2023

Ruzsinszki Zita

Harc a CakePHP-vel

2023.01.05.

HACKEREK KFT.



Tartalom

[1. MVC – modell nézet vezérlő 2](#_Toc123836776)

[2. PHP keretrendszerek 2](#_Toc123836777)

[3. CakePHP „telepítése” 3](#_Toc123836778)

[4. Táblák szerkezete az adatbázisban 4](#_Toc123836779)

[5. Sütés 5](#_Toc123836780)

[6. Helperek 8](#_Toc123836781)

[7. Nézetek testreszabása: 12](#_Toc123836782)

[8. Nyitó lap átalakítása 16](#_Toc123836783)

[9. Magyarítás 17](#_Toc123836784)

[10. Beléptető rendszer 19](#_Toc123836785)

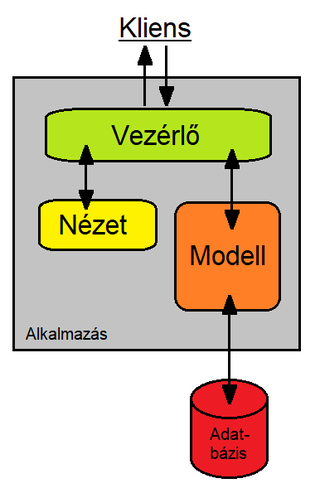
[11. Beléptető űrlap 20](#_Toc123836786)

[12. Kilépés 22](#_Toc123836787)

[13. Linkek: 23](#_Toc123836788)

[14. Számonkérés 23](#_Toc123836789)

# MVC – modell nézet vezérlő

****[**https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91**](https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91)

Az információt az adatot jelenti, illetve az adatkezelés szabályait tartalmazza

felhasználói felület

a controller össze kapcsolja a view-t a nézetet és a modellt, ami a felhasználóval is tartja a kapcsolatot, vagyis a usertől érkező akciók, billentyű leütések, illetve egér kattintások, ezeket feldolgozza és lekérdezéseket továbbít, a modellhez, és amit a modelltől vissza kap, azt továbbítja a viewnek, azaz a megjelenítésnek.

# PHP keretrendszerek

pl: CakePhp, Codelgniter, Laravel, Symfony, Zend Framework

Telepítéshez a composert kell telepítenünk.

Innen le tudod tölteni: <https://getcomposer.org/download/>

A composer ahelyett, hogy egy központi adatbázisban követné nyomon a telepített csomagokat, egyszerűen csak letölti és telepíti a csomagot egy szoftvertárból, egy mappába a számítógépeden, vagy a szervereden. Így ideális arra, hogy különböző alkalmazásokhoz különböző keretrendszereket telepítsünk a webszerverre.

# CakePHP „telepítése”

Magát a cakephp-t nem kell telepítenünk, mert ahhoz nincs telepítő program, hanem elég, ha telepítettük a composert és a következő paranccsal létre hozhatjuk a cakephp projektünket.

Nyisd meg „open folder” xampp mappa htdocs és úgy add majd ki a parancsot!

CakePHP: Document root mappát vagyis a htdocs nyissuk meg a vsc-ben és a terminálon írjuk be:

de előbb! ***a php.ini-ben engedélyezni kell az intl extensiont***, vagyis ki kell venni előle a ; jelet és menteni, majd újra indítani a xamppot; engedélyeket fog kérni, figyelni kell

<https://book.cakephp.org/4/en/installation.html>

composer create-project --prefer-dist cakephp/app **hackerek**

Úgy is telepíthed, hogy : cmd – C:\xampp\htdocs\composer create-project --prefer-dist cakephp/app:~4.0 hackerek

Ha lefutott a program, akkor látod, létre jön az hackerek mappa és tele lesz fájlokkal, jobb gomb tulajdonságok, látod hogy 5 784 fájlt, 1 081 mappában hozott létre.

El is készült az oldalad főoldala: <http://localhost/hackerek/> íme és ha látod a cakephp logót és verzió számot, akkor már jó

Specifikáció: Egy informatikai vállalkozás admin felületét készítjük el, lesznek ügyfelek és lesznek informatikusok, lesznek időpontok, amit az ügyfelek tudnak majd szolgáltatást és időpontot foglalni, ügyfeleknél megadjuk melyik szolgáltatást választották, illetve az ügyfeleknek tudunk számlákat kiállítani, lesz egy számlák tábla is.

*Hozzunk létre egy hackerek adatbázist importáljuk be az sql mappát, angol nevek vannak benne, mert a cakephp az adatszerkezet alapján az adatáblák mező alapján automatikusan el fogja készíteni majd az alaprendszert, ezt úgy csinálja, hogy megvizsgálja a mező neveket, táblaneveket, felismeri automatikusan a többes számot, pl admins, ez tudni fogja, hogy ez többes szám, és ha új admint akarunk létre hozni azt egyes számban hozza létre.*

# Táblák szerkezete az adatbázisban

**Admins:** ide jönnek az adminok, lesz egy bejelentkezési oldal. Minden tábla első mezője az id ami elsődleges kulcs, és autoincremet, int típusú, automatikusan 11. Adminnál lesz egy név, karakter típusú, email és jelszó, ami varchat karakteres típus egyszerűség kedvéért 255 –ös minden.

Minden táblában lesz majd egy mező, hogy created és modified, ezek date time típusúak, tehát dátum és idő lesz majd bennük eltárolva. Erre azért van szükség, mert a cakephp automatikusan fogja ezeket a mezőket frissíteni. Amikor elkészítünk egy új rekordot akkor a created, amikor módosítjuk akkor a modified mezőt fogja automatikusan kitölteni. Ha más neveket használnák a táblában akkor nem ismerné fel a cakephp ezeket a mezőneveket, és nem működne a rendszer amit automatikusan el tud nekünk készíteni.

**Customers tábla**: akik majd különböző szolgáltatásokat szeretnének és időpontot szeretnének foglalni, itt is van id, itt is van created és modified, ez minden táblában így van. Láthatod, hogy milyen adatokat tudunk felvenni, mailcim, cím, teló stb. és service id itt a szolgáltatás azonosítója lesz majd eltárolva.

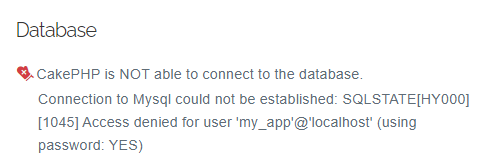
**Developers:** az informatikusok listája, itt a nevük és szabadságon van-e, szóval az állapot kerül eltárolásra.

