

**PROJECT REPORT
ON
Moto-Metric
COMPREHENSIVE VEHICLE DATA AND PRE-OWNED
MARKET PLATFORM**

Submitted By

ALVIN SIJU (SMAVBCA003)

BASIL BABU (SMAVBCA004)

VIVEK M M (SMAVBCA031)

under Guidance of

Mrs. AMBILY CB

on partial fulfilment for the award of the degree of
BACHELOR OF COMPUTER APPLICATIONS

of

UNIVERSITY OF CALICUT



DEPARTMENT OF COMPUTER APPLICATIONS

ST. MARY'S COLLEGE, SULTHAN BATHERY

DECLARATION

We hereby declare that the project report on “Moto-Metric” submitted for fulfilment of the requirements for the award of the degree of Bachelor of Computer Applications of the University of Calicut, Kerala is a Bonafide work done by me under the supervision of Mrs.Ambily C B, Assistant Professor, Department of Computer Applications, St. Mary's College, Sulthan Bathery. This submission represents my ideas in my own words and where ideas and words of others have been included. we have adequately and accurately cited and referenced the sources. we also declare that we have adhered to ethics of academic honesty and integrity and have noted that any violation of the above will be a course for disciplinary action by the institute and the University and can also evoke panel action from the sources which have thus not been appropriately cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

Name and Signature of Students:

ALVIN SIJU (SMAVBCA003)

BASIL BABU (SMAVBCA004)

VIVEK M M (SMAVBCA031)

Place: Sulthan Bathery

Date:

ST. MARY'S COLLEGE, SULTHAN BATHERY
DEPARTMENT OF COMPUTER APPLICATIONS



CERTIFICATE

This is to certify that the report entitled “**MOTO METRIC**” submitted by **ALVIN SIJU (SMAVBCA003), BASIL BABU (SMAVBCA004), VIVEK M M (SMAVBCA031)** to the University of Calicut at St. Mary's College, Sulthan Bathery in partial fulfilment of the requirements for the award of the Degree of Bachelor of Computer Applications is a bonafide record of the project work carried out by them under my guidance and supervision

Project Guide

Mrs. Ambily C B

Assistant Professor

Department of Computer Applications

Place: Sulthan Bathery

Submitted for Viva-Voice Examination on

Internal Examiner

External Exam

ACKNOWLEDGEMENT

As a collective, We express our heartfelt gratitude to everyone involved in this endeavour, contributing to its successful completion. Special thanks are extended to Mrs. Renie Mathews, the HOD of the BCA department, for not only providing the opportunity but also offering valuable guidance throughout the project, which has played a pivotal role in the culmination of my BCA journey at the college.

Our sincere appreciation goes to Mrs. Ambily C B, Mr. Anoop C Markose, Mr. Jacob Baby, Mr. Laiju George, Mrs. Elizabeth Varghese M, and Mrs. Priya, faculty members of the BCA department. Their dedication, timely suggestions, and encouragement have been instrumental in the efficient completion of the project.

A profound thank you is also extended to family members for their unwavering support, warmth, encouragement, kindness, and patience. Gratitude knows no bounds for the friends whose advice and motivation have been a constant throughout the course.

In essence, our collective thanks are extended to each and every individual who has played a role in this journey. Together, We celebrate the successful completion of this project

ABSTRACT

Introducing the **Moto Metric**. Your Compass in the World of Wheels. it is a revolutionary online platform dedicated to providing enthusiasts, buyers, and sellers with unparalleled access to the world of vehicles. From comprehensive vehicle details and effortless test drive bookings to seamless used car sales and intuitive fuel cost calculations, Moto metric redefines the automotive experience. With our user-friendly interface and cutting-edge features, Moto metric is poised to become your go-to resource for all your automotive needs. Join us on this exciting journey as WE pave the way for a smarter, more informed automotive community.

INDEX

CHAPTER 1	INTRODUCTION	8
CHAPTER 2	REQUIREMENT ANALYSIS	9-13
	2.1 Feasibility Study	
	2.2 Existing System	
	2.2.1 Problems of existing system	
CHAPTER 3	PROJECT PLAN	14 -20
	3.1 Work breakdown	
	3.2 Proposed system	
	3.2.1 Advantages	
	3.3 Modules in the system	
	3.4 Complaints statuses	
	3.5 Project Description	
	3.6 Objective & Scope	
	3.7 Advantages of the system	
	3.8 Features of the system	
CHAPTER 4	DESIGN & DEVELOPME	21-44
	4.Processes system design	
	4.2 Benefits of system design	
	4.3 Types of system design	
	4.3.1 Input Design	
	4.3.2Output Design	

4.3.3 Database Design

4.4 ER Diagram

4.5 DFD

CHAPTER 5 SOFTWARE DEVELOPMENT LIFE CYCLE 45-51

5.1 Models

5.1 Selection of the Q start model

5.1 Agile methodology

5.3.1 Advantages

5.3.2 Disadvantages

5.3.3 Application

CHAPTER 6 SYSTEM SECURITY 52-54

6.1 Security in software

6.2 Protect & security methods

CHAPTER 7 TOOLS & TECHNOLOGIES 55-71

7.1 PHP

7.2 CURSOR

7.3 Java Script

7.4 HTML

7.5 Maria DB

7.6 XAMPP

7.7 VS CODE

CHAPTER 8 SOFTWARE TESTING **75-77**

9.1 Unit testing

9.2 Integration testing

9.3 System testing

CHAPTER 9 SYSTEM IMPLEMENTATION **78-79**

CHAPTER 10 SYSTEM MAINTENANCE **80**

CHAPTER 11 CONCLUSION & FUTURE SCOPE **81**

APPENDIX **82-97**

Screenshots of the website

Sample Code

BIBLIOGRAPHY **98**

Chapter 1

INTRODUCTION

Welcome to Moto-metric: Your Compass in the World of Wheels

Moto-metric is more than just a website; it's your one-stop destination for all things related to vehicles. Whether you're a car enthusiast, a prospective buyer, or simply seeking information, Moto-metric has you covered. Our platform offers a comprehensive range of features designed to empower you in your automotive journey.

Explore Extensive Vehicle Details:

Discover in-depth information about a vast array of vehicles, from the latest models to timeless classics. Whether you're interested in sedans, SUVs, trucks, or hybrids, Moto-metric provides detailed specifications, performance metrics, and expert reviews to help you make informed decisions.

Effortless Car Test Drive Booking:

Experience the thrill of your dream car with ease by utilizing our convenient test drive booking function. Simply select your desired vehicle, choose a preferred date and time, and leave the rest to us. Our seamless booking process ensures a hassle-free experience, allowing you to focus on what truly matters – the drive.

Browse and Sell Used Cars:

In the market for a pre-owned vehicle? Moto-metric offers a diverse selection of used cars from trusted sellers. With our intuitive search functionality and comprehensive listings, finding the perfect used car has never been easier. Plus, if you're looking to sell your vehicle, our platform provides a seamless avenue to connect with potential buyers.

Calculate Your Fuel Costs:

Take the guesswork out of budgeting with our fuel cost calculator. Compare fuel efficiency across different models and calculate your average fuel expenses per week, month, and year. With this valuable tool, you can make informed decisions that align with your budget and lifestyle.

Dynamic Car Comparison:

Unable to decide between two vehicles? Our dynamic car comparison feature allows you to directly compare key attributes side by side. From fuel efficiency and performance to safety features and amenities, gain valuable insights to help you choose the perfect ride for your needs.

Moto-metric isn't just a website; it's your trusted companion on the road to automotive enlightenment. Whether you're researching your next purchase, scheduling a test drive, or simply indulging your passion for cars, Moto-metric is here to guide and inspire you every step of the way. Welcome to a world where the journey is as exciting as the destination – welcome to Moto-metric.

CHAPTER 2

REQUIREMENT STUDY AND ANALYSIS

Requirement analysis is the first stage in the systems engineering process and Software development process. Requirements analysis in systems engineering and software engineering encompasses those tasks that go into determining the needs or conditions to make for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.

Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, and related to identified business needs or opportunities and defined to the level of detail sufficient for system design. The primary goal of the system analyst is to improve the efficiency of the existing system and for that, specific requirements are very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. Investigations have been done on whether the upgrade of the system into an application program could solve the problems and eradicate the inefficiency of the existing system. The process of system analysis is largely concerned with determining, developing, and agreeing with users' requirements. The system analyst should use this opportunity to communicate well with users and conceive and understand what a system should be done together with a view of the relative importance of the system. It involves studying the ways an organisation currently retrieves and processes data to produce information, with the intention of finding out how to make things better. System analysis includes investigation and suggesting possible changes to the existing system

Requirement analysis produces a description of Software Requirements Specification (SRS), which is a complete description of the behaviour of the system to be developed. System analysis includes investigation and possible changes to the existing system. The analysis is used to gain an understanding of the existing system and what is required for that. After the system analysis, there is the system description and set of requirements for a new system. If there is no such existing system the analysis only defines their requirements. This new system may be built or by changing the existing system. Development begins by defining a model of the new system and continues this model to a working system. The model of the system shows what the system must do to satisfy these requirements. Finally, data models are converted to a database and processed to use procedures and computer programs.

2.1 FEASIBILITY STUDY

1. Technical Feasibility:

- Assess the technical requirements and challenges of developing the Moto-metric website.
- Evaluate the availability of necessary technologies, tools, and expertise.
- Ensure that the technical infrastructure can support the expected user load and features.

2. Financial Feasibility:

- Estimate the costs associated with developing, launching, and maintaining the website.
- Consider funding sources and evaluate the potential return on investment (ROI).
- Assess whether the project is financially viable within the allocated budget.

3. Operational Feasibility:

- Evaluate how well the Moto-metric website aligns with the operational goals and strategies.
- Consider the impact on existing processes and workflows.
- Assess the readiness of the organization to implement and manage the platform.

4. Market Feasibility:

- Conduct market research to understand the demand for a platform like Moto-metric.
- Analyse the target audience, their needs, and preferences.

- Evaluate the competitiveness of the online used car marketplace.

5. Legal and Regulatory Feasibility:

- Identify and analyse legal and regulatory requirements related to online car sales.
- Ensure compliance with data protection laws, consumer protection regulations, and other relevant statutes.
- Assess potential legal challenges and risks associated with the project.

6. Schedule Feasibility:

- Develop a realistic timeline for the development and launch of the Moto-metric website.
- Consider potential delays, dependencies, and external factors that could impact the schedule.
- Ensure that the project can be completed within a reasonable timeframe.

7. Resource Feasibility:

- Assess the availability of human resources, skills, and expertise required for the project.
- Identify any potential resource constraints and develop plans to address them.
- Ensure that the project team has the capacity to execute the development and maintenance tasks.

8. Risk Analysis:

- Identify potential risks and uncertainties associated with the Moto-metric project.
- Develop strategies to mitigate or manage these risks effectively.

- Evaluate the impact of potential risks on the success of the project.

9. Social and Environmental Feasibility:

- Consider the social and environmental impact of the Moto-metric website.
- Assess whether the platform aligns with ethical standards and environmental sustainability goals.

The feasibility study should provide a comprehensive understanding of whether the Moto-metric website is a viable and worthwhile project. It helps stakeholders make informed decisions about proceeding with the development based on a thorough analysis of various factors.

The feasibility study is performed at four levels:

1. **Technical feasibility**
2. **Economical feasibility**
3. **Behavioural feasibility**
4. **Operational feasibility**

2.1.1 TECHNICAL FEASIBILITY

The process of proving the concept is technically possible. The objective of the technical feasibility step is to confirm that the product will perform and verify that there are no production barriers. The product of this activity is a working model.

The technical feasibility step generates knowledge about the product of the process design, performance, production requirements, and preliminary production costs. Technical feasibility assesses the current resources (including hardware and software) and technology which are required to accomplish the user requirements in the system within the allocated time and budget. It is concerned with the existing computer

system (hardware and software) and to what extent it can support the proposed system. Since it is not a complex system, WE have the technical feasibility to develop the system.

2.1.2 ECONOMICAL FEASIBILITY

The economic feasibility term is whether the proposed system can generate financial gains for an organisation. It involves the cost incurred on the software development team, estimated cost of hardware, and cost of performing a feasibility study is economically feasible since the incurred for the development of the system produces long-term gains. The cost of hardware and software for the class of applications of our proposed system is less since it is an automated system where no Employees are needed. So, WE can save the monthly wages which makes this system economically feasible. Hence the proposed system is economically feasible.

Economical feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential Factors involved in this field of study.

The highly qualified and experienced professionals from the related field perform the research with excellent proficiency. While WE do the study, WE always take care of the essential factors needed to carry out a successful project. The study is based on cost and time. Under the cost- based study, WE evaluate the development cost and the operating cost. WE also calculate an approximate time frame to receive returns against Investment keeping in mind the future value of the project. During the process of the economic feasibility study, We follow certain best practices to get the desired result. WE make certain assumptions based on which We give you a solid plan of investment.

- **Economic feasibility cash flow**
- **Estimated total project cost**
- **Estimated total earnings**
- **Risk factors**

- **Cost benefits**

2.1.3 BEHAVIOURAL FEASIBILITY

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerised system. It is common knowledge that computer installation candidate systems require special effort to educate, sell, and train the staff on new ways of conducting.

2.1.4 OPERATIONAL FEASIBILITY

The proposed project would be beneficial to the organisation in satisfying the objective when developed and installed. One of the main problems faced during the development of a new system is getting acceptance from the management of the organisation towards the development of the project. All the operational aspects are considered carefully. Thus, the project is operationally feasible. Anyone with minimum system knowledge can use this system.

A system that has operational feasibility will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. Proposed systems are beneficial only if they can be turned into information systems that will meet the organisation's operating requirements. People are inheritably resistant to change. Computers have been known to facilitate change. An estimate should be made to know how strong the reaction of a User-staff is likely to have towards the development of a computerised system. Then this System is ready to use in the organisation if the system is operationally feasible. As this package is technically, economically, and functionally feasible, the system is judged feasible. Viewing the collected information, recommendation, and justification, a conclusion is made of the proposed system. Since the project is an academic project, conducting the various feasibility studies is beyond the scope due to time constraints and resources.

2.2 EXISTING SYSTEM

The current scenario for buying and selling used cars involves multiple fragmented platforms, both online and offline. Prospective buyers and sellers navigate

through various websites, classifieds, and dealership listings to find the right match for their preferences. The lack of a centralized platform often results in a time-consuming and less efficient process.

looking for specific car models, detailed comparisons, and a seamless buying or selling experience face challenges in consolidating information from different sources. Moreover, there is a noticeable absence of a dynamic and engaging platform that allows users to not only explore car listings but also compare various models side by side.

