Project Report: Campus Connect ERP System

Submitted By:
Akshit K. Bansal: 2024058
Ashutosh Prasad: 2024136

1. Project Overview & Architecture

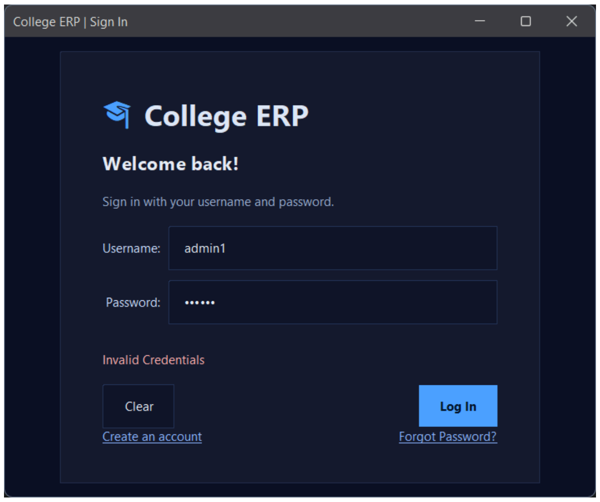
Campus Connect is a modern, desktop-based Enterprise Resource Planning (ERP) system developed for academic institutions. It serves as a centralized platform for managing the entire academic lifecycle—from course creation and registration to grading and system administration.

2. Role-Based Access Control

The system separates all users into three distinct roles in order to separate them according to their access priveledges

A. Authentication & Session Management

- **Security:** Passwords are verified using BCrypt (via PasswordUtil.CheckPassword), ensuring that raw passwords are never stored or compared directly.
- **Brute Force Protection:** If a Student account fails 5 consecutive login attempts, the LoginWindow enforces a local 30-second lockout using a HashMap to track failed attempts by username.


College ERP | Sign In

College ERP

Welcome back!

Sign in with your username and password.

Username: admin1

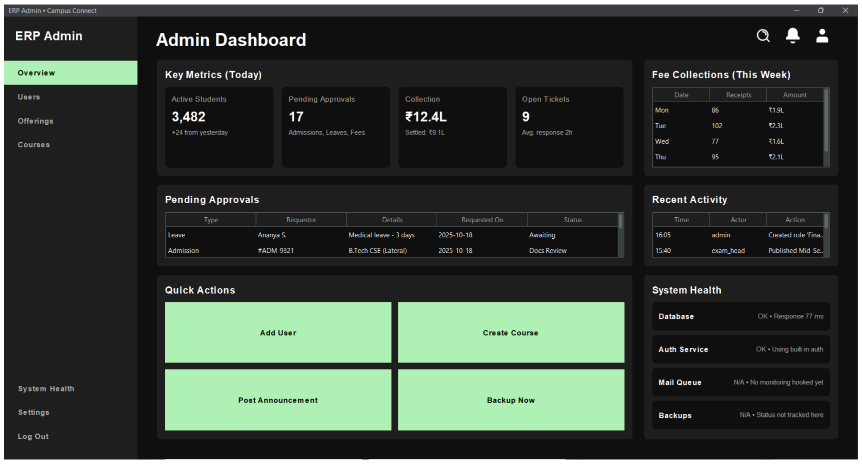
Password: ••••••

Invalid Credentials

B. Routing Strategy

Upon successful login, LoginWindow.java routes the user to a role-specific Dashboard class. This is a critical security feature: accessing another role's dashboard is impossible because the code for it is never instantiated.

- **Admin:** DashboardUIAdmin(int AdminId) - Full system control.
- **Instructor:** DashboardUIInstructor(int InstructorId) - Section management.
- **Student:** DashboardUIStudent(int RollNumber) - Personal academic view.



Admin Dashboard

Key Metrics (Today)

Active Students 3,482 +24 from yesterday

Pending Approvals 17 Admissions, Leaves, Fees

Collection ₹12.4L Settled: ₹9.1L

Open Tickets 9 Avg. response 2h

Fee Collections (This Week)

Date	Receipts	Amount
Mon	86	₹1.9L
Tue	102	₹2.3L
Wed	77	₹1.6L
Thu	95	₹2.1L

Pending Approvals

Type	Requestor	Details	Requested On	Status
Leave	Ananya S.	Medical leave - 3 days	2025-10-18	Awaiting
Admission	#ADM-9321	B.Tech CSE (Lateral)	2025-10-18	Docs Review

Recent Activity

Time	Actor	Action
16:05	admin	Created role 'Fina..
15:40	exam_head	Published Mid-Se..

