E-Doc: an EHR system

By: Team AA

Ayesha Hamid & Muhammad Ahmed Shahab

1. Introduction

An Electronic Health Record (EHR) is a digital version of a patient's medical history, designed to streamline the collection, storage, and retrieval of health information. It allows healthcare providers to access accurate, up-to-date, and complete patient data in real-time, improving decision-making and enhancing patient care. Unlike traditional paper records, EHRs are accessible across authorized systems and can be updated collaboratively by multiple healthcare providers.

- **Importance of EHRs**

EHR systems play a vital role in modern healthcare. They reduce paperwork, minimize errors, and enhance data security while enabling effective communication between healthcare providers. By centralizing patient records, EHRs improve diagnosis accuracy, enable faster treatment, and provide better continuity of care. Moreover, these systems contribute to public health by facilitating data-driven decision-making and supporting medical research.

- **History and Evolution**

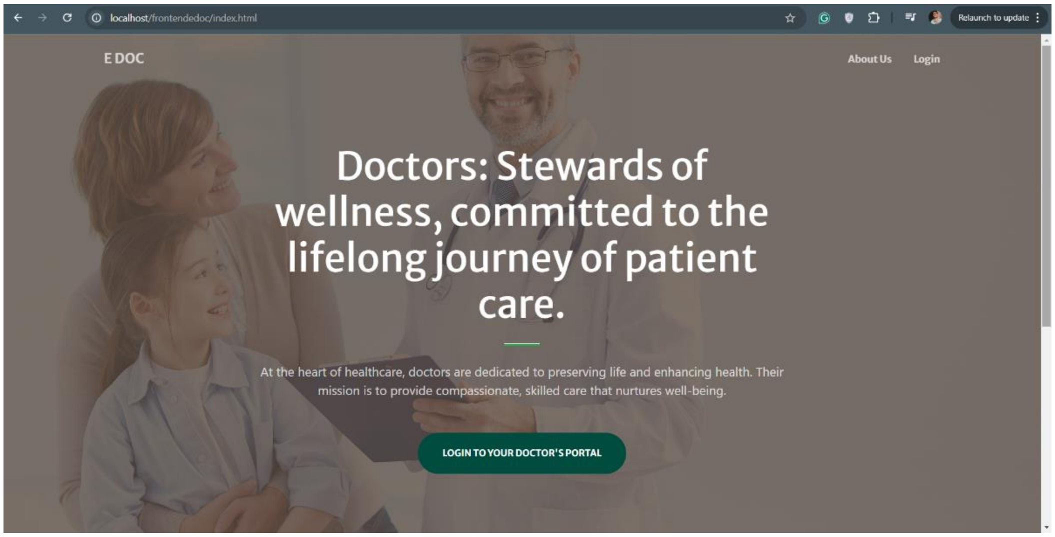
The concept of digitizing medical records dates back to the 1960s, when hospitals began experimenting with basic computer systems for managing patient data. By the 1990s, advancements in technology and the rise of the internet accelerated the adoption of EHR systems, particularly in developed nations. Governments and healthcare organizations worldwide encouraged the implementation of EHRs to standardize medical records and improve healthcare outcomes. Today, EHR systems integrate advanced technologies like cloud computing, artificial intelligence, and telemedicine to provide a comprehensive healthcare management solution.

- **How EHRs Work**

EHRs are typically accessed through secure web-based platforms or specialized software. Healthcare professionals log into the system using their credentials to record, update, or retrieve patient information. These systems store a wide range of data, including medical history, lab results, treatment plans, medications, and allergies. Features like role-based access control ensure that sensitive data is only accessible to authorized users, protecting patient confidentiality.

2. *website design and technical architecture*

The design and architecture of the EHR system are critical to ensuring a seamless user experience and robust performance. Below is an overview of the design process and the technical framework utilized for this project.


localhost/frontendedoc/index.html
Relaunch to update
E DOC
About Us
Login
LOGIN TO YOUR DOCTOR'S PORTAL

Doctors: Stewards of wellness, committed to the lifelong journey of patient care.

