## 

#### Páholy vizsgaremek dokumentáció

## https://cdn.discordapp.com/attachments/916014799312289803/1359621631139381429/ckik_logo1_5ddb5979c8.png?ex=67f825d1&is=67f6d451&hm=6fa65dbfa394768cdbab046b66c92fd0ff7fe5d781a8611ea4965d94e20e2c0e&=

Készítette:  
Agócs Dániel

Kovács Ákos

Havasi Alex Márton

# Páholy felhasználó dokumentáció

Tartalom

[Páholy felhasználó dokumentáció 2](#_Toc195254692)

[Program ismertetése 3](#_Toc195254693)

[Program indítása és futtatása 3](#_Toc195254694)

[Bejelentkező oldal: 7](#_Toc195254695)

[Az oldal felépítése 8](#_Toc195254696)

[Üzenetek felület 8](#_Toc195254697)

[Házi feladat felület 10](#_Toc195254698)

[Órarend felület 12](#_Toc195254699)

[Érdemjegyek felület 13](#_Toc195254700)

[Hiányzások felület 14](#_Toc195254701)

[Felhasználók kezelése felület 15](#_Toc195254702)

[Csoportok kezelése felület 16](#_Toc195254703)

[Program leállítása, program törlése 16](#_Toc195254704)

[Fejlesztői dokumentáció 17](#_Toc195254705)

[Frontend 17](#_Toc195254706)

[Projekt fájlrendszer elrendezése 17](#_Toc195254707)

[Kiegészítők 18](#_Toc195254708)

[Oldal felépítés 18](#_Toc195254709)

[Lekérdezések 18](#_Toc195254710)

[Adatok 19](#_Toc195254711)

[Fajták 19](#_Toc195254712)

[Visszajelzés 19](#_Toc195254713)

[API 20](#_Toc195254714)

[Projekt fájlrendszer elrendezése 20](#_Toc195254715)

[Kiegészítők: 20](#_Toc195254716)

[Végpont dokumentáció 20](#_Toc195254717)

[Adatbázis 21](#_Toc195254718)

[Táblák 21](#_Toc195254719)

[Adatbázis megtervezése 25](#_Toc195254720)

[Felhasználó rendszer 25](#_Toc195254721)

[Órák 25](#_Toc195254722)

[Tesztelés 26](#_Toc195254723)

[Hiányzás tesztek 26](#_Toc195254724)

[Jegy tesztek 28](#_Toc195254725)

[Rendszergazda funkció tesztek 29](#_Toc195254726)

[Feladat funkciók tesztelése 30](#_Toc195254727)

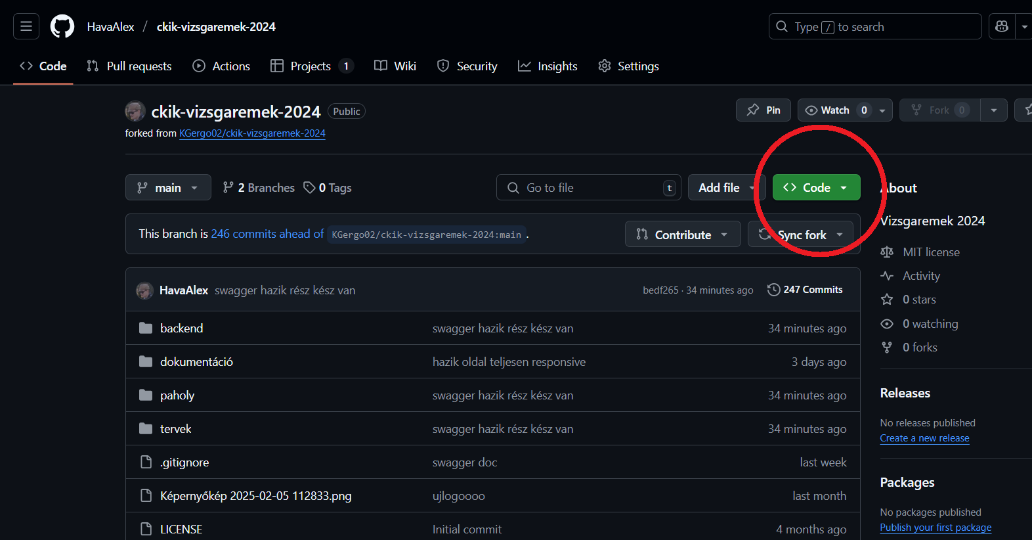
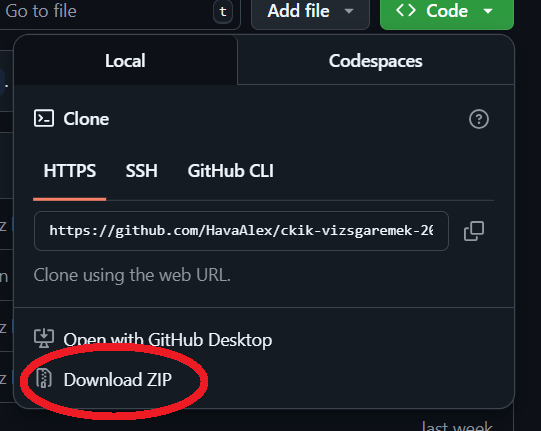
[Üzenet funkciók tesztelése 31](#_Toc195254728)

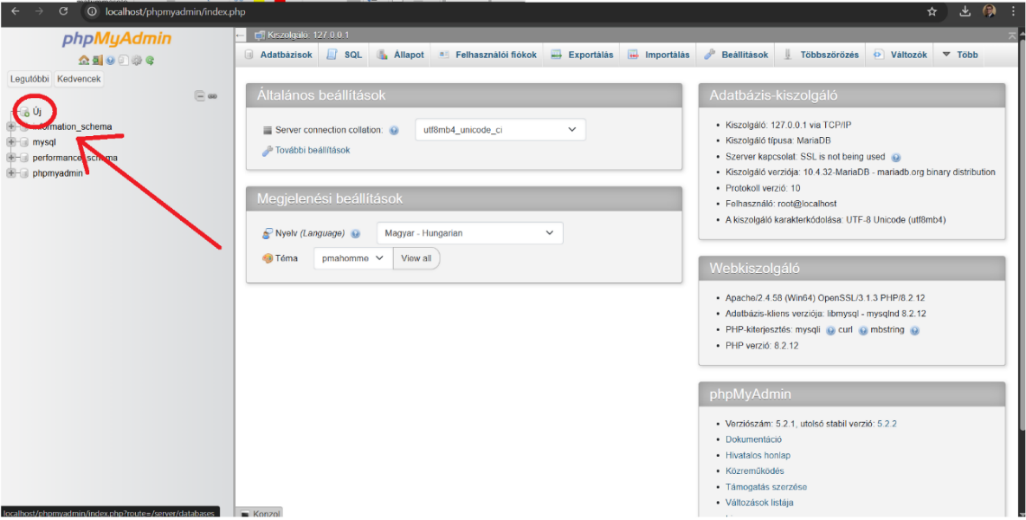
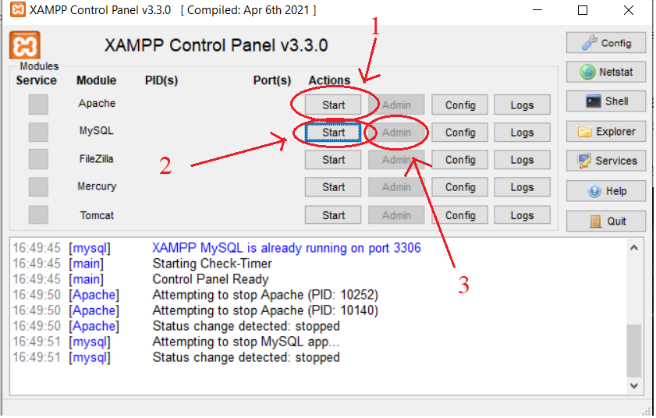
[Órarend tesztelése 32](#_Toc195254729)

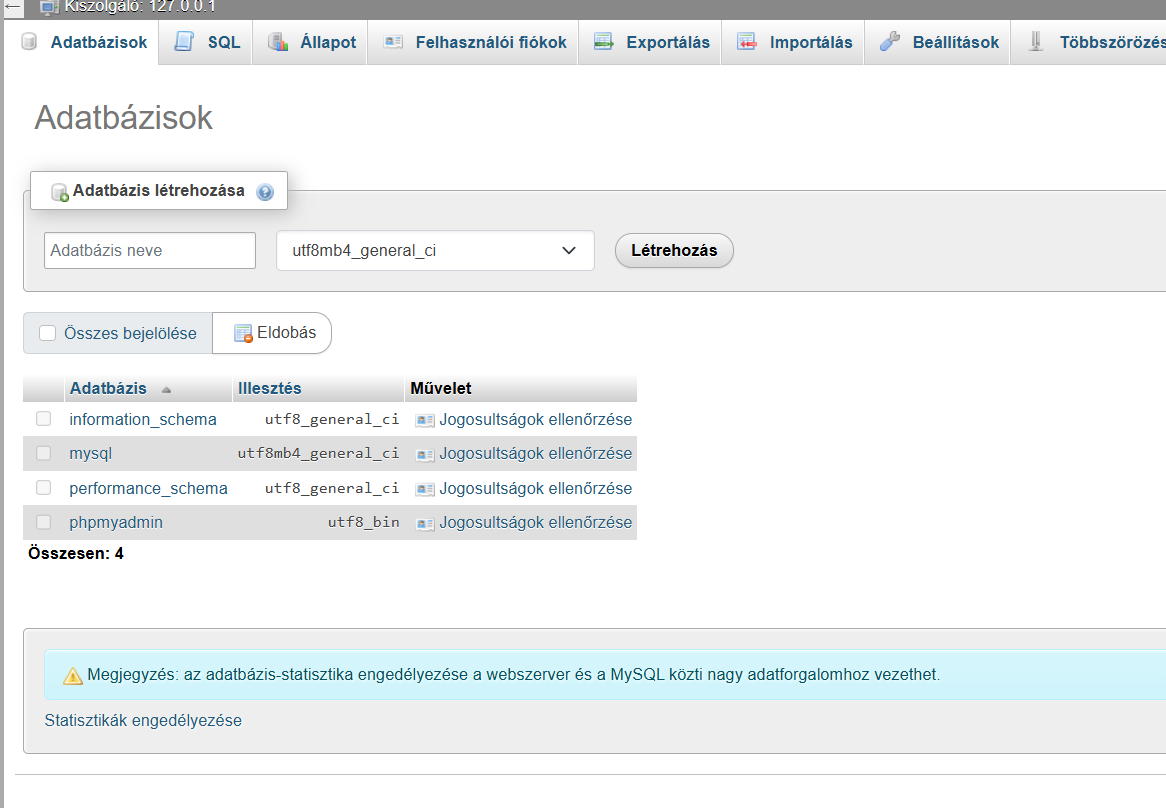
### Program ismertetése

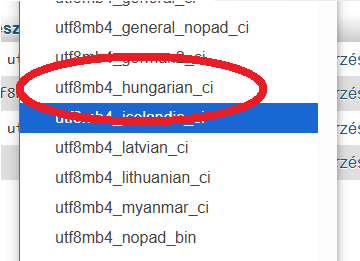
A Páholy egy digitális e-napló alkalmazás melyben a diákok, azok gondviselői és pedagógusok tudnak egymással dolgozni, kommunikálni. Ez a dokumentum bemutatja ennek a programnak a működését, futtatását, kezelését.

