**Comproller  
Dokumentációk**

Felhasználói Dokumentáció

* Bevezetés

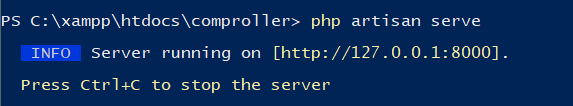
A **Comproller** egy sokoldalú, egyszerűen használható vállalatirányító program. A felhasználó képes új munkavállalókat létrehozni, módosítani és törölni. A rendszer lehetőséget nyújt arra, hogy a munkavállalókról részletes információkat tároljunk, például lakcím, TAJ-szám, telefonszám és egyéb adatok. Ezek az információk könnyen szerkeszthetők, illetve szükség esetén eltávolíthatók a nyilvántartásból.

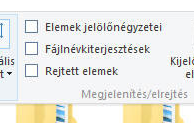
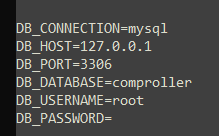
A program főbb funkciói:

* **Biztonság:** A felhasználó regisztrálhat egy fiókot a programon belül hogy hozzáférjen a funkciókhoz.
* **Eseménykezelő rendszer:** A felhasználók tudnak eseményeket rögzíteni a kiválasztott dátumon és megnevezni azt, ezek bármikor törölhetőek. Az események láthatóak az irányítópulton valamint az események oldalon is.
* **Adatkezelés:** Az adatok exportálhatók .PNG formátumba ha megosztásra van szükség.
* **Ki, -Be léptető rendszer:** A szoftvert a munkavállaló létrehozásánál legenerál egy qrcode-ot a munkavállalónak amellyel a program képes rögzíteni a ki- és belépési idejét, viszont lehetőséget kínál manuális rögzítésre is.
* **Bérszámfejtés:** A kijelölt oldalon a felhasználó kikeresheti a dolgozót és részletes információt tekinthet meg a órabéri adatairól.
* **Fiókkezelés:** A felhasználó a profil oldalon tudja kezelni bejelentkezési adatait. Változtatni a fiók nevét és jelszavát, valamint a fiókot törölni
* **Beüzemeltetés:**

*A program beüzemeltetésének lépései:*

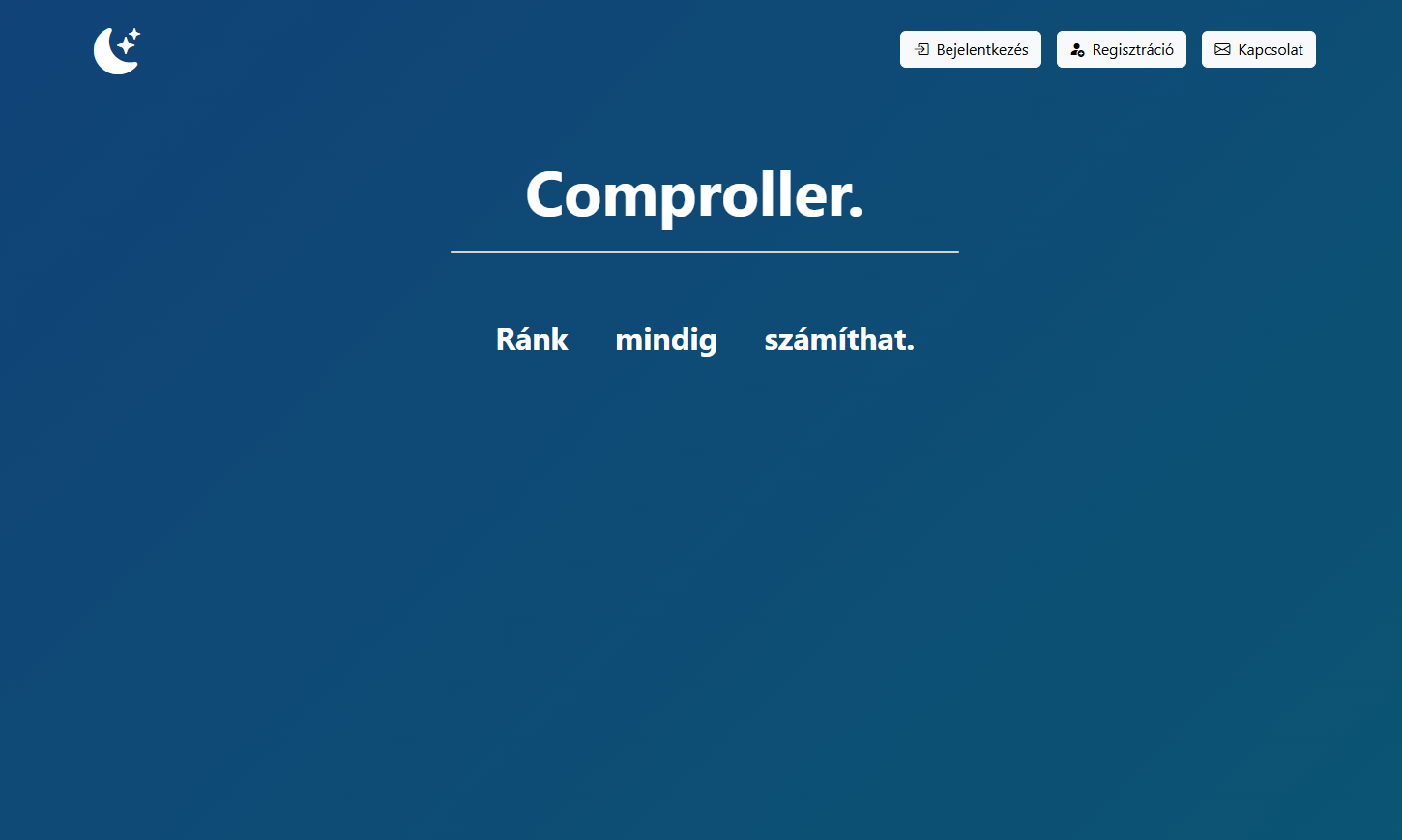
1. *Győződjünk meg róla hogy a Composer és xampp telepítve van. Valamint a fájl kezelőbe láthatók a rejtett elemek e.*
2. *Állitsuk be az .env fájl konfigurációját.*
3. *Töltsük fel a mellékelt comproller.sql fájl-t adatbázisba vagy php artisan migrate-el madj migráljuk be.*
4. *Navigáljunk a comproller mappába és shift+jobb egér kattintással menjünk rá a Powershell ablak megnyitására.*
5. *Frissitsük le a composer-t ha szükséges a következő parancssal: composer update*
6. *Állitsunk egy kúlcsot az alkalmazásnak szükség esetén a következő parancssal: php artisan key:generate*
7. *Índitsuk el az alkalmazás a php artisan serve parancssal, majd navigáljunk a localhost:8000-es link-re.*