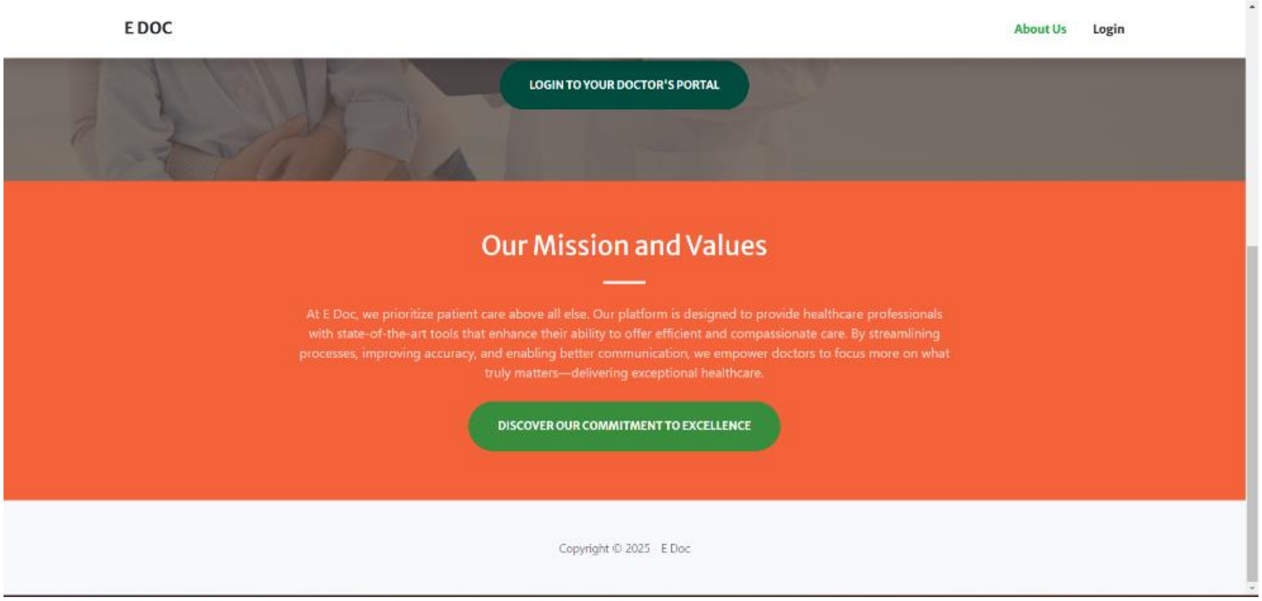
At the heart of healthcare, doctors are dedicated to preserving life and enhancing health. Their mission is to provide compassionate, skilled care that nurtures well-being.

Technical Architecture

The EHR system was developed using a modular and scalable architecture to ensure ease of maintenance and future extensibility.

1. Frontend:
    - Languages and Frameworks: HTML5, CSS3, and minimal JavaScript.
    - Design Approach: Responsive design principles were applied to ensure the application is accessible on both desktop and mobile devices.
2. Backend:
    - Languages: PHP and Python were used to handle server-side logic and integrations.
    - Frameworks: Flask for Python to handle API requests and routing.
3. Database:
    - Type: MySQL was chosen for its reliability and compatibility with the backend.
    - Schema: The database schema includes tables for users (doctors and patients), medical records, and authentication tokens.
4. Authentication:
    - JWT (JSON Web Token): Implemented for secure user authentication and session management.
5. Cross-Origin Resource Sharing (CORS):
    - Configured to enable secure communication between the frontend and backend.
6. Development Environment:
    - The project was primarily developed and tested on VS Code using a local Codespace environment for consistency.
    - Version Control: GitHub was used for version control to manage the project efficiently.

3. *EHR information system*

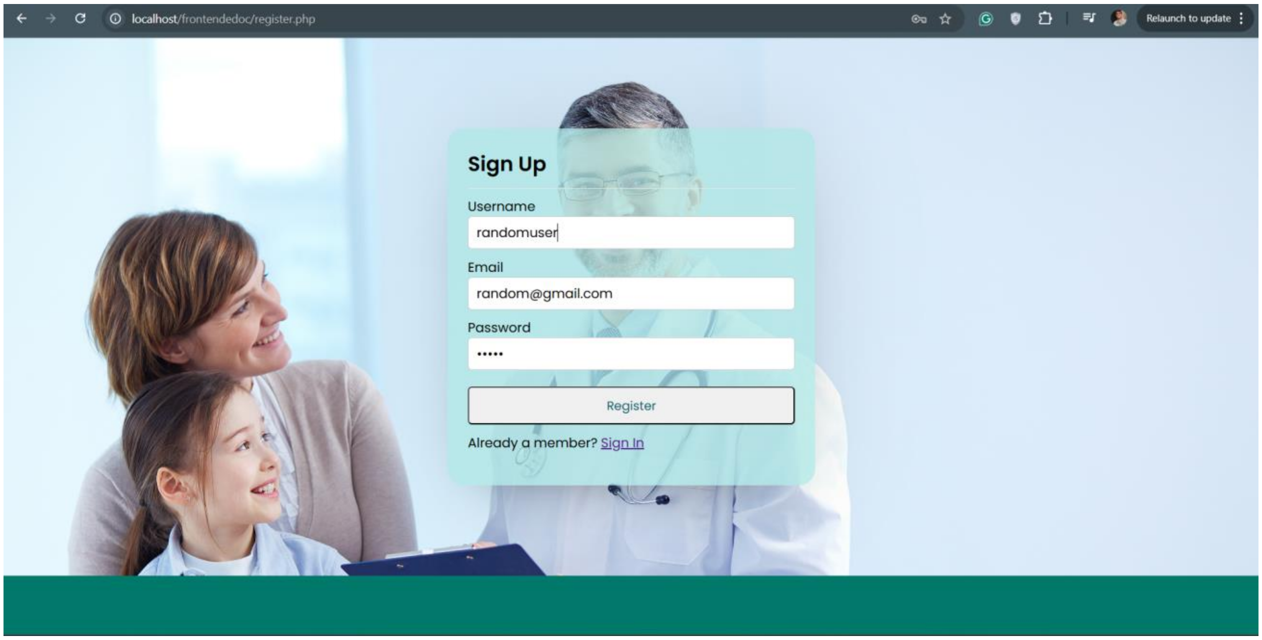

E DOC
About Us
Login
LOGIN TO YOUR DOCTOR'S PORTAL
DISCOVER OUR COMMITMENT TO EXCELLENCE

Our Mission and Values

At E Doc, we prioritize patient care above all else. Our platform is designed to provide healthcare professionals with state-of-the-art tools that enhance their ability to offer efficient and compassionate care. By streamlining processes, improving accuracy, and enabling better communication, we empower doctors to focus more on what truly matters—delivering exceptional healthcare.

