Titan Company NLP Project Report (FY 2023–24)

Abstract

This project builds a lightweight Retrieval-Augmented Generation (RAG) assistant to answer investor- and company-related questions using the Titan Company Annual Report FY 2023–24 and a Titan FAQ document. The system combines classical information retrieval (TF–IDF with cosine similarity) for fast, transparent context selection and a generative model (via OpenRouter) for fluent, grounded answers. The app is delivered as a Streamlit interface with configurable parameters and explainability through displayed source snippets.

Objectives

- Provide accurate, grounded answers strictly from Titan's authoritative documents.
- Offer explainability by surfacing the exact passages used to answer.
- Keep the solution lightweight and easily deployable on modest hardware.
- Support multiple corpora (Annual Report + FAQ PDF) and flexible retrieval settings.

Documents and Data Sources

- Annual Report: Titan Annual Report 2023-24 - 21MB 3.pdf (preprocessed into titan_chatbot_data.json with pages, vocabulary, TF–IDF matrix)
- FAQs: Titan_Company_FAQ.pdf (parsed at runtime, vectorized against the same vocabulary)

Methodology

1. Preprocessing (Annual Report)
   - Segment PDF into pages and clean text.
   - Build a vocabulary and TF–IDF matrix offline.
   - Persist artifacts into titan_chatbot_data.json for quick loading.
2. Runtime Retrieval (Annual Report + FAQ)
   - Build a CountVectorizer with the fixed vocabulary to encode queries.
   - Encode the FAQ pages into a TF–IDF matrix aligned with the same vocabulary at app startup.
   - For a user query, compute cosine similarity against both matrices and select top-k passages.
3. Answer Generation
   - Construct a prompt with the top passages (source and page metadata included).
   - Call OpenRouter (configurable model; default openai/gpt-3.5-turbo).

- Enforce groundedness: the system prompt instructs the model to only use provided context and avoid fabrication.

System Architecture

- UI Layer: Streamlit app (titan_faq_app.py) with search input, Ask button, Quick FAQs, and a context viewer.
- Retrieval Layer: TF–IDF + cosine similarity over Annual Report and FAQ corpora.
- Generation Layer: OpenRouter API integrates a hosted LLM for answer synthesis.
- Caching: Streamlit resource caching for loaded artifacts and FAQ matrix.

Implementation Overview

- File: titan_faq_app.py
  - Loads titan_chatbot_data.json (pages, vocabulary, TF–IDF matrix for Annual Report).
  - Builds a fixed-vocabulary CountVectorizer for query encoding.
  - Parses Titan_Company_FAQ.pdf using pypdf (fallback to PyPDF2) and creates a TF–IDF matrix for the FAQ corpus.
  - Retrieves top-k snippets from both sources, merges and sorts by score, and limits to k.
  - Displays the answer and the exact supporting passages with source labels (Annual Report or FAQ PDF).
  - Includes a horizontally centered Quick FAQs section to prefill common queries.
  - Adds a simple centered footer.

Key Dependencies

- Python 3.10+
- streamlit, numpy, scikit-learn, requests, pypdf or PyPDF2

Running the App

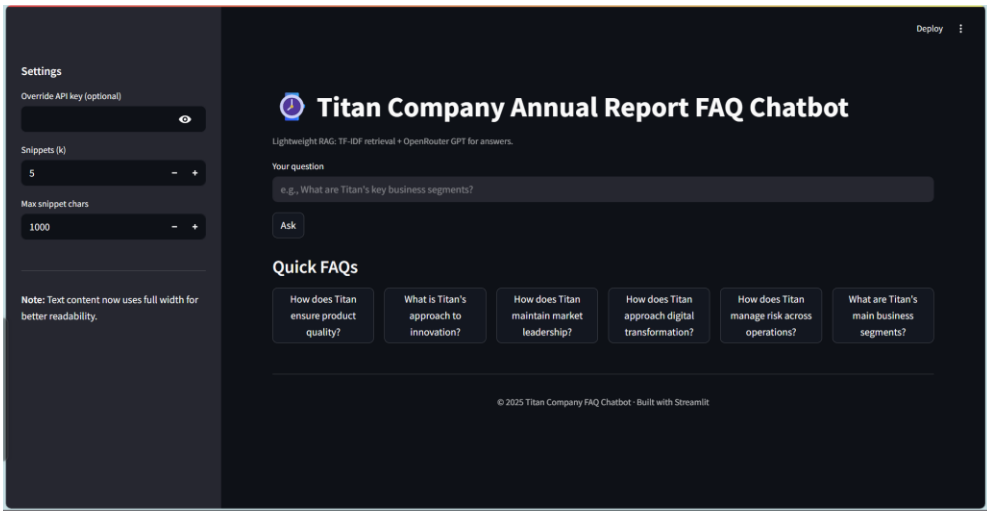
```
py -m pip install -r titan_requirements.txt  # if available

py -m pip install streamlit numpy scikit-learn requests pypdf PyPDF2

streamlit run titan_faq_app.py
```

