

# Vizsgaremek

**Gyulai SzC Szigeti Endre Technikum és Szakképző Iskola**

Lele Regina, Kovács Orsolya

2025

# Tartalomjegyzék

<b>Tartalomjegyzék.....</b>	<b>2</b>
<b>Bevezetés.....</b>	<b>3</b>
<b>Felhasználói dokumentáció.....</b>	<b>3</b>
Program funkciói.....	3
Hardver követelmények.....	4
Telepítés.....	4
<b>Fejlesztői dokumentáció.....</b>	<b>11</b>
<b>Frontendről.....</b>	<b>11</b>
Reszponzivitás:.....	12
<b>Adatbázisunk.....</b>	<b>14</b>
Dish Tábla.....	14
Courses tábla.....	16
Allergens tábla.....	17
<b>Backend.....</b>	<b>19</b>
Dishcontroller.....	19
Index function.....	19
Create function.....	20
Store function.....	20
Show function.....	22
Edit function.....	22
Update function.....	23
Destroy function.....	24
<b>Befejezés.....</b>	<b>25</b>
Csapatmunka:.....	25
Jövőbeli Tervek a programmal kapcsolatban:.....	25
Irodalomjegyzék:.....	25
Összefoglalás.....	25

# Bevezetés

Programunk egy adatbázis körül épített weboldal, ahol lehetősége van felhasználóknak recepteket megnézni, feltölteni, szerkeszteni, és törölni.

Azért választottuk ezt a témát, mert főzés szívünkhöz nagyon közel áll, és ezen keresztül tudásunk minden prospektusát be bírjuk mutatni, mint az adatbázis tervezés és kezelés, javascript, css, stb...

Ezt a weboldalt azoknak szánjuk akik ugyan ennyire szeretik a főzést, és a recepteik rendbentartását.

## Felhasználói dokumentáció

### Program funkciói

- receptek feltöltése/törlése
- recept szerkesztése
- receptek böngészése

A felhasználó egy receptek tárolására és böngészésére alkalmas weboldalt lát maga előtt.

Ahhoz, hogy lehetővé tegye ennek a weboldalnak a megtekintését, a felhasználónak rendelkeznie kell egy eszközhöz ami lehet:

- windows alapú Pc
- windows alapú laptop

### Hardver követelmények

Processzor: 1,6 GHz vagy nagyobb

Memória: 4gb vagy több

OS: Windows 10 vagy Windows 11

Rendszer típus: 64 bites operációs rendszer

Ha a felhasználó eszköze hasonló, akkor lehetséges a weboldal használata.

## Telepítés

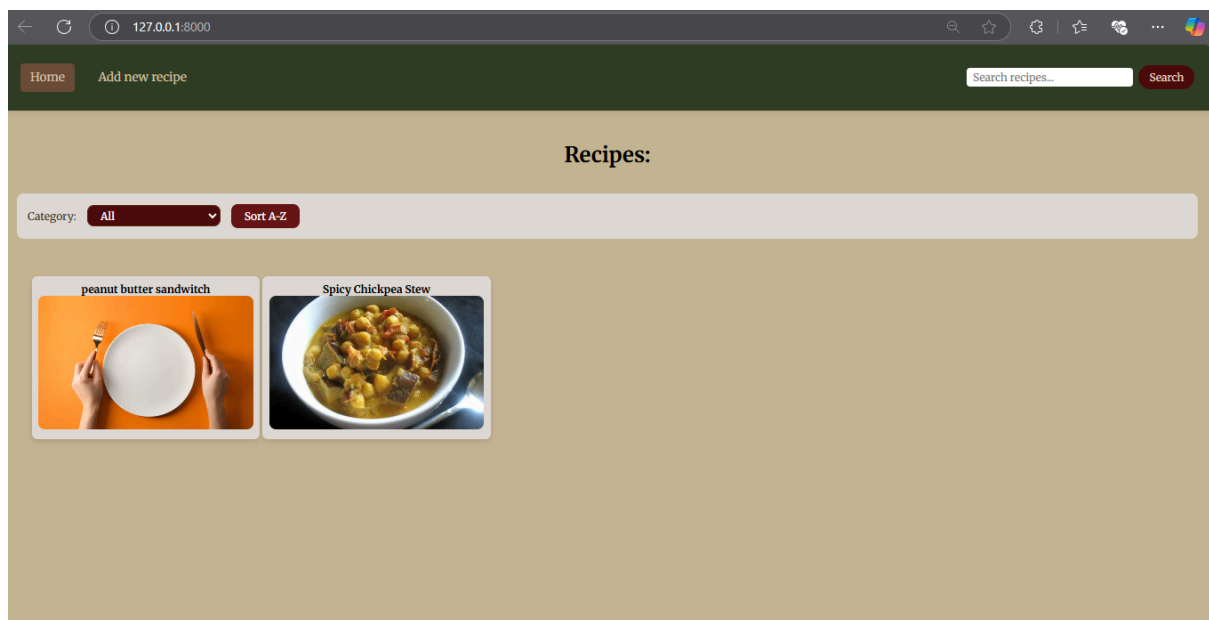
A telepítésre elsősorban szükséged lesz a visual studio code-ra.

Itt a Source controller-nél github repository klónozására van lehetőség ahol beillesztetted a github linket és lehúzod a programot.

Azon belül pedig szükséged lesz mysql-re, és xampp applikációra és artisan bővítményre.

