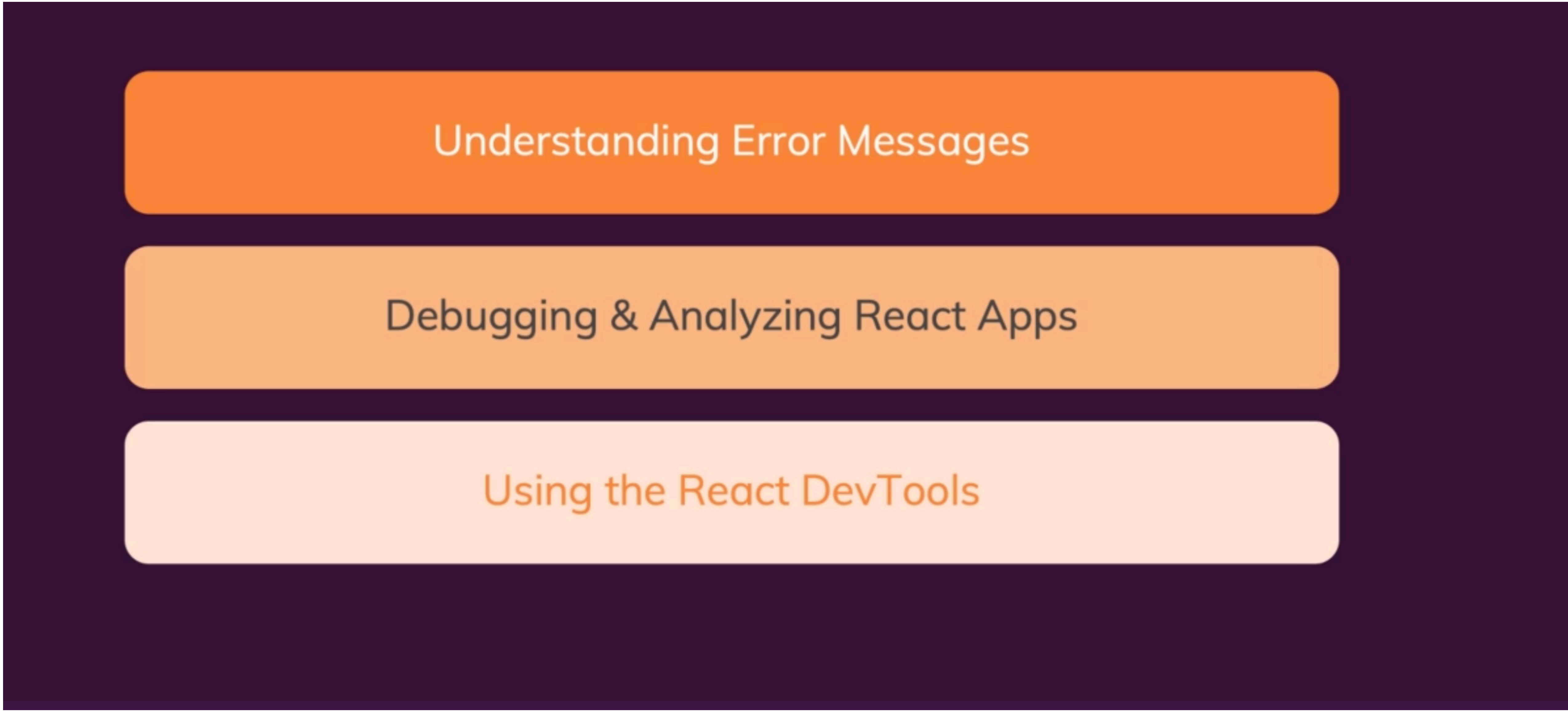
Front End

Class 8

September 1, 2021

Agenda

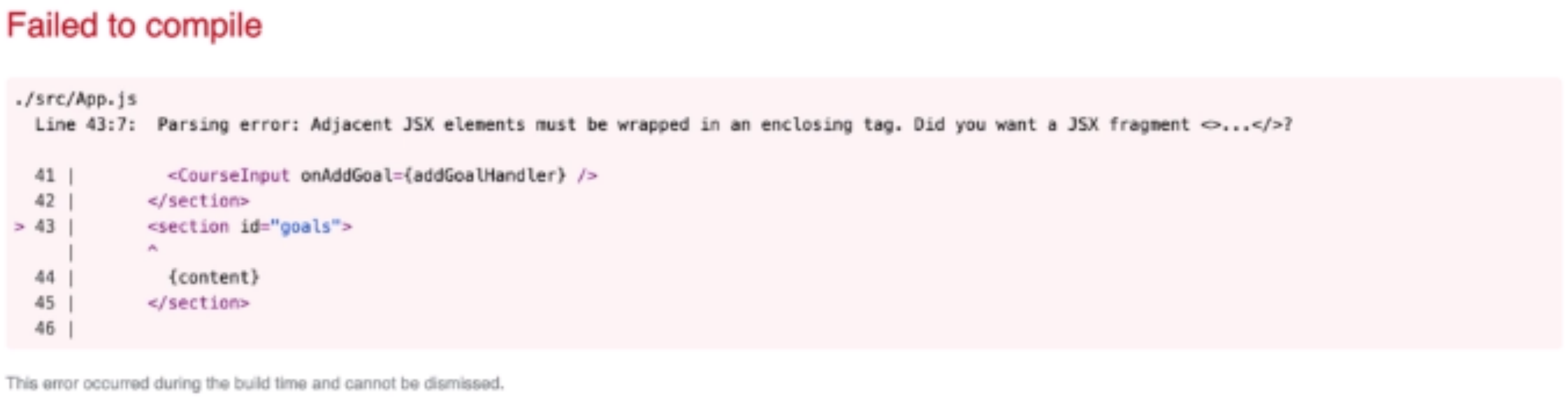
- Kahoot
- Code walkthrough with slides
- Demo’s



Understanding Error Messages

Debugging & Analyzing React Apps

Using the React DevTools



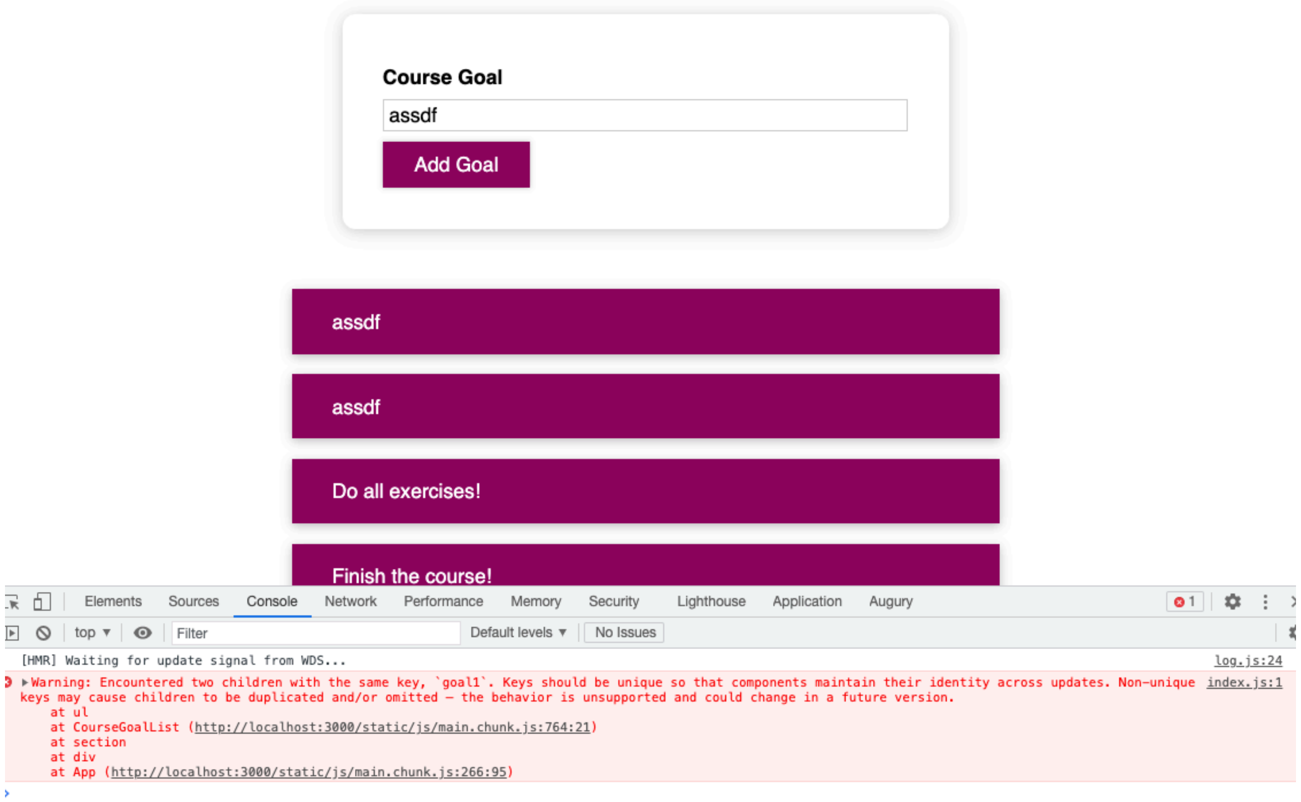
Failed to compile

```
./src/App.js
  Line 43:7:  Parsing error: Adjacent JSX elements must be wrapped in an enclosing tag. Did you want a JSX fragment <>...</>?

  41 |         <CourseInput onAddGoal={addGoalHandler} />
  42 |       </section>
> 43 |       <section id="goals">
     |       ^
  44 |         {content}
  45 |       </section>
  46 |
```

This error occurred during the build time and cannot be dismissed.

Analyzing Code Flow and Warnings.

- Logical error requires duplicating an issue and performing code analysis.



Course Goal

assdf

Add Goal

assdf

assdf

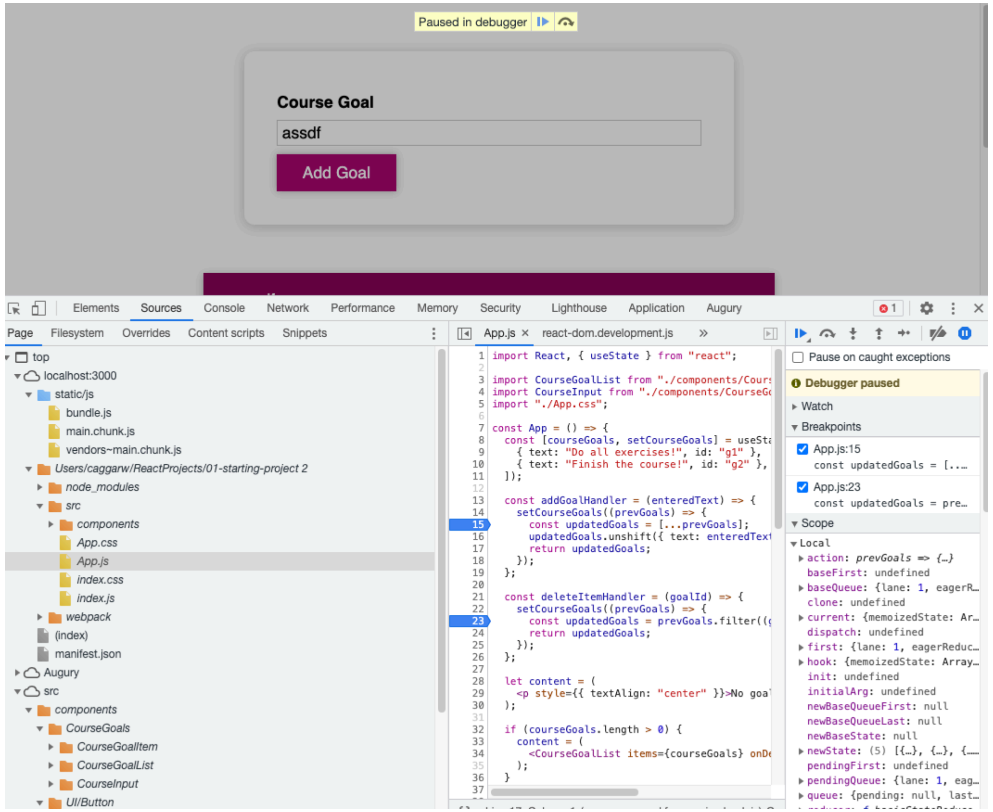
Do all exercises!

Finish the course!

```
[HMR] Waiting for update signal from WDS...                                                                                  log.js:24
▸Warning: Encountered two children with the same key, `goal1`. Keys should be unique so that components maintain their identity across updates. Non-unique  index.js:1
keys may cause children to be duplicated and/or omitted — the behavior is unsupported and could change in a future version.
    at ul
    at CourseGoalList (http://localhost:3000/static/js/main.chunk.js:764:21)
    at section
    at div
    at App (http://localhost:3000/static/js/main.chunk.js:266:95)
›
```

Working with Breakpoints.

- Can open source and add breakpoints.



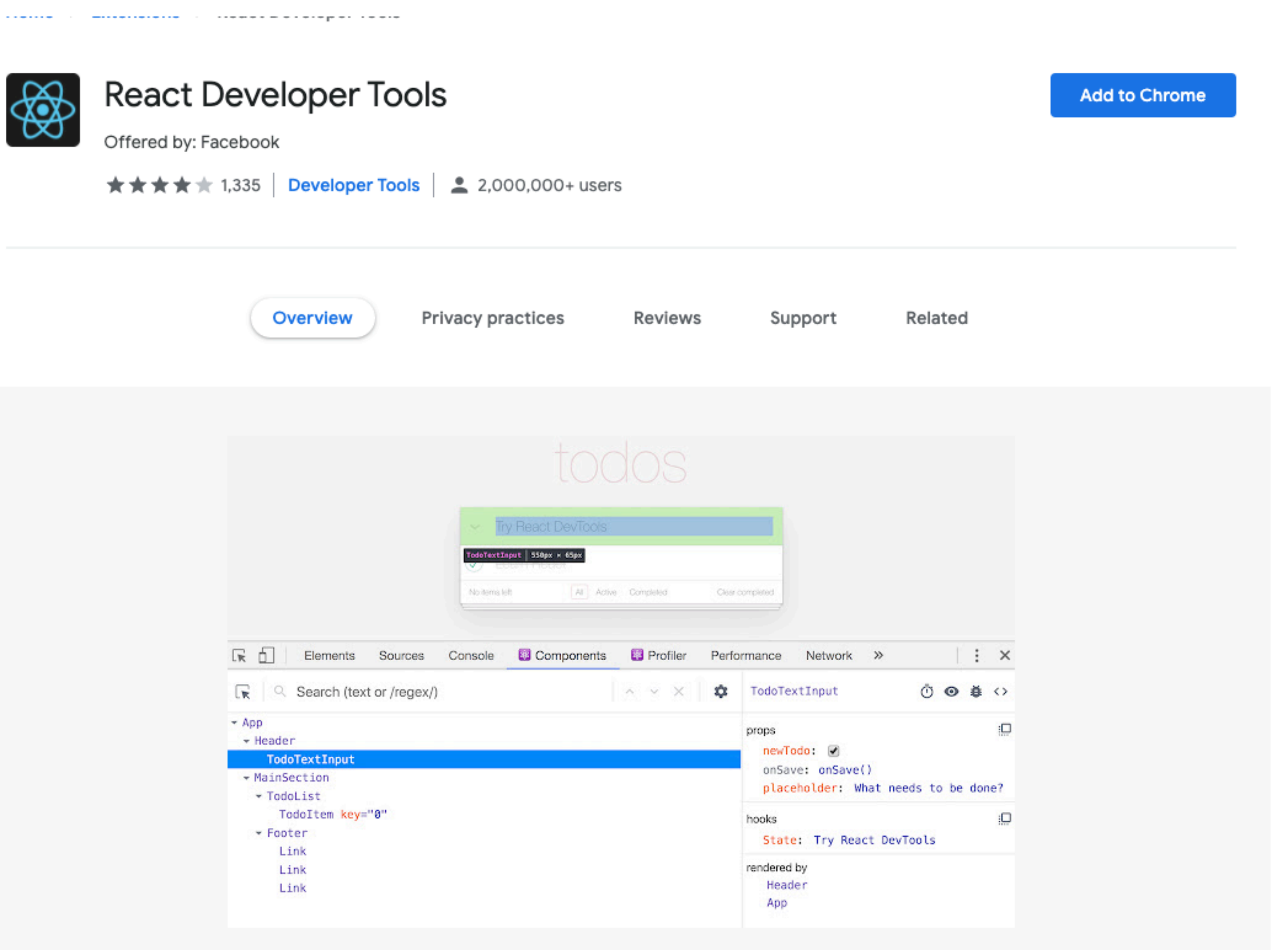
```
import React, { useState } from "react";

import CourseGoalList from "./components/Cours
import CourseInput from "./components/CourseGo
import "./App.css";

const App = () => {
  const [courseGoals, setCourseGoals] = useSta
    { text: "Do all exercises!", id: "g1" },
    { text: "Finish the course!", id: "g2" },
  ]);

  const addGoalHandler = (enteredText) => {
    setCourseGoals((prevGoals) => {
      const updatedGoals = [...prevGoals];
      updatedGoals.unshift({ text: enteredText
      return updatedGoals;
    });
  };

  const deleteItemHandler = (goalId) => {
    setCourseGoals((prevGoals) => {
      const updatedGoals = prevGoals.filter((g
      return updatedGoals;
    });
  };

  let content = (
    <p style={{ textAlign: "center" }}>No goa
  );

  if (courseGoals.length > 0) {
    content = (
      <CourseGoalList items={courseGoals} onDe
    );
  }
```

☑ App.js:15
`const updatedGoals = [...…`

☑ App.js:23
`const updatedGoals = pre…`

```
▾ Local
  ▸ action: prevGoals => {…}
    baseFirst: undefined
  ▸ baseQueue: {lane: 1, eagerR…
    clone: undefined
  ▸ current: {memoizedState: Ar…
    dispatch: undefined
  ▸ first: {lane: 1, eagerReduc…
  ▸ hook: {memoizedState: Array…
    init: undefined
    initialArg: undefined
    newBaseQueueFirst: null
    newBaseQueueLast: null
    newBaseState: null
  ▸ newState: (5) [{…}, {…}, {……
    pendingFirst: undefined
  ▸ pendingQueue: {lane: 1, eag…
  ▸ queue: {pending: null, last…
```

Using the React Dev tools.



