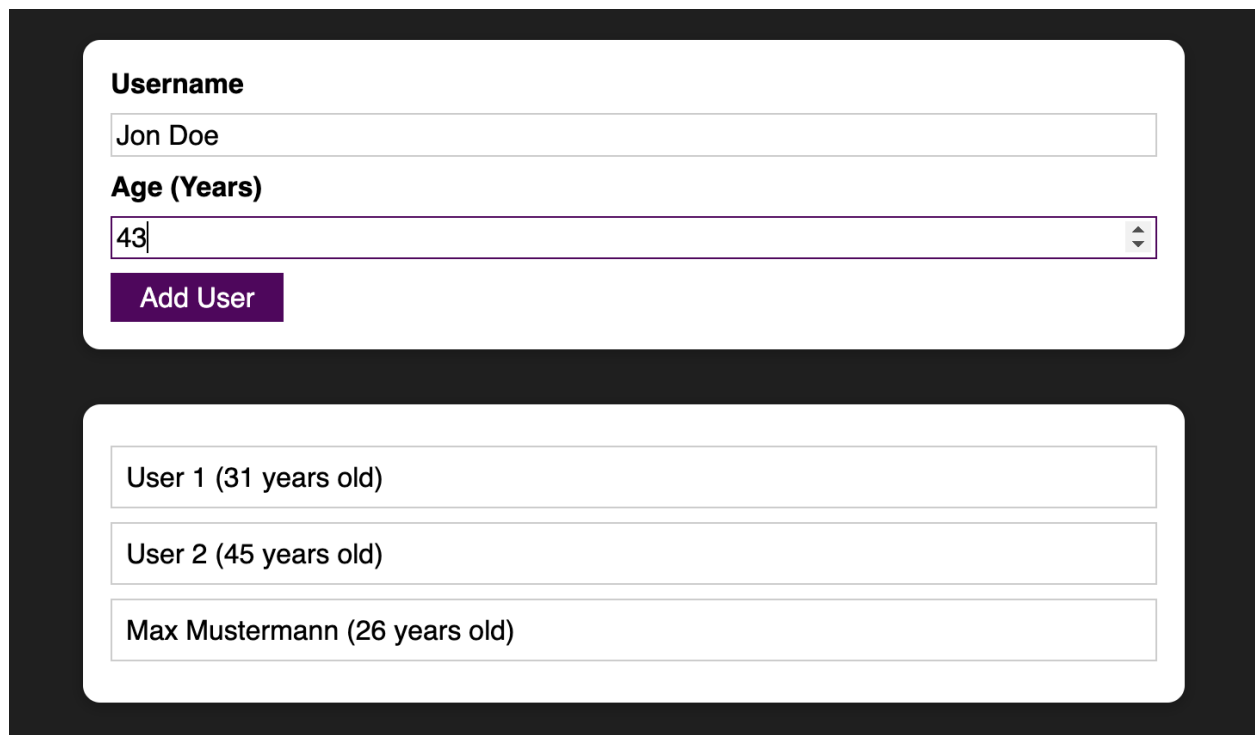


# Übungsprojekt: User Manager

Um die gelernten Konzepte der letzten Einheiten zu vertiefen und praktisch anzuwenden, werden wir gemeinsam Schritt für Schritt eine User Manager Applikation entwickeln.

Folgende Features werden in diesem Projekt umgesetzt:

- Eingabeformular für Userdaten
- Validierung des Eingabeformulars
- Darstellung der hinzugefügten User in einer Liste
- Anzeigen eines Error-Modals bei invaliden Eingaben



The image shows a user interface for a 'User Manager' application. It consists of two main white panels on a dark background. The top panel is for adding a new user, featuring a 'Username' input field with 'Jon Doe', an 'Age (Years)' input field with '43', and a purple 'Add User' button. The bottom panel displays a list of existing users: 'User 1 (31 years old)', 'User 2 (45 years old)', and 'Max Mustermann (26 years old)'.

