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 elemzéséheze

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 aláírásommal 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:

- szószerinti 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

1. fejezet

Bevezetés

A szakdolgozatom egy weboldal, amely alkalmas többféle pénzügyi adat elemzéséhez, ehhez különféle befektetési alapokat használok, továbbá egy grafikonrajzolót. 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 hétköznapokban. 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 nyelv is kiválasztásra került, többek között HTML5 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, miért volt szükség még egy weboldal elkészítésére? Többek között erre a kérdésre is választ kaphatunk a szakdolgozat elolvasása után.

2. fejezet

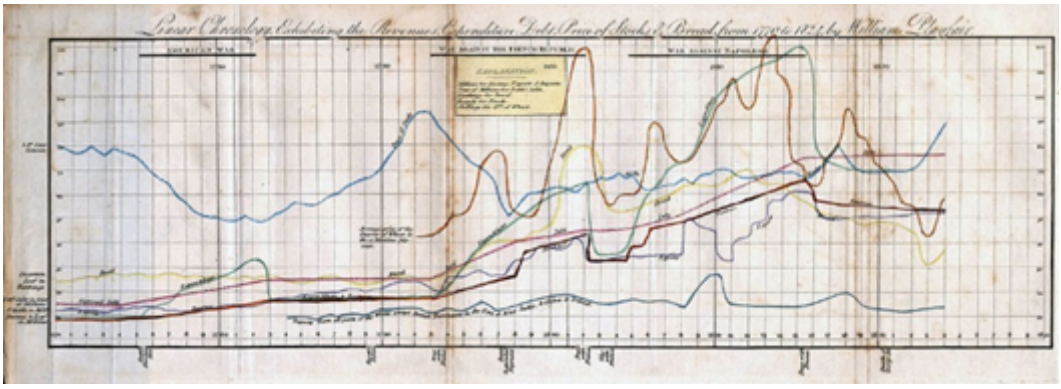
Elméleti háttér

Szakdolgozatom célja létrehozni egy olyan kliens-szerver weboldalt, amely bárki számára könnyen használható és látványos kimutatásokat tud ezáltal készíteni. Már a tervezési fázisban elhatároztam, hogy a weboldal felépítése hasonlítani fog a kutatásom során megismert tőszdei témájú weboldalakhoz, mind felépítésében, mind megvalósításában, hogy a felhasználó azt érezze egy *élő és lélegző weboldal*on böngészik. Ehhez viszont előtte szeretném ismertetni a szakdolgozatomhoz szükséges tudáshátteret.

2.1. Grafikonok története és típusai [1] [2]

Egy látványos grafikonnal könnyebb bemutatni egy elemzést, vagy történetet, mint szóban elmesélni. Ugyan ma alapkellékként értelmezzük a vonaldiagrammokat és kördiagrammokat, viszont egykor forradalmi újításnak számítottak. Volt rá példa, hogy használatuk szó szerint életbevágónak bizonyult, amikor egy angol orvosnő, diagrammok segítségével győzte meg a férfiak által vezetett orvosközösséget, hogy jobban oda kell figyelni a kórházi higiéniára. A kifogástalan és figyelem felkeltő grafikonjának hála, katonák tömegei élték túl a krími háborút.

A grafikon a diagram egyik fajtája, amely két adat kapcsolatát egy derékszögű koordináta-rendszerben ábrázolja. A két tengely jelképezi a két adatot, és a grafikon jelzi, hogy az egyik adatnak az egyes értékeihez a másik mely értékei tartoznak. Általában két mennyiség közötti lineáris kapcsolatot, vagy egy mennyiség időbeli változásának bemutatására használják. Matematikai szemszögből a grafikon egy függvényt ábrázol.



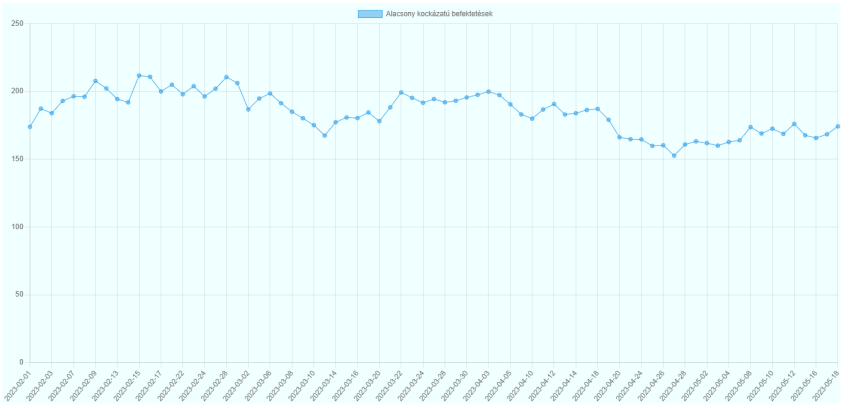
2.1. ábra. Az egyik első statisztikai gráf (forrás: [14])

2.1.1. Grafikon típusok [3]

Vonaldiagram

A vonaldiagram olyan grafikon, amely vonalakat használ az egyes adatpontok összekapcsolására. A vonaldiagram kvantitatív értékeket jelenít meg egy meghatározott időintervallumban. A pénzügyekben a vonaldiagrammokat általában egy eszköz vagy értékpapír történelmi árfolyamműveletének ábrázolására használják.

