SMART INDIA HACKATHON 2025 – PROJECT SUBMISSION

# Project Title: PaperSafe – AI-Powered Document Intelligence Platform

Team ID: (to be filled)  
Theme: Cyber Security / Digital Education / Data Management  
Proposed by: Arul Selvam P & Team

# 1. Problem Statement

Government, research, and educational institutions handle thousands of documents, reports, and research papers.   
Searching, summarizing, and verifying facts within PDFs is still done manually — consuming time and risking human error.

Challenges faced:  
• Large volumes of PDF data  
• No intelligent system for document Q&A  
• Manual verification  
• Inefficient collaboration

# 2. Proposed Solution: PaperSafe

PaperSafe is an AI-powered document intelligence platform using LangChain, FAISS, and Groq-hosted LLaMA 3.3 (70B).  
The platform enables fast, accurate, and explainable PDF Q&A through semantic search and LLM reasoning.

|  |  |
| --- | --- |
| Feature | Description |
| Intelligent PDF Q&A | Ask natural questions to extract answers from PDFs |
| Summarization | Auto-generate concise document summaries |
| Fact Grounding | Verifies that answers exist within text |
| Secure API Integration | No hardcoded keys – uses .env |
| History & Analytics | Stores Q&A history with grounding scores |
| Cloud Deployment | Streamlit Cloud for easy access |

# 3. Technical Architecture

Tech Stack:  
• Frontend: Streamlit  
• Backend: Python (LangChain, FAISS)  
• LLM: Groq-hosted LLaMA 3.3 (70B)  
• Embedding Model: all-MiniLM-L6-v2  
• Storage: FAISS Index

Workflow: PDF → Chunking → Embedding → Retrieval → LLaMA → Answer

📸 Screenshot placeholders:  
1. Streamlit UI  
2. FAISS Index Creation  
3. Answer Retrieval

# 4. Implementation Steps

1. Setup environment and install dependencies.  
2. Configure `.env` with API credentials (exclude from GitHub).  
3. Build Streamlit app (upload, embed, query).  
4. Push to GitHub and deploy on Streamlit Cloud.  
5. Query PDFs and visualize answers.

# 5. Sample Queries

• What is the main conclusion of the paper? → Summarized conclusion.

• What accuracy did the proposed model achieve? → Extracts 97.12% / 98.43%

• Explain the dataset used. → Returns dataset name and description.

• List algorithms mentioned. → Outputs GRU-CNN, RSA, etc.

# 6. Screenshots to Attach

1. Home screen of Streamlit app  
2. Console log with FAISS index built  
3. Answer + Retrieved chunks display  
4. GitHub repository page  
5. Streamlit Cloud live app page

# 7. Expected Impact

• Reduces 60–70% of manual reading effort.  
• Provides instant insights.  
• Increases accessibility and reliability.

# 8. Future Enhancements

|  |  |
| --- | --- |
| Enhancement | Description |
| Multi-PDF Querying | Cross-document search for insights |
| Persistent Database | Reusable embeddings |
| Analytics Dashboard | Track accuracy/confidence |
| Mobile App | Offline access |
| Blockchain Logging | Audit trail security |

# 9. Team Details

|  |  |  |
| --- | --- | --- |
| Name | Role | Skills |
| Arul Selvam P | Team Leader | AI, LLMs, Streamlit, Python |
| Member 2 | Backend Developer | LangChain, APIs |
| Member 3 | UI/UX Developer | Frontend, Streamlit |
| Member 4 | Research Analyst | Data, Evaluation |

# 10. References

• LangChain Documentation  
• Groq API Docs  
• Streamlit Cloud Deployment  
• FAISS Facebook AI Library