# 2., Reszponzív viselkedésű weboldal készítés és formázás

A következő feladatban egy weboldalt kell készítenie a napelemek rövid bemutatására a feladatleírás és a minta szerint.

Az első feladatrészben a forrásként kiadott weboldalon kell módosításokat végeznie a leírás és a minta alapján! Nyissa meg a napelem.html állományt és szerkessze annak tartalmát az alábbiak szerint:

1. A weboldal karakterkódolása utf-8, nyelve magyar, a böngésző címsorában megjelenő cím „Napelemek” legyen!
2. A weboldal fejrészében helyezzen el hivatkozást a napelem.css stíluslapra a meglévő azonos típusú hivatkozások után!
3. A weboldal fejrészében helyezzen el hivatkozást napelem.js állományra a meglévő azonos típusú hivatkozások után!
4. Bővítse ki a SlideShow-t a pictures mappában található kep3.jpg állománnyal. A képen megjelenő alternatív szöveg és cím „napelem3” legyen.
5. A weboldal navigációs sávjában az utolsó pont („Kinyerhető teljesítmény”') után helyezzen el egy új hivatkozást a „Megfelelő tájolás” blokkhoz is a többi hivatkozáshoz hasonlóan!
6. A weboldalon keresse meg a Dunasolar kifejezéseket, majd alakítsa a kifejezéseket hivatkozássá! Kattintásra a hivatkozott weboldal új ablakban/böngészőfülön nyíljon meg. A céloldal URL címe: [*https://hu.wikipedia.org/wiki/Napelem#A\_Dunasolar\_öröksége*](https://hu.wikipedia.org/wiki/Napelem#A_Dunasolar_öröksége) *.*

A hivatkozás elérhető a hivatkozas.txt állományban.

1. *A weboldalon készítsen egy újabb tartalmi blokkot az alábbi leírás és a minta alapján:  
   a. Az új tartalmi blokk a Bootstrap rács második sorában, a „Napelemek fajtái” blokkja után helyezkedjen el! A sor blokkjainak (oszlopainak) szélességét 8:4 arányban ossza el! Az új blokkot formázza a magyarazat azonosítókijelölővel!  
   b. A blokkba helyezze át a „Napelemek fajtái” blokkból a felsorolást! Alakítsa át a felsorolást számozott felsorolássá!*
2. *Nyissa meg a napelem.css állományt, módosítsa a következők szerint:*

*a. A 2-es szintűcímsorok szövegei legyenek nagybetűvel írva!*

*b. Az igazitas azonosítóval ellátott elem szövege vízszintes igazítását módosítsa sorkizártra!*

