Web-programozás II. beadandó

Dokumentáció

|  |  |
| --- | --- |
| **Készítették** | **Feladatkör** |
| Katona Zsolt Krisztián | Fejlesztés |
| Sándor Adrián | Fejlesztés |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Dokumentum azonosító: | WEBII-D-1 |
| Verzió: | 0.01 |
| Dátum: | 2024.11.26 |

|  |
| --- |
| **Lektorálta** |
| - |

**©** Minden jog fenntartva!

Tartalomjegyzék

[1. Bevezetés 4](#_Toc183633748)

[1.1. Dokumentum célja 4](#_Toc183633749)

[1.2. Választott projekt leírása 4](#_Toc183633750)

[1.3. Feladatok 4](#_Toc183633751)

[2. Fejlesztési környezet 5](#_Toc183633752)

[2.1. Verziókövetés: GitHub 5](#_Toc183633753)

[3. Megvalósítás 7](#_Toc183633754)

[Adatbázis 7](#_Toc183633755)

[Adatbázis táblák 7](#_Toc183633756)

[anyag tábla: 7](#_Toc183633757)

[tervezo tábla: 8](#_Toc183633758)

[erme tábla: 8](#_Toc183633759)

[akod tábla: 8](#_Toc183633760)

[tkod tábla: 9](#_Toc183633761)

[REST végpontok 10](#_Toc183633762)

[Végpontok 10](#_Toc183633763)

[/api/anyag GET végpont 11](#_Toc183633764)

[/api/tervezo GET végpont 11](#_Toc183633765)

[/api/akod GET végpont 11](#_Toc183633766)

[/api/tkod GET végpont 11](#_Toc183633767)

[/api/erme végpontok 12](#_Toc183633768)

[MNB SOAP kliens 14](#_Toc183633769)

[Design 15](#_Toc183633770)

[Felhasználók kezelése 15](#_Toc183633771)

[Menük 15](#_Toc183633772)

[Oldalak 16](#_Toc183633773)

[Főoldal 16](#_Toc183633774)

# Bevezetés

## Dokumentum célja

Jelen leírás a(z) „L-EA01 WEB-programozás II. [L-K-GINFBAL-WEBPROG2-1-EA01] [CLS 2024/25/1]” elnevezésű kurzus, Sándor Adrián és Katona Zsolt által fejlesztés alatt álló számítástechnikai projekt, weboldal fejlesztői környezetét, koncepcióját, design elemeinek tervezett megvalósítását és fejlesztési tevékenységeit ismerteti.

* 1. Választott projekt leírása

A weboldal a magyar forint érméinek részletes adatait mutatja be, különös tekintettel a címletekre, a kiadott darabszámokra, a tervezőkre és az érmék összetételére. Az oldal célja, hogy rendszerezett formában nyújtson információkat a forint érmékről, segítve ezzel a numizmatika iránt érdeklődőket, a gyűjtőket vagy akár a szakmai kutatókat. Az adatokat átlátható módon, könnyen navigálható felületen keresztül érhetik el a látogatók.

A menürendszer hierarchikus felépítése és a dinamikus funkciók biztosítják, hogy a felhasználók egyszerűen hozzáférhessenek az érmékkel kapcsolatos specifikus információkhoz, például a kibocsátási dátumokhoz vagy az érmék tervezőihez. A tervezés során kiemelt figyelmet kaptak az érmegyűjtés szempontjából releváns adatok, például az érme anyagának összetétele (arany, ezüst stb.) és az érme történeti háttere.

Ez a weboldal nemcsak az érmék műszaki és történeti adatait rendszerezi, hanem betekintést nyújt a magyar forint fejlődéstörténetébe is, hozzájárulva ezzel a pénztörténet iránti érdeklődés mélyítéséhez. Az egyetemi beadandó szempontjából remek példát nyújt arra, hogyan lehet adatbázis-alapú információkat vizuálisan és strukturáltan megjeleníteni egy modern webes környezetben.

## Feladatok

1. **feladat (Kötelező elem!)**

a, a menüpontok neveit és a menüpontokhoz tartozó oldalak azonosítóját az adatbázisban tárolja, Megoldás: [Menük](#_Menük)

b, többszintes menürendszert megvalósít Megoldás: [Menük](#_Menük)

c, Regisztráció, Bejelentkezés: legalább „látogató”, „regisztrált látogató” és „admin” szerepkör el van különítve. Megoldás: [Felhasználók kezelése](#_Felhasználók_kezelése)

d, Az első oldalon mutassa be a témát. Megoldás: [**Főoldal**](#_Főoldal)

1. Keressen és alkalmazzon egy **ingyenes reszponzív témát** az oldalaihoz. A dokumentációban írja le, hogy melyik témát választotta.

