



Software Engineering and Testing. BSC Year 2, 2024/2025

Assessment 4: Software Testing

Submitted by:

B00160049 Matthew Keenan

B00159340 Stanislav Kolev

B0016388 Fabio Melilo

Submission date

24/3/25

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: B00160049 Matthew Keenan

Dated: 24/4/25

Author: **B00159340 Stanislav Kolev**

Dated: 24/4/25

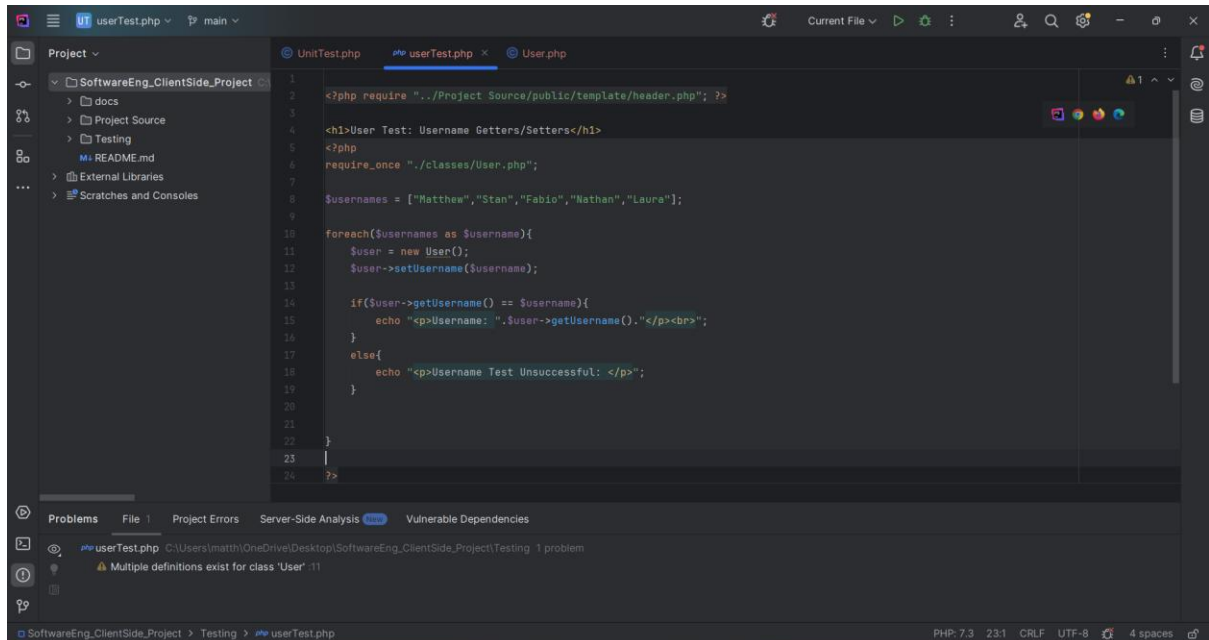
Author: **B0016388 Fabio Melilo**

Dated: 24/4/25

Unit Testing

User (getter/setters)

In user class test we take the getUsername and setUsername methods by creating an array of usernames and then in a for each loop creating a new user and setting their username as the same spot in the array



```
1
2 <?php require "../Project Source/public/template/header.php"; ?>
3
4 <h1>User Test: Username Getters/Setters</h1>
5 <?php
6 require_once "../classes/User.php";
7
8 $usernames = ["Matthew", "Stan", "Fabio", "Nathan", "Laura"];
9
10 foreach($usernames as $username){
11     $user = new User();
12     $user->setUsername($username);
13
14     if($user->getUsername() == $username){
15         echo "<p>Username: ".$user->getUsername()."</p><br>";
16     }
17     else{
18         echo "<p>Username Test Unsuccessful: </p>";
19     }
20 }
21
22
23
24 ?>
```

Problems | File | Project Errors | Server-Side Analysis | Vulnerable Dependencies

userTest.php C:\Users\matth\OneDrive\Desktop\SoftwareEng_ClientSide_Project\Testing 1 problem

Multiple definitions exist for class 'User' :11

SoftwareEng_ClientSide_Project > Testing > userTest.php PHP: 7.3 23.1 CRLF UTF-8 4 spaces

And we show it working in the browser like this:

User Test: Username Getters/Setters

Username: Matthew

Username: Stan

Username: Fabio

Username: Nathan

Username: Laura

Movie

For the movie test the method fetchAllMovies and this is done by connecting to the database checks if the data is received and then they are displayed

```
2  <?php require "../Project Source/public/template/header.php"; ?>
3
4  <h1>Movie Test: FetchAll Method</h1>
5  <?php
6  require_once "../classes/movieClass.php";
7  require_once "../src/DBconnect.php";
8
9  $movies = movieClass::fetchAllMovies($connection);
10
11 if(empty($movies)){
12     echo "<p>Failed to find Movies</p>";
13 }
14 else{
15     foreach($movies as $movie){
16         echo "<p>".$movie->getMovieId()."</p>";
17         echo "<p>".$movie->getTitle()."</p>";
18         echo "<p>".$movie->getGenre()."</p>";
19         echo "<p>".$movie->getTicketPrice()."</p>";
20     }
21 }
22
23 ?>
24 <?php require "../Project Source/public/template/footer.php";|
25
```