Működését tekintve egy vonal köti össze az egyes adatpontokat, amelyek jellemzően mennyiségi értékeket jelenítenek meg. Befektetések terén a technikai elemzés területén a vonalgrafikonok meglehetősen informatívak, lehetővé téve a felhasználó számára a trendek megjelenítését. Míg a vonaldiagrammokat sok különböző mezőben használják különböző célokra, leggyakoribb funkciójuk az értékek időbeli változásainak grafikus ábrázolása.

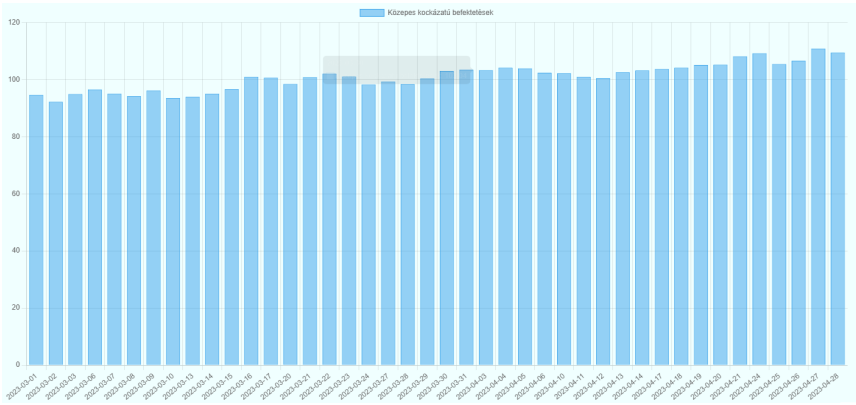


2.2. ábra. Általam készített grafikonrajzoló oldalán megjelenített vonaldiagram

Oszlopdiagram

Az oszlopdiagram az információ grafikus ábrázolására szolgál. Különböző magasságú sávokat használ az értékek kimutatására.

Működését tekintve oszlopdiagrammokat létre lehet hozni függőleges sávokkal, vízszintes sávokkal, csoportos sávokkal (több sáv, amelyek egy kategória értékeit hasonlítják össze) vagy halmozott sávokkal (több típusú információt tartalmazó oszlopok). A sávok egy adott adatkategória értékét jelenítik meg az határzorra meg annak méretét. Az oszlopdiagramokat általában az üzleti és pénzügyi elemzésekben használják a gyakran bonyolult adatok megjelenítésére, mivel gyorsan és hatékonyan tudnak információt közvetíteni.

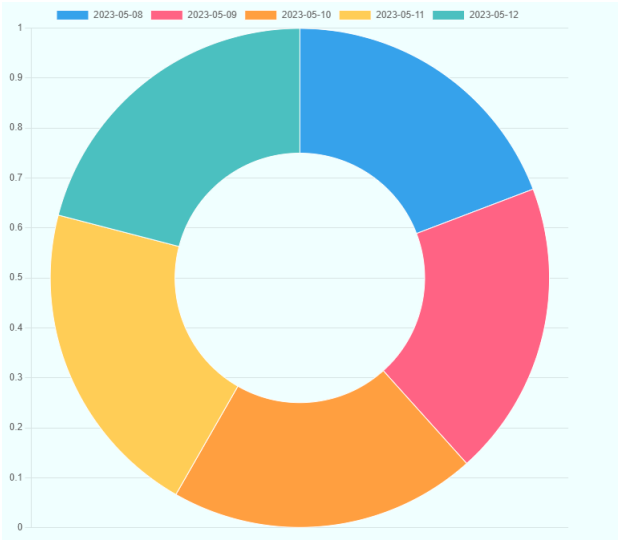


2.3. ábra. Általam készített grafikonrajzoló oldalán megjelenített oszlopdiagram

Kördiagram

A kördiagram egy névleges adathalmaz összegzésének vagy egy adott változó különböző értékeinek megjelenítésének módja (pl. százalékos eloszlás). A kördiagrammok használata meglehetősen népszerű, mivel a kör vizuális koncepciót ad az egészről. Egyik alfaja a poláris diagram.

Működését tekintve az ilyen típusú diagram egy szegmenssorozatra osztott kör. Minden szegmens egy adott kategóriát képvisel. Az egyes szegmensek területe a körnek ugyanolyan arányban van, mint a kategória a teljes adatkészletben.



