**Insurance Quote Comparison Application**

**[First create the Front-end UI and then specify what credentials are needed to connect and install the Back-end to Supabase and run the application]**

Create a black and white and grey application with front-end UI for comparing insurance quotes. (Powered by Randizone)[link www.randizone.co.za]

The user uploads various insurance quotes in PDF format on the Application which is handled in Supabase server.

Since insurance documents often use varied terminology,

Setup a sophisticated backend with **LLMWhisperer, include Layout Preserving Mode, Auto-switching between native text and OCR modes** (for scanned PDFs) and advanced Natural Language Processing (NLP) to accurately extract, and normalize data from diverse PDF layouts which include tables.

Create a React application that shows the comparison. Extract the key information from the PDFs and format it into a structured JSON.

This application, with the use of AI Agents, will then take that structured data and display a clear, comparative view, highlighting benefits, positives, and potential risks or limitations.

In order to use this application, the user must register and sign in:

There is a client-user area and an Admin-user Area.

The client-user profile area:

* This area allows the user to edit his profile, enter the Company name, Advisor name, Advisor contact details. This will show on the Heading of the Quote Report and change the footer details of the quote report.
* Make changes to their subscription. . [There are 3 Subscription options, Test Drive: Free for the first 2 reports, Monthly Standard for 25 reports per month, Monthly Professional for 50 reports per month. Via PayFast , They can pause, lower and cancel their subscription
* Request invoices.

The Admin Area has a higher-level user function which cannot be accessed by the client-user and registration, and access is via admin page link: https://www.samplewebsitedomain.co.za**/admin**

The admin area allows a section to set the email address smtp server. The details of this email settings will allow the sending of registration details and to allow the user to also send the comparison in pdf after the user has done comparisons if they don’t want to download the report.

SMTP [set up smtp with these details as a start:]

* Secure SSL Settings
* Username: no-reply@randizone.co.za
* Password: L-y[9(Gl?%&Ipwek
* Incoming Server: mail.randizone.co.za
* IMAP Port: 993 POP3 Port: 995
* Outgoing Server: mail.randizone.co.za
* SMTP Port: 465

The admin area also allows for input or changes of API connection which might be used or Tokens, database links, webhooks etc. [Give clear instructions if and where tokens are needed for example **Secrets**: store MOONSHOT\_API\_KEY, GENSPARK\_API\_KEY, AWS\_ACCESS\_KEY\_ID etc.]

Make sure all details for Register and login on the front-end UI correspond with the Server code.

How it works: 1. Data Structure: The quotes hold an array of insurance plans.

Each quote has a:

**Company**, (example: Santam)

**Plan Name**: (They select one option when they upload quotes for comparison: Commercial Short Term or Motor Personal, Life Insurance, Investments)

**Premium**: (This is the Total, final, monthly debit order amount including VAT. It includes the SASRIA and any added amounts and appears to the end of the summery)

The Reports has a Main Section and Sub-sections that define its category (e.g., Motor Insurance, Building Cover, Home content, etc. or Life Cover, Disability Cover, Severe Illness, Income protection etc),

Dynamic Table Generation: The Main Section categories and features from the loaded quotes and then dynamically generates a comparison table.( See Table samples and Quote Report sample). The Report consists of an overview Quote Company, Policy Plan and Premium summery, followed by a Main Sections Summery, followed by a Detailed Sub-Section comparison which includes all Main Section with Sub-Sections and details from each subsection. (Since there is no standard format for quote Sub-Categories, most of the Sub-Category’s details will have to be Extracting tables from PDFs accurately and preserving their structure while plotting comparison tables in the Sub-Section Template Detailed report area WITHOUT changing or editing the details. The Report can then be viewed in the Application in three tabs: Policy Summery, Section Summery (Main Category summery), Detailed Summery (Sub-Category detailed summer). The user can then also choose to download or email the report. The report can be in Black/White/Grey tones to differentiate between the quotes for comparisons.

Technical considerations:

Understanding the data in order to extract and place it accurately is important.

Extractions: Consider using RAG, Intelligent Document Processing (IDP) Solutions or Advanced Libraries for Extraction and include **LLMWhisperer HTTP request plus webhook,** (→ returns row/column JSON automatically and possibly combined with Workflow → Kimi K2 Agents from Moonshot api → 50 sub-agents query Postgres via GPT-4-turbo SQL optimiser.

