VitalTrack

Muhammad Otah Bashi

1. Introduction

An Electronic Health Record (EHR) system is often described as a digital version of the paper chart, but the practical aim is broader than digitising documents. An EHR is meant to organise patient information so it can be updated over time and accessed securely by authorised users when clinical decisions are being made. In a typical setting, this includes medical history, diagnoses, medications, treatment plans, immunisation dates, allergies, radiology images, and laboratory results—information that would otherwise be scattered across paper files or separate systems.

EHRs have become a standard part of modern healthcare because they can improve the reliability and availability of patient information. When clinicians can find up-to-date records quickly, they are better positioned to make informed decisions, coordinate care with other providers, and reduce avoidable errors that come from missing details or illegible handwriting. At the same time, real-world EHRs also show that benefits depend on design quality and how well the system supports everyday workflows. For this reason, building even a simplified EHR system requires attention not only to features, but also to usability and access control.

The development of EHRs has a long history, with early hospital computing projects appearing decades ago. However, widespread adoption became more visible in the 2000s, supported by improvements in web technologies, databases, and policy environments that encouraged digitisation. Today, EHR systems are used across most healthcare settings, and they provide a useful reference point for understanding how sensitive data is stored, structured, and accessed in a controlled way.

I developed this EHR information system as a university project to practise building a web-based healthcare application that resembles core EHR functionality. Before implementation, I reviewed examples of existing EHR systems and feature lists to understand what is commonly expected in a basic clinical record platform. That initial research helped shape the scope: doctor registration and login, patient management, appointment scheduling, recording laboratory results, uploading radiology images, and documenting medical history (especially allergies). The system is designed mainly for doctors to manage their own patients' records within a single application. Building it gave me practical experience with full-stack development, database modelling, and security decisions that are unavoidable when handling health-related information.

2. Website Design and Technical Architecture

2.1 Design Process

Before coding, I planned the system structure and user interface with the assumption that a doctor should be able to navigate quickly and predictably. Rather than designing a complex interface with deep menus, I used a simple layout with a top navigation bar that provides direct access to the main modules. This approach keeps the system understandable as the number of records increases and reduces the effort required to switch between tasks.

The user interface follows a clean, modern style. A purple gradient navigation bar includes the “VT” (VitalTrack) logo and links to key areas: Dashboard, Patients, Appointments, Lab Results, Radiology Imaging, and About Us. The navigation bar also shows the logged-in doctor’s name and specialty, with a dropdown menu for profile settings and logout. Even in a student project, this small detail reinforces that the system is account-based and that access should be controlled by login state.

2.2 Frontend Structure

The frontend is built using HTML, CSS, and Bootstrap. Pages extend from a shared base template (base.html) so that navigation and styling remain consistent throughout the system. This reduces duplicated code and makes it easier to update the interface without changing every page separately.

Technologies used include: HTML5 for structure, CSS3 for custom styling, Bootstrap 4.6.2 for responsive design, and icon libraries (Bootstrap Icons and Font Awesome) for visual cues across modules. Google Fonts (Kavoon and Mogra) provide consistent typography. Custom CSS refines elements such as the navigation bar, cards, buttons, forms, and flash messages. The colour scheme centres on purple gradients for primary elements, chosen to give the system a clear identity while keeping the overall interface professional.

2.3 Backend Structure

The application uses Flask’s application factory pattern, creating the app in app.py through a create_app() function. Routes are organised into blueprint modules, which keeps features separated and improves maintainability. The main blueprints include:

- auth.py — Registration, login, email confirmation, password reset
- main.py — Dashboard and general navigation
- patients.py — Patient management
- appointments.py — Appointment scheduling and editing

- lab_results.py — Laboratory results management
- radiology.py — Radiology imaging management
- medical_history.py — Medical history and allergy tracking
- doctor.py — Doctor profile management

Early in development, routes were placed in a single file, but splitting them into blueprints became necessary as the project grew. This change made the code easier to read and reduced the likelihood of feature interactions being hidden inside a large route file.

Technologies Used:

- Python 3 — Programming language
- Flask 2.3.3 — Web framework
- Flask-SQLAlchemy 3.0.5 — Database ORM (Object-Relational Mapping)
- Flask-Mail 0.9.1— Email functionality
- Werkzeug 2.3.7 — Password hashing and security utilities
- itsdangerous 2.1.2 — Token generation for email confirmation and password reset
- python-dotenv 1.0.0 — Environment variable management
- mysql-connector-python 8.1.0 — MySQL database connector
- Pillow 12.0.0 — Image processing for radiology uploads

2.4 Database Structure

The system uses MySQL, with the schema defined through SQLAlchemy models in models.py. The database is organised into core tables and clinical record tables that reflect the main entities and relationships needed for basic EHR functionality.

Core tables include:

- **Doctor** — Doctor identity and account data (including hashed password and specialty)
- **Specialty** — Medical specialty information
- **Patient** — Patient identifiers and basic details (name, email, phone, age, gender, date of birth)
- **DemographicInfo** — Additional patient details (address, emergency contact)
- **SocialHistory** — Patient social history (smoking, alcohol use, drug use, occupation)

Medical record tables include:

- **MedicalHistory** — Patient history entries linking to allergies/conditions with description and date
- **Allergy** — Allergy/condition names for reuse and consistency
- **LaboratoryResult** — Test name, result, unit, reference range, status, date, and notes
- **RadiologyImaging** — Imaging name/type, date, and uploaded image filename
- **Appointment** — Patient, doctor, date/time, type, status, and notes
- **Prescription** — Medication details and date ranges (included in the model for completeness)

Relationships are designed so that a doctor can have many patients, and each patient can have many appointments, lab results, radiology records, and medical history entries. The model also includes a many-to-many relationship between doctors and specialties, although the current user interface supports only one specialty per doctor. This mismatch is noted later as a limitation and potential improvement.

2.5 Data Flow

The system follows a typical web application flow. The user interacts with HTML pages through links and forms, generating HTTP requests that Flask routes handle. Protected routes use a @login_required decorator, which checks session data for a logged-in flag. When the user is authenticated, the route handler performs database operations through SQLAlchemy, validates inputs, and then passes results into Jinja2 templates to render the final HTML response.

A common example is adding a patient: a doctor submits a form via POST, the system validates required fields and basic formatting, creates a new Patient object linked to the logged-in doctor, commits the entry to the database, and then shows a success message before redirecting to the patient list. Similar patterns apply across appointments, lab results, radiology records, and medical history entries.

2.6 Security Features

Because the system deals with sensitive health information, basic security practices were built into the design:

- **Password hashing:** Passwords are hashed using Werkzeug rather than stored in plaintext. This was implemented after early development, when it became clear that plaintext storage is not acceptable even for a learning project.
- **Session management:** Authentication state is stored in Flask sessions, including the doctor's ID and basic profile fields.
- **Email confirmation:** New registrations require email confirmation before login is allowed, reducing the risk of fake or unreachable accounts.
- **Password reset:** A "forgot password" flow sends a time-limited token link via email.
- **Access control:** Doctors can only access records linked to their own account. Queries are filtered by the logged-in doctor's ID.
- **File upload checks:** Radiology uploads validate file type and size, and files are stored in patient-specific folders.
- **Environment variables:** Sensitive configuration (database credentials, secret keys, email settings) is stored in a .env file rather than hard-coded. The .env file is not committed, and a .env.example is provided.

These measures do not cover every requirement of a production healthcare system, but they establish a responsible baseline for a university project and align with common secure development expectations for applications handling private data.

