REACT-ENABLED ANALYTICS FOR HEALTHCARE OPTIMIZATION

PROJECT REPORT

Submitted by

HARESH M	**2116210701066**
HAYAGREEVAN V	**2116210701080**
JEYAPRIYAN M	**2116210701097**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled **"REACT-ENABLED ANALYTICS FOR HEALTHCARE OPTIMIZATION"** is the bonafide work of **"HARESH M (2116210701066), HAYAGREEVAN V (2116210701080), JEYAPRIYAN M (2116210701097)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. K. Anand M.E, Ph.D,

PROJECT COORDINATOR

Professor

Department of Computer Science and Engineering,

Rajalakshmi Engineering College,

Thandalam,

Chennai - 602105

Submitted to Mini Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

This project introduces an advanced analytics platform empowered by React JS to revolutionize healthcare optimization. By integrating real-time data collection, machine learning algorithms, and interactive user interfaces, this project provides healthcare professionals with actionable insights and predictive analytics. Making well-informed decisions, allocating resources, and actively managing risks will all be made possible by these insights. This technology will improve collaboration between healthcare stakeholders and expedite procedures by connecting with current technologies. The continuous improvement of operating procedures will be made easier by constant data monitoring and analysis. Working together with stakeholders guarantees the creation of a scalable analytics platform that can be adjusted to meet changing healthcare requirements. Overall, this project represents a significant advancement in healthcare technology, aiming to drive efficiency and effectiveness in medical practices and institutions.

ACKNOWLEDGEMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.**, for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.**, for providing us with the requisite infrastructure.We also express our sincere gratitude to our college Principal, **Dr. S. N. Murugesan M.E., PhD.**, and **Dr. P. KUMAR M.E., PhD**, Director computing and information science , and Head Of Department of Computer Science and Engineering and our project coordinator **Dr. K. Anand M.E.,Ph.D.**, for his encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staff for their direct and indirect involvement in successful completion of the project for their encouragement and support.

HARESH M

HAYAGREEVAN V

JEYAPRIYAN M

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATIONS	ACRONYMS
KPI	KEY PERFORMANCE INDICATOR
HTML	HYPERTEXT MARKUP LANGUAGE
CSS	CASCADING STYLE SHEETS
JS	JAVASCRIPT
DB	DATABASE
SQL	STRUCTURED QUERY LANGUAGE
EHR	ELECTRONIC HEALTH RECORDS

CHAPTER 1

INTRODUCTION

The healthcare industry is undergoing a profound transformation driven by advancements in technology and the growing need for efficient, data-driven decision-making processes. In this dynamic landscape, this project emerges as a pivotal solution designed to address the multifaceted challenges faced by the healthcare industry. By harnessing the power of React-enabled analytics, this project aims to optimize healthcare delivery, improve patient outcomes, precisive diagnostics and enhance operational efficiency.

This project leverages state-of-the-art data analytics and visualization tools to provide healthcare professionals with real-time insights and predictive analytics. This platform integrates seamlessly with existing healthcare systems, offering a comprehensive suite of features that include real-time data collection, machine learning algorithms, and interactive user interfaces. The result is a robust, user-friendly system that empowers medical staff to make informed decisions, personalize patient care, and proactively manage health services.

The core of this project lies in its ability to process and analyze vast amounts of healthcare data swiftly and accurately. Through its React-enabled architecture, the platform ensures a responsive and intuitive user experience, allowing healthcare professionals to interact with complex datasets effortlessly. This interactivity fosters a collaborative environment, enabling medical teams to work together more effectively and efficiently.

1.1 PROBLEM STATEMENT

The healthcare industry is confronted with several critical challenges that impede its ability to deliver optimal care. These challenges include inefficient data management, lack of real-time insights, fragmented patient information, and insufficient tools for predictive analytics. These issues often result in delayed decision-making, suboptimal patient outcomes, and increased operational costs.

Traditional healthcare systems struggle to integrate and analyze the vast amounts of data generated daily, ranging from patient records and diagnostic results to treatment plans and administrative processes. The lack of a cohesive and

responsive analytics platform hampers the ability of healthcare professionals to access and interpret this data quickly and accurately. Consequently, healthcare providers are often forced to rely on incomplete information, leading to inefficiencies and errors in patient care.

This project addresses these critical issues by offering a React-enabled analytics platform designed specifically for healthcare optimization. By integrating real-time data collection, informative data analysis and interactive user interfaces. This project aims to transform the way healthcare professionals manage and utilize data, ultimately enhancing patient care and operational efficiency.

1.2 SCOPE OF THE WORK

Our project aims to improve patient care and hospital efficiency through better data analysis. We'll focus on finding useful insights from different types of healthcare data, like patient records and medical images. We'll also look into collaborating with hospitals and fitting our solutions into their existing systems. The scope of work includes the following key components:

Requirement Analysis and Planning: Conducting thorough consultations with healthcare providers, administrators, and IT staff to understand specific needs and challenges. Defining project goals, objectives, and key performance indicators (KPIs). Developing a detailed project plan outlining timelines, milestones, and resource allocation.

Platform Development: Designing and developing the core architecture of this platform using React JS, ensuring scalability, flexibility, and security. Integrating real-time data collection modules to aggregate data from various sources, including electronic health records (EHR), diagnostic systems, and administrative databases. Implementing machine learning algorithms to analyze data and generate predictive insights. Developing interactive user interfaces that facilitate easy navigation, data visualization, and user engagement.

Data Integration and Management: Ensuring seamless integration with existing healthcare systems and databases. Implementation of robust data management protocols to ensure data accuracy, integrity, and security.

Developing APIs for data exchange and interoperability with third-party applications.

Evaluation and Optimization: Continuously assessing the platform's impact on healthcare optimization through predefined KPIs and user feedback. Conducting regular reviews and audits to identify areas for improvement. Adapting and refining the platform based on evolving healthcare needs and technological advancements.

1.3 AIM AND OBJECTIVES OF THE PROJECT

Our main goal is to make healthcare better by using advanced data techniques. We want to help hospitals work smoother, make better decisions, and manage risks. Specifically, we'll develop smart algorithms, integrate them into hospital systems, and keep improving them based on feedback. This project enhances healthcare optimization through real-time data insights, predictive analytics, and user-friendly interfaces. The platform will be scalable and secure, integrating real-time data from various healthcare sources and utilizing machine learning algorithms for predictive insights. Intuitive user interfaces have been designed to facilitate easy navigation and data visualization. Robust data management protocols ensure data accuracy and compliance with healthcare regulations.

