# INTRO TO REACT

**Liz Lovero**

@lizzardl elizabethlovero.com

**React** is a Javascript library for creating user interfaces
in large applications with data that changes over time.

Built by Jordan Walke (@jordwalke) at Facebook and first implemented on the newsfeed in 2011.

Open-source since 2013.

# JUST THE VIEW

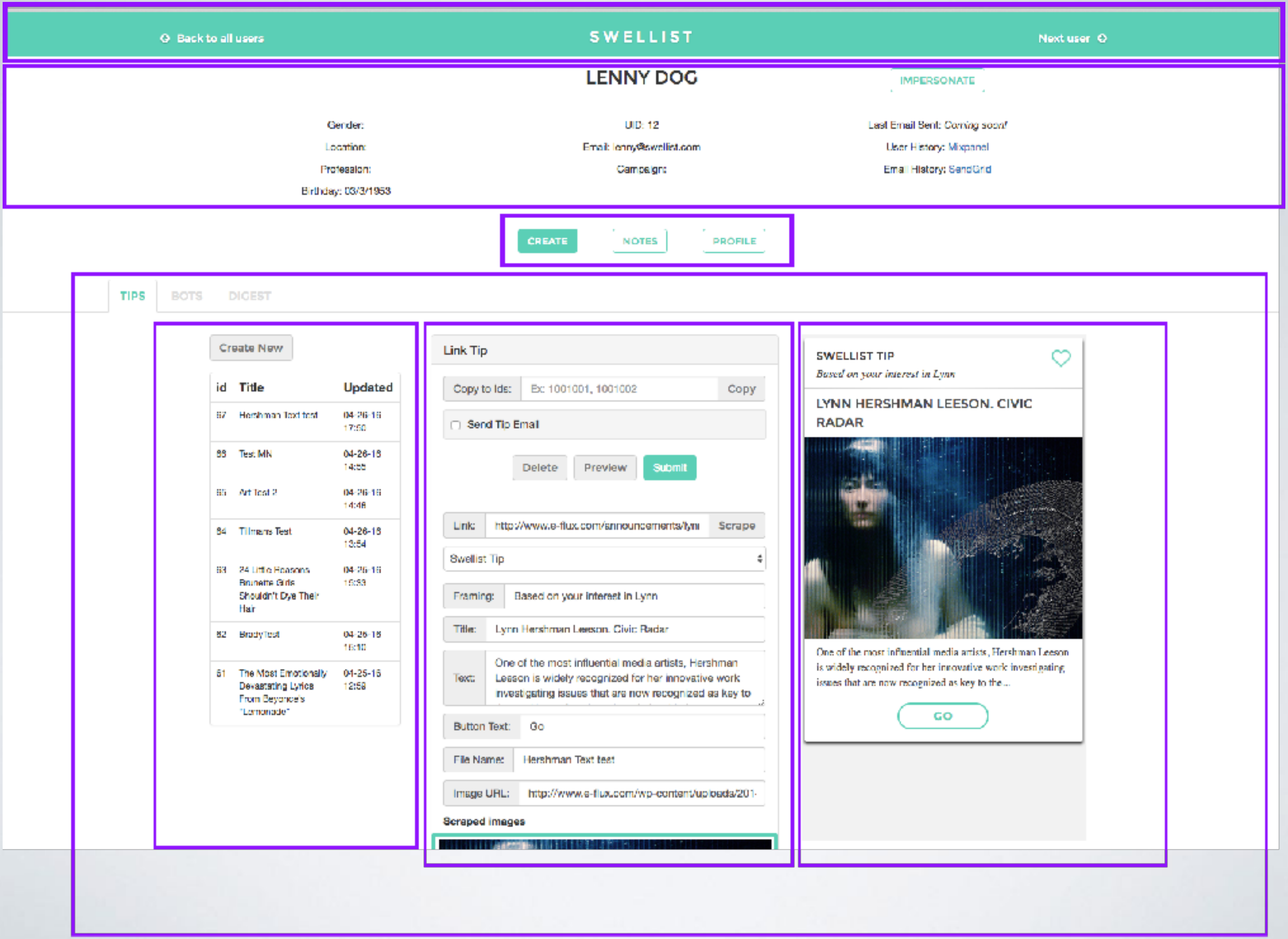
- React is not a complete web framework like Rails or Angular - its only the **V** in MVC

- Provides a language and some function hooks to render HTML. That's **all** React outputs, HTML.

