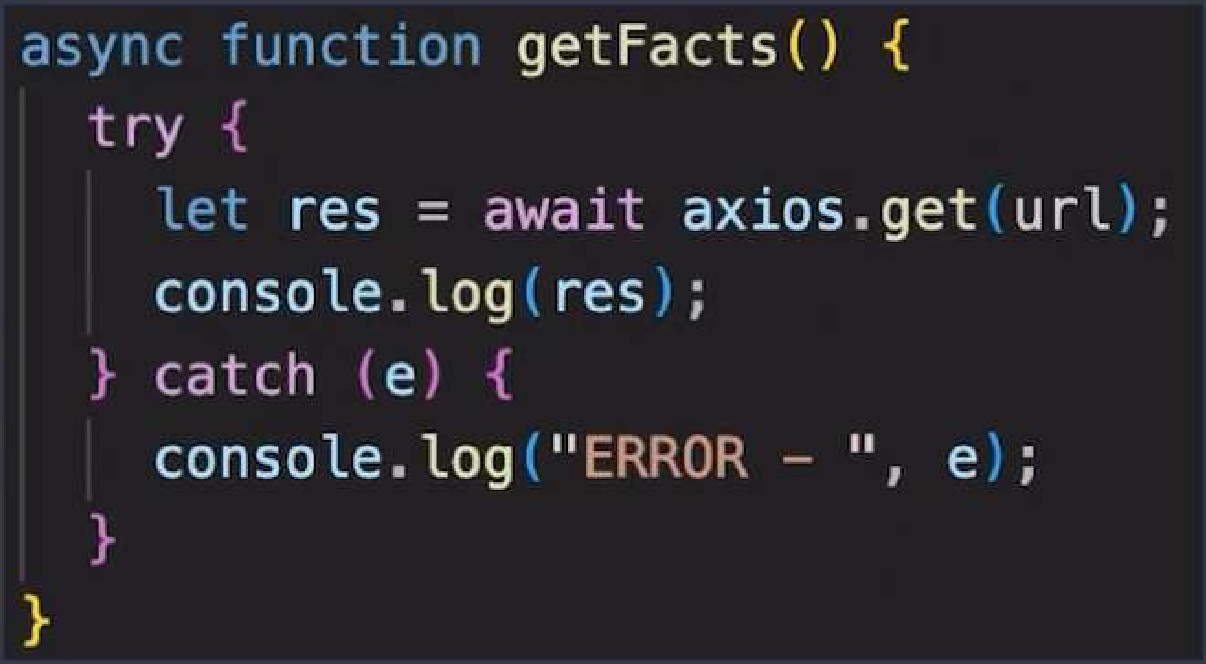
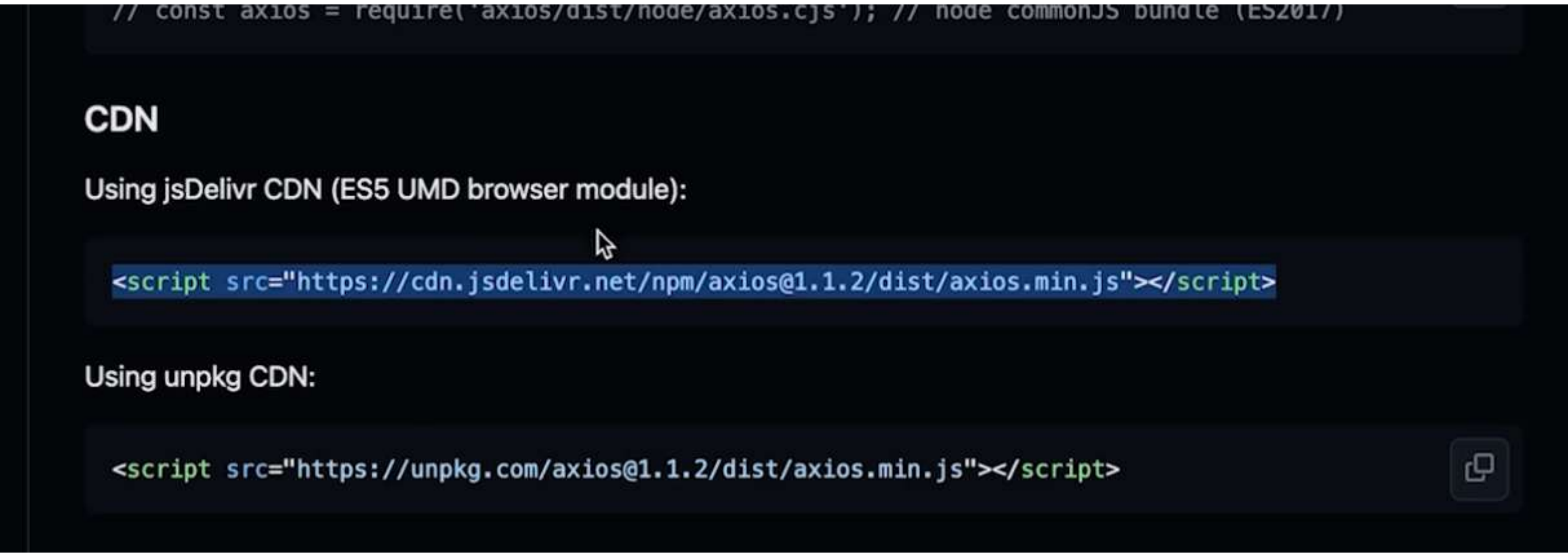
Axios

Library to make HTTP requests



```
async function getFacts() {
  try {
    let res = await axios.get(url);
    console.log(res);
  } catch (e) {
    console.log("ERROR - ", e);
  }
}
```



```
// const axios = require('axios/dist/node/axios.cjs'); // node commonJS bundle (ES2017)
```

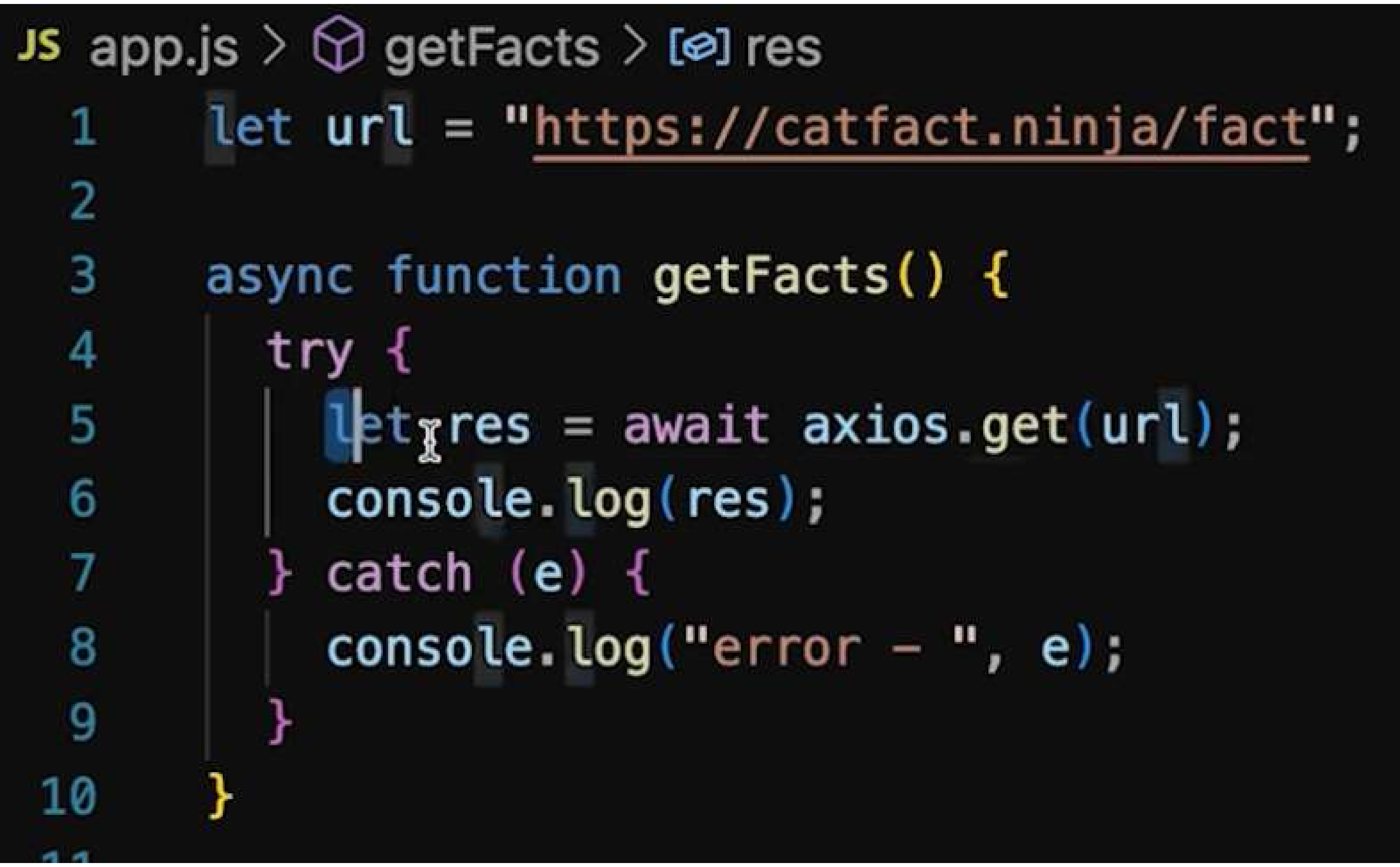
CDN

Using jsDelivr CDN (ES5 UMD browser module):

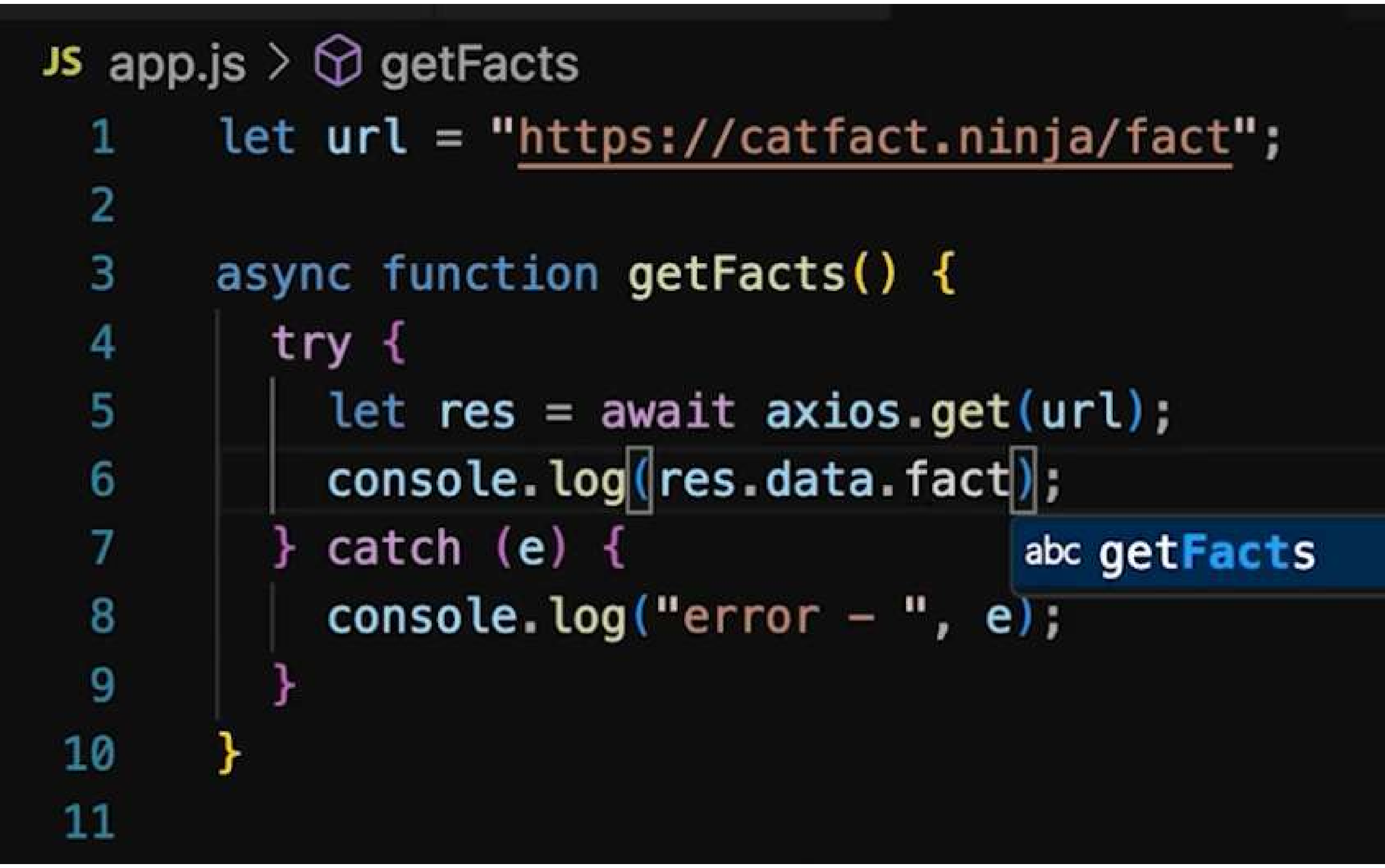
```
<script src="https://cdn.jsdelivr.net/npm/axios@1.1.2/dist/axios.min.js"></script>
```

Using unpkg CDN:

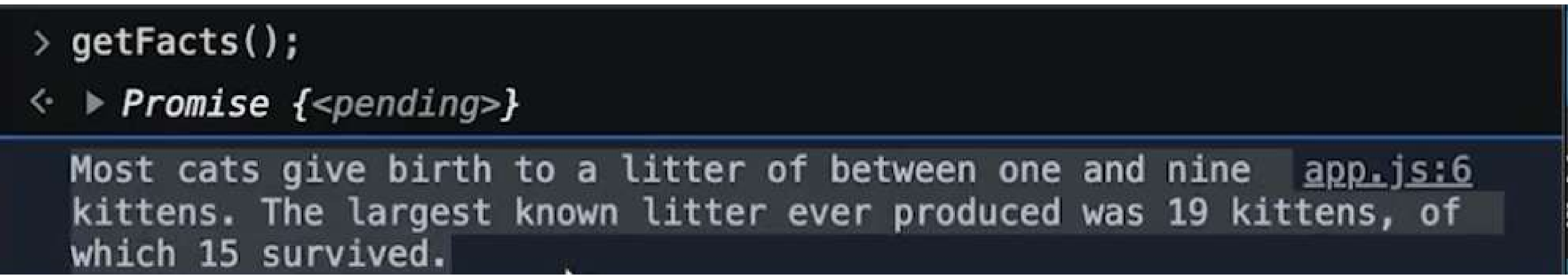
```
<script src="https://unpkg.com/axios@1.1.2/dist/axios.min.js"></script>
```