Ötleteket talál a dokumentum végén.   
Megoldás: [Téma (Stocker)](https://themewagon.com/themes/stocker/), [Design](#_Design)

1. **SOAP-szerver menü**: Készítsen egy Restful webszolgáltatást, Az adatbázis minden tábláját használja fel a feladathoz. Megoldás:
2. **SOAP-kliens menü:** Készítsen a SOAP webszolgáltatáshoz egy klienst, amivel tudja tesztelni a szolgáltatást
3. **SOAP-MNB**: Egy oldalon használja fel a Magyar Nemzeti Bank SOAP adatszolgáltatását, és valósítson meg több lekérdezési lehetőséget is az adatokból.

Lehessen keresni az alábbiakra:  
a, Egy adott devizapár (pl. Eur-Huf, Eur-Usd, …) adott napján mennyi volt az árfolyam?  
b, Egy adott devizapár egy adott hónapjában minden napra mennyi volt az árfolyam táblázatban kiíratva. Valamint az adatok megjelenítése grafikonon is. Grafikonhoz javasolt a <https://www.chartjs.org/> használata. Minta: <https://www.mnb.hu/arfolyam-lekerdezes>

1. **Restful-szerver menü:** Készítsen egy Restful webszolgáltatást az adatbázisának egyik táblájához. Valósítsa meg a GET, POST, PUT, DELETE funkciókat.

Tesztelje a webszolgáltatást cURL-el és Postman-el. A dokumentációban mutassa be a tesztelés lépéseit.

Megoldás: [REST végpontok](#_REST_végpontok)

1. **Restful-kliens menü:** Készítsen a Restful webszolgáltatáshoz egy klienst, amivel tudja tesztelni a szolgáltatást (GET, POST, PUT, DELETE funkciók).
2. **PDF menü**: Készítsen egy PDF készítő szolgáltatást a **TCPDF** segítségével. A felhasználó 3 szöveges beviteli mezőben, vagy lenyíló listában megad adatokat és a rendszer az adatbázisból olvasva készít egy letölthető PDF fájlt. Az adatbázis három tábláját használja fel.
3. Alkalmazását töltse fel és valósítsa meg Internetes tárhelyen is.

(**Kötelező elem**! **A működés ez alapján lesz javítva**) Bármelyik tárhely-szolgáltatót használhatja. **URL címben szerepeljen a csoport egyik tagjának a neve.**

1. Használják a GitHub (github.com) verziókövető rendszert.   
   (**Kötelező elem**! **A forrás ez alapján lesz javítva**)

Ne csak a kész alkalmazást töltsék fel egy lépésben, hanem a részállapotokat is még legalább 5 lépésben személyenként.  
**A GIT-en saját nevet válasszanak, ami alapján be lehet azonosítani, hogy ki mit töltött fel.**

1. A **GitHub-on a projektmunka módszert alkalmazzák**
2. Készítsen egy legalább 15 oldalas dokumentációt képernyőképekkel(**Kötelező elem**!), amiben bemutatja alkalmazását és leírja, hogy az előző pontok feladatait hogyan valósította meg. **Fontos, hogy ez utóbbit leírja a dokumentációban, mert a feladat ez alapján lesz javítva.** A dokumentációban adja meg a tárhely és a GitHub projektjének URL címét is és az Internetes tárhelyen az **admin és egy másik felhasználó belépési adatait.**

# Fejlesztési környezet

* 1. Verziókövetés: GitHub  
     A GitHub, mint vezető szoftverfejlesztési platform, lehetővé teszi a csapat számára, hogy hatékonyan kezelje a kódbázist, nyomon követhesse a változásokat, és elősegítse a kollaboratív munkát azáltal, hogy egy központosított helyen tárolja a projekt fájljait és dokumentációját. Az integrált verziókezelési rendszer előnyei közé tartozik a fejlesztési folyamatok nagyfokú automatizálása, a visszaállíthatóság, valamint a különböző fejlesztési ágak (branches) kezelése, ami lehetővé teszi a csapat számára, hogy párhuzamosan dolgozzon különböző funkciókon. A GitHub használatával a fejlesztőcsapat kihasználhatja a kódmegosztás és a közösségi hozzájárulás előnyeit is, így gyorsítva a fejlesztési ciklusokat és javítva a projekt átláthatóságát és nyomon követhetőségét.
  2. Felhasznált technológiák
     1. **Bootstrap**: Egy ingyenes és nyílt forrású front-end keretrendszer, amely elősegíti a gyors és egyszerű reszponzív webdesign kialakítását. Készen álló HTML és CSS sablonokat, valamint JavaScript komponenseket kínál, amelyekkel könnyedén lehet készíteni felhasználóbarát és esztétikailag vonzó weboldalakat.
     2. **Reszponzív webdesign**: Az a tervezési és fejlesztési módszertan, amelynek célja, hogy a weboldalak minden eszközön - asztali számítógéptől kezdve a mobiltelefonokig - jól nézzenek ki és jól működjenek. A Bootstrap keretrendszer kiválóan támogatja a reszponzív webdesign kialakítását.
     3. **Laravel**:

