interact with the car owner and owner will decide whether the car is available or not. Then if it is available, he will give rent a car to the customer. The main drawback of this system is customer need to meet the car owner; this is time waste process.

### **2.2.2 DESIGNED SYSTEM STUDIED**

In the existing system I studied 'www.indusgo.in'. In this website, the customers are able to view the cars and book the cars after registration. Cars from the provided website are very limited since they took a small category into consideration. Because it's only a site for that agency. Here we can book cars for rent without drivers. They have delivery from some dedicated places and also have some drop-off locations. They provide cars with/without fuels and they have different price tags. In indigo.in, the customer needs to contact manually the admin or the authority of that agency

## **2.3 DRAWBACKS OF EXISTING SYSTEM**

- In natural system its very time consuming for finding a good car. Sometimes we don't get a car for our time period.
- In designed system, there is no concept of individual renters. There is only a rent shop or a showroom.

- There is a limitation in pick up and drop off locations.
- Cars provided from this website are very limited.

## **2.4 PROPOSED SYSTEM**

This Car Rental system helps to automate the manual tasks and maintain the record of renting a car. It can perform all functionalities associated with renting a car fully with online. It will help to find cars rented in our locality very easily and we get many options of cars here. In the existing system I have studied that is indigo.in it is only for that indigo agency. But here I am planning to create a website that anyone can register as renter, customer. Here anyone can rent his car to anyone using this website. And here we have a user-friendly section through online. Based on the type of car required by the customer, they shall be able to make bookings.

## **2.5 ADVANTAGES OF PROPOSED SYSTEM**

- Customer can easily find nearby cars for their needs.
- Any individual person has the opportunity to rent their cars through online.
- All renting sections are in online don't need to keep the documents in their hands or in physical copy.
- Customer have a wide variety of options of cars.
- Booking sections are very user friendly and easy to use.
- It helps renters in advertise their cars through the availability of the online system.



## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

### **3.1 FEASIBILITY STUDY**

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organization in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. The premise for a feasibility study is the system proposal's viability, which includes the impact on the organization, ability to meet user needs, and effective use of resources. As a result, a feasibility evaluation is frequently performed before a new application is approved for development.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics: -

#### **3.1.1 Economical Feasibility**

Cost and benefit analyses are required to support the emerging system. Criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables. Some of the significant financial queries raised during the initial probe include the following:

- The expenses carry out a comprehensive system investigation.
- The price of the software and hardware.
- The advantages in terms of lower expenses or fewer expensive mistakes.

The proposed system was created as part of a project; hence, there are no manual expenses associated with it. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably. The cost of project, Car Rental System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

#### **3.1.2 Technical Feasibility**

A technical feasibility study examines how you intend to customers. Think about the materials, labor, transportation, where your business will be locked, and the technology that will be needed to bring it all together. The researches involved in this project will go through the same process, as the system progress, the researches will move forward in the same way. It is also important to check the state of the software and hardware because the developed of these things. Each time the

researches develop a system and put all their education into practice.

- Is the proposed technology or solution practical?

The implementation of the Car Rental System: This system would be able to save time and efforts of renters and customers. In that way some information are attained. The practicality of getting result along with its automation would be more than what the manual way does from it.

### 3.1.3 Behavioral Feasibility

The following inquiries are part of the suggested system:

- Is there enough assistance for the users?
- Will the suggested system harm anyone?

Because it would accomplish the objectives after being developed and put into action, the project would be advantageous. After carefully examining all behavioral parameters, it is determined that the project is behaviorally feasible.

### 3.1.4 Feasibility Study Questionnaire

Details of person that you have contacted for data collection?

Jomon Philip, 8943661340

Interview Q&A through direct contact.

1. What type of car that you provide?

Only one alto 800.

2. How the customers contact you or how they know you are a renter?

Almost 5 years am renting my car. The customers know me through their friends or my friends. They contact me through WhatsApp or mobile phone.

3. How customers take car?

The customers take the car from my home and return it here too.

4. What are the data that you bring from customer?

Driving license original and Aadhar copy.

5. How do you Assure quality?

My car always in a good condition. If anything happened to my car before rent, I don't rent it. I will rent it only after completion of all work.

6. How would you rate your car?

600 Rs for first 150 km and 10 Rs extra for each km after 150km.

7. Do you provide discount?

Yes, provide discount for only my friends.

8. What's about fuel?

Fuel is the responsibility of customer.

9. If anything happened to your car when it is in the hand of customer. How do you respond?

It's very simple I will take compensation. And I will not give my car to that customer if it was only his fault.

10. What methods you opt for payment?

We opt online payments more. We also have option for direct payments.

## **3.2 SYSTEM SPECIFICATION**

### **3.2.1 Hardware Specification**

Processor - Intel i3

RAM - 4 G B

Hard disk - 5 0 0 G B

### **3.2.2 Software Specification**

Front End - HTML, CSS

Backend - MYSQL,

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, jQuery, PHP, CSS, Flask

## **3.3 SOFTWARE DESCRIPTION**

### **3.3.1 PHP**

PHP is a server-side scripting language used for both web development and general-purpose programming. PHP is now used by 2.1 million web servers and more than 244 million webpages. The reference version of PHP, which Rasmus Ledorf created in 1995, is now created by the PHP group. PHP: Hypertext Preprocessor is the current meaning of the recursive acronym PHP, which previously stood for personal home page. The PHP processor module on a web server translates the PHP code to create the finished web page. Instead of calling an external file, PHP commands can be directly put into an HTML source file to handle data. The GNU General Public License is incompatible with PHP because it has evolved to incorporate a command-line interface feature and can be used independently due to restrictions on the usage of the term PHP (GPL). Most web

servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems.

### 3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open-Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- MySQL is a database management system.

MySQL, a popular SQL database management system that is utilized extensively, is created, distributed, and maintained by Oracle Corporation, the software development company. These systems are critical to computing as they enable efficient processing of large quantities of information, either as standalone applications or integrated into other software.

- MySQL databases are relational.

Relational databases utilize separate tables to efficiently store data, giving access to objects like databases, tables, views, rows, and columns while also supplying a versatile programming environment. To maintain consistency and prevent issues like duplicates or missing data, rules can be established to manage relationships between different data fields. These relationships can be defined as one-to-one, one-to-many, unique, mandatory, optional, or pointers across other tables. SQL is the most commonly used language for accessing databases, and it can be input explicitly or embedded into other programming languages. An API specific to the programming language can also be employed to mask the SQL syntax. The SQL standard has undergone several iterations since its inception in 1986, with versions such as SQL92, SQL:1999, and SQL:2003 being named after specific years. Most often, the most recent version is referred to as "the SQL standard".

- MySQL software is Open-Source.

Open-source software, like MySQL, means that it is freely accessible to anybody and that anyone can make changes to it. Everyone is welcome to use it, and they have the freedom to edit the source code as they see fit. But there are some limitations imposed by the GPL on what can and cannot be done with the application. Developers can get a licensed version by buying it from the MySQL team if the GPL license is inappropriate for a specific use case or if a commercial application has to integrate with MySQL technology. The MySQL Licensing Overview may be used to get more specific information on MySQL licensing.

## CSS

CSS, or Cascading Style Sheets, is the language that regulates how web pages appear visually, including components like colors, styles, and fonts. Because of its versatility, the presentation may

be tailored for many devices, including tiny mobile screens, enormous desktop monitors, and even printers. Additionally, CSS can be used with any XML-based markup language and is not just limited to HTML.

## **JS**

JavaScript is a computer language that may be used for both the client and server sides of web development. It employs text-based programming. Its main objective is to enrich web pages with interactive components to increase user engagement. On the other hand, HTML and CS concentrate on giving websites structure and style. A website's level of interactivity is constrained by HTML's sole support for static content and CSS's handling of a website's aesthetic design. JavaScript, however, may be used to incorporate dynamic components that significantly improve a website's aesthetic appeal and interaction, hence increasing user engagement and appeal.

## **HTML**

HTML (Hypertext Markup Language) is the code used to structure and organize the content of webpage. With HTML, web developers can use various formatting options such as paragraphs, bullet point lists, graphics, and data tables to present information in an aesthetically pleasing and organized manner. By using these tools, developers can create a user-friendly and easy-to-navigate website for their audience

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. Design is a creative process. A good design is the key to a system that works effectively. "Design" is the process of using many approaches and concepts to thoroughly outline a process or a system so that it can be physically implemented. The process of using several approaches and concepts to specify a tool, a procedure, or a system in sufficient detail to enable its physical actuality is one way to put it. Software design serves as the technical foundation of the software engineering process, regardless of the development paradigm used. The system design process results in the development of the architectural details required to create a system or product. This programme has undergone the best design phase, which involves fine-tuning all efficiency, performance, and accuracy levels. During the design process, a document that is user-oriented is transformed into a document for programmers or database professionals. System design goes through two stages: logical design and physical design.

## 4.2 UML DIAGRAM

A common language known as UML is used to design, visualize, build, and document the software system artefacts. The Object Management Group (OMG) was responsible for developing UML, and a draught of the UML 1.0 definition was presented to the OMG in January 1997. Unified Modelling Language is known as UML. Other programming languages like UML, C++, Java, and COBOL are distinct from it. A graphical language called UML is used to create software plans. An objective visual modelling language for software visualization, expression, design, and documentation is what is known as UML. Although UML is mostly used for modeling software systems, it is not limited to this. It is also used as a model instead of software.

For example, the process flow in the production room, etc. UML is not a programming language, it is a tool that can be used to generate codes in different languages using UML diagrams. UML is directly related to analysis and design. After some development, UML became the OMG standard. The UML diagram describing a system is completed using all of the elements and relationships. The crucial step in the process is visualizing the UML diagram. The remaining information is used to finish.

The following nine diagrams are part of UML.

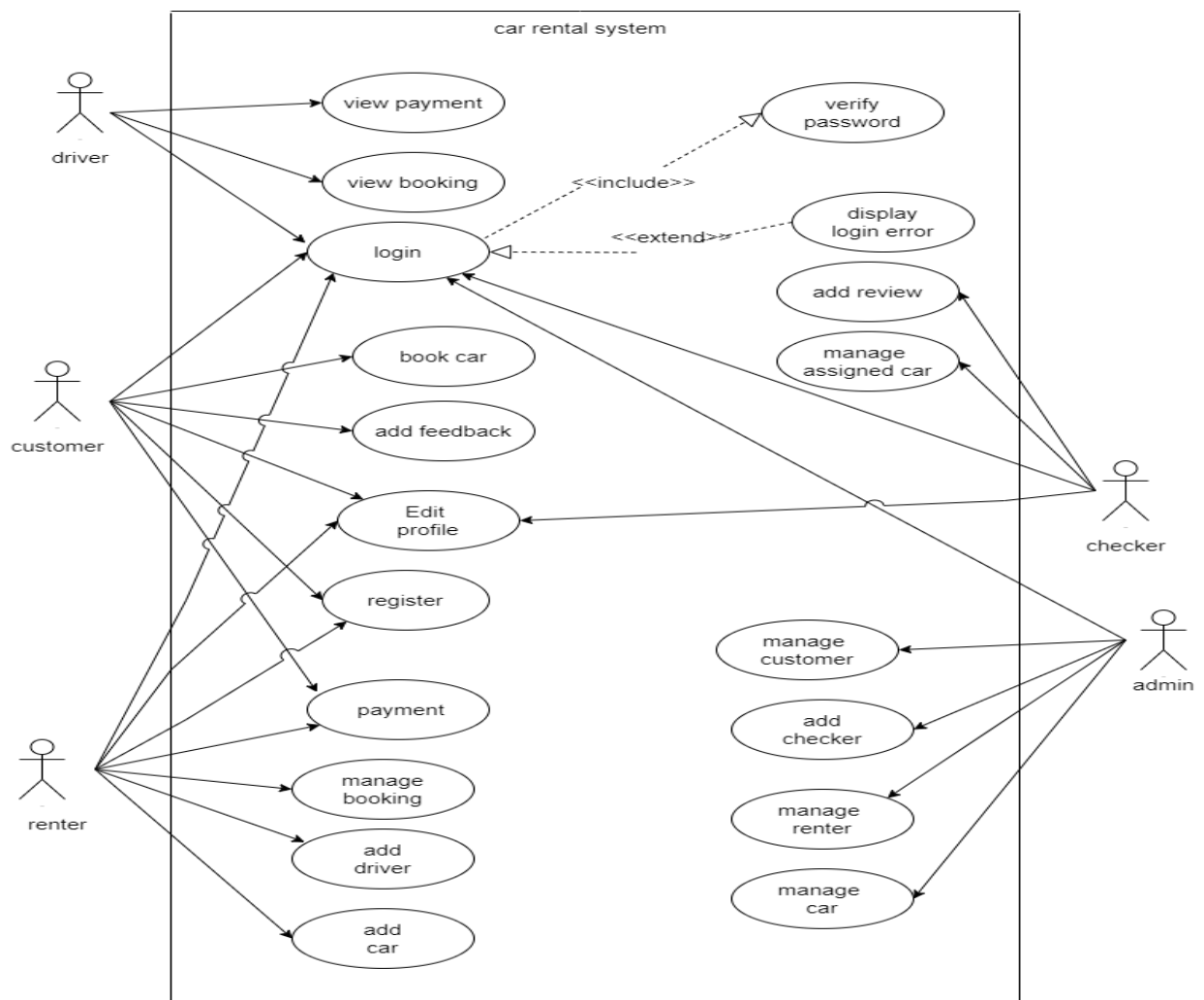
- Class diagram
- Object diagram
- Use case diagram



- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

### 4.2.1 USE CASE DIAGRAM

A use case is a method used in systems analysis to identify, clarify, and organize requirements. In this context, the word “system” refers to something that is being created or run, such as mail orders and web services. UML (Unified Modeling Language), a standard notation for modeling objects and the real world, uses diagrams. The purpose of the process will include preparation of all requirements, verification of hardware design, testing and maintenance of software under development, creation of online support documents or working in customer service. For example, use cases in a sales environment include orders, catalog updates, payments, and customers.



## 4.2.2 SEQUENCE DIAGRAM

The sequence diagram describes the interaction of objects in the order, that is, in the order in which these interactions occur. We can also refer to the map as a segment using the terms event map or event event. A system diagram describes how and in what order the components in the body work. These charts are widely used by marketers and software developers to document and understand the needs of new and existing systems.

