

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



**DBMS Mini Project Report
On**

Car Rental Shop Management System

Submitted in partial fulfillment for the award of degree of

**Bachelor of Engineering
in
Artificial Intelligence and Machine Learning**

Submitted by

Sourav N Ghorpade

1RN20AI052



RNS INSTITUTE OF TECHNOLOGY
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)
(UG programs – CSE, ECE, ISE, EIE and EEE are Accredited by NBA up to 30.6.2025)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

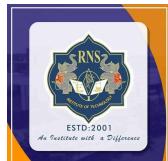
Department of AI & ML

2022 – 2023

RNS INSTITUTE OF TECHNOLOGY

(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)
(UG programs – CSE, ECE, ISE, EIE and EEE are Accredited by NBA up to 30.6.2025)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

Department of AI & ML



CERTIFICATE

Certified that the Project entitled **Car Rental Shop Management System** carried out by **Mr. Sourav N Ghorpade USN 1RN20AI052** a bonafide student of V Semester BE, **RNS Institute of Technology** in partial fulfillment for the Bachelor of Engineering in AI & ML ENGINEERING of the **Visvesvaraya Technological University**, Belagavi during the year 2022-23. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Database Management System with Mini Project Laboratory prescribed for the said Degree.

Course Teacher

Ms. Sajitha N

Asst. Professor

Department of AI&ML

RNSIT, Bengaluru

HoD

Dr. Harsha S

Department of AI & ML

RNSIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

ABSTRACT

The field of automobile industry is enormous in today's world and with technology that is constantly evolving paired with the deterioration of data management on an average scale around the world, the influx of data is rising on an exponential scale. This mini project produces a product that aims at organizing the aforementioned data and provides a way to easily access other data related to it. The Car Rental Shop Management System is a comprehensive solution for managing car rental operations. It offers a robust set of features and capabilities to manage car fleets, track customers, process payments, and provide comprehensive reporting. The system is designed to help rental shops manage their daily operations and maximize efficiency. It provides a comprehensive suite of tools for tracking and managing car fleets, customers, payments, and reports. The system also includes an integrated online booking system, allowing customers to make online reservations and manage their rental accounts from their own devices. The system also provides analytics and reporting tools to help rental shops better manage their operations and gain better insights into their performance. Additionally, the system offers an integrated customer service platform to help rental shops provide faster and better customer service. With its comprehensive feature set, the Car Rental Shop Management System is the perfect choice for rental shops looking to streamline their operations and improve their bottom line.

ACKNOWLEDGEMENT

At the very onset, we would like to place on record our gratitude to all those people who have helped us in making this project work a reality. Our Institution has played a paramount role in guiding us in the right direction.

We would like to profoundly thank **Sri. Satish R Shetty**, Chairman, RNS Group of Institutions. Bangalore for providing such a healthy environment for the successful completion of this project work.

We would also like to thank our beloved Principal, **Dr. M K Venkatesha**. for providing the necessary facilities to carry out this work.

We are extremely grateful to **Dr. Harsha S, Head of the Department** of Artificial Intelligence and Machine Learning, for having accepted to guide me in the right direction with all his wisdom.

We would like to express our sincere thanks to our Course Teacher **Ms. Sajitha N** Associate Professor, Department of Artificial Intelligence and Machine Learning for her constant encouragement that motivated us for the successful completion of this project work.

We would like to thank our guide, **Ms. Ashwini K**, Assistant professor, Department of Artificial Intelligence and Machine Learning for her continuous guidance and constructive suggestions for this Project work.

Last but not the least, we are thankful to all the teaching and non-teaching staff members of the Artificial Intelligence and Machine Learning Department for their encouragement and support throughout this work.

Sourav N Ghorpade

1RN20AI052

TABLE OF CONTENTS

ACKNOWLEDGEMENT	I
ABSTRACT	II
1. INTRODUCTION	
1.1 Overview of Database Management Systems	1
1.2 Problem Statement	2
1.3 Objectives	2
2. SYSTEM REQUIREMENTS	
2.1 Software & Hardware	3
3. SYSTEM DESIGN	
3.1 E R Diagram	4
3.2 Schema Diagram	5
3.3 Overview of GUI	5
3.4 Normalization	6
4. IMPLEMENTATION	
4.1 Table Creation	8
4.2 Description of Table	9
4.3 Populated Tables	10
4.4 SQL Triggers & Stored Procedures	11
4.5 Database Connectivity	12
4.6 Source code (Front End)	13
5. RESULTS	15
CONCLUSION & FUTURE ENHANCEMENTS	20
REFERENCES / BIBLIOGRAPHY	21

List Of Figures

Figure No.	Figure Name	Page No.
3.1	Entity Relationship diagram	4
3.2	Schema Diagram	5
4.1	Description Of Car Table	8
4.2	Description of Customer Table	9
4.3	Description of Rent Table	9
4.4	Populated Car Table	10
4.5	Populated Customer Table	10
4.6	Populated RTO Table	10
4.7	Populated Service Record Table	10
5.1	Login Form	15
5.2	Admin User Interface	16
5.3	Customer User Interface	16
5.4	Employee User Interface	17
5.5	Car Table Display and Data Entry	17
5.6	Customer Table Display and Data Entry	18
5.7	Service Table Display and Data Entry	18
5.8	RTO Table Display and Data Entry	19
5.9	Retail Table Display and Data Entry	19

Chapter 1

INTRODUCTION

1.1 Overview of Database Management System

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Any piece of information can be a data, for example name of your school. Database is actually a place where related piece of information is stored and various operations can be performed on it. A DBMS is a software that allows creation, definition and manipulation of database. DBMS is actually a tool used to perform any kind of operation on data in database. DBMS also provides protection and security to database. It maintains data consistency in case of multiple users. Here are some examples of popular DBMS, Sql, Oracle, Sybase, Microsoft Access and IBM DB2.

