Project Context for Cursor

We are building a modern LMS and course management system focused on making assignment and file handling seamless for both teachers and students. Our goal is to create a platform where teachers can easily create, edit, and organize assignments directly from the course homepage, attach relevant files (with optional protection/watermarking), and ensure all course materials are automatically accessible in the main course files section. Assignments should have dedicated “splash pages” showing all details and attached files, with consistent navigation and a persistent course bar for easy access back to course home.

File management must be tightly integrated: files uploaded for assignments should always appear in course files, and any change or deletion must be reflected everywhere (one source of truth). Protected files must only show their secure version to students, while teachers can see which files are protected. Only published assignments are visible to students and populate their “to-do” lists.

We are prioritizing teacher workflows for MVP—making assignment creation, editing, and navigation as smooth as possible—before adding student-side enhancements and submission/grading features. All UI and API choices should aim for long-term maintainability and an intuitive experience.

Additional Notes for Cursor

- Use your best judgment on API naming, endpoint usage, and frontend routing based on existing codebase structure.
- If any required endpoint is missing, generate the controller/route/model code for it, following existing patterns.
- Prioritize clear, modular code and add helpful comments for future maintainers.
- Always prefer single-click navigation for opening assignment splash pages.
  For any ambiguity, refer to the detailed conversation with the teacher for desired behaviors.

Step 1: Ensure Assignment Population on Course Home

Goal:
Assignments should display correctly on each course’s homepage in an “Assignments” section—listing all assignments related to that course.

Details to Cover:

- Each course page must fetch and display its assignments using the backend endpoint (e.g., `GET /courses/{courseId}/assignments` or as per backend structure).

- Each assignment in the list shows at least: title, due date, category, and possibly an indicator if files are attached.
- The assignments preview/list on the course homepage must match the actual assignments in the database—check for data inconsistencies or stale cache.

Nuances / Things to Check:

- Assignments created in the gradebook page *must also show up* in the course homepage assignment section (and vice versa).
- Ensure there is **one source of truth**: No duplication between gradebook and course assignment list.
- If you have assignment “status” (draft/published), make sure only published assignments show to students.

Task for Cursor:

- Check existing endpoints/controllers for assignment fetching and confirm they’re correctly populating the assignments list for each course homepage.
- If assignments are missing or not syncing, debug why (e.g., filtering, wrong course ID, caching).
- Add clear code comments explaining the data flow from backend → frontend for assignments population on course pages.

Step 2: Enable Assignment Creation and Editing on Course Homepage

Goal:
Allow teachers to create new assignments directly from the course homepage (not just the gradebook page), and edit assignments there too.

Details to Cover:

- Add a “Create Assignment” button/option in the assignments section of each course homepage.

- The assignment creation form should include:

    - Title
    - Description/instructions
    - Due date (+ optional time)
    - Category

    - Total points
    - File upload (with per-file “protect” toggle)

- Upon creation, assignment should be saved to the backend and immediately reflected in the assignments list on both course homepage and gradebook page.
- Assignment status should default to “draft” and only appear for students after teacher publishes.
- Allow teachers to edit existing assignments from the course homepage (edit button on each assignment row/card).

Nuances / Things to Check:

- File uploads: Support uploading files during assignment creation and attaching more files after creation. For each file, allow teacher to toggle “protect” before upload.
  If a file is uploaded as part of an assignment, it must also appear in the course’s main files section (with an indicator showing the assignment it’s attached to).
- Editing assignments should allow updating all assignment fields and removing/adding files, but if a file’s protection setting needs to change, teacher must delete and re-upload it.
- Deleting a file from either the assignment or course files deletes it everywhere.

Task for Cursor:

- Implement assignment creation form and edit form on the course homepage, reusing backend endpoints as needed.

- Make sure file handling logic is shared and consistent between assignments and general course files.
  Ensure all UI actions (create, edit, delete) reflect immediately in the assignments list.
- Add clear error handling and feedback for teachers (e.g., failed upload, missing fields).

Step 3: Implement Assignment Splash Pages

Goal:
On clicking an assignment from the assignments list on a course homepage, navigate to a dedicated “splash page” for that assignment.

Details to Cover:

- Navigating to an assignment splash page should use a unique route (e.g., `/courses/{courseId}/assignments/{assignmentId}`).

- The splash page must display:

  - Persistent course bar at the top (clickable to return to course home)
    Assignment title
  - Due date (+ time if provided)
  - Category
  - Total points
  - Description/instructions
  - List of attached files (with upload date, and “protected” indicator *visible to teachers only*)
  - Option to delete assignment or attached files
- (Optional: breadcrumb navigation—e.g., “AP Calc > Assignments > Final Exam”)

Nuances / Things to Check:

- Assignments should only be accessible by users with teacher permissions.
- Files attached to the assignment must show their “protected” status for teachers; students never see this indicator.
- Clicking the persistent course bar always returns to course home, not just previous page.
- When deleting a file from the splash page, remove from all file listings everywhere.
- Assignments only show up in the student’s to-do list after being published.

Task for Cursor:

- Implement routing for assignment splash pages as described.
- Ensure data fetching for a single assignment’s full details is efficient (use `GET /assignments/{assignmentId}` or as needed).
- Make sure UI is responsive and matches the rest of the course UI/branding.
- Test navigation/back logic thoroughly.

Step 4: Ensure File/Assignment Sync & Consistency

Goal:
Make sure assignment-attached files and course files always stay in sync, and protection logic is enforced everywhere.

Details to Cover:

- Files uploaded in an assignment must show up in the course’s files section (with assignment link/indicator).
- If a file is deleted from one place, it disappears everywhere.
- If a file is protected, students always see only the protected version, everywhere. Students cannot see any indicator of protection status; teachers see “protected” status on their views.

Task for Cursor:

- Audit file handling code for “one source of truth” and consistency across course files and assignment attachments.
- Make sure edge cases (e.g., deleting an assignment, bulk deleting files) don’t break the data integrity.
- Ensure proper permissions are checked throughout.