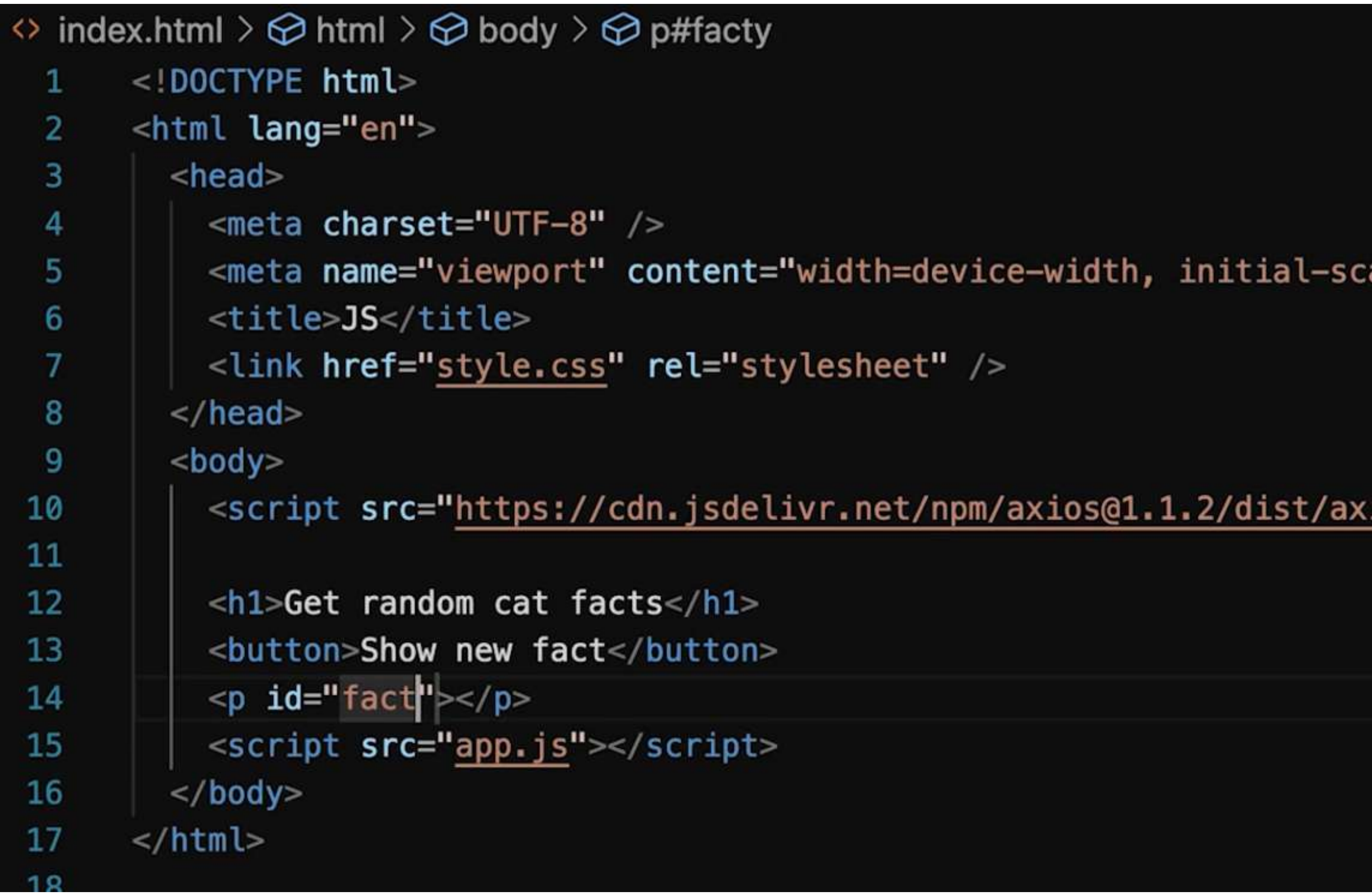
```
let url = "https://catfact.ninja/fact";

async function getFacts() {
  try {
    let res = await axios.get(url);
    console.log(res);
  } catch (e) {
    console.log("error - ", e);
  }
}
```



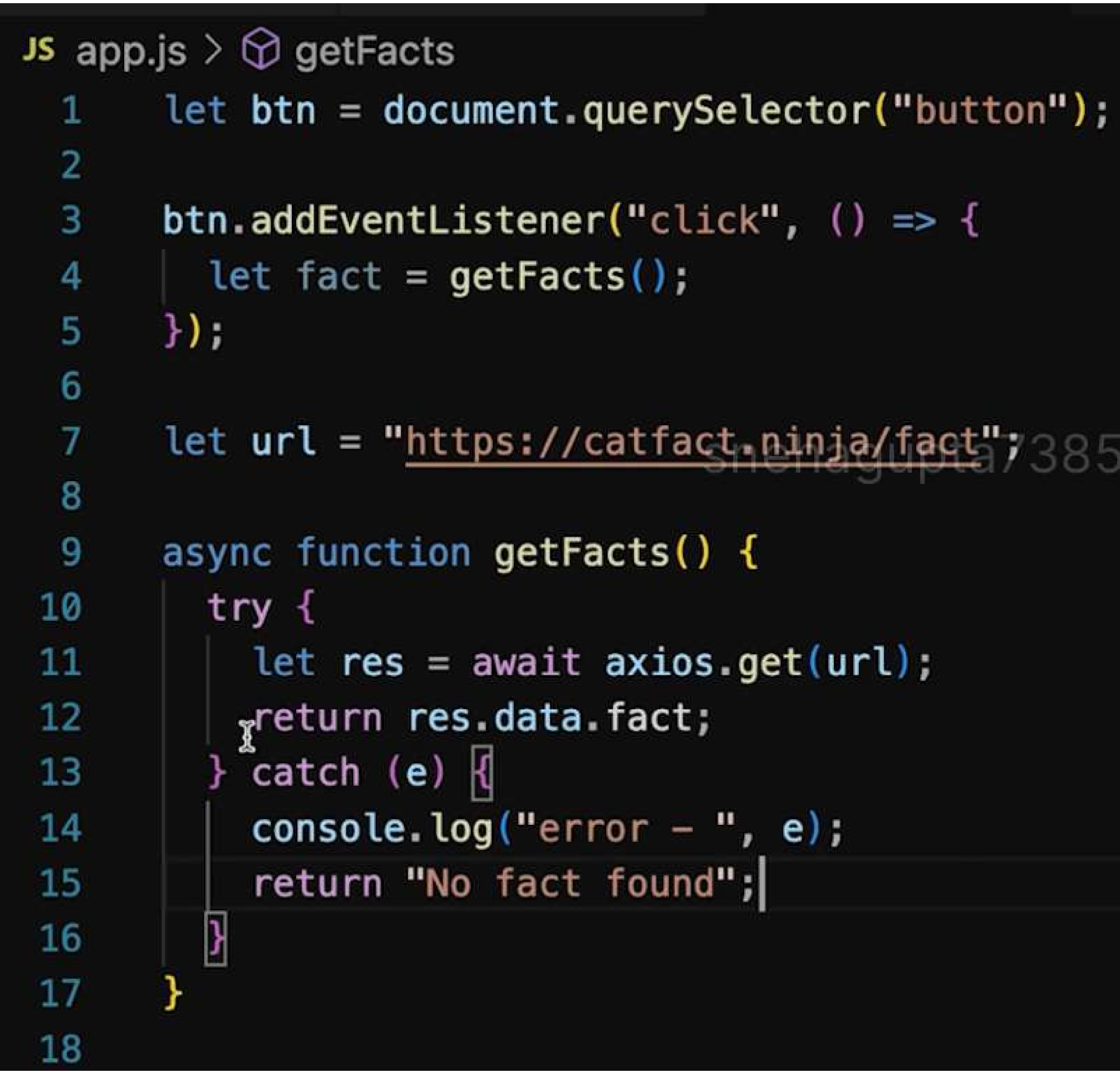
```
let url = "https://catfact.ninja/fact";

async function getFacts() {
  try {
    let res = await axios.get(url);
    console.log(res.data.fact);
  } catch (e) {
    console.log("error - ", e);
  }
}

```



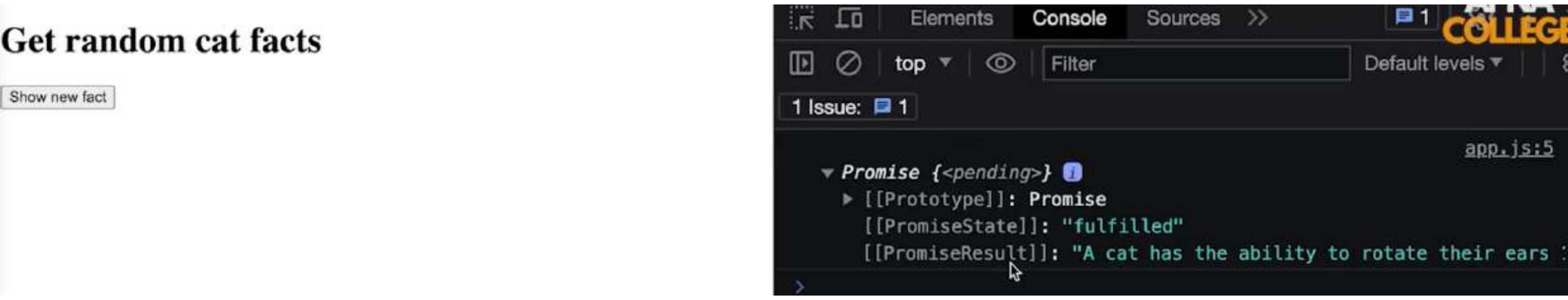
```
> getFacts();
<· ▶ Promise {<pending>}
  Most cats give birth to a litter of between one and nine  app.js:6
  kittens. The largest known litter ever produced was 19 kittens, of
  which 15 survived.
```



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-sca
    <title>JS</title>
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <script src="https://cdn.jsdelivr.net/npm/axios@1.1.2/dist/ax

    <h1>Get random cat facts</h1>
    <button>Show new fact</button>
    <p id="fact"></p>
    <script src="app.js"></script>
  </body>
</html>
```

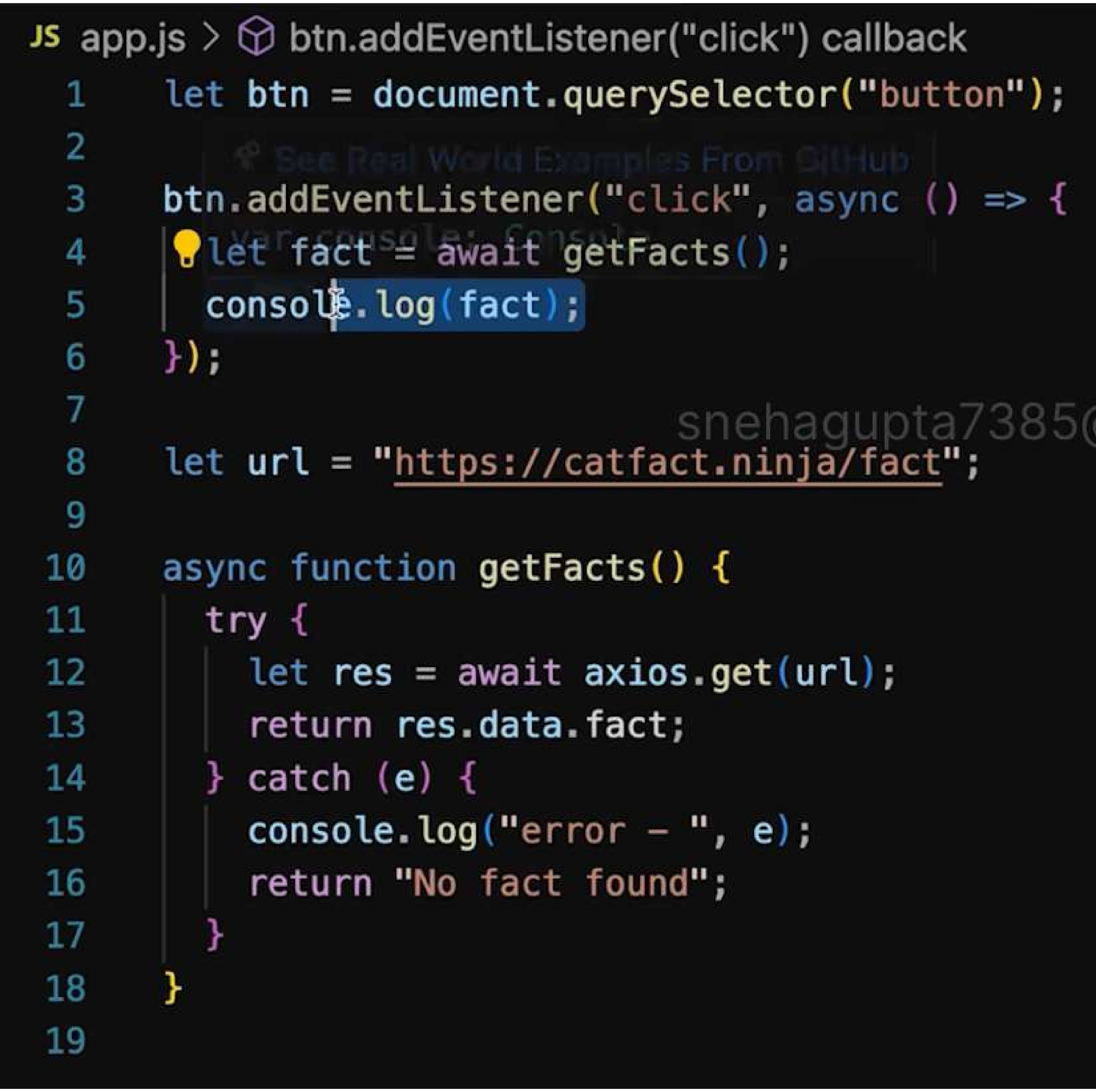


```javascript
let btn = document.querySelector("button");

btn.addEventListener("click", () => {
  let fact = getFacts();
});

let url = "https://catfact.ninja/fact";

async function getFacts() {
  try {
    let res = await axios.get(url);
    return res.data.fact;
  } catch (e) {
    console.log("error - ", e);
    return "No fact found";
  }
}

```

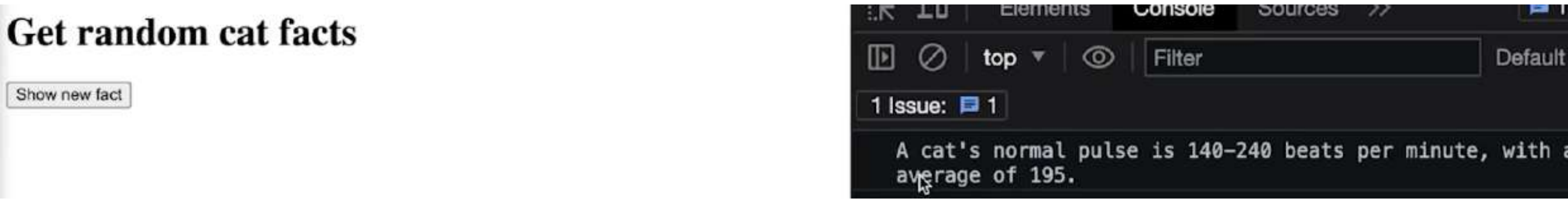


Get random cat facts