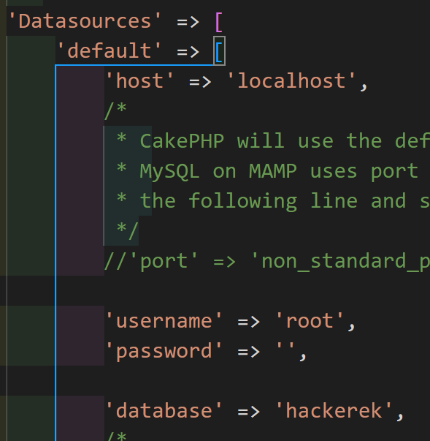
**Services:** szolgáltatások id created és modified mellett a szolgátatások neve, mint például weblapkészítés, hálózat tervezés, rendszergazdai szolgáltatások stb.

**Invoices**: ügyfelekhez kapcsoldónak, itt lesz egy customer id, az ügyfél azonosító, számla végösszege, services, milyen szolgáltatásokat vett igénybe és fizetési határidő.

**Appointments:** időpontfoglalások eltárolása. Dátum és idő stb.

Ha figyelmesen megnézed a kezdő oldalt, középtájon találkozhatsz egy ilyen kis hiba üzenettel:



Ehhez egy fájlon belül kell megadnunk az mysql adatait, jelszó, adatbázisnév ilyesmik.

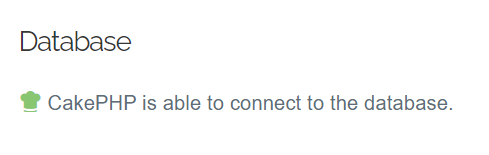
**hackerek – config mappa ezen belül az app\_local.php ezt fogjuk szerkeszteni.**

keresd meg: ctrl+f datasources, kicsit görgess le, ott is írd át a tesztnél!

ezeket írjuk felül, a szokott módon, frissítsük az oldalunkat és máris látni fogjuk, hogy megjelenik a kezdő oldalon a kapcsolat, eltűnik a hiba üzenet. Ha kész bezárhatod.

***A cakephp automatikusan el fogja nekünk készíteni a kis webes felületünket, az adatbázis szerkezet alapján***.

Most ezt kell látnod, ha az oldalra ráfrissítesz:



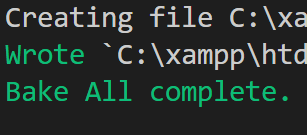
Ha megnézzük a mappákat src-n belül van controller, modell és view. MVC összetevői!

Tempalte – ebben mindenféle oldal van, de még nincs olyan, ami nekünk kellene az adattábláinkhoz, tehát a következő lépés, egyesével ki kell sütni az adattábláinkat. A cakephp nyelvezetében a bake azt jelenti kisütés. ☺ Ehhez minden adattáblára vonatkozóan ki kell adnunk a terminálban egy parancsot.

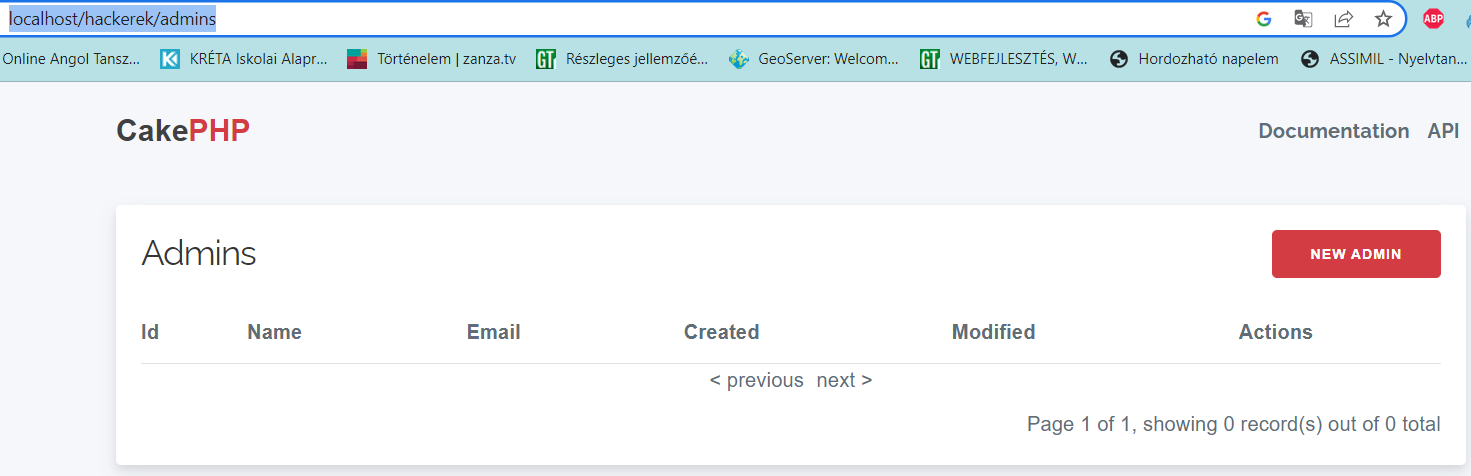
# Sütés

Lássunk hozzá a sütéshez: nyisd meg a terminált a vsc-ben, de az hackerek mappád is legyen nyitva: C:\xampp\htdocs\hackerek> majd add ki itt a sütés parancsot a következő módon:

bin\cake bake all admins

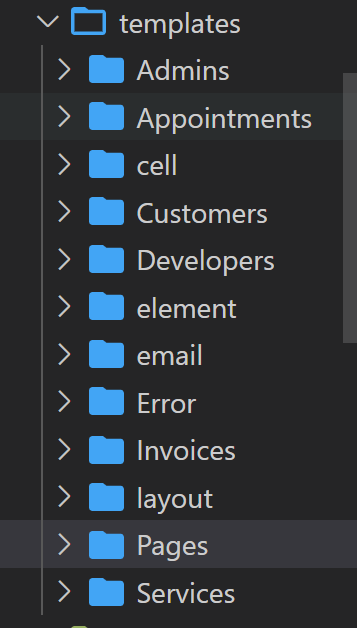
ha ezt látod: Bake All complete. az jó, mert akkor minden kisült rendesen az adatbázisod.

Írd be a böngésző címsorába: <http://localhost/hackerek/admins> ha ezt látod, ügyesen sütöttél:



Ha most megnézzük a template mappa tartalmát, akkor elkészült egy admins mappa, de az src controlleren belül is készült egy új fájl, adminsController. Nézzük meg modell mappát, ezen belül entity, ezen belül megjelent egy admin.php. Tehát ahogy kisütöttük az admins táblát, újabb mappák és fájlok jelentek meg.

Mehet a sütés tovább:

Jelöld, amelyik már megvan (húzd át a word-ben), nehogy kimaradjon egy!!!

bin\cake bake all customers

bin\cake bake all developers

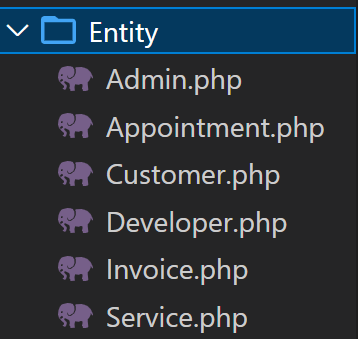
bin\cake bake all appointments

bin\cake bake all services

bin\cake bake all invoices

Ha most újra megnézed a template mappádat láthatod, hogy egy halom új mappát kaptál.