Copyright © 2025 E Doc

This is our homepage which leads to the login page and other pages like our mission and goals.



Sign Up

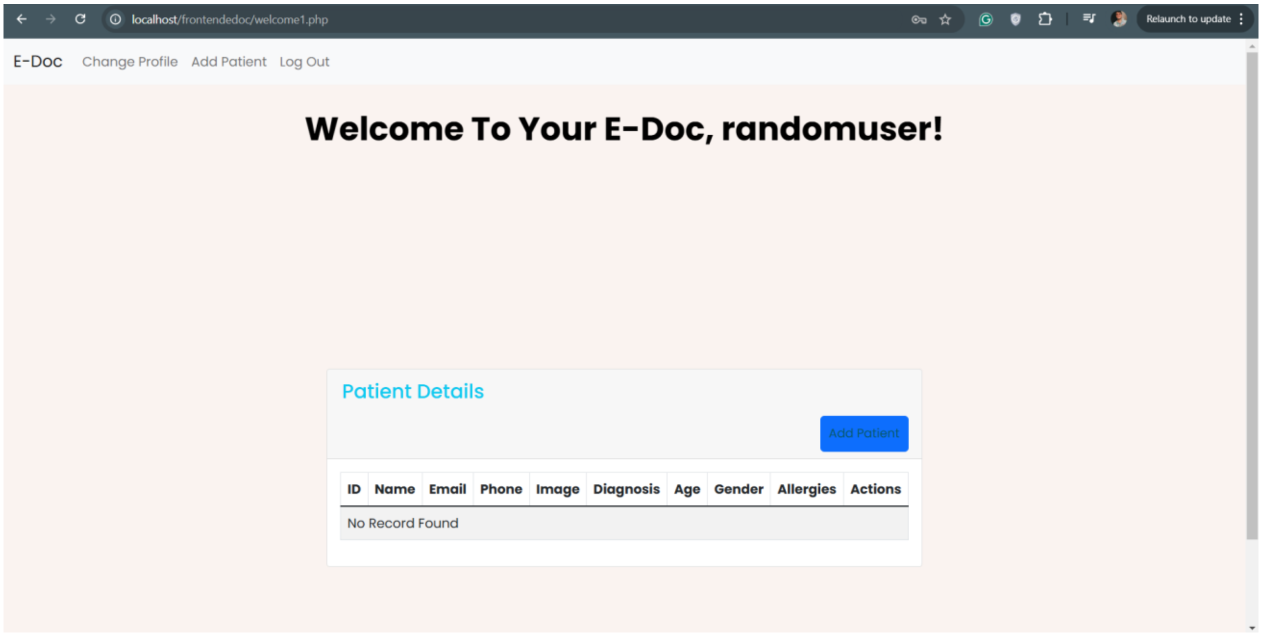
Username: randomuser

Email: random@gmail.com

Password: •••••

Already a member? Sign In

This is the new doctor registration page. If you have an account already you can click on sign in option.

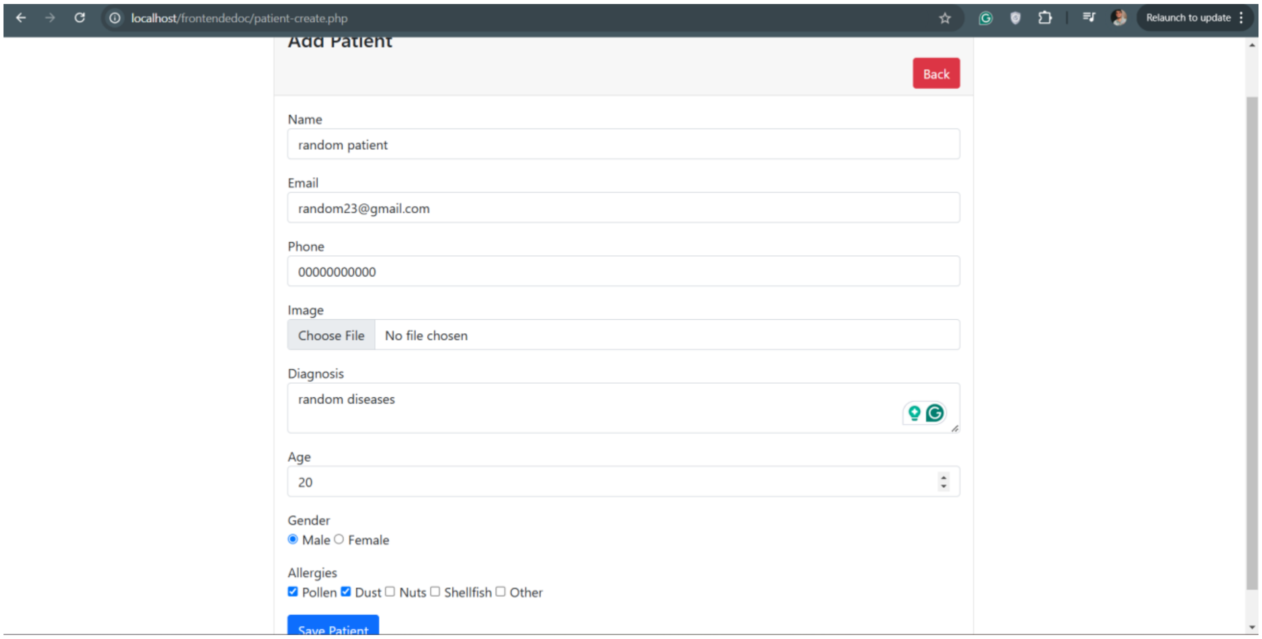


Welcome To Your E-Doc, randomuser!