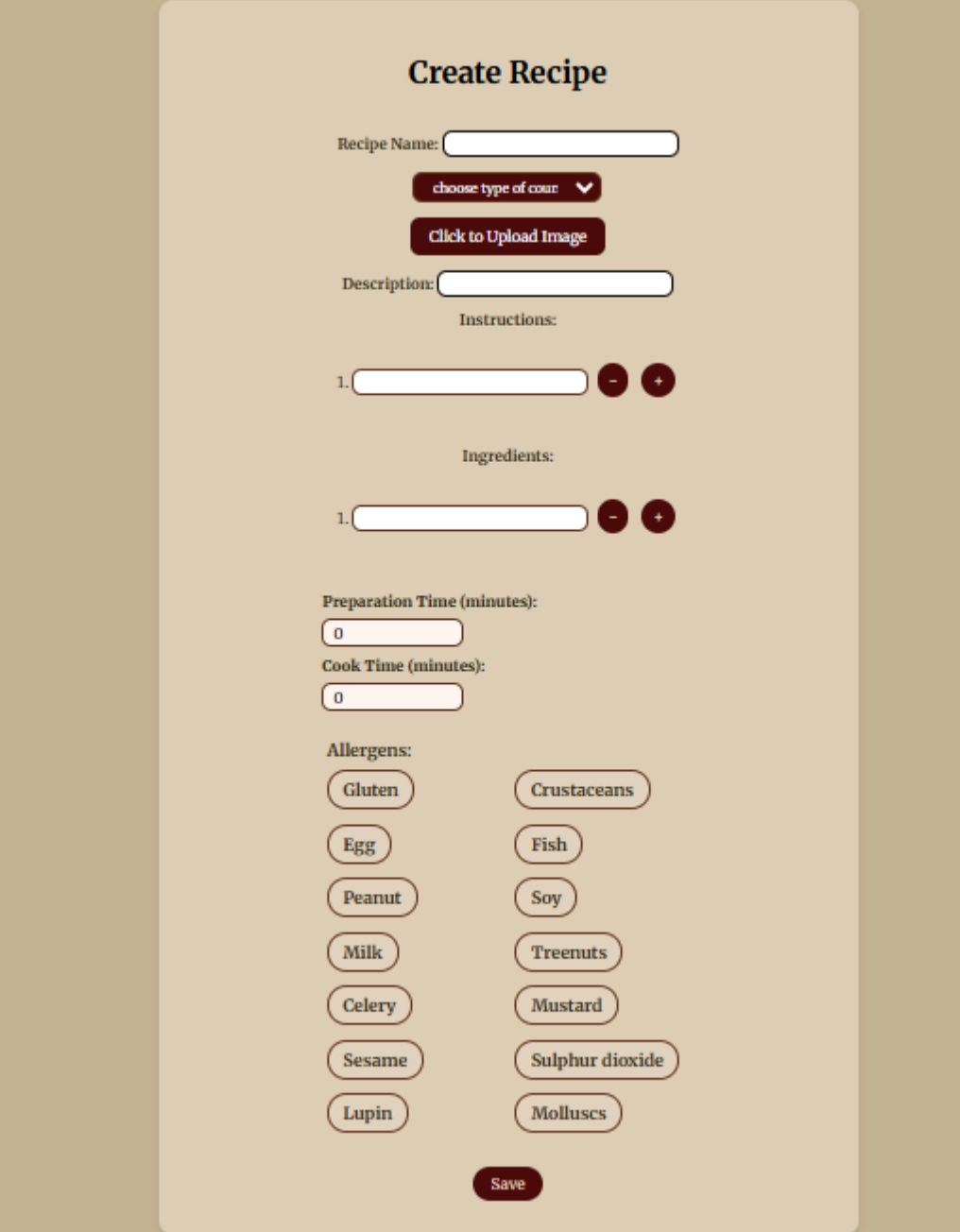
ezután lehet futtatni gördülékenyen a programunkat.

Ahogy az alkalmazás elindul, ez látható. Ez maga a főoldal.



Ugye van a fejlécben kettő menüpont, illetve egy keresés mező. Az egyik maga az otthon, ahol jelenleg tartunk. A másik pedig egy recept hozzáadása.

Hogy ha rákattintunk akkor ez az oldal jön be:

A 'Create Recipe' form on a light beige background. The form includes fields for 'Recipe Name', 'Description', and 'Instructions'. Below 'Instructions' is a numbered list (1.) with a text input and minus/plus buttons. Below 'Instructions' is an 'Ingredients' section with a numbered list (1.) and a text input with minus/plus buttons. There are input fields for 'Preparation Time (minutes)' and 'Cook Time (minutes)', both with a '0' value. An 'Allergens' section contains two columns of buttons for various allergens: Gluten, Crustaceans, Egg, Fish, Peanut, Soy, Milk, Tree nuts, Celery, Mustard, Sesame, Sulphur dioxide, Lupin, and Molluscs. A 'Save' button is at the bottom.

**Create Recipe**

Recipe Name:

choose type of course ▼

Click to Upload Image

Description:

Instructions:

1.  - +

Ingredients:

1.  - +

Preparation Time (minutes):

Cook Time (minutes):

Allergens:

Gluten Crustaceans

Egg Fish

Peanut Soy

Milk Tree nuts

Celery Mustard

Sesame Sulphur dioxide


Lupin Molluscs

Save

Itt részletesen meg lehet adni, hogy pontosan mi kell hozzá, hogyan kell megvalósítani, mennyi idő a főzés és előkészítés, emellett képet is lehet feltölteni, illetve kiválasztani, hogy milyen kategóriába esik a recept ami éppen feltöltésre kerül. Emellett van szekció arra, hogy kiválaszd a receptben lévő lehetséges allergéneket.

Egy recept feltöltés után átirányít a receptedhez, ami így fog megjelenni:

## Spicy Chickpea Stew



**Description:** A hearty and flavorful stew packed with protein-rich chickpeas, vibrant vegetables, and a kick of spice. Perfect for a cozy dinner or meal prep.

**Course:** soup

**Instructions:**

1. Stir in diced bell pepper and cook for 2-3 minutes.
2. Add chickpeas, diced tomatoes, vegetable broth, cumin, smoked paprika, chili powder, salt, and pepper.
3. Bring the mixture to a boil, then reduce heat and let it simmer for 15-20 minutes, stirring occasionally.
4. Taste and adjust seasoning as needed.
5. Serve hot, garnished with fresh cilantro if desired.

**Ingredients:**

- 1 tsp ground cumin
- 1 tsp smoked paprika
- 1/2 tsp chili powder
- 2 tbsp olive oil
- Salt and pepper to taste
- Fresh cilantro for garnish (optional)

**Preparation Time:** 10 minutes

**Cook Time:** 20 minutes

**Allergens:**

Egg Soy

[edit recipe](#) [delete recipe](#)

Minden megadott alap felsorolásra kerül.


A lap alján látható, hogy ebben a szekcióban lehetőség van szerkeszteni, vagy a receptet törölni.

Emellett, ha kép feltöltésére nem kerül sor, akkor egy placeholder kép jelenik meg, ami átveszi a jelenleg nem létező kép helyét, de csak addig míg egy másik kép át nem veszi a helyét.

