МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**

з дисципліни «Програмування мовою PHP»

*назва дисципліни*

на тему: «Розробка інтернет-магазину з продажу побутової техніки»

Виконав: студент 3 курсу групи № 631п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Пісоцький Д.В.

(прізвище й ініціали студента)

Керівник: к.т.н., доц., доцент каф. 603

Вдовітченко О.В.

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Члени комісії:

(підпис) (прізвище й ініціали)

(підпис) (прізвище й ініціали)

(підпис) (прізвище й ініціали)

Харків – 2023

ЗМІСТ

[1. АНАЛІЗ ВИМОГ ДО ВЕБ-ЗАСТОСУНКУ КІНОТЕАТРУ 4](#_Toc135052218)

[1.1 Формулювання мети і постановка завдань 4](#_Toc135052219)

[1.2 Вимоги замовника 4](#_Toc135052220)

[1.2.1 Функціональні вимоги 4](#_Toc135052221)

[1.2.2 Нефункціональні вимоги 5](#_Toc135052222)

[1.3 Побудова діаграми варіантів використання 5](#_Toc135052223)

[1.4 Огляд існуючих аналогів 6](#_Toc135052224)

[1.5 Вимоги до інтернет-магазину з продажу техніки 7](#_Toc135052225)

[1.5.1 Функціональні вимоги 7](#_Toc135052226)

[1.5.2 Нефункціональні вимоги 8](#_Toc135052227)

[2. ПРОЕКТУВАННЯ І РЕАЛІЗАЦІЯ ВЕБ-САЙТУ МЕРЕЖІ КІНОТЕАТРІВ «ПЛАНЕТА КІНО» 9](#_Toc135052228)

[2.1 Обґрунтування вибору мови програмування для реалізації серверної частини веб-застосунку для кінотеатру 9](#_Toc135052229)

[2.3 Обгрунтування вибору CУБД для реалізації бази даних 11](#_Toc135052230)

[2.4 Створення схематичної моделі бази даних інтернет-магазину 12](#_Toc135052231)

[2.5 Створення системи зручних для людського сприйняття веб-адресів 13](#_Toc135052232)

[ВИСНОВОК 13](#_Toc135052233)

[ПЕРЕЛІК ПОСИЛАНЬ 14](#_Toc135052234)

[Додаток А – Скрипт створення бази даних 15](#_Toc135052235)

[Додаток Б – Код для конфігурації БД 20](#_Toc135052236)

[Додаток В – Код функцій кінотеатру 21](#_Toc135052237)

[Додаток Г – Скріншоти працюючої програми 22](#_Toc135052238)

**ВСТУП**

У сучасному світі відсутність власного сайту може стати причиною значного зменшення успішності будь-якого бізнесу. Особливо це стосується кінотеатрів, де конкуренція надзвичайно велика, а клієнти вимагають все більшого комфорту та можливостей.

У зв'язку з цим, розробка веб-сайту стає важливим етапом в розвитку кінотеатру "Планета Кіно". Цей сайт буде створено з використанням сучасних технологій, таких як PHP, MySQL, HTML, JS, CSS, що дозволить забезпечити максимальну функціональність та зручність для користувачів.

Метою цієї курсової роботи є дослідження процесу розробки веб-сайту та реалізація проекту кінотеатру "Планета Кіно". Робота буде складатися з наступних етапів: аналіз вимог та функціональних можливостей сайту, проектування та розробка бази даних, розробка інтерфейсу користувача, програмування необхідних функцій, тестування та внесення змін.

Основна мета курсової роботи полягає в тому, щоб створити веб-сайт кінотеатру "Планета Кіно", який буде забезпечувати зручний та швидкий доступ до інформації про фільми, сеанси, квитки, акції та інші важливі аспекти діяльності кінотеатру. Для успішного розроблення інтернет-магазину на PHP необхідно мати знання з програмування мовою PHP, роботи з базою даних MySQL та фреймворків. Також потрібно мати розуміння проектування бази даних та веб-дизайну.

Розробка сайту для кінотеатру "Планета Кіно", використовуючи PHP, MySQL, HTML, JS, CSS, є викликом для будь-якого веб-розробника. Однак, результатом є можливість забезпечувати доступ до кінотеатру з будь-якої точки світу та збільшення кількості потенційних клієнтів. Використання PHP для розробки інтернет-сайту є одним з найбільш популярних підходів в сфері веб-розробки.

Крім того, розробка сайту на PHP дозволяє розширити знання та навички в галузі програмування та роботи з базою даних та фреймворками. У результаті виконання проекту, ми отримаємо функціональний сайт кінотеатру, який зможе працювати цілодобово та забезпечувати зручний спосіб придбання квитків для наших клієнтів.

# 1. АНАЛІЗ ВИМОГ ДО ВЕБ-ЗАСТОСУНКУ КІНОТЕАТРУ

## 1.1 Формулювання мети і постановка завдань

Мета: Створити веб-сайт кінотеатру, який надасть користувачам зручний та швидкий спосіб перегляду інформації про фільми та замовлення квитків в Інтернеті.

Завдання:

1. Розробити інтерфейс веб-сайту кінотеатру, що відповідає потребам користувачів.

2. Забезпечити швидку та надійну роботу сайту.

3. Забезпечити безпеку та конфіденційність даних користувачів.

4. Надати інформацію про фільми та розклад сеансів.

5. Забезпечити можливість замовлення квитків з оплатою онлайн.

6. Забезпечити якісну підтримку користувачів.

## 1.2 Вимоги замовника

### 1.2.1 Функціональні вимоги

1. Розклад сеансів: Система повинна мати можливість відображати розклад сеансів фільмів, які показуються в кінотеатрі, та оновлювати його в режимі реального часу.

2. Інформація про фільми: Система повинна мати можливість відображати інформацію про фільми, які показуються в кінотеатрі, включаючи назву, опис, трейлер, акторів та інші характеристики.

3. Квитки: Користувач повинен мати можливість замовити квитки на певний сеанс, вибравши місця та здійснивши оплату онлайн.

4. Оплата: Користувач повинен мати можливість оплатити товар онлайн за допомогою різних методів оплати (банківська картка, електронний гаманець тощо).

5. Контактна інформація: Система повинна мати можливість відображати контактну інформацію для зв'язку з підтримкою, а також забезпечувати можливість надсилання запитань через форму зворотного зв'язку.

### 1.2.2 Нефункціональні вимоги

1. Швидкість роботи: Наш веб-сайт "Планета кіно" повинен працювати швидко та без затримок, щоб користувачі могли легко шукати фільми та купувати квитки без будь-яких перешкод.

