**NEUMANN JÁNOS EGYETEM**

**GAMF Műszaki és Informatikai Kar**

**Gyakorlatvezető:**

**Dr. Subecz Zoltán**

**Készítette:**

**Tóth Bálint H9H8JP**

**Jáger Attila ALFSKQ**

Tartalomjegyzék

[1. A feladat megfogalmazása 7](#_Toc87710475)

[2. Gépi és nyelvi igények 8](#_Toc87710476)

[3. Fájlszerkezet és funkciók bemutatása 9](#_Toc87710477)

[3.1. Főbb mappák bemutatása 9](#_Toc87710478)

[3.2. Weboldal bemutatása 9](#_Toc87710479)

[3.2.1. Kezdőlap 9](#_Toc87710480)

[3.2.2. Belépés 10](#_Toc87710481)

[3.2.3. Vélemények 11](#_Toc87710482)

[3.2.4. SOAP szolgáltatás 12](#_Toc87710483)

[3.2.5. Magyar Nemzeti Bank SOAP 12](#_Toc87710484)

[3.3. Kód bemutatása 14](#_Toc87710485)

[3.3.1. Kezdőlap 14](#_Toc87710486)

[3.3.2. Belépés 16](#_Toc87710487)

[3.3.3. Vélemények 18](#_Toc87710488)

[3.3.4. SOAP szolgáltatás 19](#_Toc87710489)

[3.3.5. Magyar Nemzeti Bank SOAP 19](#_Toc87710490)

[3.4. Fejlesztési lehetőségek 20](#_Toc87710491)

[3.5. Tesztelés 20](#_Toc87710492)

[4. Hivatkozások, könyvtárak. 22](#_Toc87710493)

[Ábrajegyzék 23](#_Toc87710494)

1. A feladat megfogalmazása

A feladathoz két fős csoportokat kellett alkotnunk és egy fiktív cég részére kellett weboldalt készítenünk.

1. Az első oldalon be kellett mutatnunk a céget.(3.2.1, 3.3.1 Weboldal bemutatása)
2. A weboldalon bejelentkezési és regisztrációs lehetőséget kellett megvalósítani.(3.2.2, 3.3.2 Weboldal bemutatása)
3. Készíteni kellett egy híroldat, ahol a regisztrált felhasználók véleményeket és kommenteket írhatnak. (3.2.3, 3.3.3 Vélemények)
4. Az egész feladatban az objektum-orientált PHP elveit kellett használni.
5. MVC modellt kellett alkalmazni a fájlszerkezetben. (3.1 Főbb mappák bemutatása)
6. Egy oldalon készítenünk kellett egy SOAP szolgáltatást, ami a választott adatbázis tábláit használja fel.(3.2.4, 3.3.4 SOAP szolgáltatás)
7. A Magyar Nemzeti Bank SOAP web-szolgáltatásával kellett lekérdezéseket készíteni.(3.2.5, 3.3.5 Magyar Nemzeti Bank SOAP)
8. A feladatot Internetes tárhelyre is fel kellett tölteni.(4. Hivatkozások, könyvtárak)
9. Verziókövető rendszernek használnunk kellett a GitHub nevű weboldalt.
10. Utolsó feladatként pedig ezt a dokumentációt kellett elkészíteni.

(4. Hivatkozások, könyvtárak)

1. Gépi és nyelvi igények

A program minden operációs rendszeren futtatható és módosítható, ehhez elsősorban egy szövegszerkesztőre és egy böngészőre van szükségünk, ezen felül pedig egy szerver és egy adatbázis kezelőre.

Ehhez mi az Xampp nevű alkalmazást használtuk. A fejlesztés alapja a HTML volt, amihez elsősorban PHP nyelven valósítottuk meg a logikai részeket. Fontos, hogy a legújabb PHP 8-as verziót kell használni és engedélyezni kell a SOAP használatát.

Emellett minimális Javascriptet is használtunk, elsősorban a HighCharts nevű Javascript diagram könyvtárhoz.

A dizájn pedig a Bulma nevű CSS framework használatával készült.

1. Fájlszerkezet és funkciók bemutatása
   1. Főbb mappák bemutatása

A feladatot az MVC modell mintájára készítettük, így a root mappában ez a három fő mappa található controllers, views, és models néven. Ezen felül található még egy img és egy db\_source mappa, ahol az előbbi a weboldalon használt képeket tárolja, míg az utóbbi egy SQL forrásfájlt tartalmaz, amit importálva elkészíthetjük az adatbázist.

