AutoHub

- **Team Members:** Alvin Varghese, Anargh Jiju

- **Problem Statement:** Vehicle owners often find it challenging to buy or sell cars due to a lack of transparent pricing and trust in service providers. Additionally, managing routine car maintenance and repairs is difficult when there is no reliable platform to compare prices and services. There is a need for a comprehensive platform where car owners can buy, sell, or service their vehicles, ensuring transparency in pricing and service quality, while fostering trust between buyers, sellers, and service providers.

1. **Understanding of the Problem statement**

a) **Explanation of the Problem Context:**

In the automotive industry, buying and selling used cars, as well as maintaining them, can be a daunting process for many vehicle owners. The lack of transparent pricing often leads to uncertainty, making it difficult for buyers to assess the true value of a car. Sellers, on the other hand, struggle to get fair prices without reliable market data. This gap in transparency creates mistrust between buyers and sellers, hindering smooth transactions.

Furthermore, car maintenance and repair services are often inconsistent, with vehicle owners facing challenges in finding trustworthy service providers. Many existing platforms do not provide comprehensive options for comparing prices, services, or quality of repairs, leading to dissatisfaction and a lack of confidence in the services rendered.

b) Key Requirements Identified:

User Authentication and Authorization:

- Secure user registration and login using Firebase Authentication.
- Role-based access control (user, admin) for restricted access to specific features.

Car Listing and Management:

- Allow users to list their used cars for sale with details like make, model, year, kilometers driven, and condition.
- Automated price estimation using Gemini AI API based on car details and condition.
- Admin verification of listed cars before they appear on the platform for sale.

Explore Cars:

- Provide a comprehensive list of all available cars for buyers to browse.
- Display detailed car information, including images stored on Cloudinary and the estimated market price.

Service Booking and Management:

- List verified service providers for different car services (e.g., maintenance, repairs).
- Allow users to book car services and view available service providers.

Admin Dashboard:

- Provide analytics for admins, including total cars listed, cars sold, income generated, and top-selling car makes.
- Features for user management, including deleting users, changing roles, and viewing transaction history.
- Tools to add, manage, and approve service providers and their services.

Image Upload and Storage:

- Utilize Cloudinary for storing car images uploaded by sellers and service providers.

Notifications and Alerts:

- Send notifications to users for car bookings, service bookings, and updates on car approval status.

2. Solution Overview

a) Solution Summary

The proposed solution is a **comprehensive used car marketplace** built using the MERN stack (MongoDB, Express, React, Node.js) with TypeScript. It addresses the challenges faced by vehicle owners in buying, selling, and servicing cars by providing a **transparent, user-friendly platform**. Key features include:

- **Secure Authentication:** Firebase Authentication ensures secure user login, with role-based access control for users and admins.

- **Fair Pricing:** Integration with Gemini AI API provides automated price estimation for used cars based on their condition, offering fair market value.
- **Admin Dashboard:** A dedicated admin interface enables car listing approvals, user management, and access to real-time analytics (e.g., total cars sold, income generated).
- **Explore and Book Cars:** Users can browse detailed car listings, compare prices, and book cars directly on the platform.
- **Service Booking:** The platform lists verified service providers, allowing users to book maintenance and repair services with transparent pricing and reviews.

The solution aims to provide **transparency, trust, and convenience**, creating a reliable platform for car owners to manage all their automotive needs in one place.

b) Objective

The primary objective of the platform is to create a **trusted and transparent marketplace** for buying, selling, and servicing vehicles. It aims to simplify the process for users by providing fair price estimates, verified service providers, and comprehensive car listings in one unified platform.

1. **Fair Pricing:** The integration of AI-driven price estimation ensures that sellers receive accurate market values for their cars, fostering trust.
2. **Streamlined Buying Experience:** Buyers can explore detailed car listings, compare options, and make informed purchase decisions easily.
3. **Simplified Selling Process:** Sellers can list their cars with minimal hassle, while the admin verification process ensures quality and trustworthiness in listings.
4. **Reliable Car Services:** Users can book maintenance and repair services from verified providers, making car care more accessible and transparent.
5. **Efficient Admin Management:** The admin dashboard provides insights and control, allowing efficient monitoring of sales, user activity, and service providers.

Overall, the solution aims to enhance user experience, increase trust in transactions, and streamline automotive services, benefiting both buyers and sellers in the used car market.

3. Features and Functionalities

a) Core Features

1. **User Authentication and Authorization:**
   - Secure user login and registration using Firebase Authentication.
   - Role-based access control (Admin, User, Service Provider).
2. **Car Selling and Listing:**
   - Sellers can easily post their used cars for sale with image uploads using Cloudinary.

- AI-based price estimation using Gemini AI API to suggest a fair market value for the car.
- Admin approval for listings before they are made visible to buyers, ensuring quality control.

