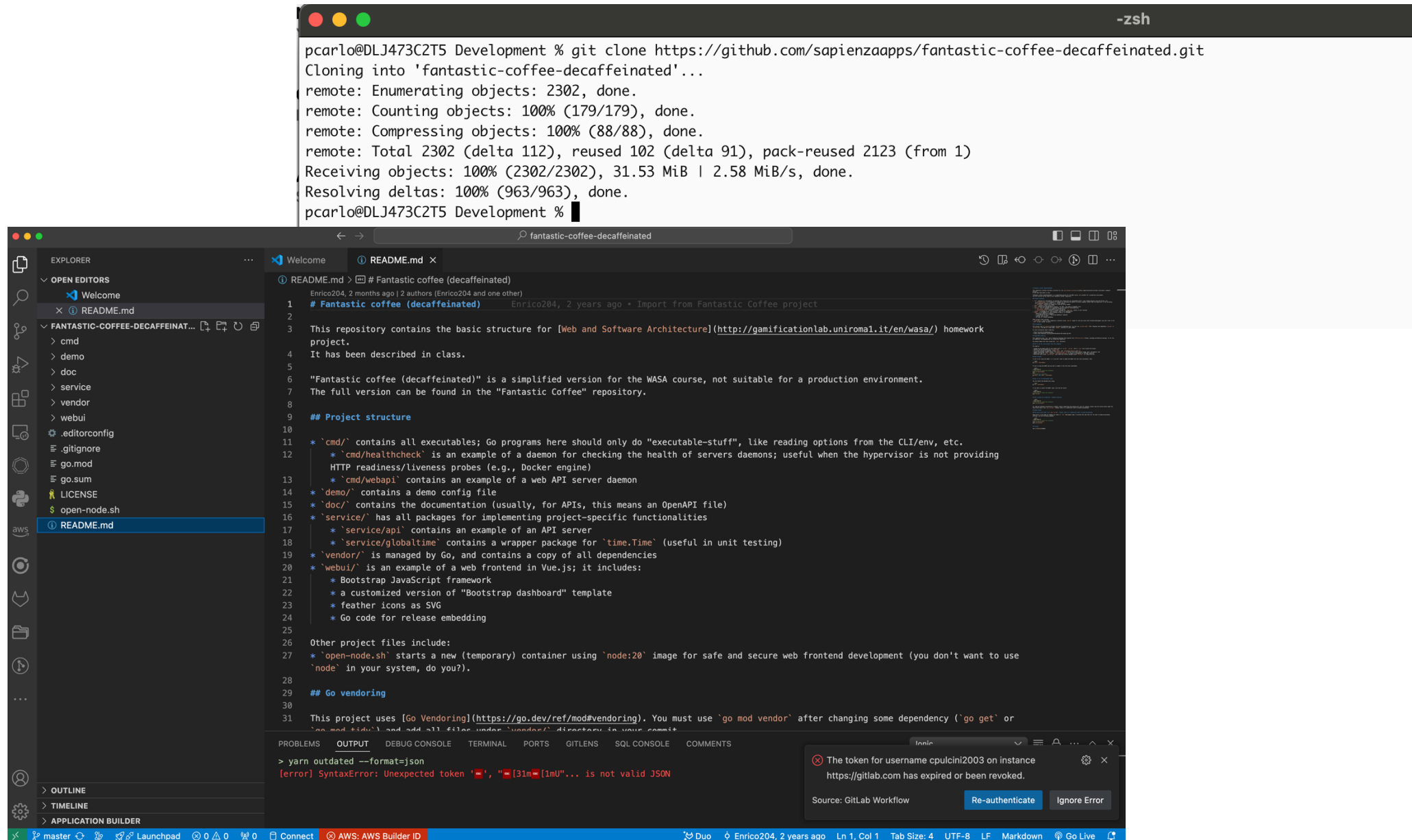
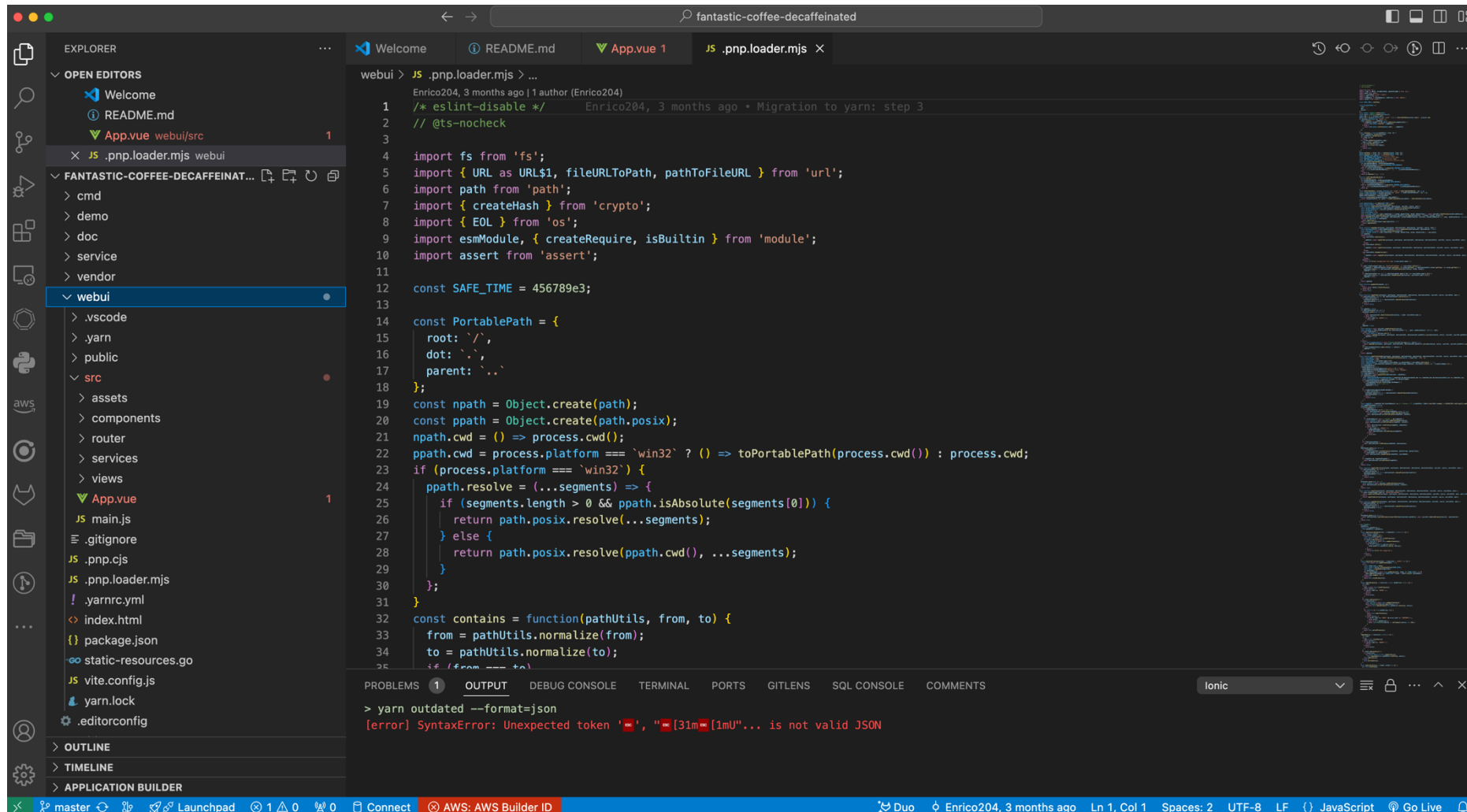