- You **cannot** build a fully functional dynamic application with React alone.  Rather its **BYOB** (Bring Your Own Backend)

# BUILD COMPONENTS, NOT TEMPLATES

- Unlike other frameworks or templating tools, React breaks down the view into bundles of HTML / Javascript, called "components".
- UI Components are cohesive units that tightly couple UI description and UI logic.
- No more hunting and guessing as to a UI state!



LENNY DOG

Gender:
Location:
Profession:
Birthday: 03/3/1953

UID: 12
Email: lenny@swellist.com
Campaign:

Last Email Sent: Coming soon!
User History: Mixpanel
Email History: SendGrid

| id | Title | Updated |
|---|---|---|
| 67 | Hershman Text test | 04-26-16 17:50 |
| 66 | Test MN | 04-26-16 14:55 |
| 65 | Art Test 2 | 04-26-16 14:48 |
| 64 | Tilmans Test | 04-26-16 13:54 |
| 63 | 24 Little Reasons Brunette Girls Shouldn't Dye Their Hair | 04-26-16 15:53 |
| 62 | BradyTest | 04-26-16 16:10 |
| 61 | The Most Emotionally Devastating Lyrics From Beyonce's "Lemonade" | 04-25-16 12:58 |

Link Tip

Copy to Ids: Ex: 1001001, 1001002 Copy

□ Send Tip Email

Link: http://www.e-flux.com/announcements/lynı Scrape

Swellist Tip

Framing: Based on your interest in Lynn

Title: Lynn Hershman Leeson. Civic Radar

Text: One of the most influential media artists, Hershman Leeson is widely recognized for her innovative work investigating issues that are now recognized as key to

