








78724 Zip


Donald J. Trump
President of the United States --- Republican



Mike Pence
Vice-President of the United States --- Republican



John Cornyn
United States Senate --- Republican


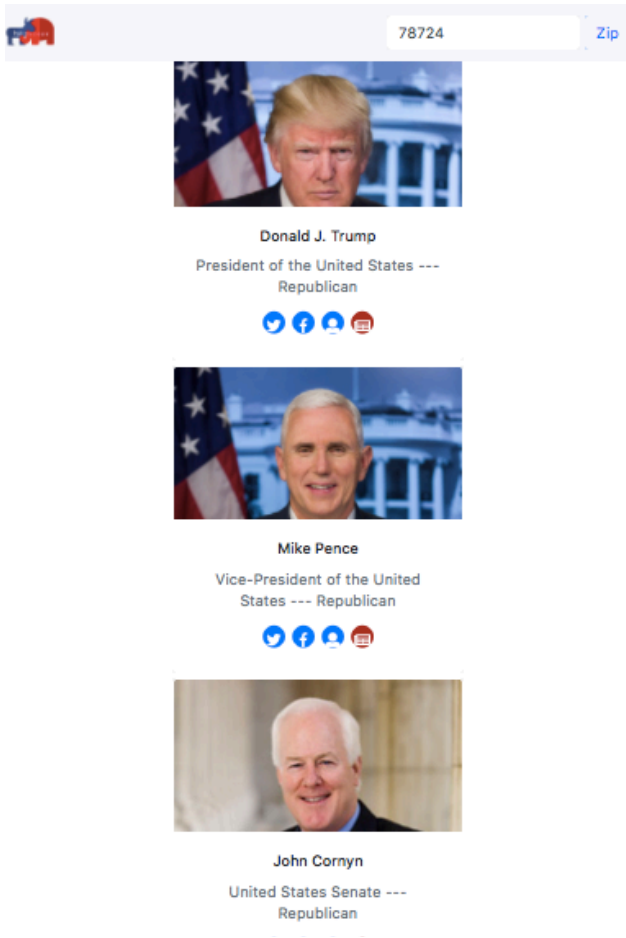

Ted Cruz
United States Senate --- Republican

Who represents you?



Politiscape


Molly Johnson
Ryan Mullowney
Ryan Bonner
B Barnett



Purpose -

Politiscope is built for the millennial. With just knowledge of your own zip code, users can find out who is representing them. Clean, "Scroll-able" profiles will dynamically populate the screen and let users see quick links to their politicians' social media channels, their website, and recent articles from the New York Times.

Look familiar?

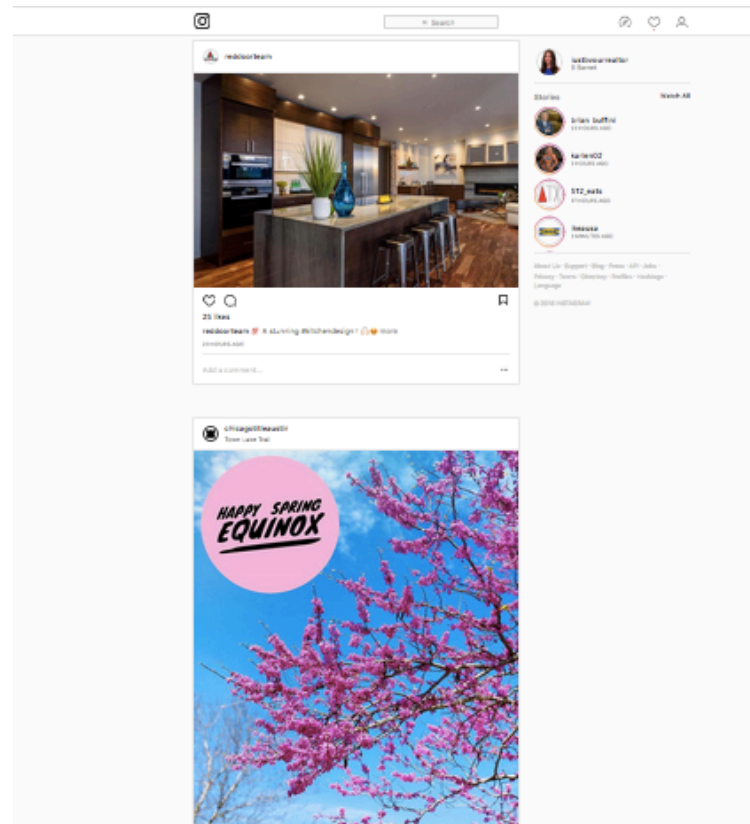




Donald J. Trump
President of the United States ---
Republican




Mike Pence
Vice-President of the United
States --- Republican




Keep it clean!

Politics is a dirty game... so we needed to keep it clean.

Pops of blue and red, and 50 shades of gray... because grayscale doesn't mean boring.

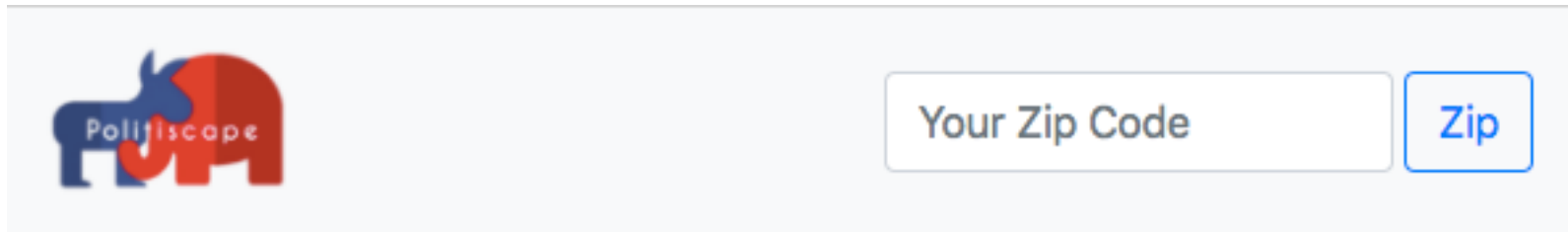


Who represents you?