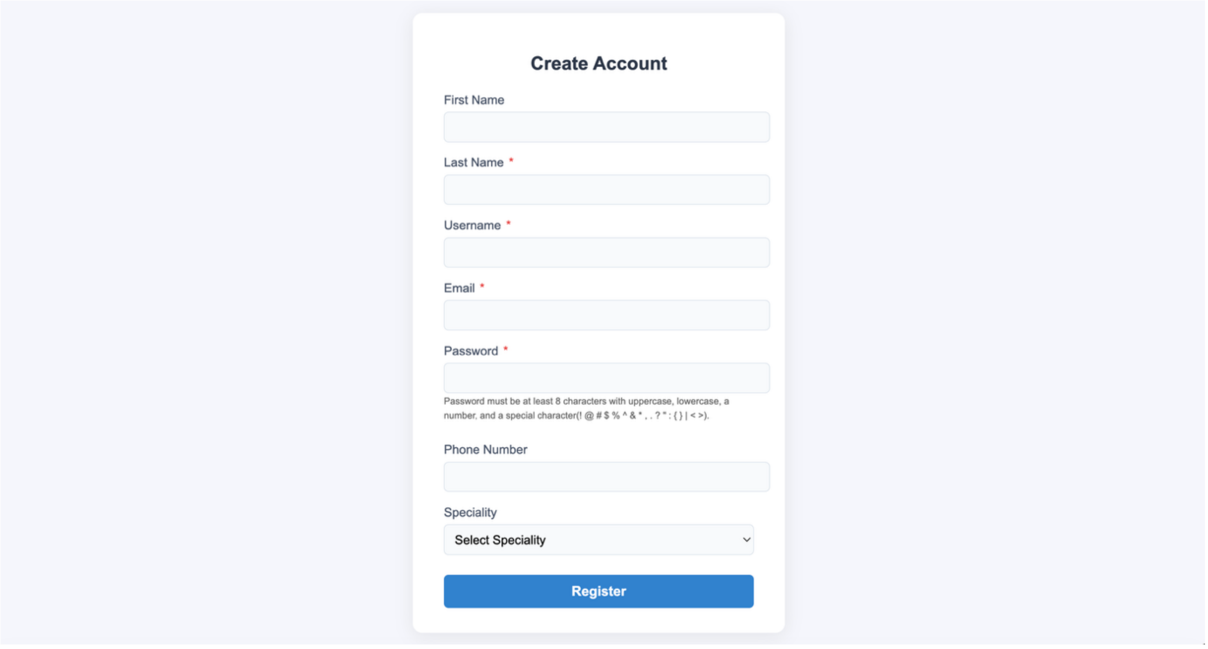
NOTE: For security reasons, the environment file (.env) is not committed to the repository. Instead, an example configuration file (.env.example) is provided.

3. EHR Information System

This section explains how to use the system and what each module supports. The system is intended to be straightforward for doctors, but the step-by-step guide clarifies workflows and helps ensure features are used correctly.

3.1 Registration

To use the system, doctors must first create an account. Figure 1 below shows the registration page.



Create Account

First Name

Last Name *

Username *

Email *

Password *

Password must be at least 8 characters with uppercase, lowercase, a number, and a special character(! @ # $ % ^ & * , . ? " : { } | < >).

Phone Number

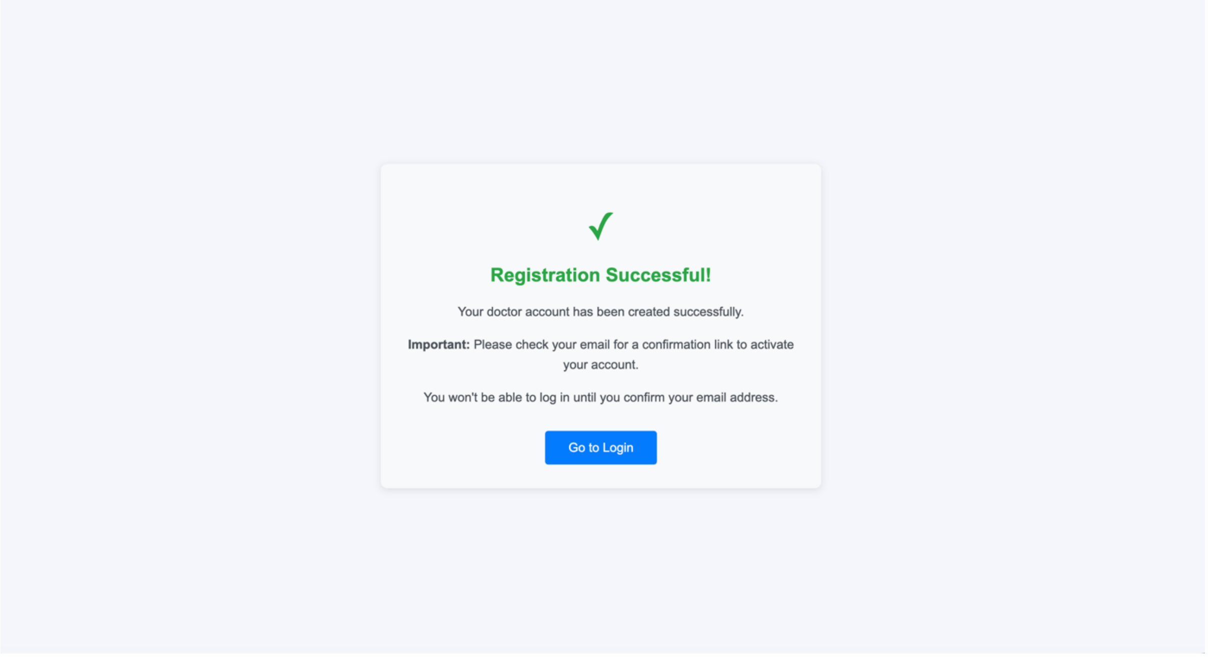
Speciality

Select Speciality

The form collects various data, with required fields marked by an asterisk (*). Although the database supports multiple specialties per doctor, the current interface only allows for the selection of one specialty.

When submitted, the system validates inputs and provides specific error messages if a field fails checks (for example, a duplicate email or weak password). On successful registration, the password is hashed and stored, and an email confirmation link is sent. The doctor must confirm their email before they can log in.

Figure 2 below shows the registration success page.



✓

Registration Successful!

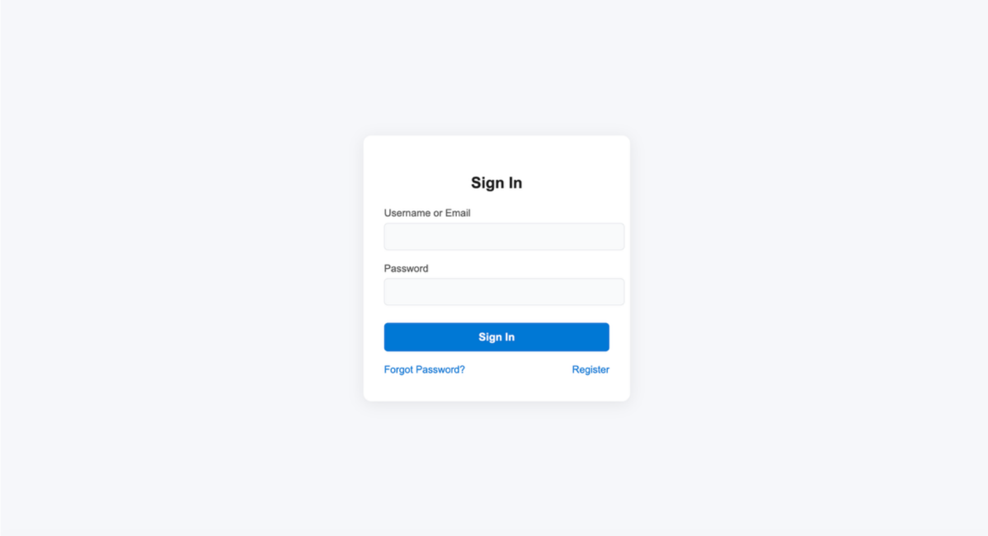
Your doctor account has been created successfully.

