Hospital Management System V1 Project Report

Student Name: Utkarsh Shukla ([21f2001497@ds.study.iitm.ac.in])
Student ID: 21f2001497
Course: MAD 1
Institution: IIT MADRAS
Submission Date: November 30, 2024
Project Title: Hospital Management System Version 1

Project Approach

1. **Modular Flask Architecture:** Blueprint-based design for scalable development and maintenance
2. **Role-Based Access Control:** Distinct interfaces and permissions for Admins, Doctors, and Patients
3. **Comprehensive Healthcare Workflow:** End-to-end appointment booking, treatment management, and medical records
4. **Professional UI/UX Design:** Healthcare-themed interface with beige/red color scheme and responsive design
5. **Security-First Development:** Multi-layer authentication, CSRF protection, and data validation

Key Features Implemented

User Management:

- Multi-role authentication system (Admin, Doctor, Patient)
- Flask-Login session management with secure password hashing
- Account activation/deactivation with soft delete functionality
- Profile management for all user types with medical history tracking

Appointment System:

- Doctor availability management with weekly scheduling
- Smart appointment booking with real-time conflict prevention
- 7-day availability calendar with time slot management
- Appointment status tracking (booked, completed, cancelled, rescheduled)
- Database constraints preventing double-booking

Treatment Management:

- Comprehensive treatment records (diagnosis, prescription, clinical notes)
- CRUD operations for treatment data with doctor authorization
- Patient medical history timeline with complete treatment access
- Print-friendly treatment summaries and medical records

Dashboard Features:

- **Admin Dashboard:** System statistics, user management, advanced search functionality
- **Doctor Dashboard:** Appointment management, patient records, availability settings, treatment tracking
- **Patient Dashboard:** Doctor browsing by department, appointment history, medical records access

AI/LLM Declaration

I utilized **Gemini 2.5 Pro** as an AI assistant for the following specific purposes:

CSS Styling and UI Design:

- Received suggestions for Bootstrap 5 component styling and responsive design principles
- Obtained guidance on professional healthcare theming with beige/red color scheme
- Got help with print-friendly CSS optimization for medical records
- Received recommendations for card-based layouts and navigation design

Learning and Understanding:

- Used AI to better understand Flask application factory pattern
- Gained insights into Jinja2 templating and template inheritance
- Learned about custom decorator implementation for role-based authorization
- Understood healthcare workflow requirements and medical record management

Important Note: Including documentation also, my usage pretty clearly falls under <10% mark.

Frameworks and Libraries Used

Backend Framework:

- **Flask 3.0.0** - Python web framework with Blueprint architecture for modular development
- **Flask-SQLAlchemy 3.1.1** - ORM for database operations and model relationships
- **Flask-Login 0.6.3** - User session management and authentication handling
- **Flask-WTF 1.2.1** - Form handling with CSRF protection and validation
- **WTForms 3.1.1** - Form validation, rendering, and custom validator implementation

Database and Security:

- **SQLite** - Lightweight relational database for development and deployment
- **Werkzeug 3.0.1** - WSGI utility library with password hashing capabilities
- **email-validator 2.1.0** - Email format validation and verification

Frontend Technologies:

- **Bootstrap 5.3.0** - Responsive CSS framework for modern web design
- **Bootstrap Icons** - Consistent iconography throughout the application

- **Custom CSS** - Professional healthcare theming with beige/red color scheme
- **Jinja2** - Template engine for dynamic content rendering and template inheritance

Development Tools:

- **Python 3.8+** - Backend runtime environment
- **pip** - Package manager for Python dependencies
- **SQLite Browser** - Database management and inspection tool

Database Tables Description

1. USERS Table

- Primary authentication table for all system users
- Stores email, hashed passwords, and role information (admin, doctor, patient)
- Includes account status (is_active) and creation timestamp
- Supports role-based access control with cascade relationships

2. DEPARTMENTS Table

- Medical departments/specializations catalog
- Referenced by doctors for organizational structure
- Includes department names and descriptions for patient browsing

3. DOCTORS Table

- Doctor profile information linked to user accounts
- Stores professional details like specialization and years of experience
- Foreign key relationship with departments and users
- Includes contact information and professional credentials

