



GEN AI – AUTOMATION OF PROCUREMENT PROCESS

Assignment 3 – Group Submission



MARCH 14, 2025

GROUP 4

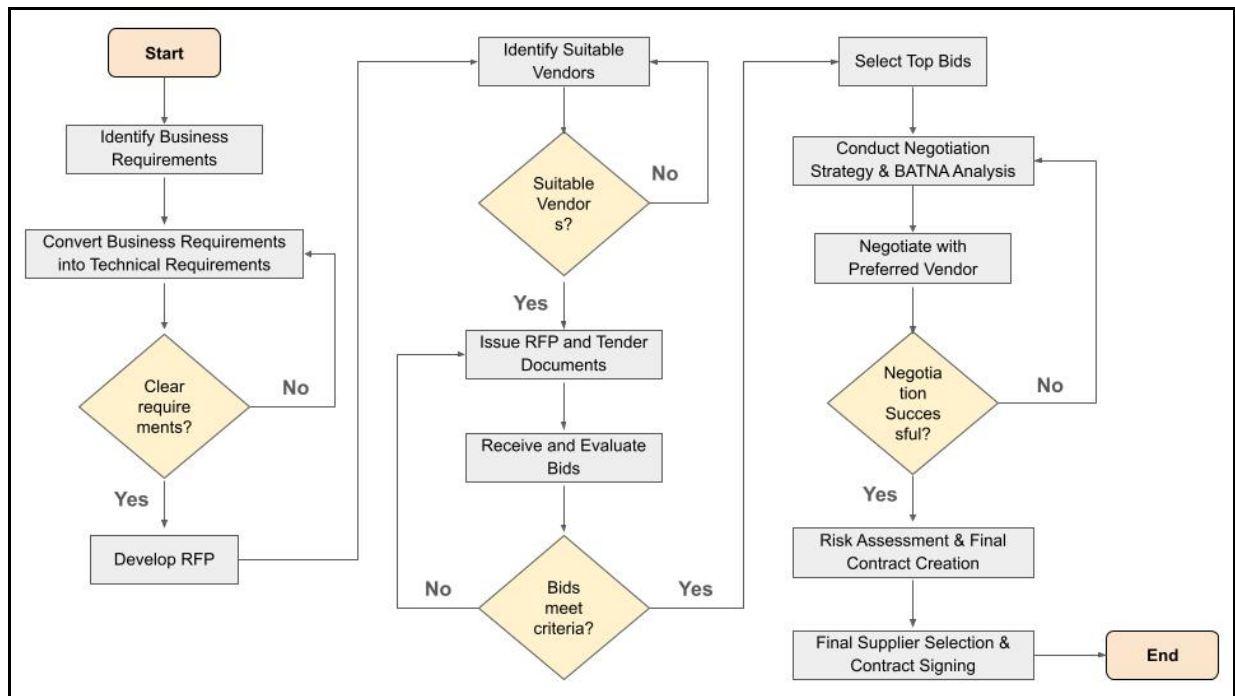
ABHINAV KUMAR – FT251002
ADITI SHARMA – FT251005 ADITYA
GUPTA - FT251006 AMBIKA
PRASAD – FT251013 MAYANK
AGRAWAL – FT201042 KAUSTUV
BHATTACHARYA - FT252043

Automating Procurement at TransGlobal Industries

Objective:

1. Modernize the procurement process at TransGlobal Industries by leveraging advanced technologies such as Large Language Models (LLMs), LangChain, and Streamlit.
2. Reduce manual intervention and streamline procurement operations.
3. Enhance accuracy in requirement translation, RFP generation, vendor selection, bid evaluation, and contract creation.
4. Accelerate decision-making across the procurement lifecycle.

Flow Chart:



Process Details:

1. Identify Business Requirements – Business units define their needs for new servers, software, or technical assets.
2. Convert Business Requirements into Technical Requirements – A detailed specification of functional and non-functional requirements is created.
3. Develop Request for Proposal (RFP) – The technical requirements are formatted into an RFP document for potential suppliers.
4. Identify Suitable Vendors – Pre-approved vendors are evaluated based on selecting top 2 vendors based on weighted average of numerical columns given in the dataset.

