**Financial QA Bot Documentation**

**Part 1: Retrieval-Augmented Generation (RAG) Model for QA Bot on P&L Data**

**Introduction:**The Financial QA Bot is a system to extract and process financial data from Profit & Loss (P&L) statements in PDF format. It employs state-of-the-art embedding models and retrieval-augmented generation (RAG) techniques for accurate financial insights based on the user's queries.

**Model Architecture**

**1. Data Extraction**

* **PDF Parsing:**
  + The system uses PyPDF2 to extract text from financial statements in PDF format.
  + Addresses the potential issues like missing text with proper fallback mechanisms.
* **Text Preprocessing:**
  + Extracts key-value pairs from the extracted text by using regex-based parsing.
  + The Structured financial data is stored in a Pandas DataFrame with columns of **Metric** and **Value** columns.

**2. Data Embedding and Storage**

* **Embedding Model:**
  + It uses `all-MiniLM-L6-v2` from the SentenceTransformers library for embedding financial terms.
  + Creates 384-dimensional embeddings to best represent financial data.
* **Vector Database:** 
  + Uses Pinecone as the vector database to store embeddings and for fast retrieval.
  + Handles index creation and upsertion of data with minimal duplication.

**3. Retrieval-Augmented Generation (RAG) Model**

* **Retrieval Mechanism:**
  + Embeds queries and compares them with the stored vectors by using cosine similarity.
  + Retrieves the top 3 closest results to give context to the generation.
* **Generative Response:**
  + It uses the GPT-4 model from OpenAI to come up with an answer depending on the retrieved financial context.
  + Use of structured prompt engineering for relevant and precise responses.

**Approach to Data Extraction and Preprocessing**

**Step 1: Load and Extract Data**

pdf\_path = "Sample Financial Statement.pdf"

text = extract\_text\_from\_pdf(pdf\_path)

* Opens the PDF and extracts text from each page.
* Tackles None values to avoid parsing errors.

**Step 2: Preprocessing Extracted Data**

pnl\_df = preprocess\_pnl\_text(text)

* A regex pattern for matching financial metrics and their corresponding values.
* Convert the financial values to a structured numerical format.

**Step 3: Storing Processed Data**

embed\_and\_store\_pnl\_data(pnl\_df)

* Embed and store financial metrics within the Pinecone vector database.

**Step 4: Querying Data**

answer = query\_pnl("What is the gross profit for Q3 2024?")

* Retrieves relevant financial context and provides accurate answers.

**Challenges Faced and Solutions**

**1.** **PDF Text Extraction Problems**

* **Problem:** Inconsistent text formatting in PDFs that leads to parsing errors.
* **Solution:** Developed robust regex-based parsing and fallback mechanisms for missing data.

**2. Dealing with Big Data**

* **Challenge:** Processing and storing large P&L reports efficiently.
* **Solution:** Optimized embedding storage with batched upsertion to Pinecone.

**3. Query Accuracy**

* **Challenge:** Generating accurate financial insights based on limited context.
* **Solution:** Implemented top-k retrieval and fine-tuned GPT-4 prompts for better context usage.

**4. Environment Management**

* **Challenge:** Secure storage and retrieval of API keys.
* **Solution:** Leveraged ‘**dotenv’** to manage environment variables securely.

**Collab Notebook Demonstration**

**Pipeline Overview**

1. **Loading and Preprocessing Financial Data**
   * Load PDFs, extract text, and convert data into structured format.
2. **Embedding Financial Terms**
   * Utilize transformer models to produce vector embeddings.
3. **Implementing RAG Model**
   * Store the embeddings in Pinecone and fetch relevant results.

**Conclusion**

The Financial QA Bot efficiently processes financial data from P&L statements offering valuable insights by using a combination of advanced NLP techniques and vector search technologies. This solution is well-suited for financial analysis tasks due to its accuracy, scalability, and ease of use.

