```python
# ==========================================================
# 1. ML Core Mock (Inside ml_core/predictor.py)
# This simulates the Sentiment Prediction Engine (FR2)
# ==========================================================

import random

def classify_review(text: str) -> tuple[str, float]:
    """
    Mocks the machine learning classification step.
    In a real project, this function would load a trained model (e.g., via pickle).
    """
    # Simple logic based on keywords (for demonstration)
    text = text.lower()

    if "great" in text or "amazing" in text or "perfect" in text:
        sentiment = "Positive"
        confidence = round(random.uniform(0.8, 0.99), 2)
    elif "broke" in text or "disappointed" in text or "poor quality" in text:
        sentiment = "Negative"
        confidence = round(random.uniform(0.7, 0.95), 2)
    else:
        sentiment = "Neutral"
        confidence = round(random.uniform(0.6, 0.8), 2)

    return sentiment, confidence

# Example usage:
# sentiment, conf = classify_review("This is a truly great product!")
# print(f"Classified: {sentiment} with {conf} confidence")
# ==========================================================
# 2. API Handler Snippet (Inside app.py)
# This handles the Review Submission API (FR1) using Flask.
# ==========================================================

from flask import request, jsonify

# Assumed: 'app' is already a Flask instance and 'classify_review' is imported.
def mock_classify_endpoint():
    """Endpoint for receiving and classifying reviews."""
    data = request.get_json()
    review_text = data.get('review_text')

    if not review_text:
        # Fulfills NFR: Reliability & Error Handling
        return jsonify({"error": "Missing review_text"}), 400

    # Call the ML Core
    sentiment, confidence = classify_review(review_text)

    # In a real app, you would now call the insert_review() function (see snippet 3)

    return jsonify({
        "status": "success",
        "review": review_text,
        "sentiment": sentiment,
        "confidence": confidence
    })

# Note: In the actual app.py, this would be decorated: @app.route('/api/classify', methods=['POST'])
```