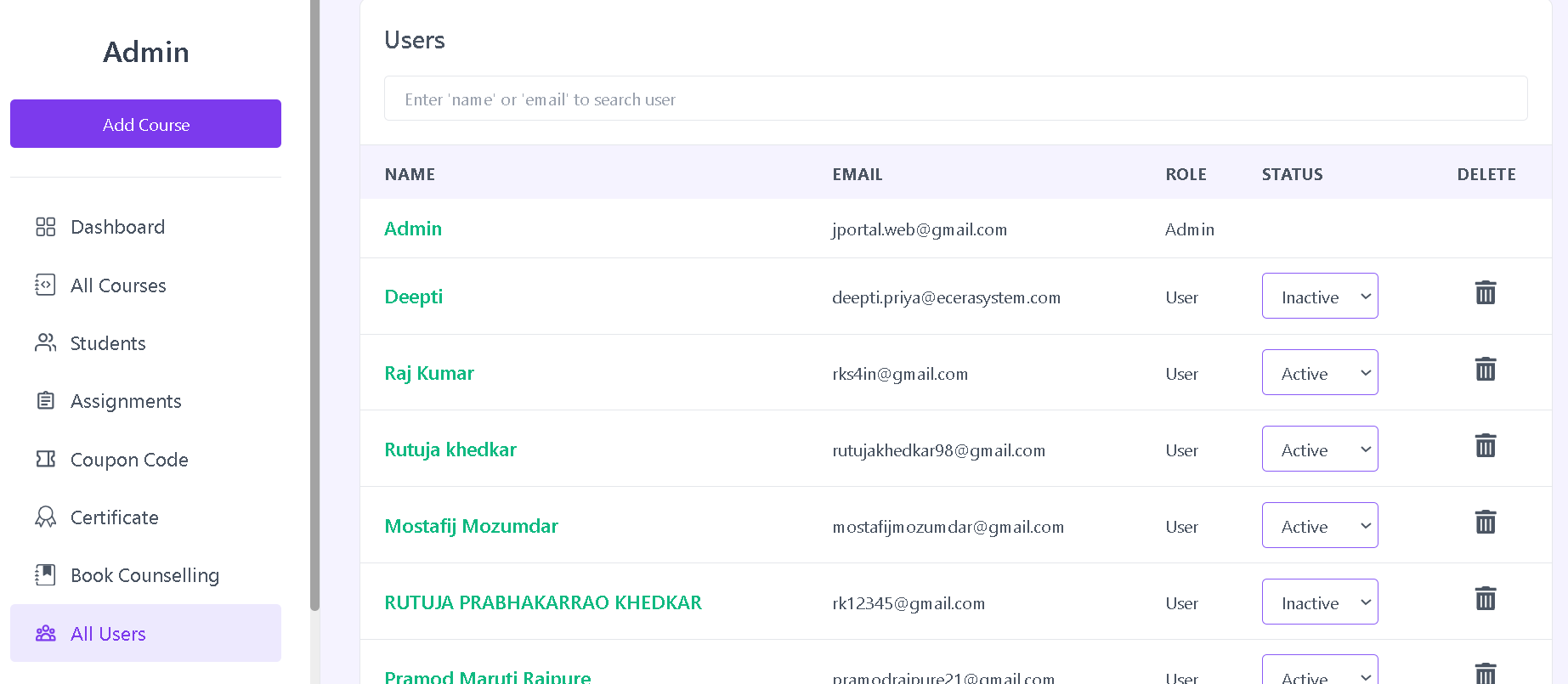
**All Users** [**https://citria.org/admin/all-users?page=1**](https://citria.org/admin/all-users?page=1)

****

This is a React component named AllUsers that provides functionality for managing user data, including pagination, search, status updates, and user deletion. Let's break down its structure and features:

Imports:

React, useState, useEffect, useContext: React-related modules for component creation, state management, and side effects.

useLocation, useNavigate: React Router hooks for accessing the current location and navigation.

axios: A library for making HTTP requests.

Various icons from react-icons for UI elements.

Custom components like PageTitle, TableLoadingSkeleton, Spinner, and UserDetails.

State Variables:

Multiple state variables manage various aspects such as user profiles, loading states, deletion states, user details, search input, and pagination-related variables (page, size, totalPage, etc.).

