Medical Portal System – API Documentation

1. Patient Login and View Reports

POST /api/auth/login
Authenticate a patient and return a session token.

Request:
{
  "email": "patient@example.com",
  "password": "mypassword"
}

Response (200):
{
  "status": "success",
  "token": "jwt_token_here",
  "userId": "patient_123"
}

Errors:
- 401 → Invalid credentials
- 403 → Account locked

GET /api/patients/{patientId}/reports
Fetch all reports for a patient.

Headers: Authorization: Bearer <token>

Response (200):
{
  "reports": [
    {
      "reportId": "rpt_001",
      "date": "2025-10-01",
      "riskScore": 0.82,
      "riskLevel": "HIGH"
    }

```
 ]
}
```

GET /api/reports/{reportId}
Get detailed report data including AI analysis.

Headers: Authorization: Bearer <token>

Response (200):
```
{
 "reportId": "rpt_001",
 "patientId": "patient_123",
 "imageUrl": "/storage/xray_123.png",
 "aiAnalysis": {
  "riskScore": 0.82,
  "confidence": 0.91,
  "recommendations": ["Schedule CT scan", "Consult specialist"]
 }
}
```

2. Radiologist Upload X-Ray Report

POST /api/reports/upload
Upload a patient X-Ray for AI analysis.

Form-Data Request:
patientId=patient_123
file=xray_image.png

Response (200):
```
{
 "reportId": "rpt_456",
 "status": "processing",
 "riskScore": null
}
```

Errors:
- 400 → Invalid image
- 413 → File too large

POST /api/ai/analyze
Run AI analysis on the uploaded image.

Request:
{
 "reportId": "rpt_456",
 "imageUrl": "/storage/tmp/xray_123.png"
}

Response (200):
{
 "riskScore": 0.65,
 "riskLevel": "MEDIUM",
 "confidence": 0.88,
 "analysisSummary": "Possible pneumonia patterns detected."
}

POST /api/notifications/send
Notify stakeholders about report and risk results.

Request:
{
 "reportId": "rpt_456",
 "recipients": ["patient_123", "doctor_001"],
 "type": "RISK_ALERT",
 "riskLevel": "MEDIUM"
}

Response (200):
{ "status": "sent" }

3. Doctor Review AI-Generated Report

GET /api/reports/priority?doctorId={doctorId}
Get prioritized report queue for a doctor.

Response (200):
{
 "reports": [

```
   { "reportId": "rpt_456", "riskLevel": "HIGH", "status": "pending" }
 ]
}
```

GET /api/reports/{reportId}/details
Get report details and AI analysis.

Response (200):
```
{
 "reportId": "rpt_456",
 "imageUrl": "/storage/xray.png",
 "aiAnalysis": {
  "riskScore": 0.65,
  "recommendations": ["Follow-up imaging"]
 }
}
```

POST /api/reports/{reportId}/review
Submit a doctor's review of an AI-generated report.

Request:
```
{
 "doctorId": "doctor_001",
 "notes": "Confirm pneumonia, suggest antibiotics.",
 "urgencyLevel": "HIGH",
 "finalDiagnosis": "Pneumonia"
}
```

Response (200):
```
{ "status": "reviewed", "reviewId": "rev_001" }
```

4. Tech Team Monitor AI Model Performance

GET /api/models/active
Fetch currently deployed models.

Response (200):
```
{
 "models": [
```

```
   { "modelId": "v3.2", "status": "active", "deployedAt": "2025-09-15" }
 ]
}
```

GET /api/models/{modelId}/metrics
Fetch performance metrics for a model.

Response (200):
```
{
 "accuracy": 0.92,
 "precision": 0.88,
 "recall": 0.90,
 "f1Score": 0.89,
 "timeRange": "last_7_days"
}
```

POST /api/models/{modelId}/retrain
Trigger model retraining.

Request:
```
{ "dataset": "lung_xray_2025", "epochs": 100 }
```

Response (200):
```
{ "status": "training_started", "jobId": "train_789" }
```

POST /api/models/{modelId}/deploy
Deploy a new trained model.

Request:
```
{ "version": "v3.3", "jobId": "train_789" }
```

Response (200):
```
{ "status": "deployed", "modelId": "v3.3" }
```

5. Complete User Journey (End-to-End)

Key APIs used:
1. Radiologist → POST /api/reports/upload

2. System → POST /api/ai/analyze
3. Notification → POST /api/notifications/send
4. Patient → POST /api/auth/login → GET /api/patients/{id}/reports
5. Doctor → GET /api/reports/priority → POST /api/reports/{id}/review
6. Patient follow-up → POST /api/appointments/schedule

6. AI Model Prediction Pipeline

POST /api/ai/validateImage
Validate uploaded image.

Response:
{ "valid": true, "issues": [] }

POST /api/ai/preprocess
Preprocess image before AI analysis.

Response:
{ "processedImageUrl": "/storage/tmp/processed_xray.png" }

POST /api/ai/extractFeatures
Extract features for ML model.

Response:
{ "featureVector": [0.12, 0.55, 0.87, ...] }

POST /api/ai/predict
Run forward pass prediction.

Response:
{ "rawPredictions": [0.1, 0.3, 0.6] }

POST /api/ai/calculateRisk
Calculate risk score from predictions.

Response:
{

```
  "riskScore": 0.6,
  "riskLevel": "MEDIUM",
  "explanation": "Detected anomalies in left lung."
}
```

POST /api/reports/{reportId}/saveAnalysis
Save AI analysis results.

Response:

```
{ "status": "saved", "analysisId": "anl_123" }
```