5. Issue RFP and Tender Documents – The tender document is sent to shortlisted vendors via formal communication.
6. Receive and Evaluate Bids – Vendor proposals are assessed based on price, quality, timelines, and capabilities.
7. Select Top Bids – The two best proposals are shortlisted for further evaluation.
8. Conduct Negotiation Strategy & BATNA Analysis – The company devises a negotiation strategy based on bid data.
9. Negotiate with Preferred Vendor – Discussions take place to secure favorable terms.
10. Risk Assessment & Final Contract Creation – Supplier risks are evaluated, and a contract is drafted.
11. Final Supplier Selection & Contract Signing – The contract is finalized, ensuring compliance and clarity.

Processes to be Automated:

1. Conversion of Business Requirements into Technical Requirements: Use LLMs to parse business documents and generate structured technical specifications.
2. RFP Generation: Automate drafting of RFPs using LangChain to ensure clarity and consistency.
3. Vendor Selection: Deploy LLM-based agents to analyze technical needs and match with pre-approved vendor profiles.
4. Bid Evaluation: Implement AI tools to score and rank bids based on criteria and select top 2 as the weighted average of numerical columns in the dataset.
5. Negotiation Strategy Development: Utilize LLM insights to simulate negotiation scenarios and recommend strategies.
6. Contract Drafting and Risk Assessment: Automate risk analysis and generate contracts with clear clauses for risk mitigation and performance guarantees.

Why Automate:

1. Minimize Errors: Eliminate mistakes in requirement translation that could lead to costly issues.
2. Ensure Consistency: Standardize RFPs and tender documents, reducing discrepancies.
3. Reduce Bias: Select vendors based purely on data-driven insights rather than human judgment.
4. Speed up Processes: Shorten the time taken for RFP creation, bid evaluation, and contract drafting.
5. Enhance Strategic Agility: Allow faster, more informed decision-making in procurement.

Advantages:

1. Operational Efficiency: Automation streamlines workflows and reduces manual effort.
2. Accuracy and Precision: AI tools minimize human errors and ensure data-driven decisions.
3. Time-to-Decision: Faster processing of RFPs, vendor shortlisting, and bid evaluations.
4. Bias-Free Selection: Objective vendor selection based on past performance and technical fit.
5. Risk Mitigation: Comprehensive, AI-backed risk assessment integrated into contracts.
6. Scalability: The automated system can easily scale with growing procurement demands.

This structured automation approach empowers TransGlobal Industries to modernize their procurement process, leveraging AI to drive efficiency, accuracy, and strategic decision-making.

Implementation:

I. Core Functionality & Structure

- **Objective:** The code implements an AI-driven procurement automation system using Streamlit for the user interface, Langchain for orchestrating the LLM interactions, and Google's Gemini LLM for the core logic.
- **Modular Design:** The system is broken down into distinct steps within the procurement process, from initial requirements gathering to contract generation. Each step is encapsulated within a separate Streamlit expander and uses a dedicated Langchain LLMChain. This promotes maintainability and readability.

II. User Input and Data Flow

1. **User Input:** The Streamlit application allows users to upload files (Business Requirements, Vendor History, Bids) or paste text directly into text areas. This provides flexibility in how data is fed into the system.
2. **Data Orchestration:** The Streamlit application manages the flow of data:
 - Data is read from uploaded files or text areas.
 - This data is then passed as input to the Langchain chains.
 - The output from each chain is stored in Streamlit's session_state for use in subsequent steps and to persist data across user interactions.
 - The final output from each step is displayed in Streamlit's UI, with download options.
3. **Processing Pipeline (LLM Chains):**
 - The code defines a sequence of Langchain LLMChains, each designed for a specific task in the procurement process.
 - Each chain uses a PromptTemplate to structure the input to the Gemini LLM, defining the context, role, task, and action that the LLM should take.
 - The LLM is initialized with a low temperature (0.1) to ensure more consistent and predictable outputs.

III. Key Steps & AI-Driven Analysis

1. **Business to Technical Requirements Conversion:**
 - tech_req_chain: This chain takes business requirements as input and generates a structured technical requirements document. The prompt is carefully crafted to instruct the LLM to create SMART requirements and to avoid incorporating details about the implementation of the agent itself (Langchain, Streamlit, etc.).

2. **Vendor Shortlisting:**

- vendor_shortlist_chain: This chain identifies potential vendors based on technical requirements and vendor history.
- *Implementation Detail:* Vendor scoring and ranking are performed which takes into account delivery punctuality, quality of goods, and contract term compliance. It extracts names of vendors from LLM and performs scoring on those vendors only.

