Video 3.3

React Events

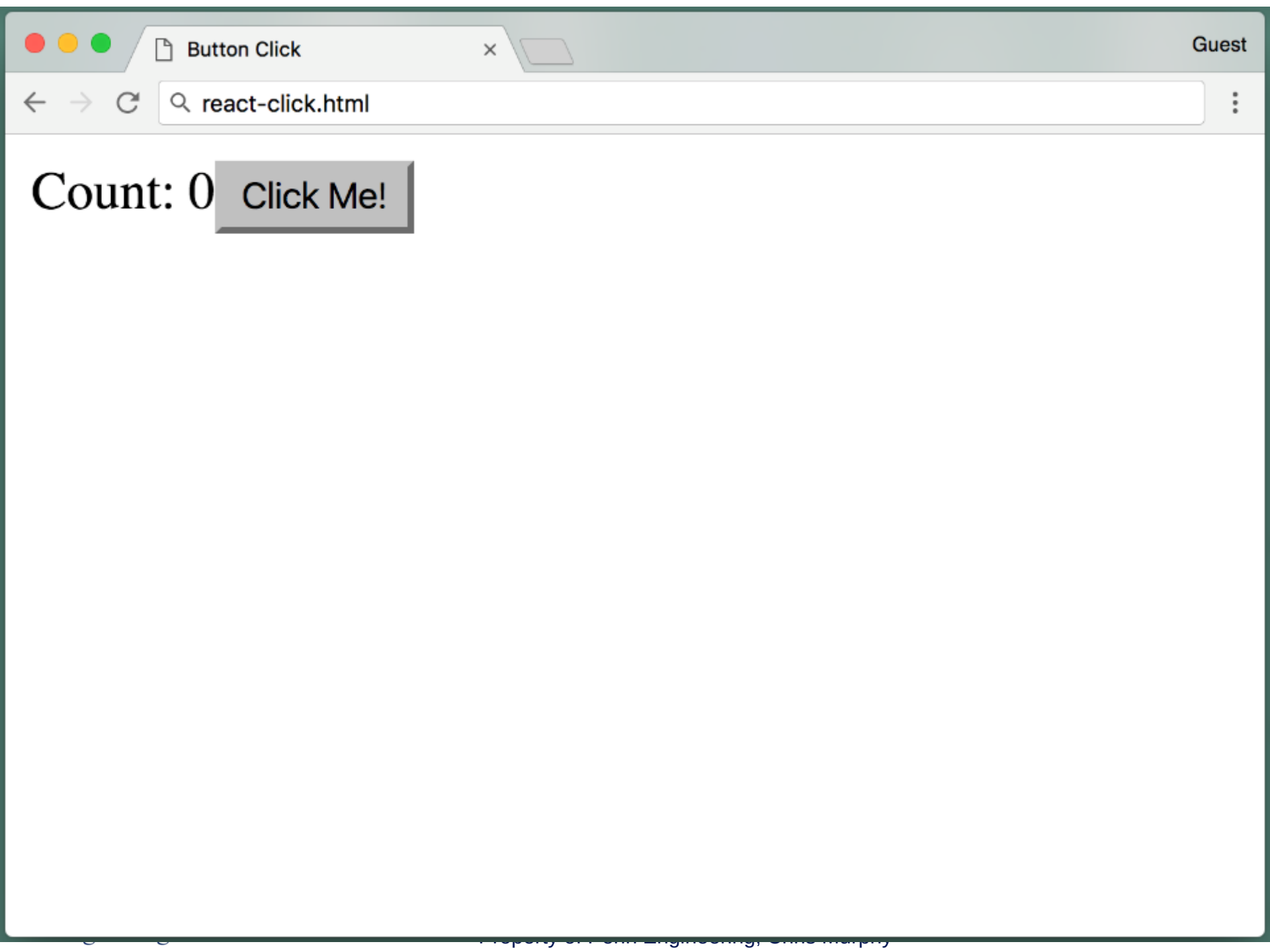
Chris Murphy

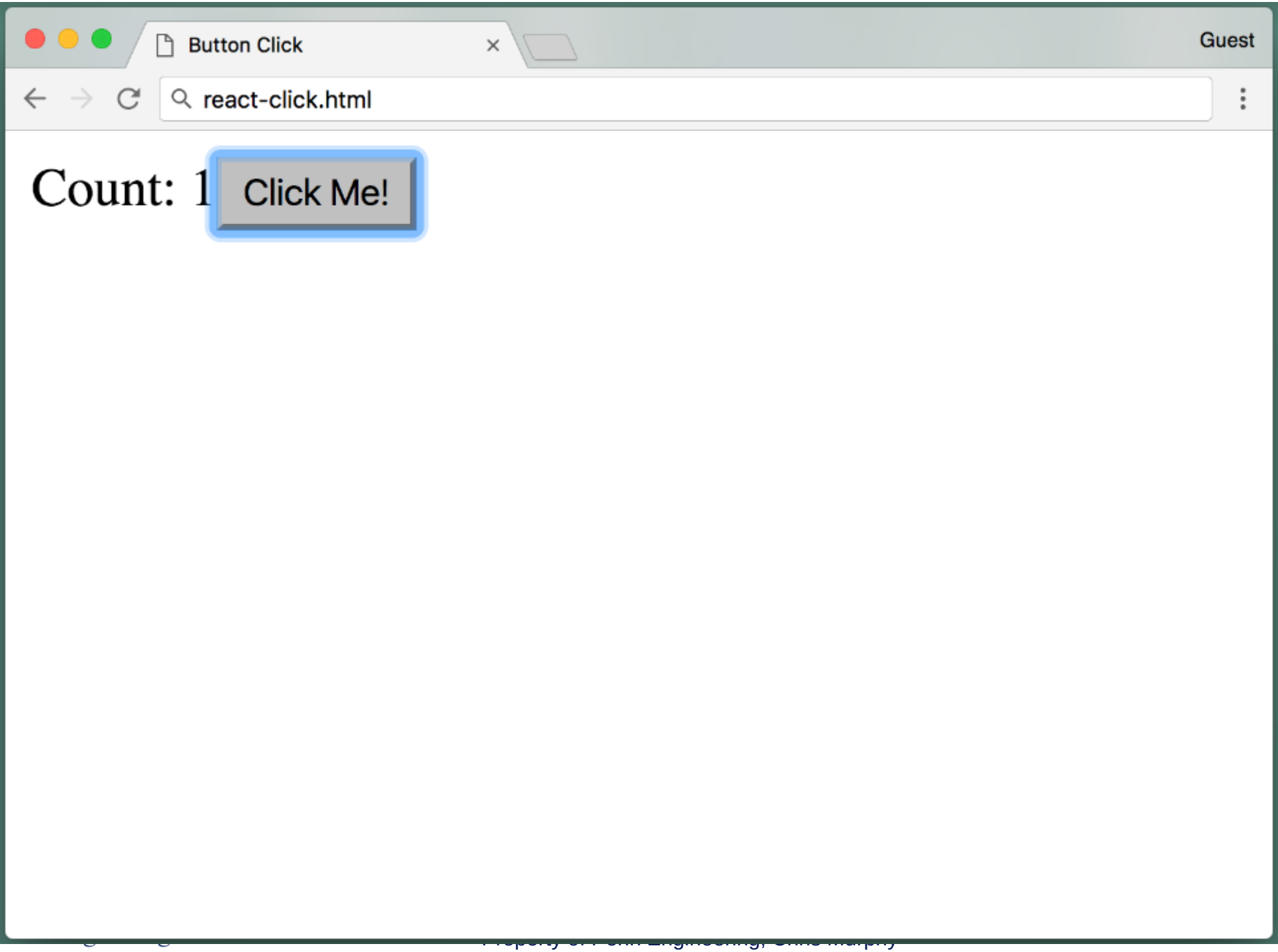
Review

- React allows us to insert JavaScript elements/components into VirtualDOM
- We can create additional components using the `React.Component` class as a base
- Components have two types of attributes
  - **Properties:** set at initialization and immutable thereafter
  - **State:** change in response to user events
- Component callback functions can be bound to HTML events

Changing Component State

- A component's state typically changes in response to some user action or "event"
- We can **bind** an event to a callback function within a React component
- That component can then change state using its `setState` function
- This will automatically re-render the component and any other affected component


Button Click
Guest
react-click.html
Count: 0
Click Me!