1.4 RESOURCES

To build our project, we'll need access to lots of healthcare data, good computers and software tools. We'll also need help from healthcare experts, researchers, and industry partners to make sure we're on the right track.

Technological resources will include access to software development tools and platforms such as React, Node.js, and relevant IDEs, as well as cloud infrastructure for data storage, processing, and hosting. Data integration tools will be necessary for seamless integration with existing healthcare systems, and security software will be crucial to protect sensitive patient data. These resources are essential to develop a robust, scalable, and secure platform.

1.5 MOTIVATION

We're excited about using data to make a big difference in healthcare. Our goal is to solve tough problems and make hospitals work better for everyone. Ultimately, we want to improve patient care and make healthcare more effective and accessible. In today's rapidly evolving healthcare landscape, there exists a pressing need for innovative solutions that can harness the vast amounts of data generated within healthcare systems and translate it into actionable insights. By leveraging advanced technologies such as React-enabled analytics, we aim to empower healthcare professionals with real-time data insights, predictive analytics, and intuitive user interfaces. This, in turn, will enable more informed decision-making, personalized patient care, and proactive management of health services.

CHAPTER 2

LITERATURE SURVEY

Healthcare organizations increasingly rely on healthcare analytics, using technologies like Hadoop and Spark, to support evidence-based decision-making. Despite the field's growth, systematic literature reviews (SLRs) are limited. Existing studies, such as those by De la Torre Diez, highlight the need for managing vast healthcare data. Design Science Research (DSR) offers a robust methodology, emphasizing relevance and rigor. Analyzing 52 articles revealed key focuses like real-time monitoring and common computing platforms, with most studies using software engineering methodologies and limited use of agile methods. This underscores the need for modern, contextually aware design approaches in healthcare analytics.

A systematic review and meta-analysis of 13 RCTs examined the impact of patient-clinician relationships on healthcare outcomes. The findings revealed a small but statistically significant positive effect. Despite the modest effect size and some methodological limitations, the results suggest that enhancing patient-clinician interactions can improve health outcomes. The studies varied in their interventions, including improved communication and empathy. Further rigorous RCTs are needed to confirm these findings and determine the most effective approaches for different patient populations.

The healthcare sector has significantly benefited from advancements in information technology (IT), transforming clinical practices and improving efficiency. IT functions have evolved to provide strategic support, enhance service quality, and streamline patient record management. Despite challenges like high costs and implementation difficulties in smaller clinics, IT has improved operational efficiency and data management. Tools such as Advanced Data Visualization, Presto, Hive, Hadoop, and cloud computing enable efficient data handling and real-time analytics, leading to better decision-making and patient outcomes. Overall, IT and data analytics are essential in modern healthcare, driving improvements in patient care and operational efficiency.

The review noted a growing trend in systematic secondary studies at the intersection of data analytics and healthcare. It found machine learning and data mining to be prevalent techniques, with opportunities for more in-depth research

in diverse healthcare contexts. Challenges such as data interoperability and privacy concerns underscored the need for robust computational infrastructure and ethical frameworks. Multidisciplinary collaboration between medical and technical experts emerged as crucial for effective implementation. Bridging the gap between analytical insights and practical healthcare applications remained a key focus for future endeavors in the field.

New ICT, OR, AI, and DM advancements are revolutionizing global healthcare, necessitating innovative approaches for data processing and decision making. Multidisciplinary collaboration across fields like Mathematics, Statistics, and Computer Science is driving innovative solutions for complex healthcare challenges. This special issue underscores the importance of analytical approaches and solution methodologies for data-intensive healthcare applications.

The research applies a system thinking approach, merging practitioner perspectives with academic literature to address healthcare informatics complexities. Embracing a pragmatic ideology, it emphasizes problem-solving and community involvement. Through a thorough review of big data literature, key themes like data mining and analytics are identified. Additionally, it explores the favorable impact of big data in healthcare while highlighting unintended repercussions such as privacy breaches and statistical sampling biases. Despite the advantages, ethical and technical hurdles persist, necessitating robust encryption and meticulous statistical methodologies to safeguard data integrity and privacy.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.2 ARCHITECTURE DIAGRAM

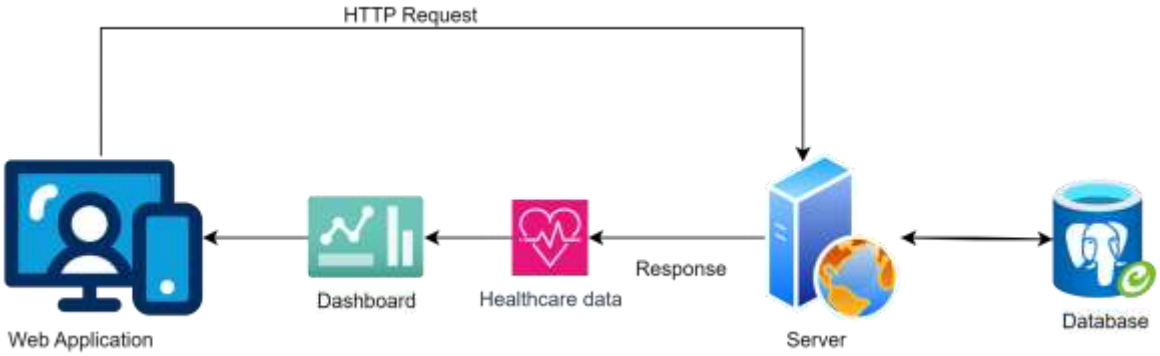

Fig 3.2.1 Architecture Diagram

3.3 DEVELOPMENT ENVIRONMENT

3.3.1 HARDWARE SPECIFICATIONS

This document offers a comprehensive overview of the hardware and its implementation, detailing the key components, their interactions, and the necessary requirements for seamless connectivity to utilities and installation.