[Home](#) [Add new recipe](#)

Search recipes... [Search](#)

### test 1



**Description:** desc

**Course:** pastry

**Instructions:**

1. Heat olive oil in a large pot over medium heat.

**Ingredients:**

- 2 cups cooked chickpeas (or 1 can, drained and rinsed)

**Preparation Time:** 20 minutes

**Cook Time:** 40 minutes

**Allergens:**

Peanut Mustard

[edit recipe](#) [delete recipe](#)

Contact:

Email adress: shiroWhite@gmail.com

Phone number: 06203699205

Ha a szerkesztés gombra kattintasz, akkor egy borzalmasan hasonló oldalra visz át, amelyben az általad megadott mezőkben megjelennek a receptről megadott információk. Nyilván ilyen esetben ahol elég hosszú a recept, nem fért bele egy képernyőképbe itt lehetőségre kerül az adatok szerkesztése, ami a mentés gomb után a megadott információval átíródik.

The screenshot shows a recipe editing form with a light beige background. At the top, there are four ingredient entries, each with a number, a text input field, and a minus button:

- 3.  -
- 4.  -
- 5.  -
- 6.  - +

Below the ingredients is the "Allergens:" section, which contains two columns of buttons. The "Egg" and "Soy" buttons are highlighted in dark red, indicating they are selected. The other buttons are light beige with a dark red border.

<input type="button" value="Gluten"/>	<input type="button" value="Crustaceans"/>
<input checked="" type="button" value="Egg"/>	<input type="button" value="Fish"/>
<input type="button" value="Peanut"/>	<input checked="" type="button" value="Soy"/>
<input type="button" value="Milk"/>	<input type="button" value="Tree nuts"/>
<input type="button" value="Celery"/>	<input type="button" value="Mustard"/>
<input type="button" value="Sesame"/>	<input type="button" value="Sulphur dioxide"/>
<input type="button" value="Lupin"/>	<input type="button" value="Molluscs"/>

At the bottom center of the form is a dark red "Save" button.

A mentés után pedig vissza kerülünk a receptekhez.

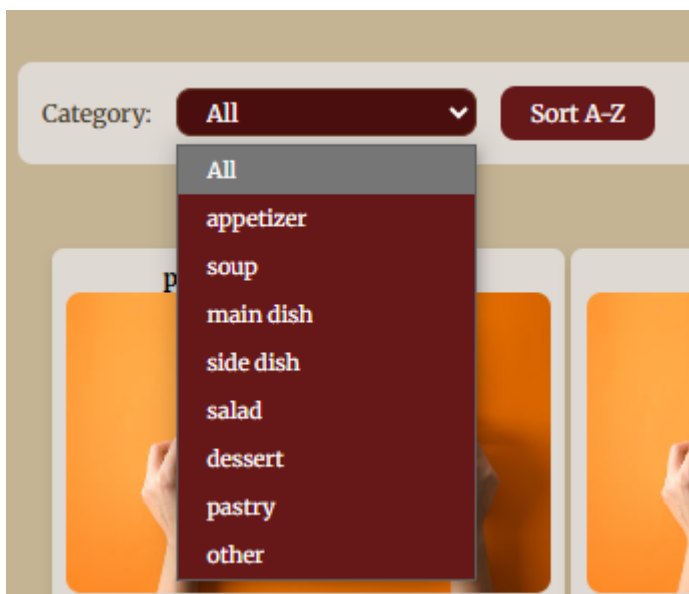
Visszatérve a főoldalunkhoz, ha szeretnénk mást csinálni, mint recepteket feltölteni, akkor van esélyünk a receptek böngészéséhez egy maréknyi segítség, mellyel könnyebbé tehetjük a dolgunkat.

ebbe segíthet a filterezési kategóriánk:



Ebben a kategóriába lehetőségünk van arra, hogy vagy A-tól Z-ig rendezzük a recepteket, vagy rákattintva Z-től A-ig rendezze a recepteket.

Emellett ugye szortírozni lehet kategóriák szerint, ugyan azok a kategóriákból amelyet láthattál a recept feltöltéseknél:

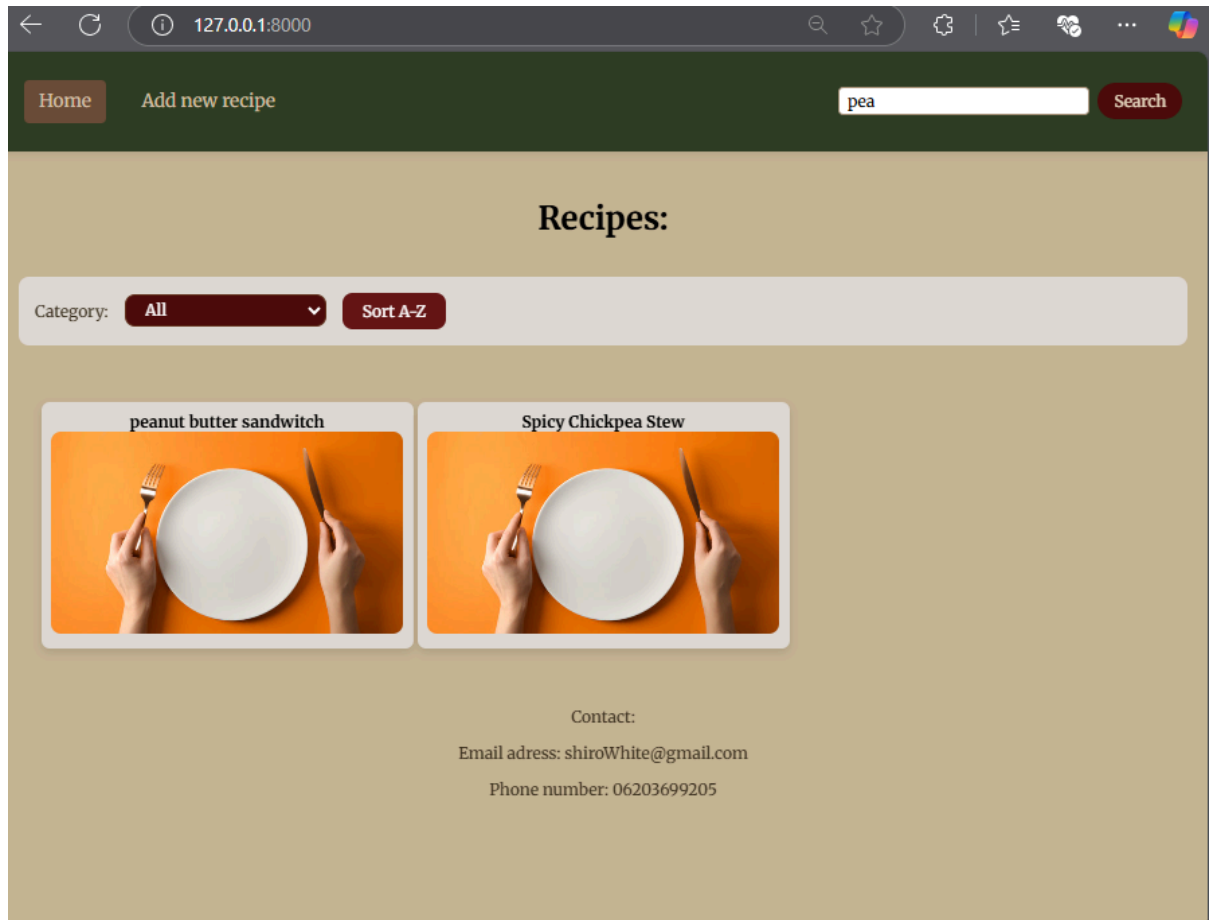


És ha ezek a funkciók nem lennének kielégítőek, akkor hasznodra veheted a keresési funkciót, amit a fejlécbe találsz.





beírt szöveg után, a “keresés” megnyomására kiadja neked a legmegfelelőbb találatokat. Emellett ha nem úgy érzed, hogy meg akarod nyomni a “keresés” gombot, akkor szimplán nyomhatsz egy entert ami ugyanazt az eredményt fogja teljesíteni.



A felhasználás közben ilyen féle hibajelzések megjelenhetnek mint például:

A felhasználó feltöltés közben elfelejtette kiválasztani, hogy milyen kategóriában van a recept amit feltölt.

The screenshot shows a web form for adding a recipe. At the top, a dark grey error message box displays the text: "127.0.0.1:8000 says" followed by "Recipe can't go through because course type isn't specified." and an "OK" button. Below the error, the form fields are as follows:

- Preparation Time (minutes):** A text input field containing the value "15".
- Cook Time (minutes):** A text input field containing the value "10".
- Allergens:** A collection of buttons for selecting allergens. The "Gluten" button is highlighted in dark red. Other buttons include "Crustaceans", "Egg", "Fish", "Peanut", "Soy", "Milk", "Treenuts", "Celery", "Mustard", "Sesame", "Sulphur dioxide", "Lupin", and "Molluscs".
- Save:** A dark red button at the bottom center of the form.

Jelenlegi állapotában minden felhasználó hozzá tud férni az összes recepthez, ezért regisztráció nem szükséges ahhoz, hogy akármilyen funkciót elérj.

Admin felhasználói lehetőségre nincs jelenleg opció.

Karakter limit nincs, vagy legalább is több mint amit bárki bármikor felhasználna.

Karakter érzékenység sincs, Kis és nagy betűk egyaránt felhasználhatók és szimbólumok is használhatók.

# Fejlesztői dokumentáció

A programunk elkészítéséhez Windows 10 és 11 számítógépeket használtunk.

Fejlesztés során használt szoftverek:

- visual studio code
- mysql

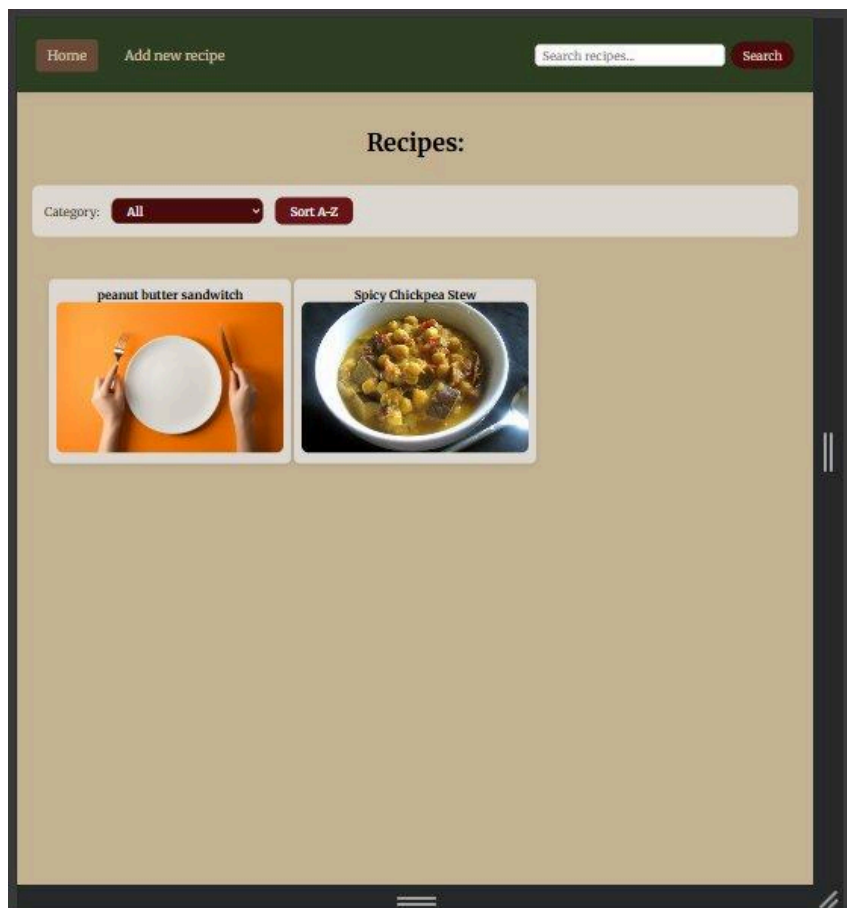
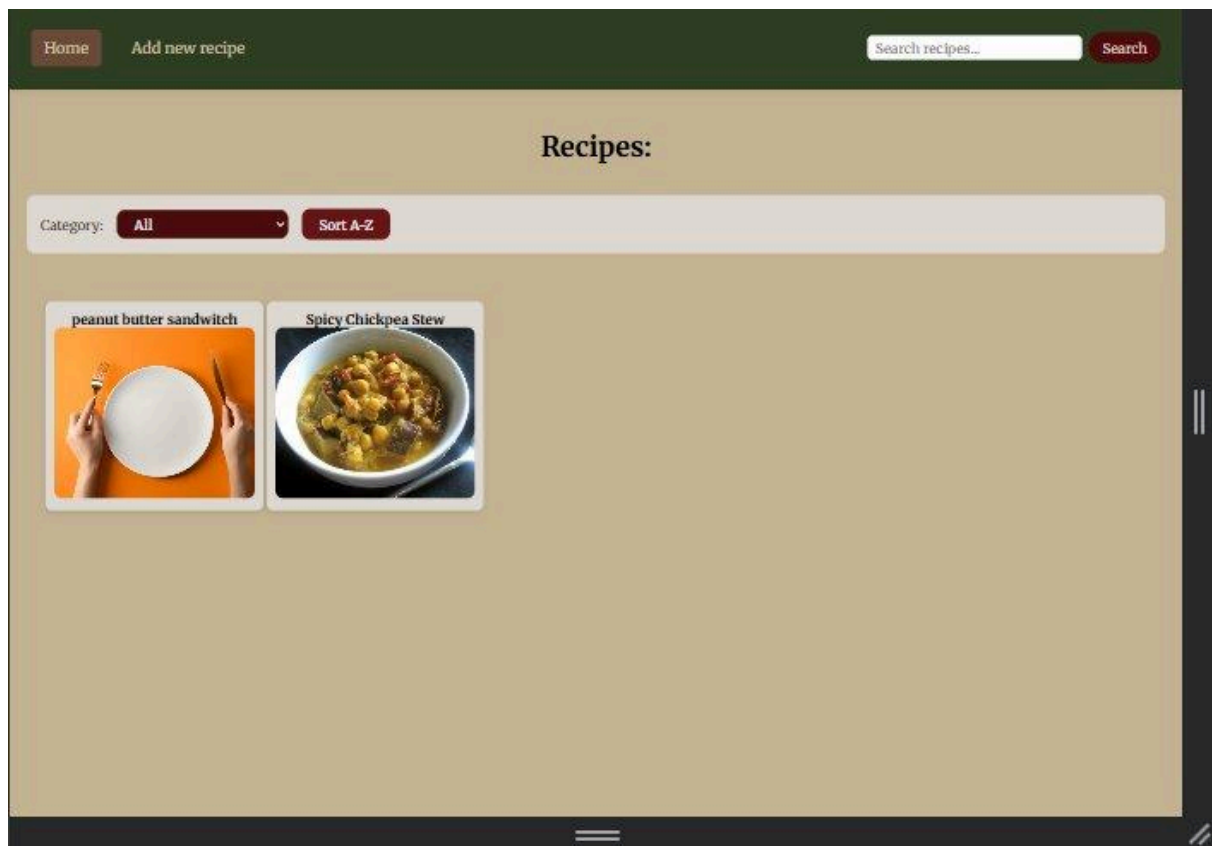
## Frontendről

