**ДЪРЖАВЕН ИЗПИТ**

**ЗА ПРИДОБИВАНЕ НА ТРЕТА СТЕПЕН НА ПРОФЕСИОНАЛНА КВАЛИФИКАЦИЯ – ЧАСТ ПО ТЕОРИЯ НА ПРОФЕСИЯТА**

**Д И П Л О М Е Н П Р О Е К Т**

**Тема: „Разработване на уеб базирано приложение за продажба на билети за кино верига“**

**професия: 481030 „Приложен програмист“**

**специалност: 4810301 „Приложно програмиране“**

**Дипломант: Александър Валериев Илиев 12Б клас**

**Ръководител на дипломния проект: Димитрина Делийска**

**Подпис:**

**(дипломант)**

**Подпис:**

**(ръководител)**

**София, 2024 г.**

Съдържание

[1. Въведение 4](#_Toc162888909)

[1.1. Задание на дипломния проект 4](#_Toc162888910)

[1.2. Анализ на заданието, основни цели и идеи 5](#_Toc162888911)

[2. Основна част 6](#_Toc162888912)

[2.1. Избор на технологии 6](#_Toc162888913)

[2.2. Избор на интернет езици 6](#_Toc162888914)

[2.2.1 HTML 6](#_Toc162888916)

[2.2.2 CSS 7](#_Toc162888917)

[2.2.3 JavaScript 7](#_Toc162888918)

[2.2.4 PHP 8](#_Toc162888915)

[2.2.5 MySQL 8](#_Toc162888919)

[2.3. Разработка на приложението 9](#_Toc162888920)

[2.3.1. Реализация на приложението 9](#_Toc162888921)

[2.3.1.1. Начална страница 9](#_Toc162888922)

[2.3.1.2. Login/Register 9](#_Toc162888923)

[2.3.1.3. Възможности на регистриралия се 9](#_Toc162888924)

[2.3.1.4. Администраторски профил и функции 10](#_Toc162888925)

[2.3.2. Функционалност на приложението 10](#_Toc162888926)

[2.3.2.1. MVC архитектура 10](#_Toc162888927)

[2.3.2.2. Dataflow 11](#_Toc162888928)

[2.3.2.3. Таблиците в база данни 12](#_Toc162888929)

[2.3.2.4. Връзките между таблиците 18](#_Toc162888930)

[2.3.2.5. CRUD заявки 19](#_Toc162888931)

[2.3.2.5.1 CRUD примери 20](#_Toc162888932)

[3. Заключение 22](#_Toc162888933)

[3.1. Сравнителна характеристика 22](#_Toc162888934)

[3.2. Перспективи за развитие 22](#_Toc162888935)

[4. Приложения 23](#_Toc162888936)

[Фигура 1: Dataflow диаграма 12](#_Toc162889451)

[Фигура 2: ER диаграма на базата данни и връзките 19](#_Toc162889452)

[Фигура 3: Проверки при регистрация 23](#_Toc162889453)

[Фигура 4: Имейлът след покупка 23](#_Toc162889454)

[Фигура 5: Мобилна версия на главната сраница 24](#_Toc162889455)

[Фигура 6: Навигационното меню според ролята (админ, гост, потребител) 24](#_Toc162889456)

[Фигура 7: Меню с напомняния 25](#_Toc162889457)

[Фигура 8: Popup за следваща премиера 25](#_Toc162889458)

[Фигура 9: Менюто в начална страница 26](#_Toc162889459)

[Фигура 10: Рейтинг на филм и бутон за напомняне 26](#_Toc162889460)

[Фигура 11: Профилна снимка 26](#_Toc162891385)

[Фигура 12: Завършена покупка с печеливш код 27](#_Toc162890780)

[Фигура 13: Админска страница 27](#_Toc162889461)

[Фигура 14: Избор на места 28](#_Toc162891246)

[5. Използвани източници 28](#_Toc162888937)

# Въведение

В съвременния свят, където дигиталните технологии проникват във всеки аспект от живота ни, онлайн платформите за закупуване на билети за кино представляват важна част от потребителския опит. Сегашната пандемия е ускорила този процес, като много хора предпочитат да избягват физическия контакт и да закупят билетите си за кино онлайн.

Онлайн сайтът предоставя удобен начин за клиентите да закупят билети за предстоящи филми. Освен покупки той предоставя възможност за бързо и лесно търсене на филми и техния график на прожекциите, избор на места, и незабавно потвърждение на поръчката. Онлайн платформите за кино билети могат да разширят достъпа до кината за хора, които са в далечни или труднодостъпни места. Те могат да привлекат нови клиенти, които предпочитат удобството на онлайн пазаруването.

Целите на дипломната работа включват:

Анализ на съществуващите онлайн кино билети сайтове: Това включва проучване на функционалността, дизайна и потребителския опит на водещите платформи за закупуване на кино билети.

Проектиране и разработване на собствен онлайн кино билети сайт: Изграждането на иновативна и удобна платформа, която предлага най-добрия опит на потребителя и отговаря на съвременните изисквания.

Изследване на потребителското поведение: Анализ на данните за потребителите, за да се разбере какви са техните предпочитания и как те взаимодействат с платформата.

Оценка на ефективността и усъвършенстване: Проучване на реакцията на потребителите и използване на обратната връзка за постоянно подобряване на платформата.

Резултатите от тази дипломна работа се очаква да включват успешното разработване на ъеб базирано приложение, което предлага изключително удобен и ефективен начин за закупуване на билети за кино. Той ще бъде основан на детайлен анализ на изискванията на потребителите и ще бъде предмет на систематично изследване и оптимизация. Като цяло, тази работа ще допринесе за подобряване на потребителския опит при пазаруването на кино билети и за развитието на сферата на онлайн кино платформите.

## Задание на дипломния проект

Целта на уеб-приложението е да предостави на потребителите възможност да разгледат предалагните услуги и да имат възможността да резервират желаната услуга.

Основни функционалности на приложението:

* Възможност за създаване на аминистраторски и потребителски профил;
* Възможност за търсене и филтриране на филми по зададени критерии;
* Възможност за запазване на билет за прожекция
* Възможност за рейтинг на филм;
* Възможност за използване на кодове за промоция;
* Възможност за получаване на нотификация за филм;

Технологии, за създаването на проекта:

* Frontend (Клиентската част) – HTML, CSS, JavaScript
* Backend (Логическата част) – PHP
* Database (База данни) – MySQL

## Анализ на заданието, основни цели и идеи

Целта на уеб приложението е да предостави на потребителите услуга, която да бъде по-лесна, по-бърза и с подобрени функционалности в сравнение с вече съществуващите алтернативи. В основата си, то е създадено със стремежа да предложи на потребителите по-ефективен и удобен начин за използване й с хубав интерфейс.

По-лесен достъп до услугата: Чрез предоставянето на уеб базирана платформа, потребителите могат да достъпват услугата от всяко устройство, което има интернет достъп. Това улеснява процеса на достъп и употреба на услугата, като предоставя гъвкавост и удобство на потребителите от вкъщи исли навън.

По-бърза и ефективна употреба: Чрез оптимизиране на интерфейса и функционалността, уеб приложението цели да осигури по-бърз и ефективен начин за използване на услугата. Това включва оптимизиране на процесите за регистрация, търсене на информация и използване на ключовите функционалности.

Подобрени функционалности: Уеб приложението може да предложи подобрени функционалности и възможности в сравнение с вече съществуващите алтернативи.

