

# SZAKDOLGOZAT

Szabó Anett

Szikora Melinda

Váradi Ákos

2022

# ATLANTIS CINEMA

Szabó Anett, Szikora Melinda,

Váradi Ákos

SZÁMALK-Szalézi Technikum és  
Szakgimnázium

Szoftverfejlesztő

## Nyilatkozat

a záródolgozat eredetiségéről

Alulírott

büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a záródolgozat saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a záródolgozat készítésre vonatkozó szabályokat betartottam.

Kijelentem, hogy ahol mások eredményeit, szavait vagy gondolatait idéztem, azt a záródolgozatomban minden esetben, beazonosítható módon feltüntettem, a dolgozatban található fotók és ábrák közlésével pedig mások szerzői jogait nem sértem.

Kijelentem, hogy a záródolgozatom elektronikus változata teljes egészében megegyezik a nyomtatott formával.

Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Szakgimnázium belső szabályzata alapján az iskola saját könyvtárában megtekinthető (olvasható) legyen a záródolgozatom.

A záródolgozat titkos/nem titkos.

Budapest, 2022. május

---

Tanuló aláírása

# Tartalom

1	Bevezetés, témaválasztás .....	7
1.1	Telepítés és indítás .....	9
1.2	Szükséges hardver eszközök és szoftverek .....	10
1.3	Futtatási környezet .....	11
2.	Tervezés, drótvázak .....	11
2.1	Fejléc, lábléc, menü .....	12
2.2	Nyitó oldal .....	13
2.3	Film oldal .....	16
2.4	Foglalás oldal .....	18
2.6	Regisztráció oldal.....	21
2.7	Bejelentkezés oldal .....	21
2.8	Jelszó visszaállítás oldal.....	21
2.9	Autentikáció oldal .....	21
2.10	Admin fejléc, menüsor.....	22
2.11	Személyek oldal.....	22
2.12	Filmek oldal .....	22
2.13	Termek oldal .....	22
2.14	Vetítések oldal .....	23
2.15	Kedvezmény oldal .....	23
2.15	Dolgozók oldal.....	23
2.16.	Pénztár főoldal .....	24
2.17.	Pénztár foglalási oldal.....	26
3	Felhasználói dokumentáció.....	28
3.1	A program célja és lényegesebb funkciói .....	28
3.1.1	Publikus felhasználói felületek .....	28
3.1.2	Pénztár oldalak.....	28
3.3	A program fő funkcióinak leírása .....	29
3.3.1	Nyitó és film oldal.....	29
3.3.2	Admin oldalak.....	30
3.3.3	Pénztár oldalak.....	35
4	Fejlesztői dokumentáció .....	38

4.1	Fejlesztéshez használt technológiák.....	38
4.2	Mappa szerkezet.....	39
4.3	Adatbázis.....	39
4.3.1	Táblák .....	40
4.3.2	Kapcsolati ábra.....	42
4.4	Backend.....	43
4.4.1	Modellek és Controllerek.....	43
4.4.1.1	Publikus felhasználói felületek .....	43
4.4.1.2	Admin Backend.....	47
4.4.1.3	Pénztáros oldalak .....	47
4.4.2	API végpontok .....	49
4.4.2.1	Publikus felhasználói felületek .....	53
4.5	Frontend .....	54
4.5.1	Publikus felhasználói felületek .....	54
4.5.2	Admin oldalak.....	66
4.5.3	Pénztáros oldalak .....	68
4.6	Reszponzivitás .....	75
4.6.1	Publikus felhasználói felületek .....	75
4.6.2	Admin felületek.....	79
4.6.3	Pénztáros oldalak .....	79
5	Tesztelés.....	82
6	Fejlesztési lehetőségek.....	89
6.1	Publikus felhasználói felületek .....	89
6.2	Admin felületek .....	89
6.3	Pénztáros oldalak .....	89
7	Irodalomjegyzék .....	90

# Ábrajegyzék

1. ábra, Nyitó oldal terve .....	14
2. ábra, Végleges nyitó oldal .....	15
3. ábra, Végleges nyitó oldal .....	15
4. ábra, Végleges nyitó oldal .....	16
5. ábra, Film oldal terve .....	17
6. ábra, Végleges film oldal .....	18
7. ábra, Foglалás oldal terve .....	19
8. ábra, Végleges foglalás oldal .....	20
9. ábra, A pénztár oldal kezdetleges terve .....	24
10. ábra, A pénztár főoldal végleges formája .....	25
11. ábra, A pénztár foglalási oldal terve .....	26
12. ábra, A pénztár foglalási oldal végleges formája .....	27
13. ábra, Adatbázis kapcsolati ábra .....	42
14. ábra, Egyszerű műfaj modell .....	43
15. ábra, Kezdetleges kontroller, film-műfaj .....	44
16. ábra, Film modell .....	45
17. ábra, Nyitó oldal fő kontrollere .....	46
18. ábra, Nyitó oldal fő kontrollerének route-ja .....	53
19. ábra, Film osztály konstruktora .....	57
20. ábra, Film osztály set metódusa és eseményei .....	58
21. ábra, Ajax osztály getAdat metódusa .....	59
22. ábra, Nyitó oldal fő szkript, filmMegjelenites .....	60
23. ábra, Film elem sablon .....	61
24. ábra, Nyitó oldal fő szkript, filmek megjelenítése .....	62
25. ábra, Nyitó oldal fő szkript, localStorage.setItem .....	63
26. ábra, Film oldal fő szkript, localStorage.getItem .....	63
27. ábra, Film oldal fő szkript, FilmOldal létrehozása .....	64
28. ábra, Galéria osztály .....	64
29. ábra, Galéria megjelenítés és lapozás .....	65
34. ábra, Nyitó oldal, tablet nézet .....	76
35. ábra, Nyitó oldal, mobil nézet .....	77
36. ábra, Film oldal, mobil nézet .....	78

# 1 Bevezetés, témaválasztás

Szikora Melinda

Szakdolgozatunk témájának egy fiktív mozit választottunk, közös ötletelés eredményeképp, amit Atlantis Cinema-nak neveztünk el. A mozi egészét meghatározza az erre utaló tengeri téma és színek.

Létrehoztuk a weboldalát, ahol a nézők tudnak tájékozódni az épp műsoron lévő filmekről, valamint elkészítettük a mozi pénztárosának felületét, ahol a dolgozó meg tudja tekinteni a foglalásokat, és el tudja adni a jegyeket. Az oldalon felhasznált film adatok tárolásához a mozinak adatbázist hoztunk létre, melyet admin jogosultsággal, külön oldalon lehet kezelni.

## **Feladatok:**

### **Felhasználók számára készült publikus oldalak**

Szikora Melinda

- Dizájn kialakítása
- Felhasználói oldalak létrehozása
- Reszponzivitás megvalósítása a felhasználói oldalakon
- Filmek megjelenítése
- Időpontok megjelenítése
- Galéria létrehozása
- Napok átváltása
- Kattintott film továbbvitele a film oldalra és annak megjelenítése
- Néző és foglalás felvitele

## **Admin oldalak**

Váradi Ákos

- Dizájn kialakítása
- Filmek, termék, személyek, székek, vetítések, műfajok megjelenítése
- Filmek, termék, személyek, székek, vetítések felvitele
- Filmek, termék, személyek, székek, vetítések, dolgozók módosítása
- Filmek, termék, személyek, vetítések, dolgozók törlése
- Regisztráció
- Bejelentkezés

## **Pénztár oldalak**

Szabó Anett

- Dizájn kialakítása
- Filmek megjelenítése
- Filmekhez tartozó időpontok megjelenítése
- Nézők megjelenítése
- Nézők közötti keresés, rendezés
- Nézők felvitele az adatbázisba
- Székek módosítása az adatbázisba



## 1.1 Telepítés és indítás

Szabó Anett

Ajánlott webszerver: Apache webszerver, legalább php 8.1, MySql adatbázis

- Nyitó oldal: <http://82.150.42.74:8081/>
- Bejelentkező oldal: <http://82.150.42.74:8081/home>

Az alábbi oldalak csak bejelentkezés után érhetőek el!

Mindkét esetben kötelező az e-mail cím és a jelszó megadása!

- Admin oldal: <http://82.150.42.74:8081/film>

E-mail cím: [varadiakos@atlantiscinema.com](mailto:varadiakos@atlantiscinema.com)

Jelszó: password

- Pénztár oldal: <http://82.150.42.74:8081/penztar>

E-mail cím: [szaboanett@atlantiscinema.com](mailto:szaboanett@atlantiscinema.com)

Jelszó: password

## 1.2 Szükséges hardver eszközök és szoftverek

Szabó Anett

Hardver eszközök:

Személyi számítógép (PC, Notebook):

- 4 GB RAM
- 10GB szabad tárhely
- HD felbontást (1920\*1080) támogató monitor
- Windows 10, vagy a legfrissebb operációs rendszer
- Internetkapcsolat

Mobiltelefon:

- 2GB RAM
- Internetkapcsolat

Szoftverek:

- XAMPP Control Panel v3.3.0, <https://www.apachefriends.org/hu/download.html>
- GitHub, <https://git-scm.com/downloads>
- Composer, <https://getcomposer.org/download/>

## 1.3 Futtatási környezet

Böngészők:

- Google Chrome \*Verziószám: 100.0.4896.75\*
- Microsoft Edge \*Verziószám: 100.0.1185.39 \*
- Firefox \*Verziószám: 99.0.1\*

## 2. Tervezés, drótvázak

Szikora Melinda

Tervezéskor meghatároztuk, hogy a mozi a vetítéseit napokra alakítja ki, amiket az oldalakon hetente töltünk be. A filmek vetítései meghatározott időpontokban kezdődnek. Idegen és magyar nyelvű filmek is kerülnek a műsorra és az idegen nyelvű filmekhez csak magyar felirat van.

Az adatbázis tervezésekor a táblák kialakításakor meghatároztuk a szükséges adatokat, amiket a későbbiekben meg akarunk jeleníteni az oldalakon.

Miután ezeket a fontosabb tartalmi részeket megterveztük, elkezdtük kialakítani az Atlantis Cinema látványtervét és design-ját. A színpaletta meghatározása után - kék, fehér, arany, fekete - kiválasztottuk a fő háttereket, végül pedig a mozi nevéhez használt jellegzetes betűtípust.

Ezek után kezdtük el kialakítani magukat az oldalakat. Minden oldalhoz több drótváz terv is készült, ezekből válogattuk össze azokat az elemeket és kialakításokat, amik mindannyiunknak tetszettek. Így a végén egy összeállított terv született minden oldalhoz, amelyekhez még leírásokat készítettünk az elképzelt funkciókról.

A publikus felhasználói oldalak kinézetéhez sok létező mozi weboldalát átnéztük. Ötleteket gyűjtöttünk, hogy milyen elemek megjelenítése lehetséges, nekünk milyenek tetszenek. Miután kikértem a csapattársaim tanácsát, hogy ők milyen formátumban látnák szívesen az oldalakat, több drótvázatot is elkészítettem, a lehetséges elrendezésekről.

Három terv közül mindenki választott egyet, ami a legjobban tetszik neki. Ezek a választások sajnos nem ugyan azokra az oldalakra estek, így elkezdtük részletesebben elemekre szedni az oldalakat. Minden változatból kiválasztottuk, hogy mi az az elem, amit mindenképp szeretnénk megtartani. Ilyen volt például a nyitó oldalon a galéria az oldal tetején, a film oldalon pedig a YouTube lejátszó. Így végül az egyeztetések után létrehoztam egy végső tervet, az előzőek össze gyűrásával és átalakításával, amiben benne volt minden kiválasztott egység.

## **2.1 Fejléc, lábléc, menü**

Szikora Melinda

A fejléc, a lábléc és a menü minden publikus oldalon megegyezik. A fejlécben a mozi neve látható legfelül, alatta helyezkedik a menü, ahol a nyitó oldalra lehet navigálni, valamint meg lehet tekinteni külön oldalakon az árakat, akciókat, a házirendet és a kapcsolati lehetőségeket.

A láblécben szintén fel van tüntetve a mozi neve, információk és az email cím, telefonszám. Közösségi média gombok is találhatók a kapcsolat részen, amik létező mozi esetén a Facebook és Twitter oldalakra irányítanak. Ide még be van ágyazva egy Google térkép, ami a mozi földrajzi helyét mutatja. Végül pedig legalul fel vannak tüntetve a készítő és a projekt neve.

Az eredeti tervek szerint ez a térkép és a közösségi média gombok nem a láblécben lettek volna elhelyezve, hanem az oldal testének egy szekciójában a filmek alatt. Ezeknek a láblécbe való levitelének a célja az volt, hogy a így minden oldalon megjelenhetnek, egységes formában.

## 2.2 Nyitó oldal

Szikora Melinda

A nyitó oldal felső részén egy galéria szekció helyezkedik el, a néző meg tudja tekinteni az éppen műsoron lévő filmek plakátjait, tud ezek között lapozni. Mellette pedig tájékozódni tud akcióinkról, és esetleges eseményekről.

Az oldal lényege, a műsoron lévő filmek bemutatása a nézőknek. Itt egy menüben ki tudja választani, hogy a hét mely napján nézne filmet, alatta pedig megjelennek felsorolva, egymás alatt a filmek.

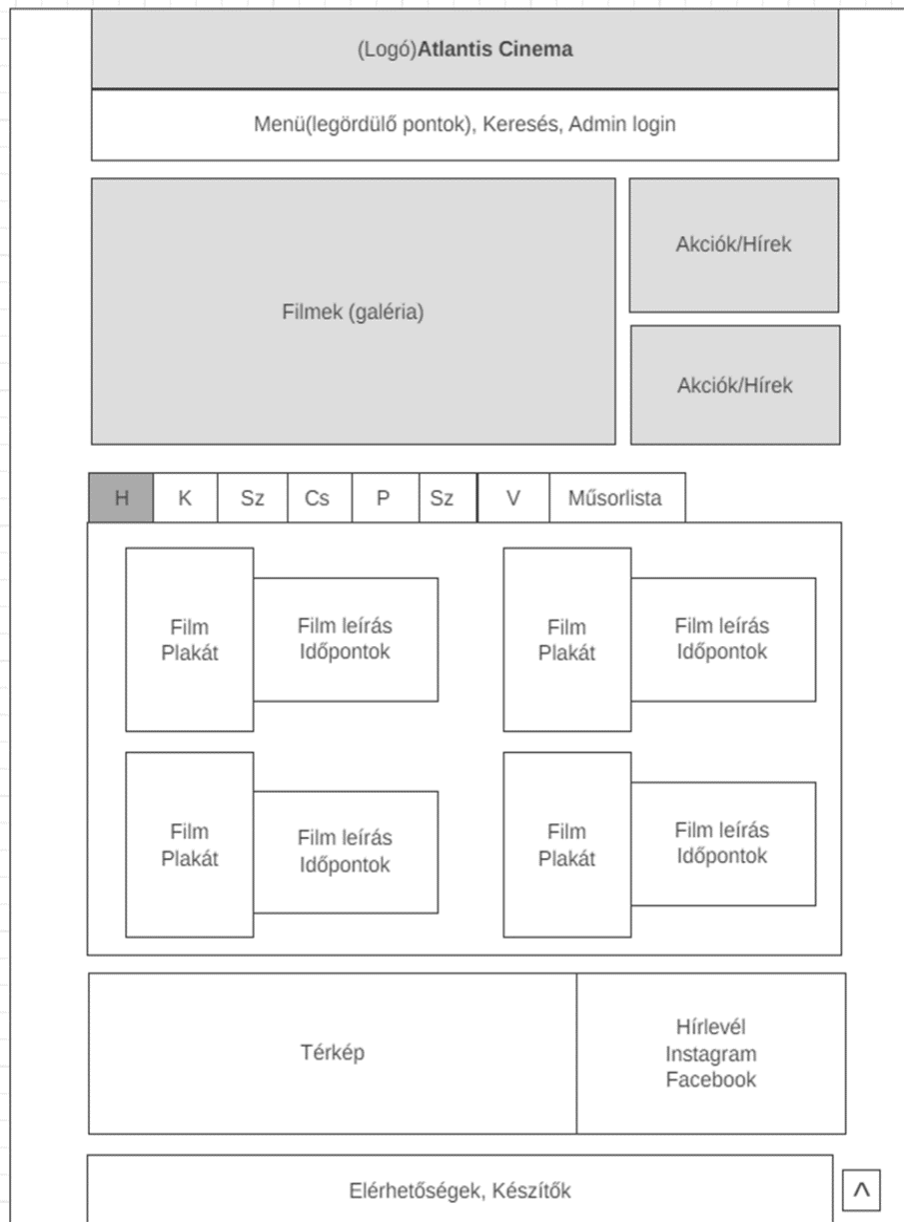
Egy film elemen belül, a bal szélén, látható a plakátja amire, ha ráviszik az egeret, akkor az kinagyul, így jobban meg tudják nézni. A jobb oldalon pedig egymás alatt helyezkedik el a film címe, műfaja, leírása és a vetítési időpontok felsorolva.