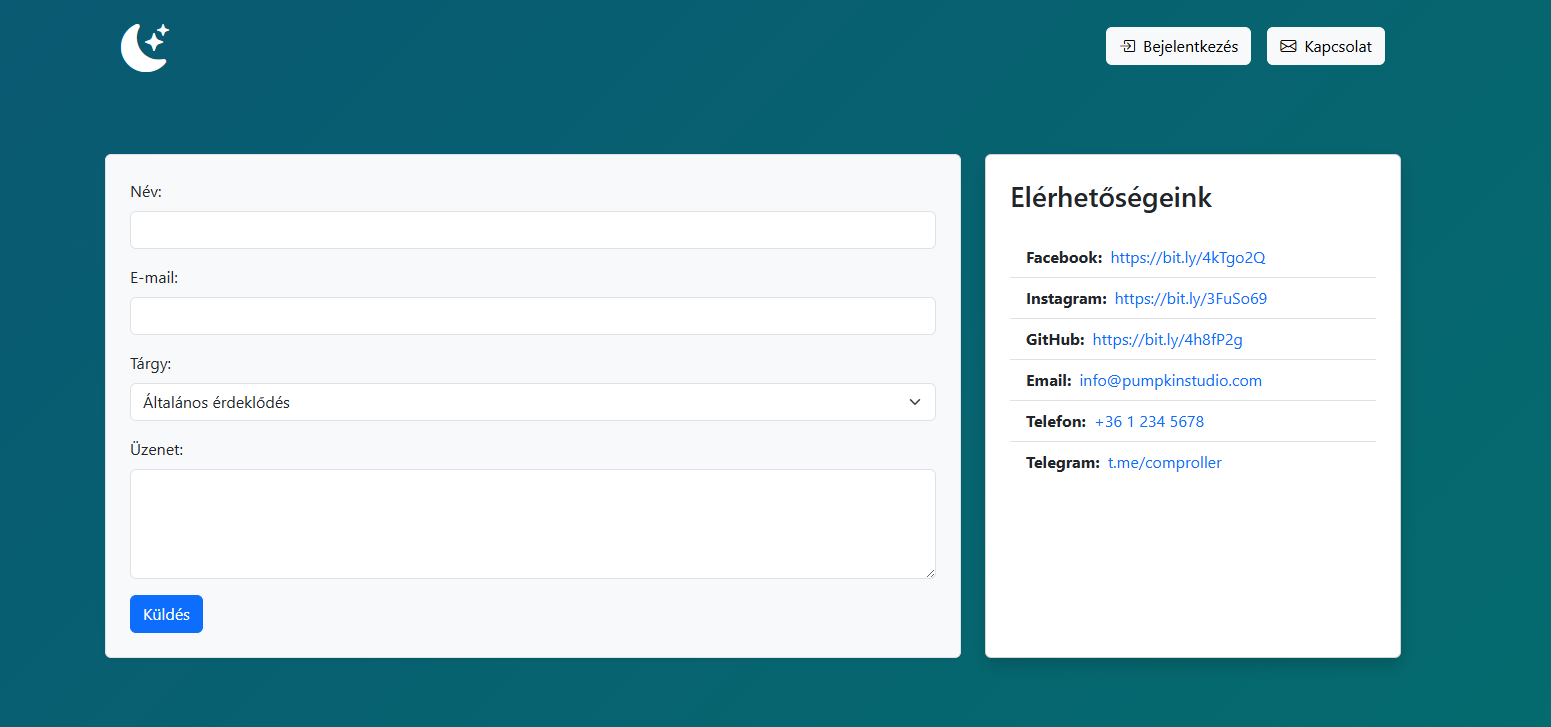
* **Főoldal:**

*A program indításakor a felhasználót a főoldal fogadja. Innen három lehetőség közül választhat: átnavigálhat a kapcsolatok részlegre, regisztrálhat vagy bejelentkezhet a fiókjába. Ha a bejelentkezés vagy regisztráció sikeres, a felhasználót az irányítópult oldalra irányítja a rendszer, ahonnan további aloldalak is elérhetők.*



* **Kapcsolat:**

*A kapcsolat oldalon a felhasználó visszajelzést küldhet a fejlesztőknek. Az oldalon megtekintheti a fejlesztők elérhetőségeit (Facebook, Instagram, GitHub, stb), valamint lehetősége van a saját nevét, e-mail címét és egy üzenet megadására, amelyet egy kiválasztott tárgy témakörében küldhet el. Ha a felhasználó sikeresen elküldte az adatokat, a rendszer átirányítja egy visszaigazoló oldalra, ahol értesítést kap arról, hogy üzenete sikeresen továbbítva lett.*