# Основна част

## Избор на технологии

Изборът се състои от проучване на кои технологии са най-подходящи за реализацията.

## Избор на интернет езици

Изборът на интернет езици като PHP, HTML, CSS и JavaScript е изключително подходящ за разработката на уеб базираното приложение поради редица причини.

## 2.2.1. HTML

HTML (HyperText Markup Language) е основен език за маркиране, който се използва за създаване и структуриране на уеб страници. Той е езикът, който определя структурата и съдържанието на уеб страниците, като предоставя средство за изграждане на текстове, графика, мултимедия и връзки.

HTML играе критична роля във всеки уеб сайт, като осигурява следните ползи:

1. **Структура и организация**: HTML предоставя ясна структура за уеб страниците, като позволява лесно организиране на различните елементи, като заглавия, параграфи, списъци, таблиц и други. Това прави уеб страниците по-лесни за разбиране и навигация за потребителите.
2. **Семантика**: HTML разполага с разнообразие от елементи, които имат семантично значение. Това означава, че не само се определя визуалният вид на елементите, но и техните значимости за търсачки и устройства за адаптивен дизайн. Например, използването на елементи като <header>, <nav>, <article>, <section>, <footer> помага за яснота и семантика на уеб страниците.
3. **Достъпност:** Правилното използване на HTML е важно за достъпността на уеб сайтовете. Семантичните елементи и правилната структура на страницата правят уеб съдържанието по-достъпно за хора с увредено зрение или други увреждания.
4. **Съвместимост**: HTML е стандартен език за маркиране, който е поддържан от всички основни уеб браузъри. Това означава, че уеб страниците, създадени с HTML, ще бъдат съвместими и достъпни за широк кръг от потребители.
5. **Интеграция с други технологии:** HTML лесно се интегрира с други уеб технологии като CSS (Cascading Style Sheets) за стилизиране на страниците и JavaScript за добавяне на интерактивност и динамичност.

## 2.2.2. CSS

CSS (Cascading Style Sheets) е стилов език, който се използва за оформяне и стилизиране на уеб страници. Ето някои ключови аспекти и важна информация за CSS:

Разделяне на съдържанието и визуалния дизайн: CSS позволява ясно разделение между съдържанието и визуалния дизайн на уеб страниците. Това означава, че HTML се използва за структуриране на съдържанието, докато CSS се използва за задаване на визуалните стилове като цветове, шрифтове, разположение на елементите и др.

1. **Каскадност**: CSS правилата се прилагат във връзка с каскадата на стиловете. Това означава, че ако има повече от едно правило, което се отнася до даден елемент, те се прилагат в определен ред, като по-специфичните правила имат по-висок приоритет пред по-общите.
2. **Селектори**: CSS използва селектори за да определи кои HTML елементи ще бъдат стилизирани с определените правила. Селекторите могат да бъдат базирани на елементи, класове, идентификатори, атрибути и други.
3. **Поддръжка на различни медии**: CSS предлага възможност за стилизиране на уеб страниците за различни медии, като екрани на компютри, принтери, мобилни устройства и други. Това позволява разработчиците да създават адаптивни и отзивчиви дизайни, които да се адаптират към различните устройства и екрани.
4. **Анимации и преходи**: CSS предоставя възможност за създаване на анимации и преходи, които да подобрят визуалния опит на потребителите. Това включва анимиране на свойства като цветове, размери, позиции и други, както и добавяне на преходи между различни състояния на елементите.

## 2.2.3. JavaScript

JavaScript (JS) е високо-нивен, интерпретиран програмен език, използван за създаване на динамични и интерактивни уеб сайтове. Той е език за скриптове, който се изпълнява в браузъра и дава възможност за взаимодействие с потребителите, манипулиране на HTML и CSS, изпращане на заявки към сървъри и други.

Някои ключови характеристики:

1. **Интерактивност**: JavaScript позволява създаването на интерактивни уеб страници, които реагират на действията на потребителя, като кликване, навеждане на мишката и въвеждане на данни в полета за въвеждане.
2. **Манипулация на HTML и CSS:** С помощта на JavaScript можем да променяме структурата и стиловете на уеб страниците динамично, като добавяме, премахваме или променяме HTML елементи и CSS свойства.
3. **Събития (Events):** JavaScript позволява добавянето на събития към елементи на уеб страницата, като например кликване, преминаване с мишката или въвеждане на клавиши. Това позволява реакция на страницата към действията на потребителя.
4. **Асинхронност:** JavaScript поддържа асинхронно програмиране, което позволява изпълнението на код, който не блокира изпълнението на останалата част от приложението. Това е особено полезно при извършване на заявки към сървъри и обработка на данни от външни източници.
5. **Обектно-ориентирана природа:** JavaScript е обектно-ориентиран език, който позволява създаването на обекти и работа с тях, което улеснява организацията и управлението на кода.

## 2.2.4. PHP

PHP (Hypertext Preprocessor) е широко използван език за програмиране, предимно за уеб разработка, който се използва за създаване на динамични уеб страници и уеб приложения. Ето някои важни аспекти и информация за PHP:

1. **Сървърна страна**: PHP е език за програмиране, който се изпълнява на страната на сървъра. Това означава, че PHP кодът се изпълнява на уеб сървъра, преди да се изпрати резултатът към уеб браузъра на потребителя.
2. **Интеграция с HTML и CSS**: PHP се интегрира лесно с HTML и CSS, което позволява разработчиците да създават динамични уеб страници, които да се адаптират и реагират на действията на потребителите.
3. **Гъвкавост:** PHP предлага голяма гъвкавост и многофункционалност, като поддържа различни операционни системи и уеб сървъри. Той може да работи с различни бази данни, като MySQL, PostgreSQL, SQLite и други.
4. **Динамично съдържание:** PHP позволява генерирането на динамично съдържание на уеб страниците, като например показване на персонализирани съобщения, обработка на формуляри, работа с бази данни и други.
5. **Широка употреба:** PHP е един от най-популярните езици за уеб програмиране и се използва от милиони уеб разработчици по целия свят. Той е отворен, безплатен и има голяма общност от разработчици, което прави неговото изучаване и използване достъпно за всички.

## 2.2.5. MySQL

MySQL е релационна база данни, която се нарежда сред най-широко използваните в света, особено в сфери като уеб разработка, софтуер за управление на данни и други.

Едно от най-големите предимства на MySQL е нейната отворена природа. Това означава, че е достъпна за свободно използване, промяна и разпространение от всеки, който има нужда от нея.

MySQL предлага различни функции, които я правят гъвкава и мощна система за управление на данни. Тя поддържа индексиране, транзакции, сигурност и други важни функции, които гарантират ефективността и сигурността на данните.

Интеграцията на MySQL с програмни езици като PHP, Python и Java я прави популярен избор за разработчиците на уеб приложения. Базата данни се използва за съхранение на данни в таблици, като всяко поле в таблицата съдържа конкретен тип информация.

Предимствата на MySQL включват отворен код, скалируемост, бързина, надеждност, лесна за използване и съвместимост с различни операционни системи и програмни езици. Тя е гъвкава и може да бъде използвана за много различни цели, като предоставя надеждност и производителност на високо ниво.

## 2.3. Разработка на приложението

Разработката на приложение, независимо от неговия характер, е сложен процес. Преди да започнем работата по същинската част, е важно да имаме ясно представление за това как всичко ще изглежда и как бихме се чувствали като потребители, когато използваме приложението.

След като съставим този план и си представим процесите, можем да започнем работата по същинската част, като започнем да пишем кода и да адаптираме дизайна към него.

