Neumann János Egyetem

GAMF Műszaki és Informatikai Kar

Webprogramozás II. tantárgy

Első beadandó feladat – 1-03-Notebook-3 táblás - dokumentációja

Készítette:

Balog Olivér

2023.10.23.

**Tartalomjegyzék**

[**Feladatkiírás** 3](#_Toc148997465)

[**A webalkalmazás használata** 3](#_Toc148997466)

[**A program felépítése** 4](#_Toc148997467)

[**Az app mappa tartalma** 4](#_Toc148997468)

[**A vezérlők** 4](#_Toc148997469)

[**A Controller osztály** 4](#_Toc148997470)

[**Az AuthenticateController osztály** 5](#_Toc148997471)

[**A RegisterController osztály** 6](#_Toc148997472)

[**A HomeController osztály** 6](#_Toc148997473)

[**Az MNBController osztály** 6](#_Toc148997474)

[**A NotebookController osztály** 7](#_Toc148997475)

[**A News Controller osztály** 8](#_Toc148997476)

[**A modellek** 10](#_Toc148997477)

[**A Model osztály** 10](#_Toc148997478)

[**A Notebook, az Opsystem, a Processor és a User, Rating és News osztályok** 10](#_Toc148997479)

[**A resources mappa tartalma** 10](#_Toc148997480)

[**Layouts mappa** 10](#_Toc148997481)

[**News, notebooks mappák** 10](#_Toc148997482)

[**Egyébb fájlok a mappán belül** 11](#_Toc148997483)

[**Továbbfejlesztési lehetőségek** 11](#_Toc148997484)

[**Képernyő képek** 12](#_Toc148997485)

# **Feladatkiírás**

Készítsen egy weboldalt egy fiktív cég részére, az Adatbázisok mappában található valamely adatbázis felhasználásával, két fős csoportban.

# **A webalkalmazás használata**

Az alkalmazás előkövetelménye a composer, ami az autoload php modul használata miatt szükséges (<https://getcomposer.org/>).   
Ezután a szükséges lépések:

1. A projekt git klónozása (<https://github.com/oliverbalog/NotebookWeb2>)
2. A projekt gyökérkönyvtárában futtatni kell terminálból a „composer install --ignore-platform-reqs” parancsot
3. Szükség lesz XAMPP vagy hasonló alkalmazásra, illetve MySQL adatbázisra és Apache futtatására
4. Meg kell nyitni a config mappában található config.php fájlt és megváltoztatni az adatbázis kapcsolathoz szükséges változók értékeit.
5. A phpmyadmin-ban importálni kell a notebooks.sql fájlt
6. A hosztoláshoz meg kell nyitni a xampp mappájában található apache/conf/extra mappát és itt megnyitni a httpd-vhosts.conf file-t szövegszerkesztővel.
7. Ezt hozzá kell adni:   
   <VirtualHost \*:80>

ServerName notebookweb2.web

DocumentRoot "/Applications/XAMPP/xamppfiles/htdocs/NotebookWeb2/public/"

<Directory "/Applications/XAMPP/xamppfiles/htdocs/NotebookWeb2/public">

DirectoryIndex index.php

AllowOverride All

Require all granted

Options Indexes FollowSymLinks Includes execCGI

</Directory>

</VirtualHost>

1. Meg kell keresni a hosts file-t (Windows: C:/Windows/System32/drivers/etc)
2. Hozzáadni: 127.0.0.1 notebookweb2.web
3. Ezután a mysql és apache indítása után meg lehet nyitni böngészőből a <http://notebookweb2.web> címen az alkalmazást.

Internetes tárhelyen elhelyezett verzióhoz az alábbi linket kövesse:  
<https://balogolivernbook.nhely.hu>  
tárhely belépéshez email cím: [asperant8@gmail.com](mailto:asperant8@gmail.com)  
jelszó: XSSzTj@sUB5BjHE

Admin felhasználó az alkalmazáson belül:  
 [admin@admin.com](mailto:admin@admin.com)  
 admin

A program elérhető GitHub-on is a következő linkről:   
<https://github.com/oliverbalog/NotebookWeb2>

# **A program felépítése**

A program felépítését megtekintve két fő mappával rendelkezik (app és resources), valamint hat kiegészítő mappából, melyek a különféle beállításokért felelnek, ugymint a core fájlok meghívása, vagy az adatbázishoz való kapcsolódási adatok, vagy a megjelenítéshez szükséges .js fájlok.