Movie Test: FetchAll Method

DB connected

1

Citizen Kane

Drama

19

2

Casablanca

Romance

12

3

The Godfather

Crime

15

4

Gone with the Wind

Epic

12

5

Booking

This is a similar test to the user test but instead of testing the username we're testing the card number

```
1
2 <?php require "../Project Source/public/template/header.php"; ?>
3
4 <h1>Booking Test: Card Getters</h1>
5 <?php
6 require_once "../classes/bookingClass.php";
7
8 $cards = ["11111111111111", "222222222222222222", "3333333333333333", "444444444444444444"];
9
10 foreach($cards as $cardNo){
11     $booking = new bookingClass(1, "2025-05-05", "A5", "Matthew", $cardNo, "D15DC1", "123", 10, 1);
12     if($booking->getCardNo() == $cardNo){
13         echo "<p>Card Number: ".$booking->getCardNo()."</p><br>";
14     }
15     else{
16         echo "<p>Card Test Unsuccessful: </p>";
17     }
18 }
19
20
21
22 ?>
23 <?php require "../Project Source/public/template/footer.php";
24
```

Booking Test: Card Getters

Card Number: 11111111111111

Card Number: 222222222222222222

Card Number: 3333333333333333

Card Number: 444444444444444444

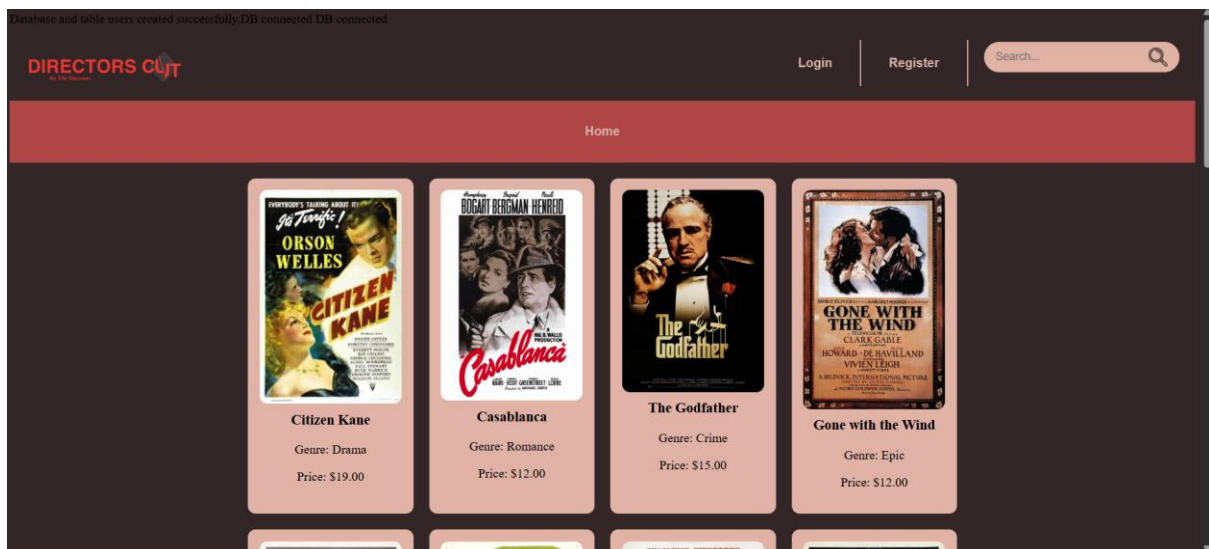
UI Testing

Consistency

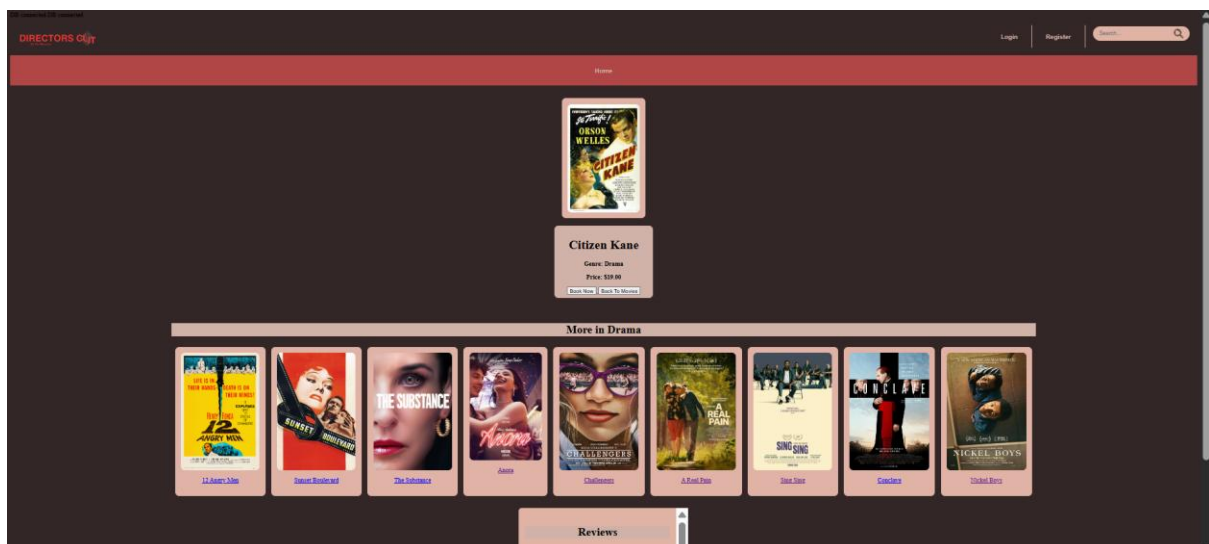
The same design across all pages.