2. Надійність: Наш веб-сайт повинен бути надійним та стійким до високого навантаження та можливих відмов, щоб користувачі могли бути впевнені в тому, що їх замовлення будуть успішно оброблені та квитки будуть заброньовані.

3. Безпека: Наш веб-сайт повинен забезпечувати захист від несанкціонованого доступу до даних користувачів та зламів, щоб користувачі могли бути впевнені в тому, що їх особисті дані та фінансова інформація захищені.

4. Користувацький досвід: Наш веб-сайт повинен бути зручним та інтуїтивно зрозумілим для користувачів, забезпечуючи їм приємний користувацький досвід, щоб вони могли з легкістю шукати фільми, купувати квитки та отримувати інформацію про кінотеатр.

5. Сумісність: Наш веб-сайт повинен бути сумісним з різними пристроями та браузерами, щоб користувачі могли використовувати сайт на будь-якому пристрої та в будь-якому браузері, незалежно від того, що вони використовують.

## 1.3 Побудова діаграми варіантів використання

Кінотеатр "Планета Кіно" має веб-сайт, на якому клієнти можуть переглядати розклад сеансів, придбати квитки на фільми та скористатись додатковими послугами. Діаграма варіантів використання допомагає високорівнево описати, як веб-додаток взаємодіє з користувачами та іншими системами. Вона включає в себе типові сценарії взаємодії між користувачем та сайтом кінотеатру. Діаграма була побудована в інструментальному середовищі app [1].

На рисунку 1.3.1 представлена діаграма варіантів використання сайту кінотеатру "Планета Кіно". Користувачі можуть переглядати розклад сеансів, купувати квитки на фільми, залишати відгуки. Адміністратор сайту може додавати та видаляти фільми з розкладу,

встановлювати ціни на квитки та оновлювати інформацію ро послуги.

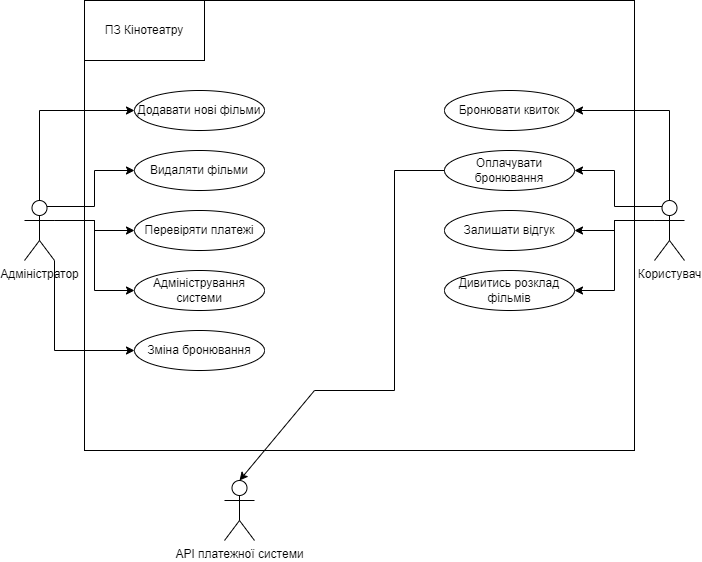


Рисунок 1.3.1 – Діаграма варіантів використання

## 1.4 Огляд існуючих аналогів

Серед існуючих на українському ринку кінотеатрів розглянемо найбільш популярні серед користувачів такі як "Multiplex" та "Wizoria".

1.4.1 Кінотеатр "Multiplex"

URI кінотеатру – "https://multiplex.ua/".

Переваги:

* Зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє знайти потрібний фільм та квитки за кілька кліків.
* Функціонал пошуку з можливістю сортування за різними параметрами (час, дата, жанр тощо).
* Можливість забронювати місця та придбати квитки онлайн.
* Наявність розкладу сеансів на сайті. Недоліки:

Недоліки:

* Обмежена кількість кінотеатрів по всій Україні.

1.4.2 Кінотеатр "Wizoria"

URI кінотеатру – "https://wizoria.ua/".

Переваги:

* Зручний та стильний дизайн сайту з легкою навігацією.
* Можливість вибрати місце та купити квиток онлайн, а також обрати додаткові послуги (наприклад, поїсти в кінотеатрі).
* Можливість використовувати промокоди та отримувати знижки на квитки.
* Наявність програми лояльності для постійних клієнтів.

Недоліки:

* Низька кількість сеансів
* Нижчий рівень звуку в залах

## 1.5 Вимоги до інтернет-магазину з продажу техніки

### 1.5.1 Функціональні вимоги

1. Розклад сеансів: Система повинна мати можливість відображати розклад сеансів, в тому числі на поточний день та майбутні дні. Для кожного сеансу повинна бути назва фільму, час початку, тривалість та зал, де буде проходити сеанс.

2. Оплата: Користувач повинен мати можливість оплатити квитки онлайн за допомогою різних методів оплати (банківська картка, електронний гаманець тощо).

3. Контактна інформація: Система повинна мати можливість відображати контактну інформацію для зв'язку з підтримкою, а також забезпечувати можливість надсилання запитань через форму зворотного зв'язку.

4. Можливість перегляду трейлерів: Користувачам повинна бути надана можливість переглянути трейлери на фільми, що допоможе їм зробити обґрунтований вибір перед відвідуванням кінотеатру.

5. Бронювання місць: Система повинна забезпечувати користувачам можливість забронювати місця на обраному сеансі, вибираючи ряд та місця у залі.

6. Система розрахунку цін: Система повинна автоматично розраховувати ціну на квитки в залежності від кількості та інших параметрів.

### 1.5.2 Нефункціональні вимоги

1. Швидкість роботи: Система повинна працювати швидко та без затримок.

2. Надійність: Система повинна бути надійною та стійкою до високого навантаження та можливих відмов.

3. Безпека: Система повинна забезпечувати захист від несанкціонованого доступу до даних користувачів та зламів.

4. Користувацький досвід: Система повинна бути зручною та інтуїтивно зрозумілою для користувачів, забезпечуючи їм приємний користувацький досвід.

5. Сумісність: Система повинна бути сумісною з різними пристроями та браузерами, щоб користувачі могли використовувати сайт на будь-якому пристрої та в будь-якому браузері.

6. Підтримка: Команда підтримки повинна бути доступною для користувачів та забезпечувати якісну підтримку в разі потреби.

7. Зручність оплати: Система повинна мати зручний та безпечний процес оплати, щоб забезпечити відсутність проблем з оплатою та збільшити шанси на повторні покупки.

