

Real-Time Vehicle Tracking System

Testing Screenshots (Photos) – Detailed Explanation

1. Introduction

Testing ensures that the Real-Time Vehicle Tracking System works correctly, reliably, and securely. This document explains different testing stages along with descriptions of testing screenshots (photos) captured during execution.

2. Testing Environment

- Frontend: Web application (Chrome browser)
- Backend: REST API server
- Database: MySQL / MongoDB
- GPS Simulation: Test GPS data / simulator
- Network: Internet connection

3. Login Page Testing Screenshot

Screenshot Description: Login page with username and password fields.

Test Case: Valid user login.

Expected Result: User should be redirected to dashboard.

Actual Result: Login successful and dashboard displayed.

Status: Pass

4. Dashboard Screen Testing Screenshot

Screenshot Description: Dashboard showing total vehicles, active vehicles, alerts panel, and live map preview.

Test Case: Verify dashboard data loading.

Expected Result: All widgets should load with correct values.

Actual Result: Vehicle count and alerts displayed correctly.

Status: Pass

5. Live Vehicle Tracking Testing Screenshot

Screenshot Description: Map displaying live vehicle marker with location details.

Test Case: Real-time GPS data update.

Expected Result: Vehicle marker should move as location updates.

Actual Result: Marker updated in real time on map.

Status: Pass

6. Vehicle Management Testing Screenshot

Screenshot Description: Add/Edit vehicle form screen.

Test Case: Add new vehicle details.

Expected Result: Vehicle should be added to database and listed.

Actual Result: Vehicle added successfully and visible in list.
Status: Pass

7. Reports & History Testing Screenshot

Screenshot Description: Trip history and route path displayed.
Test Case: View historical data for selected vehicle.
Expected Result: Correct route and trip data shown.
Actual Result: Route history loaded accurately.
Status: Pass

8. Alerts & Notifications Testing Screenshot

Screenshot Description: Alert popup for over-speed condition.
Test Case: Generate over-speed alert.
Expected Result: Alert should be displayed immediately.
Actual Result: Alert notification received successfully.
Status: Pass

9. API Testing Screenshot (Postman)

Screenshot Description: API response showing JSON location data.
Test Case: Fetch latest vehicle location API.
Expected Result: API returns valid JSON with coordinates.
Actual Result: Correct JSON response received.
Status: Pass

10. Testing Summary

Test Module	Status
Login Module	Pass
Dashboard	Pass
Live Tracking	Pass
Vehicle Management	Pass
Reports	Pass
Alerts	Pass
API Integration	Pass

11. Conclusion

All major modules of the Real-Time Vehicle Tracking System were tested successfully. The system meets functional requirements and performs reliably under testing conditions.