

# Dokumentáció

## *TimeTable*

**Mentortanárok:**

Inántszy-Pap Tamás  
Bakti András Tamás

---

---

**Csoport tagjai:**

Dandóczy Máté Attila  
Jakab Mátyás  
Mátyás Martina

---

---

---

# Órarend és nyilvántartó alkalmazás TimeTable

## Dokumentáció

### Készítette:

Dandóczy Máté Attila

Mátyás Martina

Jakab Mátyás

### 1.Szereplők és igényeik:

#### Mi volt az ötlet?

Az ötletünk egy iskolai nyilvántartó rendszer volt, mely adatbázisban tárolja az iskola tanárainak, diákjainak adatait.

Tárolja még a termeket, illetve azok férőhelyeit, az osztályokat, illetve azok névsorait. Tanár, diák(néző) és admin felülettel rendelkezik.

Fejlesztés alatt van még az órarend funkció is.

#### Milyen Céllal jön létre a projekt?

Célunk az volt, hogy megkönnyítsük az Intézmények számára az órarendek, illetve adatok nyilvántartását.

Környezetbarát mivel webes felületen érhető el így nem kell papír formájú órarendet, nyilvántartást használni ennek segítségével is védjük a környezetünket.

Bárhonnan elérhető, így nem hagyjuk el, mint a régen készült papír órarendeket, nyilvántartásokat.

A szoftverünk intézmények számára készült. Célunk egy személyre szabott Órarend készítése volt, melyben diákok, tanárok adatait, illetve tantermek, adatait tárolhatjuk el és tekinthetjük majd meg, valamint osztályok névsorát, órarendjét.

Az intézmények(iskolák) a diákok, tanárok, adatait el tudják tárolni elektronikusan.

A tantermek ajtószáma is szerepelnek a nyilvántartásba, illetve azok férőhelyét is megtekinthetik az alkalmazásban.

Alkalmazásunkban elérhetőek az osztályok listája és az azokban szereplő diákok névsora.

Megtekinthetjük a tanárokhoz tartozó tantárgyakat is.

És amiért készült az alkalmazás az órarend volt, amit a felhasználó osztályonként napokra szedve tekinthet meg.

**Fejlesztési lehetőségek:**

Órarend összeállítása, illetve tárolása

Felhasználók megfelelő kezelése

Loginos rész fejlesztése amikor a tanár az oldalra rálép a saját órarendjére akkor a fejlécbe az URL be a saját OM azonosítója szerinti órarend, illetve adatok jelenjenek meg

Órarend funkció fejlesztés alatt áll. Jelenleg csak Backend szinten tudunk felvenni új órarendet, de már a Frontend oldali elkészítése is folyamatban van.

**Elvárás a projekttől:**

- Statikus oldal
- Bejelentkező menü
- Szerkeszthető órarend (admin)
- Adatok felvétele webes felületen
- Adatok elérése

**Szereplők:**

Az általunk tervezett programhoz háromféle felhasználót szeretnénk.

Ezekhez a felhasználókhöz különféle jogok fognak tartozni.

A szoftverhez 3 féle jog tartozik:

- Néző (Diák)
- Tanár
- Adminisztrátor

## 2.Felhasználók és igényeik:

### Felhasználók:

Az első felhasználónk a **Néző**.

Az alkalmazás indítását követően egyből a nézői oldal jelenik meg.

A felhasználót egyből egy nyitóoldal fogadja, melyen egy rövid leírást talál az alkalmazásról.

A néző felhasználónak(diáknak) alapvető jogai lesznek, amely a lent leírtakat tartalmazza:

- Diákok Listájának megtekintése.
- Termek Listájának megtekintése.
- Tanárok elérhetőségének megjelenítése, valamint tantárgyainak listája.
- Osztályok listája, valamint a benne szereplő diákok listájának megtekintése.
- Tantárgyak listájának megtekintése.
- Az órarendek osztályonkénti megtekintése napokra bontva.

A második felhasználónk a **Tanár**, amelynek hasonló jogai vannak, mint a diáknak.

A tanár felhasználót is egy nyitóoldal fogadja, melyen a tanár oldalhoz kapcsolatos rövid leírás szerepel.

-Funkciói:

- Saját órarendjét megtekintése.
- Saját adatainak megtekintése.
- Fiókjához tartozó jelszó megváltoztatása.

A harmadik felhasználónk az **Adminisztrátor**, nevéből adódóan is ez a felhasználó rendelkezik a legtöbb joggal.

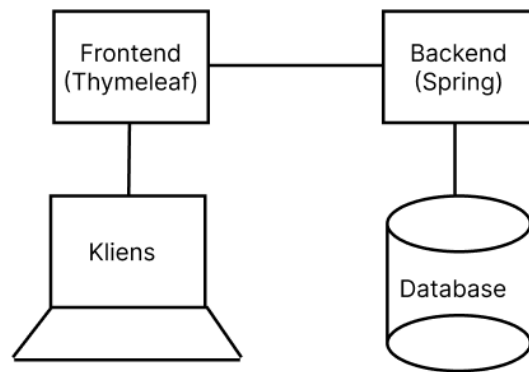
A belépés után a felhasználót itt is egy nyitóoldal fogadja, melyen szintén az oldal rövid ismertetése szerepel.

Az adminisztrátor a következő jogokkal rendelkezik:

- Tanár hozzáadása a következő adatok szerint:
  - OM-azonosító
  - Név
  - Telefonszám
  - Jelszó
  - Admin joggal rendelkezik-e (igen/nem)
- Tanár adatainak módosítása.
- Tanárok listájának megtekintése.
- Tanár törlése.

- Tantárgy hozzáadása a következő adatok szerint:
  - Tantárgy neve
- Tantárgy nevének módosítása.
- Tantárgyak listájának megjelenítése.
- Tantárgy törlése.
- 
- Tanterem hozzáadása a következő adatok szerint:
  - Ajtószám
  - Férőhely
- Tanterem férőhelyének módosítása.
- Tantermek listájának megtekintése.
- Tanterem törlése.
- 
- Diák hozzáadása a következő adatok szerint:
  - OM-azonosító
  - Név
  - Osztály
- Diák adatainak módosítása.
- Diákok listájának megtekintése.
- Diák törlése.
- 
- Osztály hozzáadása a következő adatok szerint:
  - Osztály neve
- Osztály nevének módosítása.
- Osztályok listájának megtekintése.
- Osztály törlése.
- Órarend készítése a következő adatok szerint:
  - Nap
  - Óra
  - Tantárgy
    - Terem
    - Osztály
- Órarend módosítása.
- Órarendek megtekintése.
- Órarend törlése.
- 
- Tantárgy hozzárendelése a tanárhoz.
- Tanárhoz hozzárendelt tantárgy módosítása.
- Tanárhoz hozzárendelt tantárgyak megtekintése.
- Tanárhoz hozzárendelt tantárgy törlése.

### 3. Magas szintű rendszerterv:



Böngésző(kliens)-> Frontend (Thymeleaf)->Web Server Backend (Spring)->Database(MYSQL)

Az **adatbázisunk** a saját gépünk szerverén fut-> ez később megvalósítható webes szerveren keresztül is.

Az adatbázisunk futtatásához a xampp-ot használjuk

**Web Szerverünk** (Backend) az Spring Boot. Ez a localhost:8095 porton fut a saját gépünkön

A **Frontend** az ThymeLeaf. Ez a localhost:9095 porton fut a saját gépünkön.

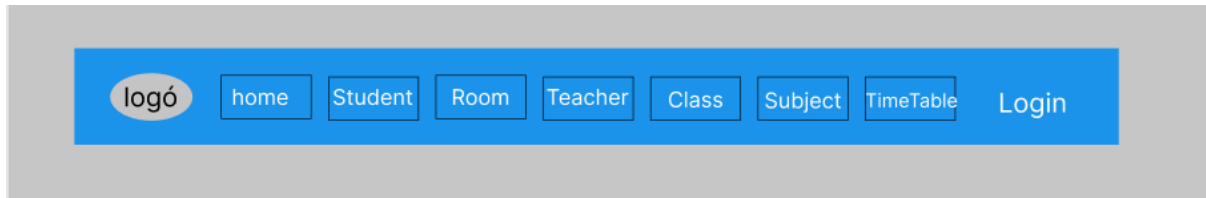
A megjelenítés az általunk választott **böngésző** segítségével jeleníthetünk meg.

#### 4.Screenek:

Oldalak egyenkénti megjelenítésének terve, illetve rövid leírása.

Oldalak kapcsolatainak leírása, navigáció.

Navigációs menü (navbar) View leírása, képernyőkép(elképzelés):



Az oldalt, illetve a megjelenést a kék szín árnyalataival képzeltek el. Az első fontos rész a navigációs menü megtervezése volt. A navigációs menü az oldal legtetején responsive-an fog megjeleneni.

Az első lépések között egy logó megtervezés is szerepelt melyet a Canva oldal segítségével készítettünk el.

A következő lépés az oldalak közötti navigáció megvalósítása volt.

A létrehozott oldalak között gombokkal tudunk navigálni.

A navigációs menüben szerepel még egy bejelentkező fül is.

Navigációs menü(navbar) Teacher leírása, képernyőkép(elképzelés):

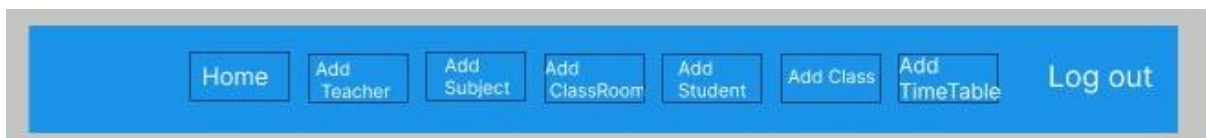
Az oldalt szintén a kék szín árnyalatával terveztük megvalósítani.

Az oldalon szerepelnek a tanár által elérhető oldalak.

Navigációs menü(navbar) Administrator leírása, képernyőkép(elképzelés):

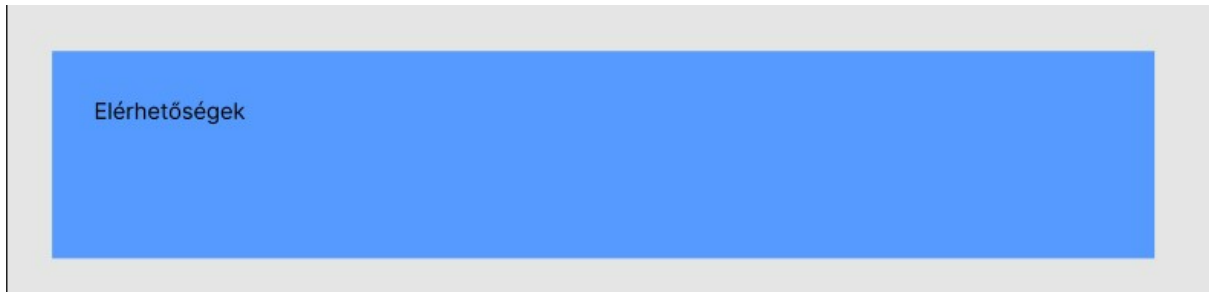
Az oldalt szintén a kék szín árnyalatával terveztük megvalósítani.

Az oldalon szerepelnek az adminisztrátor által elérhető oldalak.



Lábléc(footer) leírása, képernyőkép(elképzelés):

A lábléc az oldal alján jelenik meg responsive-an, a kék szín árnyalatában. Tartalmazza az elérhetőségeinket, melyeket az oldal létrehozásával kapcsolatosan hoztunk létre.



Bejelentkezős oldal (Login) leírása, képernyőkép(elképzelés):

A bejelentkező oldalunkon a felhasználó OM azonosítóját és jelszavát kérjük be. Az oldalunk a Spring Boot által beépített loginos felület lesz.





Kijelentkező oldal (Logout) leírása, képernyőkép(elképzelés):

A bejelentkező oldalunkon a kijelentkezést megerősítő gomb jelenik meg.  
Az oldalunk a Spring Boot által beépített log out-os felület lesz.

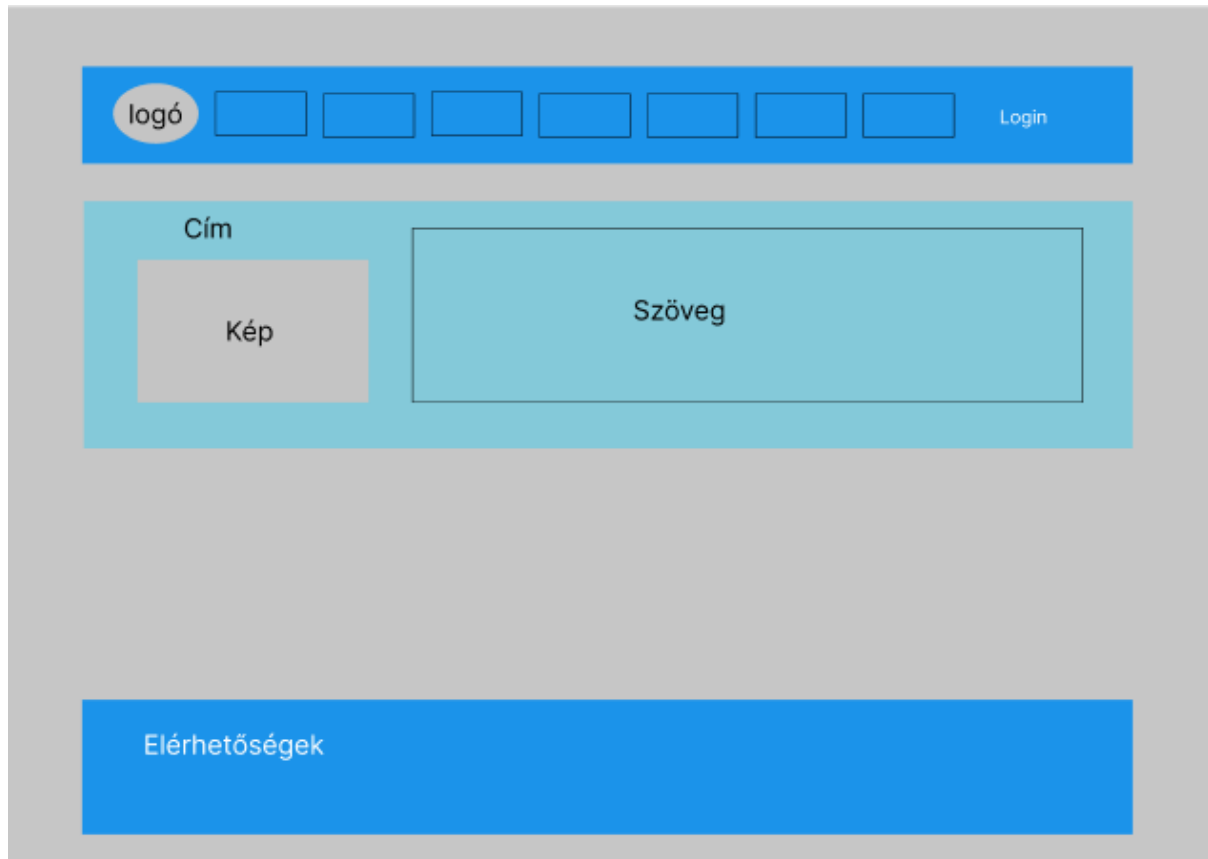
Are you sure want to log out?