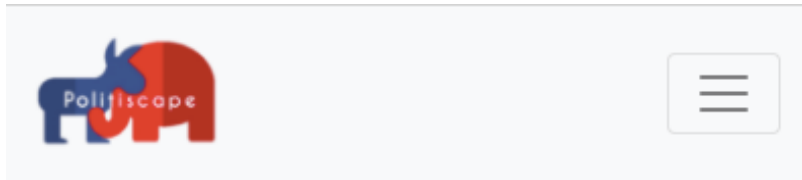


Type in your zip code and find out who your politicians are, their social media links, their website and what New York Times has to say about them.

The code! - HTML



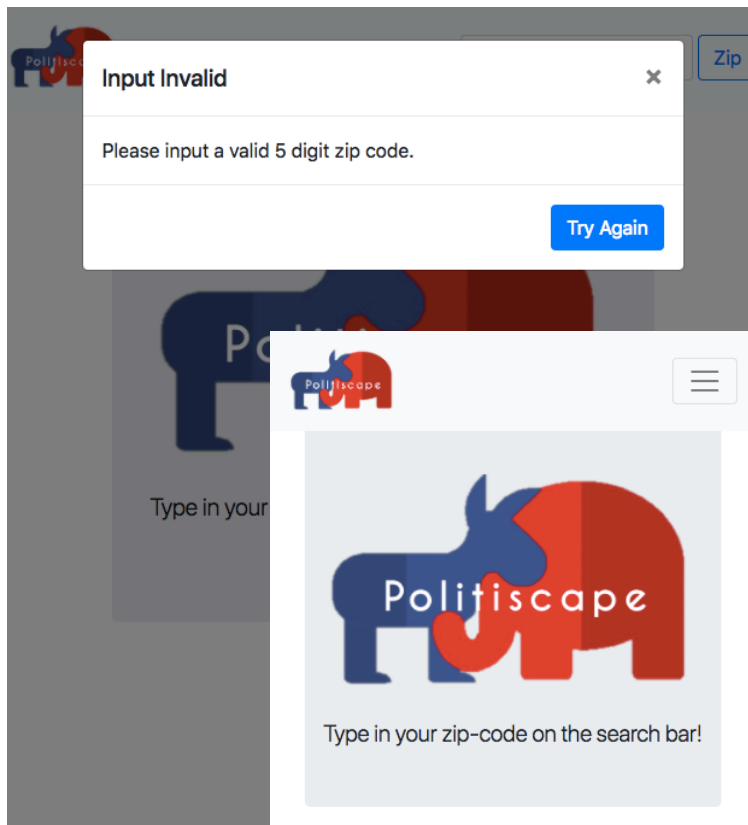
A screenshot of the top navigation bar of the Politiscope website. On the left is the Politiscope logo, which features a stylized elephant in blue and red with the word "Politiscope" in white. To the right of the logo is a light gray input field containing the placeholder text "Your Zip Code". To the right of the input field is a blue button with the word "Zip" in white.



When working with the api, we realized our site built itself through js. However, a few basic elements stayed static in the HTML.

The nav-bar is a bootstrap dream. It fits your screen, and collapses the form on smaller screens. It's static and simple for ease of use.

Zip Codes are 5 numbers!



```
<!-- Modal -->
<div class="modal fade" id="invalidInputModal" tabindex="-1" role="dialog"
aria-labelledby="invalidInputModal" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="invalidInputLabel">Input Invalid</h5>
        <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        Please input a valid 5 digit zip code.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary"
data-dismiss="modal">Try Again</button>
      </div>
    </div>
  </div>
</div>
```

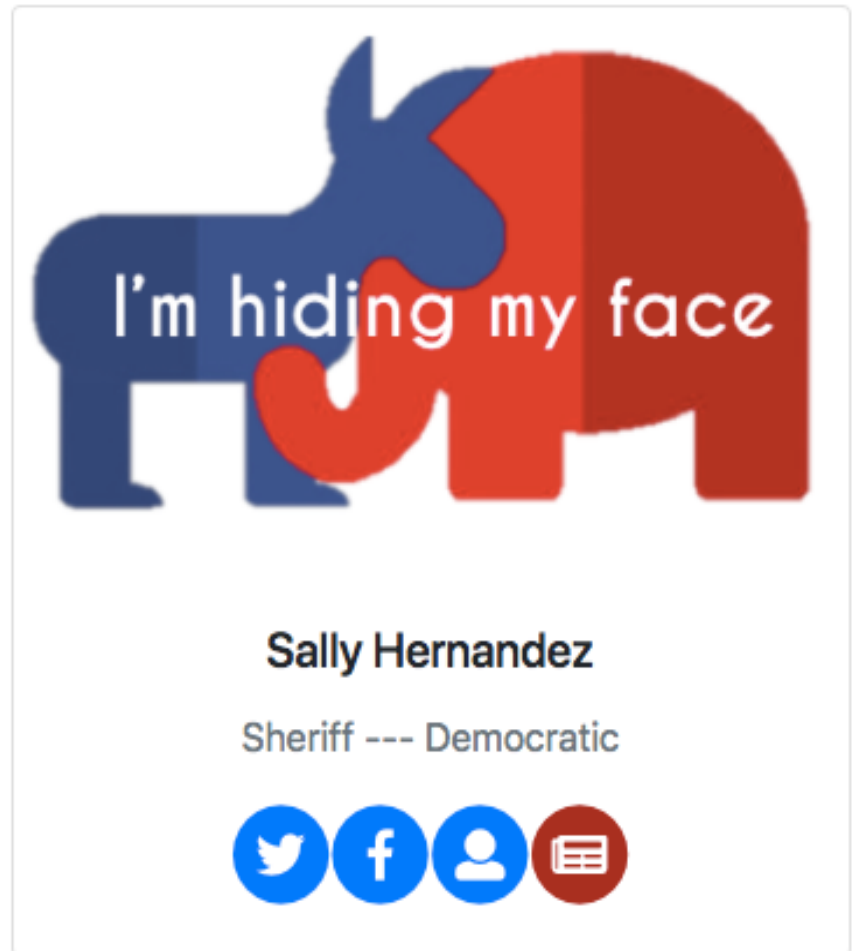
Alerts are ugly!

