

## ASSIGNMENT SUBMISSION SCHEDULING SYSTEM



# REPORT ON Assignment Submission Scheduling System

**Submitted by: MANUTOSH BENIWAL**

**Date: 18/08/2025**

# Table of Contents

1. Introduction
  2. Overview of Assignment Submission Scheduling System
  3. Literature Review
  4. Creation of Dataset for Assignment Submission Scheduling System
  5. Creation of Database for the Assignment Submission Scheduling System
  6. Data Analysis
    - 6.1 Description of Machine Learning Technique for Detecting Conflicts
    - 6.2 Analysis of the Dataset
  7. Data Visualization of the Result
  8. Conclusion
- References
- Appendix

## 1. Introduction

In academic institutions, managing assignment submissions effectively is a recurring challenge. Overlapping deadlines are a common problem for students, which can result in missed assignments, elevated stress levels, and worse academic performance. The goal of this project is to create a scheduling system that uses data analysis and machine learning to anticipate and address assignment deadline conflicts.

In order to improve time management for both students and professors, a scalable, data-driven system that enables assignment tracking, conflict detection, and submission timeline visualization is being developed.

Digital systems that offer flexibility, tracking, and integration with learning management systems have replaced paper submission for assignment management in today's digital learning environment. This has been accelerated by remote and hybrid learning models. In order to preserve academic integrity, provide prompt feedback, and lessen administrative burdens, schools and universities are consequently depending more and more on digital submission tools for assignments.

However, issues like inflexible workflow procedures, a lack of customization, and a limited capacity for conflict detection are often made worse by existing systems. Usually, missed deadlines, duplicate submissions, or ineffective feedback loops would negatively impact both students and teachers. A more advanced, data-driven assignment submission system that not only keeps a close eye on activities but also gives information on submission frequency and deadline conflicts is required due to the limitations.

## **2. Project Overview**

Planning and simulating a more complex assignment submission system with synthetic data, a contemporary database design, and machine learning-based conflict detection are the goals of this project. Among the goals are evaluation of the systems for submitting assignments and their shortcomings, using artificial academic data to create a relational data model, and putting in place an SQL database for efficient major operations.

Developing a machine learning algorithm to predict submission conflicts, Submission trends and deadline overlaps are visualized for insights.

A thorough report complete with database files, code, visualizations, and accompanying documentation will be the end result.

### **Overview of the System**

The suggested system for submitting assignments replicates a normal academic setting with lots of users, courses, assignments, and deadlines. The following elements make up the system:

#### **Functionalities & Features:**

- Monitoring submissions with timestamp logging
- Enforcing deadlines and establishing grace periods
- Integration of early-stage machine learning conflict prediction
- Features for visualizing high-risk due dates and submission obstacles

#### **Overview of Workflow:**

- Professors assign assignments with due dates.
- Through the system, students submit them.
- The program detects late submissions and timestamps them.
- Conflict detection recognizes when deadlines overlap.
- Administrators resolve conflicts and view visual reports.

### **3. Review of Literature**

#### **Examining Existing Systems**

##### **1. Moodle**

- Advantages include its open-source nature, modular design, and high degree of customization.
- Negative aspects include a very high learning curve, difficulty scaling to large institutions, and an extremely complex installation.

##### **2. The Blackboard**

- Advantages: Integrated grading and assessment tools in an enterprise learning management system
- Cons: Very costly, rigid, and generally regarded as outdated in UI design

##### **3. Google Classroom**

- Advantages include being lightweight, user-friendly, and integrating with Google Drive and Docs.
- Weaknesses: Does not support sophisticated features like bulk upload, analytics, and customizable deadlines.

##### **4. Turnitin**

- Advantages include robust plagiarism detection and LMS integration.
- Weaknesses: Emphasizing originality checking over full-cycle submission, not an all-purpose manager of assignments

#### **Important Difficulties Found**

- Overlaps in Deadlines Students who have submitted two or deadlines within constrained time frames
- Limited Notification Systems that don't provide feedback or deadline notifications in real time
- Low adaptability, tight submission guidelines and no late policy configuration

## 4. Dataset Description

For this project, a synthetic dataset was created to simulate a university assignment submission system. The dataset models interactions between students, professors, courses, assignments, and submission records. The structure aligns with a realistic academic environment, ensuring logical relationships and normalized data.

