

Páholy vizsgaremek dokumentáció



Készítette:

Agócs Dániel

Kovács Ákos

Havasi Alex Márton

Páholy felhasználó dokumentáció

Tartalom

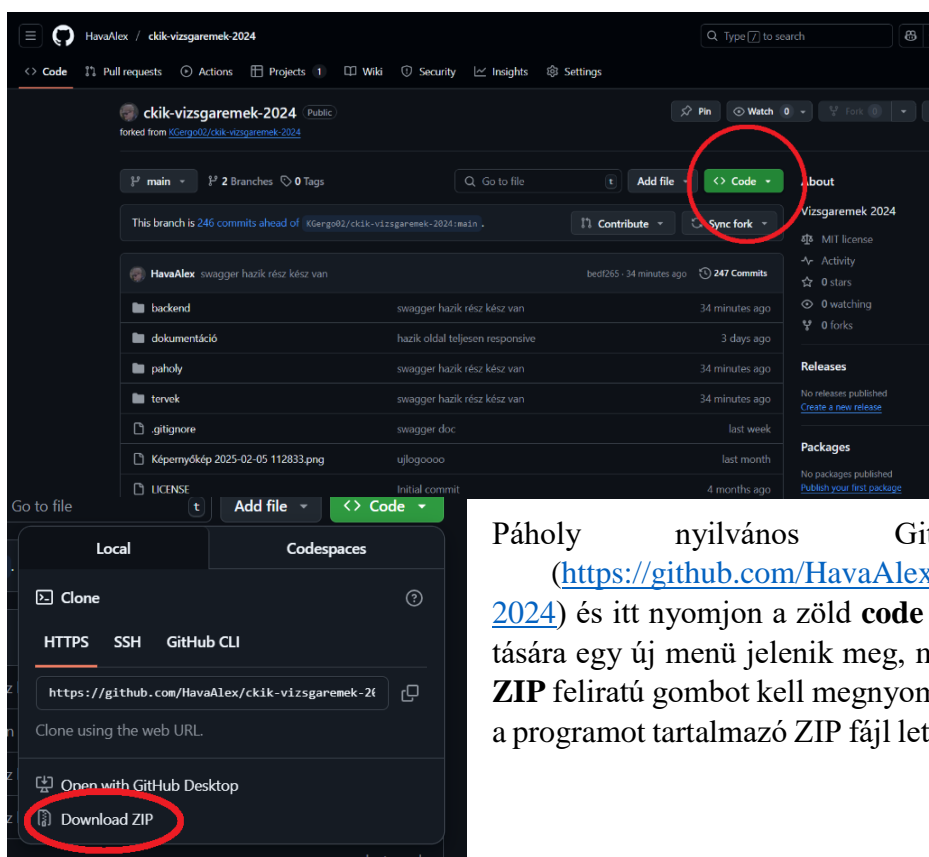
Páholy felhasználó dokumentáció	2
Program ismertetése	3
Program indítása és futtatása	3
Bejelentkező oldal:	6
Az oldal felépítése	7
Üzenetek felület	7
Házi feladat felület	9
Órarend felület	11
Érdemjegyek felület	12
Hiányzások felület	13
Felhasználók kezelése felület	14
Csoportok kezelése felület	15
Program leállítása, program törlése	15
Fejlesztői dokumentáció	16
Frontend	16
Projekt fájlrendszer elrendezése	16
Kiegészítők	17
Oldal felépítés	17
Lekérdezések	17
Adatok	18
Fajták	18
Visszajelzés	18
API	19
Projekt fájlrendszer elrendezése	19
Kiegészítők:	19
Végpont dokumentáció	19
Adatbázis	20
Táblák	20

Adatbázis megtervezése	24
Felhasználó rendszer	24
Órák	24
Tesztelés	25
Hiányzás tesztek	25
Jegy tesztek	27
Rendszergazda funkció tesztek	28
Feladat funkciók tesztelése	29
Üzenet funkciók tesztelése	30
Órarend tesztelése	31

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

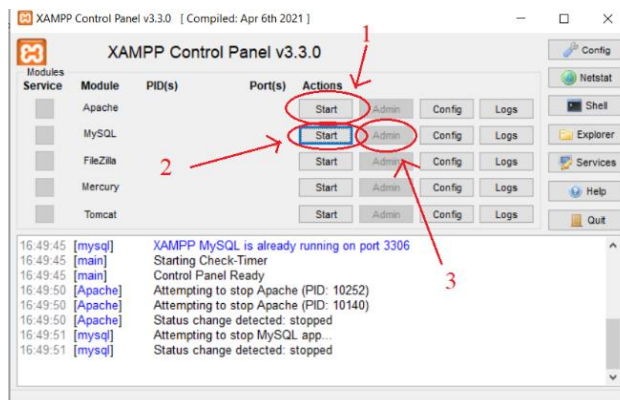


A Páholy rendszer futtatásához szükséges az XAMPP illetve a node.js telepítése előzetesen. Amennyiben nem áll rendelkezésére, vagy esetleg elveszett a Páholy rendszer forráskódja, menjen a

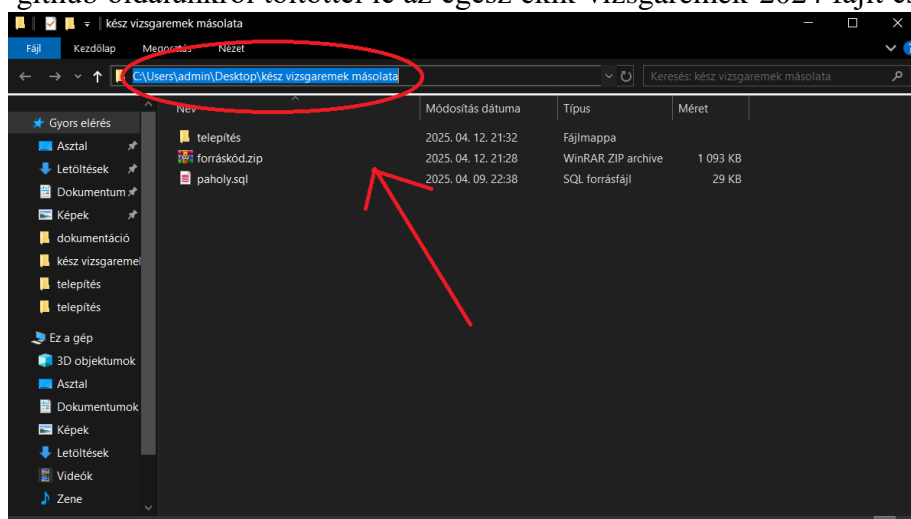
Páholy nyilvános Github oldalára (<https://github.com/HavaAlex/ckik-vizsgaremek-2024>) és itt nyomjon a zöld **code** gombra. Ennek hatására egy új menü jelenik meg, melyen a **Download ZIP** feliratú gombot kell megnyomni, hogy elinduljon a programot tartalmazó ZIP fájl letöltése. Miután a fájl

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 rendelkezésre 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 középen felül található **Importálás** feliratú gombra kell kattintania. Ennek hatására egy új felület nyílik meg ahol az importálandó fájl menüpontban fel tudja tölteni a paholy.sql fájlt. Miután ez megtörtént, görgessen le majd a középen alul található „Importálás” feliratú gombot nyomja meg. Ennek hatására az adatbázis feltöltődik táblákkal illetve példa felhasználókkal, csoportokkal, így lehetővé válik az oldal használata. A próbafelhasználók neveit és jelszavait a fejezet végén listázva találja meg.



