

Webfejlesztési keretrendszerek – 2020 tavasz

Kötelező program

Általános elvárások: A feladat egy téma alapján egy kliens oldali alkalmazás elkészítése az Angular 2+ keretrendszer használatával. Az elkészült alkalmazásnak DEMO jelleggel kell működnie, valós szerver és adatbázis használata nem követelmény. A bejelentkezés és a megjelenített adatok mock-olt jelleggel működjenek! Elvárt az alkalmazás megfelelő módon való komponensekre bontása, a responzivitás és a mai modern, mobile-first UI elemek használata (Angular Material). Az Angular Routing segítségével a funkcionalításban eltérő komponensek legyenek megkülönböztetve (pl.: login komponens más URL-en keresztül érhető el, mint a home page)! A több komponenst kiszolgáló funkciók és az esetlegesen szerverrel kommunikáló részek legyenek Service-ekbe kiszervezve! A komplexebb objektumokhoz készíts interfészeket (pl.: User), amelyben rögzítjük az adattagokat és a típusokat! A alkalmazás végső verziója legyen valahol host-olva (pl.: Firebase vagy Heroku)! Ezen keresztül történjen meg a bemutatás!

Cím	Leírás
<i>smallBay</i>	Egy aprócska ebay koppintás, ahol egy listában termékek vannak felsorolva képekkel, a termékek típus szerint kategorizálhatóak, frissesség, aktuális ár, aukció dátumának lejárata szerint állíthatóak sorrendbe. Egy terméket kiválasztva annak oldalára kerülünk, ahol bővebb leírás olvasható, további képek láthatóak, kérdés tehető fel az eladónak, illetve lehetőség van licitálásra, azonnali leütésre. A működéshez elég, ha egy localStorage-ban tárolt változó beállítása alapján megkülönböztetjük a vásárlót az eladótól, a megjelenített termékeket egy .json fájlban tárolhatjuk, a hozzájuk rendelt képeknél pedig ugyanaz a három-négy kép ismétlődik
<i>80s Hacking Style</i>	Webalkalmazás, amely egy régi, 80-as évekbeli film "hacker programját" szimulálja, fekete háttérrel, zöld szegélyekkel és színekkel (a la Mátrix filmek), retro betűtípussal, kizárólag nagy betűkkel, olyan szellemes menüpontokkal és opciókkal, amelyek jelszófeltörést és hasonlót ígérnek. A jelszótörő felületen a háttérben egy service induljon el, amely néhány másodpercre elalszik és csak utána tér vissza egy random karaktersorral, amíg fut, addig a Component-ben a jelszót jelölő szövegmezőben folyamatosan pörögjenek random generált karakterek
<i>MyTourGuide</i>	Utazó blog felület, a fő képernyő nagy részét az alap opciók (profil, kijelentkezés, beállítások) mellett egy térkép töltse ki, amelynek egy tetszőleges pontjára kattintva dialógusablak ugrik fel, amelybe képek, emlékek, dátumok tölthetők fel egy utazással kapcsolatban azon a helyen. A dialógusablak bezárása után a kattintott helyen jelöljön meg egy kis jelölő pont, amelyre kattintva megjelennek a bevitt adatok. A térkép megjelenítéséhez tetszőleges megoldás használható
<i>AngularQuest</i>	Angularban megvalósított szöveges kalandjáték a DepressionQuest mintájára, a döntések eredményeit, állapotok tárolását egy service végezze a sessionStorage-ba, a képernyő egyes részei testvérkomponensekből álljanak, amelyek egy szülő komponensen keresztül kommunikálnak egymással. Tetszőleges történet, ötlet implementálható, a történet kimenetele mindenképp valamilyen külső erőforrás fájlban, .json kiterjesztéssel szerepeljen a projektben http://www.depressionquest.com/dqfinal.html
<i>CodexEditor</i>	Egy középkori könyvtár szimuláló Angular webalkalmazás, meglevő