## **Az app mappa tartalma**

Ebben a mappában találhatóak a vezérlők (Controllers mappa), a kisegítő fájlok (Helpers mappa), az interfészeket leíró fájlok (Interfaces) és a modellek (Models).

Az interfészek a modell osztályok felépítéséhez szükségesek.

A kisegítő fájlok olyan kiegészítő funkciókat látnak el, mint a menü linkjeinek összeállítása, vagy a session beállítása.

## **A vezérlők**

### **A Controller osztály**

Ebből az osztályból származik az összes vezérlő osztály. Ez egy absztrakt osztály, tehát nem lehet példányosítani és egy leírás az ebből származtatott osztályoknak a tulajdonságokre és függvényekre.

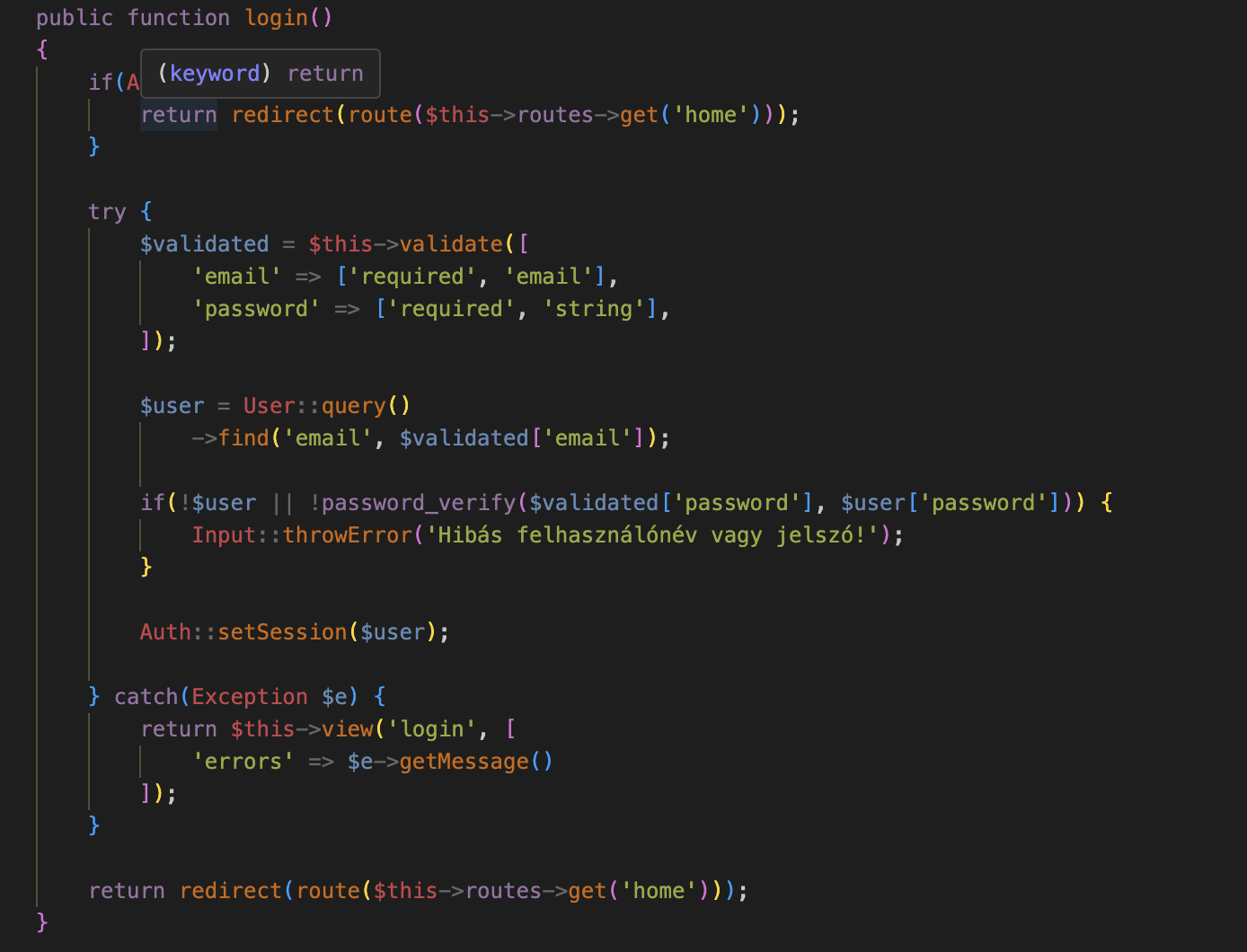
Az alábbi függvények definiálása itt történik:

* a konstruktor
* validate(array $rules): az input-ként beérkező adatok ellenőrzését szolgálja
* view(string $view, array $variables = []): a nézeti oldal meghívásáért felelős, az oldalnak átadandó adattömbbel összefűzve

### **Az AuthenticateController osztály**

A felhasználó be- és kiléptetéséért felelős osztály.

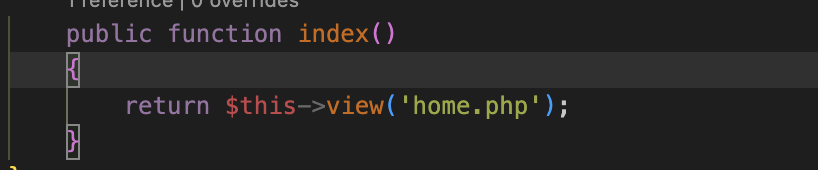
Az alábbi metódusok kerültek megvalósításra:

* login(): a belépést biztosítja, beállítja a felhasználóhoz tartozó session-t, illetve ellenörzi a beírt adatokat.   
  
* logout(): a kilépésért, a session lezárásáért felel. Kiüríti a session-t és visszairányítja a főoldalra.  
  

### **A RegisterController osztály**

A felhasználó regisztrálását biztosítja a register() metódussal.  

