Szoftverfejlesztő és tesztelő tanfolyam 24\_25 (389/24T; 390/24T)  
  
Projektfeladat: Projektmenedzsment

Grósz Ferenc Dániel

Bevezetés

Eredetileg a cél egy olyan asztali alkalmazás létrehozása volt, amely több platformon (Windows, Android) is kezelhető, és amelynek segítségével gyorsan és egyszerűen ki lehet állítani a nap végén a végzett munkát igazoló munkalapokat.

A téma komolysága és nagyságrendje visszatartotta a jelentkezőket a projektben való részvételtől.

Eközben a profilváltozás miatt a vállalat a néhány napig tartó kis munkákról áttért a párhuzamosan futó hosszú távú projektekre.

Így az egyéni munkalapok létrehozása háttérbe szorult, és helyette olyan platformra volt szükség, amelyhez minden alkalmazott hozzáférhet, és amely megkönnyíti az információáramlást.

A tervezés során fontos szempont volt, hogy a rendszer használatához ne legyen szükség informatikai szakértelemre, így még a képzetlen munkavállalók is gyorsan és könnyedén megtalálhassák a keresett információkat.

A vállalat vezetőjével és informatikai szakértőjével folytatott konzultációt és alapos mérlegelést követően egy PHP/JavaScript webalapú megoldás mellett döntöttünk, amely nem igényel semmilyen telepítést a platformokra.

A tesztelés során az Edge, a Chrome és a Firefox asztali és Android verzióit használtuk.

1. Fejlesztői dokumentáció

Az elkészült platformhoz PHP/Javascript környezetet választottunk, PhpMyadmin adatbáziskezelővel.

A Visual Studio Code fejlesztőkörnyezetben dolgoztunk.

Teljesen zárt rendszer, szigorú Adminisztrátori felügyelettel. Így nincs regisztrációs form. A munkavállalók a saját jelszavukat bármikor módosíthatják, elfelejtett jelszó esetén az adminisztrátor jogosult új jelszót adni a munkavállalónak.

Két fő hozzáférési oldal van: Admin és Dolgozo.

Mindkét hozzáférési módnál az oldal egyformán épül fel: Menü és tartalom.

A Menü reszponzív, kis felbontás esetén lenyíló menüvé válik. A tartalom Switch/Case szerkezettel include-olva töltbődik be, a get-ben megadott paraméterek szerint.

Az oldalak legnagyobb része egyszerű űrlapkitöltő felület (projektek, dolgozok, jármuvek, szabadság) backend visszaellenőrzéssel, ahol meg lett oldva az adatbevitel és módosítás az adatbáziskezelőben.

A Beosztások oldal egyszerű lekérdezés a bejelentkezett felhasználóra vonatkozó adatokra, heti lebontásban.

A Nap szervezés menü oldja meg dinamikusan magát a fő funkciót.

A lapon generálásra kerül 3 div (Szabadság,Dolgozók,Járművek) statikusan, illetve minden az adott napon nyitott projekt számára egy külön Div.

Az oldal betöltésekor lekérdezésre kerül az adatbázisból, hogy a dolgozók közül kik vannak beosztva már az adott napon. A szabadság táblában lévő dolgozók, akik az adott napra szabadságot kért, automatikusan belekerülnek a Szabadság Divbe, és kizárja őket a szervezésből. Aki nincs szabadságon és még nincs beosztva egy projektbe sem, azok automatikusan bekerülnek a Dolgozo Div-be. A Jarmuvek hasonló módon működnek, szabadság nélkül.

Az engedélyezett objektumokat (Dolgozók és Jármű div-ek tagjait) Drag&Drop módszerrel lehet hozzárendelni egy projekthez. Az adatkapcsolatot egy API oldja meg backend oldalon, ez biztosítja az adatbázis kapcsolatot is. A Projektbe helyezett objektumok ezután nem mozgathatók.

A projektből való objektum törlése, az objektumra történő dupla kattintással oldható meg.

A projektek adott naphoz tartozó Leírás mező módosítás esetén automatikusan frissül (1 másodperc késleltetéssel, hogy ne minden karakter után történjen adatküldés).

