



P R E S E N T A T I O N

book appointments with doctors

...

iDoc is an online platform that lets you schedule doctor appointments from home and track your medications, test results, and diagnoses, all in one place





TEAM

...



Fatma Soliman

fatmacivil@hotmail.com



Menna Samir

menna20.samir@gmail.com



ADHAM ELSAYED

Prg.adhamelsayed@gmail.com





Story of our project

• • •

The idea for IDOC originated from our experiences with the inefficiencies of scheduling healthcare appointments, such as long waits, difficulty finding slots, and miscommunication. We created IDOC to streamline the appointment process, improve communication, and enhance the overall experience for patients and healthcare providers.



Technologies



- **Languages:** Python
- **Frameworks:** Flask, SQLAlchemy
- **Databases:** MySQL
- **Frontend Technologies:** HTML5, CSS3, JavaScript (ECMAScript 6), Bootstrap 5, jQuery, Font Awesome
- **Charts and Visualization:** ApexCharts.js
- **Scheduler and Calendar:** Scheduler Calendar (specific library/plugin to be determined)
- **Form Validation:** jQuery Validation
- **Deployment Platforms:** Linux (Virtual or Cloud-based)
- **Version Control:** Git (for source code management)
- **Cloud Services:** Google Cloud or Azure (for hosting and database services)

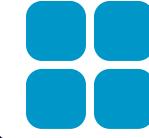


Clinic Workflow on the Website

● ● ● cycle clinic



- The user logs into their personal account on the website.
- The system checks the user's permissions (such as clinic, doctor, or admin) to grant access to the appropriate interface.



- Once logged in as a clinic, the user is directed to the Clinic Dashboard.
- The dashboard displays the number of appointments scheduled for the current day as well as the total number of appointments.
- The clinic can view a list of their appointments with details about the patients and doctors involved.



- The clinic can view details of all upcoming and past appointments.
- Appointments are categorized based on their status (such as seen or unseen).
- The dashboard also provides the ability to create, edit, and delete appointments.



- The Calendar Page gives the user a comprehensive view of the clinic's appointments.
- The calendar is automatically updated with the latest data from the database.



- The clinic is notified of any new updates regarding appointments or other system alerts.
- The clinic can view all notifications on a dedicated page and mark notifications as read or delete them.



- The user can mark any notification as read or delete it from the system.
- The notification status is updated in the database to ensure accurate tracking.



Doctor Workflow on the Website

● ● ● cycle Doctor



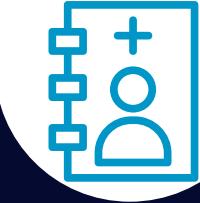
- The doctor logs into their personal account on the website.
- The system verifies the user's permissions to grant access to the doctor's interface



- After logging in, the doctor is directed to the Doctor Dashboard.
- The dashboard displays a list of the doctor's appointments for the day, sorted by time.
- The doctor can see the total number of patients scheduled for the day and the total number of appointments for the current month.
- The next upcoming appointment is highlighted for quick access.



- The doctor can mark appointments as "seen" after they have been completed.
- The dashboard is updated to reflect changes, such as marking an appointment as completed.



- The doctor can access a list of all patients who have appointments with them.
- This list provides quick access to patient information and history.



- The doctor can view and update their personal and professional details through the Doctor Profile page.
- This includes updating specializations, consultation duration, and uploading a profile photo.
- The doctor can make changes to their profile, which are saved and reflected immediately.

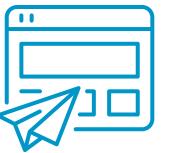


- The doctor can create and manage prescriptions for patients through the Prescription Management page.
- They can add multiple medicines, specify quantities, duration, and the times of day for each medicine.
- Once the prescription is submitted, it is saved in the patient's medical record, and the doctor receives a confirmation.

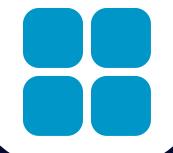


Patient Workflow on the Website

● ● ● cycle Patient



The `home()` function handles the logic for the home page, which includes a search form for doctors. It populates dropdown choices for specialization and governorate from the database and handles form submissions to redirect users to a doctor search page.



- The `patient_dashboard()` function fetches the current patient's details, upcoming appointments, blood group, allergies, and a limited number of prescriptions for display on the patient's dashboard.
- It also handles various conditions such as if the patient has an upcoming appointment or not and prepares data to be rendered in the `patient-dashboard.html` template.



- The doctor can mark appointments as "seen" after they have been completed.
- The dashboard is updated to reflect changes, such as marking an appointment as completed.



The `cancel_appointment()` function allows a patient to cancel an appointment. It checks if the appointment exists, verifies that the current user is the patient who made the appointment, and then updates the appointment status to 'Cancelled'.



