Általános információk

A diplomaterv szerkezete:

1. Diplomaterv feladatkiírás
2. Címoldal
3. Tartalomjegyzék
4. A diplomatervező nyilatkozata az önálló munkáról és az elektronikus adatok kezeléséről
5. Tartalmi összefoglaló magyarul és angolul
6. Bevezetés: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása
7. A feladatkiírás pontosítása és részletes elemzése
8. Előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések
9. A tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása
10. A megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek
11. Esetleges köszönetnyilvánítások
12. Részletesés pontos irodalomjegyzék
13. Függelék(ek)

Felhasználható a következő oldaltól kezdődő Diplomaterv sablon dokumentum tartalma. Ügyeljen a tanszék, a hallgató, a konzulens nevét és a beadás évét jelölő szövegdobozokra, mert azokra külön ki kell adni a frissítést. A mezők tartalma a sablonban a dokumentum adatlapja alapján automatikusan kerül kitöltésre (Fájl/Információ/Tulajdonságok/Speciális tulajdonságok).

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az oldalak tükörmargóval készüljenek (mindenhol 2.5cm, baloldalon 1cm-es kötéssel). Az alapértelmezett betűkészlet a 12 pontos Times New Roman, másfeles sorközzel.

Minden oldalon - az első négy szerkezeti elem kivételével - szerepelnie kell az oldalszámnak.

A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetenként decimális számmal és kifejező címmel kell ellátni. A fejezeteket decimális aláosztással számozzuk, maximálisan 3 aláosztás mélységben (pl. 2.3.4.1.). Az ábrákat, táblázatokat és képleteket célszerű fejezetenként külön számozni (pl. 2.4. ábra, 4.2 táblázat vagy képletnél (3.2)). A fejezetcímeket igazítsuk balra, a normál szövegnél viszont használjunk sorkiegyenlítést. Az ábrákat, táblázatokat és a hozzájuk tartozó címet igazítsuk középre. A cím a jelölt rész alatt helyezkedjen el.

A képeket lehetőleg rajzoló programmal készítsék el, az egyenleteket egyenlet-szerkesztő segítségével írják le.

Az irodalomjegyzék szövegközi hivatkozása történhet a Harvard-rendszerben (a szerző és az évszám megadásával) vagy sorszámozva. A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címeit azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben. Minden publikáció a szerzők után a következő adatok szerepelnek: folyóirat cikkeknél a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóirat címeket csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internet hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal tulajdonosát és tartalmát (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Fontos:

* a szakdolgozat készítő/diplomatervező nyilatkozata (a jelen sablonban szereplő szövegtartalommal) kötelező előírás Karunkon, ennek hiányában a szakdolgozat/diplomaterv nem bírálható és nem védhető!
* mind a dolgozat, mind a melléklet maximálisan 15 MB méretű lehet!

Jó munkát, sikeres szakdolgozat készítést ill. diplomatervezést kívánunk!

FELADATKIÍRÁS

A feladatkiírást a **tanszék saját előírása szerint** vagy a tanszéki adminisztrációban lehet átvenni, és a tanszéki pecséttel ellátott, a tanszékvezető által aláírt lapot kell belefűzni a leadott munkába, vagy a tanszékvezető által elektronikusan jóváhagyott feladatkiírást kell a Diplomaterv Portálról letölteni és a leadott munkába belefűzni (ezen oldal HELYETT, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell megismételni a feladatkiírást.



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Sereg András

Testre szabható koktéladatbázis készítése

React.js és Node.js segítségével

Konzulens

Kövesdán Gábor

BUDAPEST, 2023

Tartalomjegyzék

[Összefoglaló 7](#_Toc152698894)

[Abstract 8](#_Toc152698895)

[1 Bevezetés 9](#_Toc152698896)

[2 Felhasznált technológiák 10](#_Toc152698897)

[2.1 Backendhez használt technológiák 10](#_Toc152698898)

[2.1.1 nodejs 10](#_Toc152698899)

[2.1.2 mongodb 10](#_Toc152698900)

[2.1.3 … 10](#_Toc152698901)

[2.2 Frontendhez használt technológiák 10](#_Toc152698902)

