Xt-EHR T7.2 Sub-team for Imaging Reports Model

Xt-EHR Analysis Platform

Document: DEPLOYMENT_CONFIG

Generated: November 05, 2025

Analysis based on PARROT v1.0 dataset and Xt-EHR FHIR Implementation Guide

Heroku/Railway/Render Deployment Configuration

==========

Required Environment Variables

1. Flask Configuration

```bash

## Essential for Flask security

SECRET_KEY=your-long-random-secret-key-here-minimum-32-characters

## Environment mode

FLASK_ENV=production DEBUG=False ```

### 2. Server Configuration

```bash

## Host and port (Heroku sets PORT automatically)

HOST=0.0.0.0 PORT=5000 # Heroku will override this

# Application entry point

FLASK_APP=flask_app/app.py ```

### 3. Application Paths

```bash

# Base directory (usually /app on Heroku)

BASE_DIR=/app ```

### Platform-Specific Setup
```

Heroku Setup

Create Heroku app (if using Heroku CLI): bash heroku create sub-team-imaging-report-model

Set environment variables: bash heroku config:set SECRET_KEY="$(python -c 'import secrets; print(secrets.token_hex(32))')" heroku config:set FLASK_ENV=production heroku config:set DEBUG=False

Deploy: bash git push heroku main

Railway Setup

Connect GitHub repository at railway.app 2. **Set environment variables** in Railway dashboard: - SECRET_KEY: Generate a secure random string - FLASK_ENV: production - DEBUG: False

Render Setup

Connect GitHub repository at render.com 2. **Build Command**: pip install -r requirements.txt 3. **Start Command**: cd flask_app && python app.py 4. **Environment Variables**: - SECRET_KEY: Generate a secure random string - FLASK_ENV: production - DEBUG: False

Security Notes

SECRET_KEY Generation

Generate a secure secret key using Python: ```python import secrets print(secrets.token_hex(32))

Copy the output and use as SECRET_KEY

```

Or use online generator (ensure it's from a trusted source): - 64 characters minimum - Mix of letters, numbers, and symbols - Never commit to version control

## Production Security Checklist

■ Strong SECRET_KEY set - ■ DEBUG=False in production - ■ FLASK_ENV=production - ■ HTTPS enforced (platform handles this) - ■ Secure cookies enabled

## Optional Configuration
```

Performance Settings

```bash

## Web server workers (for Gunicorn)

WEB_CONCURRENCY=2 MAX_WORKERS=4

## PDF generation timeout

PDF_TIMEOUT=30 MAX_PDF_SIZE=50MB ```

### Monitoring and Logging

```bash

## Log level

LOG_LEVEL=INFO

# Enable request logging

REQUEST_LOGGING=True ```

## Platform Comparison

| Platform | Free Tier | GitHub Integration | Custom Domain | SSL |

| Railway | ■ 500 hours/month | ■ Native | ■ Yes | ■ Auto |
|---|---|---|---|---|
| Render | ■ 750 hours/month | ■ Native | ■ Yes | ■ Auto |
| Heroku | ■ No longer free | ■ CLI/Git | ■ Yes | ■ Auto |

## Quick Deploy Commands

## Generate SECRET_KEY

bash python -c "import secrets; print('SECRET_KEY=' + secrets.token_hex(32))"

## Railway Deploy

```bash

## No commands needed - automatic on git push

```

Render Deploy

```bash

## No commands needed - automatic on git push
```

```

## Heroku Deploy (if using CLI)

bash heroku config:set SECRET_KEY="$(python -c 'import secrets; print(secrets.token_hex(32))')" git push heroku main

## Environment Variables Summary

| Variable | Required | Default | Description |

| `SECRET_KEY` | ■ Yes | None | Flask session encryption key |
|---|---|---|---|
| `FLASK_ENV` | ■ Yes | production | Flask environment mode |
| `DEBUG` | ■ Yes | False | Enable/disable debug mode |
| `HOST` | No | 0.0.0.0 | Server host address |
| `PORT` | No | 5000 | Server port (platform sets this) |
| `BASE_DIR` | No | /app | Application base directory |
```

`PDF_TIMEOUT`	No	30	PDF generation timeout (seconds)
`WEB_CONCURRENCY`	No	2	Number of worker processes

Troubleshooting

Common Issues

Application won't start: Check SECRET_KEY is set 2. **Static files not loading**: Ensure DEBUG=False in production 3. **PDF generation fails**: Check file permissions and timeout settings 4. **Session issues**: Verify SECRET_KEY is properly set

Debug Commands

```bash

# Check environment variables (Railway/Render dashboards)
```

Check logs for startup messages

Verify all required files are in repository

```
```