

MedLife Hospital Project Report

1. Project Overview

The MedLife Hospital Appointment System is an advanced web application developed for efficient management of hospital appointments, patient records, doctor schedules, and healthcare communications. Utilizing ASP.NET Core Razor Pages, Entity Framework Core, and SQLite, MedLife provides secure, scalable, and user-friendly solutions tailored for patients, doctors, and administrators.

2. Functional Requirements

2.1 User Roles & Authentication

- **Patients:** Registration, appointment booking, viewing medical records, and feedback.
- **Doctors:** Appointment approvals, schedule management, medical record handling, and AI-assisted analyses.
- **Administrators:** Comprehensive management of users, appointments, and communications.

2.2 User Management

- Secure registration and robust profile management.
- Role-based authentication leveraging Microsoft Identity for stringent security.
- Admin-managed doctor registration with specialization assignment and schedule setup.

2.3 Appointment Booking & Management

- Booking appointments in 30-minute slots between 9 AM - 5 PM, excluding lunch breaks and doctor-defined off-days.
- Patients manage appointments with clear guidelines for rescheduling (48h prior) and cancellation (24h prior).
- Doctors have tools to efficiently approve or reject pending appointments.
- Administrators manage all appointments, with validations preventing overlaps and ensuring smooth scheduling.

2.4 Notifications & Reminders

- Automatic email confirmations, reminders, and notifications for appointments and changes.
- Emails include .ics files compatible with Google and Outlook calendars.
- Admins can manually send emails for debugging or communication.
- Patients control email notification preferences via their profile.

2.5 Search & Filtering

- Easy patient access to doctor information, searchable by name and specialty.
- Advanced appointment filters by date, doctor, and status, integrated seamlessly in calendar views and tables.

3. Database Design & Security

3.1 Database Schema

- **Users:** Secure storage for patient, doctor, and admin profiles.
- **Appointments:** Efficiently managed schedules with validation for booking conflicts.
- **Feedback:** Facilitates patient reviews post-appointment, enhancing service quality.
- **DoctorDaysOff:** Tracks periods when doctors are unavailable.
- **Prescriptions:** Manages medication data and statuses.
- **MedicalRecords:** Stores detailed patient medical history securely.

3.2 Security Measures

- Strict role-based access control for data protection.
- Parameterized queries with Entity Framework Core.
- Secure password management with encrypted credentials via ASP.NET Identity.

4. Project Marking Guide Evaluation

4.1 Functionality (40 Marks)

- **User Authentication & Role Management (10 Marks):** Fully integrated with Microsoft Identity.
- **Appointment Booking & Management (15 Marks):** Smooth and conflict-free booking experience.
- **Doctor Scheduling System (5 Marks):** Effective management including doctor's availability.
- **Notifications & Reminders (5 Marks):** Timely, automatic, and comprehensive.
- **Search & Filtering (10 Marks):** Intuitive and dynamic search functionalities.

4.2 Database Design & Implementation (20 Marks)

- **Schema & Relationships (10 Marks):** Robust and normalized database design ([ER Diagram](#)).
- **SQL Queries & Performance (5 Marks):** Optimized performance through EF Core.
- **Data Integrity & Validation (5 Marks):** Strong validations enforced.

4.3 Code Quality & Security (20 Marks)

- **Code Structure & Maintainability (10 Marks):** Clear and maintainable architecture.
- **Security & Validation (10 Marks):** Comprehensive security practices.

4.4 Documentation & Presentation (10 Marks)

- **Code Documentation (5 Marks):** Detailed inline documentation.
- **Presentation & Testing (5 Marks):** Robust testing and detailed project presentation.

5. User Stories Implemented

Patient Highlights

- Smooth appointment booking with clear cancellation/rescheduling policies.
- Feedback system for service improvement.
- Accessible and detailed medical record views.

Doctor Highlights

- Efficient schedule management and approval system.
- AI-assisted medical record analysis using Ollama (Qwen2.5).
- Secure, locally-run AI, ensuring privacy compliance.

6. Technical Implementation

6.1 Technologies Used

- ASP.NET Core Razor Pages
- Entity Framework Core with SQLite
- Microsoft Identity
- SMTP Email Service
- AJAX, JavaScript, and FullCalendar.js for interactive UI
- Local AI integration (Ollama – Qwen2.5)

6.2 Architecture

- MVC architecture for separation of concerns and maintainability.
- Responsive UI leveraging Bootstrap and custom CSS.

7. Security & Compliance

- Role-based access with stringent security checks.
- Data encryption and secure data management policies.

8. Performance & Scalability

- Efficient queries and scalable architecture.
- Easy expansion for future features and user growth.

9. Future Enhancements

- Telehealth consultations.
- Expanded analytical capabilities.
- Additional notification methods (e.g., SMS).

10. Challenges & Solutions

- **Appointment overlaps:** EF Core queries resolved conflicts.
- **Dynamic filtering:** Implemented using AJAX and FullCalendar.js.
- **Responsive email design:** Ensured compatibility across email clients.
- **AI integration:** Custom engineered prompts with local AI fallback.

11. Conclusion

The MedLife Hospital Appointment System significantly modernizes hospital operations, enhancing patient care, operational efficiency, and data security through innovative technology and rigorous standards.