Add **img2table as a fallback:** This tool converts PDF pages to images and then uses OCR to extract tables, capable of generating Pandas DataFrames, even for merged cells. combining these:

**Comparison Mechanisms:**

* **Programmatic Comparison:** For comparing specific numerical values or fields within tables (e.g., insurance premiums, coverage limits), write direct comparison logic against the structured data (JSON, DataFrames, database records).
* **NLP for Textual Clauses:** For less structured textual components or clauses in insurance documents, Natural Language Processing (NLP) techniques can be used for text matching and comparison. Some research suggests simpler statistical models like TF (Term Frequency) or TF-IDF (Term Frequency-Inverse Document Frequency) might perform better for matching document sections when training data is limited or categories are not predefined.
* **LLM for Semantic and LLMWhisperer, Comparison:** For nuanced semantic comparisons of clauses, you can leverage LLMs, potentially with few-shot learning, to identify similarities or differences beyond exact matches.
* **Section-Level Comparison:** Breaking down the comparison to Main/Subsection-level rather than comparing entire documents can significantly improve accuracy and relevance.
* **Structure Multi-Agent Systems for Document Comparison:**
* You would typically use an **orchestrator** or **manager agent** to coordinate the workflow among various specialized agents.

**Document Ingestor/Extractor Agent:**

* **Role:** Takes raw PDF documents.
* **Tools:** **LLMWhisperer,** for Markdown conversion. Use **img2table as a fallback** to **split, pre-process and detect if a page is image-based**.

**LLMWhisperer** **OCR** (FeatureTypes=["TABLES","FORMS"]) → returns row/column JSON automatically and possibly combined with Workflow → 50 sub-agents query Postgres via Moonshot api

Sub-Agents can be named for each Main Section for example:

1. Policy sections Sub-Agent
2. Fire Sub-Agent
3. Buildings combined Sub-Agent
4. Office contents Sub-Agent…etc.

**This will allow for easier reference and training of agents in those specific sections.**

Business interruption

Use **img2table** only for pages that contain actual tables in image format as fall back

* **Output:** Structured Markdown content for each document, with initial Sectioning and Sub-sectioning tags.
* **RAG Preparation Agent:**
* **Role:** Processes the Markdown for efficient retrieval.
* **Tools:** **LLMWhisperer** , LlamaIndex's MarkdownNodeParser for intelligent chunking, Embedding models (HuggingFaceEmbedding), Vector store client (e.g., ChromaDB).
* **Output:** Populated vector database with indexed document chunks and metadata.
* **Summery, Main-Section Table Analysis Agent:**
* **Role:** Focuses exclusively on extracting and comparing tables in sub-sections.
* **Tools:** Custom Python functions (parse\_markdown\_table, compare\_tables).
* **Output:** Detailed programmatic differences found in tables (e.g., "Row X in Document A has Y, but in Document B it has Z").
* **Sub-Section Specialist Table Analysis Agent :**
* **Role:** Focuses exclusively on extracting and comparing tables in Sub-Sections.
* **Tools:** Custom Python functions (parse\_markdown\_table, compare\_tables, use exact sub-category table structure from that quote pdf when complex comparison).
* **Output:** Detailed programmatic and understanding differences found in tables and semantics and context of the specific sub-sections (e.g., "Row X in Document A has Y, but in Document B it has Z") and that specific sub-section.
* **Textual Clause Comparison Agent (NLP/Semantic Analyzer):**
* **Role:** Compares the non-tabular textual clauses and sections semantically.
* **Tools:** Embedding models, cosine similarity calculations.
* **Output:** Semantic similarity scores for sections, flags for potential significant differences.
* **Report Generation Agent: Genspark Agent API**
* **Role:** Gathers output from all other agents and synthesizes a comprehensive, human-readable well designed comparison report following report output guidelines.
* **Tools:** LLM for natural language generation, templating engine.
* **Output:** The final comparison report.
* **Orchestrator/Manager Agent:**
* **Role:** Oversees the entire workflow. Assigns tasks to specialized agents, manages data flow between them, handles errors, and decides on the next steps based on intermediate results.
* **Logic:** Uses conditional logic (e.g., "If Table Analysis Agent finds a critical difference, pass it to Legal Interpretation Agent for detailed explanation").
* **Example Workflow with Agents:**
* **Orchestrator** receives two insurance PDF paths.
* **Orchestrator** sends both PDFs to **Document Ingestor/Extractor Agent**.
* **Orchestrator** then triggers the first 10 parallel tasks:
* Sends relevant Markdown sections (containing tables) and textual sections to Section and Sub-Section Agents.
* **Orchestrator** receives aggregated findings and triggers the following parallel workflow until all is completed.
* Finally, the **Orchestrator** sends all aggregated findings to the **Report Generation Agent** to compile the final comparison document.

