

Dokumentáció – Szerelőműhely

Skull emoji csapata

Adatbázis:

- Minden főbb adatfajának csináltunk egy táblát és minden táblához csatoltunk egy modell-t.
- A felhasználókat a USERS beépített modellel oldottuk meg, amit megváltoztattunk a saját igényeinkre.
- Az adatbázist XAMPP segítségével a phpmyadmin felületen teszteltük ami egy mysql adatbázis.

Bejelentkezés/Regisztráció:

- A felhasználóknak biztosítottunk bejelentkező és regisztráció lehetőséget, amellyel „szerelő” és „munkafeltevő” pozíciókban lehet belépni/regisztrálni.
- Ezeket úgy alakítottuk ki hogy amikor valaki regisztrál és nincs benne a szerelő/munkafeltevő táblában akkor hozzáadja a Controller az adott táblához
- Mindkettő pozícióban bejelentkezett dolgozók csak a saját számukra kijelölt dolgokat tudják elérni, ha URL alapján akarják keresni akkor visszairányítja az oldal a kezdőmenübe (Dashboard).
- Továbbá van egy ADMIN aki mindent elér az adatbázisban és az összes oldalhoz hozzáférése van, ahogy minden funkcióhoz is.

Munkafeltevő:

- A munkafeltevő listázhatja az összes munkalapot és változtathatja őket.
- A munkalapokat lezárhatja, és lezárás után ki is nyomtathatja azokat.
- Munkalapokat is létrehozhat.
- Minden parancsát a ListController nevezetű controller intézi el, beszúrást, módítást és a listázást.
- Biztosítottunk neki 4 view-t a feladatainak elvégzésére amelyek <a> tagbe beleágyazott URL és ROUTE parancsokkal oldottuk meg.

Szerelő

- Listázhatja a hozzárendelt munkalapokat és munkafolyamatokat, anyagokat és alkatrészeket adhat hozzájuk.
- Az ő feladatait 3 view-al oldottuk meg amik egymás között 1-2 adatot passzolnak át egymáshoz.

ListController Függvények:

- lista() -> A munkafelvevőnek a munkalapok kiírását/keresését biztosítja

```
public function lista(Request $request){  
    $search = $request->input('search');  
    $showZart = $request->input('zarte', false);  
  
    $munkalapok= Munkalap::query()  
        ->when($search, function ($query, $search) {  
            $query->where('szerelo_azonosito', 'like', "%{$search}%");  
        }) ->when($showZart, function ($query) {  
            $query->where('lezart', 1);  
        })  
        ->orderBy('datum', 'desc')->get();  
  
    return view('munkafelvevo.munkalaplista', compact('munkalapok',  
        'search','showZart'));  
}
```

- `modosit()` -> A munkalapok módosításánál a friss adatok kiírására felel

```
public function modosit(Request $request){  
  
    $id = $request->id;  
  
    $munkalap = Munkalap::findOrFail($id);  
  
    $munkalap = Munkalap::where('id', $id)->first(['id','munkafelvevo_azonosito',  
'szerelo_azonosito', 'gepjarmu_rendszam', 'lezart', 'osszar', 'fizetesi_mod']);  
  
    $gepjarmu = Gepjarmu::where('rendszam', $munkalap->gepjarmu_rendszam)->first(['gyartmany', 'tipus', 'tulaj_id']);  
  
    $tulaj = Tulajdonos::where('id', $gepjarmu->tulaj_id)->first(['nev', 'cim']);  
  
    return view('munkafelvevo.munkalapModositas', compact('munkalap',  
'gepjarmu', 'tulaj'));  
  
}
```

- `hozzaadas()` -> Az új munkalap hozzáadására felel a megadandó információk alapján ad hozzá egy új Gépjárművet, Munkalapot és ha nem létezik akkor Tulajdonost

```
public function hozzaadas(Request $request){  
    $request->validate([  
        'felvevoAzonosito' => ['required', 'string'],  
        'szereloAzonosito'=> ['required', 'string', 'min:6', 'max:6'],  
        'jarmuRendszam' => 'required|string|min:7|max:7',  
        'jarmuGyartmany' => 'required|string|max:30',  
        'jarmuTipus' => 'required|string',  
        'jarmuTulajNev' => 'required|string',  
        'jarmuTulajCim' => 'required|string',  
        'fizetesiMod' => 'required'  
    ]);  
  
    if(Tulajdonos::where('cim', $request->jarmuTulajCim)->count() < 1){  
        Tulajdonos::create([  
            'nev' => $request->jarmuTulajNev,  
            'cim' => $request->jarmuTulajCim  
        ]);  
    }  
  
    $tulajID = Tulajdonos::where('cim', $request->jarmuTulajCim)->get('id')->first();  
  
    Gepjarmu::create([  
        'rendszam' => $request->jarmuRendszam,  
        'gyartmany' => $request->jarmuGyartmany,  
        'tipus' => $request->jarmuTipus,  
        "tulaj_id" =>$tulajID["id"]  
    ]);  
}
```

```

Munkalap::query()->insert([
    "szerelo_azonosito" => $request->szereloAzonosito,
    "datum" => now()->format('Y-m-d'),
    "munkafelvevo_azonosito" => $request->felvevoAzonosito,
    "gepjarmu_rendszam" => $request->jarmuRendszam,
    "lezart" => 0,
    "osszar" => 0,
    "fizetesi_mod" => $request->fizetesiMod ,
    'created_at'=> now(),
    'updated_at' =>now()
]);

return redirect()->route('dashboard')->withSuccess('Sikeres Létrehozás!');
}

