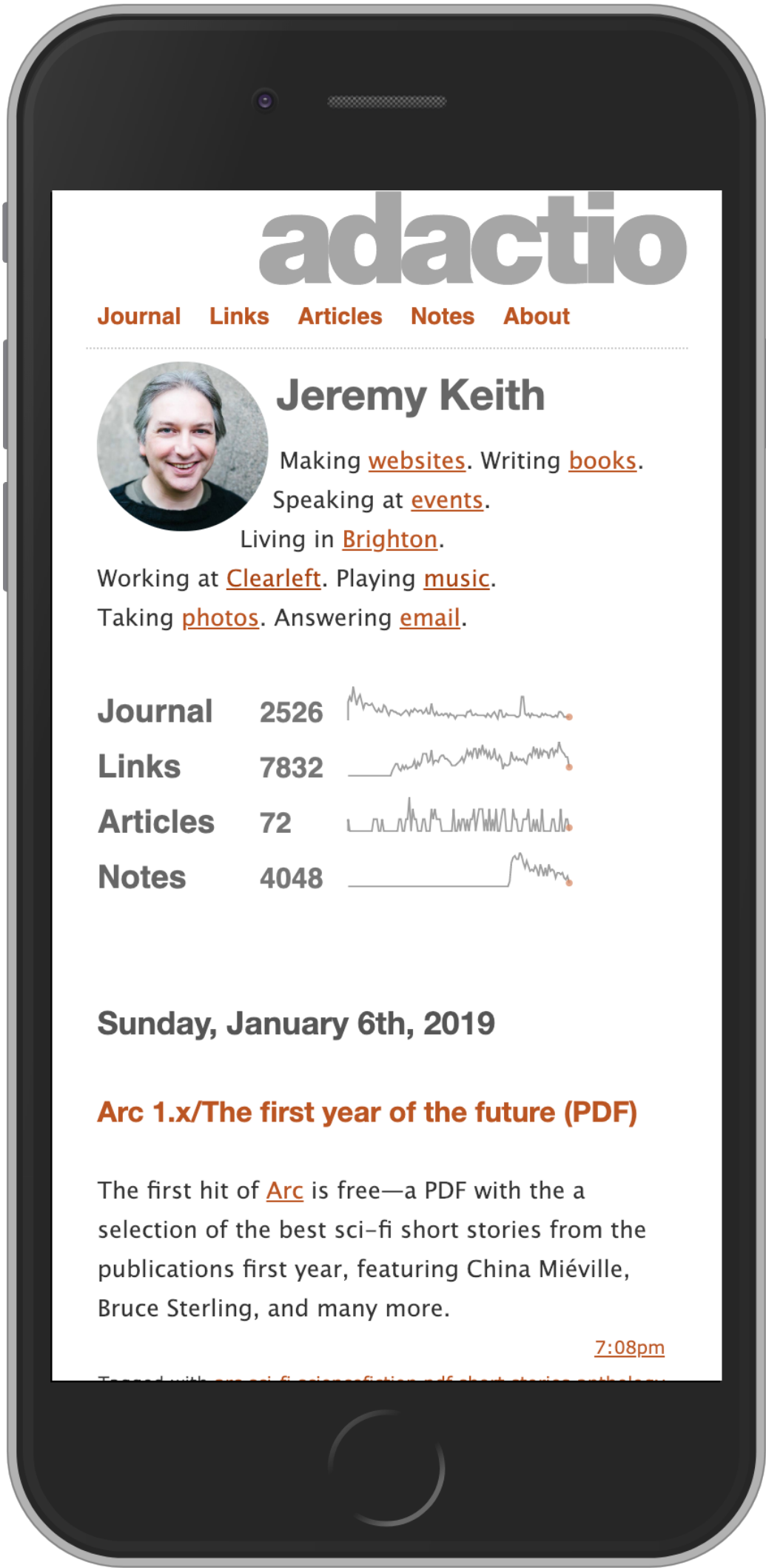
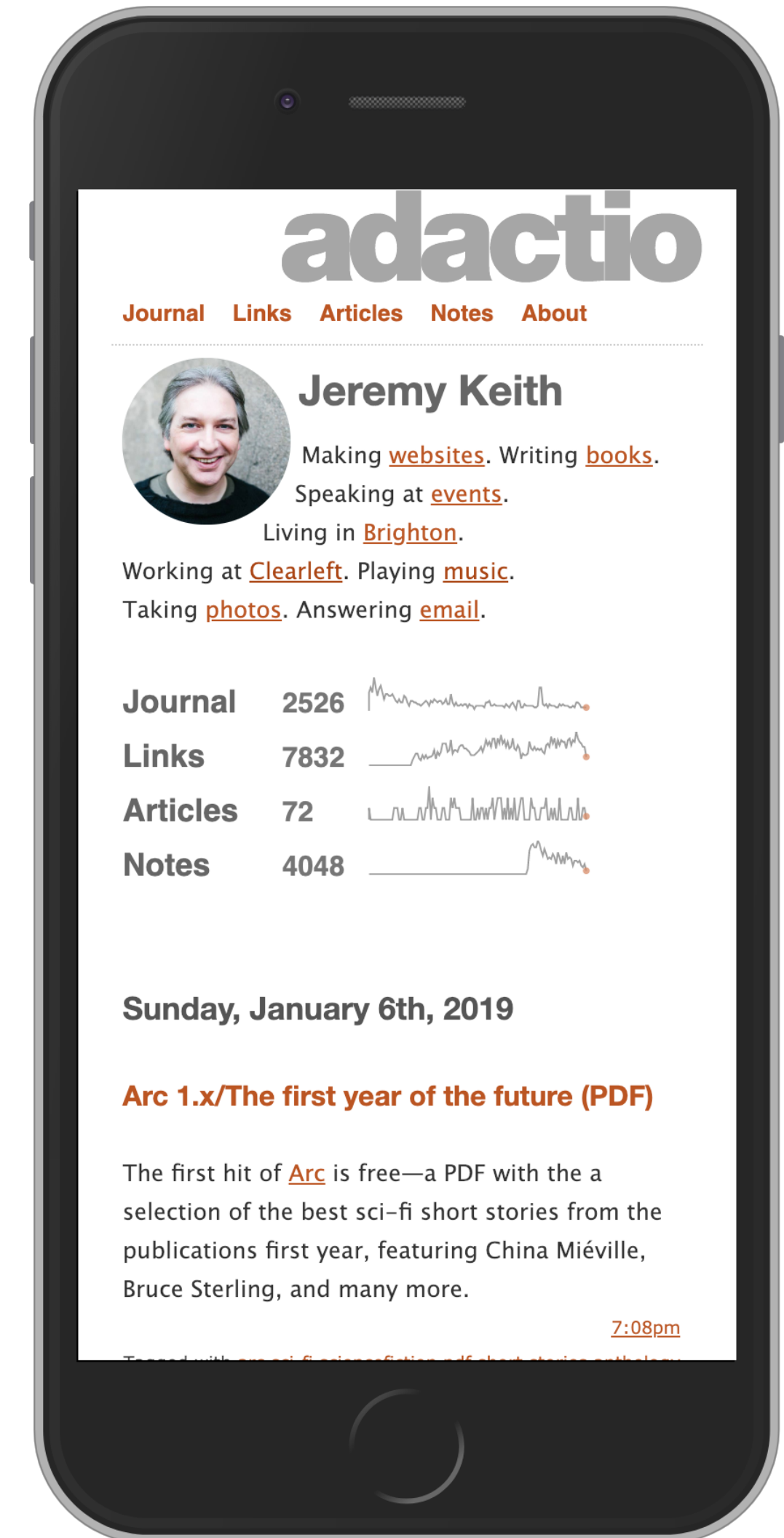


Going Offline

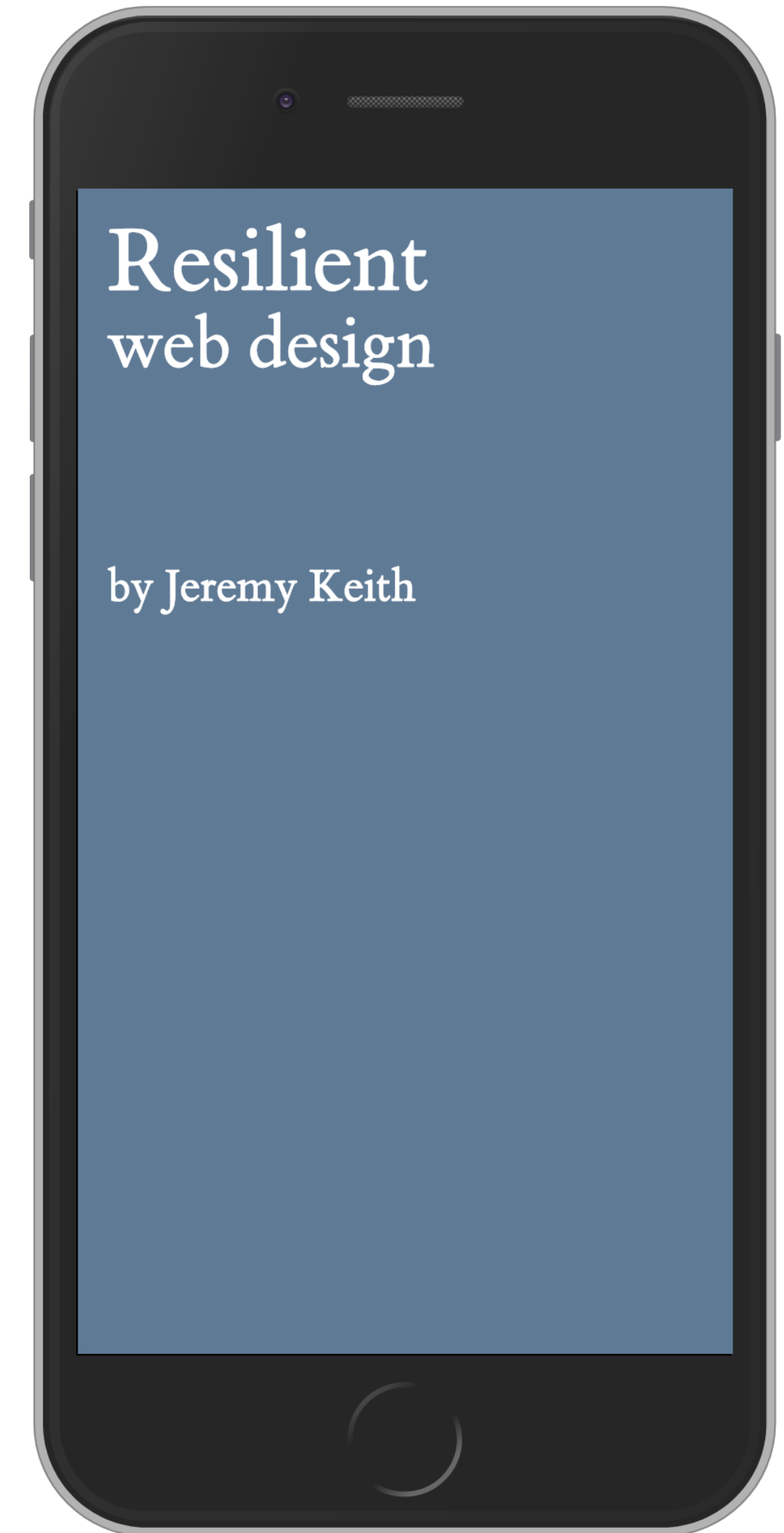
adactio.com



adactio.com



resilientwebdesign.com



DOM Scripting

Web Design with JavaScript and the Document Object Model

*Separate behavior from structure using
unobtrusive JavaScript.*

*Add dynamic effects with progressive
enhancement.*

*Ensure backwards-compatibility through
graceful degradation.*

Jeremy Keith
Foreword by Dave Shea

friendsof 
DESIGNER TO DESIGNER™
an Apress® company

2005

JavaScript

*When someone mouses over a link,
make it look different.*

*When someone fills out a form,
make sure the inputs are valid.*

JavaScript

Find stuff and do stuff to it.

JavaScript

Find stuff and do stuff to it.

jQuery

JavaScript

Find stuff and do stuff to it.

```
$('.stuff').click(doStuff);
```

jQuery

JavaScript

```
document.querySelector('.stuff')  
  .addEventListener('click', doStuff);
```

Find stuff and do stuff to it.

```
$('.stuff').click(doStuff);
```

jQuery

JavaScript

code

Logic.

JavaScript

Logic.

code

Bulletproof Ajax

Jeremy Keith

2007

Update part of this page with data from a server.

Ajax

DOM Scripting

Find stuff on this page and do stuff to it.

Update part of this page with data from a server.

Ajax

DOM Scripting

Find stuff on this page and do stuff to it.

Update part of this page with data from a server.

Ajax

DOM Scripting

web worker

Ajax



web worker

service worker

 A BOOK APART

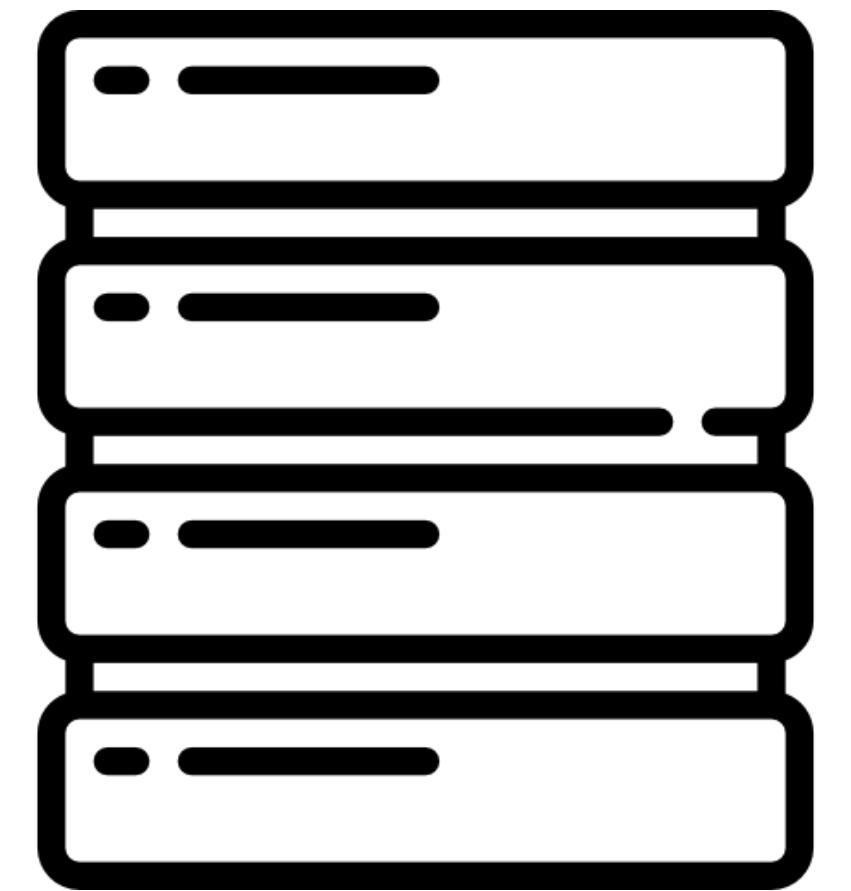
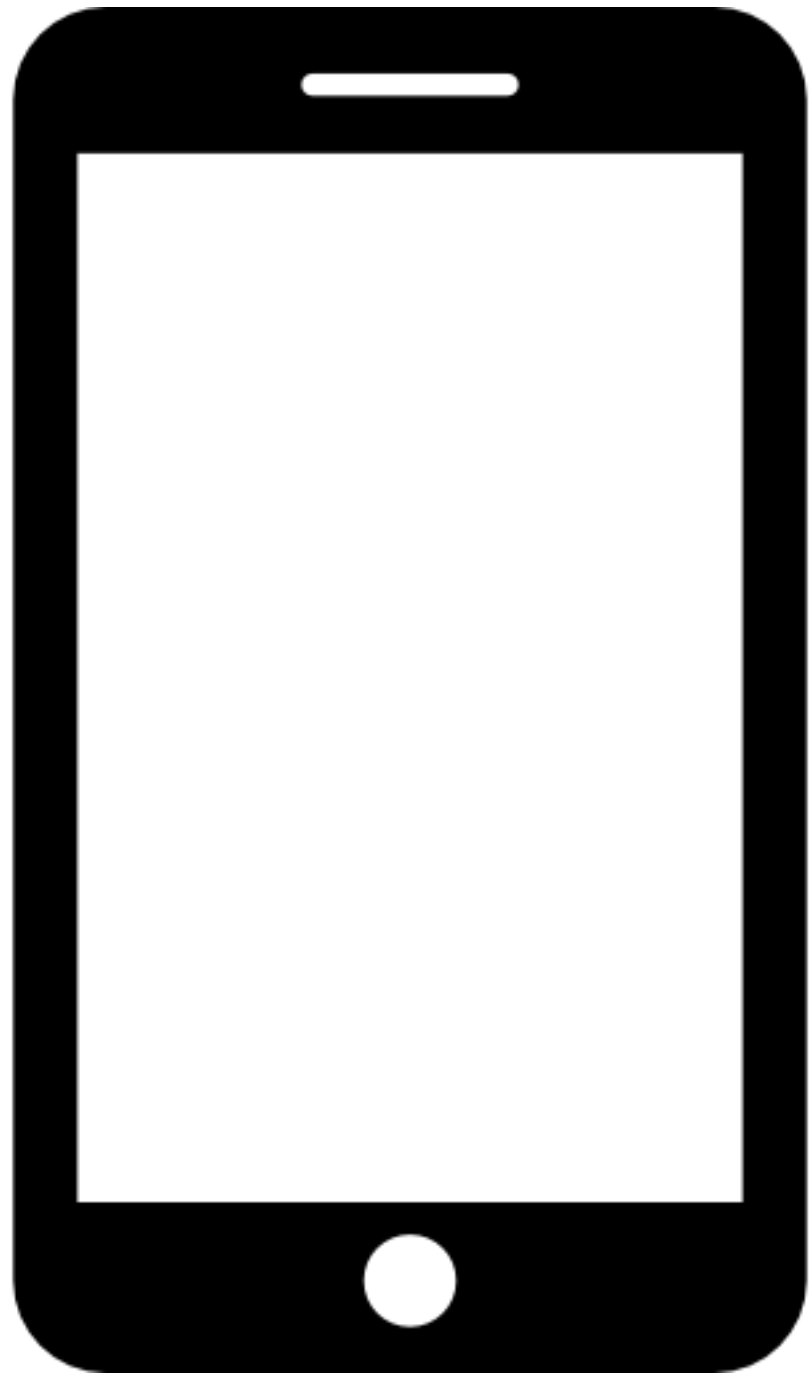
NO
26

Jeremy Keith

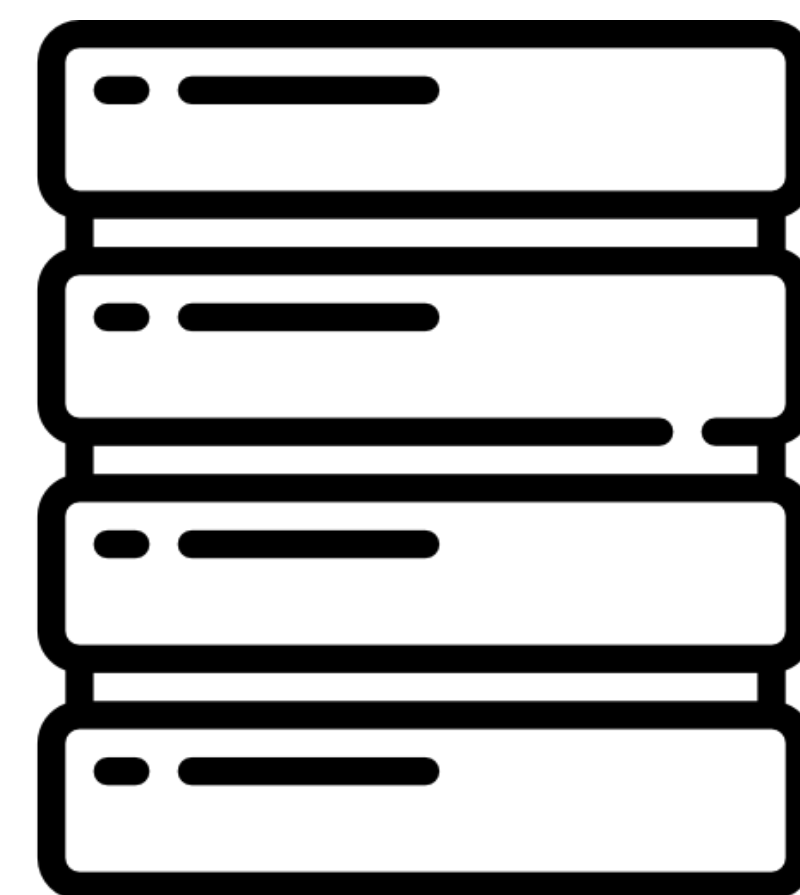
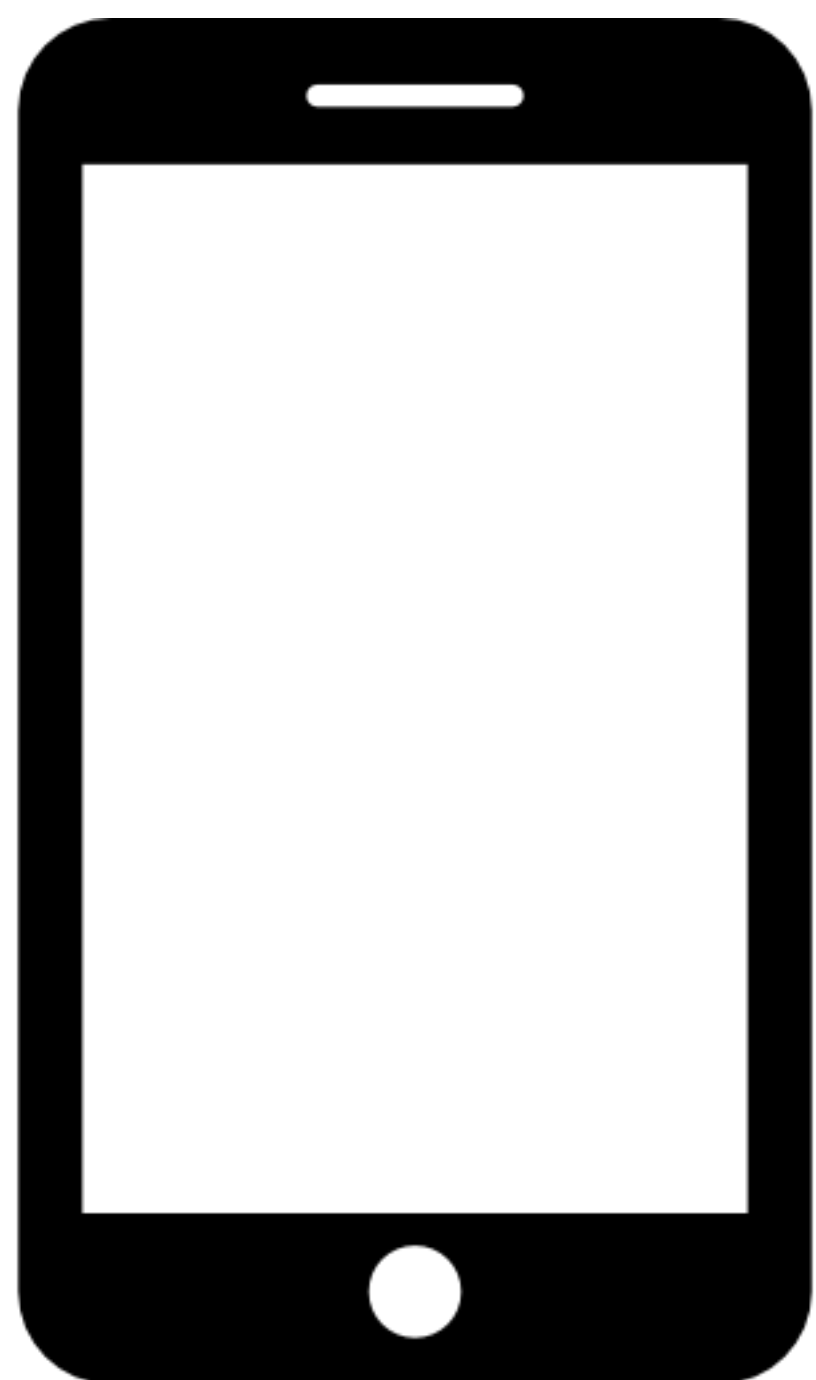
GOING OFFLINE

FOREWORD BY Aaron Gustafson

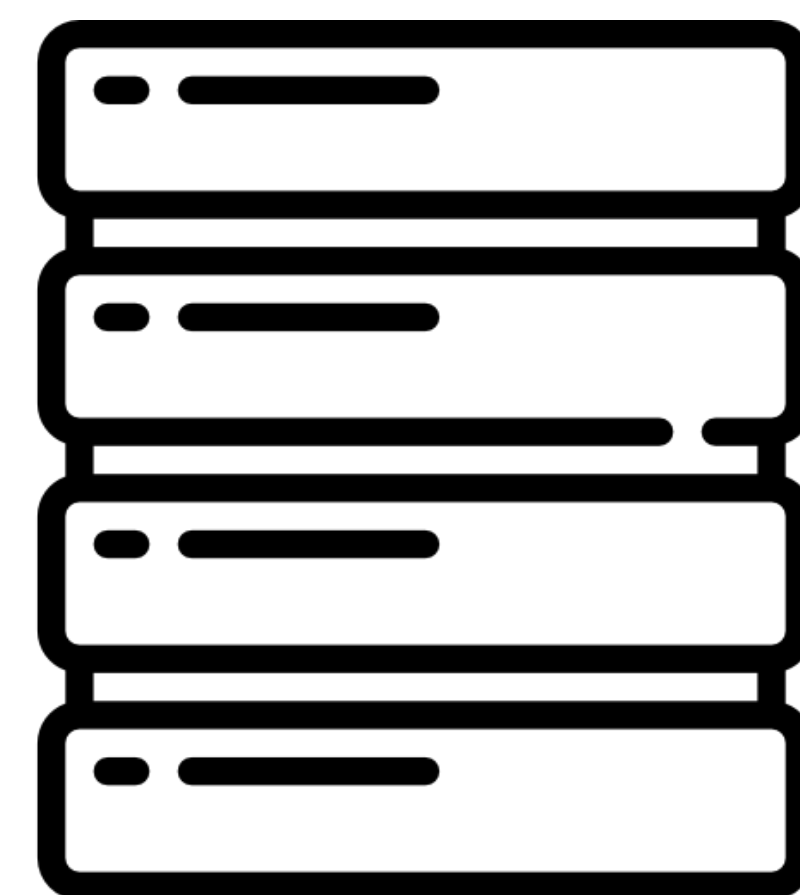
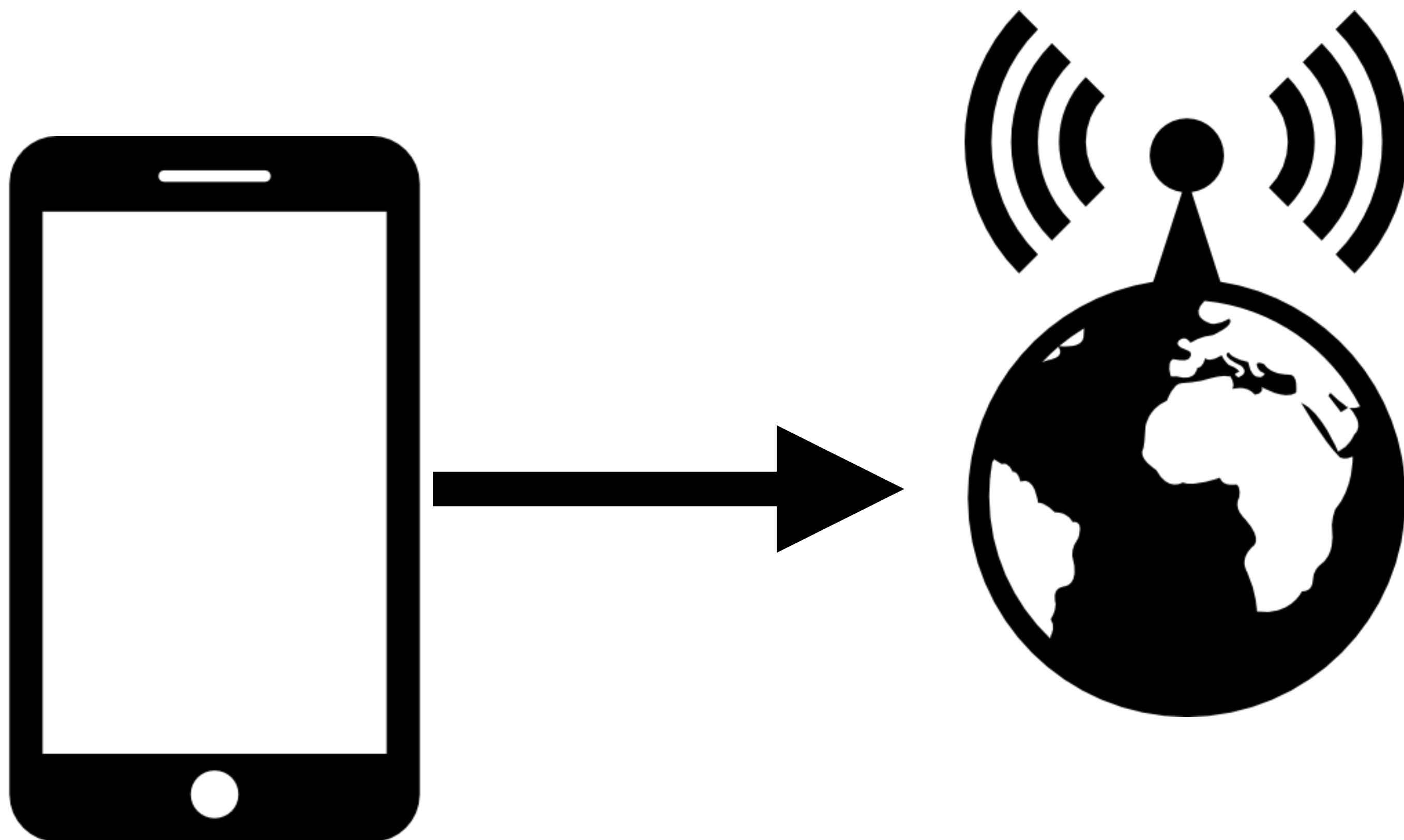
2018



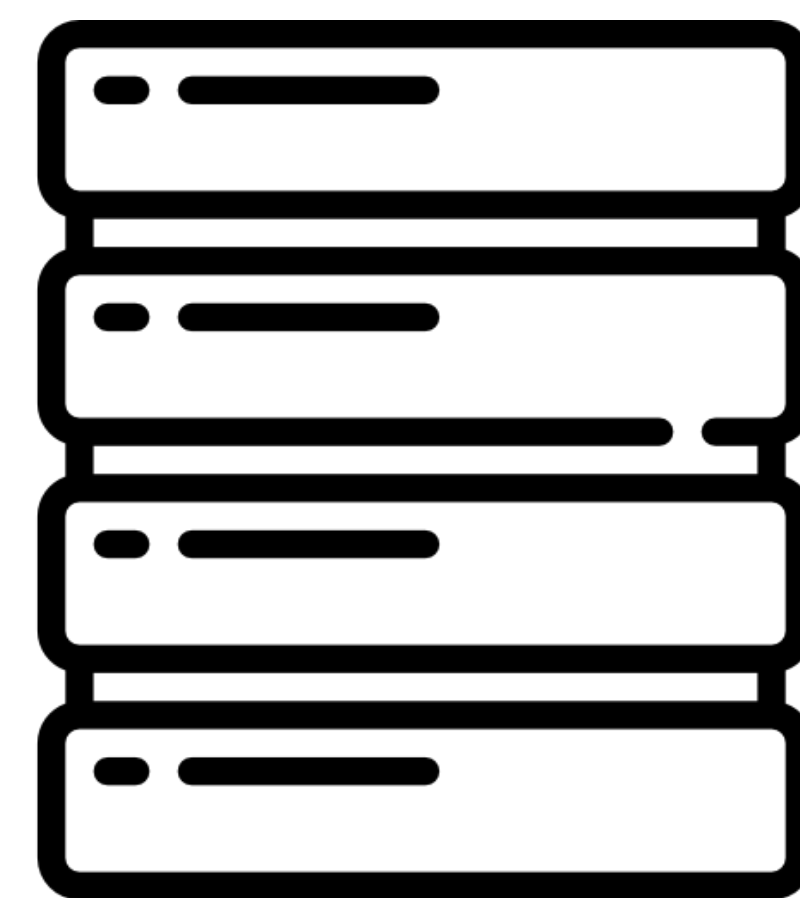
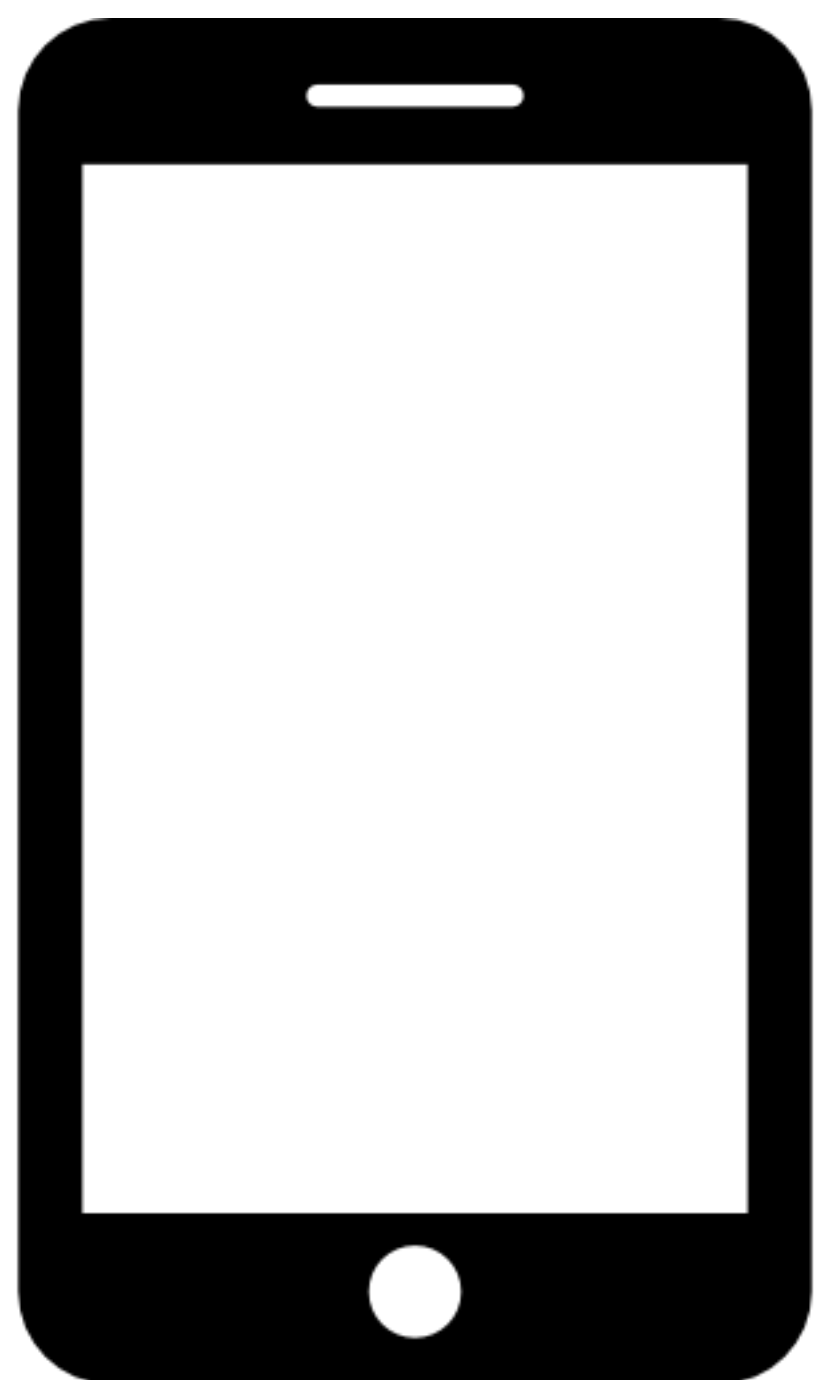
example.com



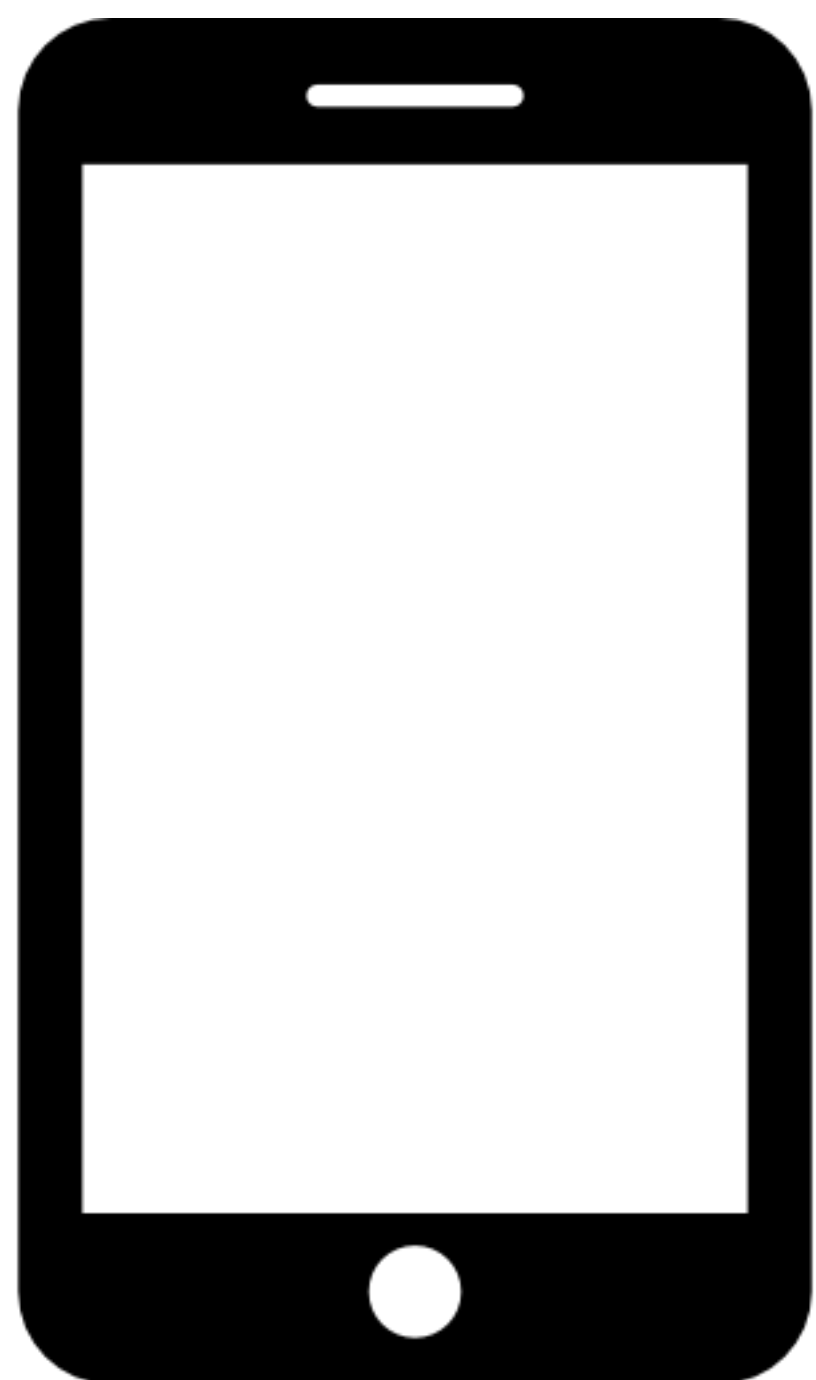
example.com



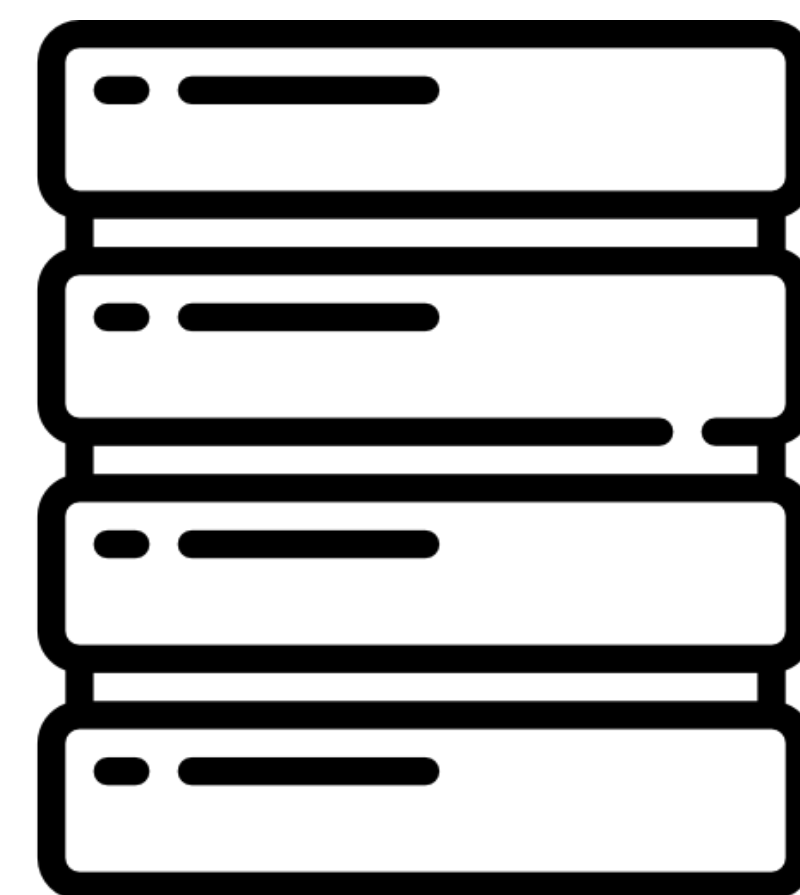
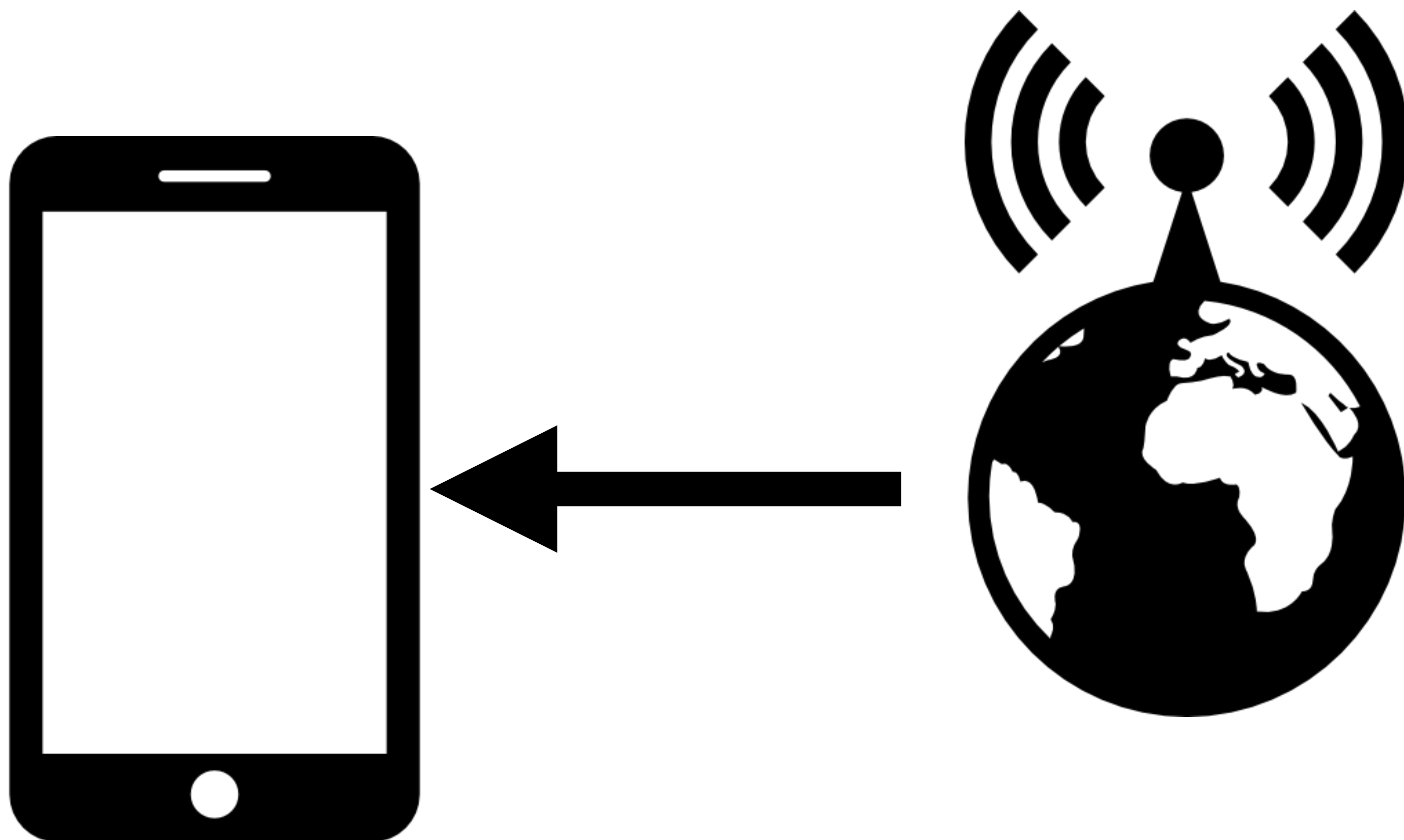
example.com



