TECHNICAL DESIGN DOCUMENT

Project:

Velocity Vehicles

Prepared by:

Byte Warriors

2025

Table of Contents

1 INTRODUCTION

1.1 THE COMPOSITION OF THE DOCUMENT

The functional design document (FDD) contains:

- Scope
- Database design : Details of the DB solution.
- Architecture solution design: Object model, Data model, System Layered Design
- User Interface
- Validation process of entry object
- Summary

1.2 ABBREVIATIONS

Abbreviations	Transcript
ER	Entity Relationship
DB	Database
FDD	Functional Design Document
TDD	Technical Design Document

2 TECH STACK

- Database: MySQL
- Backend: Java, Spring, Spring Boot.
- Frontend: Angular, Figma.

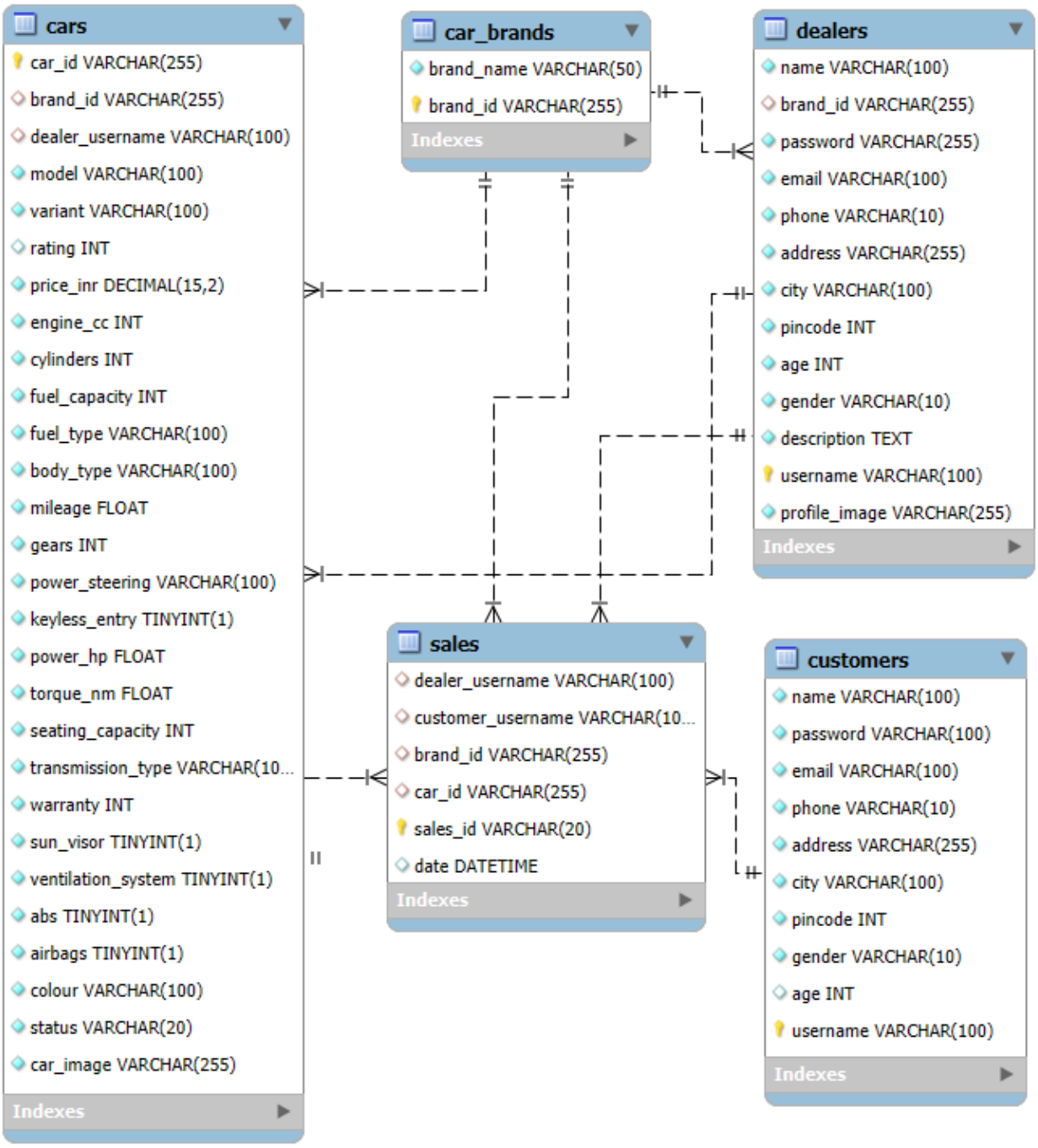
3 DATABASE

This database supports a velocity Vehicles website, enabling dealers to list cars, customers to purchase them, and sales to be tracked. The main tables are **Cars**, **Dealers**, **Customers**, **Car_brands**, and **Sales**, each with relations based on unique identifiers and foreign keys.

3.1 TABLE DETAILS

- ***Cars:*** Stores details about each car; linked to car_brands by Brand_id and to dealers by DealerUsername.
- ***Car_brands:*** Stores car brand information; referenced by both cars and dealers.
- ***Dealers:*** Stores dealer/user info; references car_brands through Brand_Id; linked to sales and cars by DealerUsername.
- ***Customers:*** Stores customer/user info; referenced by sales using customerUsername.
- ***Sales:*** Records each car sale; relates to cars, dealers, customers, and car_brands through their respective keys.

3.2 ER DIAGRAM



3.3 TABLE FIELDS DESCRIPTION

Below I describe all the fields of each table.

- ***Car_brands***

Car_brand	Name of the car brand.
Brand_id	Unique identifier for the brand.

- ***Dealers***

Name	Dealer's full name.
Password	Dealer login password.

Email	Dealer email address.
Address	Dealer's address.
City	Dealer's city.
Pin	Dealer's postal code
Age	Dealer's age.
Gender	Dealer's gender.
username	Unique dealer username (dynamically generate in the code).
Brand_Id	Car brand dealership (foreign key).