## 2.3.1. Реализация на приложението

## 2.3.1.1. Начална страница

При влизане в началната страница, потребителите имат възможност да прегледат всички филми чрез главното меню или търсачката. Освен това, могат да видят слайдшоу с най-новите филми и да разгледат менюто с храна и напитки и цените им. Допълнителна информация може да бъде намерена във фуутъра на страницата. Чрез навигацинния бар ще може да влезе в профил или да види най-близката премиера.

## 2.3.1.2. Login/Register

Разбира се, за да може потребителят да получи специални функционалностти, ще трябва да създаде акаунт, ако няма такъв. След като влезе в профила си, той ще седи логнат 1 месец или до натискане на бутона Изход.

## 2.3.1.3. Възможности на регистриралия се

При наличие на профил се откриват различни възможности:

**Профилна страница:** Получаваш достъп до лична страница, в която можеш да си видиш поръчките си, рейтингите, лични данни и възможност за променянето на някои от тях.

**Рейтинг на филм**: Без значение дали си гледал филма, получаваш възможност да дадеш рейтинг от 1 до 5 в страницата за филма.

**Задаване на напомняне:** За филми, които не са излезли в прожекция се появява бутон камбанка при датата на излизане в страницата на филма. При зададено напомняне, бутонът ще свети в синьо.

**Запазване на билет:** Освен само гледане на графика на филм, вече получаваш възможност при кликане на бутона за прожекция да се появи меню за запазване на билет и от него да отвориш страницата за запазване на място.

**Получаване на код:** След покупка полуваш възможност да генерираш код за промоция, който веднага проверяваш дали е активен.При активен такъв изписва колко % е отстъпката. Не е персонален код и друг може да го генерира също, затова трябва да се използва бързо.

## 2.3.1.4. Администраторски профил и функции

След като потребител стане администратор, той придобива права за управление на базата данни. В навигационния бар се появява ново меню за администратори, което води към специална страница, достъпна само за тях. Ако някой потребител се опита да достъпи тази страница чрез променяне на линка, ще получи грешка 404. На тази страница се предоставя списък с всички филми, включително и тези, които не са активни. Администраторът има възможност да променя статуса на филмите и да добавя нови през менюто в страничната част. Освен това, има и друго меню, което представя всички активни кодове, позволявайки на администратора да добавя или премахва нови. След менюто, администраторът може да преглежда всеки профил в списък по имейл и да види профилната страница на потребителя. Ако някой от потребителите е заявил изтриване на профила си, този профил ще бъде маркиран с жълто най-отгоре на списъка. Над този списък е и менщто за създаване на админ. В него се пише имейл и избираш да добавиш или премахнеш. Има проверка, чрез която никой не може да премахне главния админ.

Други администраторски функционалности включват възможността да се редактира цялата информация за филма от информационната страница на филма, чрез специално меню от страничната част. Също така, чрез менюто за прожекции има опция да се добавят прожекции ръчно. Тези менюта са достъпни само за потребители с администраторски права.

## 2.3.2. Функционалност на приложението

## 2.3.2.1. MVC архитектура

За да се обясни MVC (Model-View-Controller) архитектурния шаблон, е добре да разгледаме неговите компоненти и как те си взаимодействат.

1. **Модел (Model)**:

* Моделът представлява данните и бизнес логиката на приложението.
* Той се грижи за обработката, валидацията и манипулирането на данните, които приложението използва.
* Моделът не зависи от визуалния аспект на приложението и е изцяло абстрагиран от него.
* Най-често моделът е реализиран като класове или структури в програмния код.

1. **Изглед (View):**

* Изгледът представлява визуалното представяне на данните, предоставени от модела.
* Той се отговаря за визуализирането на информацията, която потребителят вижда в браузъра или друго устройство.
* Изгледът обикновено е представен в HTML, CSS и JavaScript и се грижи за интерфейса на потребителя.

1. **Контролер (Controller):**

* Контролерът служи като посредник между модела и изгледа.
* Той приема заявките от потребителите (HTTP заявки, например), обработва ги и определя кой модел и изглед трябва да бъдат използвани за тяхното обслужване.
* Контролерът обработва логиката на приложението и връща резултат към потребителя в подходящия формат.

Използвайки тези три компонента в комбинация, MVC архитектурата разделя приложението на логически отделни части, които могат да се разработват и поддържат по-лесно. Моделът се грижи за данните, изгледът за визуализацията им, а контролерът за обработката на заявките и координиране на действията между модела и изгледа.

Това е основната идея на MVC архитектурата, която помага за постигане на по-добра организация и разбираемост на кода в големи и сложни уеб приложения.

## 2.3.2.2. Dataflow

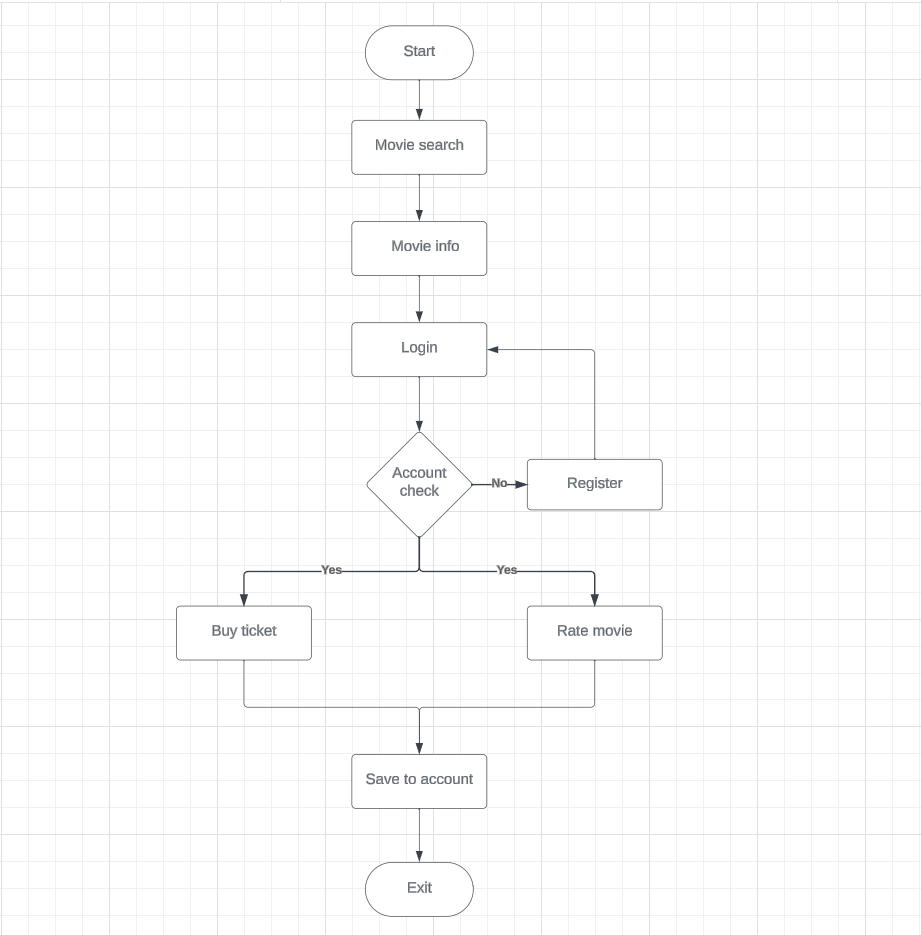
Избор на филм: Потребителите започват с избора на филм, който желаят да гледат. Този етап им предоставя възможността да разгледат различни филми, чрез меню или търсачка, за да вземат решение.

