Information Security - Assignment #4

GDPR-Compliant Mini Hospital Management System

Course Instructor: Usama Antuley

Group Members: M. Talha Yousif — 22K5146
Rehan Khan — 22K5031

Department of Computer Science

November 22, 2025

Contents

Overview

This report documents the implementation of a **GDPR-Compliant Mini Hospital Management System** built with Python, Streamlit, and SQLite. The system demonstrates privacy-centric design principles following GDPR compliance requirements, implementing robust security features including encryption, role-based access control, and comprehensive audit logging.

Key Technologies:

- **Backend:** Python 3.8+, SQLite Database
- **Frontend:** Streamlit Web Framework
- **Security:** Fernet Encryption, SHA-256 Hashing
- **Visualization:** Matplotlib, Pandas

`Note:` All Screenshots can be found at the End of the document.

1 System Features

1.1 Security & Privacy Features

The system implements comprehensive security measures to protect sensitive patient data:

Fernet Symmetric Encryption — All sensitive diagnosis data is encrypted at rest using AES-128 encryption via the Fernet library.

SHA-256 Password Hashing — User passwords are never stored in plaintext; all credentials are hashed using SHA-256.

Role-Based Access Control (RBAC) — Three distinct user roles with granular permissions control data access.

1.2 User Roles and Permissions

Admin	Full system access including decryption capabilities, patient anonymization, audit log viewing, and data export.
Doctor	View anonymized patient data only. Cannot access decrypted diagnoses or modify records.
Receptionist	Add and edit patient records with limited access. Cannot view diagnosis information.

1.3 Patient Management

The system provides comprehensive patient data management:

- Add, edit, and view patient records
- Encrypted storage of medical diagnoses
- Patient anonymization (GDPR Right to Erasure)
- Age and gender demographics tracking

1.4 Audit Trail

All system actions are logged with:

- Comprehensive logging of all data access and modifications
- Timestamp and user tracking for accountability
- Action type categorization for filtering
- Exportable audit logs for compliance reporting

1.5 Analytics & Visualization

The system includes built-in analytics dashboards:

- Activity timeline charts
- Role-based action distribution
- Hourly activity patterns
- Patient age distribution histograms
- Gender demographics pie charts

2 System Architecture

2.1 CIA Triad Implementation

The system adheres to the CIA Triad security principles:

Confidentiality	Fernet encryption for sensitive data, role-based access control, data masking for non-privileged users
Integrity	Comprehensive audit logging, transaction management, input validation on all forms
Availability	Robust error handling, database backup capabilities, CSV export for data portability

2.2 Database Schema

The SQLite database consists of three main tables:

Users Table

```
CREATE TABLE users ( user_id INTEGER PRIMARY KEY, username TEXT UNIQUE NOT NULL,
password_hash TEXT NOT NULL, role TEXT NOT NULL, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP );
```

Patients Table

```
CREATE TABLE patients ( patient_id INTEGER PRIMARY KEY, name TEXT NOT NULL, age
INTEGER, gender TEXT, contact TEXT, diagnosis TEXT, diagnosis_encrypted BLOB,
admission_date DATE, is_anonymized BOOLEAN DEFAULT 0 );
```

Logs Table

```
CREATE TABLE logs ( log_id INTEGER PRIMARY KEY, user_id INTEGER REFERENCES
users(user_id), role TEXT, action TEXT NOT NULL, timestamp TIMESTAMP DEFAULT
CURRENT_TIMESTAMP, details TEXT );
```

3 GDPR Compliance Implementation

3.1 Article 5 — Data Processing Principles

Lawfulness	Clear role-based access ensures only authorized personnel access data
Purpose Limitation	Data collected exclusively for patient care purposes
Data Minimization	Only essential patient data is stored
Accuracy	Edit functionality maintains data accuracy
Storage Limitation	Anonymization removes identifiable data
Integrity & Confidentiality	Encryption and access controls protect data

3.2 Article 17 — Right to Erasure

The system implements patient anonymization functionality that permanently removes all identifiable information while preserving statistical data for research purposes.

```python
def anonymize_patient(patient_id):
    # Replaces name with "ANONYMIZED"
    # Clears contact information
    # Removes encrypted diagnosis
    # Sets is_anonymized flag to True
    # Logs anonymization action
```

3.3 Article 30 — Record of Processing Activities

The comprehensive audit log tracks:

- Every data access event with timestamp
- User identification and role
- Action type (VIEW, ADD, EDIT, DELETE, DECRYPT)
- Detailed description of the action

3.4 Article 32 — Security of Processing

Encryption at Rest	Fernet (AES-128) for diagnosis data
Password Hashing	SHA-256 for user credentials
Access Control	Role-based permissions system
Audit Trail	Complete logging for accountability