Ezek után csomagolja ki a forráskódot tartalmazó forraskod.zip fájlt (amennyiben a github oldalunkról töltötte le az egész cik-vizsgaremek-2024 fájlt csomagolja ki). Ezek után



kattintson a fájl elérési útvonalát jelző szövegdozra (amennyiben a github oldalunkról töltötte le nyissa meg a kicsomagolt mappát és ott tegye ezt). Ennek hatására az kijelölődik. Itt a backspace gombbal törölje ki a meglévő szöveget, majd írja be

a következőt: „cmd” majd nyomja meg az Enter billentyűt. Ennek hatására megnyílik a parancssor. Fontos, hogy az oldal futtatása 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 következők: „cd backend”, „npm start”. Mindezek után nyisson meg egy új parancssort a fájl elérési útvonalára kattintva, ugyanabban a mappában mint az előbb. Első indítás esetén az alábbi parancsokat írja be: „cd paholy”, „npm i”, „npm run build”, „npm run preview”. Amennyiben nem elsőre futtatja a programot 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 Páholly gyakorlati iskolai használatra van tervezve, regisztrációra nincs lehetőség. Az új felhasználók, csoportok illetve tanórák létrehozása a rendszergazda feladata. Új felhasználók 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:

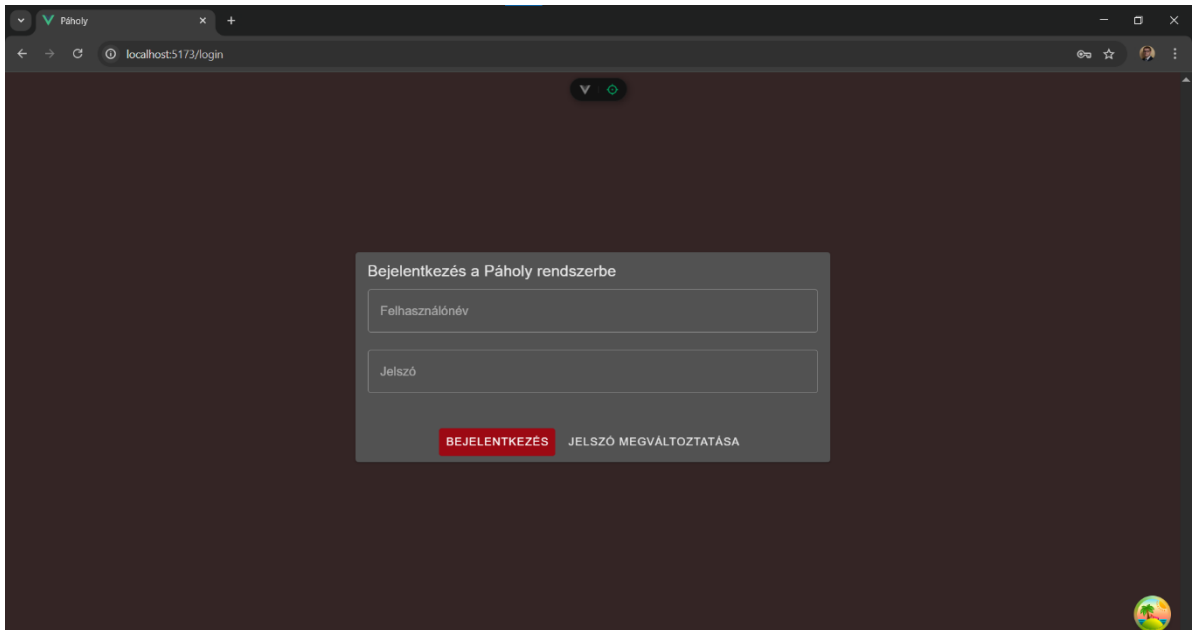
- papp.lajos
- ceglesti.eniko
- kovacsne.hajdu.borbala

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. Ezekkel a fájlokkal szemléltethető az új felhasználók, csoportok és tanórák nagymennyiségű feltöltésének működése. A rendszer txt, csv illetve xlsx típusú fájlokat fogad el.

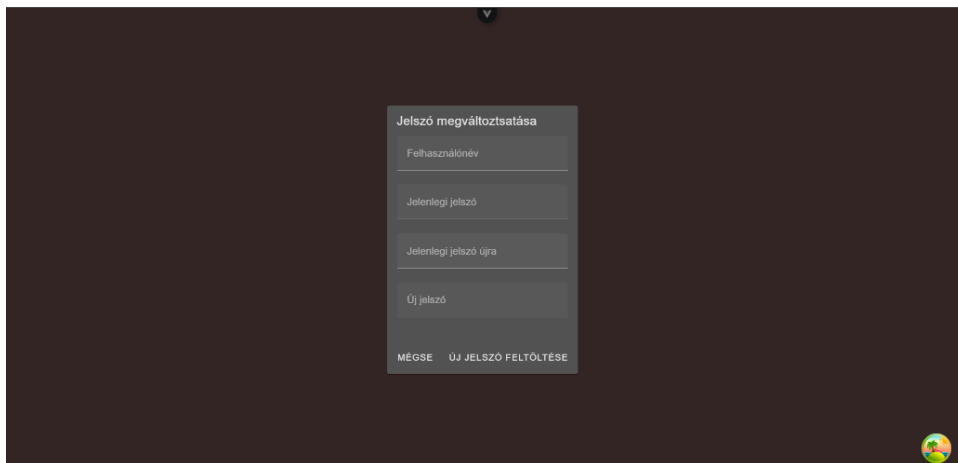
Bejelentkező oldal:

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.



The screenshot shows a web browser window with the address bar displaying 'localhost:5173/login'. The page has a dark background. In the center, there is a light gray box titled 'Bejelentkezés a Páholy rendszerbe'. Inside this box, there are two input fields: 'Felhasználónév' (Username) and 'Jelszó' (Password). Below these fields are two buttons: a red button labeled 'BEJELENTKEZÉS' and a gray button labeled 'JELSZÓ MEGVÁLTOZTATÁSA'. A small circular icon is visible in the bottom right corner of the page.

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 felhasználó-



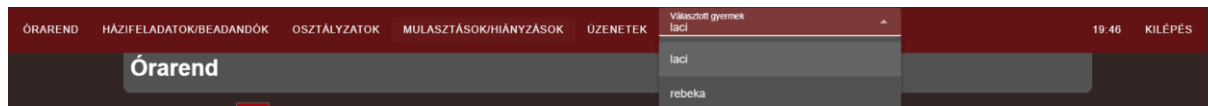
The screenshot shows a web browser window with a dark background. In the center, there is a light gray box titled 'Jelszó megváltoztatása'. Inside this box, there are four input fields: 'Felhasználónév' (Username), 'Jelenlegi jelszó' (Current password), 'Jelenlegi jelszó újra' (Confirm current password), and 'Új jelszó' (New password). Below these fields are two buttons: a gray button labeled 'MÉGSE' (Cancel) and a gray button labeled 'ÚJ JELSZÓ FELTÖLTÉSE' (Save new password). A small circular icon is visible in the bottom right corner of the page.

nak 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 lejártá 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.

