1. Introduction

This web application serves as a platform where users can browse, purchase, and manage vehicles. Targeted primarily for customers looking to buy cars, electric vehicles, and motorcycles, it offers both an intuitive user experience and an admin interface for managing vehicle listings. Customers can view detailed information about each product, create and manage orders, and handle their personal data securely. Admin users have elevated privileges to manage products, view user details, and oversee orders.

2. Technology Stack

The application is built using modern web technologies to ensure scalability, performance, and security. The primary technologies used in the project include:

ASP.NET Core : For building the web API and MVC-based application.

Entity Framework Core : For object-relational mapping (ORM) to interact with the PostgreSQL database.

SQL Server : For data storage, providing robust transaction management and support for relational data.

ASP.NET Identity : For user authentication and authorization.

Razor Pages : To render dynamic content for the front-end of the application.

JavaScript / jQuery : For front-end interactivity, such as handling the cart.

Bootstrap : For responsive, mobile-friendly design.

3. Database Structure

The database is structured around core business entities, with relationships defined between tables to ensure data integrity and support the application's functionality.

Users: Stores user information, including credentials, roles (e.g., Customer, Admin), and personal details.

Products (Vehicles): Contains vehicle details (e.g., brand, model, type, price), with additional metadata (e.g., images, categories).

Orders : Represents customer orders, with fields linking the order to the user and the products they purchased.

OrderProducts: A bridge table for the many-to-many relationship between Orders and Products, containing the quantity and price details of each product in the order.

Payments: Records transactions, linking users to their payments.

The application performs CRUD (Create, Read, Update, Delete) operations across these tables, ensuring accurate data management.

4. Authentication and Authorization

User authentication and role-based authorization are handled by ASP.NET Identity, which provides a secure method for registering, logging in, and managing users. There are two primary roles:

Customer : Can view products, manage orders, and make payments.

Admin : Has elevated privileges to manage products, view user details, and delete users.

The application uses hashed passwords for secure user authentication and enforces role-based access control to restrict actions according to user roles.

5. Application Flow

User Registration/Login : Users can sign up by providing their email, password, and basic details. Upon successful registration, they are logged in and assigned the "Customer" role by default. Admins can log in using their credentials, but only they can access the product management section.

Browsing Products : Users can browse available vehicles, filtering by category (e.g., electric cars, motorcycles).

Shopping Cart : Users can add, update, or remove products in their shopping cart. The cart is stored in the session for persistence during the browsing session.

Order Placement : Users can review their cart and proceed to checkout, creating an order and providing payment details.

Admin Actions : Admins can create, edit, or delete products, as well as manage user accounts and view all orders.

6. Functionalities

6.1. User Registration and Login

Registration : Users can register by providing their email, username, and password. The password is hashed using a secure algorithm to ensure protection. After registration, users are automatically logged in and assigned a Customer role.

Login : Registered users can log in with their credentials, and roles are used to differentiate between Admin and Customer actions. On successful login, the user is redirected to their dashboard or product browsing page.

6.2. Vehicle Management

Vehicle Retrieval : Users can filter vehicles by category and search. Pagination displays 10 vehicles at a time.

Vehicle Details : Users can view the details of a selected vehicle.

Vehicle Creation (Admin) : Admin can add a new vehicle. Vehicle information and image are uploaded and saved to the database.

Vehicle Editing (Admin) : Admin can update the details and image of an existing vehicle.

Vehicle Deletion (Admin) : Admin can mark a vehicle as deleted (set as inactive)

6.3. Cart Management

The application allows users to manage their shopping cart:

AddToCart : Users can add products to the cart by specifying the quantity.

RemoveFromCart : Products can be removed from the cart.

Update Quantity : Users can adjust the quantity of each item in their cart.

Clear Cart : The cart can be cleared at any time, removing all items.

The cart is persisted in the user's session to retain data across multiple pages.

6.4. Order Management

Order Creation : After reviewing the cart, users can place an order. The application creates an Order entry in the database and associates it with the user's account.

Stock Update : When an order is created, the stock quantity for the ordered products is reduced accordingly.

Order Details : Users can view their past orders, including the products purchased, total price, and order status.

6.5. Payment and Checkout

During the checkout process, users must provide an address and payment details. The application ensures that the user has a valid address before proceeding with the payment. Upon successful payment, the order is marked as completed, and the user's order history is updated.

6.6. Admin Management

Admin users have additional privileges, such as:

Managing Products : Admins can create, edit, and delete products from the inventory.

User Management : Admins can view all registered users, update their details, and delete users if necessary. Pagination is used to display users in manageable chunks.

Order Overview : Admins can view all orders placed by users, including details like product quantity and user information.