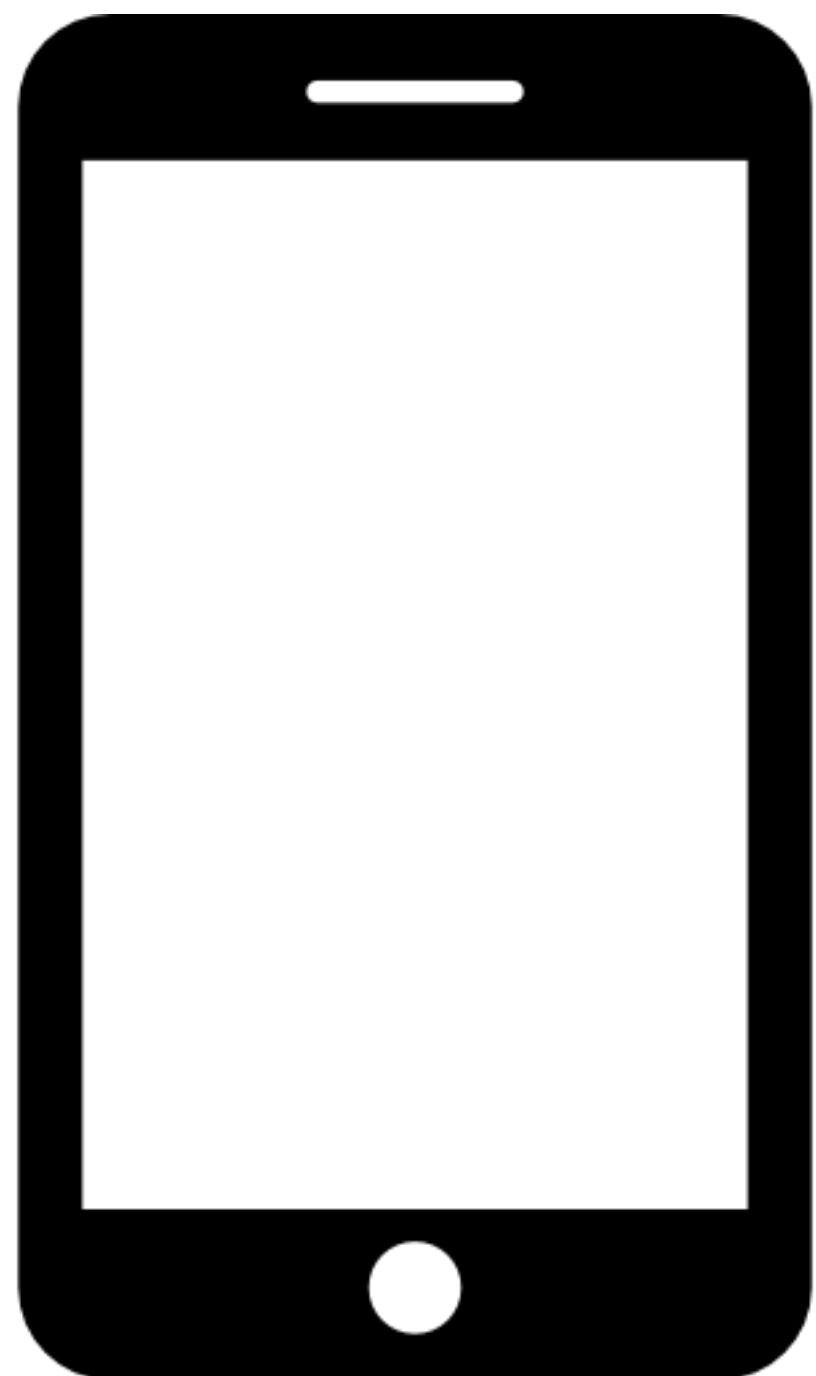
example.com



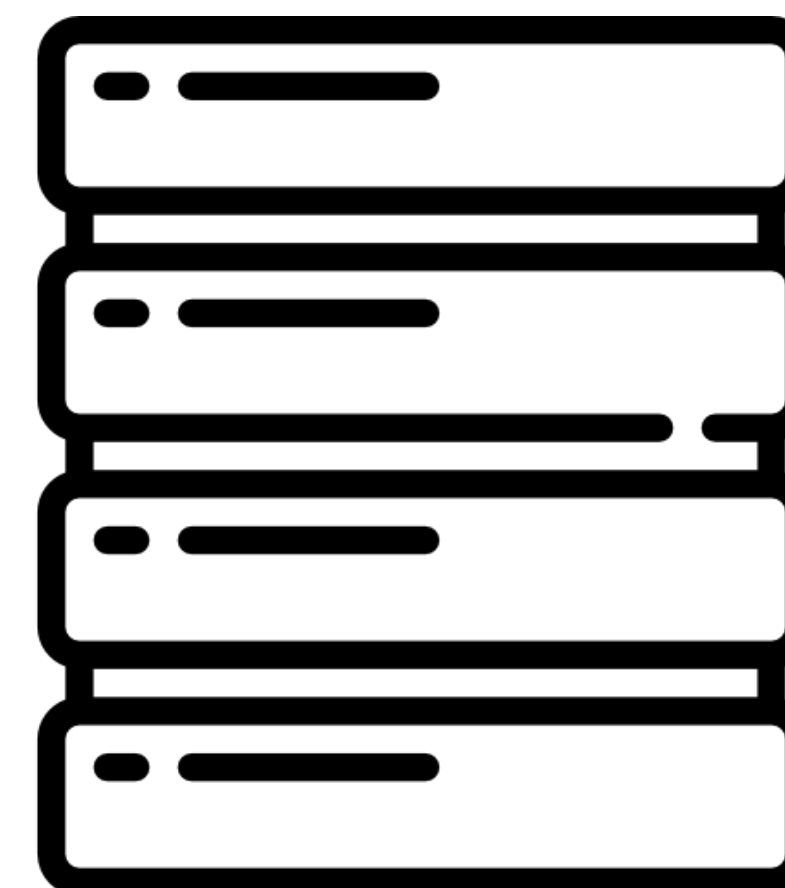
example.com



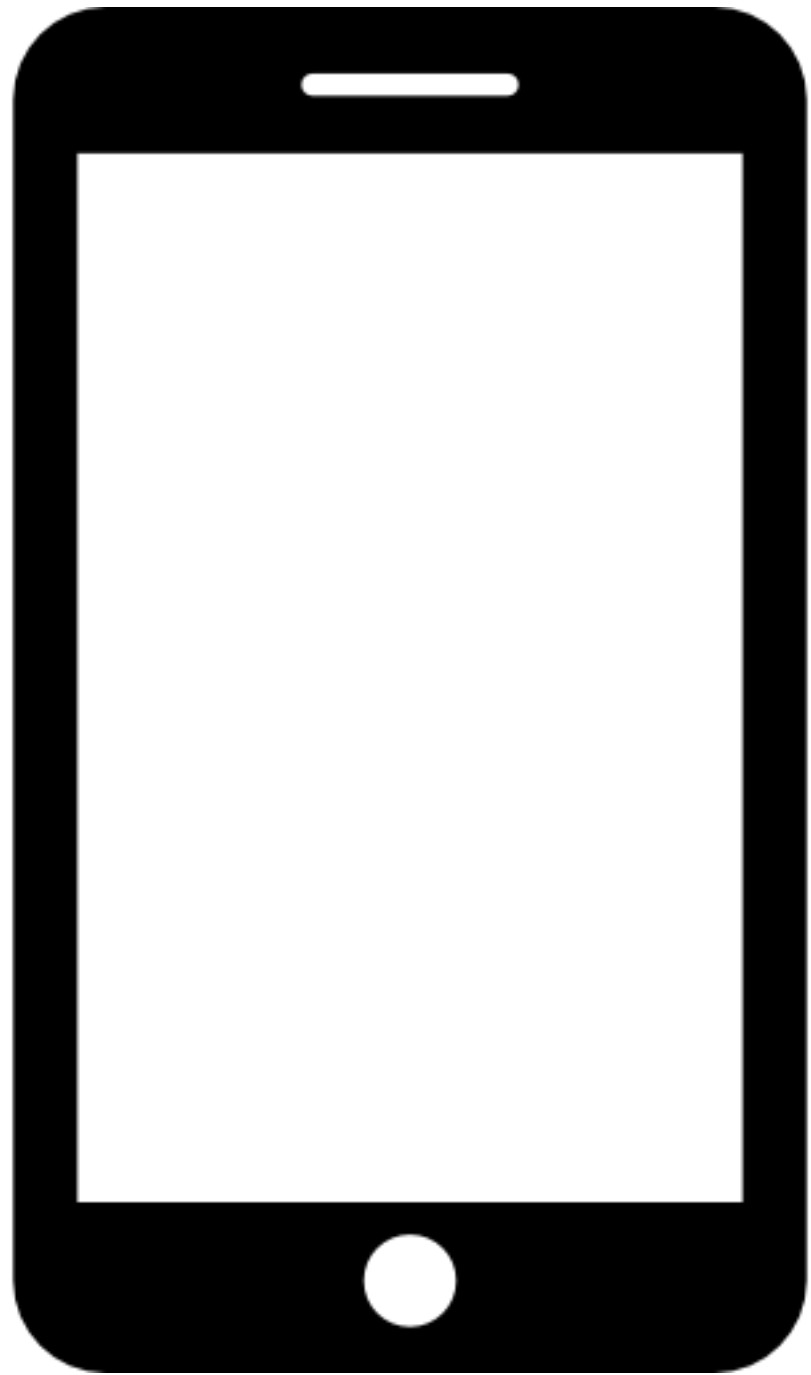
example.com



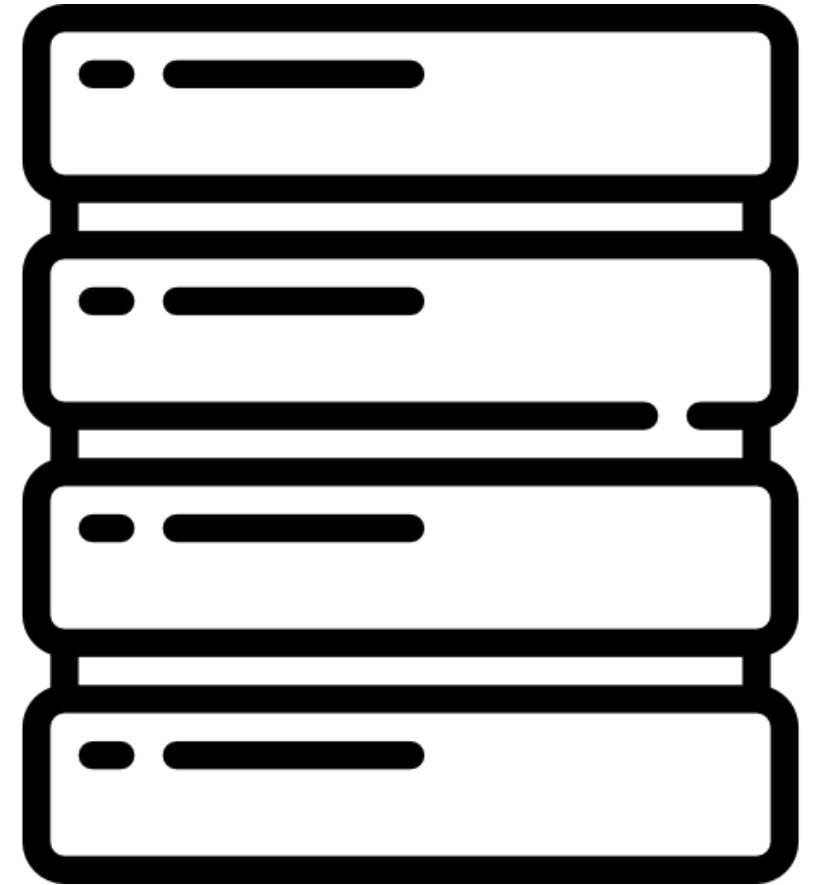
page.html
styles.css
script.js
icon.png



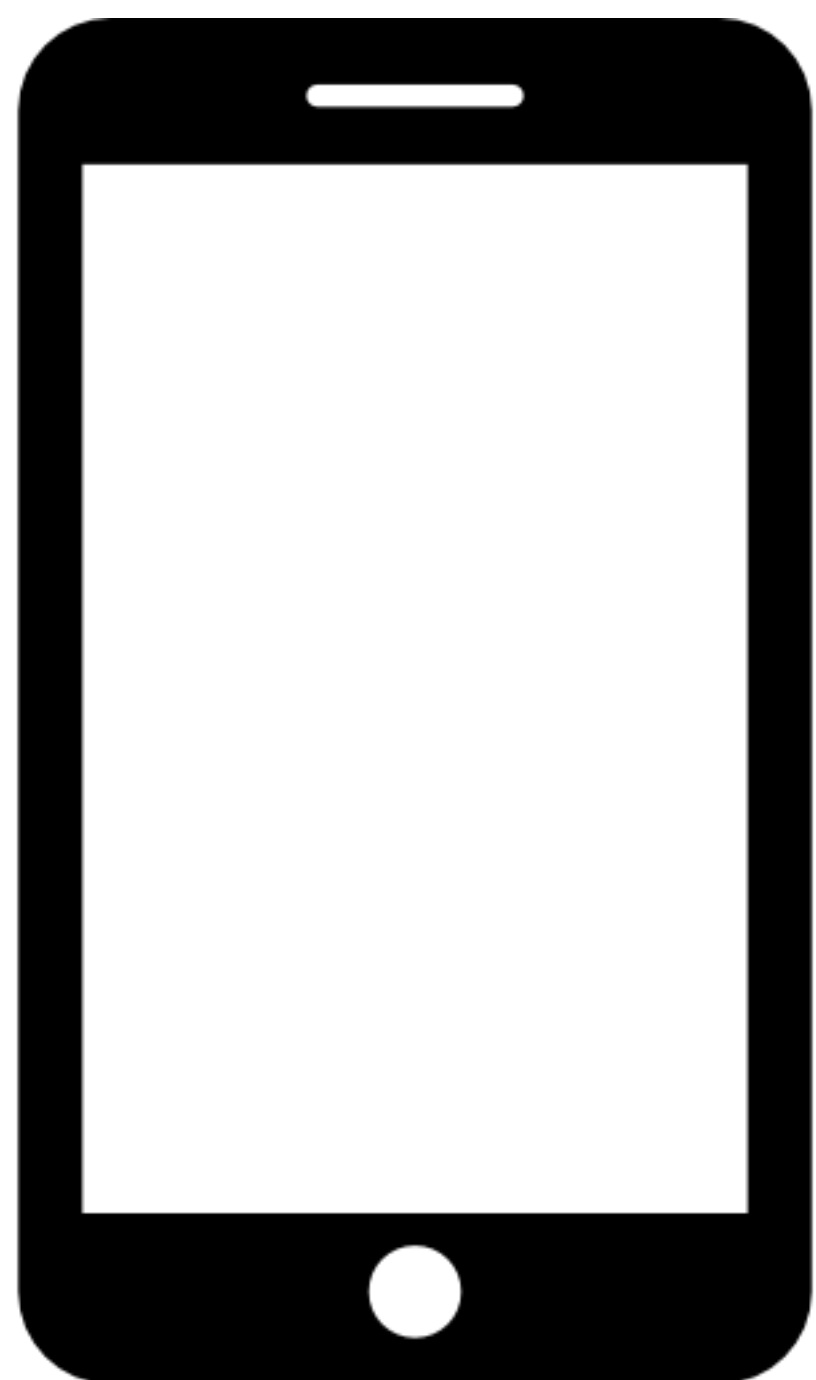
example.com



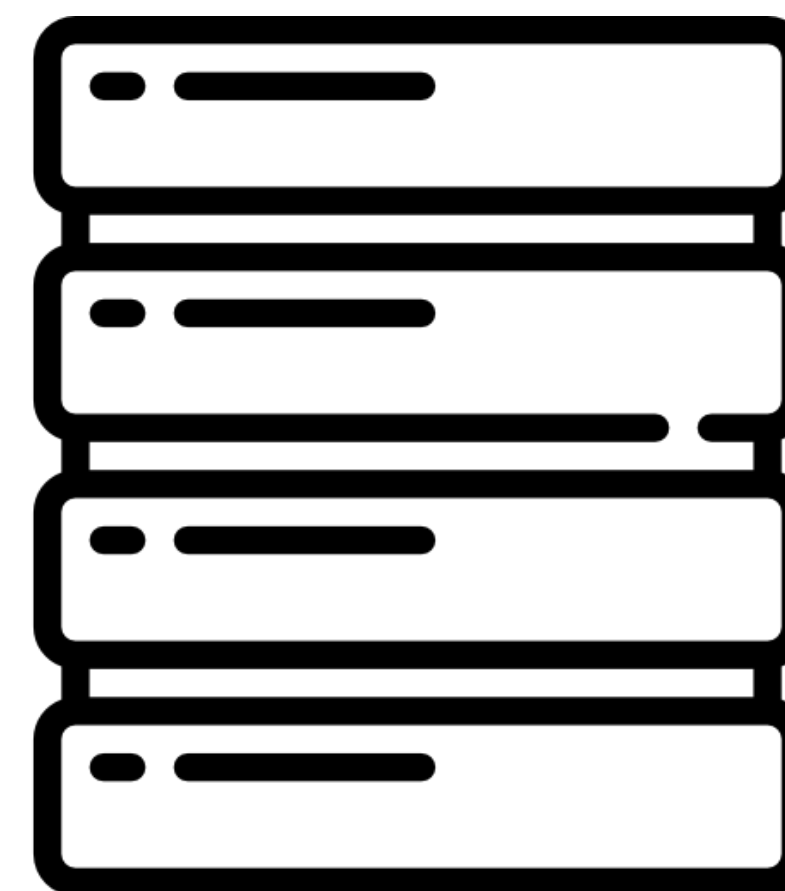
serviceworker.js



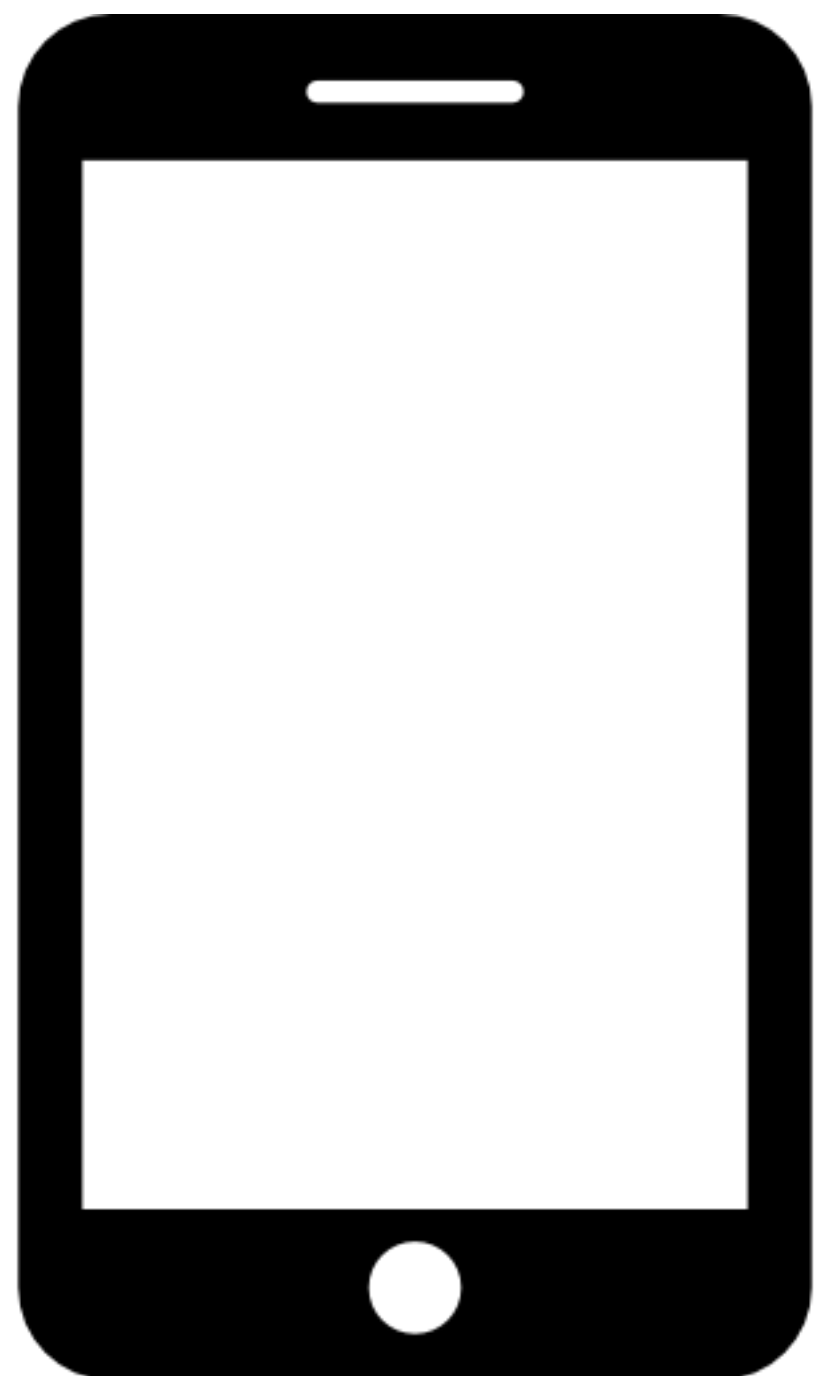
example.com



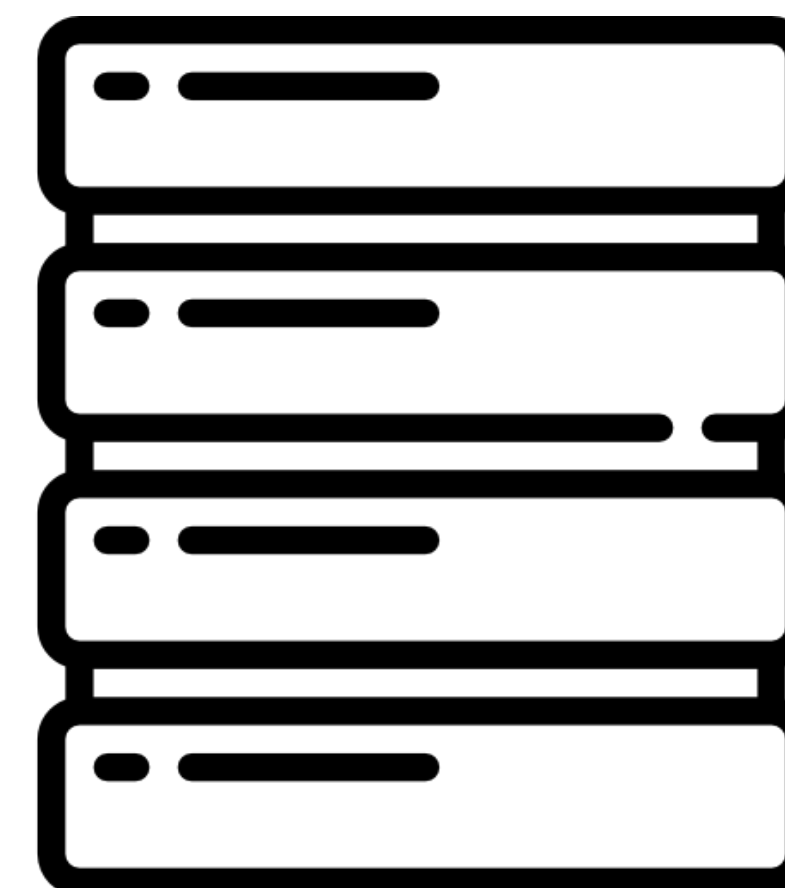
serviceworker.js



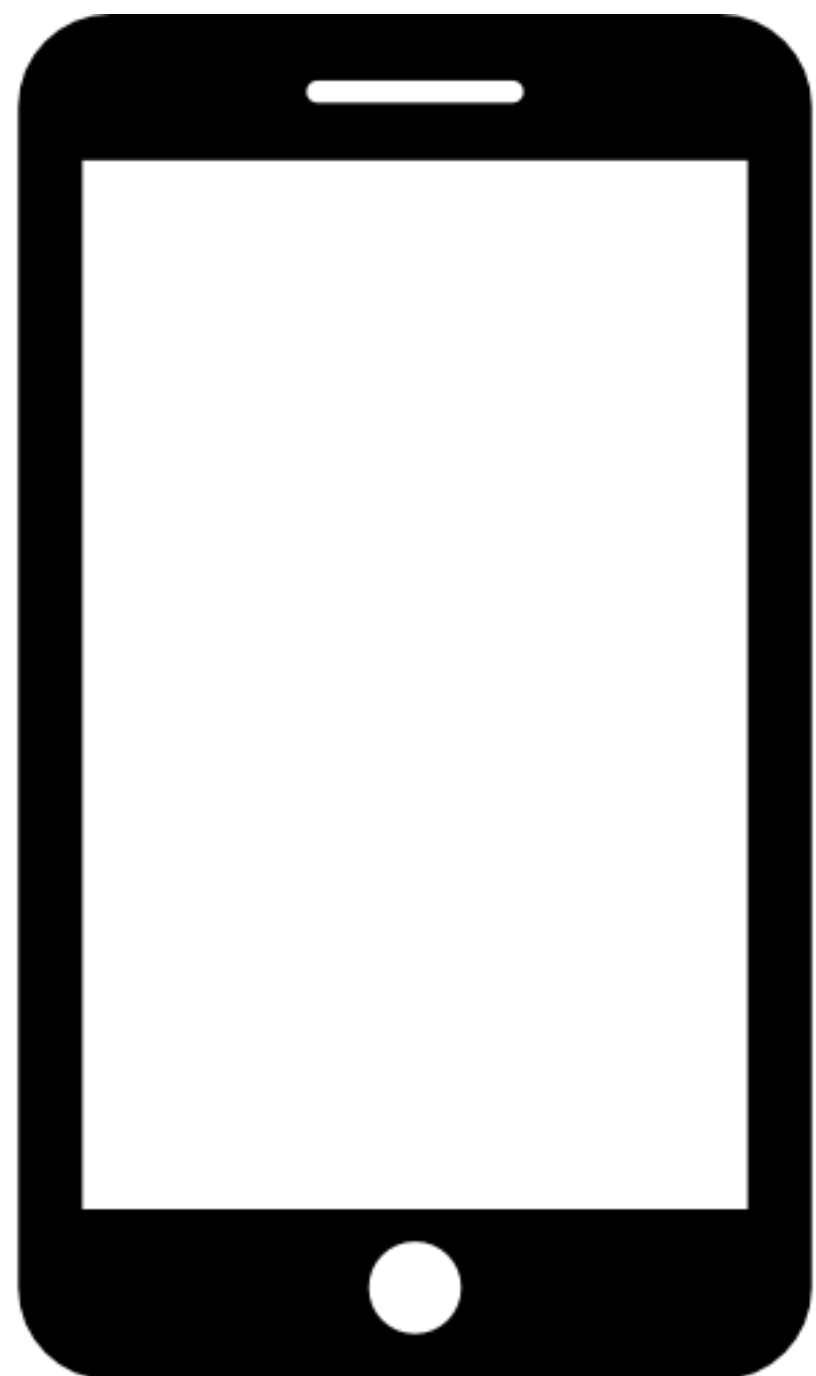
example.com



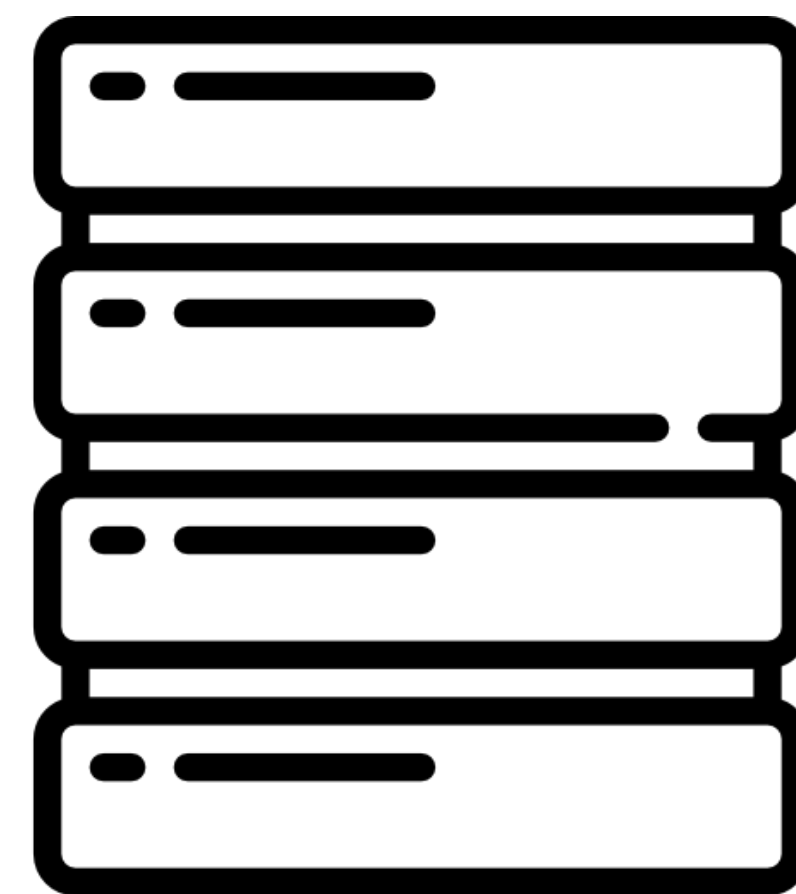
serviceworker.js



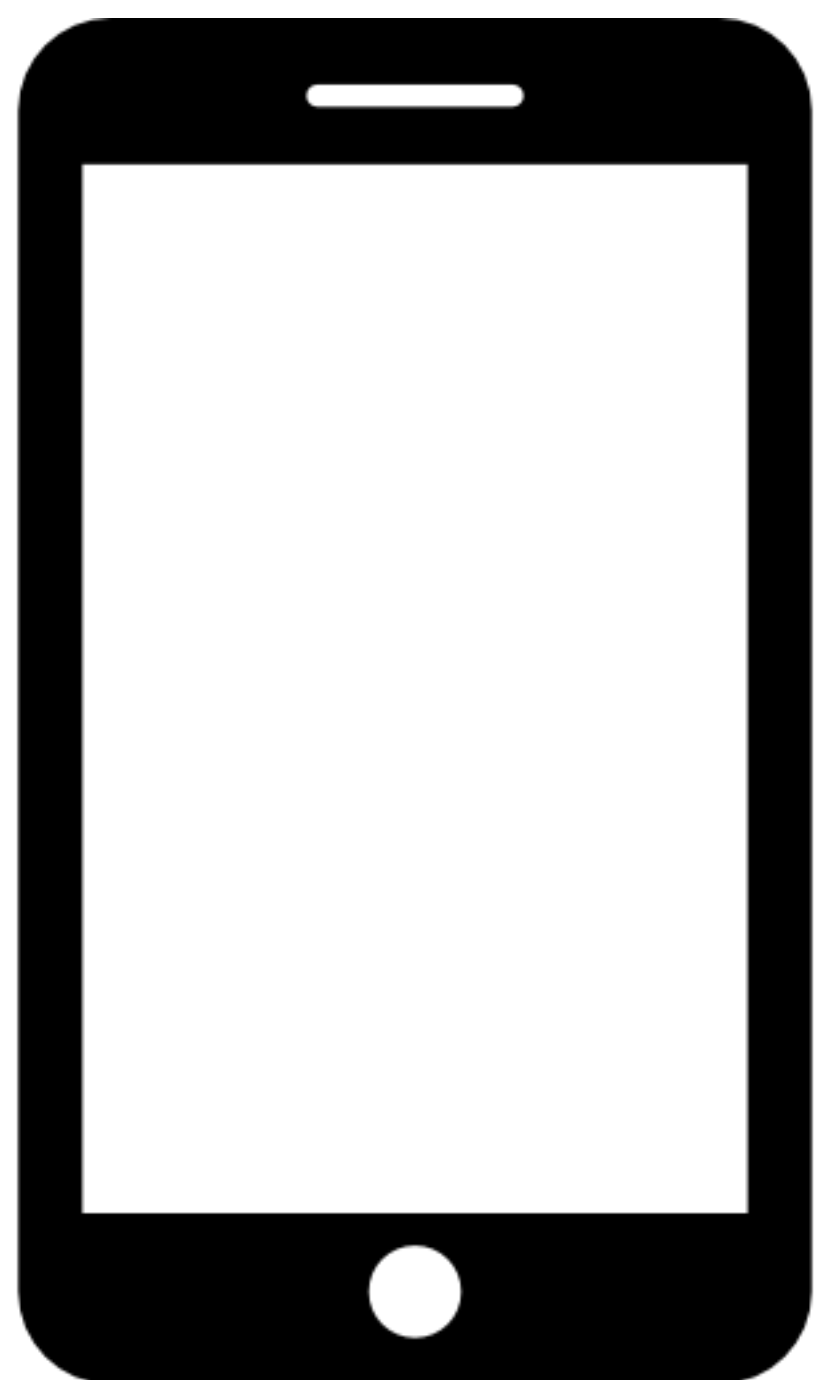
example.com



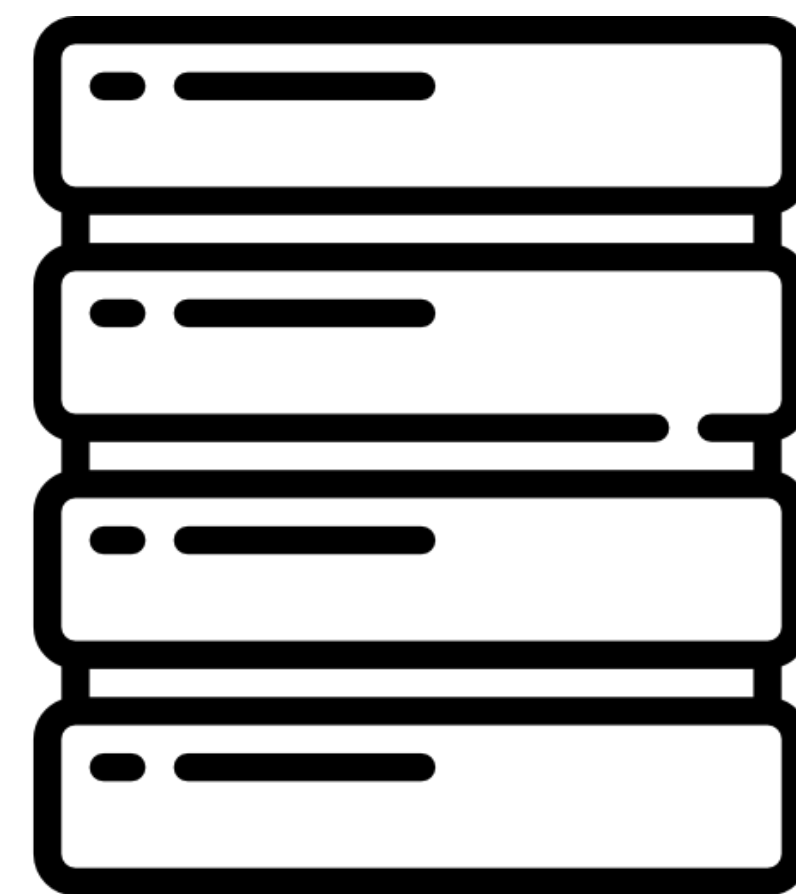
serviceworker.js



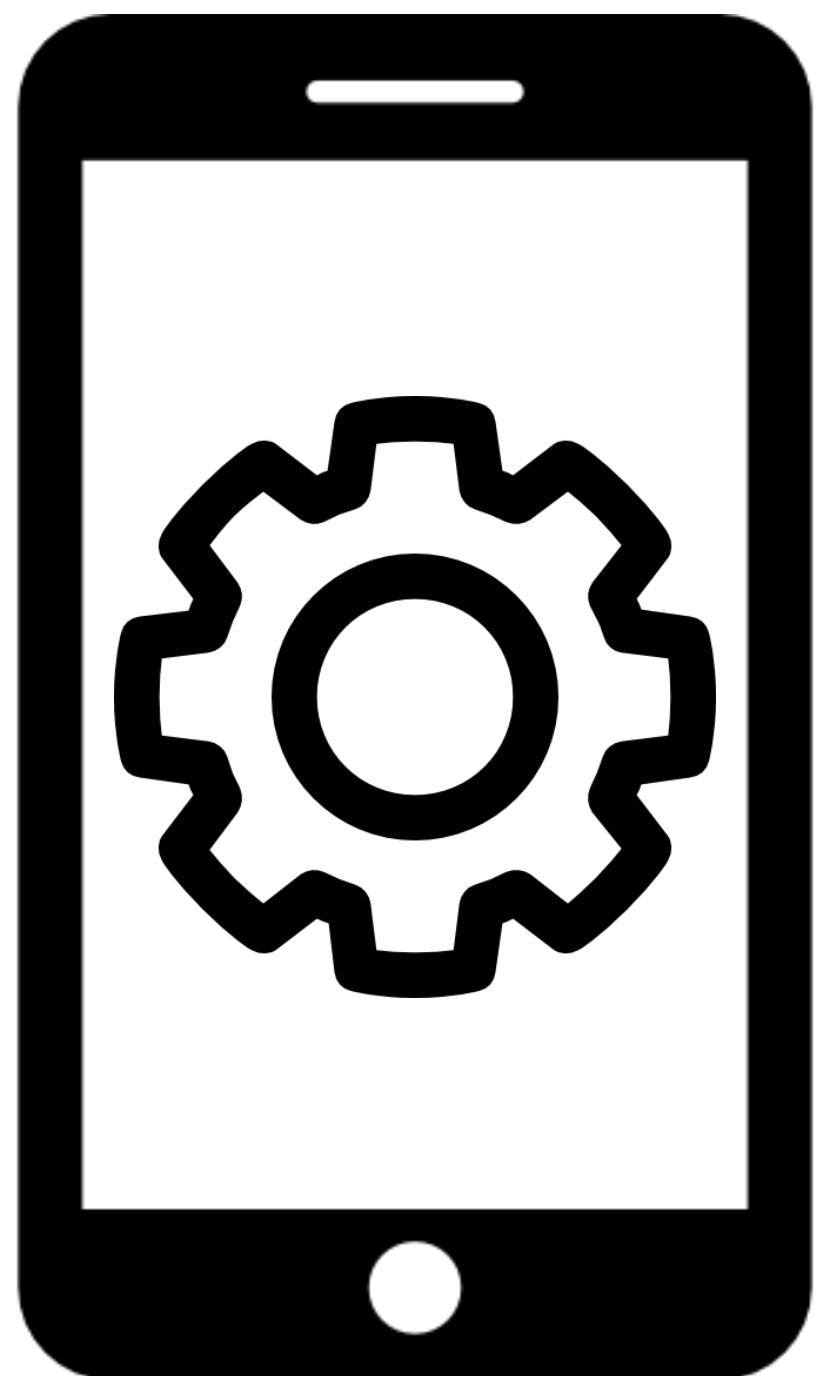
example.com



serviceworker.js



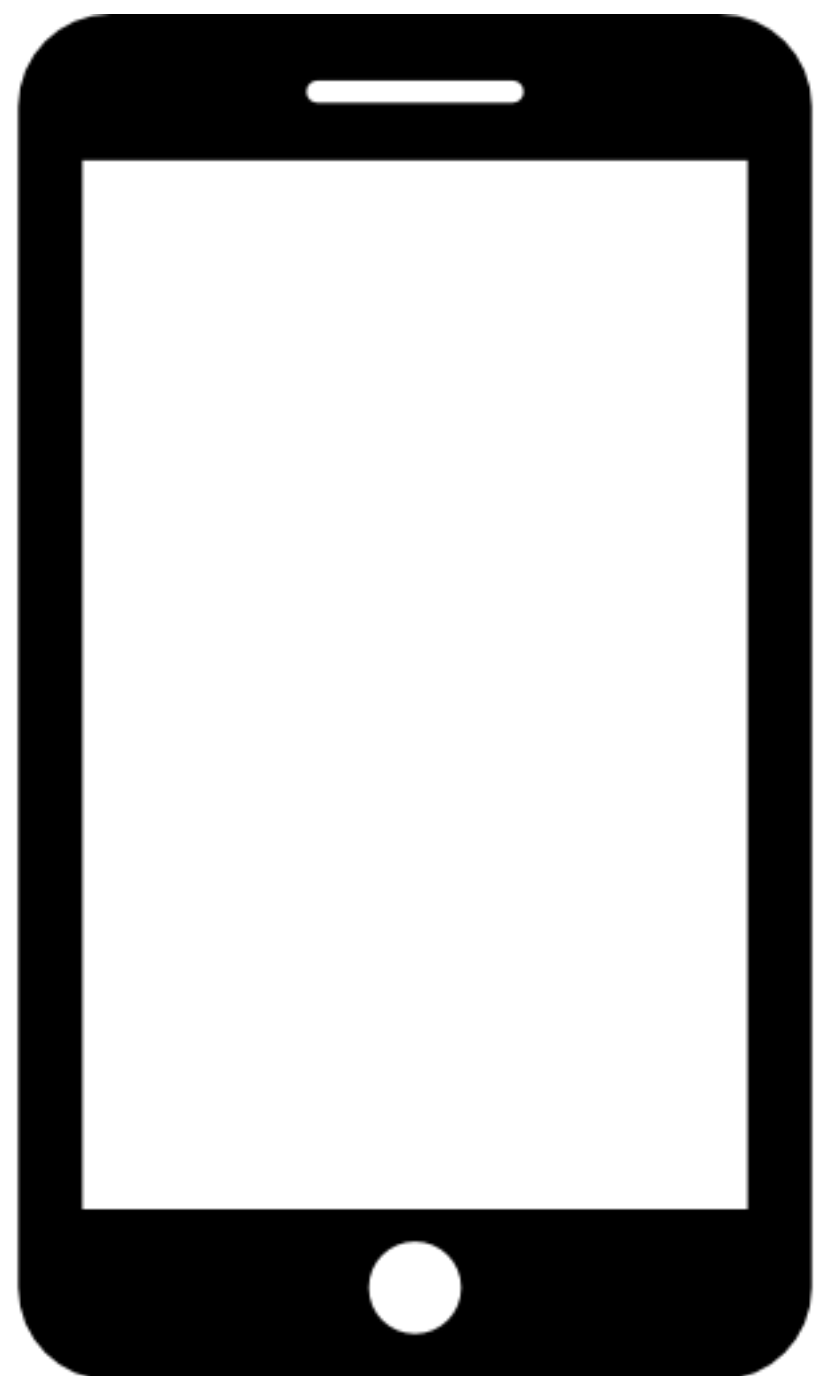
example.com



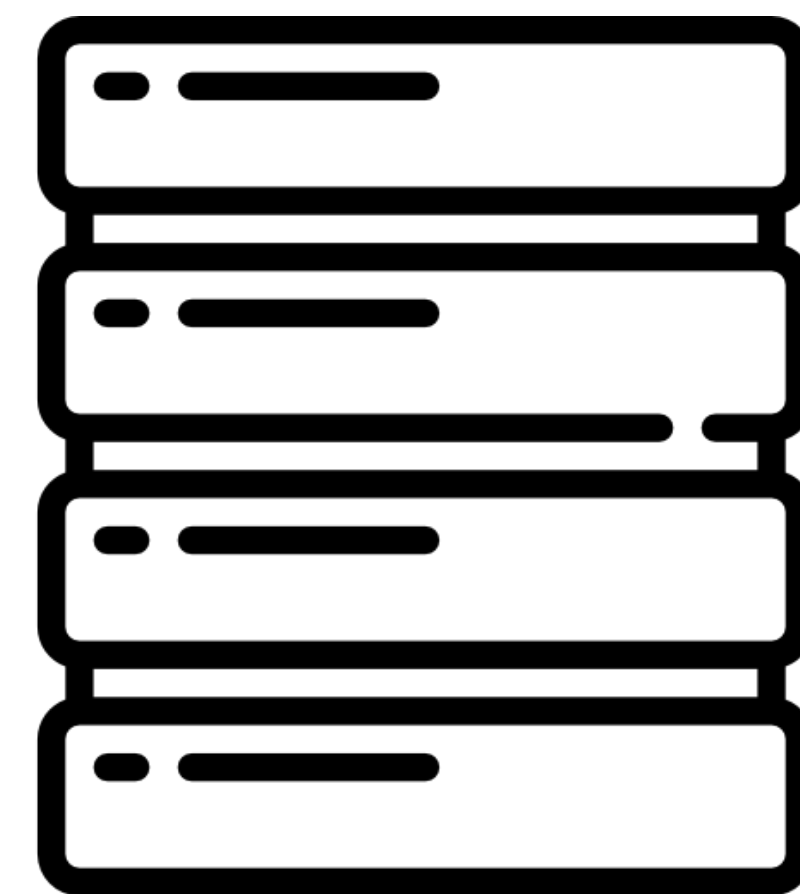
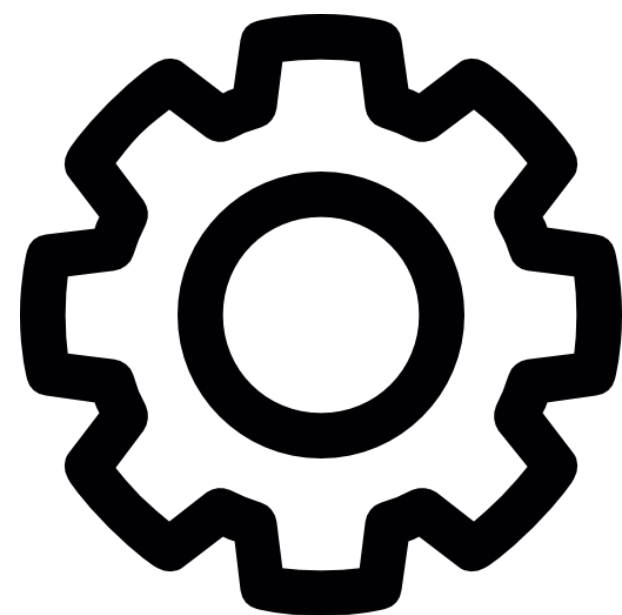
serviceworker.js



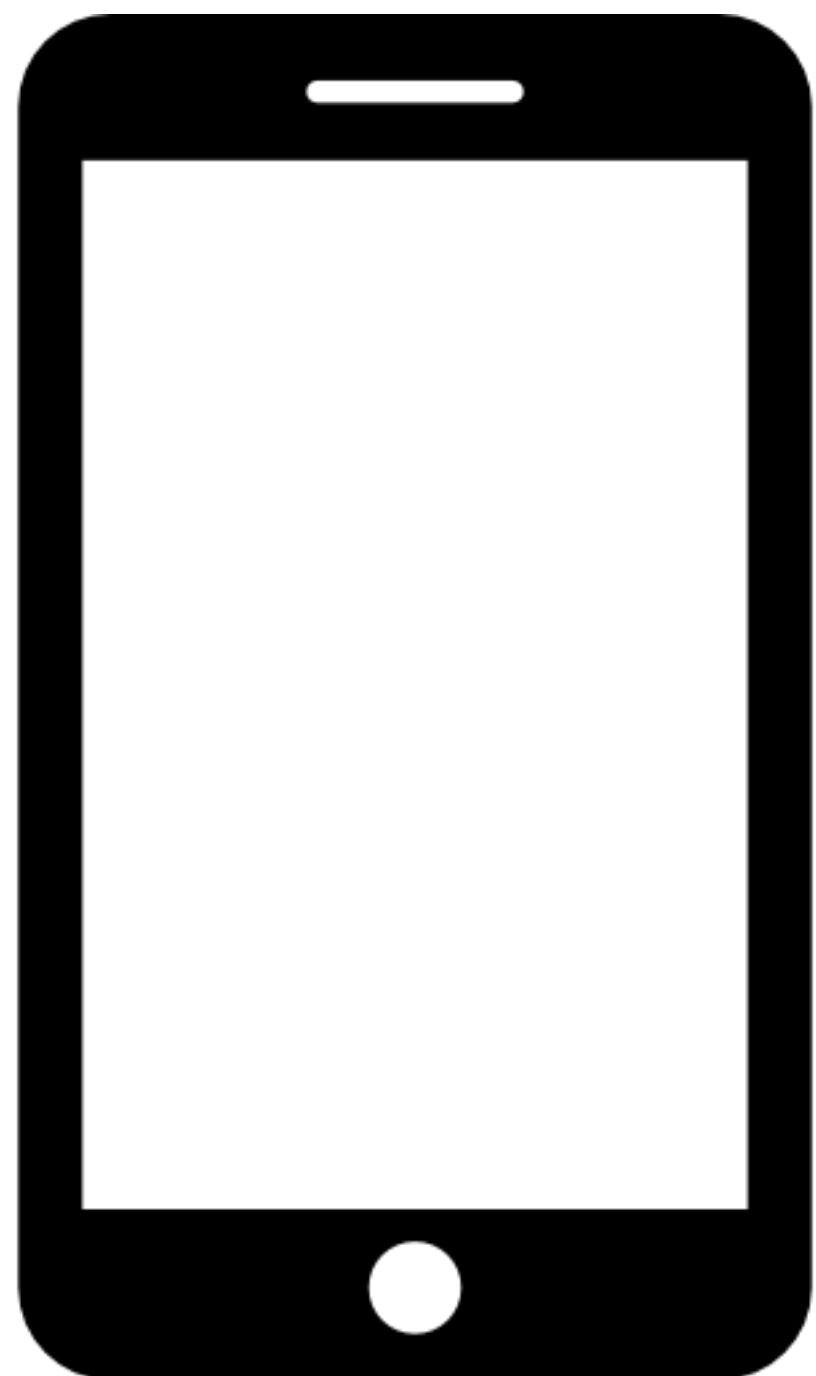
example.com



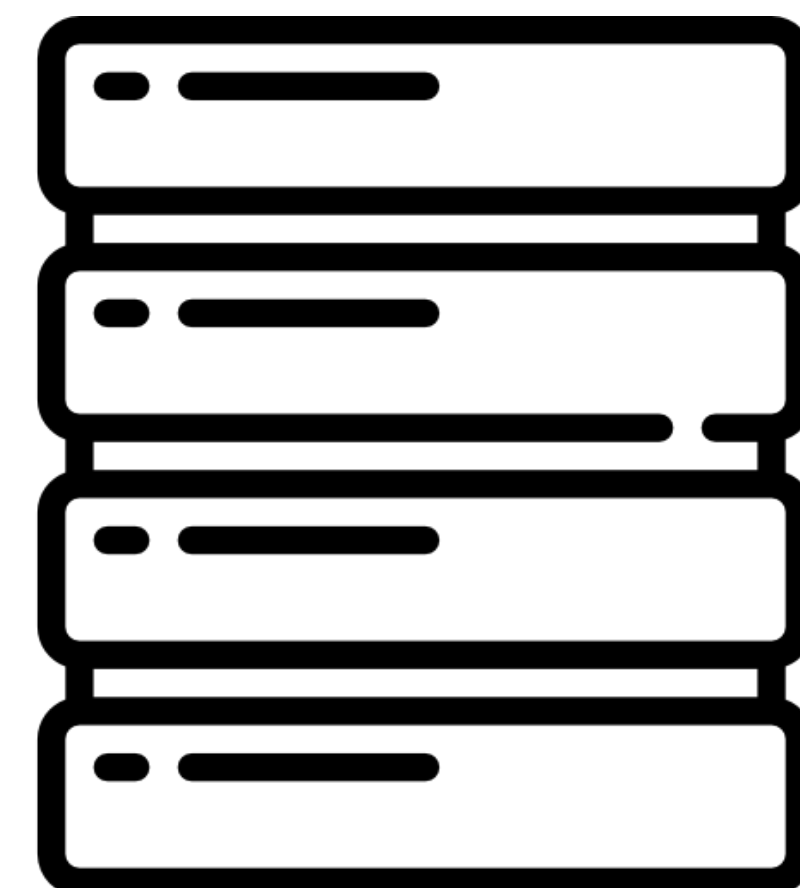
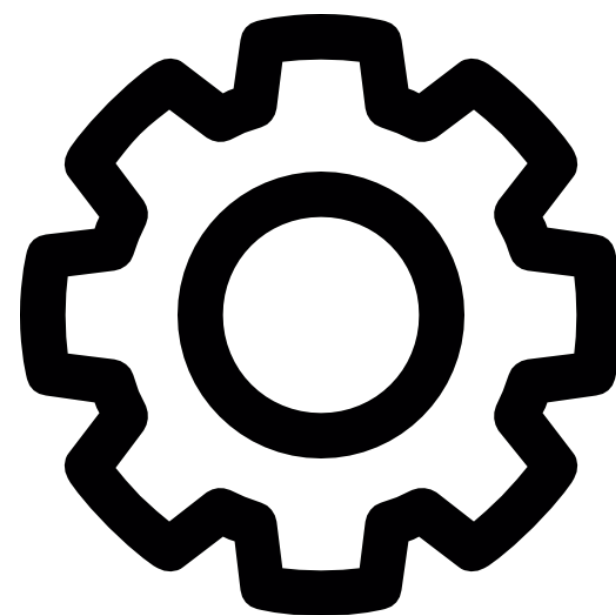
`serviceworker.js`



