1. Problem Scope (5 points)

Problem definition:
Predict whether a patient will be readmitted within 30 days after hospital discharge, using data available at or shortly after discharge.

Objectives :

1. Identify high-risk patients before discharge so care teams can apply targeted interventions (e.g., follow-up calls, care coordination).
2. Reduce unplanned 30-day readmission rates.
3. Optimize allocation of post-discharge resources (home visits, medication reconciliation).

Stakeholders :

- Clinical staff (physicians, discharge planners, nurses).
- Hospital administration / quality improvement team.
- (Also: patients and their caregivers; IT/security/compliance teams.)

2. Data Strategy (10 points)

Data sources

1. **Electronic Health Records (EHR):** diagnoses (ICD codes), procedures, vitals, lab results, problem lists, medications, length of stay, discharge disposition.
2. **Operational / administrative data:** prior admissions history, insurance type, appointment follow-ups scheduled.
   (Additional useful sources: social determinants data (address → area-level SES), pharmacy refill records, post-discharge follow-up call logs, and unstructured discharge summaries.)

Two ethical concerns

1. **Patient privacy / PHI exposure:** EHRs contain protected health information; must ensure de-identification (where appropriate), encryption, access control, and minimum-necessary use.
2. **Algorithmic bias / equity:** Model may under/over-predict risk for groups defined by race, socioeconomic status, language, or insurance. This can lead to unequal access to interventions or misallocation of resources.

Preprocessing pipeline (ordered steps)

1. **Data ingestion & access controls:** Pull needed tables via audited pipelines; log access; enforce role-based permissions.
2. **De-identification / minimal PHI flow:** Remove/transform direct identifiers for model development environments (use pseudonymized patient IDs). For production scoring within hospital, keep linkage but with strict controls.
3. **Data cleaning:**
   - Resolve duplicates; standardize timestamps and units (e.g., mg/dL vs mmol/L).
   - Align events to an index date = discharge datetime.
4. **Missing data handling:**

- Analyze missingness mechanism (MCAR/MAR/MNAR).
- Impute clinically sensible values (median for labs, explicit "not measured" indicator for some features). Add binary flags for imputed fields.

5. **Feature engineering:** (key domain features)
    - **Comorbidity score(s):** e.g., Charlson or Elixhauser aggregated from ICD codes.
    - **Prior utilization:** count of ED visits and inpatient admissions in last 6/12 months.
    - **Medication burden:** number of active medications at discharge.
    - **Recent lab/vital trends:** slope or delta of key labs (e.g., creatinine) over last 48–72 hours.
    - **Discharge factors:** length of stay, discharge disposition (home vs SNF), scheduled outpatient follow-up within 7 days (yes/no).
    - **Social risk flags:** e.g., zip-code level deprivation index, documented housing instability, or language barrier.
    - **NLP features:** extract key phrases from discharge summary (e.g., "poor follow-up", "non-compliant") using a validated pipeline; include negation handling.
6. **Encoding & scaling:**
    - One-hot or target encoding for categorical variables with cardinality control.
    - Scale numerical features if model requires it (tree models don't strictly require scaling).
7. **Train/val/test split using time awareness:** Ensure no leakage by splitting on discharge date (e.g., train on older discharges, validate on more recent dates).
8. **Data quality checks & validation:** distribution checks, schema validation, and clinical review of extreme values.
9. **3. Model Development (10 points)**

Model selection and justification

Primary model: Gradient Boosted Decision Trees (e.g., XGBoost / LightGBM / CatBoost).
Why:

- Excellent performance on structured, tabular clinical data.
- Handles missing values and mixed types well.
- Fast to train and predict; supports regularization and early stopping.
- Works well with engineered features (counts, scores, trends).
  Baseline models: Logistic regression (for a simple, interpretable baseline) and a small random forest. Use SHAP or similar explainability for clinician-facing explanations.

Confusion matrix (hypothetical)

Assume a test set of **1,000** discharged patients; actual 30-day readmissions = 120.