Log Out

Kezdőoldal néző(index) leírása, képernyőkép(elképzelés):

A kezdő oldalunkon megjelenik az imént említett navigációs menü, illetve a footer is melyben az elérhetőségeink találhatóak.

Az oldalon az alkalmazásunkkal kapcsolatos információk jelennek meg dobozokban, ami tartalmazza a leírásnak a címét, a leírást és egy látványos képet.



Diákok oldal (Students) leírása, képernyőkép(elképzelés):

A diákok névsora egy táblázatban fog megjelenni.

A táblázat tartalmazza a diákok OM azonosítóját, nevét, illetve osztályát.

Ebbe a táblázatba az összes diák neve szerepelni fog aki az adatbázisban szerepel.

A diákok névsor szerint fognak megjelenni.

The mockup shows a web interface for the 'Students' page. At the top, there is a blue header bar containing a 'logó' (logo) in a grey oval, followed by seven empty rectangular input fields, and a 'Login' button on the right. Below the header, the word 'Tanulók' (Students) is displayed. Underneath, there is a light blue rectangular area containing a table with three columns: 'Azonosító:' (Identifier), 'Név:' (Name), and 'Osztály:' (Class). The first row of the table contains the values '11 számjegyű azonosító' (11-digit identifier), 'Példa Anna' (Example Anna), and '10/b'. Below this table, there is a blue rectangular button labeled 'Elérhetőségek' (Contacts).

Azonosító:	Név:	Osztály:
11 számjegyű azonosító	Példa Anna	10/b

Termek oldal (ClassRooms) leírása, képernyőkép(elképzelés):

A termék oldalon az adatbázisban szereplő termék jelennek meg táblázat formájában.

The mockup shows a web interface for a 'Termek' (Product) page. At the top is a blue navigation bar containing a 'logó' (logo) in an oval, followed by seven empty rectangular boxes, and a 'Login' link on the right. Below the navigation bar, the word 'Termek' is displayed. The main content area is a light blue rectangle containing a table with two columns: 'Terem szám:' (Room number) and 'Hely:' (Location). The first row of data shows '101' and '23' respectively. Below this content area is a solid blue rectangular block with the text 'Elérhetőségek' (Contact information).

Terem szám:	Hely:
101	23

Elérhetőségek

Tanárok oldal (Teachers) leírása, képernyőkép(elképzelés):

A tanárok oldalon a tanárok egyesével külön-külön dobozba fognak megjelenni.

A dobozok tartalmazzák a tanárok adatait:

- Tanár neve
- Tanár OM azonosítója
- Tanár telefonszáma

Az információk alatt egy gomb található majd, mely a tanár tantárgyai oldalára navigál el.

The mockup shows a web interface for teachers. At the top is a blue header bar containing a circular 'logó' (logo) placeholder, followed by seven empty rectangular input fields, and a 'Login' button on the right. Below the header, the title 'Tanárok' is centered. Underneath is a light gray box containing sample data: 'Példa név: Példa Tíbor', 'Példa azonosító: 11 számjegyű azonosító', and 'Példa telefonszám: Telefonszám'. Below this box is a small blue button labeled 'gomb'. At the bottom of the page is a wide blue button labeled 'Elérhetőségek' (Contact Information).

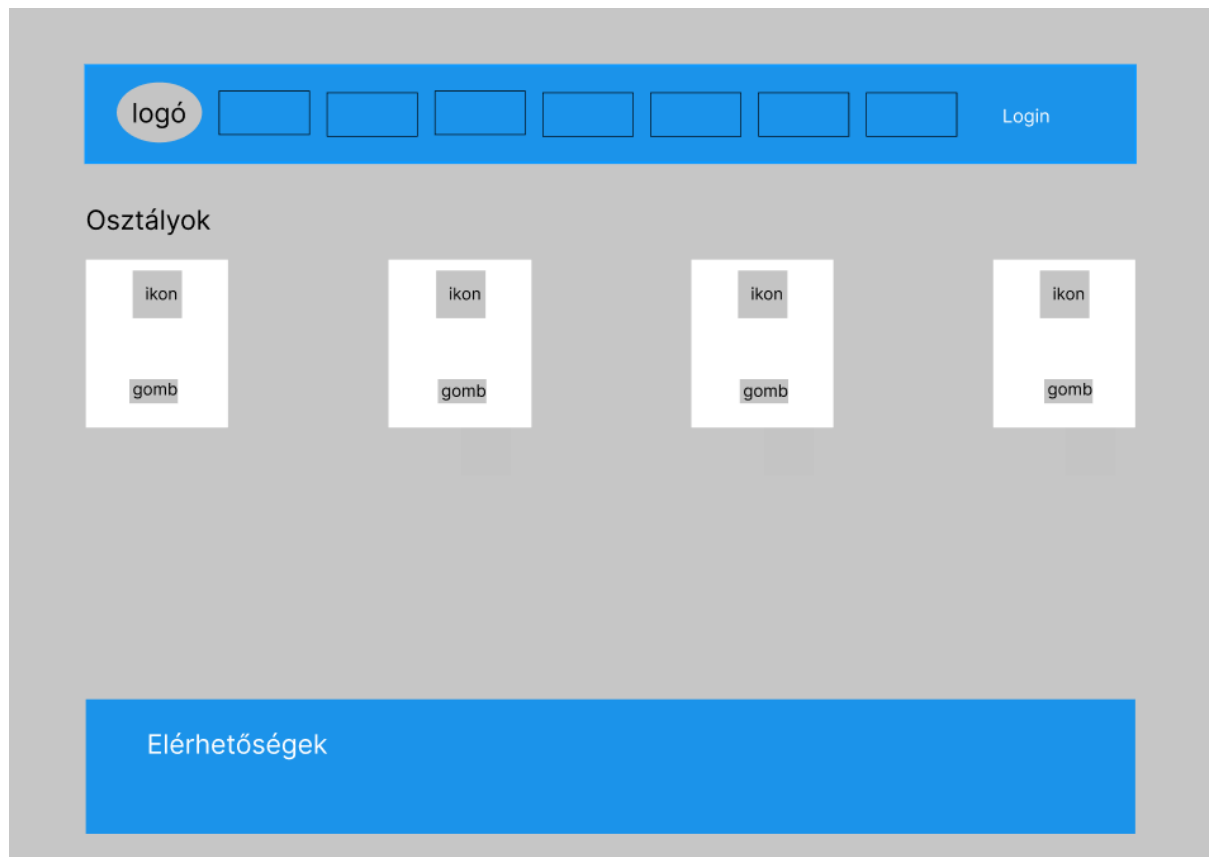
Osztályok oldal (Classes) leírása, képernyőkép(elképzelés):

Az osztályok oldalon az adatbázisban szereplő összes osztály megjelenik osztály szerint növekvően külön-külön dobozokban.

A dobozok dizájnjáért egy a témával kapcsolatos hangulatos ikon felel.

Az ikon alatt megjelenik az adat osztály neve.

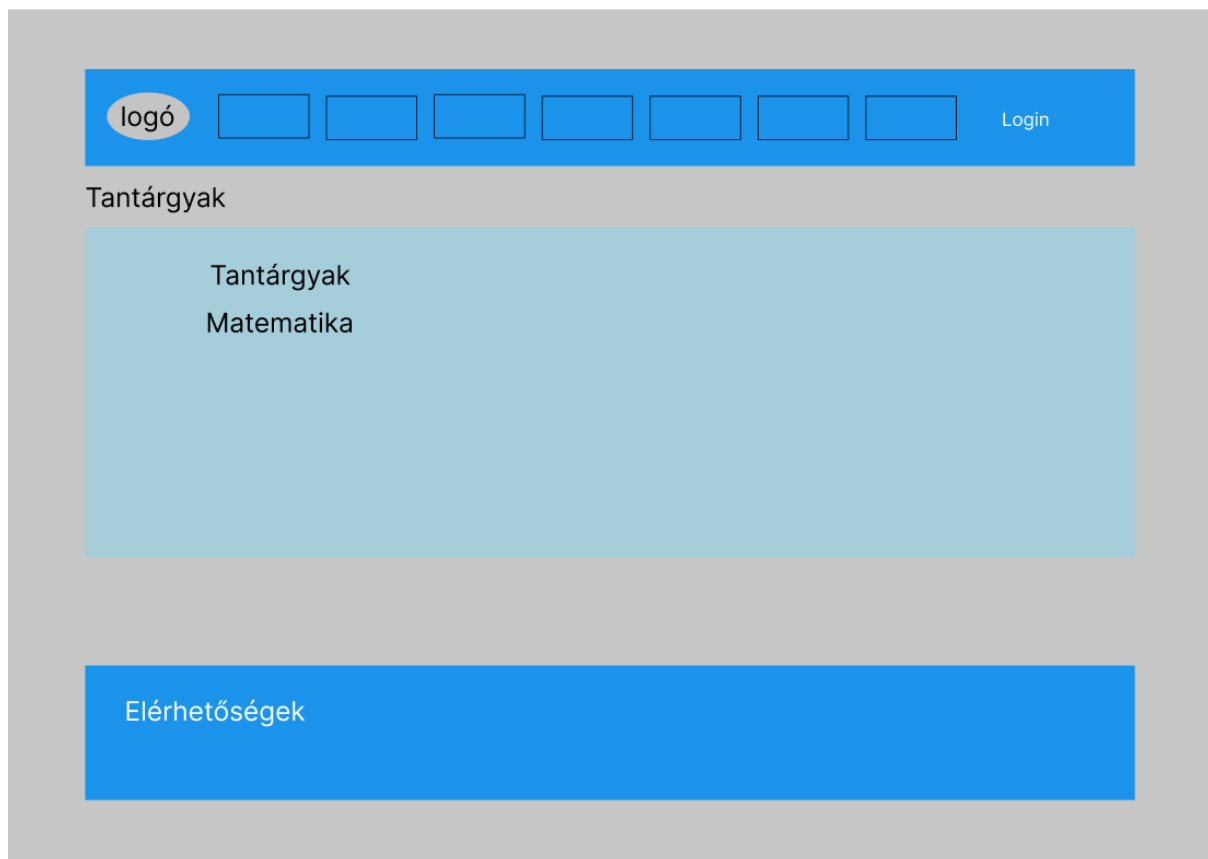
Az osztály neve alatt szerepel egy gomb, melyre kattintva az adott osztályban szereplő diákok névsora jelenik meg.



Tantárgyak oldal (Subjects) leírása, képernyőkép(elképzelés):

A tantárgyak oldalon az adatbázisban szereplő összes tantárgy meg fog jelenni ABC sorrend szerint.

A tantárgyak nevei egy táblázatban fognak megjelenni.



The mockup shows a web page layout with a light gray background. At the top is a blue header bar. On the left of the header is a white oval button with the text "logó". To its right are seven empty rectangular input fields. On the far right of the header is a white button with the text "Login". Below the header, the word "Tantárgyak" is displayed in a dark gray font. Underneath this is a large light blue rectangular area. Inside this area, the words "Tantárgyak" and "Matematika" are listed vertically in a dark gray font. At the bottom of the page is a solid blue rectangular bar containing the white text "Elérhetőségek".

## Órarendek oldal (TimeTable) leírása, képernyőkép(elképzelés):

Az órarendek oldalon az adatbázisban szereplő osztályok fognak megjelenni és a hozzá kapcsolódó órarendek.

