



SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(An Autonomous Institution)
Adyar, Mangaluru-575007
2024-25

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DBMS MINI PROJECT REPORT

ON

CAR RENTAL MANAGEMENT SYSTEM

BY

PREETI CHANDRAHAS BISANALLI
4SF22CS152

VARSHA G D
4SF22CS242

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

DATABASE MANAGEMENT SYSTEM LABORATORY

Under the guidance of

Ms. Ashwini C S

Assistant Professor, Dept. of CS&E

Dr. Suhas A Bhyratae

Associate Professor, Dept. of CS&E

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(An Autonomous Institution)
Adyar, Mangaluru-575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project “**Car Rental Management System**” is submitted in partial fulfillment for the requirement of V sem. B.E. (Computer Science & Engineering), “**DATABASE MANAGEMENT SYSTEM LABORATORY**” during the year 2024-25 is a result of bonafide work carried out by

Preeti Chandrahas Bisanalli
Varsha G D

4SF22CS152
4SF22CS242

.....
Ms. Ashwini C S
Assistant Professor
Dept. of CS&E
SCEM, Mangaluru

.....
Dr. Suhas A Bhyratae
Associate Professor
Dept. of CS&E
SCEM, Mangaluru

.....
Dr. Mustafa Basthikodi
Professor and HOD
Dept. of CS&E
SCEM, Mangaluru

Signature of the Examiners

1.

2.

ABSTRACT

A Database Management System (DBMS) serves as the cornerstone technology for creating and managing databases, providing a unified platform for organizing, retrieving, updating, and managing data. This abstract introduces the Car Rental Management System is being developed for customers so that they can book their vehicles from any part of the world. This application takes information from the customers through filling their details. A customer being registered in the website has the facility to book a car which he requires. It is an online system through which customers can view available cars, register, and book car. We developed this project to book a car on rent at the fare charges. In present system all booking work done manually and it takes very hard work to maintain the information of booking and cars. if you want to find which car is available for booking then it takes a lot of time. It only makes the process more difficult and harder. This aim of the project is to automate the work performed in the car rental management system like records of cab, cabs available for booking, rental charges for cars, store records of the customer. This is a car booking software that provides a complete solution to all your day-to-day car booking office running needs. This system helps you to keep the information of customer online. Online car rental management system is a unique and innovative product.

ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that we are submitting the Mini Project Report “**Car Rental Management System**” We have completed it as a part of the V semester DATABASE MANAGEMENT SYSTEMS LABORATORY (CS522L5C) of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Ms. Ashwini C S and Dr. Suhas A Bhyratae Department of Computer Science & Engineering** for her innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We express our sincere gratitude to **Dr. Mustafa Basthikodi, Professor & Head of the Department of Computer Science & Engineering** for his invaluable support and guidance.

We sincerely thankful to our beloved **Principal Dr. S S Injaganeri, Sahyadri College of Engineering & Management**, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Preeti Chandrahas Bisanalli

4SF22CS152

V Sem, B.E, CSE

SCEM, Mangaluru

Varsha G D

4SF22CS242

V Sem, B.E, CSE

SCEM, Mangaluru

PAGE INDEX

Chapter No.	Topic	Page No.
1	Introduction	1-3
2	Literature Survey	4-7
3	Design	
3.1	ER Diagram	8
3.2	Relational Schema	9-13
3.3	Schema Diagram	14
4	Normalization	15-17
5	Implementation	18-22
6	Results	23-27
7	Conclusion	28
	References	29

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO DATABASE MANAGEMENT SYSTEM

DBMS stands for “Database Management System.” In short, a DBMS is a database program. Technically speaking it is a software system that uses a standard method of cataloging, retrieving, and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by used or other programs. A DBMS makes it possible for end users to create, read, update, and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware). If programs use the application programming interface (API) for the database that is provided by the DBMS, developers won’t have to modify programs just because changes have been made to the database. With relational DBMSs (RDBMSs), this API is SQL, a standard programming language.

1.2 PHP

PHP is a powerful server-side scripting language primarily designed for web development but also widely utilized as a general-purpose programming language. It is currently installed on over 244 million websites and 2.1 million web servers, showcasing its vast adoption and utility. Originally created by Rasmus Lerdorf in 1995, PHP's reference implementation is now maintained by The PHP Group. While PHP initially stood for Personal Home Page, it has since been redefined as PHP: Hypertext Pre-processor. PHP code is executed on a web server using a PHP processor module, which interprets the code and generates dynamic web pages. One of its key features is the ability to embed PHP commands directly into an HTML document, eliminating the need to call external files for data processing. Supported by a global community, PHP remains a top choice for developers aiming to build secure and feature-rich web solutions. Its rich library of built-in functions, extensive support for APIs, and features like OPcache for performance optimization further enhance its utility. Additionally, PHP is equipped with security functions to help protect applications from vulnerabilities like SQL injection and cross-site scripting. Over the years, PHP has expanded to include command-line interface capabilities and can be used to develop standalone graphical applications.

1.2 MYSQL

SQL (Structured Query Language) is a powerful and standardized programming language specifically designed for managing and manipulating relational databases. It is widely used in applications that require data storage, retrieval, and manipulation. SQL enables users to create, read, update, and delete data in a structured manner, making it essential for database management systems. Beyond modifying database structures and managing rows of data, SQL is also used for creating database objects like tables, views, indexes, and stored procedures. Its ability to retrieve subsets of data efficiently makes it indispensable for transactional systems, reporting, and business intelligence applications. One of the most popular relational database management systems using SQL is MySQL. Known for its speed, reliability, and ease of use, MySQL is a preferred choice for web applications and serves as a core component of the LAMP stack, which includes Linux, Apache, MySQL, and PHP/Perl/Python. MySQL is widely employed in content management systems, e-commerce platforms, and dynamic websites due to its ability to handle large datasets while ensuring high performance and scalability. Additionally, SQL is supported by various other RDBMS platforms like PostgreSQL, Oracle, Microsoft SQL Server, and SQLite, making it a universal tool in the realm of database management.

1.3 XAMPP

XAMPP, which stands for Cross-Platform, Apache, MySQL, PHP, and Perl, is an open-source web server solution stack that provides a comprehensive environment for developers to build, test, and deploy web applications locally. It is particularly useful for developing and testing WordPress sites offline on a local server without requiring an active internet connection. XAMPP simplifies the process of setting up a web server by bundling essential components such as Apache (for hosting web pages), MySQL (for database management), PHP (for server-side scripting), and Perl (a versatile programming language). The platform is cross-platform, meaning it is compatible with multiple operating systems, including Windows, macOS, and Linux, ensuring flexibility for developers. XAMPP is widely used to create a safe and controlled testing environment, allowing developers to validate projects that use technologies like Apache, PHP, MySQL, and Perl directly on their host system. Additionally, XAMPP's user-friendly control panel enables easy management of server components, such as starting or stopping services. Developers also benefit from pre-configured settings, which reduce the complexity of manual configurations. By using XAMPP, developers can identify and resolve issues during the development phase, ensuring a seamless deployment to live servers.

1.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

1.5 HTML5

HTML5 is a robust markup language designed for structuring and presenting content on the World Wide Web, representing a significant evolution from its predecessors. As the fifth and final major version of HTML recommended by the World Wide Web Consortium (W3C), HTML5 introduces numerous features that enhance user experiences, enable modern web applications, and improve cross-browser compatibility. Its development is now overseen by the Web Hypertext Application Technology Working Group (WHATWG), a collaborative effort by major browser vendors, including Apple, Google, Mozilla, and Microsoft. The current specification is referred to as the HTML Living Standard, emphasizing its continuous updates and adaptability to emerging web technologies. HTML5 brought several key advancements, such as semantic elements like <article>, <section>, <header>, and <footer>, which provide better structure and readability for web content. It also introduced APIs and features for building dynamic and interactive web applications, including support for audio (<audio>) and video (<video>) elements without requiring external plugins like Flash. Additionally, HTML5 supports offline storage with technologies like local storage and session storage, improving performance and reliability.

CHAPTER 2

LITERATURE SURVEY

The design and development of a vehicle rental management system have been significantly influenced by various foundational and practical references. These resources collectively offer insights into database systems, system analysis, structured design techniques, security, and client server architectures, providing a comprehensive framework for building a robust and scalable system.

