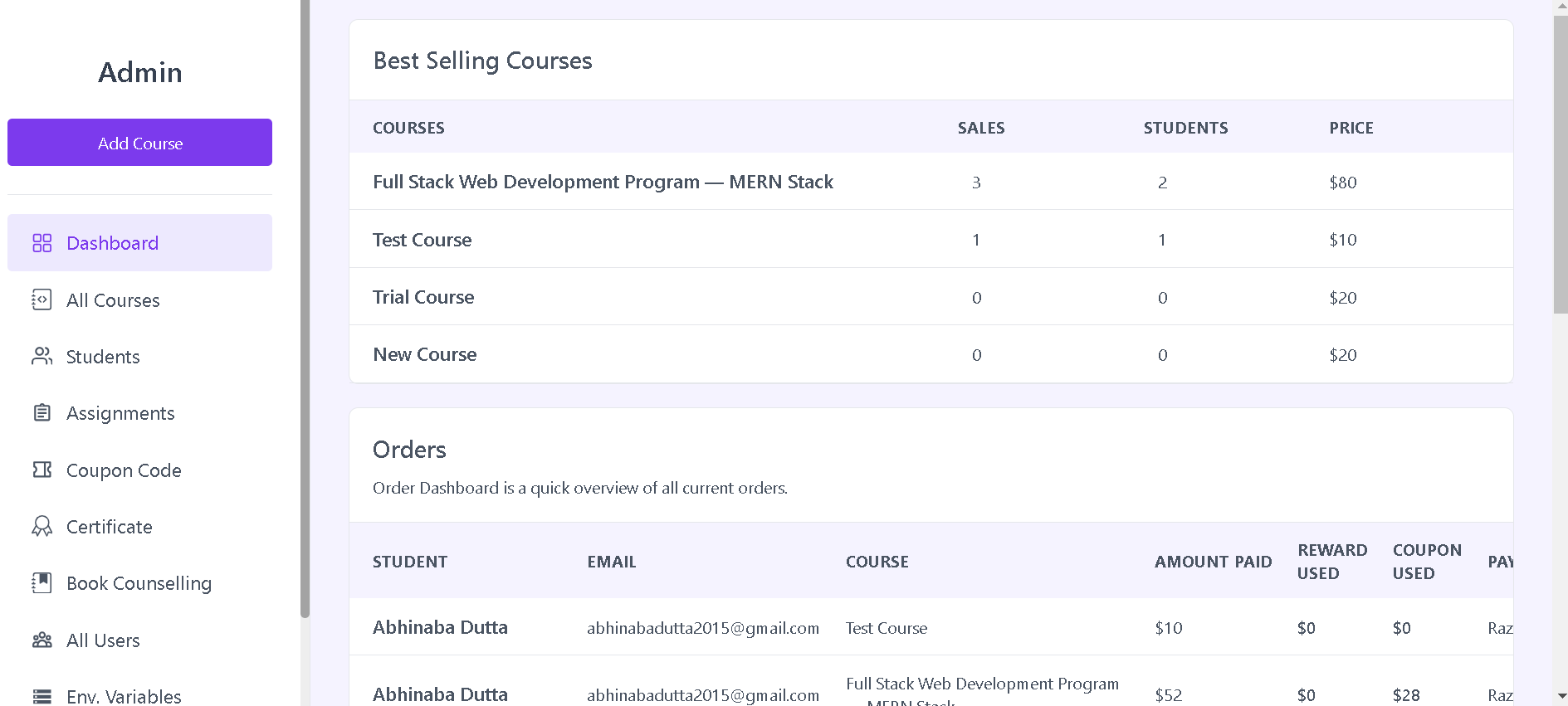
**Admin Dashboard**  <https://citria.org/admin>



Component Structure:

The component uses a fragment (<>...</>) to wrap its content, allowing multiple elements to be returned without introducing an additional parent element in the DOM.

It includes a PageTitle component, TopSaleCourse component, OrderList component, and RefundTerms component.

PageTitle Component:

The PageTitle component is imported from "../../Pages/Shared/PageTitle".

It receives the prop title="Admin Dashboard" and displays a title with the text "Admin Dashboard".

Container Division:

There's a div element with the classes "mt-5 mb-20", which likely provide top and bottom margin spacing.

TopSaleCourse Component:

The TopSaleCourse component is included, and it presumably displays information about top-selling courses.

OrderList Component:

The OrderList component is included, and it presumably displays a list of recent orders.

RefundTerms Component:

The RefundTerms component is included, suggesting that it displays information related to refund terms.

Return Statement:

The Dashboard component returns the JSX structure, including the PageTitle, TopSaleCourse, OrderList, and RefundTerms components.