8. Підтримка клієнтів: Система повинна мати ефективний механізм підтримки клієнтів, щоб дозволити швидке та якісне вирішення проблем та запитань користувачів.

# 2. ПРОЕКТУВАННЯ І РЕАЛІЗАЦІЯ ВЕБ-САЙТУ МЕРЕЖІ КІНОТЕАТРІВ «ПЛАНЕТА КІНО»

## 2.1 Обґрунтування вибору мови програмування для реалізації серверної частини веб-застосунку для кінотеатру

Мова програмування PHP є однією з найпопулярніших мов для веб-розробки і має декілька переваг для реалізації серверної частини веб-застосунку кінотеатру.

По-перше, PHP є безкоштовною та відкритою мовою програмування з великою спільнотою розробників, що забезпечує доступ до безлічі ресурсів, документації та підтримки. Це дозволяє розробникам ефективно використовувати різноманітні бібліотеки, фреймворки та інші інструменти, які значно полегшують процес розробки веб-додатків кінотеатру.

По-друге, PHP є мовою програмування з великою кількістю вбудованих функцій та можливостей, спеціально розроблених для веб-розробки. Наприклад, у PHP є вбудована підтримка роботи з базами даних та HTTP-протоколом, що дозволяє розробникам легко створювати веб-додатки та обробляти запити користувачів кінотеатру.

По-третє, PHP є мовою програмування з високою швидкістю виконання, що дозволяє ефективно обробляти великий потік запитів на сервері та забезпечувати швидкий доступ до бази даних для кінотеатру.

Загалом, вибір мови програмування для реалізації серверної частини веб-застосунку кінотеатру залежить від вимог до функціональності та особливостей проекту. Проте, з урахуванням переваг, що надає мова програмування PHP для веб-розробки, вона може бути гарним вибором для реалізації серверної частини веб-застосунку кінотеатру. AMPPS є безкоштовним кросплатформеним середовищем, що також містить всі необхідні компоненти для розробки та тестування веб-застосунків, зокрема веб-сервер Apache, базу даних MySQL та інтерпретатор PHP. Однією з головних переваг AMPPS є те, що воно має вбудований механізм для установки та керування різноманітними додатковими компонентами, які можуть бути потрібні для розробки веб-застосунків.

Також AMPPS забезпечує можливість локального тестування веб-застосунків, що є дуже зручною функцією для розробників. Завдяки цьому, розробники можуть швидко та ефективно відлагоджувати свій код та впевнитись, що веб-застосунок працює належним чином, не ризикуючи викласти його на публічний сервер.

Хоча AMPPS і має деякі переваги, які роблять його зручним для розробки веб-застосунків, вибір середовища залежить від потреб та особливостей проекту. Проте, враховуючи те, що AMPPS надає зручний та простий спосіб для розробки та тестування веб-застосунків, воно може бути гарним вибором для розробки серверної частини веб-застосунку кінотеатру.

Веб-сайти кінотеатрів, як правило, містять багато інформації, такої як розклад сеансів, інформація про фільми та їх описи. HTML є стандартною мовою розмітки веб-сторінок, що дозволяє створювати структуру та логічну організацію вмісту, а CSS дозволяє задавати вигляд сторінок та розміщувати елементи на сторінці.

З використанням HTML, CSS, та JS можна створити веб-сайт з привабливим та професійним дизайном, а також забезпечити користувачам зручний та легкий використання. Крім того, використання цих технологій забезпечує швидкість та стабільність роботи веб-сторінок, що є важливим аспектом для успішної онлайн продажу квитків у кінотеатрі.

**2.2 Обгрунтування вибору редактору коду**

Вибір редактору коду є важливим кроком для розробки будь-якого веб-застосунку. У випадку Visual Studio Code (VS Code) є декілька обґрунтувань для його використання:

Підтримка мов програмування: VS Code підтримує не тільки PHP, але й інші мови програмування, включаючи HTML, CSS, JavaScript, SQL та інші. Він має вбудовані функції автодоповнення, перевірки синтаксису, підсвічування синтаксису та інші корисні функції.

Інтеграція з іншими інструментами: VS Code інтегрується з іншими популярними інструментами, такими як Git, Docker, Node.js, іншими пакетними менеджерами та системами контролю версій, що дозволяє легко і швидко налаштувати середовище розробки.

Зручний інтерфейс: VS Code має зручний та інтуїтивно зрозумілий інтерфейс з багатим функціоналом. Він також має підтримку клавішних скорочень, що дозволяє швидко виконувати більшість операцій.

Багатофункціональний: VS Code дозволяє редагувати різні типи файлів, не обмежуючись лише редагуванням коду. Він має підтримку роботи з текстовими файлами, конфігураційними файлами, зображеннями та іншими типами файлів.

Командна розробка: VS Code також дозволяє розробляти код в команді. Його можна налаштувати для використання спільного кодування, обміну змінами через систему контролю версій, а також для спільної відлагодження інших розробників..

## 2.3 Обгрунтування вибору CУБД для реалізації бази даних

Система управління базами даних (СУБД) є програмним забезпеченням, яке дозволяє визначати базу даних, структуру її даних та задавати обмеження для збережених даних. Також СУБД надає можливість маніпулювати даними, контролювати доступ до інформації, забезпечувати безпеку даних та керувати процесами спільного доступу до даних.

MySQL є однією з найпопулярніших СУБД з відкритим вихідним кодом від компанії Oracle, яка забезпечує ефективне зберігання та операції з даними. Використання MySQL забезпечує програмістам простий та зручний інтерфейс для роботи з даними та надає декілька інструментів для виконання різних запитів, включаючи складні операції.

MySQL Workbench є програмним забезпеченням для адміністрування та управління базами даних MySQL від компанії Oracle. Він надає зручний графічний інтерфейс для керування базами даних, дозволяючи виконувати різноманітні операції з даними, такі як створення, редагування та видалення таблиць, запитів та користувачів. Також MySQL Workbench забезпечує підтримку безпеки та цілісності даних.

Вибір MySQL та MySQL Workbench обґрунтовується їх зручністю в використанні, надійністю та широкою підтримкою спільноти програмістів. Вони є стандартом у багатьох веб-застосунках, тому є велика кількість документації та підтримки у мережі Інтернет. Крім того, MySQL та MySQL Workbench працюють на різних операційних системах та платформах, що дає можливість використовувати їх на різних серверах та платформах. MySQL Workbench є офіційним інструментом для роботи з базами даних MySQL від компанії Oracle. Це потужний інструмент для розробки, керування та відлагодження баз даних, який надає ряд корисних функцій та можливостей.

