

CAR DEALERSHIP

SOFTWARE ENGINEERING PROJECT

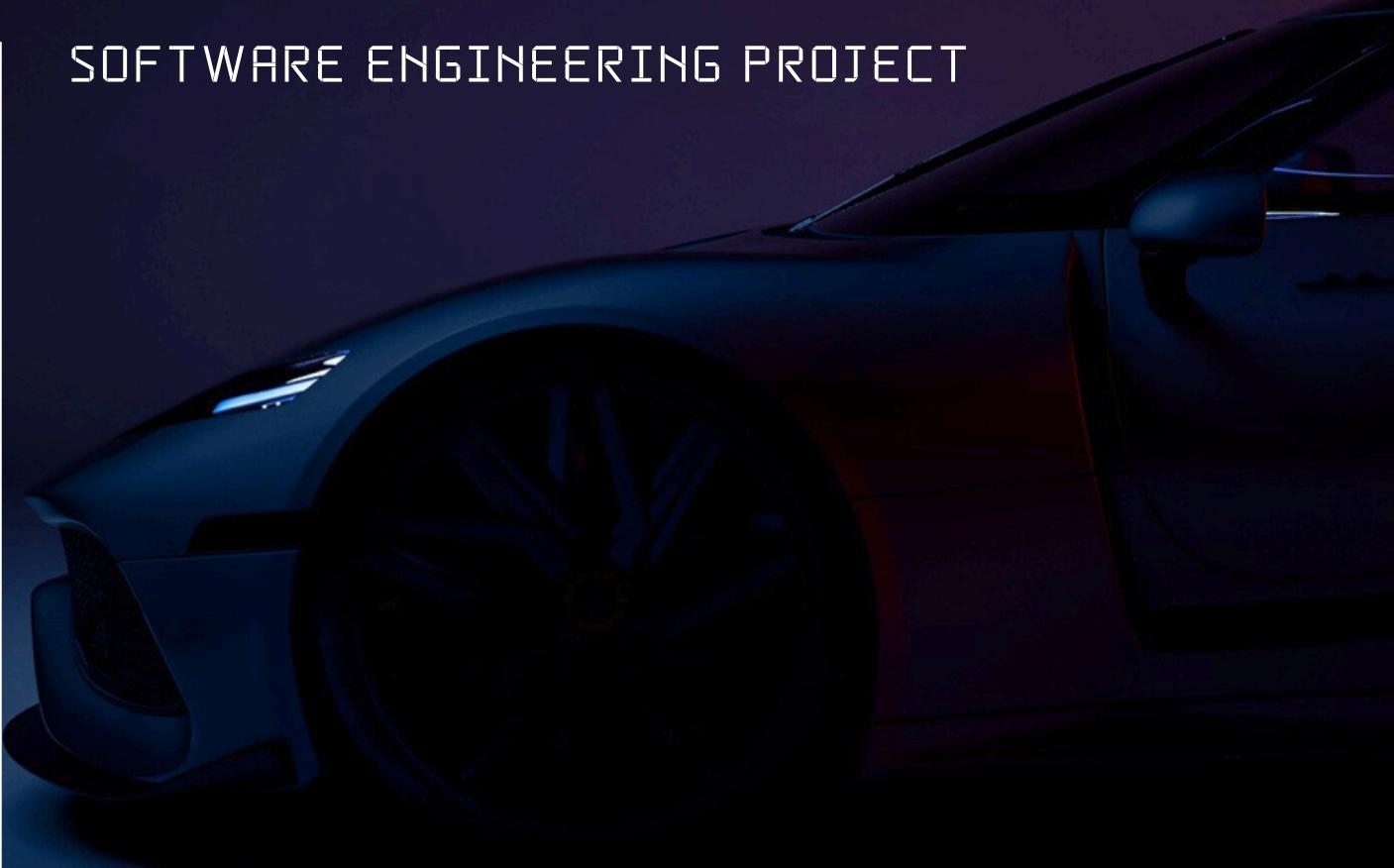




Table of Contents

INTRODUCTION

01.

- Problems identified
- 1.2 Overview of the Website
- 1.3 Project Focus and Objectives:
- 1.4 Purpose.

ANALYSIS

02.

- 2.1 Software requirements specification.
- 2.2 Activities Network.
- 2.3 Time Estimation.
- 2.4 Project scheduling.
- 2.5 Parametric cost estimation.
- 2.6 Risk Analysis.

DESIGN

03.

- 3.1 introduction
- 3.2 use case diagram
- 3.3 Class Diagram

TEST

04.

- 4.1 introduction
- 4.2 The levels of testing
- 4.3 Test Cases

05. CODE STYLE

T 24

Alaa Ebrahem Mohamed Ali	20221379220	Leader - Interface Designer - Developer
Ali mohamed sayed ahmed	20221449583	Backend programmer - Developer
Yasmine Yasser Abdelwadood	2022145650 4	System Analyst - Developer
Aysel Mahmoud hassan	20221465749	Backend programmer - Developer
Rana Alaa Ahmed Ahmed	20221456493	Interface Designer - Developer

0.0 CUSTOMER GRAND ROYAL



Meet Grand Royal, an esteemed automotive agency that has been a trusted name in Sidi Bishr for six years. Specializing in car sales, Grand Royal offers a diverse selection of vehicles and personalized payment plans to cater to every customer's needs. With a reputation for integrity and exceptional service, Grand Royal is committed to providing a seamless car buying experience, making dreams of owning a quality vehicle a reality for its valued clientele.

INTRODUCTION

Problems identified

1.1

Information Technology is widely utilized on almost every business areas at present. Because of the development on Information Technology, people are used to get the advantage on computerized work than manual work because of the efficiency, reliability and accuracy of using computerized systems. People have the tendency to think that machines are more reliable to handle sensitive and non-sensitive data. That is why most of the processing is done by machines.

The motive of our project is to control various activities performed in the office using the computer.

Car Dealership Website

1.2

Our Car Dealership Website serves as a user-friendly platform, simplifying the entire car purchasing process. Users can securely register, browse comprehensive car listings, schedule test drives, select from various payment options, and track their purchases with ease. The website is designed to be mobile-friendly and prioritizes security and compliance with industry standards. Additionally, features like user feedback channels and responsive customer support enhance the overall car buying experience while optimizing dealership operations.

