Neumann János Egyetem

GAMF Műszaki és Informatikai Kar

Webprogramozás II. kurzus

Első beadandó feladat dokumentációja

Készítette:

Burka Márk

FLQNWE

Budai Zsolt

UB3RBV

2022.11.19.

**Tartalomjegyzék**

[**Feladatkiírás** 4](#_Toc120209100)

[**A program futtatásához szükséges gépi/nyelvi igények** 4](#_Toc120209101)

[**A program felépítése** 4](#_Toc120209102)

[**Az app mappa tartalma** 4](#_Toc120209103)

[**A vezérlők** 4](#_Toc120209104)

[**A Controller osztály** 4](#_Toc120209105)

[**Az AuthenticateController osztály** 5](#_Toc120209106)

[**A RegisterController osztály** 5](#_Toc120209107)

[**A HomeController osztály** 5](#_Toc120209108)

[**Az MNBController osztály** 5](#_Toc120209109)

[**A NotebookController osztály** 5](#_Toc120209110)

[**Az OpsystemController osztály és a ProcessorController osztály** 5](#_Toc120209111)

[**A modellek** 6](#_Toc120209112)

[**A Model osztály** 6](#_Toc120209113)

[**A Notebook, az Opsystem, a Processor és a User osztályok** 6](#_Toc120209114)

[**A resources mappa tartalma** 6](#_Toc120209115)

[**Layouts mappa** 6](#_Toc120209116)

[**Shared mappa** 6](#_Toc120209117)

[**Notebooks, opsystems és processors mappák** 6](#_Toc120209118)

[**Egyébb fájlok a mappán belül** 7](#_Toc120209119)

[**Továbbfejlesztési lehetőségek** 7](#_Toc120209120)

[**Képernyő képek** 8](#_Toc120209121)

# **Feladatkiírás**

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

# **A program futtatásához szükséges gépi/nyelvi igények**

A program futtatásához a README.md fálban található utasítások elvégzése szükséges.

Bongészőből a notebooks.local link beírásával az oldal kezdőképernyője betöltődik, innen érhetjük el a program többi részét.

Internetes tárhelyen elhelyezett verzióhoz az alábbi linket kövesse:

<https://budaizsolt.kolonics.info>

A program elérhető GitHub-ról is a következő linkről: <https://github.com/BurkaMark/webpro2bead1>

# **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 (I(nterfaces) 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, absztrakt osztályként nem kerül példányosításra.

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
* logout(): a kilépésért, a session lezárásáért felel

### **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 GetExchangeRates() metódus a felhasználó által megadott két deviza közti átváltási értékeket gyűjti össze és 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 NotebookController osztály**

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

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

### **Az OpsystemController osztály és a ProcessorController osztály**

A NotebookController osztály felépítésével megegyező osztályok, melyek az opsystems és processors táblák adatainak lekérését, módosítását, vagy törlését biztosítják.

## **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 osztályok**

A notebooks, a opsystems, a processors és a users 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 a közzösségi oldalakra mutató linkeket tartalmazza.

### **Shared mappa**

Ebben a mappában található az errors.php, ami a hibaüzeneteket jeleníti meg, illetve a status.php, ami a session információkat adja át.

### **Notebooks, opsystems és processors mappák**

Ezek a mappák tartalmazzák a notebooks, az opsystems és a processors táblák adatait megjelenítő oldalakat. Mindhárom mappa ugyanazt a felépítést tartalmazza.

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.

### **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.
* 404.php: hibás hívás, adatszolgáltatás, esetén hibaoldal felülete.

# **Továbbfejlesztési lehetőségek**

A legfontosabb tovább fejlesztési feladat az alap MVC fejlesztése, hogy még robusztusabb legyen és még több mindenben hasonlítson a nagytesójára, a Laravel-re. 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**