```
                                                                          app.js:5
  ▼ Promise {<pending>}
    ▶ [[Prototype]]: Promise
      [[PromiseState]]: "fulfilled"
      [[PromiseResult]]: "A cat has the ability to rotate their ears
>
```

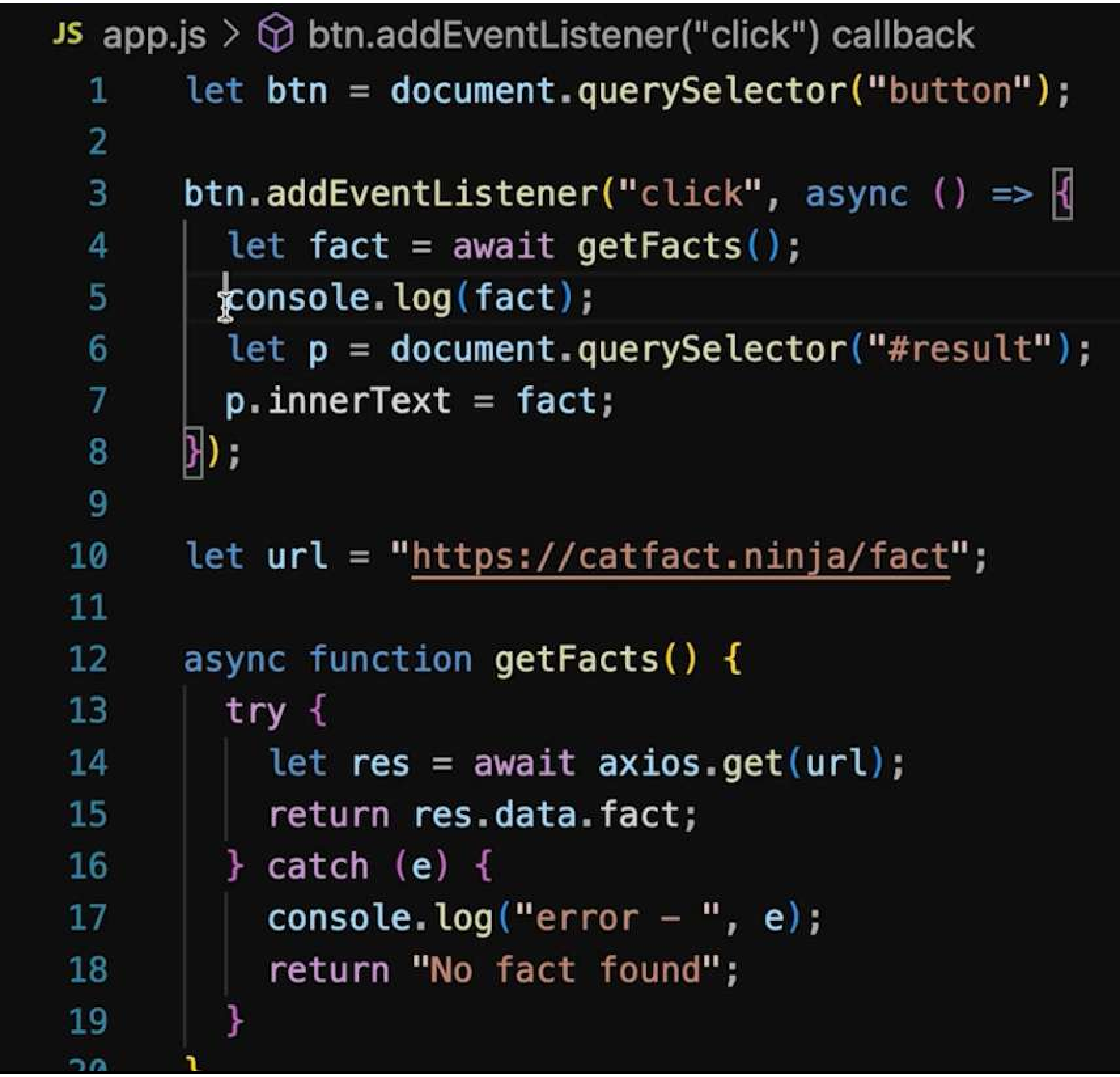

JS app.js > btn.addEventListener("click") callback
See Real World Examples From GitHub
var console: Console

```javascript
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let fact = await getFacts();
  console.log(fact);
});

let url = "https://catfact.ninja/fact";

async function getFacts() {
  try {
    let res = await axios.get(url);
    return res.data.fact;
  } catch (e) {
    console.log("error - ", e);
    return "No fact found";
  }
}
```

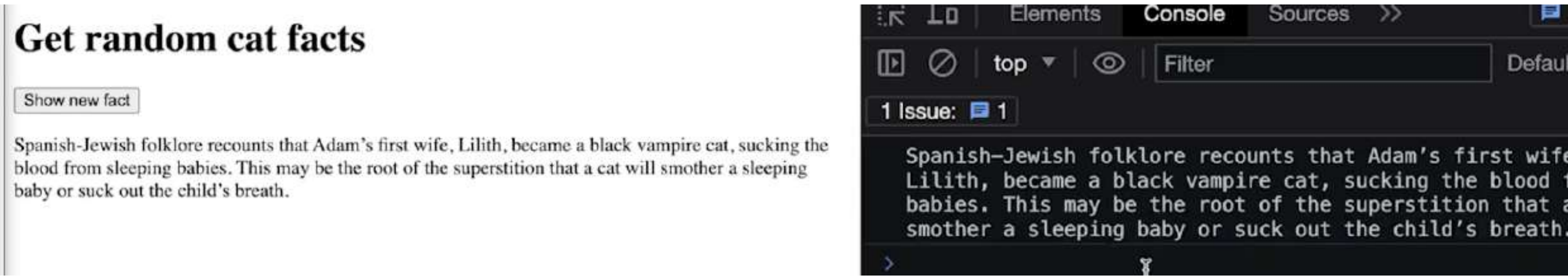


Get random cat facts

```
A cat's normal pulse is 140-240 beats per minute, with a
average of 195.
```



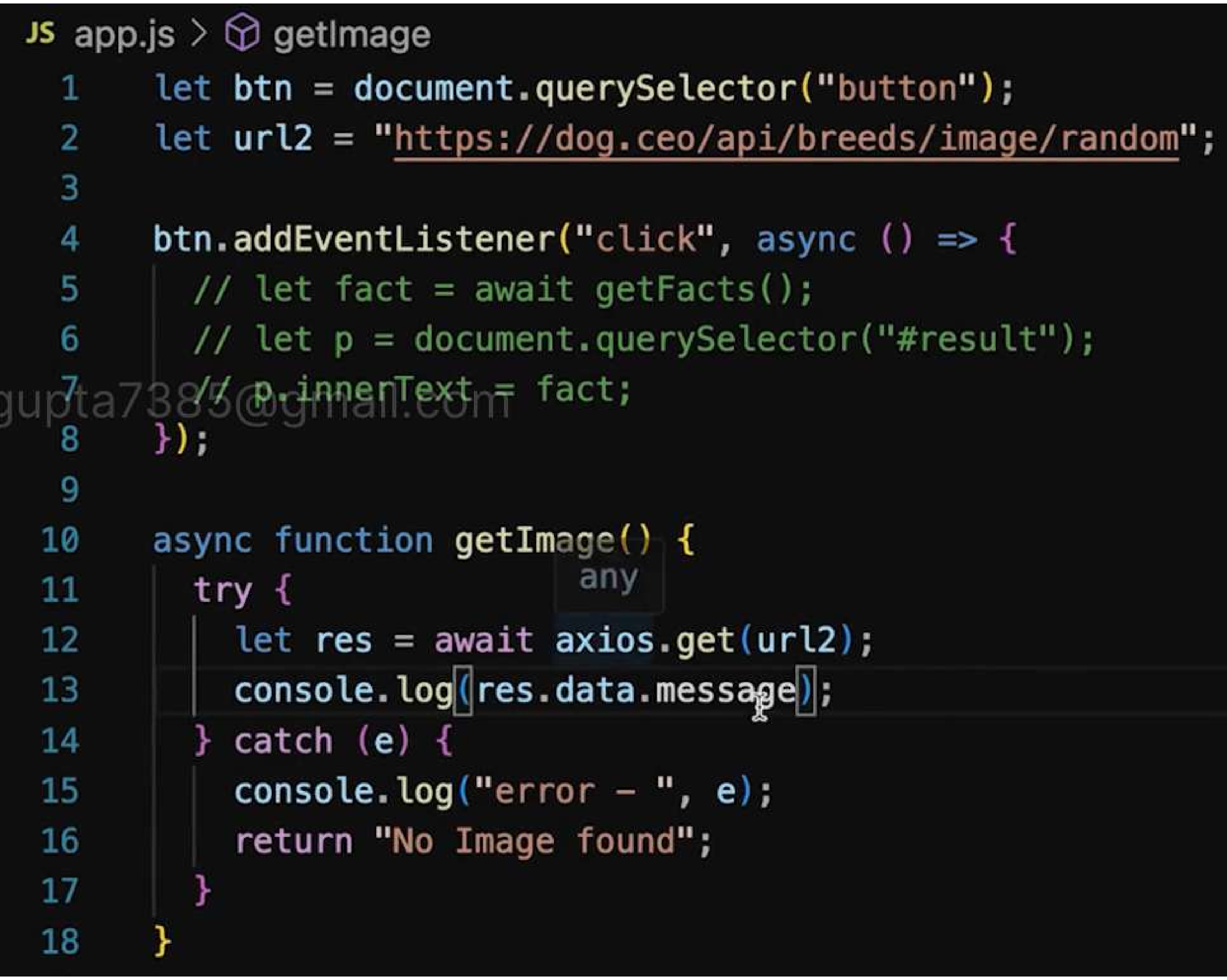
```javascript
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let fact = await getFacts();
  console.log(fact);
  let p = document.querySelector("#result");
  p.innerText = fact;
});

let url = "https://catfact.ninja/fact";