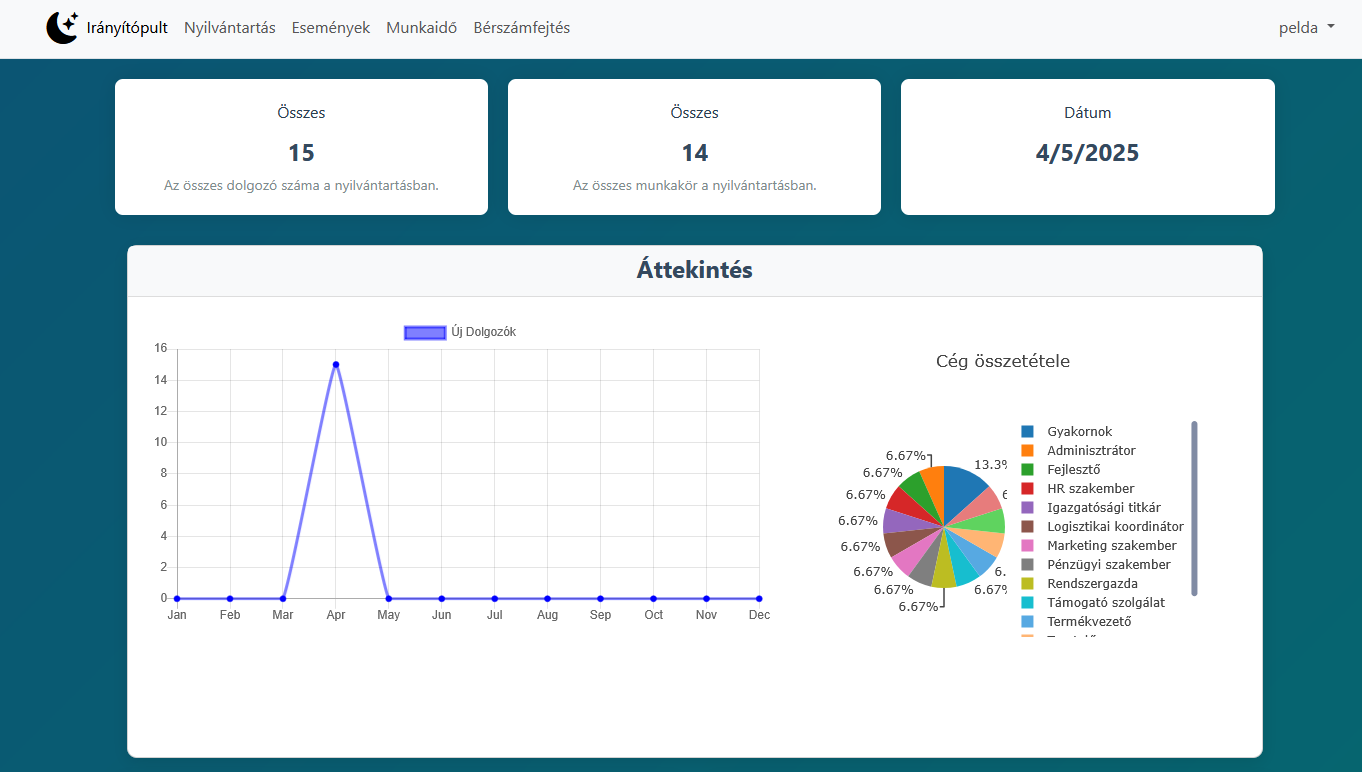
* **Irányítópult:**

*Az irányítópult oldalon részletes információkat tekinthet meg a cég dolgozóiról. Itt látható az összes dolgozó száma, a munkakörök teljes száma, valamint a jelenlegi dátum.*

*Az oldalon két grafikont is talál a felhasználó:*

1. *Az első grafikon azt mutatja, hogy hány dolgozó csatlakozott a céghez hónapok szerint.*
2. *A második grafikon pedig a cégben meglévő munkakörök számát ábrázolja. Ez a grafikon letölthető .png formátumban is.*

*Ezen kívül az oldalon megjelenik a legújabb események listája, valamint a legutóbb felvett dolgozók adatai (ID, vezetéknév, keresztnév és munkakör).*

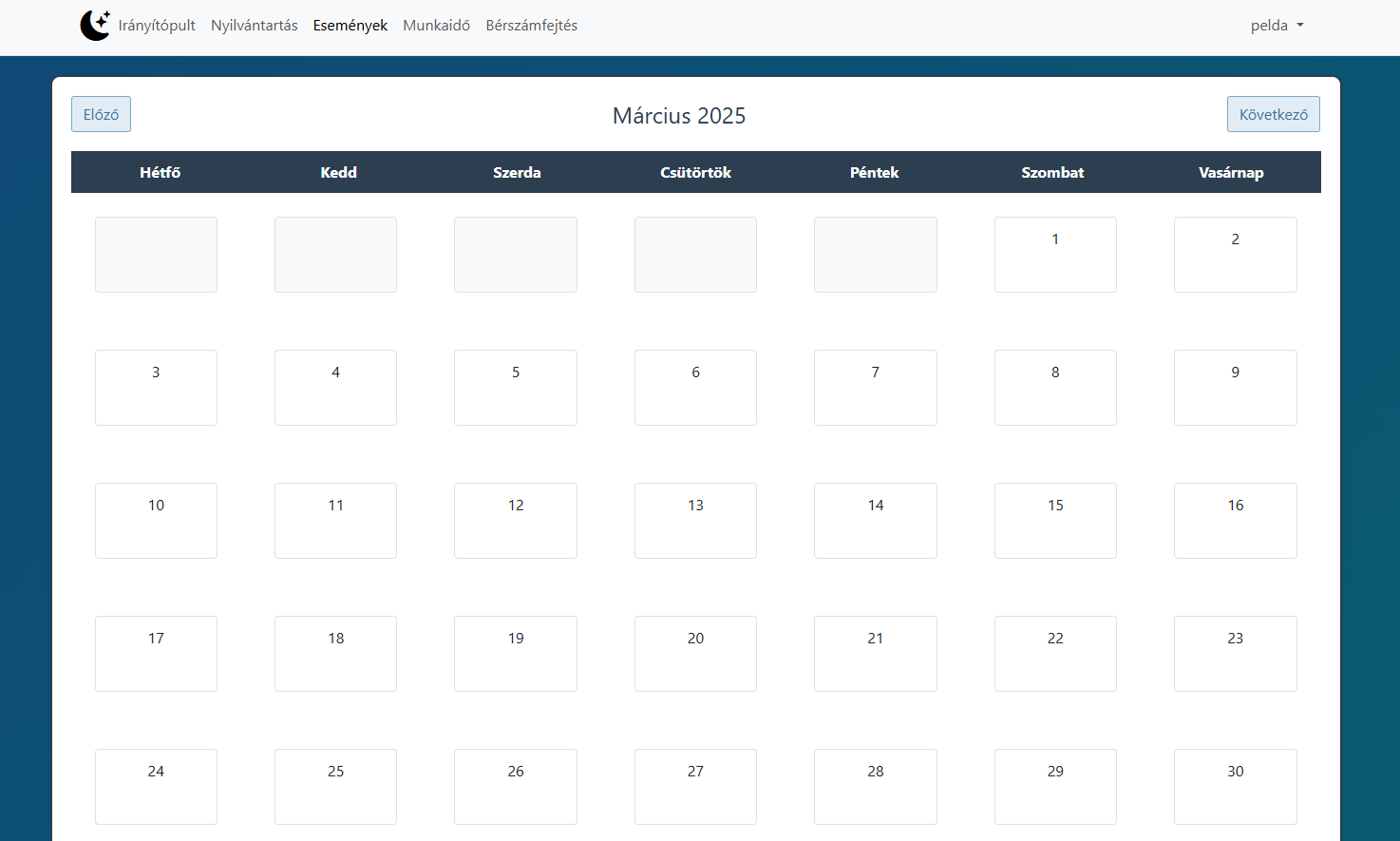


* **Események:**

*Az oldalon egy interaktív naptár található, amely lehetővé teszi események rögzítését. A naptárban a hónapokat váltogathatja a felhasználó, és egy adott napra kattintva kiválaszthatja azt.*

*A kiválasztott nap esetén egy felugró ablak jelenik meg, ahol megadhatja az esemény nevét, majd hozzáadhatja azt, vagy bezárhatja az ablakot. Ha egy eseményt hozzáad, az megjelenik az oldal alján, egy táblázatos formában. Az események mellett található egy törlés gomb, amellyel az adott eseményt eltávolíthatja.*

*Ezenkívül az újonnan hozzáadott események automatikusan megjelennek az irányítópult oldalon is, így mindig naprakész információkkal rendelkezhet a fontos dátumokról.*



* **Munkaidő:**

*A munkaidő nyilvántartás oldalon a dolgozó be- és kijelentkezési adatai tekinthetők meg. Az oldal három fő részből áll:* ***automatikus rögzítés****,* ***manuális rögzítés*** *és* ***lekérdezés****.*

