EXP NO: **DATE:**

PO–PSO Mapping Table

P01	P02	P03	P04	P05	P06	P07	PS01	PS02
3	3	3	2	3	2	1	3	3

PO Justification Table

PO Code	Program Outcome Description	Relevance (1–3)	Justification
PO1	Engineering Knowledge	3	Applies fundamental computing knowledge, OOP concepts, and JDBC to develop a practical IT Ticketing Support System.
PO2	Problem Analysis	3	Identifies and automates real-world IT issue tracking, reducing manual work and human error.
PO3	Design/Development of Solutions	3	Designs and implements an end-to-end ticketing system using Java Swing and MySQL.
PO4	Conduct Investigations of Complex Problems	2	Analyzes database connectivity, debugging, and real-time query execution for issue resolution.
PO5	Modern Tool Usage	3	Uses modern tools such as JDBC, Swing, and MySQL to integrate GUI and backend operations.
PO6	The Engineer and Society	2	Contributes to smoother IT operations and efficient service management in organizations.
PO7	Environment and Sustainability	1	Promotes paperless ticket tracking and digital automation, reducing manual paperwork.

PSO Justification Table

PSO Code	Program Specific Outcome Description	Relevance (1–3)	Justification
PSO1	Apply computing knowledge and programming skills to design and develop effective software solutions.	3	Demonstrates Java Swing and JDBC-based software development for IT issue management.
PSO2	Analyze, model, and implement database-driven applications for real-world automation.	3	Implements real-time database connectivity with MySQL to manage ticket data dynamically.
PSO3	Demonstrate professional ethics, communication skills, and project management abilities in software development.	2	Promotes documentation, responsible data handling, and structured project organization.

IT Ticketing Support System

INTRODUCTION:

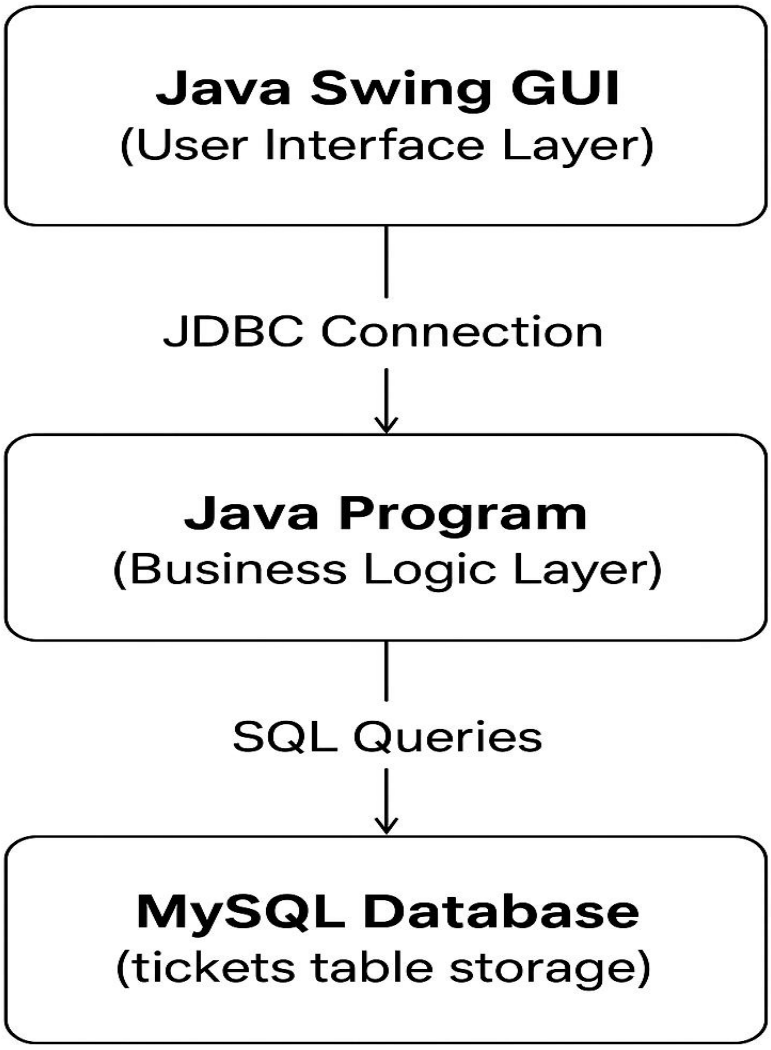
The IT Ticketing Support System is a simple and efficient tool developed using Java Swing and MySQL to automate the process of creating, managing, and resolving IT support tickets. It helps organizations record, track, and close issues raised by users efficiently. The system enables users to log complaints, monitor ticket status, and close them once resolved. Java Swing provides a user-friendly graphical interface, while MySQL serves as the backend database for persistent data storage. JDBC connectivity ensures smooth communication between the application and database. This system demonstrates core Object-Oriented Programming principles such as encapsulation, abstraction, and inheritance. The project offers a complete automation model for IT issue management and helps reduce manual workload.

