

Name: Kunal Sanjay Ekare

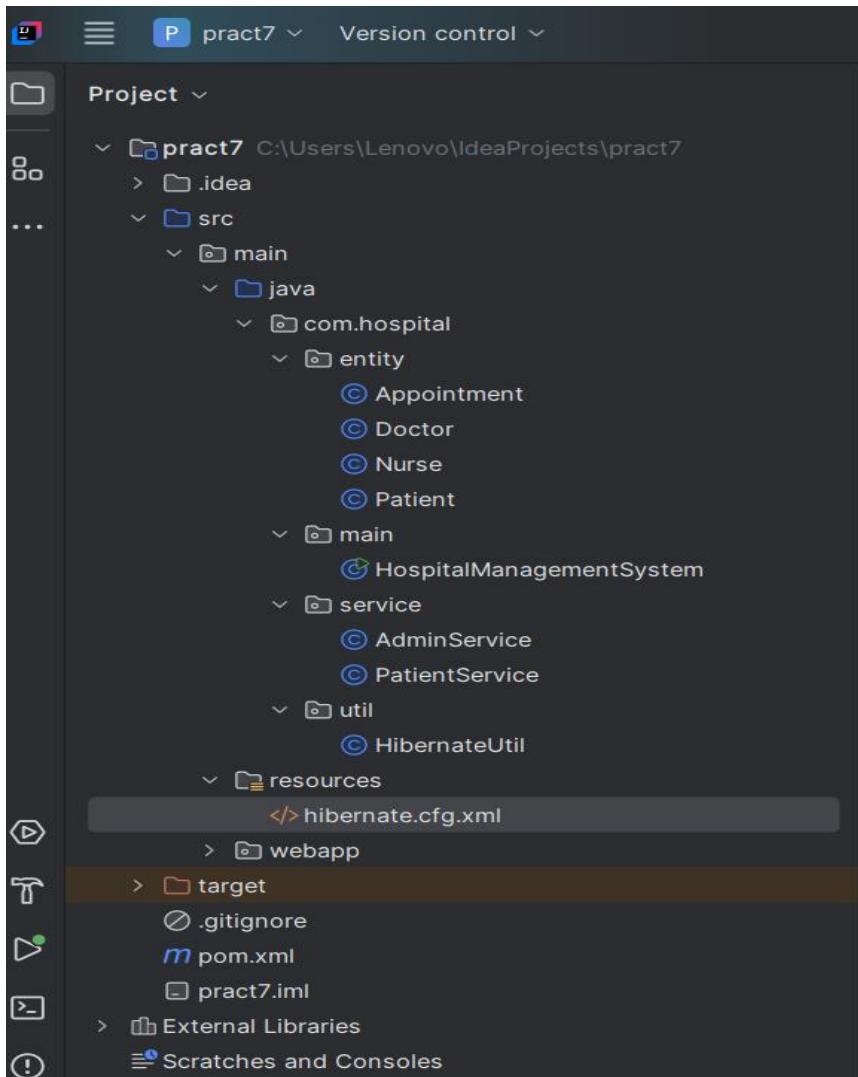
Batch/RollNo:B2-37

Subject : Software lab

Practical :7

Aim: Demonstrate the use of Hibernate to persist data in a menu driven Program
Design a menu driven application for Hospital Management System having two types of users Admin and Doctor. Admin can add new Doctor/Nurse and view all doctors appointments. Patient can book, Cancel and view his own appointments.