1. ***Automatikus rögzítés****:*  
   *Ez a rész a Qrcode olvasóhoz kapcsolódik. Ha a felhasználó sikeresen beolvassa a Qrcode-ját (bejelentkezés vagy távozás esetén), az adatok automatikusan rögzítődnek. A táblázatban megjelenik a felhasználó* ***ID-ja****,* ***vezeték- és keresztneve****, valamint az* ***érkezési és távozási idő****.*
2. ***Manuális rögzítés****:*  
   *Ha szükséges, a felhasználó kézzel is megadhatja a be- és kijelentkezési adatokat. Ha érvényesek az adatok a rendszer a találatot elmenti a rendszerbe.*
3. ***Lekérdezés****:*  
   *Ebben a részben a felhasználó megadhatja a dolgozó ID-t, majd egy kiválasztott időintervallum alapján lekérdezheti a csekkolási adatokat. A lekérdezés eredményei az oldal alján jelennek meg.*



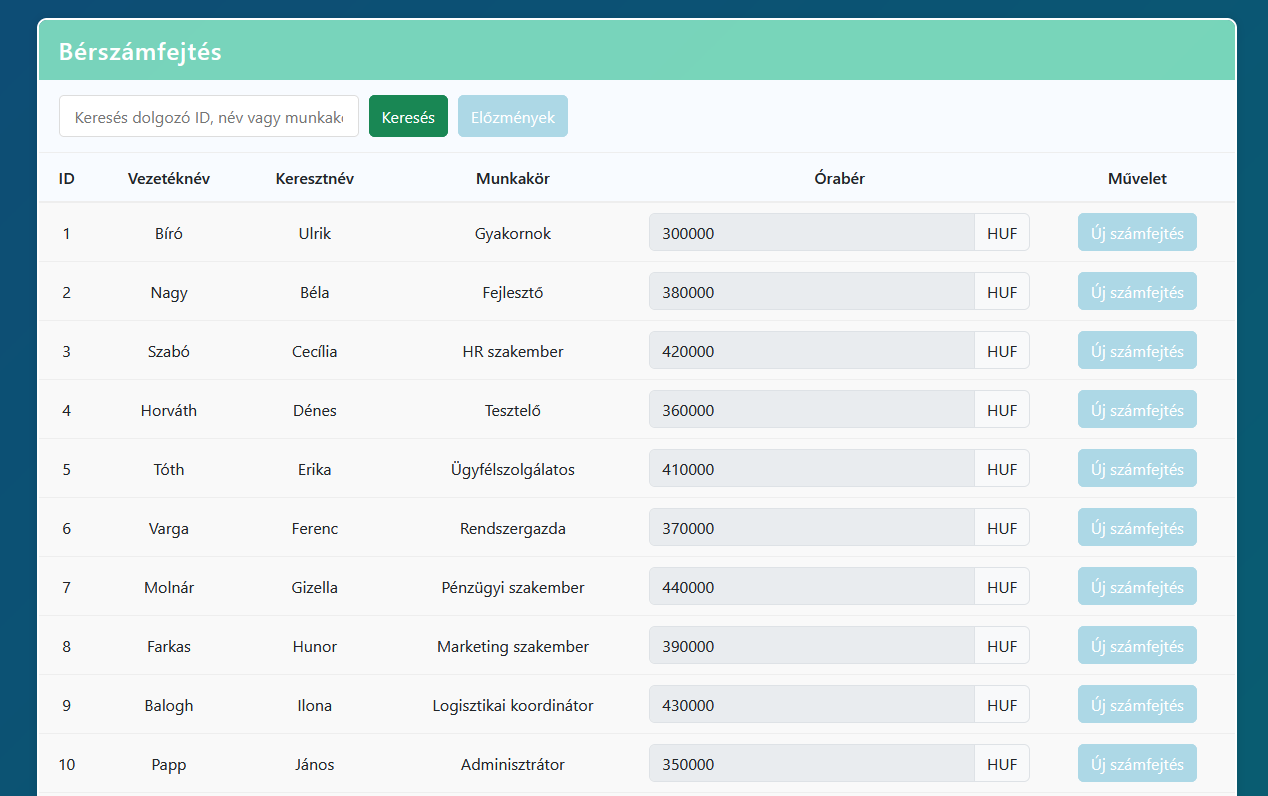
* **Bérszámfejtés:**

*A* ***bérszámfejtés*** *oldalon a felhasználó részletes csekkolási adatokat kérhet le az egyes dolgozókról és fizetést számolhat. Az oldal lehetővé teszi, hogy a dolgozók adatai alapján keresést végezzen a* ***keresés*** *gombbal.*

*Az* ***órabér mezőben*** *megjelenik a dolgozó* ***alapbére****, amely alapvető információt nyújt a fizetéséről. A táblázatban pedig további adatok láthatók, például:*

* ***Dolgozó ID****,*
* ***Vezetéknév****,*
* ***Keresztnév****,*
* ***Munkakör****,*
* ***Művelet****.*

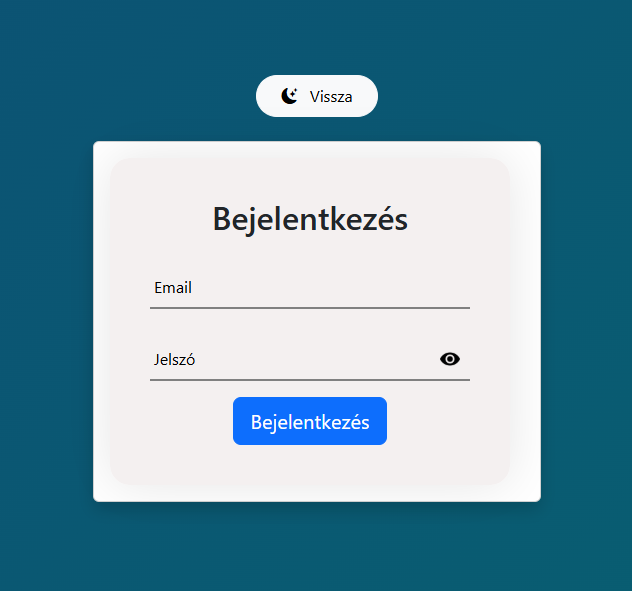
*Az* ***Új számfejtés*** *gombra kattintva még részletesebb információk érhetők el, különösen az időbeli adatok tekintetében. Ez lehetővé teszi, hogy a felhasználó kiszámolja a bér adatokat csekkolások szerint. A számolt adatokat az előzményekben lehet látni.*



* **Bejelentkezés:**

*A* ***bejelentkezés*** *oldalon a felhasználó bejelentkezhet már létező fiókjába.*

*Az ablakba megkell adni a felhasználó email címét és jelszavát, ha van találat akkor az oldal automatikusan átviszi az irányitópultra.*