Button Text: Go

File Name: Hershman Text test

Image URL: http://www.e-flux.com/wp-content/uploads/201-

Scraped images

SWELLIST TIP

*Based on your interest in Lynn*

LYNN HERSHMAN LEESON. CIVIC RADAR

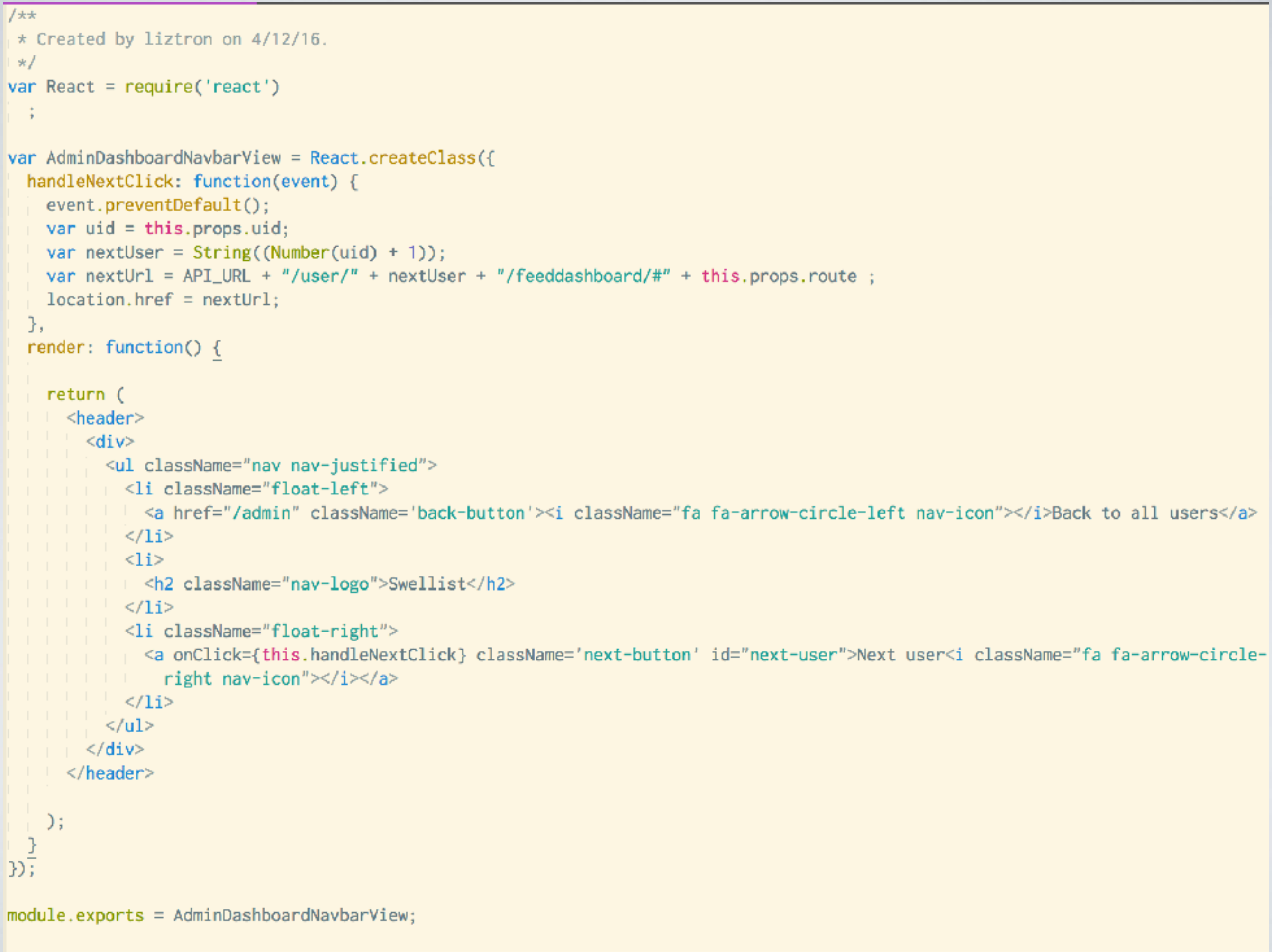
One of the most influential media artists, Hershman Leeson is widely recognized for her innovative work investigating issues that are now recognized as key to the ...

# HOW IT WORKS

- Components are JS functions that all contain a render function which uses JSX syntax to return HTML.

- Each component may contain data but the view renders in a declarative way - based only on current data state.