### 4.1 Tables Overview

The following key tables were created:

Table Name	Description
users	Stores student and professor information (name, email, role).
courses	Contains course details and links to the professor.
assignments	Holds assignments for each course.
deadlines	Includes due dates and reminder dates for each assignment.
submissions	Records assignment submissions by students.
enrollment	Maps students to the courses they are enrolled in.
combined_data	A denormalized table used for analysis and report generation.

### 4.3 Synthetic Data Generation

- 120 students were generated with unique names and email addresses.
- 15 courses were created and assigned to 12 professors.
- 30 assignments were distributed across the 15 courses.
- Each student was enrolled in 1–3 courses.
- Each assignment had a hand-in date, due date, reminder date, and submission date.
- Submission dates and grades were randomly generated within valid ranges.

CSV files were generated for each table and then imported into the database.

## 5. Database Setup and Testing

### 5.1 Database Creation

A MySQL database named `assignment_system` was created. SQL scripts were written to define all table structures with appropriate foreign keys and constraints based on the ER diagram.

#### Tools Used:

- MySQL Workbench for schema design and SQL execution
- Python (faker) and Kaggle for synthetic data creation
- Command line and Workbench for import and testing

```
mysql> create database assignment_system;  
Query OK, 1 row affected (0.01 sec)  
  
mysql> use assignment_system;  
Database changed
```

## 5.2 SQL Table Creation

All tables were created using **CREATE TABLE** statements with proper data types, primary keys, and foreign key constraints to maintain referential integrity.

```
mysql> CREATE TABLE users (  
->     user_id INT PRIMARY KEY,  
->     name VARCHAR(100),  
->     email VARCHAR(100),  
->     role ENUM('student', 'instructor') NOT NULL  
-> );  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> show tables;
```

Tables_in_assignment_system
assignments
courses
deadlines
submissions
users

5 rows in set (0.00 sec)

```
mysql> desc courses;
```

Field	Type	Null	Key	Default	Extra
course_id	int	NO	PRI	NULL	
course_name	varchar(100)	YES		NULL	
professor_id	int	YES	MUL	NULL	
start_date	date	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> desc users;
```

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
role	enum('student', 'instructor')	NO		NULL	

4 rows in set (0.00 sec)

```
mysql> desc assignments;
```

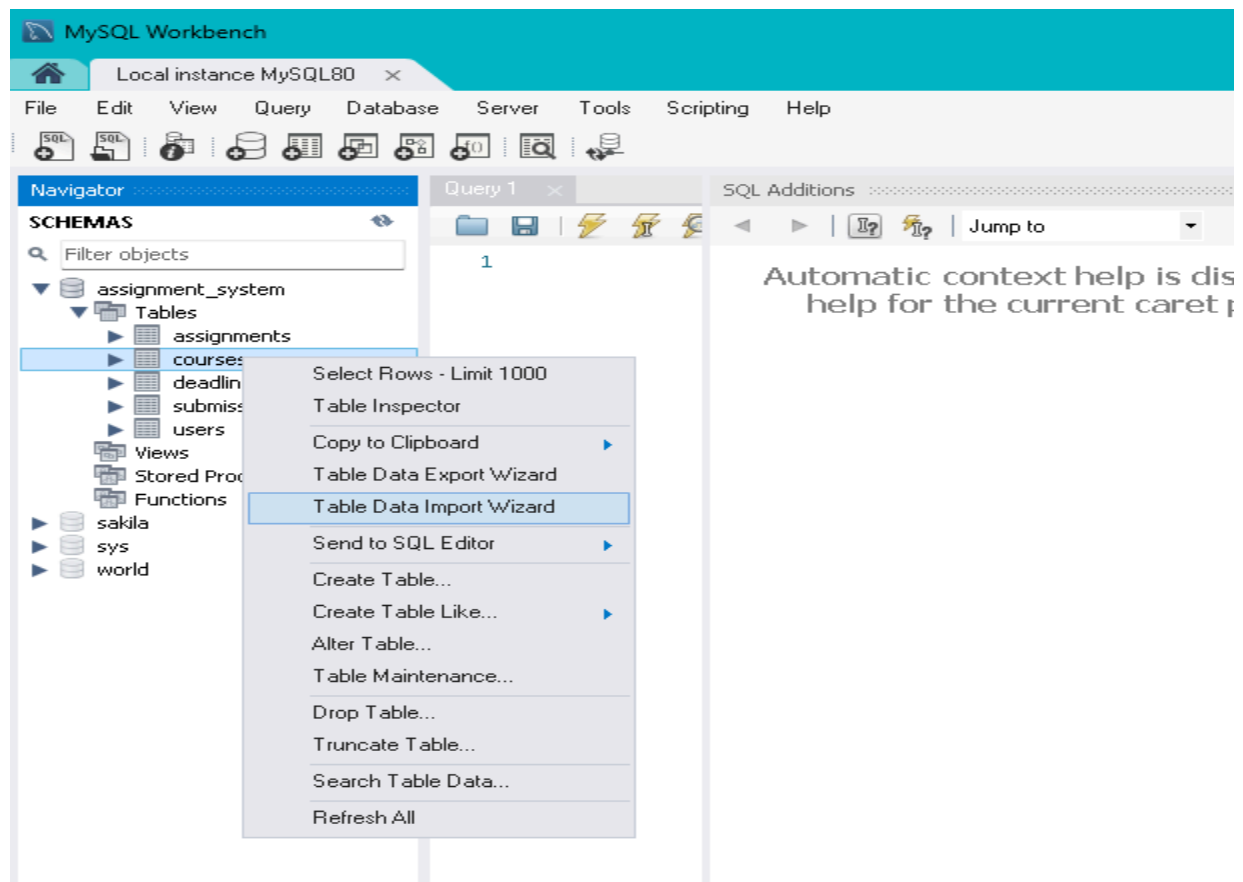
Field	Type	Null	Key	Default	Extra
assignment_id	int	NO	PRI	NULL	
course_id	int	YES	MUL	NULL	
title	varchar(255)	YES		NULL	
due_date	date	YES		NULL	

4 rows in set (0.00 sec)



### 5.3 Data Import

CSV files were imported using the **LOAD DATA INFILE** command in MySQL or the "Import" feature in Workbench. Data was validated to ensure all foreign key relationships were consistent.



### 5.4 CRUD Operation Testing

Basic functionality was tested to ensure the system can handle:

- **Insert:** Adding new users, submissions, and assignments
- **Read:** Querying data using **SELECT** to verify relationships
- **Update:** Modifying grades and deadlines
- **Delete:** Removing submissions and verifying cascade behavior

Sample queries and their results were captured as screenshots (see attached figures).