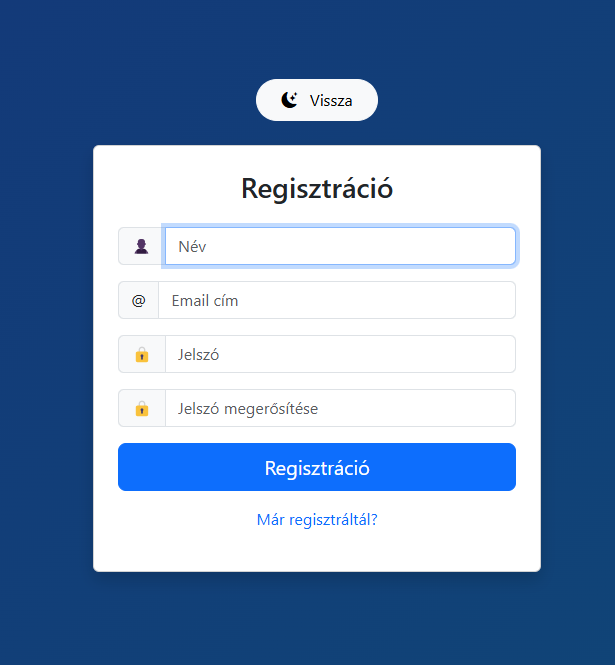


* **Regisztráció:**

*A* ***regisztráció oldalon*** *a felhasználó létrehozhatja a saját profilját, amelyet később a szoftveren belül használhat. A regisztrációhoz a következő adatok megadása szükséges:*

* ***Felhasználónév,***
* ***Email cím,***
* ***Jelszó.***

*A regisztráció sikeres befejezése után a felhasználó bejelentkezhet és használhatja a szoftvert a fiókjával.*



* **Kamera:**

*A* ***kamera oldalon*** *a felhasználó* ***Qrcode*** *beolvasásával tud becsekkolni. Az oldal először engedélyt kér a kamera használatára, és annak jóváhagyása után élő képet jelenít meg a felhasználó kamerájáról. Ezután a rendszer automatikusan felismeri és beolvassa a Qrcode-ot, lehetővé téve a sikeres becsekkolást.*



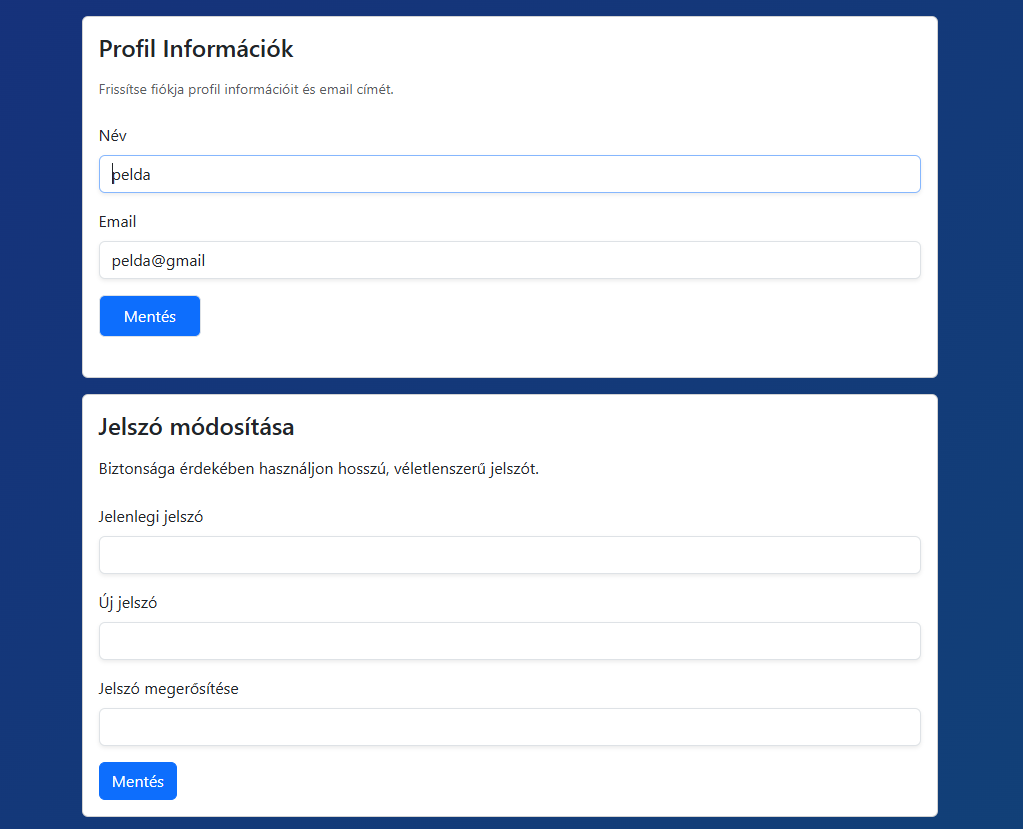
* **Hiba:**

*Ha* ***a hibaoldal*** *megjelenik, az általában azt jelenti, hogy a szerver nem tudta feldolgozni a kérést, vagy az érvénytelen volt.*



* **Profil:**

*A* ***profil*** *oldalon a felhasználó megváltoztathatja a felhasználó nevét, email címét vagy jelszavát, vagy akár a fiókját törölheti.*



**Fejlesztői dokumentáció**

* **Bevezetés**

**A rendszer célja:**

A Comproller egy vállalatirányítási rendszer, amely kisebb és nagyobb vállalatok számára lett kifejlesztve. A rendszer elsődleges célja a vállalati folyamatok digitalizálása, a humánerőforrás-kezelés optimalizálása és a munkaidő-nyilvántartás hatékony megvalósítása.

* **A szoftver célközönsége**

-Kis- és középvállalkozások

-Nagyvállalatok

-HR osztályok

-Menedzsment

* **Fejlesztői Dokumentáció**
* **Szoftver követelmények**

-Bármely modern webböngésző (Chrome, Firefox, Safari, Edge legfrissebb verziói)

-JavaScript engedélyezése

* **Felépítés**

**Frontend**

-HTML: Az oldalak alap épitő eleme.

-CSS: A kinézethez mindenhol használatos nyelv.

-Bootstrap: Ugyanaz mint a CSS, mindenhol használatos stílus nyelv.

-JavaScript: Az oldal alap funkcióihoz és egyéb hibakezelésre használatos nyelv.

**Backend**

-MySQL: Az adatbázis megvalósításához használt nyelv.

-PHP: Adatbázisba az adatok elmentéséhez és lekérdezéséhez használt nyelv.

-Laravel: Ugyanúgy mint a PHP, az adatok lekéréséhez és megjelenítéséhez használt nyelv.

-JSON: A Frontend és Backend közötti adatcseréhez használt nyelv.

* **Tervezés**

Hitelesítés:

Az oldalba bekel regisztrálni, így csak hitelesített felhasználók férhetnek hozzá.