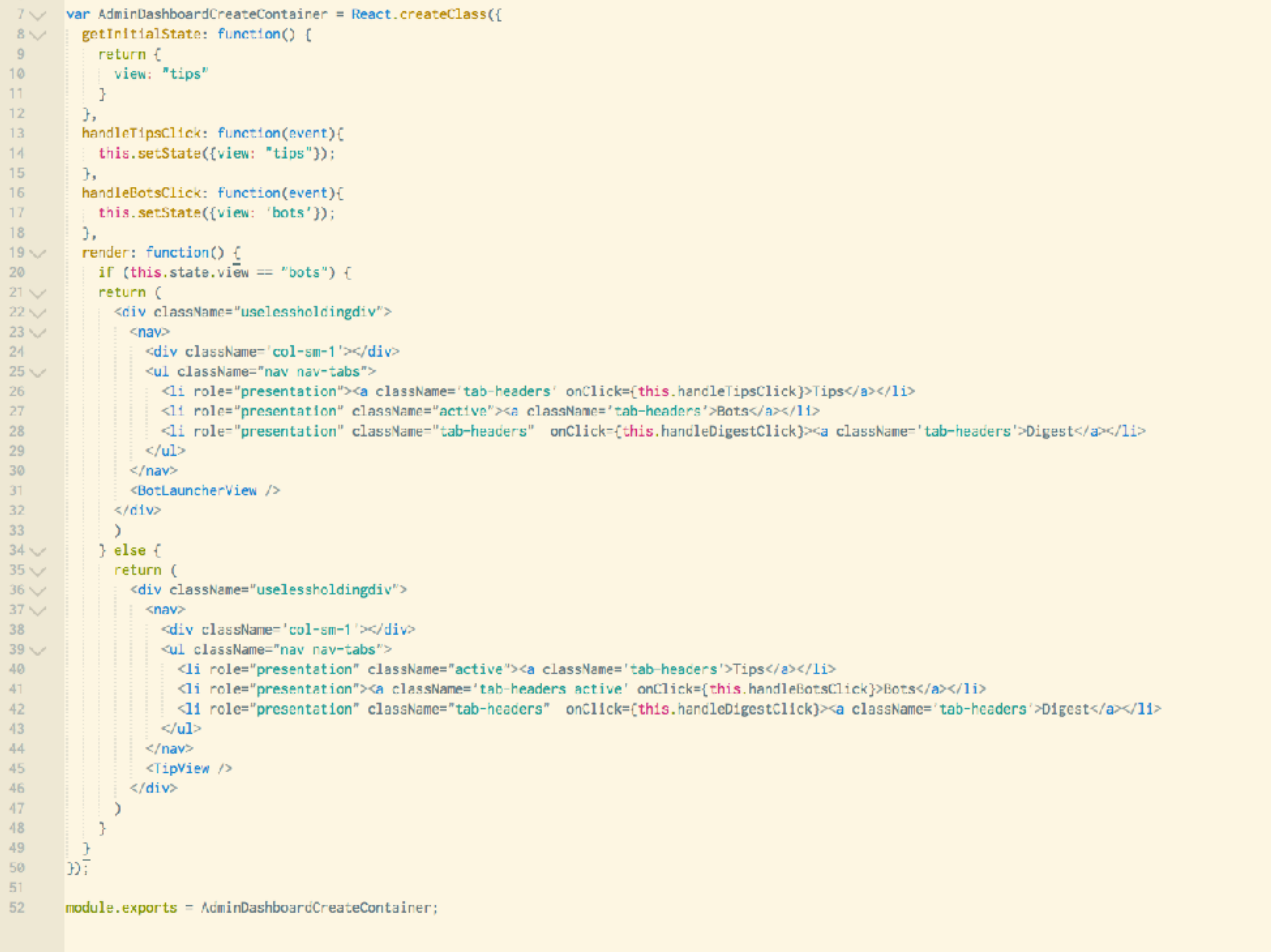
- React hits the "refresh" button any time data changes, and knows to only update the changed parts. React creates a lightweight Virtual DOM every time components are rendered.