Effect Hooks:

useEffect hooks are used for updating the URL query parameter when the page changes and fetching data based on pagination and search input.

Search Input:

Allows users to input search criteria (name or email) to filter user data.

User Table:

Displays user data in a table with columns for name, email, role, status, and a delete button.

Provides options for updating user status (active, inactive, blocked) and deleting users.

Pagination Bar:

Displays pagination controls for navigating through different pages of user data.

Delete User Confirmation:

Shows a confirmation modal when attempting to delete a user, providing options to cancel or proceed with deletion.

Loading Spinners:

Displays loading spinners (Spinner component) during status updates or user deletion.

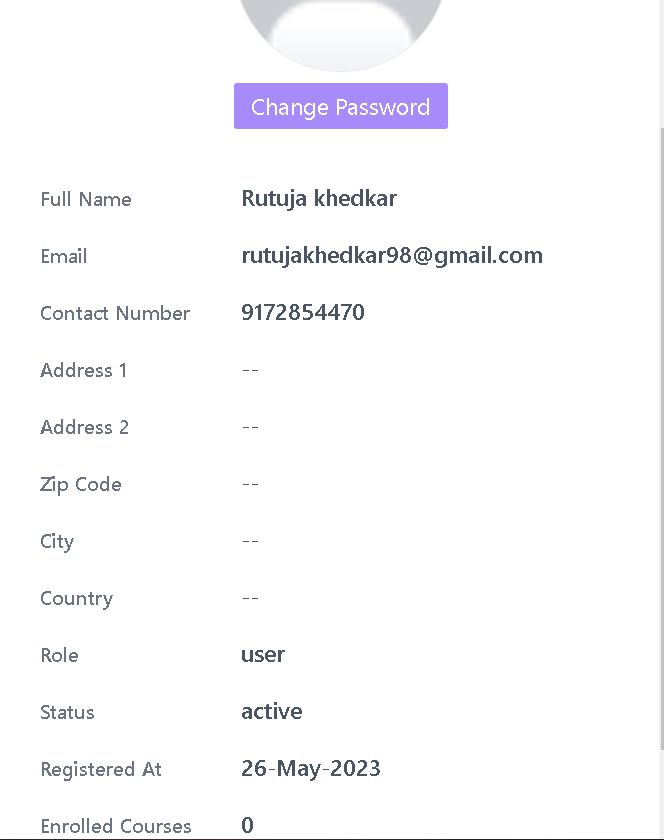
User Details Component:

Utilizes the UserDetails component to display detailed information about a selected user.

Styling:

Utilizes Tailwind CSS classes for styling.

In summary, the AllUsers component serves as a comprehensive user management interface, allowing administrators to search, view, update statuses, and delete users. The component also includes pagination for efficient navigation through a large number of user profiles.



The **UserDetails** component is designed to display detailed information about a specific user. Let's break down its structure and features:

Props:

Receives userDetails and setUserDetails as props, representing the data of the selected user and a function to set the user details to null (closing the details modal).

Overlay and Close Button:

Utilizes a full-screen overlay with a semi-transparent background (bg-black/60) to emphasize the details modal.

The overlay has a clickable button covering the entire screen to close the modal when clicked.

Details Modal:

Positioned absolutely at the top-right of the screen, with a sliding animation (duration-300) for smooth visibility.

Contains a close button (MdClose icon) positioned at the top-left, allowing users to close the modal.

User Information:

Displays user details such as full name, email, contact number, address, zip code, city, country, role, status, registration date, and the number of enrolled courses.

Provides placeholders (--) for optional information that may not be available.

Offers a "Change Password" button that links to the change password page for the respective user.

Styling:

Utilizes Tailwind CSS classes for styling, including dynamic classes for animation and visibility based on the presence of userDetails.

Image Handling:

Displays the user's avatar with a default image if not available.

Handles the avatar URL, including the case where it starts with "/images/" (appending the base API URL in such cases).

Date Formatting:

Uses the moment library to format the user's registration date.

In summary, the UserDetails component enhances the user interface by providing a detailed view of a selected user's information. It follows a clean and responsive design, ensuring a user-friendly experience.