React Developer Tools

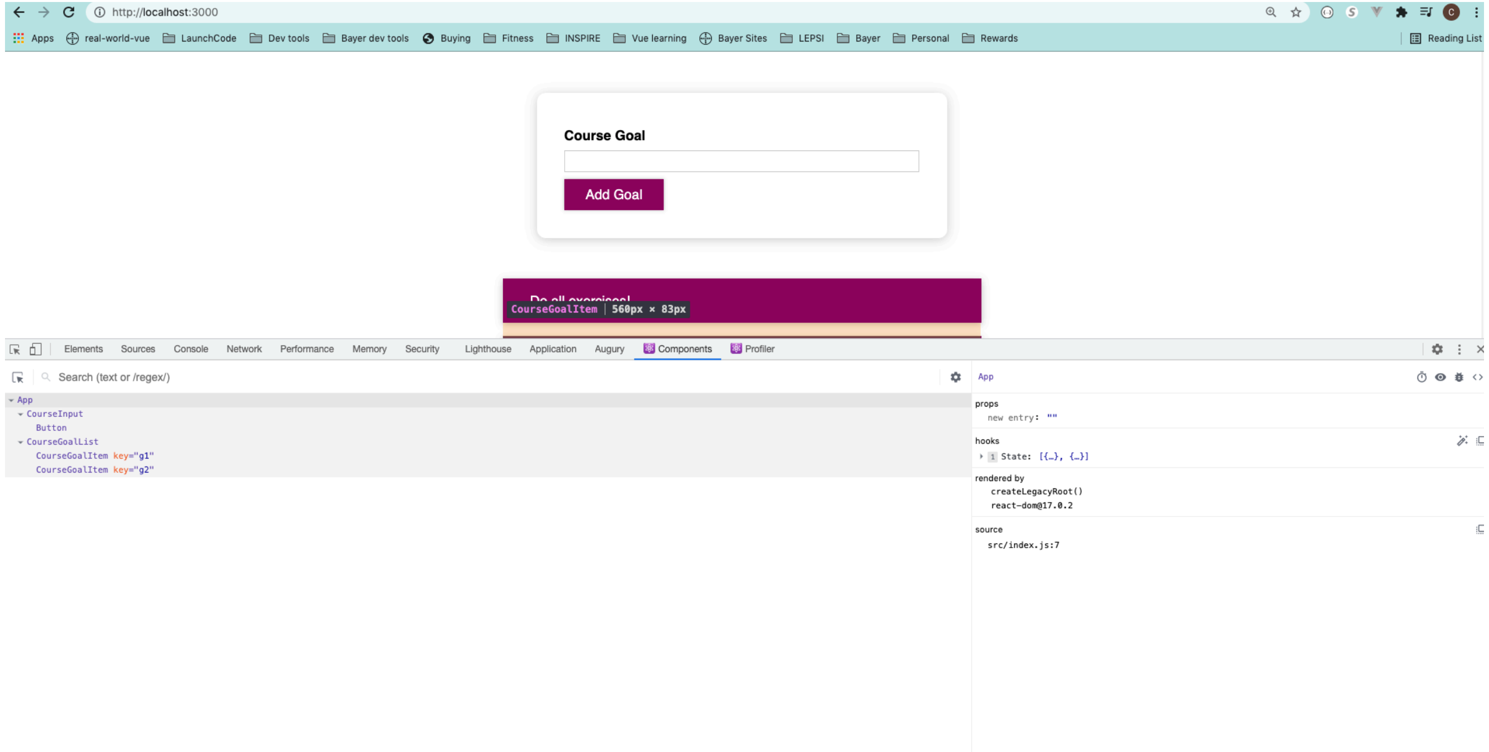
Offered by: Facebook

★★★★☆ 1,335 | Developer Tools | 2,000,000+ users

```
▾ App
  ▾ Header
      TodoTextInput
  ▾ MainSection
    ▾ TodoList
        TodoItem key="0"
    ▾ Footer
        Link
        Link
        Link
```

```
TodoTextInput
props
  newTodo: ☑
  onSave: onSave()
  placeholder: What needs to be done?
hooks
  State: Try React DevTools
rendered by
  Header
  App
```

Using the React Dev tools.



```
▾ App
  ▾ CourseInput
      Button
  ▾ CourseGoalList
      CourseGoalItem key="g1"
      CourseGoalItem key="g2"
```

```
App

props
  new entry: ""

hooks
  ▸ 1 State: [{…}, {…}]

rendered by
   createLegacyRoot()
   react-dom@17.0.2

source
  src/index.js:7
```

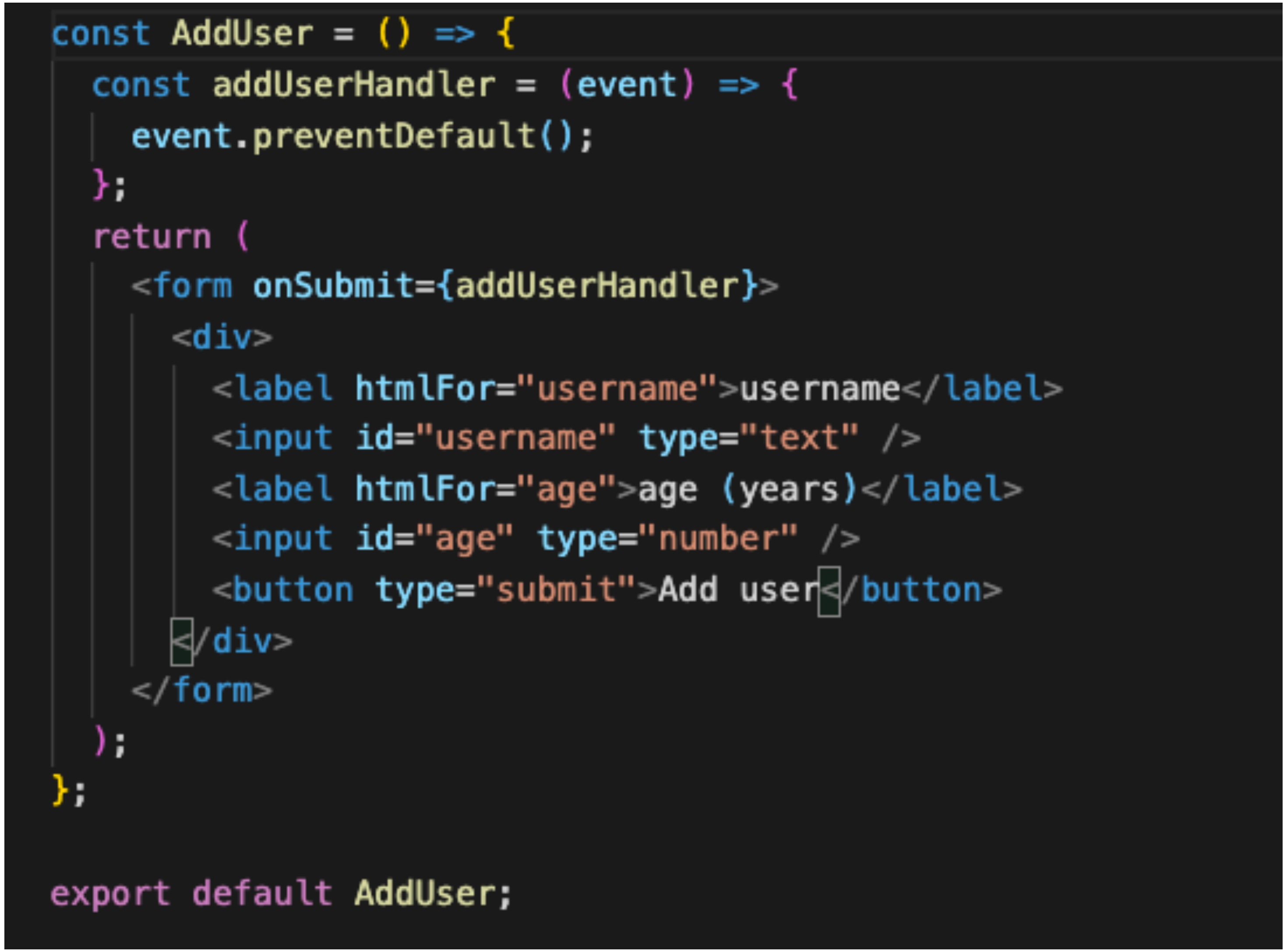
Practice Project.

- Simple app to practice principles.
- Add users with username and age.
- Validation message when creating empty user.
- Really understand core concepts.
- Building React components and adding general features.

Empty Project.

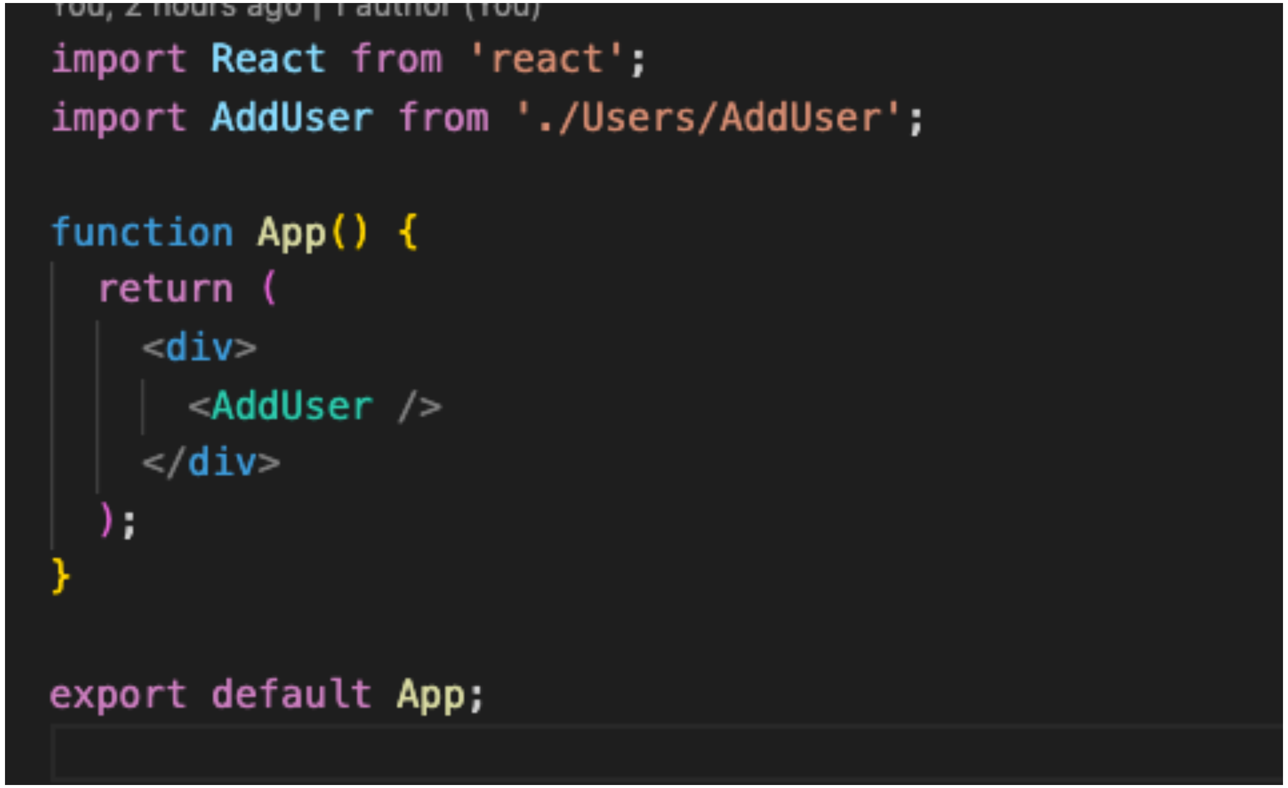
- Download starting application and run npm i

User Component



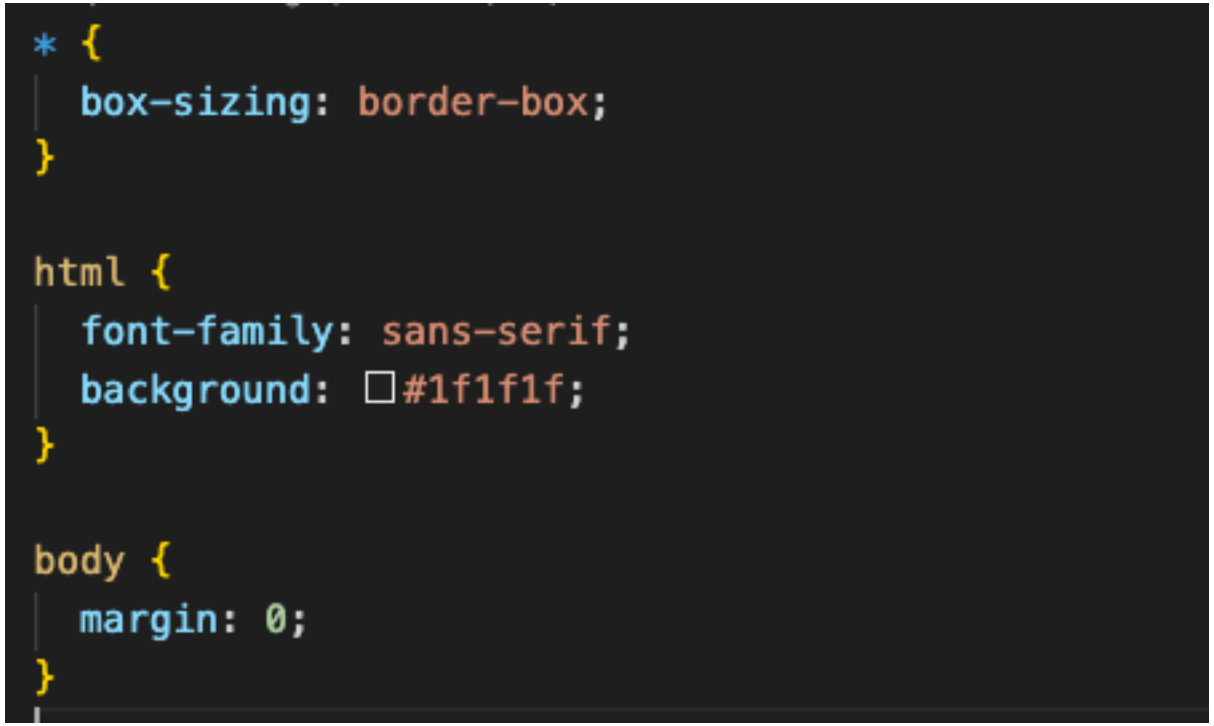
```
const AddUser = () => {
  const addUserHandler = (event) => {
    event.preventDefault();
  };
  return (
    <form onSubmit={addUserHandler}>
      <div>
        <label htmlFor="username">username</label>
        <input id="username" type="text" />
        <label htmlFor="age">age (years)</label>
        <input id="age" type="number" />
        <button type="submit">Add user</button>
      </div>
    </form>
  );
};

export default AddUser;
```

User Component



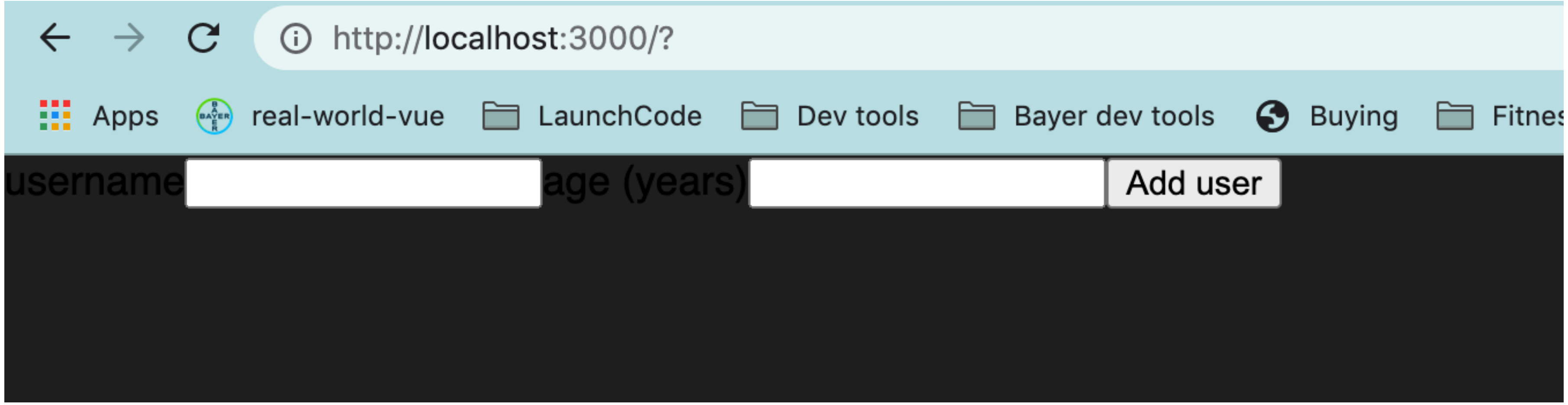
```
You, 2 hours ago | 1 author (You)
import React from 'react';
import AddUser from './Users/AddUser';

function App() {
  return (
    <div>
      <AddUser />
    </div>
  );
}

export default App;
```

User Component

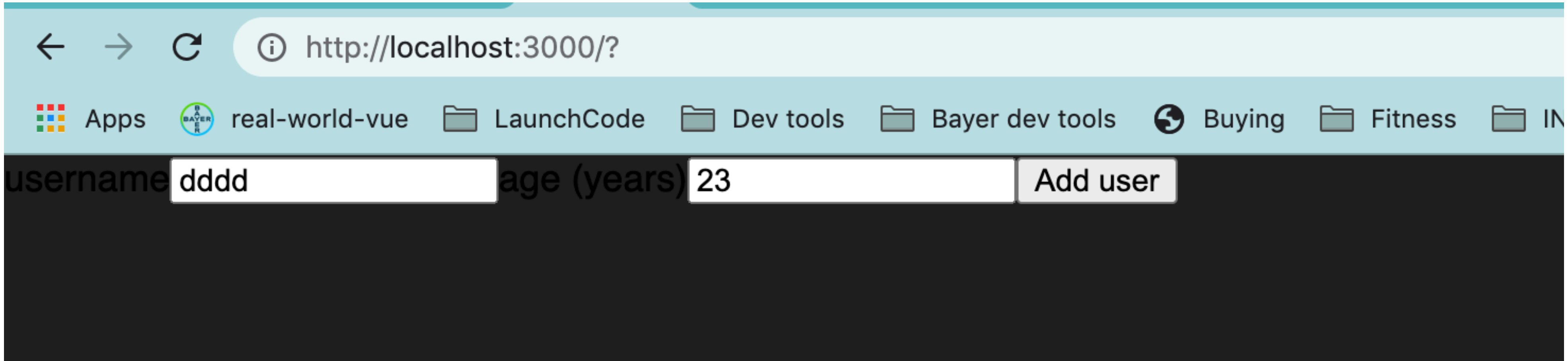


```
* {
  box-sizing: border-box;
}

html {
  font-family: sans-serif;
  background: □#1f1f1f;
}

body {
  margin: 0;
}
```

User Component



http://localhost:3000/?

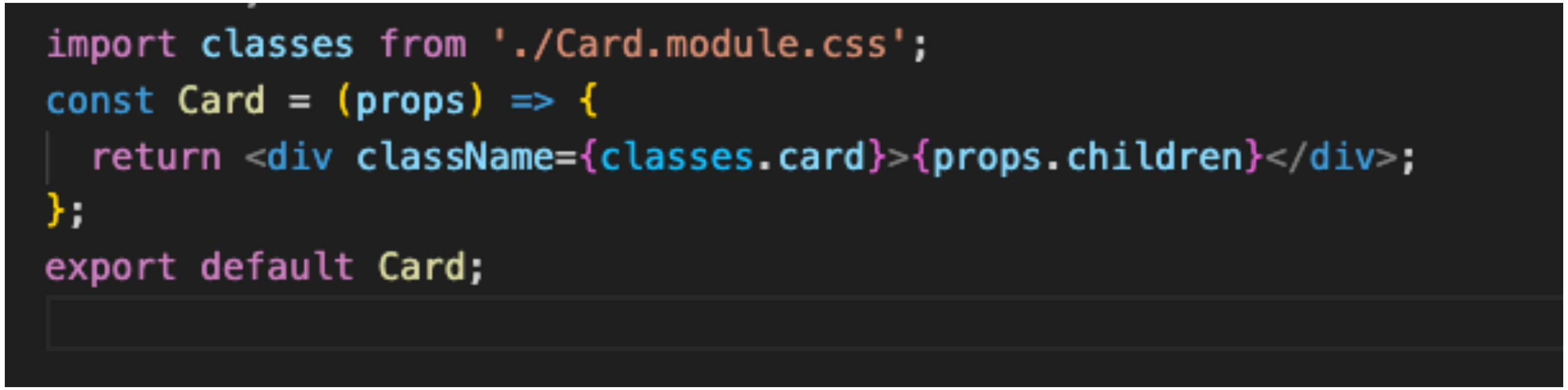


http://localhost:3000/?