The database system can be divided into four components:

1. The database system can be divided into System developer and End users.
2. Database application: Database application may be Personal, Departmental, Enterprise and Internal
3. DBMS: Software that allow users to define, create and manages database access, Ex: Sql, Oracle etc.
4. Database: Collection of logical data.

Functions of database management system:

1. Provides Recovery services
2. Provides utility
3. Provides data Independence
4. Provides a clear and logical view of the process that manipulates data.

Advantages of DBMS:

1. Segregation of application program
2. Minimal data duplication
3. Reduced development time and maintenance need
4. Easy retrieval of data

1.2 Problem statement

The current car rental shop management system is inadequate in adequately tracking customer information, vehicle availability, rental agreements, and billing information. This has caused significant problems in managing customer information and rental agreements, leading to inefficient business operations and customer service. The current system is also unable to effectively track vehicle availability, leading to lost revenue opportunities. Additionally, the current system is not able to adequately manage billing information, leading to costly billing errors and inefficiencies. As a result, the current car rental shop management system needs to be improved to provide more effective customer tracking, vehicle availability tracking, rental agreement management, and billing management.

1.3 Objective

1. To develop a stand alone application using C# as a front-end tool.
2. Creating and populating database with tables containing tuples regarding the aforementioned data.
3. Establishing secure connection between the front-end and the database.
4. Testing, Debugging and developing the application to present standards.
5. Providing means of Data Abstraction via different logins.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Software and Hardware

Software Requirements:

Operating system: Windows 10, 64 bits

Front end: C#

Server-side language: C#

Back end: MySQL

IDE: Microsoft Visual Studio

Hardware Requirements:

Processor: AMD64 and Intel 64T

Clock speed: 1.8 GHz minimum

Screen resolution: 1024 X 768

RAM: 2 GB minimum

Disk space: 15 GB

Chapter 3

SYSTEM DESIGN

3.1 E R Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope and the inter relationships among these entities.

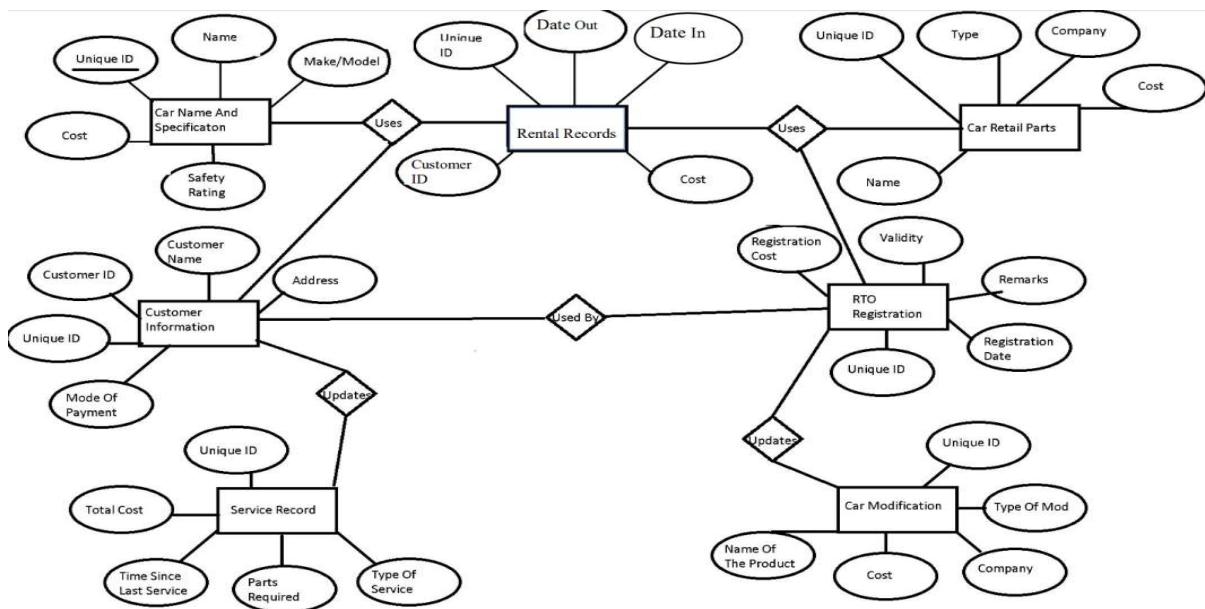


Figure 3.1: ER diagram of Car Rental Shop Management System

The above ER diagram consists of 7 entities given by rectangles. The relationships between entities are given by rhombus. The attributes are given in ovals. Key attributes represent the candidate key or super key of the tables and are represented by underlining the attributes. The numbers on the links determine the cardinality of the relation between the entities.

