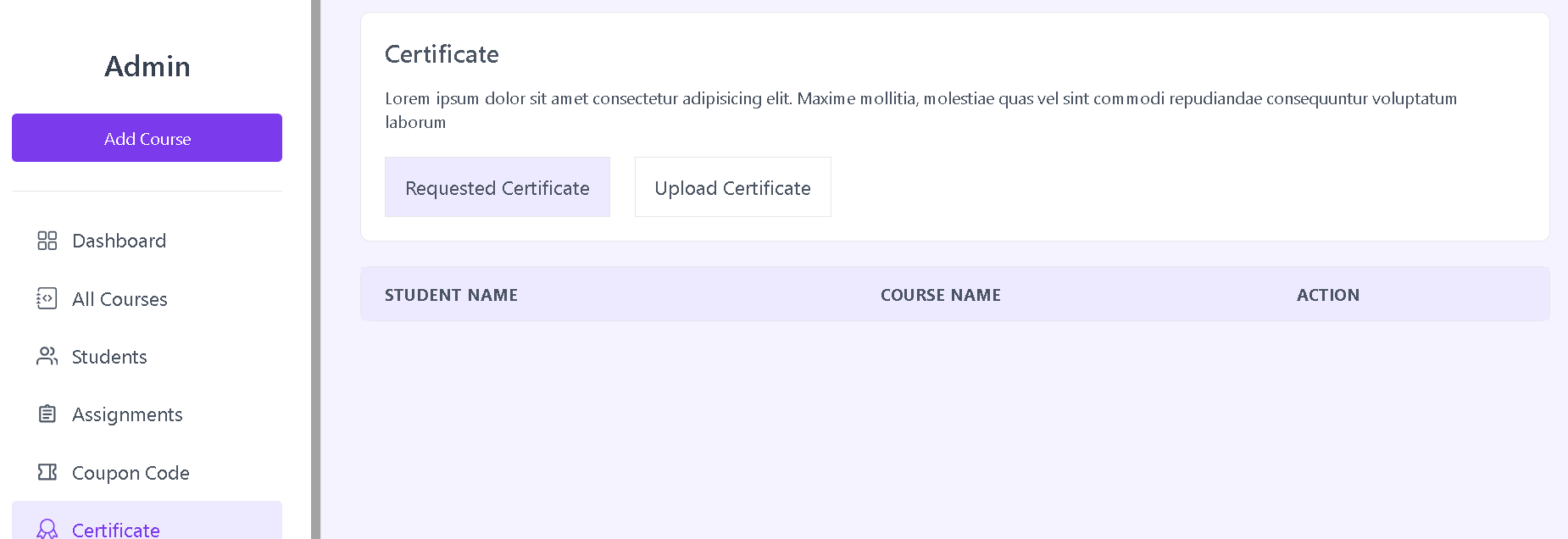
**Certificates** <https://citria.org/admin/certificate>

This component allows managing certificates in two different views:

1. It uses React state with useState hook to keep track of the active view - either "requested" or "upload".
2. It renders a title, intro text and a toggle button group to switch between:
   * Requested Certificate
   * Upload Certificate
3. Based on the state value it conditionally shows one of two components:
   * RequestedCertificate - to show certificates students have requested
   * UploadCertificate - to upload new certificates
4. The UploadCertificate component receives the state setter function as a prop. This allows it to change the state back to "requested" after uploading a certificate.

In summary, this Certificate component handles the state management and toggling between two reusable sub-components for each certificate related task. Keeping them separated this way allows reusing them easily across the application.