Az egyes osztályok dobozokban fognak megjelenni:

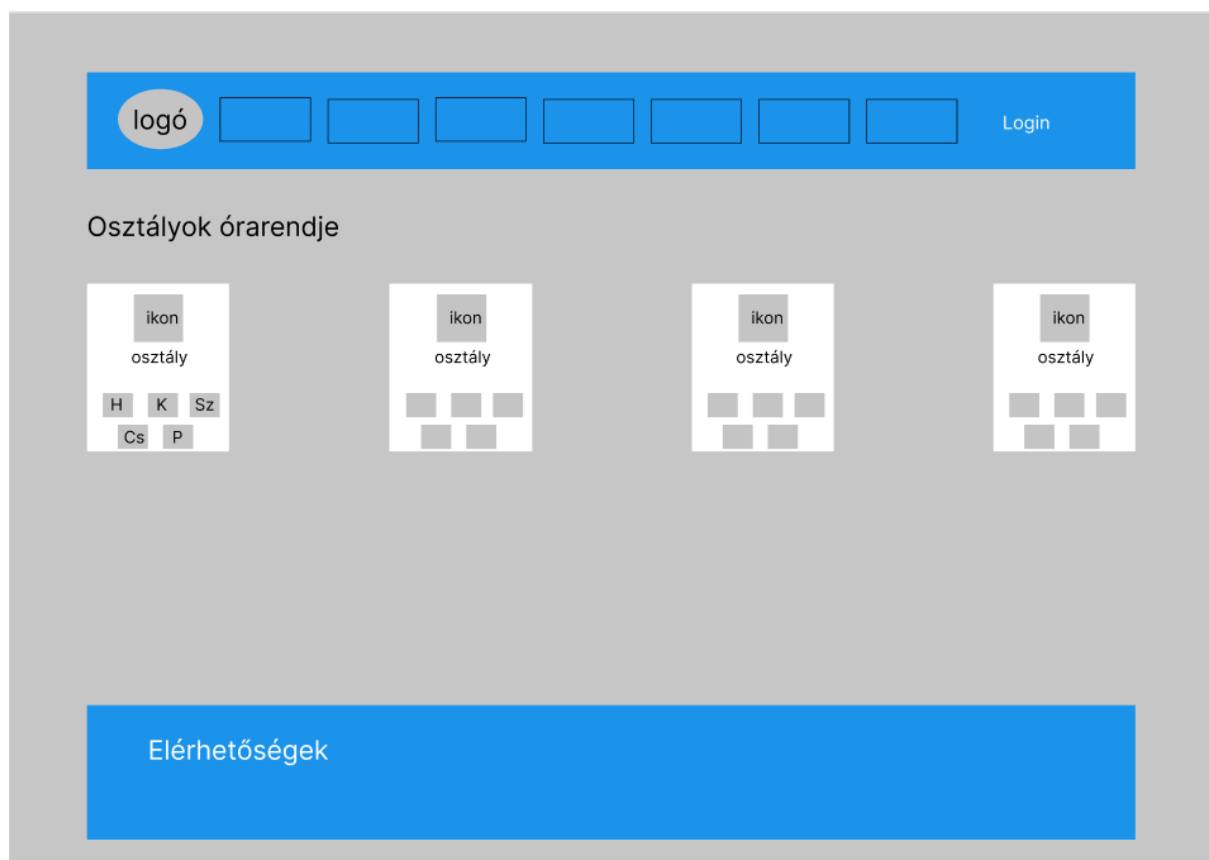
A dobozok tartalmazzák:

- Egy tematikus ikont
- Osztály nevét

Navigációs gombokat név szerint:

- M (monday-hétfő)
- T (tuesday-kedd)
- W (wednesday-szerda)
- T (thursday-csütörtök)
- F (friday-péntek)

A navigációs gombokra kattintva az adott naphoz, illetve osztályhoz kapcsolódó órarend jelenik meg táblázat formájában.





logó

Login

Nap

Óra

Ajtó

Tantárgy neve:

Tanár neve:

1.

101

Matematika

Példa Tibor

Elérhetőségek

Tanár tantárgyainak oldala (TeacherSubjects) leírása, képernyőkép(elképzelés):

Ez az oldal a tanárok oldalon megjelenő gomb megnyomásával jelenik majd meg.  
A kiválasztott tanárhoz tartozó tantárgyak jelennek meg táblázatban összefoglalva.

The mockup shows a web interface with a blue header bar. On the left of the header is a button labeled 'logó' inside a light gray circle. To its right are seven empty rectangular input boxes. On the far right of the header is a button labeled 'Login'. Below the header, the text 'Tanár tantárgya' is displayed. Underneath this is a light blue rectangular area containing two labels: 'Tanár:' and 'Tantárgy:'. Below 'Tanár:' is the text 'Példa Imre', and below 'Tantárgy:' is the text 'Matematika'. At the bottom of the page is a solid blue rectangular bar with the text 'Elérhetőségek' in white.

Kezdő oldal Adminisztrátor(index) leírása, képernyőkép(elképzelés):

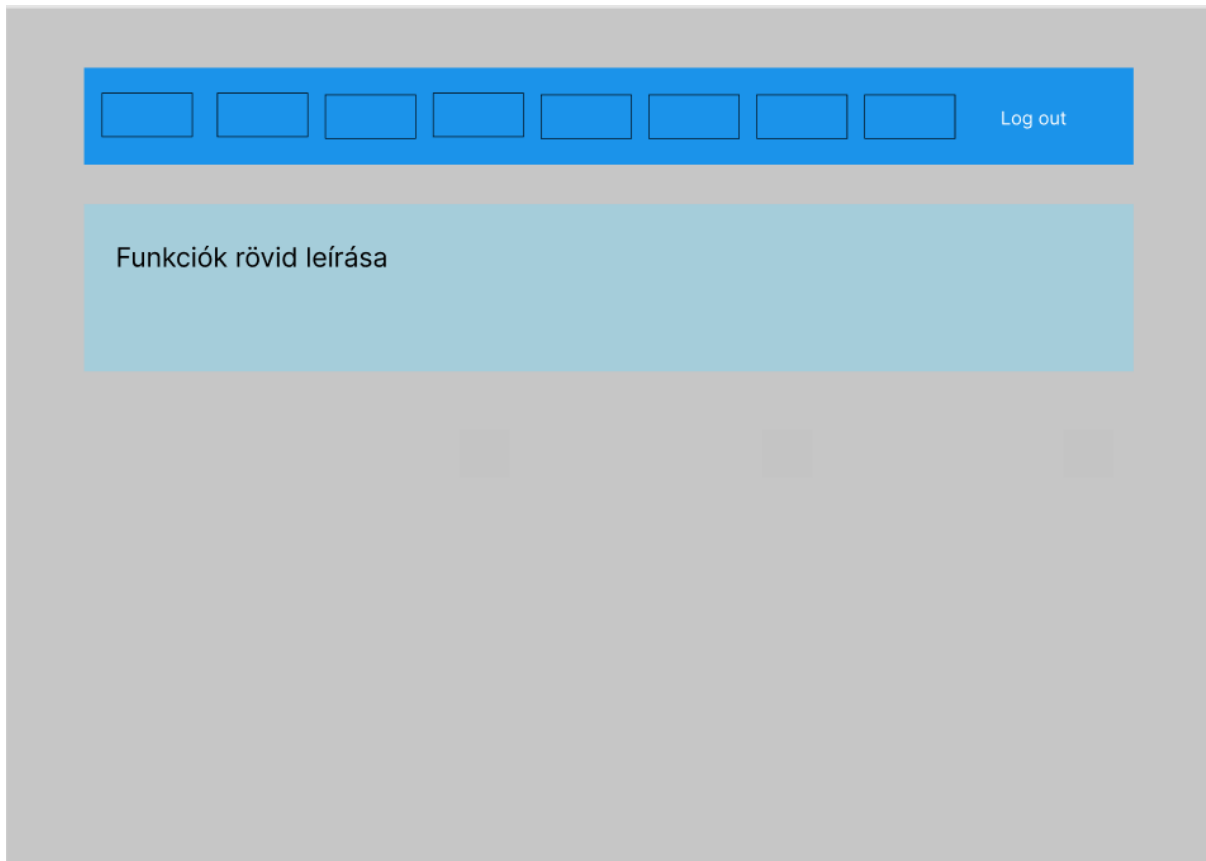
Ez az oldal az adminisztrációs joggal rendelkező felhasználók számára jelenik meg.

Az oldal szintén tartalmaz egy navigációs menüt, mely felelős az adminisztrációs oldalak közötti navigációért.

A gombok mellett megtalálható egy kijelentkező fül is.

A kezdőoldal az oldalak, illetve azok használatát foglalja össze dobozokban.

Minden egyes oldalhoz külön leírás, illetve doboz tartozik.



Tanár hozzáadása oldal(addTeacher) leírása, képernyőkép(elképzelés):

A tanár hozzáadása oldalon tanárokat tudunk felvenni az adatbázisba a következő adatok szerint:

- OM azonosító
- Név
- Telefonszám
- Jelszó
- Admin

Az oldalon megjelenik az összes tanár egy táblázatban és a hozzájuk tartozó összes adat.

A tanárok adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely a tanár törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos OM azonosító esetén az oldalunk duplikációs hibát dob.

Új tanár felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

The screenshot shows a web application interface for adding a new teacher. At the top, there is a blue navigation bar with buttons for 'Home', 'Add Teacher', 'Add Subject', 'Add Classroom', 'Add Student', 'Add Class', 'Add TimeTable', and 'Log out'. Below the navigation bar, the 'Add new Teacher' section contains a form with the following fields: 'OMA Teacher' (with a placeholder '11 számjegyű azonosító'), 'Name' (with a placeholder 'Példa Kálmán'), 'Phone' (with a placeholder 'Telszám'), 'Password' (with a placeholder 'jelszó'), and 'Admin' (with a placeholder 'true'). Below the form is a blue 'Add' button. At the bottom, there is a table with the following data:

OMA Teacher	Name	Phone	Password	Admin	Manage
11 számjegyű azonosító	Példa Kálmán	Telszám	jelszó	true	<a href="#">Swap Name</a> <a href="#">Delete</a>

Tantárgy hozzáadása oldal(addSubject) leírása, képernyőkép(elképzelés):

A tantárgy hozzáadása oldalon tantárgyakat tudunk felvenni az adatbázisba a következő adat szerint:

- Név (tantárgy neve)

Az oldalon megjelenik az összes tantárgy egy táblázatban és a hozzájuk tartozó összes adat.

A tantárgyak adatát meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely a tantárgy törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos ID esetén az oldalunk duplikációs hibát dob.

Új tantárgy felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

The screenshot shows a web application interface for adding a new subject. At the top, there is a blue navigation bar with buttons for 'Home', 'Add Teacher', 'Add Subject', 'Add Classroom', 'Add Student', 'Add Class', 'Add TimeTable', and 'Log out'. Below the navigation bar, the main content area has a light gray background. It starts with the text 'Add new Subject' followed by a label 'Name' and a text input field. Below the input field is a blue 'Add' button. Further down, there is a table with a light blue header and a light gray body. The table has three columns: 'Subject ID', 'Name', and 'Manage'. The first row of the table body contains the values 'Id', 'Példa Kálmán', and two buttons: 'Swap Space' (green) and 'Delete' (red).

Subject ID	Name	Manage
Id	Példa Kálmán	<button>Swap Space</button> <button>Delete</button>

Tanterem hozzáadása oldal(addClassRoom) leírása, képernyőkép(elképzelés):

A tanterem hozzáadása oldalon tantermeket tudunk felvenni az adatbázisba a következő adatok szerint:

- Ajtó
- Hely

Az oldalon megjelenik az összes tanterem egy táblázatban és a hozzájuk tartozó összes adat.

A tanterem adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely a tanterem törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos ID esetén az oldalunk duplikációs hibát dob.

Új tanterem felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

DoorNumber ID	Door	Space	Manage
Id	ajtó szám	Férőhely	<button>Swap Name</button> <button>Delete</button>

Diák hozzáadása oldal(addStudent) leírása, képernyőkép(elképzelés):

A diák hozzáadása oldalon diákokat tudunk felvenni az adatbázisba a következő adatok szerint:

- OMA azonosító
- Név

Az oldalon megjelenik az összes diák egy táblázatban és a hozzájuk tartozó összes adat.

A diákok adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely a diák törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos OM azonosító esetén az oldalunk duplikációs hibát dob.

Új diák felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

The screenshot shows a web application interface for adding a new student. At the top, there is a blue navigation bar with buttons for 'Home', 'Add Teacher', 'Add Subject', 'Add Classroom', 'Add Student', 'Add Class', 'Add TimeTable', and 'Log out'. Below the navigation bar, the main content area is titled 'Add new Student'. It contains two input fields: 'OMA Student' and 'Name'. Below these fields is a blue 'Add' button. At the bottom, there is a table displaying the list of students. The table has columns for 'OMA Student', 'Name', 'Class ID', and 'Manage'. The first row shows a student with the OMA ID '11 számjegyű azonosító', the name 'Példa Evelin', and the class ID 'Id'. The 'Manage' column for this row contains two buttons: 'Swap Name' (green) and 'Delete' (red).

OMA Student	Name	Class ID	Manage
11 számjegyű azonosító	Példa Evelin	Id	<button>Swap Name</button> <button>Delete</button>

Osztály hozzáadása (addClass) leírása, képernyőkép(elképzelés):

Az osztály hozzáadása oldalon osztályokat tudunk felvenni az adatbázisba a következő adat szerint:

- Név

Az oldalon megjelenik az összes osztály egy táblázatban és a hozzájuk tartozó összes adat.

Az osztályok adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely az osztályok törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos ID esetén az oldalunk duplikációs hibát dob.

Új osztály felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

The screenshot shows a web application interface for adding a new class. At the top, there is a blue navigation bar with buttons for 'Home', 'Add Teacher', 'Add Subject', 'Add Classroom', 'Add Student', 'Add Class', 'Add TimeTable', and 'Log out'. Below the navigation bar, the main content area is titled 'Add new Class'. It features a text input field labeled 'Name' and a blue 'Add' button. Below the input field, there is a table with the following structure:

Class ID	Name	Manage
Id	Osztály	<span>Swap Name</span> <span>Delete</span>



Órarend hozzáadása (addTimeTable) leírása, képernyőkép(elképzelés):

Az órarend hozzáadása oldalon órarendet tudunk felvenni az adatbázisba a következő adat szerint:

- Nap
- Óra
- Ajtószám
- Tantárgy
- Tanár neve
- Osztály

Az oldalon megjelenik az összes órarend egy táblázatban és a hozzájuk tartozó összes adat.

Az órarend adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely az órarend törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos ID esetén az oldalunk duplikációs hibát dob.

Új órarend felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

### Add new TimeTable

Day

Lesson

DoorNumber

Subject

Teacher Name

Class

Add

### Órák

Day	Lesson	DoorNumber	Subject	Teacher Name	Class	Manage
						<div>Swap Lesson</div> <div>Swap Door</div> <div>Swap Subject</div> <div>Swap Teacher</div> <div>Swap Class</div> <div>Delete</div>

Tanár órához rendelése oldal (addTeacherSubject) leírása, képernyőkép(elképzelés):  
A tanár tantárgya hozzáadása oldalon órarendet tudunk felvenni az adatbázisba a következő adat szerint:

- Név
- Tantárgy

Az oldalon megjelenik az összes tanár és tantárgya egy táblázatban és a hozzájuk tartozó összes adat.

A tanár tantárgya adatait meg lehet változtatni egy gombra kattintva.

Az adatok megváltoztatása gomb mellett található egy másik gomb is amely az tanár tantárgya törléséért felelős.

Az adatokat form-okba tudjuk beírni.