Setup front-end app a partire da template



Setup front-end app a partire da template

l'app e contenuta nella cartella webui



avvio dell'app

dal README.md si legge

```
If you want to launch the WebUI, open a new tab and launch:
```

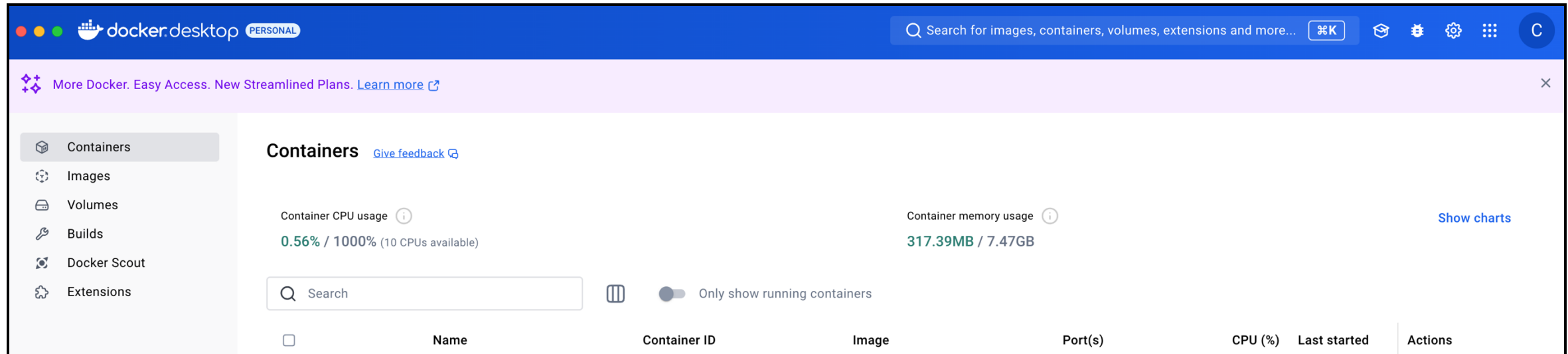
```
```shell
./open-node.sh
(here you're inside the container)
yarn run dev
```
```

```
docker run -it --rm -v "$(pwd):/src" -u "$(id -u):$(id -g)" --network host --workdir /src/webui node:20 /bin/bash
```

lo script prevede l'utilizzo di docker per avviare l'app
in DEV mode, ovvero mentre si sviluppa

avvio dell'app

requisiti: é necessario installare DOCKER DESKTOP e disporre della CLI docker

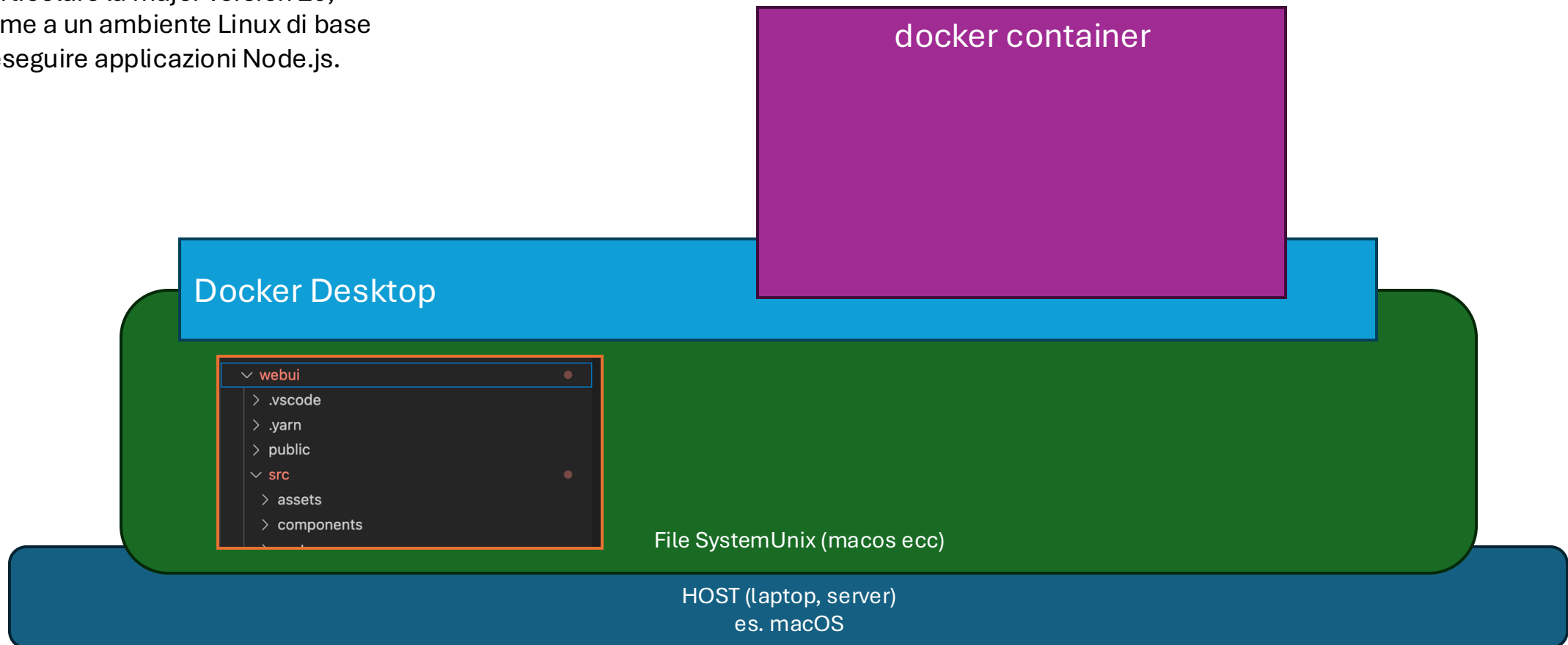


```
pcarlo@DLJ473C2T5 fantastic-coffee-decaffeinated % docker --version  
Docker version 27.3.1, build ce12230  
pcarlo@DLJ473C2T5 fantastic-coffee-decaffeinated %
```

SEMPLIFICHIAMO IL COMANDO

```
docker run -it --rm node:20 /bin/bash
```

crea il container VIOLA utilizzando l'immagine node:20 che contiene una versione preconfigurata di Node.js, in particolare la major version 20, insieme a un ambiente Linux di base per eseguire applicazioni Node.js.

