

PROJEKT AKADEMIK UNIVERSITAR

TEMA:APLIKACION WEATHER-REACT

PUNOI:DEA LLOJDIAJ

Hyrja

Ne kete detyre kursi une do te paraqes nje aplikacion ne React qe mundeson shfaqjen e motit per qytetet e Shqiperise duke perdorur nje API te jashtme.

Perdoruesit duhet te kene mundesi te shtojne qytete ne nje liste te preferuarave si bookmarks dhe te kerkojne qytete specifike.Ky projekt do kaloj ne disa faza zhvillimi ,ku do te perdor metologjine Agile dhe ato hapa jane:

- identifikimi I problemit dhe kerkesave
- dizenjimi, pra struktura e komponenteve dhe dizanji I UI me Figma
- fizibilitet ekonomik,operacional,teknik
- zhvillimi duke koduar me React
- testim
- implementim.

Identifikoj problemin:

Problemi eshte se perdoruesit perdorin faqe te ndryshme per motin dhe nuk kane mundesi personalizimi,informacion I shperndare,shume te ngarkuar si faqe dhe jo te dizenuara mire,navigim i veshtere.

Une ndertoj aplikacioni per motin per qytetet e Shqiperise ne web duke koduar ne React dhe duke perdorur API te jashtme.E ndertoj per arsyse se u mundesoj perdoruesve te shohin motin me temperaturore aktuale,te kerkojne qytete specifike,te shtojne ne nje liste te preferuar,te ofrohet kerkim I shpejte ,navigim I lehte,dizanj I thjeshte.

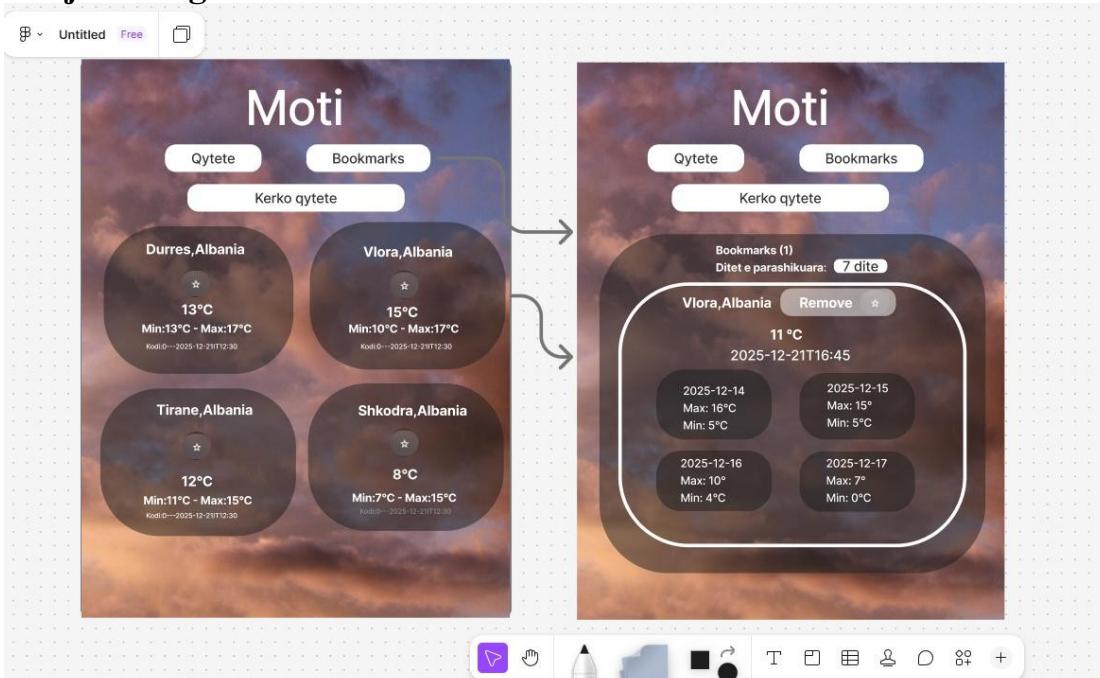
Kerkesat kryesore:

- Marrja e te dhenave perdor Fetch API per te marre te dhenat e motit per cdo qytet nga Open-Meteo dhe ruhen ne state.
- Sekcioni List i qyteteve, qe secila ka emrin,shtetin,temperaturen maksimale dhe minimale dhe nje buton Add to bookmarks qe therret funksionin toggleBookmark.
- Sekcioni Bookmarks ku jane lista e qyteteve te preferuara,ku jane emri,shteti,temperatura,ora,data dhe parashikimi per 7 dite deri ne 16 dite dhe butonin Remove.
- Funksionaliteti i kerkimit:inputi, e cila ruhet ne state si inputTerm.
- Funksionaliteti i Tabs dhe perdore nje state activeTab per te menaxhuar sekcioniin e shfaqur .Butona per te kaluar mes “Cities” dhe “Bookmarks”.
- Funksionet e Bookmark qe implementoj toggleBookmark(qytetId) per te shtuar ose hequr qytete nga bookmarks.Perdore ikona ★ per te shtuar, ☆ per te hequr. Bej removeFromBookmarks(qytetId) per te hequr qytete nga bookmarks.
- Struktura e Komponenteve ,pra detajet e qytetit dhe sekcioni i bookmarks te ndahen ne komponente te vecanta.Perdor props per te kaluar te dhenat.
- Stilizoj komponentet me CSS duke perdorur atributin className.

Kerkesat jofunksionale:

- Performance e mire.
- Siguri bazike ne frontend.
- Dizenjim unik dhe i thjeshte.
- Perdorimi I CSS per stilizim.

Dizanji me Figma:



Linku: <https://www.figma.com/board/d2mCZODpMEKqFafRNo8joz/Untitled?node-id=0-1&t=21oKpG4rSuJEYIZh-1>

Fizibiliteti:

- Operacional:Nuk kerkon trajnim te vecante dhe nderfaqe e thjeshte.
- Teknik:Node.js,Visual Studio Code,laptop per te punuar.
- Ekonomik:Asnjë lloj kosti,vetem nese e bej me domain kushton 20euro.

Zhvillimi:

Ne kete detyre kursi une menyre e zhvillimit te ndertimit te programit te kohes jam bazuar ne baze te kerkesave dhe shpjeguar cdo hap me detaje dhe me screenshots se si I kam koduar.Perdon React,Open-Meteo API te jashtme,CSS per stilizim.