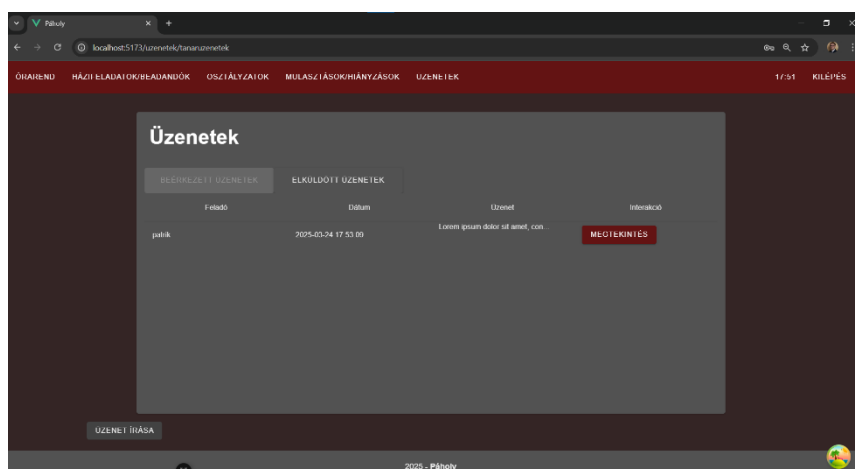


Legvégül minden oldal alján egy lábléc található, melyben az aktuális év és a Páholy név olvasható.



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 megtekinthetők az éppen bejelentkezett felhasználó számára bejövő és az általa küldött üzenetek is. Mindkét esetben megtekinthetők az üzenetek részletei egy felugró ablakban, mely a



Megtekintés feliratú gombra kattintva jelenik meg. Ebben az ablakban látható az üzenet küldőjének felhasználóneve, a többi címzett felhasználó-

neve, a küldés dátuma, illetve a teljes szöveg, mely táblázat nézetben lerövidítve jelenik meg.

Üzenet

Címzettek:

CÍMZETTEK HOZZÁADÁSA:

Az üzenet szövege

MÉGSE ÜZENET ELKÜLDÉSE

tintva az üzenet küldése megszakad és az ablak bezáródik. A **címzettek hozzáadása** gombra kattintva lenyílik egy menü, melyben kiválaszthatjuk 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 Páholly 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

Üzenetek

BEÉRKEZETT ÜZENETEK

ELKÜLDÖTT ÜZENETEK

ÖSSZES ÜZENET

Feladó	Dátum	Üzenet	Interakció
patrik	2025-03-24 17:53:09	Lorem ipsum dolor sit amet, con...	MEGTEKINTÉS TORLÉS
lajos	2025-03-24 17:54:41	Oszálykirándulás részletek -In...	MEGTEKINTÉS TORLÉS

Címzettek:

rebeka (diak) (kattintson az eltávolításhoz)

eniko (tanar) (kattintson az eltávolításhoz)

1.b (csoport) (kattintson az eltávolításhoz)

CÍMZETTEK HOZZÁADÁSA:

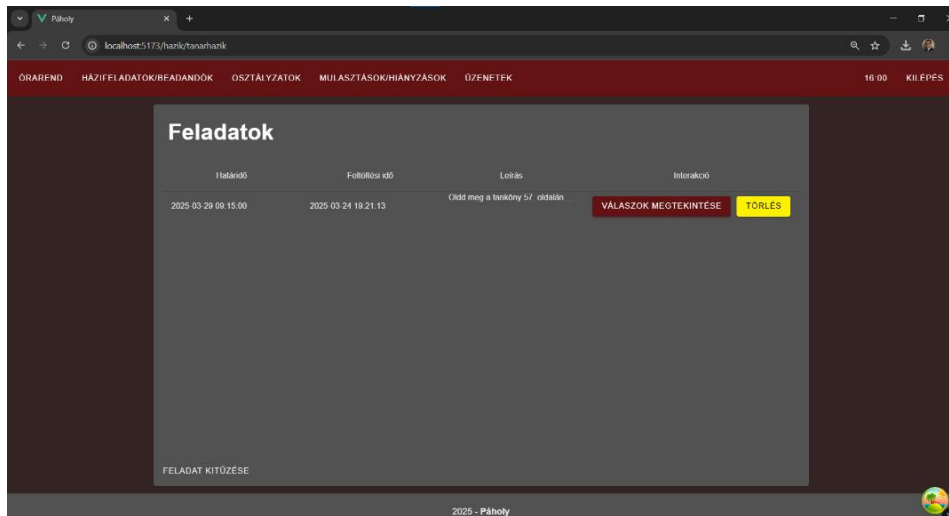
Az üzenet szövege

Üzenet írására a gondvi-selő, 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 kat-

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 oldaláról 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

The screenshot shows the 'Új feladat' (New Homework) form. It has a 'Címzett osztály' (Class) section with radio buttons for '1.b' and '1.g'. Below this is a 'SELECT DATE' section with a calendar view for March 2025, showing dates from 1 to 31. To the right of the calendar is a 'SELECT TIME' section with a circular clock interface. Below the date and time selection, there are input fields for 'Óra' (Hour) and 'Perc' (Minute). At the bottom, there's a 'Fájlok feltöltése (egyszerre tölts fel)' (Upload files (upload at once)) section. The form also includes a 'FELADAT KÜLDÉSE' (Send Homework) button and a 'BEZÁRÁS' (Close) button.

kapcsolatos interakciókat. A „Válaszok megtekintése” feliratú gombra kattintva egy felugró ablakban tekinthető meg az adott feladat részletes leírása, mellékelt fájllai, 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

Feladat:

Feladás dátuma: 2025-03-24 19:28:27
 Határidő: 2025-03-30 09:05:00
 Feladat leírása: Petőfi Sándor SZEPTEMBER VÉGÉN című művét elolvasni!
 Fájlok:
 Szeptember vÃgÃn.txt
 (A fájli letöltéséhez kattintson)

Válaszok:

kovacs.laci
Státusz: Leadva
 Válasz szövege: Rendben tanár úr!
 Fájlok: Nincsenek fájlok ehhez a válaszhhoz.

molnar.rebeka
Státusz: Leadva
 Válasz szövege: Rendben tanár úr! Nézzé kérem, hogy haladok!
 Fájlok:

versmondás.mp4
 (A fájli letöltéséhez kattintson)

antal.andris
Státusz: Nincs leadva
 Válasz szövege:
 Fájlok: Nincsenek fájlok ehhez a válaszhhoz.

szabo.daniel.elemer

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. Továbbá figyelmeztet, hogy eleve lejárt határidőt adjon meg a felhasználó.

Diák illetve szülő oldalról szintén táblázatból tekinthetők meg a házi feladatok adatai. Először a feladó tanár neve, aztán a határidő, feladási idő, rövidített 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 megtekinthetők a feladat adatai részletesen, illetve a válasz szövege, mely szerkeszthető, és a feltöltött fájlok, melyek utólag törölhetők és letölthetők.

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.