Important: Please check your email for a confirmation link to activate your account.

You won't be able to log in until you confirm your email address.

3.2 Login

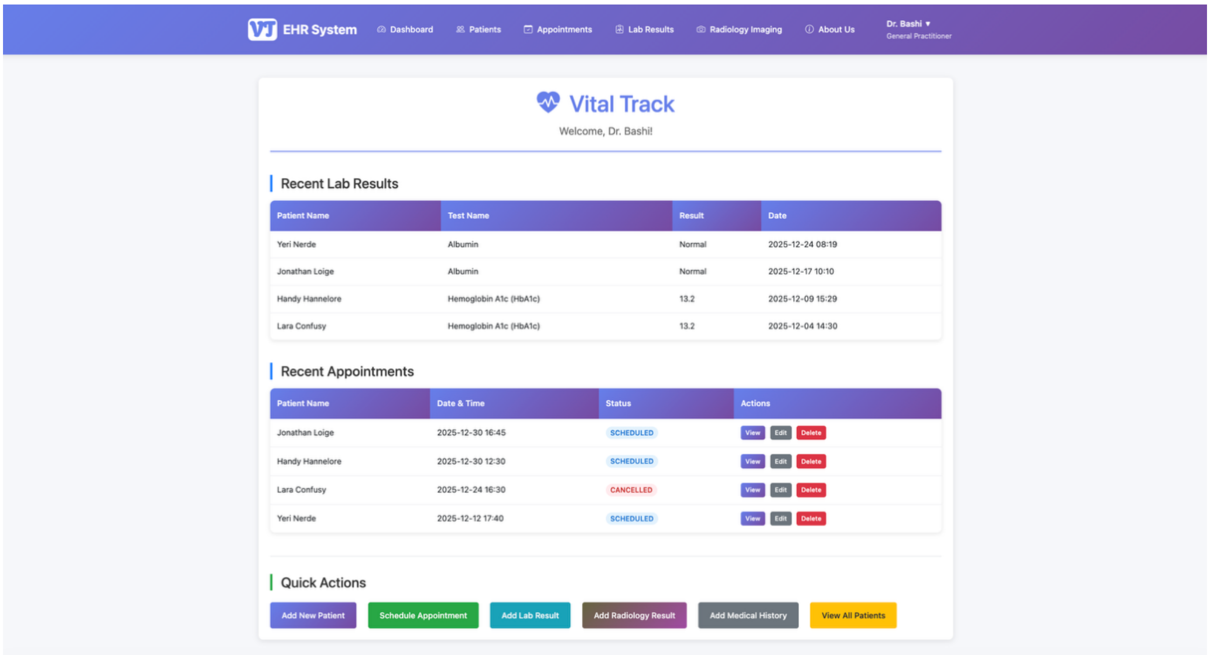
Figure 3 below shows the login page. Doctors can log in using either their username or email, plus their password.



The system looks up the doctor by username/email, checks the password against the stored hash, and blocks login if the email is not confirmed. When authentication succeeds, the system creates a session and redirects the doctor to the dashboard. Session data includes the doctor’s ID, name, specialty (currently treated as a single value), and a logged_in flag.

3.3 Dashboard

Figure 4 below shows the dashboard, which acts as the system’s main hub.



Welcome, Dr. Bashi!

Recent Lab Results

Patient Name	Test Name	Result	Date
Yeri Nerde	Albumin	Normal	2025-12-24 08:19
Jonathan Loige	Albumin	Normal	2025-12-17 10:10
Handy Hannelore	Hemoglobin A1c (HbA1c)	13.2	2025-12-09 15:29
Lara Confusy	Hemoglobin A1c (HbA1c)	13.2	2025-12-04 14:30

Recent Appointments

Patient Name	Date & Time	Status	Actions
Jonathan Loige	2025-12-30 16:45	SCHEDULED	View Edit Delete
Handy Hannelore	2025-12-30 12:30	SCHEDULED	View Edit Delete
Lara Confusy	2025-12-24 16:30	CANCELLED	View Edit Delete
Yeri Nerde	2025-12-12 17:40	SCHEDULED	View Edit Delete

Quick Actions

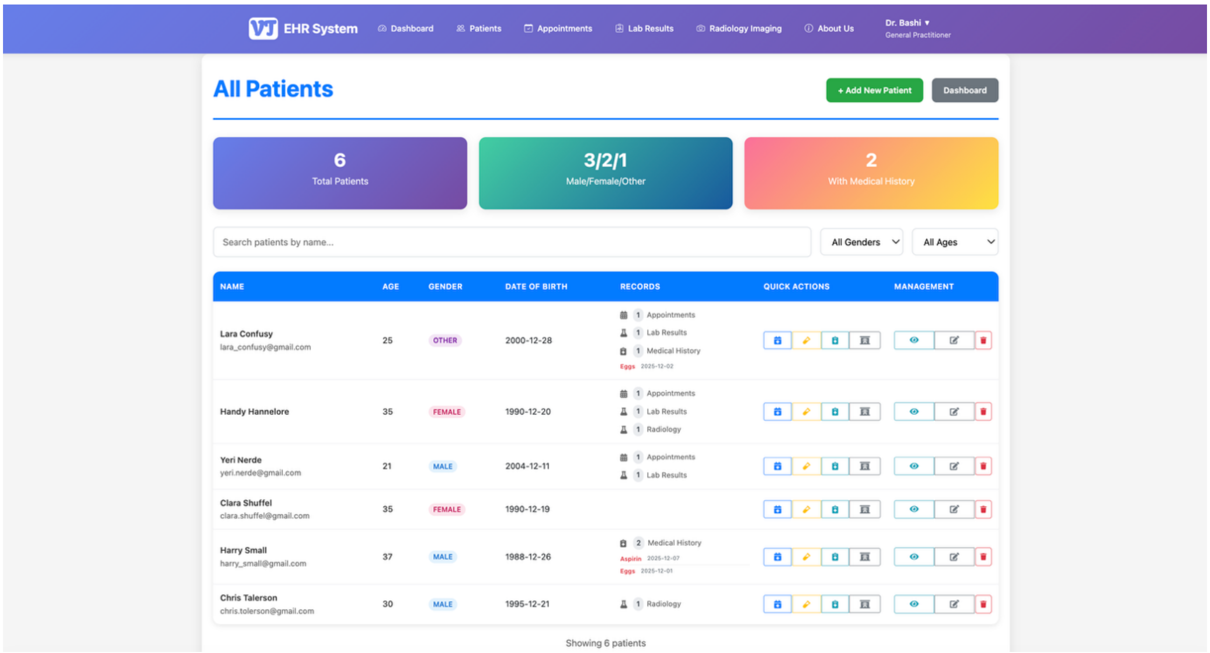
The dashboard displays recent activity across the doctor's patients: up to 10 recent lab results and up to 10 recent appointments. It also provides quick navigation cards for common tasks such as adding a patient, scheduling an appointment, recording lab results, and uploading radiology images. The goal is to reduce clicks for frequent actions and provide a quick overview of what has changed recently.

3.4 Patient Management

3.4.1 Viewing All Patients

Patients are displayed in a table with basic details (name, email, phone, age, gender). The page also shows counts of related records (appointments, lab results, radiology images, and medical history entries) and a small view of recent medical history entries, especially allergies. A statistics panel summarises totals and distributions (patient count, gender, age groups, and number of patients with medical history). Each patient row includes actions to view, edit, or delete.

Figure 5 shows the patient list page.



