



# Component 8 Documentation Index

---

## Resource Monitoring and Management System

---

### Design, Security, and Validation Report

---



## Documentation Files

---

This folder contains comprehensive documentation following the structure of the Pharmacy Inventory Management System (PharmPal) Component 8 report.

### 1. **Component\_8\_Resource\_Monitoring\_System.md** ★ MAIN REPORT

- **Purpose:** Complete technical documentation following Component 8 format
- **Length:** ~500 lines
- **Sections:**
  - A. Design of Forms & Frontend Screenshots
  - B. Security and Validation Proof
  - C. Innovative Experiment (Automated Agent Deployment)
  - D. Database Structure and Verification
  - E. System Architecture Overview
- **Use:** Submit as Component 8 report

### 2. **Component\_8\_Quick\_Reference.md**

- **Purpose:** Quick reference guide with tables and summaries
- **Length:** Condensed version
- **Sections:**
  - Security implementation summary
  - Database tables overview
  - Key file locations

- Validation examples
- Comparison with PharmPal
- **Use:** Quick lookup during presentation

### 3. HOW\_TO\_VIEW\_USER\_DATA.md

- **Purpose:** Step-by-step guide for accessing user data
- **Sections:**
  - DB Browser for SQLite method
  - Python scripts method
  - Web dashboard method
  - Direct SQL method
- **Use:** Practical reference for data access

### 4. security\_proof.md (Existing)

- **Purpose:** Detailed security analysis
- **Sections:**
  - Authentication mechanisms
  - Vulnerability assessment
  - Security best practices
- **Use:** Security documentation reference

### 5. README.md (Main project README)

- **Purpose:** Project overview
  - **Original:** Project introduction and setup
  - **Use:** General project documentation
-



# What Was Updated Based on Component 8.pdf

---

From PharmPal Document:

Section	PharmPal	Resource Monitor
Frontend	Flutter with login, inventory, detail, OCR forms	React with login, dashboard, system detail, alert config
Security	JWT + Argon2 + Multi-tenancy	JWT + PBKDF2 + Multi-user
Validation	Pydantic + Flutter validation	Pydantic + React validation
Innovation	AI chatbot with tool use	Automated agent deployment
Database	PostgreSQL with user_id filtering	SQLite with foreign keys

Adapted Subheadings:

- ✓ **A. Design of Forms** - Mapped 4 forms from PharmPal to our 4 screens
  - ✓ **B. Security and Validation Proof** - Documented our JWT auth & password hashing
  - ✓ **C. Innovative Experiment** - Changed from AI chatbot to automated deployment
  - ✓ Added **D. Database Structure** - Our 6 tables with verification
  - ✓ Added **E. System Architecture** - Complete component overview
- 



## Key Differences from PharmPal

---

PharmPal Focus:

- Mobile app (Flutter)
- Per-user inventory isolation
- AI-powered chatbot with database tool use
- OCR and barcode scanning
- Pharmacy-specific features

## Our Resource Monitor Focus:

- Web dashboard (React)
- Shared system monitoring
- Automated zero-config agent deployment
- Real-time metrics collection
- IT management features

## Similar Security Principles:

- Both use JWT authentication
- Both use strong password hashing
- Both implement backend validation with Pydantic
- Both use FastAPI backend
- Both prevent SQL injection with ORM



## Database Verification

### Quick Commands:

```
# View users in database
python inspect_db.py

# Detailed user information
python view_all_user_data.py

# Open database in GUI
open_database.bat

# View all tables
python inspect_s_admins.py
```

## Current Database Status:

Database: backend/resource\_monitor.db

Tables: 6 (users, systems, metrics, alerts, tickets, alert\_settings)

Users: 2 (nthy2355@gmail.com, Admin1@gmail.com)