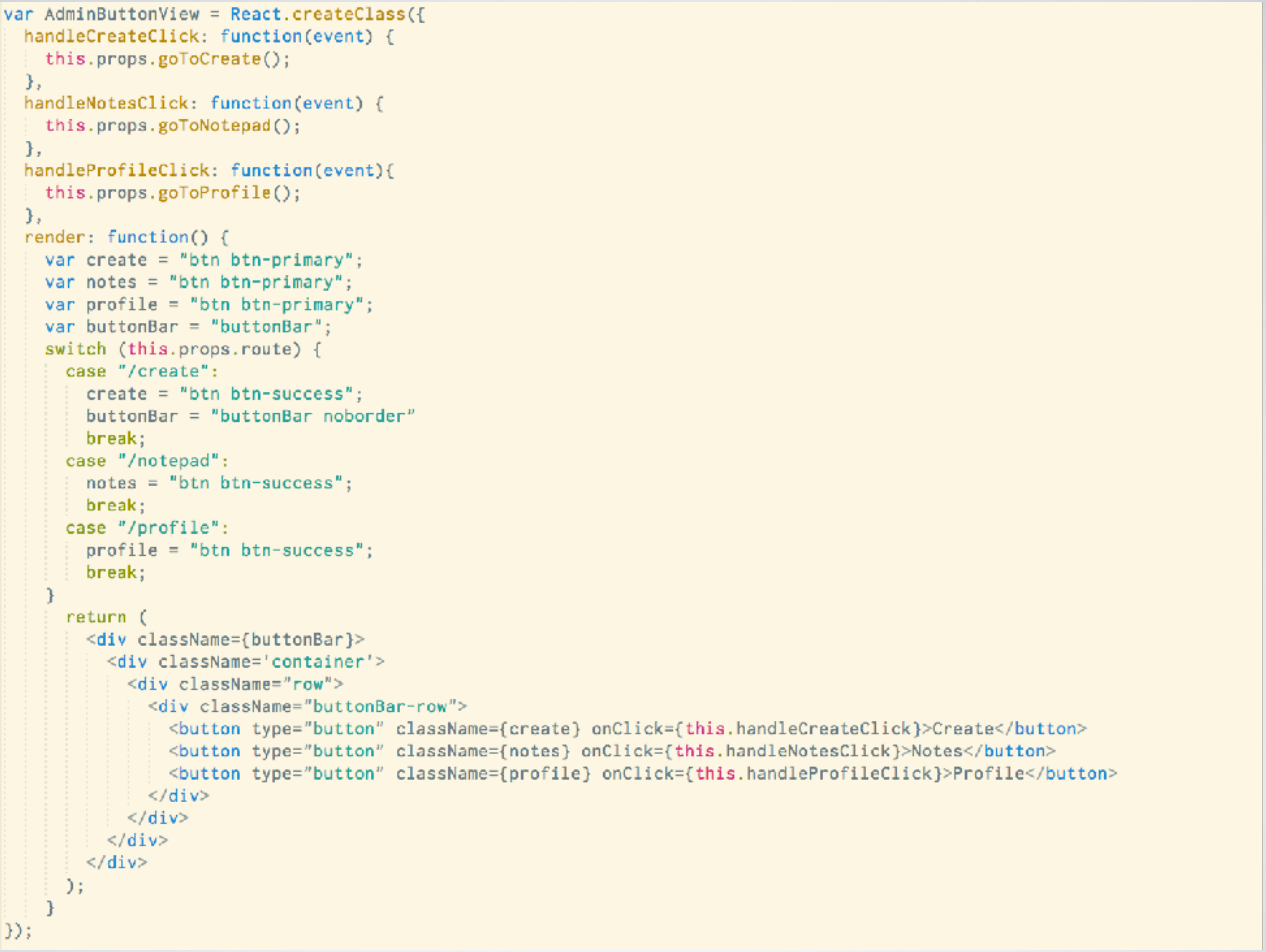
```
/**
 * Created by liztron on 4/12/16.
 */
var React = require('react')
  ;

var AdminDashboardNavbarView = React.createClass({
  handleNextClick: function(event) {
    event.preventDefault();
    var uid = this.props.uid;
    var nextUser = String((Number(uid) + 1));
    var nextUrl = API_URL + "/user/" + nextUser + "/feeddashboard/#" + this.props.route ;
    location.href = nextUrl;
  },
  render: function() {

    return (
      <header>
        <div>
          <ul className="nav nav-justified">
            <li className="float-left">
              <a href="/admin" className='back-button'><i className="fa fa-arrow-circle-left nav-icon"></i>Back to all users</a>
            </li>
            <li>
              <h2 className="nav-logo">Swellist</h2>
            </li>
            <li className="float-right">
              <a onClick={this.handleNextClick} className='next-button' id="next-user">Next user<i className="fa fa-arrow-circle-
                right nav-icon"></i></a>
            </li>
          </ul>
        </div>
      </header>

    );
  }
});

module.exports = AdminDashboardNavbarView;
```

# STATE & PROPS

- Each component has 2 inputs:
  - props - shortcut of properties, these are meant to be immutable (like a click event on a button)
  - state - mutable (such as which tab is selected in a tab view) Tend to occur at the top / parent component
- Data is passed down and events circle around back up. (aka props flow down and actions flow up.)