***Minta:***



A képen szöveg látható

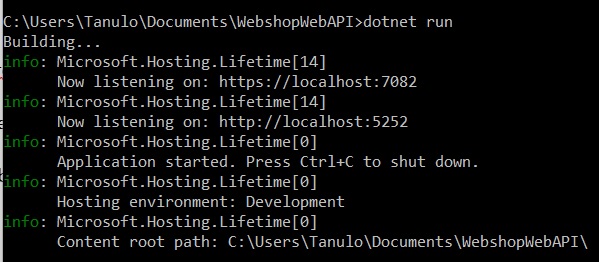
Automatikusan generált leírás

A képen szöveg látható

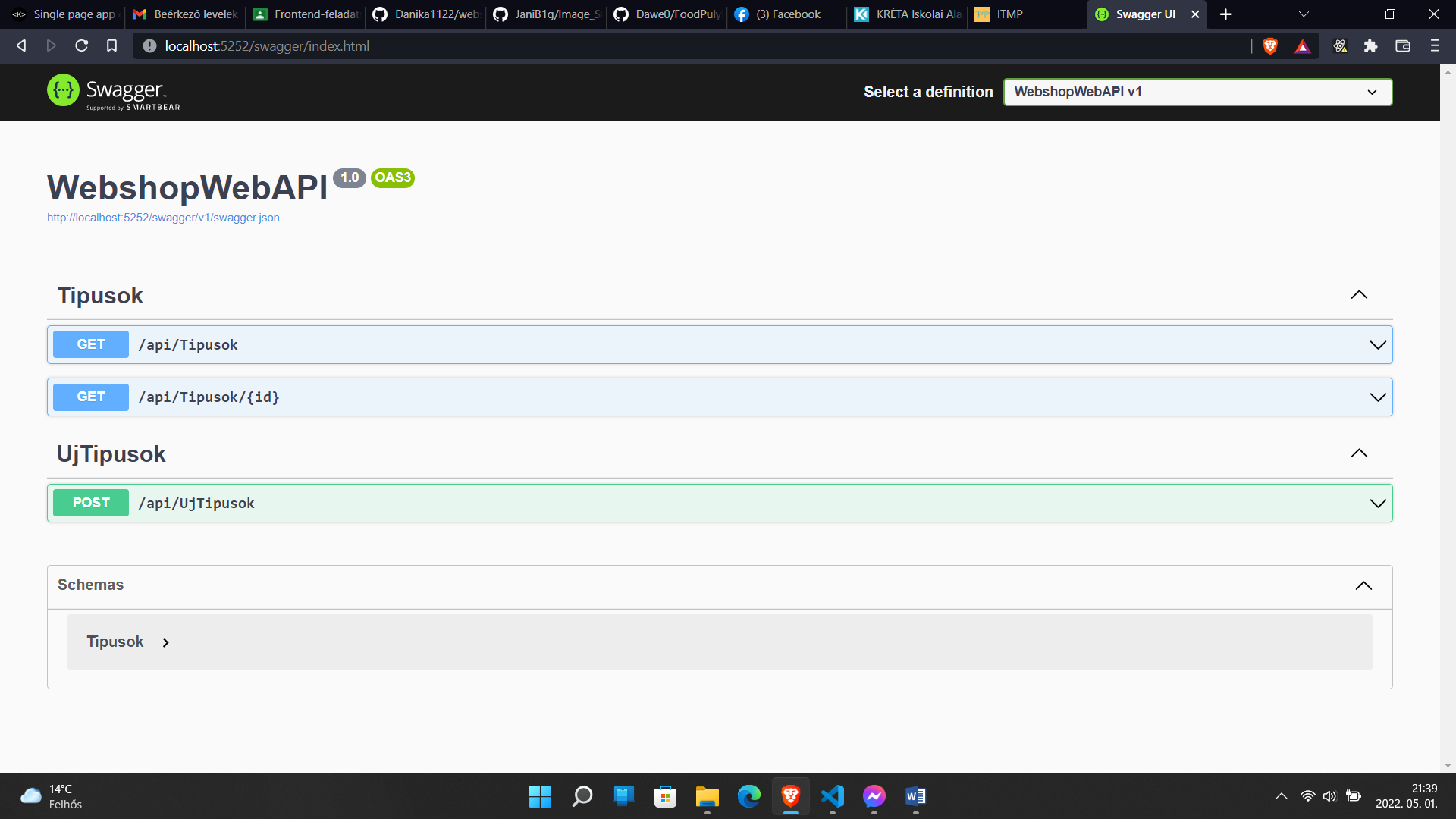
Automatikusan generált leírás

# 4. Frontend programozás

A következő feladatban egy frontend alkalmazást készítését kell elvégeznie a kiadott leírás  
szerint! Az alkalmazás felhasználói felületének kialakítását minták segítik, melyeket nem kellkövetnie a megoldásában. Indítsa el a projektkönyvtárának backend mappájából a weboldalát kiszolgáló szervert (WebshopWebAPI.exe)! A program indulása után az alábbi képernyő jelenik meg:

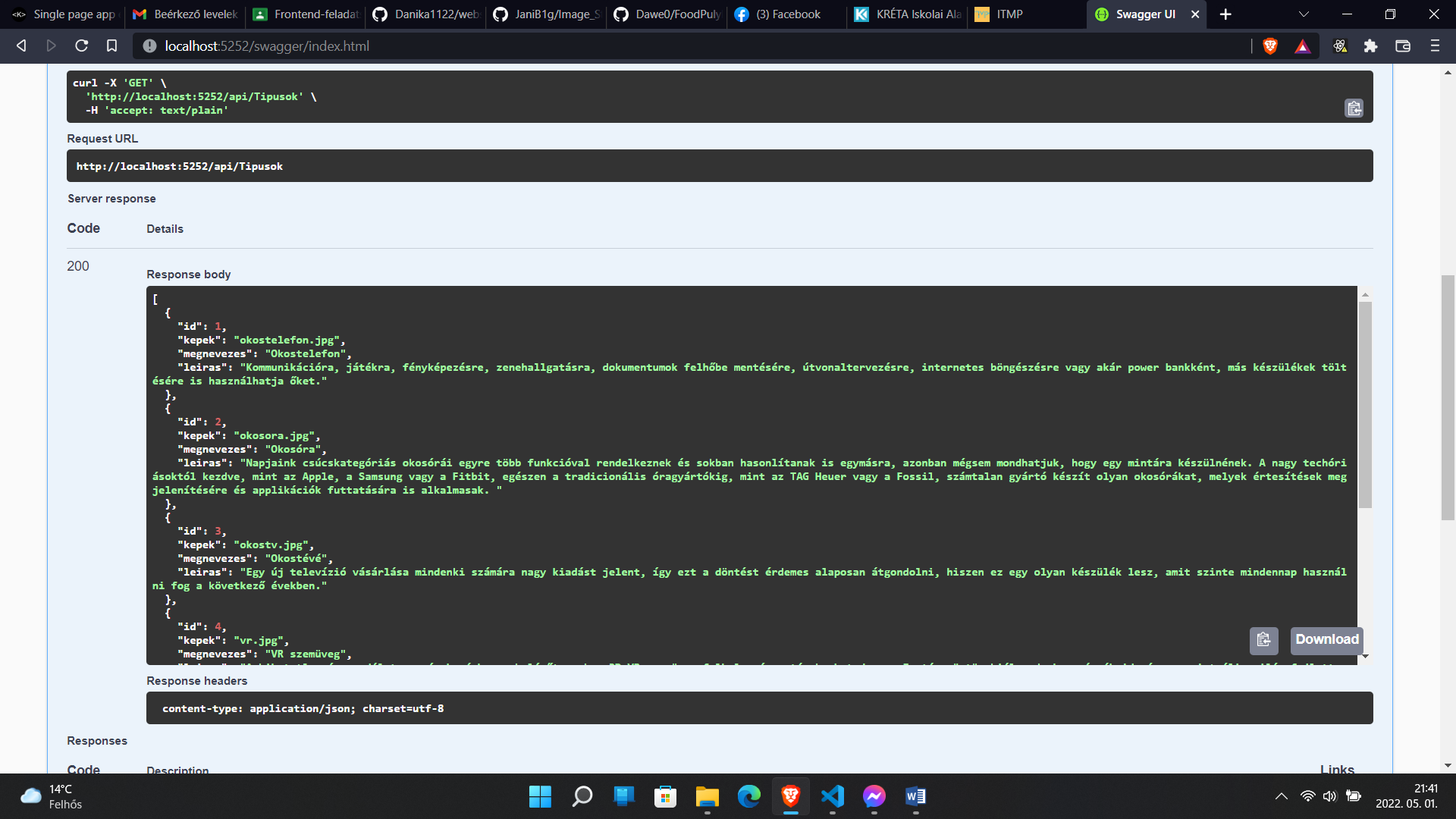


A mintán látható „Content root path” a program indításának a helyét mutatja. A webkiszolgáló a https://localhost:7082 címen fog elindulni. A böngészőbe beírva a <https://localhost:7082/swagger/index.html> URL-t a kap segítséget a webszerveren elérhető REST API függvényekhez.



