|  |  |
| --- | --- |
| A képen embléma, szimbólum, címerpajzs, jelvény látható  Automatikusan generált leírás | EÖTVÖS LORÁND TUDOMÁNYEGYETEM  Informatikai Kar  Média- és Oktatásinformatika Tanszék |

Kriptovaluták árfolyamának elemzése és korrelációk keresése SAP –

Capireban

|  |  |
| --- | --- |
| Témavezető: Dr Vincellér Zoltán  egyetemi mesteroktató | Szerző: Téglás Tamás  programtervező informatikus BSc. |

Budapest, 2023

Tartalomjegyzék

[Fejlesztői környezet 4](#_Toc136024970)

[Frontend fejlesztése 4](#_Toc136024971)

[Backend fejlesztése 4](#_Toc136024972)

[Szkriptek fejlesztése 5](#_Toc136024973)

[Továbbfejlesztési lehetőségek 6](#_Toc136024974)

[Grafikonok 6](#_Toc136024975)

[Adatok 6](#_Toc136024976)

[Adatbázis 6](#_Toc136024977)

[Kitelepítés (deploymemt) 6](#_Toc136024978)

[Felhasználók – Autentikáció 6](#_Toc136024979)

[Funkcionalitás 6](#_Toc136024980)

[Elemzés – Predikció 7](#_Toc136024981)

[Tesztelés 8](#_Toc136024982)

# Fejlesztői környezet

A program fejlesztése során egy chromium alapú Brave nevű böngészőt használtam. A kód írásához Visual Studio Code javasolt. A program futtatásához, teszteléséhez egy terminál szükséges, futtatáshoz elegendő egy *Powershell* terminál*,* de fejlesztéshezszükséges lehet több terminál ablak, vagy a *Windows terminál* nevű program, mely alapértelmezetten megtalálható a *Windows 11* operációs rendszerben.

## Frontend fejlesztése

A frontend fejlesztéséhez elég megnyitni a *react-ui* mappát, a frontend összes eleme ott található. Mielőtt a fejlesztést megkezdnénk, ajánlott a terminálban, ugyanebben a mappában lefuttatni az *npm install* parancsot, ami az internetről leszedi az frontend fordításához és futtatásához szükséges összes csomagot. Amint ez megvolt, kiadhatjuk az *npm start* parancsot mely elindítja a *React* fejlesztői szerverét a *http://localhost:3000* címen. Itt meg fog jelenni a frontend jelenlegi kinézete, és minden alkalommal amikor frissítjük és elmentjük valamelyik forrásfájlt, az frontend újra lefordul, és frissül is a megadott címen, ahol böngészővel meg tudjuk tekinteni a fordítás eredményét. Amennyiben meg vagyunk elégedve a frontenddel, az *npm run build* parancs lefordítja, és becsomagolja nekünk a kódunkat, melyet az alkalmazás futáskor fel tud használni. Ahhoz, hogy teljes mértékben tudjuk tesztelni a frontend kódját, a backendet is el kell hozzá indítani, hiszen nélküle nem tudunk kommunikálni az adatbázissal, és nem tudunk adatokat vagy grafikonokat lekérni megjelenítésre.

### Backend fejlesztése

A backend fejlesztéséhez elég megnyitni a *crypto-gaze* nevű mappát, a backend összes eleme ott található. Hasonlóan ahogy a frontendnél, itt is ajánlott a terminálban, ugyanebben a könyvtárba futtatni az *npm install* parancsot, mely a backend által használt csomagokat letölti az internetről. A backendet fejlesztés során tudjuk perzisztens, és nem perzisztens módon futtatni. Amennyiben nem perzisztens módon szeretnénk, a *cds watch* parancsot kell kiadni a terminálban. Ez a parancs elindítja a backendet egy memóriába töltött *SQLite* adatbázissal, ami addig őrzi meg az adatokat amíg le nem állítjuk. Ide bekerülnek kezdőadatként a *\db\csv* mappában található *CSV* fájlok. Hasonlóan, mint a *React* fejlesztői szervernél, itt is minden forrásfájl módosítására az backend újraindul, de a backend mindig a *http://localhost:4004* címen fog elindulni és futni. Ha perzisztenciával szeretnénk fejleszteni, először is, amennyiben nincs még létrehozva adatbázisfájl, le kell futtatnunk a *cds deploy –to sqlite* parancsot, ami létrehozza az adatbázisfájlt. és beletölti a *\db\csv* mappában található *CSV* fájlokat. Ezután a *cds serve* vagy a *cds run* parancsot kell kiadni, és a backend el is indul. Itt viszont minden módosítás életbeléptetéséhez szükséges a backendet újraindítani.

### Szkriptek fejlesztése

Az alkalmazást támogató szkriptek fejlesztéséhez elég megnyitni a *python\_scripts* mappát, mivel ott található az össze szkript amit az alkalmazás felhasznál. Ezek a rövid kódok mind szoros kapcsolatban állnak a backend végpontjaival, és az adatbázissal is, így futtatásukhoz és tesztelésükhöz is egyaránt szükséges, hogy a backend fusson.