PROCESSOR	Intel Core i5
RAM	8GB
GPU	Intel Iris XE
HARD DISK	40GB

Table 3.3.1 Hardware Specifications

3.3.2 SOFTWARE SPECIFICATIONS

The below table constitutes a thorough evaluation of requirements that precedes the more detailed phases of system design, aiming to minimize the need for subsequent revisions. Furthermore, it should offer a practical foundation for estimating product expenses, potential risks, and project timelines.

FRONT END	HTML, TailwindCSS, JavaScript, ReactJS, ChartJS
BACK END	NodeJS, Express
DATABASE	PostgreSQL
CODE EDITOR	Visual Studio Code
BROWSER	Chrome

Table 3.3.2 Software Specifications

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGY

The methodology for developing an analytics-integrated hospital management system involves several key steps. First, the project setup begins with creating a project directory, initializing a Git repository for version control, and setting up a package.json file for managing dependencies and scripts. For the backend, Express.js is installed and configured to handle HTTP requests, along with necessary middleware like body-parser and cors, and Nodemon for automatic server restarts during development. The frontend is initialized using Create React App, with additional installations of Chart.js and React-Chartjs-2 for data visualization. The next phase involves database design and setup, starting with schema planning to include tables for users, patients, appointments, and medical records, using an entity-relationship (ER) diagram for mapping relationships. PostgreSQL is installed and a new database is created, followed by table creation using SQL commands or a migration tool like Sequelize. In the backend development stage, a set of RESTful API endpoints is developed using Express.js for handling CRUD operations, with organized routing and controller functions for business logic, and custom middleware for authentication and validation. The frontend development focuses on designing the user interface with a component-based architecture in React, managing state with React's state management and context API, and integrating API endpoints using Axios or Fetch API. Data visualization is implemented using Chart.js to display patient statistics and appointment trends. Security measures include implementing JWT-based authentication, using bcrypt for password hashing, validating user inputs, and ensuring HTTPS for data transmission. This comprehensive approach ensures the development of a high-quality, secure, and efficient hospital management system with integrated analytics.

4.2 MODULE DESCRIPTION

4.2.1 DASHBOARD

The dashboard module in the hospital management system serves as a comprehensive and user-friendly interface for hospital administrators and staff to monitor and manage various aspects of hospital operations. This module aggregates and presents critical data in an easily digestible format, using real-time data visualization tools powered by Chart.js. It displays key metrics such as patient admissions, discharge rates, appointment schedules, and resource utilization, providing an at-a-glance overview of the hospital's performance. Interactive charts and graphs allow users to drill down into specific data points for more detailed analysis. By centralizing vital information and providing actionable insights, the dashboard module enhances decision-making, operational efficiency, and overall management effectiveness within the hospital.

4.2.2 DOCTOR DATABASE

The doctor information module in the hospital management system is designed to maintain comprehensive and up-to-date profiles of all medical practitioners associated with the hospital. This module captures essential details including personal information, contact details, specializations, qualifications, years of experience, and professional certifications. It also records schedules, availability, and assigned wards or departments. The module facilitates easy access to doctors' profiles for both administrative staff and patients, ensuring that the right doctor is matched to the right patient needs. By providing a centralized repository of doctor information, this module helps streamline administrative tasks, supports effective resource management, and enhances communication within the hospital.

4.2.3 INPATIENT ADMISSION

The inpatient admission module in the hospital management system streamlines the process of admitting patients for inpatient care, ensuring a smooth and efficient transition from outpatient to inpatient status. This module manages the entire admission workflow, starting from the initial admission request to bed assignment and patient transfer between departments. It captures and stores detailed patient information, including medical history, admission reason, attending physician, and any special requirements. The module facilitates real-time tracking of bed availability and occupancy rates, helping to optimize bed

utilization and reduce wait times.By centralizing and automating the admission process, the inpatient admission module enhances patient experience, improves operational efficiency, and supports effective hospital management.

4.2.4 PATIENT ROOM ALLOCATION

The room allocation module in the hospital management system is designed to efficiently manage the assignment and utilization of hospital rooms. This module automates the process of allocating rooms to patients based on various criteria such as medical needs, room availability, and patient preferences. It keeps real-time records of room occupancy, ensuring that the hospital can maximize its capacity while minimizing downtime between patient admissions. The module supports different types of room configurations, including private rooms, shared wards, and specialty care units, allowing for flexible and optimized space management. Integration with other modules, such as the inpatient admission and discharge modules, ensures seamless transitions and updates across the system. By providing a comprehensive overview of room status and facilitating quick decision-making, the room allocation module enhances operational efficiency, reduces administrative burden, and improves patient satisfaction by ensuring appropriate and timely room assignments.

4.2.5 APPOINTMENT FOR OUTPATIENTS

The outpatient appointment module in the hospital management system facilitates the scheduling and management of appointments for patients seeking outpatient care. It captures essential patient information, including personal details, medical history, and preferred appointment times. The module integrates with the doctor information module to enable selection of preferred physicians and specialists based on availability and expertise. Administrators and staff can manage appointment schedules, view availability, and assign appointments to specific doctors or departments.By automating appointment management and improving patient access to care, the outpatient appointment module enhances operational efficiency, patient satisfaction, and overall quality of outpatient services within the hospital.

4.2.6 TREATMENTS HISTORY

The treatment history module in the hospital management system provides a comprehensive record of all treatments and procedures administered to patients during their hospital visits. This module captures and organizes detailed

information about each treatment, including medications prescribed, procedures performed, diagnostic tests ordered, and therapeutic interventions given. It integrates with other modules, such as the patient information module, to maintain a complete patient profile that includes medical histories, allergies, and ongoing treatments. Healthcare providers can easily access and update treatment records, ensuring continuity of care and accurate documentation. The module supports the retrieval of historical treatment data for patient consultations, enabling healthcare providers to make informed decisions about current and future treatments.

4.2.7 PATIENT DISCHARGE HANDLING

Administrators and staff can track discharge statuses and manage discharge tasks, such as medication reconciliation and patient education, to ensure a smooth transition from hospital care to home or another healthcare setting. The module also supports discharge planning for patients who require ongoing care or specialized services post-discharge. By automating and optimizing the discharge process, this module enhances patient safety, improves bed turnover rates, and reduces hospital readmissions. It also supports effective communication and collaboration among healthcare providers, ensuring continuity of care and improving overall patient experience and satisfaction.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working Application.

