

ENHANCED INSTRUCTOR MANAGEMENT SYSTEM

PHASE 1: ADD ROLES TO DATABASE (Optional Enhancement)

If you want formal instructor-student separation, run this migration:

```
-- Add role column to User table
ALTER TABLE "User" ADD COLUMN role VARCHAR(20) DEFAULT 'student';

-- Add instructor-student relationship table
CREATE TABLE "InstructorStudent" (
    id TEXT PRIMARY KEY DEFAULT gen_random_uuid()::text,
    "instructorId" TEXT NOT NULL,
    "studentId" TEXT NOT NULL,
    "assignedAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT "InstructorStudent_instructorId_fkey"
        FOREIGN KEY ("instructorId") REFERENCES "User"(id) ON DELETE CASCADE,
    CONSTRAINT "InstructorStudent_studentId_fkey"
        FOREIGN KEY ("studentId") REFERENCES "User"(id) ON DELETE CASCADE,
    CONSTRAINT "InstructorStudent_instructorId_studentId_key"
        UNIQUE ("instructorId", "studentId")
);

-- Update existing test user to instructor
UPDATE "User" SET role = 'instructor' WHERE email = 'john@doe.com';
```

PHASE 2: INSTRUCTOR-STUDENT LINKING

Option A: Auto-assign all students to all instructors (Simple)

```
-- Assign all students to all instructors
INSERT INTO "InstructorStudent" ("instructorId", "studentId")
SELECT i.id as "instructorId", s.id as "studentId"
FROM "User" i CROSS JOIN "User" s
WHERE i.role = 'instructor' AND s.role = 'student';
```

Option B: Manual assignment (Controlled)

```
-- Assign specific students to specific instructors
INSERT INTO "InstructorStudent" ("instructorId", "studentId")
VALUES
    ('instructor_uuid_here', 'student_uuid_here'),
    ('instructor_uuid_here', 'another_student_uuid_here');
```

 **PHASE 3: UPDATE ANALYTICS TO FILTER BY INSTRUCTOR**

Modify the analytics API to show only assigned students per instructor.