## 2.4 Створення схематичної моделі бази даних інтернет-магазину

Для створення бази даних інтернет-магазину потрібно спочатку визначити сутності між якими будемо будувати зв'язки. Сутності представлені у таблиці 1.

Табл.2.4.1 – Опис сутностей БД інтернет-магазину техніки

|  |  |
| --- | --- |
| *Сутність* | *Опис* |
| movietable | Дана сутність містить інформацію про фільми |
| bookingtable | У даній сутності буде зберігатись інформація бронювання квитків |
| users | Данна сутність буде зберігати інформацію про користувачів |
| feedbacktable | У цій сутності будуть зберігатися відгуки користувачів |

Для того щоб створити базу даних потрібно також зробити UML-діаграму. Вона відображена на рис.2.4.1.

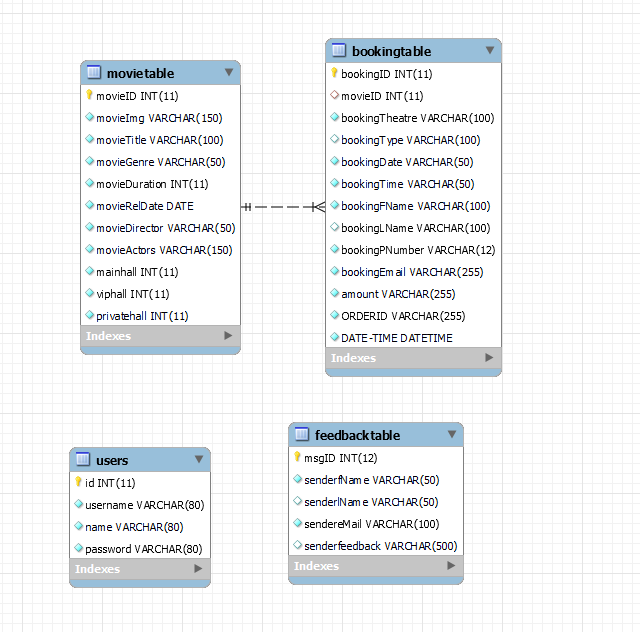


Рисунок 2.4.1 - UML діаграмма БД кінотеатру «Планета Кіно»

## 2.5 Створення системи зручних для людського сприйняття веб-адресів

Так як у застосунку буде використовуватись 2 види інтерфейсу - для звичайних користувачів та адміністратора, тому усі веб адреси для адміна будуть починатись з /admin/\*, а для користувача /public/\*. Цей розподіл допоможе адміну для навігації та полегшує розробку ПЗ.

# ВИСНОВОК

У даній курсовій роботі було розроблено сайт кінотеатру з можливістю придбання квитків онлайн. Для реалізації були використані сучасні технології та інструменти, зокрема мова програмування PHP, яка є популярною для створення веб-застосунків, СУБД MySql та інтерфейс для неї MySQL Workbench, редактор коду VS Code, який дозволяє ефективно працювати з кодом, AMPSS як локальний веб-сервер для розгортання веб-застосунку, а також JS для створення користувацького інтерфейсу.

В результаті було розроблено функціональний сайт кінотеатру, який дозволяє користувачам швидко та зручно придбати квитки на покази. Реалізація була здійснена з використанням сучасних технологій, що дозволило створити досить швидкий та надійний веб-застосунок.

Отже, використання мови програмування PHP, СУБД MySql та MySQL Workbench, редактора коду VS Code, локального веб-серверу AMPSS, а також JS для створення інтерфейсу є досить ефективним способом розробки веб-застосунків та дозволяє отримати функціональний та надійний сайт кінотеатру.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Ларман, Крег Застосування UML і шаблонів проектування. 2-е видання: Пер. з англ. - М: Видавничий дім «Вільямс», 2004. - 624 с .: іл. - Парал. тит. англ.

2. Діаграми [Електронний ресурс] – Режим доступу до ресурсу: https://app.diagrams.net/.

4. Сайт кінотеатру [Електронний ресурс] – Режим доступу до ресурсу: https://multiplex.com.ua/.

5. Сайт кінотеатру [Електронний ресурс] – Режим доступу до ресурсу: https://wizoria.ua/.

7. PHP DOCS [Електронний ресурс] – Режим доступу до ресурсу: https://www.php.net/docs.php

8. MYSQL DOCS [Електронний ресурс] – Режим доступу до ресурсу: https://dev.mysql.com/doc/

9. JavaScript DOCS [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/JavaScript

10. AMPPS Documentation [Електронний ресурс] – Режим доступу до https://ampps.com/docs/

# Додаток А – Скрипт створення бази даних

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `cinema\_db`

--

-- --------------------------------------------------------

--

-- Table structure for table `bookingtable`

--

CREATE TABLE `bookingtable` (

`bookingID` int(11) NOT NULL,

`movieID` int(11) DEFAULT NULL,

`bookingTheatre` varchar(100) NOT NULL,

`bookingType` varchar(100) DEFAULT NULL,

`bookingDate` varchar(50) NOT NULL,

`bookingTime` varchar(50) NOT NULL,

`bookingFName` varchar(100) NOT NULL,

`bookingLName` varchar(100) DEFAULT NULL,

`bookingPNumber` varchar(12) NOT NULL,

`bookingEmail` varchar(255) NOT NULL,

`amount` varchar(255) NOT NULL,

`ORDERID` varchar(255) NOT NULL,

`DATE-TIME` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `bookingtable`

--

INSERT INTO `bookingtable` (`bookingID`, `movieID`, `bookingTheatre`, `bookingType`, `bookingDate`, `bookingTime`, `bookingFName`, `bookingLName`, `bookingPNumber`, `bookingEmail`, `amount`, `ORDERID`, `DATE-TIME`) VALUES

(71, 6, 'private-hall', 'imax', '14-3', '15-00', 'xyz', 'abc', '000000000', '000@gmail.com', '5000.00', 'cash', '2020-12-14 12:20:31');

-- --------------------------------------------------------

--

-- Table structure for table `feedbacktable`

--

CREATE TABLE `feedbacktable` (

`msgID` int(12) NOT NULL,

`senderfName` varchar(50) NOT NULL,

`senderlName` varchar(50) DEFAULT NULL,

`sendereMail` varchar(100) NOT NULL,

`senderfeedback` varchar(500) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `movietable`