We used a modal.

Creating the profile

9 Elements exist within each profile:

1. The card
2. A photo
3. Their name
4. Their office
5. Their party
6. Twitter handle
7. Facebook page
8. Website
9. Related New York Times headlines





78724

Zip



Christi Craddick
Commissioner, Railroad
Commission --- Republican



George P. Bush
Commissioner of General Land
Office --- Republican



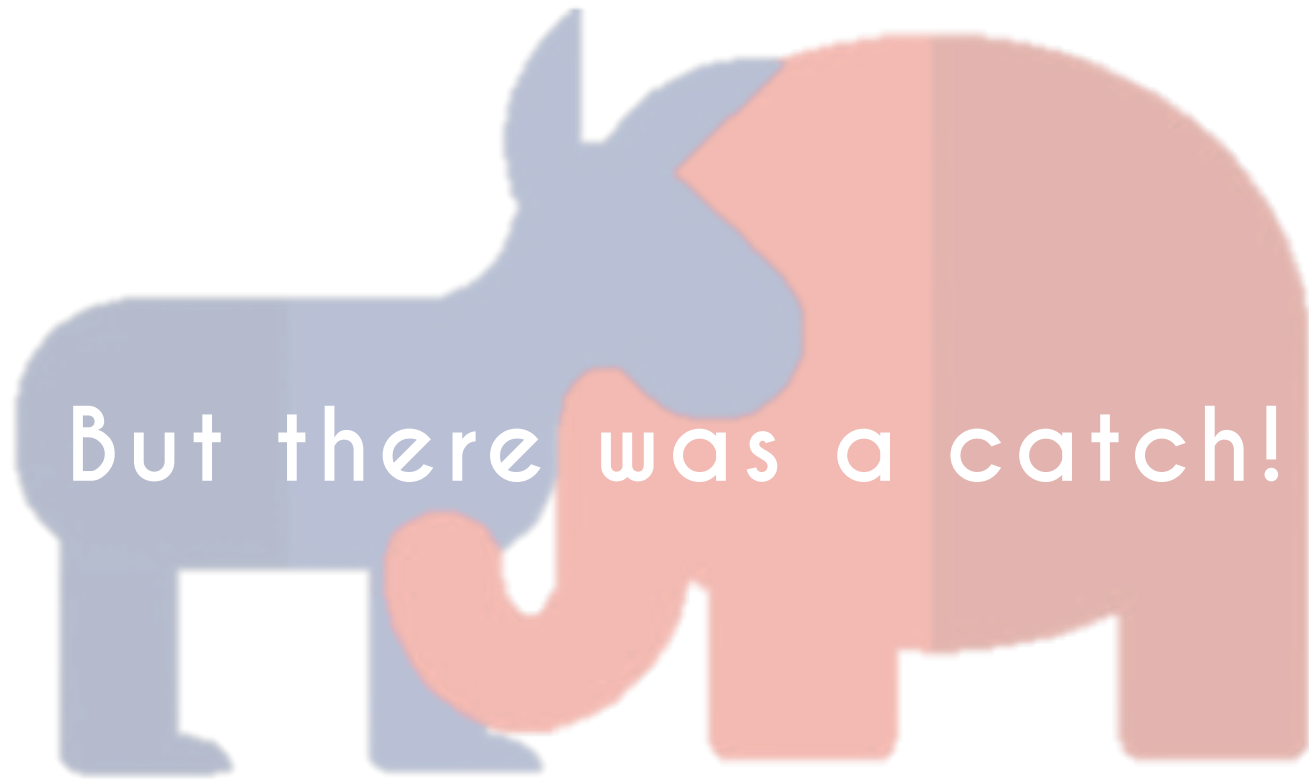
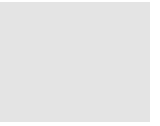
```
var nameMore = "#" + nameID
var more = $("<a>")
more.addClass("moreBtn")
more.append(moreBtn)
website.append(websiteBtn)

var newOfficial = $("<div class='card profile-card'>");
// creates a newOfficial variable with a class profile-card
var wrapper = $("<div class='imgwrapper'>");
// adds a div with a class imgwrapper and an id with the officials name
var officialPhoto = $("<img class='card-img-top img-fluid img-responsive profile-img'>")
.attr("src", photoSrc);
// adds a img with a class card-img-top img-fluid img-responsive profile-img and sets the
source to the officials photo from the civic information api
var officialBody = $("<div class='card-body text-center'>");
// adds a div with a class card-body text-center for formatting
officialBody.attr("id", nameID)
// adds name to the official body so that headlines can be appended to correct place
var officialName = $("<h5 class='card-title'>").text(name);
// adds a h5 with a class card-title with the officials name
var officialPosition = $("<p class='card-text text-muted'>").text(position + " --- " +
party);
// adds a p with a class card-text text-muted with the officials position and party

var officialSocialDiv = $("<div class='social-links'>");
officialSocialDiv.addClass(nameID)
officialSocialDiv.attr("id", name)
```

Dynamic “cards” -through javascript

Each time the API returns a name, a profile card is created.



But there was a catch!