Code:



Code: Appointment.java

```
package com.hospital.entity;

import jakarta.persistence.*;
import java.util.Date;

@Entity
@Table(name = "appointments")
public class Appointment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @ManyToOne
    @JoinColumn(name = "doctor_id", nullable = false)
    private Doctor doctor;

    @ManyToOne
    @JoinColumn(name = "patient_id", nullable = false)
    private Patient patient;

    @Column(name = "date", nullable = false)
    @Temporal(TemporalType.DATE)
    private Date date;

    public Appointment() {}

    public Appointment(Doctor doctor, Patient patient, Date date) {
        this.doctor = doctor;
        this.patient = patient;
        this.date = date;
    }

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public Doctor getDoctor() { return doctor; }
    public void setDoctor(Doctor doctor) { this.doctor = doctor; }

    public Patient getPatient() { return patient; }
    public void setPatient(Patient patient) { this.patient = patient; }

    public Date getDate() { return date; }
    public void setDate(Date date) { this.date = date; }
}
```

doctor.java

```
package com.hospital.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "doctors")
public class Doctor {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "name", nullable = false)
    private String name;

    @Column(name = "specialization", nullable = false)
    private String specialization;

    public Doctor() {}

    public Doctor(String name, String specialization) {
        this.name = name;
        this.specialization = specialization;
    }

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getSpecialization() { return specialization; }
    public void setSpecialization(String specialization) {
        this.specialization = specialization;
    }
}
```

Nurse.java

```
package com.hospital.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "nurses")
public class Nurse {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "name", nullable = false)
    private String name;

    @Column(name = "department", nullable = false)
    private String department;
```

```

public Nurse() {}

public Nurse(String name, String department) {
    this.name = name;
    this.department = department;
}

public int getId() { return id; }
public void setId(int id) { this.id = id; }

public String getName() { return name; }
public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }
public void setDepartment(String department) { this.department =
department; }
}

```

Patients.java

```

package com.hospital.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "patients")
public class Patient {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "name", nullable = false)
    private String name;

    public Patient() {}

    public Patient(String name) {
        this.name = name;
    }

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}

```

HospitalManagementSystem.java

```
package com.hospital.main;

import com.hospital.service.AdminService;
import com.hospital.service.PatientService;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class HospitalManagementSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AdminService adminService = new AdminService();
        PatientService patientService = new PatientService();

        while (true) {
            System.out.println("\nHospital Management System");
            System.out.println("1. Add Doctor");
            System.out.println("2. Add Nurse");
            System.out.println("3. View All Doctor Appointments (Admin)");
            System.out.println("4. Book Appointment (Patient)");
            System.out.println("5. Cancel Appointment (Patient)");
            System.out.println("6. View My Appointments (Patient)");
            System.out.println("7. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();
            sc.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter Doctor Name: ");
                    String docName = sc.nextLine();
                    System.out.print("Enter Specialization: ");
                    String specialization = sc.nextLine();
                    adminService.addDoctor(docName, specialization);
                    System.out.println("Doctor added successfully!");
                    break;
                case 2:
                    System.out.print("Enter Nurse Name: ");
                    String nurseName = sc.nextLine();
                    System.out.print("Enter Department: ");
                    String department = sc.nextLine();
                    adminService.addNurse(nurseName, department);
                    System.out.println("Nurse added successfully!");
                    break;
                case 3:
                    adminService.viewAllDoctorAppointments();
                    break;
                case 4:
                    System.out.print("Enter Patient ID: ");
                    int patientId = sc.nextInt();
                    System.out.print("Enter Doctor ID: ");
                    int doctorId = sc.nextInt();
                    sc.nextLine(); // Consume newline
                    System.out.print("Enter Appointment Date (yyyy-MM-dd): ");
            }
        }
    }
}
```

```
");  
        String dateStr = sc.nextLine();  
        try {  
            Date date = new SimpleDateFormat("yyyy-MM-  
dd").parse(dateStr);  
            patientService.bookAppointment(doctorId, patientId,  
date); //  Fixed here  
            System.out.println("Appointment booked  
successfully!");  
        } catch (Exception e) {  
            System.out.println("Invalid date format! Please use  
yyyy-MM-dd.");  
        }  
        break;  
    case 5:  
        System.out.print("Enter Appointment ID to Cancel: ");  
        int appointmentId = sc.nextInt();  
        patientService.cancelAppointment(appointmentId);  
        System.out.println("Appointment cancelled  
successfully!");  
        break;  
    case 6:  
        System.out.print("Enter Patient ID: ");  
        int patId = sc.nextInt();  
        patientService.viewAppointments(patId);  
        break;  
    case 7:  
        System.out.println("Exiting...");  
        sc.close();  
        System.exit(0);  
        break;  
    default:  
        System.out.println("Invalid choice! Try again.");  
    }  
}  
}
```

