|  |  |
| --- | --- |
|  |  |

**Hospital Appointment Management System**

**A PROJECT REPORT**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELORS OF COMPUTER APPLICATION**

|  |  |
| --- | --- |
| ***Submitted By:***  **Vishesh**  **GU23R1174** | ***Submitted To:*** |

**Department of CS & IT**

**Avviare Educational Hub**

**Noida, UP – 201301, Indi**

**ABSTRACT**

Hospital Appointment Management System is a Java and MySQL application developed with JDBC connectivity, intended to automate the scheduling process and ensure data integrity. It provides an option for patients to create accounts and make appointments with doctors, based on a validation system which assures that both patient and doctor IDs exist prior to execution. When a patient chooses a doctor and a desired date, the system verifies the availability of the doctor in the database. When the doctor is available on the date, the appointment is made successfully; otherwise, the user is informed about the unavailability. The system is implemented on a formal database using specific tables for patients, doctors, and appointments and provides basic functions like viewing, updating, and deleting records. By automating these, the application decreases waiting time, prevents scheduling conflicts, reduces manual errors, and improves the overall user experience.

**ACKNOWLEDGEMENT**

I would like to extend my sincere thanks to my peers and faculty for their support and encouragement throughout the process of developing this project. I am particularly grateful to Ms. Anjali mam for providing useful inputs and steady encouragement throughout the process. I am deeply thankful to Ms. Kanika Singh, Operations Director at Avviare Educational Hub, Noida, for providing me with the privilege to work on this large project. I am also extremely thankful to Ms. Anjali, IT Department Head at Avviare Educational Hub, Noida, whose constant support and encouragement have greatly helped in the successful completion of this project. I also appreciate the intellectual assistance offered by other members of the faculty of the IT Department at Avviare Educational Hub, Noida. Finally, I am grateful to all those who helped, directly or indirectly, towards the successful submission of this report.

**Names: Vishesh**

**Date:**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **F. No.** | **Figure Description** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **T. No.** | **Table Description** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Content** | **Page No.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Format for Major Project Report**

**Title page Page**

*Abstract i*

*Acknowledgement ii*

*List of Figures iii*

*List of Tables iv*

*Table of Contents v*

**Chapter 1 Introduction**

1.1 Introduction to Project

1.2 Technical Requirements (Hardware and Software Used)

1.3 Objectives

**Chapter 2. Implementation**

2.1 Introduction to Languages, IDE’s, Tools and Technologies used for Implementation

**Chapter 3. Results, Flowcharts and Outputs**

3.1 Flowcharts/Diagram of the Program

3.2 Snapshots of Program and Output

**Chapter 4. Conclusion and Future Scope**

**References and Bibliography**

# Chapter 1: Introduction

## 1.1 Introduction to Project

The hospital Appointment Management System is a Java application with a MySQL backend designed to automate and manage appointment scheduling between doctors and patients effectively. It allows the creation of user accounts and permits patients to schedule appointments based on real-time availability of doctors, thus reducing conflicts in scheduling and maximizing time utilization. The system uses a strong validation process that ensures the presence of both patient and doctor IDs in their respective tables in the database before it allows any booking to proceed. Having ensured the IDs, it goes ahead to verify the availability of the doctor for the specified date. The appointment is only booked when all stipulations—correct patient and doctor IDs and free time slot—are fulfilled. If any of these verifications fail, the system declines the appointment request in order to preserve data integrity and scheduling integrity. It also keeps detailed records of patients and physicians to aid effective hospital administration.