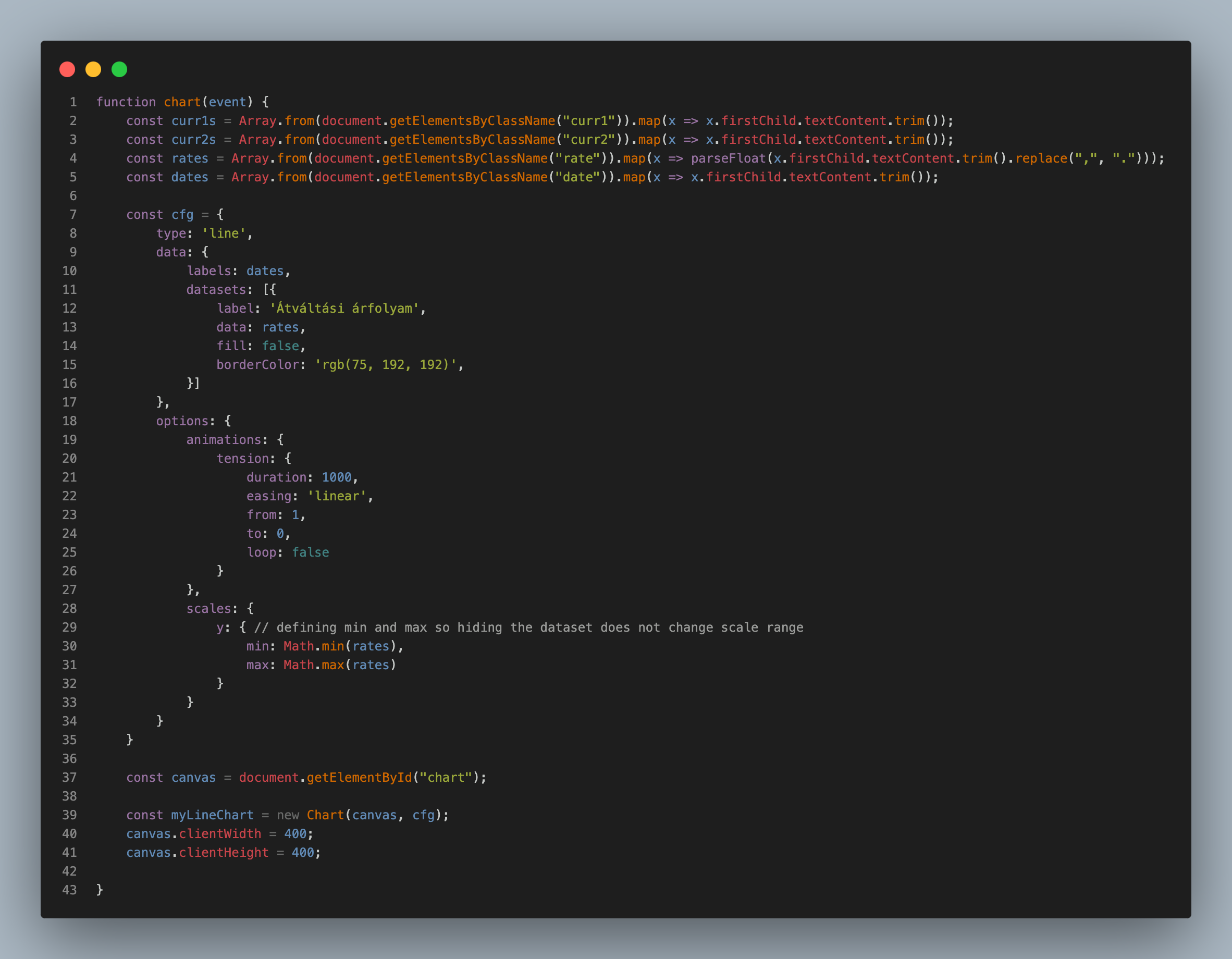