Consistency is shown on the website has the same colour and style throughout the website and the navbar styling is consistent throughout.

Index Page



Products Page



Booking Page

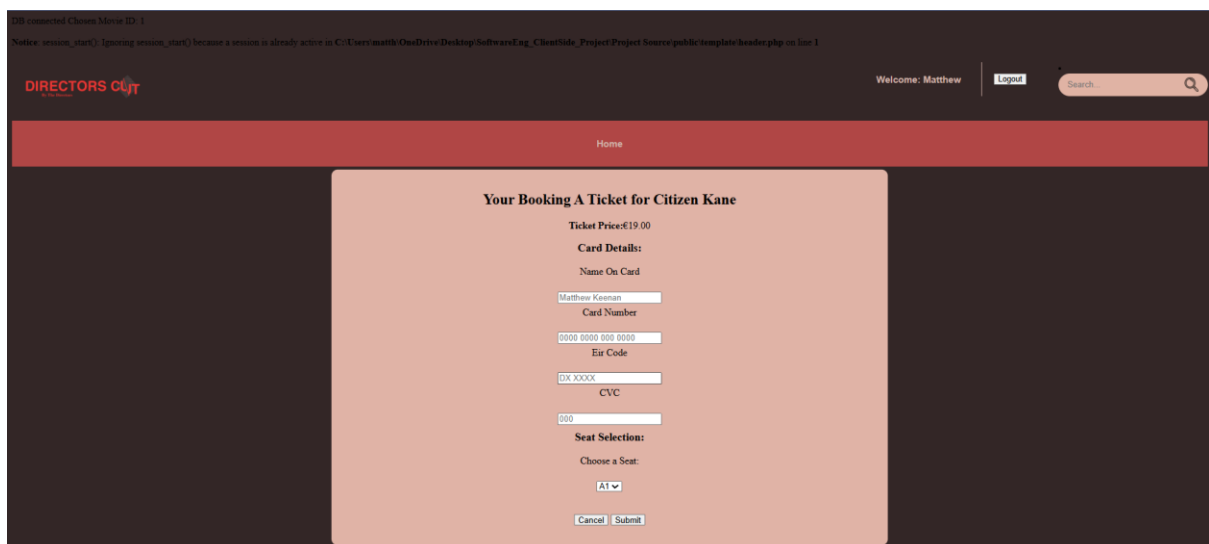
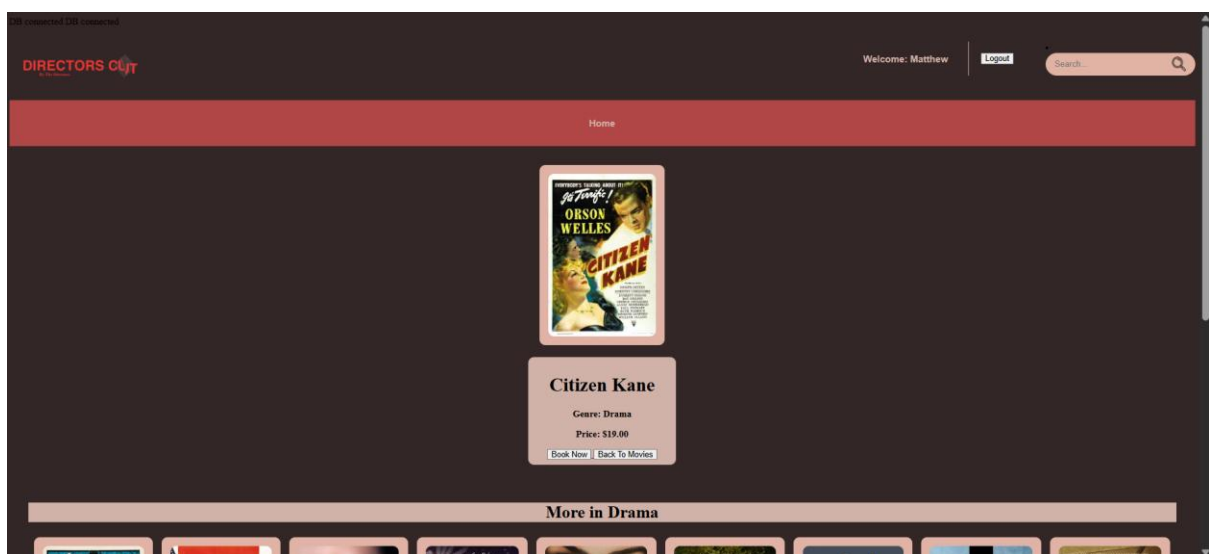
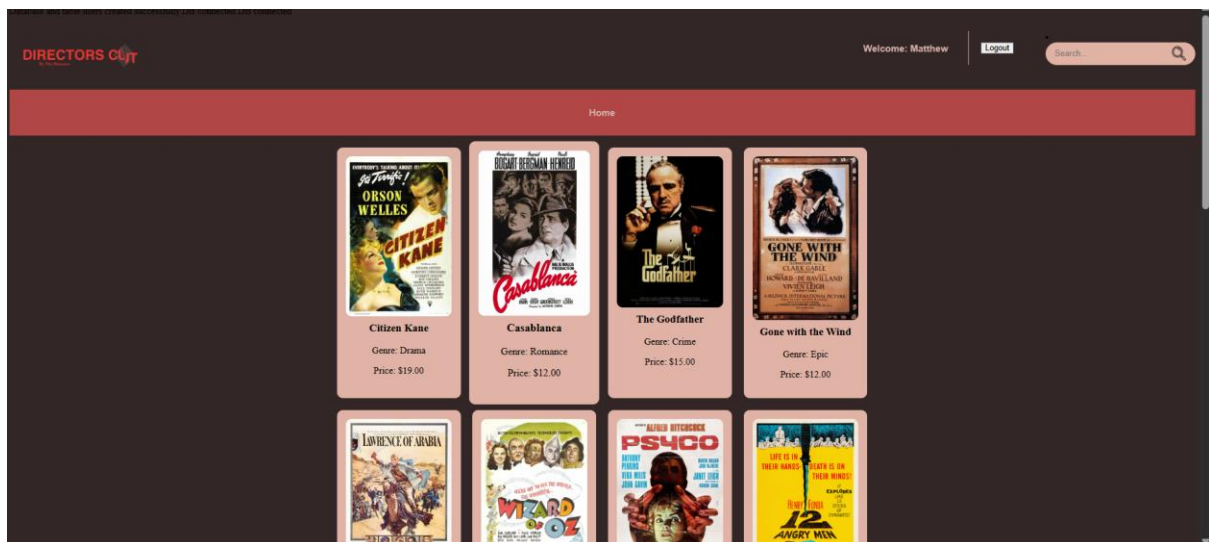
The screenshot shows the 'Directors Cut' website interface. At the top, there is a navigation bar with the site logo, a user greeting 'Welcome: Matthew', a 'Logout' button, and a search bar. Below the navigation bar is a red banner with the word 'Home'. The main content area features a light orange box titled 'Your Booking A Ticket for Casablanca'. Inside this box, the following information is displayed: 'Ticket Price: £12.00', 'Card Details:', 'Name On Card' (Matthew Keenan), 'Card Number' (0000 0000 0000 0000), 'Exp Code' (DX XXXX), 'CVC' (000), and 'Seat Selection:'. Under 'Seat Selection', there is a 'Choose a Seat:' dropdown menu currently set to 'A1'. At the bottom of the form are 'Cancel' and 'Submit' buttons.

