**Quality Assurance Plan for "Artisans & Craftsmen Marketplace"**

**1. Introduction**

* **1.1 Purpose:** This document outlines the Quality Assurance (QA) plan for the "Artisans & Craftsmen Marketplace" web platform. The plan defines the QA objectives, methodologies, and procedures to ensure the delivery of a high-quality and user-friendly platform.
* **1.2 Scope:** This plan covers all aspects of the platform development, including:
  + **Functionality:** Ensuring all features work as intended (e.g., user registration, product browsing, order placement, payment processing).
  + **Usability:** Ensuring the platform is easy to navigate and use for both artisans/craftsmen and customers.
  + **Performance:** Ensuring the platform performs efficiently under expected load.
  + **Security:** Ensuring the platform is secure from cyber threats (e.g., data breaches, hacking).
  + **Accessibility:** Ensuring the platform is accessible to users with disabilities.
  + **Compatibility:** Ensuring the platform is compatible with different browsers and devices.
* **1.3 Project Overview:**
  + The "Artisans & Craftsmen Marketplace" is a web platform that connects artisans and craftsmen with interested customers.
  + Key features:
    - User registration and profiles
    - Product catalog with search and filtering options
    - Shopping cart and checkout process
    - Order management and tracking
    - Secure payment gateway integration
    - Communication tools (e.g., messaging, reviews)
    - Marketing and promotional features

**2. Quality Objectives**

* Deliver a bug-free web platform with all core functionalities working as intended.
* Ensure an intuitive and responsive design for all users, including artisans, customers, and administrators.
* Achieve compatibility across major browsers and devices.
* Ensure the security of user data and payment transactions.
* Deliver a high-quality product that meets user expectations and business requirements.

#### ****3. Standards****

* **Coding Standards**: Ensure the development aligns with clean code practices.
* **Security Standards**: Follow OWASP guidelines to secure user data and payment information.
* **Accessibility Standards**: Implement WCAG 2.1 for inclusive design.
* **Browser Standards**: Maintain compliance with W3C web standards.

**4. Testing Types**

* **Unit Testing:** Individual components (modules, functions) will be tested to ensure they function correctly.
* **Integration Testing:** Tested to ensure that different components work together as expected.
* **System Testing:** The entire system will be tested to verify that it meets the specified requirements.
* **User Acceptance Testing (UAT):** Real users will test the platform to provide feedback and ensure it meets their needs.
* **Regression Testing:** After any code changes, regression tests will be performed to ensure that existing functionality still works correctly.
* **Performance Testing:** Load and stress tests will be conducted to evaluate the platform's performance under different traffic loads.
* **Security Testing:** Validate safe login, secure payments, and data encryption
* **Accessibility Testing:** The platform will be tested for accessibility compliance with relevant standards (e.g., WCAG).
* **Compatibility Testing:** The platform will be tested on different browsers and devices to ensure compatibility.

#### ****7. Testing Strategy****

* **Pre-Development**:
  + Review platform requirements and wireframes for clarity and testability.
  + Prepare a comprehensive test plan and identify the testing environment.
* **Development Phase**:
  + Conduct unit testing for individual modules (e.g., user registration, product listings).
  + Perform integration testing to ensure modules work together (e.g., product search with filtering).
* **System Testing**:
  + Execute end-to-end tests for the complete user flow, from registration to checkout.
  + Test administrative features, including managing artisan profiles and product approvals.
* **User Acceptance Testing**:
  + Conduct UAT sessions with selected artisans and customers to collect feedback.
* **Post-Deployment**:
  + Run smoke tests to ensure critical functions work as expected after go-live.
  + Schedule periodic regression tests to maintain quality during updates.

#### ****10. Tools and Resources****

* **Bug Tracking Tools**: Jira or Trello.
* **Testing Frameworks**: Selenium (automation), Postman (API testing).
* **Performance Testing**: JMeter, Lighthouse.
* **Collaboration**: Slack, Google Workspace.

#### ****9. QA Risk Management****

* **Risks Identified**:
  + Missing critical features during testing due to unclear requirements.
  + Poor performance during high traffic periods (e.g., sale events).
  + Security vulnerabilities in payment processing.
* **Mitigation Strategies**:
  + Engage stakeholders early to clarify requirements.
  + Use automated testing tools to ensure full coverage.
  + Stress-test the platform under simulated high loads.
  + Conduct thorough security audits before deployment.