

Established – 1961

Subject:
osdbms

**SEVA SADAN'S
R. K. TALREJA COLLEGE
OF
ARTS, SCIENCE & COMMERCE
ULHASNAGAR – 421 003**



CERTIFICATE

This is to certify that Mr./Ms. Ayur Chendwankar of S.Y. Information Technology (SYIT) Roll No. 2542005 has satisfactorily completed the Open Source DataBase Management System Mini Project entitled

during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for submission.

PROF. INCHARGE

HEAD OF DEPT

INDEX

Sr. No.	Chapter Title	PAGE NO
1	Introduction	
2	Problem Definition	
3	Objectives of the Project	
4	Scope of the Project	
5	Requirement Specification	
6	System Design	
7	Database Design	
8	UML Diagrams	
9	SQL Implementation	
10	System Testing and Result	
11	Security, Backup and Recovery	
12	Future Scope and Conclusion	
13	References	
14	Glossary	

Acknowledgement

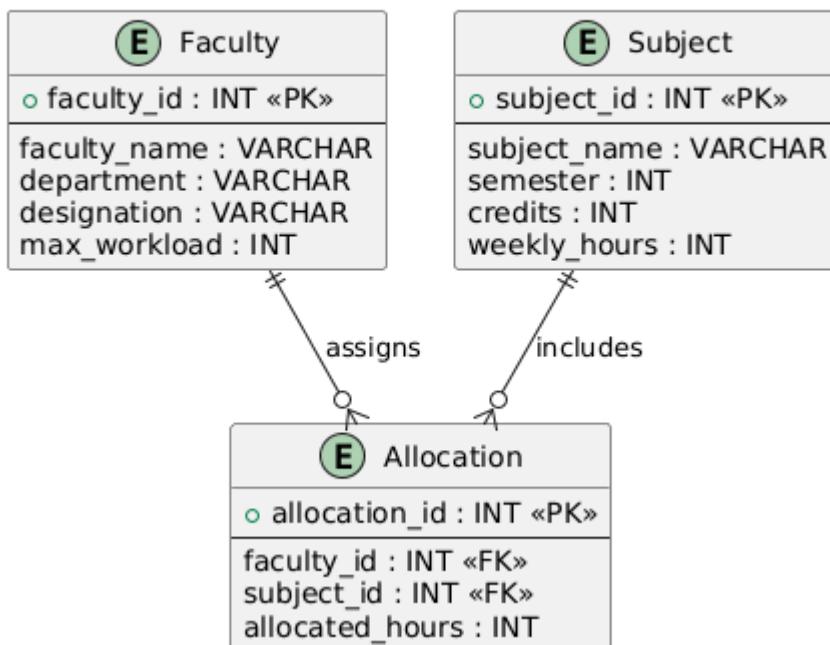
I would like to express my sincere gratitude to my project guide for continuous support and guidance throughout this project. I also thank my faculty members and friends who helped me understand database concepts clearly. This project helped me gain practical knowledge of MySQL, relational database design, and SQL queries.

Chapter 2: Literature Review

Traditional institutions use manual registers or spreadsheets to track workload. These methods lack automation and relational integrity. Modern institutions use database systems to manage academic records efficiently.

Relational Database Management Systems (RDBMS) such as MySQL provide structured storage, integrity constraints, and powerful query mechanisms. Using SQL, institutions can easily generate reports and maintain workload balance.

ER Diagram - Faculty Workload & Subject Allocation Database



Chapter 3: System Analysis

3.1 Existing System

- Manual registers
- Excel sheets
- No automatic overload detection
- No relational integrity

3.2 Proposed System

- MySQL database-based solution
- Automatic workload calculation
- Foreign key constraints
- Overload detection using HAVING clause

3.3 Feasibility Study

Technical Feasibility

MySQL is easy to install and widely supported.

Economic Feasibility

Open-source software, no additional cost.

Chapter 4: System Design

4.1 Database Design

The system contains three tables:

Faculty Table

Stores faculty details including maximum workload.