**Username**

Jon Doe

**Age (Years)**

43

Add User

User 1 (31 years old)

User 2 (45 years old)

Max Mustermann (26 years old)

## Lernziele

- Component-based Development verstehen
- Unterschied React zu purem JavaScript verstehen
  - Declarative vs. imperative Entwicklung
- React Komponenten verwenden
- Styling der Komponenten mit CSS Modules
- Komponentenkommunikation mit props und Event Callbacks
- Erstellung Wrapper Komponenten mit props.children
- Handling State mit useState()
- Conditional Styling
- Ausgeben von Listen (Arrays)

## Aufgaben

### 1. Starter Projekt herunterladen und starten

- 01-starting-project Ordner öffnen
- Projektordner mit integriertem Terminal öffnen
- **npm install** und **npm start** ausführen

### 2. Erstellen von Eingabeformular



The image shows a user registration form with a dark background. The form is a white rounded rectangle containing two input fields and a button. The first field is labeled 'Username' and the second is labeled 'Age (Years)'. Below the second field is a purple button with the text 'Add User' in white.

**Username**

**Age (Years)**

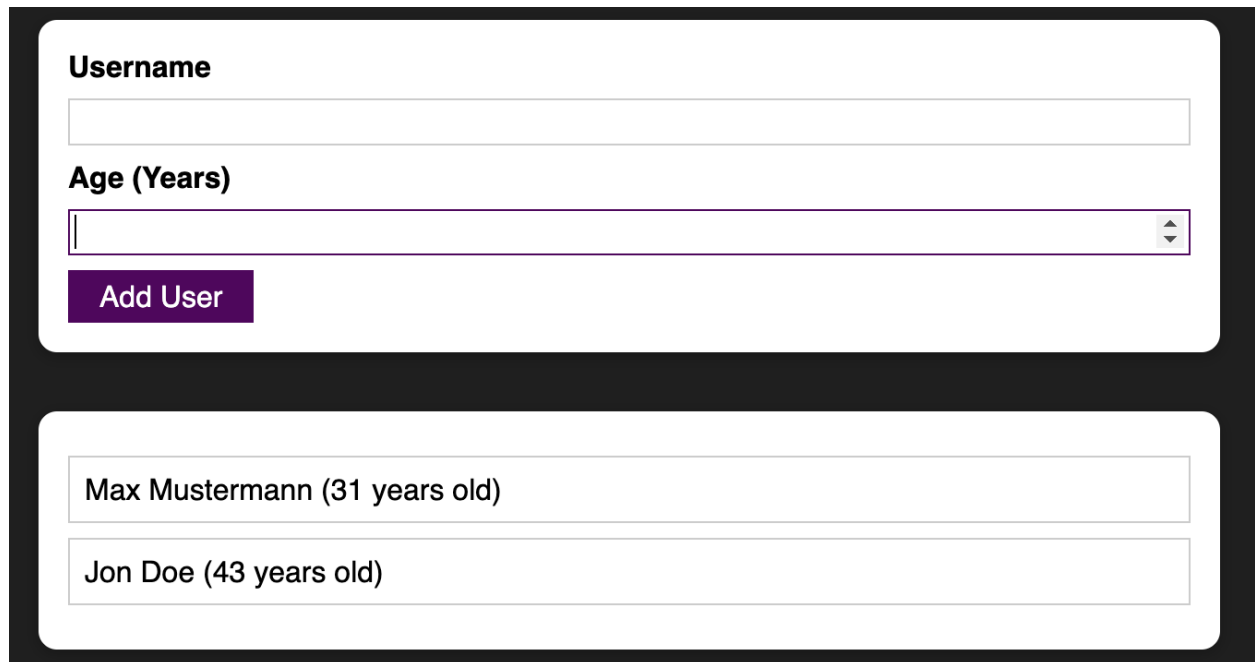
**Add User**

- Implementierung eines Eingabeformulars für
  - Usernamen
  - Alterseingabe
  - Submit Button
- UI Elemente und das Formular sollen, wenn es für notwendig erachtet wird, in eine eigene React Komponente ausgelagert werden. Vorschlag:
  - AddUser.js
  - Button.js
  - Card.js
- Bei Klick auf “Add User” soll ein alert() ausgegeben werden, dass der Button geklickt wurde. Die Eingabedaten werden erst im nächsten Schritt weiterverarbeitet.
- CSS Dateien werden im Slack bereitgestellt und können in das Projekt eingefügt werden!
  - Aufgabe: Import des CSS Modules und Hinzufügen der Klassen zu den HTML Elementen