3.2 Schema Diagram

1. Car Name And Specification

Unique ID(Primary Key)	Model/Make	Name	Safety Rating	Cost
------------------------	------------	------	---------------	------

2. Rental Records

Unique ID(Foreign Key)	Date Out	Date In	Customer ID	Cost
------------------------	----------	---------	-------------	------

3. Customer Information

Customer ID(Primary Key)	Customer Name	Unique ID(Foreign Key)	Address	Mode Of Payment
--------------------------	---------------	------------------------	---------	-----------------

4. Car Retail Parts

Unique ID(Foreign Key)	Type	Manufacturing Company	Name Of Product	Cost
------------------------	------	-----------------------	-----------------	------

5. Car Modification

Unique ID(Foreign Key)	Type of Modification	Manufacturing Company	Name Of Product	Cost
------------------------	----------------------	-----------------------	-----------------	------

6. RTO Registration

Unique ID(Foreign Key)	Registration Date	Registration Cost	Validity	Remarks
------------------------	-------------------	-------------------	----------	---------

7. Service Record

Unique ID(Foreign Key)	Type of Service	Parts Required	Time since last service	Total Cost
------------------------	-----------------	----------------	-------------------------	------------

Figure 3.2: Schema diagram of Car Rental Shop Management System

A Database Schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and the relations among them are associated. It formulates all the constraints that are to be applied on the data. It contains a descriptive detail of the database, which can be depicted by means of schema diagram.

3.3 Overview of GUI

GUI is a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. C#, otherwise known as C Sharp, is a type of general-purpose, object-oriented programming language developed by Microsoft in 2000. C# is based on the C++ language, and is designed to be a simpler and more efficient alternative to C++. It is a versatile language, used for a wide variety of applications, including web and mobile applications, games, enterprise software, and more. C# is a popular language among developers due to its straightforward syntax, and its support for Object-Oriented Programming (OOP) principles. This allows developers to create robust applications that are easy to maintain and extend.

3.4 Normalization

Normalization is a process of analyzing the given relation schema based on their functional dependencies and primary key to achieve desirable properties of minimizing redundancy and minimizing insert, delete, update anomaly. The normalization process takes a relation schema through a series of tests to certify whether it satisfies a certain normal form. The normal form of a relation refers to the highest normal form condition that it meets, and hence the degree to which it has been normalized.

Normalization rule are divided into following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. Boyce-Codd Normal Form or (3.5NF)

3.4.1 First Normal Form

First normal form states that the domain of an attribute must include only atomic (simple, individual) values and that the value of any attribute in a tuple must be a single value from the domain of the attribute.

3.4.2 Second Normal Form

Second normal form is based on the concept of full functional dependency. A functional dependency $X \rightarrow Y$ is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore. A relational schema R is in 2NF if every non prime attribute A in R is fully functionally dependent on the primary key of R.

3.4.3 Third Normal Form

Third normal form is based on the concept of transitive dependency. A relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key. A relation schema R is in 3NF if every nonprime attribute of R meets both of these following conditions:

1. It is fully functionally dependent on every key of R
2. It is non transitively dependent on every key of R

The relations used in this database are fully functionally dependent on its key attribute and does not hold any transitive dependencies. Hence all the relations are in 3NF.

3.4.4 Boyce Codd Normal Form or (3.5NF)

Boyce normal form states that on any given relation, the functional dependency frame should definitely be a candidate key or a super key. A table is in BCNF if every functional dependency $X \rightarrow Y$ where X is the super key of the table.

We can say it is in Boyce Codd Normal Form or BCNF since the left part of all the functional dependencies is a super key. Hence all the relations are in BCNF.

Thus we can say that the given relational schema satisfies all the normal forms up to Boyce Codd Normal Form or 3.5NF. Since our relation satisfies many of these normal forms, we can say that the overall redundancy of the data that will be inserted into the tables will be minimized and overall there will be less occurrences of insert, delete and update anomalies taking place. The higher the degree to which it is normalized the less error prone the application's database will be. In conclusion, normalization helps the application to be organized and minimize the redundancy of data.

Chapter 4

IMPLEMENTATION

4.1 Table Creation

Here we create the tables: Car Name, Rental Records, Car Retail Parts, Service Record, RTO Registration, Customer Info, Car Modification. In the tables, data is organized in row and column format. The rows represent a unique record and the columns represent the attributes of a given table. The tables provide the overall structure of the different tables that are used; showcasing the various attributes and the respective data types.

Car Name Table:

```
CREATE TABLE [dbo].[CarNameAndSpec] (
[UniqueID] INT NOT NULL,
[Name] VARCHAR (10) NOT NULL,
[Make_Model] VARCHAR (10) NOT NULL,
[Cost] INT NOT NULL,
[SafetyRating] INT NOT NULL,
PRIMARY KEY CLUSTERED ([UniqueID]));
```

Customer Information:

```
CREATE TABLE [dbo].[CustomerInfo] (
[Customer ID] INT NOT NULL,
[UniqueID] INT NOT NULL,
[Customer Name] VARCHAR (20) NOT NULL,
[Address] VARCHAR (20) NOT NULL,
[Mode Of Payment] VARCHAR (5) NOT NULL,
PRIMARY KEY CLUSTERED ([Customer ID] ASC),
CONSTRAINT [FK_CustomerInfo_ToTable] FOREIGN KEY ([UniqueID]) REFERENCES [dbo].[CarNameAndSpec] ([UniqueID));
```

4.2 Description of Table

`desc Car Name And Spec;`

	Name	Data Type	Allow Nulls	Default
1	UniqueID	int	■	
2	Name	varchar(10)	■	
3	Make_Model	varchar(10)	■	
4	Cost	int	■	
5	SafetyRating	int	■	

Figure 4.1: Description of Car Name And Spec table

In the above figure, the structure of the Car Name And Spec table is given as:

Uniqueid: is the primary key and has integer data type. Is used to uniquely identify the Car.