To get their office, we had to use a different response....

```
{
  "name": "John Cornyn",
  "address": {
    "line1": "517 Hart Senate Office Building",
    "city": "Washington",
    "state": "DC",
    "zip": "20510"
  },
  "party": "Republican",
  "phones": {
    "(202) 224-2934"
  },
  "urls": {
    "http://www.cornyn.senate.gov/public/"
  },
  "photoUrl": "http://bioguide.congress.gov/bioguide/photo/C/C001056.jpg",
  "channels": [
    {
      "type": "Facebook",
      "id": "Sen.JohnCornyn"
    },
    {
      "type": "Twitter",
      "id": "JohnCornyn"
    },
    {
      "type": "YouTube",
      "id": "SenJohnCornyn"
    }
  ]
},
{
  "name": "Ted Cruz",
  "address": {
    "line1": "404 Russell Senate Office Building",
    "city": "Washington",
    "state": "DC",
    "zip": "20510"
  },
  "party": "Republican",
  "phones": {
    "(202) 224-5922"
  },
  "urls": {
    "http://www.cruz.senate.gov/"
  },
  "photoUrl": "http://www.cruz.senate.gov/files/images/OfficialPortrait.jpg",
  "channels": [
    {
      "type": "Facebook",
      "id": "SenatorTedCruz"
    },
    {
      "type": "Twitter",
      "id": "SenTedCruz"
    }
  ]
}
```

```
{
  "name": "United States Senate",
  "divisionId": "ocd-division/country:us/state:tx",
  "levels": [
    "country"
  ],
  "roles": [
    "legislatorUpperBody"
  ],
  "officialIndices": [
    2,
    3
  ]
}
```

Push to a new array!

```
$.ajax({
  url: queryURL,
  method: "GET"
}).then(function (response) {
  for (var k = 0; k < response.offices.length; k++) {
    var position = response.offices[k].name;
    // person's official position

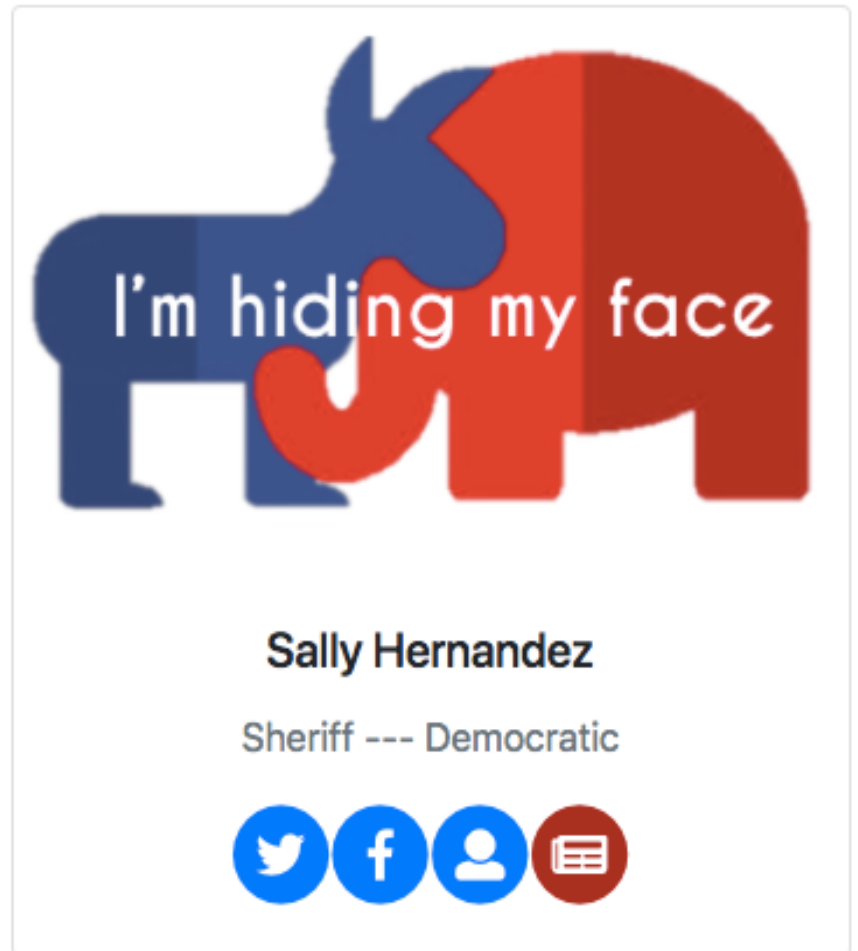
    var officialInd = [];
    // create an empty array

    for (var m = 0; m < response.offices[k].officialIndices.length; m++) {
      officialInd.push({
        position: position,
        index: response.offices[k].officialIndices[m]
      });
      // pushes an object into the empty array
    }
  }
})
```


Creating the profile

9 Elements exist within each profile:

1. ~~The card~~
2. A photo
3. ~~Their name~~
4. ~~Their office~~
5. ~~Their party~~
6. Twitter handle
7. Facebook page
8. Website
9. Related New York Times headlines



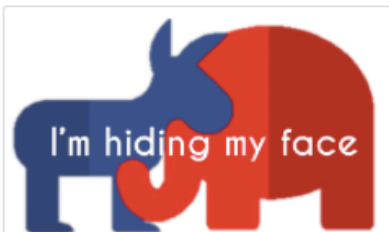
What IF they don't have a photo?