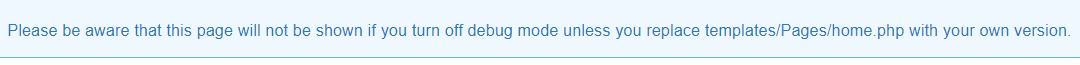
Kell lennie: Admin, Appointments, Customers, Developers, Invoices, Services mappáidnak. Ellenőrizd le újra megvannak e!



Az **=>src =>modell =>entity** mappán belül pedig megjelent az összes php fájl az adatbázis alapján.

***Itt megfigyelhető, hogy az adattábla többes számából egyes számban hozta létre a neveket. Tehát ezért kell az angol nyelvet használni, amikor az adatszerkezetet elkészítjük, mert ebből tudja a cakephp ezeket elkészíteni***.

A nyitó lapunk még továbbra is ugyanaz lesz, ha ráfrissítünk, viszont ott figyelmezetet, melyik mappában lévő fájlt szerkesszük, ha saját nyitó lapot szeretnénk készíteni. templates/Pages/home.php



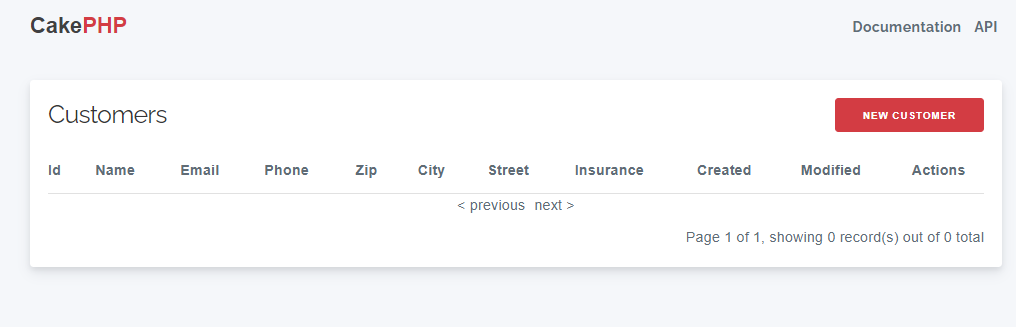
Jelenleg még a hibakereső üzemmód fut, jobbra alul ha kattintunk a  ikonra, akkor sok lehetőség megnyílik.



itt rákattintva az egyes menüpontokra ki lehet őket nyitni, próbáld is ki

**Na de nézzük mit sütöttünk ☺**

<http://localhost/hackerek/customers>



Egy gyönyörű reszponzív kész oldalt kapunk, igaz még vannak olyan menüpontjai felül, amik külső oldalakra mutatnak, mint a Documentation és API.

Ahhoz, hogy ügyfelet, vagy akár fejlesztőt tudjunk fel venni, először mindenekelőtt fel kell venni szolgáltatásokat, hogy ki tudjuk választani, hogy milyen szolgáltatást rendelt meg tőlünk. Nagyon egyszerű újat felvenni, mert csak be kell írni az informatikai szolgáltatás nevét. Pl: webdesign, weboldal készítés, adatbázis készítés, rendszerfelügyelet, hálózat tervezés.

Vegyünk fel ügyfeleket, látni fogjuk, hogy a szolgáltatás nevéből linket csinál, ha rá kattintunk, megjelenik a szolgáltatások oldala a kapcsolt ügyféllel, aki az adott szolgáltatást megrendelte tőlünk.

Vegyünk fel az informatikusokat – ide mehet az osztály névsorából a csapatod ☺ -, check bokszal tudjuk jelölni, ha valaki szabin van, de most nagyon ne menjetek szabira, mert ha valaki hiányzik, nem tudjuk végigcsinálni együtt.

Érdemes rátekinteni az adatszerkezetre, hogy miből alkotta meg nekünk ezt a checkboxot, mennyire okos a rendszer. Ha kattintunk a megtekintésre, akkor láthatóvá válnak a kapcsolódó adatai is.

Vigyünk fel számlát is.

**Felső menüsor módosítása**, két linkünk van a dokumentáció és az api. Oda rakjuk inkább az adminok rendezése és kilépés menüpontot. Itt találod, amit módosítani kell:

**template – layout –default.php**

A két linknél a target\_blankra nem lesz szükség, mert ugyanott, ugyanabban az ablakban szeretnénk megnyitni a linkeket. (kb 44, 45-ik sor)

A „rel="noopener"” ***linktípus,*** amit itt látsz, azt tudja, hogy arra utasítja a böngészőt, hogy navigáljon a célerőforráshoz anélkül, hogy az új böngészési környezet számára hozzáférést adna a dokumentumhoz, amely megnyitotta – azáltal, hogy nem állítja be a tulajdonságot a megnyitott ablakban. Ez különösen akkor hasznos, ha nem megbízható hivatkozásokat nyit meg a hivatkozásunk, hogy biztosítsa, hogy azok ne tudják manipulálni az eredeti dokumentumot a Window.opener tulajdonságon keresztül. De most erre sincs szükségünk, hiszen helybe topogunk.

Egyik menünk legyen az Admins, és egy Logout. *(Ne magyarítsd, és ugyanígy írd, deary John Wick & Milan, kérlek, ne írjatok be funny szavakat, mert nem fog működni a fordító po fájl, amit adok!!!)*

Adminsnak megadjuk az útvonalat: href="/hackerek/admins" a Logutnál egyenlőre #.

Nézd meg a változást ☺

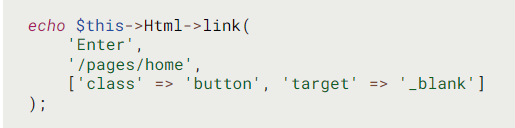
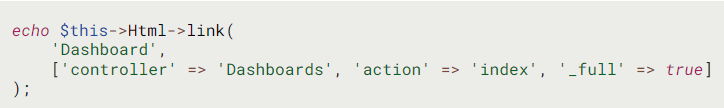
# Helperek

A linkeket ún. **helperekkel** tudjuk legprofibban megadni. Az hogy mik azok a helperek megnézhtjük a cakePHP dokumentációjában. cakephp.org – documentation – book – és megnyílik a cookbook ún szakácskönyv, a keresőjébe írjuk be azt, hogy helpers: <https://book.cakephp.org/4/en/views/helpers.html>

Itt láthatod, hogy a helperek componens osztályok, amelyek segítenek a megjelenés egységesítésében. Ha a cookbookban lefelé görgetsz, látható milyen beépített helper osztályok találhatóak a CakePhp-ben. Ezek között megtalálod azt amire nekünk szükségünk lesz a HTML nevű helper. A HTML helper arra jó, hogy html kódot fog beilleszteni helyettünk az oldalra, mi csak meghívjuk ezt a helpert, átadunk neki paramétereket, és ebből ő majd elkészíti a html kódot.

jobbra a tartalom jegyzékben találunk egy creating links – kattintsunk rá

<https://book.cakephp.org/4/en/views/helpers/html.html#creating-links>

Itt látunk is egy példát, *echo $this->Html->link()* ezzel tudunk hivatkozni erre a html helper osztályra.

ezzel tudunk hivatkozni erre a html osztályra. Ennek a link funkciójának működését fogjuk megnézni, ennek a link függvénynek, legalább 2 paramétert kell majd átadnunk.

