HackReact

2016 / 03 / 05

Topics

JavaScript

- JavaScript Basics
    - ECMAScript 6
    - Babel
    - JSX
    - Webpack

React

- React
    - Components
    - Listeners
    - State
    - Props
- Redux
- React Native
- HackReact

Hack

- Teams
- React or React Native
- Goals

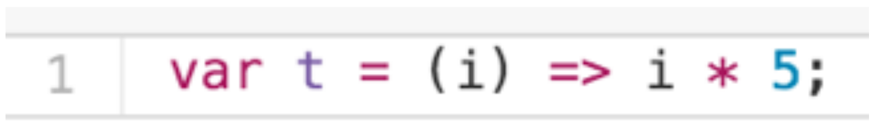
ECMAScript 6 (ES6)

- Also known as ECMAScript 2015
- Successor of ECMAScript 5 (since 2009)
- Standardized scripting language (ISO certified)
- JavaScript is an implementation of ECMAScript
- Compiler is required to run ES6 code in browsers or engines (2016/03)

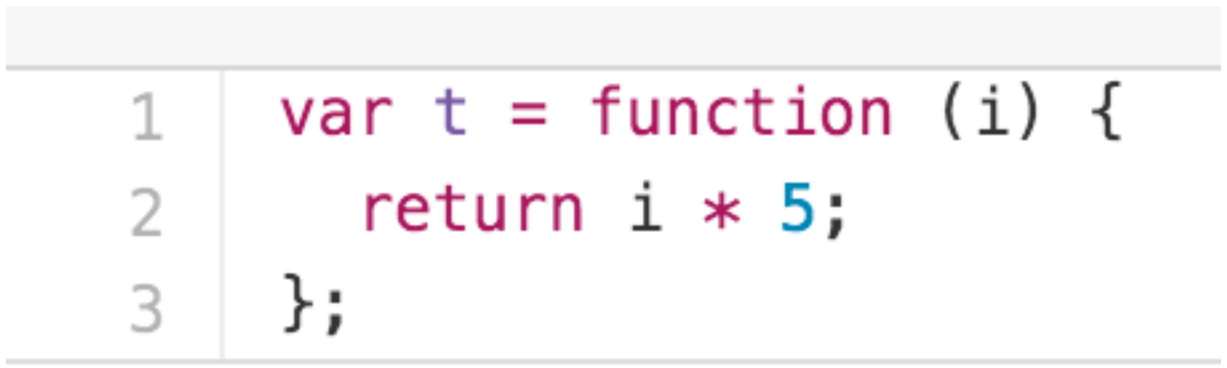
- Programming React makes more fun with ES 6

Babel

- JavaScript compiler
- Ability to use ES6 now without waiting for browser or engine support
- Plugin-based



```
var t = (i) => i * 5;
```

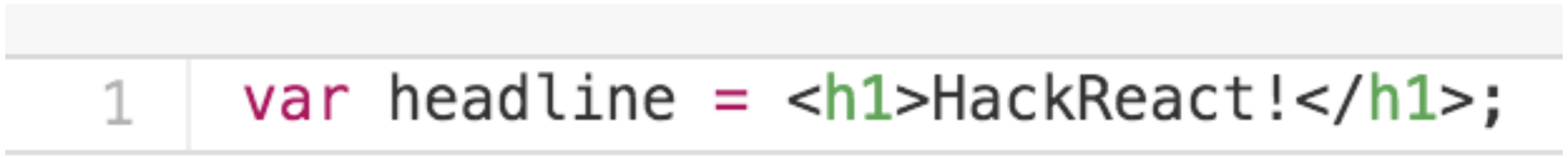


```
var t = function (i) {
  return i * 5;
};
```