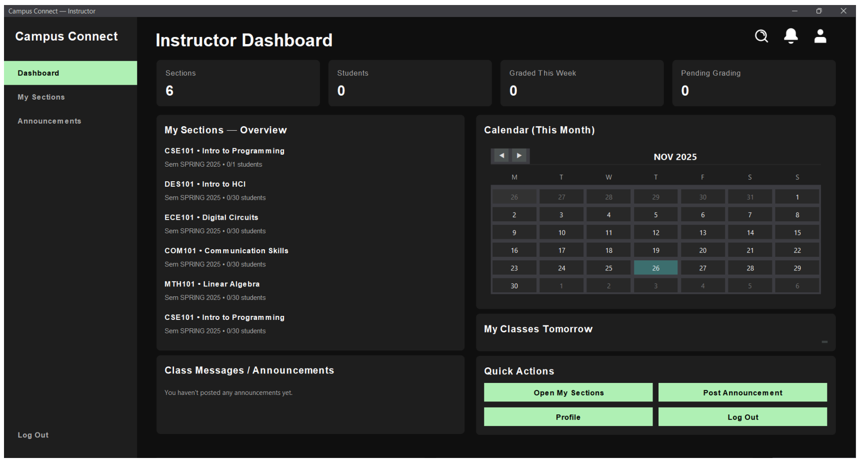
System Health

Database OK • Response 77 ms

Auth Service OK • Using built-in auth

Mail Queue N/A • No monitoring hooked yet

Backups N/A • Status not tracked here



Instructor Dashboard

Sections 6

Students 0

Graded This Week 0

Pending Grading 0

My Sections — Overview

CSE101 • Intro to Programming
Sem SPRING 2025 • 0/1 students

DES101 • Intro to HCI
Sem SPRING 2025 • 0/30 students

ECE101 • Digital Circuits
Sem SPRING 2025 • 0/30 students

COM101 • Communication Skills
Sem SPRING 2025 • 0/30 students

MTH101 • Linear Algebra
Sem SPRING 2025 • 0/30 students

CSE101 • Intro to Programming
Sem SPRING 2025 • 0/30 students

Calendar (This Month)

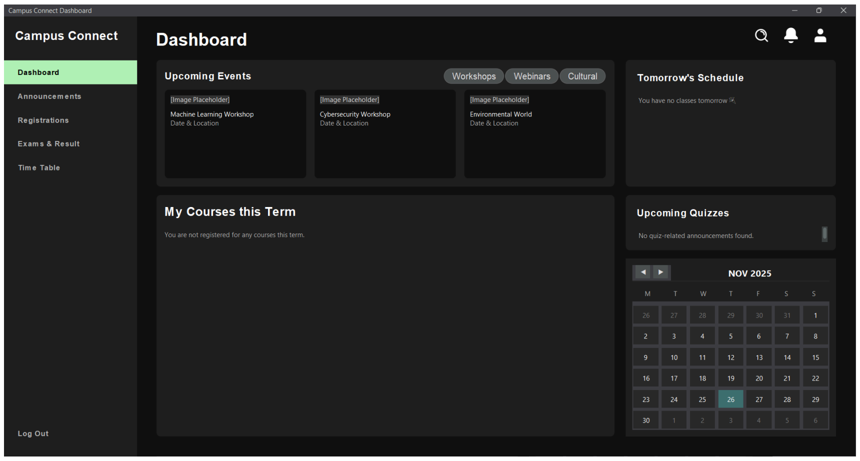
NOV 2025

M	T	W	T	F	S	S
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

My Classes Tomorrow

Class Messages / Announcements

You haven't posted any announcements yet.



Dashboard

Upcoming Events

[Image Placeholder]
Machine Learning Workshop
Date & Location

[Image Placeholder]
Cybersecurity Workshop
Date & Location

[Image Placeholder]
Environmental World
Date & Location

Tomorrow's Schedule

You have no classes tomorrow

My Courses this Term

You are not registered for any courses this term.

Upcoming Quizzes

No quiz-related announcements found.