A film címére kattintva a betöltődik a film oldal, ahol az adott filmet tekintheti meg a néző nagyban, részletesebb adatokkal. Egy időpontra kattintva pedig egyből a foglalás oldalra irányítja tovább.

Tablet nézetben az akciók és hírek eltűnnek, így csak a galéria látszik a filmek fölött. A napok menüszalag pedig lenyitható menüvé válik, így kevesebb helyet foglal.

Ennél kisebb képernyőn, mobilos nézetben, már a galéria se látszik, a főmenü pontjai és a vetítési időpontok pedig egymás alá csúsznak. A leírás is eltűnik, ekkor már csak a film oldalán olvasható el.

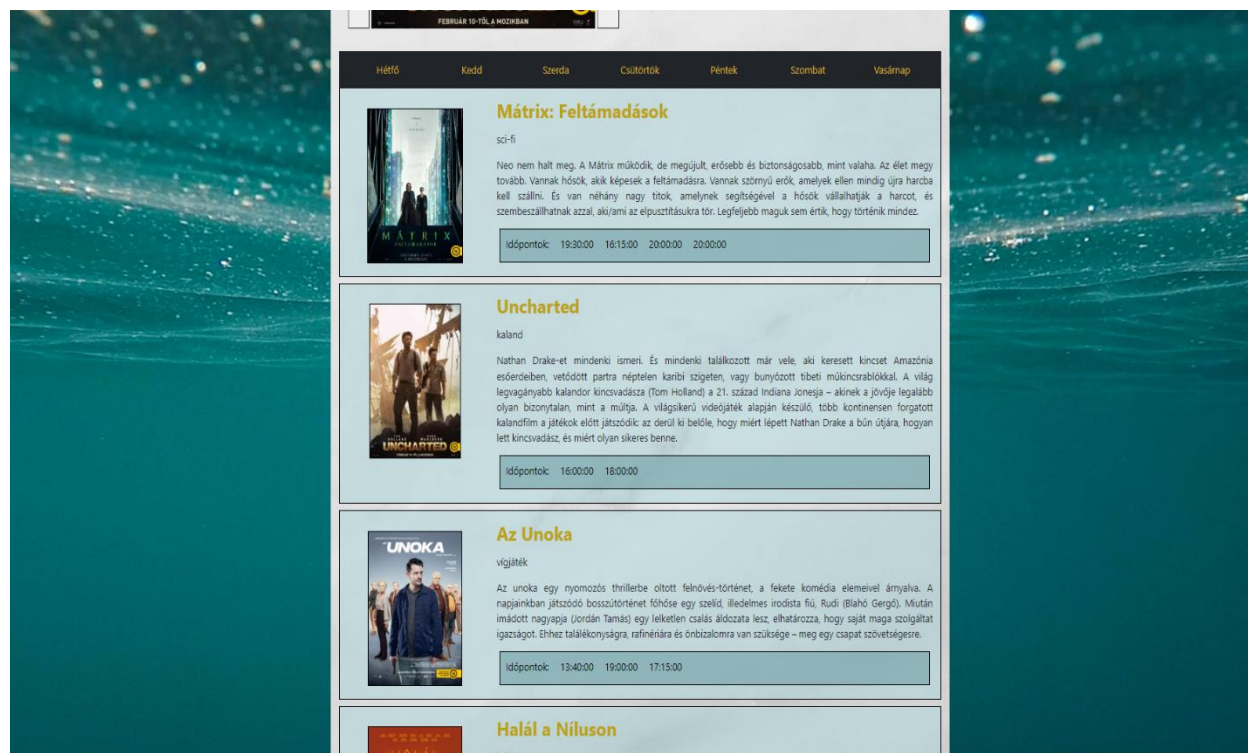
A tervhez képest a film elemek egy sorosra lettek módosítva, így nagyobbak és jobban áttekinthetőek. A térkép és a közösségi média rész le lett vité a láblécbe, így minden oldalon elérhető.



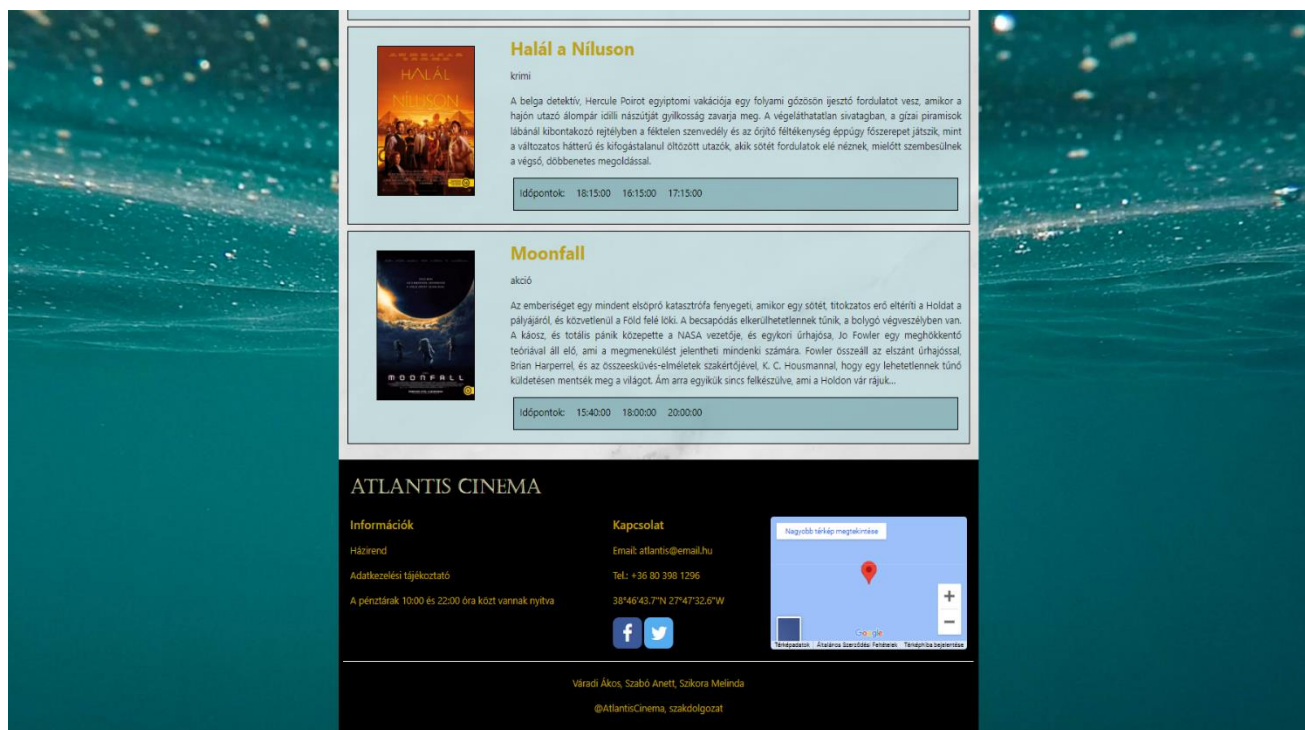
1. ábra, Nyitó oldal terve



3. ábra, Végleges nyitó oldal



2. ábra, Végleges nyitó oldal



4. ábra, Végleges nyitó oldal

## 2.3 Film oldal

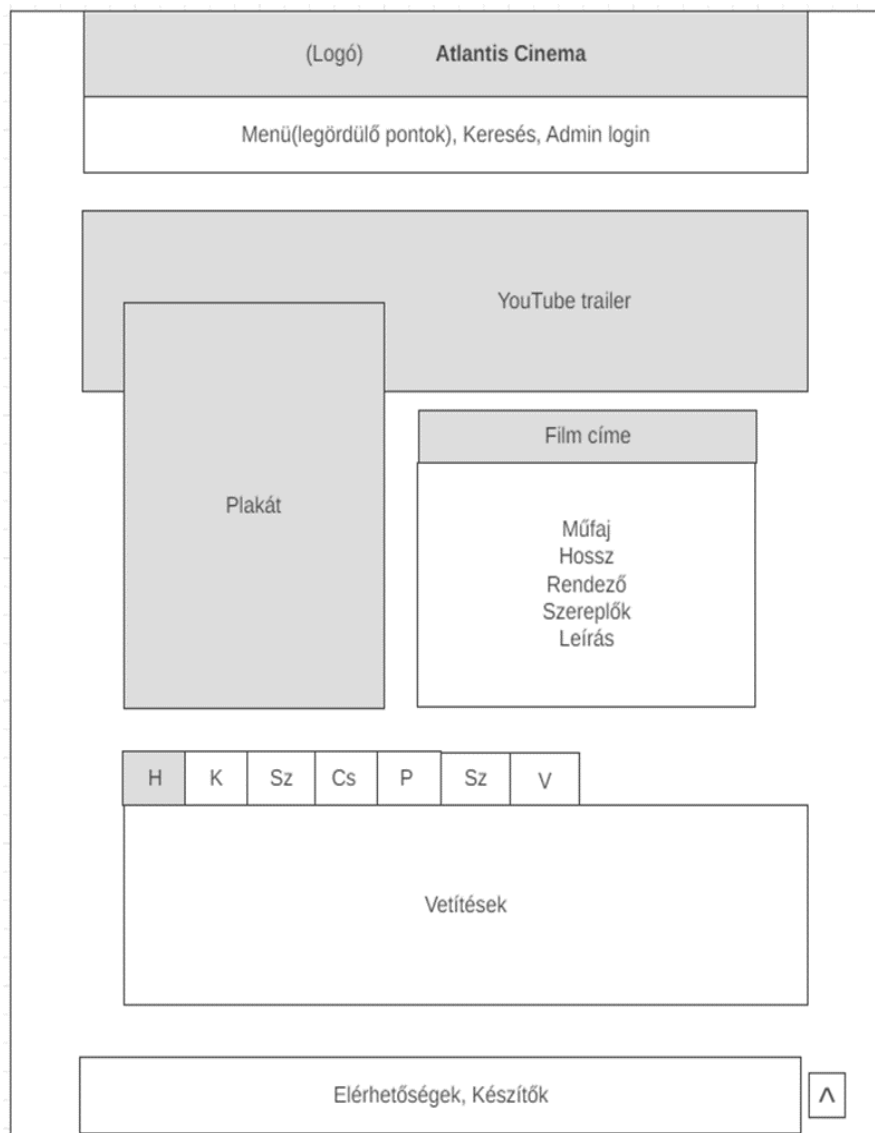
Szikora Melinda

A nyitó oldalon a film címére kattintva jutunk a film oldalra, ahol csak az adott film jelenik meg. Itt található egy YouTube előzetes beágyazva az oldalra, felül. Alatta a nyitó oldalhoz hasonlóan jelenik meg a film, bal oldalon a film plakátja, mellette pedig a cím, műfaj, leírás kiegészülve a film hosszával, a rendezővel és színészekkel.

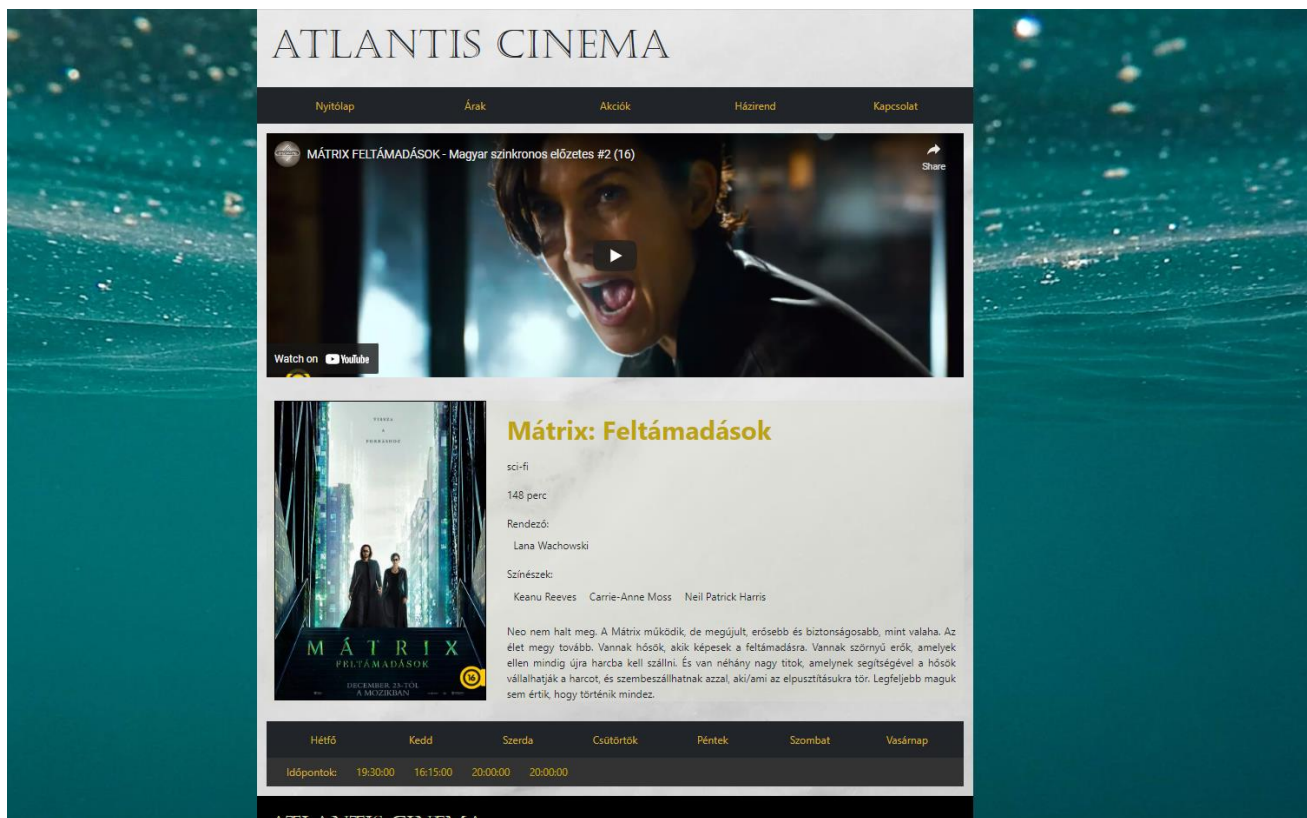
Ezek alatt pedig a filmhez tartozó vetítési időpontok találhatóak meg, itt szintén lehet napot választani, egy időpontra kattintva pedig betöltődik a foglalás oldal.

Kisebb képernyőkön, tableten és mobilon a film plakátja nem jelenik meg, a vetítések nap választója pedig, mint a nyitó oldalon lenyitható menüvé változik.





5. ábra, Film oldal terve



6. ábra, Végleges film oldal

## 2.4 Foglalás oldal

Szikora Melinda

Egy vetítés időpontra kattintva, betölt a foglalás oldal. Az eredeti tervek szerint itt külön oldalon adta volna meg a néző először az adatait, majd a következő oldalon választotta volna ki a jegyét és helyét. A végső formában ezek egy oldalra kerültek, az egyszerűség érdekében.

Az oldal tetején láthatóak a választott film adatai, a cím, a vetítés nyelve, napja és időpontja. Ez alatt a néző adatainak bekérése található, ahol nevet és email címet kell megadni, lejjebb pedig a jegy választó mezők vannak. Végül látható a terem rajza, amin kiválaszthatóak a székek.

Legalul találhatóak a foglalás gomb. Ezzel tudja a néző foglalni a helyeket, ekkor egy felugró ablak tájékoztatja annak sikerességéről, majd visszairányítja a nyitó oldalra.

A vásárlás funkció jelen programban nincs üzemben, a bankkártya adatok kezelésének kényessége végett.