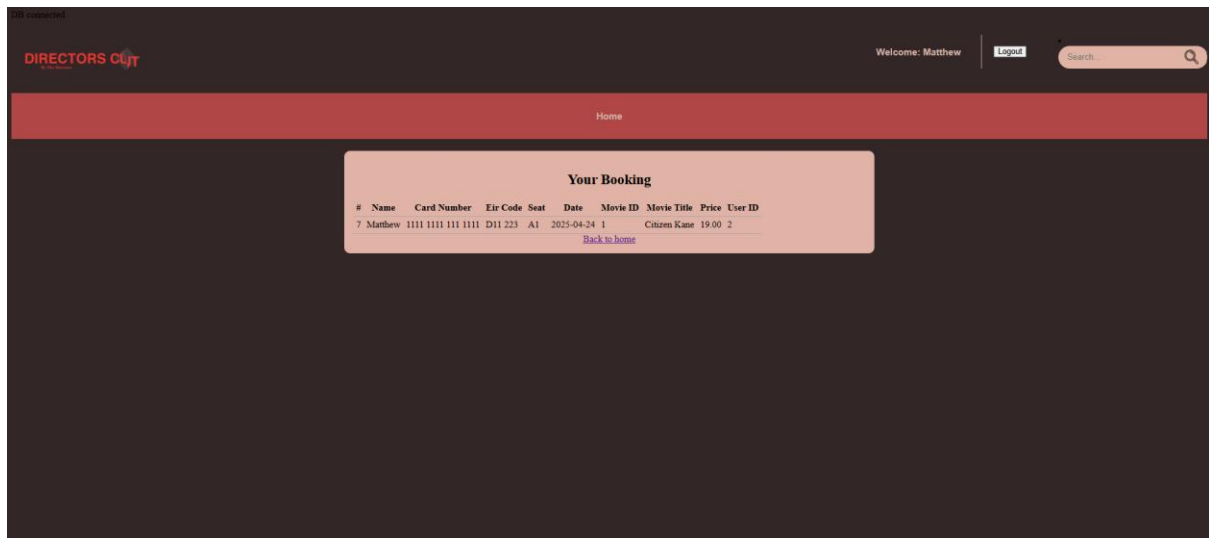
Minimal Clicks

Keeping the clicks required to complete the main function of the website e.g. booking a ticket or writing a review