Name: has data type varchar (15) and gives the Car name.

Cost: has data type int and tells us the Cost

desc Customer Info;

	Name	Data Type	Allow Nulls	Default
1	Customer ID	int	<input checked="" type="checkbox"/>	
2	UniqueID	int	<input checked="" type="checkbox"/>	
3	Customer Name	varchar(20)	<input checked="" type="checkbox"/>	
4	Address	varchar(20)	<input checked="" type="checkbox"/>	
5	Mode Of Payment	varchar(5)	<input checked="" type="checkbox"/>	

Figure 4.2: Description of Customer Info table

In the above figure, the structure of the Customer Info table is given as:

Customer ID: is the primary key and has integer data type. Is used to uniquely identify the Customer.

Unique ID: is a foreign key and has data type integer. Is used to indicate the store to which car did the customer rent.

Customer Name: has data type varchar (35) and tells us the Customer name.

Cost: has data type integer and provides the total cost.

desc Rent Out;

	Name	Data Type	Allow Nulls	Default
1	UniqueID	int	<input checked="" type="checkbox"/>	
2	CustomerID	int	<input checked="" type="checkbox"/>	
3	DateOUT	date	<input checked="" type="checkbox"/>	
4	DateIN	date	<input checked="" type="checkbox"/>	
5	TotalCost	int	<input checked="" type="checkbox"/>	

Figure 4.3: Description of Rent Out table

In the above figure, the structure of the Rent Out table is given as:

Unique ID: is the Foreign Key and has integer data type. Is used to uniquely identify the Car.

Customer ID: is a foreign key and has data type integer. Is used to indicate which Customer Took the Car

Cost: has data type int and tells us the Total Cost.

4.3 Populated Tables

Select * from CarNameAndSpec;

	UniqueID	Name	Make_Model	Cost	SafetyRating
	35	Skyline	GTR	50000	4
	55	Supra	Cute	69000	5
	66	Celerio	Budget	80000	5
	87	Benz	Sedan	400000	3
	98	Miata	Headlights	50000	2

Figure 4.4: Values of Car Name And Spec table

The above figure provides us the records of the Car Name And Spec Table.

Select * from Customer Info;

	Customer ID	UniqueID	Customer Name	Address	Mode Of Paym...
	12	35	sourav	HPT	UPI
	69	66	Kashyap	BNGL	Cash
	47	87	Prajwal	Mngl	UPI

Figure 4.5: Values of Customer Info table

The above figure provides us the records of the Customer Info Table.

Select * from RTO Registration;

	UniqueID	RegistrationAg...	RegistrationCost	Validity	CarNumberAll...
	35	Prajwal	356	5	6969
	55	Dheeraj	415	5	4201
	87	Harshith	485	4	3567

Figure 4.6: Values of RTO Registration table

The above figure provides us the records of the RTO Registration Table.

Select * from Service Record;

	UniqueID	TypeOfService	LastServiceDate	NumberOfParts...	TotalCost
	35	Routine	20-11-2015	5	5200
	66	Engine	11-04-2003	2	7200

Figure 4.7: Values of Service Record table

The above figure provides us the records of the Service Record Table

4.4 SQL Triggers and Stored Procedures

4.4.1 Trigger

A trigger is a piece of code inserted in the SQL database in an order to execute or run certain action when a condition is met or usually a command is given upon which we need to notify the admin or the user at the same. The below code is the trigger that is implemented in the project.

go

```
CREATE TRIGGER MyTrigger ON dbo.RTORегистрации  
INSTEAD OF INSERT  
AS  
  
if not exists (  
select * from RTORегистрации t  
inner join inserted i  
on i.CarNumberAlloted = t.CarNumberAlloted )  
begin  
Insert into RTORегистрации  
(UniqueID,RegistrationAgent,RegistrationCost,Vaidity,CarNumberAlloted) Select  
UniqueID,RegistrationAgent,RegistrationCost,Vaidity,CarNumberAlloted from  
inserted  
end  
else  
THROW 51000, 'The Selected Car Number is Already Allotted for another Car!!!', 1;
```

4.4.2 Stored Procedure

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group. So if a query has to be written over and over again, instead of having to write that query each time, it can be saved as a stored product and can be executed just by calling the procedure. In addition, parameters can also be passed to the stored procedure. So depending on the need, the stored procedure can act accordingly. This Stored procedure is for inserting a new car tuple in the Car Name And Spec Table. This procedure will take the arguments and then add them into the table accordingly.

```
CREATE PROCEDURE [dbo].[AdminCarAdd_SP]
    @UniqueID INT ,
    @Name VARCHAR (10) ,
    @Make_Model VARCHAR (10) ,
    @Cost INT ,
    @SafetyRating INT
AS
    insert into CarNameAndSpec (UniqueID,Name,Make_Model,Cost,SafetyRating) values
    (@UniqueID,@Name,@Make_Model,@Cost,@SafetyRating)
    RETURN 0
```

4.5 Database Connectivity

A Database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the form of a result set. We use an in built module available in C# to connect the database to the form.

