Project Proposal: AI-Powered Expense Tracker with Smart Insights

Problem Statement

Managing personal expenses is a daily challenge for students and professionals alike. Most tools available are either too complex or lack intelligent analysis features. This project proposes the development of a privacy-first, AI-powered expense tracker. The system will use a relational database to record transactions, apply indexing and query optimization for efficiency, and integrate advanced features such as retrieval-augmented Q&A and data mining for deeper insights.

Scope and Features

The project will cover the complete database lifecycle and demonstrate advanced analytical capabilities. Key features include:

- Entity-Relationship model and relational schema for users, transactions, categories, and budgets. CRUD operations for managing expenses and budgets through a simple interface.
- SQL queries to calculate monthly totals, category breakdowns, and budget variance. Relational algebra expressions for selected queries to reinforce theoretical foundations.
- Transaction management to ensure consistent updates when logging expenses and adjusting budgets. Indexing strategies, such as composite indexes on user and date fields, to improve performance. Query optimization analysis using EXPLAIN plans before and after indexing.
- Retrieval-Augmented Generation (RAG) to allow natural-language questions over generated monthly summaries. Basic data mining techniques such as anomaly detection and clustering to identify unusual spending patterns.

Key Queries and Indexes

- Identify months where actual spending exceeded the allocated budget. List the top spending categories for each month. Compare execution plans of analytic queries with and without indexes. Retrieve spending summaries through natural-language questions.

Solution Approach

- Design ERD and create a normalized schema in PostgreSQL/SQLite.
- Implement CRUD operations using SQL Alchemy and Python (Flask/Fast API).
- Write at least 8 SQL queries for analytics and translate 2–3 into relational algebra.
- Implement atomic transactions to update budgets and expenses consistently.
- Create indexes and compare query performance using EXPLAIN/ANALYZE.
- Build a RAG module to generate monthly summaries and support natural-language queries.
- Apply anomaly detection (z-score) and clustering (k-means) to detect unusual spending.
- Test with CSV datasets and validate system correctness.

Expected Outcomes

- A working expense tracker with CRUD and analytics features.
- Optimized SQL queries showing measurable performance improvements.
- Demonstrated transaction consistency through atomic updates.
- RAG Q&A system retrieving accurate monthly summaries.
- Anomaly detection outputs highlighting unusual spending behaviour.
- Deliverables: schema diagrams, SQL scripts, query results, optimization analysis, RAG demo, and final report.

Timeline

- **Week 1 & 2**: Requirement analysis, ERD design, schema creation, CRUD operations, core SQL analytics, and relational algebra mappings.
- **Week 3 & 4**: Add transactions, indexes, and query optimization comparisons. Build the RAG Q&A module, apply anomaly detection, and finalize documentation and presentation.