M. Naga Keerthi and Dollu Bhargav Ashok Kumar[1] focuses on designing and implementing a comprehensive software solution to streamline car rental business operations. The system addresses challenges such as vehicle tracking, booking management, and customer service, aiming to enhance efficiency and user experience. Key features include vehicle inventory management, a user-friendly booking system, secure payment integration, customer profile management, and administrative tools for analytics and reporting. The proposed solution leverages modern technologies like dynamic web frameworks and robust databases, ensuring scalability and reliability. This research contributes to automating rental business workflows, ultimately benefiting both businesses and their customers.

The study by M. Sumithra, B. Buvaneswari, S. Ahilesharan, T. Fenix Raja Singh, and J. Harish [2] presents a comprehensive approach to developing an Online Vehicle Rental System. The system automates the traditional rental process, allowing users to book vehicles conveniently through an online platform. Key features include vehicle browsing, booking management, user registration, and secure payment options, enabling a seamless user experience. Additionally, the platform includes admin functionalities to manage bookings and monitor vehicle inventory efficiently. By leveraging modern technologies, the system aims to enhance customer convenience, reduce manual effort, and improve the operational efficiency of rental businesses.

Analysis & Design of Information Systems by James A. Senn [3] is a foundational text that explores methodologies for developing robust information systems. It focuses on requirement gathering, system modeling, and process mapping to ensure that system architectures align with user needs and organizational objectives. The book provides a structured approach to system development, covering techniques like data flow diagrams, entity-relationship modeling, and prototyping. By emphasizing the alignment of business goals with technical solutions, it serves as a critical guide for designing efficient and user-centric systems, such as a vehicle rental management system.

Systems Analysis by Jerry Fitzgerald and Ardra Fitzgerald [4] introduces structured analysis techniques that are essential for designing effective and organized information systems. The book focuses on creating a clear blueprint for system processes and data flows, using tools like data flow diagrams, decision tables, and process specifications. These techniques provide a systematic approach to understanding and documenting system requirements, ensuring accuracy and efficiency in system development. The text is

particularly valuable for creating well-defined and user-focused systems by emphasizing clarity and consistency throughout the analysis and design phases.

Modern Structured Analysis by Edward Yourdon[5] presents contemporary approaches to system decomposition and process modelling, focusing on creating clear and maintainable designs. The book emphasizes modularity, top-down design, and the use of structured methods to break down complex systems into manageable components. It introduces advanced tools like data flow diagrams, process hierarchies, and control flow analysis, which improve the precision and organization of system design. Yourdon's methodologies are instrumental in developing systems that are not only efficient and scalable but also easier to understand, modify, and maintain over time.

The Practical Guide to Structured System Design by Meilir Page-Jones[6] emphasizes creating modular, efficient, and adaptable system architectures. It provides practical insights into designing systems that are easy to modify, maintain, and expand, ensuring long-term usability and scalability. The book introduces principles like cohesion, coupling, and structured programming, which are essential for building robust systems. Its focus on modular design makes it a valuable resource for projects like a vehicle rental management system, where future enhancements and scalability are critical to meeting evolving business requirements.

Systems Analysis and Design by Kenneth and Julie Kendall[7] provides strategies for aligning system functionalities with project objectives, emphasizing user-centric design and efficient workflows. The book offers a comprehensive approach to system development, covering techniques such as system requirements gathering, process modeling, and prototyping. It highlights the importance of involving users throughout the design process to ensure the system meets their needs and facilitates seamless operations. This resource is particularly valuable for creating systems like a vehicle rental management system, where functionality and alignment with business goals are crucial.

Security in Computing by Charles P. Pfleeger [8] is a foundational resource that emphasizes the importance of safeguarding sensitive data, preventing unauthorized access, and ensuring secure online transactions. The book covers a wide range of security principles, including encryption, authentication, secure coding practices, and risk management. It also addresses emerging threats and strategies to mitigate vulnerabilities in web-based applications. These principles are critical for building trust with users and maintaining the integrity of systems, particularly in applications like a vehicle rental management system where secure handling of personal and financial data is paramount.

The "Online Vehicle Rental System" by Ansh Agrawal and Rishabh Mathur [9] focuses on creating an online platform for users to easily browse, book, and pay for vehicle rentals. The system aims to simplify the rental process, offering features like vehicle search, user registration, secure payments, and booking management. It enhances customer convenience and streamlines operations for service providers, improving efficiency and reducing administrative tasks.

The research paper titled "Car Rental Management System" by Labroo, Abhiti and Gupta, Deepak[10] focuses on the development and implementation of an efficient management system for car rental businesses. The paper presents a comprehensive approach to automating the entire rental process, from vehicle booking to payment processing. Key features include user-friendly interfaces for customers to browse available cars, make reservations, and process payments securely. The system also offers admin functionalities for tracking inventory, managing customer details, and overseeing booking records. By incorporating modern technologies, this management system aims to streamline operations, reduce manual effort, and enhance the overall customer experience, making it easier for both service providers and users to manage car rentals effectively.

The research by Qurratul Aini[11] titled "Development of Car Rental Management Information System" Case Study of Avis Indonesia discusses the creation and implementation of a car rental management system aimed at enhancing the efficiency and competitiveness of Avis Indonesia. The study highlights how an automated system can streamline the booking process, manage car inventories, and track customer interactions. By analyzing the case study of Avis Indonesia, the paper showcases how such a system can improve overall business performance by reducing manual efforts, improving operational speed, and increasing customer satisfaction. The research emphasizes the significance of information technology in automating processes, ensuring better data management, and helping businesses maintain a competitive edge in the rental market.

The research by P. Nahnisha, B. Jai Sai Anvitha, and R. Prema [12] introduces an Online Car Rental System that modernizes vehicle rental services by automating traditional processes. The platform allows users to book cars conveniently from anywhere, catering to various needs like temporary transportation or replacement vehicles. Key features include car browsing, booking, secure payments, profile management, and real-time notifications, while an admin dashboard enables companies to manage vehicles and track bookings effectively. Built with PHP, XAMPP Server, and MySQL, the system provides a scalable and user-friendly solution, enhancing customer satisfaction and streamlining operations for rental businesses.

The research by Rahul Kulkarni, Chaitanya R, Pratibha, Pooja A Patil, and Nikeeta Biradar[13] introduces an Online Car Rental System designed to optimize and simplify the car rental process. The system enables users to search, browse, and book cars online, enhancing convenience and accessibility. It includes essential features such as user registration, vehicle listing, booking management, secure payment options, and real-time notifications. The platform also supports administrative functions like vehicle tracking and booking oversight for rental companies. By integrating modern technologies, the system ensures a seamless and efficient experience for both users and service providers, reducing operational complexities and improving service quality.

The work by Linghuraj P. R., Aravind K., and Giritharan R.[14] titled Car Rental Management System explores the development of an online platform designed to streamline and modernize the car rental process. The system addresses the inefficiencies of traditional manual booking methods by introducing a web-based application that allows customers to register, browse available vehicles, and book cars online. It integrates a database management system to securely store and manage customer information, vehicle availability, and rental records. By automating these processes, the system reduces paperwork, saves time, and improves customer satisfaction while enabling rental businesses to operate more efficiently and effectively.

The study by Arijit Singh Yadav, Akash Shukla, Anand Srivastava, and Amit Pandey[15] titled "Online Car Rental System" presents a web-based platform designed to streamline car rental processes. It replaces traditional manual booking methods with an automated system that allows users to register, browse vehicles, and book cars online. The system integrates database management and web technologies to enhance efficiency, reduce paperwork, and improve user experience. It benefits customers with convenience and businesses by optimizing operations and boosting scalability.

CHAPTER 3

DESIGN

3.1 E-R DIAGRAM

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge.