async function getFacts() {
  try {
    let res = await axios.get(url);
    return res.data.fact;
  } catch (e) {
    console.log("error - ", e);
    return "No fact found";
  }
}
```



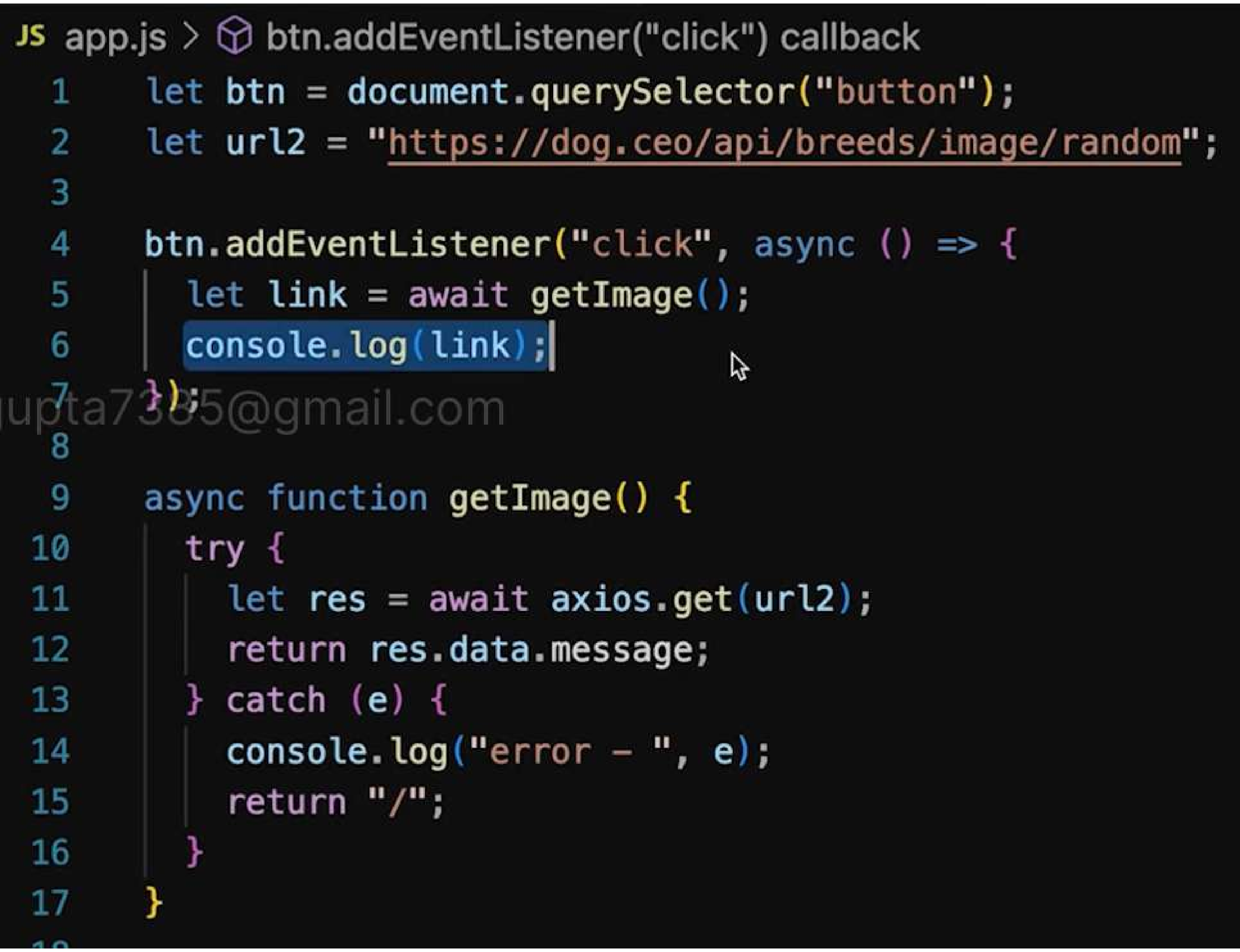
Get random cat facts

Spanish-Jewish folklore recounts that Adam’s first wife, Lilith, became a black vampire cat, sucking the blood from sleeping babies. This may be the root of the superstition that a cat will smother a sleeping baby or suck out the child’s breath.

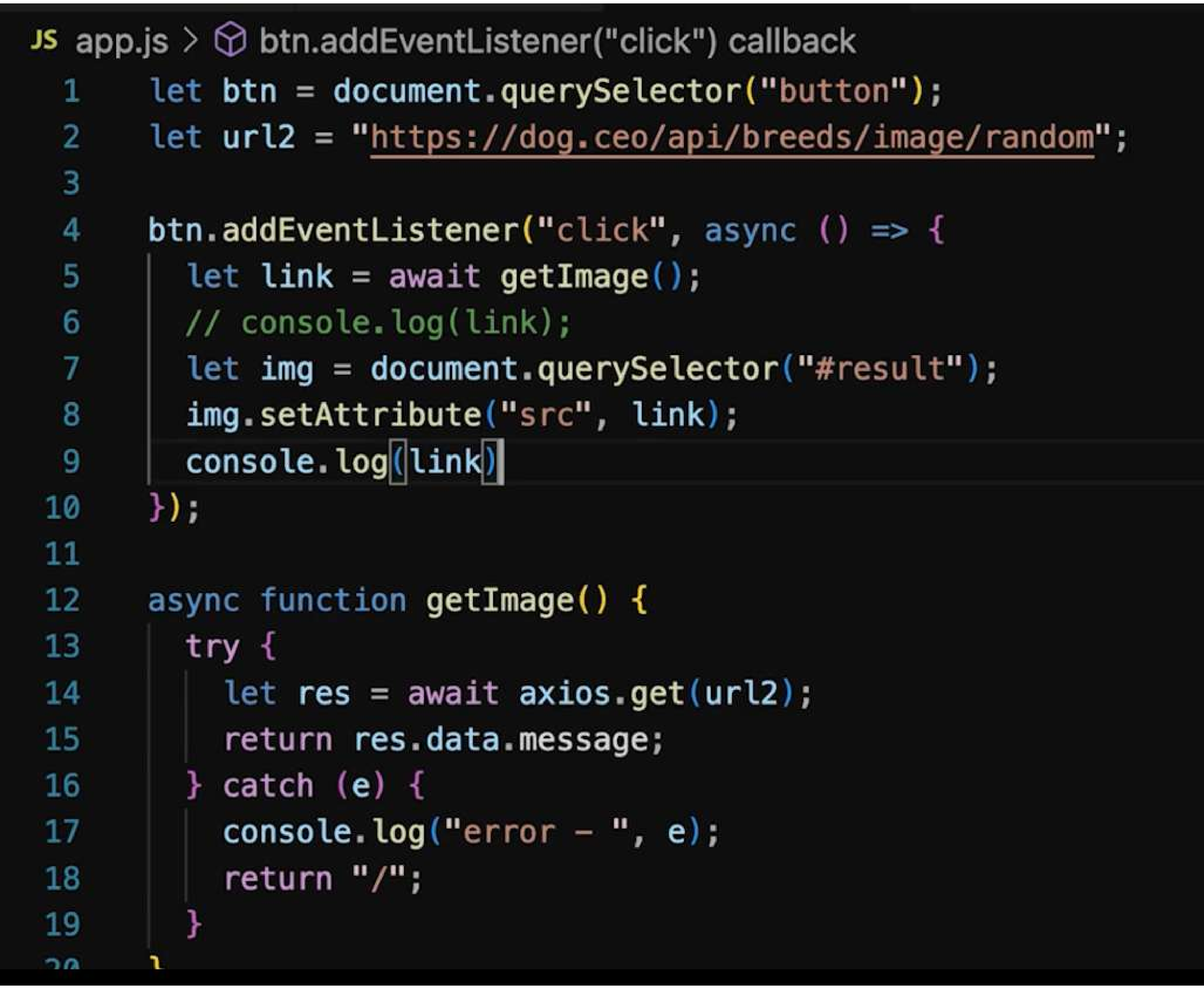
```
Spanish—Jewish folklore recounts that Adam’s first wife
Lilith, became a black vampire cat, sucking the blood f
babies. This may be the root of the superstition that a
smother a sleeping baby or suck out the child’s breath.
```



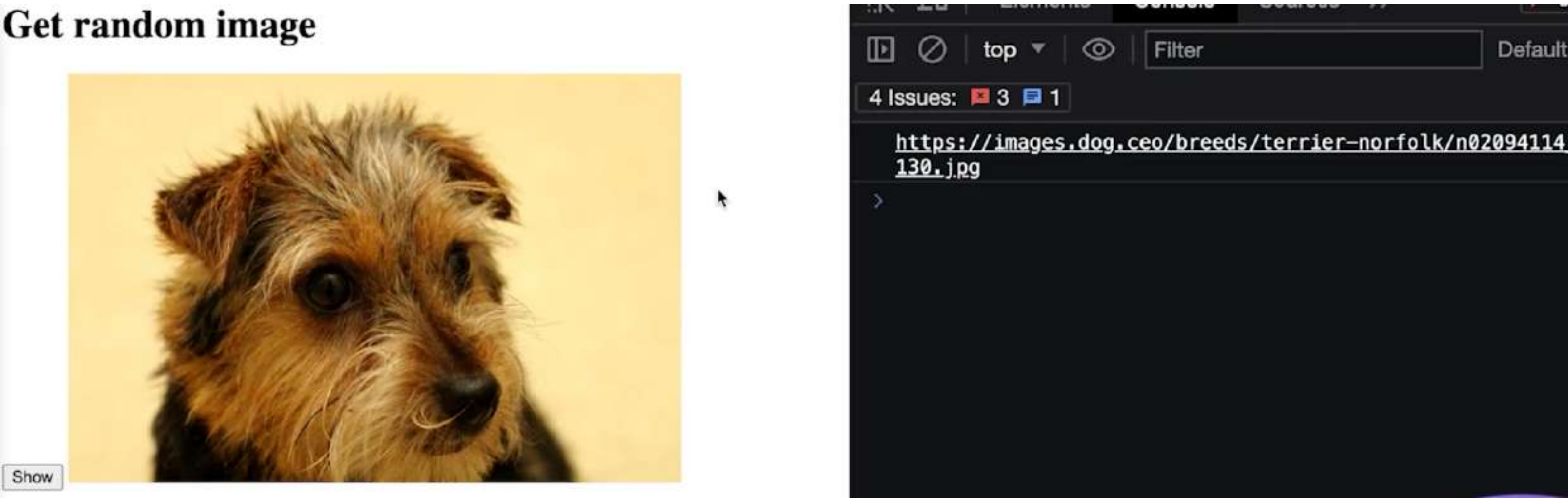
```javascript
let btn = document.querySelector("button");
let url2 = "https://dog.ceo/api/breeds/image/random";

btn.addEventListener("click", async () => {
  // let fact = await getFacts();
  // let p = document.querySelector("#result");
  // p.innerText = fact;
});

async function getImage() {
  try {
    let res = await axios.get(url2);
    console.log(res.data.message);
  } catch (e) {
    console.log("error - ", e);
    return "No Image found";
  }
}
```



```javascript
let btn = document.querySelector("button");
let url2 = "https://dog.ceo/api/breeds/image/random";

btn.addEventListener("click", async () => {
  let link = await getImage();
  console.log(link);
});

async function getImage() {
  try {
    let res = await axios.get(url2);
    return res.data.message;
  } catch (e) {
    console.log("error - ", e);
    return "/";
  }
}
```

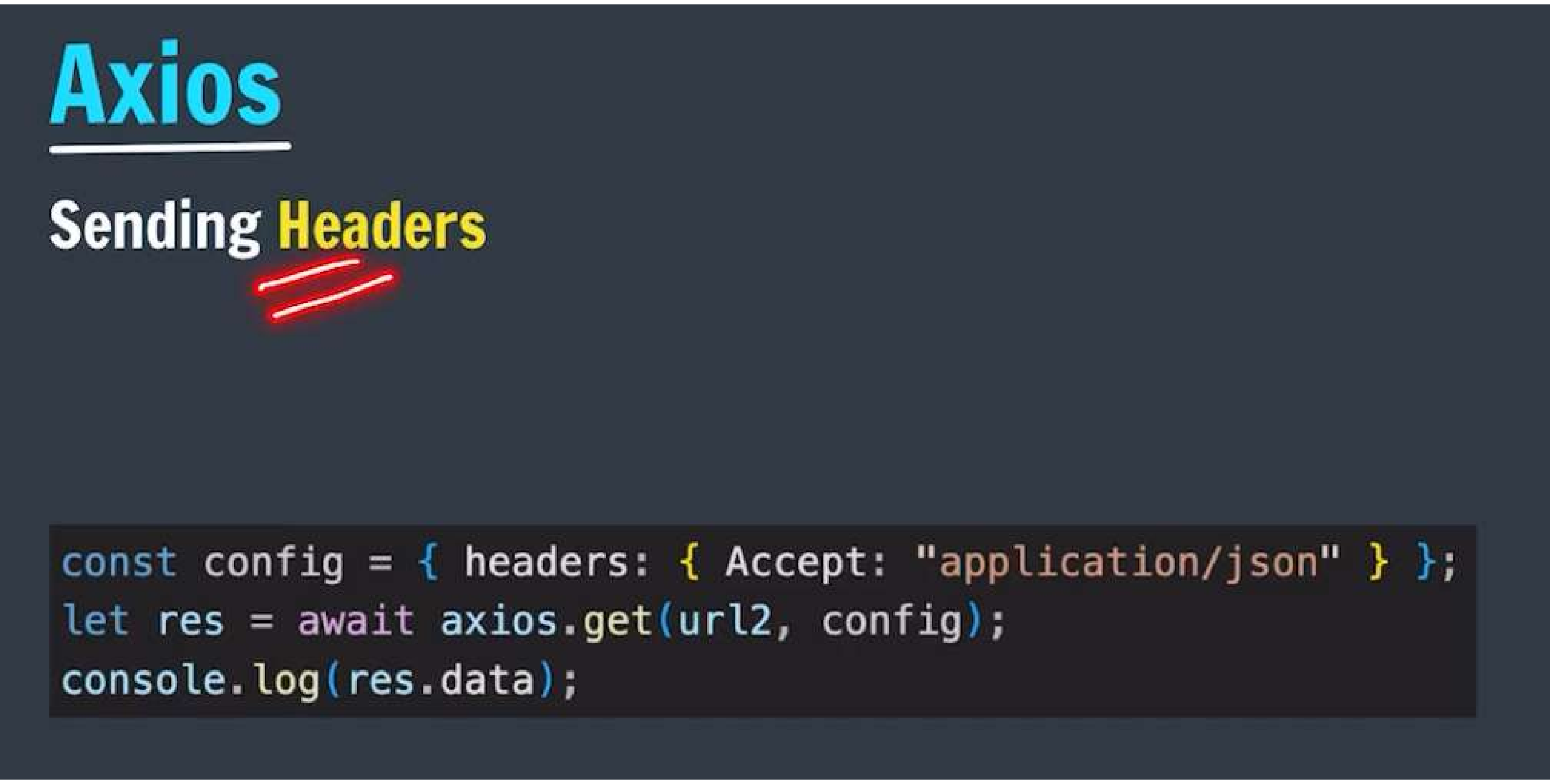


```javascript
let btn = document.querySelector("button");
let url2 = "https://dog.ceo/api/breeds/image/random";

btn.addEventListener("click", async () => {
  let link = await getImage();
  // console.log(link);
  let img = document.querySelector("#result");
  img.setAttribute("src", link);
  console.log(link)
});

async function getImage() {
  try {
    let res = await axios.get(url2);
    return res.data.message;
  } catch (e) {
    console.log("error - ", e);
    return "/";
  }
}
```



Get random image

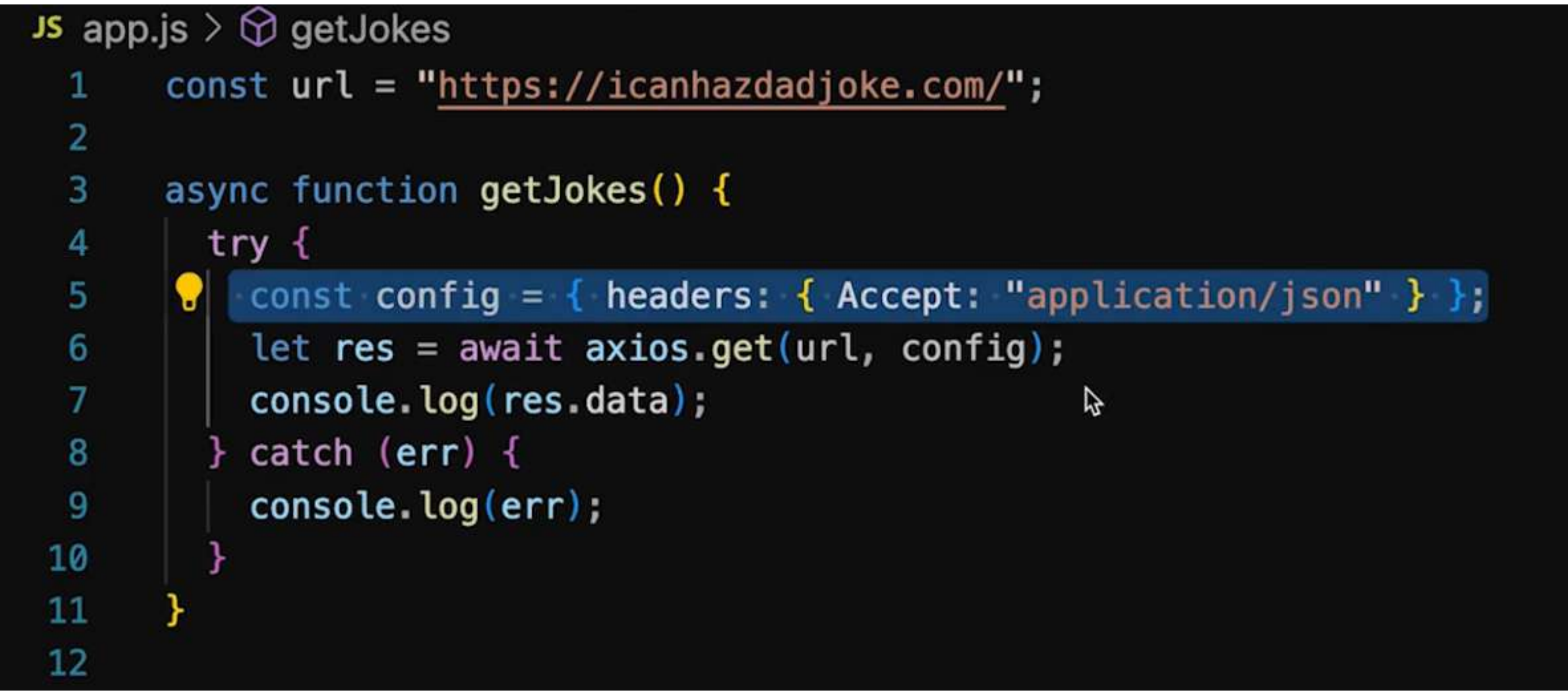
```
https://images.dog.ceo/breeds/terrier-norfolk/n02094114
130.jpg
```



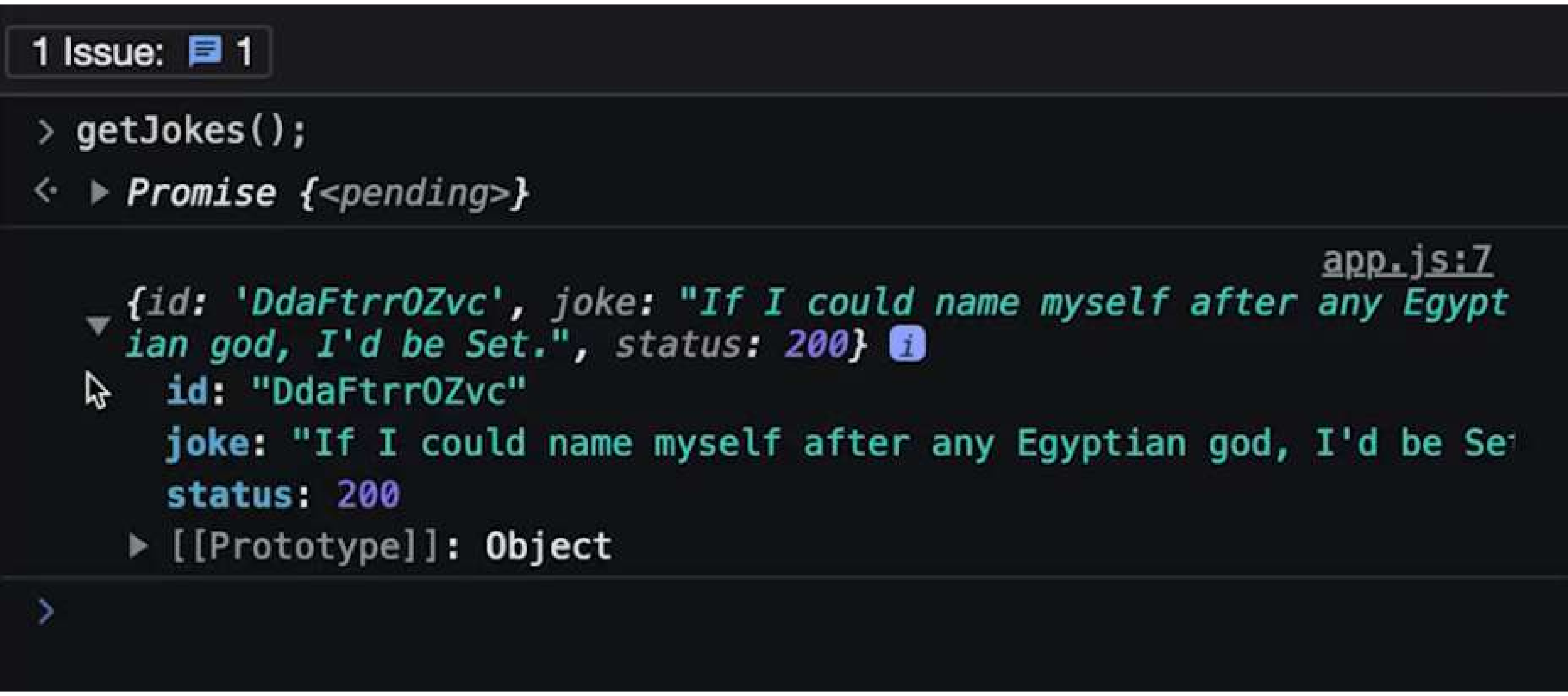
Axios

Sending Headers