**Part 2: Interactive QA Bot Interface for Financial Data**

**Overview**

The Financial P&L Query Bot is a streamlit-based application where users can upload financial documents (PDFs), extract structured data, and query this information using a retrieval-augmented generation (RAG) model. The system utilizes OpenAI's GPT model coupled with Pinecone for vector-based search. It thus produces accurate and contextually relevant financial insights.

**Features**

* **PDF Upload:** Users upload financial PDFs with P&L tables.
* **Data Extraction:** Takes in uploaded PDFs, structures the financial data, and places it in the Pinecone instance.
* **Embedding & Storage:** It embeds extracted financial data in Pinecone.
* **Financial Querying:** Supports asking questions in relation to extracted financial data
* **User-Friendly Interface:** Built with Streamlit for interactive use.

**Project Architecture**

**1. Data Extraction and Preprocessing**

* **extract\_text\_from\_pdf(uploaded\_file)**
  + Read the uploaded file using `pdfplumber` with high efficiency of table extraction.
  + Extractions text from every page and combines it into a single string.
* **preprocess\_pnl\_text(text)**
  + Splits the extracted text into lines.
  + Uses regex extraction of key-value pairs of financial metrics.
  + Converts extracted data into a structured Pandas DataFrame.

**2. Data Embedding and Storage**

* **embed\_and\_store\_pnl\_data(pnl\_df, index)**
  + Embeds financial metrics using SentenceTransformer.
  + Stores the embeddings in a Pinecone vector index.

**3. Query Processing**

* **query\_pnl(question, index, pnl\_df)**
  + Encode the user's question and fetch relevant embeddings.
  + Fetch matched records from Pinecone.
  + Answer generation using OpenAI's GPT model.

**Usage Guide**

**Uploading Documents**

1. Open the Streamlit app in your browser.
2. Click on the "Upload a PDF file" button.
3. Select a financial statement PDF.
4. The app will automatically extract and display the structured data.

**Asking Questions**

1. Type your financial question in the provided text box.
2. Example questions:
   * "What is the gross profit for Q3 2024?"
   * "What are the total operating expenses?"
   * "Provide details on net income."
   * Read the response generated and relevant data segments from the table below.

**Interpreting Responses**

* **Context:** The extracted text used in generating the response.
* **Answer:** AI-generated answer for the context.
* **Relevant Data Segments:** Specific P&L metrics related to the question.

**Best Practices**

**Frontend (Streamlit)**

* UI elements must be simple and intuitive.
* Error messages will be displayed to enhance user experience.
* Optimization of dataframe for better readability

**Backend (Data Processing)**

* Use of `pdfplumber` to ensure text extraction is reliable
* Regex patterns to ensure correct financial data is fetched
* Batch upsert operations in Pinecone for effective storage

**Security Considerations**

* Use environment variables for sensitive API keys
* Sanitize user input to prevent possible abuse

**Challenges Faced and Solutions**

**Challenge 1: Poor Text Extraction**

* **Problem:** Inconsistent extraction of text from PDFs.
* **Solution:** Replaced `PyPDF2` with `pdfplumber` for handling financial tables.

**Challenge 2: Handling Large P&L Statements**

* **Problem:** Performance issues in handling large datasets.
* **Solution:** Optimized data processing and implemented batch-wise embedding.

**Challenge 3: Improving Query Accuracy**

* **Problem:** Inaccurate query responses due to irrelevant context.
* **Solution:** Tuned embeddings and improved query retrieval parameters.

**Example Interactions**

**Example 1: Gross Profit Query**

**User Input:**

What is the gross profit for Q3 2024?

**Bot Response:**

Gross profit for Q3 2024 is $450,000.