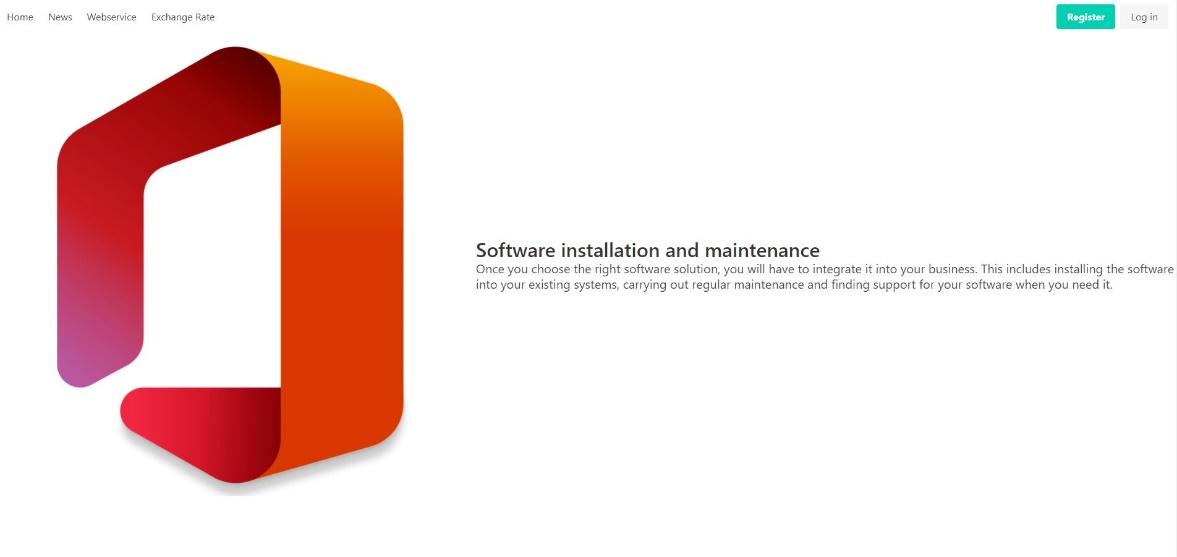
Ezen felül a views mappában találhatóak még további almappák, mivel bizonyos oldalakhoz több kinézet tartozik.(1. ábra)

|  |  |
| --- | --- |
|  | C:\Users\tothb\AppData\Local\Microsoft\Windows\INetCache\Content.Word\views.png |
| Root | Views |

1. ábra: Mappa szerkezet

* 1. Weboldal bemutatása
     1. Kezdőlap

A weboldal a Home nevű oldallal indul el, ez az első feladat megoldása, röviden pár sort írunk a fiktív cégről. (2. ábra)



2. ábra: Kezdőlap

* + 1. Belépés

Ilyenkor még nincs bejelentkezve a senki, ezt a jobb felső sarokban lehet megtenni, míg a baloldalon további menüpontok közül lehet választani.

Előszőr szükséges lesz regisztrálni, ezután tudunk bejelentkezni a weboldalra.(3.ábra)

|  |  |
| --- | --- |
|  |  |
| Regisztráció | Bejelentkezés |

3. ábra: Belépés

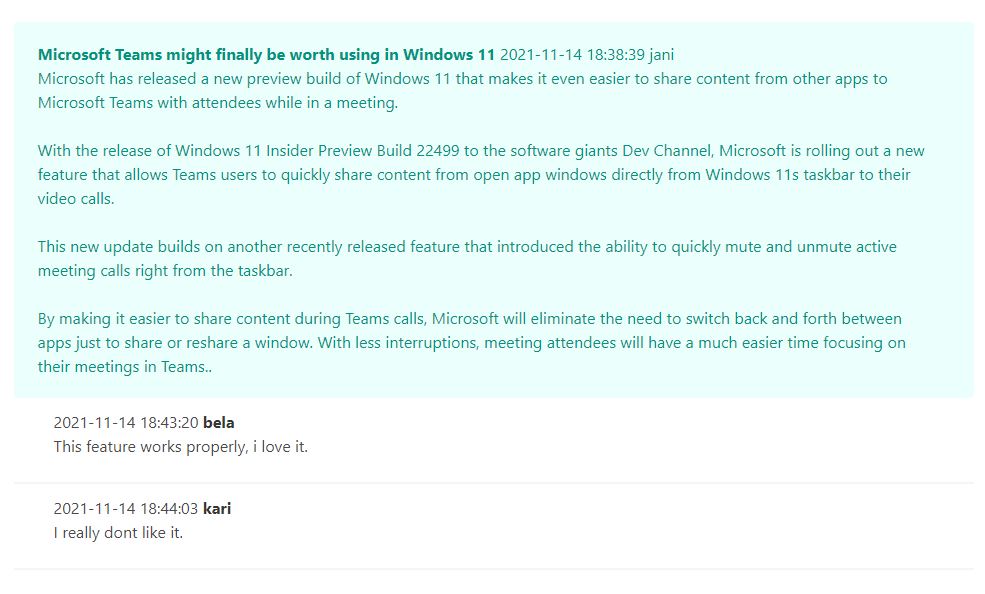
Ha sikeresen bejelentkeztünk, akkor a jobb felső sarokban a gombok megváltoznak egy kijelentkezési gombra és a feladatban megadottak szerint megjelenik a bejelentkezett felhasználó- és teljes név.(4. ábra)