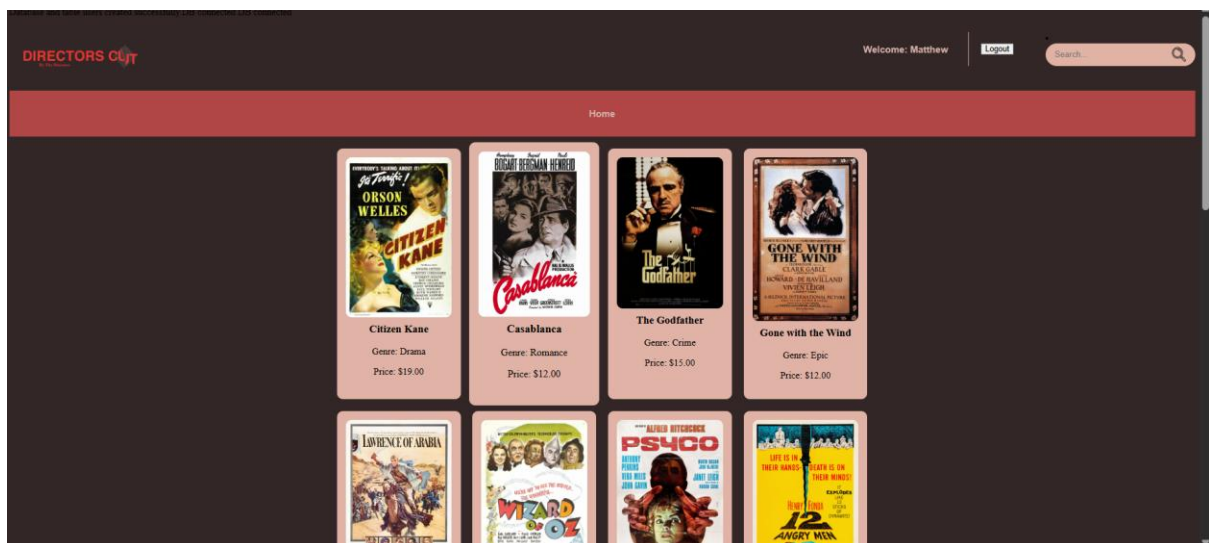
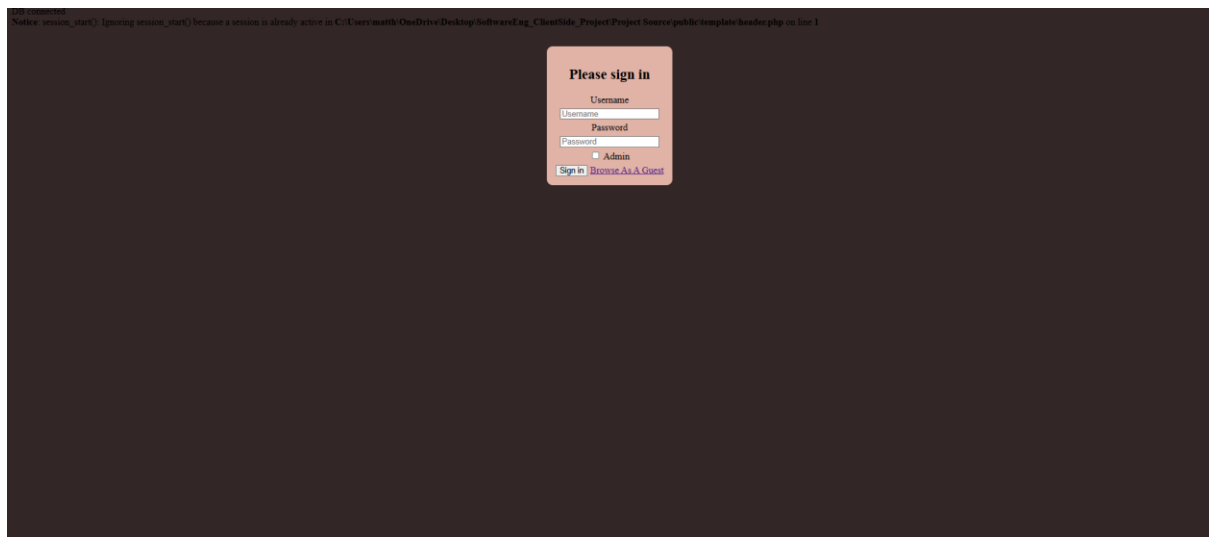
Booking takes five clicks as you must login, confirm, pick movie, click book, and click purchase.

The screenshot shows the 'Directors Cut' website interface with a login form. At the top, there is a navigation bar with the site logo, a user greeting 'Welcome: Matthew', a 'Logout' button, and a search bar. Below the navigation bar is a red banner with the word 'Home'. The main content area features a light orange box titled 'Please sign in'. Inside this box, there are two input fields: 'Username' and 'Password'. Below these fields are two buttons: 'Sign in' and 'Forgot As A Guest'. At the bottom of the form are 'Cancel' and 'Submit' buttons.





Were as making a review only takes four clicks as the reviews can be made on the product page




200 connected DBs connected

DIRECTORS CUT

Welcome: Matthew | Logout

Search

Home












Citizen Kane

Genre: Drama

Price: \$19.00

[Book Now](#) | [Back To Movies](#)


More in Drama




Price: \$19.00

[Book Now](#) | [Back To Movies](#)


More in Drama




[12 Angry Men](#)




[Sunset Boulevard](#)




[The Substance](#)




[Anora](#)




[Challengers](#)




[A Real Pain](#)



[Sing Sing](#)



[Conclave](#)



[Nickel Boys](#)

Reviews

Write your review here...

Score (1-10):

[Submit Review](#)