Description	Brief description about the dealer
Profile_image	Dealer's profile image url

- ***Customers***

Name	Customer's full name.
Password	Customer login password.
Email	Customer email address.
Address	Customer's address.
City	Customer's city.
Pin	Customer's postal code
Age	Customer's age.
Gender	Customer's gender.
username	Unique Customer username (dynamically generate in the code).

- ***Cars***

Car_id	Unique identifier for the car (dynamically generate in the code).
Brand_ID	Brand ID from car_brands (foreign key).
DealerUsername	The dealer's username (foreign key).
Model	Car model.
Variant	Variant of the car.
Price_INR	Price in INR.
Engine (CC)	Engine capacity in CC.
Cylinders	Number of engine cylinders.
Fuel_capacity	Fuel tank capacity.
Fuel_type	Type of fuel used.
Body_type	Car body type (SUV, Sedan, etc.).
Mileage	Car mileage.
Gears	Number of gears.
Power_steering	Indicates power steering type or presence.
Keyless_entry	Indicates if the car has keyless entry.
Power (HP)	Power of the Engine in Horsepower.
Torque (NM)	Rotational force of the car in Newton-meters.
Seating_capacity	Number of seats.
Transmission_type	Manual/Automatic/CVT.
Warranty	Warranty years.
Sun_visor	Is the car have sun visor or not
Ventilation_system	Is the car have advance ventilation system or not

ABS	Is the car have Advance Breaking System or not
Airbags	Is the car have Airbags or not
Colour	Colour of the car
Status	Car Status (Sold or Available)

- **Sales**

DealerUsername	Related dealer (foreign key).
customerUsername	Related customer (foreign key).
Brand_ID	Sold car's brand (foreign key).
Car_ID	Sold car (foreign key).
sales_ID	Unique transaction identifier (dynamically generate in the code).
Date	Selling date

3.4 RELATIONSHIP

- cars → dealers: Cars reference the dealer using DealerUsername.
- cars → car_brands: Cars reference the brand using Brand_ID.
- dealers → car_brands: Dealers are associated with a brand using Brand_Id.
- sales → dealers, customers, cars, car_brands: Each sale records the involved dealer, customer, car, and brand.