Ted Cruz

United States Senate --- Republican



Greg Abbott

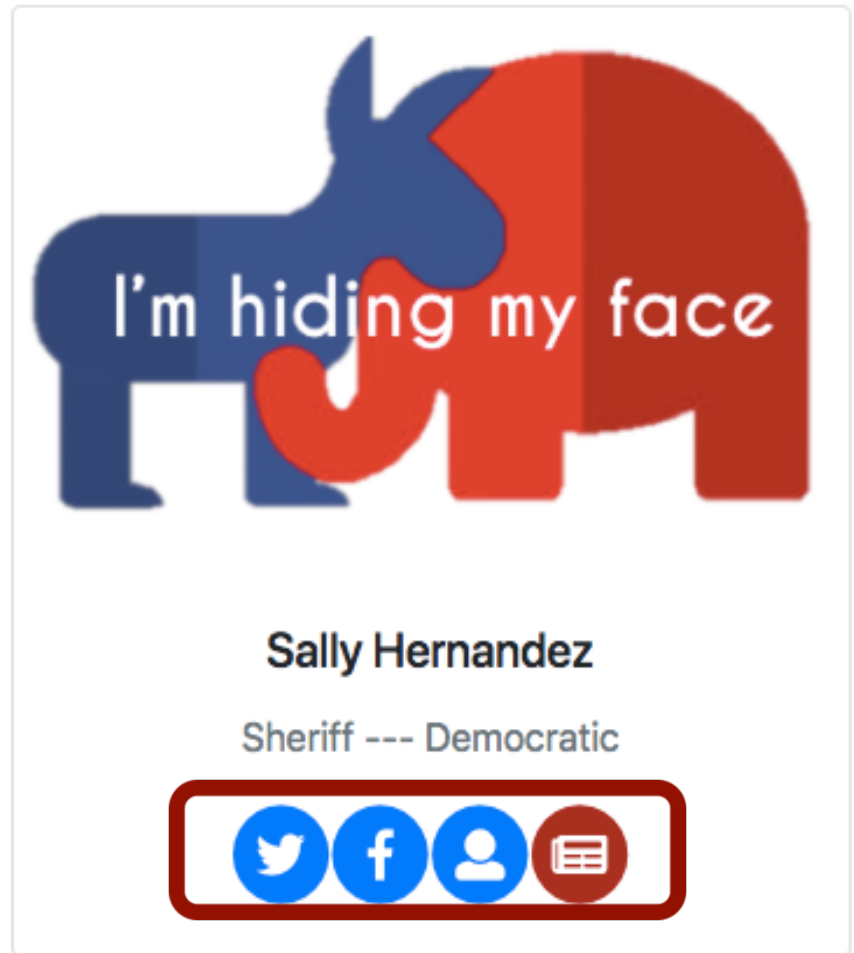
Governor --- Republican

```
if (!response.officials[indexofName].photoUrl) {  
    var photoSrc = "assets/images/nophoto.png"  
} else {  
    var photoSrc = photo  
}
```

4 little buttons left...

9 Elements exist within each profile:

1. ~~The card~~
2. ~~A photo~~
3. ~~Their name~~
4. ~~Their office~~
5. ~~Their party~~
6. Twitter handle
7. Facebook page
8. Website
9. Related New York Times headlines



Just + 4 + little buttons = famous last words!



```
var socialMedia = response.officials[indexofName].channels

if (!socialMedia) {
  // do nothing
} else {
  for (sm = 0; sm < socialMedia.length; sm++) {
    if (socialMedia[sm].type === "Twitter") {
      var twitterID = socialMedia[sm].id
    }
    if (socialMedia[sm].type === "Facebook") {
      var facebookID = socialMedia[sm].id
    }
  }
}

//*****Create social media buttons*****
```

```
function buttonGenerator(twitterID, facebookID, photoSrc, nameID, position, name,
  officialSocialDiv, officialURL) {
  var cardLinks = ["https://twitter.com/" + twitterID, "https://facebook.com/" +
    facebookID, officialURL]
  var websiteIconClasses = ["fab fa-twitter fa-stack-1x fa-inverse", "fab fa-facebook-f
    fa-stack-1x fa-inverse", "fa fa-user fa-stack-1x fa-inverse", "fa fa-newspaper
    fa-stack-1x fa-inverse"]

  for (var i = 0; i < 4; i++) {
    // For loop to cycle through the arrays above
    var websiteBtn = $("<span class='fa-stack fa-lg'>")
    var websiteURL = cardLinks[i] // Grabs the button's website from the cardLinks array
    var buttonCircle = $("<i>") // Creates the button circle background
    buttonCircle.addClass("fa fa-circle fa-stack-2x")
    var websiteIcon = $("<i>") // Creates the website icon
    websiteIcon.addClass(websiteIconClasses[i])
    websiteBtn.append(buttonCircle).append(websiteIcon) // Adds circle and icon to button

    if (i < 3) {
      var website = $("<a>") // Builds up the <a> tag to hold each button's link
      website.attr("href", websiteURL)
      website.attr("target", "_blank")
      website.append(websiteBtn)

      officialSocialDiv.append(website);
    } else {
      // Different button creation process for Headlines when hidden
      websiteBtn.addClass("headlines hidden")
      websiteBtn.attr("id", nameID)
      var website = $("<a>")
      website.append(websiteBtn)

      officialSocialDiv.append(website);
    }
  }
  // returns the div with added buttons back to the main body
  return officialSocialDiv
}
```

