**Car Dealership Management System**

**How to setup and run the Project**

1. Open terminal and go to frontend folder : **cd frontend**
2. now install all dependencies for frontend project: **npm install or npm i**
3. now run the frontend project : **npm run start**
4. now open beside new terminal and go to backend folder: **cd backend**
5. now install all dependencies for backend project : **npm install or npm i**
6. once all dependencies installed, backend server by running either: **node index.js or nodemon index.js (if nodemon is installed in the system)**

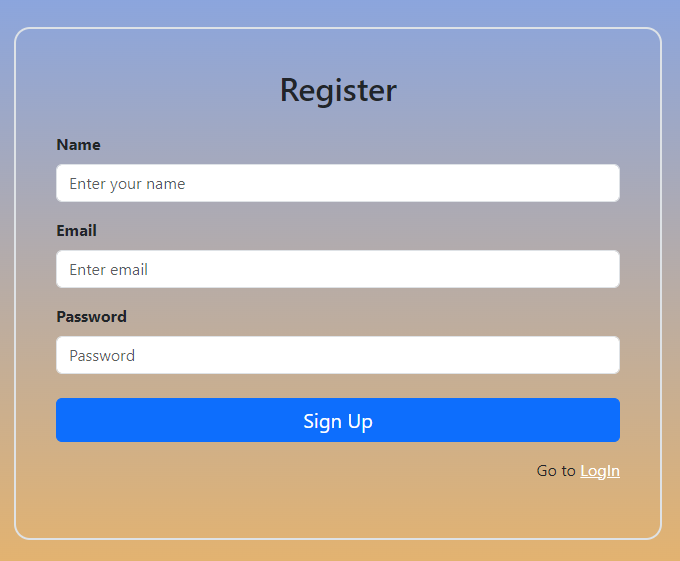
**All API Documentation**

1. **SignUp**
   1. Method: “POST”
   2. End point: <http://localhost:8000/register>
   3. Body: { "name": "dax", "email": "dax@gmail.com", "password" : "123" }
2. **Login**
   1. Method: “POST”
   2. End point: <http://localhost:8000/login>
   3. Body: { "email": "dax@gmail.com", "password" : "123" }

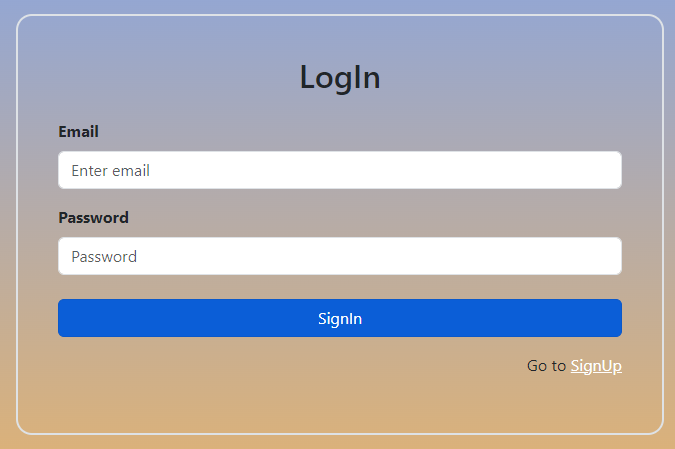
If you want to login as admin then use the below credentials:

* **Email:** [**admin@gmail.com**](mailto:admin@gmail.com)
* **Password: “admin@123”**