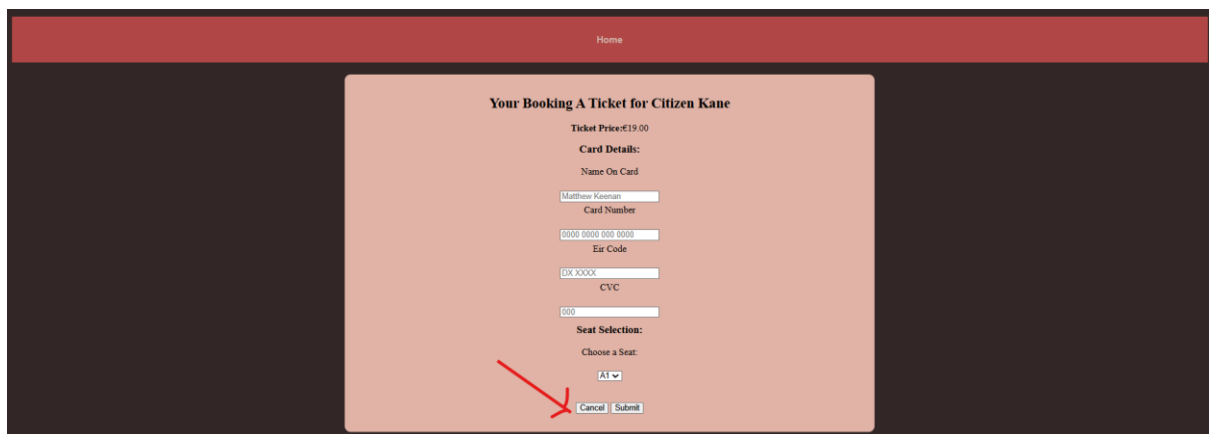
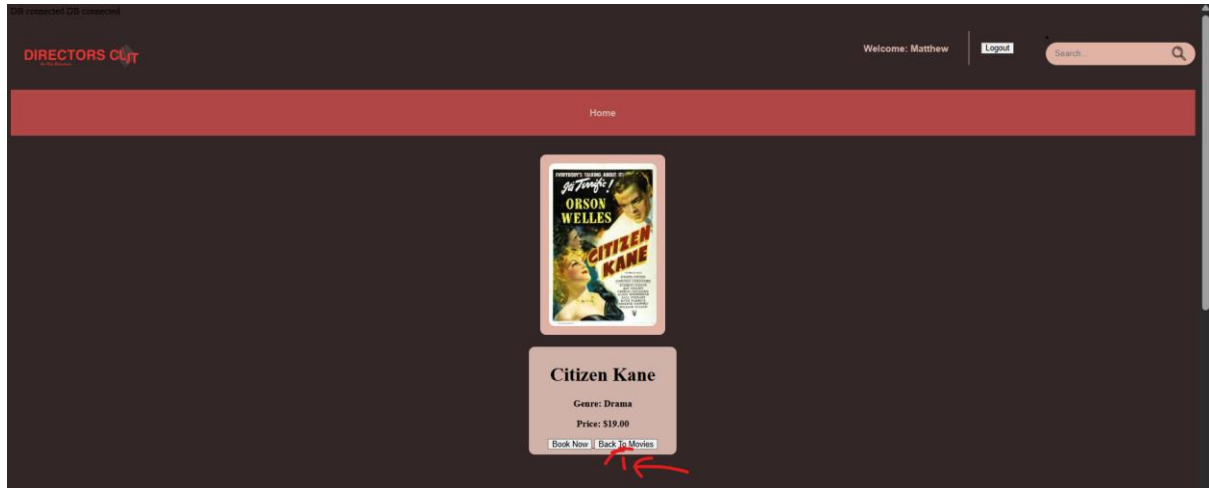
Anonymous rated it 10/10

never watched

Recoverability

Being able to undue mistakes made on the website e.g. going back to a previous page.

On the website recoverability is shown by each page's ability to go back to the home page for the movie page it's the back to movies button and cancel for booking.