4 Security Features Implementation

4.1 Encryption System

The system uses Fernet symmetric encryption for protecting sensitive diagnosis data:

```python
from cryptography.fernet import Fernet

def generate_key():
    """Generate and save encryption key"""
    key = Fernet.generate_key()
    with open('fernet.key', 'wb') as key_file:
        key_file.write(key)
    return key

def encrypt_text(plaintext):
    """Encrypt sensitive data"""
    fernet = Fernet(load_key())
    return fernet.encrypt(plaintext.encode())

def decrypt_text(ciphertext):
    """Decrypt sensitive data (Admin only)"""
    fernet = Fernet(load_key())
    return fernet.decrypt(ciphertext).decode()
```

4.2 Data Masking

Non-privileged users see masked versions of sensitive data:

```python
# Name masking: "John Doe" -> "J**n D*e"
def mask_name(name):
    # Masks middle characters of each word

# Contact masking: "+1234567890" -> "+12****7890"
def mask_contact(contact):
    # Shows only first 3 and last 4 characters
```

4.3 Role-Based Access Matrix

Feature	Admin	Doctor	Receptionist
View Patients	Full	Masked	Basic
Decrypt Diagnosis			
Add Patient			
Edit Patient			Limited
Anonymize Patient			
View Audit Logs			
Export Data			

5 Project Structure

```
hospital_management/
|
|-- db_init.py              Database initialization
|-- encryption_utils.py     Encryption/decryption functions
|-- auth.py                 Authentication & RBAC
|-- db_helpers.py           Database CRUD operations
|-- graphs.py               Data visualization
|-- streamlit_app.py        Main UI application
|
|-- requirements.txt        Python dependencies
|-- README.md               Documentation
|
|-- hospital.db            SQLite database (auto-generated)
+-- fernet.key             Encryption key (auto-generated)
```

5.1 Dependencies

```
streamlit # Web UI framework
pandas # Data manipulation
matplotlib # Data visualization
cryptography # Fernet encryption
```

6 Installation and Usage

6.1 Prerequisites

- Python 3.8 or higher
- pip (Python package manager)

6.2 Installation Steps

Step 1: Install Dependencies

```
pip install -r requirements.txt
```

Step 2: Initialize Database

```
python db_init.py
```

This creates:

- `hospital.db` — SQLite database
- `fernet.key` — Encryption key
- Default user accounts
- Database schema

Step 3: Start Application

```
streamlit run streamlit_app.py
```

The application opens at `http://localhost:8501`

6.3 Default Login Credentials

Role	Username	Password
Admin	`admin`	`admin123`
Doctor	`doctor1`	`doctor123`
Receptionist	`receptionist1`	`recept123`

7 Key Workflows

7.1 Adding a Patient (Receptionist)

1. Login as receptionist
2. Navigate to " Add Patient"
3. Fill patient details (name, age, gender, contact, diagnosis)
4. Submit form
5. Patient data encrypted and action logged

7.2 Viewing Sensitive Data (Admin)

1. Login as admin
2. Navigate to " Manage Patients"
3. Select " Decrypted View"
4. View decrypted diagnoses
5. Access logged in audit trail

7.3 Anonymizing Patient (GDPR Compliance)

1. Login as admin
2. Navigate to " Anonymize Patient"
3. Enter patient ID
4. Confirm anonymization
5. Patient data permanently anonymized

7.4 Exporting Data

1. Login as admin
2. Navigate to " Export Data"
3. Select export type (Patients or Audit Logs)
4. Download CSV file
5. GDPR-compliant data portability

8 Security Considerations

8.1 Production Deployment Checklist

Important Security Notes:

Change all default passwords before deployment

Use secure key management (AWS KMS, Azure Key Vault)

Implement HTTPS/TLS for all communications

Add rate limiting to prevent brute force attacks

Enable regular database backups

Set up monitoring and security alerts

Implement 2FA for admin accounts

Add input sanitization for SQL injection prevention

Implement session timeout

Add CAPTCHA for login attempts

8.2 Key Protection

Encryption Key Security:

- The `fernet.key` file contains the master encryption key
- Never commit this file to version control
- In production, use cloud-based key management services
- Implement key rotation policies

Conclusion

This project successfully demonstrates the implementation of a GDPR-compliant hospital management system that prioritizes patient privacy and data security. The system effectively implements:

- Strong encryption for sensitive medical data
- Role-based access control for granular permissions
- Comprehensive audit logging for accountability
- Patient anonymization for GDPR Right to Erasure
- Data export capabilities for portability compliance

Key Takeaway: By implementing proper encryption, access controls, audit logging, and anonymization features, healthcare applications can maintain both functionality and compliance with stringent data protection regulations like GDPR.