**Requested Certificate**

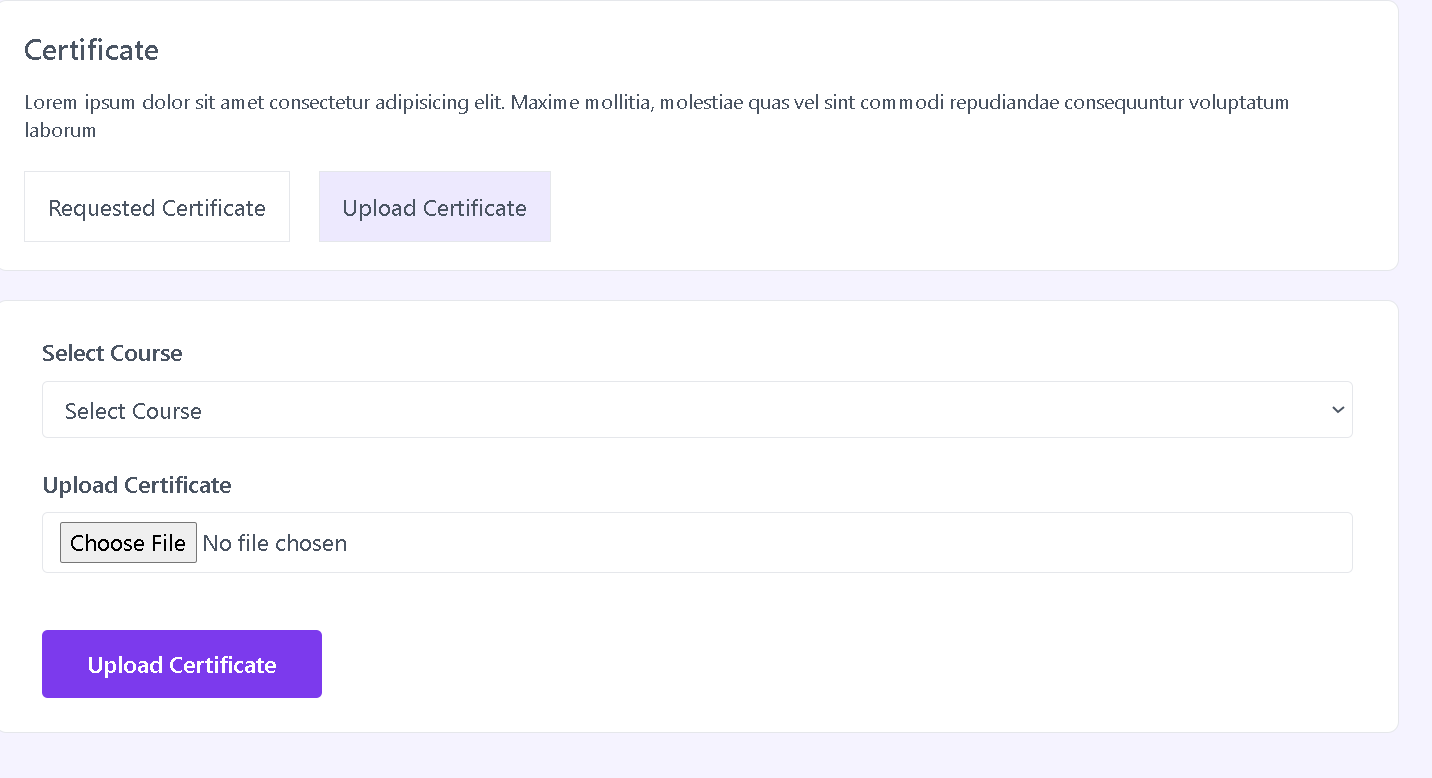
****

This component displays a list of certificates students have requested. Here are the key things:

1. It has commented out code to demonstrate downloading a certificate file from the API when clicking a button.
2. Renders a table with columns for:
   * Student Name
   * Course Name
   * Actions
3. The table data is currently hardcoded. But it would dynamically map through a list of certificate requests and display each one in a table row.
4. Each row would display the student name and course they have requested a certificate for.
5. The Actions column would have a button to download the certificate file for that student and course.
6. There is also commented out code showing how a row could be mapped from API data.

So in summary, this provides a reusable table to display certificate requests, with actions to easily download the certificate file for each student. It abstracts the downloading logic from parent components.

**Upload Certificate**



This is a React component to upload a certificate image for a course. Here are the key things it does:

1. Uses state and context API to manage form data, errors, loading state, and to show toast notifications.
2. Calls a custom hook useGetAllCourses to fetch list of courses for the select dropdown.
3. Renders a form with a course select field and file upload for the certificate image.
4. Validates on form submit to check for required fields.
5. Handles the image file preview and sets it to state on change.
6. On submit, makes API call to upload the image and updates certificate for that course.
7. Handles loading state, errors, success/error notifications.
8. Passes state setter function to parent to switch back to default view after upload.

In summary, it is a reusable certificate upload form that encapsulates all the data handling, validation, API integration and error handling that can be reused across the app. The parent component manages the overall view state.