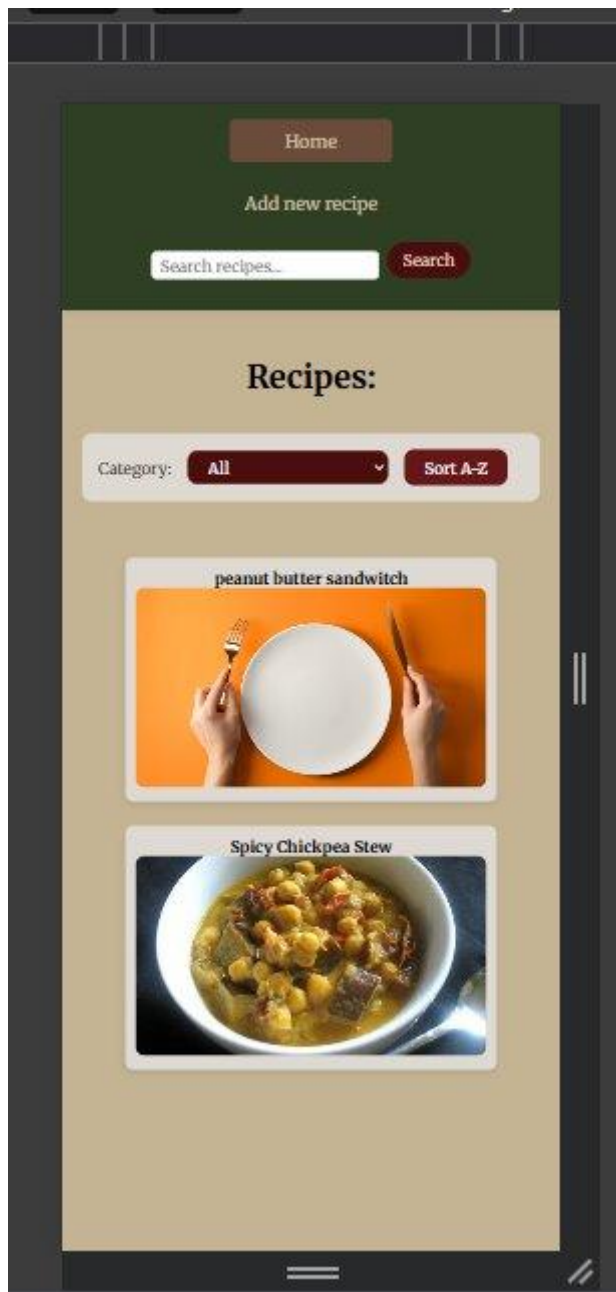
A program megírásához laravelt, pontosabban blade sablonmotort használtunk amiben css, html és javascript használatával valósítottuk meg programunkat

- a blade template egy beépített framework, ami lehetővé teszi, hogy egy könnyen kezelhető web appikációt hozhassunk létre.
- Sok előnyt kínál ez a sablonmotor, mint:
  - egyszerű szintaxis: szimpla és könnyen olvasható a szintaxisa
  - Dinamikus html generálás: lehetővé teszi a dinamikus adatkezelést
  - Újrahasználható kódrészletek: A Blade sablonok segítségével moduláris és újrahasználható kódot hozhatunk létre, amely javítja a fejlesztési folyamat hatékonyságát .