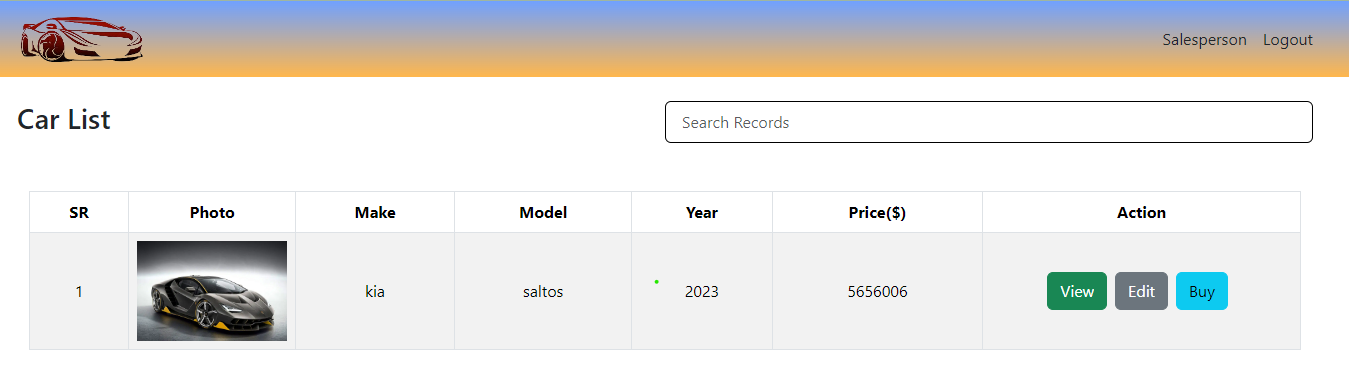
1. **Logout**
   1. Method: “GET”
   2. End point: <http://localhost:8000/logout>
2. **Get All Cars**
   1. Method: “GET”
   2. End point: <http://localhost:8000/car>
3. **Get particular Car**
   1. Method: “GET”
   2. End point: <http://localhost:8000/car/123>
4. **Create Car**
   1. Method: “POST”
   2. End point: <http://localhost:8000/car>
   3. Body: { "make": "Honda", "model": "CR-V", "year": 2016, "color": "white", "price": 200000, "mileage": 18, "fuel": "petrol", "transmission": "Automatic", "engineSize": 1600”, "vin": “s12f1sd2f14sa5da”, "file": Select file }
   4. Note: All data pass as a from-data in postman collection.
5. **Update Car**
   1. Method: “PUT”
   2. End point: <http://localhost:8000/car/123>
   3. Body: { "make": "Honda", "model": "CR-V", "year": 2016, "color": "white", "price": 200000, "mileage": 18, "fuel": "petrol", "transmission": "Automatic", "engineSize": 1600”, "vin": “s12f1sd2f14sa5da”, "file": Selected file }
   4. Note: All data pass as a from-data in postman collection.
6. **Delete Car**
   1. Method: “DELETE”
   2. End point: <http://localhost:8000/car/123>
7. **Car buy request to admin**
   1. Method: “POST”
   2. End point: http://localhost:8000/carStatus
   3. Body: Body: {"userId" : "dasd81487f8eaf1e04", "carId" : "a8dds24787f8eaf1easdad", "carStatus": "Pending" }
8. **Fetch car request**
   1. Method: “GET”
   2. End point :<http://localhost:8000/CarStatus>
9. **Approve request from admin side**
   1. Method: “PUT”
   2. End point: http://localhost:8000/carStatus/:id
   3. Body: Body: {"carStatus": "Sold" }

**1. Register salesperson in the Website**  
Salespersons are required to input their name, email, and password as mandatory fields to initiate the registration process.

Once registered, they can access the login page by clicking the "Login" link, which may appear below the signup button.