All Patients

6 Total Patients

3/2/1 Male/Female/Other

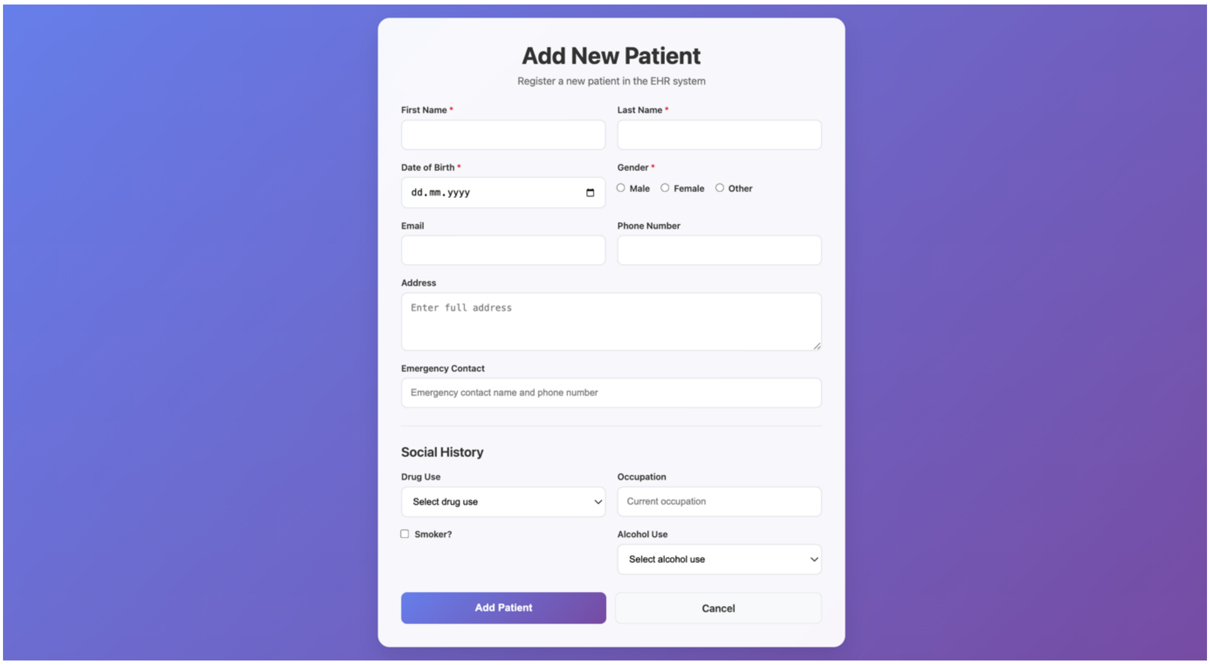
2 With Medical History

NAME	AGE	GENDER	DATE OF BIRTH	RECORDS	QUICK ACTIONS	MANAGEMENT
Lara Confusy lara_confusy@gmail.com	25	OTHER	2000-12-28	1 Appointments 1 Lab Results 1 Medical History Eggs 2025-12-02		
Handy Hannelore	35	FEMALE	1990-12-20	1 Appointments 1 Lab Results 1 Radiology		
Yeri Nerde yeri.nerde@gmail.com	21	MALE	2004-12-11	1 Appointments 1 Lab Results		
Clara Shuffel clara.shuffel@gmail.com	35	FEMALE	1990-12-19			
Harry Small harry_small@gmail.com	37	MALE	1988-12-26	2 Medical History Aspirin 2025-12-07 Eggs 2025-12-01		
Chris Talerson chris.tolerson@gmail.com	30	MALE	1995-12-21	1 Radiology		

Showing 6 patients

Patients are displayed in a table with basic details. The page also shows counts of related records and a small view of recent medical history entries, especially allergies. A statistics panel summarises totals and distributions. Each patient row includes actions to view, edit, or delete.

3.4.2 Adding a New Patient



Add New Patient

Register a new patient in the EHR system

First Name *

Last Name *

Date of Birth * dd.mm.yyyy

Gender * ○ Male ○ Female ○ Other

Email

Phone Number

Address: Enter full address

Emergency Contact: Emergency contact name and phone number

Social History

Drug Use: Select drug use

Occupation: Current occupation

□ Smoker?

Alcohol Use: Select alcohol use

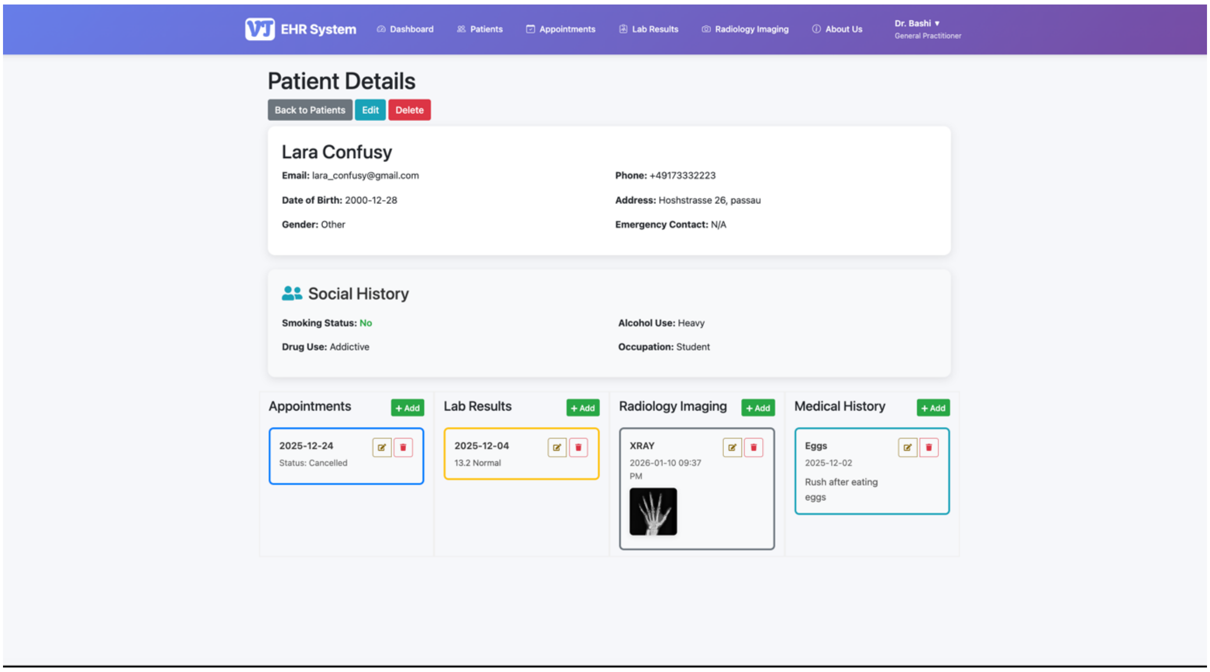
Figure 6 shows the add patient form.

The form collects various data, with required fields marked by an asterisk (*). The system validates required fields and date of birth before creating the patient record linked to the logged-in doctor.

3.4.3 Viewing Patient Details

This page combines: basic patient information, demographic details, social history (if available), and recent clinical records. It shows the last five appointments, lab results, radiology entries, and medical history items. Action buttons allow the doctor to edit patient data and add new records directly from the patient page, which keeps workflow centred on the patient rather than forcing the user to move across modules repeatedly.

Figure 7 shows the patient detail page.



