**Client**

**SRC/App.js:**

import './App.css';

import React, {useEffect} from "react"

import Axios  from 'axios';

function App() {

  const [carName,setCarName]= React.useState("");

  const [carModel,setModel]=React.useState(0);

  const [carList,setCarList]=React.useState([]);

  const [newCarName,setNewCarName]= React.useState("");

  useEffect(()=>{

    Axios.get("http://localhost:3001/read").then(response =>{

      setCarList(response.data);

    })

  })

  const add\_c=()=>{

    Axios.post('http://localhost:3001/insert',{carName:carName,carModel:carModel})

  }

  const update\_c=(id)=>{

    Axios.put("http://localhost:3001/update",{id:id,newCarName:newCarName})

  }

  const delete\_c=(id)=>{

    Axios.delete(`http://localhost:3001/delete/${id}`)

  }

  return (

      <div className='App'>

        <h1>Car Management System </h1>

        <label >Car Name : </label>

        <input  type="text" placeholder='Enter Car Name' onChange={(event)=>{

          setCarName(event.target.value);

        }} />

        <label> Car Model : </label>

        <input type="number" placeholder='Enter Car Model' onChange={(event)=>{

          setModel(event.target.value);

        }}  />

        <br></br>

        <button className="btn btnAdd"  onClick={add\_c}>ADD</button>

        <hr/>

        {

          carList.map((val, key)=>{

            return <div key={key}>

              <h1>{val.carName}</h1>

              <h1>{val.carModel}</h1>

              <input type="text" placeholder='Enter Name to Update or Remove Car'  onChange={(event)=>{

          setNewCarName(event.target.value);

        }} />

        <br></br>

              <button className="btn btnUpdate" onClick={()=>update\_c(val.\_id)}>Update</button>

              <button className="btn btnRemove" onClick={()=>delete\_c(val.\_id)}>Remove</button>

            </div>

          })

        }

        </div>

    );

}

export default App;

**SRC/App.css:**

body {

  font-family: 'Poppins';font-size: 22px;

  flex-direction: column;

}

.App {

  text-align: center;

  flex-direction: column;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: row;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

.btn {

  background-color: #4CAF50; /\* Green \*/

  border: none;

  color: white;

  padding: 16px 32px;

  text-align: center;

  text-decoration: none;

  display: inline-block;

  font-size: 16px;

  margin: 4px 2px;

  transition-duration: 0.4s;

  cursor: pointer;

}

.btnRemove {

  background-color: white;

  color: black;

  border: 2px solid #f44336;

}

.btnRemove:hover {

  background-color: #f44336;

  color: white;

}

.btnUpdate {

  background-color: white;

  color: black;

  border: 2px solid #008CBA;

}

.btnUpdate:hover {

  background-color: #008CBA;

  color: white;

}

.btnAdd {

  background-color: white;

  color: black;

  border: 2px solid #4CAF50;

}

.btnAdd:hover {

  background-color: #4CAF50;

  color: white;

}

input {

  padding: 12px 20px;

  margin: 8px 0;

  box-sizing: border-box;

  border: 3px solid #ccc;

  -webkit-transition: 0.5s;

  transition: 0.5s;

  outline: none;

}

input:focus {

  border: 3px solid #555;

}

**Server**

**Models/Car.js:**

const mongoose= require('mongoose');

const CarSchema= new mongoose.Schema({

    carName:{

        type:String,

        required: true,

    },

    carModel:{

        type:Number,

        required:true,

    }

});

const Car = mongoose.model("Car",CarSchema)

module.exports=Car;

**Index.js:**

const express = require('express');

const mongoose= require('mongoose');

const cors= require('cors');

const app= express()

const CarModel= require("./models/Car");

app.use(express.json());

app.use(cors());

mongoose.connect('mongodb+srv://haris:haris123@cluster0.krlfb.mongodb.net/?retryWrites=true&w=majority',{

    useNewURLParser:true,

});

app.post("/insert",async (req, res)=> {

    const carName=req.body.carName;

    const carModel=req.body.carModel;

    const car= new CarModel({ carName:carName, carModel:carModel});

    try{

        await car.save();

        res.send("inserted data");

    }catch(err){

        console.log(err);

    }

})

app.get("/read",async (req, res)=> {

    CarModel.find({},(err,result)=>{

        if(err){

            res.send(err);

         }

         res.send(result);

    })

})

app.put("/update",async (req, res)=> {

    const newCarName=req.body.newCarName;

    const id=req.body.id;

    try{

        await CarModel.findById(id,(err,updatedCar)=>{

            updatedCar.carName= newCarName;

            updatedCar.save();

            res.send("update");

        })

    }catch(err){

        console.log(err);