*Átláthatóság*:

A szoftverbe több fajta módon vizsgálhatunk dolgozókat, így átláthatóbb a jelenleg kezelt cég.

*Egyszerűség*:

A weboldal felhasználóbarát, letisztult dizájnnal rendelkezik, ami gyors és egyszerű navigációt tesz lehetővé.

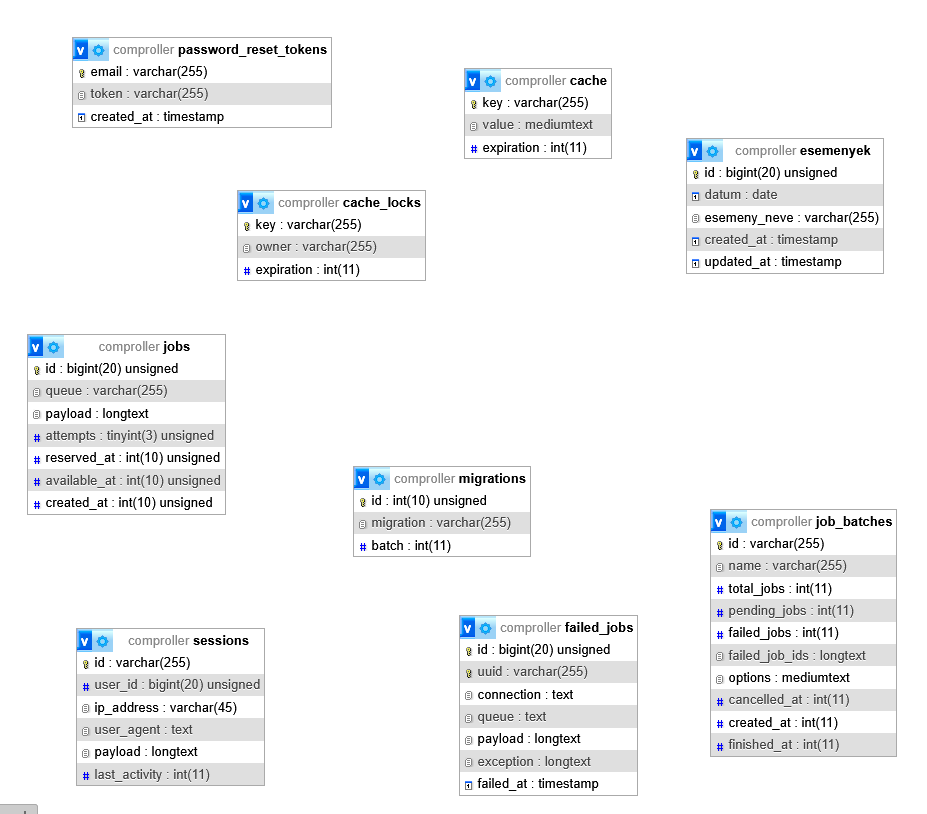
*Reszponzivitás*:

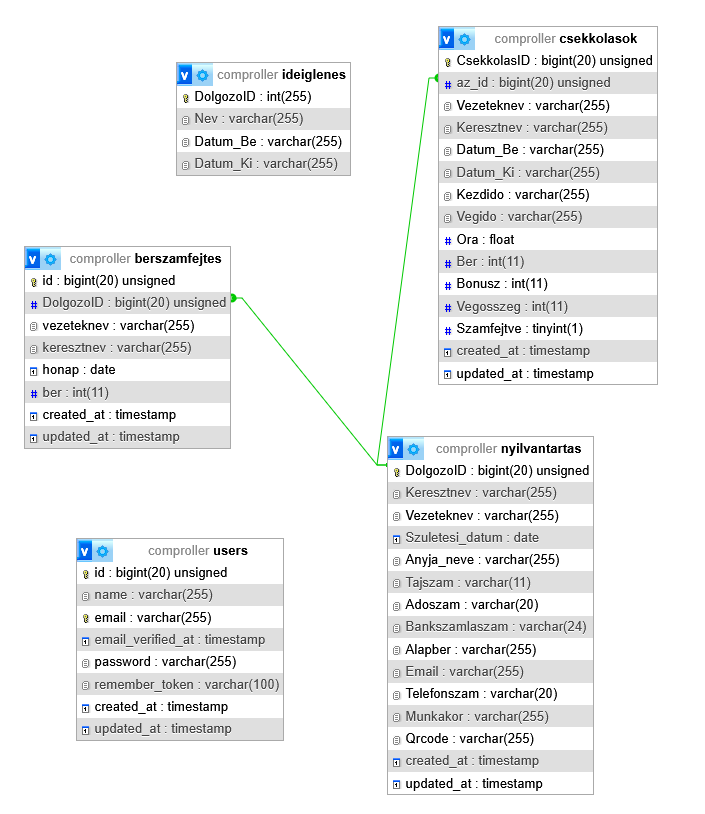
A weboldalba van reszponzivítás. Különböző

méretekhez igazítva is megállja a helyét.

*Gyorsaság*:

Az oldal optimalizálva van gyors betöltési időkre, így nem kell aggódni hogy az adatok elérése időigényes.

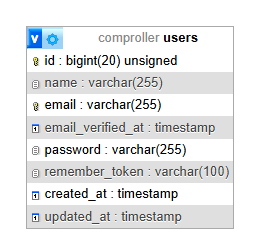
* **Az adatbázis szerkezete**



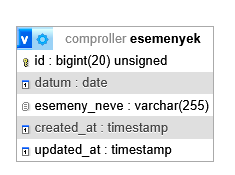
* **Használt táblák**
* **users**

A users táblában találhatóak az elmentett fiókok.

Ezt a táblát már a bejelentkezésnél használjuk, így kulcsfontosságú szerepe van.

* **id**: A fiók azonosítója. Autómatikusan növekszik minden új fióknál.
* **name**: A fiók felhasználó neve, megjelenik a nav-ban.
* **email**: A fiók email címe, szükséges regisztrációnál és bejelentkezésnél.
* **email\_verified\_at**: Az email cím megőrsítésének dátuma.
* **password**: A fiók jelszava, ezt kell bejelentkezéskor megadni. remember\_token: A fiókra token azonosító.
* **created\_at**: Fiók készitésének időpontja.
* **updated\_at**: Fiók frissítésének időpontja.
* **esemenyek**

A tábla az események oldalhoz.Ez felel azért hogy az új események el legyenek tárolva és később megjeleníthetőek legyenek.

* **id**: Az esemény azonosítója. Autómatikusan növekszik minden új eseménynél.
* **datum**: Az esemény mentett dátuma.
* **esemeny\_neve**: Az eseménynek a mentett neve.
* **created\_at**: Az esemény elkészítésének időpontja.
* **updated\_at**: Az esemény frissítésének dátuma. 
* **Nyilvantartás:**

A nyilvántartás egy kulcsfontosságú tábla,hiszen ebbe mentjük az újonnan felvett dolgozók adatait.