Patient Details

Lara Confusy

Email: lara_confusy@gmail.com

Phone: +49173332223

Date of Birth: 2000-12-28

Address: Hoshstrasse 26, passau

Gender: Other

Emergency Contact: N/A

Social History

Smoking Status: No

Alcohol Use: Heavy

Drug Use: Addictive

Occupation: Student

Appointments

2025-12-24
Status: Cancelled

Lab Results

2025-12-04
13.2 Normal

Radiology Imaging

XRAY
2026-01-10 09:37 PM

Medical History

Eggs
2025-12-02
Rush after eating eggs

3.4.4 Editing Patient Information

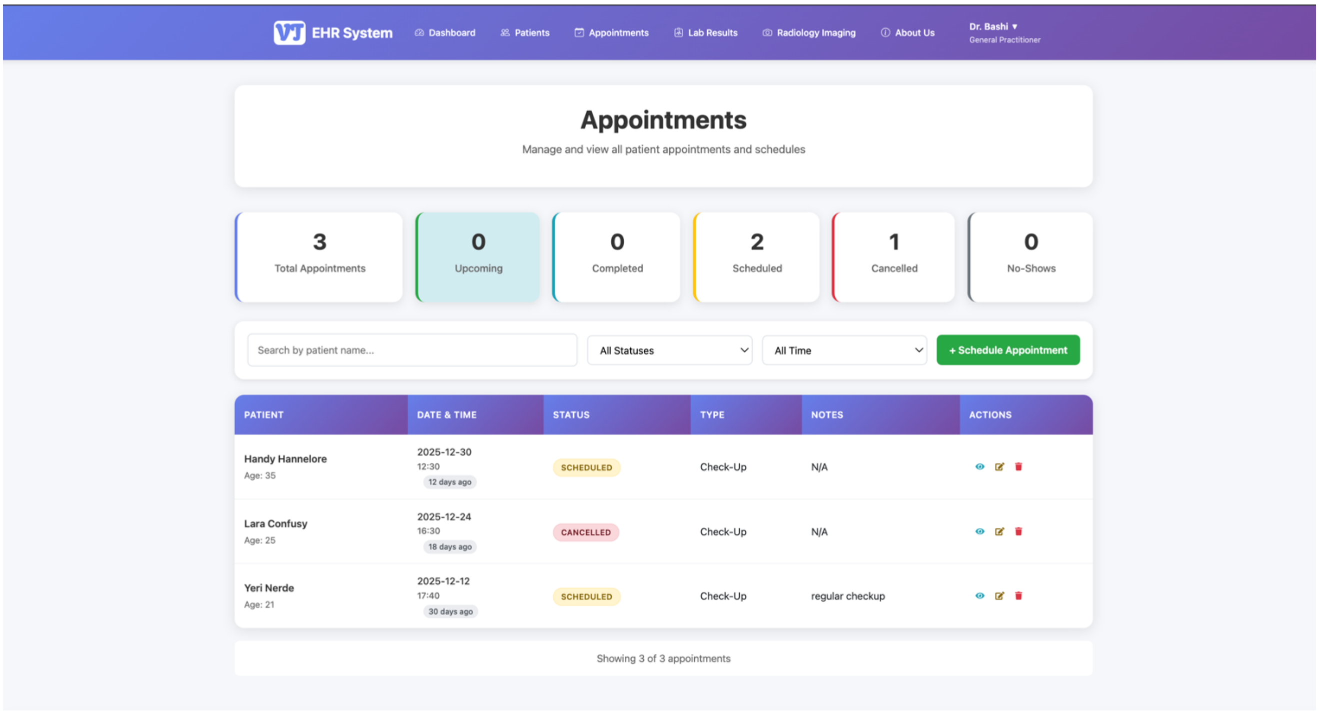
Doctors can update fields and save changes. Validation rules remain the same as when adding a patient, which helps prevent incomplete or inconsistent records.

3.4.5 Deleting Patients

Doctors can delete patient records. When a patient is deleted, related records (appointments, lab results, radiology entries, and medical history) are also removed to maintain database integrity. The system asks for confirmation before deletion.

3.5 Appointment Management

3.5.1 Viewing Appointments



Appointments

Manage and view all patient appointments and schedules

3	0	0	2	1	0
Total Appointments	Upcoming	Completed	Scheduled	Cancelled	No-Shows

PATIENT	DATE & TIME	STATUS	TYPE	NOTES	ACTIONS
Handy Hannelore Age: 35	2025-12-30 12:30 12 days ago	SCHEDULED	Check-Up	N/A	👁 ✎ 🗑
Lara Confusy Age: 25	2025-12-24 16:30 18 days ago	CANCELLED	Check-Up	N/A	👁 ✎ 🗑
Yeri Nerde Age: 21	2025-12-12 17:40 30 days ago	SCHEDULED	Check-Up	regular checkup	👁 ✎ 🗑

Showing 3 of 3 appointments

Figure 9 shows the appointments list page.

Appointments display patient name, type, date/time, status, and notes. A statistics area summarises totals and status breakdown (upcoming, completed, cancelled, no-show). Each appointment can be viewed, edited, or deleted.

3.5.2 Scheduling an Appointment

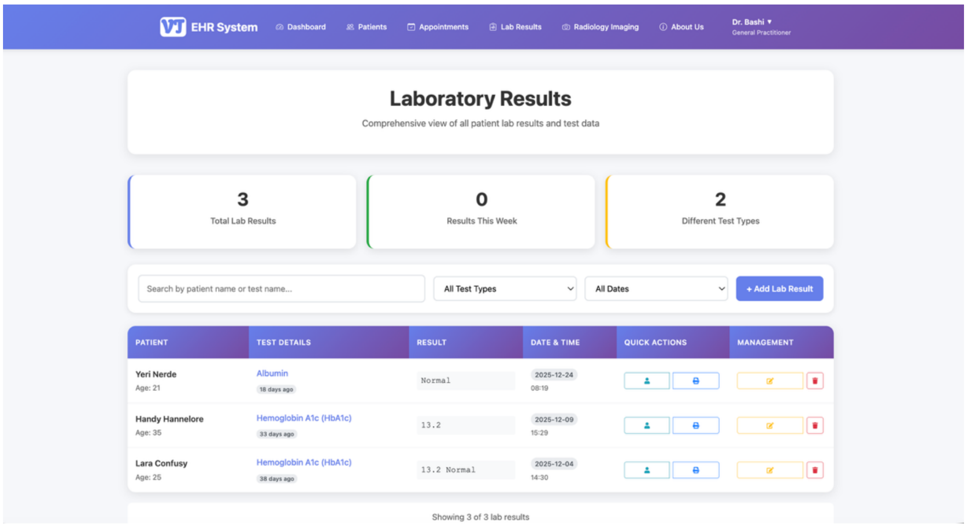
The doctor selects a patient, chooses date/time, optionally selects appointment type, and can add notes. The system validates required fields and checks for scheduling conflicts (another appointment at the same time). After saving, the appointment appears in the list and dashboard.

3.5.3 Viewing and Editing Appointments

Doctors can update time, type, status, and notes. Conflict checks apply during edits to prevent overlapping appointments.

3.6 Laboratory Results Management

3.6.1 Viewing Lab Results



Laboratory Results

Comprehensive view of all patient lab results and test data

3 Total Lab Results	0 Results This Week	2 Different Test Types

Search by patient name or test name...

All Test Types

All Dates

+ Add Lab Result

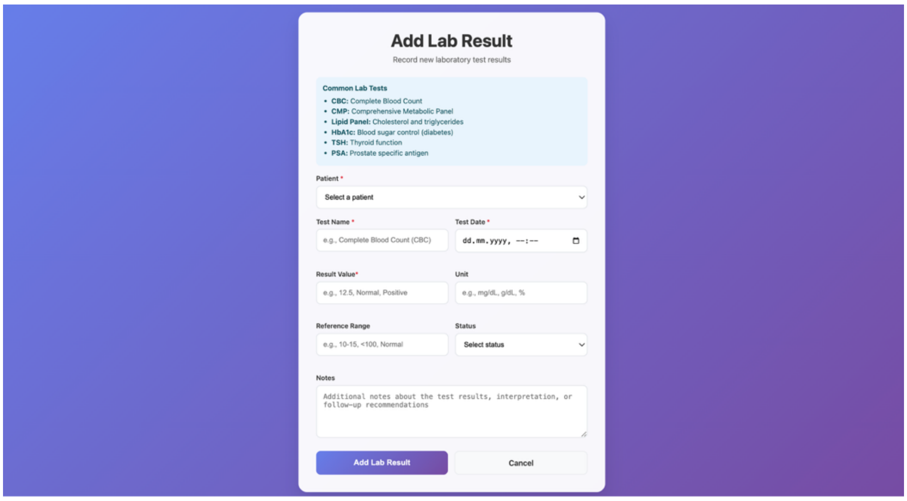
PATIENT	TEST DETAILS	RESULT	DATE & TIME	QUICK ACTIONS	MANAGEMENT
Yeri Nerde Age: 21	Albumin 18 days ago	Normal	2025-12-24 08:19		
Handy Hannelore Age: 35	Hemoglobin A1c (HbA1c) 33 days ago	13.2	2025-12-09 15:29		
Lara Confusy Age: 25	Hemoglobin A1c (HbA1c) 38 days ago	13.2 Normal	2025-12-04 14:30		

Showing 3 of 3 lab results

Figure 12 shows the lab results list page.