`example.com`



serviceworker.js



<https://example.com>

https

register

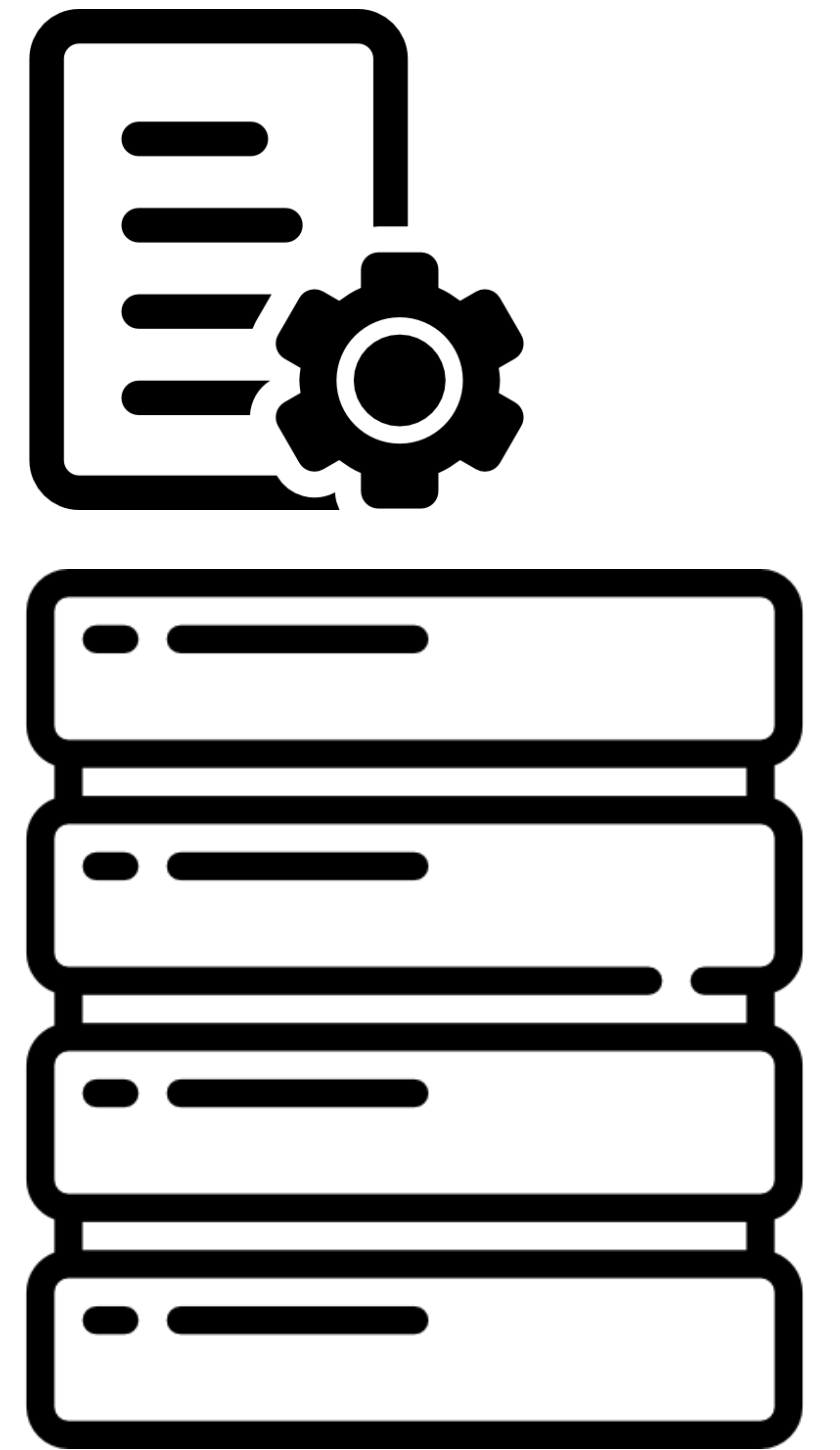
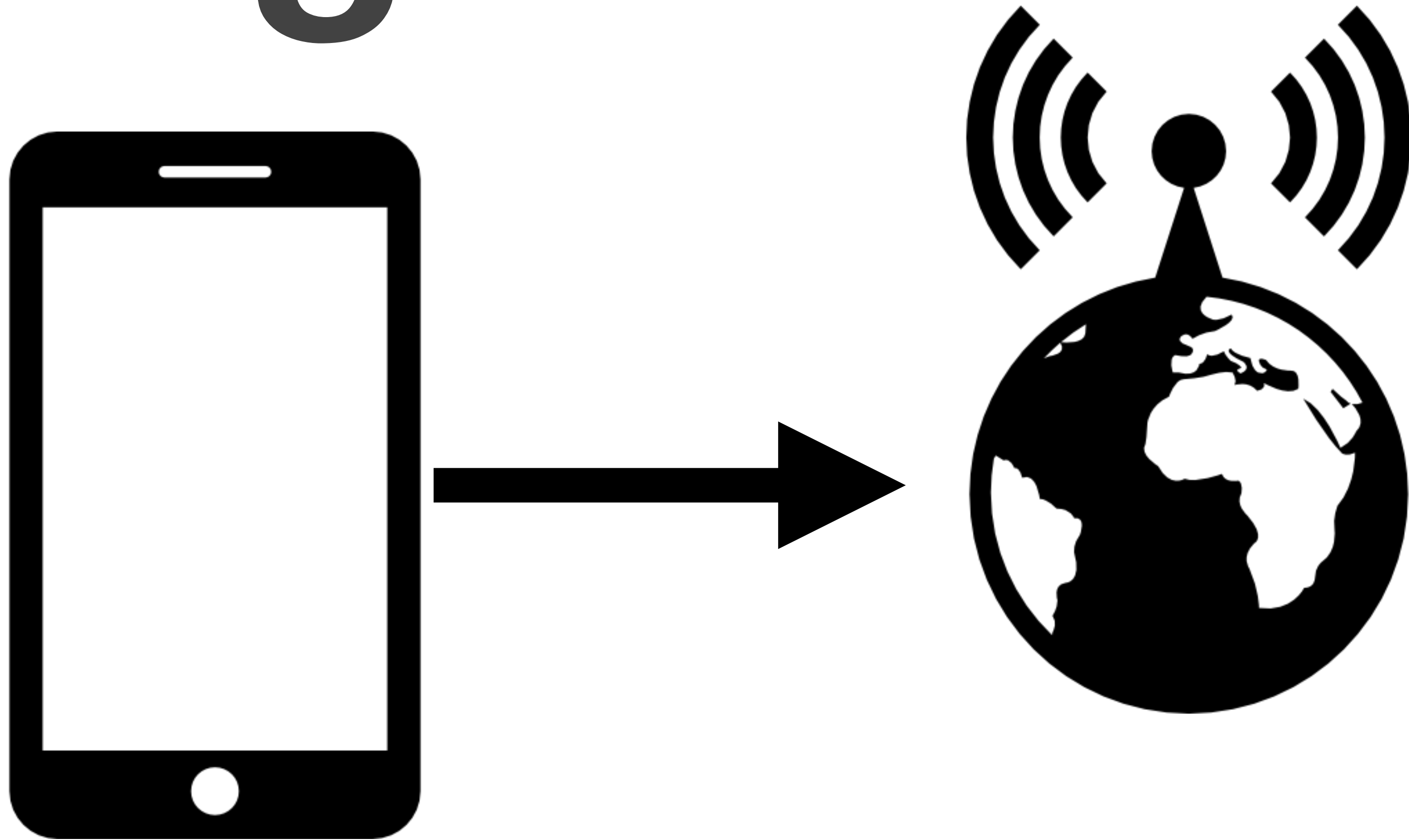
install

active

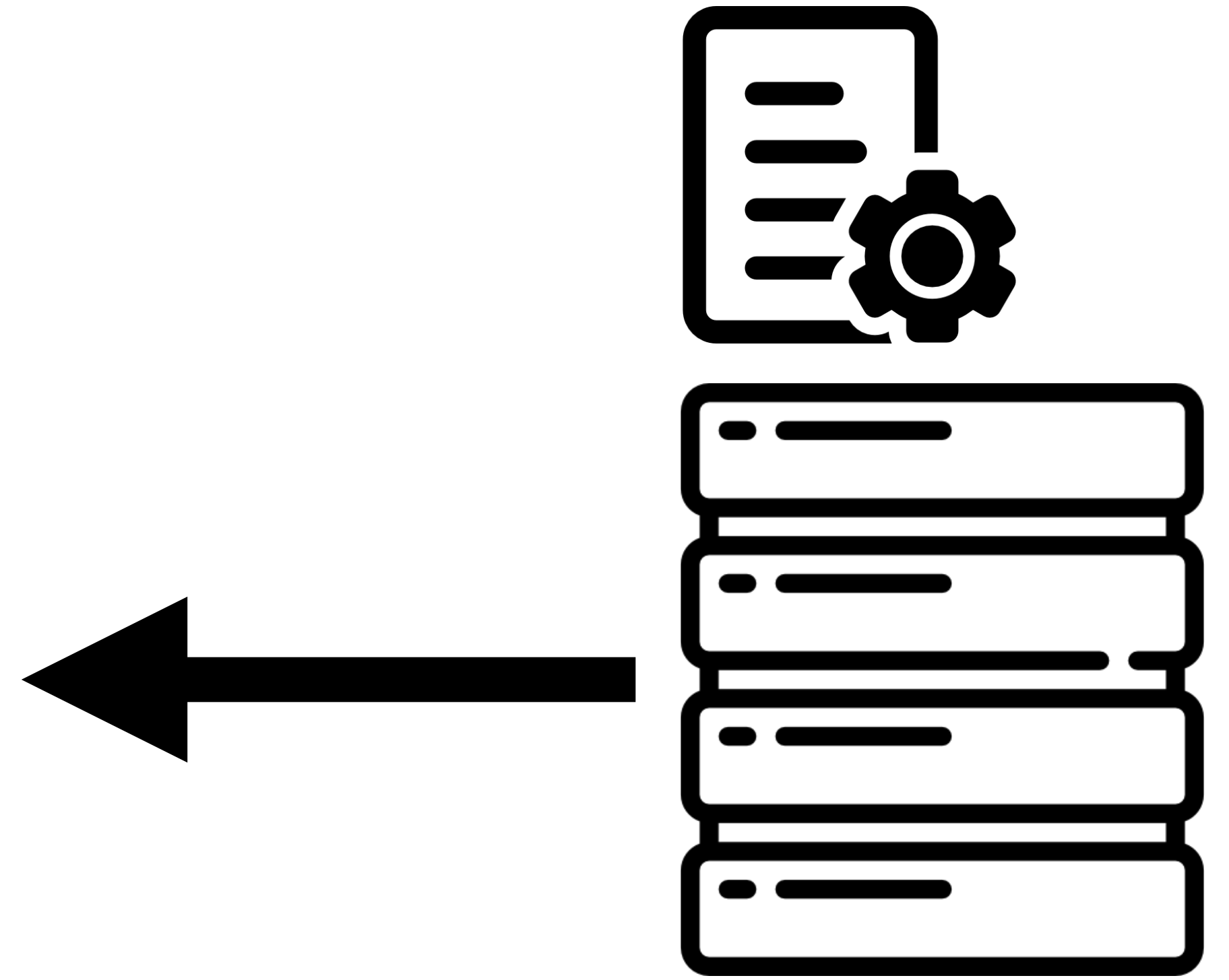
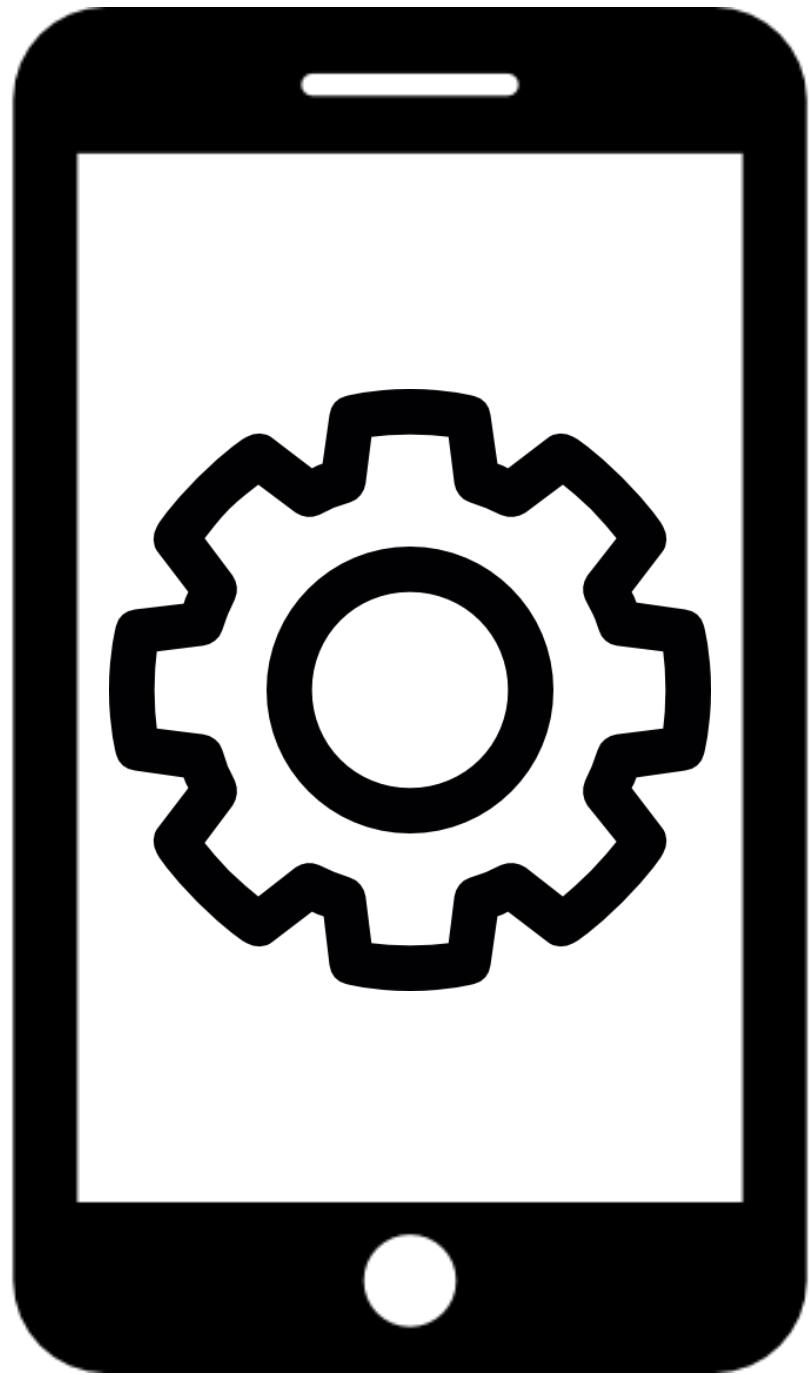
update

life cycle

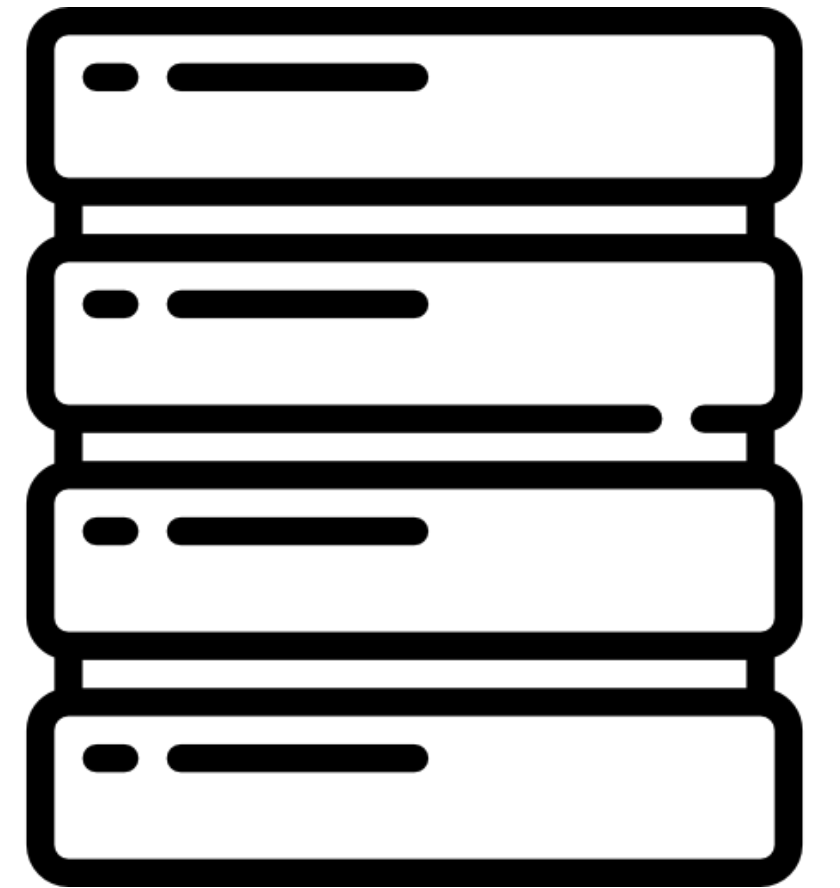
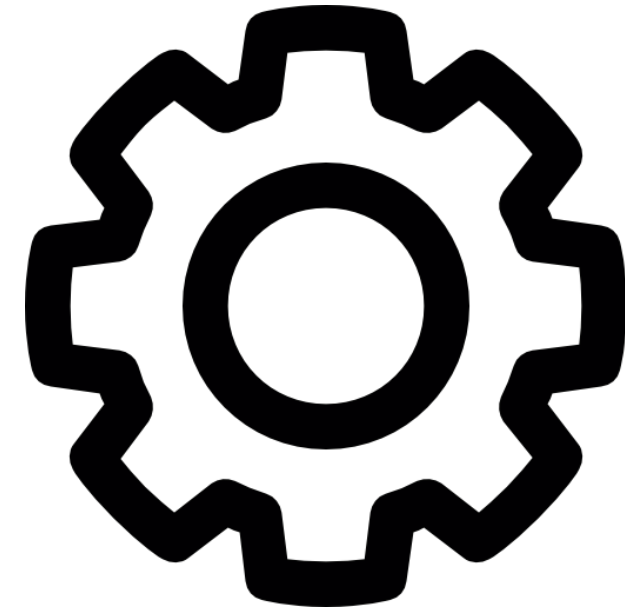
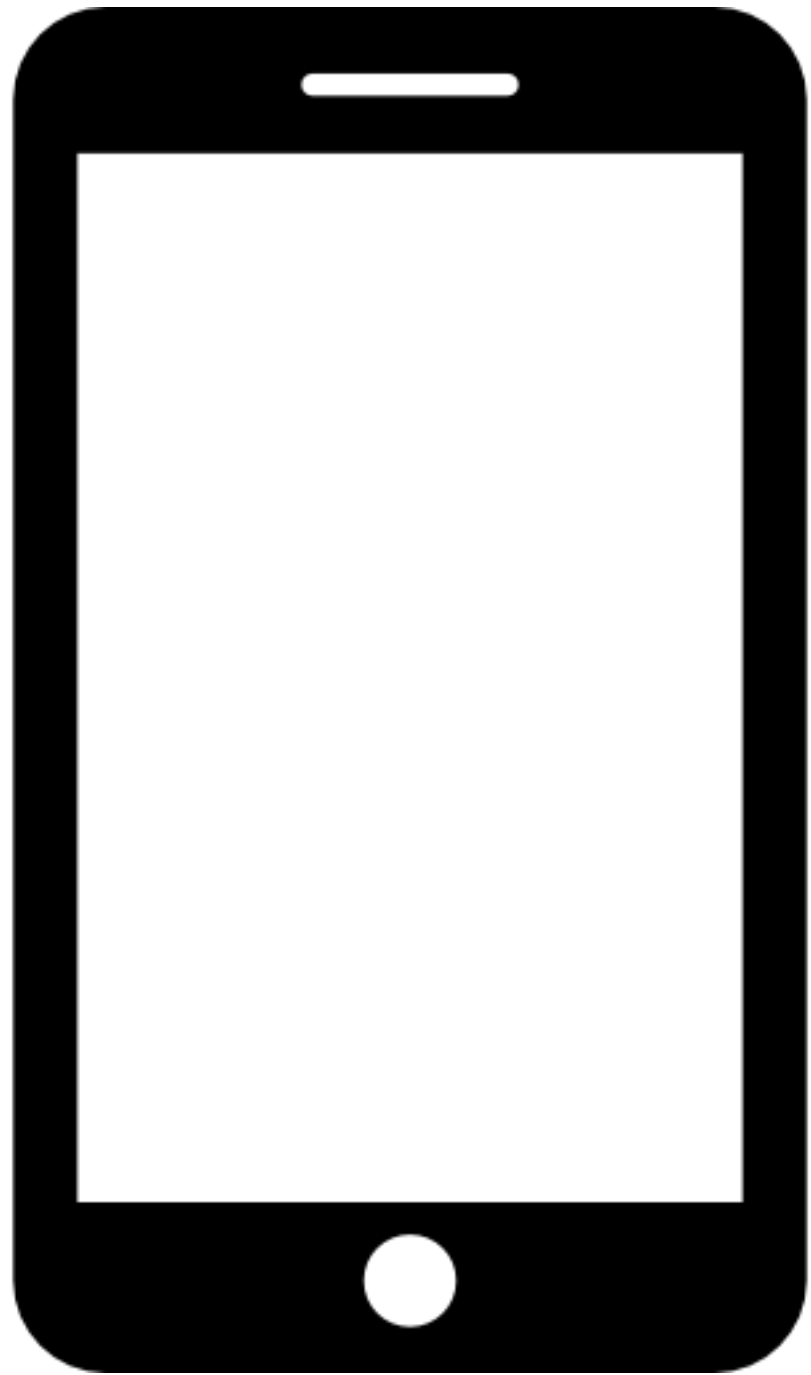
register



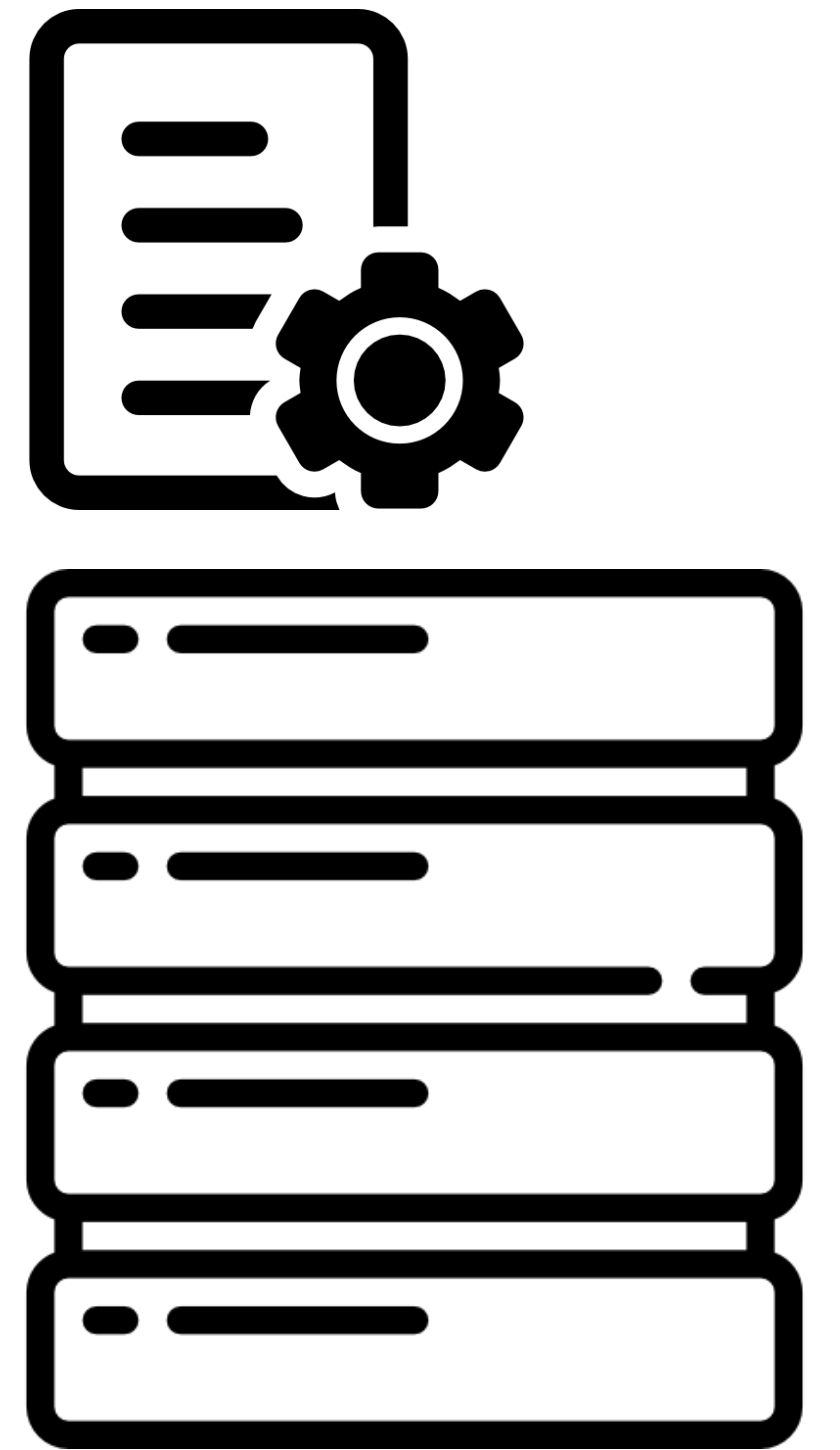
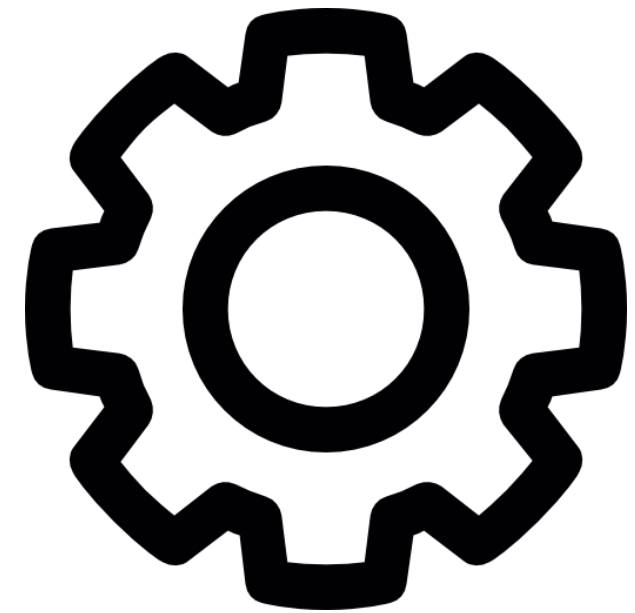
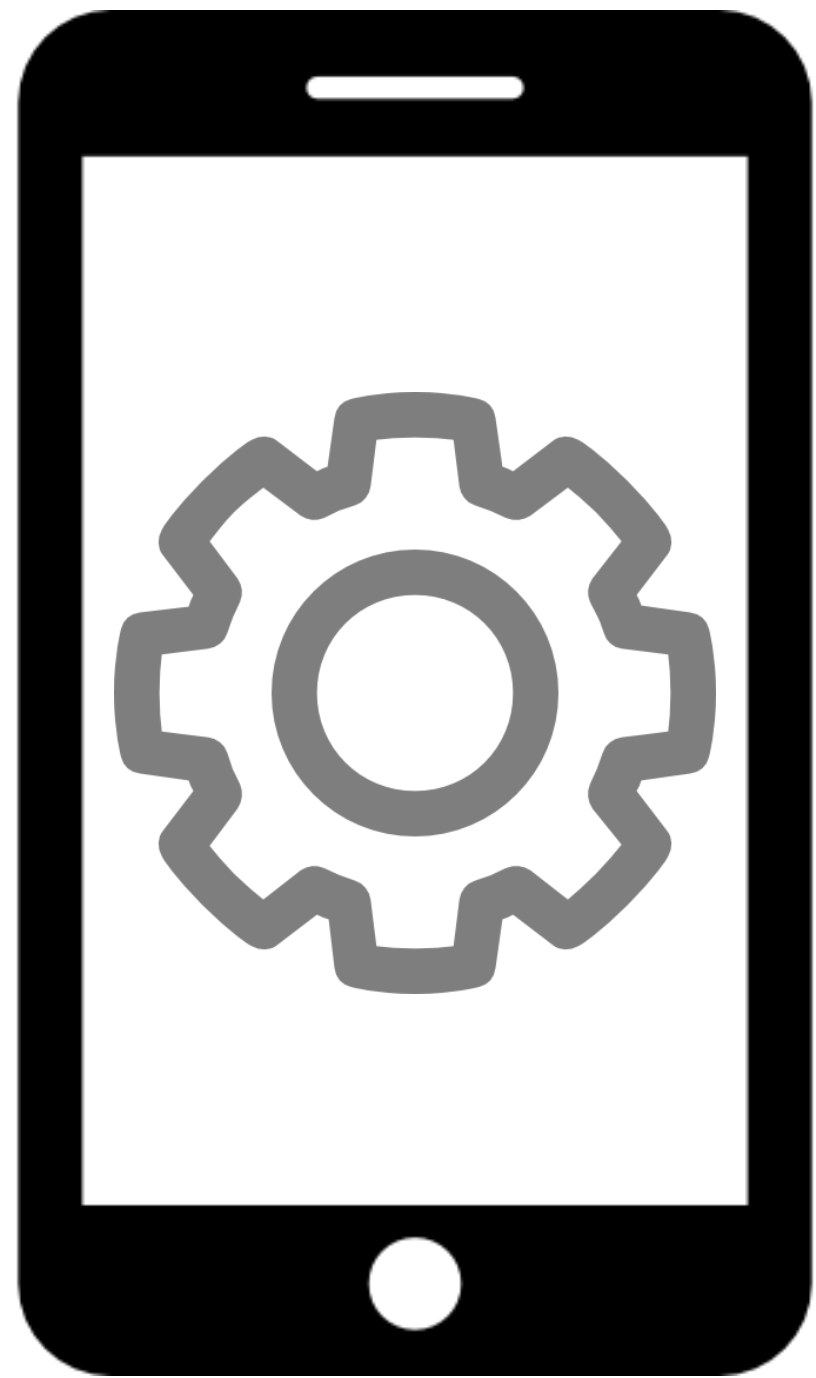
install



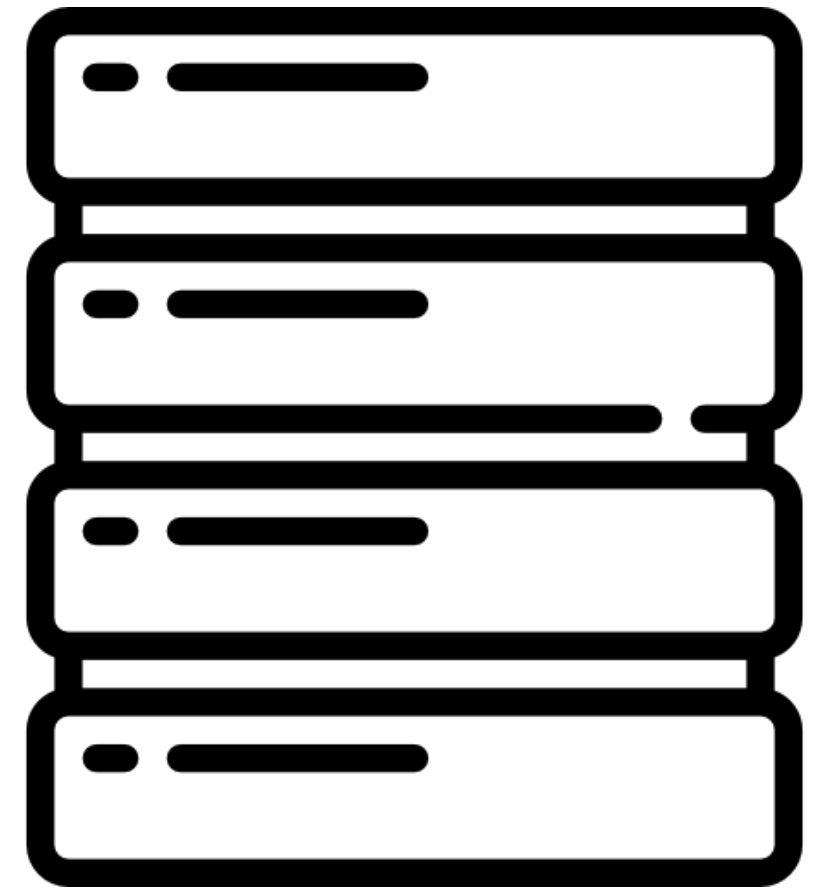
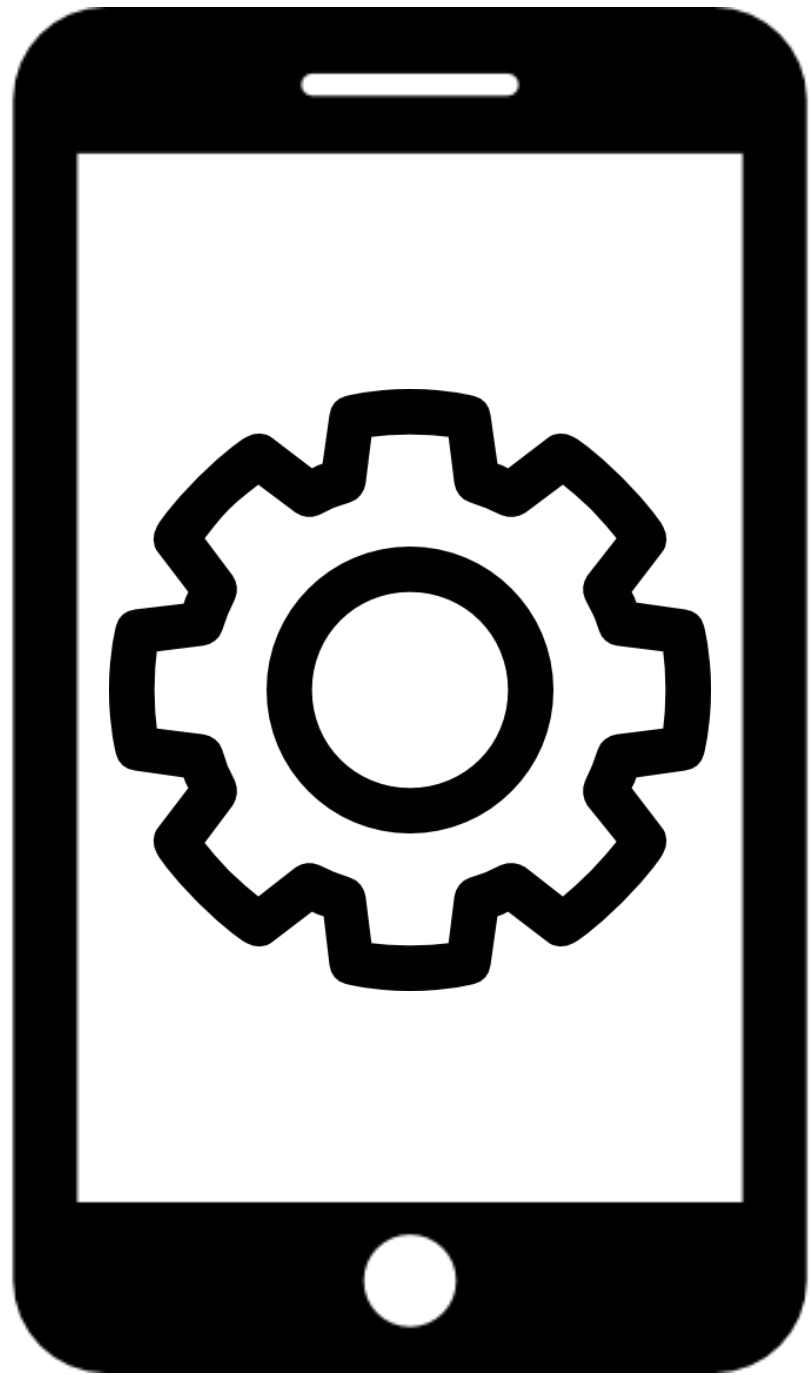
active



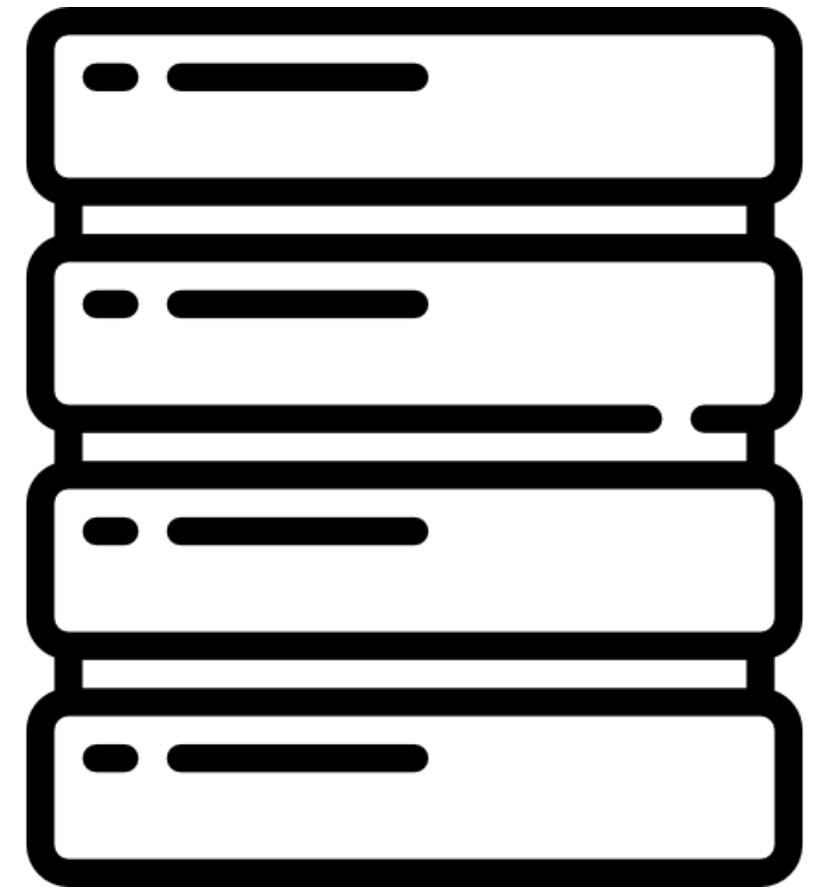
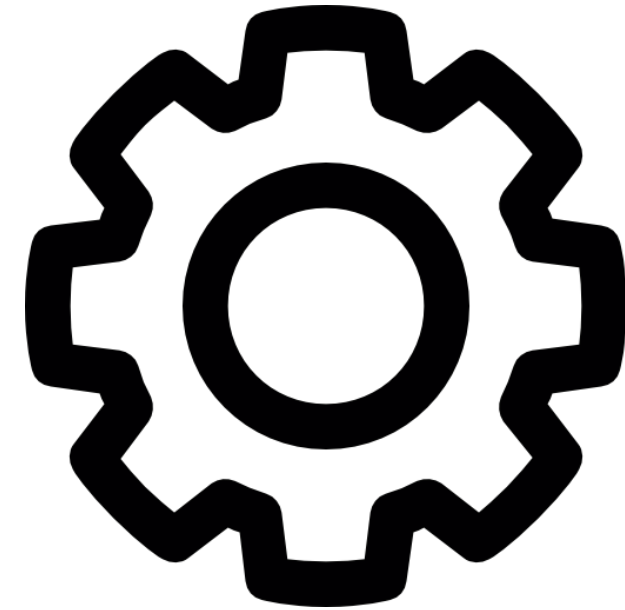
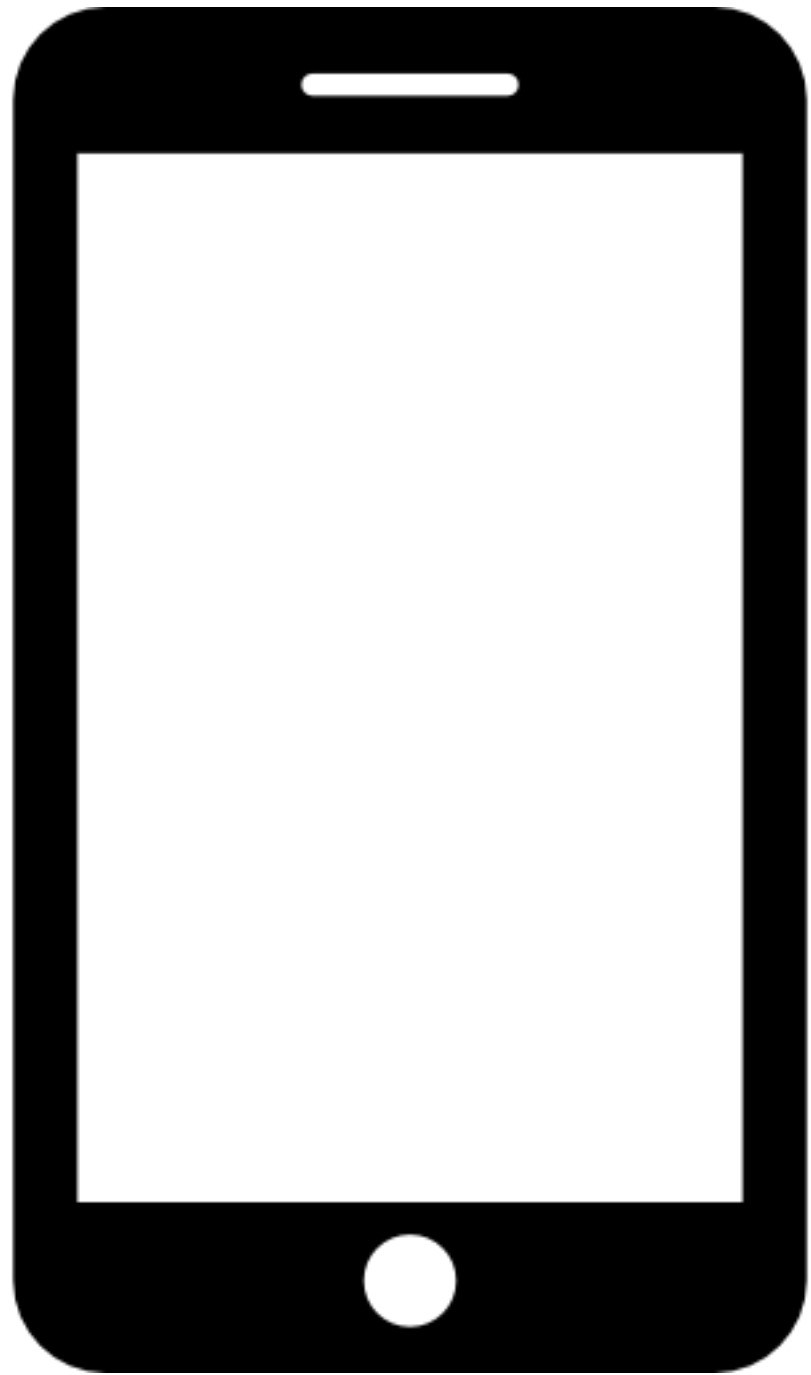
update



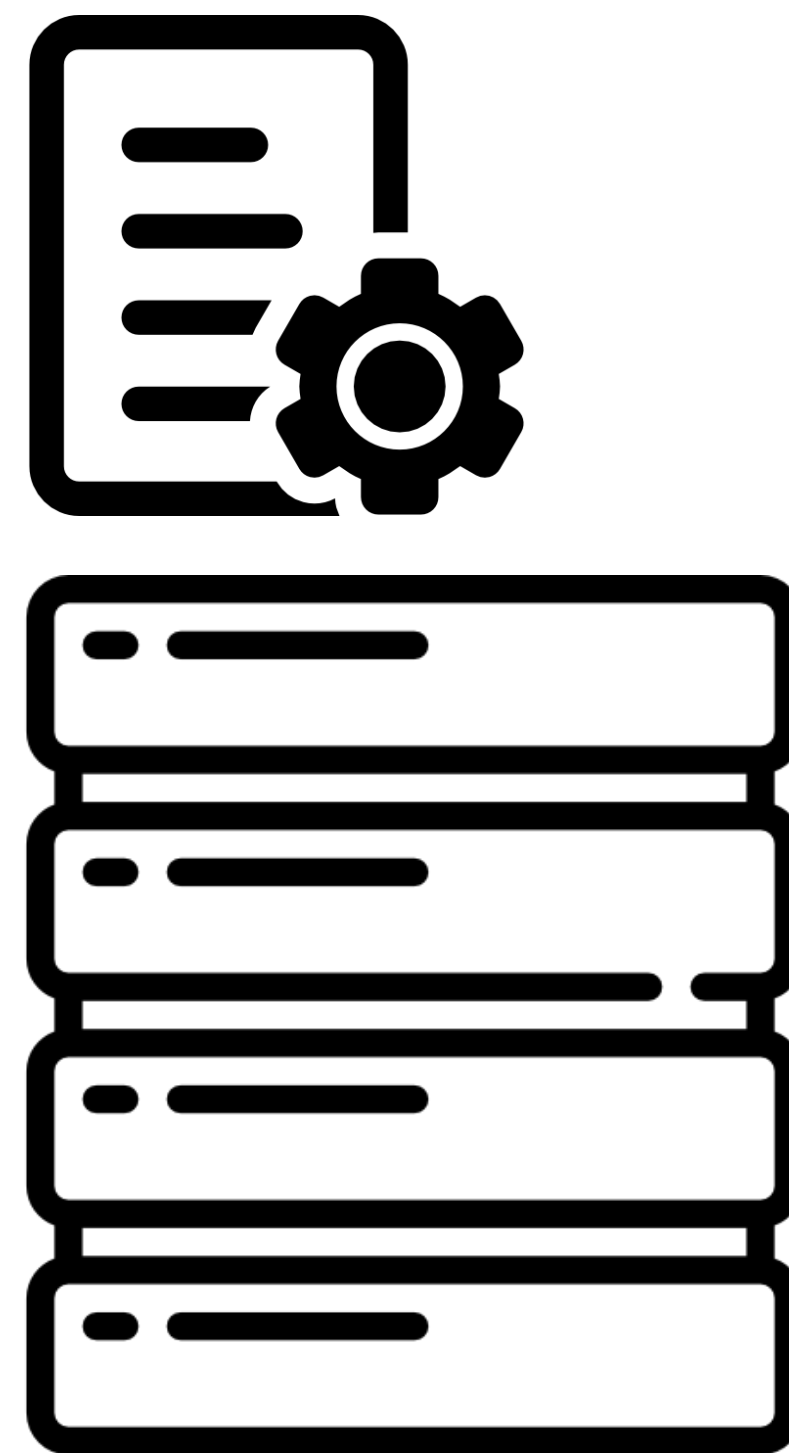
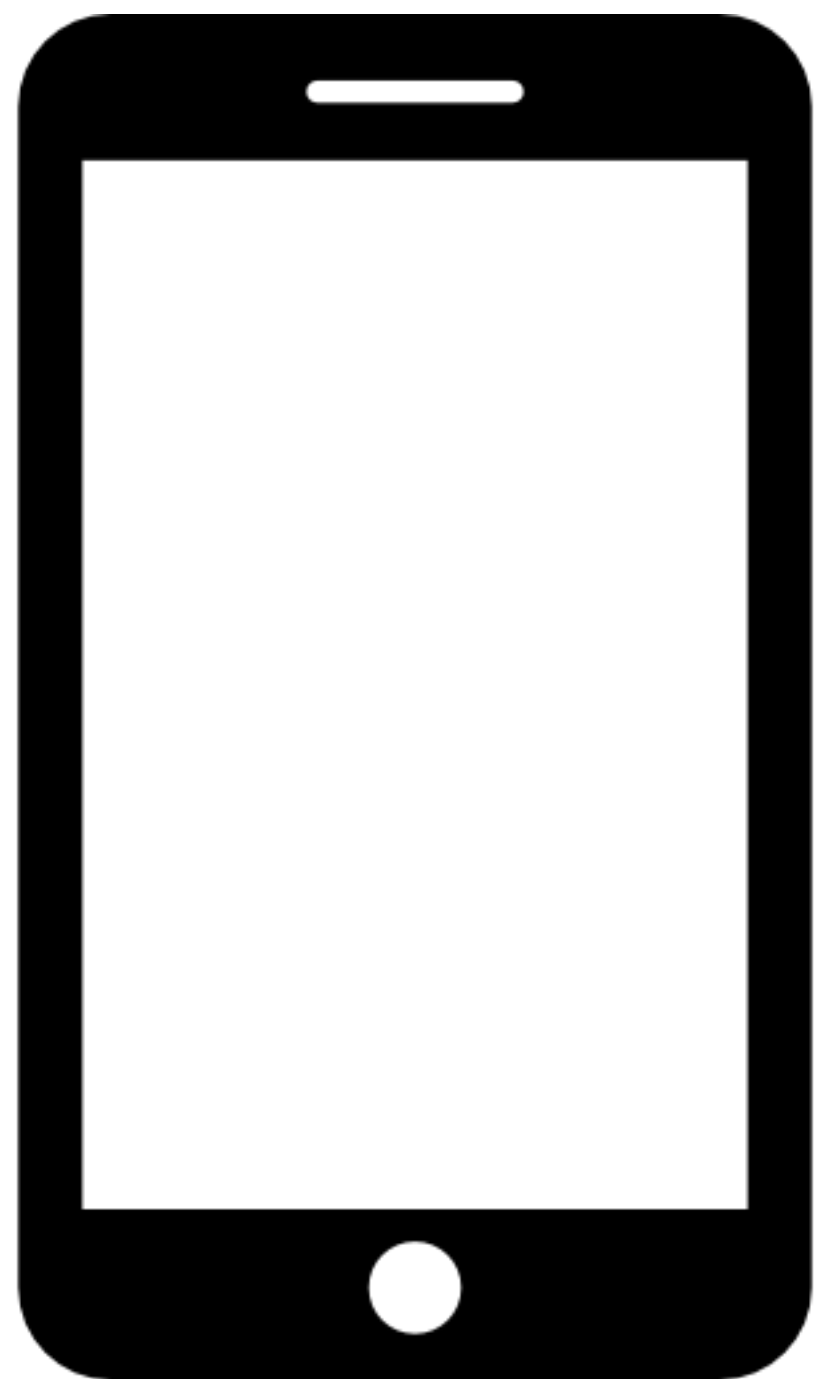
update



active



register



register

Get the JavaScript file 'serviceworker.js'

register

```
<script src="/serviceworker.js">  
</script>
```

register

```
<script src="/serviceworker.js">  
</script>
```

register

```
<script>
```

```
navigator.serviceWorker.register();
```

```
</script>
```

register

```
<script>  
navigator.serviceWorker.register  
('/serviceworker.js');  
</script>
```

register

```
<script>  
navigator.serviceWorker.register  
('/assets/js/serviceworker.js');  
</script>
```

register

```
<script>  
navigator.serviceWorker.register  
(('/assets/js/serviceworker.js'));  
</script>
```

register

```
<script>  
navigator.serviceWorker.register  
('/serviceworker.js');  
</script>
```


register

```
<script>  
if (navigator.serviceWorker) {  
  navigator.serviceWorker.register  
    ('/serviceworker.js');  
}  
</script>
```

register

Get the JavaScript file 'serviceworker.js'

```
<script>
```

```
if (navigator.serviceWorker) {  
    navigator.serviceWorker.register  
    ('/serviceworker.js');  
}
```

```
</script>
```

register

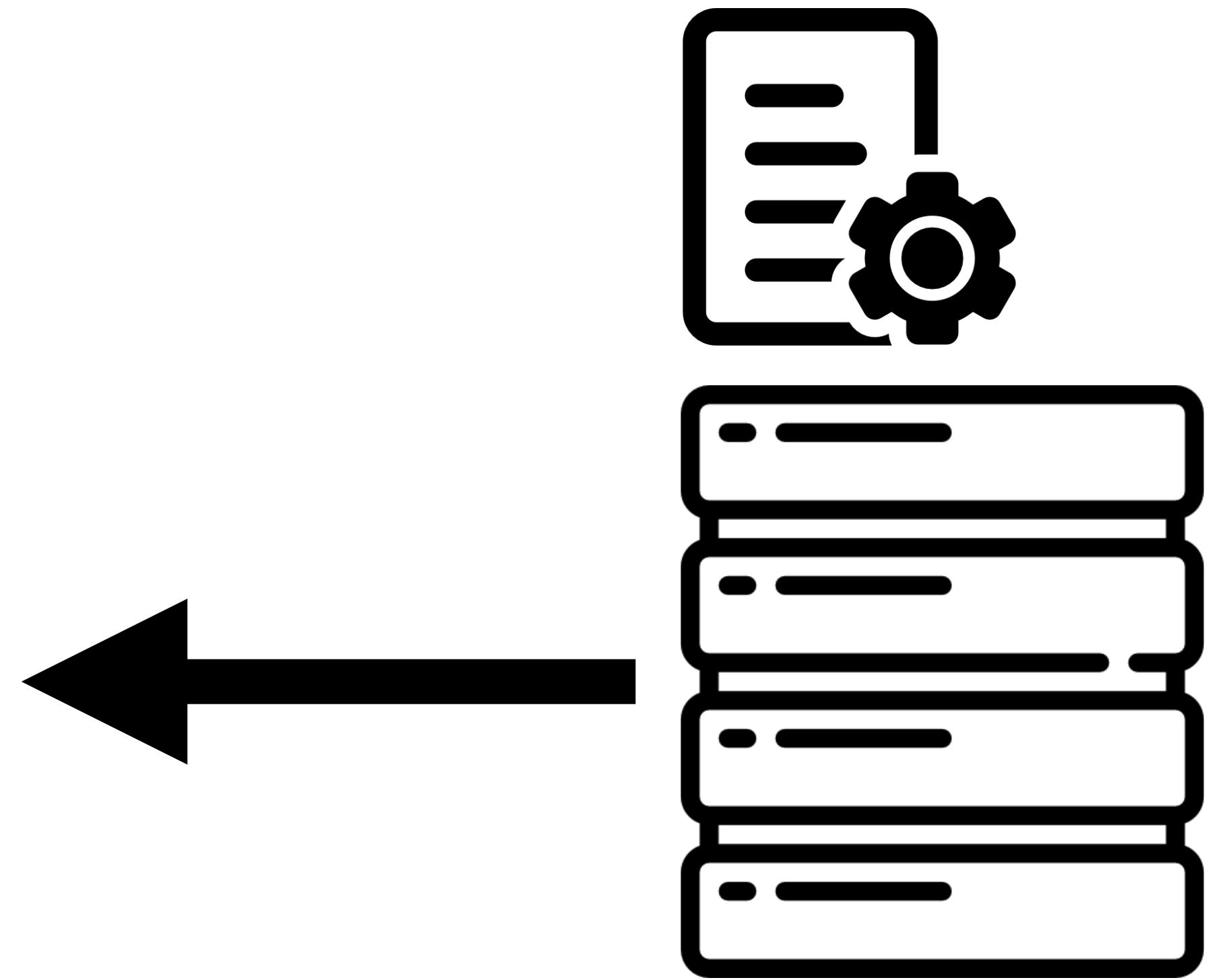
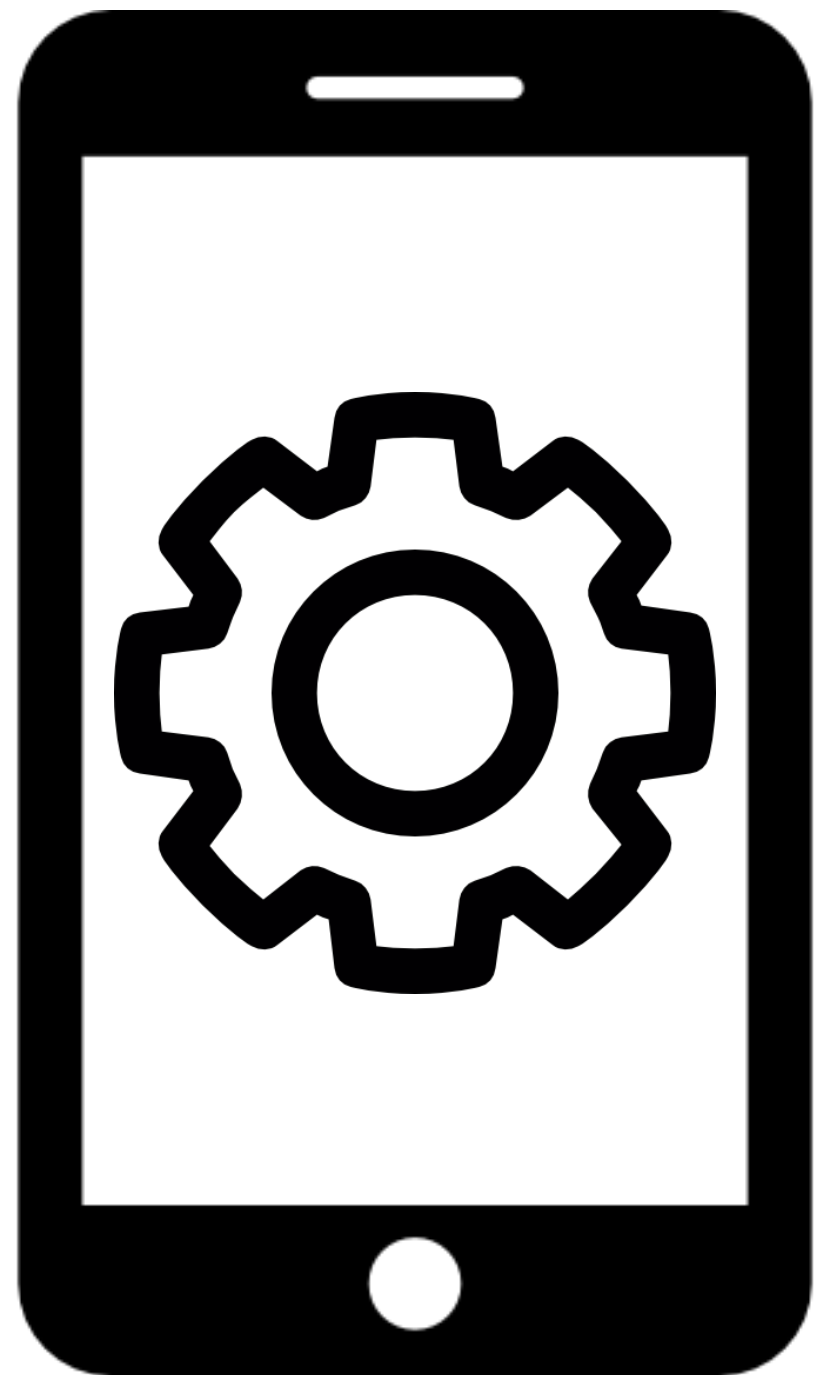
Install the JavaScript file 'serviceworker.js'

```
<script>
```

```
if (navigator.serviceWorker) {  
  navigator.serviceWorker.register  
    ('/serviceworker.js');  
}
```

```
</script>
```