	kötetek is kiválaszthatóak egy könyvespolcra olvasásra, vagy egy új hozható létre. A menüben nem, viszont a kötet megnyitásakor a kötet szövegét megfelelő kódexeket idéző gótikus betűstílussal legyenek megvalósítva, a könyvek legyenek .json formátumban exportálhatóak és letölthetőek is.
MusicLibrary	Zenelejátszó - és kategorizáló webalkalmazás, zenéket soroljon fel egy listában előadó, dátum, műfaj szerint szűrhető formában, egy zenére kattintva navigáljon át annak adatlapjára, ahol további információk mellett a teljes szám vagy annak egy része is lejátszható lejátszó gombok kattintgatásával
SpeakToMe	Szöveglista, amelyek dátum és címkék szerint szűrhetőek egy listából, a kiválasztott szöveg egy gomb lenyomása után egy text-to-speech library segítségével legyenek lejátszhatóak/felolvastathatóak is
Username666	A híres Youtube-os kísértettörténet implementációja, a kezelőfelület egy klasszikus Youtube csatornát szimuláljon, viszont a háttérben a user minden navigációja, kattintása legyen regisztrálva egy számlálóval, ahogy a számláló növekszik és a user navigál a profil különböző komponensei, oldalai között, az oldal vegyen egyre kísértetesebb megjelenést - torzuljanak a szövegek, essen szét a layout, lassan vegyenek fel vörös, véres textúrákat az egyes elemek, a legvégső szinten a teljesen horrorisztikus megjelenés mellett a háttérben sikolyok, kísérteties hangklippek játszódnak le
FotoGallery	Nagyfelbontású képgaléria egy fotós alkalmazáshoz, amelyben egy blog/hírfelület és egy kapcsolat felület mellett a fő része a galéria, ahol dátumok, rendezvények szerint elkülönítve nyithatóak meg a képsorok, ennél a projektnél kiemelt fontosságú a rezponzivitás, illetve a demóhoz legyen több nagyobb felbontású kép is felhasználva. Legyen továbbá egy minta felület, amelyen a gépről több fényképet lehetne kiválasztani és feltölteni (a tallózás, képek kijelölése működjön, a tényleges feltöltésnek értelemszerűen nem muszáj)
InfBox	Egy felhő alapú fájl szinkronizáló elkészítése a feladat. A Dropbox SDK-jának segítségével szeretnénk karbantartani Dropbox mappáinkat. Az adott account-hoz tartozó Dropbox tárhely menedzselhető, fájlok feltölthetőek, letölthetőek, törölhetőek.
TextEditor	Egy online szövegszerkesztő elkészítése a feladat. Ennek megvalósításához használható bármilyen meglévő modul (pl.: CKEditor). Az elkészült, megszerkesztett dokumentumból tudjunk PDF-et generálni, azt lementeni, illetve feltölteni bármilyen felhő alapú tárhelyre API használatával (pl.: Dropbox, Google Drive, stb.)
myimdb	Az IMDB egy közismert szolgáltatás, amely valamennyi filmről, színésről információt nyújt a felhasználónak. A feladat, az IMDB adatbázisának segítségével egy hasonló felület elkészítése. A lekérdezésekhez használjuk az IMDB API-ját, és a lekért adatokat jelenítsük meg egy igényes Material Design elkészítésével. Az alkalmazás listázza ki a legnépszerűbb (pl.: top 50) filmet a főbb kategóriákban (vígjáték, akció, thriller, horror, dráma, romantikus), jelenítse meg a főszereplőket (a szereplő listában az első 5-öt).
MovieComparator	Készítsünk egy filmösszehasonlító oldalt, ahol két tetszőlegesen kiválasztott filmet tudunk összehasonlítani. Az adatok megjelenítésére ötletként használható az IMDB, azonban az adatokat elegendő lokálisan egy JSON fájlban tárolni. A fájl beolvasása történjen Service-en keresztül. Az összehasonlítás során figyeljünk arra, hogy kizárólag azonos kategóriában szereplő filmeket hasonlítsunk össze, és a magasabb értékelést kapott filmet jelöljük jól látható módon (pl.: zöld kiemelés). Az összehasonlítás