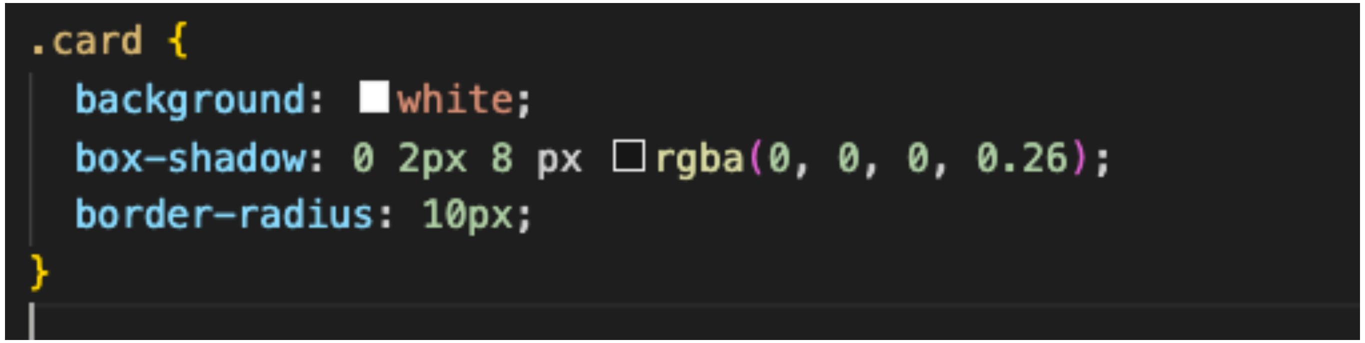
username dddd

age (years) 23

Card Component

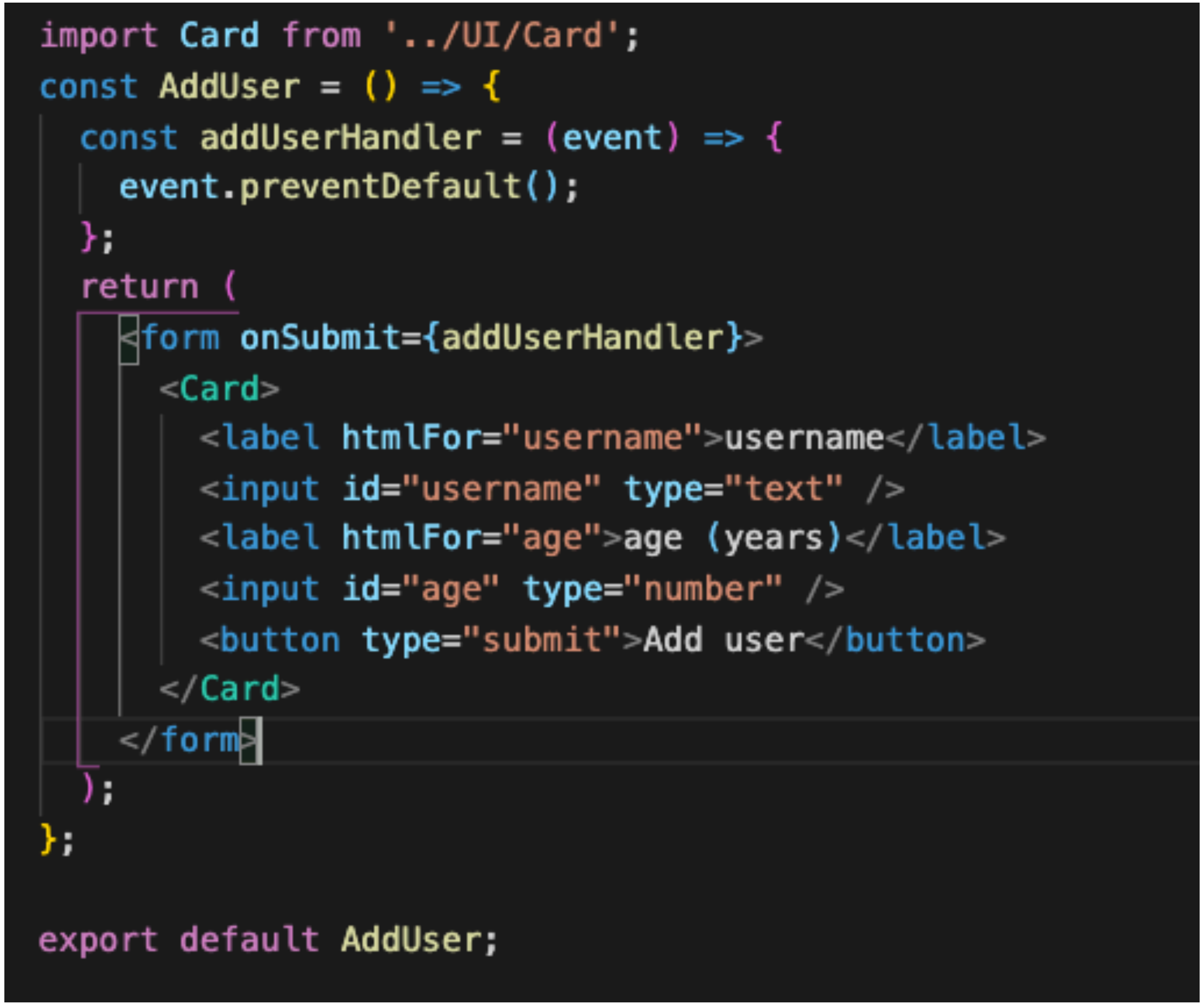


```
import classes from './Card.module.css';
const Card = (props) => {
  return <div className={classes.card}>{props.children}</div>;
};
export default Card;
```



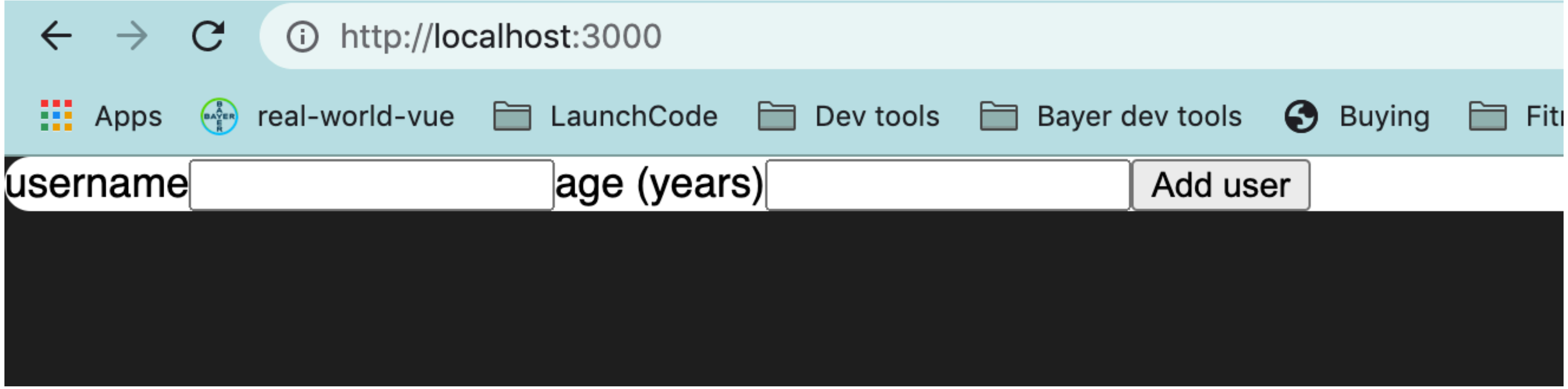
```
.card {
  background: white;
  box-shadow: 0 2px 8 px rgba(0, 0, 0, 0.26);
  border-radius: 10px;
}
```

Card Component



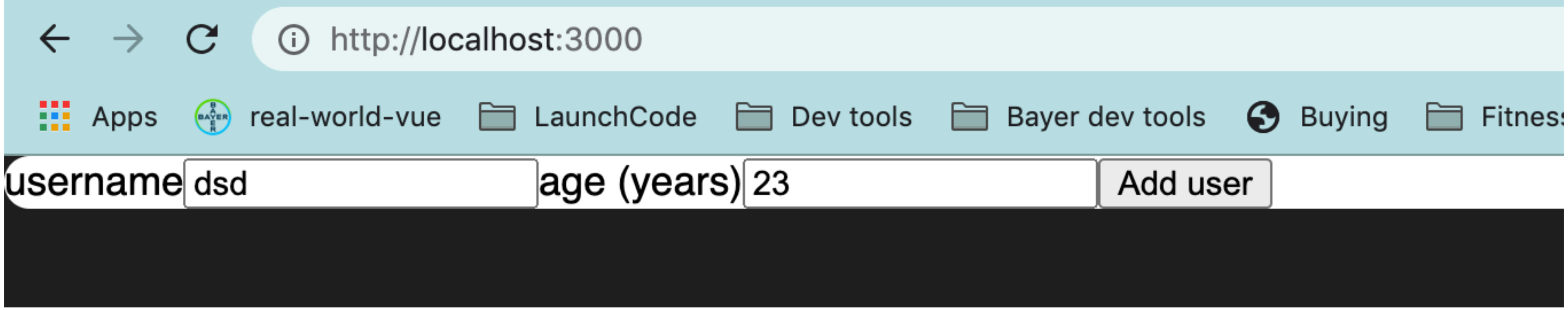
```
import Card from '../UI/Card';
const AddUser = () => {
  const addUserHandler = (event) => {
    event.preventDefault();
  };
  return (
    <form onSubmit={addUserHandler}>
      <Card>
        <label htmlFor="username">username</label>
        <input id="username" type="text" />
        <label htmlFor="age">age (years)</label>
        <input id="age" type="number" />
        <button type="submit">Add user</button>
      </Card>
    </form>
  );
};

export default AddUser;
```

Card Component



username

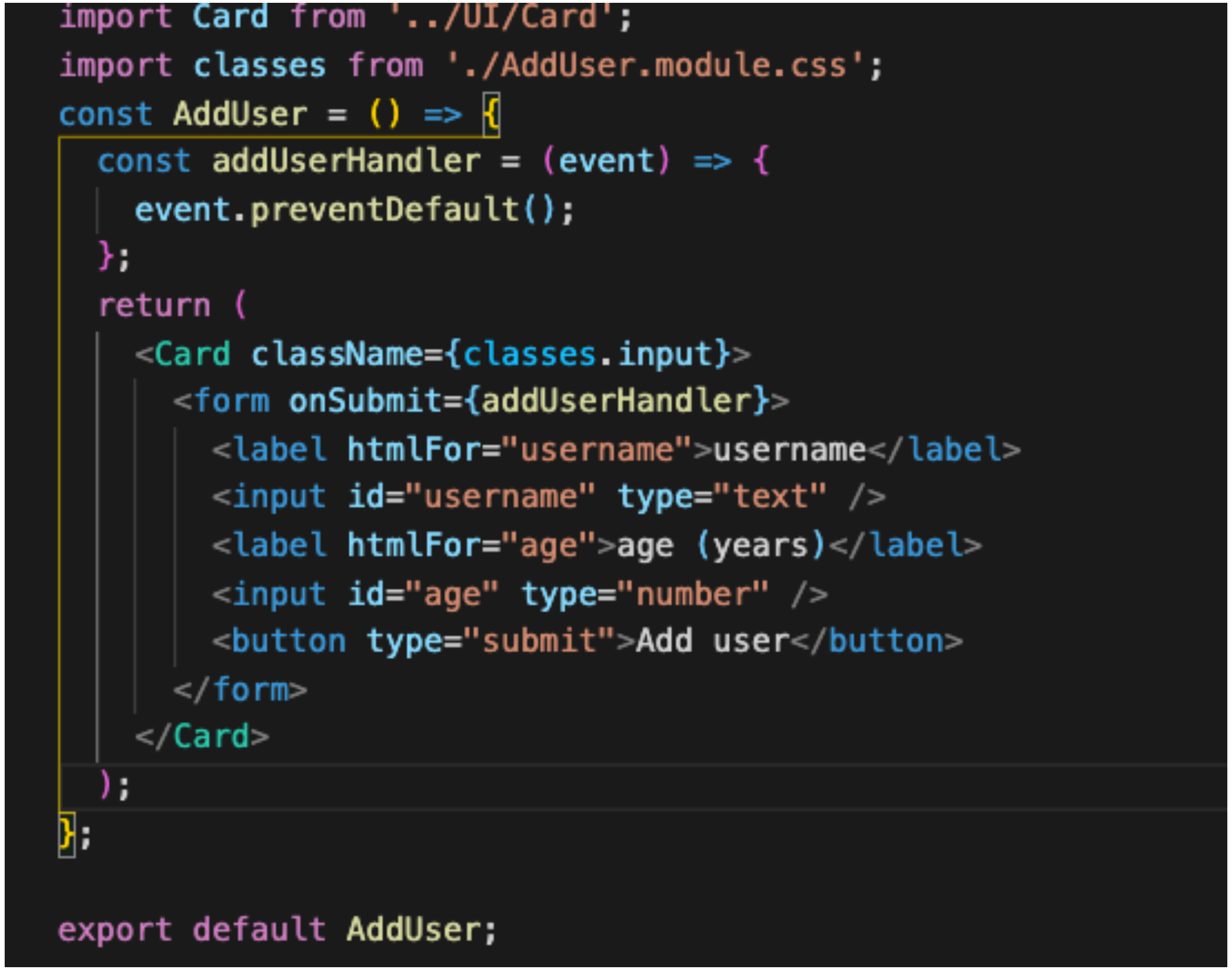
age (years)



username dsd

age (years) 23

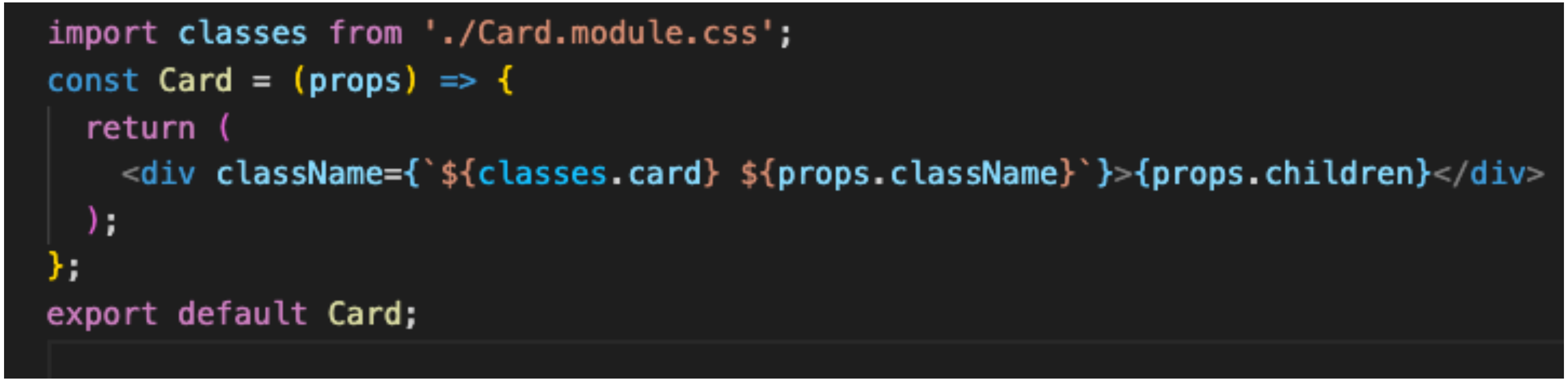
Card Component



```
import Card from '../UI/Card';
import classes from './AddUser.module.css';
const AddUser = () => {
  const addUserHandler = (event) => {
    event.preventDefault();
  };
  return (
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input id="username" type="text" />
        <label htmlFor="age">age (years)</label>
        <input id="age" type="number" />
        <button type="submit">Add user</button>
      </form>
    </Card>
  );
};

export default AddUser;
```

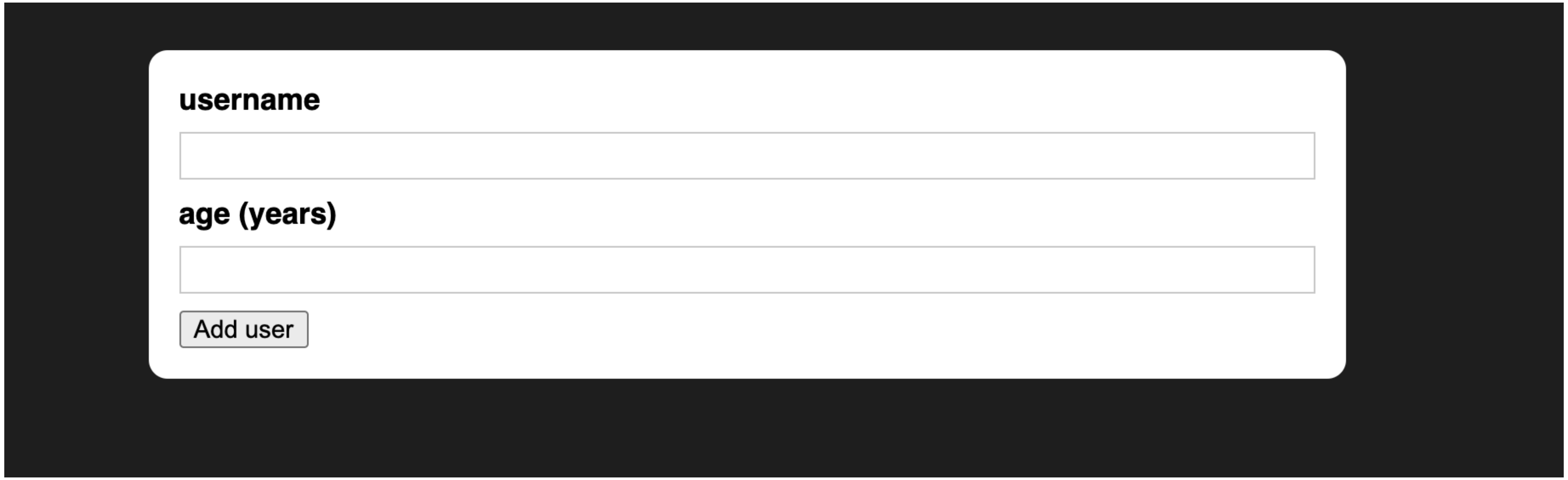
Card Component

https://github.com/academind/react-complete-guide-code/blob/08-practice-project/extra-files/AddUser.module.css



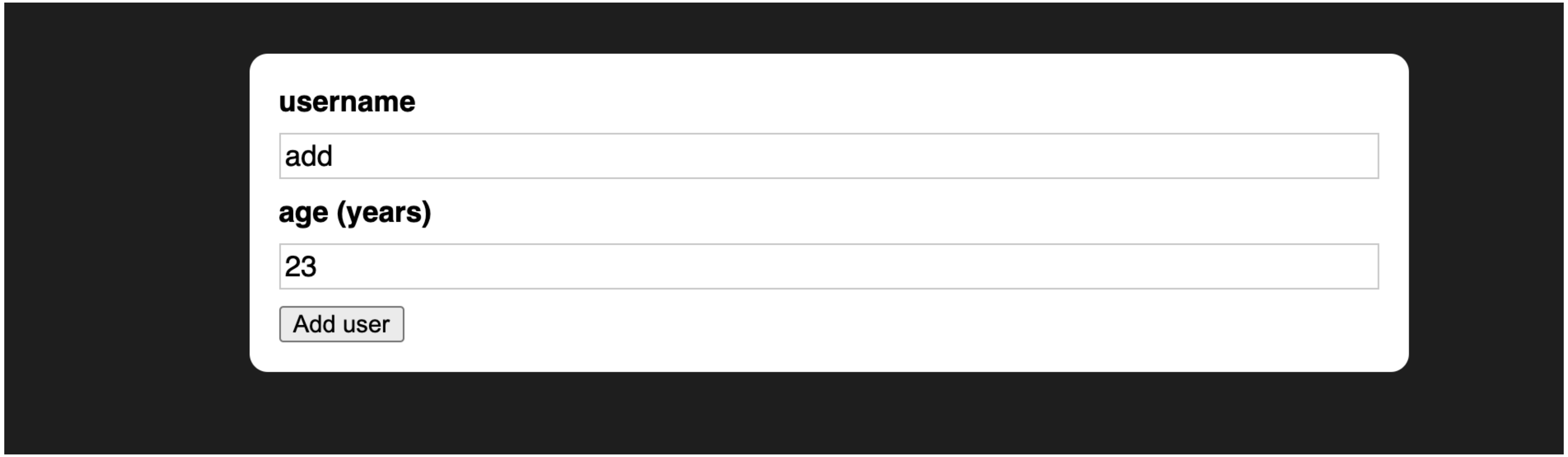
```
import classes from './Card.module.css';
const Card = (props) => {
  return (
    <div className={`${classes.card} ${props.className}`}>{props.children}</div>
  );
};
export default Card;
```