```
mysql> desc submissions;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| submission_id | int | NO | PRI | NULL | |
| user_id | int | YES | MUL | NULL | |
| assignment_id | int | YES | MUL | NULL | |
| submission_date | date | YES | | NULL | |
| grade | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into submissions values(801,102,3,'2025-08-23',0);
Query OK, 1 row affected (0.01 sec)

mysql> |
```

```
mysql> select submission_id,user_id,submission_date,grade,name,due_date_x,due_date_y from combined_data where submission_date>due_date_x
+-----+-----+-----+-----+-----+-----+-----+
| submission_id | user_id | submission_date | grade | name | due_date_x | due_date_y |
+-----+-----+-----+-----+-----+-----+-----+
| 3 | 25 | 2025-07-29 00:00:00 | 83 | Dustin Lopez | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 8 | 92 | 2025-08-01 00:00:00 | 0 | Jake Fisher | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 10 | 20 | 2025-08-01 00:00:00 | 0 | Melanie Romero | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 14 | 40 | 2025-08-05 00:00:00 | 51 | Paula McClain | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 16 | 6 | 2025-08-05 00:00:00 | 61 | Linda Morris | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 19 | 75 | 2025-08-06 00:00:00 | 51 | Deborah Vasquez | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 20 | 82 | 2025-08-06 00:00:00 | 77 | Natalie Henderson | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 21 | 93 | 2025-08-04 00:00:00 | 53 | John Suarez | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 23 | 103 | 2025-08-04 00:00:00 | 30 | Maria Mejia | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 25 | 99 | 2025-08-03 00:00:00 | 93 | Tracy Kramer | 2025-08-02 00:00:00 | 2025-08-02 00:00:00 |
| 26 | 7 | 2025-08-08 00:00:00 | 37 | Scott Warren | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 28 | 50 | 2025-08-06 00:00:00 | 68 | Gary Bruce | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 30 | 56 | 2025-08-06 00:00:00 | 59 | Karen Daniels | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 33 | 55 | 2025-08-06 00:00:00 | 83 | Geoffrey Rogers | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 34 | 105 | 2025-08-05 00:00:00 | 56 | Amanda Castillo | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 35 | 36 | 2025-08-07 00:00:00 | 33 | Brett Nixon | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 36 | 66 | 2025-08-08 00:00:00 | 20 | Jennifer Valdez | 2025-08-03 00:00:00 | 2025-08-03 00:00:00 |
| 39 | 120 | 2025-08-03 00:00:00 | 24 | Laura Lee | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 43 | 80 | 2025-08-03 00:00:00 | 43 | Michael Hays | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 44 | 68 | 2025-07-31 00:00:00 | 67 | Dana Ewing | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 46 | 17 | 2025-08-03 00:00:00 | 61 | Thomas Ramirez | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 47 | 65 | 2025-08-01 00:00:00 | 43 | Ricky Smith | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 49 | 21 | 2025-08-02 00:00:00 | 60 | Joshua Smith | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 51 | 36 | 2025-08-04 00:00:00 | 57 | Brett Nixon | 2025-07-30 00:00:00 | 2025-07-30 00:00:00 |