2.2.1 Problems in Existing System:

1. Fragmented Experience:

- Users have to visit multiple platforms to get information about different cars, making the process cumbersome and time-consuming.

2. Limited Comparison Options:

- There is a lack of a dedicated feature that enables users to compare different cars comprehensively, including specifications, pricing, and other relevant details.
-

3. Inefficient Buying/Selling Process:

- The current system lacks a streamlined process for buying and selling used cars, resulting in potential buyers and sellers facing difficulties in connecting with each other.

4. Static Car Listings:

- The existing system may provide static car listings, but it lacks an engaging and dynamic interface, making the user experience less interactive.

5. Limited User Interaction:

- Users might not have the ability to interact with the platform beyond basic searches, making it challenging to find the right information and make informed decisions.

Moto metric Solution:

Moto-metric aims to address these challenges by providing a centralized platform that offers a comprehensive solution for users interested in buying, selling, and comparing used cars. With dynamic scrolling for an engaging user experience, detailed car listings, and an intuitive comparison feature, Moto-metric seeks to streamline the process and enhance the overall user journey in the world of used car transactions.

CHAPTER 3

PROJECT PLAN

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. Project planning is often used to organise different areas of a project, including project plans, workloads and the management of teams and individuals. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Project planning is inherently uncertain as it must be done before the project is actually started. Therefore the duration of the tasks is often estimated through a weighted average of optimistic, normal, and pessimistic cases. The critical chain method adds "buffers" in the planning to anticipate potential delays in project execution. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project schedule may be optimised to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the project schedule becomes what is known as the baseline schedule.

Progress will be measured against the baseline schedule throughout the life of the project. Analysing progress compared to the baseline schedule is known as earned value management. In this project, the basic goal of planning was to look into the future, identify the activities that need to be done to complete the project successfully, and plan the schedule and resource allocation for the activities. The activities that were identified

are, Identification of Need, Feasibility Study, Data Collection, Structured Analysis, Design, Coding, Testing and Implementation.

3.2 PROPOSED SYSTEM

Moto metric envisions a revolutionary shift in the landscape of buying, selling, and comparing used cars by introducing a comprehensive and user-friendly platform. The proposed system is designed to address the shortcomings of the existing model, providing users with a seamless, centralized experience. Moto metric will serve as a one-stop solution, offering dynamic features that simplify the car transaction process and enhance user interaction.

Advantages of Proposed System:

1. Centralized Platform:

- Moto metric consolidates the car buying and selling experience into a centralized platform, eliminating the need for users to navigate multiple sources.

2. Comprehensive Car Comparison:

- The proposed system introduces a dedicated feature for users to compare different cars comprehensively, including specifications, pricing, and other relevant details. This empowers users to make informed decisions based on their preferences.

3. Streamlined Buying/Selling Process:

- Moto metric streamlines the buying and selling process by providing a user-friendly interface that facilitates efficient connections between potential buyers and sellers. This helps in reducing the complexities associated with transactions.

4. Dynamic and Engaging Interface:

- With dynamic scrolling and an engaging interface, Moto metric enhances the user experience, making the exploration of car listings more interactive and enjoyable.

5. Enhanced User Interaction:

- Moto metric encourages user interaction beyond basic searches, allowing users to engage with the platform in various ways. This includes comparing cars, exploring detailed listings, and participating in the buying and selling process seamlessly.

6. Detailed Car Listings:

- The proposed system ensures that car listings are not only comprehensive but also presented in a detailed manner. This helps users gather all the necessary information about a particular car without having to visit multiple sources.

7. User-Friendly Design:

- Moto metric prioritizes a user-friendly design, ensuring that the platform is accessible and easy to navigate for users of all backgrounds and technical expertise.

8. Secure Transactions:

- Security is a paramount concern. The proposed system implements robust security measures to protect user data and ensure secure transactions, instilling confidence in users when engaging in buying or selling activities.

Moto metric aims to redefine the user experience in the used car market, offering a platform that not only addresses existing challenges but also sets a new standard for efficiency, engagement, and reliability in the online car marketplace.

3.3 COMPREHENSIVE VEHICLE DATA AND PREOWNED MARKET PLATFORM

- 1. User Registration**
- 2. Search**
- 3. Used Cars**
- 4. Comparison**
- 5. Booking**
- 6. Car Listing and Details**
- 7. Community forum**

1. User Registration:

- Allows users to create accounts on the platform.
- Captures essential user information for personalized experiences.
- Provides secure login credentials for future access.

2. Search:

- Enables users to search for specific vehicles based on various criteria.

- Supports filters for make, model, price range, and other relevant parameters.
- Enhances user experience by delivering accurate and tailored search results.

3. Used Cars:

- Dedicated section for listing and browsing available used cars.
- Displays comprehensive information about each vehicle, including specifications and pricing.
- Facilitates the buying and selling process for both buyers and sellers.

4. Comparison:

- Allows users to compare multiple cars side by side.
- Compares specifications, features, and pricing to aid decision-making.
- Enhances the user's ability to make informed choices.

5. Booking:

- Provides a booking feature for users interested in a particular car.
- Enables users to schedule test drives or reserve a vehicle.
- Streamlines the booking process for both buyers and sellers.

6. Car Listing and Details:

- Presents a detailed listing of all available cars on the platform.
- Includes comprehensive information such as images, descriptions, and specifications.
- Facilitates easy navigation and exploration for users.
-

These modules collectively form a robust comprehensive vehicle data and preowned market place, offering a user-friendly and comprehensive

experience for individuals interested in buying, selling, or comparing used cars.

3.5 PROJECT DESCRIPTION

Moto metric – Your Compass in the World of Wheels

Moto metric is an innovative online platform that revolutionizes the experience of buying, selling, and comparing used cars. This comprehensive website serves as a centralized hub for automotive enthusiasts, providing a user-friendly interface and dynamic features for seamless transactions and informed decision making.

Key Features:

1. User Registration:

- Create a personalized account to unlock a tailored experience.
- Secure login credentials for hassle-free access.

2. Search:

- Effortlessly search for your dream car using various criteria.
- Filters for make, model, price range, and more for precise results.

3. Used Cars:

- Explore a dedicated section featuring an extensive range of used cars.
- Detailed listings with comprehensive information, including specifications and pricing.

4. Comparison:

- Make informed decisions by comparing multiple cars side by side.
- Comprehensive comparisons of specifications, features, and pricing.

5. Booking:

- Reserve or schedule a test drive for your preferred vehicle.
- Streamlined booking process for both buyers and sellers.

6. Car Listing and Details:

- Navigate through a visually engaging interface with dynamic scrolling.
- In-depth details, images, and specifications for each listed car.

Advantages:

- Centralized Platform:
- Consolidate your entire car-buying experience in one place.
- Comprehensive Comparison:
- Empower yourself with the ability to compare and choose the best fit.
- User-Friendly Design:
- Navigate seamlessly with an intuitive and accessible design.
- Secure Transactions:
- Conduct transactions with confidence, knowing that your data is secure.
- Engaging Interface:
- Enjoy a dynamic and visually appealing platform for a memorable user experience.

Moto metric is not just a website; it's a transformative approach to the used car market. Whether you're a buyer looking for your next ride, a seller seeking a streamlined platform, or a car enthusiast exploring the

latest models, Moto metric is your go-to destination for all things automotive.

Experience the future of used car transactions with Moto metric – where every journey begins with a click!

3.6 OBJECTIVE AND SCOPE

1. Create a Centralized Platform:

- Establish Moto metric as a centralized hub for buying, selling, and comparing used cars, simplifying the user experience.

2. User Registration and Authentication:

- Implement a secure and user-friendly registration system to allow users to create accounts and authenticate their identities.

3. Efficient Search Functionality:

- Develop an efficient and accurate search system, enabling users to find specific vehicles based on various criteria such as make, model, and price range.

4. Comprehensive Used Cars Section:

- Create a dedicated section for listing and browsing used cars with detailed information, facilitating both buyers and sellers.

5. Car Comparison Feature:

- Introduce a robust car comparison feature, allowing users to make informed decisions by comparing specifications, features, and pricing of multiple cars.

6. Booking System:

- Implement a secure and streamlined booking system for users interested in reserving or scheduling test drives for specific vehicles.

7. Dynamic and Engaging Interface:

- Design a visually appealing and dynamic interface with features like dynamic scrolling, providing an engaging user experience.

8. Detailed Car Listings:

- Ensure comprehensive and detailed listings for each car, including images, descriptions, and specifications, to assist users in making informed choices.

9. User-Friendly Design:

- Prioritize a user-friendly design to enhance accessibility and navigation for users with varying levels of technical expertise.

10. Security Measures:

- Implement robust security measures to protect user data, ensuring secure transactions and maintaining user trust.

11. Encourage User Interaction:

- Facilitate user interaction beyond basic searches, allowing users to actively engage with the platform through features like reviews, ratings, and comments.

12. Optimize Performance:

- Optimize the website's performance to ensure a smooth and responsive user experience, regardless of the device or browser used.

13. Legal Compliance:

- Ensure that the website adheres to relevant legal and regulatory requirements, especially regarding user data privacy and online transactions.

14. Promote a Positive Marketplace Experience:

- Foster a positive online marketplace experience by providing a reliable, transparent, and trustworthy platform for users.

These objectives collectively contribute to the overarching goal of making Moto metric a go-to destination for individuals involved in the used car market, offering a comprehensive and user-centric solution to their needs.

Scope:

The future scope for Moto metric is promising and expansive. As technology continues to advance and consumer preferences evolve, there are several avenues through which Moto metric can further enhance its offerings and solidify its position as a leader in the automotive industry:

1.Enhanced User Experience: Constant refinement of the platform's user interface and experience will ensure that Moto metric remains intuitive, accessible, and enjoyable for users of all backgrounds and skill levels.

2.Integration of Emerging Technologies: Embracing emerging technologies such as artificial intelligence, augmented reality, and machine learning can revolutionize how users interact with the platform, offering personalized recommendations, predictive analytics, and immersive experiences.

3.Expansion of Services: Diversifying the range of services offered, such as vehicle financing, insurance comparisons, and maintenance scheduling, can transform Motometric into a comprehensive automotive ecosystem that caters to all aspects of vehicle ownership and management.

4.Global Expansion: Scaling the platform to cater to international markets will unlock new opportunities for growth and engagement, allowing Motometric to serve a diverse range of users with unique needs and preferences worldwide.

5.Sustainability Initiatives: Integrating sustainability-focused features, such as carbon footprint calculations, eco-friendly vehicle recommendations, and information on alternative fuel options, can align Moto metric with the growing demand for environmentally conscious transportation solutions.

By continually evolving, adapting, and innovating, Moto metric can remain at the forefront of the automotive industry, empowering users and shaping the future of mobility in meaningful and impactful ways.

3.7 ADVANTAGES:

1. Centralized Platform:

- All-in-one solution for buying, selling, and comparing used cars.

2. Efficient Search:

- Quick and tailored vehicle searches for users.

3. Comprehensive Listings:

- Wide range of detailed car listings for buyers and sellers.

4. Car Comparison:

- Informed decision-making with a robust comparison feature.

5. Streamlined Booking:

- Simplified, secure booking system for test drives.

6. Engaging Interface:

- Visually appealing and dynamic design for an enjoyable experience.

7. Detailed Listings:

- Comprehensive information for confident decision-making.

8. User-Friendly Design:

- Easy navigation for users with varying technical expertise.

9. Security Measures:

- Robust security for data protection and secure transactions.

10. User Interaction:

- Active engagement through reviews, ratings, and comments.

11. Optimized Performance:

- Smooth and responsive user experience.

12. Positive Marketplace Experience:

- Reliable, transparent, and trustworthy online marketplace.

These concise advantages highlight the key benefits of the Moto metric website project.

3.8 Features:

1. User Registration:

- Simple and secure registration process for creating personalized accounts.
- 2. Advanced Search Filters:**
 - Efficient search functionality with filters for make, model, price range, and more.
- 3. Comprehensive Listings:**
 - Detailed listings of used cars with specifications, images, and pricing.
- 4. Car Comparison Tool:**
 - Robust comparison feature allowing users to compare multiple cars side by side.
- 5. Secure Booking System:**
 - Streamlined booking system for test drives or reservations.
- 6. Dynamic Interface with Scrolling:**
 - Engaging and dynamic design with visually appealing scrolling features.
- 7. Detailed Car Information:**
 - In-depth information on each listed car, aiding informed decision-making.
- 8. User Reviews:**
 - Feature for users to leave, enhancing transparency.
- 9. Mobile-Friendly Design:**
 - Responsive design ensuring a seamless experience across devices.
- 10. Interactive User Dashboard:**
 - Personalized dashboard for users to manage their preferences and activities.
- 11. User Feedback Mechanism:**
 - Feature for users to provide feedback, contributing to platform improvement.
- 12. Quick Contact Options:**
 - Convenient options for users to reach out to sellers or buyers.

These features collectively contribute to making Moto metric a comprehensive and user-friendly platform within the used car market.

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

System Designing involves the analysis, design, and configuration of the necessary hardware and software components to support your solution's architecture. The purpose of System Design is to create an online job portal that is supposed to provide an online facility to offer job vacancies. At this point in the project lifecycle, there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organisational entities that will use the new system. The challenge is to translate all of this information into Technical Specifications that accurately describe the design of the system, and that can be used as input to System Construction.

4.1 Processes in System Designing

This phase consists of the following processes:

- **Prepare for System Designing** where the existing project repositories are expanded to accommodate the design work products, the technical environment and tools needed to support System Design are established, and training needs of the team members involved in System Design are addressed.
- **Define Technical Architecture**
Where the foundation and structure of the system are identified in terms of system hardware, system software, and supporting tools, and the strategy is developed for the distribution of the various system components across the architecture.
- **Define System Standards**

Where common processes, techniques, tools, and conventions that will be used throughout the project are identified in an attempt to maximise efficiencies and introduce uniformity throughout the system.

- **Create a Physical Database** where the actual database to be used by the system is defined, validated, and optimised to ensure the completeness, accuracy, and reliability of the data.
- **Prototype System Components** where various components of the solution may be developed or demonstrated in an attempt to validate preliminary functionality, to better illustrate and confirm the proposed solution, or to demonstrate “proof-of-concept.”

- **Produce Technical Specifications**

Where the operational requirements of the system are translated into a series of technical design specifications for all components of the system, setting the stage for System Construction.

4.2 Benefits of system design

A System Design engagement typically provides the following benefits:

- **Improved system performance:**
Individually tailored configuration advice demonstrates where improvement is necessary, and how to improve the system to regain lost performance.
- **Understanding:**
Customers gain a detailed understanding of how their users use their system. This Usage Profile can be leveraged to develop future architecture changes.
- **Potential:**
Potential to learn of future concerns, allowing customers to take protective measures to avoid problems.
- **Performance:**
A baseline performance level is established against which benefits can be compared and changes to the system predicted or foreseen.