Each result includes patient name, test name, result value, unit and reference range (if provided), status label, date, and notes. The page summarises totals and recent results and lists the test types stored in the system.

3.6.2 Adding Lab Results



Add Lab Result

Record new laboratory test results

Common Lab Tests

- **CBC:** Complete Blood Count
- **CMP:** Comprehensive Metabolic Panel
- **Lipid Panel:** Cholesterol and triglycerides
- **HbA1c:** Blood sugar control (diabetes)
- **TSH:** Thyroid function
- **PSA:** Prostate specific antigen

Patient *: Select a patient

Test Name *: e.g., Complete Blood Count (CBC)

Test Date *: dd.mm.yyyy, --:--

Result Value*: e.g., 12.5, Normal, Positive

Unit: e.g., mg/dL, g/dL, %

Reference Range: e.g., 10-15, <100, Normal

Status: Select status

Notes: Additional notes about the test results, interpretation, or follow-up recommendations

Figure 13 shows the add lab result form.

Doctors select a patient and enter test name, test date, and result value. Unit, reference range, status, and notes are optional but supported. The system validates required fields and ensures the patient belongs to the logged-in doctor. Saved results appear in the patient record and on the dashboard.

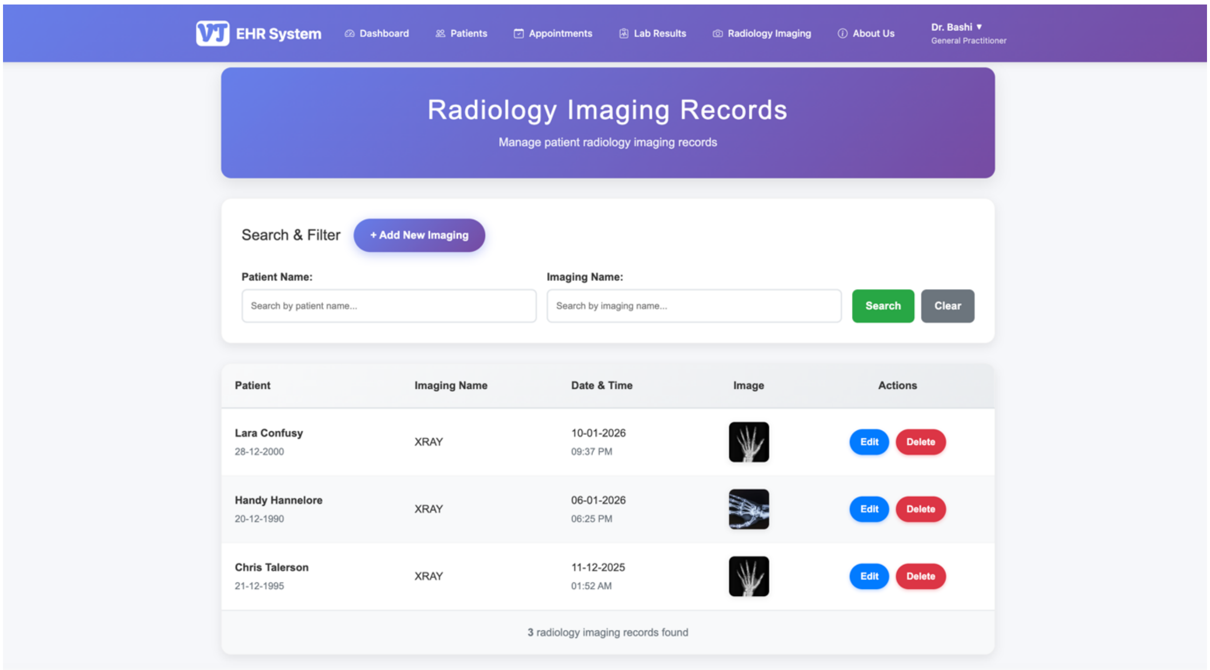
3.6.3 Editing and Deleting Lab Results

Doctors can edit lab results using a pre-filled form and can delete results that were entered incorrectly. This keeps records maintainable without requiring database access outside the application.

3.7 Radiology Imaging Management

3.7.1 Viewing Radiology Images

Figure 14 shows the radiology imaging list page.



Radiology Imaging Records

Manage patient radiology imaging records

Search & Filter

Patient Name: Search by patient name...

Imaging Name: Search by imaging name...

Patient	Imaging Name	Date & Time	Image	Actions
Lara Confusy 28-12-2000	XRAY	10-01-2026 09:37 PM		Edit Delete
Handy Hannelore 20-12-1990	XRAY	06-01-2026 06:25 PM		Edit Delete
Chris Talerson 21-12-1995	XRAY	11-12-2025 01:52 AM		Edit Delete

3 radiology imaging records found

Records display patient name, imaging name/type, date, and a thumbnail preview if an image was uploaded. Search supports filtering by patient name or imaging name.

3.7.2 Adding Radiology Imaging

Doctors select a patient, enter imaging name/type and date, and optionally upload an image file (supported formats include common image types and DICOM-related extensions). The system enforces file type validation and a maximum size of 10MB. Files are stored in static/uploads/radiology/patient_X/, and the filename is saved in the database for retrieval.

3.7.3 Editing and Deleting Radiology Records

Doctors can edit imaging metadata (name/date). When deleting a radiology record, the system also deletes the associated image file to avoid unused storage and leftover data.

3.8 Medical History Management

3.8.1 Adding Medical History

Medical history entries link patients to allergies or conditions. Doctors select a patient, enter or select an allergy/condition name, add a description (reaction details or clinical notes), and provide a date. If the allergy/condition name does not exist, the system creates a new Allergy entry automatically, allowing flexible documentation without losing consistency across patients.