## Reszponzivitás:

Weboldalunk nemcsak hasznos, de majdnem teljesen reszponzív is!itt láthatóan:



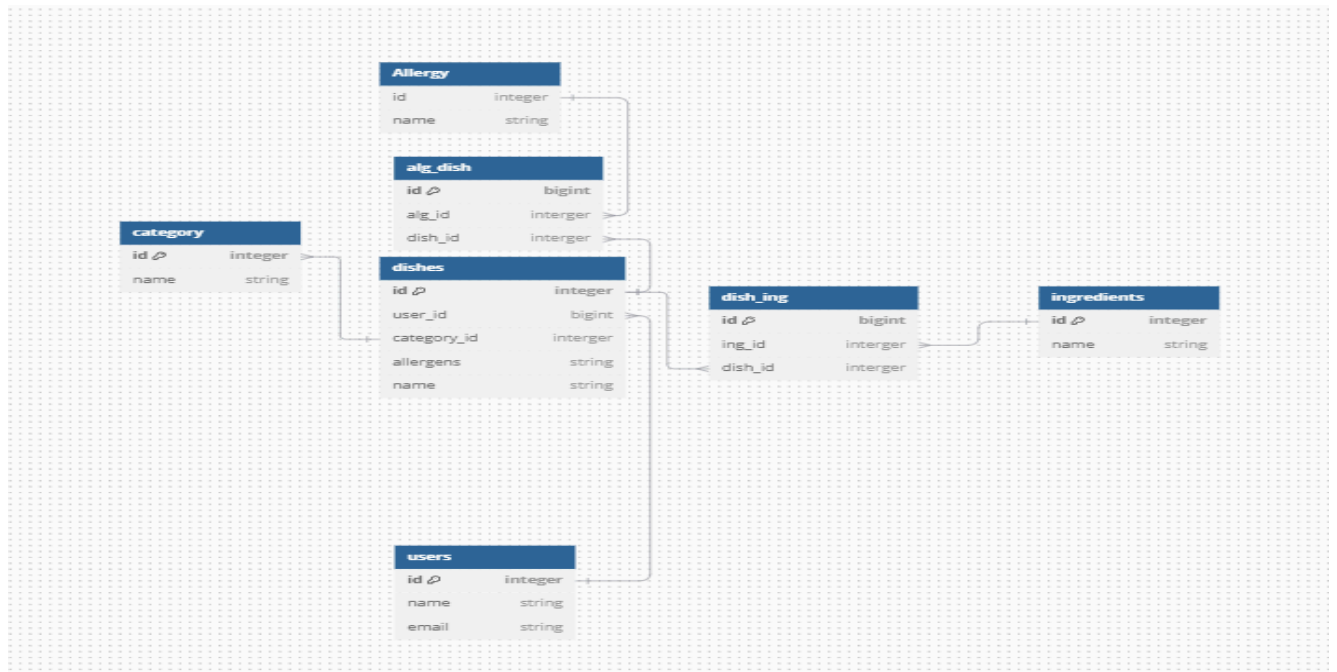


Innen látható, hogy weboldalunk igenis kompatibilis több eszközzel, mind Pc vagy laptop.

# Adatbázisunk

Az adatbázisokhoz MySql rendszert használtunk.

kezdeti vázlat:



Habár első terv óta változott, az elképzelés ugyanaz maradt.. a fő Dish táblához vannak hozzákötve.

## Dish Tábla

A Dish táblába van a legtöbb adat elmentve, mint főbb adatok mint név, kép, stb..foreign key a courses táblához, és adattáblák amik átmenetileg vannak ott mint hozzávalók, és idő ami később saját projektként külön táblába lesz átrakva szűrési szempontból.

create\_dish migration:

```
public function up(): void
{
    Schema::create('dishes', function (Blueprint $table) {
        $table->id();
        $table->foreignId('courseId')->references('id')->on('courses')->default(8)->onDelete('cascade');
        $table->string('name');
        $table->string('desc')->nullable();
        $table->string('img')->nullable()->default('dish.jpg');
        $table->json('ing');
        $table->json('inst');
        $table->integer('prep')->nullable(); //minutes
        $table->integer('cooktime')->nullable(); //minutes
        $table->timestamps();
    });
}
```

Dish model:

```
class Dish extends Model
{
    0 references | 0 overrides
    public function course(): BelongsTo{
        return $this->belongsTo(related: Course::class, foreignKey: "courseId");
    }
    0 references | 0 overrides
    public function allergens(): BelongsToMany{
        return $this->belongsToMany(related: Allergen::class, table: "alg_dish", foreignPivotKey: "dishid", relatedPiv... "alg");
    }
    0 references
    public $table= 'dishes';
    0 references
    protected $casts=[
        'ing'=>'array',

        "inst"=>"array"
    ];
    0 references
    protected $fillable=["courseId", "name", "desc", "ing", "inst", "img", "prep", "cooktime"];
}
```

Dish seeder:

```
0 references | 0 overrides
public function run(): void
{
    $Dishes=[
        [
            "name"=>"peanut butter sandwich",
            "courseId"=>"8",
            "desc"=>"yummy pbj suitable for every meal",
            "img"=>"dish.jpg",
            "inst"=>array("take 2 slices of bread", "put peanut butt
            "ing"=>array("bread", "peanut butter", "jelly",),
            "prep"=>5,
            "cooktime"=>null,
        ],
        [
            "name"=>"garlic bread",
            "courseId"=>"1",
            "desc"=>"yummy garlic bread great with pasta",
            "img"=>"dish.jpg",
            "inst"=>array("take a baguette and cut into it", "slather
            "ing"=>array("baguette", "garlic butter", "cheese"),
            "prep"=>5,
            "cooktime"=>10,
        ],
    ];
    foreach($Dishes as $d){
        Dish::create(attributes: $d);
    }
}
```

```
$this->call(class: [
    DishSeeder::class, AlgDishSeeder::class
]);
```

## Courses tábla

A Courses tábla a receptnek a kurzus megadásáért felelős tábla, amit a Dish tábla a courseId foreignId-n keresztül ér el.

Az adatokat egy [wikipédia oldal](#) alapján szereztük.