Címzett osztály

☒ 1.b
☐ 1.g

Feladó tanár	Feladás dátuma	Leírás	Határidő	Interakció
ceglédi eniko	2025-03-20 09:41:11	Fess egy avant garde festményt!	2025-03-20 09:44:00	RÉSZLETEK TORLÉS
papp lajos	2025-03-24 19:21:13	Oldd meg a tankönyv 57. oldalán ...	2025-03-29 09:15:00	RÉSZLETEK TORLÉS
papp lajos	2025-03-24 19:28:27	Petőfi Sándor SZEPTEMBER V...	2025-03-30 09:05:00	RÉSZLETEK TORLÉS
ceglédi eniko	2025-03-24 19:52:46	Rajzolj egy kubista stílusú lábrólpet	2025-03-24 19:55:00	RÉSZLETEK TORLÉS

2025 - Páholy

Házifeladatok:

Feladó tanár	Határidő	Feltöltési idő	Leírás	Státusz	Interakció
ceglédi eniko	2025-03-20 09:44:00	2025-03-20 09:41:11	Fess egy avant gar...	Határidő lejárt	FELADAT MEGTEKINTÉSE
papp lajos	2025-03-29 09:15:00	2025-03-24 19:21:13	Oldd meg a tanköny ...	Nincs leadva	FELADAT MEGTEKINTÉSE
papp lajos	2025-03-30 09:05:00	2025-03-24 19:28:27	Petőfi Sándor SZE...	Leadva	FELADAT MEGTEKINTÉSE
ceglédi eniko	2025-03-24 19:55:00	2025-03-24 19:52:46	Rajzolj egy kubista...	Határidő lejárt	FELADAT MEGTEKINTÉSE

Feladat részletei

Feladás dátuma: 2025-03-24 19:21:13
Dátum: 2025-03-29 09:15:00
Leírás: Old meg a tankönyv 57. oldalán az 1. és 2. feladatot.
Fájlok: Nincsenek fájlok ehhez a feladathoz.
Beadott fájlok:
Megoldás.txt
LETÖLTÉS TORLÉS

Az ön válasza:
Mellékeltem a válaszokat

Fájlok feltöltése (egyszerre tölts fel)

Utóljára módosítva: 2025-03-24 20:01:18

BEZÁRÁS MÓDOSÍTÁS

Rendszergazda típusú felhasználóknak lehetőségük van minden osztály minden feladatát megtekinteni. Egy baloldalt megjelölő 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

Órarend felület képernyőképe. A felület tetején a navigációs sáv látható: ÓRAREND, HÁZIFELADATOK/BEADANDÓK, OSZTÁLYZATOK, MÚLASZTÁSOK/HIÁNYZÁSOK, ÜZENETEK. A fő tartalom az Órarend táblázat, amely a 2025-03-24 - 2025-03-30 időtartamra mutat. A táblázat fejléce: Hétfő, Kedd, Szerda, Csütörtök, Péntek. A táblázatban az órák időpontjai (18:00, 19:00, 20:00) és a tanár neve (Papp Lajos) látható. A táblázat jobb oldalán a KÖVETKEZŐ HÉT gomb látható.

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 könnyebben megjeleníthető legyen a hiányzás illetve a „különleges óra”. A Páholly rendszere minden az alap órarendtől eltérő órát, így a helyettesített órákat, elmaradó órákat, illetve egyéb foglalkozásokat is egyéb óráként kezeli. Ezek megjelenítése színben is eltér a szokásos órákétól. Pirossal a rendszeresen zajló, ismétlődő órák láthatóak, míg sárgával a rendkívüli órákat jelöljük.

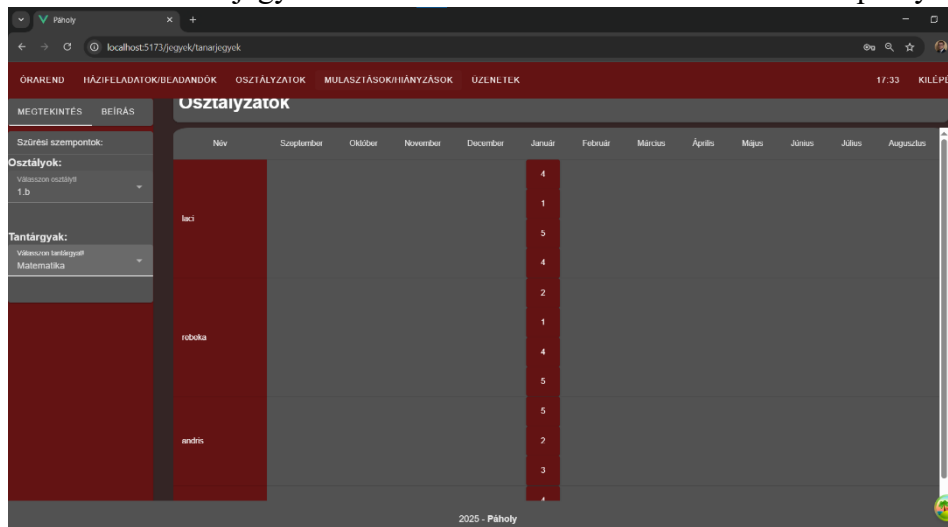
Matematika Tanár: cegledi.eniko
Matematika Tanár: papp.lajos

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 nevét, a csoport nevét, kezdő órát és percet, az óra hosszát, napját illetve magát a tantárgyat.

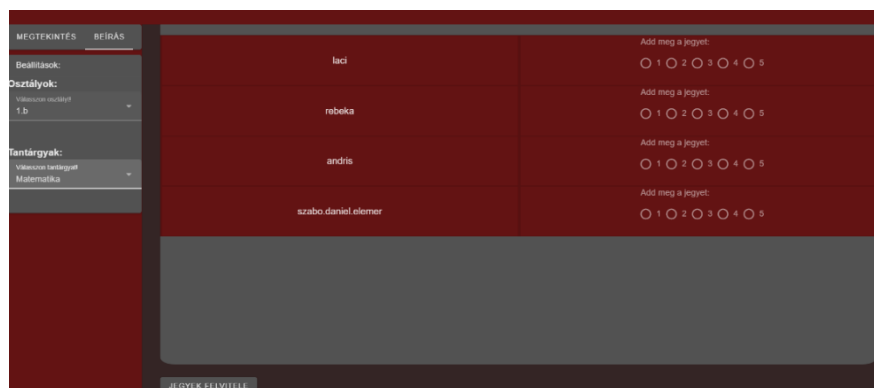
Érdemjegyek felület

Tanári oldalról a jegyek felület az alábbi módon kezelendő: A képernyő bal oldalán található



egy két gombból álló menü. Alap esetben 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 tan-

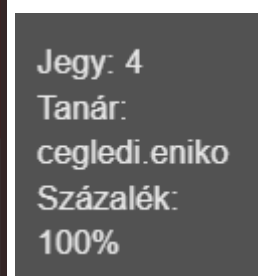
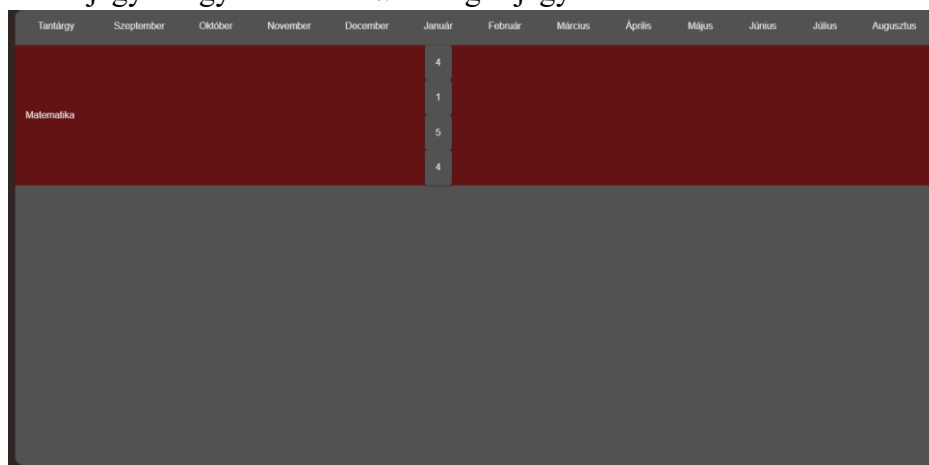
tárgyban. Az egeret a jegy fölé húzva megtekinthetők annak adatai. A **beírás** gombra kattintva



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 felvitel**e feliratú gombbal van lehetőség. Sikeres feltöltés esetén a rendszer visszajelez egy üzenet formájában.

