﻿1. Title of the Project﻿Automated Car Catalog System for Enhanced Showroom Management﻿2. Introduction﻿The automotive showroom industry faces challenges such as manual data handling, inefficient customer engagement, and lack of real-time inventory visibility. This project proposes an automated car catalog system that simplifies vehicle listing, enhances customer interaction, and streamlines inventory management.﻿3. Objectives﻿To automate the car cataloging process in the showroom.﻿To enable quick access to car details by showroom staff and customers.﻿To maintain a real-time inventory of available cars.﻿To support customer-centric features such as car comparisons and search filters.﻿4. Scope of the Project﻿Car information management (model, variant, color, fuel type, price).﻿Staff and admin panel for managing car entries.﻿Customer interface for browsing cars.﻿Search and filter functionalities.﻿Optional integration with CRM or booking systems.﻿5. System Architecture﻿Frontend: Web-based UI using HTML/CSS/JavaScript or frameworks like React/Angular.﻿Backend: Server-side logic using PHP, Python (Django/Flask), or Node.js.﻿Database: MySQL, PostgreSQL, or MongoDB for storing car and user data.﻿Optional: REST API for third-party integrations.﻿6. Modules of the System﻿a) Admin Panel﻿Add/Edit/Delete car entries﻿Manage inventory status﻿View booking or enquiry logs﻿b) Customer Interface﻿Browse available cars﻿Search and filter options (brand, model, fuel type, price)﻿Car comparison feature﻿Enquiry or request test drive option﻿c) Inventory Management﻿Real-time tracking of available cars﻿Stock updates﻿Notifications for low stock or sold-out vehicles﻿7. Functional Requirements﻿Admin login and role-based access﻿Car listing with details (images, specs, price)﻿Advanced search and filter mechanism﻿Inquiry submission system for customers﻿8. Non-Functional Requirements﻿User-friendly and responsive UI﻿Secure login system﻿Scalable architecture to add more features (e.g., loan calculator, insurance module)﻿Fast loading times﻿9. Technology Stack﻿Frontend: HTML, CSS, JavaScript (React/Angular optional)﻿Backend: Python Flask/Django OR PHP OR Node.js﻿Database: MySQL / PostgreSQL / MongoDB﻿Tools: Git, Visual Studio Code, XAMPP/WAMP (for PHP-based)﻿10. Advantages﻿Reduces manual workload﻿Improves accuracy and speed in cataloging﻿Enhances customer experience﻿Provides insights into inventory trends﻿11. Future Enhancements﻿Integration with CRM for sales tracking﻿Mobile app version﻿Real-time chat or chatbot for customer support﻿AI-based car recommendation engine﻿12. Conclusion﻿The automated car catalog system provides a modern solution to showroom management by digitizing and streamlining the catalog process. It improves operational efficiency, supports informed customer decision-making, and enhances overall showroom productivity.