Button Click
Guest
react-click.html
Count: 1
Click Me!

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

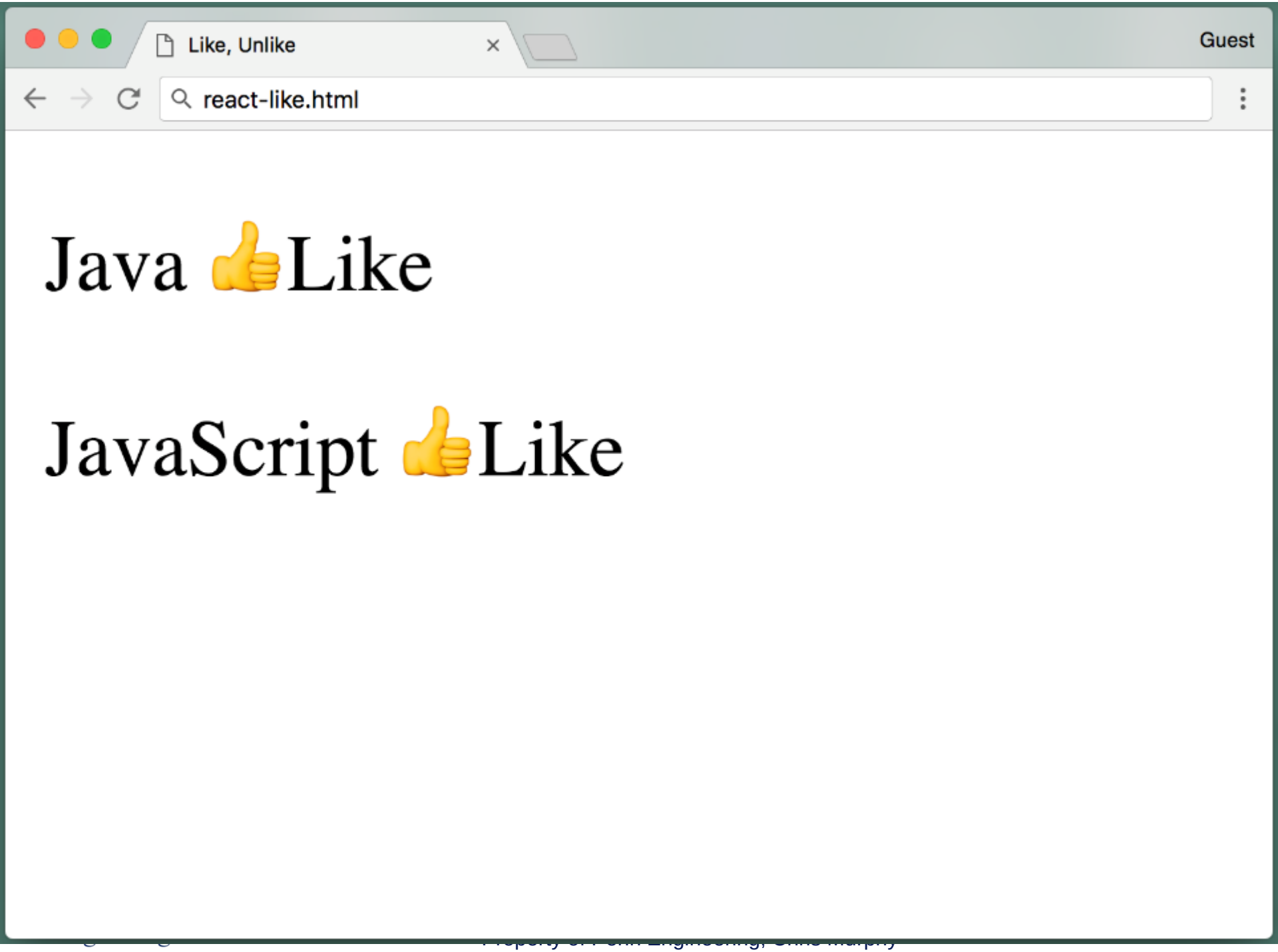
```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

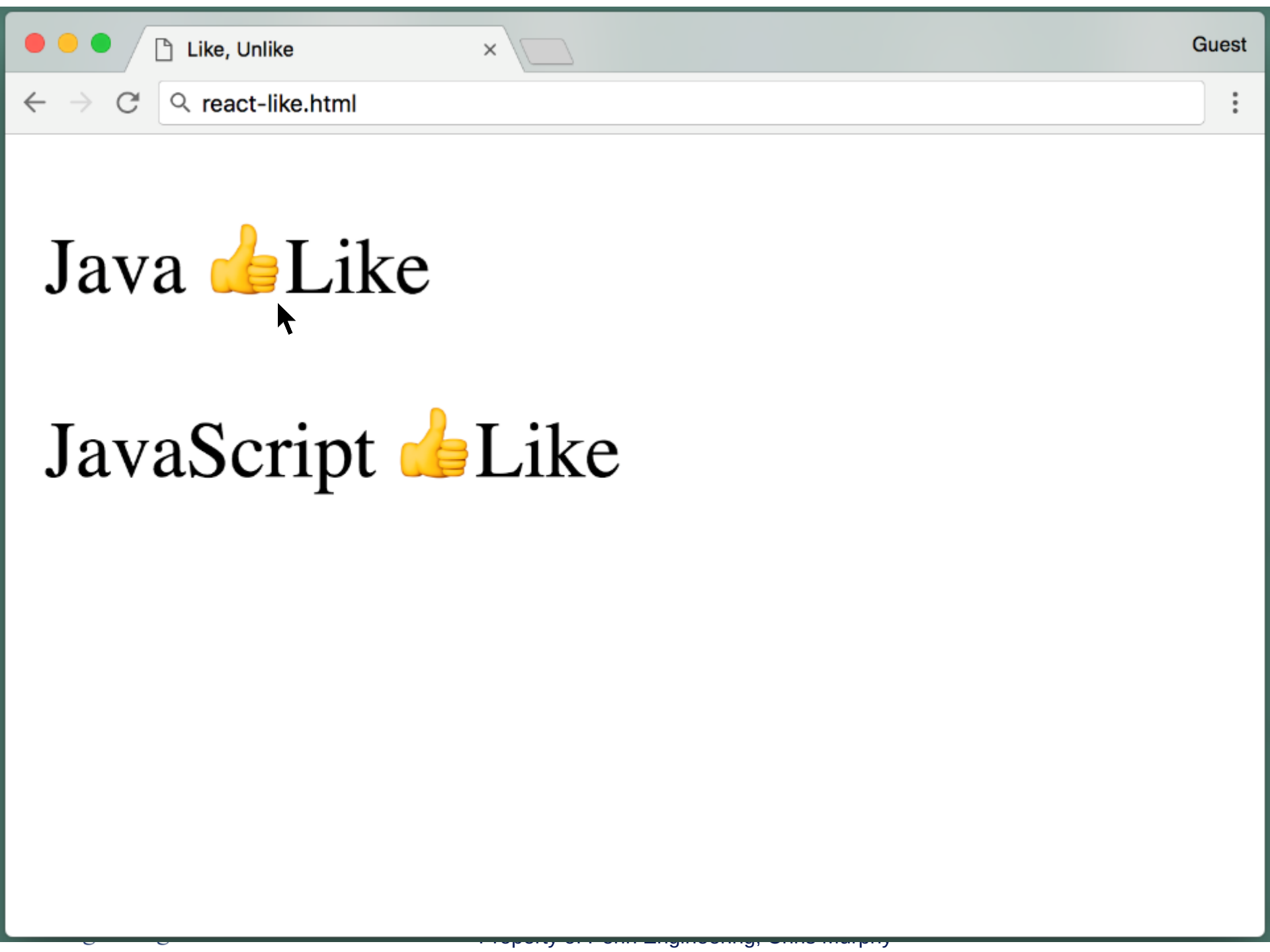
```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

