# 📘 Project Documentation: Student Management Application in Java with MySQL

## 1. \*\*Project Overview\*\*

The \*\*Student Management Application\*\* is a GUI-based desktop system built with \*\*Java Swing\*\* and integrated with \*\*MySQL\*\* using \*\*JDBC\*\*. It facilitates CRUD operations on student records in a secure, modular, and scalable way—ideal for academic institutions or training environments.

---

## 2. \*\*Key Objectives\*\*

- Allow administrators to manage student data via a clean user interface

- Ensure persistent storage through a relational database

- Modularize logic using DAO (Data Access Object) design pattern

---

## 3. \*\*Features\*\*

- Add new student details (Name, Course, Fees, etc.)

- Update or delete existing student records

- View all student entries in a table

- Persistent data handling via MySQL

- GUI-based interactions using Java Swing

---

## 4. \*\*Tech Stack\*\*

| Layer | Technology |

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

| Frontend (UI) | Java Swing |

| Backend Logic | Java (OOP Concepts) |

| Database | MySQL |

| Connectivity | JDBC (Java Database Connectivity) |

| Build Tool | Manual (javac/java) or IDE-based |

---

## 5. \*\*Database Setup\*\*

Before running the application, create the database and table using the provided SQL script:

### ✅ MySQL Configuration

```sql

CREATE DATABASE studentdb;

USE studentdb;

CREATE TABLE student (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

course VARCHAR(100) NOT NULL,

fee DECIMAL(10,2)

);

```

---

## 6. \*\*Project Structure\*\*

```

├── src/

│ ├── Main.java // Launches the GUI

│ ├── Student.java // POJO class for student entity

│ ├── DBConnection.java // Establishes MySQL connection

│ ├── StudentDAO.java // Handles DB operations (CRUD)

│ └── StudentForm.java // GUI form with event listeners

├── lib/

│ └── mysql-connector-java.jar // MySQL JDBC Driver

├── studentdb.sql // SQL schema file

└── README.md

```

---

## 7. \*\*Usage Guide\*\*

### 📦 Prerequisites

- JDK 17 or above

- MySQL Server

- MySQL JDBC Driver added to classpath

- IDE (e.g., IntelliJ IDEA, Eclipse) or command line

### ▶️ Run the Application

1. Clone the repo

2. Set up the database (`studentdb`)

3. Add the MySQL JDBC driver to your project

4. Compile and run `Main.java`

---

## 8. \*\*Code Flow Summary\*\*

- `Main.java` triggers the UI via `StudentForm`

- `StudentForm.java` provides buttons & form fields

- Event listeners call `StudentDAO` methods

- `DBConnection.java` supplies the live database connection

---

## 9. \*\*Possible Enhancements\*\*

- Input validation and error handling

- Admin login system

- Pagination and search in JTable

- Export to Excel or PDF

- REST API layer for integration with web frontend