4.3 Types of system design

4.3.1 Input Design

Input design is the process of converting user-oriented input to a computer-based format. Input design is a part of the overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system.

The main objectives of the input design are:

1. Produce a cost-effective method of input
2. Achieve the highest possible level of accuracy
3. Ensure that the input is acceptable to and understood by the staff.

Input Data

The goal of designing input data is to make entry easy, logical, and free from errors as possible. The entering data entry operators need to know the allocated space for each field, field sequence and which must match with that in the source document. The format in which the data fields are entered should be given in the input form.

Here data is input from the user in the end then it makes use of a processor that accepts commands and data from the operator through a keyboard. The input required is analysed by the processor. It is then accepted or rejected. Input stages include the following processes

- Data Recording
- Data Transcription
- Data Conversion
- Data Verification
- Data Control
- Data Transmission

One of the aims of the system analyst must be to select data capture methods and devices, which reduce the number of stages so as to reduce both the changes of errors and the cost. Input types can be characterised as

- External
- Internal
- Operational
- Computerised
- Interactive

4.3.2 Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of this result for later consultation. Computer output is the most important and direct source of information for users. Designing computer output should proceed in an organised well throughout the manner. The right output must be available for the people who find the system easy to use. The outputs have been defined during the logical design stage. If not, they should be defined at the beginning of the output designing terms of types of output connect, format, response, etc.

Various types of outputs are

- External outputs
- Internal outputs
- Operational outputs
- Interactive outputs
- Turn around outputs

All screens are informative and interactive in such a way that the user can fill his requirements by asking queries.

4.3.3 Database design

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organised around user requirements. The general objective is to make information accessible, easily quick, inexpensive, and flexible for other users. During database design the following objectives are concerned:

- Controlled Redundancy
- Data independence
- Accurate and integrating
- More information at a low cost
- Recovery from failure
- Privacy and security
- Performance
- Ease of learning and use

User Interface

- **Greater Accessibility** – you have more capability at your cursor tip.
- **Lower Cognitive Load** – By having everything laid out in front of you, you don't have to remember a lot of mundane things (like the proper formatting and the list of text commands needed to copy a document). The GUWE takes care of most of that, freeing up your mental processing power for the important stuff.
- **Higher Productivity** – when you get down to it the GUWE is all about productivity.

4.4 ER DIAGRAM

An entity-relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them, and databases. An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain. When We speak of an entity, We normally speak of some aspect of the real world that can be distinguished from other aspects of the real world. A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in the database, so by showing the relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

Entities and relationships can both have attributes. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

Entity-relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets and relationship sets.

Components of ER Diagram

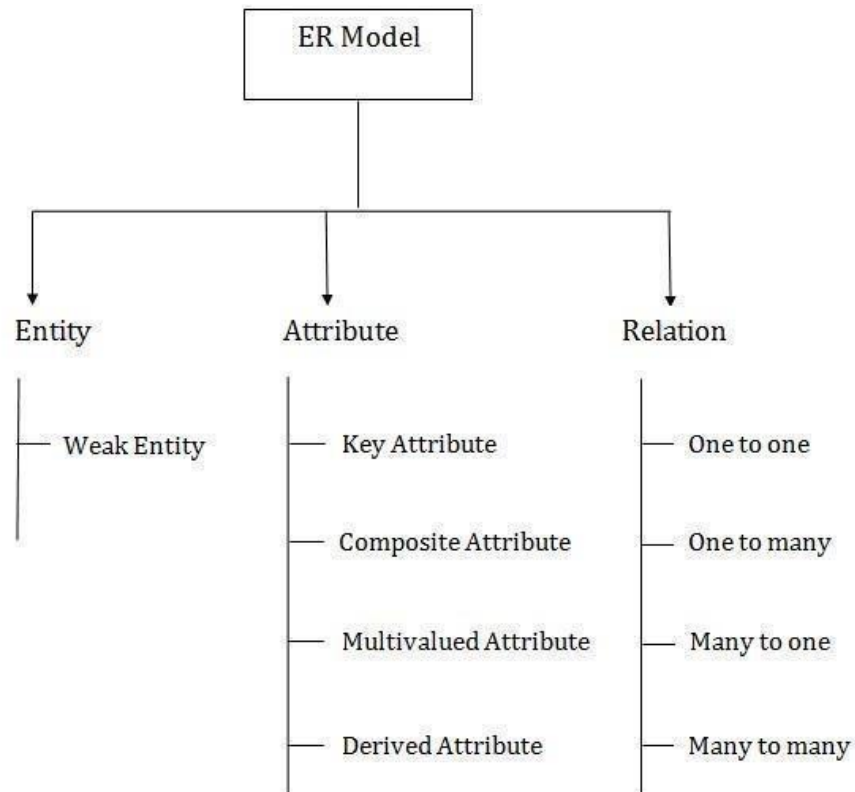
Entity

Attribute

An attribute describes the property of an entity. An attribute is represented as an Oval in an ER diagram.

Relationship

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.



ER Diagram Symbols and Notations

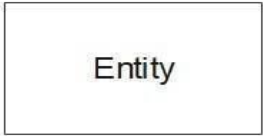
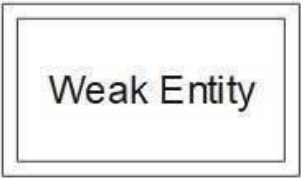
ER diagrams consist of three major things – Entity, Relationship, and Attribute.

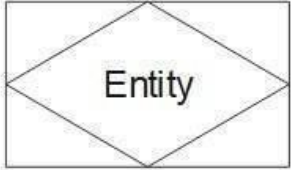
Therefore, let's uncover various Entity-relationship diagram notations based on these components.

ER Diagram Entity Symbols

An entity can be any real-world object that forms the basic structure of a database. For instance, a person, a tree, an organisation, an event, or any other living/non-living thing can be an entity.

The functions include executing the required matters, checking the correctness of the current system, and checking the existing system's mistakes.

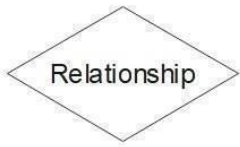

Symbol	Name	Description
	Entity	This is a basic entity that is represented by a rectangle with its name inside.
	Weak Entity	This is an entity that can't solely be identified with its attributes (due to the absence of a primary key). It inherits the identifier of its parent entity and often integrates it with a partial key.

	Associative Entity	This is a special entity that is mostly used in manytomany relationships with all its relationships as “many”.
---	--------------------	--

ER Diagram Relationship Symbols

In an ER diagram, relationships would simply define how two or more entities are connected to each other.

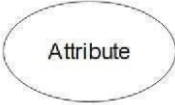
Symbol	Name	Description
--------	------	-------------


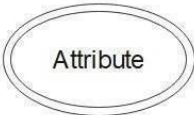
	Strong Relationship	A strong relationship is depicted by a single rhombus with its name inside. In this, an entity is independent – that is, its primary key for any child doesn't contain the primary key of the linked entity.
	Weak or Identifying Relationship	A weak relationship is depicted by a double rhombus with the name inside. In this, the child is dependent on the parent entity as its primary key would contain a component of the parent's primary key.


ER Diagram Attribute Symbols

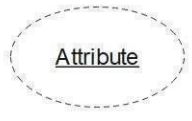
In any set of ER diagram symbols, you can find all kinds of attributes that would define the value or property of any entity. For instance, if a pen is an entity, then its attributes could be colour, size, material, and so on.

Symbol	Name	Description
--------	------	-------------

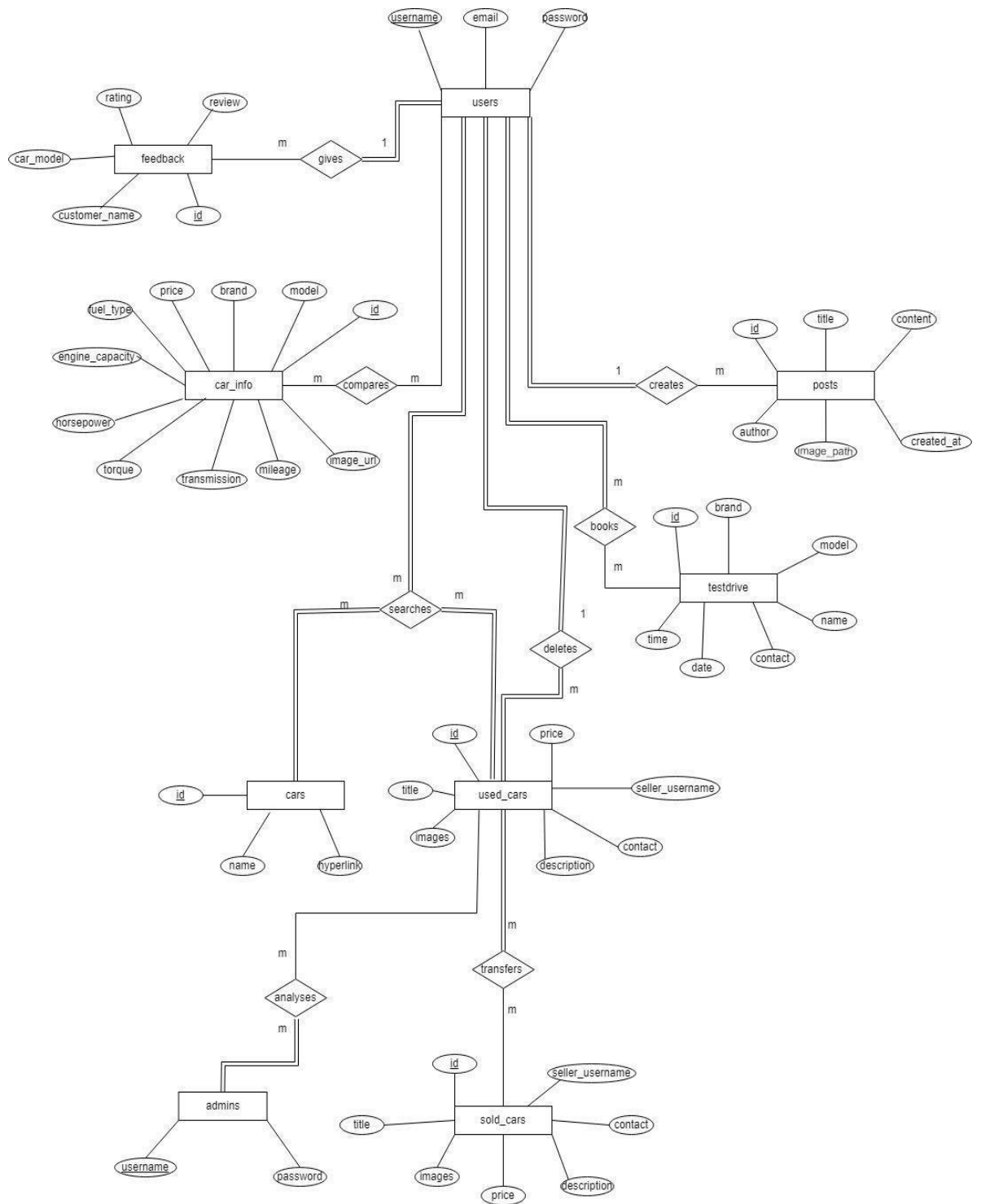
	Attribute	A basic attribute is represented by a single oval with its name written inside.
---	-----------	---

	Key Attribute	This is a special attribute that is used to uniquely identify an entity. It is represented by an oval with its name underlined.
	Multi-valued Attribute	These are the attributes that can have multiple values (like the Name attribute can have a First and Last name) and are represented by a double oval.

	<p>Derived Attribute</p>	<p>A derived attribute might not be physically present in the database and could be logically derived from any other attribute (represented by a dotted oval).</p>
---	------------------------------	--

	<p>Weak Key Attribute</p>	<p>It is an attribute that might be derived from any other attribute, but it would have unique identifiers for the entity. It is represented by a dotted oval with its name underlined.</p>
---	-------------------------------	---

ER Diagram



4.5 DATA FLOW DIAGRAM (DFD)

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have a control flow and no loops or decision rules are present. A flowchart can explain specific operations depending on the type of data.

It is a graphical tool, useful for communicating with users, managers, and other personnel. It is useful for analysing existing as well as proposed systems.

Does it provide an overview of

- What data is system processes?
- What transformations are performed?
- What data is stored?
- What results are produced, etc.

Data Flow Diagrams can be represented in several ways. The DFD belongs to structured analysis modelling tools. Data Flow diagrams are very popular because they help us to visualise the major steps and data involved in software-system processes.

Components of DFD

The Data Flow Diagram has 4 components:

- **Process:** Input to output transformation in a system occurs because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle, or circle. The process is named a short sentence, in one word or a phrase to express its essence
- **Data Flow:** describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modelled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bidirectional.
- **Warehouse:** The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a

data file rather it can be anything like a folder with documents, an optical disc, or a filing cabinet. The data warehouse can be viewed independently of its implementation. When the data flows from the warehouse it is considered data reading and when data flows to the warehouse it is called data entry or data updation.

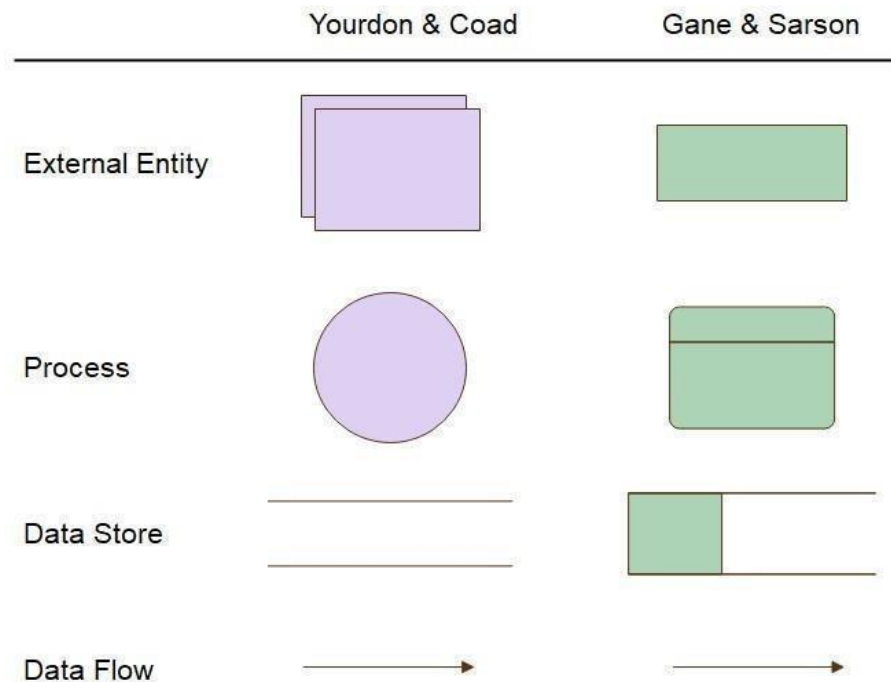
- **Terminator:** The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organisations like banks, groups of people like customers, or different departments of the same organisation, which is not a part of the model system and is an external entity.