Részlet a Javascript kódból:

function dragStart€{

    if(!e.target || !e.target.dataset) return;

    e.dataTransfer.setData(’application/json’, JSON.stringify({

        id: e.target.dataset.id,

        tipus: e.target.dataset.tipus,

        nev: e.target.textContent

    }));

}

function initDrag(){

    document.querySelectorAll(’.item’).forEach(item => {

        try { item.removeEventListener(’dragstart’, dragStart); } catch€{}

        if (item.getAttribute(’draggable’) === ’true’) {

            item.addEventListener(’dragstart’, dragStart);

        }

        item.ondblclick = null;

        if (item.closest(’.project’)) {

            item.ondblclick = function€ {

                const parentProj = item.closest(’.project’);

                if (!parentProj) return;

                const projekt\_nap\_id = parentProj.dataset.id;

                if (!projekt\_nap\_id) return;

                let tipus = parentProj.querySelector(’.dolgozok’).contains(item) ? ’dolgozo’ : ’jarmu’;

                fetch(’api.php’, {

                    method: ’POST’,

                    headers: {’Content-Type’: ’application/json’},

                    body: JSON.stringify({

                        action:’delete’,

                        projekt\_nap\_id,

                        tipus,

                        nev: item.textContent

                    })

                })

                .then(r => r.json())

                .then(res => {

                    if(res.success){

                        item.remove();

                        updateUnassigned();

                    } else {

                        alert(res.error || ’Hiba történt!’);

                    }

                })

                .catch(err => {

                    console.error(’delete hiba:’, err);

                    alert(’Hálózati hiba történt.’);

                });

            };

        }

    });

    document.querySelectorAll(’.project’).forEach(proj => {

        proj.ondragover = e => e.preventDefault();

        proj.ondrop = e => {

            e.preventDefault();

            let data;

            try {

                data = JSON.parse(e.dataTransfer.getData(’application/json’));

            } catch (err) {

                console.error(’Hibás dataTransfer:’, err);

                return;

            }

            let projekt\_nap\_id = proj.dataset.id;

            const projekt\_id = proj.dataset.projektId;

            fetch(’api.php’, {

                method: ’POST’,

                headers: {’Content-Type’: ’application/json’},

                body: JSON.stringify({

                    projekt\_nap\_id,

                    projekt\_id,

                    tipus: data.tipus,

                    id: data.id,

                    datum: ’<?= $datum ?>’

                })

            })

            .then(r => r.json())

            .then(res => {

                if(res.projekt\_nap\_id){

                    proj.dataset.id = res.projekt\_nap\_id;

                }

                if(res.success || res.projekt\_nap\_id){

                    const targetDiv = proj.querySelector(’.’ + (data.tipus===’dolgozo’ ? ’dolgozok’ : ’jarmuvek’));

                    const d = document.createElement(’div’);

                    d.className = ’item’;

                    d.textContent = data.nev;

                    targetDiv.appendChild(d);

                    initDrag();

                    updateUnassigned();

                } else {

                    alert(res.error || ’Hiba történt!’);

                }

            })