4 BACKEND

4.1 PACKAGE STRUCTURE:

- config (for JDBC configuration)
- controllers (for handling HTTP requests)
- models (for entity classes)
- dao (interfaces and implementations for DB access)
- services (interfaces and implementations for business logic)
- utils (helper classes, if needed)
- exceptions (for error handling)

4.2 SYSTEM LAYERED ARCHITECTURE:

- Presentation Layer (Controllers)
- Business Logic Layer (Services)
- Data Access Layer (DAOs)
- Database (MySQL)

4.2.1 CONTROLLER

Responsible for handling **HTTP requests** from the frontend. Maps requests appropriate **service methods**.

a. AuthController: This controller manages user authentication flows, including user login, new customer/dealer registration, and authenticated user profile retrieval.

Type API	API Name	Functionality
POST	/auth/login	Authenticates user credentials and returns a JWT.
GET	/auth/profile	Retrieves the profile details of the authenticated user.
POST	/auth/register-customer	Register a new customer and log them in.
POST	/auth/register-dealer	Registers a new dealer with an optional profile image and logs them in.

b. CarBrandController: This controller provides public access to car brand master data.

Type API	API Name	Functionality
GET	/car-brand/all	Fetches a list of all available car brands.

c. **CustomerCarController:** This controller provides public-facing car browsing, search, and filtering functionalities, along with customer-initiated car sales.

Type API	API Name	Functionality
GET	/customer/cars/{carId}	Retrieves detailed information for a specific car by its ID.

GET	/customer/cars/search/mileage	Filters available cars based on a specified mileage range.
GET	/customer/cars/get-available-cars	Fetches a list of all cars currently marked as available.
POST	/customer/cars/{carId}/sell	Records the sale of a specific car to the authenticated user.
GET	/customer/cars/by-brand	Retrieves available cars filtered by a specific car brand ID.
GET	/customer/cars/get-top-available-cars	Fetches a limited number of top available cars.
POST	/customer/cars/filter	Filters available cars based on multiple comprehensive criteria.

d. CustomerController: This controller provides access to individual customer details.

Type API	API Name	Functionality
GET	/customers/{username}	Retrieves the details of a specific customer by username.

e. CustomerSalesController: This controller provides access to sales records from a customer's perspective, focusing on their purchases.

Type API	API Name	Functionality
GET	/customer/sales/by-customer	Retrieves sales records associated with a given customer username.
GET	/customer/sales/cars-bought	Fetches a list of all cars bought by the authenticated customer.

f. DealerCarController: This controller manages car inventory operations specific to a dealer.

Type API	API Name	Functionality
POST	/dealer/cars/add	Allows a dealer to add a new car to their inventory.
GET	/dealer/cars/inventory/{dealerUsername}	Retrieves the car inventory for a specified dealer.

g. DealerController: This controller provides public access to general dealer information.

Type API	API Name	Functionality
GET	/dealers/search/by-brand	Finds dealers associated with a specific car brand ID.
GET	/dealers/top-by-brand	Fetches a limited list of top-performing dealers.

h. DealerSalesController: This controller manages and retrieves sales records from a dealer's or administrative perspective.

Type API	API Name	Functionality
GET	/dealer/sales/all	Retrieves all sales records in the system.
POST	/dealer/sales/add	Adds a new sales record.
GET	/dealer/sales/{salesId}	Retrieves a specific sales record by its ID.
GET	/dealer/sales/by-car/{carId}	Retrieves the sales record associated with a specific car ID.
GET	/dealer/sales/by-dealer	Retrieves sales records for a specified dealer username.
GET	/dealer/sales/sold-cars-details	Fetches detailed information about cars sold by the authenticated dealer

4.2.2 SERVICE LAYER

The service layer contains all the business logic and its intermediary between the controller and dao.Each major entity will have a corresponding service interface and implementation:

a. AuthService : This service handles core user authentication functionalities, including user login, registration for both customers and dealers, and retrieval of authenticated user profile information.

Method	Functionality
login(AuthRequest request)	Authenticates a user based on provided credentials and generates a secure authentication response.
registerCustomer(CustomerCreationDto customerCreationDto)	Registers a new customer account in the system.
registerDealer(DealerCreationDto dealerCreationDto, MultipartFile profileImageFile)	Registers a new dealer account, optionally including a profile image.
getMyProfile(String username, UserDetails userDetails)	Retrieves the profile details for a given authenticated user, identified by username.

b. CarBrandService : This service defines the contract for operations related to car brands, allowing for retrieval of all available brands and finding a brand by its unique identifier.