|  |  |
| --- | --- |
|  |  |
| Kijelentkezve | Bejelentkezve |

4. ábra: Belépés

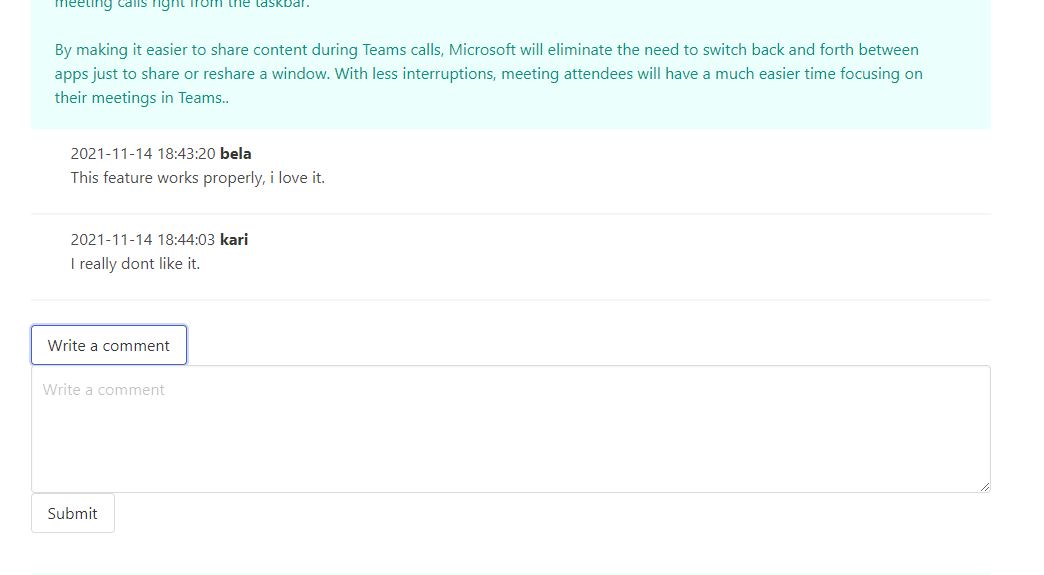
* + 1. Vélemények

A News menüpontra kattintva jelenik meg a véleményezési menü, itt ha nem vagyunk bejelentkezve, csak olvashatjuk a kommenteket.(5. ábra.)



5. ábra: Kijelentkezve a vélemények

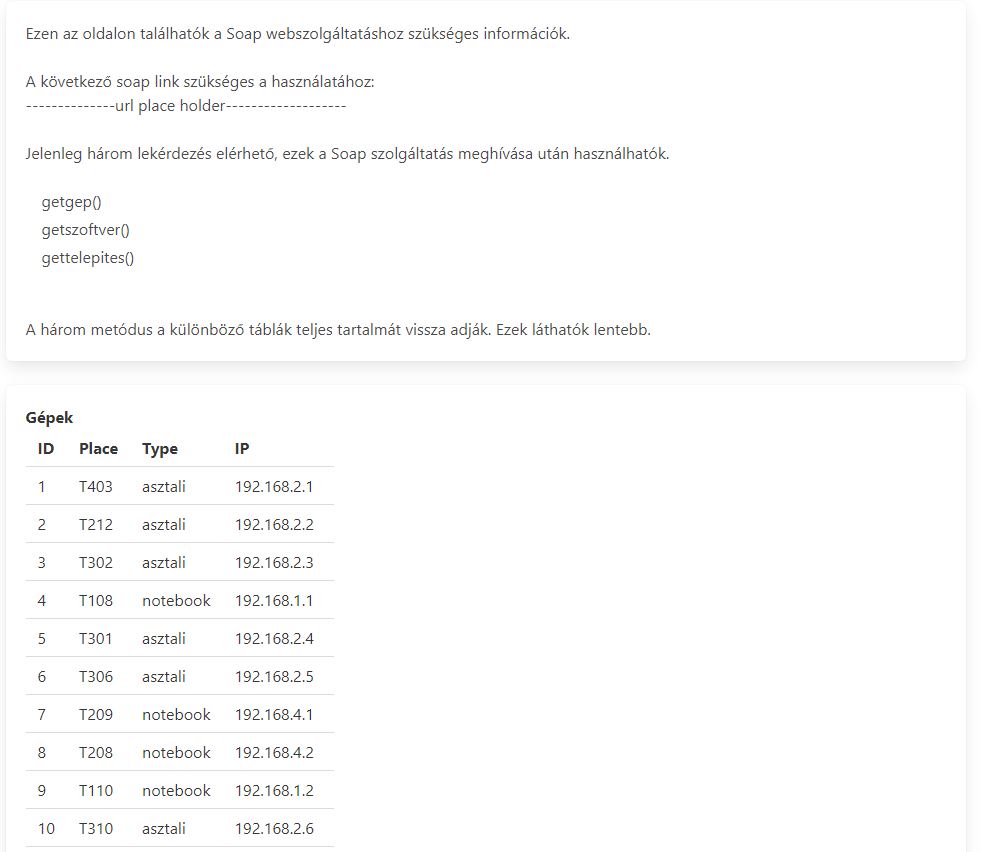
Ám ha bejelentkezünk, akkor megjelenik a véleményezési lehetőség és egy gomb a meglévő vélemények alatt, amire rákattintva megjelenik a kommentálási lehetőség is.(6. ábra.)



