

TeaChat – Fejlesztői dokumentáció

Készítette:

Bodnár Dávid

Jokán Gergő

Fekete Máté

Budapest, 2022.

Tartalomjegyzék

I. A program feladata	4
1.1. Bemenet:	4
1.2. Kimenet:	4
1.3. Előfeltétel:	4
II. A futási környezet	5
2.1. Program futtatása:	5
2.2. A szerver adatai:	5
III. Fejlesztői környezet leírása	6
3.1. Fejlesztői környezet:	6
3.2. Választott programnyelvek, kiegészítők és verziószámai;	6
IV. Az algoritmusok és az adatok leírása, ezek kapcsolata	7
4.1. Regisztráció, bejelentkezés:	7
4.2. Elfelejtett jelszó kérése:	7
4.3. „Emlékezz rám”:	7
4.4. Munkamenet kezelés:	7
4.5. Kezdőképernyő:	8
4.6. Új poszt:	8
4.7. Hozzászólások listája:	9
4.8. Hozzászólás szerkesztése:	10
4.9. Hozzászólás lájkolása:	10
4.10. Kommentelés:	10
4.11. Komment lájkolása:	11
4.12. Profil:	11
4.13. Keresés:	11
4.13.1. Felhasználó keresése:	11
4.13.2. Hozzászólás keresése:	11
4.14. Készítők:	12
4.15. Kijelentkezés:	13
V. Tiszta kód, implementációs szabványok, döntések;	14
5.1 Tiszta kód:	14
5.2 Implementációs szabványok, döntések	14
VI. Tesztesetek	15

6.1. Feature tesztek:	15
6.2. Unit tesztek:	15
6.3. PieceOfUsers:	17
VII. Felhasznált tartalom	19
VIII. Fejlesztési lehetőségek	20

I. A program feladata

A program lehetőséget nyújt tanároknak egy közös fórumon beszélgetni, megvitatni különböző témákat. Segítséget kérni, illetve másoknak segíteni, vagy érdeklődési körének megfelelő témákat olvasni, azokhoz hozzászólni.

1.1. Bemenet:

A felhasználó megírja hozzászólását, amelyet utána kommentelni, kedvelni lehet. A bevitt adatok adatbázisban kerülnek letárolásra. További adatok kerülnek naplózásra az adatbázisba, mint például a hozzászólás időpontja, illetve a felhasználó megadhat további vele kapcsolatos adatokat (intézmény neve ahol tanít, vagy oktatott tantárgy) melyek szintén letárolásra kerülnek.

1.2. Kimenet:

A hozzászólások és a kommentek szövegei megjelennek a weboldalon. A felhasználóhoz köthető adatok pedig a profil oldalon jelennek meg.

1.3. Előfeltétel:

A program használatához regisztráció és bejelentkezés szükséges.

II. A futási környezet

2.1. Program futtatása:

A program futtatása webböngészőben történik. A <http://remek3.elg-bp.edu.hu> címen érhető el. Az alkalmazás optimalizálva van minden népszerű böngészőre (pl.: Google Chrome, Mozilla Firefox, Opera, Microsoft Edge). Az oldal reszponzív. Megjeleníthető számítógépen, tableten, mobiltelefonon.

2.2. A szerver adatai:

A fájlok a BGéSZC Eötvös Loránd Technikum szerverére kerültek feltöltésre.

- A szerveren Debian Linux 11. operációs rendszer fut.
- A kernel verziószáma: 5.10.0.
- Szerver API: Apache 2.0.
- Apache verziószám: 2.4.52.
- mySql verziószám: 8.1.1.

III. Fejlesztői környezet leírása

3.1. Fejlesztői környezet:

A projekt fejlesztését a Visual Studio Code (v1.63.2.) programban végeztük. Az adatbázis tervezése MySql típusú adatbázisban történt. A Xampp Control Panel (v3.3.0.) segítségével phpMyadmin felületen. A megtervezett adatbázis alapján a Visual Studio Code alkalmazásban hajtottuk végre a migrációt. Ez után tesztadatokkal töltöttük fel az adatbázis tábláit. A fájlok szerverre való feltöltéséhez a WinSCP (v5.19.5.) alkalmazást használtuk. A projekt a BGéSZC Eötvös Loránd Technikum szerverére került feltöltésre.