## 1.2 Technical Requirements

Hardware:  
- Standard PC or Laptop  
- Minimum 4 GB RAM  
- 500 MB Free Disk Space  
  
Software:  
- Java JDK 8 or above  
- MySQL Server  
- JDBC Driver  
- Visual Studio Code IDE

## 1.3 Objectives

# To develop an efficient appointment management system.

# To store patient and doctor data securely.

# To allow appointment creation only when the doctor is available.

# To reduce manual errors and scheduling conflicts.

# To develop a user-friendly interface for appointment management.

# To implement a database for storing patient and doctor information.

# To ensure data integrity through validation checks during appointment booking.

# To enhance the efficiency of appointment scheduling in hospitals.

# Chapter 2: Implementation

## 2.1 Tools, Languages, IDEs and Technologies Used

Programming Language: Java

Database: MySQL

Database Connectivity: JDBC

IDE: Visual Studio Code

Version Control: Git

**System Components:**

**Patient Module:**

* Add Patient: Functionality to register new patients by entering their details such as name, age, gender, and generating a unique patient ID.
* Update Patient: Allows modification of existing patient records to keep information current.
* View Patient: Enables viewing of patient details and appointment history.
* Delete Patient: Functionality to remove a patient record from the system.
* Check Patient: Validates if a patient exists in the database using their ID.

**Doctor Module:**

* View Doctor: Enables viewing of doctor details and their availability status.
* Check Availability: Checks if a doctor is available for appointments on a specific date.

**Appointment Module:**

* Create Appointment: Allows patients to book an appointment with a doctor. The system checks:

1. If the patient ID exists.
2. If the doctor ID exists.
3. If the doctor is available on the requested date.
4. If all conditions are met, the appointment is booked; otherwise, an error message is displayed.

* View Appointment: Enables patients and doctors to view scheduled appointments.