install



install

caching

memory cache

browser cache

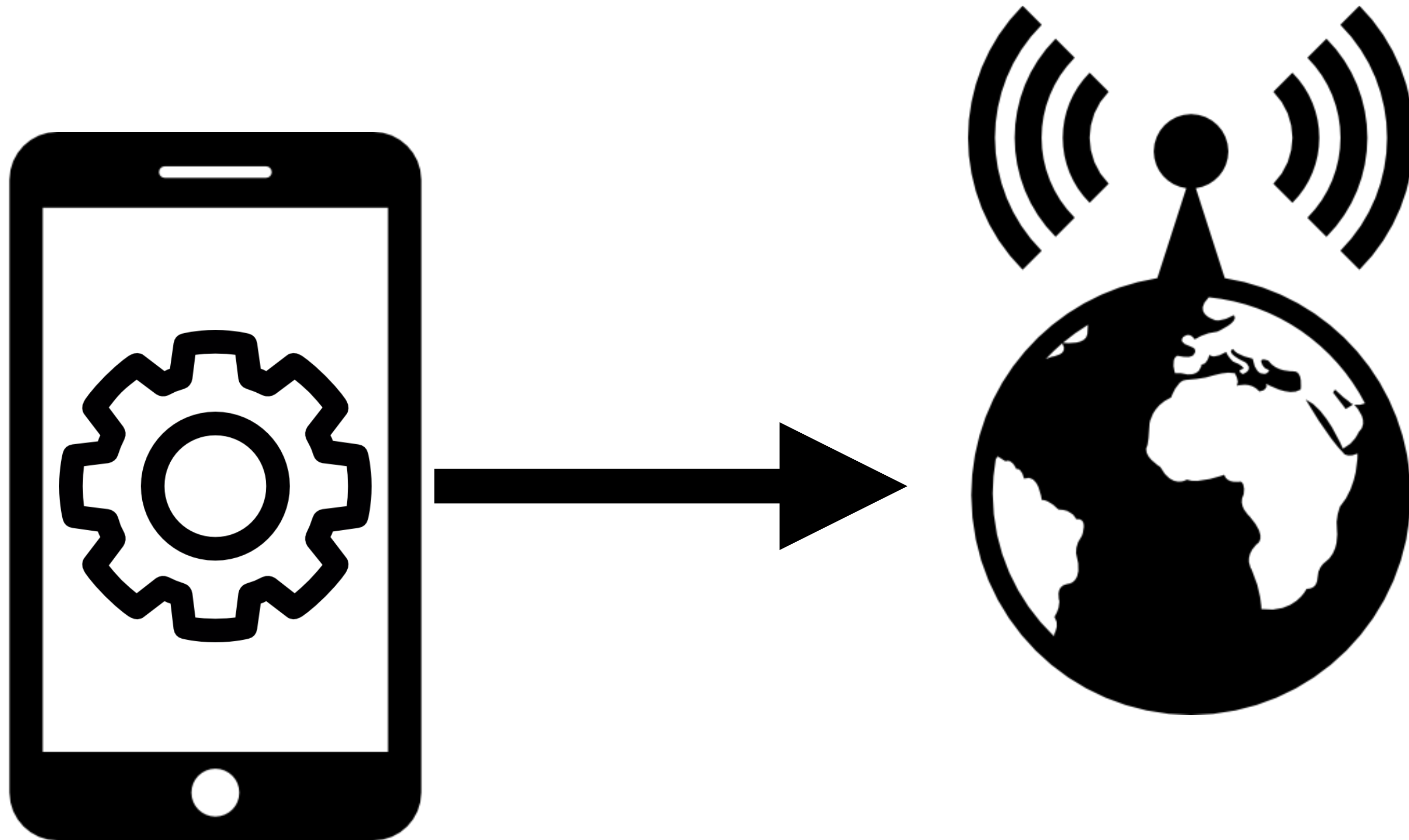
Cache API

caching

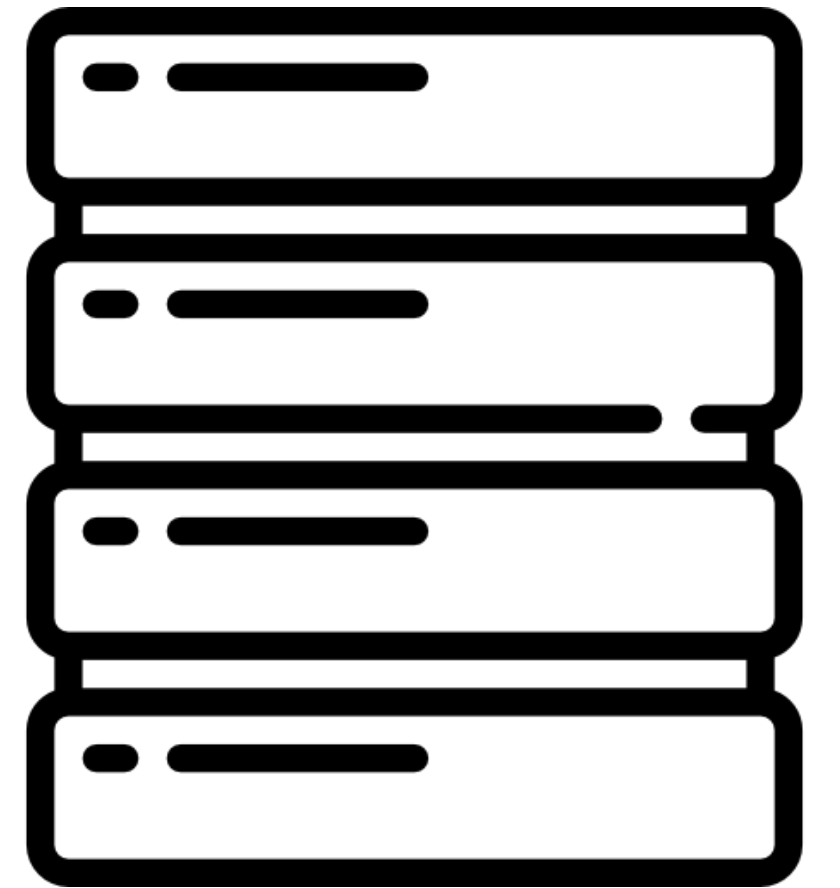
install

preaching

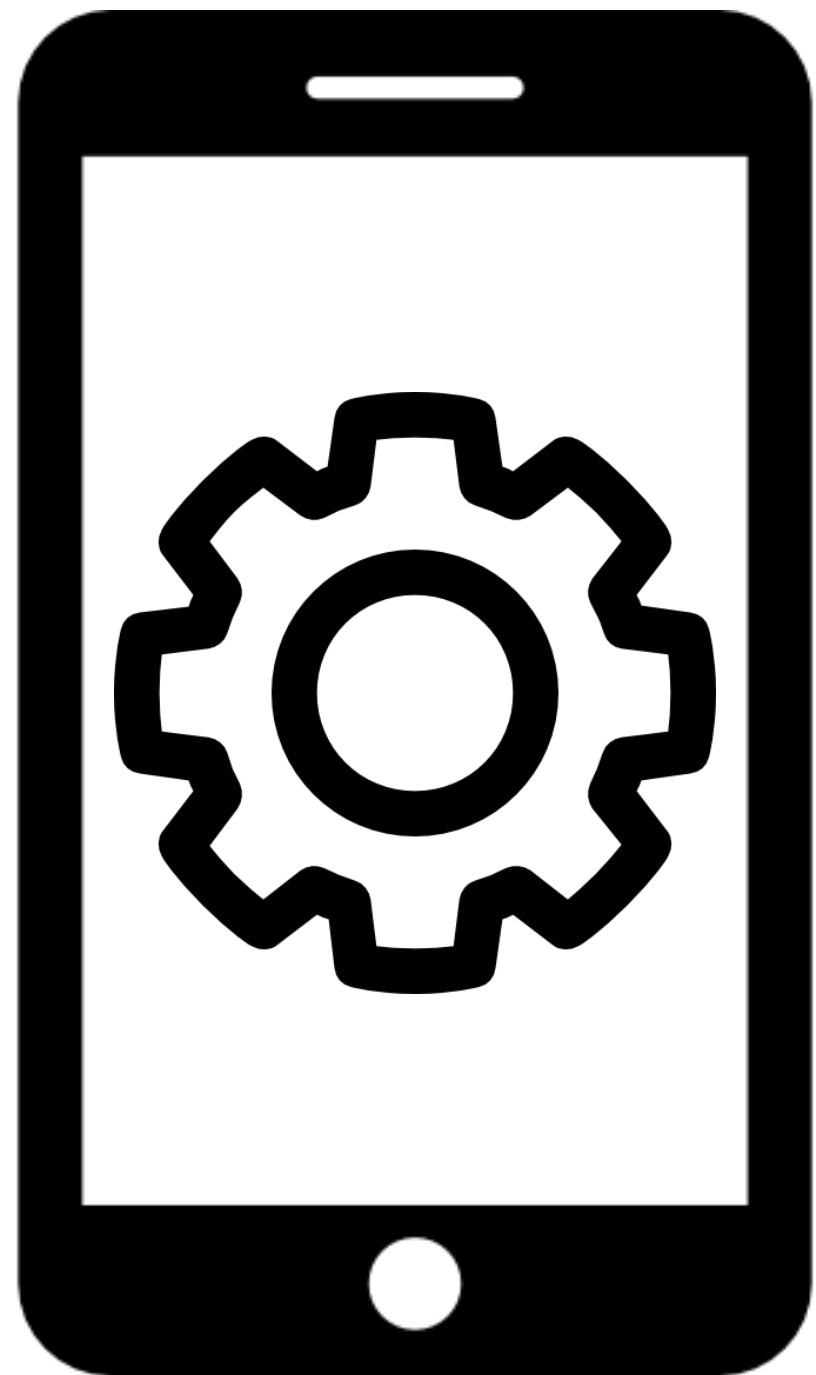
install



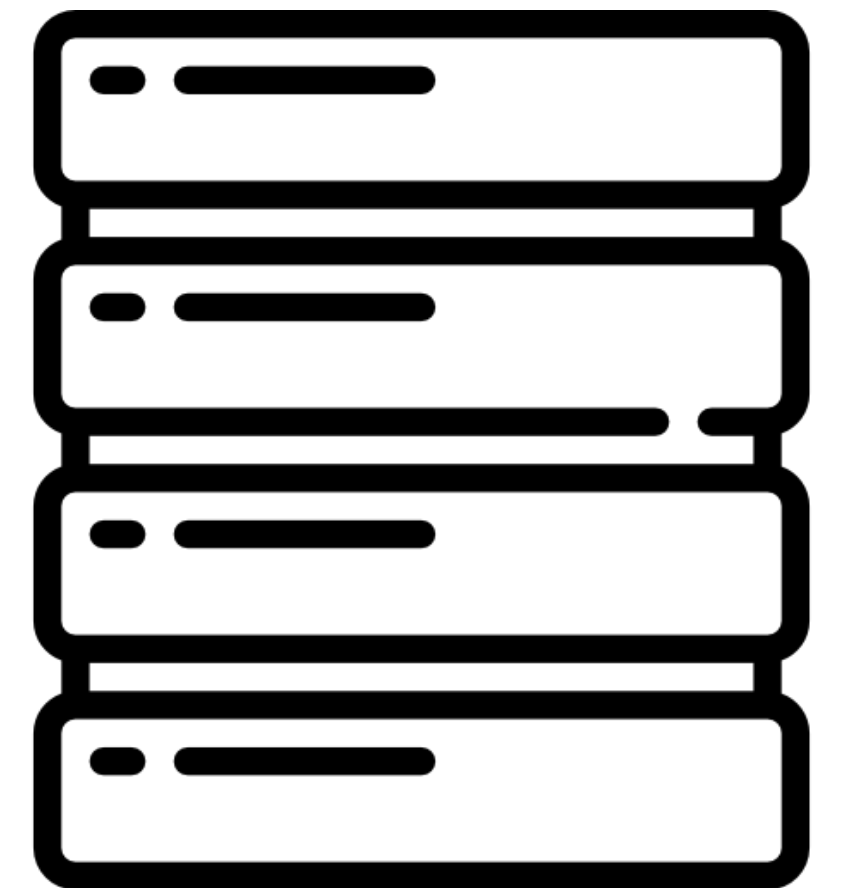
`styles.css`
`script.js`
`icon.png`



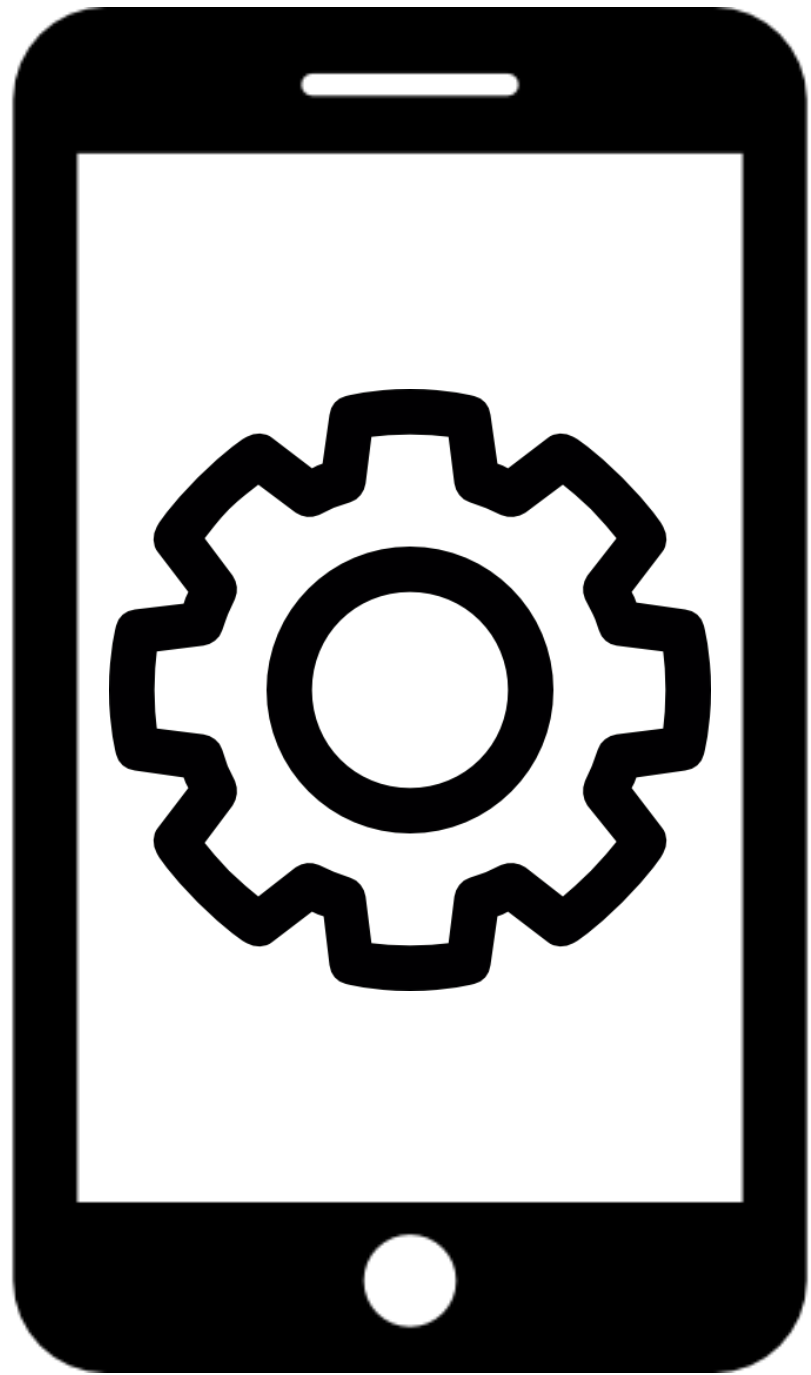
install



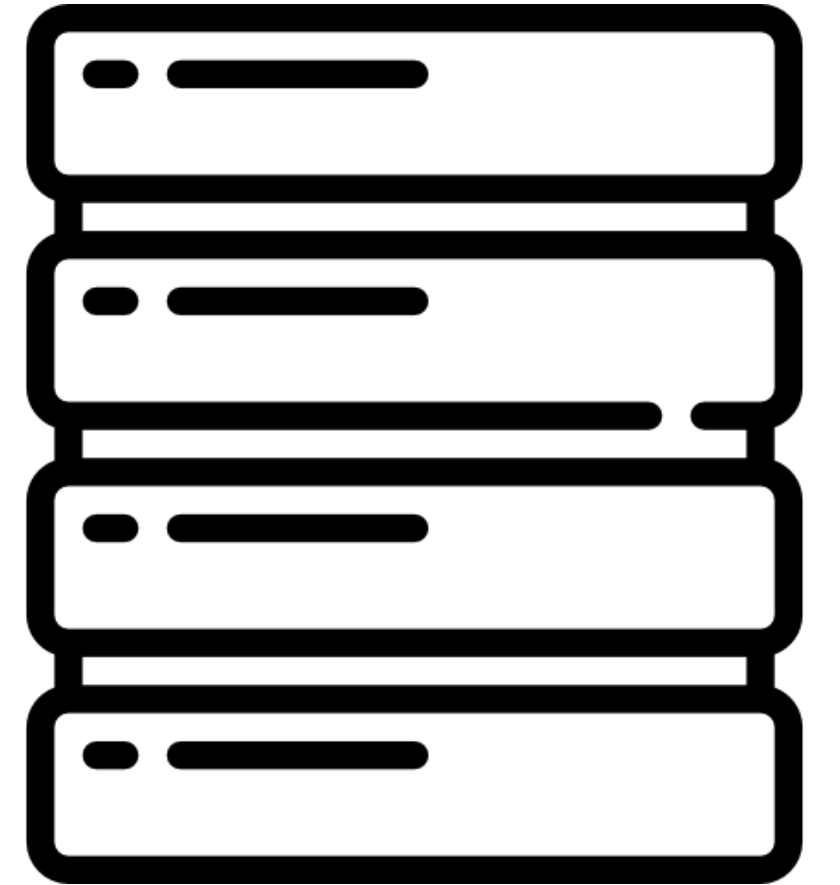
`styles.css`
`script.js`
`icon.png`



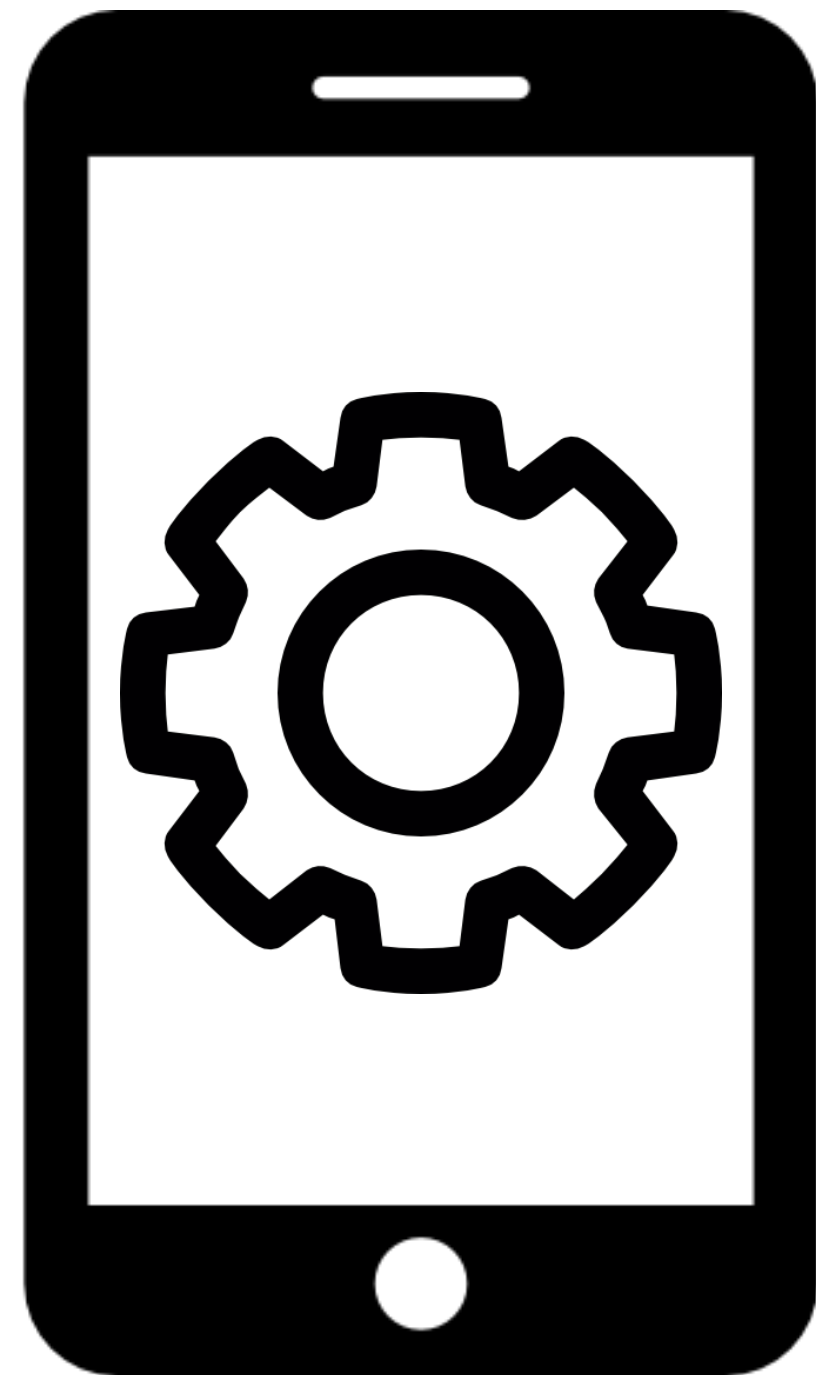
install



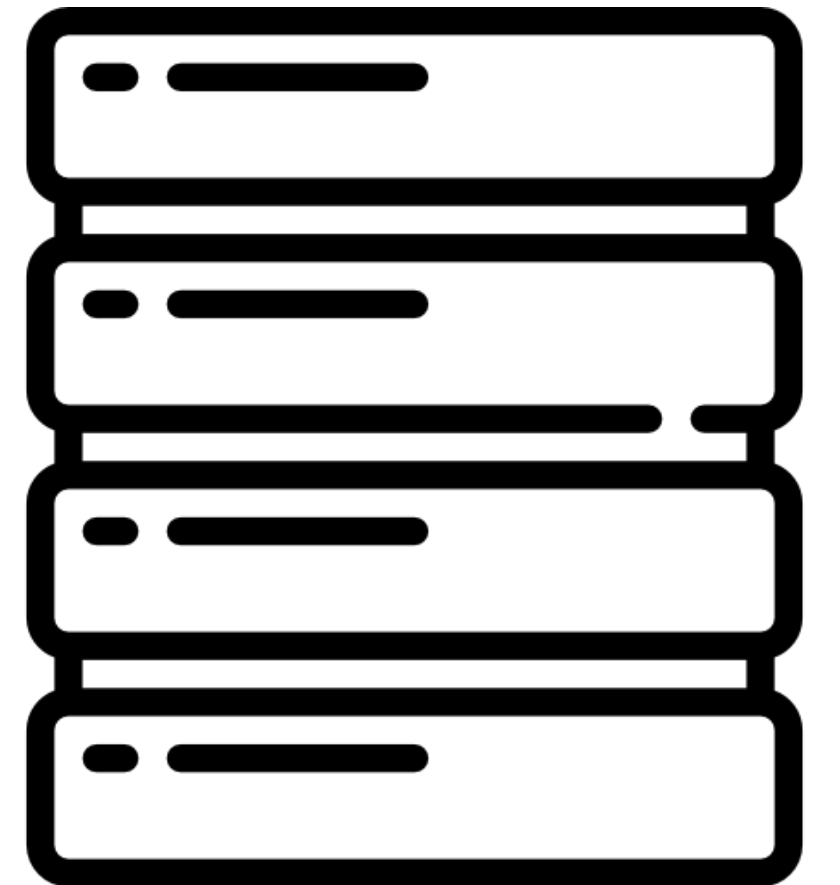
`styles.css`
`script.js`
`icon.png`



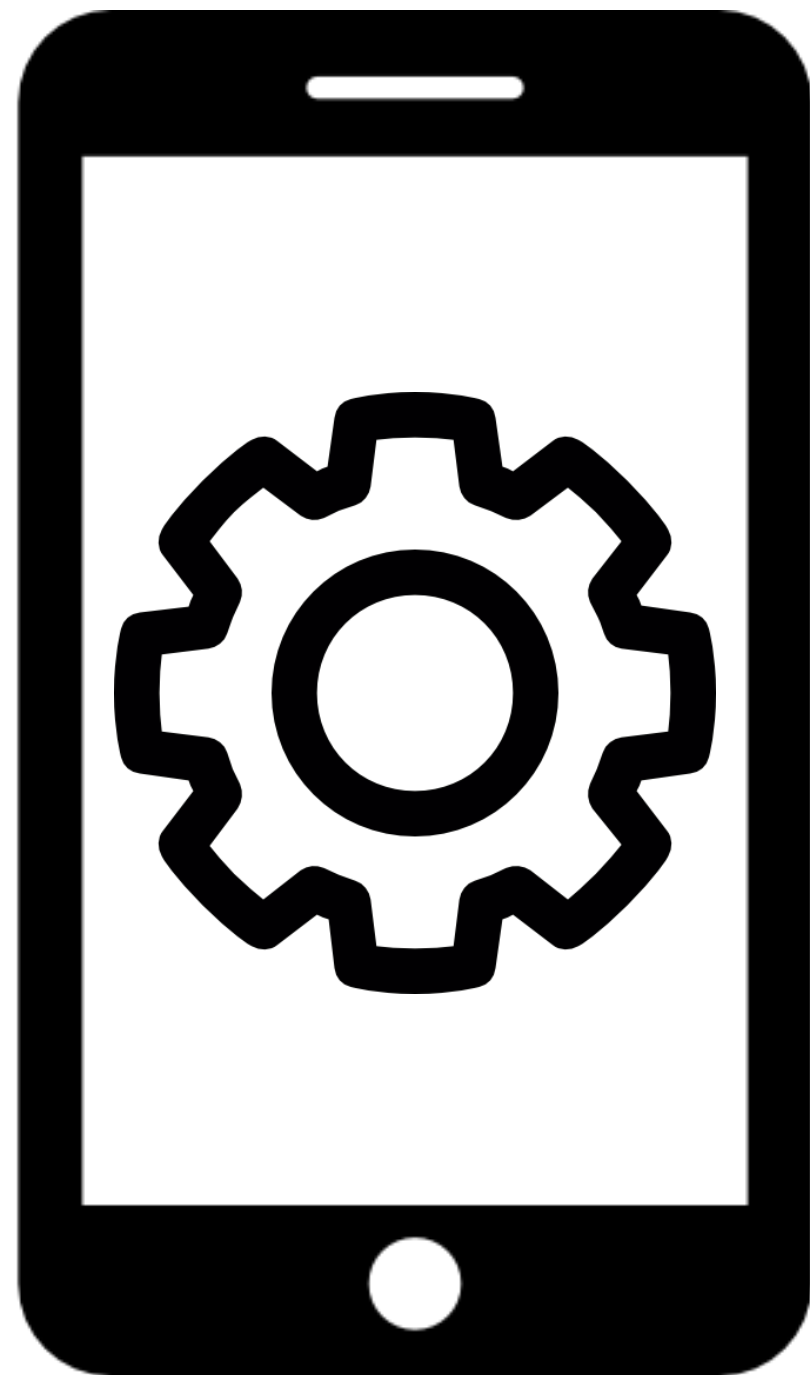
install



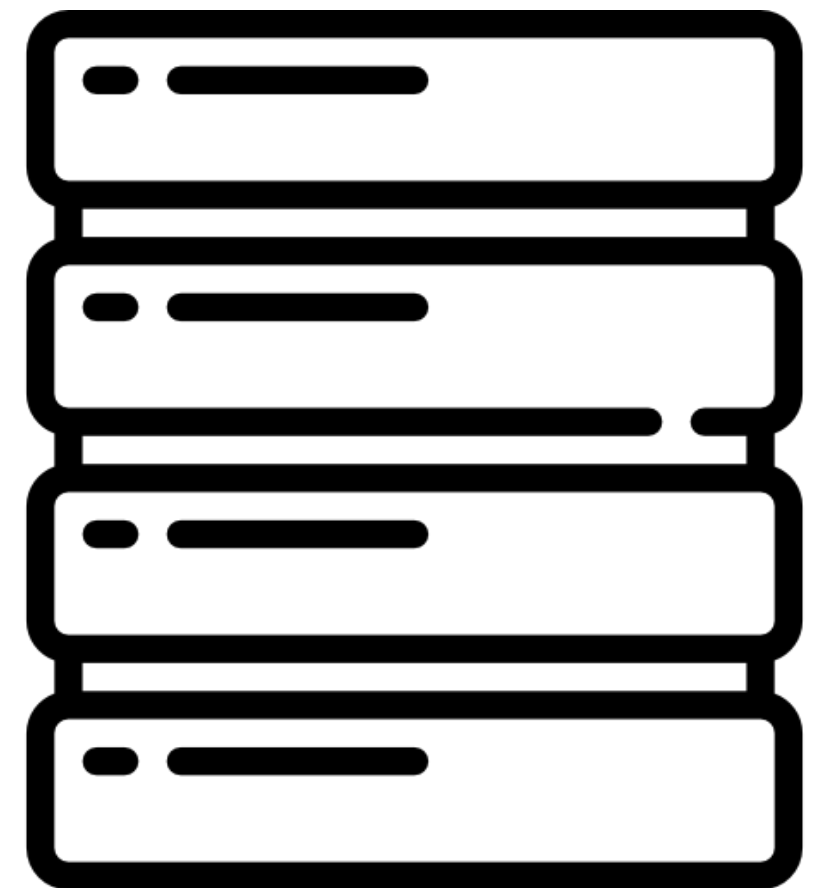
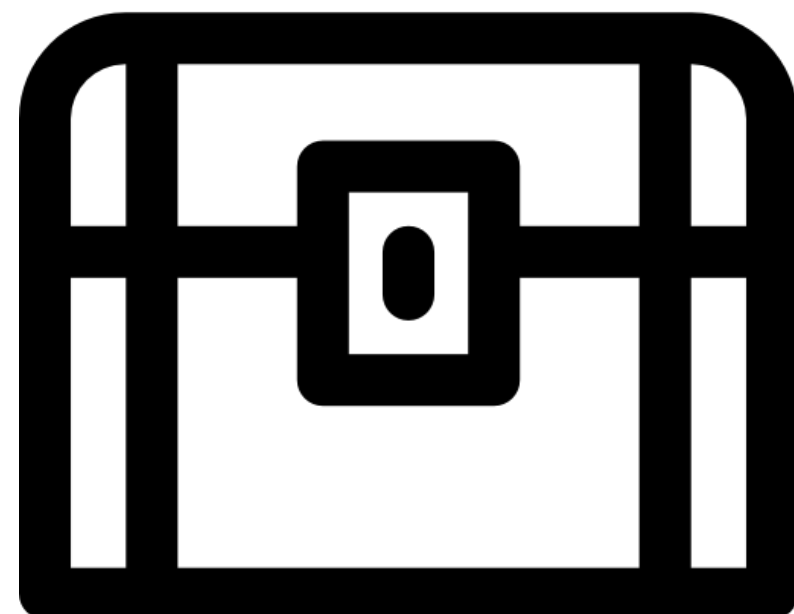
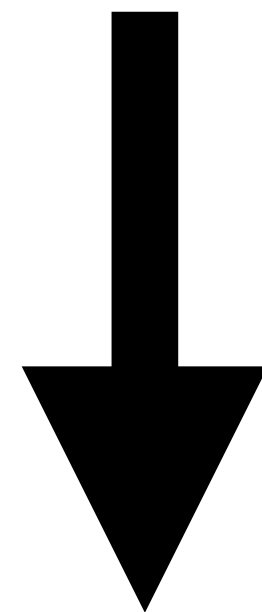
styles.css
script.js
icon.png



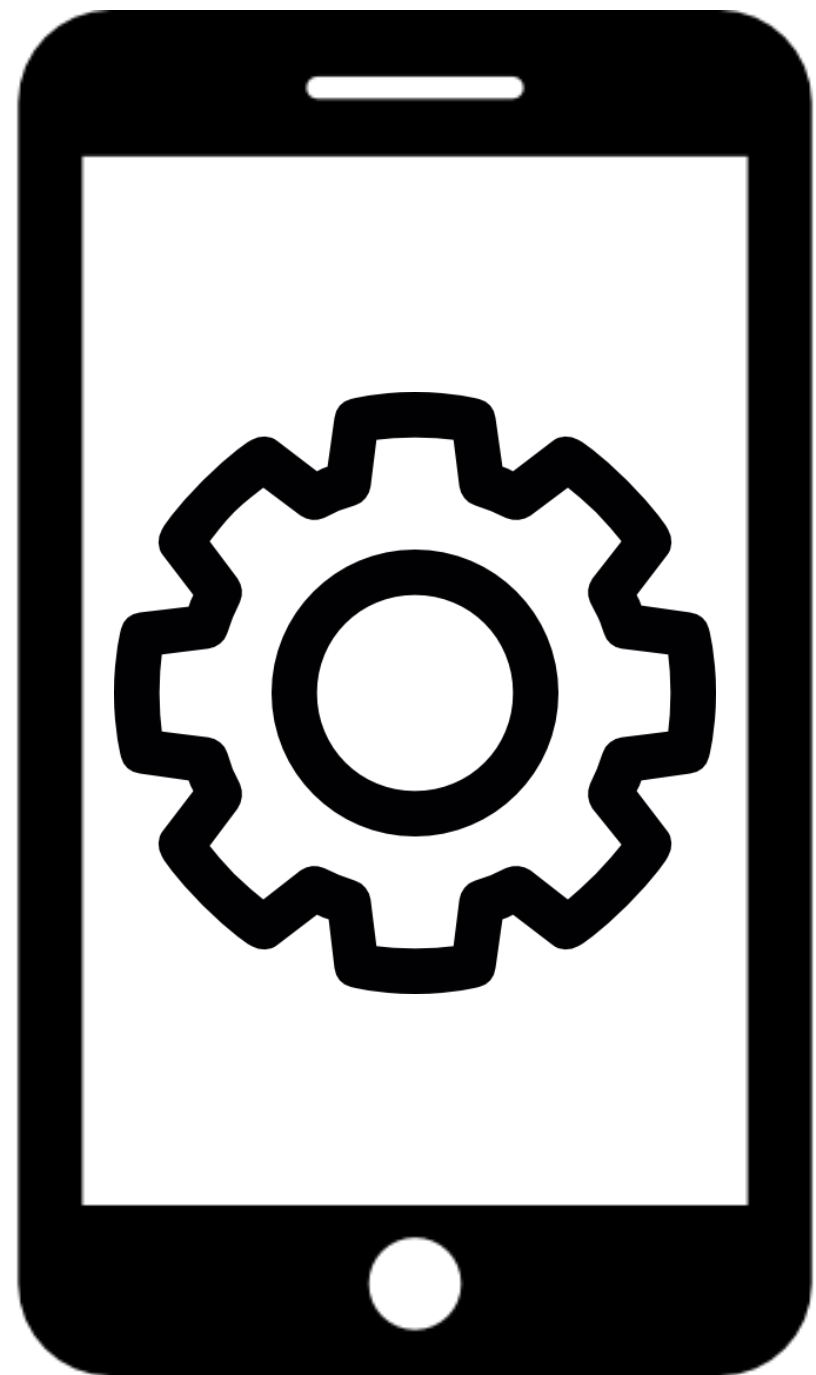
install



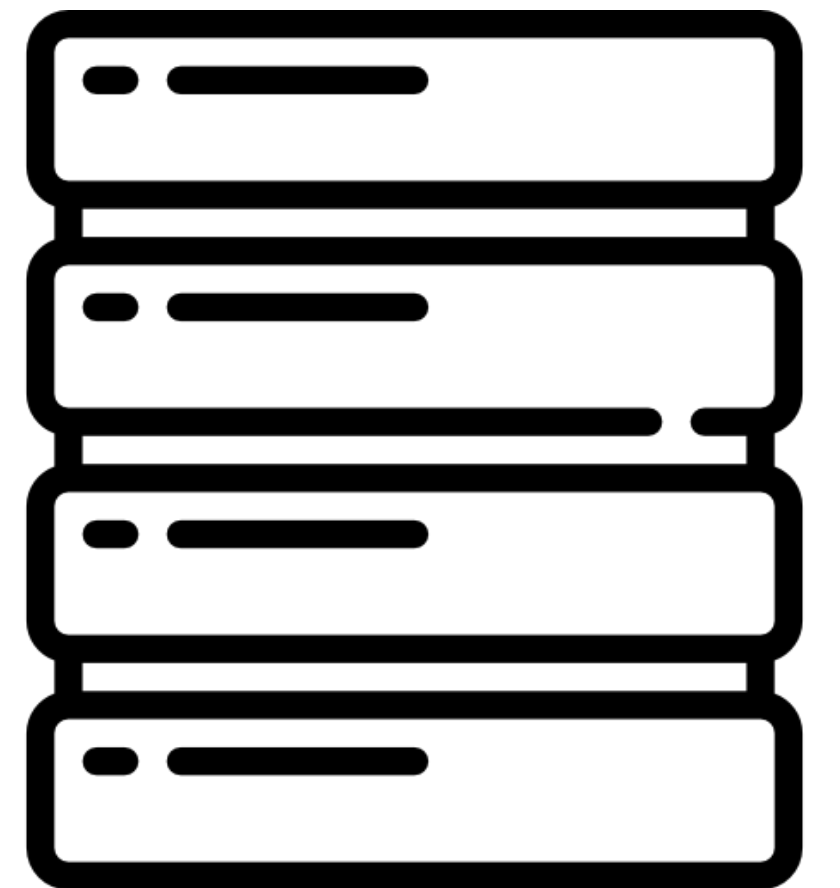
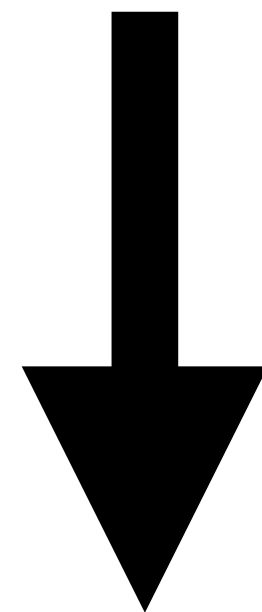
styles.css
script.js
icon.png



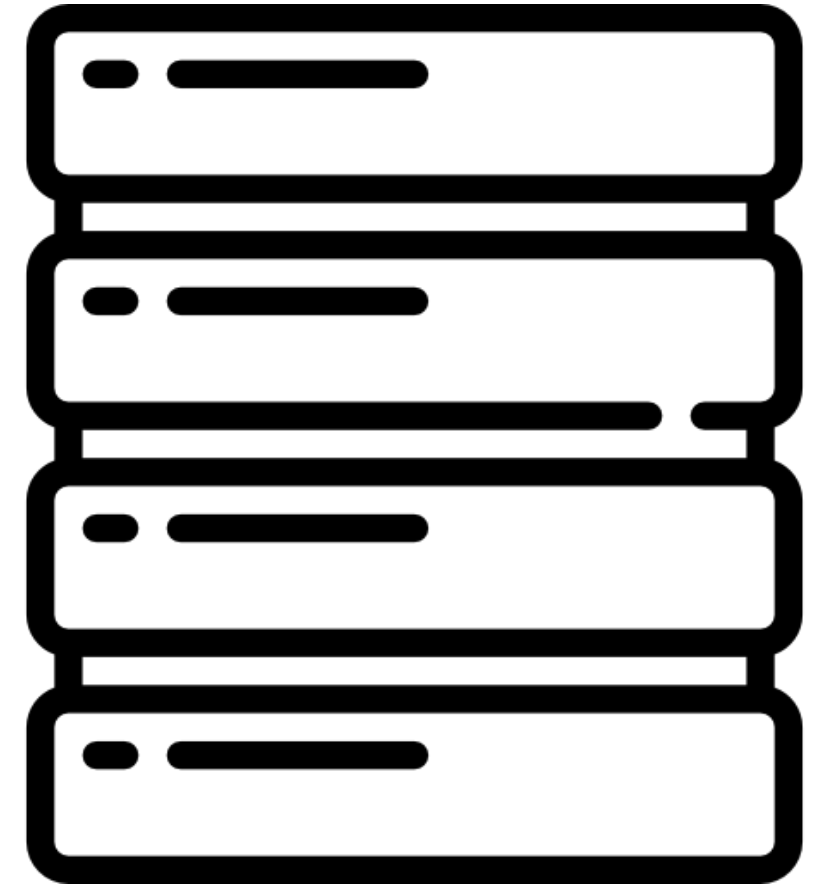
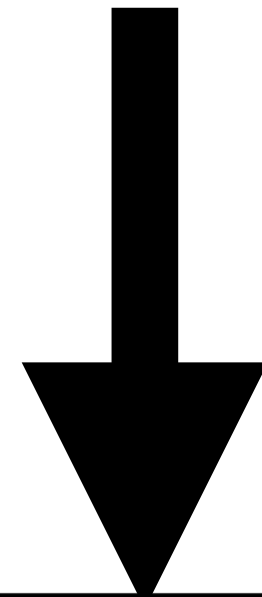
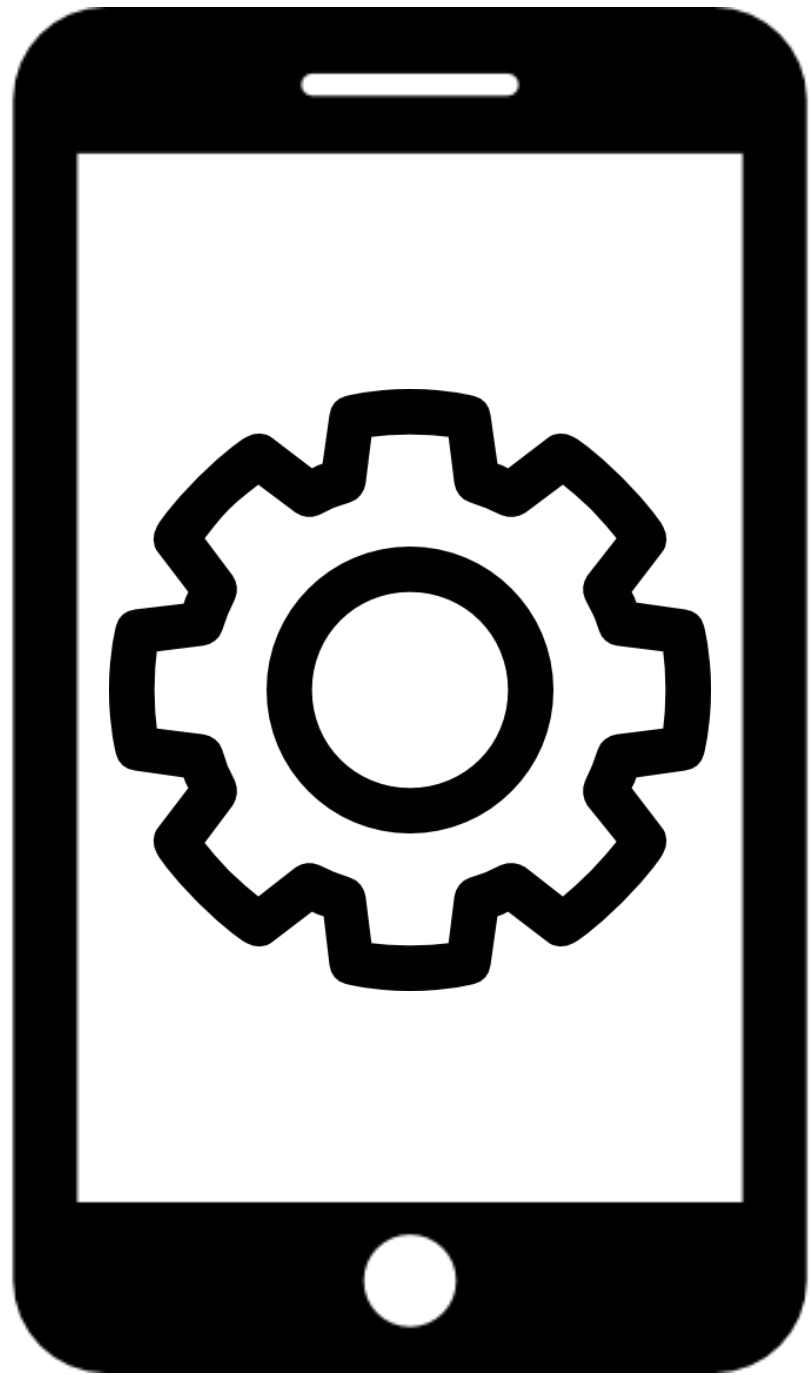
install



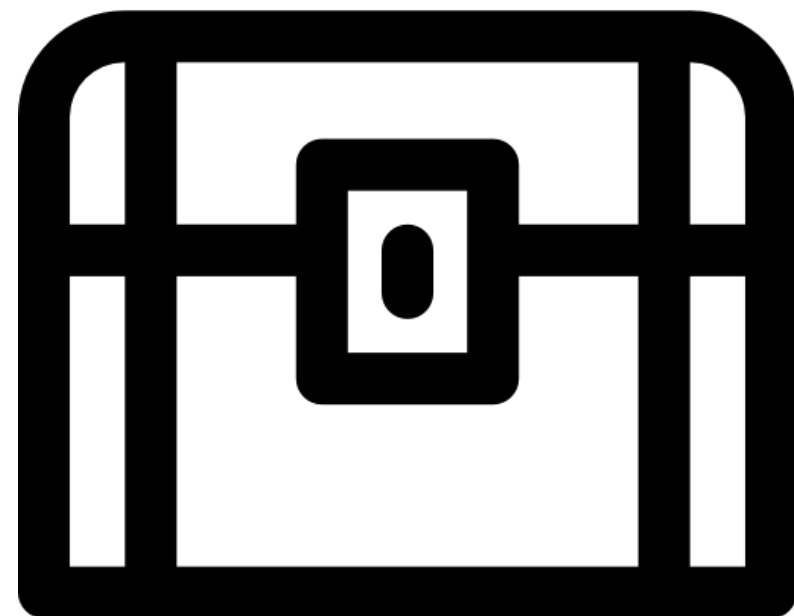
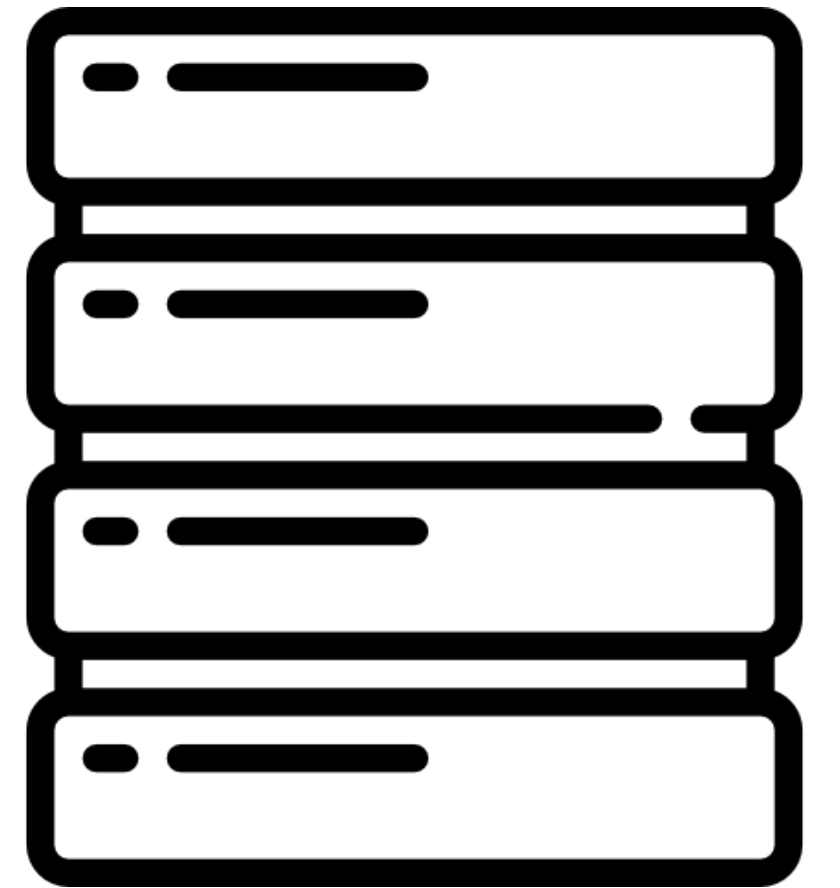
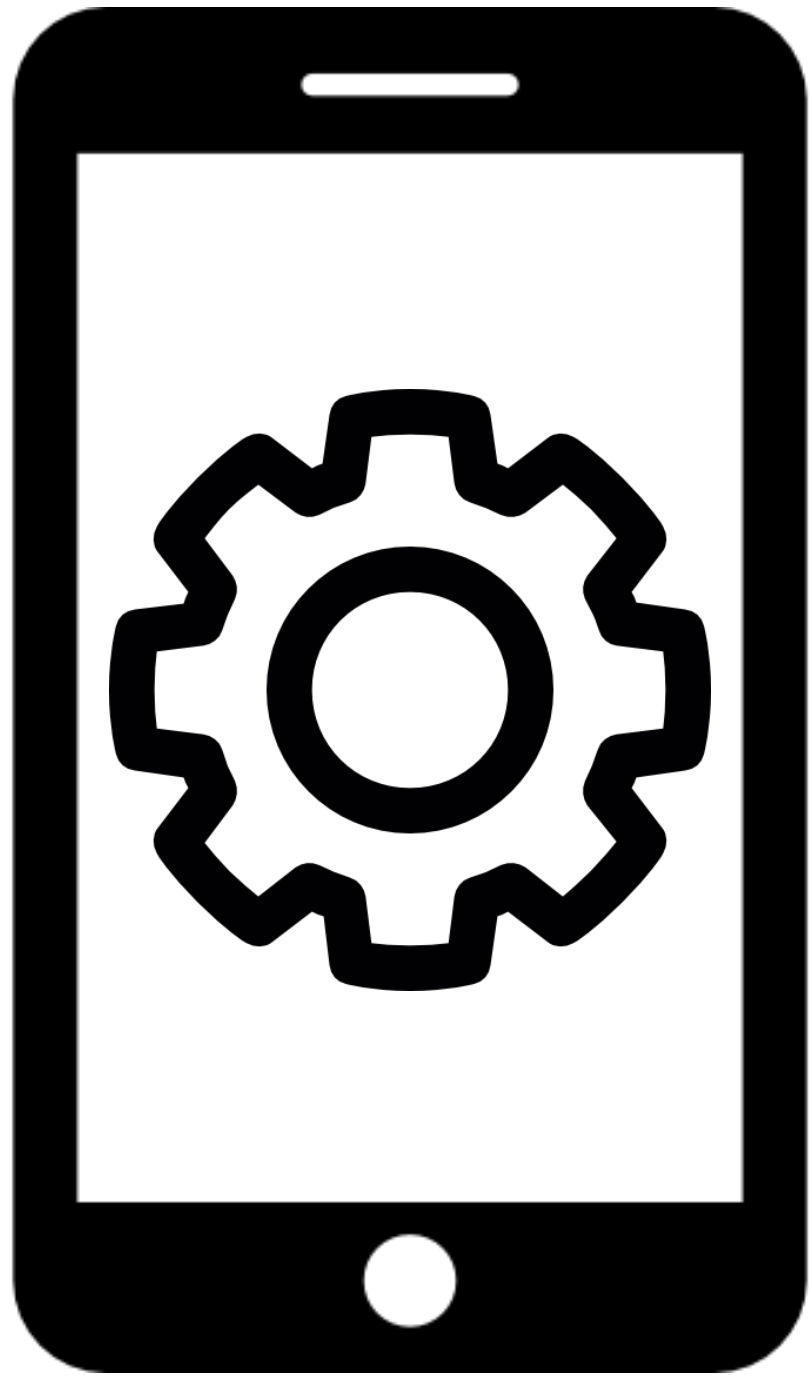
styles.css
script.js
icon.png