AdminService.java

```
package com.hospital.service;

import com.hospital.entity.Appointment;
import com.hospital.entity.Doctor;
import com.hospital.entity.Nurse;
import com.hospital.entity.Patient;
import com.hospital.util.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

import java.util.Date;
import java.util.List;

public class AdminService {
```

```
public void addDoctor(String name, String specialization) {
    Session session = HibernateUtil.getSessionFactory().openSession();
    Transaction tx = session.beginTransaction();

    Doctor doctor = new Doctor(name, specialization);
    session.save(doctor);

    tx.commit();
    session.close();
}

public void addNurse(String name, String department) {
    Session session = HibernateUtil.getSessionFactory().openSession();
    Transaction tx = session.beginTransaction();

    Nurse nurse = new Nurse(name, department);
    session.save(nurse);

    tx.commit();
    session.close();
}

public void viewAllDoctorAppointments() {
    Session session = HibernateUtil.getSessionFactory().openSession();
    List<Appointment> appointments = session.createQuery("FROM
Appointment", Appointment.class).list();

    if (appointments.isEmpty()) {
        System.out.println("No appointments found.");
    } else {
        for (Appointment appointment : appointments) {
            System.out.println("Appointment ID: " + appointment.getId() +
", Doctor: " + appointment.getDoctor().getName() + ", Patient: " +
appointment.getPatient().getName() + ", Date: " + appointment.getDate());
        }
    }
    session.close();
}
}
```

PatientService.java

```
package com.hospital.service;

import com.hospital.entity.Appointment;
import com.hospital.entity.Doctor;
import com.hospital.entity.Patient;
import com.hospital.util.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

public class PatientService {

    public void bookAppointment(int doctorId, int patientId, Date date) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction tx = session.beginTransaction();

        try {

            Date today = new Date();
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            Date todayWithoutTime = sdf.parse(sdf.format(today));

            if (date.before(todayWithoutTime)) {
                System.out.println(" Error: Appointment date must be in the
future.");
                tx.rollback();
                return;
            }

            Doctor doctor = session.get(Doctor.class, doctorId);
            if (doctor == null) {
                System.out.println("Error: Doctor not found!");
                tx.rollback();
                return;
            }

            Patient patient = session.get(Patient.class, patientId);
            if (patient == null) {
                System.out.println(" Patient not found! Adding new
patient... ");
                patient = new Patient();
                patient.setName("Patient " + patientId);
                session.save(patient); // Correctly save new patient
                session.flush(); // Ensure it's saved immediately
                System.out.println(" New patient added: " +
patient.getName());
            }
        }
    }
}
```

```

        Query<Appointment> query = session.createQuery(
            "FROM Appointment a WHERE a.doctor.id = :doctorId AND
a.date = :date", Appointment.class);
        query.setParameter("doctorId", doctorId);
        query.setParameter("date", date);
        List<Appointment> existingAppointments = query.list();

        if (!existingAppointments.isEmpty()) {
            System.out.println(" Error: Doctor is already booked on this
date!");
            tx.rollback();
            return;
        }

        Appointment appointment = new Appointment(doctor, patient, date);
        session.persist(appointment);
        tx.commit();
        System.out.println("☑ Appointment booked successfully for " +
sdf.format(date) + "!");

    } catch (Exception e) {
        if (tx != null) tx.rollback();
        System.out.println("Error: " + e.getMessage());
    } finally {
        session.close();
    }
}

public void cancelAppointment(int appointmentId) {
    Session session = HibernateUtil.getSessionFactory().openSession();
    Transaction tx = session.beginTransaction();

    try {
        Appointment appointment = session.get(Appointment.class,
appointmentId);
        if (appointment == null) {
            System.out.println(" Error: Appointment not found!");
            tx.rollback();
            return;
        }

        session.remove(appointment);
        tx.commit();
        System.out.println(" Appointment ID " + appointmentId + " "
canceled successfully!");
    } catch (Exception e) {
        if (tx != null) tx.rollback();
        System.out.println(" Error: " + e.getMessage());
    } finally {
        session.close();
    }
}