Hapi 1:Instaloj Node.js dhe Visual Studio Code.

Node.js me ndihmon, sepse eshte motor qe ben te mundur punen e React ne kompjuterin tone gjate zhvillimit. Ai perdoret per te instaluar paketat qe nevojiten, per te perqatitur kodin React, qe cdokush mund ta kuptoje dhe per te nisur serverin lokal per testim. Ndersa VSC eshte editor per te shkruar kodin.

```
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dea> node -v
v24.12.0

C:\Users\Dea>
```

Hapi 2:Aktivizoj librarite e Reactit.

Ne VSC te terminali shkruaj keto:

The screenshot shows the VS Code interface with the Terminal tab selected. The terminal output is as follows:

```
PS C:\Users\Dea\Downloads\dea_llojdaj_web> npm create vite@latest weather-app --template react
q
npm notice
npm notice New minor version of npm available! 11.6.2 -> 11.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.7.0
npm notice To update run: npm install -g npm@11.7.0
npm notice
PS C:\Users\Dea\Downloads\dea_llojdaj_web> npm create vite@latest weather-app --template react
>> cd weather-app
>> npm install

> npx
> create-vite weather-app --template react

diamond Target directory "weather-app" is not empty. Please choose how to proceed
diamond Remove existing files and continue
diamond Use rollup-vite (Experimental)?:
No

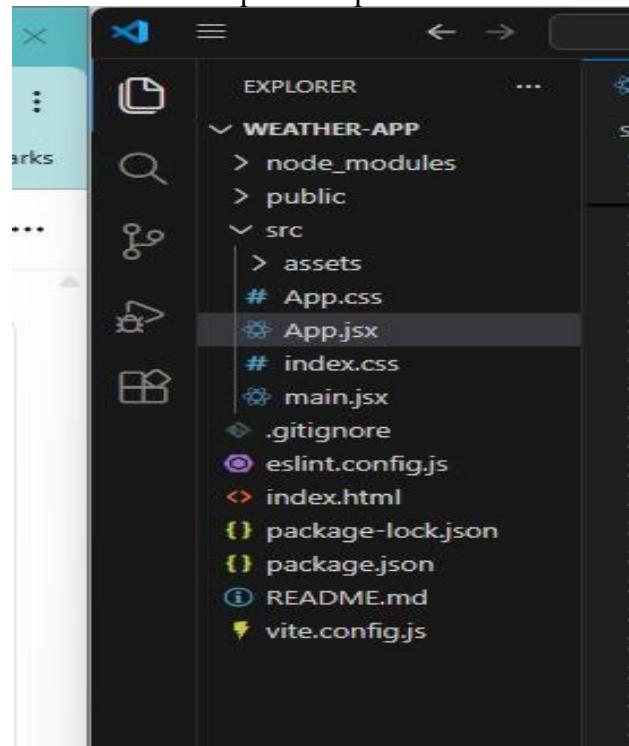
diamond Install with npm and start now?
Yes

diamond Scaffolding project in C:\Users\Dea\Downloads\dea_llojdaj_web\weather-app...
...
diamond Installing dependencies with npm...
```

To the right of the terminal, a sidebar displays the project structure and build status:

- FOUND 0 VULNERABILITIES
- Starting dev server...
- > weather-app@0.0.0 dev
- > vite
- VITE v7.2.7 ready in 3431 ms
- Local: <http://localhost:5173>
- Network: use --host to expose
- press h + enter to show help

U ndertua folderi perkates per React weather:



Ky hap krijon projektin React me Vite, instalon te gjitha paketat qe i nevojiten dhe te percatit ambientin per zhvillim, ne menyre qe te mund te nisesh te shkruash kodin direkt.

<http://localhost:5173/> – Me kete, eshte adresa ku do shfaqet aplikacioni im lokalish ne kompjuterin dhe une mund te shoh cdo ndryshim qe bej. Me ane te terminalit qe do dua te shikoj si eshte aplikacioni ne React ne web do shkruaj npm run dev dhe ate link e shkruaj ne web dhe hapet faqja.

Hapi 3: Struktura e skedareve si do te ndertohet aplikacion ne React ose Arkitektura e Aplikacionit:

Folder:

Weather/

--Src/

-----**Components/**

----**DetajeQyteti.jsx** (Komponent per informacion te detajuar per nje qytet)

----**SekzioniBookmarks.jsx** (menaxhon ose shfaq qytete e ruajtura)

-----**Data/**

----**qytete.js** (liste objektesh qe perfaqeson objekte)

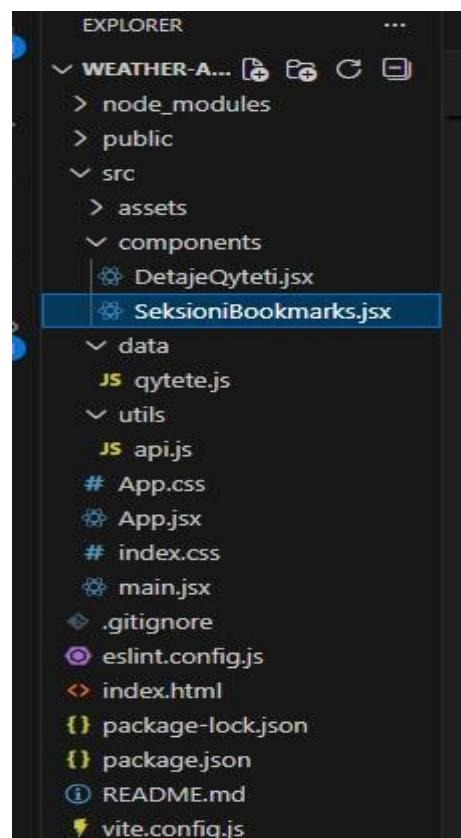
-----**Utils/**

-----**api.js**(funksione ndihmese)

-----**App.jsx** (komponenti kryesore I aplikacionit)

-----**App.css** (stilim)

-----**main.jsx**(pika hyrese e aplikacionit)