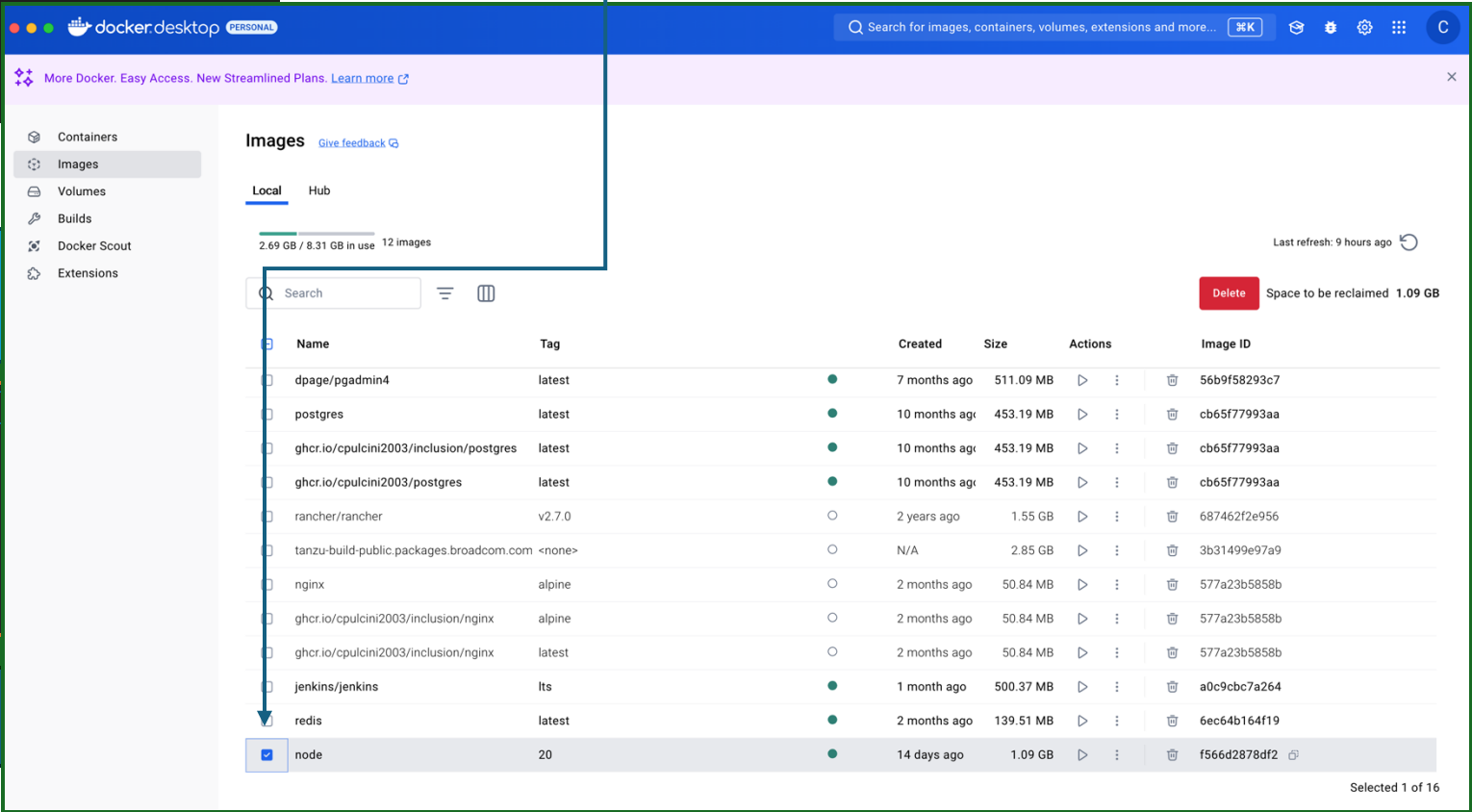


SEMPLIFICHIAMO IL COMANDO

```
docker run -it --rm node:20 /bin/bash
```

la prima volta viene scaricata l'immagine da registry docker e viene salvata nelle immagini del docker locale

```
pcarlo@DLJ473C2T5 fantastic-coffee-decaffeinated % docker run -it --rm node:20 /bin/bash
Unable to find image 'node:20' locally
20: Pulling from library/node
82312fccb35f: Pull complete
4ac722d9cf93: Pull complete
261351ed796d: Pull complete
a9d319298afc: Downloading [=====>] 27.85MB/202.7MB
e566ebd8ca22: Download complete
b70f3a35e4ad: Waiting
4ff8c378ac46: Downloading [=====>]
f0a653e5ccbd: Waiting
```

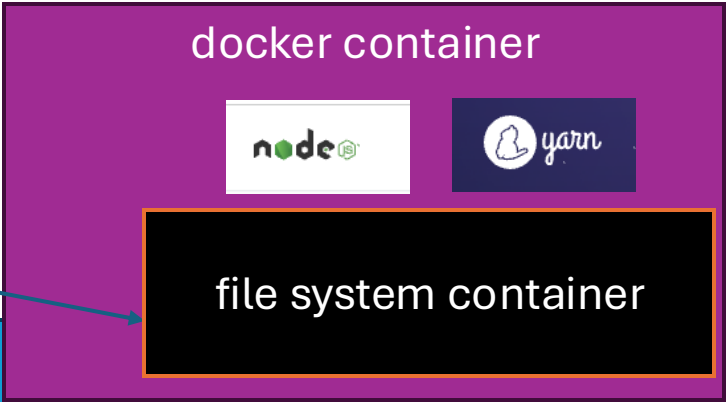


SEMPLIFICHIAMO IL COMANDO

```
docker run -it --rm node:20 /bin/bash
```

al termine del download viene creato il container (VIOLA che contiene i tool necessari per far girare l'APP

```
pcarlo@DLJ473C2T5 fantastic-coffee-decaffeinated % docker run -it --rm node:20 /bin/bash
Unable to find image 'node:20' locally
20: Pulling from library/node
82312fccb35f: Pull complete
4ac722d9cf93: Pull complete
261351ed796d: Pull complete
a9d319298afc: Pull complete
e566ebd8ca22: Pull complete
b70f3a35e4ad: Pull complete
4ff8c378ac46: Pull complete
f0a653e5ccbd: Pull complete
Digest: sha256:f4755c9039bdeec5c736b2e0dd5b47700d6393b65688b9e9f807ec12f54a8690
Status: downloaded newer image for node:20
root@58bb1f4380f7: /# df -k
Filesystem      1K-blocks    Used Available Use% Mounted on
overlay         61202244 12744916  45316004  22% /
tmpfs           65536      0    65536      0% /dev
shm            65536      0    65536      0% /dev/shm
/dev/vda1       61202244 12744916  45316004  22% /etc/hosts
tmpfs          4012680      0   4012680      0% /proc/scsi
tmpfs          4012680      0   4012680      0% /sys/firmware
root@58bb1f4380f7: /#
```



Docker Desktop

```
webui
├── .vscode
├── .yarn
├── public
├── src
│   ├── assets
│   └── components
```

File SystemUnix (macos ecc)

HOST (laptop, server)
es. macOS

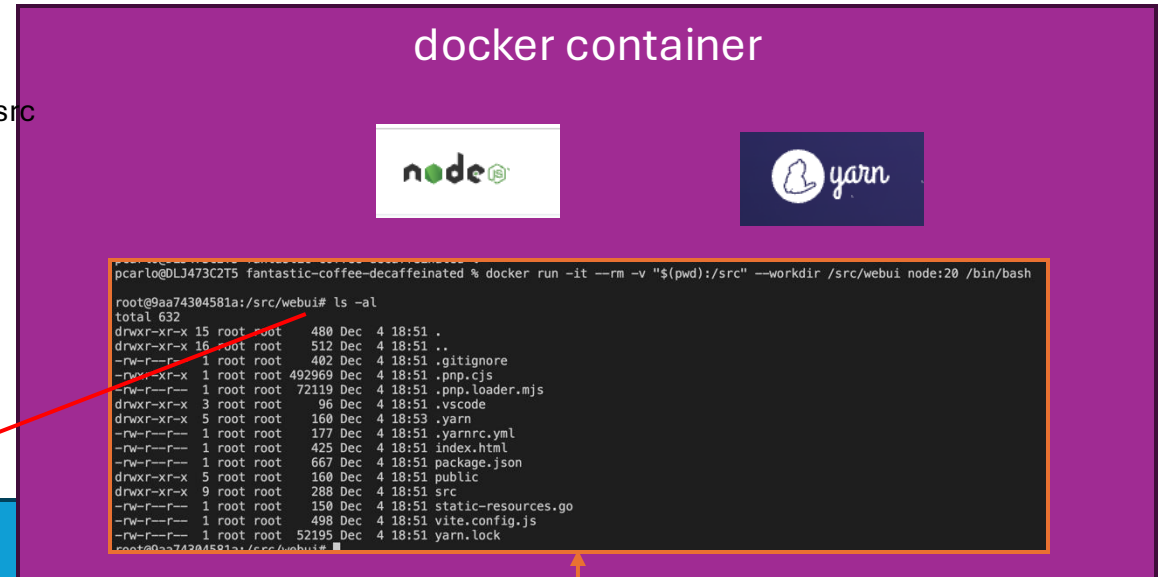
SEMPLIFICHIAMO IL COMANDO

```
docker run -it --rm -v "$(pwd) :/src" --workdir /src/webui node:20 /bin/bash
```