Appendix — Screenshots

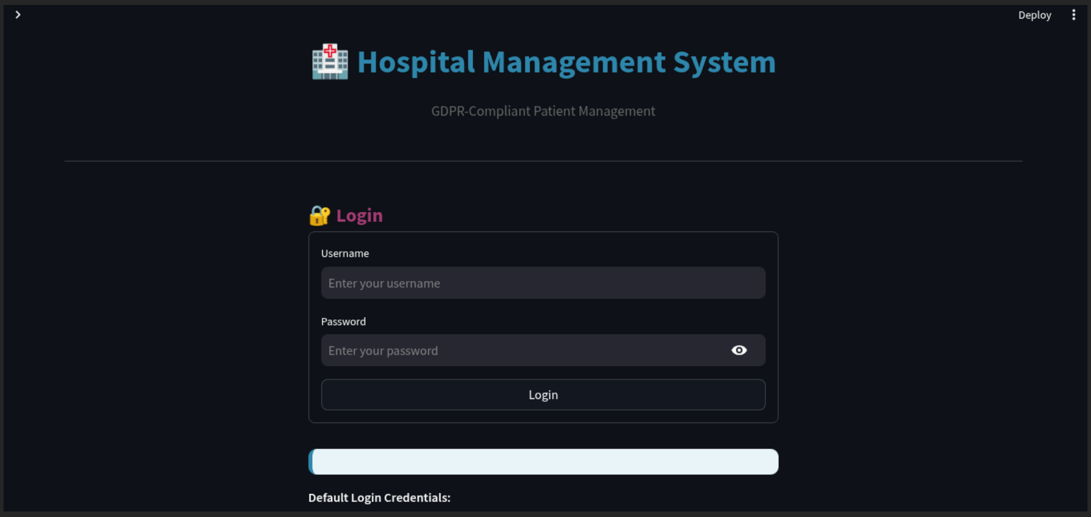
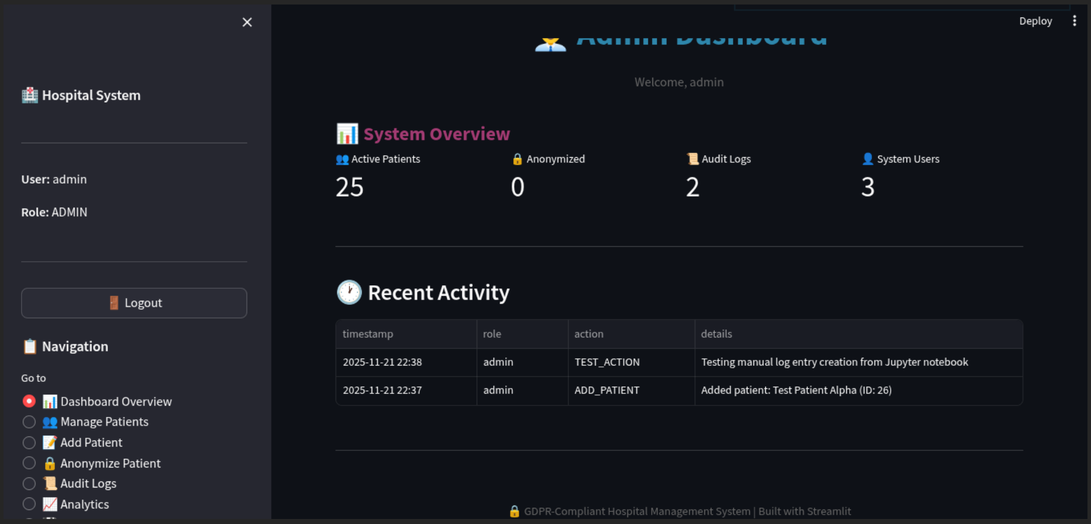
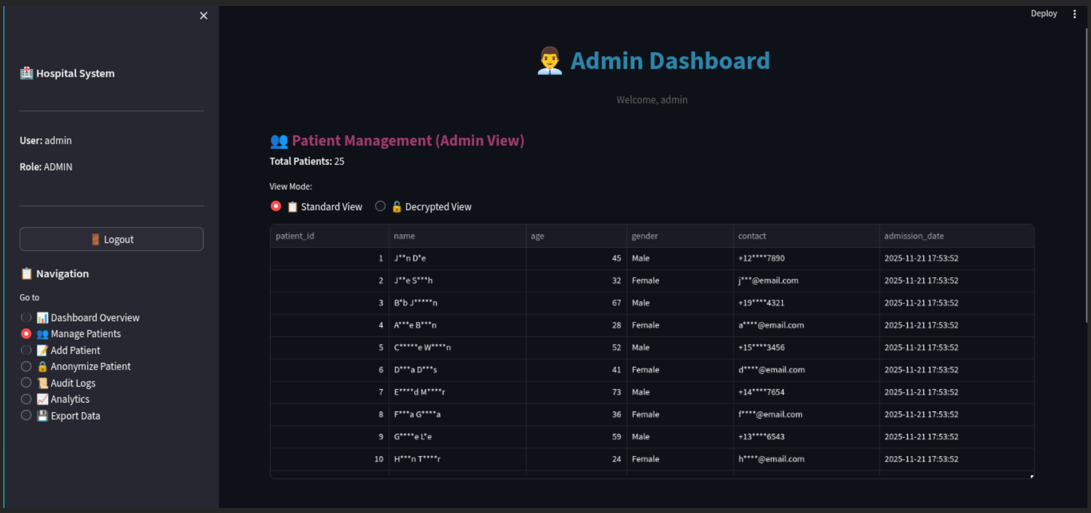