The ER Diagram of our project is shown in the Figure 3.1.1

Symbols in Entity Relationship:

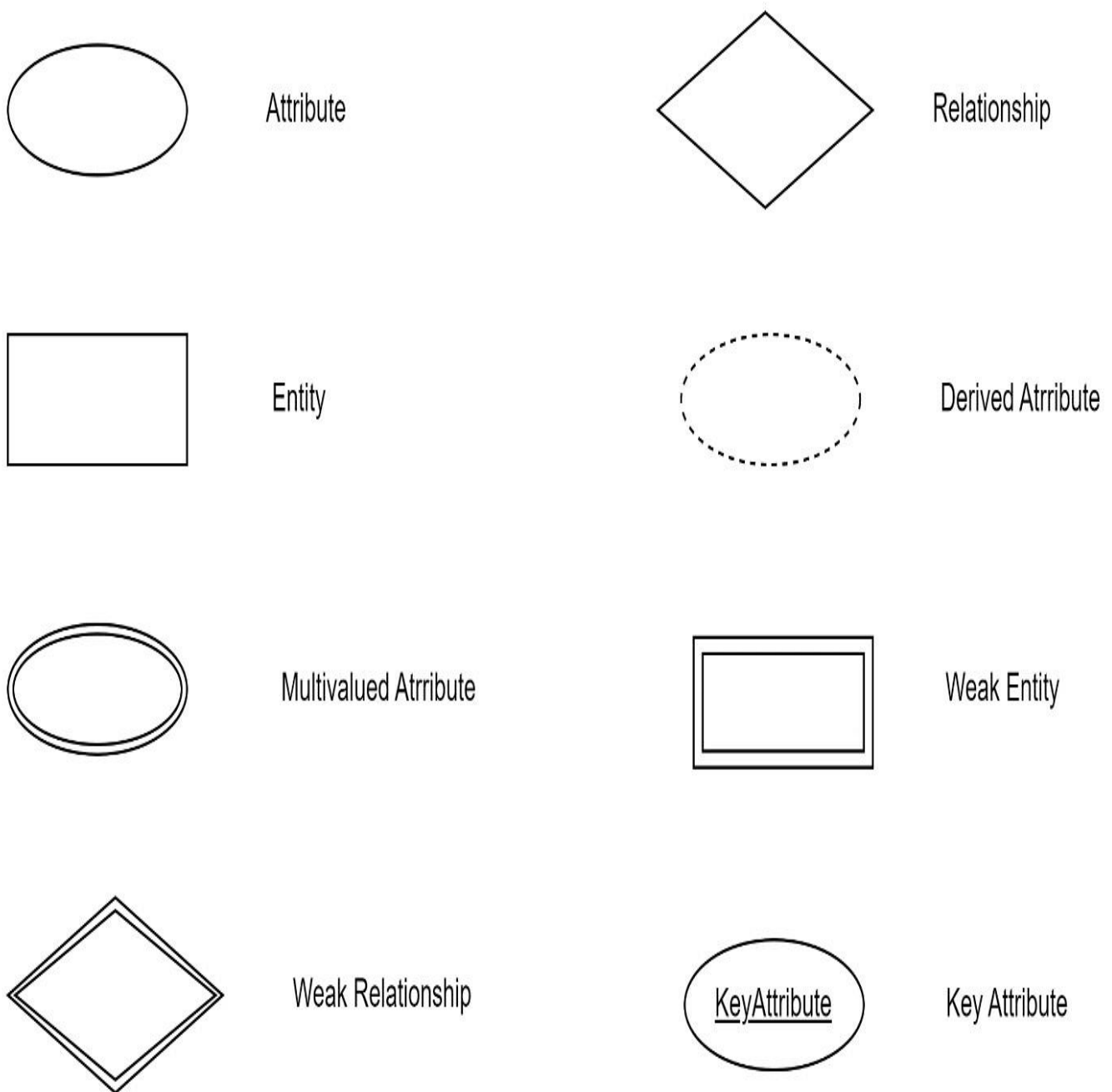


Figure 3.1.1 ER Notation

ER Diagram

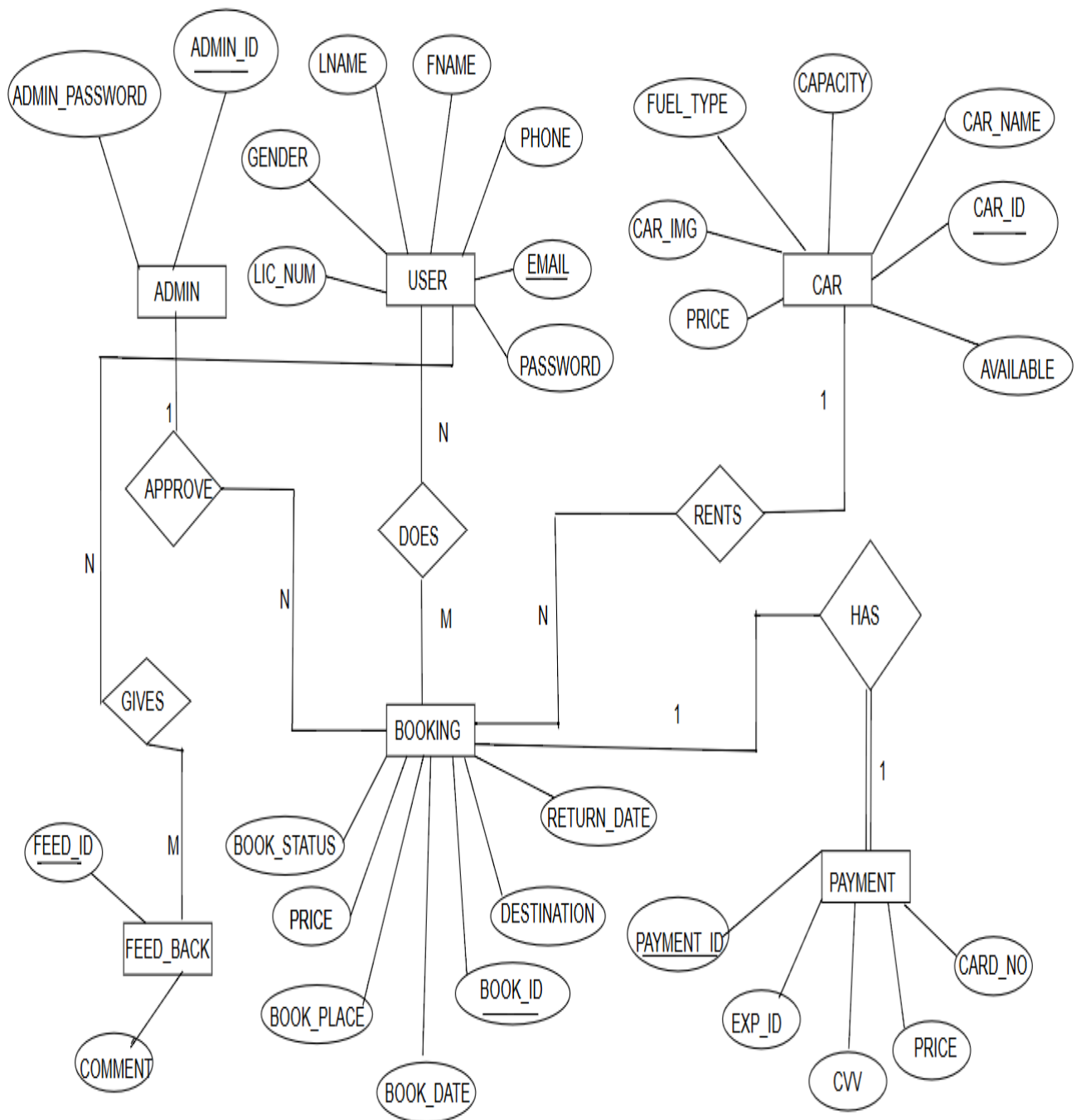


Figure 3.1.2 ER diagram of Car Rental Management System

3.2 RELATIONAL SCHEMA

3.2.1 Mapping of Regular Entity Type

For every entity in our relationship diagram, we have created a separate relation. These created relations contain the respected attributes and respected primary key.

