PROJECT: JAVA AND WEB APPLICATION DEVELOPMENT COURSE CODE: DLBCSPJWD01

Development/Reflection Phase

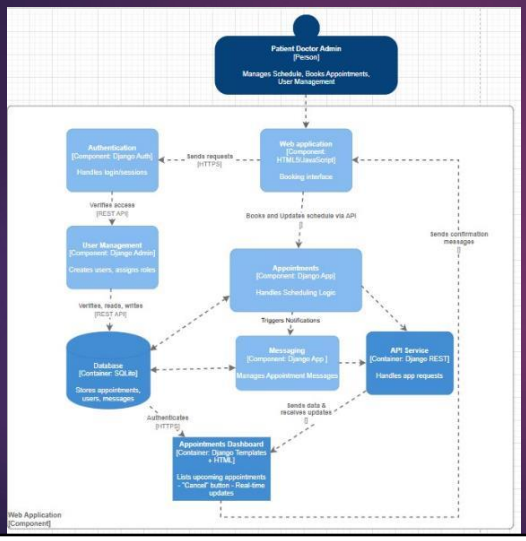
- GITHUB LINK: HTTPS://GITHUB.COM/GABBYDEPSALMIST/MEDICONNECT

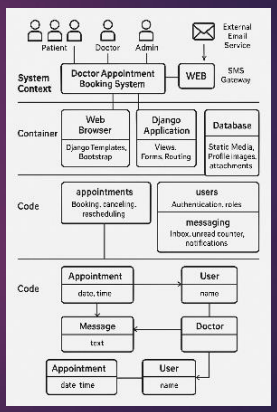
- NAME: GABRIEL AGYEMANG

Purpose of the Web App

- **Purpose:**
  This web application streamlines the process of booking medical consultations by enabling patients to easily schedule, manage, and confirm doctor appointments online. It enhances communication between patients and healthcare providers through real-time notifications and an integrated messaging system.
- **Key Features:**
  - Browse doctors and their working hours (Mon–Fri, 8:00 AM–6:00 PM)
  - Book 2-hour appointment slots with flexibility to cancel or reschedule
  - Simulated payment system marking appointments as "Paid" upon transaction completion
  - In-app notifications for confirmations, reminders (24h & 1h before), and cancellations
  - Patients confirm or cancel appointments directly through their inbox
  - Doctors receive real-time updates on appointment changes and slot availability
- **Target Users:**
  - Patients seeking convenient self-service scheduling
  - Clinics or independent practitioners aiming to reduce manual appointment handling

Architecture Diagram





Technical Approach:

- **Frontend:** HTML, CSS, JavaScript (Browser-based UI)
- **Backend:** Django REST API for business logic and data processing
- **Database:** SQLite for persistent appointment and user data

Overview of Architecture

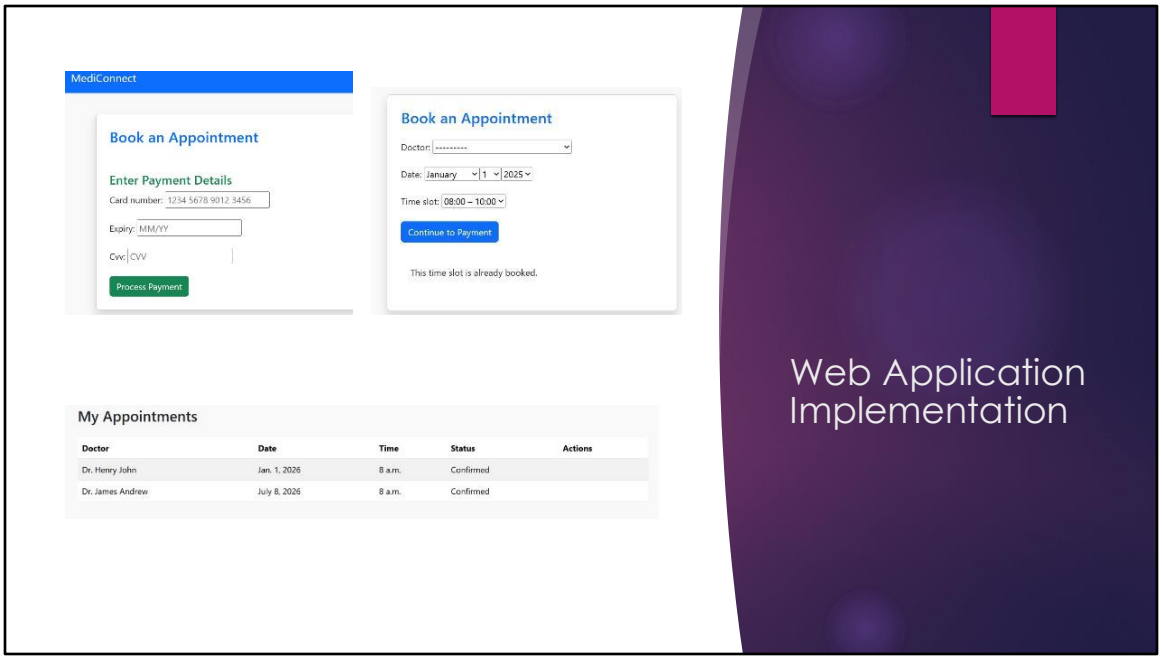
- **Frontend:**
  - **HTML, CSS, JavaScript** — Provides a responsive, user-friendly interface for patients and doctors to interact with the system via their web browsers.
- **Backend:**
  - **Django REST Framework** — Handles API requests, business logic, authentication, and appointment management in a scalable and secure manner.
- **Database:**
  - **SQLite** — Lightweight, file-based database used for storing user data, appointment schedules, and messaging information during development and early phases.