Deploy
🏥 Hospital Management System
GDPR-Compliant Patient Management
🔐 Login
Username
Enter your username
Password
Enter your password
Login
Default Login Credentials:

Figure 1: Login Page with Role-Based Authentication



timestamp	role	action	details
2025-11-21 22:38	admin	TEST_ACTION	Testing manual log entry creation from Jupyter notebook
2025-11-21 22:37	admin	ADD_PATIENT	Added patient: Test Patient Alpha (ID: 26)

Figure 2: Admin Dashboard with System Overview



Hospital System

User: admin

Role: ADMIN

Logout

Navigation

Go to

- Dashboard Overview
- [✓ Manage Patients]
- Add Patient
- Anonymize Patient
- Audit Logs
- Analytics
- Export Data

Deploy

Admin Dashboard

Welcome, admin

Patient Management (Admin View)

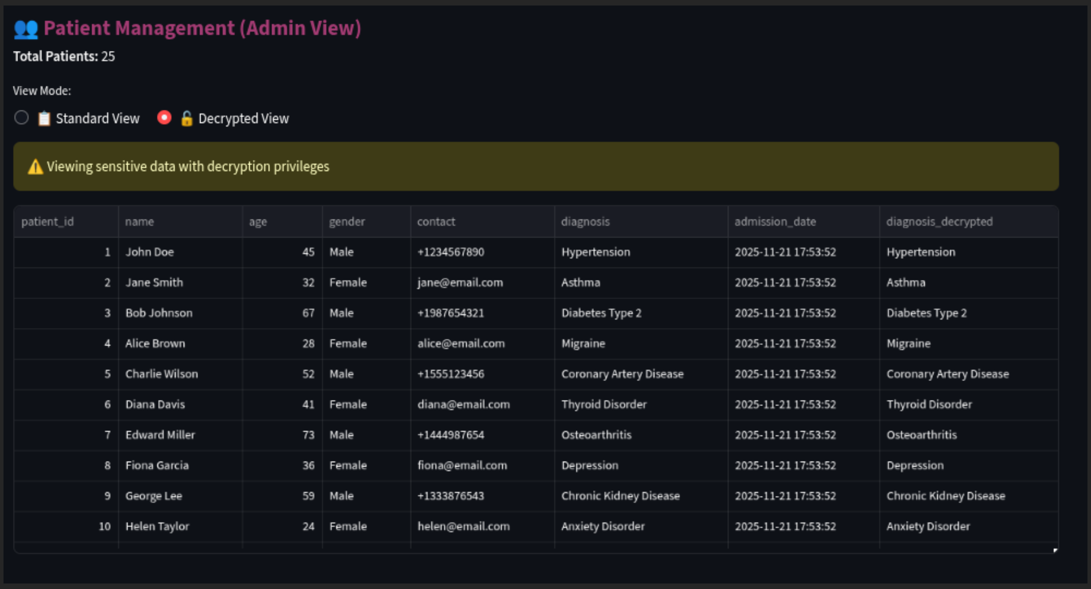
Total Patients: 25

View Mode:

[✓ Standard View] Decrypted View

patient_id	name	age	gender	contact	admission_date
1	J**n D*e	45	Male	+12****7890	2025-11-21 17:53:52
2	J**e S***h	32	Female	j***@email.com	2025-11-21 17:53:52
3	B*b J*****n	67	Male	+19****4321	2025-11-21 17:53:52
4	A***e B***n	28	Female	a****@email.com	2025-11-21 17:53:52
5	C*****e W****n	52	Male	+15****3456	2025-11-21 17:53:52
6	D***a D***s	41	Female	d****@email.com	2025-11-21 17:53:52
7	E****d M****r	73	Male	+14****7654	2025-11-21 17:53:52
8	F***a G****a	36	Female	f****@email.com	2025-11-21 17:53:52
9	G****e L*e	59	Male	+13****6543	2025-11-21 17:53:52
10	H***n T****r	24	Female	h****@email.com	2025-11-21 17:53:52

Figure 3: Patient List with Encrypted Data View



Patient Management (Admin View)