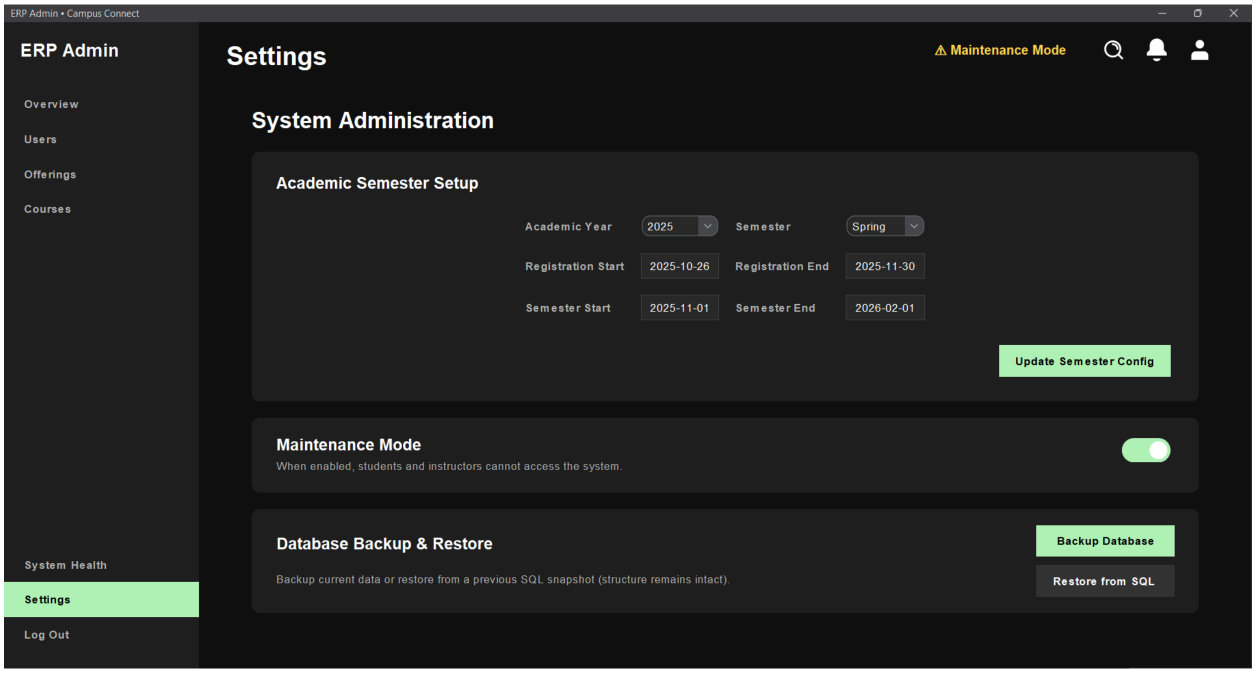
---

3. Maintenance Mode

The system features a global **Maintenance Mode** to prevent data inconsistency during critical administrative tasks (e.g., finalizing a semester).

Implementation Details

- **State Storage:** The boolean state is stored in a database table and retrieved via SemestersHandler.isMaintenanceMode().
- **Admin Control:** The AdminSettingsPanel.java contains a toggle switch to enable/disable this mode.
- **Enforcement:**
  - **Global Warning:** All Dashboards (DashboardUIStudent, DashboardUIInstructor, DashboardUIAdmin) display a yellow "⚠️ Maintenance Mode" label in the top bar when active.
  - **Functional Lockout:** Critical write operations are disabled at the UI level. For example, in AssignGradesDialog.java, the "Publish Grades" button is disabled, and editing tools are locked.


ERP Admin • Campus Connect

ERP Admin

Settings

⚠ Maintenance Mode

System Administration

Academic Semester Setup

Academic Year	2025	Semester	Spring
Registration Start	2025-10-26	Registration End	2025-11-30
Semester Start	2025-11-01	Semester End	2026-02-01

Update Semester Config

Maintenance Mode

When enabled, students and instructors cannot access the system.

Database Backup & Restore

Backup current data or restore from a previous SQL snapshot (structure remains intact).

Backup Database

Restore from SQL

---

3. Courses and Offering Management

The Campus Connect ERP distinguishes between **Courses** and **Offerings** to ensure efficient curriculum management.

1. Courses: Administrators maintain a master list of **Courses** that represents the university's curriculum independent of time. A Course record defines invariant attributes such as the Course Code (e.g., "CSE101"), Course Title, and Credit value. Creating a course in the system makes it available to be taught in future terms but does not automatically schedule it.