Patient Details

ID	Name	Email	Phone	Image	Diagnosis	Age	Gender	Allergies	Actions
No Record Found									

This is what a new registered user sees. They can CRUD patients.



Add Patient

Name: random patient

Email: random23@gmail.com

Phone: 00000000000

Image: No file chosen

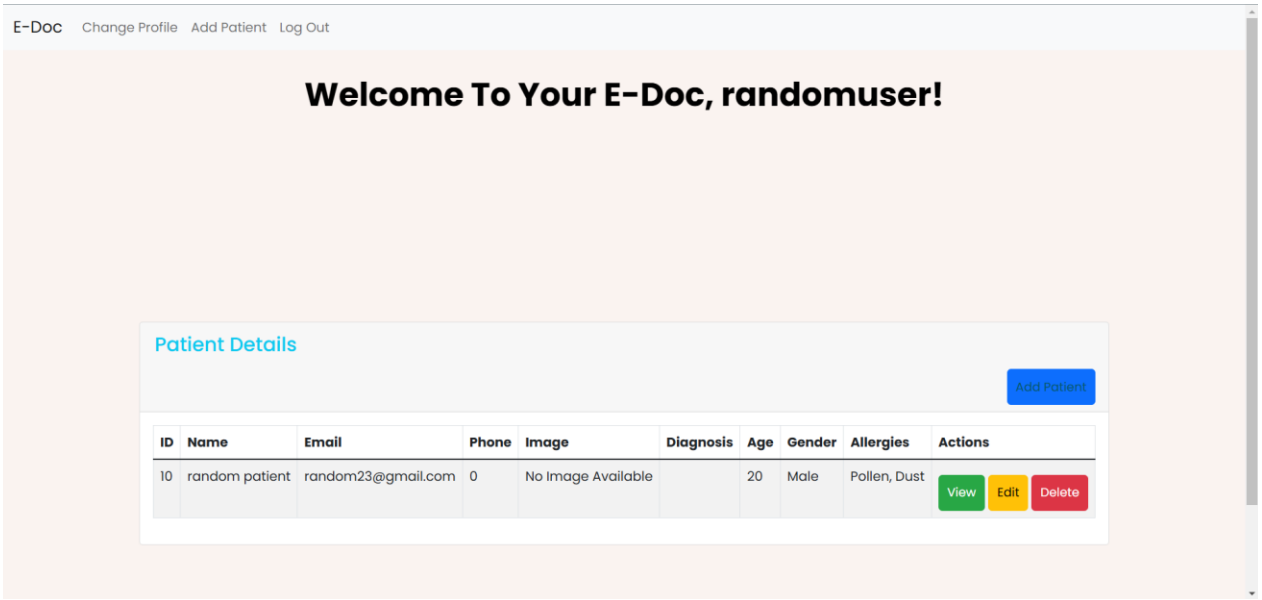
Diagnosis: random diseases

Age: 20

Gender
● Male ○ Female

Allergies
☑ Pollen ☑ Dust □ Nuts □ Shellfish □ Other

This is create/adding a new patient. It has 5 input data, 1 radio button, 1 check box and 1 image upload tab.

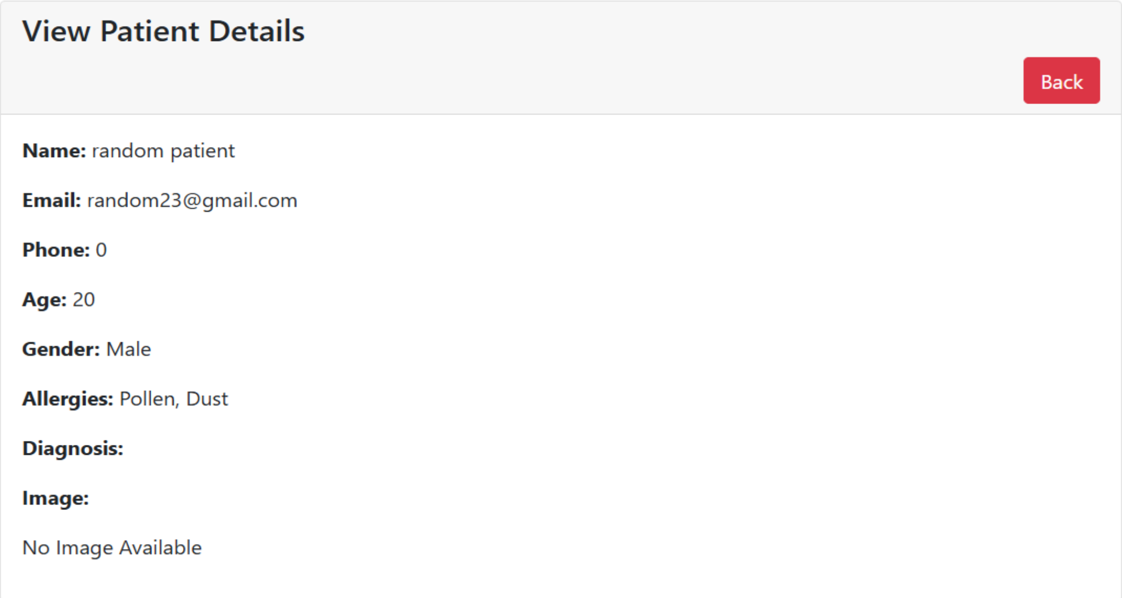


Welcome To Your E-Doc, randomuser!