    }

})

app.delete("/delete/:id",async (req, res)=> {

   const id= req.params.id;

   await CarModel.findByIdAndRemove(id).exec();

   res.send("deleted");

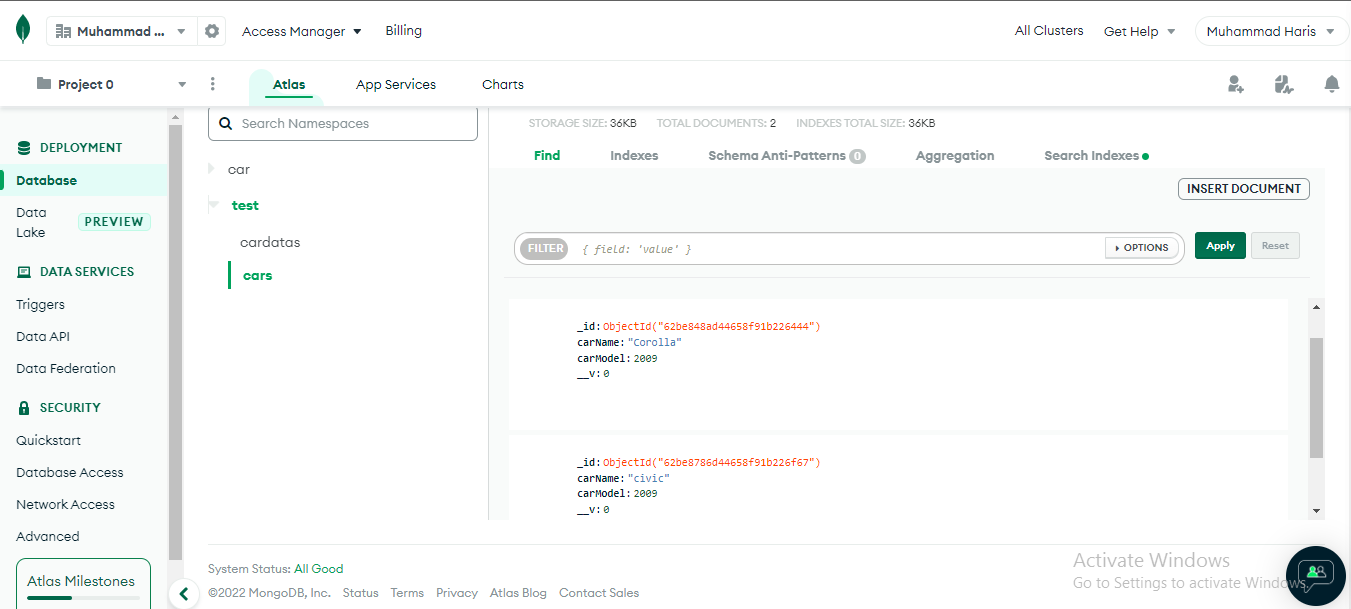
})

app.listen(3001, () => {

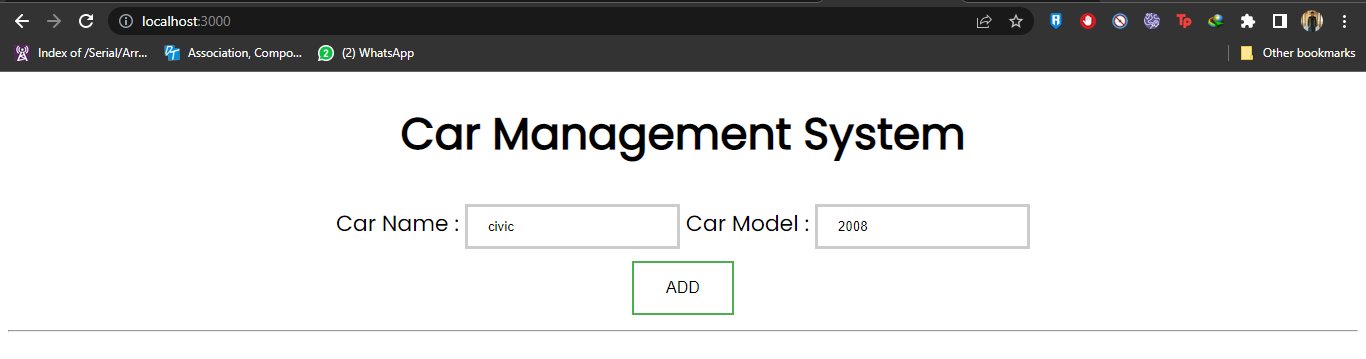
    console.log("Server running");