A szkripteket lehet kézzel futtatni a terminálon keresztül, illetve a frontenden keresztül, a *Run script* menüpontban. Amennyiben a terminálon keresztül futtatjuk, a szkript kimenete a *standard output*-on megtalálható, ahová, ha hibába ütközne, azt is kiírja. Amikor a szkriptet a frontenden keresztül futtatjuk, ott viszont a kimenetel látszódni fog a böngészőablakban, illetve a terminálon, ahol a backendet indítottuk is.

# Továbbfejlesztési lehetőségek

A projekt moduláris mivoltija miatt több különböző fejlesztési irány áll rendelkezésünkre.

A *CDS* segítségével könnyen adhatunk modelleket a backendbe, szkriptek használatával extra adatmanipulációs funkciókat tudunk implementálni párhuzamosan. A *React* frontend komponensekből való összeállása miatt, minimális változtatásokkal tudunk hozzáadni új oldalakat, menüpontokat.

## Grafikonok

Gyertyagrafikonok segítségével összetetteb képet lehet adni a különböző valutákról. A jelenlegi *React* csomagok között nem nagyon található gyertyagrafikon, melyen több különböző valuta összehasonlítását lehet kijelezni.

## Adatok

A *CDS* lehetőséget ad számunkra, hogy statikus értékek helyett, élőben webszolgáltatásokon keresztül tudjunk különböző szolgáltatóktól adatokat lekérni.

## Adatbázis

A *Capire* szolgáltat nekünk különböző eszközöket, melyekkel könnyen tudjuk az alkalmazást csatlakoztatni egy *Hana* adatbázishoz, mely során a backend funkciókat fel is tudjuk tölteni a *SAP* felhőbe, és ott *hostolni.*

## Kitelepítés (deploymemt)

Az alkalmazást lehetőségünk van konténerizálni, és kitelepíteni különböző felhőszolgáltató cégek rendszerébe, hogy folyamatosan, a világ bármely pontjáról elérhető legyen.

## Felhasználók – Autentikáció

Amennyiben szeretnénk, hogy az alkalmazást több felhasználó használja, lehetőségünk van a *SAP* által készített *XSUAA* rendszert használni, mellyel könnyen hozzá tudunk az alkalmazáshoz felhasználókat adni, és a végpontokat felhasználókra, és jogkörökre korlátozni.

## Funkcionalitás

Az *URL* melyet a letöltési szkript konstruál, kriptovalutákra van állítva, de minimális változtatással rá lehet állítani, hogy részvényeket, vagy *fiat* valutákat töltsön le. A grafikonokat is be lehet állítani, hogy más modellek értékeit mutassák.

## Elemzés – Predikció

A *python* szkriptek fejlesztése során, több elemzési kalkulációt is hozzá lehet adni, mely akkurátusabb képet nyújt a kriptovaluta állapotáról. A különböző modellek használatával, pedig pontosabb predikciót is hozzá lehet adni. Ha a backend pedig egy szervergépen futna, nagyobb erőforrás igényű gépi tanulásos algoritmusokat is lehet az előrejelzésekhez alkalmazni.

# Tesztelés

## Frontend tesztelése

A frontend működését manuális tesztelés során biztosítottam, itt főleg a felhasználói felület elemeinek működése került tesztelésre. Itt tesztelésre került néhány végpont funkcionalitás is a különböző funkciók tesztelése során.