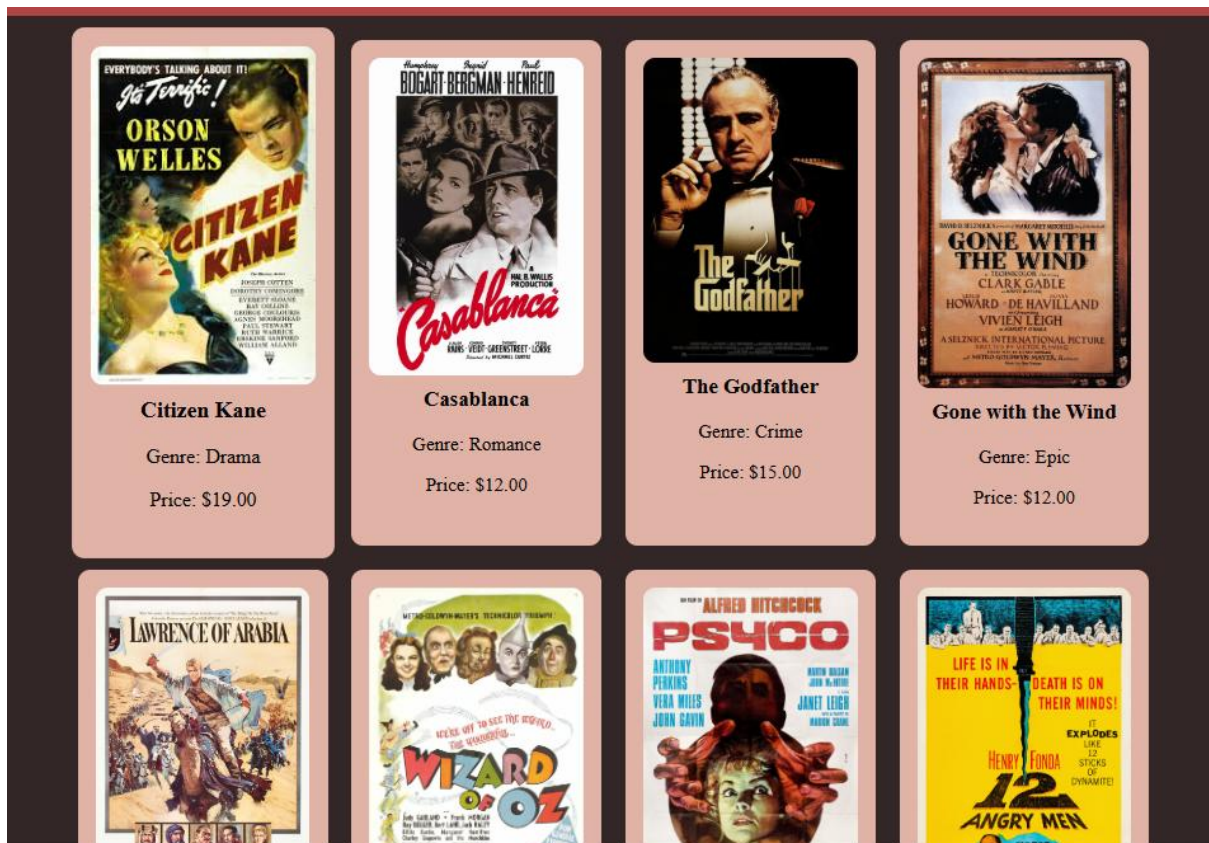


Guidance

The UI design assists users in how to use the website e.g. instructions displayed throughout on buttons like submit, back to home etc

The website uses guidance as the movies are front and centre on the main page and on the product page there are buttons to indicate that you should go back to movies or book a movie and a large banner showing more movie options

On The Booking page it guides the user by having placeholder to show what inputs expected as well as a submit button to tell the user what to do once the forms filled out



Citizen Kane

Genre: Drama

Price: \$19.00

Casablanca

Genre: Romance

Price: \$12.00

The Godfather

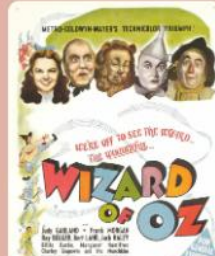
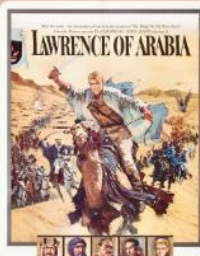
Genre: Crime

Price: \$15.00

Gone with the Wind

Genre: Epic

Price: \$12.00



Citizen Kane

Genre: Drama

Price: \$19.00

[Book Now](#) | [Back To Movies](#)

More in Drama



12 Angry Men



Sunset Boulevard



The Substance



Anora



Challengers



A Real Pain



Sing Sing



Conclave



Nickel Boys

Your Booking A Ticket for Citizen Kane

Ticket Price:€19.00

Card Details:

Name On Card

Matthew Keenan

Card Number

0000 0000 000 0000

Eir Code

DX XXXX

CVC

000

Seat Selection:

Choose a Seat:

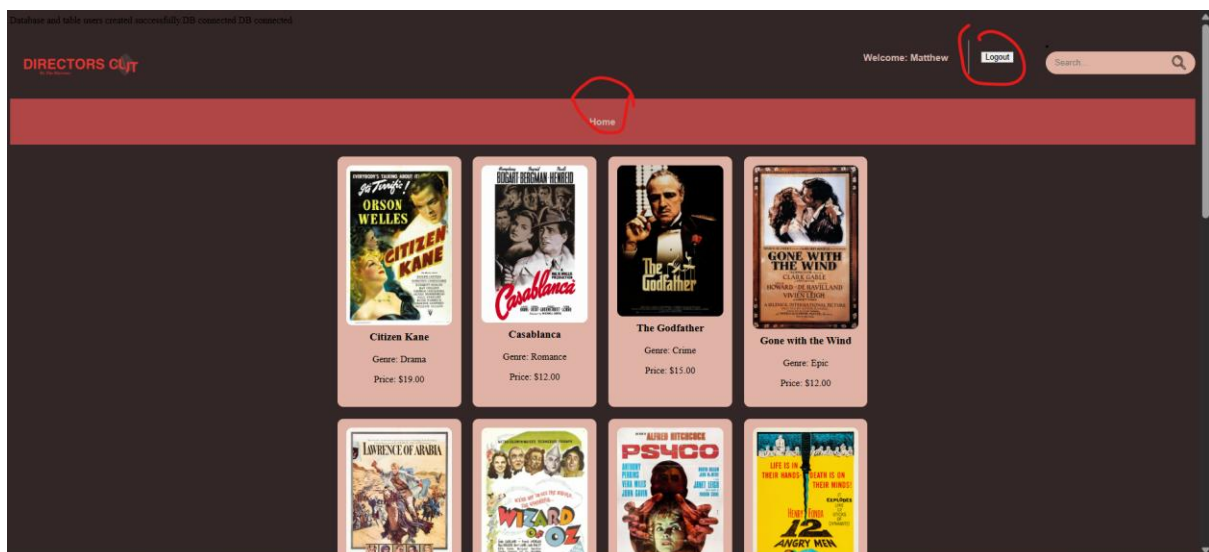
A1 ▼

Cancel Submit

Minimal Surprise

The user interface must not surprise the user. Everything should be as shown e.g a booking button under the godfather should bring the user to a booking ticket for the godfather and not another movie

The website uses minimal surprise as every input and button does exactly what is stated for example the back to movies button brings you back to the main page and submit buttons submit the forms users' input.