Total Patients: 25

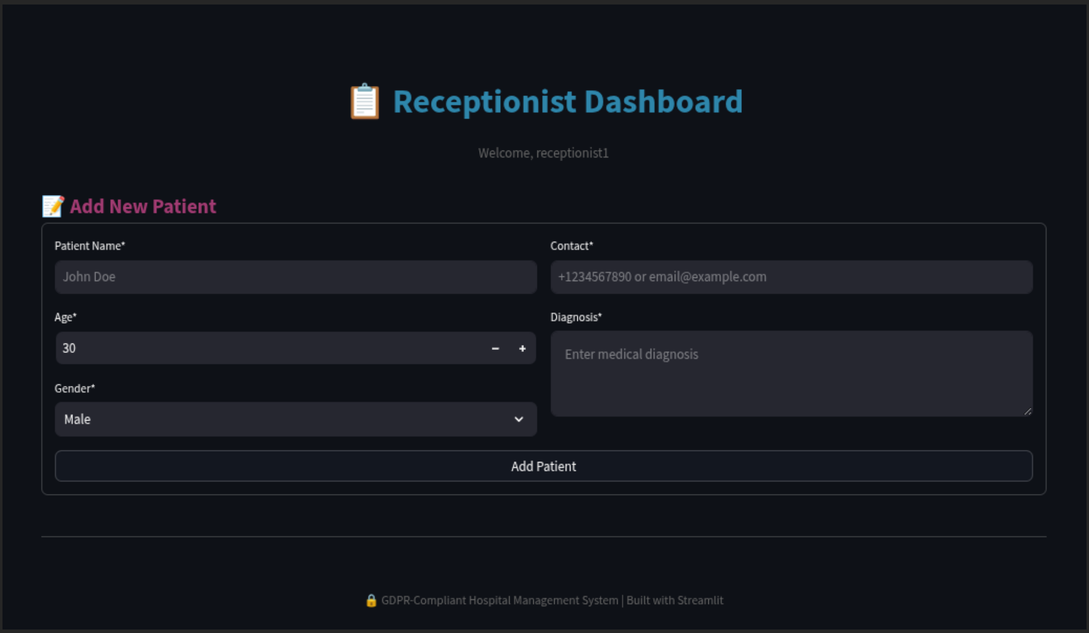
View Mode:

Standard View [✓ Decrypted View]

⚠️ Viewing sensitive data with decryption privileges

patient_id	name	age	gender	contact	diagnosis	admission_date	diagnosis_decrypted
1	John Doe	45	Male	+1234567890	Hypertension	2025-11-21 17:53:52	Hypertension
2	Jane Smith	32	Female	jane@email.com	Asthma	2025-11-21 17:53:52	Asthma
3	Bob Johnson	67	Male	+1987654321	Diabetes Type 2	2025-11-21 17:53:52	Diabetes Type 2
4	Alice Brown	28	Female	alice@email.com	Migraine	2025-11-21 17:53:52	Migraine
5	Charlie Wilson	52	Male	+1555123456	Coronary Artery Disease	2025-11-21 17:53:52	Coronary Artery Disease
6	Diana Davis	41	Female	diana@email.com	Thyroid Disorder	2025-11-21 17:53:52	Thyroid Disorder
7	Edward Miller	73	Male	+1444987654	Osteoarthritis	2025-11-21 17:53:52	Osteoarthritis
8	Fiona Garcia	36	Female	fiona@email.com	Depression	2025-11-21 17:53:52	Depression
9	George Lee	59	Male	+1333876543	Chronic Kidney Disease	2025-11-21 17:53:52	Chronic Kidney Disease
10	Helen Taylor	24	Female	helen@email.com	Anxiety Disorder	2025-11-21 17:53:52	Anxiety Disorder