### Program indítása és futtatása

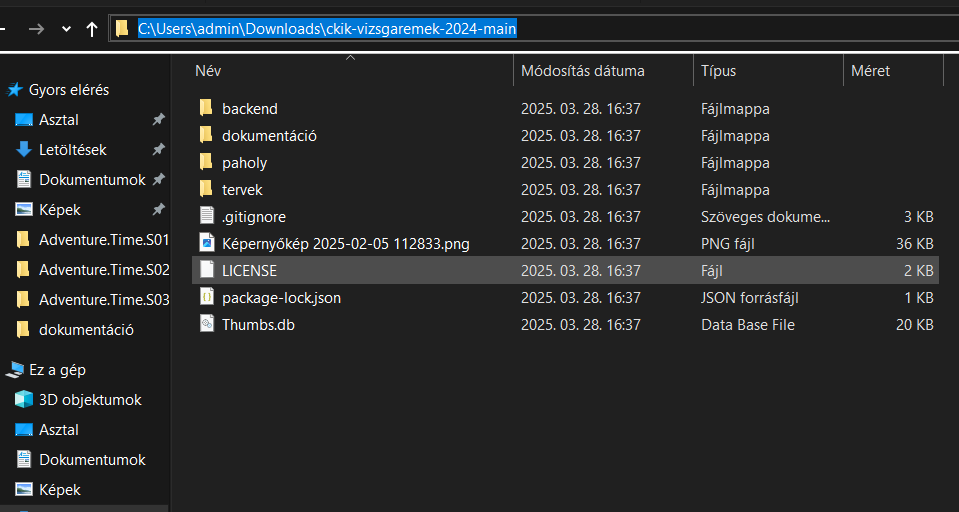
 A Páholy rendszer futtatásához szükséges az XAMPP illetve a node.js telepítése előzetesen. Amennyiben nem áll rendelkezésére, vagy esetleg elveszett a Páholy rendszer forráskódja, menjen a Páholy nyilvános Github oldalára (<https://github.com/HavaAlex/ckik-vizsgaremek-2024>) és itt nyomjon a zöld **code**gombra. Ennek hatására egy új menü jelenik meg, melyen a **Download ZIP** feliratú gombot kell megnyomni, hogy elinduljon a programot tartalmazó ZIP fájl letöltése. Miután a fájl letöltött, csomagolja ki egy tömörítőprogram segítségével. (pl.: WinRAR) Ezután helyezze a tőlünk kapott .env file-t a backend mappába (ammennyiben ez is elveszett, vegye fel velünk a kapcsolatot)

Amennyiben sikerrel járt vagy a forráskód eleve rendelkezésére állt, indítsa el az XAMPP alkalmazást. Itt az alábbi képen látható felületen nyomja meg a kijelölt gombokat a kijelölt sorrendben. (**1. Start, 2. Start, 3. Admin**) Ez megnyitja a phpMyAdmin oldalát, melyen Önnek a bal felső sarokban található **új** gombra kell nyomnia. Az itt megjelent felületen az „Adatbázis neve” feliratú beviteli mezőbe, írja, hogy „**paholy**”. A mellette lévő utf8mb4\_general\_ci feliratra kattintva, megjelenik egy legördülőmenü, melyben Önnek az utf8mb4\_hungarian\_ci opciót kell kiválasztania. Ez biztosítja, hogy a magyar ábécé különleges betűi is szerepelhessenek az adatbázisban. Miután ezeket kiválasztotta, kattintson a létrehozás gombra.



Itt látható a legördülő menü tartalma.

Az Importálás gombra kattintva új felület nyílik meg ahol az importálandó fájl menüpontban fel tudja tölteni a paholy.sql fájlt. Miután ez megtörtént, a középen alul található „Importálás” feliratú gombot nyomja meg. Ennek hatására az adatbázis feltöltődik példa felhasználókkal, csoportokkal, így lehetővé válik az oldal használata. A próbafelhasználók neveit és jelszavait a fejezet végén listázva találja meg. Ez létrehozza az adatbázist és példa adatokkal tölti fel.

 Először csomagolja ki a forráskodot tartalmazo forraskod.zip fájlt. Ezek után nyissa meg a forráskódot tartalmazó mappát és kattintson a fájl elérési útvonalát jelző szövegdobozra. Ennek hatására az kijelölődik. Itt a backspace gombbal törölje ki a meglévő szöveget, majd írja be a következőt: „cmd”. Ennek hatására megnyílik a parancssor. Fontos, hogy az oldal futattása alatt ne zárja be a parancssort. Itt az első elindítás esetén írja be a „cd backend” parancsot, majd az „npm i” illetve az „npm start” parancsokat. Amennyiben a programot nem először futtatja, a parancsok a következőek: „cd backend”, „npm start”.

Mindezek után nyisson meg egy új parancssort a Páholy program forráskódjának mappájában a fájl elérési útvonalára kattintva. Első indítás esetén az alábbi parancsokat írja be: „cd paholy”, „npm i”, „npm run build”, „npm run preview”. Ammennyiben nem elsőre futtatja a progarmot a következő kódot írja be: „cd paholy”, „npm run build”, „npm run preview”. Ezek után az egyetlen teendő, hogy az Ön által preferált böngészőprogramban az URL helyére írja be a <http://localhost:4173/> szöveget (vagy kattintson rá a dokumentációban a Ctrl-bal kattintás gombkombinációval).

Jelenleg az adatbázisban szerepel 2 osztály, 6 illetve 5 diákkal. Minden diákhoz tartozik legalább egy szülő és naponta 5 tanóra. Az adatbázisban szerepel 3 tanár és 1 rendszergazda. Mivel a Páholy gyakorlati iskolai használatra van tervezve, regisztrációra nincs lehetőség. Az új felhasználók, csoportok illetve tanórák létrehozása a rendszergazda feladata. Új felhasználok létrehozásakor generálódik egy 12 véletlenszerű karakterből álló jelszó. Ezt később megváltoztathatják. Ezt a véletlenszerűre generált jelszót a rendszergazdának fizikai vagy elektronikus formában kell eljuttatnia a felhasználóknak. Jelenleg a rendszerben 25 db felhasználó szerepel. Közülük diákok:

|  |  |
| --- | --- |
| 1.a:  -litvan.laci  -uborka.ubul  -kapalacs.karoly  -cinege.cecil  -elemer.eniko  -antal.aniko | 1.b: -cinege.katalin  -ordas.odon  -papp.petra  -rokus.robert  -palacsinta.peter |

Gondviselők:

|  |  |
| --- | --- |
| -litvan.jeno  -uborkane.matilda  -kalapacs.bence  -cinegene.sara  -elemer.emil | -antal.andras  -ordas.kristof  -papp.terez  -rokus.jozsef  -palacsinta.pedro |

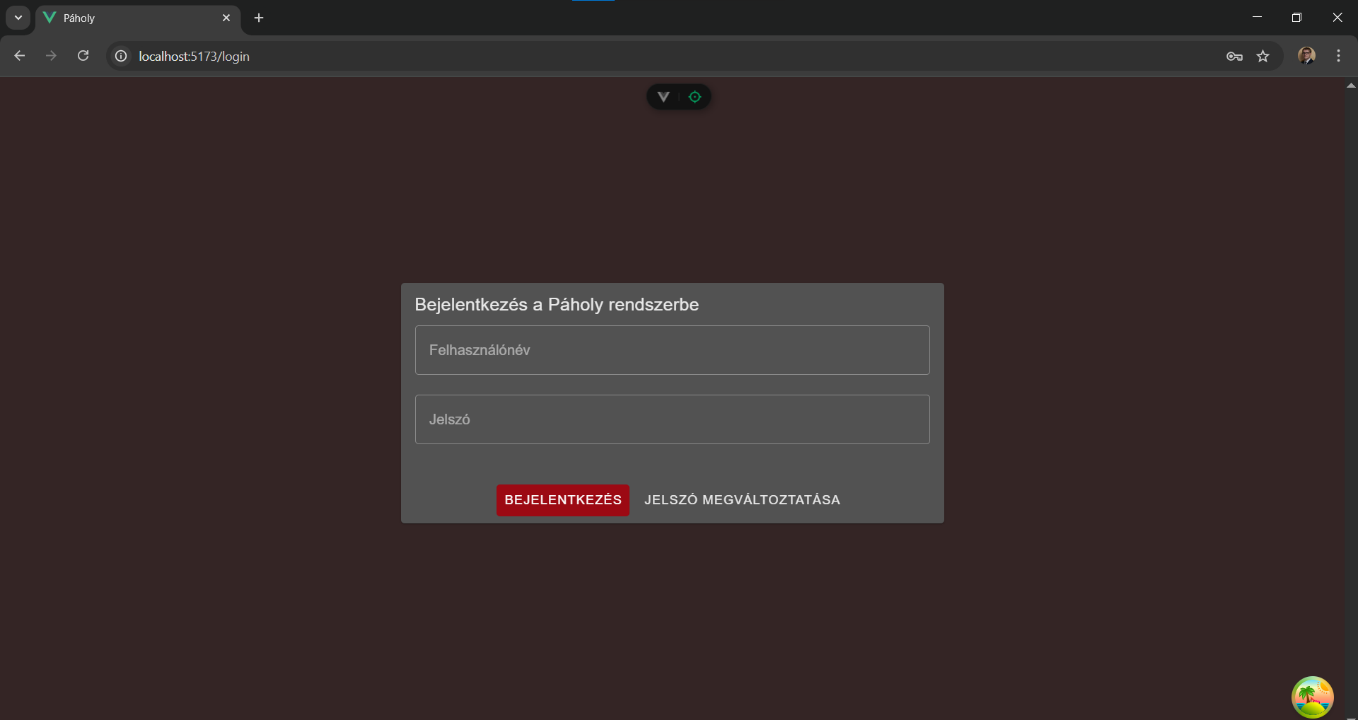
Pedgógusok:

|  |
| --- |
| -papp.lajos  -cegledi.eniko  -kovacsne.hajdu.borbala |

Jelenleg a szemléltetés és az átláthatóság érdekében minden felhasználónak „jelszo123” a jelszava. Szerepel az adatbázisban egy „szabo.patrik”, nevű rendszergazda is, neki is ugyanaz a jelszava.

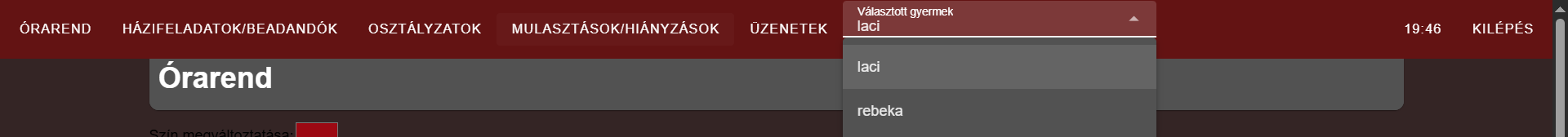
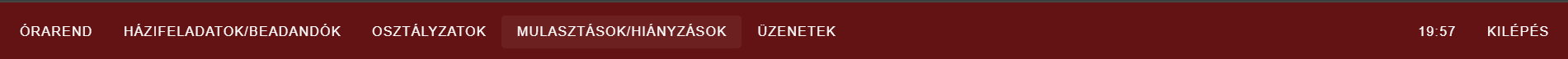
A példadokumentumok mappában található fájlokat a rendszergazda típusú felhasználó tudja feltölteni az adatbázisba. A rendszer txt, csv illetve xlsx típusú fájlokat fogad el.