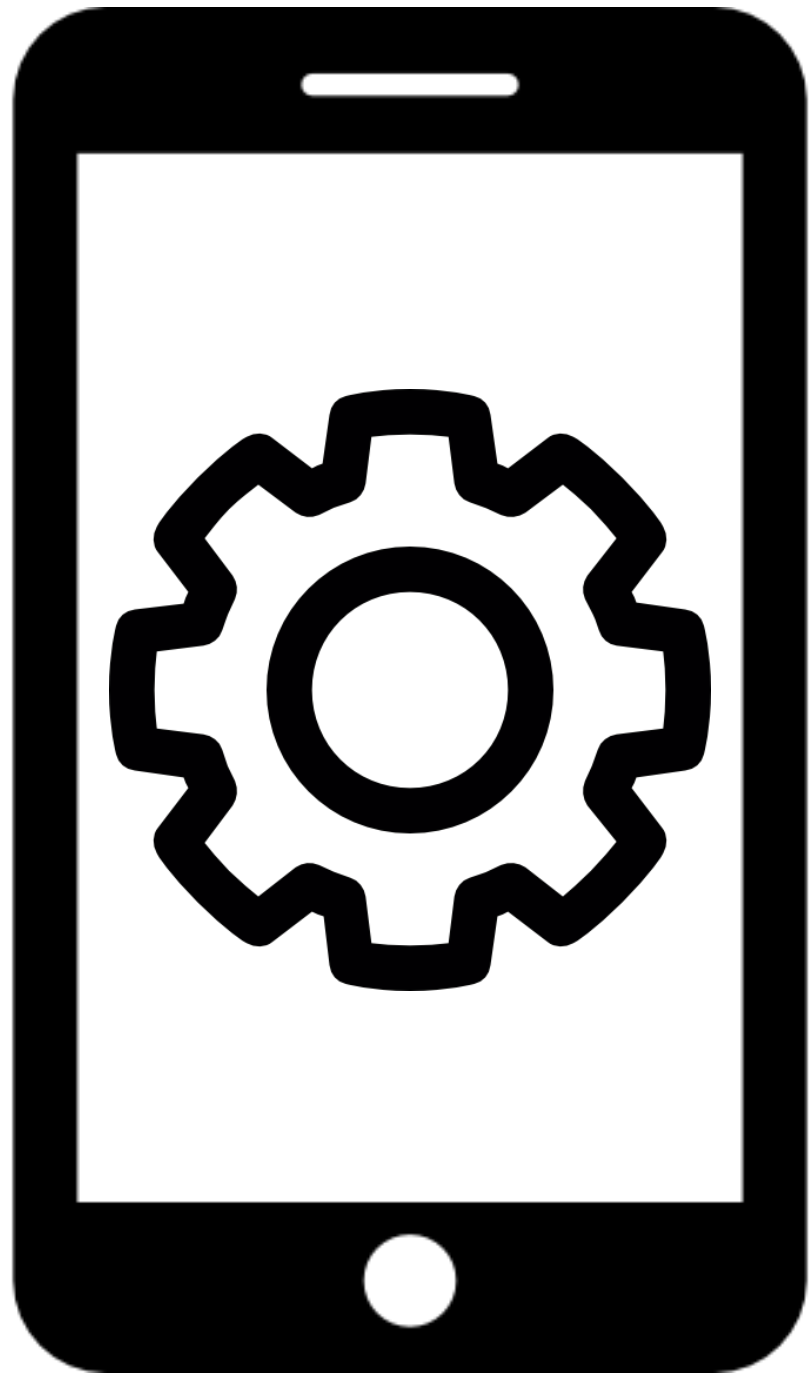
install



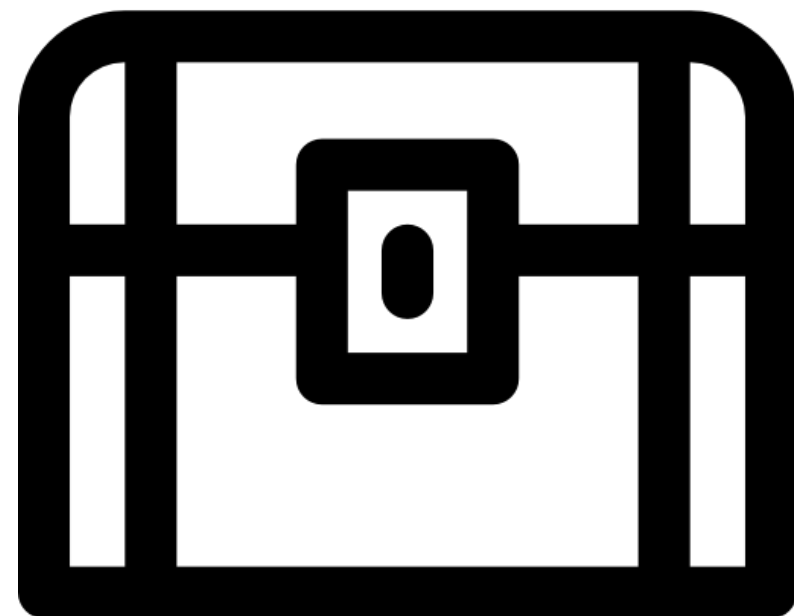
install



install



precaching



precaching

*When this service worker is installing
open a cache and then
put a bunch of files in that cache.*

precaching

When this service worker is installing

precaching

When this service worker is installing

addEventListener()

precaching

When this service worker is installing

```
addEventListener('install')
```

precaching

When this service worker is installing

```
addEventListener('install', function(event) {  
  
});
```

precaching

When this service worker is installing

```
addEventListener('install', event => {  
  
});
```

precaching

open a cache

precaching

open a cache

cache.open()

precaching

open a cache

```
cache.open('files')
```

precaching

open a cache and then

`caches.open('files')`

precaching

open a cache and then

```
cache.open('files').then()
```

precaching

open a cache and then

```
 caches.open('files').then( cache => {  
  
});
```

precaching

put a bunch of files in that cache.

precaching

put a bunch of files in that cache.

cache.addAll()

precaching

put a bunch of files in that cache.

```
cache.addAll([  
  '/css/styles.css',  
  '/js/script.js',  
  '/img/icon.png'  
]);
```

preaching

precaching

When this service worker is installing

```
addEventListener('install', event => {  
  
});
```

precaching

*When this service worker is installing
open a cache*

```
cache.open('files')
```

precaching

*When this service worker is installing
open a cache and then*

```
 caches.open('files').then( cache => {  
  
});
```

precaching

*When this service worker is installing
open a cache and then
put a bunch of files in that cache.*

```
cache.addAll([  
  '/css/styles.css',  
  '/js/script.js',  
  '/img/icon.png'  
]);
```

precaching

*When this service worker is installing
open a cache and then
put a bunch of files in that cache.*

precaching

```
addEventListener('install', event => {
```

```
});
```

precaching

```
addEventListener('install', event => {  
  caches.open('files')
```

```
});
```

precaching

```
addEventListener('install', event => {  
  caches.open('files').then( cache => {
```

```
  });  
});
```

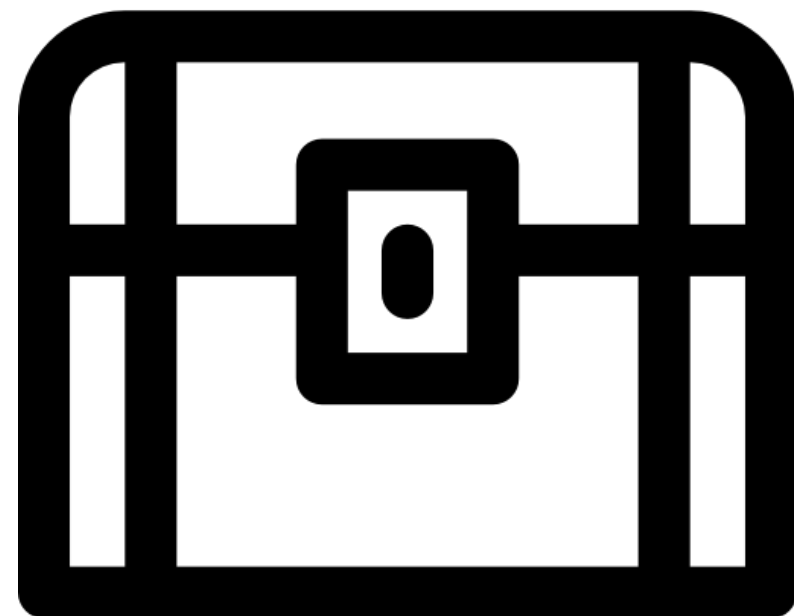
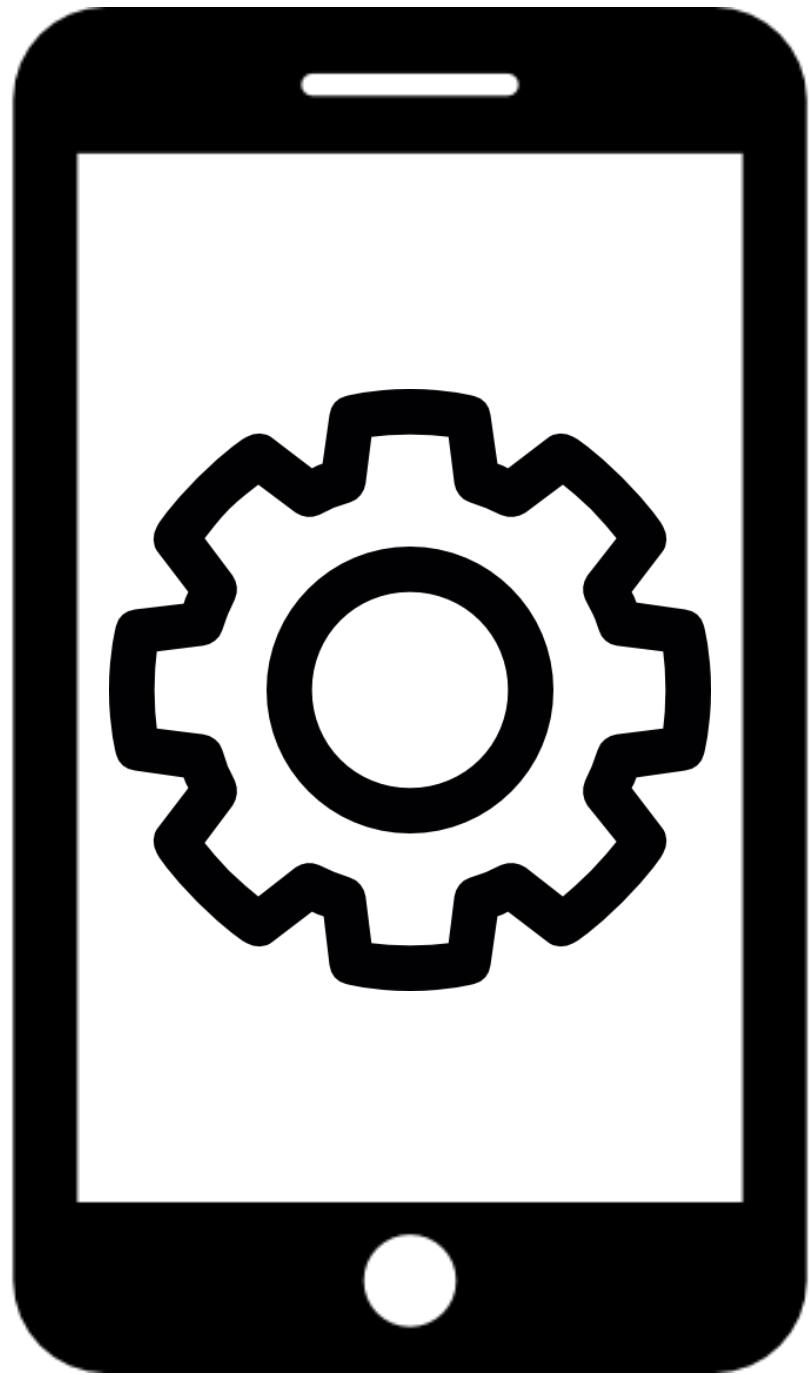

precaching

```
addEventListener('install', event => {  
  caches.open('files').then( cache => {  
    cache.addAll([  
      '/css/styles.css',  
      '/js/script.js',  
      '/img/icon.png'  
    ]);  
  });  
});
```

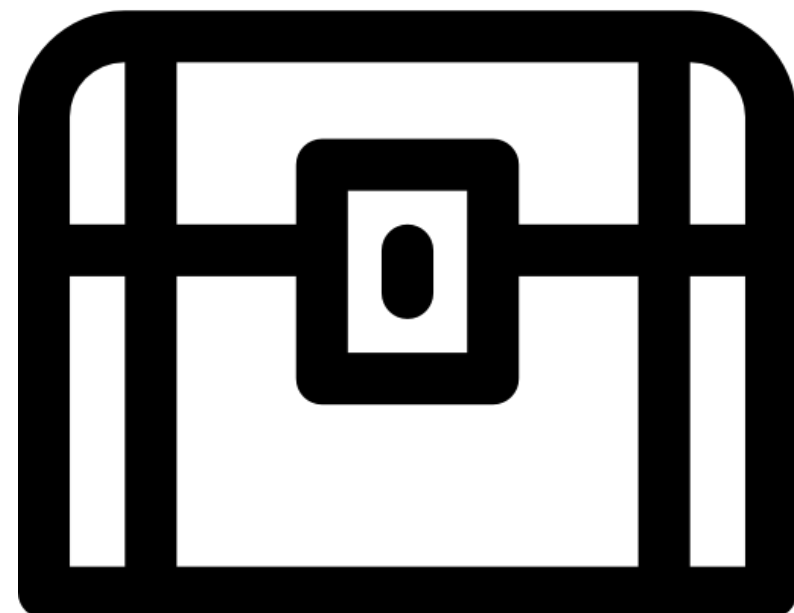
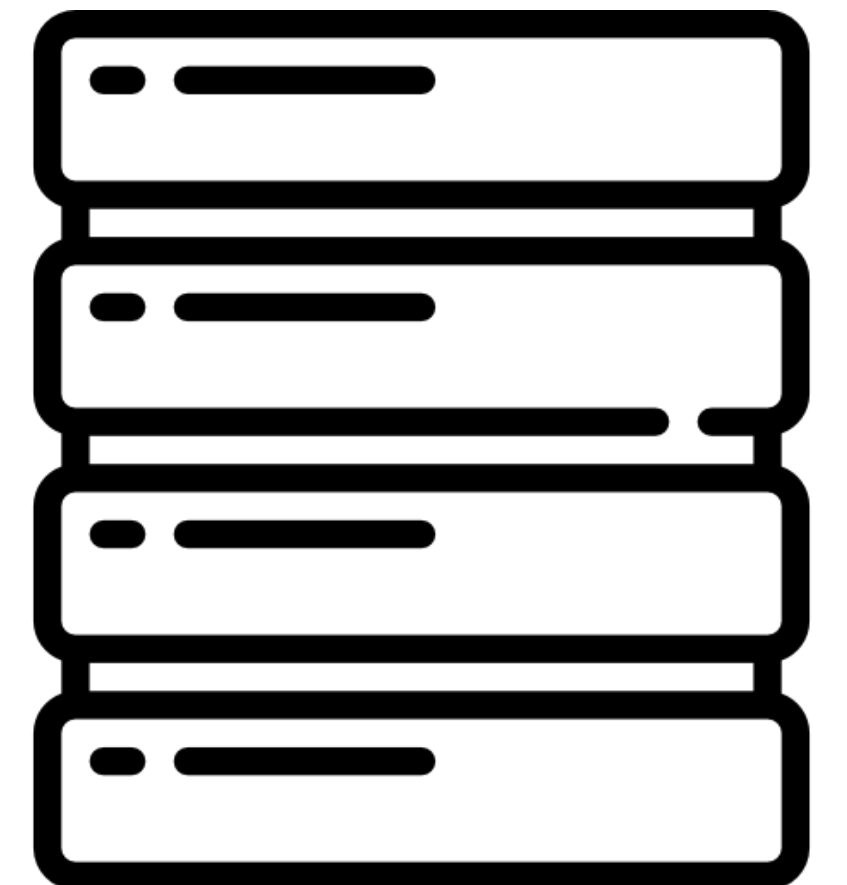
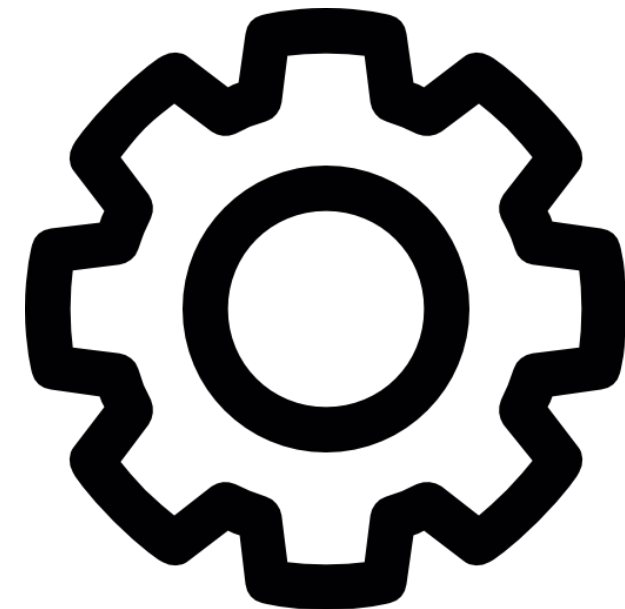
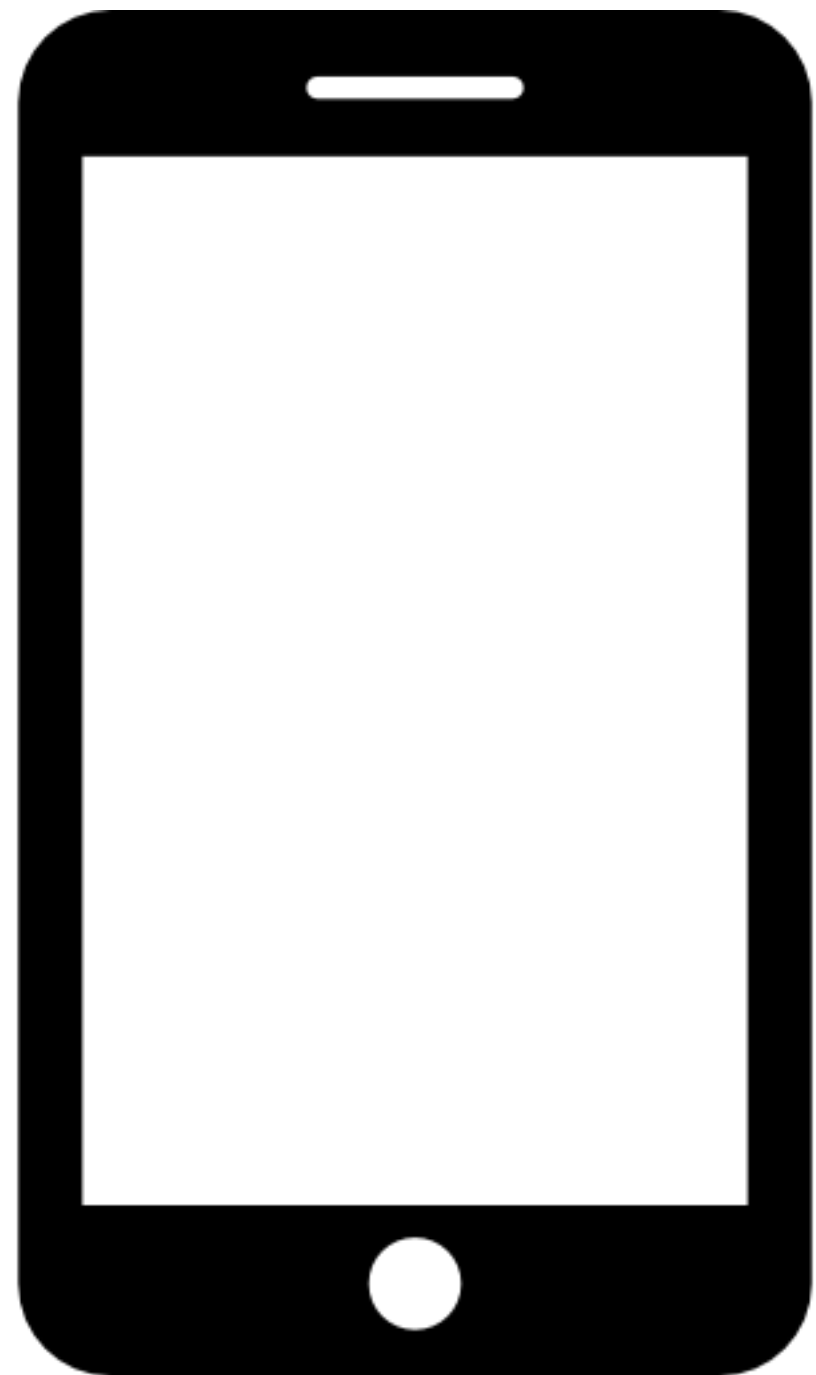
precaching

```
addEventListener('install', event => {  
  caches.open('files').then( cache => {  
    cache.addAll([  
      '/css/styles.css',  
      '/js/script.js',  
      '/img/icon.png'  
    ]);  
  });  
});
```

install



active



cache first

*When a file is requested
try to find the file in the cache first
otherwise fetch the file from the network.*

cache first

When a file is requested

cache first

When a file is requested

addEventListener()

cache first

When a file is requested

addEventListener('fetch')

cache first

When a file is requested

```
addEventListener('fetch', event => {  
  
});
```

cache first

When a file is requested

```
addEventListener('fetch', event => {  
  event.respondWith(  
  
  );  
});
```

cache first

try to find the file in the cache first

cache first

try to find the file in the cache first

```
caches.match(event.request)
```

cache first

try to find the file in the cache first

```
cache.match(event.request)
  .then( response => {

})
```

cache first

try to find the file in the cache first

```
cache.match(event.request)
  .then( response => {
    return response;
  })
```

cache first

*try to find the file in the cache first
otherwise fetch the file from the network.*

```
cache.match(event.request)
  .then( response => {
    return response;
  })
```

cache first

*try to find the file in the cache first
otherwise fetch the file from the network.*

```
cache.match(event.request)
  .then( response => {
    return response || fetch(event.request);
  })
```


cache first

*When a file is requested
try to find the file in the cache first
otherwise fetch the file from the network.*

cache first

```
addEventListener('fetch', event => {  
  event.respondWith(  

```

```
);  
});
```

cache first

```
addEventListener('fetch', event => {  
  event.respondWith(  
    caches.match(event.request)  
      .then( response => {  
        return response  
      })  
  );  
});
```

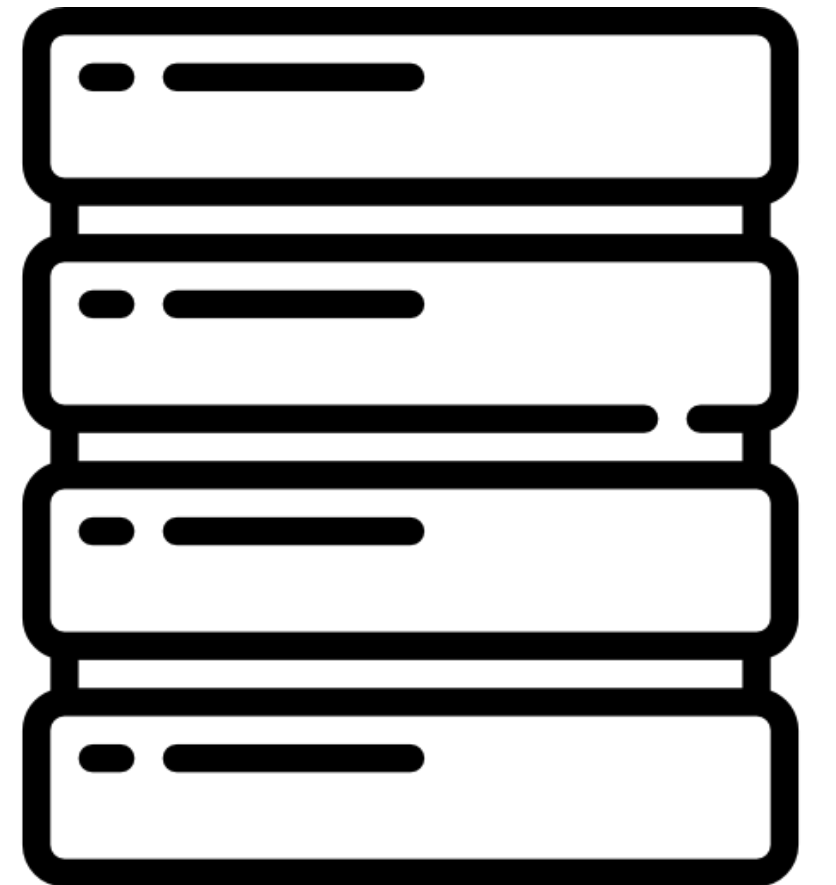
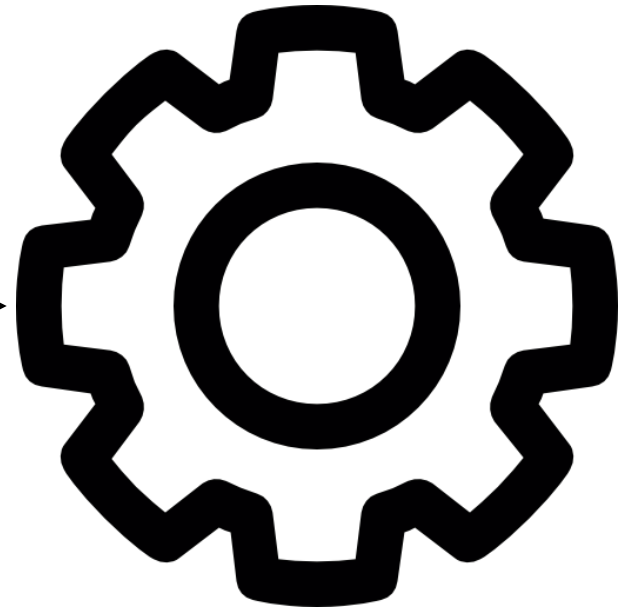
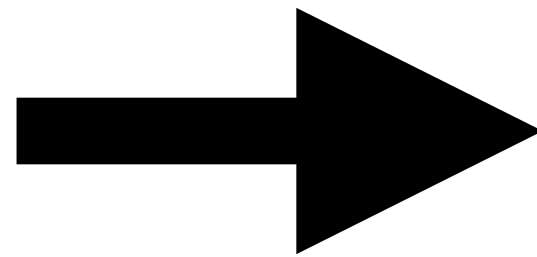
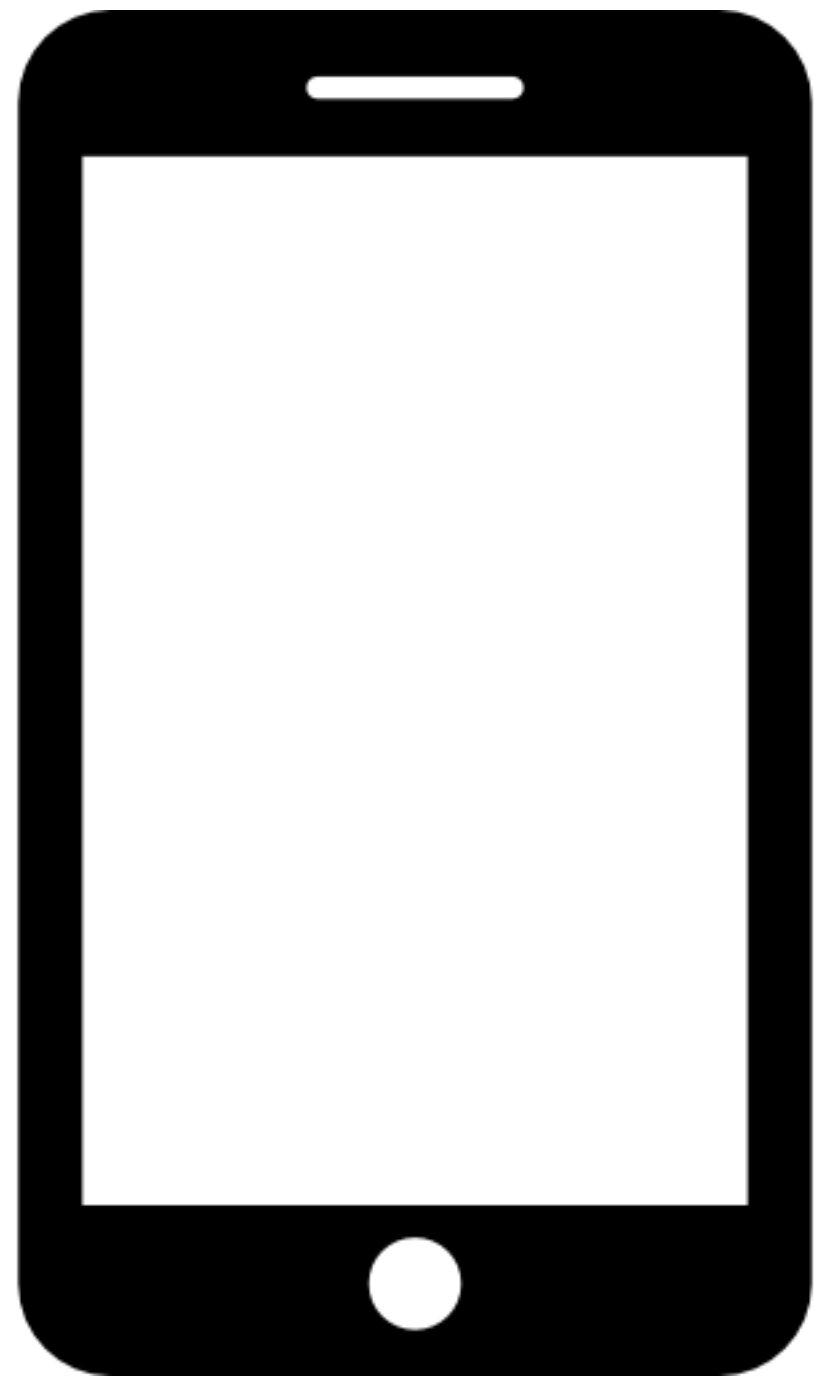
cache first

```
addEventListener('fetch', event => {  
  event.respondWith(  
    caches.match(event.request)  
      .then( response => {  
        return response || fetch(event.request);  
      })  
  );  
});
```

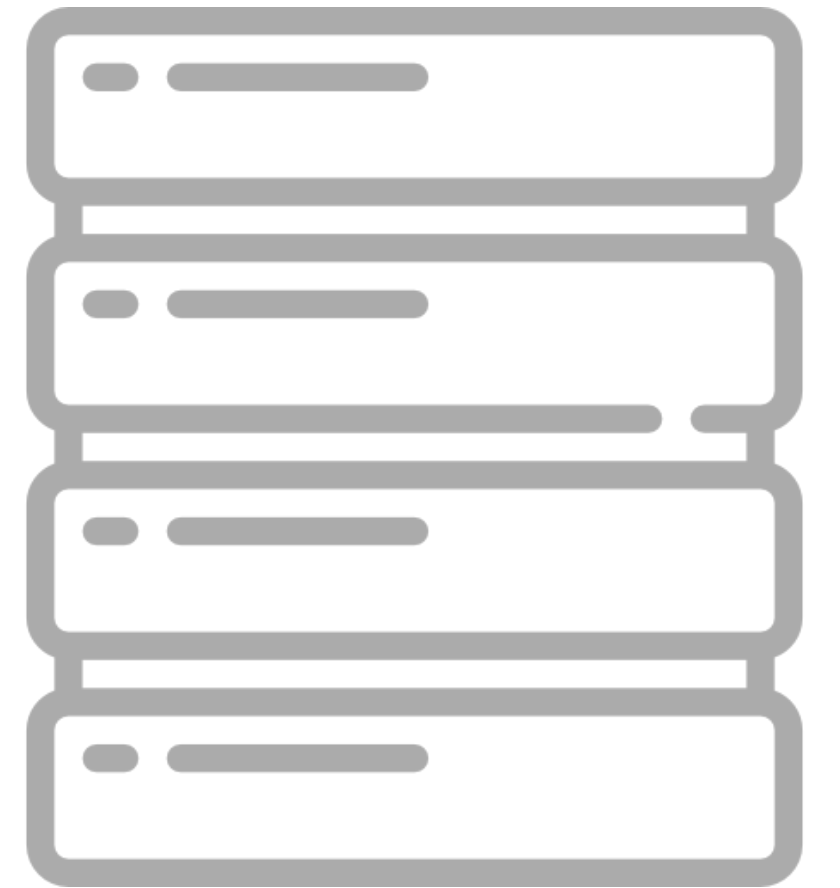
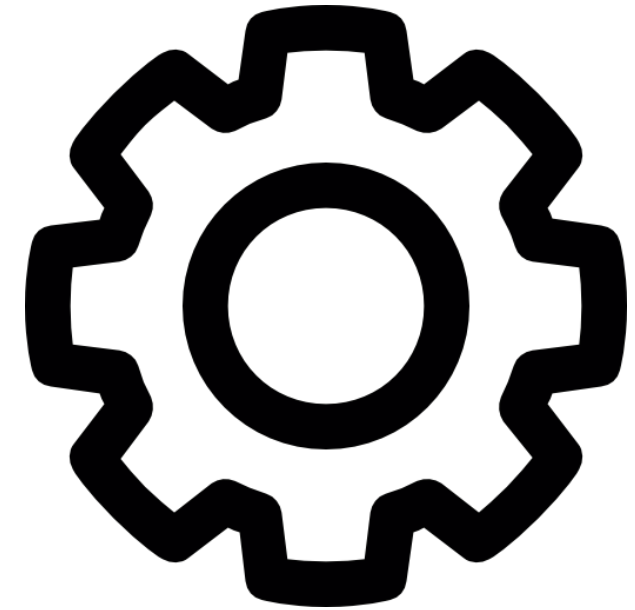
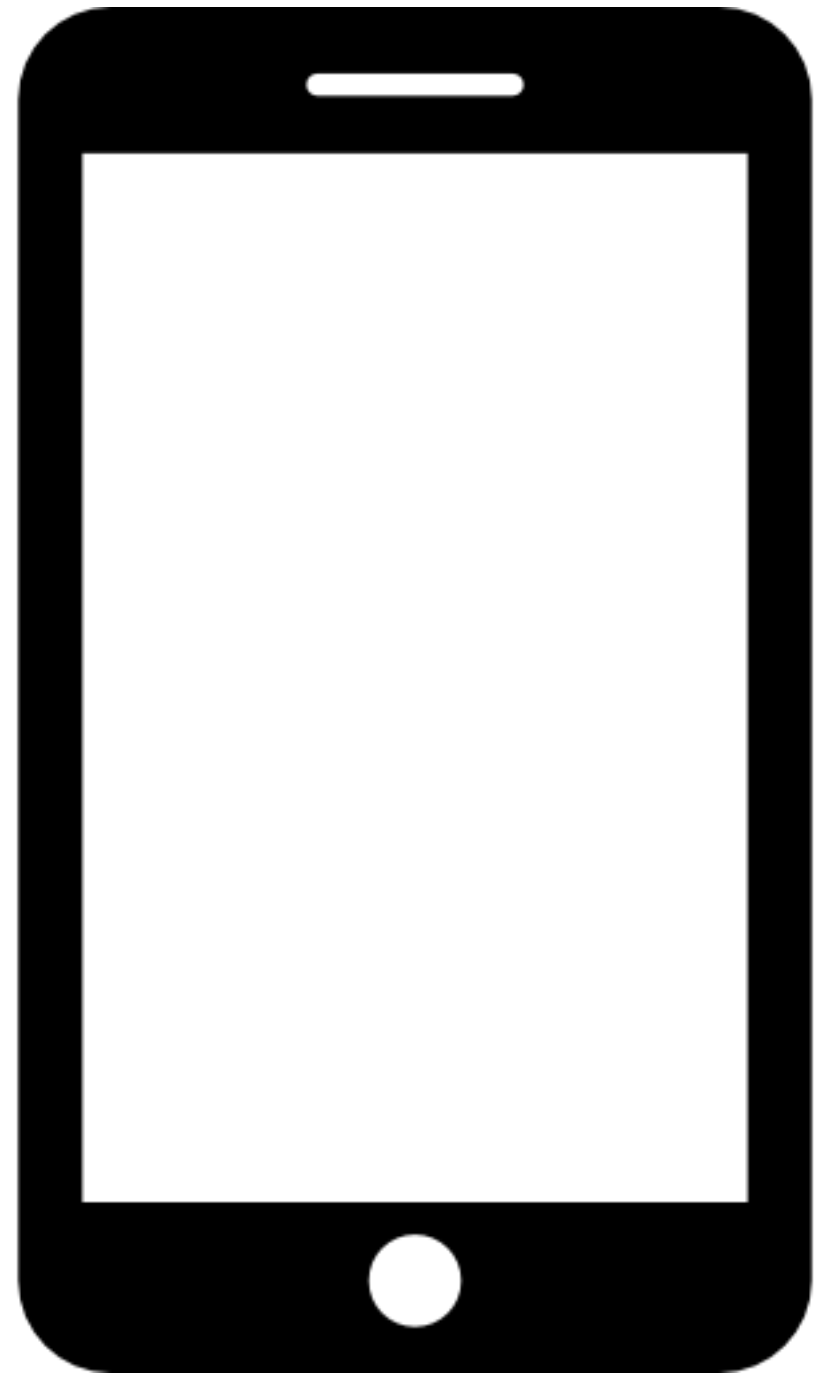
cache first

```
addEventListener('fetch', event => {  
  event.respondWith(  
    caches.match(event.request)  
      .then( response => {  
        return response || fetch(event.request);  
      })  
  );  
});
```

cache first



offline first



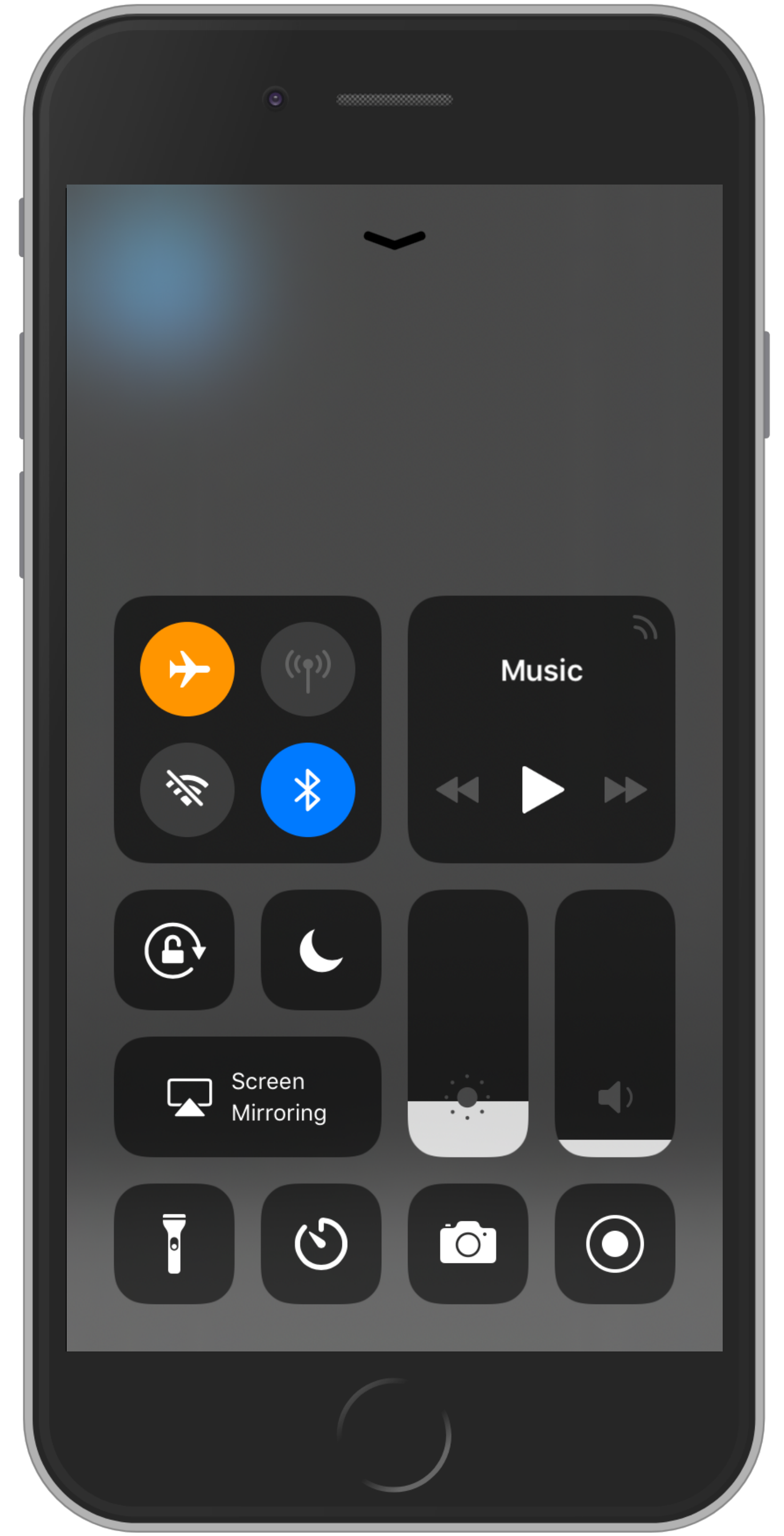
offline first



offline first



offline first



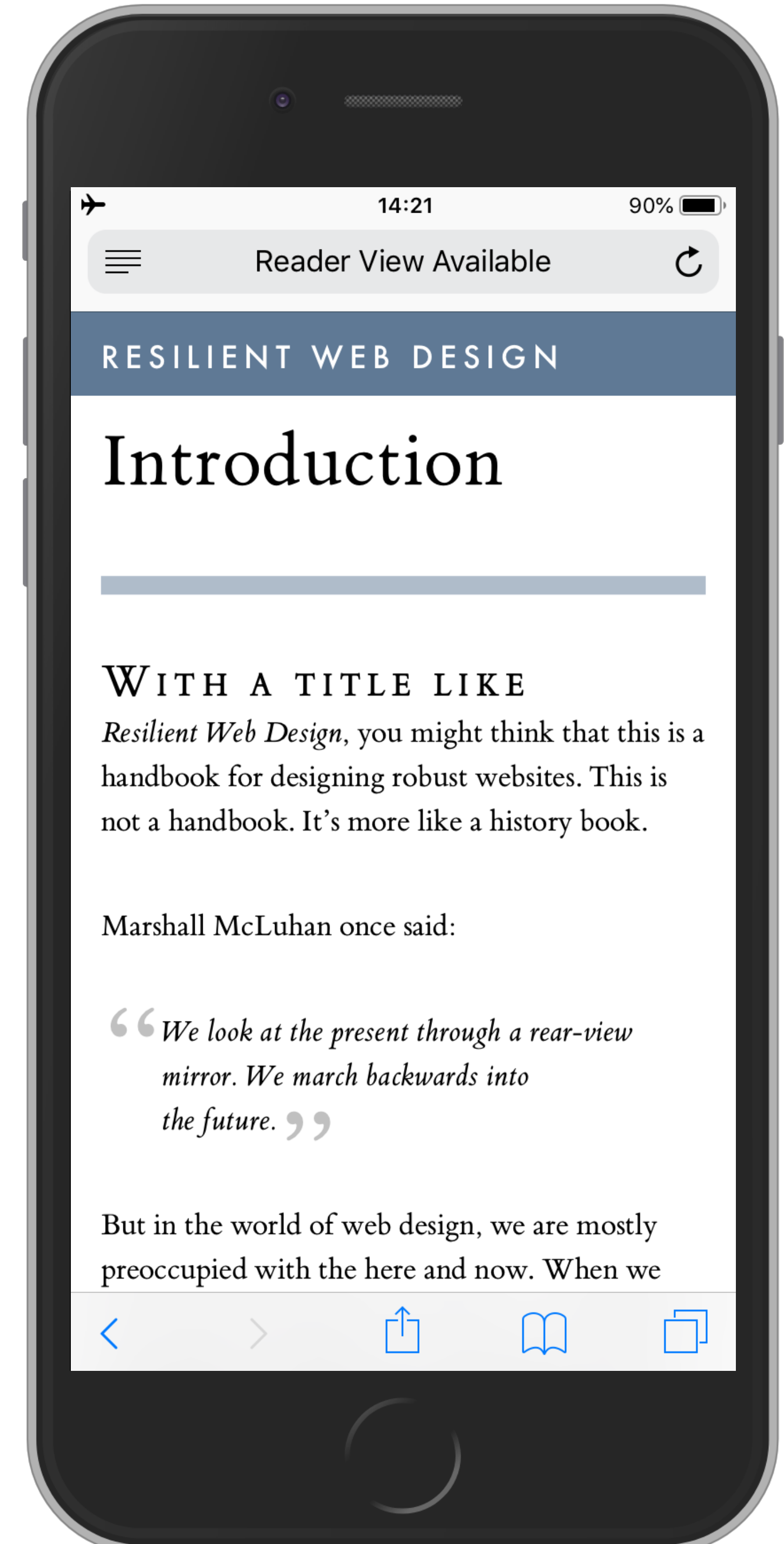
offline first



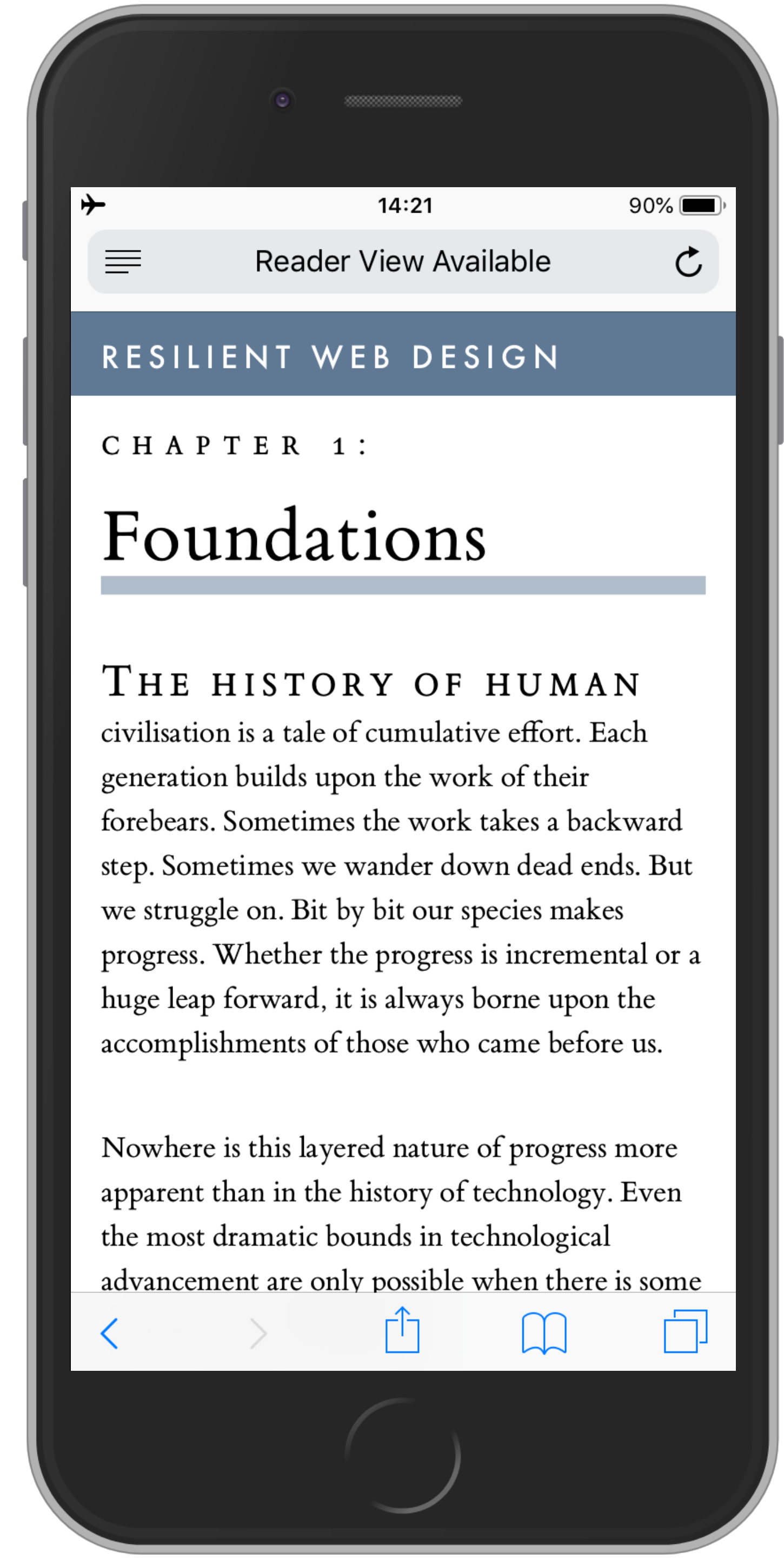
offline first



offline first



offline first



offline first

precaching

cache first

install

active

css js img

precaching

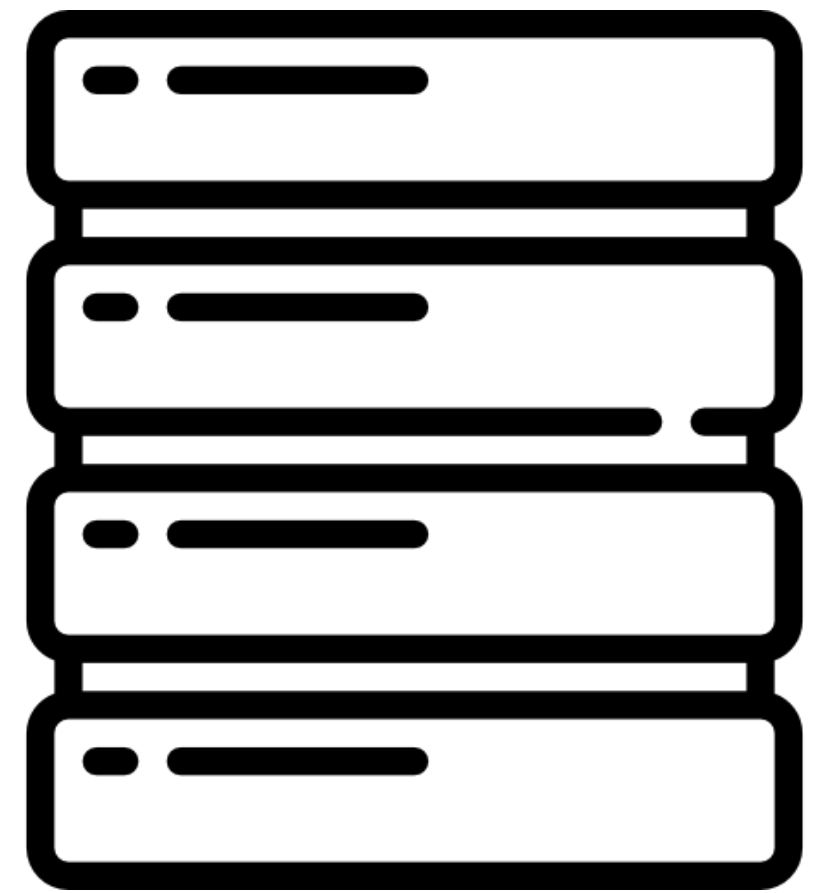
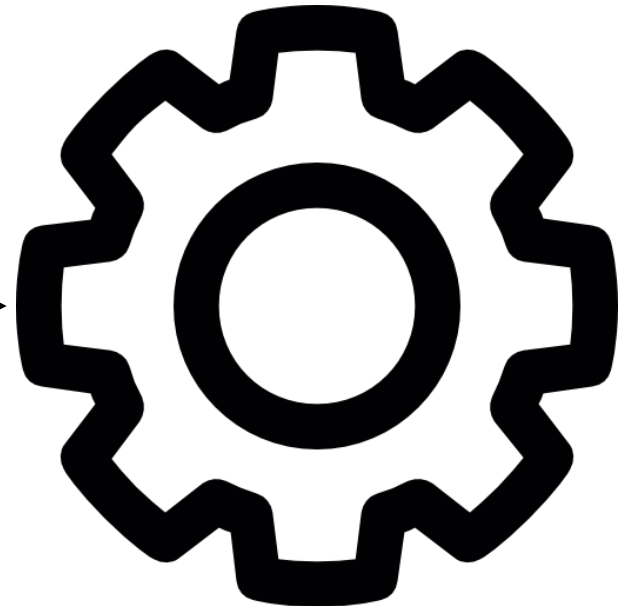
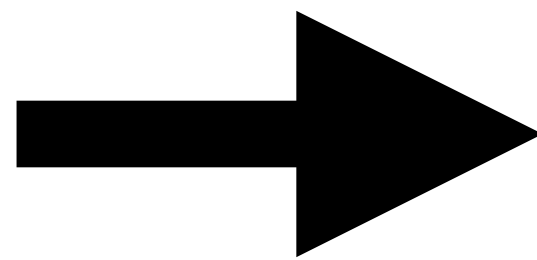
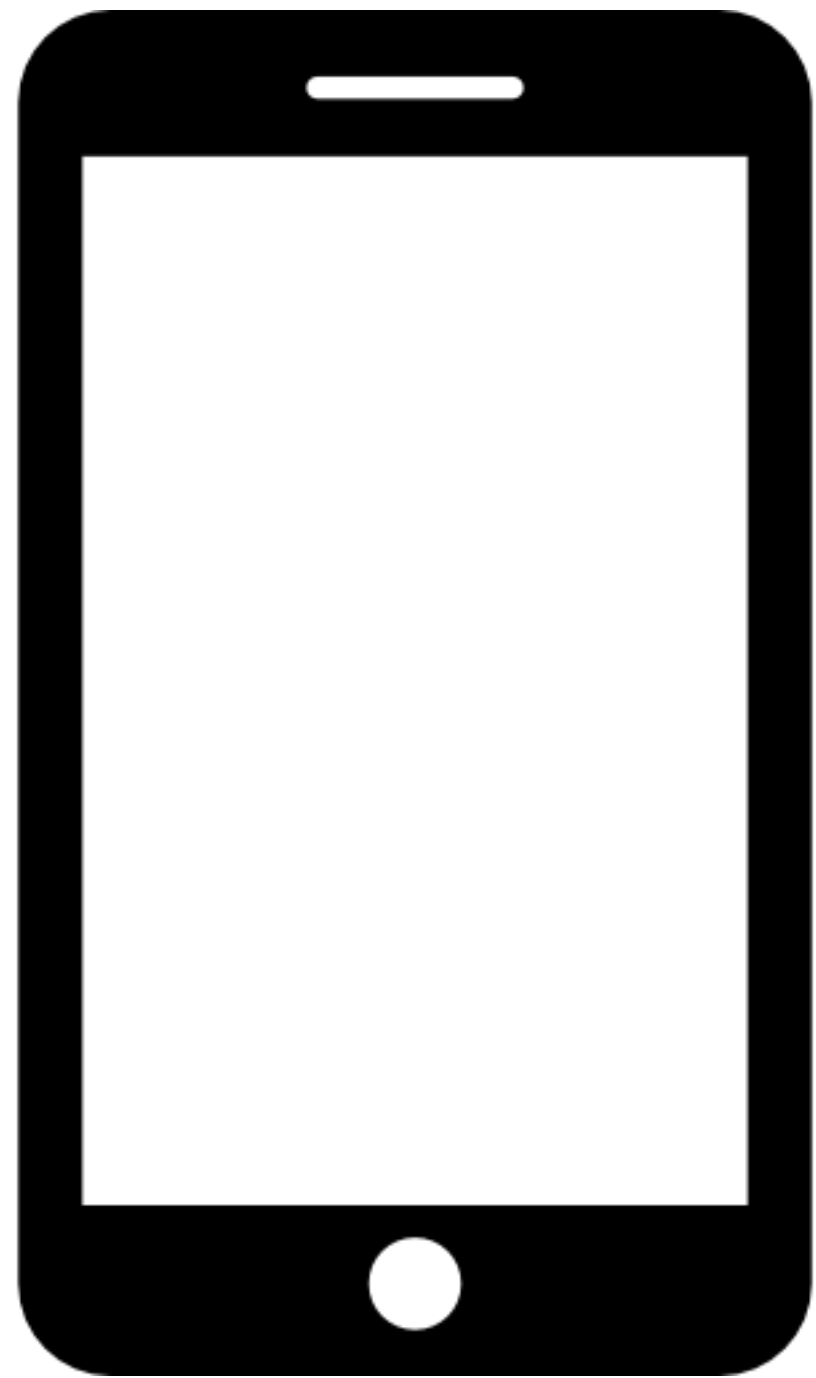
cache first