Figure 4: Admin Decrypted View of Patient Diagnoses



📋 Receptionist Dashboard

Welcome, receptionist1

📝 Add New Patient

Patient Name*: John Doe

Contact*: +1234567890 or email@example.com

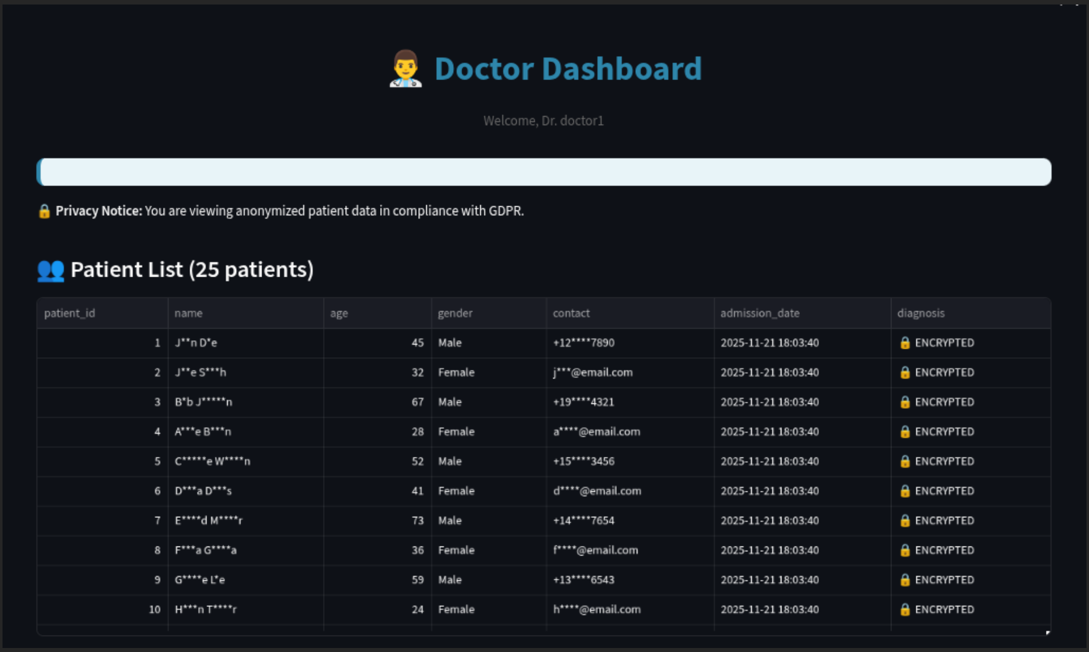
Age*: 30

Diagnosis*: Enter medical diagnosis

Gender*: Male

🔒 GDPR-Compliant Hospital Management System | Built with Streamlit

Figure 5: Add Patient Form (Receptionist View)



👨‍⚕️ Doctor Dashboard

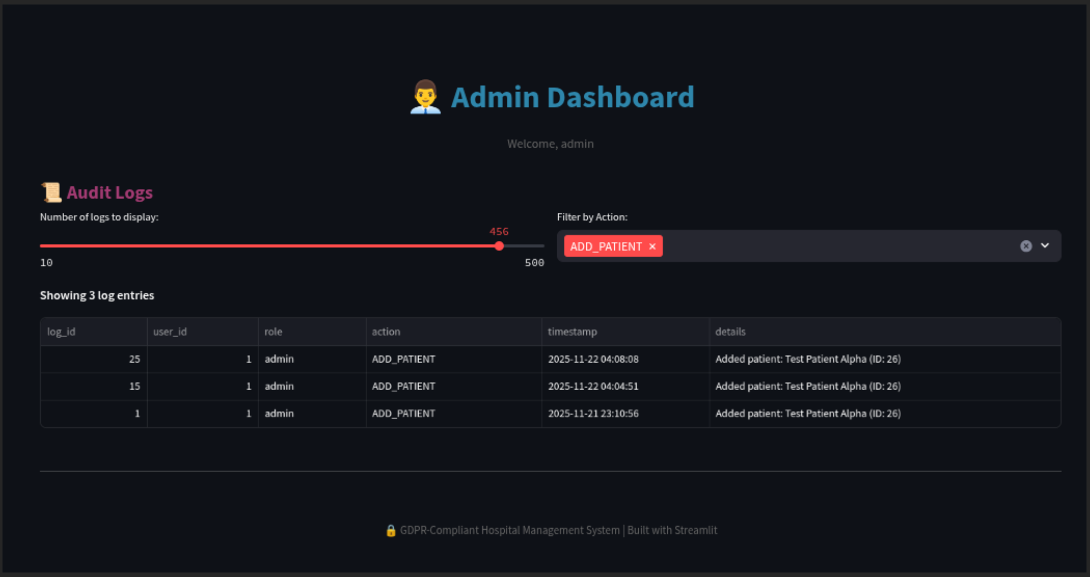
Welcome, Dr. doctor1

🔒 **Privacy Notice:** You are viewing anonymized patient data in compliance with GDPR.

👥 Patient List (25 patients)

patient_id	name	age	gender	contact	admission_date	diagnosis
1	J**n D*e	45	Male	+12****7890	2025-11-21 18:03:40	🔒 ENCRYPTED
2	J**e S***h	32	Female	j***@email.com	2025-11-21 18:03:40	🔒 ENCRYPTED
3	B*b J*****n	67	Male	+19****4321	2025-11-21 18:03:40	🔒 ENCRYPTED
4	A***e B***n	28	Female	a****@email.com	2025-11-21 18:03:40	🔒 ENCRYPTED
5	C*****e W****n	52	Male	+15****3456	2025-11-21 18:03:40	🔒 ENCRYPTED
6	D***a D***s	41	Female	d****@email.com	2025-11-21 18:03:40	🔒 ENCRYPTED
7	E****d M****r	73	Male	+14****7654	2025-11-21 18:03:40	🔒 ENCRYPTED
8	F***a G****a	36	Female	f****@email.com	2025-11-21 18:03:40	🔒 ENCRYPTED
9	G****e L*e	59	Male	+13****6543	2025-11-21 18:03:40	🔒 ENCRYPTED
10	H***n T****r	24	Female	h****@email.com	2025-11-21 18:03:40	🔒 ENCRYPTED

