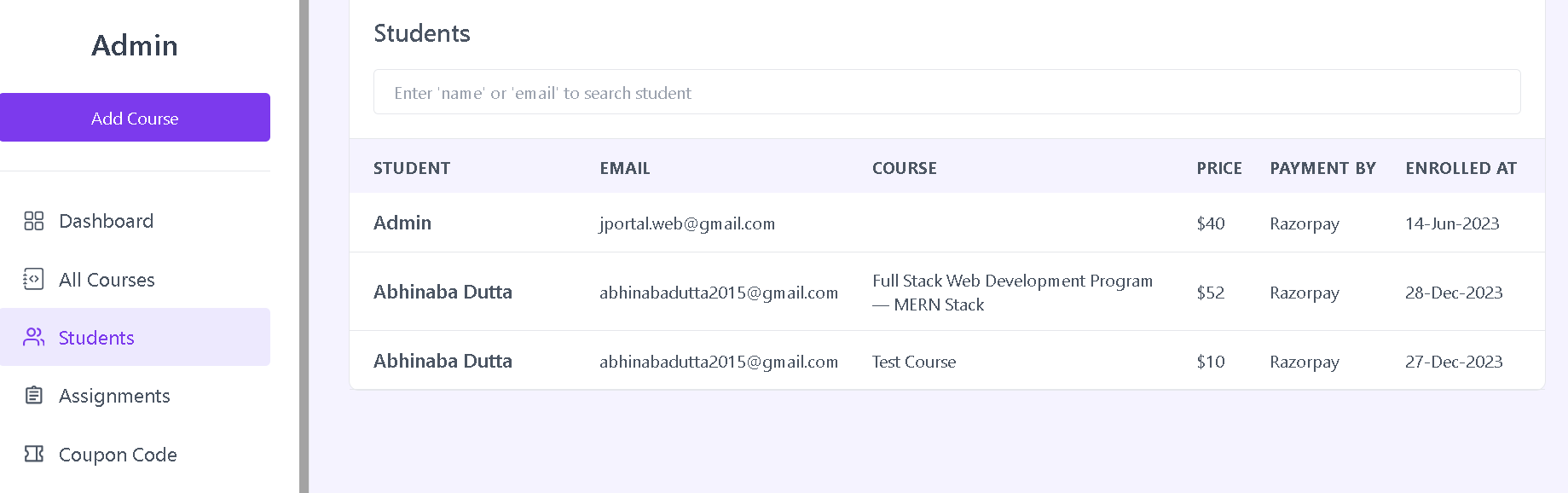
**Students** [**https://citria.org/admin/students?page=1**](https://citria.org/admin/students?page=1)

****

The **Students** component in React manages the display of student information, including a search functionality, paginated table, and an option to view detailed information about a specific student. Here's an explanation of its structure and functionality:

Initialization:

Uses React hooks like useState, useEffect, and useContext for managing state.

Imports external libraries such as moment for date formatting.

Data Fetching:

Fetches student data from the server based on the current page, search query (inputValue), and pagination size (size). The fetched data includes details like student name, email, enrolled course, price, payment method, and enrollment date.

Pagination:

Implements pagination with navigation buttons (MdArrowBackIosNew and MdArrowForwardIos). Updates the URL query parameter when the page changes to maintain the page state.

Search Functionality:

Provides a search input field for filtering students based on name or email. Dynamically updates the displayed students based on the search query.

Table Display:

Presents student information in a table with columns for name, email, enrolled course, price, payment method, and enrollment date.

Utilizes a loading skeleton (TableLoadingSkeleton) while the data is being fetched.

Student Details Sidebar:

Allows users to click on a row in the table to view detailed information about a specific student. The student details are displayed in a sidebar (StudentDetails component).

Error Handling:

Manages errors during data fetching, displaying error messages through the shared context (showToast).

URL Query Parameter Update:

Updates the URL query parameter whenever the page changes to maintain the correct page state even when the user navigates back or forward in the browser.

UI Elements:

Includes UI elements like headers, input fields, and buttons to facilitate user interactions.

Conditional Rendering:

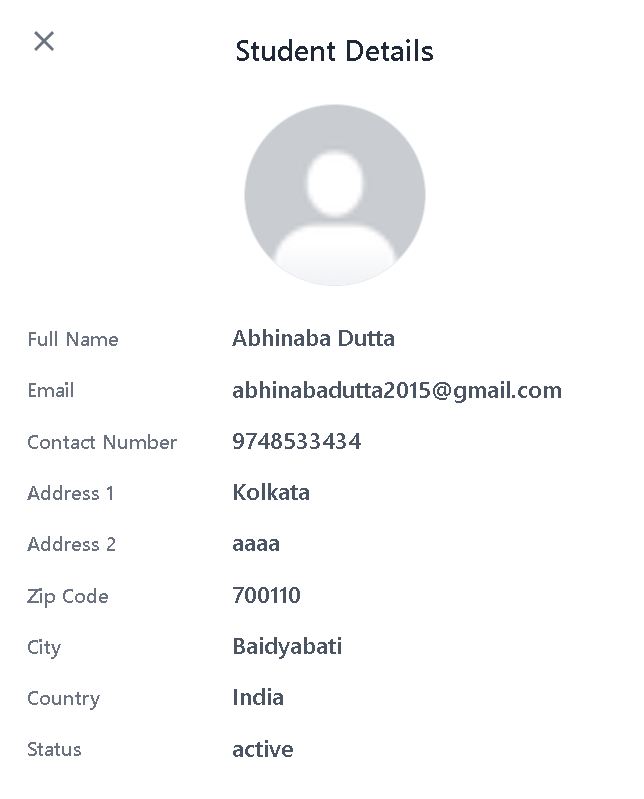
Conditionally renders a message when no student data is found.

Moment.js Usage:

Formats the enrollment date using moment for a more readable display.

In summary, the Students component provides a comprehensive view of student data with pagination, search functionality, and the ability to view detailed information about individual students.

**Student details**

****

The StudentDetails component in React is designed to display detailed information about a specific student in a sidebar. Here's an explanation of its structure and functionality:

Conditional Rendering:

Uses inline styles to control the visibility and opacity of the sidebar based on whether studentDetails is present.

Utilizes the duration-300 class for smooth transitions.

Background Overlay:

Presents a full-screen clickable button with a crosshair cursor that acts as a background overlay.

Clicking on this overlay triggers the closure of the student details sidebar.

Close Button (1st):

Implements a full-width and full-height button with a custom cursor that covers the entire screen. This button serves as an additional means to close the sidebar when clicked.

Sidebar (Aside Element):

Utilizes an aside element for the sidebar, which has a dynamic positioning based on the presence of studentDetails.

The position is transitioned using the duration-300 class for a smooth sliding effect.

Has a fixed width of 30rem and takes up the full height of the screen.

Close Button (2nd):

Positioned in the top-left corner of the sidebar with an absolute position.

Clicking this button triggers the closure of the sidebar.

Student Details Display:

Displays a list of information about the student, including personal details and order-related information.

Shows the student's avatar, full name, email, contact number, address, zip code, city, country, and status.

Displays order-related details such as enrollment date, course, price, payment status, payment method, and transaction ID.

Moment.js Usage:

Uses the moment library to format the enrollment date in a human-readable format.

Dynamic Image Source:

Constructs the source URL for the student's avatar, either from the server or using a default image if the avatar is not available.

Styling:

Applies various styles such as text colors, font sizes, and spacing for a visually appealing presentation.

In summary, the StudentDetails component provides a clean and organized display of detailed information about a student in a sidebar, with interactive buttons for closing the sidebar.