A Laravel egy modern PHP keretrendszer, amely az egyszerűségre és az eleganciára épít. MVC (Model-View-Controller) architektúrát követ, ami megkönnyíti az alkalmazások fejlesztését és karbantartását. Számos beépített funkcióval rendelkezik, például routing, ORM (Eloquent), beépített felhasználói autentikáció, csomagkezelés (Composer), valamint Blade sablonmotor. A Laravel népszerűsége főként annak köszönhető, hogy gyorsan lehet vele skálázható és biztonságos webes alkalmazásokat fejleszteni.

* + 1. Laravel Auth

A Laravel beépített autentikációs rendszere egyszerűsíti a felhasználók hitelesítését és azonosítását. Az artisan parancssor segítségével gyorsan generálható login, regisztráció, jelszóemlékeztető és e-mail-verifikáció funkció. A Laravel Auth támogatja a session-alapú hitelesítést, az API tokeneket (pl. Passport vagy Sanctum segítségével), és testreszabható middleware-k segítségével biztosítja a jogosultságkezelést. A Laravel 10-től kezdve a Breeze és a Fortify csomagok is elérhetők, amelyek egyszerű és testreszabható megoldásokat kínálnak.

* + 1. Livewire

A Livewire egy Laravel-specifikus front-end keretrendszer, amely lehetővé teszi, hogy dinamikus felhasználói felületeket hozz létre JavaScript írása nélkül. A Livewire mögötti alapötlet az, hogy a komponenseket PHP-ben írod meg, és a háttérben AJAX hívásokkal frissíti az oldal egyes részeit. Ez különösen hasznos, ha interaktív elemeket, például űrlapokat vagy valós idejű adatmódosításokat szeretnél létrehozni. A Livewire egyszerűen integrálható a Blade sablonmotorral, és remekül működik együtt más Laravel-funkciókkal.

* 1. Szoftverek és futtatókörnyezet
     1. XAMPP

A XAMPP egy ingyenes, multiplatform szervercsomag, amely az Apache webszervert, a MySQL vagy MariaDB adatbázist, valamint a PHP és Perl nyelvek futtatókörnyezetét tartalmazza. A XAMPP célja, hogy egyszerűsítse a fejlesztési környezet beállítását, így gyorsan tesztelheted és fejlesztheted webalkalmazásaidat. Könnyen telepíthető, és tartalmaz egy vezérlőpanelt, ahol a szervereket (pl. Apache, MySQL) egyszerűen elindíthatod vagy leállíthatod.

* + 1. phpMyAdmin

A phpMyAdmin egy nyílt forráskódú webes felület, amely lehetővé teszi a MySQL és MariaDB adatbázisok kezelését. Ezzel az eszközzel könnyen létrehozhatsz, módosíthatsz vagy törölhetsz adatbázisokat, táblákat, mezőket, valamint futathatsz SQL lekérdezéseket. A phpMyAdmin egyszerűsíti az adatbázis-kezelést a grafikus felhasználói felülete révén, így nem szükséges mély SQL-ismeret az alapműveletek elvégzéséhez.

* + 1. IntelliJ PHPStorm

A PHPStorm egy professzionális fejlesztőkörnyezet (IDE), amely a JetBrains által lett kifejlesztve, és kifejezetten PHP fejlesztésre optimalizálták. Számos fejlett funkcióval rendelkezik, például kódkitöltés, hibakeresés (debugging), verziókövető rendszerek (pl. Git) integráció, valamint beépített támogatás a modern keretrendszerekhez (pl. Laravel, Symfony). A PHPStorm további előnye, hogy kiválóan együttműködik a front-end technológiákkal (HTML, CSS, JavaScript), valamint a PHPUnit teszteléssel.

# Megvalósítás

# Adatbázis

Az Adatbázis a weboldalunk alapvető és kritikus eleme, amely két fő célra használatos. Elsődlegesen szolgál a felhasználók hitelesítésére, ugyanis itt tároljuk azokat az adatokat, amelyek az azonosításhoz és az egyedi felhasználói profilok létrehozásához szükségesek, mint a felhasználónevek, jelszavak, jogosultságok és id. Másrészt az adatbázis tartalmazza a feladat eredetileg meghatározott tábláit és az importálást követően a letöltött adatokat.

A felhasználóknak lehetőségük van üzenetet küldeni az üzemeltetőnek. Ezek az üzenetek szintén az adatbázisban vannak tárolva. Egy-egy rekordon, a felhasználó azonosítójával megállapítható ki küldte az üzenetet.

Miután bejelentkezés nélkül is van lehetőség üzenetküldésre, ilyenkor pedig nincs azonosítható felhasználó, null értékkel kerülnek rögzítésre az üzenetek, megjelenítéskor ezek a rekordok a „Vendég” megnevezést kapják.