```

```

        }

    }

    public void viewAppointments(int patientId) {
        Session session = HibernateUtil.getSessionFactory().openSession();

        try {
            Query<Appointment> query = session.createQuery(
                "FROM Appointment a WHERE a.patient.id = :patientId",
Appointment.class);
            query.setParameter("patientId", patientId);
            List<Appointment> appointments = query.list();

            if (appointments.isEmpty()) {
                System.out.println(" No appointments found for patient ID: " +
+ patientId);
                return;
            }

            System.out.println("Your Appointments:");
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            for (Appointment a : appointments) {
                System.out.println(" Appointment ID: " + a.getId() +
                    " | Doctor: " + a.getDoctor().getName() +
                    " | Specialization: " +
a.getDoctor().getSpecialization() +
                    " | Date: " + sdf.format(a.getDate()));
            }
        } catch (Exception e) {
            System.out.println(" Error: " + e.getMessage());
        } finally {
            session.close();
        }
    }
}

```

HibernateUtil.java

```

package com.hospital.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory =
buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            return new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }
}

```

```
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

hibernate.cfg.xml

```
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/hospital_db</prop
erty>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">pass123</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>

        <!-- Entity Mappings -->
        <mapping class="com.hospital.entity.Doctor"/>
        <mapping class="com.hospital.entity.Patient"/>
        <mapping class="com.hospital.entity.Appointment"/>
        <mapping class="com.hospital.entity.Nurse"/>
    </session-factory>
</hibernate-configuration>
```

Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" ...  
  
Hospital Management System  
1. Add Doctor  
2. Add Nurse  
3. View All Doctor Appointments (Admin)  
4. Book Appointment (Patient)  
5. Cancel Appointment (Patient)  
6. View My Appointments (Patient)  
7. Exit  
Enter your choice: 1  
Enter Doctor Name: Kunal Ekare  
Enter Specialization: mbbs  
Mar 30, 2025 7:01:19 AM org.hibernate.Version logVersion  
INFO: HHH000412: Hibernate ORM core version 6.0.0.Final  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure  
WARN: HHH0001002: Using built-in connection pool (not intended for production use)  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator  
INFO: HHH10001005: Loaded JDBC driver class: com.mysql.cj.jdbc.Driver  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator  
INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://localhost:3306/hospital_db]  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator  
INFO: HHH10001001: Connection properties: {password=****, user=root}  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator  
INFO: HHH10001003: Autocommit mode: false  
Mar 30, 2025 7:01:21 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>  
INFO: HHH10001115: Connection pool size: 20 (min=1)  
Mar 30, 2025 7:01:23 AM org.hibernate.engine.jdbc.dialect.internal.DialectFactoryImpl logSelectedDialect  
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect  
  
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@5b39e  
Mar 30, 2025 7:01:26 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
Hibernate: insert into doctors (name, specialization) values (?, ?)  
Doctor added successfully!  
  
Hospital Management System  
1. Add Doctor  
2. Add Nurse  
3. View All Doctor Appointments (Admin)  
4. Book Appointment (Patient)  
5. Cancel Appointment (Patient)  
6. View My Appointments (Patient)  
7. Exit  
Enter your choice: 2  
Enter Nurse Name: pratiksha  
Enter Department: healthcare  
Hibernate: insert into nurses (department, name) values (?, ?)  