3. **Tender Document & RFP Preparation:**

- tender_doc_chain: Generates a comprehensive tender document and RFP, combining technical and business requirements.

4. **Tender Email Generation:**

- tender_email_chain: Creates a professional email to send the tender document to shortlisted vendors. The prompt specifies the email's format and content, ensuring clarity and professionalism.

5. **Bid Evaluation:**

- bid_evaluation_chain: Evaluates bids based on price, quality, delivery, and technological capability, identifying the top two bids.

6. **Negotiation Strategy & BATNA Analysis:**

- negotiation_strategy_chain: Develops a negotiation strategy and identifies the Best Alternative to a Negotiated Agreement (BATNA) based on the top two bids.

7. **Risk Assessment Report Generation:**

- risk_assessment_chain: Generates a risk assessment report for the preferred vendor, covering delivery, quality, compliance, performance, and communication risks.

8. **Contract Document Generation:**

- contract_doc_chain: Drafts a comprehensive contract document, including clauses on risk mitigation, performance guarantees, and dispute resolution.

IV. UI Visualization (Streamlit)

- **Organized Layout:** The Streamlit UI is well-structured, using a sidebar to list procurement steps and expanders to contain the input and output sections for each step.
- **Clear Instructions:** The UI provides clear instructions to the user on how to upload files or enter text.
- **Download Buttons:** Download buttons are provided for each generated document, allowing users to save the results.
- **Error Handling:** The code incorporates basic error handling (e.g., checking for missing API keys or business requirements). More robust error handling within each step's processing would improve the user experience.

V. Technology Stack

- **Key Technologies:**
 - **Streamlit:** For creating the interactive web application and managing user input and output.
 - **Langchain:** For building and orchestrating the LLM chains.
 - **Gemini-2.0-flash:** The LLM that powers the core logic of the system.
 - **Pandas:** Used to handle data from the uploaded Vendor History and Bids CSV files.

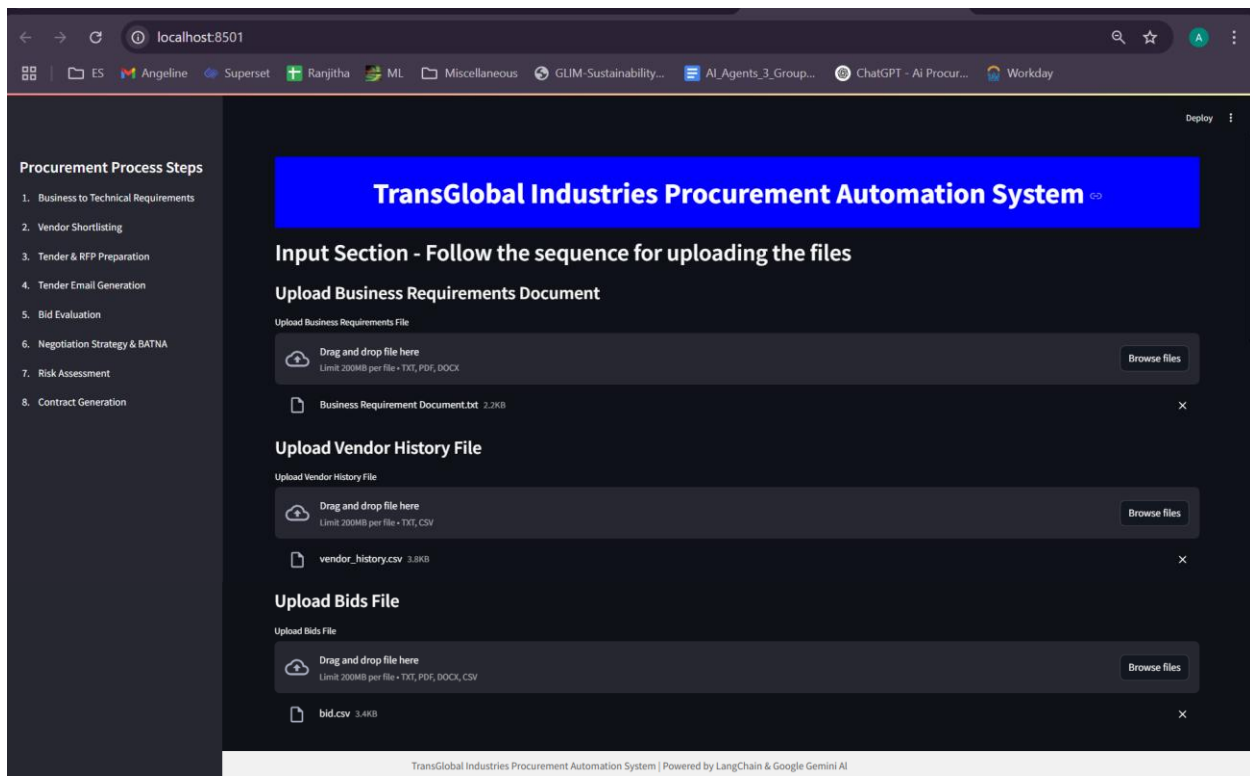
VI. Key Considerations and Potential Improvements

