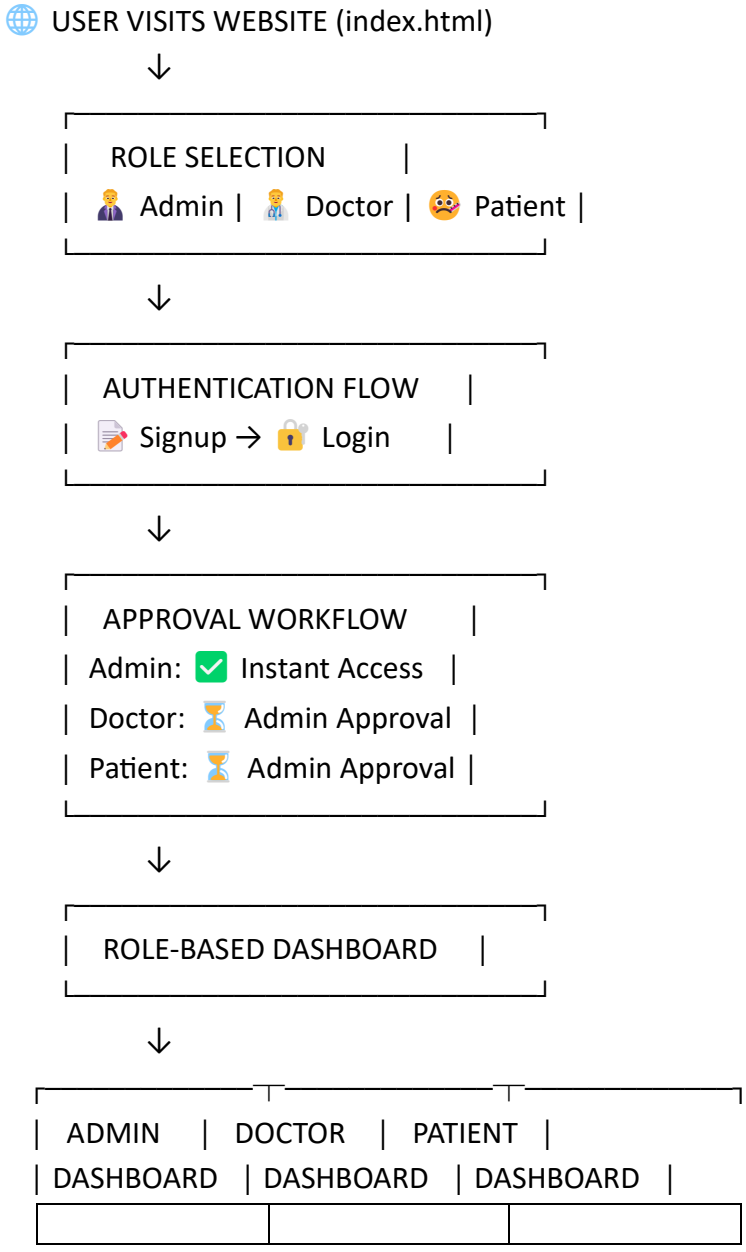
KL-Hospital

COMPLETE HOSPITAL MANAGEMENT SYSTEM FLOW CHART & DETAILED

SYSTEM ARCHITECTURE FLOW CHART



🔧 TECHNOLOGY STACK & TOOLS

Backend Framework

- **Django 3.2.23** - Python web framework
  - **Purpose**: MVC architecture, ORM, authentication
  - **Importance**: Rapid development, security, scalability

Database System

- **SQLite3** - File-based database
  - **Location**: db.sqlite3 file in project root
  - **Purpose**: Store all application data
  - **Tables**: 19 migration files show database evolution

Frontend Technologies

- **HTML5** - 67 template files
- **CSS3** - Custom styling (static/style.css)
- **Bootstrap** - Responsive design framework
- **JavaScript** - Interactive elements

Additional Libraries

- **django-widget-tweaks** - Form styling
- **Pillow** - Image processing
- **sqlparse** - SQL parsing

Deployment Tools

- **Docker** - Containerization
- **Docker Compose** - Multi-container orchestration
- **Git** - Version control

💾 DATA STORAGE ARCHITECTURE

Database Location

📁 Project Root
├── 📄 db.sqlite3 (Main Database File)
├── 📁 static/
│   ├── 📁 profile_pic/
│   │   ├── 📁 DoctorProfilePic/
│   │   └── 📁 PatientProfilePic/
│   └── 📁 appointment_images/
└── 📁 media/ (Runtime uploads)

Database Tables Structure

1. User Management Tables

- **auth_user** - Django's built-in user table
  - Fields: username, password, email, first_name, last_name
- **auth_group** - User roles (ADMIN, DOCTOR, PATIENT)
- **auth_user_groups** - User-role relationships

2. Hospital-Specific Tables

- **hospital_doctor**
  - user_id, profile_pic, address, mobile, department, status
- **hospital_patient**
  - user_id, profile_pic, address, mobile, symptoms, assignedDoctorId, admitDate, status
- **hospital_appointment**
  - patientId, doctorId, appointmentDate, appointmentTime, description, image, status
- **hospital_patientdischargedetails**
  - patientId, admitDate, releaseDate, roomCharge, medicineCost, doctorFee, total

🔄 **COMPLETE SYSTEM WORKFLOW**

Phase 1: User Registration & Authentication

🏠 Homepage (index.html)
↓
👤 User Selects Role
↓
📝 Registration Form
├── Admin: Instant account creation
├── Doctor: Account created, status=False
└── Patient: Account created, status=False
↓
🔐 Login Attempt
↓
🔍 afterlogin_view() checks:
├── is_admin() → Admin Dashboard
├── is_doctor() + status=True → Doctor Dashboard
├── is_doctor() + status=False → Wait for Approval
├── is_patient() + status=True → Patient Dashboard
└── is_patient() + status=False → Wait for Approval