Method	Functionality
getAllBrands()	Retrieves a list of all available car brands.
findById(String id)	Finds a car brand by its unique identifier.

c. CarService: This service manages car entities and their associated data, providing methods for car creation, conversion, retrieval by various criteria, and comprehensive search capabilities.

Method	Functionality
getCar(String carId)	Retrieves a car entity by its unique identifier.
convertToDto(Car car)	Converts a Car entity to a CarDto.
convertToCar(CarCreationDto carCreationDto, MultipartFile carImageFile)	Converts car creation DTO and image file into a Car entity.
saveCar(CarCreationDto carCreationDto, MultipartFile carImageFile)	Saves a new car record to the system, including handling its image.
getAllCars()	Retrieves a list of all cars available in the system.
getCarsByMilageRange(double minMileage, double maxMileage)	Filters and retrieves cars within a specified mileage range.
getCarById(String carId)	Retrieves a car's DTO by its unique identifier.
getDealerInventory(String dealerUsername)	Retrieves all cars associated with a specific dealer's inventory.
getCarsByBrandId(String brandId)	Retrieves a list of cars filtered by a specific brand ID.
getTopCars(int limit)	Fetches a limited number of top cars based on certain criteria.
searchCars(CarSearchCriteriaDto criteria)	Filters cars based on comprehensive search criteria provided in a DTO.

d. **Customer Service :** This service is responsible for managing customer entities, including their creation, retrieval by username, and conversion to data transfer objects.

Method	Functionality
convertToCustomerDto(Customer customer)	Converts a Customer entity to a CustomerDto.
findByUsername(String username)	Finds a customer entity by their username.
addCustomer(CustomerCreationDto customerCreationDto)	Adds a new customer to the system, handling unique username and password encoding.
getCustomerById(String username)	Retrieves a customer's DTO by their username.
getAllCustomers()	Retrieves a list of all customer DTOs.
findById(String username)	Finds a customer entity by their unique identifier (username).

e. **DealerService:** This service handles operations related to dealer entities, including account creation, retrieval by various criteria, and conversion to data transfer objects.

Method	Functionality
findByUsername(String username)	Finds a dealer entity by their username.
convertToDealerDto(Dealer dealer)	Converts a Dealer entity to a DealerDto.
getDealerByUsername(String username)	Retrieves a dealer entity by their username.
createDealer(DealerCreationDto dealerCreationDto, MultipartFile profileImageFile)	Creates a new dealer account, including handling the profile image upload.
getDealersByBrandId(String brandId)	Retrieves a list of dealers associated with a specific car brand.
getAllDealers()	Retrieves a list of all dealer DTOs.
getTopDealersByBrand(int limit)	Retrieves a limited number of top dealers based on certain criteria.
getDealerById(String username)	Retrieves a dealer's DTO by their username.

f. **ImageUploadService:** This service provides functionalities for managing image uploads, including saving images to storage and retrieving default image URLs.

Method	Functionality
uploadImage(MultipartFile file, String folder, String publicId)	Uploads a given image file to a specified folder with a unique public ID.
getDefaultProfileImageUrl()	Retrieves the default URL for a user or dealer profile image.
getDefaultCarImageUrl()	Retrieves the default URL for a car image.

g. **SalesService:** This service manages car sales transactions, encompassing the addition of new sales, retrieval of sales records based on various criteria, and detailed reporting of sold cars for both dealers and customers.

Method	Functionality
convertToDto(Sales sales)	Converts a Sales entity to a SalesDto.
addSales(SalesDto salesDto)	Adds a new sales record based on the sales provided details.
getSalesById(String salesId)	Retrieves a sales record DTO by its unique identifier.
getSalesByDealerUserName(String username)	Retrieves sales records associated with a specific dealer's sales.
getSalesByCustomerUserName(String username)	Retrieves sales records associated with a specific customer's purchases.

getSalesByCarId(String carId)	Retrieves a sales record DTO by the unique ID of the car involved in the sale.
getAllSales()	Retrieves a list of all sales records in the system.
getSoldCarsDetailsByDealerUsername(String dealerUsername)	Retrieves detailed information about cars sold by a specific dealer.
getAllCarsBoughtByCustomer(String customerUsername)	Retrieves detailed information about all cars bought by a specific customer.
sellCar(String carId, String customerUsername)	Records a car sale transaction, linking a specific car to a customer.