6. ábra: Bejelentkezett vélemények

* + 1. SOAP szolgáltatás

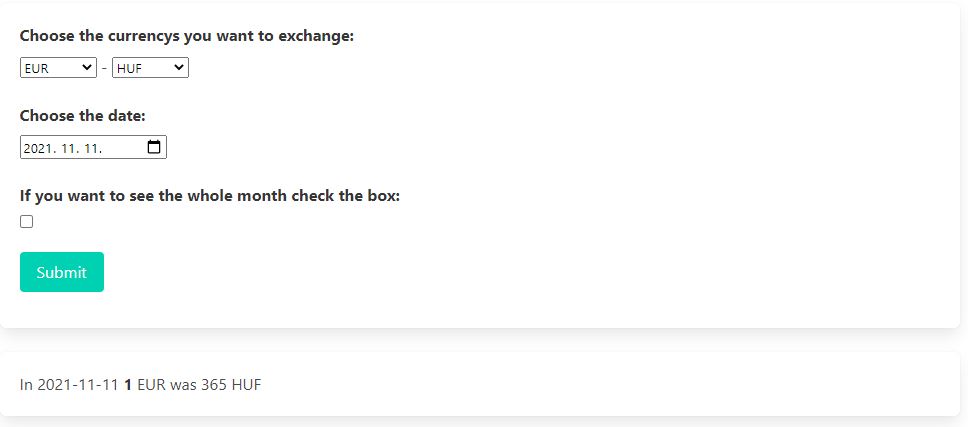
Ennek a szolgáltatásnak az érdemi részét majd a kód bemutatásánál fejtem ki jobban, mivel a szerver oldali része itt nem látszik. A Webservice menüpont alatt egy teszt klienst láthatunk, ami ezt a szolgáltatást használja, és írja le a fontosabb jellemzőit. Összesen három meghívható metódusa van, ami lekéri az adatbázis három táblájának értékeit, a weboldal pedig megjeleníti ezeket.(7.ábra.)