Using Windows.Data.SqlClient;

```
SqlConnection con = new SqlConnection("DataSource=(LocalDB)\\
MSSQLLocalDB;AttachDbFilename=\"A:\\
DBMS\\Project\\WindowsFormsApp1\\Properties\\CarDB.mdf\";Integrated Security=True");
```

The above line of code creates an instance of the database with the object name **con** which can be used anywhere in the code.

4.6 Source Code (Front End)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class LoginForm : Form
    {
        public LoginForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            String u1 = "Sourav", p1 = "Deadlifts", u2 = "Prajwal", p2 = "PJ94824" , u3
= "Kashyap" , p3 = "MangloreBuns" , u4 = "Dheeraj" , p4 = "Dheeru123",name,
pass;
            name = textBox1.Text;
            pass = textBox2.Text;
            if (name.Equals(u1) && pass.Equals(p1))
            {
                MessageBox.Show("Admin Login Success");
                AdminPage obj1 = new AdminPage();

                obj1.Show();
            }
            else if (name.Equals(u2) && pass.Equals(p2))
            {
                MessageBox.Show("Customer Login Success");
                Customer obj1 = new Customer();

                obj1.Show();
            }
        }
    }
}
```

```
        else if (name.Equals(u3) && pass.Equals(p3))
    {
        MessageBox.Show("Customer Login Success");
        Customer obj1 = new Customer();

        obj1.Show();
    }
    else if (name.Equals(u4) && pass.Equals(p4))
    {
        MessageBox.Show("Employee Login Success");
        Employee obj1 = new Employee();

        obj1.Show();
    }
    else
    {
        MessageBox.Show("Wrong Credentials !!");
    }