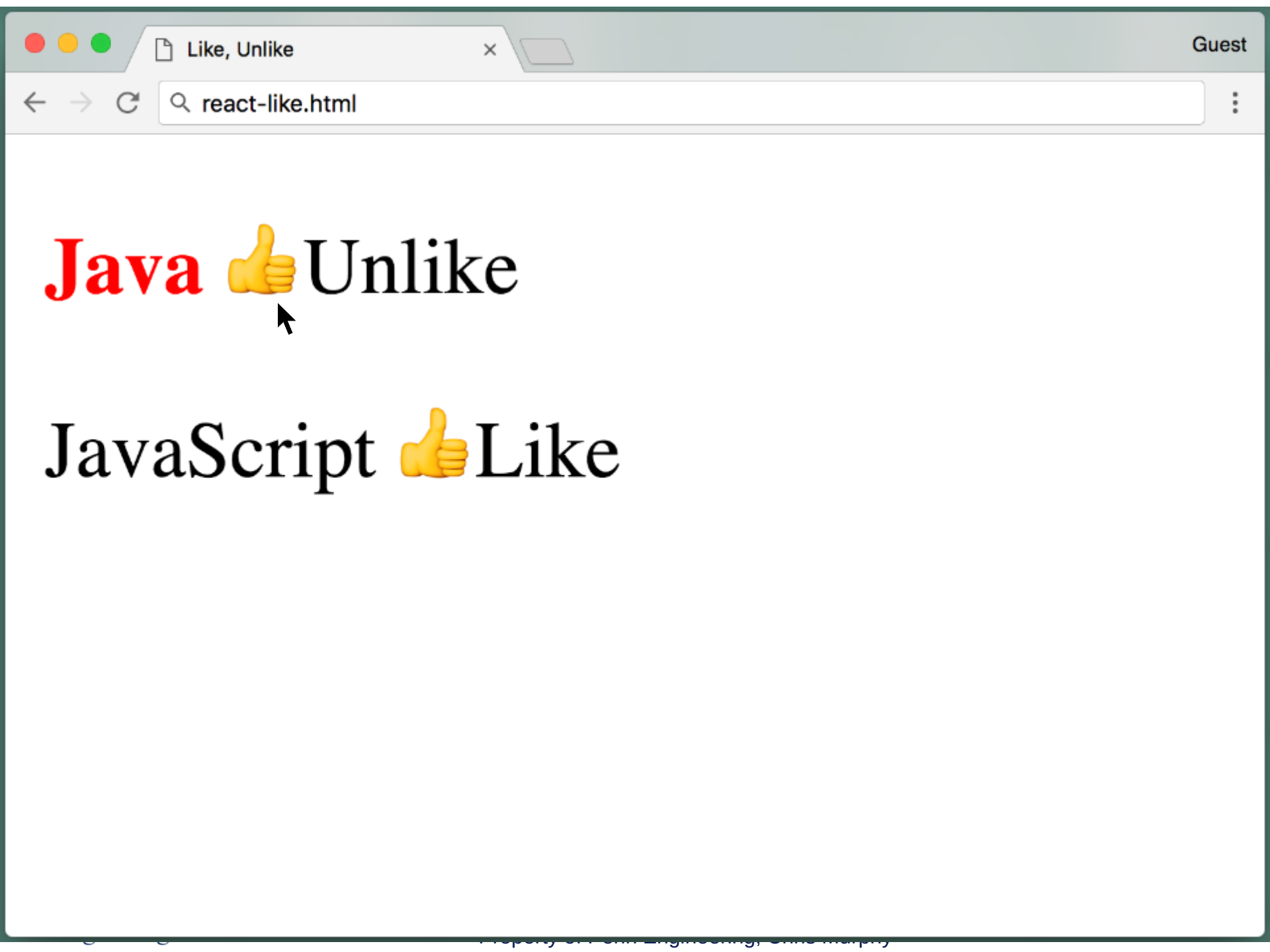
```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

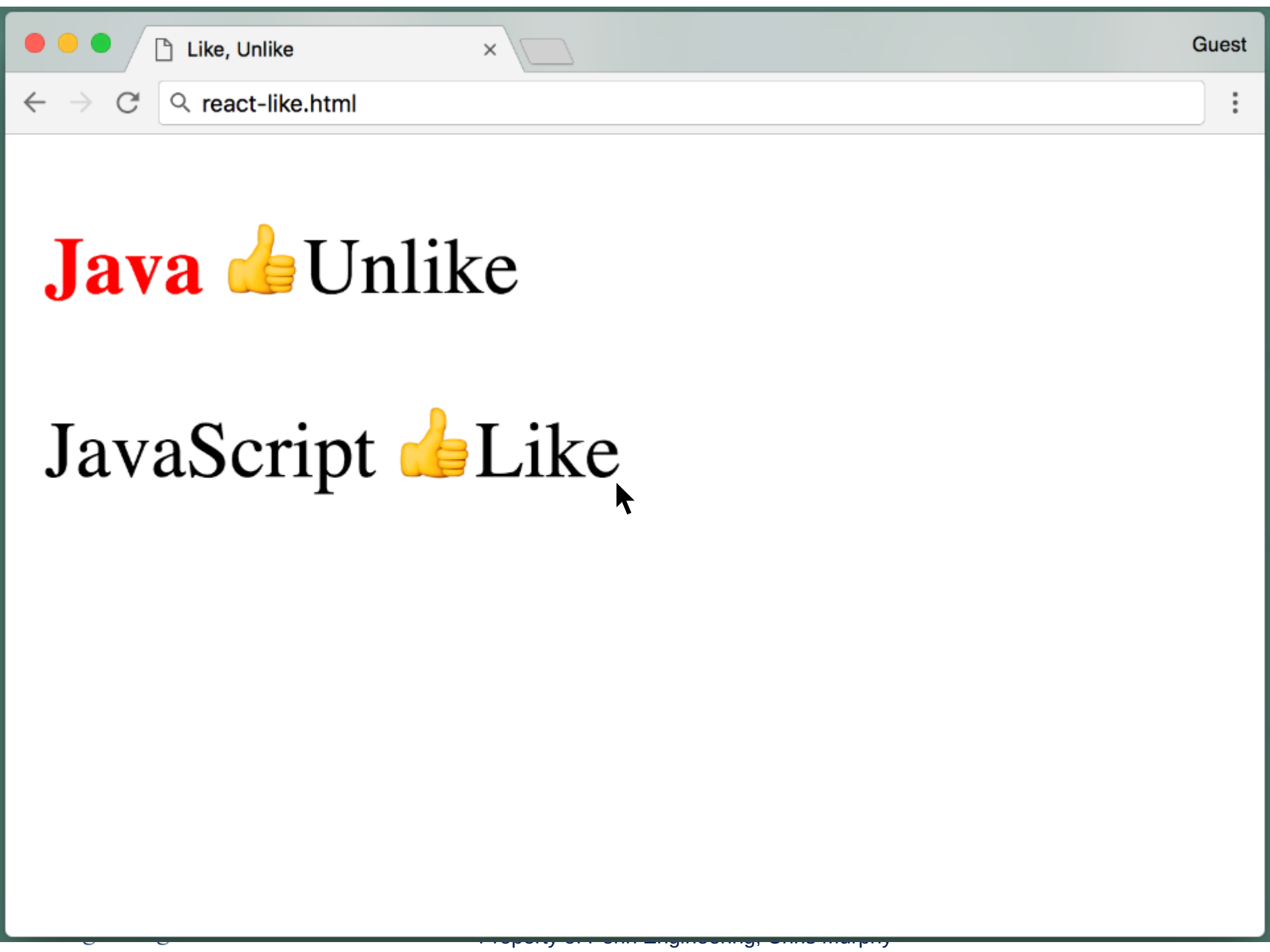
```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