1.3. Project Focus and Objectives:

Our project is centered on innovating the car sales landscape, utilizing technology to make the buying process more accessible and enjoyable for customers. By envisioning a scenario where individuals can seamlessly browse, purchase, and manage their car transactions online from the comfort of their homes, we strive to provide a smooth, efficient, and secure experience. Our goal is to ensure that our system is not only reliable but also adaptable to future developments in the automotive industry.

ANALYSIS

Initially, we recognized the importance of thorough requirement gathering to establish a solid foundation for our project. Through various techniques such as field visits, observation, questionnaires, we diligently gathered comprehensive requirements.

Notably, we engaged in multiple discussions with the owners of the dealership, who provided invaluable insights and feedback. Their input was instrumental in refining and validating our requirements to ensure they align closely with the project's objectives.

Now, why the waterfall model? The waterfall model is a sequential approach, where each phase of the project is completed before moving on to the next. This linear progression appealed to us as it provides a clear and structured framework for project execution. Given the nature of our project and the well-defined requirements gathered through extensive discussions, we believed that the waterfall model would be suitable. It allows us to proceed with development in a systematic manner, ensuring that each stage is completed thoroughly before advancing to the next.

Furthermore, the waterfall model offers clarity and predictability, which are crucial for our project's success. By following a predefined sequence of phases - including requirements gathering, design, implementation, testing, deployment, and maintenance - we can maintain focus and direction throughout the project lifecycle. This approach minimizes the risk of scope creep and ensures that we deliver a high-quality product that meets the specified requirements. In conclusion, while the incremental model offers agility and adaptability,

the waterfall model provides structure and predictability, making it a suitable choice for our Car Dealership Website project.

2.1

Software requirements (Functional)

2.1.1 For User

1. Account Registration:

Users can register conveniently by completing a user-friendly form with required details: full name, valid email, password, phone number, address, and job title. Upon successful submission, an automated confirmation email is sent with activation instructions.

2. Approval Check for Payment Plans:

Users can upload income verification documents securely, such as income proof, through the sign-in feature. Clear instructions are provided regarding document types accepted and approval criteria. Upon submission, users receive a confirmation message and notifications on the approval process status.

3. Search and View Products:

Users can search for products based on various parameters like year, type, and price range. Detailed product information includes specifications, features, pricing breakdown (including interest rates and down payments), and transparent total cost calculations. Product reviews and ratings from other users aid in informed decision-making.

4. Order Placement:

Users can place orders directly from the product detail page with a prominent "Order Now" button. Confirmation prompts ensure intentional submissions. Users can edit or cancel orders within a specified timeframe via their account dashboard.

5. Appointment Scheduling:

Users can schedule appointments for physical product inspections using an integrated calendar feature. They select preferred dates and times with real-time availability updates. Confirmation emails and reminders are sent upon successful scheduling.

6. Customer Support:

Users have access to a comprehensive help center with FAQs, troubleshooting guides, and video tutorials. Inquiries or complaints submitted through the ticketing system are categorized by urgency and assigned to appropriate support agents. Timely updates via email or SMS are provided on inquiry status, including resolution notifications.

7. Feedback:

Users can submit feedback, ideas, or suggestions via a dedicated form accessible from the website's footer or main navigation. Feedback submissions are categorized and tagged for easy analysis, with options for additional comments or attaching screenshots. An acknowledgment system notifies users of their submissions and expresses gratitude for their input.

2.1

Software requirements (Functional)

2.1.2 For System

1. Document Upload and Verification:

- The system should allow users to upload income verification documents. There should be a manual verification process by administrators to ensure submitted documents meet the criteria for desired payment plans.

2. Product Search and Display:

- The system should provide a search feature allowing users to find products based on various criteria. Detailed product information should be displayed, including pricing, specifications, and reviews.

3. Order Management:

- The system should support order processing, editing, cancellation, and tracking. Users should receive notifications at various stages of order processing.

4. Appointment Scheduling System:

- The system should allow users to schedule appointments for physical inspections of products. The scheduling system should prevent double booking and manage appointment calendars effectively.

5. Customer Support Ticketing System:

- The system should provide a ticketing system for users to submit inquiries or complaints. Support agents should be able to view, assign, and resolve tickets efficiently.

6. Feedback and Rating System:

- Users should be able to provide feedback and ratings for products and services.
- The system should aggregate and display ratings and reviews to aid other users in decision-making.

2.1

Software requirements (Functional)

2.1.3 For Business

Enhanced Marketing Integration:

- Connect the website with email marketing platforms and social media channels. Customize marketing campaigns based on how users interact with the website.

Customer Acquisition Strategies:

- Create forms on the website to collect potential client information. Run targeted advertising campaigns to attract specific groups of people.

Content Optimization for SEO:

- Make sure website content is easy for search engines to understand. Regularly update the website with useful information related to the industry.

Conversion Rate Optimization (CRO):

- Test different parts of the website to see what works best for getting people to take action. Look at how people use the website to make it easier for them to do what we want them to do.

Payment Plans:

- Implement a system to calculate payment plans with a 23% interest rate and a 30% down payment. Develop a process for immediate payment plans with a 35% interest rate and a 40% down payment.

Customer Support:

- Implement a customer support system to assist users with inquiries or issues related to their purchases

2.1

Software requirements (non Functional)

For User

Secure Sign-In:

- Ensure the sign-in process is secure, protecting user accounts and data.

For System

Data Safety: Implement strong security measures to protect user information and financial transactions.

2. Reports and Stats: Provide tools and analytics to generate reports on sales, user preferences, and system performance.

3. Updates and Fixes: Plan for regular updates and fixes to maintain the website's functionality and security.

4. Working with Other Systems: Establish integrations with other systems, such as credit checks, to enhance verification processes.

5. Easy to Use: Design the website to be accessible and user-friendly, catering to users with disabilities..

6. Performance Requirement: Response time for certain search operation should not take more than 10 seconds. Responses to queries shall take no longer than 7 seconds to load onto the screen after the user submits the query. The website pages should load in 3 seconds

7. Security Requirement: Users shall be required to log in to the system for all operations. The system shall permit only staff members who are on the list of authorized administrators to create or edit customers and vendors

For Business

Promotions and Discounts: Ensure the system identifies and applies discounts or promotions to eligible users, such as loyal customers or during specified periods.

2.2 NETWORK ACTIVITY

In Critical Path Analysis, this helps the teams to comprehend the specific event sequences driving time requirements for objective achievement.

01 Requirement Gathering Phase:

We'll organize and conduct various requirement gathering activities to get a clear understanding of what the dealership owners and stakeholders need from the website. We'll schedule meetings and interviews with them to discuss their goals and expectations in detail. Field visits might be necessary to observe their current processes firsthand, and we'll prepare questionnaires to gather specific information. Once we've gathered all the requirements, we'll review them as a team to ensure we haven't missed anything important.