Hypothetical model predictions result in:

- True Positives (TP) = **90**
- False Positives (FP) = **60**

- False Negatives (FN) = **30**
- True Negatives (TN) = **820**

Confusion matrix format:

	Predicted Readmit (+)	**Predicted Not Readmit (−)**
Actual Readmit (+)	TP = 90	FN = 30
Actual Not Readmit (−)	FP = 60	TN = 820

Precision and recall (step-by-step arithmetic)

Precision = TP / (TP + FP)

- TP + FP = 90 + 60 = 150.
- Precision = 90 ÷ 150.
- 90 ÷ 150 = 0.6 → **60.0% precision**.

Recall (sensitivity) = TP / (TP + FN)

- TP + FN = 90 + 30 = 120.
- Recall = 90 ÷ 120.
- 90 ÷ 120 = 0.75 → **75.0% recall**.

(You can also compute F1 = 2 * (Precision * Recall) / (Precision + Recall) = 2*(0.60.75)/(0.6+0.75)=20.45/1.35=0.666... → ~66.7% F1.)

4. Deployment (10 points)

Integration steps (practical rollout)

1. **Clinical validation & stakeholder buy-in:** Present retrospective performance and decision thresholds to clinicians; run a prospective silent (shadow) pilot where model predictions are recorded but not acted upon.
2. **Technical integration:**
    - Expose model as a secured internal API (FHIR or REST) that accepts EHR discharge payload and returns risk score + top contributing features.
    - Integrate API with EHR workflow (e.g., a CDS alert or a dashboard in the clinician portal). Use HL7/FHIR for compatibility.
3. **UX & explainability:** Display risk category (low/medium/high) and 2–3 top contributing factors (via SHAP) with short, clinician-friendly explanations and suggested next steps.
4. **Pilot & controlled rollout:** Start in one ward or service line, collect clinician feedback, measure process metrics (interventions triggered) and outcome metrics (readmission rate change).
5. **Monitoring & logging:** Real-time logging for inputs, predictions, model latency, and downstream actions. Implement alerting for runtime failures.

6. **Feedback loop:** Capture post-discharge outcomes to retrain/refresh model; implement a data labeling pipeline.
7. **Maintenance plan:** Scheduled model re-training cadence, drift detection, and governance reviews.

Ensuring compliance with healthcare regulations (e.g., HIPAA)

(High-level controls applicable to HIPAA and similar international regulations.)

1. **Data governance & contractual:** Execute Business Associate Agreements (BAAs) with vendors handling PHI. Define roles, responsibilities, and permitted uses.
2. **Minimum necessary principle:** Only use data fields required for modeling; implement field-level access control.
3. **Encryption & secure transport:** Encrypt PHI at rest (AES-256 or equivalent) and in transit (TLS 1.2+).
4. **Access control & audit logging:** Role-based access, multi-factor auth for admin interfaces, and immutable audit logs for data access and model inference events.
5. **De-identification for development:** Use de-identified or limited datasets for model development and testing where possible. Keep linkage keys in a separate, highly restricted environment.
6. **Risk assessments & DPIA:** Perform security risk assessments and Data Protection Impact Assessments; remediate findings.
7. **Patient consent & notices:** Where required, update privacy notices and obtain consent or meet lawful basis for secondary uses.
8. **Clinical safety & validation:** Maintain clinical validation records; involve institutional review boards (IRB) if model use is research rather than operational.
9. **Incident response:** Have a breach/incident plan with notification timelines per regulation.

5. Optimization (5 points)

Method to address overfitting:
Use **k-fold cross-validation combined with early stopping and regularization** on the chosen model.

- **Why:** k-fold CV provides robust estimates of generalization across different patient samples; **early stopping** prevents the gradient boosting model from training beyond the point where out-of-fold performance degrades; **regularization parameters** (e.g., L1/L2, tree depth, min_child_weight, learning rate) reduce model complexity.
- **Practical step:** Run stratified time-aware k-fold (or rolling window CV) to respect temporal structure, choose the number of boosting rounds by early stopping on validation folds, and tune regularization hyperparameters.