4. PATIENTS Table

- Patient profile information and comprehensive medical history
- Demographic information including date of birth, address, blood group
- Medical history field for chronic conditions and allergies
- Linked to user accounts for authentication and profile management

5. APPOINTMENTS Table

- Central appointment scheduling table with conflict prevention
- Links patients and doctors with specific date/time slots
- Tracks appointment status throughout lifecycle (booked, completed, cancelled, rescheduled)
- Includes unique database index on doctor_id, date, time to prevent double-booking

6. TREATMENTS Table

- Medical records for completed appointments
- Stores comprehensive diagnosis, prescriptions, and clinical notes
- One-to-one relationship with appointments for detailed medical documentation
- Supports complete patient medical history tracking

7. AVAILABILITY Table

- Doctor availability schedule management system
- Weekly recurring schedule with day-of-week mapping (0=Monday, 6=Sunday)
- Supports multiple time slots per day per doctor
- Includes availability status for temporary schedule changes

Application Routes and Endpoints

Authentication Endpoints

```
GET/POST  /login                   - User authentication with role-based redirection
GET/POST  /register                - Patient registration with medical information
GET       /logout                  - Secure user logout with session cleanup
GET       /                        - Landing page with system overview
```

Admin Endpoints (Protected: @admin_required)

```
GET       /admin/dashboard         - System statistics and overview
GET       /admin/doctors           - List all doctors with department filtering
GET/POST  /admin/doctors/add       - Add new doctor with department assignment
GET/POST  /admin/doctors/<id>/edit - Edit doctor profile and credentials
POST      /admin/doctors/<id>/delete - Permanent doctor account deletion
POST      /admin/doctors/<id>/deactivate - Soft delete (deactivate) doctor account
GET       /admin/patients          - List all patients with search functionality
GET/POST  /admin/patients/<id>/edit - Edit patient profile and medical information
POST      /admin/patients/<id>/delete - Permanent patient account deletion
GET/POST  /admin/search            - Advanced search across doctors and patients
```

Doctor Endpoints (Protected: @doctor_required)

```
GET       /doctor/dashboard        - Doctor dashboard with appointment statistics
GET       /doctor/appointments     - List all doctor's appointments with filtering
POST      /doctor/appointments/<id>/complete - Mark appointment as completed
POST      /doctor/appointments/<id>/cancel   - Cancel appointment with reason
GET/POST  /doctor/appointments/<id>/treatment - Add comprehensive treatment record
GET       /doctor/treatments/<id>/view       - View detailed treatment information
GET/POST  /doctor/treatments/<id>/edit       - Edit existing treatment records
GET       /doctor/availability               - View current availability schedule
GET/POST  /doctor/availability/add           - Add new availability time slots
POST      /doctor/availability/<id>/delete   - Remove availability slot
```

Patient Endpoints (Protected: @patient_required)

```
GET       /patient/dashboard        - Patient dashboard with appointment overview
GET       /patient/doctors          - Browse doctors by department with availability
GET       /patient/doctors/<id>     - View doctor profile and 7-day availability
GET/POST  /patient/appointments/book/<doctor_id> - Book appointment with conflict
checking
GET       /patient/appointments     - View upcoming and past appointments
POST      /patient/appointments/<id>/cancel      - Cancel appointment
GET       /patient/treatments/<id>/view          - View treatment details
GET       /patient/medical-history               - Complete medical history timeline
GET/POST  /patient/profile                      - View/edit patient profile and medical
info
```

Presentation Video

Drive Link: [https://drive.google.com/drive/folders/1KdX10084BEzPa7XtyY0OsIXm5-Q9EjGm?usp=drive_link]

Conclusion

Key Achievements:

- **Complete Healthcare Workflow**: End-to-end appointment booking, treatment management, and medical record keeping
- **Advanced Security Implementation**: Multi-layer authentication, CSRF protection, and role-based authorization
- **Intelligent Conflict Prevention**: Database-level constraints and real-time validation for appointment scheduling
- **Professional Healthcare Interface**: Responsive design with healthcare-specific theming and print optimization
- **Modular Architecture**: Scalable Flask Blueprint structure supporting future enhancements and maintenance