	eredményeképp jelenítsük meg a film főszereplőit (első 5 a listában), a film hosszát, rendezőjét, és rövid leírását. Minden filmhez tartozzon egy borítókép is, ezt is jelenítsük meg.
MovieCategoryComparator	A feladat egy olyan web-alkalmazás elkészítése, amely különböző filmkategóriákat vet össze egymással. Az alapötlet hasonlít az IMDB megvalósítására, azonban itt nem konkrét filmösszehasonlítást végzünk, hanem a kategóriákba tartozó filmek értékelései és adatai alapján kategóriákat. Az eredmény néhány diagram lenne, amely ezen adatokból állítható elő. Az adatokat elegendő lokálisan egy JSON fájlban tárolni, és azt Service-en keresztül beolvasni. A tárolt adatok a következők legyenek: film címe, kategóriája, értékelése, szereplők listája, rendező, film hossza. Ahhoz, hogy a diagramok érdekesek legyenek, minden filmkategóriában legyen legalább 15 film felvéve. A statisztika érintse az alábbi kategóriákat: vígjáték, akció, thriller, horror, dráma, romantikus. A diagramok elkészülhetnek az alábbi statisztikákról: filmek átlagos hossza adott kategóriában, értékelések átlaga adott kategóriában, leghosszabb filmek a kategóriákban, legrövidebb filmek az adott kategóriákban, legrosszabb értékelés az adott kategóriában, legjobb értékelés az adott kategóriában.
SzegedTransport	A feladat egy olyan web-alkalmazás elkészítése, amely a szegedi tömegközlekedési adatbázis alapján (linkelve) kilistáz a szűrésnek megfelelő járatokat. Az adatok CSV formátumban megtalálhatóak. Ezeket kell beolvasni, és a beolvasott adatokat feldolgozva kilistázni a járat opciókat a felhasználónak. Pl. tudjunk időintervallumot megadni, érintett helyszínt, akár vonalszámot (1-es villamos) megadni. Egy feltétel megadásával a további szűrési feltételek is aktualizálódnak (pl.: ha kiválasztom érintett állomásnak az Európa Liget-et, akkor ne tudjak 4-es villamos választani). Adatbázis: http://members.upcweb.hu/cseh.attila/google_transit.zip
SzegedTransportMap	A feladat egy olyan alkalmazás elkészítése, amely a szegedi tömegközlekedési adatbázis adatai alapján (linkelve) egy választott járat útvonalát megmutatja, térképre illeszti, és a megállókat marker-rel megjelöli a térképen. A térkép integrációhoz használható a Google Maps, MapKit JS, Mapbox.js, Leaflet.js, stb. Adatbázis: http://members.upcweb.hu/cseh.attila/google_transit.zip
Browser Cinema	Online mozi, ahol lehetőség van egy filmlistában keresgélni, a film adatlapjára kattintva pedig vetítésre feliratkozni - ha eljön a vetítés időpontja, a böngésző értesítést jelenít meg a felhasználónak és lehetősége van átnavigálni egy vetítőterembe, ahol egy beágyazott livestream formájában elindul a film (a demó erejéig bármilyen livestream beágyazható)
Online Arcade	Egy régi 80-as évekbeli játéktér hangulatát idéző webalkalmazás háttérzenével, ahol év, stílus, műfaj szerint lehet keresgélni régi játékok között és értékelni őket, hozzászólásokat írni az adatlapjukhoz. Legalább egy játéknál legyen lehetőség átnavigálni egy CabinetComponent nevű komponensre, amelyben egy játéktérmi automata designjába ágyazva ki lehet próbálni az adott játék HTML5 - Canvas változatát (lehet Tetris szintű apróság is, githubon szép számmal akadnak ilyen böngészős portok klasszikusokból)
No Adblock Please	Böngészős játék, ahol a feladat az oldalon belül elnavigálni egy megadott gombhoz/kattintható képhez, amelyen kattintva felugrik egy "Nyertél!" dialógusablak. Minden más elem csak gif, villogó gomb, link, amelyek jobb esetben csak felugró dialógusablakokat és oldalon belüli navigációkat eredményeznek, rosszabb esetben átnavigálást online kaszinó oldalra, stb
Alice in Browserland	Webshop, amelyben minden gomb, minden menüelem különböző