2. Course Offerings (Semester Instances) To schedule classes for a specific term, Administrators create **Offerings**. An Offering is an active instance of a Course linked to a specific Semester (e.g., "Spring") and Year.

- **Multiple Sections:** The system supports a one-to-many relationship, allowing a single Course to have multiple Offerings in the same semester. This feature is used to create different sections for the same subject (e.g., Section A and Section B).
- **Enrollment Tracking:** Each offering functions as a registration container. It tracks the **Current Enrollment** against a defined capacity to determine availability status (e.g., Open or Closed) during student registration.

4. Grading System: Flexible & Weighted

Each course can have its own grading system with custom components and grading slabs that can be edited by the course instructor.

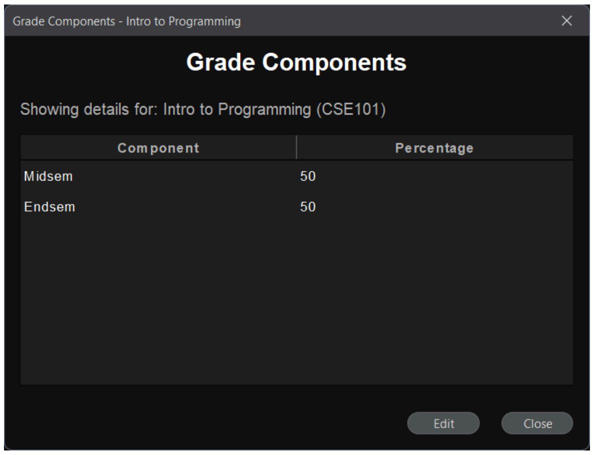
Design & Storage

- **No Fixed Columns:** Instead of fixed SQL columns like Midterm_Score, the Offerings table uses a **JSON column** named GradeComponents.
  - *Example:* {"Quiz 1": 10, "Midterm": 30, "Final Project": 60}
- **Data Handling:** OfferingHandler.java uses the **Jackson** library (ObjectMapper) to serialize/deserialize this JSON data into Java Maps.
- **Final Grade Calculation**: The final grade for a course is determined by calculating the sum of the marks scored in the individual components, and comparing the total against the grade slabs defined by the instructor.
- **SGPA Calculation**: SGPA is calculated using the following formula

SGPA = **Sum of (**Credits of Course * Grade Point in Course**)** / Total Credits

Workflow

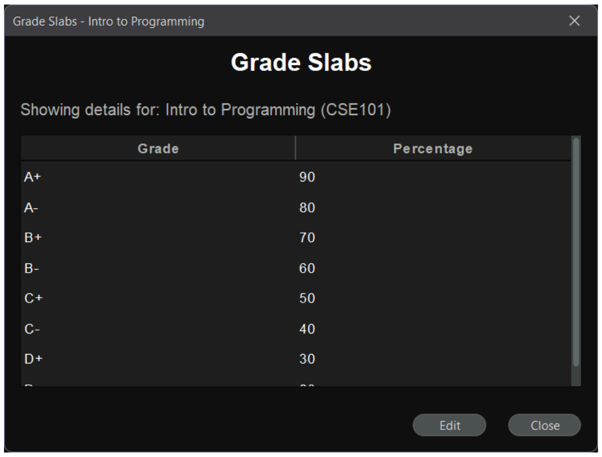
1. **Setup:** Instructors define components in GradeComponentsDialog.java. The system ensures the total weight equals 100%.
2. **Grading:** In AssignGradesDialog.java, the table dynamically generates columns based on the JSON definition.
3. **Import/Export:** Instructors can export the grade sheet to CSV, fill it offline, and import it back. The import logic (ImportGradesFromCSV) is smart—it automatically updates the course structure if the CSV headers differ from the current settings.