JSX

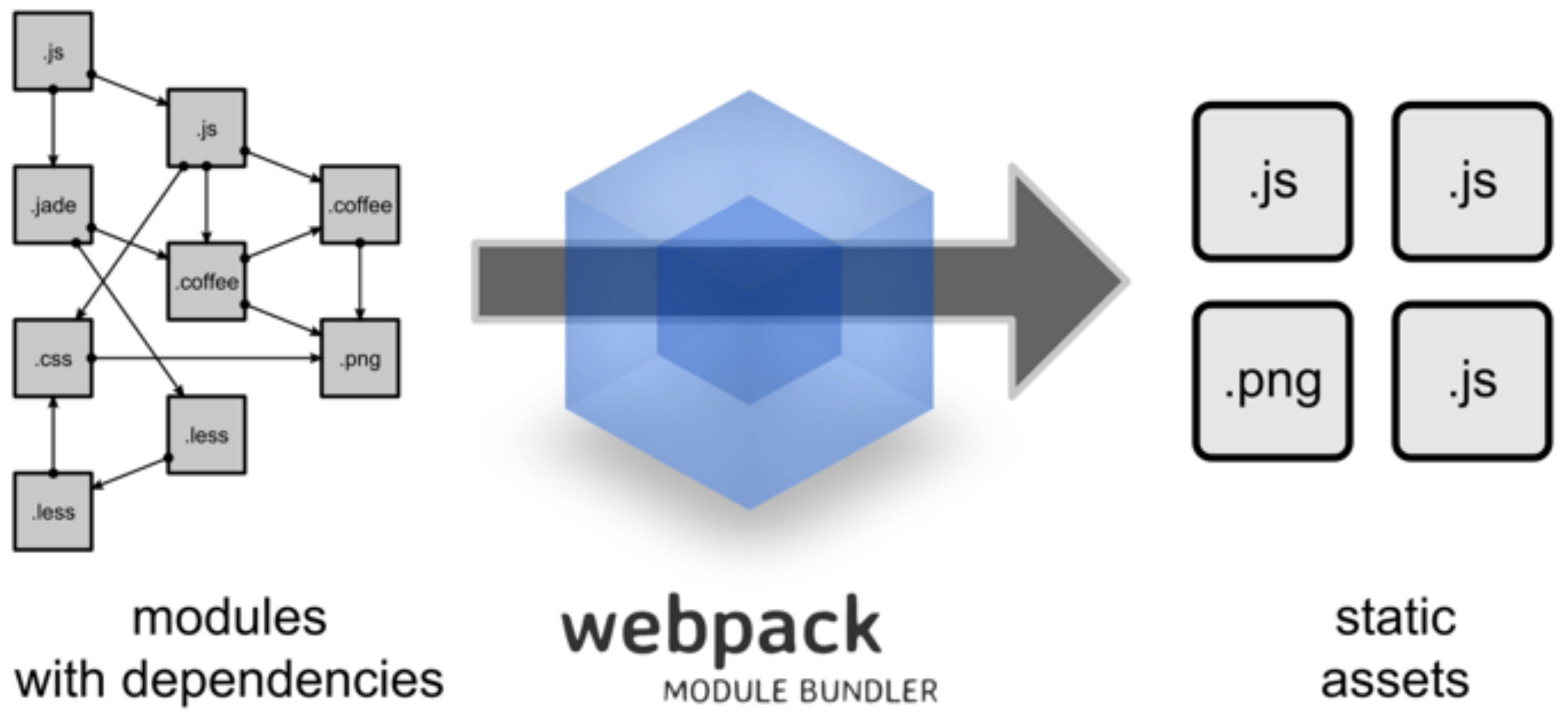
~~https://jsx.github.io/~~

https://facebook.github.io/jsx/ 👍



```
var headline = <h1>HackReact!</h1>;
```

Webpack



Webpack compared: http://survivejs.com/webpack_react/webpack_compared/

Image: https://webpack.github.io/

React

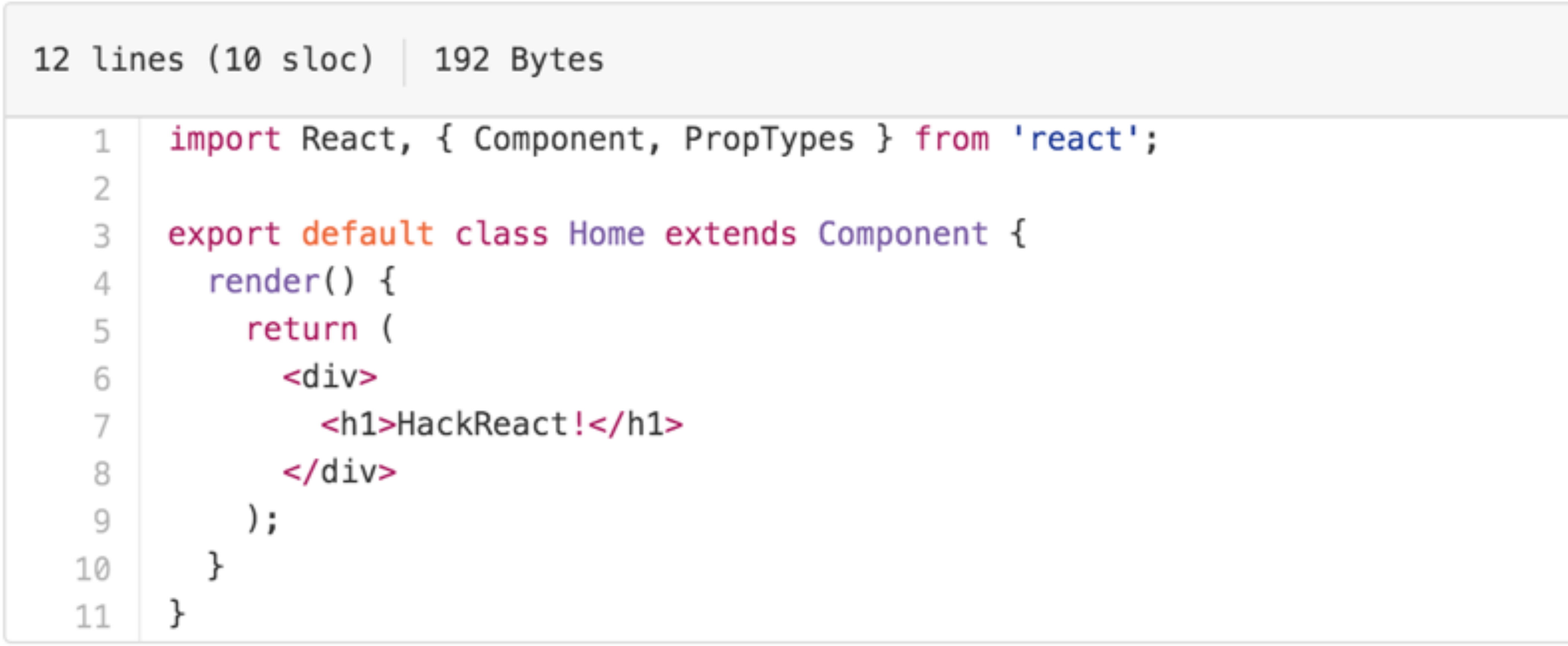
The following code snippets could be written more efficient and beautiful

They are just kept easy for presentation purposes

Image: http://images.mysafetylabels.com/img/lg/L/bold-caution-label-lb-2305.png

React

- React is a library, not a framework
- What React does: Render UI components

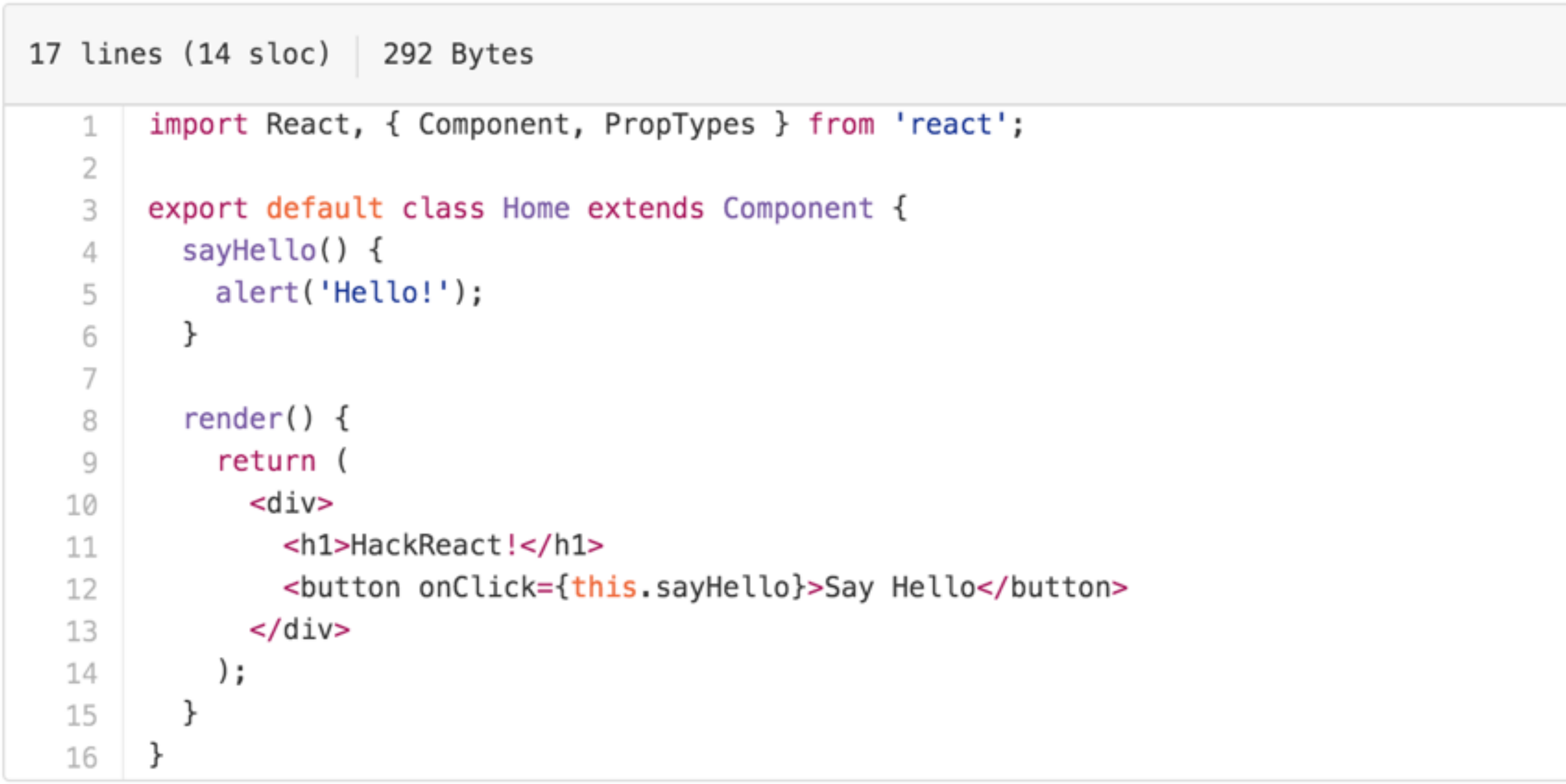


```
12 lines (10 sloc) | 192 Bytes
```

```
import React, { Component, PropTypes } from 'react';

export default class Home extends Component {
  render() {
    return (
      <div>
        <h1>HackReact!</h1>
      </div>
    );
  }
}
```