Configuration

- API: OpenRouter key can be set in code or overridden via the sidebar.
- Retrieval: k (number of snippets) and MAX_SNIPPET_CHARS are adjustable from the sidebar.

Results (Illustrative)

- The merged retrieval from Annual Report and FAQ improved coverage for investor-centric questions.
- Displaying source metadata (Report vs FAQ) increases trust and traceability.
- Quick FAQs reduced time-to-answer for common queries.



Note: Text content now uses full width for better readability.

Titan Company Annual Report FAQ Chatbot

Lightweight RAG: TF-IDF retrieval + OpenRouter GPT for answers.

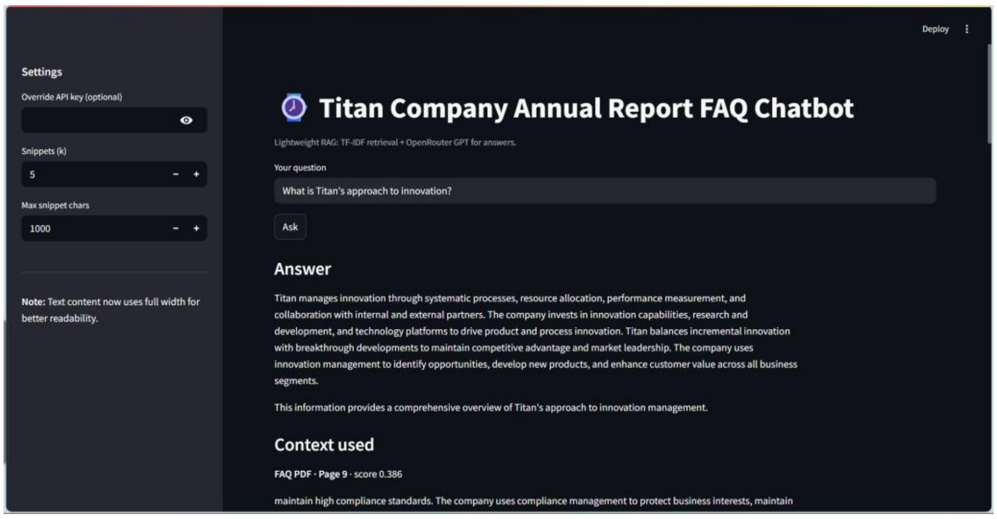
Your question

e.g., What are Titan's key business segments?

Quick FAQs

How does Titan ensure product quality? | What is Titan's approach to innovation? | How does Titan maintain market leadership? | How does Titan approach digital transformation? | How does Titan manage risk across operations? | What are Titan's main business segments?

© 2025 Titan Company FAQ Chatbot · Built with Streamlit



Note: Text content now uses full width for better readability.

Titan Company Annual Report FAQ Chatbot

Lightweight RAG: TF-IDF retrieval + OpenRouter GPT for answers.

Your question

What is Titan's approach to innovation?