(Logó) Atlantis Cinema	
Menü(legördülő pontok), Keresés, Admin login	
Választott Film	Film adatai Választott vetítés
Név megadása	
E-mail cím megadása	
Személy ellenőrzése (nem robot)	
Tovább	
Elérhetőségek, Készítők	

(Logó) Atlantis Cinema	
Menü(legördülő pontok), Keresés, Admin login	
Választott Film	Film adatai Választott vetítés
Jegyek számának és típusának megadása	
Terem rajz Szék(ek) kiválasztása	
Foglalás	Vásárlás
Elérhetőségek, Készítők	

7. ábra, Foglalás oldal terve


ATLANTIS CINEMA

NyitólapÁrakAkiókHázi rendKapcsolat



Mátrix:  
Feltámadások

2022-05-05  
20:00:00  
angol



Néző adatai

Név\*: Kiss Zoli

Email\*: zoli@email.hu

Székek száma\*:

	1	2	3	4	5	6	7	8	9	10
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

SzabadÉppen foglalEldott

FoglalásVásárlás

ATLANTIS CINEMA

Információk

Házi rend

A pénztárak 10:00 és 22:00 óra közt vannak nyitva

Kapcsolat

Email: atlantis@email.hu


Tel: +36 80 398 1296

38°46'43.7"N 27°47'32.6"W

f

t

Nagyobb térkép megtekintése



8. ábra, Végleges foglalás oldal

20

## **2.6 Regisztráció oldal**

Váradi Ákos

Egy egyedi Http cím beírása után eljutunk a bejelentkezés oldalra, ahol a felső menüsorban a mozi címe, valamint a bejelentkezés és a regisztráció gomb található. A regisztráció gombra kattintva a regisztráció oldalra érkezünk. Ezen az oldalon tudják a dolgozók regisztráltatni magukat egy név, email, jelszó és egy jelszó megerősítés segítségével.

## **2.7 Bejelentkezés oldal**

Váradi Ákos

A dolgozók itt tudnak bejelentkezni egy email és jelszó páros segítségével. Az adataikat tudják menteni az adatok megjelenítés bepipálásával.

## **2.8 Jelszó visszaállítás oldal**

Váradi Ákos

Amennyiben a dolgozó elfelejtette a jelszavát, akkor az elfelejtett jelszó részre kattintva eljutnak a jelszó visszaállítás oldalra. Az email cím megadásával új jelszót tud magának igényelni.

## **2.9 Autentikáció oldal**

Váradi Ákos

A bejelentkezés után egy köztes oldalra jutunk, ahol a jogosultságuknak megfelelő link jelenik meg. Fent, a menüsorban a mozi neve és a dolgozó neve is látható.

## **2.10 Admin fejléc, menüsor**

Váradi Ákos

A fejléc és menüsor minden oldalon megegyezik. A fejléc teljesen megegyezik az autentikáció oldalon bemutatott fejléccel. A menüsor hat menüpontot tartalmaz, mégpedig az alábbiakat: Személyek, Filmek, Termek, Vetítések, Kedvezmény és Dolgozók.

## **2.11 Személyek oldal**

Váradi Ákos

Az oldalon lehet megadni a rendező és főszereplő nevét. Ezen felül lehet törölni, módosítani, keresni és rendezni az adatokat.

## **2.12 Filmek oldal**

Váradi Ákos

Az oldalon lehet felvinni új filmet. Lehetőség van az adatok törlésére, módosítására, keresésére és rendezésére.

## **2.13 Termek oldal**

Váradi Ákos

Az oldalon lehetőségünk van új termeket felvinni, de itt is törölhetjük, módosíthatjuk kereshetjük és rendezhetjük az adatokat.

## **2.14 Vetítések oldal**

Váradi Ákos

Egy új vetítés létrehozásán kívül lehet még törölni, módosítani, keresni és rendezni az adatokat. Valamint az egyes vetítésnél meghatározni, hogy premier vagy sem és publikussá tenni a vetítést.

## **2.15 Kedvezmény oldal**

Váradi Ákos

Az oldalon lehet megadni új kedvezményt. Ezen felül lehet törölni, módosítani.

## **2.15 Dolgozók oldal**

Váradi Ákos

Az oldalon lehet új dolgozót módosítani és törölni. Ezen felül lehet keresni és rendezni az adatokat.

## 2.16. Pénztár főoldal

Szabó Anett

### *Előzetes elképzelés*

A pénztároldalak kialakításánál arra törekedtem, hogy minél átláthatóbbak, és a pénztáros számára könnyen használhatóak legyenek. Törekedtem a minimalista stílusra (kevés szín), azonban azt is szerettem volna, hogy a felhasználó oldal főszíneivel passzoljon, ezért megtartottam a főszíneket (fekete, arany).

H 2021.01.01		A hét napjai,dátummal		Dátum	Név
		Aktuális nap			
Film cím	3.Terem 10:30 HU 3/57	Terem Időpont Szinron Hely	Terem Időpont Szinron		
Film cím					
Film cím					
Film cím					
Film cím					
Kijelentkezés					

9. ábra, A pénztár oldal kezdetleges terve



Bejelentkezés után látható, melyik dolgozó jelentkezett be. A navigációs mezőben az aktuális naptól számított egy héttel előre láthatóak a napok és a hozzájuk tartozó dátumok. Amennyiben rányomunk az egyik napra, és van az adott napon vetítés, akkor betöltődnek a filmcímek és a hozzájuk tartozó időpontok. A vetítéseknél láthatjuk a termeket, az időpontot és a szinkront. A “Foglalj” gombra kattintva átirányít a pénztáros foglalási oldalára.

Kijelentkezni a saját nevére kattintva tud, ahol megjelenik egy “kijelentkezés” opció, megnyomva átvissz a bejelentkezés oldalra.

Atlantis Cinema				Szabó Anett ▾		
Vasárnap	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat
2022-05-01	2022-05-02	2022-05-03	2022-05-04	2022-05-05	2022-05-06	2022-05-07
2022-05-02						
Harry Potter						
		Óriás terem 10:00:00 HU <a href="#">Foglalj!</a>				
Game of Thrones						
		Óriás terem 23:53:48 HU <a href="#">Foglalj!</a>	Óriás terem 20:30:00 HU <a href="#">Foglalj!</a>			
Az erő						
		Sárkány terem 10:00:00 HU <a href="#">Foglalj!</a>				

10. ábra, A pénztár főoldal végleges formája

## 2.17. Pénztár foglalási oldal

Szabó Anett

A kiválasztott vetítés után, betöltődik a foglalási oldal. A pénztáros itt is információt kap melyik filmről, melyik teremről, melyik időpontról és melyik napról van szó.

Baloldalon azokat a nézőket láthatjuk, akik már lefoglalták a jegyüket. Tudunk a nevük alapján köztük keresni. Ezen felül ABC sorrendbe, és vissza is tudjuk őket rendezni (A-Z vagy Z-A). Középen látható az ülésterv, láthatjuk az üres-, megvásárolt- és a foglalt székeket. Jobboldalon meg kell adnunk a vásárló nevét, az e-mail címét és azt, hogy hány darab székot szeretne vásárolni, majd ezek után tudjuk kiválasztani a kívánt székot/székeket.

A “Vásárlás” gomb megnyomása után megtörtént a vásárlás. A “Vissza” gombra kattintva átirányít a pénztár főoldalára.

Film cím,dátum,időpont,terem

Adott foglalások

Keresés Rendezés

Vászon

Szabad székek

DB

Diák 1500Ft

Jegytípus

Jegytípus

Foglalás Vásárlás

Vissza

11. ábra, A pénztár foglalási oldal terve

Keresés

Rendezés ▾

ID	Név	E-mail
85	Kiss Andrea	ka@gmail.com
86	Szabó Levente	szabolevente@gmail.com
87	Kovács Béla	bela76@gmail.com
88	Nagy Erika	erika9473@gmail.com
89	Szabó Anett	szanett888@gmail.com
90	Váradi Ákos	varadiakos@gmail.com
91	Szikora Melinda	szikoramelinda@gmail.com
92	Varga László	varga81@gmail.com
93	Molnár Áron	maron2001@gmail.com
94	Toth Imre	imre85@gmail.com

Sárkány terem  
11:00:00  
HU

1 2 3 4 5 6 7 8 9 10 11 12

1 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

2 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

3 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

4 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

5 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

6 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

7 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

8 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

9 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

10 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Szabad ☒ Eladott

Név\*:

Email\*:

Székek száma\*:

Kezdés

Vásárlás

Vissza

12. ábra, A pénztár foglalási oldal végleges formája

27

## **3 Felhasználói dokumentáció**

### **3.1 A program célja és lényegesebb funkciói**

#### **3.1.1 Publikus felhasználói felületek**

Szikora Melinda

A program publikus felhasználói felületeinek célja, hogy a nézőket a lehető legtöbb információval ellássák és hogy tájékoztasson a moziban vetített filmekről. Valamint, foglalni tudnak ezeknek a filmeknek a vetítéseire.

#### **3.1.2 Pénztár oldalak**

Szabó Anett

A legfőbb célja az volt, hogy a néző képes legyen a helyszínen is jegyet venni, és a dolgozó egy letisztult, könnyen használható felületen tudjon dolgozni.

Lényegesebb funkciói:

- Filmek, napok, vetítések, nézők megjelenítése
- Nézők közötti keresés, rendezés
- Vásárlás felvitele
- Székek módosítása

## **3.3 A program fő funkcióinak leírása**

### **3.3.1 Nyitó és film oldal**

Szikora Melinda

A mozi oldalának betöltésekor megjelennek műsoron lévő filmek, valamint a galéria azoknak a plakátjaival. A főmenü első pontja a nyitó oldalra mutat. Az árak menüpontra kattintva jegytípusok árazása található. Az akciók pont alatt érvényben lévő különleges ajánlatainkat láthatja a néző. Házi rendünk a következő menüpontban olvasható el, valamint a láblécben is fel van tüntetve a mozi szabályzata. Utolsó menüpontban a kapcsolat tartási lehetőségeket találják a vendégek.

A nyitóoldal galériájában a két oldalán található gombokkal tud lapozni. A filmek címe és időpontjaik kattinthatóak, a cím a film oldalt hozza be, az időpont pedig a vetítés oldalt.

Az oldalak láblécében található egy beágyazott térkép, melyen a mozi helye található. Interakcióba lehet lépni ezzel a térképpel a beágyazott kis formában is, valamint be lehet hozni külön ablakban.

A film oldalon található beágyazott YouTube lejátszóra kattintva a néző, itt helyben meg tudja tekinteni a film előzetesét.

Foglalás oldalon, a néző, adatainak megadása után tud széket választani, majd a foglalás gombra kattintva ezt tudja rögzíteni.

### **3.3.2 Admin oldalak**

Váradi Ákos

#### **Dolgozók felülete**

A felület eléréséhez külön HTTP címet (localhost/8000/home) kell beírni, ahol a dolgozók tudnak bejelentkezni. A bejelentkezés előtt először regisztrálniuk kell. Ezt, a felső menüsorban a regisztráció menüpontra kattintva tehetik meg, ezáltal eljutnak a regisztráció oldalra.

#### **Regisztráció**

Itt meg kell adnia a nevét, email címét, egy általa választott jelszót és a megerősítés érdekében meg kell ismételnie a jelszavát. A regisztráció gombra kattintva ellenőrzi az adatokat, és ha minden mezőt helyesen töltött ki, akkor létrejön az adatbázisban a felhasználói profil. Ezután a bejelentkezés oldalon már be tud jelentkezni. Viszont, ha nem megfelelően töltött ki egy vagy több mezőt, akkor figyelmeztetés jelenik meg a helytelenül kitöltött mező(k) miatt. Hiányos kitöltéskor az adott mezőnél egy figyelmeztetés jelenik meg. A menüsorban a bejelentkezés gombra kattintva visszajutunk a bejelentkezés oldalra.

#### **Bejelentkezés**

A dolgozó bejelentkezésekor meg kell adnia az email címét és jelszavát. A bejelentkezés gombra kattintva bejelentkezik a felhasználó, és tovább visz egy köztes oldalra, ahol a jogosultságának megfelelő link jelenik meg, és ennek segítségével tovább mehet az adott jogosultságú oldalra. A bejelentkezés oldalon az adatok megjegyzése rész bepipálásával megjegyzi az email címét és jelszavát a dolgozónak. Alatta van egy “elfelejtett jelszavak” link, ami átvisz egy jelszó visszaállítás oldalra.

## **Jelszó visszaállítás**

Itt meg kell adnia az email címét és ha rányom a “Jelszó-visszaállítási” link küldése gombra, akkor a dolgozó a megadott email címre kap egy levelet, amiben egy link található, aminek segítségével vissza tudja állítani az email címét. Viszont, ha nem adja meg az email címét, vagy ha olyan email címet ad meg, ami nem szerepel az adatbázisban, akkor egy figyelmeztetést kap.

## **Admin**

A jogosultsági linkre kattintva a dolgozót átirányítja a vagy a pénztáros vagy az admin kezdő oldalára. Az admint a film oldalra irányítja. Az oldal tetején a mozinak a neve jelenik meg és a dolgozó neve látszik. A mozi nevére kattintva visszairányít minket a bejelentkezés oldalra. A nevére kattintva egy lenyíló rész jelenik meg aminek kijelentkezés menüpontjára kattintva kijelentkezik a dolgozó és átirányítja a mozi nyitó oldalára. Ezek alatt található a menüsor, aminek segítségével az adott jogosultsággal elérhető oldalak között váltogathatunk. Az alábbi menüpontokat jelennek meg.

## **Személyek**

Itt egy új főszereplőnek vagy rendezőnek a nevét vehetjük fel. A név beviteli mezőben megadhatjuk a nevét és a felvitel gombra kattintva elmenti az adatbázisba. Ha nem töltötte ki a mezőt vagy megsértette a mező szabályait és a felvitel gombra kattintott akkor egy lebegő ablak jelenik meg azzal a felírással, hogy ” Feltöltés sikertelen”. Az adatok törlődik a mezőből. Ahhoz, hogy folytatni tudjuk a munkát rá kell kattintanuk az ok gombra. Az adatokat táblázatos formában jelenítjük meg. Lehet keresni a nevek között a keresés mezőbe beírt karakterek vagy szavak alapján. Az egyezést a nevek között ellenőrzi. Amennyiben a keresésre nincsen egyezés akkor nem jelenik meg adat a táblázatban. Azok a tartalmak fognak megjelenni, amik ezekkel megegyeznek. A név alapján lehet rendeztetni növekvő vagy csökkenő abc sorrendbe. Az adatokat tudjuk törölni a törlés gombra kattintva vagy a módosítás gombra nyomva vissza tölteni az adatot a mezőbe. A felvitel gomb helyett módosít gomb jelenik meg. A nevet meg tudjuk változtatni és a módosít gomb segítségével menthetjük a változtatást az adatbázisba és a módosított tartalom fog megjelenni az oldalon is. Ha megsérti az adatbeviteli szabályt akkor egy lebegő ablak jelenik meg

azzal a felirattal, hogy ” Módosítás sikertelen”. Az adat törlődik a mezőből. A munkát akkor tudja folytatni, ha az ok gombra rákattint.

## **Filmek**

A filmek oldalon az admin felvehet egy új filmet, amihez meg kell adni a film címét, leírását, a film hosszát, ki kell választania a film nyelvét az előre megadott lehetőségek közül, a műfaját, a főszereplő(ke)t és a rendező(ke)t az adatbázisban tárolt adatokat tölti be és ezek közül tud választani, a film poszterét és a youtube linkjét. Amikor a felvitel gombra kattint az adatok bekerülnek az adatbázisba és a mezők kiürítésre kerülnek. Ellenkező esetben helytelen adatokat ad meg amik nem felelnek meg a mező szabályainak vagy ha nem tölti ki a mezőket akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy ” Feltöltés sikertelen”. Az adatbázisban tárolt adatokat külön álló elemekben jelenítjük meg. Amik között tudunk szűrni a keresés mezőbe beírt karakterek vagy szavakkal megegyező tartalmakra. Az egyezést a film címében és a leírásban ellenőrzi. Amennyiben a keresésre nincsen egyezés akkor nem jelenik meg adat a táblázatban. A tartalmat tudjuk rendeztetni a film cím alapján növekvő vagy csökkenő abc sorrendbe. Az egyes elemeket tudjuk törölni a törlés gombra kattintva. A gombra kattintva a tartalom törlődik az adatbázisból és az oldalon megjelenített tartalmak közül. A módosítás gomb segítségével vissza töltjük az adatokat a megfelelő beviteli mezőkbe. Ez után az adatokat meg tudjuk változtatni természetesen a megengedett adatbeviteli szabályokon belül. A felvitel gomb helyén módosít gomb jelenik meg és mikor rányomunk az adatbázisba a módosított adatok kerülnek bele és a felületen is a módosított adatok jelennek meg. A mezők kiürítésre kerülnek és a felvitel gomb visszakerül a módosít gomb helyére. A táblázatban megjelennek a módosított adatok. Viszont, ha megsérti az adatbeviteli szabályt akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy ”Módosítás sikertelen”. Az adatok törlődnek a mezőkből. A munkát akkor tudja folytatni, ha az ok gombra rákattint.