|  |  |  |  |
| --- | --- | --- | --- |
| Műveletsorozat | Környezet | Elvárt eredmény | Valódi eredmény |
| Oldalsávon *Menu* feliratra kattintás | Az oldalsáv normál mérettel jelenik meg | Oldalsáv minimalizálódik |  |
| Oldalsávon *Menu* feliratra kattintás | Az oldalsáv minimalizált mérettel jelenik meg | Oldalsáv normál méretre vált |  |
| Oldalsávon menüpontra kattintunk | Nem azon a menüponton vagyunk amire kattintunk | Az oldalsáv mellett a menüpontnak megfelelő tartalom jelenik meg |  |
| A *Custom Charts* menüben átírjuk a rendezési szempontot | - | A megjenelített grafikonok eltűnnek |  |
| A *Custom Charts* menüben átírjuk a rendezési szempontot, majd megnyomjuk a *Show Charts* gombot | A rendezési szempont melyet beállítottunk más, mint ami eredetileg volt, illetve vannak **nem** rejtett grafikonok az adatbázisban | A grafikonok megjelennek az adott rendezésben |  |
| Egy grafikonon a *Hide Chart* gombot megnyomjuk | A grafikonon megtalálható a *Hide Chart* gomb | A grafikonon megjelenik az *Unhide Chart* gomb, a grafikont csak a *Hidden Charts* menüben lehet megtekinteni, a program átnavigál minket a *Hidden Charts* menübe |  |
| Egy grafikonon az Unh*ide Chart* gombot megnyomjuk | A grafikonon megtalálható az Unh*ide Chart* gomb | A grafikonon megjelenik a H*ide Chart* gomb, a grafikont meg lehet találni a *Custom Charts*, a program átnavigál minket a *Custom Charts* menübe |  |
| Egy grafikonon a *Bookmark* gombot megnyomjuk | A grafikonon megtalálható a *Bookmark* gomb | A grafikonon megjelenik az *Remove Bookmark* gomb, a grafikon megjelenik a *Bookmarked Charts* menüben, a program átnavigál minket a *Bookmarked Charts* menübe |  |
| Egy grafikonon a Remove *Bookmark* gombot megnyomjuk | A grafikonon megtalálható a *Remove* *Bookmark* gomb | A grafikonon megjelenik az *Bookmark* gomb, a grafikont eltűnik a *Bookmarked Charts* menüből, a program átnavigál minket a *Custom Charts* menübe |  |
| A *Create UI5* menüben, a beviteli mezők kitöltése után rákattintunk a *Preview Chart* gombra | A beviteli mezők olyan szimbólumot, és dátumokat tartalmaznak, melyekhez megfelelő adatok megtalálhatóak az adatbázisban | A beviteli mezők alatt megjelenik a grafikon előnézete, és legalább egy adatpont megjelenik rajta |  |
| A *Create UI5* menüben, a beviteli mezők kitöltése után rákattintunk a *Add Chart* gombra | A beviteli mezők olyan szimbólumot, és dátumokat tartalmaznak, melyekhez megfelelő adatok megtalálhatóak az adatbázisban | A program átnavigál a *Custom Chart* menüpontra, és a készített grafikon megjelenik a kijelzett grafikonok közt |  |
| A *Create Chart-js* menüben, a beviteli mezők kitöltése után rákattintunk a *Preview Chart* gombra | A beviteli mezők olyan szimbólumot, és dátumokat tartalmaznak, melyekhez megfelelő adatok megtalálhatóak az adatbázisban | A beviteli mezők alatt megjelenik a grafikon előnézete, és legalább egy adatpont megjelenik rajta |  |
| A *Create Chart-js* menüben, a beviteli mezők kitöltése után rákattintunk a *Add Chart* gombra | A beviteli mezők olyan szimbólumot, és dátumokat tartalmaznak, melyekhez megfelelő adatok megtalálhatóak az adatbázisban | A program átnavigál a *Custom Chart* menüpontra, és a készített grafikon megjelenik a kijelzett grafikonok közt |  |
| Az *Update or Delete* menüpontban egy grafikont kiválasztunk, és megnyomjuk a *Delete Chart* gombot | Legalább egy saját készítésű grafikon megtalálható az adatbázisban, és nincs rejtettre állítva | A program átnavigál a *Custom Charts* menüpontra, és a grafikont, melyet kiválasztottunk, nem jelen meg sehol |  |
| Az *Update or Delete* menüpontban egy grafikont kiválasztunk, és megnyomjuk a *Edit Chart* gombot | Legalább egy saját készítésű grafikon megtalálható az adatbázisban | A program, a grafikon típusának megfelelő beviteli mezőket jeleníti meg |  |
| Az *Update or Delete* menüpontban egy grafikont kiválasztunk, megnyomtuk az *Edit Chart* gombot, majd átírunk egy értéket, és megnyomjuk az *Update Chart* gombot | Legalább egy saját készítésű grafikon megtalálható az adatbázisban, és nincs rejtettre állítva, és a változtatás során **nem** állítottuk rejtettre | A program átnavigál a *Custom Charts* menüpontra, és a grafikont, melyet kiválasztottunk, megjelenik a kirajzolt grafikonok között változott konfigurációval |  |
| A *Run Script* menüben az *Add new ticker data* szkriptet lefuttatjuk a *Run Script* gomb megnyomásával | A Ticker mezőbe olyan kriptovaluta szimbólumát írtuk, mely az adatbázisban nem található, és a dátum melyet beírtuk mai nap vagy korábbi dátum | Az adatbázisban megjelenik a kiválasztott kriptovaluta az adott kezdődátumtól kezdődően, és a Home menüpontban megtaláljuk a kriptovaluta ehavi értékének változását mutató grafikont |  |

## Backend tesztelése

A backend és a szkript működését a végpontok viselkedésével ellenőriztük.

# Tapasztalatok

A szoftver fejlesztése során megismerkedhettem újra mélységben a React keretrendszer sajátosságaival, illetve a megannyi különböző csomaggal, melyek segítségével a különböző komponenseket raktam össze, főleg a ui-webcomponents csomaggal, illetve a material-ui csomaggal, melyek a komponensek kinézetének nagyrészéért felelősek.