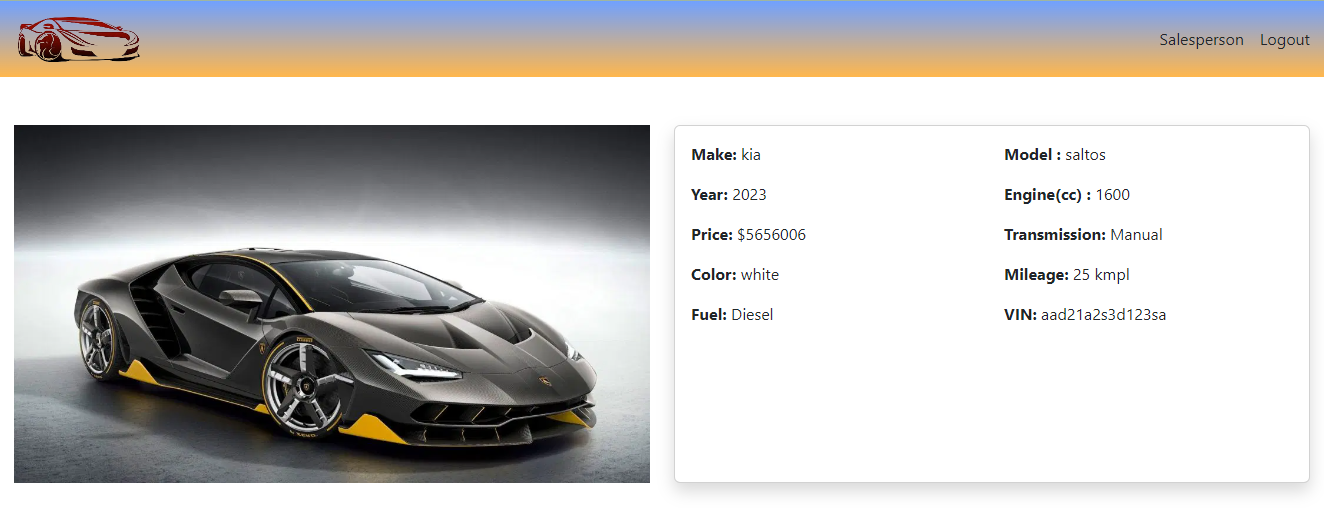
**2. Login for salesperson or admin**  
To login, the Salesperson or admin needs to input their registered email ID and password. If the correct credentials are provided, they'll be directed to the dashboard page; otherwise, an error message will appear.

**3. Dashboard page for salesperson**

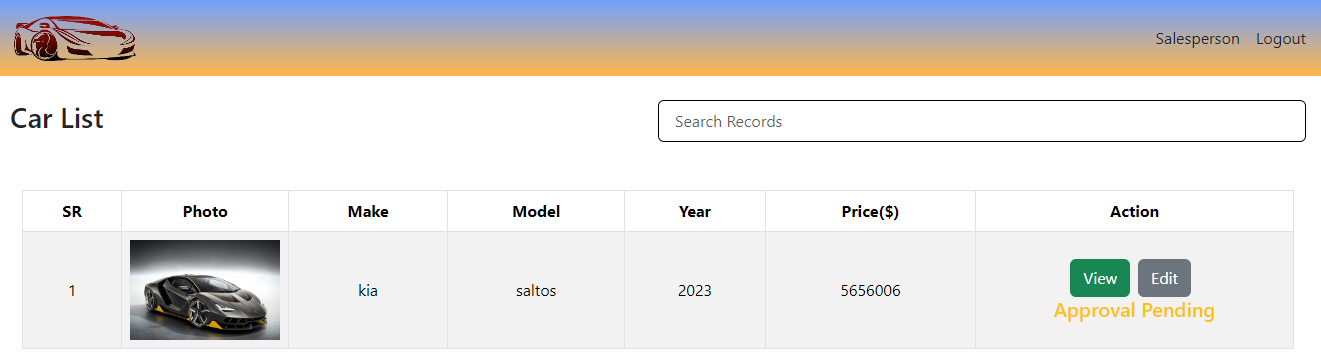


The dashboard design consists of a header area featuring a logo, login person and logout option, along with functionalities for searching records, complemented by a collection of cars showcased in table.