Your Booking A Ticket for The Godfather

Ticket Price: €15.00

Card Details:

Name On Card

Card Number

Eir Code

CVC

Seat Selection:

Choose a Seat:

Your Booking

#	Name	Card Number	Eir Code	Seat	Date	Movie ID	Movie Title	Price	User ID
8	Matthew	1111 1111 1111 111	D15 C878	A1	2025-04-24	3	The Godfather	15.00	2

[Back to home](#)

Validation Testing

This validation test uses the ContentModerator classes to filter a select of words if they appear in the inputted text.

```
1 <?php
2 include "Classes/ContentModerator.php";
3
4
5 $filtered = "";
6 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
7     $moderator = new ContentModeration();
8     $inputText = $_POST['usertext'];
9     $filtered = $moderator->filterWords($inputText);
10 }
11
12
13 <!DOCTYPE html>
14 <html>
15 <head>
16     <title>Content Moderation Test</title>
17 </head>
18 <body>
19 <h1>Test Content Moderation</h1>
20 <form method="post">
21     <label>Enter text:</label><br>
22     <textarea name="usertext" rows="5" cols="40"></textarea><br><br>
23     <input type="submit" value="Filter">
24 </form>
25
26 <?php if ($filtered != "") { ?>
27     <h2>Filtered Output:</h2>
28     <p><?php echo htmlspecialchars($filtered); ?></p>
29 <?php } ?>
30 </body>
31 </html>
```

Test Content Moderation

Enter text:

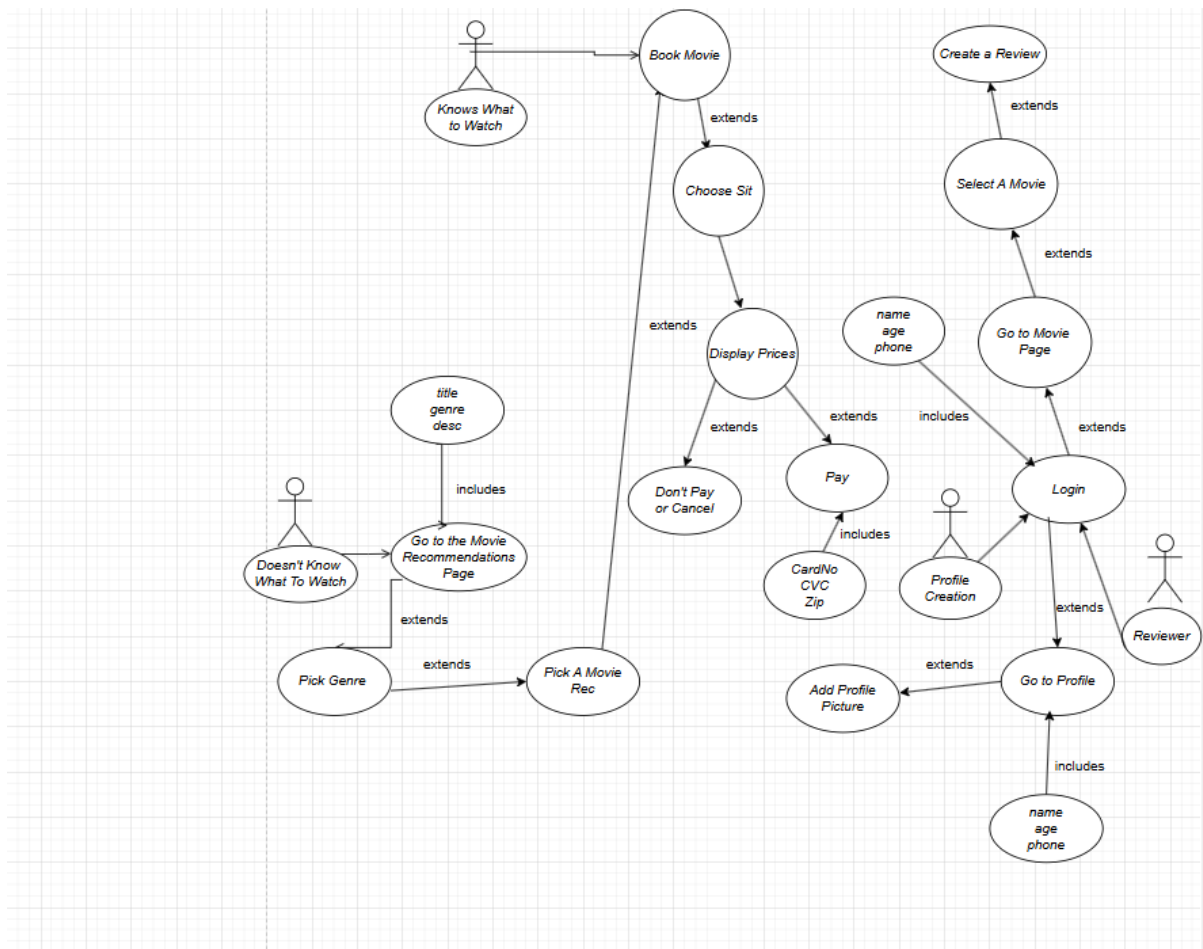
fuck

Test Content Moderation

Enter text:

Filtered Output:

Requirements Testing



Case 1: Reviewer

Login- Completed

Movie Page- Completed

Select A Movie- Completed

Create A Review- Completed

Case 2: Knows What to Watch

Books a movie – completed

Chooses seat- completed

Display price- completed

Pay/cancel- completed

Case 3: Doesn't Know What to Watch

Movie recommendation page- completed

Pick a genre- completed its based on search no though

Pick a movie- completed

Books a movie – completed

Chooses seat- completed

Display price- completed

Pay/cancel- completed