Full Stack MERN Application with AI Integration and AWS Deployment

Mini Project 1

Module: MongoDB Basics

Project Title: Car Listings Database

Project Overview
Create a MongoDB database to store car details (make, model, year, price, images). Students will practice creating collections, inserting documents, and running queries.**Objective**
To set up and manage a database for storing car listings. Students will practice:

- Designing a MongoDB schema for cars
- CRUD operations on the database
- Querying with filters (e.g., price range, year)
- Indexing for faster searches

Scope of the Project

- Create `cars` collection with fields: make, model, year, price, seller info, image URLs
- Insert sample car documents
- Retrieve cars by filters
- Update and delete records
- Create indexes on frequently searched fields (e.g., make, price)

Technical Requirements

- Language: JavaScript (Node.js for database scripts)
- Database: MongoDB
- Tools: MongoDB Compass, VS Code, Node.js CLI

Mini Project 2

Module: Express.js Basics

Project Title: Car Listings API

Project Overview
Build a REST API using Express.js to interact with the car database. Implement routes for adding, updating, deleting, and fetching car details.

Objective
To create an API layer between the database and frontend. Students will practice:

- RESTful route design
- Middleware for logging and authentication
- Error handling
- Integration with MongoDB

Scope of the Project

- Routes: `GET /cars`, `GET /cars/:id`, `POST /cars`, `PUT /cars/:id`, `DELETE /cars/:id`
- Middleware for request logging
- Basic token-based authentication for adding/updating cars
- Error handling for invalid data

Technical Requirements

- Language: JavaScript
- Framework: Express.js
- Database: MongoDB
- Tools: Postman, VS Code

Mini Project 3

Module: React Basics

Project Title: Car Listing Frontend

Project Overview
Develop a React-based interface to view available cars. Users can browse car listings, search by filters, and view details of each car.

Objective
To build a clean, responsive frontend that fetches data from the backend API. Students will practice:

- Component-based architecture
- State and props
- Fetching API data
- Event handling

Scope of the Project

- Display car listings in a grid/list view
- Search/filter cars by make, price, and year
- View individual car details
- Responsive design for mobile and desktop

Technical Requirements

- Language: JavaScript
- Framework: React.js
- Tools: VS Code, Browser DevTools

Mini Project 4

Module: File Upload Handling

Project Title: Car Image Upload Service

Project Overview
Implement backend functionality to upload and store car images, either locally or on a cloud service (e.g., Cloudinary).

Objective
To enable image uploads for car listings. Students will practice:

- Handling multipart form data
- Integrating with file storage (local or cloud)
- Linking image URLs to car records in the database

Scope of the Project

- API endpoint to upload images
- Store image URLs in MongoDB alongside car data
- Validate image formats and sizes

Technical Requirements

- Language: JavaScript (Node.js)
- Framework: Express.js
- Tools: Multer (for file uploads), Cloudinary SDK (optional), Postman

Mini Project 5

Module: Full MERN Integration

Project Title: Car Selling Platform

Project Overview
Combine all the mini projects into one complete MERN application where users can sign up, list cars for sale, upload images, and browse cars.

Objective
To integrate all modules into a working platform. Students will practice:

- Full-stack MERN development
- User authentication and authorization
- Connecting React frontend to backend APIs
- Handling form submissions and file uploads

Scope of the Project

- User authentication (JWT)
- Car listing creation with images
- Browsing, searching, and viewing cars
- Editing and deleting listings by the owner

Technical Requirements

- Language: JavaScript
- Stack: MERN (MongoDB, Express.js, React.js, Node.js)
- Tools: VS Code, MongoDB Atlas/Compass, Postman

Mini Project 6

Module: AI Chatbot Integration

Project Title: AI-Powered Car Sales Support Chatbot

Project Overview
Integrate an AI chatbot into the car selling platform to answer customer queries, provide details about car listings, and guide users through the buying/selling process — 24/7. The chatbot will be trained or connected to relevant data such as FAQs, car details, and policies.

Objective
To implement a conversational AI assistant that can:

- Answer general questions about the platform
- Provide details about specific cars from the database
- Assist sellers with listing their cars
- Offer buyers guidance on searches and purchases

Students will learn:

- Using AI APIs (e.g., OpenAI GPT API)
- Integrating chatbot UI in React
- Connecting chatbot to backend data
- Handling real-time messaging

Scope of the Project

- Chatbot interface embedded in React frontend (floating widget)
- Backend API endpoint to handle chatbot queries
- Integration with AI API for language understanding and responses
- Contextual responses by querying MongoDB for car details
- Fallback to FAQ database for unanswered queries

Technical Requirements

- Language: JavaScript
- Stack: MERN (MongoDB, Express.js, React.js, Node.js)
- AI API: OpenAI GPT (or similar LLM API)

Tools:

- React Chatbot UI library (e.g., react-chatbot-kit, Botpress, custom chat widget)
- Postman for testing
- API keys for AI provider
- Concepts Used: API integration, NLP basics, real-time communication

Mini Project 7

Module: Version Control with Git & GitHub

Project Title: Car Platform Source Code Management

Project Overview
Set up Git for the project and push code to GitHub. Students will learn the complete workflow for managing source code in teams, tracking changes, and collaborating.

Objective
To introduce students to Git and GitHub for version control, ensuring:

- Code history tracking
- Collaboration through branches and pull requests
- Handling merge conflicts
- Maintaining a professional project repository

Scope of the Project

- Initialize a local Git repository
- Create `.gitignore` to exclude environment files and dependencies
- Commit and push to GitHub repository
- Create branches for features (e.g., `feature/car-upload`)
- Merge changes via pull requests
- Use GitHub Issues for task tracking

Technical Requirements

- Tools: Git, GitHub, VS Code
- Concepts Used: Git basics, branching, merging, pull requests, issue tracking

Mini Project 8

Module: Cloud Deployment

Project Title: Deploying Car Selling Platform to the Cloud

Project Overview
Deploy the complete MERN + AI Chatbot application to a cloud platform, making it accessible to users worldwide. Students will configure environment variables, connect to a live MongoDB database, and ensure production readiness.

Objective
To host the application on a cloud platform so users can:

- Access the frontend React app via a public URL
- Use the backend API live
- Interact with the AI chatbot in real-time

Students will practice:

- Building the frontend for production
- Deploying backend with cloud hosting
- Connecting to MongoDB Atlas
- Managing secrets and environment variables

Scope of the Project

- Deploy React frontend (Vercel, Netlify, or similar)
- Deploy Node.js + Express backend (Render, Railway, or AWS EC2)
- Connect backend to MongoDB Atlas
- Configure API keys for AI chatbot securely
- Test live application with Postman & browser

Technical Requirements

- Language: JavaScript (MERN stack)
- Tools:
  1. GitHub (for repo source)
  2. Vercel/Netlify (frontend)
  3. AWS (backend)/Render/Railway
  4. MongoDB Atlas (database)
  5. AI API provider (OpenAI or similar)+
- Concepts Used: CI/CD basics, environment configuration, cloud hosting

Key learnings

Now, your **Car Upload & Selling Website** full learning path looks like this:

1. **MongoDB** – Car Listings Database
2. **Express.js** – Car Listings API
3. **React.js** – Car Listing Frontend
4. **File Upload** – Car Image Upload Service
5. **Full MERN Integration** – Complete Platform
6. **AI Chatbot Integration** – Smart 24/7 Car Sales Assistant
7. **Git & GitHub** – Version Control & Collaboration
8. **Cloud Deployment** – Live, Production-Ready Application