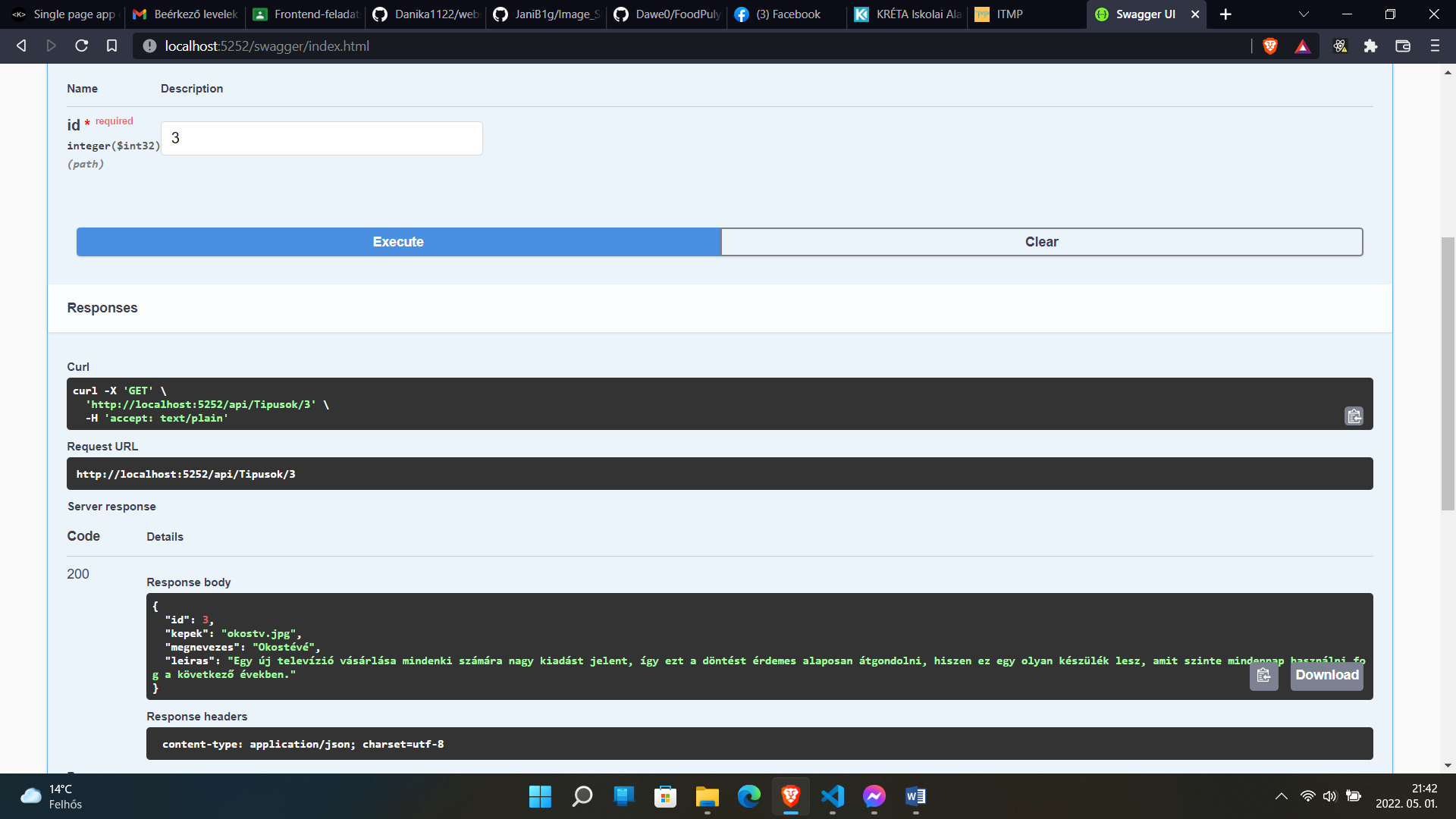
Az alábbi REST API függvények érhetőek el a szerveren:

* /api/Tipusok
* Metódus: GET  
  Válasz: Az adatbázisban lévő típusok azonosítói, megnevezései, képnevei és leírásai.  
  Válasz minta:



* /api/Tipusok/{id}

Metódus: GET  
Válasz: Az adatbázisban lévő egyetlen típus azonosítója, megnevezése, képneve és leírása.