3.8.2 Viewing Medical History

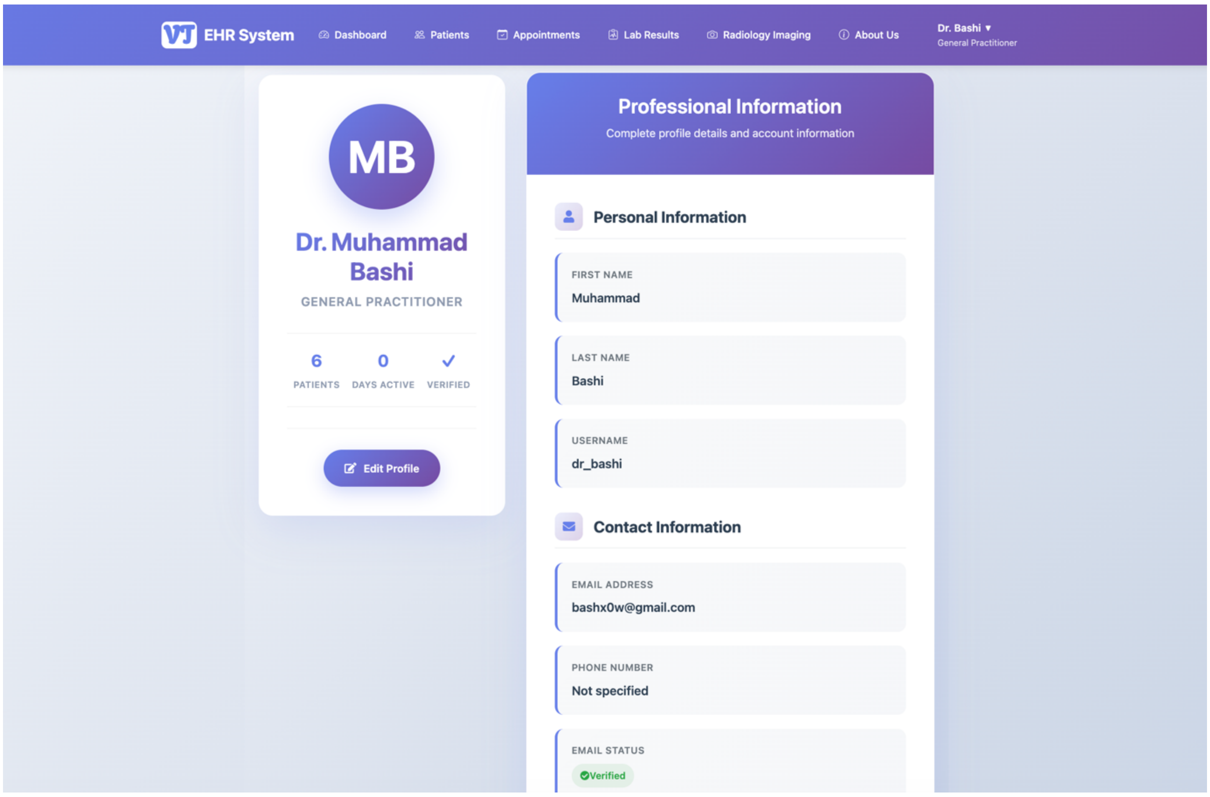
Medical history entries appear in multiple places: the patient detail page, the patient list page (recent entries), and when reviewing an individual patient record. Entries display the allergy/condition name, description, and date.

3.8.3 Editing and Deleting Medical History

Doctors can edit medical history entries to correct details and can delete entries that were added incorrectly.

3.9 Doctor Profile Management

Figure 17 shows the doctor profile page.



Dr. Muhammad Bashi

GENERAL PRACTITIONER

6	0	✓
PATIENTS	DAYS ACTIVE	VERIFIED

Professional Information

Complete profile details and account information

Personal Information

FIRST NAME
Muhammad

LAST NAME
Bashi

USERNAME
dr_bashi

Contact Information

EMAIL ADDRESS
bashx0w@gmail.com

PHONE NUMBER
Not specified

EMAIL STATUS
Verified

Doctors can view and update first name, last name, email address, and phone number. The system validates updates and ensures email addresses remain unique.

3.10 Password Management

3.10.1 Forgot Password

Figure 18 shows the forgot password form.

Doctors can request a reset by entering their email address. The system sends a reset link containing a secure token that expires after 24 hours.

3.10.2 Reset Password

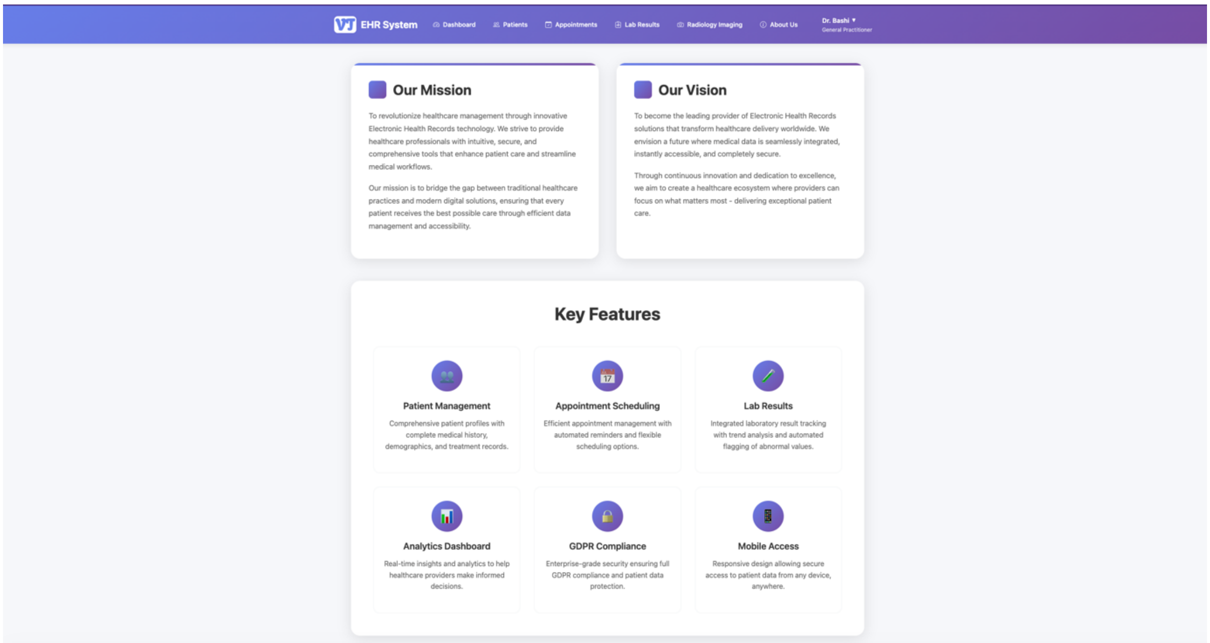
Figure 19 shows the reset password form.

Doctors enter a new password that must meet the same strength rules as registration. After saving, they can log in with the new password.

3.11 About Us Page

Figure 20 and Figure 21 show the About Us page.

This page provides system-wide statistics: total patients, appointments, lab results, radiology records, and registered doctors. It gives a simple view of system usage and helps demonstrate scale during testing or presentation.



Our Mission

To revolutionize healthcare management through innovative Electronic Health Records technology. We strive to provide healthcare professionals with intuitive, secure, and comprehensive tools that enhance patient care and streamline medical workflows.

Our mission is to bridge the gap between traditional healthcare practices and modern digital solutions, ensuring that every patient receives the best possible care through efficient data management and accessibility.

Our Vision

To become the leading provider of Electronic Health Records solutions that transform healthcare delivery worldwide. We envision a future where medical data is seamlessly integrated, instantly accessible, and completely secure.

Through continuous innovation and dedication to excellence, we aim to create a healthcare ecosystem where providers can focus on what matters most - delivering exceptional patient care.

Key Features

Patient Management
Comprehensive patient profiles with complete medical history, demographics, and treatment records.