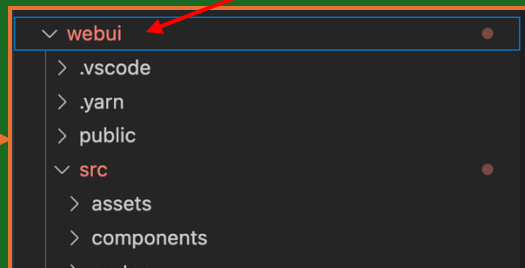
se facciamo girare il comando con gli altri parametri:

-v "\$(pwd):/src": Monta la directory corrente del sistema host (\$(pwd)) nella directory /src del container, consentendo di accedere ai file locali dal container. Questo è utile per lavorare su file locali senza doverli copiare nel container.

--workdir /src/webui: Imposta la directory di lavoro all'interno del container su /src/webui.



Docker Desktop



File System Unix (macos ecc)

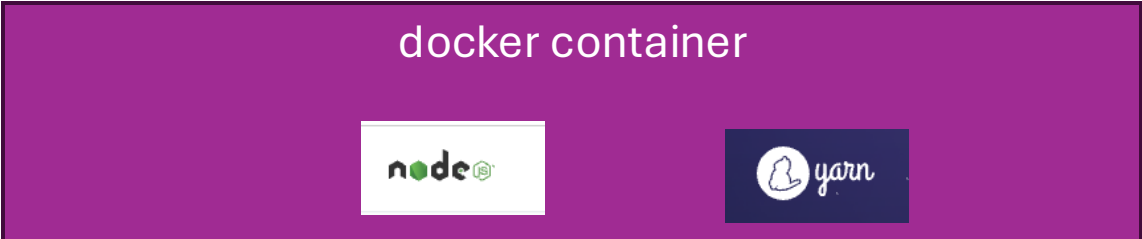
HOST (laptop, server)
es. macOS

SEMPLIFICHIAMO IL COMANDO

```
docker run -it --rm -v "$(pwd) :/src" -u "$(id -u):$(id -g)" --network host --workdir /src/webui node:20 /bin/bash
```

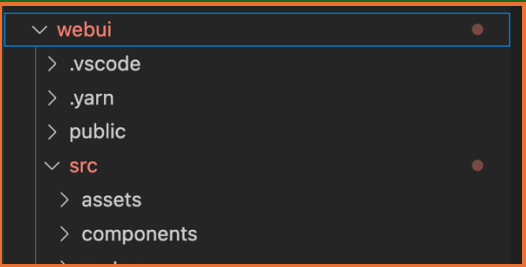
-u "\$ (id -u):\$ (id -g)": Esegue i comandi all'interno del container utilizzando l'ID utente (id -u) e l'ID gruppo (id -g) del sistema host. Questo evita problemi di permessi sui file condivisi tra host e container.

notare che ora é presente l'utente 503 anziche root



```
pcarlo@DLJ473C2T5 fantastic-coffee-decaffeinated % docker run -it --rm -v "$(pwd) :/src" -u "$(id -u):$(id -g)" --network host --workdir /src/webui node:20 /bin/bash
I have no name!@docker-desktop:/src/webui$ ls -al
total 632
drwxr-xr-x 15 root root    480 Dec  4 18:51 .
drwxr-xr-x 16 root root    512 Dec  4 18:51 ..
-rw-r--r--  1 root root    402 Dec  4 18:51 .gitignore
-rwxr-xr-x  1 root root 492969 Dec  4 18:51 .pnpm.cjs
-rw-r--r--  1 root root 72119 Dec  4 18:51 .pnpm.loader.mjs
drwxr-xr-x  3 502 dialout   96 Dec  4 18:51 .vscode
drwxr-xr-x  5 502 dialout  160 Dec  4 18:53 .yarn
-rw-r--r--  1 root root   177 Dec  4 18:51 yarnrc.yml
-rw-r--r--  1 root root   425 Dec  4 18:51 index.html
-rw-r--r--  1 root root   667 Dec  4 18:51 package.json
drwxr-xr-x  5 502 dialout  160 Dec  4 18:51 public
drwxr-xr-x  9 502 dialout  288 Dec  4 18:51 src
-rw-r--r--  1 root root   150 Dec  4 18:51 static-resources.go
-rw-r--r--  1 root root   498 Dec  4 18:51 vite.config.js
-rw-r--r--  1 root root 52195 Dec  4 18:51 yarn.lock
I have no name!@docker-desktop:/src/webui$
```

Docker Desktop



File SystemUnix (macos ecc)

HOST (laptop, server)
es. macOS

AVVIAMO L'APP COME DA README.md

If you want to launch the WebUI, open a new tab and launch:

```
```shell
./open-node.sh
(here you're inside the container)
yarn run dev
```
```

non funziona
c'è qualcosa che non va nella directory .yarn
della nostra app. E' la directory contenuta
nell'immagine che abbiamo scaricato all'inizio
da GITHUB

```
drwxr-xr-x 16 root root    512 Dec  4 18:51 ..
-rw-r--r--  1 root root    402 Dec  4 18:51 .gitignore
-rwxr-xr-x  1 root root 492969 Dec  4 18:51 .pnp.cjs
-rw-r--r--  1 root root 721119 Dec  4 18:51 .pnp.loader.mjs
drwxr-xr-x  3 502 dialout   96 Dec  4 18:51 .vscode
drwxr-xr-x  5 502 dialout  160 Dec  4 18:53 .yarn
-rw-r--r--  1 root root   177 Dec  4 18:51 .yarnrc.yml
-rw-r--r--  1 root root   425 Dec  4 18:51 index.html
-rw-r--r--  1 root root   667 Dec  4 18:51 package.json
drwxr-xr-x  5 502 dialout  160 Dec  4 18:51 public
drwxr-xr-x  9 502 dialout  288 Dec  4 18:51 src
-rw-r--r--  1 root root   150 Dec  4 18:51 static-resources.go
-rw-r--r--  1 root root   498 Dec  4 18:51 vite.config.js
-rw-r--r--  1 root root 52195 Dec  4 18:51 yarn.lock
I have no name!@docker-desktop:/src/webui$ yarn run dev
Internal Error: EACCES: permission denied, mkdir './.yarn'
Error: EACCES: permission denied, mkdir './.yarn'
I have no name!@docker-desktop:/src/webui$
```