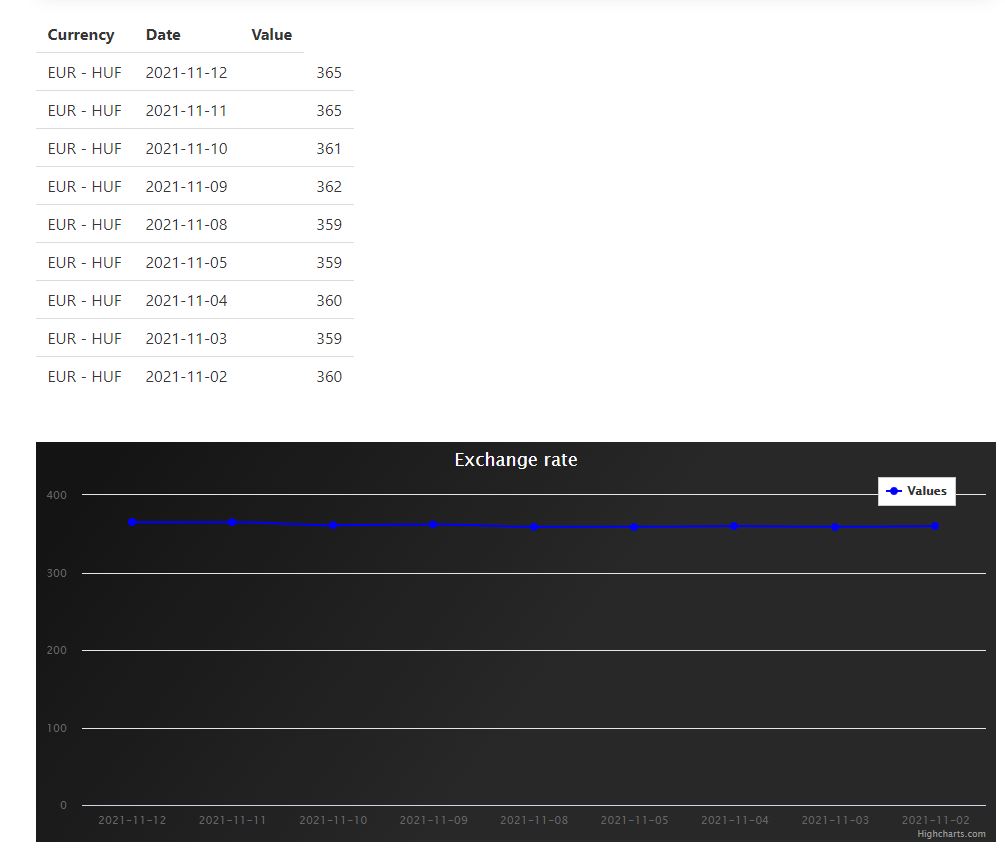
7. ábra: SOAP Webszervíz

* + 1. Magyar Nemzeti Bank SOAP

Ebben a feladatban a magyar nemzeti bank SOAP szolgáltatását kellett használni, amivel valuták értékét tudhatjuk meg adott napokon. Két lenyíló listából választhatjuk ki a kívánt valutákat, majd alatta az időpontot, ha kiválasztottunk egy napot és nem jelöljük be a négyzetet, akkor azt a napot kapjuk vissza, ha történt aznap jegyzés.(8. ábra.) Amennyiben kiválasztjuk a négyzetet, akkor a program nem az adott napot fogja vizsgálni, hanem az egész hónapot, majd ezt adja vissza táblázatban kiírva és egy diagrammon is jelzi azt.(9, ábra.)



8. ábra: Napi jegyzés



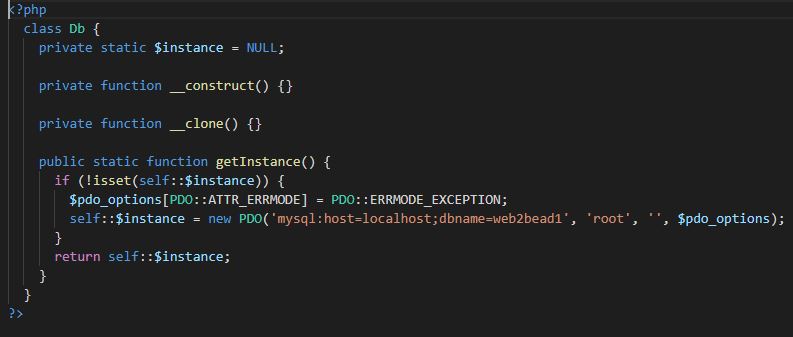
9. ábra: Havi jegyzés

* 1. Kód bemutatása
     1. Kezdőlap

A program az index.php fájlon keresztül indul el, ami megvizsgálja, hogy létezik-e már az url, vagyis valaki kiválasztott-e már valamilyen menüt, ha nem akkor automatikusan a kezdőlapot tölti be.(10. ábra.) Ezen kívül 2 require található, még benne, az első az adatbázis kapcsolat PDO-val, míg a másik a layout.php fáljt várja.(11.ábra.)



10. ábra: Index.php



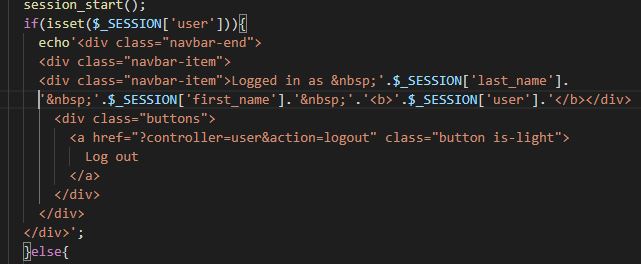
11. ábra: Connection.php

A Layout.php fájlban valósítottuk meg az oldal törzsét. Itt található a head, ahová stílusnak megadtuk a Bulma nevű könyvtár elérését és elhelyeztük a Highcharts diagram könyvtárat. Ezután az oldal menüszerkezetét valósítottuk meg, és Querry stringként adtuk meg az elérési utakat.(12. ábra.)



12. ábra: Head

Ez után indítunk egy session-t és megvizsgáljuk, hogy létrejött-e már, vagyis be van-e jelentkezve a felhasználó. Ettől függően echoval jelenítjük meg a be- és kijelentkezés gombokat.(13. ábra.)



13. ábra: Session

Majd a routes.php fájl meghívása történik. Amelyben egy call() nevű funkciónak adjuk át a Querry stringeket. Mivel ezek kulcs – érték párok, így az egyik érték pár controller néven az osztályt, míg a másik action néven a funkciót fogja jelölni. Attól függően melyik oldal lett megnyitva, azt az osztályt példányosítjuk és az adott funkcióját hívjuk meg. Bizonyos esetekben plusz modell fájlokra is szükség van, ezeket egy requirrel csatoljuk. (14. ábra.) Ezután egy tömbbel vetjük össze a kapott eredményt, ami a fájlok neveit tartalmazza, hibakezelés céljából.