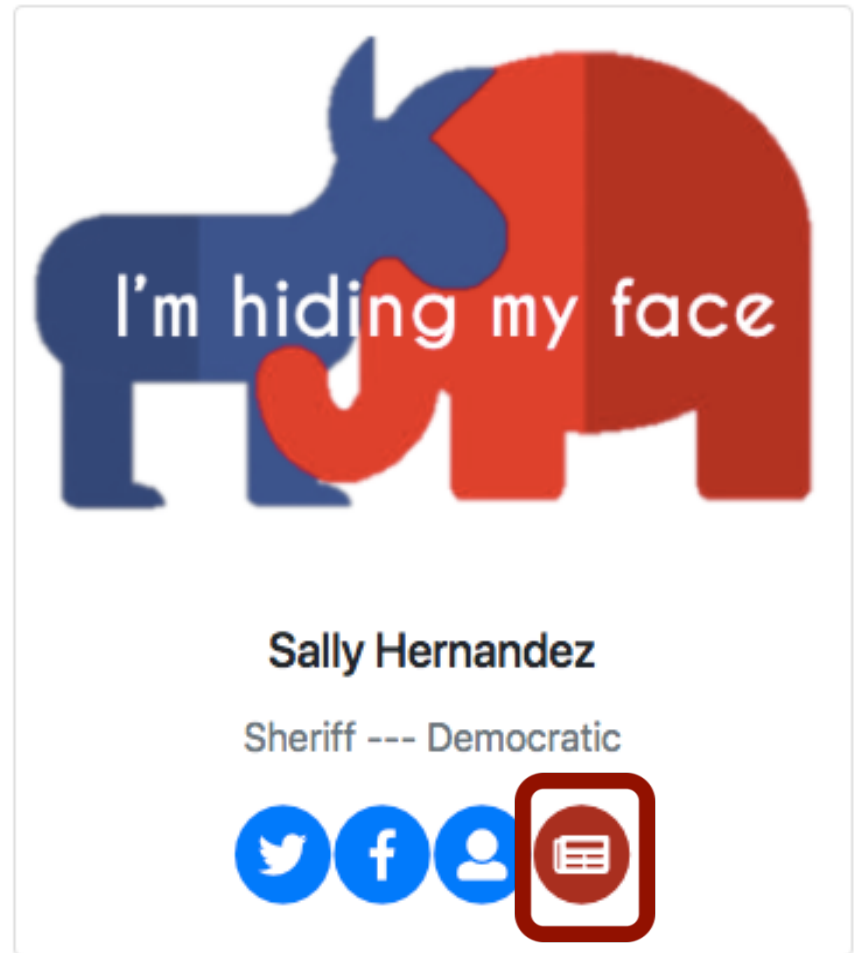
First we had to establish if they had social media... and then we had to make the button. Thanks FontAwesome.com

Now the real fun starts!

One API to go...

9 Elements exist within each profile:

1. ~~The card~~
2. ~~A photo~~
3. ~~Their name~~
4. ~~Their office~~
5. ~~Their party~~
6. ~~Twitter handle~~
7. ~~Facebook page~~
8. ~~Website~~
9. Related New York Times headlines



The “Headlines” Button

```
/**creating the headlines button
var moreBtn = $("
```

```
for (var n = 0; n < nameArr.length; n++) {
  if (nameArr[n] === ".") {
    nameArr.splice([n], 1)
  }
  if (nameArr[n] === " ") {
    nameArr[n] = "-";
  } else {
    nameArr[n] = nameArr[n]
  }
}

var nameID = nameArr.join("")
```

Donald J. Trump	main.js:64
▶ Array(14)	main.js:65
Donald-J-Trump	main.js:82

The names had to be split; remove punctuation; replace spaces with hyphens; and, finally, join the name. This gave us “nameID”. We used it as values and IDs to use in the next step of the API.

API #2 – New York Times

```
var cardClickedID = $(this).attr('id') // Grabs ID of specific
button's card clicked so headlines div can be added to the right card

// Creating div to hold list of headlines
var headlinesDiv = $("

- Now that we built “nameID” the headlines button can register the value to search in the NYT-API.
- By placing the “Get” within the button clicked function, we activate the 2nd API, and our page loads faster initially.


```

Remember: “Keep it clean”

```
if (headlinesDisplayed === false) {
  var cardClickedID = $(this).attr('id') // Grabs ID of
  specific button's card clicked so headlines div can be
  added to the right card

  // Creating div to hold list of headlines
  var headlinesDiv = $("

```
var headlinesDisplayed = false; // Switch to determine whether
to show or hide headlines on button click
$(".headlines").on("click", function () {

 var headlinesDivID = $(this).attr('id') + "headlines" //
 Grabs the ID of the specific button clicked
 if (headlinesDisplayed === false) {
 var cardClickedID = $(this).attr('id') // Grabs ID of
 specific button's card clicked so headlines div can be
 added to the right card
```



```
 $("##" + cardClickedID).append(headlinesDiv) //Appends
 the headlines div to the clicked-on card body
 $("##" + headlinesDivID).html(headlinesList) // Appends
 headlines to the newly created headlinesDiv
 headlinesDisplayed = true;
 } else {
 $("##" + headlinesDivID).empty()
 headlinesDisplayed = false;
 }
});
```



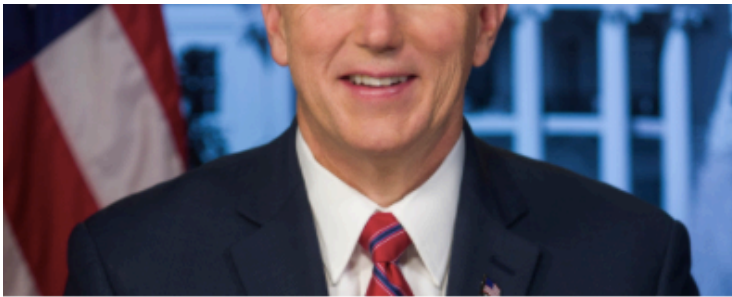
To stay true to the “keep it clean” branding style, the headlines button can clear the headlines by just being clicked again... but it took some code!



True & False!


```

Headlines button complete!



Mike Pence

Vice-President of the United States ---
Republican



- White House Job Requirement: Signing a Nondisclosure Agreement
- Children's Book About Mike Pence's Gay Bunny Jumps to No. 1