- **Error Handling:** Implement more robust error handling within each processing step, providing informative messages to the user when errors occur.
- **Input Validation:** Add input validation to ensure that uploaded files are in the correct format and contain the required data. Also, more comprehensively check the dataframes used for scoring and ranking.
- **Asynchronous Processing:** For larger inputs or more complex chains, consider using asynchronous processing to prevent the UI from freezing while the LLM is running.
- **Modularity:** While the code is already modular, consider further breaking down the LLM prompts into smaller, reusable components.
- **Security:** If this application is deployed in a production environment, ensure that appropriate security measures are in place to protect sensitive data.

Final output and functional details:

Step 1: Initial Setup & Configuration

1. Library Import:
 - Uses libraries like `streamlit` for UI, `langchain` for LLM processing, `pandas` for data handling, and `google.generativeai` for AI interactions.
 2. API Key Loading:
 - Loads the Google API key from environment variables to use Gemini AI models.
 - Stops the script if the key is missing.
 3. LLM Initialization:
 - Configures a low-temperature (0.1) `ChatGoogleGenerativeAI` model to ensure deterministic responses.
-



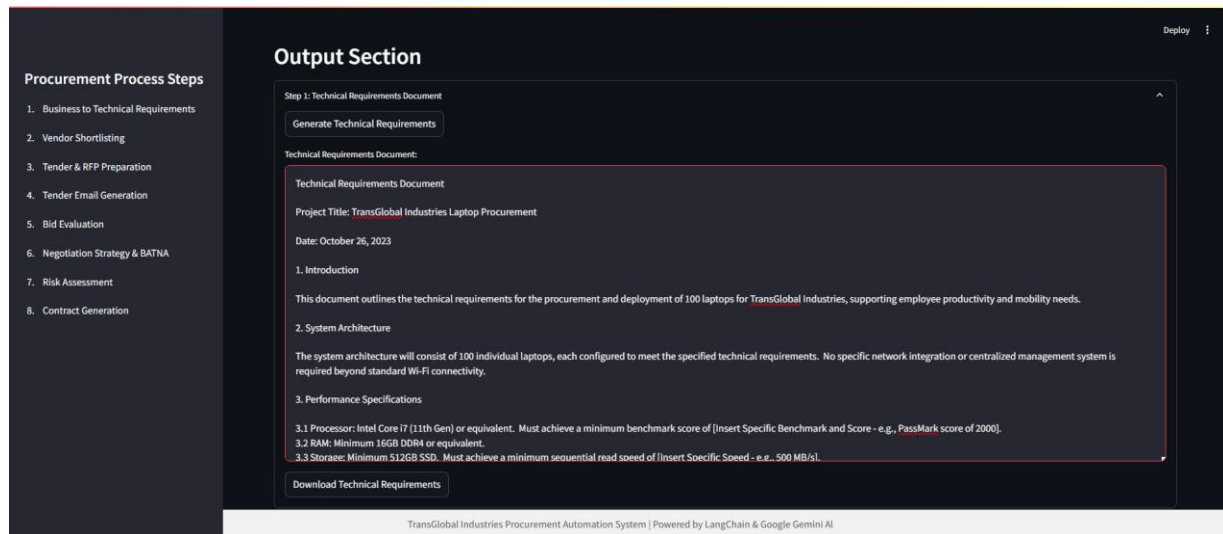
Step 2: Business to Technical Requirements Conversion (Step 1 of Procurement)

1. Prompt Template Definition:

- Uses **PromptTemplate** to define how business requirements are converted to technical requirements.
- Ensures the output is structured, precise, and includes SMART (Specific, Measurable, Achievable, Relevant, Time-bound) criteria.

