

Vehicle Service Management System - Setup Instructions

1. Clone the Repository

```
git clone https://github.com/abahadur29/vehicle-service-management-system.git  
cd vehicle-service-management-system
```

2. Database Setup

1. Install SQL Server (if not already installed)

2. **Update Connection String:** Open

`backend/VehicleServiceManagement/VehicleServiceManagement.API/appsettings.json`

3. Update the `DefaultConnection` string with your SQL Server instance:

```
{  
  "ConnectionStrings": {  
    "DefaultConnection":  
      "Server=YOUR_SERVER_NAME\\SQLEXPRESS;Database=VehicleServiceDB;Trusted_Connection=True;MultipleActiveResultSets=true;TrustServerCertificate=True"  
  }  
}
```

Replace `YOUR_SERVER_NAME` with your SQL Server instance name (e.g., `localhost` or your computer name).

For LocalDB (Default - if using Visual Studio):

```
{  
  "ConnectionStrings": {  
    "DefaultConnection": "Server=  
      (localdb)\\MSSQLLocalDB;Database=VehicleServiceDB;Trusted_Connection=True;Multiple  
      ActiveResultSets=true;TrustServerCertificate=True"  
  }  
}
```

3. Backend Setup

1. Navigate to the backend directory:

```
cd backend/VehicleServiceManagement/VehicleServiceManagement.API
```

2. Restore NuGet packages:

```
dotnet restore
```

3. Apply database migrations:

```
dotnet ef database update
```

If you encounter issues with Entity Framework tools, install them first:

```
dotnet tool install --global dotnet-ef
```

4. (Optional) Verify the database was created in SQL Server Management Studio

4. Frontend Setup

1. Navigate to the frontend directory:

```
cd frontend/vehicle-service-management-ui
```

2. Install Angular CLI globally (if not already installed):

```
npm install -g @angular/cli
```

3. Install project dependencies:

```
npm install
```

How to Run

Running the Backend

1. Navigate to the backend project directory:

```
cd backend/VehicleServiceManagement/VehicleServiceManagement.API
```

2. Run the application:

```
dotnet run
```

Or use Visual Studio:

- Open the solution file
`backend/VehicleServiceManagement/VehicleServiceManagement.sln`
- Press F5 or click the "Run" button

3. The API will start on:

- **HTTP:** `http://localhost:5293`
- **HTTPS:** `https://localhost:7033`

4. Access Swagger UI:

- Open your browser and navigate to: `https://localhost:7033/swagger`
- This provides interactive API documentation

Running the Frontend

1. Navigate to the frontend directory:

```
cd frontend/vehicle-service-management-ui
```

2. Start the Angular development server:

```
ng serve
```

Or:

```
npm start
```

3. Access the application:

- Open your browser and navigate to: `http://localhost:4200`
- The frontend is configured to communicate with the backend API running on
`https://localhost:7033`

Running Both Together

Option 1: Separate Terminals

- Open two terminal windows/command prompts
- Run backend in one terminal

- Run frontend in another terminal

Option 2: PowerShell (Windows)

```
# Terminal 1
cd "backend\VehicleServiceManagement\VehicleServiceManagement.API"
dotnet run

# Terminal 2
cd "frontend\vehicle-service-management-ui"
ng serve
```

Default Test Users (Development)

The system includes seed data for development. After running the application and migrations, the following default users are automatically created:

Role	Email	Password	Full Name	Permissions
Admin	admin@vehicle.com	Admin@123	System Administrator	Full system access - Manage users, categories, and inventory
Service Manager	manager@vehicle.com	Manager@123	Ramesh Kumar	Assign service tasks, monitor technician workload, approve services, and view reports & dashboards
Technician	tech@vehicle.com	Tech@123	Rajiv Pratap	View assigned tasks, update service status, complete services
Customer	customer@vehicle.com	Customer@123	Rahul Verma	Book services, view service history, manage vehicles, and make payments

Additional Test Users:

- Service Managers:** `sunita.manager@vehicle.com`, `amit.manager@vehicle.com` (Password: `Manager@123`)
- Technicians:** `priya.tech@vehicle.com`, `anita.tech@vehicle.com`, `vikram.tech@vehicle.com`, `kavita.tech@vehicle.com`, `arjun.tech@vehicle.com` (Password: `Tech@123`)
- Customers:** `sneha.customer@vehicle.com`, `mohit.customer@vehicle.com`, and 8 more (Password: `Customer@123`)

Note: All default users are automatically seeded when the application runs in Development mode. The primary customer user (`customer@vehicle.com`) has sample vehicles and service history for testing purposes.