}
}
}
```

Chapter 5

RESULTS

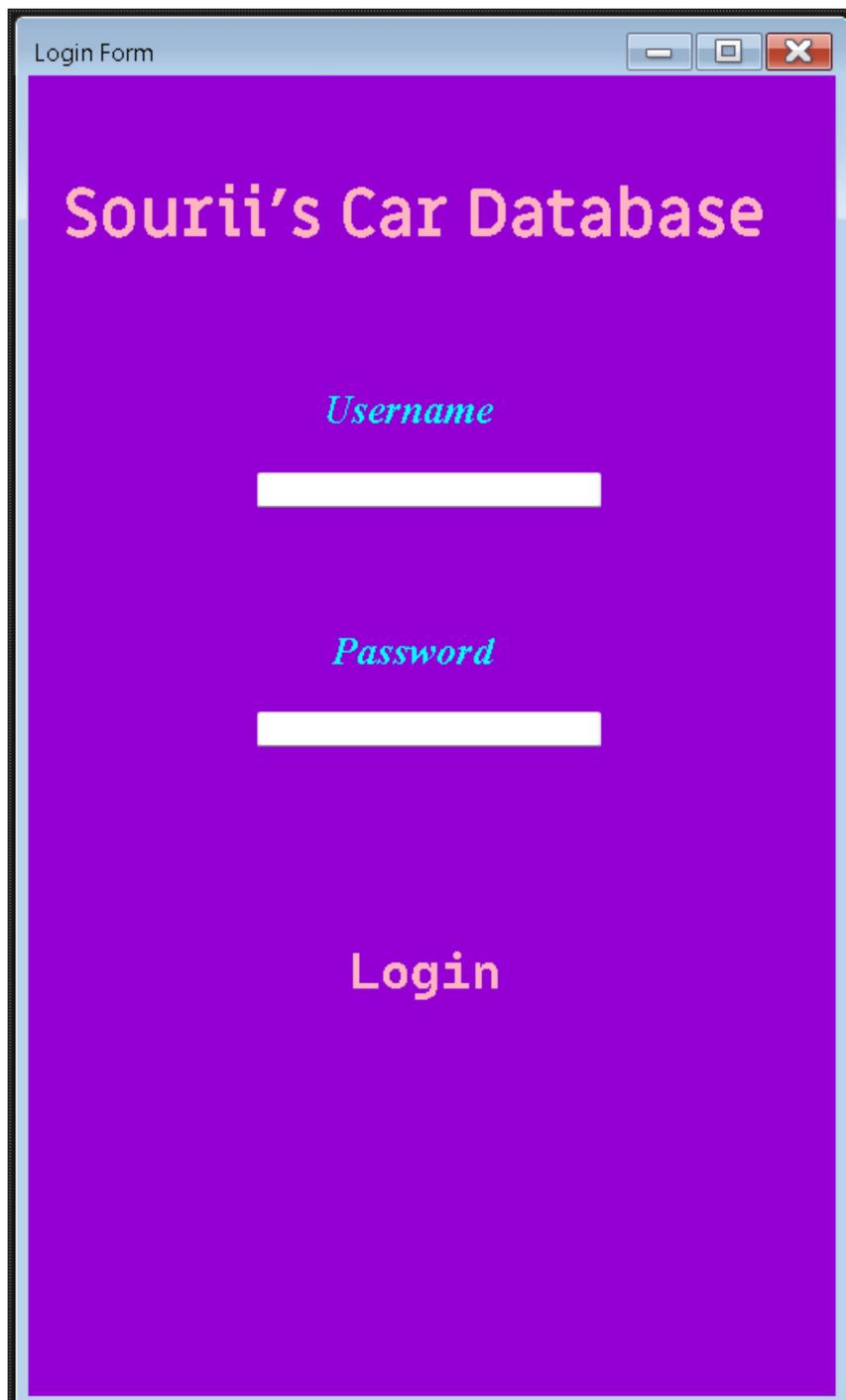


Figure 5.1: Login Form

The above page is the first form in the app. It is a simple login form designed to abstract data via providing 3 different levels of login namely Admin, Customer and Employee.

Based on the credentials entered we will be directed to the next different forms.

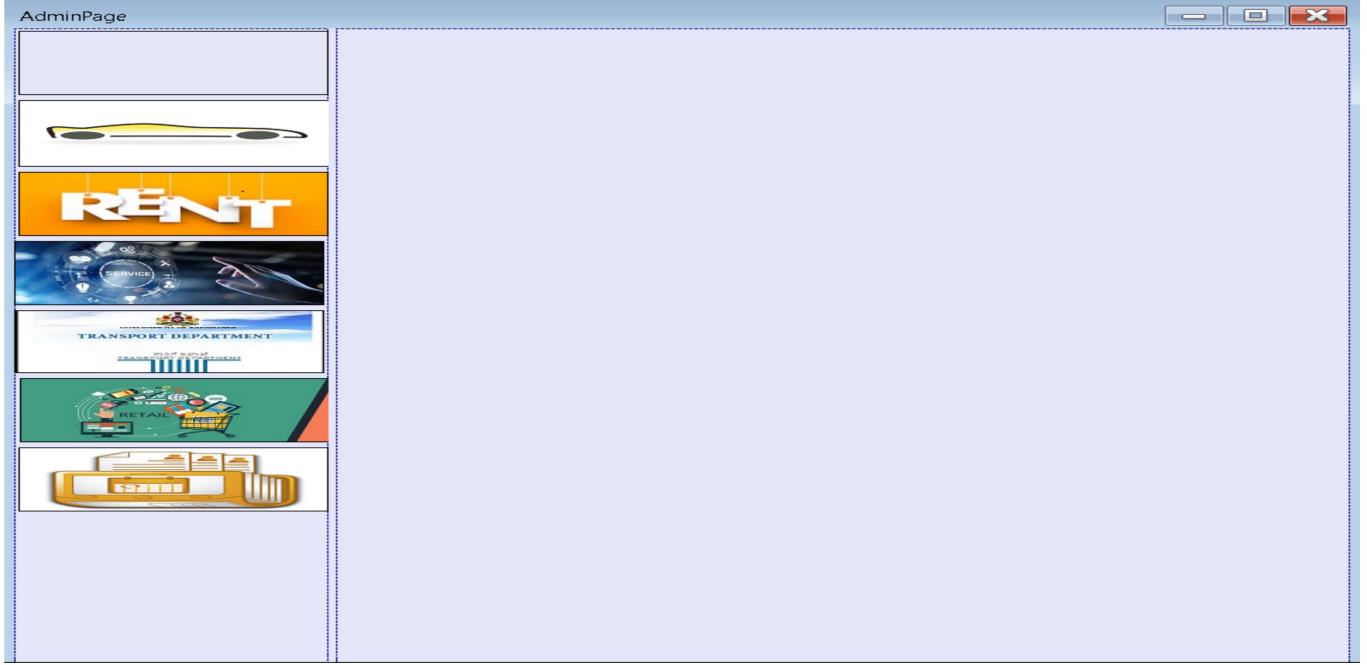


Figure 5.2: Admin User Interface

The above page can be accessed by entering the credentials for the admin. This page has access to all the tables and can edit/modify the same.

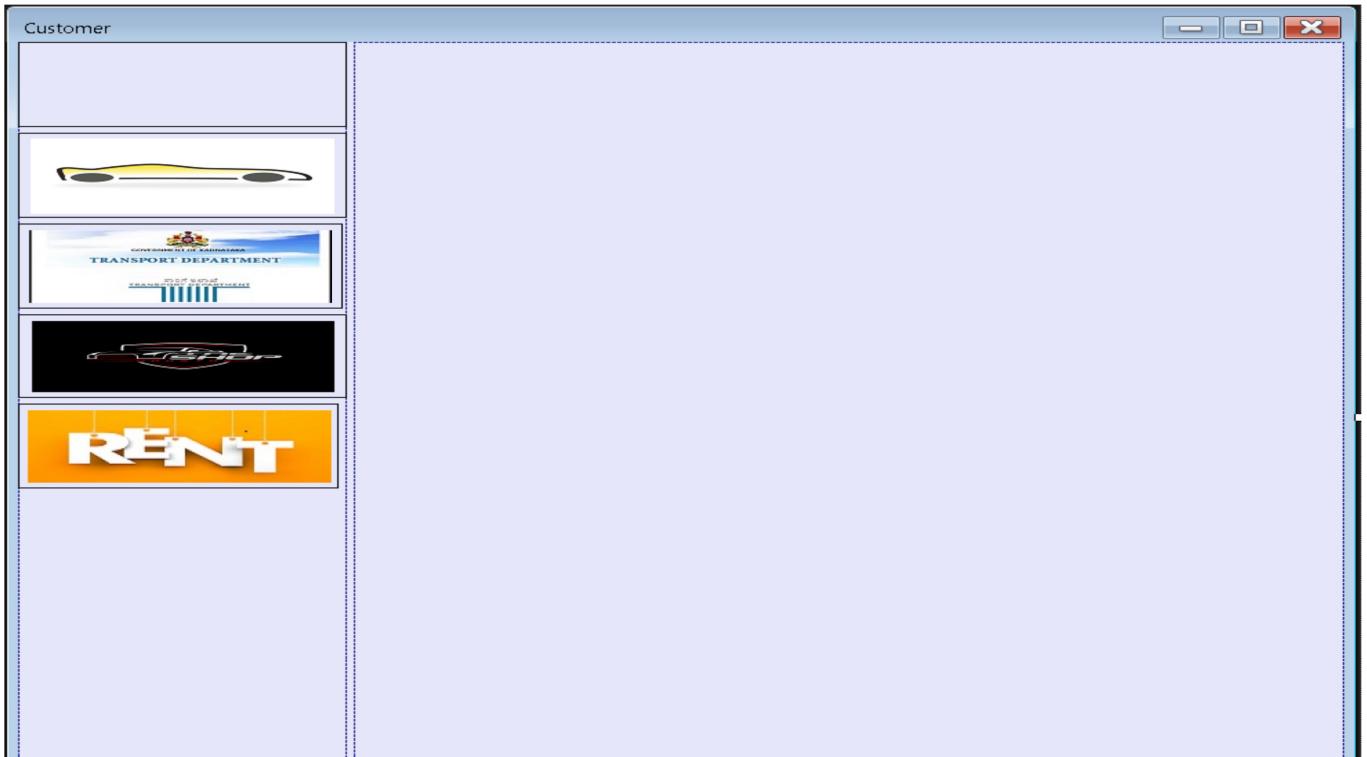


Figure 5.3: Customer User Interface

The above page can be accessed by entering the credentials for the Customer. This page has access to Car Name And Spec , RTO Registration , Mod and Rental Records Table.

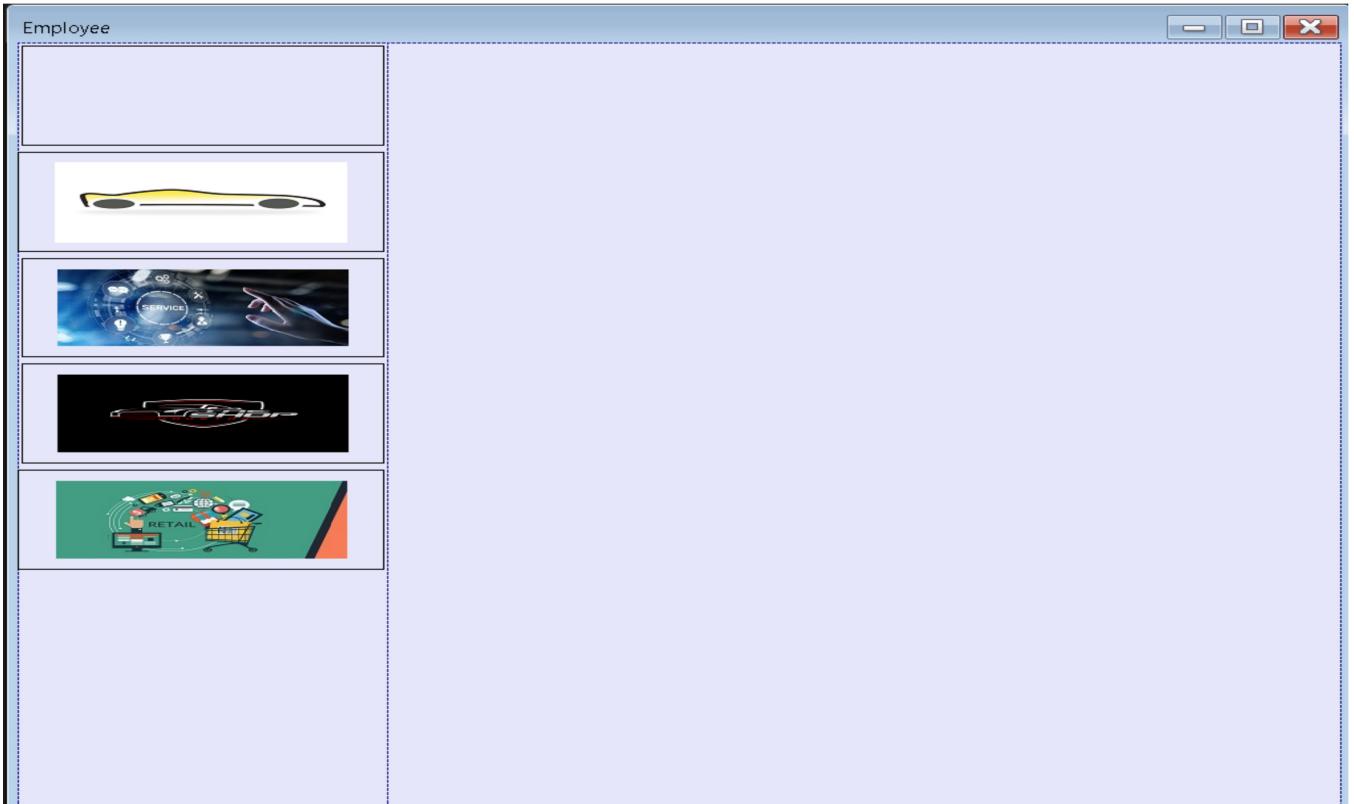


Figure 5.4: Employee User Interface

The above page can be accessed by entering the credentials for the Employee. This page has access to few tables on which they can modify and manipulate data.

The screenshot shows an 'AdminPage' window. On the left, there is a sidebar with several icons: a car silhouette, a hand interacting with a service interface, the word 'RENT' in large letters, a service interface, the 'GOVERNMENT OF KARNATAKA TRANSPORT DEPARTMENT' logo, a shopping cart, and a retail store interior. The main area contains a search interface with a 'Search' button and a table displaying car data. Below the table are buttons for 'Add', 'Delete', 'Update', and 'Clear', along with input fields for 'Car ID', 'Car Name', 'Make/Model', 'Cost', and 'Safety Rating'.

UniqueId	Name	Make_Model	Cost	SafetyRating
35	Skyline	GTR	50000	4
55	Supra	Cute	69000	5
66	Celerio	Budget	80000	5
87	Benz	Sedan	400000	3
98	Miata	Headlights	50000	2

Figure 5.5: Car Table Display and Data Entry

This page is used to access and modify the data in the car name and spec table. The Grid view displays all the changes that happens in the database.

AdminPage

UniqueId	CustomerID	DateOUT	DateIN	TotalCost
35	12	02-11-2020	05-11-2020	3600
66	69	02-11-2020	02-12-2020	600

Add Delete Update Clear

Car ID: _____

Customer ID: _____

Date Out: _____

Date IN: _____

Total Cost: _____

Figure 5.6: Customer Table Data input and View

The Above page is the form used for customer table data input and view

AdminPage