Figure 6: Patient Anonymization Feature (GDPR Compliance)


Admin Dashboard

Welcome, admin

Audit Logs

Number of logs to display: 456 (10–500)

Filter by Action: ADD_PATIENT

Showing 3 log entries

log_id	user_id	role	action	timestamp	details
25	1	admin	ADD_PATIENT	2025-11-22 04:08:08	Added patient: Test Patient Alpha (ID: 26)
15	1	admin	ADD_PATIENT	2025-11-22 04:04:51	Added patient: Test Patient Alpha (ID: 26)
1	1	admin	ADD_PATIENT	2025-11-21 23:10:56	Added patient: Test Patient Alpha (ID: 26)

🔒 GDPR-Compliant Hospital Management System | Built with Streamlit

Figure 7: Comprehensive Audit Log View

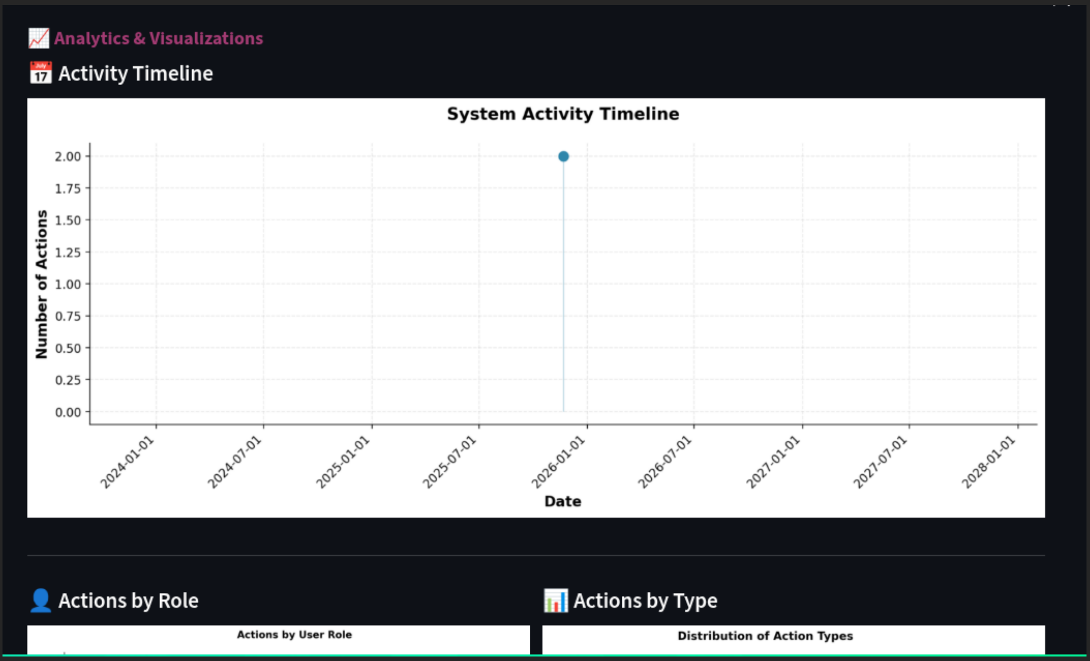
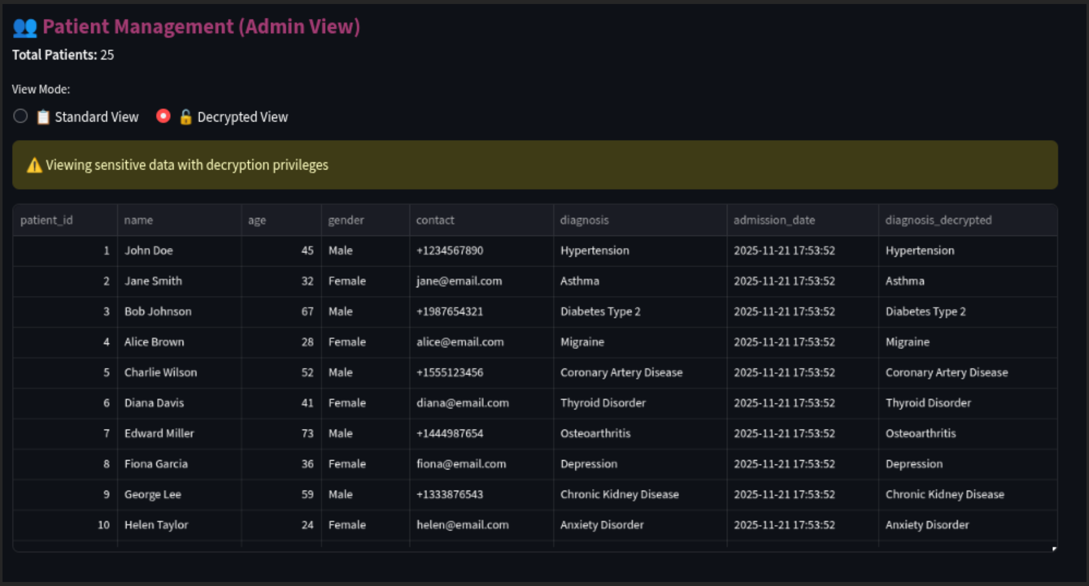