3. **Car Buying and Booking:**
   - Comprehensive car listings with detailed specifications and images.
   - Buyers can view car details, compare prices, and book a car directly from the platform.
   - Option to mark cars as "Sold" after a successful transaction.
4. **Service Booking:**
   - Users can explore a list of verified service providers for car maintenance and repairs.
   - Booking services for various maintenance tasks (e.g., oil change, tire replacement).
   - Admin management for adding, removing, and updating service providers and their offerings.
5. **Admin Dashboard and Analytics:**
   - Detailed analysis of platform activity (number of cars sold, pending approvals, total revenue).
   - User management features (view users, delete users, update roles).
   - Insights on the most popular car makes, sales trends, and service usage statistics.
6. **Real-Time Notifications:**
   - Notifications for users on booking confirmations, approval status, and updates on listed cars.
   - Admin notifications for new car listings requiring approval.
7. **Secure Image Storage and Retrieval:**
   - Image handling using Cloudinary for efficient storage and fast retrieval of car images.
8. **Ex-Showroom Price Lookup:**
   - Users can view the current ex-showroom price of any vehicle to compare with the listed used car price.
9. **Comprehensive Car Details:**
   - Detailed information on each car, including make, model, year, kilometers driven, ownership details, and price.
10. **User-Friendly Design:**

- Responsive UI built with React and Bootstrap, ensuring a seamless experience across devices.

These features collectively create a robust and user-friendly platform for buying, selling, and servicing vehicles.

b) Additional Features

AI-Powered Price Estimation:

- Integration with the Gemini AI API to provide a dynamic market price estimate based on car condition and specifications, helping sellers get a fair price and increasing

transparency for buyers.

Integrated Search and Filters:

- Advanced search functionality allowing users to filter cars based on make, model, price range, year, kilometers driven, and ownership history.

Review and Rating System:

- Users can leave reviews and ratings for both cars and service providers, helping future buyers make informed decisions.

Push Notifications:

- Users receive push notifications for new listings, price updates, and service booking reminders.

Data Insights and Reporting for Admin:

- Detailed reporting tools for admins, including metrics on user engagement, sales performance, and service usage trends.

c) User Flows

Register/Login:

- Users sign up or log in using email/social accounts.
- User details are stored; they proceed to the dashboard.

Sell a Car:

- Seller fills in car details and uploads photos.
- Estimated price is shown using Gemini AI API.
- Listing is sent for admin approval before it appears on the site.

Buy a Car:

- User searches and filters through car listings.
- Views details and can save, book a test drive, or inquire about a car.

Book a Service:

- User selects a service type and chooses a provider.
- Books an appointment and receives confirmation.

Admin Dashboard:

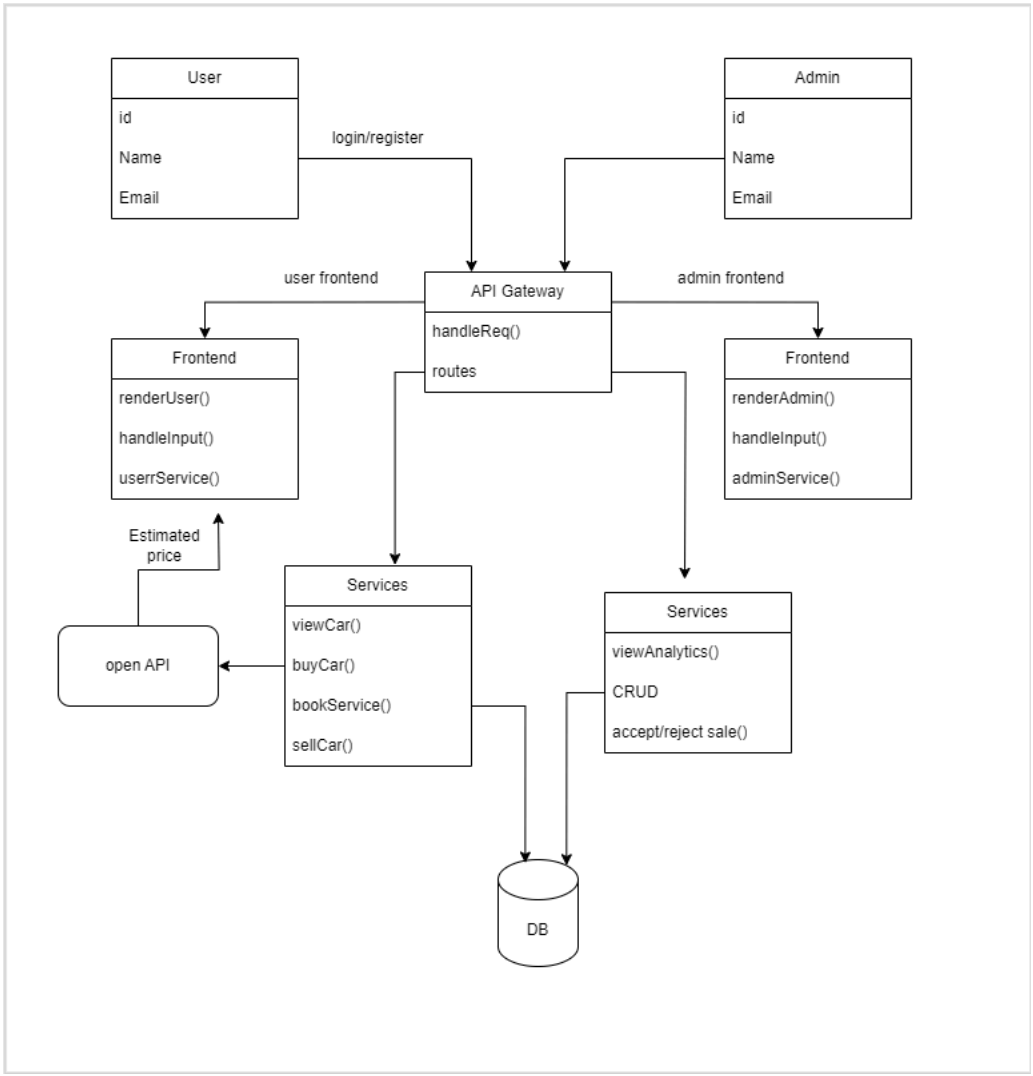
- Admin reviews and approves new listings.
- Manages users, car listings, and analytics.
- Adds service providers and updates services.