Hapi 4: Krijojme sektionin e listes se qyteteve ne VSC= src/data/qytete.js

Krijova nje array qyteteShqiptare ku me ane te export e ben te disponueshme per tu marr ne skedare te tjere ku secili elemente eshte nje objekt ku kemi id, e cila eshte key ne React ,lookup ne API dhe perdoret URL,emri,shteti ,gjatesi dhe gjeresi.Keto te dhena perdoren per shfaqjen e listes ,per kerkimin ,per bookmarks dhe per thirrjen e API.

```
c> data > JS qyteteShqiptare > emri
1  export const qyteteShqiptare = [
2    { id: "tirana", emri: "Tirana", shtet: "Albania", gjeresi: 41.3275, gjatesi: 19.8187 },
3    { id: "durres", emri: "Durres", shtet: "Albania", gjeresi: 41.3231, gjatesi: 19.4414 },
4    { id: "vlore", emri: "Vlora", shtet: "Albania", gjeresi: 40.4667, gjatesi: 19.4894 },
5    { id: "shkoder", emri: "Shkodra", shtet: "Albania", gjeresi: 42.0687, gjatesi: 19.5126 },
6    { id: "elbasan", emri: "Elbasan", shtet: "Albania", gjeresi: 41.1128, gjatesi: 20.0828 },
7    { id: "fier", emri: "Fier", shtet: "Albania", gjeresi: 40.7167, gjatesi: 19.5667 },
8    { id: "korce", emri: "Korce", shtet: "Albania", gjeresi: 40.6167, gjatesi: 20.7833 },
9    { id: "berat", emri: "Berat", shtet: "Albania", gjeresi: 40.7058, gjatesi: 19.9522 },
10   { id: "lushnje", emri: "Lushnja", shtet: "Albania", gjeresi: 40.9419, gjatesi: 19.7050 },
11   { id: "sarande", emri: "Saranda", shtet: "Albania", gjeresi: 39.8756, gjatesi: 20.0053 },
12   { id: "pogradec", emri: "Pogradec", shtet: "Albania", gjeresi: 40.9000, gjatesi: 20.6583 },
13   { id: "kukes", emri: "Kukes", shtet: "Albania", gjeresi: 42.0769, gjatesi: 20.4219 },
14   { id: "lezhe", emri: "Lezhe", shtet: "Albania", gjeresi: 41.7861, gjatesi: 19.6464 },
15   { id: "gjirokaster", emri: "Gjirokaster", shtet: "Albania", gjeresi: 40.0750, gjatesi: 20.1389 },
16   { id: "peshkopi", emri: "Peshkopi", shtet: "Albania", gjeresi: 41.6833, gjatesi: 20.4333 },
17   { id: "kruje", emri: "Kruje", shtet: "Albania", gjeresi: 41.5092, gjatesi: 19.7928 },
18   { id: "kavaje", emri: "Kavaje", shtet: "Albania", gjeresi: 41.1856, gjatesi: 19.5561 },
19   { id: "mallakaster", emri: "Mallakasta", shtet: "Albania", gjeresi: 40.5333, gjatesi: 19.7167 },
20   { id: "tepelene", emri: "Tepelena", shtet: "Albania", gjeresi: 40.2956, gjatesi: 20.0186 }
21 ];
22 
```

Hapi 5: Krijojme utils/api.js = funksion per API.

Une ketu krijova një URL per Open-Meteo API duke perdorur koordinatat e një qyteti dhe disa opsiione, per te marre motin aktual, historik dhe parashikimin ditor. Funksioni merret export dhe me parametrat gjatesi,gjeresi dhe një objekt opsiونesh. Paraqet një variabel te pandryshueshme dhe merr opsiونe timezone ose merr Europe/Tirane:

Paraqet një diteKaluara te pandryshueshme nga objekti option nese ekziston nese jo merr vleren 7.

Sa dite parashikimi moti do paraqitet =7 dhe per cdo dite do mar maksimum,minumum dhe kodin e motit.

Te rreshti I 8 krijova një objekt që krijon automatikisht string per parametrat URL dhe cdo key:vlere kthehet key=value ne URL.

```
c> utils > JS api.js > ...
1  export function UrlMotit(gjeresi, gjatesi, opzion = {}) {
2    const timezone = opzion.timezone || "Europe/Tirane";
3    const diteKaluara = opzion.diteKaluara ?? 7;
4    const diteParashikimi = opzion.diteParashikimi ?? 7;
5    const daily = "temperature_2m_max,temperature_2m_min,weathercode";
6
7    const search = new URLSearchParams({
8      latitude: String(gjeresi),
9      longitude: String(gjatesi),
10     current_weather: "true",
11     timezone,
12     past_days: String(diteKaluara),
13     forecast_days: String(diteParashikimi),
14     daily,
15   });
16
17   return `https://api.open-meteo.com/v1/forecast?${search.toString()}`;
18 }
19 
```

Hapi 6: Krijojme DetajeQyteti.jsx.

Une krijoja nje komponent React qe shfaq detajet e qytetit dhe motin e tij, me nje buton per te shtuar ose hequr qytetin nga bookmarks.

-Me ane te export e ben komponentin te importohet direkt ne skedare te tjere ,me ane te parametrave props qe jane qytet,weather,ToggleBookmark eshte funks qe ekzekutohet kur shtypet butoni dhe Bookmark eshte boolean nese qyteti eshte I ruajtur atje.

Div shfaq emrin dhe shtetin e qytetit.

Kur klikoj ToggleBookmark me ate id qyteti ajo menjeher identifikohet kush qytet eshte na id unike qe ka dhe shfaq yll te mbushur na kushti ose te pa mbushur.

-Shfaqja e motit rreshti 18 e poshte:

Kushti I motit qe ne qofte se eshte True do te shfaq temperaturen aktuale qe e merr na API dhe te div tjeter motit aktual me temperature maksimale dhe minimale ,perndryshe ne pjesen False te kushtit do dal Loading.

```
src > components > DetajeQyteti.jsx > ...
1  import React from "react";
2
3  export default function DetajeQyteti({ qytet, weather, ToggleBookmark, Bookmark }) {
4    return (
5      <div className="card">
6        <div style={{ display: "flex", justifyContent: "space-between" }}>
7          <div>
8            <div className="qytet-emri">{qytet.emri}, {qytet.shtet}</div>
9            <div className="vogel">Gjeresi: {qytet.gjeresi} • Gjatesi: {qytet.gjatesi}</div>
10           </div>
11           <button onClick={ToggleBookmark}>{Bookmark ? "★" : "☆"}</button>
12         </div>
13         {weather ? (
14           <div style={{ marginTop: 15 }}>
15             <div style={{ fontSize: 34, fontWeight: 700 }}>
16               <div className="time-code">{parseInt(weather.current_weather.temperature)}°C</div>
17               <div className="vogel">
18                 Min: {Math.round(weather.daily.temperature_2m_min[0])}°C •
19                 Max: {Math.round(weather.daily.temperature_2m_max[0])}°C
20               </div>
21               <div className="code-1">Kodi: {weather.current_weather.weathercode} – {weather.current_weather.time}</div>
22             </div>
23           </div>
24         ) : (
25           <div className="vogel">Po ngarkon...</div>
26         )
27       </div>
28     );
29   }
30 }
```

Hapi 7: Krijoj SeksioniBookmarks.jsx.