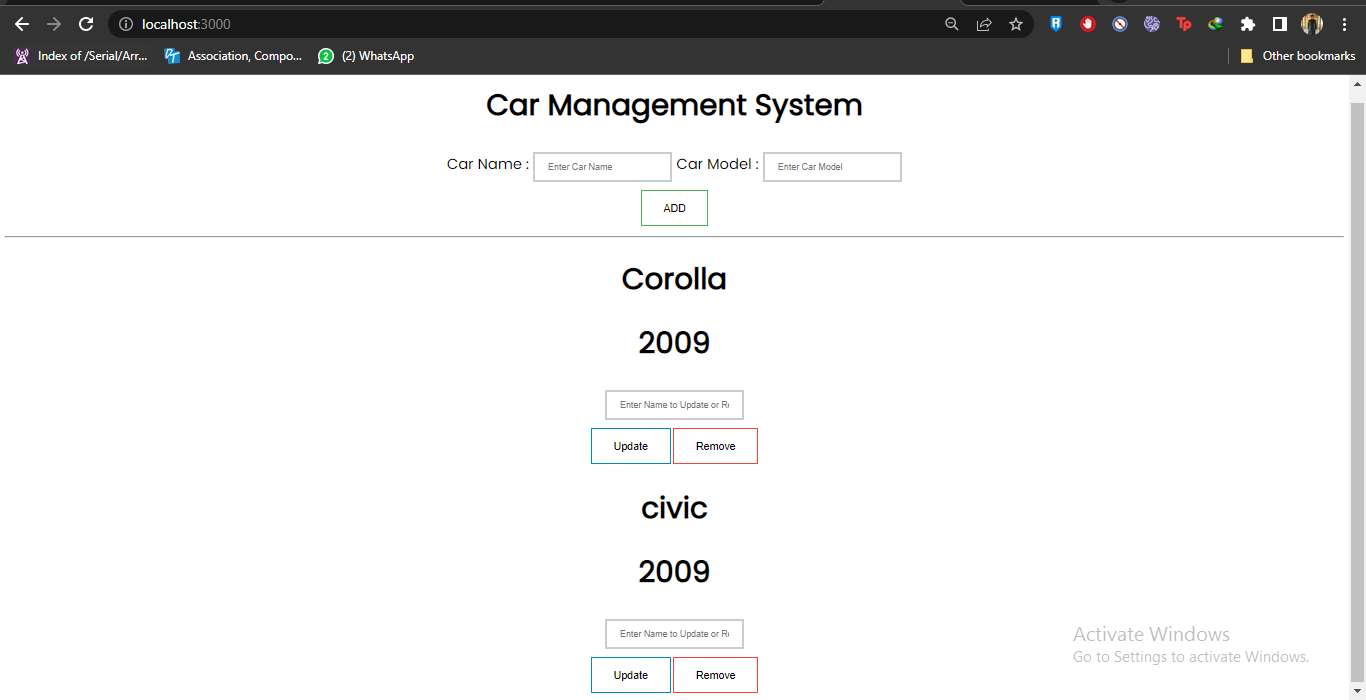
})