User Profile:

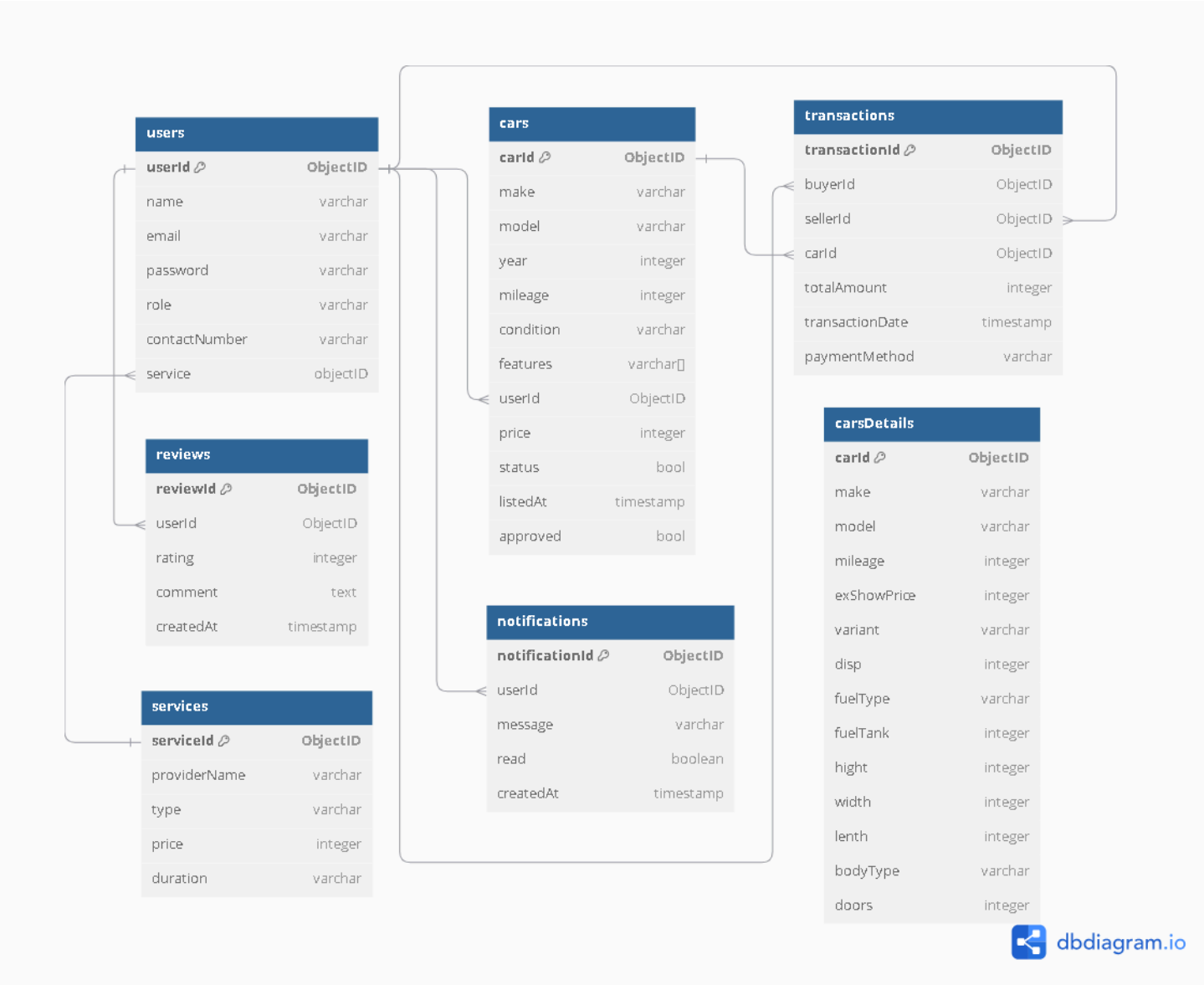
- Users manage their profile and view activity history.
- Tracks cars sold, bought, and services booked.
- Receives notifications for updates.