PROJECT DESCRIPTION:

The IT Ticketing Support System is designed to streamline the management of technical issues in an organization. Users can easily create new tickets by entering their name and issue details. The application stores this information in a MySQL database through JDBC connectivity. Tickets can be viewed along with their status (OPEN or CLOSED), and users can close tickets once resolved. The system ensures data accuracy and reliability by managing real-time updates to the database. Java Swing components such as JFrame, JLabel, JTextArea, JButton, and JOptionPane are used for interactive GUI design. The application's architecture is divided into three layers – user interface, business logic, and database layer. This modular design makes it easy to extend and maintain. The project reflects the effective use of OOP concepts and database integration to solve real-world IT support challenges.

SYSTEM ARCHITECTURE:

The following diagram illustrates the layered architecture of the IT Ticketing Support System:



CODING: FRONTEND

The complete source code of the project is given below:

```
package support;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class TicketMain extends JFrame implements ActionListener {

    JTextField userName;
    JTextArea issue;
    JButton createBtn, closeBtn, viewBtn;
    Connection con;

    TicketMain() {
        setTitle("IT Ticketing Support System");
        setSize(450, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        userName = new JTextField(12);
        issue = new JTextArea(3, 12);
        createBtn = new JButton("Create Ticket");
        closeBtn = new JButton("Close Ticket");
        viewBtn = new JButton("View Tickets");

        setLayout(new GridLayout(4, 2, 10, 10));
```

```
        add(new JLabel("User Name:"));
        add(userName);
        add(new JLabel("Issue Description:"));
        add(new JScrollPane(issue));
        add(createBtn);
        add(closeBtn);
        add(viewBtn);

        createBtn.addActionListener(this);
        closeBtn.addActionListener(this);
        viewBtn.addActionListener(this);

        connectDB();
        setVisible(true);
    }

    void connectDB() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/supportdb",
                    "root",
                    "Harshini S@2007");
            JOptionPane.showMessageDialog(this, "Database Connected!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "DB Error: " +
e.getMessage());
        }
    }

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == createBtn) {
            try {
                String user = userName.getText();
```

```
            String iss = issue.getText();
            PreparedStatement ps = con.prepareStatement(
                    "INSERT INTO tickets (user_name, issue, created_time,
status) VALUES (?, ?, NOW(), 'OPEN')");
            ps.setString(1, user);
            ps.setString(2, iss);
            int rows = ps.executeUpdate();
            if (rows > 0)
               JOptionPane.showMessageDialog(this, "Ticket Created
Successfully!");
         } catch (Exception ex) {
            JOptionPane.showMessageDialog(this, "Error: " +
ex.getMessage());
         }
      }

      if (e.getSource() == closeBtn) {
         try {
            String user = userName.getText();
            PreparedStatement ps = con.prepareStatement(
                    "UPDATE tickets SET closed_time = NOW(), status =
'CLOSED' WHERE user_name = ? AND status='OPEN'");
            ps.setString(1, user);
            int rows = ps.executeUpdate();
            if (rows > 0)
               JOptionPane.showMessageDialog(this, "Ticket Closed!");
            else
               JOptionPane.showMessageDialog(this, "No Open Ticket for
this user.");
         } catch (Exception ex) {
            JOptionPane.showMessageDialog(this, "Error: " +
ex.getMessage());
         }
      }
```

```
        if (e.getSource() == viewBtn) {
            try {
                Statement st = con.createStatement();
                ResultSet rs = st.executeQuery("SELECT * FROM tickets");
                StringBuilder data = new StringBuilder("ID\tUser\tStatus\n");
                data.append("-----------------------------\n");
                while (rs.next()) {
                    data.append(rs.getInt("ticket_id")).append("\t")
                        .append(rs.getString("user_name")).append("\t")
                        .append(rs.getString("status")).append("\n");
                }
                JTextArea area = new JTextArea(data.toString(), 15, 40);
                JOptionPane.showMessageDialog(this, new JScrollPane(area));
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, "Error: " +
ex.getMessage());
            }
        }
    }

    public static void main(String[] args) {
        new TicketMain();
    }
}
```