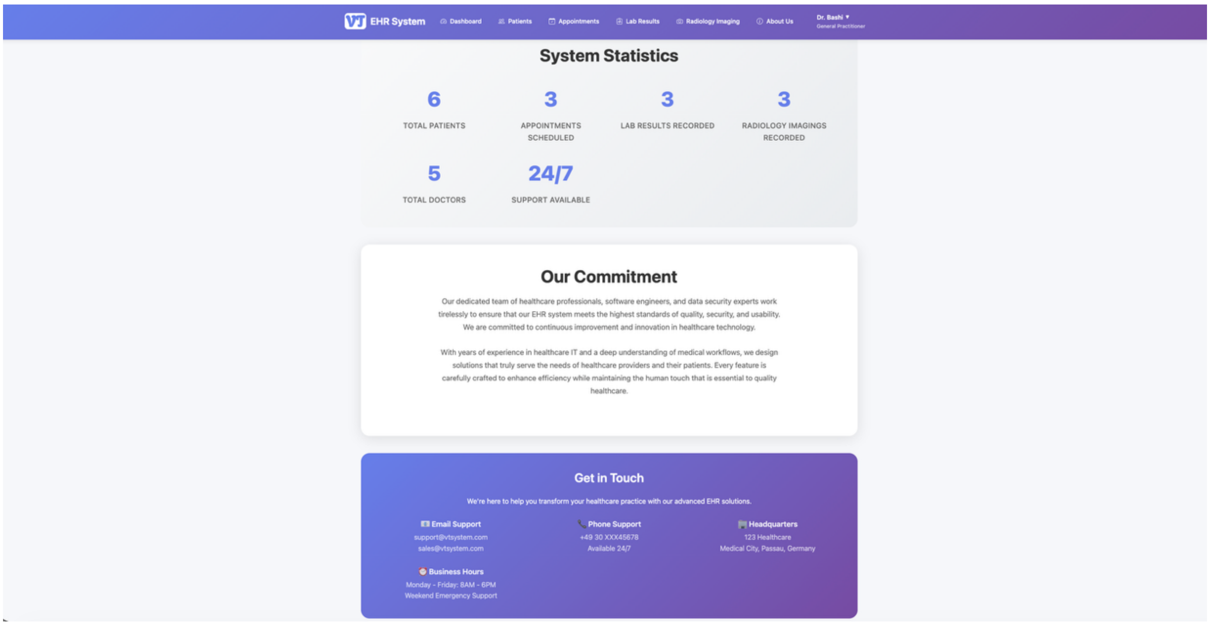
Appointment Scheduling
Efficient appointment management with automated reminders and flexible scheduling options.

Lab Results
Integrated laboratory result tracking with trend analysis and automated flagging of abnormal values.

Analytics Dashboard
Real-time insights and analytics to help healthcare providers make informed decisions.

GDPR Compliance
Enterprise-grade security ensuring full GDPR compliance and patient data protection.

Mobile Access
Responsive design allowing secure access to patient data from any device, anywhere.



System Statistics

6	3	3	3
TOTAL PATIENTS	APPOINTMENTS SCHEDULED	LAB RESULTS RECORDED	RADIOLOGY IMAGINGS RECORDED
5	24/7		
TOTAL DOCTORS	SUPPORT AVAILABLE		

Our Commitment

Our dedicated team of healthcare professionals, software engineers, and data security experts work tirelessly to ensure that our EHR system meets the highest standards of quality, security, and usability. We are committed to continuous improvement and innovation in healthcare technology.

With years of experience in healthcare IT and a deep understanding of medical workflows, we design solutions that truly serve the needs of healthcare providers and their patients. Every feature is carefully crafted to enhance efficiency while maintaining the human touch that is essential to quality healthcare.

Get in Touch

We're here to help you transform your healthcare practice with our advanced EHR solutions.

📧 **Email Support**
support@vtsystem.com
sales@vtsystem.com

📞 **Phone Support**
+49 30 XXX45678
Available 24/7

🏢 **Headquarters**
123 Healthcare
Medical City, Passau, Germany

⏰ **Business Hours**
Monday - Friday: 8AM - 6PM
Weekend Emergency Support

3.12 System Features Summary

The EHR system supports essential record-keeping tasks for a doctor managing their own patients. Patient management includes creating, viewing, editing, and deleting patient profiles with demographic and social history data. Appointment scheduling supports different types and statuses and prevents double booking. Laboratory results and radiology imaging modules store diagnostic information with structured fields and optional notes. Medical history tracking focuses on allergies and conditions, allowing entries to be added, edited, and reviewed within patient context. Security measures—hashed passwords, confirmation and reset workflows, access control by doctor ID, safer configuration management, and file upload checks—support responsible handling of sensitive data within the scope of a university project.

4. Evaluation of the EHR Information System

To evaluate usability, I used the System Usability Scale (SUS), a common questionnaire-based method for measuring perceived usability. SUS provides a single score from 0 to 100 based on responses to ten statements. It is widely used because it is quick to administer and easy to interpret, especially for early-stage systems.

4.1 SUS Methodology

SUS consists of 10 statements rated on a 5-point Likert scale from “Strongly Disagree” (1) to “Strongly Agree” (5). The statements alternate between positive and negative wording. Scoring follows a standard transformation: odd-numbered items are scored as (response − 1), even-numbered items as (5 − response). The adjusted values are summed and multiplied by 2.5 to produce the final SUS score (Brooke, 1996).

The SUS statements used were:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

4.2 Evaluation Process

For this project, I asked one evaluator (a fellow student) to use the system and complete the SUS questionnaire. Although SUS is often used with multiple participants, using one evaluator allowed me to practise the method carefully and get direct feedback after a first-time use session.

The evaluator was asked to complete key tasks: register a doctor account, log in, add a patient, schedule an appointment, add a lab result, add a radiology imaging record, add medical history, view patient details, and then complete SUS. The evaluator used the system for approximately 15–20 minutes before submitting the questionnaire.

4.3 SUS Score Calculation

The evaluator’s responses were:

- Statement 1: 4
- Statement 2: 2
- Statement 3: 5
- Statement 4: 2
- Statement 5: 4
- Statement 6: 2
- Statement 7: 4
- Statement 8: 2
- Statement 9: 4
- Statement 10: 2

Adjusted scoring:

- Item 1: 4 − 1 = 3
- Item 2: 5 − 2 = 3
- Item 3: 5 − 1 = 4
- Item 4: 5 − 2 = 3
- Item 5: 4 − 1 = 3
- Item 6: 5 − 2 = 3
- Item 7: 4 − 1 = 3
- Item 8: 5 − 2 = 3
- Item 9: 4 − 1 = 3
- Item 10: 5 − 2 = 3

Sum = 31

SUS Score = 31 × 2.5 = 77.5

4.4 SUS Score Interpretation

SUS scores range from 0 to 100. A commonly used benchmark for “average” usability is around 68, and scores above that are typically interpreted as above average. A score of 77.5 suggests the evaluator found the system generally easy to use and learn. Since this was based on a single participant, the score should be treated as an early indicator rather than a final judgement. Additional evaluations with more users would give a more reliable picture of usability strengths and weaknesses (Bangor et al., 2008).

5. Limitations/Future Improvements

Scalability is limited, and the system is best suited to a single doctor or small practice scenario. Data export and interoperability features are not yet implemented, so patient records cannot be exchanged using standard healthcare formats. Reporting is basic; the system does not generate summaries or trend views (for example, lab result trends over time). Mobile responsiveness relies mainly on Bootstrap defaults and could be improved for routine use on smaller screens. Backup and recovery are not automated, and there is no audit trail of access or changes, which would be important for compliance in real clinical settings. Security could be strengthened through multi-factor authentication, encryption at rest, and more detailed role-based access control. Finally, although the database structure supports multiple specialties per doctor, the interface currently supports only a single specialty selection. Extending this would require changes across registration, profile editing, and display logic to match the underlying model.

5. Conclusion

This project involved building a web-based Electronic Health Record information system using HTML, CSS, and Python with Flask. The system supports core EHR-style functions for a doctor managing patient records digitally: registration and secure login, patient profile creation and maintenance, appointment scheduling, lab result recording, radiology image uploads, and medical history tracking focused on allergies and conditions.

A key outcome of the project was learning how feature development connects to architectural and security decisions. The system evolved over time: password handling was improved through hashing, route organisation was made more maintainable through blueprints, and configuration secrets were separated into environment variables. File upload handling required practical safeguards, and access control needed to be enforced consistently at the database query level so that doctors only see their own patient data.

The system successfully implements the main functional goals, but it also has clear limitations that point toward future improvements.

References

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction, 24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Brooke, J. (1996). SUS: A “quick and dirty” usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.