In summary, the Dashboard component represents the main view for the admin dashboard. It consists of a title, top-selling courses section, recent order list section, and refund terms section. Each section is modularized into its respective component for better organization and maintainability.

**orderList.jsx**

Imports:

axios: A library for making HTTP requests.

moment: A library for handling dates and times.

React, useContext, useEffect, useState, Link, useNavigate: React-related modules for component creation, state management, and navigation.

contextProvider: Presumably, this is a context provider that contains a method named showToast.

TableLoadingSkeleton: A component for rendering a loading skeleton for tables.

StudentDetails: A component for displaying details about a student.

State Variables:

orders: State variable to store the list of orders.

loading: State variable to manage the loading state.

studentDetails: State variable to store details about a selected student.

Fetching Refund Terms:

The component uses the useGetRefundTerms hook to fetch refund terms data.

Fetching Recent Orders:

The component uses the useEffect hook to make an HTTP GET request to fetch recent orders from the API.

The fetched orders are stored in the orders state variable, and the loading state is updated accordingly.

If an error occurs during the request, it displays an error toast and redirects to the sign-in page if the user is not authenticated.

Rendering Table:

The component renders a table to display order details, including student name, email, course, amount paid, reward used, coupon used, payment method, enrolled date, and refund status.

If loading is true, a loading skeleton is displayed; otherwise, the orders are mapped to table rows.

The refund status is determined based on whether a refund is requested, the time since enrollment, and the refund terms.

Conditional Rendering:

If no orders are found, it displays a message indicating no orders data.

If there are orders, it provides a link to view all orders.

StudentDetails Component:

The StudentDetails component is rendered to display additional details about a selected student.

In summary, the OrderList component fetches and displays recent orders, including various details. It also uses the StudentDetails component to show additional information about a selected student.

**TopsaleCourse.jsx**

Imports:

axios: A library for making HTTP requests.

React, useEffect, useState, Link, useNavigate: React-related modules for component creation, state management, and navigation.

contextProvider: Presumably, this is a context provider that contains a method named showToast.

TableLoadingSkeleton: A component for rendering a loading skeleton for tables.

State Variables:

topSales: State variable to store the list of top-selling courses.

loading: State variable to manage the loading state.

Fetching Top-Selling Courses:

The component uses the useEffect hook to make an HTTP GET request to fetch top-selling courses from the API.

The fetched data is stored in the topSales state variable, and the loading state is updated accordingly.

If an error occurs during the request, it displays an error toast and redirects to the sign-in page if the user is not authenticated.

Rendering Table:

The component renders a table to display information about the best-selling courses, including course title, sales, number of students, and price.

If loading is true, a loading skeleton is displayed; otherwise, the top-selling courses are mapped to table rows.

Conditional Rendering:

If no top-selling courses are found, it displays a message indicating that no courses have been sold yet.

Table Structure:

The table has columns for "Courses," "Sales," "Students," and "Price."

Mapping Data to Table Rows:

If not loading, it maps the top-selling courses to table rows, displaying details such as course title, sales count, number of students, and price.

In summary, the TopSaleCourse component fetches and displays information about the best-selling courses. It uses the TableLoadingSkeleton component for a loading skeleton effect and includes conditional rendering for cases where there are no top-selling courses.

**RefundTerms.jsx**

Imports:

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

useGetRefundTerms: A custom hook for fetching refund terms data.

axios: A library for making HTTP requests.

contextProvider: Presumably, this is a context provider that contains a method named showToast.

State Variables:

refundTermsData: State variable to store the refund terms data.

setRefundTermsData: State updater function for refundTermsData.

fetchRefundTermsData: Function from the custom hook to fetch refund terms data.

newReturnWindow: State variable to store the new value for the return window.

setNewReturnWindow: State updater function for newReturnWindow.

newRegistrationFees: State variable to store the new value for registration fees.

setNewRegistrationFees: State updater function for newRegistrationFees.

HTTP Requests to Update Refund Terms:

handleUpdateReturnWindow: Function to handle updating the return window. It sends a PATCH request to the server with the new return window value.

handleUpdateRegistrationFees: Function to handle updating registration fees. It sends a PUT request to the server with the new registration fees value.

Toast Notifications:

The component uses the showToast method from the context provider to display success or error messages.

Rendering Refund Terms Information:

Displays the current refund terms information, including the return window and registration fees.

Input Fields for Updating Refund Terms:

Provides input fields for the user to enter new values for the return window and registration fees.

Update Buttons:

Buttons triggering the update functions (handleUpdateReturnWindow and handleUpdateRegistrationFees).

Styling:

Uses Tailwind CSS classes for styling.

In summary, the RefundTerms component allows users to view current refund terms, input new values for the return window and registration fees, and update them through HTTP requests. The component provides a user interface for managing refund-related settings.