

# **Project Description**

## **Requirement Gathering and Analysis:**

### **1. Project Overview**

We are making a small web application to help manage vehicles and drivers. It will track when vehicles and drivers enter or leave the yard by scanning QR codes. The system will also save information about vehicles, drivers, and show basic reports. It will make sure that only the right people can see and edit the data.

### **2. Goals and Objectives**

- Build a website to manage vehicles and drivers easily.
- Track check-in and check-out times using QR codes.
- Track vehicle movement in and out of the company gate using QR scanning.
- Show simple reports for how vehicles and drivers are being used.

### **3. Scope of Work**

#### **Vehicle Management:**

- Add, edit, and delete vehicle details (like plate number, model, etc.).
- Create a unique QR code for each vehicle.
- Save extra details (like maintenance and insurance records).

#### **Driver Management:**

- Add, edit, and delete driver details (like name, contact, license number).
- Link drivers to vehicles.

#### **Time Tracking (QR Code Based):**

- Allow scanning of QR codes through a phone or computer camera.
- Save the time when a vehicle or driver checks in and out.
- Keep track of which driver and which vehicle.

#### **Yard Entry/Exit Tracking:**

- Scan QR codes at the gate to track vehicle entry and exit.
- Record time and driver/vehicle information.

## Reporting:

- Create simple reports like:
  - How much a vehicle was used (hours, trips, mileage, fuel).
  - How much a driver drove.
  - Time in and out records.
  - Yard entry/exit logs.
- Create a dashboard showing important numbers.

## 4. Front end:

We are building it's front end in react.

## 4. Back end:

The back end of this project is being built in node.

## 6. Database:

PostgreSQL is used as a database.

## Database schema:

Below is the relational schema of each entity,

Driver: (driver\_id PK , role\_id FK, name, contact, license number)

Vehicle: (vehicle\_id PK, license plate, make, model, year, maintenance records, insurance expiry)

DriverVehicleAssignment: (assignment\_id PK, driver\_id FK, vehicle\_id FK, assigned\_date)

TimeLog: (timelog\_id PK, driver\_id FK, vehicle\_id FK, check\_in\_time, check\_out\_time)

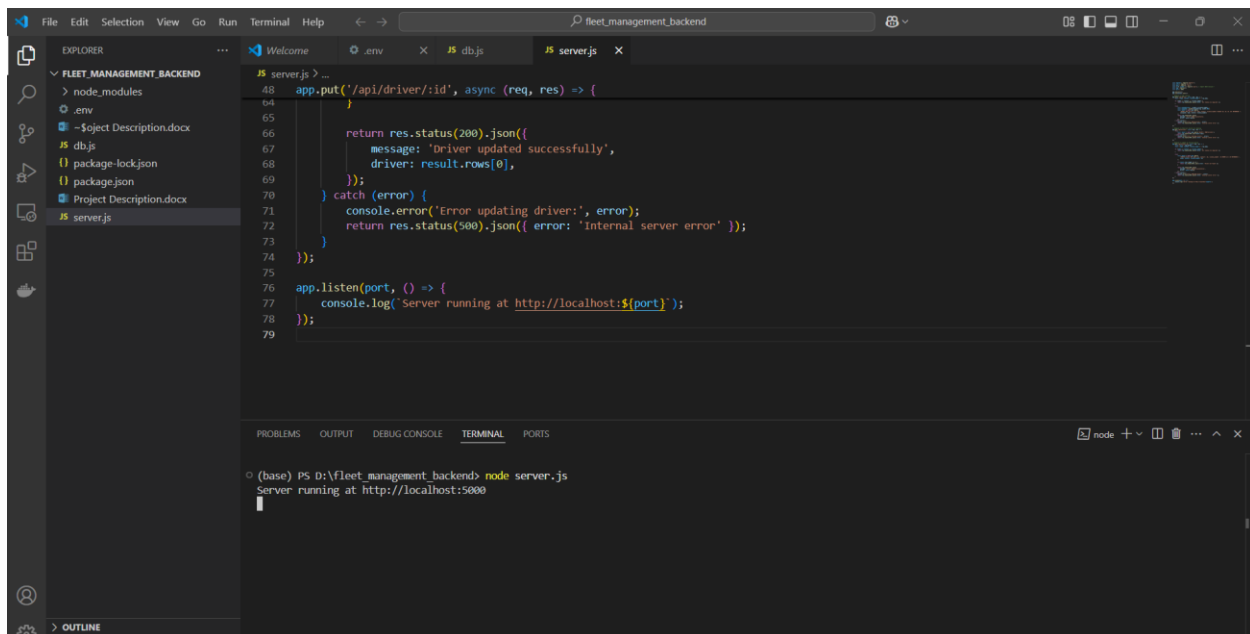
YardLog: (yardlog\_id PK, vehicle\_id FK, user\_id FK, entry\_time, exit\_time)