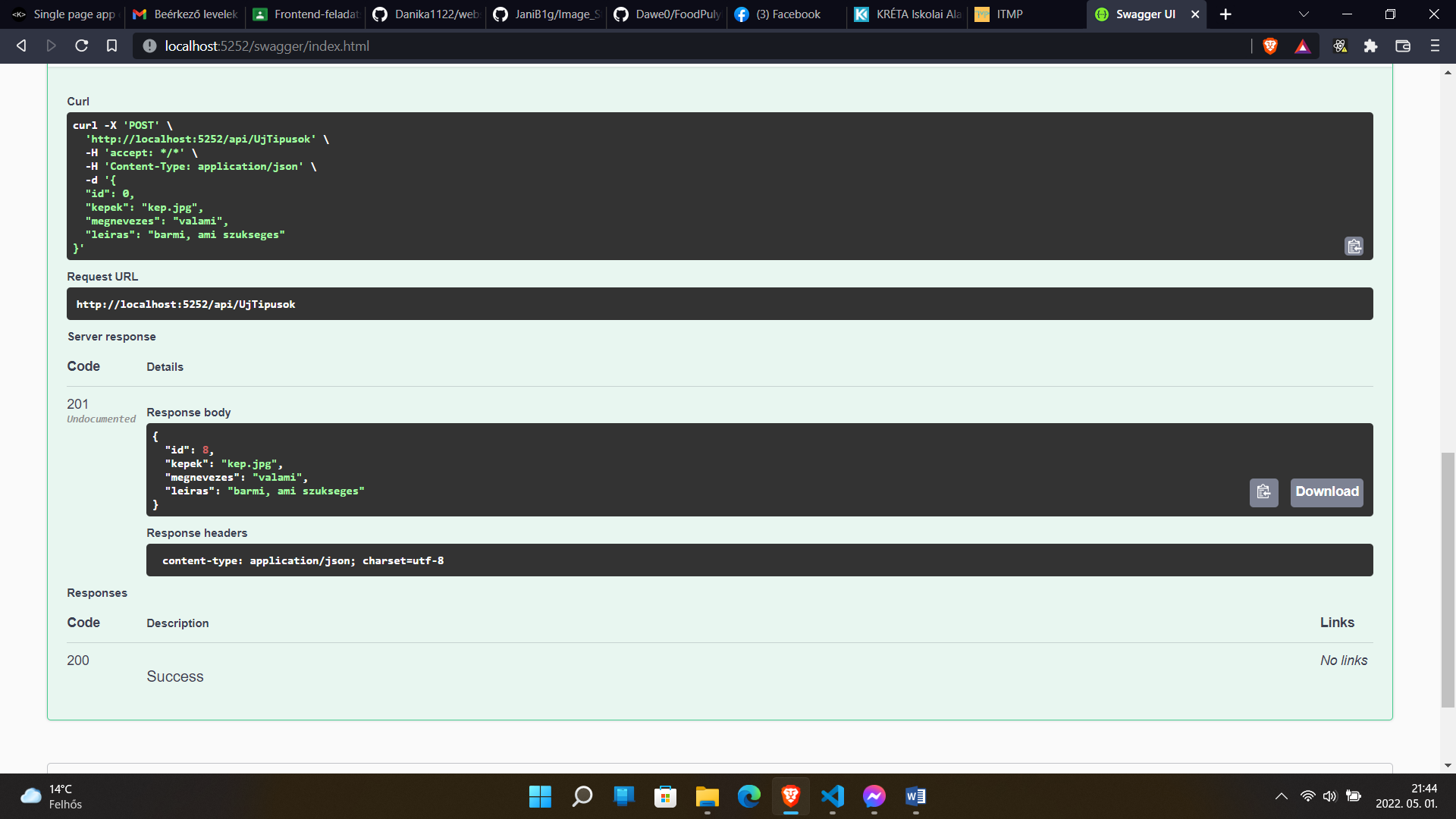


* /api/UjTipusok

Metódus: POST

Válasz: A kérés törzsében kell a szerver felé elküldeni a rögzíteni kívánt új típus adatait!

Egy lehetséges POST metódus:



FONTOS! A serverhez nem tartozik fizikai adatbázis, a program újraindításakor az adatok „visszaállnak” az eredeti állapotukba!

# Feladatok:

Az Ön által tanult Javascript keretrendszer (React) segítségével készítse el az alábbi feladatokat:

1., Adja hozzá az alkalmazáshoz a választott UI keretrendszer (pl.: Bootstrap) legfrissebb  
verzióját!

2., Hozzon létre komponenst a típusok megjelenítéséhez!