Mike Pence

Vice-President of the United States ---
Republican

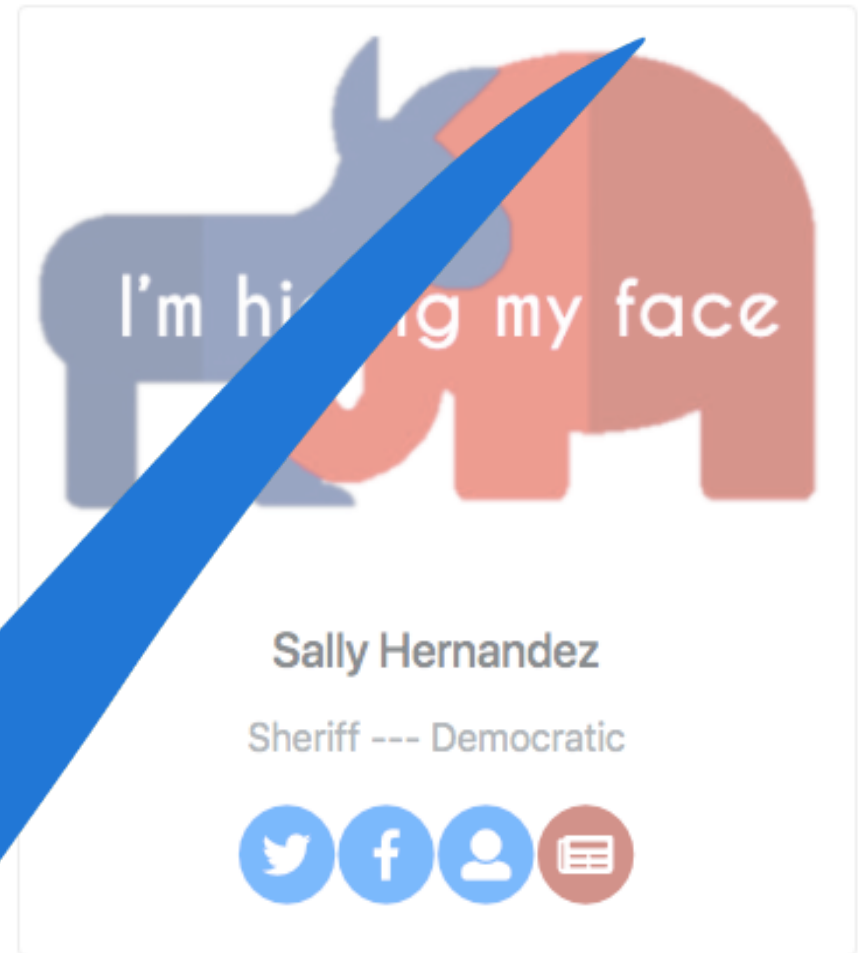
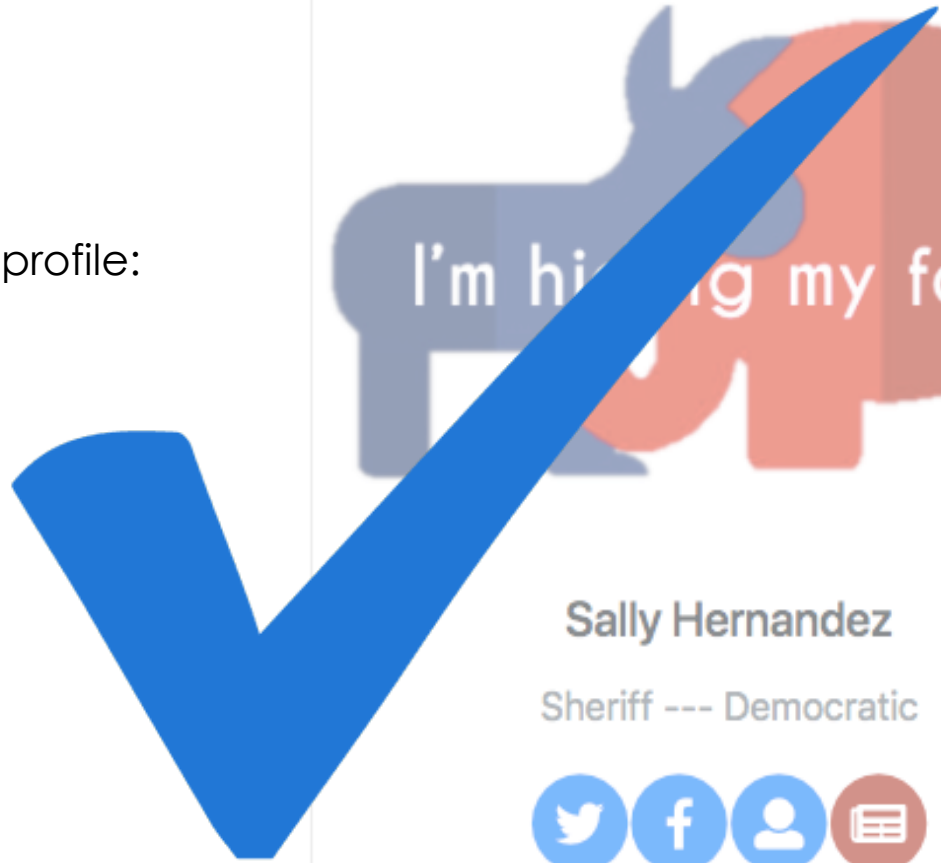


We did it!!!!