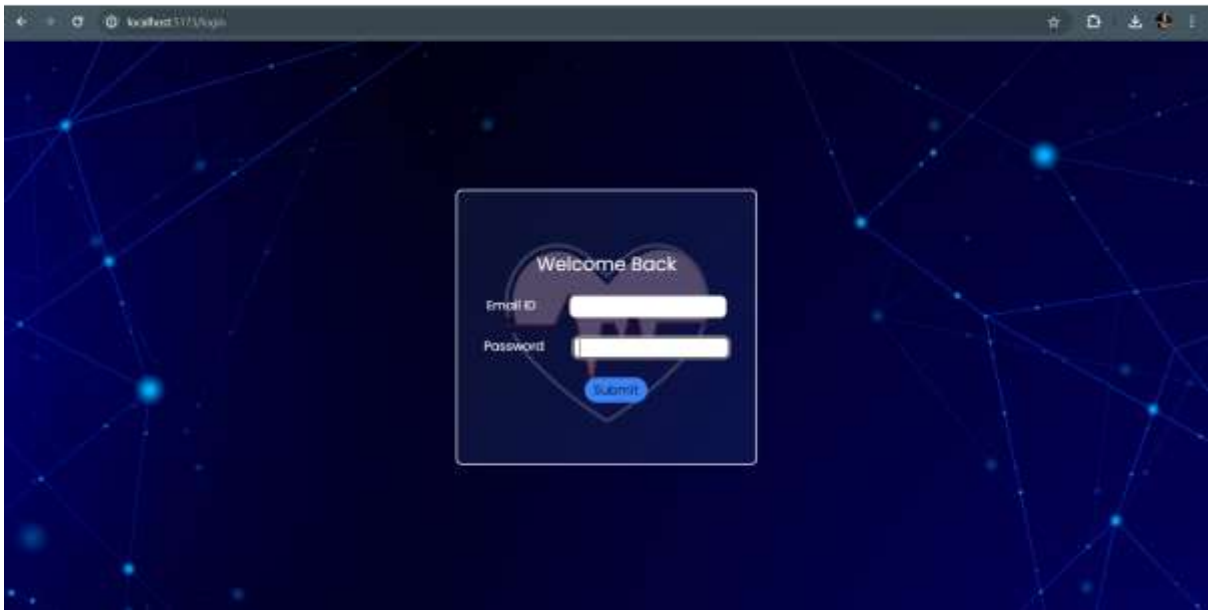

Fig 5.1.1 Login Page

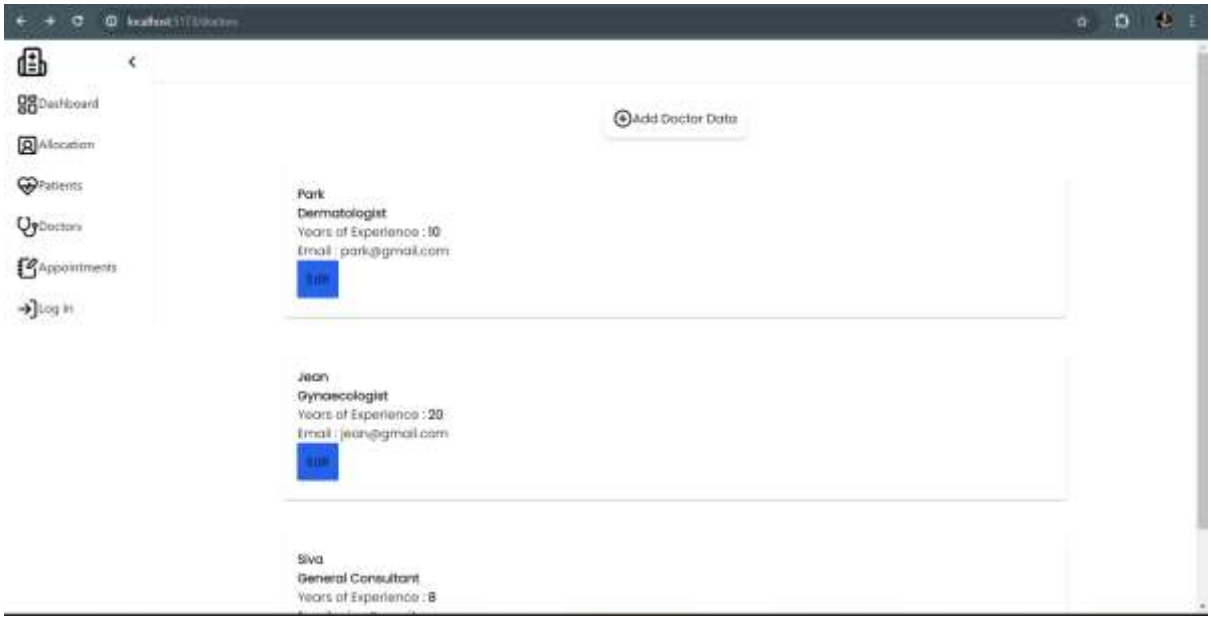

Fig 5.1.2 Doctors Page

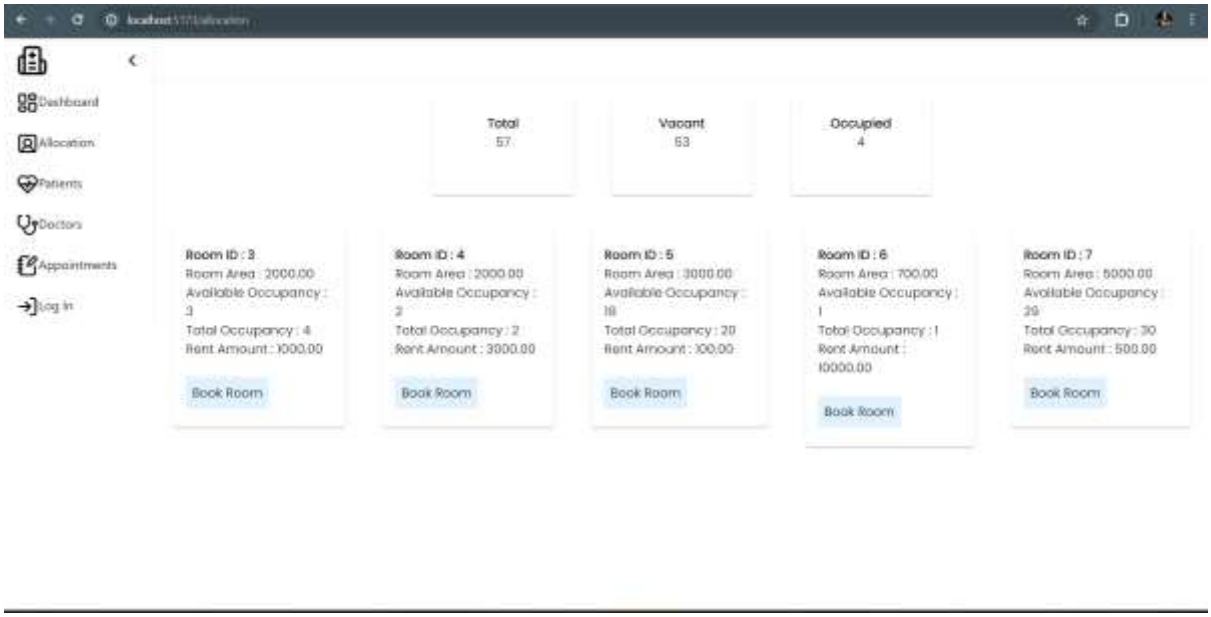

Fig 5.1.3 Room Allocation Page

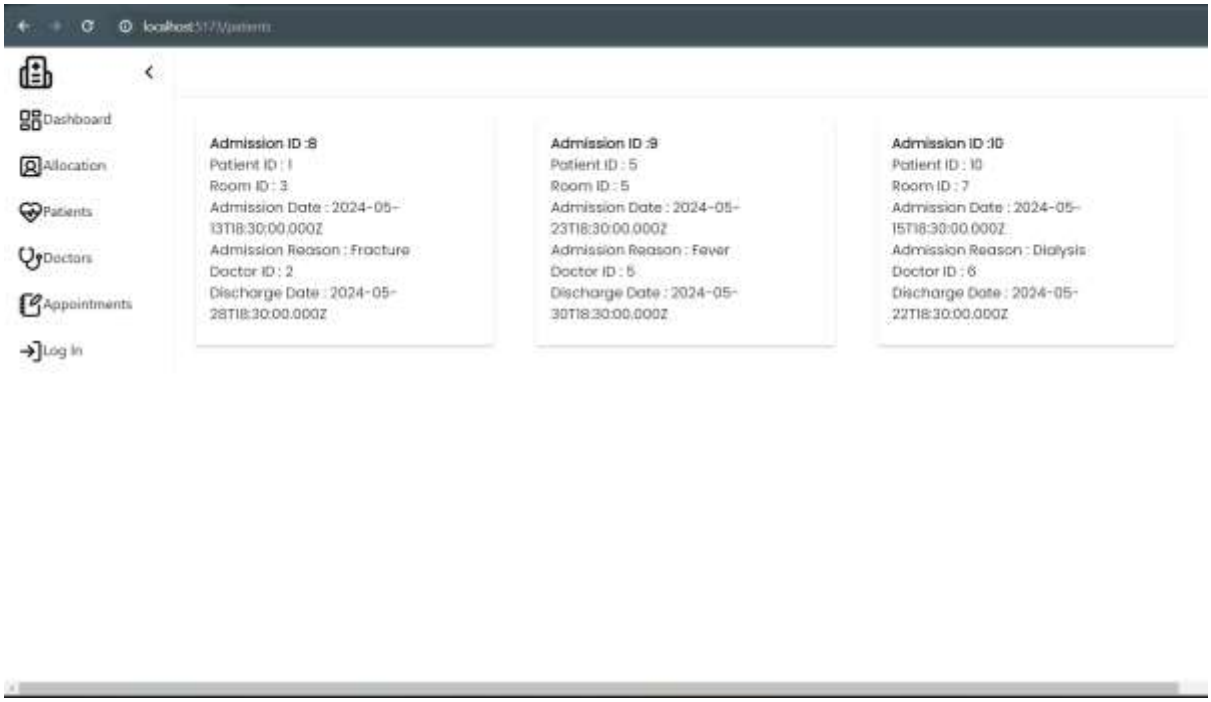

Fig 5.1.4 Admissions Page

Fig 5.1.5 Dashboard Page

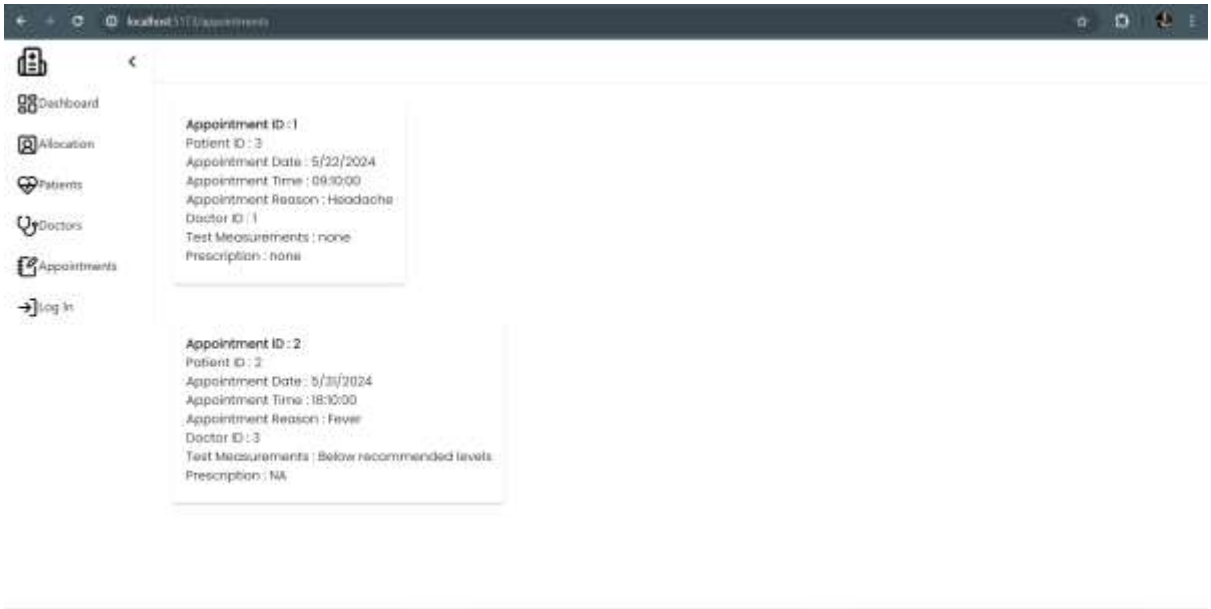

Fig 5.1.6 Appointments Page

Fig 5.1.7 Doctor Registration Form

```
healthcare=# select * from rooms;
 room_area | curr_avbl_occ_capacity | total_occ_capacity | rent_amt | availability | room_id
-----------+------------------------+--------------------+----------+--------------+---------
    700.00 |                      1 |                  1 | 10000.00 | Y            |       6
   2000.00 |                      3 |                  4 |  1000.00 | Y            |       3
   5000.00 |                     29 |                 30 |   500.00 | Y            |       7
   3000.00 |                     18 |                 20 |   100.00 | Y            |       5
   2000.00 |                      2 |                  2 |  3000.00 | Y            |       4
(5 rows)
```

Fig 5.1.8 Database - Rooms Table

```
healthcare=# select * from inpatients;
 admission_id | patient_id | room_id | admission_date | admission_reason | primary_doctor_id | discharge_date | discharge_status
--------------+------------+---------+----------------+------------------+-------------------+----------------+------------------
            8 |          1 |       3 | 2024-05-14     | Fracture         |                 2 | 2024-05-29     | N
            9 |          5 |       5 | 2024-05-24     | Fever            |                 5 | 2024-05-31     | N
           10 |         10 |       7 | 2024-05-16     | Dialysis         |                 6 | 2024-05-23     | N
           12 |         56 |       5 | 2024-05-24     | Surgery          |                 2 | 2024-05-31     | N
           11 |          7 |       4 | 2024-05-03     | Stomach ache     |                 4 | 2024-05-15     | Y
(5 rows)
```

Fig 5.1.9 Database - InPatients Table

```
healthcare=# select * from users;
          email          |                                  password                                  | name | user_id
