The error `net::ERR_CERT_DATE_INVALID` means the SSL certificate you're using is either expired, not yet valid, or otherwise invalid for `localhost`. Here's how to fix it, along with explanations and best practices:

1. Generate a Valid Certificate for `localhost` (Recommended)

The most likely issue is that your certificate isn't correctly configured for `localhost`. Browsers are very strict about certificates for local development. You shouldn't be using a production certificate for local development.

Here are the best ways to generate a suitable certificate:

- Using `mkcert` (Easiest): `mkcert` is a tool specifically designed to generate locally trusted development certificates. It's highly recommended.
    1. Install: If you don't have it, install `mkcert`. Instructions are on their GitHub page (https://github.com/FiloSottile/mkcert). It's very simple: `brew install mkcert` (macOS), `choco install mkcert` (Windows), or other methods for Linux.
    2. Generate: Run `mkcert -install` once to configure your system's trust store. Then, run `mkcert localhost` to create `localhost.pem` and `localhost-key.pem`.
    3. Use in Vite Config:

       JavaScript
       ```javascript
       server: {
           port: 44343,
           https: {
               cert: './localhost.pem', // Or wherever mkcert placed it
               key: './localhost-key.pem', // Or wherever mkcert placed it
           },
           watch: {
               usePolling: true
           }
       },
       ```
- Using `openssl` (More involved): You can generate a self-signed certificate with `openssl`, but you'll likely have to manually add it to your browser's trusted root

certificates. This is more complex and not as user-friendly. If you *must* use `openssl`:

Bash

```bash
openssl req -x509 -newkey rsa:2048 -nodes -keyout localhost-key.pem
-out localhost.pem -days 365 -subj "/C=US/ST=CA/L=Mountain View/O=My
Company/CN=localhost"
```

You'll need to research how to add the `localhost.pem` (the certificate) to your browser's trusted root authorities. This process varies by browser and is a hassle.

2. Why `basicSsl` is Not Recommended

The `basicSsl` plugin is primarily for *very* basic HTTPS setup. It generates a self-signed certificate on the fly, which browsers will almost always reject with a warning like you're seeing. It's not suitable for serious development, especially with `localhost`. Remove `basicSsl` from your `vite.config.js`:

JavaScript

```javascript
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react-swc'
import path from 'path';

// ... (rest of your config)

plugins: [react()], // Remove basicSsl

// ...
```

3. Why `usePolling: true`?

The `watch: { usePolling: true }` option is a workaround for file system watching issues on some systems (especially Windows or WSL). It's not related to the certificate problem. Keep it if you need it.

4. Port 44343

Port 44343 is a non-standard HTTPS port. While it's *possible* to use it, port 443 is the standard HTTPS port. If you're not specifically required to use 44343, consider switching to 443. However, this won't fix the certificate issue.

5. Complete Example (using `mkcert`)

JavaScript

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react-swc'
import path from 'path';

export default defineConfig({
    plugins: [react()],
    resolve: {
        alias: {
            '@/': `${path.resolve(__dirname, 'src')}/`,
            '@helpers': path.resolve(__dirname,
'./src/_metronic/helpers'),
        },
    },
    build: {
        chunkSizeWarningLimit: 3000,
        rollupOptions: {
            external: []
        },
    },
    server: {
        port: 44343, // Or 443 if you prefer
        https: {
            cert: './localhost.pem', // Path from mkcert
            key: './localhost-key.pem', // Path from mkcert
        },
        watch: {
            usePolling: true
        }
    },
})
```

Steps:

1. Install `mkcert`: Follow the instructions on the `mkcert` GitHub page.
2. Generate certificates: Run `mkcert -install` and then `mkcert localhost`.
3. Update `vite.config.js`: Use the example above, replacing the paths to your certificate and key files if necessary.
4. Restart Vite: Restart your development server.

This approach using `mkcert` is the most reliable and easiest way to resolve the `net::ERR_CERT_DATE_INVALID` error for local development. It avoids browser warnings and ensures a secure local development environment.