Picit lefelé görgetve látsz egy másik példát, ezt fogjuk mi használni. Tehát az első paraméter az lesz, ami a link szövege, vagyis mi jelenjen meg, ami egy egyszerű szöveges típus lesz. A következő paraméter pedig egy tömb lesz. A tömbben pedig két elemet fogunk megadni, az első a kontroller lesz, pl ha az ügyfeleket szeretnénk kilistázni, akkor a kontroller az lesz, hogy costumers. Az action, a listázás az pedig az index. Tehát ezt a két elemet kell felvennünk a tömbe.

A **default.php fájl 44 ik** sorában először a customers linkjét fogjuk átalakítani így, de ezt még ne illesztd be gyorsba! Várj, had mondjam végig! <?=$this->Html->link('Customers', [])?>

Van egy módszer, amivel a többnyelvűsítést a weboldal lefordítását lehet megoldani, a customerst zárójelbe kell tenni, és elé írni két darab alsó aláhúzást, a cakephp pedig ha ennek megtalálja a magyar fordítását, oda be fogja helyettesíteni. <?=$this->Html->link(\_\_('Customers'), [])?> még, mindig várj!

Tehát valahogy: <?=$this->Html->link(\_\_('Customers'), ['controller' => 'Customers', 'action' => 'index', '\_full' => true])?> így fog kinézni a kód, ami bonyolultnak tűnik, de ha módosítani akarsz később, akkor csak egy helyen kell megtenni. Ez a lényege az MVC keretrendszernek, hogy minden módosítást csak egy helyen kell megtenni, de ugyanakkor sok helyen egyszerre módosulnak pl a menük. Így ha a html helpert használjuk, akkor egységes lesz a linkek megjelenítése, és ha ezen változtatni akarunk, akkor csakis egy helyen kell módosítani. Végig átírhatjuk az összeset, annyi hogy a kilépés még nem fog működni.

Ezeket illeszd be a két sor link helyére, a 44-45-ik sorba: Most mehet ☺

<?=$this->Html->link(\_\_('Customers'), ['controller' => 'Customers', 'action' => 'index'])?>

<?=$this->Html->link(\_\_('Services'), ['controller' => 'Services', 'action' => 'index'])?>

<?=$this->Html->link(\_\_('Developers'), ['controller' => 'Developers', 'action' => 'index'])?>

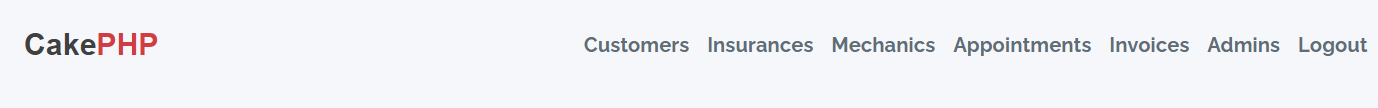
<?=$this->Html->link(\_\_('Appointments'), ['controller' => 'Appointments', 'action' => 'index'])?>

<?=$this->Html->link(\_\_('Invoices'), ['controller' => 'Invoices', 'action' => 'index'])?>

<?=$this->Html->link(\_\_('Admins'), ['controller' => 'Admins', 'action' => 'index'])?>

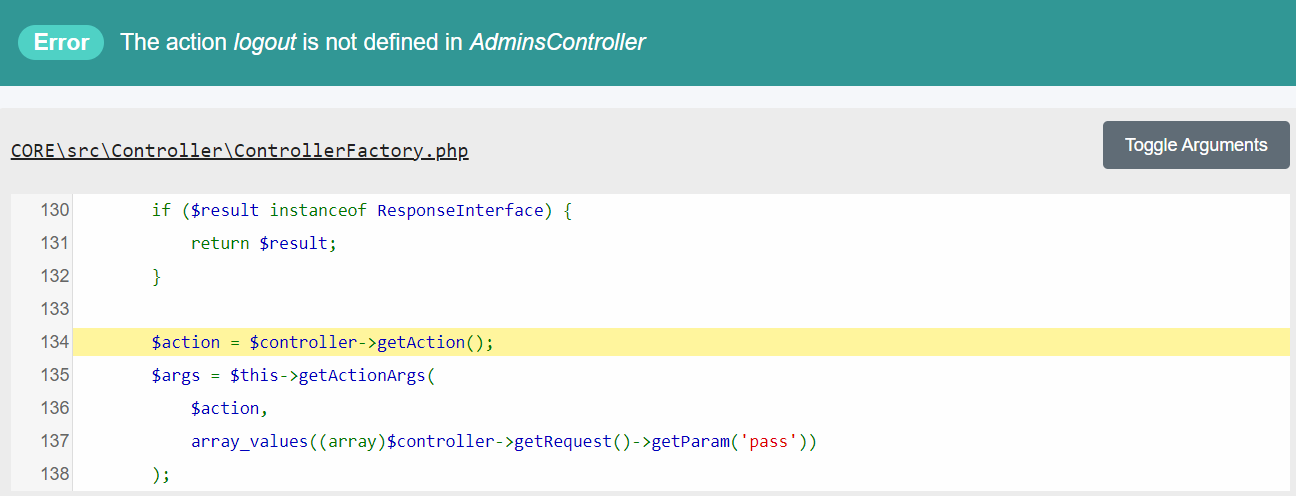
<?=$this->Html->link(\_\_('Logout'), ['controller' => 'Admins', 'action' => 'logout'])?>

Az utolsónál a controllert szintén az admin az aciton a pedig logout lesz.