[2.2.1 React 10](#_Toc152698903)

[2.2.2 React-bootstrap 10](#_Toc152698904)

[2.2.3 axios 10](#_Toc152698905)

[2.2.4 Typescript 10](#_Toc152698906)

[2.2.5 … 10](#_Toc152698907)

[2.3 Szolgáltatások 10](#_Toc152698908)

[2.3.1 Google Maps API 10](#_Toc152698909)

[2.3.2 Firebase auth 10](#_Toc152698910)

[3 Hasonló alkalmazások 11](#_Toc152698911)

[3.1 Untappd 11](#_Toc152698912)

[4 Tervezés 12](#_Toc152698913)

[4.1.1 Architektúra 12](#_Toc152698914)

[4.1.2 Felhasználói szerepkörök 12](#_Toc152698915)

[4.1.3 Adatmodellek 12](#_Toc152698916)

[4.1.4 REST API végpontok 12](#_Toc152698917)

[4.1.5 Fontosabb funkciók 12](#_Toc152698918)

[5 Implementáció 13](#_Toc152698919)

[5.1 -Backend: 13](#_Toc152698920)

[5.1.1 mongoose schema-k, 13](#_Toc152698921)

[5.1.2 routeok, 13](#_Toc152698922)

[5.1.3 autentikáció 13](#_Toc152698923)

[5.2 -Frontend: 13](#_Toc152698924)

[5.2.1 komponensek, 13](#_Toc152698925)

[5.2.2 kommunikáció 13](#_Toc152698926)

[5.2.3 Jogosultságok 13](#_Toc152698927)

[6 Telepítési és használati útmutató 14](#_Toc152698928)

[7 Összegzés 15](#_Toc152698929)

[8 Köszönetnyilvánítás 16](#_Toc152698930)

[9 Irodalomjegyzék 17](#_Toc152698931)

[10 Bevezetés 18](#_Toc152698932)

[10.1 Formázási tudnivalók 18](#_Toc152698933)

[10.1.1 Címsorok 18](#_Toc152698934)

[10.1.2 Képek 18](#_Toc152698935)

[10.1.3 Kódrészletek 18](#_Toc152698936)

[10.1.4 Irodalomjegyzék 19](#_Toc152698937)

[11 Utolsó simítások 20](#_Toc152698938)

[Irodalomjegyzék 21](#_Toc152698939)

[Függelék 22](#_Toc152698940)

Hallgatói nyilatkozat

Alulírott **Sereg András**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2023. 12. 04.

...…………………………………………….

Sereg András

Összefoglaló

Abstract

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

# Bevezetés

Ahány ember, annyi féle preferencia. Nincs ez másképp, ha a szórakozási vagy italfogyasztási szokásokról beszélünk. Van, hogy egyedül is nehezen tudunk dönteni arról, hogy hol és hogyan töltsük el a péntek-szombat estéinket, egy több fős társasággal pedig még nehezebb dűlőre jutni.

A dolgozatom célja, hogy egy olyan alkalmazás készítsek, mellyel a felhasználók minél több szempont szerint alakíthassák a preferenciáikat, és azok alapján megtalálják Budapest számukra legmegfelelőbb szórakoztatói egyégét.

A dolgozatom során a 2. fejezetben ismertetem az alkalmazás elkészítése során használt technológiákat, a 3. fejezetben a már létező, az alkalmazáshoz hasonló szolgáltatásokat hasonlítom össze. A 4. fejezetben a tervezési lépéseket ismertetem, az 5. fejezetben pedig az implementáció részleteit mutatom be.

# Felhasznált technológiák

## Backendhez használt technológiák

### nodejs

### mongodb

### …

## Frontendhez használt technológiák

### React

A React.js egy nyílt forráskódú Javascript könyvtár, amellyel felhasználói felületeket készíthetünk, elsősorban Single Page Application (későbbiekben SPA) formájában.

Egyik főbb jellemzője a virtuális DOM, ami lehetővé teszi a hatékony és gyors tartalomfrissítéseket az alkalmazásunkban, anélkül, hogy az oldalt újra kelljen töltenünk. Ezt úgy éri el, hogy a komponensek változásakor a DOM-nak egy absztrahált változatát módosítja, ezzel csak az érintett állapotok változnak meg

A Reactban az alkalmazások úgynevezett komponensekből épülnek fel, melyek elősegítik az újrahasználhatóságot, és könnyebben karbantarthatóvá és tesztelhetővé teszi a kódot.