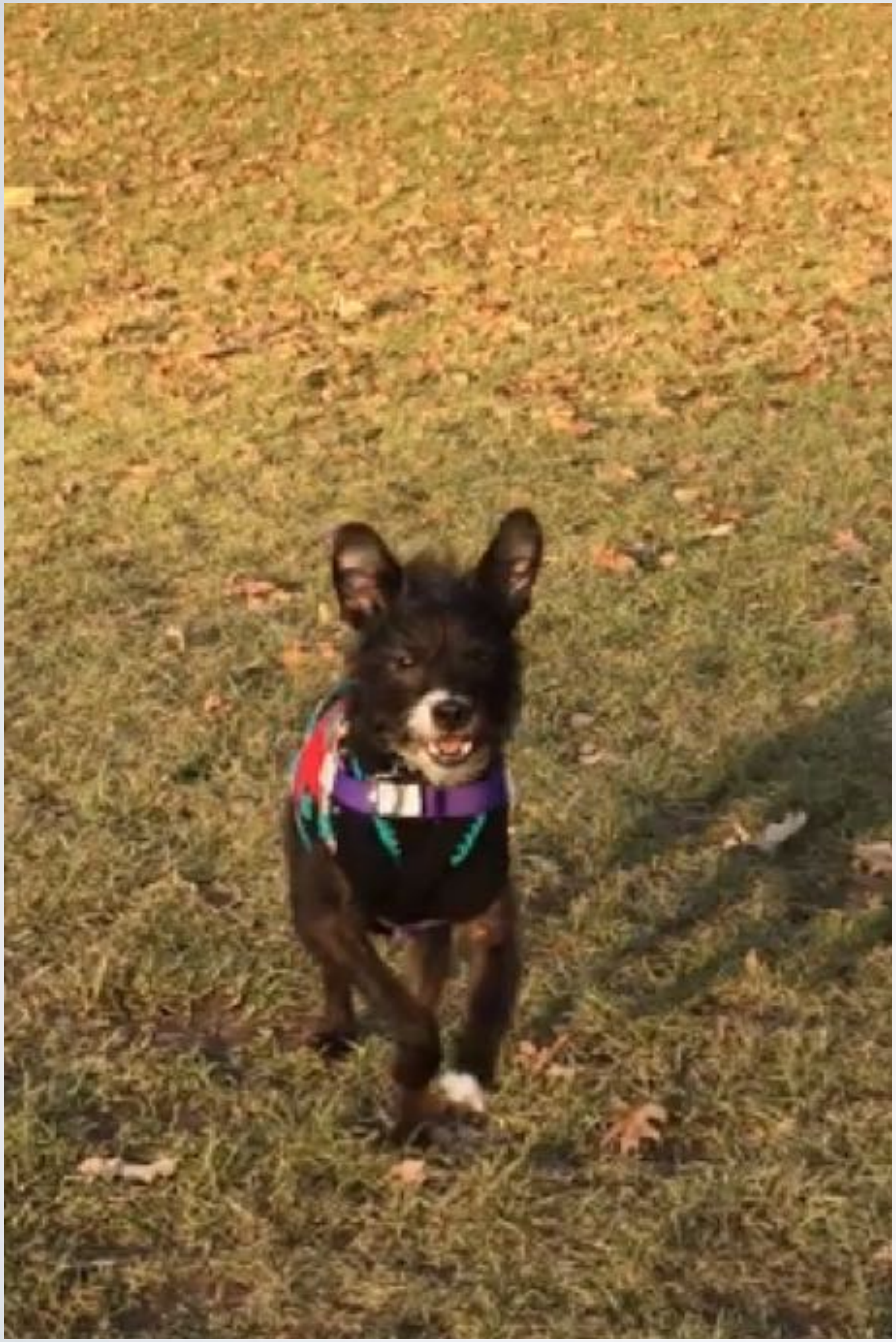
```
var AdminDashboardCreateContainer = React.createClass({
  getInitialState: function() {
    return {
      view: "tips"
    }
  },
  handleTipsClick: function(event){
    this.setState({view: "tips"});
  },
  handleBotsClick: function(event){
    this.setState({view: 'bots'});
  },
  render: function() {
    if (this.state.view == "bots") {
    return (
      <div className="uselessholdingdiv">
        <nav>
          <div className='col-sm-1'></div>
          <ul className="nav nav-tabs">
            <li role="presentation"><a className='tab-headers' onClick={this.handleTipsClick}>Tips</a></li>
            <li role="presentation" className="active"><a className='tab-headers'>Bots</a></li>
            <li role="presentation" className="tab-headers"  onClick={this.handleDigestClick}><a className='tab-headers'>Digest</a></li>
          </ul>
        </nav>
        <BotLauncherView />
      </div>
      )
    } else {
      return (
        <div className="uselessholdingdiv">
          <nav>
            <div className='col-sm-1'></div>
            <ul className="nav nav-tabs">
              <li role="presentation" className="active"><a className='tab-headers'>Tips</a></li>
              <li role="presentation"><a className='tab-headers active' onClick={this.handleBotsClick}>Bots</a></li>
              <li role="presentation" className="tab-headers"  onClick={this.handleDigestClick}><a className='tab-headers'>Digest</a></li>
            </ul>
          </nav>
          <TipView />
        </div>
      )
    }
  }
});

module.exports = AdminDashboardCreateContainer;
```



```
var AdminButtonView = React.createClass({
  handleCreateClick: function(event) {
    this.props.goToCreate();
  },
  handleNotesClick: function(event) {
    this.props.goToNotepad();
  },
  handleProfileClick: function(event){
    this.props.goToProfile();
  },
  render: function() {
    var create = "btn btn-primary";
    var notes = "btn btn-primary";