14. ábra: Call()

A controllers mappában találhatóak ezek a PHP fájlok. A pagesController.php hívj meg a views mappában található nézetek közül az oldalak nagy részét. Itt található az error() oldal is és a kezdőlap is, amik egyszerű HTML fájlok. A nézeteket az általuk meghívott controllerek neve szerinti mappákban találhatóak a views mappán belül.

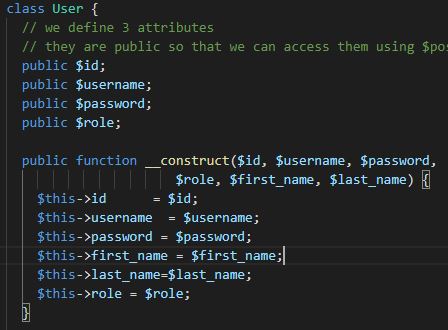
* + 1. Belépés

A belépés és regisztráció oldalakat is a pagesController.php fájl hívja meg, ami betölti a nézetet a views/pages mappából. Ezek a fájlok formokat tartalmaznak, amik action paramétere egy újabb Querry sztring, ami a userController.php fájlt fogja megjeleníteni. Az ábrán a regisztrációs nézet látható, itt a controller register funkciója fog betöltődni.(15. ábra) A bejelentkezésé is ezen az elven működik.



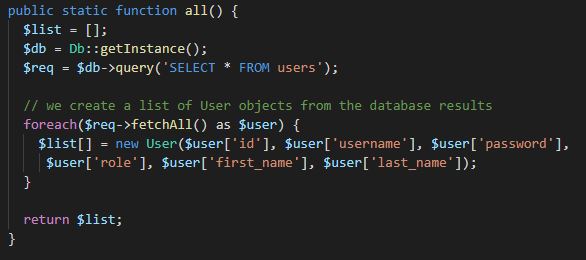
15. ábra: Regisztráció

Ehhez a controllerhez egy modelre is szükség van, a user modelben kezeljük a felhasználó adatait.(16. ábra.)

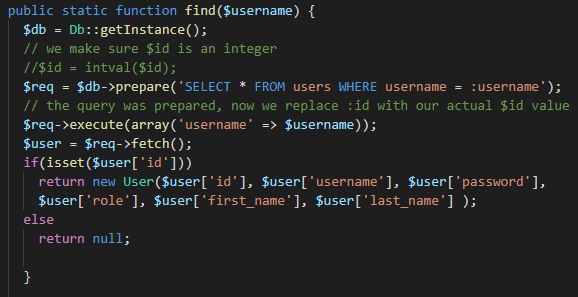


16. ábra: User

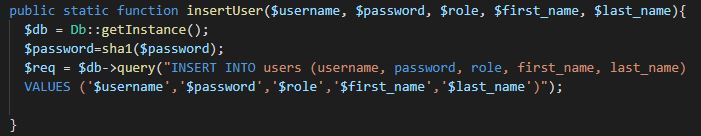
Ebben az osztályban három funkció található, az all(), ami lekérdezi az összes felhasználót.(17. ábra) A find(), ami felhasználónév alapján visszaadja az adott felhasználó adatait.(18. ábra) És az insertUser() ami egy újonnan regisztrált felhasználó adatait adja hozzá az adatbázishoz.(19. ábra)



17. ábra: User all()



18. ábra: User find()



19. ábra: User insert()

Mivel ezek a modellek az adatbázis kapcsolatokat valósítják meg, a userController osztályban használjuk fel őket. Itt egy login() nevű funkció található, ami a bejelentkezést valósítja meg, ez például a modell find() metódusát használja, és veti össze a bejelentkezési formban található adatokat az adatbázisban lévő, regisztrált felhasználók adataival. Található még ebben az osztályban egy logout() és egy register() metódus is. Az előbbi a futó sessiont állítja le, míg utóbbi a regisztrációt valósítja meg.

* + 1. Vélemények

Ha a menüben a News pontot választjuk, akkor a routes.php a newsController osztályt példányosítja és a home() funkcióját híja meg, ami betölti az összes posztot és kommentet. Található még itt egy error() metódus, ami hiba esetén jeleníti meg az error nézete. Egy createNews() funkció ami egy új posztot hoz létre és egy creatComment() ami egy új kommentet. A utóbbi két metódus modellek osztályaiból hív meg funkciókat, így ezeket is meg kell hívnunk a require metódussal még a routes.php fájlban.