ADMIN

<u>Admin_ID</u>	Admin_Password
-----------------	----------------

USER

<u>Email</u>	FName	LName	Password	Phone	Lic_Number	Gender
--------------	-------	-------	----------	-------	------------	--------

CAR

<u>Car_ID</u>	Car_Name	Fuel_Type	Capacity	Available	Price	Car_IMG
---------------	----------	-----------	----------	-----------	-------	---------

FEEDBACK

<u>Feed_ID</u>	Comment	Email
----------------	---------	-------

PAYMENT

<u>Payment_ID</u>	Card_No	Exp_Date	CVV	Price
-------------------	---------	----------	-----	-------

BOOKING

Book_ID	Car_ID	Book_Date	Book_Place	Destination	Return_Date	Book_Status	Price	Payment_ID
---------	--------	-----------	------------	-------------	-------------	-------------	-------	------------

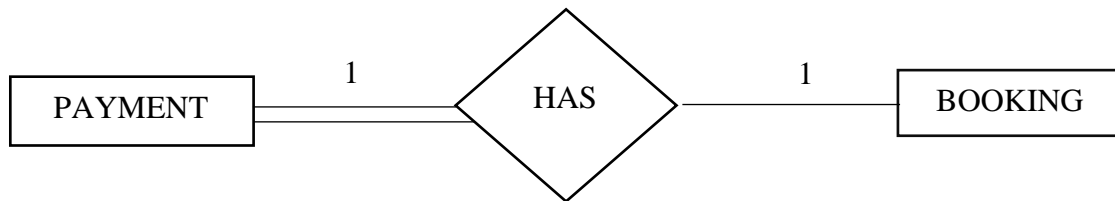
Figure 3.2.1: Mapping of Regular Entities

3.2.2 Mapping of Weak Entity Types

The ER Diagram has no weak entity, and hence we ignore this part.

3.2.3 Mapping of Binary 1:1 Relationship Types

In a 1:1 binary relationship, each instance of one entity is associated with exactly one instance of another entity. When representing this in a schema diagram, you would include foreign keys in both entities to establish the relationship. Booking Details has 1:1 relationship between Payment.



PAYMENT

<u>Payment_ID</u>	Card_No	Exp_Date	CVV	Price
-------------------	---------	----------	-----	-------

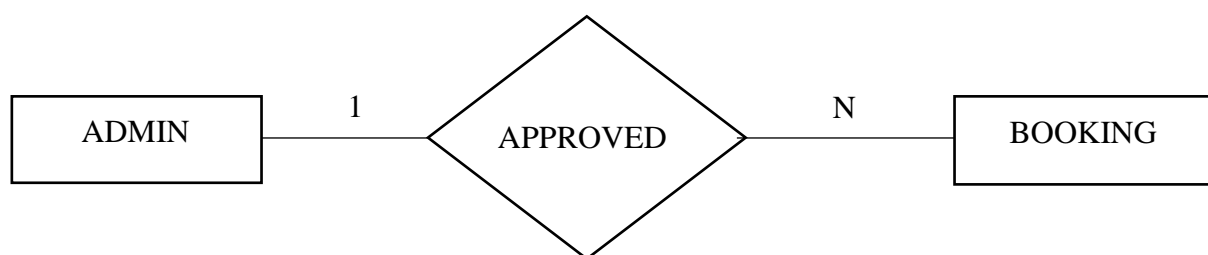
BOOKING

<u>Book_ID</u>	Car_ID	Book_ Date	Book_ Place	Destination	Return_ Date	Book_ Status	Price	Payment_ ID
----------------	--------	---------------	----------------	-------------	-----------------	-----------------	-------	----------------

3.2.4 Mapping of Binary 1:N Relationship Types

There are two 1:N relation in schema diagram are shown below:

1. ADMIN has 1:N relationship with BOOKING entity as APPROVED



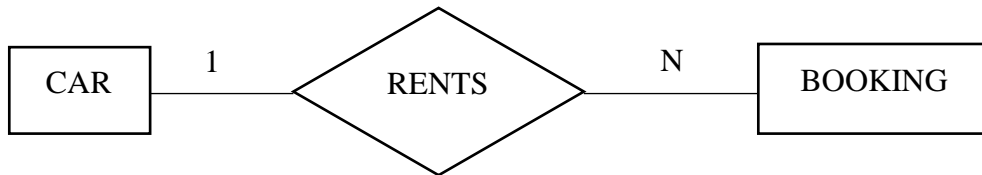
ADMIN

<u>Admin_ID</u>	Admin_Password
-----------------	----------------

BOOKING

Book_ID	Car_ID	Book_ Date	Book_ Place	Destination	Return_ Date	Book_ Status	Price	Payment_ ID
---------	--------	---------------	----------------	-------------	-----------------	-----------------	-------	----------------

2. CAR has 1:N relationship with BOOKIN entity as RENTS.



CAR

<u>Car_ID</u>	Car_Name	Fuel_Type	Capacity	Available	Price	Car_IMG
---------------	----------	-----------	----------	-----------	-------	---------

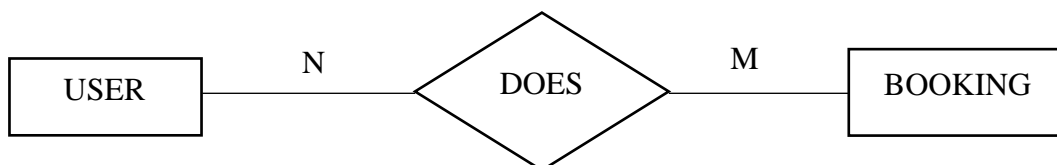
BOOKING

<u>Book_ID</u>	Car_ID	Book_Date	Book_Place	Destination	Return_Date	Book_Status	Price	Payment_ID	Email
----------------	--------	-----------	------------	-------------	-------------	-------------	-------	------------	-------

3.2.5 Mapping of M:N Relationship Types

There are two M:N relations in schema diagram and are shown below:

1. USER entity has relationship DOES with BOOKING entity with N:M cardinality ratio since many users can does many bookings.



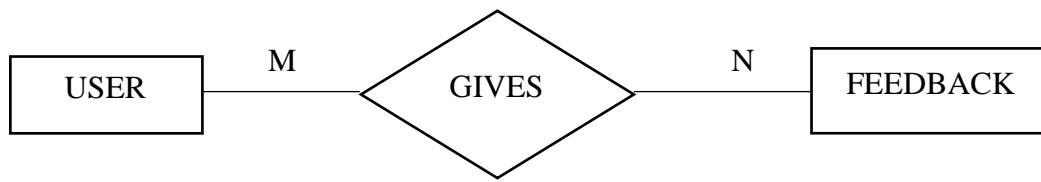
USER

<u>Email</u>	FName,	LName	Password	Phone	Lic_Number	Gender
--------------	--------	-------	----------	-------	------------	--------

BOOKING

<u>Book_ID</u>	Car_ID	Book_Date	Book_Place	Destination	Return_Date	Book_Status	Price	Payment_ID	Email
----------------	--------	-----------	------------	-------------	-------------	-------------	-------	------------	-------

2. The relation User has M:N relationship named GIVES with Feedback because Many user can give many feedbacks.

**USER**

<u>Email</u>	FName,	LName	Password	Phone	Lic_Number	Gender
--------------	--------	-------	----------	-------	------------	--------

FEEDBACK

<u>Feed_ID</u>	Comment	Email
----------------	---------	-------

3.3 SCHEMA DIAGRAM

ADMIN

<u>Admin_ID</u>	Admin_Password
-----------------	----------------

CAR

<u>Car_ID</u>	Car_Name	Fuel_Type	Capacity	Available	Price	Car_IMG
---------------	----------	-----------	----------	-----------	-------	---------

USER

<u>Email</u>	FName,	LName	Password	Phone	Lic_Number	Gender
--------------	--------	-------	----------	-------	------------	--------

BOOKING

<u>Book_ID</u>	Car_ID	Book_ Date	Book_ Place	Destination	Return_ Date	Book_ Status	Price	Email
----------------	--------	---------------	----------------	-------------	-----------------	-----------------	-------	-------

PAYMENT

<u>Payment_ID</u>	Card_No	Exp_Date	CVV	Price	Book_ID
-------------------	---------	----------	-----	-------	---------

FEEDBACK

<u>Feed_ID</u>	Comment	Email
----------------	---------	-------

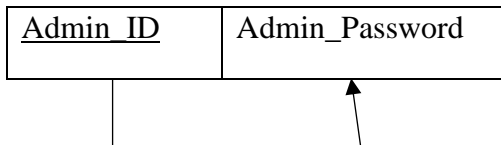
Figure 3.3.1: Schema Diagram of Car Rental Management System

CHAPTER 4

NORMALIZATION

Normalization is the process of organizing a database into smaller, well-structured tables to minimize data redundancy and improve data integrity. It involves breaking down large tables into smaller ones while maintaining relationships between them. The process uses normal forms (NF) as a guide, each with specific rules to refine the database design step by step.