-------------------------+----------------------------------------------------------------------------+------+---------
 haresh@gmail.com        | $2b$10$TLtbWWJ.t7iUVMS2pJB2c.N.qRDuYhQReB0/E1tRjPgBIqLB0NQWG               |      |       2
 hayagreevan@gmail.com   | $2b$10$/dFFmlYQn4aSWxGn45mF1O.SpPqbfrXTAI8UFCIZpTpz8E2foATfq               |      |       3
 210701066@rajalakshmi.edu.in | $2b$10$iXUcQsM.ShPgGUk8IlkTJu6P.Pp/s10XOgTWAeKkQxKwQ8drGC5me           |      |       4
 sam@gmail.com           | $2b$10$k4wBRPQUw1CmFPrOjdcOMOXdU6p8TjXembdHxxYUH8AX/2La5MbUu               |      |       5
 firefly@gmail.com       | $2b$10$zEaKfUCXUmYv78EzzWL6i.PkDnnd4DI82ArisOcadrxzRIpIAfsIW               |      |       6
 maxwell@gmai.com        | $2b$10$LmpR27vGlAFqM7oLoRxZ5uLPukeprLAKIlPxEhG/sfqnaI.pzJwi6               |      |       7
(6 rows)
```

Fig 5.1.10 Database - Users Table

5.2 RESULT

The hospital management system project aims to deliver improved operational efficiency, enhanced patient care, and data-driven decision-making through automation of administrative tasks, real-time analytics, and streamlined workflow management. It will optimize resource allocation, improve communication among healthcare providers, and ensure accurate documentation and compliance. Overall, the system aims to enhance hospital management, reduce costs, and improve patient outcomes.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The hospital management system project aims to deliver a robust and integrated software solution that enhances operational efficiency, improves patient care quality, and supports informed decision-making within the hospital setting. By automating administrative tasks such as patient admission, room allocation, and discharge processes, the system reduces manual errors and streamlines workflows, allowing healthcare providers to focus more on patient treatment and care. Real-time analytics and reporting capabilities enable hospital administrators to make data-driven decisions based on critical metrics like patient admissions, resource utilization, and clinical outcomes. Overall, the system aims to optimize resource management, improve communication and coordination among healthcare teams, and enhance patient satisfaction by ensuring efficient and effective healthcare delivery.