**Database Management:**

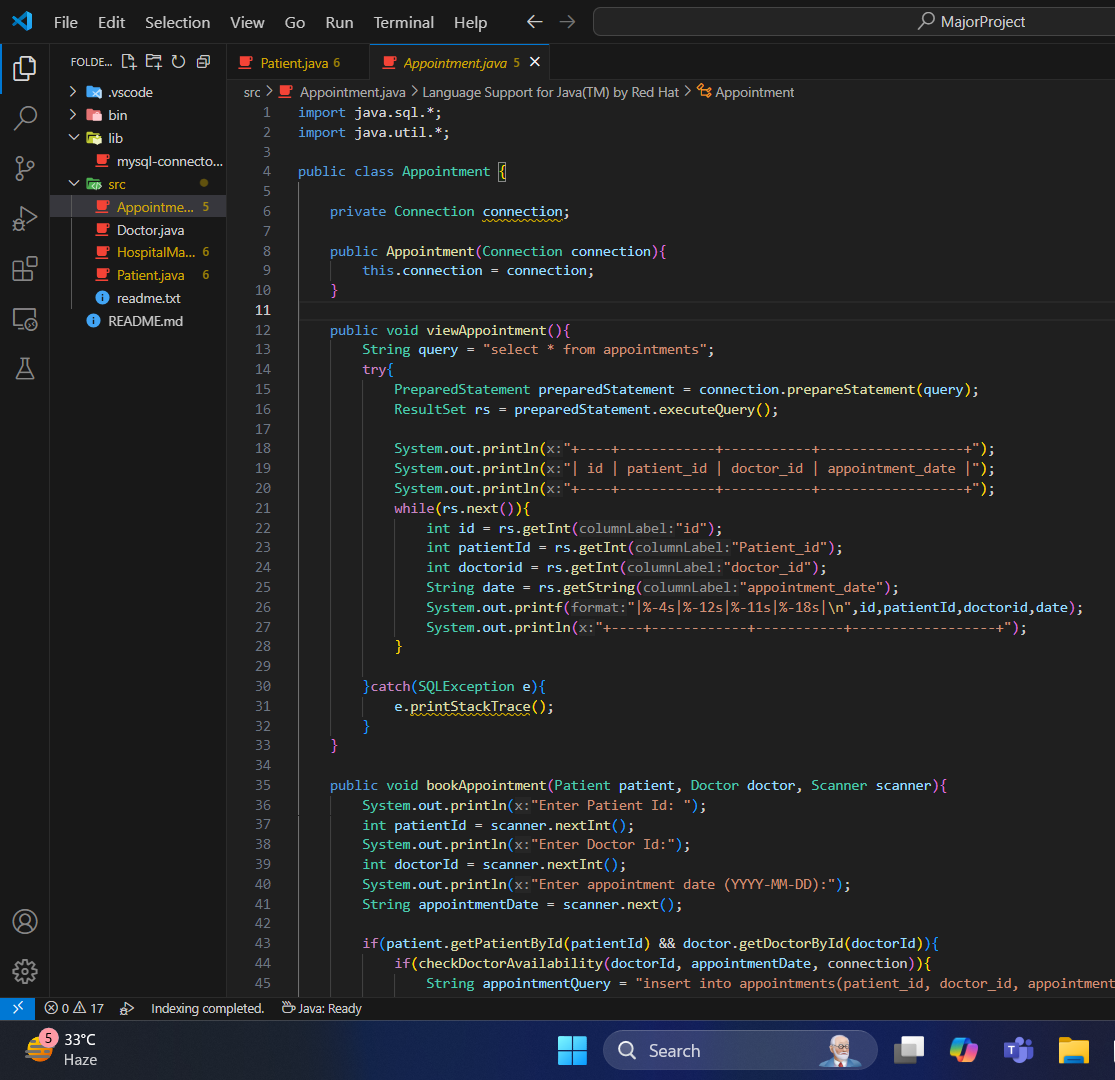
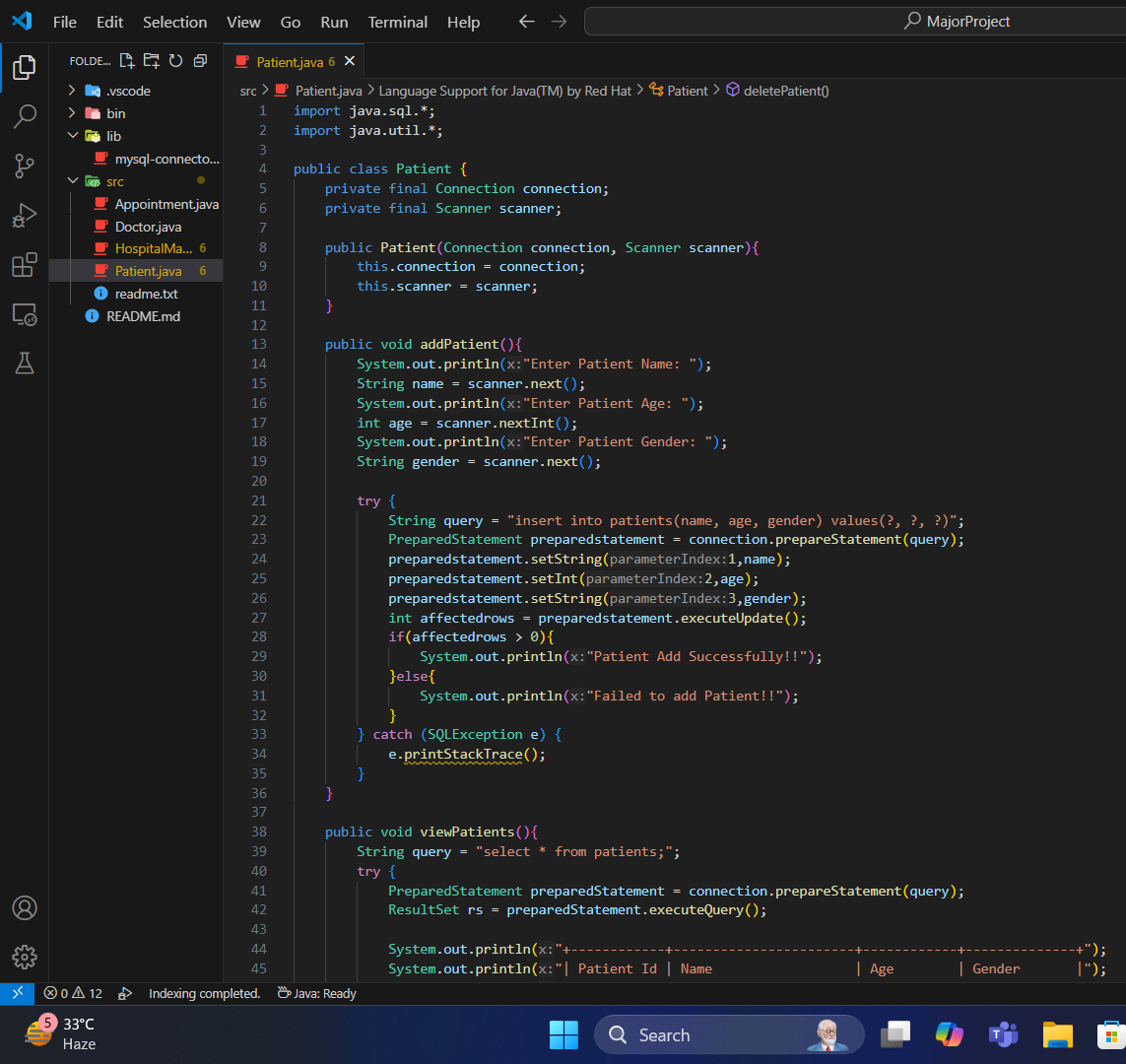
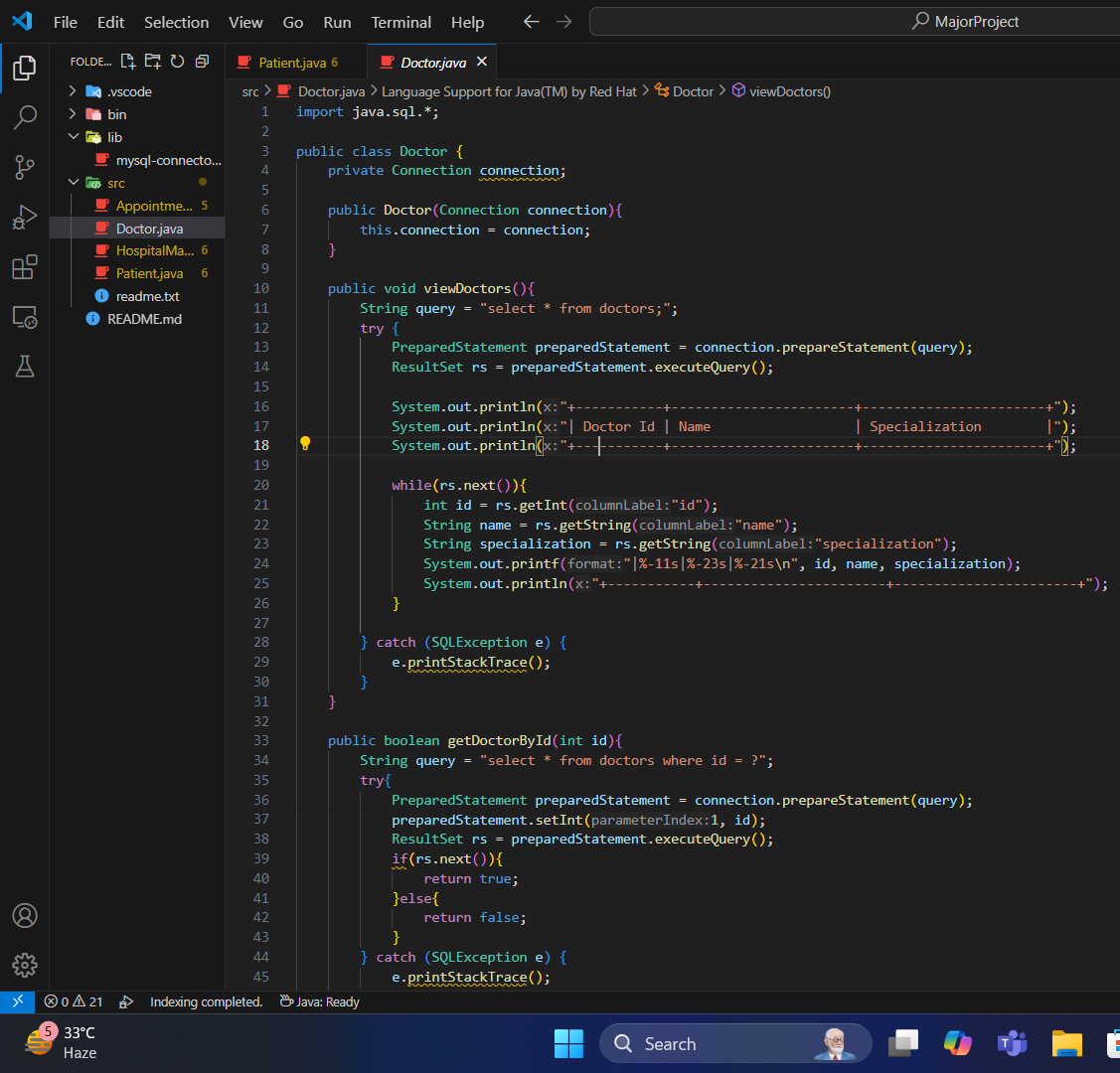