Card Component



username

age (years)



username

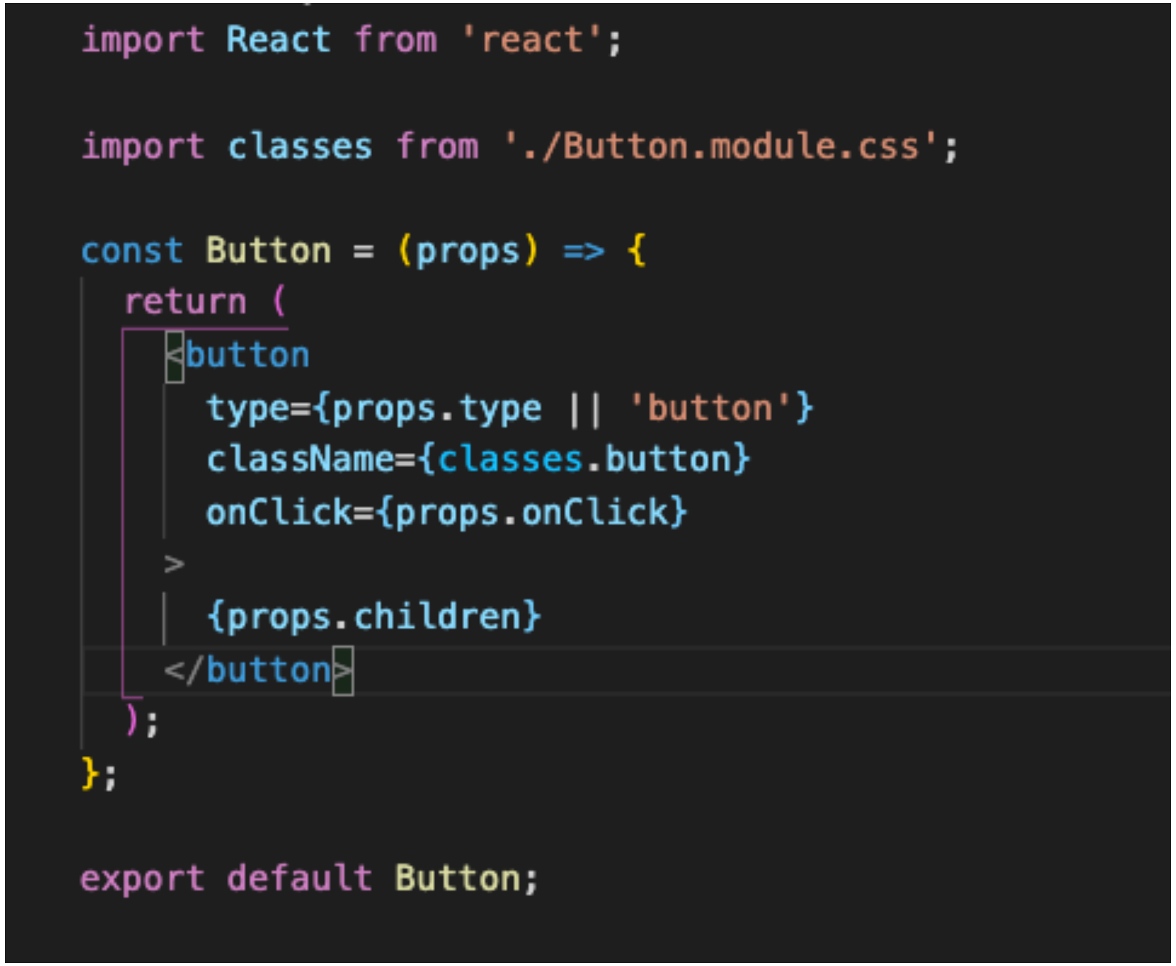
add

age (years)

23

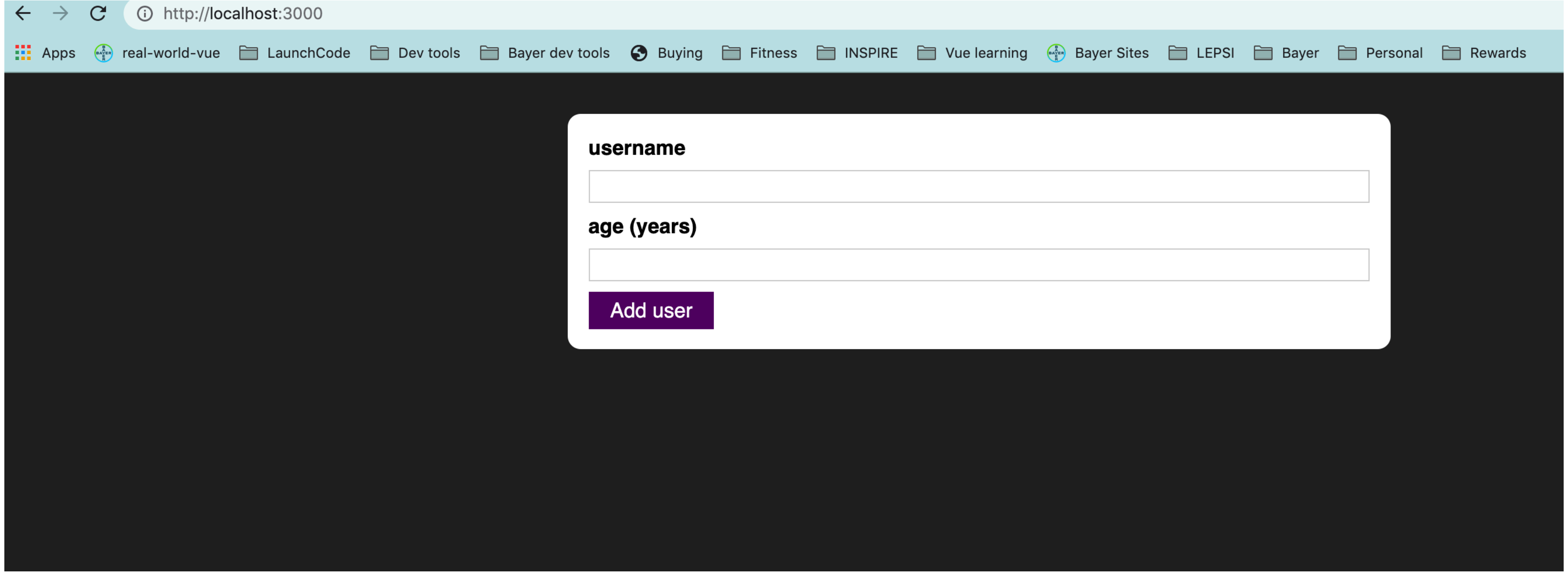
Button Component

https://github.com/academind/react-complete-guide-code/blob/08-practice-project/extra-files/Button.module.css



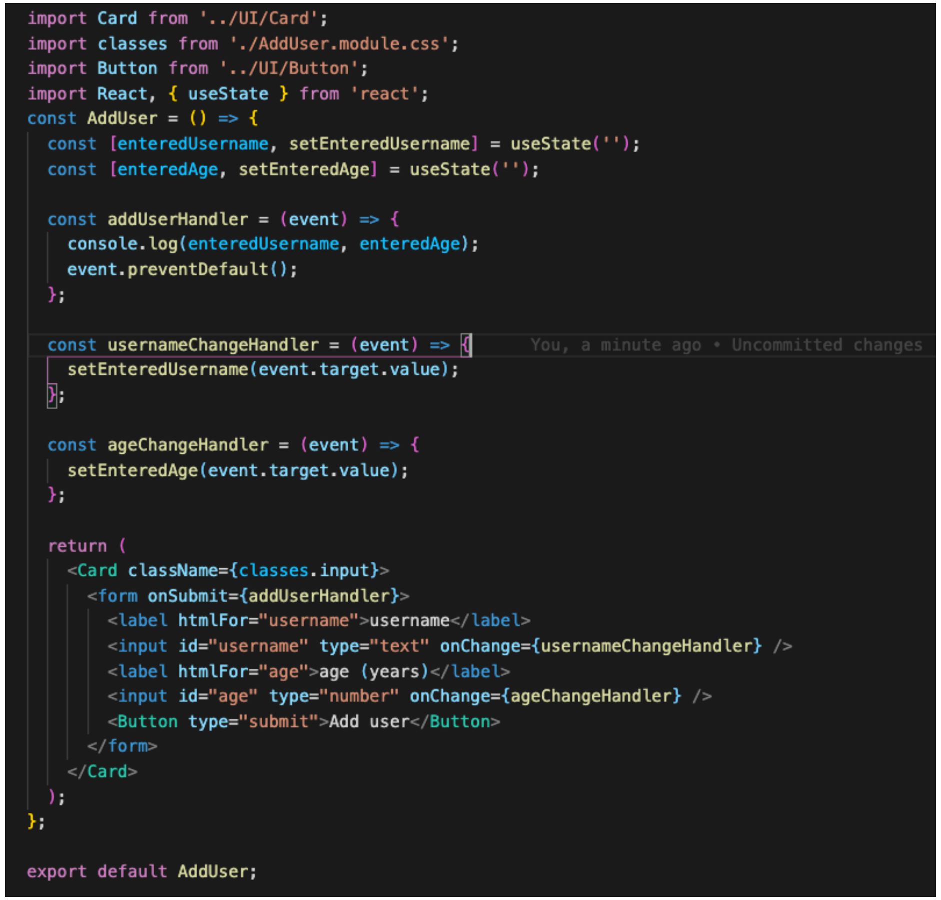
```
import React from 'react';

import classes from './Button.module.css';

const Button = (props) => {
  return (
    <button
      type={props.type || 'button'}
      className={classes.button}
      onClick={props.onClick}
    >
      {props.children}
    </button>
  );
};

export default Button;
```

Button Component



Managing the User Input State

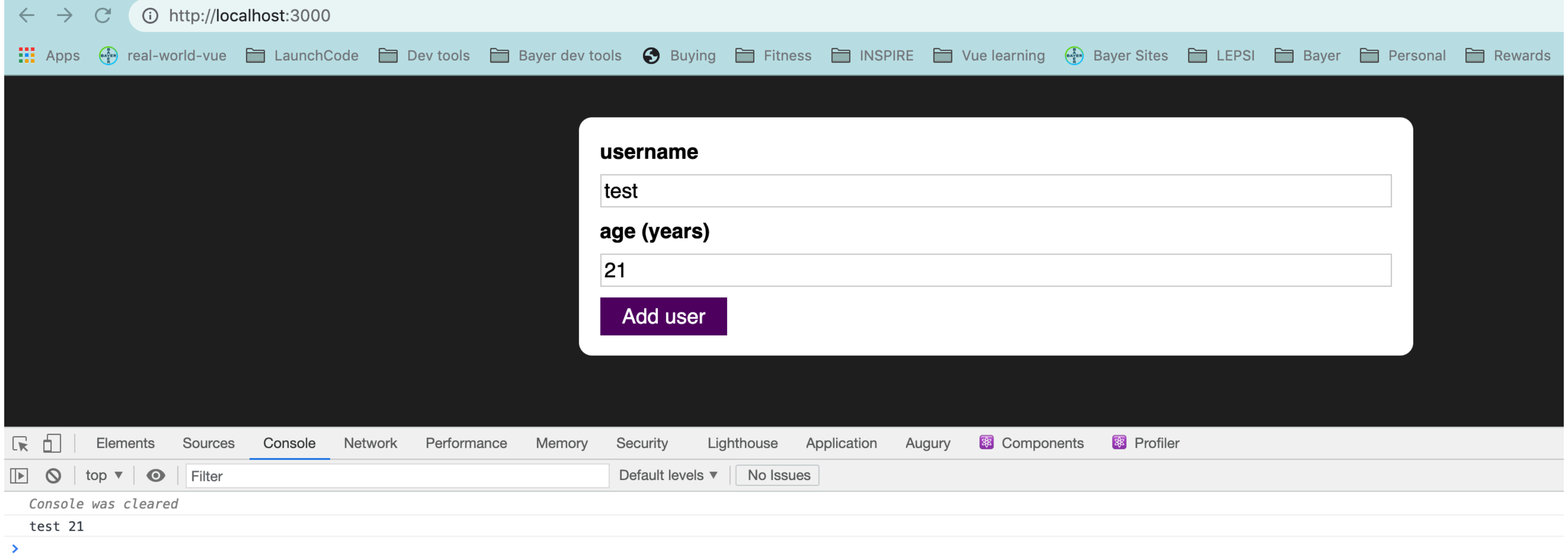
Collect values and save them using State Management



```
import Card from '../UI/Card';
import classes from './AddUser.module.css';
import Button from '../UI/Button';
import React, { useState } from 'react';
const AddUser = () => {
  const [enteredUsername, setEnteredUsername] = useState('');
  const [enteredAge, setEnteredAge] = useState('');

  const addUserHandler = (event) => {
    console.log(enteredUsername, enteredAge);
    event.preventDefault();
  };

  const usernameChangeHandler = (event) => {      You, a minute ago • Uncommitted changes
    setEnteredUsername(event.target.value);
  };

  const ageChangeHandler = (event) => {
    setEnteredAge(event.target.value);
  };

  return (
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input id="username" type="text" onChange={usernameChangeHandler} />
        <label htmlFor="age">age (years)</label>
        <input id="age" type="number" onChange={ageChangeHandler} />
        <Button type="submit">Add user</Button>
      </form>
    </Card>
  );
};

export default AddUser;
```

Managing the User Input State

Collect values and save them using State Management



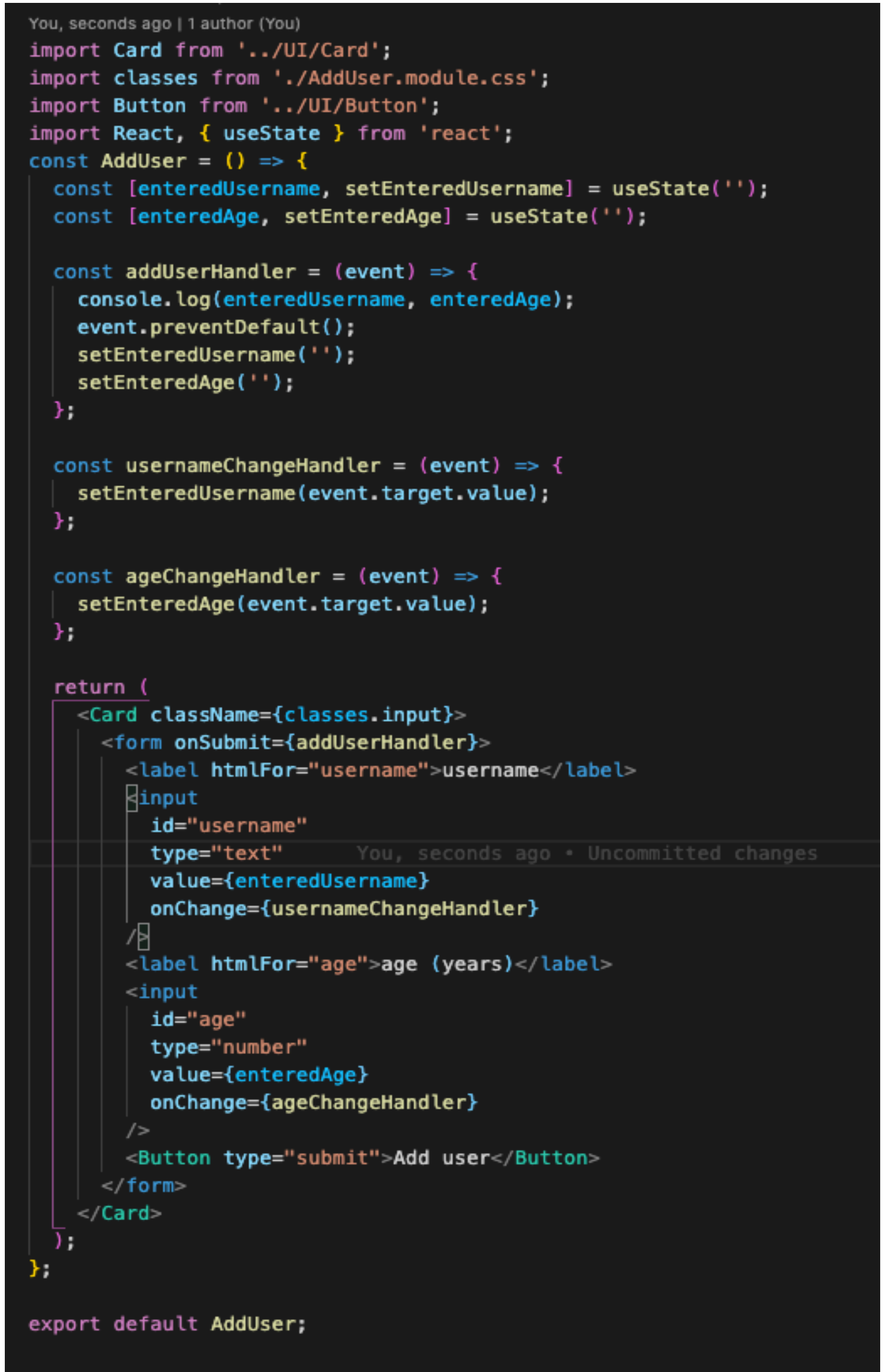
username: test

age (years): 21

```
Console was cleared
test 21
>
```

Adding Validation and Resetting Logic

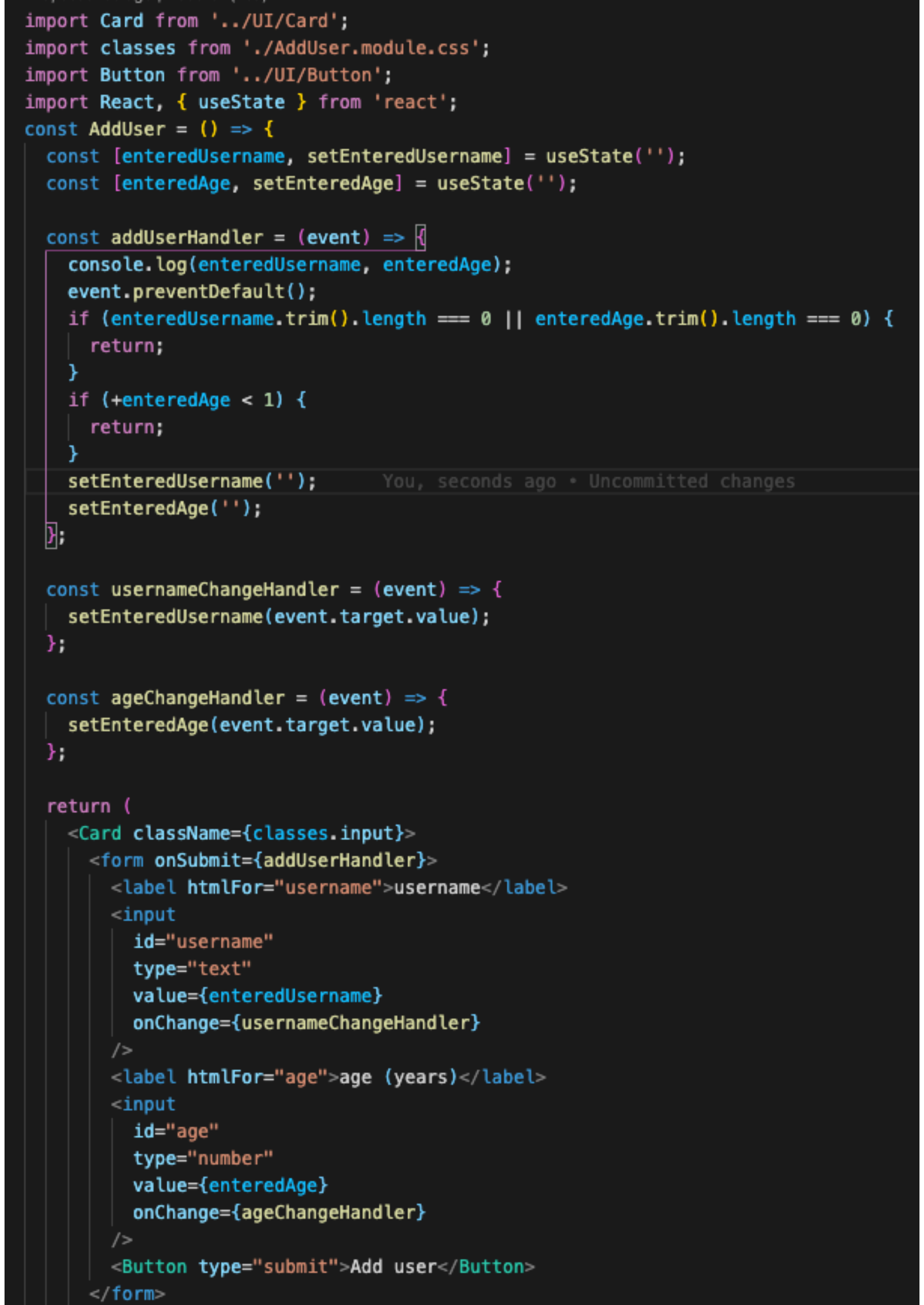
Resetting values after form submission.



```
You, seconds ago | 1 author (You)
import Card from '../UI/Card';
import classes from './AddUser.module.css';
import Button from '../UI/Button';
import React, { useState } from 'react';
const AddUser = () => {
  const [enteredUsername, setEnteredUsername] = useState('');
  const [enteredAge, setEnteredAge] = useState('');

  const addUserHandler = (event) => {
    console.log(enteredUsername, enteredAge);
    event.preventDefault();
    setEnteredUsername('');
    setEnteredAge('');
  };

  const usernameChangeHandler = (event) => {
    setEnteredUsername(event.target.value);
  };

  const ageChangeHandler = (event) => {
    setEnteredAge(event.target.value);
  };

  return (
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input
          id="username"
          type="text"       You, seconds ago • Uncommitted changes
          value={enteredUsername}
          onChange={usernameChangeHandler}
        />
        <label htmlFor="age">age (years)</label>
        <input
          id="age"
          type="number"
          value={enteredAge}
          onChange={ageChangeHandler}
        />
        <Button type="submit">Add user</Button>
      </form>
    </Card>
  );
};

export default AddUser;
```

Adding Validation and Resetting Logic

Validation after form submission.

