Offline-first PWA with OPFS deployed to Github Pages

1. Set up Sveltekit project

Install Sveltekit

```
npx sv create YOUR_PROJECT_NAME
```

- Which template would you like?
  - SvelteKit minimal
- Add type checking with Typescript?
  - no
- What would you like to add to your project?
  - prettier
  - eslint
  - tailwincss
- tailwindcss: Which plugins would you like to add?
  - don't select any
- Which package manager do you want to install dependencies with?
  - npm

Update *tailwind.config.js*

```
/** @type {import('tailwindcss').Config} */
export default {
    content: ['./src/**/*.{html,js,svelte,ts}'],

    theme: {
        extend: {
            zIndex: {
        '100': '100',
        '1000': '1000',
        '2000': '2000',
        '3000': '3000',
        '5000': '5000',
        '10000': '10000',
        '20000': '20000',
      },
      scale: {
        '200': '2.00',
        '250': '2.50',
        '300': '3.00',
      }
        }
    },
    plugins: []
};
```

Modify *app.html*

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="theme-color" content="#FFFFFF">đ
        <link rel="icon" href="%sveltekit.assets%/favicon.png" />
        <title>YOUR_PROJECT_NAME</title>
        <meta
            name="viewport"
            content="width=device-width, initial-scale=1.0, maximum-scale=1.0, 
user-scalable=0, interactive-widget=resizes-visual"
        >
        %sveltekit.head%
    </head>
    <body data-sveltekit-preload-data="hover" class="overscroll-contain">
        <div style="display: contents">%sveltekit.body%</div>
    </body>
</html>
```

Add *navbar* and modify *virtual keyboard* setting in *+layout.svelte*

```
<script>
    import '../app.css';
    let { children } = $props();
    import { base } from '$app/paths';

  if ("virtualKeyboard" in navigator) {
    navigator.virtualKeyboard.overlaysContent = true;
  }

</script>

<div class="grid w-screen h-screen grid-rows-[auto_1fr] bg-slate-300">
    <nav class="flex items-center gap-4 p-2 m-0 text-xl font-bold bg-lime-400">
        <a href="{base}/">Home</a>
        <!--a href="{base}/map">Map</a-->
    </nav>
    {@render children()}
</div>
```

Create a *+layout.js* in *routes* folder

```
export const prerender = true;
export const ssr = false;
```

Open VsCode in *YOUR_PROJECT_NAME* folder: `code .`

---

2. Init Git

```
git init
git branch -M main
git add .
git commit -m "initial commit"
```

Run developer server

```
npm run dev -- --open
```

if *SvelteKitError: Not found: /favicon.ico* raised in terminal then

- create a *public* folder in *static folder*
- replace *favicon.png* intto *static/public/favicon.png*

---

3. Set up PWA

Create a manifest.json file and insert into *static* folder

```
{
    "id": "YOUR_PROJECT_NAME-pwa",
    "short_name": "YOUR_PROJECT_NAME",
    "start_url": "/YOUR_PROJECT_NAME/",
    "scope": "/YOUR_PROJECT_NAME/",
    "display": "standalone",
    "orientation": "portrait",
    "theme_color": "#A3E635",
    "background_color": "#ffffff",
    "dir": "ltr",
    "lang": "en",
    "icons": [
        {
            "src": "icons/icon192.png",
            "sizes": "192x192",
            "type": "image/png"
        },
        {
            "src": "icons/icon512.png",
            "sizes": "512x512",
            "type": "image/png"
        }
    ]
}
```

For local installation modify *manifest.json*:

- only slash needed for "start_url"
- delete "scope"

```
    "short_name": "YOUR_PROJECT_NAME",
    "start_url": "/",
    "display": "standalone",
    ...
```

Insert *manifest.json* link into *app.html*

```
<meta name="theme-color" content="#FFFFFF">
<link rel="manifest" href="%sveltekit.assets%/manifest.json" />
<link rel="icon" href="%sveltekit.assets%/favicon.png" />
```

Insert *icons* folder with icons into *static* folder

Copy and insert service-worker.js into *scr* folder

Commit changes

---

4. Static site generation for Github Pages

Install adapter-static

`npm install -D @sveltejs/adapter-static`

Modify *svelte.config.js*

```
import adapter from '@sveltejs/adapter-static';