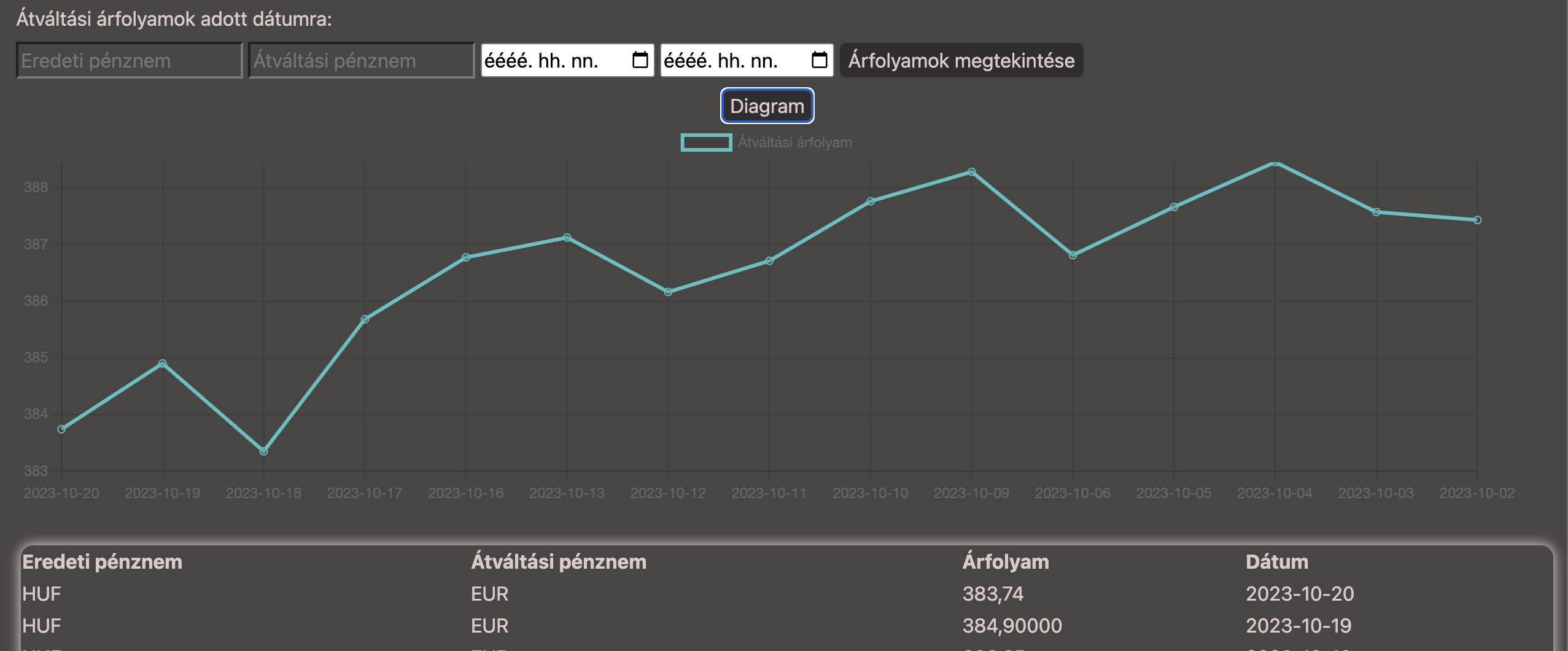