Adatbázis táblák

anyag tábla:

A tábla a korlátozott darabszámban kiadott magyar forint érmékhez használt nyersanyagokat tartalmazza. Minden rekord egy konkrét fémet reprezentál. A femid mező, a tábla elsődleges kulcsa, mely azonosítja a fémeket.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

tervezo tábla:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásA tervezo tábla a forint érmék tervezőinek adatait tárolja. Minden sor egy konkrét tervezőt reprezentál, beleértve a tervező egyedi azonosítóját és nevét. Ez a tábla a tervezők azonosítására szolgál, amely lehetővé teszi az egyes érmék készítőinek nyomon követését.

erme tábla:

Az erme tábla a limitált darabszámban kiadott magyar forint érmék alapvető adatait tárolja. Minden sor egy érmeadatot reprezentál, annak címletével, tömegével, kiadott darabszámával, valamint a kibocsátás és bevonás dátumával. Ez a tábla az érmék részleteinek, információinak tárolására szolgál.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

akod tábla:

Az akod tábla egy kapcsolótábla. Az alapanyagoknak használt fémek, azaz anyag tábla és az érmék adatait tartalmazó erme adatbázistábla adainak kapcsolatait tartalmazza. A kapcsolótáblára azért volt szükség, mert az anyagok és érmék kapcsolata több a többhöz, egy érméhez több alapanyagot is használhattak, valamint egy alapanyagot több érme anyagaként is felhasználtak.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

tkod tábla:

A tkod tábla szintén egy kapcsolótábla, ami a tervezők és érmék kapcsolatait reprezentálja. Az érmék és tervezőik közt szintén több a többhöz a kapcsolat, több tervező is közreműködhetett egy érme megalkotásában, valamint egy-egy tervező több érménél is szerepet játszott.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Az adatbázis elkészülte után a Laravelben a megfelelő modelleket valósítottuk meg, amelyek közvetlen kapcsolatot biztosítanak az adatbázis és az alkalmazás logikája között. Ezek segítségével könnyen kezelhetők az adatbázis műveletek, mint például az adatok lekérdezése, beszúrása, frissítése vagy törlése. A modellek reprezentálják az egyes táblákat, könnyítik az adatokkal való munkát, támogatják a kapcsolatok kezelését és ezek az üzleti logika központjai.

Az adatbázis tartalmaz a fent felsorol táblákon kívül több adattáblát is, mint például a sessions, cache táblák. Ezek nem szerepeltek az importálandó állományban, viszont a Laravel automatikusan létrehozta ezeket a megfelelő működés miatt. Ezek a táblák az alkalmazás működéséhez szükségesek és bizonyos komponensek alapértelmezett konfigurációjával együtt kerülnek létrehozásra.

# REST végpontok

A REST hívások HTTP protokollon keresztül történnek. Általában négy fő műveletet használunk, GET az adatok lekérésére, POST új adatok rögzítéséhez, PUT az adatok frissítésére, módosítására szolgál, végül a DELETE végpontok adatok törlésére használtak.

A kérések törzs adata (amennyiben van), többnyire xml vagy json formában kerül küldésre, valamint jellemzően ugyan így kapja vissza a választ a kliens, kiegészülve még egy lehetőséggel, a html-el.

A REST végpontokhoz Basic authentikáció szükséges. Az oldalon regisztrációkor megadott e-mail cím és jelszó használható a végpontok elérésére.

A routes/api.php fájlban definiáltuk a Laravel alkalmazás API végpontjait, amelyek RESTful kéréseket fogadnak. Ez a fájl külön van választva a webes útvonalaktól, hogy elkülönítse az API-funkcionalitást. Itt határozzuk meg, hogy milyen HTTP metódusokat használhatnak a kliensek, és melyik controller kezelje ezeket a kéréseket.

Az API útvonalak automatikusan a /api prefixel vannak ellátva, például egy érme adatainak lekérése a GET /api/erme végponton történhet. Ez a fájl tehát az API-központú forgalom irányításának egyik legfőbb része.

Fejlesztés során Postmannel teszteltük a végpontokat, ezen kérésekről, egy importálható Postman collection elérhető a GIT repositoryban a documents mappában, a Forint\_postman\_collection néven, minden a következőkben felsorolt végponthoz.

Az elkészült végpontokat cUrl-el is teszteltük, itt például a 25-ös azonosítójú érme törlése látható.