Информация за филма: След като са избрали филм, потребителите могат да прегледат подробна информация за него. Това им помага да се запознаят по-добре с филма, преди да вземат окончателно решение.

Проверка за акаунт: Ако потребителят вече има акаунт, може да продължи напред с покупката на билет или да рейтне филма. Ако не, има възможност да създаде нов акаунт, което също допинася за по-удобен достъп и управление на билетите си в бъдеще.

Купуване на билет: След успешно влизане в системата, потребителите могат да закупят билети за желаната прожекция. Този процес е лесен и бърз, като включва избор на места и плащане.

Съхранение на информацията: Накрая, информацията за закупените билети се съхранява в потребителския акаунт. Това позволява лесен достъп до билетите и историята на посещенията, както и възможност за рейтване на филмите.



Фигура 1: Dataflow диаграма

## 2.3.2.3. Таблиците в база данни

За проекта са създадени 8 таблици, с възможност за добавяне и на още в бъдеше за нови функционалности или оптимизации.

|  |  |  |
| --- | --- | --- |
| Таблица users | | |
| Име на поле | Тип | Описание |
| id\_user | INT | Уникален идентификатор за всеки потребител. Използва за лесното идентифициране на потребителите в системата и за свързване с други таблици в базата данни. |
| f\_name | VARCHAR | Съхранява първото име на потребителя. Това е важно за персонализацията на потребителския опит и за идентификация на потребителите във визуалните елементи на уеб сайта. То ще бъде изписано в навигационния бар |
| l\_name | VARCHAR | Съхранява фамилното име, което се вижда само в профилната страница |
| email | VARCHAR | Много важно поле за идентификация на потребителите. Email адресът се използва за влизане в системата. Разбира се, не може да се повтаря в таблицата. Използва се за изпращане на email при закупка на билет. |
| password | VARCHAR | Съхранява се като хашнат символен низ за повишаване на сигурността на потребителските данни. Паролата трябва да бъде криптирана, за да се предотврати злоупотреба или неоторизиран достъп до профилите на потребителите. Може да бъде сменена през профилната страница. |
| birth | DATE | Информацията за датата на раждане. За момента служи като лична информация в профилната страница. |
| admin | TINYINT | Съхранява 0 или 1 в зависимост дали потребителя е админ. Може да бъде оптимизирано с отделна таблица само за потребители админи. Автоматично полуава 0 при създаване на акаунта. |
| joined | DATE | Датата на регистриране на потребителя |
| pfp | VARCHAR | Линк към картинка за профилна снимка. При създаване на профил, потребителят автоматично има сложена такава. |
| access | TINYINT | Полето access може да съхранява информация за заявките на потребителите за изтриване на профилите им. Оптимизацията на този процес може да включва използване на отделна таблица, която да съхранява заявките за изтриване или в полето за админ да се добави възможност за стойност 2 . Нуждата за полето се породи от това, че когато потребител иска да изтрие профила си поради лични причини, това ще доведе то триене и на историята му за покупки. Затова чрез него се следи кой потребител желае да бъде изтрит, а админът е този, който финално трие от базата данни. Потребителя не може да влезе в акаунта си сред тази заявка, но може да се опита да се свърже с компанията за отказ. |

|  |  |  |
| --- | --- | --- |
| Таблица movies | | |
| Име на поле | Тип | Описание |
| id\_movie | INT | Идентификатор на филма |
| title/director | VARCHAR | Заглавие/режисьор |
| genre | VARCHAR | Жанрът на филм. Може да се изреждат поджанрове или няколко жанра чрез запетаи, но трябва да има поне 1 главен, ако искаме да се филтрира в менюто с жанрове |
| release\_date | TIMESTAMP | Премиерата на филм. От това поле зависи филмовото меню. Чрез него филмите се делят на „в продажба“, „очаквайте скоро“ и „премиера“ |
| duration | INT | Съдържа времетраенето в минути |
| descritpion | TEXT | Описание за филм |
| trailer | VARCHAR | Съдържа съкратен линк от YouTube, който се ипозлва за iframe |
| icon | VARCHAR | Името на файла за иконата на филм |
| age\_rating | INT | Възрастова оценка на филма |
| active | TINYINT | Съдържа 0 или 1 в зависимост дали филмът е в продажба. Ако е 0, филмът е неактивен и само админ може да го види |

|  |  |  |
| --- | --- | --- |
| Таблица ratings | | |
| Име на поле | Тип | Описание |
| id\_rating | INT | Идентификатор на рейтинг |
| id\_user | INT | Външен ключ за потребителите |
| id\_movie | INT | Външен ключ за филм |
| rating\_value | DECIMAL | Запазва рейтина на потребител като число от 1-5. Ипозлвано е decimal за бъдещи промени с десетични числа. |

|  |  |  |
| --- | --- | --- |
| Таблица notifications | | |
| Име на поле | Тип | Описание |
| id\_notif | INT | Идентификатор на нотификацията |
| id\_user/id\_movie | INT | Външни ключове за потребител и филм |
| notif\_date | DATE | Определя кога потребителят ще получи напомняне за предстоящата премиера. При настъпване на указаната дата, потребителят ще бъде уведомен в сайта за предстоящата премиера |

|  |  |  |
| --- | --- | --- |
| Таблица halls | | |
| Име на поле | Тип | Описание |
| id\_hall | INT | Номер на залата |
| seats | INT | Брой места в нея. От това зависи колко места ще има в страницата за избор на място |

|  |  |  |
| --- | --- | --- |
| Таблица projections | | |
| Име на поле | Тип | Описание |
| id\_projection | INT | Идентификатор на рейтинг |
| id\_movie/id\_hall | INT | Външни ключове за филм и зала |
| time | TIME | Часът на прожекцията |
| date | DATE | Датата на прожекцията. Тези полета могат да се комбинират в timestamp |

|  |  |  |
| --- | --- | --- |
| Таблица discounts | | |
| Име на поле | Тип | Описание |
| id\_discount | INT | Идентификатор на код |
| code | VARCHAR | Наименованието на кода, например буквена комбинация |
| amount | INT | Число от 0-100, което ще бъде използвано като % |
| uses | INT | Брой на използвания. Служи като статистика. При всяко използване се прибавя 1 |
| active | TINYINT | Това поле показва дали отстъпката е активна в момента или не. Ако отстъпката е активна, потребителите могат да я използват при покупка на билети. В противен случай, кодът може да бъде неактивен, ако е използван или е деактивиран от администратор. |

|  |  |  |
| --- | --- | --- |
| Таблица tickets | | |
| Име на поле | Тип | Описание |
| id\_ticket | INT | Идентификатор на поръчката/билета |
| id\_user | INT | Външен ключо за потребител |
| id\_projection | INT | Външен ключ за прожекция |
| id\_movie | INT | Външен ключ за филм. Може да се премахне нуждата от това поле заради ключа от прожекция, но с него по-бързо и лесно се достъпва информация за филма |
| price | DOUBLE | Финалната цена на поръчката |
| purchase\_date | TIMESTAMP | Дата и час на поръчката |
| seat\_number | VARCHAR | Пазят се избраните числа на местата чрез запетаи напр (59, 60, 61). Чрез това поле и връзката с прожекция се генерират свободните места в страницата за избор на място. |

## 2.3.2.4. Връзките между таблиците

Връзките в релационните бази данни са основен аспект от базовия модел на данни и играят важна роля в организацията и управлението на данните. Те позволяват на базите данни да бъдат ефективно структурирани и свързани помежду си, като същевременно запазват цялостта и консистентността на данните.