A form-ok közül az összes kitöltése kötelező.

Azonos ID esetén az oldalunk duplikációs hibát dob.

Új tanár tantárgya felvételéhez az Add gomb megnyomása szükséges.

Sikeres felvételnél, illetve törlésnél visszajelzést kapunk.

### Add Teacher Subject

Name

Subject

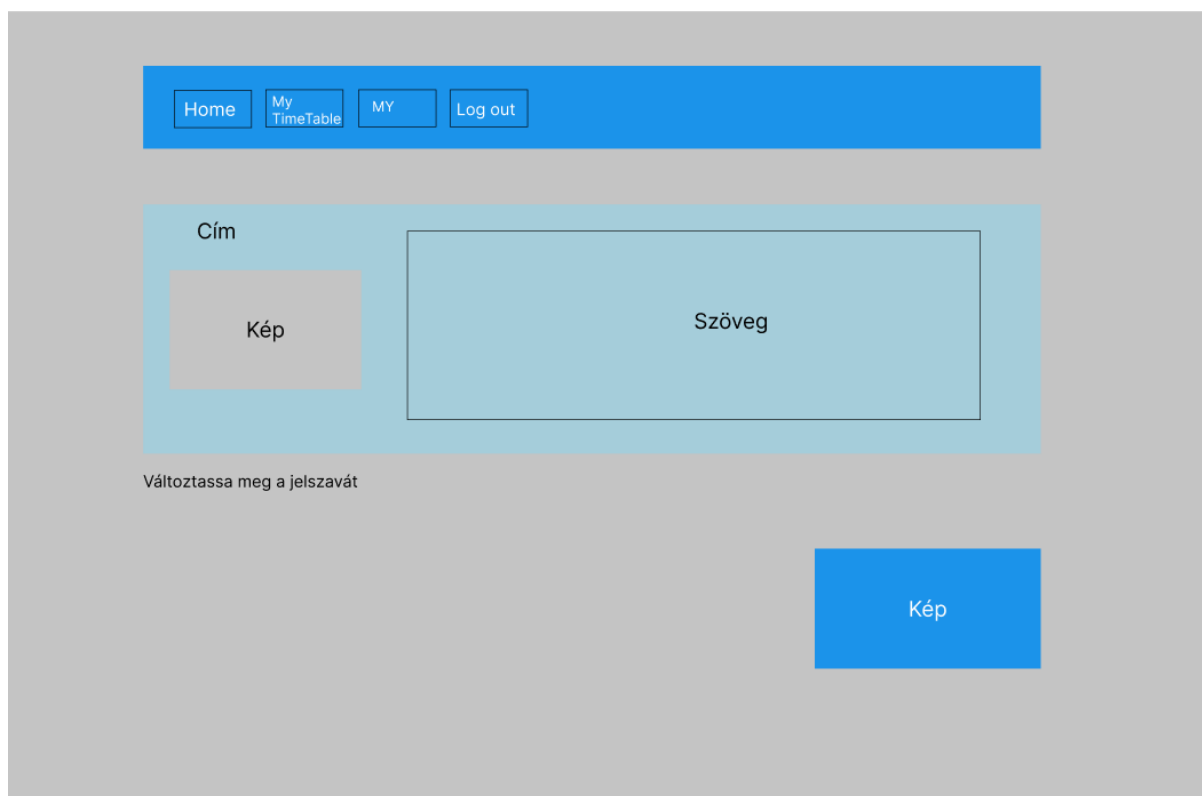
Add

Name	Subject	Manage
Példa Sándor	Tantárgy	Delete

Kezdőoldal Tanár(index) leírása, képernyőkép(elképzelés):

A kezdő oldalunkon megjelenik a navigációs menü.

Az oldalon a tanár oldallal kapcsolatos információk jelennek meg dobozokban, ami tartalmazza a leírásnak a címét és a leírást.



Tanár órarendje oldal (TeacherTimeTable) leírása, képernyőkép(elképzelés):

A tanár órarendje oldalon az adatbázisban szereplő tanár órarendje fog megjelenni.

A tanár tantárgyai egy táblázatban fognak megjelenni, amit egy legördülő menün keresztül lehet elérni naponként.

The mockup shows a web interface for a teacher's timetable. At the top is a blue login bar with seven input fields and a 'Login' button. Below this is a section labeled 'Nap' (Day) containing a table with the following data:

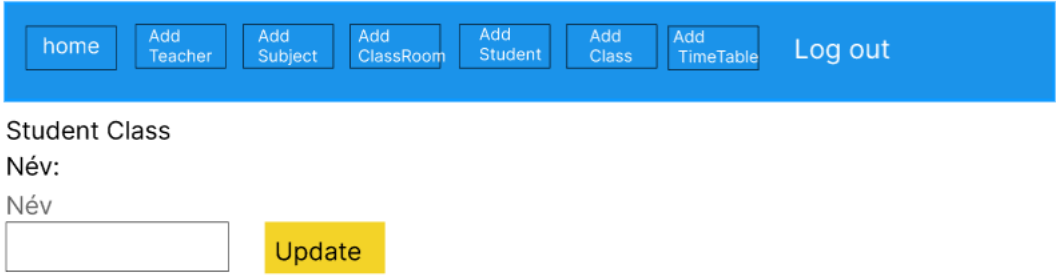
Óra	Ajtó	Tantárgy neve	Osztály
1	105	Testnevelés	10/b

Below the table is a blue bar with the text 'Elérhetőségek' (Contact information).

Tanár adatai oldal (My) leírása, képernyőkép(elképzelés):

Update: Az update oldalaknál azért nem készítettünk minden egyes index oldalról elképzelést mert mindegyik ugyan úgy fog kinézni csak az adott dologgal kapcsolatos adatokat lehet benne módosítani.

Osztály nevének módosítása:

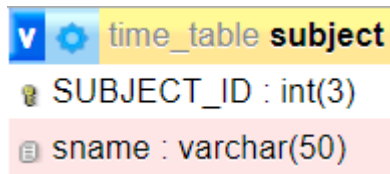


The screenshot shows a web application interface. At the top, there is a blue navigation bar with several buttons: 'home', 'Add Teacher', 'Add Subject', 'Add Classroom', 'Add Student', 'Add Class', 'Add TimeTable', and 'Log out'. Below the navigation bar, the page title 'Student Class' is displayed. Underneath, the label 'Név:' is followed by a text input field. To the right of the input field is a yellow button labeled 'Update'.

Ez az oldal az adminisztrátornak fog szintén csak megjelenni, az addclass oldalon a swap name gombra kattintva.

A név mezőben az eddig szereplő alap érték jelenik meg és azt tudjuk átírni majd az update gombra kattintva át is íródik az adatbázisunkban is.

## 5.Modellek:



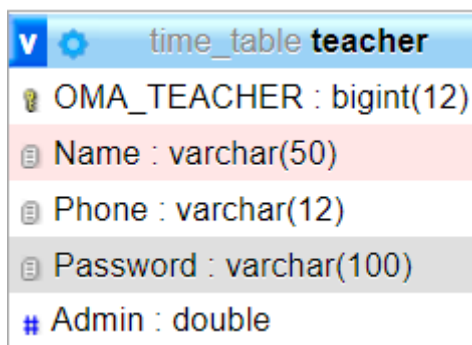
time_table subject	
🔑	SUBJECT_ID : int(3)
📄	sname : varchar(50)

A tantárgyakat a következő táblában tárolom el.  
A tantárgyhoz tartozik egy ID, ami nélkülözhetetlen.  
Az ID az elsődleges kulcs.  
Az ID AUTO INCREMENT, ami azt takarja, hogy automatikusan ad neki értéket az adatbázisunk.

Az ID-t integer típusú változóként tároljuk el.

Az ID értékének hosszúsága 3.

Még rendelkezik egy névvel is a tantárgy, amely varchar típusú, hosszúsága 50 karakter.



time_table teacher	
🔑	OMA_TEACHER : bigint(12)
📄	Name : varchar(50)
📄	Phone : varchar(12)
📄	Password : varchar(100)
#	Admin : double

A tanárokat a következő táblában tároltam el.  
Ebben a táblában a tanárokkal kapcsolatos adatok szerepelnek.

Első lépésként egy ID-t vettem fel, ami itt az OMA TEACHER ez az elsődleges kulcs.

Az Oktatási azonosító 11 karakterből áll, ezért ezt nem integerben, hanem bigintegerben tárolom el, mert ebben fér el.

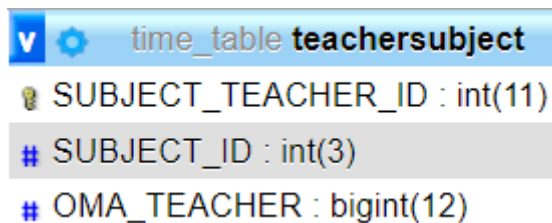
A tábla még a következő adatokat tárolja el:

-Tanár neve: amely varchar típusú és 50 karaktert képes tárolni.

-Telefonszám: szintén varchar típusú ez 12 karaktert képes tárolni. Azért a varcharat választottuk a telefonszámhoz, mert a telefonszámba feltehető módon nem csak számok, hanem karakterek is szerepelhetnek, mint például a "+".

-A jelszavát kezdetlegesen egy varchar típusú változóban tároljuk, mely 100 karakter hosszú is lehet.

-Az Admin az egy double típusú változó. Ennek értéke a bejelentkezésénél lesz fontos, mert egy tanárról ez dönti el, hogy a tanár rendelkezik-e adminisztrációs joggal.



time_table teachersubject	
🔑	SUBJECT_TEACHER_ID : int(11)
#	SUBJECT_ID : int(3)
#	OMA_TEACHER : bigint(12)

A tanárok tantárgyait a következő táblában tároltam el.

Ez egy kapcsolótábla, rendelkezik elsődleges kulccsal, illetve két idegen kulccsal.

time_table classroom
DOORNUMBER_ID : int(3)
door : varchar(10)
# space : int(3)

Az OMA\_TEACHER kivételével mindegyiket integer változóban tároltam el, kivéve a tanár OM azonosítóját (OMA\_TEACHER) mivel az biginteger. A termék adatait a következő táblában tároltam el. A tábla rendelkezik elsődleges kulccsal mely a DOORNUMBER\_ID ez integer típusú változó.

A táblában nem szerepel idegen kulcs.

A táblában eltároltam még az ajtó nevét(számát)

amely tárolására varchar típusú változót választottunk, mert a terem nevében szerepelhet szöveg, illetve egyéb karakterek.

A táblában a terem férőhelyének tárolása érdekében a space változót hoztam létre.

time_table student
OMA_STUDENT : bigint(12)
name : varchar(50)
# CLASS_ID : int(3)

A diákok adatait a következő táblában tároltam el.

A diák OM azonosítóját egy BIGINTEGER típusú változóban tároltam el mely 12 a másodikon karaktert képes tárolni. Erre azért volt szükségünk mert a diákok OM azonosítója egységesen 11

karakter hosszú pl.: 72126374812.

A diákok nevét varchar típusú változóban tároljuk.

A diákok OM azonosítója az elsődleges kulcs a táblákban. A táblában szerepel még idegen kulcs is, amely a CLASS\_ID -> mely az osztálynak az azonosítója.

time_table class
CLASS_ID : int(3)
cname : varchar(10)

Az osztályok adatait a következő táblában tároltam el. Az osztályok elsődleges kulcsa a CLASS\_ID. Integer típusú változó.

Rendelkezik még a tábla egy Cname változóval is, amely az osztály nevét tartalmazza.

time_table timetable
TIMETABLE_ID : int(3)
# SUBJECT_TEACHER_ID : int(3)
# CLASS_ID : int(3)
# DOORNUMBER_ID : int(3)
day : varchar(10)
# lesson : int(2)

Az órarendek tábla a következőképpen épül fel:

Az elsődleges kulcs természetesen a timetable ID-ja mely a TIMETABLE\_ID nevet kapta. Ez Integer típusú változó.

A táblában számos idegen kulcs szerepel:

-SUBJECT\_TEACHER\_ID: ez a tanár

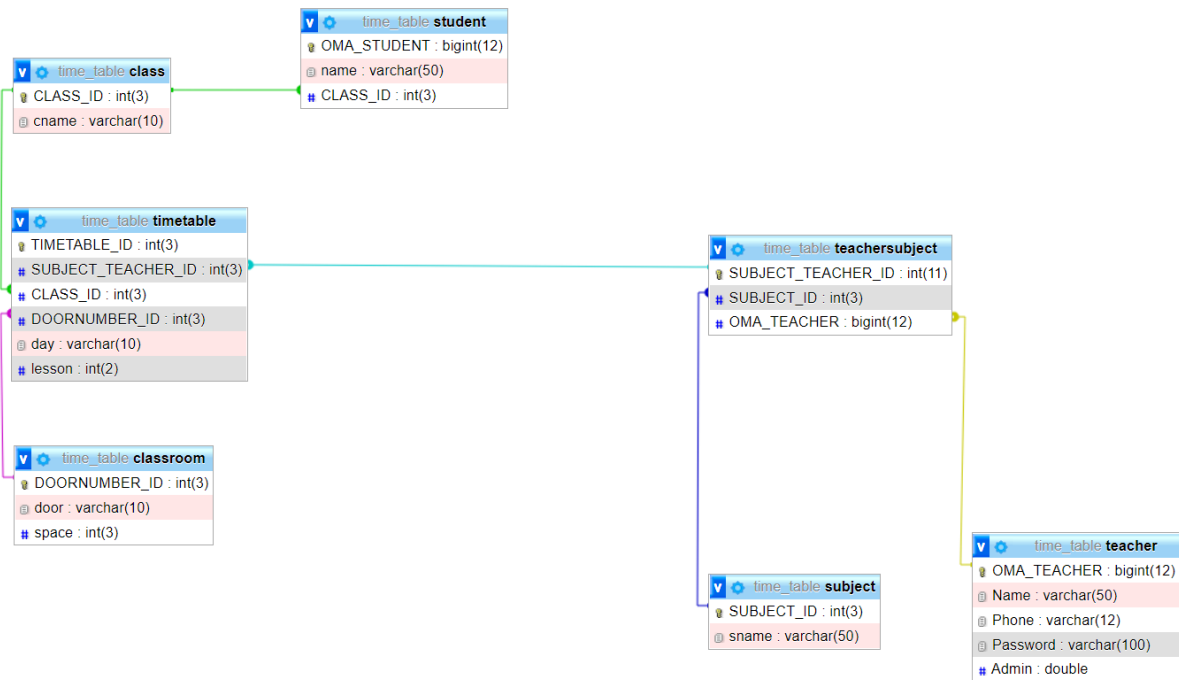
tantárgyainak tábla ID-ja.

