

## PROGRAM CODE:

```
import streamlit as st
import mysql.connector

# Connect to MySQL database
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Sh@ji2617",
    database="STD_Marks"
)
c = conn.cursor()

# Function to add student
def add_student(name, department, section):
    c.execute("INSERT INTO students (name, department, section, marks) VALUES (%s, %s, %s, %s)", (name, department, section, 0))
    conn.commit()

# Function to update marks
def update_marks(student_id, marks):
    c.execute("UPDATE students SET marks = %s WHERE id = %s", (marks, student_id))
    conn.commit()

# Function to get student by ID
def get_student_by_id(student_id):
    c.execute("SELECT * FROM students WHERE id = %s", (student_id,))
    return c.fetchone()

# Function to get all students and their marks
def get_all_students_marks():
    c.execute("SELECT * FROM students")
    return c.fetchall()

# Function to get all students in a department
def get_students_by_department(department):
    c.execute("SELECT * FROM students WHERE department = %s", (department,))
    return c.fetchall()

# Frontend code with Streamlit
def main():
    st.title("Student Result Management System")

    user_type = st.radio("Select User Type", ("Faculty", "Student"))

    if user_type == "Faculty":
        faculty_action = st.radio("Select Action", ("Add Student", "Update Marks", "Fetch Student Details by ID", "Fetch All Students Details by Department"))
```

```

if faculty_action == "Add Student":
    st.subheader("Add Student")
    name = st.text_input("Enter Student Name")
    department = st.selectbox("Select Department", ["Computer Science", "Electrical Engineering", "Mechanical Engineering"])
    section = st.selectbox("Select Section", ["A", "B", "C", "D"])
    add_button = st.button("Add Student")
    if add_button:
        add_student(name, department, section)
        st.success("Student added successfully!")

elif faculty_action == "Update Marks":
    st.subheader("Update Marks")
    student_id = st.text_input("Enter Student ID")
    marks = st.number_input("Enter Marks", min_value=0)
    update_button = st.button("Update Marks")
    if update_button:
        update_marks(student_id, marks)
        st.success("Marks updated successfully!")

elif faculty_action == "Fetch Student Details by ID":
    st.subheader("Fetch Student Details by ID")
    student_id = st.text_input("Enter Student ID")
    fetch_button = st.button("Fetch Details")
    if fetch_button:
        student = get_student_by_id(student_id)
        if student:
            st.write(f"Name: {student[1]}, Department: {student[2]}, Section: {student[3]}, Marks: {student[4]}")
        else:
            st.warning("Student not found with the given ID.")

elif faculty_action == "Fetch All Students Details by Department":
    st.subheader("Fetch All Students Details by Department")
    department = st.selectbox("Select Department", ["Computer Science", "Electrical Engineering", "Mechanical Engineering"])
    fetch_button = st.button("Fetch Details")
    if fetch_button:
        students = get_students_by_department(department)
        if students:
            for student in students:
                st.write(f"ID: {student[0]}, Name: {student[1]}, Department: {student[2]}, Section: {student[3]}, Marks: {student[4]}")
            else:
                st.warning("No students found in the selected department.")

elif user_type == "Student":
    student_action = st.radio("Select Action", ("View My Marks",))

```

```
if student_action == "View My Marks":
    st.subheader("View My Marks")
    student_id = st.text_input("Enter Your ID")
    if student_id:
        student = get_student_by_id(student_id)
        if student:
            st.write(f"Name: {student[1]}, Department: {student[2]}, Section: {student[3]},
Marks: {student[4]}")
        else:
            st.warning("Student not found with the given ID.")

if __name__ == "__main__":
    main()
```