Project Blueprint - Personal Finance Analyst

Project Name: Personal Finance Analyst

One-liner: An AI Assistant that answers personal finance questions by running real computations over my transactions CSV, and always returns evidence rows and a computation trace.

Primary Goal: Trustworthy finance insights (no made-up numbers)

Target user: A person with a transactions spreadsheet who wants trustworthy insights.

What it does:
- Upload CSV —> system ingests + normalizes it
- Ask a natural-language question and get a computed answer
- Monthly summary ("show my spending/income for Nov 2025")
- Category analysis ("how much on groceries vs eating out last month?")
- Merchant analysis ("top merchants and trends")
- Anomalies ("Find unusual high transactions this month")
- Comparison ("compare spending between Oct and Nov")

Input data columns:
- Date (example: Sat, 24 May 2025)
- Amount (example: $6.15)
- Where? (example: Uber)
- What? (example: Cab - home to FedEx)
- Category (one of: Travel, Food, Essentials, Personal, Home, Others)
- Source (one of: Chase, Credit, Starbucks, BofA, Cash)

Normalization rules:
- Parse Date into ISO format: YYYY-MM-DD
- Amount convention: Expenses are positive, Incomes (i.e. settlements from friends, etc.) are negative
- Parse Amount: strip $ and commas, and store as decimal (example: 6.15)
- Create derived fields:
  - year_month = YYYY-MM
  - is_income = amount < 0
  - is_expense = amount > 0
  - abs_amount = abs(amount)
- Standardize text: trim spaces, keep original strings
- Missing or unspecifed categories become "Uncategorized"
- Assign a stable transaction_id per row
- If any required column is missing, return a clear ingest error

Expected output format:

Every response must include:
1. final_answer: short, human-readable answer
2. numbers: structured computed values
3. evidence: list of transaction rows used
4. trace: tool calls executed

Example (conceptual):

1. final_answer: “You spent $512 on food in Nov 2025.”
2. numbers: { “merchant”: “Uber”, “month”: “2025-11”, “expense_total”: 112.40, “count”: 7 }
3. evidence: list of transaction IDs + date + merchant + amount
4. trace: SQL query or tool params used

Architecture:

Components:

- API layer (FastAPI)
  - /ingest : upload + validate CSV
  - /summary/monthly : deterministic monthly stats
  - /query : natural language —> tool execution —> final response
- Storage
  - SQLite initially (easy), Postgres optional later
  - Tables: transactions, ingests, query_logs (optional)
- Computation engine (deterministic)
  - Aggregations: totals by month/category/merchant/source
  - Anomaly candidates (basic rules)
- LLM orchestrator
  - Converts user question —> structured intent + filters
  - Calls tools (SQL / compute functions)
  - Formats final answer using computed results
- Evaluation harness
  - Runs a set of standard questions
  - Compares results with deterministic calculations

Data flow:

1. User uploads CSV
2. System stores normalized transactions
3. User asks question
4. LLM selects tool call
5. Tool computes answer
6. LLM explains using computed values + evidence
7. Log + evaluate

Guardrails:

1. No numeric hallucinations
   - The model cannot output numbers unless they come from tool results.
2. Evidence required
   - Every numeric answer must include evidence transaction rows.
3. Trace required
   - Every numeric answer must include a trace of what was computed and how.
4. Clarify missing scope
   - If month/category/source/merchant isn’t specified and needed, ask a follow-up.
5. Read-only tools
   - Tool layer must not modify transaction data during query (SELECT-only SQL or safe compute functions).
6. Refuse gracefully
   - If data doesn’t contain the needed info, say “I don’t know from the data provided” and suggest what to upload/clarify.

Supported question types (MVP):
Must support:
- Monthly total spending: “How much did I spend in May 2025?”
- Category totals: “Food spend in May 2025?”
- Merchant totals: “How much on Uber in May 2025?”
- Source totals: “How much did I spend via Cash in May 2025?”
- Top merchants: “Top 5 spending merchants in May 2025”
- Income totals: “How much income/settlements in May 2025?” (negative amounts)
- Net: “Net in May 2025 (income - expenses)”

Nice-to-have:
- Comparison: “Compare Food spend May vs Jun 2025”
- Basic anomaly: “What are my biggest unusual spends in May 2025?”

Evaluation Plan:
Create about 30 evaluation questions, covering:
- totals (month/category/merchant/source)
- comparisons
- edge cases (ambiguous scope)
- refusal cases (asking for info not present)

Metrics to measure:
- Numeric accuracy: matches deterministic computation exactly (or within $0.01)
- Evidence presence: evidence rows included for numeric answers
- Trace presence: trace included for numeric answers
- Refusal correctness: refuses when question cannot be answered from data

Pass criteria:
- >= 90% numeric accuracy on the eval set
- 100% trace presence for numeric answers
- 100% evidence presence for numeric answers

Evaluation Question Set Categories:
1. **I.** Monthly totals
2. **II.** Category analysis
3. **III.** Merchant (“Where?”) analysis
4. **IV.** Source analysis
5. **V.** Comparisons + edge cases

Demo Test Script (questions I’ll always demo):
1. Total expenses in June 2025
2. Food spend in June 2025 + evidence rows
3. Top 5 merchants in June 2025

Scope boundaries:
- No bank API integrations
- No perfect auto-categorization (because the dataset already has categories)
- No fancy dashboards initially
- No multi-currency support
- No complex anomaly models (simple rules only)