3.2. Választott programnyelvek, kiegészítők és verziószámai;

- Html 5;
- Laravel v8.79.0 (PHP v8.1.1);
- Laravel Breeze starter kit;
- tailwindcss v3.0.13;
- jQuery 3.4.1;
- AlpineJs 3.4.2.

IV. Az algoritmusok és az adatok leírása, ezek kapcsolata

4.1. Regisztráció, bejelentkezés:

Az oldal használata bejelentkezéshez és regisztrációhoz kötött. Bejelentkezés nélkül a tartalom nem látható. Ennek háttérében az áll, hogy a program egy bizonyos szűkebb felhasználó csoportnak, a tanároknak készült. A regisztráció ennek ellenére nem igényel adminisztrátori engedélyt. A regisztráció és bejelentkezés funkciók elérhetőek a kezdőoldaltól, vagy a <http://remek3.elg-bp.edu.hu/register>, illetve a <http://remek3.elg-bp.edu.hu/login> oldalakon. Továbbá mindkét funkcióra navigálni lehet egymásról.

4.2. Elfelejtett jelszó kérése:

A regisztrációhoz szorosan kapcsolódó funkció. Amennyiben a felhasználó elfelejti a jelszavát, akkor az e-mail címének megadását követően egy regisztrációs linket küld az oldal, amelyre ha rákattint, akkor új jelszót adhat meg magának. Ez a funkció a <http://remek3.elg-bp.edu.hu/forgot-password> oldalon található meg, továbbá a bejelentkezés oldalról lehet ide navigálni.

A laravel breeze beépített forgot-password funkcióját használtuk. Létrehoztunk egy teachat.remek3@gmail.com fiókot, amelyről a jelszó visszaállításához szükséges e-mail üzenet érkezik a felhasználónak. A gmail smtp beállításait használtuk fel az üzenet küldéséhez.

4.3. „Emlékezz rám”:

Bejelentkezést megelőzően a felhasználó megadhatja, hogy az oldal megjegyezze-e a bejelentkezési adatait az adott eszközön történő további látogatásaira. Ezen funkció a kijelentkezéssel erejét veszti. Elérhető a bejelentkezési adatok megadása alatt egy checkbox-ban.

4.4. Munkamenet kezelés:

Az információknak a szerveren történő ideiglenes tárolására session-t használunk. Azért tároljuk ideiglenesen (kijelentkezésig) ezeket az adatokat, hogy az oldalak közötti navigáció során ne kelljen újra és újra bejelentkeznie a felhasználónak. A cookie-val ellentétben ezek az adatok a webszerveren kerülnek eltárolásra, nem pedig lokálisan a felhasználónál.

4.5. Kezdőképernyő:

Bejelentkezés után a felhasználó a kezdőképernyőre jut, ahol egy üdvözlő szöveg fogadja. Ahogy a fentiekben már kifejtésre került a kezdőképernyő és a további oldalak, funkciók már csak a bejelentkezés után érhetőek el. A kezdőképernyőn lehetséges például új hozzászólást létrehozni.

4.6. Új poszt:

A gomb lenyomását követően a <http://remek3.elg-bp.edu.hu/newpost> oldalra jutunk. Itt lehetséges új hozzászólást írni. A hozzászólás címe maximum 150, a hozzászólás szövege maximum 400 karakter lehet. A maximális méretet két helyen is lekezeljük. Egyrészt a html kódban az input tagnek maxlength értéket adtunk meg, másrészt a createPost.php controllerben a validálásnál is megadtuk a maximum értéket. A tartalom maximális hosszán kívül további kritériumokat adtunk meg a validálás során. Validálásoknál minden esetben magyar hiba üzenetet adtunk meg.

Az új poszt létrehozása során a felhasználó meg tudja adni a tartalomhoz kapcsolódó évfolyamat és tantárgyat is, valamint fájlt is fel tud tölteni. A küldés gombra kattintva, a tartalom rögzítésre került az adatbázisban és megjelenik az oldalon. A felhasználó átirányításra kerül a kezdőképernyőre.

