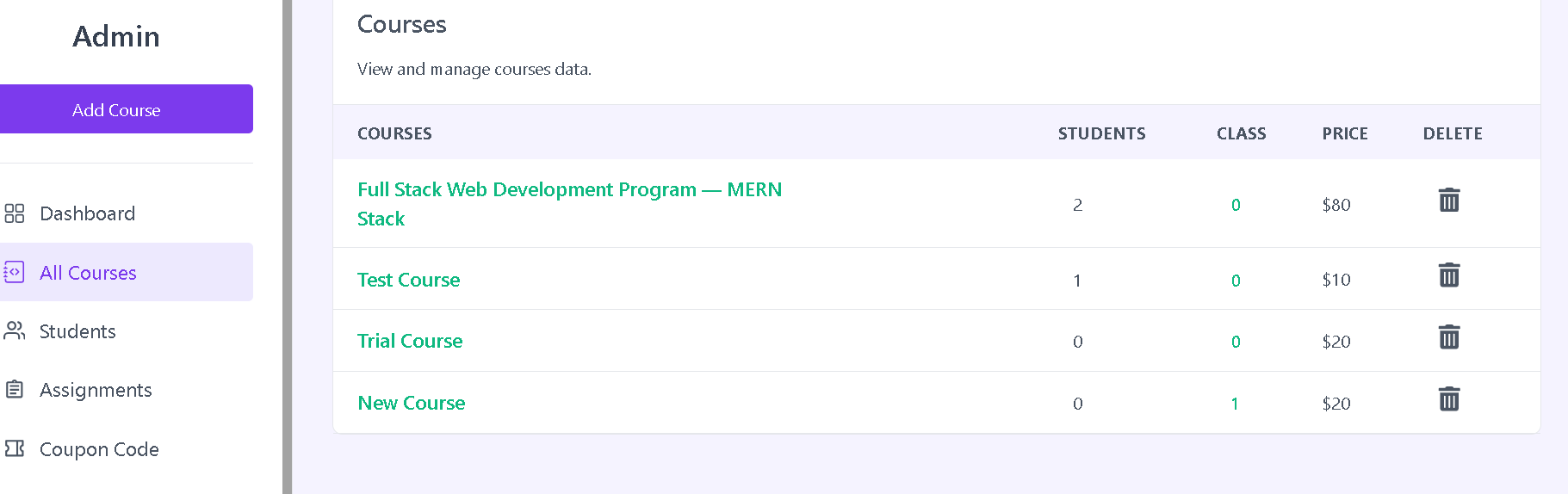
**All Courses** [**https://citria.org/admin/all-courses**](https://citria.org/admin/all-courses)

****

The **AllCourses** component is designed to display a list of courses, allowing administrators to view, manage, and potentially delete courses. Let's break down its structure and features:

Fetching Data:

Uses the useGetAllCourses custom hook to fetch the list of courses. The loading variable indicates whether the data is still being fetched.

State Management:

Manages state variables like content to track the currently selected course for detailed viewing and isDelete to confirm course deletion.

Uses setIsDelete to set the course ID when the delete action is triggered.

Delete Course Functionality:

Provides a delete button for each course, triggering a confirmation modal when clicked.

The delete confirmation modal allows users to cancel or proceed with the deletion.

Delete Course Function (handleDelete):

Sends a DELETE request to the server using Axios when the user confirms the deletion.

Displays success or error messages using the showToast function from the context.

If the deletion is successful, resets the isDelete state variable.

Viewing Course Details:

Allows administrators to click on a course title to view more details about the course. This triggers the setContent function.

Loading Spinner:

Displays a loading spinner (Spinner component) while data is being fetched or during the course deletion process.

Styling and UI Elements:

Utilizes Tailwind CSS for styling, including responsive table styling, button styles, and modal layouts.

Course Content Page:

If content is not null, renders the CourseContent component to display detailed information about the selected course.

Empty State Handling:

If there are no courses and data loading is complete, displays a message indicating that no courses have been created yet.

Page Title:

Sets the page title to "All Courses" using the PageTitle component.

In summary, the AllCourses component provides a comprehensive interface for administrators to manage courses, including viewing course details, deleting courses, and handling loading states. The use of modals enhances user experience and ensures confirmation before critical actions like course deletion.

The **CourseContent** component is responsible for displaying detailed information about a selected course, including its modules and associated videos. Let's break down its structure and features:

State Management:

Utilizes the useState hook to manage state variables such as moduleIndex, videoIndex, and cntIndex for tracking the currently selected module, video, and module's content list index.

Navigation Functions:

handleNextVideo: Advances to the next video within the current module or moves to the next module if the current video is the last one.

handlePreviousVideo: Navigates to the previous video within the current module or goes to the last video of the previous module if the current video is the first one.

Rendering Course Content:

Displays the course title, total module count, and information about the currently selected video within a module.

Includes video playback using the <video> HTML element with controls, autoplay, fullscreen, and nodownload attributes.

Navigation Buttons:

Provides "Previous" and "Next" buttons for navigating between videos within a module.

Button states are managed to disable navigation when at the beginning or end of the videos or modules.

Course Content List:

Renders a list of modules and their associated videos in a responsive layout.

Allows expanding/collapsing each module's content list by clicking on the module title.

Styling and UI Elements:

Utilizes Tailwind CSS for styling, including button styles, layout structure, and responsiveness.

Uses icons such as MdKeyboardArrowDown for module expansion and MdOutlineArrowBackIos for the "Back" button.

Back Button:

Provides a "Back" button for returning to the previous view.

Empty State Handling:

Displays messages when there are no modules or videos available for the selected course.

Conditional Rendering:

Adjusts the display based on whether videos or modules are present, providing informative messages when necessary.

In summary, the CourseContent component enhances the user experience by allowing users to explore the content of a course in a structured and navigable manner, complete with video playback and interactive module listings.