Sample API Requests for Medical Report Simplifier

Base URL (Local): http://localhost:8000/api/v1
Base URL (Railway): https://your-project-name.railway.app/api/v1

1. Health Check

```
curl -X GET "http://localhost:8000/api/v1/health"
```

2. Process Text Input - Basic Case

```
curl -X POST "http://localhost:8000/api/v1/process-text" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "CBC: Hemoglobin 10.2 g/dL (Low), WBC 11,200 /uL (High)"
  }'
```

3. Process Text Input - OCR-like with Errors (Shows AI Error Correction)

```
curl -X POST "http://localhost:8000/api/v1/process-text" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "CBC: Hemglobin 10.2 g/dL (Low), WBC 11200 /uL (Hgh)"
  }'
```

PROF

4. Process Text Input - Multiple Tests

```
curl -X POST "http://localhost:8000/api/v1/process-text" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "Complete Blood Count Results: Hemoglobin 8.5 g/dL (Low),
WBC 15000 /uL (High), RBC 3.2 million/uL (Low), Platelet 450000 /uL
(Normal)"
  }'
```

5. Demo Problem Statement Format (Perfect for Assignment Evaluation)

```
curl -X POST "http://localhost:8000/api/v1/demo-problem-statement" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "CBC: Hemglobin 10.2 g/dL (Low), WBC 11,200 /uL (Hgh)"
  }'
```

6. Debug Processing Steps - Text

```
curl -X POST "http://localhost:8000/api/v1/debug-steps" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "CBC: Hemoglobin 10.2 g/dL (Low), WBC 11,200 /uL (High)"
  }'
```

7. Process Image Input

```
curl -X POST "http://localhost:8000/api/v1/process-image" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@sample_medical_report.png"
```

8. Debug Processing Steps - Image

```
curl -X POST "http://localhost:8000/api/v1/debug-steps-image" \
  -H "Content-Type: multipart/form-data" \
  -F "file=@sample_medical_report.png"
```

PROF

9. Error Case - No Medical Data (Should return "unprocessed")

```
curl -X POST "http://localhost:8000/api/v1/process-text" \
  -H "Content-Type: application/json" \
  -d '{
    "text": "This is just regular text without any medical information."
  }'
```

Expected Response Formats

Success Response (Final Output):

```
{
  "tests": [
    {
      "name": "Hemoglobin",
      "value": 10.2,
      "unit": "g/dL",
      "status": "low",
      "ref_range": {"low": 12.0, "high": 15.0}
    },
    {
      "name": "White Blood Cell Count",
      "value": 11200.0,
      "unit": "/uL",
      "status": "high",
      "ref_range": {"low": 4000.0, "high": 11000.0}
    }
  ],
  "summary": "The lab results show a low hemoglobin level and a high
white blood cell count.",
  "explanations": [
    "Low hemoglobin may indicate that your body doesn't have enough red
blood cells.",
    "A high white blood cell count can sometimes happen when your body
is fighting something, like an infection."
  ],
  "status": "ok",
  "reason": null
}
```

Error Response:

PROF

```
{
  "status": "unprocessed",
  "reason": "No medical tests found in input text"
}
```

Demo Problem Statement Format:

```
{
  "step1_ocr_extraction": {
    "tests_raw": ["CBC: Hemglobin 10.2 g/dL (Low), WBC 11,200 /uL
(Hgh)"],
    "confidence": 0.95
  },
  "step2_normalized_tests": {
    "tests": [...],
    "normalization_confidence": 0.95
  },
  "step3_patient_friendly": {
    "summary": "...",
    "explanations": [...]
  },
  "step4_final_output": {
    "tests": [...],
    "summary": "...",
    "status": "ok"
  }
}
```

For Screen Recording Demo

Use this sequence to show all features:

1. **Health Check**: Show API is running
2. **Demo Format**: Use `/demo-problem-statement` to show exact 4-step format
3. **OCR Error Fixing**: Show input with typos gets corrected
4. **Error Handling**: Show invalid input returns "unprocessed"
5. **Image Processing**: Upload a test image (if available)
6. **Final Production**: Show clean `/process-text` endpoint

Postman Collection

Import this into Postman for easy testing:

1. Create new collection "Medical Report Simplifier"
2. Add requests with above endpoints
3. Set environment variable for base URL
4. Test all endpoints systematically

Railway Deployment Testing

PROF

Once deployed to Railway, replace localhost with your Railway URL:

```
export API_URL="https://your-project-name.railway.app"
curl $API_URL/api/v1/health
```