The `search_doctor()` function handles the search functionality for doctors. It allows users to filter doctors based on specialization, governorate, and doctor name. It also paginates the results and passes the relevant data to the `search.html` template.



The `doctor_appointments()` function generates available timeslots for a doctor based on their working hours and appointment duration. It handles the booking form submission and redirects to the checkout process.

Discussion of process

...

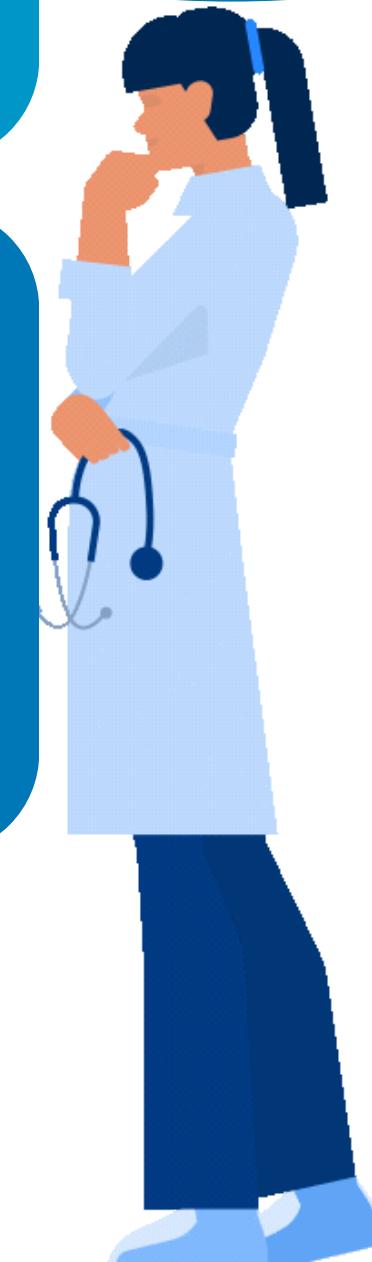
The development process for the IDOC Doctor Appointment Booking System is meticulously structured to ensure a high-quality, user-centric solution. It begins with requirements gathering to understand the needs of all stakeholders, followed by detailed design planning to create a robust system architecture and user interface. The implementation phase involves building the backend and frontend, ensuring seamless integration and functionality.

Requirements Gathering

Design

Implementation

Testing



Timeline

1

Day 1-2: Planning and Environment Setup

- Set up the development environment (Flask, database, etc.).
- Review and finalize the project requirements and specifications.

Day 3-4: Setting Up Models and Basic Structure

- Design the database schema.
- Create basic models (Users, Patients, Doctors, Appointments, etc.).
- Set up the basic project structure using Flask.

Day 5-7: Front-End Development

- Design and implement the basic user interface (Home, Patient Dashboard, Doctor Search).
- Integrate forms into the UI (SearchForm, AppointmentForm, etc.).

2

Day 8-9: Back-End Development

- Develop necessary APIs (doctor search, appointment booking, user management).
- Integrate data models with the database and develop CRUD operations.

Day 10-11: Testing and Debugging

- Conduct unit testing and integration testing.
- Fix any bugs identified during testing.

Day 12-13: Adding Advanced Features and Final Touches

- Implement advanced features like real-time notifications, email sending.
- Refine the user interface and enhance user experience.

Day 14: Final Review and Delivery

- Perform a final review of functionalities and ensure they meet the requirements.
- Document the project and prepare it for delivery.



Challenge

API Integration Complexity

Integrating various APIs for different user roles posed communication and data consistency challenges.

Solution

Allocated additional time for testing, refined error handling mechanisms, and debugged extensively to ensure smooth communication.

Real-Time Data Updates

Maintaining real-time updates for doctor availability and appointments led to delays and data mismatches.

Solution

Implemented WebSockets for instant communication, ensuring immediate updates without inconsistencies.

Regulatory Compliance

Navigating diverse regional regulations for data protection and health records posed complexities.

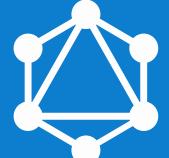
Solution

Consulted legal experts, implemented additional security measures, and adapted protocols to ensure compliance.

Learnings

• • •

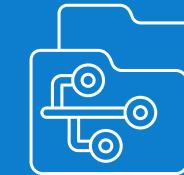
about technical interests as a result of this project



Framework and
Language Choice



Database Selection



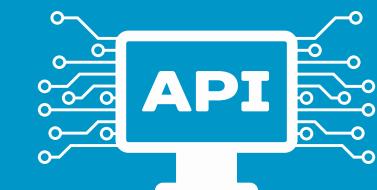
Version Control and
Collaboration



Deployment Strategy



Testing Practices



API Design



QUESTIONS



Thank You.

For Your Attention