Sequence Diagram Symbol —

**Actors** – An actor in a UML diagram represents a type of actor that interacts with the system and its objects. It should be noted that the actors are always outside the system we want to model using the UML diagram. We use actors to describe various characters, including human users and other external subjects. We use stickman symbols to represent actors in UML diagrams. We can have many players in a sequence of images. Diagram.

**Lifelines** – Still life is a noun that identifies an actor in a series of paintings. So basically, every instance in the diagram is represented by life. Lifeline content is at the top of the chart.

**Messages** – Communication between objects is explained in words. Words appear alive. We use arrows to represent words. Lifelines and design words are the basis of the diagram. Messages can be divided into the following two categories::

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

**Guards** – We use guards in UML to model the model. They are used when we need to limit the flow of words according to the relevance of the situation. Supervisors are crucial in ensuring that software developers are aware of the constraints imposed by a given process or system.

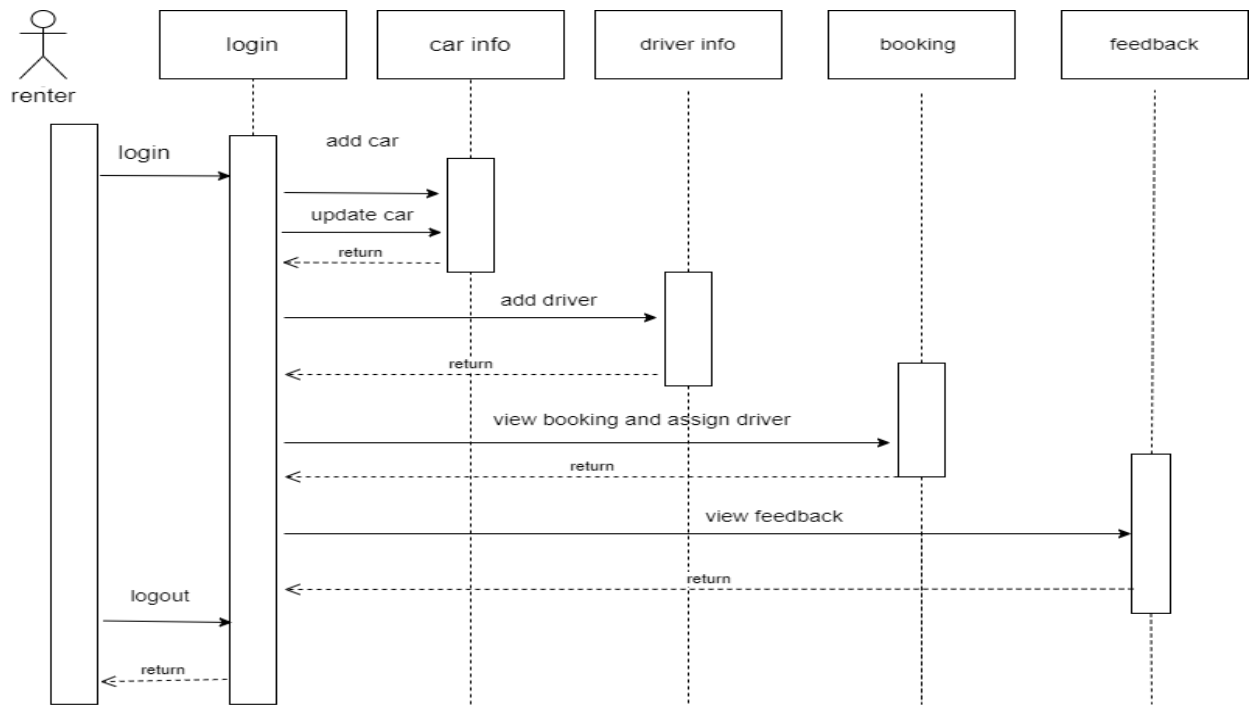
### Sequence diagram applications

They are also used to display specifics of UML use case diagrams, as well as to describe and

visualize the logic underlying complex functions, operations, or procedures.

- Used to comprehend the precise operation of present or upcoming systems.
- Imagine the flow of information between different system elements or objects.

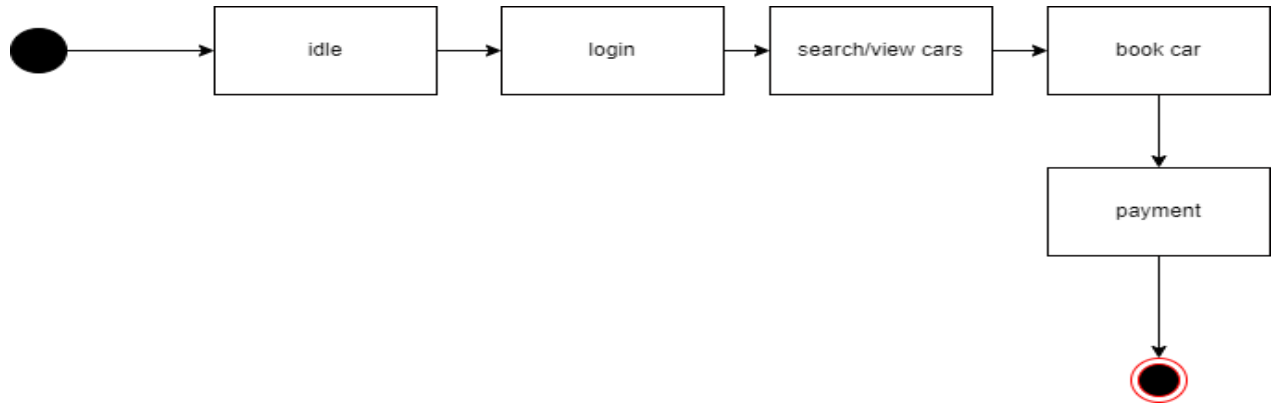
### Renter module



### 4.2.3 State Chart Diagram

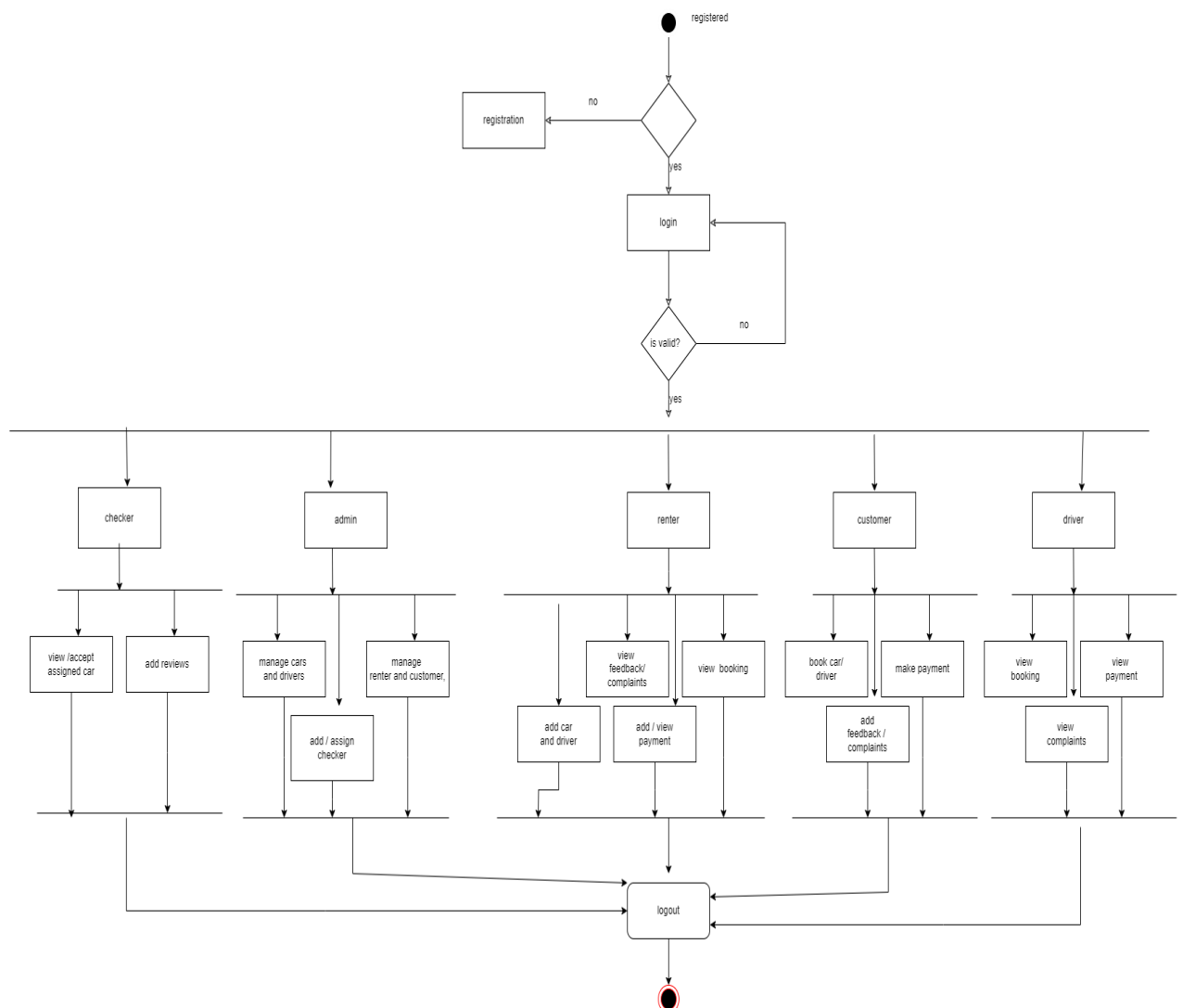
A State Chart Diagram is a type of UML (Unified Modeling Language) diagram used to model the behavior of a system or object over time. A State Chart Diagram shows the various states that an object or system can be in, and how it transitions from one state to another. Each state represents a particular condition or mode that the object or system can be in, and may have associated actions or behaviors. Transitions between states are triggered by events, which may be external (such as a user input) or internal (such as a timer). A transition may also have a guard condition, which must be true for the transition to occur. State Chart Diagrams can be used to model complex systems with many states and transitions, such as user interfaces, control systems, or business processes. State Chart Diagrams can also show hierarchical states, where a state may contain sub-states with their own transitions and behaviors. State Chart Diagrams are often used in conjunction with other UML diagrams, such as Use Case Diagrams or Class Diagrams. State Chart Diagrams can be used to communicate system behavior to stakeholders, including developers, testers, and users. State Chart Diagrams can be implemented using various programming languages and tools, such as finite state machines or event-driven programming. State Chart Diagrams can help improve system reliability and maintainability by providing a clear and concise model of system behavior that can

be used for testing, debugging, and maintenance.



#### 4.2.4 Activity Diagram

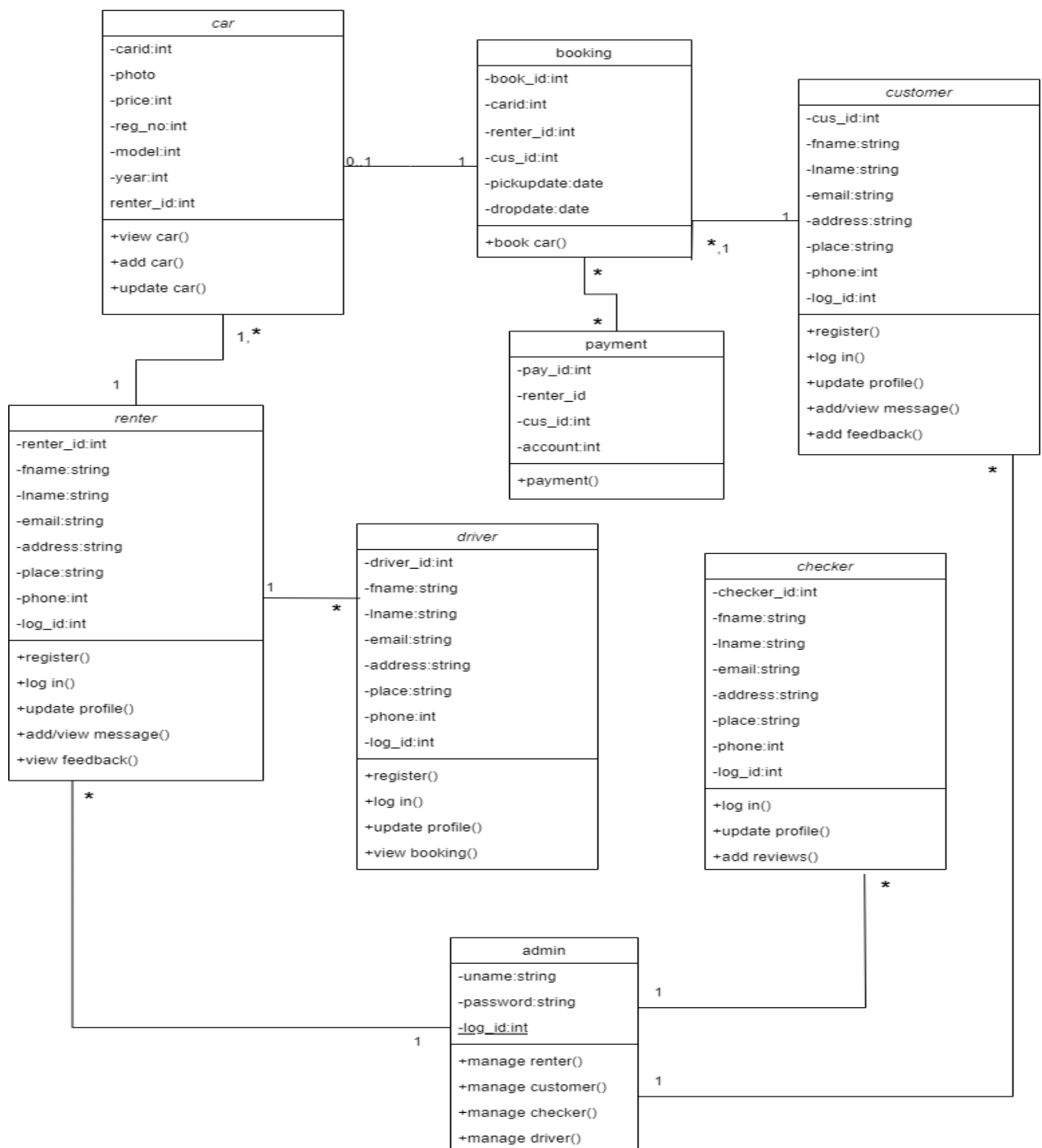
We use diagrams to show the control flow in the system and to refer to the steps that affect the operation of the application. We use diagrams to model the array and the array. So, we begin to explain the job visually using graphs. Diagrams focus on the nature of processes and the systems in which they occur. We use game maps to explain or explain what led to a particular event.