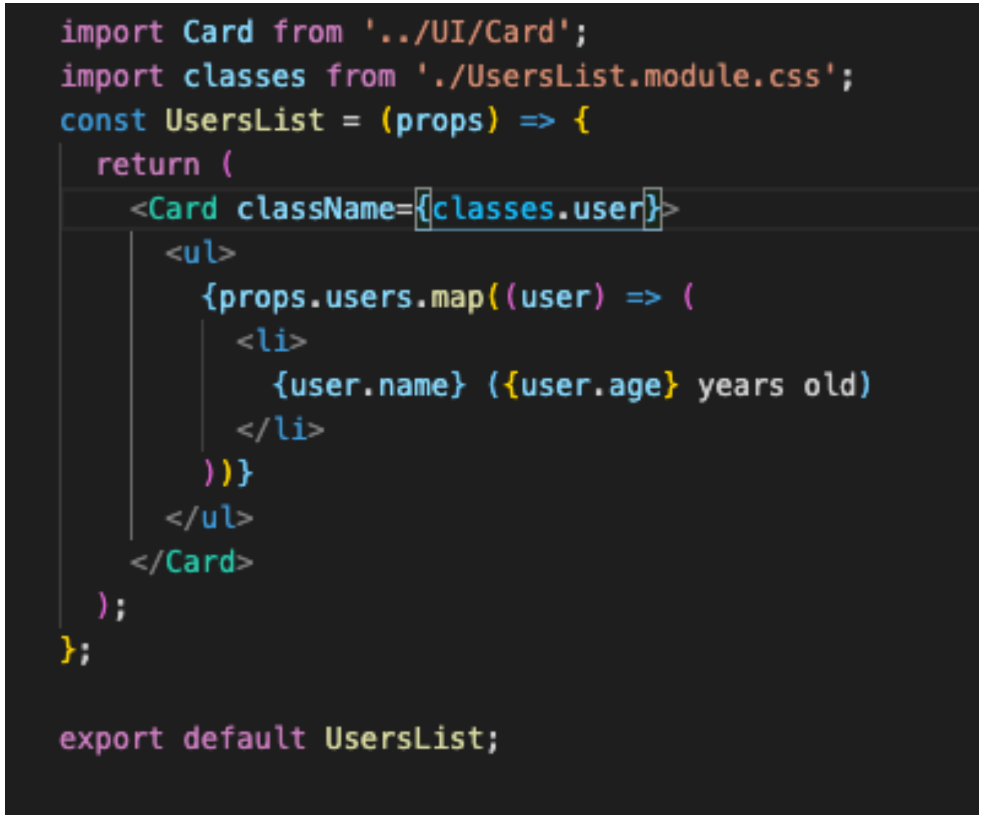


```
import Card from '../UI/Card';
import classes from './AddUser.module.css';
import Button from '../UI/Button';
import React, { useState } from 'react';
const AddUser = () => {
  const [enteredUsername, setEnteredUsername] = useState('');
  const [enteredAge, setEnteredAge] = useState('');

  const addUserHandler = (event) => {
    console.log(enteredUsername, enteredAge);
    event.preventDefault();
    if (enteredUsername.trim().length === 0 || enteredAge.trim().length === 0) {
      return;
    }
    if (+enteredAge < 1) {
      return;
    }
    setEnteredUsername('');      You, seconds ago • Uncommitted changes
    setEnteredAge('');
  };

  const usernameChangeHandler = (event) => {
    setEnteredUsername(event.target.value);
  };

  const ageChangeHandler = (event) => {
    setEnteredAge(event.target.value);
  };

  return (
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input
          id="username"
          type="text"
          value={enteredUsername}
          onChange={usernameChangeHandler}
        />
        <label htmlFor="age">age (years)</label>
        <input
          id="age"
          type="number"
          value={enteredAge}
          onChange={ageChangeHandler}
        />
        <Button type="submit">Add user</Button>
      </form>
```

Adding a Users List Component

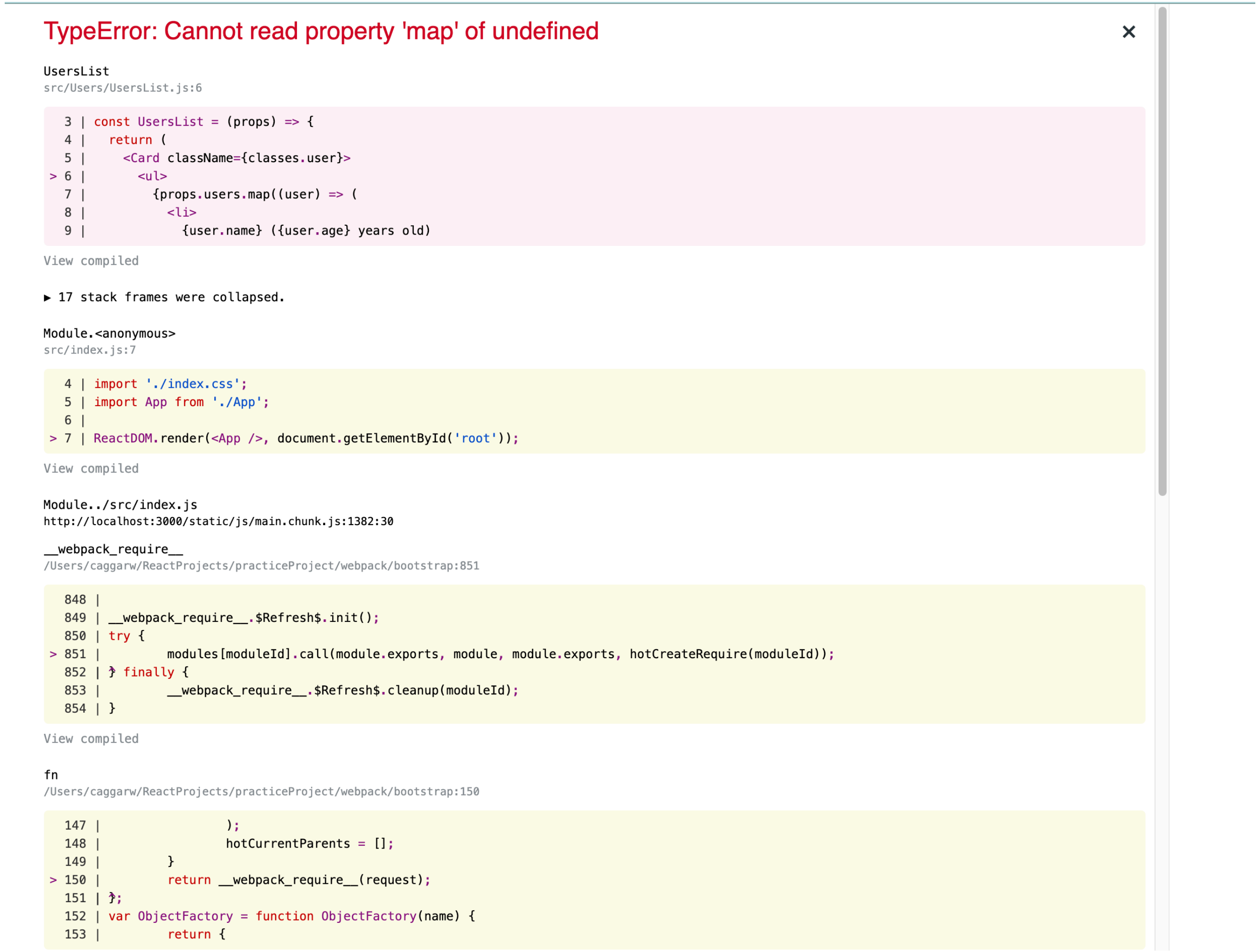
Where do we add the list of users?
Could add to AddUsers component
but that may not be the best idea.

We can have AddUsers to fetch users
and another component called UsersList
Component to output list of users.



```
import Card from '../UI/Card';
import classes from './UsersList.module.css';
const UsersList = (props) => {
  return (
    <Card className={classes.user}>
      <ul>
        {props.users.map((user) => (
          <li>
            {user.name} ({user.age} years old)
          </li>
        ))}
      </ul>
    </Card>
  );
};

export default UsersList;
```

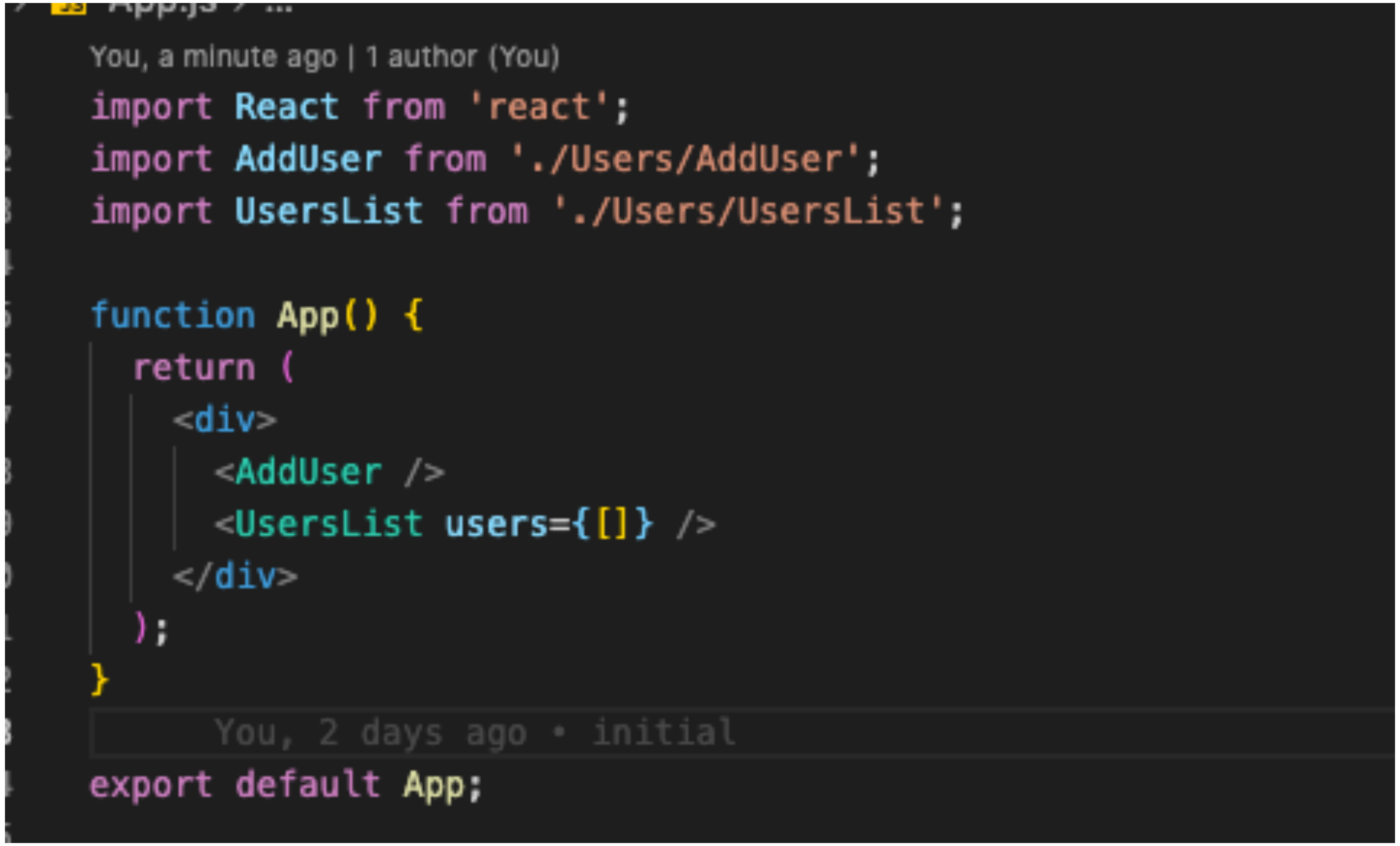
Adding a Users List Component



TypeError: Cannot read property 'map' of undefined

```
UsersList
src/Users/UsersList.js:6

  3 | const UsersList = (props) => {
  4 |   return (
  5 |     <Card className={classes.user}>
> 6 |       <ul>
  7 |         {props.users.map((user) => (
  8 |           <li>
  9 |             {user.name} ({user.age} years old)

View compiled

▶ 17 stack frames were collapsed.

Module.<anonymous>
src/index.js:7

  4 | import './index.css';
  5 | import App from './App';
  6 |
> 7 | ReactDOM.render(<App />, document.getElementById('root'));

View compiled

Module../src/index.js
http://localhost:3000/static/js/main.chunk.js:1382:30

__webpack_require__
/Users/caggarw/ReactProjects/practiceProject/webpack/bootstrap:851

  848 |
  849 | __webpack_require__.$Refresh$.init();
  850 | try {
> 851 |         modules[moduleId].call(module.exports, module, module.exports, hotCreateRequire(moduleId));
  852 | } finally {
  853 |         __webpack_require__.$Refresh$.cleanup(moduleId);
  854 | }

View compiled

fn
/Users/caggarw/ReactProjects/practiceProject/webpack/bootstrap:150

  147 |                 );
  148 |                 hotCurrentParents = [];
  149 |         }
> 150 |         return __webpack_require__(request);
  151 | };
  152 | var ObjectFactory = function ObjectFactory(name) {
  153 |         return {
```

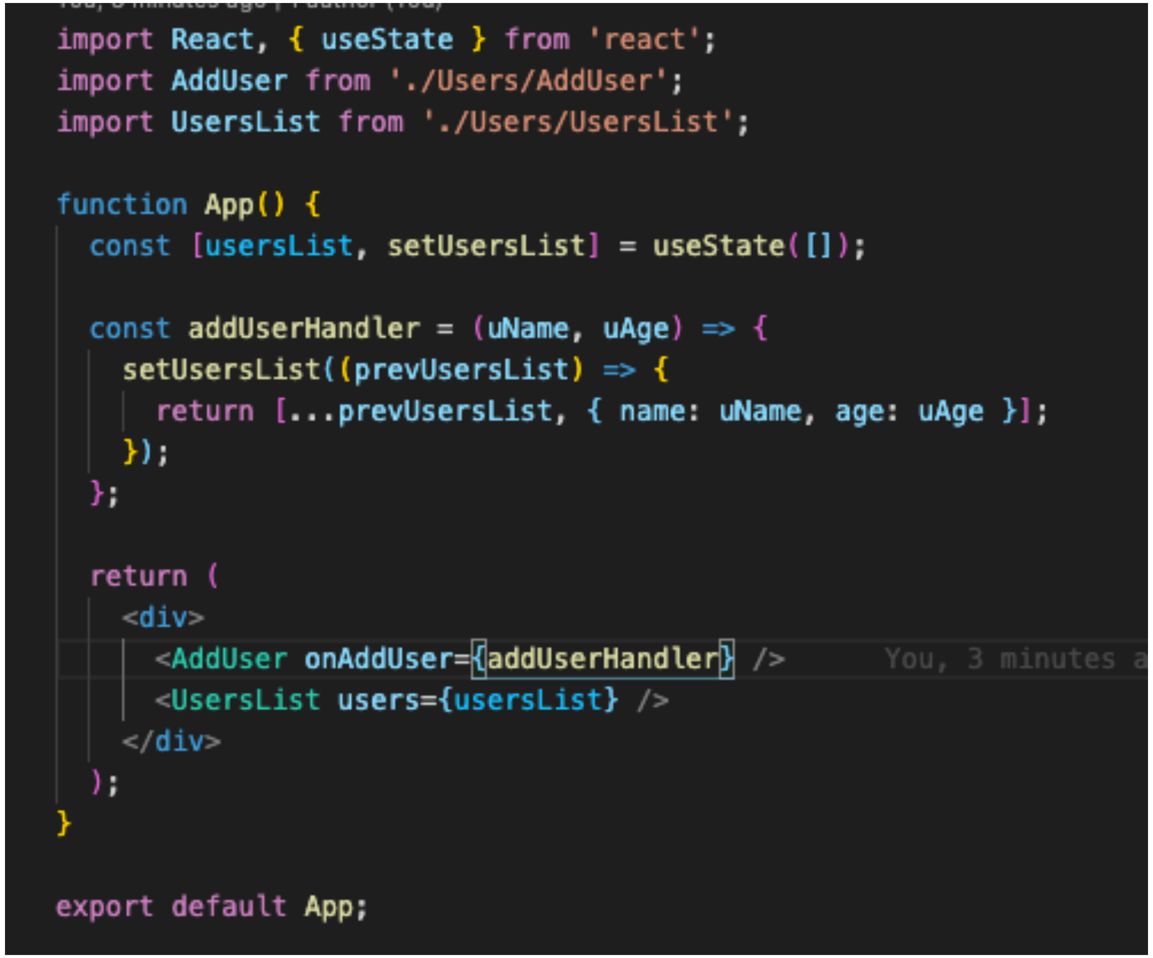
Adding a Users List Component



```
You, a minute ago | 1 author (You)
import React from 'react';
import AddUser from './Users/AddUser';
import UsersList from './Users/UsersList';

function App() {
  return (
    <div>
      <AddUser />
      <UsersList users={[]} />
    </div>
  );
}
      You, 2 days ago • initial
export default App;
```

