

# SZAKDOLGOZAT



MISKOLCI EGYETEM

## Interaktív megjelenítő eszköz pénzügyi adatok elemzéséhez

**Készítette:**

Bencze Zsombor

Programtervező informatikus

**Témavezető:**

Piller Imre

MISKOLC, 2023

MISKOLCI EGYETEM

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

Szám:

## SZAKDOLGOZAT FELADAT

Bencze Zsombor (LP5J4B) programtervező informatikus jelölt részére.

**A szakdolgozat tárgyköre:** Webfejlesztés, megjelenítő eszköz

**A szakdolgozat címe:** Interaktív megjelenítő eszköz pénzügyi adatok elemzéséhez

**A feladat részletezése:**

*A szakdolgozat célja olyan interaktív, dinamikus megjelenítési módok tervezésének, működésének és használatának a bemutatása, amelyekkel egyazon idősor különböző részei, különböző forrásból származó idősorok, az azokból származtatott értékek összehasonlíthatók, az aggregáláshoz használt paraméterek rugalmasan változtathatók.*

*A pénzügyi adatok (például tőzsdei árfolyamok) elemzéséhez elengedhetetlen, hogy a rendelkezésre álló információk a szakértők számára könnyen áttekinthető formában rendelkezésre álljanak*

*Az alkalmazásnak webes környezetben, szerver-kliens architektúrának megfelelően kell elkészülnie. Ehhez szerver oldalon Node.JS programnyelvet fogok használni. Míg a kliens weboldal megvalósításához HTML5, CSS, JavaScript programnyelveket kell használni. Az alkalmazás adatai adatbázisban lesznek eltárolva*

**Témavezető:** Piller Imre (egyetemi tanársegéd)

**A feladat kiadásának ideje:**

.....  
szakfelelős

## EREDETISÉGI NYILATKOZAT

Alulírott **Bencze Zsombor**; Neptun-kód: LP5J4B a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *Interaktív megjelenítő eszköz pénzügyi adatok elemzéséhez* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, ..... év ..... hó ..... nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat ..... szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve: .....

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata: .....

a bíráló javaslata: .....

a szakdolgozat végleges eredménye: .....

Miskolc, .....

.....

a Záróvizsga Bizottság Elnöke

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
<b>2. Elméleti háttér</b>	<b>2</b>
2.1. Grafikonok története . . . . .	2
2.2. Tartalom és felépítés . . . . .	2
2.3. Amit csak említés szintjén érdemes szerepeltetni . . . . .	3
<b>3. Tervezés</b>	<b>4</b>
3.1. Táblázatok . . . . .	4
3.2. Ábrák . . . . .	4
3.3. További környezetek . . . . .	4
<b>4. Megvalósítás</b>	<b>6</b>
<b>5. Tesztelés</b>	<b>7</b>
<b>6. Összefoglalás</b>	<b>8</b>
<b>Irodalomjegyzék</b>	<b>9</b>

# 1. fejezet

## Bevezetés

A szakdolgozatom egy weboldal, amely alkalmas többféle pénzügyi adat elemzéséhez, ehhez különféle befektetési alapokat használok, továbbá egy grafikonrajzoló. Ennek az oldalnak a szervezeti felépítését, elkészítését és az alkalmazás működését mutatom be. A pénzügyi adatok, vagy tőzsdei árfolyamok megismerése lehet nagyon egyszerű, de gyakran nehézséget jelent azoknak, akik először próbálkoznak meg vele a hétköznapiakban. Ezért a dolgozatom olyan lehetőségeket mutat be, amelyekkel átláthatóbban lehet megérteni és kezelni őket.

Ezek a lehetőségeket kétféle alternatívára osztottam szét. Az egyik opció a részletes leírása és ismertetése a befektetési alapoknak, kiegészítve egy táblázattal, amely ábrázolja a hozam-kockázat profilt, továbbá egy alap által elért éves nettó hozam sáv. A másik lehetőségként külön oldalon grafikonrajzoló használható. Lehetőség van kezdési és zárópontot megadni, így különböző intervallumokat kiválasztva tudjuk vizsgálni az adott befektetési alapot. A dolgozat az utóbbira fektet nagyobb hangsúlyt több okból kifolyólag is. Ezt elsősorban az indokolja, hogy az így nyert adatokat rugalmasabban tudja kezelni a felhasználó, mivel egyszerre több kiválasztási lehetőség áll rendelkezésre, amivel részletesebb adathalmazt nyerhető ki. A teljesség igénye nélkül felsorolok pár lehetőséget. Például napi adatokat lehet vizsgálni olyan kategóriák szerint, mint napi legmagasabb, vagy éppen napi legalacsonyabb árfolyam. Továbbá a legelterjedtebb ábrázolási forma, ha vonaldiagrammon ábrázoljuk a kívánt adatokat. Az alkalmazás során számomra szempont volt, hogy ettől függetlenül több lehetőséget kínáljak a felhasználónak, így lehetősége nyílik több ábrázolási módszer közül is választani, úgy mint mondjuk oszlopdiagram, vagy kördiagram.