5 USER INTERFACE

This section covers the High-level UI design description of elements of the Velocity Vehicles website.

5.1 HOME PAGE

The home page of Velocity Vehicles is scrollable and divided into several parts which are revealed as the user scrolls, which are described below.

5.1.1 Navbar

The navbar is present at the top of the landing page. The buttons in the navbars are :

- Logo: The logo is present at the top left corner of the home page.
- Home: It redirects to the default home landing page.
- About us: It redirects to the footer section containing **Why Choose Us?** section.
- Contact us: It redirects to the footer section containing details to communicate with us.
- Sign in: It will redirect to the sign in page.

5.1.2 Landing Section

- The landing page has our tagline followed by six primary category buttons- **SUV, Sedan, Hatchback, Coupe, MPV, MUV**.
- On clicking any one of the buttons four different filters follow showed with dropdown buttons- Brand Type, Fuel Type, Mileage Range and Price Range.
- The **"Search Car"** button positioned on the right of the filter section redirects to the page with the list of cars as per the criteria.

5.1.3 Explore Our Premium Brands

- This section contains a set of **9 clickable brand cards – Tata, BMW, Ford, Mercedes Benz, Hyundai, Mahindra** which on being clicked redirects to the page showing all the cars of the selected brand present with the dealers.

5.1.4 Explore All Vehicles

- This section displays a catalogue of cars in the sytem having cards containing image of the car, model name, price, mileage, fuel type and transmission type. On clicking the **"View Details"** hyperlink it redirects us to the **Selected Car Details.**

5.1.5 Dealers

- This section contains the list of dealer's details in the form of cards having image of the Car Dealer, his name, email address, contact number and address.

5.1.6 Footer

- This section contains Company, Our Brands, Vehicles Type , Connect with Us , CopyWrite and Terms and Conditions and Privacy notice.

5.2 THE CAR DETAILS PAGE

The Car Details Page two main containers.

- The left container has filters shown with dropdown buttons such as:
  - **Price Filter –** It has checkbox inputs for price range starting from the option Upto 3 lakhs and the last range option being more than 1 crore(INR)

- **Vehicle Type -** Select one of the checkboxes- SUV, Sedan, Hatchback, Coupe, MPV, MUV.
- **Fuel Type –** Select one of the three radio button – Petrol, Diesel, CNG
- **Transmission Type -** Select one of the two radio button – Automatic, Manual
- **Seating Capacity –** Select one of the radio button – 2, 4, 6, 7, 8, 9.

- The right container has a list of car cards with image of the respective car, its name, rating, vehicle type, price, mileage, fuel type and a button **"Check for more details"** clicking on which redirects us to the page with the selected car.

5.3 SELECTED CAR DETAILS

On clicking the **"Check for more details"** button it redirects it to this page having four different sections.

- **Image carousel** with swipe feature, dot indicators showing different image of the car in each slide .
- **The Car Specification container** having features – Engine type, Cylinders, Gears, Mileage, Power, Torque, Transmission, Fuel Tank Capacity, Seating Capactity, Waranty, Boot Space, Steering, Car Location.
- **The Car Details container** having Car model name, Rating, Car type, Mileage, Fuel Type, Home Test Drive available or not, Dealer Name, Price amd two buttons – BOOK NOW and FREE TEST DRIVE.
- **Top Features of the Car container** will have the Car's best features.

5.4 SIGN UP AND LOGIN PAGE

5.4.1 User Sign Up and Login

- When the user clicks on the **Search Cars** button without being loged in they are redirected to the Login/Sign Up page.
- The user if not registered will click on **"Sign Up"** and will be redirected to the Sign-Up page where they fill details in the text box such as- Name, Email Id, Password, Confirm password, age, Gender, Phone Number, address, City and Pin and then click **Sign Up**.
- If the user is a registered user, he/she will enter the username and password select the **Remember me** checkbox and login or login through their respective Facebook, Google or LinkedIn accounts.