Nurse added successfully!  
  
Hospital Management System  
1. Add Doctor  
2. Add Nurse  
3. View All Doctor Appointments (Admin)  
4. Book Appointment (Patient)  
5. Cancel Appointment (Patient)  
6. View My Appointments (Patient)  
7. Exit  
Enter your choice: 3
```

```
1. Add Doctor
2. Add Nurse
3. View All Doctor Appointments (Admin)
4. Book Appointment (Patient)
5. Cancel Appointment (Patient)
6. View My Appointments (Patient)
7. Exit
Enter your choice: 3
Mar 30, 2025 7:01:59 AM org.hibernate.jpa.internal.LegacySpecHelper getValue
WARN: HHH900000021: Encountered deprecated setting [javax.persistence.lock.timeout], use [jakarta.persistence.lock.timeout] instead
Hibernate: select a1_0.id,a1_0.date,a1_0.doctor_id,a1_0.patient_id from appointments a1_0
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Appointment ID: 2, Doctor: Dr. Alice Smith, Patient: Sarah Williams, Date: 2025-04-05
Appointment ID: 3, Doctor: Dr. Robert Brown, Patient: David Martinez, Date: 2025-04-10
Appointment ID: 4, Doctor: raju, Patient: Patient 4, Date: 2025-04-20
Appointment ID: 6, Doctor: Mitesh, Patient: Patient 6, Date: 2025-06-03
Appointment ID: 7, Doctor: raju, Patient: Patient 7, Date: 2025-05-03
Appointment ID: 8, Doctor: Kunal Ekare, Patient: Patient 8, Date: 2025-06-03
```

```
Hospital Management System
1. Add Doctor
2. Add Nurse
3. View All Doctor Appointments (Admin)
4. Book Appointment (Patient)
5. Cancel Appointment (Patient)
6. View My Appointments (Patient)
7. Exit
Enter your choice: 4
Enter Patient ID: 09
Enter Doctor ID: 7
Enter Appointment Date (yyyy-MM-dd): 2025-04-20
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
  Patient not found! Adding new patient..
Hibernate: insert into patients (name) values (?)
New patient added: Patient 9
Mar 30, 2025 7:02:35 AM org.hibernate.jpa.internal.LegacySpecHelper getValue
WARN: HHH900000021: Encountered deprecated setting [javax.persistence.lock.timeout], use [jakarta.persistence.lock.timeout] instead
Hibernate: select a1_0.id,a1_0.date,a1_0.doctor_id,a1_0.patient_id from appointments a1_0 where a1_0.doctor_id=? and a1_0.date=?
Hibernate: insert into appointments (date, doctor_id, patient_id) values (?, ?, ?)
  Appointment booked successfully for 2025-04-20!
Appointment booked successfully!
```

```
Hospital Management System
1. Add Doctor
2. Add Nurse
3. View All Doctor Appointments (Admin)
4. Book Appointment (Patient)
5. Cancel Appointment (Patient)
6. View My Appointments (Patient)
7. Exit
Enter your choice: 5
Enter Appointment ID to Cancel: 9
Hibernate: select a1_0.id,a1_0.date,d1_0.id,d1_0.name,d1_0.specialization,p1_0.id,p1_0.name from appointments a1_0 join doctors d1_0 on d1_0.id=a1_0.doctor_id join patient
Hibernate: delete from appointments where id=?
  Appointment ID 9 canceled successfully!
Appointment cancelled successfully!
```

```
Hospital Management System
1. Add Doctor
2. Add Nurse
3. View All Doctor Appointments (Admin)
4. Book Appointment (Patient)
5. Cancel Appointment (Patient)
6. View My Appointments (Patient)
7. Exit
Enter your choice: 6
Enter Patient ID: 7
Hibernate: select a1_0.id,a1_0.date,a1_0.doctor_id,a1_0.patient_id from appointments a1_0 where a1_0.patient_id=?
Mar 30, 2025 7:03:26 AM org.hibernate.jpa.internal.LegacySpecHelper getValue
WARN: HHH90000021: Encountered deprecated setting [javax.persistence.lock.timeout], use [jakarta.persistence.lock.timeout] instead
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Your Appointments:
Appointment ID: 7 | Doctor: raju | Specialization: mbbs | Date: 2025-05-03
```

```
Hibernate: select d1_0.id,d1_0.name,d1_0.specialization from doctors d1_0 where d1_0.id=?
Hibernate: select p1_0.id,p1_0.name from patients p1_0 where p1_0.id=?
Your Appointments:
Appointment ID: 7 | Doctor: raju | Specialization: mbbs | Date: 2025-05-03

Hospital Management System
1. Add Doctor
2. Add Nurse
3. View All Doctor Appointments (Admin)
4. Book Appointment (Patient)
5. Cancel Appointment (Patient)
6. View My Appointments (Patient)
7. Exit
Enter your choice: 7
Exiting...

Process finished with exit code 0
```