A megjelenítéshez és az algoritmusok fejlesztéséhez több programozási nyelvet is került választásra, többek között HTML5 képezi a weboldal vázát, CSS a weboldal megjelenítését képezi, Node.JS amellyel a weboldal szervere működik és végezetül JavaScript nyelven válik elérhetővé a grafikonrajzoló teljes egésze. Ezeken felül még használtam Javascript könyvtárat, mint a JQuery, illetve az oldal reszponzivitásáért felel a nyílt forráskódú kliens oldali keretrendszer a Bootstrap5.

Több grafikonrajzoló és eszközkezelő weboldal elérhető a különböző befektetési vállalatoknak az interneten. Ebből kifolyólag felmerülhet a kérdés, hogy akkor miért volt szükség még egy weboldal elkészítésére? Többek között erre a kérdésre is választ kaphatunk a szakdolgozat elolvasása után.

## 2. fejezet

# Elméleti háttér

később látványos kimutatások készíthetők

### 2.1. Grafikonok története

Ez a fejezet még nem a saját eredményekkel foglalkozik, hanem bemutatja, mi a problémakör, milyen módszerekkel, milyen eredményeket sikerült elérni eddig másoknak.

A hivatkozások jelentős része ehhez a fejezethez szokott kötődni. (Egy hivatkozás például így néz ki [1].) Itt lehet bemutatni a hasonló alkalmazásokat.

### 2.2. Tartalom és felépítés

A fejezet tartalma témától függően változhat. Az alábbiakat attól függően különböző arányban tartalmazhatják.

- Irodalomkutatás. Amennyiben a dolgozat egy módszer kidolgozására, kifejlesztésére irányul, akkor itt lehet részletesen végignézni (módszertani vagy időrendi bontásban), hogy az eddigiekben milyen eredmények születtek a témakörben.
- Technológia. Mivel jellemzően kutatásról vagy szoftverfejlesztésről van szó, ezért annak a jellemző elemeit, technikai részleteit itt kell bemutatni. Ez tehát egy módszeres bevezetés ahhoz, hogy ha valaki nem jártas a témakörben, akkor tudja, hogy a dolgozat milyen aktuálisan elérhető eredményeket, eszközöket használt fel.
- Piackutatás. Bizonyos témáknál új termék vagy szolgáltatás kifejlesztése a cél. Ekkor érdemes annak alaposan utánanézni, hogy aktuálisan milyen eszközök érhetők el a piacon. Ez szoftverek esetében a hasonló alkalmazások bemutatását, táblázatos formában történő összehasonlítását jelentheti. Szerepelhetnek képek és észrevételek a viszonyításként bemutatott alkalmazásokhoz.
- Követelmény specifikáció. Külön szakaszban érdemes részletesen kitérni az elkészítendő alkalmazással kapcsolatos követelményekre. Ehhez tartozhatnak forgatókönyvek (*scenario*-k). A szemléletesség kedvéért lehet hozzájuk képernyőkép vázlatokat is készíteni, vagy a használati eseteket más módon szemléltetni.

## 2.3. Amit csak említés szintjén érdemes szerepeltetni

Az olvasóról annyit feltételezhetünk, hogy programozásban valamilyen szinten járatos, és a matematikai alapfogalmakkal sem ebben a dolgozatban kell megismertetni. A speciális eszközök, programozási nyelvek, matematikai módszerek és jelölések persze jó, hogy ha említésre kerülnek, de nem kell nagyon belemenni a közismertnek tekinthető dolgokba.



## 3. fejezet

### Tervezés

Itt kezdődik a dolgozat lényegi része, úgy érte, hogy a saját munka bemutatása. Jellemzően ebben szerepelni szoktak blokkdiagramok, a program struktúrájával foglalkozó leírások. Ehhez célszerű UML ábrákat (például osztály- és szekvenciadiagramokat) használni.

Amennyiben a dolgozat inkább kutatás jellegű, úgy itt lehet konkretizálni a kutatási módszertant, a kutatás tervezett lépéseit, az indoklást, hogy mit, miért és miért pont úgy érdemes csinálni, ahogyan az a későbbiekben majd részletezésre kerül.

Ebben a fejezetben az implementáció nem kell, hogy túl nagy szerepet kapjon. Ez még csak a tervezési fázis. (Nyilván ha olyan a téma, hogy magának az implementációnak a módjával foglalkozik, adott formális nyelvet mutat be, úgy a kód példákat már innen sem lehet kihagyni.)

#### 3.1. Táblázatok

Táblázatokhoz a `table` környezetet ajánlott használni. Erre egy minta a 3.1. táblázat. A hivatkozáshoz az egyedi `label` értéke konvenció szerint `tab:` prefixszel kezdődik.

3.1. táblázat. Minta táblázat. A táblázat felirata a táblázat felett kell legyen!

a	b	c
1	2	3
4	5	6

#### 3.2. Ábrák

Ábrákat a `figure` környezettel lehet használni. A használatára egy példa a 3.1. ábrán látható. Az `includegraphics` parancsba Az ábrák felirata az ábra alatt kell legyen. Az ábrák hivatkozásához használt nevet konvenció szerint `fig:`-el célszerű kezdeni.