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

Végpontok

Az Adatbázis mind az 5, a feladat által meghatározott adattáblája lekérhető GET metódussal, valamint az érme táblához kapcsolódóan megvalósításra kerültek POST, PUT és DELETE végpontok is.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás/api/anyag GET végpont

Az anyag tábla az érmék alapanyagaival elérhető, http GET kérést indítva a következő címen: <http://katonazsoltweb2.testhosting.hu/api/anyag> . Az adathalmaz szűkítésére femid query paraméter használható. A képen a válasz json formátuma látható.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás/api/tervezo GET végpont

A tervezők adatainak eléréséhez http GET kérést szükséges indítani a következő címre: <http://katonazsoltweb2.testhosting.hu/api/tervezo> Az adathalmaz szűkítésére a tid query paraméter használható. A képen a válasz json formátuma látható.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás/api/akod GET végpont

Az anyag-érme táblák kapcsolótáblája a <http://katonazsoltweb2.testhosting.hu/api/akod> címre indított http GET kéréssel érhető el. Az adathalmaz szűkítésére query paraméterként a femid megadható. A képen a válasz json formátuma látható.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás/api/tkod GET végpont

Az érme-tervező kapcsolótábla adatai szintén elérhetők, ehhez http GET kérést kell indítani a <http://katonazsoltweb2.testhosting.hu/api/tkod> címre. A tervezoid query paraméterrel szűrhető az adathalmaz. A képen a válasz json formátuma látható.

/api/erme végpontok

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásA program lehetőséget nyújt új érme felvételére, meglévő módosítására és akár egy érme törlésére is az adatbázisból.

Érme adatok lekérése az előző végpontokhoz hasonlóan történik. GET http kérést kell indítani a <http://katonazsoltweb2.testhosting.hu/api/erme> címre, melyet az ermeid query paraméterrel szűrhetünk. A képen a válasz json formátuma látható.

Új adat rögzítésekor a használt Url nem különbözik a GET kérésnél használttól, csupán annak metódusát szükséges módosítani.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Egy ilyen kérésnél a küldött adatok vizsgálata elhagyhatatlan, így az ErmeControllerhez kapcsolódóan létrehoztunk egy ErmeRequest osztályt, mely az adatok validációs feladatait látja el. Ez ellenőrzi a bejövő kérést, megvizsgálja a mezők értékét, típusát, kötelezőségét és ha minden rendben talál, visszaadja a vezérlést az ErmeControllernek és folytatódhat az adatrögzítés, módosítás. Ellenkező esetben, a probléma okával tér vissza.

Már meglévő rekord módosításakor az url-ben szerepelnie kell az érme azonosítójának a kérést törzsében pedig a többi adatnak a POST kéréshez hasonlóan. Módosításkor az url-ben megadott id-val rendelkező érme adatait módosítjuk.

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírásTörlés szintén lehetséges, erről korábban cUrl-t is csatoltunk. Egy érme törléséhez http DELETE kérést kell indítani a /api/erme végpontra. Amennyiben a megadott azonosító nem létezik, a kérést elutasítjuk 404-es státuszkóddal.

# MNB SOAP kliens

A Laravel nem tartalmaz alapértelmezetten SOAP implementációt, miután ez a REST-nél egy régebbi, mára kevésbé népszerű protokoll. A feladat elvégzésében a Laminas keretrendszert használtuk, ami a számos összetevője közt a SOAP szerverek kezelésében is segítséget nyújt.

Az MNBService osztály egy szolgáltatásosztály, mely az MNBController-t szolgálja ki és SOAP kliens segítségével kommunikál a Magyar Nemzeti Bank (MNB) hivatalos árfolyamadatokat nyújtó webszolgáltatásával. Az osztály felelős az MNB SOAP végpontjainak eléréséért, az árfolyamadatok lekérdezéséért és azok feldolgozásáért.

A getWsdl() metódus a WSDL fájl tartalmát tölti le, mely alapján létrehozható volt a többi metódus olyan célokra mint az elérhető devizanemek letöltése, a napi árfolyamok letöltése, havi árfolyamok letöltése.

Az MNBService egy robusztus SOAP kliens, amely egyszerűsíti az MNB webszolgáltatásaival való integrációt.

Az MNBController osztály a Magyar Nemzeti Bank SOAP szolgáltatásához készült kliensfukciókat biztosítja az alkalmazás nézetei számára.

A devizanemeket több lenyíló lista is megjeleníti, melyeket adatbázisban nem tárolunk külön, de az oldal megfelelő sebessége érdekében ezeket gyorsítótárazzuk. A devizanemeket a rendszer a Laravel Cache rendszerében tárolja.

Az MNB szolgáltatásokhoz kapcsolódó útvonalak, a web.php-ban kerültek meghatározásra. Az összes végpont egy /mnb prefixel van ellátva, így külön csoportosítva az alkalmazás egyéb végpontjaitól.

A nézeteket, vagyis a weboldalon történő megjelenítést a resources/views/MNB és a resources/view/livewire mappában található php fájlok valósítják meg, utóbbi a dinamikus tartalommegjelenítésben nyújt segítséget.