▼ FANTASTIC-COFFEE-DECAFFEINATED

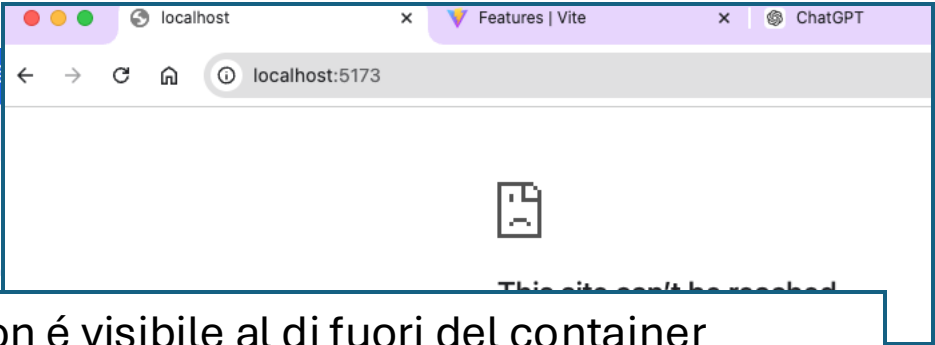
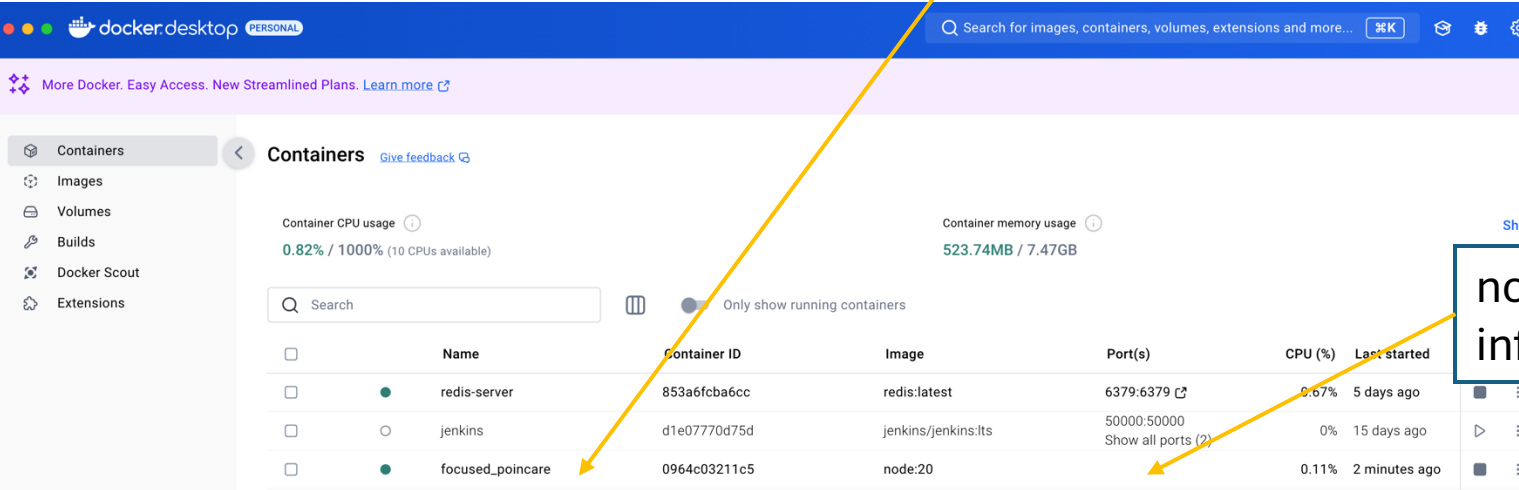
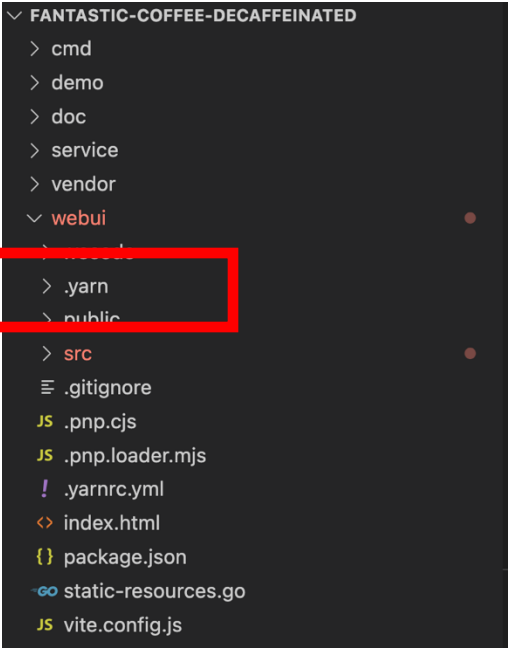
- > cmd
- > demo
- > doc
- > service
- > vendor
- ▼ webui
 - > .vscode
 - > .yarn
 - > public
 - > src
- ≡ .gitignore
- JS .pnp.cjs
- JS .pnp.loader.mjs
- ! .yarnrc.yml
- <> index.html
- { } package.json
- GO static-resources.go
- JS vite.config.js

AVVIAMO L'APP COME DA README.md

puliamo le configurazioni **dall'interno del container**

```
If you want to launch the WebUI, clean yarn cache do the following
```shell
./open-node.sh
(here you're inside the container)
yarn config set globalFolder /src/webui/.yarn
yarn cache clean
yarn install
(test app)
yarn run dev
```