02 Design Phase:

- During the design phase, our team will work closely together to translate the gathered requirements into actionable design specifications. We'll collaborate on creating use case diagrams and class diagrams to visualize how the website will function and what components it will include. Design reviews will be a team effort, where we'll provide feedback and suggestions to improve usability and functionality. It's important that everyone on the team has a clear understanding of the design before moving forward.

03 Development Phase:

- In the development phase, each member of our team will play a specific role in bringing the design to life. Some of us will focus on building the user interface, while others will work on implementing backend functionalities. We'll work closely together, sharing code and ideas, and participating in regular code reviews to maintain consistency and quality. If any issues arise during development, we'll troubleshoot them as a team to find the best solutions.

04 Testing Phase:

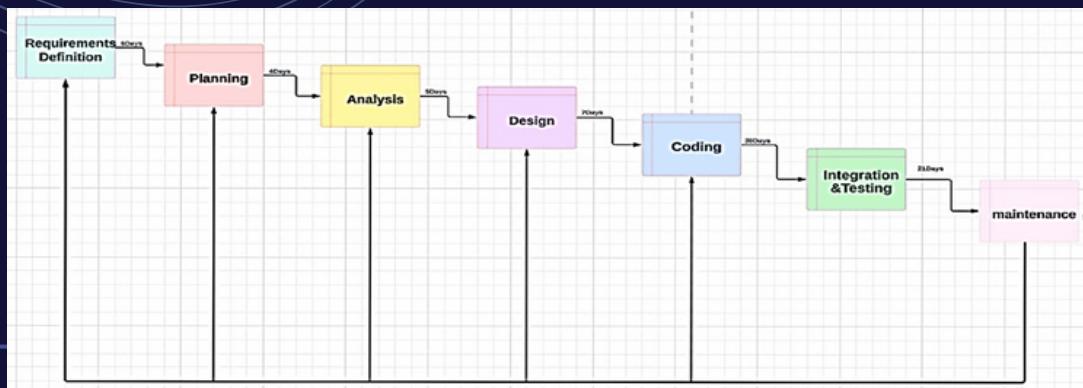
- Testing will be a collaborative effort among all team members. We'll work together to create comprehensive test cases and scenarios to ensure that every aspect of the website functions as intended. Unit testing will be done by individual developers, while system testing and user acceptance testing will involve the whole team. We'll communicate openly about any bugs or issues we encounter and work together to address them promptly.

05 Deployment Phase:

- As we prepare to deploy the website to the production environment, our team will work together to ensure a smooth transition. We'll coordinate closely to schedule the deployment and perform any necessary setup or configuration tasks. If any unexpected issues arise during deployment, we'll collaborate to troubleshoot and resolve them quickly to minimize downtime.

06 Maintenance Phase:

- Once the website is live, our team will continue to work together to provide ongoing support and maintenance. We'll monitor the website closely for any issues or bugs reported by users and prioritize them based on severity. Regular team meetings will allow us to discuss and plan future updates and enhancements, ensuring that the website remains up-to-date and responsive to the dealership's needs.



2.3 Time Estimation:

- Start: February 17, 2024
- Requirements Definition: February 18, 2024 - February 24, 2024
- Planning: February 25, 2024 - February 29, 2024
- Analysis: February 29, 2024 - March 3, 2024
- Design: March 4, 2024 - March 9, 2024
- Coding: March 10, 2024 - April 9, 2024
- Integration & Testing: April 10, 2024 - April 30, 2024
- Maintenance: May 1, 2024 - May 10, 2024

Taking the Maximum number of days and placing the possibility of any disruptions occurring during the work, we will set 10 reserve days to avoid any problems occurring during the project delivery date

2.4

Project scheduling

	February				March				April			
Requirements gathering												
Analysis												
Design												
Coding												
Testing												
	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4

Risk

2.5 Analysis

In a Waterfall approach, risks are typically addressed at each phase before progressing to the next, and changes are generally not accommodated once a phase is completed. Therefore, thorough planning and risk mitigation strategies must be in place from the outset to minimize the impact of identified risks.

In waterfall:

- Long time and cannot test the system until the end of project
- Lack of flexibility
- Does not support user feedback

Risk Analysis		Severity				
		Insignificant	Minor	Moderate	Major	Critical
Likelihood	Certain <90%>	Moderate	High	Very High	Very High	Very High
	Likely <50% - 90%>	Moderate	High	High	Very High	Very High (7)
	Moderate <10% - 50%>	Low	Moderate	High (2)	High (1),(5)	Very High (3)
	Unlikely <3% - 10%>	Low	Low	Moderate	Moderate	High
	Rare <3%>	Low	Low	Low (4)	Low (6)	Moderate

ID	Risk Name	Risks	Solutions
1	Payment Plans	Complex financial calculations could lead to inaccuracies in payment plans, impacting user trust and financial transactions.	<ul style="list-style-type: none"> Thorough Testing and Validation. Establish mechanisms for continuous monitoring of payment plan performance and financial calculations to detect any emerging issues or areas for improvement.
2	Customer Support	Inadequate customer support system could result in unresolved user issues, leading to dissatisfaction and negative feedback.	<ul style="list-style-type: none"> Implement a Comprehensive Support System. Encourage users to provide feedback on their support experience through surveys, ratings, or feedback forms.
3	Secure Sign-In	Vulnerabilities in the sign-in process could lead to unauthorized access and data breaches.	<ul style="list-style-type: none"> Use Secure Sign-In Protocols. Regular Security Audits and Penetration Testing. Monitor and Analyze Sign-In Activities.
4	Clear Prices	Inaccuracies or inconsistencies in pricing display could lead to user distrust and dissatisfaction.	<ul style="list-style-type: none"> Regularly Update Pricing Information. Break down pricing into components to provide users with a detailed understanding of what they are paying for. Include itemized lists of charges, taxes, fees.
5	Updates and Fixes	Delayed or inadequate updates could lead to security vulnerabilities and system instability.	<ul style="list-style-type: none"> Implement Automated Update Mechanisms. Schedule Downtime for Maintenance.
6	Staff	No suitable staff available to work in project continuously	<ul style="list-style-type: none"> Conduct thorough resource planning to identify the required skill sets, expertise, and staff levels needed for the project. Recruitment and Training. Team Morale and Motivation.
7	Requirements	Change in requirements	<ul style="list-style-type: none"> Documenting Requirements. Prioritization and Impact Analysis:

Parametric cost estimation

is a technique that uses a set of parameters or factors, such as the size of the project, the complexity of the project, the number of software developers, and the estimated duration of the project to generate development cost estimates for the project.