-CLASS\_ID: ez az osztály tábla ID-ja.

-DOORNUMBER\_ID: ez a termék tábla ID-ja.

A táblában szerepel még a day (nap) melyet varchar típusú változóban tárolunk el, illetve egy sorszám is, amely azt jelzi, hogy hányadik óra.

Az adatbázisban kialakított kapcsolatok:



Adatbázisban szereplő táblák:

- Student
- Class
- ClassRoom
- TeacherSubject
- Subject
- Teacher
- TimeTable

Az adatbázisban a táblák közötti kapcsolat a következőképpen épül fel:

A **student** tábla idegen kulccsal kapcsolódik a **class** táblához. Az idegen kulcs a CLASS\_ID.

A köztük lévő kapcsolat(class-student) ManyToOne (sok az egyhez) kapcsolat.

A **subject** és a **teacher** tábla egy kapcsolótábla segítségével kapcsolódik össze(**teachersubject**).

A TeacherSubject két idegen kulccsal rendelkezik:

- SUBJECT\_ID (tantárgy ID-ja)
- OMA\_TEACHER (a tanár om azonosítója)

A táblák között ManyToMany(sok a sokhoz) kapcsolat van kialakítva.

A **TimeTable** táblához kapcsolódik a **ClassRoom** tábla, a **TeacherSubject** tábla, illetve a **Class** tábla.

A táblák idegenkulccsal kapcsolódnak egymáshoz.

A TimeTable-ban szereplő idegen kulcsok:



- SUBJECT\_TEACHER\_ID (tanár által tanítható tantárgy ID-ja)
- DOORNUMBER\_ID (a terem ID-ja)
- CLASS\_ID (az osztály ID-ja)

A kapcsolatok:

- Timetable-ClassRoom OneToOne (egy az egyhez) kapcsolat
- Timetable-TeacherSubject OneToOne (egy az egyhez kapcsolat)
- TimeTable-Class OneToOne (egy az egyhez kapcsolat)

## **6. Alkalmazások kiválasztása:**

### **Az adatok tárolásához MySQL-t választottunk.**

A MySQL szabadon terjeszthető az interneten keresztül, valamint nyílt forráskódú így bármelyik programozó alakíthatja a kódját. Magas biztonsági jogosultságokkal rendelkezik. Képes nagy mennyiségű adat kezelésére. A működéséhez nem szükséges nagy mennyiségű erőforrás, ez pedig alacsony költségekkel jár emiatt. A rétegek és szerkezetek modulokat foglalnak magukba, ami nagy stabilitást biztosít ezáltal.

### **Az alkalmazás kiválasztásán sokat gondolkodtunk, de végül a Spring mellett döntöttünk.**

#### **A Springről röviden:**

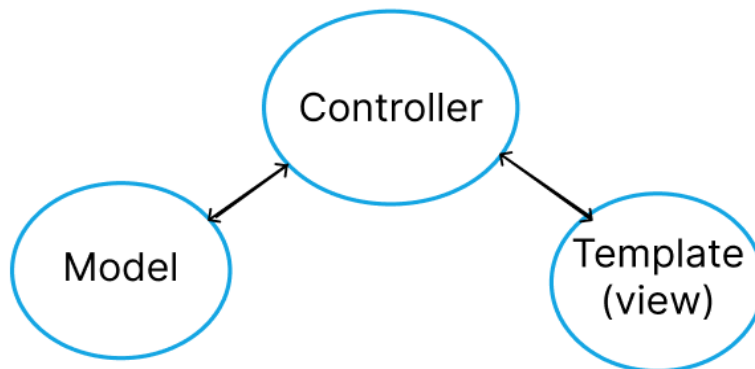
A legtöbb Spring Boot alkalmazás minimális Spring konfigurációt igényel. Könnyű hozzáférést biztosít a parancssori felülethez. A megkönnyíti az önálló Spring- alapú alkalmazások létrehozását. Könnyen hozzáférhető a beágyazott HTTP-kiszolgálóhoz. Legnagyobb előnye a rövidebb idő, minimálisra csökkenti a termelékenység fejlesztésére és növelésére fordított időt.

**Xamppról röviden:** Azért ezt választottuk, mert ezt tanultuk. Az alkalmazásunkat ezen tudjuk lokálisan futtatni. Apache webszervert használunk, amit itt futtatunk. Ezen kívül az adatbázisunk is innen fut (MySQL).

**IntelliJ-ről röviden:** Alkalmazáshoz azért az IntelliJ-t választottuk, mert ezt tanultuk és itt támogatott a Spring Boot és az SQL. Hatékonyan lehet benne dolgozni. Könnyű a használata.

## 7. Routing:

MVC Modell leírása és szemléltetése:



Model-> Adatbázis kezelés

View-> Megjelenítés (HTML)

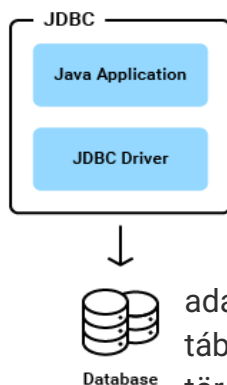
Control-> Vezérlés, illetve hidat képez a Modell és a View között.

**Mi az a MVC modell?** model-view-controller (model-nézet-vezérlő)

A modell: Itt tároljuk az adatokat és az üzleti logikát. Egy web applikációban ez a rész foglalkozik az adatbázisokkal és egyéb objektumokkal, amivel az applikáció dolgozik.

A nézet: A tényleges vizuális leképezése a modellnek. Webalkalmazásban ez a weblap, ami megjeleníti a modellt a felhasználó részére. A nézetnek nem kell egy modell minden részét mutatnia, és egy modellhez tartozhat több nézet is.

A vezérlő: Kezeli a modell és a nézet közötti kommunikációt. Webalkalmazásban a vezérlő metódusai vannak végrehajtva amikor a felhasználó egy weblapot tölt be vagy egy gombra kattint. A vezérlő ezekkel frissíti a modellt, ha szükséges, majd új nézetet ad vissza.



**Mi az a JDBC?** Java Database Connectivity.

Egy Java API, amelyet h a lekérdezés végrehajtására és összekapcsolására használnak az adatbázis mellett. Ez a Java Standard Edition része. A JDBC API a JDBC illesztőprogramjait használja fel annak érdekében, hogy kapcsolatba kerüljenek az adatbázissal. Felhasználható bármely relációs adatbázisban tárolt táblázatos adatok eléréséhez. Használatával megmentheti, frissítheti, törölheti és az adatbázisból letöltheti az adatokat.

## **A Routing-ról:**

A következő részek között a routing hasonlóan működik ezért nincsenek egyesével részletezve, hanem csoportosítva vannak:

### **ClassRoom, Class, Subject, Teacher**

**GET->Class-ok** kilistázása(/studentclasses): Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a kilistázó metódust tömbként, amely lekérdezi az adatbázisból az adatokat egy sql lekérdezéssel egy listát kapunk vissza ezt a listát a controller átadja a view-nek és ezt a kliens megjeleníti.

**GET->ClassRoom** kiíratása(/classrooms/DOORNUMBER\_ID) Az oldalra kattintva a backend-re controller vezérlése alapján meghívja a kiíró metódust tömbként, amely lekérdezi az adatbázisból az általunk kiválasztott adatot egy sql lekérdezéssel egy tömböt kapunk vissza ezt a tömböt a controller átadja a view-nek és ezt a kliens megjeleníti.

**DELETE->Subject** törlése(/subject/SUBJECT\_ID) Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a delete metódust, amely az adatbázisba egy lekérdezést küld, melyről egy status kódot kapunk vissza a műveletek eredményéről

**POST- >Teacher** hozzáadása(/teachers) add gomb megnyomásával a backend-re controller vezérlése alapján meghívja a post metódust, amely az adatbázisba json tömbként felküldi az adatokat és a kérést.

**PATCH->ClassRoom** módosítása(/ClassRoom/DOORNUMBER\_ID) Módosítás gomb megnyomásával a backend-re controller vezérlése alapján meghívja a patch metódust, mely az adatbázisba json tömbként felküldi a módosítani kívánt adatokat és a kérést a backend felé.

### **Student**

**GET-> Students**-ek kilistázása(/studentlists): Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a kilistázó metódust tömbként, amely lekérdezi az adatbázisból az adatokat egy sql lekérdezéssel egy listát kapunk vissza ezt a listát a controller átadja a view-nek és ezt a kliens megjeleníti.

**GET-> Student** kiíratása(/students/OMA\_STUDENT) Az oldalra kattintva a backend-re controller vezérlése alapján meghívja a kiíró metódust tömbként, amely lekérdezi az adatbázisból az általunk kiválasztott adatot egy sql lekérdezéssel egy tömböt kapunk vissza ezt a tömböt a controller átadja a view-nek és ezt a kliens megjeleníti.

**DELETE-> Student** törlése(/student/OMA\_STUDENT) Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a delete metódust, amely az adatbázisba egy lekérdezést küld, melyről egy status kódot kapunk vissza a műveletek eredményéről

**POST- > Student** hozzáadása(/students) add gomb megnyomásával a backend-re controller vezérlése alapján meghívja a post metódust, amely az adatbázisba json tömbként felküldi az adatokat és a kérést.

**PATCH**->Student módosítása(/student/OMA\_STUDENT) Módosítás gomb megnyomásával a backend-re controller vezérlése alapján meghívja a patch metódust, mely az adatbázisba json tömbként felküldi a módosítani kívánt adatokat és a kérést a backend felé.

### **TeacherSubject**

**GET**->TeacherSubjects-ek kilistázása(/teacherssubjects): Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a kilistázó metódust tömbként, amely lekérdezi az adatbázisból az adatokat egy sql lekérdezéssel egy listát kapunk vissza ezt a listát a controller átadja a view-nek és ezt a kliens megjeleníti.

**GET**-> TeacherSubjects kiírása(/teacherssubjects/OMA\_TEACHER) Az oldalra kattintva a backend-re controller vezérlése alapján meghívja a kiíró metódust tömbként, amely lekérdezi az adatbázisból az általunk kiválasztott adatot egy sql lekérdezéssel egy tömböt kapunk vissza ezt a tömböt a controller átadja a view-nek és ezt a kliens megjeleníti.

**DELETE**-> TeacherSubject törlése(/teacherssubjects/TEACHERSUBJECT\_ID) Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a delete metódust, amely az adatbázisba egy lekérdezést küld, melyről egy status kódot kapunk vissza a műveletek eredményéről

**POST**- > TeacherSubject hozzáadása(/teacherssubjects) add gomb megnyomásával a backend-re controller vezérlése alapján meghívja a post metódust, amely az adatbázisba json tömbként felküldi az adatokat és a kérést.

**PATCH**->TeacherSubject

módosítása(/teacherssubject/TEACHERSUBJECT\_ID) Módosítás gomb megnyomásával a backend-re controller vezérlése alapján meghívja a patch metódust, mely az adatbázisba json tömbként felküldi a módosítani kívánt adatokat és a kérést a backend felé.

### **TimeTable**

**GET**-> TimeTable kiírása(/TimeTables/day/CLASS\_ID) Az oldalra kattintva a backend-re controller vezérlése alapján meghívja a kiíró metódust tömbként, amely lekérdezi az adatbázisból az általunk kiválasztott adatot egy sql lekérdezéssel egy tömböt kapunk vissza ezt a tömböt a controller átadja a view-nek és ezt a kliens megjeleníti.

**DELETE**-> TimeTable törlése(/TimeTables/day/CLASS\_ID) Gomb megnyomásával a backend-re controller vezérlése alapján meghívja a delete metódust, amely az adatbázisba egy lekérdezést küld, melyről egy status kódot kapunk vissza a műveletek eredményéről

**POST**- > TimeTable hozzáadása(/timetables) add gomb megnyomásával a backend-re controller vezérlése alapján meghívja a post metódust, amely az adatbázisba json tömbként felküldi az adatokat és a kérést.

**PATCH**->TimeTable módosítása(/TimeTables/day/CLASS\_ID) Módosítás gomb megnyomásával a backend-re controller vezérlése alapján meghívja a patch metódust, amely az adatbázisba json tömbként felküldi a módosítani kívánt adatokat és a kérést a backend felé.