Patient Details

ID	Name	Email	Phone	Image	Diagnosis	Age	Gender	Allergies	Actions
10	random patient	random23@gmail.com	0	No Image Available		20	Male	Pollen, Dust	View Edit Delete

This is what the main page looks like when a patient is created. You can View, Edit and Delete the data from the buttons on the right.


View Patient Details

Name: random patient

Email: random23@gmail.com

Phone: 0

Age: 20

Gender: Male

Allergies: Pollen, Dust

Diagnosis:

Image:

No Image Available

This is how view/read patient data looks like.

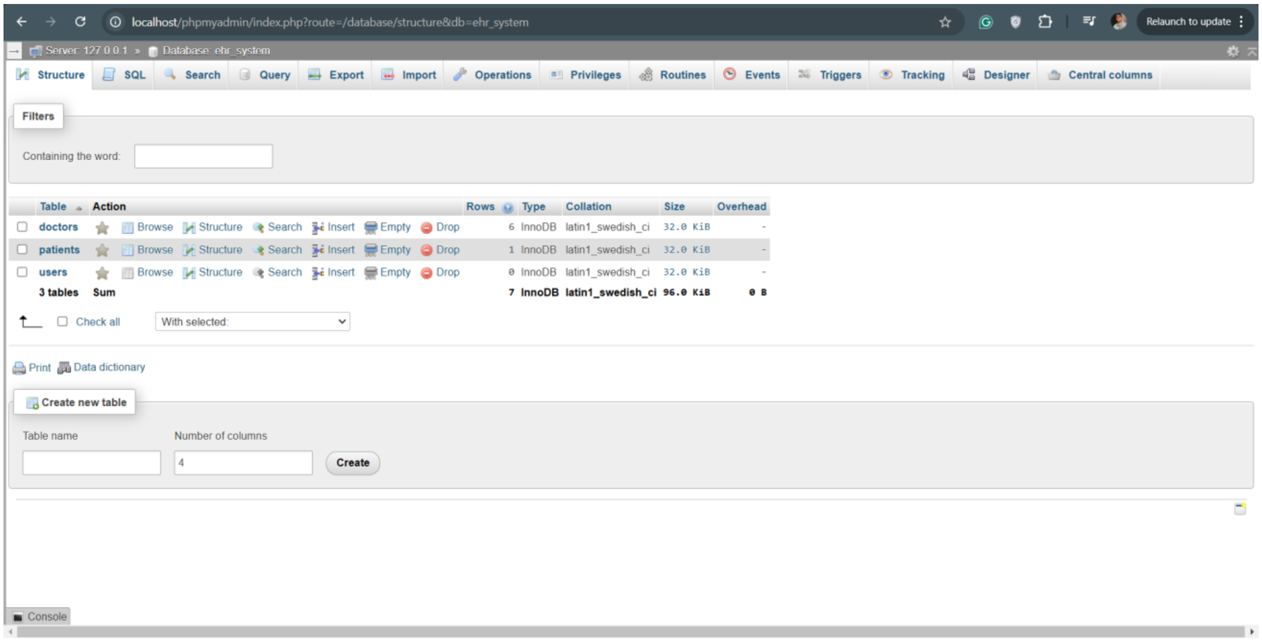
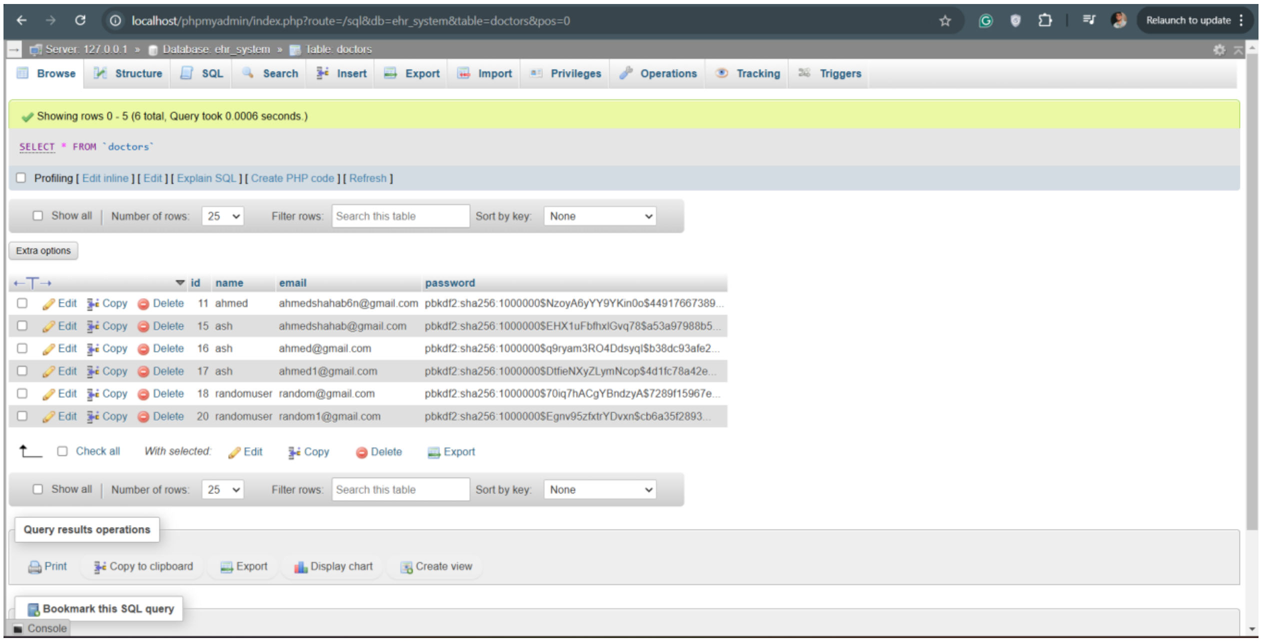