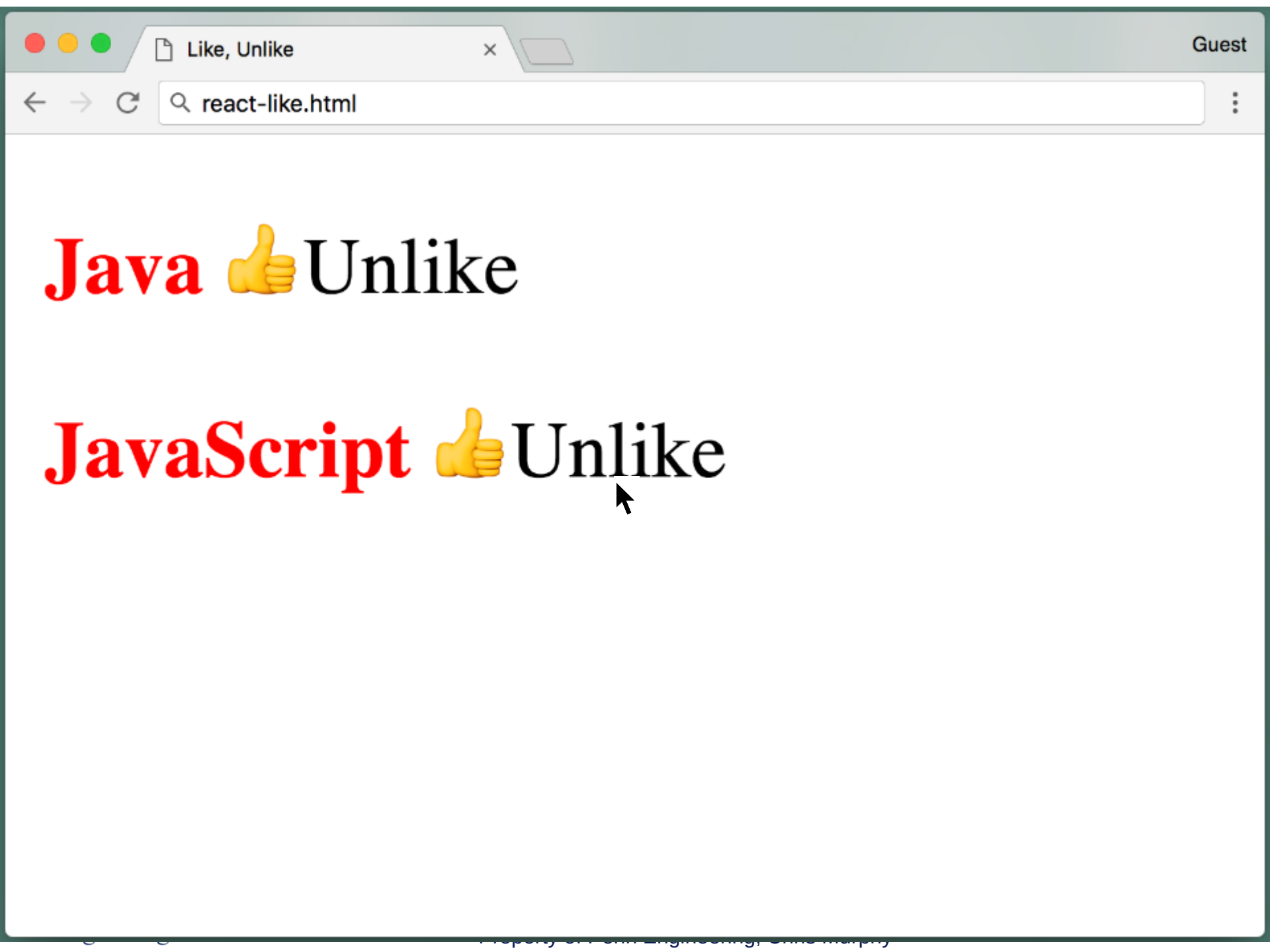
```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  incrementCount() { // callback function
    this.setState( { count: this.state.count + 1; } );
  }

  render() { // invoked when setState is called
   return (
      <div> Count: { this.state.count }
      <button type="button"
        onClick={ this.incrementCount.bind(this); } >
         Increment
      </button>
      </div>
   );
  }
};
```