	méretű, stílusú (erős ngClass és ngStyle használat), viszont minden kattintás a megszokott, elvárható navigáció, funkcionalitás mellett megváltoztatja pl. a betűtípust, méretet, háttérszínt, szövegszínt, gombok, feliratok méretét
Mario Needed Money	Kisköltségvetésű, általános iskolásoknak készített oktató / vetélkedő oldal, látványosan más, ismertebb játékokból lopott grafikus elemekkel (pl. Super Mario, mint oktató) a különböző komponenseken matematikai, értő olvasási feladatokat kell megoldani, amelyek megoldásáért a felhasználó aranyérméket gyűjtöget, amelyek localStorage-ban tárolódnak, bónusz esztétikai elemnek számít, ha a háttérben valamilyen low budget MIDI zene szól, a háttérrek, menüelemek pedig MS Paint-ben készültek a tökéletes összehátas érdekében
EarthSaver	Közösségi hálózat demó a Twitter mintájára, melynek a célja a környezettudatosság promotálása, a felhasználók ezzel kapcsolatban posztolhatnak, követhetik, megoszthatják a saját károsanyag-kibocsátásukat vagy spórolásukat, részt vehetnek promóciókban, megmozdulásokban
NewsFlash	Angular webalkalmazás, amelynek elrendezése egy tetszőleges hírportál dashboardját másolja, képekkel, lead szövegekkel, megnyitható, továbbolvasható szövegekkel, kategóriák szerint rendszerezve. A dashboardon legyen hely kihagyva hirdetőfelületeknek, bannereknek, a jobb sávban a legolvasottabb cikkek, friss hozzászólások listája elérhető (ezek külön komponensként szerepelnek az alkalmazásban)
Sanitizer+	HTML gyakorlófelület, ahol a képernyő egyik oldalán egy szövegbeviteli mezőbe lehet html kódot írni, a másik oldalán pedig az adott kód az ngSanitizer modulnak köszönhetően megjelenik a komponens részeként. A két sáv alatt menüsor található, amely különböző tipikus html elemeket sorol fel, amelyekre kattintva annak HTML tagje automatikusan beszűrődik a kurzor helyére a szerkesztőfelületen. A létrehozott projektek kimenthetőek localStorage-ba, melynek tartalmát egy ListComponent felsorolja, ahonnan bármelyik korábbi mentett projekt, forrásállomány visszatölthető, megnyitható
HouseManagement	Interfész egy okosház vezérléséhez. Különböző felületeken lehessen szellőztetni, főzőberendezéseket bekapcsolni, mosógépet, szárítót üzemeltetni, fűtést állítani, illetve a többi családtag hozzáférését állíthatni. A rendszer automatikusan figyelmeztessen, ha bizonyos tevékenységek, mint a mosogatás, mosdótisztítás, személtürítés régóta nem voltak elvégezve. Az ötletet bővítsd nyugodtan saját elképzelésekkel is!
COVID-19	Olyan alkalmazás, amely statisztikákat mutat a 2020-as koronavírus terjedéséről. A statisztikákat ki lehet tenni táblázatokba is, azonban a különböző statisztikáknak külön lapon kell megjeleníteniük. Legalább 5 statisztika elkészítése kötelező. A táblázatos megjelenítés nem kötelező, de diagram vagy térkép használata igen. Ehhez használható pl.: chart.js, canvasJS, stb. Az adatok begyűjtéséhez használni kell egy elérhető API-t: https://rapidapi.com/collection/coronavirus-covid-19