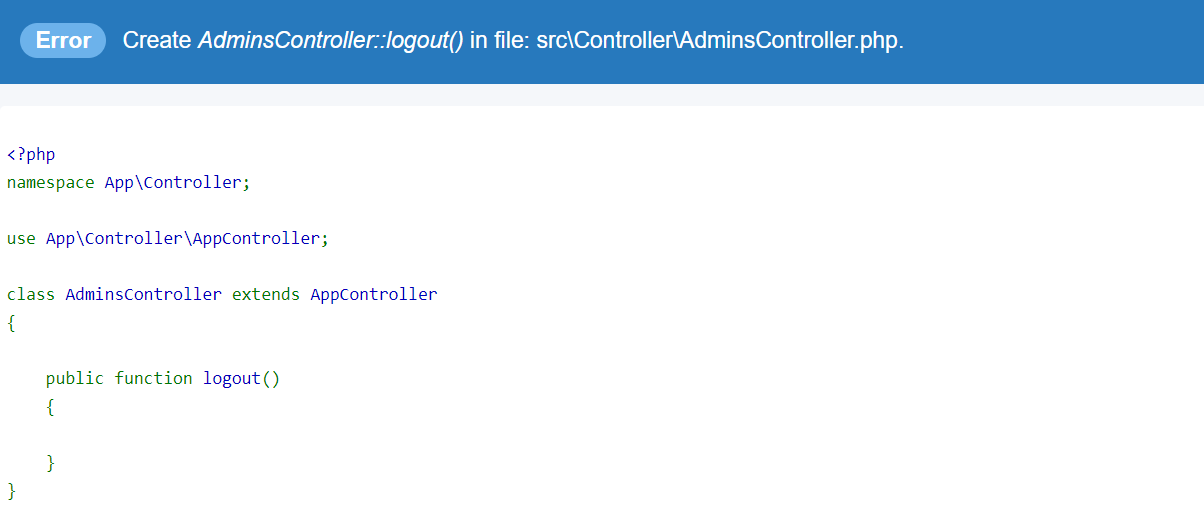


Próbáljuk ki a menüket.

A Logout - kilépésre hibaüzenetet fog adni, azt mondja: The action logout is not defined in AdminsController.



Nagyon szuper, mert megmondja, mit kell tennünk:



Create AdminsController::logout() in file: src\Controller\AdminsController.php. és alul írja is mi a dolgunk, hozzá kell adni egy üres metódust a: public function logout(){} kell felvennünk az ->src -> controllers -> adminscontrollerbe, de vigyázz az osztályon belül, tehát az utolsó zárójel elé!

*Az évek során nagyon sokat javítottak a keretrendszeren, az oldalsávból eltűnt a menü, minden menüre kattintva külön tudsz hozzá adni új elemeket. Alul automatikusan megjelenik a gördítő menü, nem csúsznak egymásra a telefonszámok, email címek, és minden sor végén ott van a szerkesztési lehetőség.*

Ha most kattintunk a logutra újabb hibaüzenetet kapunk, hogy a template hiányzik, szépen le is írja mit kér: C:\xampp\htdocs\autoszerviz\templates\Admins\logout.php



A fenti képen láthatod, hogy hol és milyen nevű fájlt kell létrehoznod. Tehát templates mappa, azon belül Admins mappába hozd létre a logout.php nevű fájlt. Hozzuk létre, és egy üres oldalt kapunk a hiba helyett, ami már jó.

# Nézetek testreszabása:

Nyisd meg a következőt: templates - customers – index.php

itt pl kivehetjük azokat az oszlopokat, amelyeket nem akarunk megjeleníteni. pl id, created és modifieyd, kivehetjük a fejlécből és a tartalomból, kényelmesebb a nézet így már nem kell görgetni sem, de ha akarod az alkalmazásodban benne is hagyhatod. Ha viszont törölsz, nézd meg, hogy ne csak a fejlécből töröld ki, hanem az alatta lévő cellákban sem jelenjen meg, különben el fogja tolni a táblázatod.

A view módban viszont még mindig benne van, (kattints a böngészőbe a valamelyik view-ra, ott látni fogod) de ha nem akarjuk, hogy ott is megjelenjen, akkor onnan is törölni kell. Itt nézd: http://localhost/autoszerviz/customers/view/ … itt tudjuk kivenni a fölös 3 sort a view.php-ből

Az oszlopok pont olyan szélességűek, mint amilyenre szüksége van a tartalomnak, erre sincs szükség, hogy állítgassuk és teljesen reszponzív az oldal.

**Javítsunk tovább a megjelenésen:**

Kattints a Developers menüre. Állítsd be valamelyik fejlesztőnél, hogy szabin van. Így már láthatod, hogy a fejlesztők esetében 1 jelenik meg az on vacation-nál, viszont a view-ban irja, hogy yes, tehát a view.php-ből át tudjuk másolni a kódot a developer index.php-be, a foreach alá a kb 27-ik sorba így:

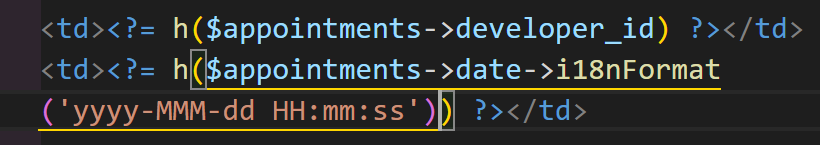
<td><?= $developer->on\_vacation ? \_\_('Yes') : \_\_('No'); ?></td>

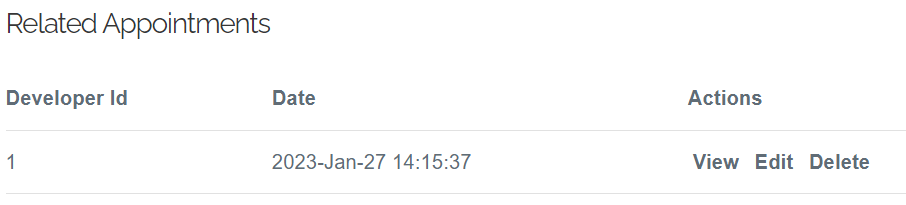
**h()** függvény, amit a view.php-ben látsz (58 sor környékén) a htmlspecialchars függvényt helyettesíti, igazából így egy kényelmi függvény, vagyis a speciális html karaktereket átalakítja, és úgy jelenti meg.

**Jelenítsük meg az ügyfélnél a kapcsolattartó fejlesztő nevét.** Adjunk hozzá egyeztetett időpontot valamelyik fejlesztőnkhöz, hogy alul lássuk a szerelő id-jét.

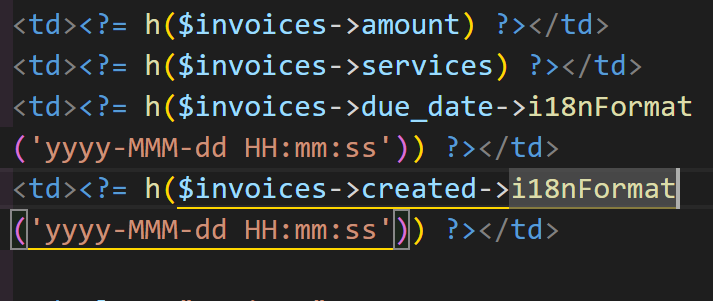
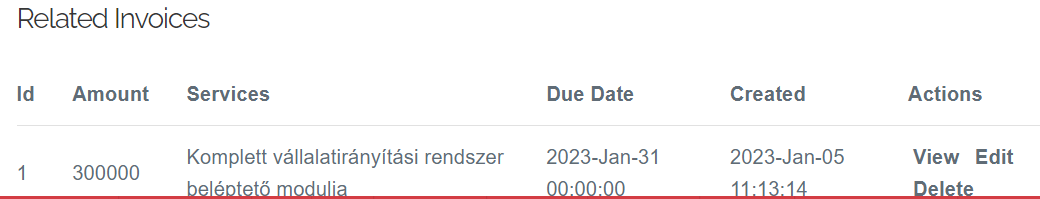
Saját helper készítése: ha az ügyfelekre kattintunk és kiválasztunk egyet a viewre kattintva, ott a fejlesztő azonosítója jelenik meg nem pedig a neve, pedig jobb lenne a neve, és a dátum formátuma sem a legmegfelelőbb.