SeksioniBookmarks merr disa props na prindi:qyteteBookmarks,weatherMap,removeFromBookmarks....

Bookmarks({})= tregon sa qytete jane brenda saj.

select = menu me opsione ,ku value=ditet e parashikuara jane te zgjedhura dhe diteParashikuar eshte nje state.

onChange kur perdoruesi zgjedh nje x opzion tjeter merr vleren e opzionit te targetuar me e qe te jete numer dhe e ruan kete vlera ne diteParashikuar si funksion qe eshte setDiteParashikuar.

7,10....map =array me numra dhe map krijon nje opzion per secilen numer qe key te gjurmohet dhe vlera shfaqet ajo qe eshte ne key.

```
src > components > SeksioniBookmarks.jsx > SeksioniBookmarks
1  import React from "react";
2
3  export default function SeksioniBookmarks({ qyteteBookmark, weatherMap, removeFromBookmarks, diteParashikuar, setDiteParashikuar }) {
4    return (
5      <div>
6        <div>
7          <div>Bookmarks ({qyteteBookmark.length})</div>
8          <div>
9            Forecast days:
10            <select value={setDiteParashikuar} onChange={(e) => setDiteParashikuar(Number(e.target.value))}>
11              {[7, 10, 14, 16].map(dite => (
12                <option key={dite} value={dite}>{dite}dite</option>
13              ))
14            </select>
15          </div>
16        </div>
17      </div>
18    );
19  }
```

Bej nje kontroll per qytete e bookmark nese =0 kushti True jep nuk ka bookmarks ,nese eshte False=do te behet mapim per cdo qytet duke e lidhur me qytet id.Te rikthimi celesi do te jet id unike e qytetit me emrin dhe shtetin perkates dhe butonin remove qe hiqet na id e qytetit.

```

18     {qyteteBookmark.length === 0 ? (
19       <div>
20         | Nuk ka bookmarks.
21       </div>
22     ) : [
23       qyteteBookmark.map(qytet => {
24         const w = weatherMap[qytet.id];
25
26         return (
27           <div key={qytet.id}>
28             <div>
29               <div>
30                 <div>{qytet.emri}, {qytet.shtet}</div>
31               </div>
32               <button onClick={() => removeFromBookmarks(qytet.id)}>Remove ☆</button>
33             </div>

```

Shfaqja e motit ku w eshte objekti I motit per qytetin dhe behet kushte ,ne qofte se eshte True shfaq temperaturen aktuale dhe kohen te parashikuara,daten dhe temperaturnen max dhe min, perndryshe shfaq po e ngarkon ne qofte se eshte False kushti.

```

29   | {w ? (
30     <div>
31       <div>{w.current_weather.temperature}°C – {w.current_weather.time}</div>
32       <div className="vogel" style={{ marginTop: 5 }}>Dite e parashikuara:</div>
33
34       <div className="forecast-row">
35         {[w.daily.time.slice(0, diteParshikuar).map((a, i) => (
36           <div key={a} className="forecast-card">
37             <div className="forecast-date">{a}</div>
38             <div className="vogel">Max: {w.daily.temperature_2m_max[i]}°C</div>
39             <div className="vogel">Min: {w.daily.temperature_2m_min[i]}°C</div>
40           </div>
41         ))]
42       </div>
43     )
44     : <div className="vogel">Po ngarkon...</div>
45   </div>
46   );
47 }
48 )
49 </div>
50 );
51

```

Hapi 8:Krijoj App.jsx .

State Management ne App.jsx dhe importet :

Use State perdoret per te ruajtur gjendjen ne komponent.

Bookmarks=lista e qyteteve ne bookmarks ruhen ne localStorage dhe incializohet me funx setBookmarks qe merr na alb... ose do jete bosh.

InputTerm =kerkimi I qyteteve.

activeTab eshte secilen tab do jete ose cities ose bookmarks.

UseEffect eshte hook qe menaxhon efektet anesore dhe ketu sinkronizohet me localStorage qe sa here ndryshon bookmarks ,hook shkruan gjendjn aktuale.

```

: > App.jsx > App > toggleBookmark
1 import React, { useState, useEffect } from "react";
2 import { qyteteShqiptare } from "./data/qytete";
3 import DetajeQyteti from "./components/DetajeQyteti";
4 import SeksioniBookmarks from "./components/SeksioniBookmarks";
5 import { UrlMotit } from "./utils/api";
6 import "./App.css";
7
8 export default function App() {
9   const [hartaMoti, setHartaMoti] = useState({}); // State for current map
10  const [bookmarks, setBookmarks] = useState(() => JSON.parse(localStorage.getItem("motiBookmarks") || "[]")); // State for bookmarks
11  const [inputTerm, setInputTerm] = useState("");
12  const [tabAktiv, setTabAktiv] = useState("qytete");
13  const [diteParashikuar, setDiteParashikuar] = useState(7);
14
15  useEffect(() => {
16    localStorage.setItem("motiBookmarks", JSON.stringify(bookmarks));
17  }, [bookmarks]);
18}

```

Funksion moti asynk eshte nje funksioni punon me internet ku qytet dhe dite jane parametra te funksionit.

Try and catch perdoret per menaxhimin e gabimeve dhe kodi gjenerohet brenda try dhe ne qofte se ska gabime nuk ekzekutohet.

Ndertoj URL e motit .Marr motin na interneti:await fetch(url).

Await res.json()=kthen pergjigje ne object JavaScript.

Ruaj motin e ketij qyteti funksioni setHartaMoti ,sepse prev eshte state I vjeter ,e kopjon dhe shkon moti per qytetin aktual.