```
VITE v3.2.11 ready in 461 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
```

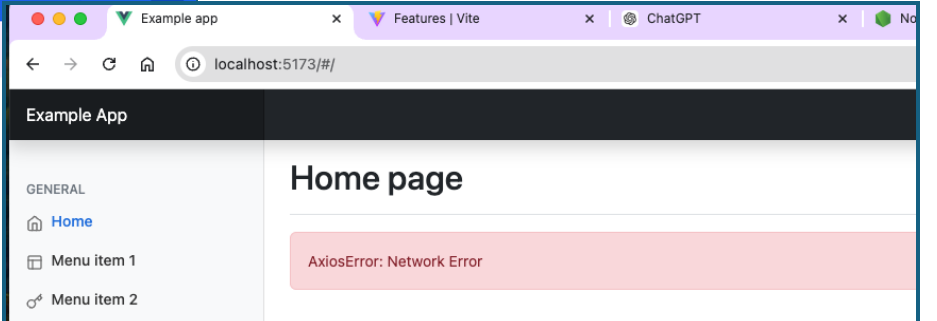
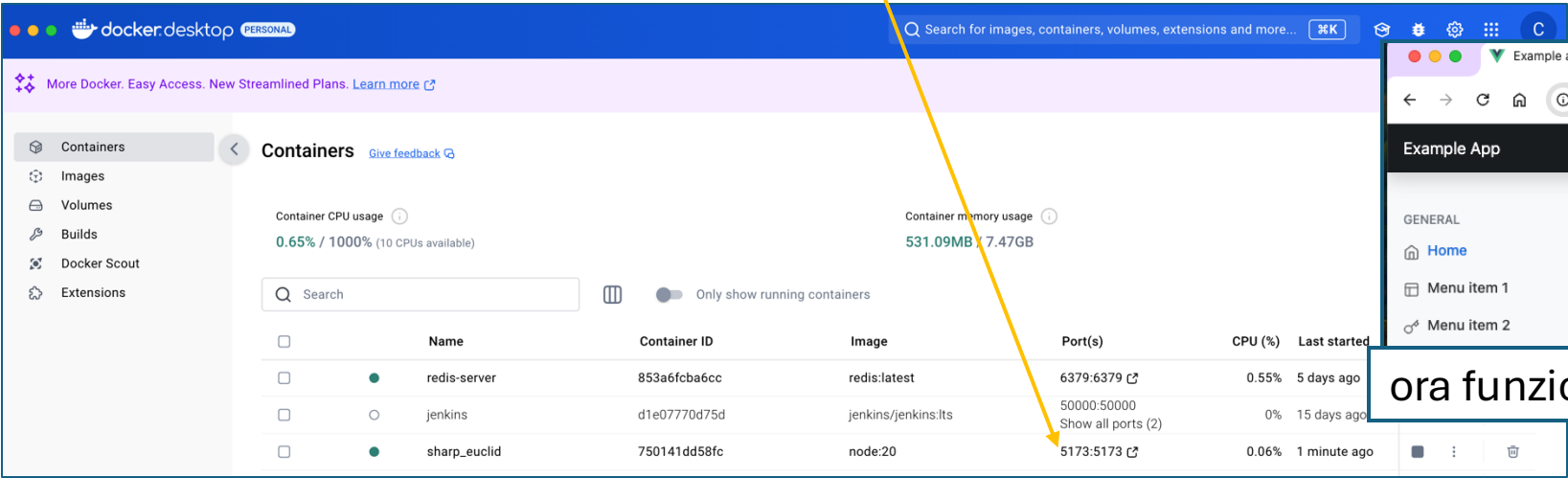
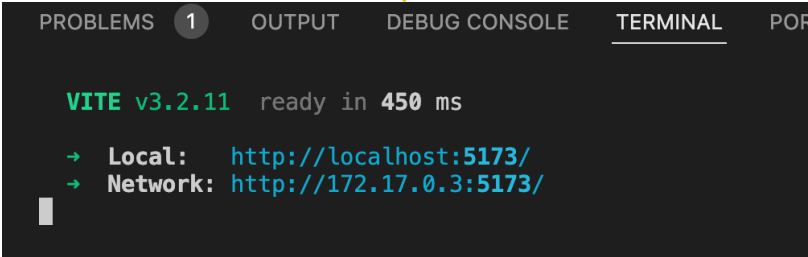
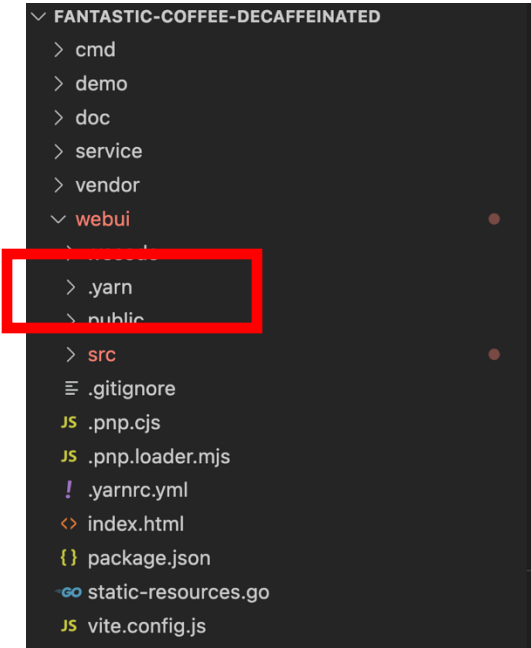


non é visibile al di fuori del container  
infatti manca la configurazione della porta

# AVVIAMO L'APP COME DA README.md

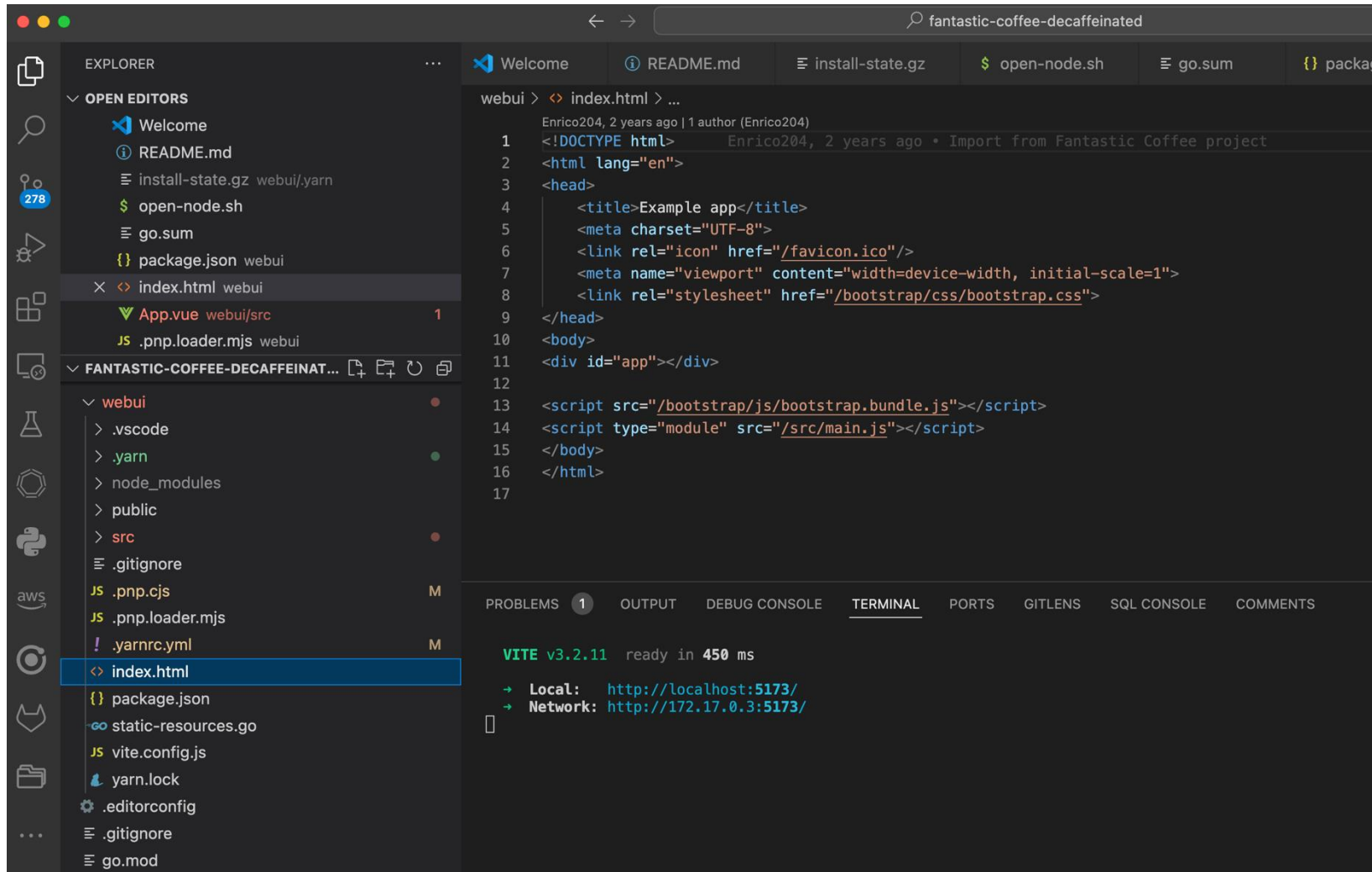
modifichiamo il comando docker per avviare ed esportare l'app