## 8.Middleware-ek:

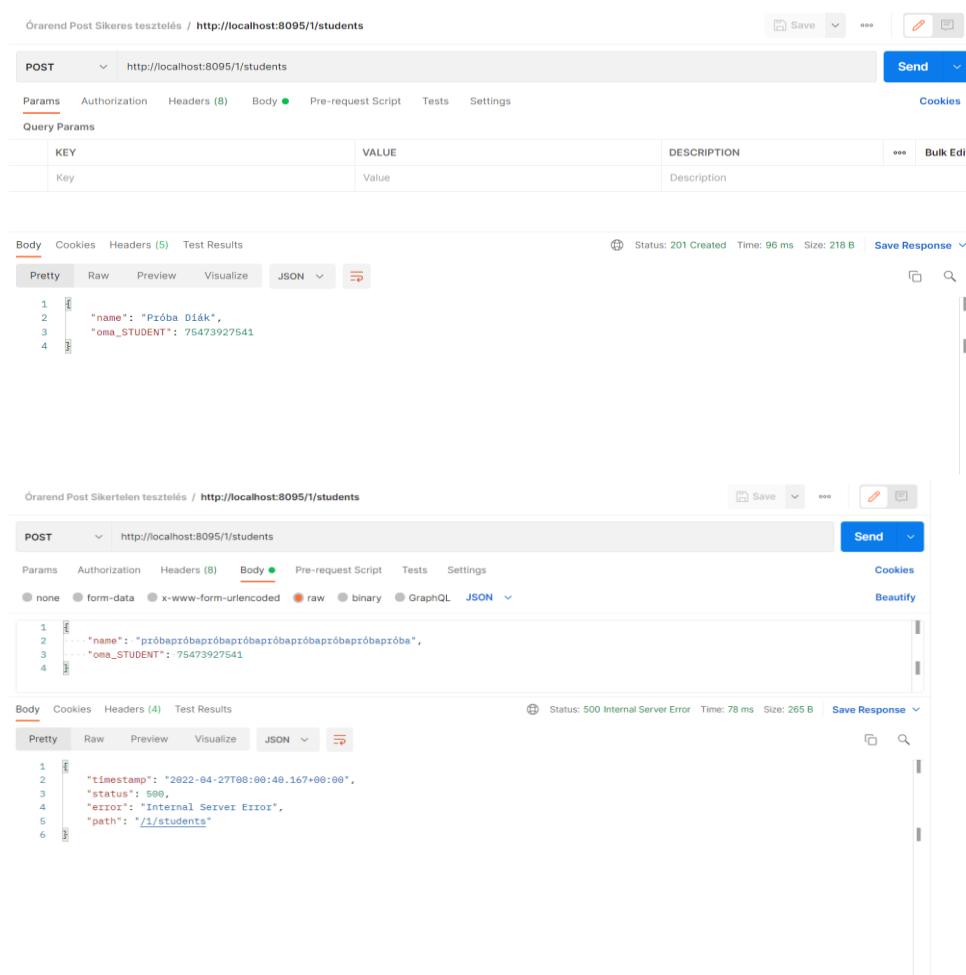
Dependencies-ek melyek könnyebbé tették a munkánkat a Backendben:

Az általunk használt Dependencies-ek:

- spring-boot-starter-data-jpa: Megkönnyíti és javítja a JPA-alapú adattárak egyszerű megvalósítását.
- spring-boot-starter-thymeleaf: Ezzel a motorral készített sablonok könnyen használhatók webdesign prototípusként. HTML, JavaScript CSS fájlokat dolgoz fel.
- spring-boot-starter-validation: Bean-(e)k validálására használjuk.
- spring-boot-starter-web: Automatikusan konfigurálható és webfejlesztéssel kompatibilis. Alapértelmezetten beállítja a webfejlesztéshez szükséges dolgokat.
- mysql-connector-java: Segít csatlakozni a MSQl-hez
- spring-boot-starter-test: Segít az alkalmazásunk tesztelésében
- org.jetbrains: A kód ellenőrzést támogatja annotations
- spring-boot-maven-plugin: Spring Boot támogatást nyújt Maven-ben

Dependencies-ek melyek könnyebbé tették a munkánkat a Frontendben:

- spring-boot-starter-parent: Alapértelmezett konfigurációkat biztosít az alkalmazásunkhoz
- spring-boot-starter-security: Nagymértékben testre szabható hitelesítés és hozzáférési keretrendszer
- spring-boot-starter-web: Automatikusan konfigurálható és webfejlesztéssel kompatibilis. Alapértelmezetten beállítja a webfejlesztéshez szükséges dolgokat.
- spring-boot-starter-thymeleaf: Ezzel a motorral készített sablonok könnyen használhatók webdesign prototípusként. HTML, JavaScript CSS fájlokat dolgoz fel.
- thymeleaf-extras-springsecurity5: Modern szerveroldali sablon webes környezethez
- httpclient: Apache komponens kliens
- spring-boot-starter-test: Segít az alkalmazásunk tesztelésében
- spring-security-test: Biztonság tesztelése
- spring-boot-maven-plugin: Spring Boot támogatást nyújt Maven-ben





- **Módosítás:**

## PATCH

A kérés JSON formátumban módosítja egy diák adatát OM azonosító szerint.

A diák adatai a validációnak megfelelnek.

A diák adatát sikeresen módosítani tudjuk.

A kérés sikeres akkor a módosított adatot rögzítettük az adatbázisunkba

A kérés JSON formátumban módosítja egy diák adatát OM azonosító szerint.

A diák adatai a validációnak nem felelnek meg.

Akkor a diák módosított adatát nem engedi felvenni.

A kérés sikertelen akkor a diák módosított adatát nem rögzíti az adatbázisunkba

Órend Patch Sikeres tesztelése / <http://localhost:8095/students/75473927541>

PATCH <http://localhost:8095/students/75473927541> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#) [Beautify](#)

```
1 {
2   "name": "Próba Próba",
3   "oma_STUDENT": 75473927541
4 }
```

Body Cookies Headers (5) Test Results [Status: 200 OK](#) [Time: 43 ms](#) [Size: 214 B](#) [Save Response](#)

Pretty Raw Preview Visualize [JSON](#) [Beautify](#)

```
1 {
2   "name": "Próba Próba",
3   "oma_STUDENT": 75473927541
4 }
```

Órend Patch Sikertelen Tesztelés / <http://localhost:8095/students/75473927541>

PATCH <http://localhost:8095/students/75473927541> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#) [Beautify](#)

```
1 {
2   "name": "",
3   "cname": "9",
4   "oma_STUDENT": 75473927541
5 }
```

Body Cookies Headers (4) Test Results [Status: 500 Internal Server Error](#) [Time: 37 ms](#) [Size: 275 B](#) [Save Response](#)

Pretty Raw Preview Visualize [JSON](#) [Beautify](#)

```
1 {
2   "timestamp": "2022-04-27T07:58:38.262+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "path": "/students/75473927541"
6 }
```

- **Törlés:**  
**DELETE**

A kérés törli a kiválasztott diákot.

A diákot sikeresen töröltük.

A kérés sikeres akkor a diák törlődik az adatbázisunkból

Órarend Delete tesztelés / <http://localhost:8095/students/75473927541>

DELETE <http://localhost:8095/students/75473927541> [Send](#)

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings [Cookies](#)

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

**Body** Cookies Headers (4) Test Results [Save Response](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#) [Text](#) [Copy](#) [Find](#)

1

- **Listázás:**

**GET:**

A kérés a diákok listája.

A diákok listáját sikeresen lekérjük.

A kérés sikeres akkor a diákok listája az adatbázisunkból megjelenik  
JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/students>

GET <http://localhost:8095/students> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings [Cookies](#)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

**Body** Cookies Headers (5) Test Results [Save Response](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#) [JSON](#) [Copy](#) [Find](#)

```
1 {
2   {
3     "name": "Anna Tóth",
4     "cname": "9",
5     "class_ID": 1,
6     "oma_STUDENT": 75346622123
7   },
8   {
9     "name": "Emma Szűcs",
10    "cname": "9",
11    "class_ID": 1,
12    "oma_STUDENT": 73492743982
13  },
14  {
15    "name": "Gréta Gulyás",
16    "cname": "9",
```

- Keresés:

GET:

A kérés a diákok listából egy OM azonosító által meghatározott diák.

A diák adatait sikeresen lekérjük.

A kérés sikeres akkor a diák listája az adatbázisunkból megjelenik

JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/students/75957745931>

GET [Save](#) [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 31 ms Size: 215 B [Save Response](#)

Pretty Raw Preview Visualize JSON [Send](#)

```

1 {
2   "name": "Máté Földes",
3   "oma_STUDENT": 75957745931
4 }
```

## -ClassRoom:

- Hozzáadás:

POST

A kérés JSON formátumban felküld egy tantermet.

A tanterem adatai a validációnak megfelelnek.

A tantermet sikeresen fel kell tudnunk venni.

A kérés sikeres volt a tanteremnél rögzítettük az adatbázisunkba.

A kérés JSON formátumban felküld egy tantermet.

A tanterem adatai a validációnak nem felelnek meg.

Akkor a tantermet nem engedi felvenni.

A kérés sikertelen akkor a tantermet nem rögzíti az adatbázisunkba.

Órarend Post Sikeres tesztelés / <http://localhost:8095/classrooms>

POST [Save](#) [Send](#)

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON [Beautify](#)

```

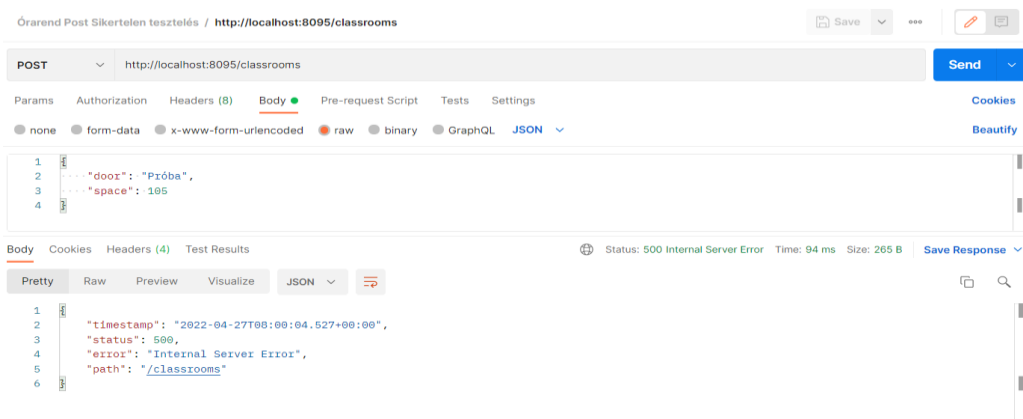
1 {
2   "dooz": "Próba",
3   "space": 12
4 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 32 ms Size: 216 B [Save Response](#)

Pretty Raw Preview Visualize JSON [Send](#)

```

1 {
2   "dooz": "Próba",
3   "space": 12,
4   "dooznumber_ID": 58
5 }
```



- Módosítás:

## PATCH

A kérés JSON formátumban módosítja egy tanterem adatát ID szerint.

A tanterem adatai a validációnak megfelelnek.

A tanterem adatát sikeresen módosítani tudjuk.

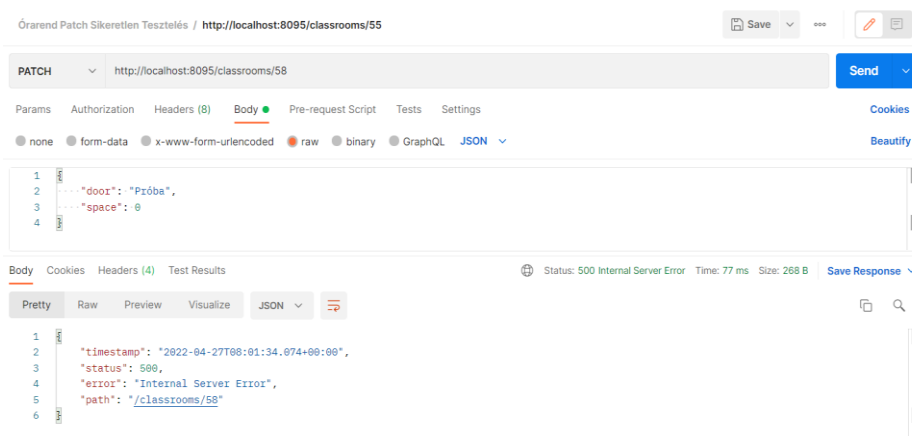
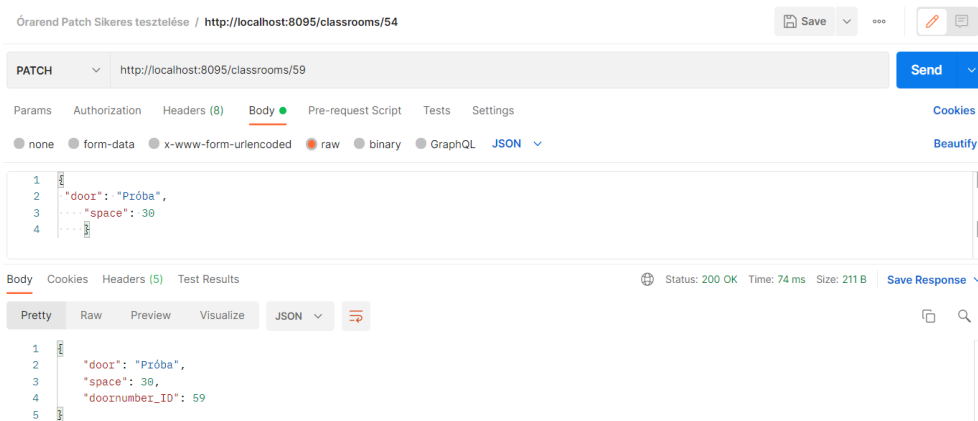
A kérés sikeres akkor a módosított adatot rögzítettük az adatbázisunkba

A kérés JSON formátumban módosítja egy tanterem adatát ID szerint.

A tanterem adatai a validációnak nem felelnek meg.

Akkor a tanterem módosított adatát nem engedi felvenni.

A kérés sikertelen akkor a tanterem módosított adatát nem rögzíti az adatbázisunkba.



- **Törlés:**  
DELETE: A kérés törli a kiválasztott tantermet ID szerint.  
A tantermet sikeresen töröltük.  
A kérés sikeres akkor a tanterem törlődik az adatbázisunkból

Órarend Delete tesztelés / <http://localhost:8095/classrooms/54>

DELETE <http://localhost:8095/classrooms/59> [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results [Save Response](#)

Pretty Raw Preview Visualize Text [Copy](#) [Find](#)

1

- **Listázás:**  
GET:  
A kérés a tantermek listája.  
A tantermek listáját sikeresen lekérjük.  
A kérés sikeres akkor a tantermek listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/classrooms>

GET <http://localhost:8095/classrooms> [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#) [Find](#)