5.4.2 Dealer Sign Up and Login

- For the dealers they directly click on **"Sign Up"** button on the top right corner they will be redirected to the Sign-Up page where they fill details in the text box such as- Name, Email Id, Password, Confirm password, Age, Gender, Car Company, Phone Number, Address, City and Pin and then click **Sign Up**, if not registered.
- If the dealer is a registered user, he/she will enter the username and password select the **Remember me** checkbox and login or login through their respective Facebook, Google or LinkedIn accounts.

6 VALIDATION

6.1 FRONTEND VALIDATION:

Frontend validation focuses on ensuring that user inputs are correctly formatted and meet the required criteria before being sent to the server. This helps in providing immediate feedback to users and reducing unnecessary server load.

6.1.1 Home page

- **Navbar -** Validate that all buttons (Logo, Home, About Us, Contact Us, Sign In) are clickable and redirect to the correct sections/pages.
- **Landing Section –**
  - Ensure primary category buttons (SUV, Sedan, Hatchback, Coupe, MPV, MUV) are clickable.
  - Validate dropdown filters (Brand Type, Fuel Type, Mileage Range, Price Range) function correctly.
  - Ensure "Search Car" button redirects to the correct page based on selected criteria.
- **Explore Our Premium Brands -** Validate brand cards are clickable and redirect to the correct brand page.
- **Explore All Vehicles -** Ensure car cards display correct information and "View Details" hyperlink redirects correctly.
- **Dealers -** Validate dealer cards display correct information and are clickable.
- **Footer -** Ensure links in the footer (Company, Our Brands, Vehicle Type, Connect with Us, Terms and Conditions, Privacy Notice) are functional.

6.1.2 Car Details Page

- **Filters -** Validate dropdown filters (Price, Vehicle Type, Fuel Type, Transmission Type, Seating Capacity) function correctly.
- **Car Cards -** Ensure car cards display correct information and "Check for more details" button redirects correctly.

6.1.3 Selected Car Details

- **Image Carousel -** Validate swipe feature and dot indicators.
- **Car Specification Container -** Ensure all specifications are displayed correctly.
- **Car Details Container -** Validate information and functionality of "BOOK NOW" and "FREE TEST DRIVE" buttons.
- **Top Features Container -** Ensure top features are displayed correctly.

6.1.4 Sign Up and Login Page

- **User Sign Up**
  - Validate input fields (Name, Email, Password, Confirm Password, Age, Gender, Phone Number, Address, City, Pin) for correct format and required fields.
  - Ensure "Sign Up" button functions correctly.
- **User Login**
  - Validate input fields (Username, Password) and "Remember me" checkbox.
  - Ensure login through Facebook, Google, LinkedIn functions correctly.
- **Dealer Sign Up**
  - Validate input fields (Name, Email, Password, Confirm Password, Age, Gender, Car Company, Phone Number, Address, City, Pin) for correct format and required fields.
  - Ensure "Sign Up" button functions correctly.
- **Dealer Login**
  - Validate input fields (Username, Password) and "Remember me" checkbox.
  - Ensure login through Facebook, Google, LinkedIn functions correctly.

7 SUMMARY

The **Velocity Vehicles** project is designed to create a website for car dealers and customers. The site will facilitate car listings, purchases, and sales tracking. Key components include:

- **Database**: Structured with tables for Cars, Dealers, Customers, Car_brands, and Sales, interconnected through unique identifiers and foreign keys.
- **Backend Architecture**: Organized into layers (Presentation, Business Logic, Data Access) with controllers managing HTTP requests, services handling business logic, and DAOs accessing the database.
- **Controllers**: Specific controllers for different car types (SUV, Sedan, Hatchback, etc.), dealers, customers, and sales.
- **Service Layer**: Services for managing operations related to cars, dealers, customers, and sales.
- **User Interface**: Features a landing page with a navbar and car type buttons for easy navigation and filtering.

The project aims to provide a seamless experience for dealers and customers, ensuring efficient management of car listings and sales.