### Bejelentkező oldal:

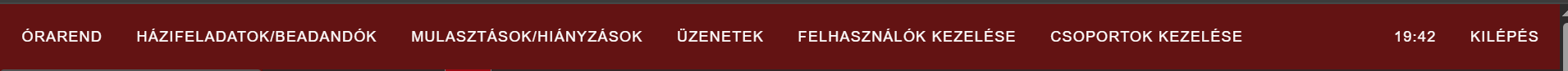
A Páholy oldalára a képen látható bejelentkező felületen keresztül lehet eljutni. Ezen a felületen a felhasználó a számára kiosztott felhasználónév-jelszó párossal tud belépni. Ezt a jelszót a rendszer generálja és a fiók létrehozását végző rendszergazda juttatja el a felhasználóknak. Amennyiben a felhasználó-jelszó páros valamelyike helytelen, a rendszer egy felugró szövegbuborékkal jelez a felhasználónak.

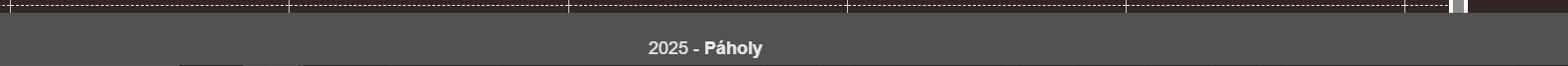
Abban az esetben, ha a felhasználó meg akarja változtatni a jelszavát azt a **Jelszó megváltoztatása** gombra kattintva teheti meg. Ez a gomb egy új oldalra vezet, ahol a felhasználónak lehetősége van a jelenlegi jelszava és felhasználóneve használatával újat feltölteni a rendszerbe. A rendszer hibát dob, ha a jelenlegi jelszó kétszeres megadása nem egyezik, ha a jelenlegi jelszó azonos az újjal illetve ha nincs ilyen felhasználónévvel rendelkező felhasználó. Amennyiben a jelszó megváltoztatása sikeresen megtörténik, az oldal visszalép a belépés felületre.

### Az oldal felépítése

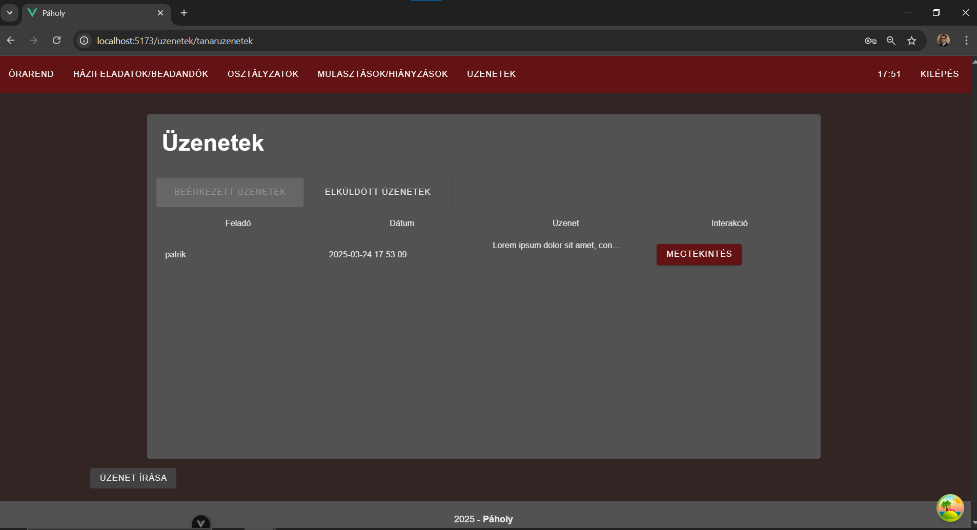
A Páholy rendszere könnyen kezelhető és letisztult. Az oldalon fejlécként minden időben látható egy menü, mely tartalmazza a gombokat az oldalak közötti navigációhoz.  Itt a felhasználó 5 oldal között tud váltani. A menü jobb szélén egy kilépés gomb található, mellette pedig egy visszaszámláló, mely lejárta után a felhasználó automatikusan kijelentkezik. Ez segít elkerülni az illetéktelen hozzáféréseket a számítógép/mobileszköz őrizetlenül hagyása esetére.

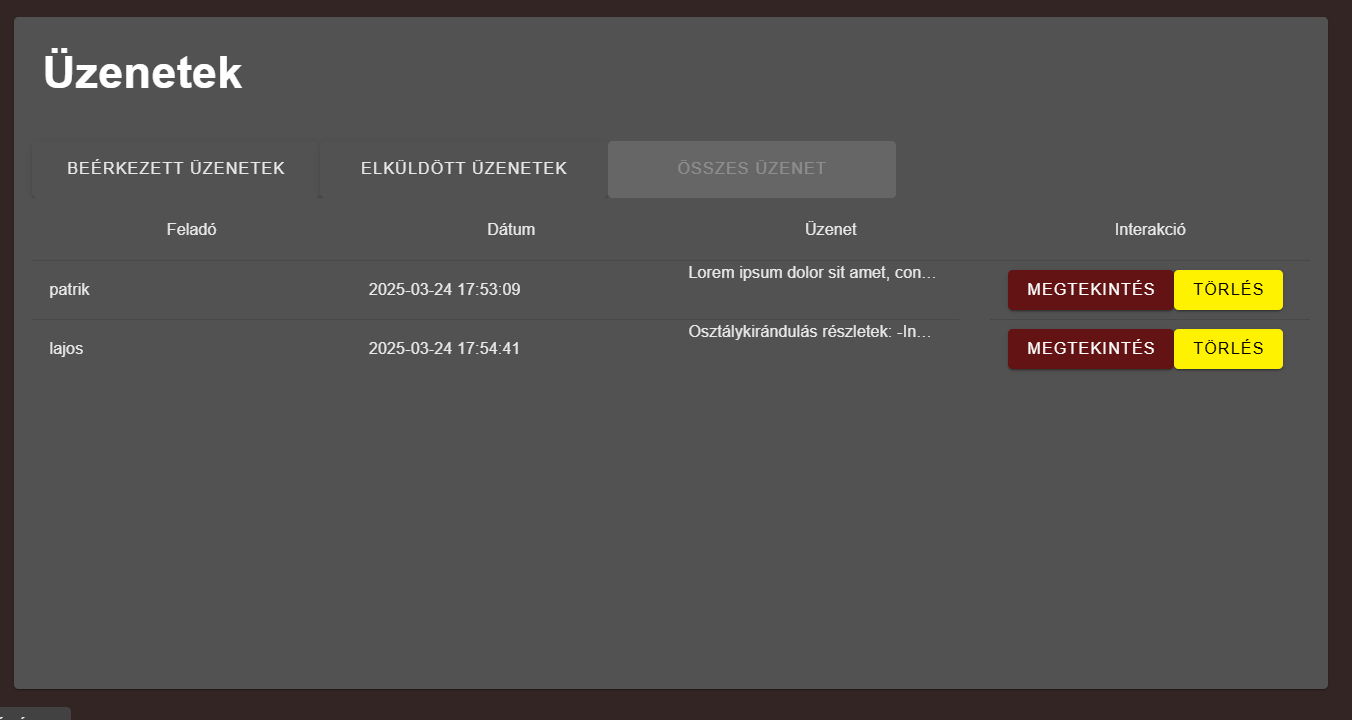
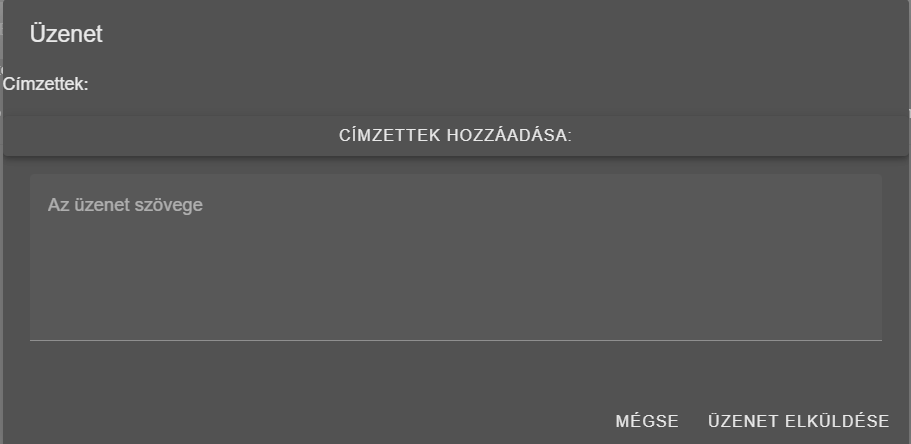
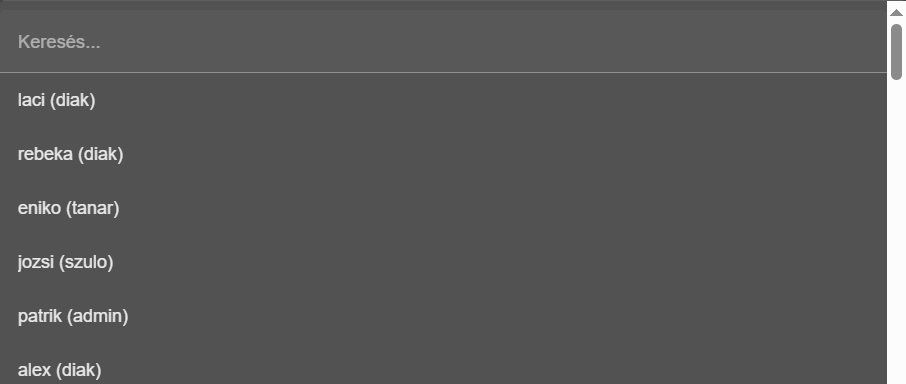
A menü eltér a gondviselő típusú felhasználók esetében. Itt egy leugró menüben tudnak váltani a hozzájuk tartozó tanulók között. Ennek jelentősége az órarend, a házifeladatok, az osztályzatok és a hiányzások megtekintésekor van. Az üzenetek oldalon nem történik változás a váltáskor.

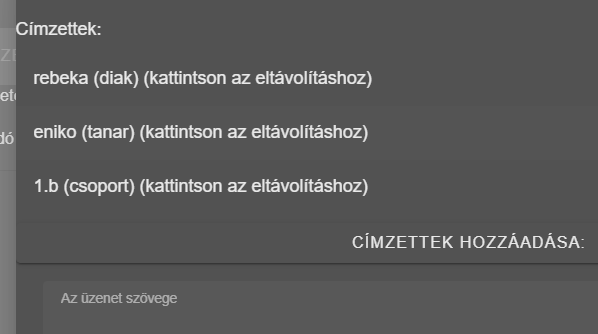
Rendszergazda típusú felhasználók számára két különleges oldal is rendelkezésükre áll a menüben. Ezek a „Felhasználók” kezelése és a „Csoportok kezelése” opciók. 

Legvégül minden oldal alján egy lábléc található, melyben az aktuális év és a Páholy név olvasható.