DATABASE QUERIES(MYSQL) BACKEND:

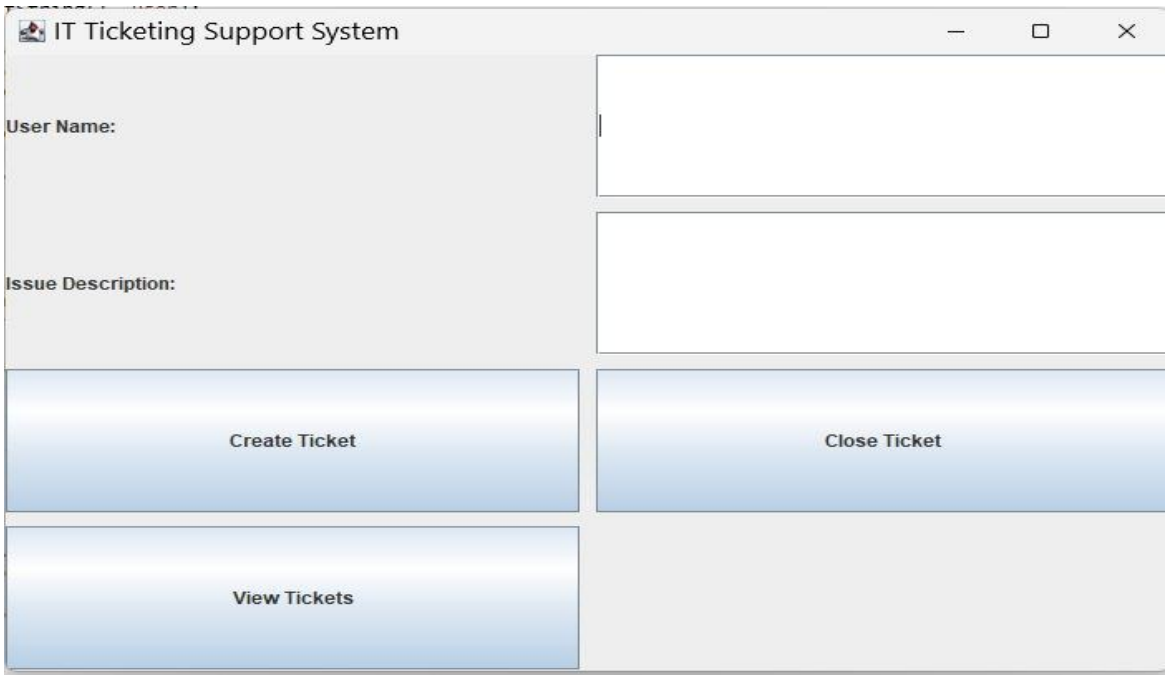
```
CREATE DATABASE supportdb;
USE supportdb;
CREATE TABLE tickets(
    ticket_id INT AUTO_INCREMENT PRIMARY KEY,
    user_name VARCHAR(50) NOT NULL,
    issue VARCHAR(200) NOT NULL,
    created_time DATETIME,
```

```
    closed_time DATETIME,
    status VARCHAR(20)
);
USE supportdb;
SELECT * FROM tickets;
```