Catch =kapja e gabimeve .

useEffect ekzekutohet kur komponenti hapet dhe sa here e ndryshoj diteParashikuar por ketu une marr listen e qyteteve dhe me ane te loop marr cdo qytet dhe theras funksionin moti ku q eshte nje objekt pra qytet dhe [] lista e array qe varet efekti.

```

async function Moti(qytet, dite = 7) {
  try {
    const url = UrlMotit(qytet.gjeresi, qytet.gjatesi, {
      diteParashikimi: dite,
      diteKaluara: 7,
    });
    const res = await fetch(url);
    const data = await res.json();

    setHartaMoti(prev => ({
      ...prev,
      [qytet.id]: data,
    }));
  } catch (e) {
    console.error("Gabim në marrjen e motit:", qytet.emri);
  }
}

useEffect(() => {
  qyteteShqiptare.forEach(q =>
    Moti(q, diteParashikuar)
  );
}, [diteParashikuar]);

```

Bej filtrim e qyteteve qe krijon nje array te ri qe te plotesohet kushti dhe cdo objekt te merret emri me germa te vogla dhe inputi I perdoruesit inputTerm dhe e konverton me te vogla dhe include nqs ekziston brenda emri .

Krijoj nje list bookmark ku behet array I ri na id e qyteteve me kusht qe objekt te kene id te njejte.

Funksioni toggleBookmark merr id si parameter ,perditeson state setBookmarks me bazen aktuale dhe vendos kusht nese qyteti perfshihet ne bookmarks hiqe nese jo shtoje.

```

47
48   const qyteteFiltruara = qyteteShqiptare.filter(q =>
49     q.emri.toLowerCase().includes(inputTerm.toLowerCase())
50   );
51
52   const qyteteBookmark = bookmarks
53     .map(id => qyteteShqiptare.find(q => q.id === id))
54     .filter(Boolean);
55
56   function toggleBookmark(id) {
57     setBookmarks(prev =>
58       prev.includes(id)
59         ? prev.filter(b => b !== id)
60         : [...prev, id]
61     );
62   }

```

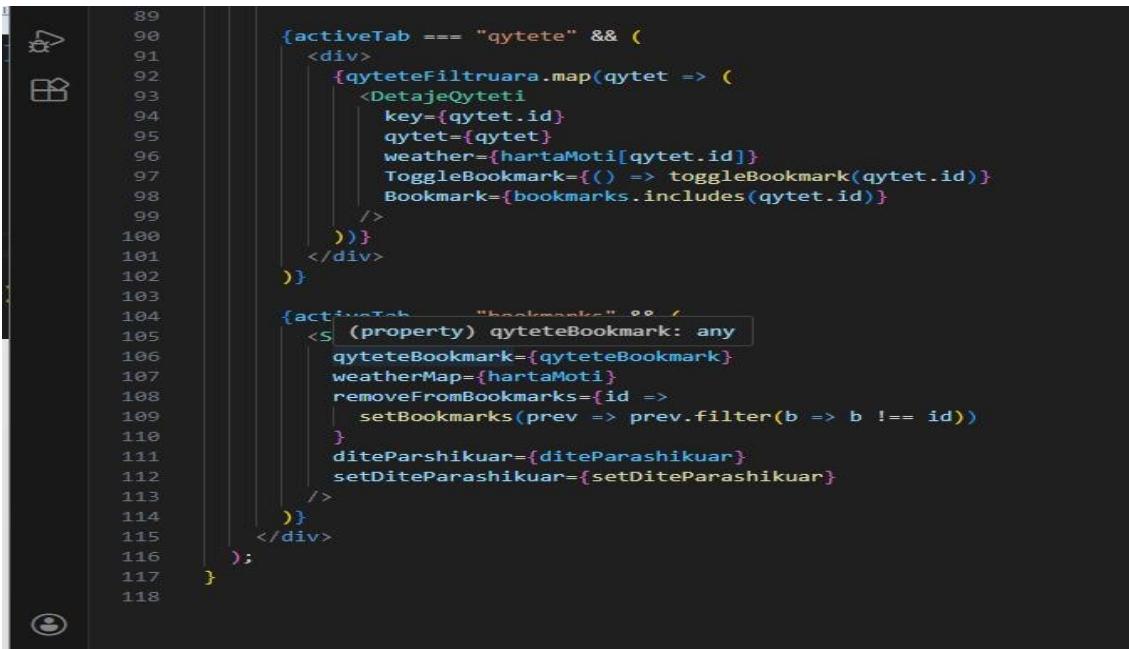
Titulli I faqes,krijohen 2 butona ku ben tab e zgjedhur aktive dhe kur klikohet ndryshon me kete funksion. Input=krijon nje fushe per te shkruar qytetin ,input=state dhe me onChange perditeson input sa here shkruhet.

```

63
64   return (
65     <div className="app">
66       <h1>Moti □ Shqipëria</h1>
67
68       <div className="tabs">
69         <button
70           className={activeTab === "qytete" ? "active" : ""}
71           onClick={() => setActiveTab("qytete")}
72         >
73           Qytete
74         </button>
75         <button
76           className={activeTab === "bookmarks" ? "active" : ""}
77           onClick={() => setActiveTab("bookmarks")}
78         >
79           Bookmarks
80         </button>
81       </div>
82
83       <input
84         className="search"
85         placeholder="Kërkoi qytet"
86         value={inputTerm}
87         onChange={e => setInputTerm(e.target.value)}
88       />

```

Behet kontrolli I tabit dhe behet shfaqja e qyteteve duke u krijuar list me detajet specifike,id ,qyteti,koha ,funksioni togglebookmark per shtimin dhe heqjen nga bookmarks.Gjithashtu shfaqja e bookmarks ku dergohen props per qyteteBookmarks ,weatherMap =te dhenat e motit ,funksioni per te hequr qytetet dhe ditet e parashikuara.Pra ,une zgjedh nje tab ke te dua dhe activeTab ndryshon dhe JSX kontrollon nese qytet kan DetajeQytete dhe bookmarks SeksioniBookmarks.



```
89
90     {activeTab === "qytete" && (
91       <div>
92         {qyteteFiltruara.map(qytet => (
93           <DetajeQyteti
94             key={qytet.id}
95             qytet={qytet}
96             weather={hartaMoti[qytet.id]}
97             ToggleBookmark={() => toggleBookmark(qytet.id)}
98             Bookmark={bookmarks.includes(qytet.id)}
99           />
100        ))}
101      </div>
102    )}
103
104    {activeTab === "parashikuar" && (
105      <div>
106        {property} qyteteBookmark: any
107        qyteteBookmark={qyteteBookmark}
108        weatherMap={hartaMoti}
109        removeFromBookmarks={id =>
110          setBookmarks(prev => prev.filter(b => b !== id))
111        }
112        diteParashikuar={diteParashikuar}
113        setDiteParashikuar={setDiteParashikuar}
114      </div>
115    );
116  );
117}
118
```

Hapi 9: Kodimi I faqes App.css per stilizimin e faqes:

Me html,body ku shfaqet permbytja dhe root =elementi ku React ngarkon gjithe aplikacioni dhe dizenjoj me height dhe width zene tere ekranin,margin =hapesira jashte do jete 0,padding=hapesira brenda 0.Background-size e mbulon tere ekranin.Overflow =ndo permbajtje qe del jashte ekranit fshihet.

