Major Project - I

Title: Cloud Based Emergency Health Record Access system

Presented by:
R2142211228 – Afreen Ali
R2142211380 - Manvita Goel
R2142211206 – Harsh Verma

Mentored By
Dr. Shreshth Gupta

Content

- Introduction
- Literature Review
- Objectives
- Methodology
- Working Model
- Results
- Conclusion
- References

Colour Code:
Synopsis: Red
Mid term: Orange
End Term: Purple
Note: The Following presentation will have all the content of the later.

Introduction

The challenges of traditional health record systems, particularly in emergency situations where immediate access to patient data is crucial. These systems can be slow, especially when patients are treated outside their usual healthcare network, potentially leading to delays, medical errors, and poor outcomes.

To address this issue, we present the **Cloud-Based Emergency Health Record Access System**. This innovative solution leverages the power of cloud technology to ensure healthcare providers to access vital patient information, such as allergies, medications, and medical history, in real-time from anywhere. This system enhances patient safety, improves healthcare outcomes, and ensures that critical information is available when needed, regardless of location or network barriers.

Technologies Used

Django

A powerful Python web framework used for backend development, providing a robust and scalable foundation for the EHR system.

React JS

A JavaScript framework for building building user interfaces, used to develop the frontend of the application, providing a responsive responsive and interactive user experience.

PostgreSQL

A reliable and efficient open-source relational database used for for storing patient and system data data securely.

RESTful APIs

The system utilizes RESTful APIs to to establish communication between the frontend and backend, allowing seamless data exchange.

Installation and Setup

1

Clone the Repository

Clone the repository using the following command:

```
git clone git@github.com:HukumaBob/emr.git
```

2

Install Dependencies

Install the necessary dependencies for both the backend and frontend.

3

Database Setup

Create a PostgreSQL database for the project and update the database configuration in the backend settings file.

4

Run Migrations

Run the migrations to create the database tables.

5

Start Development Servers

Start the backend and frontend development servers.

6

Access the EHR System

Open your web browser and visit http://localhost:3000/ to access the EHR system.

Purpose of the Project:

- Improve accessibility and availability of critical patient health records during emergencies.
- Centralize patient data in a secure, interoperable cloud platform.
- Provide authorized medical personnel with real-time access to critical information (e.g., allergies, medications, medical history).
- Ensure real-time access even in urgent situations.
- Enhance the quality and speed of care during emergencies.
- Minimize medical errors.
- Potentially save lives during emergencies.

Problem Statement:

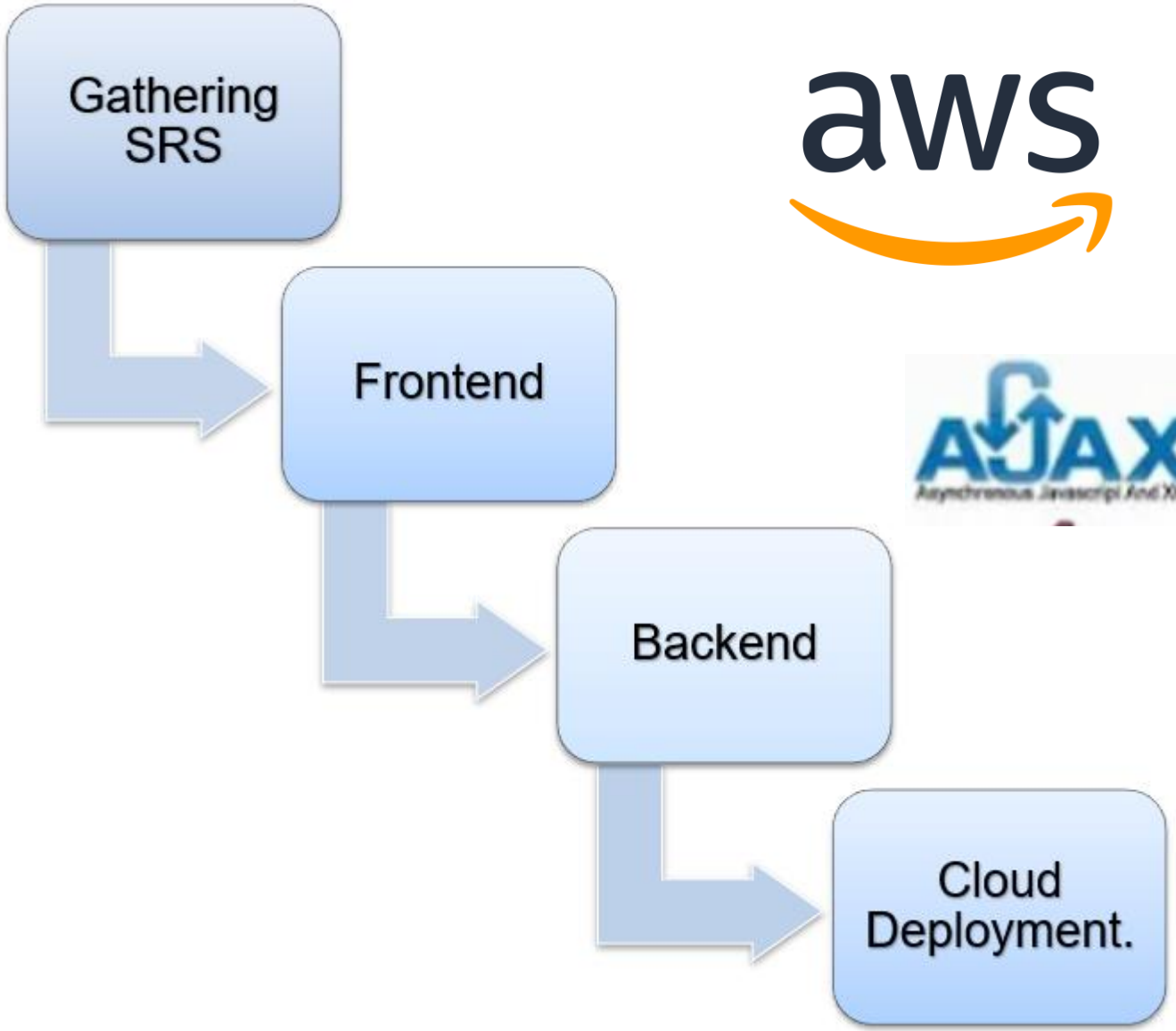
- Lack of immediate access to accurate and comprehensive patient health records in emergencies.
- Results in delayed treatment, medical errors, and adverse outcomes.
- Traditional health record systems are fragmented and confined to specific healthcare networks.
- Difficult for providers to obtain vital information (allergies, medications, medical history) when treating patients outside their regular care settings.
- This inefficiency is critical during emergencies where every second counts.
- Need for a secure, cloud-based system for quick and reliable access to patient data.
- Ensures authorized medical personnel can access essential patient data regardless of location.
- Aims to improve care quality and patient outcomes in urgent situations.

Area of application:

This system is primarily applicable in the healthcare sector, specifically:

- **Emergency Departments**: To provide real-time access to critical patient data.
- **Paramedics and Ambulance Services**: To access patient records and route to hospitals.
- **Healthcare Institutions**: Hospitals, clinics, and urgent care centers for integrated and efficient patient care.
- **Public Health Systems**: For improving the coordination and efficiency of emergency medical services.

