Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

AI Problem:
"Predicting hospital readmission risk within 30 days of discharge."

Objectives:

1. Identify high-risk patients to enable early intervention.
2. Reduce overall hospital readmission rates.
3. Optimize resource allocation for patient follow-up care.

Stakeholders:

- **Hospital administrators** (interested in reducing penalties and costs).
- **Healthcare providers** (interested in improving patient outcomes).

Key Performance Indicator (KPI):

- **30-day readmission rate reduction** (percentage of patients readmitted within 30 days of discharge).

---

2. Data Collection & Preprocessing (8 points)

Data Sources:

1. **Electronic Health Records (EHRs)** — including diagnoses, procedures, lab results, and discharge notes.
2. **Insurance Claims Data** — includes medication history and previous hospital visits.

Potential Bias:

- **Socioeconomic bias**: Patients from lower-income areas may have fewer follow-up visits, affecting model accuracy and fairness.

Preprocessing Steps:

1. **Handle missing data** (e.g., impute missing lab values using median).
2. **Normalize numerical features** (e.g., age, length of stay) to ensure consistent scale.
3. **Encode categorical variables** (e.g., diagnosis codes using one-hot encoding).

---

3. Model Development (8 points)

Chosen Model:

- **Random Forest** — It handles non-linear relationships well, is robust to missing data, and provides feature importance for interpretability.

Data Splitting Strategy:

- **70% training**, **15% validation**, **15% test** — to ensure reliable tuning and unbiased evaluation.

Hyperparameters to Tune:

1. **Number of trees (n_estimators):** Affects model performance and runtime.
2. **Maximum tree depth (max_depth):** Controls overfitting and generalization.

---

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

1. **Precision** — To minimize false positives (e.g., flagging low-risk patients as high-risk).
2. **Recall** — To catch as many true high-risk patients as possible.

Concept Drift:

- **Definition:** A change in the data distribution or relationships over time (e.g., new treatment protocols).
- **Monitoring Strategy:** Periodically retrain the model and track performance metrics over time.

Technical Challenge:

- **Scalability** — Ensuring the model can handle real-time predictions across thousands of patients daily without delays.

Part 2: Case Study Application (40 points)

Scenario: Predicting Patient Readmission Risk within 30 Days of Discharge

---

1. Problem Scope (5 points)

Problem Definition:
The hospital seeks an AI system that predicts the likelihood of a patient being readmitted within 30 days after discharge, helping reduce avoidable readmissions and improve patient care.

Objectives:

1. Accurately identify patients at high risk of readmission.

2. Enable timely interventions such as follow-up care.
3. Reduce costs and readmission penalties for the hospital.

Stakeholders:

- **Hospital Administrators:** Focused on reducing readmission rates and associated costs.
- **Healthcare Providers (Doctors/Nurses):** Interested in early identification of high-risk patients.
- **Patients:** Beneficiaries of improved care and reduced complications.

---

2. Data Strategy (10 points)

Proposed Data Sources:

1. **Electronic Health Records (EHRs):** Diagnoses, vitals, lab results, discharge summaries.
2. **Demographic Data:** Age, gender, ethnicity, socioeconomic status.
3. **Medication Records:** Prescriptions and compliance.
4. **Previous Hospitalization History:** Number of past admissions, reasons.

Ethical Concerns:

1. **Patient Privacy & Data Security:** Medical data must be securely stored and processed, in compliance with **HIPAA**.
2. **Bias & Fairness:** If the model learns from biased historical data, it may unfairly predict higher risk for certain racial or socioeconomic groups.

Preprocessing Pipeline:

1. **Data Cleaning:** Remove duplicates, handle missing lab values (e.g., median imputation).
2. **Encoding:** Convert categorical data (e.g., diagnosis codes, gender) using one-hot or label encoding.
3. **Feature Engineering:**
   - Create a feature for “number of prior admissions.”
   - Calculate length of hospital stay.
   - Create interaction terms between diagnosis and age group.
4. **Scaling:** Normalize numerical features (e.g., blood pressure, lab results).
5. **Train-Test Split:** Stratified sampling to ensure balanced readmission cases across sets.

---

3. Model Development (10 points)

Chosen Model:

- **Gradient Boosting Machine (GBM)** (e.g., XGBoost) — Offers high performance with tabular data, handles missing values, and provides interpretability via feature importance.

