Project Management Plan (PMP)

# 1. Scope

Project Name: Car Purchasing Web Application  
  
Objective: To build a platform that allows sellers to list and market their cars, and buyers to search, view, and reserve cars based on various criteria.  
  
In Scope:  
- User registration and login.  
- Car listing page showing available cars.  
- Search functionality with filters.  
- Reservation system for available cars.  
- Admin panel with special access and features.  
- Web-based system accessible via PC.  
  
Out of Scope:  
- Mobile app version.  
- Online payment gateway integration.  
- AI-based car recommendation system.

# 2. Roles & Responsibilities (R&R)

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Ebrahem Mostafa | Project Manager | Plan, monitor, and control the project. Facilitate communication, manage risks, and ensure timely delivery. |
| Omar Abdou | Developer | Implement backend and frontend features. Ensure code quality and integration. |
| Arsany Monier | Developer | Work on frontend/backend tasks. Collaborate with tester and designer for functionality. |
| Abdelaziz | Tester | Prepare test cases, perform testing, report bugs, and ensure the product meets requirements. |
| Anas | Tester | Support test planning and execution. Perform regression and sprint testing. |
| Ehab | Designer | Create UI/UX designs, wireframes, and ensure visual consistency of the website. |

# 3. Configuration Management (CM)

Version Control System: Git  
Repository Platform: GitHub  
Repository Link: https://github.com/ebrahem1812/Car-Purchasing  
  
Branching Strategy:  
- Main Branch: Stable version, reflects the latest approved version.  
- Development Branch: Ongoing work, feature development, and updates.  
  
Configuration Items Tracked:  
- Source code  
- Design files  
- PMP documents  
- Requirement specs  
- Testing artifacts  
- Review documents  
  
Change Tracking:  
- Changes are tracked using commits with clear messages.  
- Team members contribute via pull requests or direct commits (based on internal agreements).  
- History of changes is maintained and reviewed via Git logs.

# 4. Phases (Sprints Breakdown)

|  |  |  |
| --- | --- | --- |
| Sprint | Week | Focus |
| Sprint 1 | Week 1 | Initial planning: SIQ, CRS, PMP |
| Sprint 2 | Week 2 | SRS and project timeline definition |
| Sprint 3 | Week 3 | UI/UX Design: Wireframes, mockups, and user flow creation |
| Sprint 4 | Week 4 | Frontend & Backend development (Phase 1): User registration, login, car listing |
| Sprint 5 | Week 5 | Development (Phase 2): Search functionality, reservation system, admin features |
| Sprint 6 | Week 6 | Testing: Functional, usability, and regression testing. Bug fixing. Final delivery |

# 5. Change Requests (CR)

Change Requests Process:  
- All change requests must be submitted through a formal CR Form or via documented email/meeting notes.  
- The Project Manager (Ebrahem Mostafa) reviews the request and assesses its impact on scope, cost, and schedule.  
- Based on the analysis, the change is either approved or rejected.  
- Approved changes are added to the backlog for future sprints.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Change Request | Impact | Status |
| 1 | Add a contact form to allow users to message the admin | Adds 1–2 days of dev & testing time | Under Review |
| 2 | Include an image gallery for each car listing | Requires redesign + backend update (~3 days) | Approved (Sprint 5) |
| 3 | Change reservation system from “hold” to “instant confirmation” | Major backend logic update | Rejected (out of scope) |
| 4 | Add “Sort by price” feature on car search | Small frontend change, ~1 day | Approved (Sprint 5) |

# 6. Risks

|  |  |  |
| --- | --- | --- |
| Risk | Impact | Mitigation Strategy |
| Unclear or changing customer requirements | Rework and delays | Use SIQ/CRS documents, get approvals early, handle via CR process |
| Team member unavailability | Slowed progress in sprints | Assign backup members when possible, have clear documentation |
| Integration issues between frontend and backend | Technical delays and extra testing needed | Early and continuous integration, code reviews, daily sync-ups |
| Poor communication between team members | Misunderstandings and rework | Use tools like Teams and GitHub, hold regular standups |
| Underestimation of tasks | Sprint delays | Break tasks down, use sprint retrospectives to improve estimation |
| Customer feedback late in the project | May require major changes at the end | Frequent reviews with customer, demo progress regularly |

# 7. Reviews

Internal Team Reviews:  
- Every 2 days: Team members meet on Discord for short sync meetings to review progress, align tasks, and raise blockers.  
- Daily follow-ups via WhatsApp group to ensure continuous communication and quick updates.  
  
Sprint Review Meetings:  
- End of every sprint: A formal review meeting will be held to:  
 - Demonstrate sprint deliverables  
 - Get team feedback  
 - Prepare for the next sprint  
  
Customer Reviews:  
- Sprint review outcomes will be shared with the customer for feedback.  
- All feedback will be documented and considered for the next sprint or raised as CRs.

# 8. Communication Plan (Meetings & Tools)

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Tool / Platform | Frequency | Purpose |
| Daily Follow-Up | WhatsApp | Daily | Quick status updates, blockers, coordination |
| Team Sync Meeting | Discord | Every 2 days | Discuss progress, align tasks, clarify blockers |
| Sprint Review Meeting | Discord / In-person | End of each sprint | Review deliverables, gather feedback, plan next sprint |
| Customer Communication | Email / Discord | Weekly (or as needed) | Share progress, review documents, collect feedback |
| Documentation Sharing | Google Drive / GitHub | As needed | Share and collaborate on project documents |