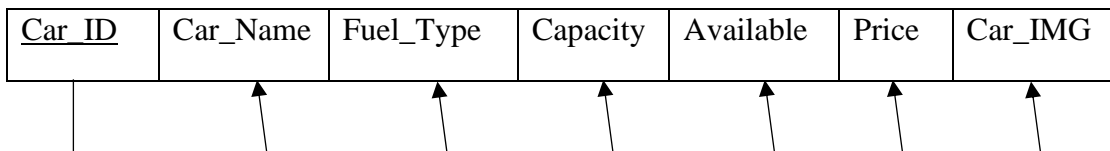
4.1 ADMIN Table



FD1: {Admin_ID} -> {Admin_Password}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF since Admin_ID Is the primary key, and all attributes are fully dependent on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.

4.2 CAR Table



FD2: {Car_Id} -> {Car_Id, Car_Name, Fuel_Type, Capacity, Available, Price, Car_IMG}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF because there is a single primary key Car_ID, and all attributes depend fully on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.

4.3 USER Table

<u>Email</u>	FName,	LName	Password	Phone	Lic_Number	Gender
--------------	--------	-------	----------	-------	------------	--------

FD3: {Email} -> {Email, FName, Lname, Password, Phone, Lic_Number, Gender}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF since Email is the primary key, and all attributes depend fully on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.

4.4 BOOKING Table

<u>Book_ID</u>	Car_ID	Book_Date	Book_Place	Destination	Return_Date	Book_Status	Price	Email
----------------	--------	-----------	------------	-------------	-------------	-------------	-------	-------

FD4: {Book_ID} -> {Car_Id, Book_Date, Book_Place, Destination, Return_Date, Book_Status, Price, Email}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF since Book_ID is the primary key, and all attributes are fully dependent on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.


4.5 PAYMENT Table

<u>Payment_ID</u>	Card_No	Exp_Date	CVV	Price	Book_ID
-------------------	---------	----------	-----	-------	---------

FD5: {Payment_ID} -> {Card_No, Exp_Date, Cvv, Price, Book_Id}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF since Payment_ID is the primary key, and all attributes depend fully on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.

4.6 FEEDBACK Table

<u>Feed_ID</u>	Comment	Email
		

FD6: {Feed_ID} -> {Comment, Email}

- **1NF:** Already satisfies 1NF as all attributes are atomic.
- **2NF:** Already satisfies 2NF since Feed_ID is the primary key, and all attributes depend fully on it.
- **3NF:** Satisfies 3NF as there are no transitive dependencies.

CHAPTER 5

IMPLEMENTATION

5.1 TABLE STRUCTURE

5.1.1 ADMIN Relation

```
CREATE TABLE ADMIN (
  ADMIN_ID VARCHAR(255) NOT NULL,
  ADMIN_PASSWORD VARCHAR(255) NOT NULL);
```

DESC ADMIN;

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
ADMIN_ID	varchar(255)	NO	PRI	NULL	
ADMIN_PASSWORD	varchar(255)	NO		NULL	

Figure 5.1.1: ADMIN RELATION

5.1.2 CAR Relation

```
CREATE TABLE CARS (
  CAR_ID INT(11) NOT NULL,
  CAR_NAME VARCHAR(255) NOT NULL,
  FUEL_TYPE VARCHAR(255) NOT NULL,
  CAPACITY INT(11) NOT NULL,
  PRICE INT(11) NOT NULL,
  CAR_IMG VARCHAR(255) NOT NULL,
  AVAILABLE VARCHAR(255) NOT NULL);
```

DESC CARS;

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
CAR_ID	int(11)	NO	PRI	NULL	auto_increment
CAR_NAME	varchar(255)	NO		NULL	
FUEL_TYPE	varchar(255)	NO		NULL	
CAPACITY	int(11)	NO		NULL	
PRICE	int(11)	NO		NULL	
CAR_IMG	varchar(255)	NO		NULL	
AVAILABLE	varchar(255)	NO		NULL	

Figure 5.1.2: CAR RELATION

5.1.3 BOOKING Relation

```
CREATE TABLE BOOKING (
  BOOK_ID INT(11) NOT NULL,
  CAR_ID INT(11) NOT NULL,
  EMAIL VARCHAR(255) NOT NULL,
  BOOK_PLACE VARCHAR(255) NOT NULL,
  BOOK_DATE DATE NOT NULL, VARCHAR(255) NOT NULL,
  RETURN_DATE DATE NOT NULL,
  PRICE INT(11) NOT NULL,
  BOOK_STATUS VARCHAR(255) NOT
  PHONE_NUMBER BIGINT(20) NOT NULL,
  DESTINATION
  NULL DEFAULT 'UNDER PROCESSING' );
```

DESC BOOKING;

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Extra options

Field	Type	Null	Key	Default	Extra
BOOK_ID	int(11)	NO	PRI	NULL	auto_increment
CAR_ID	int(11)	NO	MUL	NULL	
EMAIL	varchar(255)	NO	MUL	NULL	
BOOK_PLACE	varchar(255)	NO		NULL	
BOOK_DATE	date	NO		NULL	
PHONE_NUMBER	bigint(20)	NO		NULL	
DESTINATION	varchar(255)	NO		NULL	
RETURN_DATE	date	NO		NULL	
PRICE	int(11)	NO		NULL	
BOOK_STATUS	varchar(255)	NO		UNDER PROCESSING	

Figure 5.1.3: BOOKING RELATION

5.1.4 FEEDBACK Relation

```
CREATE TABLE FEEDBACK (
  FED_ID INT(11) NOT NULL,
  EMAIL VARCHAR(255) NOT NULL,
  COMMENT TEXT NOT NULL);
```

DESC FEEDBACK;

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Extra options

Field	Type	Null	Key	Default	Extra
FED_ID	int(11)	NO	PRI	NULL	auto_increment
EMAIL	varchar(255)	NO	MUL	NULL	
COMMENT	text	NO		NULL	

Figure 5.1.4: FEEDBACK RELATION

5.1.5 PAYMENT Relation

```
CREATE TABLE PAYMENT (
  PAY_ID INT(11) NOT NULL,
  BOOK_ID INT(11) NOT NULL,
  CARD_NO VARCHAR(255) NOT NULL,
  EXP_DATE VARCHAR(255) NOT NULL,
  CVV INT(11) NOT NULL,
  PRICE INT(11) NOT NULL);
```

DESC PAYMENT;

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
PAY_ID	int(11)	NO	PRI	NULL	auto_increment
BOOK_ID	int(11)	NO	UNI	NULL	
CARD_NO	varchar(255)	NO		NULL	
EXP_DATE	varchar(255)	NO		NULL	
CVV	int(11)	NO		NULL	
PRICE	int(11)	NO		NULL	

Figure 5.1.5: PAYMENT RELATION

5.1.6 USER Relation

```
CREATE TABLE USER (
  FNAME VARCHAR(255) NOT NULL,
  LNAME VARCHAR(255) NOT NULL,
  EMAIL VARCHAR(255) NOT NULL,
  LIC_NUM VARCHAR(255) NOT NULL,
  PHONE_NUMBER BIGINT(11) NOT NULL,
  PASSWORD VARCHAR(255) NOT NULL,
  GENDER VARCHAR(255) NOT NULL);
```

DESC USERS;

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
FNAME	varchar(255)	NO		NULL	
LNAME	varchar(255)	NO		NULL	
EMAIL	varchar(255)	NO	PRI	NULL	
LIC_NUM	varchar(255)	NO		NULL	
PHONE_NUMBER	bigint(11)	NO		NULL	
PASSWORD	varchar(255)	NO		NULL	
GENDER	varchar(255)	NO		NULL	