Password Hashing: PBKDF2-SHA256

Status: ✓ Fully functional

---

## Project File Structure

```

Resource-Monitoring-and-Management-System/
├── backend/
│   ├── app/
│   │   ├── api/
│   │   │   ├── auth.py           ← JWT authentication
│   │   │   └── endpoints.py      ← Main API
│   │   ├── core/
│   │   │   ├── security.py       ← Password hashing
│   │   │   ├── alerts.py         ← Alert system
│   │   │   └── discovery.py      ← Auto-discovery
│   │   ├── models/
│   │   │   └── models.py         ← 6 database tables
│   │   └── schemas/
│   │       └── schemas.py        ← Pydantic validation
│   └── resource_monitor.db       ← SQLite database
├── dashboard/
│   └── src/
│       ├── pages/
│       │   ├── Login.jsx         ← Login form
│       │   ├── Dashboard.jsx     ← Main dashboard
│       │   └── SystemDetail.jsx  ← System details
│       └── services/
│           └── api.js            ← API calls
├── agent/
│   ├── main.py                  ← Agent logic
│   └── gui.py                   ← System tray
├── setup_lab.py                 ← Auto IP detection
├── Component_8_Resource_Monitoring_System.md ← MAIN REPORT
├── Component_8_Quick_Reference.md  ← Quick guide
├── HOW_TO_VIEW_USER_DATA.md      ← Data access guide
└── security_proof.md             ← Security details

```



## Quick Start for Review

---

### 1. Read Main Report

```
# Open in your editor  
Component_8_Resource_Monitoring_System.md
```

### 2. Verify Database

```
python view_all_user_data.py
```

### 3. View in DB Browser

```
open_database.bat
```

### 4. Check Security Implementation

```
# View authentication code  
code backend/app/api/auth.py  
  
# View password hashing  
code backend/app/core/security.py  
  
# View database models  
code backend/app/models/models.py
```



## Component 8 Requirements Checklist

---

Based on the PharmPal document structure:

- [x] **Design of Forms** - 4 key interfaces documented
- [x] **Security Proof**
- [x] Authentication (JWT)

- [x] Password hashing
- [x] Database constraints
- [x] API protection
- [x] **Validation Proof**
- [x] Frontend validation
- [x] Backend validation (Pydantic)
- [x] Database validation
- [x] **Innovative Experiment**
- [x] Description of innovation
- [x] Implementation details
- [x] Architecture benefits
- [x] **Code Examples** - Included throughout
- [x] **Database Verification** - Scripts provided
- [x] **Screenshots** - Described (can add real screenshots)



## Notes for Submission

---

### What to Submit:

1. **Component\_8\_Resource\_Monitoring\_System.md** (Main report)
2. **Component\_8\_Quick\_Reference.md** (Optional - for presentation)
3. **Screenshots** (if required - can be taken from running application)

### Supporting Files (for verification):

- `inspect_db.py` - Shows user table
- `view_all_user_data.py` - Detailed user data
- `open_database.bat` - Opens DB Browser
- `component8_extracted.txt` - Original PharmPal document for reference



## Related Documentation

---

- **Main Project README:** `README.md`
- **Security Analysis:** `security_proof.md`



- **User Data Guide:** [HOW\\_TO\\_VIEW\\_USER\\_DATA.md](#)
- **Original Component 8:** [Component 8.pdf](#) (Pharmacy system - for reference)

---

## Team Information

---

Update these fields in the main report:

Submitted by  
[Your Name]  
[Roll Number]

Current template uses placeholders - replace with actual information.

---

## Questions & Support

---

For questions about this documentation:

1. **Database access:** See [HOW\\_TO\\_VIEW\\_USER\\_DATA.md](#)
2. **Security details:** See [security\\_proof.md](#)
3. **Technical implementation:** See [Component\\_8\\_Resource\\_Monitoring\\_System.md](#)

---

**Last Updated:** December 27, 2025

**Project:** Resource Monitoring and Management System

**Based on:** Component 8.pdf (Pharmacy Inventory Management System)

**Status:** ✓ Ready for submission

---



# Document Statistics

Document	Lines	Purpose	Status
Component_8_Resource_Monitoring_System.md	~500	Main report	✓ Complete
Component_8_Quick_Reference.md	~200	Quick guide	✓ Complete
HOW_TO_VIEW_USER_DATA.md	~150	Data access	✓ Complete
security_proof.md	~100	Security	✓ Existing
INDEX.md (this file)	~250	Navigation	✓ Complete

**Total documentation:** ~1,200 lines covering all aspects of design, security, and validation.

**This completes the Component 8 documentation for your Resource Monitoring and Management System!** 🎉