--

CREATE TABLE `movietable` (

`movieID` int(11) NOT NULL,

`movieImg` varchar(150) NOT NULL,

`movieTitle` varchar(100) NOT NULL,

`movieGenre` varchar(50) NOT NULL,

`movieDuration` int(11) NOT NULL,

`movieRelDate` date NOT NULL,

`movieDirector` varchar(50) NOT NULL,

`movieActors` varchar(150) NOT NULL,

`mainhall` int(11) NOT NULL,

`viphall` int(11) NOT NULL,

`privatehall` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `movietable`

--

INSERT INTO `movietable` (`movieID`, `movieImg`, `movieTitle`, `movieGenre`, `movieDuration`, `movieRelDate`, `movieDirector`, `movieActors`, `mainhall`, `viphall`, `privatehall`) VALUES

(1, 'img/movie-poster-1.jpg', 'Captain Marvel', ' Action, Adventure, Sci-Fi ', 220, '2018-10-18', 'Anna Boden, Ryan Fleck', 'Brie Larson, Samuel L. Jackson, Ben Mendelsohn', 0, 0, 0),

(3, 'img/movie-poster-3.jpg', 'The Lego Movie', 'Animation, Action, Adventure', 110, '2014-02-07', 'Phil Lord, Christopher Miller', 'Chris Pratt, Will Ferrell, Elizabeth Banks', 0, 0, 0),

(5, 'img/movie-poster-5.jpg', 'VICE', 'Biography, Comedy, Drama', 132, '2018-12-25', 'Adam McKay', 'Christian Bale, Amy Adams, Steve Carell', 0, 0, 0),

(6, 'img/movie-poster-6.jpg', 'The Vanishing', 'Crime, Mystery, Thriller', 107, '2019-01-04', 'Kristoffer Nyholm', 'Gerard Butler, Peter Mullan, Connor Swindells', 0, 0, 0);

-- --------------------------------------------------------

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`username` varchar(80) NOT NULL,

`name` varchar(80) NOT NULL,

`password` varchar(80) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`id`, `username`, `name`, `password`) VALUES

(1, '123', 'john', '123');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `bookingtable`

--

ALTER TABLE `bookingtable`

ADD PRIMARY KEY (`bookingID`),

ADD UNIQUE KEY `bookingID` (`bookingID`),

ADD KEY `foreign\_key\_movieID` (`movieID`),

ADD KEY `foreign\_key\_ORDERID` (`ORDERID`);

--

-- Indexes for table `feedbacktable`

--

ALTER TABLE `feedbacktable`

ADD PRIMARY KEY (`msgID`),

ADD UNIQUE KEY `msgID` (`msgID`);

--

-- Indexes for table `movietable`

--

ALTER TABLE `movietable`

ADD PRIMARY KEY (`movieID`),

ADD UNIQUE KEY `movieID` (`movieID`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `bookingtable`

--

ALTER TABLE `bookingtable`

MODIFY `bookingID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=72;

--

-- AUTO\_INCREMENT for table `feedbacktable`

--

ALTER TABLE `feedbacktable`

MODIFY `msgID` int(12) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT for table `movietable`

--

ALTER TABLE `movietable`

MODIFY `movieID` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=69;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `bookingtable`

--

ALTER TABLE `bookingtable`

ADD CONSTRAINT `foreign\_key\_movieID` FOREIGN KEY (`movieID`) REFERENCES `movietable` (`movieID`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

INSERT INTO `movietable` (`movieID`, `movieImg`, `movieTitle`, `movieGenre`, `movieDuration`, `movieRelDate`, `movieDirector`, `movieActors`, `mainhall`, `viphall`, `privatehall`) VALUES

(2, "img/movie-poster-2.jpg", "Bohemian Rhapsody", "Biography, Musical", "144", "2018-10-24", "Bryan Singer, Dexter Fletcher", "Rami Malek, Lucy Boynton", 0,0,0);

use cinema\_db;

INSERT INTO `movietable` (`movieID`, `movieImg`, `movieTitle`, `movieGenre`, `movieDuration`, `movieRelDate`, `movieDirector`, `movieActors`, `mainhall`, `viphall`, `privatehall`) VALUES

(4, "img/movie-poster-4.jpg", "Inception", "Drama, Sci-Fi", "158" ,"2010-07-22", "Christopher Nolan", "Leonardo DiCaprio, Ken Watanabe" , 0,0,0);

select \* from movietable;

# Додаток Б – Код для конфігурації БД

<?php

$host = "localhost"; /\* Host name \*/

$user = "root"; /\* User \*/

$password = "Suma1l24\_"; /\* Password \*/

$dbname = "cinema\_db"; /\* Database name \*/

$con = mysqli\_connect($host, $user, $password,$dbname);

// Check connection

if (!$con) {

die("Connection failed: " . mysqli\_connect\_error());

}

?>

# Додаток В – Код функцій кінотеатру

*booking.php*

<!DOCTYPE html>

<html lang="en">

<?php

$id = $\_GET['id'];

//conditions

if ((!$\_GET['id'])) {

echo "<script>alert('You are Not Suppose to come Here Directly');window.location.href='index.php';</script>";

}

include "connection.php";

$movieQuery = "SELECT \* FROM movieTable WHERE movieID = $id";

$movieImageById = mysqli\_query($con, $movieQuery);

$row = mysqli\_fetch\_array($movieImageById);

?>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="style/styles.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

<title>Book <?php echo $row['movieTitle']; ?> Now</title>

<link rel="icon" type="image/png" href="img/logo.png">

<script src="\_.js "></script>

</head>

<body style="background-color:#6e5a11;">

<div class="booking-panel">

<div class="booking-panel-section booking-panel-section1">

<h1>RESERVE YOUR TICKET</h1>

</div>

<div class="booking-panel-section booking-panel-section2" onclick="window.history.go(-1); return false;">

<i class="fas fa-2x fa-times"></i>

</div>

<div class="booking-panel-section booking-panel-section3">

<div class="movie-box">

<?php

echo '<img src="' . $row['movieImg'] . '" alt="">';