	Table	Action	Rows	Type	Collation	Size	Overhead
□	doctors	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	32.0 KiB	-
□	patients	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	32.0 KiB	-
□	users	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 KiB	-
	3 tables	Sum	7	InnoDB	latin1_swedish_ci	96.0 KiB	0 B

This is what the SQL database looks like, as you can see the main tables are doctors and patients.



localhost/phpmyadmin/index.php?route=/sql&db=ehr_system&table=doctors&pos=0
Relaunch to update
Server: 127.0.0.1 » Database: ehr_system » Table: doctors
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

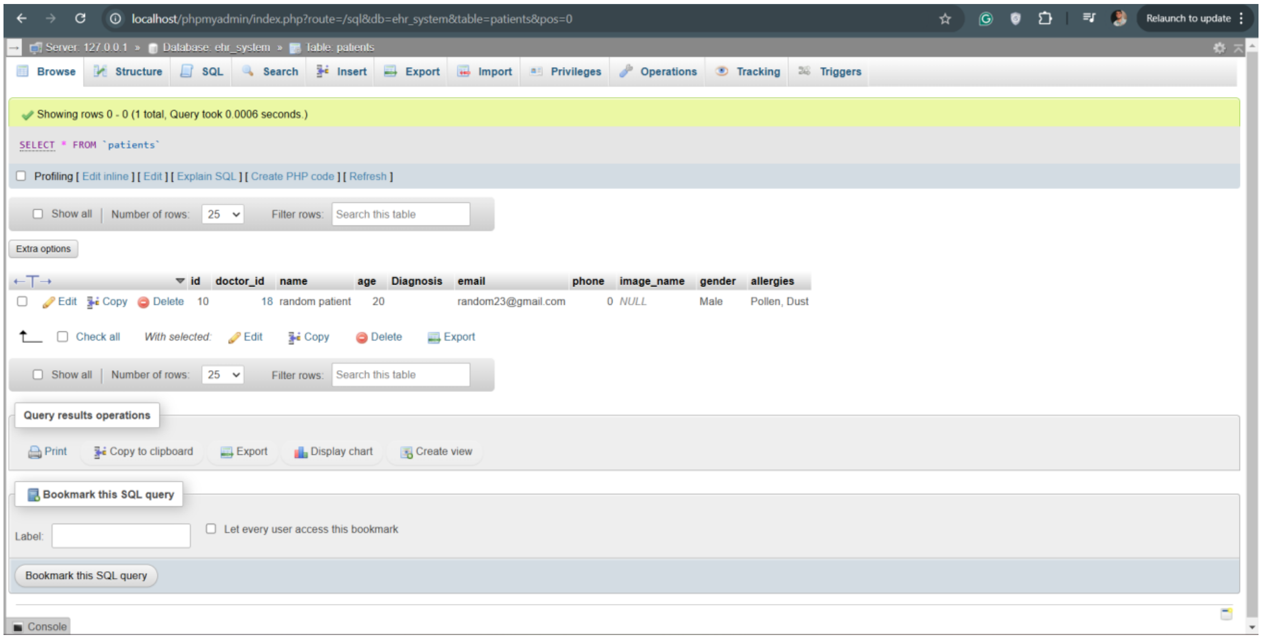
Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT * FROM `doctors`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

		id	name	email	password
□	Edit Copy Delete	11	ahmed	ahmedshahab6n@gmail.com	pbkdf2:sha256:1000000$NzoyA6yYY9YKin0o$44917667389...
□	Edit Copy Delete	15	ash	ahmedshahab@gmail.com	pbkdf2:sha256:1000000$EHX1uFbfhxlGvq78$a53a97988b5...
□	Edit Copy Delete	16	ash	ahmed@gmail.com	pbkdf2:sha256:1000000$q9ryam3RO4Ddsyql$b38dc93afe2...
□	Edit Copy Delete	17	ash	ahmed1@gmail.com	pbkdf2:sha256:1000000$DtfieNXyZLymNcop$4d1fc78a42e...
□	Edit Copy Delete	18	randomuser	random@gmail.com	pbkdf2:sha256:1000000$70iq7hACgYBndzyA$7289f15967e...
□	Edit Copy Delete	20	randomuser	random1@gmail.com	pbkdf2:sha256:1000000$Egnv95zfxtrYDvxn$cb6a35f2893...