## **Termek**

Ahol új termet tudunk felvenni úgy, hogy megadjuk a terem nevét, sorok és oszlopok számát és utána rányomunk a felvitel gombra. Ha az admin helyesen töltötte ki a mezőket akkor az adatok mentésre kerülnek. Ellenkező esetben helytelen adatokat ad meg amik nem felelnek meg a mező szabályainak vagy ha nem tölti ki a mezőket akkor egy lebegő ablak jelenik meg azzal a



felirattal, hogy ” Feltöltés sikertelen”. Az adat törlődik a mezőkből. Ahhoz, hogy folytatni tudjuk a munkát rá kell kattintanuk az ok gombra. Az adatokat ezen az oldalon is táblázatos formátumban jelenítjük meg. Lehet keresni a termék nevei között a keresés mezőbe beírt karakterek vagy szavak alapján. Az egyezést a termék nevei között ellenőrzi. Olyan adatok fognak megjelenni, amik megegyeznek a keresés feltételével. Amennyiben a keresésre nincsen egyezés akkor nem jelenik meg adat a táblázatban. Lehet a terem név alapján a tartalmat rendezni növekvő vagy csökkenő abc sorrendbe. Ezen felül még lehet törölni az adatokat a törlés gombra kattintva. Valamint a módosítás gombra kattintva az adatok betöltődnek az aktuális mezőbe és ilyenkor tudjuk őket változtatni. A felvitel gomb helyett módosít gomb jelenik meg. A módosít gombra kattintva mentésre kerülnek a változások és utána kiürítésre kerülnek a mezők és a felvitel gomb visszakerül a módosít gomb helyére. A táblázatban megjelennek a módosított adatok. Viszont, ha megsérti az adatbeviteli szabályt akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy ”Módosítás sikertelen”. Az adatok törlődnek a mezőkből. A munkát akkor tudja folytatni, ha az ok gombra rákattint.

## **Vetítések**

Fel tudunk venni egy új vetítést. Ehhez a terem nevét és film címét az adatbázisban tárolt adatokból tölti be és ezek közül tudja kiválasztani a megfelelőt, a feliratos, szinkronizált részekenél előre megadott lehetőségek közül tudja bejelölni optimális adatot, a vetítés napját és kezdés időpontját a mezőjükbe kattintva egy lenyíló részből tudják kiválasztani. A vetítés napja mező melletti premier gomb segítségével állíthatja be, hogy az adott vetítés premier legyen és ilyenkor megváltozik a gomb kiírása ”Nem premier”-re és ha rákattint akkor nem lesz premier a vetítés. A teljes jegyár alapértelmezetten be van állítva, de a mezőbe kattintva ez módosítható. Ezek az adatok akkor kerülnek elmentésre, ha a felvitel gombra kattintunk. Viszont, ha valamelyik mezőt nem töltött ki vagy megsérti a mező biztonsági szabályát akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy ” Feltöltés sikertelen” és az adatok törlődnek a mezőkből. Csak akkor tudja folytatni a munkát, ha az ok gombra rákattintunk. Az adatok táblázatos formában jelennek meg. Az egyes vetítéseket lehet publikusra állítani, törölni és módosítani. Publikusra úgy tudjuk állítani, ha a publikus gombra rákattintunk ezután lesznek csak láthatók a vetítések a mozi fő oldalán és a pénztáros oldalon és csak ezután lehet az adott vetítésre helyet foglalni vagy vásárolni. A gomb ezután elszürkül és nem kattintható se módosítható ez az állapota. A törlésgombra kattintva tudunk

törölni adatot az adatbázisból és a táblázatból. A módosítás gomb segítségével tudjuk módosítani az értékeket, mégpedig úgy, hogy az adatok betöltődnek a megfelelő mezőbe és itt meg tudjuk változtatni őket. A felvitel gomb helyén a módosít gomb jelenik meg és ennek megnyomásával tudjuk menteni a változtatásokat. Ha megsérti a mezőszabályokat akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy "Módosítás sikertelen". A táblázat adatait tudjuk szűrni a keresőmezőbe beírt karakterekkel vagy szavakkal megegyező részek jelenjenek meg. Az egyezést a termék nevei és film címei között ellenőrzi. Amennyiben a keresésre nincsen egyezés akkor nem jelenik meg adat a táblázatban. Tudunk rendeztetni terem és film cím alapján növekvő vagy csökkenő abc sorrendbe.

## **Kedvezmény**

Ezen az oldalon tudjuk megadni az egyes jegyárak nevét, mennyi százalékos kedvezmény jár az egyes jegyárakra és egy leírást, hogy milyen szabályok érvényesek az egyes kedvezményekre. Ezeket egy űrlap segítségével tudja felvinni. Az előbb felsorolt adatokat adja meg a megfelelő mezőben és a felvitel gombra kattintva tudja felvinni az adatokat az adatbázisba. Amennyiben nem megfelelően vannak kitöltve az űrlap elemei vagy üresen lettek hagyva egy lebegő ablak jelenik meg azzal a felirattal, hogy "Feltöltés sikertelen" és az adatok törlődnek a mezőkből. Csak akkor tudja folytatni a munkát, ha az ok gombra rákattintunk. Az adatok táblázatos formában jelennek meg. Ezen az oldalon kivételesen nem lehet keresni, se rendezni mivel nem keletkezik annyi adat a jegyáraknál, hogy ezekre a műveletekre szüksége legyen. Az adatokat lehet törölni a törlés gombra kattintva. Valamint a módosítás gombra kattintva az adatok betöltődnek az aktuális mezőbe és ilyenkor tudjuk őket változtatni. A felvitel gomb helyett módosít gomb jelenik meg. A módosít gombra kattintva mentésre kerülnek a változások és utána kiürítésre kerülnek a mezők. A táblázatban megjelennek a módosított adatok. Viszont, ha megsérti az adatbeviteli szabályt akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy "Módosítás sikertelen". Az adatok törlődnek a mezőkből. A munkát akkor tudja folytatni, ha az ok gombra rákattint.

## **Dolgozók**

Ezen az oldalon főként csak módosítani tudjuk a már előre regisztrált dolgozók egyes adatait. Az űrlap alatti táblázatban láthatóak a dolgozók adatai. Lehet keresni a nevek között a keresés mezőbe beírt karakterek vagy szavak alapján. Azok a tartalmak fognak megjelenni, amik ezekkel megegyeznek. Az egyezést a dolgozók neve között ellenőrzi. Amennyiben a keresésre nincsen egyezés akkor nem jelenik meg adat a táblázatban. A név alapján lehet rendeztetni növekvő vagy csökkenő abc sorrendbe. Az adatokat tudjuk törölni a törlés gombra kattintva vagy a módosítás gombra nyomva vissza tölteni az adatot az űrlap mezőibe. A felvitel gomb helyett módosít gomb jelenik meg. Az űrlapban tudjuk módosítani a dolgozó nevét, email címét és a munkakörét, ami előre megadott opciók közül tud kiválasztani. A módosít gombra kattintva menti el az adatokat. Viszont, ha megsérti az adatbeviteli szabályt akkor egy lebegő ablak jelenik meg azzal a felirattal, hogy ” Módosítás sikertelen”. Az adatok törlődnek a mezőkből. A munkát akkor tudja folytatni, ha az ok gombra rákattint.

### **3.3.3 Pénztároldalak**

Szabó Anett

#### **Pénztárfőoldal**

A pénztároldalt bejelentkezés után érjük el. A bejelentkezésnél meg kell adni a korábban regisztrált nevet és e-mail címet. Amennyiben az adatbázisban is szerepel, és “Pénztáros” státusszal rendelkezik, akkor átirányít a pénztárfőoldalra.

Az oldal felső részén (fejlécen) látjuk, ki jelentkezett be, a névre rányomva és a “kijelentkezés” opciót választva tudunk kijelentkezni. Ezután a bejelentkezés oldalra kerülünk.

Az oldal navigáció részén a napokat és a hozzájuk tartozó dátumokat találjuk, amelyeket ugyancsak az aktuális naptól számított egy 1 héttel előre látunk. Először egy figyelmeztetés jelenik meg: “Kattintson egy dátumra”. A kattintás után megjelennek az adatbázisban lévő filmek, és a hozzájuk tartozó vetítések, de csak abban az esetben, ha van a kiválasztott napon az adatbázisban

vetítés és az publikus is. Amennyiben nincs, vagy nem publikus, azok a vetítések nem jelennek meg.

A filmek és a hozzájuk tartozó vetítések táblázatszerűen jelennek meg. A vetítéseknel látjuk hol, mikor és milyen nyelven vannak vetítések. A dátum és a film kiválasztása után, a “Foglalj” gombra kattintva, átugrunk a foglalási oldalra.

### **Pénztár foglalásioldal**

Ezt az oldalt az időpont kiválasztása után érjük el. A felső részen itt is látjuk, melyik dolgozó van bejelentkezve.

Az előző oldalról kapott információ alapján itt is megjelenik, hogy hol, mikor és milyen nyelven van a választott vetítés, így pontosan tudjuk melyik időpontról van szó. Ezzel egyidőben, baloldalon megjelennek a jegyfoglalók adatai.

Tudunk keresni a nevük alapján, valamint a név beírásával tudunk szűrni, és amennyiben egyezik az adatbázisban lévő névvel, akkor csak azok a nézők jelennek meg, amelyben szerepel a kiválasztott betű. Amennyiben nincs ilyen név az adatbázisban, akkor nem jelenik meg egy néző sem. A kereső már egy betű lenyomásával is képes szűrni. Itt megkeresi, van-e a névben ilyen betű és azokat a neveket jeleníti meg az adatbázisból, ahol van ilyen betű. Amelyikben nincs a keresett betű, azokat nem jeleníti meg. Ezen kívül tudunk a nevük alapján rendezni, itt két opció közül választhatunk: ABC sorrendben és visszafelé.

Középső részen látjuk a teremhez tartozó üléstervet és a vásznat. Akkor tudunk választani, ha kötelezően kitöltöttük a jobb oldalon található mezőt.

A vásárlás gomb kezdetben még nem kattintható. A név és az e-mail mezőt követ egy székek száma mező, ahol meg kell adni mennyi székot szeretnénk kiválasztani. Amennyiben az összes mezőt kitöltöttük, akkor a “Kezdés” gombra nyomva kapunk egy információt arról, hogy kezdhetjük a székek kiválasztását és már tudunk választani. Hiányos kitöltés esetén egy felugró ablak figyelmeztet az összes mező kitöltésére. Ezután ki tudjuk választani a székeket. A kiválasztás után fontos, hogy annyi székot válasszunk ki amennyit az “székek száma” mezőbe beírtunk, ha

nem így teszünk és rányomunk a “vásárlás” gombra, akkor szintén egy felugró ablak figyelmeztet arra, hogy annyi széket válasszunk amennyit az elején beírtunk. Ha minden rendben van, a “vásárlás” gombot megnyomva felkerül az adatbázisba a név, az e-mail cím és a kiválasztott székek adatai. Az oldalon is kapunk egy visszaigazolást a vásárlás sikerességéről.

## 4 Fejlesztői dokumentáció

### 4.1 Fejlesztéshez használt technológiák

Váradi Ákos

- HTML
- CSS
- JavaScript
- Bootstrap5
- Ajax
- jQuery
- PHP
- Laravel
- MVC
- XAMPP
  - Mariadb
- GitHub
- Visual Studio Code

## 4.2 Mappa szerkezet

Váradi Ákos

Az azonos típusú, tartalmú vagy funkcióval bíró fájlokat az átláthatóság és működés érdekében külön mappába szerveztük. A mappák nagy részét a Laravel keretrendszer biztosította mivel az MVC architektúrát valósítja meg és amire még szükség volt azokat létrehoztuk saját magunknak.

## 4.3 Adatbázis

Szikora Melinda

Az adatbázis tervezése és kialakítása közös munka volt. Miután kiválasztottuk a témát, ami a mozi lett, elkezdtek meghatározni a szükséges adatokat és így, ezek alapján kezdtük el létrehozni a táblákat.

Az adatbázis feladata a film, vetítés adatok tárolása, a termék és székek számon tartása, kedvezmény információk kezelése, valamint a foglalással rendelkező nézők és a dolgozók belépési adatainak tárolása.

Az Atlantis Cinema adatbázisa 10 táblát tartalmaz. Több esetben összekapcsolt táblákat kell használni a programban, hogy egy feladathoz minden szükséges adatot meg lehessen jeleníteni. Ezen adatok szétválasztásának az oka az adatbázisban az volt, hogy egyszerűen lehessen bővíteni az adattárat, például új műfajokkal, vagy korábban nem megjelenített rendezővel/színésszel.

### 4.3.1 Táblák

Szikora Melinda

Műfaj {**műfajId**, műfaj}

Személy {**személyId**, neve}

Film\_Személy {**filmId**, **személyId**, **poszt**}

Film {**filmId**, cím, filmleírás, hossz, nyelv, poszter, youtubeLink, *műfajId*}

Terem {**teremId**, teremnév, sor, oszlop}

Vetítés {**vetítésId**, *teremId*, *filmId*, felirat, szinkron, vetítésnap, kezdésidő, teljesjegyar, premier, publikus}

Néző {**nézőId**, név, email}

Szék {**sor**, **oszlop**, **vetítésId**, *nézőId*, *kedvezményId*, elvesztesidőpont, állapot}

Kedvezmény {**kedvezményId**, típus, százalék, leírás}

User {**id**, name, email, password, munkakor}

A műfaj tábla azonosítókhoz tartozó műfaj megnevezéseket tartalmaz, melyek a filmekhez vannak kapcsolva a későbbiekben.

A filmekben dolgozó rendezőket és színészeket a személy táblában gyűjtjük. Itt a személyek azonosítókkal vannak ellátva. A film\_személy kapcsolótábla segítségével kötjük az egyes



személyeket az adott filmekhez, és itt az is meg van adva, hogy milyen poszton dolgoztak azon a filmen, színészként vagy rendezőként.

A film táblában tároljuk a filmeket. Minden filmnek van egyedi azonosítója. Valamint meg van adva a címük, leírásuk, hosszuk, eredeti nyelvük, egy plakát kép elérési útvonal és egy YouTube link, ami a film előzetesére mutat. Ezeken kívül még itt meg van adva a műfajának azonosítója.

A terem táblában az egyes termek sor és oszlop száma van megadva és minden teremnek van egy egyedi neve.

A vetítés táblában tároljuk a létrehozott vetítéseket. A saját azonosítóján kívül itt meg van adva a játszott film és a terem azonosítója. Meg kell adni a vetítés napját, kezdési idejét, azt, hogy szinkronos vagy feliratos, a jegy árát, valamint azt, hogy mikor volt a film premierje, végül pedig, hogy a vetítés már publikus-e. Egy vetítésre akkor lehet foglalni, ha már az admin publikusra állította a vetítést.

A foglalással rendelkező nézőknek a nevét és az email címét tároljuk, melyhez egy ID-t párosítunk.

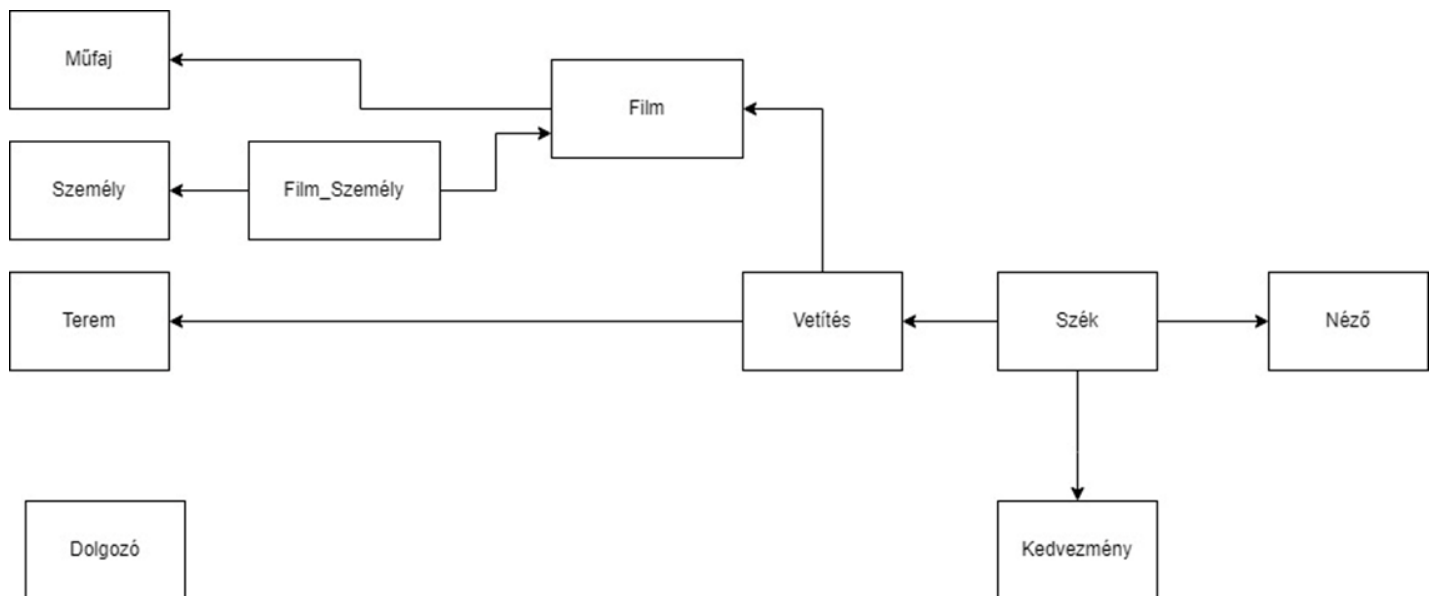
A foglalások miatt tároljuk az egyes székeket is. Itt meg van adva a széket azonosító sor és oszlop szám, valamint a vetítés ID. A széknek van állapota, ami lehet szabad, foglalt vagy eladott. Foglalás vagy vásárlás esetén a néző azonosítóját hozzákapcsoljuk a székhez. Ha a néző valamilyen kedvezményes jegyet vásárol, diák vagy nyugdíjas, az is fel van itt tüntetve. Arra az esetre, ha a néző a foglalást nem fejezi be és ott hagyja az oldalt, beállítottunk egy elvesztési időpontot, amely elérésekor ez esetben a foglalás törlődik.