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

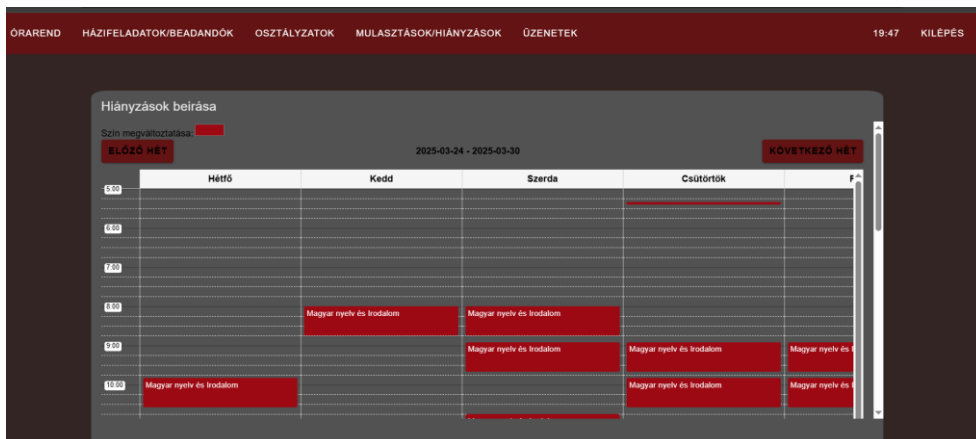


A rendszergazdának lehetősége van a saját oldalán

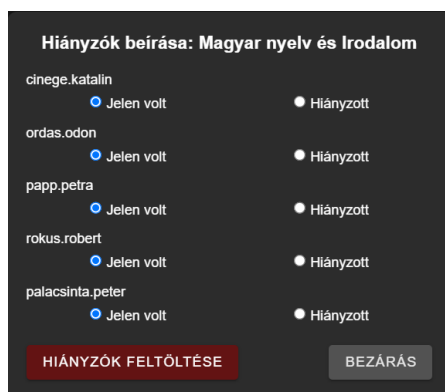
osztály illetve tantárgy alapján szűrni és megtekinteni a jegyeket, akárcsak a pedagógusoknak viszont neki joga van törölni is azokat a bal egérgombbal való kattintással.

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 lenyílómenüben 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.

Hiányzások						
2025. 03. 25. (3 hiányzás)						
OMID	Diák neve:	Tantárgy	Időtartam	Tanár	Igazolás státusza	Interakció
1231231	litvan.jaci	N/A	N/A	1	Igazolatlan	<button>LEIGAZOLÁS</button> <button>TÖRLÉS</button>
1231232	uborka.ubul	N/A	N/A	1	Igazolatlan	<button>LEIGAZOLÁS</button> <button>TÖRLÉS</button>
1231233	kalapacs.karoly	N/A	N/A	1	Igazolva	<button>IGAZOLÁS VISSZAVONÁSA</button> <button>TÖRLÉS</button>

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

Felhasználók kezelése:

Felhasználók száma: 68

Keresés felhasználónév alapján

Név ▲	Azonosító	Szerepkör	Interakció
abigail adams	47	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
abigail evans	42	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
alex	7	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
andrew lewis	49	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
andris	68	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
angyal denes	61	tanár	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE
asd	67	diák	FELHASZNÁLÓ MÓDOSÍTÁSA FELHASZNÁLÓ TÖRLÉSE

Új felhasználók hozzáadása

[DIÁKOK HOZZÁADÁSA](#) [TANÁROK HOZZÁADÁSA](#) [GONDVISELŐK HOZZÁADÁSA](#)

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 adatoknak 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

Diákok hozzáadása

Diákok manuális hozzáadása:

Név

Születési dátum
éééé. hh. nn. 📅

Cím

Telefon

Email

OM azonosító

DIÁK HOZZÁADÁSA

Diákok fájlból történő feltöltése
Kizárólag txt, csv és xlsx fájlok tölthetők fel!
(A txt fájlban az adatokat tabulátorral elválasztva kell megadni)

📎 Fájlok feltöltése (egyszerre)

FÁJLOK BEOLVASÁSA

Bekerülő diákok listája:

[BEZÁRÁS](#) [FELTÖLTÉS AZ ADATRÁZISBA](#)

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

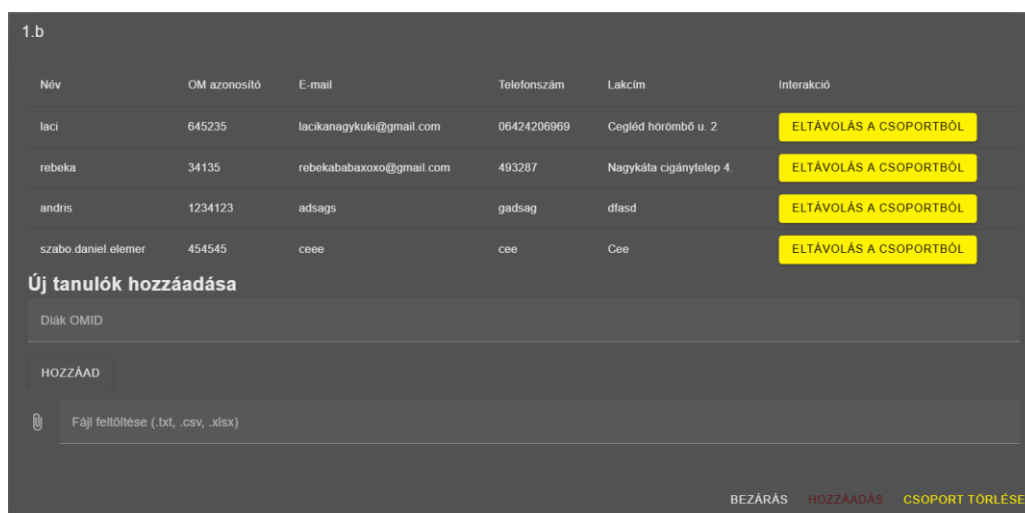
A küldés gomb megnyomása után egy txt file automatikusan letöltődik. Ebben a fájlban szerepelnek az újonnan létrehozott felhasználók felhasználónevei és jelszavai. Ezeket a rendszergazda juttatja el az illetékeseknek.