6.2 FUTURE ENHANCEMENT

For future enhancements, the hospital management system project could integrate telemedicine capabilities for virtual consultations and remote patient monitoring, predictive analytics for optimizing resource allocation and anticipating healthcare trends, and a mobile application for accessing medical records and scheduling appointments. Additionally, incorporating Internet of Things (IoT) devices for patient monitoring, strengthening cybersecurity measures, implementing machine learning for clinical decision-making, and developing a patient portal for health information access would further enhance patient care and operational efficiency. Ensuring interoperability with external systems such as electronic health records and pharmacies would facilitate seamless data exchange and improve care continuity, making the system adaptable to future healthcare needs.

APPENDIX

SOURCE CODE

server.js :

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const bcrypt = require("bcrypt");
const morgan = require("morgan");
const cron = require('node-cron');

const app = express();
const PORT = process.env.PORT || 3500;

const authRouter = require("./routes/auth.router");
const docRouter = require('./routes/doctors.router');
const roomsRouter = require("./routes/rooms.router");
const inPatientRouter = require('./routes/inpatient.router');
const appointmentRouter = require('./routes/appointments.router');
const treatmentRouter = require('./routes/treatments.router');

const dischargeUpdate = require('./discharge-update');

const db = require("./config/dbConn");
db.connect().then(()=>{
   console.log("Postgres database is connected!");
});

app.get('/',(req,res)=>{
   res.send("Nothing here!");
});
app.use(bodyParser.urlencoded({extended:true}));
app.use(morgan('tiny'));
app.use(cors({
   credentials: true,
   origin: "http://localhost:5173"
```

```
}));

app.use("/auth",authRouter);
app.use("/doctor",docRouter);
app.use("/rooms",roomsRouter);
app.use("/inpatient",inPatientRouter);
app.use("/appointments",appointmentRouter);
app.use("/treatments",treatmentRouter);