### 4.2.5 Class Diagram

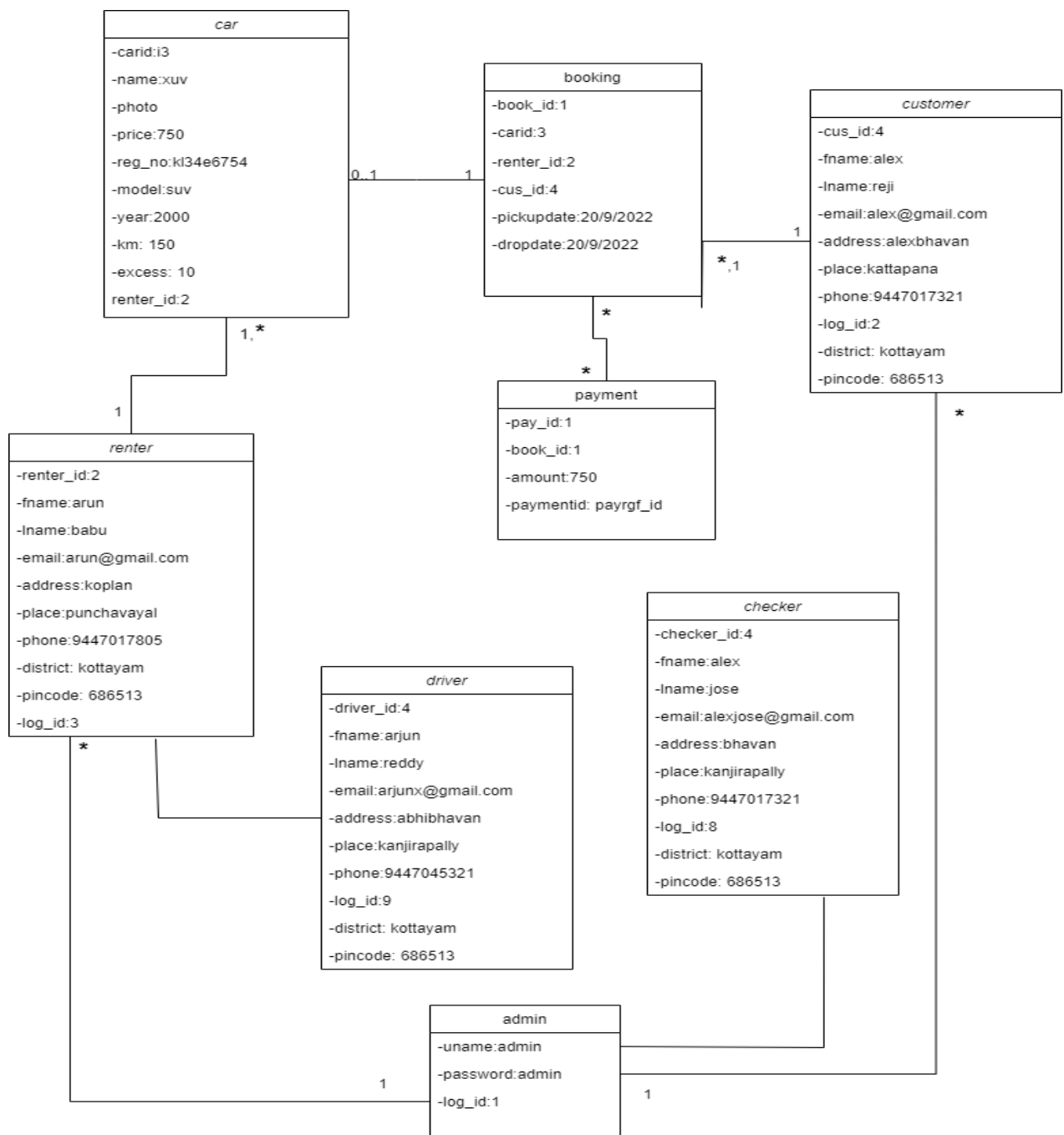
Graphs are static graphs. Represents the static view of the application. Class diagrams are not only used to visualize, explain, and document the different languages of the system, but are also used to generate the execution code of the software application.

Class diagrams describe the properties and functions of classes and constraints in the system. Class diagrams are commonly used to model object orientation, as they are the only UML diagrams that can be drawn directly in the routing language. The diagram shows collection classes, intersections, integrations, collaborations, and constraints. Also known as an art form.



### 4.2.6 Object Diagram

A chart represents an instance of a chart. The basic concepts of art and art are the same. The diagram also represents a static view of the system, but this static view is a snapshot of the system at a particular time. A diagram is used to illustrate a group of objects and their relationships as an example. The intended use of the image should be clearly understood. Charts are used in the same way as charts. The difference is that the diagram represents an abstract model of classes and their relationships. However, a painting represents a sample of a particular time and is authentic in nature. This means that the graph is close to the actual system behavior. The goal is to capture a static view of the system at a given time.

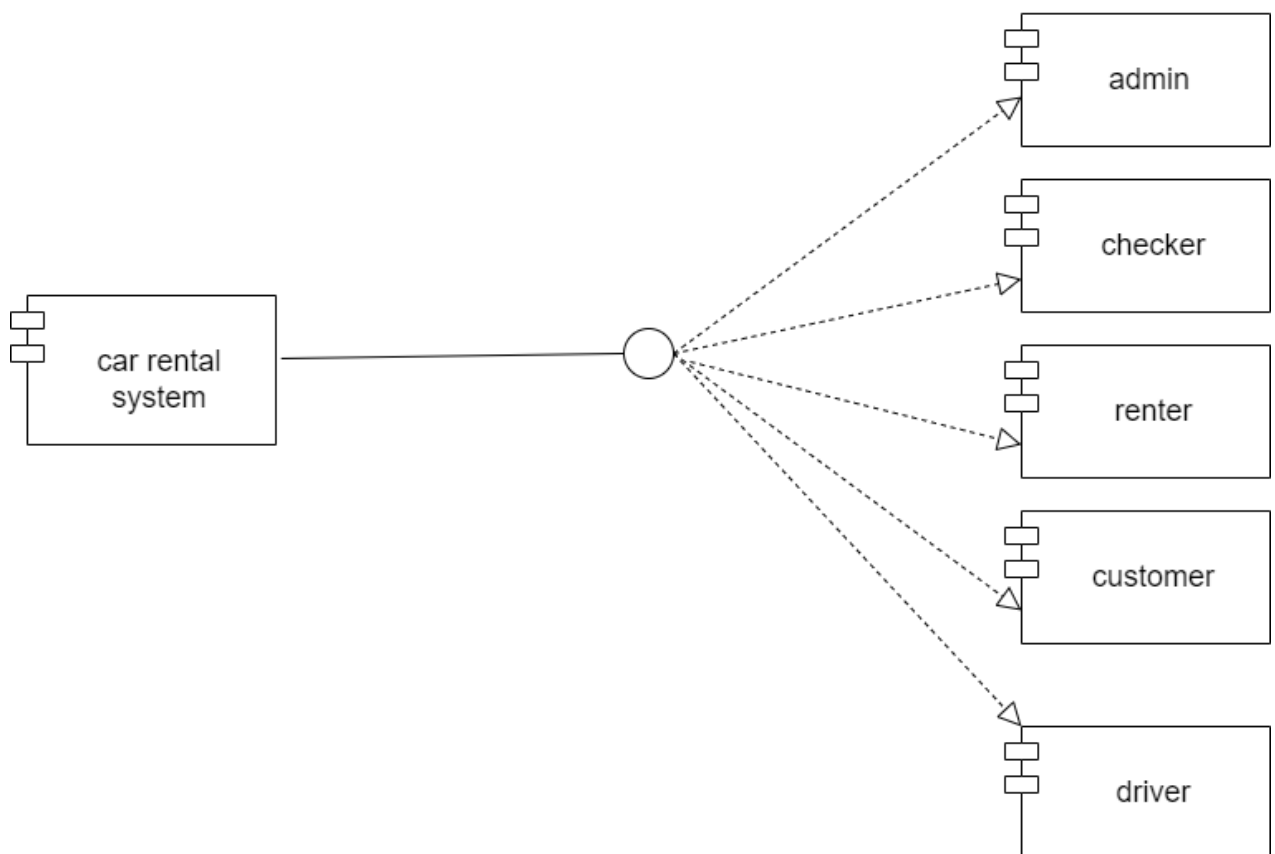


### 4.2.7 Component Diagram

UML diagrams are used to model the body of object-oriented systems used to visualize, clarify and document as a process and to create a complete process from front and back work. Diagrams are important in the diagramming class, which focuses on systems that are often used to model the static operation of a system.

Graphics destroy real systems designed to multitask. Each role is responsible for a specific purpose in the whole process and only interacts with other key points to know.

#### component diagram

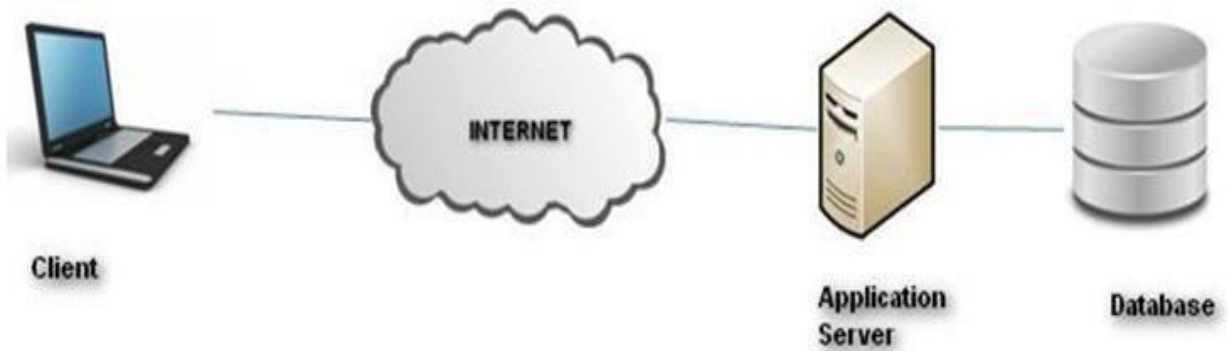


### 4.2.8 Deployment Diagram

A scatterplot is a type of UML diagram that shows the entire operation of a system, including nodes, such as hardware or software operating environments, and the intermediate objects that connect them.

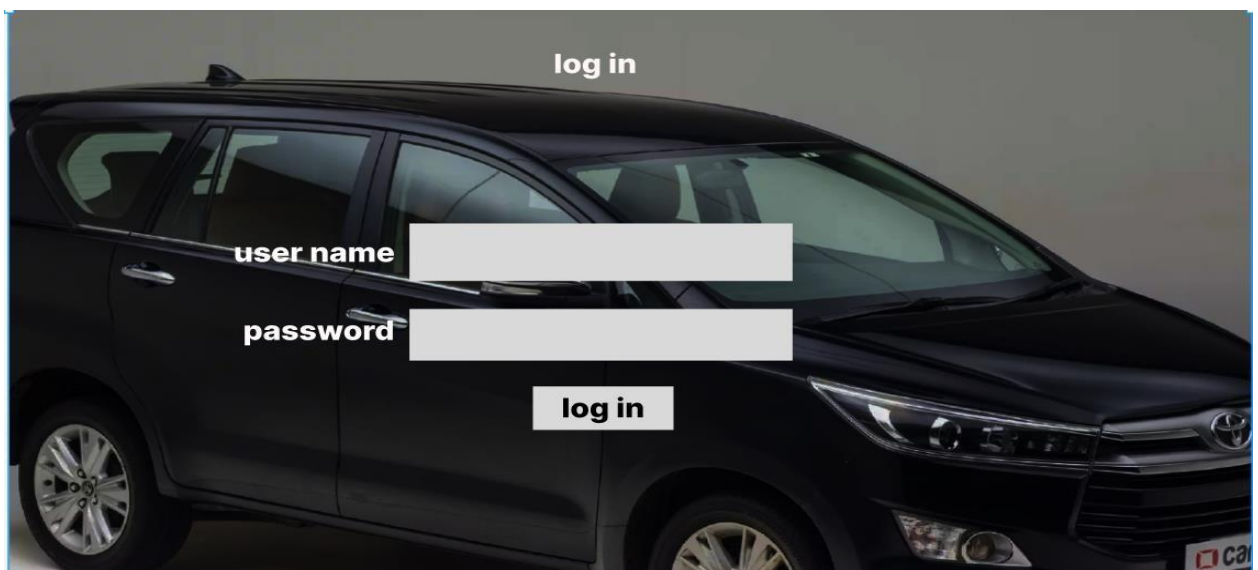
The export image is often used to show the hardware and software of the system. Using it, you can understand how the system will use physical resources.

Scatter diagrams help model the hardware topology of a system compared to other types of UML diagrams that typically describe physical details.