```
docker run -it --rm -v "$(pwd):/src" -u "$(id -u):$(id -g)" --workdir /src/webui -p 5173:5173 node:20 yarn run dev --host
```



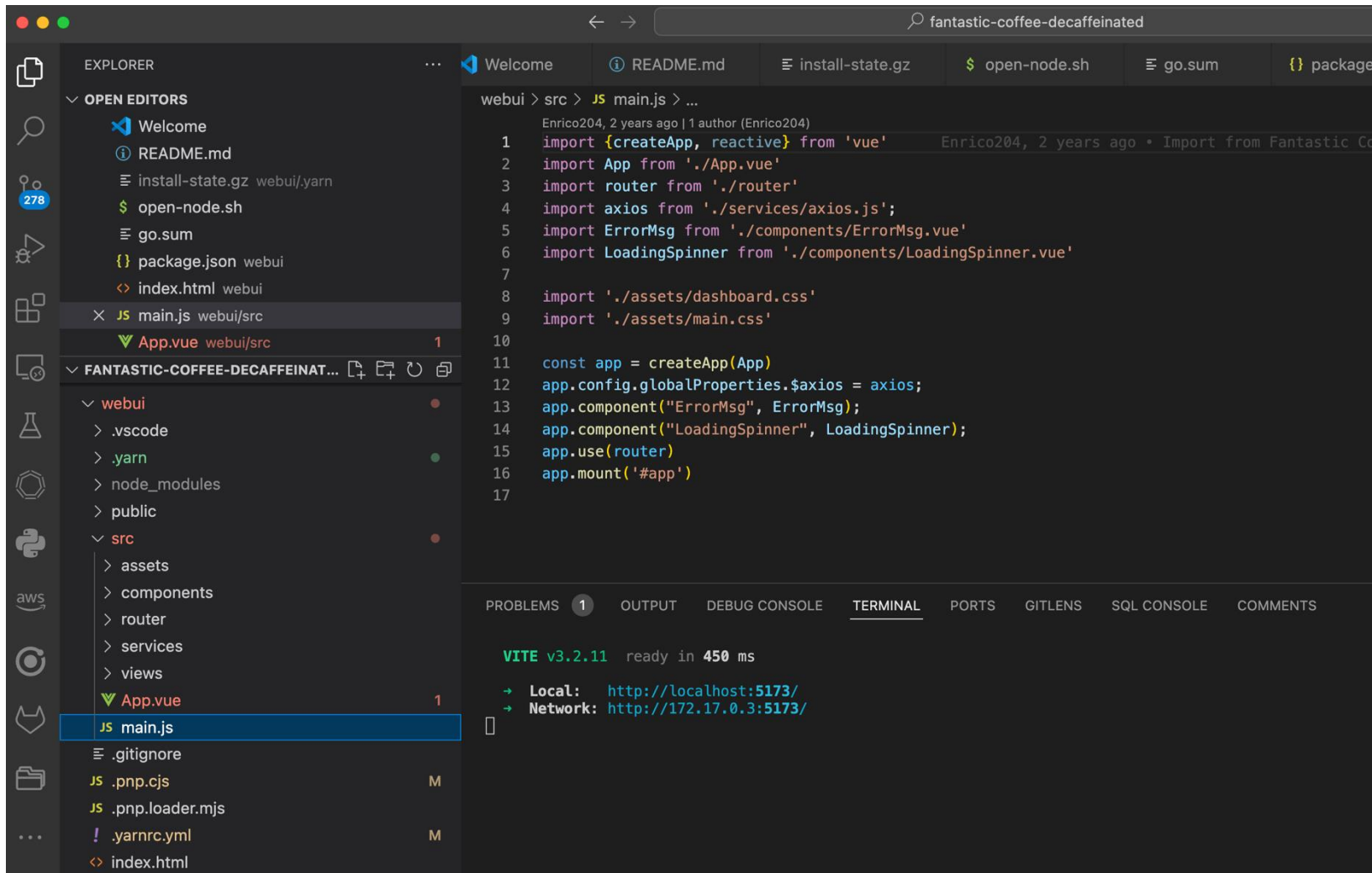
ora funziona

## PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)



il file index.html  
carica il fraemwirk css bootstrapp  
ed avvia l'app con id="app"

## PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)



il file main.js  
carica i module VUE ed avvia l'app  
#app

# PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)

modifichiamo il file App.vue  
come di fianco

```
<template>
<div id="app">
 <nav class="navbar">
 <!-- <div class="navbar-title">Chats</div> -->
 <!-- Left-aligned buttons -->
 <div class="navbar-left">
 <router-link to="/" class="navbar-button">
 <button>Home</button>
 </router-link>
 <router-link to="/users" class="navbar-button">
 <button>Users</button>
 </router-link>
 <router-link to="/about" class="navbar-button">
 <button>About</button>
 </router-link>
 </div>

 <!-- Right-aligned login/logout button -->
 <div class="navbar-right">
 <div class="navbar-title" v-if="username">
 {{ username }}
 </div>
 <router-link v-if="username" to="/login" class="navbar-button">
 <button>Login</button>
 </router-link>
 <router-link v-if="username" to="/" class="navbar-button">
 <button @click="logout">Logout</button>
 </router-link>
 </div>
 </nav>

 <!-- Routed components will be rendered here -->
 <router-view></router-view>
</div>
</template>

<script>
// import { mapState } from 'vuex';

export default {
 name: "App",
 // computed: {
 // ...mapState(['username']), // Map the username from Vuex store to this component
 // },
 methods: {
 logout() {
 // Clear username in Vuex store and sessionStorage
 this.$store.commit("setUsername", null); // Reset username in Vuex store
 },
 },
};
</script>

<style>
.navbar {
 display: flex;
 justify-content: space-between;
 align-items: center;
 background-color: #1ce036;
 padding: 1em;
}

.navbar-title {
 color: white;
 font-size: 1.5em;
 font-weight: bold;
}

.navbar-buttons {
 display: flex;
 gap: 1em;
}

.navbar-button button {
 background-color: white;
 color: #2c3e50;
 padding: 10px 15px;
 border: 1px solid white;
 border-radius: 4px;
 font-size: 1em;
 cursor: pointer;
 transition: background-color 0.3s, color 0.3s;
}

.navbar-button button:hover {
 background-color: #34495e;
 color: white;
}

/* Add padding or margin between buttons */
.navbar-left .navbar-button {
 margin-right: 10px; /* Space between buttons */
}

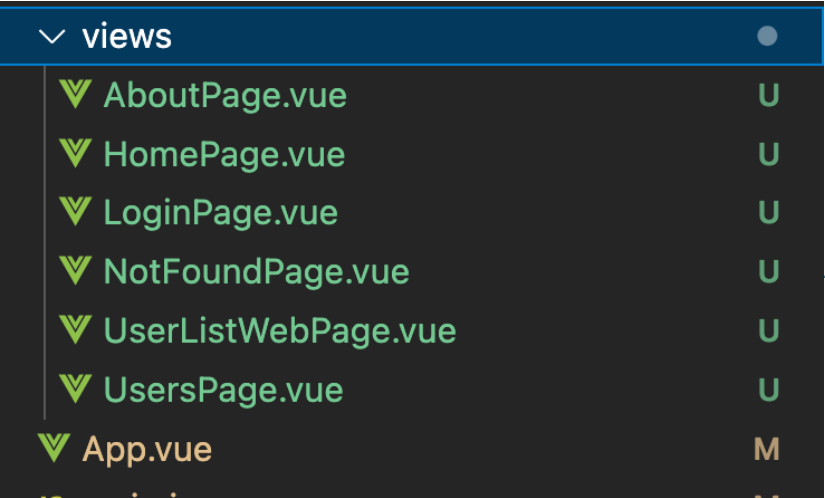
.navbar-left .navbar-button:last-child {
 margin-right: 0; /* Remove margin for the last button */
}

