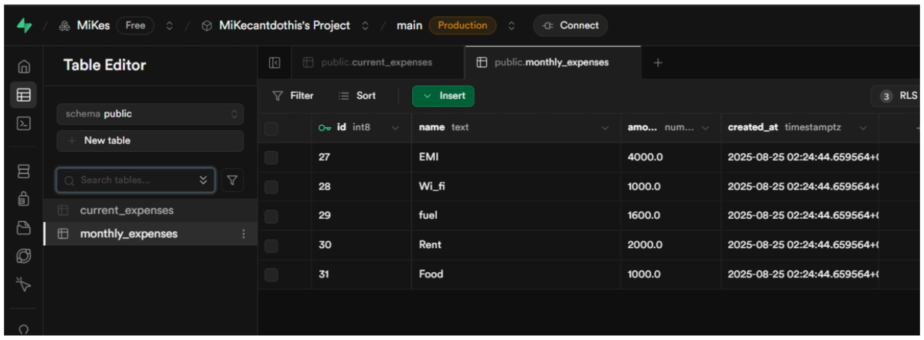
LLM CAPSTONE PROJECT

Name: TEJAS C
Reg. no: 2448553
Introduction:
A Smart Expense Tracking APP that will not only show you where your money is going, but also provides an interactive chatbot that will tell you where you are losing money and how you can save.
Backend - Context Driven chatbot uses your supabase table as a context vector to retrieve answers for a question.
Uses a simple Streamlit UI.

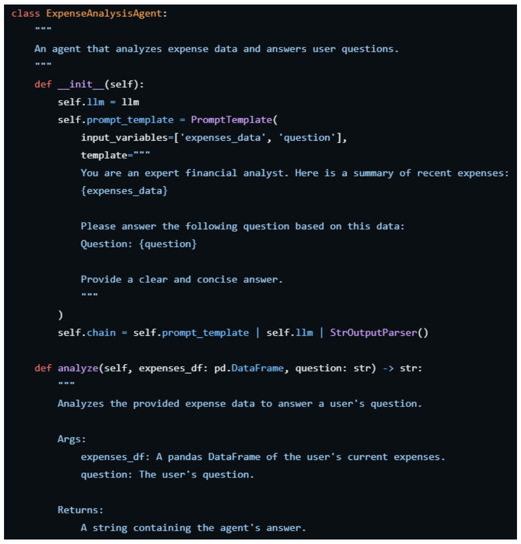
Initial Code and set up: 3 modules -
1. DataBase Operations
2. Agents
3. User interface

Code:
Uses supabase to push and pull information to a SQL Database.

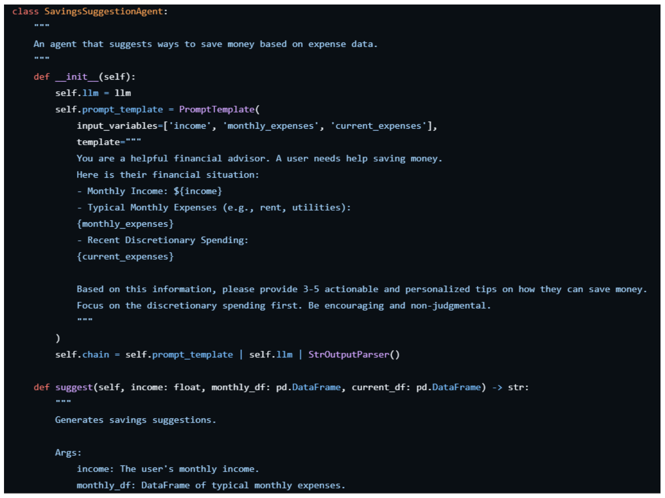


id int8	name text	amo... num...	created_at timestamptz
27	EMI	4000.0	2025-08-25 02:24:44.659564+
28	Wi_fi	1000.0	2025-08-25 02:24:44.659564+
29	fuel	1600.0	2025-08-25 02:24:44.659564+
30	Rent	2000.0	2025-08-25 02:24:44.659564+
31	Food	1000.0	2025-08-25 02:24:44.659564+

Code to set up Agents that will produce Expense Analysis and Spending Analysis:

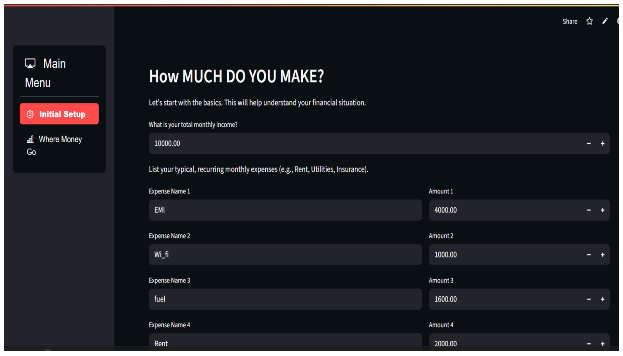


```python
class ExpenseAnalysisAgent:
    """
    An agent that analyzes expense data and answers user questions.
    """
    def __init__(self):
        self.llm = llm
        self.prompt_template = PromptTemplate(
            input_variables=['expenses_data', 'question'],
            template="""
            You are an expert financial analyst. Here is a summary of recent expenses:
            {expenses_data}

            Please answer the following question based on this data:
            Question: {question}

            Provide a clear and concise answer.
            """
        )
        self.chain = self.prompt_template | self.llm | StrOutputParser()

    def analyze(self, expenses_df: pd.DataFrame, question: str) -> str:
        """
        Analyzes the provided expense data to answer a user's question.

        Args:
            expenses_df: A pandas DataFrame of the user's current expenses.
            question: The user's question.

        Returns:
            A string containing the agent's answer.
```



```python
class SavingsSuggestionAgent:
    """
    An agent that suggests ways to save money based on expense data.
    """
    def __init__(self):
        self.llm = llm
        self.prompt_template = PromptTemplate(
            input_variables=['income', 'monthly_expenses', 'current_expenses'],
            template="""
            You are a helpful financial advisor. A user needs help saving money.
            Here is their financial situation:
            - Monthly Income: ${income}
            - Typical Monthly Expenses (e.g., rent, utilities):
            {monthly_expenses}
            - Recent Discretionary Spending:
            {current_expenses}

            Based on this information, please provide 3-5 actionable and personalized tips on how they can save money.
            Focus on the discretionary spending first. Be encouraging and non-judgmental.
            """
        )
        self.chain = self.prompt_template | self.llm | StrOutputParser()

    def suggest(self, income: float, monthly_df: pd.DataFrame, current_df: pd.DataFrame) -> str:
        """
        Generates savings suggestions.

        Args:
            income: The user's monthly income.
            monthly_df: DataFrame of typical monthly expenses.
```

Note: Uses the csv files from my supabase table to create a context vector and answer the questions asked to the chat bot.
User Interface:
Initial Set-up:



Main Menu
Initial Setup
Where Money Go

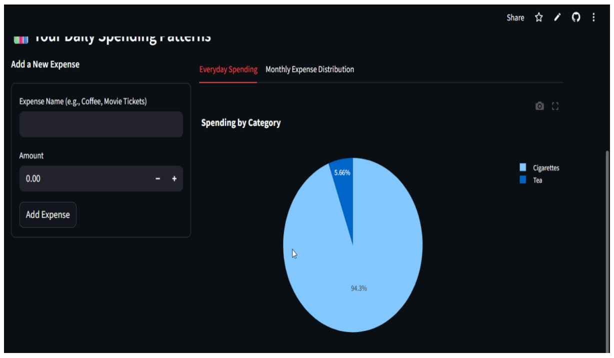
How MUCH DO YOU MAKE?