### **3. Verwalten und Validieren von Usereingaben**

- Username und Age sollen bei jeder Eingabe mit useState() in der AddUser Komponente gespeichert werden
- Bei Formular Submit sollen die gespeicherten Werte validiert werden
  - Keine leeren Eingabewerte
  - Age muss > 1 sein
- Nach erfolgreicher Validierung sollen die Formularwerte zurückgesetzt werden

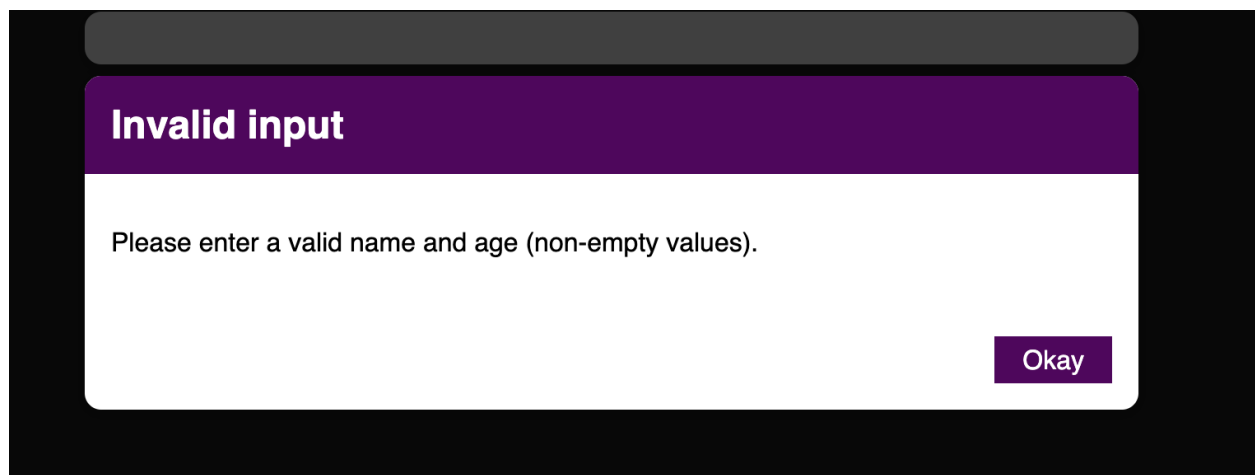
### **4. Verwalten der User und Ausgeben in einer Liste**



The image shows a user management interface. The top section is a form with two input fields: 'Username' and 'Age (Years)'. The 'Age (Years)' field is a dropdown menu. Below the inputs is a purple 'Add User' button. The bottom section is a list of users, showing 'Max Mustermann (31 years old)' and 'Jon Doe (43 years old)' in separate list items.

- User State verwalten in App.js mit `useState()`
- Erstellen einer `UserList.js` Komponente
  - User Array von App.js ausgeben
  - Jeder User kann als ein Listenelement ausgegeben werden
  - CSS Module von Starterpaket importieren und Klassen verknüpfen

### Zusatzaufgabe: Darstellen von Error-Modal bei invalider Eingabe



The image shows an error modal dialog. It has a purple header with the text 'Invalid input'. Below the header, the message 'Please enter a valid name and age (non-empty values).' is displayed. In the bottom right corner, there is a purple 'Okay' button.

- Erstellen einer generischen ErrorModal.js Komponente (Styling in Starterpaket)
- **Title**, **Message** und **onConfirm Callback** können von Parent Komponente übergeben werden
- Zurvor erstelle Komponenten Card und Button sollen wiederverwendet werden