



HÁZI FELADAT

Kliensoldali rendszerek

Rajacsics Tamás

2024.

Verziótörténet:

1.0	2024.09.23.	Első verzió
1.1	2024.10.26	Házi specifikáció leadás pontosítása, határidő kitolása 9. héttig, backend szerepe

BEVEZETÉS

Az aláírás megszerzésének feltétele a házi feladat határidőre történő beadása és legalább elégsges szint elérése. Az alábbiakban összefoglaljuk ennek részkövetelményeit

TÉMAVÁLASZTÁS

Egyedi feladatot kell készíteni. A házi feladat rövid, 5-10 mondatos specifikációját a (8.) 9. oktatási hét végéig Moodle-be fel kell tölteni a „Házi feladat specifikáció” beadandónál, **vagy** más csatornán (Teams, email) egyeztetni gyakorlatvezetőddel.

Ha nincs ötleted, akkor íme néhány:

- Kalkulátor
- Bevásárló lista
- Növénylocsoló kezelő
- Fitness App
- Recept App
- E-book olvasó
- Mini játék (2048, tetris, pacman, stb.)

TECHNOLÓGIA

A feladatokat a tárgyban oktatott React technológiával kell megvalósítani.

KIDOLGOZÁS

A házi feladat kidolgozása a félév során **önálló** feladat. **Az elkészült feladat másról való másolása és másokkal történő megosztása egyaránt tilos.** Az újrahasznált/másolt kód részleteket automatizált eszközökkel vizsgáljuk, **bizonyítható plágium esetén minden érintett fél háziját érvénytelennek tekintjük és az aláírást megtagadjuk.**

BEADÁS

Az elkészült házi feladat forráskódját a tanszéki portálon keresztül fel kell tölteni, majd az utolsó gyakorlaton a gyakorlatvezetőnek az alkalmazást működés közben bemutatni. A demonstrációhoz a laborgép, illetve saját számítógép (laptop, távoli asztal stb.) egyaránt használható. A bemutató során a laborvezetőnek be kell mutatni:

- A működő alkalmazást
- Az alkalmazás forráskódját

A gyakorlatvezető a pontozást helyben végzi, ehhez a beadáskor kérheti a forráskód értelmezését és akár kisebb módosítását is.

A pótbéadáson (melyet a pótlási héten szerdán tartunk) lehetőség van azok számára is részt venni, akik az előző beadáson elért pontszámukat (az alkalmazáson időközben elvégzett módosításokkal) javítani szeretnék.

PONTOZÁSI RENDSZER

A gyakorlatvezetők a beadott feladatokat az alábbi pontozási rendszer szerint értékelik. Fontos, hogy (amint az a lenti pontozásból látszik) az értékelés során kizárolag a frontendet releváns, a házi feladat lehet olyan alkalmazás is, ami nem igényel backendet, vagy a backendet külső szolgáltatás (pl. valamelyen online szolgáltatás publikus API-ja) adja.

Követelmény	Elérhető pontszám
Funkcionalitás	7
Komplexitás	5
Platform konvenciók követése	2
JS API-k használata	2
Kommentezés	2
Design, ergonómia	2

A pontozási szempontokhoz az alábbi útmutatót adjuk. Az értékelők a teljes skálán értékelik a feladatokat, a lenti példák az egyes pontszámokért elvárt tartalmat illusztrálják.

FUNKCIONALITÁS (0-7)

- 0 – Az alkalmazást nem sikerült működés közben bemutatni, vagy a forráskódja nem elérhető
- 1 – Az alkalmazás elindul, de alapvető funkciók működésképtelenek benne
- 2-3 – Az alkalmazás elindul, egy-két fontos funkció nincs megvalósítva
- 4-5 – Az alkalmazás elindul, egy-két funkció bemutatása során hibára fut (vagy a funkció lefut, de a háttérben vannak pl. Javascript futtatási hibák, vagy adatkötési hibák)
- 6-7 – Az alkalmazás teljes mértékben megvalósítja a feladatkiírásban foglalt funkcionális követelményeket

KOMPLEXITÁS (0-5)

- 1 – az alkalmazás csak pár komponensből áll
- 2 – az alkalmazás legalább 10 komponensből áll
- 3 – az alkalmazás legalább 10 komponensből áll és helyesen használ useState és useEffect hookokat.
- 4-5 – az alkalmazás legalább 15 komponensből áll és helyesen használ legalább három fajta hook-ot (useState, useEffect, useContext, useRef, useCallback, useMemo, vagy megvalósít saját hook-ot, és azt használja)

KÓD SZERVEZÉSE (0-2)

- 0 – Nincs semmi logika a kód szervezésében
- 1 – Van logika, de a komponensek rossz helyen vannak elválasztva, bizonyosak túl nagyok, mások értelmetlenül kicsik (nem csinálnak semmit)
- 2 – Komponensek külön fájlban vannak egyesével, vagy ha több is van egy fájlban, akkor csak egy exportált. Jól van strukturálva a kód.

JS API-K HASZNÁLATA (0-2)

A következő JS API-kat lehet használni:

- Local Storage, Session Storage vagy Indexed DB

- fetch
- requestAnimationFrame
- Audio, vagy Video API
- Location API
- PWA Share target
- PWA push notification

Ezek használata a következő pontokat éri.

- 0 – egyiket sem használja
- 1 – egyet használ
- 2 – többet használ

KOMMENTEZÉS (0-2)

Fontos, hogy a kódban a nyilvános (export) függvények és típusok a platform konvenciónak megfelelően kommentezve legyenek (TypeScriptben TSDoc), illetve a komplexebb függvények logikájának megértését is külön kommentek segítsék. Alacsony komplexitású házi esetén (kevesebb mint 15 kommentezett függvény) pont nem adható.

- 2 – Saját készítésű publikus elemek 90%-a kommentezett platformkonvenciók szerint
- 1 – Saját készítésű publikus elemek 50%-a kommentezett (platformkonvenciókat nem minden követve)
- 0 – Kevesebb mint 15 függvény van kommentezve, vagy platformkonvenciókat nem követi a kommentezés

DESIGN, ERGONÓMIA (0-2)

- 0 – Az alkalmazás alapvető ergonómiai/esztétikai hibákat tartalmaz (pl. túl kis gomb/közeli gombok az érintőképernyőn, nem elég kontrasztos színek, visszajelzés elmaradása stb.)
- 1 – Az alkalmazás szép, funkciói letisztultak, de nem követ különösebb UI irányelvet
- 2 – Az alkalmazás szép, használ ikonokat és/vagy követi valamely nagy szoftvergyártó UI konvencióit. Pl. [Material design](#). Ennek érdekében külső osztálykönyvtárak felhasználhatóak.

IMSc

IMSc pontozás esetén az elvárás, hogy a házi feladat komplexitása elérje a 25 komponenst. Az IMSc pontszám azonos lesz az alap pontszámmal. Amennyiben IMSc pontot is szeretnél kapni, ezt bemutatáskor jelezd gyakorlatvezetődnek!