            .catch(err => {

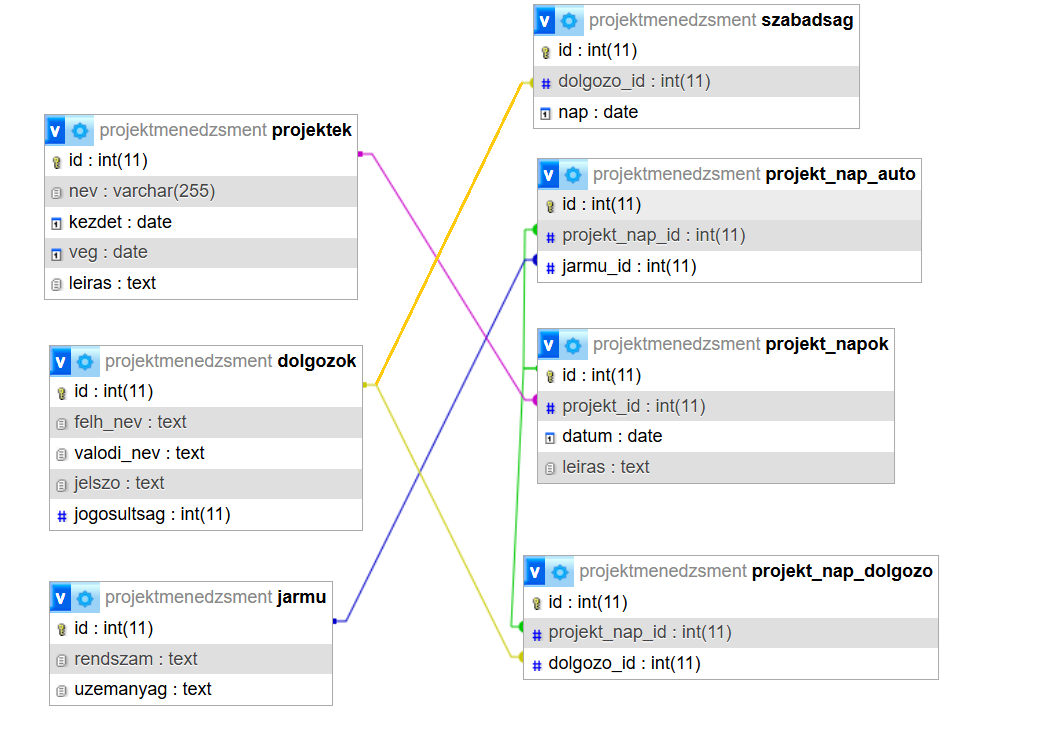
                console.error(’Hozzárendelés hiba:’, err);

                alert(’Hálózati hiba történt.’);

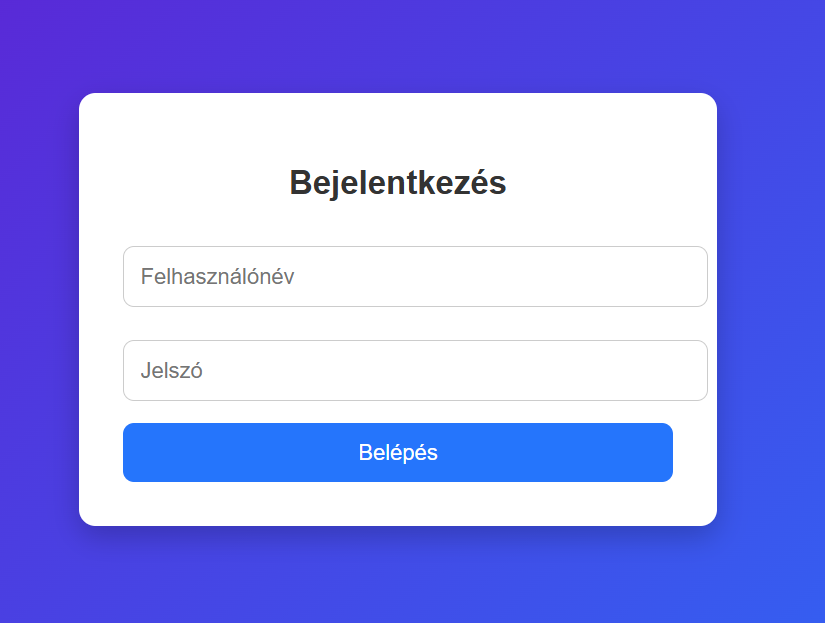
            });

        };

    });

Adatbázis kapcsolatok:

1. Felhasználói Dokumentáció

A bejelentkezési oldal Felhasználó név és jelszó alapján automatikusan eldönti, hogy Admin vagy Dolgozói felületre irányítja e a belépőt.