3. If a feature is explicitly stated as "Not covered" or "Not included" in its description, the font will be highlighted in red.

4. Search Functionality: You can use the search bar to filter the displayed Main categories and sub-categories details, making it easier to find specific aspects of the policies.

5. The entire list of Main Section items should be in the Summery for Main Sections.

**Your output must:**

1. Extract key data points from each quote or policy.
2. Categorize them into standard sections and sub-sections.
3. Identify all material disclosures and compliance requirements. (Location for legal reference, South Africa)
4. Generate a clear side-by-side comparison table.
5. Highlight key differences, advantages, and risks.
6. Provide a short summary of the best option based on the facts (not a final recommendation — the human broker will give final advice).

**🗂️ Use the following standard sections and sub-sections for data mapping:**

**1️⃣ General Insurer Information**

* Insurer name
* Product name

**2️⃣ Premium & Fees**

* Monthly premium, Total Monthly Payable (incl. VAT) (Including SASRIA and VAS)
* Commission %
* Broker fees / Admin fees / Advisor fees

**3️⃣ Main Sections for (Plan Name: Commercial Short Term Insurance only –[** Motor Personal short term, Life Insurance and Investments will be updated on the next update. **)**

1. Policy sections
2. Fire
3. Buildings combined
4. Office contents
5. Business interruption
6. General
7. Theft
8. Money
9. Glass
10. Fidelity guarantee
11. Goods in transit
12. Business all risks
13. Accidental damage
14. Public liability
15. Employers' liability
16. Stated benefits
17. Group personal accident
18. Motor personal accident
19. Motor General
20. Motor Specific
21. Motor Fleet
22. Electronic equipment
23. Umbrella liability
24. Accounts receivable
25. Accidental Damage
26. Employers' Liability
27. Group Personal Accident
28. Motor Industry Risks
29. Houseowners
30. Machinery Breakdown
31. Householders
32. Personal, All Risks
33. Watercraft
34. Personal Legal Liability
35. Deterioration of Stock
36. Personal Umbrella Liability
37. Greens and Irrigation Systems
38. Commercial Umbrella Liability
39. Professional Indemnity
40. Cyber
41. Assist/Value services/ VAS
42. SASRIA
43. Events Liability Insurance
44. Directors & Officers Liability
45. Agri insurance
46. Engineering insurance
47. Community insurance

**4️⃣ Excess / Deductibles for Main Section and Various Sub-sections**

* Peril-specific excess amounts
* Total excess payable
* Sliding scale % or flat excess

**5️⃣ Conditions & Warranties for Main Section and Various Sub-sections**

* Security requirements (e.g., alarms, tracking devices)
* Fire extinguishing warranties
* Maintenance conditions
* Any suspensive conditions?

**6️⃣ Exclusions for Main Section and Various Sub-sections**

* Notable standard exclusions
* Notable special exclusions

**7️⃣ Claims Process (if available)**

* Contact details
* Timeframes
* Any unusual restrictions?

**8️⃣ Cooling-Off & Cancellation (if available)**

* Applicable or not
* Notice periods

**9️⃣ Compliance Disclosures ( User enters details via profile area)**

* FSP name and license number
* Conflicts of interest (if mentioned)

**🔟 Other Material Terms**

* Anything unique to this policy
* Any compliance triggers

**🗃️ Key Additional Instructions**

* Identify any conflicting terms or gaps in cover.
* Flag unusual clauses or unfair terms.
* Highlight differences, if any, in:
  + Sum insured levels
  + Excesses
  + Warranties
  + Exclusions
  + Special conditions
  + Premium differences
  + Service levels orclaims reputation (if stated)

**✏️ Output Format**

1. **Structured Comparison Table**

**See the comparison example file for guidance:**

**B) Key Disclosures Summary**

* List each material disclosure per quote/policy.
* Note any non-disclosures or missing info.

1. Observability – add “Error Trigger” workflows for retries & email alerts.

──────────────────────────────────────── SAMPLE OUTPUT (markdown snippet) ────────────────────────────────────────

Refine the SAMPLE OUTPUT (markdown snippet) by using the below and attached Report sample:

Below is the **markdown output** that now conforms **exactly** to the two layouts you requested.

