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](#_Toc121161486)

[**Feladatok megvalósítása** 3](#_Toc121161487)

[**A program futtatásához szükséges gépi/nyelvi igények** 4](#_Toc121161488)

[**Belépési adatok** 4](#_Toc121161489)

[**A program felépítése** 4](#_Toc121161490)

[**Az app mappa tartalma** 4](#_Toc121161491)

[**A vezérlők** 5](#_Toc121161492)

[**A Controller osztály** 5](#_Toc121161493)

[**Az AuthenticateController osztály** 6](#_Toc121161494)

[**A RegisterController osztály** 6](#_Toc121161495)

[**Az ApiOrderController osztály** 6](#_Toc121161496)

[**A HomeController osztály** 8](#_Toc121161497)

[**Az OrderController osztály** 8](#_Toc121161498)

[**A PizzaController osztály** 8](#_Toc121161499)

[**A modellek** 9](#_Toc121161500)

[**A Model osztály** 9](#_Toc121161501)

[**A Category, az Order, a Pizza és a User osztályok** 9](#_Toc121161502)

[**A Menu osztály** 9](#_Toc121161503)

[**A resources mappa tartalma** 9](#_Toc121161504)

[**Layouts mappa** 9](#_Toc121161505)

[**Shared mappa** 10](#_Toc121161506)

[**Pizzas mapa** 10](#_Toc121161507)

[**Orders mappa** 10](#_Toc121161508)

[**Egyébb fájlok a mappán belül** 10](#_Toc121161509)

[**Továbbfejlesztési lehetőségek** 11](#_Toc121161510)

[**Képernyő képek** 12](#_Toc121161511)

# **Feladatkiírás**

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

# **Feladatok megvalósítása**

Az egész projekt a Laravel keretrendszer mintájára épült követve az OOP, SOLID és MVC elveket.

1. **Feladat:** resources/views mappabán van az összes view nézet fájl. A header komponens a layouts/header.php -ban, a footer pedig a footer.php-ban van. Itt található többek között a navbar, a flowbite.css, flowbite.js és a jquery.js script tagje is. A flowbite.css egy tailwind css-re épülő keretrendszer, javascriptel kiegészítve. Innen jönnek az alap stílus osztályok és ez adja az oldal reszponzívitását. A home.php a főoldali nézet, a login.php a bejelentkezési oldal, a register.php a regisztrációs oldal és a 404.php a nem található URL oldala. Az pizzas mappa egy CRUD mappa. Index, create, edit, show, form php fájlokat vannak benne. Ezek a CRUD műveletek mappái. A form egy megosztott fájl, a create, show és az edit is használja.
2. **Feladat:** 
   1. Az objektumorientált elvet úgy sikerült elérni, hogy minden táblát egy Model osztály reprezentál az app/Models mappában. Ezek extendelik az alap Model.php fájlt, ami pedig implementálja a ModelInteface-t és a CrudInterface-t. Ennek a konstruktora állítja be singleton static módon az adatbázist.
   2. A menüpontokat a Menu.php model osztályon keresztül érhető el. Ez a menus táblából hívja le menüpontokat. Található benne egy mapMenus function ami rekurzív módon mappolja a menüpontokat és ez által vissza adja a gyerekeket.
   3. A resources/views/header.php fájlban ez a többszintes menü látható is, ahogyan kilistázza a menüpontokat és a gyerekeket egy dropdown-ba.
   4. A jogosultsági jogkörök a User model osztályban vannak, azaz a users táblában a role mezőben. Ennek a segítségére van az app/Helpers/Auth.php ami mindig a bejelentkezett felhasználót adja vissza statikusan. Statikus függvényekkel lehet lekérni a felhasználó id-ját, user objectjét és a jogosultságát.
3. **Feladat:** Az ajax megvalósítás nézete a resources/views/orders/index.php-ban található. A javascript fájl pedig a public/js/orders.js-ben. Itt minden egyes dropdown módosításra az app/Controllers/OrderController.php ajax function-ja vissza adja a megfelelő adatot.
4. **Feladat:** A Restful webszolgáltatlás backend-je az app/Controllers/ApiOrderController-ben van. Itt megtalálható a GET index, POST store, PUT update és DELETE delete function is. A Postman tesztelés lépései a dokumentációban később, az **ApiOrderController** -nél található.
5. **Feladat:** A TCPDF szolgálatás nézete a resources/views/orders/export.php -ban van, ezt a nézetet az app/Controllers/OrderController.php export function-ja adja vissza. A letöltés gombra kattintva POST metódussal a download function betölti az exportált PDF-et.
6. **Feladat: -**
7. **Feladat:** A grafikon szintén a resources/views/orders mappában található, a chart.php fájlban. Ennek a nézetét szintén az OrderController.php tölti be, ezen belül pedig a chart function. Az orders táblából kiszedi a pizzák rendeléseinek mennyiségét. Ezeket az adatokat betölti a chart.js, és beállítja egy fektetett grafikonba random színnel.
8. **Feladat:** [**https://budaizsolt2.kolonics.info/**](https://budaizsolt2.kolonics.info/)
9. **Feladat:** [**https://github.com/Zsolt148/pizza-php**](https://github.com/Zsolt148/pizza-php)
10. **Feladat:** [**https://github.com/Zsolt148/pizza-php/commits/main**](https://github.com/Zsolt148/pizza-php/commits/main)

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

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

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

Internetes tárhelyen elhelyezett verzióhoz az alábbi linket kövesse:

<https://budaizsolt2.kolonics.info>

A program elérhető GitHub-ról is a következő linkről:

<https://github.com/Zsolt148/pizza-php>

# **Belépési adatok**

Admin:

Email: [admin@admin.com](mailto:admin@admin.com)

Jelszó: admin

User:

Email: [user@user.com](mailto:user@user.com)

Jelszó: user

# **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.

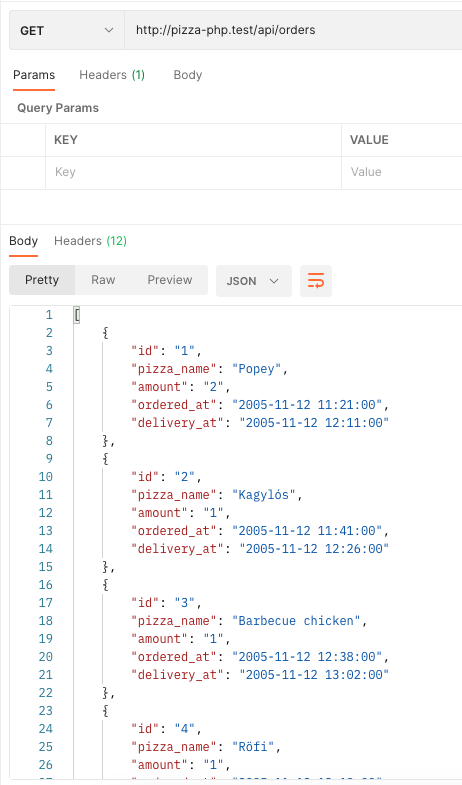
### **Az ApiOrderController osztály**

Postman-nel meghívható procedúrákat tartalmazza.

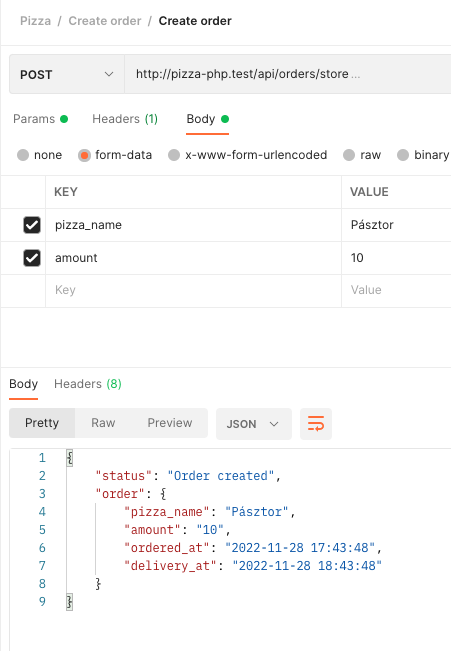
Metódusai:

* 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

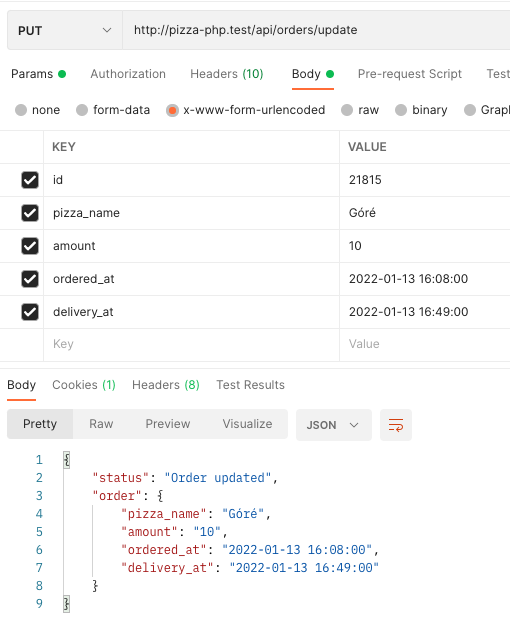
**Tesztelés lépései:**



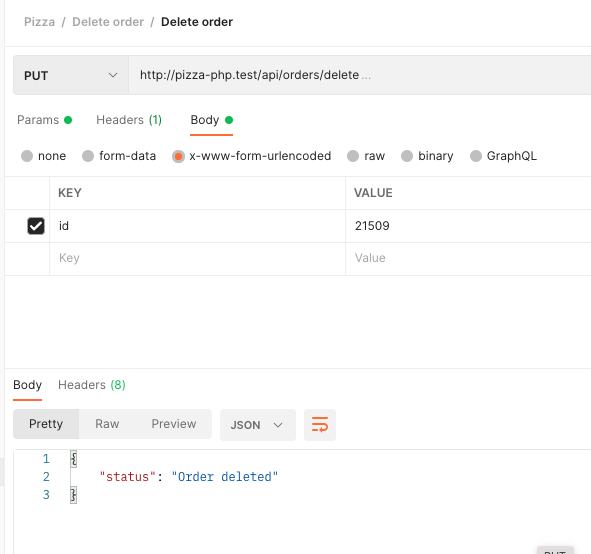
Restful API GET lekérdezés



Restful API POST létrehozás



Restful API PUT frissítés



Restful API DELETE törlés

### **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.
* Model factory-k.
* Több validációs lehetőség.
* Singleton app konténer.
* Automatikus dependency injection Reflection osztállyal.
* Middleware-ek.
* Konzolparancsok.
* .env (enviroment) fájl.

# 

# **Képernyő képek**