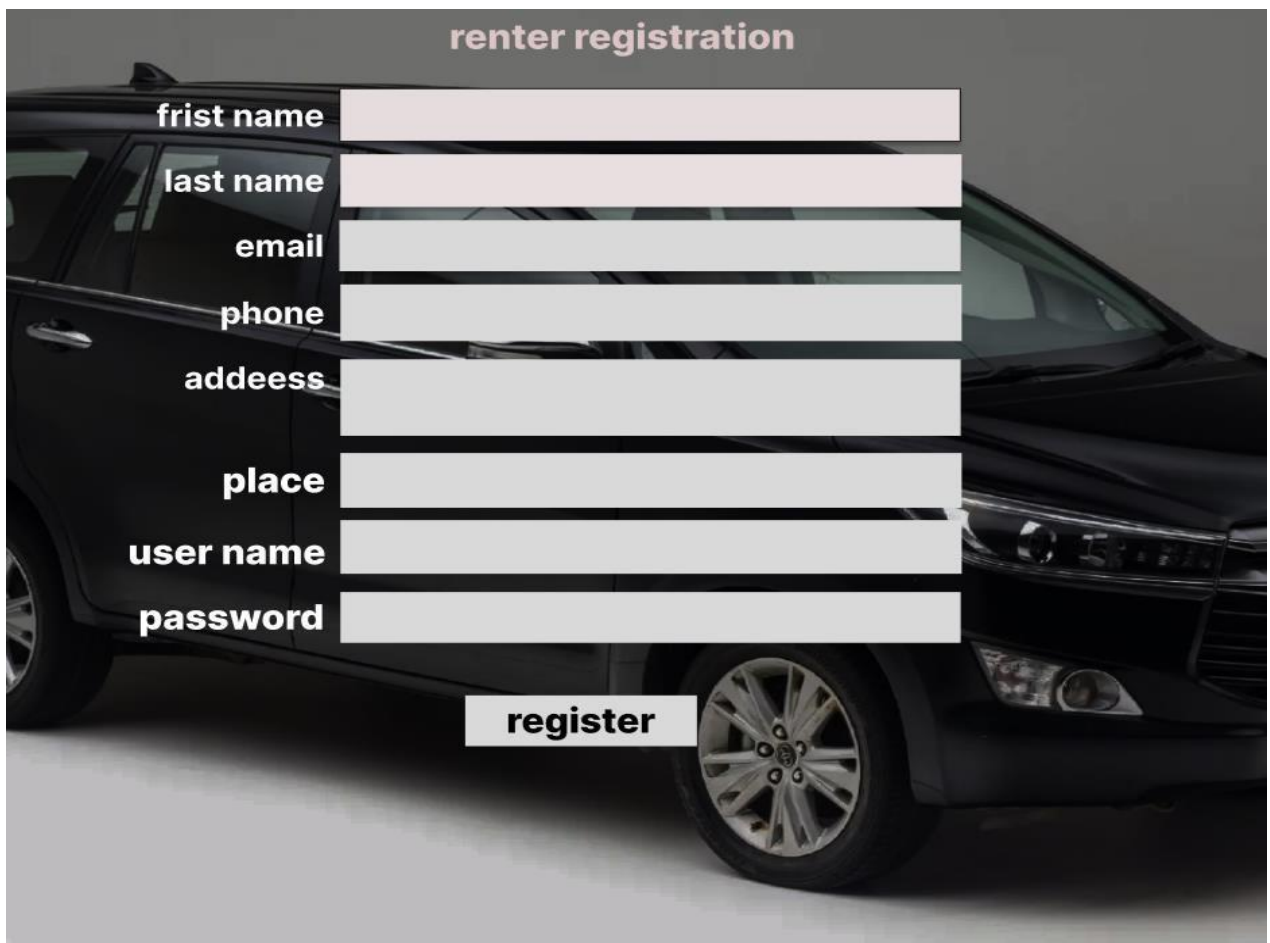
## 4.3 USER INTERFACE DESIGN USING FIGMA

### 4.3.1 Form Name: login





#### 4.3.2 Form Name: registration



**renter registration**

frist name

last name

email

phone

addeess

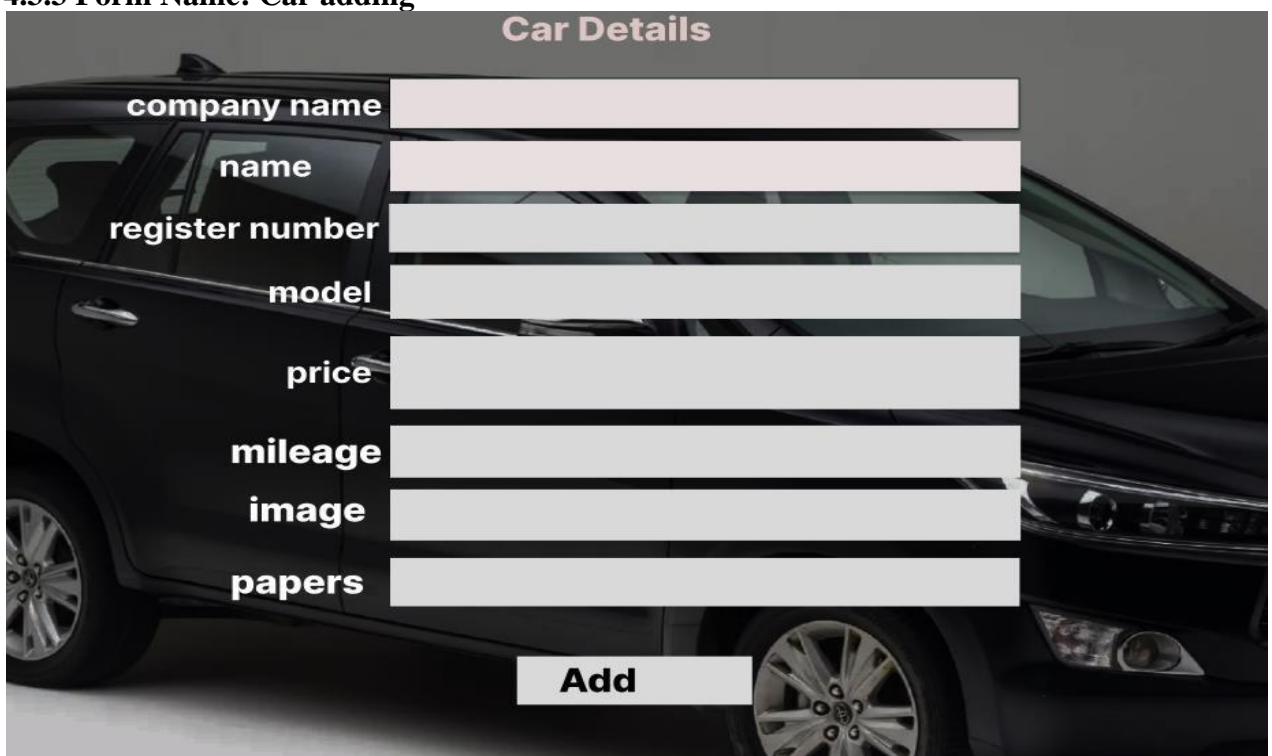
place

user name

password

**register**

#### 4.3.3 Form Name: Car adding



**Car Details**

company name

name

register number

model

price

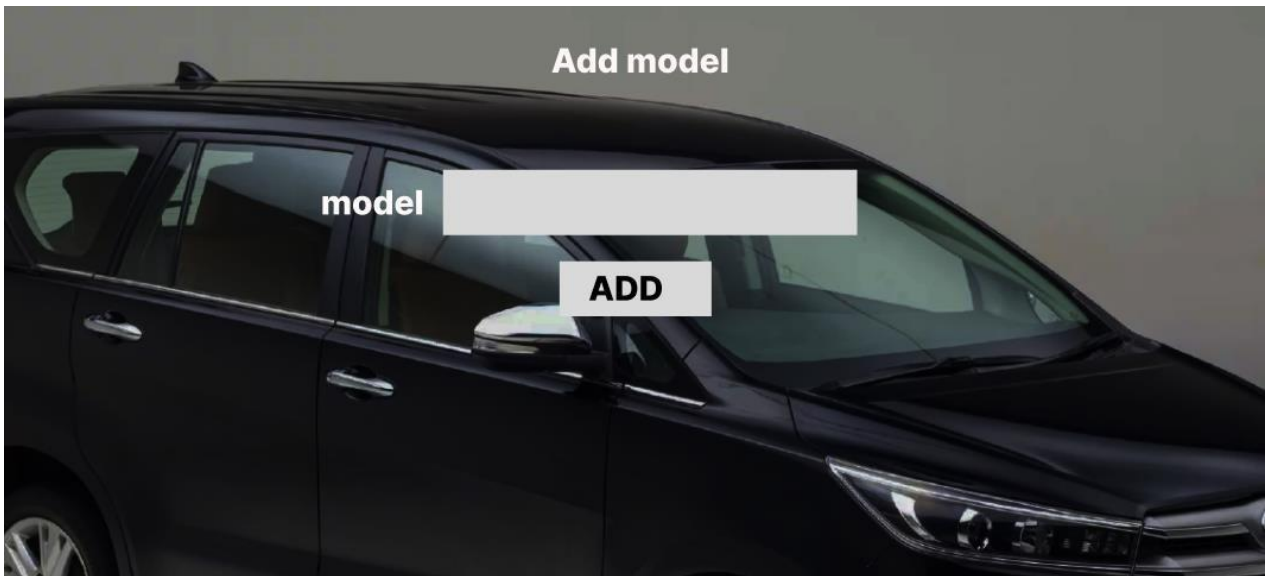
mileage

image

papers

**Add**

#### 4.3.4 Form Name: Add model



### 4.4 DATABASE DESIGN

A database is a process with the ability to store information to retrieve stored information from users in a quantity and quality. Data is the purpose of any database and must be protected. A database is created in two steps. Write out the user's needs in order to build a document that satisfies them as precisely as you can. This process, known as data-level creation, is separate from each DBMS.

This data-level design is converted into a design for the specific DBMS being used to carry out the queries in the second stage. The details of the exact DBMS to be used are included in this step, which is referred to as physical layer design. System design and database design are complementary processes. A database's informational organization serves two basic objectives.

- Data Integrity
- Data independence

#### 4.4.1 Relational Database Management System (RDBMS)

Data is represented via a relational model as a set of relationships. Every relationship is like a table of values or datasets. In relational model terminology, rows are called tuples, column headings are called attributes, and tables are relations. A relational file consists of tables, each of which is given a unique name. A row in the chart represents a group of related values.

#### 4.4.2 Normalization

The table is a relationship. Rows in a table are called tuples. A tuple is an ordered collection of n elements. Lines are called attributes. Relationships are established between all tables in the database. This ensures the integrity of information and the relationship between organizations.

Domain D is a set of atomic charges. One way to clarify the author is to determine the file type by extracting the main file that created the name. It's also helpful to give a field a name to help define its value. All values in a relationship are atomic and cannot be parsed.

### **1st Normal Form (1NF)**

- If a table's atomicity is 1, it is said to be in its first normal form.
- Atomicity dictates that a single cell cannot contain multiple values in this instance. It can only include one attribute with a single value.
- The multi-valued attribute, composite attribute, and their combinations are forbidden by the first normal form.

#### **4.4.2.2 Second Normal Form**

In relations when the primary key includes more than one attribute, no non-key attribute should be functionally dependent on a component of the main key. This entails creating a new relation for each partial key and dissecting it into its dependent properties. Keep your database's original primary key and any characteristics that are wholly dependent on it. Data that only depends on a small amount of the key can be removed with the help of this approach. A relation is said to be in second normal form if and only if the primary key satisfies all the criteria for first normal form and all of the relation's non-primary key properties are wholly dependent on the primary key alone.

#### **4.4.2.3 Third Normal Form**

According to the Third Normal Form, a relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a group of non-key attributes. In other words, the main key shouldn't be transitively dependent. The deconstructed non-key characteristics are then arranged in relation to the non-key qualities that they functionally determine. Anything not fully dependent on the primary key is removed using this process. A relation is only regarded as being in third normal form when it is in second normal form and, more significantly, when none of its non-key qualities depend on any other non-key features.

### **4.4.3 Sanitization**

Data sanitization entails removing or erasing data from a storage device on purpose and permanently to make sure it cannot be restored.

Normally, when data is removed from storage media, the medium is not truly wiped, and an attacker who gains access to the device can recover the data. Serious questions about security and data privacy are raised by this. Sanitization involves cleaning storage media so that no data remains on the device and that no data can be recovered even with cutting-edge forensic techniques.

#### **The Need for Data Sanitization**

As IT assets frequently retain sensitive business data after they are decommissioned due to the continued rise in the usable lifetime and storage capacity of storage equipment. These resources could be mobile devices, laptop disc drives, Flash media, dedicated storage equipment, and desktop and laptop PCs.

#### 4.4.4 Indexing

By reducing the number of disc accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes.

- The search key, which is the first column, contains a duplicate of the table's primary key or potential primary key. These values are kept in sorted order so that it is easy to get the corresponding data.

Note: The data may or may not be kept in sorted order.

- The address of the disc block on which that specific key value is stored is held by a series of pointers in the second column, which is designated as the Data Reference or Pointer.

### 4.5 TABLE DESIGN

#### 4.5.1. Tbl\_login

Primary key: **log\_id**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	log_id	int (11)	Primary key	Primary key for login table.
2	username	Varchar (50)	Not null	To store usernames for all users
3	passwd	Varchar (50)	Not null	To store passwords for user's login
4	usertype	int (11)	Not null	It indicates types for each user.
5	status	int (11)	Not null	It indicates the status of user where they are active or deactive

**4.5.2. Tbl\_customer**Primary key: **cus\_id**

Foreign key: log\_id references Tbl\_login

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	cus_id	int (11)	Primary key	Primary key for Tbl_customer table.
2	log_id	int (11)	Foreign key	Foreign key references from Tbl_login
3	fname	Varchar (50)	Not null	To store first name of customer
4	lname	Varchar (50)	Not null	To store last name of customer
5	email	Varchar (50)	Not null	To store email id of customer
6	phone	bigint (20)	Not null	To store contact number of customer
7	address	Varchar (50)	Not null	To store house name of customer.
8	Pincode	int (11)	Not null	To store pincode of customer.
9	District	Varchar (20)	Not null	To store district of customer.
10	place	Varchar (50)	Not null	To store place of customer
11	licence	Varchar (100)	Not null	To store licence of customer

**4.5.3. Tbl\_renter**

Primary key: **renter\_id**

Foreign key: log\_id references Tbl\_login

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	renter_id	int (11)	Primary key	Primary key for Tbl_renter table.
2	log_id	int (11)	Foreign key	Foreign key references from Tbl_login
3	fname	Varchar (50)	Not null	To store first name of renter
4	lname	Varchar (50)	Not null	To store last name of renter
5	email	Varchar (50)	Not null	To store email id of renter
6	phone	bigint (20)	Not null	To store contact number of renter
7	address	Varchar (50)	Not null	To store house name of renter
8	place	Varchar (50)	Not null	To store place of renter
9	district	Varchar (20)	Not null	To store district of renter.
10	pincode	int (11)	Not null	To store pincode of renter.
11	licence	Varchar (100)	Not null	To store licence of renter

**4.5.4. Tbl\_checker**

Primary key: **checker\_id**

Foreign key: log\_id references Tbl\_login

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	renter_id	int (11)	Primary key	Primary key for Tbl_checker table.
2	log_id	int (11)	Foreign key	Foreign key references from Tbl_login
3	fname	Varchar (50)	Not null	To store first name of checker
4	lname	Varchar (50)	Not null	To store last name of checker
5	email	Varchar (50)	Not null	To store email id of checker
6	phone	bigint (20)	Not null	To store contact number of checker.
7	home	Varchar (50)	Not null	To store house name of checker.
8	city	Varchar (50)	Not null	To store place of checker.
9	district	Varchar (20)	Not null	To store district of checker.
10	pincode	int (11)	Not null	To store pincode of checker.

#### 4.5.5. Tbl\_model

Primary key: **model\_id**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	model_id	int (11)	Primary key	Primary key for Tbl_model table.
2	model	Varchar (50)	Not null	To store car model name.