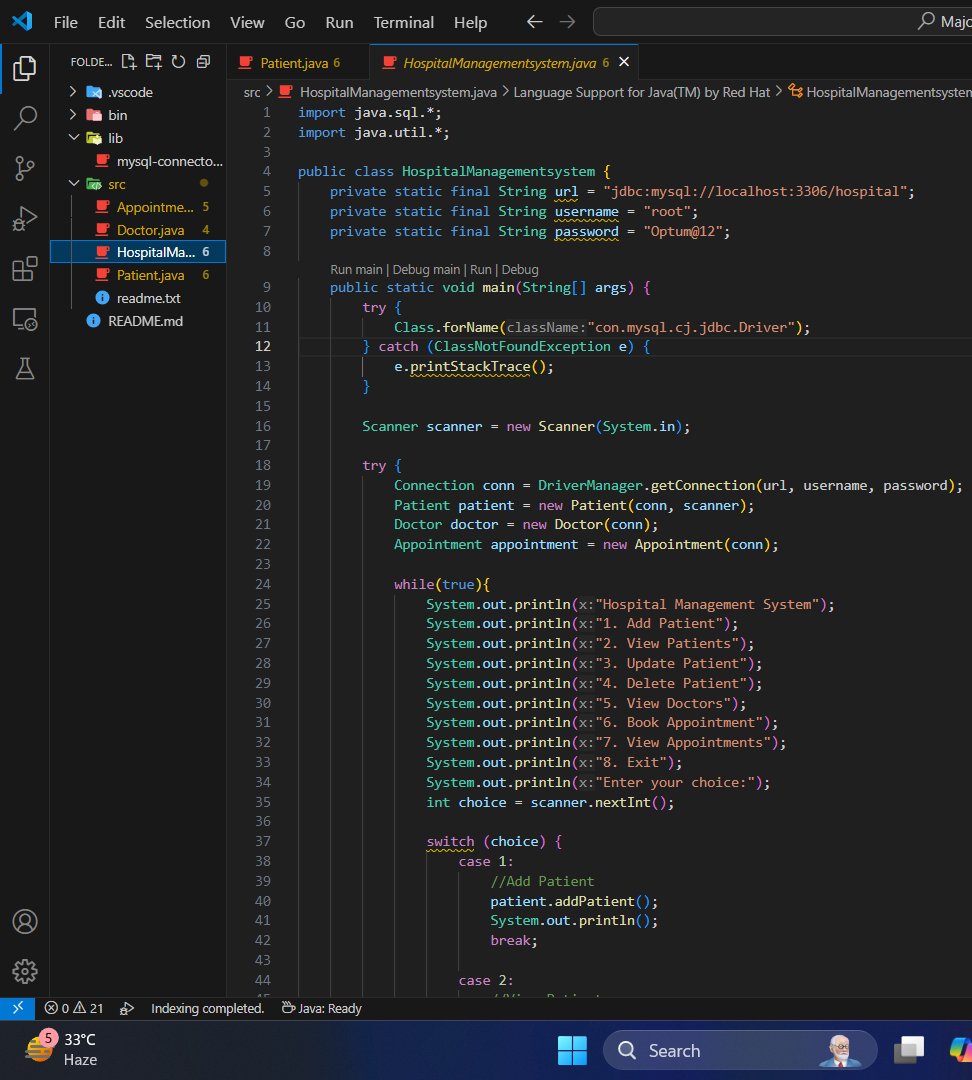
* Patient Table: Stores patient information including ID, name, age, and gender.
* Doctor Table: Stores doctor information including ID, name, and specialization.
* Appointment Table: Manages appointment records linking patient IDs and doctor IDs with appointment details.

