**Travel-blog**

**LIVI ÖNÁLLÓ ALKALMAZÁS létrehozása:**

-létrehozok egy mappát a gépen

vs-codeban belépek a mappába **cd mappa neve**

**-git init parancs** → létrehozom a lokális repót, létrejön a .git mappa, elindul a verziókezelés

- **npx create-react-app alkalmazás neve** react app létrehozása

- **githubon: create new repository**- összekötöm a lokális repót a githubos repóval

**- npm install react-router-dom react router feltelepítése,** ezt használjuk az útvonalválasztáshoz (package.json-ba kerül):

* importáljuk a routert App-js-be és bekötjük pl a homepage-t.

import './App.css';

import {RouterProvider, createBrowserRouter} from 'react-router-dom';

import Homepage from './pages/Homepage';

const router= createBrowserRouter([

  {

    path:'/',

    element: <Homepage />

  }

])

function App() {

  return (

    <div className="App">

      <RouterProvider router={router}></RouterProvider>

    </div>

  );

}

export default App;

* **Összegezve beimportáltunk egy függvényt amivel egy útvonalválasztót tudunk létrehozni ( createBrowserRouter) és a hozzá kapcsolódó komponenst(RouterProvider). Létrehoztunk egy routert, annak megadtam 1 szabályt (const router), hogy ha per van az url-ben akkor az App komponenst jelenítse meg, és ezt a RouterProvidernek átadtuk. Az url-tól függően fogja megjeleníteni a komponenseket.**

**- Layout (elrendezés) létrehozása:**

**import {Outlet} from 'react-router-dom';**

const Layout = () => {

    return (

        <>

            <Navigation />

            <main>

                <Outlet />

            </main>

            <Footer />

        </>

    );

}

export default Layout;

* **2 dolog kell hozzá:**
* **-Outlet komponens a layout komponensben**
* **a route szabályok egymásba legyenek ágyazva ( szülő-gyerek viszony)**
* csinálunk egy Layout komponenst, ami tartalmazza a weboldal általános szerkezetét, azokat az oldalakat amiket akarjuk hogy abba jelenjenek meg
* a **Layout komponensbe** pedig el kell helyezni az **Outlet komponenst**, az adja meg hogy hol jelenjen meg a childrenben megadott komponensek tartalma, tehát az outletet fogja kicserélni arra a komponensre amit meghívunk a childrenben

**React social-icons:**

npm install react-icons --save

import { FaFacebookF, FaInstagram, FaTwitter, FaYoutube } from "react-icons/fa";

**Footer kifixálása oldal aljára:**

.App {

    min-height: 100vh;

    display: flex;

    flex-direction: column;

}

main {

    flex-grow: 1;

}

const Layout = () => {

    return (

        <>

            <Navigation />

            <main>

                <Outlet />

            </main>

            <Footer />

        </>

    );

}

**Gombra kattintással átnavigálás új oldalra:**

 function navigateToPage(postId) {

        const url = '/post?id=' + postId;

        window.location.href = url;

    };

<button className="blog-item-button" onClick={() => navigateToPage(1)}>View post</button>

a gombokhoz hozzárendelünk egy navigateToPage JavaScript függvényt, amelynek átadunk egy postId paramétert .A navigateToPage függvény létrehozza az új oldal URL-jét a postId segítségével, majd beállítja az window.location.href értékét a létrehozott URL-re. A postId a konkrét bejegyzés azonosítója.

**Firebase:**

Itt tárolom a blog bejegyzéseket.A realtime database-ben lesz egy hatalmas javascript objektum amibe fetch hívásokat küldünk.

**Firebase projekt létrehozása:**

npm install firebase

Go to console → Add project → Megadjuk a projekt nevét → Default Account for Firebase → Create Project → ezzel kész van a projekt

Build → Realtime Database → ha még nincs adatbázisunk akkor meghelenik a Create Database gomb → kiválasztjuk hogy hol legyen, Európai helyszín kell így gyorsabb lesz a kommunikáció→

start in locked mode → ezzel kész az adatbázis, megjelenik az url

**Dropdown menu létrehozása:**

**-** Listába az egyik li-be kell egy uj ul

Feladat: Utazási blog alkalmazás

Leírás: Készíts egy webalkalmazást, amely lehetővé teszi a felhasználók számára, hogy megosszanak utazási élményeket és tippeket. Az alkalmazás segítségével a felhasználók új utazási bejegyzéseket hozhatnak létre, böngészhetnek a már meglévő bejegyzések között, és hozzászólásokat fűzhetnek hozzájuk.

Elvárt funkcionalitás:

1. Az alkalmazásnak egy kezdőoldalt kell tartalmaznia, ahol a legújabb utazási bejegyzések jelennek meg időrendben.
2. A felhasználóknak lehetőségük kell adni a bejegyzések böngészésére, és különböző országok, városok vagy régiók szerint szűrhetik azokat.
3. Az alkalmazásnak meg kell jelenítenie a bejegyzések részleteit, beleértve a címet, a szerzőt, a tartalmat és a hozzászólásokat.
4. A felhasználóknak lehetőséget kell adni új utazási bejegyzések létrehozására. Egy bejegyzés tartalmazhat címet, leírást, fényképeket és opcionálisan helyszíneket vagy látnivalókat.
5. Az alkalmazásnak lehetőséget kell biztosítania a felhasználóknak a bejegyzésekhez történő hozzászólások írására és megjelenítésére.
6. Opcionálisan, adhatsz hozzá funkciókat, mint például az utazási bejegyzések értékelése, a kedvencek mentése vagy egy térkép megjelenítése, ahol a felhasználók megjelölhetik a látogatott helyeket.