In terms of lines of code (LOC), a small to medium-sized car shop software system could range from once of thousands to a few hundred thousand lines of code, depending on factors such as:

2.6 Scope and Complexity

The more features and functionalities included in the software, the larger the codebase is likely to be. Complex features like automated scheduling algorithms or advanced reporting capabilities can contribute to increased size.

Technology Stack:

The choice of programming languages, frameworks, and libraries can impact codebase size. For example, using a high-level language with built-in libraries may result in a smaller codebase compared to a lower-level language requiring more manual implementation.

Architecture and Design:

Well-structured and modular codebases tend to be more manageable and maintainable but may require additional code for abstraction and separation of concerns.

Reuse and Third-Party Components:

Leveraging existing code libraries, frameworks, or third-party components can reduce the overall size of the software system. However, integrating these components may introduce additional dependencies and complexity.

Based on these considerations, an average size for a car shop software system could fall within the range of 1,000 to 3,000 lines of code.

Every line costs 1EGP

Every developer will take 200 EGP

$$(1000,3000) + 200 *5 = (2000 , 4000)$$

Our project well cost (2000 EGP , 4000 EGP).



3.1 INTRODUCTION

the design phase is where our vision starts to take shape and gain momentum. We understand that design isn't just about aesthetics; it's about creating a functional and intuitive platform that meets the needs of our users.

With the system requirements thoroughly analyzed, our focus shifts to crafting a design that seamlessly integrates these requirements into a cohesive whole. We recognize that design lays the foundation for everything that follows, shaping the user experience and ensuring the success of our project.

By approaching design as the first step in our development journey, we set ourselves up for success, building a solid framework upon which we can iterate and refine. Our goal is to create a website that not only meets but exceeds the expectations of our stakeholders, providing them with a valuable and enjoyable experience every step of the way.

3.2 USE CASE

OUR ACTORS

Users:

- are individuals who visit our website with the intention of purchasing a car
- play a crucial role in driving the purchasing process and are the primary beneficiaries of our platform's features and functionalities.

Admins:

- Admins are authorized personnel responsible for managing the backend operations of the car dealership website.



Actions For User

Search and View Products:

- Users can efficiently search for cars based on their preferences by filtering them according to year, type, or price. Alternatively, they can browse through all available products. Upon selecting a specific car, users are presented with comprehensive details, including transparent pricing displaying all costs such as interest, down payments, and the total amount. They can also access product reviews for informed decision-making.

Order Placement:

- Users have the option to directly make an order without scheduling an appointment to physically inspect the car. Once an order is placed, users can confirm, edit, or delete it within a specified period. This ensures flexibility and convenience in the purchasing process.

Appointment Scheduling:

- Users can schedule appointments to physically inspect the cars before making a purchase. The appointment scheduling system collects user details and schedules the meeting to avoid conflicting schedules. This ensures personalized interaction and enhances user satisfaction.

Customer Support:

- The website features a robust ticketing system where users can submit inquiries or complaints through the frontend interface. Backend endpoints manage the receipt, storage, and assignment of tickets to support agents. Integration with communication channels such as phone or live chat enables seamless interaction between users and support agents. Notifications keep users informed about the status of their inquiries or complaints, ensuring prompt resolution and customer satisfaction.

Feedback:

- is provided for users to share feedback, ideas, or suggestions about the website. This could involve a dedicated section for users to leave comments and ratings, facilitating continuous improvement and enhancing user experience.

For Admin:

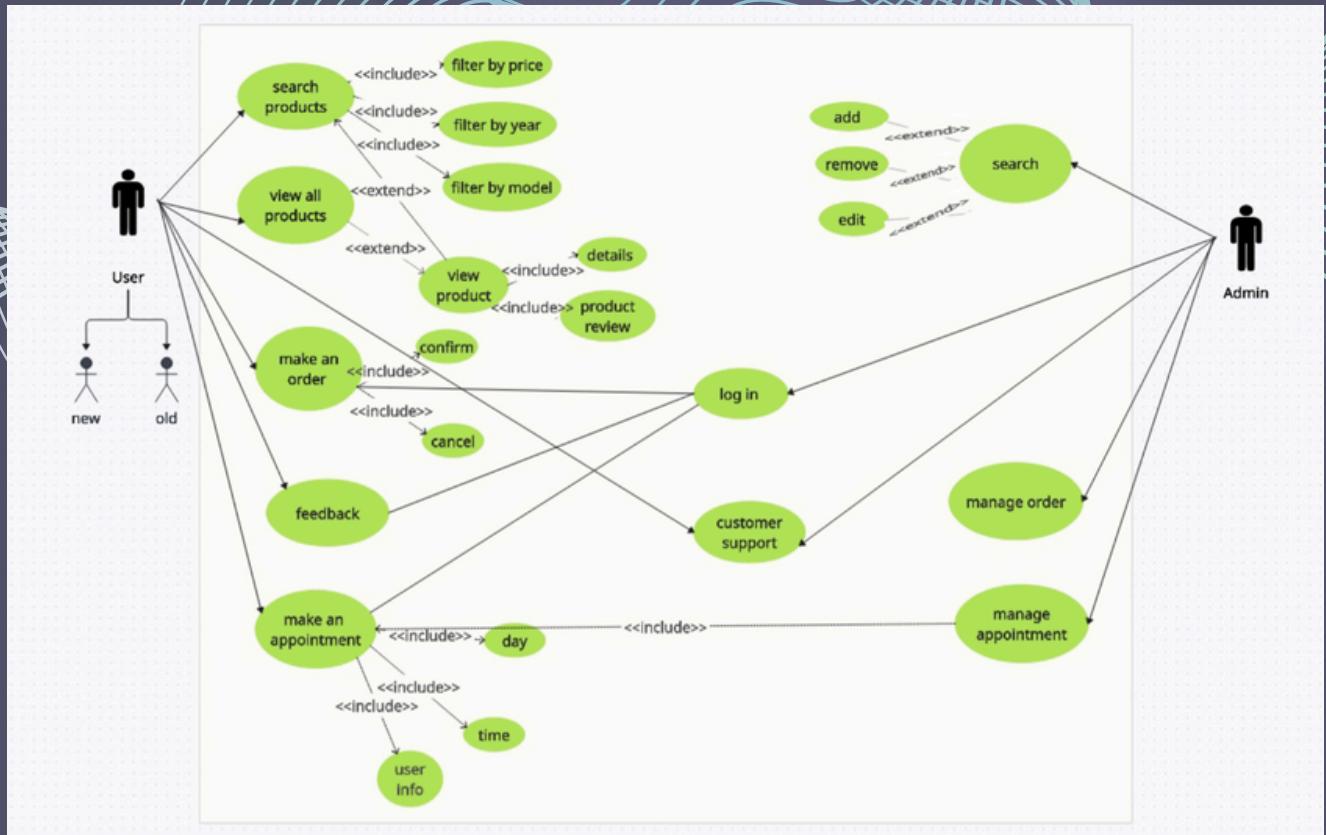
Admins are required to log in to access and manage the system.