Modelled systems also communicate with terminators.

Data Flow Diagram Symbols

DFD symbols are consistent notations that depict a system or a process. It entails the use of short-text labels, arrows, circles, and rectangles to describe data flow direction. Also forming part of DFDs are varied sub-processes, data storage points, and data inputs and outputs.

A data flow diagram has four basic elements. The elements include external entities, data stores, processes, and data flows. The elements are best represented by the two main methods of notation used in DFDs – Yourdon & Coad, and Gane & Sarson. DFD symbols vary slightly depending on methodology. Even so, the basic ideas remain the same.



Data Flow Diagram Notations

External Entity - An outside process or system that sends or receives data to and from the diagrammed system. They are also known as sources, terminators, sinks, or actors and are represented by squares.

Process - This procedure manipulates the data by changing or processing incoming data to an output. Processes (that mainly entail input-processing-output) are portrayed by rectangles with rounded corners, which contain 3 descriptive elements:

Firstly an identification number appears in the upper left-hand corner. This is allocated arbitrarily at the top level and serves as a unique reference.

Secondly, a location appears to the right of the identifier and describes where in the system the process takes place. This may, for example, be a department or a piece of hardware. Finally, a descriptive title is placed in the centre of the box. This should be a simple imperative sentence with a specific verb, such as 'maintain customer records' or 'find the driver.'

Data Store - A data store is a holding place for information within the system. It is represented by an open-ended narrow rectangle. Data stores may be long-term files such as sales ledgers or shortterm accumulations: for example, batches of documents that are waiting to be processed.

Each data store should be given a reference, followed by an arbitrary number.

Data Flow - A data flow shows the flow of information from its source to its destination. A data flow is represented by a line with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal, or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

Data Flow Diagram Levels

Data flow diagrams are also categorised by level. Starting with the most basic, level 0, DFDs get increasingly complex as the level increases. As you build your own data flow diagram, you will need to decide which level your diagram will be.

Level 0 DFD

Also known as context diagrams are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

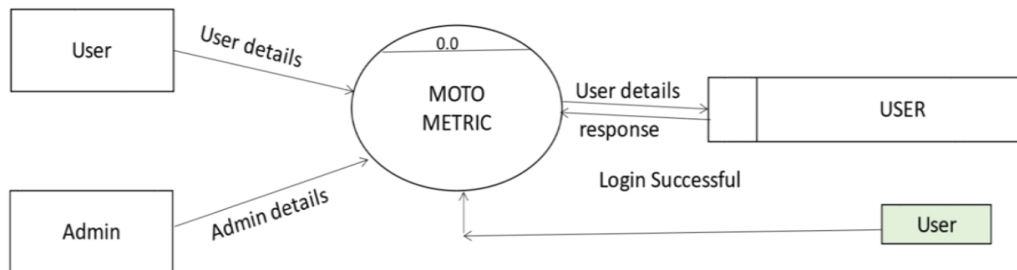
Level 1 DFD

Are still a general overview, but they go into more detail than a context diagram. In level 1 DFD, the single process node from the context diagram is broken down into sub-processes. As these processes are added, the diagram will need additional data flows and data stores to link them together.

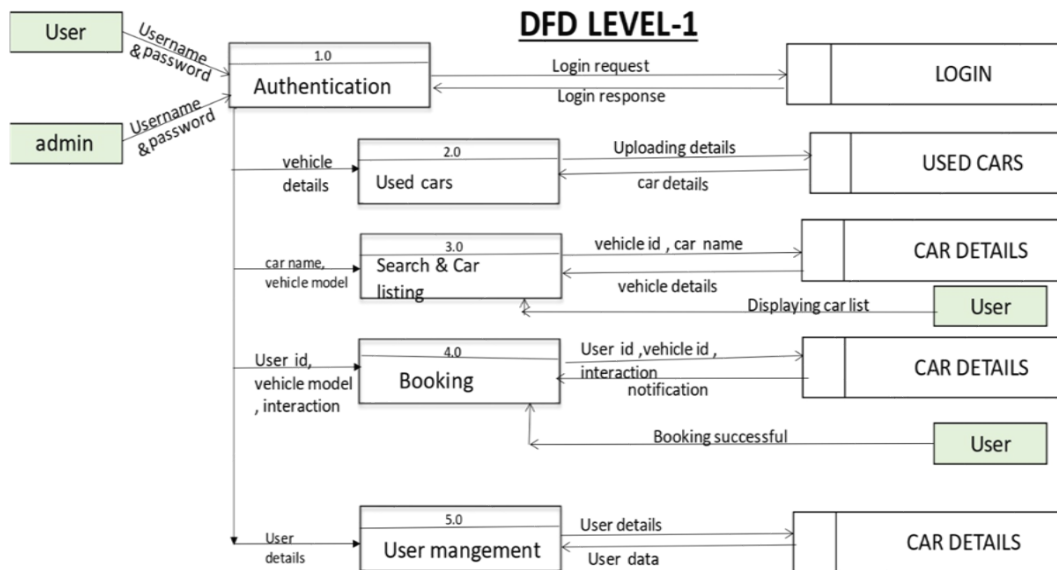
Level 2 DFD

Simply breaks processes down into more detailed sub-processes. In theory, DFDs could go beyond level 3, but they rarely do. Level 3 data flow diagrams are detailed enough that it doesn't usually make sense to break them down further.

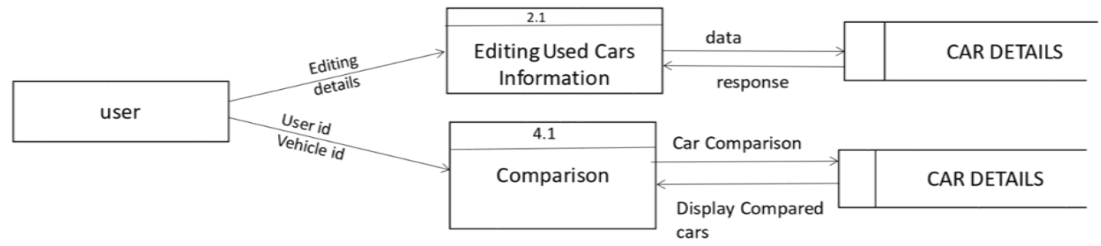
DFD LEVEL 0



DFD LEVEL-1



DFD LEVEL-2



CHAPTER 5

SOFTWARE DEVELOPMENT LIFE CYCLE

The software development life cycle (SDLC), also referred to as the application development life cycle, is a term used in systems engineering, information systems, and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software, or a combination of both.



5.1 SDLC Models

The software development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals.

Many development life cycle models have been developed to achieve different required

objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of the model has a very high impact on the testing that is carried out.

It will define the what, where, and when of our planned testing, influence regression testing, and largely determine which test techniques to use. There are various Software development models or methodologies.

They are as follows:

1. Waterfall model
2. V model
3. Incremental model
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model
8. Prototype model

Choosing the right model for developing the software product or application is very important.

Based on the model the development and testing processes are carried out.

5.2 Selection of the Moto Metric model

The selection of the Moto metric model is a meticulous process driven by the project's core objectives and technical considerations. To ensure a seamless development and deployment of the online platform for buying, selling, and comparing cars and used cars, the chosen model must align with user requirements, offer scalability, and demonstrate cost-effectiveness.

Key factors in the selection process include assessing the technical feasibility, customization capabilities, ease of integration, and the model's ability to provide a positive user experience. Cost implications, vendor reputation, and security features are also critical considerations. The selected model should not only meet the project's immediate needs but also be scalable to accommodate future growth and enhancements.

By prioritizing user-centric design, technical feasibility, and security, the Moto metric model is chosen strategically to create an engaging and reliable platform that full fills the diverse needs of both buyers and sellers in the used car market.

5.3 Agile Methodology

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations or parts that do not directly involve long-term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration, and the scope of each iteration are clearly defined in advance.



Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimise the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software

development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Why should the Agile model has used

The tenets of Agile—adaptability, iteration, continuous delivery, and short time frames, among others—make it a project management style that’s better suited for ongoing projects and projects where certain details aren’t known from the outset. That means if a project doesn’t have clear constraints, timelines, or available resources, it’s a good candidate for an Agile approach.

For example, designing and launching a new product might push a team against several unforeseen challenges. Having an Agile approach can mean the project already has the methodology to test products as often as needed, iterate quickly, and communicate changes with stakeholders.

Traditional project management approaches like Waterfall can be easier to plan out and progress easier to measure. This can make projects with clearly demarcated constraints (like a strict budget or timeline), or projects where teams are expected to work independently of stakeholders better suited for traditional approaches.

The agile model contains four fundamental values, including

- **Individuals over tools:** Agile methodology values working with individual team members in collaboration efforts rather than relying on tools to accomplish a project. For example, a manager using this strategy may encourage a full team meeting rather than running tests with a software tool in order to solve a process problem.
- **Working software over documentation:** While documentation is important in any project for accuracy and accountability purposes, the agile methodology strategy generally prefers real-time reaction to problems rather than continuous documentation. For example, when reacting to a problem in a system, an agile project manager may work to fix the issue before documenting every detail concerning the incident.
- **Customer collaboration over contract negotiation:** In agile project management, a group working on a project may prefer to collaborate with other departments or

companies rather than negotiate the terms of their contract or promised materials. This is so that they adhere to their set goals more reliably, rather than changing goals to meet projected results.

- **Change response over plan rigidity:** While project managers who use agile methodology prefer to not change contract terms, they generally prefer change responses instead of adhering to a plan fully. For example, if the team recognizes that the plan currently in place is not feasible, they change aspects of their process rather than the current goal to help meet needs.

5.3.1 Agile model advantages:

1. In Agile methodology the delivery of software is unremitting.
2. The customers are satisfied because after every Sprint working feature of the software is delivered to them.
3. Customers can have a look at the working feature which fulfilled their expectations.
4. If the customers have any feedback or any change in the feature then it can be accommodated in the current release of the product.
5. In Agile methodology daily interactions are required between the business people and the developers.
6. In this methodology attention is paid to the good design of the product.
7. Changes in the requirements are accepted even in the later stages of development.
8. An Agile/Scrum approach can improve organisational synergy by breaking down organisational barriers and developing a spirit of trust and partnership around organisational goals.

5.3.2 Agile model disadvantages:

- In Agile methodology, the documentation is less.
- Sometimes in Agile methodology the requirement is not very clear hence it's difficult to predict the expected result.
- In a few of the projects at the start of the software development life cycle it's difficult to estimate the actual effort required.
- Because of the ever-evolving features, there is always a risk of the ever-lasting project.
- For complex projects, the resource requirement and effort are difficult to estimate.

- **Industries that use Agile methods**

Agile grew from the minds of a group of software development project managers. Since then, it has continued to be popular in software development but has expanded to many other industries as well. These include finance, IT, business, fashion, biotechnology, and even construction— among many others.

When to use the Agile model

- When new changes need to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the waterfall model in the agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are everchanging in a dynamic business and IT world. Changes can be discussed and features can be newly affected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

The agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time-critical applications. Extreme Programming (XP) is currently one of the most well-known agile development life cycle models.

5.3.3 Application of Agile model

Various software life cycle models in the SDLC approach are available and used by all software development companies. Each software development application has different requirements based on internal and external factors. The agile model is the standard and upgraded SDLC model. There have many requirements and useful situations where this agile model is highly suitable and handle the problem very smartly:

- An Agile model is useful when a system has rapidly new changes that need to be implemented. The freedom agile gives to change is very important. New changes can be implemented at a very less cost because of the frequency of new increments that are produced.
- When implementing a new feature, the development team required losing only the work of a few days, or even only hours, to roll back and implement it.
- When both system developers and stakeholders alike interact with the underdevelopment project.
- Applicable when the developer required more freedom of time and options than if the software was developed in a more rigid sequential way.
- Provide the ability to the authorised development team member to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

CHAPTER 6

SYSTEM SECURITY

6.1 Security in Software

1. SSL/TLS Encryption:

- Secure Socket Layer (SSL) or Transport Layer Security (TLS) encryption is essential for securing the communication between a user's browser and the web server. This encryption protects sensitive data, such as login credentials and personal information, during transmission.

2. Secure Password Practices:

- Enforcing strong password policies, including the use of complex passwords and regular updates, is crucial. Multi-factor authentication (MFA) adds an extra layer of security by requiring users to verify their identity through multiple means.

3. Regular Software Updates:

- Keeping all software components, including the web server, content management system (CMS), and plugins, up to date is essential. Regular updates include security patches that address vulnerabilities, reducing the risk of exploitation.

4. Firewalls:

- Implementing a web application firewall (WAF) helps monitor and filter incoming and outgoing traffic. WAFs are effective in preventing common web application attacks, such as SQL injection and cross-site scripting (XSS).

5. Data Backups:

- Regularly backing up website data is a crucial preventive measure. In the event of data loss or a security breach, backups provide a means to restore the website to a previous state.

6. User Authentication and Authorization:

- Implementing secure user authentication processes ensures that only authorized users can access sensitive information. Regularly reviewing and updating user permissions helps maintain a secure environment.

7. Security Headers:

- Utilizing security headers, such as Content Security Policy (CSP), HTTP Strict Transport Security (HSTS), and X-Frame-Options, enhances the overall security posture of the website and protects against specific types of attacks.

8. File Upload Security:

- If the website allows file uploads, implementing security measures to validate file types, limit file sizes, and store uploads in secure locations helps prevent malicious activities.

9. Monitoring and Logging:

- Implementing robust logging mechanisms helps track and monitor user activities, error logs, and security events. Regularly reviewing logs is essential for identifying and responding to potential security incidents.

10. Intrusion Detection and Prevention Systems (IDPS):

- Employing IDPS aids in detecting and preventing unauthorized access, attacks, or abnormal behaviour on the website. This real-time monitoring helps identify and respond to security threats promptly.

11. Incident Response Plan:

- Developing and maintaining an incident response plan outlines the steps to be taken in the event of a security incident. Having a well-defined plan ensures a timely and organized response to mitigate potential damage.