/** @type {import('@sveltejs/kit').Config} */
const config = {
    kit: {
        adapter: adapter({
            fallback: '404.html'
        }),
        paths: {
            base: process.argv.includes('dev') ? '' : process.env.BASE_PATH
        }
    }
};

export default config;
```

Create the *.github/workflows* folder in *YOUR_PROJECT_NAME* folder

Copy and insert deploy.yml into *YOUR_PROJECT_NAME/.github/workflows*

Create a new repo in Github

- name : *YOUR_PROJECT_NAME*
- *YOUR_PROJECT_NAME/settings/pages*: set *Source* to ***Github Actions***

Create remote repo and upload to Github

```
git remote add origin git@github.com:YOUR_NAME/YOUR_PROJECT_NAME.git
git add .
git commit -m "Github Pages set up"
npm run build
git push -u origin main
```

---

5. Enable installation from locale computer

Install vite-plugin-mkcert

`npm install vite-plugin-mkcert`

Trust the Local Certificate

To avoid the "unsafe site" warning entirely, you can manually trust the certificate generated by vite-plugin-mkcert.
Locate the certificate generated by mkcert:

- By default, mkcert uses the system's trusted CA.
- On Windows, certificates are stored in the *C:/Users/USER_NAME/.vite-plugin-mkcert* directory.
- Install and trust the certificate on your Android device:
  - Copy the CA certificate file (usually named *rootCA.pem*) to your device.
  - Install it via Settings → find CA-certificate → Install from storage.
  - Trust the certificate for your browser.

Add --host to *package.json* script part

```
"scripts": {
    "dev": "vite --host",
    "build": "vite build",
    "preview": "vite preview --host",
    ...
},
```

Modify *vite.config.js*

```
import { sveltekit } from '@sveltejs/kit/vite';
import { defineConfig } from 'vite';
import mkcert from 'vite-plugin-mkcert';

export default defineConfig({
  server: {
    https: true,
    proxy: {},
  },
  plugins: [
    sveltekit(),
    mkcert(),
  ]
});
```

Commit changes

---

6. SQLocal and OPFS

Install SQLocal

```
npm install sqlocal
```

Modify *vite.config.js*

```
import { sveltekit } from '@sveltejs/kit/vite';
import { defineConfig } from 'vite';
import mkcert from 'vite-plugin-mkcert';

export default defineConfig({
  server: {
    https: true,
    proxy: {},
  },
  plugins: [
    sveltekit(),
    mkcert(),
    {
      name: 'configure-response-headers',
      configureServer: (server) => {
        server.middlewares.use((_req, res, next) => {
          res.setHeader('Cross-Origin-Embedder-Policy', 'require-corp');
          res.setHeader('Cross-Origin-Opener-Policy', 'same-origin');
          next();
        });
      },
    },
  ],
  optimizeDeps: {
    exclude: ['sqlocal'],
  },
});
```

Insert cross-origin headers into the end of *service-worker.js*

```
if (response.status === 200) {
    cache.put(event.request, response.clone());
}

/* Cross-origin isolation headers start*/
const newHeaders = new Headers(response.headers);
newHeaders.set("Cross-Origin-Embedder-Policy", "require-corp");
newHeaders.set("Cross-Origin-Opener-Policy", "same-origin");
const moddedResponse = new Response(response.body, {
    status: response.status,
    statusText: response.statusText,
    headers: newHeaders,
});
return moddedResponse;
/* Cross-origin isolation headers end*/
return response;  //Delete this line from original file
```

Insert *crossOrigin : true* into every *Tilelayer*

```
<TileLayer
    name={'OSM'}
    url={'https://tile.openstreetmap.org/{z}/{x}/{y}.png'}
    options={{
        minZoom: 7,
        maxZoom: 19,
        attribution: '&copy; OpenstreetMap',
        crossOrigin : true
    }}
    selected
/>
```

Commit changes

---

7. Create a map with Leaflet

Install Leaflet

`npm install leaflet`

Install leaflet markercluster

`npm install leaflet.markercluster`

Install Leaflet markercluster layersupport

`npm install leaflet.markercluster.layersupport --save`

Install necessary Turf.js packages

`npm install @turf/bearing @turf/destination @turf/distance` `npm install @turf/midpoint @turf/point-on-feature` `npm install @turf/nearest-point-on-line @turf/explode @turf/helpers`

Insert *map* folder into *routes*

Create and insert +*page.svelte* into *map* folder

Create and insert +*page.js* into *map* folder

`export const prerender = false;`

7. Calendar and Time picker

Install Svelty Picker

- `npm install svelty-picker`