Az új hozzászólás teljes validálása, és a hozzá tartozó adatbázis lekérdezés:

```
public function create(Request $req){
    $req->validate([
        "title" => "required|min:5|max:150",
        "post" => "required|min:5|max:400",
        "class" => "required",
        "subject" => "required"],

        ["title.required" => "A mező kitöltése kötelező! Minimum 5
        karakter, maximum 150 karakter.",
        "title.min" => "Minimum 5 karakter!",
        "title.max" => "Maximum 150 karakter!",
        "post.required" => "A mező kitöltése kötelező! Minimum 5 karakter,
        maximum 400 karakter.",
        "post.min" => "Minimum 5 karakter!",
```



```

        "post.max" => "Maximum 400 karakter!",
        "class.required" => "Kötelező választani!",
        "subject.required" => "Kötelező választani!"
    ]);

    $title=strip_tags($req->title);
    $post=strip_tags($req->post);
    $class=strip_tags($req->class);
    $subject=strip_tags($req->subject);

    $file=$req->file("inputfile");
    if(gettype($file) != "NULL"){
        $filename=Storage::putFile("", $file);
        $fileID=DB::table("files")->insertGetId(["PATH" => $filename]);
    }
    else{
        $fileID=1;
    }

    $user = Auth::user();

    DB::insert("INSERT INTO `post`(`user_ID`, `text`, `title`, `class`, `file_ID`,
        subject_ID`, created_at`)
VALUES (?,?,?,?,?,?)",
    [$user->id,$post,$title,$class,$fileID,$subject, date("Y-m-d H:i:s")]);

    return Redirect('/dashboard');
}

```

4.7. Hozzászólások listája:

A főoldalon találhatóak a hozzászólások. Jól elkülönítve egymás alatt helyezkednek el. Egy sorban A létrehozó avatarjának képe, a létrehozó személy neve, a poszt címe, a rá adott kommentek szám, a lájkjainak száma és a létrehozás dátuma szerepel. Továbbá minden sor

végén elhelyezésre került egy „Poszt megnyitása” gomb, amely <http://remek3.elg-bp.edu.hu/postview/>... oldalra navigál át.

A hozzászólások listájának létrehozásánál paginationt alkalmaztunk, a könnyebb átláthatóság érdekében. A hozzászólásokat foreach ciklussal listázzuk.

4.8. Hozzászólás szerkesztése:

A hozzászólásokat lehetséges szerkeszteni. A szerkesztés gomb csak annak a felhasználónak jelenik meg akitől az eredeti poszt származik.

4.9. Hozzászólás lájkolása:

A hozzászólásokat lehetséges lájkolni, minden felhasználó csak egyszer lájkolhat egy hozzászólást. Amennyiben már korábban lájkoltunk egy adott hozzászólást, a gombra történő ismételt kattintás azt eredményezi, hogy a kedvelésünket eltávolítjuk a posztról.

4.10. Kommentelés:

A hozzászólásokat kommentálhatják a felhasználók. A hozzászólás megnyitását követően láthatóvá válik a hozzászólás teljes szövege, alatta pedig kommentek láthatóak, valamint az új komment létrehozásának lehetősége.

A validálás és az adatbázisban elhelyezés a comment.php controllerben történik az alábbiak szerint:

```
public function create($ID, Request $req){

    $req->validate([
        "comment" => "min:3|max:400"
    ],
    ["comment.min" => "Minimum 3 karakter!",
    "comment.max" => "Maximum 400 karakter!"
    ]);

    $comment=strip_tags($req->comment);

    $user = Auth::user();

    DB::insert("INSERT INTO `comment`(`post_ID`,`user_ID`,`text`,`created_at`)
    VALUES (?,?,?,?)", [$ID,$user->id,$comment, date("Y-m-d H:i:s")]);

    return Redirect("/postview/$ID");

}
```

A komment hossza html tag-el is maximalizálásra került.

4.11. Komment lájkolása:

A kommenteket is lehetséges kedvelni, ugyan azon a metodika mentén, mint a hozzászólások esetében.

4.12. Profil:

A profil oldalon a felhasználó szerkesztheti saját adatait, új információkat adhat meg magáról. A regisztráció során nevet, e-mail címet kell megadnia a felhasználónak, továbbá a bejelentkezéshez szükséges jelszavát. A saját profil a <http://remek3.elg-bp.edu.hu/profil> oldalon érhető el. A felhasználónak itt lehetősége nyílik megadni, hogy hányadik osztálytól hányadik osztályig tanít. Megadhatja, hogy melyik megyében és melyik oktatási intézményben tanít, valamint profilképet tölthet fel magának, amely az oldalon több helyen is megjelenik a neve mellett. Amennyiben nem választ a felhasználó magának profilképet, akkor az alapértelmezett kép marad használatban.