```

1  {
2    {
3      "door": "103",
4      "space": 12,
5      "doornumber_ID": 3
6    },
7    {
8      "door": "104",
9      "space": 20,
10     "doornumber_ID": 4
11   },
12   {
13     "door": "105",
14     "space": 30,
15     "doornumber_ID": 5
16   },

```

[Cookies](#) [Capture requests](#) [Bootcamp](#) [Runner](#) [Trash](#)

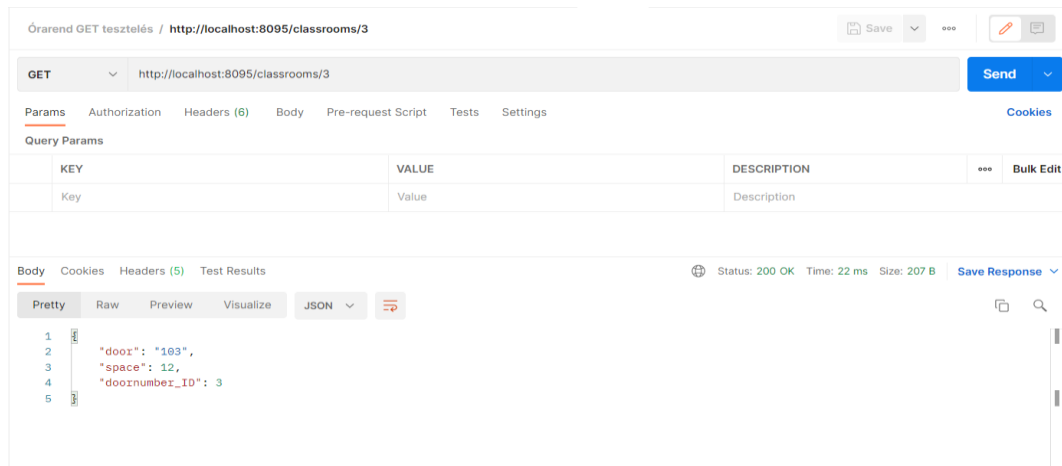
- Keresés:

GET:

A kérés a tanteremek listából ID által meghatározott tanterem.

A tanterem adatait sikeresen lekérjük.

A kérés sikeres akkor a tanterem listája az adatbázisunkból megjelenik JSON formátumban



## -Class:

- Hozzáadás:

POST

A kérés JSON formátumban felküld egy osztályt.

Az osztály adatai a validációnak megfelelnek.

Az osztályt sikeresen fel kell tudnunk venni.

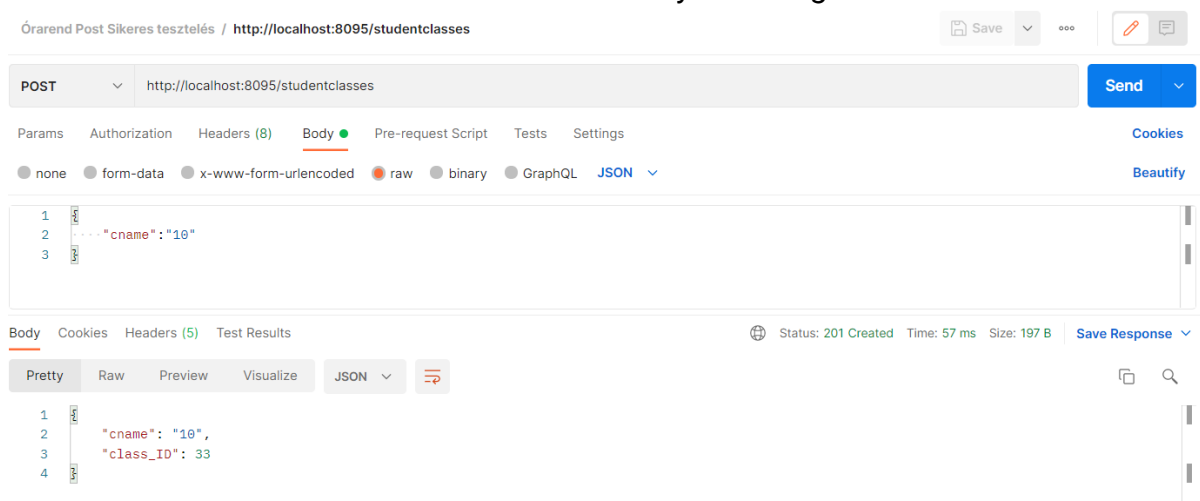
A kérés sikeres volt az osztálynál rögzítettük az adatbázisunkba.

A kérés JSON formátumban felküld egy osztályt.

Az osztály adatai a validációnak nem felelnek meg.

Akkor az osztályt nem engedi felvenni.

A kérés sikertelen akkor az osztályt nem rögzíti az adatbázisunkba.



Órarend Post Sikertelen tesztelés / <http://localhost:8095/studentclasses>

POST <http://localhost:8095/studentclasses> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 { "cname": "11" }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 26 ms Size: 269 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2022-04-27T08:02:43.504+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "path": "/studentclasses"
6 }
```

- **Módosítás:**

## PATCH

A kérés JSON formátumban módosítja egy osztály adatát ID szerint.

Az osztály adatai a validációnak megfelelnek.

Az osztály adatát sikeresen módosítani tudjuk.

A kérés sikeres akkor a módosított adatot rögzítettük az adatbázisunkba

A kérés JSON formátumban módosítja egy osztály adatát ID szerint.

Az osztály adatai a validációnak nem felelnek meg.

Akkor az osztály módosított adatát nem engedi felvenni.

A kérés sikertelen akkor az osztály módosított adatát nem rögzíti az adatbázisunkba.

Órarend Patch Sikeres tesztelése / <http://localhost:8095/studentclasses/31>

PATCH <http://localhost:8095/studentclasses/31> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "cname": "11"
3 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 26 ms Size: 192 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "cname": "11",
3   "class_ID": 33
4 }
```

Órarend Post Sikertelen tesztelés / <http://localhost:8095/studentclasses>

POST <http://localhost:8095/studentclasses> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 `{ "cname": " " }`

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 26 ms Size: 269 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2022-04-27T08:02:43.504+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "path": "/studentclasses"
6 }
```

- **Törlés:**  
DELETE: A kérés törli a kiválasztott osztályt ID szerint.  
Az osztályt sikeresen töröltük.  
A kérés sikeres akkor az osztály törlődik az adatbázisunkból

Órarend Delete tesztelés / <http://localhost:8095/studentclasses/31>

DELETE <http://localhost:8095/studentclasses/31> Send

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 35 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

1



- Listázás:

GET:

A kérés az osztályok listája.

Az osztályok listáját sikeresen lekérjük.

A kérés sikeres akkor az osztályok listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/studentclasses>

GET [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 41 ms Size: 332 B Save Response

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```
1 {
2   {
3     "cname": "9",
4     "class_ID": 1
5   },
6   {
7     "cname": "10",
8     "class_ID": 2
9   },
10  {
11    "cname": "11",
12    "class_ID": 3
13  },
14  {
15    "cname": "12",
16    "class_ID": 9
17  }
18 }
```

[Cookies](#) [Capture requests](#) [Postman](#) [Docker](#) [Trash](#)

- Keresés:

GET:

A kérés az osztályok listából ID által meghatározott osztály.

Az osztály adatait sikeresen lekérjük.

A kérés sikeres akkor az osztály listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/studentclasses/3>

GET [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 54 ms Size: 191 B Save Response

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```
1 {
2   "cname": "11",
3   "class_ID": 3
4 }
```

## -Teacher:

- Hozzáadás:

A kérés JSON formátumban felküldi a tanárt.

A tanár adatai a validációnak megfelelnek.

A tanárt sikeresen fel kell tudnunk venni.

A kérés sikeres akkor a tanár adatait rögzítettük az adatbázisunkba.

A kérés JSON formátumban felküldi a tanárt.

A tanár adatai a validációnak nem felelnek meg.

Akkor a tanárt nem engedi felvenni.

A kérés sikertelen akkor a tanárt nem rögzíti az adatbázisunkba.

The image displays two screenshots of a REST client interface, likely Postman, showing the results of a POST request to the endpoint `http://localhost:8095/teachers`.

**Top Screenshot: Successful Request**

- Method:** POST
- URL:** `http://localhost:8095/teachers`
- Body (JSON):**

```
{  "name": "Próba Tanár",  "password": "próbajelszó",  "phone": "0620111111",  "oma_TEACHER": 12345678453,  "admin": false}
```
- Status:** 201 Created
- Time:** 252 ms
- Size:** 282 B

**Bottom Screenshot: Failed Request**

- Method:** POST
- URL:** `http://localhost:8095/teachers`
- Body (JSON):**

```
{  "name": "Próba Tanár",  "password": "próbajelszó",  "phone": "0620312312",  "oma_TEACHER": ,  "admin": false}
```
- Status:** 400 Bad Request
- Time:** 35 ms
- Size:** 243 B
- Response (JSON):**

```
{  "timestamp": "2022-04-27T07:54:57.529+00:00",  "status": 400,  "error": "Bad Request",  "path": "/teachers"}
```

- Módosítás:

## PATCH

A kérés JSON formátumban módosítja egy tanár adatát OM azonosító szerint.

A tanár adatai a validációnak megfelelnek.

A tanár adatát sikeresen módosítani tudjuk.

A kérés sikeres akkor a módosított adatot rögzítettük az adatbázisunkba

A kérés JSON formátumban módosítja egy tanár adatát OM azonosító szerint.

A tanár adatai a validációnak nem felelnek meg.

Akkor a tanár módosított adatát nem engedi felvenni.

A kérés sikertelen akkor a tanár módosított adatát nem rögzíti az adatbázisunkba.

Órend Patch Sikeres tesztelése / <http://localhost:8095/teachers/12345678453>

PATCH

<http://localhost:8095/teachers/12345678453>

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Próba Tanár",
3   "password": "próba_jelszó",
4   "phone": "06206543215",
5   "oma_TEACHER": 12345678453,
6   "admin": false
7 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 40 ms Size: 277 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "name": "Próba Tanár",
3   "admin": false,
4   "oma_TEACHER": 12345678453,
5   "password": "próba_jelszó",
6   "phone": "06206543215"
7 }
```

Órend Patch Sikertelen Tesztelés / <http://localhost:8095/teachers/87546346321>

PATCH

<http://localhost:8095/teachers/12345678453>

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Próba Tanár",
3   "password": "prób",
4   "phone": "0620312312",
5   "oma_TEACHER": 12345678453,
6   "admin": false
7 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 56 ms Size: 275 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2022-04-27T07:57:27.787+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "path": "/teachers/12345678453"
6 }
```

- **Törlés:**  
DELETE: A kérés törli a kiválasztott tanár OM azonosítója szerint.  
A tanárt sikeresen töröltük.  
A kérés sikeres akkor a tanár törlődik az adatbázisunkból

Órarend Delete tesztelés / <http://localhost:8095/teachers/12345678453>

DELETE [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (4) Test Results Status: 200 OK Time: 60 ms Size: 123 B [Save Response](#)

Pretty Raw Preview Visualize Text [Copy](#) [Search](#)

1

- **Listázás:**  
GET:  
A kérés a tanárok listája.  
A tanárok listáját sikeresen lekérjük.  
A kérés sikeres akkor a tanárok listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/teachers>

GET [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 106 ms Size: 814 B [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```

1  {
2    {
3      "name": "Ákos Szabó",
4      "admin": false,
5      "oma_TEACHER": 71123567212,
6      "password": "adsfas",
7      "phone": "0620312312"
8    },
9    {
10     "name": "András Bakti",
11     "admin": false,
12     "oma_TEACHER": 71123567112,
13     "password": "jelszó",
14     "phone": "06203420711"
15   },
16   {

```

- Keresés:

GET:

A kérés a tanárok listából OM azonosító által meghatározott tanár.

A tanár adatait sikeresen lekérjük.

A kérés sikeres akkor a tanár listája az adatbázisunkból megjelenik

JSON formátumban

Örarend GET tesztelés / <http://localhost:8095/teachers/71123567212> Save ... ✎ 💬

**GET** ▼ <http://localhost:8095/teachers/71123567212> Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 33 ms Size: 268 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```
1 {
2   "name": "Ákos Szabó",
3   "admin": false,
4   "oma_TEACHER": 71123567212,
5   "password": "adsfas",
6   "phone": "0620312312"
7 }
```

## -Subject:

- Hozzáadás:

### POST

A kérés JSON formátumban felküld egy tantárgyat.

A tantárgy adatai a validációnak megfelelnek.

A tantárgy sikeresen fel kell tudnunk venni.

A kérés sikeres volt a tantárgyat rögzítettük az adatbázisunkba.

A kérés JSON formátumban felküld egy tantárgyat.

A tantárgy adatai a validációnak nem felelnek meg.

Akkor a tantárgyat nem engedni felvenni.

A kérés sikertelen akkor a tantárgyat nem rögzíti az adatbázisunkba.

The image displays two screenshots of a REST client interface, likely Postman, showing the results of a POST request to the endpoint `http://localhost:8095/subjects`.

**Top Screenshot: Successful Request**

- Header:** `POST http://localhost:8095/subjects`
- Body:** A JSON object: `{ "sname": "Próba" }`
- Status:** 201 Created
- Time:** 68 ms
- Size:** 203 B

**Bottom Screenshot: Failed Request**

- Header:** `POST http://localhost:8095/subjects`
- Body:** A JSON object: `{ "sname": "" }`
- Status:** 500 Internal Server Error
- Time:** 27 ms
- Size:** 263 B
- Response Body:** A JSON object: `{ "timestamp": "2022-04-27T08:04:22.705+00:00", "status": 500, "error": "Internal Server Error", "path": "/subjects" }`

- **Módosítás:**

## PATCH

A kérés JSON formátumban módosítja egy tantárgy adatát ID szerint.

A tantárgy adatai a validációnak megfelelnek.

A tantárgy adatát sikeresen módosítani tudjuk.

A kérés sikeres akkor a módosított adatot rögzítettük az

adatbázisunkba

A kérés JSON formátumban módosítja egy tantárgy adatát ID szerint.

A tantárgy adatai a validációnak nem felelnek meg.

Akkor a tantárgy módosított adatát nem engedi felvenni.

A kérés sikertelen akkor a tantárgy módosított adatát nem rögzíti az

adatbázisunkba.