Igazából az id-re és a costumer\_id-re, created és modify oszlopokra sincs itt szükség, jobb lenne, ha csak a kapcsolattartó fejlesztő neve látszana, így ezeket a sorokat ki is törölhetjük a view-ból.

A dátum sem a legtetszetősebb formátumban jelenik meg. <https://book.cakephp.org/4/en/core-libraries/time.html> ezen az oldalon lefelé görgetve megtalálható, i18nFormat('yyyy-MMM-dd HH:mm:ss') ezzel lehet nemzetközi idő és dátumformátumot megadni. Ha ezzel kiegészítjük a a cosutmers – view.php kb 67-ik sorát, akkor már szépen fog megjelenni az oldalunkon a dátum is, a kívánt formában.



A fizetési határidőnél is javítsuk ki, a 101-ik sor környékén fogod megtalálni, ahol ugyanígy ki kell egészíteni a kódot.

Ahol van related invoices ott is nézzük meg, mi az amire valójában szükségünk van. A számla id-re szükség lehet, de a costumer\_id fölösleges, hiszen az adott ügyfél alatt jelenik meg, még talán a módosítás sem szükséges, tehát ezeket is törölhetjük a costumer view.php fájlunkból. A dátumot a created-nél is formázzuk át. A böngésző frissítése után szépen átalakul a táblázatunk. 

Az összeghez kiírathatod azt is hogy forint: <td><?= h($invoices->amount) ?> **Ft**</td>

Nem néz ki jól ott az az Developer Id, helyette, jó lenne, ha a fejlesztő neve jelenne meg.

A customers view.php-ben meg kell keresni azt a cellát ahol ez a kiírás megtörténik. 69-ik sor körül meg is található a <td><?= h($appointments->developer\_id) ?></td> , tehát itt a fejlesztő azonosítója van kiíratva, ennek a helyére illesszük be a saját helperünket, aminek az lesz a neve, hogy Format és ehhez egy format.php-t is létre kell hoznunk, tehát így fog kinézni:

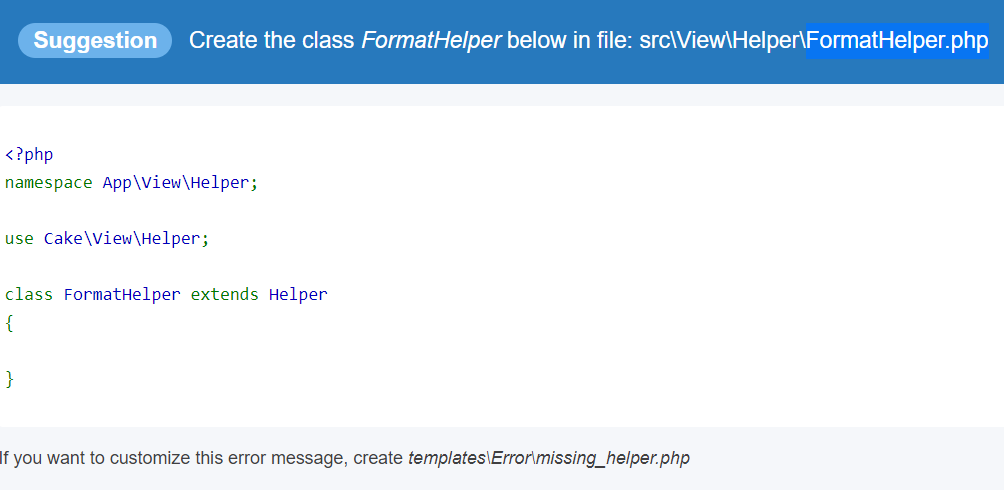
<td><?= **$this->Format->getName**($appointments->developer\_id, **'developers'**) ?></td>

Tehát a helperünk neve az lesz, hogy format, és ezt a getName függvényt fogjuk itt meghívni, vagyis a név lekérdezése, majd átadjuk neki a fejlesztő azonosítóját, és a controllert is vagyis a developers-t.

Így a getName függvény a Format-on belül le fogja kérdezni az adott fejlesztőt az id alapján és vissza adja a nevét.

Ha most ráfrissítünk az oldalra, hiba üzenetet ad, mert még nem hoztuk létre a format helpert, és nem találja.

Ugyanakkor azt is megmondja mit, hol és milyen tartalommal kell létrehoznunk.



SRC-VIEW-HELPER . és FormatHelper.php

Ezt itt alul egy az egybe bemásolhatod:

<?php

namespace App\View\Helper;

use Cake\ORM\TableRegistry;

use Cake\View\Helper;

class FormatHelper extends Helper{

**public function getName($id, $resource){** *//a forrásból a szerelőkből lekérdezzük rekordokat*

**$records = TableRegistry::get($resource);**

**$query = $records->find()->where(['id' => $id]); *//*** *itt megkeressük az rekorodok között azt az elemet, ami egyenlő azzal id-vel amit az első sornál megkaptunk, ez az eredmény bekerül a query változóba,*

**return $query->first()->name;** //majd itt a queryből az első elemet fogjuk lekérdezni

}

}

Kicsit sírni fog a get miatt, mert depricated, és az 5.0-nál majd kivezetik, de még nekünk most jó lesz.

Lefuttatjuk, és lám, kiírja a fejlesztő nevét, következő lépésként ebből a névből egy linket készíthetünk. Tudom, ezt már nem akarjátok, nagyon nem, és Nektek már így tetszik, de kitartás! ☺

Ehhez az időpontok listáját megnyithatjuk egy új lapon, <http://localhost/hackerek/appointments>

A date-et itt is javíthatjuk: i18nFormat('yyyy-MMM-dd HH:mm:ss') mind az index és view-ban is.

és itt látható lesz mi a célunk, így szeretnénk megjeleníteni linként, a fejlesztőt, innen ki is másolhatjuk a kódot, amit itt találsz:

templates – appointments – index.php a 28-ik sorból kimásoljuk

$this->Html->link*($appointment->mechanic->name*, ['controller' => 'Developers', 'action' => 'view', $appointment->developer->id])

*viszed tovább:*

costumer – view ahová az imént beillesztettük a html helpert,amivel kiírattuk a nevet az elé bemásolhatnánk és a név helyére beilleszthetnénk a helperünket: (66-ik sorban)

<td><?= $this->Html->link($this->Format->getName($appointments->developer\_id, 'developers') , ['controller' => 'Developers', 'action' => 'view', $appointments->developer\_id, 'developers'])?></td>

Ha frissítünk, akkor látható, hogy a fejlesztő neve egy link lett, ami az adatlapjára vezet.

