Learning Management System Report

**1. Marking all notifications as read instead of just one (feature)**

**Description:**

Introduce a functionality that allows users to mark all their unread notifications as read in a single action. This enhances user experience by providing a convenient way to manage notifications, especially when dealing with a large number.

**Expected Result:**

* All notifications associated with the user are marked as read.
* The user's notification list reflects the updated read status.

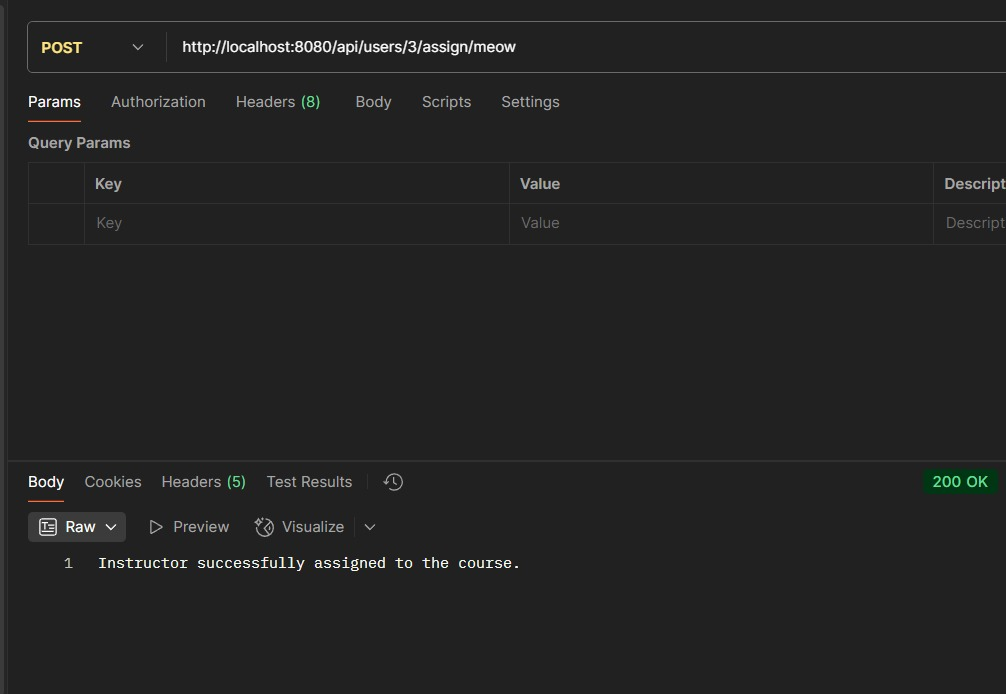
**Affected Classes:**

* NotificationController.markAllNotificationsAsRead(Long userId)
* NotificationService.markAllNotificationsAsRead(Long userId)

**Benefits:**

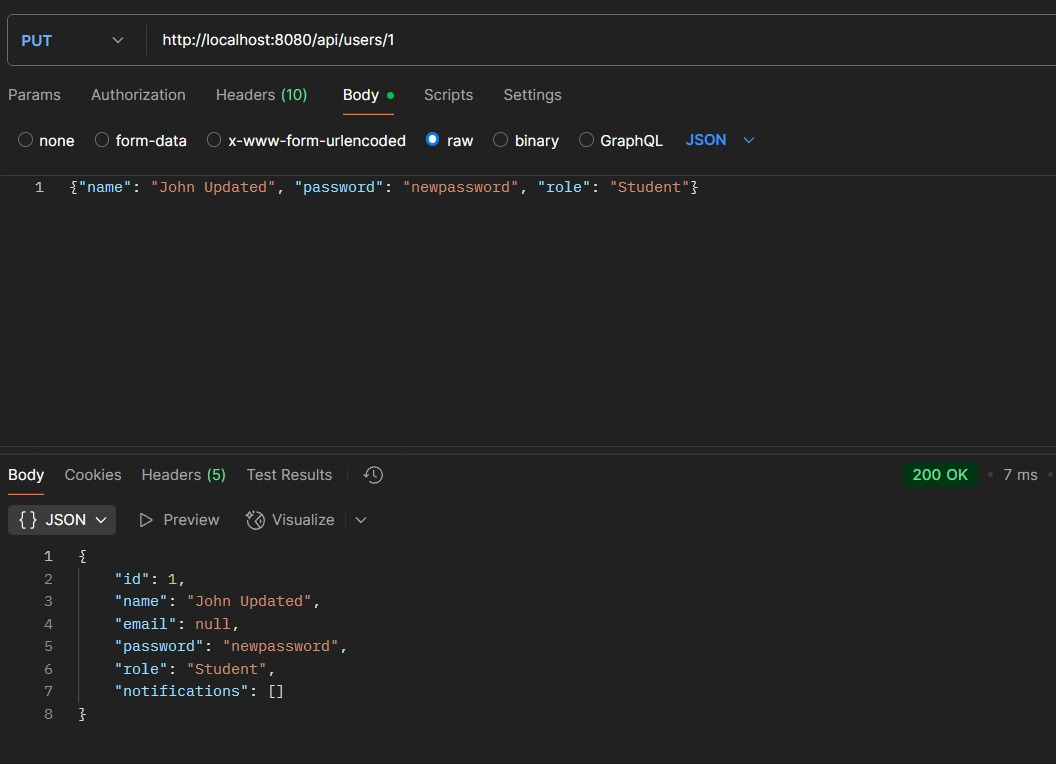
* Improves user satisfaction by simplifying notification management.
* Reduces the need for users to mark each notification individually.

