# Progressive Web App

Gratis BoligSalg

# Demonstration of the app

https://walaito.github.io/grati 1

GRATISBOLIGSALG.DK

SEARCH...

search by filters

Bjerregårdsvej 36 C

6960 Hvide Sande

Føj GBS til startskærmen ✕

https://walaito.github.io/grati 1

GRATISBOLIGSALG.DK

SEARCH...

search by filters

**Føj til startskærm?**

Berør og hold på et ikon, eller tryk på Tilføj for at føje det til startskærmen.

GBS     1 × 1

Annuller               Tilføj

GBS
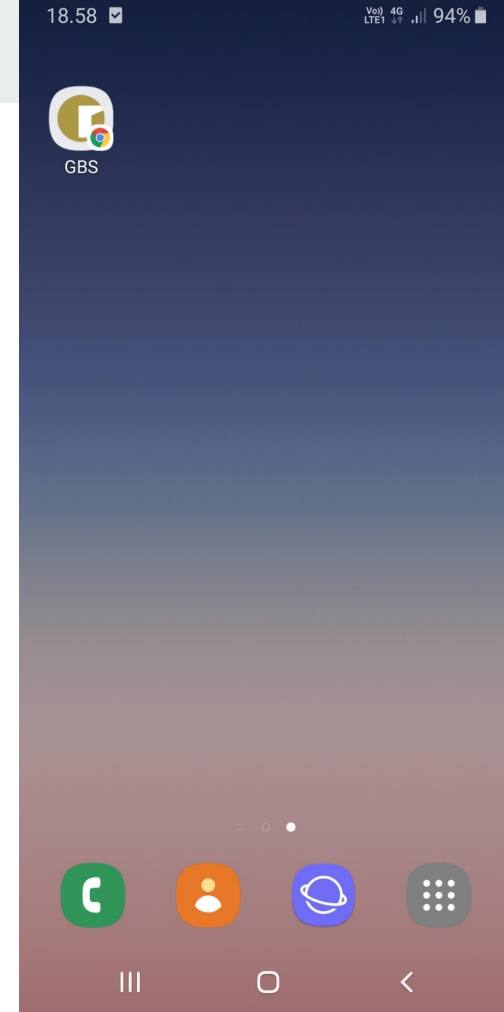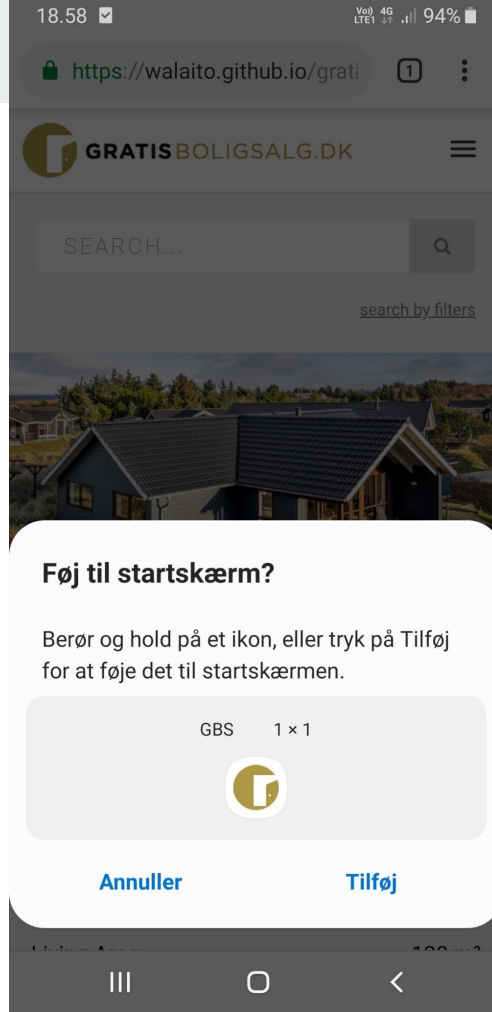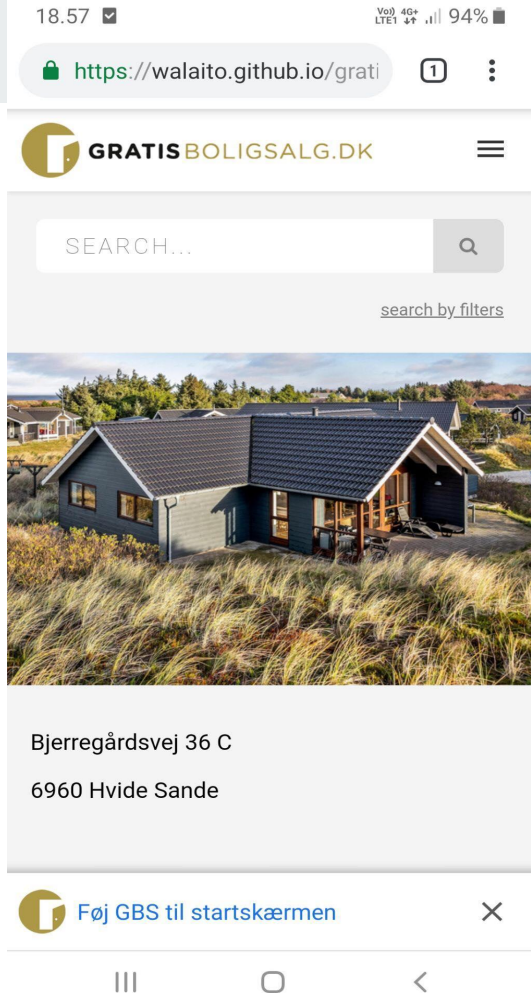
# What is Offline-first?