Answer

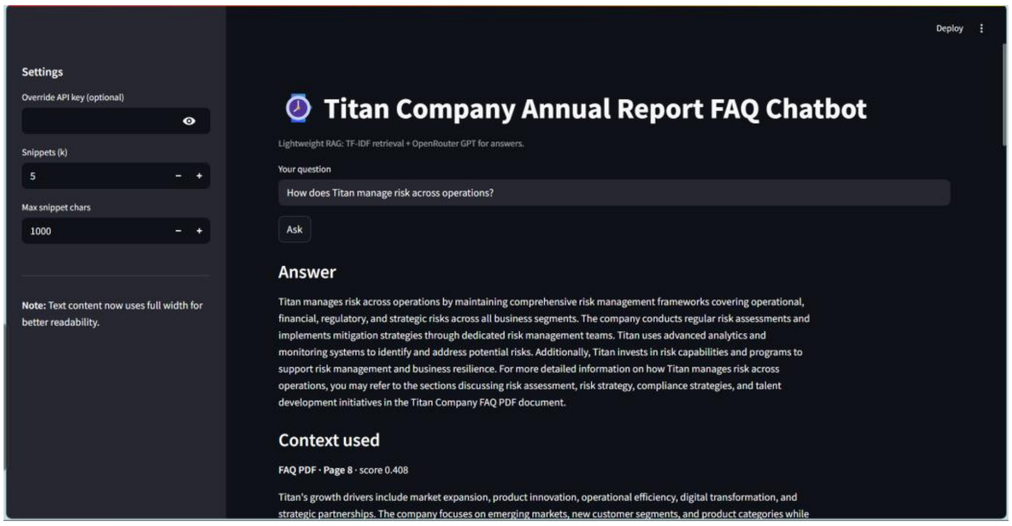
Titan manages innovation through systematic processes, resource allocation, performance measurement, and collaboration with internal and external partners. The company invests in innovation capabilities, research and development, and technology platforms to drive product and process innovation. Titan balances incremental innovation with breakthrough developments to maintain competitive advantage and market leadership. The company uses innovation management to identify opportunities, develop new products, and enhance customer value across all business segments.

This information provides a comprehensive overview of Titan's approach to innovation management.

Context used

FAQ PDF · Page 9 · score 0.386

maintain high compliance standards. The company uses compliance management to protect business interests, maintain



Note: Text content now uses full width for better readability.

Titan Company Annual Report FAQ Chatbot

Lightweight RAG: TF-IDF retrieval + OpenRouter GPT for answers.

Your question

How does Titan manage risk across operations?

Answer

Titan manages risk across operations by maintaining comprehensive risk management frameworks covering operational, financial, regulatory, and strategic risks across all business segments. The company conducts regular risk assessments and implements mitigation strategies through dedicated risk management teams. Titan uses advanced analytics and monitoring systems to identify and address potential risks. Additionally, Titan invests in risk capabilities and programs to support risk management and business resilience. For more detailed information on how Titan manages risk across operations, you may refer to the sections discussing risk assessment, risk strategy, compliance strategies, and talent development initiatives in the Titan Company FAQ PDF document.

Context used

FAQ PDF · Page 8 · score 0.408

Titan's growth drivers include market expansion, product innovation, operational efficiency, digital transformation, and strategic partnerships. The company focuses on emerging markets, new customer segments, and product categories while

Conclusion

This project delivers a practical, explainable RAG assistant for Titan Company by combining fast TF–IDF retrieval with an LLM to generate grounded answers. Merging the Annual Report with the FAQ corpus improves coverage for investor-focused questions while keeping the system lightweight and easy to deploy. The app's "Context used" view increases trust through transparent citations, and the Quick FAQs streamline common queries. With the proposed enhancements (dense retrieval, reranking, and richer citations), the system can further improve recall and user confidence while maintaining simplicity.

References

- Titan Company Annual Report 2023–24
- Titan Company FAQ document 2023–24
- OpenRouter API documentation (https://openrouter.ai/docs)
- Scikit-learn documentation (https://scikit-learn.org/)
- Streamlit documentation (https://docs.streamlit.io/)

Appendix A: Environment and Variables

```
# OpenRouter
OPENROUTER_API_KEY=<your_key>
```

Appendix B: High-Level Pseudocode

```
load pages, vocabulary, tfidf_matrix from titan_chatbot_data.json
vectorizer = CountVectorizer(vocabulary)
faq_pages = read_pdf(FAQ_PDF_PATH)
faq_matrix = TfidfTransformer().fit_transform(vectorizer.transform(text_of(faq_pages)))

def retrieve(query):
    q = normalize(vectorizer.transform([query]))
    s1 = cosine_similarity(q, matrix).argsort(desc)[:k]
    s2 = cosine_similarity(q, faq_matrix).argsort(desc)[:k]
    return top_k_by_score(merge(s1_pages, s2_pages), k)

snippets = retrieve(user_query)
prompt = build_prompt(snippets, user_query)
answer = openrouter(prompt)
display(answer, snippets)
```