# Chapter 3: Results, Flowcharts and Outputs

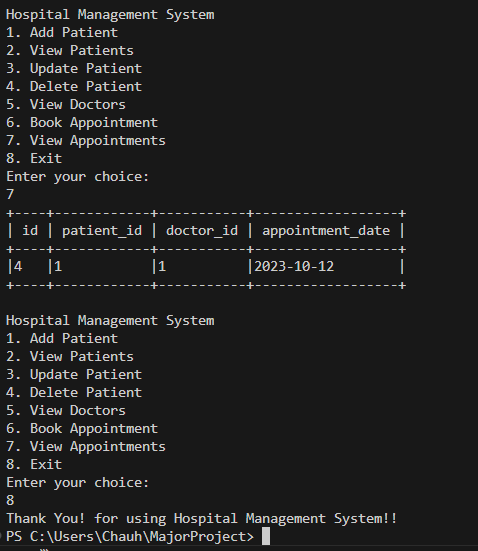
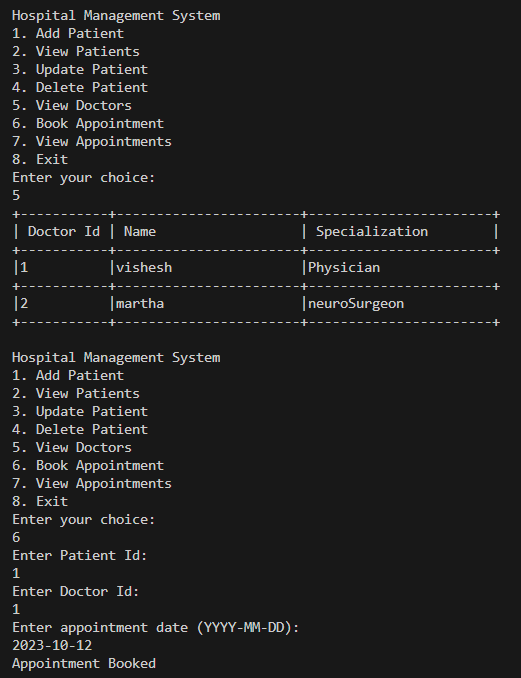
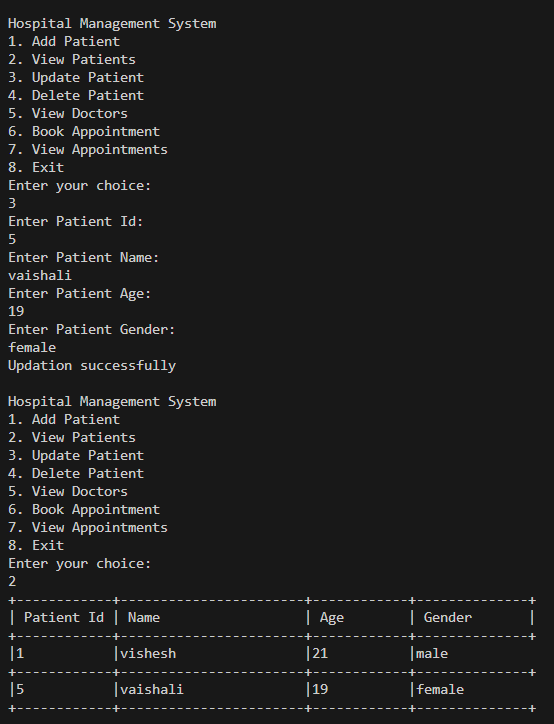
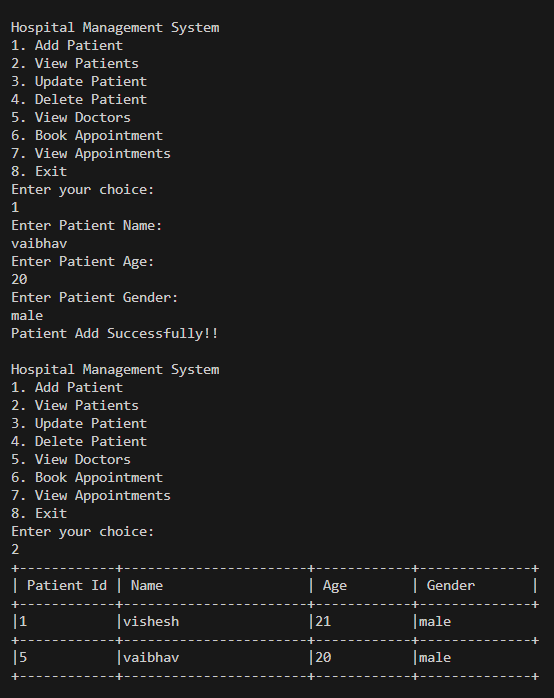
## 3.1 Flowcharts/Diagram of the Program

