Neumann János Egyetem

GAMF Műszaki és Informatikai Kar

Webprogramozás II. kurzus

Második beadandó feladat dokumentációja

Készítette:

Burka Márk

FLQNWE

Budai Zsolt

UB3RBV

2022.12.01.

**Tartalomjegyzék**

[**Feladatkiírás** 3](#_Toc120827430)

[**A program futtatásához szükséges gépi/nyelvi igények** 3](#_Toc120827431)

[**A program felépítése** 3](#_Toc120827432)

[**Az app mappa tartalma** 3](#_Toc120827433)

[**A vezérlők** 3](#_Toc120827434)

[**A Controller osztály** 3](#_Toc120827435)

[**Az AuthenticateController osztály** 4](#_Toc120827436)

[**A RegisterController osztály** 4](#_Toc120827437)

[**Az ApiOrderController osztály** 4](#_Toc120827438)

[**A HomeController osztály** 4](#_Toc120827439)

[**Az OrderController osztály** 4](#_Toc120827440)

[**A PizzaController osztály** 4](#_Toc120827441)

[**A modellek** 5](#_Toc120827442)

[**A Model osztály** 5](#_Toc120827443)

[**A Category, az Order, a Pizza és a User osztályok** 5](#_Toc120827444)

[**A Menu osztály** 5](#_Toc120827445)

[**A resources mappa tartalma** 5](#_Toc120827446)

[**Layouts mappa** 5](#_Toc120827447)

[**Shared mappa** 6](#_Toc120827448)

[**Pizzas mapa** 6](#_Toc120827449)

[**Orders mappa** 6](#_Toc120827450)

[**Egyébb fájlok a mappán belül** 6](#_Toc120827451)

[**Továbbfejlesztési lehetőségek** 7](#_Toc120827452)

[**Képernyő képek** 8](#_Toc120827453)

# **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/Zsolt148/pizza-php

# **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, 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.

### **Az ApiOrderController osztály**

Postman-nel meghívható procedúrákat tartalmazza.

Metódusai:

* index(): az összes orders rekordot összegyűjti
* store(): egy új orders rekordot hoz létre
* update(): egy meglévő orders rekord mezőit módosítja
* delete(): egy meglévő orders rekordot töröl
* response(): a megejelenítendő válaszokat, vagy hibaüzeneteket adja át

### **A HomeController osztály**

A kezdő oldal betöltése a feladata.

### **Az OrderController osztály**

Az orders tábla rekordjait gyűjti össze, vagy kapcsolja össze az adatbázis többi táblájával.

Metódusok:

* index(): az összes rekordot megjeleníti az orders, a pizzas és a categories táblákból
* ajax(): az ajax-os lekérdezést biztosítja, szintén mindhárom táblából kigyűjtve az adatokat
* create(): az új orders rekord létrehozását teszi lehetővé
* export(): egy orders rekord PDF-be való kiexportálásához előkészíti az adatokat
* download(): a tényleges exportálást elvégző metódus, TCPDF osztály használatával
* chart(): az orders tábla rekordjai alapján elkészített statisztikai adatokat megjelenítő grafikonnak adja át az adatokat

### **A PizzaController osztály**

Az adatbázis pizzas táblájából gyűjti össze a pizzák adatait és adja át a nézeti felületnek.

Az alábbi metódusok hajthatók végre:

* index(): az összes rekordot összegyűjti
* create(): egy új rekord létrehozását teszi lehetővé
* store(): az újonnan létrehozott pizza adatainak mentését végzi

## **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
* getPrimaryKey(): az adott tábla elsődleges kulcsát 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
* findByArray(array $params): megkeresi az adott rekordott a tömbben megadott paraméterek alapján
* 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 Category, az Order, a Pizza és a User osztályok**

A notebooks, a opsystems, a processors és a users táblázatok feltölthető mezőit állítják össze.

### **A Menu osztály**

A menüket tartalmazó menu táblából kiolvasható adatokat állítja össze a nézeti fájloknak.

Függvényei:

* mapMenus(): felméri a táblában található összes rekordot, illetve kigyűjti az összes szülő rekordot
* recursive($menus, $parent\_id = null): összeállítja a szülő-gyerek kapcsolatokat a rekordok között

## **A resources mappa tartalma**

Ebben a mappában kapott helyet a a weboldal összes nézete, a kezdőoldaltól az adatbázist lekérő és megjelenítő oldalakig, 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.

### **Pizzas mapa**

Ebben a mappában található a pizzas tábla adatait megjelenítő oldalakat.

Felépítése:

1. index.php: a „kezdő” oldal, ahol megjelenítésre kerül az összes rekord a táblából. Innen lehet új rekordot létrehozni.
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.

### **Orders mappa**

Az orders tábla adataival kapcsolatos nézeti fájlok találhatóak itt:

1. index.php: az összes rekordot megjelenítő oldal. Továbbá itt lehet az Ajax használatával egy adott rekord minden adatát lekérni.
2. chart.php: az orders statisztikai adatai alapján elkészült grafikont megjelenítő oldal
3. create.php: az új rekordot létrehozó oldal
4. export.php: a PDF-be exportálás felületét jeleníti meg

### **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 Pizza-php weblap kezdőoldala
* 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**