### **A HomeController osztály**

A kezdő oldal betöltése a feladata.  


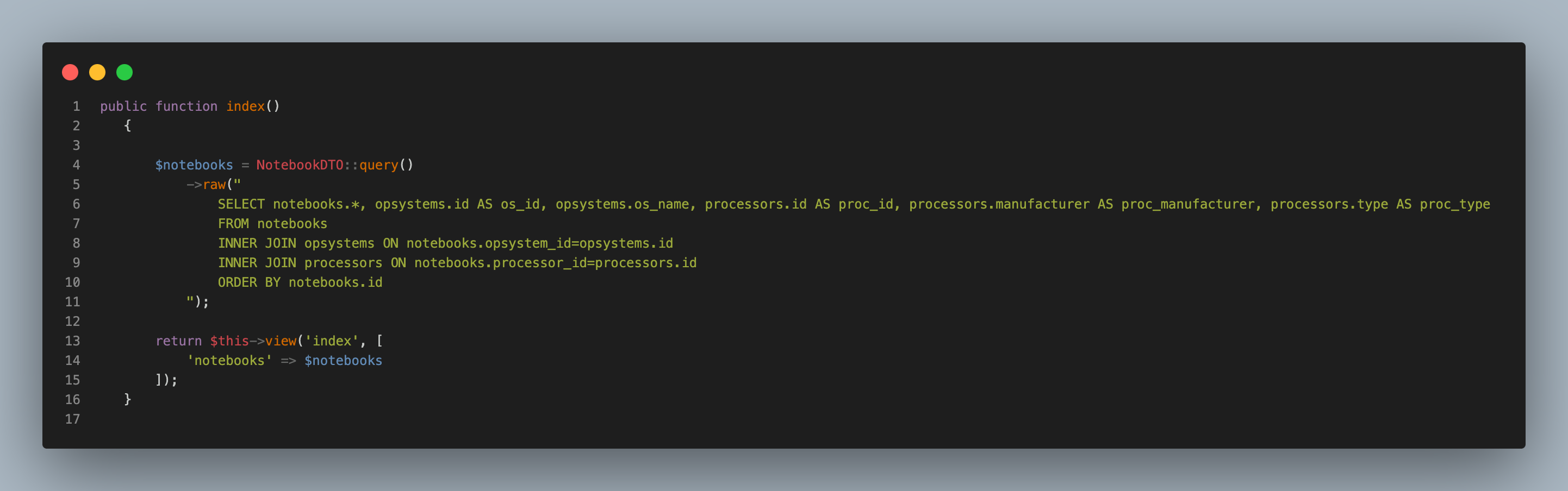
### **Az MNBController osztály**

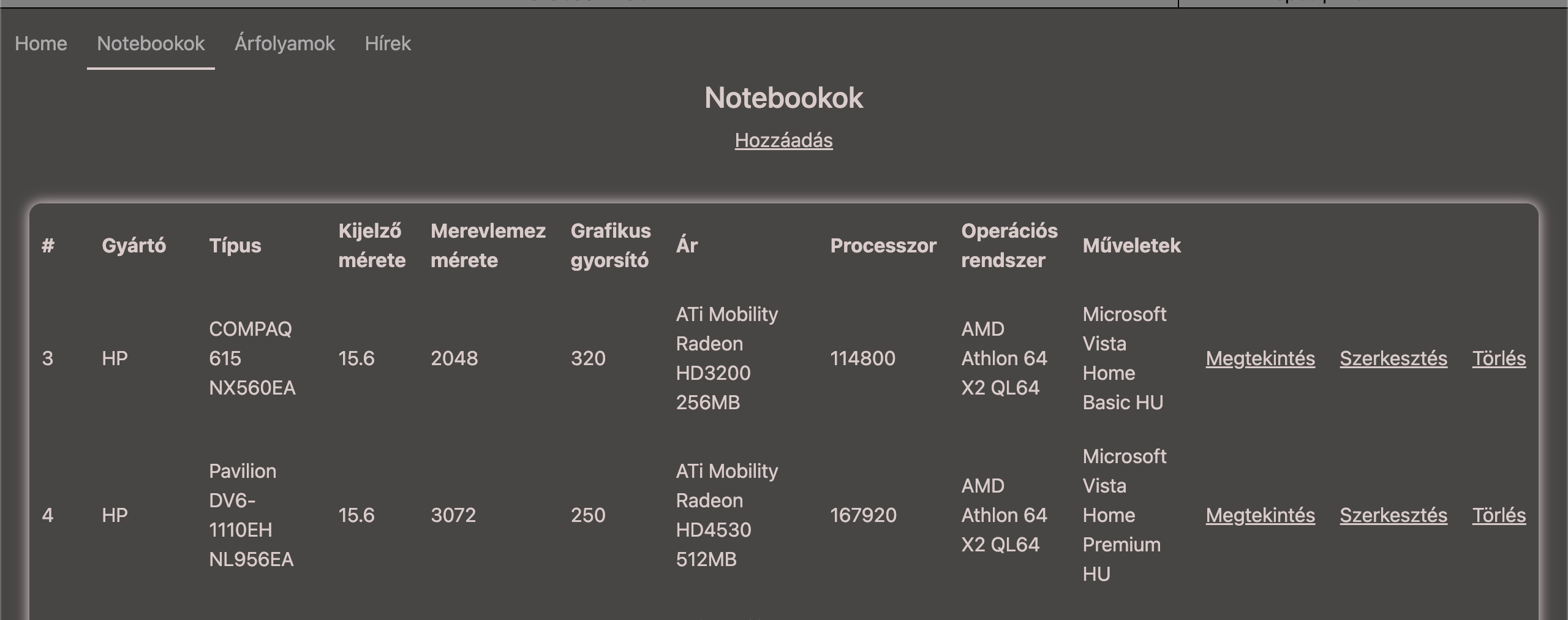
A Magyar Nemzeti Bank SOAP szerverétől lekérdezhető adatokat gyűjti fel.

A GetCurrencyRates() metódus a felhasználó által megadott két deviza közti legutolsó átváltási árfolyamot adja át a nézeti felületnek. Ellenőrzi egyrészt, hogy a felhasználó megfelelő formátumban, létező devizát ad-e meg, valamint, hogy az adott napon arról a devizáról van-e adat.

Ha minden rendben, összegyűjti az adatokat, elvégzi a szükséges átváltásokat és átadja a nézeti felületnek.  
  
A GetExchangeRates() metódus ezzel szemben egy megadott időtartományra és két megadott devizanem között átváltási árfolyamokat hozza el az MNB Soap szolgáltatástól és írja ezeket ki táblázatosan. Ilyenkor megjelenik a felületen egy „Diagram” gomb, amire rákattintva egy diagramon is meglehet nézni, hogy két devizanem között milyen árfolyamon lehetett átváltani.  
Az mnb.js-ben:  
   
A felületen így jelenik meg:  
 

### **A NotebookController osztály**

A notebooks adatbázis notebooks táblájából gyűjti össze az eszközök adatait és adja át a nézeti felületnek.  