html

network first

custom offline page

network first



precaching

```
addEventListener('install', event => {  
  caches.open('files').then( cache => {  
    cache.addAll([  
      '/css/styles.css',  
      '/js/script.js',  
      '/img/icon.png'  
    ]);  
  });  
});
```

precaching

```
addEventListener('install', event => {  
  caches.open('files').then( cache => {  
    cache.addAll([  
      '/css/styles.css',  
      '/js/script.js',  
      '/img/icon.png',  
      '/offline.html'  
    ]);  
  });  
});
```

network first

*When a page is requested
try to fetch the page from the network first
otherwise retrieve the offline page from the cache.*

network first

When a page is requested

```
addEventListener('fetch', event => {  
  
});
```

network first

When a page is requested

```
addEventListener('fetch', event => {  
  if (event.request.headers  
    .get('Accept').includes('text/html')) {  
  
  }  
});
```

network first

try to fetch the page from the network first

```
event.respondWith(  
  fetch(event.request)  
    .then( response => {  
      return response;  
    })  
);
```


network first

otherwise retrieve the offline page from the cache.

```
.catch( error => {  
  return caches.match('/offline.html');  
})
```

network first

network first

```
addEventListener('fetch', event => {  
  if (event.request.headers.get('Accept').includes('text/html')) {
```

```
  }  
});
```

network first

```
addEventListener('fetch', event => {  
  if (event.request.headers.get('Accept').includes('text/html')) {  
    event.respondWith(  
      fetch(event.request)  
        .then( response => {  
          return response;  
        })  
    );  
  }  
});
```

network first

```
addEventListener('fetch', event => {  
  if (event.request.headers.get('Accept').includes('text/html')) {  
    event.respondWith(  
      fetch(event.request)  
        .then( response => {  
          return response;  
        })  
        .catch( error => {  
          return caches.match('offline.html');  
        })  
    );  
  }  
});
```

network first

```
addEventListener('fetch', event => {  
  if (event.request.headers.get('Accept').includes('text/html')) {  
    event.respondWith(  
      fetch(event.request)  
        .then( response => {  
          return response;  
        })  
        .catch( error => {  
          return caches.match('offline.html');  
        })  
    );  
  }  
});
```

"AMPERSAND"

29 June 2018, Duke of York's
Picturehouse, Brighton, UK

OFFLINE

Something seems to be up with the internet
connection. Sorry about that.

Here's what you need to know about the
Ampersand conference:

- Ampersand is happening on Friday, June 29th,
2018.
- It will be at the Duke of York's Picturehouse in
Brighton, UK.
- Registration starts at 9am.
- If you have any questions, you can write to [Alis](#)

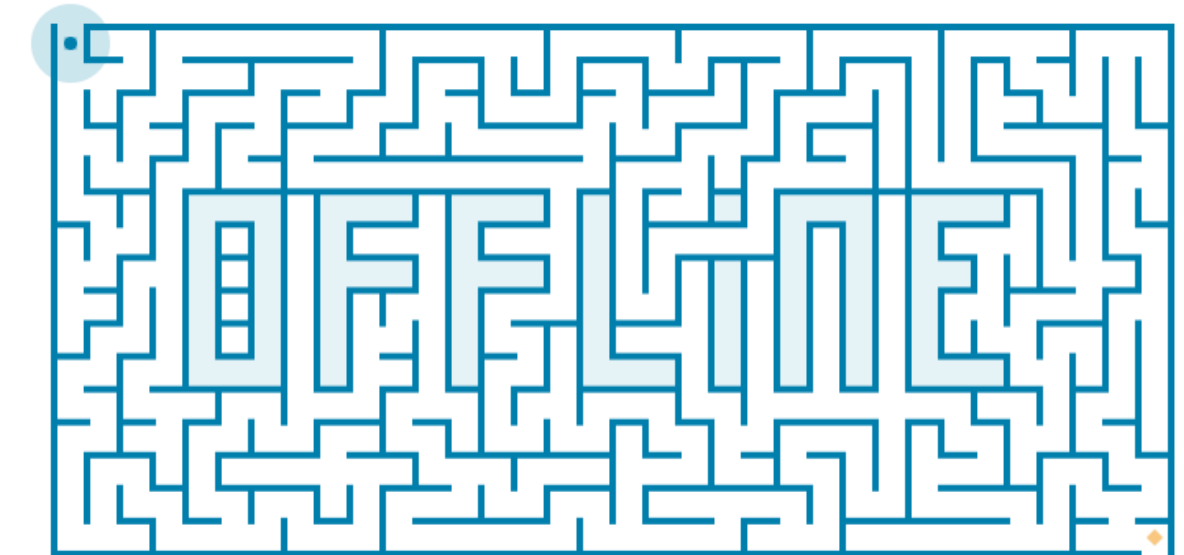
trivago

You are currently offline

We will try to reconnect you in 10 second(s)...

Reconnect

In the meantime, how about a trip in the offline maze?



00:00

html

network first

custom offline page

html

network first

cache as you go

custom offline page

html

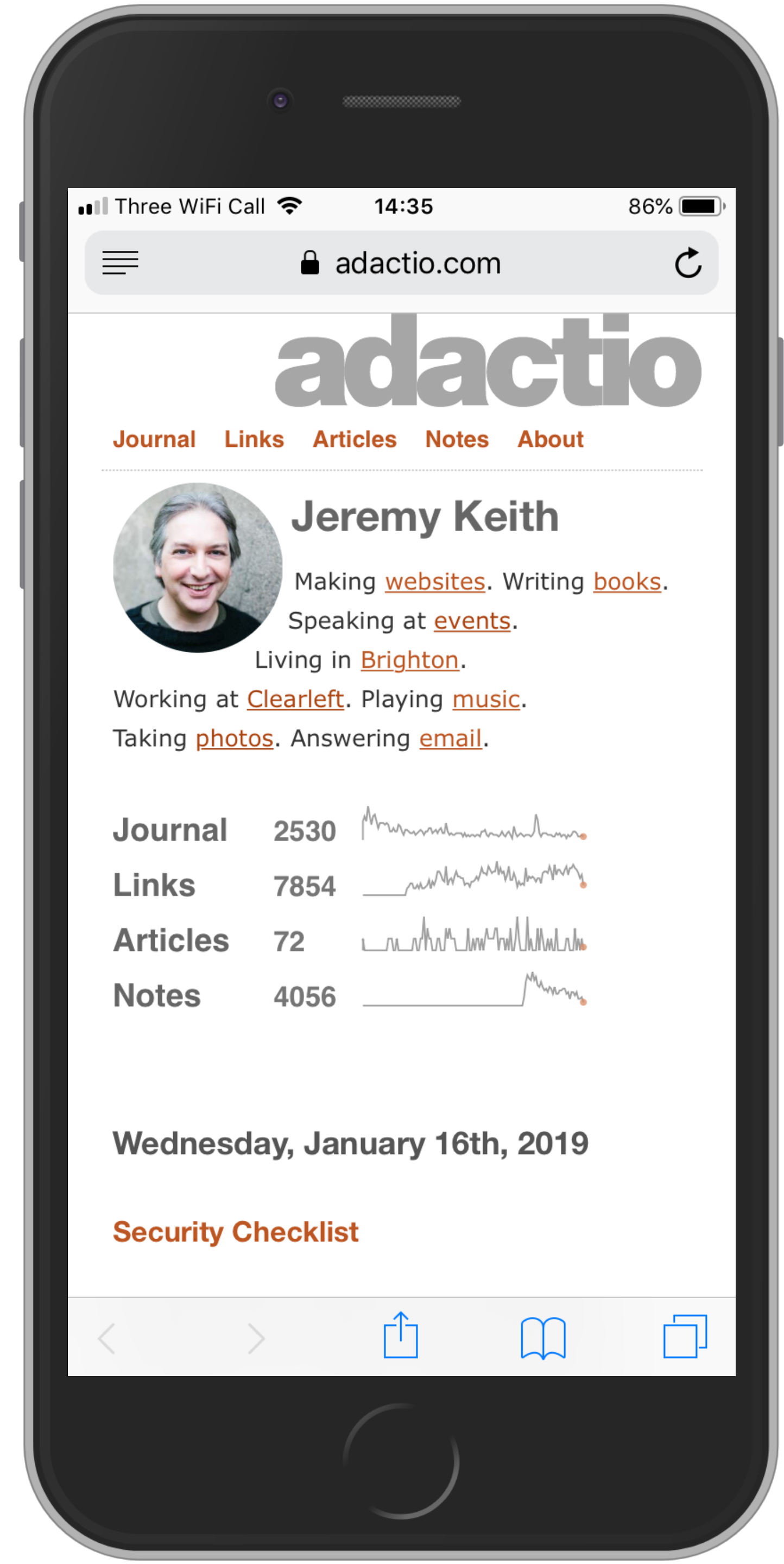
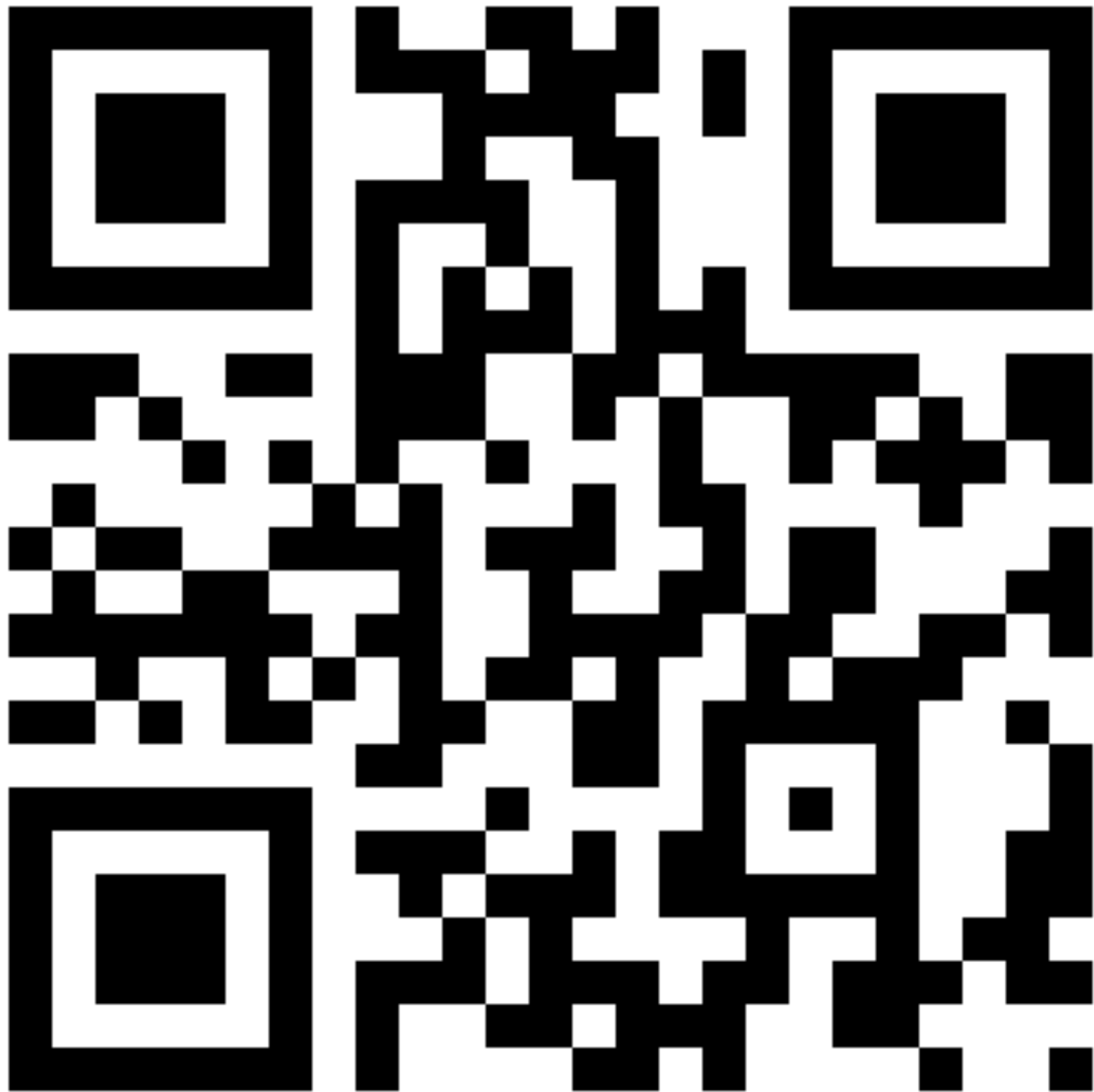
network

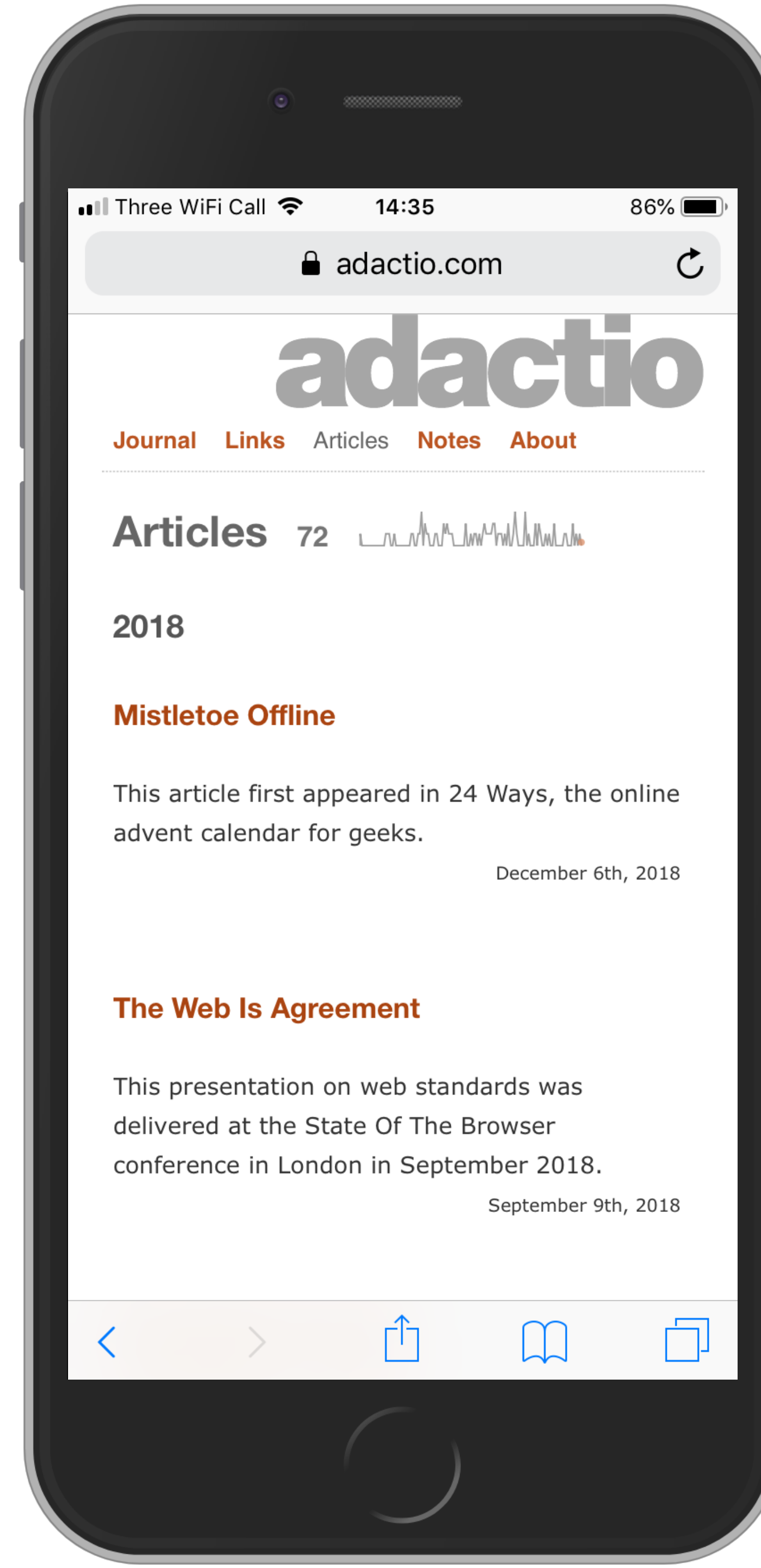
cache

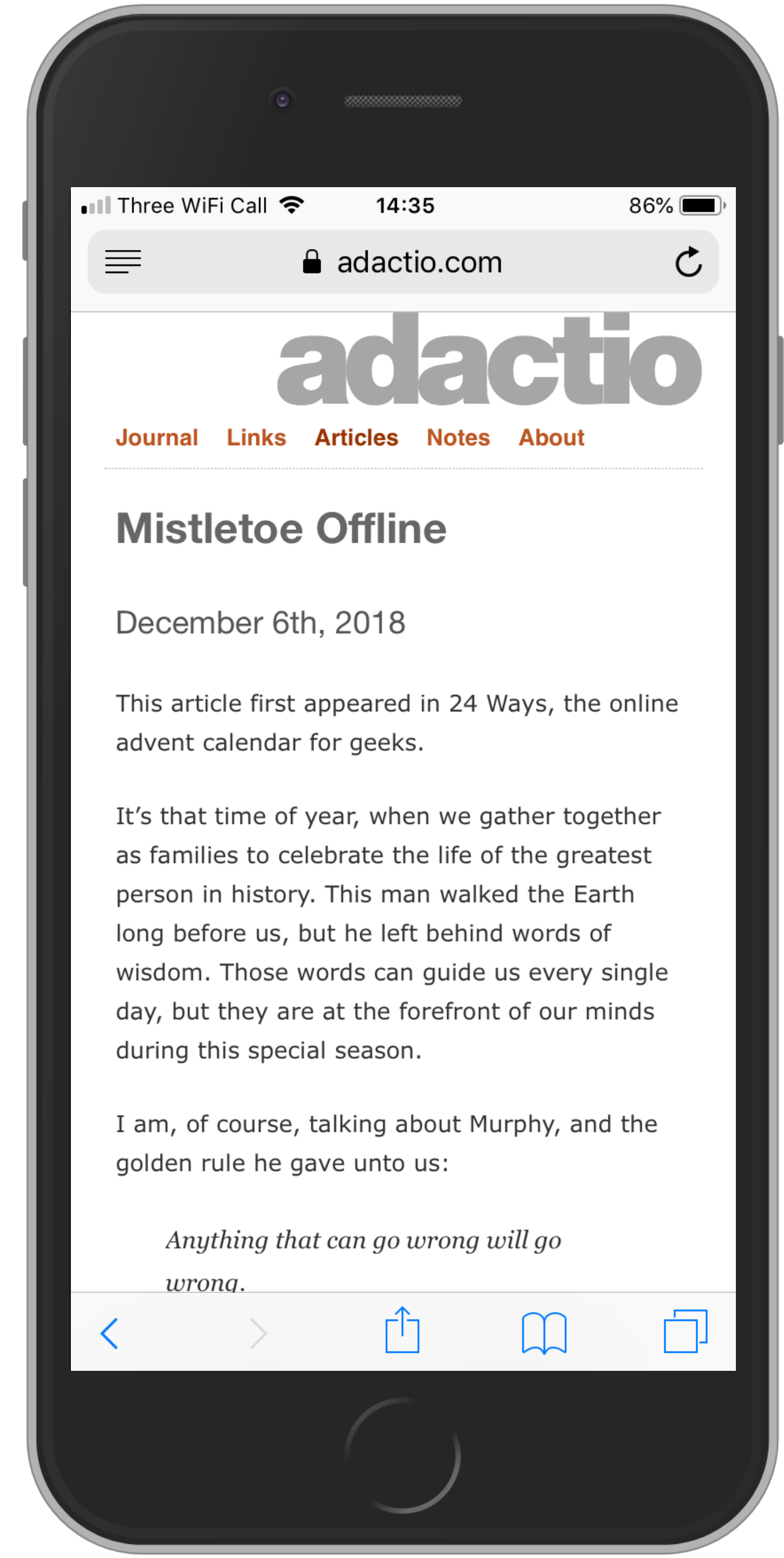
custom offline page

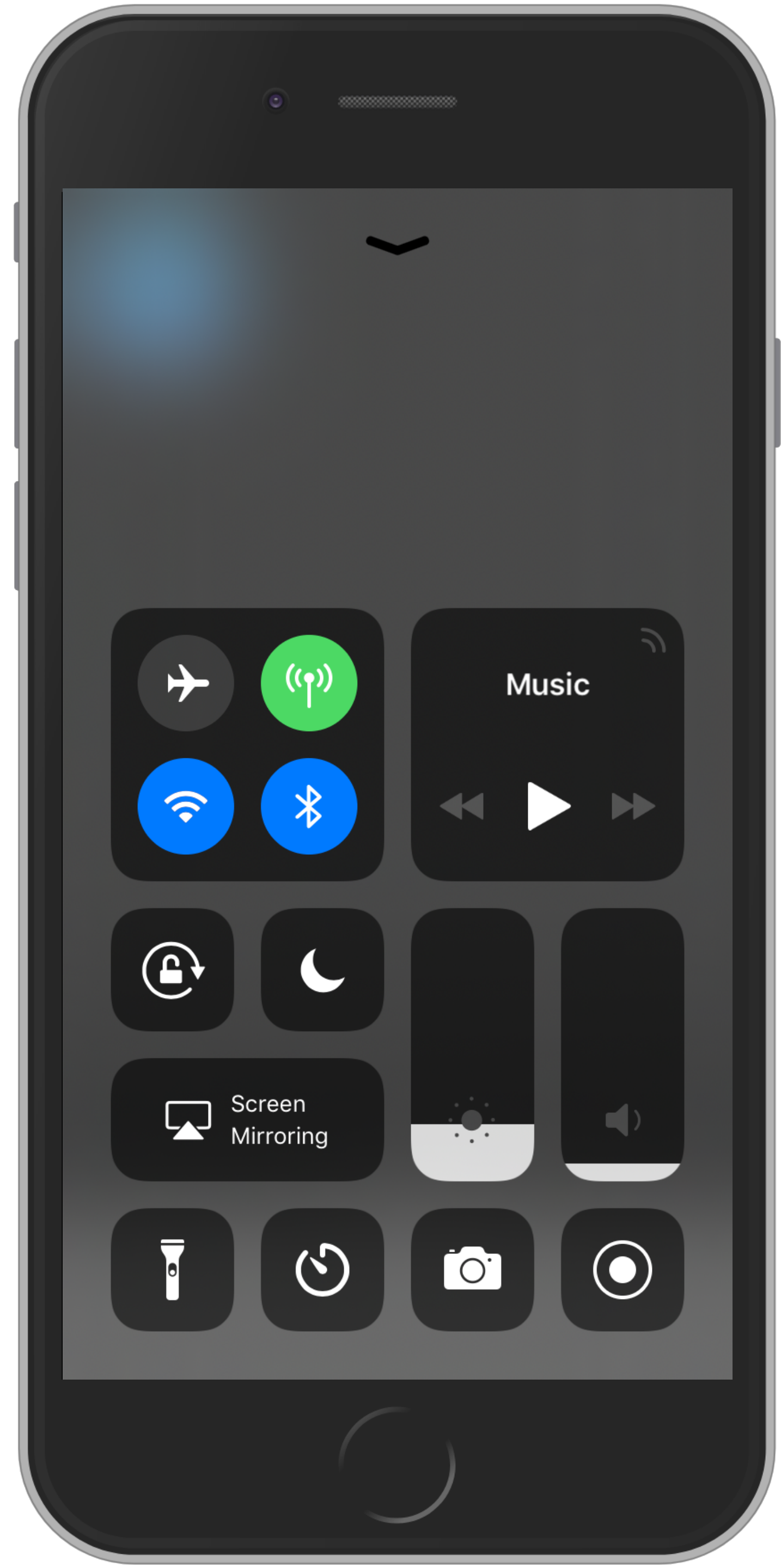
cache as you go

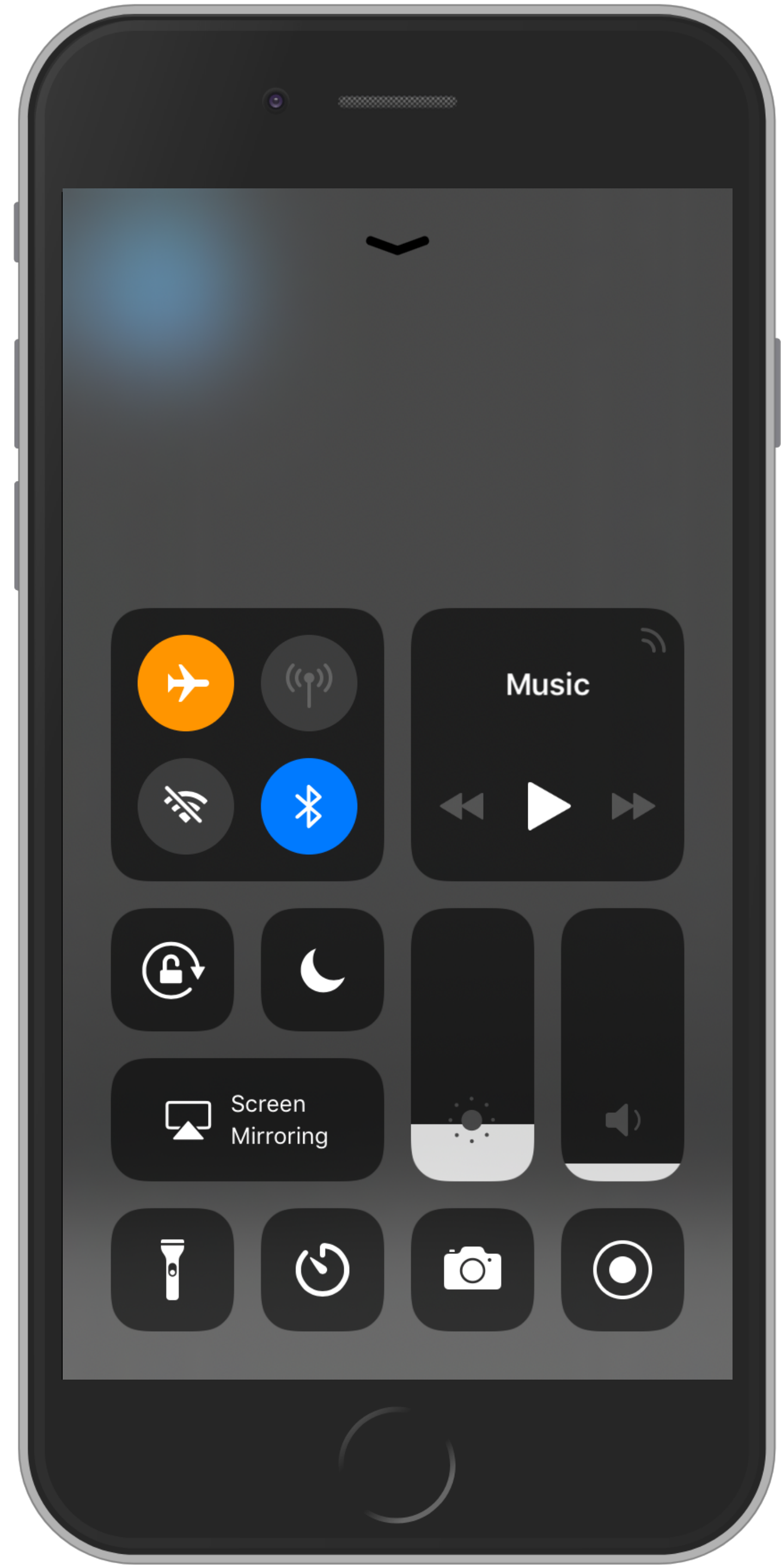
*When a page is requested
try to fetch the page from the network first
and store a copy in the cache
otherwise try to retrieve the page from the cache
otherwise retrieve the offline page from the cache.*

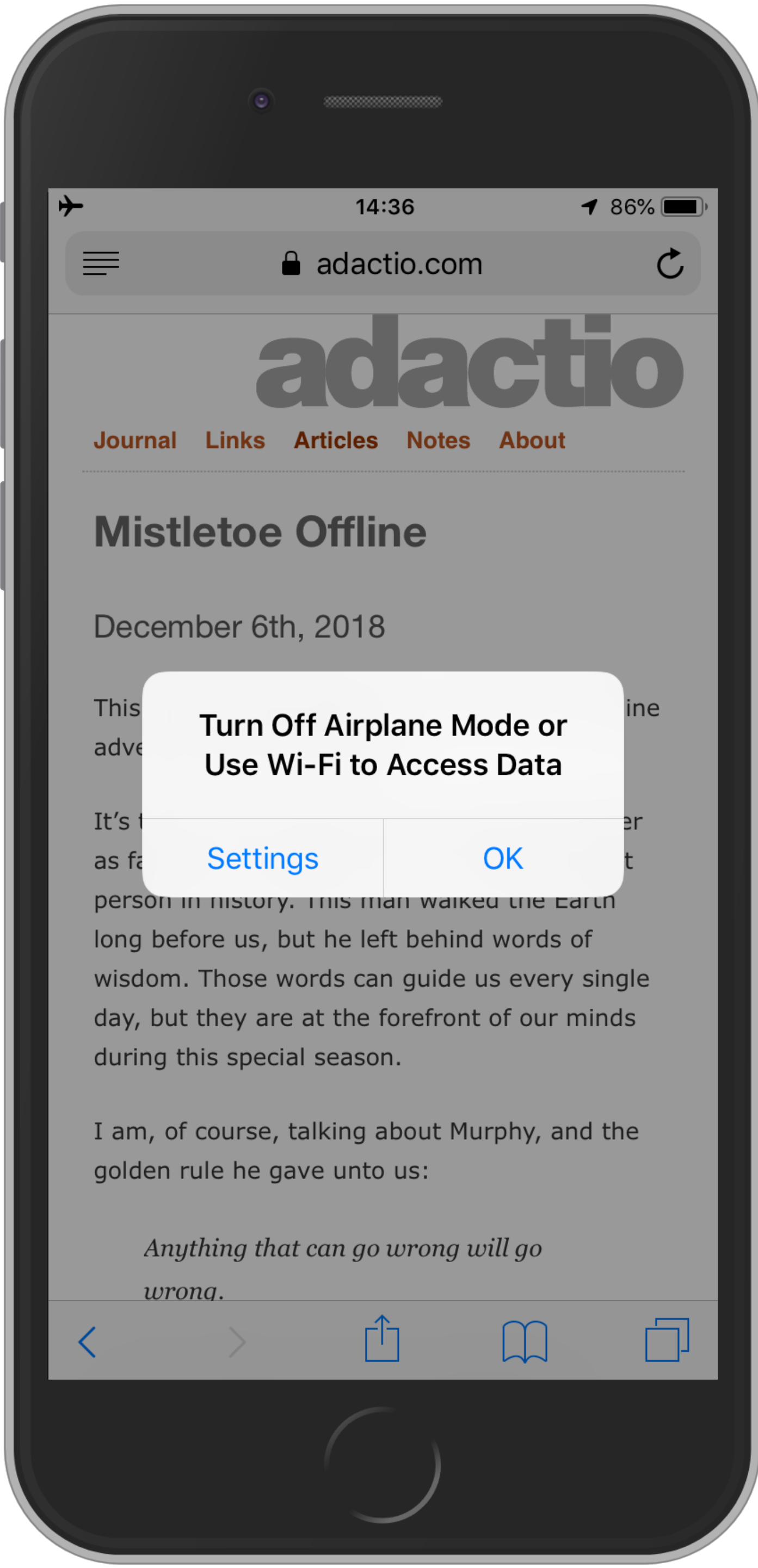
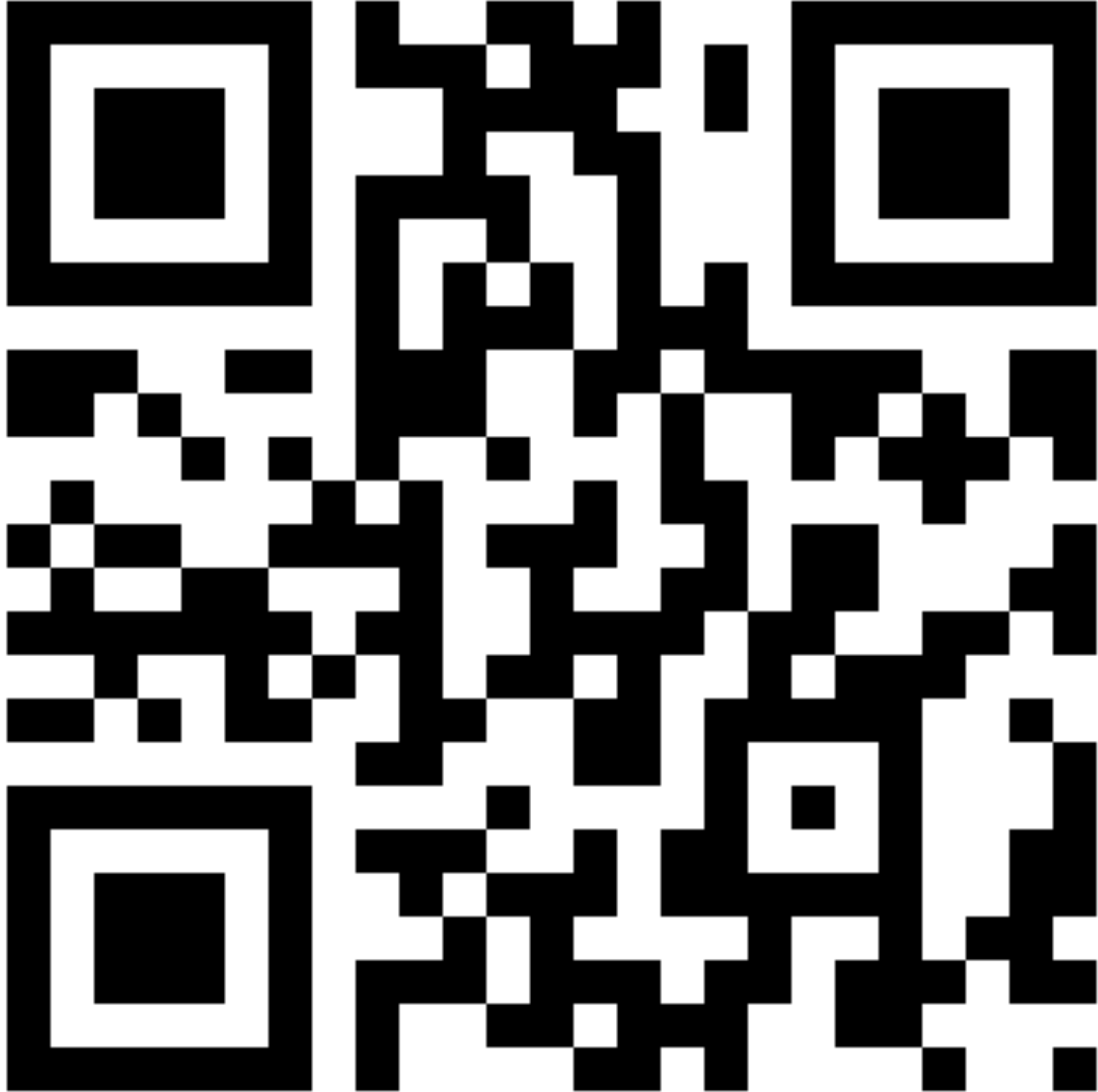


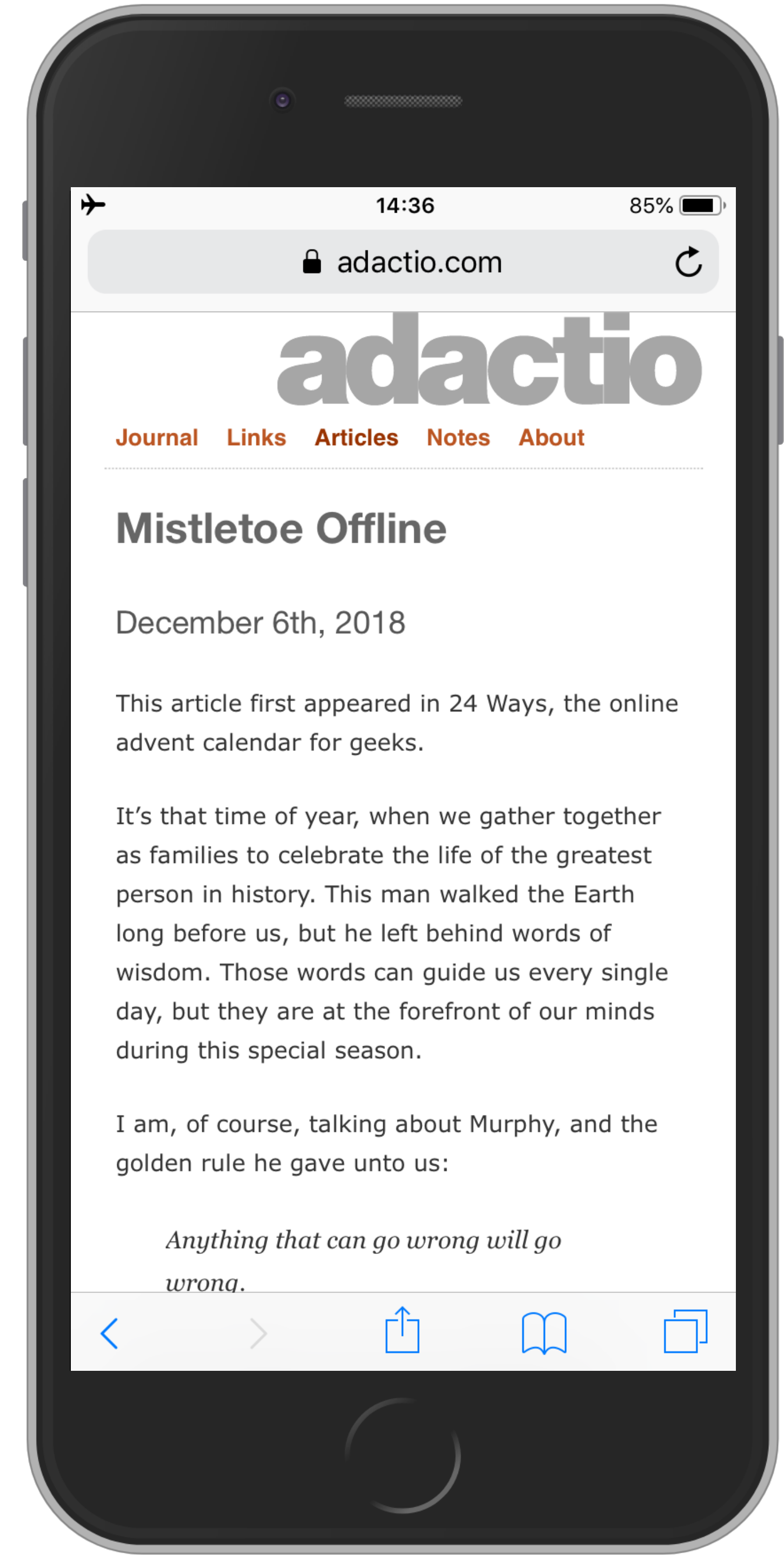
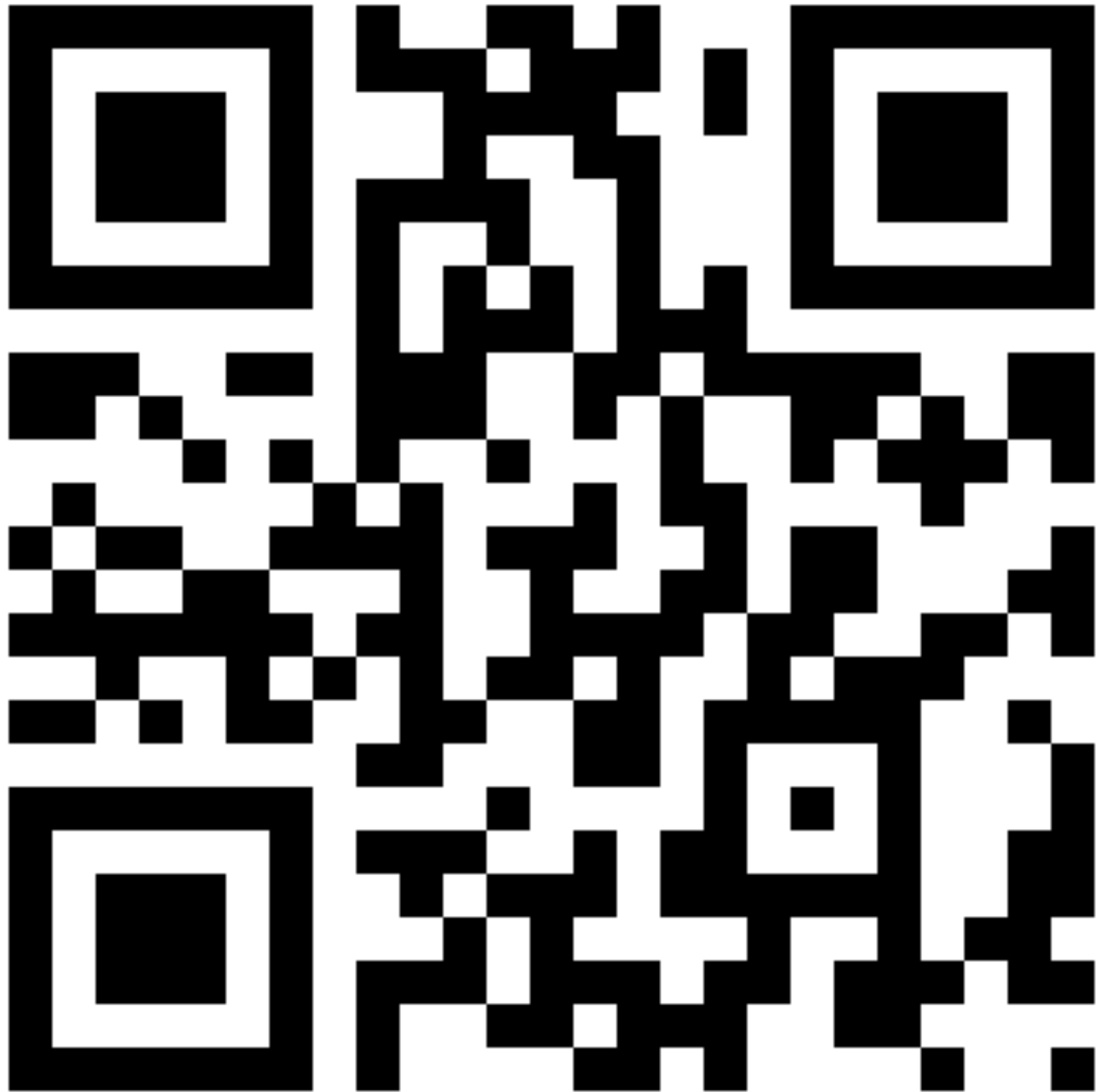