- Traditional web apps used to be entirely dependent on the server.
- Offline and low connectivity conditions are inevitable.
- **'Offline-first'** is a development approach that ensures an app will work as well offline as it does online or at least core features to function with or without an internet connection.
- **'Offline-first'** is a key characteristic of modern Progressive Web Applications.

# Caching Strategy

**Network, falling back to cache with frequent updates**

- This strategy always attempts to fetch the latest version from the network, falling back to the cached version only if the network request fails.
- Every time the network is accessed successfully, it updates the cache with the network response.
- Once user has visited a page, it will be stored in a cache.
- It will serve the latest version of a resource available. This is important for app/web which needs its content to be updated frequently.
- The downside is that we are passing on the chance to improve the load time of an HTML file that we may already have in the cache.

```javascript
//Call Fetch Event
self.addEventListener("fetch", e => {
  console.log("Service Worker: Fetching");
  e.respondWith(
    fetch(e.request)
      .then(res => {
        // Make a copy/clone of response
        const resClone = res.clone();
        //Open cache
        caches.open(cacheName).then(cache => {
          //Add response to cache
          cache.put(e.request, resClone);
        });
        return res;
      })
      .catch(err => caches.match(e.request).then(res => res))
  );
});
```

# Challenges

Javascript - implementing search functionality on the main page

```javascript
function search_listing() {
    let input = document.getElementById('js-search').value
    input = input.toLowerCase();
    let x = document.getElementsByClassName("wrapper");

    for (i = 0; i < x.length; i++) {
        if (!x[i].innerHTML.toLowerCase().includes(input)) {
            x[i].style.display = "none";
        } else {
            x[i].style.display = "wrap";
        }
    }
}
```

```javascript
//house information stored
var houses =[
    {price:2095000 , year:2018 , size:1284 , rooms:4},
    {price:2600000 , year:2018 , size:914 , rooms:6},
    {price:525000 , year:1990 , size:129 , rooms:3},
    {price:375000 , year:1990 , size:105 , rooms:3}
];
```

```javascript
//check if houses data is true of false
if (
    pmin <= houses[0].price && pmax >= houses[0].price &&
    ymin <= houses[0].year && ymax >= houses[0].year &&
    smin <= houses[0].size && smax >= houses[0].size &&
    rmin <= houses[0].rooms && rmax >= houses[0].rooms &&
    opt == "terracedhouse" ||
    pmin <= houses[0].price && pmax >= houses[0].price &&
    ymin <= houses[0].year && ymax >= houses[0].year &&
    smin <= houses[0].size && smax >= houses[0].size &&
    rmin <= houses[0].rooms && rmax >= houses[0].rooms && opt == ""
)
{
    document.getElementById("house1").style.display = "block";
}
else
{
    document.getElementById("house1").style.display = "none";
}
```

# Most proud of

- Planning of the project and splitting up the workload
- Finishing within the time schedule

# What we have learned

- Working as a team on a pwa while collaborating online

- Better understanding of using github for a group project with *git push / git pull*

# Thank you

😊