### 1️⃣ Main Policy Section Summary

*(one line per policy section)*

| Section | Section applicable | Sum Insured | Monthly premium incl VAT |
| --- | --- | --- | --- |
| Fire | Y | Y | Y | R4 500 000 | R7 771 809 | R10 000 000 | R257.36 | R971.45 | R653.54 |
| Buildings Combined | Y | Y | Y | R1 155 000 | R7 771 809 | — | R244.95 | R971.45 | R653.54 |
| Office Contents | N | Y | Y | — | R80 990 | R20 247 | — | R67.49 | R50.62 |
| Business Interruption | N | N | Y | — | — | — | — | — | R125.50 |
| Theft | Y | Y | Y | — | R20 000 | — | R4.23 | R83.33 | R2 490.84 |
| Money | Y | Y | Y | — | R19 965 | — | R5.00 | R5.00 | R14 623.61 |
| Glass | N | Y | Y | — | R10 000 | — | — | R83.19 | R180.16 |
| Fidelity Guarantee | N | Y | Y | — | R325 000 | — | — | R316.66 | R6 548.85 |
| Goods in Transit | N | N | Y | — | — | — | — | — | R13.00 |
| Business All Risks | N | Y | Y | — | R5 000 000 | — | — | R125.00 | R85.94 |
| Accidental Damage | N | Y | Y | — | R5 000 000 | — | — | R19.44 | R256.53 |
| Public Liability | Y | Y | Y | — | R5 000 000 | — | R316.66 | R234.07 | R2 200.27 |
| Employers’ Liability | N | Y | Y | — | — | — | — | Included | Included |
| Stated Benefits | N | N | Y | — | — | — | — | — | Included |
| Group Personal Accident | Y | Y | Y | — | — | — | Included | Included | Included |
| Motor Personal Accident | Y | N | Y | — | — | — | Included | — | Included |
| Motor General | Y | N | Y | — | — | — | Included | — | Included |
| Motor Specific/Specified | Y | N | Y | — | — | R403 809 | Included | — | R3 419.33 |
| Electronic Equipment | N | N | Y | — | — | — | — | — | Included |
| Umbrella Liability | N | N | Y | — | — | — | — | — | Included |
| SASRIA | Y | Y | Y | — | — | — | R229.07 | R99.38 | Included |

Each cell shows **Quote 1 | Quote 2 | Quote 3** values.

### 2️⃣ Detailed Comparison Tables

*(one table per policy section; repeat rows for each sub-section)*

#### **Fire & Allied Perils**

| Quote nr | Description of SUB-Sections Info | Sum Insured | Included YES/NO | Premium |
| --- | --- | --- | --- | --- |
| Quote 1 | See quote report template, Fire, lightning, explosion, earthquake, storm, impact, riot & strike, malicious damage, subsidence & landslip | R4 500 000 | YES | R257.36 |
| Quote 2 | See quote report template Fire, lightning, explosion, earthquake, storm, impact, riot & strike, malicious damage, subsidence & landslip | R7 771 809 | YES | R971.45 |
| Quote 3 | See quote report template Fire, lightning, thunderbolt, subterranean fire, explosion, storm, earthquake, aircraft, impact, theft | R10 000 000 | YES | R653.54 |

#### **Buildings Combined**

| Quote nr | Description of SUB-Sections Info | Sum Insured | Included YES/NO | Premium |
| --- | --- | --- | --- | --- |
| Quote 1 | PROPERTY:  1.Fire, lightning, thunderbolt, subterranean fire, explosion.  2. Storm, wind, water, hail or snow other than:  a. that arising from its undergoing any process necessarily involving the use or application of water;  b. wear and tear or gradual deterioration;  c. loss or damage:  i. to retaining walls;  ii. caused or aggravated by:  – subsidence or landslip;  – the Insured’s failure to take all reasonable precautions for the maintenance and safety of  the property insured and for the minimisation of any destruction or damage.  3. Earthquake.  4. Aircraft and other aerial devices or articles dropped therefrom.  5. Impact by animals, trees, aerials, satellite dishes or vehicles excluding damage to such animals, trees, aerials, satellite dishes, vehicles or property in or on such vehicles.  6. Theft (or any attempt thereat) accompanied by forcible and violent entry into or exit from such building. If any building insured or containing the insured property becomes unoccupied for 30 consecutive days, this item is suspended as regards the property affected unless the Insured before the occurrence of damage obtains the written agreement of the Company to continue this extension. During the period of the initial unoccupancy of 30 consecutive days the Insured shall become a co-insurer with the Company and shall bear a rateable proportion of any damage equal to % of the claim before deduction of any first amount  payable.  7. Accidental damage to sanitary ware, but the amount payable will be reduced by R250 for each and every such damage.  8. Power surge: Electrical power surge  PUBLIC SUPPLY:  RENT:  LIABILITY:   * Limit * Special exemption   CLAUSES AND EXTENSIONS:  DEDUCTABLES/FIRST AMOUNT PAYABLE/  Risk Address:  Excess / First amount payable / Deductibles | R1 155 000 | YES | R244.95 |
| Quote 2 | Public supply connections, rent 25 %, architects’ fees 20 %, demolition, temporary removal, locks & keys | R7 771 809 | YES | R971.45 |
| Quote 3 | Fire, lightning, storm, earthquake, impact, theft, malicious damage, subsidence & landslip | — | YES | R653.54 |