4.Architecture Diagram

a) System Architecture



Schema Diagram



User:

- **Role**: The user interacts with the platform to perform actions such as buying, selling, or servicing vehicles. The user can view available cars, submit their car listings, view services offered, etc.

Admin:

- **Role**: The admin is responsible for managing the platform. This includes tasks such as verifying car listings, managing users, handling service provider information, and approving transactions. The admin can also access the backend systems for oversight and control.

API Gateway:

- **Role**: The API Gateway serves as a central entry point for requests from the frontend (both for users and admins). It routes requests to the appropriate microservices (user service, car service, etc.). The API Gateway also handles load balancing, security (such as authentication), and possibly rate limiting.

Frontend (User and Admin):

- **Role**: The frontend for both users and admins provides a user interface that allows interaction with the platform.
  - **User Frontend**: Provides views for searching, browsing, and listing cars, as well as viewing services.
  - **Admin Frontend**: Provides views for managing users, approving or rejecting car listings, reviewing service providers, and performing other administrative tasks.

Services (User and Admin):

- **User Services**: These services handle user-specific actions, such as managing user profiles, fetching user-specific data, handling authentication, etc.
- **Admin Services**: These services manage administrative actions, such as approving car listings, managing service providers, reviewing car orders, and other admin-specific tasks.

Open API (Gemini):

- **Role**: The Open API (possibly named Gemini here) serves as an external or standardized interface for integrating third-party services. For instance, it could offer car price estimations, vehicle history reports, or integrate with external service providers.

DB (MongoDB):

- **Role**: MongoDB serves as the database for storing data related to users, cars, transactions, service providers, orders, and other platform entities. It is a NoSQL database used to store structured and unstructured data efficiently, providing quick access and scalability.

5. **Tech Stack**

a) Frontend

- **React** with **TypeScript** and **Context API**.
- **Bootstrap** and **Material UI** for UI

b) Backend:

- **Node.js** and **Express** for REST APIs.

c) Database

- **MongoDB** for document-based storage.
- **Mongoose** for schema modeling.

d) Other Technologies and Tools:

- **Gemini AI API** for market price estimation.
- **Cloudinary** for image storage.
- **Firebase** for authentication.

6 . Prerequisites and Requirements

a) Technical Requirements:

- **Hardware:** Development machines (8GB RAM, multi-core) and cloud server access for hosting and testing.
- **Software**: Node.js, React.js, MongoDB, Docker for development; GitHub for version control; Cloudinary for image storage.
- **Cloud Services:** Hosting on AWS/GCP, Firebase for authentication.

b) Data Requirements:

- **Sample Data:** User profiles, car listings, service provider details, and orders in JSON format.
- **External APIs:** Gemini Open API for car price estimations and history reports.
- **Test Data:** Mock data for unit testing, including user authentication tokens and sample car listings.

c) Access Permissions:

- **Version Control:** GitHub/GitLab access with proper roles (admin, write, read) for team members.
- **CI/CD:** Permissions for configuring pipelines in GitHub Actions or Jenkins for automated builds and deployments.
- **Cloud & API:** Access to Cloudinary for image storage, Firebase for authentication, and MongoDB Atlas for database management.

7. Future improvements

- Payment gateway integration for secure transactions.
- Real-time chat between buyers and sellers.
- Enhanced search filters using advanced algorithms.
- AI-based recommendation engine for cars and services.

8. Conclusion

a) Summary of Achievements:

- **Comprehensive Platform:** Built a vehicle platform with features for buying, selling, and servicing vehicles.
- **Scalable Architecture:** Implemented microservices for user management, car listings, and services, ensuring scalability.
- **Cloud Integration:** Integrated Cloudinary for image storage and Firebase for secure user authentication.

b) Value Provided:

- **Seamless User Experience:** Simplifies buying, selling, and managing cars with intuitive interfaces for users and admins.
- **Trust & Transparency:** Admin verification and external price estimations provide transparency and reliability.
- **Efficient Management:** Admins can efficiently manage users, listings, and services, ensuring platform security and trust.