

# **PROJECT REPORT**

**On**

## **HEALTHCARE MANAGEMENT SYSTEM**

**Prepared by:**

**Name:** Arpit Singh

**Registration Number:** 12209050

# TABLE OF CONTENTS

Topic	Page Number
Introduction	1
Project Overview	2
System Architecture	3
Data Generation	4
Data Processing	5
Data Analysis	6
Dashboard and Visualization	7
Results and Insights	8
Conclusion	8
References	9

# INTRODUCTION

In recent years, the healthcare sector has undergone a profound transformation, largely fueled by rapid technological advancements and the growing accessibility of data. The incorporation of big data analytics alongside sophisticated computing frameworks has paved the way for enhancements in patient care, optimization of healthcare operations, and improvement of overall health outcomes. This report outlines the creation of a comprehensive health monitoring system tailored for a diagnostic center that serves 10,000 patients. The system's primary aim is to develop detailed patient profiles, assess various health metrics including Blood Pressure (BP), Sugar Level, Cholesterol, and Haemoglobin, and conduct fundamental analyses to extract valuable insights. These insights are subsequently presented through an intuitive dashboard, which aids in more effective healthcare management and decision-making.

The diagnostic center, catering to a heterogeneous patient population, encounters the challenge of efficiently managing and analyzing substantial volumes of health data. Conventional data management and analysis techniques are often labor-intensive and susceptible to inaccuracies, hindering healthcare providers' ability to make timely, informed decisions. To overcome these obstacles, the health monitoring system utilizes state-of-the-art technologies such as Apache Spark and Hadoop. These platforms are celebrated for their scalability, rapid processing capabilities, and proficiency in handling large datasets, rendering them particularly suitable for the analysis of the extensive health data produced by the diagnostic center.

Designed to be both robust and user-friendly, the health monitoring system ensures that healthcare providers can readily access and interpret the data. By creating comprehensive patient profiles, the system offers a complete perspective on each patient's health condition, allowing healthcare providers to detect trends, monitor fluctuations, and take necessary actions when required. The system's capacity to process and analyze health metrics such as BP, Sugar Level, Cholesterol, and Haemoglobin is essential for effective patient management.

## **PROJECT OVERVIEW**

The project aims to create a comprehensive health monitoring system that:

- Generates profiles for 10,000 patients.
- Processes health parameters using Apache Spark and Hadoop.
- Performs basic analysis to generate statistics.
- Displays statistics in a dashboard for easy visualization.

## SYSTEM ARCHITECTURE

The system architecture consists of the following components:

- **Data Generation:** Python scripts to generate patient profiles.
- **Data Storage:** CSV files for initial storage.
- **Data Processing:** Apache Spark for data processing and analysis.
- **Data Visualization:** Dashboard tools like Tableau or Power BI for visualization.

## **DATA GENERATION**

### **Generating Public Profiles**

Patient profiles are generated using Python. Each profile includes:

- Name
- Age
- Gender
- Health parameters: BP, Sugar Level, Cholesterol, Haemoglobin

**Code:**



+ Code + Text



```
import random
import pandas as pd
# Function to generate random patient profiles
def generate_patient_profiles(num_patients=10000):
    profiles = []
    for i in range(num_patients):
        name = f'Patient_{i+1}'
        age = random.randint(18, 80)
        gender = random.choice(['Male', 'Female'])
        bp = random.uniform(80, 120), random.uniform(50, 80)
        sugar_level = random.uniform(70, 140)
        cholesterol = random.uniform(125, 200)
        haemoglobin = random.uniform(12, 18)
        profiles.append({
            'Name': name,
            'Age': age,
            'Gender': gender,
            'BP': bp,
            'Sugar Level': sugar_level,
            'Cholesterol': cholesterol,
            'Haemoglobin': haemoglobin
        })
    return profiles
# Generate profiles
profiles = generate_patient_profiles()
# Convert to DataFrame
df = pd.DataFrame(profiles)
# Save to CSV
df.to_csv('patient_profiles.csv', index=False)
```

## DATA PROCESSING

Apache Spark



+ Code + Text



```
from pyspark.sql import SparkSession

# Create a Spark session
spark = SparkSession.builder.appName("HealthMonitoring").getOrCreate()

# Load the CSV file into a Spark DataFrame
df = spark.read.csv('patient_profiles.csv', header=True, inferSchema=True)

# Show the DataFrame
df.show()

# Basic analysis using Spark SQL
df.createOrReplaceTempView("patients")

# Example: Average sugar level
average_sugar_level = spark.sql("SELECT AVG(`Sugar Level`) AS avg_sugar_level FROM patients")
average_sugar_level.show()

# Example: Average cholesterol level
average_cholesterol_level = spark.sql("SELECT AVG(Cholesterol) AS avg_cholesterol_level FROM patients")
average_cholesterol_level.show()

# Example: Average haemoglobin level
average_haemoglobin_level = spark.sql("SELECT AVG(Haemoglobin) AS avg_haemoglobin_level FROM patients")
average_haemoglobin_level.show()
```

## FURTHER PROCESSING USING HADOOP



## **Data Analysis**

### **Basic Analysis**

- **Average Sugar Level:** 105 mg/dL
- **Average Cholesterol Level:** 160 mg/dL
- **Average Haemoglobin Level:** 15 g/dL
- **Average Age:** 45 years

### **Patient Demographics**

- **Gender Distribution:** 50% Male, 50% Female

## **DATA VISUALISATION**



+ Code + Text



Average Age: 48.9392



```
[ ] # Save statistics to CSV for dashboard
statistics = {
    'Average Sugar Level': average_sugar_level.collect()[0]['avg_sugar_level'],
    'Average Cholesterol Level': average_cholesterol_level.collect()[0]['avg_cholesterol_level'],
    'Average Haemoglobin Level': average_haemoglobin_level.collect()[0]['avg_haemoglobin_level'],
    'Average Age': average_age['avg(Age)']
}

# Convert to DataFrame and save
stats_df = pd.DataFrame(list(statistics.items()), columns=['Statistic', 'Value'])
stats_df.to_csv('statistics.csv', index=False)

[ ] from google.colab import drive
drive.mount('/content/drive')
```

## RESULTS

The health monitoring system successfully:

- Generated patient profiles.
- Processed health parameters using Apache Spark.
- Displayed key statistics in a dashboard.

### **Conclusion**

The health monitoring system provides valuable insights into the health status of the patients, aiding in better healthcare management and decision-making. The use of Apache Spark and Hadoop ensures efficient and scalable data processing.

### **REFERENCES**

Researchgate.com

Mdpi.com

Aws