// Discharge scheduled to happen every day at 8:30 PM
cron.schedule('30 20 * * *', () => {
   dischargeUpdate();
});

app.listen(PORT, ()=>{
   console.log(`Server is running at PORT ${PORT}`);
   dischargeUpdate();
})
```

auth.router.js

```
const express = require("express");
const jwt = require("jsonwebtoken");
const bcrypt = require("bcrypt");
require("dotenv").config();


const router = express.Router();
const db = require("../config/dbConn");

router.get('/',(req,res)=>{
   res.send("Auth Route");
});

router.post('/login',async(req,res)=>{
   // console.log(req.body);
   const result = await db.query('SELECT * from users where email =
$1',[req.body.email]);
```

```
    if(result.rows.length==0){
        res.status(301).send("User not found");
    }
    const authentication = await
bcrypt.compare(req.body.password,result.rows[0].password);
    if(authentication){
        const token =
jwt.sign({email:req.body.email,user_id:result.rows[0].user_id,role:
result.rows[0].role},process.env.ACCESS_TOKEN_SECRET);
        res.cookie('jwt',token);
        console.log(token);
        res.status(200).send({accessToken:token});
    }else{
        res.status(400).send("Invalid Password");
    }
});

router.post('/register',async(req,res)=>{
    // console.log(req.body);
    const result = await db.query('SELECT * from users where email =
$1',[req.body.email]);
    if(result.rows.length==0){
        const hashedPwd = await bcrypt.hash(req.body.password,10);
        const query = await db.query("Insert into users(name,email,password,role)
values($1,$2,$3,$4);",[req.body.name,req.body.email,hashedPwd,req.body.role]
);
        console.log(query.rows);

Const token =
jwt.sign({email:req.body.email},process.env.ACCESS_TOKEN_SECRET);
        console.log(token);
        // res.status(200).send(token);
        res.cookie('jwt',token);
        res.status(200).send({accessToken:token});
    }
    else{
        res.status(400).send("User Already exists!");
```

```
    }
});

router.post('/logout',async(req,res)=>{
    res.cookie('jwt','',{ maxAge: 1, httpOnly: true });
    res.status(200).send("Logged Out");
})

module.exports = router;
```

appointments.router.js

```
const express = require('express');
const db = require('../config/dbConn');
const router = express.Router();

router.get('/',(req,res)=>{
    res.send("Appointment Route");
});

router.get('/all',async(req,res)=>{
    const result = await db.query('SELECT * FROM OUTPATIENTS');
    res.status(200).send(result.rows);
});

router.get('/today',async(req,res)=>{
    const result = await db.query('SELECT * FROM OUTPATIENTS WHERE
APPOINTMENT_DATE = $1',[new Date()]);
    res.status(200).send(result.rows);
});

router.post('/add', async(req, res)=>{
    const data = req.body;
    console.log(data);
        await db.query('INSERT INTO
OUTPATIENTS(PATIENT_ID,APPOINTMENT_DATE,
APPOINTMENT_TIME, APPOINTMENT_REASON, DOCTOR_ID,
TEST_MSRMNT_DTL, PRESCRIPTION_DTL)
```

```
VALUES($1,$2,$3,$4,$5,$6,$7)',[data.patient_id, data.appointment_date,
data.appointment_time, data.appointment_reason, data.doctor_id,
data.test_msrmnt_dtl, data.prescription_dtl]);
   res.status(201).send("Appointment Created!");
})

router.get('/getappointments', async(res, req)=>{

})

module.exports = router;
```

rooms.router.js

```
const express = require('express');
const db = require('../config/dbConn');
const router = express.Router();

router.get('/',(req,res)=>{
   res.send("Rooms Route");
});

router.get('/details',async(req,res)=>{
   const result = await db.query("SELECT * FROM ROOMS WHERE
AVAILABILItY ='Y' ORDER BY ROOM_ID");
   res.status(200).send(result.rows);
});

router.get('/summary',async(req,res)=>{
   const result = await db.query("SELECT SUM(curr_avbl_occ_capacity) AS
CURR_AVBL, SUM(total_occ_capacity) AS TOTAL FROM ROOMS
WHERE AVAILABILITY = $1",['Y']);
   res.status(200).send(result.rows[0]);
});

module.exports = router;
```

Doctors.jsx

```
import NavigationBar from "../components/NavigationBar"
import SideMenu from "../components/SideMenu"
import Card  from "@mui/material/Card";
import CardContent from "@mui/material/CardContent";
import axios from "axios";
import { PlusCircleIcon } from "lucide-react";
import { useEffect, useState } from "react";
import { useNavigate, Link } from "react-router-dom";

export const Doctors = () => {
  const navigate = useNavigate();
  const cardStyles = "m-6 h-fit w-3/4 hover:scale-105 hover:transition-all
hover:bg-gradient-to-r from-slate-600 to-slate-300 hover:text-white";
  const [doctors,setDoctors] = useState([]);

  const getData= async() =>{
    await
axios.get(import.meta.env.VITE_SERVER_URL+'/doctor/all').then((response)=
>{
      setDoctors(response.data);
      console.log(response.data);
    });
  }

  useEffect(()=> {
    getData();
  },[]);

  return (
    <div className="flex flex-row">
      <SideMenu/>
      <div className="h-screen w-screen bg-white font-poppins">
      <NavigationBar/>
      <div className="flex flex-col items-center">
```

```
    <Link to = "/insertdoctor"><div className="h-fit p-3 m-5 hover:bg-slate-
500 hover:scale-110 hover:transition-all hover:text-white rounded-lg shadow-
md flex flex-row"><PlusCircleIcon/> Add Doctor Data</div></Link>
      {
        doctors?.map((doctor,index)=>{
          return(
                                    <Card key = {index} className={cardStyles}>
              <CardContent>
                                <h1><strong>{doctor.name}</strong></h1>
                                <p><strong>{doctor.specialization}</strong></p>
                <p>Years of Experience : <strong>{doctor.years_expr}
</strong></p>
                <p>Email : {doctor.email}</p>
                <button onClick =
{()=>{navigate(`/updatedoctor/${doctor.doctor_id}`)}} className="bg-blue-
600 p-3 rounded-sm hover:bg-pink-200 hover:scale-110  hover:transition-all w-
fit">Edit</button>
                        </CardContent>
                        </Card>
        )})
      }
    </div>
    </div>
   </div>
  )
}

export default Doctors
```

RoomAllocation.jsx

```
import SideMenu from "../components/SideMenu"
import NavigationBar from "../components/NavigationBar"
import { useState, useEffect } from "react"
import axios from "axios"
import Card from "@mui/material/Card"
import CardContent from "@mui/material/CardContent"
import { useNavigate } from "react-router-dom"
```