Grade Components

Showing details for: Intro to Programming (CSE101)

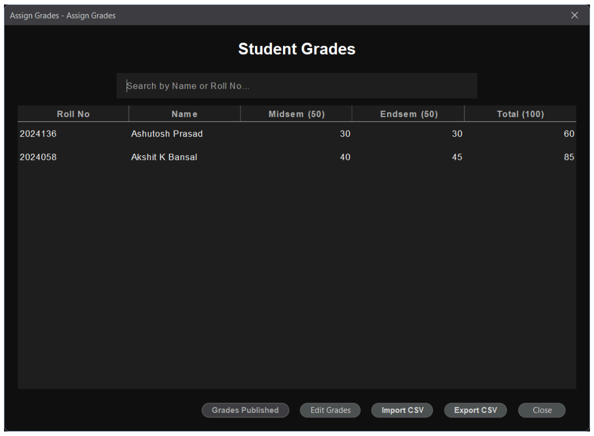
Component	Percentage
Midsem	50
Endsem	50



Grade Slabs

Showing details for: Intro to Programming (CSE101)

Grade	Percentage
A+	90
A-	80
B+	70
B-	60
C+	50
C-	40
D+	30



Student Grades

Roll No	Name	Midsem (50)	Endsem (50)	Total (100)
2024136	Ashutosh Prasad	30	30	60
2024058	Akshit K Bansal	40	45	85

---

5. User Management (Admin Module)

The Admin is responsible for managing the institution's users.

Unified Creation Flow

The OnAddUser method handles the addition of new users and the different data requirements for different roles:

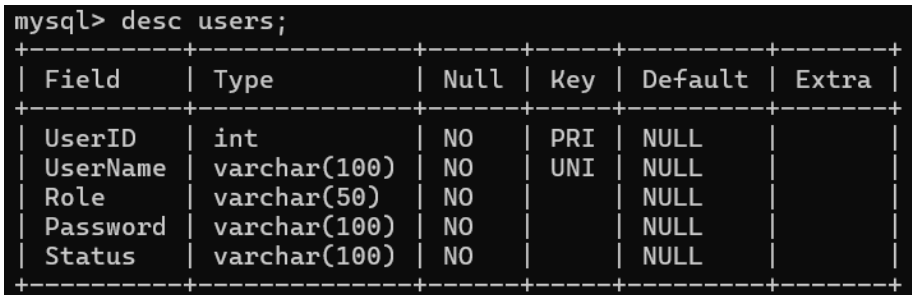
- **Student:** Requires Roll Number, Program, and Year.
- **Instructor:** Requires Qualification, Department, and Joining Date.
- **Admin:** Basic credentials only.

Security in Creation

The UserRoleFactory class encapsulates the logic for creating the base User record. Once the user ID is generated, the specific handler (StudentHandler or InstructorHandler) inserts the profile details into the respective child tables (Students or Instructors). This ensures referential integrity across the database.