ELBSTACK codecentric

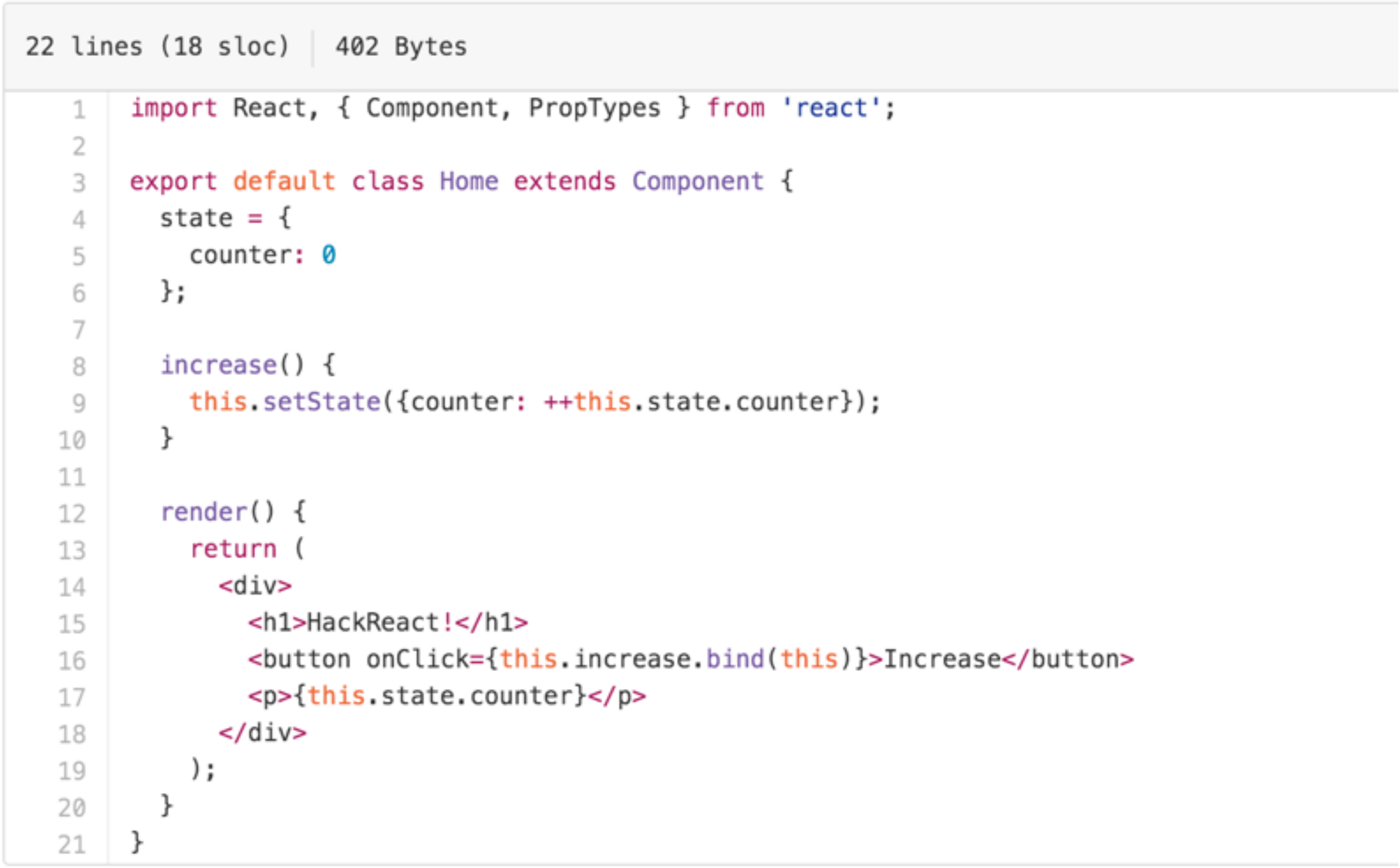
React - Listeners



17 lines (14 sloc) | 292 Bytes

```
import React, { Component, PropTypes } from 'react';

export default class Home extends Component {
  sayHello() {
    alert('Hello!');
  }

  render() {
    return (
      <div>
        <h1>HackReact!</h1>
        <button onClick={this.sayHello}>Say Hello</button>
      </div>
    );
  }
}
```

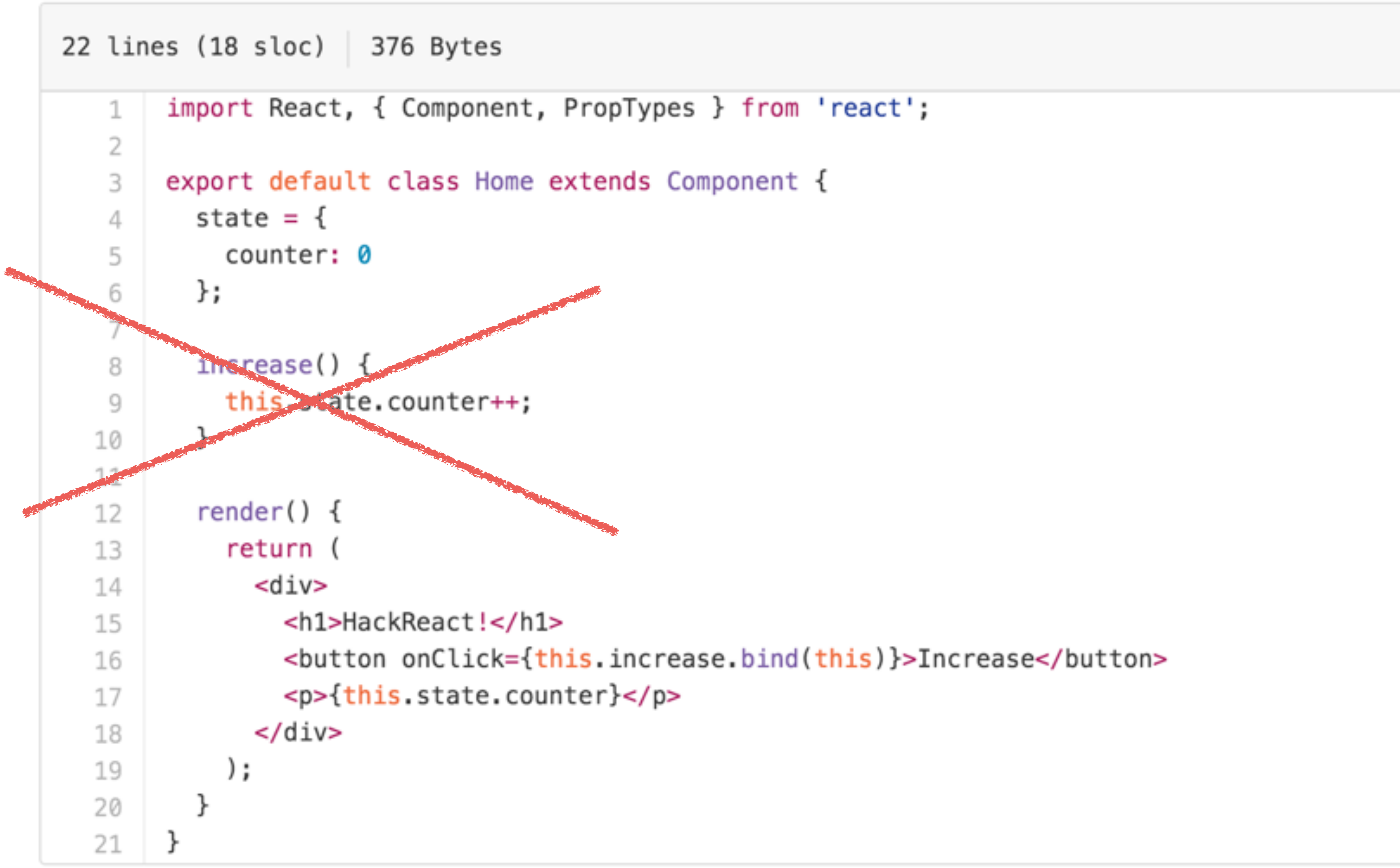
React - State



22 lines (18 sloc) | 402 Bytes

```
import React, { Component, PropTypes } from 'react';

export default class Home extends Component {
  state = {
    counter: 0
  };

  increase() {
    this.setState({counter: ++this.state.counter});
  }

  render() {
    return (
      <div>
        <h1>HackReact!</h1>
        <button onClick={this.increase.bind(this)}>Increase</button>
        <p>{this.state.counter}</p>
      </div>
    );
  }
}
```

