**Project:** *CarCare – AI-Powered Vehicle Maintenance Web App*
**Team:** CarCare
**Version:** 1.0
**Date:** December 2025

## 1. Introduction

### 1.1 Purpose

This document describes the features, requirements, and expected behavior of the **CarCare Web Application**.
It helps developers, designers, testers, and faculty understand **what the product is**, **who it serves**, and **how it works**.

### 1.2 Product Overview

CarCare is a web application that makes **vehicle maintenance simple and stress-free**.
It helps users get:

- On-time service reminders
- Nearby garage suggestions
- AI-based maintenance predictions
- Expense tracking
- A clean dashboard showing car health

The app's onboarding flow (already live on your website) shows a **clean, easy-to-use experience** for new users.

### 1.3 Target Users

CarCare is designed for:

1. **Everyday Car Owners** – Students, professionals, families
2. **Tech-Savvy Users** – People who want AI insights from car data
3. **Business Owners / Fleet Managers** – Who manage multiple vehicles
4. **Mechanics & Garages** – Who want more customers and better visibility

## 2. Problem Statement

Most car owners forget service dates, lose track of expenses, and struggle to find reliable garages.
Mechanics have difficulty reaching customers and managing service records.

This leads to:

- Missed maintenance
- Unplanned breakdowns
- High repair costs
- Poor communication

CarCare solves this by creating **one unified digital platform** for vehicles, garages, reminders, and insights.

## 3. Product Goals

CarCare aims to:

- Remind users about upcoming service needs
- Help them find trustworthy garages easily

- Reduce maintenance costs through better planning
- Track all service history digitally
- Provide usable AI suggestions for better car care
- Support mechanics with job listings and visibility

**4. Functional Requirements**

**4.1 User Accounts**

- Users can sign up using email/phone.
- Users can log in securely.
- Onboarding collects basic details and guides the user smoothly.

**4.2 Vehicle Management**

- Users can add one or more vehicles.
- Users can enter vehicle details (model, year, mileage, fuel type).
- Users can update their mileage anytime.
- The system stores all service history for each vehicle.

**4.3 Maintenance Reminders**

- CarCare automatically calculates when the next service is due.
- The app sends reminders based on mileage or time.
- Upcoming service reminders appear clearly on the dashboard.

**4.4 Garage Locator**

- Shows nearby garages based on the user's location.
- Displays garage details (service type, distance, cost range).
- Users can contact or request a service.

**4.5 AI-Based Insights (Growing Feature)**

- The system gives simple insights like predicted maintenance.
- It alerts users when a part *may* need attention.
- It uses past service data and mileage trends to provide suggestions.

**4.6 Expense Tracking**

- Users can enter service costs or upload bills.
- CarCare displays total monthly/yearly spending.

**4.7 Notifications**

- Reminder alerts
- Service booking updates
- Important AI insights
- Expense summary notifications

**4.8 Mechanic/Garage Features**

- Mechanics can receive service requests.
- They can update service status.
- They can view relevant customer details (vehicle, problem).

**4.9 Admin Features**

- Manage users and mechanics
- Manage garages
- View platform-wide activity

**5. Non-Functional Requirements (How the system should behave)**

**5.1 Usability**

- Simple, beginner-friendly onboarding
- Clean UI with easy navigation
- Works well on mobile and desktop

**5.2 Performance**

- Pages should load within ~3 seconds
- Dashboard must show real-time updates

**5.3 Security**

- Data transmission must be encrypted (HTTPS)
- Passwords must be securely stored
- User data must not be shared with third parties

**5.4 Reliability**

- System should work consistently without crashes
- Data should never be lost
- Backups must be maintained

**5.5 Scalability**

- Should support thousands of users as the application grows

**6. User Stories**

**Car Owner**

- *I want reminders so I don't miss vehicle services.*
- *I want to add my car details easily.*
- *I want to find nearby garages in emergencies.*

**Tech User**

- *I want AI insights that explain car issues in simple words.*

**Fleet Manager**

- *I want to manage several vehicles in one dashboard.*

**Mechanic**

- *I want more customers through online visibility.*
- *I want booking requests I can accept or reject easily.*

**7. Acceptance Criteria**

CarCare will be considered successful when:

- Users can sign up and finish onboarding without confusion
- Vehicles can be added and managed smoothly
- Reminders appear at the right time
- Garage listing works based on location
- Expense tracking is accurate
- AI insights are understandable and helpful
- Mechanics are able to view and respond to service requests

**8. System Overview**

CarCare uses:

- **Firebase Authentication** for secure login
- **Firestore Database** to store vehicles, services, users, and expenses
- **Firebase Hosting** for hosting the web app
- **Cloud Functions** for processing reminders and insights

This setup ensures speed, reliability, and scalability.

**9. Constraints**

- Requires stable internet
- Location accuracy depends on device permissions
- AI predictions depend on the amount of available data

**10. Future Enhancements**

- Integrating real OBD (On-Board Diagnostics) devices
- Full service booking and payment gateway
- Voice-based assistant
- Predictive cost forecasting
- Insurance integration