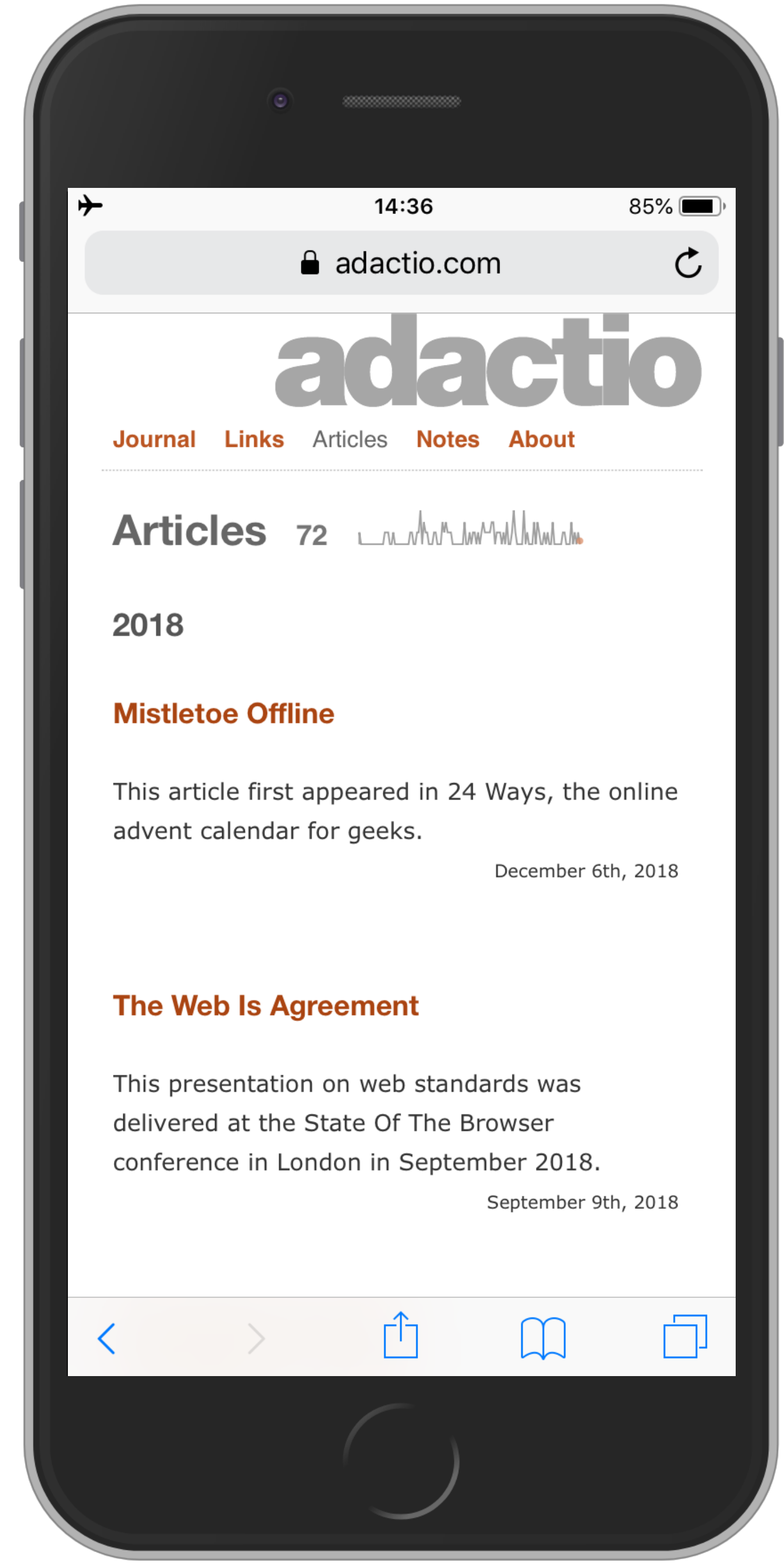


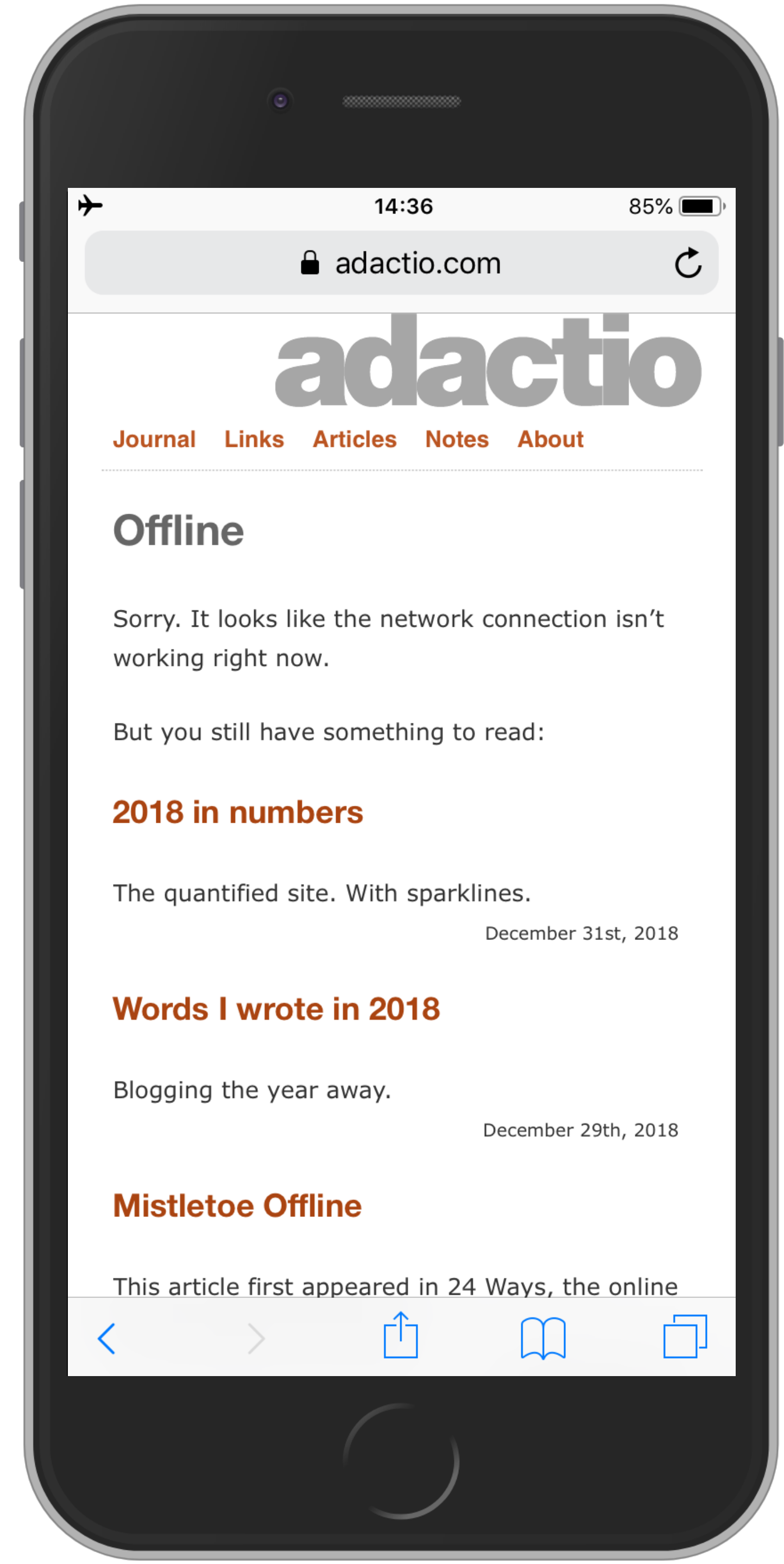
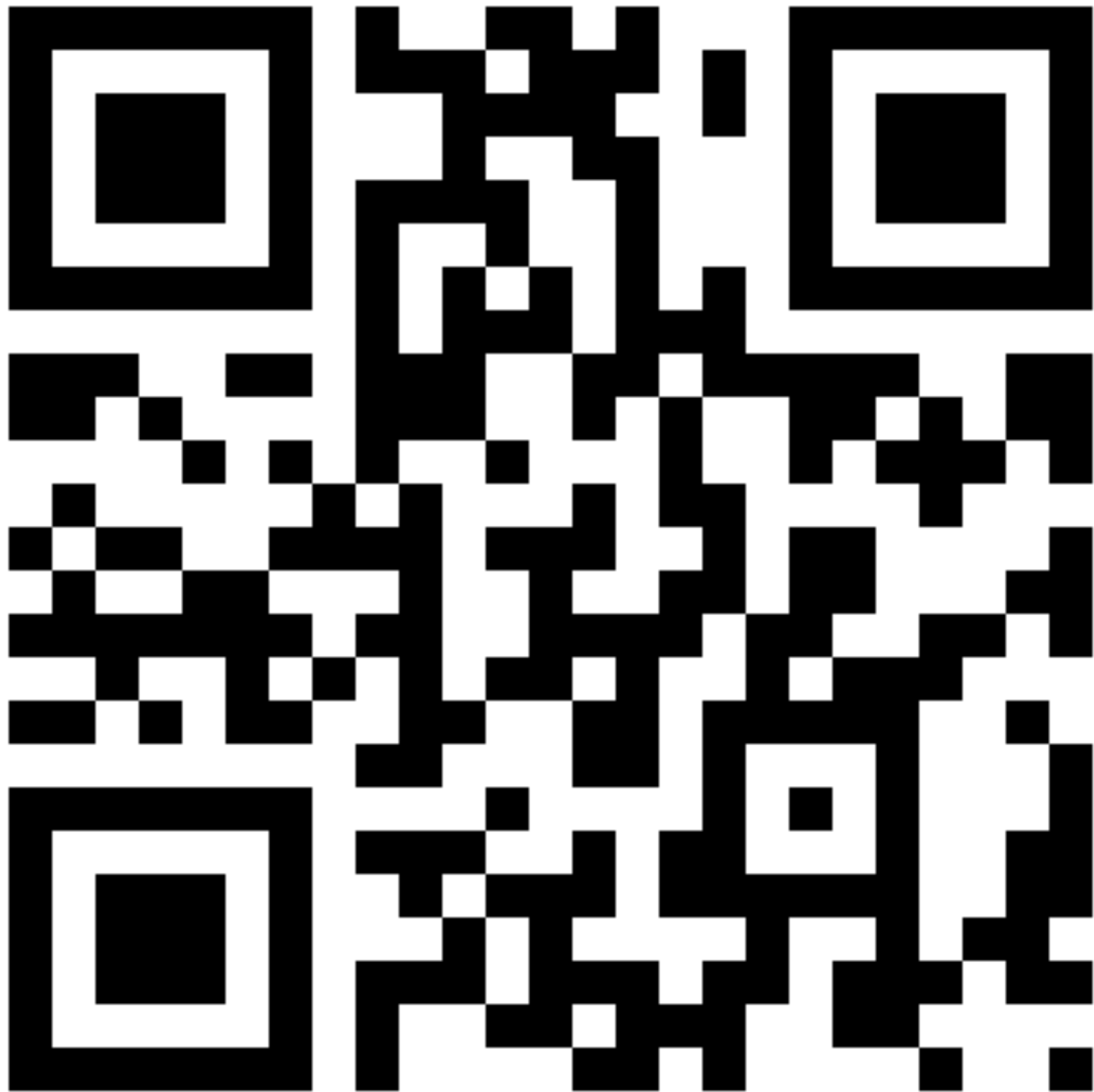




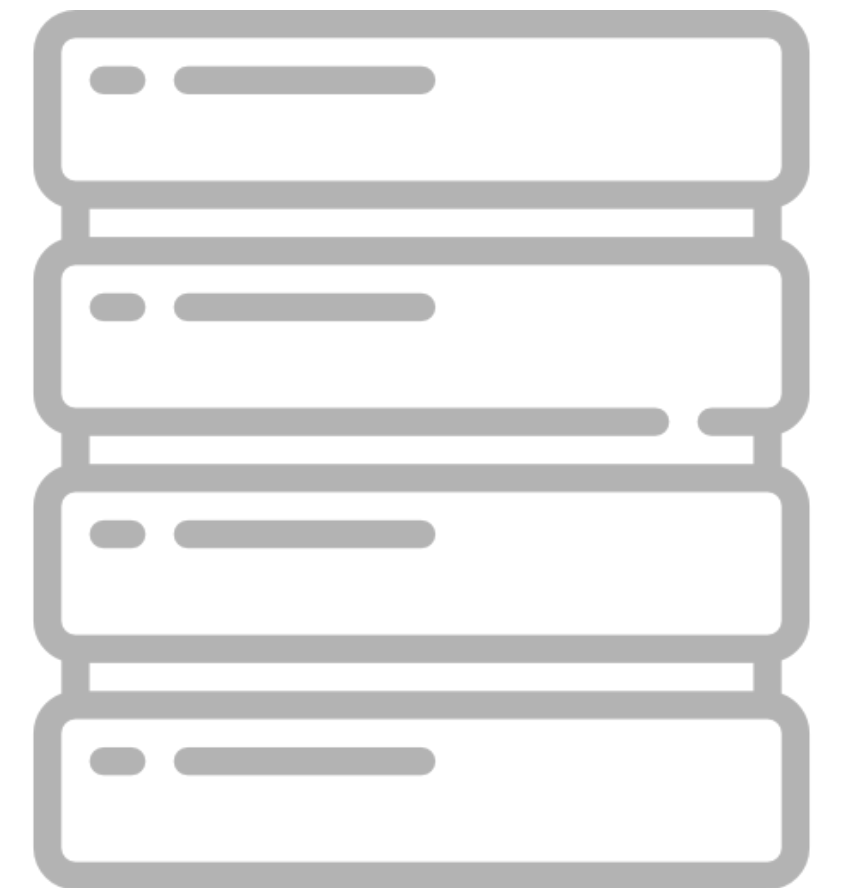
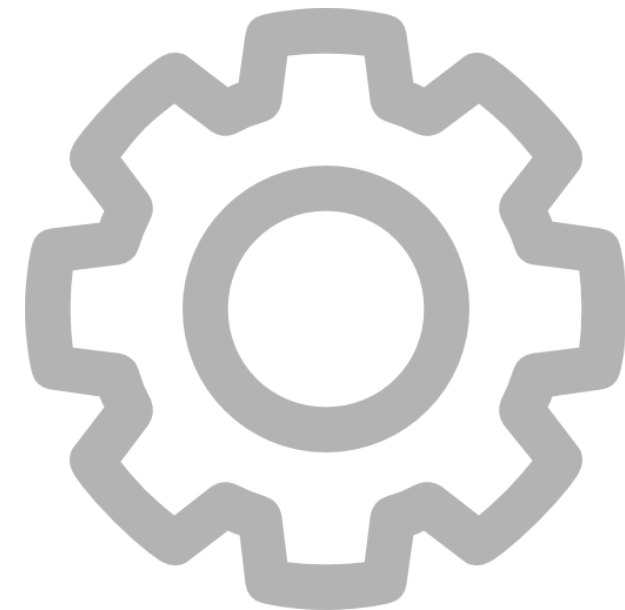
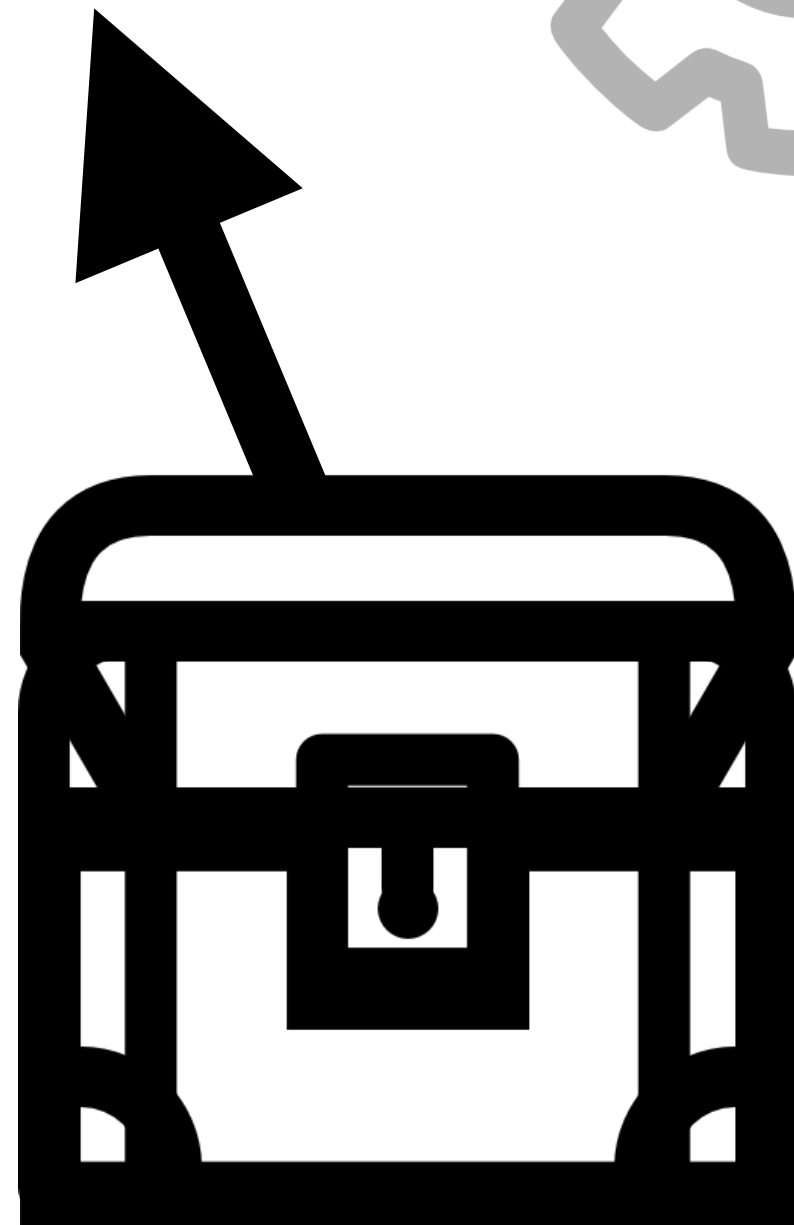


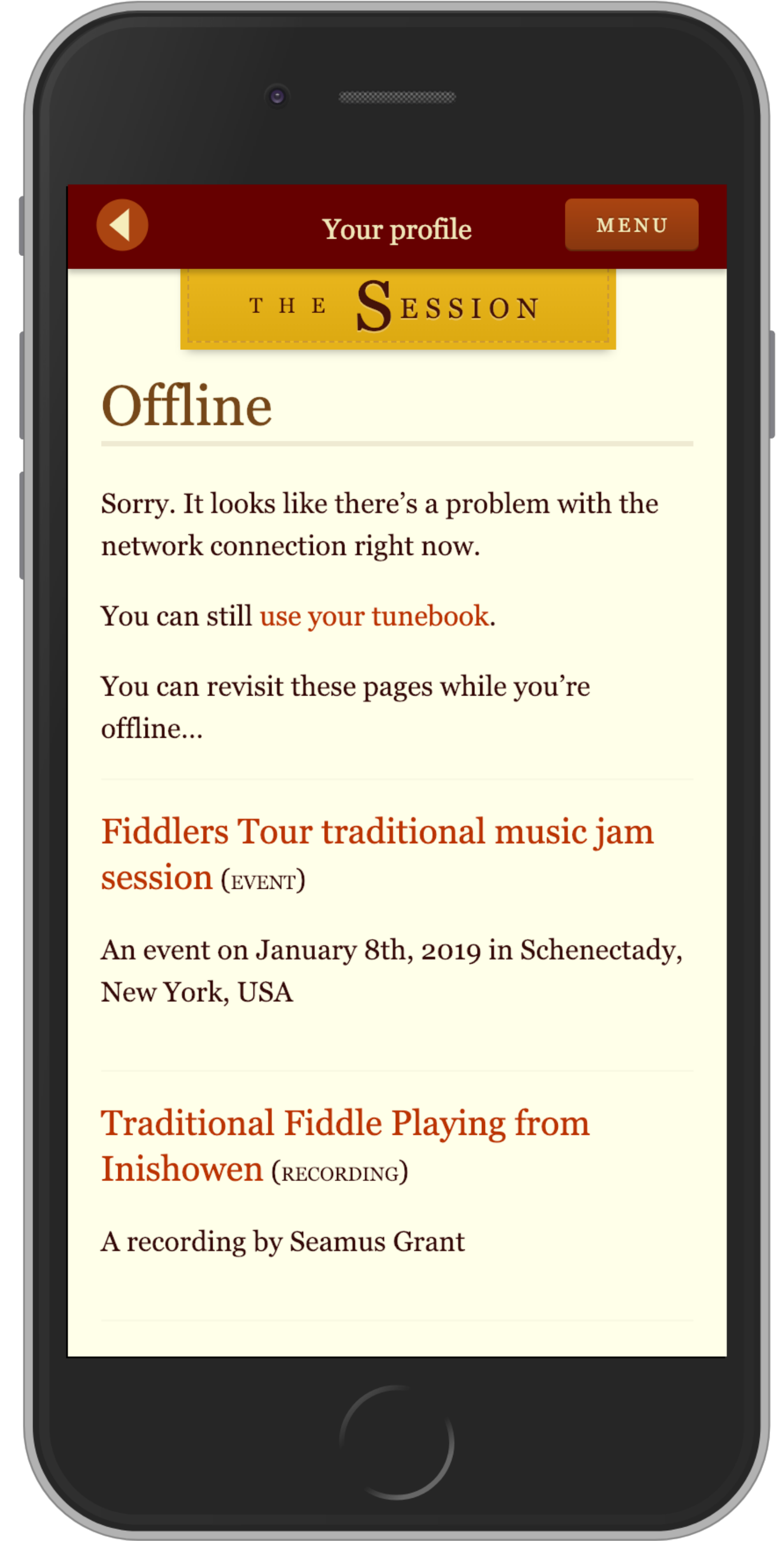
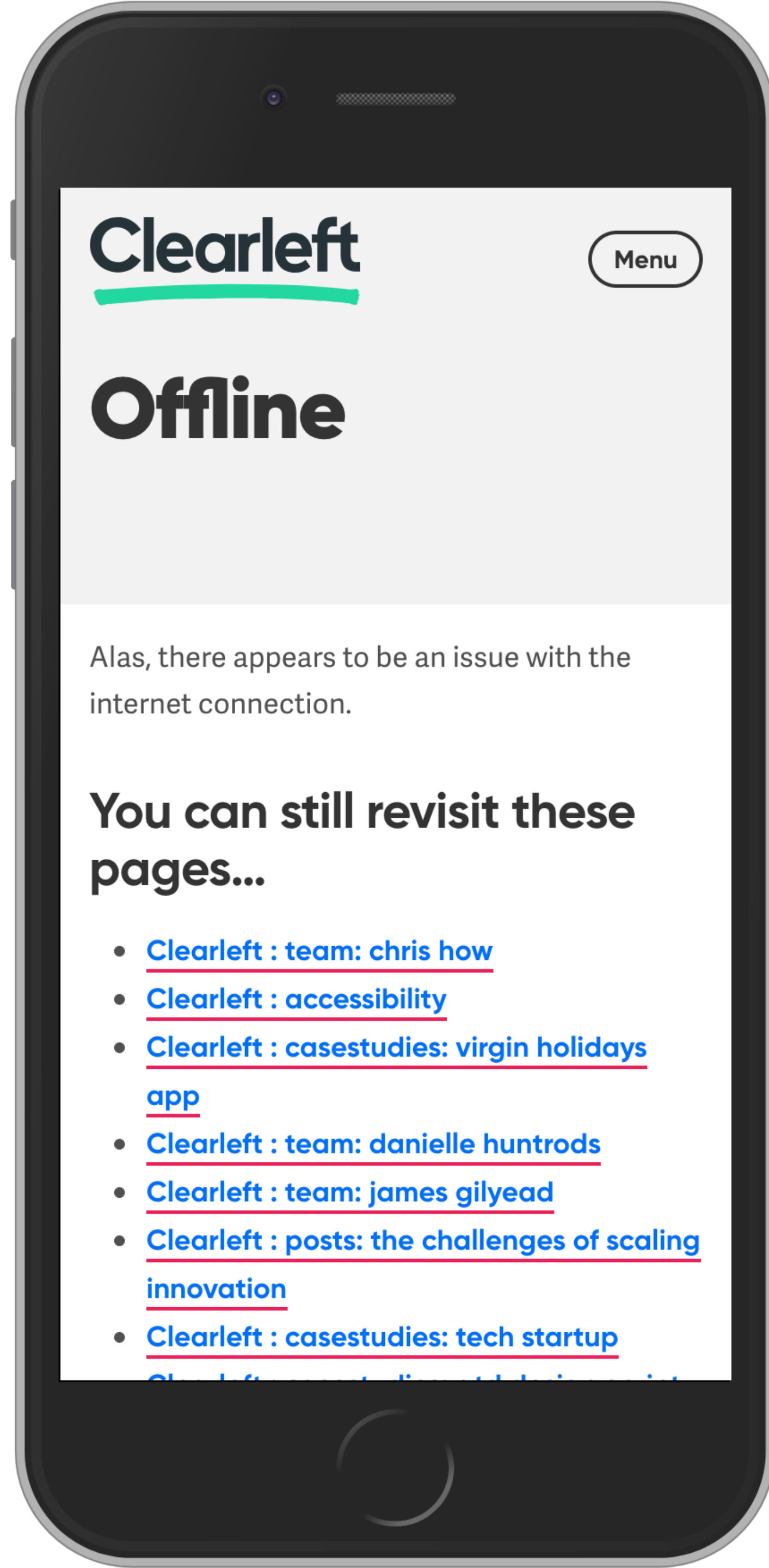






custom offline page





html css js img

network cache

cache network

fallback

html css js img

freshness speed

articles

shopping

travel

games

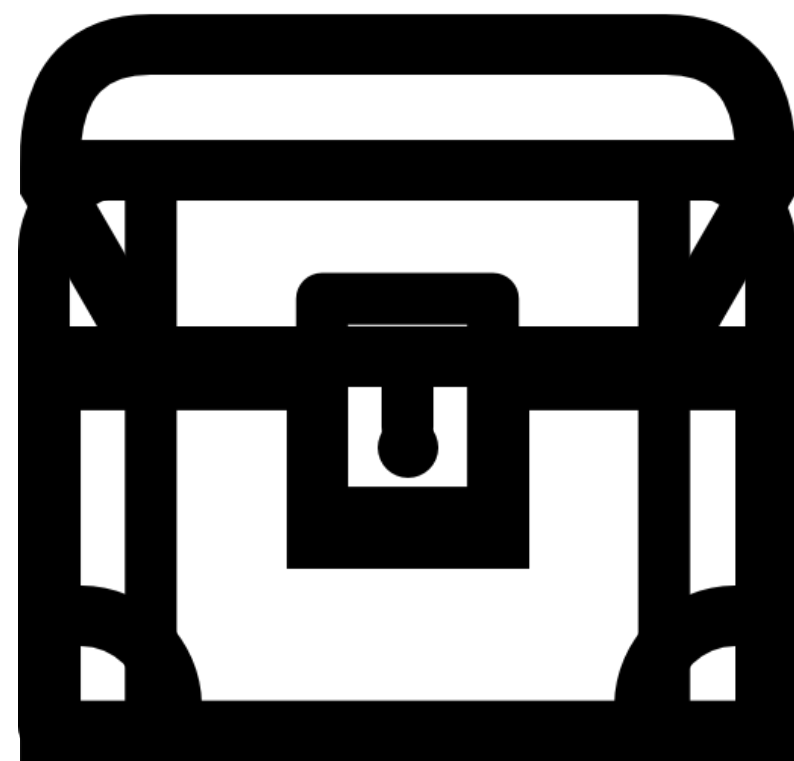
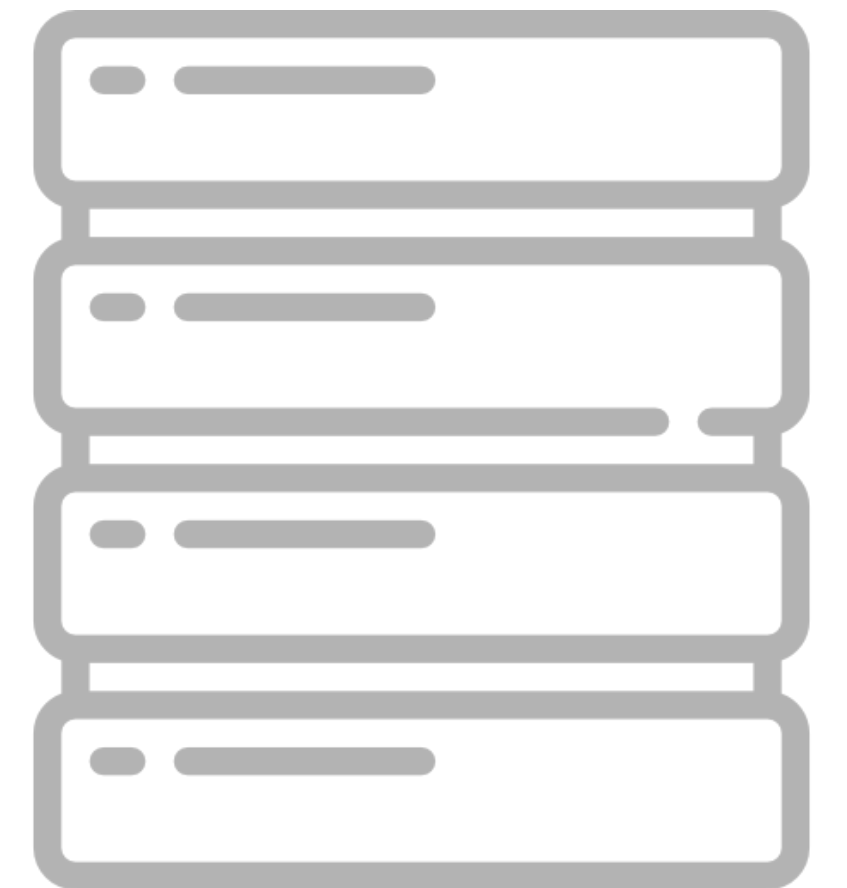
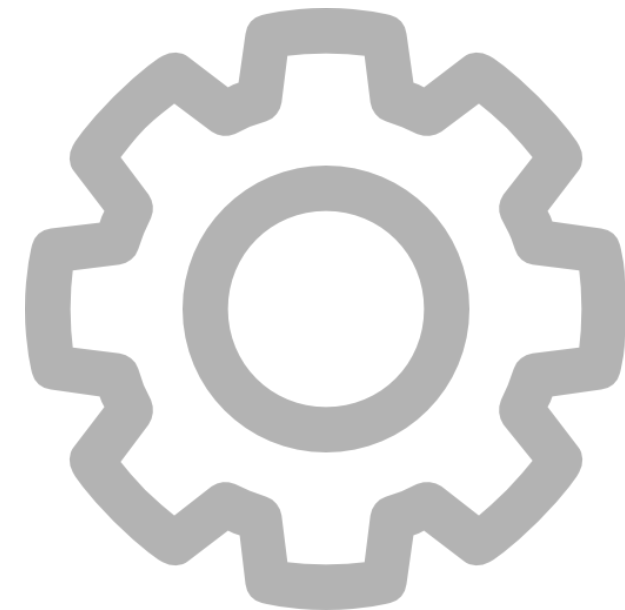
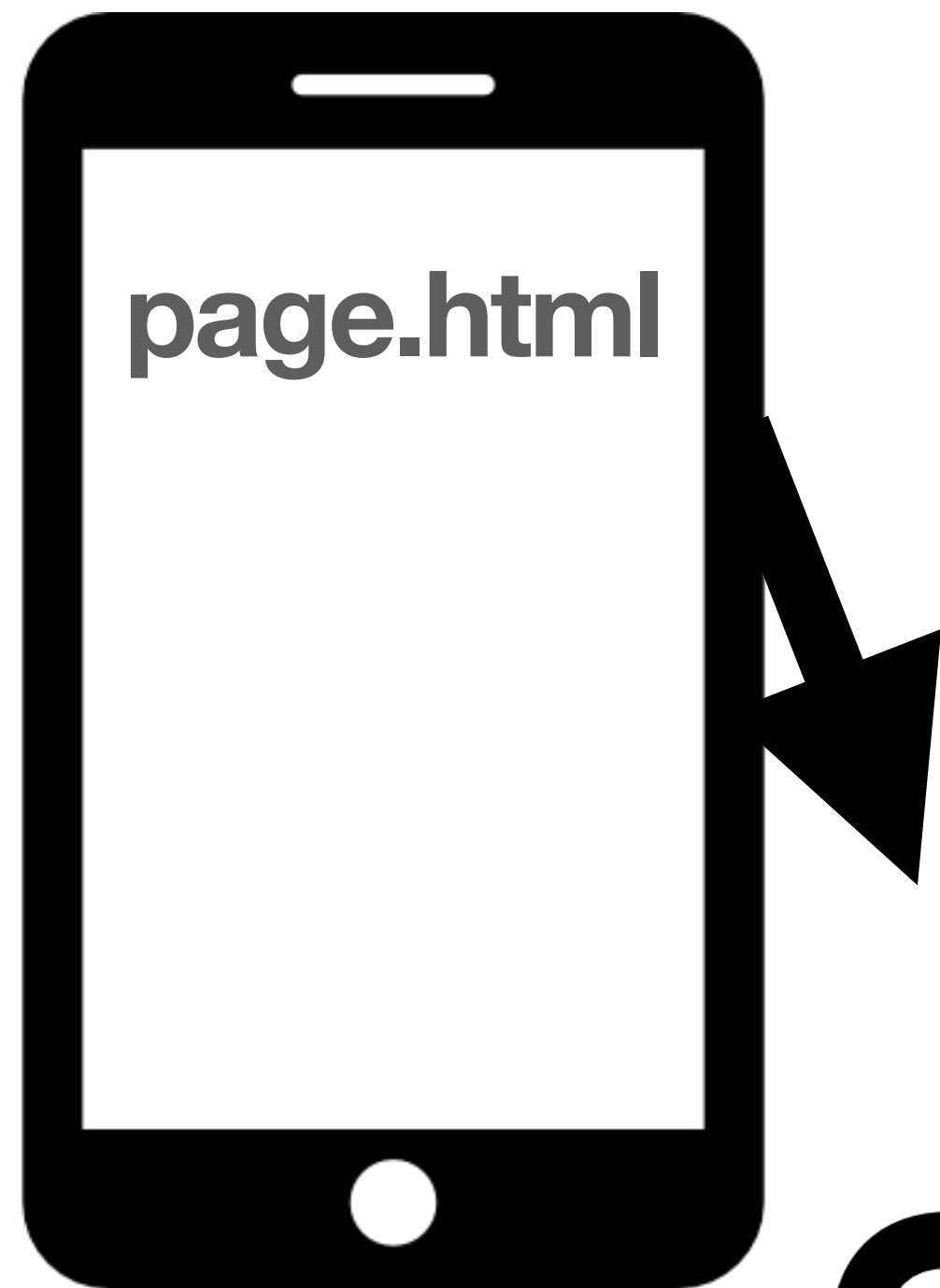
reference