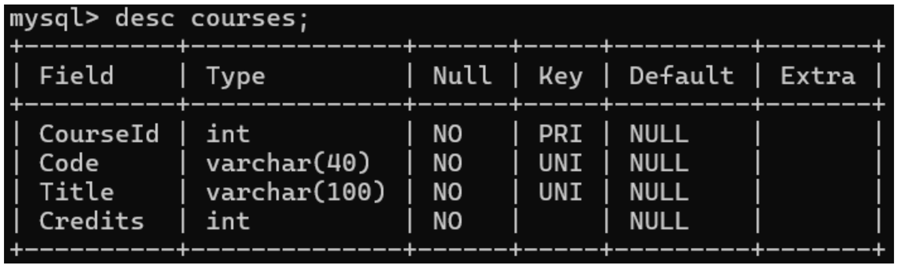
6. Database Schema Overview

- **auth_db:** users

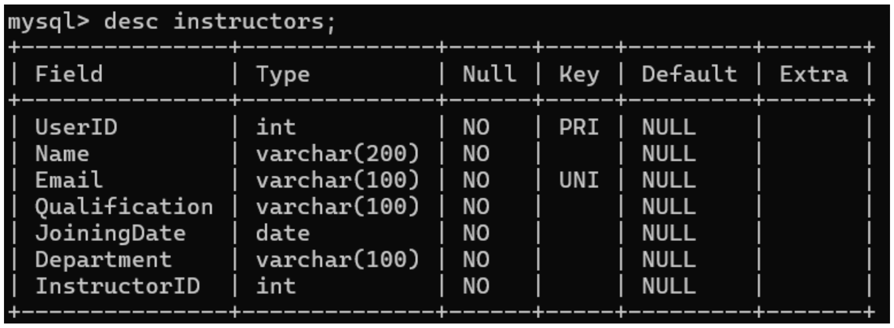


```
mysql> desc users;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| UserID   | int          | NO   | PRI | NULL    |       |
| UserName | varchar(100) | NO   | UNI | NULL    |       |
| Role     | varchar(50)  | NO   |     | NULL    |       |
| Password | varchar(100) | NO   |     | NULL    |       |
| Status   | varchar(100) | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

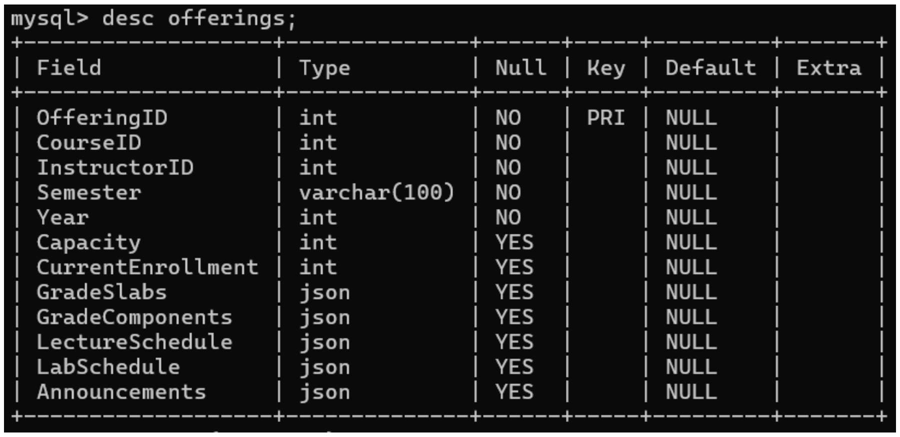
- **erp_db:** courses, instructors, offerings, registrations, semesters, studentgraderecord, studentnotifications, students



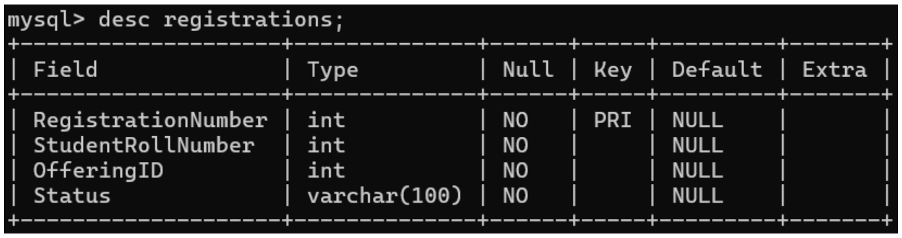
```
mysql> desc courses;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| CourseId | int          | NO   | PRI | NULL    |       |
| Code     | varchar(40)  | NO   | UNI | NULL    |       |
| Title    | varchar(100) | NO   | UNI | NULL    |       |
| Credits  | int          | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```