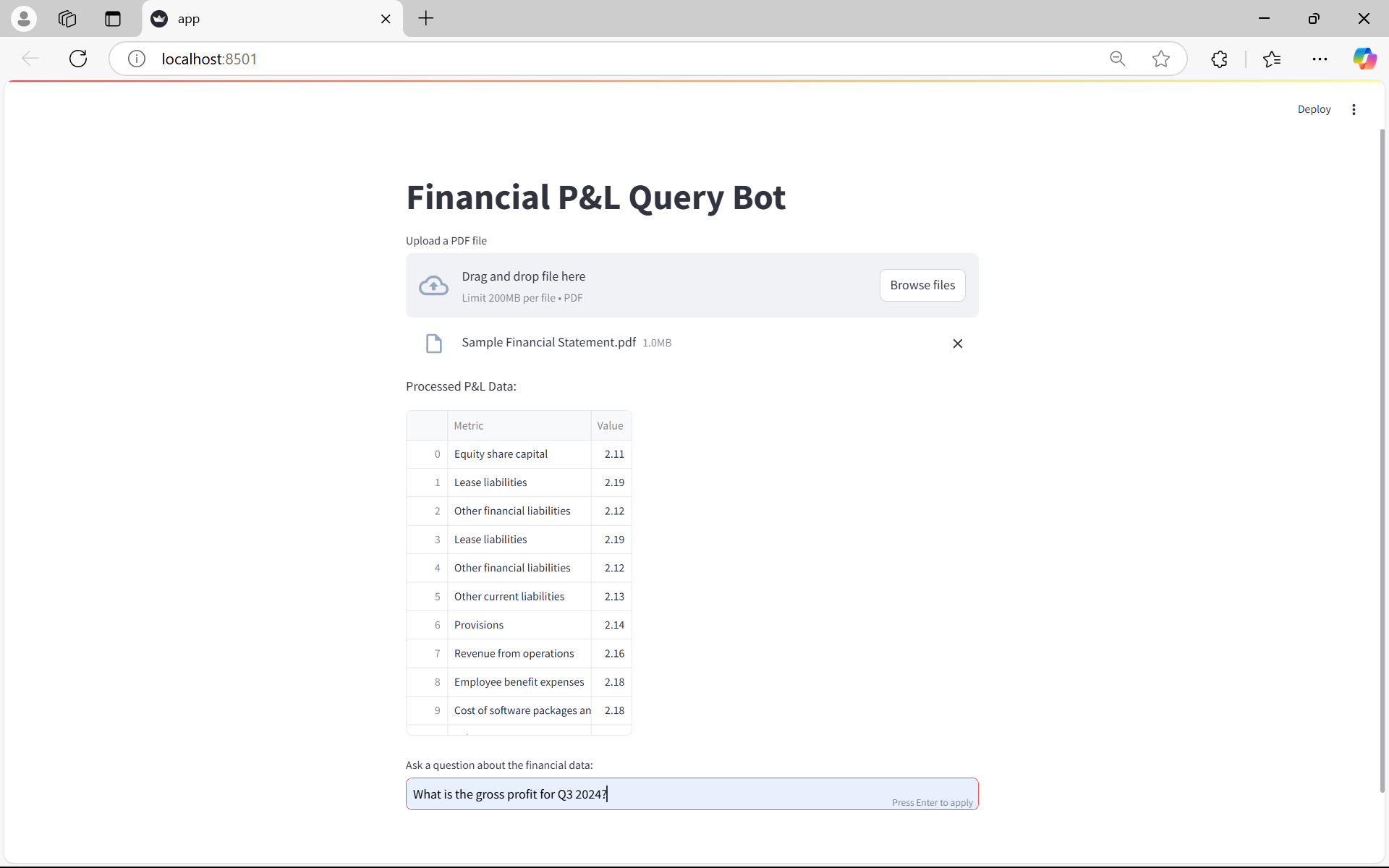
**Example 2: Checking Operating Expenses**