    var profile = "btn btn-primary";
    var buttonBar = "buttonBar";
    switch (this.props.route) {
      case "/create":
        create = "btn btn-success";
        buttonBar = "buttonBar noborder"
        break;
      case "/notepad":
        notes = "btn btn-success";
        break;
      case "/profile":
        profile = "btn btn-success";
        break;
    }
      return (
        <div className={buttonBar}>
          <div className='container'>
            <div className="row">
              <div className="buttonBar-row">
                <button type="button" className={create} onClick={this.handleCreateClick}>Create</button>
                <button type="button" className={notes} onClick={this.handleNotesClick}>Notes</button>
                <button type="button" className={profile} onClick={this.handleProfileClick}>Profile</button>
              </div>
            </div>
          </div>
        </div>
      );
    }
});
```

# THE GOOD

- You can always tell how your component will render by looking at one source file. You don't have to trace program flow.
- Works great for teams, strongly enforcing UI and workflow patterns
- UI code is readable, reusable, and maintainable
- Every time components are rendered, React diffs them to figure out the smallest amount of real DOM changes needed. The result is super fast performance since DOM changes are the slowest part.



# THE BAD

- "JSX Violates Separation Of Concerns''

- Will slow development down tremendously at the start.

- Does not support any browser below IE8, and never will.

- If your application / website doesn't have very much dynamic page updating, you will be implementing a lot of code for a very small benefit.

# LEARN MORE

- Wes Bos - https://reactforbeginners.com/
- Facebook React Docs - https://facebook.github.io/react/docs/why-react.html
- Introduction to React.js (by its authors) https://youtu.be/XxVg_s8xAms
- Flux - the pattern for ‘one way’ data flow http://blog.andrewray.me/flux-for-stupid-people/