```
const config = { headers: { Accept: "application/json" } };
let res = await axios.get(url2, config);
console.log(res.data);
```



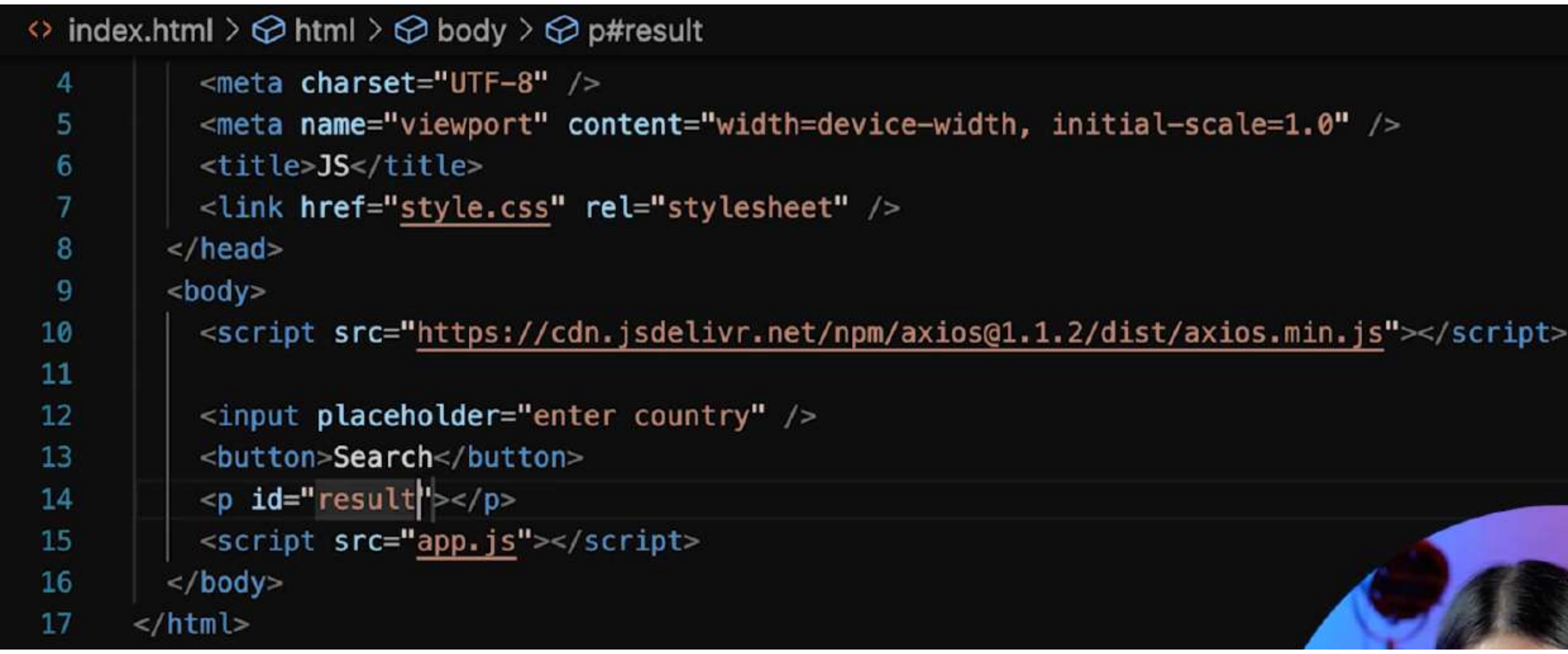
```javascript
const url = "https://icanhazdadjoke.com/";

async function getJokes() {
  try {
    const config = { headers: { Accept: "application/json" } };
    let res = await axios.get(url, config);
    console.log(res.data);
  } catch (err) {
    console.log(err);
  }
}

```



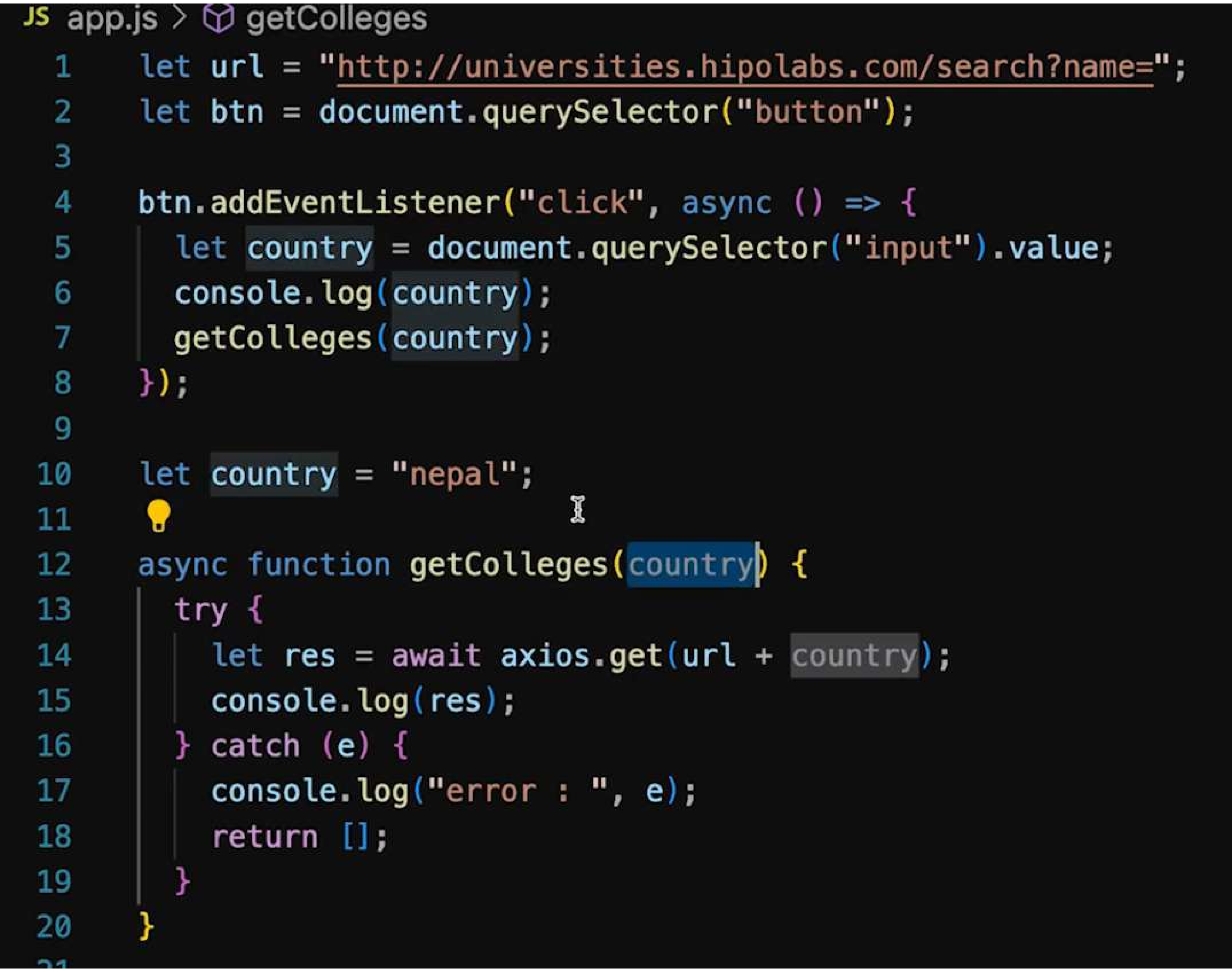
```
1 Issue:  1
> getJokes();
<· ▶ Promise {<pending>}
                                                                app.js:7
  ▼ {id: 'DdaFtrrOZvc', joke: "If I could name myself after any Egypt
     ian god, I'd be Set.", status: 200} i
       id: "DdaFtrrOZvc"
       joke: "If I could name myself after any Egyptian god, I'd be Se
       status: 200
     ▶ [[Prototype]]: Object
>
```

Axios

Updating Query Strings



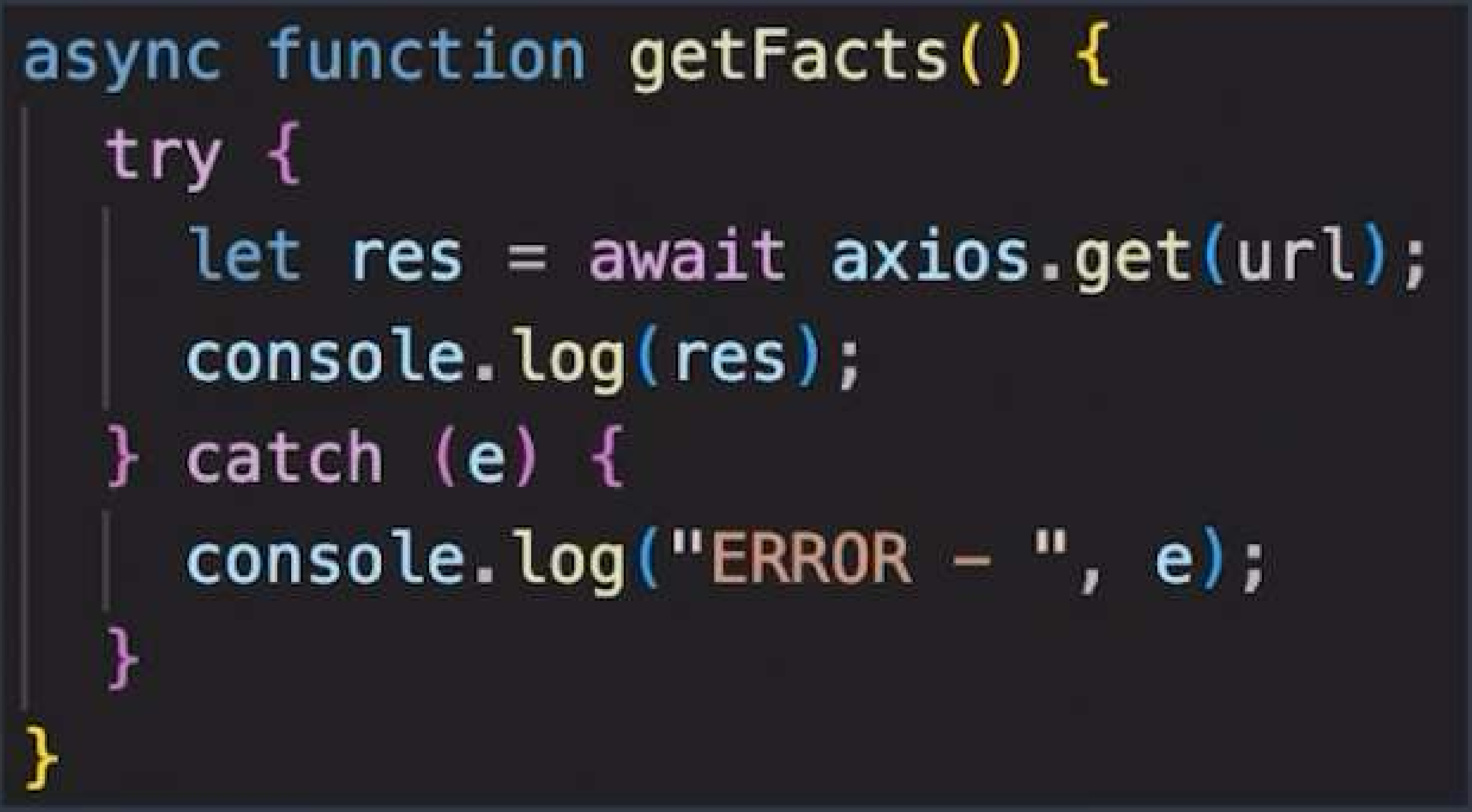
```html
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JS</title>
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <script src="https://cdn.jsdelivr.net/npm/axios@1.1.2/dist/axios.min.js"></script>