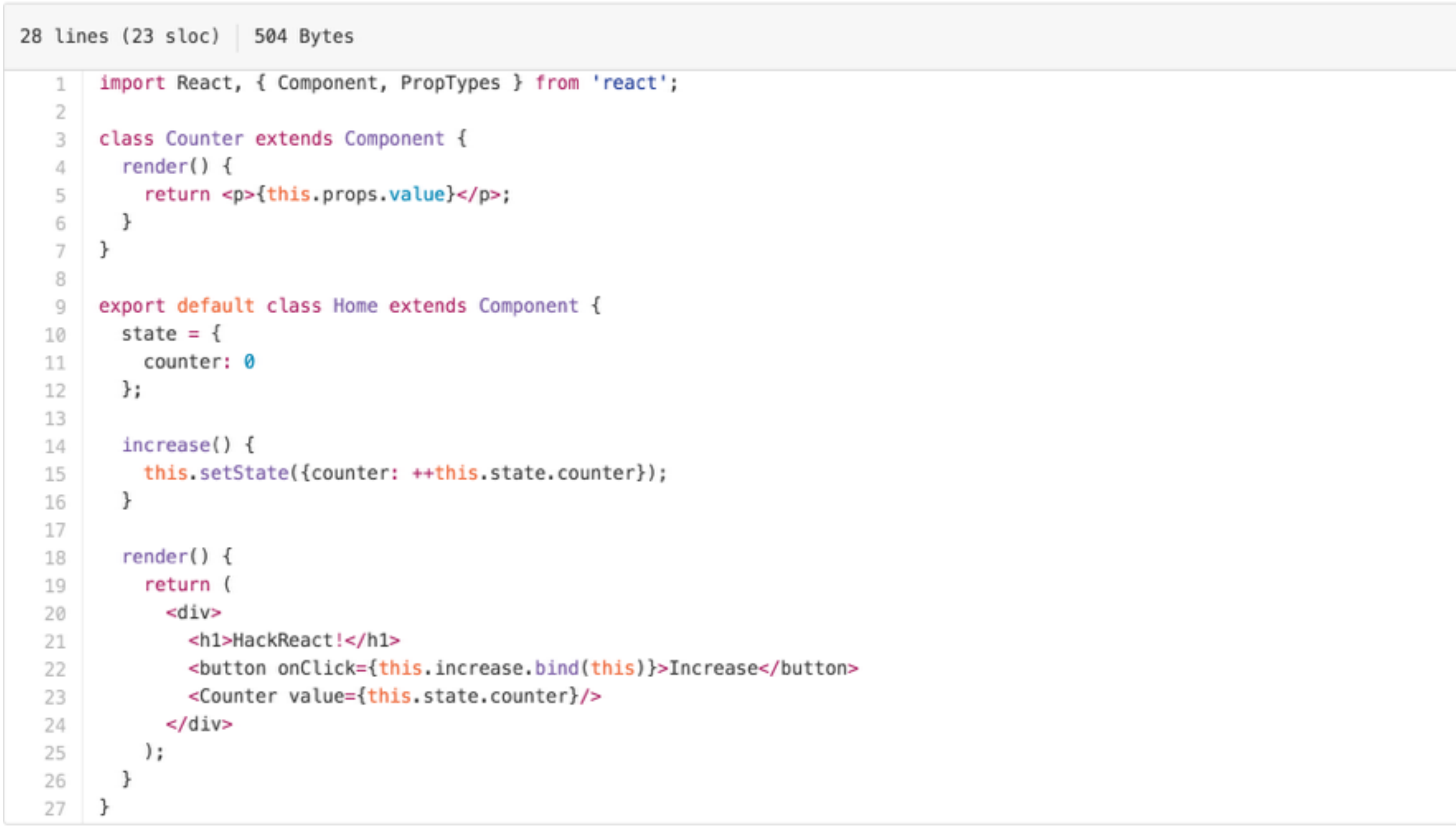
React - State (DON'T DO THIS)



22 lines (18 sloc) | 376 Bytes

```
import React, { Component, PropTypes } from 'react';

export default class Home extends Component {
  state = {
    counter: 0
  };

  increase() {
    this.state.counter++;
  }

  render() {
    return (
      <div>
        <h1>HackReact!</h1>
        <button onClick={this.increase.bind(this)}>Increase</button>
        <p>{this.state.counter}</p>
      </div>
    );
  }
}
```

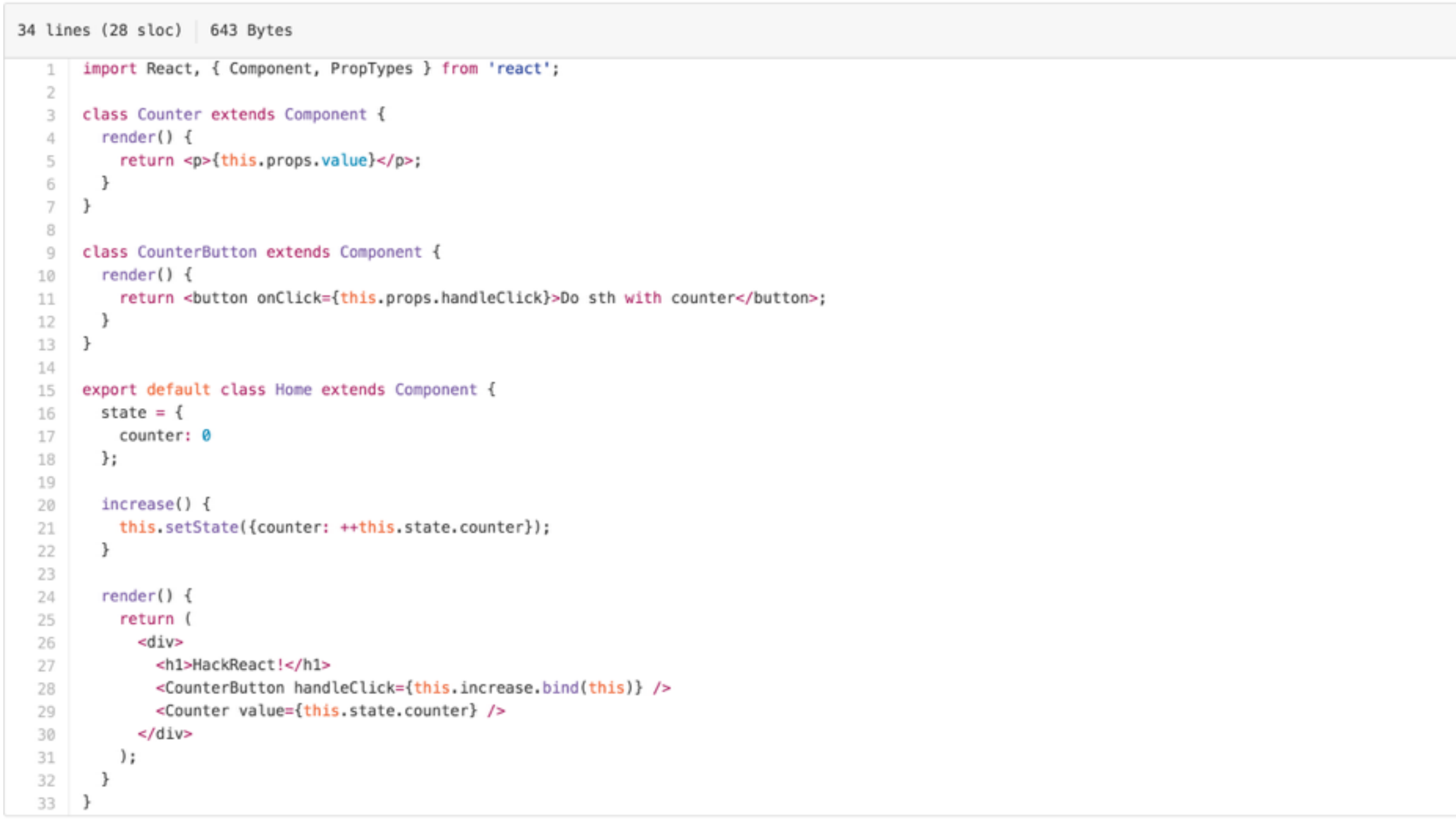
React - Props



28 lines (23 sloc) | 504 Bytes

```
import React, { Component, PropTypes } from 'react';

class Counter extends Component {
  render() {
    return <p>{this.props.value}</p>;
  }
}

export default class Home extends Component {
  state = {
    counter: 0
  };

  increase() {
    this.setState({counter: ++this.state.counter});
  }

  render() {
    return (
      <div>
        <h1>HackReact!</h1>
        <button onClick={this.increase.bind(this)}>Increase</button>
        <Counter value={this.state.counter}/>
      </div>
    );
  }
}
```