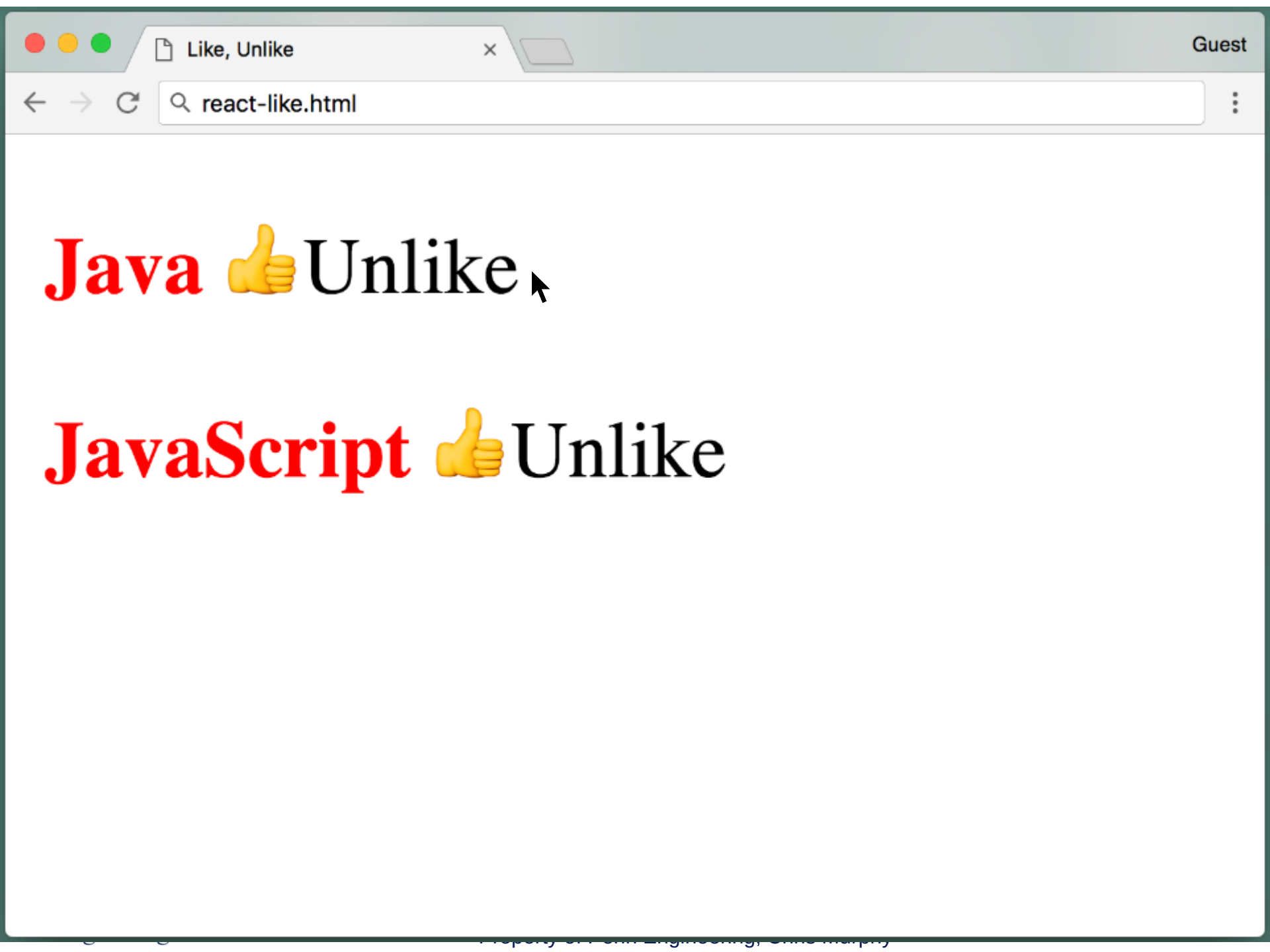

Like, Unlike
Guest
react-like.html
Java 👍Like
JavaScript 👍Like

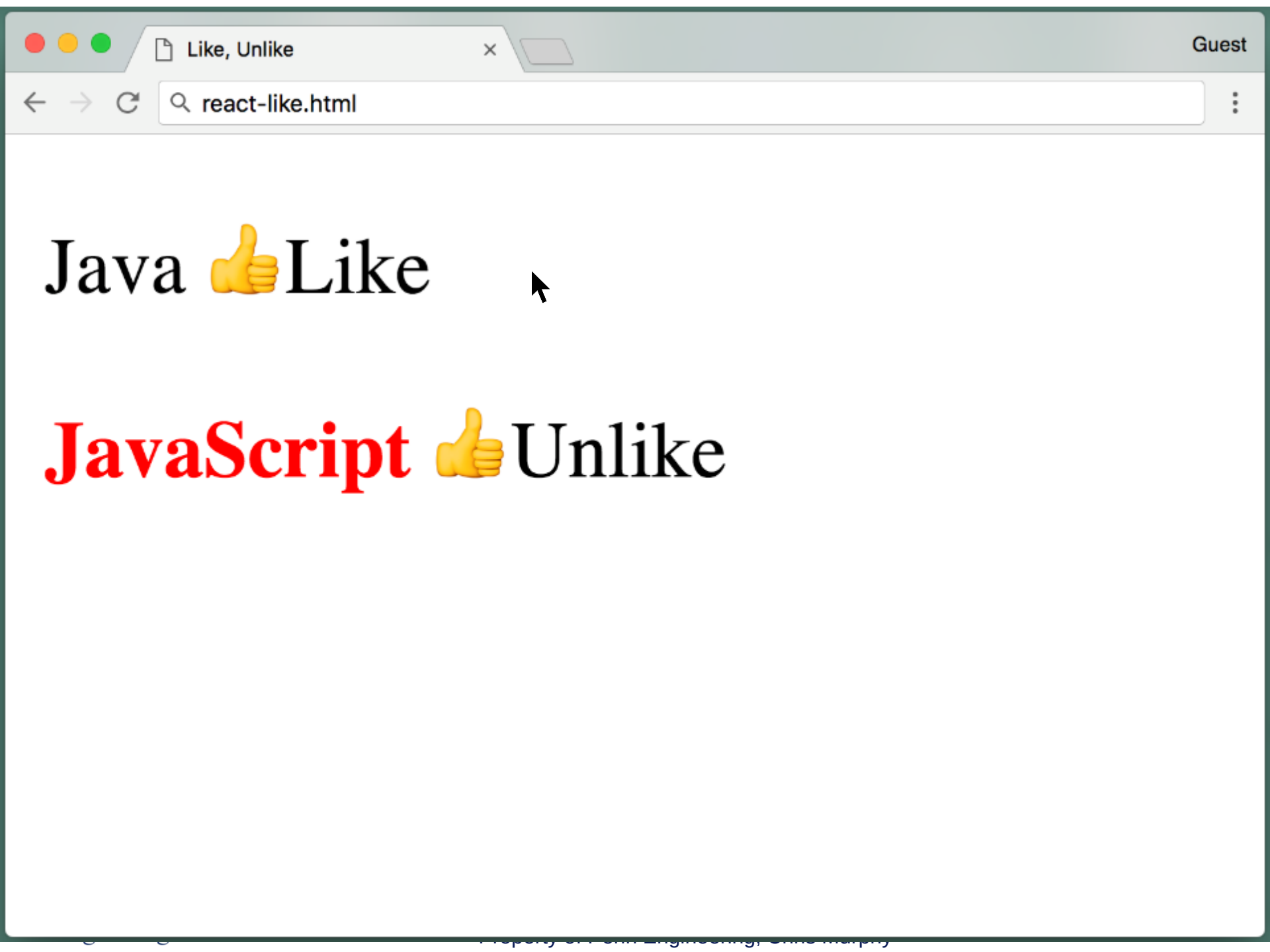

Like, Unlike
Guest
react-like.html
Java 👍Like
JavaScript 👍Like


Like, Unlike
Guest
react-like.html
Java 👍Unlike
JavaScript 👍Like


Like, Unlike
Guest
react-like.html
Java 👍Unlike
JavaScript 👍Like


Like, Unlike
Guest
react-like.html
Java 👍Unlike
JavaScript 👍Unlike


Like, Unlike
Guest
react-like.html
Java 👍Unlike
JavaScript 👍Unlike


Like, Unlike
Guest
react-like.html
Java 👍Like
JavaScript 👍Unlike

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
ReactDOM.render(
  <div>
    <LikeButton name="Java" />
    <LikeButton name="JavaScript" />
  </div>,
document.getElementById('container'));
```