.navbar-button button {
 padding: 8px 16px; /* Optional: Style individual buttons */
}

/* Style the right section as a flexbox */
.navbar-right {
 display: flex;
 align-items: center; /* Align items vertically */
 gap: 10px; /* Add spacing between UNO and DUE */
}

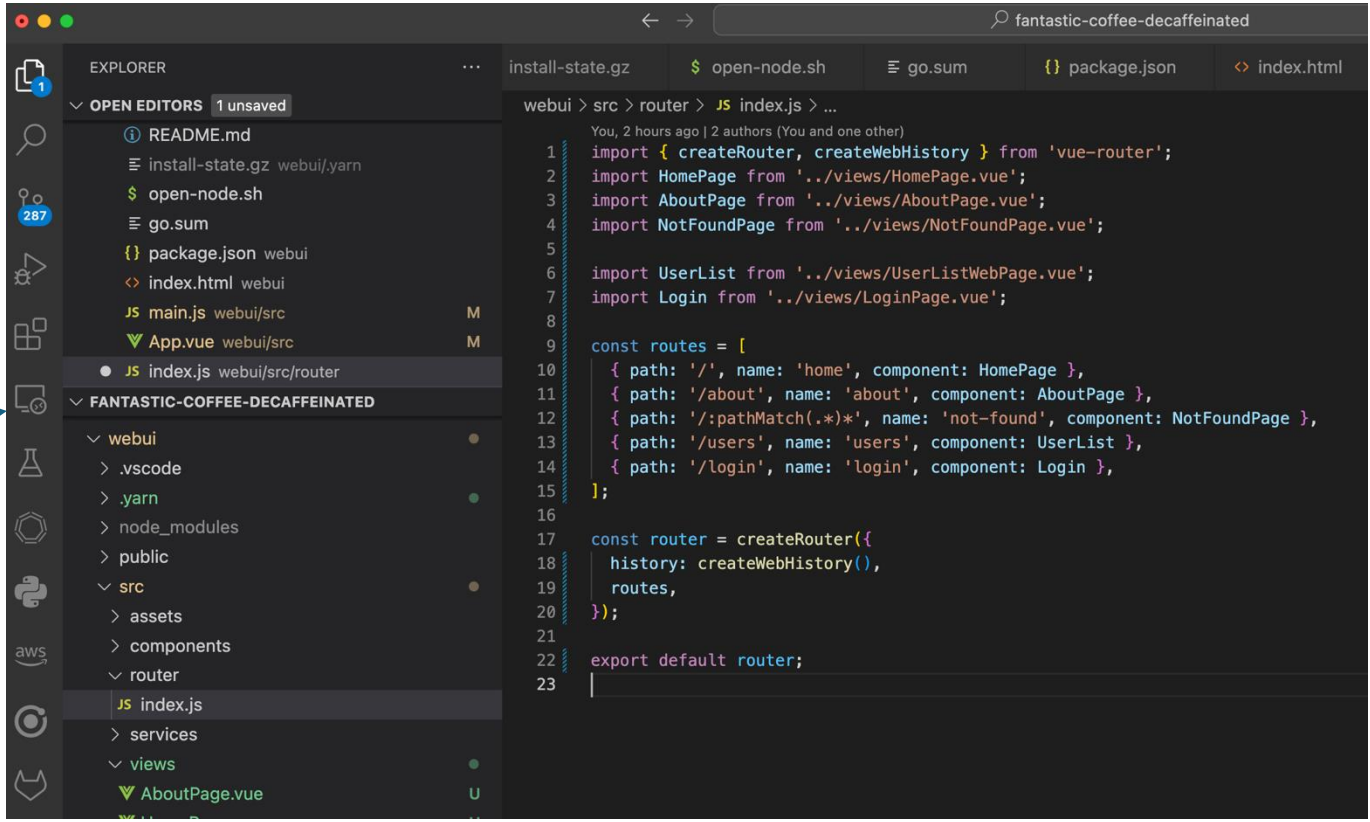
</style>
```

# PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)



sostituiamo le pagine (VIEWS)  
con le nostre

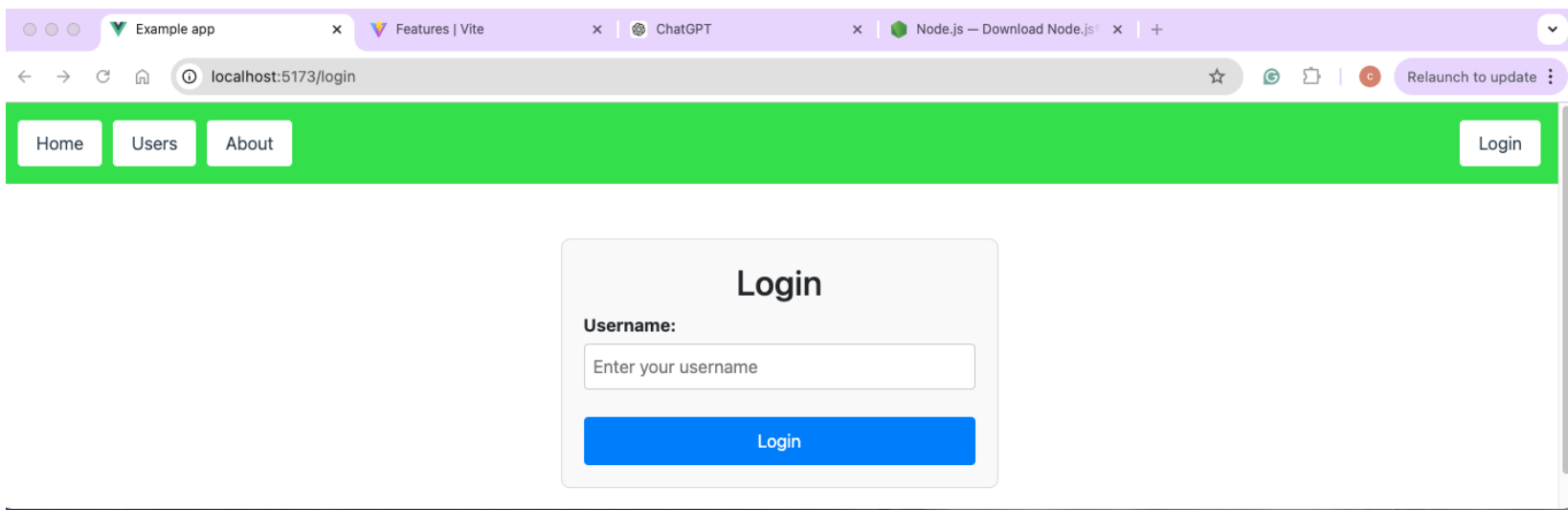
modifichiamo il router





# PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)

la nostra app funziona



# PERSONALIZZIAMO L'APP PER LE NOSTRE ESIGENZE (CHAT)

## LE VIEWS