?>

</div>

</div>

<div class="booking-panel-section booking-panel-section4">

<div class="title"><?php echo $row['movieTitle']; ?></div>

<div class="movie-information">

<table>

<tr>

<td>GENGRE</td>

<td><?php echo $row['movieGenre']; ?></td>

</tr>

<tr>

<td>DURATION</td>

<td><?php echo $row['movieDuration']; ?></td>

</tr>

<tr>

<td>RELEASE DATE</td>

<td><?php echo $row['movieRelDate']; ?></td>

</tr>

<tr>

<td>DIRECTOR</td>

<td><?php echo $row['movieDirector']; ?></td>

</tr>

<tr>

<td>ACTORS</td>

<td><?php echo $row['movieActors']; ?></td>

</tr>

</table>

</div>

<div class="booking-form-container">

<form action="verify.php" method="POST">

<select name="theatre" required>

<option value="" disabled selected>THEATRE</option>

<option value="main-hall">Main Hall</option>

<option value="vip-hall">VIP Hall</option>

<option value="private-hall">Private Hall</option>

</select>

<select name="type" required>

<option value="" disabled selected>TYPE</option>

<option value="3d">3D</option>

<option value="2d">2D</option>

<option value="imax">IMAX</option>

<option value="7d">7D</option>

</select>

<select name="date" required>

<option value="" disabled selected>DATE</option>

<option value="12-3">July 12,2023</option>

<option value="13-3">July 13,2023</option>

<option value="14-3">July 14,2023</option>

<option value="15-3">July 15,2023</option>

<option value="16-3">July 16,2023</option>

</select>

<select name="hour" required>

<option value="" disabled selected>TIME</option>

<option value="09-00">09:00 AM</option>

<option value="12-00">12:00 AM</option>

<option value="15-00">03:00 PM</option>

<option value="18-00">06:00 PM</option>

<option value="21-00">09:00 PM</option>

<option value="24-00">12:00 PM</option>

</select>

<input placeholder="First Name" type="text" name="fName" required>

<input placeholder="Last Name" type="text" name="lName">

<input placeholder="Phone Number" type="text" name="pNumber" required>

<input placeholder="email" type="email" name="email" required>

<input type="hidden" name="movie\_id" value="<?php echo $id; ?>">

<button type="submit" value="save" name="submit" class="form-btn">Book a seat</button>