A kedvezményke táblában tárolódnak a diák, nyugdíja és egyéb kedvezményke, melyeket az admin felületen lehet bővíteni. Itt százalékban adjuk meg a kedvezmény mértékét, valamint egy rövid leírás tartozik még hozzá.

Végül pedig még tároljuk a dolgozókat, akiket ID-vel is azonosítunk. Meg kell adniuk a nevüket, egy email címet és egy jelszót, amelyekkel be tudnak jelentkezni a dolgozói felületekre.

Minden dolgozónak meg van adva a munka köre, ami meghatározza, hogy mely felületekhez van hozzáférés.

### 4.3.2 Kapcsolati ábra



13. ábra, Adatbázis kapcsolati ábra

## 4.4 Backend

### 4.4.1 Modellek és Controllerek

#### 4.4.1.1 Publikus felhasználói felületek

Szikora Melinda

Az adatbázisban lévő adatok felhasználásához és az oldalon való megjelenítéséhez a backend részen model és controller php fájlokat kell létrehozni. A modellben meghatározzuk magukat az adatokat, a controllerben pedig a megjelenés módja van megadva. A controllerben használni kívánt modellt, csatolni kell annak a tetején.

A projektmunka elején, csak egyszerű, egy táblás modelleket hoztunk létre. Majd idővel egyre több táblát kapcsoltunk össze és egyre bonyolultabb modelleket és controllereket írtunk meg. Az összekapcsolásokon kívül használtunk még szűréseket is.

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Mufaj extends Model
9  {
10     use HasFactory;
11     protected $primaryKey='mufajId';
12 }
13
```

14. ábra, Egyszerű műfaj modell

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8
9  class filmMufajController extends Controller
10 {
11     public function index()
12     {
13         return DB::table('films')
14             -> join('mufajs', 'films.mufajId', '=', 'mufajs.mufajId')
15             -> select(
16                 'films.cim',
17                 'films.filmLeiras',
18                 'films.hossz',
19                 'films.nyelv',
20                 'films.pozster',
21                 'films.youtubeLink',
22                 'mufajs.mufaj',
23             )
24
25             ->get();
26     }
27 }
28
29 }

```

15. ábra, Kezdetleges kontroller, film-műfaj

A következő a film adatoknak a modellje és ebben egyesével funkciókkal lettek megadva a hozzá kapcsolni kívánt egyszerű modellek. Itt meg kell adni azt a mezőt, amin keresztül tudnak ezek a táblák kapcsolódni.

```
1  <?php
2  namespace App\Models;
3
4  use Illuminate\Database\Eloquent\Factories\HasFactory;
5  use Illuminate\Database\Eloquent\Model;
6
7  class Film extends Model
8  {
9      use HasFactory;
10     protected $primaryKey='filmId';
11     protected $table='films';
12     protected $fillable = ['cim','filmLeiras','hossz','nyelv','poszter','youtubeLink','mufajId'];
13     public $timestamps=false;
14
15     public function terem()
16     {
17         return $this->belongsToMany(Terem::class,'vetites','filmId','teremId')->as('kapcsolat')
18             ->withPivot('vetitesId','szinkron','kezdesiIdo','vetitesNap','publikus');
19
20         return $this->belongsToMany(Terem::class,'vetites','filmId','teremId')->as('kapcsolat')
21             ->withPivot('vetitesId','szinkron','kezdesiIdo','vetitesNap');
22     }
23
24     public function személyek()
25     {
26         return $this->belongsToMany(Szemely::class, 'film_szemelies','filmId','szemelyId')->as('kapcsolat')
27             ->withPivot('poszt');
28     }
29
30     public function rendezok()
31     {
32         return $this->belongsToMany(Szemely::class, 'film_szemelies','filmId','szemelyId')
33             ->wherePivot('poszt','=','rendező')->as('kapcsolat')->withPivot('poszt');
34     }
35
36     public function szereplok()
37     {
38         return $this->belongsToMany(Szemely::class, 'film_szemelies','filmId','szemelyId')
39             ->wherePivot('poszt','=','színész')->as('kapcsolat')->withPivot('poszt');
40     }
41
42     public function mufaj()
43     {
44         return $this->hasOne(Mufaj::class, 'mufajId', 'mufajId');
45     }
46
47     public function vetites()
48     {
49         return $this->hasMany(Vetites::class, 'filmId', 'filmId');
50     }
51 }
```

16. ábra, Film modell

A controllerben pedig meg van adva, hogy a film modellben meghatározott adatokkal térjen vissza. A film a vetítéssel, műfajjal és a személyekkel együtt.

A létrehozott controllereket az api.php fájlban hívtuk meg, ahol ez után létrejön az az API végpont, melynek meghívásával a programban lekérhetővé válnak az adatok az adatbázisból, JSON fájl formájában.

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Film;
7  use App\Models\Vetites;
8
9
10 /* FILM - MŰFAJ - VETÍTÉS - SZEMÉLYEK */
11
12 class FilmMuVetSzeController extends Controller
13 {
14     public function index(Request $request)
15     {
16         $name = $request->query('q', '');
17
18         return Film::with(['vetites', 'mufaj', 'szemelyek'])
19             ->get();
20     }
21 }
22 }
```

17. ábra, Nyitó oldal fő kontrollere

#### **4.4.1.2 Admin Backend**

Váradi Ákos

Az api végpontok használatához létre kell hoznunk útvonalakat. Api végpontokon keresztül érjük el a controllereket ahol tudunk adatokat megjeleníteni, törölni, módosítani és feltölteni újakat. Attól függően, hogy mit szeretnénk végrehajtani, az útvonalnál meg kell mondani, hogy az api végpontot hogyan hivatkozzuk, melyik controllert szeretnénk használni és annak melyik metódusát.

A controllerekben mondjuk meg milyen műveletet (index = megjelenítés, store = feltöltés, update = módosítani, delete = törlés) szeretnénk végrehajtani. Ahoz, hogy elérjük az adatbázisban lévő adatokat, szükségünk van egy Modell-re, ami azt fogja meghatározni, hogy melyik táblából szeretnénk lekérni az adatokat.

#### **4.4.1.3 Pénztáros oldalak**

Szabó Anett

A fő cél az volt, hogy adatbázisból tudjunk adatokat megjeleníteni, vagy különféle kéréseket indítani (GET, POST, PUT, DELETE).

Éppen ezért, Api végpontokat hoztunk létre, amely kommunikál a szerver és a kliens között. Ez dolgozza fel a kéréseket.

A pénztároldalnál szintén modelleket hoztam létre, amiben tárolódnak az adatok. Ez kéri le az adatokat az adatbázisból.

```

class Film extends Model
{
    use HasFactory;
    protected $primaryKey='filmId';
    protected $table='films';
    protected $fillable = ['cim','filmLeiras','hossz','nyelv','poszter',
        'youtubeLink','mufajId'];
    public $timestamps=false;

    public function terem()
    {
        return $this->belongsToMany(Terem::class,'vetites','filmId','teremId')->as('kapcsolat')
            ->withPivot('vetitesId','szinkron','kezdesiIdo','vetitesNap','publikus');

        return $this->belongsToMany(Terem::class,'vetites','filmId','teremId')->as('kapcsolat')
            ->withPivot('vetitesId','szinkron','kezdesiIdo','vetitesNap');
    }
}

```

Itt látható egy többtáblás összekapcsolás, ahol a filmeket és a termeket kapcsolom a vetítés táblához.

A controller kommunikál a modellel és a route-val. Itt állítom elő az API végpontot.

```

use Illuminate\Http\Request;
use App\Models\Film;
class penztarFooldalController extends Controller
{
    public function index()
    {
        $filmek=Film::with('terem')->get();
        return $filmek;
    }
}

```

Az index metódusban visszaadom az összes filmet, amit később a frontend egy JSON formátumban kap vissza.



A controllerben létrejött Api végpontokat hívom meg, és a route kommunikál a kliens oldallal.

```
Route::get('/penztarFoOldal', [penztarFoOldalController::class, 'index']);
```

A controllernek az index metódusát hívom s egy általam választott névvel tudok majd hivatkozni frontend oldalon az Api végpontomra.

#### 4.4.2 API végpontok

Api végpont	Leírása	Oldal
Get /api/ személy	Visszaadja a személyek adatait	Admin személyek oldal
Delete /api/ személy	Törli az adott személyt adatait	
Post /api/ személy	Feltölti az új személy adatait	
Put /api/ személy	Módosítja egy adott személy adatait	
Get /api/filmSzemelyMufaj	Visszaadja a filmeket és a hozzá tartozó személyeket, postjukat is film műfaját	Admin filmek oldal
Get /api/mufaj	Visszaadja a műfajok adatait	
Get /api/szemely	Visszaadja a személyek adatait	

Delete /api/film	Törli az adott filmet adatait és filmszemélye táblában a film és a személy kapcsolatot és az ehhez tartozó posztot.	
Get /api/terem	Visszaadja a terem adatait	Admin termék oldal
Put /api/terem	Módosítja egy adott terem adatait	
Delet /api/terem	Törli az adott terem adatait	
Post /api/terem	Feltölti az új terem adatait	
Get /api/filmVetites	Visszaadja a vetítéseket és a hozzá tartozó film cím és teremi azonosító adatait	Admin vetítések oldal
Get /api/film	Visszaadja a filmek adatait	
Get /api/terem	Visszaadja a termék adatait	
Put /api/vetites	Módosítja egy adott vetítés adatait	
Delet /api/vetites	Törli az adott vetítés adatait	
Post /api/vetites	Feltölti az új vetítés adatait	
Post /api/szek	Legenerálja az adott vetítés terméhez tartozó székeket.	
Get /api/kedvezmeny	Visszaadja a kedvezmények adatait	Admin kedvezmény oldal
Put /api/kedvezmeny	Módosítja egy adott kedvezmény adatait	

Delet /api/kedvezmeny	Törli az adott kedvezmény adatait	
Post /api/kedvezmeny	Feltölti az új kedvezmény adatait	
Get /api/dolgozo	Visszaadja a dolgozók adatait	Admin dolgozó oldal
Put /api/dolgozo	Módosítja egy adott dolgozó adatait	
Delet /api/dolgozo	Törli az adott dolgozó adatait	
Get/api/filmMufaj	Film és műfaj modellek összekapcsolva	Nyitó oldal  Szikora Melinda
Get/api/filmVetites	Film és vetítés modellek összekapcsolva	
Get/api/vetitesFilmMufaj	Vetítés, film és műfaj modellek összekapcsolva	
Get/api/komplettFilm	Film, műfaj, vetítés és személyek modellek összekapcsolva.  A nyitó oldal végleges, fő api-ja	

Get/api/filmPlakat	<p>Csak a plakátokat adja vissza a film modellből.</p> <p>A galéria megjelenítését biztosítja</p>	
--------------------	---	--

Get/api/penztarFoOldal	3 összekötött tábla adatait adja vissza (Vetítés, terem, film)	<p>Pénztár főoldal</p> <p>Szabó Anett</p>
Get/api/nezo	Visszaadja a néző adatát	<p>Pénztár foglalási oldal</p> <p>Szabó Anett</p>
Post/api/nezo	Feltölti a néző adatát	

### 4.4.2.1 Publikus felhasználói felületek

Szikora Melinda

Annak érdekében, hogy a frontend elérje az adatbázis adatait, és tudjon dolgozni azokkal, API végpontokat hoztunk létre a backend részen. Ezek a végpontok tudnak lekérni, feltölteni az adatbázisba, valamint lehet törölni is és módosítani is az ott lévő adatokat. Tehát a kliens és a szerver kérései és válaszai ezeken a pontokon mennek keresztül.

```
Route::get('/komplettFilm', [filMuVetSzeController::class, 'index']);
```

18. ábra, Nyitó oldal fő kontrollerének route-ja

Route létrehozásakor meg kell adni azt a címet amire a későbbiekben a frontend fog hivatkozni, majd meg kell adni a használni kívánt controller osztályt és annak metódusát.

A projekt létrehozása közben sok végpontot hoztunk létre, van amelyik egy egész film megjelenítését szolgálja, és van, ami csak a plakátokkal tér vissza, a galéria megjelenítéséhez.

## 4.5 Frontend

### 4.5.1 Publikus felhasználói felületek

Szikora Melinda

A frontend feladata a megjelenítés. HTML, CSS és JavaScript fájlok használatával alakítjuk ki az oldalak elemeinek az elrendezését és a látványt. A megjelenített adatok az adatbázisból API végpontokon keresztül érkeznek. Ezeket Ajax hívások segítségével töltjük be tömbökbe, így azok eltárolódnak a frontend részen is. Ezekbe a hívásokba callback metódusokat is használunk, melyek általában a megjelenítést végzik. Így az adat lekérés után rögtön meg is jelennek a felhasználói felületen.

Az oldalak létrehozásához blade-et használunk. Az oldalak megegyező részei ki lettek szervezve külön fájlokba, ilyenek a fejléc és a lábléc. Ezek a részek be vannak hivatkozva a különböző oldalakon, amiken ezeknek meg kell jelenniük. Így a megegyező kódrészek nincsenek minden oldalon ismételve.

CSS fájlokban állítottuk a háttereket, a színeket, és az egyes elemek elrendezését. Az elrendezésekhez főleg két módszert használtunk, gridet és flexboxot. Gridet a bonyolultabb, összetettebb több elemes részekben, mivel a grid több dimenziós rendezésre alkalmas. A flex az egyszerűbb, egy soros vagy egy oszlopos rendezéskere megfelelő, mint például a vetítési időpontok egy sorba rendezése.

Azokban az esetekben, ahol nem az általunk létrehozott CSS fájlokkal alakítottuk az oldalak kinézetét, ott Bootstrap5 osztályok segítségével formáztuk a tartalmak megjelenését és elrendezését.

# Osztályok

Szikora Melinda

Az objektum-orientált programozás elméletét követve, az oldalainkon objektumokat hozunk létre. Az objektumok egységbe zárják az adatokat és a hozzájuk tartozó műveleteket. Az osztály a terv, ami alapján a példány/objektum létrejön.

Osztályt hoztam létre a filmeknek, így a nyitó oldalon minden film elem ugyan annak az osztálynak a példánya. Így mindegyik film objektum ugyan azokkal a tulajdonságokkal rendelkezik, valamint dinamikusan lehet ezeket létrehozni az adatbázisból lekért adatokkal.

A film oldal egy hasonló osztály példánya, de ez esetben csak egy film jön létre, több adattal. Ehhez hasonló a foglalás oldalon megjelenő film elem.

- Film osztály
  - konstruktor(fElem, fAdat) : Feladata, hogy beállítsa a film objektum paramétereit. A meghatározott elemek adat feltöltését a setAdatok metódus végzi.
  - setAdatok(ertek) : A konstruktorban átadott film adatokat betölti a példány megfelelő HTML elemébe.
  - filmTovabbToltTrigger(): Eseményt hoz létre, mely a film címre való kattintáskor, az adott film adataival tér vissza.
  - idopontTovabbToltTrigger(): Eseményt hoz létre, mely egy időpontra való kattintáskor az adott időt adtja vissza, ennek a foglalás oldalon van szerepe

- FilmOldal osztály

- konstruktor(fAdat) : Feladata, hogy beállítsa a film oldal paramétereit. Meghatározza az oldal HTML elemeit, amiknek az átadott adattal való feltöltését a setAdatok metódus végzi.
- setAdatok(ertek) : A konstruktorban átadott film adatot betölti a példány megfelelő elemeibe.
- idopontTovabbToltTrigger(): Eseményt hoz létre, mely egy időpontra való kattintáskor az adott időt adtja vissza, ennek a foglalás oldalon van szerepe

- FilmFoglalas osztály

- konstruktor(fAdat) : Feladata, hogy beállítsa a film objektum paramétereit. Meghatározza a HTML elemeket, amiknek az átadott adattal való feltöltését a setAdatok metódus végzi.
- setAdatok(ertek) : A konstruktorban átadott film adatot betölti a példány megfelelő elemeibe.



