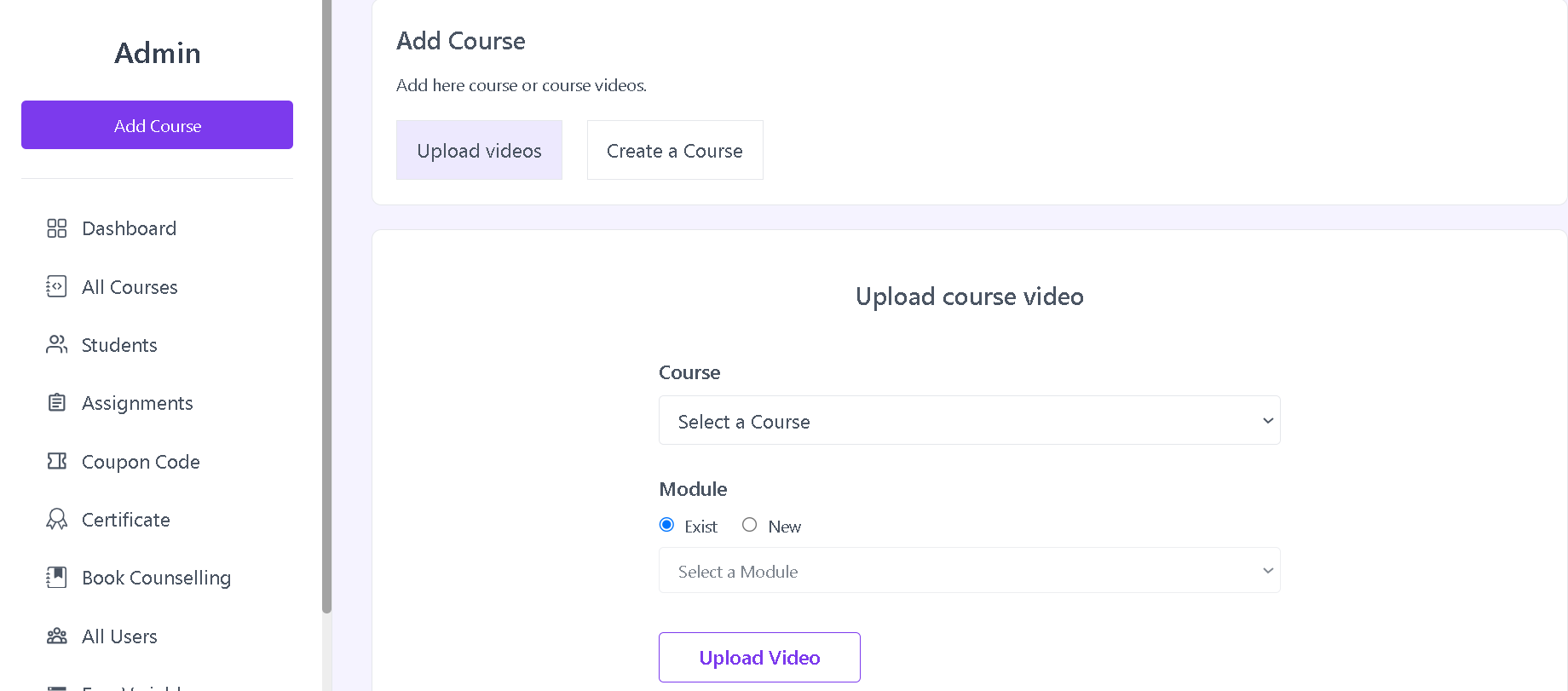
**AddCourse** [**https://citria.org/admin/add-course**](https://citria.org/admin/add-course)

****

The **AddCourses** component serves as a container for adding new courses or uploading course videos. Let's go through its features and structure:

State Management:

Utilizes the useState hook to manage the state variable upload, which determines whether the user intends to upload videos or create a new course.

Page Title:

Includes a PageTitle component, presumably for displaying the title of the page. However, the title is currently set to an empty string (title={''}).

Main Content Section:

Consists of a section with a border, rounded corners, and a white background to visually separate it from the rest of the page.

Header Section:

Contains a heading (<h3>) indicating the purpose of the section, such as "Add Course."

Includes a paragraph (<p>) providing a brief description or instructions for adding courses or videos.

Upload Method Selection:

Provides two buttons for selecting the type of upload:

"Upload videos": Sets the upload state to "video" when clicked.

"Create a Course": Sets the upload state to "course" when clicked.

The selected upload method is highlighted with a different background color.

Conditional Rendering:

Renders the appropriate content based on the selected upload method:

If upload is "course," the CreateACourse component is rendered.