WebDevelopment –Roadmap &Tech Stack:



Literature Review

Author(s)	Year	Title	Key Focus	Findings	Relevance to Cloud-Based Emergency Health Record Access
Smith et al.	2020	"Improving Emergency Medical Response with Cloud Technologies"	Application of cloud computing in emergency medical services (EMS)	Demonstrated improved response times and better decision-making using cloud-based data systems	Highlights the speed and availability benefits of cloud-based EHR access during emergencies
Johnson et al.	2021	"Interoperability in Cloud-Based Healthcare Systems"	Investigated interoperability between cloud platforms and healthcare systems	Cloud systems improved the seamless exchange of data between different healthcare providers	Shows how cloud systems enhance EHR accessibility across multiple healthcare facilities in emergencies

SWOT Analysis

Strengths	Weaknesses	Opportunities	Threats
Improved Emergency Care: Ensures critical health information is available during emergencies.	**Data Integration Challenges:** Complexity in integrating data from different EHR systems.	**Global Market Expansion:** Adaptation for use in different countries.	**Cybersecurity Risks:** Vulnerability to cyber-attacks.
Accessibility: Provides access to patient records anywhere, enhancing continuity of care.	**Dependency on Internet:** Limited effectiveness in areas with poor internet connectivity.	**Integration with Emerging Technologies:** Potential to incorporate AI for predictive analytics.	**Regulatory Changes:** Impact of changing healthcare regulations.
Security: Strong encryption and access controls protect patient data.	**High Initial Costs:** Significant investment required for development and deployment.	**Patient Empowerment:** Allowing patients controlled access to their records.	**Resistance to Change:** Potential reluctance from healthcare providers to adopt a new system.
Interoperability: Integration with existing EHR systems.	**Privacy Concerns:** Potential concerns over the security of cloud-stored health information.	**Partnerships:** Collaborations with healthcare providers for further system development.	**Technical Failures:** Risks of downtime or technical issues hindering access during emergencies.

Objective

Main Objective:

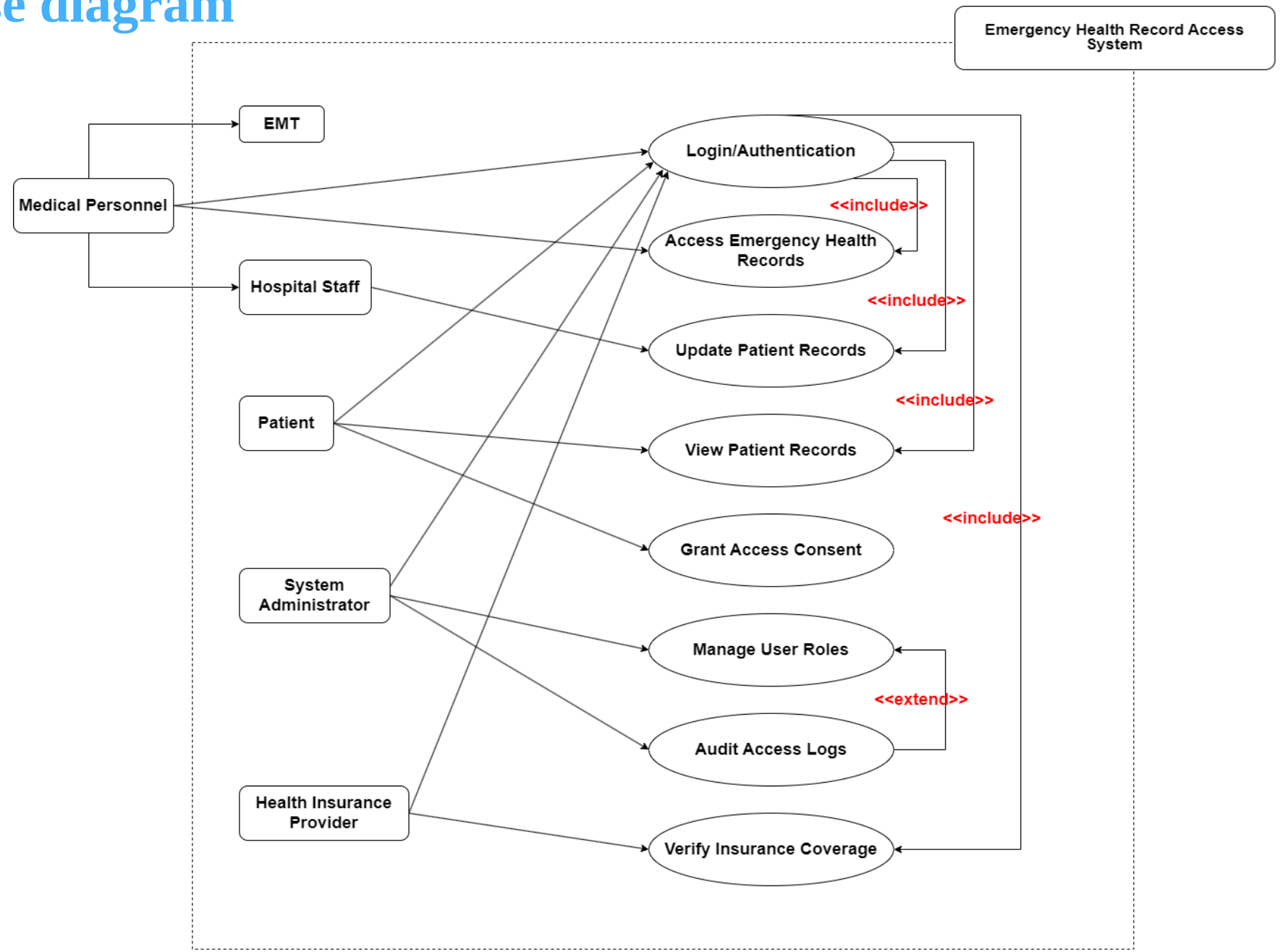
- **Enhance Emergency Care Efficiency**: Develop a cloud-based platform that enables healthcare providers to access critical patient health records in real-time during emergencies, improving the speed and quality of care.

- **Ensure Secure Data Access:** Implement robust security measures to ensure that patient data is accessible only to authorized personnel, safeguarding sensitive information from unauthorized access.

