**Header**



The **Header component** is a React functional component designed for creating a navigation header with both desktop and mobile views. It includes features such as a logo, responsive menu, and dynamic styling based on scroll position.

Components Used

Link: Imported from 'react-router-dom' for creating navigation links.

HiBars3CenterLeft and VscClose: Imported from 'react-icons/hi2' and 'react-icons/vsc' respectively, for rendering menu icons.

DesktopView and MobileView: Components that handle menu items for desktop and mobile views.

State

scroll: A state variable to track whether the page is scrolled or not.

openMenu: A state variable to control the visibility of the mobile menu.

Event Listener

The component utilizes a scroll event listener to dynamically update the scroll state based on the vertical scroll position. This triggers a change in the header's appearance.

Styling

The component applies dynamic styling based on the scroll state using Tailwind CSS classes.

It features a fixed header with a shadow when scrolled, and a transparent header when at the top.

Structure

Logo Section:

Displays the website logo as a link to the home page.

Menu Sections:

Desktop View: Renders menu items for larger screens using the DesktopView component.

Mobile View: Displays a toggle button and the MobileView component for smaller screens when the menu icon is clicked.

Usage

import Header from 'path-to-Header';

// Include Header in your React component

function App() {

return (

<div>

<Header profile={profileData} textColor='text-gray-600' />

{/\* Other components and content \*/}

</div>

);

}

Conclusion

The Header component enhances the user experience by providing a responsive and visually appealing navigation bar. It adapts to different screen sizes and incorporates dynamic styling based on the scroll position for a modern and user-friendly design.



The provided code is a React functional component named MobileView. This component is designed to render a mobile navigation menu with various options, including Programs, Projects, Placements, and user authentication features.

Component Structure:

Import Statements:

The component starts with import statements for React and various dependencies like useContext, useState, and several React Router components.

Context Usage:

It utilizes the useContext hook to access the state variable isLoggedIn from the contextProvider. This indicates whether a user is currently logged in.

State Management:

The component uses the useState hook to manage the state of the sign-in and sign-up pop-ups (signInPopUp and signUpPopUp), as well as the state for the visibility of the Programs dropdown (openPrograms).

Rendering UI:

The main structure is an unordered list (<ul>) with dynamic CSS classes based on the openMenu prop to control its visibility.

Depending on the user's login status, it renders either a HeaderProfile component or sign-in/sign-up buttons.

Programs Dropdown:

There's a section for Programs, which, when clicked, toggles the visibility of a nested unordered list (<ul>) containing navigation links.

Navigation Links:

Navigation links for Projects and Placements are rendered using React Router's NavLink component.

Authentication Buttons:

If the user is not logged in, it renders sign-in and sign-up buttons with corresponding click handlers.

Sign-in and Sign-up Modals:

Conditionally renders SignIn and SignUp components when the respective state variables are true, creating modal pop-ups for user authentication.

Styling:

The component uses Tailwind CSS classes for styling, including responsiveness and hover effects.

Usage in Documentation:

To explain this component in documentation, you can cover its purpose, the props it accepts (like openMenu), and how it enhances the user experience on mobile devices.