This is what doctors table look like. They have their specific id to link patient data and passwords are hashed for privacy.

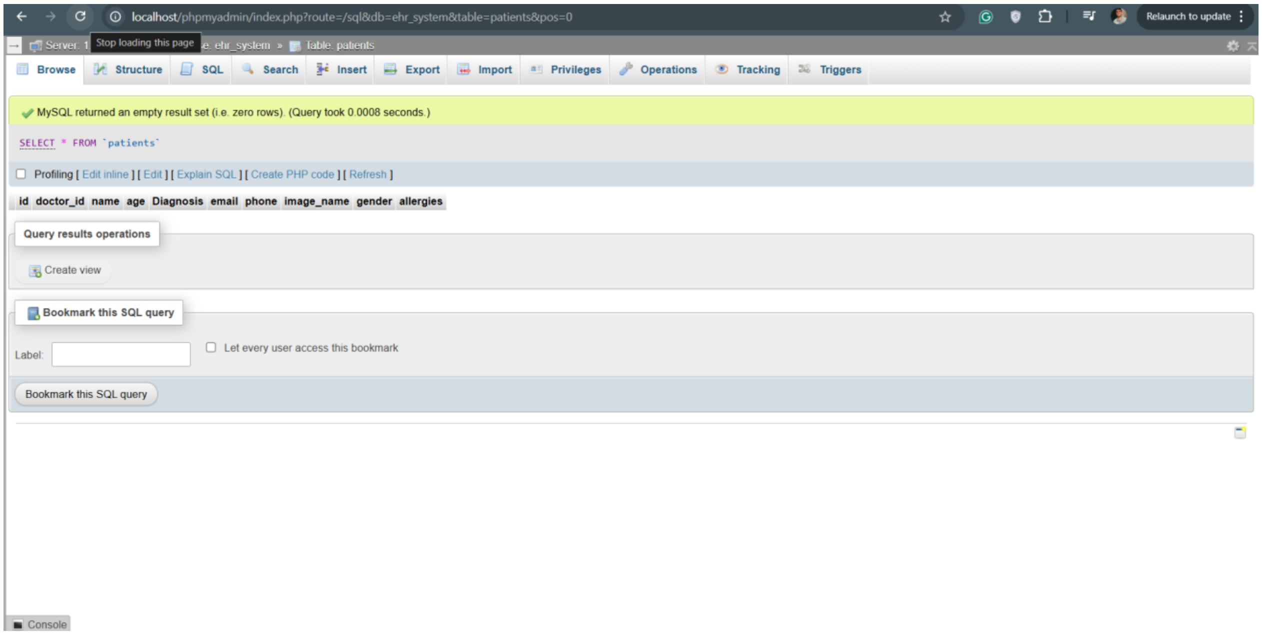


✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM `patients`
```

				id	doctor_id	name	age	Diagnosis	email	phone	image_name	gender	allergies
□	Edit	Copy	Delete	10	18	random patient	20		random23@gmail.com	0	NULL	Male	Pollen, Dust

This is what patient table looks like. This patient is associated doctor with doctor id number 10.


localhost/phpmyadmin/index.php?route=/sql&db=ehr_system&table=patients&pos=0
Relaunch to update
Server: 1
Stop loading this page
e: ehr_system » Table: patients
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Query results operations
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query
Console

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
SELECT * FROM `patients`
```

id	doctor_id	name	age	Diagnosis	email	phone	image_name	gender	allergies

This is what the table looks like when a patient is deleted from the system.

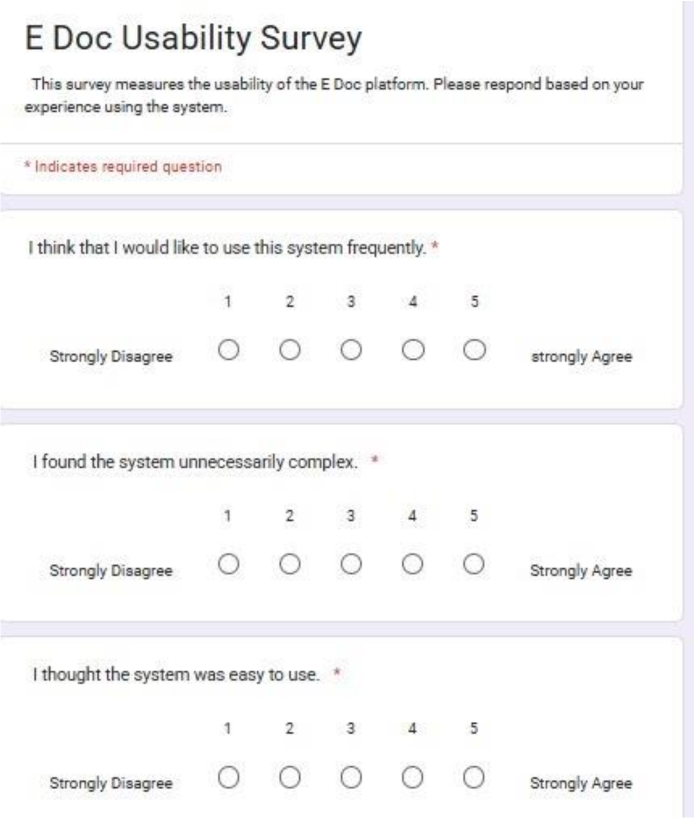
4. *Evaluation of the EHR information system*

This is the form we created for the SUS score calculation. We have calculated the score from 10 respondents.