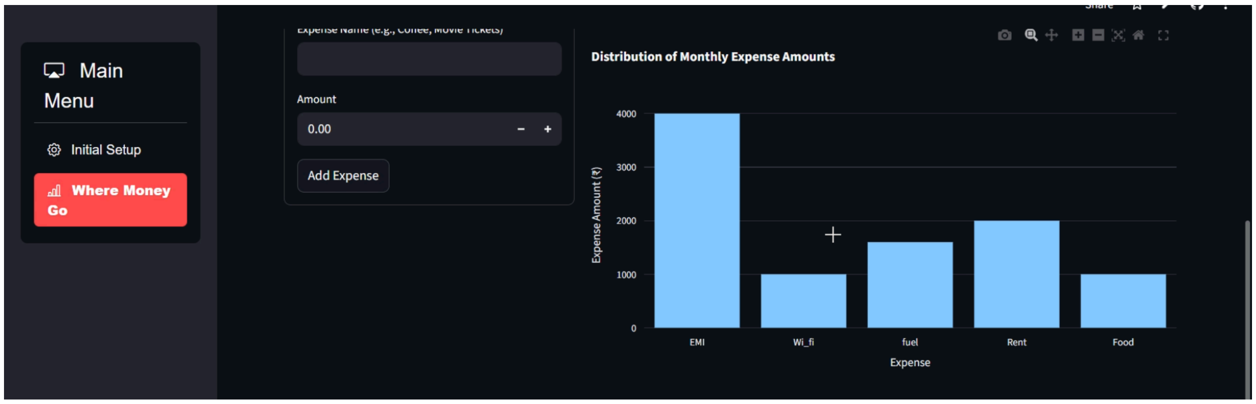
Let's start with the basics. This will help understand your financial situation.

What is your total monthly income?
10000.00

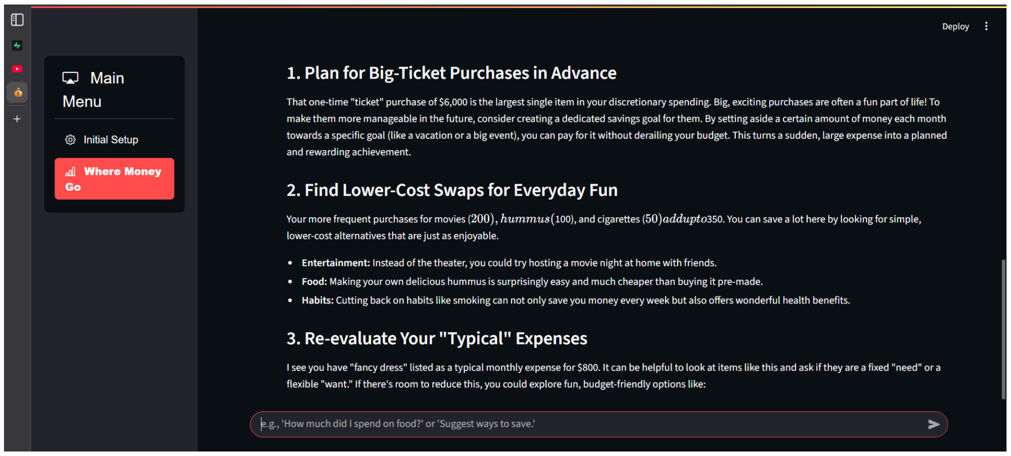
List your typical, recurring monthly expenses (e.g., Rent, Utilities, Insurance).

Expense Name	Amount
Expense Name 1: EMI	Amount 1: 4000.00
Expense Name 2: Wi_fi	Amount 2: 1000.00
Expense Name 3: fuel	Amount 3: 1600.00
Expense Name 4: Rent	Amount 4: 2000.00





ChatBot inference:



Main Menu
Initial Setup
Where Money Go

Deploy

1. Plan for Big-Ticket Purchases in Advance

That one-time "ticket" purchase of $6,000 is the largest single item in your discretionary spending. Big, exciting purchases are often a fun part of life! To make them more manageable in the future, consider creating a dedicated savings goal for them. By setting aside a certain amount of money each month towards a specific goal (like a vacation or a big event), you can pay for it without derailing your budget. This turns a sudden, large expense into a planned and rewarding achievement.

2. Find Lower-Cost Swaps for Everyday Fun

Your more frequent purchases for movies $(200), hummus(100)$, and cigarettes $(50) add up to$350. You can save a lot here by looking for simple, lower-cost alternatives that are just as enjoyable.

- **Entertainment:** Instead of the theater, you could try hosting a movie night at home with friends.
- **Food:** Making your own delicious hummus is surprisingly easy and much cheaper than buying it pre-made.
- **Habits:** Cutting back on habits like smoking can not only save you money every week but also offers wonderful health benefits.

3. Re-evaluate Your "Typical" Expenses

I see you have "fancy dress" listed as a typical monthly expense for $800. It can be helpful to look at items like this and ask if they are a fixed "need" or a flexible "want." If there's room to reduce this, you could explore fun, budget-friendly options like:

e.g., 'How much did I spend on food?' or 'Suggest ways to save.'