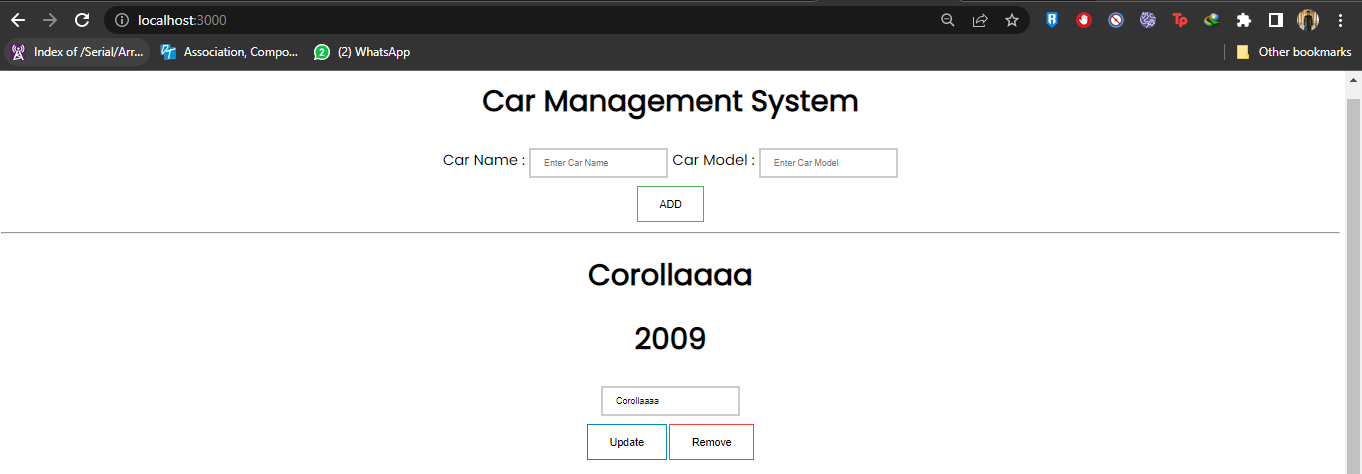
**Mongo dB Atlas**

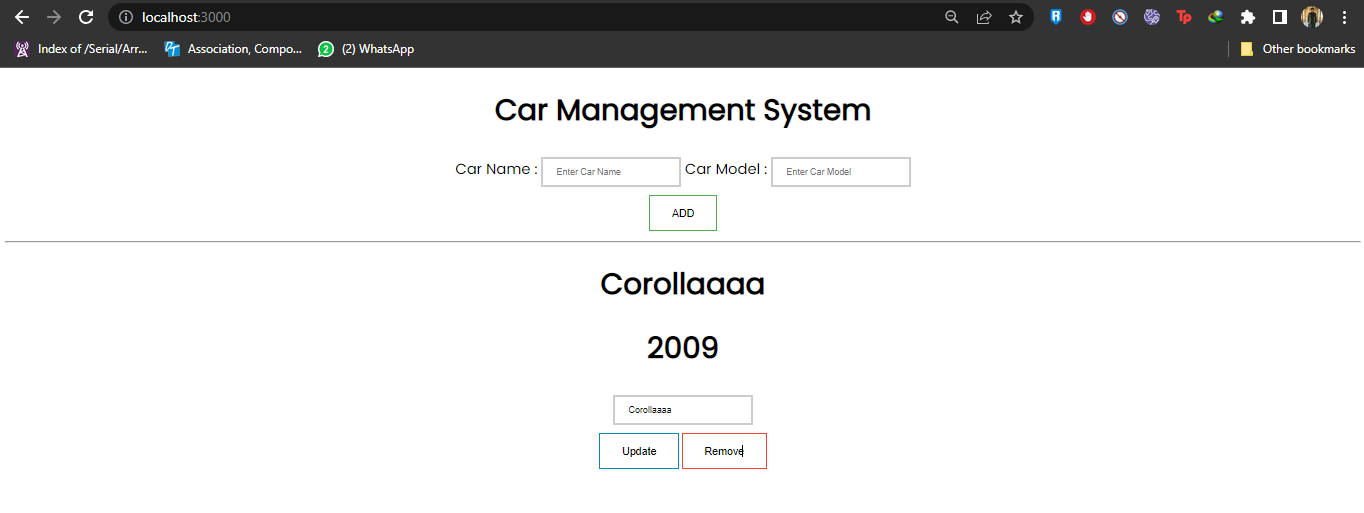


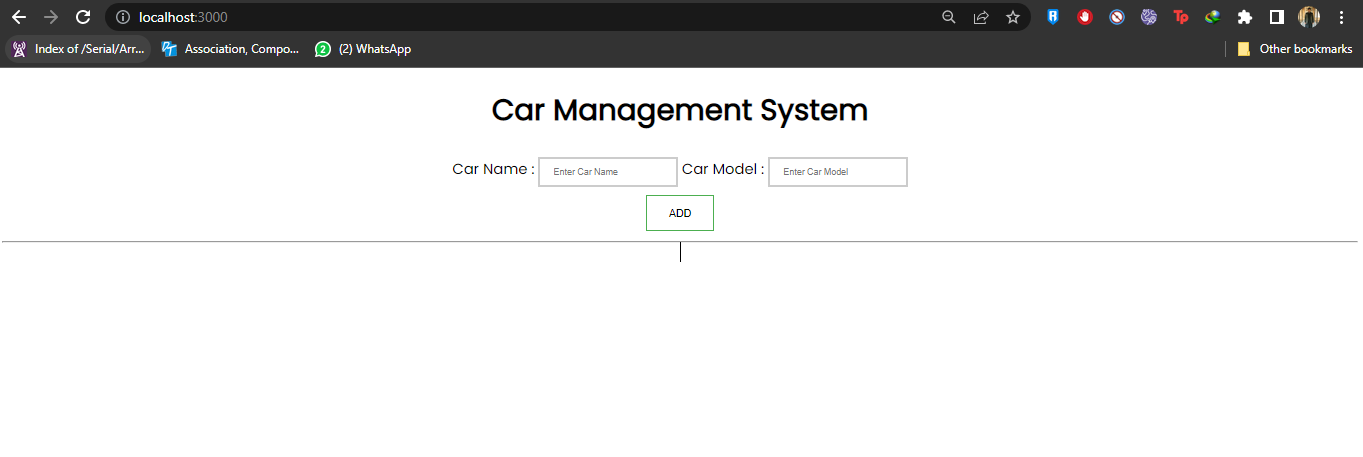
**Outputs**











**Github Link**

**https://github.com/haris-7523/web-final-oel**