# GDPR-Compliant Hospital Management System

## Information Security (CS-3002) - Assignment #4


### Group Members:

1. Riyyan Siddiqui _ 22k-4452
2. Muhammad Fasih _ 22k-4486


# 1. System Overview

### Project Description

A comprehensive, privacy-centric hospital management system that implements the CIA Triad (Confidentiality, Integrity, Availability) while ensuring GDPR compliance. The system manages patient records with role-based access control, data anonymization, and complete audit logging.

### Technology Stack

- **Frontend**: Streamlit (Python web framework)
- **Backend**: Python 3.8+
- **Database**: MySQL 8.0
- **Security**:
    - Cryptography library (Fernet encryption)
    - hashlib (SHA-256 password hashing)
    - Role-Based Access Control (RBAC)

### System Architecture Diagram

# 2. CIA Triad Implementation

### 2.1 Confidentiality (C) - Data Protection & Privacy

**Implementation Details:**

https://drive.google.com/file/d/1bQM-8i7_5nvZDZsNRD1Bnm8POufIxDht/view?usp=sharing

### A. Role-Based Access Control (RBAC)

- **Admin**: Full access to both raw and anonymized data, system configuration
- **Doctor**: Access only to anonymized patient records and diagnoses
- **Receptionist**: Can add/edit records but cannot view sensitive information

### B. Data Anonymization

```
# Name Anonymization
Name: "John Doe" → "ANON_0001"

# Contact Anonymization
Contact: "123-456-7890" → "XXX-XXX-7890"
```

### C. Fernet Encryption (Bonus Feature)

- Symmetric encryption using Cryptography library
- Reversible encryption for data recovery
- Encrypted data stored in separate columns
- Key: `8cozhW9kSi6zJQw3xLvMp_6T3Nq3qjWPHvXFnwi4IxE=`

### D. Password Security

- SHA-256 hashing for all passwords
- No plain-text password storage
- Salted hashing for enhanced security

### Code Snippet:

```
def encrypt_data(data):
    f = Fernet(ENCRYPTION_KEY)
    return f.encrypt(data.encode()).decode()

def anonymize_name(name, patient_id):
    return f"ANON_{patient_id:04d}"

def anonymize_contact(contact):
    return "XXX-XXX-" + contact[-4:]
```

## 2.2 Integrity (I) - Data Accuracy & Accountability

### Implementation Details:

**A. Comprehensive Activity Logging** Every action is logged with:

- User ID and username
- User role
- Action type (Login, Add Patient, View, Anonymize, etc.)
- Timestamp
- Action details

**B. Audit Trail Features**

- Searchable logs by action type
- Filterable by date and user
- Export capability for compliance audits
- Real-time activity monitoring

**C. Database Integrity**

- Foreign key constraints
- Data validation before insertion
- Transaction management
- Rollback on errors

**Logged Actions:**

- User Login/Logout
- Patient Record Addition
- Data Anonymization (Individual & Bulk)
- Patient Data Viewing
- GDPR Consent
- System Backup
- Data Deletion

**Code Snippet:**

```
def log_activity(user_id, role, action, details=""):
    cursor.execute(
        "INSERT INTO logs (user_id, role, action, details) VALUES (%s, %s,
%s, %s)",
        (user_id, role, action, details)
    )
```

## 2.3 Availability (A) - System Access & Reliability

**Implementation Details:**

**A. Error Handling**

```
try:
    connection = create_connection()
    # Database operations
```

```
except Error as e:
    st.error(f"Error: {e}")
    # Graceful degradation
```

**B. Data Backup & Recovery**

- CSV export for all tables
- One-click backup creation
- Separate exports for patients and logs
- Timestamped backup files

**C. System Monitoring**

- Real-time uptime tracking
- Last synchronization time display
- Connection status monitoring
- Responsive dashboard design

**D. Database Connection Management**

- Connection pooling
- Automatic reconnection
- Timeout handling
- Resource cleanup

# 3. GDPR Compliance

## 3.1 GDPR Principles Implementation

| Principle | Implementation |
|---|---|
| **Lawfulness, Fairness, Transparency** | ⇨ User consent banner on first access |
| **Purpose Limitation** | ⇨ Data used only for healthcare purposes |
| **Data Minimization** | ⇨ Only essential patient data collected |
| **Accuracy** | ⇨ Integrity logs maintain data accuracy |
| **Storage Limitation** | ⇨ 90-day retention policy with auto-deletion |
| **Integrity & Confidentiality** | ⇨ Encryption and anonymization active |
| **Accountability** | ⇨ Complete audit trail of all processing |

## 3.2 GDPR Features (Bonus)

**A. Data Retention Timer**

- Automatic 90-day retention period
- `data_retention_date` column in patients table

- Visual warnings for expiring records
- One-click deletion of expired data

**B. User Consent Banner**

- Displayed on first login
- Explains data processing activities
- Requires explicit consent to proceed
- Consent logged in audit trail

**C. Right to be Forgotten**

- Admin can delete expired records
- Bulk deletion capability
- Complete data removal from system
- Action logged for compliance

### 3.3 GDPR Compliance Dashboard

Features include:

- Real-time retention monitoring
- Expired records counter
- Compliance checklist
- Encryption status
- Data processing activities log