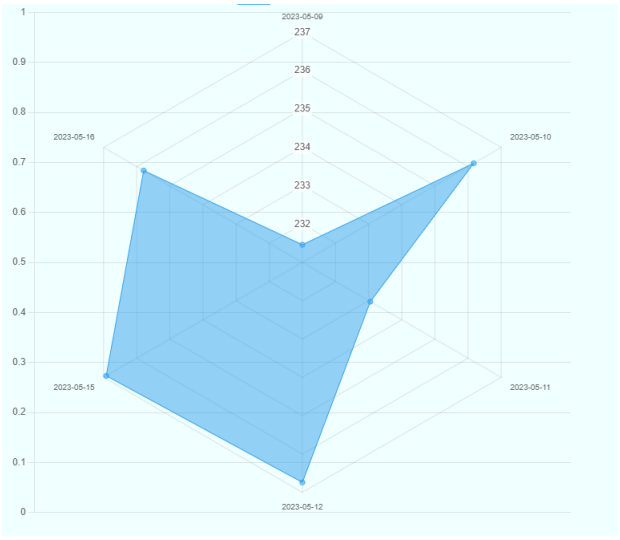
2.4. ábra. Általam készített grafikonrajzoló oldalán megjelenített kördiagram

Radardiagram

A radardiagram segít szemléltetni a különböző jellemzőkkel rendelkező adatcsoportok összehasonlítását. Segítségével könnyen össze lehet mérni másfajta paraméterekkel rendelkező elemeket.

Működését tekintve az adatokat ugyanarra a központi pontra helyezi, és átlátszó árnyalatokkal és mintákkal mutatja be a kontrasztot az olvasó számára. Egy adott

pont értékének nagyságát az jelzi, minél távolabbra nyúlik az origótól a pókháló szerű diagrammon.



2.5. ábra. Általam készített grafikonrajzoló oldalán megjelenített radardiagram

2.2. Tőzsde [4]

A tőzsde talán a modernt világ egyik legjobban félreértelmezett kifejezése.A hétköznapi ember lelki szemei előtt számok végtelen sokasága lebeg, amikor meghallja és csak legyint rá, hogy ez számára túl komplikált, ahhoz, hogy átlássa és megértse. Nos, ez az állítás részben igaz is, hiszen lehet nagyon bonyolult a tőzsde, viszont lehet nagyon egyszerű is.

A tőzsde egy olyan nyilvános, központosított és szervezett piac, ahol meghatározott árukat, meghatározott időben, azon belül meghatározott személyek adhatnak, vagy vehetnek szigorú eljárási szabályok szerint. Ez mit is jelent a gyakorlatban? A tőzsde is egy olyan piac, mint amilyet akármelyik városban, vagy faluban találhatunk. A különbség az, hogy vásárlás során maga az áru nem materiális formában jelenik meg, illetve nem csak az adott környezettel, hanem az egész világgal lehetőségünk van kereskedni. Viszont, hogy mibe érdemes fektetni az bonyolultabb, mint elsőre képzelnénk.

A tőzsdék fajtái lehetnek:

- Árutőzsde
- Értéktőzsde, ezen belül:
  - Devizatőke
  - Értékpapírtőzsde
  - Hírpiac

2.2.1. Részvény [5]

A részvény tulajdonjogot és egyéb gazdasági jogokat megtestesítő értékpapír, nincsen lejárati ideje. Ezeket az értékpapírokat a részvénytársaságok a részvény birtokosainak,

annak részesedése arányában osztják szét jövedelem és szavazati jogok arányos formájában.

A részvények nagyobbrészt névre szóló értékpapírok, amelyeket leginkább elektronikus úton, dematerizált formában állítanak elő, hogy kereskedelmi forgalomba hozhatóak legyenek. A fizikai értékpapírok tárolása leginkább bankokban, vagy takarékszövetkezetekben történik. A részvényekkel tőzsdén kívüli piacokon is kereskednek, amelyekre általában lazább szabályok vonatkoznak, mint a központosított piacokon.

A részvénytulajdonlás jövedelem növelése érdekében történik. Egy vállalat termelő, vagy szolgáltató tevékenységének célja a profitszerzés, hogy ezt elérhetővé tegye részvénytársasággá kell vállalnia. A vállalkozásban lévő részesedést az alapító tagok között szokás meghatározni, hogy mely tag mekkora összeggel járult hozzá az adott vállalkozás elindításához. Abban az esetben, ha csak az alapítók rendelkeznek részesedéssel, akkor zárt részvénytársaságról beszélünk. Amennyiben az alapítók úgy határoznak, hogy a céget a nyilvánosság elé tárják, akkor zártkörű helyett a cég elkezd részvényeket kibocsájtani. Ezen részvények összessége arányos azzal a tulajdonjoggal, amiről az alapítók lemondtak.

Többfajta részvénytípusok elérhetőek, a teljesség igénye nélkül felsorolok párat, például: befektetési életbiztosítások, **befektetési alap**ok, nyugdíj célú megtakarítások.

2.3. Befektetési alapok [6] [7]

Azon személyek, akik pénzüket beszerették volna fektetni valamilyen szolgáltatásba, már nagy eséllyel hallottak a befektetési alapokról. Ezen termékek kedvezőek és meggyőzőek lehetnek bárki számára. A tőzsde egyik „alapszabálya”, hogy befektetéskor nem ajánlott egyetlen terméket vásárolni. A diverzifikáció ugyanis nagyon fontos, hiszen nem csak ezzel csökkenthetjük a kockázatot, hanem sokkal több tapasztalatot is tudunk gyűjteni és minél több csapot nyit meg a befektető annál nagyobb eséllyel lesz sikeres.