# Design

A weboldal a feladatnak megfelelően egy ingyenesen elérhető CSS+HTML témát használ, némi bootstrap kiegészítéssel.

Téma alapja: <https://themewagon.com/themes/stocker/>

A téma tökéletesen illik az átalunk választott témához, mely a Magyar Forint pénzérméire fókuszál.

# Felhasználók kezelése

A felhasználók kezelésében a Laravel Auth van segítségünkre. A Laravel Auth legenerálta a kellő kontrollereket, middlewareket és a User modellt.

A User modell került bővítésre egy „role” mezővel, amely a felhasználó jogosultságaiért felel.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásA képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Jogosultságok a feladatot követően a következők:  
„látogató”, „regisztrált látogató” és „admin”

# Menük

A feladatnak megfelelően a menük az adatbázisból töltődnek be és többszintes menürendszert valósítanak meg.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

A menük az Admin menüpont „Menük” almenüpontja alatt szerkeszthetőek.

Az adminmenü kontroller egy Livewire kontroller így az oldal frissítése nélkül mennek végbe a feladatok.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, szoftver, Weblap látható

Automatikusan generált leírásMinden menühöz megadható:  
- Menü megjelenített neve

- Linkje (View)

- Menü címe

- Láthatósága (Szerepkör: none,user,admin)

- Szülő (ha van)

-Sorrend

- Státusz (Aktív?)

Megjelenítés:

A menüket az AppServiceProvider osztja meg a nézetekkel. A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A menü layout a \resources\views\layouts\menu.blade.php -ben kapott helyet.

# Oldalak

## Főoldal

A **Főoldal** a magyar forint érméit és bankjegyeit mutatja be. Az oldal tetején egy szembetűnő üdvözlő szöveg hívja fel a figyelmet a forint fontosságára és történetére. Az oldal középső részén egy táblázat jelenik meg, amely részletes adatokat tartalmaz az érmékről, például címlet, súly, kiadás és bevonás dátuma, összetételük, valamint a tervezőik nevei.

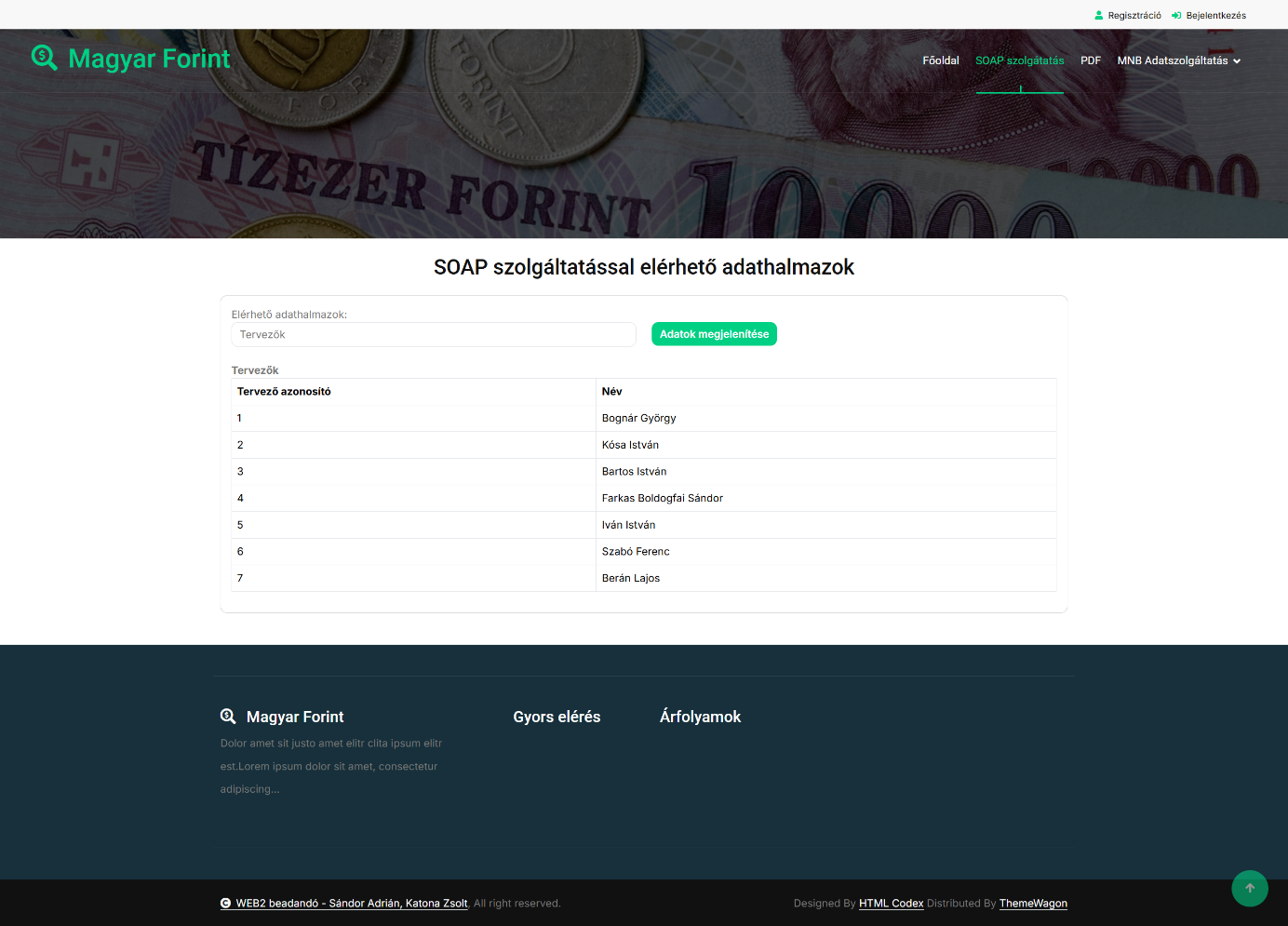
A táblázat alatt vizuális elemekkel illusztrált bankjegyek és érmék láthatók, mindegyikhez rövid leírás tartozik, amely ismerteti az adott címlet érdekességeit és hátterét. Az oldal kiemeli a forgalomban lévő bankjegyeket és érméket, külön figyelmet fordítva azok dizájnjára és történeti értékére. A megjelenés modern és reszponzív, felhasználóbarát navigációval, így könnyen kezelhető bármilyen eszközön. Az oldal célja az ismeretterjesztés és az érmék, valamint bankjegyek történetének bemutatása egy esztétikus és informatív platformon.