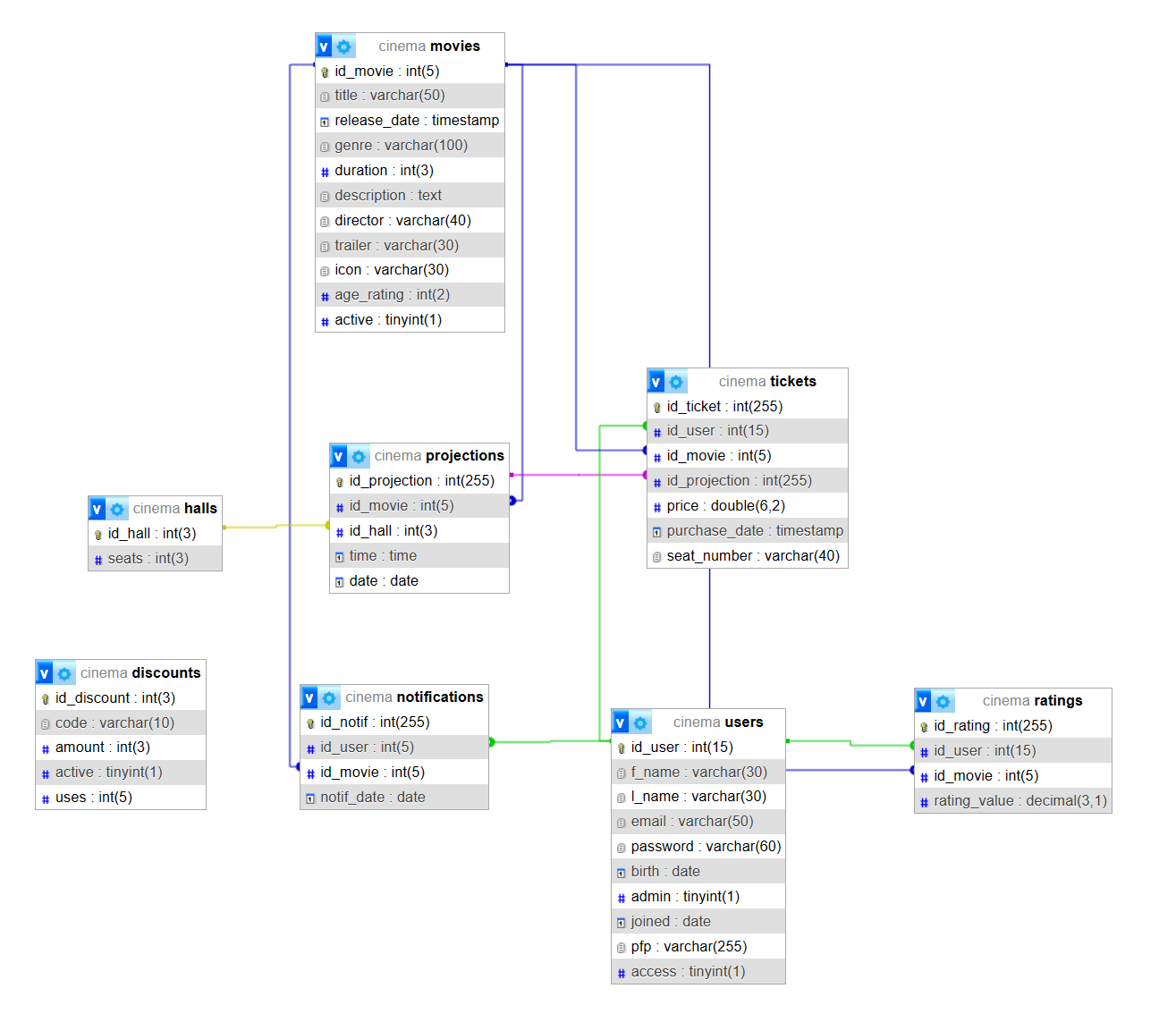
**Външни ключове** (Foreign Keys): Външният ключ е поле или комбинация от полета в таблица, които се свързват с основния ключ (Primary Key) в друга таблица. Това създава връзка между две таблица и определя зависимост между тях. Външният ключ в една таблица указва на основния ключ в друга таблица и това се използва за запазване на целостта на данните.

**Основни ключове** (Primary Keys): Основният ключ е уникален идентификатор на запис в таблица. Той се използва за единствено идентифициране на записите в таблицата и осигурява уникалност на данните. Обикновено основният ключ се свързва с външен ключ в друга таблица, за да се създаде връзка между тях.

**Едно към едно** (One-to-One Relationship): Тук всяко поле в една таблица има точно един запис, свързан с него в друга таблица. Например, един потребител може да има точно един профил.

**Едно към много** (One-to-Many Relationship): Тук всяко едно значение в едно поле в таблица може да бъде свързано с множество стойности в съответното поле в друга таблица. Това създава връзка един към много.

**Много към много** (Many-to-Many Relationship): В този случай множество стойности в едно поле в таблица може да бъде свързано с множество стойности в съответното поле в друга таблица. Тази връзка обикновено се постига чрез добавяне на трета таблица, наречена "промеждутъчна таблица" или "таблица за асоциация", която съдържа два външни ключа - към съответните основни ключове в двата съответни столба.

****

Фигура 2: ER диаграма на базата данни и връзките

## 2.3.2.5. CRUD заявки

CRUD (Create, Read, Update, Delete) парадигмата е широко приложимо явление в уеб разработката, тъй като предоставя структуриран начин за създаване на функционалността на приложенията и изграждане на полезни и лесни за използване модели.

В контекста на CRUD, абревиатурата означава следното:

Create (Създаване) операциите позволяват създаването на нови записи, като например нов акаунт или задача. В SQL бази данни, за създаване на записи се използва операцията INSERT.

Read (Четене) операциите позволяват достъп до съществуващите записи чрез потребителския интерфейс. В релационни бази данни за избиране на записи се използва операцията SELECT.

Update (Обновяване) операциите позволяват промяна на съществуващи записи, отново чрез потребителския интерфейс. В релационни бази данни за обновяване на записи се използва операцията UPDATE.

Delete (Изтриване) операциите позволяват премахване на записи от базата данни. В SQL бази данни, за изтриване на записи се използва операцията DELETE.

## 2.3.2.5.1. CRUD примери

|  |  |  |
| --- | --- | --- |
| Вид | Заявка | Описание |
| INSERT | INSERT INTO users (f\_name, l\_name, email, password, birth, joined) VALUES ('$first\_name', '$last\_name', '$email', '$hashed\_password', '$birth\_date', NOW()) | Създаването на потребител след регистрация. Може да се види, че не всички полета полуават данни от регистрацията, а са автоматични |
| INSERT | INSERT INTO movies (title, genre, director, release\_date, duration, description, trailer, icon, age\_rating) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?) | Създаване на нов филм автоматично неактивен. Админът го създава чрез бутон, слагащ още временни стойности в полетата |
| SELECT | SELECT \* FROM movies WHERE active = 1 AND release\_date <= CURRENT\_DATE | Функционалността на менюто с филми, които са в продажба става чрез селекция на полета активно и премиера да е по-малка или равна на днешна дата |
| SELECT | SELECT \* FROM movies WHERE genre LIKE '%Анимация%' AND active = 1 AND release\_date <= CURRENT\_DATE | Филтриране на менюто по жанр |
| UPDATE | UPDATE users SET admin = " ($action === 'add' ? '1' : '0') . " WHERE email = '$email' | Заявката за променяне на статус на потребител да стане админ |
| DELETE | DELETE FROM ratings WHERE id\_user = $user\_id AND id\_movie = $movieId; | Триене на рейтинг на потребителя за някой филм. Това става чрез бутон, който той клика |
| DELETE | DELETE FROM users WHERE id\_user = ? | Триенето на потребител става само чрез тази заявка, поради автоматичните каскадни операции за всички таблици |