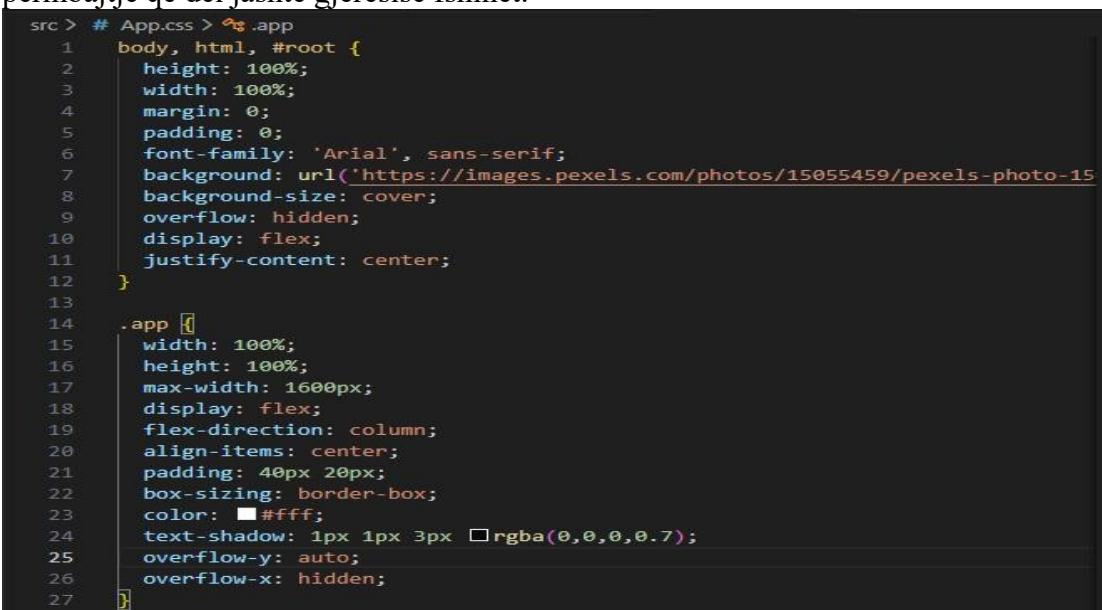
Te klasa .app eshte zona kryesore e aplikacionit qe dizenjohet.

Display flex te rregullohen mire elementet e brendshem,flex-direction vendosen elementet njeri mbi tjeterin,align-items:center= qendrojne ne qender elementet.

Padding =hapesira brenda saj 40 px lart e poshte dhe 20px majtas dhe djathtas.

Box-sizing :border-box =padding dhe border te jen brenda lartesise dhe gjerese totale.

Overflow-y:auto =nese kalon lartesia te behet scroll ,overflow-x:hidden=ndo permbajtje qe del jashte gjerese fshihet.



```
src > # App.css > .app
1  body, html, #root {
2    height: 100%;
3    width: 100%;
4    margin: 0;
5    padding: 0;
6    font-family: 'Arial', sans-serif;
7    background: url('https://images.pexels.com/photos/15055459/pexels-photo-15055459.jpeg?auto=compress&cs=tinysrgb&h=350');
8    background-size: cover;
9    overflow: hidden;
10   display: flex;
11   justify-content: center;
12 }
13
14 .app {
15   width: 100%;
16   height: 100%;
17   max-width: 1600px;
18   display: flex;
19   flex-direction: column;
20   align-items: center;
21   padding: 40px 20px;
22   box-sizing: border-box;
23   color: #fff;
24   text-shadow: 1px 1px 3px rgba(0,0,0,0.7);
25   overflow-y: auto;
26   overflow-x: hidden;
27 }
```

Stilizimi I titullit:

```
28
29   h1 {
30     font-size: 3rem;
31     margin-bottom: 30px;
32     text-align: center;
33     text-shadow: 0 0 10px ■#fff, 0 0 20px, 0 0 30px;
34     font-family: 'Times New Roman', Times, serif;
35   }
36
```

Me .tabs stiloj butonat e skedareve,ku butonat jane njeri pas tjetrit me hapesira 15px dhe hapesira poshte permbajtjes 25 px.Kur nje buton eshte i aktivizuar ose kur eshte hover domethene kur pulsori kalon mbi te, sfondi ndryshon dhe teksti kthehet ne te zi.

```
36
37   .tabs {
38     display: flex;
39     gap: 15px;
40     margin-bottom: 25px;
41   }
42
43   .tabs button {
44     padding: 12px 28px;
45     border-radius: 25px;
46     border: none;
47     background: ■rgba(255,255,255,0.2);
48     color: ■#fff;
49     cursor: pointer;
50     font-weight: bold;
51     transition: all 0.3s ease;
52   }
53   .tabs button.active,
54   .tabs button:hover {
55     background: ■rgba(255,255,255,0.6);
56     color: ■#000;
57 }
```

Stiloj search=inputin me tekst ne qender,me hark ,gjeresi 400px,nuk tejkalon 90 te gjeresise ,margin-bottom=hesira poshte inputit,outline=konturi nuk fokusohet.
.cards-container stilizon kete me: grid per te krijuar layout 2 dimensional si matric me rreshta dhe kolona.Justify-items qendron horizontalisht cdo karte.

```
{activeTab === "qytete" && (
  <div className="cards-container">
    {qyteteFiltruara.map(qytet => (
      <DetajeQyteti
        key={qytet.id}
        qytet={qytet}
        weather={hartaMoti[qytet.id]}
        ToggleBookmark={() => toggleBookmark(qy
        Bookmark={bookmarks.includes(qytet.id)}
```