A táblázat lapozható és a fent helyet kapott bevitelimezővel szűrhető is. (Keresés)

Az oldal alján helyet kapott egy a csalásokra figyelmeztető szöveg.

## SOAP Szolgáltatás

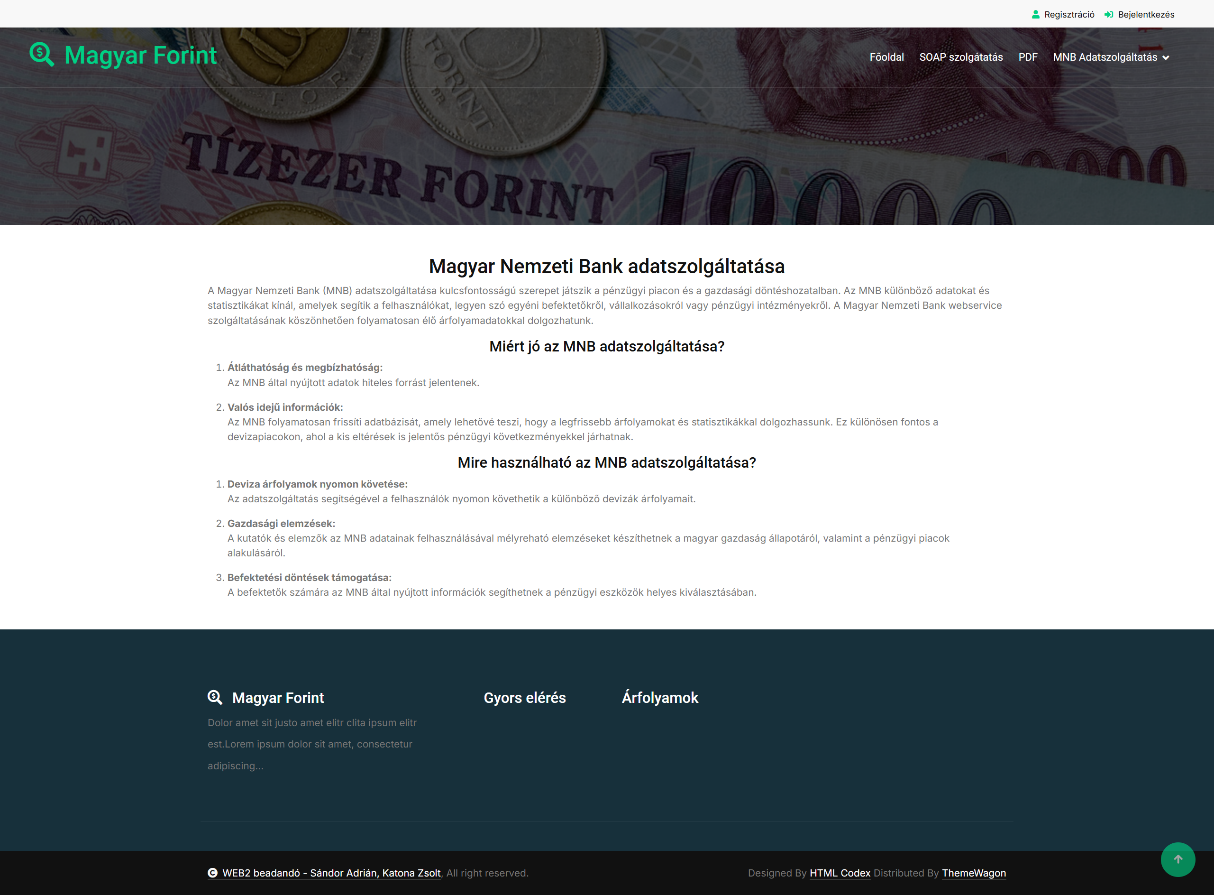
Az oldalon a weboldal saját SOAP szolgáltatása tesztelhető egy modern, de letisztult környezetben. A lenyíló menüben kiválaszthatóak az adatbázis táblái majd az adatok lekérdezése gomb megnyomására táblázatba rendezve jelennek meg a kapott adatok.