Managers of the establishment are empowered to manage cars with equal permissions among themselves. They are able to add, delete, and edit cars on the website (**manage cars**)

3. he (**Manage order**) by seeing all orders , he oversee the processing, tracking, and fulfillment of customer purchases while ensuring inventory accuracy and resolving any issues that arise during the order lifecycle, aiming for seamless transactions and customer satisfaction.

4.he can (**Mange Appointments**),he is responsible for scheduling, coordinating, and organizing appointments for clients as team members are available and the car, ensuring smooth communication and efficient allocation of resources. Additionally, he handles rescheduling, cancellations, and any necessary adjustments

USECASE



CLASS DIAGRAM

We need class diagrams because they help us understand the structure of a system or software application at a high level.

They provide a blueprint for developers to follow when implementing the system, aiding in the organization of code and ensuring that all components work together seamlessly.

Class diagrams also facilitate communication among stakeholders, such as developers, designers, and clients, by providing a common visual language to discuss the architecture and design of the system.

In summary, class diagrams are essential tools for designing, communicating, and implementing software systems effectively.

INTRODUCTION FOR OURS

This report provides an overview of our car dealership software system, focusing on its class diagram and key functionalities.

The system is designed to streamline operations and enhance the customer experience.

Through this report, we aim to illustrate the structure and relationships of the classes within our system, highlighting their attributes and functions. Understanding these components is essential for grasping the system's architecture and functionality.

Aggregation, and composition.

USER CLASS:

This part of the system helps us know who our customers are and how to communicate with them. It stores details like their name, contact information, and job title. We can use it to send personalized messages or offers based on their interests or past interactions.

Additional uses:

01 **Targeted Marketing:**

Send personalized offers or promotions based on customer details.

02 **Loyalty Programs:**

Reward frequent customers with special discounts or benefits

03 **Customer Insights:**

Learn more about our customers' preferences and behaviors to improve our services.

CAR CLASS:

Here, we keep all the important information about the cars we have for sale. Customers can browse through this information to find a car they like. We also use it to manage our inventory, adding new cars and removing ones that are sold.

Additional uses:

01 **Inventory Management:**

Keep track of which cars are available and which ones have been sold.

02 **Vehicle Customization:**

Offer personalized options for customers to choose from.

MAKE APPOINTMENT CLASS:

This part helps customers schedule appointments to come and see a car in person. They can choose a day and time that works for them, and we'll make sure someone is available to assist them.

Additional uses:

01 **Staff Allocation:**

Assign staff members to appointments based on availability and expertise.

02 **Service Reminders:**

Send automated reminders to customers about their appointments

03 **Appointment Analytics:**

Track appointment data to improve scheduling efficiency.

MAKE AN ORDER CLASS

When customers decide to buy a car, they can use this class to place an order. We record details about the car they want and keep them updated on the status of their order.

Additional uses:

01 **Cross-Selling Opportunities:**

Recommend additional products or services during the ordering process.

02 **Order Tracking:**

Keep customers informed about the progress of their order.

CLASS DIAGRAM

OWNER CLASS:

This is for the owner or manager of the dealership. It helps them keep track of everything that's happening, from inventory and appointments to customer information.

Additional uses:

01 Performance Monitoring:

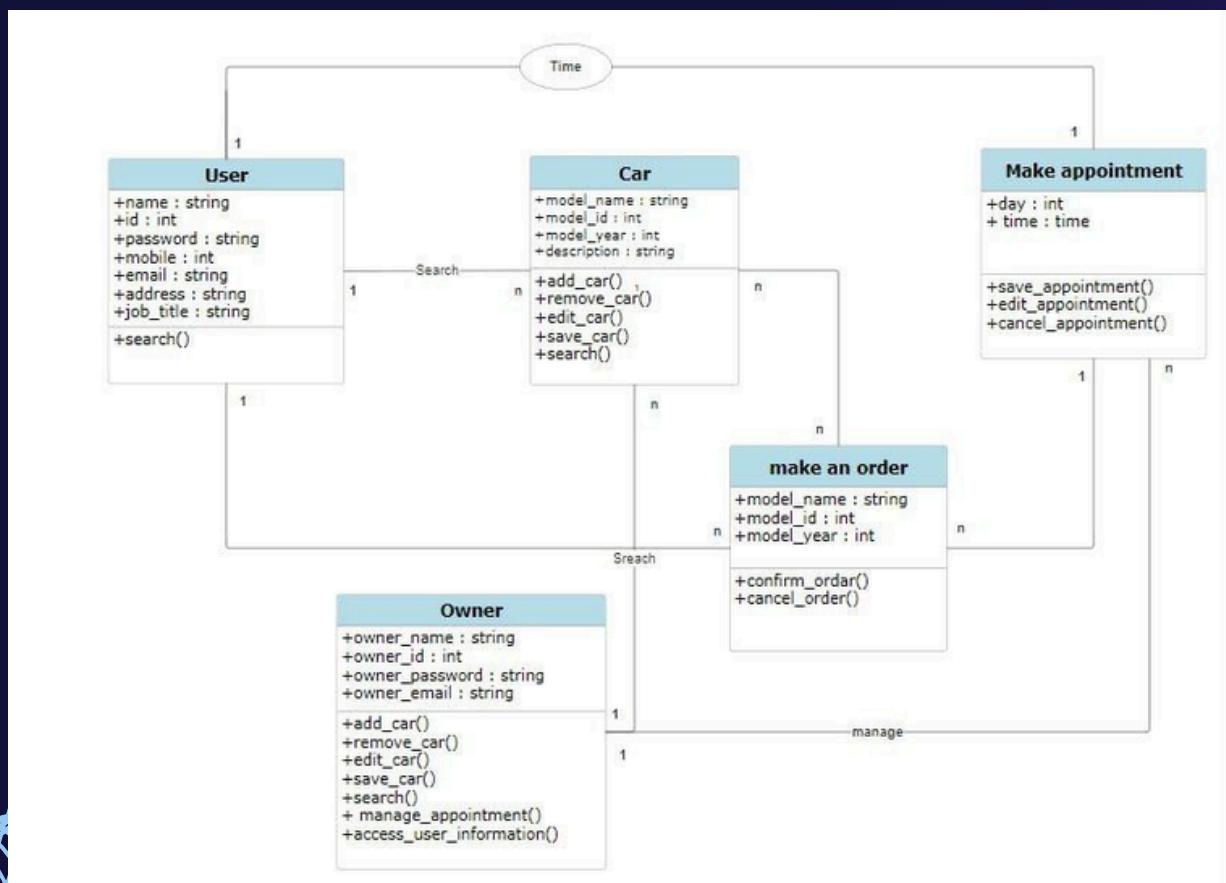
- Track sales and customer satisfaction to improve business operations.