Figure 5.1.6: USER RELATION

5.2 ALTER TABLE

ALTER TABLE ADMIN

ADD PRIMARY KEY (ADMIN_ID);

ALTER TABLE BOOKING

ADD PRIMARY KEY (BOOK_ID),

ADD KEY CAR_ID (CAR_ID),

ADD KEY EMAIL (EMAIL);

ALTER TABLE CARS

ADD PRIMARY KEY (CAR_ID);

ALTER TABLE FEEDBACK

ADD PRIMARY KEY (FED_ID),

ADD KEY TEST (EMAIL);

ALTER TABLE USERS

ADD PRIMARY KEY (EMAIL);

ALTER TABLE PAYMENT

ADD PRIMARY KEY (PAY_ID),

ADD UNIQUE KEY BOOK_ID (BOOK_ID);

ALTER TABLE BOOKING

MODIFY BOOK_ID INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=71;

ALTER TABLE CARS

MODIFY CAR_ID INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=22;

ALTER TABLE FEEDBACK

MODIFY FED_ID INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

ALTER TABLE PAYMENT

MODIFY PAY_ID INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=26;

ALTER TABLE BOOKING

ADD CONSTRAINT BOOKING_IBFK_1 FOREIGN KEY (CAR_ID) REFERENCES

CARS(CAR_ID) ON DELETE CASCADE ON UPDATE CASCADE,

```
ADD CONSTRAINT BOOKING_IBFK_2 FOREIGN KEY (EMAIL) REFERENCES USERS  
(EMAIL) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE FEEDBACK
```

```
ADD CONSTRAINT TEST FOREIGN KEY (EMAIL) REFERENCES USERS (EMAIL) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE PAYMENT
```

```
ADD CONSTRAINT PAYMENT_IBFK_1 FOREIGN KEY (BOOK_ID) REFERENCES BOOKING  
(BOOK_ID) ON DELETE CASCADE ON UPDATE CASCADE;
```

5.3 FUNCTIONALITY

CONNECTION TO DATABASE:

```
<?php  
    mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);  
    $con = mysqli_connect('localhost','root','','carproject');  
    if(!$con)  
    {  
        echo 'please check your Database connection';  
    }  
?>
```

CHAPTER 6

RESULTS

6.1 HOMEPAGE

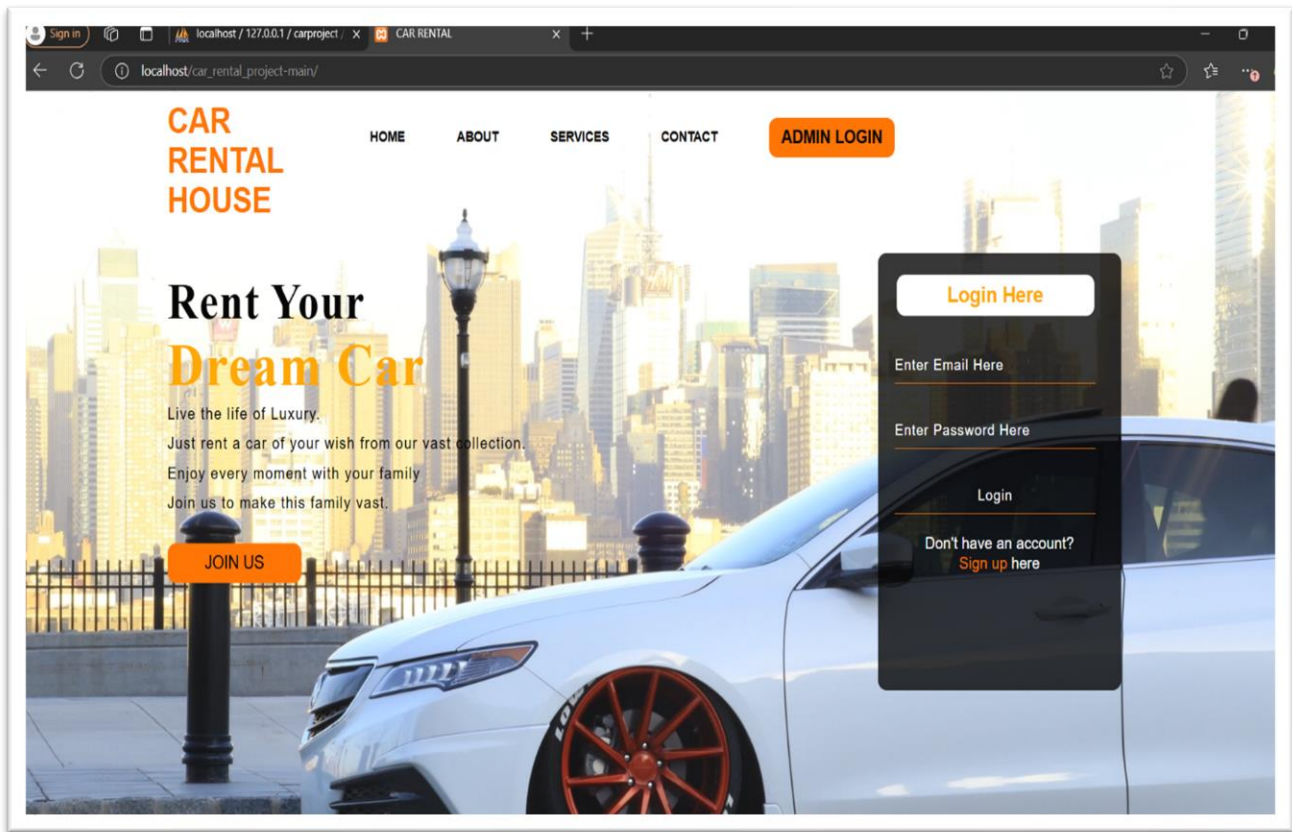


Figure 6.1: Homepage of Car Rental Management System

Figure 6.1 illustrates the homepage of the Car Rental Management System website, which serves as the primary entry point for users. It includes a navigation bar that allows seamless movement to other sections of the site, enhancing user accessibility and convenience. The homepage also introduces the website, offering essential details about its purpose and features. This design aims to engage visitors by presenting the most relevant information upfront. The layout is intuitive, ensuring that users can quickly understand the website's offerings. In addition to the navigation bar, the homepage includes a prominent search feature, allowing users to quickly look up available cars based on their preferences. The responsive design ensures that the website performs consistently across various devices, catering to a wide audience. Furthermore, testimonials or reviews from previous customers may be highlighted to build trust and credibility. Overall, the homepage sets the tone for a user-friendly and informative experience while encouraging users to explore further.