Admin felhasználónév: gfd

Admin jelszó:1234

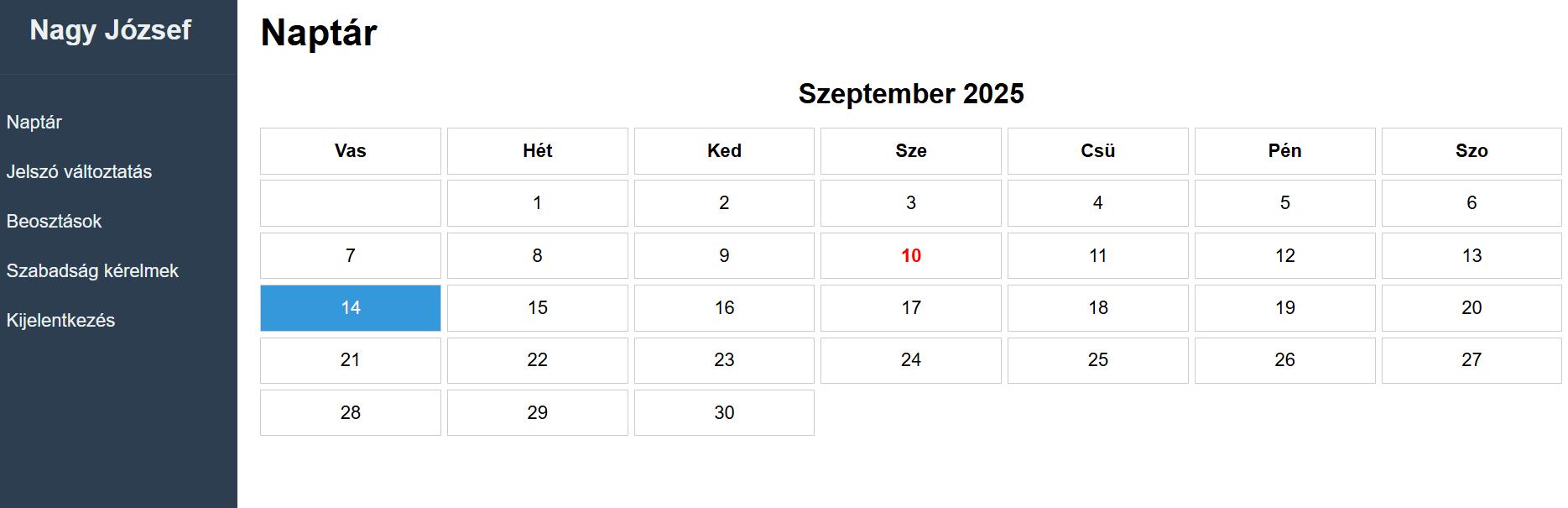
Dolgozói felhasználónév: joci

Dolgozói jelszó: joci

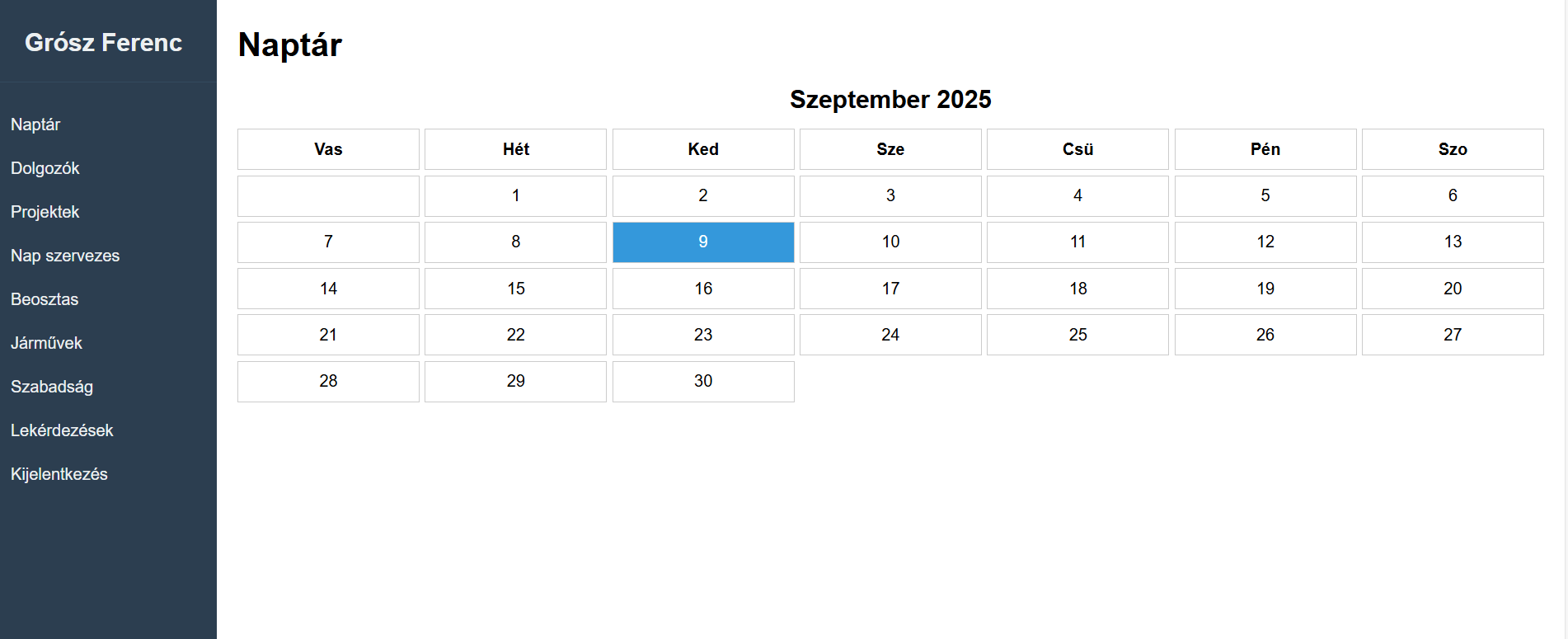
Nincs regisztrációs és elfelejtett jelszó lehetőség, mivel teljesen zárt hozzáférésű rendszer, így csak az Adminisztrátor adhat hozzáférést az oldalhoz. Ha a dolgozó elfelejti a jelszavát, szintén az Admin tud neki új jelszót adni.