# Заключение

## Сравнителна характеристика

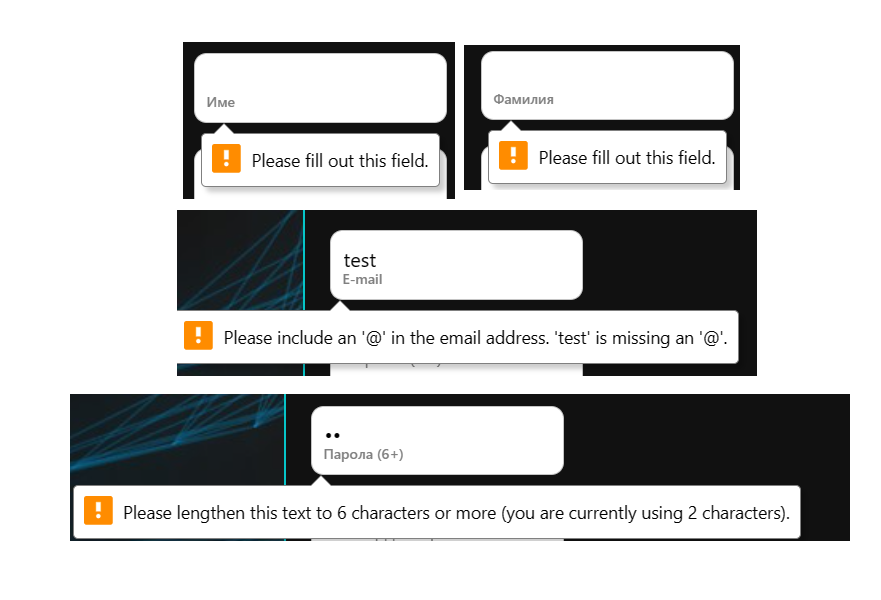
Всички поставени цели и изисквания са успешно постигнати. Дизайнът на приложението отговаря на моите очаквания за привлекателно и лесно за използване приложение. Всички функции работят коректно, след като бяха изпробвани, тествани заедно с премахване на проблеми през разработването. Като краен резултат, може да заключим, че приложението функционира оптимално и безпроблемно.

## Перспективи за развитие

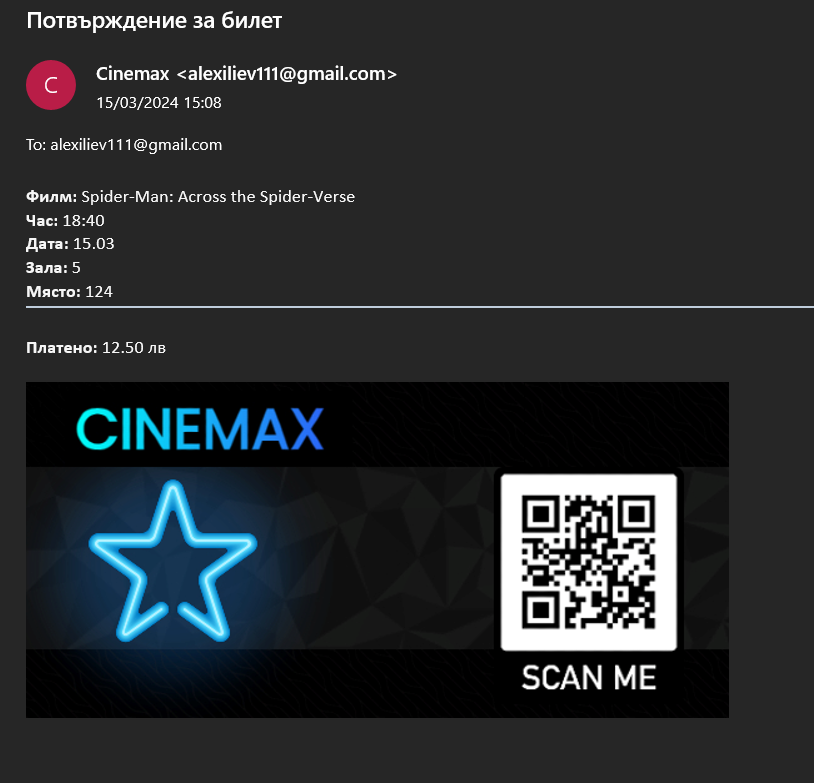
Въпреки успешното реализиране на всички критерии, винаги едно приложение може да приеме много подобрения или нови функционалности. Например още по-добра защита, нови функции за улесняване на потребителите или редизайн на нещо недостатъчно перфектно. Може да се добавят или подобрят следните неща:

1. **Нотификация по имейл при премиера на филм** – значително подобрение на системата за уведомления. Това позволява на потребителите да бъдат информирани, когато нов филм е достъпен, без да е необходимо да посещават сайта, за да проверяват наличността на нови премиери.
2. **Нови видове кодове за промоция** – в момента тази система е доста проста. Може да се добавят нови видове кодове като предметни награди или храни и напитки за следващо посещение. Това ще доведе до промяна на таблицата от бази данни и най-вероятно на генерирането на кодове.
3. **Различни цени на билети** – може да се добавят билети за различни възрасти. Това ще доведе до по-ниски и привлекателни цени за деца, ученици и възрастни.
4. **Коментари за филм** – това ще стимулира по-голямо взаимодействие между потребителите, давайки възможност за споделяне на мнения и впечатления за различните филми, освен само оценяване с рейтинг. Това ще доведе и до включването на имената на потребителите и профилните им снимки в коментарите, но ще се наложи да се обърне внимание на функционалността за качване на профилната снимка.
5. **Промяна в системата за избор на място** – включва промяна към стандартизиран модел с редове и номера на седалките (например, от 1 до 15). Това би улеснило потребителите при избора на места, като въведе по-структуриран и лесно разбираем метод за идентификация на седалките, вместо да има различни индивидуални номера за всяка седалка.