Note:(rar file is attached in Google classroom) all file and code inside it.

Database :

```

mysql> select * from doctors;
+----+-----+-----+
| id | name           | specialization |
+----+-----+-----+
| 1  | Dr. John Doe   | Cardiology    |
| 2  | Dr. Alice Smith | Neurology     |
| 3  | Dr. Robert Brown| Orthopedics   |
| 4  | Dr. John        | Cardiology    |
| 5  | kunal           | ekare         |
| 6  | dcoss           | ccsv          |
| 7  | kunal           | mbbs          |
| 8  | Kunal Ekare    | mbbs          |
| 9  | Mitesh          | mbbs          |
| 10 | raju            | mbbs          |
| 11 | soumya          | mbbs          |
| 12 | kunal           | mbbs          |
| 13 | kunal ekare    | mbbs          |
| 14 | KunalEkare     | mbbs(nagpur) |
| 15 | Kunal Ekare    | mbbs          |
| 16 | Kunal Ekare    | mbbs          |
+----+-----+-----+
16 rows in set (0.00 sec)

mysql> select * from patients;
+----+-----+
| id | name           |
+----+-----+
| 1  | Michael Johnson|
| 2  | Sarah Williams |
| 3  | David Martinez |
| 4  | Patient 4       |
| 5  | Patient 25      |
| 6  | Patient 6       |
| 7  | Patient 7       |
| 8  | Patient 8       |
| 9  | Patient 9       |
+----+-----+
9 rows in set (0.00 sec)

mysql> select * from appointments;
+----+-----+-----+-----+
| id | doctor_id | patient_id | date      |
+----+-----+-----+-----+
| 2  | 2         | 2         | 2025-04-05 |
| 3  | 3         | 3         | 2025-04-10 |
| 4  | 10        | 4         | 2025-04-20 |
| 6  | 9         | 6         | 2025-06-03 |
| 7  | 10        | 7         | 2025-05-03 |
| 8  | 8         | 8         | 2025-06-03 |
| 9  | 7         | 9         | 2025-04-20 |
+----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> select * from appointments;
+----+-----+-----+-----+
| id | doctor_id | patient_id | date      |
+----+-----+-----+-----+
| 2  | 2         | 2         | 2025-04-05 |
| 3  | 3         | 3         | 2025-04-10 |
| 4  | 10        | 4         | 2025-04-20 |
| 6  | 9         | 6         | 2025-06-03 |
| 7  | 10        | 7         | 2025-05-03 |
| 8  | 8         | 8         | 2025-06-03 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Conclusion :

The **Hospital Management System** successfully implements **Hibernate** for efficient data persistence in a **menu-driven program**. The system allows **Admins** to manage **Doctors and Nurses** while enabling **Patients to book, cancel, and view appointments**. By leveraging Hibernate's ORM capabilities, the system ensures **seamless database interactions, efficient querying, and structured data handling**, making it a robust solution for hospital management.