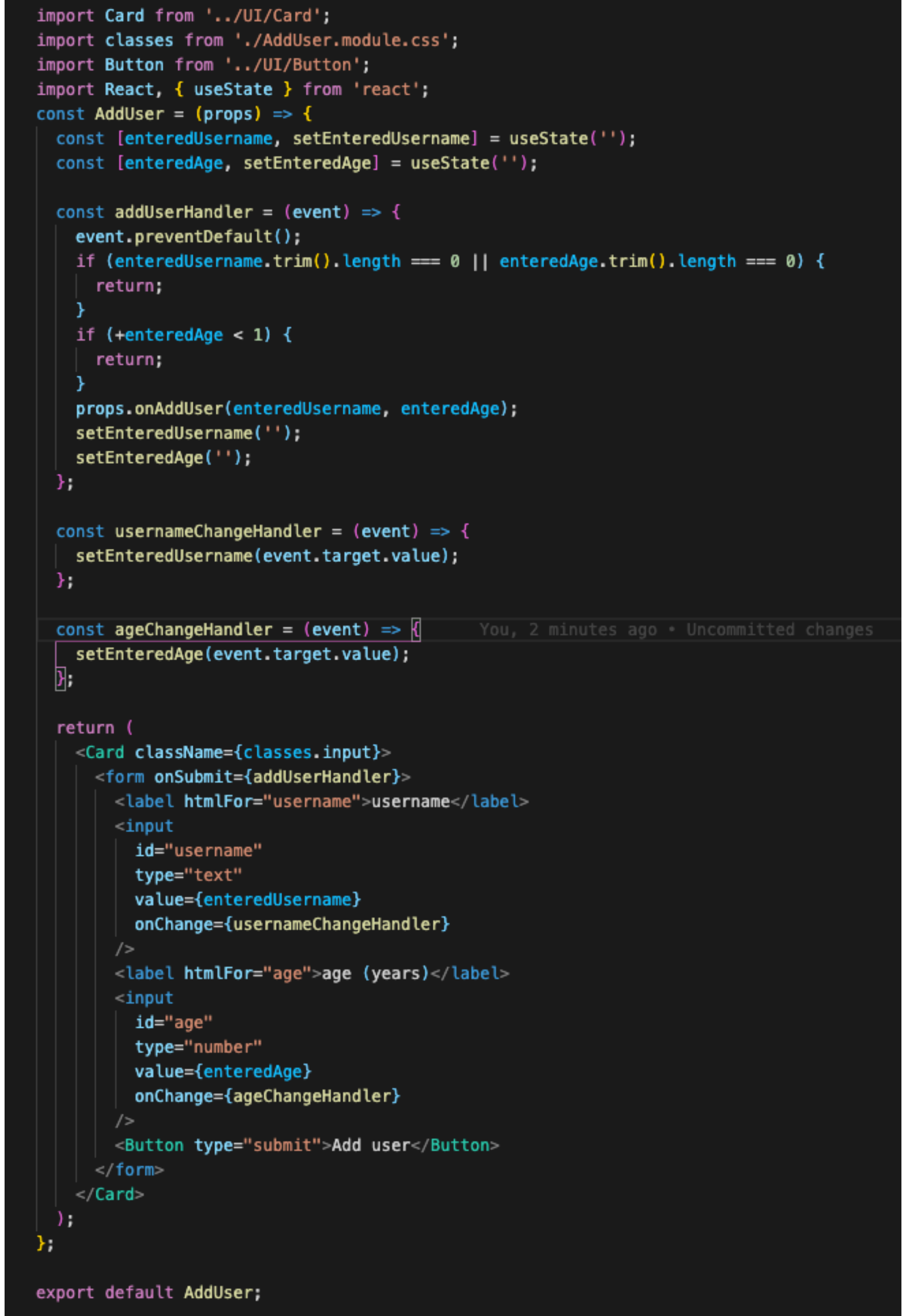
Adding a Users List Component

Need to lift the state up.
Add Users – users List
Need that in Users List
Need to manage state in App.js that is above both components.
Need to lift state to App.js
useState and state managed.



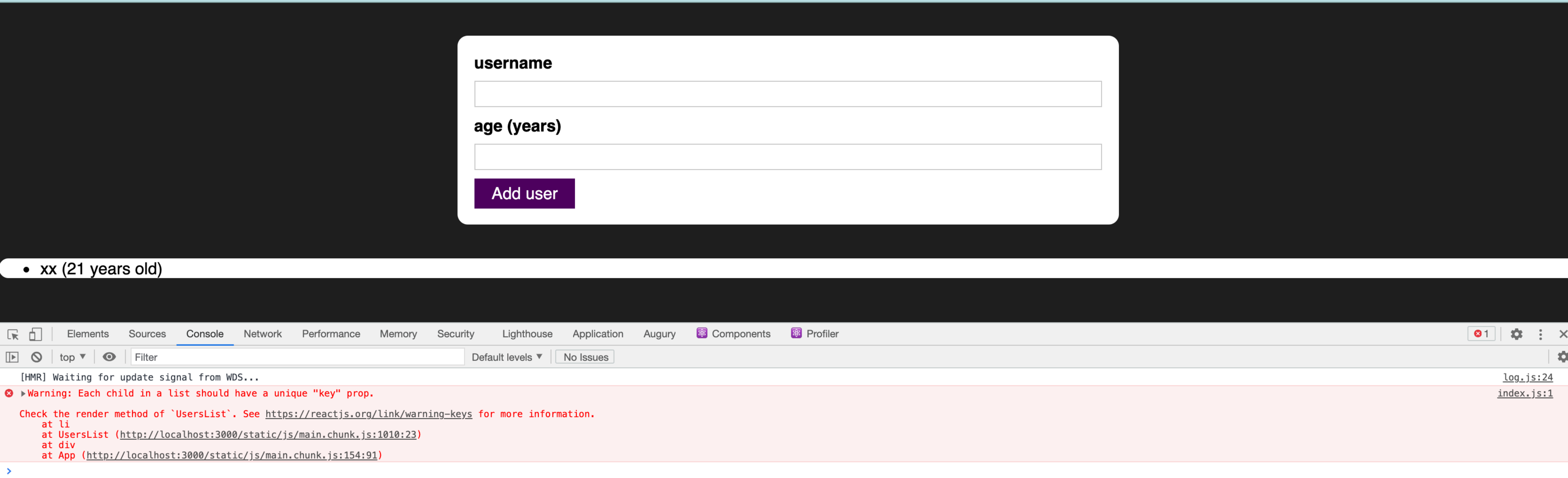
```
import React, { useState } from 'react';
import AddUser from './Users/AddUser';
import UsersList from './Users/UsersList';

function App() {
  const [usersList, setUsersList] = useState([]);

  const addUserHandler = (uName, uAge) => {
    setUsersList((prevUsersList) => {
      return [...prevUsersList, { name: uName, age: uAge }];
    });
  };

  return (
    <div>
      <AddUser onAddUser={addUserHandler} />      You, 3 minutes a
      <UsersList users={usersList} />
    </div>
  );
}

export default App;
```

Adding a Users List Component



```
import Card from '../UI/Card';
import classes from './AddUser.module.css';
import Button from '../UI/Button';
import React, { useState } from 'react';
const AddUser = (props) => {
  const [enteredUsername, setEnteredUsername] = useState('');
  const [enteredAge, setEnteredAge] = useState('');

  const addUserHandler = (event) => {
    event.preventDefault();
    if (enteredUsername.trim().length === 0 || enteredAge.trim().length === 0) {
      return;
    }
    if (+enteredAge < 1) {
      return;
    }
    props.onAddUser(enteredUsername, enteredAge);
    setEnteredUsername('');
    setEnteredAge('');
  };

  const usernameChangeHandler = (event) => {
    setEnteredUsername(event.target.value);
  };

  const ageChangeHandler = (event) => {        You, 2 minutes ago • Uncommitted changes
    setEnteredAge(event.target.value);
  };

  return (
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input
          id="username"
          type="text"
          value={enteredUsername}
          onChange={usernameChangeHandler}
        />
        <label htmlFor="age">age (years)</label>
        <input
          id="age"
          type="number"
          value={enteredAge}
          onChange={ageChangeHandler}
        />
        <Button type="submit">Add user</Button>
      </form>
    </Card>
  );
};

export default AddUser;
```

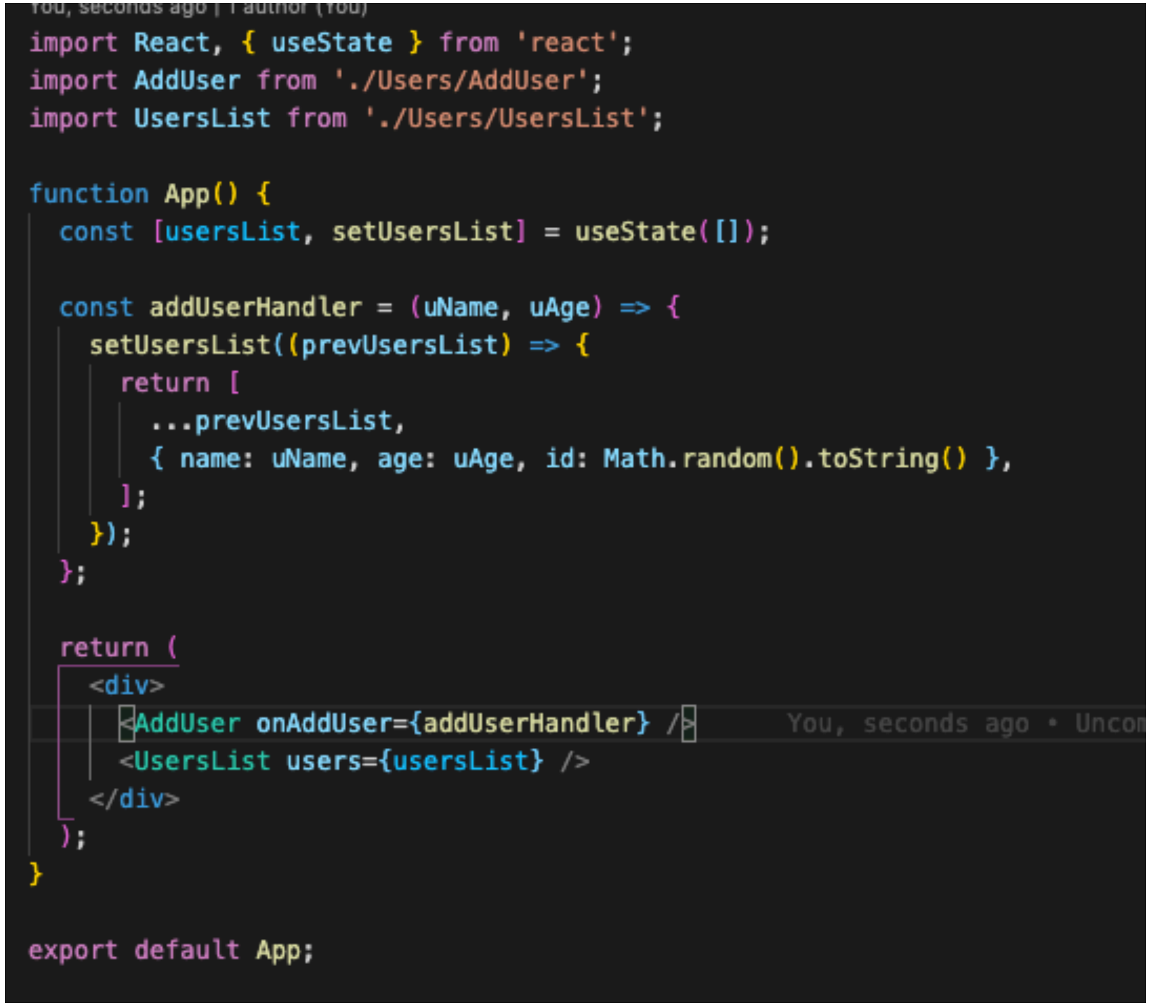
Adding a Users List Component



- xx (21 years old)

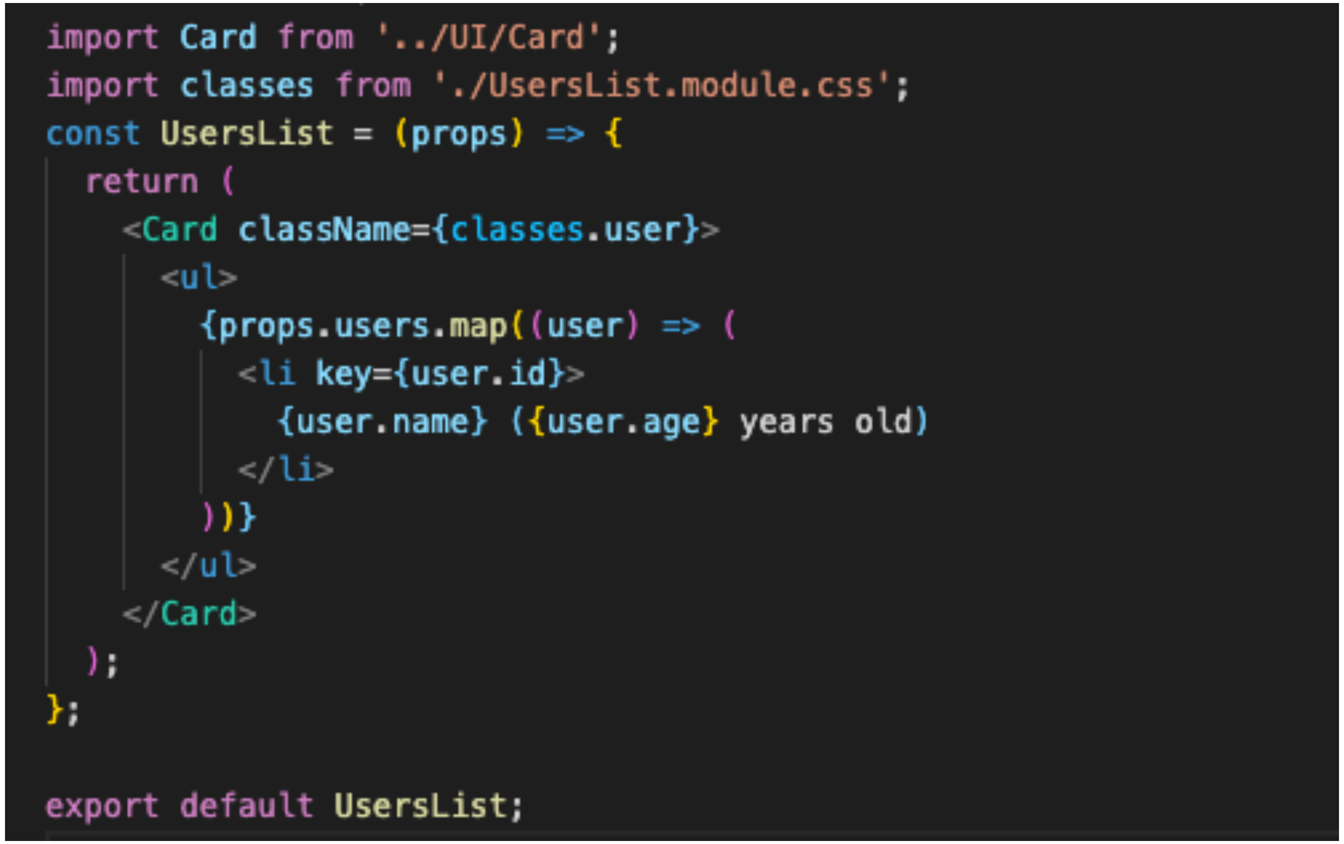
```
[HMR] Waiting for update signal from WDS...                                  log.js:24
▸Warning: Each child in a list should have a unique "key" prop.              index.js:1

Check the render method of `UsersList`. See https://reactjs.org/link/warning-keys for more information.
    at li
    at UsersList (http://localhost:3000/static/js/main.chunk.js:1010:23)
    at div
    at App (http://localhost:3000/static/js/main.chunk.js:154:91)
```

Adding a Users List Component



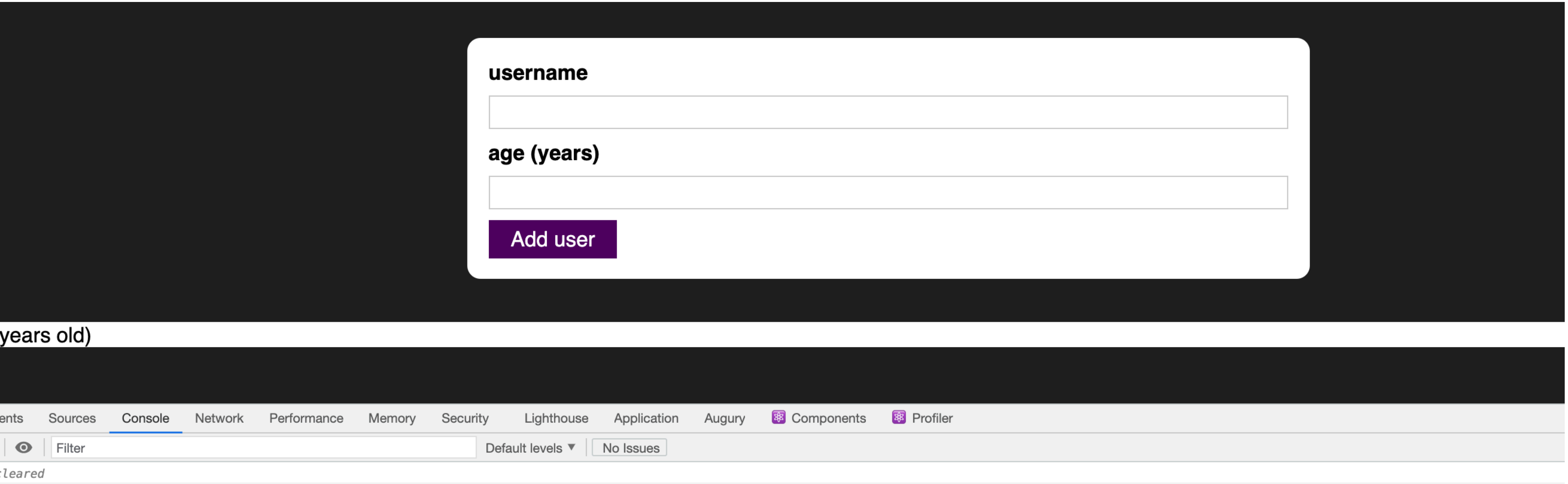
```
You, seconds ago | 1 author (You)
import React, { useState } from 'react';
import AddUser from './Users/AddUser';
import UsersList from './Users/UsersList';

function App() {
  const [usersList, setUsersList] = useState([]);

  const addUserHandler = (uName, uAge) => {
    setUsersList((prevUsersList) => {
      return [
        ...prevUsersList,
        { name: uName, age: uAge, id: Math.random().toString() },
      ];
    });
  };

  return (
    <div>
      <AddUser onAddUser={addUserHandler} />      You, seconds ago • Uncom
      <UsersList users={usersList} />
    </div>
  );
}

export default App;
```

Adding a Users List Component



```
import Card from '../UI/Card';
import classes from './UsersList.module.css';
const UsersList = (props) => {
  return (
    <Card className={classes.user}>
      <ul>
        {props.users.map((user) => (
          <li key={user.id}>
            {user.name} ({user.age} years old)
          </li>
        ))}
      </ul>
    </Card>
  );
};

export default UsersList;
```