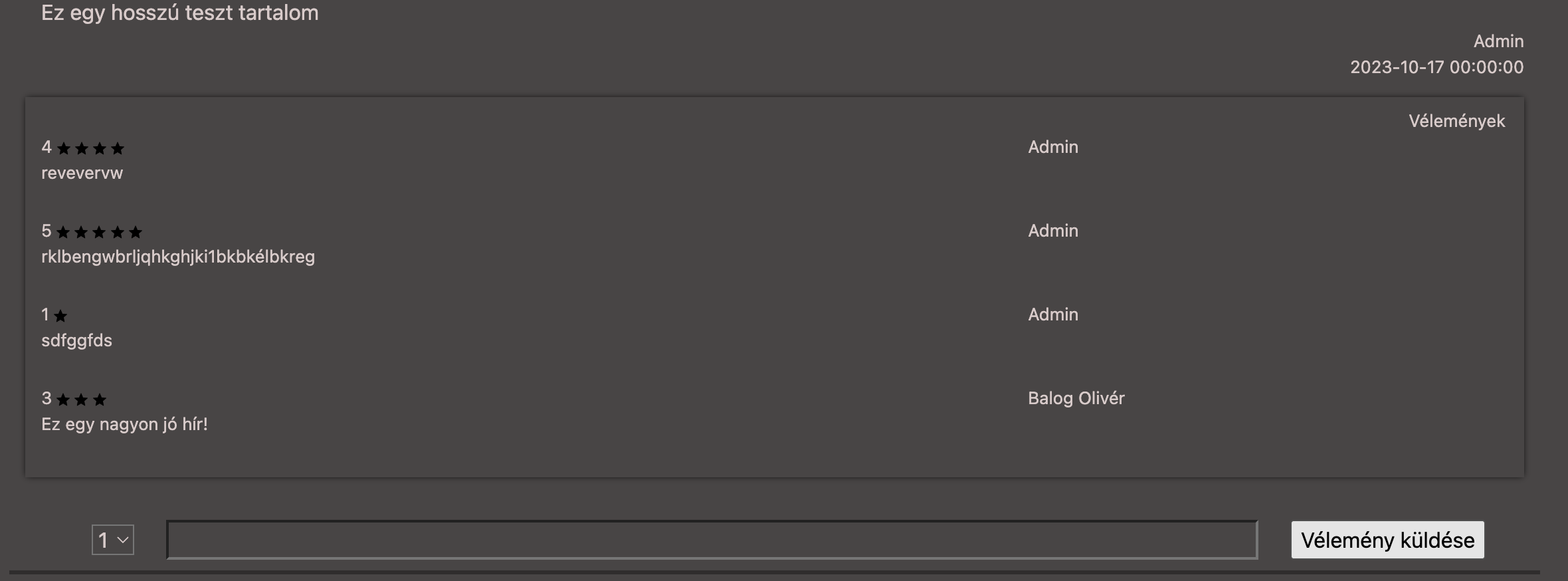
A felületen így néz ki:  


Az alábbi metódusok hajthatók végre:

* create(): egy új eszköz létrehozását teszi lehetővé
* store(): az újonnan létrehozott eszköz adatainak mentését végzi
* show(int $id): a kiválasztott eszközről gyűjti össze az összes adatot
* edit(int $id): a kiválasztott eszköz adatait módosíthatjuk vele
* update(int $id): a módosított adatok mentését végzi
* delete(int $id): a kiválasztott eszköz törléséért felel

### **A News Controller osztály**

A news és news\_rating adatbázis táblákból dolgozik és a hírek menüpontot szolgálja ki. Ez kezeli a hírek megjelenítését, illetve azok véleményezését, és új hírek írását. A menüpont a felületen csak akkor látható, ha bejelentkezett már felhasználóként.  
Metódusai:

* index(): Az összes hír kilistázása  
  
* rate(): egy hír új véleményének írása  
   
* store(): Egy új hír mentésére szolgál

## **A modellek**

### **A Model osztály**

Ebből az osztályból származik az összes model osztály, absztrakt osztályként nem kerül példányosításra. A ModelInterface és a CrudInterface szülő osztályokat implementálja.