# 4. Database Schema

## 4.1 Users Table

```
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,  -- SHA-256 hashed
    role ENUM('admin', 'doctor', 'receptionist') NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

**Default Users:**

- admin / admin123 (Role: admin)
- dr_bob / doc123 (Role: doctor)
- alice_recep / rec123 (Role: receptionist)

## 4.2 Patients Table

```
CREATE TABLE patients (
    patient_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(200) NOT NULL,
    contact VARCHAR(50) NOT NULL,
    diagnosis TEXT,
    anonymized_name VARCHAR(50),
    anonymized_contact VARCHAR(50),
    encrypted_name TEXT,           -- Fernet encrypted
    encrypted_contact TEXT,        -- Fernet encrypted
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    data_retention_date DATE,      -- GDPR compliance
    is_anonymized BOOLEAN DEFAULT FALSE
);
```

### 4.3 Logs Table

```
CREATE TABLE logs (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    role VARCHAR(50),
    action VARCHAR(255),
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    details TEXT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

# 5. Feature Implementation

## 5.1 Core Features

#### Login & Authentication

- Secure login page with credential validation
- SHA-256 password hashing
- Session state management
- Failed login attempt logging

#### Dashboard Overview

- Real-time metrics display
- Total patients counter
- Anonymization percentage
- Daily activity count
- Expired records alert

#### Patient Management

- Add new patients (Receptionist)
- View patients (Role-dependent access)
- Anonymize individual records (Admin)

- Bulk anonymization (Admin)
- CSV export functionality

**Audit Logs**

- Comprehensive activity tracking
- Filter by action type
- Filter by date
- Export to CSV
- Admin-only access

### 5.2 Bonus Features

**Real-Time Analytics (+0.5)**

1. **Daily Activity Chart**
    - Line graph showing actions per day
    - Last 7 days visualization
    - Interactive Plotly charts
2. **Action Distribution**
    - Pie chart of action types
    - Percentage breakdown
    - Color-coded categories
3. **Activity Heatmap**
    - Hour-by-hour activity
    - Date-wise comparison
    - Color intensity mapping

**Fernet Encryption (+1)**

- Reversible encryption option
- Checkbox during patient addition
- Separate encrypted columns
- Decryption capability for admin

**GDPR Features (+0.5)**

- 90-day retention timer
- User consent banner
- Expired record deletion
- Compliance checklist
- Data processing transparency

## 6. User Workflows

### 6.1 Admin Workflow

1. Login with admin credentials
2. Accept GDPR consent banner
3. View dashboard overview
4. Check expired records
5. Anonymize patient data (individual or bulk)
6. Review audit logs
7. Analyze real-time statistics
8. Create system backup
9. Manage GDPR settings

### 6.2 Doctor Workflow

1. Login with doctor credentials
2. Accept GDPR consent banner
3. View anonymized patient records
4. Access patient diagnoses
5. Export patient data for analysis
6. Review own activity

### 6.3 Receptionist Workflow

1. Login with receptionist credentials
2. Accept GDPR consent banner
3. Add new patient records
4. Enable encryption option
5. View recent additions (limited)
6. Cannot access sensitive data

## 7. Security Measures

### 7.1 Authentication Security

- Password hashing (SHA-256)
- Session management
- Login attempt logging
- Timeout handling

### 7.2 Data Security

- Encryption at rest (Fernet)
- Anonymization techniques
- Role-based data masking
- Secure key storage

### 7.3 Network Security

- Local database connection
- No external API calls
- Secure session tokens
- HTTPS ready (production)

### 7.4 Application Security

- Input validation
- SQL injection prevention (parameterized queries)
- XSS protection (Streamlit built-in)
- Error handling without info leakage

# 8. Testing & Validation

## 8.1 Functionality Testing

All user roles can login, RBAC enforces correct permissions, Anonymization works correctly, Encryption/decryption functional, Logs capture all activities, CSV exports work properly, Analytics display correctly, GDPR features operational

## 8.2 Security Testing

✅ Passwords are hashed, Sensitive data is masked, Unauthorized access prevented, SQL injection protected, Error messages don't leak data

## 8.3 GDPR Compliance Testing

Consent banner appears, Retention timer works, Expired data can be deleted, Audit trail is complete, Data minimization enforced

# 9. Conclusion

This Hospital Management System successfully implements:

1. **Complete CIA Triad**
    - Confidentiality through RBAC and encryption
    - Integrity via comprehensive logging
    - Availability with error handling and backups
2. **GDPR Compliance**
    - All seven principles implemented
    - User consent mechanism
    - Data retention management
    - Right to be forgotten
3. **Bonus Features**
    - Fernet reversible encryption

- Real-time analytics dashboard
- Complete GDPR feature set

The system demonstrates practical application of information security principles while maintaining usability and regulatory compliance.

# 10. Future Enhancements

- Multi-factor authentication (MFA)
- Advanced encryption (AES-256)
- Automated data breach detection
- Email notifications for retention
- Mobile application support
- Cloud database integration
- Advanced analytics (ML-based insights)
- Biometric authentication

**Video Link:**

https://drive.google.com/file/d/1F6oWo6t9RD_lubiaBMEhk0FnehdeTIC7/view?usp=sharing