BFA Technical Specification: The Dual-Stream Feasibility Pipeline

Project Name: Business Feasibility Analyzer (BFA)

Module: Core Processing Pipeline v1.0

Architecture Type: Parallel Processing with Cross-Modal Validation

1. Executive Summary

The BFA Pipeline is designed to automate the feasibility analysis of new business proposals. It utilizes a **Dual-Stream Architecture** that separates unstructured semantic data (proposal documents) from structured quantitative data (financials/projections). This separation allows for specialized processing of "Intent" versus "Metrics" before synthesizing them into a unified risk assessment with visualized weak points and actionable feedback.

---

2. High-Level Architecture Flow

The pipeline operates in five distinct sequential stages:

1. **Dual-Input Ingestion:** User submits qualitative and quantitative data separately.
2. **Parallel Pre-Processing:** Simultaneous NLP parsing (Text) and Statistical Analysis (Numeric).
3. **The Synthesis Core:** Integration of user data with internal company databases (HR/Finance) for validation.
4. **Logic & Visualization Engine:** Detection of risk patterns ("Falling Graphs") and generation of feedback.
5. **Dashboard Rendering:** Final output display to the decision-maker.

---

3. Detailed Stage Breakdown

Stage 1: The Dual-Input Gateway (Frontend)

The user interface is strictly divided to ensure data hygiene at the source.

- **Input Stream A: Context Layer (The "Why" & "How")**
  - **Data Type:** Unstructured Document (PDF, DOCX, TXT).
  - **Content:** Executive Summary, Technical Architecture, Market Strategy, Operational Plan.
  - **Constraint:** File size limit (e.g., 10MB), text-selectable format required.
- **Input Stream B: Metrics Layer (The "What")**
  - **Data Type:** Structured Data (CSV upload or Interactive Grid).

  - **Content:**
    - **Budget:** CAPEX, OPEX, Marketing spend.
    - **Projections:** 12-36 month Revenue, User Growth, ROI.
    - **Resources:** Headcount required, Time-to-Market (months).

Stage 2: Parallel Pre-Processing

Once submitted, the backend splits the workload into two processing tracks.

Track A: Semantic Understanding (NLP)

- **Text Extraction:** OCR/Parser removes formatting to extract raw text.
- **Entity Recognition (NER):** Identifies key technologies, target demographics, and competitors mentioned in the text.
- **Intent Parsing:** Summarizes the "Project Goal" (e.g., *“To launch a mobile-first SaaS platform in the APAC region”*) to provide context for the numeric analysis.

Track B: Numeric Normalization

- **Standardization:** Converts all currency to the corporate base currency (e.g., USD) and aligns timelines to Fiscal Quarters (Q1-Q4).
- **Sanity Check:** Validates that $Revenue > 0$, $Cost > 0$, and $Profit = Revenue - Cost$.

Stage 3: The Integration & Validation Core

This stage determines *truth* by comparing User Inputs against Internal Reality.

- **Internal Benchmarking (The "Reality Check"):**
  - *Input:* User claims project takes 3 months.
  - *Internal Data:* Historical R&D data shows similar projects take 7 months.
  - *Outcome:* Flagged as **"Optimism Bias Risk."**
- **Cross-Modal Validation:**
  - Checks for consistency between the Document (Stream A) and the Numbers (Stream B).
  - *Example:* Document mentions "Aggressive Marketing Push" but Spreadsheet shows "Zero Marketing Budget."
  - *Outcome:* Flagged as **"Strategic Mismatch."**

Stage 4: Visualization & Logic Engine

This module generates the specific "Weak Points" analysis and graph data.

- **Trend Analysis (The "Falling Graph" Detector):**
  - The engine calculates the derivative (slope) of the Revenue Projection curve.
  - **Logic:** If the slope becomes negative (<0) for two consecutive periods, or if the Net Profit margin dips below the company threshold (e.g., 10%), it marks those data points as **Critical Weak Points**.
- **Feedback Generation:**
  - Constructs natural language sentences based on the flags.

    - *Positive:* "Budget allocation aligns with Q3 fiscal goals."
    - *Negative:* "Projected growth creates a 'Falling Graph' trajectory in Year 2; ROI is insufficient."

Stage 5: Final Output Dashboard

The user receives a structured JSON object rendered into a UI.

- **The Verdict:** Feasible / Feasible with Adjustments / Not Feasible.
- **Interactive Graphs:**
    - **Revenue Projection:** Line graph. Red zones highlight "Falling" periods.
    - **Budget Burn:** Bar chart. Shows projected spend vs. Company Cap.
- **Textual Feedback:** Bulleted list of Pros and Cons derived from the analysis.

---

4. Data Payload Specification (Sample Output)

This is the JSON structure the backend sends to the frontend to visualize the results.

JSON

```
{
  "project_id": "BFA-2024-001",
  "feasibility_score": "Feasible with Adjustments",
  "score_confidence": 0.85,
  "analysis_summary": {
    "aim": "Launch AI-driven Logistics Platform",
    "market_fit": "High",
    "financial_risk": "Medium"
  },
  "feedback": {
    "positive": [
      "Technical stack aligns with current R&D capabilities.",
      "Initial 6-month growth projection is strong."
    ],
    "negative": [
      "Marketing budget is 40% below industry standard.",
      "Year 2 Q3 shows a significant revenue dip (Falling Graph)."
    ]
  },
  "graph_data": {
    "projections": [
      {"period": "Q1", "revenue": 10000, "status": "healthy"},
      {"period": "Q2", "revenue": 15000, "status": "healthy"},
      {"period": "Q3", "revenue": 12000, "status": "warning_falling"},
      {"period": "Q4", "revenue": 9000, "status": "critical_falling"}
    ],
    "weak_points": [
      {
        "type": "Budget Mismatch",
        "description": "Proposed $10k marketing vs Required $50k",
        "severity": "High"
      }
    ]
  }
}
```

5. Recommended Technology Stack

Component	**Technology**
Frontend	React.js (for interactive Graphs via Recharts or Chart.js)
Backend API	Python (FastAPI or Django REST Framework)
Document Parsing	Apache Tika / PyMuPDF
Numeric Processing	Pandas / NumPy
Database	PostgreSQL (Structured data), MongoDB (Logs/Documents)
Security	OAuth2.0, AES-256 Encryption for stored proposals