**2. Assigning Instructor to Non-Existent Course (bug)**

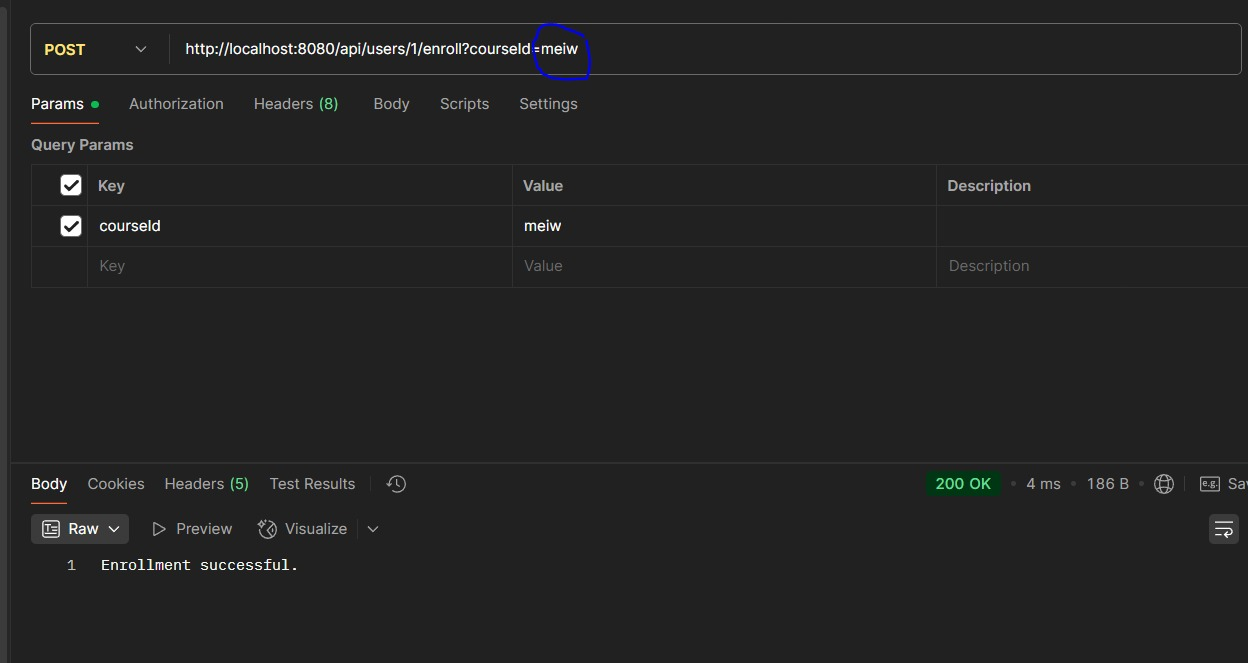
* **Endpoint:** POST /api/courses?instructorId={id}
* **Steps to Reproduce:**
  1. Choose instructorId=1 and payload { "title": "New Course" }.
  2. Provide a courseId in payload that doesn’t exist or omit it.
* **Actual Result:** Course is created or updated without validating the provided instructorId/courseId relation.
* **Expected Result:** If courseId exists, update; if not, error. Similarly, binding an instructor to an unknown course should return 404 Not Found.
* **Affected Classes:**
  1. CourseController.createCourse(CourseDto dto, Long instructorId)
  2. CourseService.create(...)
* **Root Cause:** No check in CourseService to verify the course (or instructor-course link) exists before assignment.
* **Recommended Fix:** In CourseService, before saving, call:

**3. PUT Request Nullifies Unspecified Sections (bug)**

* **Endpoint:** PUT /api/users/{id}
* **Steps to Reproduce:**
  1. Create user with multiple fields (name, email, sections).
  2. Send PUT /api/users/5 with body { "email": "new@example.com" }.
* **Actual Result:** All other fields (including sections) are set to null in the database.
* **Expected Result:** Only email changes; other fields remain intact.
* **Affected Classes:**
  1. UserController.updateUser(User dto, Long id)
  2. UserService.updateUser(User dto, Long id)
* **Root Cause:** Full entity overwrite via userStore.put(id, dto) without merging existing data.
* **Recommended Fix:** Merge non-null fields



**4. Enrollment Success into Non-Existent Course (bug)**

* **Endpoint:** POST /api/courses/{courseId}/enroll or similar
* **Steps to Reproduce:**
  1. Send request for courseId=9999 (no such course).
  2. Payload: { "studentId": 2 }.
* **Actual Result:** API returns 200 OK and a success message.
* **Expected Result:** 404 Not Found or 400 Bad Request indicating invalid course.
* **Affected Classes:**
  1. EnrollmentController.enroll(Long courseId, EnrollmentRequest dto)
  2. EnrollmentService.enrollStudent(...)
* **Root Cause:** No check for course existence; enrollStudent blindly adds to in-memory list or database.
* **Recommended Fix:** In EnrollmentService, validate before proceeding to enroll.