Az alábbi függvények definiálása itt történik:

* a konstruktor
* query(): egy új statikus Model osztályt ad vissza
* getId(): a rekord ID-ját adja vissza
* raw(string $query, $fetch = 'fetchAll'): felgyűjti az összes rekordot egy nyers query alapján
* getAll($joins = null): felgyűjti az összes rekordot a táblából
* find($idOrKey, $value = null): megkeresi az adott rekordot ID, vagy értékpáros alapján
* findOrFail($idOrKey, $value = null): megkeresi az adott rekordot ID, vagy értékpáros alapján, ha nem található 404-es hibát dob
* insert(array $data): új rekord betöltéséért felel
* update(int $id, array $data): adott rekord módosítását végzi el
* delete(int $id): a rekord törlése
* db(): PDO adatbázis kapcsolatot állít fel
* checkFillableFields(array $fields): feltölthető mezők ellenőrzését végzi el

### **A Notebook, az Opsystem, a Processor és a User, Rating és News osztályok**

A notebooks, a opsystems, a processors, rating, user, news táblázatok feltölthető mezőit állítják össze.

## **A resources mappa tartalma**

Ebben a mappában kapott helyet a a weboldal összes nézete kezdve a kezdőoldaltól, az adatbázist lekérő és megjelenítő oldalakon át az MNB adatait lekérő és bemutandó oldalig, valamint a regisztrációs és belépési felület is.

### **Layouts mappa**

Ebben a mappában találhatóak a header és footer fájljai, melyek minden oldalon megtalálhatóak.  
A header jeleníti meg az oldal nevét, állítja be a public/css/app.css fájlt a formázáshoz, valamint a menüszalag megjelenítése is itt történik.

A footer jelenleg bemutató jellegű.

### **News, notebooks mappák**

Ezek a mappák tartalmazzák a notebooks, és a news táblák adatait megjelenítő oldalakat.   
A notebooks mappa tartalma:

1. index.php: a „kezdő” oldala az adott témakörnek. Itt kerül megjelenítésre az összes rekord az adott táblából. Innen lehet rekordokat megtekinteni, módosítani. vagy éppen törölni.
2. create.php: új rekord létrehozásához a felület.
3. edit.php: a create.php által létrehozott felülethez hasonló, az adott rekord módosításához szükséges felületet jelenít meg.
4. form.php: a create.php és az edit.php felületén megjelenítendő űrlap.
5. show.php: egy adott rekord külön megjelenítését állítja össze.

A news mappa tartalma:

1. create.php: az új rekordokat ezen a felületen lehet létrehozni
2. index.php: a hírek megjelenítése
3. ratings.php: a vélemények megjelenítése, ez egy beágyazott nézet az index.php fájlon belül

### **Egyébb fájlok a mappán belül**

A mappában további nézeti fájlok találhatóak:

* login.php: a regisztrált felhasználó beléptetési felülete.
* register.php: a regisztrációs oldal felülete.
* home.php: a Notebooks-php weblap kezdőoldala.
* mnb.php: a Magyar Nemzeti Banktól lekérdezhető adatok megjelenítéséért felelős fájl.
* mnbrates.php: az MNB-től lekérdezett listás adatok és a grafikon megjelenítéséhez
* status.php: session státuszok megjelenítése
* 404.php: hibás hívás, adatszolgáltatás, esetén hibaoldal felülete.

# **Továbbfejlesztési lehetőségek**

A legfontosabb tovább fejlesztési feladat az alap MVC fejlesztése, hogy még robusztusabb legyen és tágabb használati stílust lefedjen. Pl.:

* Route-ok statikus deklarálása, put, patch, delete, options HTTP metódusokkal. RouteCollection háttérbe helyezése.
* Saját query builder, az absztract Modellen kívül.
* Hiba oldalnézetek.
* Logolás.
* Adatbázis migrációk.
* Adatbázis seederek.
* Factoryk.
* Több validációs lehetőség.
* Singleton app konténer.
* Automatikus dependency injection Reflection osztállyal.
* Middleware-ek.
* Konzolparancsok.
* .env (enviroment) fájl.

# **Képernyő képek**