Lehetőségünk van különböző Hookok használatára, aminek segítségével függvényként készíthetjük el a komponenseinket, valamint lehetőséget biztosít a fejlesztőknek hogy különválasszák a megjelenítést a működési logikát.

### React-bootstrap

A React Bootstrap a Bootstrap grafikus keretrendszer React-komponensekkel való kiegészítése, melynek segítségével hatékony módon vihetünk különböző UI elemeket az alkalamazásunkba, amelyek követik a Bootstrap stílusrendszerét.

Egyes komponensek megjelenítését függővé tehetjük a kijelző méretétől, így néhány sornyi kóddal több különböző eszközre optimalizált kódot írhatunk.

A könyvtár használatával számos előre összerakott komplex komponenseket is tudunk használni, mint például különböző űrlap elemek, vagy felugró ablakok

### axios

Webalkalmazások létrehozásakor gyakori feladat a HTTP protokollon keresztül való kommunikáció. A böngészők képesek fogadni HTTP kéréseket a beépített fetch() metódusukkal, ezek küldését pedig az Axios javascript könyvtár segítségével lehet megoldani.

### Typescript

A javascript webfejlesztésben való elterjedésével rohamosan derültek ki a nyelv anomáliái és hiányosságai. Ezt hivatott orvosolni a Typescript, ami a Javascript egy szigorúbb kiterjesztése. Segítségével fordítási időben is kezelhetünk hibákat, az erős típusellenőrzésnek köszönhetően pedig a legtöbb hiba már a fejlesztőkörnyezetünkben jelentkezik. Létrehozhatunk benne egyedi típusokat és interfaceket is

## Szolgáltatások

### Verziókezelés

Fejlesztés során fontos nyomon követni a fájlokban végrehajtott változtatásokat. A verziókezelés lehetővé teszi, hogy visszatérjünk korábbi változatokhoz és új verziókat hozzunk létre. Erre a legelterjedtebb eszköz az, ingyenes és nyílt forráskódú git..  
Kisebb és nagyobb projektekben is használható, egyedül inkább verziókövetésre alkalmas, de hasznos tud lenni ha különböző eszközökön fejlesztünk. Több felhasználó számára lehetőséget biztosít arra, hogy ugyanazon a kódon dolgozzanak, és könnyedén kommunikáljanak egymás között.

A git parancsait kiadhatjuk manuálisan az operációs rendszerünk parancssorából, vagy a git saját Command Line Interface-ében (CLI), de léteznek különböző asztali alkalmazások, mint például a Github Desktop, ami gyorsabbá és felhasználóbarátabbá teszi a könyvtárak verziókezelését. Emellett a legtöbb modern fejlesztőkörnyezetben elérhetők különböző bővítmények, melyek segítségével mellőzhető a CLI-k használata

A Github egy olyan szolgáltatás, ami git könyvtárak(repository-k) tárolására szolgál. Könnyen átlátható felületet biztosít a változtatásokhoz és lehetővé teszi különböző műveletek végrehajtását, mint a kódértékelés, vagy úgynevezett issue-k létrehozása, melyek segítségével hatékonyan nyilvántarthatóvá válnak az elvégzendő részfeladatok.