SCREENSHOTS:

FRONTEND:

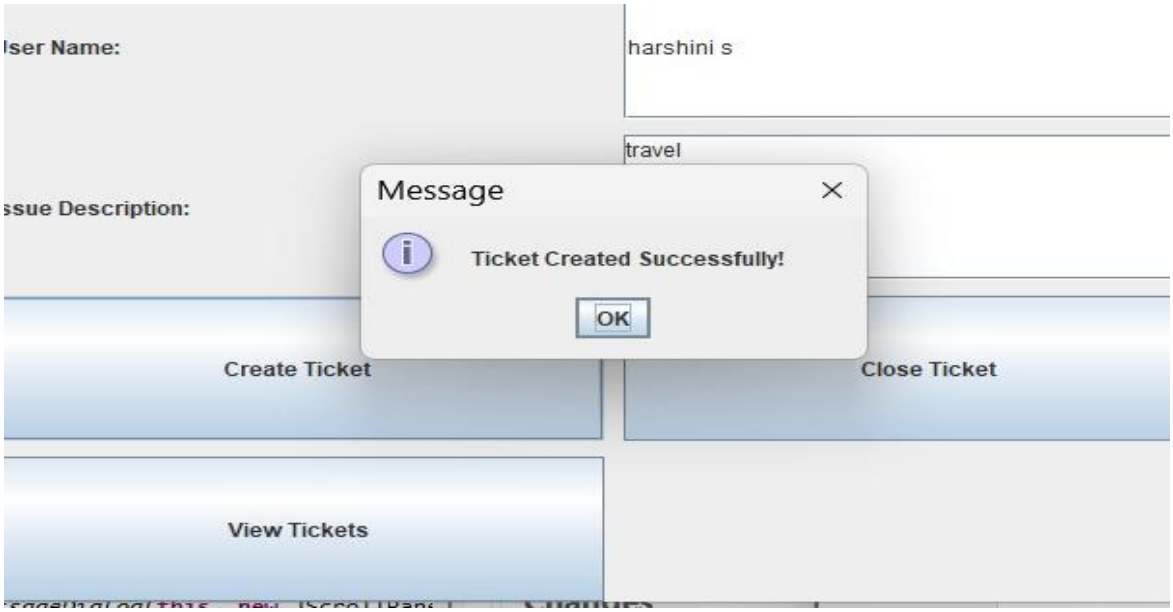
Screenshot 1: Main Interface of IT Ticketing Support System



DESCRIPTION:

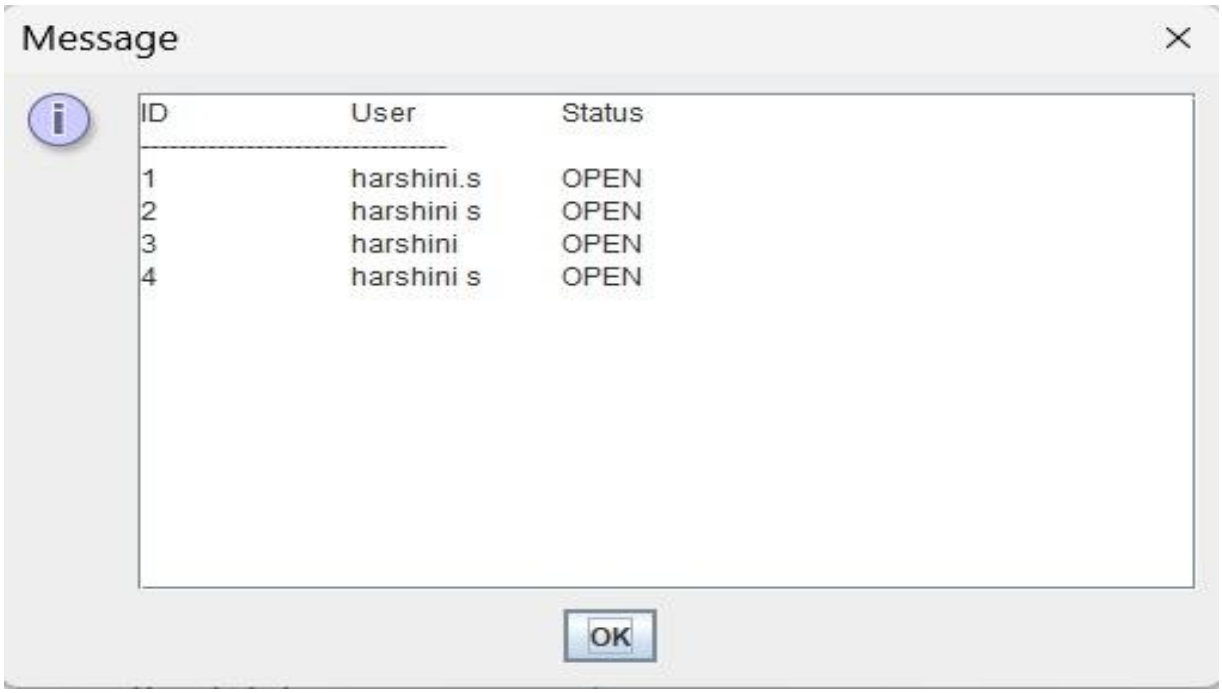
This screenshot displays the main interface of the *IT Ticketing Support System* developed using Java Swing.It allows users to enter their name and issue details, then create, view, or close tickets easily.

Screenshot 2: Ticket created successfully.



DESCRIPTION : It displays a confirmation dialog box saying “Ticket Created Successfully!” after a user submits a new ticket.This indicates successful interaction between the user interface and the database through JDBC connectivity.