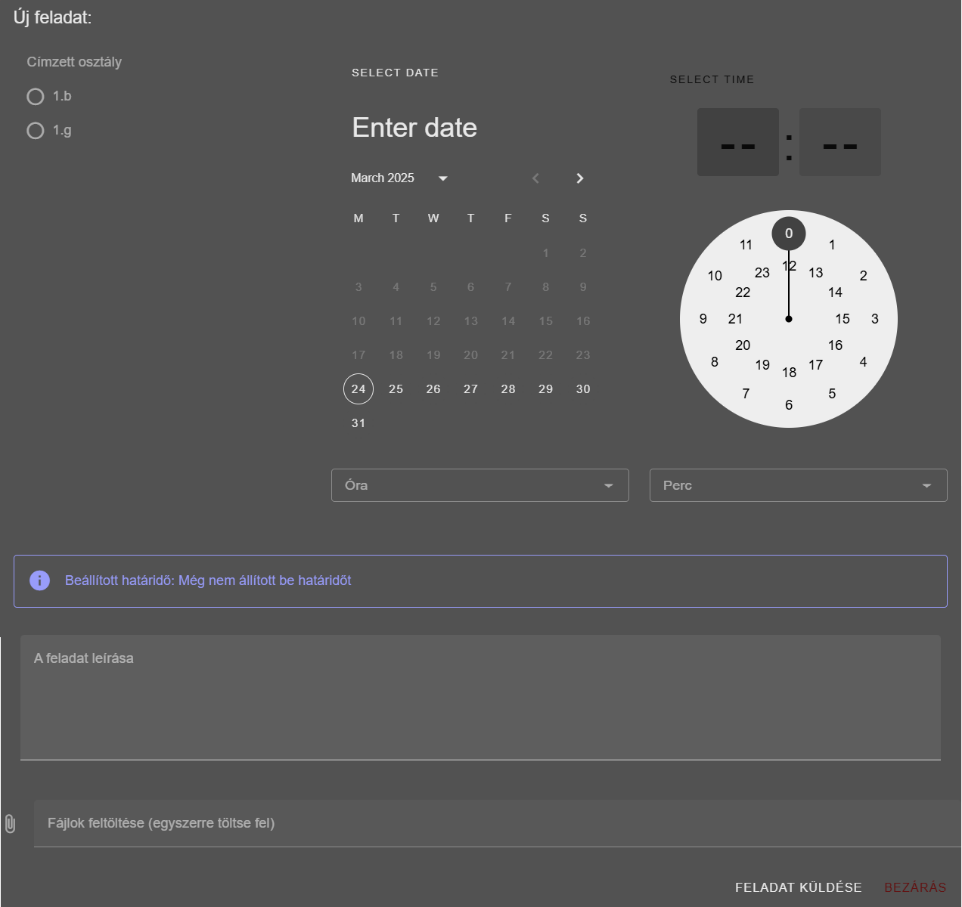
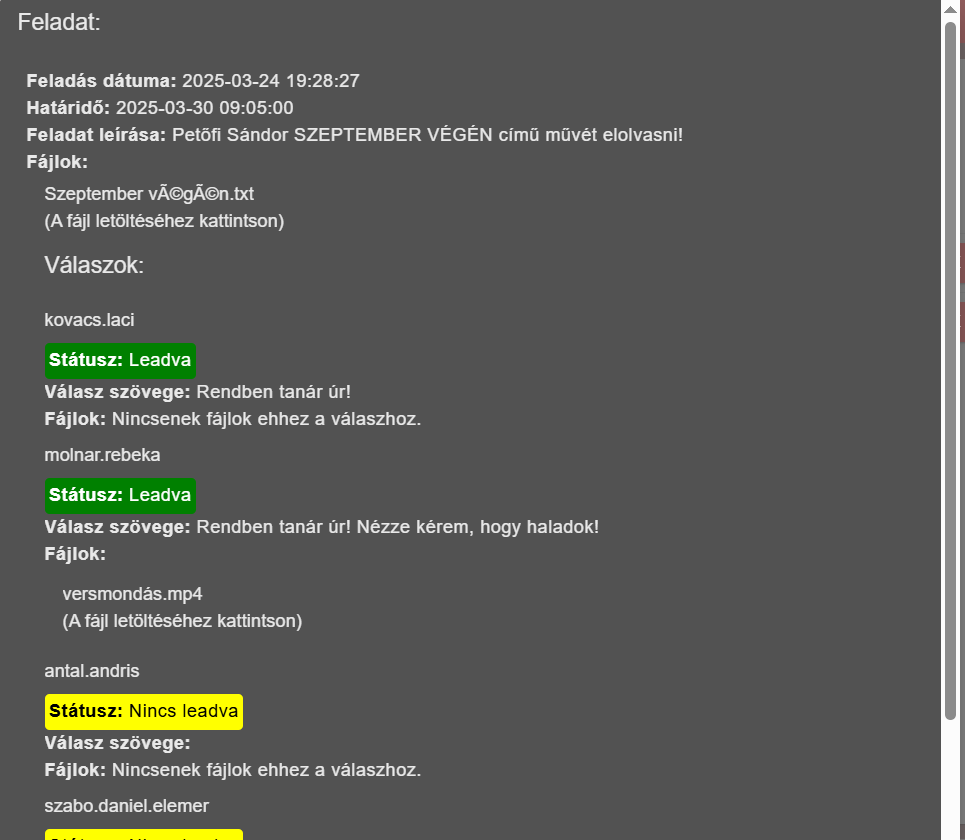
### Üzenetek felület

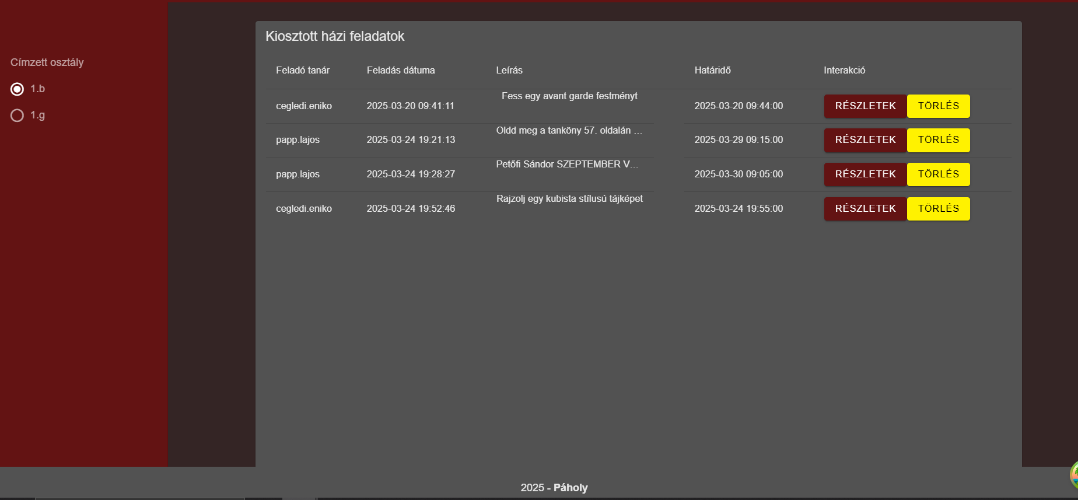
A Páholy rendszerében a diák, gondviselő és pedagógus fülek esetében megegyezik az üzenetek fül kinézete, mely a képen megtekinthető. Az igazán számottevő különbség köztük az, hogy a diák felhasználók számára az üzenet küldés funkciója nem elérhető. Itt megtekinthetőek az éppen bejelentkezett felhasználó számára bejövő és az általa küldött üzenetek is. Mindkét esetben megtekinthetők az üzentek részletei egy felugró ablakban, mely a **Megtekintés** feliratú gombra kattintva jelenik meg. Ebben az ablakban látható az üzenet küldőjének felhasználóneve, a többi címzett felhasználóneve, a küldés dátuma, illetve a teljes szöveg, mely táblázat nézetben lerövidítve jelenik meg.

Üzenet írására a gondviselő, rendszergazda és pedagógus típusú felhasználók jogosultak. Ez a bal alsó sarokban található **üzenet** gomb megnyomásakor egy felugró ablakban történik. Ebben az ablakban lehetőségünk van egyes felhasználóknak vagy akár egész diákcsoportoknak üzenetet küldeni. A mégse gombra kattintva az üzenet küldése megszakad és az ablak bezáródik. A **címzettek hozzáadása** gombra kattintva lenyílik egy menü, melyben kiválaszhatjuk az üzenet címzettjeit. Mivel a rendszer akár több száz vagy ezer felhasználó tárolására alkalmas, beépítésre került egy keresőmező. Az ebbe beírt betűk alapján a rendszer leszűri a felhasználókat illetve a csoportokat. Minden lehetséges címzett mellett ott van a Páholy rendszerben betöltött szerepköre. Amikor a címzettet kiválasztják, az az üzenet küldésére szolgáló felugró ablakon belüli Címzettek menüpont alá kerül a felhasználónév. A rendszer automatikusan elkerüli a hiányos üzenetek küldését. Amennyiben szöveg vagy címzettek nélkül kerül sor az üzenet küldésére, a rendszer hibát dob, de az ablakot nem zárja be, ezzel elkerülve a szöveg vagy a kiválasztott címzettek eltűnését.

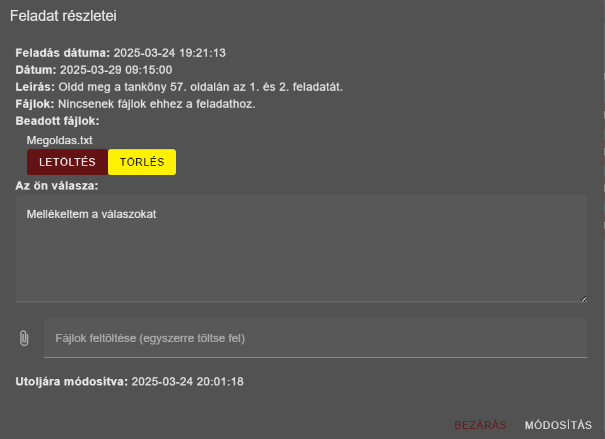
 A rendszergazda típusú felhasználók számára eggyel több funkció áll rendelkezésre. Nekik lehetőségük van az összes üzenet megtekintésére, azok törlésére is. Megtekintéskor ugyanaz a felugró ablak jelenik meg, mint bármely más esetben. A törléskor egy másik felugró ablakban visszaigazolást kell adnia arról, hogy biztosan törölni szándékozik-e.