UniqueId	TypeOfService	LastServiceDate	NumberOfPartsRequired	TotalCo
35	Routine	20-11-2015	5	5200
66	Engine	11-04-2003	2	7200

Add Delete Update Clear

Car ID: _____

Type Of Service: _____

Last Service Date: _____

Number Of Parts Required: _____

Total Cost: _____

Figure 5.7: Service Table Data Input and View

The above page is used to input and view data in service records

18CSL58/Car Rental Shop Management System

The screenshot shows a Windows application window titled "AdminPage". On the left side, there is a vertical sidebar containing several icons: a car, a hand holding a key, a "RENT" sign, a service wrench, a transport department logo, a retail shopping cart, and a building. The main panel has a light blue header with a search bar labeled "Enter Car ID" and a green "Search" button. Below the header is a table with columns: UniqueID, RegistrationAgent, RegistrationCost, Validity, and CarNumber. The table contains five rows of data. At the bottom of the main panel are four green buttons: "Add", "Delete", "Update", and "Clear". Below these buttons are four input fields with labels: "Car ID", "Registration Agent", "Validity", and "Car Number Alloted".

	UniqueID	RegistrationAgent	RegistrationCost	Validity	CarNumber
▶	35	Prajwal	356	5	6969
	55	Dheeraj	415	5	4201
	87	Harshith	485	4	3567
	66	Shashank	742	9	1156
●					

Figure 5.8: RTO Registration Table Display and Data Entry

The above page is the form used to modify/manipulate and view the RTO Registration Table

This screenshot shows a Windows application window titled "AdminPage". It features a sidebar with icons for a car, a hand holding a key, a "RENT" sign, a service wrench, a transport department logo, a retail shopping cart, and a building. The main panel has a green header with a search bar labeled "Enter Car ID" and a green "Search" button. Below the header is a table with columns: UniqueID, NameOfTheProduct, ManufacturingCompany, Cost, and YearsofUseAllow. The table contains two rows of data. At the bottom of the main panel are four green buttons: "Add", "Delete", "Update", and "Clear". Below these buttons are four input fields with labels: "Car ID", "Name Of The Product", "Manufacturing Company", and "Cost". There is also a field labeled "Years Of Use Allowed".

	UniqueID	NameOfTheProduct	ManufacturingCompany	Cost	YearsofUseAllow
▶	87	IntakeValve	Mercedes	5200	5