4.13. Keresés:

Az oldalon lehetőség van a felhasználók és a posztok közötti keresésre. A keresés oldal a <http://remek3.elg-bp.edu.hu/search> címen érhető el. A találatok megjelenítése ugyan ezen az oldalon található. A backend műveletek a search.php oldalon kerültek definiálásra.

4.13.1. Felhasználó keresése:

Felhasználóra a neve, e-mail címe vagy megye adatok alapján lehetséges keresést végrehajtani

4.13.2. Hozzászólás keresése:

A hozzászólásokra is három módon lehet keresést végrehajtani. Cím alapján, létrehozás dátuma alapján és tantárgy adatok alapján tud a felhasználó keresni.

A dátum alapján végrehajtott keresésnél a search.blade.php fájlban jquery datepicker-t használtunk a dátum megadására.

```
<script>
$(function() {
    $('#dateFrom').datepicker({
        dateFormat: 'yy-mm-dd',
        changeMonth: true,
        changeYear: true,
```

```

onSelect: function(datetext){

    datetext=datetext+" 00:00:00";
    $('#dateFrom').val(datetext);
    },
});

$('#dateTo').datepicker({
    dateFormat: 'yy-mm-dd',
    changeMonth: true,
    changeYear: true,
    onSelect: function(datetext){

        datetext=datetext+" 00:00:00";
        $('#dateTo').val(datetext);
        },
    });
});
</script>

```

A felhasználó kényelmesen dátum választón adhatja meg a keresési feltételeket. Kettő, a mellékelt ábrán látható dátum választón kell kiválasztani a kezdő- és vég dátumot, és ez után lehet a keresést elvégezni. Évet, hónapot és napot lehetséges választani.

A jquery kód a kiválasztott évhez, hónaphoz és napot hozzáad 00 óra, 00 perc és 00 másodpercet, hogy igazodjon a kiválasztott dátum az adatbázisban tárolt

dátum formátummal, ami az alábbi: yyyy-mm-dd hh:ii:ss.

4.14. Készítők:

A készítők oldalon található egy-egy rövid bemutatkozás a készítőkről. A <http://remek3.elg-bp.edu.hu/search> oldalon érhető el.

4.15. Kijelentkezés:

Ahogy a korábbiakban már kifejtésre került a kijelentkezés után az emlékezz rám funkció hatályát veszti. Ezen túl a kijelentkezéssel a felhasználó biztosíthatja azt, hogy a felügyelet nélkül hagyott eszközén illetéktelen ne használhassa a profilját.

V. Tiszta kód, implementációs szabványok, döntések;

5.1 Tiszta kód:

Az oldalon magyar nyelven jelenítjük meg a tartalmakat. A programozás viszont angol nyelven történik. Angol elnevezést kaptak a függvények, változók, oldalak stb.

A programban egészében a függvények és változók nevét a camelCase szabály szerint alkottuk meg. Kivételt képeznek ez alól a tesztek, ezek esetében a snake_case elvet alkalmaztuk.

5.2 Implementációs szabványok, döntések

Az adatbázisból kinyert adatokat, a tartalmak megjelenítésénél minden esetben foreach ciklusokat használtunk.

A welcome oldalt leszámítva minden oldal alján található egy vissza gomb, amely az előzmények alapján az előző oldalra navigál:

```
<script>
  function goBack() {
    window.history.back()
  }
</script>
```

A gomb jquery history back metódus alapján működik.

Az input adatokból strip_tags() függvénnyel eltávolítjuk a HTML, XML, és PHP címkéket.

VI. Tesztesetek

A projekt tesztelése során a breeze starter kit beépített tesztjein túl feature, és unit teszteket is lefuttattunk.

6.1. Feature tesztek:

A feature tesztek egyes funkciók és funkciók láncolatának működését ellenőrzésére szolgál.

```
public function test_login_page()
{
    $response = $this->get('/login');

    $response->assertStatus(200);
}

public function test_register_page()
{
    $response = $this->get('/register');

    $response->assertStatus(200);
}
```

A fenti két teszt ellenőrzi a login és a register oldalak elérhetőségét. Amennyiben 200-as számú választ kapunk vissza a szervertől, akkor az oldal elérhető.