A befektetési alap olyan vagyontömeg, amely állhat értékpapírból, ingatlanokból, bankbetétekből és részvényekből. Befektetési alapot csakis befektetési alapkezelő hozhat létre és kezelhet. Egy befektetési alapkezelő több befektetési alapot is kezelhet. A befektetési alapok futamideje változó, beszélhetünk akár pár hónapos futamidőről, de akár évtizedekig tartó időintervallumról is. A rövid futamidejű alapokat likvidálási alapoknak hívjuk és mindig nyitott végű alapok. Mit jelent az a kifejezés, hogy *nyitott végű alap*ok?

A befektetési alapokat két fajta típus alapján különböztetjük meg, ezen típusok jelzik, hogyan lehet csatlakozni egy alaphoz.

- **Nyíltvégű alapok**: bármelyik forgalmazási napon lehetőségünk nyílik csatlakozni, illetve vissza is válthatjuk belőle a megtakarításainkat. Ezek az alapok általában határozatlan futamidőre jönnek létre és mi dönthetjük el, meddig tartjuk bent a megtakarított pénzünket. Mindig érdemes figyelni a javasolt befektetési időtávokat, hogy a befektetésünk jó eséllyel az elvárt teljesítményt tudja nekünk szolgáltatni.

- **Zártvégű alapok**: a zártvégű alapokhoz csak az alap indulása előtti jegyzési időszakban van lehetőségünk csatlakozni. Ezen alapok általában határozott futamidőre jönnek létre, tehát az alap lejárattal rendelkeznek, ezen időszakig a

megtakarításainkat az alapban kell tartanunk. Természetesen, ha a futamidő alatt mégis szükségünk lenne a befektetett összegre, a zártvégű alapokat tőzsdei megbízás útján értékesíteni tudjuk, ilyenkor a pillanatnyi kereslet fogja meghatározni a befektetésünk ellenértékét, nem pedig annak az értéke. Ezért, fontos jól megfontolni, ha zártvégű alapot választunk, célszerű és javaslott a befektetésünket a lejáratig megtartani.

Az alapokat az alapján is csoportosíthatjuk, hogy azok milyen típusú eszközbe fektetik a befektetők megtakarításait.

- **Pénzpiaci befektetési alap**: pénzpiaci alapok, állampapírok és különböző banki betétek.
- **Kötvényalapok**: különböző kötvények vásárolhatók.
- **Részvényalapok**: vállalatok által kibocsátott részvények.
- **Vegyes alapok**: többféle portfólióban csoportosított részvények és kötvények.
- **Ingatlan befektetési alap**: már megépült vagy építésben levő ingatlanokba való befektetés.
- **Speciális befektetési alapok**: abszolút hozamú, tőkevédett és származtatott alapok.

Mivel ilyen sokrétű lehet egy befektetési alap, így felmerülhet a vásárlóban, hogy miért érdemes befektetési alapba fektetnie a pénzét? Erre a kérdésre az egyik legjobb érv a kockázatmegosztás. Ugyanis már egyetlen befektetési alap megvásárlásával is több egyedi értékpapírból álló, szakszerűen összeállított portfólióhoz juthatunk hozzá, amit egyéni befektetőként rengeteg idő és erőfeszítés árán révén érhetnénk el. Amennyiben befektetési alapot választunk megtakarításaink elhelyezésére nincsen más dolgunk, mint vásárolni az alap befektetési jegyeiből. Ezzel az egyszerű tranzakcióval gyakorlatilag megbízunk egy szakértői gárdát, akik az összes a befektetésünkkel kapcsolatos terhet levesznek a vállunkról, és különféle értékpapírokba helyezik el megtakarításaikat.

2.6. ábra. Általam készített weboldal egyik befektetési alapja

2.3.1. Tőzsdei árfolyam [8]

Tőzsdei árfolyamoknak nevezett árfolyamokat a globális pénzügyi piacon határozzák meg, ahol a bankok és más pénzügyi intézmények éjjel-nappal kereskednek valutákkal ezen tényezők alapján. Az árfolyamot általában az általa képviselt nemzeti valuta betűszóval jegyzik, mint például a leggyakrabban használt valuta az *USD*, ami az amerikai dollárt jelenti. Mérésének számos módja van, a leggyakoribb módszer a kétoldalú árfolyam mérése. A bilaterális árfolyam az egyik valuta másikhoz viszonyított értékére vonatkozik. A tőzsdén található árfolyamoknak több típusát különböztetjük meg, ezek az alábbiak lehetnek:

- Nyitó árfolyam
- Napi maximum érték
- Napi minimum érték
- Napi záró érték
- Napi átlag érték

3. fejezet

Specifikáció

Az előző fejezetben taglaltam milyen fogalmakkal találkozhat a felhasználó. Ebben a fejezetben ismertetem a szoftver tervezésének lépéseit, a felhasznált technológiákat technikai hátterét, továbbá bemutatom az általam választott grafikus könyvtárat a Chart.JS-t és hogy miért erre esett a döntésem.

3.1. Tervezés [9]

A programom tervezése során több szempontot vettem figyelemben. Elődleges szempont volt, hogy webalkalmazásként működne legjobban. Ezt az érvet azzal tudom alátámasztani, hogy szem előtt tartva az előnyeit (könnyű hozzáférés, naprakész adatok, jól szemléltethető, skálázható), illetve személyes szakmai tudásomat is a frontend fejlesztés során szereztem. Másodlagos érvemet pedig azzal tudom alátámasztani, hogy az egyetemi pályám során is webfejlesztésre szakosodtam, így a legkézenfekvőbb lehetőség mellett tettem le a voksomat.