    <input placeholder="enter country" />
    <button>Search</button>
    <p id="result"></p>
    <script src="app.js"></script>
  </body>
</html>
```



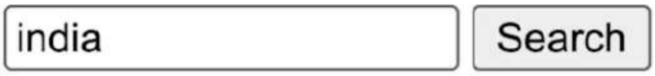
```javascript
let url = "http://universities.hipolabs.com/search?name=";
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let country = document.querySelector("input").value;
  console.log(country);
  getColleges(country);
});

let country = "nepal";

async function getColleges(country) {
  try {
    let res = await axios.get(url + country);
    console.log(res);
  } catch (e) {
    console.log("error : ", e);
    return [];
  }
}
```

Axios

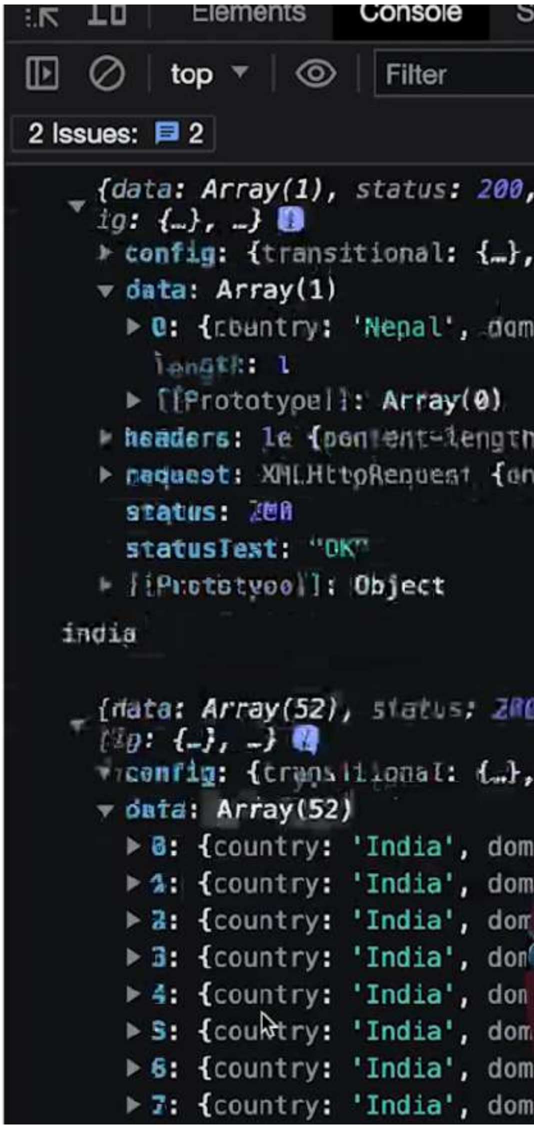
Library to make HTTP requests



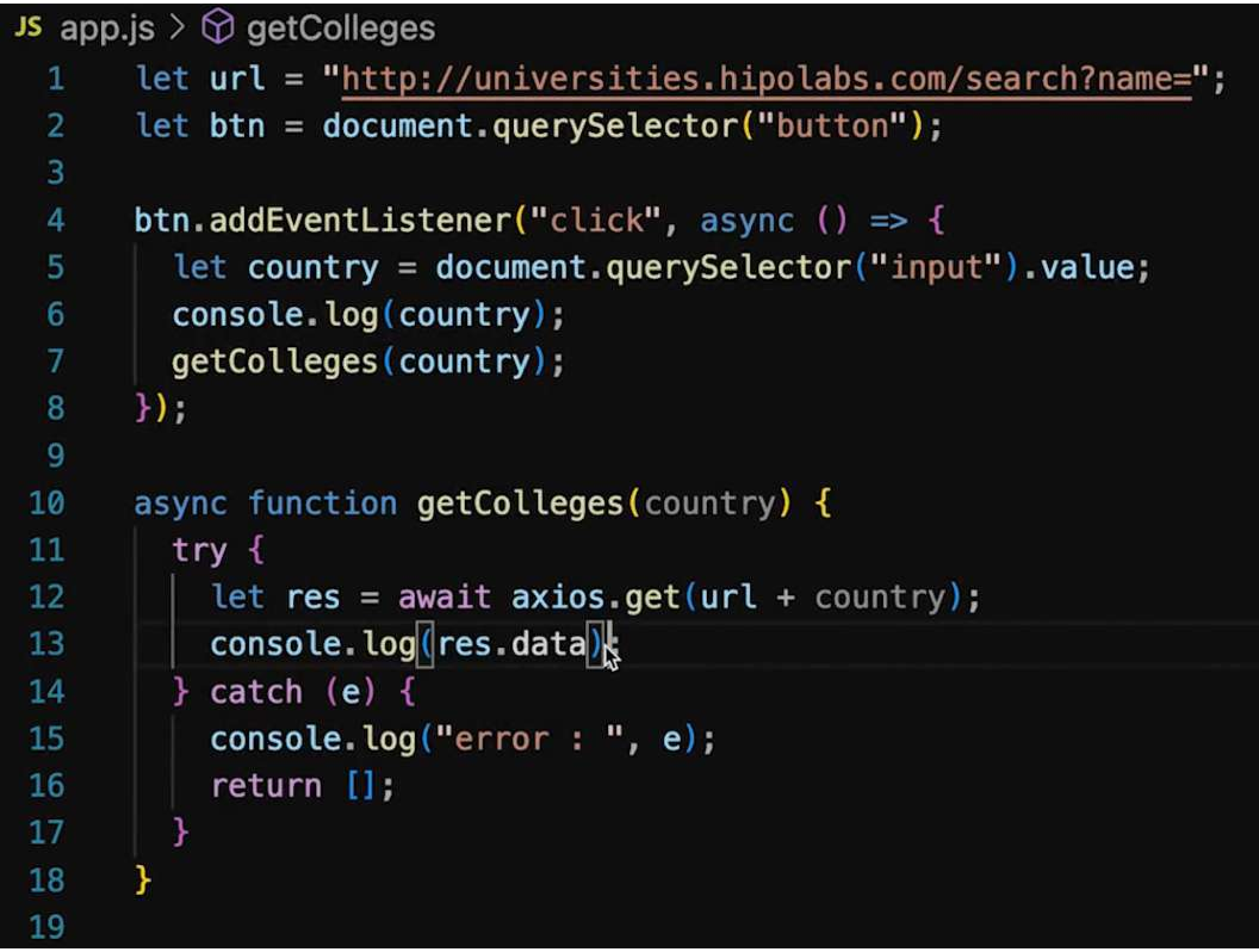
```
async function getFacts() {
  try {
    let res = await axios.get(url);
    console.log(res);
  } catch (e) {
    console.log("ERROR - ", e);
  }
}
```



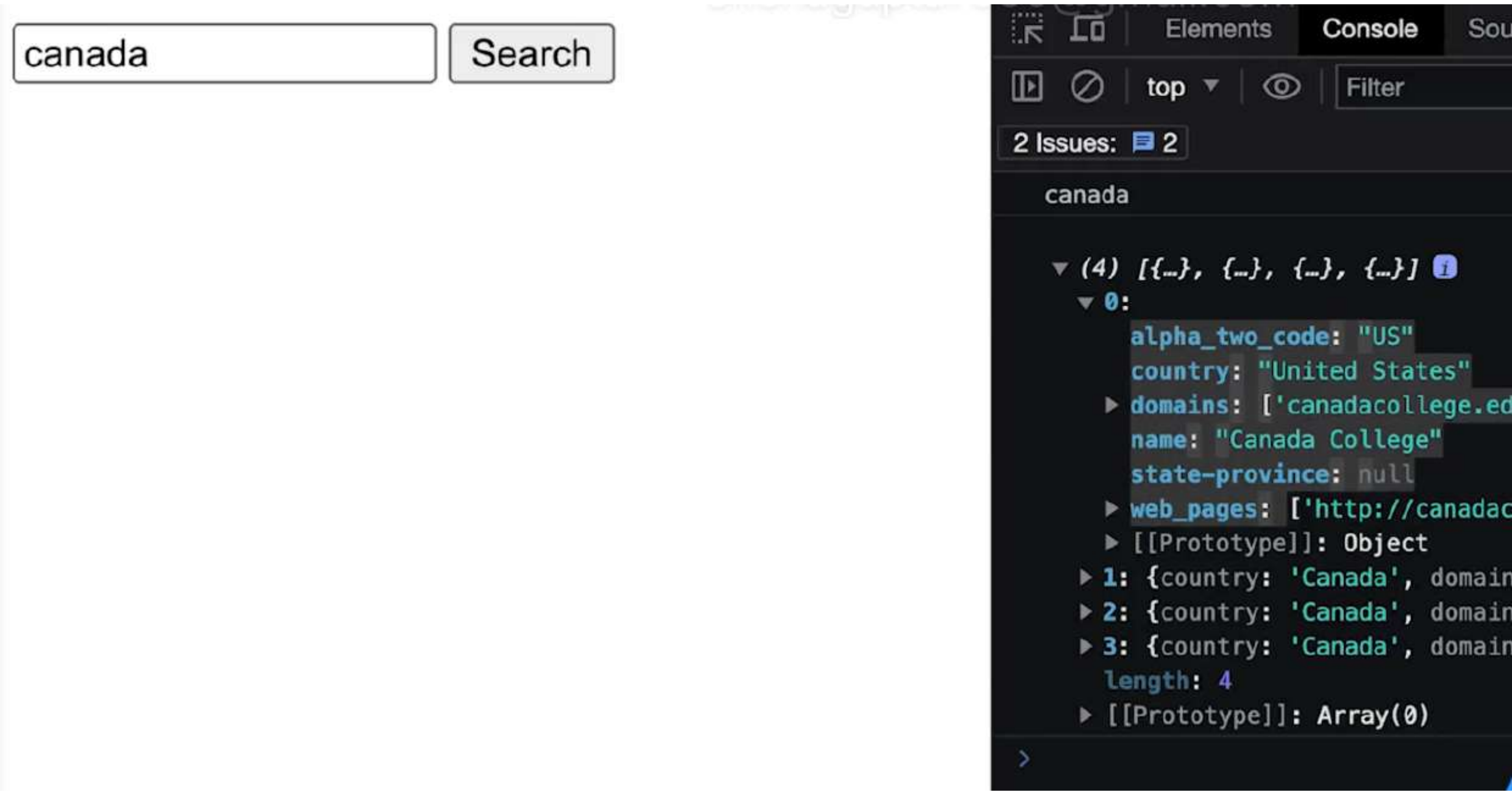
india



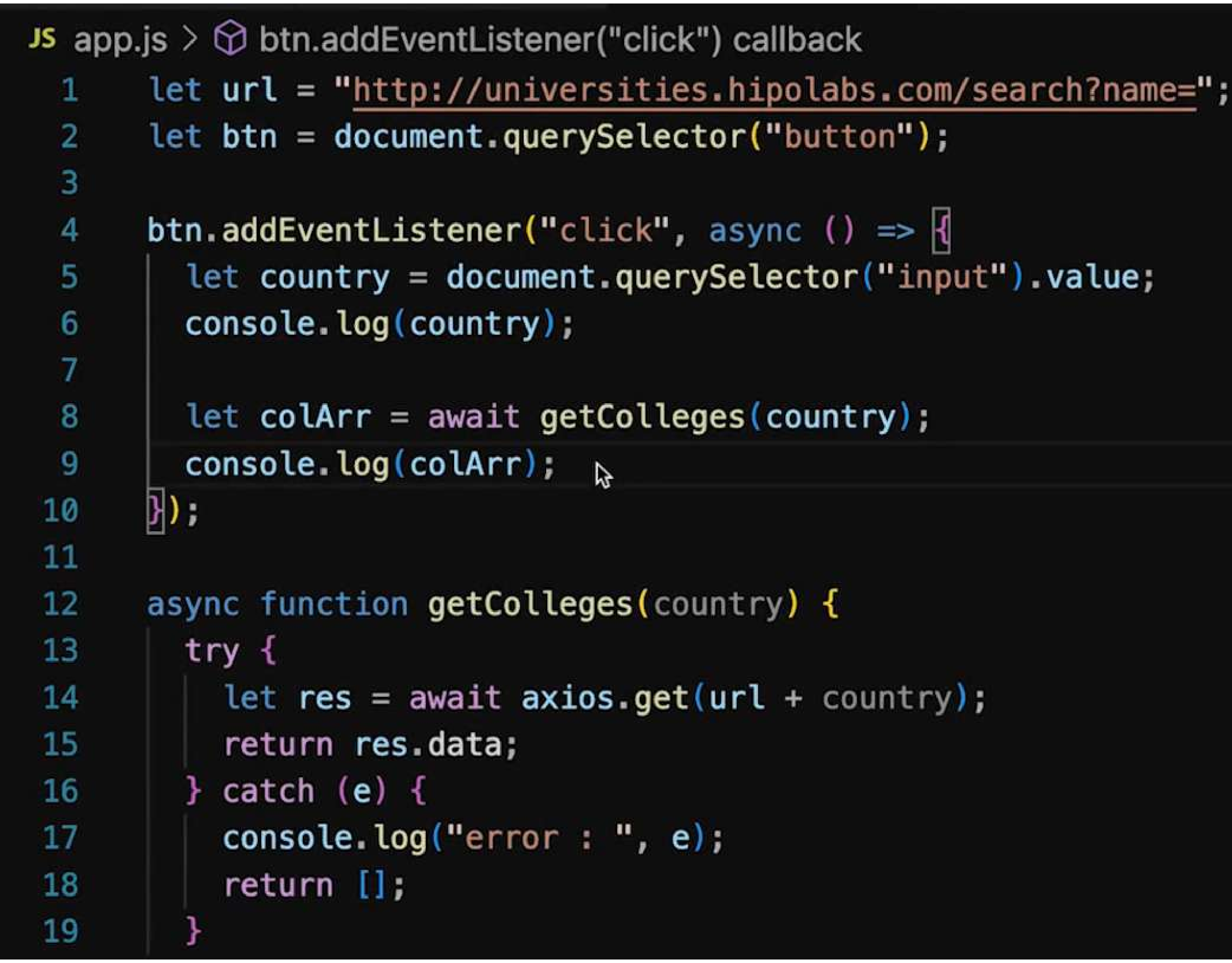
```
  ▼ {data: Array(1), status: 200,
    ig: {…}, …}
    ▶ config: {transitional: {…},
    ▼ data: Array(1)
      ▶ 0: {country: 'Nepal', dom
        length: 1
      ▶ [[Prototype]]: Array(0)
    ▶ headers: le {content-length
    ▶ request: XMLHttpRequest {on
      status: 200
      statusText: "OK"
    ▶ [[Prototype]]: Object
india