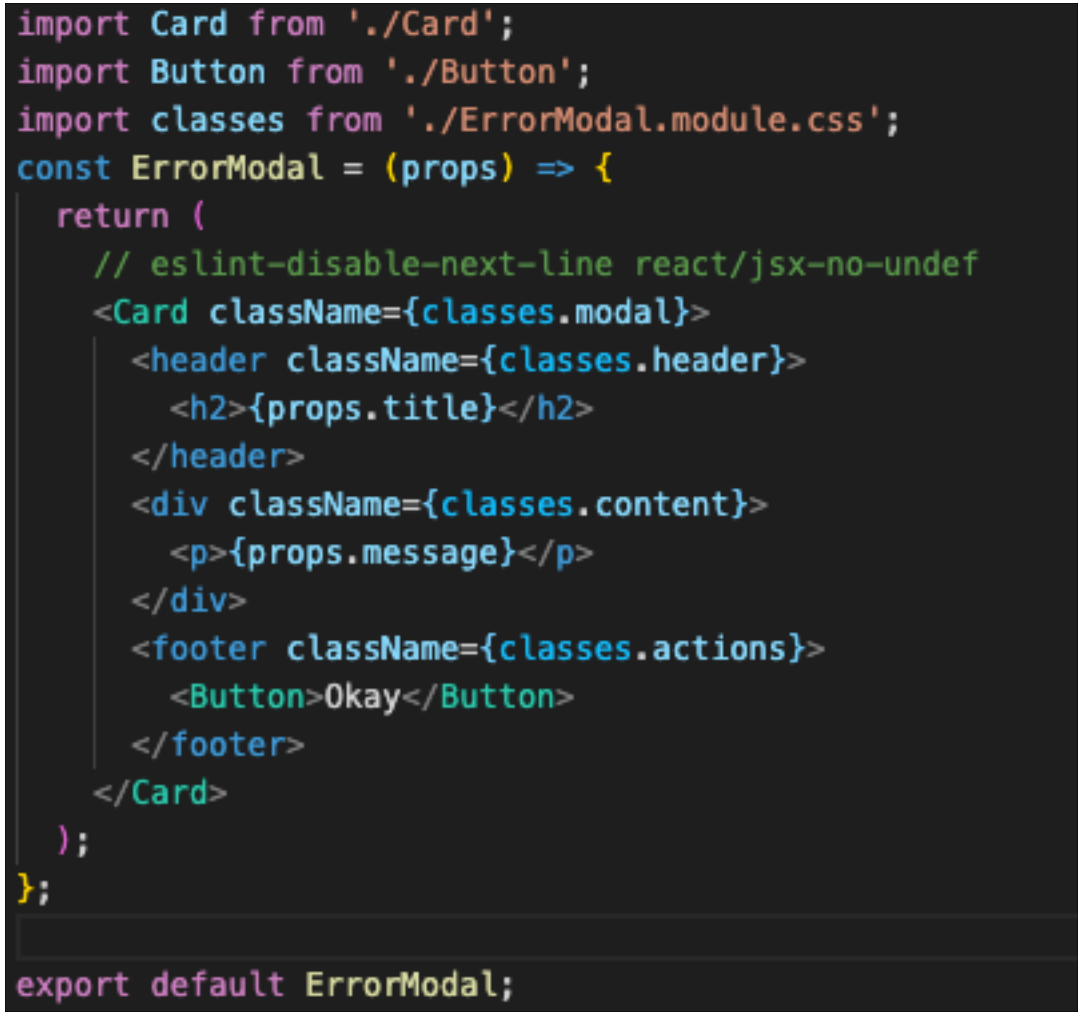
Adding a Users List Component



years old)

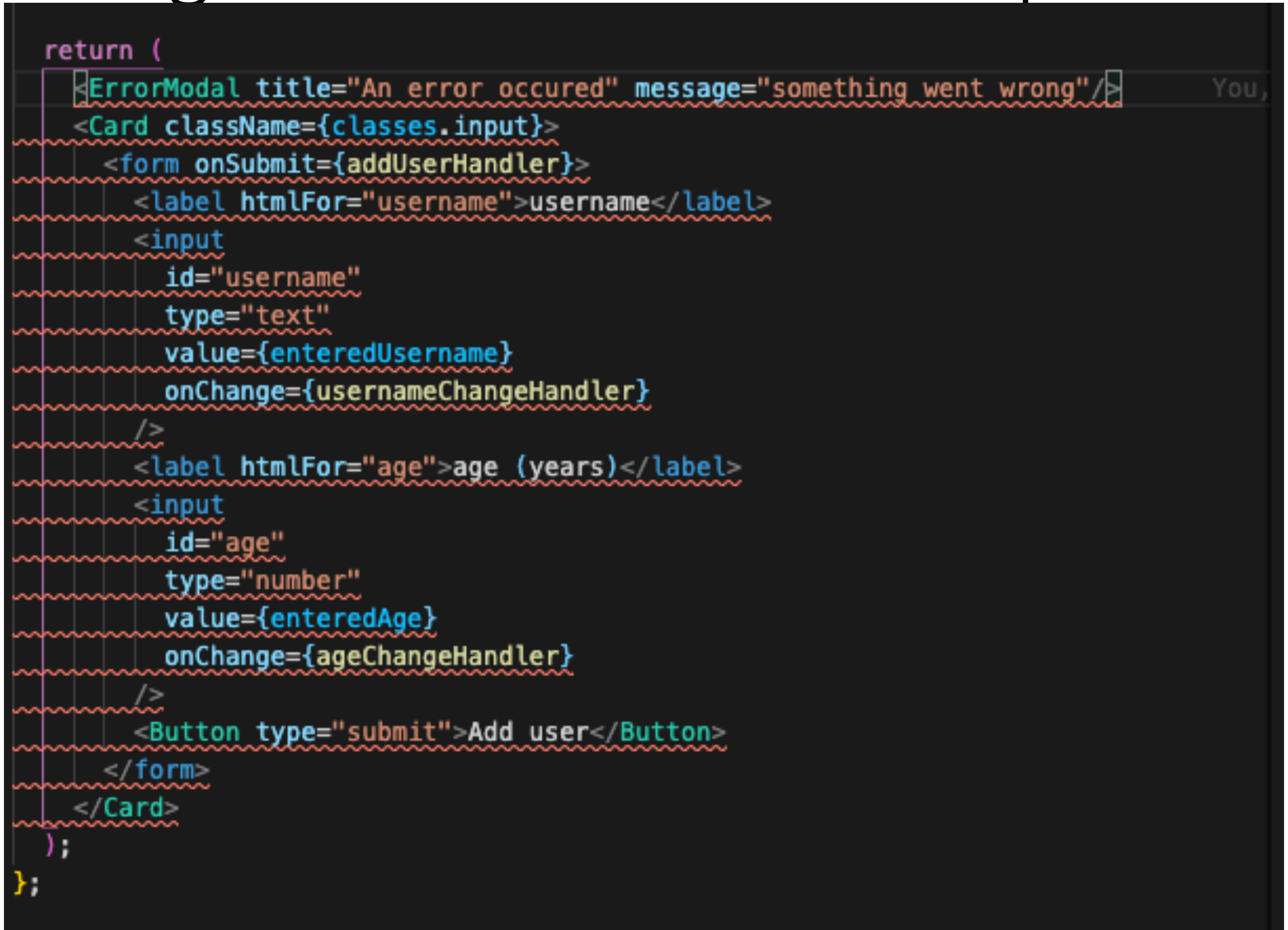
leared

Adding the Error Modal Component



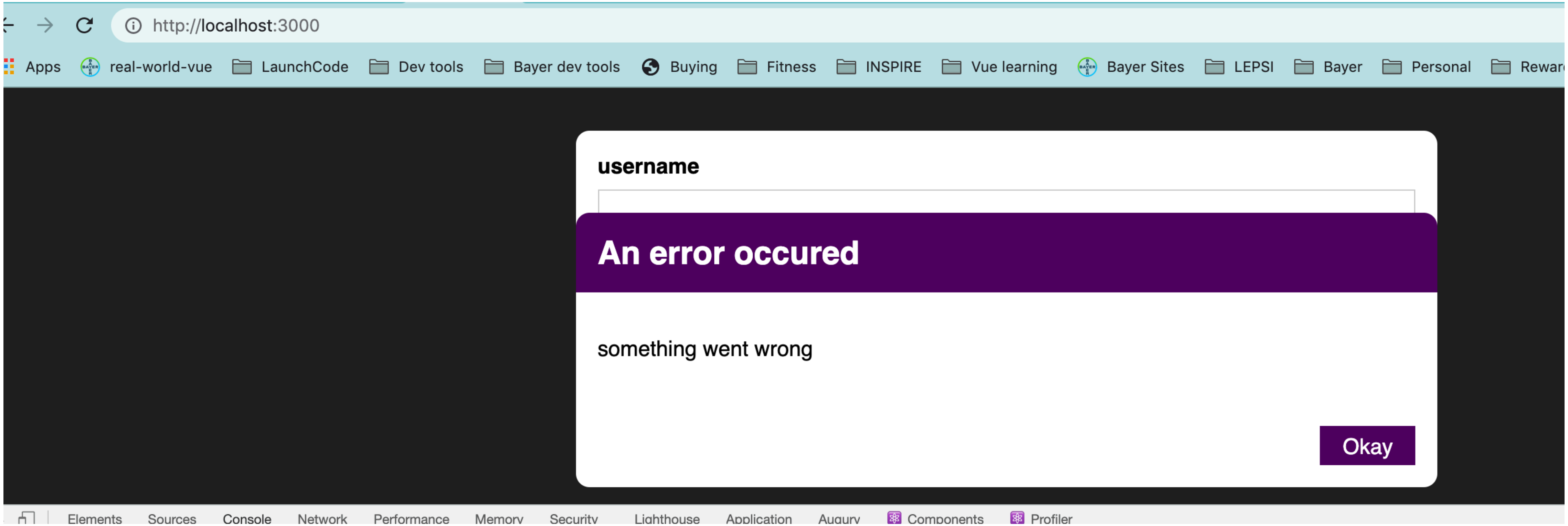
```
import Card from './Card';
import Button from './Button';
import classes from './ErrorModal.module.css';
const ErrorModal = (props) => {
  return (
    // eslint-disable-next-line react/jsx-no-undef
    <Card className={classes.modal}>
      <header className={classes.header}>
        <h2>{props.title}</h2>
      </header>
      <div className={classes.content}>
        <p>{props.message}</p>
      </div>
      <footer className={classes.actions}>
        <Button>Okay</Button>
      </footer>
    </Card>
  );
};

export default ErrorModal;
```

Adding the Error Modal Component



```
  return (
    <ErrorModal title="An error occured" message="something went wrong"/>      You,
    <Card className={classes.input}>
      <form onSubmit={addUserHandler}>
        <label htmlFor="username">username</label>
        <input
          id="username"
          type="text"
          value={enteredUsername}
          onChange={usernameChangeHandler}
        />
        <label htmlFor="age">age (years)</label>
        <input
          id="age"
          type="number"
          value={enteredAge}
          onChange={ageChangeHandler}
        />
        <Button type="submit">Add user</Button>
      </form>
    </Card>
  );
};
```

Adding the Error Modal Component

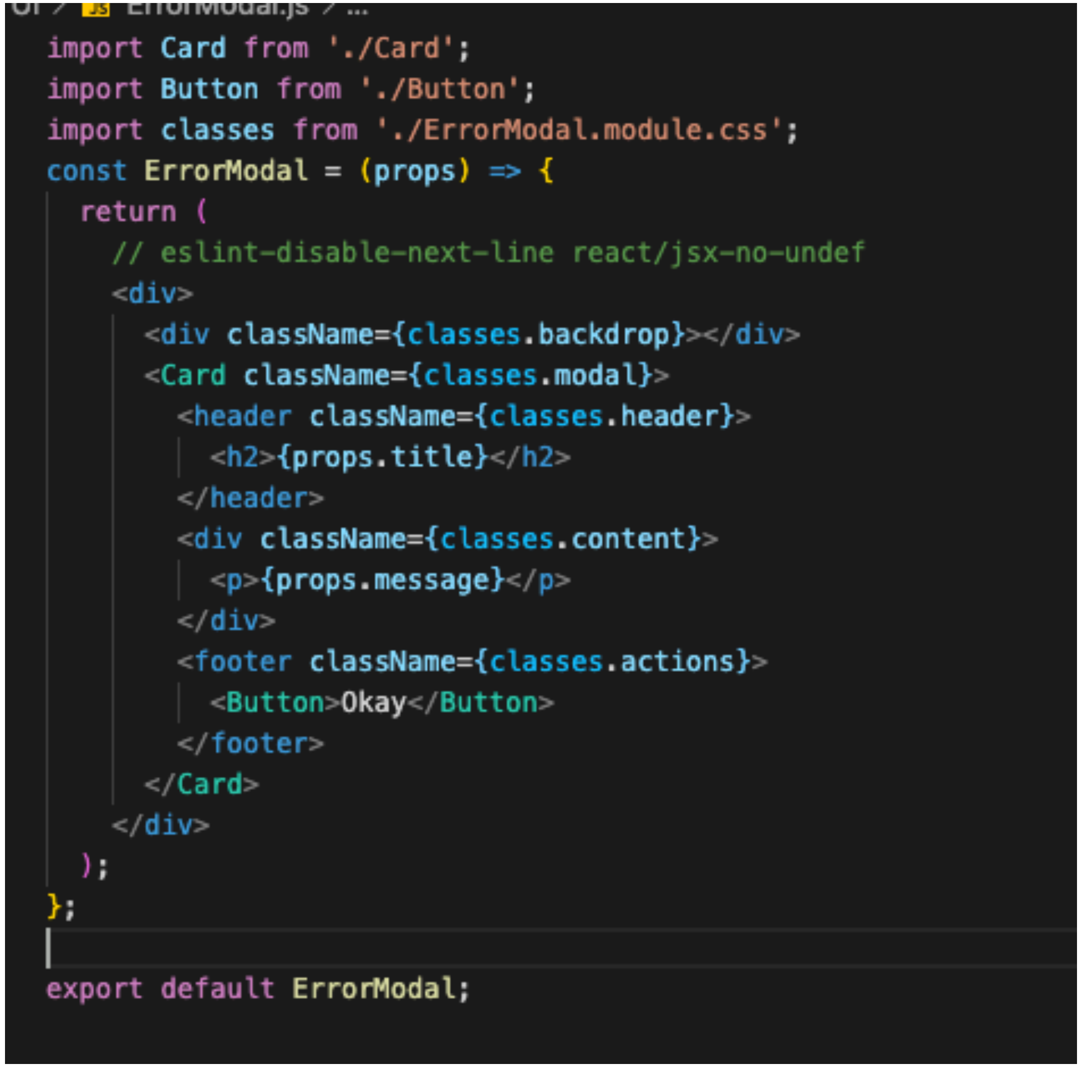


username

An error occured

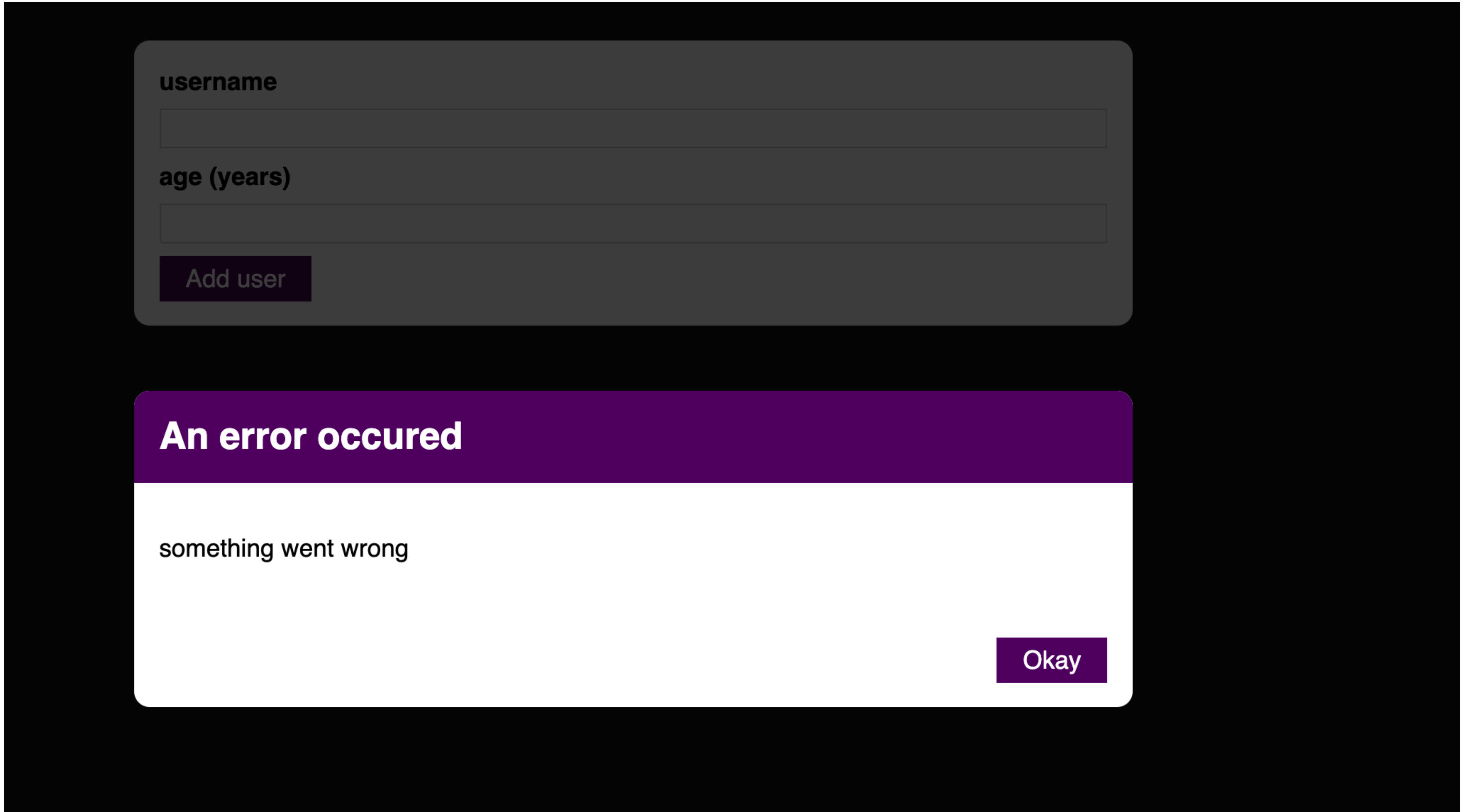
something went wrong

Adding the Error Modal Component



```
UI > JS ErrorModal.js > ...
import Card from './Card';
import Button from './Button';
import classes from './ErrorModal.module.css';
const ErrorModal = (props) => {
  return (
    // eslint-disable-next-line react/jsx-no-undef
    <div>
      <div className={classes.backdrop}></div>
      <Card className={classes.modal}>
        <header className={classes.header}>
          <h2>{props.title}</h2>
        </header>
        <div className={classes.content}>
          <p>{props.message}</p>
        </div>
        <footer className={classes.actions}>
          <Button>Okay</Button>
        </footer>
      </Card>
    </div>
  );
};

export default ErrorModal;
```