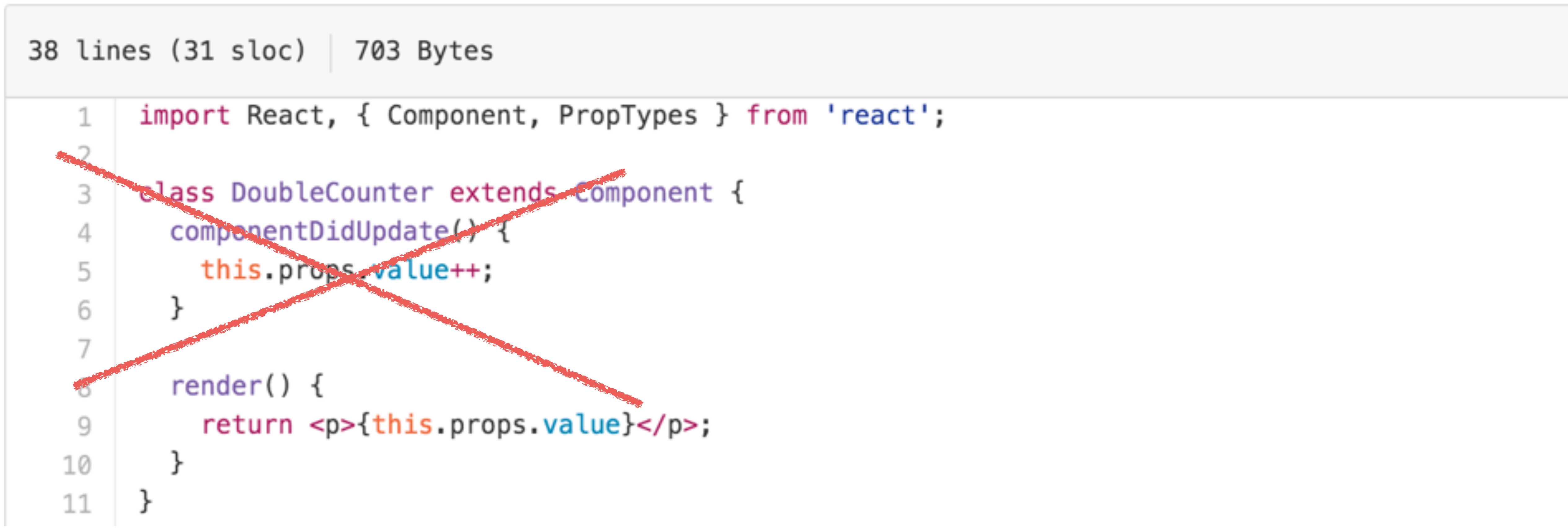
React - Function props



34 lines (28 sloc) | 643 Bytes

```
import React, { Component, PropTypes } from 'react';

class Counter extends Component {
  render() {
    return <p>{this.props.value}</p>;
  }
}

class CounterButton extends Component {
  render() {
    return <button onClick={this.props.handleClick}>Do sth with counter</button>;
  }
}

export default class Home extends Component {
  state = {
    counter: 0
  };

  increase() {
    this.setState({counter: ++this.state.counter});
  }

  render() {
    return (
      <div>
        <h1>HackReact!</h1>
        <CounterButton handleClick={this.increase.bind(this)} />
        <Counter value={this.state.counter} />
      </div>
    );
  }
}
```

React - Props (DON'T DO THIS)



38 lines (31 sloc) | 703 Bytes

```
import React, { Component, PropTypes } from 'react';

class DoubleCounter extends Component {
  componentDidUpdate() {
    this.props.value++;
  }

  render() {
    return <p>{this.props.value}</p>;
  }
}
```

Redux

- Predictable state container for JavaScript apps
- NOT REACT SPECIFIC
- NOT THE REACT COMPONENT STATE
- Can be used with different JavaScript libraries or frameworks
  - e.g. react-redux

Redux - Terms

Store

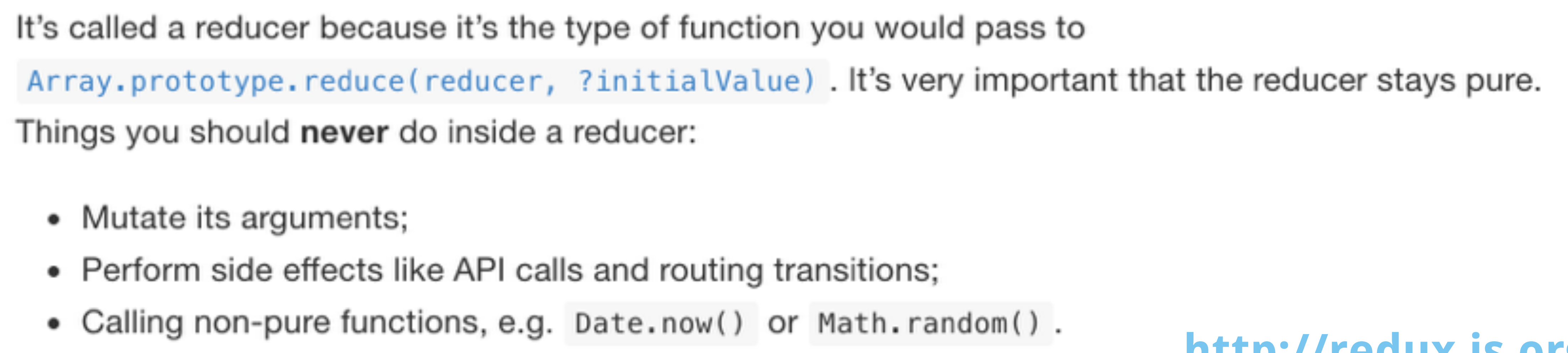
- Holds the state
- Enables us to dispatch actions on it

Actions

- Payloads of information
- Describe the fact that something happened

Reducers

- Specify how the state should change



It's called a reducer because it's the type of function you would pass to `Array.prototype.reduce(reducer, ?initialValue)`. It's very important that the reducer stays pure. Things you should **never** do inside a reducer:

- Mutate its arguments;
- Perform side effects like API calls and routing transitions;
- Calling non-pure functions, e.g. `Date.now()` or `Math.random()`.

http://redux.js.org/

React + Redux - When to use?

- Communication between components (not parent <-> child relation)
- Keep API communication away from components
- Keep complex business logic away from components
- Non-UI logic will be reusable and is not coupled to React
- Can be re-used in a React Native project

Any code examples?

Live!

💪

React Native

- Also developed by Facebook
- Framework on top of React library
- Develop Native apps for iOS and Android
- React components are mapped to native UI components
- JavaScript business logic runs in a separate thread
- „Native like performance“
- No comparison to Ionic or other hybrid app frameworks