If upload is "video," the UploadVideos component is rendered.

CreateACourse Component:

Displays the CreateACourse component when the user selects the "Create a Course" option. This component likely handles the creation of new courses.

UploadVideos Component:

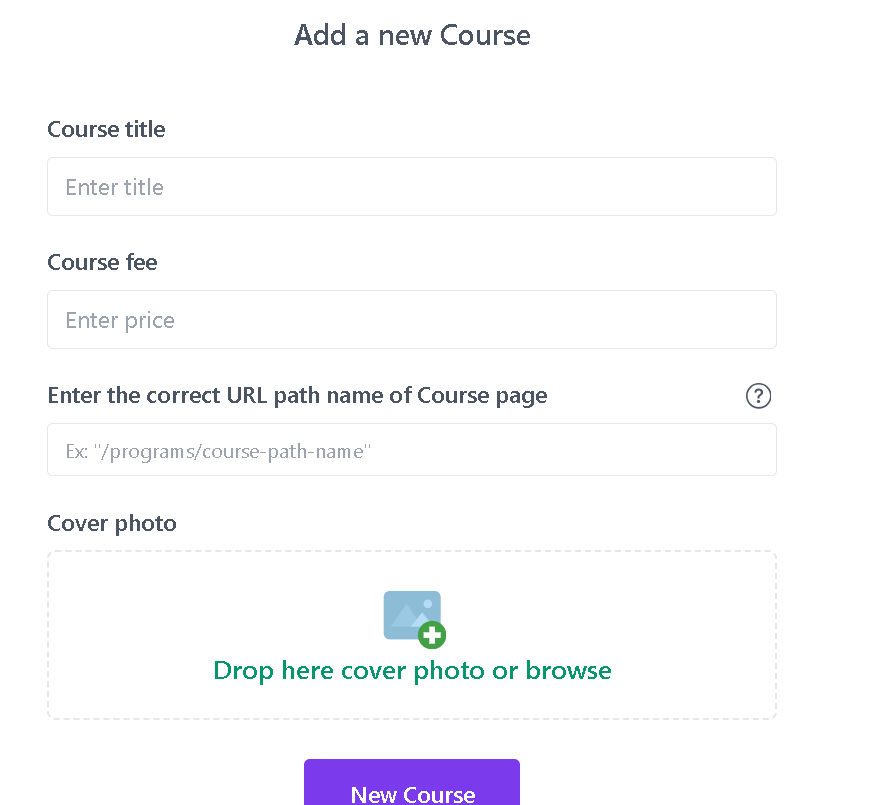
Displays the UploadVideos component when the user selects the "Upload videos" option. This component likely handles the process of uploading course videos.

Styling and UI Elements:

Applies styling using Tailwind CSS classes for buttons, spacing, and visual elements.

In summary, the AddCourses component provides a user-friendly interface for users to choose between creating a new course or uploading course videos. It dynamically renders the corresponding content based on the user's selection, facilitating a seamless and intuitive experience.

**Create Course**

****

The **CreateACourse** component in React is designed to facilitate the creation of a new course by providing a form for users to input relevant information. Let's break down its features:

State Management:

Utilizes the useState hook to manage various state variables such as error, loading, openPathImg, coverPhoto, and dragActive.

Page Title:

Includes a PageTitle component, setting the title to "Add a new course title."

Form Structure:

Contains a form that allows users to input information about the new course.

Input Fields:

Course Title: Accepts a text input for the title of the course.

Course Fee (Price): Accepts a numeric input for the price of the course.

URL Path (src\_path): Accepts a text input for the URL path of the course page.

Cover Photo: Allows users to upload a cover photo for the course, with drag-and-drop functionality.

Error Handling:

Displays error messages below the respective input fields if validation fails.

Errors include handling for missing course title, price, URL path, and cover photo.

Drag and Drop Feature:

Provides a drag-and-drop area for uploading a cover photo.

Changes styles when the user is actively dragging a file.

Help Icon for URL Path:

Includes a question mark icon that, when clicked, provides information about the correct format for the URL path.

Submission Button:

A submit button triggers the form submission.

Displays a loading spinner when the form is being processed to indicate ongoing activity.

Event Handlers:

handleCoverPhoto: Handles the selection of a cover photo from the file input.

handleDrag: Handles events related to drag-and-drop functionality.

handleDrop: Handles the drop event after dragging a file into the designated area.

Form Submission:

Sends a POST request to the server with the entered course information, including title, price, URL path, and cover photo.

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

If successful, triggers a change in the upload mode by invoking the setUpload function.