#### **Office Contents**

| Quote nr | Description of SUB-Sections Info | Sum Insured | Included YES/NO | Premium |
| --- | --- | --- | --- | --- |
| Quote 1 | — | — | NO | — |
| Quote 2 | Contents + rent + documents + legal liability docs + increased cost of working + SASRIA | R80 990 | YES | R67.49 |
| Quote 3 | Defined events, average clause, malicious damage, riot & strike, theft extension | R20 247 | YES | R50.62 |

*See Quote Report template:*

*(Repeat the same* ***Detailed Comparison*** *structure for every remaining section—Theft, Money, Glass, Fidelity Guarantee, Goods in Transit, Business All Risks, Accidental Damage, Public Liability, Employers’ Liability, Stated Benefits, Group Personal Accident, Motor Personal Accident, Motor General, Motor Specific, Electronic Equipment, Umbrella Liability, SASRIA, etc.—each with its own sub-section rows.)*

Developer Notes:

* **Make sure to add a separate “type” file and make sure all files refer to the correct paths and information needed.**
* **Take special care to avoid syntax errors.**
* **Load all required tools and database requirements.**

**NB. Check any additional requirements and possible improvements and security requirements.**

**Back-end using Railway:**

**1. Stage 1 – Textract: PDF → JSON**

* **Input**: PDF(s) land in an S3 bucket (single- or multi-page).
* **Action**:
  1. Trigger an AWS Lambda (or Step Function) on the S3 PUT event.
  2. Lambda calls **LLMWhispere API** (start\_document\_analysis) with FeatureTypes=["TABLES","FORMS"] .
  3. Poll get\_document\_analysis until JobStatus == "SUCCEEDED".
  4. Parse the Block array into a compact JSON of **text, key-value pairs, and tables**.
  5. Save the JSON to a **staging bucket** or DynamoDB table.

**2. Stage 2 – Orchestrator + 10x Worker Agents**

* **Input**: the JSON produced above.
* **Action**:
  1. A lightweight FastAPI / Lambda orchestrator **splits** the JSON into 10 logical slices
* The Plan, Company and Premium details and The Main Category Summery details,
* Then the first 9 Sub-Categories
* Then the next 10 Sub-Categories
* Then the next 10 Sub-Categories etc..
  1. Fire **10 parallel async calls** to the **Moonshot AI API** (Kimi) asking each worker to enrich its slice with reasoning, classification, or additional lookups (you can stuff the JSON chunk into the prompt or send a pre-signed S3 URL).
  2. Wait with asyncio.gather (Python) or Promise.all (Node).

**3. Stage 3 – Genspark Agent API: Final Report**

* **Input**: the 10 enriched results.
* **Action**:
  1. Concatenate the 10 outputs into a single prompt, or send them as structured JSON.
  2. POST to the **Genspark Agent API** endpoint (https://api.genspark.ai/v1/agent/execute or the latest URL from their docs) with your agent’s agent\_id and the payload.
  3. Genspark returns the polished report (markdown, HTML, or PDF).
  4. Persist the report to S3 / DynamoDB / send via webhook. **Railway Storage + Postgres Row** or direct S3

| Step | Command / UI |

| ---- | ---------------------------------------------------------------------------- |

| 1 | Create project → \*\*Railway Dashboard\*\* → \*\*New Project\*\*. |

| 2 | \*\*Database → Extensions\*\* → enable `pg\_stat\_statements`, `uuid-ossp`. |

| 3 | \*\*Settings → Database\*\* → copy `DATABASE\_URL` into your `.env`. |

| 4 | \*\*Edge Functions\*\* tab → `Railway functions new orchestrator` |

| 5 | CLI deploy: `Railway functions deploy orchestrator` |

| 6 | \*\*Storage\*\* → create bucket `reports`, set public or private policy. |

| 7 | Secrets: `Railway secrets set MOONSHOT\_API\_KEY=sk-xxx GENSPARK\_API\_KEY=xxx` |