Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

- Define a hypothetical AI problem (e.g., "Predicting student dropout rates").
- List **3 objectives** and **2 stakeholders**.
- Propose **1 Key Performance Indicator (KPI)** to measure success.

2. Data Collection & preprocessing (8 points)

- **Identify 2 data sources for your problem.**
  - Electronic Health Records (EHR) Database (Primary)
  - Census or Socioeconomic Data (External)
- **Explain 1 potential bias in the data.**
  - The dataset may disproportionately represent patients from a single hospital system, geographic area, or demographic group (e.g., a specific race or payer type).
- **Outline 3 preprocessing steps (e.g., handling missing data, normalization).**
  - Handling Missing/Unseen Categorical Data
  - Feature Engineering and Filtering
  - Feature Scaling (Normalization/Standardization)

3. Model Development (8 points)

- **Choose a model (e.g., Random Forest, Neural Network) and justify your choice.**

Model	**Justification**
XGBoost Classifier	**High Performance & Robustness:** XGBoost is an optimized implementation of gradient boosted decision trees. It is highly effective for structured data (like EHR data) and consistently wins machine learning competitions due to its ability to handle complex, non-linear relationships without extensive manual feature engineering.
	Feature Importance: It naturally provides **feature importance** scores, which is crucial in a clinical setting for explaining *why* a certain patient was flagged as high-risk.
	Handles Mixed Data & Scale: It inherently handles both numerical and one-hot encoded categorical features (as done in your training.py) and is relatively **insensitive to feature scaling**, making the overall pipeline development simpler and more stable.

- **Describe how you would split data into training/validation/test sets.**

Set	Typical Percentage	Purpose
Training Set	70%-80%	Used to **fit** the model (i.e., teach the model the underlying patterns).
Validation Set	10%-15%	Used for **hyper parameter tuning** (e.g., finding the optimal max_depth or learning_rate). The model never sees this data during training.
Test Set	10%-15%	Used only once, right at the end, to provide the final, **unbiased evaluation** of the selected model configuration.

- **Name 2 hyper parameters you would tune and why.**
  - learning_rate
  - max_depth

4. Evaluation & Deployment (8 points)

- **Select 2 evaluation metrics and explain their relevance.**
  - F1-Score
  - Recall
- **What is concept drift? How would you monitor it post-deployment?**
  - Refers to the phenomenon where the statistical properties of the target variable, which the model is trying to predict, change over time in an unforeseen way. This causes the predictions to become less accurate, even if the deployment data is still well-formatted.
- **Describe 1 technical challenge during deployment (e.g., scalability).**
  - **Latency (Response Time):** Healthcare decisions are often time-critical. If the Stream lit application is used in an **Electronic Health Record (EHR)** system, the prediction for a patient must be near-instantaneous