#### 3.3. További környezetek

A matematikai témájú dolgozatokban szükség lehet tételek és bizonyításaik megadására. Ehhez szintén vannak készen elérhető környezetek.



3.1. ábra. A Miskolci Egyetem címere.

**3.1. definíció.** Ez egy definíció

**3.2. lemma.** *Ez egy lemma*

**3.3. tétel.** *Ez egy tétel*

**Bizonyítás.** Ez egy bizonyítás

□

**3.4. következmény.** *Ez egy tétel*

**3.5. megjegyzés.** Ez egy megjegyzés

**3.6. példa.** Ez egy példa

## 4. fejezet

# Megvalósítás

Ez a fejezet mutatja be a megvalósítás lépéseit. Itt lehet az esetlegesen előforduló technikai nehézségeket említeni. Be lehet már mutatni a program elkészült részeit.

Meg lehet mutatni az elkészített programkód érdekesebb részeit. (Az érdekesebb részek bemutatására kellene szorítkozni. Többségében a szöveges leírásnak kellene benne lennie. Abból lehet kiindulni, hogy a forráskód a dolgozathoz elérhető, azt nem kell magába a dolgozatba bemásolni, elegendő csak behivatkozni.)

A dolgozatban szereplő forráskódrészletekhez külön vannak programnyelvenként stílusok. Python esetében például így néz ki egy formázott kódrészlet.

```
import sys

if __name__ == '__main__':
    pass
```

A stílusfájlok a `styles` jegyzékben találhatók. A stílusok között szerepel még C++, Java és Rust stílusfájl. Ezek használatához a `dolgozat.tex` fájl elején `usepackage` paranccsal hozzá kell adni a stílust, majd a stílusfájl nevével megegyező környezetet lehet használni. További példaként C++ forráskód esetében ez így szerepel.

```
#include <iostream>

class Sample : public Object
{
    // An empty class definition
}
```

Stílusfájlokból elegendő csak annyit meghagyni, amennyire a dolgozatban szükség van. Más, C szintaktikájú nyelvekhez (mint például a JavaScript és C#) a Java vagy C++ stílusfájlok átszerkesztésére van szükség. (Elegendő lehet csak a fájlnevet átírni, és a fájlban a környezet nevét.)

Nyers adatok, parancssori kimenetek megjelenítéséhez a `verbatim` környezetet lehet használni.

```
$ some commands with arguments
1 2 3 4 5
$ _
```

A kutatás jellegű témáknál ez a fejezet gyakorlatilag kimaradhat. Helyette inkább a fő vizsgálati módszerek, kutatási irányok kaphatnak külön-külön fejezeteket.

## 5. fejezet

### Tesztelés

A fejezetben be kell mutatni, hogy az elkészült alkalmazás hogyan használható. (Az, hogy hogyan kell, hogy működjön, és hogy hogy lett elkészítve, az előző fejezetekben már megtörtént.)

Jellemzően az alábbi dolgok kerülhetnek ide.

- Tesztfuttatások. Le lehet írni a futási időket, memória és tárigényt.
- Felhasználói kézikönyv jellegű leírás. Kifejezetten a végfelhasználó szempontjából lehet azt bemutatni, hogy mit hogy lehet majd használni.
- Kutatás kapcsán ide főként táblázatok, görbék és egyéb részletes összesítések kerülhetnek.

## 6. fejezet

### Összefoglalás

Hasonló szerepe van, mint a bevezetésnek. Itt már múltidőben lehet beszélni. A szerző saját meglátása szerint kell összegezni és értékelni a dolgozat fontosabb eredményeit. Meg lehet benne említeni, hogy mi az ami jobban, mi az ami kevésbé jobban sikerült a tervezettnél. El lehet benne mondani, hogy milyen további tervek, fejlesztési lehetőségek vannak még a témával kapcsolatban.

# Irodalomjegyzék

- [1] James H Coombs, Allen H Renear, and Steven J DeRose. Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11):933–947, 1987.

# CD Használati útmutató

Ennek a címe lehet például *A mellékelt CD tartalma* vagy *Adathordozó használati útmutató* is.

Ez jellemzően csak egy fél-egy oldalas leírás. Arra szolgál, hogy ha valaki kézhez kapja a szakdolgozathoz tartozó CD-t, akkor tudja, hogy mi hol van rajta. Jellemzően elég csak felsorolni, hogy milyen jegyzékek vannak, és azokban mi található. Az elkészített programok telepítéséhez, futtatásához tartozó instrukciók kerülhetnek ide.

A CD lemezre mindenképpen rá kell tenni

- a dolgozatot egy `dolgozat.pdf` fájl formájában,
- a LaTeX forráskódját a dolgozatnak,
- az elkészített programot, fontosabb futási eredményeket (például ha kép a kimenet),
- egy útmutatót a CD használatához (ami lehet ez a fejezet külön PDF-be vagy Markdown fájlként kimentve).