Az alkalmazás tervezésekor az egyik általam ismert agilis módszertan szerint dolgoztam, amelynek neve *Extreme Programming* (XP). Az Extreme Programming egy specifikus keretrendszer, amelynek célja nem csak a kiváló minőségű szoftver(ek) előállítása, hanem az egész folyamat megkönnyítése is a fejlesztő számára. A fejlesztési folyamatomat 5 fázisra lehet bontani.

Tervezés (planning)

- Piackutatás
- Funkciók meghatározása
- Szerver-kliens architektúra megtervezése
- Felhasználandó technológiák

Elemzések (analysis)

- Alkalmazás struktúrákra felosztása
- Elkészüléshez szükséges idő meghatározása
- Szakmai tudás felmérése

- Erőforrás tervezés fejlesztői oldalon

Design

- A feladatok lebontása
- Összpontosított megjelenés létrehozása és végrehajtása

Végrehajtás (execution)

- Kódolás
- Kész egységek tesztelése
- Hibajelentés generálása
- Iteráció közbeni és végi áttenkintés
- Folyamatfejlesztések

Zárás (closure)

- Az elkészült termék üzembe helyezése
- Felhasználói kézikönyv
- Végső tesztelés

3.2. Felhasznált technológiák

3.2.1. HTML5

Mivel a programom online felületre lett tervezve, annak szerkezeti felépítése HTML-ben íródott. A HTML azaz Hypertext Markup Language, magyarul hipertext jelölő nyelv, egy leíró nyelv, amely egy weblap felépítését specifikálja az internetes böngészők felé. Ahogy nevében is láthatjuk, nem egy programozási nyelv, ez azt jelenti, hogy nem programozási logikák megírására, vagy adatok kezelésére használjuk, hanem különböző utasításokkal meghatározzuk az oldalunk struktúráját.

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

3.1. ábra. HTML felépítés és elemek megjelenítése

HTML Canvas

Canvas elem a HTML5 része és lehetővé teszi két dimenziós alakzatok és dinamikus bittérképek megjelenítését. Több elterjedt 2D API-khoz hasonló rajzfunkciók teljes készletén keresztül érheti el a felhasználni kívánt területet, ezzel elérhetővé téve a dinamikusan generált grafikákat. Főbb felhasználási területe: grafikonok, animációk, játékok és képalkotás.

Mivel grafikonok megvalósításához elengedhetetlen, ezért az én applikációm részét is képezi. A Canvas elem a következőképpen épül fel:

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
</canvas>

</body>
</html>
```

3.2. ábra. Egy üres négyzet, megvalósítva Canvas elem által

3.2.2. CSS

Ahhoz, hogy a HTML által definiált oldalunkon megjelenjenek a kívánt elemek és ezeket jól olvashatóan és igényesen tudjuk megformázni, szükségünk van egy olyan nyelvre, ami a HTML elemekre tud hivatkozni, és különböző nézeti tulajdonságokat tud neki átadni. Ezen célra alkalmas a CSS azaz Cascading Style Sheets, magyarul lépcsőzetes stíluslapok, egy leíró nyelv, amely a HTML elemek megjelenítését, azok stílusát (például: elhelyezkedés, betűszín, karakter formátumok stb.) írj a le a böngésző számára.

Különböző választókkal jelölhetünk ki elemeket, azok nevére, osztályára vagy típusára hivatkozva. A lépcsőzetesség arra utal, hogy mely választók vannak priorizálva a weblap megjelenítésének szempontjából. Ez fontos, hisz összetett weblapok esetén többszáz CSS szabály is használatban lehet.
CSS stílusokat megadhatunk különféle eljárások szerint. a 3.3

Közvetlenül egy HTML elemen keresztül pontosvesszőkkel elválasztva. Ezt a változatot *sorközi stílus*nak (inline CSS) nevezzük.

Használhatjuk *belső stílus*ként (internal CSS), amely egyedi kinézetet biztosít egyetlen dokumentumhoz. A HTML <head> részében kell meghatározni <style> címkén belül.

Szakmailag legelterjedtebb felhasználása a *külső stílus* (external CSS). A külső stíluslapot általában akkor használjuk, ha több oldalon szeretnénk változtatni. Ideális erre az állapotra, mert megkönnyíti a teljes webhely megjelenésének megváltoztatását egyetlen fájl módosításával. Fejrészen belül elhelyezzük <link> címkék közé a külső fájl forrását, amiben a kódunk szerepel.

3.3. ábra. Különböző CSS elérési útvonalak. (forrás: [15])

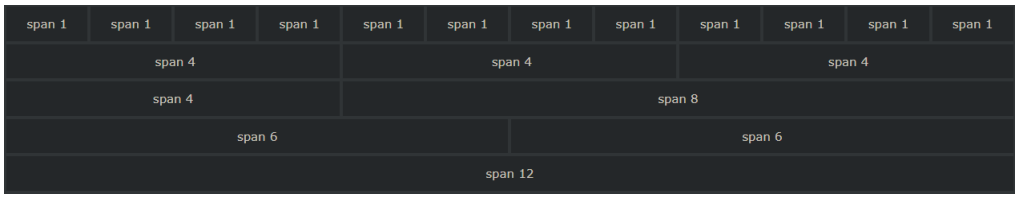
Bootstrap5

A Bootstrap egy ingyenes, nyílt forráskódú frontend fejlesztői keretrendszer webhelyek és webes alkalmazások létrehozásához. Úgy tervezték, hogy lehetővé tegye a webhelyek reszponzív fejlesztését , és a Bootstrap szintaxis gyűjteményt biztosít a sablontervekhez, így a fejlesztőknek csak be kell illeszteni a kódot egy előre meghatározott CSS-keretrendszerbe (grid). Ez a rendszer 12 oszlopos rácsrendszert használ, ezáltal egy reszponzív weboldalt több módon egyenletesen lehet felosztani, ahogy az eszköz nézete megváltozik különféle felbontásban.

3.4. ábra. Bootstrap beillesztése HTML dokumentumban és rácsrendszer felosztása

Rácsrendszer: A rácsrendszerek olyan segédeszközök, amelyeket a tervezők a nehezen átlátható weboldalak megoldásaként valósították meg, hogy az információk rendezettek legyenek és következetes felhasználói élményt biztosítsanak a felhasználók

számára. Logikáját tekintve szakított az informatikában gyakran használt decimális számokkal, mivel ezen számokat nem lehet felezni, harmadolni, úgy, hogy egész számokat kapjunk. Így merült fel az a meglátás, miszerint a tároló konténert 12 egyenlő részre osztja fel. Szabadon beállíthatjuk az adott sor(ok), vagy oszlop(ok) milyen szélességgel rendelkezzenek, esetleg a tároló kontéren mekkora részének feleljen meg a képernyőn.



3.5. ábra. 12 oszlopos rácsrendszer felosztása (forrás: [16])

Font Awsome

A legegyszerűbb módja annak, hogy figyelemfelkeltő ikonokat helyezzünk el weboldalunkon, ha ikon készletet használunk. Én a Font Awsome internetes ikonkönyvtárát és eszközkészletét használtam fel.

3.2.3. NodeJS [10]

A Node.js egy nyílt forráskódú, többplatformos futási környezet amiben a gépünkön JavaScript kódot tudunk végrehajtani. Széles körben használják szerveroldali programozáshoz, így a fejlesztők használhatják a JavaScriptet kliensoldali és szerveroldali kódokhoz anélkül, hogy további nyelvet kellene megtanulniuk. Általában egy bizonyos címen és porton figyel az szerver, és mikor kérést kap, elvégzi a leprogramozott funkciókat, majd tétlen állapotban várakozik a következő hívásig. Mivel a Node.js nem folytat közvetlenül kimeneti, vagy bementi műveleteket, így erőforrások blokkolása nem merül fel, ezáltal nem kell félni, hogy a program elakad. Tehát a Node.js ideális interaktív megjelenítő alkalmazások fejlesztésére.

Node Package Manager

