PART 1: SHORT ANSWER QUESTIONS

1. Problem Definition

- ✓ **Hypothetical AI Problem;**
  - Predicting crop diseases at their early stage using images taken and give relevant solutions.
- ✓ **Objectives;**
  - Accurately identify symptoms of common plant diseases through image classification.
  - Embed the model locally for farmers unable to access internet connection.
  - Provide farmers with advice and solutions for the treatment and prevention of the plant disease as well as nearby agrovets for further consultation.
- ✓ **Stakeholders;**
  - **Smallholder farmers** who benefit directly from timely disease diagnosis and guidance.
  - **Agricultural NGOs and Extension Officers** who organize and support farmers interested in scalable tech solutions.
- ✓ **Key Performance Indicator (KPI);**
  - Measure how often the AI system correctly predicts outcomes based on input data.
  - Can be done by measuring the percentage of correctly identified crop diseases on a test dataset compared to expert labels
  - Accuracy=Number of Correct Predictions/Total Number of Predictions×100%
  - If the accuracy is greater or equal to 90%, the labeled test dataset ensures the model is dependable and ready for deployment.

2. Data Collection & Preprocessing

- ✓ **Data Sources;**
  - **PlantVillage Dataset:** An open-source dataset containing thousands of labeled crop disease images across multiple plant species. It contains clear visual samples taken under controlled conditions hence widely used in academic research.
  - **Farmer-Contributed Mobile Images:** These will be collected in real-time from smallholder farmers via the mobile app. These images should be taken at multiple angles so as to offer authentic examples of crop disease conditions.
- ✓ **Potential Data Bias;**
  - The images from the PlantVillage dataset are often taken in well lit environments with clean backgrounds which may not be the actual case in farm conditions. This may make the AI model perform poorly and struggle to interpret the real-world scenario images.
- ✓ **Preprocessing Steps;**
  - **Handling Missing Data:** Remove images without labels or ones taken with poor resolution.

- **Image Normalization:** Resize all images to a consistent resolution to match the input requirement of CNN models or Standardize pixel intensity values.
- **Data Augmentation:** Simulate variations such as rotating, cropping or flipping to the image so as to make the model robust to diverse input styles.

3. Model Development

- ✓ **Model of Choice; A Convolutional Neural Network (CNN)**
  - It is highly effective for image classifications due to its ability to automatically detect features such as shapes and textures.
  - The CNN can learn patterns of leaf discoloration, spots or textures for specific diseases.
- ✓ **How I would split data;**
  - A higher percentage (70%) would be for the **training set** used to teach the model.
  - 15% for the **validation set** used to check performance during training.
  - The other 15% for **test set** used for final performance evaluation.
  - Technique I would go for is the Stratified Shuffle Split to ensure disease classes are evenly represented across all sets.
- ✓ **Hyperparameter to tune;**
  - **Learning rate;**

  To control how quick the model adjusts weights so as to ensure faster convergence with good accuracy.
  - **Batch size;**

  Will determine how many images will be fed to the model in one step.

4. Evaluation & Deployment

- ✓ **Evaluation Metrics and their Relevance;**
  - **Accuracy;**

  Measures the proportion of correct predictions out of all predictions and is relevant since we want the AI model to correctly identify diseases from images.
  - **F1 Score;**

  This is the harmonic mean of precision and recall when disease classes are imbalanced and is relevant to ensure that the diseases that are less frequent than others aren't ignored just because the model is good at predicting common diseases.

- ✓ **Concept Drift;**
  - This is what occurs when the statistical properties of the target variable change over time such as when a disease may evolve due to climate change or new crop strains.
  - To monitor the drift post-deployment, we can collect new labeled data periodically and use tools such as Drift Detection Methods to statistically detect changes in input distributions.

- ✓ **Technical challenge during deployment;**
  - A major technical challenge is scalability on low-resource mobile devices since a majority of farmers in rural areas use older smartphones having limited memory, processing power and intermittent internet access.
  - Running a heavy AI model on such devices can lead to slow performance and frequent crashes without being of any help to the farmer. This will eventually affect both usability and trust in the system.
  - To solve this, TensorFlow Lite can come in handy to convert models to run offline and MobileNet which is optimized for mobile deployment.

PART 2: CASE STUDY APPLICATION

1. Problem Scope

- ✓ **Problem**
  - Hospitals face significant challenges with patients who are readmitted within 30 days of discharge.
  - These readmissions are costly, may indicate poor post-discharge care or premature discharge decisions, and are often used as a benchmark for healthcare quality.
  - The AI system cannot analyze patient records and predict the likelihood of readmissions.
- ✓ **Objectives**
  - To predict which patients are most likely to be readmitted within 30 days based on the structured health records and clinical data.
  - To provide actionable insights to clinicians and care coordinators so that they can easily follow up or allocate resources.
  - To help the hospital reduce readmission rates by enabling preventive measures such as post-discharge monitoring.
- ✓ **Stakeholders**
  - **Hospital administrators** who are responsible for managing operational costs and regulatory compliances.
  - **Clinical and care managers** who will use the AI-generated risk scores to guide decisions about discharge planning and patient follow-up.

2. Data Strategy

- ✓ **Data sources**
  - **Electronic Health Records (EHRs)** which will include the admission history, discharge summaries, diagnosis codes, medications, lab results and length of stay.
  - **Demographic Information** that includes the age, gender, ethnicity, marital status, and address.
  - **Clinical notes** from the physician observations that may contain context that has not been captured by structured fields.