| 61 | 6 | 2025-07-29 00:00:00 | 79 | Linda Morris | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 63 | 109 | 2025-07-28 00:00:00 | 73 | Megan Berry | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 65 | 66 | 2025-08-01 00:00:00 | 0 | Jennifer Valdez | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 67 | 89 | 2025-07-27 00:00:00 | 92 | Kelly Le | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 68 | 31 | 2025-07-31 00:00:00 | 30 | Kimberly Singleton | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 69 | 13 | 2025-07-31 00:00:00 | 43 | James Porter | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 75 | 99 | 2025-07-27 00:00:00 | 74 | Tracy Kramer | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 77 | 59 | 2025-07-30 00:00:00 | 71 | Amy Ortiz | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 78 | 24 | 2025-08-01 00:00:00 | 0 | Gloria Ryan | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 80 | 37 | 2025-07-27 00:00:00 | 81 | Paul Jones | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 81 | 30 | 2025-08-01 00:00:00 | 0 | Parker Wiley | 2025-07-26 00:00:00 | 2025-07-26 00:00:00 |
| 84 | 30 | 2025-07-26 00:00:00 | 62 | Parker Wiley | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 86 | 107 | 2025-07-25 00:00:00 | 79 | Mia Boyd | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 87 | 89 | 2025-07-28 00:00:00 | 47 | Kelly Le | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 89 | 96 | 2025-07-27 00:00:00 | 58 | Dr. Shawn Benson | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 91 | 98 | 2025-07-26 00:00:00 | 58 | Joshua Nguyen | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 94 | 41 | 2025-07-26 00:00:00 | 79 | Tina Miller | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 95 | 31 | 2025-07-25 00:00:00 | 95 | Kimberly Singleton | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 96 | 15 | 2025-07-26 00:00:00 | 60 | Jasmine Campbell | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
| 97 | 67 | 2025-07-30 00:00:00 | 0 | Donna Pittman | 2025-07-24 00:00:00 | 2025-07-24 00:00:00 |
```

```
mysql> select * from submissions where submission_id=800;
+-----+-----+-----+-----+-----+
| submission_id | user_id | assignment_id | submission_date | grade |
+-----+-----+-----+-----+-----+
|          800 |     102 |           3 | 2025-08-23      |     0 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> update submissions set grade=40 where submission_id=800;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from submissions where submission_id=800;
+-----+-----+-----+-----+-----+
| submission_id | user_id | assignment_id | submission_date | grade |
+-----+-----+-----+-----+-----+
|          800 |     102 |           3 | 2025-08-23      |    40 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from submissions where submission_id=800;
+-----+-----+-----+-----+-----+
| submission_id | user_id | assignment_id | submission_date | grade |
+-----+-----+-----+-----+-----+
|          800 |     102 |           3 | 2025-08-23      |    40 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> delete from submissions where submission_id>=800;
Query OK, 2 rows affected (0.01 sec)
```

```
mysql> select * from submissions where submission_id>400;
Empty set (0.00 sec)
```

```
mysql> select * from submissions where submission_id>=400;
+-----+-----+-----+-----+-----+
| submission_id | user_id | assignment_id | submission_date | grade |
+-----+-----+-----+-----+-----+
|          400 |     102 |           3 | 2025-08-23      |     0 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## 5.5 Deadline Enforcement Trigger

A trigger named `check_deadline` was implemented to enforce submission constraints. The trigger prevents inserting a submission **after the assignment's due date**. If the submission date exceeds the deadline, an error is raised, preserving data validity.

```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER check_deadline
-> BEFORE INSERT ON submissions
-> FOR EACH ROW
-> BEGIN
->     DECLARE due DATETIME;
->     SELECT due_date INTO due FROM deadlines WHERE assignment_id = NEW.assignment_id;
->
->     IF NEW.submission_date > due THEN
->         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Submission past due date!';
->     END IF;
-> END $$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```