2. Execution:

- Uses **LLMChain** to connect the prompt with the model.
- Generates a technical requirements document from the provided business requirements.



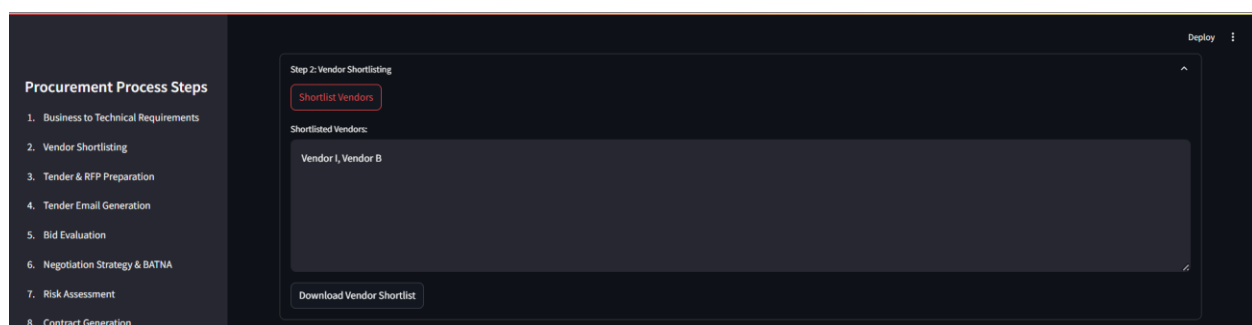
Step 3: Vendor Shortlisting (Step 2 of Procurement)

1. Prompt Template for Vendor Identification:

- Uses the technical requirements and vendor history to identify suitable vendors.
- Returns a plain-text list of vendors.

2. Composite Scoring:

- Reads vendor data from an uploaded CSV or text file.
- Calculates a composite score based on delivery punctuality, quality of goods, and contract compliance.
- Sorts vendors based on their scores and selects the top two.

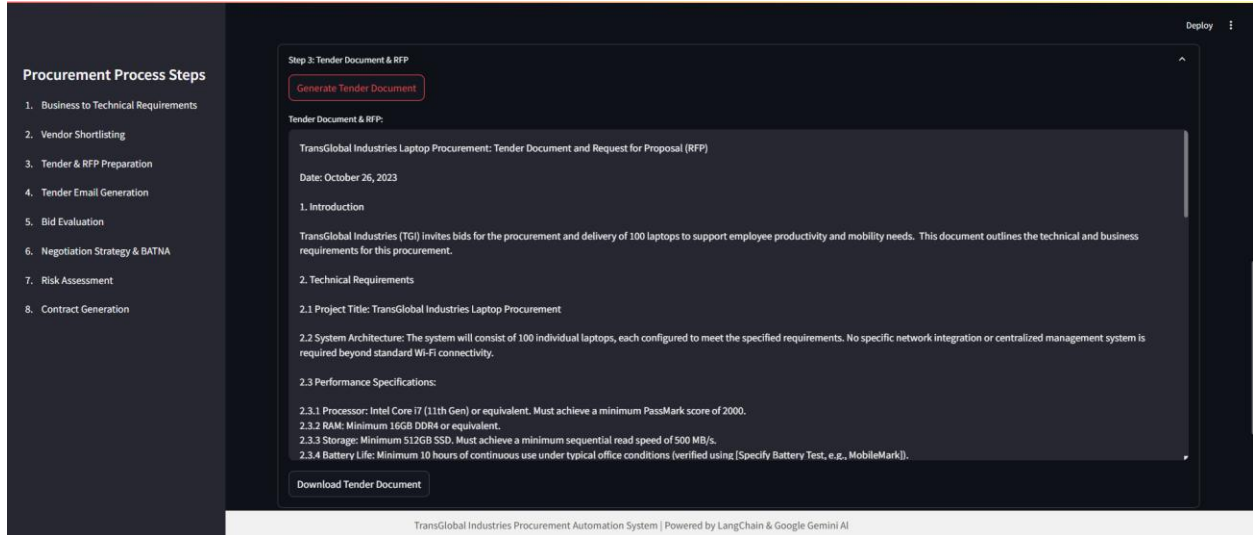


Step 4: Tender Document & RFP Generation (Step 3 of Procurement)

1. Tender Document Generation:

- Combines technical and business requirements to generate a comprehensive tender document.

