Tab 1

Project Title: Stock Transaction Recording System.

Subject: Open Source Database (MySQL)

Course / Semester: SY BSc IT / Semester IV

Student Name: IRFAN KAYUM MANIYAR

Roll Number: 2542023

ERM / PRN Number:

College Name: R.K. Talreja College

University: University of Mumbai

Academic Year: 2025 – 2026

Submitted To: Asst.Prof. Sahil Shukla

---

1. Introduction

The Stock Transaction Recording System is a database-driven project designed to record, store, and manage stock trading activities such as buying and selling shares. Managing stock transactions manually is time-consuming and error-prone. A database is required to store large volumes of transaction data in an organized manner. MySQL is used as the backend database to ensure data accuracy, fast access, and secure storage of stock-related information.

—

2. Problem Statement

Without a proper database system, stock transaction records face several issues:

Manual record keeping leads to errors and inconsistency.

Data duplication occurs frequently.

Searching and updating records is slow.

No proper security and access control.

---

3. Objectives of the Project

- To design a structured database using MySQL for stock transactions.

- To store buy and sell transaction data efficiently.
- To retrieve stock and transaction details using SQL queries
- To maintain data integrity using database constraints

---

4. Scope of the Project

This system covers recording of stock details, user (investor) details, and transaction history. It is intended for students and small-scale academic use to understand database concepts. The project does not include real-time stock market integration and is limited to offline data handling for learning purposes.

---

5. Tools & Technologies Used

Component	**Description**
Database	**MySQL**

Interface	**MySQL Workbench**
Language	**SQL**
Platform	**Windows / Linux**
Type Open	**Source DBMS**

---

6. Database Design

6.1 Tables (Entities)

- **User**
- **Stock**
- **Transaction**
- **Portfolio**

6.2 Attributes

- **User** (user_id, name, email, phone)
- **Stock** (stock_id, stock_name, symbol, price)
- **Transaction** (transaction_id, user_id, stock_id, transaction_type, quantity, transaction_date)

- **Portfolio** (portfolio_id, user_id, stock_id, total_quantity)

6.3 Constraints Used

PRIMARY KEY

FOREIGN KEY

NOT NULL

UNIQUE

---
7. SQL Concepts Used

- CREATE, INSERT, UPDATE, DELETE statements
- SELECT queries with WHERE clause
- JOIN operations
- GROUP BY and HAVING
- Subqueries
- Views
- User privileges
- Transactions (COMMIT, ROLLBACK)

- Transaction for Buy/Sell Operation

In this system, every buy or sell action is treated as a database transaction. When a user buys a stock, a record is inserted into the Transaction table and the Portfolio table is updated by increasing the stock quantity. When a user sells a stock, the Portfolio table is updated by decreasing the stock quantity and a sell record is stored in the Transaction table. Both actions are executed together as a single transaction to ensure data accuracy.

- Balance / Portfolio Consistency

Portfolio consistency means that the stock quantity stored in the Portfolio table must always match the total quantity calculated from buy and sell transactions. This is maintained using constraints and transaction control. If a user tries to sell more stocks than available in the portfolio, the system does not allow the operation. This ensures that incorrect or inconsistent data is not stored in the database.

- Rollback Example

If a user performs a buy operation and an error occurs while updating the portfolio (for example, invalid stock ID or system failure), the transaction is rolled back. This cancels all changes made during that transaction. As a result, neither the transaction record nor the portfolio update is saved. Rollback ensures that either all operations are completed successfully or none are applied, thus maintaining database integrity.

8. Security & Backup

User authentication and role-based access control

Password-protected database users

Regular database backup using mysqldump

---

9. Expected Outcome

Faster access to stock transaction records

Reduced data redundancy

Accurate and secure data storage

Easy generation of transaction reports

---

10. Conclusion

This project provides practical exposure to MySQL and database design concepts. It helps in understanding how stock transaction data can be efficiently stored and managed using an open-source database system. The project strengthens SQL knowledge and real-world database application skills.

---

11. Future Enhancements

Integration with a web-based user interface

Real-time stock price updates and advanced reporting

---

12. References

MySQL Official Documentation

Database Management System Textbooks

Online SQL learning resources