```

class Film {

  constructor(fElem, fAdat, datum) {

    this.fElem = fElem;
    this.fAdat = fAdat;
    this.datum=datum;

    this.fCim = this.fElem.children("div:last").children(".filmCim");
    this.tovabb = this.fElem.find(".tovabb");

    this.leiras = this.fElem.children("div:last").children(".leiras");
    this.plakat = this.fElem.children("div:first").children("img");
    this.mufaj = this.fElem.children("div:last").children(".mufaj");

    this.idopontSzulo = this.fElem.children("div:last").children("#idopontSzulo");

    this.setAdatok(this.fAdat);

    this.fCim.on("click", () => {

      this.filmTovabbToltTrigger();

    });

    for (let i = 0; i < this.idopontok.length; i++) {

      $(this.idopontok[i]).on("click", () => {

        this.kivalasztottIdopont=$(this.idopontok[i]).attr("id");
        this.idopontTovabbToltTrigger();

      });

    }

  }
}

```

19. ábra, Film osztály konstruktora

A konstruktor feladata, hogy létrehozáskor beállítsa a példány értékeit. Így a film osztály konstruktorában átadásra fognak kerülni HTML elemek, melyhez a film objektum adatai hozzá lesznek adva. Valamint meg van adva maga az adat, amit be kell tölteni a film objektum meghatározott elemébe. Ezeket a meghívott set metódus tölt fel.

```

setAdatok(ertek) {
    this.fCim.html(ertek.cim);
    this.leiras.html(ertek.filmLeiras);
    this.plakat.attr("src", ertek.pozster);

    for (const key in ertek.mufaj) {
        this.mufaj.html(ertek.mufaj.mufaj);
    }

    for (const key in ertek.vetites) {
        if(this.datum == ertek.vetites[key].vetitesNap) {
            this.idopontSzulo.append("<a class='idopont' id=" + ertek.vetites[key].vetitesId + " href='/foglalasOldal'" + ertek.vetites[key].kezdesiIdo + "</a>");
        }
    }

    this.idopontok=this.idopontSzulo.find(".idopont");
}

```

```

filmTovabbToltTrigger() {
    let filmEsemeny = new CustomEvent("filmTovabbTolt", {detail: this.fAdat});
    window.dispatchEvent(filmEsemeny, 1);

    console.log("filmTovabbToltTrigger");
}

idopontTovabbToltTrigger() {
    let idopontEsemeny = new CustomEvent("idopontTovabbToltTrigger", {detail: this});
    window.dispatchEvent(idopontEsemeny);
}

```

20. ábra, Film osztály set metódusa és eseményei

Mivel a filmhez tartozó műfaj és vetítések a modellben lettek hozzá kapcsolva, így ezeket még ciklussal kell bejárni, hogy elérjük a szükséges adatot. Vetítési időpont pedig több is tartozik egy filmhez, így azokat a cikluson belül kell hozzá adni. Valamint a napok szerinti rendezés miatt, a vetítésben még egy feltétellel ellenőrizzük ezeket.

A film osztályban van még egy létrehozott esemény, ami a filmre való kattintáskor az adott film adatait fogja visszaadni. Ennek a film oldal betöltésében lesz szerepe.

Az adatok tömbökbe való betöltését és azoknak a megfelelő megjelenítését ajax hívásokkal valósítottuk meg. Létrehozott ajax osztály metódusai tudják az API-ból jövő adatot tömbökbe tölteni, vagy módosított adatokat a tömbből az adatbázisba küldeni.

```
getAdat(apivegpont,tomb,myCallback){
  tomb.splice(0,tomb.length);
  $.ajax({
    url:apivegpont,
    type:"GET",
    headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')},
    success:function(result){
      result.forEach(element => {
        tomb.push(element)
      });
      myCallback();
    },
    error: function(result) {
      alert("Adatbetöltés sikertelen!");
    }
  });
}
```

21. ábra, Ajax osztály getAdat metódusa

## Nyitó szkript

Szikora Melinda

A film objektumok létrehozásához a nyitó oldal fő szkriptjében meghívtam az ajax osztály get metódusát, ami az API-n keresztül eléri az adatbázis adatait, és egy tömbbe tölti, valamint a callback metódust kellett még megírni, ami a film megjelenítését végzi.

```

function filmMegjelenites() {

    let itemDatume = localStorage.getItem("itemdatumertek");
    szuloelem.show();
    sablonElem.show();

    komplettFilmTomb.forEach(function (elem, index) {

        elem.vetites.forEach(function (ka, i) {

            if (itemDatume == ka.vetitesNap && ka.publikus == 1 ) {

                const ujElem = sablonElem.clone().appendTo(szuloelem);
                const ujFilm = new Film(ujElem, komplettFilmTomb[index],itemDatume);

            }

        });

    });

    sablonElem.hide();
}

```

22. ábra, Nyitó oldal fő szkript, filmMegjelenites

Ahhoz, hogy minden film megjelenjen a megjelenítés metódusban, meg van adva konstansként az a HTML elem, amihez minden film objektum hozzá lesz adva, valamint az a sablon elem, ami alapján a film osztályban meg lettek határozva azok a tag-ek, amikbe az adatok fognak kerülni. Minden film példány ennek a másolatát tölti majd fel a saját adataival, és ezek adódnak hozzá a szülő elemhez. A filmek a vetítéseik alapján naponként szűrve vannak.

```

<!-- Csempe elemek -->

<div id="musorListaCsempeSzulo">

  <div class="csempeSablon row">

    <div class="divCsempePlakat col-md-3">
      <img class="csempePlakat" src="" alt="plakát">
    </div>

    <div class="csempeLeiras col-md-9" >

      <a href="/filmOldal" class="filmCim"> </a>

      <p class="mufaj"> </p>
      <p class="leiras"> </p>

      <div id="idopontSzulo">

        <p> Időpontok: </p>

      </div>

    </div>

  </div>

</div>

```

23. ábra, Film elem sablon

Egy `forEach` ciklussal megyek végig azon a tömbön, amibe a szükséges adatok vannak. A cikluson belül létrejön az üres, új elem, amit ezután egyenlővé teszek egy új film objektummal. A tömb adott adatai pedig bekerülnek ennek az objektumnak az osztályban meghatározott helyeikre.

```

const myAjax = new MyAjax();

/* API végpontok: */

//const vfmApi = "/api/vetitesFilmMufaj";
const komplettFilmApi = "/api/komplettFilm";
const plakatiApi = "/api/film";

/* Adat tömbök: */

const komplettFilmTomb = [];
const plakatiTomb = [];

/* Adatok Lekérése, tárolása és megjelenítések: */

myAjax.getAdat(komplettFilmApi, komplettFilmTomb, filmekMegjelenites); // FILMEK
myAjax.getAdat(plakatiApi, plakatiTomb, galeriaMegjelenites); // GALÉRIA

```

24. ábra, Nyitó oldal fő szkript, filmek megjelenítése

Ezek után a 4-es ábrán látható módon megjelennek a filmek a nyitó oldalon. Minden filmnek a címe és az időpontjai kattinthatóak, ezek további oldalakra viszik tovább a nézőt. Ennek működését a localStorage segítségével oldottam meg. A webes alkalmazások a felhasználó böngészőjében, helyben tudnak adatokat tárolni ezen a módon.

Ha a néző a film címére vagy egy időpontra kattint, akkor a film osztályban létrehozott események, annak az adott filmnek az adatait vissza fogja adni. Ezt az eseményt a fő szkriptben elkapva, ezeket az adatokat betöltöm a böngésző localStorage-ába.

```

/* Oldal váltás, film tovább adása */

$(window).on("filmTovabbTolt", (filmEsemeny) => {

    localStorage.setItem("filmKulcs", JSON.stringify(filmEsemeny.detail));
});

```

25. ábra, Nyitó oldal fő szkript, localStorage.setItem

Így a kattintott film adatai bekerülnek helyben a néző böngészőjének a tárhelyébe, ahonnan a következő oldalon ezt az adatot le lehet kérni és be lehet tölteni az oldalra.

```

/* Oldal váltás, film betöltése */

let filmAdat = [];

filmAdat = JSON.parse(localStorage.getItem("filmKulcs"));
console.log(filmAdat);

```

26. ábra, Film oldal fő szkript, localStorage.getItem

A JSON objektumot a feltöltéskor át kell alakítani string formátumúvá, mivel a webböngésző így tudja eltárolni. Letöltéskor pedig vissza kell alakítani objektummá. Ezek a stringify és a parse funkciók.

Miután a kattintott film adatai bekerültek egy tömbbe, onnan már a fő oldalon használt módhoz hasonlóan, meg lehet jeleníteni a filmet. A film oldal egy olyan objektum, mint a nyitó oldalon a kis film elemek. Valamint a foglalás oldalon található film is egy ilyen példány.

```
const filmOldal = new FilmOldal(filmAdat);
```

27. ábra, Film oldal fő szkript, FilmOldal létrehozása

A galéria létrehozásához is osztályt hoztam létre, amiben az erre a célra létrehozott apin keresztül csak a plakátok kerülnek majd átadásra, amit betölt az oldal megfelelő elemébe. Majd a fő szkriptben van létrehozva ez a galéria megjelenítés, és a hozzá tartozó lapozás funkciók.

```
class GaleriaPlakat {  
  
    constructor(gElem, gAdat) {  
  
        this.gElem = gElem;  
        this.gAdat = gAdat;  
  
        this.plakat = this.gElem.children("img");  
  
        this.setAdatok(this.gAdat);  
    }  
  
    setAdatok(ertek) {  
  
        this.plakat.attr("src", ertek.pozster);  
    }  
  
}
```

28. ábra, Galéria osztály



```

let ujGaleriaPlakat=[];

function galeriaMegjelenites() {

    const galeriaSzulo = $("#carousel");
    let sablonElem = $("#korhintaGaleria");

    const ujElem = sablonElem.clone().appendTo(galeriaSzulo);
    ujGaleriaPlakat = new GaleriaPlakat(ujElem, plakatoTomb[0]);

    sablonElem.remove();
}

let index = 0;

function jobbra() {

    index++;

    if(index > plakatoTomb.length-1) {

        index = 0;
    }

    ujGaleriaPlakat.setAdatok(plakatoTomb[index]);
}

function balra() {

    index--;

    if(index < 0) {

        index = plakatoTomb.length-1;
    }

    ujGaleriaPlakat.setAdatok(plakatoTomb[index]);
}

$("#bal").on("click", ()=> {balra()} );
$("#jobb").on("click", ()=> {jobbra()} );

```

29. ábra, Galéria megjelenítés és lapozás

## 4.5.2 Admin oldalak

Váradi Ákos

### Admin osztályok

Váradi Ákos

Mindegyik admin oldalnak külön osztályt hoztam létre, így az egyes oldalak mindig a számukra megfelelő osztálynak a példányát használják. Ezáltal az egyes oldalak objektumai ugyanazokkal a tulajdonságokkal rendelkeznek és ezért úgy alakítottam ki, hogy dinamikusan lehessen őket megjeleníteni az adatbázisból lekért adatok segítségével.

Az osztályokban a constructor feladata, hogy beállítsa a példányok értékét mikor azok létrejönnek. A construtornak átadom paraméterként azt, hogy melyik szülő elemhez szeretném hozzáadni az adatokat és átadom magát az adatot, amit majd hozzáadok az elemhez. Itt meghívom a `tablaSorGeneralas` metódust. Ami létrehozza az elemeket, betölti az adatokat és az átadott szülőelem belerakja mindezt. Az adatok tömb formájában kerültek átadásra ezért ahhoz, hogy végig tudjunk menni az elemein egy `for` ciklussal be kell járnunk az adatokat és ezen belül hozzuk létre az elemet és adjuk hozzá az adatokat. A szülőelemhez hozzáadunk még egy törlés és egy módosítás gombot. Aminek a klikk eseményét megvizsgáljuk, hogy ha valaki rákattint, akkor meghívódik egy törlés vagy módosítás metódus. A metódusokban létrehozunk vagy törlést, vagy egy módosítás eseményt. Amit majd a fő szkriptben fogunk elkapni.

Az osztálynak átadott adatok egy JSON formátumú objektumból nyerünk ki és ezeket tároljuk el egy tömbben.

## Admin szkript

Váradi Ákos

Mindegyik oldalnak lett készítve egy fő szkript. Ahol az oldal tetején példányosítottam az ajax osztályomat. Az ajax osztályban metódusok lettek létrehozva, amik különféle kéréseket (GET, POST, DELETE, PUT) kéréseket indítanak a szerver felé az API végpontokon keresztül.

Létre lett hozva megjelenítés függvény, ami az adatok megjelenítéséért felel. A függvényen belül meg lett vizsgálva az api végpont. Ez azért volt szükséges, hogy a kereső mezőbe beírt karakterek miatt változik az api végpont, amit kezelni kellett, hogy megjelenítéskor jó adatok látszódnak. A JSON fájl adatait egy tömbben eltároltuk, ezen a tömbön végig kell mennünk, hogy a korábban bemutatott osztálynak átadhassuk a szülő elemet és az adatokat.

Lére lettek hozva különféle metódusok, amik vagy egy gomb lenyomását vizsgálják, vagy, hogy a kereső mezőbe milyen karakterek lettek begépelve, vagy azt, hogy a rendezésnél melyik szempont lett kiválasztva. Valamint létre lett hozva két esemény figyelő a törlés és a módosítás eseménynek, ami a korábban említett osztályban fog végbe menni.

Ahhoz, hogy tudjunk adatokat megjeleníteni, törölni, módosítani, és felvinni ezekhez mind szükség van egy Ajax metódus hívásra, ami az api végpontokon keresztül kérést indít a szervernek és a szerver egy JSON fájlban adja vissza az adatokat, amit az Ajax fel tud dolgozni.

### 4.5.3 Pénztáros oldalak

Szabó Anett

## Osztályok

Az objektum orientált programozás alap elveit követve osztályokat hoztam létre. Mindegyik osztálynak megvan a maga feladata.

- Filmek osztály

```
class Filmek{
    constructor(elem, adat){
        this.adat=adat;
        this.elem=elem;
        this.filmcim=this.elem.find(".filmCim");
        this.filmIdopont=this.elem.find(".filmidopont");
        this.setAdatok(this.adat);
    }

    setAdatok(ertekek){
        this.filmcim.append(ertekek.cim);
    }
}
```

A filmek osztály felelős a filmcímek megjelenítéséért. A konstruktorban inicializálom a paraméter adatait és definiálom a HTML elemeket. A setAdatok a konstruktorban átadott filmcím adatot betölti a példány megfelelő elemeibe.

- Időpontok osztály

```
class Idopontok{  
  constructor(elem, adat,nyelv){  
    this.adat=adat;  
    this.elem=elem;  
    this.nyelv=nyelv;  
    this.foglalj=this.elem.children(".foglalj");  
    this.terem=this.elem.children(".terem");  
    this.idopont=this.elem.children(".idopont");  
    this.szinkron=this.elem.children(".szinkron");  
    this.setAdatok(this.adat,this.nyelv);  
  
    this.foglalj.on("click",()=>{  
      this.idopontbetoltTrigger();  
      this.nyelvKiir();  
    });  
  
    this.vasarlas1.on("click",()=>{  
      this.idopontbetoltTrigger();  
    });  
  };  
  
  setAdatok(ertekek,nyelv){  
    this.terem.html(ertekek.teremNev);  
    this.idopont.html(ertekek.kapcsolat.kezdesiIdo);  
    this.szinkron.html(nyelv);  
  }  
}
```

```
idopontbetoltTrigger(){  
  let esemeny=new CustomEvent("idopontBetolt",{  
    detail:this.adat,  
  });  
  window.dispatchEvent(esemeny);  
}  
  
nyelvKiir(){  
  let esemeny=new CustomEvent("nyelvKiir",{  
    detail:this.nyelv,  
  });  
  window.dispatchEvent(esemeny);  
}
```

Ez az osztály felelős a terem, időpont és a szinkron megjelenítésért.

Az időpontok osztály konstruktorában inicializálom a paraméter adatait és definiálom a HTML elemeket. Itt kezelem még a kattintás eseményeket is.

A setAdatok a konstruktorban átadott terem, időpont és szinkron adatait betölti a példány megfelelő elemeibe.

Ezen felül létrehoztam két függvényt, amikben vannak a kattintás események.

Az előre meghatározott HTML elemre kattintva lefutnak ezek a metódusok, de csak abban az esetben, ha szkriptben megtörtént a példányosítás.

Mindkét metódus átadja az eseményt az ablaknak(window). Ami úgyszintén a szkriptben fogja “elkapni”, vagyis figyelni, hogy megtörtént-e a kattintás.

- Foglалás osztály

```
class Foglалas{  
  
    constructor(elem, adat){  
        this.elem=elem;  
        this.adat=adat;  
        this.adatGeneralas();  
    }  
  
    adatGeneralas(){  
        for (const key in this.adat) {  
            if(key!="created_at" && key!="updated_at"){  
                this.elem.append("<td>"+this.adat[key]+"</td>");  
            }  
        }  
    }  
}
```

Ebben az osztályban is inicializálom a paraméter adatait.

Ezen felül meghívom az adatGeneralas() metódust, amiben megmondom, hogy hova rakja be majd a példányosítás során az adatokat (mikor létrehozok egy objektumot).

Az adatGeneralas() metódusban végig megyek a konstruktor paraméterében lévő adaton, lekezelem azt, hogy ha az adat kulcsa nem egyenlő a 2 szöveggel, akkor csak abban az esetben töltse bele az adat kulcsait(elemeit).

## Pénztáros szkriptek

A 2 oldalamhoz 2 szkriptet hoztam létre. Az egyik a főoldal, a másik a foglalásioldal működéséért felel.

Mindkettő szkriptem elején eltároltam az Api végpontokat, amikkel dolgozni fogok.

Eltárolom a tömböket, változókat és meghívom a metódusokat, amiket létrehoztam.

Példányosítom az Ajax osztályt, ennek segítségével tudom kezelni az Api végpontokat és megfelelő kérést indítani.

Ezen felül mindkettőben kezelem a kattintásokat.

- Pénztár főoldal szkript

Ebben a szkriptben jelenítem meg a napokat és a hozzájuk tartozó dátumokat.

Figyelem a kattintás eseményeket.

És a legfontosabb, hogy betöltöm a filmcímeket és a hozzájuk tartozó időpontokat.