```
class LikeButton extends React.Component {
  constructor(props) {
     super(props);
     this.state = { liked : false };
  }

  toggle() {
     this.setState( {liked: !this.state.liked} );
  }

  render() {
     var name = this.props.name;
     var txt = this.state.liked ? 'Unlike' : 'Like';
     var myColor = this.state.liked ? 'red' : 'black';
     var weight = this.state.liked ? 'bold' : 'normal';
     return (
     <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
     super(props);
     this.state = { liked : false };
  }

  toggle() {
     this.setState( {liked: !this.state.liked} );
  }

  render() {
     var name = this.props.name;
     var txt = this.state.liked ? 'Unlike' : 'Like';
     var myColor = this.state.liked ? 'red' : 'black';
     var weight = this.state.liked ? 'bold' : 'normal';
     return (
     <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
     super(props);
     this.state = { liked : false };
  }

  toggle() {
     this.setState( {liked: !this.state.liked} );
  }

  render() {
     var name = this.props.name;
     var txt = this.state.liked ? 'Unlike' : 'Like';
     var myColor = this.state.liked ? 'red' : 'black';
     var weight = this.state.liked ? 'bold' : 'normal';
     return (
     <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
      {name} </span>
      <span onClick={this.toggle.bind(this)}>
      {'\ud83d\udc4d' + txt}
      </span> </p> );
  }
};
```

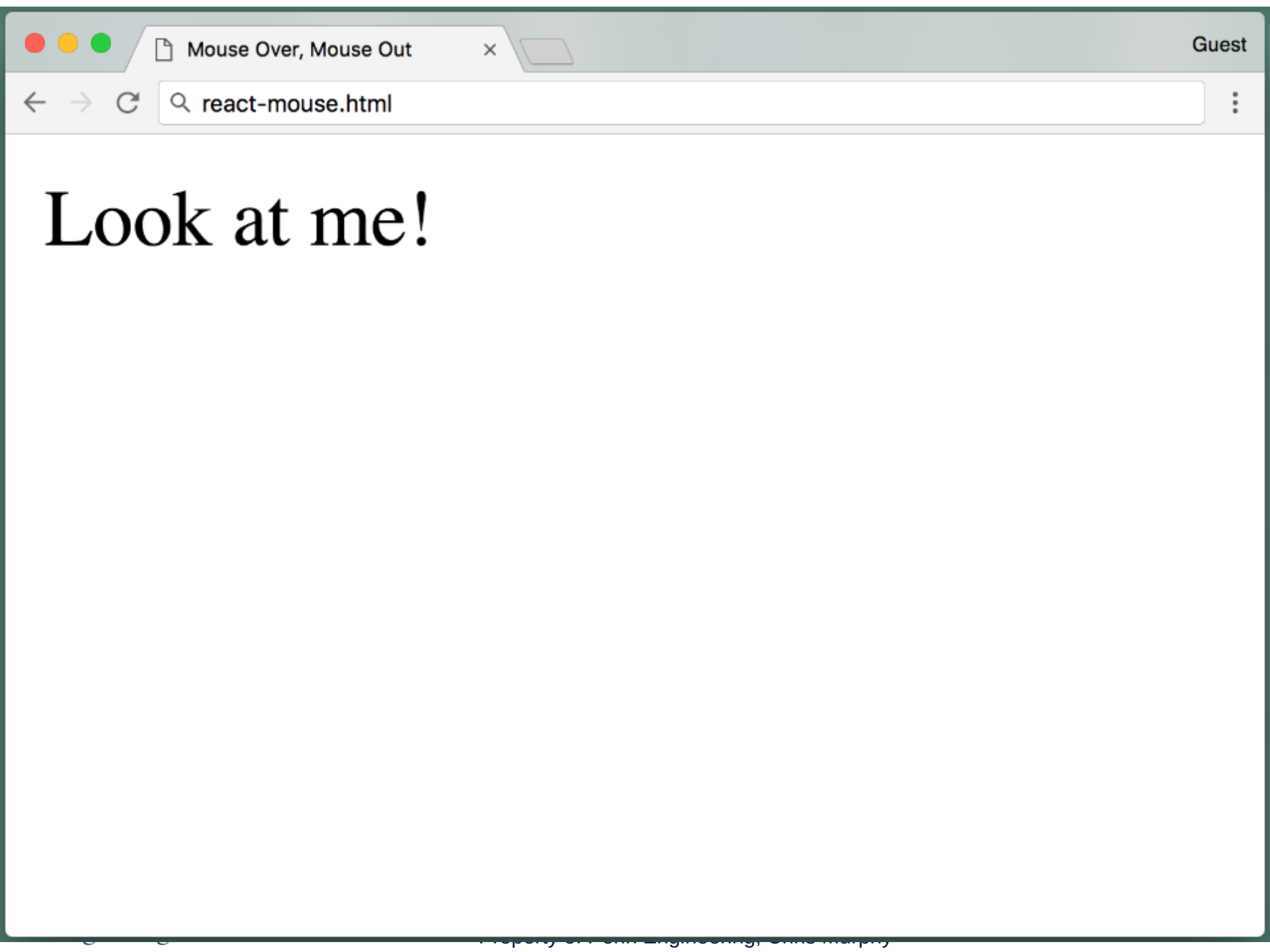
```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