Adding the Error Modal Component

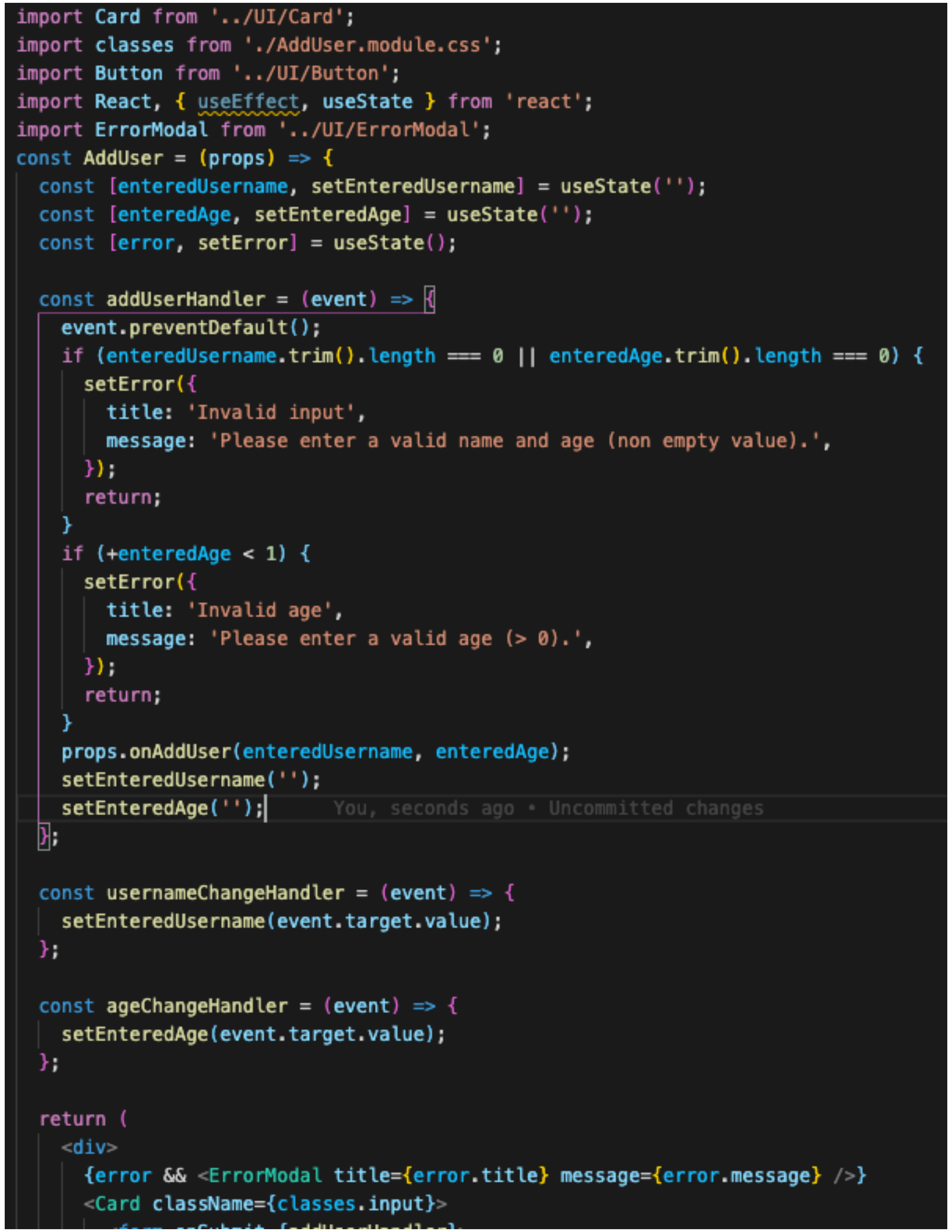


An error occured

something went wrong

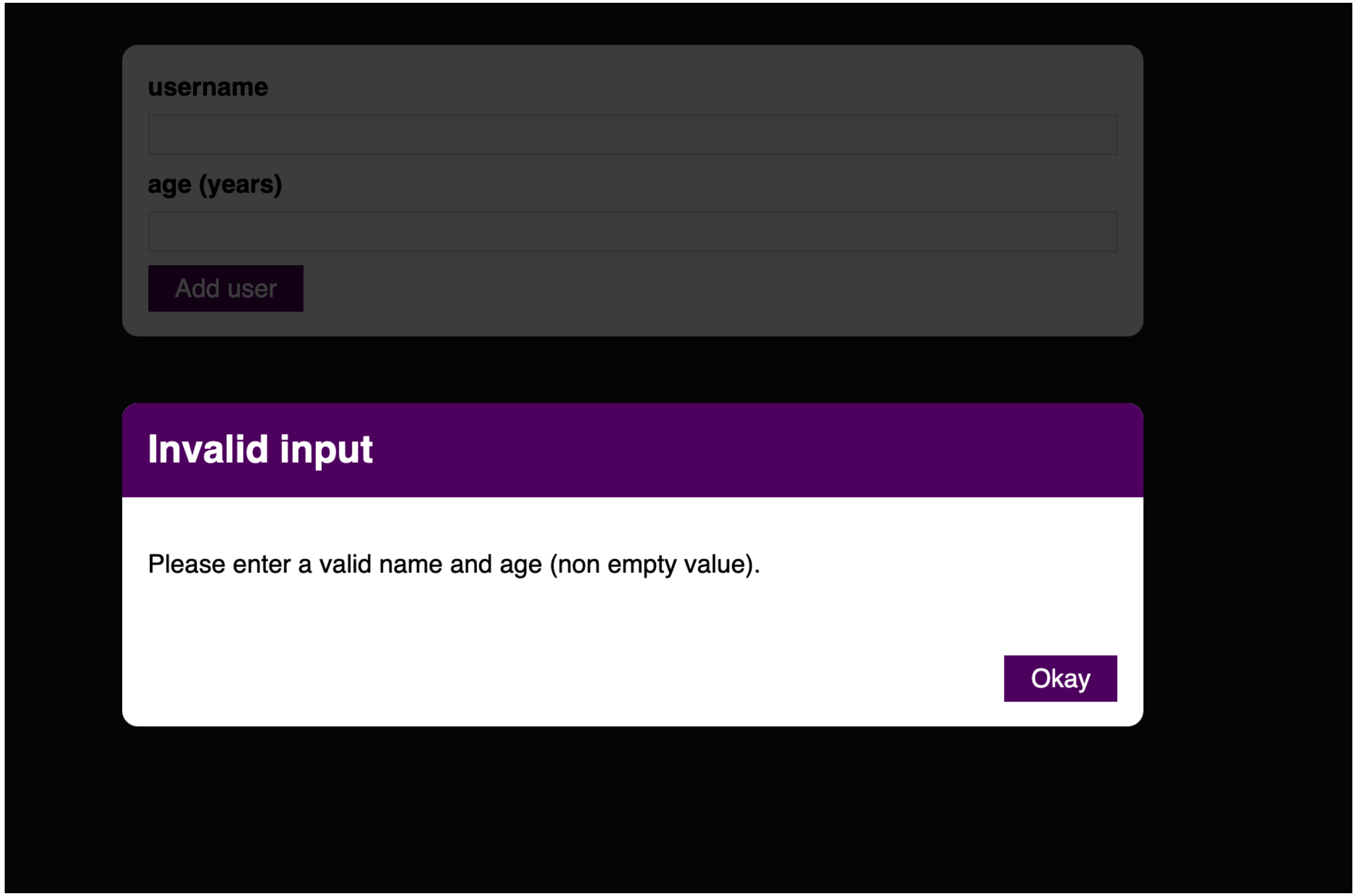
Managing the Error State

Need to add logic to dismiss it,
and also conditionally show the modal.
Update UI to show error state.
Need to manage Error State.
Have an error to show Error Modal
or no error for no Error Modal.



```
import Card from '../UI/Card';
import classes from './AddUser.module.css';
import Button from '../UI/Button';
import React, { useEffect, useState } from 'react';
import ErrorModal from '../UI/ErrorModal';
const AddUser = (props) => {
  const [enteredUsername, setEnteredUsername] = useState('');
  const [enteredAge, setEnteredAge] = useState('');
  const [error, setError] = useState();

  const addUserHandler = (event) => {
    event.preventDefault();
    if (enteredUsername.trim().length === 0 || enteredAge.trim().length === 0) {
      setError({
        title: 'Invalid input',
        message: 'Please enter a valid name and age (non empty value).',
      });
      return;
    }
    if (+enteredAge < 1) {
      setError({
        title: 'Invalid age',
        message: 'Please enter a valid age (> 0).',
      });
      return;
    }
    props.onAddUser(enteredUsername, enteredAge);
    setEnteredUsername('');
    setEnteredAge('');     You, seconds ago • Uncommitted changes
  };

  const usernameChangeHandler = (event) => {
    setEnteredUsername(event.target.value);
  };

  const ageChangeHandler = (event) => {
    setEnteredAge(event.target.value);
  };

  return (
    <div>
      {error && <ErrorModal title={error.title} message={error.message} />}
      <Card className={classes.input}>
```

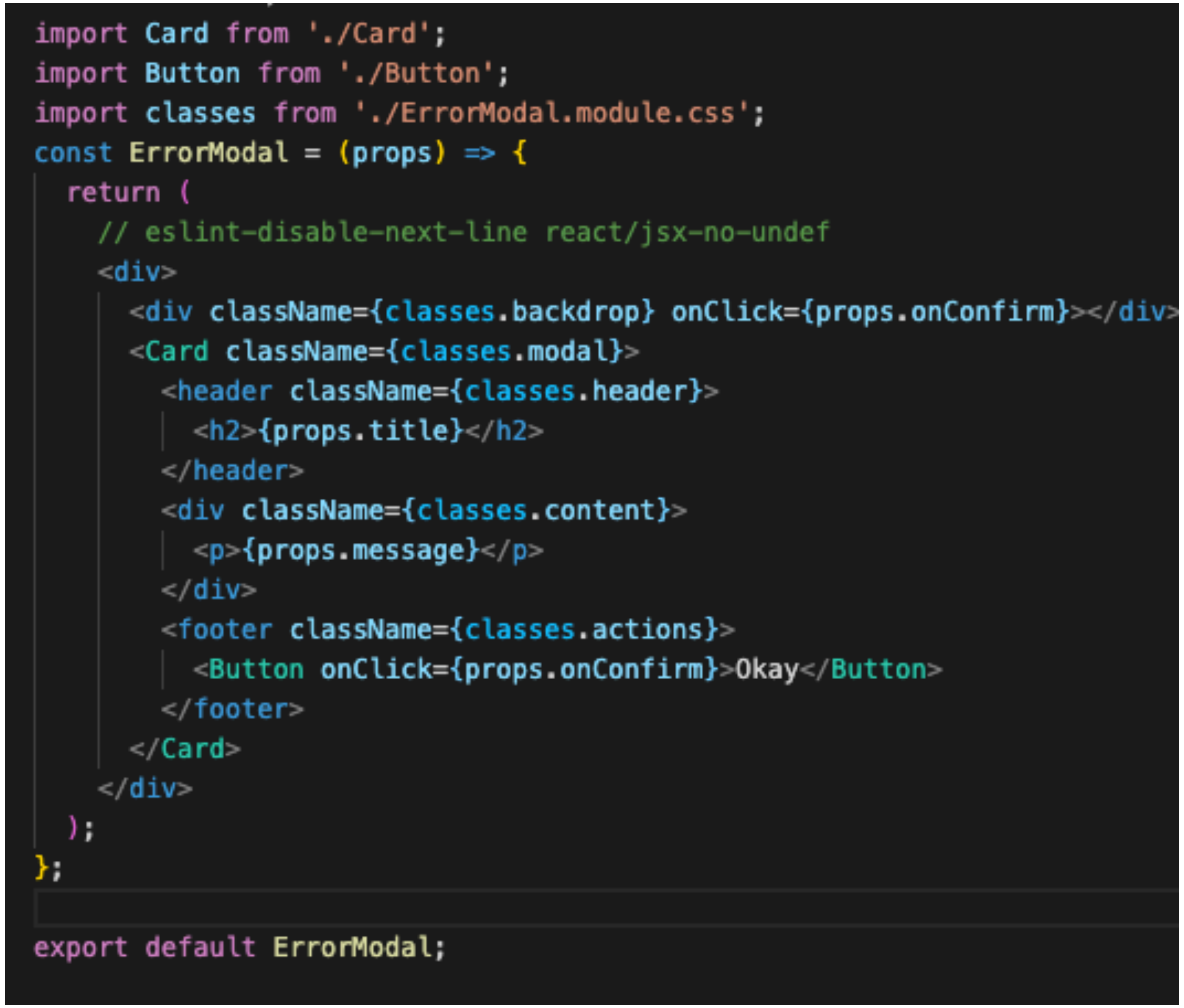
Managing the Error State



Invalid input

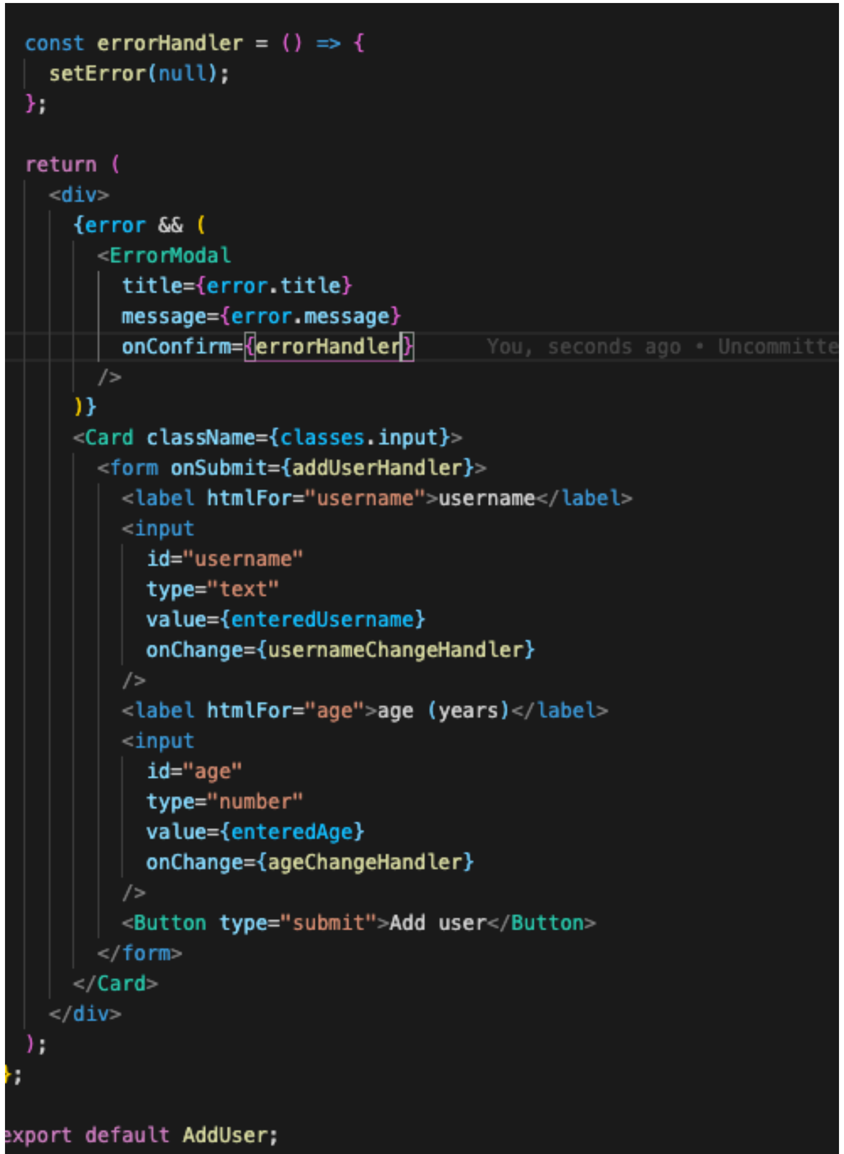
Please enter a valid name and age (non empty value).

Managing the Error State



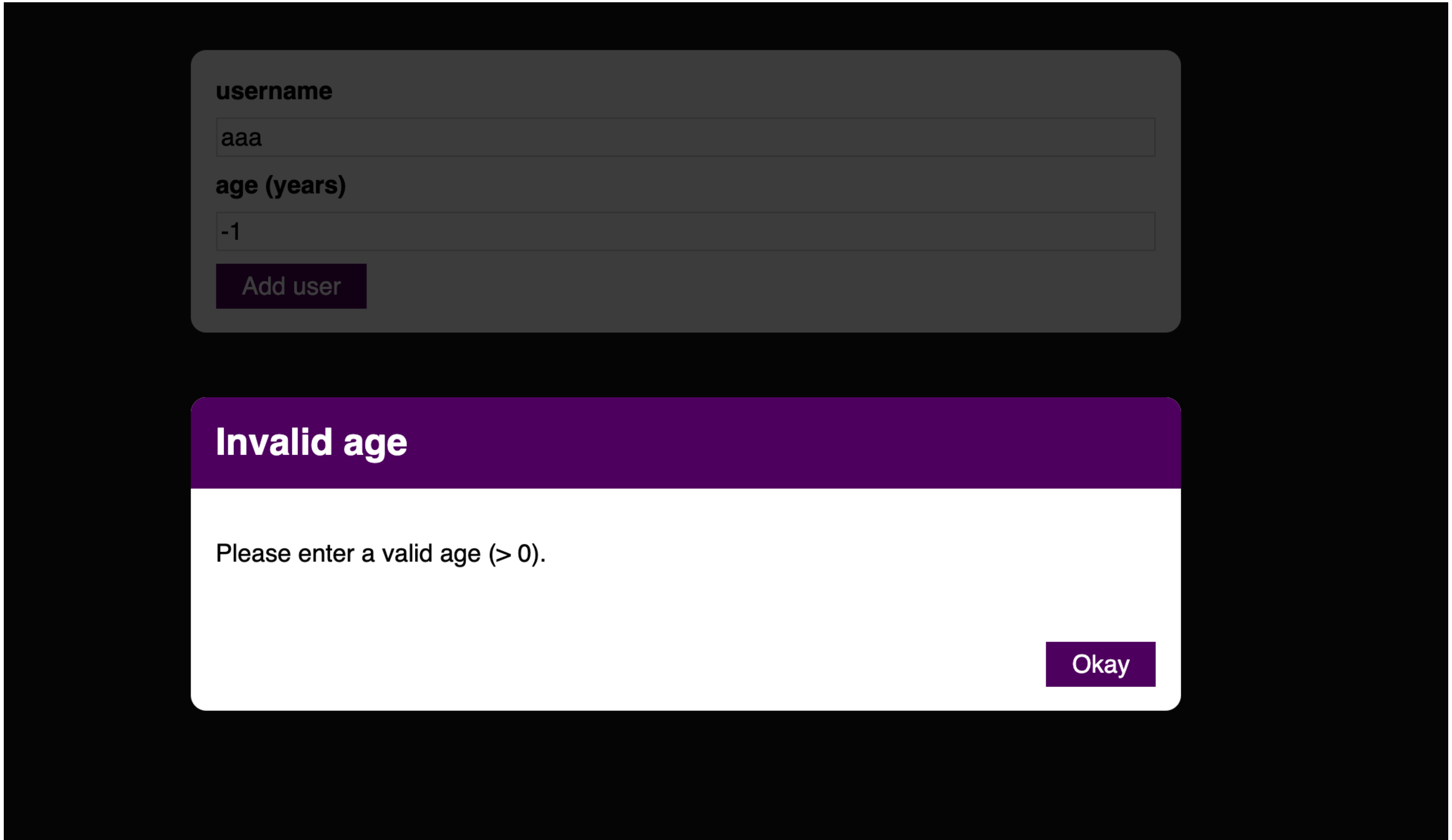
```
import Card from './Card';
import Button from './Button';
import classes from './ErrorModal.module.css';
const ErrorModal = (props) => {
  return (
    // eslint-disable-next-line react/jsx-no-undef
    <div>
      <div className={classes.backdrop} onClick={props.onConfirm}></div>
      <Card className={classes.modal}>
        <header className={classes.header}>
          <h2>{props.title}</h2>
        </header>
        <div className={classes.content}>
          <p>{props.message}</p>
        </div>
        <footer className={classes.actions}>
          <Button onClick={props.onConfirm}>Okay</Button>
        </footer>
      </Card>
    </div>
  );
};

export default ErrorModal;
```

Managing the Error State



```
  const errorHandler = () => {
    setError(null);
  };

  return (
    <div>
      {error && (
        <ErrorModal
          title={error.title}
          message={error.message}
          onConfirm={errorHandler}      You, seconds ago • Uncommitte
        />
      )}
      <Card className={classes.input}>
        <form onSubmit={addUserHandler}>
          <label htmlFor="username">username</label>
          <input
            id="username"
            type="text"
            value={enteredUsername}
            onChange={usernameChangeHandler}
          />
          <label htmlFor="age">age (years)</label>
          <input
            id="age"
            type="number"
            value={enteredAge}
            onChange={ageChangeHandler}
          />
          <Button type="submit">Add user</Button>
        </form>
      </Card>
    </div>
  );
};

xport default AddUser;
```

Managing the Error State



username

aaa

age (years)

-1

Invalid age

Please enter a valid age (> 0).

Demo