</form>

</div>

</div>

</div>

<script src="scripts/jquery-3.3.1.min.js "></script>

<script src="scripts/script.js "></script>

</body>

</html>

# Додаток Г – Скріншоти працюючої програми



Рисунок 1 – частина першої сторінки сайту

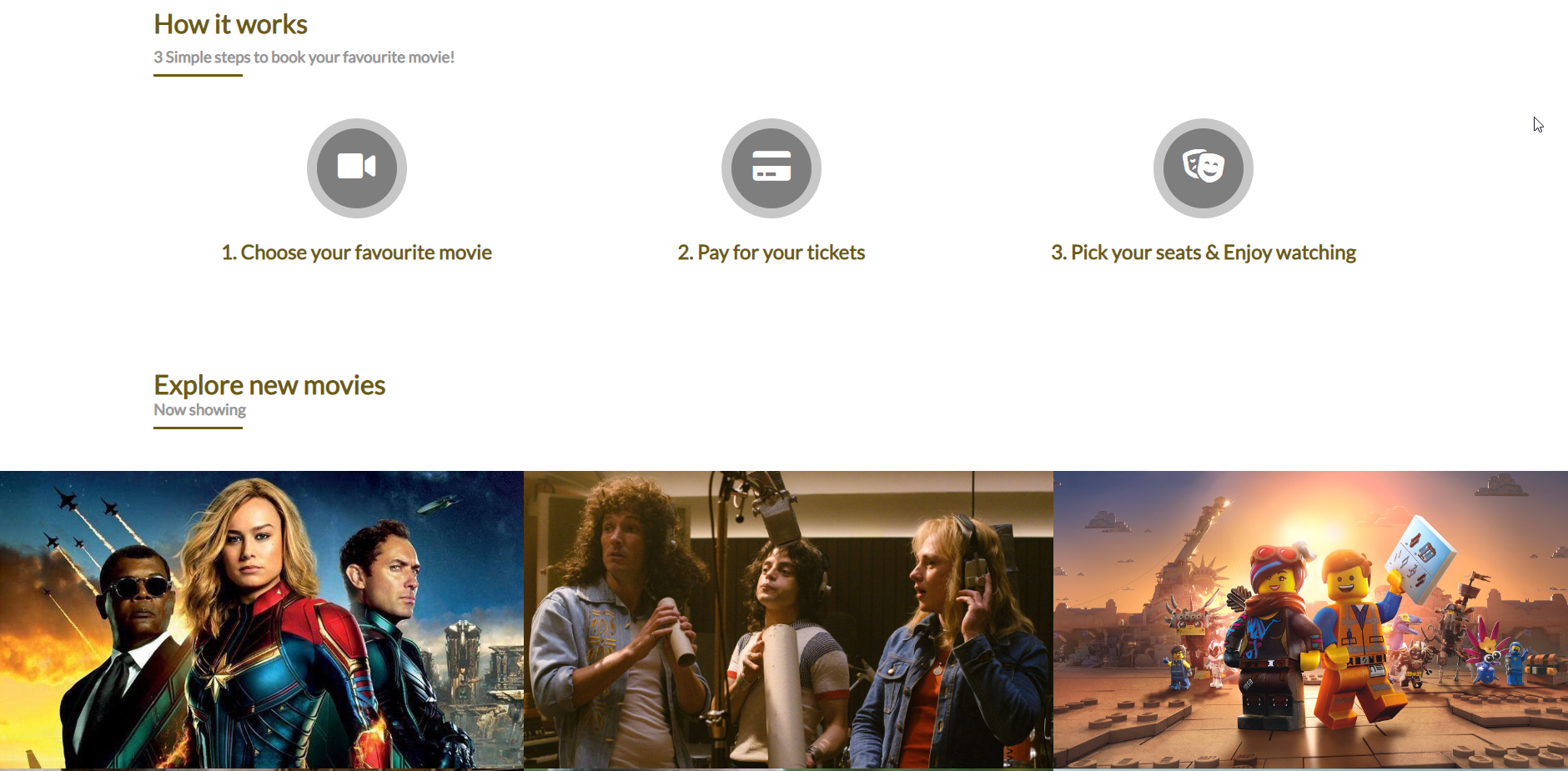


Рисунок 2 – частина першої сторінки сайту

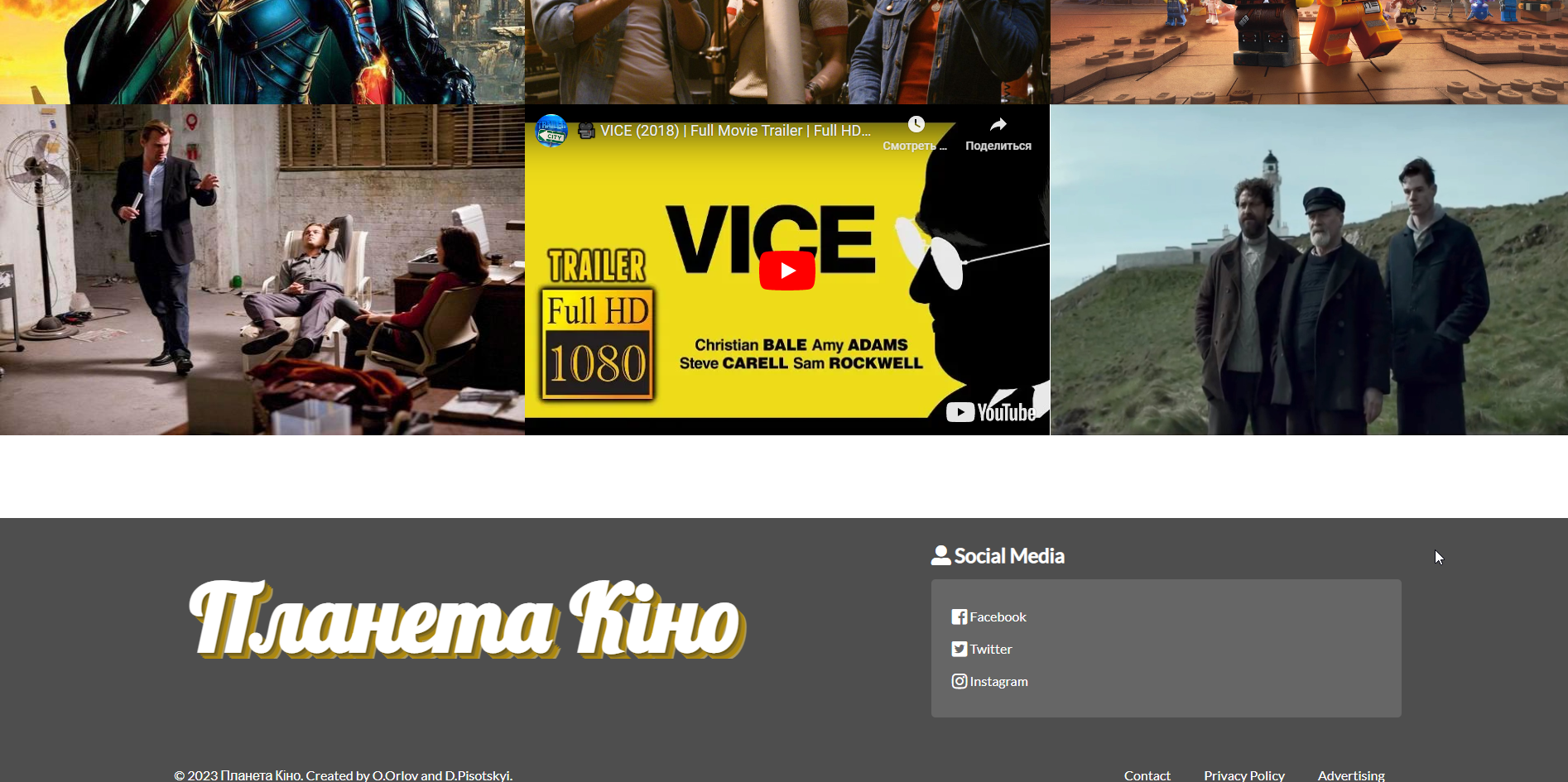