## 3.2 Snapshots of Program and Output

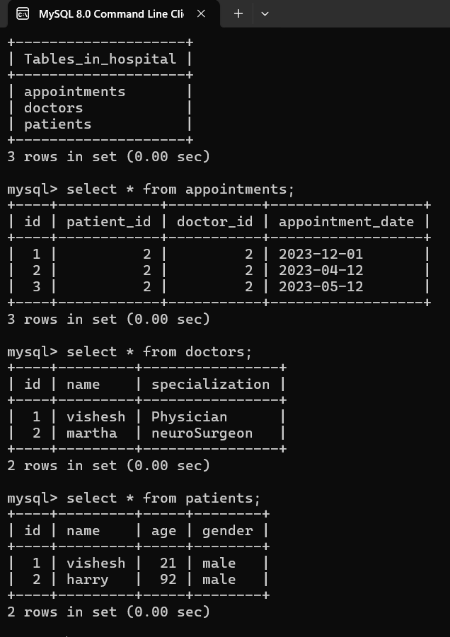
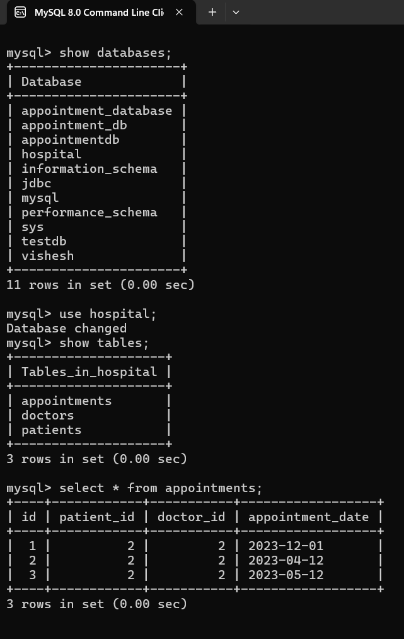
**1. Java Programs:**



**2. Output :**



**3. Database :**



# Chapter 4: Conclusion and Future Scope

# The Appointment Management System is an effective solution for managing patient-doctor appointments in a reliable manner, minimizing manual work burden and maximizing operational effectiveness. It maintains both data consistency and appointment schedule integrity through simple database operations. The project is an effective solution to certain critical issues in hospital scheduling due to automatic validation of patient and doctor IDs and real-time doctor availability checks, thus ensuring conflict-free and hassle-free booking. The system optimizes the total user experience while reducing dependence on manual processes. For the future, possible enhancements include the addition of SMS and email notifications for appointment reminders, role-based access control for medical and administrative personnel, integration with hospital billing systems, and the implementation of a secure login authentication scheme. Also, adding features like online payment processing, telemedicine support, and web-based interfaces to the system would further enhance its functionality and accessibility.

# References and Bibliography

Java Documentation

MySQL Reference Manual

JDBC API Guide