  ▼ {data: Array(52), status: 20
    fig: {…}, …}
    ▶ config: {transitional: {…},
    ▼ data: Array(52)
      ▶ 0: {country: 'India', dom
      ▶ 1: {country: 'India', dom
      ▶ 2: {country: 'India', dom
      ▶ 3: {country: 'India', dom
      ▶ 4: {country: 'India', dom
      ▶ 5: {country: 'India', dom
      ▶ 6: {country: 'India', dom
      ▶ 7: {country: 'India', dom
```



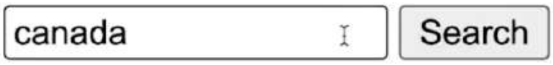
```javascript
let url = "http://universities.hipolabs.com/search?name=";
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let country = document.querySelector("input").value;
  console.log(country);
  getColleges(country);
});

async function getColleges(country) {
  try {
    let res = await axios.get(url + country);
    console.log(res.data);
  } catch (e) {
    console.log("error : ", e);
    return [];
  }
}

```



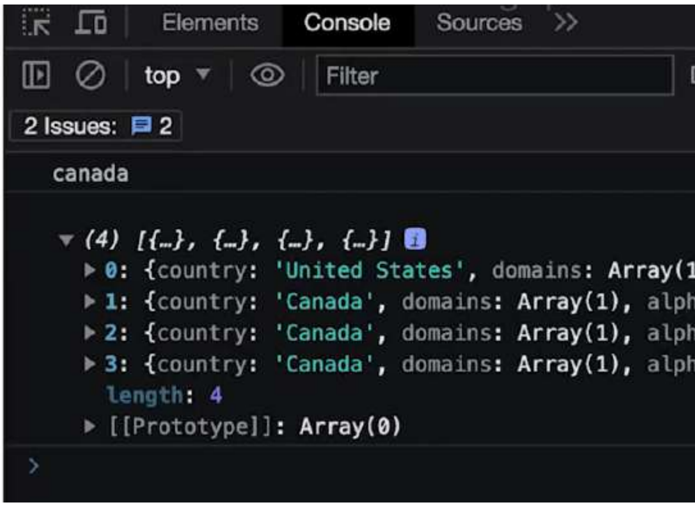
```
canada

▼ (4) [{…}, {…}, {…}, {…}] i
  ▼ 0:
      alpha_two_code: "US"
      country: "United States"
    ▶ domains: ['canadacollege.ed
      name: "Canada College"
      state-province: null
    ▶ web_pages: ['http://canadac
    ▶ [[Prototype]]: Object
  ▶ 1: {country: 'Canada', domain
  ▶ 2: {country: 'Canada', domain
  ▶ 3: {country: 'Canada', domain
    length: 4
  ▶ [[Prototype]]: Array(0)
>
```



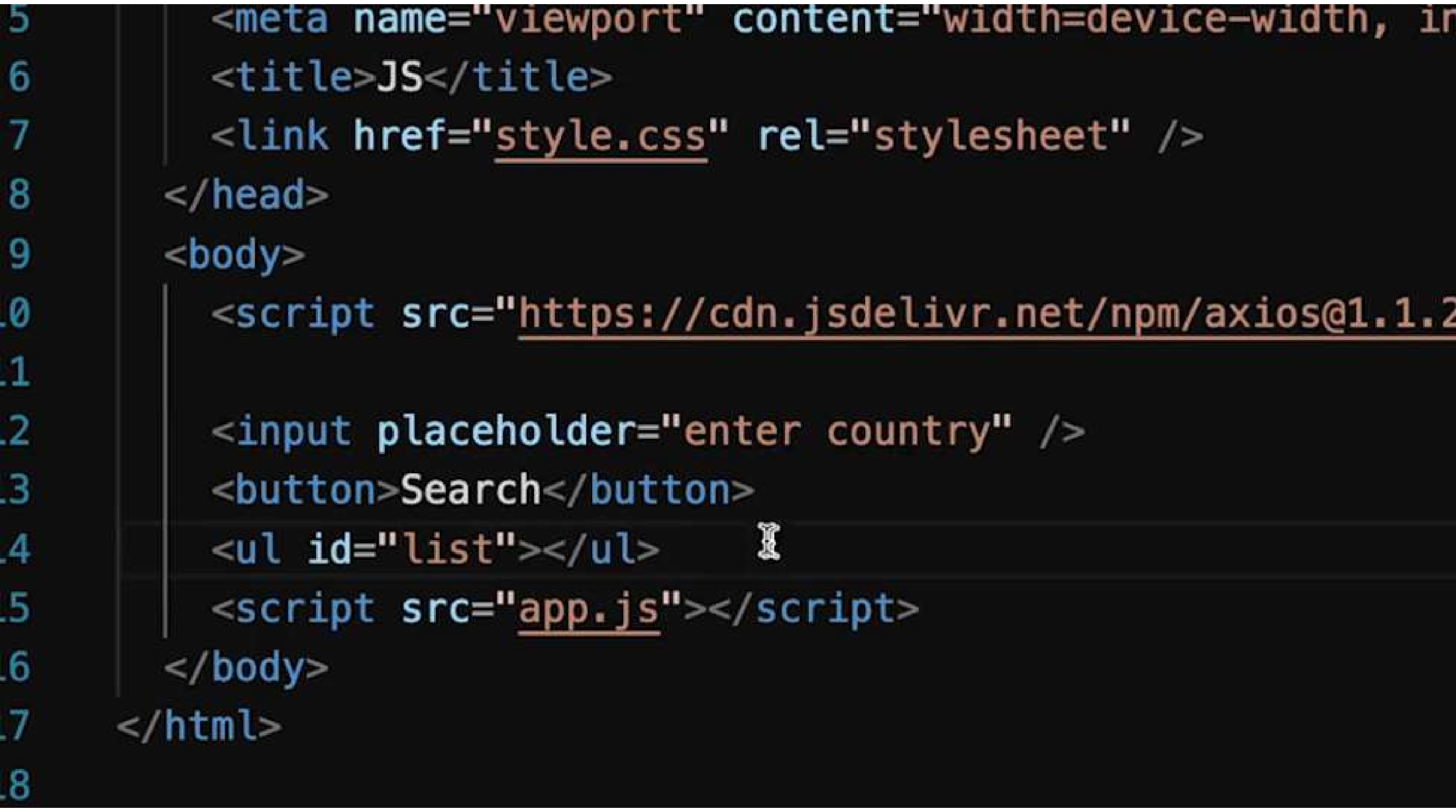
```javascript
let url = "http://universities.hipolabs.com/search?name=";
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let country = document.querySelector("input").value;
  console.log(country);

  let colArr = await getColleges(country);
  console.log(colArr);
});

async function getColleges(country) {
  try {
    let res = await axios.get(url + country);
    return res.data;
  } catch (e) {
    console.log("error : ", e);
    return [];
  }
```



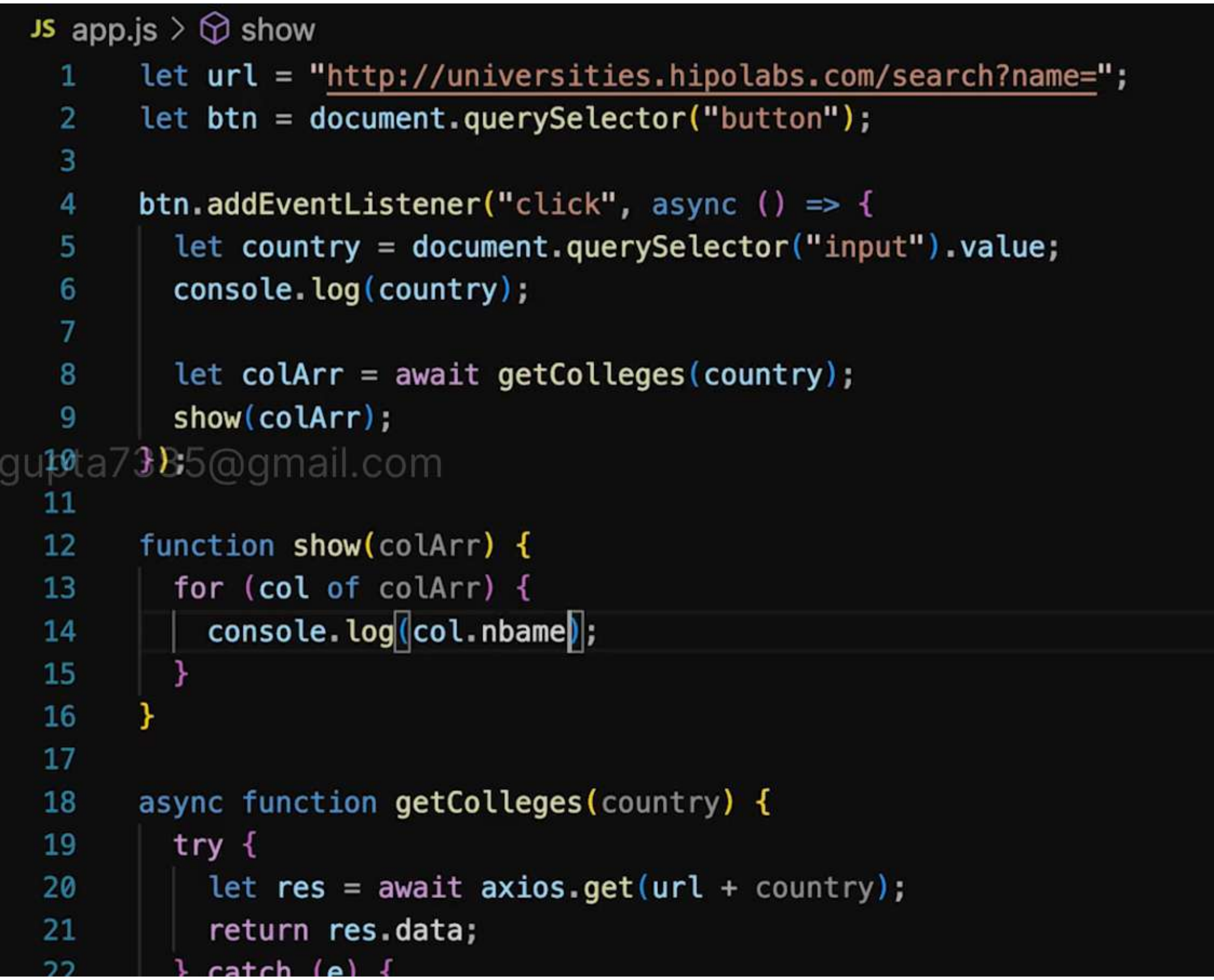
canada



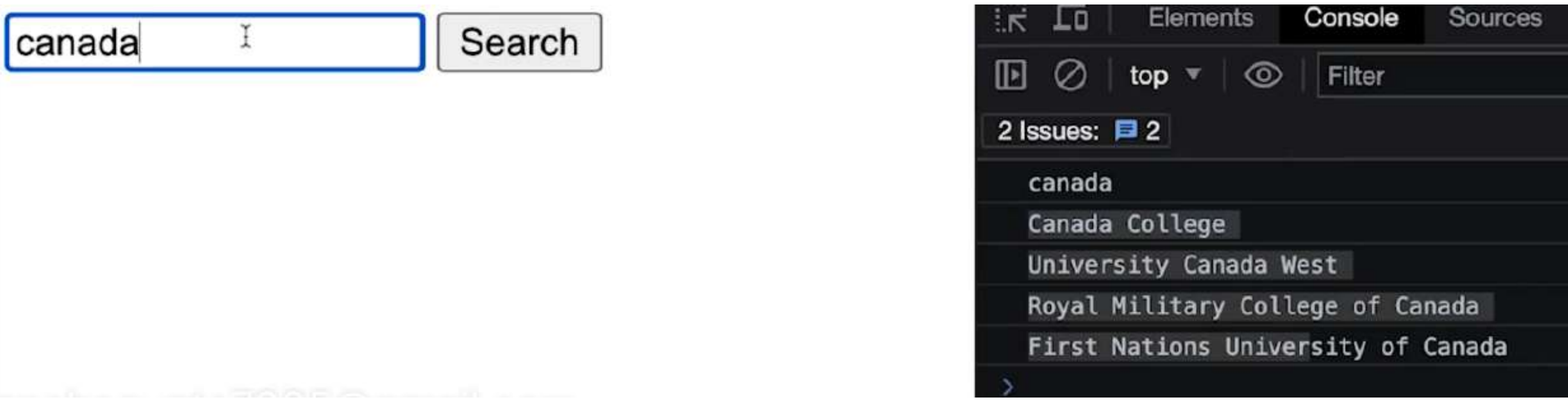
```
canada

