# **Vehicle Maintenance Log Management System**

## Introduction

The Vehicle Maintenance Log Management System is designed to help companies manage their fleet of vehicles, track maintenance activities, schedule service appointments, manage driver information, and handle payments. This system uses PHP Laravel for the backend, Axios for making HTTP requests, JavaScript for frontend functionality, and MySQL for database management.

## Table of Contents

1. [Introduction](#introduction)
2. [Technology Stack](#technology-stack)
3. [Installation](#installation)
4. [Configuration](#configuration)
5. [Usage](#usage)
   * [User Registration and Authentication](#user-registration-and-authentication)
   * [Vehicle Management](#vehicle-management)
   * [Driver Management](#driver-management)
   * [Maintenance Logs](#maintenance-logs)
   * [Service Scheduling](#service-scheduling)
   * [Notifications](#notifications)
   * [Fuel Consumption Tracking](#fuel-consumption-tracking)
6. [Role-Based Access Control](#role-based-access-control)
7. [Security](#security)
8. [Performance and Scalability](#performance-and-scalability)
9. [Usability](#usability)
10. [Reliability and Maintainability](#reliability-and-maintainability)
11. [Contributing](#contributing)
12. [License](#license)

## Technology Stack

* **Backend:** PHP Laravel
* **Frontend:** HTML, CSS, Bootstrap, JavaScript
* **HTTP Requests:** Axios
* **Database:** MySQL

## Installation

1. **Clone the repository:**

bash

Copy code

git clone https://github.com/yourusername/vehicle-maintenance-log.git

cd vehicle-maintenance-log

1. **Install dependencies:**

bash

Copy code

composer install

npm install

1. **Set up the environment file:**

bash

Copy code

cp .env.example .env

1. **Generate application key:**

bash

Copy code

php artisan key:generate

1. **Set up the database:**
   * Update your .env file with your database credentials.
   * Run the migrations:

bash

Copy code

php artisan migrate

1. **Run the development server:**

bash

Copy code

php artisan serve

npm run dev

## Configuration

Ensure the .env file contains the necessary configuration for database and mail services. Example:

env

Copy code

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=vehicle\_maintenance

DB\_USERNAME=root

DB\_PASSWORD=password

MAIL\_MAILER=smtp

MAIL\_HOST=smtp.mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

MAIL\_FROM\_ADDRESS=hello@example.com

MAIL\_FROM\_NAME="${APP\_NAME}"

## Usage

### User Registration and Authentication

* **User Registration:**
  + Access the registration page and fill in the required details.
  + A confirmation email will be sent upon successful registration.
* **User Login:**
  + Access the login page and enter your credentials to log in.

### Vehicle Management

* **Add Vehicle:**
  + Navigate to the vehicle management page and submit the vehicle details.
* **View/Edit/Delete Vehicle:**
  + View the list of vehicles and perform edit or delete actions as needed.

### Driver Management

* **Add Driver:**
  + Navigate to the driver management page and submit the driver details.
* **View/Edit/Delete Driver:**
  + View the list of drivers and perform edit or delete actions as needed.
* **Assign Driver to Vehicle:**
  + Select a vehicle and assign a driver.

### Maintenance Logs

* **Add Maintenance Log:**
  + Navigate to the maintenance logs page and submit the log details.
* **View/Edit/Delete Maintenance Log:**
  + View the list of maintenance logs and perform edit or delete actions as needed.

### Service Scheduling

* **Schedule Service:**
  + Select a vehicle, schedule a service, and receive confirmation and reminder notifications.

### Notifications

* Users receive notifications for account activities, service reminders, and expirations.

### Fuel Consumption Tracking

* **Add Fuel Record:**
  + Submit fuel details for a vehicle.
* **View/Edit/Delete Fuel Record:**
  + View the list of fuel records and perform edit or delete actions as needed.

## Role-Based Access Control

* Define and manage different roles (e.g., Super Admin, Admin, Support Staff) with specific permissions.

## Security

* Secure authentication and authorization mechanisms.
* Data encryption for sensitive information.
* Regular security audits and updates.

## Performance and Scalability

* Efficient database queries.
* Fast load times for pages and actions.
* Scalability to handle increasing numbers of users and data.

## Usability

* Intuitive user interface.
* Responsive design for mobile and desktop users.
* Clear navigation and helpful prompts.

## Reliability and Maintainability

* Regular backups of the database.
* Failover mechanisms to ensure high availability.
* Clean and modular codebase.
* Detailed documentation for developers.
* Automated tests for key functionalities.

## Contributing

* Fork the repository.
* Create a new branch (git checkout -b feature-branch).
* Make your changes and commit them (git commit -m 'Add new feature').
* Push to the branch (git push origin feature-branch).
* Open a Pull Request.