6.2 CAR OVERVIEW PAGE

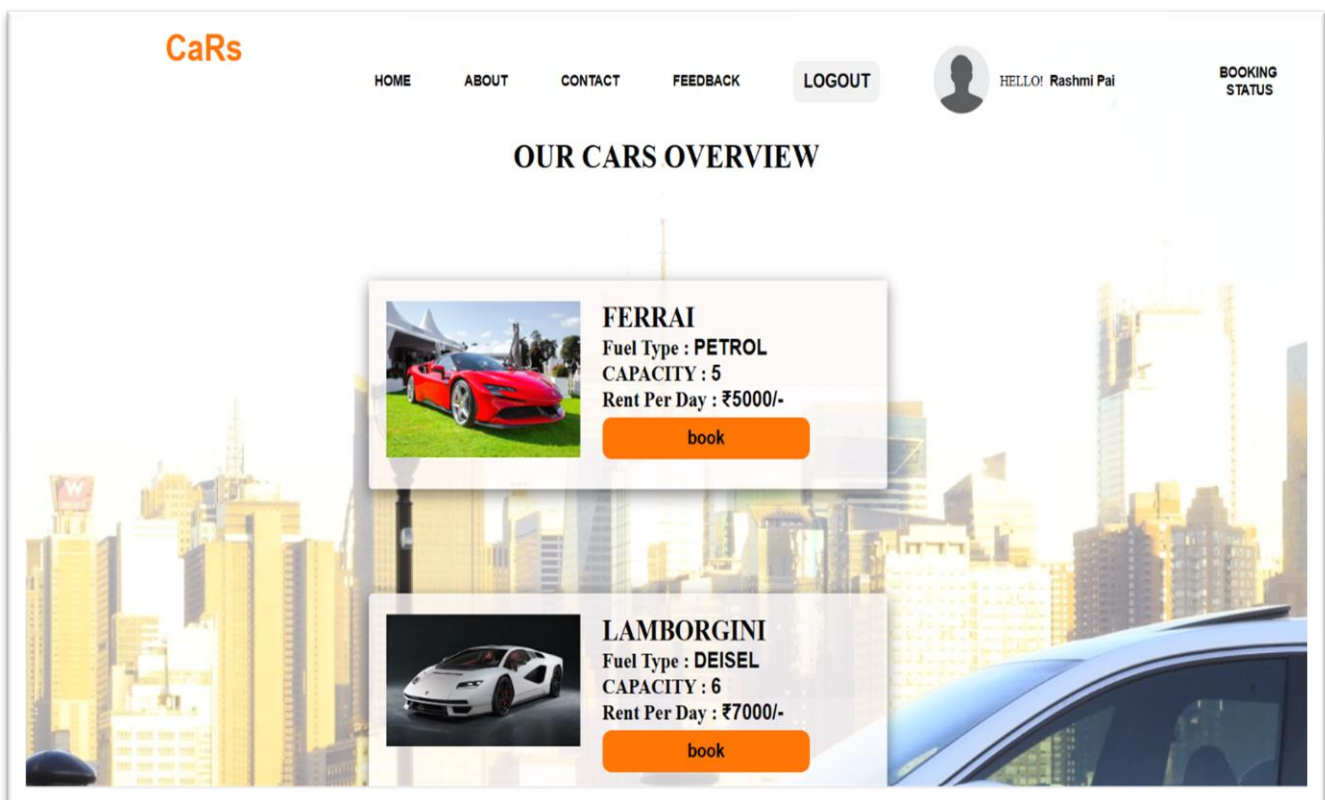


Figure 6.2: Car Rental Service Overview Interface

Figure 6.2 depicts the “Car Overview” page of the website, meticulously designed to assist users in selecting a car that aligns with their specific preferences and needs. This page serves as an interactive interface where users can effortlessly browse and filter through an extensive array of available cars. Each car listing is enriched with comprehensive details, such as the type of fuel it utilizes (petrol, diesel, or electric), seating capacity to accommodate different group sizes, and the rental cost per day to help users evaluate their budget options effectively. Additionally, the page integrates advanced filtering options, allowing users to sort cars by categories such as price range, fuel type, or seating capacity. Each car listing includes a high-quality image and user reviews or ratings to further assist decision-making. The layout is not only visually appealing but also thoughtfully designed to ensure clarity and ease of navigation. Key information is displayed prominently, minimizing the need for users to click through multiple pages to gather essential details. Responsive design ensures that the page performs seamlessly across various devices, enhancing accessibility.

6.3 BOOKING PAGE

CAR RENTAL HOUSE

USERS

FEEDBACKS

BOOKING REQUEST

LOGOUT

BOOKINGS

CAR ID	EMAIL	BOOK PLACE	BOOK DATE	PHONE NUMBER	DESTINATION	RETURN DATE	BOOKING STATUS	APPROVE	CAR RETURNED
20	rash@gmail.com	Bonda Factory	2025-01-02	9012347689	Bangalore	2025-01-14	UNDER PROCESSING	APPROVE	RETURNED
2	pranu@gmail.com	Sahyadri College	2025-01-01	9012345678	Puttur	2025-01-04	UNDER PROCESSING	APPROVE	RETURNED
1	varshagdas@gmail.com	Bonda Factory	2025-01-01	8088365346	Puttur	2025-01-04	UNDER PROCESSING	APPROVE	RETURNED
3	varshagdas@gmail.com	Adyar	2025-01-01	8088365346	Puttur	2025-01-04	UNDER PROCESSING	APPROVE	RETURNED
1	varshithvh@gmail.com	mysore	2022-03-22	6363549133	moodabidri	2022-04-02	RETURNED	APPROVE	RETURNED
2	swasthik@gmail.com	bangalore	2024-12-27	6363549133	mangalore	2024-12-31	UNDER PROCESSING	APPROVE	RETURNED

Figure 6.3: Bookings Dashboard

Figure 6.3 represents the booking page of the car rental service, where users can manage their rental bookings. It includes a table that displays details such as Car ID, Email, Booking Place, Booking Date, Phone Number, Destination, Return Date, and the current Booking Status. Users are required to provide essential details like the booking place, booking date, rental duration, phone number, destination, and return date while making a reservation. The interface also allows administrators to approve bookings through the “Approve” button and mark cars as “Returned” once the rental is complete. The system ensures transparency and clarity for both users and administrators by displaying real-time booking statuses like “Under Processing.” The page is designed with a logical and intuitive structure, enabling users to quickly locate and review their booking details. Furthermore, its responsive layout ensures seamless accessibility on various devices, enhancing the overall user experience. This comprehensive layout helps streamline the booking and return process, making it more efficient and user-friendly. Additionally, it simplifies record-keeping for the rental service by maintaining all critical data in one view.