The Hackathon

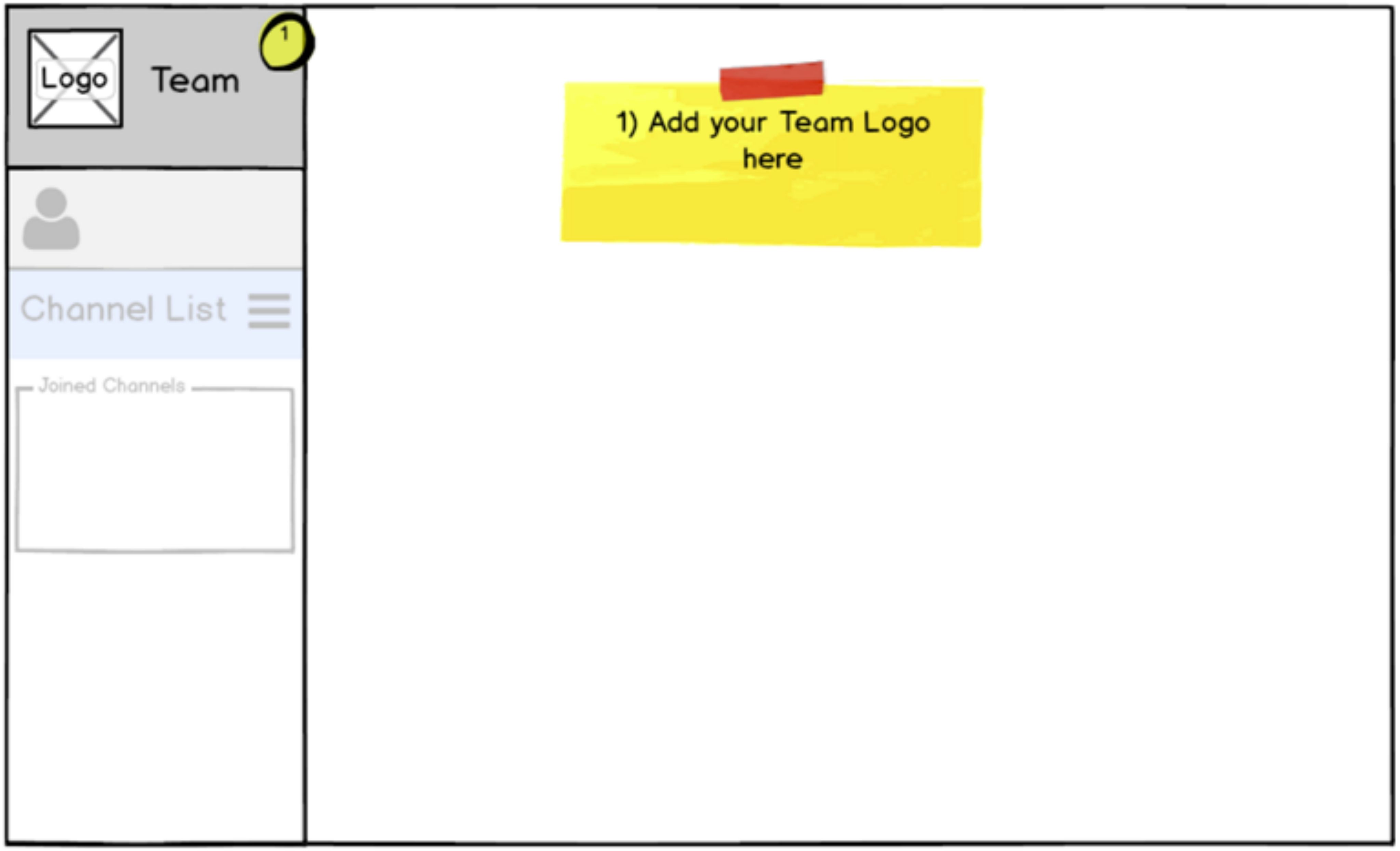
- Lets develop a messenger!
  - Using Sendbird as a third party service
- Decide if you want to hack React for the web or React Native
- Goal for mobile:
  - Have a runnable messenger app for iOS or/and Android
- Goal for web:
  - Push to your git repository
  - Codeship will deploy your project to Heroku

The Hackathon (2)

- Build web or native teams of 3-4 people
- Find your React „expert“
- For web checkout: https://github.com/elbstack/elbstack-hackreact-web-<team-number>.git
- For native checkout: https://github.com/elbstack/elbstack-hackreact-native/
- Type as fast as you can

