Overview:

The project is an interactive, AI-powered diagnostic reasoning simulator that helps users learn to think like doctors. In the game, users are presented with a new patient case each round. They must decide which symptoms and test results matter, update their diagnosis as more evidence becomes available, and ultimately make a call, just like a real clinician.

This tool blends clinical education, Bayesian reasoning, and interactive storytelling in an accessible web app designed to teach critical diagnostic skills through play.

Core Functionalities

1. **Randomized Patient Cases**
    - Age, sex, race, symptoms, vitals, and possible diseases are randomized using real-world data distributions
        - For now, symptoms and vitals are presented as text, but could possibly integrate graphics/imaging from public medical radiology collections (like CT scans, X-rays, etc.)
    - No two patients are the same, increasing replayability
2. **Bayesian Inference Engine**
    - Behind the scenes, the system uses Bayes' rule to update disease probabilities based on user actions (asking about symptoms, ordering tests)
        - These probabilities are pulled from NIH research and medical literature; while they are well-documented, they aren't broadly used among doctors in practice, so this tool helps improve their evidence-based reasoning through probabilistic thinking
3. **Explainable AI**
    - When users request test results or symptoms, the system explains what they mean and why they matter diagnostically
4. **Diagnosis + Feedback Loop**
    - Users submit a final diagnosis
    - The system gives tailored feedback on:
        - Accuracy
        - Efficiency (did you over-order tests?)
        - Cognitive biases (anchoring, confirmation bias)
        - What an expert might have done differently
5. **User Interface**
    - Clean, minimalist interface with buttons and text (no graphics yet but can be added later; I was thinking of making the graphics myself)

Why This Matters

- **Medical education is limited by access to patients and mentorship**

    - This tool fills that gap by simulating real clinical decision-making in a low-stakes, accessible environment
- **Diagnostic error is a leading cause of medical harm**
    - Training better reasoning early could reduce real-world mistakes
- **It's useful beyond medical students.**
    - This could be played by premeds, clinicians-in-training, or even curious non-experts who want to understand healthcare better
- **It teaches critical thinking** in a world increasingly shaped by AI and information overload
- **Medical simulators do exist** but not any that update probabilities with every test and symptoms, even though those are well-documented in medical literature and research; this project helps connect future practitioners with this empirical scientific knowledge to improve medical diagnosis

Current State

- I have JSON files of 4 similar digestive system diseases: appendicitis, acute cholecystitis, small bowel obstruction, acute mesenteric ischemia. Each file includes:
    - Disease name
    - Demographics of patients with the disease (age distributions, probability of certain gender, etc.)
    - Symptoms with probability of having that symptom given disease present
    - Vitals with distributions expected for the disease
    - Diagnostic tests that can be used to test for the disease and their sensitivity/specificity
- I am currently implementing the Bayesian engine, which updates the probabilities of having one of a certain set of diseases (in this case, the four digestive system diseases) based on the current symptoms, vitals, and demographics
- Random case generation: This should be functioning but theoretically, the game will randomly select a disease from a set of diseases based on the distributions of those diseases in the population (which would be then be used to form the Bayesian priors), then using the demographic distribution of the disease described in the JSON, a random case is generated and presented
- Ordering tests: This is really buggy right now but essentially, a user should be able to click a button and order a test, then the test result is returned as positive with the probability of the test's sensitivity and then the probabilities of all the diseases should be updated by the Bayesian engine (I am trying to get this to work)
- Submit diagnosis: User should be able to submit a diagnosis (I have it manually set to appendicitis I think) and is told whether or not it is correct, will implement feedback later