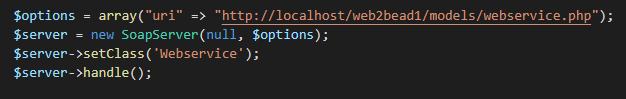
A Post osztály egy poszt tartalmát és adatait tárolja, ezeket adja át és kéri el az adatbázistól. Három metódusa van, egy all() ami lekérdezi az összes posztot majd vissza adja azokat. Egy find() aminek bemeneti értéke egy id, ez alapján keresi ki az adott posztot. És végül egy insertNews(), ami az új poszt tartalmát és adatait adja az adatbázishoz. A Comment osztály ehhez nagyon hasonló. A kommenteket tárolja, és azokkal készít lekérdezéseket. Az all() metódus visszaadja az összes kommentet. A find() funkció itt azt várja értékül, hogy melyik id-jű poszthoz lett hozzá szólva. Végül az insertComment() az adatbázishoz adja az adott komment adatait.

* + 1. SOAP szolgáltatás

A Webservice menüpontra kattintva a WebserviceController osztály lesz példányosítva. Jól látszik, hogy a webservice.php modellt nem hívja meg a routes.php fájl, mégis betölt az oldal a SOAP szolgáltatásnak köszönthetően. A kontrollernek egy show() függvénye van, ami létrehozza a SOAP kapcsolatot a szerverrel, ami jelen esetben a webservice.php modell. Ezután meghívja annak metódusait és megjeleníti, a views/webservice/show.php nézeten keresztül. Itt az oldal leírása található HTML kóddal és a modellből visszakapott értékek jellenek meg táblázatos alakban.

A webservice.php modell egy egyszerű SOAP webszervízt valósít meg. Három metódus található benne, ami az adatbázis 3 táblájának összes értékét visszaadja táblánként. A getgep() a gep táblát, a getszoftver() a szoftver táblát, a gettelepites() pedig a telepites táblát.

Majd az osztály végén állítjuk be a SOAP szolgáltatást.(20. ábra)



20. ábra: SOAP Szerver

* + 1. Magyar Nemzeti Bank SOAP

Az utolsó menüpont az Exchange Rate, itt az ExchangerateController osztályt pédányosítjuk. A Magyar Nemzeti Bank (továbbiakban MNB) SOAP szolgáltatását kellett használnunk, amit a beépített SoapClient függvény segítségével példányosítottunk.. Először a show() funkció fut le ha megnyitjuk ezt az oldalt. Itt kérést küldünk az MNB-nek, ami a lekérdezhető valutákat és az kezdő és vég dátumot adja vissza. Ezután meghívjuk a views/exchangerate/show.php nézetet, ami megjeleníti ezeket a valutákat lenyíló listában, ahonnan választhat a felhasználó. Majd ezek a valutákat és a kiválasztott dátumot ismét visszaadjuk az exchangerateControllernek. Attól függően, hogy a felhasználó egy napra vagy az egész hónapra akar keresni, az a funkció fog lefutni. Adott napi keresésnél az exchange() ami adott napra adj vissza a kért valuta értékét. Míg ha bejelölik a havi nézetet, akkor az exchangeMonth(), ami az egész havi lebontásban adja vissza, majd táblázatban és grafikonon is megjeleníti. A show() nézet végén található a Highcharts könyvtárral készített grafikon.

* 1. Fejlesztési lehetőségek

További fejlesztést több oldal is igényelne. Ilyen például a News lap, ahol nincs mód a kommentek és vélemények törlésére vagy módosítására. Ezen kívül a SOAP szerviz jelenleg csak a 3 adatbázis teljes tartalmát adja vissza. Ezt lehetne bővíteni különböző lekérdezésekkel. Az MNB klienst is további lehetőségekkel lehetne bővíteni.

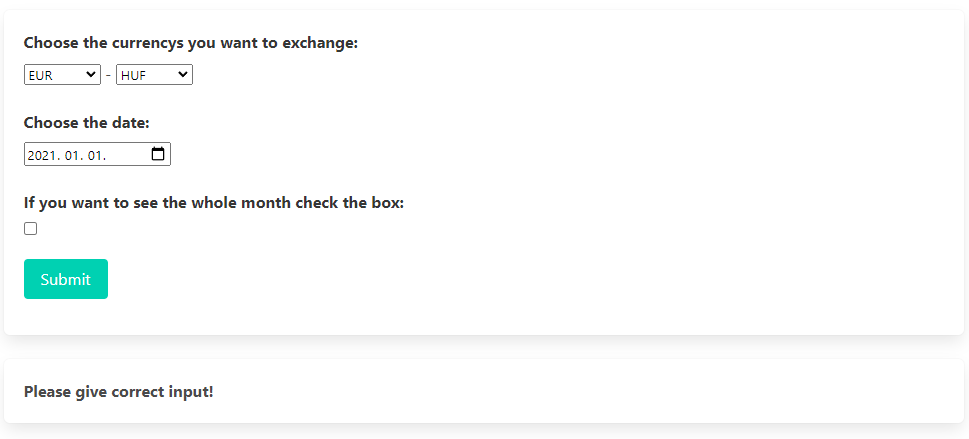
* 1. Tesztelés

A weboldal egyes részeit külön teszteltük manuálisan, próbálkozások után tártunk fel pár hibát. Ezek elkerülése érdekében a News oldalon ne tudjuk üresen elküldeni a kommenteket.(21. ábra)