## MNB Adaszolgáltatás

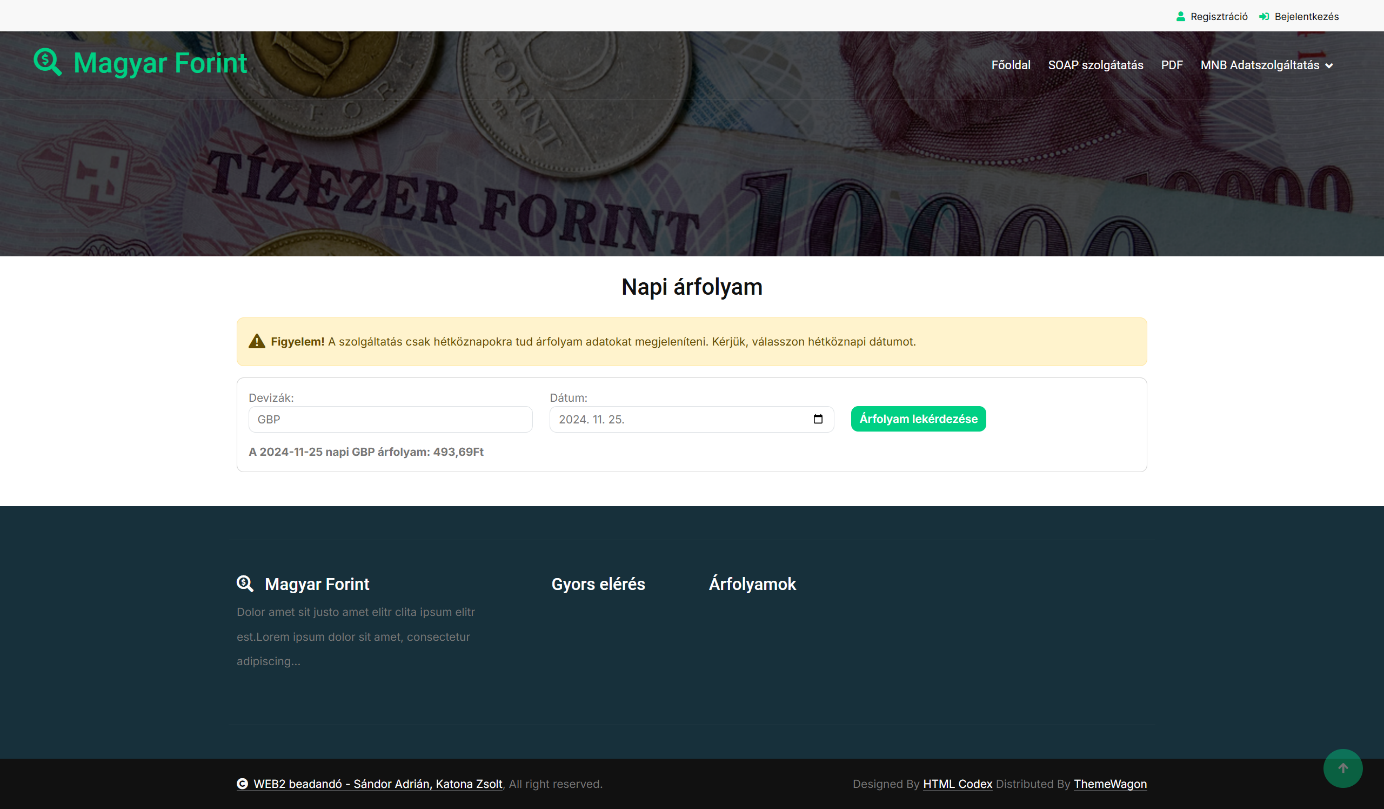
### MNB Adatszolgáltatás

Rövid ismertető az MNB Soap adatszolgáltatásról.



### Napi árfolyam

Egy valuta adott napi árfolyamának lekérdezése.



### Devizák havi árfolyamai