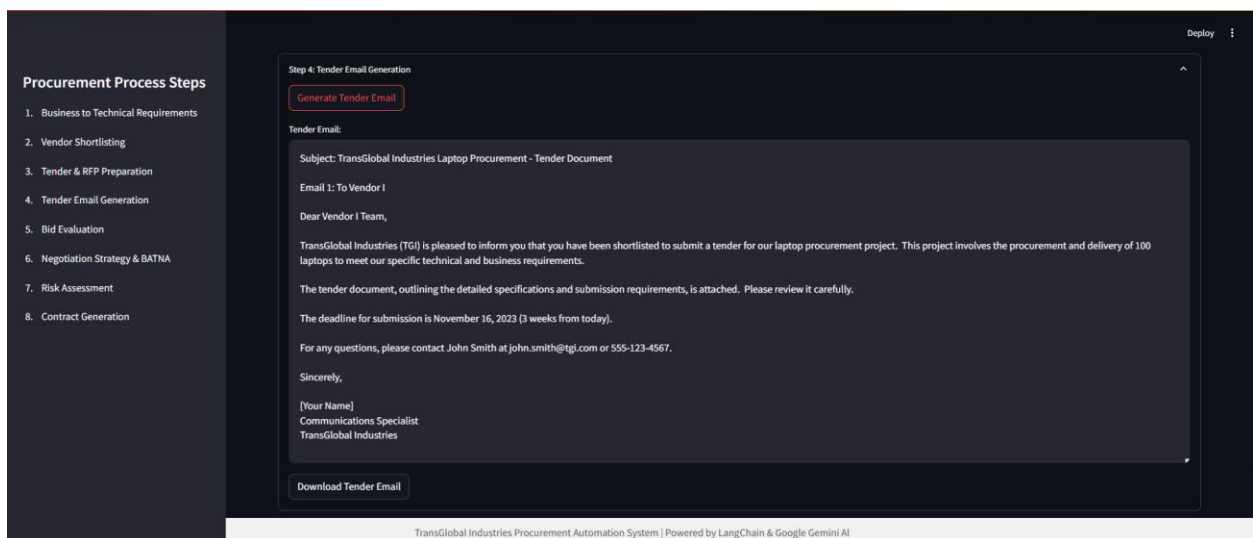
- Uses a structured format to include essential procurement details.



Step 5: Tender Email Generation (Step 4 of Procurement)

1. Tender Email Template:

- Creates a professional email to send the tender and RFP to shortlisted vendors.
- Includes:
 - Subject line.
 - Company introduction.
 - Mention of shortlisting.
 - Deadline for submission.
 - Contact information.



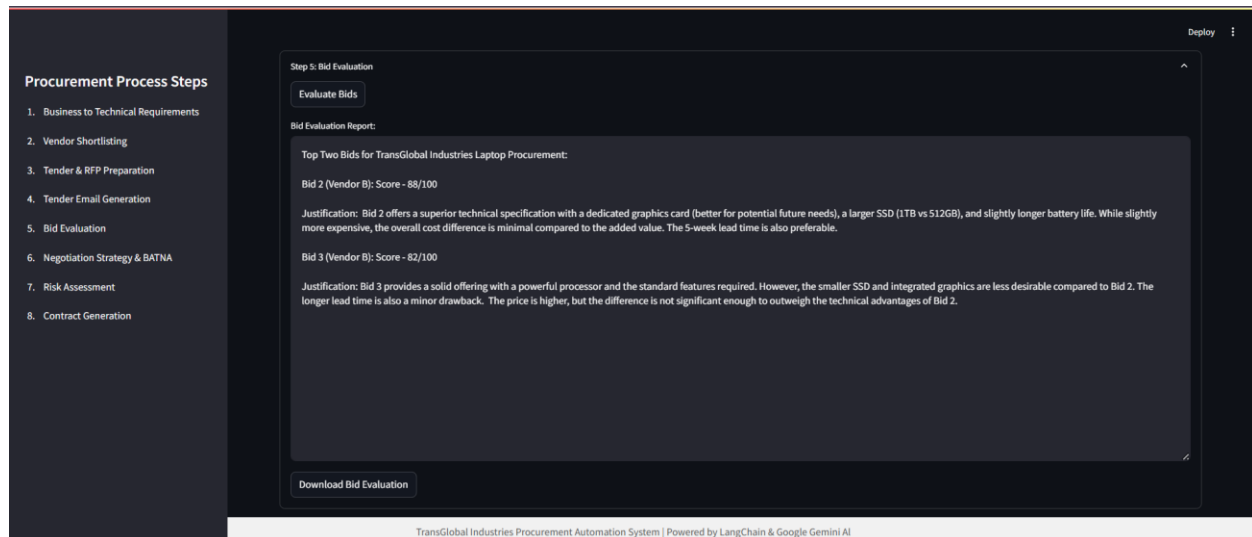
Step 6: Bid Evaluation (Step 5 of Procurement)