Ha már a fejlesztő lapjára érkeztünk, alul szintén nem az ügyfelet látjuk, hanem az ügyfél azonosítóját. Ezt majdnem ugyanúgy tudjuk megoldani, csak itt nem kell új helpert létrehozni, hanem ugyanazt a helpert tudjuk használni.

Amit szerkesztenünk kell az a developers view és az 59-ik sorba a szerelőnél elkészített iménti linket bemásoljuk, annyi, hogy ahol a developers van át kell írnunk customers-re. Vagy másold be innen:

<td><?= $this->Html->link($this->Format->getName($appointments->customer\_id, 'customers') , ['controller' => 'customers', 'action' => 'view', $appointments->customer\_id, 'customers'])?></td>

Ha frissítünk, megjelenik az ügyfél neve kattintható linkként, és az ügyfél oldalára fog vezetni, érdemes a fejlécből is kivenni az Id a Costumer után, ha már eleve a név jelenik meg linkként.

# Nyitó lap átalakítása

Ha a kezdő lapot megnyitjuk <http://localhost/autoszerviz/> a cakePHP alapértelmezett nyitó lapja jelenik meg. Innen nem tudunk sehová sem ugrani, csak ha a linkhez kézzel beírjuk, customers és akkor jelenik meg a lista.

**hackers – config – routs.php**

Az 53-ik sorban meg van adva, hogy

$builder->connect('/', ['controller' => 'Pages', 'action' => 'display', 'home']);

a controller a pages, vagyis nem az ügyfelek, nem a fejlesztők és nem az időpontok, az oldalakon belül az action a display, itt viszont egy harmadik oldalt a home page-t jeleníti meg.

Ez fölötte a megjegyzésben is látható, hogy ezt honnan veszi, templates, pages és azon belül home.php ha megnyitjuk ez a nyitólap tartalma, amit itt látunk. Itt megteheted, hogy ezt az oldalt módosítod, vagy azt is, hogy egy teljesen más fájlt állítasz be nyitólapnak.

Módosítsuk a nyitólapot a home.php-t ami a template/pages mappán belül van.

<a href="../hackerek/Customers"><?= $this->Html->**image('hackerek.PNG')?></**a>

A bodyban az image nevű függvényt fogjuk meghívni, és azt a webroot mappán belül az img mappából fogja venni. Ide kell bemásolnunk ezt a hackerek.PNG-t.

Ha php kódot kitöröljük felül, akkor még a menü is megjelenik. A layout default.php-ben tudjuk módosítani a szöveget a menü előtt.

Írjuk át a szöveget is, itt találtam tudós emojit: <https://emojipedia.org/technologist/>

hackerek – config – routs.php kikommentezhetjük a nyitólap útvonalát és megadhatjuk az ügyfeleket így is így egyből az ügyfelek listájához visz, de az előző verzió jobb.

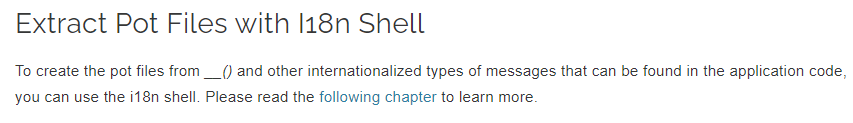
$builder->connect('/', ['controller' => 'Customers', 'action' => 'index']);

# Magyarítás

Vannak a cakePHP-ben olyan szövegek, amik angolul vannak, de mindent tudunk magyarítani.

<https://book.cakephp.org/4/en/core-libraries/internationalization-and-localization.html>

Itt olvashatjuk, hogyan csomagoljuk a neveket, ha nemzetközivé szeretnék alakítani \_\_()

C:\xampp\htdocs\hackerek\templates\layout\default.php itt megnézhetjük hogyan formáztuk a neveket a fejlécben; a dokumentációban olvashatjuk, melyik mappában hol kell létrehozni. PO fájlt kell létrehozni, amire kézzel macerás lenne, de van rá automatizált lehetőség. Kicsit lejjebb görgetve olvashatjuk, hogy ki tudunk vonni, el tudunk készíteni automatikusan fájlokat, amelyek tartalmazzák jó esetben az összes szöveget, ami a weblapunkon megjelenthet.

a following chapterben megtaláljuk milyen parancsot kell kiadnunk a projectünk mappáján belül

a dokumentumokban ha a címsorba ütjük, hogy cmd amegfelelő helyre jutunk ahol csak el kell indítani ezt a parancsolt,

C:\xampp\htdocs\hackerek - cmd

**bin\cake i18n extract** *fontos a perjelet megfordítani windows alatt*

utána kérdez, entert, entert, entert, és yest kell ütni mire szépen, kicsomagol mindent, amit kell.

A resource mappában elkészült egy locale mappa, amiben két pot kiterjesztésű fájl került, itt ha meg akarjuk nézni akkor egy gettext extensiont nem árt hozzáadni a vsc-hez.

Ha megnézzük mit kaptunk a default.pot-ban vannak msgid nevű angol szövegek, amik szerepelnek a weblapon, és alatta pedig msgstr ahol idézőjeleken belül adhatjuk meg a fordítást.

A pot fájlokat át kell nevezni po-ra és készíts egy hu mappát, abba belerakni.

Itt még nincs vége, egyet lépjünk vissza a dokumentáció oldalán, és ott meg van adva hol kell még a nyelvet beállítanunk. A config mappa app.php-jén hu-ra.

A fordítással el lehet bíbelődni, ezt kapjátok készen, ha még plusz szeretnél hozzá adni fordításokat, a fájlban látható minta szerint folytathatod szépen a sort, vagy fogod és a szerverről, a feladat mappából, (php/cakePHP) –ből kimásolod a tiéd helyére.

Amit nem fordít át, ez egy jó játék lesz megkeresni a fájlok között a forrást, és a benne lévő szavakat becsomagolni a \_ \_() –be. Frissítsd az oldalt, és kész a magyarítás.

# Beléptető rendszer

Jó lenne, ha kívül állók azért nem tudnának hozzáférni ezekhez az oldalakhoz.

Az adminokhoz adjuk hozzá Macska Jancsit, [jancsi@gmail.com](mailto:jancsi@gmail.com) juliska jelszóval. Viszont, ha most megnézzük az adatbázisban, szépen látszik a beírt jelszó. Ez az, amiről már beszéltünk, hogy ez így nem túl biztonságos, így ezt itt nem szeretnénk megjeleníteni.

src -modell - entity admin.php

és a végére az osztályon belül felveszünk egy új függvényt, vigyázva, hogy az osztályon{} belülre kerüljön

**protected function \_setPassword($password) {**

**return (new DefaultPasswordHasher)->hash($password);**

**}**

ez csak akkor fog működni, ha a fájl elején megadjuk:

**use Cake\Auth\DefaultPasswordHasher**;

különben hibát ír.