	98	Bearings	Mazda	4000	2
●					

Figure 5.9: Retail Table Display and Data Entry.

The above page is the form used to modify/manipulate and view the Retail Parts Table.

CONCLUSION

The Car Rental Shop Management System has been a successful project and has provided many benefits to the business. It has allowed the business to better manage their fleet of vehicles, track their customer's rental data, and improve customer satisfaction. The system has also enabled the business to better monitor the performance of their staff, streamline their workflow, and increase their overall efficiency. The implementation of this system has been a success, with the business experiencing increased efficiency, customer satisfaction, and profitability. This system has been able to reduce the administrative burden of managing a fleet of vehicles and customers whilst providing real-time insights into the performance of the business. This system has also been able to provide improved customer service and better customer experiences. The Car Rental Shop Management System has been an important step forward for the business, providing a platform to manage the fleet, customers, and staff in an efficient and effective manner. The system has enabled the business to reduce administrative costs and time, while providing greater visibility into the performance of the business. Overall, the Car Rental Shop Management System has been a successful project and has provided many benefits to the business. The system has allowed the business to better manage their fleet of vehicles, track their customer's rental data, and improve customer satisfaction, while providing a platform to improve the efficiency and profitability of the business. The successful implementation of this system has been a major factor in the business's success and should be seen as a positive step forward for the business.

FUTURE ENHANCEMENTS

1. Bill Generation Table with data from all the tables.
2. Adding triggers to identify and alert admin for malicious attempt to login usnig wrong credentials
3. Implementing a multiple customer registration and login.
4. Making a new login for RTO officials

References/Bibliography

1. w3schools.org
2. geeksforgeeks.org
3. quora.com
4. reddit.com
5. tutorialsteacher.com
6. stackoverflow.com
7. Microsoft Visual Studio Documentation
8. C# Documentation