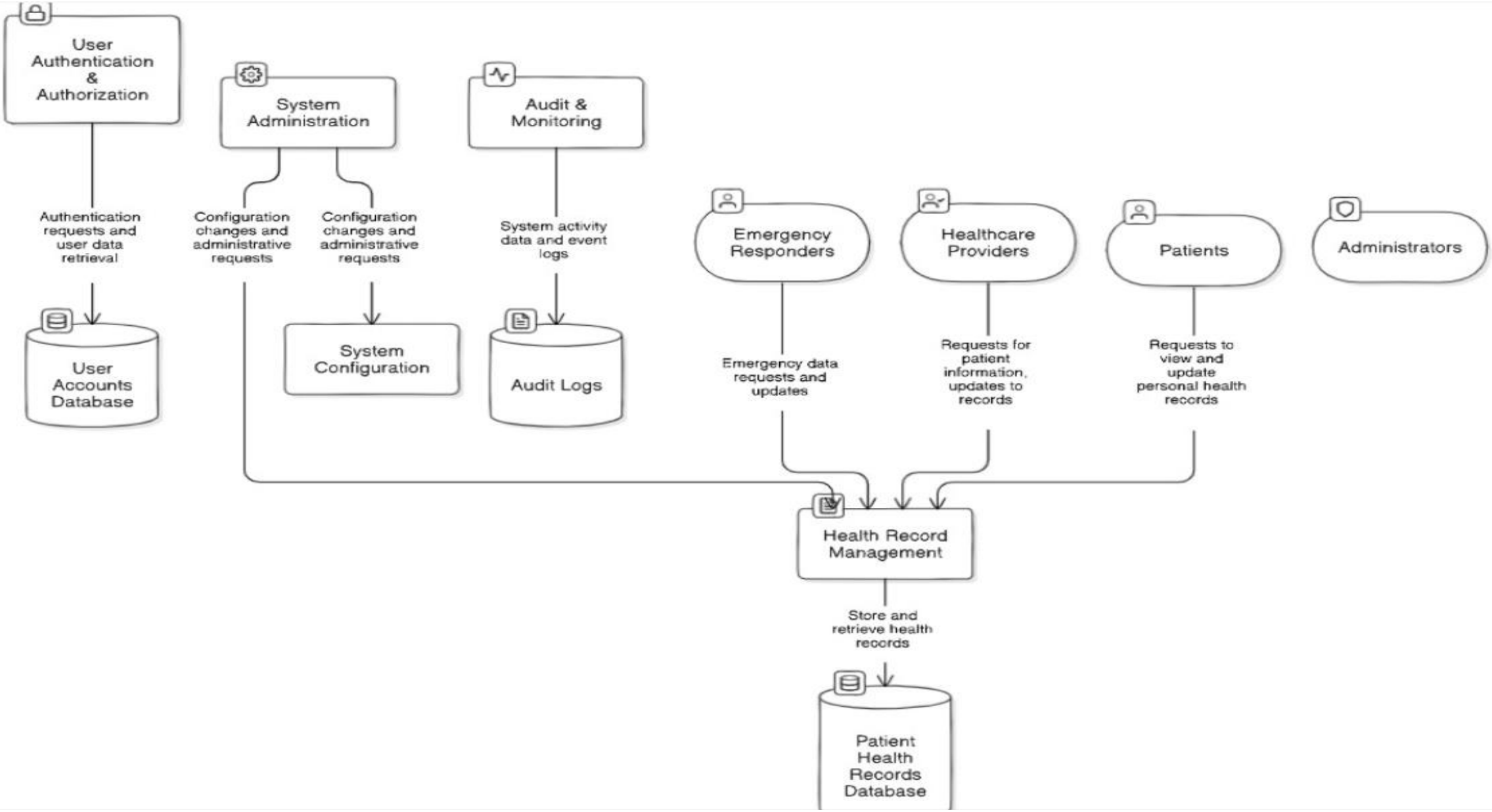
- **Facilitate Interoperability**: Design the system to integrate seamlessly with existing Electronic Health Record (EHR) systems, ensuring compatibility and smooth data exchange across different healthcare networks.

- **Improve Patient Outcomes**: By providing timely and accurate health information to emergency responders, the system aims to reduce medical errors and enhance patient safety, ultimately improving health outcomes in critical situations.

Use Case diagram



Methodology (DFD)



Methodology

Steps:

- System Design and Architecture Development: Design a scalable and secure cloud-based platform with a user-friendly interface for healthcare providers.

- Integration with Existing EHR Systems: Implement APIs to connect and centralize patient data from various EHR systems into the cloud platform.

- Security and Privacy Implementation: Incorporate encryption, access controls, and regulatory compliance to protect patient data.

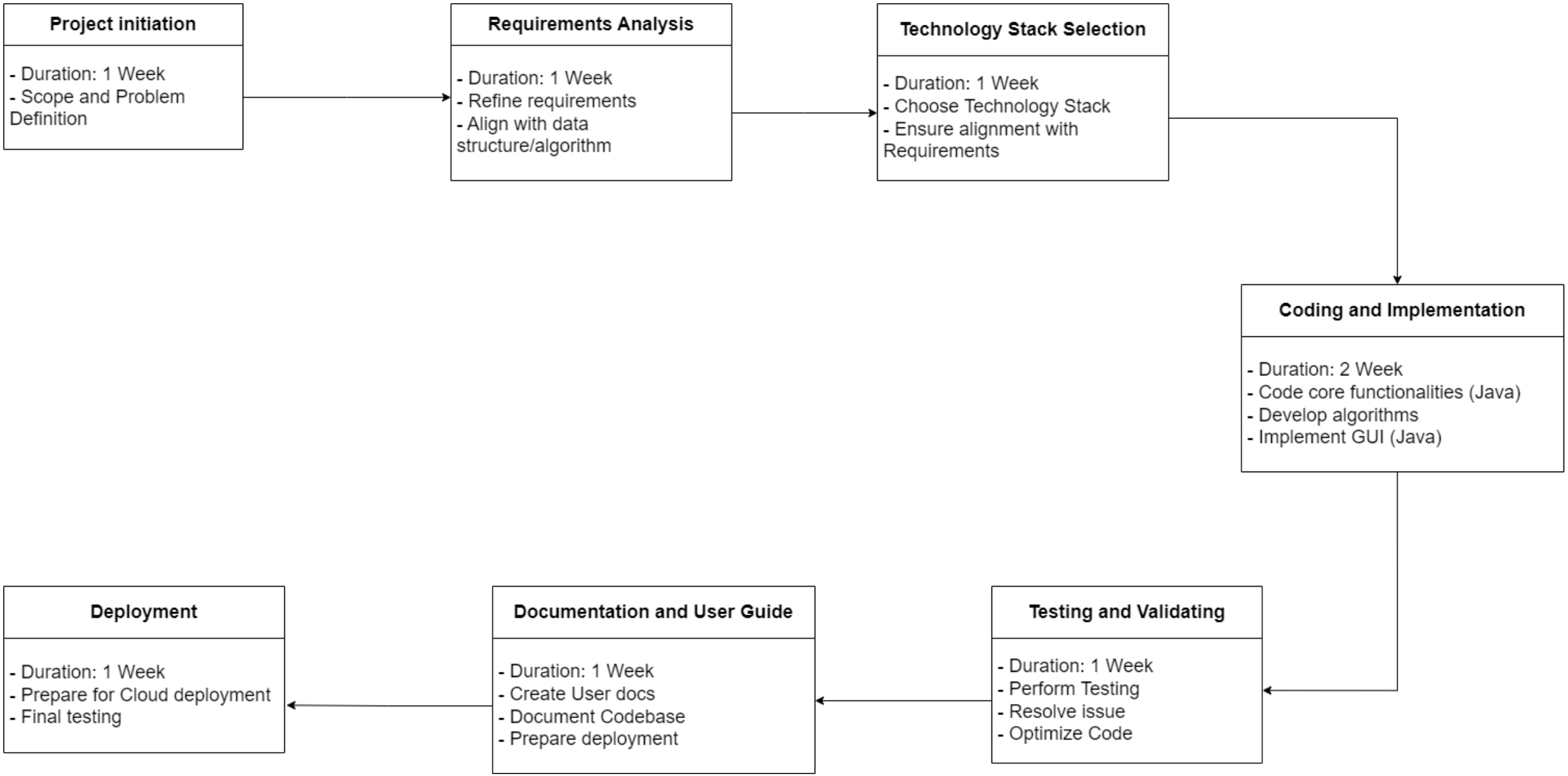
- Prototype Development and Testing: Build and rigorously test a prototype to ensure system reliability and gather user feedback.

- Deployment and Training: Deploy the system and provide training to healthcare providers for effective use during emergencies.

- Ongoing Maintenance and Updates: Establish continuous monitoring and updates to maintain system performance and security compliance.