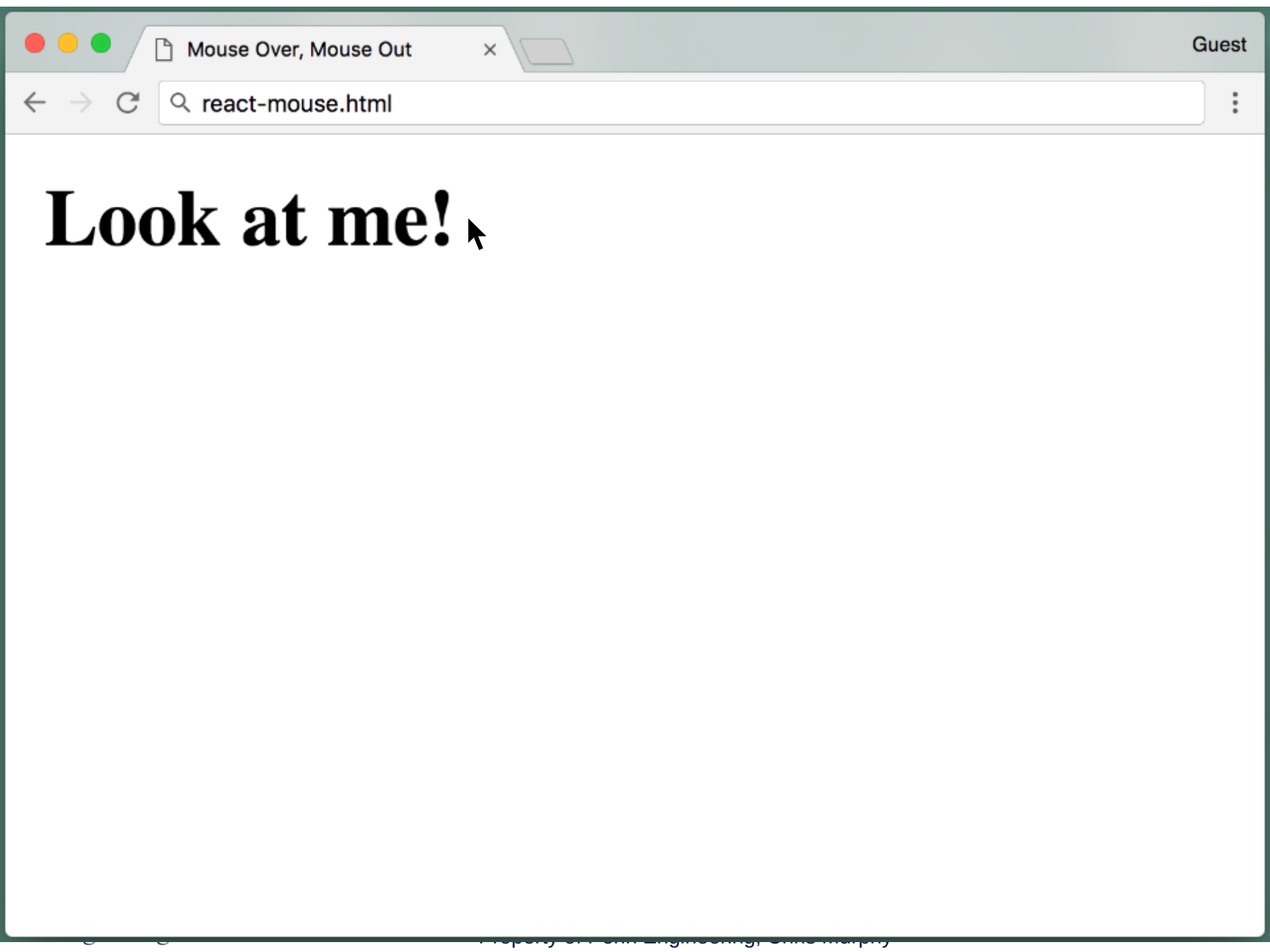
```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

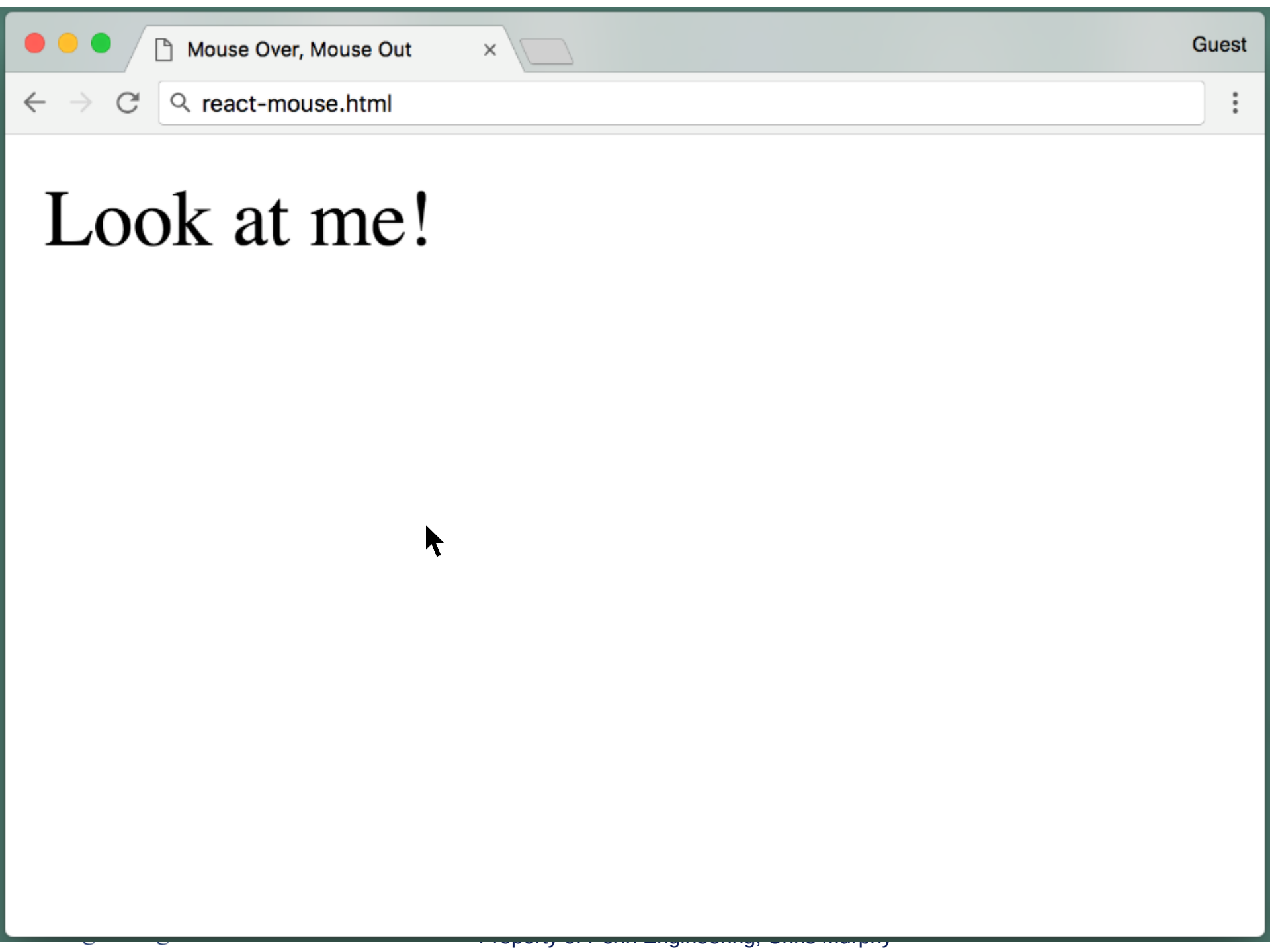
```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

