# Deployment Plan

## Reasons why Firebase Hosting is chosen

As we navigate through the initial stages of development, our current strategy involves deploying our website locally. Given our current usage of Firebase for the database and backend of our project, using Firebase Hosting is an easy way to only utilize one tool with our existing Firebase services.

Firebase Hosting offers a relatively simple development process, allowing for a smooth transition into hosting our application on the web. Furthermore, Firebase Hosting supports continuous deployment practices by using automatic updates deployment when new changes are pushed.

## Reasons why Capacitor is chosen

In parallel, we plan on transforming our project into a mobile app. We are considering Capacitor as the tool of choice. Capacitor is a great choice since it allows using the current web-based code and transforms it into a mobile app. Given that our project is initially developed as a website, this feature that Capacitor offers significantly reduces the need for rewriting the entire or parts of the application.

Furthermore, Capacitor allows for cross-platform development. It enables the creation of mobile apps for both iOS and Android.

## Other Options that were not chosen

For hosting, we thought about using the ENCS Server, although the setup is quite tedious and any new pushes to the code would need to be re-uploaded to the server which doesn't align with our needs.

Other options for hosting would be Google and Amazon services, but they are not free.

For mobile application development, we are not considering any of the tools in which projects have to be developed in a specific framework or IDE, like React or Android Studio since we are already set on using Angular from experience. Therefore, it would be very inconvenient to use a new framework or IDE that most of the team hasn't worked with.

There are not many ways to convert an existing website into an app, but there is AppMySite that we have found, although it's not compatible with Angular projects.

## Steps to deploy Using Firebase

**Initial Setup:**
1. Make sure firebase CLI is installed. Run the following command in the root directory of the project to install.
   With mac: sudo npm install -g firebase-tools
   With windows: npm install -g firebase-tools

2. Run the following command "firebase experiments: enable webframeworks".

**Steps:**
Step 1: Run the "ng build –configuration production" command in the project root folder in vs code terminal.

Step 2: Check the output from Step 1. If there are warnings about size limitations, go to angular.json and change the capacities of the budgets attributes depending on the results.

```
"configurations": {          You, 2 months ago • in
  "production": {
    "budgets": [
      {
        "type": "initial",
        "maximumWarning": "2mb",
        "maximumError": "5mb"
      },
      {
        "type": "anyComponentStyle",
        "maximumWarning": "2kb",
        "maximumError": "4kb"
      }
    ],
    "outputHashing": "all"
  },
```

In our case, we had to change the parameters:

```
},
        "configurations": {
            "production": {
                "budgets": [
                    {
                        "type": "initial",
                        "maximumWarning": "5mb",          Guillaume-Lachapelle, 1 ho
                        "maximumError": "8mb"
                    },
                    {
                        "type": "anyComponentStyle",
                        "maximumWarning": "5kb",
                        "maximumError": "8kb"
                    }
                ],
                "outputHashing": "all"
            },
```

Step 3: Update the firebase.json file. Change the value in public with the new 'dist' directory created from Step1. Also, add the rewrites parameter shown below.

```
Guillaume-Lachapelle, 1 hour ago | 2 authors (Guillaume-L
{
    "hosting": {
        "public": "dist\\decagon",          Gui
        "rewrites": [
            {
                "source": "**",
                "destination": "/index.html"
            }
        ]
    }
}
```

Step 4: Run the "firebase deploy" command in the project root folder in vs code terminal. This will provide you with the deployed url.

You can also access the url in the Firebase interface.

Before:

🔥 Firebase

🏠 Project Overview  ⚙️

Project shortcuts

▤ Realtime Database

👥 Authentication

🖼 Storage

What's new

🎟 Extensions  NEW

🚀 Release Monito...  NEW

Product categories

Build  ⌄

Release & Monitor  ⌄

Analytics  ⌄

Engage  ⌄

▦ All products

Customize your nav!

You can now focus your console experience by customizing your navigation

Learn more     Got it

Spark
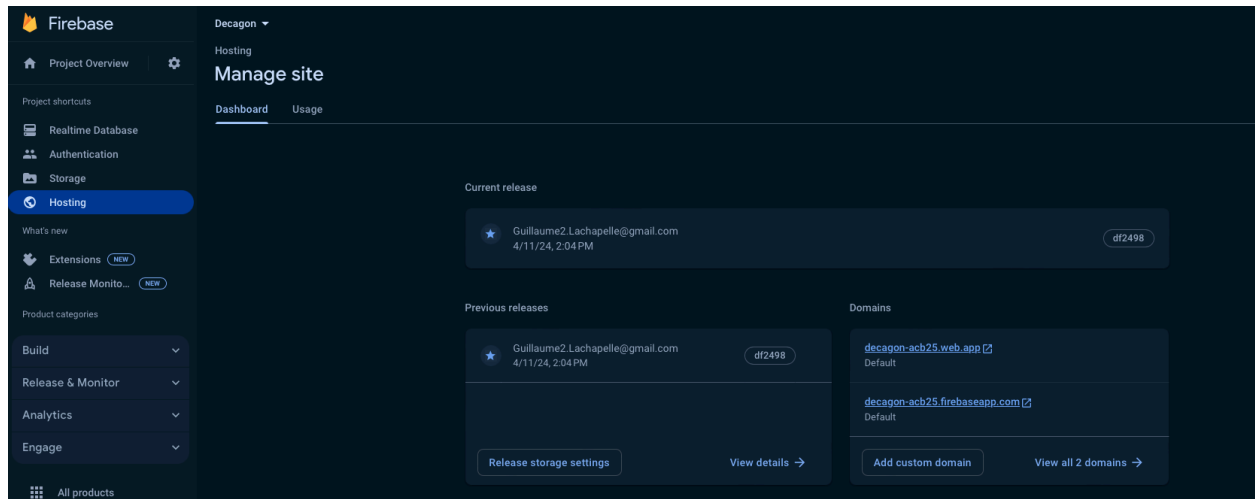Upgrade

Decagon ▼

# Project settings

General          Cloud Messaging          Integrations

After:



Now there is a current release with the provided domain.

Sprint 5: Deployment Plan

The domain provided in the Firebase interface includes all features implemented before April 11th 2024. Near the end of sprint 5, we will rerun the steps provided above to return the new domain containing the updated features implemented after April 11th 2024.

Useful Links:

https://www.youtube.com/watch?v=V2Wn2JROUEo

https://firebase.google.com/docs/hosting/use-cases#:~:text=Firebase%20Hosting%20is%20a%20fully,content%20is%20always%20delivered%20securely.

https://www.appmysite.com/blog/mobile-app-builder-benefits-developers/