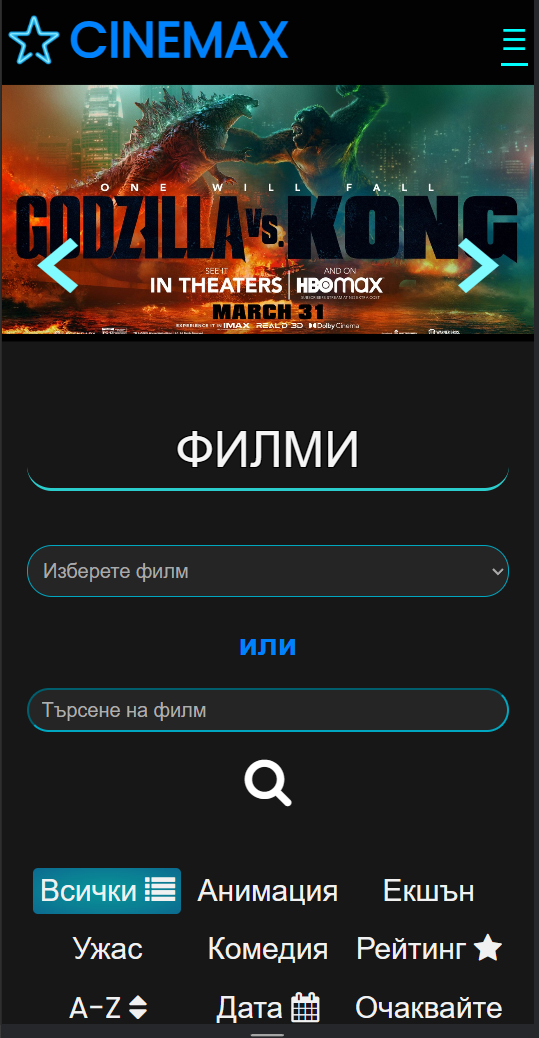
# Приложения



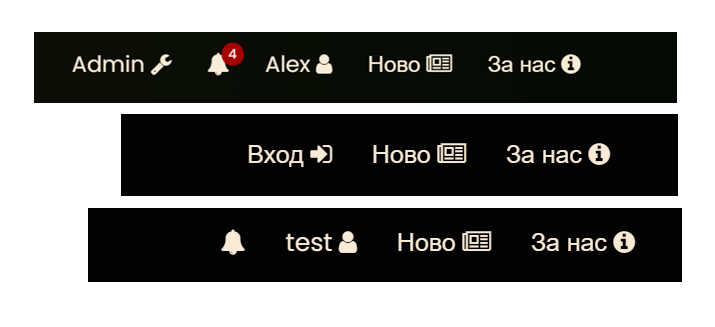
Фигура 3: Проверки при регистрация



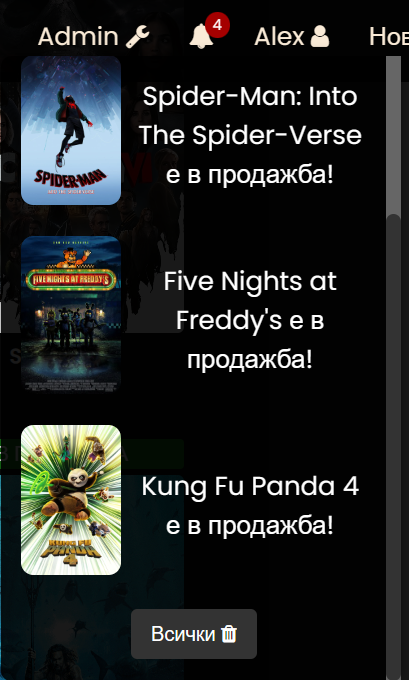
Фигура 4: Имейлът след покупка



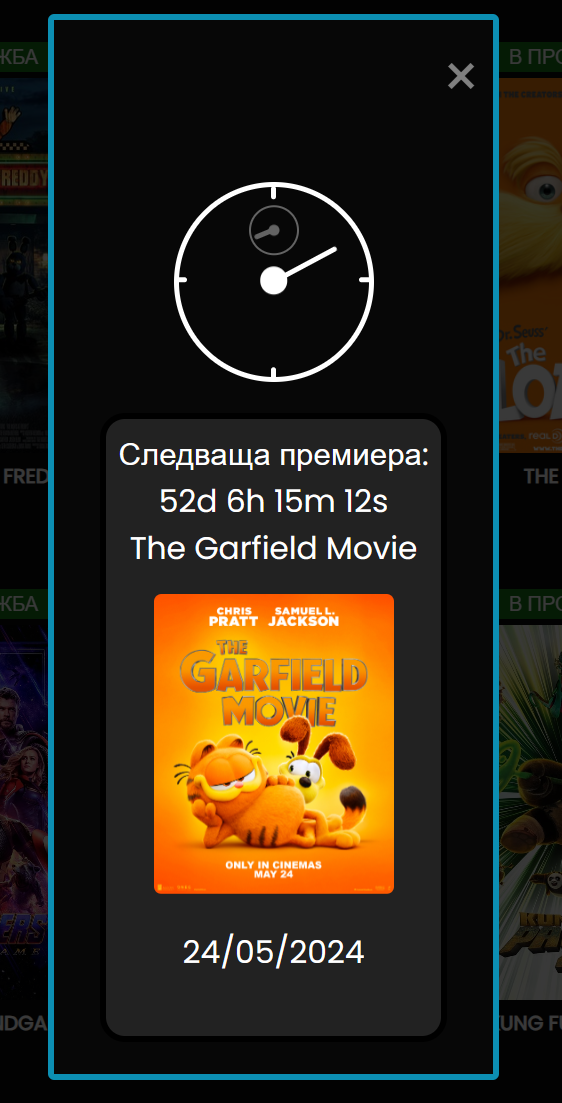
Фигура 5: Мобилна версия на главната сраница



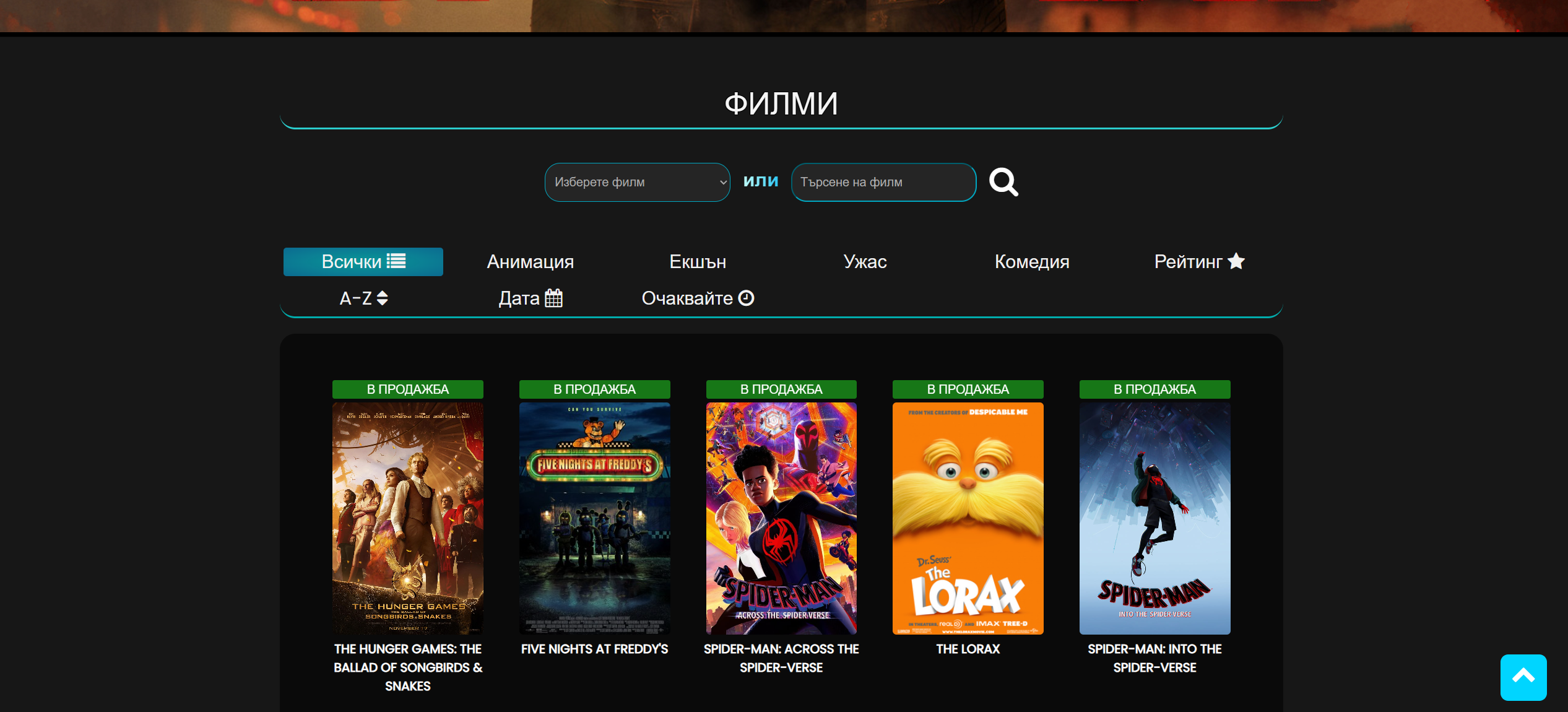
Фигура 6: Навигационното меню според ролята (админ, гост, потребител)



