

Expense Feedback use case

Problem Statement:

In expense management, users often lack adequate feedback mechanisms to ensure the accuracy, compliance, and completeness of their expense reports before submission for approval. The absence of such feedback can lead to errors, non-compliance with company policies, and incomplete submissions.

Solution:

Integrating an Expense Feedback system into the expense management process to provide users with timely and actionable feedback on their expense reports. This system will leverage language models to analyze expense reports and offer suggestions for improvement, addressing issues such as missing information, errors, and non-compliance with company policies.

Key Features:

1. **Expense Analysis:** The language model analyzes expense reports to identify inaccuracies, compliance issues, and completeness.
2. **Compliance Check:** The system ensures adherence to company policies and regulatory requirements, reducing the risk of non-compliance.
3. **Feedback Generation:** Based on the analysis, the system generates feedback for users, offering specific suggestions to enhance the accuracy and compliance of their reports. Users receive instant feedback as they prepare their expense reports, allowing them to make necessary adjustments before submission.

Outcome:

- Improved accuracy, compliance, and completeness of expense reports.
- Enhanced user experience through proactive feedback mechanisms.
- Streamlined expense approval process with fewer errors and delays.

Timeline:

Week	Tasks
1	Requirements Gathering, Environment Setup
2	System Design, and Architecture Development
3	Building Frontend – UI/UX
4 and 5	Model Building
6	Integration with Frontend
7	Testing and Deployment
8	Documentation and Fine-tuning

Tech-Stack:

- Programming Languages: Python (Full stack)
- Framework: Streamlit (Python web client)
- Libraries/ APIs: OpenAI, LangChain
- Database: FAISS (Vector)
- Version Control: Git
- Deployment: Streamlit cloud