User: (user\_id PK, username, password\_hash, role\_id FK, email, contact\_number)

Role : (role\_id PK, role\_name)

## Server Setup:

The initial server is already setup with initial front end.



```
server.js
48 app.put('/api/driver/:id', async (req, res) => {
49   // ...
65   // ...
66   return res.status(200).json({
67     message: 'Driver updated successfully',
68     driver: result.rows[0],
69   });
70 } catch (error) {
71   console.error('Error updating driver:', error);
72   return res.status(500).json({ error: 'Internal server error' });
73 }
74 });
75
76 app.listen(port, () => {
77   console.log(`Server running at http://localhost:${port}`);
78 });
79
```

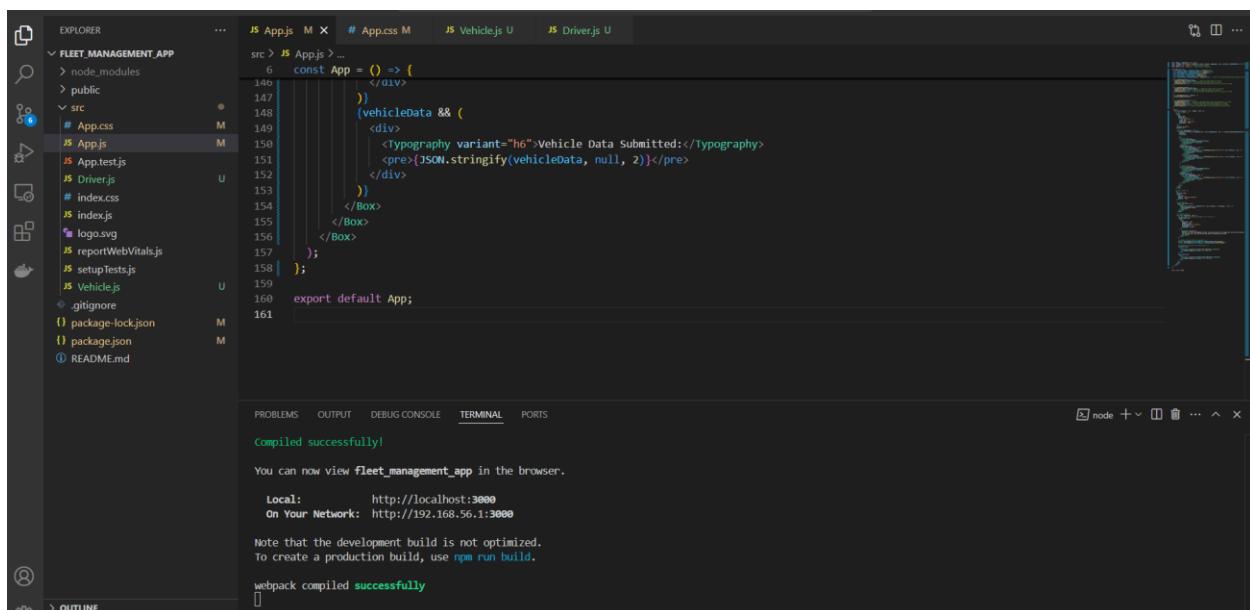
```
(base) PS D:\fleet_management_backend> node server.js
Server running at http://localhost:5000
```

## Initial API endpoint:

An initial api endpoint is built which is send http post request from front end to back end.

## Initial Front-end setup:

The initial front-end setup is also built, and initial UI component are also built.



```
App.js
6 const App = () => {
140   // ...
147   // ...
148   (vehicleData && (
149     <div>
150       <Typography variant="h6">Vehicle Data Submitted:</Typography>
151       <pre>{JSON.stringify(vehicleData, null, 2)}</pre>
152     </div>
153   ))
154   </Box>
155 </Box>
156 </Box>
157 );
158 };
159
160 export default App;
161
```

```
Compiled successfully!

You can now view fleet_management_app in the browser.

Local:    http://localhost:3000
On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

## Basic UI:

Below are some screenshots of basic UI,

Dashboard

Driver

Add

Edit

Vehicle

Fleet Management System

Add Driver

Name

Contact

License Number

ADD

Dashboard

Driver

Add

Edit

Vehicle

Fleet Management System

Edit Driver

Select Driver

Maria - M0987654321

Joshua - D1234567890

Dashboard

Driver

Add

Edit

Vehicle

Fleet Management System

Edit Driver

Select Driver

Maria - M0987654321

Field to Edit

Name

Contact

License Number

Dashboard

Driver

Add

Edit

Vehicle

Fleet Management System

Edit Driver

Select Driver

Maria - M0987654321

Field to Edit

Contact

New Value

UPDATE

Dashboard

Driver

Vehicle

Add

Fleet Management System

Add Vehicle

Model

License Plate

Type

ADD

The designs and theme of the app is not finalized yet so it will be changing as per the requirements and as the project goes forward.