### Házi feladat felület

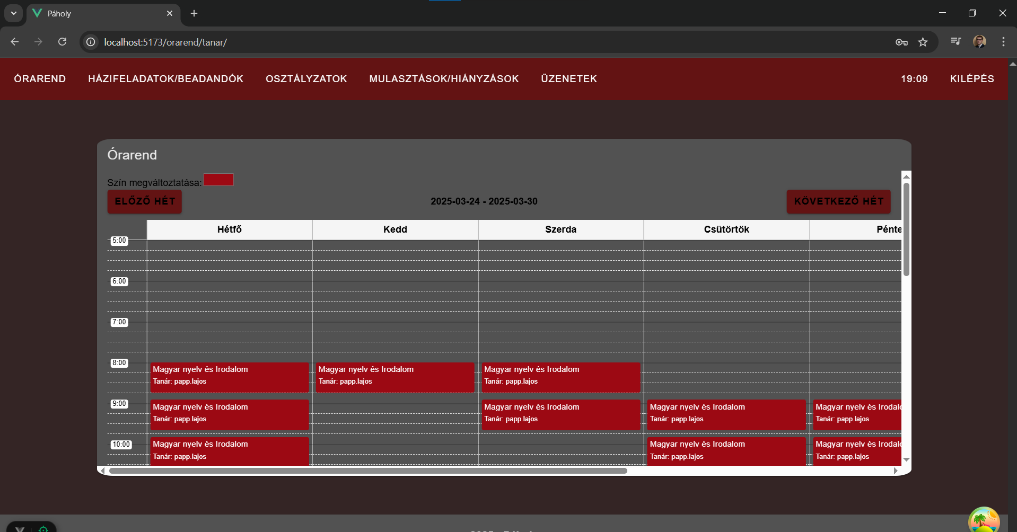
 A házi feladat a tanár oldalról a képen látható. Egy táblázatban mutatja az éppen bejelentkezett felhasználó által kiküldött házi feladatokat. Minden házi feladatnál kiírja annak az elkészítési határidejét, feltöltési idejét, rövidített leírását illetve a vele kapcsolatos interakciókat. A „Válaszok megtekintése” feliratú gombra kattintva egy felugró ablakban tekinthető meg az adott feladat részletes leírása, mellékelt fájljai, illetve válaszok, melyeket a diákok adtak és az általuk feltöltött fájlok. Színes mezővel jelöli az oldal a leadás státuszát. A zöld a leadott, a sárga a még le nem adott feladatra, a vörös a határidő lejártáig meg nem történt leadásra utal. Amennyiben a határidő lejár, a diák nem tud sem szöveges választ, sem fájlokat beküldeni. A fájlokra való kattintás annak letöltését eredményezi. A törlés gombra kattintáskor egy visszaigazoló ablak jelenik, és csak az ott történő visszaigazolás után történik meg a törlés. Új feladat kitűzésére az oldal bal alsó sarkában lévő feladat kitűzése gomb megnyomásával van lehetőség. Ennek megnyomására újabb felugró ablak jelenik meg. Ezen az ablakon kiválasztható egy darab csoport mely számára a feladat kiküldésre kerül. Mellette egy interaktív naptárban és órában adható meg a határidő. Ezek alatt egy kék információs mezőben tekinthető meg a beállított határidő. Abban az esetben, ha az óra nem válna a felhasználó tetszésére, két legördülő menü is elérhető a határidő órájának és percének beállítására. Amennyiben még nem történt meg a határidő beállítása a „Még nem állított be határidőt” felirat jelenik meg. Lentebb görgetve látható egy szövegbeviteli mező, melybe a feladat szövege fog kerülni, illetve egy fájlbeviteli mező, melybe el lehet helyezni a feladathoz tartozó fájlokat. A rendszer figyelmeztet, amennyiben leírás, határidő vagy kiválasztott csoport nélkül próbálunk házi feladatot feltölteni. Továbba figyeli és titlja, hogy eleve lejárt határidőt adjon meg a felhasználó.

 Diák illetve szülő oldalról szintén táblázatból tekinthetőek meg a házi feladatok adatai. Először a feladó tanár neve, aztán a határidő, feladási idő, rövídett leírás, a feladat státusza illetve a **Feladat megtekintése** gomb. Akárcsak a tanári oldalon, jól elkülöníthető színekkel van jelezve a feladat állapota. A feladat megtekintése gombra kattintva egy felugró ablak tűnik elő, melyben megtekinthetőek a feladat adatai részletesen, illetve a válasz szövege, mely szerkeszthető, és a feltötlött fájlok, melyek utólag törölhetőek és letölthetőek. Minden mentetlen módosítás elveszik, amennyiben a felhasználó a **bezárás** gombra kattint. A módosítás gombra kattintás után az oldal visszajelzi, hogy a módosítás sikeresen megtörtént.

Amennyiben a gondviselő kívánja megtekinteni gyermeke házi feladatait, a kezelőfelület azonos lesz a diák oldalon tapasztaltakkal. A fenti diák választó menüvel lehet frissíteni a megjelenő házifeladatok listáját.

 Rendszergazda típusú felhasználóknak lehetőségük van minden osztály minden feladatát megtekinteni. Egy baloldalt megjelő menün tudják kiválasztani az osztályt, amelynek feladatait meg kívánják tekinteni. Ezt, illetve azt leszámítva, hogy nem képesek feladatot kítűzni, ez az oldal megegyezik a pedagógusi nézettel.

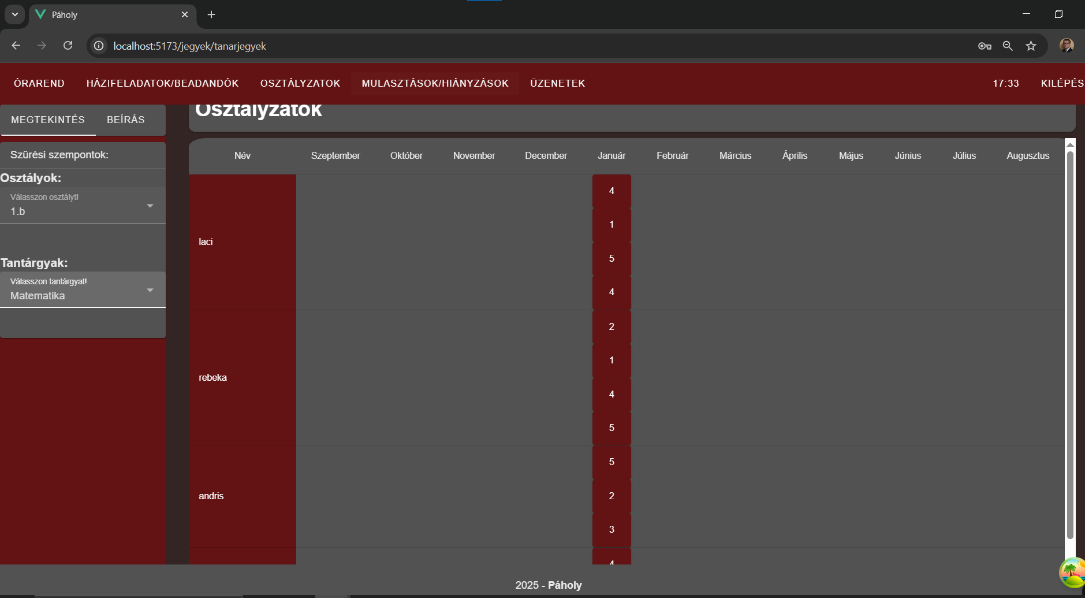
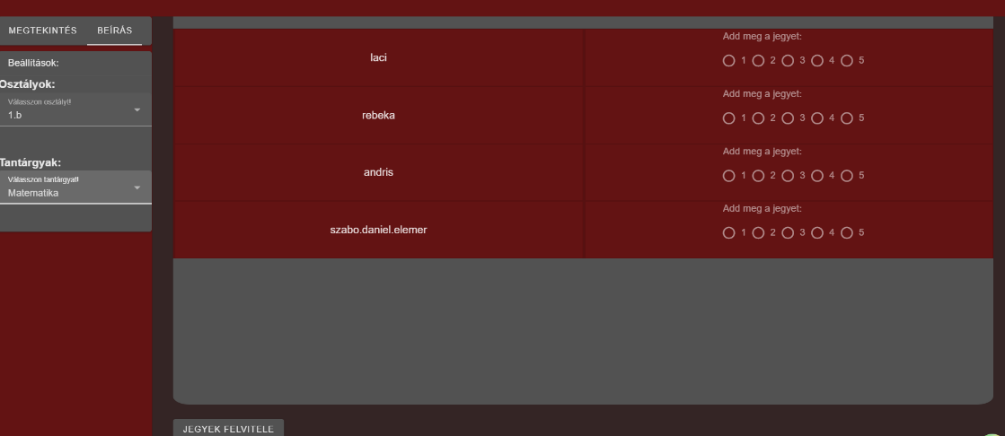
### Órarend felület

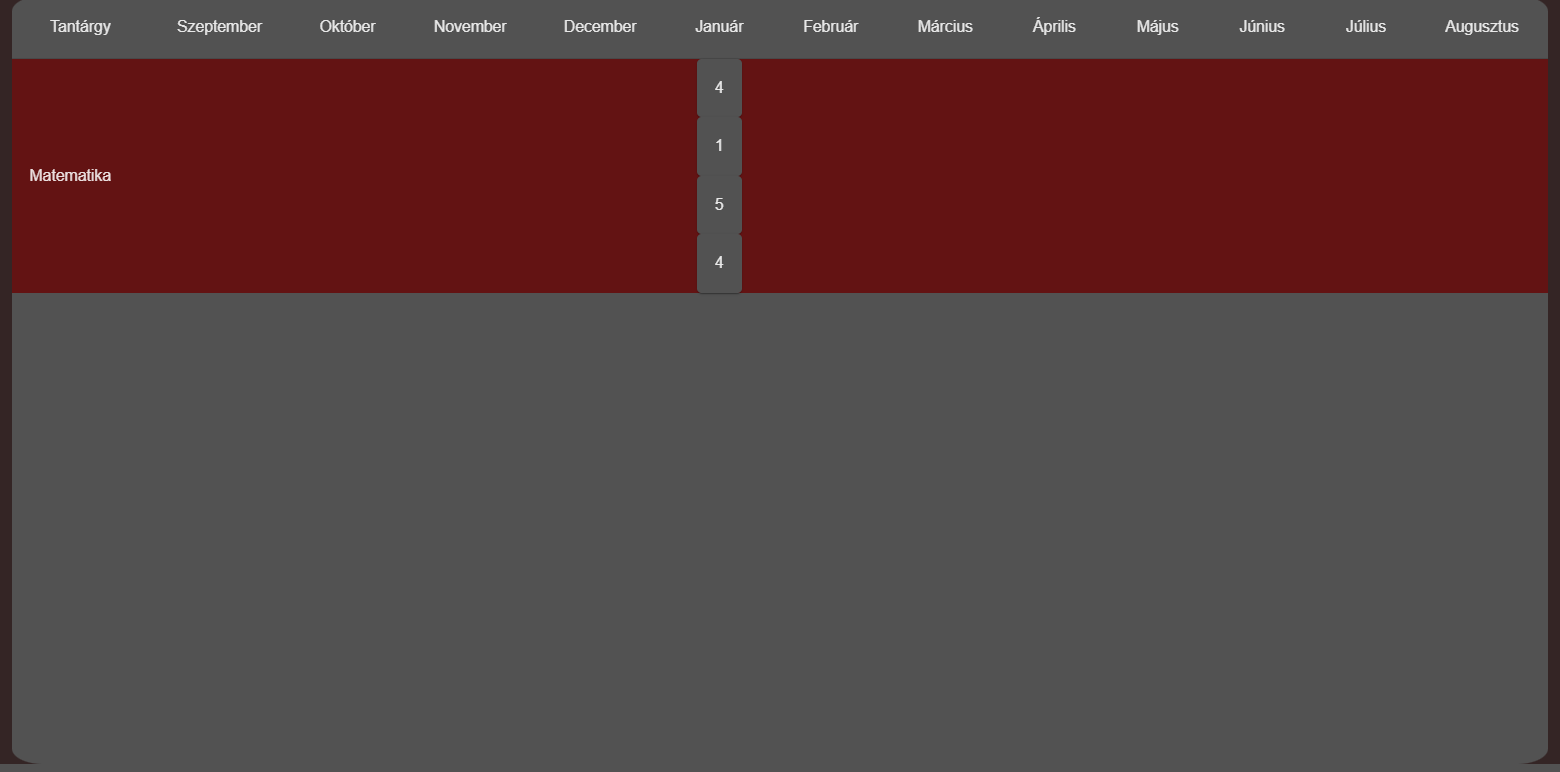
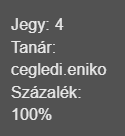
Az órarend felülete szinte minden típusú felhasználó számára egyforma. Egy táblázat, melyben tizenöt perces időbeosztásokban jelenik meg az órarend. Az órarend felett áll két váltás gomb, melyen vissza illetve előre lehet tekerni az órarendet egy héttel (függőleges tájolású eszközökön egy nappal). Ennek az a szerepe, hogy könnyeben megjeleníthető legyen a hiányzás illetve a „különleges óra”. A Páholy rendszere minden az alap órarendtől eltérő órát, így a helyettesített órákat, elmaradó órákat, illetve egyéb foglalkozásokat is egyéb óraként kezeli. Ezek megjelenítése színben is eltér a szokásos órákétól. Pirossal a rendszeresen zajló, ismétlődő órák láthatóak, míg sárgával a rendkívüli órákat jelöljük.