Рисунок 3 – вигляд футеру та трейлеру при натисканні на картинку

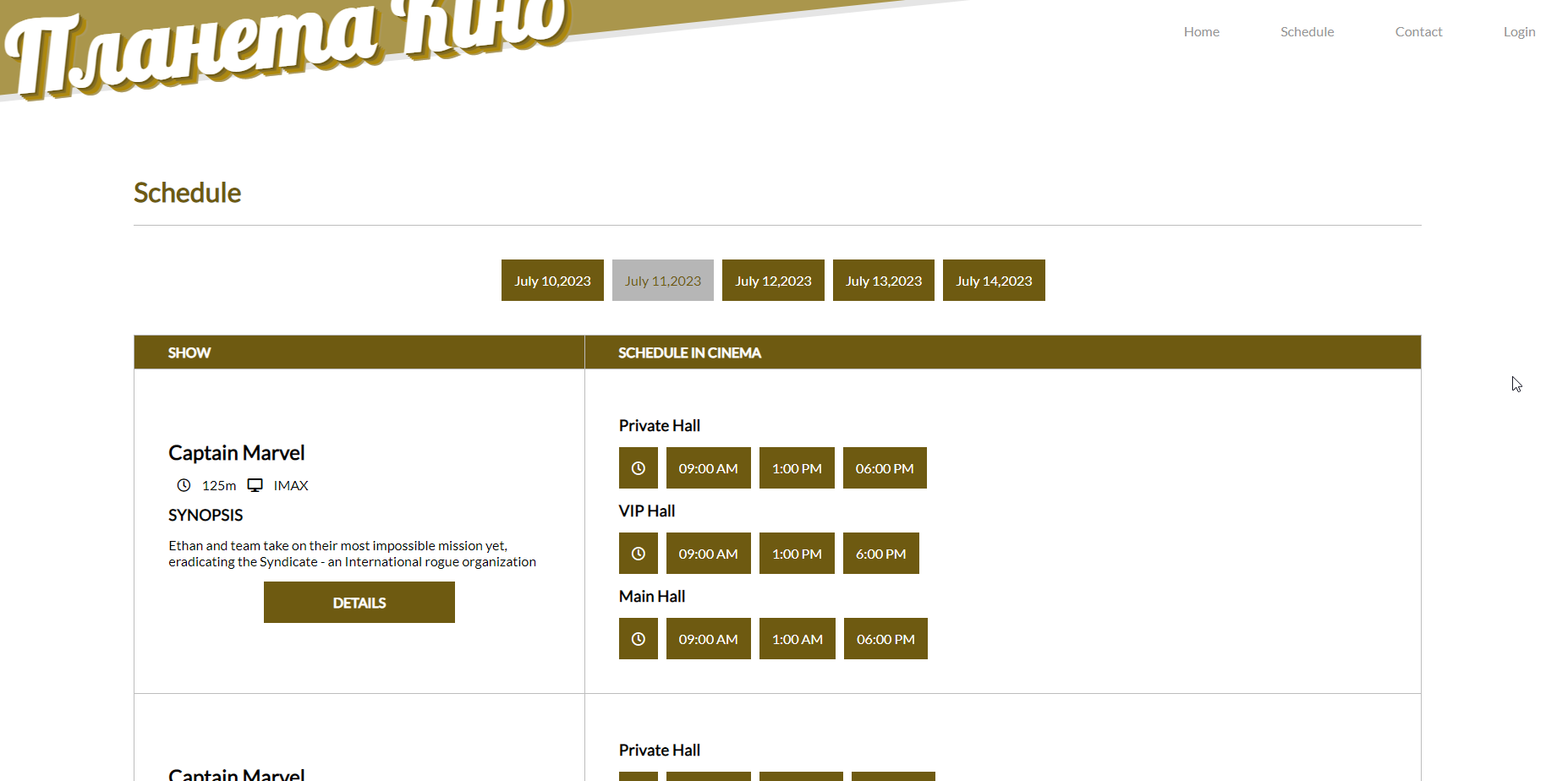


Рисунок 4 – вигляд сторінки з розкладом



Рисунок 5 – вигляд сторінки з контактною інформацією та формою для відгуків

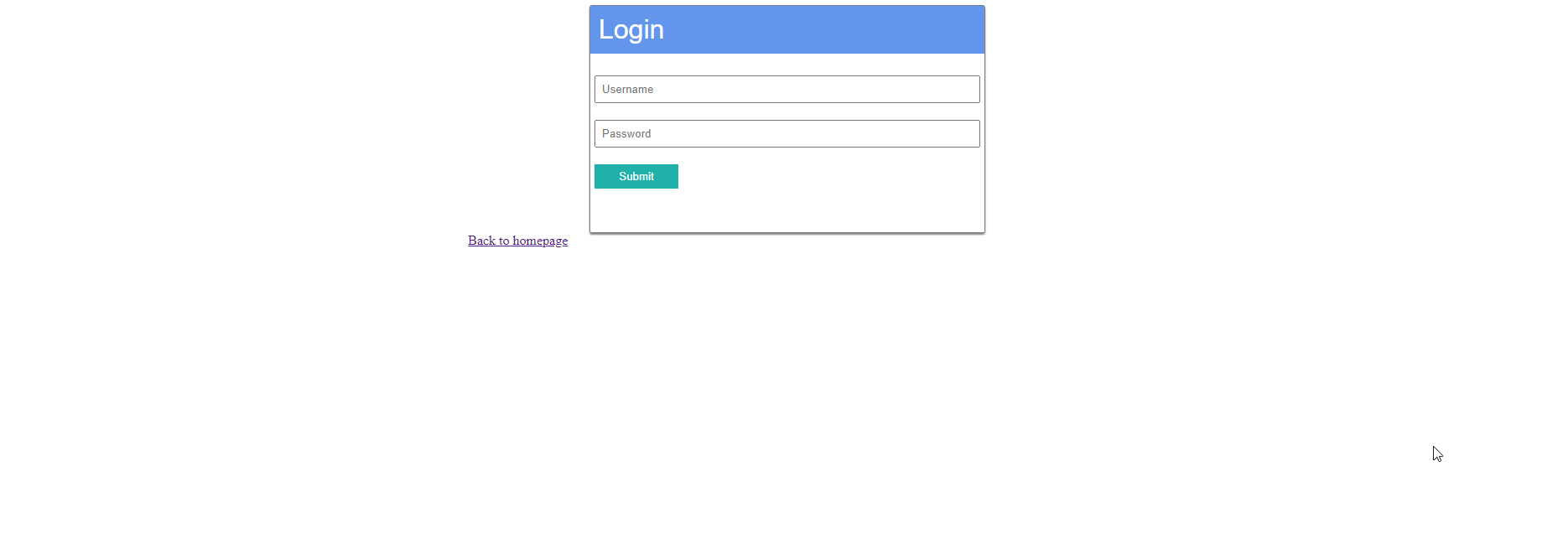


Рисунок 6 – login page

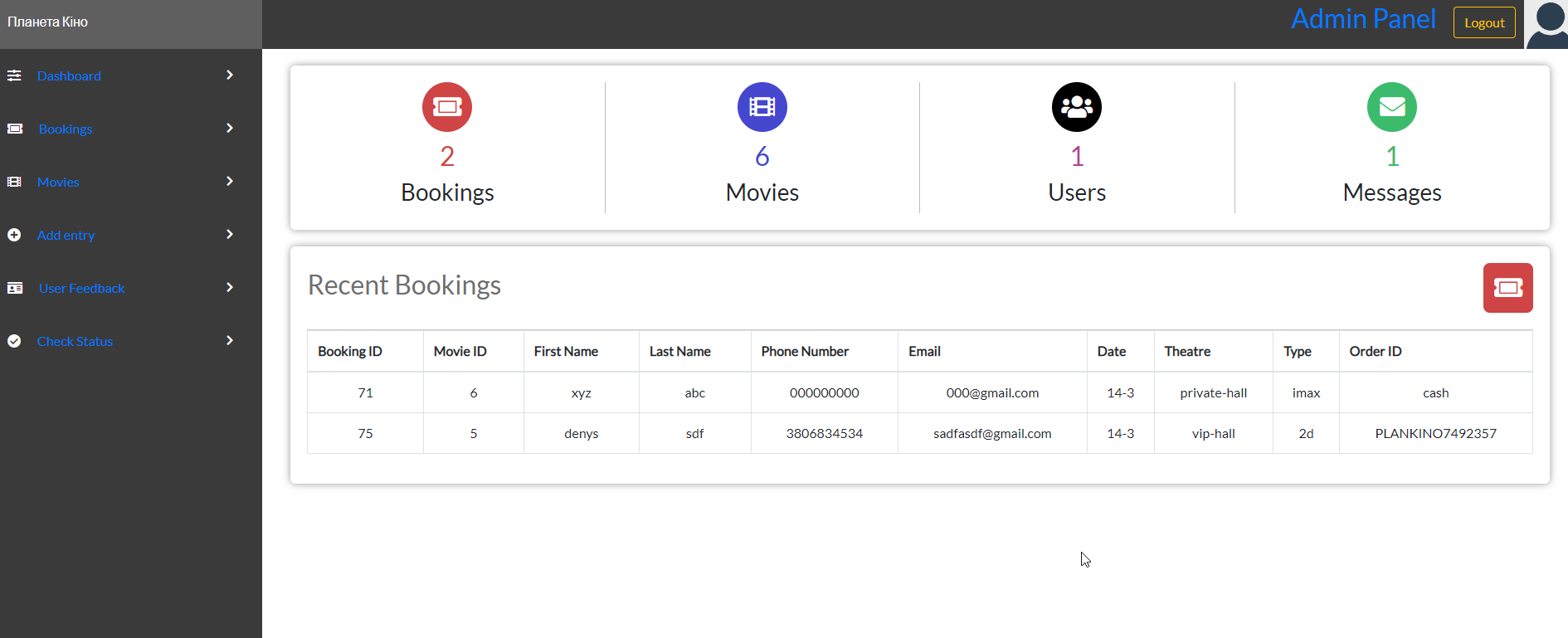


Рисунок 7 – вигляд сторінки адміну