A Node.js futtatásához Node Package Manager (NPM), azaz Node Csomagkezelőt alkalmazunk. Az NPM egy nyílvántartott adatbázis, amelyben szotfverek és a hozzájuk található metaadtok szerepelnek. Három részre lehet felosztani, úgy mint weboldal kezelés, parancssori felület és nyilvántartás.

Express.JS

Az Express.js egy Node.js backend keretrendszer, amelyet arra terveztek, hogy az API webalkalmazásait gyorsan és platformokon átívelő almazásokat készítsen, és megkönnyítse a node js-t dolgát. Webes alkalmazások és RESTful API-k építésére tervezték anélkül, hogy bármiben is korlátozná a szerver funkcióit.

3.2.4. JavaScript [11]

A JavaScript, amelyet gyakran JS-ként is szoktak említeni, egy olyan többparadigmás, dinamikus programozási nyelv, ami lehetővé teszi, hogy a weblapokon komplex funkciókat valósíthassunk meg HTML és CSS használatával. A webhelyek majdnem 100%-a JavaScriptet használ az elsőrendű függvényeivel, továbbá az alacsony erőforrásigényével. Támogatja a prototípus-alapú objektumorientációval és első osztályú funkciókkal rendelkező kódolást, továbbá az eseményvezérelt, funkcionális és kötelező programozási stílusokat.

Működését tekintve, nem típusos nyelv, így nem szükséges a változókat deklarálni használat előtt, a megfelelő típust futás közben veszi fel. A JavaScriptet támogató böngésző betölti a HTML oldalt, amennyiben script kód található benne a JavaScript elemző motor kerül előtérbe és a script kód betöltődik. Alkalmazásprogramozási felületekkel (API-kkal) rendelkezik a szöveggel, dátumokkal, reguláris kifejezésekkel való munkához. A legnépszerűbb futásidejű rendszer erre a felhasználásra a Node.js.
(Bár a Java és a JavaScript nevüket és szintaxisukat tekintve hasonlóak, a két nyelv különbözik, és nagymértékben eltér egymástól.)

JQuery

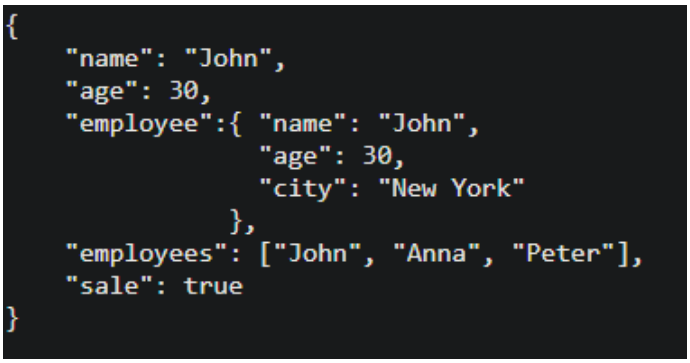
A jQuery egy nyílt forráskódú, gazdag JavaScript-keretrendszer, amely leegyszerűsíti a HTML/CSS-dokumentumok, pontosabban a Dokumentum Objektum Model (DOM) és a JavaScript közötti interakciókat. Szintaxisát úgy tervezték, hogy megkönnyítse a dokumentumban való navigálást, kezelést, a DOM- elemek kiválasztását, az animációk létrehozását, az események kezelését és az Aszinkron JavaScript XML (Ajax)- alkalmazások fejlesztését.

A jQuery lehetőséget biztosít, a fejlesztők számára, hogy absztrakciókat hozzanak létre alacsony szintű interakcióhoz és animációhoz, fejlett effektusokhoz és magas szintű, témára alkalmas widgetekhez. A könyvtár moduláris megközelítése lehetővé teszi hatékony dinamikus weboldalak és webalkalmazások létrehozását.

JSON

A JavaScript Objektum Jelölés (JSON) egy nyílt szabványos, nyelvtől független fájlformátum és adatcsere -formátum, amely ember által olvasható szöveget használ az adatobjektumok tárolására és továbbítására. Nagyon elterjedt adatformátum, amelyet sokrétűen alkalmaznak az elektronikus adatcserében, beleértve az adatcserét webes alkalmazások és szerverek között. A JSON két struktúrára épül:

Név - érték párok gyűjteménye. Különböző nyelveken ez objektumként , rekordként, struktúraként, szótárként, hash-táblaként, kulcsos listaként vagy asszociatív tömbként valósul meg . Az értékek listája rendezett. Ezen értékek lehetnek, sztingek, számok, másik JSON, tömbök, vagy boolean értékek. A .json fájlnevet ezen objektumok kiterjesztésére használják, nem csak JavaScript programozási nyelven íródott kódokra is.



3.6. ábra. Egy alkalmazott adatai JSON-ként tárolva (forrás: [?])

3.3. Miért a Chart.JS? [12]

3.3.1. ChartJS bemutatása

A Chart.js egy ingyenes, nyílt forráskódú JavaScript-könyvtár adatvizualizációhoz , amely nyolc diagramtípust támogat : vonal, oszlop, terület, torta (fánk), buborék, radar, poláris és szórt diagram. JavaScript nyelven használt beépülő modulokat, diagramtípusokat és testreszabási lehetőségeket kínál. A számtalan beépített modul mellett, a közösség által karbantartott diagramtípusokat is elérhetővé tesz. Ezen felül több diagramtípus kombinálható vegyes diagrammá ugyan azon a Canvas elemen belül.

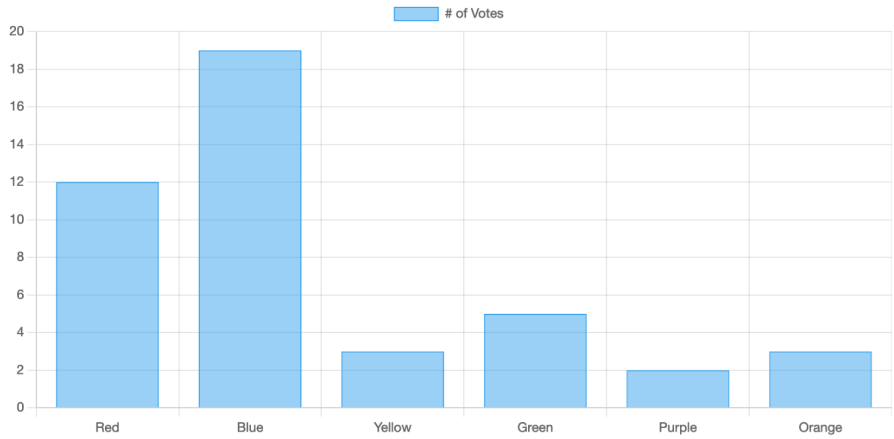
```
<div>
  <canvas id="myChart"></canvas>
</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>
  const ctx = document.getElementById('myChart');

