Egy webshop frontend alkalmazásának az elkészítése lesz a feladata. Az applikáció a mellékelt backend alkalmazástól tud majd adatokat letölteni. A backend alkalmazás indítás után a 8000-es porton fut, csomagok telepítését nem igényli. A backend alkalmazás az **npm start** paranccsal indítható. Talál egy kiinduló frontend környezetet is. amelybe telepítve lett a **react-router-dom** csomag, valamint a **tailwind css** is. A frontend környezet az **npm run dev** paranccsal indul. Természetesen használhat ettől eltérő css eszközöket is, de ezeket telepítenie kell. Egyes komponensekhez tartoznak felhasználandó sablonok. A feladat a sablonok alkalmazása nélkül is megoldható, de ekkor pontot veszít.

A feladat **mintak** mappájában találja az alkalmazás egyes oldalainak mintaképeit, valamint egy ábrát a backend által használt adatbázis tábláinak mezőiről és kapcsolatairól.

A feladat **src/sablonok** mappájában az egyes komponensek elkészítéséhez használandó sablonokat találja.

Az egyes komponenseket az **src/components** mappába hozza létre!

A mintában látható színek a tailwindcss előre definiált színei közül a **text-indigo-100 – text-indigo-900** színtartományból, a háttérszínek a tailwindcss **bg-indigo-100 – bg-indigo-900** színtartományból származnak. Természetesen más színeket is használhat, amelyek hasonlítanak a mintához.

1. Készítsen egy komponenst **Header** névvel! A komponens kívülről kapja meg az átadott szöveget. A komponensben készítsen egy **<H1>** elemet amely megjeleníti a komponensnek átadott szöveget, a mintának megfelelően! A komponenst töltse be az App.jsx-ben, itt adja át a tanult módszerrel a komponensnek a megjelenítendő „**Webshop**” szöveget! Állítsa be a komponens szövegszínét (**indigo-200**) és háttérszínét (**indigo-800**)
2. Készítsen egy komponenst **Main** néven! Használja fel az **src/sablonok/main.html** fájlban található sablont! Az **Assets** mappából importálja be a **BUSINESS\_SVG.svg** fájlt **KEP** néven. Ezt a képet jelenítse meg a komponens! A megjelenő szövegeket állítsa be a mintának megfelelően!
3. Készítsen komponenst **Vevok** néven. Egyelőre csak a „**A webshop eddigi vevői**” szöveget jelenítse meg a komponens. A további működést a navigáció elkészítése után kell majd megvalósítania.
4. Készítsen komponenst **UjVevo** néven. Egyelőre csak az **„Új vevő felvitele”** szöveget jelenítse meg a komponens. A további működést a navigáció elkészítése után kell majd megvalósítania.
5. Készítsen egy komponenst **Menu** néven. Használja a mellékelt sablont (**src/sablonok/menu.html**)! Az elkészített komponenst töltse be az **App.jsx**-ben! Hozzon létre navigációt az **App.jsx**-ben a következő útvonalakkal:
   * „/” – ez az útvonal navigálja a főoldalra az applikációt itt a **Main** komponens töltődjön be!
   * „/vevok” – ez az útvonal a **Vevok** komponenst töltse be!
   * „/ujvevo – ez az útvona az **UjVevo** komponenst töltse be!
   * oldja meg, hogy ha nem létező útvonalat adnak meg, akkor az alkalmazás fő oldala jelenjen meg!
   * A navigáció során ne történjen oldalletöltés!

A navigációs útvonalak létrehozása után a **Menu** komponensben állítsa be a menüpontok linkjeit!

1. A **Vevok** komponensben valósítsa meg az adatok backendről való letöltését, majd megjelenítését a mintának megfelelően. Az adatokat a <http://localhost:8000/vevok> címről tudja letölteni a komponens.

A végpont válasza egy lista, amely a következő objektumokból áll:  
**{id, nev, iranyitoszam, telepules, utcahazszam}**

Egy adat megjelenítéséhez készítse el a **Vevo** komponenst, amihez használja fel a **src/sablonok/vevo.html** fájlt. Oldja meg, hogy a **Vevo** megkapja egy járat adatait, majd a **Vevo** a mintának megfelelően jelenítse meg az egyes adatokat.

1. Oldja meg, egy **Törlés** feliratú gomb segítségével, hogy az adott Vevő törölhető legyen az adatbázisból!

A törlés címe: <http://localhost:8000/vevok/:id>, ahol **:id** paraméter, a törlendő vevő azonosítója.

A törlés után navigáljon át az alkalmazás a főoldalra! Oldja meg, hogy nagy kijelző méret esetén a vevők adatait megjelenítő **Vevo** komponens négy oszlopba, közepes kijelző esetén két oszlopba, kisméretű kijelző esetén egy oszlopba rendezve jelenjen meg.

1. Az **UjVevo** komponensben valósítsa meg, hogy egy form segítségével fel lehessen venni egy új vevőt az adatbázisba! A form létrehozásához használja fel a **src/sablonok/ujvevo\_form.html** fájlt! A backend megfelelő POST végpontja a <http://localhost:8000/>vevok címen érhető el. Az adatfelvitel után a komponens navigáljon a főoldalra!

A küldendő adatok: **{nev, iranyitoszam, telepules, utcahazszam}**, az id autoincrement tulajdonságú.

1. A megfelelő parancs használatával készítsen buildet az alkalmazás ból!