MVP Specification: Scope 3 Emissions Estimation Platform Overview: This document provides a detailed technical specification for building a Minimum Viable Product (MVP) that extracts data from supplier invoices and estimates Scope 3 emissions using simple emission factors. The goal is to enable automated ingestion, parsing, categorization, emissions calculation, and visualization. Pipeline Components: 1. File Upload & OCR - Input: PDF or image invoices. - Use Tesseract or EasyOCR. - Output: Raw text string extracted from invoice. 2. Rule-Based Parsing - Extract supplier name, item description, quantity (kg or tons), amount (USD), transport mode, and distance if present. - Use regex patterns and string matching. - Produce structured data as a list of dictionaries. 3. Categorization Module - Map item descriptions to categories (steel, transport, packaging, other). - Use simple keyword-based mapping. 4. Emission Factor Table (Hardcoded) Example emission factors: - steel_per_kg = 2.0 kg CO2e/kg - packaging_per_kg = 1.5 kg CO2e/kg - transport_per_tkm = 0.06 kg CO2e/ton-km - other_per_usd = 0.4 kg CO2e/USD 5. Emissions Calculation Logic Use formula based on available fields: - Steel: qty_kg × steel_per_kg - Packaging: qty_kg × packaging_per_kg - Transport: (tons × km) × transport_per_tkm - Fallback: amount_usd × other_per_usd 6. Aggregation - Total emissions per invoice. - Aggregate emissions by supplier. - Aggregate emissions by category. - Compute supplier carbon intensity score (normalized 0–100). 7. Dashboard / UI Requirements - Upload input box. - Display total emissions (kg CO2e). - Bar chart: emissions by supplier. - Bar chart or pie chart: emissions by category. - Table with supplier, emissions, score, comments. - Auto-generated recommendation: identify worst supplier and output a hint. 8. File/Folder Structure (Suggested) src/ ocr.py parser.py categorize.py factors.py emissions.py aggregate.py app.py (Streamlit or Flask) data/ sample_invoices/ emission_factors.json output/ parsed_data.json 9. Example Output JSON [{ "supplier": "SteelCo", "description": "Hot rolled steel", "qty_kg": 5000, "amount_usd": 10000, "category": "steel", "emissions_kg": 10000 }, { "supplier": "ShipFast", "description": "Road freight", "weight_tons": 20, "distance_km": 1000, "category": "transport", "emissions_kg": 1200 }] 10. Recommendation Logic Worst supplier = max emissions. Output message: "Focus decarbonization efforts on to reduce Scope 3 emissions." This document is ready to be fed into Claude Code or any code generator to scaffold the entire MVP.