Phase 2: Admin Workflow

👨‍💼 ADMIN DASHBOARD

- 📊 Statistics Display
  - Total Doctors (approved/pending)
  - Total Patients (approved/pending)
  - Total Appointments (confirmed/pending)
- 👨‍⚕️ DOCTOR MANAGEMENT
  - View All Doctors
  - Add New Doctor (status=True)
  - Approve Pending Doctors
  - Update Doctor Info
  - Delete Doctor
- 🤒 PATIENT MANAGEMENT
  - View All Patients
  - Add New Patient
  - Approve Pending Patients
  - Assign Doctor to Patient
  - Update Patient Info
  - Discharge Patient
- 🗓️ APPOINTMENT MANAGEMENT
  - View All Appointments
  - Create Appointment
  - Approve Patient Requests
  - Reject Appointments
- 💰 BILLING SYSTEM
  - Generate Discharge Bill
  - Calculate: Room + Medicine + Doctor + Other
  - Create PDF Invoice

Phase 3: Doctor Workflow

- 👨‍⚕️ DOCTOR DASHBOARD
  - 📊 Statistics
    - Assigned Patients Count
    - Appointments Count
    - Discharged Patients Count
  - 🥺 PATIENT MANAGEMENT
    - View Assigned Patients
    - Search Patients (by name/symptoms)
    - Update Patient Status
    - View Discharged Patients
  - 🗓️ APPOINTMENT MANAGEMENT
    - View My Appointments
    - Delete Completed Appointments
    - Manage Schedule

Phase 4: Patient Workflow

- 🥺 PATIENT DASHBOARD
  - 👨‍⚕️ DOCTOR INFO
    - Assigned Doctor Details
    - Doctor Specialization
    - Contact Information
  - 🗓️ APPOINTMENT SYSTEM
    - Book New Appointment
    - Select Doctor
    - Choose Date/Time
    - Upload Medical Images
    - View Appointment Status
    - Search Available Doctors
  - 💰 BILLING
    - View Discharge Status
    - Download Invoice PDF
    - Payment Details

🗂️ FILE STRUCTURE & COMPONENTS

Core Django Files

- **manage.py** - Django management commands
- **settings.py** - Configuration (database, static files, email)
- **urls.py** - URL routing (80+ routes)
- **models.py** - Database models (4 main models)
- **views.py** - Business logic (50+ view functions)
- **forms.py** - Form handling (8 form classes)
- **admin.py** - Django admin interface

Template System (67 HTML Files)

📁 templates/hospital/
├── 🏠 Homepage: index.html, homebase.html
├── 🔐 Authentication: *login.html, *signup.html
├── 👨‍💼 Admin: admin_*.html (20 files)
├── 👨‍⚕️ Doctor: doctor_*.html (12 files)
├── 🤒 Patient: patient_*.html (15 files)
├── 📄 Utility: footer.html, navbar.html
└── 📞 Contact: contactus.html, aboutus.html

Static Assets

📁 static/
├── 🎨 style.css - Custom styling
├── 🖼️ images/ - UI icons (admin, doctor, patient)
├── 📸 profile_pic/ - User profile pictures
└── 📱 screenshots/ - Documentation images

🔒 SECURITY & AUTHENTICATION

Role-Based Access Control

- **Django Groups**: ADMIN, DOCTOR, PATIENT
- **Decorators**: @login_required, @user_passes_test
- **Status Approval**: Admin must approve doctors/patients

Data Validation

- **Form Validation**: Django forms with field validation
- **CSRF Protection**: Built-in Django middleware
- **Password Hashing**: Django's built-in system

📊 DATA FLOW PROCESS

User Registration Flow

1. User fills signup form
2. User object created in auth_user table
3. Profile object created (Doctor/Patient table)
4. User added to appropriate group
5. Status set to False (except Admin)
6. Admin approval required for access

Appointment Booking Flow

1. Patient selects doctor from dropdown
2. Patient chooses date/time
3. Appointment object created with status=False
4. Admin receives approval request
5. Admin approves/rejects appointment
6. Status updated to True/deleted
7. Both doctor and patient can view

Billing Process Flow

1. Admin initiates patient discharge
2. System calculates days spent
3. Admin enters: room charge, medicine cost, doctor fee
4. Total = (room_charge × days) + medicine + doctor + other
5. PatientDischargeDetails record created
6. PDF invoice generated
7. Patient can download bill

🚀 DEPLOYMENT ARCHITECTURE

Docker Configuration

- **Dockerfile**: Python 3.11, Django setup
- **docker-compose.yml**: Service orchestration
- **requirements.txt**: Dependency management

Production Settings

- **Static Files**: Collected for production
- **Media Files**: User uploads handling
- **Database**: SQLite for development, PostgreSQL for production

Links:-

1. **GITHUB:-** https://github.com/2300033173/KLU-Hospital

2. **Bashboard Render:-** https://dashboard.render.com/web/srv-d3do836r433s73efkdlg/deploys/dep-d3do83er433s73efkdsg?r=2025-09-30%4007%3A16%3A34%7E2025-09-30%4007%3A21%3A38

3. **PROJECT LIVE LINK:-** https://klu-hospital.onrender.com/

https://neonstatus.com/azure-east-us-2-virginia-eastus-2