   - **Follow-up and appointment records** that shows if patients attended post-discharge appointments or missed follow-ups so as to identify the care gaps.
- ✓ **Ethical concerns**
   - **Patient privacy and data confidentiality** since medical records are extremely sensitive. The AI system must comply with the data protection laws to ensure encrypted storage, secure transmission and restricted access.
   - **Bias and fairness in prediction.** If the historical data reflect systemic biases such as underserved populations being discharged earlier, the AI might unfairly predict higher readmission risks leading to discriminatory treatment recommendations.
- ✓ **Preprocessing pipeline**
   - **Data cleaning** that removes duplicates or corrupted entries.
   - **Feature engineering** that creates new features from raw data and encoding categorical variables.
   - **Normalization** of continuous features such as age, lab results or vitals using a StandardScaler.
   - **Text processing** in the case of clinical notes where it can be tokenized and vectorized using transformer-based models.

3. <u>Model Development</u>

- ✓ **Model; Random Forest Classifier**
   - The Random Forest Classifier model that handles structured tabular data like the EHRs. It builds multiple decision trees and aggregates their predictions, reducing variance and improving generalization.
   - It is ideal since it handles both numerical and categorical features with ease.
   - It manages missing values and imbalanced data.
   - It also provides feature importance scores, allowing clinicians to understand which factors influence readmission risk the most.

- ✓ **Hypothetical data (confusion matrix and calculation precision)**

	Predicted: No Readmission	**Predicted: Readmission**
Actual: No	80	20
Actual: Yes	10	90

True Positives: 90
True Negatives: 80
False Positives:20
False Negatives:10

Precision (positive predicted value) = True Positives/ True Positives + False Positives
Recall (true positive rate) = True Positives/ True Positives + False Negatives

- If the precision is above 80%, it means the model is effective at minimizing false alarms.
- If the recall is above 90%, it means the model catches most true readmission risks which is crucial for patient care planning.

4. Deployment

✓ **Integrating the model into the hospital's system**
- **Model packaging:** Convert the trained Random Forest model into a deployable format and wrap the model in a secure API using backend framework like Flask.
- **System integration:** Connect the API to the hospital's HER system and enable the API to extract real-time patient data at discharge.
- **User Interface Deployment:** Embed the model output into existing physician tools where the risk scores can be displayed alongside discharge planning tools to assist clinicians.
- **Security configuration:** Use HTTPs encryption for data exchange between hospitals and the model API.
- **Monitoring and feedback loop:** Allow clinical staff to flag incorrect predictions, creating a feedback loop for continuous model retraining.

✓ **Compliance with healthcare regulations**
- Ensure all data storage is encrypted both at rest and in transit.
- Limit access to sensitive data via role-based access controls and audit trails.
- Borrow consent from the patients and explain that the predictive modeling is used as part of care planning.
- Regularly monitor the model outputs to ensure fairness is enhanced.

5. Optimization

✓ **Implementing regularization**
- To address overfitting, an effective method is implementing regularization such as the L2 regularization.
- Overfitting normally happens when the model learns not just the meaningful patterns but also noise from the training data which hurts performance on unseen data.
- The L2 regularization combats this by adding penalty to large weights, encouraging the model to prefer simpler functions that generalize better.

PART 3: CRITICAL THINKING

✓ **Ethics & Bias**
- Underrepresentation of marginalized groups might result in lower predicted risk scores for these populations causing care teams to overlook their need for proactive follow-up or home health services.

- Overrepresentation of certain conditions could make the model label patients with those diagnoses as high risks even if they had successful treatment histories leading to unnecessary interventions.
- Can be mitigated by conducting fair audits and rebalance the training dataset before deploying the model and after to ensure the issue is mitigated.

✓ **Trade-offs between interpretability and accuracy in healthcare**
- In healthcare applications, there's often a tension between choosing highly accurate models and models that are easy to interpret.
- Complex models may deliver superior predictive performance but they may be difficult for clinicians to understand how predictions are made leading to limited clinical adoption and challenge regulatory approval.
- Simpler models on the other hand offer high interpretability allowing medical professionals to trace predictions back to specific variables which supports informed decision-making and ethical accountability. However, these models may sacrifice predictive power especially when dealing with nonlinear relationships or high-dimensional datasets.
- Hospitals may prefer interpretable models to understand why a patient is considered high risk and help tailor discharge plans and justify interventions. A balance must therefore be struck depending on clinical goals, risk tolerance and deployment environment.

✓ **Impact of limited computational resources on model choice**
- The limited computational resources such as outdated servers, limited memory or no access to GPU acceleration significantly affects the model selection.
- High complexity models demand substantial computational power for both training and inference which could lead to delays, crashes or degraded performance.

PART 4: REFLECTION & WORKFLOW DIAGRAM

✓ **Challenging part of the workflow**
- Most challenging phase of the workflow was data preprocessing and feature engineering.
- Working with EHRs posed difficulty especially inconsistent formats, missing values and high dimensional categorical data like diagnosis codes.
- Converting the raw, messy data into a structured format required a lot of cleaning and domain-specific decisions.
- Also, ensuring that engineered features were both informative and unbiased added complexity, especially when accounting for ethical implications and regulatory constraints in healthcare.

✓ **Improvements with more time and resources**
- A robust data pipeline using tools like Apache Airflow and Pandas Profiling for automated data validation and feature tracking.
- Collaboration with clinical experts to refine feature selection and incorporate unstructured data using NLP models.
- Invest in explainability tools that would enhance interpretability ensuring clinicians understand the rationale behind predictions and increasing trust in the system.

✓ **Sketch of the flowchart of the AI development workflow**

(Problem Definition)

↓

(Data Collection & Preprocessing)

↓

(Model Selection & Training)

↓

(Model Evaluation)

↓

(Deployment)

↓

(Monitoring & Maintenance)

↓

(Optimization)

↓

(Ethical Review & Bias Mitigation)

↓

(Final Reflection & Reporting)