Subject Table

Stores subject details including credits and weekly hours.

Allocation Table

Connects faculty and subjects and records allocated hours.

4.2 ER Diagram (Text Representation)

Faculty (1) —— (M) Allocation (M) —— (1) Subject

One faculty can teach many subjects.

One subject can be allocated to many faculty (if needed).

4.3 Data D

ictionary

Faculty Table

Field Name	Data Type	Description
faculty_id	INT	Primary Key
faculty_name	VARCHAR	Name of faculty
department	VARCHAR	Department name
designation	VARCHAR	Designation
max_workload	INT	Maximum teaching hours

Subject Table

Field	Type	Description
subject_id	INT	Primary Key
subject_name	VARCHAR	Subject name
Semester	INT	Semester number
Credits	INT	Credit value
weekly_hours	INT	Weekly teaching hours

Allocation Table

Field	Type Description
allocation_id	INT Primary Key
faculty_id	INT Foreign Key
subject_id	INT Foreign Key
allocated_hours	INT Assigned hours

Chapter 5: Implementation

5.1 Database Creation

Database: FacultyWorkloadDB

Tables:

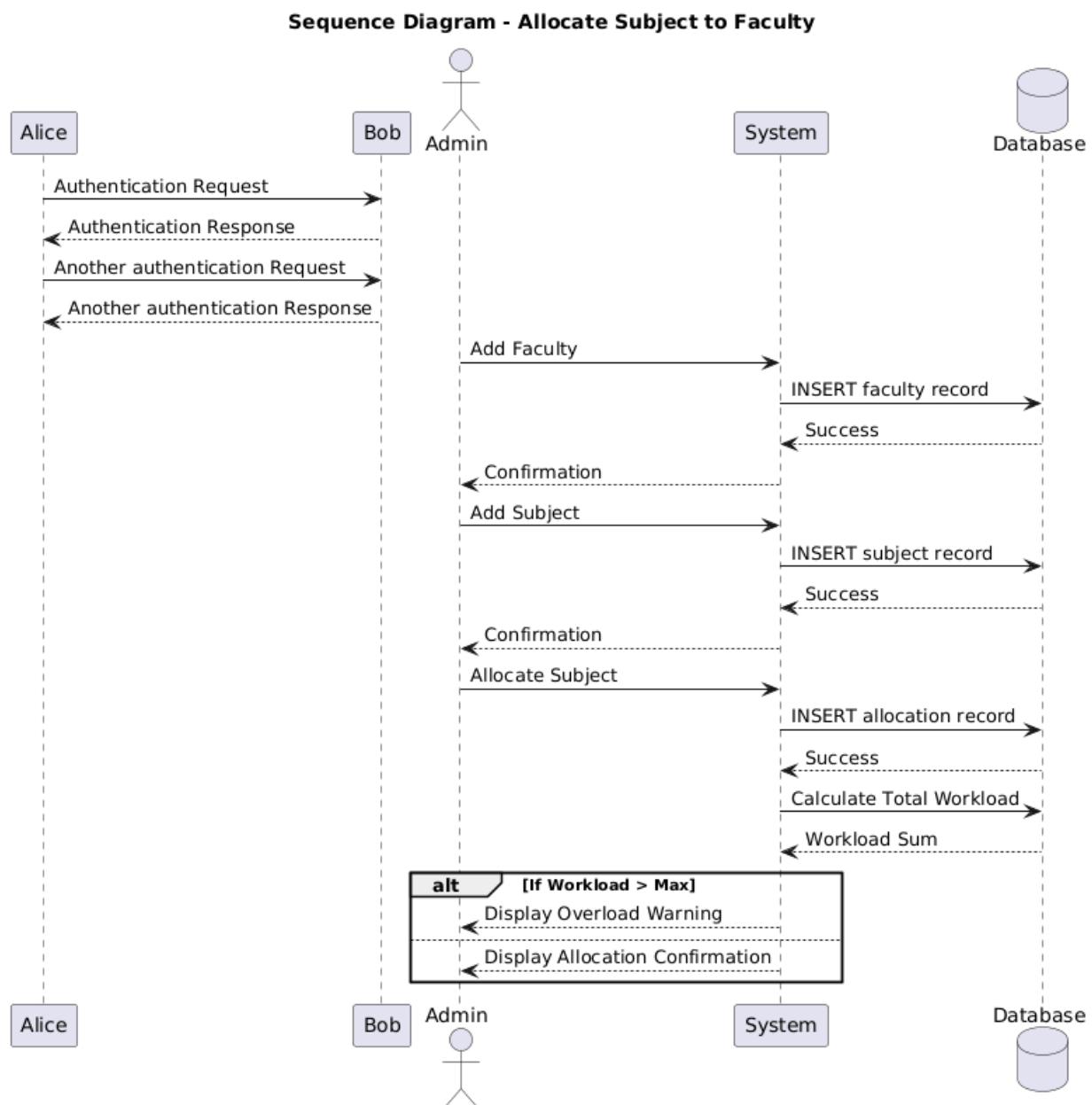
- Faculty
- Subject
- Allocation

5.2 SQL Operations Used

- CREATE DATABASE
- CREATE TABLE
- PRIMARY KEY
- FOREIGN KEY
- AUTO_INCREMENT
- INSERT
- SELECT
- JOIN
- GROUP BY
- HAVING
- SUM()

Chapter 6: Output and Results

- 1. Faculty details are stored in Faculty table.**
- 2. Subject details are stored in Subject table.**
- 3. Subjects are allocated using Allocation table.**
- 4. Total workload is calculated using SUM().**
- 5. HAVING clause checks overload condition.**



6.1 Total Workload Calculation

Displays total workload of each faculty using:

```
SELECT f.faculty_name,  
       SUM(a.allocated_hours) AS total_workload  
  FROM Faculty f  
 JOIN Allocation a ON f.faculty_id = a.faculty_id  
 GROUP BY f.faculty_name;
```

6.2 Faculty Subject Allocation

Displays faculty with subjects assigned.

6.3 Overload Detection

Uses HAVING clause:

```
HAVING SUM(a.allocated_hours) > f.max_workload;