* **DolgozoID**: A dolgozó azonosítója, minden új dolgozó esetébe növekszik.
* **Keresztnev**: A dolgozó keresztneve.
* **Vezeteknev**: A dolgozó vezetékneve.
* **Szuletesi\_datum**: A dolgozó születési dátuma.
* **Anyja\_neve**: A dolgozónak az anyja neve.
* **Tajszam**: A dolgozó tajszáma.
* **Adoszam**: A dolgozó adószáma.
* **Bankszamlaszam**: A dolgozó bankszámlaszáma.
* **Alapber**: A dolgozó alapbére.
* **Email**: A dolgozó email címe.
* **Qrcode**: Egy egyedi qrcode a dolgozónak amellyel becsekkolhat a kamera oldalon.
* **created\_at**: Dolgozó hozzáadásának ideje.
* **updated\_at**: Dolgozó adatainak frissítési ideje.



* **csekkolasok**

A csekkolások tábla a munkaidő

és bérszámfejtés

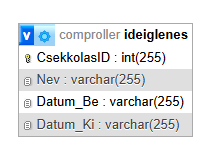
használatához szükséges.

Ehhez adjuk hozzá a ki és be

csekkolási időket, valamit

az órabért ebből számoljuk.

* **CsekkolasID:** A csekkolás azonosítója, autómatikusan növekszik.
* **Vezeteknev**: A dolgozó vezetékneve.
* **Keresztnev**: A dolgozó keresztneve.
* **Datum\_Be**: A dolgozó becsekkolásának ideje dátum formába.
* **Datum\_Ki**: A dolgozó kicsekkolásának ideje dátum formába.
* **Kezdido**: A dolgozó munkaidejének kezdete órás formába.
* **Vegido**: A dolgozó munkaidejének vége órás formába.
* **Ora**: Ledolgozott órák száma.
* **Ber**: A dolgozó bére.
* **Bonusz**: A dolgozó bónuszának száma.
* **Vegosszeg**: A bér és a bónuszból számolt összeg.
* **Szamfejtve**: Egy igaz/hamis érték a számfejtés ellenőrzésére.
* **created\_at**: A csekkolás készítésének ideje.
* **updated\_at**: A csekkolás frissítésének ideje.



* **Ideiglenes**

Az ideiglenes tábla tárolja a

munkaidőben keresett adatokat,

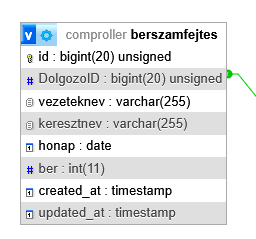
majd ebből jelenítjük meg őket.

**CsekkolasID**: A csekkolás azonosítója.

**Nev**: A dolgozó teljes neve.

**Datum\_Be**: Becsekkolási dátum.

**Datum\_Ki**: Kicsekkolási dátum.



* **berszamfejtes**

A bérszámfejtés tábla

használatos a bérszámfejtés

funkciók megvalósításához.

**id**: A bérszámfejtésnek egyedi ID.

**DolgozoID**: A dolgozó ID.

**vezeteknev**: A dolgozó vezetékneve.

**keresztnev**: A dolgozó keresztneve.

**honap**: A számfejtés hónapja.

**ber**: A számolt bér.

**created\_at**: Készités időpontja.

**updated\_at**: A frissítés időpontja.

***Tesztelői dokumentáció***

## Böngésző kompatibilitás

Az oldal a következő böngészőkkel lett letesztelve:

* Google Chrome
* Firefox
* Microsoft Edge

## Fejlesztési lehetőségek

### Keresés/Szűrés nyilvántartásban

* Mint bérszámfejtésben, legyen ott is egy mező, amin belül kereshetünk dolgozókat.

### Bónusz számlálása manuális oldalról

* A munkaidő manuális részén megadott bónuszt működő képessé tenni, mert jelen állapotban nem automatikus a számolás.

### Céges adatok hozzáadása

* A felhasználó bejelentkezés után megadhatja a céges adatait (azaz nem csak a dolgozókat), így azt is tudná kezelni a felületen.

### Kapcsolatok felfejlesztése

* Több opció hozzáadása a visszajelzés küldésénél (pl. a tárgy kibővítése).
* Akár bejelentkezve visszajelzést küldeni.

### Esemény címkék

* Az eltárolt eseménynek a dátumához megjelenhetne egy címke, ami jelzi az aznapi esemény nevét.

### Keresési mezők felfejlesztése

* A keresési mezőkbe már meglévő emberekről adatokat dobhatna fel a program, amit használhatunk. Ez könnyítené a keresést az adatbázisban.

### Oldal frissítések és ismétlődés

* Néhány oldal fejlesztésre szorúl, hiszen nem jól frissít új adat rögzítése esetén. Valamint van olyan oldal is, ahol az adatok többszöri hozzáadás esetén nem szűrődnek ki.

### Google Calendar API integráció

* A fejlesztés célja egy API-val működő naptármodul létrehozása az események oldalán, amely lehetővé teszi, hogy a felhasználók a programon kívül is hozzáférjenek saját teendőikhez.
* Az integráció során a felhasználók megtekinthetik a Google-fiókjukban létrehozott naptárakat, valamint szinkronizált módon kezelhetik eseményeiket mind az alkalmazáson belül, mind a Google Calendar felületén.

### Több lekezelés a nyilvántartás modalba

* A nyilvántartás modalba nem minden van részletesen lekezelve így fejlesztésre szorúl.

### Automatizálás: Bérlap készítése és dolgozói számfejtések egyéni előzményei

* A fejlesztés célja a manuális munkaidő-rögzítés esetén is automatizált bér- és bónuszszámítás bevezetése, amely a dolgozók bérszámfejtését gyorsabbá és pontosabbá teszi. Munkaidő manuális rögzítésekor a rendszer automatikusan kiszámolja a dolgozó bérét és esetleges bónuszait a megadott adatok alapján.
* Bérlap generálása az egyedi számfejtési adatokból, amelyet a dolgozó korábbi havi vagy heti teljesítménye alapján készít a rendszer.
* Dolgozónként egyéni bér- és számfejtési előzménylista megtekintése a felületen.

### Élő chat funkció

* A fejlesztés célja egy valós idejű (élő) chat funkció beépítése a programba, amely lehetővé teszi, hogy a felhasználók különböző modulokból közvetlenül kommunikálhassanak egymással.

### Pénzügyi modul és bérszámfejtés integrálása

* A fejlesztés célja, hogy a bérszámfejtett dolgozók adatai automatikusan továbbkerüljenek a pénzügyi modulba, így a felhasználók könnyedén kifizethetik az adott dolgozókat, vagy akár az összes számfejtett dolgozót egy tranzakcióval. Ez az integráció biztosítja, hogy a bérkifizetések és az adózási előírások zökkenőmentesen kezelhetők, miközben az automatizált fizetési rendszerek (például Stripe vagy PayPal) gondoskodnak a gyors és biztonságos kifizetésekről.

