**GARAGE BOOKING AND MANAGEMENT SYSTEM**

**Concept:**

A website for a garage that allows clients to login and book an appointment, receive appointment alerts, receives car progress reports when prompted and gets car completion notifications.

**Basic features:**

* A homepage that showcases company details, a container on the banner prompting the user to book an appointment with them and an e-commerse button suggesting a page selling spareparts
* A client registration page where the user inputs their name, phone no., email address, car plate, car type and askes for them to choose a date and time. Once registered, the site sends a notification to the users email and phone number confirming the booking.
* A Car-progress page where the user is taken once registration is complete. Here the user has a refresh button that requests for a progress update. This sennds a notification to the admin where after manual update on the admin site, sends a notification to the client with a link to the progress page showcasing the updated message.
* An admin page that logs in the specific admin using their name and email address, monitors and manages all booking inquiries, views all bookings and cars in repair, is responsible for updating all prompted progress inquiries and once published, sends a notification to the specific client on the updated car progress. The page is also responsible for sending car completion notifications to the specific client.

garage-booking-system/

│

├── client/ # React frontend

│ ├── public/

│ ├── src/

│ │ ├── assets/ # Images, icons, logos

│ │ ├── components/ # Reusable UI components (buttons, navbar, etc.)

│ │ ├── pages/ # Individual page components

│ │ │ ├── HomePage.jsx

│ │ │ ├── BookingPage.jsx

│ │ │ ├── ClientDashboard.jsx

│ │ │ ├── AdminDashboard.jsx

│ │ │ └── LoginPage.jsx # Admin login

│ │ ├── context/ # Context API (for global user/session state)

│ │ ├── hooks/ # Custom hooks (useAuth, useBookings, etc.)

│ │ ├── services/ # Axios API functions

│ │ │ ├── api.js

│ │ │ └── bookingService.js

│ │ ├── styles/ # Global CSS or Tailwind config

│ │ ├── App.jsx

│ │ └── main.jsx

│ ├── .env

│ └── package.json

│

├── server/ # Node.js + Express backend

│ ├── src/

│ │ ├── config/ # DB connection & environment setup

│ │ │ ├── db.js

│ │ │ └── dotenv.js

│ │ ├── models/ # Sequelize models (User, Booking, Car)

│ │ │ ├── User.js

│ │ │ ├── Booking.js

│ │ │ └── Car.js

│ │ ├── routes/ # Express routes (auth, booking, admin)

│ │ │ ├── authRoutes.js

│ │ │ ├── bookingRoutes.js

│ │ │ └── adminRoutes.js

│ │ ├── controllers/ # Route logic handlers

│ │ │ ├── authController.js

│ │ │ ├── bookingController.js

│ │ │ └── adminController.js

│ │ ├── middleware/ # JWT verification, error handling

│ │ │ ├── authMiddleware.js

│ │ │ └── errorHandler.js

│ │ ├── utils/ # Helpers (email, sms, etc.)

│ │ │ ├── emailService.js

│ │ │ ├── smsService.js

│ │ │ └── progressHelper.js

│ │ ├── index.js # Main Express entry

│ │ └── server.js # Server startup

│ ├── .env

│ └── package.json

│

├── docker-compose.yml # Optional for containerized deployment

├── README.md

└── .gitignore