1. Bid Data Extraction:

- Filters and extracts bids from shortlisted vendors using LLM.
- Generates a plain-text list of relevant bids.

2. Bid Evaluation:

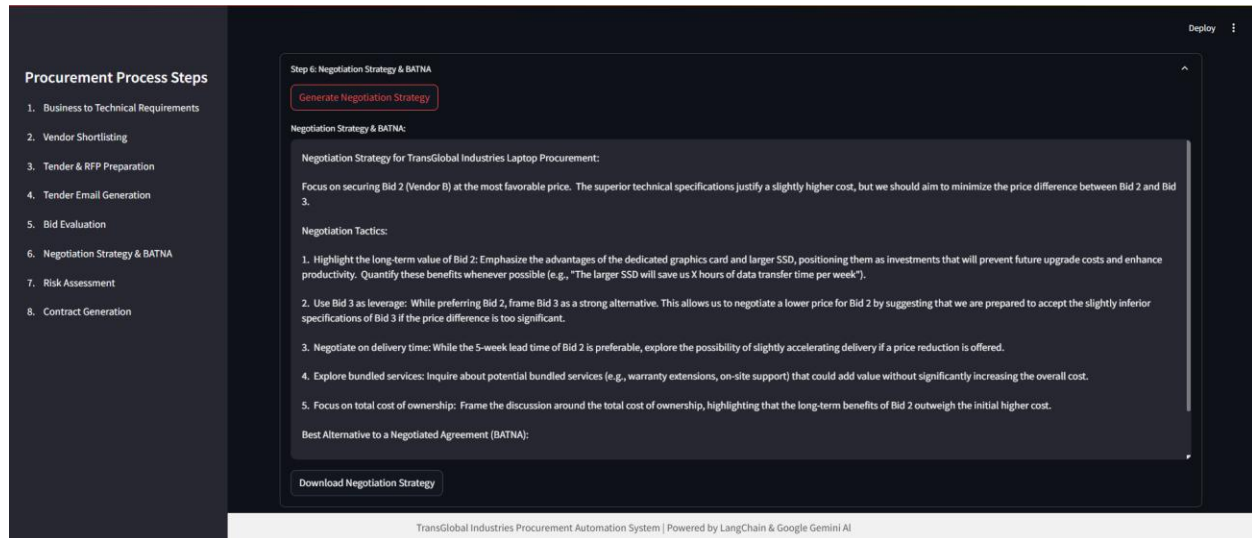
- Uses scoring metrics to evaluate the bids on price, quality, delivery, and technology.
- Selects the top two bids based on the evaluation criteria.



Step 7: Negotiation Strategy & BATNA (Step 6 of Procurement)