12. Security Education and Training:

- Training website administrators and users on security best practices is crucial. Increasing awareness and knowledge play a significant role in preventing security vulnerabilities and breaches.

By implementing these security measures, website owners can create a robust and resilient online environment, instilling confidence in users and protecting both the website and its data from potential threats. Regular security assessments and proactive measures contribute to an ongoing commitment to website security.

6.2 Protection and Security Methods

The different methods that may provide protect and security for different computer systems are,

- **Authentication**

This deals with identifying each user in the system and making sure they are who they claim to be. The operating system makes sure that all the users are authenticated before they access the system. The different ways to make sure that the users are authentic are:

1. Username/ Password

Each user has a distinct username and password combination and they need to enter it correctly before they can access the system.

2. User Key/ User Card

The users need to punch a card into the card slot or use their individual key on a keypad to access the system.

3. User Attribute Identification

Different user attribute identifications that can be used are fingerprint, eye retina etc. These are unique for each user and are compared with the existing samples in the database. The user can only access the system if there is a match.

- **One Time Password**

These passwords provide a lot of security for authentication purposes. A one time password can be generated exclusively for a login every time a user wants to enter the system. It cannot be used more than once. The various ways a one time password can be implemented are –

1. Random Numbers

The system can ask for numbers that correspond to alphabets that are pre-arranged. This combination can be changed each time a login is required.

2. Secret Key

A hardware device can create a secret key related to the user id for login. This key can change each time.

- **Authorization**

Authorization means to ensure whether you have permission to access on the network or not. It's simply a verification of permission whether the user has access or not. Some

authorization methods are ACLs (Access Control Lists), Secure objects and methods, Access control for URLs, etc.

- **Biometric System**

A Biometric system is one of the most secure systems as it provides high security to the computer network. This system verifies the user's identity based on some important characteristics that are physiological and behavioural features. Physiological features include face, eyes, fingerprints, and hands. Behavioural features include voice, signature, etc.

- **Firewall**

A firewall is a method of network security that prevents the computer network from users that are not authorised to have access to a network. Firewalls can either be hardware or software or both. It acts as a barrier between unauthorised Internet users and private computer networks connected to the Internet. It blocks the message, viruses, hackers if they do not have authorised access and do not meet the security criteria as per requirement. Any message entering or leaving private computer networks connected to the Internet, especially Intranet, passes through the firewall.

Firewall then checks each message and blocks if found unauthorised. There are several types of firewall techniques:

- Packet Filter
- Application-level gateway
- Circuit-level gateway
- Stateful inspection firewall
- Next-Generation Firewall (NGFW)
 - Proxy server

CHAPTER 7

TOOLS AND TECHNOLOGIES

7.1 PHP



PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. Created by Rasmus Lerdorf in 1994, PHP has evolved into one of the most popular programming languages for building dynamic websites and web applications. Here are key aspects and features of PHP:

1. Server-Side Scripting:

- PHP is primarily designed for server-side scripting, where the script is executed on the server before sending the result to the client's web browser. This enables dynamic content generation and interaction with databases.

2. Open Source:

- PHP is open-source, meaning it is freely available for use, distribution, and modification. The PHP community actively contributes to its development, resulting in continuous improvements and updates.

3. Cross-Platform Compatibility:

- PHP is cross-platform and can run on various operating systems like Windows, Linux, macOS, and others. This makes it a versatile choice for developers working on different platforms.

4. Integration Capabilities:

- PHP can easily integrate with various databases, including MySQL, PostgreSQL, MongoDB, and others. Its compatibility with multiple databases makes it a preferred choice for database-driven web applications.

5. Simplicity and Flexibility:

- PHP is known for its simplicity and ease of learning, especially for beginners. Its syntax is similar to C and Java, making it accessible to a broad audience. It also offers flexibility, allowing developers to choose procedural or object-oriented programming paradigms.

6. Wide Community Support:

- PHP has a vast and active community of developers who contribute to forums, provide support, and share resources. This community-driven aspect ensures that developers can find solutions to common problems and stay updated on best practices.

7. Extensive Library Support:

- PHP has a rich set of libraries and frameworks, such as Laravel, Symfony, and CodeIgniter, that facilitate rapid development and enhance code reusability. These frameworks offer features like MVC architecture, routing, and ORM support.

8. Web Development Capabilities:

- PHP is primarily used for web development to create dynamic web pages and server-side scripting. It supports various protocols, including HTTP, POP3, IMAP, and FTP, enabling developers to create a wide range of web applications.

9. Security Features:

- PHP incorporates security features to protect against common vulnerabilities, but developers must follow best practices to ensure secure coding. Features like the PHP Filter extension and the ability to integrate with SSL/TLS for encrypted connections contribute to creating secure web applications.

In summary, PHP is a versatile, open-source scripting language with a vast community and a history deeply rooted in web development. Its simplicity, integration capabilities, and extensive library support make it a reliable choice for building dynamic and interactive web applications.

7.2 CURSOR



In databases, a cursor is a database object used to traverse through the result set of a query. Cursors are particularly useful when dealing with SQL operations that involve fetching, updating, or deleting data one row at a time. Cursors are commonly employed within stored procedures, triggers, or other database programming contexts. Here are key concepts related to cursors:

1. Declaration:

- Cursors are declared to define a dataset to be processed. The declaration includes the SELECT statement that defines the result set structure.

2. Opening and Fetching:

- After declaring a cursor, it must be opened before fetching data. The 'OPEN' statement initiates the cursor, and the 'FETCH' statement retrieves a row from the result set.

3. Looping through Rows:

- Cursors are typically used within loops to iterate through the result set one row at a time.

4. Closing and Deallocating:

- Once all rows have been processed, the cursor should be closed and deallocated to release associated resources.

5. Types of Cursors:

- SQL supports different types of cursors with varying characteristics. Common types include:
 - **STATIC:** A static cursor does not reflect changes made to the result set by other users.

- **DYNAMIC:** A dynamic cursor reflects changes made by other users while the cursor is open.
- **FAST_FORWARD:** A fast-forward-only cursor allows forward-only traversal and is optimized for fetching rows in a single direction.

6. Performance Considerations:

- Cursors can be resource-intensive, especially when dealing with large result sets. In some cases, set-based operations (manipulating data in bulk) can provide better performance than cursor-based row-by-row processing.

7. Avoiding Cursors:

- In SQL, set-based operations are generally preferred over cursors for better performance. It's advisable to use cursors judiciously and consider alternatives when dealing with large datasets.

Understanding the specific use case and requirements is crucial when deciding whether to use a cursor or opt for a set-based solution for database operations. Cursors provide a valuable tool for certain scenarios, but they should be used carefully to optimize database performance.

7.3 JavaScript



JavaScript is a versatile and widely used programming language that is primarily known for its role in web development. Originally developed

by Netscape, JavaScript has evolved over the years and become a crucial technology for creating dynamic and interactive content on the web. Here are key aspects of JavaScript:

1. **Client-Side Scripting:**

- JavaScript is mainly used as a client-side scripting language. It runs in the user's web browser, allowing developers to create dynamic and interactive web pages. It can manipulate the Document Object Model

(DOM), enabling changes to the structure and content of a webpage in realtime.

2. **Cross-Platform:**

- JavaScript is supported by all major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and others. This cross browser compatibility ensures that JavaScript-powered features work consistently across different platforms.

3. **Object-Oriented:**

- JavaScript is an object-oriented language, allowing developers to create and use objects, which can encapsulate properties and methods. This object-oriented nature facilitates modular and reusable code.

4. **Asynchronous Programming:**

- JavaScript supports asynchronous programming through mechanisms such as callbacks, Promises, and async/await. This enables the execution of nonblocking code, crucial for handling tasks like fetching data from servers without freezing the user interface.

5. **Event-Driven Programming:**

- JavaScript is heavily based on event-driven programming. Actions by the user (like clicks, keypresses, etc.) or changes in the browser environment trigger events, and JavaScript code can respond to these events with specified actions.

6. **Libraries and Frameworks:**

-Numerous libraries and frameworks have been developed to simplify and streamline JavaScript development. Popular libraries include jQuery for DOM manipulation, while frameworks like React, Angular, and Vue.js are widely used for building complex, single-page applications (SPAs).

7. Node.js:

- Node.js extends JavaScript beyond the browser, allowing developers to use it for server-side programming. This server-side JavaScript execution environment enables the creation of scalable and efficient server applications.

8. ECMAScript Standards:

- JavaScript adheres to ECMAScript standards. Regular updates to these standards introduce new features and improvements. ES6 (ECMAScript 2015) brought significant enhancements, such as arrow functions, classes, and Promises.

9. Dynamic Typing:

- JavaScript is dynamically typed, meaning that variable types are determined at runtime. This flexibility allows for more straightforward code but requires careful consideration to avoid unexpected behavior.

10. Security Considerations:

- Security is a critical aspect of JavaScript development. Cross-Site Scripting (XSS) attacks can occur if input validation and security best practices are not followed. Developers need to be aware of security vulnerabilities and employ measures to mitigate risks.

11. Web APIs:

- JavaScript interacts with various Web APIs (Application Programming Interfaces) to enable additional functionality. This includes APIs for handling AJAX requests, manipulating the DOM, accessing geolocation data, and more.

JavaScript is a fundamental technology for modern web development, playing a crucial role in creating engaging user experiences and enabling dynamic, interactive web applications. Its widespread adoption and continuous evolution make it an essential skill for developers working on the front end, back end, or full-stack development.

7.4 HTML



HTML (Hypertext Markup Language) is a standard markup language used to create the structure and present content on the World Wide Web. Developed by Tim Berners-Lee in the early 1990s, HTML serves as the backbone for web development, providing a standardized way to describe the structure and semantics of web documents. Here are key aspects of HTML:

1. **Markup Language:**

- HTML is a markup language that uses a set of predefined tags to structure content on a webpage. Tags are enclosed in angle brackets ("`<`" and "`>`") and are used to define elements like headings, paragraphs, links, images, and more.

2. **Document Structure:**

- HTML documents follow a hierarchical structure known as the Document Object Model (DOM). Documents typically start with an '`<!DOCTYPE html>`' declaration, followed by the '`<html>`', '`<head>`', and '`<body>`' elements.

3. **Elements and Tags:**

- HTML elements are the building blocks of a webpage. Each element consists of an opening tag, content, and a closing tag. Some elements, like line breaks or images, may be self-closing.

4. **Attributes:**

- HTML tags can include attributes that provide additional information about an element. Attributes are added to the opening tag and typically have name/value pairs.

5. **Semantic Elements:**

- HTML5 introduced semantic elements that convey meaning about the structure of the content. Examples include ``<header>``, ``<nav>``, ``<section>``, ``<article>``, ``<footer>``, and more.

6. **Lists:**

- HTML supports ordered and unordered lists using ````, ````, and ```` tags.

7. **Forms:**

- HTML provides ``<form>`` elements to create interactive forms. Form elements include input fields, buttons, checkboxes, radio buttons, and more.

8. **Multimedia:**

- HTML supports the embedding of multimedia content, including images, audio, and video.

9. **Links:**

- Hyperlinks are created using the ``<a>`` tag, allowing users to navigate to other web pages or resources.

10. **Validation:**

- HTML documents are validated against a standard to ensure proper structure.

Validators help developers identify and fix errors in their HTML code.

HTML is a fundamental technology in web development, providing the essential structure for web content. It works in conjunction with CSS (Cascading Style Sheets) and JavaScript to create visually appealing, responsive, and interactive web pages. HTML continues to evolve with new standards and features, ensuring its relevance in modern web development.

7.5 MariaDB



MariaDB is an open-source relational database management system (RDBMS) that was forked from MySQL. It is developed by the MariaDB Corporation and the MariaDB Foundation, led by Michael "Monty" Widenius, one of the original developers of MySQL. MariaDB aims to maintain open-source freedom and guarantee backward compatibility with MySQL, while also introducing new features and improvements.

Here are key aspects of MariaDB:

1. **Open Source:**

- MariaDB is distributed under the GNU General Public License (GPL), making it free and open-source software. The open-source nature encourages collaboration and allows developers to view, modify, and distribute the source code.

2. **Compatibility with MySQL:**

- MariaDB retains compatibility with MySQL, which means that existing MySQL databases, applications, and code can generally be seamlessly migrated to MariaDB.

The two databases share a common ancestry, and MariaDB was created to ensure a vibrant and open future for the MySQL ecosystem.

3. **Features and Enhancements:**

- MariaDB introduces additional features and enhancements beyond MySQL.

These include improved performance optimizations, new storage engines (like Aria, TokuDB, and more), and support for additional data types. MariaDB's commitment to innovation allows it to evolve independently while maintaining compatibility.

4. **Storage Engines:**

- MariaDB supports multiple storage engines, each optimized for different use cases. InnoDB, TokuDB, Aria, and others offer varying trade-offs between performance, transaction support, and storage requirements.

5. **High Performance:**

- MariaDB includes performance improvements, making it competitive with other modern relational database systems. It incorporates optimizations for query execution, indexing, and caching, enhancing overall database performance.

6. **Community and Support:**

- MariaDB benefits from an active and growing community of developers and users. The MariaDB Foundation fosters collaboration and ensures that the development process remains transparent and community-driven. Additionally, commercial support options are available for enterprises through the MariaDB Corporation.

7. **Security Features:**

- MariaDB includes security features such as data encryption, authentication plugins, and user roles. Regular security updates are provided to address vulnerabilities and ensure the protection of sensitive data.

8. **Clustering and Replication:**

- MariaDB supports clustering and replication for improved scalability and high availability. Galera Cluster, in particular, provides synchronous multi-master clustering, enabling real-time replication across multiple database nodes.

9. **Dynamic Columns:**

- MariaDB introduces dynamic columns, a feature that allows developers to add columns to a table dynamically. This flexibility is particularly useful in scenarios where the schema may evolve over time.

10. **Cross-Platform Compatibility:**

- MariaDB is designed to run on various operating systems, including Linux, Windows, and macOS. This cross-platform compatibility ensures flexibility in deployment.

11. **Community Contributions:**

- The MariaDB community actively contributes to the development of new features, bug fixes, and enhancements. Users are encouraged to participate in discussions, report issues, and contribute code.

In summary, MariaDB is a powerful and feature-rich relational database system that builds on the foundation of MySQL. Its commitment to open-source principles, compatibility with MySQL, and ongoing innovation make it a popular choice for a wide range of applications, from small-scale projects to enterprise-level solutions.

7.6 CSS



Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications

Why use CSS

1. Solves a big problem

Before CSS, tags like font, colour, background style, element alignments, border, and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and colour information are added on every single page, it will become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

2. Saves a lot of time

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

3. Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

Advantages

1. CSS saves time

You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

2. Pages load faster

If you use CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

3. **Easy maintenance**

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

4. **Superior styles to HTML**

CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

5. **Multiple Device Compatibility**

Style sheets allow content to be optimised for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or printing.