**4. Car details page**

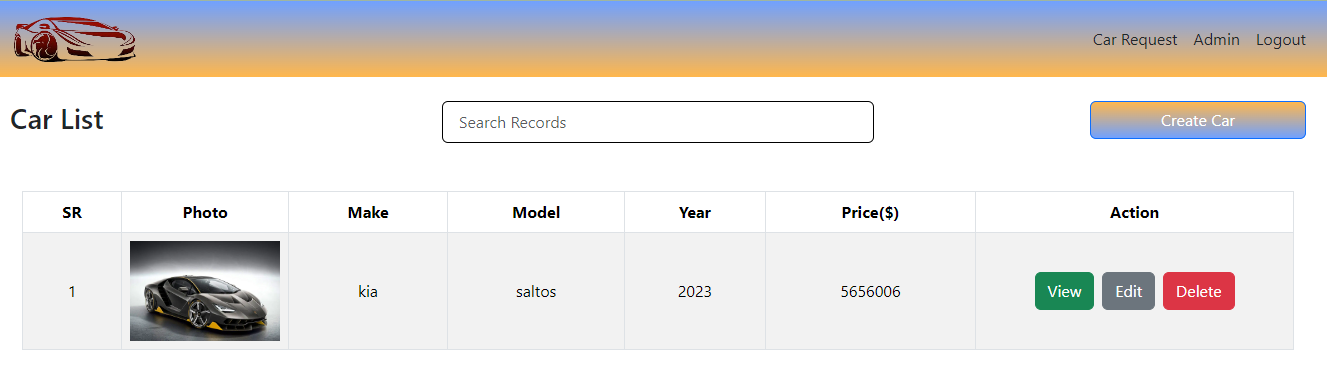


Clicking the view button redirects you to the car details page, view the image above. On this page, you'll find comprehensive information about the selected car, such as its image, model, make, year, price, fuel type, color, and more.

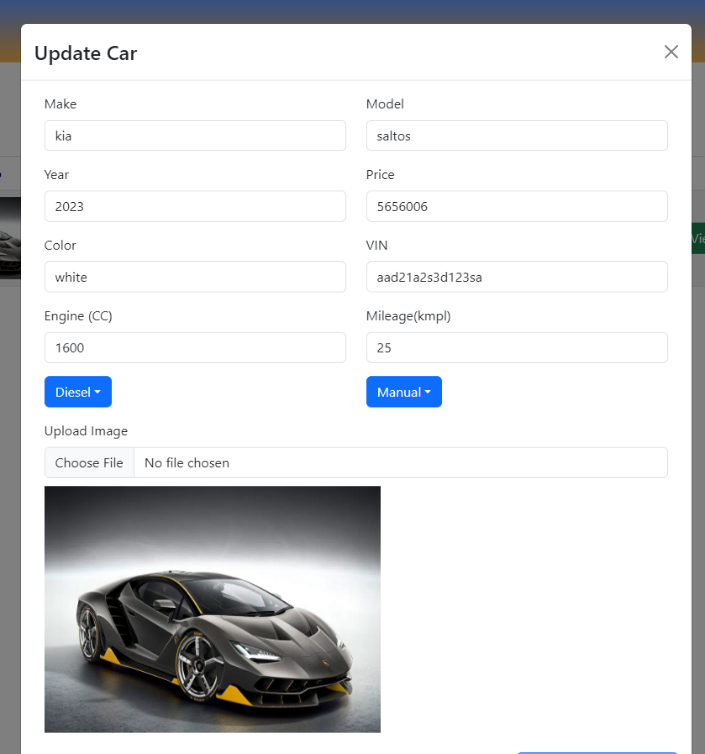
**5. Mark a car as pending sale**

Upon clicking the "Buy" button, a request is dispatched to the admin for car approval, during which the status reflects as "Approval Pending."   
  
Should the admin approve the request, the salesperson's dashboard will then indicate the "Sold" otherwise “Rejected” status.

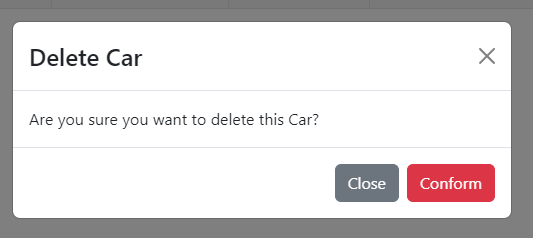
**6. Dashboard for admin**



The admin has full access to create, update, and delete cars through the dashboard provided above.



Clicking the "Create Car" button triggers the opening of a modal window, enabling you to input all required details and upload an image. Upon form completion, simply click "Submit" to generate a new car entry. Similarly, selecting the "Edit" button prompts the modal for editing car details. Upon implementing necessary changes, click "Submit" to update the car data accordingly.

Clicking the "Delete" button triggers a confirmation modal to appear. Upon confirming, the car will be deleted.

The administrator can perform CRUD operations, while salespeople have only read access.

1. **Approve Car Request from admin side**



When clicking on the car request, you will be redirected to a page where two buttons are displayed: one for approval and the other for rejection of the request.