**ResetPassword**

The ResetPassword component is a React component responsible for handling the process of resetting a user's password using a provided token. Let's explore its key features and functionality.

1. State and Hooks:

The component uses React hooks like useState and useContext to manage state.

State variables include passwords (to store the new password and its confirmation), loading (to track loading state), and showToast (to display notifications).

2. URL Parameters:

It utilizes the useParams hook from React Router to extract the token parameter from the URL. This token is used for securely identifying and verifying the user.

3. Form Handling:

The handleResetPassword function is triggered on form submission.

It validates the password and confirmation fields, ensuring they are not empty and that the password is at least 6 characters long.

It makes an asynchronous POST request to the server (axios.post) to reset the password using the provided token.

It handles success and error responses, updating the UI accordingly.

4. UI/UX and Styling:

The component is styled using Tailwind CSS, providing a clean and responsive design.

It includes a page title (PageTitle component) for better organization.

5. Routing and Navigation:

The useNavigate hook is used for programmatic navigation after successful actions.

Users can cancel the password reset process and return to the sign-in page by clicking the "Cancel" link (Link to={'/sign-in'}).

6. External Components:

The component includes the Spinner component, which is displayed during loading operations, providing feedback to the user.

7. API Interaction:

The component communicates with the server via Axios to trigger the password reset process.

Success and error responses from the server are handled, and appropriate notifications are displayed to the user.

8. Form Structure:

The form includes input fields for the new password and its confirmation, allowing the user to securely reset their password.

It includes buttons to submit the form or cancel the process.

9. Responsive Design:

The component is designed to be responsive, ensuring a consistent user experience across different devices.