**1. User Entity**

* **Purpose:** Represents system users, such as super admins and school admins.
* **Fields:**
  + name: User's name.
  + email: User's email (unique).
  + password: Encrypted password (hashed with bcrypt).
  + role: Numeric role identifier (e.g., 1 for superadmin, 2 for school admin).
  + schoolIds: Array of school IDs the user has access to.
* **Relationships:**
  + A user can be associated with multiple schools via schoolIds.
  + Super Admin schoolIds array will be empty because they will have access to all the schools.

**2. School Entity**

* **Purpose:** Represents schools in the system.
* **Fields:**
  + name: Name of the school.
  + address: Address of the school.
  + createdBy: Reference to the user who created the school (likely a superadmin).
  + classIds: Array of class IDs associated with the school.
* **Relationships:**
  + Each school is created by a user (createdBy references the User entity).
  + A school can have multiple classes (classIds).

**3. Class Entity**

* **Purpose:** Represents classes within schools.
* **Fields:**
  + name: Name of the class (e.g., "Grade 1").
  + capacity: Maximum number of students in the class.
  + resources: Array of resources available in the class (e.g., ["Books", "Projector"]).
  + studentIds: Array of student IDs enrolled in the class.
  + schoolId: The school to which this class belongs.
* **Relationships:**
  + Each class belongs to a school (schoolId references the School entity).
  + A class can have multiple students (studentIds) till it’s capacity.

**4. Student Entity**

* **Purpose:** Represents students enrolled in classes.
* **Fields:**
  + name: Student's name.
  + age: Student's age.
  + gender: Gender of the student.
  + classId: The class the student is enrolled in.
* **Relationships:**
  + Each student belongs to a class (classId references the Class entity).

**Entity Relationships Summary**

1. **User ↔ School:**
   * A user can manage multiple schools via the schoolIds array in the User entity.
2. **School ↔ Class:**
   * A school can have multiple classes (classIds in the School entity).
   * Each class belongs to one school (schoolId in the Class entity).
3. **Class ↔ Student:**
   * A class can have multiple students (studentIds in the Class entity).
   * Each student belongs to one class (classId in the Student entity).

**Database Schema Relationships Visualization**

User

|--< School

|--< Class

|--< Student

**Usage Example**

* **User Roles:**
  + A superadmin creates schools.
  + A school admin manages schools and classes within their assigned schoolIds.
* **Adding a Class:**
  + Check if the schoolId exists in the school collection.
  + Add a class to the school and reference the classId in the School document.
* **Enrolling a Student:**
  + Check if the classId exists in the class collection.
  + Add the student to the Class document's studentIds.
  + If the class capacity is reached than student cannot be added.