```
const RoomAllocation = () => {
	const [summary, setSummary] = useState({});
	const [data, setData] = useState([]);

	const navigate = useNavigate();
	const cardStyling1 = "m-6 h-[120px] w-[180px] text-center"
	const cardStyling2 = "m-6 h-[120px] w-[180px] text-center"

	async function fetchData(){
		await
axios.get(import.meta.env.VITE_SERVER_URL+'/rooms/summary').then((resp
onse)=>{
			setSummary(response.data);
			console.log(response.data);
		});

		await
axios.get(import.meta.env.VITE_SERVER_URL+'/rooms/details').then((respon
se)=>{
			setData(response.data);
			console.log(response.data);
		});
	}
	useEffect(()=>{
		fetchData();
	},[]);

	return (
	<div className="flex flex-row">
		<SideMenu/>
		<div className="h-screen w-screen bg-white font-poppins">
			<NavigationBar/>
			<div className="flex flex-row justify-center">

				<Card className={cardStyling2}>
					<CardContent>
```

```
                                    <h1 className="font-
bold">{"Total"}</h1>
                                    <p>{summary?.total}</p>
                                </CardContent>
                            </Card>

                            <Card className={cardStyling1}>
                                <CardContent>
                                    <h1 className="font-
bold">{"Vacant"}</h1>
                                    <p>{summary?.curr_avbl}</p>
                                </CardContent>
                            </Card>

                            <Card className={cardStyling2}>
                                <CardContent>
                                    <h1 className="font-
bold">{"Occupied"}</h1>
                                    <p>{summary?.total -
summary?.curr_avbl}</p>
                                </CardContent>
                            </Card>
                        </div>
                        <div className="flex flex-row justify-center">
                        {
                            data?.map((row,index)=>{
return(
    <div className="p-6 h-fit w-fit">
        <Card key={index}>
            <CardContent>
            <h1 className="font-bold">Room ID :
    {row.room_id}</h1>
            <p>Room Area : {row.room_area}</p>
            <p>Available Occupancy :
{row.curr_avbl_occ_capacity}</p>    <p>Total Occupancy :
{row.total_occ_capacity}</p>
            <p>Rent Amount : {row.rent_amt}</p><br/>
```

```
        <button onClick={()=>{navigate(`/createinpatient/${row.room_id}`);}}
className="bg-sky-100 p-2 rounded-sm hover:bg-pink-200 hover:scale-110
hover:transition-all">Book Room</button>
                        </CardContent>
                        </Card>
                        </div>
                                )
                        }
                )
        }
        </div>
        </div>
        </div>
  )
}

export default RoomAllocation
```

SideNavBar.jsx

```
import { useContext, useState } from "react";
import { ChevronLeft, ChevronRight, LayoutDashboardIcon, HospitalIcon}
from 'lucide-react';
import { createContext } from "react";

const SideBarContext = createContext();
const SideNavBar = ({ children }) => {
   const [isOpen, setIsOpen] = useState(true);
   return (
      <div className={`flex h-fit bg-gray-200 ${isOpen ? "w-48" : "w-16"}
transition-all duration-300`}>
         {/* Sidebar */}
         <div className={`bg-white border-r ${isOpen ? "" : "border-gray-300"}
w-full flex flex-col`}>
            {/* Sidebar Header */}
            <div className="flex items-center justify-between p-4 h-14">
               <HospitalIcon className = "h-10 w-10" />
```

```
            <button onClick={() => setIsOpen(!isOpen)} className="focus:outline-none">
                {isOpen ? <ChevronLeft className= "w-6 h-6" /> : <ChevronRight className="w-4 h-4" />}
            </button>
          </div>
          {/* Sidebar Content */}
          <SideBarContext.Provider value = {{isOpen}}>
            <ul className="flex flex-col flex-1 overflow-hidden">
            {children}
            </ul>
          </SideBarContext.Provider>
        </div>
      </div>
  );
};

export default SideNavBar;
export const SideBarItem = ({text, icon}) =>{
  const {isOpen} = useContext(SideBarContext)
  return(
    <li className={`hover:bg-slate-500 hover:text-white overflow-hidden h-15 p-3 cursor-pointer ${isOpen ? "w-64" : "w-16" }`}>
      <div className="flex flex-row justify-start hover:scale-110 hover:transition-all">
        {icon}
        <span className={`overflow-hidden ${isOpen ? "" : "w-0"}`}>{text}</span>
      </div>
    </li>
  )
}
```

REFERENCES

1. Miah, S. J., Gammack, J., & Hasan, N. (2019, December 26). Methodologies for designing healthcare analytics solutions: A literature analysis. Health Informatics Journal, 26(4), 2300–2314. https://doi.org/10.1177/1460458219895386

2. Kelley, J. M., Kraft-Todd, G., Schapira, L., Kossowsky, J., & Riess, H. (2014, April 9). The Influence of the Patient-Clinician Relationship on Healthcare Outcomes: A Systematic Review and Meta-Analysis of Randomized Controlled Trials. PLoS ONE, 9(4), e94207. https://doi.org/10.1371/journal.pone.0094207

3. Taipalus, T., Isomöttönen, V., Erkkilä, H., & Äyrämö, S. (2022, December 9). Data Analytics in Healthcare: A Tertiary Study. SN Computer Science, 4(1). https://doi.org/10.1007/s42979-022-01507-0

4. Yang H, Kundakcioglu OE, Zeng D. Healthcare data analytics. Inf Syst e-Bus Manag. 2015;13(4):595–7. https://doi.org/10.1007/s10257-015-0297-0

5. Strang, K. D., & Sun, Z. (2019, July 2). Hidden big data analytics issues in the healthcare industry. Health Informatics Journal, 26(2), 981–998. https://doi.org/10.1177/1460458219854603