Hypothetical Confusion Matrix (100 samples):

	Predicted Yes	**Predicted No**
Actual Yes	25	10
Actual No	5	60

Calculations:

- **Precision =** TP / (TP + FP) = 25 / (25 + 5) = **0.83**
- **Recall =** TP / (TP + FN) = 25 / (25 + 10) = **0.71**

---

4. Deployment (10 points)

Steps to Integrate the Model:

1. **API Deployment:** Wrap the model in an API (e.g., using Flask/FastAPI).
2. **EMR Integration:** Connect the model API with the hospital's Electronic Medical Records (EMR) system.
3. **Real-time Prediction:** Enable risk scores to be generated automatically at discharge.
4. **Alert System:** Notify care teams for high-risk patients for follow-up planning.
5. **Logging & Monitoring:** Continuously track model performance and log predictions for auditing.

Ensuring Regulatory Compliance (e.g., HIPAA):

- Use **end-to-end encryption** for data in transit and at rest.
- Conduct **regular audits and access controls**.
- Use **de-identified data** for training.
- Maintain **data minimization** (only collect necessary features).
- Document all **risk assessments** and obtain patient consent where applicable.

---

5. Optimization (5 points)

Method to Address Overfitting:

- **Cross-Validation with Early Stopping:** Use k-fold cross-validation and monitor validation loss during training. Stop training once the validation loss increases to prevent overfitting.

Part 3: Critical Thinking (20 points)

---

1. Ethics & Bias (10 points)

How might biased training data affect patient outcomes in the case study?
Biased training data can result in unfair treatment of certain patient groups. For example, if the historical data underrepresents patients from low-income or minority communities, the model may consistently assign them a lower risk score—even if they are at higher risk—leading to a lack of necessary interventions. Conversely, some groups might be over-predicted as high risk, leading to unnecessary resource allocation and patient anxiety.

This bias can worsen healthcare disparities, reduce trust in the system, and result in poorer outcomes for vulnerable populations.

Strategy to Mitigate Bias:

- **Fairness-Aware Sampling:** During data preparation, ensure balanced representation across sensitive attributes (e.g., race, age, gender) using techniques like **re-weighting** or **stratified sampling**. Additionally, apply fairness metrics (e.g., disparate impact ratio) and use tools like **IBM AI Fairness 360** to audit and mitigate bias before deployment.

---

2. Trade-offs (10 points)

Trade-off Between Model Interpretability and Accuracy in Healthcare:

- **High Accuracy Models** (e.g., Neural Networks, Gradient Boosting) may produce better predictive performance but are often "black boxes," making it difficult to explain why a certain prediction was made. In healthcare, this lack of transparency can reduce trust among clinicians and complicate compliance with legal and ethical standards.
- **Interpretable Models** (e.g., Logistic Regression, Decision Trees) offer clear reasoning behind predictions but might sacrifice some accuracy, especially with complex datasets.

In healthcare, **interpretability is crucial** for clinical validation and decision support. Therefore, a **hybrid approach**—using high-performing models with explainability tools (e.g., SHAP, LIME)—can help balance this trade-off.

Impact of Limited Computational Resources on Model Choice:
If the hospital has limited computational power (e.g., older systems or no GPUs), it may:

- **Avoid computationally expensive models** like deep neural networks.

- Prefer lightweight, interpretable models such as **logistic regression** or **random forests** with fewer trees.
- Use **model quantization or compression techniques** to reduce memory and CPU usage for deployment.

This constraint emphasizes the need for **efficiency and scalability**, even if it means choosing slightly less accurate models.

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

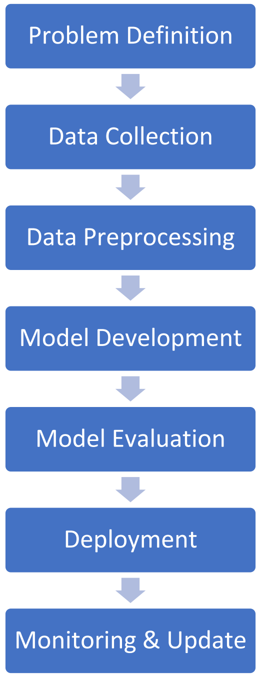
Most Challenging Part:
The most challenging part of the workflow was **ensuring fairness and ethical integrity** in the model. Balancing model performance with fairness across diverse patient groups is difficult, especially when historical data may reflect systemic biases. Healthcare decisions carry high stakes, so biased predictions can directly impact lives.

How I Would Improve with More Time/Resources:

- Implement **fairness-aware machine learning tools** (e.g., IBM AI Fairness 360) during model training.
- Collaborate with **domain experts** (e.g., clinicians, ethicists) to validate feature choices and model outputs.
- Collect **more comprehensive and balanced datasets**, especially covering underrepresented patient groups.
- Include **explainable AI (XAI)** methods (e.g., SHAP, LIME) to enhance model transparency for clinical use.

Diagram (5 points)

AI Development Workflow Diagram:



Each stage feeds into the next, creating a continuous loop where **monitoring results** may trigger **retraining or updates**, ensuring long-term performance and fairness.