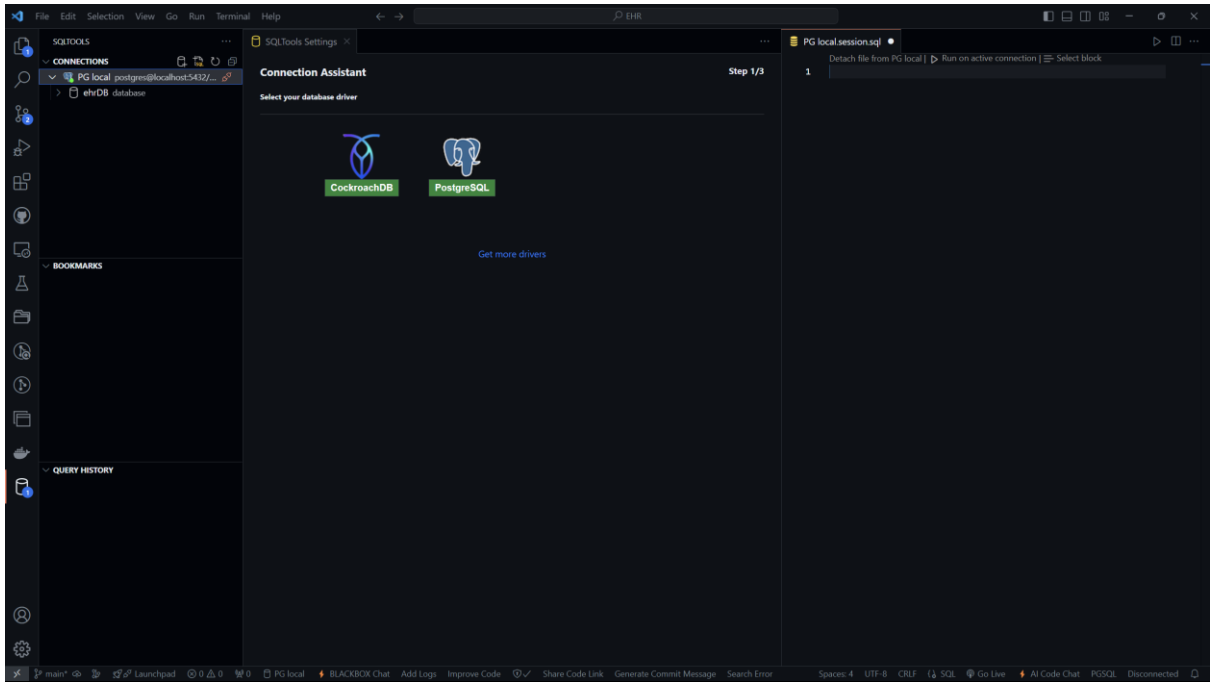
Timeline.

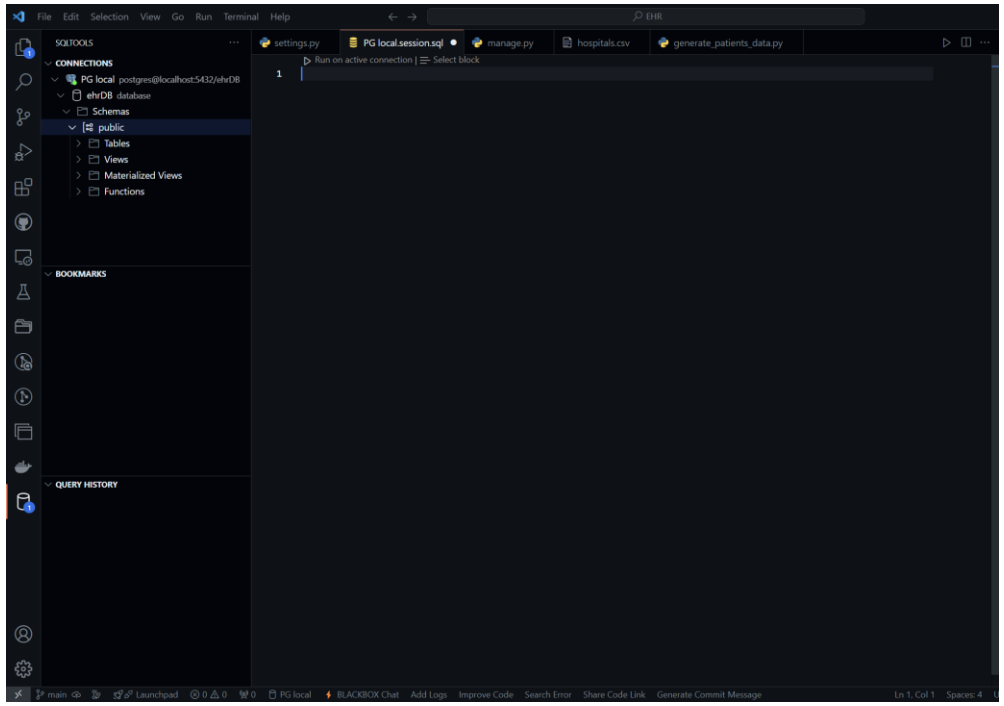
Pert Chart :


Project initiation
- Duration: 1 Week
- Scope and Problem Definition
Requirements Analysis
- Duration: 1 Week
- Refine requirements
- Align with data structure/algorithm
Technology Stack Selection
- Duration: 1 Week
- Choose Technology Stack
- Ensure alignment with Requirements
Coding and Implementation
- Duration: 2 Week
- Code core functionalities (Java)
- Develop algorithms
- Implement GUI (Java)
Testing and Validating
- Duration: 1 Week
- Perform Testing
- Resolve issue
- Optimize Code
Documentation and User Guide
- Duration: 1 Week
- Create User docs
- Document Codebase
- Prepare deployment
Deployment
- Duration: 1 Week
- Prepare for Cloud deployment
- Final testing

Results:

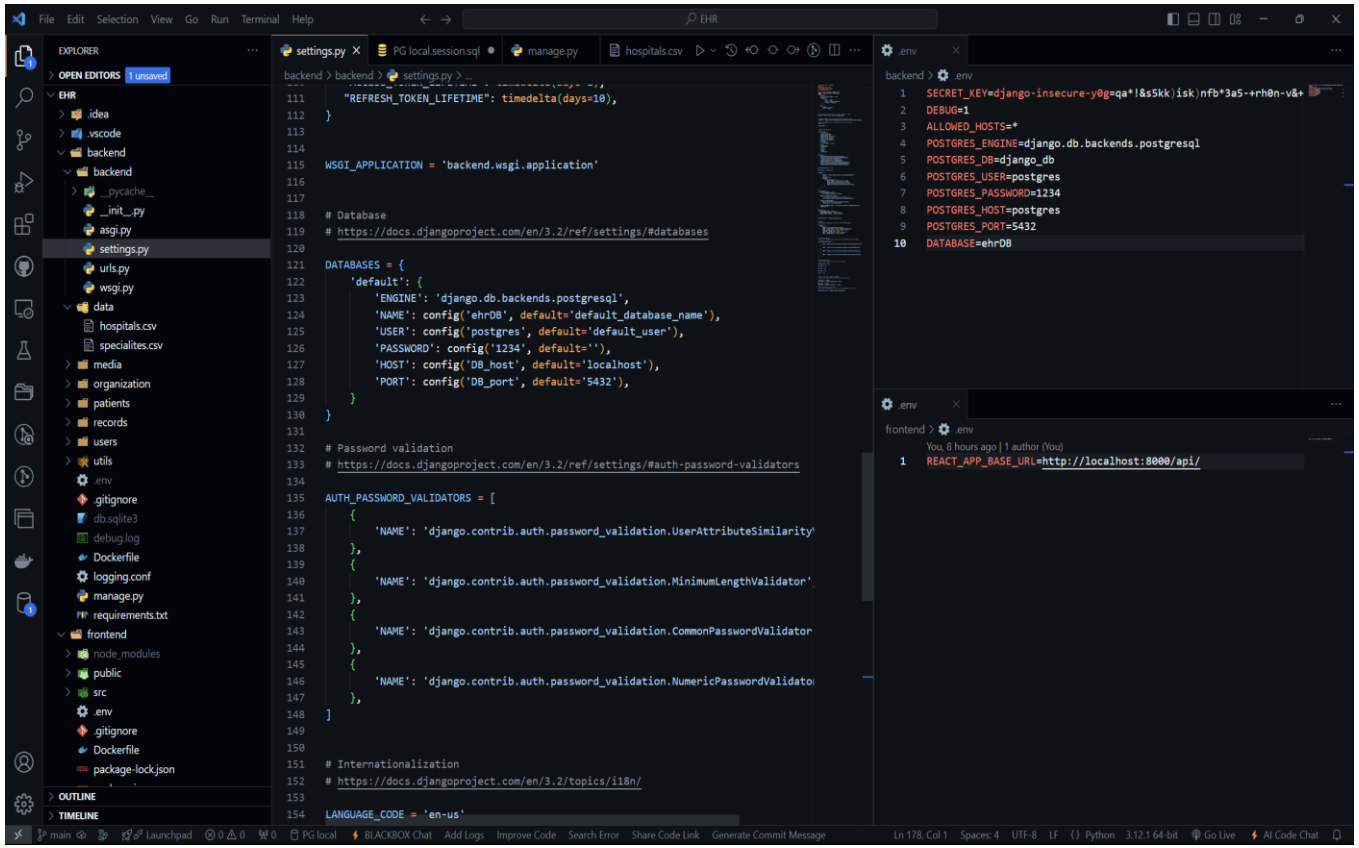
Database connection:





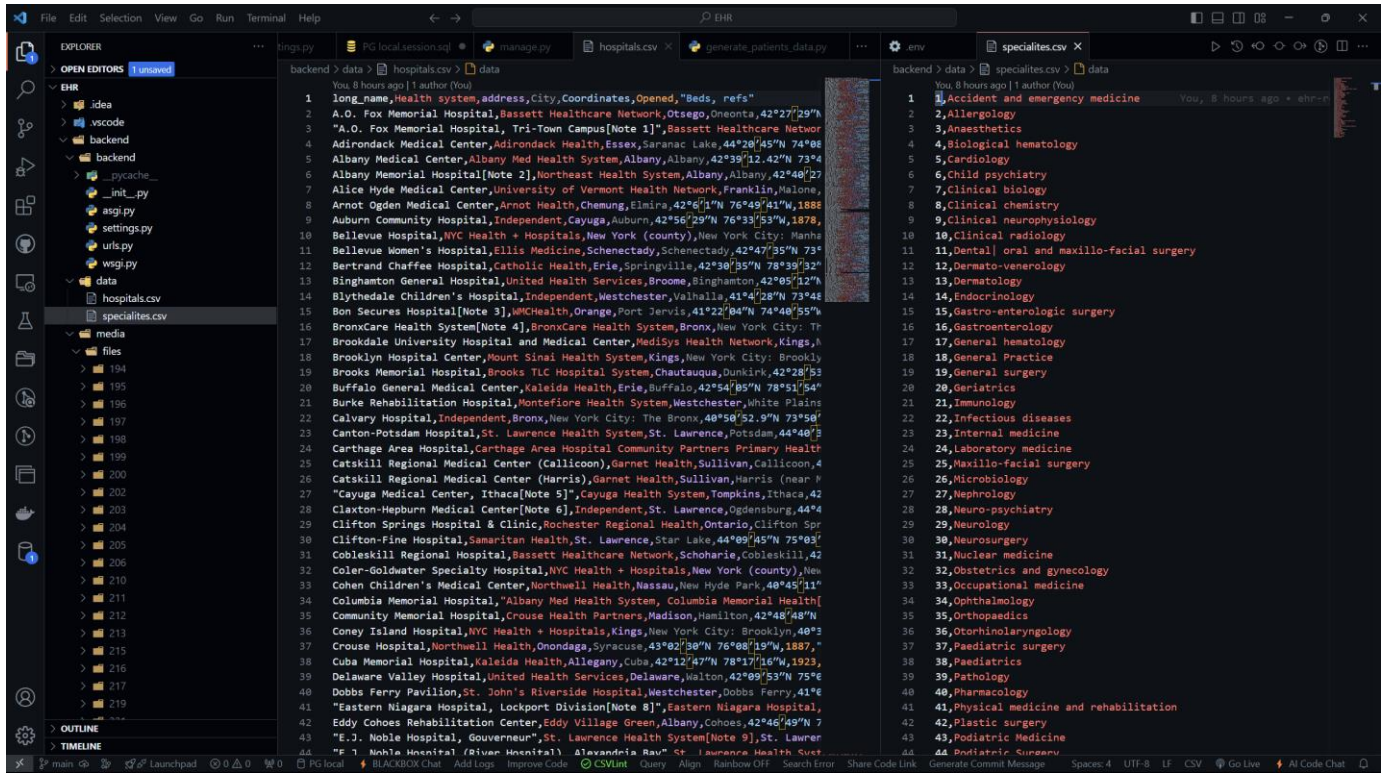
Results:

Database connection details: env file for frontend and backend and connection details in settings.py