A rendszergazdának lehetősége van az órarendeket csoportokra lebontva megtekinteni, módosítani, törölni, továbbá helyettesítéseket, elmaradó órákat beírni egy felugró ablakon belül. A helyettesítés beírásához írja be az adatokat, majd kattintson az **Óra helyettesítése** feliratú gombra. Ez egy új felugró ablakot nyit meg, melyben lehetősége van kiválasztani a napot, amikor helyettesítésre kerül sor. Az óra módosítása gombra kattintva az óra adatai megváltoznak a szövegbeviteli mezőbe beírtak szerint. Az órákat törölni is lehet.

A bal felső sarokban található navigáció sáv segítségével, mely alapértelmezetten **Megtekintés** állapotú, átléphetünk az Órarend felvitel oldalra, ahol lehetőségünk van manuálisan vagy akár fájlokból órát feltölteni. Ezeknek a fájloknak tartalmaznia kell: a tanár Páholy rendszer azonosítóját, a csoport Páholy azonosítóját, kezdő órát és percet, az óra hosszát, napját illetve magát a tantárgyat.

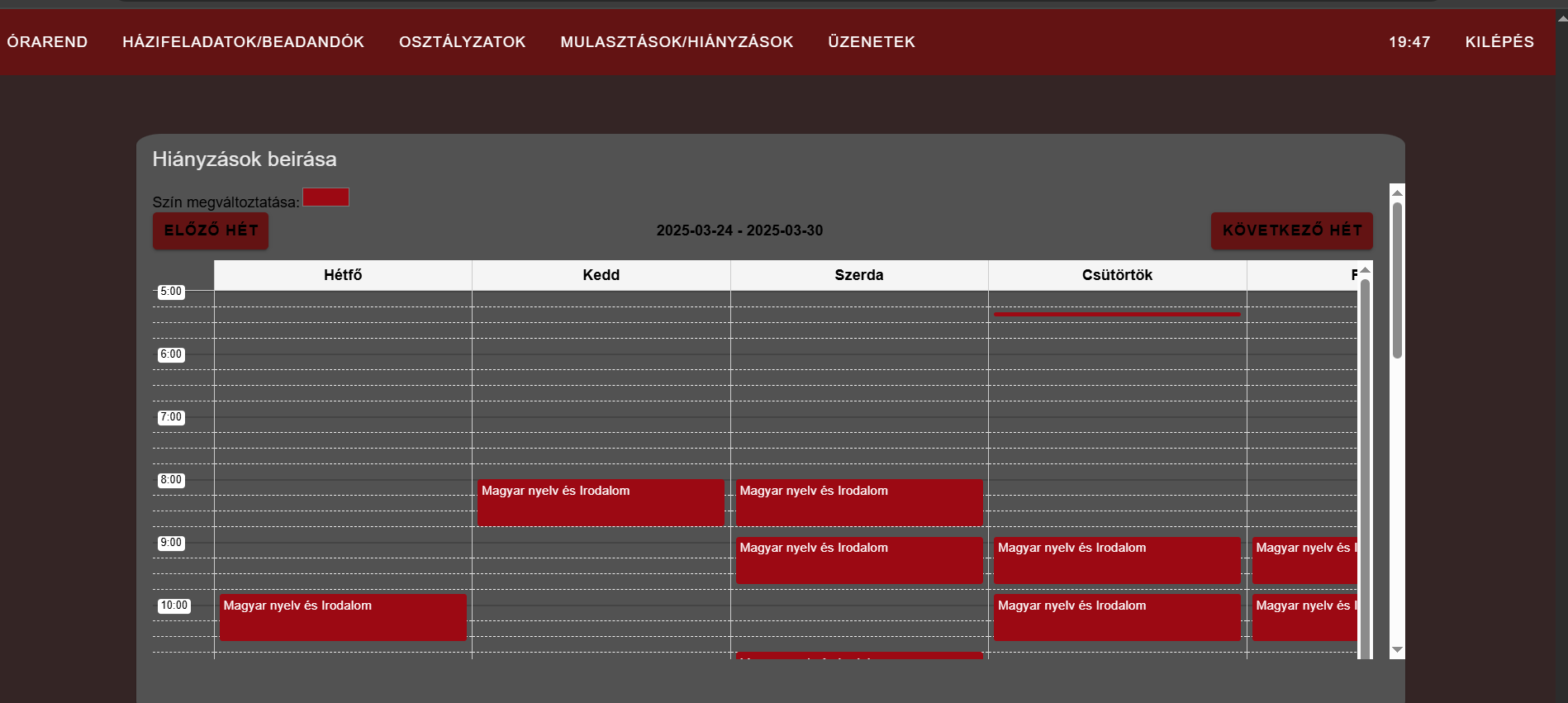
### Érdemjegyek felület

Tanári oldalról a jegyek felület az alábbi módon kezelendő: A képernyő bal oldalán található egy két gombból álló menü. Alapesetben a **megtekintés** fül van kiválasztva. Itt lehetőség van először osztályok, majd tantárgyak alapján keresni a jegyeket. A felület hónapokra lebontva mutatja az egyes diákok érdemjegyeit az adott tantárgyban. Az egeret a jegy fölé húzva megtekinthetők annak adatai. A **beírás** gombra kattintva egy hasonló felület jelenik meg, mely először minden adattól mentes. Itt is először osztályt és tantárgyat kell beállítani, hogy jegyet tudjunk rögzíteni. Itt minden tanár típusú felhasználó csak az általa tanított osztályok számára tud jegyet beírni. A jegyek felvitelére azok beírása után a **Jegyek felvitele** feliratú gombbal van lehetőség. Sikeres feltöltés esetén a rendszer visszajelez egy üzenet formájában.

Diák és gondviselői oldalról hónap és tantárgy szintű lebontásban tekinthetőek meg az érdemjegyek egymás mellett. Az egér jegy fölé csúsztatása esetében megtekinthetőek annak részletes adatai .

A rendszergazdának lehetősége van a saját oldalán osztály illetve tantárgy alapján szűrni és megtekinteni a jegyektet, illetve azokat törölheti a bal egérgombbal való kattintással.

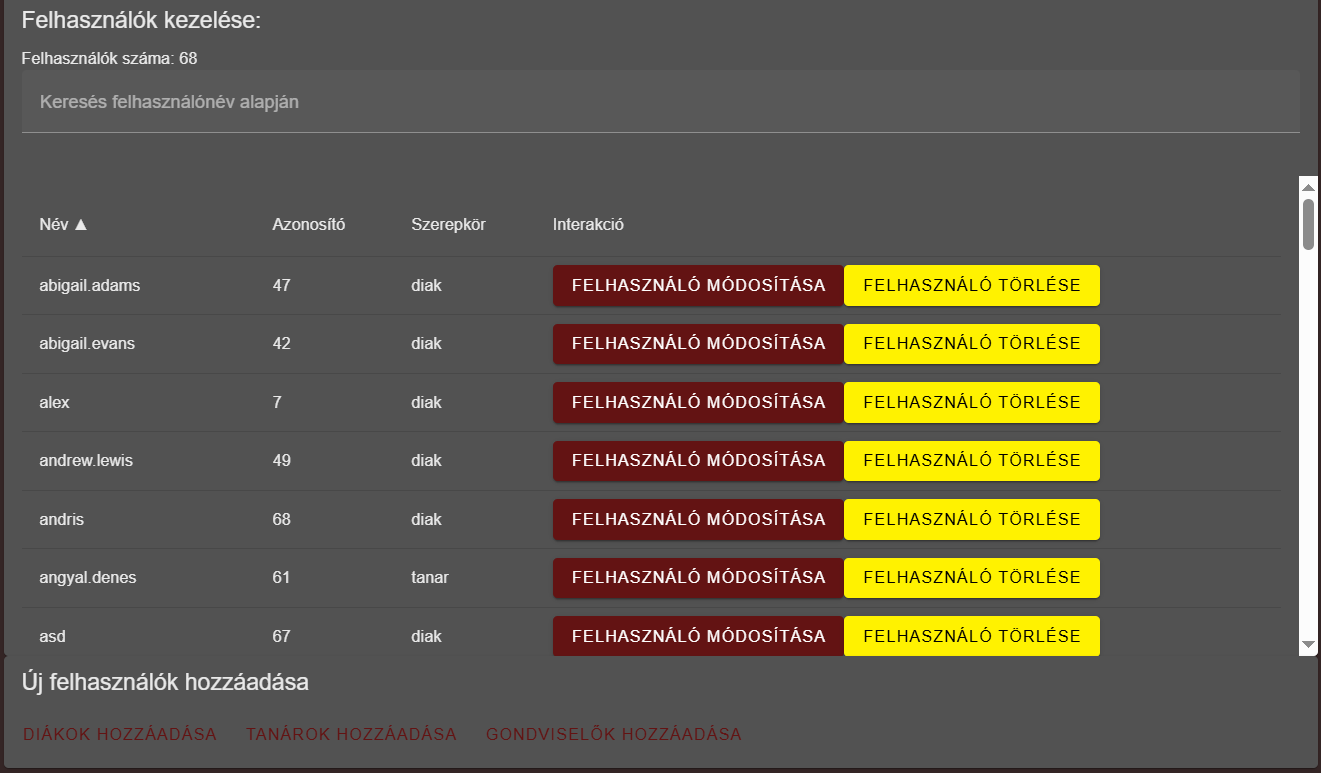
### Hiányzások felület

 A hiányzások felület pedagógusi oldalról szinte azonos az órarend felülettel, viszont itt az órákra kattintva megjelenik egy felugróablak, amiben a felhasználónak lehetősége van óránként feljegyezni a diákokat. Miután ez megtörtént, az oldal visszajelez a sikeres hiányzás felvitelről.

A hiányzásokat a diák és gondviselő napokra lebontva lenyílómenükben látja.

 Rendszergazda nézetben van lehetőség a hiányzások törlésére illetve igazolására. Ilyenkor napokra bontva megjelenik az összes aznapi hiányzás, mellette az adatokkal, mint például, ki hiányzott, mikor, milyen tanóráról és mennyi ideig.

