**Project Synopsis**

**Title:**

**Console-Based C++ MySQL Application for Student Record Management**

**Introduction**

Database management systems play a vital role in handling and organizing data efficiently. This project demonstrates the integration of **C++ programming** with **MySQL database** to create a menu-driven console application. The system provides a simple yet effective way to perform basic operations such as inserting, retrieving, updating, and deleting student records. The application serves as a learning project for understanding database connectivity, query execution, and result handling in C++.

**Objectives**

* To establish a connection between a C++ application and a MySQL database.
* To perform **CRUD operations** (Create, Read, Update, Delete) on student records through a user-friendly menu-driven interface.
* To demonstrate the practical implementation of database operations using SQL queries within C++.
* To provide a foundation for developing more advanced database applications in the future.

**Scope of the Project**

The project is limited to managing a **student database table** with three fields: ID, Name, and Grade. It is designed to run on the Windows platform and is intended primarily for academic and demonstration purposes. Although basic, the system can be extended into larger applications such as school management systems, employee record systems, or any application requiring database-driven storage.

**System Requirements**

***Hardware Requirements***

* Processor: Intel i3 or above
* RAM: Minimum 4 GB
* Storage: 200 MB free space

***Software Requirements***

* Operating System: Windows 7 or above
* Compiler: g++ / MinGW / Visual Studio C++
* Database: MySQL Server
* MySQL C API / Connector installed

**Methodology**

1. **Database Setup**
   * Create a database my\_db in MySQL.
   * Create a table student(ID INT, Name VARCHAR(100), Grade FLOAT).
2. **Application Development**
   * Establish a connection to MySQL using the MySQL C API.
   * Implement a menu-driven program with the following options:
     + **Insert Data** – Add new student records.
     + **Read Data** – View all stored student records.
     + **Update Data** – Modify grades of students by their ID.
     + **Delete Data** – Remove student records using ID.
     + **Exit** – Terminate the program gracefully.
3. **Execution**
   * The program runs on the console and allows the user to perform operations interactively.
   * Error handling is incorporated to display database-related issues.

**Expected Outcomes**

* A functional C++ application capable of performing CRUD operations on a MySQL database.
* Hands-on understanding of database connectivity and SQL integration in C++.
* A working demonstration of how database-driven applications can be built from scratch.

**Limitations**

* The application is platform-dependent (Windows only).
* Passwords are stored in plain text, which is insecure.
* Queries are constructed through string concatenation, which makes the system vulnerable to SQL injection.
* No graphical user interface (GUI) is provided; interaction is text-based only.

**Future Enhancements**

* Replace direct queries with **prepared statements** for enhanced security.
* Move database credentials to a **configuration file or environment variables**.
* Implement a **cross-platform version** by replacing Windows-specific functions.
* Add a **graphical user interface** for better usability.
* Extend the project to manage larger datasets with advanced functionalities such as search, filters, and reporting.

**Conclusion**

This project successfully demonstrates how **C++ applications can interact with MySQL databases** to manage records effectively. It provides a practical understanding of database operations and serves as a strong foundation for future development of larger-scale database-driven systems.