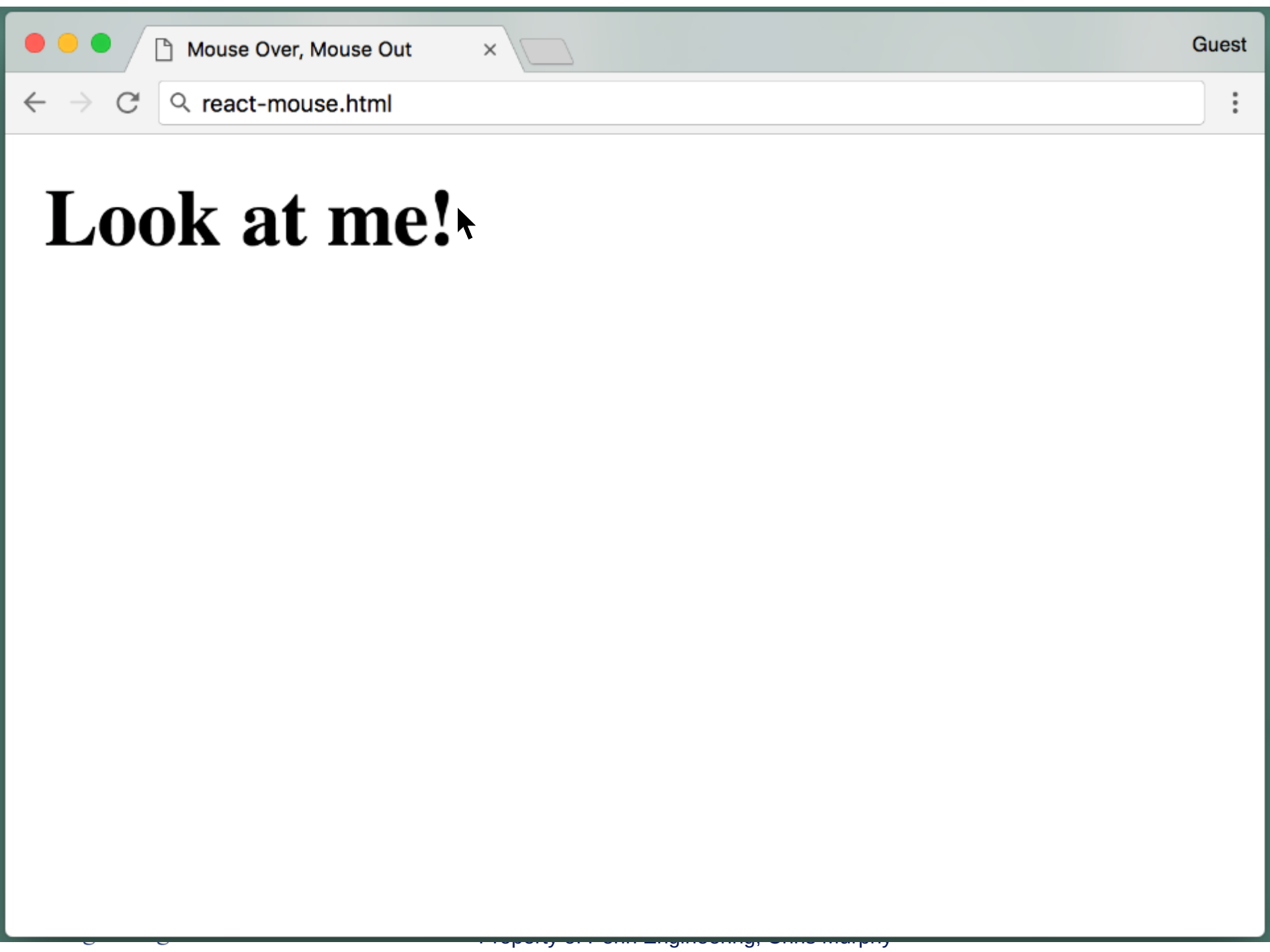
```
class LikeButton extends React.Component {
  constructor(props) {
     super(props);
     this.state = { liked : false };
  }

  toggle() {
     this.setState( {liked: !this.state.liked} );
  }

  render() {
     var name = this.props.name;
     var txt = this.state.liked ? 'Unlike' : 'Like';
     var myColor = this.state.liked ? 'red' : 'black';
     var weight = this.state.liked ? 'bold' : 'normal';
     return (
     <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```

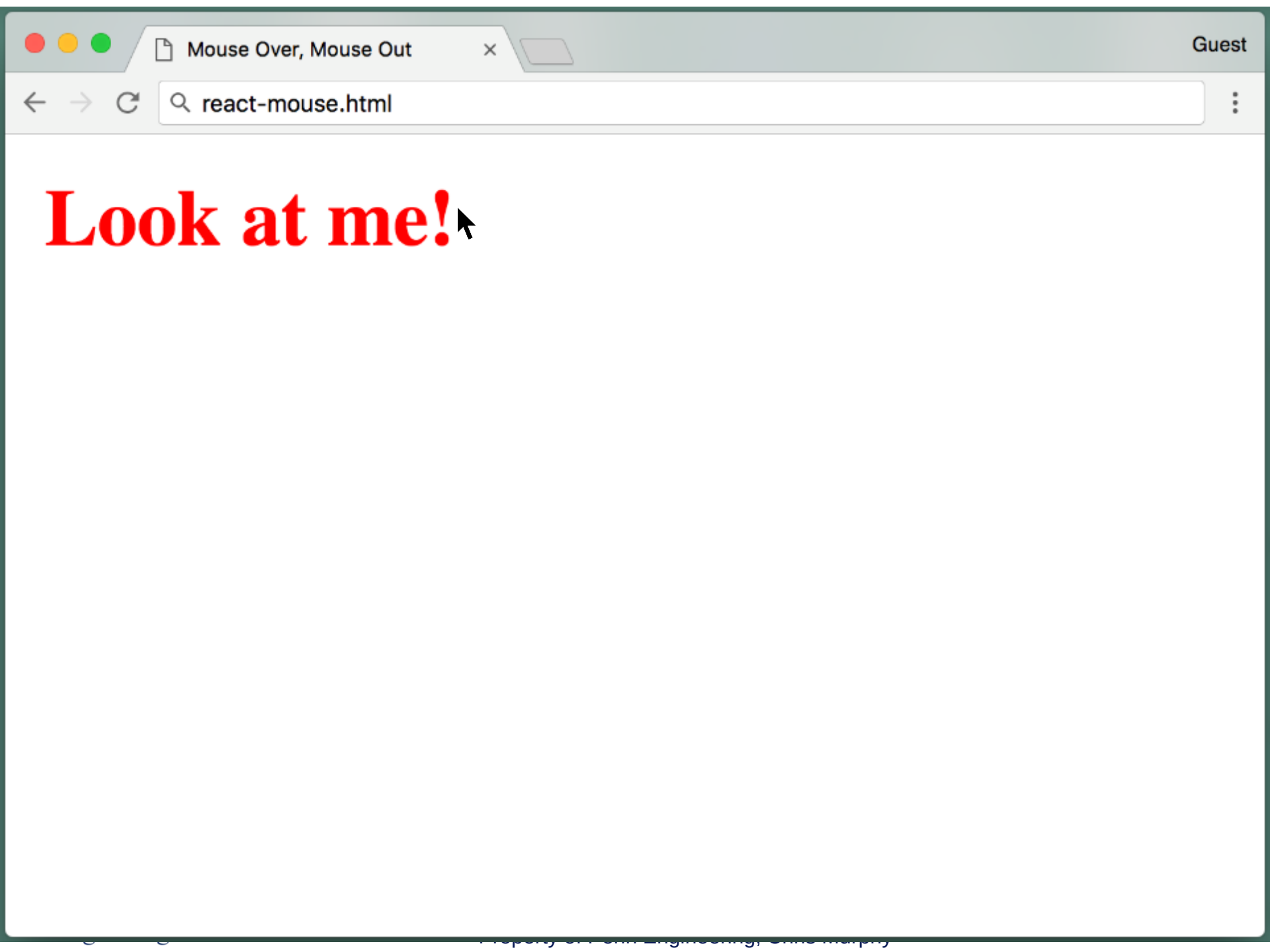
```
class LikeButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { liked : false };
  }

  toggle() {
    this.setState( {liked: !this.state.liked} );
  }

  render() {
    var name = this.props.name;
    var txt = this.state.liked ? 'Unlike' : 'Like';
    var myColor = this.state.liked ? 'red' : 'black';
    var weight = this.state.liked ? 'bold' : 'normal';
    return (
    <p><span style={{color:myColor, fontWeight:weight }}>
        {name} </span>
        <span onClick={this.toggle.bind(this)}>
        {'\ud83d\udc4d' + txt}
        </span> </p> );
  }
};
```


Mouse Over, Mouse Out
Guest
react-mouse.html
Look at me!


Mouse Over, Mouse Out
Guest
react-mouse.html
Look at me!


Mouse Over, Mouse Out
Guest
react-mouse.html
Look at me!


Mouse Over, Mouse Out
Guest
react-mouse.html
Look at me!


Mouse Over, Mouse Out
Guest
react-mouse.html
Look at me!

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

  constructor(props) {
      super(props);
      this.state = { bold : false, color : 'black' };
  }

  handleMouseOver() {
      this.setState( { bold : true } );
  }

  handleMouseOut() {
      this.setState( { bold: false } );
  }

  handleClick() {
     if (this.state.color == 'red' )
         this.setState( { color: 'black' } );
     else
         this.setState( { color: 'red' } );
  }

  . . .
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
     );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

```
class MyText extends React.Component {

   . . .

  render () {
     var myColor = this.state.color;
     var weight = this.state.bold ? 'bold' : 'normal' ;
     return (
       <span style={ {color:myColor, fontWeight:weight} }
       onClick={this.handleClick.bind(this)}
           onMouseOver={this.handleMouseOver.bind(this)}
           onMouseOut={this.handleMouseOut.bind(this)} >
       {this.props.text}
       </span>
      );
   }
};

ReactDOM.render(
    <div><MyText text="Look at me!" /></div>,
    document.getElementById('container')
);
```

Summary

- We can bind user events in HTML elements to callback functions in React components

- When we invoke a component's `setState` function, the `render` function will automatically be called and the component's appearance can change accordingly