### Google Maps API

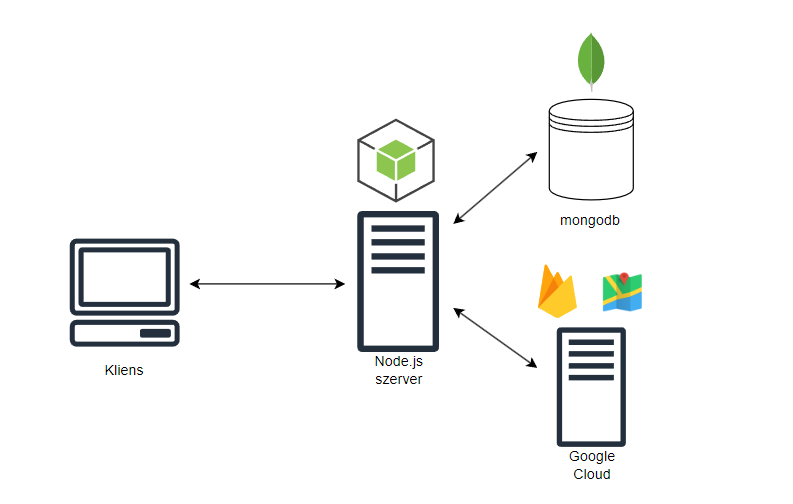
### Firebase auth

# Hasonló alkalmazások

## Untappd

# Tervezés

### Architektúra



### Felhasználói szerepkörök

Felhasználó

Admin

### Adatmodellek

### REST API végpontok

### Fontosabb funkciók

# Implementáció

## -Backend:

### mongoose schema-k,

### routeok,

### autentikáció

## -Frontend:

### komponensek,

### kommunikáció

### Jogosultságok

# Telepítési és használati útmutató

# Összegzés

# Köszönetnyilvánítás

# Irodalomjegyzék

# Bevezetés

A következő fejezet pár példán keresztül bemutatja a diplomatervekben és szakdolgozatokban szokásosan előkerülő formázások megvalósítását.

## Formázási tudnivalók

A dokumentum folyószövegéhez használjuk a **Normál** (angol Word esetén Normal) stílust.

### Címsorok

A fejezetcímek esetén a **Címsor 1-4** (Heading 1-4) stílusokat használjuk.

### Képek

A képhez használjuk a **Kép** stílust.

Képaláírást a képen jobb gombbal kattintva a Képaláírás beszúrása… opcióval adhatunk hozzá, így az automatikusan **Képaláírás** (Caption) stílusú lesz.



1.1. ábra: Példa képaláírásra

### Kódrészletek

Kódrészletek beillesztése esetén használjuk a **Kód** stílust.

using System;

namespace MyApp

{

class Program

{

static void Main( string[] args )

{

Console.WriteLine( "Szia Világ!" );

}

}

}

### Irodalomjegyzék

Az Irodalomjegyzékben szereplő hivatkozásokat **Irodalomjegyzék sor** stílussal formázzuk, a címüket pedig **Irodalomjegyzék forrás** stílussal emeljük ki.

A szövegbe a hivatkozásokat a Kereszthivatkozás beszúrása (Insert cross-reference) funkcióval helyezzük el (példa egy így beszúrt hivatkozásra: [1]), így azok automatikusan frissülnek a hivatkozások átrendezésekor.

# Utolsó simítások

Miután elkészültünk a dokumentációval, ne felejtsük el a következő lépéseket:

* Kereszthivatkozások frissítése: miután kijelöltük a teljes szöveget (Ctrl+A), nyomjuk meg az F9 billentyűt, és a Word frissíti az összes kereszthivatkozást. Ilyenkor ellenőrizzük, hogy nem jelent-e meg valahol a "Hiba! A könyvjelző nem létezik." szöveg.
* Dokumentum tulajdonságok megadása: a dokumentumhoz tartozó meta adatok kitöltése (szerző, cím, kulcsszavak stb.). Erre való a Dokumentum tulajdonságai panel, mely a Fájl / Információ / Tulajdonságok / Dokumentumpanel megjelenítése úton érhető el.
* Kinézet ellenőrzése PDF-ben: a legjobb teszt a végén, ha PDF-et készítünk a dokumentumból, és azt leellenőrizzük.

Irodalomjegyzék

1. Levendovszky, J., Jereb, L., Elek, Zs., Vesztergombi, Gy.: Adaptive statistical algorithms in network reliability analysis, Performance Evaluation - Elsevier, Vol. 48, 2002, pp. 225-236
2. National Istruments: LabVIEW grafikus fejlesztői környezet leírása, <http://www.ni.com/> (2010. nov.)
3. Fowler, M.: UML Distilled, 3rd edition, ISBN 0-321-19368-7, Addison-Wesley, 2004
4. Wikipedia: Evaluation strategy, <http://en.wikipedia.org/wiki/Evaluation_strategy> (revision 18:11, 31 July 2012)

Függelék