Clean & simple

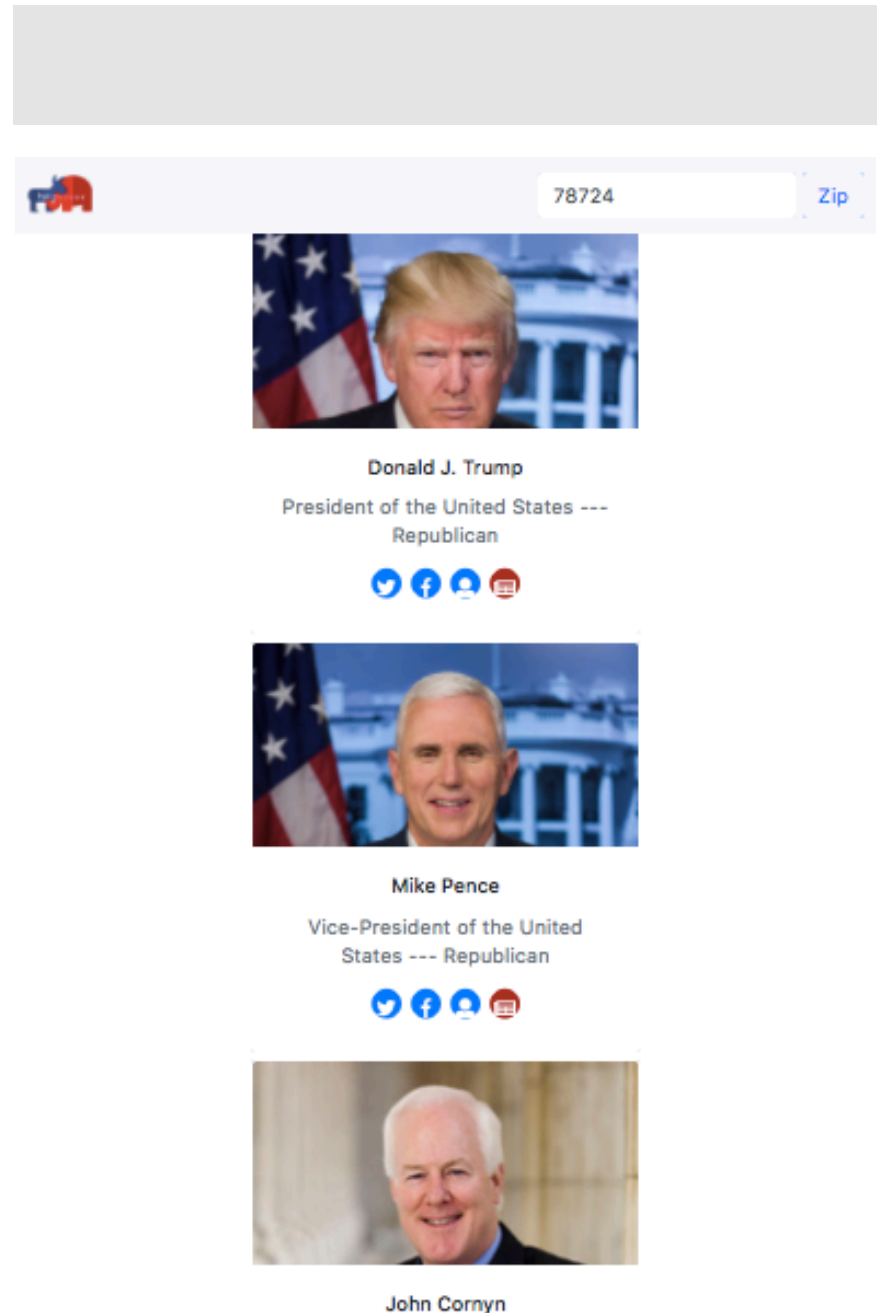
9 Elements exist within each profile:

1. ~~The card~~
2. ~~A photo~~
3. ~~Their name~~
4. ~~Their office~~
5. ~~Their party~~
6. ~~Twitter handle~~
7. ~~Facebook page~~
8. ~~Website~~
9. ~~Related New York Times headlines~~



Did we meet the criteria?

- ✓ 2 APIs - Google Civics & NYT
- ✓ Use Ajax
- ✓ New Library - FontAwesome.com
- ✓ Polished Front-end
- ✓ Quality code-standards
- ✓ No alerts - We used a modal
- ✓ Repeating element - The Profiles
- ✓ Use Bootstrap
- ✓ Deploy on GitHub
- ✓ User input validation - Zip Code Input



Who represents you?



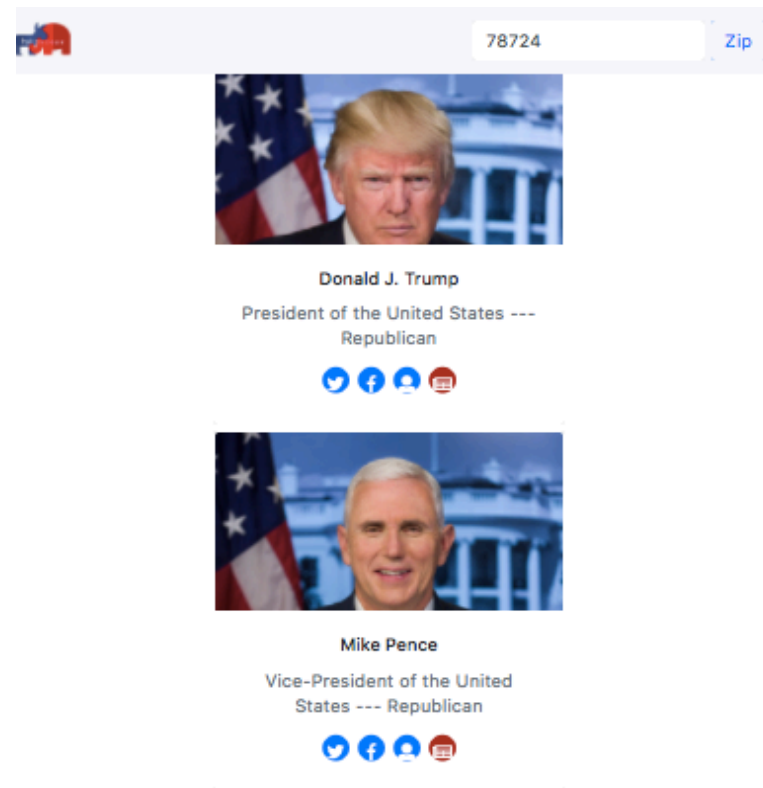
Type in your zip code and find out who your politicians are, their social media links, their website and what New York Times has to say about them.

Wanna see?

We know there is room to EXPAND!

Search-ability

- Create searches for special interest groups:
 - Form Letters
 - What to post, who to tag.
- Search by office, federal vs local, and other more specific queries.
- Create searches for upcoming elections:
 - Who is running?
 - Who is on the chopping block?
- Create user login –
 - Allow users to register their party and pull info based on that.



What is “sell-able”?

- Sell data to polling efforts.
- Sell advertising space on scrolling profiles.
- Let Special interest groups pay to host their form letters and campaigns.
- Let campaigns pay to post their invitations to rallies, speeches and other politically-motivated events.