1. Problem Definition (6 points)

AI Problem:
Develop an AI model that can automatically detect human emotions (e.g., happiness, anger, sadness, surprise, fear, neutral) from facial images.

Objectives:

1. Build a convolutional neural network (CNN) model to classify emotions from facial images.
2. Improve user experience in applications like virtual classrooms, security systems, and health monitoring.
3. Provide real-time emotion insights for human-computer interaction.

Stakeholders:

- **Developers/Researchers:** Building emotionally intelligent applications.
- **Organizations:** Using emotion analytics for customer feedback, marketing, or therapy support.

KPI (Key Performance Indicator):

- **Model Accuracy (%)** on emotion classification across all categories.

---

2. Data Collection & Preprocessing (8 points)

Data Sources:

1. **FER-2013 Dataset** (available on Kaggle) — contains 35,000+ labeled facial images across 7 emotions.
2. **AffectNet Dataset** — large-scale emotion dataset annotated with both categorical and dimensional labels.

Potential Bias:

- Overrepresentation of certain demographics (e.g., age, ethnicity) may cause bias, leading to inaccurate emotion recognition for underrepresented groups.

Preprocessing Steps:

1. Resize and normalize all images (e.g., 48×48 grayscale).
2. Perform data augmentation (rotation, flipping, zoom) to improve generalization.
3. Convert categorical labels (e.g., “happy”, “sad”) into numerical format using one-hot encoding.

---

3. Model Development (8 points)

Chosen Model:
ResNet-18 (Transfer Learning) — chosen for its balance between performance and efficiency. It captures complex image features effectively.

Data Split:

- 70% Training
- 15% Validation
- 15% Testing

Hyperparameters to Tune:

1. learning_rate — affects convergence speed and accuracy.
2. batch_size — controls model stability and training speed.

---

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

1. **Accuracy:** Overall correctness of emotion predictions.
2. **F1-Score:** Evaluates balance between precision and recall, especially useful when some emotions are less frequent.

Concept Drift:
Occurs when real-world emotion distributions shift (e.g., different lighting, new camera types, or cultural differences).

- **Monitoring:** Regularly retrain model with new, diverse datasets to maintain fairness and accuracy.

Technical Challenge:

- **Real-time Inference:** Running CNNs efficiently on edge devices (e.g., phones or webcams) may require optimization with TensorRT or ONNX.

Part 2: Case Study Application (40 points)

Scenario: Predicting Patient Readmission Risk (Healthcare Example)

---

1. Problem Scope (5 points)

Problem:
Hospitals want to use emotion detection to monitor patient stress and mental health via facial analysis.

Objectives:

1. Identify emotional distress from patient video streams.
2. Support early intervention for anxiety or depression.

Stakeholders:

- **Psychologists / Doctors:** For therapy support and mood tracking.
- **Hospital IT Teams:** Integrating emotion monitoring into health systems.

---

2. Data Strategy (10 points)

Data Sources:

- Live webcam feeds or hospital monitoring cameras (with consent).
- FER and AffectNet datasets for model pretraining.

Ethical Concerns:

1. **Privacy:** Capturing facial data must comply with data protection laws (e.g., GDPR, HIPAA).
2. **Informed Consent:** Patients must be aware of how their data is used.

Preprocessing Pipeline:

1. Face detection (e.g., using MTCNN or OpenCV).
2. Cropping and normalizing faces to uniform size.
3. Feature extraction (emotion embeddings) using pretrained CNNs.

---

3. Model Development (10 points)

Chosen Model:
Fine-tuned ResNet-18 or MobileNetV2 for facial emotion classification.

Confusion Matrix (Hypothetical):

	Predicted: Happy	**Predicted: Sad**	**Predicted: Angry**
Actual: Happy	95	3	2
Actual: Sad	5	87	8
Actual: Angry	4	6	90

Precision (Happy): 95 / (95 + 9) = **0.91**
Recall (Happy): 95 / (95 + 5) = **0.95**

---

4. Deployment (10 points)

Integration Steps:

1. Convert model to **TorchScript** or **ONNX** for deployment.
2. Build a **Streamlit web dashboard** for real-time emotion visualization.
3. Use **FastAPI** to expose REST endpoints for applications to query emotion results.

Regulatory Compliance:

- Use **data anonymization** and secure storage (AES-256 encryption).
- Comply with **HIPAA** and **GDPR** for facial data handling.

---

5. Optimization (5 points)

Overfitting Prevention:

- Apply **dropout layers**, **early stopping**, and **data augmentation** during training.

Part 3: Critical Thinking (20 points)

1. Ethics & Bias (10 points)

Impact of Biased Data:
If certain ethnic or age groups are underrepresented, the model may misclassify emotions (e.g., misinterpreting cultural facial expressions).

Mitigation Strategy:

- Use **bias-aware training** with demographically diverse datasets.
- Apply fairness metrics (e.g., demographic parity, equal opportunity).
- Regularly audit predictions using tools like **IBM AI Fairness 360**.

---

2. Trade-offs (10 points)

Interpretability vs Accuracy:

- Deep CNNs (e.g., ResNet) are accurate but hard to interpret.
- Simpler models (e.g., SVM with facial landmarks) are interpretable but less powerful.
- In healthcare or education, a balance is needed for ethical transparency.

Resource Constraints:

- Edge devices (Raspberry Pi, mobile) may require lightweight models like **MobileNet** instead of ResNet.

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

The most challenging part was ensuring fairness and accuracy across different demographic groups. With more time, I would integrate **federated learning** to train across multiple institutions without sharing private data.

Diagram (5 points)

AI Development Workflow:

Problem Definition

↓

Data Collection & Preprocessing

↓

Model Development & Training

↓

Evaluation & Validation

↓

Deployment

↓

Monitoring & Bias Mitigation