#### 4.5.6. Tbl\_car

Primary key: **car\_id**

Foreign key: renter\_id references Tbl\_renter and model\_id references Tbl\_model

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	car_id	int (11)	Primary key	Primary key for tbl_car table.
2	renter_id	int (11)	Foreign key	Foreign key references from Tbl_renter
3	model_id	int (11)	Foreign key	Foreign key references from Tbl_model
4	company	Varchar (100)	Not null	To store company name of car
5	name	Varchar (50)	Not null	To store name of car
6	year	int (11)	Not null	To store year of model car
7	reg_no	Varchar (50)	Not null	To store register number of car
8	price	bigint (20)	Not null	To store price of car for a limited km.
9	km	int (11)	Not null	To store km without excess price of car
10	Image	Varchar (300)	Not null	To store image of car



11	papers	Varchar (300)	Not null	To store papers of car
12	mileage	float	Not null	To store mileage of car per liter
13	C_stat	int (11)	Not null	To store status and avilability of car
14	excess	int (11)	Not null	To store excess price of the car after the specified km.
15	availability	Int (11)	Not null	To set the availability of the car.
16	sntscore	decimal (10,3)	Not null	To store the average sentiment score of the car.

#### 4.5.7. Tbl\_check

Primary key: **ch\_id**

Foreign key: car\_id references Tbl\_car and checker\_id references Tbl\_checker

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	ch_id	int (11)	Primary key	Primary key for tbl_check table.
2	car_id	int (11)	Foreign key	Foreign key references from Tbl_car
3	checker_id	int (11)	Foreign key	Foreign key references from Tbl_checker
4	details	Varchar (100)	Not null	To store details of the assigned car.
5	stat	int (11)	Not null	To store status of car

**4.5.8. Tbl\_driver**

Primary key: **driver\_id**

Foreign key: log\_id references Tbl\_login and renter\_id references Tbl\_renter

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	driver_id	int (11)	Primary key	Primary key for Tbl_driver table.
2	log_id	int (11)	Foreign key	Foreign key references from Tbl_login
3	renter_id	int (11)	Foreign key	Foreign key for Tbl_renter table.
4	fname	Varchar (50)	Not null	To store first name of driver.
5	lname	Varchar (50)	Not null	To store last name of driver.
6	email	Varchar (50)	Not null	To store email id of driver.
7	phone	bigint (20)	Not null	To store contact number of driver.
8	address	Varchar (50)	Not null	To store house name of driver.
9	place	Varchar (50)	Not null	To store place of driver.
10	district	Varchar (20)	Not null	To store district of driver.
11	pincode	int (11)	Not null	To store pincode of driver.
12	licence	Varchar (100)	Not null	To store licence of driver
13	dim	Varchar (100)	Not null	To store image of the driver.
14	availability	int (11)	Not null	To store availability of driver

**4.5.9. Tbl\_temp**Primary key: **id**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	id	int (11)	Primary key	Primary key for Tbl_temp table.
2	email	Varchar (50)	Not null	To store email address of the user.
3	token	int (11)	Not null	To store otp number of users.

**4.5.10. Tbl\_regpay**Primary key: **rpay\_id**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	rpay_id	int (11)	Primary key	Primary key for Tbl_regpay table.
2	name	varchar (50)	Not null	To store name of user.
3	email	Varchar (50)	Not null	To store email address of the user.
4	Pay_id	Varchar (50)	Not null	To store payment id.
3	amount	int (11)	Not null	To store paid amount.

**4.5.11. Tbl\_booking**

Primary key: **booking\_id**

Foreign key: car\_id references Tbl\_car and cus\_id references Tbl\_customer

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	book_id	int (11)	Primary key	Primary key for Tbl_booking table.
2	car_id	int (11)	Foreign key	Foreign key references from Tbl_car
3	cus_id	int (11)	Foreign key	Foreign key references from Tbl_customer
4	Book_date	date	Not null	To store pick up date for car
5	Drop_date	date	Not null	To store drop-off date for car
6	stat	int (11)	Not null	To store status of booking. That is booking accepted or not.
7	drive_stat	int (11)	Not null	To store status of driver that is driver is in this booking or not.
8	start_km	bigint (20)	Not null	To store starting km of the car.
9	end_km	bigint (20)	Not null	To store ending km of the car.
10	amount	int (11)	Not null	To store amount of car.
11	damount	int (11)	Not null	To store amount of driver.

**4.5.12. Tbl\_dbook**

Primary key: **dbook\_id**

Foreign key: carbook\_id references Tbl\_booking and driver\_id references Tbl\_driver

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	dbook_id	int (11)	Primary key	Primary key for Tbl_dbook table.
2	carbook_id	int (11)	Foreign key	Foreign key references from Tbl_booking
3	driver_id	int (11)	Foreign key	Foreign key references from Tbl_driver
4	stat	int (11)	Not null	To store the status of booking.

**4.5.13. Tbl\_carpay**

Primary key: **pay\_id**

Foreign key: book\_id references Tbl\_booking

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	pay_id	int (11)	Primary key	Primary key for Tbl_carpay table.
2	book_id	int (11)	Foreign key	Foreign key references from Tbl_booking
3	amount	int (11)	Not null	To store amount paid.
4	pdate	date	Not null	To store the date when the amount paid.
5	payment_id	varchar (11)	Not null	To store the payment id of the payment.

**4.5.14. Tbl\_driverpay**Primary key: **pay\_id**

Foreign key: dbook\_id references Tbl\_dbook

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	pay_id	int (11)	Primary key	Primary key for Tbl_driver table.
2	dbook_id	int (11)	Foreign key	Foreign key references from Tbl_dbook
3	amount	int (11)	Not null	To store amount paid.
4	pdate	date	Not null	To store the date when the amount paid.
5	payment_id	varchar (11)	Not null	To store the payment id of the payment.

**4.5.15. Tbl\_feedback**Primary key: **fid**

Foreign key: cus\_id references Tbl\_customer and car\_id references Tbl\_car and book\_id references table Tbl\_booking

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	fid	int (11)	Primary key	Primary key for Tbl_feedback table.
2	cus_id	int (11)	Foreign key	Foreign key references from Tbl_customer
3	car_id	int (11)	Foreign key	Foreign key references from Tbl_car
4	book_id	int (11)	Foreign key	Foreign key references from Tbl_booking
5	feedback	Varchar (200)	Not null	To store feedback about car.
6	score	decimal (10,3)	Not null	To store sentiment score of the feedback.

**4.5.16. Tbl\_complaint**

Primary key: **cp\_id**

Foreign key: book\_id references Tbl\_booking

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	cp_id	int (11)	Primary key	Primary key for Tbl_complaint table.
2	book_id	int (11)	Foreign key	Foreign key references from Tbl_booking
3	from_id	int (11)	Not null	To store id of user who write the complaint.
4	to_id	int (11)	Not null	To store id of complaint receiver.
5	from	Varchar (10)	Not null	To store the type of user who write complaint.
6	to	Varchar (10)	Not null	To store the type of user who receive the complaint.
7	complaint	Varchar (100)	Not null	To store the complaint.

## **CHAPTER 5**

### **SYSTEM TESTING**



## 5.1 INTRODUCTION

Software testing is a detailed process of examining software performance to determine whether it functions as intended. This process, also referred to as verification and validation, involves assessing a product to ensure it complies with applicable requirements and fulfills user needs. Various methods, such as reviews, analyses, inspections, and walkthroughs, are used to achieve software testing, along with related procedures like static analysis. There are two approaches to evaluating software: static analysis, which examines source code without execution, and dynamic analysis, which monitors software behavior during operation and generates data such as execution traces, timing profiles, and test coverage statistics. Testing is a collection of tasks that can be pre-organized and carried out systematically. For computer-based systems, the testing process starts with individual modules and progresses through system integration. There are several regulations that may be utilized as testing goals, and testing is required for the goals of the system testing to be effective. Those are:

Running a programmer via testing is how errors are found.

- A test case that has a strong possibility of spotting an undiscovered flaw is a good one.
- A test that identifies an unknown error is successful.

When a test is conducted according to its objectives, it can expose software defects while also demonstrating that the software features are working as per the given specifications and that the performance requirements have been met.

There are three ways to test program.

- For implementation efficiency
- For correctness
- For the complexity of computing

Ensuring the accuracy of a program can be a daunting task, particularly for larger programs, as it involves confirming that it operates precisely as intended

## 5.2 TEST PLAN

A test plan outlines the procedures required to execute various testing techniques, including the activities that need completion. Software developers hold the responsibility of creating computer programs, documentation, and data structures. To ensure that each component of the program functions as intended, it is necessary to conduct individual testing. An independent test group (ITG) is available to provide feedback to developers and identify any issues that may arise. The testing objectives must be quantified and included in the test plan. These objectives may include

metrics such as mean time to failure, cost of defect correction, residual defect density, frequency of occurrence, and the number of hours required for regression testing.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

### **5.2.1 Unit Testing**

Unit testing is a crucial stage of software verification that concentrates on testing individual components or modules, which are the fundamental building blocks of software design. This testing process entails reviewing the component-level design specifications to detect any errors within the module's boundaries by analyzing key control paths. The complexity of the test and the untested areas are determined during the process of unit testing, which is designed to be white-box focused, and multiple components can be tested simultaneously. To ensure proper functionality of the program unit being tested, the modular interface undergoes checks to ensure correct data flow in and out. Additionally, the integrity of temporarily stored data in the local data structure is examined throughout the algorithm's execution. The evaluation of boundary conditions confirms that every statement within the module has been executed at least once. Finally, inspection of each error management path is conducted to ensure proper error handling. Performing data flow tests across module interfaces prior to any other testing is crucial for ensuring the effectiveness of the testing process. This is because if data cannot flow in and out of the system correctly, all other tests will be useless. Error handling channels must be established, and fault scenarios must be anticipated during the design stage to ensure that the system can redirect or stop working when an error occurs. The final stage of unit testing is boundary testing, which involves testing the software at its boundaries. This is because software often fails at its boundaries.

### **5.2.2 Integration Testing**

Integration testing is a rigorous procedure that entails creating the structure of a program and executing tests to identify any problems related to the connections between different components. The objective is to establish a program structure based on design, utilizing components that have undergone unit testing. The overall program is then tested as a whole. However, correcting any issues can be challenging since the program's size makes it difficult to isolate the root causes. Once these errors are corrected, new ones may emerge, and the process may seem to repeat itself endlessly. After completing unit testing on all modules, they are integrated into the system to verify

that there are no interface inconsistencies. This integration process also leads to the development of a distinctive program structure, as any discrepancies in the program structures are eliminated.

### **5.2.3 Validation Testing or System Testing**

After conducting the testing phase, the entire system, including its code, modules, and class modules, was thoroughly examined using a process known as system tests or black box testing. Black box testing specifically aims to ensure that the software's functional requirements are met by creating input conditions that replicate all possible scenarios. The objective of this testing procedure is to detect and address a range of issues spanning from inadequate or inaccurate functionalities, interface discrepancies, errors in data arrangement or external data retrieval, performance drawbacks, initialization malfunctions, to termination glitches.

### **5.2.4 Output Testing or User Acceptance Testing**

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The program me should consult the user and the perspective system while it is being developed in order to make any necessary adjustments. This was accomplished in regards to the following areas:

- Input Screen Designs
- Output Screen Designs

A variety of test data are used to conduct the aforementioned tests. The process of system testing requires the preparation of test data. After the preparation of sample data, the system being analyzed is put to the test using that data. Test data errors are discovered once more and resolved using the testing techniques mentioned above when the system is tested. Additionally, the fixes are noted for future use.

### **5.2.5 Automation Testing**

Test automation, a software testing method, uses automated testing tools to run a predetermined set of test cases. In contrast to manual testing, where a person performs the test procedures, automation testing programs are capable of analyzing both expected and actual results, generating detailed reports, and inputting test data into the System Under Test.

Implementing software test automation is a costly endeavor that requires a substantial investment of both financial and human resources. However, the benefit of automating the testing process is that the same test suite can be repeatedly used in subsequent development cycles, with the help of test automation tools that record and replay the suite as necessary. When the test suite is fully

automated, human involvement is no longer required.

### 5.2.6 Selenium Testing

Selenium Testing is a selected collection of different software solutions, each aiming to support test automation in a unique way. This extensive collection of tools demonstrates their capacity to meet the requirements for testing all different kinds of web applications. Flexibility and Selenium Testing are really similar. It is essential for contrasting expected test results with an application's real behavior. It offers a range of options for finding UI elements. In essence, Selenium Testing is a crucial mechanism for supporting the execution of tests across many browser platforms.

#### Example:

##### Test Case 1

##### Code

```
import selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
import time
PATH= 'C:\Program Files\chromedriver.exe'
service = Service(PATH)
driver = webdriver.Chrome(service=service)
driver.get("http://localhost/car/in.php")
driver.maximize_window()
driver.find_element(By.ID, "modalo").click()
driver.implicitly_wait(60)
time.sleep(2)
driver.find_element(By.ID, "username").send_keys("babu12")
time.sleep(2)
driver.find_element(By.ID, "password1").send_keys("1234")
time.sleep(2)
driver.find_element(By.ID, "submit").click()
expectedurl = "http://localhost/car/c-h.php"
currenturl=driver.current_url
if expectedurl == currenturl:
    print("Test Case Passed")
else:
    print("Testcase Failed")
driver.quit()
```

## Screenshot

```
login test x
C:\testing\Scripts\python.exe "C:/Users/ARUN BABU/PycharmProjects/testing/login test.py"
Test Case Passed
Process finished with exit code 0
```

## Test report

Test Case 1					
Project Name: Car Rental System					
Login Test Case					
Test Case ID: Test_1			Test Designed By: Arun Babu		
Test Priority (Low/Medium/High):			Test Designed Date: 12-05-2023		
Module Name: Login			Test Executed By: Nimmy Francis		
Test Title: Login test			Test Execution Date: 12-05-2023		
Description: Testing the login module					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		User should be able to login	User is navigated to the home page with successful login.	pass
2	Provide valid username	Username – babu12	User should be able to login	User is navigated to home page	pass
3	Provide valid password	Password-admin			
4	Click on login button				
Post-Condition: User is validated with database and successfully login into account.					