### Felhasználók kezelése felület

Ezt az oldalt csak rendszergazda típusú felhasználók képesek elérni. Itt lehetőség van a jelenlegi felhasználók adatainak módosítására, a felhasználók törlésére, és új felhasználók bevitelére. A **Felhasználó módosítása** feliratú gomb megnyomásakor felugró ablakban változtathatjuk meg az egyes felhasználók adatait. A rendszer figyelmeztet, ha már foglalt OM azonosítót próbálunk megadni, illetve ha már foglalt felhasználó névre próbáljuk módosítani, a felhasználónevet. Törléskor felugró ablakban van lehetőség a visszaigazolás megadására a törlést illetően. Az oldal tetején helyet kapott egy keresőmező, melyben felhasználónév alapján kereshetünk rá az emberekre. Alul három különböző gomb van **diákok**, **gondviselők** és **tanárok feltöltéséhez.** Ezek megnyomása egy egy felugró ablakot aktivál. Az ablakokban lehetőség van manuálisan hozzáadni, illetve fájlból feltölteni az új felhasználók adatait. Mindegyik ablakban meg van adva milyen adattagoknak kell szerepelniük egy felhasználó feltöltéséhez. Diákoknál ezek az adatok a következők:

* Teljes név
* Születési dátum
* Lakcím
* Telefonszám
* E-mail cím
* OM azonosító

Pedagógusok esetén:

* Teljes név
* Telefonszám
* E-mail cím
* Születési dátum

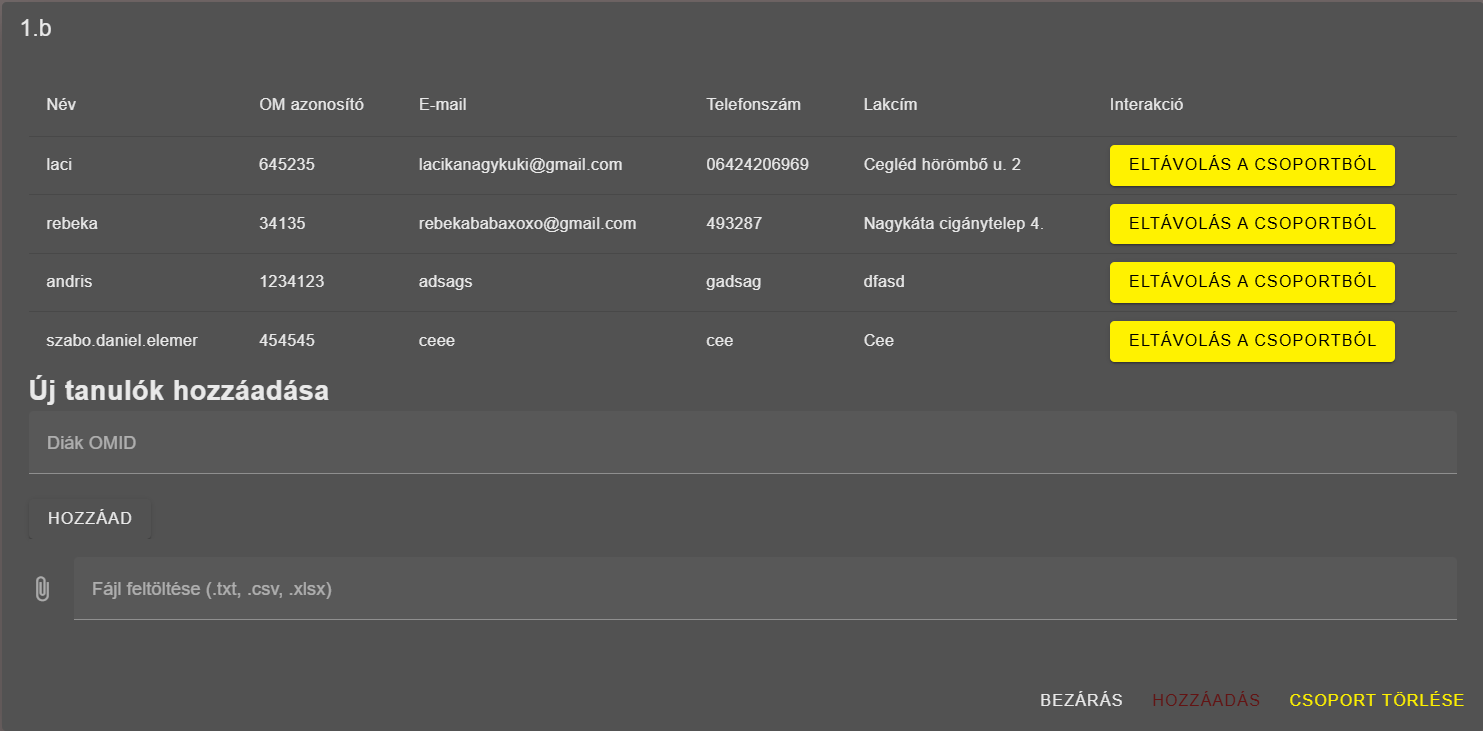
Gondviselők esetén:

* Teljes név
* Születési dátum
* Lakcím
* Telefonszám
* E-mail cím
* A gondviselőkhöz tartozó diákok OM azonosítói vesszővel elválasztva

Minden felvitel esetében lehetőség van .txt, .csv, .xlsx típusú fájlokból való feltöltésre. Elöször fel kell tölteni ezeket a fájlokat, aztán a „Fájlok beolvasása” gombra kattintva ezek beolvasásra kerülnek. Ezek kipróbálásához példa fájlokat a telepítés mappában talál.

A küldés gomb megnyomása után egy txt file automatikusan letöltődik. Ebben a fájlban szerepelnek az újonnan létrehozott felhasználók felhasználónevei és jelszavai. Ezeket a rendszergazda juttatja el az illetékeseknek.

### Csoportok kezelése felület

A Csoportok kezelése felület nagyban hasonlít a Felhasználók kezelése funkcióhoz. Egymás alatt jelennek meg a csoportok a nevükkel és a tagjaik számával együtt. A csoportra kattintáskor megtekinthetők annak tagjai. Itt lehetőség van új diákok hozzáadásár OM azonosító alapján, illetve a jelenlegiek eltávolítására a csoportból vagy akár az egész csoport törlésére. Ez nem távolítja el a felhasználókat, csak a csoporttal kapcsolatos információkat, például a tanórákat.

### Program leállítása, program törlése

A program futtatásának leállításához egyszerűen zárja be a futtatáshoz korábban megnyitott parancssorokat, illetve az XAMPP-ot.

Ahhoz, hogy az XAMPP alkalmazásból eltávolíthassuk a Páholy adatbázist, kattintsunk a bal szélen található paholy gomra, majd középen felül az SQL gombra és írjuk be a következőt a nagy szövegbeviteli mezőbe: „drop database paholy”, majd kattintsunk az indítás gombra. Ezzel törlődött az adatbázis, ami a Páholy rendszeréhez tartozott. A forráskód eltávolításához helyezze a letöltött fájlokat a lomtárba.

# Fejlesztői dokumentáció

A Páholy rendszer egy olyan felület, ahol egy iskola dolgozói és tanulói egyaránt igénybe vehetik a rendszer által adott információkat, szolgáltatásokat könnyedén egy weboldalon keresztül, ezek között:

* Órarend
* Érdemjegyek
* Hiányzások
* Üzenetek

Az adott felhasználó lekérdezheti ezekkel kapcsolatban az adatait amit az oldal megjelenít egy rendezett módon, és akár fel is tölthet új adatokat (pl tanár beír egy jegyet)

## Frontend

A kezelőfelület a „Vue” keretrendszerrel lett megoldva ami könnyítette az adatok megjelenítését és a fejlesztés folyamatát, egy alap HTML-hez képest sokkal több funkcióval rendelkezik amit próbáltunk a legjobban kihasználni

Emellett a frontend használ más kiegészítőket is, mint például a pinia store-t(lokális adatok kezeléséhez)és az axios-t (API hívásokhoz)

### Projekt fájlrendszer elrendezése

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Fontosabb mappák/fájlok:

* src: „Forrás” mappa, ez tartalmazza a kódot
* api: Lekérdezésekkel kapcsolatos kódok
* lib: Oldal elrendezéssel (route) kapcsolatos kódok
* router: oldalak/aloldalak elérési útvonalait tartalmazza
* stores: Lokálisan/memóriában elmentett adatokat kezelő kódokat tartalmaz (pinia store)
* utils: Lekérdezési kulcsokat tartalmazza (tanstack)
* views: Oldalak/aloldalak kódját tartalmazza, ezek a fájlok vannak behívva a layout mappában lévő rendező fájlok által
* lib: létrehozza a kiegészítőkhöz kellő példányokat amivel behívhatók, például az axiosClient példány amin keresztül lekérdezéseket lehet elbonyolítani

App.vue: Ez a fő fájl, ami létrehozza az oldal példányát és megjeleníti a weboldalt

### Kiegészítők

* Axios: Egy http lekérdezés klienset biztosít amivel kéréseket lehet küldeni
* Tanstack: Lekérdezések gyorsítótárba való lementését biztosítja
* Pinia: Adatok lokális kezelésére létrehozható globális tárolók
* jwt-decode: JSON Web Token dekódolásáért felel
* vuetify: bootstrap megfelelője a Vue-hoz, ad egy alap stílust a Vue tagekhez, + behoz új tageket

### Oldal felépítés

Ha nem vagy bejelentkezve a bejelentkezés oldalra dob be, egyébként meg az órarendhez

Bejelentkezve az oldalon elérhetőek a felhasználó fajtához tartozó aloldalak:

Oldal betöltésének lépései:

1. App.vue inicializálja a klienst, behívja a lib mappából a kiegészítők példányait is
2. RouterView segítségével betöltenek az url alapján, a routerben megadott komponensek
3. Betölt a layoutban megadott .vue elrendezés ami a megfelelő komponenst jeleníti meg
4. A megjelenített komponens(ek)-ben megtörténnek a lekérdezések

A „layout” .vue fájlok szükségesek, mivel így belehet tölteni különböző felületeket ugyan arra az oldalra (pl egy sidebar amit újra szeretnénk használni máshol is, de mondjuk egy másik oldalon meg egyáltalán nem akarjuk hogy megjelenjen)

A „views” mappa tartalmazza ezeket a betöltött komponenseket, funkcionalitásuk alapján vannak mappába rendezve

### Lekérdezések

A lekérdezések az Axios kliensen keresztül történnek meg, és az „api” elnevezési mappa tartalmazza őket, funkcionalitásuk alapján vannak mappába rendezve

Lekérdezések:

* admin: admin felülettel kapcsolatos lekérdezések (pl.: órarend feltöltés, felhasználó feltöltés)
* auth: bejelentkezéssel kapcsolatos lekérdezések
* hazik: feladat felülettel kapcsolatos lekérdezések (feladatok lekérdezése, válasz felküldés)
* hianyzasok: hiányzásokkal kapcsolatos lekérdezések (hiányzások lekérdezése, hiányzás beírás)
* jegyek: érdemjegyekkel kapcsolatos lekérdezések (érdemjegyek lekérdezése, érdemjegyek beírás)
* orarend: órarenddel kapcsolatos lekérdezések (órarend lekérdezés)
* szulo: szülő oldali lekérdezések (minden féle lekérdezés szülő oldalra átírva, hogy ne a szülő adatait kérdezze le, hanem a kiválasztott gyerek adatait)
* uzenetek: üzenetekkel kapcsolatos lekérdezések (üzenet lekérdezés)

Minden lekérdezés egy „useQuery” vagy „useMutation” függvényben van a tanstack miatt, így azokkal az adatokkal könnyebb dolgozni, pl láthatjuk a kérés státuszát (sikerült, folyamatban, nem sikerült), és könnyen lehet újra kérni ezeket az adatokat (refetchQueries függvénnyel)

### Adatok

A legtöbb adat már elrendezve jön meg az API-tól, viszont sok helyen még az oldal kódján belül is rendezésre kerül. A vue egyszerűvé teszi a megjelenítését v-forral, sok táblázat cellái megvannak magában az oldalban írva egyszer, mint egy sablon, beolvassa a lekérdezett adatot és megjeleníti adattagonként

A Vue egyik fontos része a „ref” megnevezés amivel egy referencia objektumot hozhatunk létre, ezért nagyon hasznos új adatok felvitelénél még a lokális tároláshoz, majd abból kiolvasásához, és akár több helyről is belehet olvasni ugyan azt az adatot úgy hogy ha változik akkor az a többi helyen is megváltozik