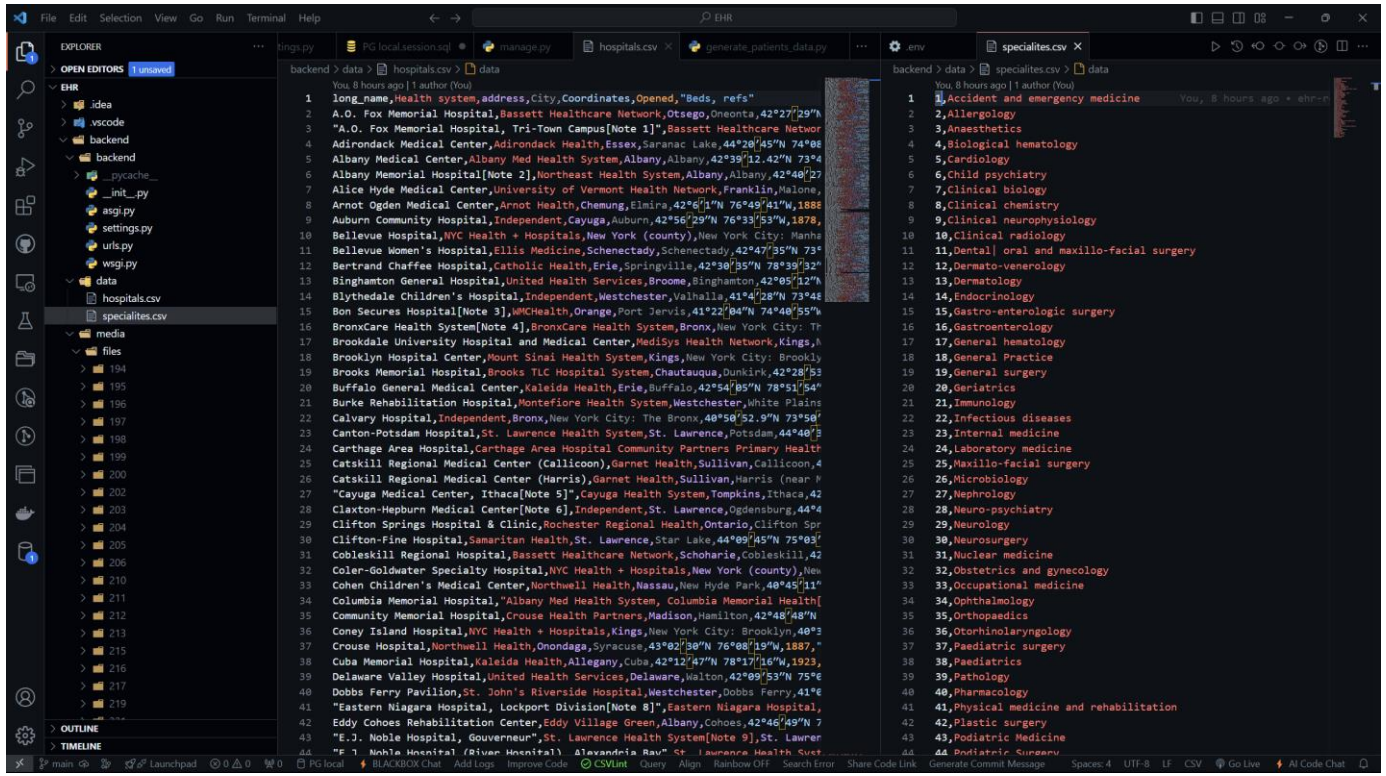
Results:

Data: hospital.csv and specialities.csv



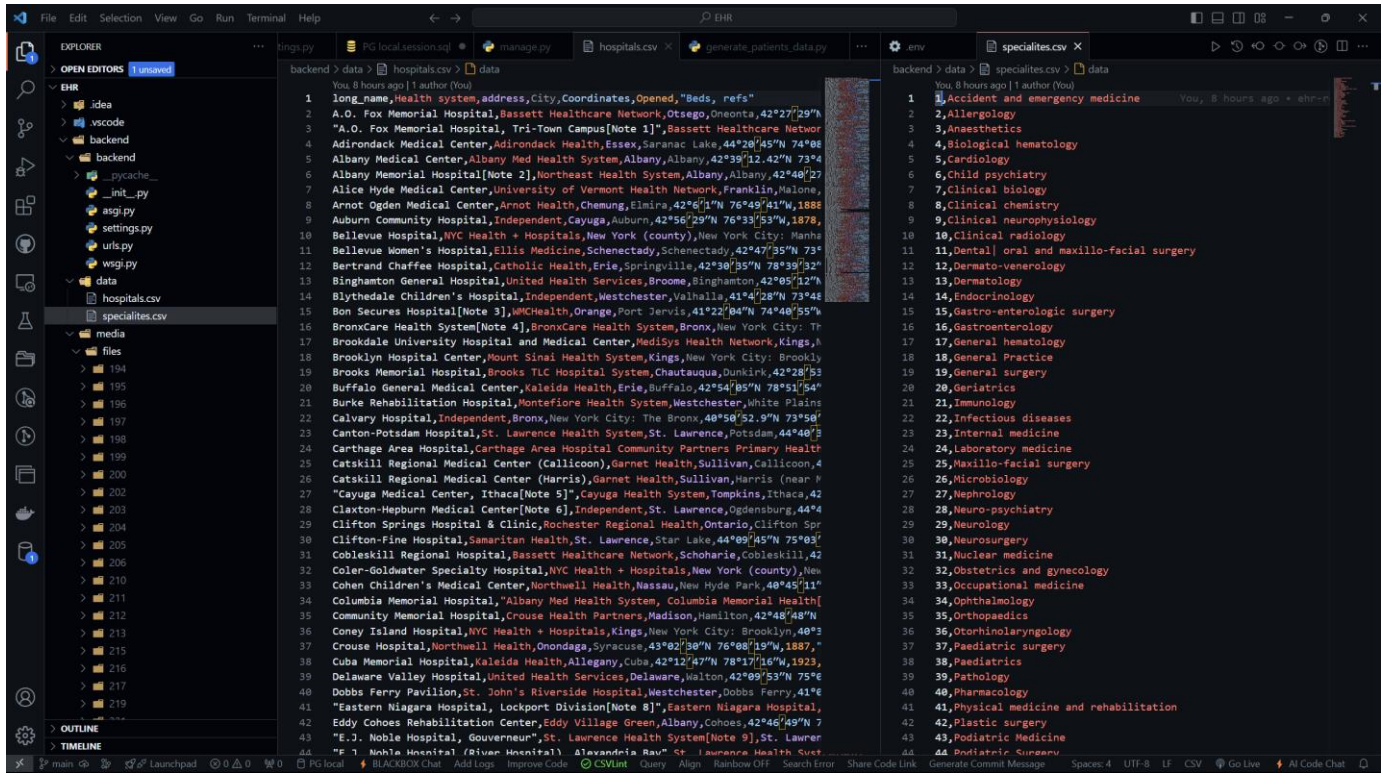
Results:

Backend admin.py for admin and patient:



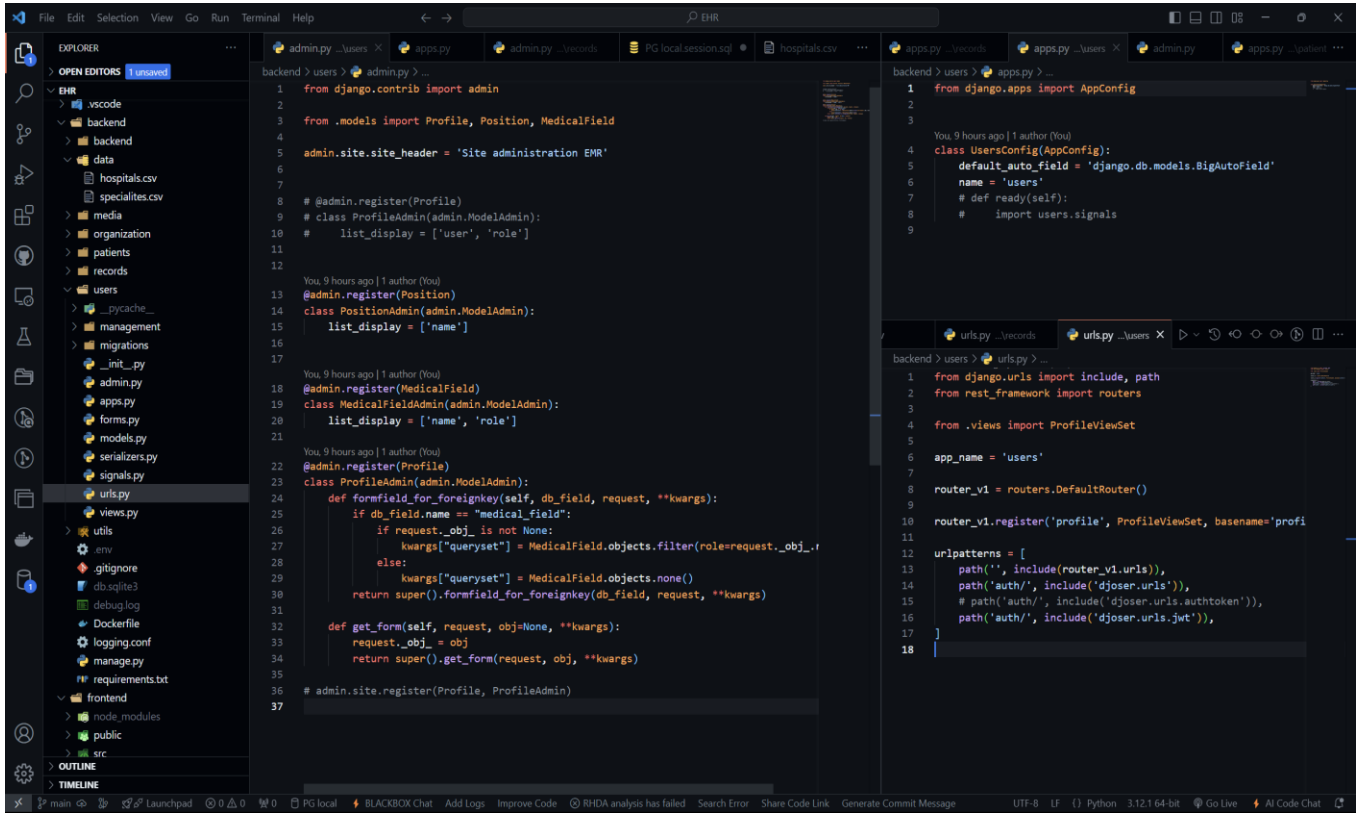
Results:

Backend : Records admin.py



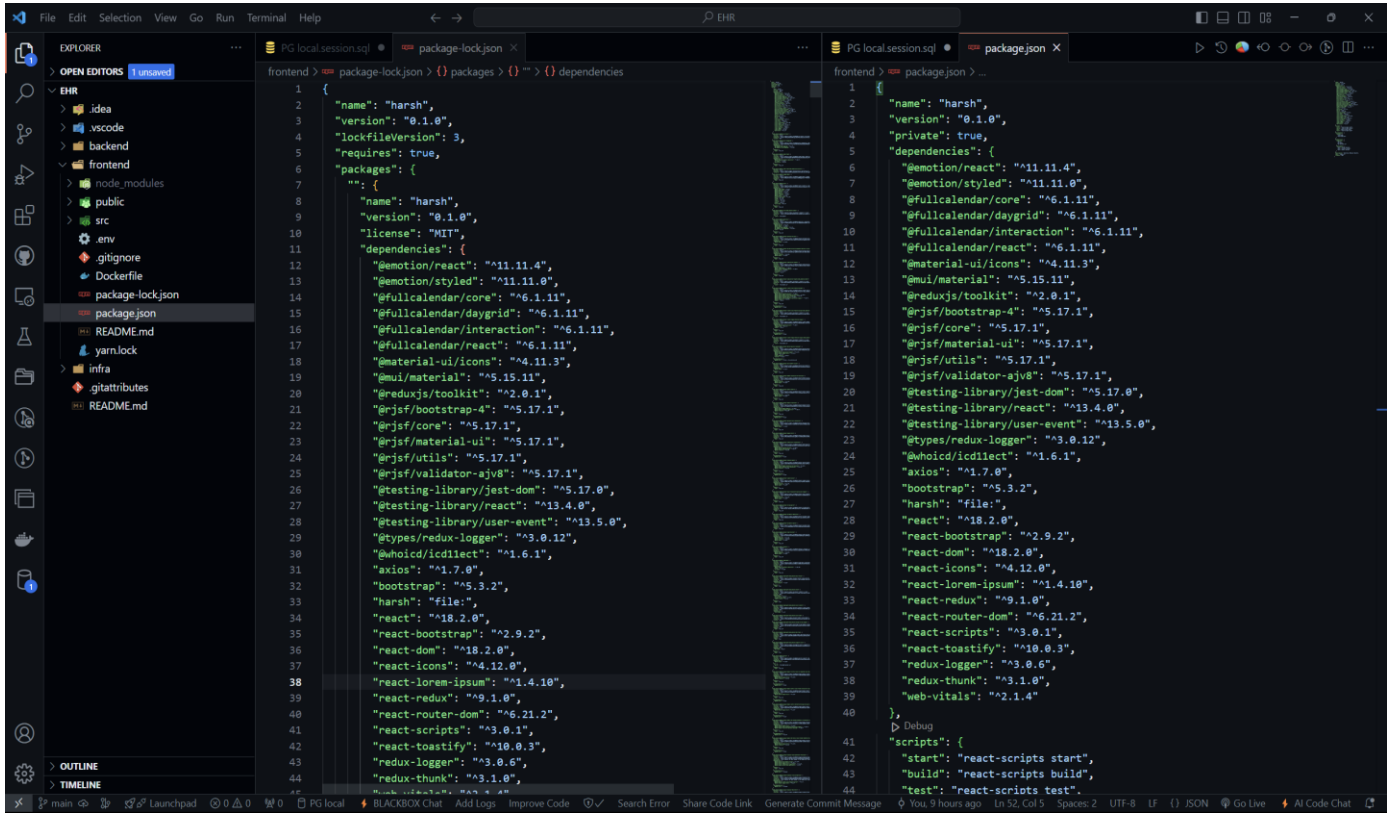
Results:

Backend: Users admin.py



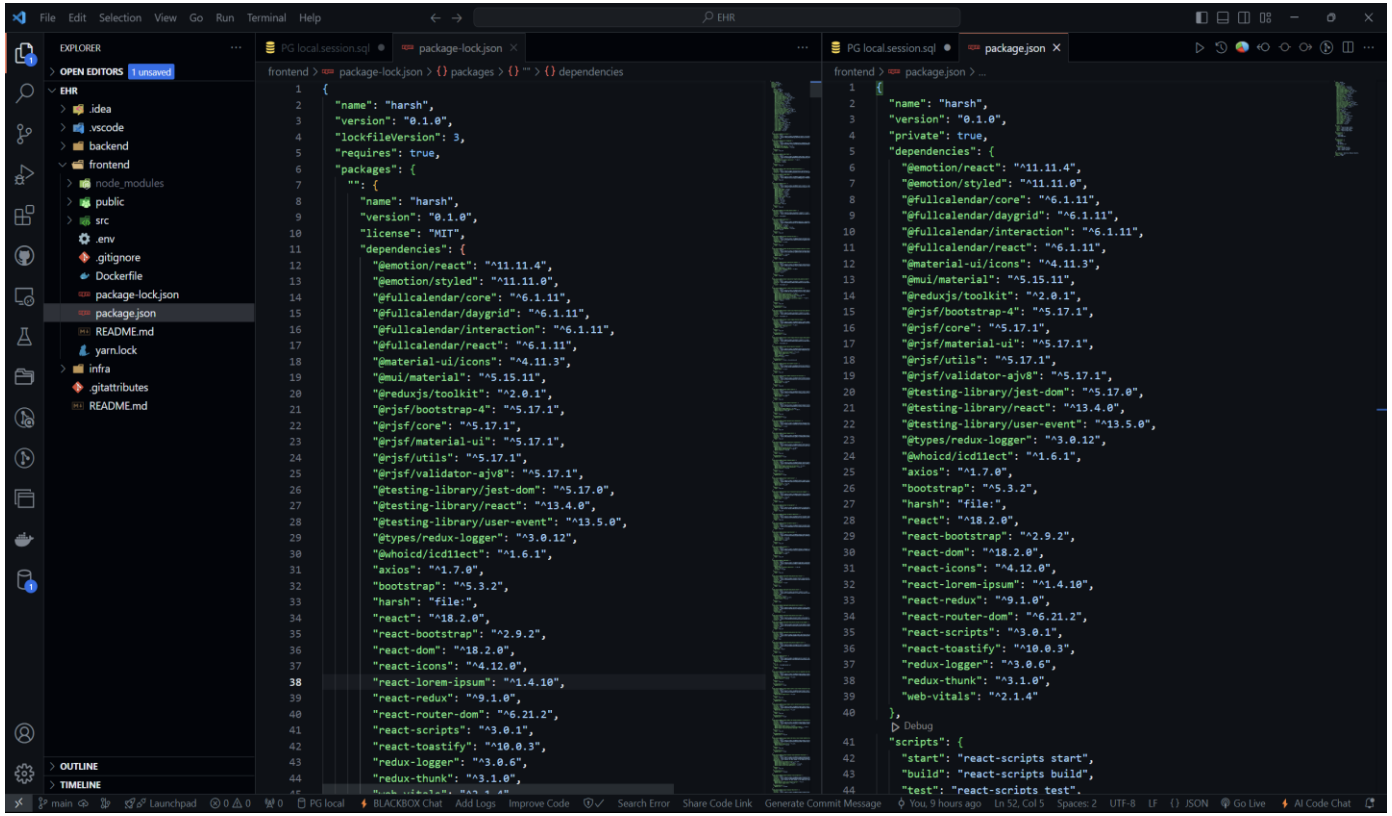
Results:

Frontend part: package json files



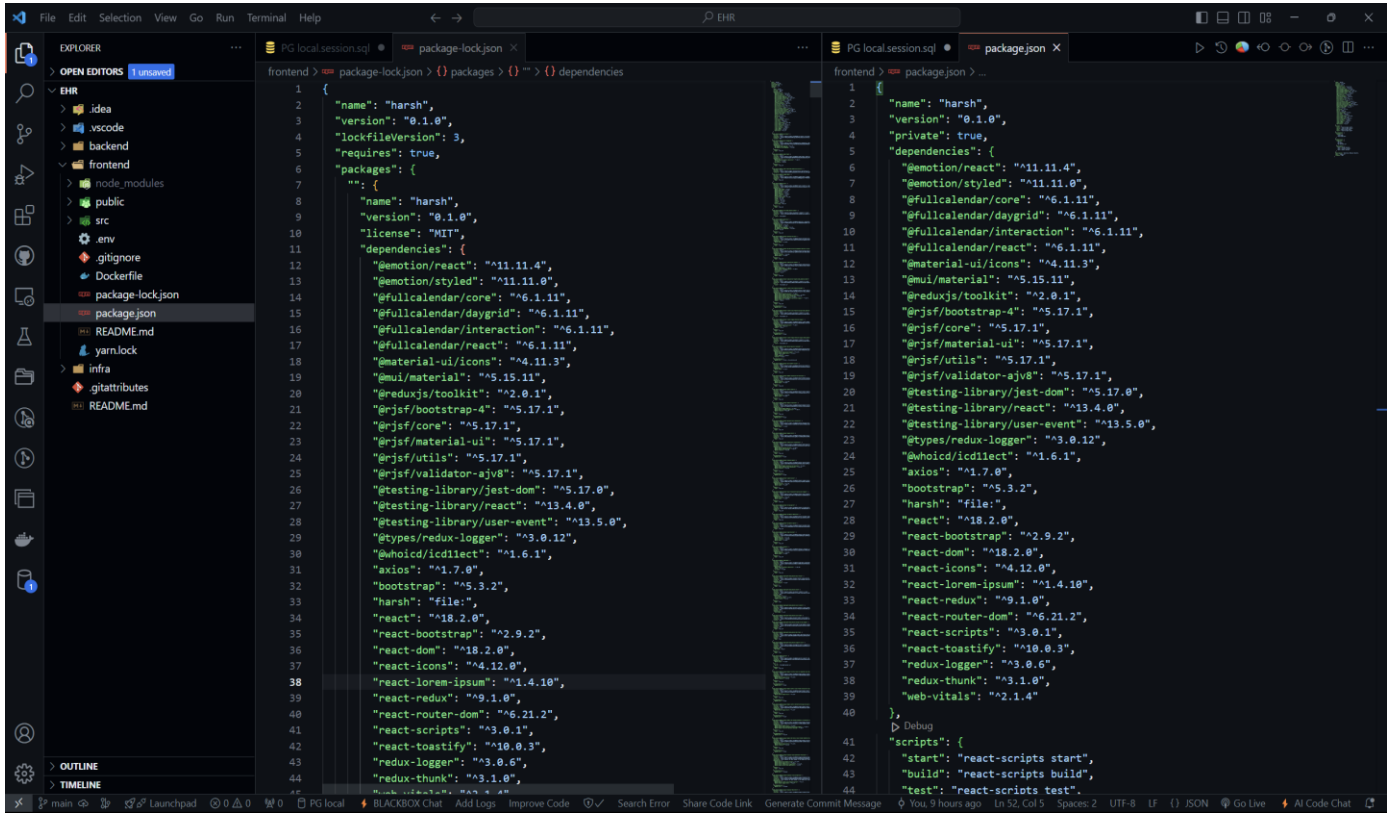
Results:

Frontend part: app.js and api configs and nginx file configs



Results:

Features and tech stack used:



Conclusions:

The **Cloud-Based Emergency Health Record Access System** addresses critical challenges in accessing patient health records during emergencies, ensuring real-time availability of vital medical information such as allergies, medications, and medical history. This innovative solution leverages cloud technology to overcome the limitations of traditional systems, offering significant improvements in patient care by enhancing speed, accuracy, and accessibility.

Through the integration of robust technologies like Django, React JS, PostgreSQL, and RESTful APIs, the system ensures a scalable, secure, and interoperable platform. By focusing on secure data access, seamless interoperability, and user-centric design, this platform is poised to transform emergency healthcare services, minimizing errors, reducing treatment delays, and potentially saving lives.

The implementation and testing phases have demonstrated the system's reliability and effectiveness, validating its potential to improve healthcare outcomes. Moving forward, its adoption across emergency departments, paramedic services, and public health systems can revolutionize the management of critical patient information, ensuring timely and accurate medical interventions during emergencies.

Reference

B: List of cited papers :

[1] : Smith, J., & Doe, A. (2020). The Impact of Electronic Health Records on Emergency Care: A Case Study. Journal of Emergency Medicine, 30(4), 123-130.

[2] : U.S. Department of Health and Human Services. (1996). Health Insurance Portability and Accountability Act (HIPAA). Retrieved from hhs.gov.

[3] Johnson, R., & Lee, M. (2021). Interoperability in Healthcare: Challenges and Opportunities. Health Informatics Journal, 27(3), 456-467.

[4]. Chen, L., & Zhao, Y. (2021). Integrating heterogeneous health data using cloud computing: A case study. Journal of Biomedical Informatics, 112, 103602.

[5]. Garcia, M., & Roberts, P. (2017). Real-time access to patient data during emergencies: A cloud-based approach. Journal of Emergency Medicine, 45(3), 225-232.

Thank You