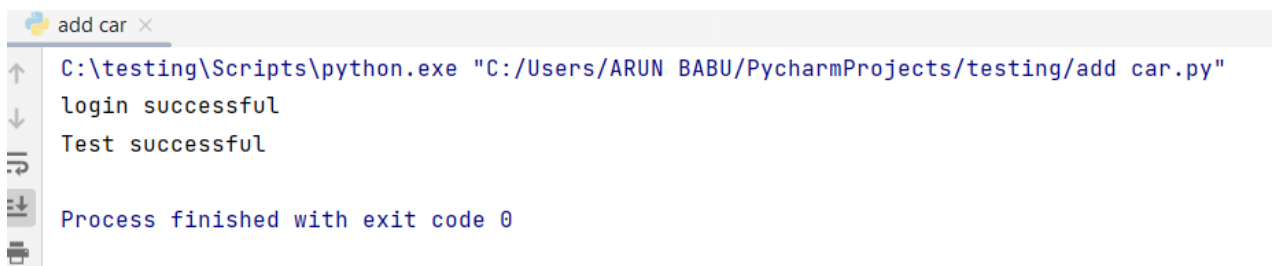
## Test Case 2

### Code

```
from selenium import webdriver
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select
import time
PATH= 'C:\Program Files\chromedriver.exe'
service = Service(PATH)
driver = webdriver.Chrome(service=service)
driver.get("http://localhost/car/in.php")
driver.maximize_window()
driver.find_element(By.ID,"modalo").click()
driver.implicitly_wait(1)
driver.find_element(By.ID,"username").send_keys("abhi")
driver.find_element(By.ID,"password1").send_keys("abhi")
driver.find_element(By.ID,"submit").click()
expectedurl ="http://localhost/car/renter-home.php"
currenturl=driver.current_url
if expectedurl == currenturl:
    print('login successful')
    time.sleep(4)
    driver.get('http://localhost/car/my-cars.php')
    time.sleep(1)
    driver.find_element(By.ID, "carmodal").click()
    driver.implicitly_wait(1)
    driver.find_element(By.ID, "company").send_keys("Hyundai")
    time.sleep(1)
    driver.find_element(By.ID, "cname").send_keys("Grand i10 Nios")
    time.sleep(1)
    driver.find_element(By.ID, "reg").send_keys("KL35H7865")
    time.sleep(1)
    upload_field = driver.find_element(By.ID,"cimage")
```

```
time.sleep(1)
upload_field.send_keys("C:/Users/ARUN BABU/Downloads/Hyundai-i10.jpg")
time.sleep(1)
driver.find_element(By.ID,"papers").send_keys("C:/Users/ARUN BABU/Downloads/gg.pdf")
time.sleep(1)
driver.find_element(By.ID, "mil").send_keys("11")
time.sleep(1)
driver.find_element(By.ID, "price").send_keys("800")
time.sleep(1)
driver.find_element(By.ID, "km").send_keys("150")
time.sleep(1)
driver.find_element(By.ID, "excess").send_keys("10")
time.sleep(1)
driver.find_element(By.ID, "year").send_keys("2019")
time.sleep(1)
select_box1 = Select(driver.find_element(By.ID, "car"))
select_box1.select_by_visible_text("suv")
driver.find_element(By.ID,"submit").click()
time.sleep(1)
alert = Alert(driver)
alert_text = alert.text
assert alert_text == "your car will confirm after checking"
if alert_text == "your car will confirm after checking":
    print("Test successful")
else:
    print("Test failed")
driver.quit()
```

### Screenshot



### Test report

Test Case 2					
Project Name: Car Rental System					
Add car Test Case					
Test Case ID: Test_2			Test Designed By: Arun Babu		
Test Priority (Low/Medium/High):			Test Designed Date: 12-05-2023		
Module Name: add car			Test Executed By: Nimmy Francis		
Test Title: Add car			Test Execution Date: 12-05-2023		
Description: Testing the add car page					
Pre-Condition: Renter has valid username and password.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page.		Renter should be able to login.	Renter login successfully.	Pass
2	Provide valid username and password	Username – abhi Password- abhi	Renter should be able to login and navigate to add car page and click on the add car button.	Renter login successfully and click on the add car button.	Pass
3	Navigate to add car page and click on add car button				
4	Provide car details.	Company- hyundai Car name – Grand i10 Registration number - KL35H7865			
5	Upload car image and papers.	Car image - Hyundai-i10.jpg Car paper- gg.pdf	Car details should be added.	Car added successfully.	Pass
6	Provide remaining car details.	Mileage – 11, Price – 800, Km – 150 Excess- 10, Year- 2019,Model - SUV			
7	Click on Submit button				
Post-Condition: Renter is validated with database and successfully login into account. Then car details successfully inserted into database.					



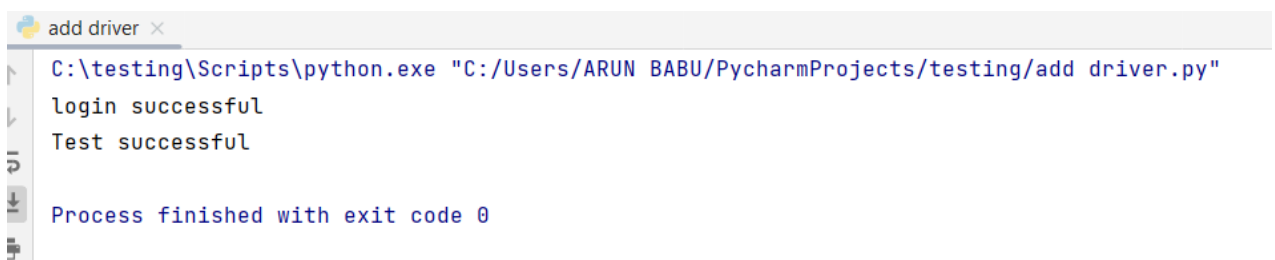
### Test Case 3

#### Code

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
import time
PATH= 'C:\Program Files\chromedriver.exe'
service = Service(PATH)
driver = webdriver.Chrome(service=service)
driver.get("http://localhost/car/in.php")
driver.maximize_window()
driver.find_element(By.ID,"modalo").click()
driver.implicitly_wait(1)
driver.find_element(By.ID,"username").send_keys("abhi")
driver.find_element(By.ID,"password1").send_keys("abhi")
driver.find_element(By.ID,"submit").click()
expectedurl ="http://localhost/car/renter-home.php"
currenturl=driver.current_url
if expectedurl == currenturl:
    print('login successful')
    time.sleep(4)
    button = driver.find_element(By.XPATH, "/html/body/div[2]/div/nav/div/div/a[3]")
    button.click()
    time.sleep(1)
    button2 = driver.find_element(By.XPATH, "/html/body/section[2]/div/div/button")
    button2.click()
    time.sleep(1)
    driver.find_element(By.ID, "myCheckbox").click()
    driver.implicitly_wait(1)
    driver.find_element(By.XPATH, "/html/body/section[2]/div/div/div[1]/div/div/div[4]/button")
    .click()
    driver.implicitly_wait(1)
    driver.find_element(By.ID, "dfname").send_keys("Benny")
    time.sleep(1)
```

```
driver.find_element(By.ID, "dlname").send_keys("jacob")
time.sleep(1)
driver.find_element(By.ID, "dem").send_keys("farij43553@appxapi.com")
time.sleep(1)
driver.find_element(By.ID, "dphn").send_keys("9767521234")
time.sleep(1)
driver.find_element(By.NAME, "daddresss").send_keys("Pullayil")
time.sleep(1)
driver.find_element(By.ID, "pincode").send_keys("686513")
time.sleep(5)
driver.find_element(By.ID, "dimage").send_keys("C:/xampp/htdocs/car/images/license.pdf")
time.sleep(1)
driver.find_element(By.ID, "dun").send_keys("benny")
time.sleep(1)
driver.find_element(By.NAME, "dpasswd").send_keys("benny")
time.sleep(1)
driver.find_element(By.XPATH, "/html/body/section[2]/div/div/div[2]/div/div/div[3]/input").c
lick()
time.sleep(3)
expectedurl2 = "http://localhost/car/regdriv.php"
currenturl2 = driver.current_url
if expectedurl == currenturl:
    print("Test successful")
else:
    print("Test failed")
driver.quit()
```

### Screenshot



```
add driver x
C:\testing\Scripts\python.exe "C:/Users/ARUN BABU/PycharmProjects/testing/add driver.py"
login successful
Test successful
Process finished with exit code 0
```

**Test report**

Test Case 3					
Project Name: Car Rental System					
Add driver Test Case					
Test Case ID: Test_3			Test Designed By: Arun Babu		
Test Priority (Low/Medium/High):			Test Designed Date: 12-05-2023		
Module Name: Add driver			Test Executed By: Nimmy Francis		
Test Title: Add driver			Test Execution Date: 12-05-2023		
Description: Testing the add driver section.					
Pre-Condition: Renter has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page. Provide username and password	Username – abhi Password - abhi	Renter should be able to login.	Renter login successfully.	Pass
2	Navigate to driver add page by click the button.		Renter should be able to login and navigate to add driver page and click on the add car button.	Renter login successfully and click on the add driver button.	Pass
3	Click on add driver button				
4	Provide Driver details	Fname - “Benny” Lame- “Jacob”			
5	Provide Driver details	Email-“farij43553@appxapi.com” Phone-8787876367 Address- Pullayil	Driver details should be added.	Driver added successfully.	Pass
6	Click on Submit button				
Post-Condition: Renter is validated with database and successfully login into account. Then driver details successfully inserted into database.					

**Test Case 4****Code**

```
from selenium import webdriver
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
import time
PATH= 'C:\Program Files\chromedriver.exe'
service = Service(PATH)
driver = webdriver.Chrome(service=service)
driver.get("http://localhost/car/in.php")
print('testing started')
driver.maximize_window()
driver.find_element(By.ID,"modalo").click()
driver.implicitly_wait(1)
driver.find_element(By.ID,"username").send_keys("babu12")
driver.find_element(By.ID,"password1").send_keys("1234")
driver.find_element(By.ID,"submit").click()
expectedurl ="http://localhost/car/c-h.php"
currenturl=driver.current_url
if expectedurl == currenturl:
    print('login successful')
    time.sleep(4)
    button = driver.find_element(By.XPATH,
"/html/body/section[2]/div[1]/div/div[1]/div/div[2]/form[1]/input[1]")
    button.click()
    time.sleep(2)
    driver.find_element(By.ID, "pdate").send_keys("11-05-2023")
    driver.find_element(By.ID, "ddate").send_keys("11-05-2023")
    driver.find_element(By.ID, "sub").click()
    time.sleep(1)
    alert = Alert(driver)
    alert_text = alert.text
```

```

assert alert_text == "your booking will confirm after checking"
if alert_text == "your booking will confirm after checking":
print("Test successfull")
else:
    print("Test failed")
driver.quit()

```

### Screenshot

```

book x
C:\testing\Scripts\python.exe "C:/Users/ARUN BABU/PycharmProjects/testing/book.py"
testing started
login successful
Test successfull
Process finished with exit code 0

```

### Test report

Test Case 4					
Project Name: Car Rental System					
Car booking Test Case					
Test Case ID: Test_4			Test Designed By: 12-05-2023		
Test Priority (Low/Medium/High):			Test Designed Date: Arun Babu		
Module Name: Car booking			Test Executed By: Nimmy Frencis		
Test Title: Car booking			Test Execution Date: 12-05-2023		
Description: Car booking page test case					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page. Provide username and password	Username – babu12 Password - 1234	Customer should be able to login.	Customer login successfully.	Pass
2	Navigate to car booking page		Booking added successfully	Booking added successfully.	Pass
3	Provide details	Pdate- “12-05-2023” Ddate-“12-05-2023”			

4	Click on submit button				
<b>Post-Condition:</b> Customer is validated with database and successfully login into account. Then booking details successfully inserted into database.					

## Test Case 5

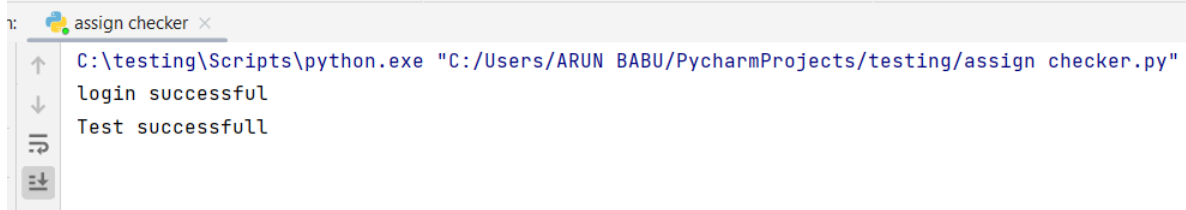
### Code

```

from selenium import webdriver
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select
import time
PATH= 'C:\Program Files\chromedriver.exe'
service = Service(PATH)
driver = webdriver.Chrome(service=service)
driver.get("http://localhost/car/in.php")
driver.maximize_window()
driver.find_element(By.ID, "modalo").click()
driver.implicitly_wait(1)
driver.find_element(By.ID, "username").send_keys("admin")
driver.find_element(By.ID, "password1").send_keys("admin")
driver.find_element(By.ID, "submit").click()
expectedurl = "http://localhost/car/admin/ad.php"
currenturl=driver.current_url
if expectedurl == currenturl:
    print('login successful')
    time.sleep(4)
    driver.find_element(By.ID, "cars").click()
    time.sleep(1)
    driver.find_element(By.ID, "acar").click()
    time.sleep(1)
    select_box1 = Select(driver.find_element(By.ID, "checker"))
    select_box1.select_by_visible_text("arjun rajeev")
    driver.find_element(By.NAME, "assign").click()
    time.sleep(1)
    alert = Alert(driver)
    alert_text = alert.text
    assert alert_text == "checker assigned"
    if alert_text == "checker assigned":
        print("Test successfull")
    else:
        print("Test failed")
driver.quit()

```

## Screenshot