  new Chart(ctx, {
    type: 'bar',
    data: {
      labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple', 'Orange'],
      datasets: [{
        label: '# of Votes',
        data: [12, 19, 3, 5, 2, 3],
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
</script>
```

3.7. ábra. diagram létrehozása JSON formátumban (forrás: [12])



3.8. ábra. Oszlopdiagram ami szavazatok számát mutatja meg (forrás: [12])

3.3.2. ChartJS összehasonlítása más grafikus könyvtárakkal [13]

Chart.JS összehasonlítása D3.JS keretrendszerrel

A csillagok száma alapján a GitHubon a második legnépszerűbb JavaScript diagramkönyvtár a D3.js után, amelyet ugyan lényegesen könnyebben használhatónak tartanak, de kevésbé testreszabható, mint az utóbbi. A Chart.js HTML5 Canvas elemen belül jelenik meg, és széles körben elismert, mint az egyik legjobb adatvizualizációs könyvtár. A Chart.js nagymértékben testreszabható, hogy néhányat említsek: egyéni beépülő modulokkal, amelyek segítségével megjegyzéseket, nagyítást vagy húzással hozhat létre, hogy néhány dolgot említsünk

A Chart.js a diagramelemeket HTML5 Canvas jeleníti meg, ellentétben számos más, többnyire D3.js-alapú diagramkönyvtárral, amelyek Skálázható Vektor Grafika (SVG)-ként jelennek meg. A Canvas megjelenítés nagyon hatékonyvá teszi a Chart.js-t, különösen nagy adatkészletek és összetett vizualizációk esetén, amelyek egyébként több ezer SVG-csomópontot igényelnének a DOM-fában. Ugyanakkor a Canvas renderelés nem engedélyezi a CSS stílust, ezért ehhez beépített opciókat kell használni, vagy különböző diagramtípusokaz kell létrehozni, hogy a kívánt gráfokat jelenjenek meg.

A Chart.js nagyon jól használható nagy adathalmazokhoz. Az ilyen adatkészletek hatékonyan feldolgozhatóak belső formátum használatával, így eredményes az adatok elemzéséhez és normalizálásához. Végül a Chart.js által használt Canvas megjelenítés csökkenti a DOM-fájának terhelését, míg az SVG-megjelenítéshez nagyon erőforrás igényes tud lenni.

ChartJS előnyei

- Mivel a Chart.js egy JavaScript könyvtár, így lehetővé teszi, hogy bármilyen választott JS keretrendszerrel használjuk, mint például az Angular.js vagy a React.js..
- Használata könnyű és minimális fejlesztési ismeretet igényel.
- Nyílt forráskódú, így csak a meglévő könyvtárat kell felhasználni és nem a fejlesztőnek kell mindent felépíteni.

- World Wibe Web Consortion (W3C) szabványt követi, így a használatához nincs szükség más böngészőhöz tartozó technológiára, vagy bővítményre.
- Több megjelenítő eszközzel ellentétben 8 féle diagram megjelenítését engedélyezi.
- Adatok hatákony manipulálását teszi lehetővé. Rendkívül gyorsan jelenik meg és saját animációkkal rendelkezik.
- Kiváló dokumentációval van ellátva.

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] Forrás: https://www.portfolio.hu/prof/20181005/a-grafikon-ami-eleteket-mentett-299542, *Felhasználtam a történet bevezetését.*

[2] Forrás: https://hu.wikipedia.org/wiki/Grafikon_(matematika), *Felhasználtam az oldalon található grafikon megfogalmazását.*

[3] Forrás: https://www.uni-miskolc.hu/~wwwfemsz/exc5.htm, *Felhasználtam Dr. Szabó László: Segédlet az Excel 5.0 használatához leírását a grafikon típusokhoz.*

[4] Forrás: https://hu.wikipedia.org/wiki/TÅŠzsde, *Felhasználtam az oldalon található tőzsde megfogalmazását.*

[5] Forrás: https://hu.wikipedia.org/wiki/RÃľszvÃľny, *Felhasználtam az oldalon található részvény leírását.*

[6] Forrás: https://hu.wikipedia.org/wiki/BefektetÃľsi_alap, *Felhasználtam az oldalon található befektetési alap megfogalmazását.*

[7] Forrás: https://www.otpbank.hu/otpalapkezelo/hu/Befektetesi_alapok/Tudnivalok, *Felhasználtam az oldalon található Tudnivalók leírását.*

[8] Forrás: https://docplayer.hu/7069114-Arfolyamok-miskolci-egyetem-mesterkepzes.html, *Felhasználtam az PDF-ben lévő árfolyamok leírását.*

[9] Forrás: https://promanconsulting.hu/mi-az-agilis-modszertan-legelterjedtebb-agili
Felhasználtam az oldalon található Extreme Programming leírását.

[10] Forrás: https://nodejs.org/en/about, *Felhasználtam az oldalon található Node.js leírását.*

[11] Forrás: https://en.wikipedia.org/wiki/JavaScript, *Felhasználtam az oldalon található JavaScript, JQuery és JSON leírásokat.*

[12] Forrás: https://www.chartjs.org/docs/latest/, *Felhasználtam az oldalon található leírásokat.*

[13] Forrás: https://en.wikipedia.org/wiki/Chart.js, *Felhasználtam az oldalon található adatokat.*

[14] Forrás: https://dataschools.education/history-of-graphs/.

[15] Forrás: https://www.bitdegree.org/learn/inline-css, *Felhasználtam az oldalon található CSS elérési útvonalakat.*

[16] Forrás: https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp, *Felhasználtam az oldalon található képet.*

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).