6.4 BOOKING STATUS PAGE

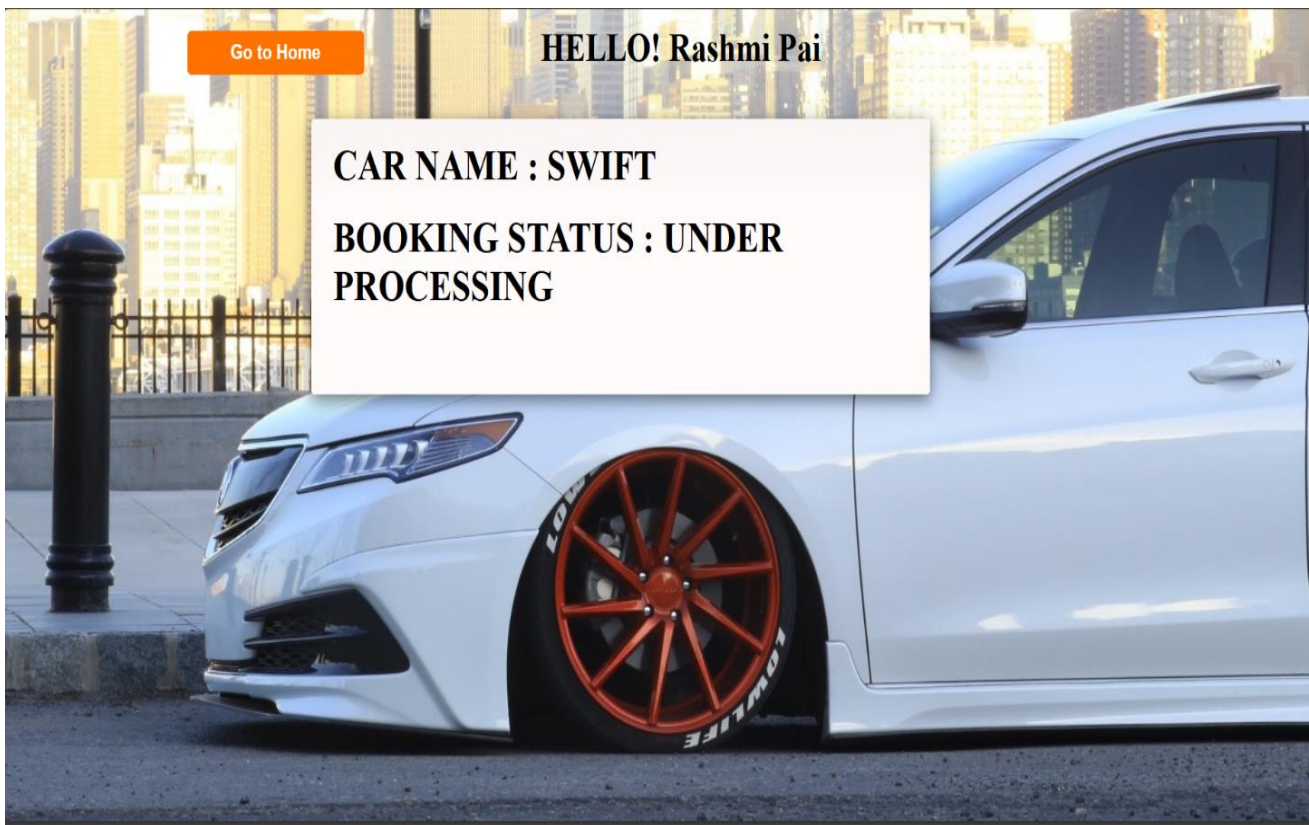


Figure 6.4: Booking Status Display Page

Figure 6.4 illustrates the booking status page of the system. This interface provides a personalized user experience by greeting the user with their name, “Rashmi Pai,” prominently displayed at the top. Below the greeting, the page outlines specific booking details, including the name of the selected car, which in this instance is “SWIFT,” and the current booking status as “UNDER PROCESSING.” The information is presented within a clean, visually distinct card positioned centrally on the page, making it easy for the user to focus on important details. A bright orange button at the top left corner labelled “Go to Home” allows users to navigate back to the homepage seamlessly. The page is fully responsive, ensuring users can access and interact with it effortlessly across devices such as smartphones, tablets, and desktops. Furthermore, the design incorporates a minimalist aesthetic, using ample white space and vibrant colours to guide user attention and maintain an uncluttered interface. This design balances functionality and visual appeal, ensuring clarity and ease of use while providing essential information briefly. This thoughtful layout effectively communicates booking information in a user-friendly and professional manner.

6.5 TRIGGER

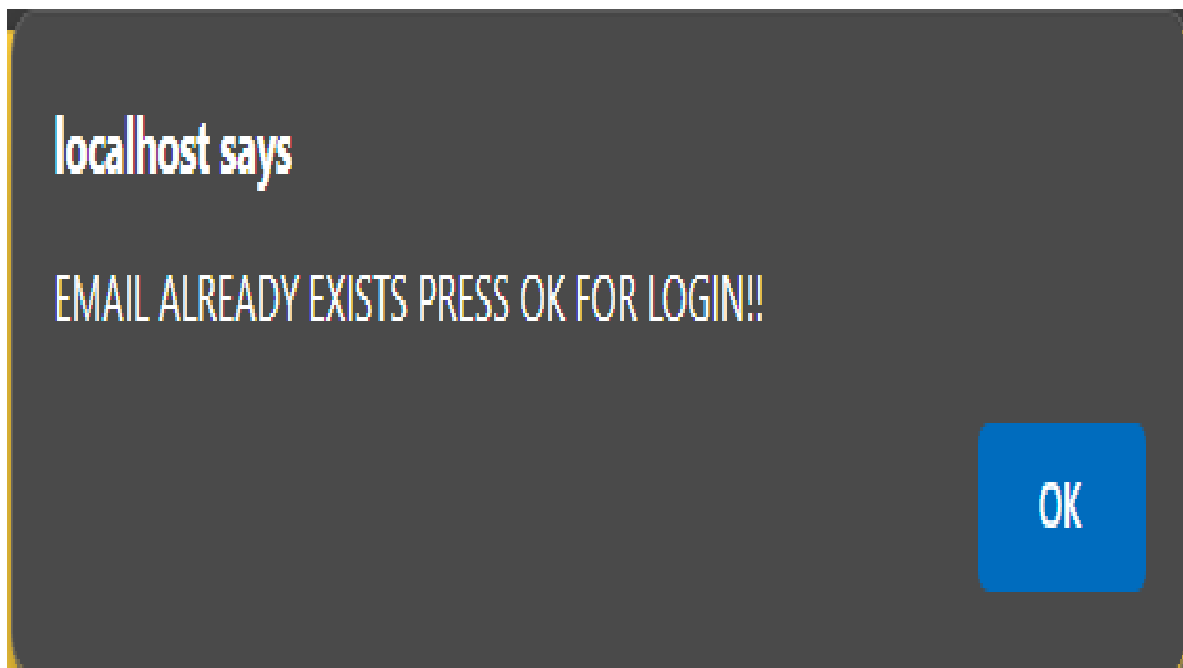


Figure 6.5: Email Registration Trigger Alert

Figure 6.5 depicts an alert box that is triggered during the user registration process in a car rental management system. This alert is displayed when a user tries to register with an email address that already exists in the system's database. The primary purpose of this alert is to inform the user that the email they entered is already associated with an account, thereby preventing duplicate entries. This ensures that the database maintains accurate and consistent user records by avoiding redundant data. The message also provides a clear action for the user, prompting them to click "OK" to proceed to the login page instead of attempting to register again. This feature enhances the overall user experience by offering a seamless transition to the login process for returning users. It reduces frustration and eliminates the need for unnecessary re-registration attempts, saving the user time and effort. From a system management perspective, this alert improves operational efficiency by ensuring the database only contains unique email entries. It also simplifies user identification and account management processes. Overall, this alert serves as an important mechanism for maintaining data integrity, improving usability, and streamlining user interactions within the car rental management system.

CHAPTER 7

CONCLUSION

The **Car Rental Management System** is a versatile, user-friendly, and highly customizable software solution designed to enhance the efficiency and effectiveness of car rental companies. By automating critical operations such as car inventory management, booking processes, feedback collection, and payment handling, the system minimizes manual effort, reduces errors, and ensures a seamless experience for both administrators and customers. One of the standout features of this system is its adaptability. It can be tailored to meet the unique requirements of businesses, making it suitable for companies of any size. The flexibility of the software ensures that it can be upgraded or extended to include additional features, catering to future needs and technological advancements. The online accessibility of the system allows businesses to reach a wider audience, providing customers with the convenience of booking and managing rentals from anywhere, at any time. The intuitive interface simplifies navigation, reducing training time for administrators and enhancing the overall user experience for customers.

Additionally, the system fosters customer satisfaction by ensuring streamlined operations and quick resolution of issues through features like automated booking confirmation and feedback management. It also supports secure payment gateways, offering customers a safe and hassle-free transaction experience. With its robust design and focus on scalability, the Car Rental Management System is a valuable tool for companies looking to maintain a competitive edge in the dynamic car rental industry. It not only addresses current operational needs but also positions businesses for growth by providing a reliable and modern foundation for future upgrades.

In conclusion, this software is an indispensable solution for car rental companies aiming to improve efficiency, enhance customer satisfaction, and achieve long-term success.

REFERENCES

- [1] M. Naga Keerthi, Dollu Bhargav Ashok Kumar "Rental Car Management System" 2024
- [2] M.Sumithra, B.Buvaneswari, S.Ahilesharan, T.Fenix Raja Singh, J. Harish 2021
- [3] Jame A Senn. "Analysis & Design of Information Systems" (2nd Edition) Mc Graw-Hill, Inc., 1989
- [4] Fitzgerald J. and Fitzgerald A. "Fundamentals of Systems Analysis, Using Structured analysis, and design techniques" (3rd Edition) John Wiley & Sons, Inc., 1992
- [5] Meilir Page-Jones "The Practical Guide to Structured System Design" (2nd Edition) Prentice-Hall International, Inc.,1988
- [6] Kenneth E. Kendall, Julie E. Kendall "Systems Analysis and Design" (2nd Edition) Prentice-Hall International, Inc., 1992
- [7] Charles P. Pfleeger "Security in Computing" Prentice-Hall, Inc., USA., 1989.
- [8] Robert Orfali, Dan Harkey, Jeri Edwards "Essential Client I Server Survival" 1994.
- [9] Ansh Agrawal, Rishabh Mathur "Online Vehicle Rental System" IJSRET, 2020
- [10] Labroo, Abhiti and Gupta, Deepak "Car Rental Management System" 2023
- [11] Qurratul Aini, "Development of Car Rental Rental Managemnet Information System" ICISBC, 2011
- [12] P. Nahnisha; B. Jai Sai Anvitha; R. Prema3 "Online Car Rental System" 2022.
- [13] Rahul Kulkarni, Chaitanya R, Pratibha, Pooja A Patil, Nikeeta "Online Car Rental System" 2021
- [14] Linghuraj P. R., Aravind K., and Giritharan R. "Car Rental Management System" 2024
- [15] Arijit Singh Yadav, Akash Shukla, Anand Srivastava, Amit Pandey "Online Car Rental System" 2023.