```

C:\testing\Scripts\python.exe "C:/Users/ARUN BABU/PycharmProjects/testing/assign_checker.py"
login successful
Test successfull

```

## Test report

Test Case 5					
Project Name: Car Rental System					
Assign checker Test Case					
Test Case ID: Test_5			Test Designed By: Arun Babu		
Test Priority (Low/Medium/High):			Test Designed Date: 12-05-2023		
Module Name: Assign checker			Test Executed By: Nimmy Francis		
Test Title: Assign Checker			Test Execution Date: 12-05-2023		
Description: Checker assigned by admin to each car					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page. Provide username and password	Username – admin Password - admin	Admin should be able to login.	Admin login successfully.	Pass
2	Navigate to checker assign page		Checker assigned successfully	Checker assigned successfully	Pass
3	Choose checker	Checker- Arjun Rajeev			
4	Click on assign button				
Post-Condition: Admin is validated with database and successfully login into account. Then assign checker.					

## **CHAPTER 6**

# **IMPLEMENTATION**



## 6.1 INTRODUCTION

The execution phase of a project is a critical stage that involves converting the conceptual design into a fully functional and dependable system that meets the needs of its users. User documentation and training are the primary focus during this stage, which may coincide with the conversion process. The final stage of the transition from the old system to the new one is the implementation phase, and a poorly planned or managed implementation can lead to confusion and disruption, especially for the user department. The effectiveness of an implementation is determined by the quality of system testing and the level of staff training and preparation for the transition. The complexity of the system will dictate the amount of work required for system analysis and design, education and training, system testing, and changeover. Ultimately, the success of an implementation relies on the reliability and dependability of the system in meeting the organization's needs.

The following tasks are part of the implementation state:

- Careful preparation.
- A system and constraint investigation.

Designing strategies to implement the transition.

## 6.2 IMPLEMENTATION PROCEDURES

When we talk about software implementation, it encompasses the entire process of setting up and customizing a software program to meet its specific requirements in the designated environment, along with an evaluation of its functionality and ability to support the applications it was designed for. This process involves ensuring that the software is compatible with the underlying hardware and software infrastructure, and that it can perform its intended tasks efficiently and effectively. Often, someone who will not be utilizing the program commissions the software development project.

### 6.2.1 User Training

The objective of user training is to equip users with the necessary skills and knowledge to test and modify the computer-based system with confidence, ultimately achieving the intended goals. The objective of user training is to equip users with the necessary skills and knowledge to test and modify the computer-based system with confidence, ultimately achieving the intended goals.

## **6.2.2 Training on the Application Software**

Before using the new application software, users must first undergo basic computer literacy training. They should be taught how to navigate through the software screens, access help resources, handle errors that may arise while inputting data, and update entered information. The training should also cover specific concepts relevant to the user group and their role in utilizing the system or its components. The program's training should be tailored to meet the needs of different user groups and hierarchy levels.

### **6.2.1 System Maintenance**

The maintenance phase of a software product's lifecycle plays a crucial role in ensuring that the system is well-maintained and can adapt to changes in its environment. This stage is critical in the software development life cycle since it allows the system to respond effectively to any modifications. Although identifying errors is an essential part of software maintenance, it is not the only aspect that requires attention.

### **6.2.2 Hosting**

When a web hosting provider allocates space on a web server for an online site, they are enabling the site to be accessible on the internet. This process of web hosting makes all the website files, including code and images, available for online viewing. The amount of space allocated to a website on a server depends on the type of web hosting. The most common types of web hosting include shared, dedicated, VPS, and reseller hosting. These types of hosting are distinguished by the technology used for the server, the level of service provided, and the additional services on offer.

#### **infinityFree**

Car Rental System is hosted on infinityFree website, which is a popular web hosting service provider. infinityFree offers a range of features like a user-friendly control panel, free SSL certificate, and automatic backups, making it an ideal choice for hosting web applications. The system is configured to work seamlessly with the hosting environment, ensuring optimal performance and security. Additionally, infinityFree provides 24/7 customer support, which ensures that any issues related to hosting are addressed promptly.

**Procedure for hosting a website on infinityFree:**

**Step 1:** Create an account on infinityFree by visiting their website and clicking on the "Sign Up" button.

**Step 2:** Click "Create Account" after logging in to your account to create a new website.

**Step 3:** Choose a name for your website, write a subdomain name, and then choose the appropriate domain extension and click the create button.

**Step 4:** Now we get into a new page and click on to the "Open Control Panel" and agree the terms and conditions.

**Step 5:** To upload your project files, click on the "File Manager" button, select the files you want to upload, and click on the "Upload" button.

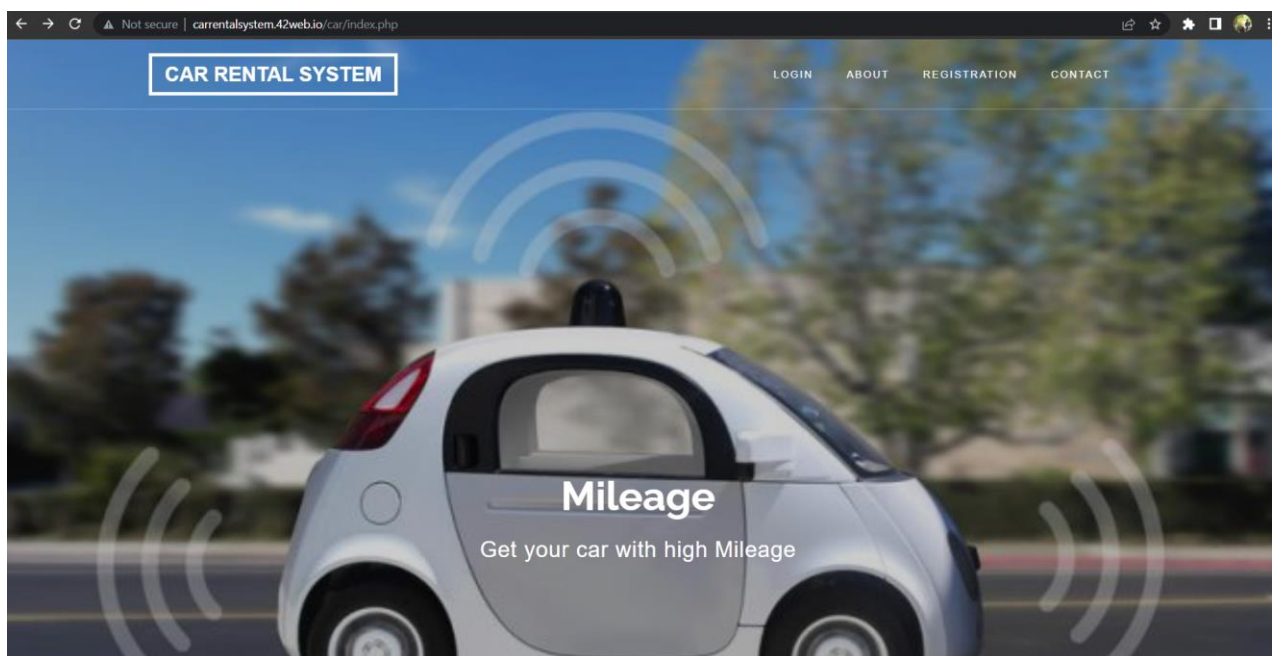
**Step 6:** To create database go to mysql databases option and create database and import your database into it.

**Step 7:** Then change the configuration php code in file manager with the username, password, server name according to the infinityFree web server.

**Step 8:** Then you can publish your website by using the subdomain name that we already choose.

**Hosted Website:**

**Hosted Link:** <http://carrentalsystem.42web.io/car/>

**Screenshot**

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## 7.1 CONCLUSION

The project entitled “Car Rental System” have been developed to help people in different parts of Kerala to rent various kinds of car. It brings different renters across the city to rent different types of cars and the customers can take cars for rent according to their needs. The customers also have an option to choose drivers, it’s an optional case. This project helps renters to manage their cars and drivers. By providing a streamlined process for booking and renting cars, such a system can improve efficiency, reduce costs, and enhance the overall rental experience. Using HTML as the front end and PHP as back end, along with MySQL as the database, the system was developed and tested with all possible samples of data. The performance of the system is provided to be efficient and can meet all the requirements of the user. Thus, providing a useful website to rent a car.

## 7.2 FUTURE SCOPE

- The current system is confined only to the website later on it can be extended to an android application for user convenience.
- Keep track of previously rented cars of each customer and provide suggestions based on their history.
- Customer can extent their booking dates if they need that car after drop-off date.
- Customers can add a car to their wish list for future use.
- Keep tracking the cars by implementing IOT devices.

## **CHAPTER 8**

### **BIBLIOGRAPHY**

**REFERENCES:**

- Saurabh Pal, “System Analysis and Design”, 2017
- R.L. Glass, I. Vessey, V. Ramesh, “Research in software engineering: an analysis of the literature”, 2002
- PankajJalote, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

- <https://www.w3schools.com/php/>
- <https://stackoverflow.com/questions/9436534/ajax-tutorial-for-post-and-get>
- <https://getbootstrap.com/docs/4.0/components/modal/>
- <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

## **CHAPTER 9**

### **APPENDIX**



## 9.1 Sample Code

### Booking page

```
<?php
include("config.php");
include("session.php");
$sql = "SELECT * FROM car";
$result = $conn->query($sql);
$log_id = $_SESSION['log_id'];
$sql34 = "SELECT * FROM customer where log_id='$log_id'";
$result34 = $conn->query($sql34);
$row34 = $result34->fetch_assoc();
$cus_id = $row34['cus_id'];
$licence = $row34['license'];
if (isset($_POST['sub'])) {
    $car_id = $_POST['car-id'];
    $pdate = $_POST['pdate'];
    $ddate = $_POST['ddate'];
    $stat = 2;
    $sql1 = "INSERT INTO `tbl_booking`(`cus_id`, `car_id`, `driver_id`, `book_date`,
`drop_date`, `stat`, `drive_stat`) VALUES ('$cus_id','$car_id','0','$pdate','$ddate','2','1')";
    if ($conn->query($sql1) === TRUE) { ?>
        <script>
            if (window.confirm('your booking will confirm after checking')) {
                window.location.href = 'c-h.php'; };
        </script> <?php
    } else { ?>
        <script>
            if (window.confirm('Oops!!!! failed ')) {
                window.location.href = 'c-h.php';    };
        </script>
    <?php
    }} ?>
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <title>Carbook - Free Bootstrap 4 Template by Colorlib</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link
href="https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,700,800&display=
swap" rel="stylesheet">
  <link rel="stylesheet" href="css2/open-iconic-bootstrap.min.css">
  <link rel="stylesheet" href="css2/animate.css">
  <link rel="stylesheet" href="css2/owl.carousel.min.css">
  <link rel="stylesheet" href="css2/owl.theme.default.min.css">
  <link rel="stylesheet" href="css2/magnific-popup.css">
  <link rel="stylesheet" href="css2/aos.css">
  <link rel="stylesheet" href="css2/ionicons.min.css">
  <link rel="stylesheet" href="css2/bootstrap-datepicker.css">
  <link rel="stylesheet" href="css2/jquery.timepicker.css">
  <link rel="stylesheet" href="css2/flaticon.css">
  <link rel="stylesheet" href="css2/icomoon.css">
  <link rel="stylesheet" href="css2/style.css">
  <script>
    window.onload = function() {
      var input = document.getElementById("pdate");
      var currentDate = new Date();
      input.setAttribute("min", currentDate.toISOString().slice(0, 10));
      input.onchange = function() {
        doSomething();
      };
    }
    function doSomething() {
      var input1 = document.getElementById("ddate");
      var userDate = document.getElementById("pdate");
      var userDateValue = new Date(userDate.value);
      input1.setAttribute("min", userDateValue.toISOString().slice(0, 10));
    } </script>
```

```

</head>
<body>
    <!-- Topbar Start -->
    <div class="container-fluid bg-dark py-3 px-lg-5 d-none d-lg-block">
        <div class="row">
            <div class="col-md-6 text-center text-lg-left mb-2 mb-lg-0"></div></div></div>
    <!-- Topbar End -->
    <!-- Navbar Start -->
    <div class="container-fluid position-relative nav-bar p-0">
        <div class="position-relative px-lg-6" style="z-index: 9;">
            <nav class="navbar navbar-expand-lg bg-secondary navbar-dark py-3 py-lg-0 pl-3 pl-lg-5">
                <a href="" class="navbar-brand">
                    <h1 class="text-uppercase text-primary mb-1">Car Rental</h1></a>
                    <button type="button" class="navbar-toggler" data-toggle="collapse" data-
                    target="#navbarCollapse"><span class="navbar-toggler-icon"></span></button>
                    <div class="collapse navbar-collapse justify-content-between px-3" id="navbarCollapse">
                        <div class="navbar-nav ml-auto py-0">
                            <a href="c-h.php" class="nav-item nav-link">Home</a>
                            <a href="c-h.php" class="nav-item nav-link">Cars</a>
                            <div class="nav-item dropdown">
                                <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown">My Bookings</a>
                                <div class="dropdown-menu rounded-0 m-0">
                                    <a href="view-book.php" class="dropdown-item">My Bookings</a>
                                    <a href="currentcbook.php" class="dropdown-item">Ongoing booking</a>
                                    <a href="carpaid.php" class="dropdown-item">Payment Done</a></div></div>
                                <div class="nav-item dropdown">
                                    <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"><svg
                                    xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-
                                    person-circle" viewBox="0 0 16 16"><path d="M11 6a3 3 0 1 1-6 0 3 3 0 0 1 6 0z" /><path fill-
                                    rule="evenodd" d="M0 8a8 8 0 1 1 16 0A8 8 0 0 1 0 8zm8-7a7 7 0 0 0-5.468 11.37C3.242 11.226
                                    4.805 10.8 10s4.757 1.225 5.468 2.37A7 7 0 0 0 8 1z" /></svg>
                                    <?php echo strtoupper($row34['fname']); ?> </a>
                                    <div class="dropdown-menu rounded-0 m-0">
                                        <a href="update-cus.php" class="dropdown-item">My profile</a>

```

```

<a href="logout.php" class="dropdown-item">Logout</a></div></div></nav></div></div>
<div class="hero-wrap ftco-degree-bg" style="background-image: url('images/bg_1.jpg');" data-
stellar-background-ratio="0.5">
<div class="overlay"></div>
<div class="container">
<div class="row no-gutters slider-text justify-content-start align-items-center justify-content-
center">
<div class="col-lg-8 ftco-animate">
<div class="text w-100 text-center mb-md-5 pb-md-5">
<h1 class="mb-4">Fast & Easy Way To Rent A Car</h1>
</div></div></div></div></div>
<section class="ftco-section ftco-no-pt bg-light">
<div class="container"><div class="row no-gutters">
<div class="col-md-12 featured-top"><div class="row no-gutters">
<div class="col-md-4 d-flex align-items-center">
<form action="" class="request-form ftco-animate bg-primary" method="post">
<h2>Make your trip</h2>
<div class="d-flex"><div class="form-group">
<label for="" class="label">Pick-up date</label>
<input type="date" class="form-control" placeholder="Date" name="pdate" id="pdate" required>
<span class="message text-danger" id="ms" style="font-size: 16px"></span><br></div>
<div class="form-group">
<label for="" class="label">Drop-off date</label>
<input type="date" class="form-control" placeholder="Date" id="ddate" name="ddate">
<span class="message text-danger" id="ms1"></span></div></div>
<?php
$car_id = $_POST['cid'];
$selectc = "SELECT * FROM `car`,`driver` WHERE car_id='$car_id' AND
car.renter_id=driver.renter_id and driver.availability=1";
$sqd = $conn->query($selectc);
if ($sqd->num_rows > 0) {
?>
<div class="form-group">
Do you need driver<input type="radio" id="age1" name="driver" value="1">

```