Styling:

Uses Tailwind CSS classes for styling, providing a clean and responsive design.

In summary, the CreateACourse component encapsulates the process of creating a new course, guiding users through the input of essential details and handling various aspects of the form submission process.

**Success Message**

The SuccessMessage component is a React component designed to display a success message when a course has been added successfully. Let's break down its structure and functionality:

Structure:

The component is comprised of a div containing a flex container (flex items-center flex-col) to align its contents vertically at the center.

Checkmark SVG:

Includes an SVG element with a class of "checkmark" that depicts a checkmark inside a circle.

The SVG has two components:

A circle (checkmark\_\_circle) representing the background circle.

A path (checkmark\_\_check) representing the checkmark itself.

Heading and Text:

Contains an h3 element with a class specifying text styles (text-xl font-medium text-emerald-500 mb-1).

Displays the text "The Course Added Successfully" as a success message.

Conditional Commented-out Section:

Includes a commented-out span element with the text "Admin will be Approve soon." This suggests that there might be a future feature or additional information related to the course approval process.

Styling:

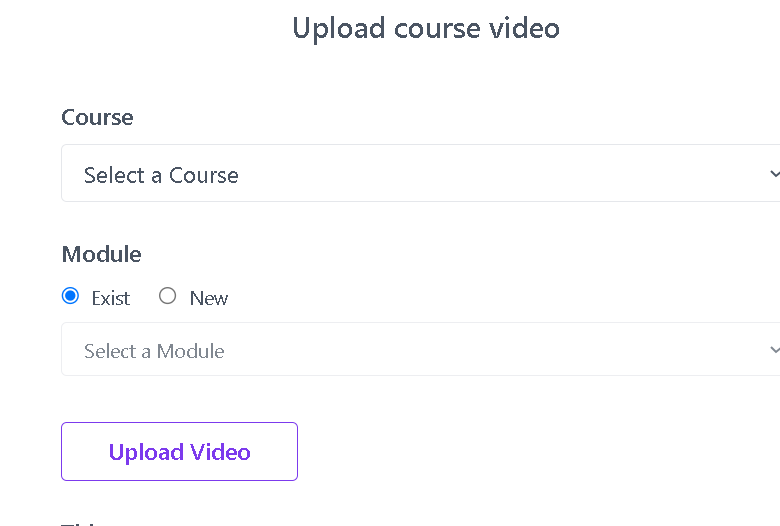
Uses Tailwind CSS classes (p-8 border-b, text-xl, font-medium, text-emerald-500, mb-1) for styling, providing padding, a border at the bottom, and a visually appealing appearance for the success message.

Responsive Design:

The component seems to be designed in a way that is visually centered and responsive, making it suitable for use in various layouts.

In summary, the SuccessMessage component is a simple and visually appealing React component that serves the purpose of informing users about the successful addition of a course. The checkmark SVG adds a visual element to emphasize the positive outcome, and the component's structure and styling contribute to a clean and effective display of success feedback.

**Upload Video**



The UploadVideos component in React is designed for handling the process of uploading videos for a course. Here's an explanation of its structure and functionality:

Initialization:

The component uses various React hooks such as useState, useEffect, useRef, and useContext for managing state and side effects.

Cloudinary Integration:

Utilizes Cloudinary for video uploads. It sets up a Cloudinary widget through the createUploadWidget function provided by Cloudinary.

Form State:

Manages form-related state using the formData and formErrors variables to store the entered data and any validation errors, respectively.

Module Selection:

Allows the user to either select an existing module or create a new one, controlled by the newModule state.

Form Validation:

Defines a validateForm function to check for errors in the form data, such as missing course/module selection, absence of a video file, and a required video title.

Form Submission:

Implements a handleSubmit function triggered upon form submission. It validates the form data, makes a request to the server to upload the video, and redirects to the course list upon success.

User Interface:

The component presents a form with fields for selecting a course, choosing or creating a module, uploading a video, specifying a title, and providing an optional description.

Error Handling:

Displays error messages for form validation issues, such as missing course/module, required video upload, and video title.

Loading Indicator:

Displays a loading spinner (<Spinner />) while waiting for the video upload or form submission to complete.

Integration with Context:

Uses the useContext hook to access the showToast function from a shared context, enabling the display of success or error messages.

Dynamic Course and Module Selection:

Dynamically populates the course and module dropdowns based on data fetched from the server using the useGetAllCourses custom hook.

In summary, the UploadVideos component encapsulates the functionality of uploading course videos, providing a user-friendly interface with dynamic options for course and module selection, real-time video uploading through Cloudinary, and effective form validation.