▼ (4) [{…}, {…}, {…}, {…}]
  ▶ 0: {country: 'United States', domains: Array(1
  ▶ 1: {country: 'Canada', domains: Array(1), alph
  ▶ 2: {country: 'Canada', domains: Array(1), alph
  ▶ 3: {country: 'Canada', domains: Array(1), alph
    length: 4
  ▶ [[Prototype]]: Array(0)
>
```



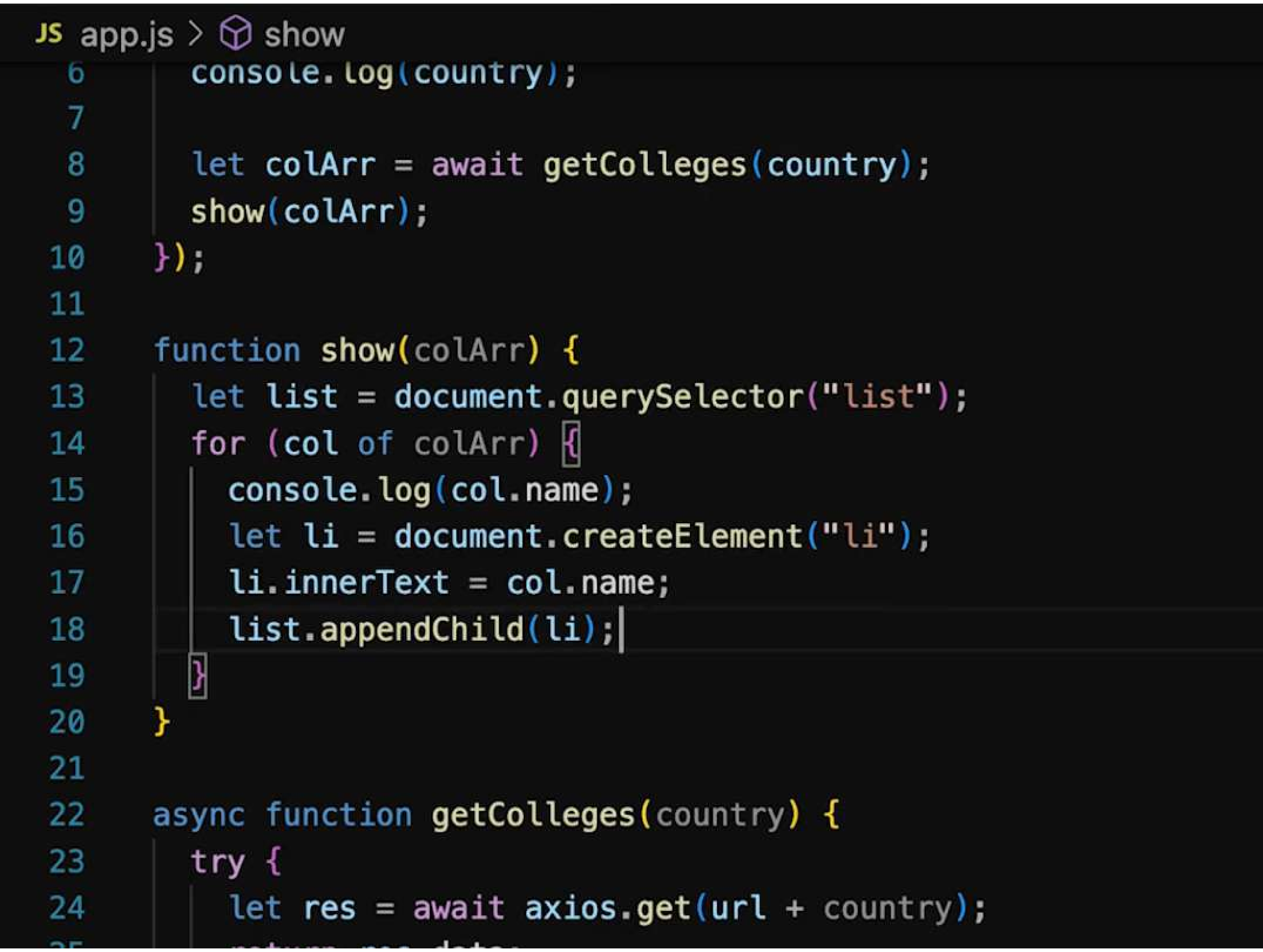
```
    <meta name="viewport" content="width=device-width, ir
    <title>JS</title>
    <link href="style.css" rel="stylesheet" />
  </head>
  <body>
    <script src="https://cdn.jsdelivr.net/npm/axios@1.1.2

    <input placeholder="enter country" />
    <button>Search</button>
    <ul id="list"></ul>
    <script src="app.js"></script>
  </body>
</html>
```



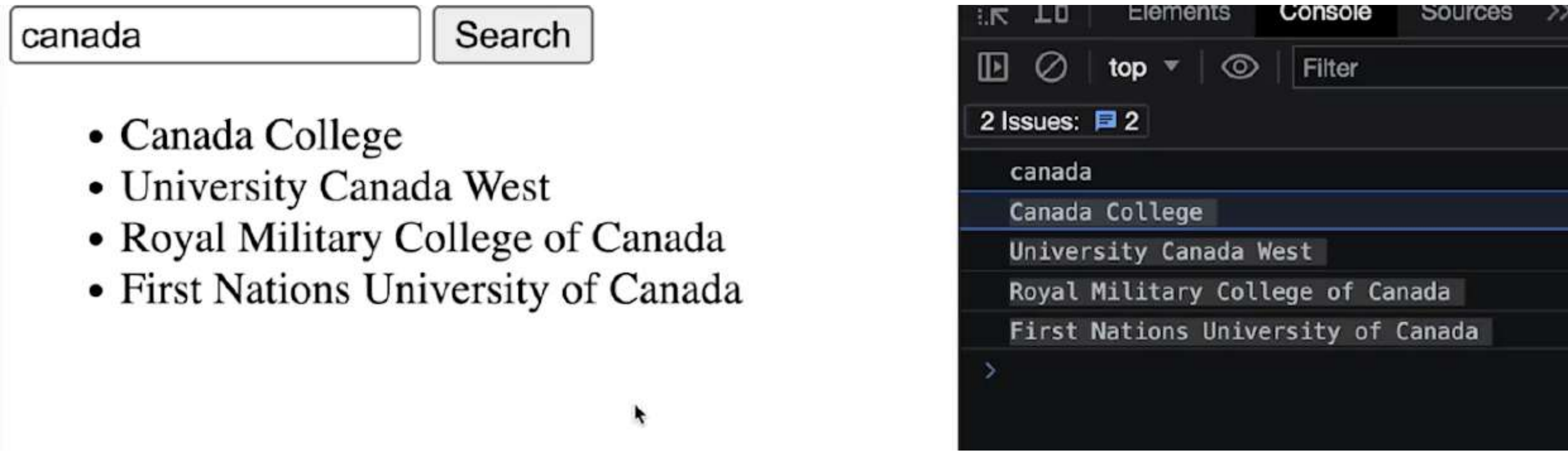
```
let url = "http://universities.hipolabs.com/search?name=";
let btn = document.querySelector("button");

btn.addEventListener("click", async () => {
  let country = document.querySelector("input").value;
  console.log(country);

  let colArr = await getColleges(country);
  show(colArr);
});

function show(colArr) {
  for (col of colArr) {
    console.log(col.nbame);
  }
}

async function getColleges(country) {
  try {
    let res = await axios.get(url + country);
    return res.data;
  } catch (e) {
```



```
canada
Canada College
University Canada West
Royal Military College of Canada
First Nations University of Canada
```

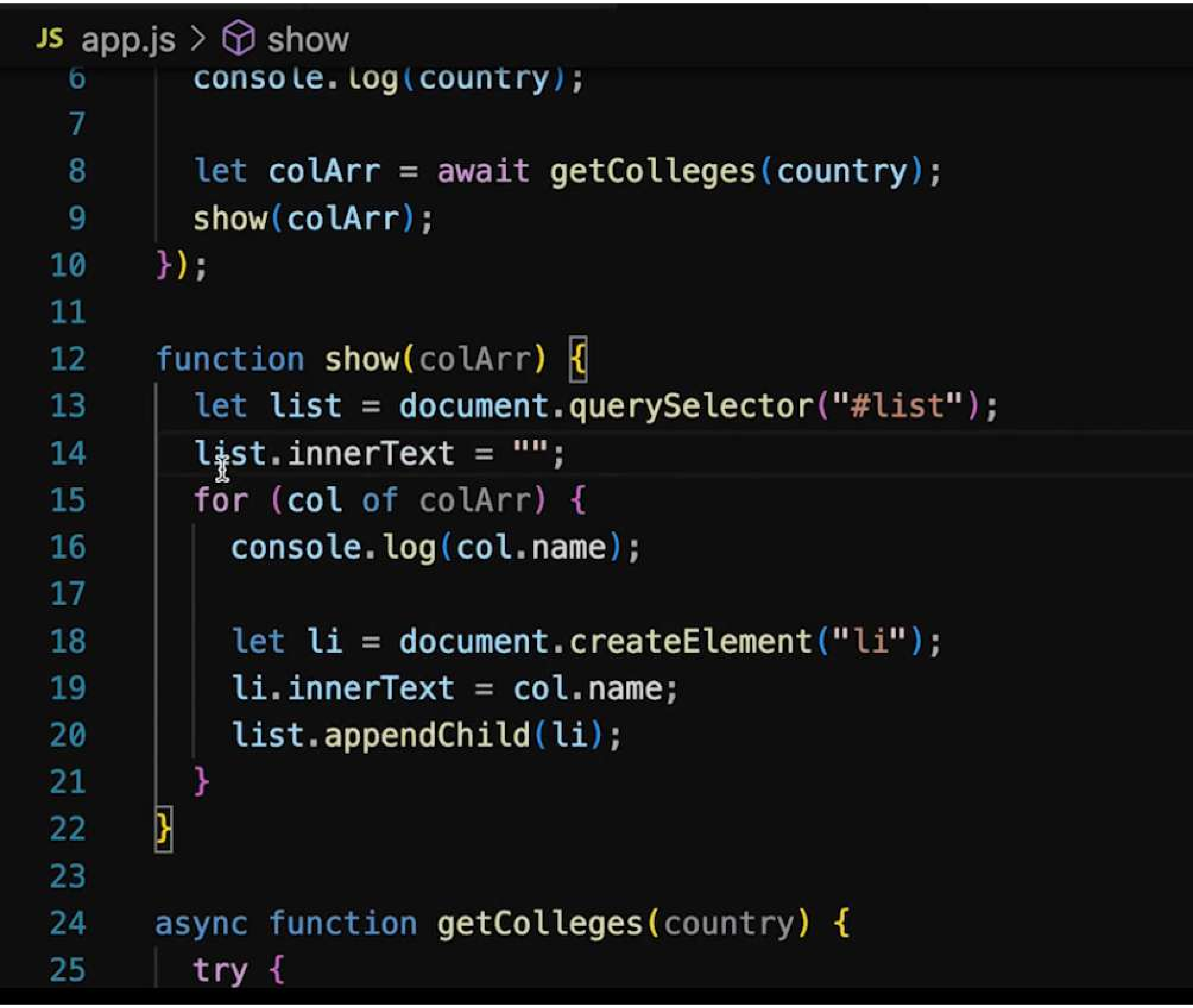


```
  console.log(country);

  let colArr = await getColleges(country);
  show(colArr);
});

function show(colArr) {
  let list = document.querySelector("list");
  for (col of colArr) {
    console.log(col.name);
    let li = document.createElement("li");
    li.innerText = col.name;
    list.appendChild(li);
  }
}

async function getColleges(country) {
  try {
    let res = await axios.get(url + country);
    return res.data;
```

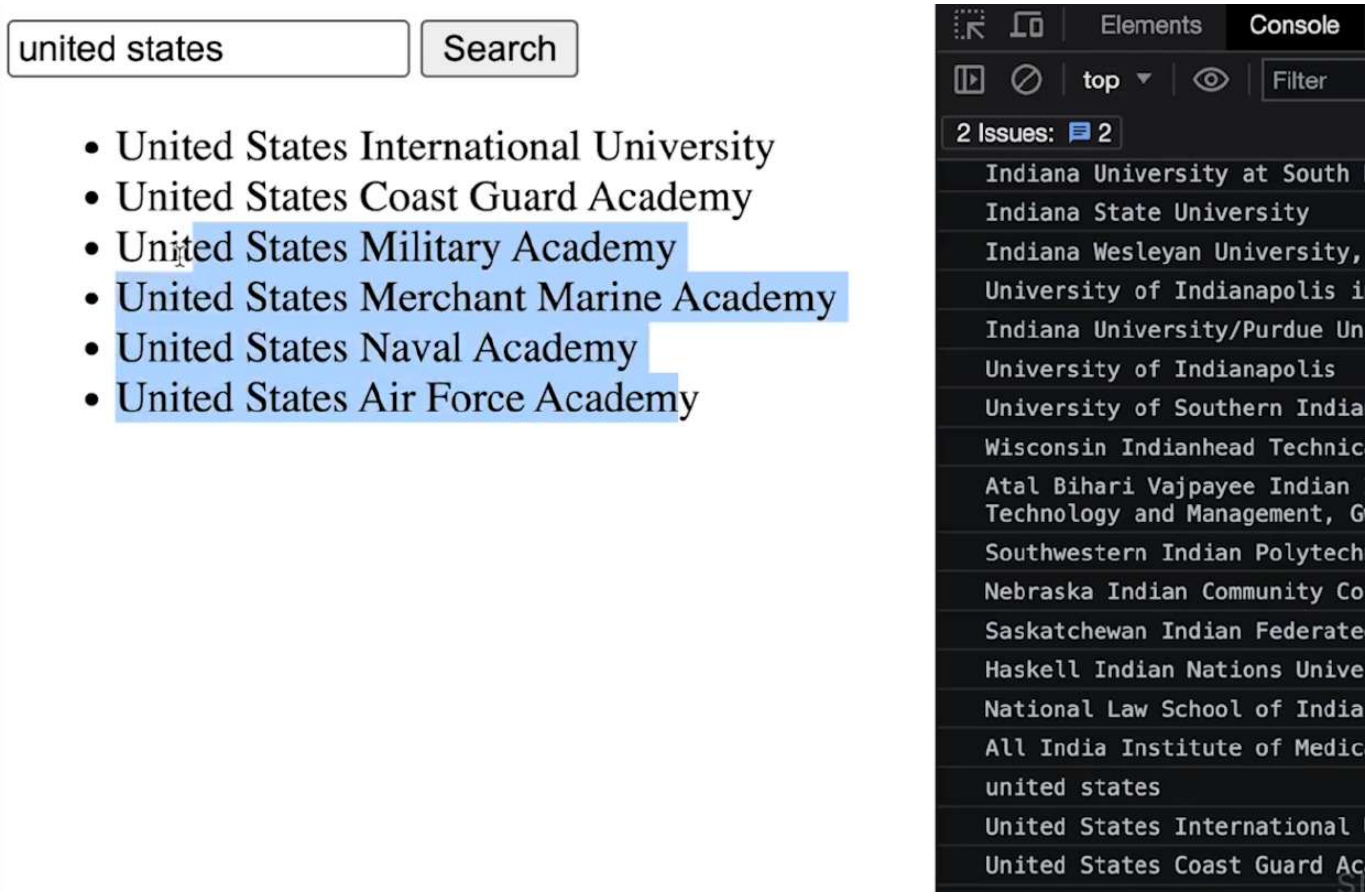


- Canada College
- University Canada West
- Royal Military College of Canada
- First Nations University of Canada

```
canada
Canada College
University Canada West
Royal Military College of Canada
First Nations University of Canada
```



```
  console.log(country);

  let colArr = await getColleges(country);
  show(colArr);
});

function show(colArr) {
  let list = document.querySelector("#list");
  list.innerText = "";
  for (col of colArr) {
    console.log(col.name);

    let li = document.createElement("li");
    li.innerText = col.name;
    list.appendChild(li);
  }
}

async function getColleges(country) {
  try {
```



united states

- United States International University
- United States Coast Guard Academy
- United States Military Academy
- United States Merchant Marine Academy
- United States Naval Academy
- United States Air Force Academy

```
Indiana University at South
Indiana State University
Indiana Wesleyan University,
University of Indianapolis i
Indiana University/Purdue Un
University of Indianapolis
University of Southern India
Wisconsin Indianhead Technic
Atal Bihari Vajpayee Indian
Technology and Management, G
Southwestern Indian Polytech
Nebraska Indian Community Co
Saskatchewan Indian Federate
Haskell Indian Nations Unive
National Law School of India
All India Institute of Medic
united states
United States International
United States Coast Guard Ac
```