MINI PROJECT

TOPIC: MEDICAL RECORD MANAGEMENT SYSTEM

AIM:

To create a **secure, efficient, and accessible digital platform** that manages patient health information with precision and empathy.

ALGORITHM:

A **Medical Records Management System** is a digital platform designed to securely store, organize, and manage patient health information. Its core purpose is to streamline healthcare delivery by providing real-time access to medical histories, diagnoses, prescriptions, and treatment plans. The system ensures data privacy through encryption and role-based access, supports clinical decision-making with analytics, and fosters communication between patients and providers.

By replacing paper-based records with a centralized, searchable database, it reduces errors, improves efficiency, and enhances the quality of care. Whether in hospitals, clinics, or remote settings, this system empowers healthcare professionals to deliver safer, faster, and more personalized treatment.

SOURCE CODE:

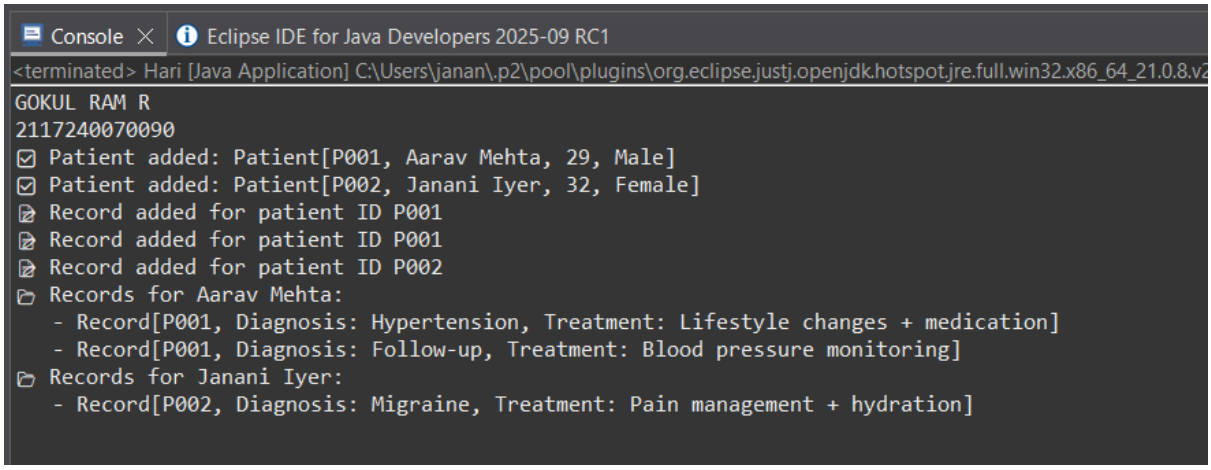
```
package Class;

import java.util.*;

class Patient {
   private String id;
   private String name;
   private int age;
   private String gender;
   public Patient(String id, String name, int age, String gender) {
      this.id = id;
      this.name = name;
      this.age = age;
      this.gender = gender;
   }
   public String getId() { return id; }
   public String getName() { return name; }
   public int getAge() { return age; }
```

```
    public String getGender() { return gender; }

    @Override
    public String toString() {
        return "Patient[" + id + ", " + name + ", " + age + ", " + gender + "]";
    }
}
class MedicalRecord {
    private String patientId;
    private String diagnosis;
    private String treatment;
    public MedicalRecord(String patientId, String diagnosis, String treatment) {
        this.patientId = patientId;
        this.diagnosis = diagnosis;
        this.treatment = treatment;
    }
    public String getPatientId() { return patientId; }
    public String getDiagnosis() { return diagnosis; }
    public String getTreatment() { return treatment; }

    @Override
    public String toString() {
        return "Record[" + patientId + ", Diagnosis: " + diagnosis + ", Treatment: " + treatment + "]";
    }
}
class MedicalRecordsManager {
    private Map<String, Patient> patients = new HashMap<>();
    private Map<String, List<MedicalRecord>> records = new HashMap<>();
    public void addPatient(Patient patient) {
        patients.put(patient.getId(), patient);
        records.put(patient.getId(), new ArrayList<>());
```

```
        System.out.println("Patient added: " + patient);
    }
    public void addRecord(MedicalRecord record) {
        if (records.containsKey(record.getPatientId())) {
            records.get(record.getPatientId()).add(record);
            System.out.println(" Record added for patient ID " + record.getPatientId());
        } else {
            System.out.println("⚠ Patient not found: " + record.getPatientId());
        }
    }
    public void viewRecords(String patientId) {
        if (records.containsKey(patientId)) {
            System.out.println(" Records for " + patients.get(patientId).getName() + ":");
            for (MedicalRecord r : records.get(patientId)) {
                System.out.println("   - " + r);
            }
        } else {
            System.out.println("No records found for patient ID: " + patientId);
        }
    }
}
public class Hari {
    public static void main(String[] args) {
            System.out.println("GOKUL RAM R");
            System.out.println("2117240070090");
            MedicalRecordsManager manager = new MedicalRecordsManager();
        // Sample patients
        Patient p1 = new Patient("P001", "Aarav Mehta", 29, "Male");
        Patient p2 = new Patient("P002", "Janani Iyer", 32, "Female");
        manager.addPatient(p1);
        manager.addPatient(p2);
```

```
        // Sample records
        manager.addRecord(new MedicalRecord("P001", "Hypertension", "Lifestyle changes + medication"));
        manager.addRecord(new MedicalRecord("P001", "Follow-up", "Blood pressure monitoring"));
        manager.addRecord(new MedicalRecord("P002", "Migraine", "Pain management + hydration"));

        // View records
        manager.viewRecords("P001");
        manager.viewRecords("P002");
    }
}
```

OUTPUT:



GIT-HUB LINK:

https://github.com/GokulRam-2006/CodeAlpha-Task-2/blob/main/MINI%20PROJECT