**Feature Specification: User Feedback Mechanism**

## 1. Introduction

- **1.1 Feature Name:** User Feedback Mechanism
- **1.2 Feature ID:** FEAT-006
- **1.3 Version:** 1.0
- **1.4 Date:** June 12, 2025
- **1.5 Author(s):** Maximo Valencia
- **1.6 Stakeholders:** Product Management, UI/UX Team, Engineering Team, QA Team, Customer Success, Marketing.
- **1.7 Category:** Frontend, Backend, Database

## 2. Overview

This feature will implement a direct and easily accessible way for users to submit feedback (suggestions, bug reports, general comments) directly within the application. The goal is to streamline the feedback collection process, making it simpler for users to contribute and easier for our team to categorize and respond to their input.

## 3. Goals and Objectives

- **3.1 Primary Goal:** To establish an effective channel for continuous user feedback, enabling iterative product improvement.
- **3.2 Objectives:**
    - Provide a persistent and easily discoverable "Feedback" entry point within the application.
    - Allow users to submit different types of feedback (bug, suggestion, general comment).
    - Collect essential user and context information automatically with each submission.
    - Integrate feedback submissions with a central tracking system for review and action.

    - Provide users with confirmation of their submission.

**4. User Stories**

- As a user, I want an easy way to report a bug I encounter so the development team can fix it.
- As a user, I want to submit a suggestion for a new feature so my ideas can be considered.
- As a user, I want to provide general comments about my experience without having to leave the application.
- As a product manager, I want to receive organized user feedback in a central location to inform product decisions.
- As a developer, I want clear, contextual information (e.g., browser, OS) when a bug is reported to aid in debugging.

**5. Functional Requirements**

- **FR-5.1 Feedback Entry Point:** A floating "Feedback" button or a prominent link in the footer/help menu must be present on all application pages.
- **FR-5.2 Feedback Form:** Clicking the entry point must open a modal or dedicated page with a feedback submission form.
- **FR-5.3 Feedback Type Selection:** The form must include a dropdown or radio buttons for users to select the feedback type:
    - **Bug Report:** For reporting technical issues.
    - **Feature Suggestion:** For proposing new features or enhancements.
    - **General Comment:** For anything else.
- **FR-5.4 Text Input:** A multi-line text area for the user to describe their feedback (minimum 50 characters, maximum 1000 characters).
- **FR-5.5 Optional Screenshot/Attachment (V2, but consider for future):** Ability to attach screenshots or files to submissions. (Out of scope for V1 but good to mention for future planning).

- **FR-5.6 Automatic Data Collection:** Automatically capture and include the following with each submission (invisible to the user):
    - User ID (if logged in)
    - Current URL / page path
    - Browser name and version
    - Operating System
    - Timestamp of submission
- **FR-5.7 Submission Confirmation:** Upon successful submission, display a confirmation message to the user.
- **FR-5.8 Integration with Tracking System:** Submitted feedback must be sent to a designated internal system (e.g., Jira, Asana, Zendesk Support) as a new issue/ticket, populating relevant fields.

**6. Non-Functional Requirements**

- **NFR-6.1 Security:** User-submitted content must be sanitized to prevent XSS or other injection attacks.
- **NFR-6.2 Performance:** The feedback form should load quickly and not disrupt the user's current workflow.
- **NFR-6.3 Data Integrity:** Ensure reliable delivery of feedback to the tracking system.
- **NFR-6.4 Scalability:** The system should handle a potentially high volume of feedback submissions.

**7. UI/UX Specifications**

- **7.1 Feedback Button:** A small, consistent, and easily recognizable "Feedback" button, potentially floating on the side of the screen or fixed in the footer.
- **7.2 Modal/Form Layout:** A clean, intuitive form within a modal that doesn't obscure the entire page.
- **7.3 Clear Instructions:** Concise labels and placeholder text for form fields.

- **7.4 Confirmation Message:** A friendly and reassuring message after submission, thanking the user for their input.
- **7.5 Error Handling:** Clear validation messages for invalid or missing input.
- **7.6 Branding:** Consistent branding with the rest of the application's design.

**8. Technical Design & Implementation Details (High-Level)**

- **8.1 Technology Stack:** Javascript, Node.js, Python, Java
- **8.2 Frontend Component:** Develop a reusable React/Vue/Angular component for the feedback button and modal form.
- **8.3 API Endpoint:** Create a dedicated API endpoint (e.g., /api/feedback/submit) that accepts the feedback data.
- **8.4 Backend Integration:**
    - When the API endpoint receives data, it will perform validation.
    - Then, it will use the SDK/API of the chosen tracking system (e.g., Jira API, Zendesk API) to create a new ticket.
    - Map the feedback type to a corresponding issue type or tag in the tracking system.
    - Include automatic data as custom fields or part of the ticket description.
- **8.5 Error Logging:** Implement robust error logging for failed feedback submissions to the tracking system.

**9. Test Cases (Examples)**

- **TC-9.1:** Verify clicking the "Feedback" button opens the form correctly.
- **TC-9.2:** Submit a "Bug Report" with valid text; verify confirmation message and ticket creation in the tracking system with correct type and auto-collected data.
- **TC-9.3:** Submit a "Feature Suggestion" with valid text; verify confirmation message and ticket creation.

- **TC-9.4:** Submit a "General Comment" with valid text; verify confirmation message and ticket creation.
- **TC-9.5:** Attempt to submit with empty text field; verify validation error message.
- **TC-9.6:** Verify the auto-collected data (User ID, URL, Browser, OS) is accurately captured and sent.
- **TC-9.7:** Test responsiveness of the feedback form across different device sizes.

**10. Open Questions / Dependencies**

- Which specific internal tracking system (Jira, Asana, Zendesk, etc.) will feedback be sent to?
- What are the exact fields and mappings required for the chosen tracking system?
- What is the desired response time for acknowledging or acting on submitted feedback? (More of a process question, but relevant).
- Should users be able to receive replies to their feedback via email? (Out of scope for V1, but a common follow-up request).