Secure Health

A Security-Focused Medical Web App with Ethical Hacking & Full Access Control

Project for Cyber Security students (Info Mgmt Sec)

Secure Health is a comprehensive, security-oriented medical appointment and records system designed to simulate a real-world environment for exploring and defending against cyber threats. This project challenges students to develop, secure, and attack their own system, reinforcing theoretical knowledge through practical implementation.

The system will support multiple user roles (Admin, Doctor, Patient) and will enforce Role-Based Access Control (RBAC) at both the application and database levels. It will also include secure API endpoints, logging and encryption mechanisms, XSS protections, Two-Factor Authentication (2FA) using Google Authenticator, and a dedicated ethical hacking phase to test and harden the system.

User Roles and Access Control

Role	Capabilities	DB Access
Admin	Full CRUD, manage users, assign/revoke privileges, audit logs	All privileges
Doctor	CRUD on assigned patient records, view appointments	SELECT/UPDATE on assigned patients
Patient	Book appointments, view prescriptions, edit personal info	SELECT on own records only

The system must implement:

Application-level RBAC using token-based authentication

Database-level DCL (GRANT / REVOKE) to enforce fine-grained SQL access control

Admin panel to manage user privileges and DB roles securely

Required Functionalities per Role

Doctor:

- **Create**: Add diagnoses, write prescriptions
- **Read**: View own patients' records
- **Update**: Modify treatment notes
- **Delete**: Remove draft records (if allowed)

Patient:

- **Create**: Book appointments
- **Read**: View own profile and prescriptions
- **Update**: Edit contact info, change password
- **Delete**: Cancel appointments

Admin:

- Full CRUD on all data
- Grant/Revoke SQL permissions via secure web interface
- Enable/disable user accounts
- Monitor logs for suspicious activity

Security Requirements

Authentication & Authorization

- Use **JWT-based login** for all roles
- Implement **2FA with Google Authenticator** for Doctors and Admins.

Web Security

- Secure all routes with token checks and role guards
- Protect against:
    - **SQL Injection** using parameterized queries
    - **XSS** via output escaping and input sanitization
- Use **HTTPS** with OpenSSL-generated self-signed certificates

Logging & Monitoring

- Log every login attempt, data change, privilege update
- Logs must be exportable (.log or .csv)
- Admins can audit logs and download for offline analysis

Encryption

- Sensitive fields (diagnoses, notes) must be encrypted
- HTTPS enforced on all endpoints

Ethical Hacking Phase

After implementation, you will deploy a **vulnerable version** of their app to:

- Simulate real-world attacks using **Kali Linux**, **SQLMap**, **Nmap**, and manual XSS injection
- Document all successful attacks
- Apply security fixes
- Re-test and confirm defenses

Attacks to Simulate:

- SQL Injection on login or search forms (using SQLMap)
- XSS in comment/message input
- Nmap scanning of server ports
- Brute-force login attacks using simple scripts

Defenses to Implement:

- Input validation + escaping
- Proper SQL binding/prepared statements
- 2FA on login