```

<label for="age1">Yes</label><input type="radio" id="age2" name="driver" value="0">
<label for="age2">No</label><br></div>
<?php } ?>
<div class="form-group">
<input type="hidden" name="car-id" value="<?php echo $car_id; ?>">
<input type="submit" value="Rent A Car Now" name="sub" id="sub" class="btn btn-secondary
py-3 px-4"></div></form></div>
<div class="col-md-8 d-flex align-items-center">
<div class="services-wrap rounded-right w-100">
<h3 class="heading-section mb-4"></h3> <div class="row d-flex mb-4">
<div class="col-md-4 d-flex align-self-stretch ftco-animate">
<div class="services w-100 text-center"></div>
<div class="col-md-4 d-flex align-self-stretch ftco-animate">
<div class="services w-100 text-center"></div>
<div class="col-md-4 d-flex align-self-stretch ftco-animate">
<div class="services w-100 text-center"></div></div></div></div></div>
</section>
<div class="container-fluid bg-dark py-4 px-sm-3 px-md-5">
<p class="mb-2 text-center text-body">&copy; <a href="#">CAR RENTAL SYSTEM</a></p>
</div>
<!-- Footer End -->
<!-- loader -->
<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px"
height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4"
stroke="#eeeeee" /><circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4"
stroke-miterlimit="10" stroke="#F96D00" /></svg></div>
<script src="js1/jquery.min.js"></script>
<script src="js1/jquery-migrate-3.0.1.min.js"></script>
<script src="js1/popper.min.js"></script>
<script src="js1/bootstrap.min.js"></script>
<script src="js1/jquery.easing.1.3.js"></script>
<script src="js1/jquery.waypoints.min.js"></script>
<script src="js1/jquery.stellar.min.js"></script>
<script src="js1/owl.carousel.min.js"></script>

```

```
<script src="js1/jquery.magnific-popup.min.js"></script>
<script src="js1/aos.js"></script>
<script src="js1/jquery.animateNumber.min.js"></script>
<script src="js1/bootstrap-datepicker.js"></script>
<script src="js1/jquery.timepicker.min.js"></script>
<script src="js1/scrollax.min.js"></script>

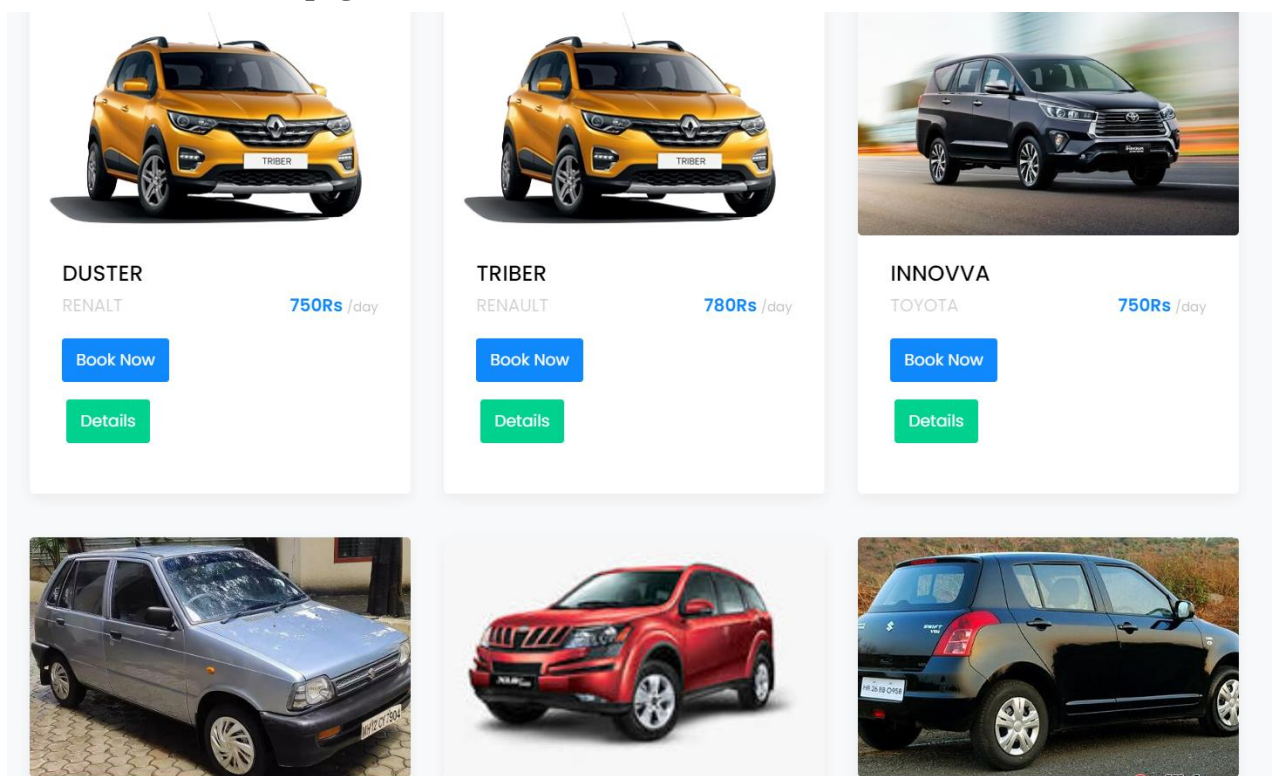
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7SQ74_uJiULgl1J
Y0H2s&sensor=false"></script>

<script src="js1/google-map.js"></script>
<script src="js1/main.js"></script>

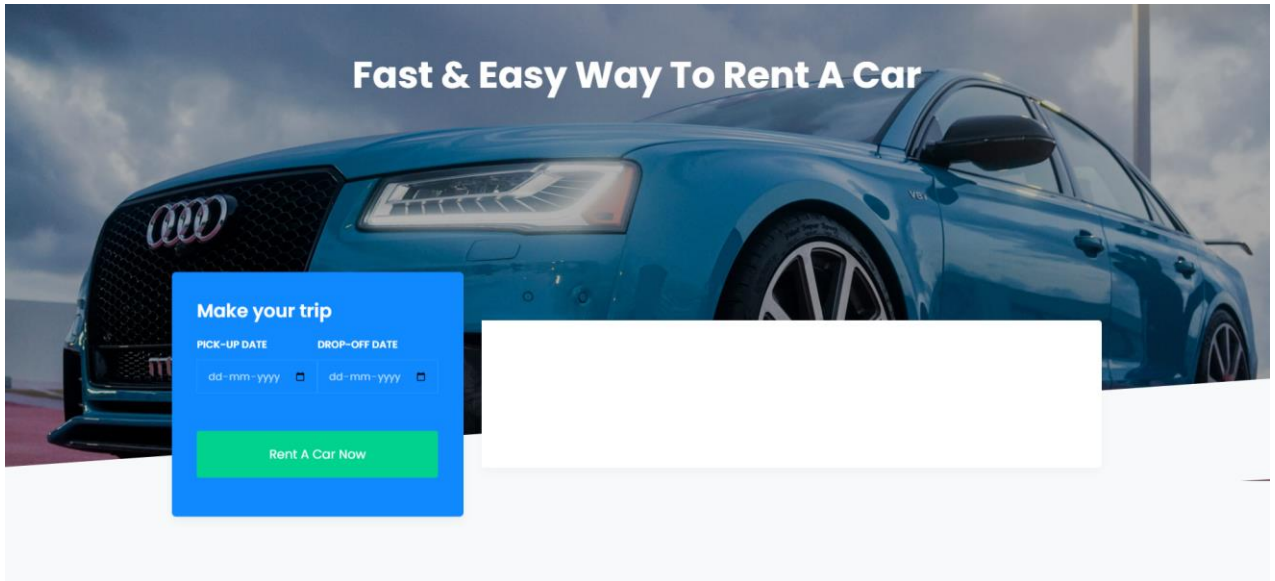
</body>
</html>
```

## 9.2 Screen Shots

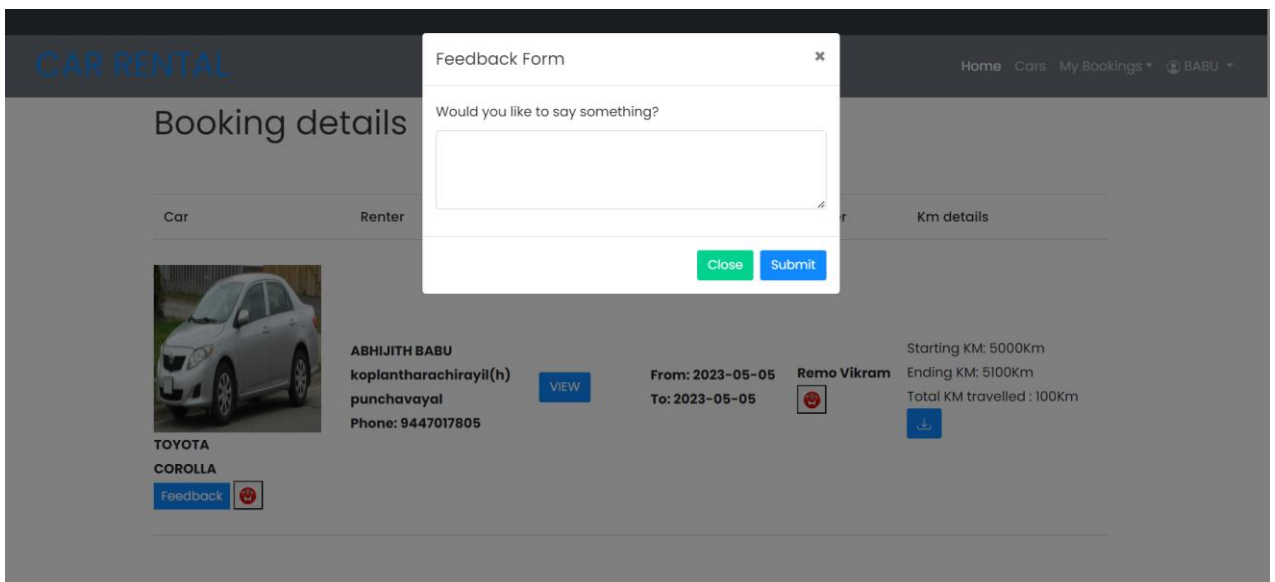
### 9.2.1 Customer home page



### 9.2.2 Booking page



### 9.2.3 Adding feedback



## 9.2.4 Profile page

My Profile Home

Profile Picture

**babu suresh**

Account Details

First name Last name

babu suresh

Email Phone number

babu@gmail.com 9888888888

Save changes

Account Details

User Name Enter Your Old Password

babu12 Enter Your Password

New Password Confirm Password

Password Confirm Password

Update Password

## 9.2.5 Bill generation

cuscarpaypdf.php

1 / 1 100%

**Car Rental**

PAGE DATE

1 2023-05-13

Payment Done By:

babu suresh  
Email : babu@gmail.com  
Phone: 9888888888

Payment To:

abhiijth babu  
Email : abhi@gmail.com  
Phone: 9447017805

MODE OF PAYMENT DATE OF PAYMENT PAYMENT Id

Online 2023-05-05 pay\_Llq45FxbHvplZS

NO	DESCRIPTION	Days	KM driven	Amount
1	This is your payment details	from: 2023-05-05 To : 2023-05-05	100KM	1589



### 9.2.6 Adding car

Car details please

Company Name

Car Name

Reg Number

Choose image...

Choose File No file chosen

Choose Car papers in pdf(polution,book,insurance)...

Choose File No file chosen

Mileage per Liter

Price

### 9.2.7 Adding Driver

Details please

First Name

Last Name

Email

Phone number


Address

Pincode


Select Post Office

submit Close

### 9.2.8 Driver paid section



## Booking details

Car	Customer	Date	Driver	Action
 <b>TOYOTA</b> <b>COROLLA</b>	<b>BABU SURESH</b> <b>koplantharachirayil(h)</b> <b>punchavayal</b> <b>Phone: 9888888888</b>	<b>From: 2023-05-05</b> <b>To: 2023-05-05</b>	<b>Remo Vikram</b>	Starting KM: 5000Km Ending KM: 5100Km Total KM travelled : 100Km Total Amount received: 1589₹ Total Amount paid: 800₹

### 9.2.9 Checker adding opinion about car

Mileage  
12km/l

Registration Number: kl05as3

Papers:

Commits - georgejacob1\_A

#### Add your Findings

All papers clear? : ☒ Yes ☐ No

Vehicle is in running condition? : ☒ Yes ☐ No

Over damage in car? : ☒ Yes ☐ No

Details about the car is true : ☒ Yes ☐ No

Is price given is apt?? : ☒ Yes ☐ No

It is apt for our website? : ☒ Yes ☐ No

Your Suggestion

Add your Findings about car

### 9.2.10 Admin adding checker

ADMIN ▾

Dashboard  
Customers  
renter  
Driver  
Model  
checker  
Cars

## Registration Form

First Name

Last Name

House Name/No

City

District

-Select your district-

Pincode

Phone Number

Email Address

User Name

Password

Submit

### 9.2.11 Add model

ADMIN ▾

Dashboard  
Customers  
renter  
Driver  
Model  
checker  
Cars


## Car Modals

Model

submit




Sl.No.	Model
1	suv
2	muv
3	sedan

### 9.2.12 Car view



ADMIN ▾

Renters

RENTER Name	COMPANY & CAR	REGISTER NO	Papers	MILEAGE	PRICE	IMAGE	ACTION
WARGOD DRAX	TOYOTA INNOVVA Year: 2000	KL05AS1752 MUV	<a href="#">VIEW</a>	1	First 150km : 750₹ Excess for each km : 10₹		<a href="#">Block</a>
MANU SUNU	MARUTI MARUTI 800 Year: 2010	KL34E4567 SEDAN	<a href="#">VIEW</a>	17	First 150km : 600₹ Excess for each km : 10₹		<a href="#">Block</a>
ABHIJITH	MAHINDRA	KL05AS1756	<a href="#">VIEW</a>	6	First 150km :		<a href="#">Block</a>

**Attach Plagiarism Report**