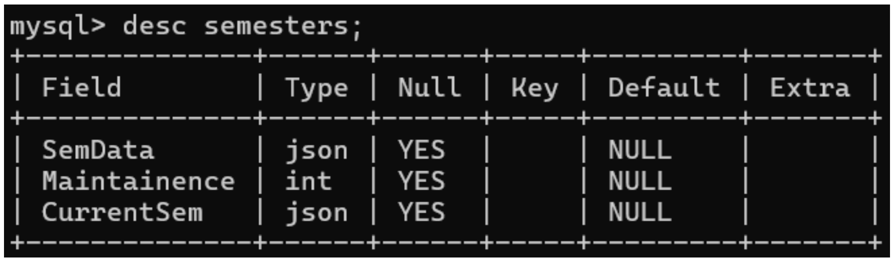
```
mysql> desc instructors;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| UserID        | int          | NO   | PRI | NULL    |       |
| Name          | varchar(200) | NO   |     | NULL    |       |
| Email         | varchar(100) | NO   | UNI | NULL    |       |
| Qualification | varchar(100) | NO   |     | NULL    |       |
| JoiningDate   | date         | NO   |     | NULL    |       |
| Department    | varchar(100) | NO   |     | NULL    |       |
| InstructorID  | int          | NO   |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```



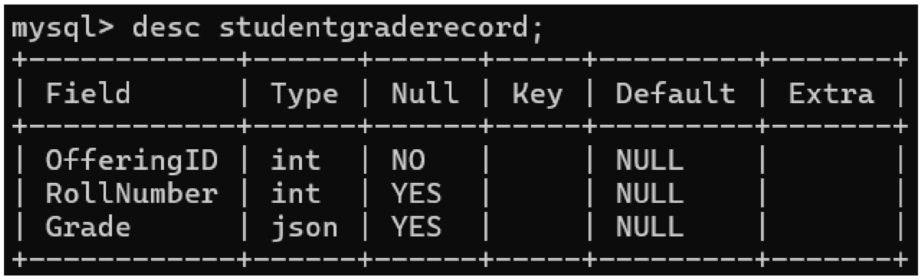
```
mysql> desc offerings;
+-------------------+--------------+------+-----+---------+-------+
| Field             | Type         | Null | Key | Default | Extra |
+-------------------+--------------+------+-----+---------+-------+
| OfferingID        | int          | NO   | PRI | NULL    |       |
| CourseID          | int          | NO   |     | NULL    |       |
| InstructorID      | int          | NO   |     | NULL    |       |
| Semester          | varchar(100) | NO   |     | NULL    |       |
| Year              | int          | NO   |     | NULL    |       |
| Capacity          | int          | YES  |     | NULL    |       |
| CurrentEnrollment | int          | YES  |     | NULL    |       |
| GradeSlabs        | json         | YES  |     | NULL    |       |
| GradeComponents   | json         | YES  |     | NULL    |       |
| LectureSchedule   | json         | YES  |     | NULL    |       |
| LabSchedule       | json         | YES  |     | NULL    |       |
| Announcements     | json         | YES  |     | NULL    |       |
+-------------------+--------------+------+-----+---------+-------+
```



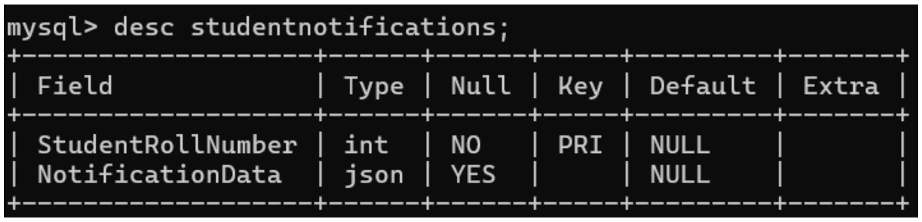
```
mysql> desc registrations;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| RegistrationNumber | int          | NO   | PRI | NULL    |       |
| StudentRollNumber  | int          | NO   |     | NULL    |       |
| OfferingID         | int          | NO   |     | NULL    |       |
| Status             | varchar(100) | NO   |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
```



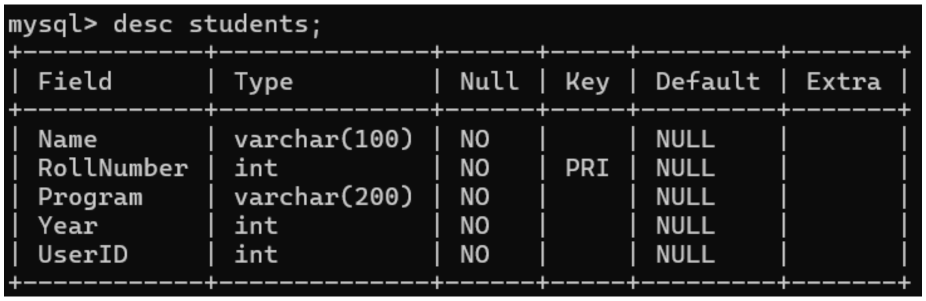
```
mysql> desc semesters;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| SemData      | json | YES  |     | NULL    |       |
| Maintainence | int  | YES  |     | NULL    |       |
| CurrentSem   | json | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
```



```
mysql> desc studentgraderecord;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| OfferingID | int  | NO   |     | NULL    |       |
| RollNumber | int  | YES  |     | NULL    |       |
| Grade      | json | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
```



```
mysql> desc studentnotifications;
+-------------------+------+------+-----+---------+-------+
| Field             | Type | Null | Key | Default | Extra |
+-------------------+------+------+-----+---------+-------+
| StudentRollNumber | int  | NO   | PRI | NULL    |       |
| NotificationData  | json | YES  |     | NULL    |       |
+-------------------+------+------+-----+---------+-------+
```



```
mysql> desc students;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Name       | varchar(100) | NO   |     | NULL    |       |
| RollNumber | int          | NO   | PRI | NULL    |       |
| Program    | varchar(200) | NO   |     | NULL    |       |
| Year       | int          | NO   |     | NULL    |       |
| UserID     | int          | NO   |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
```