Web Application Implementation

New users can create accounts and old users can log in to their accounts. The screenshots show MediConnect, a doctor appointment web app. The login page (left) has fields for username/password and a registration link. The signup page (right) includes validation rules for usernames (150 chars max), mandatory names, email, and secure passwords (e.g., no common phrases). Both pages use a clean, user-friendly design for easy access.

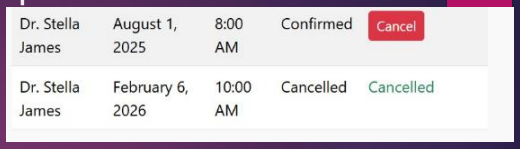
Web Application Implementation

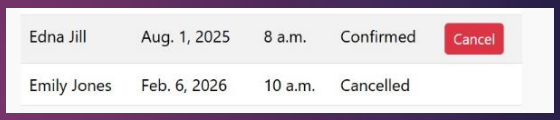
This is the landing page it allows the users who are doctors, admins and patients to register, login, check their appointments, book appointment, messages, cancel appointments and logout using the navigation bar options or the my appointments section below the login. The screenshots show MediConnect, a doctor appointment management system. The interface displays a "My Appointments" section listing patient names, dates, times, and statuses (e.g., "Confirmed"). Each appointment includes a "Cancel" button for easy management. The clean layout ensures quick navigation for doctors to track and modify schedules securely.

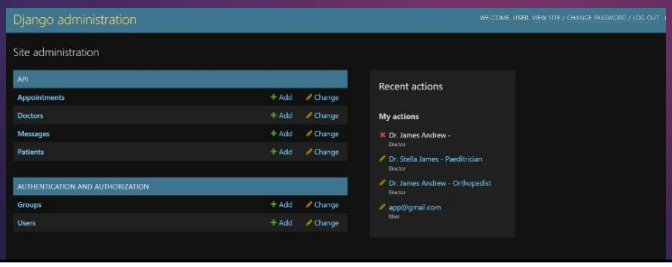


This is the book an appointment page and it allows the patients to book an appointment and pay for it so it can be registered as booked. After booking an appointment it registers in real-time for all the users, marks slot as booked and each one of them doctors or patients can cancel the appointment. The screenshots show Book an Appointment and Payment interfaces in a medical booking system. Users select doctors, dates, and time slots (with booked slots flagged). The My Appointments section displays scheduled visits with statuses. The payment page collects card details securely. The design prioritizes clear scheduling and transaction processing.

Web Application Implementation





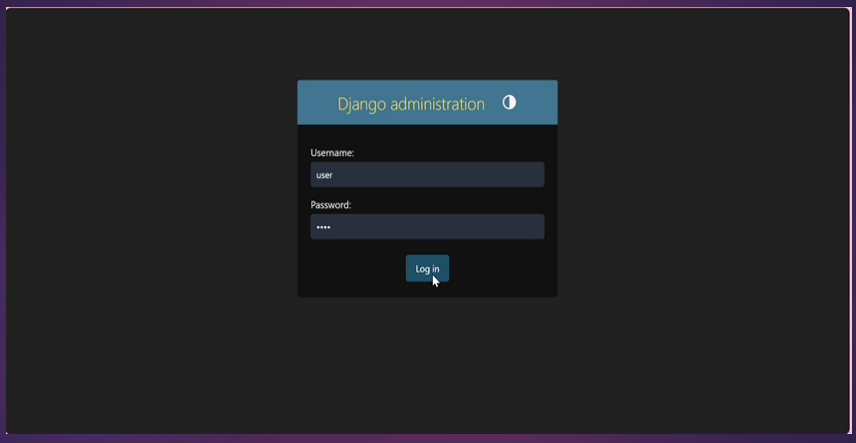


Messages indicates the messages and new appointments it updates in real time. It shows all appointments and cancellations and updates for Doctors and Patients. The admin panel can be used to manage the users and appointments from the backend. The screenshots captures multiple interfaces of a doctor appointment system. The Inbox shows new appointment notifications, while the appointment management view displays confirmed and cancelled bookings with action buttons. The Django admin panel appears below, revealing backend controls for site administration. The interface combines patient communication with administrative functionality in a unified system.

Chart on Changes

Aspect	Original Proposal	Change Made	Reason / Benefit
Database	PostgreSQL	Switched to Django's default SQLite	Simpler setup and easier local development/testing during Phase 1
Messaging & Notifications	In-app messages for all appointment updates	Added appointment status summary on login, alongside messages	Quick appointment overview upon login; retained messaging for detailed communication
Appointment Management	Manage appointments via messages	Added direct cancellation option with status summary	Simplifies appointment management for patients and doctors, improves user experience
System Management	No dedicated admin interface	Added admin panel accessible from landing page	Easier management of doctors, appointments, and users in the backend

Screencast Video



Test Action and Expected Result

Login as Admin - Session is created, redirected to dashboard

Create New Appointment - Appointment is stored in DB, confirmation message sent

Click "Cancel" on appointment - Appointment is deleted from DB, UI updates in real-time

Patient books via Web UI - Appointment saved, user receives confirmation message

View Appointment Dashboard - Upcoming appointments are listed with “Cancel” option