### Nettó bér megadása a dolgozónak a zökkenőmentes folyamatokhoz

* A nettó bér megadása és kezelése kulcsfontosságú a bérszámfejtési folyamatok zökkenőmentessége szempontjából. A rendszer lehetőséget biztosít arra, hogy a felhasználók könnyedén megadhassák a dolgozók nettó bérét, miközben az összes szükséges levonás és adó automatikusan kiszámításra kerül. Az automatizált bérszámfejtési folyamatok segítik elkerülni a hibákat, biztosítják a pontos kifizetéseket és megfelelnek a jogszabályi előírásoknak.

## *Tesztek*

### Bejelentkezési form tesztelése

**Eredménye:**

* A mezők hibakezelése működik, hibás adatok esetében törli a beírt adatokat, sikeres bejelentkezés átvisz az irányítópultra.

### Kapcsolatok formok tesztelése

**Eredménye:**

* A mezők hibakezelése működik, hiányzó mezők esetében nem küld el adatot, viszont az elküldött adatok elé beszúr egy vesszőt.

### Legutóbbi események és utoljára felvitt dolgozók táblázatainak tesztelése

**Eredménye:**

* Az események és dolgozók sikeresen lekérődnek új adat esetén.

### Események hozzáadása

**Eredménye:**

* Az események hozzáadásánál a mezőbe beírt adatok megmaradnak, ha új modalt nyitunk. Mentéskor elmentődnek az adatok az adatbázisba, viszont nem azonnal jelennek meg oldal frissítés esetén se.

### Események törlése

**Eredménye:**

* Az események törlődnek az adatbázisból, viszont nem azonnal jelenik meg a változás (frissíteni kell az oldalt).

### Dolgozó hozzáadása

**Eredménye:**

* A születési dátum hibakezelése működik (azaz nincs a jövőbe), továbbá a többi mezőben is van hibakezelés (hossz, csak számok, név részek, email (frontend és backend), stb).

### Dolgozó adatainak frissítése

**Eredménye:**

* A dolgozó adatainak frissítése minden esetbe működik és a mezők hibakezelése is működik.

### Qrcode olvasás

**Eredménye:**

* A Qrcode olvasásakor sikeresen becsekkol a dolgozó.

### Manuális csekkolás.

**Eredménye:**

* A manuális csekkolás működik, ha a nyilvántartásba a dolgozó megtalálható akkor az adatok várt módon elmentődnek, ha nem akkor csak táblázatba jelennek meg. Az oldalt frissíteni kell hogy a csekkolási táblázatba megjelenjenek az adatok. A hibakezelés a mezőkön működik.

### A bejelentkezés tesztelése

**Eredménye:**

* A bejelentkezés tesztelve volt, sikeresen be enged lépni ha érvényesek az adatok, máshogy jelzi hogy melyik mezőbe van hiba. A jelszó megjelenítése és elrejtése szintén működik.

### A kijelentkezés gomb működésének tesztelése

**Eredménye:**

* Gomb sikeresen kijelentkeztet, visszadob az index oldalra.

### Csekkolások lekérdezése

**Eredménye:**

* A csekkolások lekérdezésének funkciója működik, az adott dátum hosszon belül és név szerint lekérdezi az ahhoz tartozó dolgozó csekkolási adatait. Viszont az oldalt frissíteni kell hogy az adatok megjelenjenek vagy törlés esetén eltűnjenek. A törlés gomb szükség esetén működik. Ha a felhasználó többször lekérdezi ugyanazokat az adatokat akkor azok is hozzáadódnak a táblázathoz.

### Bérszámfejtésen a keresés tesztelése

**Eredménye:**

* Az oldalon a keresési mező tesztelve volt, sikeresen talál adatokat keresztnév, vezetéknév és id meg munkakör alapján, viszont teljes név megadása esetén nem. Az új számfejtés gomb sikeresen talál adatokat és az előzményeken belül kiszámolva el is tárolja.

### A regisztráció tesztelése

**Eredménye:**

* A születési dátum hibakezelése működik (azaz nincs a jövőbe), továbbá a többi mezőben is van hibakezelés (hossz, csak számok, név részek, email (frontend és backend), stb).

## *Talált hibák*

### Főoldal

* Nem működik a favicon. (Javítva)

### Kapcsolat

* Üzenet elküldésekor egy vessző van az üzenetbe.

### Bejelentkezés

* A bejelentkezési oldalon nem működik a jelszó elrejtése és megjelenítése. (Javítva)

### Irányítópult

* A kördiagramon véletlenszerűek az értékek. (Javítva)
* Az események és újonnan felvitt dolgozók nincsenek le korlátozva, ha túl sok dolgozó van. (Javítva)

### Nyilvántartás

* A nyilvántartás listája nincs le korlátozva, ha túl sok dolgozó van. (Javítva)
* Az oldal túl hamar küldi el az információkat validáció nélkül. (Javítva)
* A dátumot meg lehet adni a jövőbe. (Javítva)
* A QR-kód nem jelenik meg vagy tűnik el bevitel változásakor. (Javítva)
* Nincs ellenőrzés szám típusok esetén. (Javítva)

### Események

* Az oldal nem mindig frissül jól.
* Az adatok nem mentődnek az adatbázisba. (Javítva)
* A mező modal nem jelenik meg. (Javítva)
* A dátum nem változik gombnyomás esetén. (Javítva)
* A törlés gomb nem jelent meg vagy nem törölt. (Javítva)

### Munkaidő

* A csekkolások nem jelennek meg a listában. (Javítva)
* A manuális rész nem menti el az adatokat. (Javítva)
* A mezők kezelése nem működik, azaz lehet dátumot megadni a jövőbe, szám mezőkbe betűk stb. (Javítva)
* Név nélkül el lehet küldeni az adatokat. (Javítva)
* A lekérdezés rész nem talál adatokat. (Javítva)
* A lekérdezés rész csak egy dátumot vesz figyelembe és az alapján keres adatot. (Javítva)
* A lekérdezés rész nem veszi figyelembe a második dátumot. (Javítva)
* A törlés gomb nem csinál semmit. (Javítva)
* A táblázatba többször le lehet kérni ugyanazt az értéket.

### Bérszámfejtés

* A keresés mező nem talált adatot. (Javítva)
* A keresési mezőbe nem lehet teljes név szerint keresni.
* Új számfejtéskor csak az oldal frissül. (Javítva)
* Új számfejtés csak az első ID-t számolja ki. (Javítva)
* Az előzményekbe nem jelenik meg adat. (Javítva)

### Kamera

* A kamera nem ad vissza helyes értéket QR-kód olvasása esetén. (Javítva)
* A csekkolás nem mentődik el az adatbázisba. (Javítva)