Individual Scores:

Respondent 1: 50.0 Respondent 2: 92.5
Respondent 3: 87.5 Respondent 4: 90.0
Respondent 5: 87.5 Respondent 6: 75.0
Respondent 7: 80.0 Respondent 8: 82.5
Respondent 9: 77.5 Respondent 10: 70.0

Average SUS Score: The overall average SUS score is **79.25**, which indicates the usability of the system. This score falls in the "Good" range according to SUS interpretative standards.



E Doc Usability Survey

This survey measures the usability of the E Doc platform. Please respond based on your experience using the system.

* Indicates required question

I think that I would like to use this system frequently. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	strongly Agree

I found the system unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought the system was easy to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

5. *Conclusion*

Ayesha Hamid was responsible for building the Frontend.

Frontend Development Contributions

Her primary focus in this project was on the frontend, ensuring the user interface was aesthetically pleasing, responsive, and easy to navigate. Key contributions include:

1. User Interface Design:

- Designed the layout using HTML5, CSS3, and Bootstrap 5+, ensuring a clean and professional appearance.
- Created a consistent color scheme with shades of green, symbolizing health and vitality, enhancing the platform's visual identity.
- Developed a responsive layout to ensure compatibility across desktops, tablets, and mobile devices.

2. Core Pages:

- Login and Registration Pages: Designed secure and straightforward pages for doctors to register and log in, with proper form validation.
- About Us Page: Presented the mission and values of the E Doc platform, emphasizing usability and commitment to healthcare professionals.
- Navigation Bar: Implemented a responsive navigation menu for seamless access to all system features, including login, about, and patient record sections.

3. Enhanced User Experience:

- Integrated Bootstrap components for a polished appearance, including buttons, modals, and grids for layout consistency.
- Added a hero section with impactful text and images to welcome users, ensuring a positive first impression.
- Ensured intuitive navigation by maintaining a logical flow across all pages.

4. Optional Features:

- Added file upload functionality, enabling doctors to upload and view patient-related documents or images directly.
- Used forms to capture comprehensive patient details, including demographics, medical history, and notes.

Challenges Faced in Frontend Development

1. Responsive Design: Ensuring the interface worked flawlessly across various devices required extensive testing and adjustments.
2. Maintaining Simplicity: Balancing visual appeal with ease of use was critical, especially in healthcare-related applications where clarity is paramount.
3. Feature Integration: Adding advanced features like file uploads and dynamic date-pickers without using modern JavaScript libraries required creative solutions.
4. Color Scheme Consistency: Establishing a visually appealing and accessible green-based theme that aligned with the project’s objectives involved iterative refinements.

Muhammad Ahmed Shahab was responsible for Backend development, API development and debugging and testing.

Backend Development and Integration

The backend development for the EHR system focused on creating a robust, secure, and scalable infrastructure that serves as the foundation for managing user data, medical records, and seamless integration between various system components. Below is an outline of the work done:

Backend and Database Development

1. Backend Logic:
    - Designed and implemented the backend using PHP and Python to handle server-side logic efficiently.
    - Used Flask (Python) for creating RESTful APIs to manage routing, user authentication, and data exchanges between the client and the server.
2. Database Design:
    - Chose MySQL for its reliability and ability to handle structured data effectively.
    - Developed a well-organized schema with tables for:
        - Users: Storing data for doctors and patients.
        - Medical Records: Centralized storage of patient history, diagnoses, and treatments.
        - Authentication: Maintaining session and token data for secure logins.
    - Ensured data consistency, integrity, and normalization throughout the schema.

API Development and Integration

1. API Implementation:
    - Created RESTful APIs using Flask to handle CRUD operations for patient records, user registrations, and logins.
    - Implemented endpoints for secure data access by authorized users.
2. Authentication and Security:
    - Integrated JWT (JSON Web Tokens) for authentication, ensuring secure session management and API communication.
    - Applied input validation and error handling to prevent unauthorized access and protect sensitive user data.
3. Cross-Origin Resource Sharing (CORS):
    - Configured CORS policies to enable safe and controlled communication between the frontend and backend, especially for API calls.
4. Debugging and Testing:
    - Thoroughly tested API endpoints using tools like Postman to ensure reliability and proper functionality.
    - Implemented debug logging to identify and fix issues during development.

Integration and Deployment

1. Integration:
   - Successfully integrated the backend with the frontend, ensuring a seamless connection for user interactions and data management.
   - Used Python scripts to automate backend operations and streamline workflows.
2. Deployment:
   - Prepared the backend for deployment on a local environment, ensuring easy replication for other developers or instructors.
   - Documented the deployment process to facilitate smooth setup on other systems.

Challenges and Solutions

- Challenge: Ensuring data security and preventing unauthorized access.
  - Solution: Implemented robust authentication mechanisms and enforced strict validation.
- Challenge: Synchronizing the backend with the frontend during development.
  - Solution: Regular testing and iterative adjustments to maintain seamless integration.

This backend development forms the backbone of the EHR system, ensuring secure and efficient handling of user data and enabling smooth communication between all components.