02 Strategic Planning:

Make informed decisions based on market trends and competitor activity.

03 Regulatory Compliance:

Ensure the dealership meets legal requirements and industry standards.



Testing 4.0

In the context of our car dealership software system, robust testing is particularly important due to the critical nature of the operations it supports, such as inventory management, sales processing, and customer relationship management. By investing time and resources into testing,

By conducting thorough testing across these areas, We ensured that our car dealership software system meets the highest standards of quality, reliability, and usability.

1. User Interface Testing:

- We Verified that all user interface elements (buttons, forms, menus) are displayed correctly and are functional.
- Test the navigation flow throughout the application to ensure ease of use.
- Then Validate that user input fields have appropriate validation and error handling.

2. Functionality Testing:

- Test the functionality of core features such as adding, editing, and deleting cars from the inventory.
- Verify that search and filter functionalities work correctly to help users find specific cars.
- Test the ability to generate and print invoices or receipts for car sales.

3. Integration Testing:

- Ensure that the software integrates seamlessly with any external systems or databases used for inventory management or finance.
- Test the integration with any third-party services such as payment gateways or vehicle history report providers.

4. Performance Testing:

- Evaluate the system's performance under various loads by simulating multiple users accessing the system simultaneously.
- Measure the time taken to perform critical operations like adding a new car or processing a sale.

Testing

4.0

Continued

5. Security Testing:

- Check for vulnerabilities such as SQL injection or cross-site scripting (XSS) attacks.
- Ensure that user authentication and authorization mechanisms are robust and protect sensitive data.

6. Compatibility Testing:

- Test the software on different web browsers (e.g., Chrome, Firefox, Safari) and ensure compatibility.
- Verify that the software works seamlessly on various devices, including desktops, laptops, and mobile devices.

7. Localization and Internationalization Testing:

- Verify that the software supports multiple languages and locales if applicable.
- Test the software's ability to handle different date formats, currencies, and other regional settings.

8. Usability Testing:

- Conduct usability tests with representative users to gather feedback on the software's ease of use and intuitiveness.
- Incorporate user feedback to make necessary improvements to the user interface and workflow.

9. Regression Testing:

- After any updates or changes to the software, conduct regression testing to ensure that existing functionality has not been affected.
 - Automate regression tests wherever possible to streamline the testing process and ensure consistent results.
10. Documentation Review:
- Review all user manuals, technical documentation, and help guides to ensure accuracy and completeness.
 - Verify that the documentation adequately supports both end users and administrators in using and maintaining the software.

5.0 CODE STYLE

1. HTML Structure:

- We utilize semantic HTML5 elements to enhance accessibility and SEO, ensuring our website is easily navigable for users and optimized for search engines.
- Our team maintains a logical and consistent organization of HTML markup, facilitating easier understanding and maintenance of the codebase.
- Indentation is used to clearly delineate the hierarchy of nested elements, promoting readability and efficient debugging.

2. CSS Styling:

- We employ external CSS files to segregate styling concerns from HTML markup, fostering modularity and maintainability within our codebase.
- Adhering to a consistent naming convention for CSS classes and IDs, such as BEM or SMACSS, helps us streamline collaboration and facilitate code comprehension.
- Comments within our CSS code provide context and documentation, enhancing clarity and aiding future modifications.

3. JavaScript Interactivity:

- Behavior is separated from HTML markup by employing external JavaScript files, promoting code organization and facilitating code reuse.
- We develop modular and reusable JavaScript functions to enhance interactivity, enabling the implementation of complex features with ease.
- Consistent naming conventions for variables and functions are maintained, promoting code consistency and easing maintenance efforts.

4. Responsive Design:

- We design the website to be responsive and mobile-friendly using CSS media queries, ensuring optimal user experience across a variety of devices and screen sizes.
- Thorough testing on various devices and screen resolutions is conducted to verify compatibility, guaranteeing that our website remains accessible to all users.

5. Accessibility:

- We ensure accessibility by employing appropriate HTML elements, attributes, and ARIA roles, making our website usable for individuals with disabilities.
- Descriptive alternative text for images and multimedia content is provided, ensuring accessibility and compliance with web accessibility standards.
- Testing the website with screen readers and other accessibility tools helps us identify and address potential issues, ensuring an inclusive user experience for all.

Our Web Site

HOME
PAGE

GRAND ROYAL

HOME ABOUT SERVICE CARS TEAM CONTACT

Pickup Location Drop Location Pickup Date Pickup Time Select A Car Search

BUY A CAR

Best Cars In Your Location

Reserve Now

Activate Windows
Go to Settings to activate Windows.

01

WELCOME TO ROYAL CARS



At Royal Cars, we're dedicated to providing you with a regal automotive experience from start to finish. As your premier destination for luxury vehicles, exceptional service, and unparalleled expertise, we're here to make your car-buying journey truly majestic. Explore our curated collection of premium vehicles from the world's most prestigious brands. Whether you're in search of a sleek sedan, a powerful SUV, or a timeless classic, we have the perfect car to match your discerning taste. We believe that luxury should be accessible to all, which is why we offer competitive pricing and flexible financing options to help you drive home in the car of your dreams without compromise.

 WE SELL CARS NON STOP

 CAR RESERVATION ANYTIME

 ANY MEETUP LOCATIONS

Activate Windows
Go to Settings to activate Windows 

Details about the Car Agency:

We provided comprehensive information about our car agency on the homepage to establish credibility and transparency with users. This includes our history, mission, values, and team members.