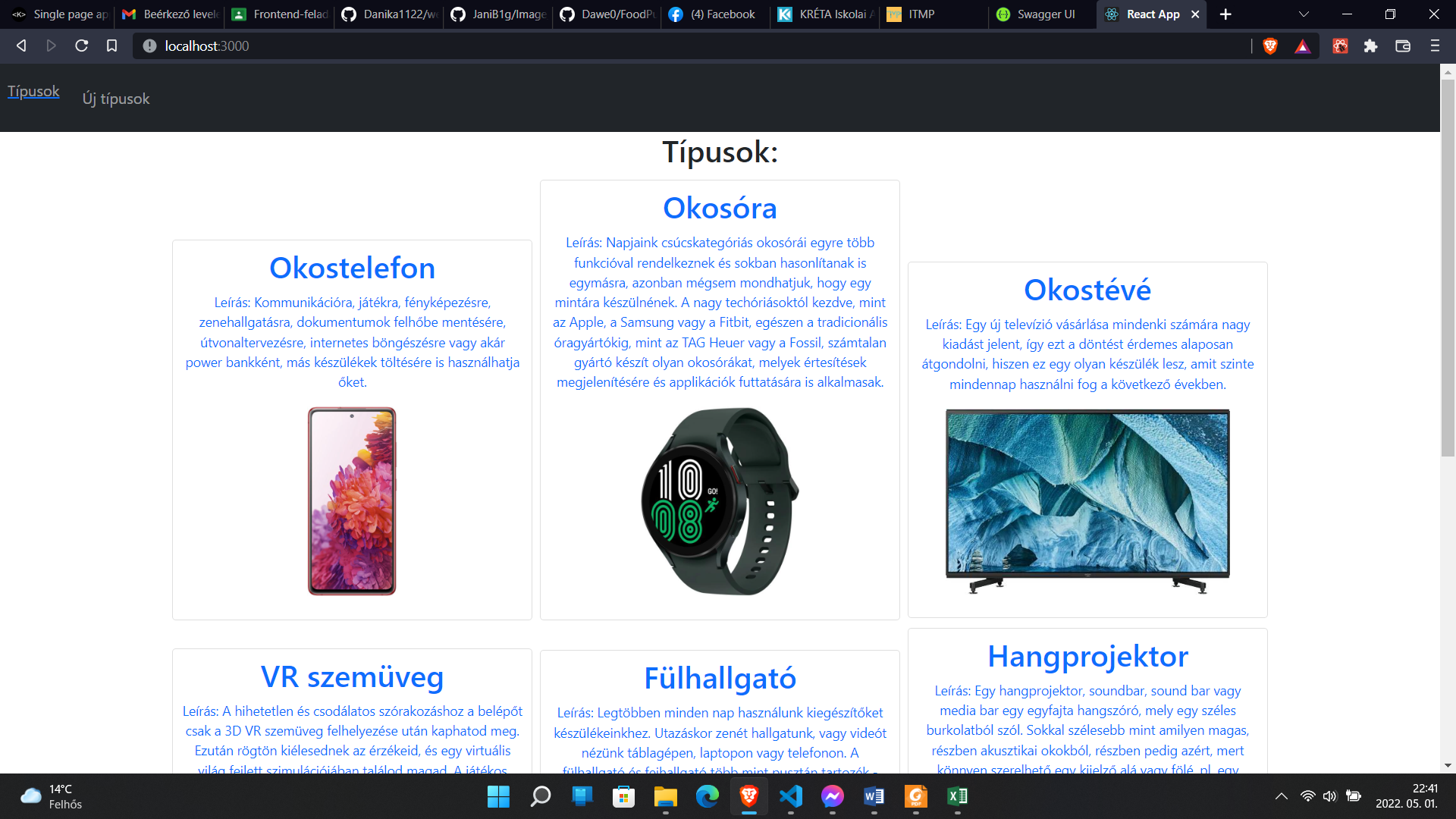
a., Az oldalon szereplő típusokat tartalmazó adatokat a kiszolgáló szerver  
/api/Tipusok URL-jén szereplő REST API függvénnyel kérdezze le!

b., A vonatkozó képek neveit a backend-ből kapja, ugyanakkor a frontend-ben statikusan elérhetők a képek a public/images könyvtár alól. A szükséges linkeléseket tegye be az oldalba!

c., A lekérdezéshez fetch api-t vagy axios-t használjon! A kártyák sorrendje a mintának megfelelő legyen!

3., Ha rákattintunk egy termékre, akkor az a termék egy külön oldalon jöjjön be, pl. /tip/1 alatt

a., Az oldalon szereplő típust tartalmazó adatokat a kiszolgáló szerver  
/api/Tipusok/{id} URL-jén szereplő REST API függvénnyel kérdezze le!



4., A kezdőoldalon legyen egy gomb, Új típusok névvel, arra kattintva navigáljon el a /uj-tip útvonalra!

a., Új típus felvételénél az adatok átküldése után navigáljon vissza a főoldalra.

b., Az oldalon szereplő új típushoz a tartalmazó adatokat a kiszolgáló szerver  
/api/UjTipusok URL-jén szereplő REST API függvénnyel küldje el!

5., Forráskódját a React-nak megfelelően adja le:

a., tömörítés előtt törölje a felesleges fájlokat (pl. node\_modules)!

b., Ügyeljen arra, hogy a feladatmegoldást tartalmazó mappát ne töröljön!

c., A forráskódját tömörítse be Vezetéknév\_Keresztnév\_frontend.zip néven!