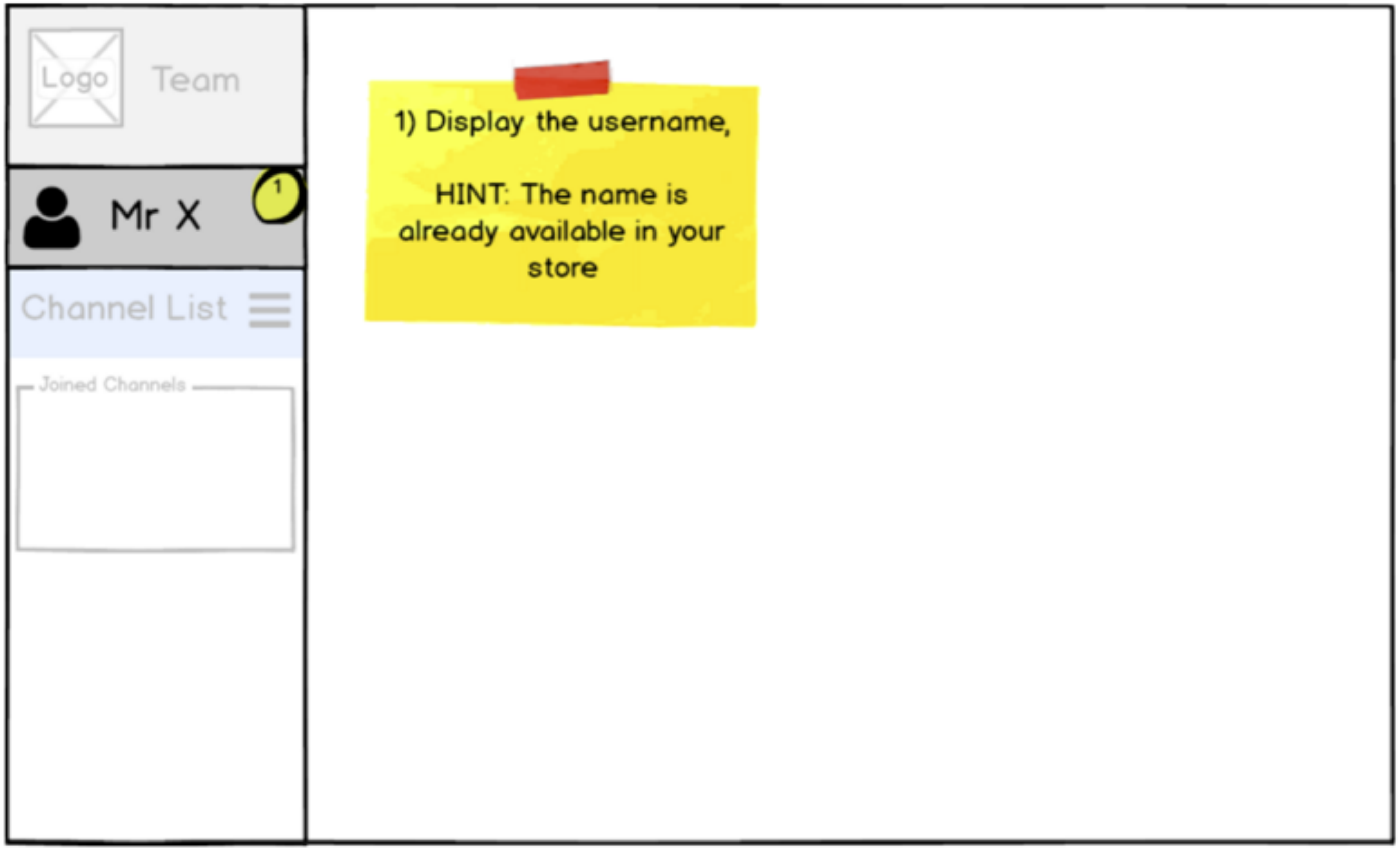
Exercise 1

Embed your team name + logo



Exercise 2

Display username

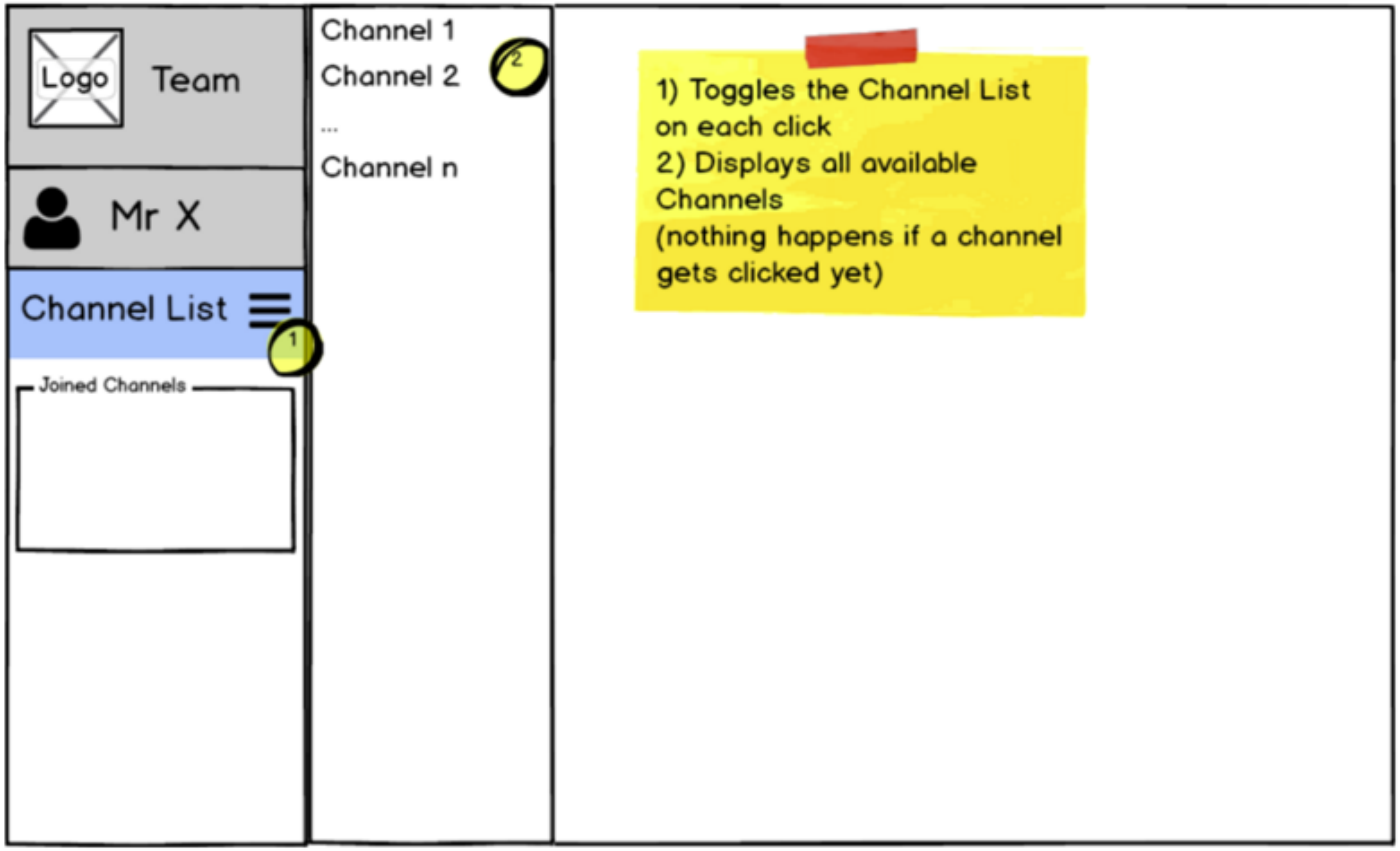


1) Display the username,

HINT: The name is already available in your store

Exercise 3

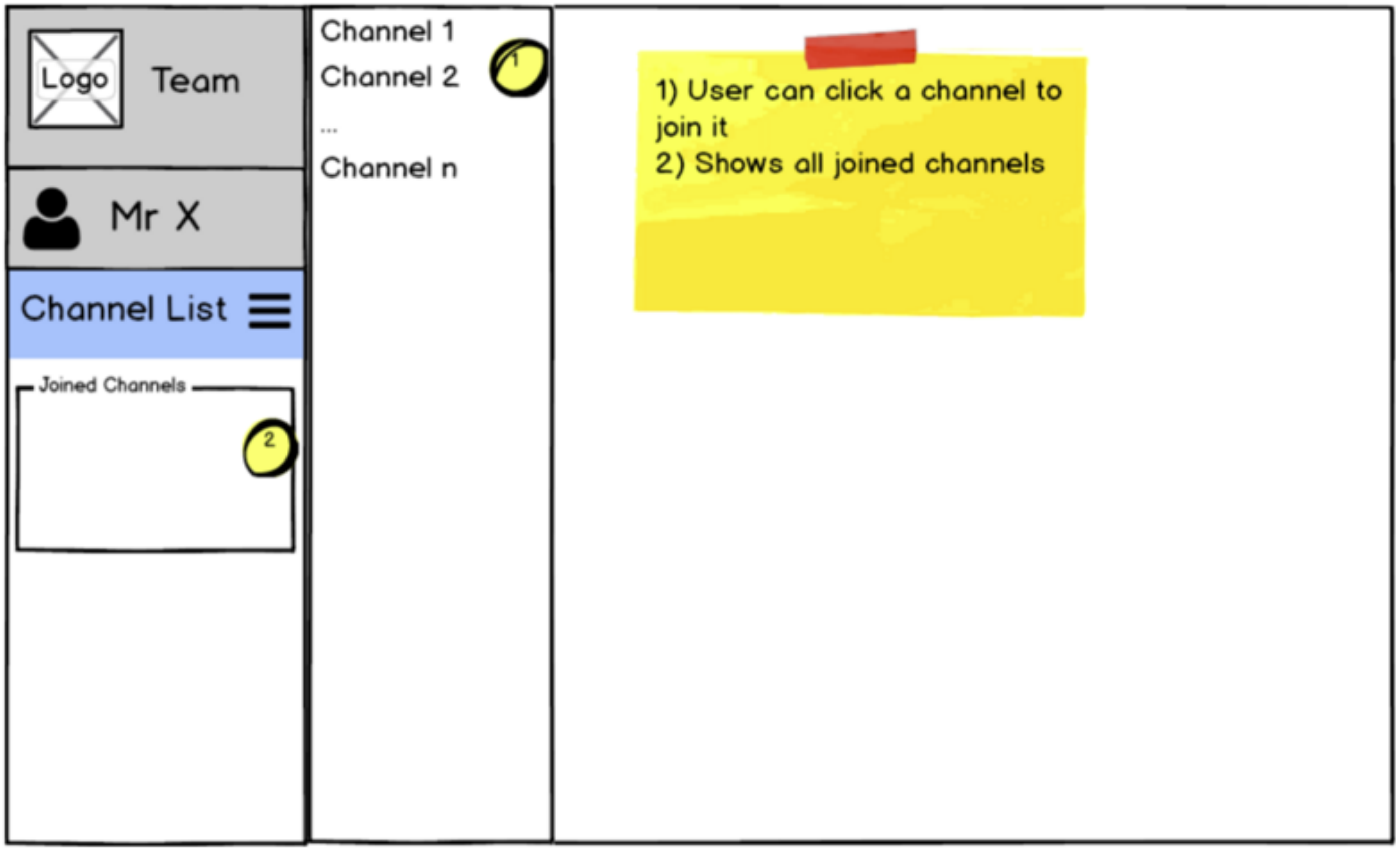
Display all available chat channels



1) Toggles the Channel List on each click
2) Displays all available Channels
(nothing happens if a channel gets clicked yet)