### Fajták

Typescript-ben lett megírva minden lekérdezés és számítás kód, így sok helyen kell fajtákat használni, a lekérdezés fájlok mellett a „kerdezes\_nev.d.ts” nevű fájlokban találhatóak a visszakapott adatok fajtáinak deklarálása

### Visszajelzés

A képen szöveg, képernyőkép, embléma, zöld látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Az „errorHandler” és „statusHandler” tároló felel a lekérdezések visszajelzésére, ami bárhonnan meghívható és a „setError” és „setStatus” függvénnyel beküldhető egy szöveg amit megjelenít az oldal alján. Az App.vue-ban helyezkedik el a megjelenítő kód hogy bárhol látszódjon, akár még aloldal váltás után is a visszajelzés

## API

Az adatbázishoz tartozó API lehetővé teszi hogy lekéréseket kérhessen a weboldal, egy kontrolált módon, esetleges hiba esetén pedig lekezelje azokat és meg

### Projekt fájlrendszer elrendezése

A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Fontosabb mappák/fájlok:

* \_mocks\_: teszteléshez használt adatbázis másolat létrehozó fájlt tartalmazza
* tests: tesztek
* routes: végpontokat tartalmazza
* controllers: irányítók amik egyenesen a routes-ból vannak meghívva
* services: függvények amik az adatbázisból lekérdezett adatokkal számítanak, legtöbbször csak magát az adatot küldik tovább
* repositories: a táblákhoz való lekérdezéseket tartalmazza
* models: sequelize-al itt vannak meg adva az adatbázis táblái és köttetései
* app.js: Ez a fő fájl, ami létrehozza az adatbázist/megnézi hogy létezik e az adatbázis, meghívja a routes-ból a végpontokat

### Kiegészítők:

* express: web applikáció keretrendszer, lehetővé teszi a végpontok definiálását
* nodemon: automatikusan újra indítja az applikációt egy változtatás után
* jest: tesztelési keretrendszer
* sequelize: ORM adatbázis létrehozását teszi lehetővé
* bcrypt: titkosítháshoz
* jsonwebtoken: token generáló rendszer, felhasználók ellenőrzésére

### Végpont dokumentáció

Végpontokhoz egy kigenerált dokumentációt készítettünk el, amely elmagyarázza az egyes végpontokhoz beküldendő adat formátumát, és válaszait, „/documentation” végponton elérhető webes formában

## Adatbázis

Az adatbázis nem nyers SQL-ben lett megírva hanem a sequelize segítségével, legfőbb előnye hogy json objektum formában lehet megadni és módosítani a táblákat, és az adatok lekérdezése és feltöltése is egyszerűbben megoldható

### Táblák

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Minden fájl egy külön tábla mezőit és tulajdonságait tartalmazza, ez alapján készíti el az SQL-es adatbázist

Táblák rövid leírásai:

* absences: tanulók hiányzását rögzíti

1. ID: azonosító (elsődleges kulcs)
2. studentID: hiányzó diák azonosítója
3. teacherID: hiányzást naplózó tanár azonosítója
4. date: hiányzás dátuma
5. excused: igazolás státusza
6. lessonID: hiányzott óra azonosítója

* admins: adminisztrátori fiókok adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. name: adminisztrátor teljes neve
3. phone: adminisztrátor telefon száma
4. email: adminisztrátor email címe
5. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója

* assignmentfiles: feladatokhoz csatolt fájlokat rögzíti

1. ID: azonosító (elsődleges kulcs)
2. assignmentID: feladat azonosítója amihez tartozik a fájl
3. buffer: fájl
4. mimetype: fájl kiterjesztése
5. filename: fájl neve

* assignments: tanárok által kitűzött feladatokat rögzíti

1. ID: azonosító (elsődleges kulcs)
2. teacherID: adminisztrátor teljes neve
3. desc: feladat leírása
4. deadline: feladat határideje
5. uploadDate: feladat kitűzésének dátuma

* classdisruptions: helyettesített/módosult órák adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. date: helyettesített/módsítás dátuma
3. groupID: csoport azonosítója akikhez az óra lett rendelve
4. start\_Hour: óra kezdete órában (óra:perc)
5. start\_Minute: óra kezdete percben (óra:perc)
6. length: óra hossza
7. day: melyik napon van az óra (hétfő,kedd, stb…)
8. subjectName: óra tantárgya

* completedassignments: feladatokra beadott választ rögzíti

1. ID: azonosító (elsődleges kulcs)
2. assignmentID: feladat azonosítója amihez tartozik a válasz
3. studentID: feladatot leadó tanuló azonosítója
4. date: feladat leadásának dátuma
5. textAnswer: leadott válasz szöveges tartalma
6. status:leadás státusza (leadva, nincs leadva, határidő lejárt)

* completedassignmentfiles: elkészült feladatokhoz csatolt fájlokat rögzíti

1. ID: azonosító (elsődleges kulcs)
2. assignmentID: feladat azonosítója amihez tartozik a fájl
3. buffer: fájl
4. mimetype: fájl kiterjesztése
5. filename: fájl neve

* groups: csoportokat rögzíti

1. ID: azonosító (elsődleges kulcs)
2. name: csoport neve

* guardians: szülő fiókok adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. name: szülő teljes neve
3. phone: szülő telefon száma
4. email: szülő email címe
5. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója

* guardianstudents: kapcsoló tábla (szülő<gyerek)

1. GuardianID: szülő azonosító (idegenkulcs)
2. StudentID: gyerek(diák) azonosító (idegenkulcs)

* lessons: órákat rögzíti

1. ID: azonosító (elsődleges kulcs)
2. groupID: csoport azonosítója akikhez az óra lett rendelve
3. start\_Hour: óra kezdete órában (óra:perc)
4. start\_Minute: óra kezdete percben (óra:perc)
5. length: óra hossza
6. day: melyik napon van az óra (hétfő,kedd, stb…)
7. subjectName: óra tantárgya

* marks: jegyeket rögzíti

1. ID: azonosító (elsődleges kulcs)
2. teacherID: jegyet naplózó tanár
3. studentID: jegyet kapó tanuló
4. Value: jegy értéke (1,2,3,4,5)
5. Multiplier: hány százalékos a jegy
6. subjectName: tantárgy amiből kapta a jegyet
7. date: jegy beírásának dátuma

* messagereceivers: kapcsoló tábla (üzenet-felhasználó)

1. MessageID: üzenet azonosító (idegenkulcs)
2. UserID: felhasználó azonosító (idegenkulcs)

* messages: üzeneteket rögzíti

1. ID: azonosító (elsődleges kulcs)
2. senderUserID: felhasználó azonosító (idegenkulcs)
3. message: üzenet tartalma
4. date: üzenet küldésének dátuma

* studentgroups: kapcsoló tábla (csoport-tanuló)

1. GroupID: csoport azonosító (idegenkulcs)
2. StudentID: tanuló azonosító (idegenkulcs)

* students: tanuló fiókok adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. name: tanuló teljes neve
3. DoB: születési dátum
4. address: tanuló lakcíme
5. phone: tanuló telefon száma
6. email: tanuló email címe
7. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója
8. OMID: oktatási azonosító

* teachers: tanár fiókok adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. name: tanár teljes neve
3. phone: tanár telefon száma
4. email: tanár email címe
5. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója

* users: fiókok bejelentkezési adatait rögzíti

1. ID: azonosító (elsődleges kulcs)
2. username: felhasználónév
3. password: jelszó
4. role: felhasználó szerepköre (diák,tanár,szülő,admin)

A képen szöveg, képernyőkép, diagram, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Adatbázis megtervezése

Igyekeztünk minél jobban megtervezni az adatbázist hogy később a fejlesztés során ne hátráljon minket az adatbázis megoldás módja, így több helyen is más megoldásokkal kellett feljönnünk mint amit eredetileg kigondoltunk

### Felhasználó rendszer

A rendszerünk alapköve a 4 felhasználó fajta (diák,tanár,szülő,admin), tehát úgy kellett megterveznünk ezeket a táblákat hogy könnyedén validálni lehessen a bejelentkezéseket amellett hogy különböző fajta felhasználókat is kezelünk, így született meg a „users” tábla ami csak a bejelentkezési adatokat menti le egy adott fiókhoz, és validálás után a felhasználó „role” mezője alapján letudja kérdezni az adott felhasználó adatait.

### Órák

Óráknál nem akartuk az összes létező dátumra lementeni az órákat külön-külön, így csak egy „hétnyi” órát mentünk le, tehát nem kell annyi órával dolgozni. Helyettesítéseket és óra módosulásokat pedig a „classdisruption” táblán keresztül kezeljük le, ahol meglehet adni adott dátumra az ilyen információkat, amit azután frontenden a rendes órarenddel össze kombinálva meglehet jeleníteni minden hétre, duplikálva az órákat és utána rájuk húzva a módosulásokat

## Tesztelés

Teszteléshez a Jest segítségét hívtuk ami adott nekünk egy keretrendszert a backend teszteléséhet, a projekt fájl mappájában található egy „tests” nevű mappa ami tartalmazza a teszteket, a tesztek külön-külön fájlokba vannak rendezve „.test.js” végződéssel

Az adatbázis szimulálva van a projekt fájlban megadott táblák és kapcsolatok alapján, és az adataik nem mentődnek az egyes teszt futtatások után, hanem mindig újra legenerálódnak és feltöltésre kerülnek az szimulált adatbázisba

Minden egyes tesztnél az adott felhasználónak aki „szemszögéből” végezzük a kérést, annak generálunk egy token-t amit beállítunk a kérés fejlécében a kérés előtt

A képen szöveg, Betűtípus, képernyőkép, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az oldal egyes főbb funkcióit teszteljük le, amik így gyorsíthatják a hibák megkeresését

Az összes tesztfájlban a teszteknek megvan adva szövegesen hogy milyen tesztek és hogy mit tesztelnek le

### Hiányzás tesztek

Leteszteli a hiányzások lekérdezését az első két tesztben a végpontokon keresztül, a másik két teszt pedig a feltöltésüket, ha jól és egy rosszul megformázott objektumot küldesz fel

A várt és felküldött objektumok megegyeznek:

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Jegy tesztek

Leteszteli a jegyek lekérdezését az első két tesztben a végpontokon keresztül (diák és tanár oldalról is), a másik két teszt pedig a feltöltésüket, ha jól és egy rosszul megformázott objektumot küldesz fel

A várt és felküldött objektumok megegyeznek:

A képen szöveg, képernyőkép, menü látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Rendszergazda funkció tesztek

Leteszteli az összes üzenet lekérését és egy üzenet törlését, mindkettőnél nézi hogy egyezik e a lekérdezett adatok hossza az elvárttal

Lekérdezés:

A képen szöveg, Betűtípus, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, Betűtípus, képernyőkép, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

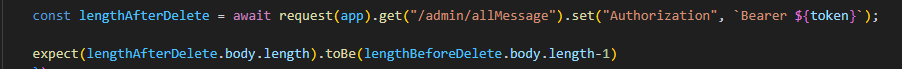
A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Törlés:

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.



### Feladat funkciók tesztelése

Teszteli a feladatokkal kapcsolatos funkciókat tanári, illetve diák oldalról: feladatok lekérdezése, törlése, leadása, módosítása

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Üzenet funkciók tesztelése

Leteszteli az üzenetek lekérdezését, küldését, és egyes hiba eseteket

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Órarend tesztelése

Leteszteli hogy az órarend vissza adja e megfelelően az adott felhasználónak az óráit

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.