Scope and Complexity

The more features and functionalities included in the software, the larger the codebase is likely to be. Complex features like automated scheduling algorithms or advanced reporting capabilities can contribute to increased size.

Technology Stack:

The choice of programming languages, frameworks, and libraries can impact codebase size. For example, using a high-level language with built-in libraries may result in a smaller codebase compared to a lower-level language requiring more manual implementation.

Architecture and Design:

Well-structured and modular codebases tend to be more manageable and maintainable but may require additional code for abstraction and separation of concerns.

Reuse and Third-Party Components:

Leveraging existing code libraries, frameworks, or third-party components can reduce the overall size of the software system. However, integrating these components may introduce additional dependencies and complexity.

03 FIND YOUR CAR
A grid of six car thumbnails with details:

- JEEP RANGER - 2018
- MERCEDES-Benz E3
- BMW 520i - 2010
- CHEVROLET OPTIMA 2020
- KIA SOUL
- MERCEDES-BENZ GLC 300

04 CONTACT US
A contact form with fields for Name, Email, Subject, Message, and a Send button. Below the form are office locations and customer service information.

GRAND ROYAL

HOME ABOUT SERVICE CARS TEAM CONTACT

Pickup Location Drop Location Pickup Date Pickup Time Select A Car Search

ABOUT

HOME / ABOUT

Details about the Website:

We provided informative content about our website on the About page to help users understand our platform better. This includes details about our website's purpose

Feedback from Customers:

One of the highlights of our About page is the inclusion of authentic feedback and testimonials from our valued customers. We showcase real experiences and testimonials from users who have interacted with our car agency, providing social proof of our reliability, quality of service, and customer satisfaction. By featuring positive feedback, we aim to build trust and credibility with potential customers, reassuring them of the exceptional experience they can expect when choosing our services.

WELCOME TO GRAND ROYAL

The best direct and bank installment systems for all types and models of cars in the best cases.

WE SELL CARS NON STOP CAR RESERVATION ANYTIME LOTS OF MEETUP LOCATIONS

OUR CLIENT'S SAY

AHMED ELSAYED
Test Automation Engineer

It is Amazing system where you can find, buy and sell cars, very easy to deal with this site!

HABIBA MOHAMED
Data Science

Their customer service is second to none.

MOAMEN ALAA
Lawyer

The product quality is consistently outstanding, exceeding my expectations every time.

CARS LIST

CARS ▾

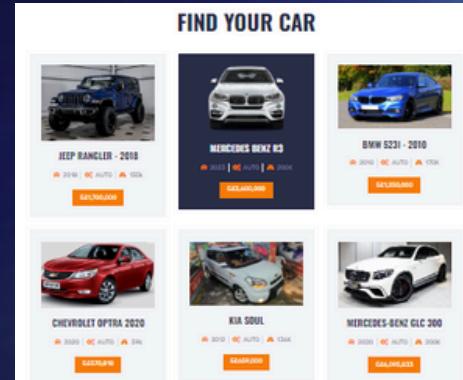
Car Listing

Car Detail

Car Booking

Page 1: Car Listing

Our primary goal with the Car Listing page was to create a comprehensive showcase of all the cars available on our website. We carefully curated the listings to include a diverse range of vehicles, catering to various preferences and needs. Users can effortlessly scroll through the page, exploring different makes and models at their own pace. Each listing provides essential information such as the make, model, year, and price, giving users a quick overview of the available options.



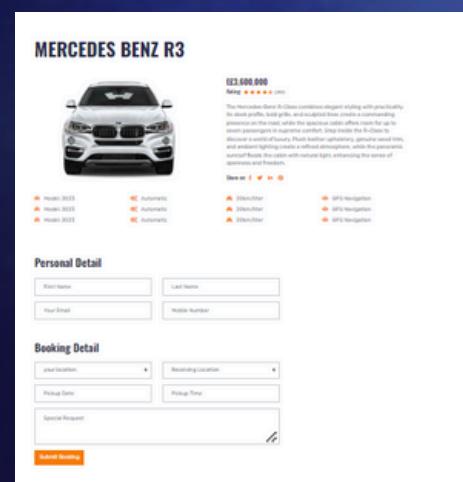
Page 2: Car Detail

Upon selecting a specific car from the Car Listing page, users are seamlessly directed to the Car Detail page, where they can delve deeper into the vehicle's specifications and features. Here, we provide users with a wealth of information, including detailed specifications, features, and high-quality images of the car. Our goal is to empower users to make informed decisions by offering a comprehensive understanding of the car's condition, performance, and amenities. By providing this level of detail, we aim to instill confidence in users and help them narrow down their choices effectively.