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 hozzáadására 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 szélén 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

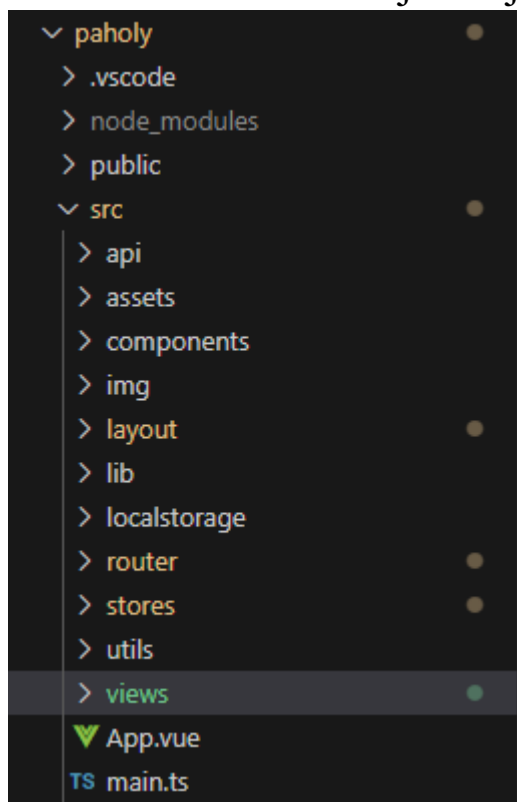
Az adott felhasználó lekérdezheti ezekkel kapcsolatban az adatait amit az oldal megjelenít egy rendezett módon, és akár fel is tölthet új adatokat (pl tanár beír egy jegyet)

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



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 (tan-stack)
- 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 fajtaához 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)
- hiányzasok: 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 átirva, 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 megjelenítést v-forral, sok táblázat cellái megvannak magában az oldalban írva egyszer, mint egy sablon, beolvassa a lekérdezett adatot és megjeleníti adattagonként

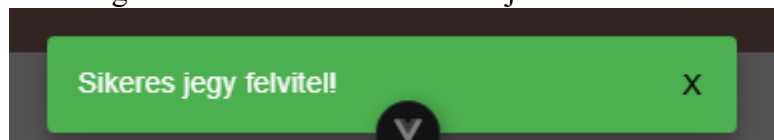
A Vue egyik fontos része a „ref” megnevezés amivel egy referencia objektumot hozhatunk létre, ezért nagyon hasznos új adatok felvitelénél még a lokális tároláshoz, majd abból kiolvasásához, és akár több helyről is belehet olvasni ugyan azt az adatot úgy hogy ha változik akkor az a többi helyen is megváltozik

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 viz-szakapott adatok fajtáinak deklarálása

Visszajelzés

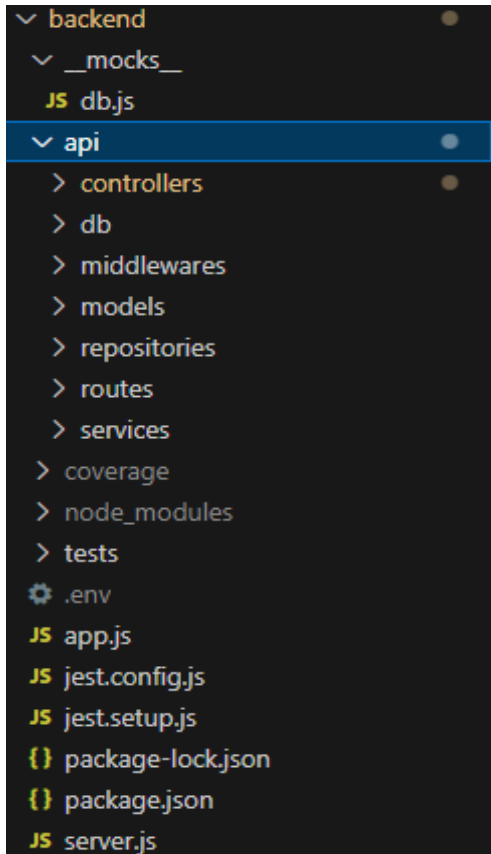
Az „errorHandler” és „statusHandler” tároló felel a lekérdezések visszajelzésére, ami bárhol meghívható és a „setError” és „setStatus” függvénnyel beküldhető egy szöveg amit megjelenít az oldal alján. Az App.vue-ban helyezkedik el a megjelenítő kód hogy bárhol látszódjon, akár még aloldal váltás után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



API

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

Projekt fájlrendszer elrendezése



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 kötetései
- `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

