**Paginated User List Documentation**

**Overview**

This React component displays a paginated list of users fetched from an external API. Users can navigate through different pages using pagination controls.

**Features**

* Fetches user data from https://dummyjson.com/users
* Displays 3 users per page
* Provides pagination with page numbers and "Previous" & "Next" buttons
* Highlights the current page button
* Disables "Previous" on the first page and "Next" on the last page
* Styled user cards with responsive design

**Installation & Setup**

1. Ensure you have Node.js and React installed.
2. Add the UserList component to your project.
3. Import the component where needed:
4. import UserList from "./UserList";
5. Ensure you have the CSS file PaginatedUserList.css in the styles folder.

**Code Breakdown**

**State Management**

* users: Stores the list of users fetched from the API.
* currentPage: Tracks the current page number.
* usersPerPage: Fixed to 3 users per page.

**API Fetching**

* The useEffect hook fetches user data when the component mounts.
* The fetched data is stored in the users state.

**Pagination Logic**

* Calculates the total pages using Math.ceil(users.length / usersPerPage).
* Determines which users to display using array slicing.
* Updates the page when a pagination button is clicked.

**UI Components**

* **User List**: Displays user details (name, email, phone).
* **Pagination Controls**:
  + "Previous" and "Next" buttons navigate between pages.
  + Numbered buttons allow direct page selection.
  + The current page button is highlighted.

**Styling**

Located in PaginatedUserList.css:

* .container centers the component.
* .card-container styles user cards.
* .pagination-controls styles pagination buttons.
* .btn:hover adds a hover effect to buttons.

**Example Usage**

import React from "react";

import UserList from "./src/UserListList";

function App() {

return (

<div>

<PaginatedUserList />

</div>

);

}

export default App;

**Future Improvements**

* Add search and filtering options.
* Implement a loading state while fetching data.
* Improve accessibility with better ARIA attributes.

This documentation provides a clear understanding of the PaginatedUserList component and how to use it effectively in a React project.