A Nyitó oldalon mindkét hozzáférési lehetőségnél egy dinamikusan generált naptár jelenik meg, aminek a napjaira kattintva átirányít, az Admin-okat a Nap szervezésre, a dolgozókat az adott heti beosztásra.

Dolgozói oldal:

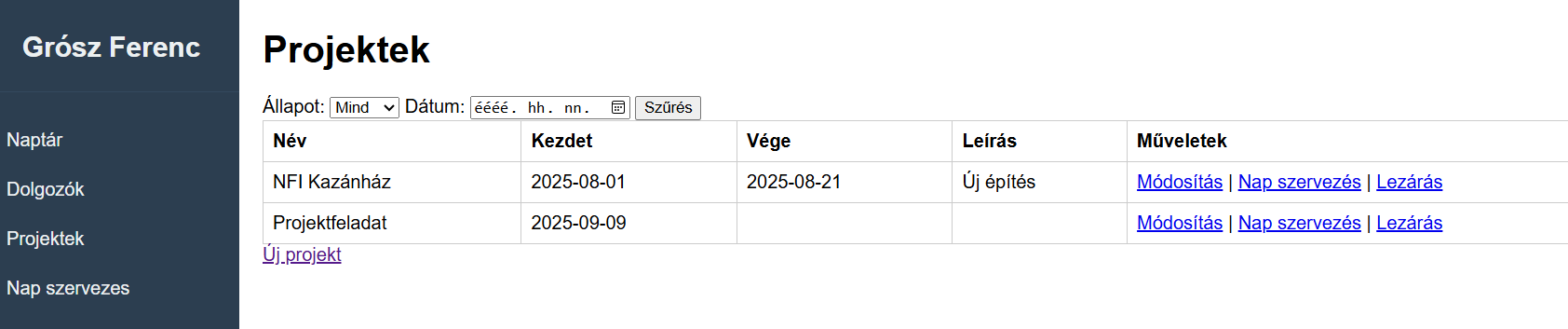


Admin oldal:

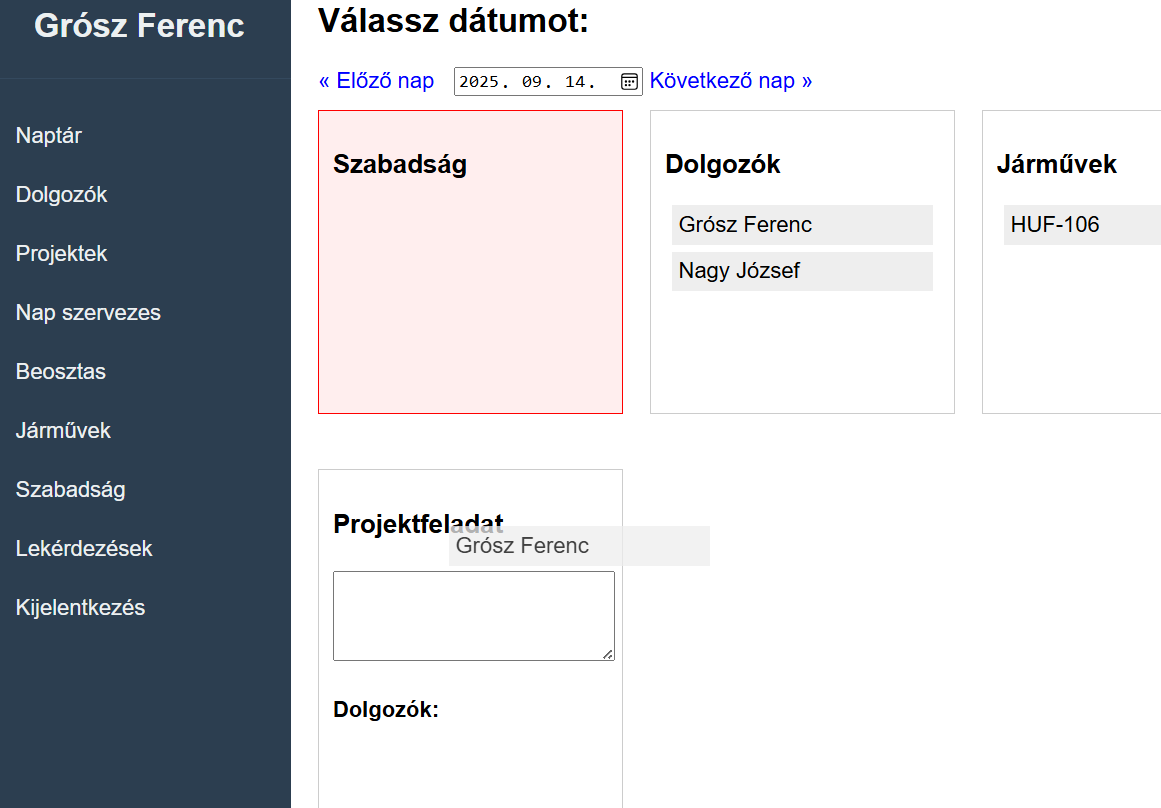


Űrlapkitöltő felületek:



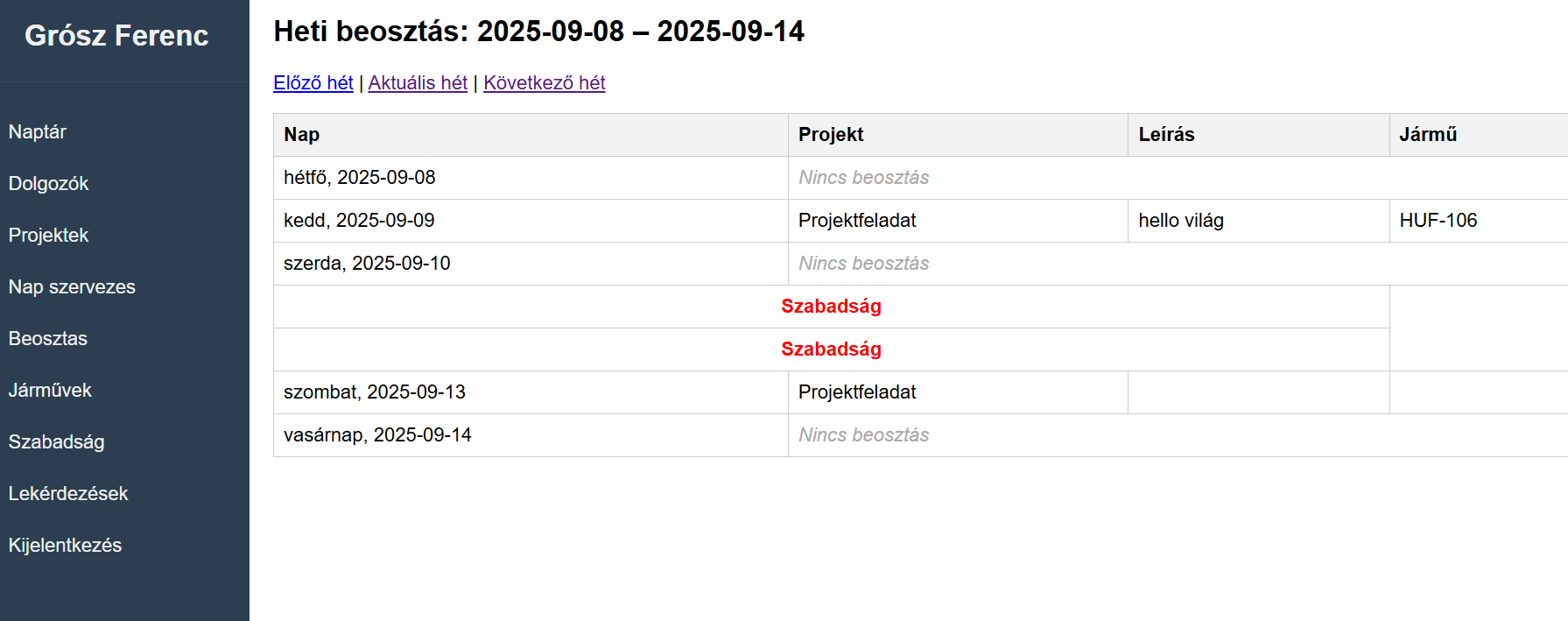




Nap szervezés:

A dolgozók és Járművek Drag&drop módszerrel rendelhetők a Projektekhez, majd Dupla kattintásssal törölhetőek a projektből.

Beosztások:



Az adott hétre vonatkozó beosztások a dolgozónak.

Lekérdezések:



A projekt adatait listázza ki, az összes munkával töltött napra vonatkozólag, illetve összesítve.

**3. Tesztelés**

**3.1 Tesztelési stratégia**

* **Cél:** Kisvállalkozás több projektre vonatkozó munka szervezése
* **Módszerek:**
  + Egységteszt
  + Integrációs teszt
  + Manuális funkcionális teszt
* **Tesztkörnyezet:** Windows 10 + XAMPP, Chrome/Edge böngésző, mobilnézet (Chrome DevTools).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tesztazonosító** | **Leírás** | **Lépések** | **Elvárt eredmény** | **Tényleges eredmény** | **Sikeres?** |
| T001 | Regisztráció helyes adatokkal | Megadom felhasználónevet, jelszót, megerősítem | Új felhasználó létrejön, bejelentkezik | ✅ Létrejött | ✅ |
| T002 | Bejelentkezés hibás jelszóval | Rossz jelszó beírása | Hibaüzenet jelenik meg | ✅ Megjelent | ✅ |
| T003 | Projekt létrehozása | Megadom a projekt adatait | Projekt mentődik DB-be | ✅ Létrejött | ✅ |
| T004 | Admin csak admin oldalt lát | Admin felhasználóval belépés | Látja: Az űrlap kitöltő oldalakat, Nap szervezést | ✅ Működik | ✅ |
| T005 | Dolgozói belépés | „Joci” felhasználó belépés | Csak a Dolgozói Menü látszik | ✅ Működik | ✅ |
| T006 | Mobil nézet menü | Telefonos méret szimulálva | Menü hamburgerben nyílik | ✅ Működik | ✅ |

**3.3 Hibajegyzék**

|  |  |  |  |
| --- | --- | --- | --- |
| **Hibaazonosító** | **Leírás** | **Állapot** | **Megjegyzés** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**3.4 Tesztelési jegyzőkönyv**

* **Tesztelő neve:** Grósz Ferenc
* **Dátum:** 2025. 09. 14.
* **Eredmény:** minden kritikus funkció tesztelve, hibákat nem talált.