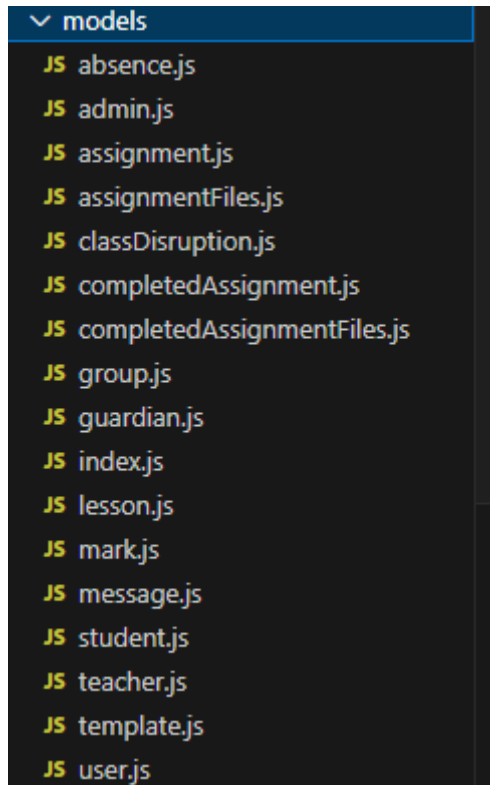
Végpont dokumentáció

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



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

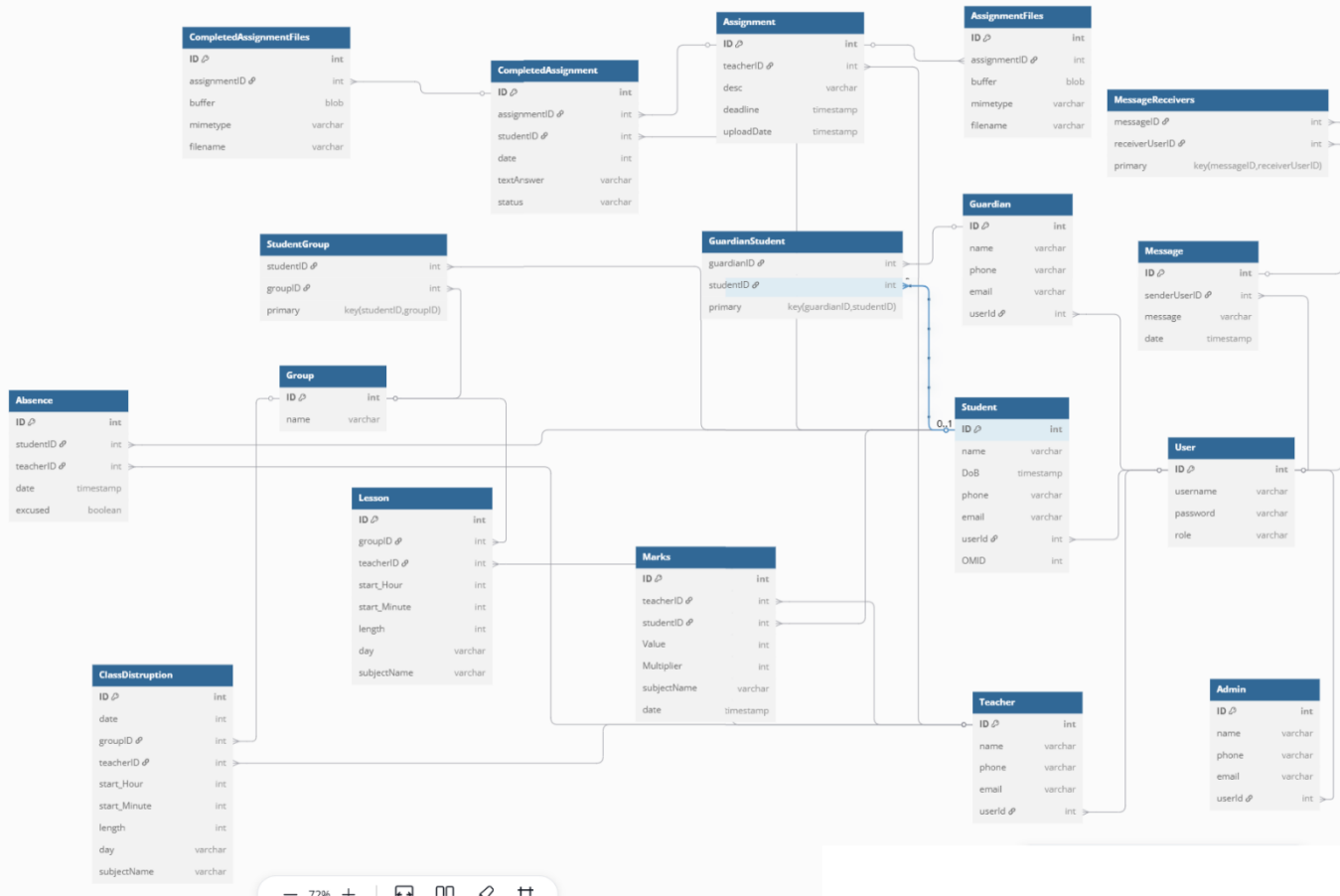
Táblák rövid leírásai:

- 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
 6. ID: azonosító (elsődleges kulcs)
 7. date: helyettesített/módsítás dátuma
 8. groupID: csoport azonosítója akikhez az óra lett rendelve
 9. start_Hour: óra kezdete órában (óra:perc)
 10. start_Minute: óra kezdete percben (óra:perc)
 11. length: óra hossza
 12. day: melyik napon van az óra (hétfő,kedd, stb...)
 13. subjectName: óra tantárgya
- completedassignments: feladatokra beadott választ rögzíti
 6. ID: azonosító (elsődleges kulcs)
 7. assignmentID: feladat azonosítója amihez tartozik a válasz
 8. studentID: feladatot leadó tanuló azonosítója
 9. date: feladat leadásának dátuma
 10. textAnswer: leadott válasz szöveges tartalma
 11. status: leadás státusza (leadva, nincs leadva, határidő lejárt)
- completedassignmentfiles: elkészült feladatokhoz csatolt fájlokat rögzíti
 14. ID: azonosító (elsődleges kulcs)
 15. assignmentID: feladat azonosítója amihez tartozik a fájl

- 16. buffer: fájl
- 17. mimetype: fájl kiterjesztése
- 18. filename: fájl neve

- groups: csoportokat rögzíti
 - 19. ID: azonosító (elsődleges kulcs)
 - 20. name: csoport neve
- guardians: szülő fiókok adatait rögzíti
 - 6. ID: azonosító (elsődleges kulcs)
 - 7. name: szülő teljes neve
 - 8. phone: szülő telefon száma
 - 9. email: szülő email címe
 - 10. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója
- guardianstudents: kapcsoló tábla (szülő<gyerek)
 - 11. GuardianID: szülő azonosító (idegenkulcs)
 - 12. StudentID: gyerek(diák) azonosító (idegenkulcs)
- lessons: órákat rögzíti
 - 21. ID: azonosító (elsődleges kulcs)
 - 22. groupID: csoport azonosítója akikhez az óra lett rendelve
 - 23. start_Hour: óra kezdete órában (óra:perc)
 - 24. start_Minute: óra kezdete percben (óra:perc)
 - 25. length: óra hossza
 - 26. day: melyik napon van az óra (hétfő,kedd, stb...)
 - 27. subjectName: óra tantárgya
- marks: jegyeket rögzíti
 - 28. ID: azonosító (elsődleges kulcs)
 - 29. teacherID: jegyet naplózó tanár
 - 30. studentID: jegyet kapó tanuló
 - 31. Value: jegy értéke (1,2,3,4,5)
 - 32. Multiplier: hány százalékos a jegy
 - 33. subjectName: tantárgy amiből kapta a jegyet
 - 34. date: jegy beírásának dátuma
- messagereceivers: kapcsoló tábla (üzenet-felhasználó)
 - 13. MessageID: üzenet azonosító (idegenkulcs)
 - 14. UserID: felhasználó azonosító (idegenkulcs)
- messages: üzeneteket rögzíti
 - 35. ID: azonosító (elsődleges kulcs)
 - 15. senderUserID: felhasználó azonosító (idegenkulcs)
 - 16. message: üzenet tartalma
 - 17. date: üzenet küldésének dátuma
- studentgroups: kapcsoló tábla (csoport-tanuló)
 - 18. GroupID: csoport azonosító (idegenkulcs)
 - 19. StudentID: tanuló azonosító (idegenkulcs)

- students: tanuló fiókok adatait rögzíti
 - 20. ID: azonosító (elsődleges kulcs)
 - 21. name: tanuló teljes neve
 - 22. DoB: születési dátum
 - 23. address: tanuló lakcíme
 - 24. phone: tanuló telefon száma
 - 25. email: tanuló email címe
 - 26. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója
 - 27. OMID: oktatási azonosító
- teachers: tanár fiókok adatait rögzíti
 - 28. ID: azonosító (elsődleges kulcs)
 - 29. name: tanár teljes neve
 - 30. phone: tanár telefon száma
 - 31. email: tanár email címe
 - 32. userID: felhasználó(user) táblában lévő fiókhoz kötött bejelentkezési adatok azonosítója
- users: fiókok bejelentkezési adatait rögzíti
 - 33. ID: azonosító (elsődleges kulcs)
 - 34. username: felhasználónév
 - 35. password: jelszó
 - 36. role: felhasználó szerepköre (diák,tanár,szülő,admin)



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 feljöttünk mint amit eredetileg kigondoltunk

Felhasználó rendszer

A rendszerünk alapkőve a 4 felhasználó fajta (diák,tanár,szülő,admin), tehát úgy kellett megterveznünk ezeket a táblákat hogy könnyedén validálni lehessen a bejelentkezéseket amellet 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

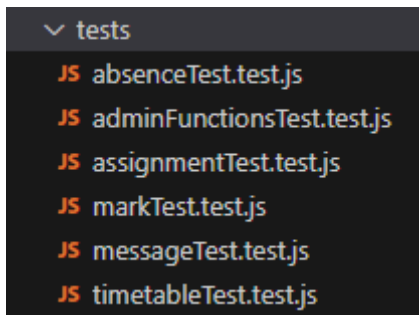
Óráknál nem akartuk az összes létező dátumra lementeni az órákat külön-külön, így csak egy „hétnyi” órát mentünk le, tehát nem kell annyi órával dolgozni. Helyettesítéseket és óra módosulásokat pedig a „classdisruption” táblán keresztül kezeljük le, ahol meglehet adni adott dátumra az ilyen információkat, 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 teszteléséhez, 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