6. **Global web standards**

Now HTML attributes are being deprecated and it is recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

Disadvantages

1. **Come in different levels**

There's CSS, CSS 1 up to CSS3, which has resulted in confusion among developers and web browsers. One type of CSS should be enough. It would be preferable to having to choose which CSS level to use.

2. **Fragmentation**

With CSS, what works with one browser may not always work with another. This is why web developers have to test for compatibility, running the program across multiple browsers before a website is set live. If only people use Mozilla or Chrome, but they don't.

3. **Lack of security**

Because it is an open text-based system, CSS doesn't have built-in security that will protect it from being overridden. Anyone who has read/write

access to a website can change the CSS file, alter the links or disrupt the formatting, whether by accident or design.

7.9 SOFTWARE REQUIREMENT

TOOLS 7.9.1 XAMPP



XAMPP is a free and open-source cross-platform web server solution stack developed by Apache Friends. The name "XAMPP" is an acronym, where "X" stands for cross-platform, "A" for Apache (the web server), "M" for MariaDB or MySQL (the database system), "P" for PHP (the programming language), and "P" for Perl (another programming language).

Here are key aspects of XAMPP:

1. Cross-Platform:

- XAMPP is designed to be cross-platform, meaning it can run on various operating systems such as Windows, Linux, and macOS. This makes it a versatile solution for developers working on different platforms.

2. Apache Web Server:

- The "A" in XAMPP represents Apache, which is one of the most widely used web servers. XAMPP includes Apache as the default web server, allowing users to locally host and test web applications.

3. Database Support:

- XAMPP includes support for databases, and by default, it comes with either MariaDB or MySQL. Both are popular relational database management systems (RDBMS) used to store and manage data for web applications.

4. PHP, Perl, and More:

- XAMPP bundles several programming languages and tools that are commonly used in web development. PHP is included for server-side scripting, and Perl is also provided for additional scripting capabilities.

5. phpMyAdmin:

- XAMPP includes phpMyAdmin, a web-based application for managing MySQL or MariaDB databases. It provides an intuitive graphical interface to perform various database operations.

6. Easy Installation and Configuration:

- XAMPP is known for its simplicity in installation and configuration. Users can quickly set up a local development environment without dealing with complex server configurations.

7. Development Environment:

- XAMPP is primarily used as a local development environment. It allows developers to work on their web projects offline before deploying them to a live server. This is beneficial for testing and debugging applications.

8. **Open Source and Community-Supported:**

- XAMPP is open-source software, and it benefits from an active community. Users can find support through forums, documentation, and community-contributed resources.

9. **Version Compatibility:**

- XAMPP is regularly updated to include the latest versions of Apache, MariaDB or MySQL, PHP, and other components. This ensures compatibility with the latest web development tools and practices.

10. **Versatility:**

- While commonly used for PHP development, XAMPP is versatile and supports other server-side scripting languages. It is suitable for a range of web development projects, including those using content management systems like WordPress.

In summary, XAMPP is a user-friendly and widely adopted solution for setting up a local web development environment. It provides a convenient way for developers to work on their projects offline before deploying them to production servers. Whether you're a beginner or an experienced developer, XAMPP is a valuable tool for local web development and testing.

7.9.2 VS CODE



Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, build code, and publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools,

graphic designers, and many more features to enhance the software development process. Visual Studio for Mac is a .NET integrated development environment on the Mac that can be used to edit, debug, and build code and then publish an app. In addition to a code editor and debugger, Visual Studio for Mac includes compilers, code completion tools, graphic designers, and source control features to ease the software development process. **Some of the key features and benefits of VS Code are**

1. **Intelligent Code Editor:**

VS Code has a powerful code editor with syntax highlighting, autocompletion, and intelligent code analysis. It provides an intuitive interface and features like multiple cursors, searches and replaces, and keyboard shortcuts, which makes code editing a breeze.

2. **Integrated Terminal:**

VS Code includes an integrated terminal for running commands and scripts within the editor. This eliminates the need for switching between the terminal and the editor and enables developers to execute commands in the context of their project.

3. **Debugger:**

VS Code has a built-in debugger that allows developers to debug code directly within the editor. It provides advanced debugging features like breakpoints, watch expressions, and call stacks, which makes it easy to identify and fix issues in the code.

4. **Git Integration:**

VS Code provides a built-in source control manager that integrates with Git, one of the most popular version control systems. It enables developers to track and manage changes in their codebase, collaborate with others, and perform common Git operations like commit, push, and pull.

5. **Extensions and Marketplace:**

VS Code has a vibrant and growing ecosystem of extensions and plugins, which provide additional features and functionality to the editor. The Marketplace contains thousands of extensions, themes, and snippets that can be installed and used in the editor with just a few clicks.

6. **Cross-Platform:**

VS Code is a cross-platform editor that runs on Windows, macOS, and Linux, which makes it an ideal choice for developers who work on multiple operating systems.

CHAPTER 8

SOFTWARE TESTING

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product. Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrast to the actual requirements. In our project 'Sparsham NSS Unit website', testing and validation have been done in order to ensure that the developed web application is defect free and is performing up to its expectations. Validations and checks have been done to all the modules and forms. Testing has been done from the perspective of the administrator, employee and also from the perspective of the customer. System testing involves unit testing, integration testing, acceptance testing. Careful planning and scheduling are required to ensure that modules will be available for integration into the evolving software product when needed at plan has the following step:

- Prepare a test plan.
- Specify conditions for user acceptance testing.
- Prepare test data for program testing.
- Prepare test data for transaction path testing.
- Plan user training.
- Compile/assemble programs.
- Prepare job performance aids.

9.1 UNIT TESTING

Unit testing is where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. The unit test cases writing and execution is done by the developers and not the testers to make sure that the individual units are working as expected.

For example, in our project the unit testing is done in the login form. WE have checked whether the entered values are successfully saved in the database.

9.2 INTEGRATION TESTING

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of the integration testing is to verify functional, performance and reliability requirements.

In our project, We have integrated Admin with Worker and Admin with User and it worked successfully.

9.3 SYSTEM TESTING

System testing is a level of software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements Software testing determines the correctness, completeness, and quality of software being developed. Validation refers to the process of checking that the developed software meets the requirements specified by the user. The activities involved in the testing phase basically evaluate the capability of that system to meet its requirements. The main objective of software testing is to detect errors in the software. Errors occur if some part of the developed system is found to be incorrect, incomplete

or inconsistent. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects). It involves the execution of a software component or system to evaluate one or more properties of interest.

In our project, WE have integrated admin, worker, user in an environment that works successfully.

In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development.
- responds correctly to all kinds of inputs.
- performs its functions within an acceptable time, and is sufficiently usable.
- Can be installed and run in its intended environments.

CHAPTER 9

SYSTEM IMPLEMENTATION

System implementation is the final stage of software development life cycle. For the successful implementation and cooperation of new systems users must be selected, educated and trained. Unless the users are not trained, the system will become complex and it will feel as a burden for them. A product software implementation method is a systematically structured approach to effectively integrate software based service or components into the workflow of an organisational structure or an individual end-user. A product software implementation method is a blueprint to get users and organisations running with a specific software product. The method is a set of rules and views to cope

with the most common issues that occur when implementing a software product: business alignment from the organisational view and acceptance from human view. It is stated that the implementation of (product) software consumes up to 1/3 of the budget of a software purchase. The complexity of implementing product software differs on several issues. Examples are: the number of end users that will use the product software, the effects that the implementation has on changes of tasks and responsibilities for the end user, the culture and the integrity of the organisation where the software is going to be used and the budget available for acquiring product software.

The implementation stage of the system begins by preparing a plan for implementation of the system. According to this plan, activities are to be carried out, discussions are made regarding the equipment to be required and resources and additional facilities required for implementing the system. The most critical stage is based on their requirements and being effective. This method also offers the greatest success a successful system is by giving users confidence that the system will work achieving securities since the old system can take over if the errors are found or inability to handle certain types of transactions while using the new system.

WE created this project for the college purpose, so WE have provided the builded version (apk) of our software to all departments and other office staff. And WE have hosted our website using Git hub.

WE have taken proactive and reactive measures for implementing our project.

The implementation involves the following formalities:

- careful planning
- investigation of the systems and constraints
- design the methods to achieve the changes Implementation procedure:

The implementation stage of the software development is the process of converting a system specification into an executable system. Implementation is the process in which the theoretical design is changed into a working system. The system can be

implemented only after completing the testing. During implementation the software design is realised as a set of programs or programs.

CHAPTER 10

SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which the software products perform useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development lifecycle. The system maintenance is to make it adaptable to the changes in the environment. There may be social, technical and other environmental changes, which affect the systems behaviour. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to come up with these changes. The first maintenance activity occurs because it is unreasonable to assume that testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and connection of one or more errors is called corrective maintenance. The second activity that contributes to a definition of maintenance occurs because of rapid change that is encountered in every aspect of computing. Therefore, adapting maintenance is an activity that modifies software to properly interface with a changing environment is both necessary and commonplace. The third activity that may be applied to the definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modification for existing functions, and general enhancement are received from the user. To satisfy requests in this category, perfective maintenance is performed. The fourth maintenance activity occurs when the software is changed to improve maintainability or reliability or to provide a better basis for future enhancements.

CHAPTER 11

CONCLUSION AND FUTURE SCOPE

In conclusion, Moto metric stands as a beacon of innovation and convenience in the realm of automotive exploration. Through our platform, users have access to a wealth of information, tools, and services tailored to enhance every aspect of their vehicle-related endeavours. From empowering buyers with detailed vehicle insights to simplifying the process of test driving and selling used cars, Moto metric strives to elevate the automotive experience for all. As I continue to evolve and expand our offerings, our commitment to excellence remains unwavering. Join us on this exhilarating journey as I revolutionize the way the world interacts with vehicles. Together, let's drive towards a future where knowledge, efficiency, and passion converge seamlessly. Experience the difference with Moto metric – where the road ahead is always paved with possibility.

Scope:

The future scope for Moto metric is promising and expansive. As technology continues to advance and consumer preferences evolve, there are several avenues through which Moto metric can further enhance its offerings and solidify its position as a leader in the automotive industry:

1. **Enhanced User Experience:** Constant refinement of the platform's user interface and experience will ensure that Moto metric remains intuitive, accessible, and enjoyable for users of all backgrounds and skill levels.
2. **Integration of Emerging Technologies:** Embracing emerging technologies such as artificial intelligence, augmented reality, and machine learning can revolutionize how users interact with the platform, offering personalized recommendations, predictive analytics, and immersive experiences.

3.Expansion of Services: Diversifying the range of services offered, such as vehicle financing, insurance comparisons, and maintenance scheduling, can transform Moto metric into a comprehensive automotive ecosystem that caters to all aspects of vehicle ownership and management.

4.Global Expansion: Scaling the platform to cater to international markets will unlock new opportunities for growth and engagement, allowing Moto metric to serve a diverse range of users with unique needs and preferences worldwide.

5.Sustainability Initiatives: Integrating sustainability-focused features, such as carbon footprint calculations, eco-friendly vehicle recommendations, and information on alternative fuel options, can align Moto metric with the growing demand for environmentally conscious transportation solutions.

By continually evolving, adapting, and innovating, Moto metric can remain at the forefront of the automotive industry, empowering users and shaping the future of mobility in meaningful and impactful ways.

APPENDIX

Front End:

- **HTML** (Hypertext Markup Language): HTML serves as the foundational markup language for structuring the content on web pages. It defines the structure and elements that make up the user interface.
- **CSS** (Cascading Style Sheets): CSS is employed for styling and formatting the HTML elements, enhancing the visual presentation of the web pages. It controls layout, colors, fonts, and overall aesthetics.

- **JavaScript:** JavaScript is a dynamic scripting language that adds interactivity to web pages. It allows for client-side behaviour, enabling features such as form validation, dynamic content updates, and asynchronous communication with the server.

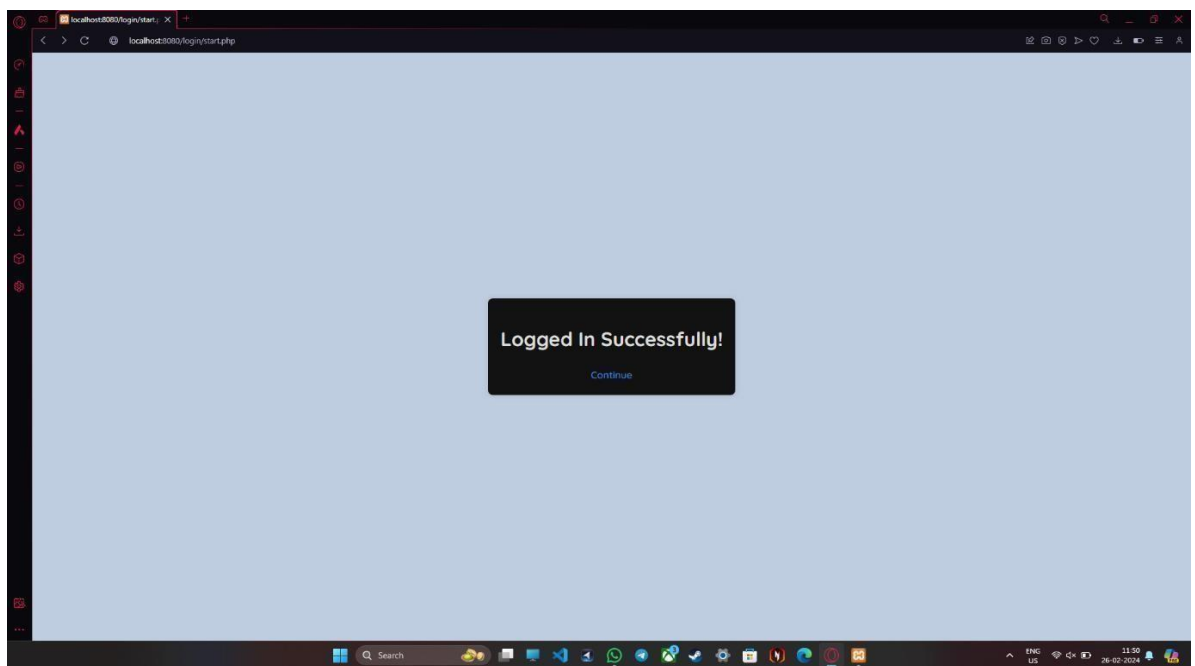
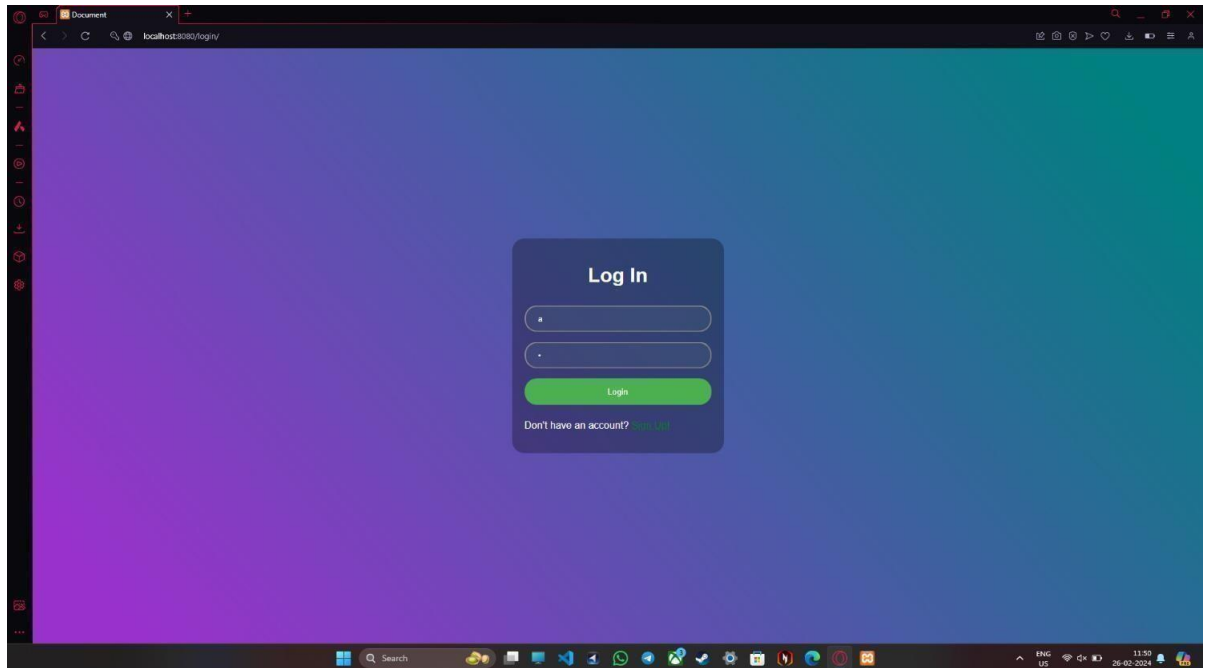
Back End:

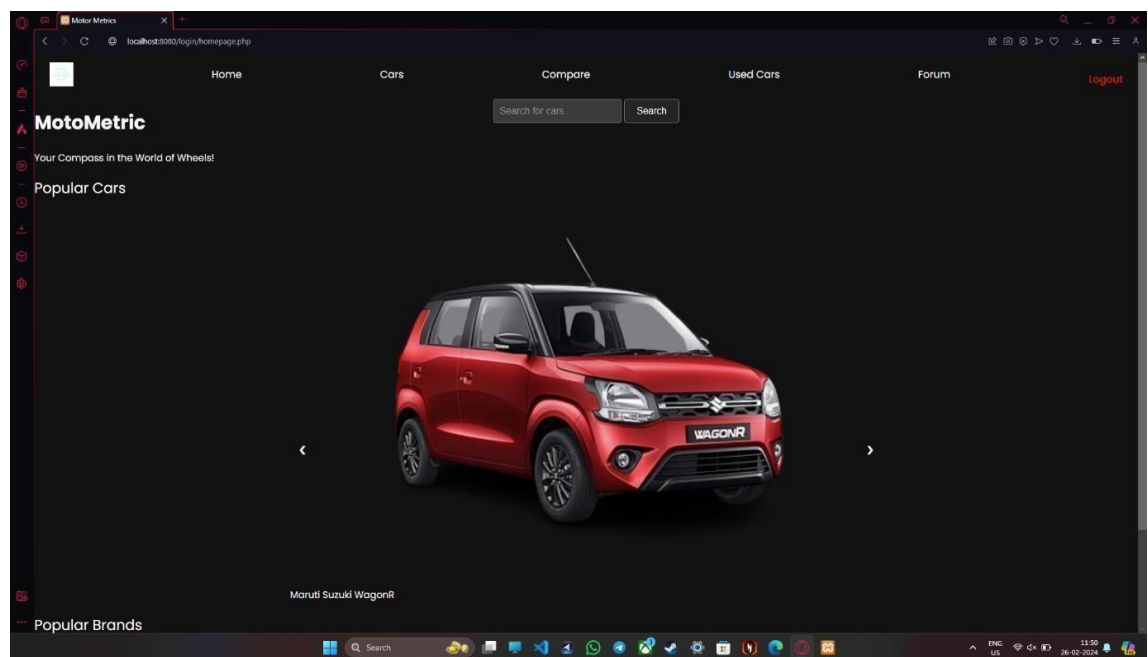
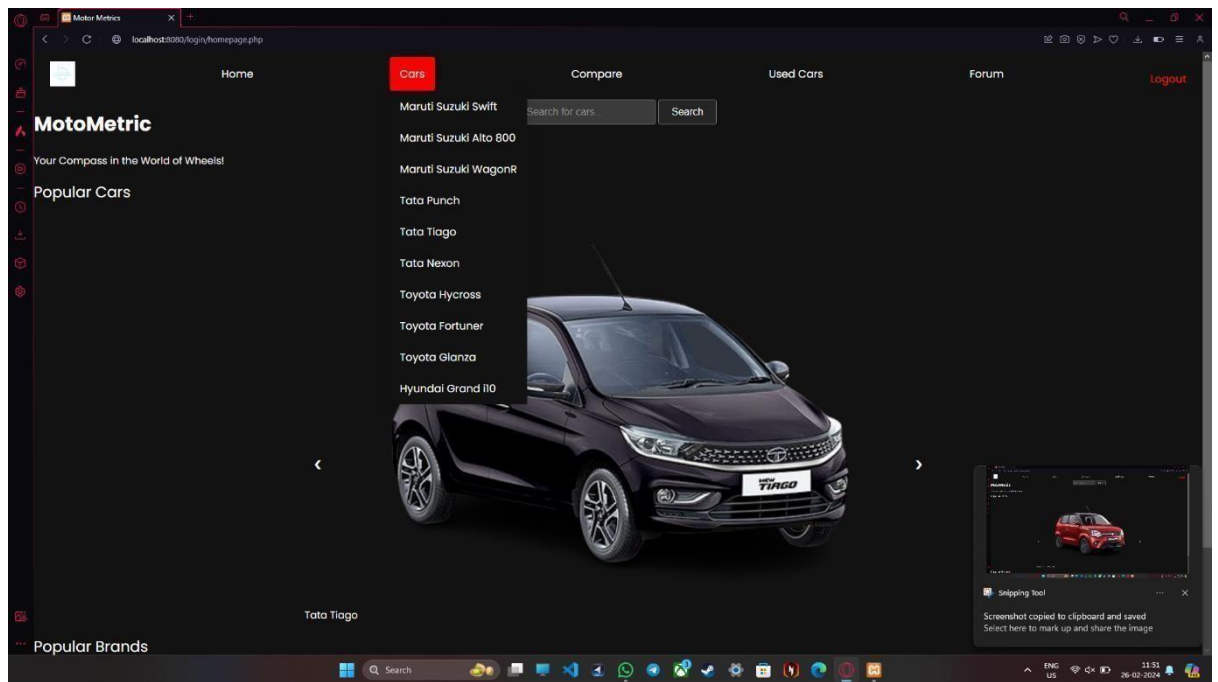
- **PHP:** PHP (Hypertext Preprocessor) is utilized for server-side scripting. It handles the logic and processing on the server, interacting with the database and generating dynamic content to be served to the front end.
- **MariaDB:** MariaDB, a relational database management system, is chosen for the backend database. It stores and manages data efficiently, providing a reliable and scalable solution for data storage and retrieval.
- **Testing:** The web application undergoes rigorous testing at both the front end and back end. Testing frameworks and methodologies are employed to ensure functionality, performance, and security.

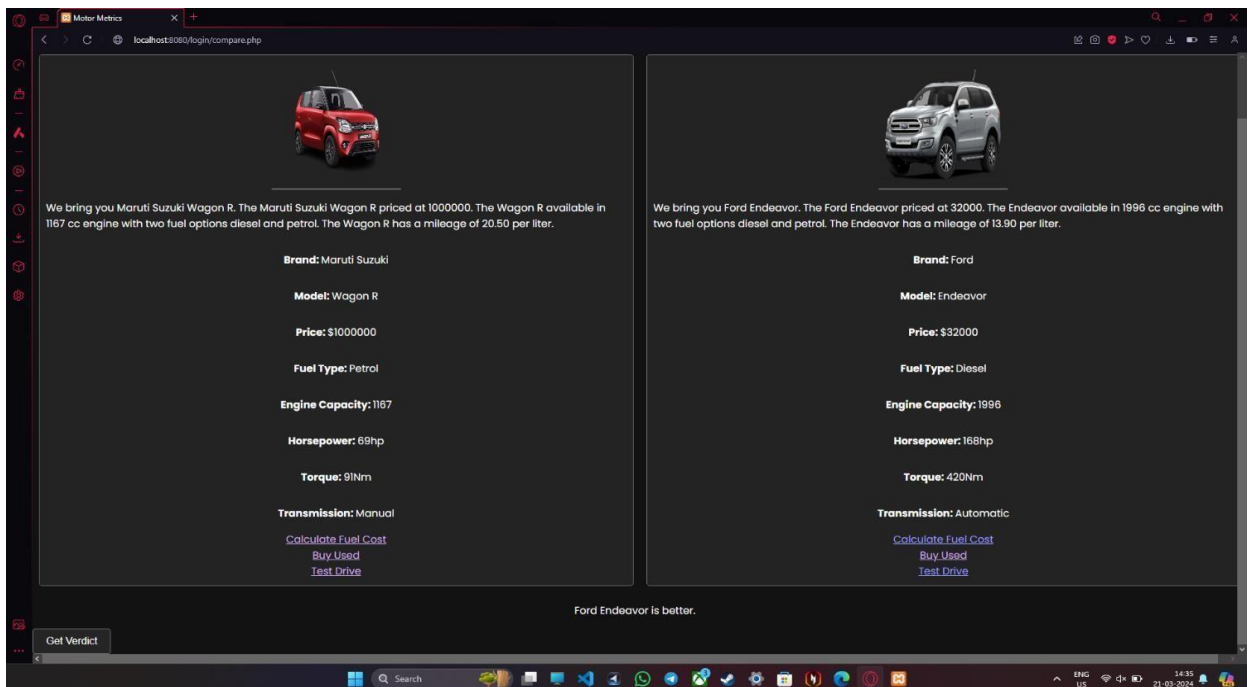
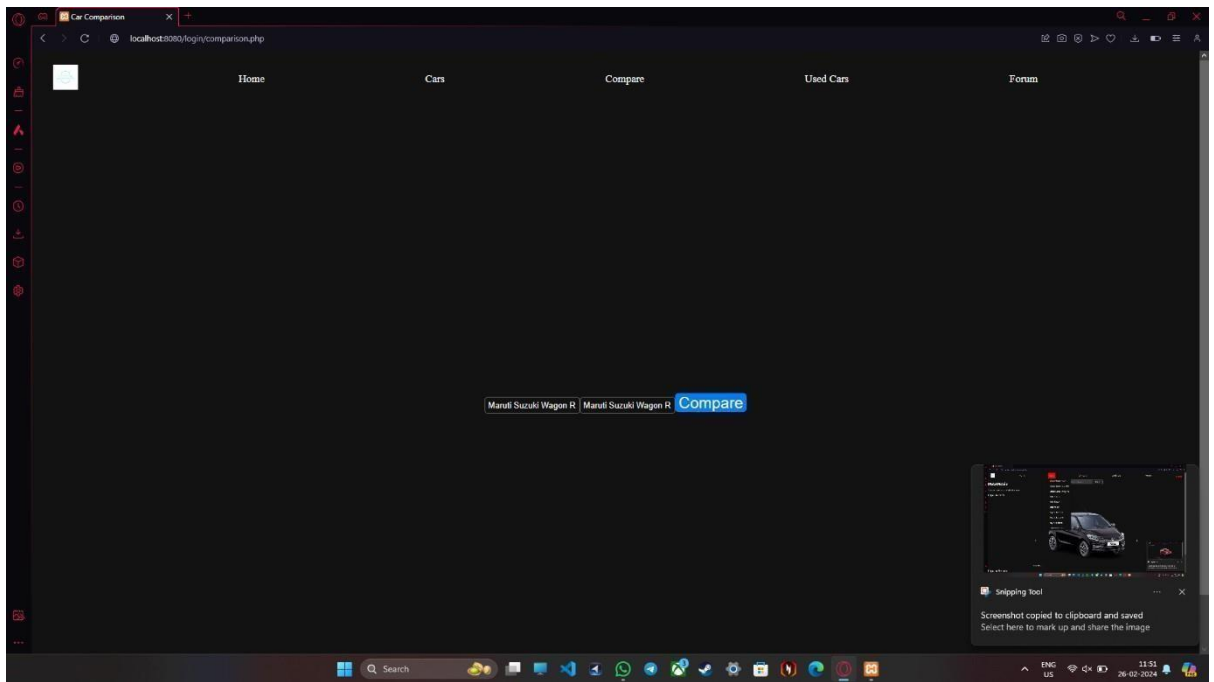
Conclusion:

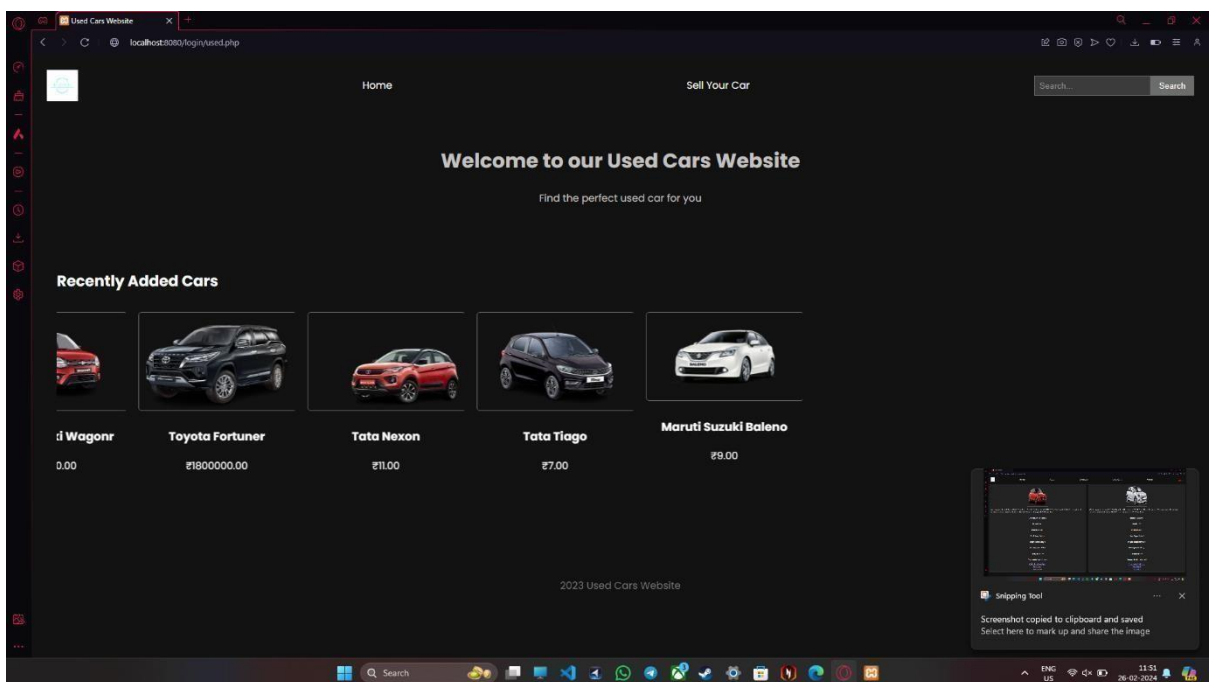
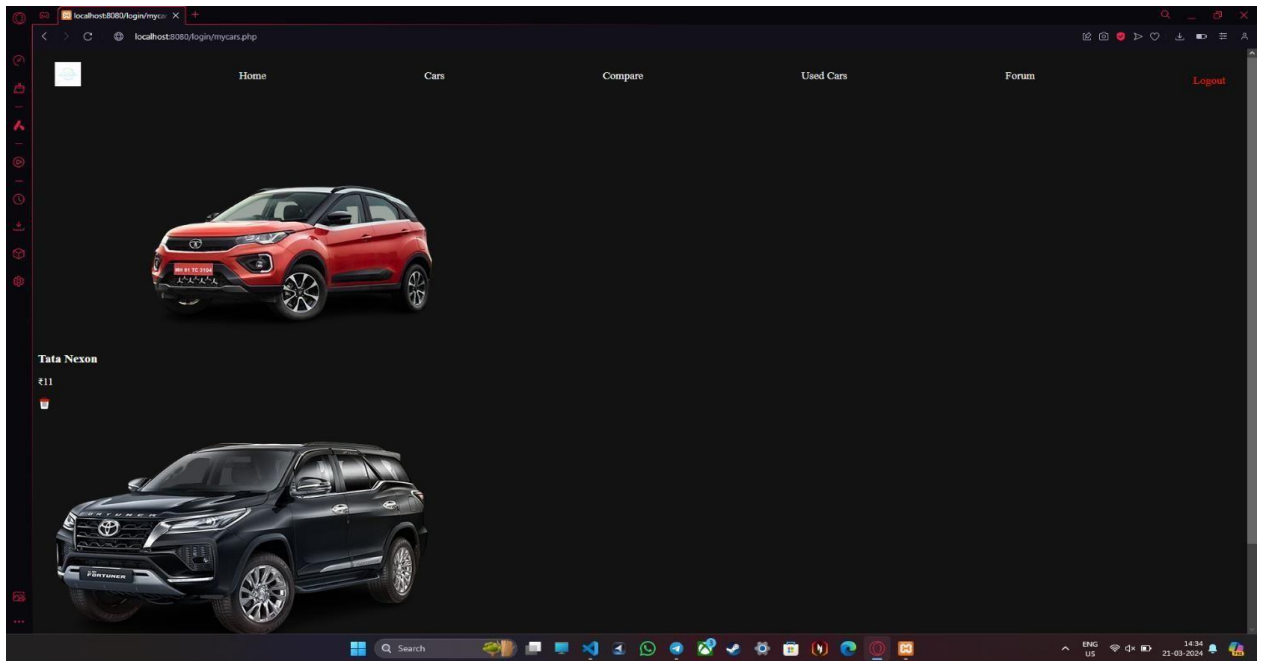
Comprising HTML, CSS, JavaScript, PHP, and MariaDB, creates a well-rounded and efficient environment for web application development. The integration of front-end and back-end technologies results in a seamless user experience, with a responsive and interactive interface backed by a robust server-side infrastructure.

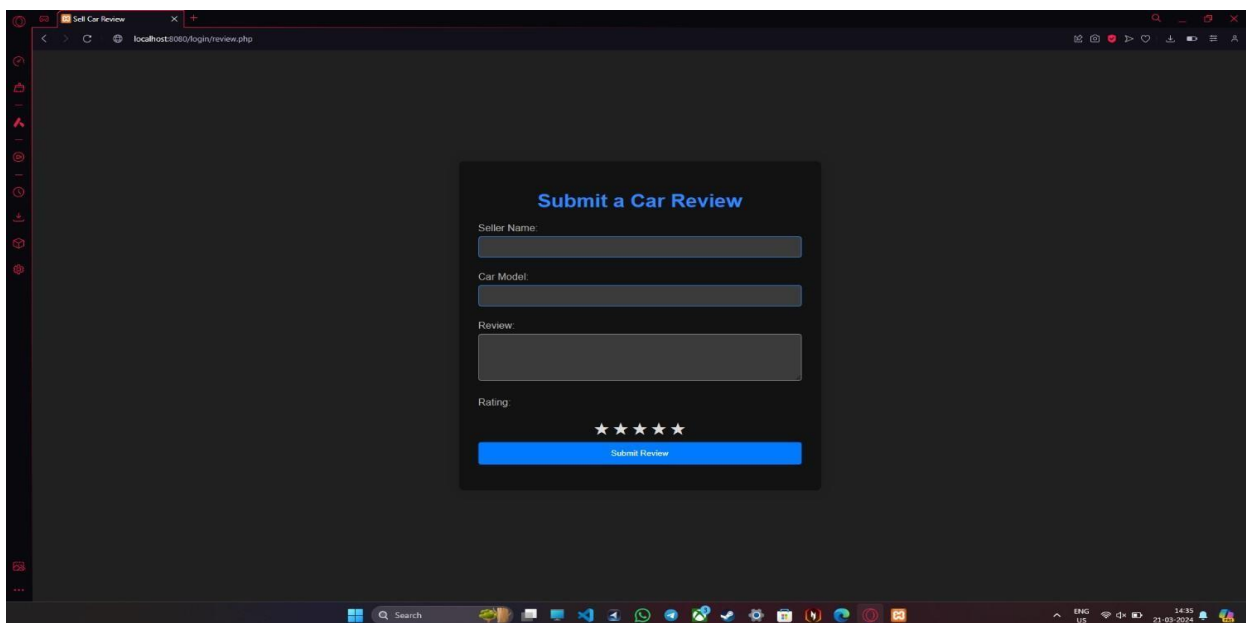
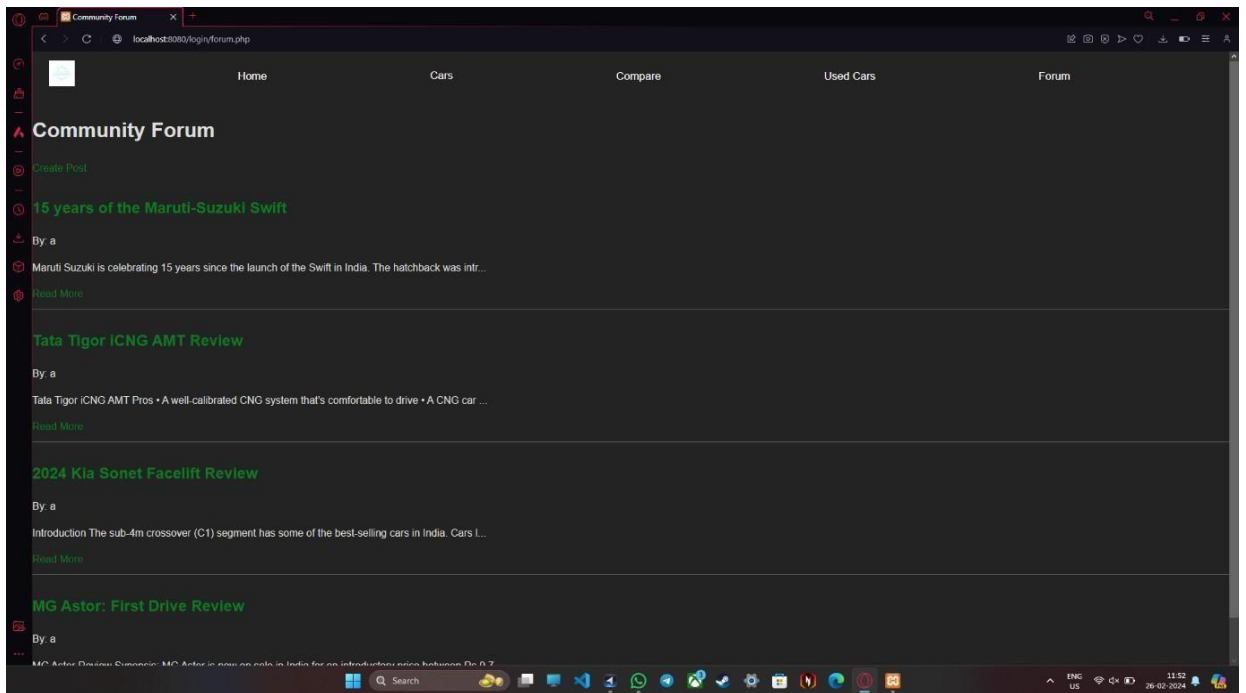
SCREENSHOTS OF THE APPLICATION

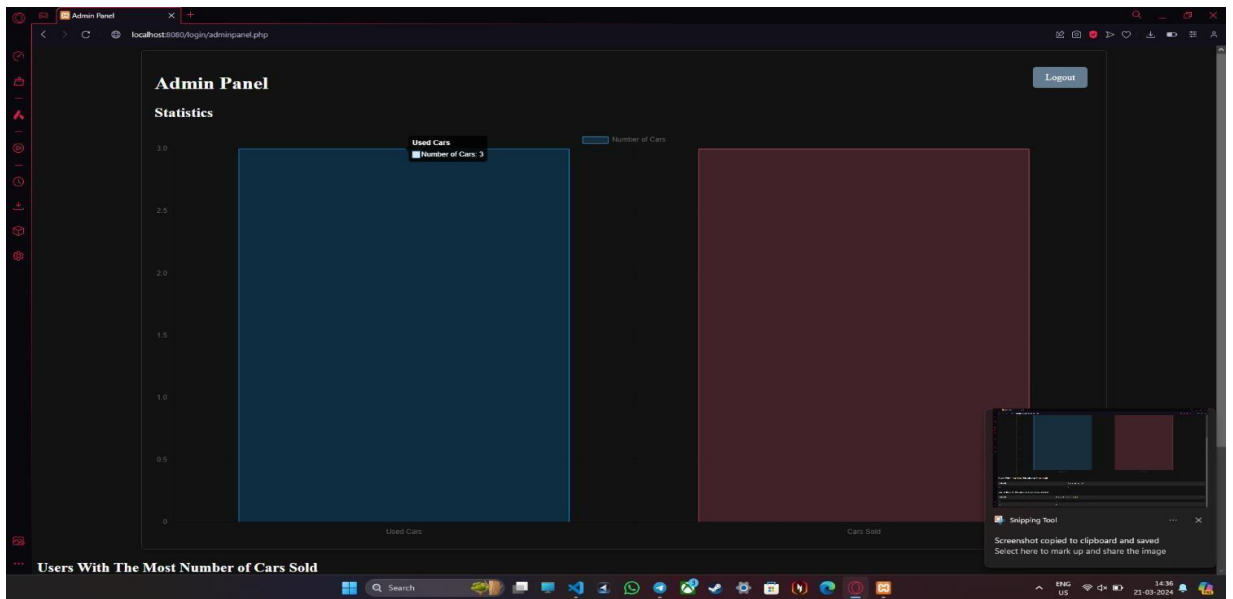












Users With The Most Number of Cars Sold	
Seller ID	Number of Cars Sold
a	3

Top Sellers by Number of Used Cars Posted	
Seller ID	Number of Used Cars Posted
a	2
b	1

SAMPLE CODE

```
<!DOCTYPE html>
<html>
```

```

<head>
  <title>Motor Metrics</title>
  <link rel="stylesheet" href="homepage.css">
</head>
<body> <?php  include 'validation.php'; include
'navbar.php';
?>

<div class="search-container">
  <form action="search_cars.php" method="GET">
    <input type="text" name="search" placeholder="Search for cars...">
    <button type="submit">Search</button>
  </form>
</div>

<div class="main">
  <h1>MotoMetric</h1>
  <p>Your Compass in the World of Wheels! </p>
</div>
<h2>Popular Cars</h2>
<div class="slideshow-container">

  <div class="mySlides fade">
    <a href="WagonR/wagonr.php">
      
    </a>
    <div class="text">MarutWe SuzukWe WagonR</div>
  </div>

  <div class="mySlides fade">
    <a href="tiago/tiago.php">
      
    </a>
    <div class="text">Tata Tiago</div>
  </div>

  <div class="mySlides fade">
    <a href="Swift/swift.php">
      
    </a>
    <div class="text">MarutWe SuzukWe Swift</div>
  </div>

  <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
  <a class="next" onclick="plusSlides(1)">&#10095;</a>

```

```

</div>
<h2> Popular Brands</h2>
<div class="car-brands">
  <div class="scrolling-wrapper">
    
    
    
    
  </div>
</div>

<script>    var slideIndex = 0;    showSlides();    function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  for (i = 0; i < slides.length; i++) {
slides[i].style.display = "none";
  }
  slideIndex++;    if (slideIndex > slides.length) {slideIndex =
1}    slides[slideIndex-1].style.display = "block";
setTimeout(showSlides, 2000); // Change image every 2 seconds
  }

  function plusSlides(n) {
showSlides(slideIndex += n);
  }
</script> <script>    function showCars(brand) {
  // Pass the brand name as a parameter to search_cars.php    window.location.href =
"search_cars.php?search=" + brand;
  }
</script>
</body>

```

BIBLIOGRAPHY

1. <https://www.udemy.com/>
2. <https://www.youtube.com/@BrototypeMalayalam>