🚂 Railway Deployment Guide for Medical Report Simplifier

Quick Deployment Steps

1. Prepare Your Repository

Ensure you have these files in your repo:

- `requirements.txt` - Python dependencies
- `railway.json` - Railway configuration
- `Procfile` - Alternative start command
- `app/main.py` - Your FastAPI application

2. Deploy to Railway

1. **Sign up/Login to Railway**
    - Go to railway.app
    - Sign in with your GitHub account
2. **Create New Project**
    - Click "New Project"
    - Select "Deploy from GitHub repo"
    - Choose your medical-report-simplifier repository
    - Railway will automatically detect it's a Python project
3. **Configure Environment Variables**
    - In your Railway project dashboard
    - Go to "Variables" tab
    - Add these environment variables:

```
GEMINI_API_KEY=your_gemini_api_key_here
AI_TEMPERATURE=0.3
VALIDATION_CONFIDENCE_THRESHOLD=0.7
MAX_FILE_SIZE=10485760
```

4. **Deploy**
    - Railway will automatically start building and deploying
    - Build process installs Python dependencies and system packages
    - Tesseract OCR is automatically included in Railway's environment

3. Access Your Deployed API

PROF

After successful deployment:

- **API URL**: `https://your-project-name.railway.app`
- **API Documentation**: `https://your-project-name.railway.app/docs`
- **Health Check**: `https://your-project-name.railway.app/api/v1/health`

4. Test Your Deployed API

```
# Replace with your actual Railway URL
export API_URL="https://your-project-name.railway.app"

# Test health check
curl $API_URL/api/v1/health

# Test text processing
curl -X POST "$API_URL/api/v1/process-text" \
  -H "Content-Type: application/json" \
  -d '{"text": "CBC: Hemoglobin 10.2 g/dL (Low), WBC 11,200 /uL
(High)"}'

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

Troubleshooting

Common Issues

1. **Build Fails - Missing Dependencies**
   - Check `requirements.txt` has all packages
   - Ensure versions are compatible

PROF

2. **Runtime Error - Tesseract Not Found**
   - Railway includes Tesseract by default
   - If issues persist, add to `railway.json`:

```
{
  "build": {
    "builder": "NIXPACKS",
    "buildCommand": "apt-get update && apt-get install -y
tesseract-ocr"
  }
}
```

3. **Environment Variables Not Working**

    - Double-check variable names match your code
    - Restart deployment after adding variables

4. **Port Issues**
    - Ensure you use `$PORT` environment variable
    - Railway automatically assigns the port

Viewing Logs

In Railway dashboard:

- Go to "Logs" tab to see build and runtime logs
- Check for any error messages during startup

Custom Domain (Optional)

1. In Railway dashboard, go to "Settings"
2. Under "Domains", click "Generate Domain" or add custom domain
3. Railway provides HTTPS automatically

Production Checklist

Before going live:

- [ ] Environment variables configured
- [ ] API endpoints tested
- [ ] Error handling verified
- [ ] Rate limiting considered
- [ ] Monitoring setup (Railway provides basic metrics)
- [ ] Backup API key stored securely

Monitoring & Maintenance

PROF

Railway provides:

- **Automatic deployments** on git push
- **Basic metrics** (CPU, memory, requests)
- **Log aggregation**
- **Automatic HTTPS**
- **Custom domain support**

For production use, consider:

- Setting up proper monitoring (e.g., Sentry)
- Implementing rate limiting
- Adding request logging
- Setting up health check alerts

Alternative Deployment Options

If Railway doesn't work for you:

Render

1. Connect GitHub repo to Render
2. Set build command: `pip install -r requirements.txt`
3. Set start command: `uvicorn app.main:app --host 0.0.0.0 --port $PORT`

Heroku

1. Add tesseract buildpack: `heroku buildpacks:add --index 1 heroku-community/apt`
2. Create `Aptfile`: `tesseract-ocr`
3. Deploy with Procfile

DigitalOcean App Platform

1. Connect GitHub repo
2. Auto-detects Python app
3. Configure environment variables
4. Deploy

Choose Railway for the easiest setup with automatic OCR support!