6.2. Unit tesztek:

A unit tesztek a kód egy-egy speciális funkciójának ellenőrzésére valók. Itt ismételten ellenőrzésre kerültek a login és register oldalak, valamint megnéztük, hogy lehetséges-e azonos e-mail címmel regisztrálni az oldalra. Azonos e-mail esetén hibára futott a teszt. Ezt követően kicseréltük a függvényben az egyik e-mail címet, hogy ne fusson hibára a teszt.

```
public function test_login_form(){
    $response = $this->get('/login');
    $response->assertStatus(200);
}
```

```

public function test_register_form(){
    $response = $this->get('/register');
    $response->assertStatus(200);
}

public function test_user_duplication(){
    $user1 = User::make([
        'name' => 'Teszt Elek',
        'email' => 'tesztelek@gmail.com'
    ]);

    $user2 = User::make([
        'name' => 'Proba Maria',
        'email' => 'probamaria@gmail.com'
    ]);

    $this->assertNotEquals($user1->email , $user2->email);
}

```

Unit tesztelések között a school táblát és seedereket is teszteltük, az alábbiak szerint:

```

public function test_database(){
    $this->assertDatabaseHas('school', [
        'name' => "
    ]);
}

public function test_if_seeder_works(){
    $this->seed();
    $this->assertDatabaseHas('school', [
        'name' => 'BGÉSZC Eötvös SZKI'
    ]);
}

```


6.3. PieceOfUsers:

Bár nem nevezhetjük klasszikus tesztelésnek, de a getUser.php nevű controllerben az alábbi tesztelést/ellenőrzést hajtottuk végre:

```
public function pieceOfUsers(Request $req){

    $piece = DB::select("SELECT COUNT(`id`) AS piece FROM users;");

    if(count($piece)==0){
        $jsonData['error'] = true;
        $jsonData['errorText'] = "Nincs regisztrált felhasználónk!";
    }else{
        $jsonData['error'] = false;
        $jsonData['pieceOfUsers'] = $piece[0];
    }
    return response()->json($jsonData);
    return view('authors',["piece"=>$piece]);

}
```

Egyszerű adatbázis lekérdezéssel megállapítja a függvény, hogy a users táblában hány darab id található, magyarul, hány regisztrált felhasználója van az oldalnak. Json adattal tér vissza a függvény. Amennyiben nincs regisztrált felhasználója az oldalnak akkor az error mező igaz állítással tér vissza és a „Nincs regisztrált felhasználónk!” szöveget kapjuk válaszként. Ebben az esetben arra következtethetünk, hogy valamilyen hiba folytán üres a users tábla. Hibakeresésen túl az is gyorsan ellenőrizhető, hogy hány felhasználója van éppen az oldalnak. Amennyiben a Json adat error mezője false, azaz hamis adattal tér vissza, akkor található adat a users táblában, és kiírja a sorok számát a .../api/pieceofusers oldalon.

Az adatok megjelenítésére a www.postman.com weboldalt használjuk:

remek3.elg-bp.edu.hu/api/pieceofusers

Save

</>

GET

remek3.elg-bp.edu.hu/api/pieceofusers

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

Cookies (2)

Headers (10)

Test Results

Status: 200 OK

Time: 475 ms

Size: 1.18 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "error": false,
3    "pieceOfUsers": {
4      "piece": 12
5    }
6  }

```

VII. Felhasznált tartalom

Az oldal gerincét a Laravel v8.79.0 és a Laravel Breeze starter kit adják. A starter kit tartalmazza a regisztráció, bejelentkezés, elfelejtett jelszó, emlékezz rám, funkciókat, illetve további tartalmakat. A kész terméket saját weboldalunkhoz igazítottuk és magyar nyelvre fordítottuk.

Az ikonokat a <https://heroicons.com/> weboldalról töltöttük le.

VIII. Fejlesztési lehetőségek

Fejlesztési lehetőségként egy admin felületet képzeltünk el, amelyről elérhetővé válik az adminisztrátorok számára a hozzászólások moderálása, törlése, valamint felhasználói tevékenység korlátozása, végső esetben felhasználó törlése. Ezekre a műveletekre a phpMyadmin felületen jelenleg is van lehetőség, azonban a fejlesztés után közvetlenül az oldalról is elérhetőek lennének ezek a funkciók.

Bár az oldal reszponzív, de további fejlesztési lehetőségként felmerült egy mobil applikáció készítése is.

A jövőre nézve oldalon belüli, push üzeneteket is elképzeltünk a programunkban.