1 garlic bread

2 appetizer  
test

```
public function up(): void
{
    Schema::create(table: 'courses', callback: function (Blueprint $table): void {
        $table->id();
        $table->string(column: 'courseName');
    });
}
```

```
$Courses = ['appetizer', 'soup', 'main dish', 'side dish', 'salad', 'dessert', 'pastry', 'other'];
foreach ($Courses as $c){
    Course::create(attributes: ['courseName'=> $c]);
}
```

```
class Course extends Model
{
    0 references | 0 overrides
    public function dishes(): HasMany{
        return $this->hasMany(related: Dish::class);
    }
    0 references
    public $timestamps= false;
    0 references
    public $table= 'courses';
    0 references
    protected $fillable=['id', "courseName"];
}
```



## Allergens tábla

Az Allergens tábla hasonló a Courses táblához, viszont neki az adatai az Alg\_Dish outer join táblában vannak elmentve.

az adatokat a Eus 14 allergén táblázat alapján, [erről az oldalról](#) szereztük

create\_allergens migration:

```
public function up(): void
{
    Schema::create(table: 'allergens', callback: function (Blueprint $table): void {
        $table->id(column: 'id');
        $table->string(column: 'allergenName');
    });
}
```

Model:

```
8 references | 0 implementations
class Allergen extends Model
{
    0 references | 0 overrides
    public function algdish(): HasMany{
        return $this->hasMany(related: alg_dish::class);
    }
    0 references
    public $timestamps= false;
    0 references
    public $table= 'allergens';
    0 references
    protected $fillable=['id', "allergenName"];
}
```

seeder:

```
$Allergens=[
    ["id"=>"1","allergenName"=> "Gluten"],
    ["id"=>"2","allergenName"=> "Crustaceans"],
    ["id"=>"3","allergenName"=> "Egg"],
    ["id"=>"4","allergenName"=> "Fish"],
    ["id"=>"5","allergenName"=> "Peanut"],
    ["id"=>"6","allergenName"=> "Soy"],
    ["id"=>"7","allergenName"=> "Milk"],
    ["id"=>"8","allergenName"=> "Treenuts"],
    ["id"=>"9","allergenName"=> "Celery"],
    ["id"=>"10","allergenName"=> "Mustard"],
    ["id"=>"11","allergenName"=> "Sesame"],
    ["id"=>"12","allergenName"=> "Sulphur dioxide"],
    ["id"=>"13","allergenName"=> "Lupin"],
    ["id"=>"14","allergenName"=> "Molluscs"]
];
```

## Alg\_Dish tábla

Az Alg\_dish tábla egy outer join tábla ami a Dish tábla id-ját mint dishid, és az Allergens tábla id-ját mint alg.

```
public function up(): void
{
    Schema::create(table: 'alg_dish', callback: function (Blueprint $table): void {
        $table->foreignId(column: 'dishid')->references(column: 'id')->on(table: 'dishes')->onDelete(action: 'cascade');
        $table->foreignId(column: 'alg')->references(column: 'id')->on(table: 'allergens')->nullable()->onDelete('cascade');
    });
}
```

```
class alg_dish extends Model
{
    0 references | 0 overrides
    public function algdish(): BelongsToMany{
        return $this->belongsToMany(related: Dish::class, table: Allergen::class);
    }
    0 references
    public $timestamps= false;
    0 references
    public $table= 'alg_dish';
    0 references
    protected $fillable=['dishid', "alg"];
    //protected $casts=["id"=> "array"];
}
```

```
public function run(): void
{
    $alg=[
        ["dishid"=> 1,"alg"=>1],
        ["dishid"=> 1,"alg"=>5],
        ["dishid"=> 2, "alg"=>1],
        ["dishid"=> 2, "alg"=>7],
    ];
    foreach ($alg as $a) {
        alg_dish::create(attributes: $a);
    }
}
```

```
$this->call(class: [
    DishSeeder::class, AlgDishSeeder::class
]);
```

## Backend

A backend részét a laravel blade keretrendszerének a DishController előre generált function-eivel használva írtuk meg. Ezek a web.php-ban vannak meghívva.

web.php route-jai

```
Route::get(uri: '/', action: [DishController::class, 'index']);
Route::get(uri: '/dish/{id}', action: [DishController::class, 'show'])->name(name: "dish");
Route::get(uri: '/create', action: [DishController::class, 'create'])->name(name: "create");
Route::post(uri: '/store', action: [DishController::class, 'store'])->name(name: "store");
Route::get(uri: '/dish/edit/{id}', action: [DishController::class, 'edit'])->name(name: "edit");
Route::post(uri: '/update/{id}', action: [DishController::class, 'update'])->name(name: "update");
Route::get(uri: 'dish/delete/{id}', action: [DishController::class, 'destroy'])->name(name: "dish/delete");
```

## Dishcontroller

### Index function

Az index function lehívja az összes rekordot a Dish táblából, és átadja tömbként a front view-be a \$dish változón keresztül.

```
public function index()
{
    $dish= Dish::all();
    return view('front', compact('dish'));
}
```

### Create function

A create function hasonló, csak itt az Allergens és a Courses tábláknak hívja le az adatait, amit az \$alg és \$course változókban adja át amikor visszaadja a create view-t.

```
public function create(): View
{
    $alg= Allergen::all();
    $course= Course::all();
    return view('create', data: ['alg'=>$alg, 'course'=>$course]);
}
```

## Store function

A store function már komplikáltabb, de mindenek előtt fut egy validálás.

```
public function store(Request $request): RedirectResponse
{
    $request->validate(rules: [
        'name' => 'required|min:3',
        'courseId' => 'required',
        'desc' => 'string',
        'img' => 'nullable|mimes:png,jpg,jpeg',
        'inst' => 'array',
        'ing' => 'array',
        'allergens' => 'array',
        'prep' => 'integer',
        'cooktime' => 'nullable|integer',
    ]);
}
```

Ezután megnézi, hogy jött-e a request-ben egy kép.

Ha igen, akkor a feltöltött képet a public/dishimg mappába menti el, és a fájl nevét a \$filename változóban menti el.

Viszont ha nem akkor a public/placeholder mappában lévő dish.jpg képre hivatkozik.

Ez a rész egy [youtube videó](#) alapján lett megírva

```
if($request->has(key: 'image')){
    $img = $request->file(key: 'image');
    $ext = $img->getClientOriginalExtension();
    $filename = time().'.'.$ext;
    $imgpath = 'dishimg/';
    $img->move(directory: $imgpath, name: $filename);
}
else{
    $filename = 'dish.jpg';
}
```

Ezt követően létrehozza a Dish táblában szereplő adatokat, a `::create` function segítségével, amit egyúttal a `$dish` változóba is belerak.

```
$dish= Dish::create(attributes: [
  'name'=> $request['name'],
  'courseId'=>$request['courseId'],
  'desc'=>$request['desc'],
  'img'=>$filename,
  'inst'=> $request['instructions'],
  'ing'=>$request['ingredients'],
  'prep'=>$request['prep'],
  'cooktime'=>$request['cooktime']
]);
```

Végül megnézi, hogy a requestben érkezett allergens tömbben adat, ha nem akkor redirect-el a show view-re.