Figure 8: Analytics Dashboard with Visualizations



Patient Management (Admin View)

Total Patients: 25

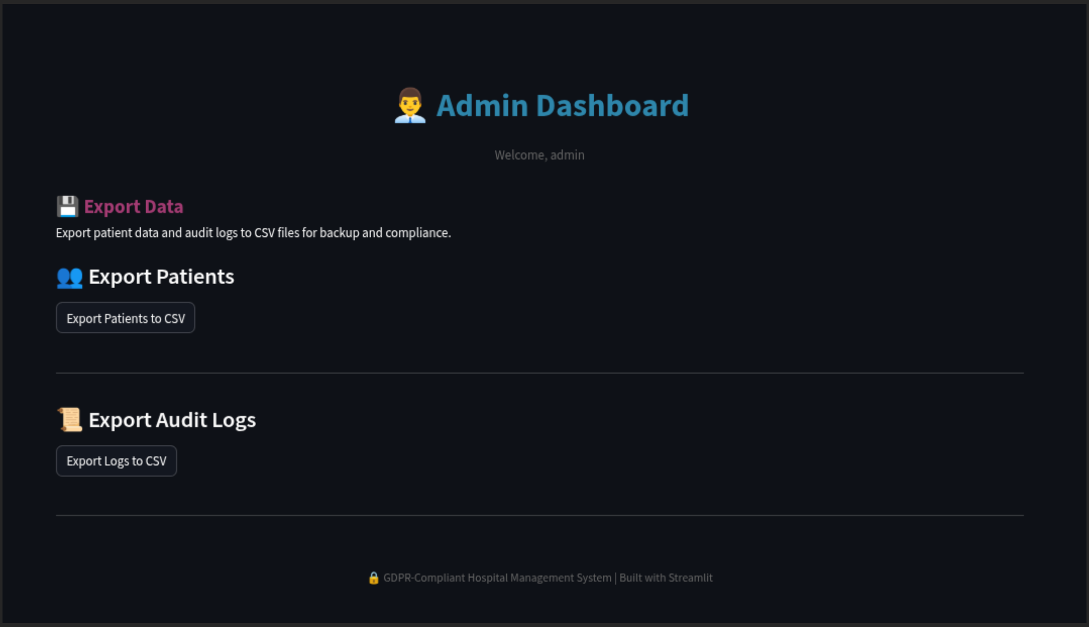
View Mode:

Standard View [✓ Decrypted View]

⚠ Viewing sensitive data with decryption privileges

patient_id	name	age	gender	contact	diagnosis	admission_date	diagnosis_decrypted
1	John Doe	45	Male	+1234567890	Hypertension	2025-11-21 17:53:52	Hypertension
2	Jane Smith	32	Female	jane@email.com	Asthma	2025-11-21 17:53:52	Asthma
3	Bob Johnson	67	Male	+1987654321	Diabetes Type 2	2025-11-21 17:53:52	Diabetes Type 2
4	Alice Brown	28	Female	alice@email.com	Migraine	2025-11-21 17:53:52	Migraine
5	Charlie Wilson	52	Male	+1555123456	Coronary Artery Disease	2025-11-21 17:53:52	Coronary Artery Disease
6	Diana Davis	41	Female	diana@email.com	Thyroid Disorder	2025-11-21 17:53:52	Thyroid Disorder
7	Edward Miller	73	Male	+1444987654	Osteoarthritis	2025-11-21 17:53:52	Osteoarthritis
8	Fiona Garcia	36	Female	fiona@email.com	Depression	2025-11-21 17:53:52	Depression
9	George Lee	59	Male	+1333876543	Chronic Kidney Disease	2025-11-21 17:53:52	Chronic Kidney Disease
10	Helen Taylor	24	Female	helen@email.com	Anxiety Disorder	2025-11-21 17:53:52	Anxiety Disorder

Figure 9: Doctor Dashboard with Masked Patient Data



Admin Dashboard

Welcome, admin

Export Data

Export patient data and audit logs to CSV files for backup and compliance.

Export Patients

Export Patients to CSV

Export Audit Logs

Export Logs to CSV

GDPR-Compliant Hospital Management System | Built with Streamlit

Figure 10: Data Export Feature for GDPR Portability