REQUIREMENTS

of Business, User, Developer

Ahad Moeen and Muhammad Mubashir

moenahad@gmail.com

Table of Contents

1. Business Requirements

1.1 Objective

PakWheels aims to create a robust and user-friendly online marketplace for automobiles, offering users the ability to buy and sell vehicles and engage in automotive discussions. The platform should also provide a space for users to interact with one another, share knowledge, and create a community around automobile culture in Pakistan.

1.2 Target Audience

- **Buyers:** Individuals looking to purchase new or used cars, motorcycles, and automotive parts.

- **Sellers:** Individuals or dealers listing vehicles for sale.

- **Automobile Enthusiasts:** Users participating in discussions or seeking automotive-related information.

1.3 Key Business Goals

- Provide an easy-to-use platform for buying and selling vehicles.

- Increase user engagement through features like reviews, forums, and vehicle ratings.

- Enable users to easily search and filter listings by various criteria (make, model, price, etc.).

- Generate revenue through advertisements, premium listings, and subscription services for sellers.

- Ensure the platform remains scalable to handle increasing numbers of listings and users.

1.4 Success Metrics

- Increase in registered users and active buyers.

- High volume of vehicle listings from sellers.

- Increased interaction and forum participation.

- Positive feedback and user reviews.

2. User Requirements

2.1 Functional Requirements

- **Account Creation:** Users must be able to register and create accounts using email or social media.

- **Vehicle Listing:** Sellers must be able to post vehicles with detailed information, photos, and price.

- **Search and Filters:** Buyers should be able to search for vehicles based on specific criteria (make, model, year, price, etc.).

- **Discussion Forums:** Users should be able to participate in automotive discussions and ask questions.

- **User Profile:** Users should be able to create and manage personal profiles with their listings and preferences.

2.2 Non-Functional Requirements

- **Usability:** The system should be easy to navigate and understand for both novice and experienced users.

- **Performance:** The platform should load quickly, with minimal downtime.

- **Security:** User data and transaction details must be secure with encryption and protection against unauthorized access.

2.3 User Interface (UI) Requirements

- The platform must have a clean and intuitive interface that adapts to different screen sizes (desktop, mobile).

- The vehicle listing and search features should be straightforward, with clear call-to-action buttons.

- Discussion forums and user interaction areas should be easy to access and participate in.

---

3. Software Developer Requirements

3.1 Technical Requirements

- Web Development: The system will be developed using modern web technologies such as HTML5, CSS3, and JavaScript frameworks (React.js, Vue.js, etc.).

- **Backend:** The backend will be built using a scalable framework such as Node.js or Python (Django/Flask).

- **Database:** A relational database (e.g., PostgreSQL or MySQL) to store user information, vehicle listings, and forum data.

- **APIs:** Integration with third-party APIs for payment gateways, location-based services, and social media authentication.

- **Hosting:** The system will be hosted on cloud platforms such as AWS or Azure, ensuring scalability and reliability.

3.2 Security Requirements

- **User Authentication:** Implement role-based access control (RBAC) for different types of users (buyers, sellers, admins).

- **Data Encryption:** Sensitive user data such as passwords and payment information should be encrypted using secure algorithms (e.g., AES, RSA).

- **Secure Payment Gateway:** Integration with trusted payment systems (e.g., PayPal, Stripe) to process transactions securely.

- **Session Management:** Implement secure session management practices to prevent unauthorized access.

3.3 Performance Requirements

- **Response Time:** The platform should load within 3 seconds for most pages, and transaction-related actions should be processed within 5 seconds.

- **Scalability:** The platform should be able to handle a large number of concurrent users and listings without significant performance degradation.

- **High Availability:** The platform must be available 99.9% of the time, excluding scheduled maintenance periods.

3.4 Maintainability

- The codebase should follow best practices in software engineering, with clear documentation for future developers.

- Modular code and reusable components should be prioritized for easy maintenance and updates.

- Automated testing and continuous integration (CI) pipelines should be implemented for faster development cycles and quality assurance.