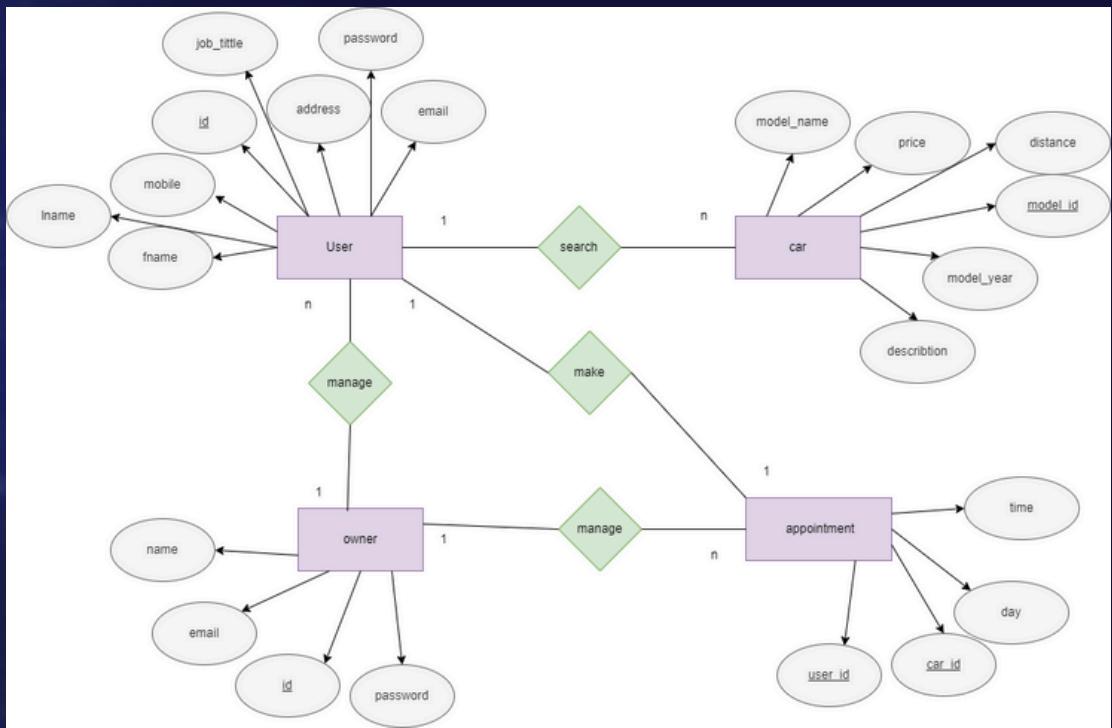
Page 3: Car Booking

Once users have identified a car they're interested in purchasing, we streamline the process of scheduling an appointment for examination through the Car Booking page. Here, users can conveniently select a date and time for their appointment, as well as provide personal details and specify the location for the examination. Our objective is to make the booking process as seamless and hassle-free as possible, ensuring that users can easily arrange to inspect the car in person before making a purchase decision. By facilitating this step, we demonstrate our commitment to transparency and customer satisfaction, allowing users to thoroughly evaluate the car and make an informed purchase.



Functions we made

First Our DataBase



- **booking_details**: This table stores details about bookings made by users, including their personal information, pickup/drop locations, appointment date and time, and any special requests.
- **car**: The "car" table contains information about the cars available in our inventory. It includes details such as the model name, model ID, model year, description, price, and distance traveled.
- **contact_form_submissions**: This table logs submissions made through the contact form on our website. It captures details such as the submitter's name, email, subject, message, and submission time.
- **make_appointment**: Here, we record appointments made by users to examine specific cars. It includes the user ID, appointment date, time, and the corresponding car ID.
- **owner**: This table stores information about car owners, including their name, ID, password, and email.
- **user**: The "user" table contains data about registered users of our website. It includes their first name, last name, ID, password, mobile number, email, address, and job title.

First Two functions signup & login

- **Database Connection:** I implemented the database connection logic to establish a secure connection to the MySQL database where user data is stored. This involved using the provided credentials to create a connection object (\$conn) using the mysqli class.
- **Error Handling:** I enabled error reporting to ensure that any errors encountered during script execution are displayed for debugging purposes. This helps in identifying and resolving issues promptly.
- **Sign-Up Functionality:** I developed the sign-up functionality to allow users to create new accounts on the website. This involved sanitizing and validating user input data to prevent SQL injection attacks and ensure data integrity. I also implemented checks to verify the uniqueness of email addresses before inserting new user records into the database.
 - I checked if all fields were filled using PHP's empty() function.
 - Password complexity was enforced by ensuring it was at least 9 characters long and contained both letters and numbers. This was achieved using strlen() and preg_match() functions respectively.
 - Email format was validated using PHP's filter_var() function with the FILTER_VALIDATE_EMAIL filter.
 - I also ensured that the email contained '@' and ended with '.com' using strpos() function.
 - Phone number format was checked using a regular expression (preg_match()) to ensure it consisted of 11 digits without any spaces or special characters.
 - I performed a database check to verify the uniqueness of the email address before inserting a new user record.
 - e sign-up functionality to allow users to create new accounts on the website. This involved sanitizing and validating user input data to prevent SQL injection attacks and ensure data integrity. I also implemented checks to verify the uniqueness of email addresses before inserting new user records into the database.
 - **Login Functionality:** I implemented the login functionality to authenticate users based on their email and password credentials. This involved querying the database to validate user credentials and setting session variables upon successful authentication. I also handled error messages for unsuccessful login attempts.

Function three Booking

- **Include Database Connection:** The first line includes the PHP file containing the database connection code (db_connection.php). This file establishes a connection to the database.
- **Retrieve Form Data:** We retrieve the form data submitted via the POST method and store it in respective variables. This data includes first name, last name, email, mobile number, pickup location, drop location, pickup date, pickup time, and any special requests.
- **Prepare SQL Statement:** We prepare an SQL INSERT statement to insert the form data into the booking_details table in the database. The statement uses placeholders (?) for the values to be inserted.
- **Bind Parameters:** We bind the retrieved form data to the prepared SQL statement using the bind_param() method. This helps prevent SQL injection attacks by automatically escaping special characters in the input values.
- **Execute SQL Statement:** We execute the prepared SQL statement using the execute() method. This inserts the form data into the database.
- **Close Statement and Database Connection:** We close the prepared statement and the database connection to free up resources and ensure data integrity.
- **Redirect:** Finally, we redirect the user back to the booking page (booking.html) or any other desired page using the header() function. This ensures a smooth user experience after the form submission.

Function Four Contact Us

Script 1:

- **Include Database Connection Script:** It includes a script named db_connection.php that establishes a connection to the database.
- **Form Submission Check:** It checks if the form is submitted using the `$_SERVER["REQUEST_METHOD"]` variable.
- **Data Validation and Sanitization:** It sanitizes and validates form input data (name, email, subject, message) using `mysqli_real_escape_string()` to prevent SQL injection attacks.
- **Check Table Existence:** It verifies if the table (contact_form_submissions) exists in the database by executing a `SHOW TABLES LIKE '$tableName'` query.
- **Insert Data into Database:** If the table exists, it constructs an SQL query to insert the form data into the table.
- **Redirection:** After successful data insertion, it redirects the user to index.html.
- **Error Handling:** If there's an error during the insertion process, it displays an error message.
- **Close Database Connection:** Finally, it closes the database connection

Script 2:

- **This script handles the submission of a booking form.**
- **Include Database Connection Script:** It includes a script named db_connection.php that establishes a connection to the database.
- **Form Submission Check:** It checks if the form is submitted using the `$_SERVER["REQUEST_METHOD"]` variable.
- **Data Validation and Sanitization:** It sanitizes and validates form input data (first_name, last_name, email, mobile_number, pickup_location, drop_location, pickup_date, pickup_time, adult, child, special_request) using `mysqli_real_escape_string()`.
- **Insert Data into Database:** It constructs an SQL query to insert the form data into the booking_details table.
- **Redirection:** After successful data insertion, it redirects the user to index.html.
- **Error Handling:** If there's an error during the insertion process, it displays an error message.
- **Close Database Connection:** Finally, it closes the database connection.