screenshot3:ticketswindowview

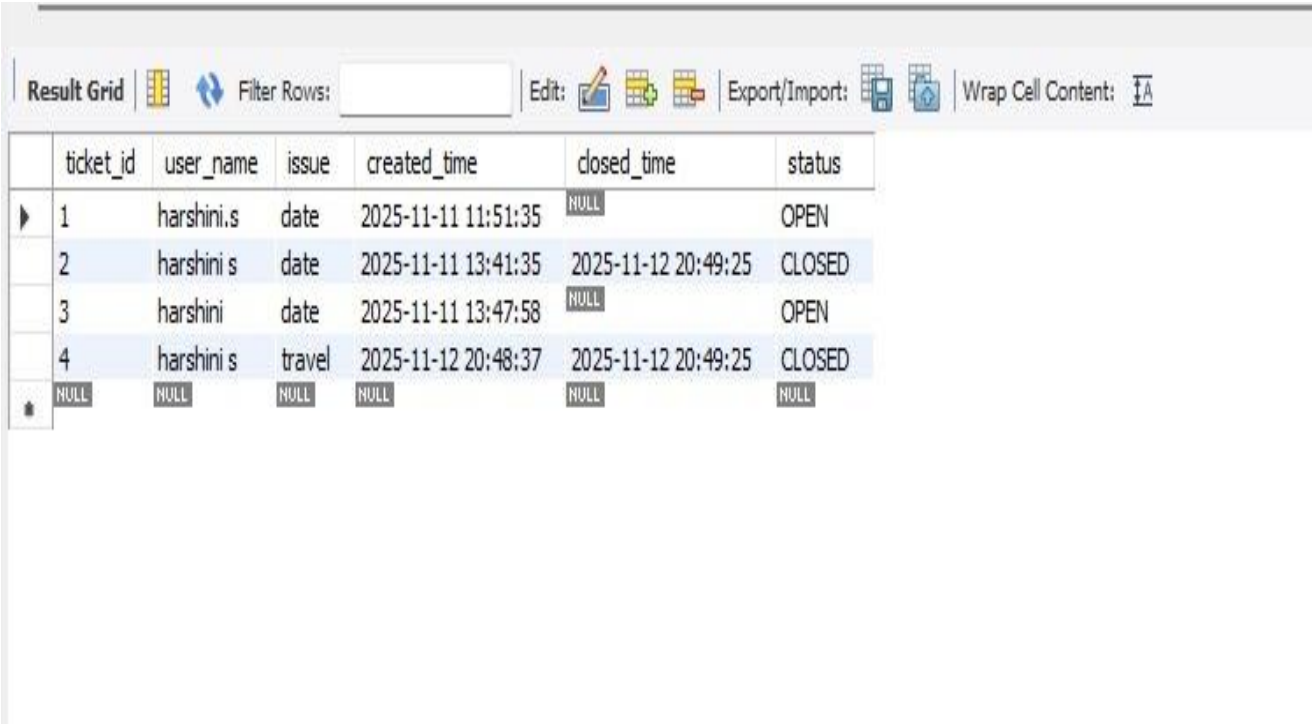


DESCRIPTION:

This screenshot shows the View Tickets window of the IT Ticketing Support System.It displays a list of tickets retrieved from the MySQL database, showing ID, User, and Status columns.All tickets are currently marked as OPEN, indicating active or unresolved issues.

BACKEND:

Screenshot 4: MySQL Database Records View



ticket_id	user_name	issue	created_time	closed_time	status
1	harshini.s	date	2025-11-11 11:51:35	NULL	OPEN
2	harshini s	date	2025-11-11 13:41:35	2025-11-12 20:49:25	CLOSED
3	harshini	date	2025-11-11 13:47:58	NULL	OPEN
4	harshini s	travel	2025-11-12 20:48:37	2025-11-12 20:49:25	CLOSED
NULL	NULL	NULL	NULL	NULL	NULL

DESCRIPTION: This screenshot shows the MySQL database table used in the IT Ticketing Support System.It stores ticket details such as ticket_id, user_name, issue, created_time, closed_time, and status.The table reflects both OPEN and CLOSED tickets, confirming successful backend data updates through JDBC.

CONCLUSION:

The IT Ticketing Support System effectively automates IT issue tracking using Java Swing for GUI and MySQL for backend storage. It allows users to create, view, and close tickets in real-time. The project demonstrates core OOP concepts, JDBC connectivity, and GUI design. The system simplifies manual operations, reduces human error, and enhances IT management efficiency. Future improvements can include adding an admin login, email notifications, search filters, and cloud-based integration. This project serves as a reliable, scalable solution for IT support automation in organizations.

FUTURE WORK:

1. **Admin Panel Integration:**
Add a separate admin module to manage users, assign tickets to IT staff, and monitor performance.

2. **Email & SMS Notifications:**
Implement automatic notifications to inform users about ticket creation, updates, or closure.

3. **Priority-Based Ticketing:**
Introduce ticket priority levels (High, Medium, Low) for better issue management and response time.

4. **Search and Filter Options:**
Allow users and admins to filter tickets by date, status, or username for easier navigation.

5. **Cloud Database Integration:**
Move the MySQL database to the cloud for remote access and centralized ticket tracking.

6. **User Authentication System:**
Add login and role-based access control to enhance data security and restrict unauthorized usage.