```

Chapter 7: Advantages (Brief)

- 1. Prevents faculty overload** – The system calculates total workload to ensure no faculty member exceeds their maximum teaching hours.
- 2. Maintains proper workload records** – All allocations are stored in the database, making it easy to track assignments.
- 3. Ensures data integrity** – Using primary and foreign keys, the system avoids inconsistent or invalid data.
- 4. Easy to generate reports** – SQL queries allow quick generation of workload summaries and allocation details.
- 5. Efficient workload distribution** – Subjects are assigned in a structured way, balancing workload across faculty.

Chapter 8: Limitations (Brief)

- 1. No graphical user interface** – Users interact directly with SQL commands, making it less user-friendly.
- 2. No login authentication** – There's no system to restrict access; anyone with database access can modify data.
- 3. Limited to basic allocation tracking** – The system only handles faculty, subjects, and workload; it doesn't manage advanced academic processes.

Chapter 9: Future Enhancements (Brief)

- 1. Add admin login system** – Secure access with authentication to prevent unauthorized changes.
- 2. Web-based interface** – Create a user-friendly interface for easy management of faculty and subjects.
- 3. Automatic timetable generation** – The system could generate weekly teaching schedules automatically.
- 4. Report generation in PDF format** – Export workload reports for printing or record-keeping.
- 5. Integration with student database** – Connect with student data for a complete academic management system.

11. Conclusion

The Faculty Workload & Subject Allocation Database successfully demonstrates relational database design using MySQL. The system efficiently manages faculty teaching load and prevents overload using aggregate functions and SQL constraints. This project improves understanding of joins, foreign keys, grouping, and data integrity in DBMS.