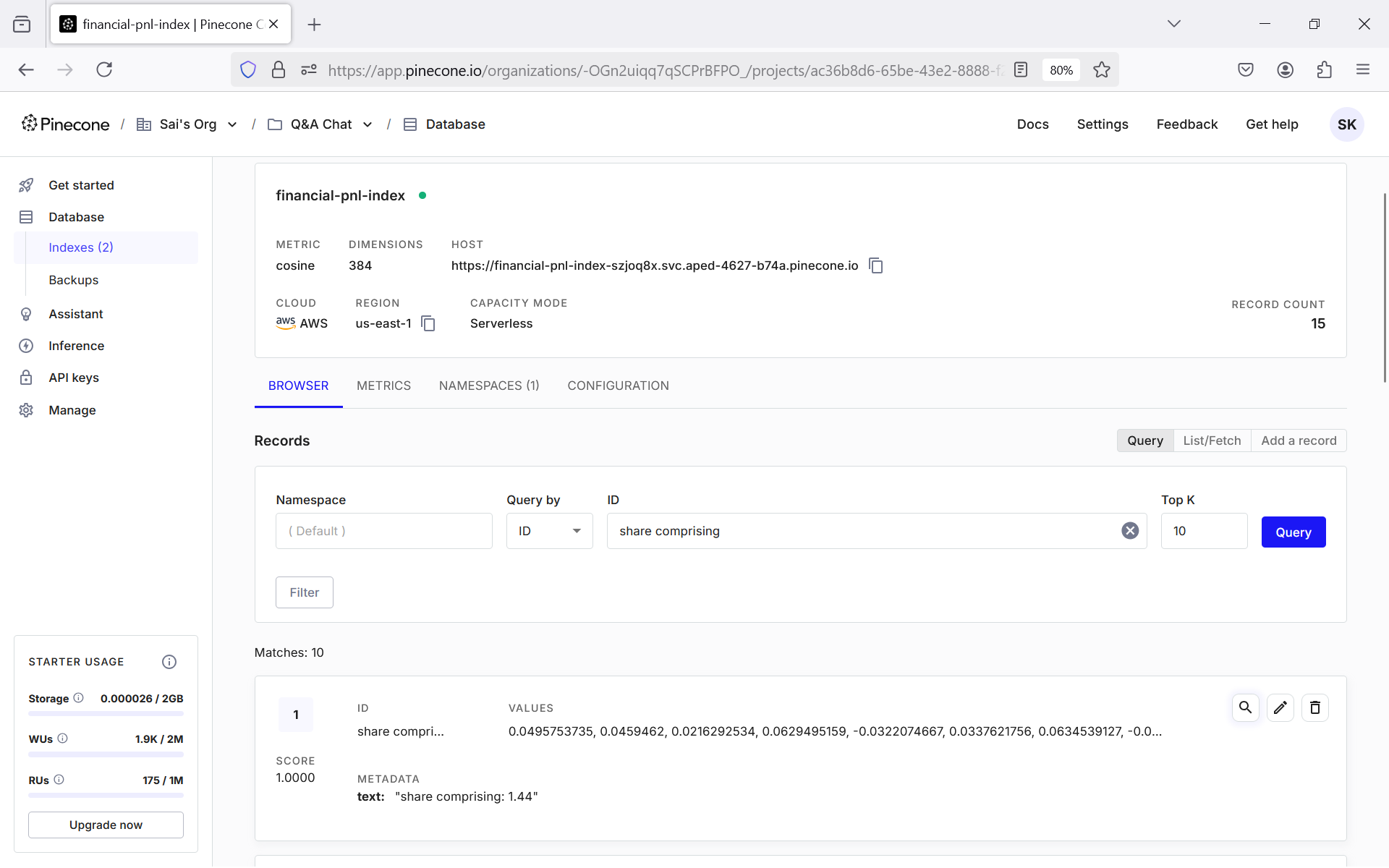
**User Input:**

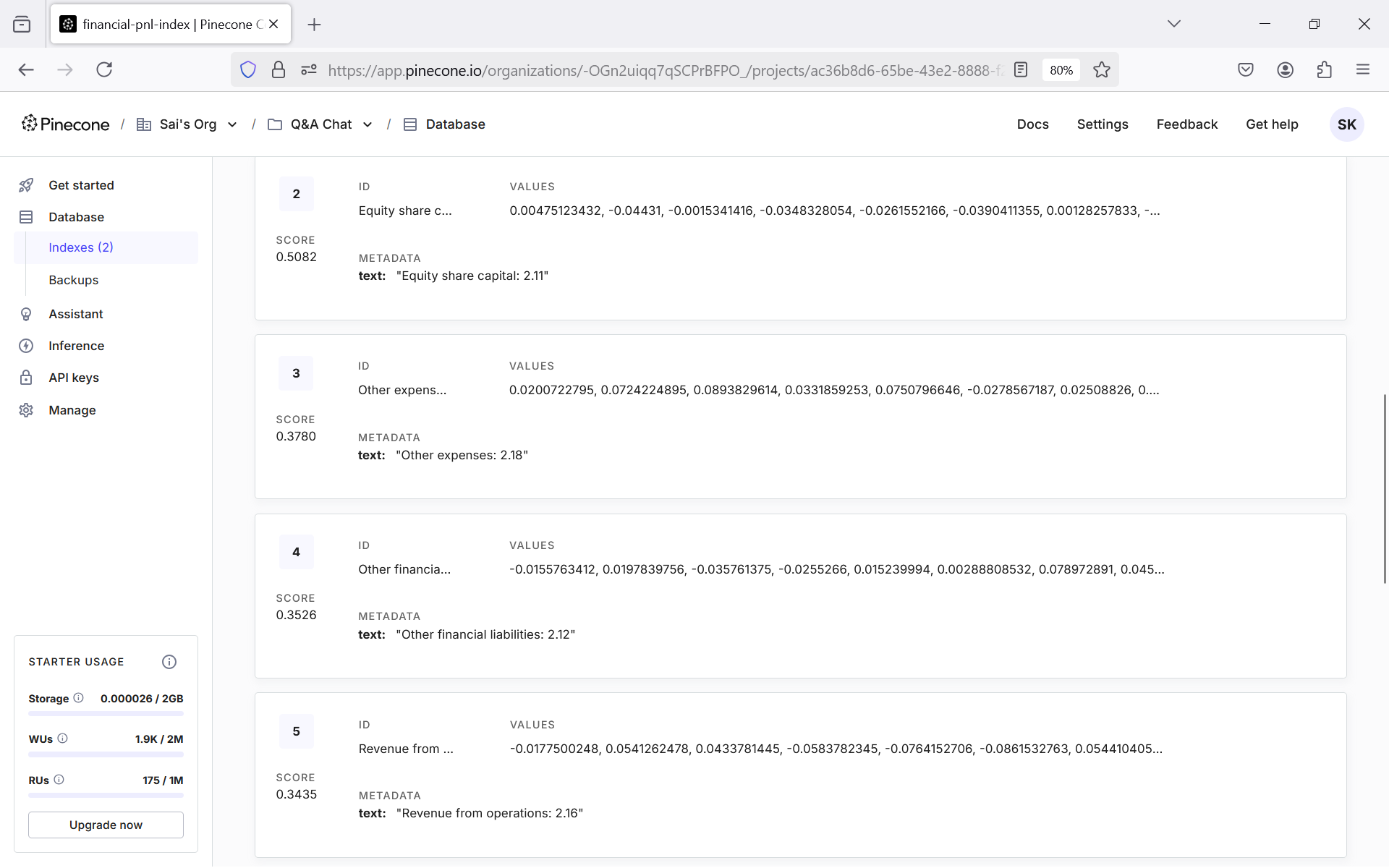
What are the total operating expenses?

**Bot Response:**

Total operating expenses amount to $200,000.







**Conclusion**

This documentation is a comprehensive guide on using the Financial P&L Query Bot. The bot is an efficient method to analyse and extract insights from financial documents by leveraging modern AI technologies and best development practices.