1. Negotiation Strategy Prompt:

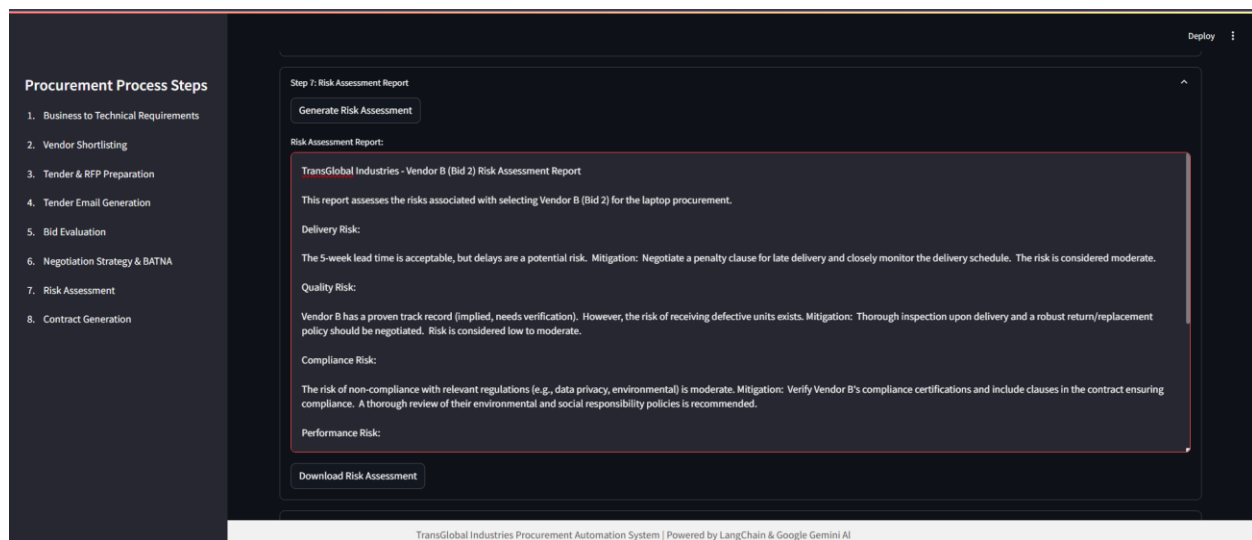
- Uses the top two bids to formulate a negotiation strategy.
- Identifies the Best Alternative to a Negotiated Agreement (BATNA).
- Generates strategic recommendations.



Step 8: Risk Assessment Report Generation (Step 7 of Procurement)

1. Risk Assessment Prompt:

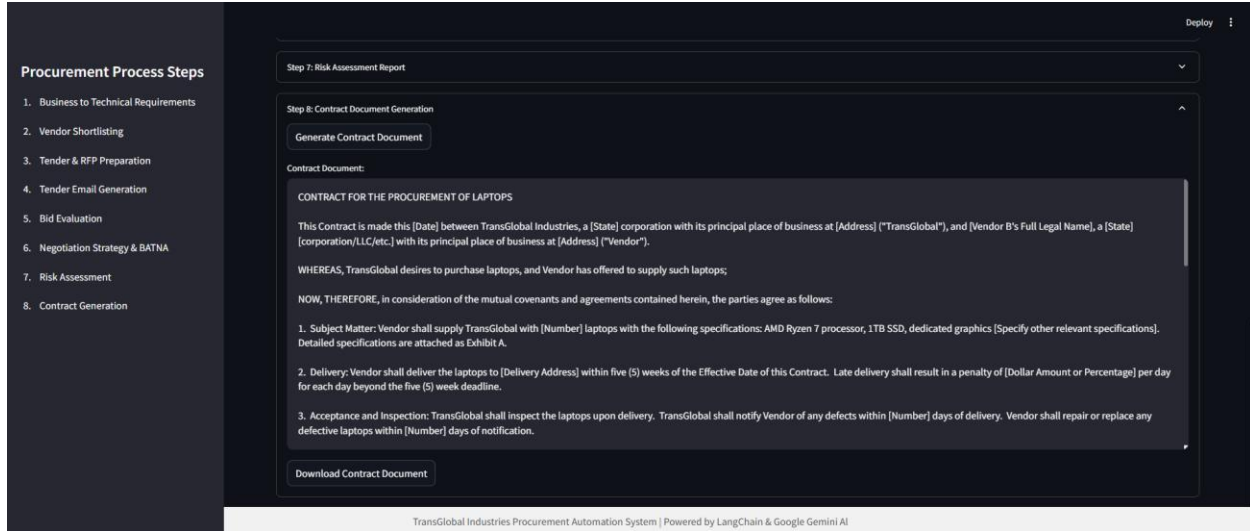
- Analyzes the negotiation strategy and selected bids.
- Evaluates risks related to:
 - a. Delivery.
 - b. Quality.
 - c. Compliance.
 - d. Performance.
 - e. Communication.



Step 9: Contract Document Generation (Step 8 of Procurement)

1. Contract Drafting:

- Uses the risk assessment report to draft a detailed contract.
- Includes clauses on risk mitigation, performance guarantees, and dispute resolution.



Step 10: User Interface (UI) and Interaction

1. Streamlit UI:

- Displays each step of the procurement process.
- Allows users to upload necessary documents and generate outputs through buttons.
- Provides download buttons for each generated document (technical requirements, vendor shortlist, tender documents, bid evaluation, etc.).

2. Session State Management:

- Uses Streamlit's session state to store intermediate results between steps.