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:

```
newAbsence1 = {  
  ID: 1,  
  teacherID: 1,  
  studentID: 1,  
  lessonID: 1,  
  date: "2025-03-24T09:10:46.626Z",  
  excused: false,  
};
```

```
newAbsence2 = {  
  ID: 2,  
  teacherID: 1,  
  studentID: 1,  
  lessonID: 1,  
  date: "2025-03-25T10:15:30.000Z",  
  excused: false,  
};
```

```
newAbsence3 = {  
  ID: 3,  
  teacherID: 1,  
  studentID: 2,  
  lessonID: 1,  
  date: "2025-03-26T08:00:00.000Z",  
  excused: false,  
};
```

```
newAbsence4_rossz = {  
  ID: "rossz",  
  teacherID: 1,  
  lessonID: 1,  
  date: "2025-03-27T11:45:00.000Z",  
  excused: false,  
};
```

Jegy tesztek

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

A várt és felküldött objektumok megegyeznek:

```
newMark1 =
{
  ID: 1,
  teacherID: 1,
  studentID:1,
  Value:5,
  Multiplier:100,
  date:"2025-03-24T09:10:46.626Z",
  subjectName:"Majom programozás",
};
newMark2 =
{
  ID: 2,
  teacherID: 1,
  studentID:2,
  Value:3,
  Multiplier:100,
  date:"2025-03-24T09:10:46.626Z",
  subjectName:"Majom programozás",
};
newMark3 =
{
  ID: 3,
  teacherID: 1,
  studentID:2,
  Value:1,
  Multiplier:200,
  date:"2025-03-24T09:10:46.626Z",
  subjectName:"Matematika",
};
newMark4_rossz =
{
  ID: 3,
  teacherID: 4,
  studentID:2,
  Multiplier:200,
  date:"2025-03-24T09:10:46.626Z",
  subjectName:"Matematika",
};
```

Rendszergazda funkció tesztek

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

Lekérdezés:

```
message: "Test1",  
date: "2000-01-01",  
receiverlist: [{ ID: user2.ID }],  
receiverGrouplist: []  
});
```

```
.send({  
message: "Test2",  
date: "2000-01-01",  
receiverlist: [{ ID: user2.ID }],  
receiverGrouplist: []  
});
```

```
const res = await request(app)  
  .get("/admin/allMessage").set("Authorization", `Bearer ${token}`);  
  
expect(res.status).toBe(201);  
  
expect(res.body).toBeDefined();  
expect(res.body).toHaveLength(2)
```

Törlés:

```
const createRes = await request(app)  
  .post("/uzenet")  
  .set(setUserHeader())  
  .send({  
    message: "Message to delete",  
    date: "2000-01-01",  
    receiverlist: [{ ID: user2.ID }],  
    receiverGrouplist: []  
  });
```

```
const lengthAfterDelete = await request(app).get("/admin/allMessage").set("Authorization", `Bearer ${token}`);  
  
expect(lengthAfterDelete.body.length).toBe(lengthBeforeDelete.body.length-1)
```

Feladat funkciók tesztelése

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

```
describe("PATCH /feladat/modifycompletedassignment",()=>{
  test("módosítás megtörténik sikeresen",async()=>{
    const modifiedCompletedAssignment = await request(app)
      .patch("/feladat/modifycompletedassignment")
      .set(setUserHeader()).send({
        ID:2,
        assignmentID: goodtestAssignment.ID,
        date: Date.now(),
        status: "Nincs leadva",
        studentID: newStudent2.ID,
        textAnswer: "TestValasz"
      });

    expect(modifiedCompletedAssignment.body.assignmentID).toBe(goodtestAssignment.ID)
    expect(modifiedCompletedAssignment.body.status).toBe("Leadva")
    expect(modifiedCompletedAssignment.body.textAnswer).toBe("TestValasz")
  })
})
```

```
describe("GET haziktanar",()=>{ //korábban már töltötünk fel sikeresen ezért azzal most nem foglalkozunk
  test("Visszadja a tanárnak a feladatait az arra érkezett válaszokkal együtt.",async()=>{
    const sentAssignemnts = await request(app)
      .get("/feladat/haziktanar")
      .set(setUserHeader())

    expect(Array.isArray(sentAssignemnts.body)).toBe(true)
    expect(sentAssignemnts.body[0]).toHaveProperty("feladat")
    expect(sentAssignemnts.body[0]).toHaveProperty("answers")
    expect(sentAssignemnts.body[0].feladat.teacherID).toBe(teacher1.ID)
    expect(sentAssignemnts.body[0].answers.length).toBe(2)

    console.log("tanarnak lejön e valasz kerdem en: ", sentAssignemnts)
    console.log("tanarnak lejön e valasz kerdem en: ", sentAssignemnts.body[0].answers)
  })
})
```

Üzenet funkciók tesztelése

Leteszteli az üzenetek lekérdezését, küldését, és egyes hiba eseteket

```
test("Új üzenet sikeres feltöltése egy csoportnak", async () => {
  const res = await request(app)
    .post("/uzenet")
    .set(setUserHeader())
    .send({
      message: "Test üzenet2",
      date: "2000-01-01",
      receiverlist: [],
      receiverGrouplist: [{ID: newGroup1.ID, studentList: [newStudent1.userId, newStudent2.userId] }]
    });

  expect(res.status).toBe(201);
  expect(res.body).toHaveProperty("ID");
  expect(res.body.senderUserID).toBe(user1.ID);
  expect(res.body.message).toBe("Test üzenet2");
  const messagereceiverData = await messageReceiverRepository.getMessageReceiverByMessageID(res.body.ID)
  expect(messagereceiverData[0].UserID).toBe(user4.ID)
  expect(messagereceiverData[1].UserID).toBe(user5.ID)
});
```

```
describe("GET potentialReceivers", () => {
  test("Leszedi az elérhető címzettet (kivéve önmagát)", async () => {
    const res = await request(app)
      .get("/uzenet/uzenetekreceivers")
      .set(setUserHeader());

    expect(res.status).toBe(201);
    //megvan e minden része
    expect(res.body).toHaveProperty("singleUsers");
    expect(res.body).toHaveProperty("groups");
    expect(Array.isArray(res.body.singleUsers)).toBe(true);
    expect(Array.isArray(res.body.groups)).toBe(true);
  });
});
```

Órarend tesztelése

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

```
newStudentGroups1 =
{
  GroupID:1,
  StudentID:1,
}
newStudentGroups2 =
{
  GroupID:2,
  StudentID:2,
}
newLesson1 =
{
  ID:1,
  groupID:1,
  teacherID:1,
  start_Hour:'5',
  start_Minute:'10',
  length:'45',
  day:'hetfo',
  subjectName:"Majom Programozás"
}
newLesson2 =
{
  ID:2,
  groupID:2,
  teacherID:1,
  start_Hour:'6',
  start_Minute:'11',
  length:'50',
  day:'kedd',
  subjectName:"Majom Programozás csak menőbb"
}
```