Фигура 7: Меню с напомняния



Фигура 8: Popup за следваща премиера

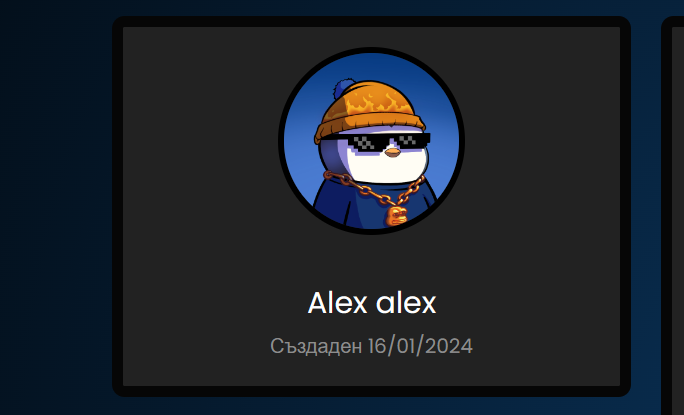


Фигура 9: Менюто в начална страница

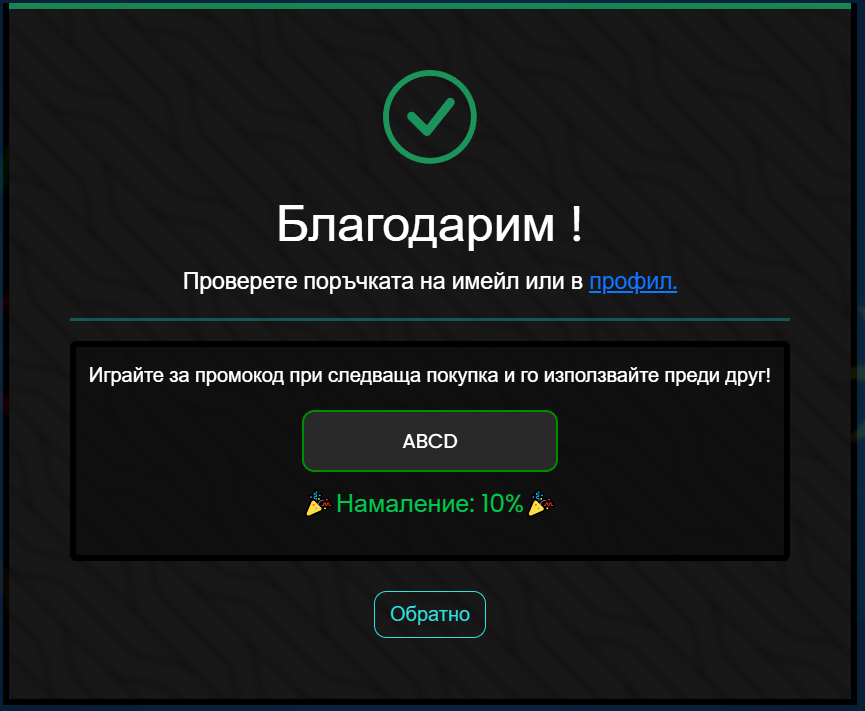




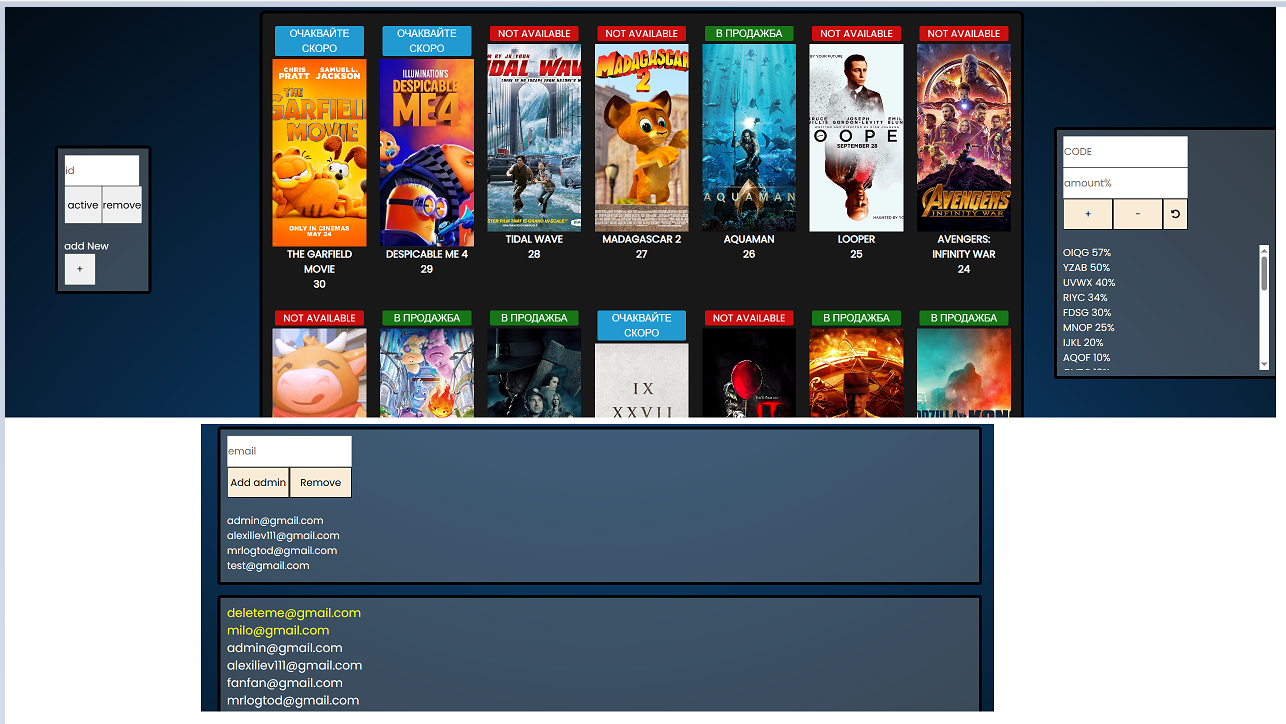
Фигура 10: Рейтинг на филм и бутон за напомняне



Фигура 11: Профилна снимка



Фигура 12: Завършена покупка с печеливш код



Фигура 13: Админска страница



Фигура 14: Избор на места

# Използвани източници

[HTML basics - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)

[CSS Introduction - GeeksforGeeks](https://www.geeksforgeeks.org/css-introduction/)

[Overview of ASP.NET Core MVC | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0)

[YouTube](https://www.youtube.com/)

[ChatGPT (openai.com)](https://chat.openai.com/)