Exercise 4

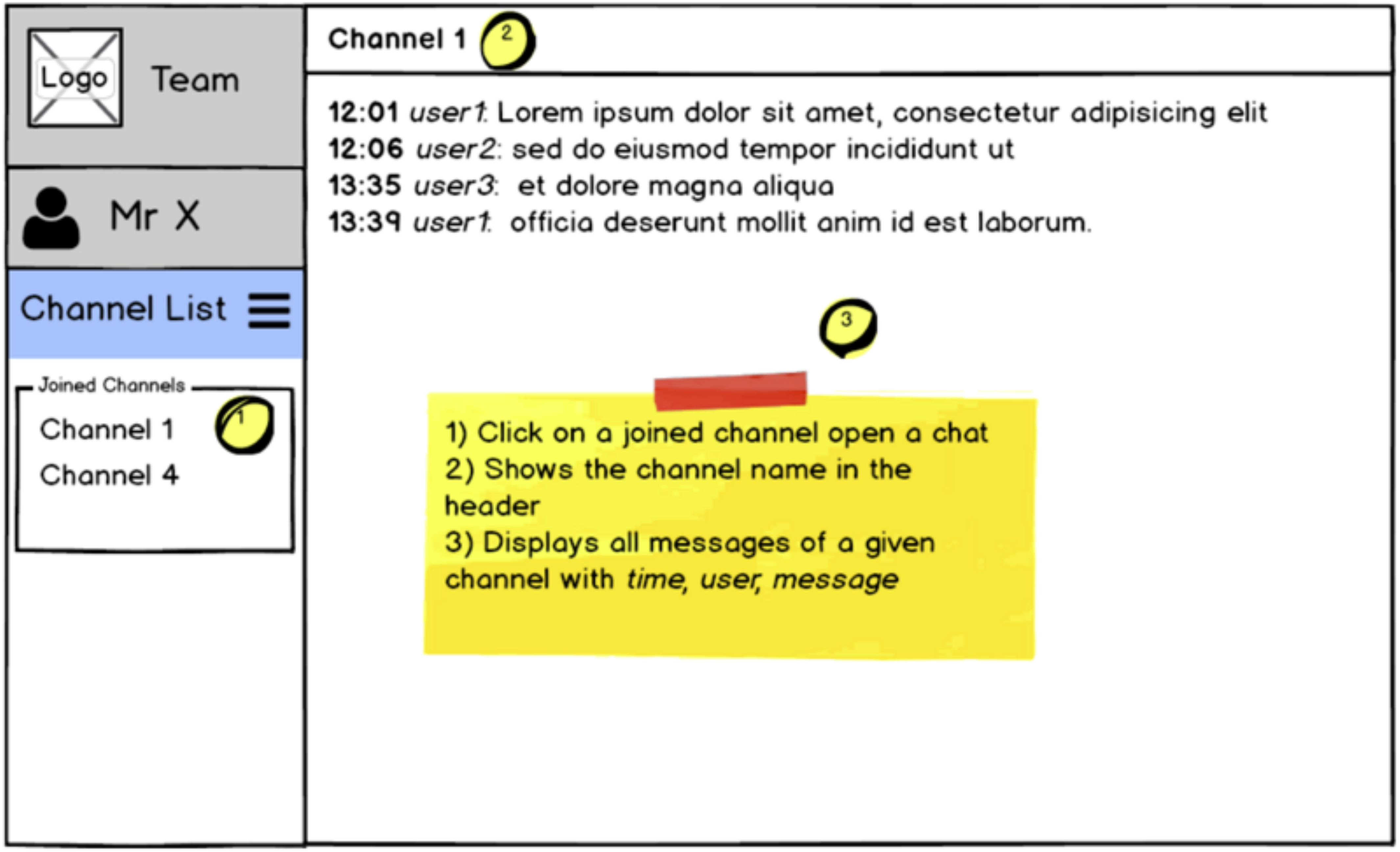
Join channel



1) User can click a channel to join it
2) Shows all joined channels

Exercise 5

Display channel content



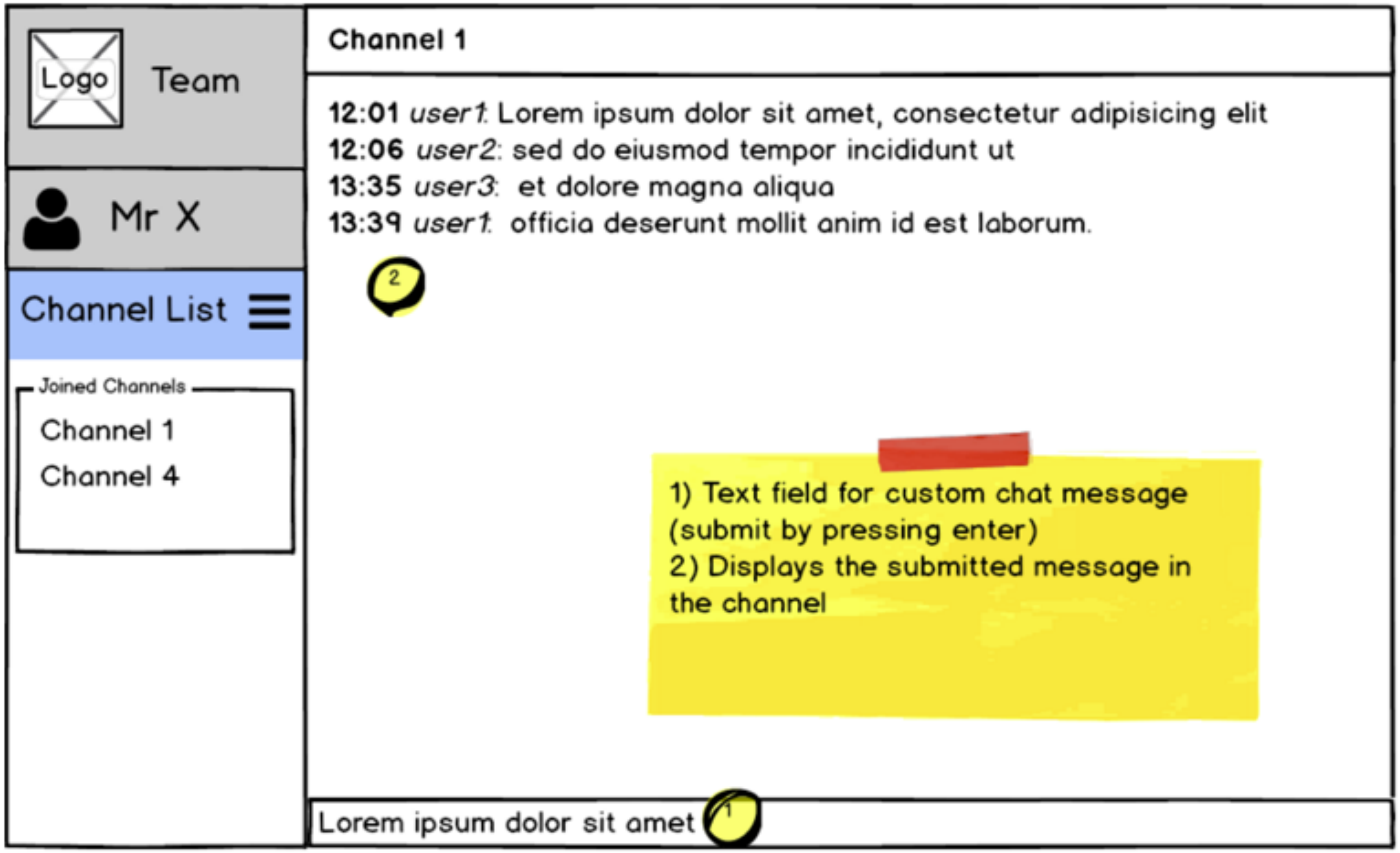
Channel 1

12:01 *user1*: Lorem ipsum dolor sit amet, consectetur adipisicing elit
12:06 *user2*: sed do eiusmod tempor incididunt ut
13:35 *user3*: et dolore magna aliqua
13:39 *user1*: officia deserunt mollit anim id est laborum.

1) Click on a joined channel open a chat
2) Shows the channel name in the header
3) Displays all messages of a given channel with *time, user, message*

Exercise 6

Send a message



12:01 *user1*: Lorem ipsum dolor sit amet, consectetur adipisicing elit
12:06 *user2*: sed do eiusmod tempor incididunt ut
13:35 *user3*: et dolore magna aliqua
13:39 *user1*: officia deserunt mollit anim id est laborum.

1) Text field for custom chat message (submit by pressing enter)
2) Displays the submitted message in the channel

Lorem ipsum dolor sit amet

Exercise 7

Live update of channel messages

Exercise 8

Choose from

- Typing indicator
- Edit / Delete messages
- Images
- URL-Preview
- Emojis

ELBSTACK

elbstack.com

codecentric.de

codecentric