Órarend Patch Sikeres tesztelése / http://localhost:8095/subjects/26

PATCH

http://localhost:8095/subjects/28

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

- **Törlés:**  
DELETE: A kérés törli a kiválasztott tantárgyat ID szerint.  
A tantárgyat sikeresen töröltük.  
A kérés sikeres akkor a tantárgy törlődik az adatbázisunkból

Órarend Delete tesztelés / <http://localhost:8095/subjects/26> Save ... ✎ 💬

**DELETE** ▼ <http://localhost:8095/subjects/28> Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (4) Test Results 🌐 Status: 200 OK Time: 31 ms Size: 123 B Save Response ▼

Pretty Raw Preview Visualize Text ▼ 🔍

1

- **Listázás:**  
GET:  
A kérés a tantárgyak listája.  
A tantárgyak listáját sikeresen lekérjük.  
A kérés sikeres akkor a tantárgyak listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/subjects> Save ... ✎ 💬

**GET** ▼ <http://localhost:8095/subjects> Send ▼

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body ▼ 🌐 200 OK 832 ms 457 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```

1  {
2    {
3      "sname": "Matek",
4      "subject_ID": 21
5    },
6    {
7      "sname": "SZAa",
8      "subject_ID": 8
9    },
10   {
11     "sname": "BP",
12     "subject_ID": 7
13   },
14   {
15     "sname": "PAAF",
16     "subject_ID": 6

```



- Keresés:

GET:

A kérés a tantárgyak listából ID által meghatározott tantárgy.

A tantárgy adatait sikeresen lekérjük.

A kérés sikeres akkor a tantárgy listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/subjects/1>

GET <http://localhost:8095/subjects/1> [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings [Cookies](#)

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```
1 {
2   "sname": "DATABASE",
3   "subject_ID": 1
4 }
```

## -TimeTable:

- Listázás:

GET:

A kérés az órarendek listája Nap szerint.

Az órarendek listáját sikeresen lekérjük.

A kérés sikeres akkor az órarendek listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/timetablelists/2/Monday>

GET <http://localhost:8095/timetablelists/2/Monday> [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings [Cookies](#)

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```
1 [
2   {
3     "name": "Gábor Huzi",
4     "sname": "FPET",
5     "cname": "10",
6     "doot": "103",
7     "day": "Monday",
8     "lesson": 2,
9     "oma_TEACHER": 0,
10    "class_ID": 0,
11    "timetable_ID": 13
12  },
13  {
14    "name": "Gábor Huzi",
15    "sname": "FPET",
16    "cname": "10",
17    ...
18  }
19 ]
```

- Keresés:

GET:

A kérés az órarendek listából ID által meghatározott órarend.

Az órarend adatait sikeresen lekérjük.

A kérés sikeres akkor az órarend listája az adatbázisunkból megjelenik

JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/timetables/10> Save ... ✎ 💬

GET ▼ <http://localhost:8095/timetables/10> Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 106 ms Size: 394 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```
1 {
2   "day": "Wednesday",
3   "lesson": 1,
4   "classRoom": {
5     "door": "106",
6     "space": 30,
7     "doornumber_ID": 6
8   },
9   "studentClass": {
10    "cname": "14",
11    "class_ID": 6
12  },
13  "teacherSubject": {
14    "oma_TEACHER": 71123567112,
15    "subject_ID": 2,
16    "subject_TEACHER_ID": 2
17  }
```

## -TeacherSubject:

- Listázás:

GET:

A kérés a tanár tantárgyainak listája.

A tanár tantárgyainak listáját sikeresen lekérjük.

A kérés sikeres akkor a tanár tantárgyainak listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / <http://localhost:8095/teachersubjectlist/71123567012>

GET [Send](#)

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results [Save Response](#)

Pretty Raw Preview Visualize JSON [Copy](#) [Search](#)

```
1  {
2    {
3      "sname": "DATABASE",
4      "name": "Gábor Huri"
5    },
6    {
7      "sname": "JAVA",
8      "name": "Gábor Huri"
9    },
10   {
11     "sname": "FPET",
12     "name": "Gábor Huri"
13   }
14 }
```

- Keresés:

GET:

A kérés a tanárok tantárgyainak listából OM azonosító által meghatározott tanár tantárgya.

A tanár tantárgya adatait sikeresen lekérjük.

A kérés sikeres akkor a tanár tantárgyai listája az adatbázisunkból megjelenik JSON formátumban

Órarend GET tesztelés / http://localhost:8095/teachersubjectlist/71123567212

GET http://localhost:8095/teachersubjectlist/71123567212 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 34 ms Size: 280 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "sname": "FPET",
4     "name": "Ákos Szabó"
5   },
6   {
7     "sname": "JAVA",
8     "name": "Ákos Szabó"
9   },
10  {
11    "sname": "BP",
12    "name": "Ákos Szabó"
13  }
14 }
```

**Frontend és egyéb tesztelésre** nem volt lehetőségünk.

## 11.Felhasználói dokumentáció

Az adatbázis futtatására xamppot használni. Az Apache és MYSQL futtatása után.

Az alkalmazás indításának lépései:

Első lépés a adatbázis futtatása xampp használatával.

A xampp controllerén elindítjuk a Apache és a MYSQL szerverünket.

Második lépésként a Backend appot indítjuk el a IntelliJ-n.

Harmadik lépés a Frontend futtatása szinten IntelliJ használatával.

A két app külön-külön porton fut.

Az oldal <http://localhost:9095/> linken érhető el ezt beírva rögtön a főoldal fogad minket, amihez tartozik egy navigációs menü a saját logónkkal a bal oldalon. A navigációs menü alatt található leírás az alkalmazásról. A leírás alatt pedig egy záró footer található az elérhetőségekkel.

A navigációs menüben a Student-re kattintva elvisz minket a Tanulók névsora oldalra, ami egy táblázatban jelenik meg. Ez a táblázat tartalmazza a diákok OM azonosítóját, illetve nevét. Itt az összes diák neve megtalálható.

A navigációs menüben a Room-ra kattintva elvisz minket a Termek oldalra. Ezen az oldalon az adatbázisban szereplő termek jelennek meg táblázatban.

A navigációs menüben a Tanárok-ra kattintva elvisz minket a Tanárok oldalra. Ezen az oldalon dobozokba külön-külön jelennek meg a tanárok a saját adataikkal, ami a tanár nevét, OM azonosítóját és telefonszámát tartalmazza. Az információk alatt egy gomb található, amire, ha kattintunk elvisz minket a tanár tantárgyai oldalra. A tanár tantárgyai oldalon listába találhatóak a tanár tanított tantárgyai.

A navigációs menüben a Class-ra kattintva elvisz minket az Osztályok oldalra. Ezen az oldalon az adatbázisban szereplő osztályok jelennek meg növekvően külön-külön dobozokban. A dobozban található egy a témának megfelelő ikon az ikon alatt pedig az adott osztály neve. Az osztály neve alatt szerepel a diákok névsora egy táblázatban.

A navigációs menüben a Subject-re kattintva elvisz minket a Tantárgyak oldalra. A tantárgyak oldalon az adatbázisban szereplő összes tantárgy jelenik meg ABC sorrendben egy táblázatban.

A navigációs menüben a TimeTable-re kattintva elvisz minket az Órarend oldalra. Az órarend oldalon az adatbázisban szereplő osztályok jelennek meg a hozzájuk kapcsolódó órarenddel osztályonként külön-külön dobozokban. A dobozok felső részén a témához megfelelő ikon található az ikon alatt az osztályok nevei jelennek

meg. Az osztályok nevei alatt a napok kezdőbetűivel gombok jelennek meg (angol nevű napok). A gombokra kattintva az adott naphoz, illetve osztályhoz tartozó órarend jelenik meg egy táblázatban.

A navigációs menüben a Login Administrator-ra kattintva elvisz minket az Adminisztrátor oldalra egy bejelentkező felületet követően melybe Administrator felhasználónév és asdfg jelszó után tudunk bejelentkezni.

Bejelentkezést követően egy Adminisztrátor főoldal fogad minket. A főoldalon található szintén egy navigációs bár fogad minket. Az adminisztrátorhoz kapcsolódó gombokkal. A navigációs bár alatt egy az oldalhoz kapcsolódó leírás látható a gombokhoz tartozó funkciók leírásával.

A navigációs menüben az Add Teacher-re kattintva elvisz minket a Tanár hozzáadása oldalra. A tanár hozzáadása oldalon új tanárokat tudunk felvenni az adatbázisunkba OM azonosító, név, telefonszám, jelszó és admin szerint. Az oldalon megjelenik az összes tanár a hozzá tartozó adataival. A tanárok adatait egy gombra kattintva meg tudjuk változtatni. A megváltoztatás gomb mellett található egy törlés gomb is, amivel törölni tudjuk a tanárokat. Az adatokat formokba tudjuk beírni, aminek az összes rész kitöltése kötelező. OM azonosító duplikáció esetén az oldalunk hibát ad. Új tanárt az ADD gomb megnyomásával tudjuk megtenni. Sikeres felvételnél, illetve törlésnél üzenetet kapunk.

A navigációs menüben az Add Subject-re kattintva elvisz minket az új tantárgyak hozzáadása oldalra. A tantárgy hozzáadása oldalra kattintva az adatbázisunkba tudunk felvenni új tantárgyat a nevük megadása után. Az oldalon megjelenik az összes tantárgy egy táblázatban a hozzájuk tartozó adatokkal. A tantárgyak adatait egy gombra kattintva lehet megváltoztatni a megváltoztatás gomb mellett található egy törlés gomb is amivel lehet törölni az adott tantárgyat. Az adatokat formokba tudjuk beírni itt az összes mező kitöltése kötelező azonos adat esetén hibát dob az oldalunk. Új tantárgy felvételét az ADD gombra kattintva tudjuk megtenni. Sikeres felvétel és törlés esetén üzenetet kapunk az oldalon.

A navigációs menüben az Add Classroom-ra kattintva elvisz minket az új tanterem hozzáadása oldalra. A tanterem hozzáadása oldalra kattintva az adatbázisunkba tudunk felvenni új tantermet az ajtó és hely megadása után. Az oldalon megjelenik az összes terem egy táblázatban a hozzájuk tartozó adatokkal. A termék adatait egy gombra kattintva lehet megváltoztatni a megváltoztatás gomb mellett található egy törlés gomb is, amivel lehet törölni az adott tantermet. Az adatokat formokba tudjuk beírni itt az összes mező kitöltése kötelező azonos adat esetén hibát dob az oldalunk. Új tanterem felvételét az ADD gombra kattintva tudjuk megtenni. Sikeres felvétel és törlés esetén üzenetet kapunk az oldalon.

A navigációs menüben az Add Student-re kattintva elvisz minket az új diák hozzáadása oldalra. A diák hozzáadása oldalra kattintva az adatbázisunkba tudunk felvenni új diákot az OM azonosító és név megadása után. Az oldalon megjelenik az összes diák egy táblázatban a hozzájuk tartozó adatokkal. A diákok adatait egy gombra kattintva lehet megváltoztatni a megváltoztatás gomb mellett található egy törlés gomb is, amivel lehet törölni az adott tantárgyat. Az adatokat formokba tudjuk beírni itt az összes mező kitöltése kötelező azonos adat esetén hibát dob az oldalunk. Új diák felvételét az ADD gombra kattintva tudjuk megtenni. Sikeres felvétel és törlés esetén üzenetet kapunk az oldalon.

A navigációs menüben az Add Class-ra kattintva elvisz minket az új osztály hozzáadása oldalra. Az osztály hozzáadása oldalra kattintva az adatbázisunkba tudunk felvenni új osztályokat a nevük megadása után. Az oldalon megjelenik az összes osztály egy táblázatban a hozzájuk tartozó adatokkal. Az osztály adatait egy gombra kattintva lehet megváltoztatni a megváltoztatás gomb mellett található egy törlés gomb is, amivel lehet törölni az adott tantárgyat. Az adatokat formokba tudjuk beírni itt az összes mező kitöltése kötelező azonos adat esetén hibát dob az oldalunk. Új osztály felvételét az ADD gombra kattintva tudjuk megtenni. Sikeres felvétel és törlés esetén üzenetet kapunk az oldalon.

A Log out gombra kattintva egy megerősítő oldal jelenik meg biztosan ki szeretne lépni felirattal (angolul). A logout gombra kattintva pedig kijelentkezhetünk az adminisztrátor oldalról. Ismét a bejelentkező oldal fogad minket ezután.

A főoldalra visszatérve a Login Teacher gombra kattintva elvisz minket a Tanár oldalra egy bejelentkező felületet követően melybe 71123567012 felhasználónévvel és asdfg jelszó után tudunk bejelentkezni.

Bejelentkezést követően egy Tanár főoldal fogad minket. Navigációs menüvel és az oldalhoz kapcsolódó leírással a navigációs menü alatt.

A TimeTable legördülő menüre rá kattintva a napok jelennek meg egymás alatt felsorolva angolul. Egy adott napra kattintva az adott napi órarend látható egy táblázatban, ami a bejelentkezett tanár órarendje. A napok alatt található egy back gomb is, amely a főoldalra visz vissza.

A My gombra kattintva a tanár adatai jelennek meg egy témához megfelelő ikonnal a doboz tetején. Az ikon alatt a tanár saját adatai láthatóak egymás alatt. Az adatok alatt a tanár órarendje gombra kattintva elvisz minket a tanár órarendje oldalra, ahol megjelenik a tanár összes tanított órája, ami az adatbázisban szerepel. A tanár órarendje gomb alatt található egy adat módosító gomb, amire, ha rákattintunk átvisz minket egy Tanár adatai módosító oldalra, amelyben a tanár neve, telefonszáma és

jelszava módosítható az Update gombra kattintás után. A sikeres frissítés esetén üzenetet kapunk.

A Log out gombra kattintva egy megerősítő oldal jelenik meg biztosan ki szeretne lépni felirattal (angolul). A logout gombra kattintva pedig kijelentkezhetünk a tanár oldalról. Ismét a bejelentkező oldal fogad minket ezután.

A Login Administrator-nál és Login Teacher-nél egyszerre nem tudunk bejelentkezi. Mielőtt be akar jelentkezni bármelyik felületre előtte ki kell lépni a másikból.



# Tartalom

- Borító 1. oldal
- Szereplők és igényeik 2-3. oldal
- Felhasználók és igényeik 4-5. oldal
- Magas szintű rendszerterv 6. oldal
- Screenek 7-29. oldal
- Modellek 30-33. oldal
- Alkalmazások kiválasztása 34. oldal
- Routing 35-38. oldal
- Middleware-ek 39. oldal
- Implementáció, Tesztelés 40-60
- Felhasználói dokumentáció 61-64