```

function idopontKiir(szuloelem, idopontindex, terem) {
    const sablonElem = $(".secSablon .filmidopontSablon");
    sablonElem.show();
    let ujElem = sablonElem.clone().appendTo(szuloelem);
    const ujIdopontok = new Idopontok(ujElem, terem,
        vetitesekTomb[idopontindex].nyelv);
    sablonElem.hide();
}

function filmetkiir() {
    let filmId = 0;
    let ujFilm;
    let itemDatum = localStorage.getItem("itemdatumertek");
    const szuloelem = $(".vetitesek");
    const sablonElem = $(".secSablon .filmmezok");
    szuloelem.empty();
    sablonElem.show();
    vetitesekTomb.forEach(function (elem, index) {
        elem.terem.forEach(function (ka, i) {
            if (itemDatum == ka.kapcsolat.vetitesNap) {
                if (ka.kapcsolat.publikus == 1) {
                    $(".filmmezok").css("border", "inline");
                    $(".foglalj").css("display", "inline");
                    if (filmId != elem.filmId) {
                        let ujElem = sablonElem.clone().appendTo(szuloelem);
                        ujFilm = new Filmek(ujElem, elem);
                    }
                    idopontKiir(ujFilm.filmIdopont, index, ka);
                    filmId = elem.filmId;
                }
            }
        });
    });
    sablonElem.hide();
}

```

Az idopontKiir() metódusban:

Definiálom egy változóban, amit a blade oldalamon létrehoztam a sablon elemet.

Egy új változóban lemásolom a sablon elemet és a belerakom a szülő elembe, amit a paraméterben találok és a metódus meghívásakor fogom megmondani mi a szülőelem.

Példányosítom az Idopontok osztályt, így már az osztály összes metódusát elértem.



A `filmetkiir()` metódusban:

Eltárolom egy változóban a kattintott napot a `localStorage`-ból kiolvasva.

Két változóban eltárolom a szülő és a sablon elemet.

Mivel az API végpont egy tömbös JSON formátumot ad vissza, így két ciklusban végig megyek a két tömbön.

Ezután összehasonlítom, hogy a kattintott dátum az adatbázisban lévő nappal megegyezik-e és azt is, hogy publikus-e.

Amennyiben igen, fölül írom a HTML elemeken lévő alap formázást.

Majd pedig megvizsgálom, hogy a kezdetben létrehozott változó nem egyenlő az adatbázisban lévő `filmId`-val.

Ha ez is igaz akkor egy új változóban lemásolom a sablon elemet és belerakom a szülő elembe.

Példányosítom a filmek osztályt.

Ezt követően meghívom az `időpontKiir()` metódust, ahol megmondom, hogy a filmek osztály `filmId` pont adattagja a szülő elem és megmondom hányadik filmhez rakja be az időpontot.

Végül pedig az adatbázisban lévő `filmId` egyenlővé teszem a metódusban létrehozott változóval, így tudok egy filmhez több időpontot megjeleníteni.

Valamint új filmeket létrehozni időpontokkal.

- Foglалás oldal szkript

Itt jelenítem meg a nézőket. Keresek, rendezek közöttük.

Kezelem a vásárlást.

Viszek fel új nézőt és módosítom a székeket.

```
function foglalasKiir() {  
    szuloelem = $(".tablafoglaSzulo");  
    szuloelem.children().remove();  
    foglalasTomb.forEach(function (elem) {  
        let tablaElem=$("<tr></tr>").appendTo(szuloelem);  
        const ujFoglалas = new Foglалas(tablaElem, elem);  
    });  
}
```

Ez a metóduѕ jeleníti meg a nézőket. Eltárolom egy változóban a szülő elemet.Ezt követően törlöm a szülőelem gyereket.Végig megyek a foglalás tömbön és létrehozom a tábla elemet és hozzá rakom a szülő elemhez.Végül pedig példányosítom a Foglалas osztályt.