Ha töröljük, és újra felvisszük akkor az adatbázisba az admint, akkor már a titkosított jelszó jelenik meg. Ha ráfrissítesz már láthatod is.

Adjunk hozzá új admint ugyanazzal a jelszóval Macska Julis, [julis@gmail.com](mailto:julis@gmail.com) juliska jelszóval, Nézzük meg az adatbázisban a jelszavakat. Láthatjuk, hogy habár ugyanazt adtuk két teljesen különböző kódot generált, tehát ha valaki megfejti az egyiket, azt hiszi a másik más.

A jelszó biztonsági okokból már kikerült a kódból a weblapon, nem kell a törlésével bajlódnunk.

# Beléptető űrlap

tempaltes –admins és itt egy új fájlt kell létrehoznunk login.php néven.

ezt máshova kéri az új verzió templates mappán belül kér egy login mappát és azon belül egy index.php oldal létrehozását, de én simán létrehoztam a templatesen belül egy login.php-t

az osztályok már nem kerek,mert nem a régi foundation tampletet használja a cakePHP

<div class="index large-4 medium-4 large-offset-4 medium-offset-4 columns content">

<div class="panel">

<h3 class="text-center"><?= \_\_("Admin Login")?></h3>

<?= $this->Form->create(); ?>

<?= $this->Form->input('email'); ?>

<?= $this->Form->input('password', array('type'=>'password')); ?>

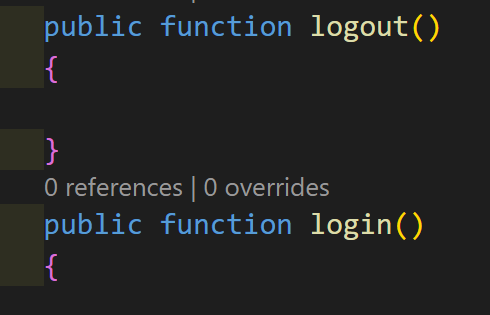
<?= $this->Form->submit(\_\_('Login'), array('class'=>'button')); ?>

<?= $this->Form->end(); ?>

</div>

</div>

Az űrlapon a form nevű helpert használjuk, ha megpróbáljuk megnyitni hibát ad, hogy az adminscontroller hiányzik. <http://localhost/hackerek/admins/login>

ugyanis az AdminsController.php fájlban nem hoztuk még létre a login metódust. A korábban felvett logout fölé be is illeszthetjük. Újra megnézed meg is jelenik a bejelentkezési űrlap, már magyarra fordítva, mivel a po fájlba már beírásra kerültek ezek is.

src\Controller\TemplatesController.php

**templates – layout default a logint ki kell egészíteni**

<?= $this->Html->link(\_\_('Logout')." (".$belepve.")",['controller' => 'admins', 'action' => 'logout']); ?>

src-controller-AdminsControllert.php kiegészítése

public function login(){

if ($this->request->is('post')) { //itt megvizsgáljuk kaptunk e az űrlapról post tipusú kérést

$user = $this->Auth->identify(); // ha igen akkor meghívjuk az identity metódust ami ellenőrzi a megadott adatokat

if ($user) { //siker esetén visszaadja az user objektumot

$this->Auth->setUser($user); //amit ha visszakapunk beállítjuk felhasználót

$this->redirect($this->Auth->redirectUrl()); //és átirányítjuk a böngészőt a megfelelő oldalra

}

else {

$this->Flash->error(\_\_('Incorrect login')); //ezt is a nyelvi fájlban lekell fordítani

}

}

}

**Hibát fog adni, be** kell lépnünk, az SRC –Controller – AppController.php fájlba és a 44-ik sorban, ahol a különböző komponenseket töltjük be, és be kell töltenünk az auth komponenst, ami majd a hitelesítést fogja elvégezni.

*46-ik sorba folytatólagosan írd hozzá:*

$this->loadComponent('Auth', [

'authenticate' => [

'Form' => [

'userModel' => 'Admins',

'fields' => [

'username' => 'email',

'password' => 'password'

]

]

],

'loginAction' => [

'controller' => 'Admins',

'action' => 'login'

]

]);

Fontos felvenni a 'userModel' => 'Admins', sort, mivel nem user-ünk van hanem admins, vagyis nem users-táblát kell keresnie alapértelmezetten, hanem az admins táblát az adatbázisban.

Tehát betöltjük az Auth komopnenst, amelynek az első paramétere egy tömb lesz, és azon belül további tömbök vannak, az első adat egy 'authenticate' => [ és ezen belül fognak majd megjelenni a bejelentkezési adatok, megadjuk a mezőket, email, password, általában ez username, de mi az emailt használjuk most. Utána megadjuk, hogy mi történjen, ha bejelentkezik a felhasználó, megadjuk a controllert, ami az admins, az action pedig a login metódus.

Próbáljuk ki rossz, majd jó jelszóval is. Ha nincs hiba üzenet a nyitó lapra kerülünk. Ha minden jól ment, akkor mostmár sikeresen működik a bejelentkezés.

# Kilépés

Viszont még kilépést is meg kell oldanunk, hiszen ha a kilépésre kattintunk, a logout függvényünk nem működik.

Nyissuk meg az src-controller-AdminsController.php fájlunkat és vegyük fel a logut függvény tartalmát:

public function logout() {

$this->Flash->success(\_\_('You are logged out'));

$this->redirect($this->Auth->logout());

}

Amit ez okoz, az az, hogy megjelenít egy üzenetet, hogy kiléptél, a második sora pedig meghívja az auth logout függvényét, mi kijelentkezteti a felhasználót az alkalmazásból.

Teszteljük, ki tudunk e lépni, ha sikerült, akkor szuper! ☺

# Linkek:

MVC definíciója:https://hu.wikipedia.org/wiki/Modell-n%C3%A9zet-vez%C3%A9rl%C5%91

CakePHP weblapja: <https://cakephp.org/>

Composer weblapja, letöltés: <https://getcomposer.org/download/>

# Számonkérés

A közösen megoldott feladat alapján építs fel egy új weboldalt a saját ötletetek alapján. Átírhatod az adatbázist a saját ötletre, és aszerint valósítsd meg a lépéseket. Érdemes valamilyen szolgáltató vállalatban gondolkodni. A feladathoz dokumentációt is kell készítened (felhasználhatod az én jegyzetemet is az elkészítéséhez, de lefordíthatod magyarra a Cakephp oldalán lévő dokumentációkból felhasznált lépéseket is).

A feladat lényege, az önálló tevékenységre való felkészítés, mennyire tudod adaptálni egy meglévő feladat megoldását a saját elképzelésed megvalósításához, valamint az, hogy ha egy olyan új feladatot kapsz, amit még nem próbáltál addig, ne ijedj meg tőle, hanem bátran vágj bele, hiszen az interneten mindenre is találsz megoldást!