**Project Overview**

This project is part of The Great AI Hackathon Malaysia organized by APU Malaysia.

**Problem Statement**

Healthcare professionals in Malaysia struggle to efficiently access critical information from vast, unstructured medical PDFs. Manual searches are time-consuming, error-prone, and limit clinical insights, impacting patient care. Our proposed AI query tool enables natural language smart search across all medical PDF documents via an intuitive chat-like interface.

**Our Solution**

From the problem statement, we thought how can we provide clinicians with a fast way for them to read summarised patient records?

The way we approached it is by developing our AI-powered chatbot called Medical Co-Pillot. Medical Co-Pilot is a chatbotlike AI-powered system that enables the ability to understand patient records, and gives the ability for doctors and nurses to search patient data with the usage of prompts. Furthermore, the answer that would be given by our chatbot can be traced back to the original pdf file. This is to ensure accuracy, transparency and confidence for doctors to cross-check and ensure the information received is accurate.

This system acts as a clinical assistant, helping doctors to save time by giving the patient records without having to manually search for it

**Technology used**

**Front End (UI):** For front end, we use HTML, CSS, and JavaScript.

**Backend API:** For our backend, we wrote our own logic for scaling and chunking in python. What this means is that the system would break down large datasets into digestible, processible format.

**AWS Technologies:**

| **Service** | **Role in the System** |
| --- | --- |
| **Amazon Bedrock** | This provides access to AI models that is being used in our system |
| **AWS Lambda** | Host our python function, and run our function only when needed. Also integrated with DynamoDB for efficient data transfers. This setup is fully scalable and cost-effective. |
| **Amazon S3 Bucket** | Stores raw documents (PDFs) |
| **AWS Amplify** | How we host our frontend of the website |
| **Amazon CloudWatch** | Where we check the logs for any issues, and make sure everything is running expectedly. |
| **IAM (Identity & Access Management)** | Controls **who** can access the system and which services can talk to each other securely. |
| **API Gateway** | Allow us to expose API so that our client can access the backend service. In our case, it triggers the lambda function, which then interacts with dynamoDB, it also provides security, monitoring, and automatic scaling to ensure the system can follow up with any traffic increase. |

**Key features:**

1. **Three modes –** General for medical Q&A, Patient for case tracking, PDF for document summarization
2. **Fast, concise medical Q&A –** Ask about drugs, symptoms, diseases with quick and accurate answers
3. **Patient memory system –** Store and recall patient info (name, ID, unique cases) for case-specific assistance
4. **PDF summarization –** Summarize patient reports, lab results, and other medical documents
5. **Voice input –** Use speech-to-text for faster, hands-free interaction
6. **Light/Dark mode –** Switch themes based on working environment
7. **Session system (per mode) –** Create and manage separate chat sessions in each mode
8. **Session restore –** Click any session to reopen and continue previous conversations