Viszont hogy ha érkezett adat, akkor egy foreach végigmegy a tömbön, és az Alg\_dish model a `::create` function-jével minden allergénnek készít egy rekordot, és hozzáköti a már létrehozott rekordhoz a `$dish` változó id-jával.

Végül redirect-el a show view-re.

<u>2</u>	<u>1</u>		<u>2</u>	<u>1</u>
2	garlic bread		2	7
3	6		3	6

```
if(empty($request['allergens'])){\n  return redirect(to: 'dish/'.$dish->id)->with(key: 'success',value: '');\n}\nelse{\n  foreach ($request['allergens'] as $a) {\n    alg_dish::create(attributes: [\n      'dishid'=>$dish->id,\n      'alg' => $a,\n    ]);\n  }\n}\n\nreturn redirect(to: 'dish/'.$dish->id)->with(key: 'success',value: '');
```

## Show function

A show function hasonló az index function-hoz, viszont itt csak egy recept adatait hívjuk le id alapján.

Mielőtt még bármi máshoz nyulna, megnézi hogy létezik-e rekord ezzel az id-val, ha nem, akkor redirect-el a front view-ra.

Ha viszont van, akkor lehívja az Allergens, és a Courses táblákból az összes adatot. Végül mindezt átadva redirect-el a show view-re.

```
public function show($id): RedirectResponse|View
{
    $dish=Dish::find( id: $id );
    if(!$dish){
        return redirect(to: '/')->with(key: 'fail',value: "no dish found");
    }

    $alg= Allergen::all();
    $course= Course::all();
    return view(view: "show", data: ['alg'=>$alg, 'course'=>$course,"dish"=>$dish]);
}
```

## Edit function

Az edit function szinte ugyanaz mint a show function, csak itt már nem kell megvizsgálni hogy létezik-e a rekord, mivel a szerkesztés a show view-en keresztül érhető el.

```
public function edit($id): View
{
    $dish=Dish::find( id: $id );
    $alg= Allergen::all();
    $course= Course::all();

    return view(view: 'edit', data: ['alg'=>$alg, 'course'=>$course,'dish'=>$dish]);
}
```

## Update function

Az update function hasonló a show function-hoz.

Először is megkeresi a rekordot az id alapján, majd fut egy validálás.

```
public function update(Request $request, $id, ): RedirectResponse
{
    $dish= Dish::find(id: $id);

    $request->validate(rules: [
        'name' =>'required|min:3',
        'courseId'=>'required',
        'desc'=> 'string',
        'img'=> 'nullable|mimes:png,jpg,jpeg',
        'inst'=> 'array',
        'ing'=>'array',
        'allergens'=>'array'
    ]);
}
```

Ezután megnézi, hogy jött-e a requestbe kép, ha igen, akkor törli az előzőt, majd létrehoz egy újat ugyanúgy mint a create-ben.

Ez a módszer megelőzi a már nem használt képek tárolását

```
if($request->has(key: 'image')){
    $imgpath= public_path(path: 'dishimg/'.$dish->img);
    if (File::exists(path: $imgpath)) {
        File::delete(paths: $imgpath);
    }
    $img = $request->file(key: 'image');
    $ext= $img->getClientOriginalExtension() ;
    $filename=time().'.'.$ext;
    $imgpath='dishimg/';
    $img->move(directory: $imgpath,name: $filename);
}
```

Majd az update-eli a dish tábla többi adatát

```
$dish->update([
    'name'=> $request['name'],
    'courseId'=>$request['courseId'],
    'desc'=>$request['desc'],
    'img'=>$filename,
    'inst'=> $request['instructions'],
    'ing'=>$request['ingredients'],
]);
```

Végül miután törli azokat a rekordokat ahol a dishid megegyezik a \$dish id-jával, ugyanúgy mint a create-ben létrehozta a requestben kapottak alapján. Ez a módszer megelőzi a duplikált adatokat, valamint nem marad az adatbázisban rekord hogyha a felhasználó kiveszi onnan.

```
alg_dish::where(column: 'dishid', operator: $id)->delete();
if(empty($request['allergens'])){
    return redirect(to: 'dish/'.$dish->id)->with(key: 'success',value: '');
}
else{
    foreach ($request['allergens'] as $a) {
        alg_dish::create(attributes: [
            'dishid'=>$dish->id,
            'alg' => $a,
        ]);
    }
}
return redirect(to: "dish/$id");
```

## Destroy function

A destroy function ugyanúgy mint a szerkesztés, szintén a show view-en keresztül érhető el.

Mivel maga a rekord törlése nem törli a public/dishimg-ből a képet, így ha létezik a fájl, az unlink function-el elérhető annak is a törlése.

Végül redirect-el a front view-ra.

kép törlése egy [youtube videó](#) alapján lett elkészítve

```
public function destroy(string $id): RedirectResponse
{
    $dish=Dish::find(id: $id);
    $imgpath= public_path(path: 'dishimg/'.$dish->img);
    if (File::exists(path: $imgpath)) File::delete(paths: $imgpath);

    $dish->delete();

    return redirect(to: '/')->with(key: 'success');
}
```



# Befejezés

## Csapatmunka:

kommunikáció személyesen, és discordon keresztül.

Lele regina: frontend, css, javascript, adatok feltöltése

Kovács Orsolya: backend, adatbázis kezelés, és tervezés

## Jövőbeli Tervek a programmal kapcsolatban:

-felhasználói felület

-admin felület

## Irodalomjegyzék:

- [Laravel gyorsalpaló - YouTube](#)
- [W3Schools Online Web Tutorials](#)
- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [Blade Templates - Laravel 12.x - The PHP Framework For Web Artisans](#)

## Összefoglalás

A csapatmunkánkat jóként értékelném, mivel magunkhoz képest jól teljesítettünk.

Emellett a program hasznolására a jövőben remélhetően a továbbfejlesztés fog rá kerülni és remélhetőleg egy nagyobb körű felhataltságot tud nyerni.

nyilvános elérés: <https://github.com/kovkrist5/recipe-website>