```

- frissites() -> A munkalapok módosítására felel

```

public function frissites(Request $request)
{
    $id = $request->id;

    $request->validate([
        'szereloAzonosito' => 'required|string|size:6|exists:szerelos,azonosito',
        'fizetesiMod' => 'required|in:kartya,keszpenz',
    ]);

    Munkalap::query()->where('id', $id)->update(['szerelo_azonosito' =>
    $request->szereloAzonosito, 'fizetesi_mod'=> $request->fizetesiMod, 'lezart' =>
    $request->lezart]);

    return redirect()->route('lista')->with('success', 'Munkalap sikeresen
    módosítva.');
```

- worklist() -> A szerlőkhöz hozzárendelt munkalapok megkereséséért felel

```

public function worklist(Request $request){
    $azonosito = Auth::user()->azonosito;

    if($azonosito != "ADMIN!")
        $munkalapok = Munkalap::query()->where('szerelo_azonosito', $azonosito)-
        >get();
    else
        $munkalapok = Munkalap::query()->get();

    return view('szerelo.munkaListazas', compact('munkalapok'));
}
```

- bovito() -> A munkalaphoz való művelet hozzáadásánál ez mutatja meg az eddigi anyagokat, alkatrészeket és munkafolyamatokat.

```
public function bovito(Request $request){  
    $munkalapID = $request->id;  
  
    $munkalapAnyagok = MunkalapAnyagok::query()->where('munkalap_id',  
$munkalapID)->get();  
  
    $anyagok = Anyag::query()->whereIn('id', $munkalapAnyagok-  
>pluck('anyag_id'))->get('nev');  
  
    $munkafolyamat = Munkafolyamat::query()->where('munkalap_id',  
$munkalapID)->get();  
  
    $munkaFolyamatok = Munkafolyamatok::query()->whereIn('id',  
$munkafolyamat->pluck('munkafolyamatok_id'))->get('nev');  
  
    $alkatresz = MunkalapAlkatresz::query()->where('munkalap_id',  
$munkalapID)->get();  
  
    $alkatreszek = Alkatresz::query()->whereIn('id', $alkatresz-  
>pluck('alkatresz_id'))->get('nev');  
  
    return view('szerelo.munkalapBovites',  
compact('munkalapID','munkalapAnyagok','anyagok','munkafolyamat',  
'munkaFolyamatok', 'alkatresz', 'alkatreszek'));
```

- bovit() -> A munkalaphoz való hozzáadásnál az űrlap elkészítéséhez adatokat szolgáltat

```
public function bovit(Request $request){  
    $fajta = $request->fajta;  
    $munkalapID = $request->munkalapID;  
    if($fajta == 'anyag'){  
        $cucc = Anyag::query()->get('nev');  
    }else if($fajta == 'munka'){  
        $cucc = Munkafolyamatok::query()->get('nev');  
    }else{  
        $cucc = Alkatresz::query()->get('nev');  
    }  
    return view('szerelo.Bovit', compact('fajta', 'munkalapID', 'cucc'));  
}
```


- adas() -> A munkalaphoz hozzáadja az felhasználónál specifikált anyagokat, alkatrészeket és munkafolyamatokat.

```
public function adas(Request $request){

    $request->validate([

        'cucc' => 'required|string',

        'mennyiseg' => 'required|string'

    ]);

    $munkalapID = $request->munkalapID;

    if($request->fajta == 'anyag'){

        $yes = Anyag::query()->where('nev', '=', $request->cucc)->pluck('id')->first();

        MunkalapAnyagok::query()->insert(['munkalap_id' => $munkalapID,
        'anyag_id' => $yes, 'mennyiseg' => $request->mennyiseg, 'created_at'=>
        now(), 'updated_at' => now()]);

    }else if($request->fajta == 'munka'){

        $yes = Munkafolyamatok::query()->where('nev', '=', $request->cucc)->pluck('id')->first();

        Munkafolyamat::query()->insert(['munkalap_id' => $munkalapID,
        'munkafolyamatok_id' => $yes, 'idotartam' => $request->mennyiseg,
        'created_at'=> now(), 'updated_at' => now()]);

    }else{

        $yes = Alkatresz::query()->where('nev', '=', $request->cucc)->pluck('id')->first();

        MunkalapAlkatresz::query()->insert(['munkalap_id' => $munkalapID,
        'alkatresz_id' => $yes, 'mennyiseg' => $request->mennyiseg, 'created_at'=>
        now(), 'updated_at' => now()]);

    }

    return redirect()->route('worklist', ['azonosito' => Auth::user()->azonosito]);

}
```

- megletekint() -> A nyomtatáshoz a nézetet ez csinálja meg.

```
public function megletekint(Request $request){  
    $id = $request->id;  
  
    $munkalap = Munkalap::findOrFail($id);  
  
    $munkalap = Munkalap::where('id', $id)->first(['id','munkafelvevo_azonosito',  
'szerelo_azonosito', 'gepjarmu_rendszam', 'lezart', 'osszar', 'fizetesi_mod']);  
  
    $gepjarmu = Gepjarmu::where('rendszam', $munkalap->gepjarmu_rendszam)->first(['gyartmany', 'tipus', 'tulaj_id']);  
  
    $tulaj = Tulajdonos::where('id', $gepjarmu->tulaj_id)->first(['nev', 'cim']);  
  
    return view('munkafelvevo.munkalapMegtekintes', compact('munkalap',  
'gepjarmu', 'tulaj'));  
}
```