# Semantic Medical Question Search Engine

## 🩺 Problem Statement

Doctors and patients often struggle to find relevant answers across vast medical forums or databases. Traditional keyword-based search engines lack the ability to understand the true semantic meaning of medical queries, leading to irrelevant or outdated information being retrieved. This project aims to address this challenge by implementing a semantic search engine tailored for the medical domain using the MedQuAD dataset.

## 🎯 Project Objective

To develop a scalable, efficient semantic search engine that:  
- Uses medical sentence embeddings to understand the semantic intent of queries  
- Retrieves top-N semantically relevant questions and expert answers from the MedQuAD dataset  
- Serves results via a simple FastAPI interface for easy querying and integration

## 🧠 Techniques and Tools Used

Below is a detailed list of technologies and techniques used throughout the project:

### 📦 Datasets

- MedQuAD: Medical Question Answering Dataset (CSV format from Kaggle)

### 📚 Preprocessing

- Text normalization: punctuation removal, whitespace cleanup  
- CSV parsing and column standardization

### 🤖 Embedding Models

- SentenceTransformers: `pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb`  
- Sentence-level embedding tailored for biomedical and clinical language  
- Normalized output for cosine similarity

### 📦 Vector Database

- FAISS (Facebook AI Similarity Search):  
 - Used for fast nearest neighbor retrieval  
 - `IndexFlatIP` used with cosine similarity over normalized vectors

### ⚙️ Semantic Search Logic

- Encodes the user query using the same embedding model  
- Computes semantic similarity between query and indexed medical questions  
- Returns top-N relevant Q&A pairs

### 🖥️ API Interface

- FastAPI: Lightweight and fast web framework  
- Swagger UI and ReDoc enabled for testing and documentation  
- `/search` endpoint accepts query and returns results in JSON

### 💻 Development Environment

- Python 3.10+  
- VSCode (for development)  
- Kaggle (for embedding generation using GPU)

## 📁 Directory Structure

semantic\_search\_medical/  
├── preprocess.py # Data loading & cleaning  
├── embed.py # Sentence embedding generation  
├── index.py # FAISS index creation  
├── search.py # Semantic search logic  
├── api.py # FastAPI interface  
data/  
└── medquad.csv, clean\_medquad.csv, question\_embeddings.npy  
models/  
└── faiss\_index.index, faiss\_index\_texts.npy