Stiloj me .card kete :

```
3   export default function DetajeQyteti({ qytet, we
4     return (
5       <div className="card">
6         <div style={{ display: "flex", justifyCont
7           <div>
8             <div className="qytet-emri">{qytet.emr
9               <div className="vogel">Gjeresi: {qytet
10              </div>
11              <button onClick={ToggleBookmark}>{Bookma
12              </div>
13              {weather ? (
14                <div style={{ marginTop: 15 }}>
15                  <div style={{ fontSize: 34, fontWeight
16                    <div className="time-code">{weather.c
17                    <div className="code-1">Kodi: {weather
18                  </div>
```

```

57
58 .search {
59   padding: 12px 20px;
60   border-radius: 25px;
61   border: none;
62   width: 400px;
63   max-width: 90%;
64   margin-bottom: 30px;
65   outline: none;
66   text-align: center;
67   font-size: 1rem;
68 }
69
70 .cards-container {
71   display: grid;
72   grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
73   gap: 25px;
74   width: 100%;
75   justify-items: center;
76   padding-bottom: 50px;
77 }
78
79 .card {
80   background: linear-gradient(135deg, transparent 0%, transparent 45px, #f0f0f0 45px, #f0f0f0 50px, black 50px, black 100px);
81   border-radius: 20px;
82   padding: 20px;
83   box-shadow: 0 8px 25px linear-gradient(135deg, transparent 0%, transparent 45px, #f0f0f0 45px, #f0f0f0 50px, black 50px, black 100px);
84   display: flex;
85   flex-direction: column;
86   gap: 12px;
87   backdrop-filter: blur(6px);
88   width: 280px;
89   min-width: 280px;
90   text-align: center;
91   flex-shrink: 0;
92 }
93
94

```

Stiloj :emrin e qytetit,.vogel=informacionet shtese,.time-code per te dhena me te vogla,buttonat,Butonat kur vendoset pulsori siper, .card>div =brendesia e kartes organizohet vertikalish me hapesire midis elementeve dhe qendruar horizontalisht.

```

95 .qytet-emri {
96   font-weight: bold;
97   font-size: 1.3rem;
98 }
99
100 .vogel {
101   font-size: 0.9rem;
102   color: #ddd;
103 }
104
105 .time-code {
106   font-size: 0.65rem;
107   color: linear-gradient(135deg, transparent 0%, transparent 45px, #f0f0f0 45px, #f0f0f0 50px, black 50px, black 100px);
108 }
109 button {
110   border-radius: 25px;
111   padding: 8px 15px;
112   cursor: pointer;
113   border: none;
114   transition: all 0.3s ease;
115   background: linear-gradient(135deg, transparent 0%, transparent 45px, #f0f0f0 45px, #f0f0f0 50px, black 50px, black 100px);
116   color: #fff;
117   font-size: 0.95rem;
118 }
119
120 button:hover {
121   transform: scale(1.05);
122   background: linear-gradient(135deg, transparent 0%, transparent 45px, #f0f0f0 45px, #f0f0f0 50px, black 50px, black 100px);
123   color: #000;
124 }
125
126 .card > div {
127   display: flex;
128   flex-direction: column;
129   gap: 10px;
130   align-items: center;
131 }

```

Ketu krijoj një faqe ku te gjitha kartat e parashikimit dhe bookmark-et duken moderne dhe te organizuara bukur. Kartat jane te rrumbullakosura, me sfond gjysme-transparent dhe çdo tekst brenda tyre qendron i qarte dhe i qendruar. Dropdown-et dhe butonat jane te bukur dhe interaktive, me efekte kur kalon miu mbi ta. Kartat dhe reshtat e tyre pershtaten automatikisht me madhesine e ekranit, duke perdorur flex ose grid, keshtu qe gjithmone duken te organizuara dhe responsive.

```
133     .empty {
134         text-align: center;
135         padding: 20px;
136         font-style: italic;
137     }
138
139     select {
140         border-radius: 15px;
141         padding: 5px 10px;
142         outline: none;
143         border: none;
144         cursor: pointer;
145     }
146
147     .forecast-grid {
148         display: flex;
149         flex-wrap: wrap;
150         gap: 10px;
151         justify-content: center;
152     }
153
154     .forecast-card {
155         min-width: 160px;
156         background: #rgba(255,255,255,0.15);
157         border-radius: 15px;
158         padding: 12px;
159         text-align: center;
160         flex-shrink: 0;
161         backdrop-filter: blur(5px);
162     }
163
164     .code-1{
165         font-size: 10px;
166     }
167     .time-code_{
168         font-size: xx-large;
169     }
170
171     .bookmark-card {
172         width: 100%;
173         max-width: 100%;
174         align-items: stretch;
175     }
176 }
177
178
179     .bookmark-city {
180         width: 100%;
181         box-sizing: border-box;
182         background: #rgba(255,255,255,0.08);
183         border-radius: 15px;
184         padding: 15px;
185     }
186
187     .bookmark-city .forecast-card {
188         min-width: 120px;
189     }
190
191     .forecast-row {
192         width: 100%;
193         display: flex;
194         flex-wrap: wrap;
195         gap: 15px;
196         justify-content: flex-start;
197         padding: 10px 0;
198         box-sizing: border-box;
199     }
200
21     .forecast-card {
22         flex: 1 1 160px;
23         max-width: 200px;
24         background: #rgba(255,255,255,0.15);
25         border-radius: 15px;
26         padding: 12px;
27         text-align: center;
28         backdrop-filter: blur(5px);
29     }
30
31
```

Hapi 10: Paraqitja vizuale ne web/Implementimi.