```
myAjax.getAdat(fajlnevFoglалas, foglalasTomb, foglalasKiir);
```

Az ajax osztály getAdat metóduѕát hívom meg, ahol átadok egy api végpontot, egy tömböt és a callback függvényt. Ez a hívás az oldal betöltődésekor fut le és megjeleníti a nézőket.

## 4.6 Reszponzivitás

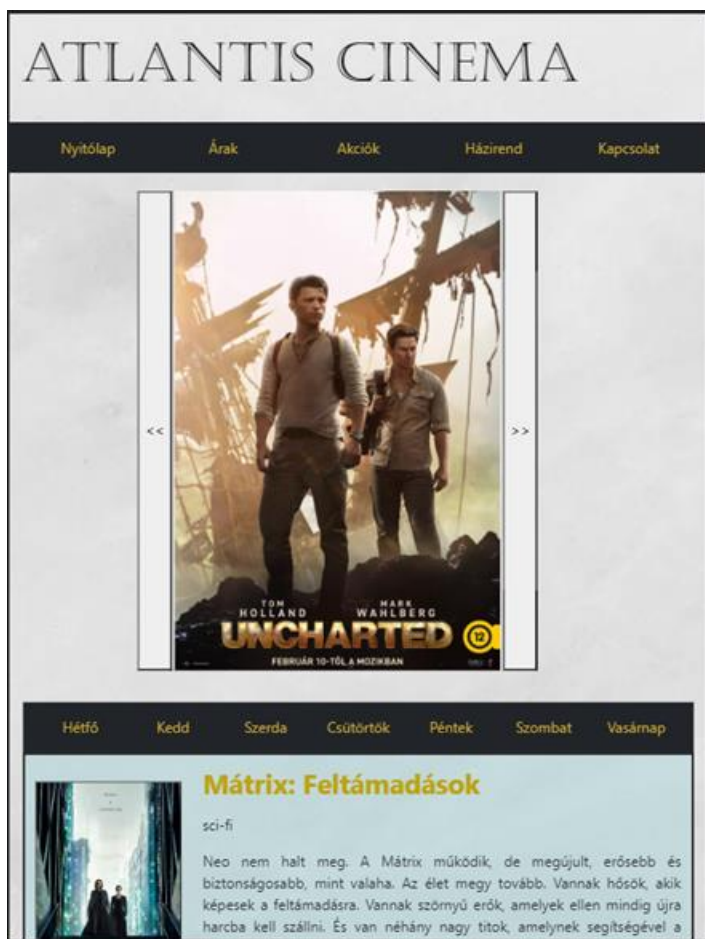
### 4.6.1 Publikus felhasználói felületek

Szikora Melinda

A felhasználói oldalakon a részponzivitás nagyon fontos volt, mivel a nézők a legkülönbélebb monitorokon és telefon kijelzőkön nézhetik az oldalt. Mindegyiken úgy kell megjelennie, hogy a képernyő méretéhez alkalmazkodva a lehető legtöbbet hozza ki belőle.

Az oldal teljes nagyságú alakja a 2.2-es fejezetben látható. A kisebb képernyőkön a nem szükséges elemek eltűnnek, a fő oldalon ilyenek a kép galéria és a hírek. Így a kis kijelző nem érződik túl zsúfoltnak, és nem túl apróak a megjelent elemek. Valamint, ha a néző telefonon nyitja meg az oldalt, akkor nagyobb az esélye annak, hogy nem wifi-re van csatlakozva az eszköz és így nem használ feleslegesen adatforgalmat a képek betöltése.

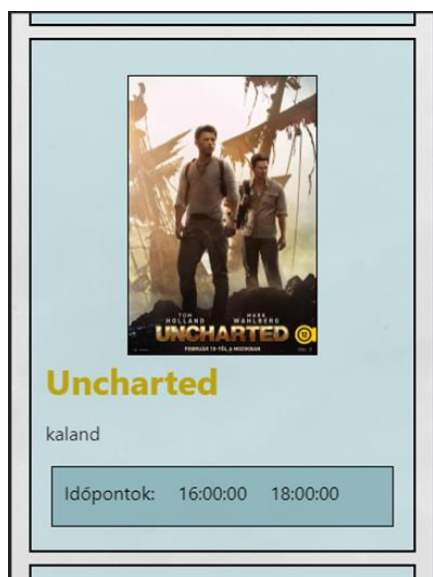
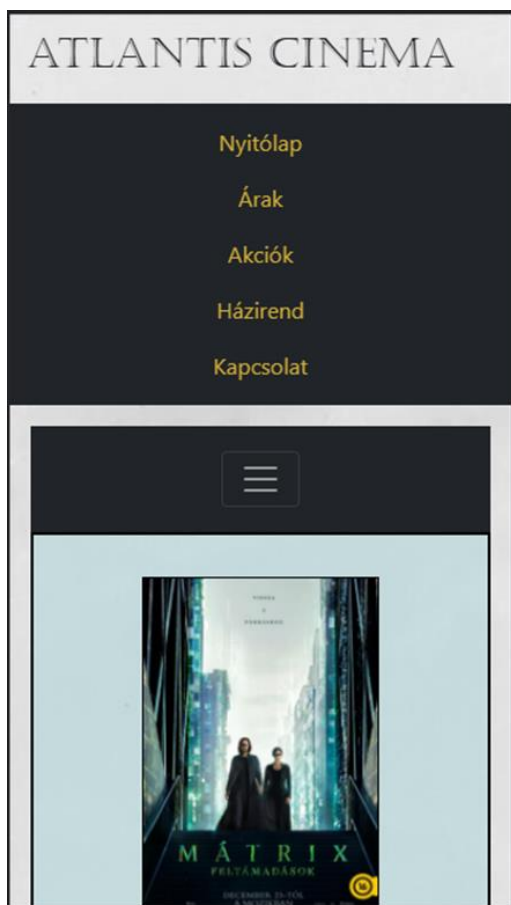
A részponzivitáshoz css-ben grid és flex elrendezéseket használtunk az oldalainkon, valamint elemek elrejtését. Használtunk bootstrap-et sok helyen, aminek előre elkészített osztályait használtuk. Bootstrap5 keretrendszer, részponzív oldalak kialakításához készült eszköztár.



30. ábra, Nyitó oldal, tablet nézet

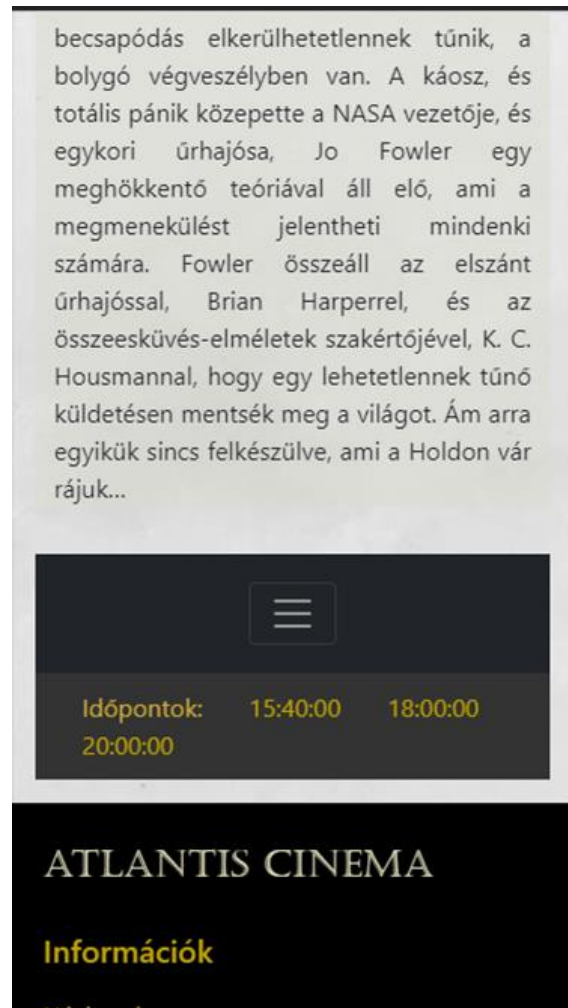
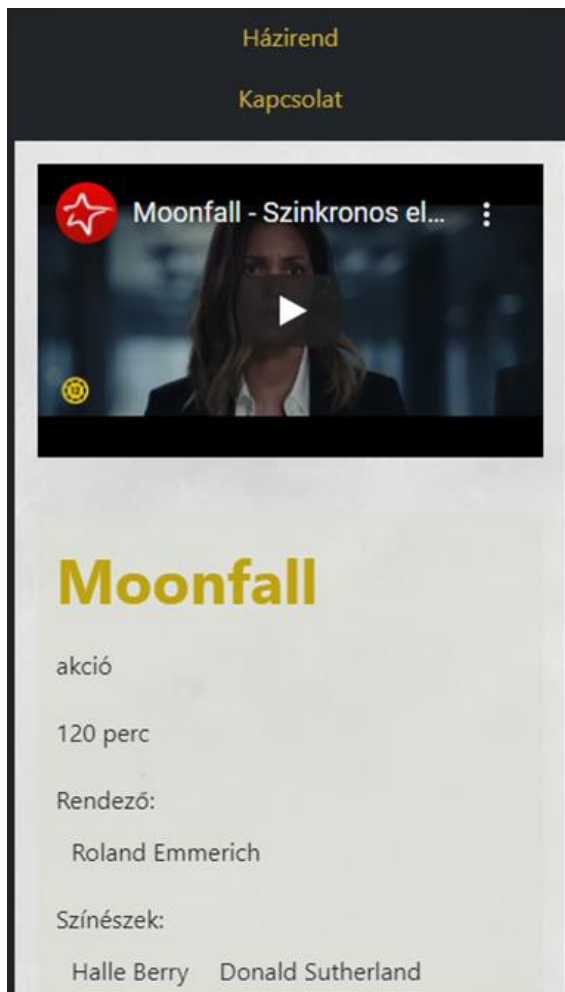


Mobil nézetben a főmenü pontjai egymás alá csúsznak, a napok pedig egy menüben le nyithatóvá válnak. A filmek leírása ekkora képernyőn a nyitó oldalon nem jelennek meg, csak tovább lépve a film oldalon olvasható így el. A láblécben lévő menüpontok is egymás alá kerülnek.



31. ábra, Nyitó oldal, mobil nézet

A film oldalon mobil nézetben a poszter nem jelenik meg, a napok pedig itt is lenyitható menübe kerül.



32. ábra, Film oldal, mobil nézet

A foglalás oldalon se látszanak kis képernyőn a posztterek, így csak a lényeges információk és az adatbeviteli mezők láthatóak.

## **4.6.2 Admin felületek**

Váradi Ákos

Az admin felület kialakításánál a teljes reszponzivitás nem volt elsődleges szempont. Mivel úgy terveztük, hogy mobilos és tabletes méretben az adatok felvitele és megjelenítése nehézséget okozna a felhasználónak és ezáltal nem használja ki a lehetőségét. Ennek ellenére törekedtem minél szebb reszponzív nézet kialakítására. A tervezés folyamatában is próbáltam úgy megtervezni az oldalakat, hogy azok kinézete és az egyes elemek elhelyezése minél felhasználóbb barát legyen és könnyen olvasható legyen.

## **4.6.3 Pénztáros oldalak**

Szabó Anett

A pénztáros oldalaknál sem volt a teljes reszponzivitás szempont. Alapesetben a pénztáros monitor méretű képernyőn látja a vetítéseket, kezeli a vásárlást, számára így a legkönnyebb a kezelés. Mobilos nézetben az vetítések megjelenítése, vásárlás felvitele nehézséget okozna. Azonban fejlesztés során arra is gondoltam, hogy ha bármi technikai hiba lenne a monitorokkal, abban az esetben is lehessen vásárolni, így mindegyik oldalam teljesen reszponzív. Az oldal reszponzivitását Bootstrap5 technológia segítségével alakítottam ki.

Dimensions: Galaxy S8 ▾ 360 x 740 100% ▾ No throttling ▾

5										
6										
7										
8										
9										
10										

Szabad

Éppen foglal

Eladott

Név\*:

Email\*:

Székek száma\*:

Kezdés

Vásárlás

Vissza

Dimensions: Galaxy S8 ▾ 360 x 740 100% ▾ No throttling ▾

Sárkány terem  
11:00:00  
HU

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

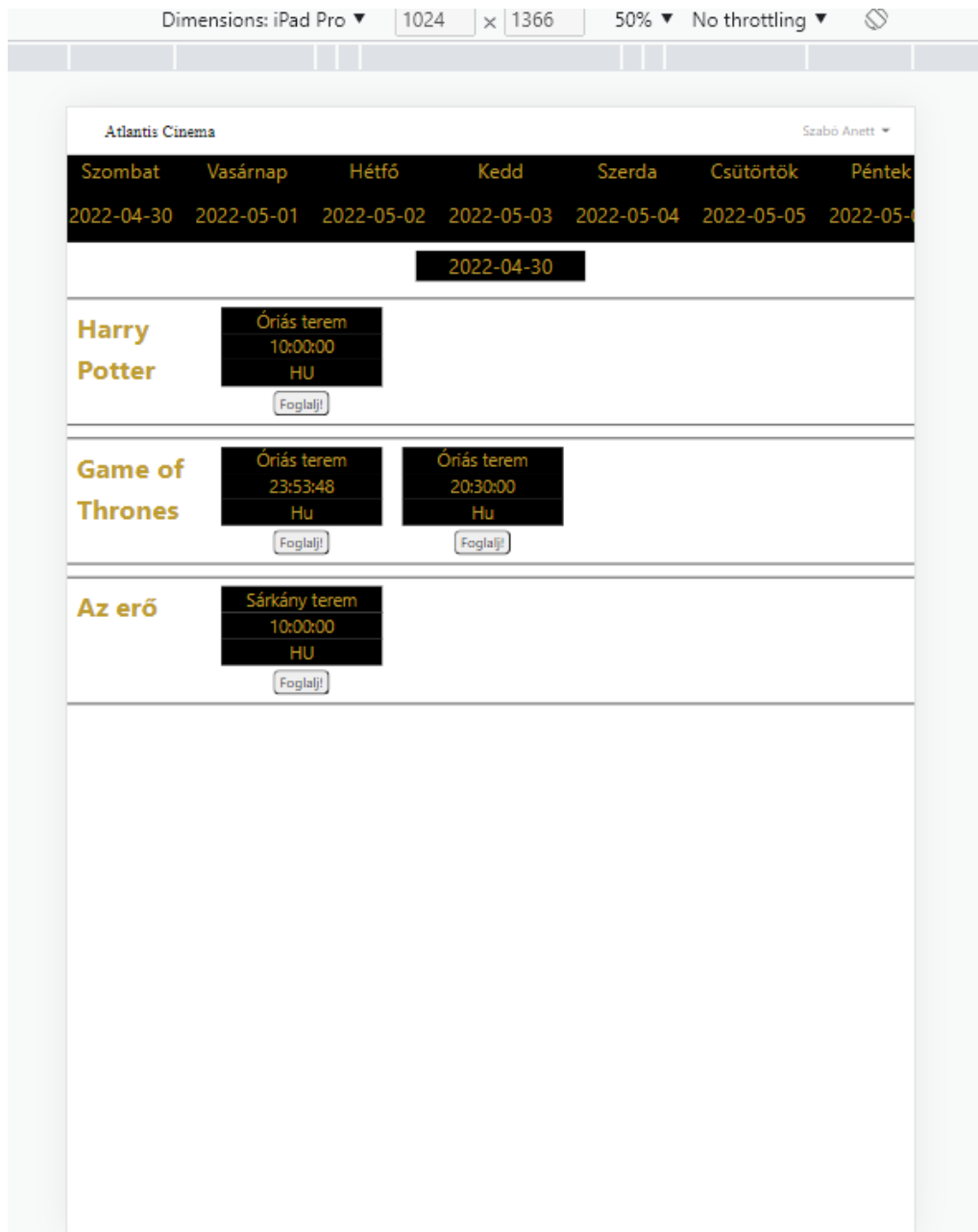
Szabad

Éppen foglal

Eladott

Jól látható, hogy mobil nézetben a kitöltendő mező az ülésterv alatt helyezkedik el.





## 5 Tesztelés

Teszteset	Várt eredmény	Kapott eredmény	Státusz
Szikora Melinda			
Filmek megjelenése a nyitó oldalon betöltéskor	Adatbázisba tárolt filmek megjelennek felsorolva az oldalon	Megjelennek a filmek	Működő
Filmekhez kapcsolt műfajok megjelenése	Filmeknél a műfaj megnevezése jelenik meg	A műfaj azonosító kódja jelenik meg	Javítandó
Filmekhez kapcsolt műfajok megjelenése	Filmeknél a műfaj megnevezése jelenik meg	A műfaj elnevezések jelennek meg a filmeknél, helyesen	Működő
Időpontok megjelenése a film elemekben	A filmhez tartozó létrehozott vetítések időpontjai megjelennek felsorolva	Az időpontok láthatóak minden filmnél	Működő
Plakát galéria megjelenése a nyitó oldalon betöltéskor	A galéria megjelenik, az első kép, de a lapozás még nincs működő állapotban	Megjelenik a galéria első plakátja	Tovább fejlesztendő
Plakát galéria lapozása	A galéria gombjai jobbra és balra is tovább léptetnek a következő plakátra, a végén	A gombok tovább léptetik a képeket, az elején és a végén nem akad el, körbe váltanak a plakátok	Működő

	visszaugrik az első képre, az elején pedig hátra az utolsóra		
Film cím kattintása, film oldal betöltése	Egy film címére való kattintáskor, annak az adott filmnek az oldala betölt, az adataival	A kattintott film megjelenik a film oldalon, minden adat helyesen jelenik meg, <u>kivéve</u> a színészek, nem sorolja fel az összeset, csak az első jelenik meg	Javítandó
Film oldalon a színészek megjelenítése	Az adott film színészei megjelennek felsorolva	A filmhez tartozó, adatbázisban található összes színész megjelenik felsorolva	Működő
Nyitó oldali vetítés időpont kattintása, foglalás oldal betöltése	Megjelenik a foglalás oldal a kattintott film adataival	A film helyesen jelenik meg, de az időpont nem, a legutolsó időpont íródik ki, nem a kattintott	Javított
Szék foglalás	Foglalás oldalon az adatok megadása és székválasztás után a foglalás rögzül	A foglalás rögzül az adatbázisban	Működő
<b>Szabó Anett</b>			
Főoldalon a napok és a hozzájuk tartozó dátum betöltése	Betöltéskor megjelennek a napok és a hozzájuk tartozó dátumok az aktuális naptól számított 1 héttel előre	1 héttel előre jelenek meg a napok és a dátumok	Működő

Dátum kattintáskor megjelennek a filmek és a hozzájuk tartozó időpontok	Amennyiben a választott napon van vetítés és az admin által publikus, akkor kattintásra megjelennek azok a filmek és a hozzájuk tartozó időpontok	Kattintásra megjelennek azok a vetítések, amelyek publikusak és vannak a kattintott napon	Működő
A "Foglalj" gombra kattintva átirányít a foglalás oldalra	A kiválasztott időpont után a gombra kattintva átvissz a foglalás oldalára	A "Foglalj" gomb átirányít a foglalás oldalra	Működő
A foglalás oldalon megjelenik a kattintott időpont adat adatai	A kiválasztott nap után megjelennek az időpont adatai (terem, kezdési idő, szinkron)	Megjelennek az adatok	Működő
Betöltődnek a nézők, akik foglaltak az adott vetítésre	Megjelennek azok a nézők, akik foglaltak az adott vetítésre széket	Az összes néző jelenik meg	Javítandó
Keresés a nézők között	Bármely betű lenyomására képes szűrni a nevek alapján és megjeleníti azokat	Gyorsan törlés vagy a shift gomb lenyomására többször jelennek meg a nézők, minden más esetben megjelenik a keresett néző	Javítandó
Rendezés a nézők között	A-Z opcióra nyomva rendezi ABC sorrendbe a nézőket, a Z-A opcióra nyomva visszafelé rendezi a nézőket	ABC sorrendben és visszafelé is berendezi a nézőket	Működő
Kezdetben nem lehet kattintani az ülés tervet	Az oldal betöltődésekor még nem lehet kattintani az üléstervet csak ha kitöltendő mezőket	Kezdetben nem lehet kattintani	Működő

A kitöltendő mezőknél a "Kezdés" gombra kattintva, az ülésterv kattintható	Amennyiben az összes adatot kiöltöttük akkor a székek kattinthatóak lesznek	A "Kezdés" gombra kattintva, az ülésterv kattintható	Működő
Ha nincs kitöltve minden adat (név, email, székek száma) és a "Kezdés" gombra kattintunk figyelmeztetést kapunk	Abban az esetben, ha nem töltöttük ki az összes mezőt egy felugró ablakot kapunk és nem mehetünk tovább, csak abban az esetben, ha minden mezőt kitöltöttünk	Egy felugró ablak figyelmeztet, hogy minden mezőt ki kell tölteni és nem mehetünk tovább, csak akkor, ha minden mező ki van töltve	Működő
Ha NEM annyi széket választottunk ki amennyit beírtunk és rányomunk a vásárlás gombra figyelmeztetést kapunk	Ha rászereznénk nyomni a vásárlásra gombra, de nem annyi széket választottunk, ki, mint kezdetben, akkor egy felugró ablak figyelmeztet erre	Egy felugró ablak figyelmeztet, hogy annyi széket válasszunk ki amennyit az elején beírtunk, ha ez megvan tovább enged	Működő
Vásárlás gombra kattintva felviszi az nézőket és módosítja a székeket az adatbázisba	Amennyiben a kiválasztottuk az székeket és rányomunk a vásárlás gombra felviszi a nézőket az adatbázisba és módosítja a székeket	A nézőket felviszi az adatbázisba és módosítja a székek állapotát és a nézőt	Működő
Betöltődéskor megjeleníti a foglalt, vásárolt székeket és ezeket NEM lehet módosítani	Amely időpontra rákattintunk megjelenik az adott napon lévő foglalt és vásárolt székek, amelyeket nem lehet módosítani	Nem jeleníti meg a foglalt és a vásárolt székeket	Tovább fejlesztendő
<b>Váradi Ákos</b>			
Személyek oldalon a	Az adatbázisban tárolt személyek adatai táblázatos	A személyek adatai táblázatos formában jelennek meg.	Működő

személyek adatai táblázatos formában jelennek meg.	formában jelennek meg.		
Személyek oldalon a felvitel gombra kattintva az űrlap adatok felkerülnek az adatbázisba és megjelennek a táblázatba.	Az űrlap elem megfelelően van kitöltve. Rányomunk a felvitel gombra akkor az adat bekerülnek az adatbázisba az adat és az űrlap mező kitörlődik. A táblázatalján megjelenik az új adat.	Az űrlapmező helyes kitöltése után felvitel gombra kattintva az adato felkerül az adatbázisba és megjelenik a táblázatban.	Működő
Személyek oldalon a felvitel gombra való kattintáskor az űrlap adata nincsen kitöltve.	Az űrlap adata nincs kitöltve egyáltalán és így nyomunk rá a felvitel gombra akkor egy lebegő ablak jelenik meg az oldal tetején azzal a felirattal, hogy ” Adatbetöltés sikertelen!” és akkor folytathatjuk a munkát, ha az ok gombra rákattintottunk.	Az űrlap üresen lett hagyva és a felvitel gombra kattintva egy lebegő ablak jeleni meg az oldal tetején ” Adatbetöltés sikertelen!” felirattal és az ok gombra való kattintáskor tudjuk folytatni a munkát.	Működő
Személyek oldalon a felvitel gombra való kattintáskor az űrlap adata helytelenül van kitöltve.	Az űrlap adata helytelenül van kitöltve és így nyomunk rá a felvitel gombra akkor egy lebegő ablak jelenik meg az oldal tetején azzal a felirattal, hogy ” Adatbetöltés	Az űrlap helytelenül lett kitöltve és a felvitel gombra kattintva egy lebegő ablak jeleni meg az oldal tetején ” Adatbetöltés sikertelen!” felirattal és az ok gombra való kattintáskor tudjuk folytatni a	Működő

	sikertelen!” és akkor folytathatjuk a munkát, ha az ok gombra rákattintottunk. Az adat pedig törlődik űrlap mezőből.	munkát. Az űrlap mező adatai kitörlődtek.	
Személyek oldalon törlés gomb.	A törlés gombra kattintva az aktuális sorban szereplő adatok törlődnek az adatbázisból és a tábla soraiból.	A törlés gombra kattintva adott személy adatai törlődnek mind az adatbázisból mind a táblázatból.	Működő
Személyek oldalon módosítás gomb.	A módosítás gombra kattintva a sorban szereplő adat betöltődik a megfelelő űrlap mezőbe és a felvitel gomb eltűnik és helyett megjelenik a módosít gomb.	A módosítás gombra kattintva az adat megjelenik az űrlap mezőben és a felvitel gomb helyett a módosít gomb látható.	Működő
Személyek oldalon módosít gomb.	Az űrlapban szereplő adatot lehet módosítani a módosítás gombra kattintva a megváltozott adatok mentésre kerülnek az adatbázisba és a táblázatban is megjelennek ezek az értékek. Az űrlap elemek törlésre kerülnek. A módosít gomb helyett megjelenik a felvitel gomb.	Az űrlap adat mentésre kerül az adatbázisba és megjelenik a táblázatban. A módosít gomb helyett a felvitel gomb lesz látható.	Működő
Személyek oldalon a módosít gombra való	Az űrlap adata helytelenül van kitöltve és így	Az űrlap helytelenül lett kitöltve és a módosít gombra	Működő

kattintáskor az űrlap adata helytelenül van kitöltve	nyomunk rá a módosít gombra akkor egy lebegő ablak jelenik meg az oldal tetején azzal a felirattal, hogy ” Adatmódosítás sikertelen!” és akkor folytathatjuk a munkát, ha az ok gombra rákattintottunk. Az adat pedig törlődik űrlap mezőből.	kattintva egy lebegő ablak jeleni meg az oldal tetején ” Adatmódosítás sikertelen!” felirattal és az ok gombra való kattintáskor tudjuk folytatni a munkát. Az űrlap mező adatai kitörlődtek.	
--	---	---	--



## **6 Fejlesztési lehetőségek**

### **6.1 Publikus felhasználói felületek**

Szikora Melinda

- Filmek megjelenítése egyéb módok, például táblázatban
- Működő foglalási és vásárlási rendszer
- Email szerver létesítése a kapcsolat tartáshoz

### **6.2 Admin felületek**

Váradi Ákos

- Statisztikák készítése.
- Adatbázis táblák bővítése filmek beszerzése táblával.
- Munkakörönként szétosztani a feladatokat.

### **6.3 Pénztáros oldalak**

Szabó Anett

- Jelenítse meg a foglalási oldalon
- Kezelje a vetítéseknél hány szabad szék van még
- Kezelje a kedvezményeket
- Lehessen foglalni is a pénztárban
- Összesítse a vásárlásokat

## 7 Irodalomjegyzék

- Bootstrap5
  - <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- Castellar font
  - <https://fontsforweb.com/font/show?id=21688>
- YouTube lejátszó
  - [https://www.w3schools.com/html/html\\_youtube.asp](https://www.w3schools.com/html/html_youtube.asp)
- Laravel
  - <https://laravel.com/docs/9.x/migrations>
- Mozi inspiráció
  - <https://corvinmozi.hu/>
  - <https://sugarmozi.hu/>
  - <https://www.cinemacity.hu/#/buy-tickets-by-cinema?in-cinema=1132&at=2022-05-03&view-mode=list>
- Felugró ablak
  - [https://www.w3schools.com/howto/howto\\_css\\_modals.asp](https://www.w3schools.com/howto/howto_css_modals.asp)