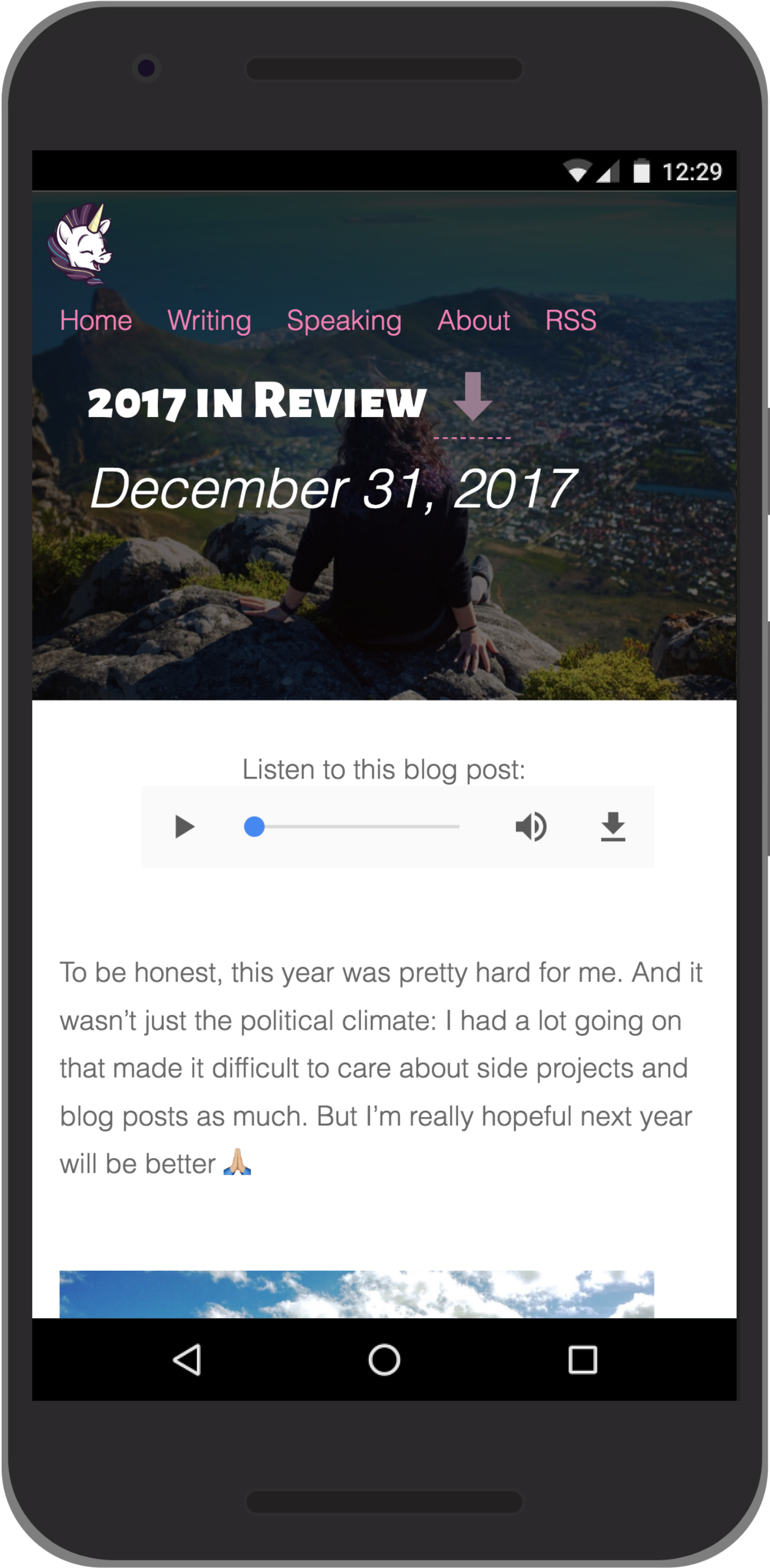
cache on demand
articles

cache on demand

*When the user presses a button
put a copy of this page in the cache.*

cache on demand

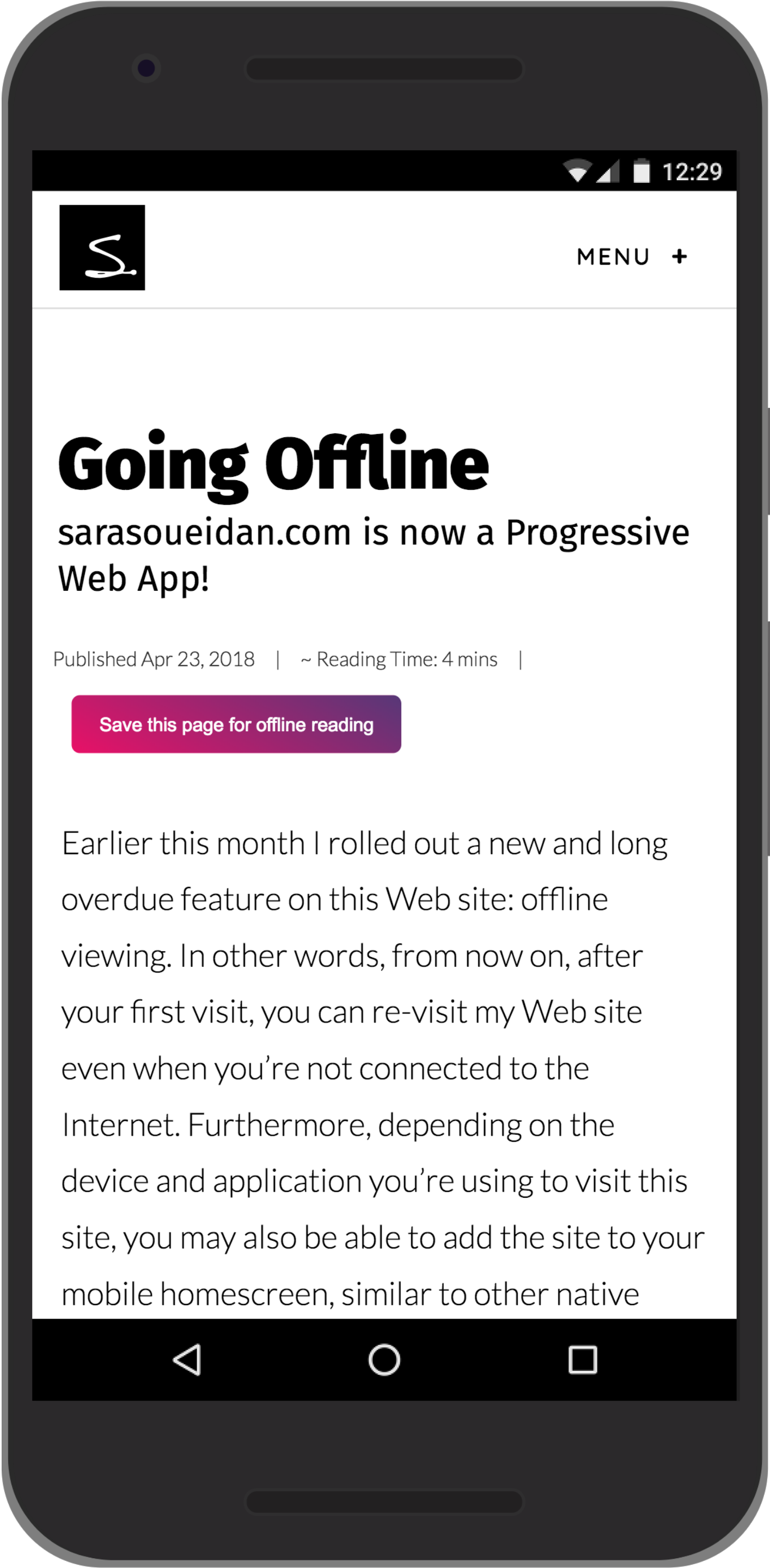




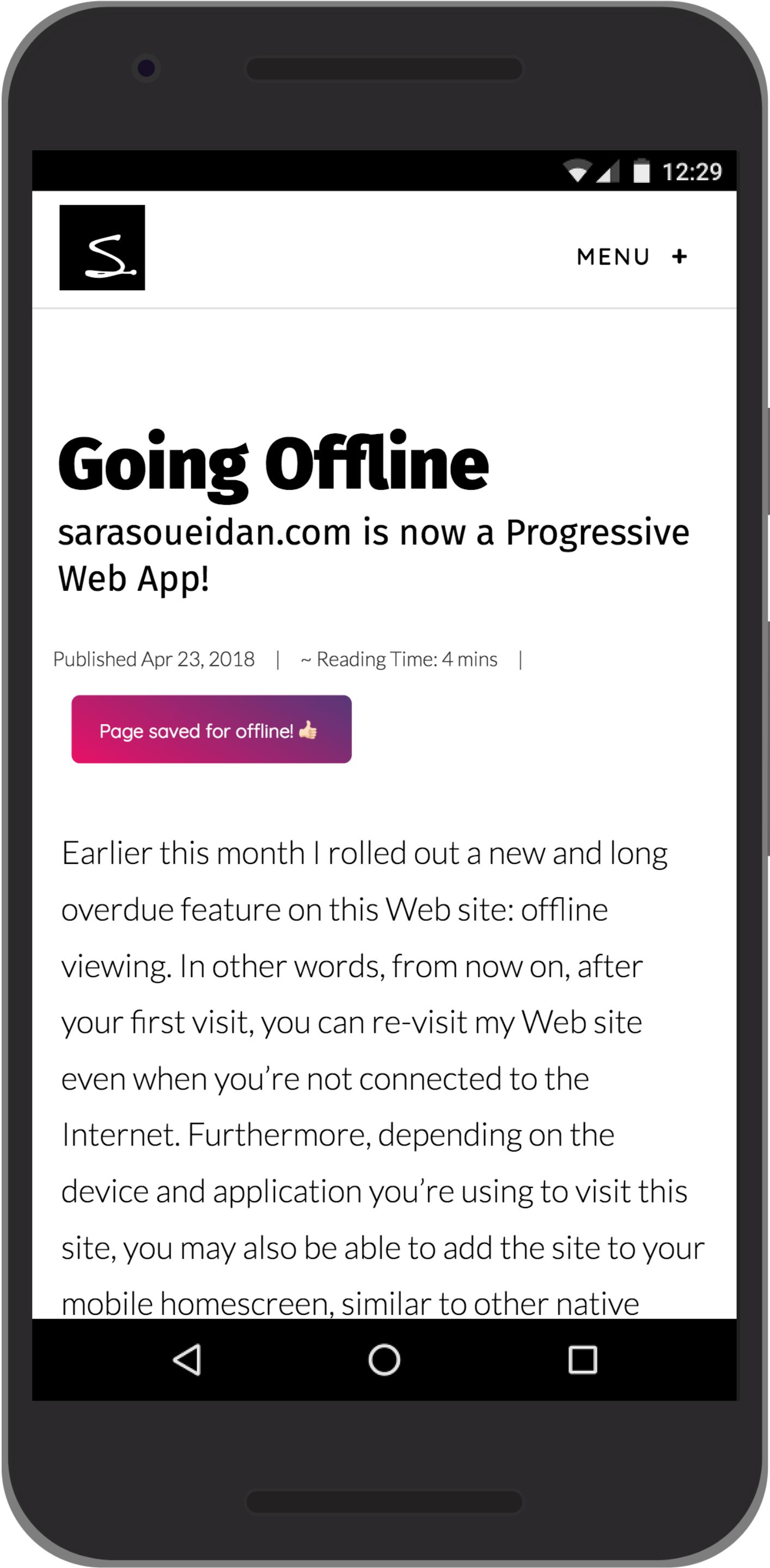




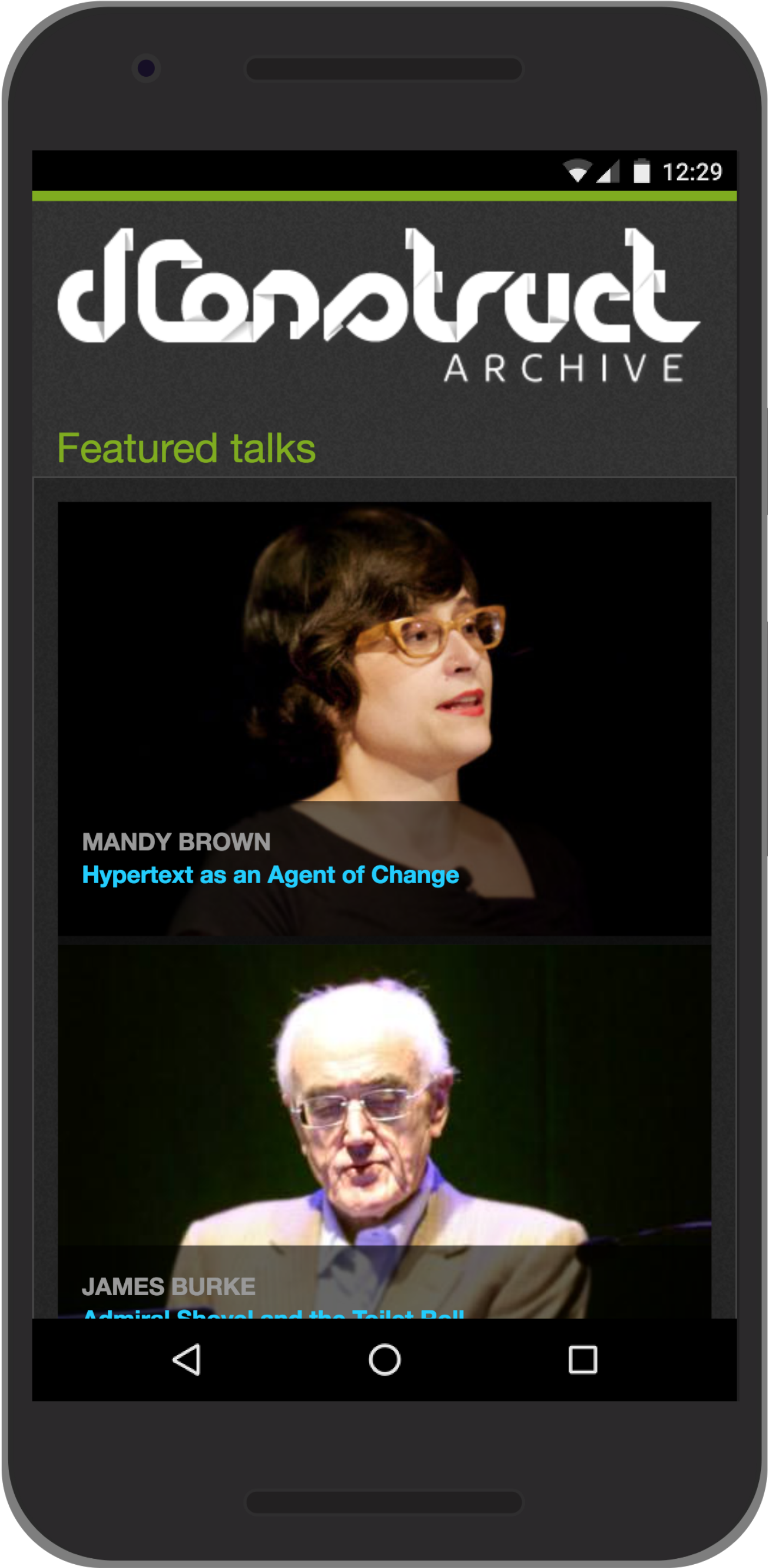
sarasoueidan.com



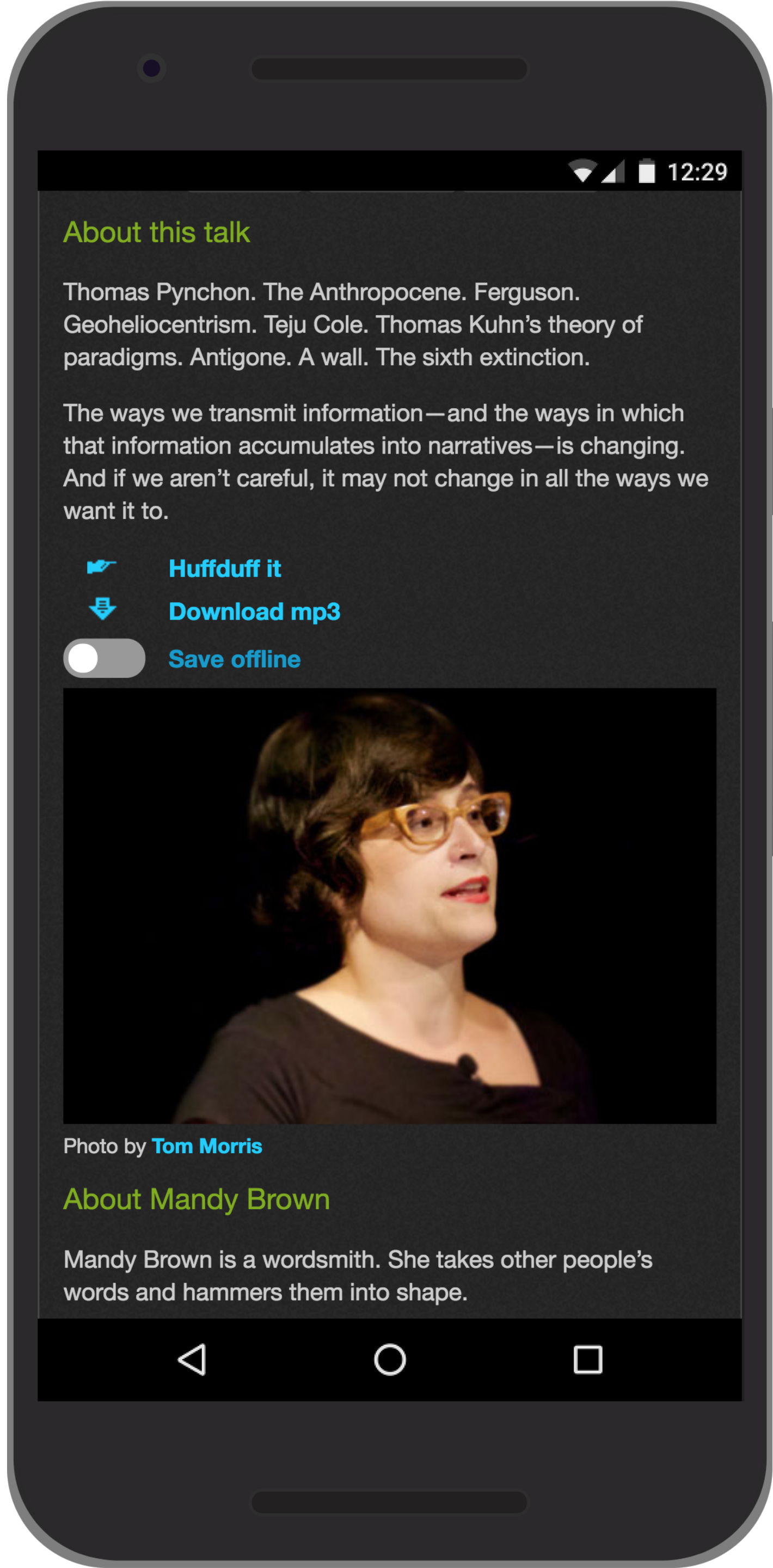
sarasoueidan.com



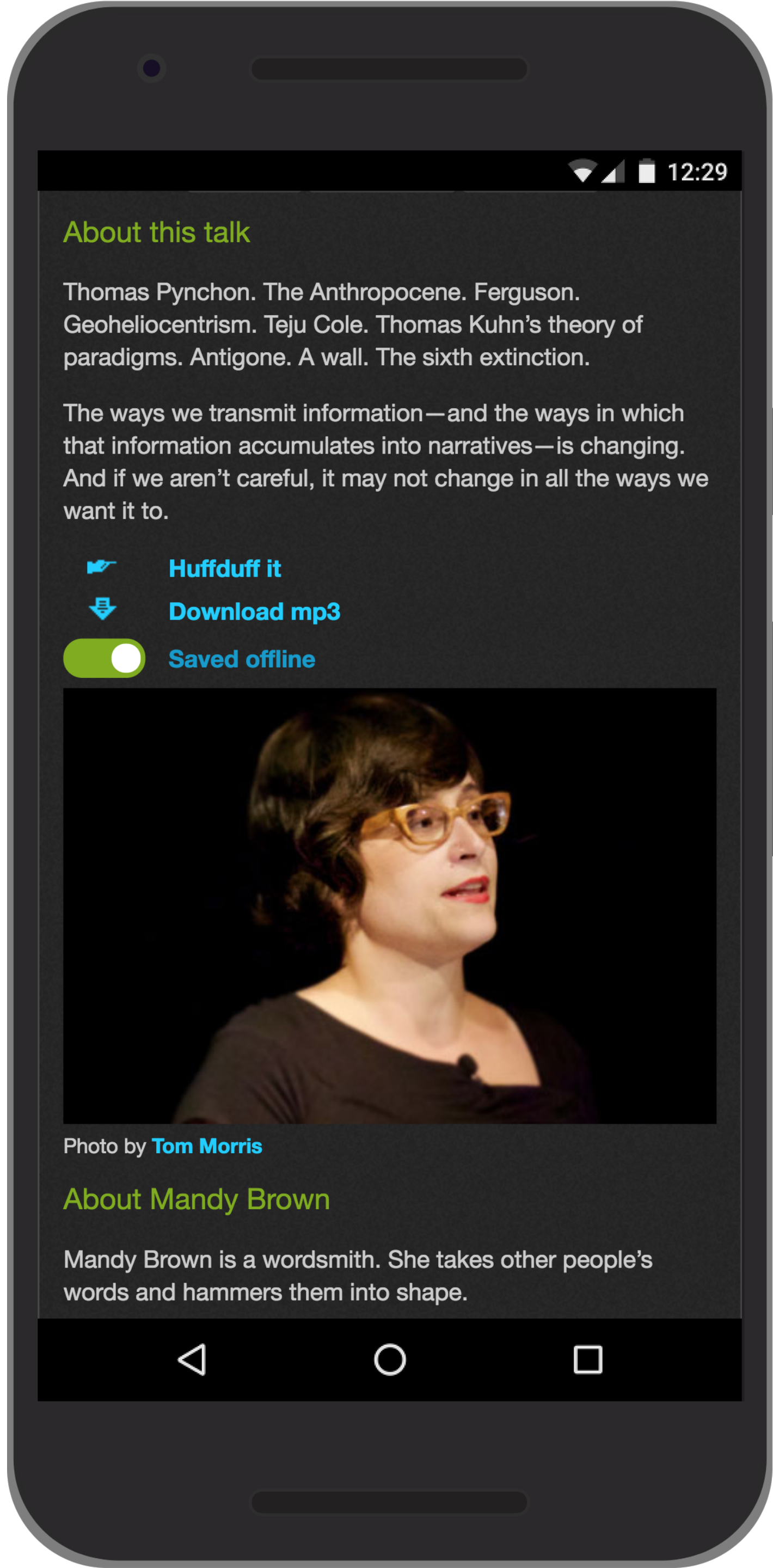
archive.dconstruct.org



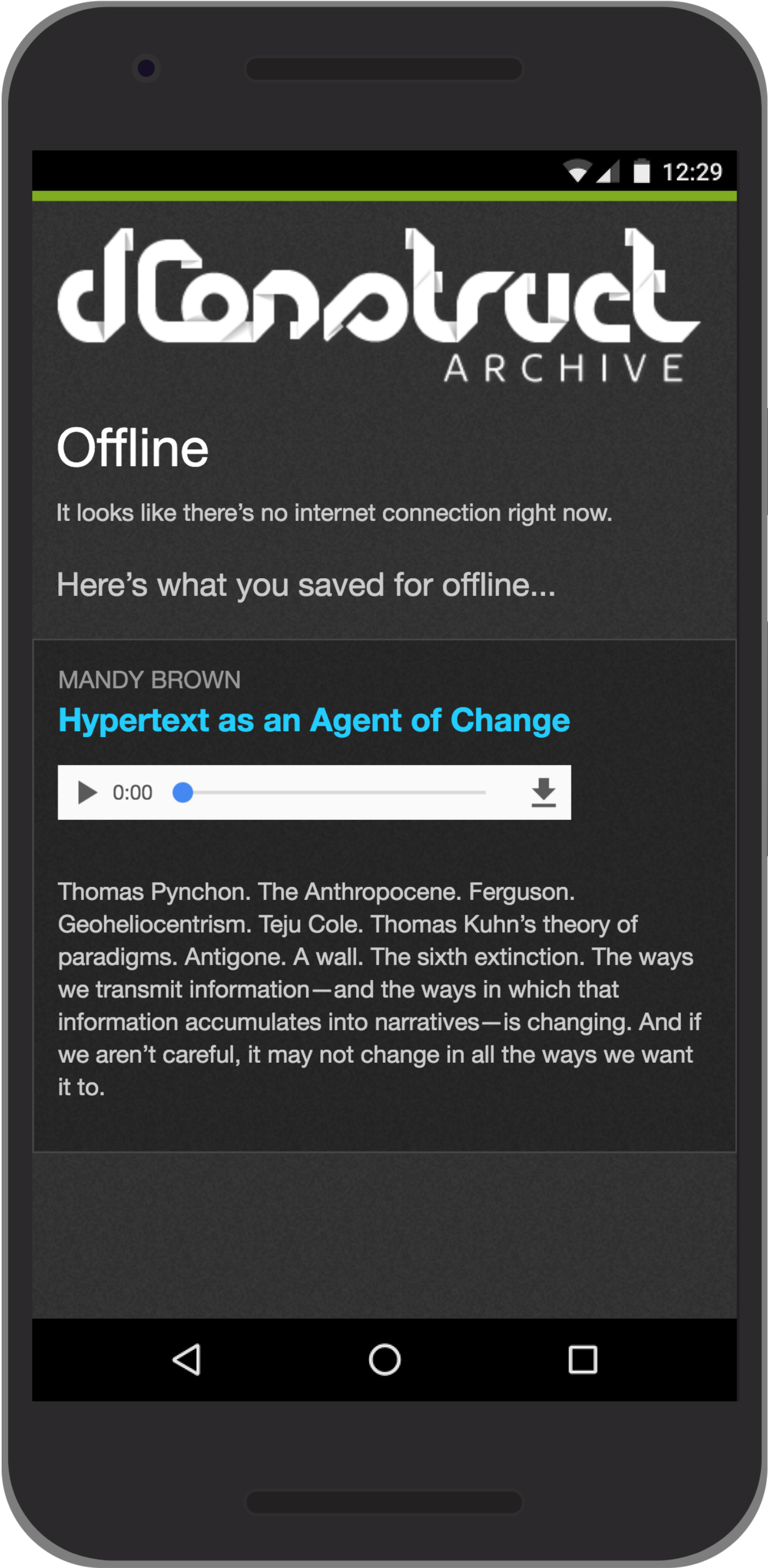
archive.dconstruct.org



archive.dconstruct.org



archive.dconstruct.org



web app manifest

https

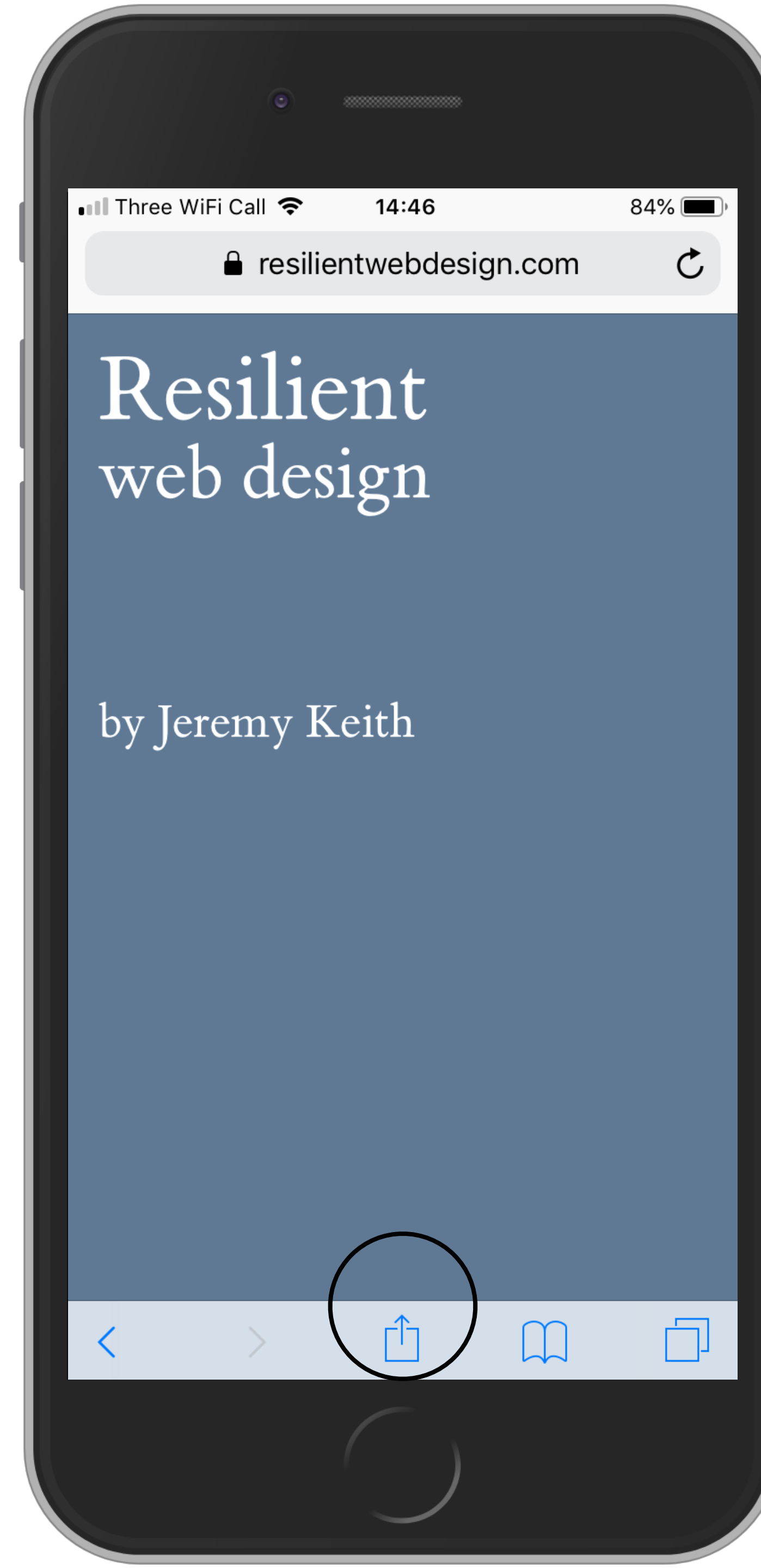
service worker

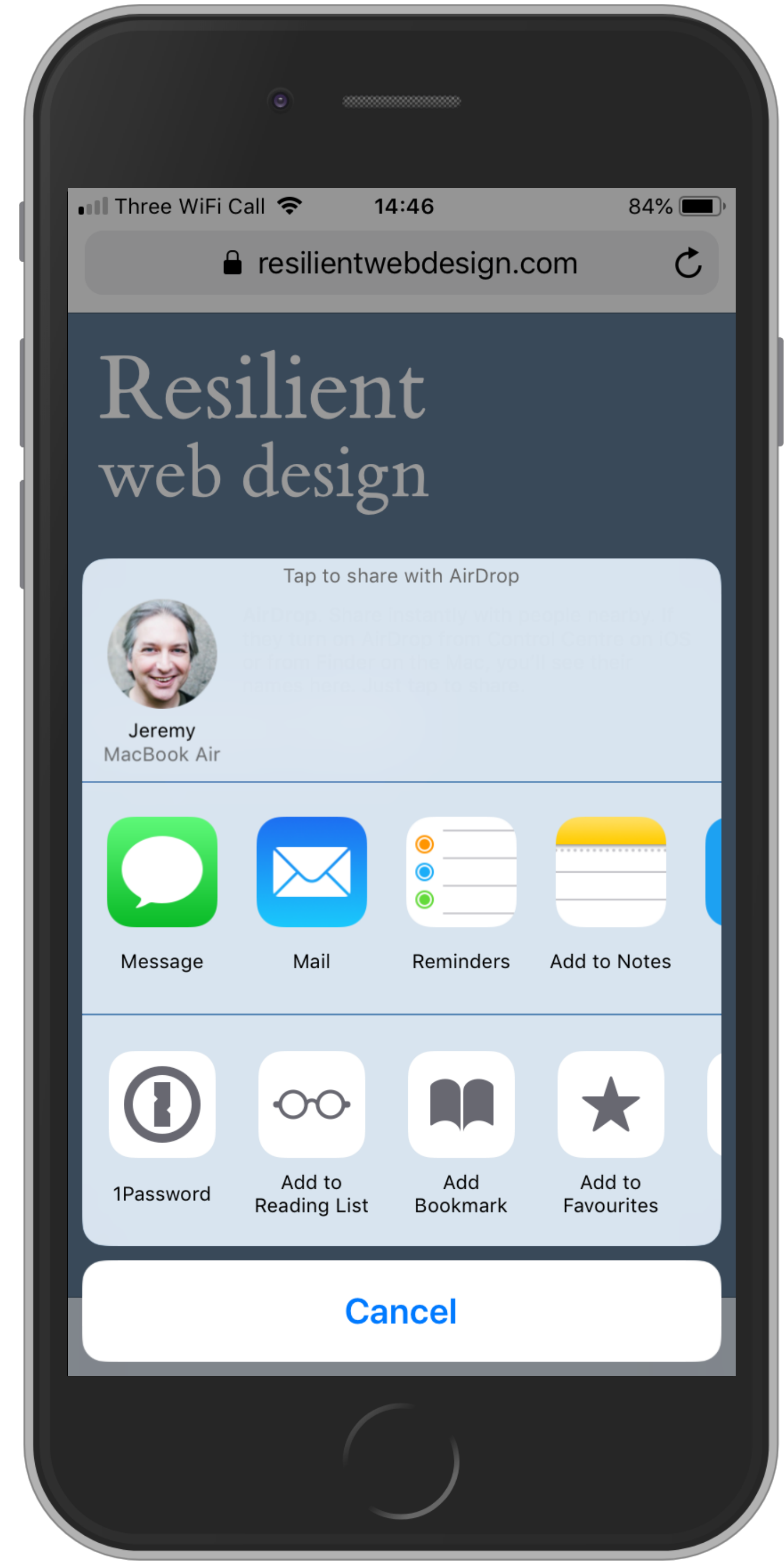
web app manifest

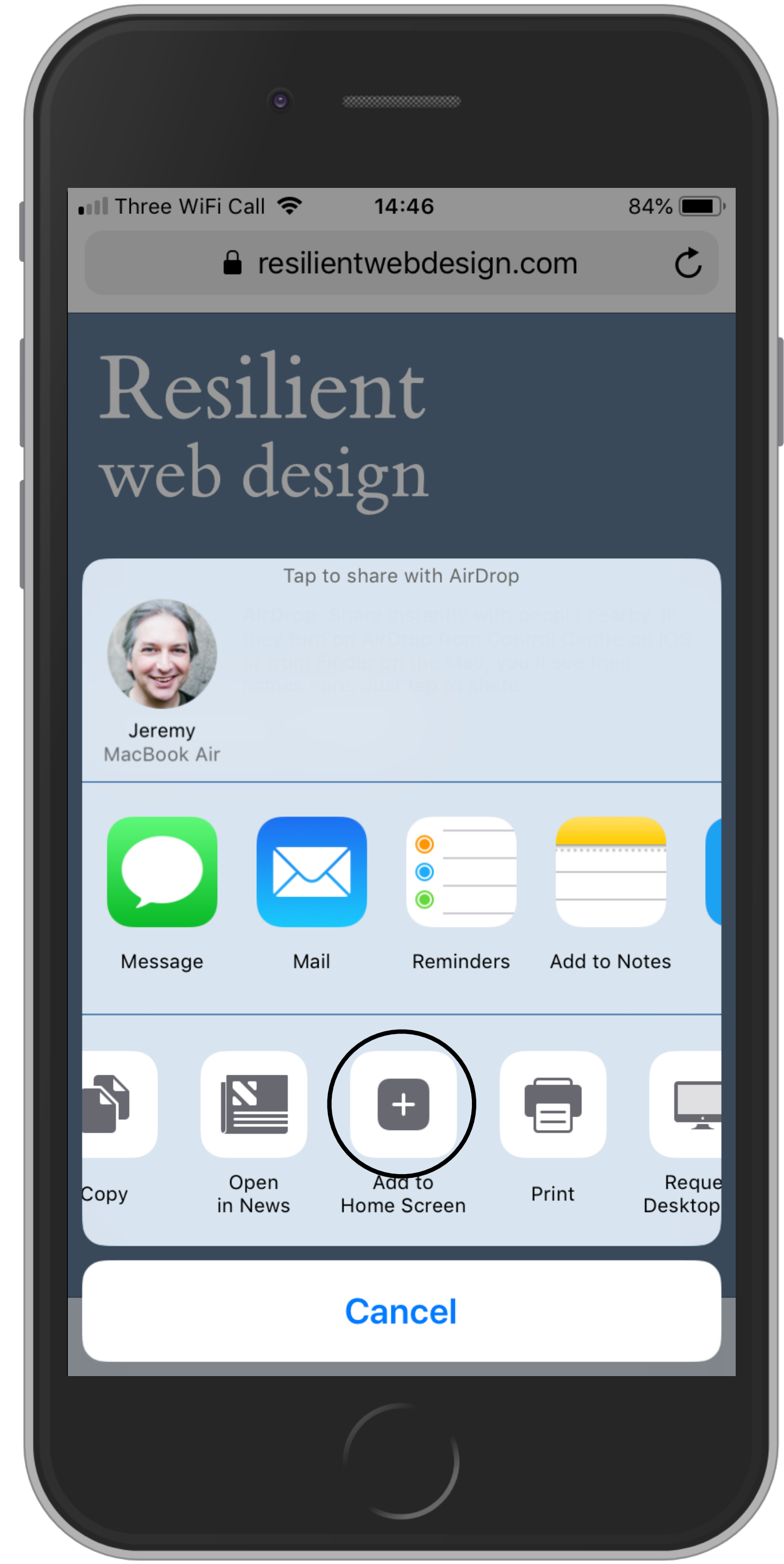
```
<link rel="manifest"  
      href="/manifest.json">
```

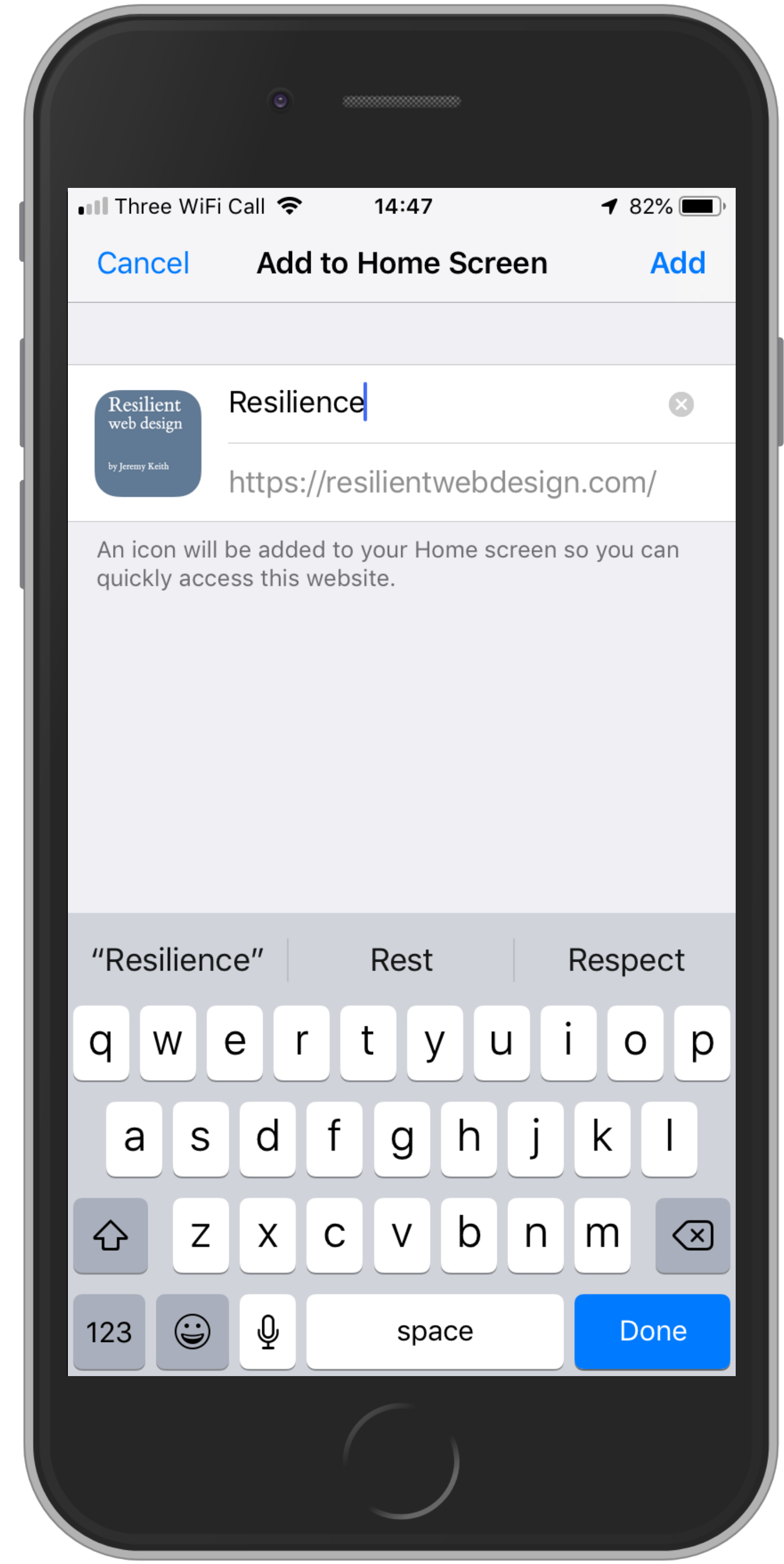
web app manifest

```
{  
  name: "Resilient Web Design",  
  short_name: "Resilience",  
  display: "standalone",  
  start_url: "/",  
  theme_color: "#5f7995",  
  icons: {  
    {  
      src: "/icon.png",  
      sizes: "512x512",  
      type: "image/png"  
    }  
  }  
}
```

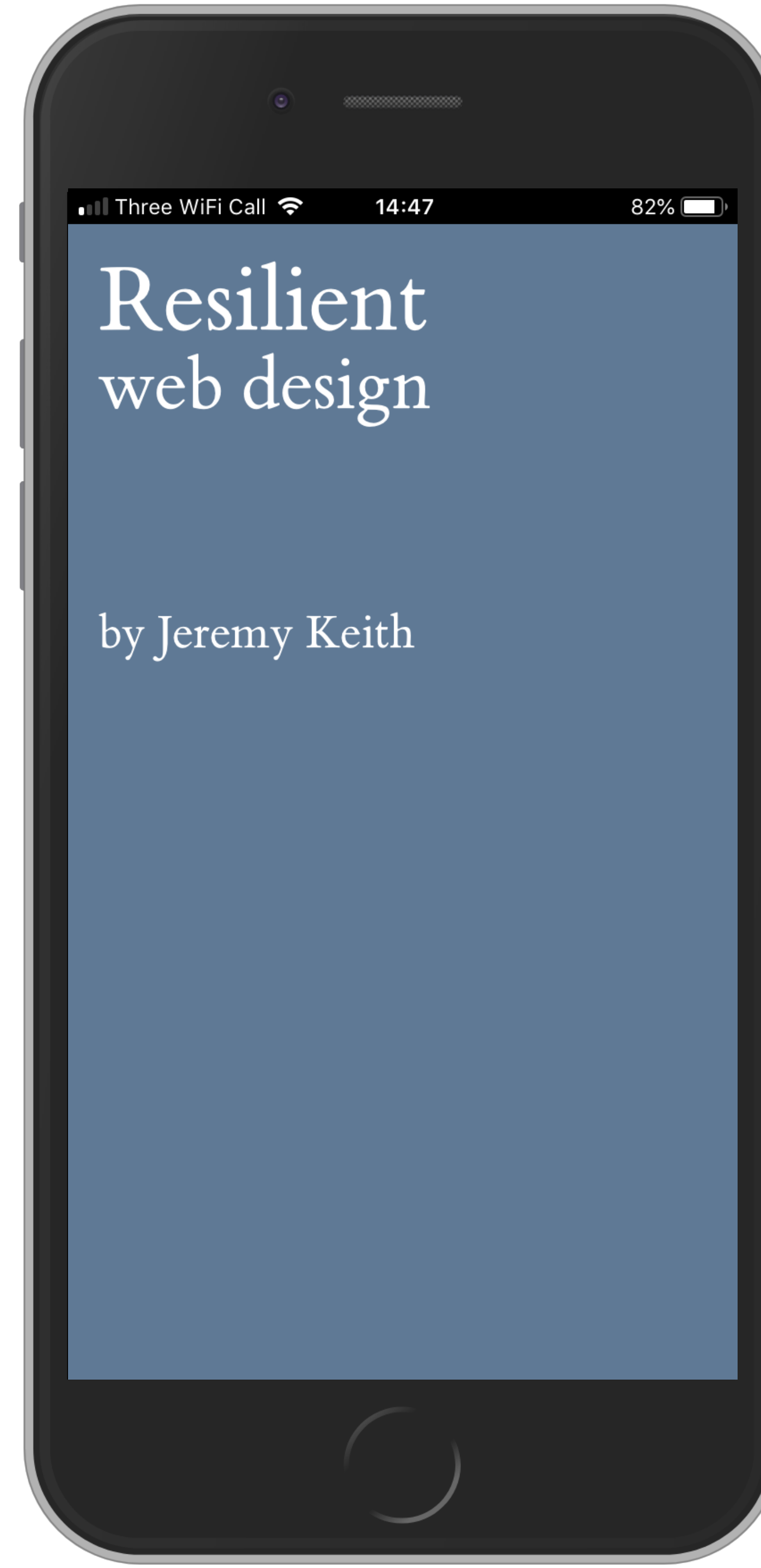



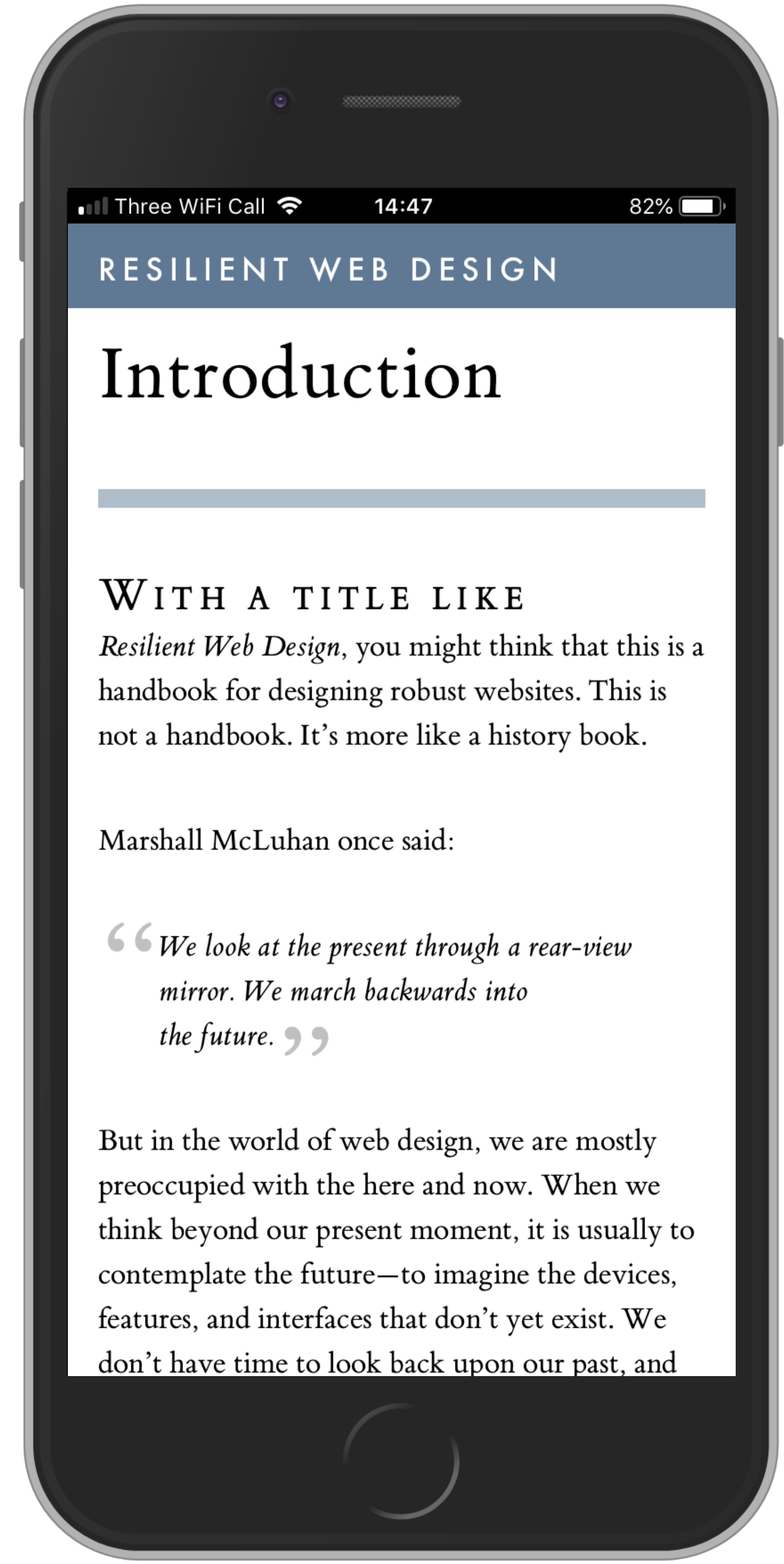




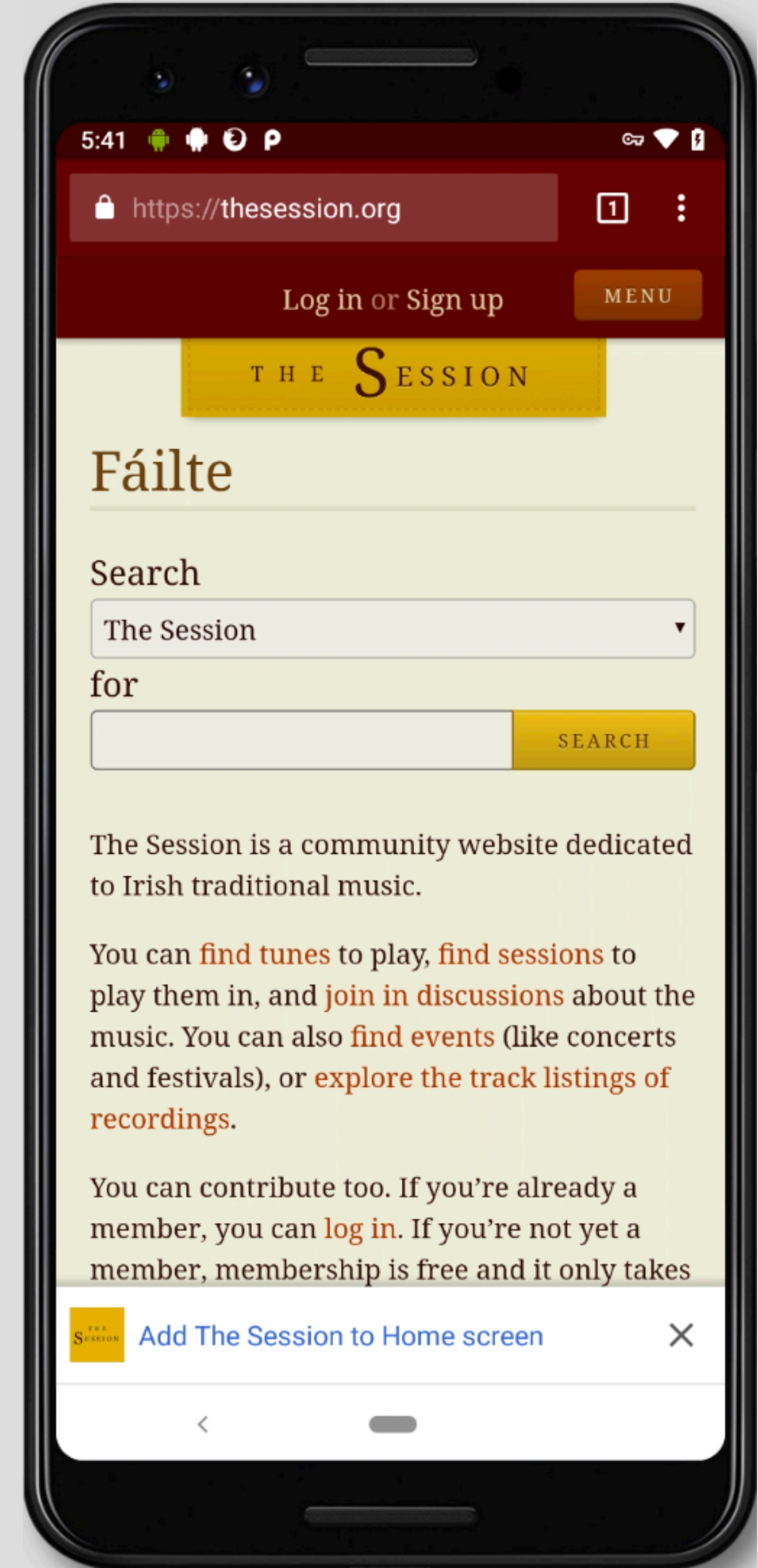


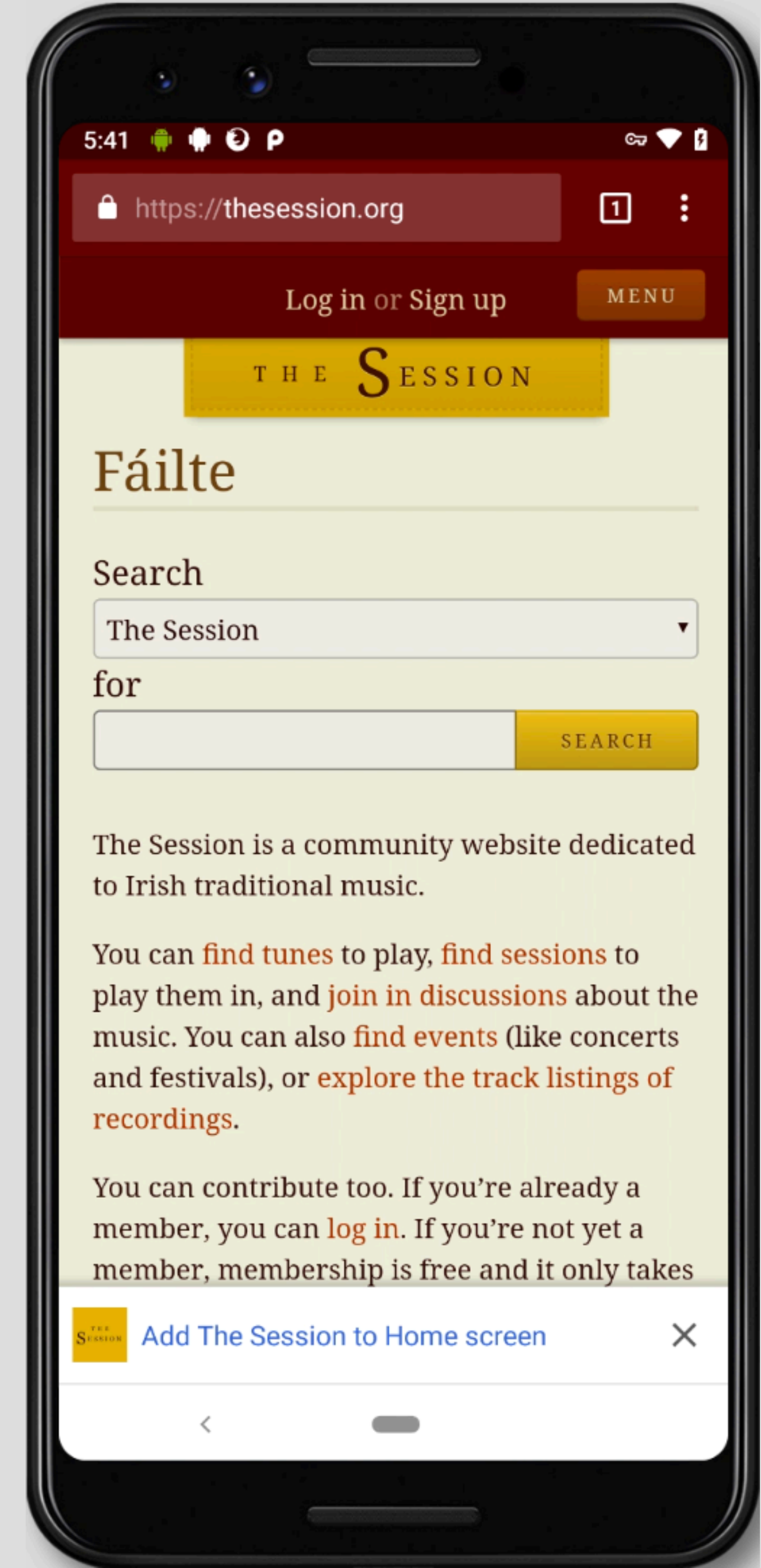












web app manifest

https

service worker

progressive web app

“I have a really hard time describing what a progressive web app actually is.”

—What the heck is a “Progressive Web App”? Seriously.

“A progressive web app is an enhanced version of a Single Page App (SPA) with a native app feel.”

—Building Progressive Web Apps

“Before You Build a PWA You Need a SPA”

—hackernoon.com

“I’ve seen too many devs assume PWA is a subset of SPA. We need to improve our messaging”

—Jake Archibald

“A Progressive Web App is a regular website following a progressive enhancement strategy to deliver native-like user experiences by using modern Web standards.”

—What is a PWA

“The name isn’t for you and worrying about it is distraction from just building things that work better for everyone. ... It’s marketing, just like HTML5 had very little to do with actual HTML.”

—Naming Progressive Web Apps

Site is served over HTTPS

All app URLs load while offline

Metadata provided for Add to Home screen

Pages are responsive

First load fast even on 3G

Site works cross-browser

Each page has a URL

Page transitions don't block on the network

Site is served over HTTPS

All app URLs load while offline

Metadata provided for Add to Home screen

Pages are responsive

First load fast even on 3G

Site works cross-browser

Each page has a URL

Page transitions don't block on the network

HTTPS

offline

Metadata

HTTPS

offline

Metadata

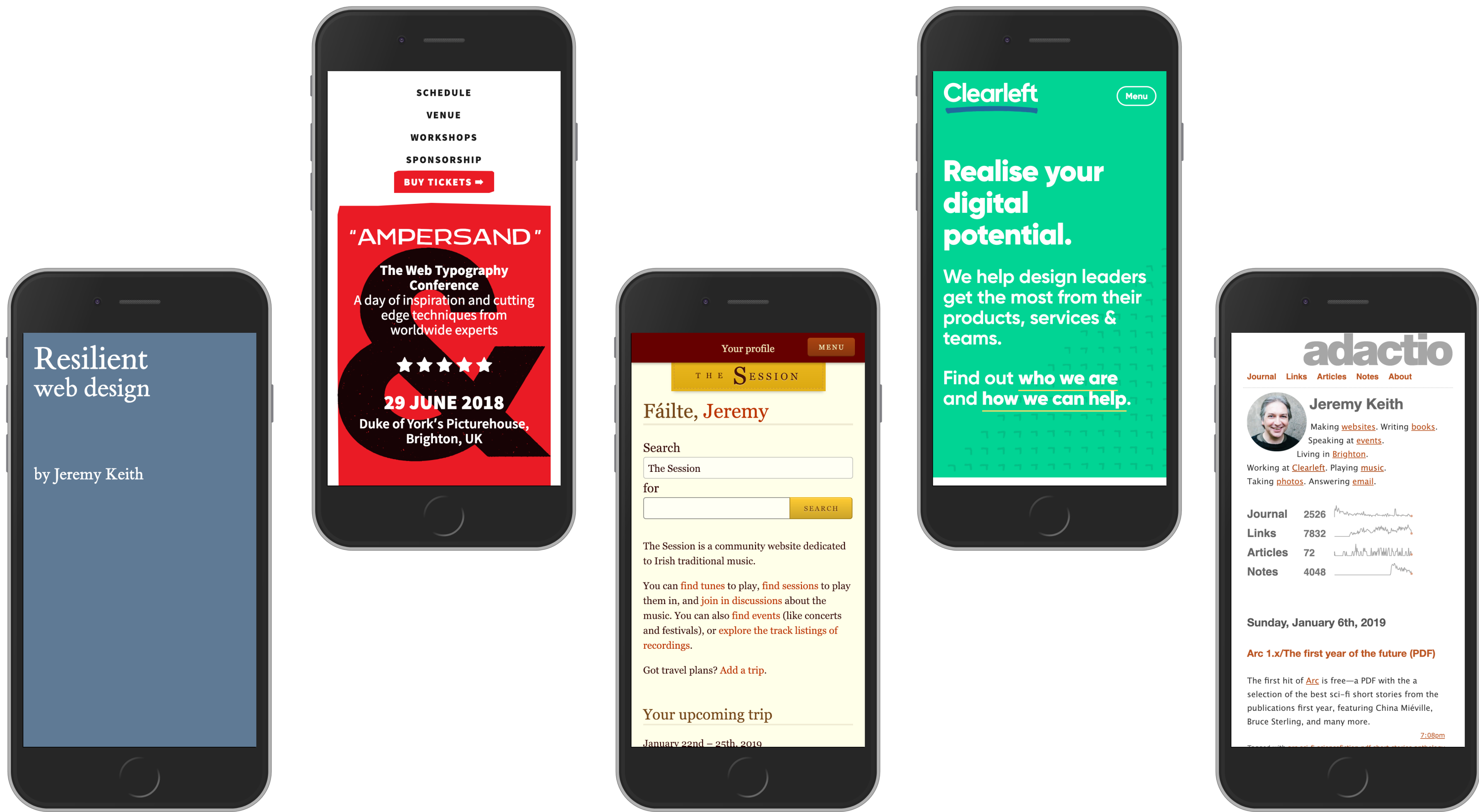
HTTPS

+ service worker

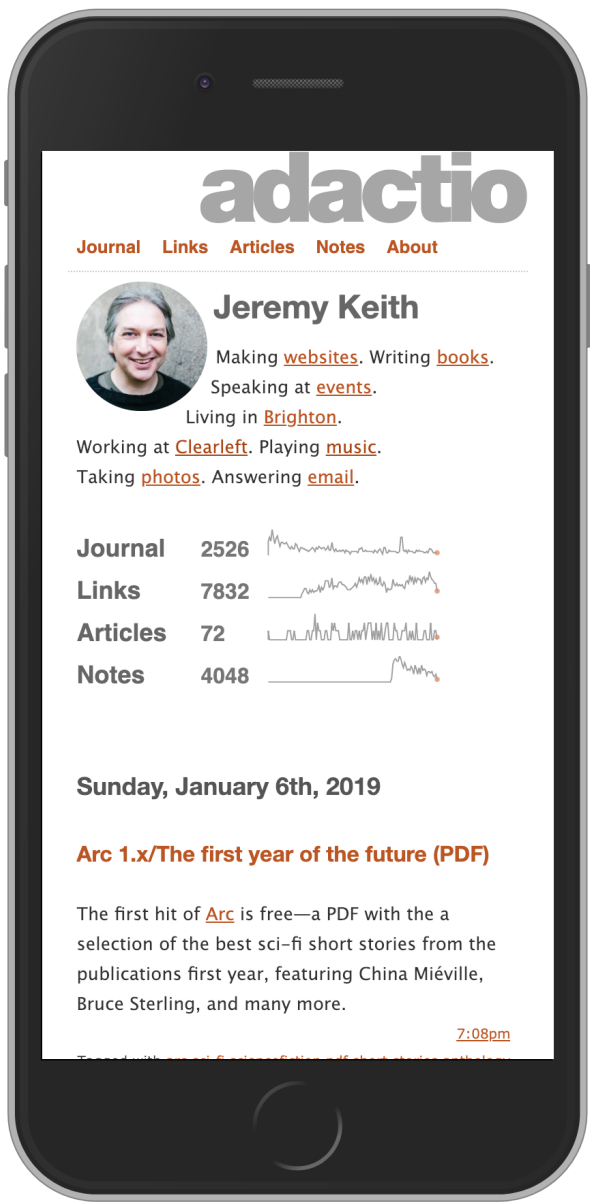
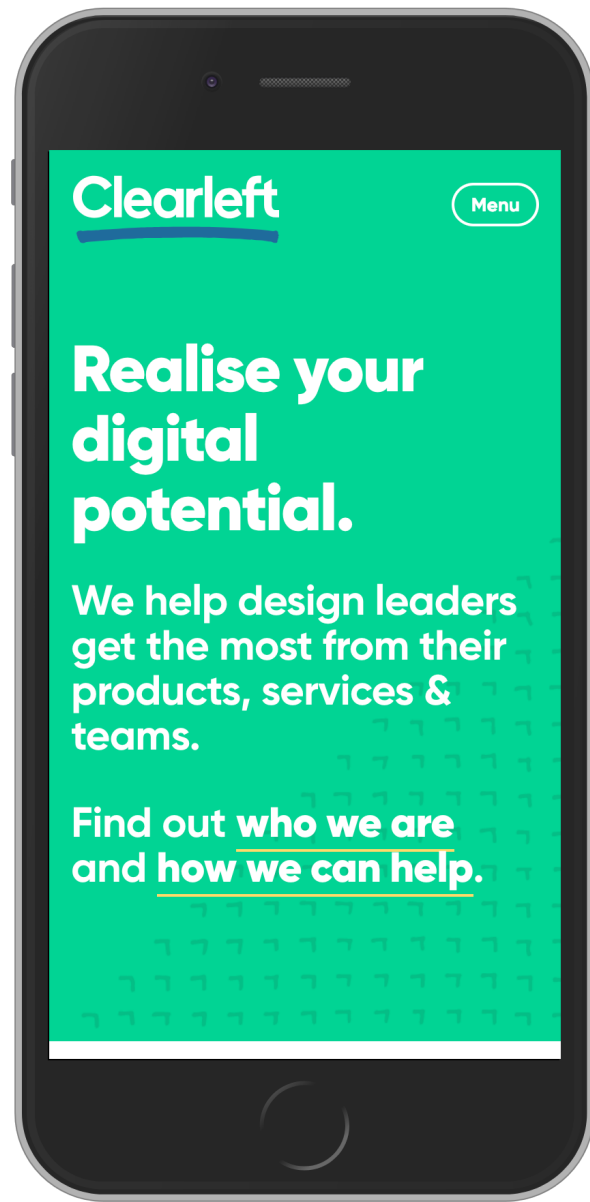
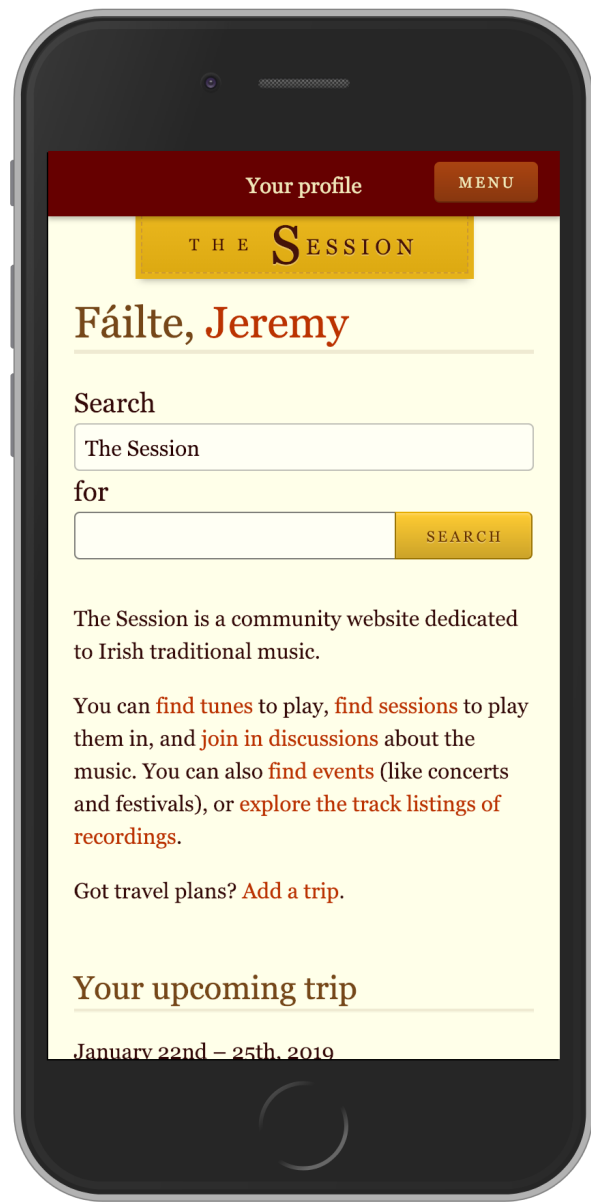
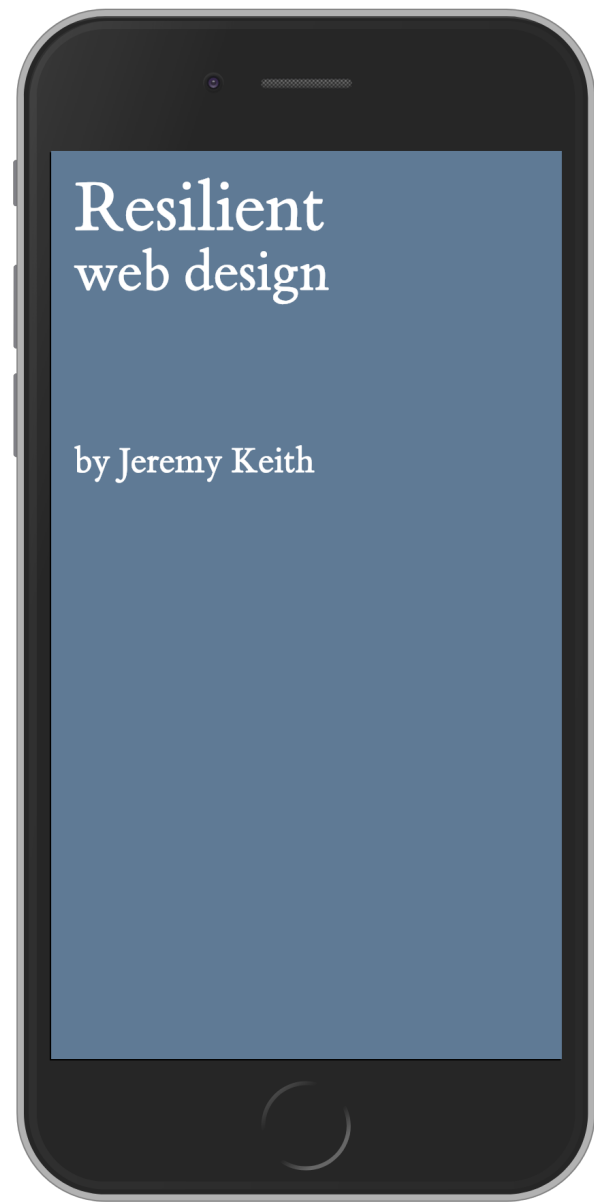
+ web app manifest

=

progressive web app



progressive



progressive

Thank you