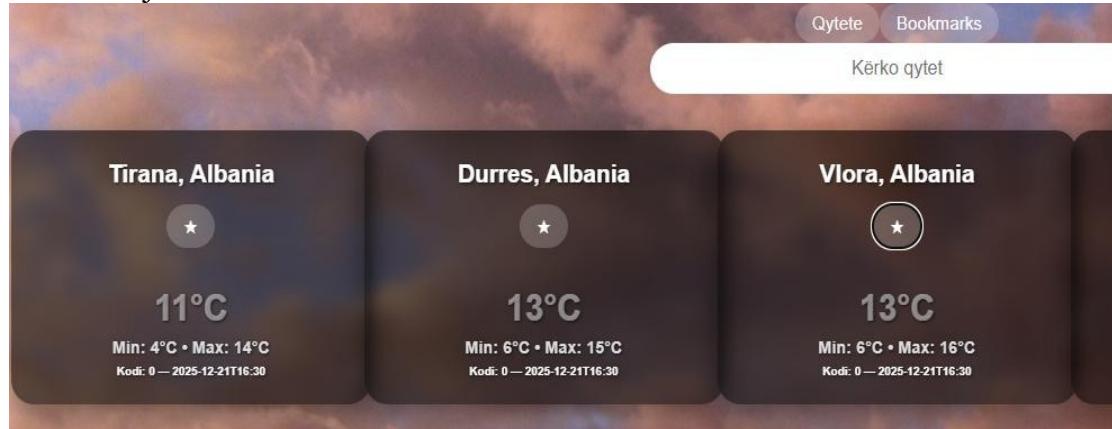
Hapi 11: Testimi.

Behet per te pare ne qoftë se funksionon sic duhet aplikacioni.

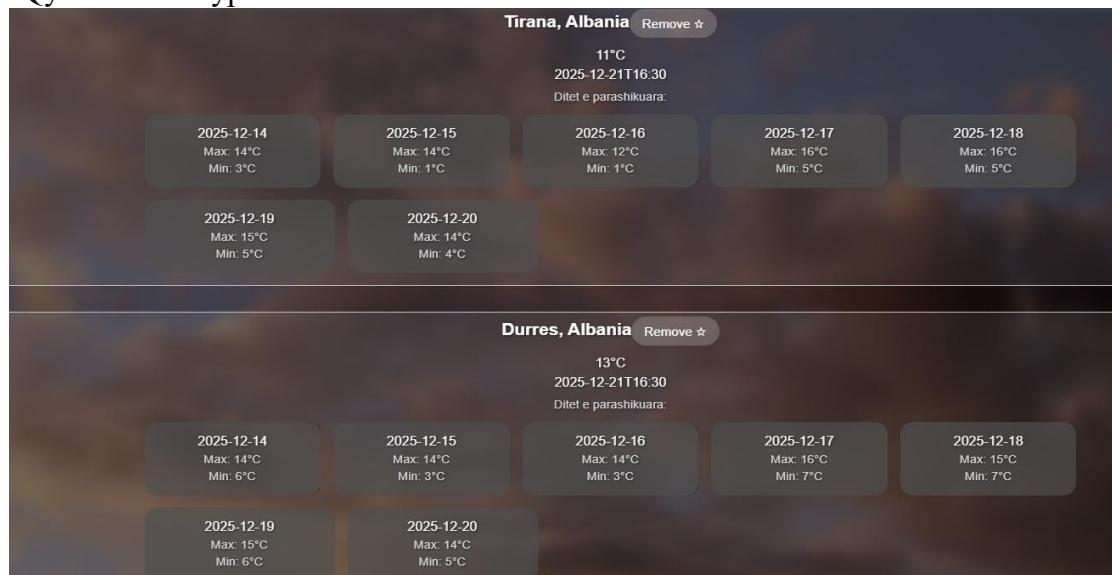
-Kur kerkoj ne input emrin Tirana:



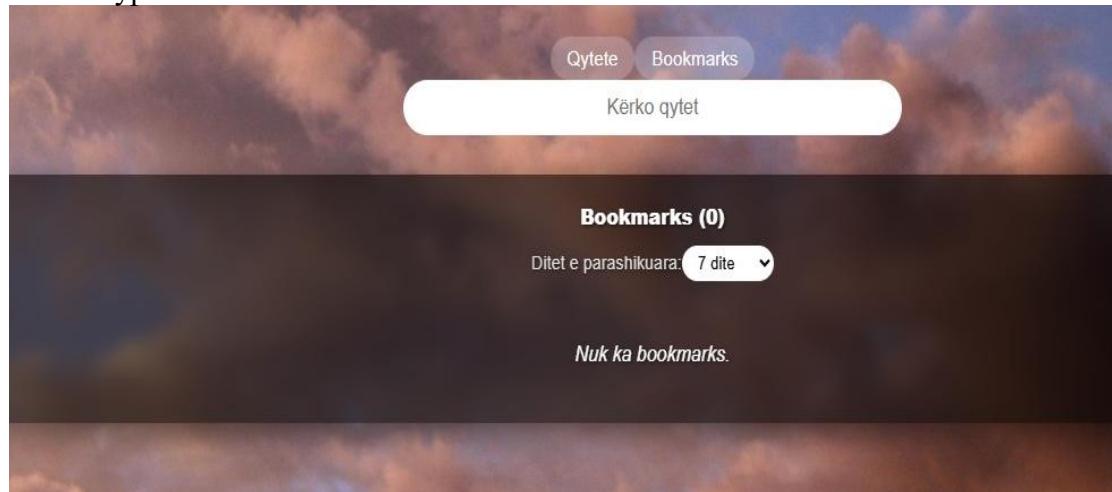
-Kur klikoj butonin favorite :



-Qytetet kur shtyp butonin favorite ruhen ne Bookmarks:



-Kur shtyp butoni remove:



-Kur selekton sa dite do te te shfaqen per parashikimin e motit:



Konkluzioni:

Ne kete detyre kursi une kam krijuar nje aplikacion ne React ,qe tregon motin per qytetet e Shqiperise duke perdorur nje API te jashtme. Gjate punes kam mesuar si funksionojne komponentet e React-it, si menaxhohen state-t me useState, si perdoret useEffect per te marre te dhena nga API-ja, dhe si te lidhen komponentet me props dhe funksione.Kam shtuar funksionalite si lista e qyteteve te preferuara (bookmarks), mundesine per te kerkuar qytete, si dhe tabs per te nderruar mes “Cities” dhe “Bookmarks”. Kam bere edhe stilizimin me CSS qe faqja te duket moderne dhe te rregullohet mire ne ekran te ndryshem.Puna ne kete projekt me ndihmoi te kuptoj me mire si funksionon React, si te punoj me te dhena dinamike dhe si te ndertoj nje aplikacion te thjeshte, por funksional.

Referencia:

- API: <https://open-meteo.com/en/docs>
- <https://react.dev/>
- <https://nodejs.org/en>
- <https://code.visualstudio.com/>
- <https://www.w3schools.com/REACT/DEFAULT.ASP>
- https://www.w3schools.com/REACT/react_components.asp