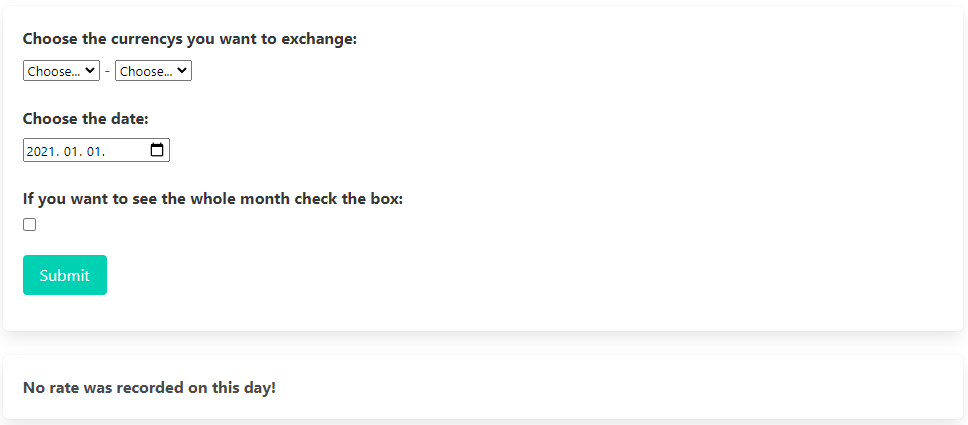


21. ábra: Üres mező

Az Exchane Rate lapon is így került be két hiba üzenet is, az egyik, ha a felhasználó hibás vagy kevés adatot ad meg.(22. ábra) A másik pedig, hogy nem minden nap történt jegyzés az MNB-nél, és ilyenkor hibával tért vissza a SOAP szerver, de ez kezelve lett, így már csak egy üzenet látszódik.(23. ábra)



22. ábra: Helytelen adatok.



23. ábra: Nincs jegyzés

1. Hivatkozások, könyvtárak.

A programban használtuk a Bulma nevű CCS frameworkot a nézet elkészítéséhez.

* Link: <https://bulma.io/>
* Csatolás: <https://cdn.jsdelivr.net/npm/bulma@0.9.3/css/bulma.min.css>

A diagram elkészítéséhez a Highcharts könyvtárat használtuk:

* Link: <https://www.highcharts.com/>
* Csatolás: <https://code.highcharts.com/highcharts.js>

A Magyar Nemzeti Bank SOAP szolgáltatás egy wsdl importálással történt:

* <http://www.mnb.hu/arfolyamok.asmx?WSDL>

A projekthez a GitHub verziókezelőt használtuk:

* Link: <https://github.com/atjager/web2bead1>

A projektet internetes tárhelyen is megvalósítottuk:

* Link: <http://web2bead1.nhely.hu/>

Ábrajegyzék

[1. ábra: Mappa szerkezet 5](#_Toc87712951)

[2. ábra: Kezdőlap 6](#_Toc87712952)

[3. ábra: Belépés 6](#_Toc87712953)

[4. ábra: Belépés 6](#_Toc87712954)

[5. ábra: Kijelentkezve a vélemények 7](#_Toc87712955)

[6. ábra: Bejelentkezett vélemények 7](#_Toc87712956)

[7. ábra: SOAP Webszervíz 8](#_Toc87712957)

[8. ábra: Napi jegyzés 9](#_Toc87712958)

[9. ábra: Havi jegyzés 9](#_Toc87712959)

[10. ábra: Index.php 10](#_Toc87712960)

[11. ábra: Connection.php 10](#_Toc87712961)

[12. ábra: Head 11](#_Toc87712962)

[13. ábra: Session 11](#_Toc87712963)

[14. ábra: Call() 12](#_Toc87712964)

[15. ábra: Regisztráció 13](#_Toc87712965)

[16. ábra: User 13](#_Toc87712966)

[17. ábra: User all() 13](#_Toc87712967)

[18. ábra: User find() 14](#_Toc87712968)

[19. ábra: User insert() 14](#_Toc87712969)

[20. ábra: SOAP Szerver 15](#_Toc87712970)

[21. ábra: Üres mező 16](#_Toc87712971)

[22. ábra: Helytelen adatok. 17](#_Toc87712972)

[23. ábra: Nincs jegyzés 17](#_Toc87712973)