

WEB APPLICATION DEVELOPMENT ASSIGNMENT

Website Proposal & Report

THE GOLDEN TRIO

NSBM 20.1 PLY - SEM 2.3

WEB APPLICATION DEVELOPMENT ASSIGNMENT

WEBSITE REPORT

Website URL:

[http:// fantasycinema.infinityfreeapp.com](http://fantasycinema.infinityfreeapp.com)

GitHub Repositories:

<https://github.com/sayuru-akash/movieinfo-website>

<https://github.com/suvindi/movieinfo-website>

GROUP MEMBERS

1. *S.A.K. Amarasinghe – 22016368*
2. *P.P.S.W. Bandara – 18933*
3. *P.P. Rathnayake – 20119*

CONTENTS

<i>Introduction-----</i>	<i>4</i>
<i>Executive Summary-----</i>	<i>5</i>
<i>Solution Outline-----</i>	<i>6</i>
<i>Website Structure-----</i>	<i>7</i>
<i>Website GUI-----</i>	<i>8</i>
<i>Website Code-----</i>	<i>17</i>
<i>Website Database-----</i>	<i>25</i>
<i>Our Development Process-----</i>	<i>28</i>
<i>Pricing-----</i>	<i>29</i>
<i>Agreement-----</i>	<i>30</i>

INTRODUCTION

**Project Manager,
Fantasy Cinema,**

Thank you for the interest in developing your new movie website with us. With the experience and the knowledge gained from the university and a variety of online resources, we can provide you a quality website. While thanking you for this opportunity we gained, I would like to mention that our main goal is 100% customer satisfaction.

Our mission is to help our clients achieve their goals and to stand out at competitors on terms of website features, Website performance and delivering on time and on budget.

We have already designed and developed a sample website meeting your requirements. This sample website features a dark theme and an outstanding modern design so that you can stand out within the industry.

In this proposal, I have included all the details needed to understand our web development process, Identify the technology, Identify the features and clarify our transparent work procedure. You will also find project terms along with other project details.

As we appreciate the simplicity and clarity of our contracts, Necessary terms are included at the end of this proposal.

EXECUTIVE SUMMARY

A fully functional website build is needed to a grow and expose your cinema brand through social media and other online media platforms. This can include search engine optimization to reach higher in internet searchers, (ex: Google, Bing) and digital advertising. This will lead to an increased audience awareness, so we should be able to cater the website as per your requirements as well as your audience preferences.

We are proud to have this opportunity to be the web developers of a reputable company like you.

SOLUTION OUTLINE

As per your requirement of a website where users can view movie details such as movie title along with the release year, genre, synopsis, etc. At lengths regarding your requirements we have developed a sample website for you. We feel that this will improve your company brand status, expand your company in the digital market and increased the sales

We built this website using multiple languages such as HTML, CSS, CSS3, JS, JQUERY and PHP. These languages are used by more than a million websites around the world for its ease of accessibility and low resource utilization.

Our website will mainly allow your users to do the following once the website is released:

- Browse through the movies.
- View information about movies.
- Know movie ratings from trusted partners. (ex: IMDB)
- Read movie synopsis.
- Submit new movies.
- And more...

We recommend that your site be hosted with a “A” grade hosting provider using Apache, MySQL and an ftp server to ensure that your website is capable of performing at its best while being secure from malicious entities.

WEBSITE STRUCTURE

Your website currently has the following pages:

- Home
- About
- Contact
- Additional movie page 1
- Additional movie page 2
- Additional movie page 3
- Additional movie page 4
- Additional movie page 5

Pages with special features:

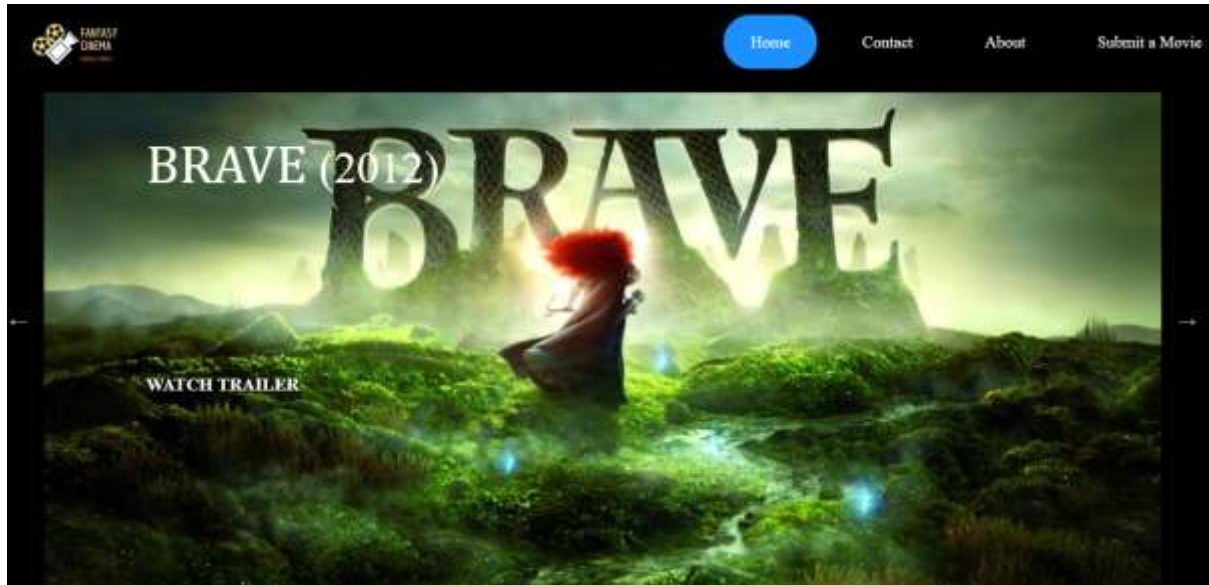
- Movie submission
- Submission result page
- Submission output page

Site integrations:

- CSS and JS files from cloud delivery networks. (CDN)
- JQUERY library support.
- MariaDB database.
- IMDB ratings HTML plugin.
- Social media buttons.

WEBSITE GUI

Home page:



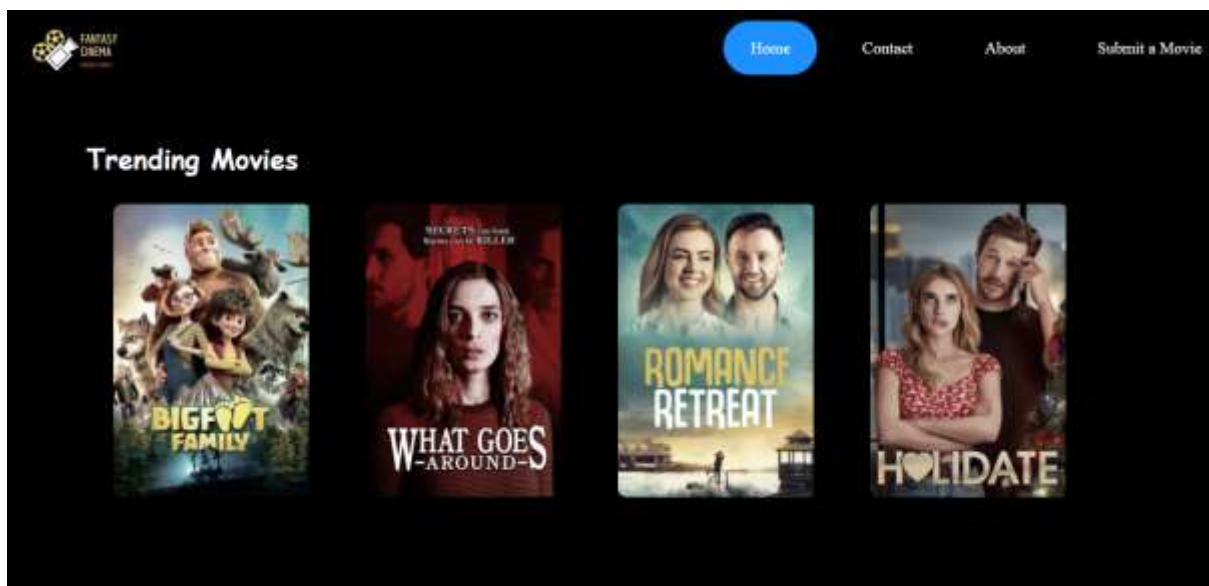
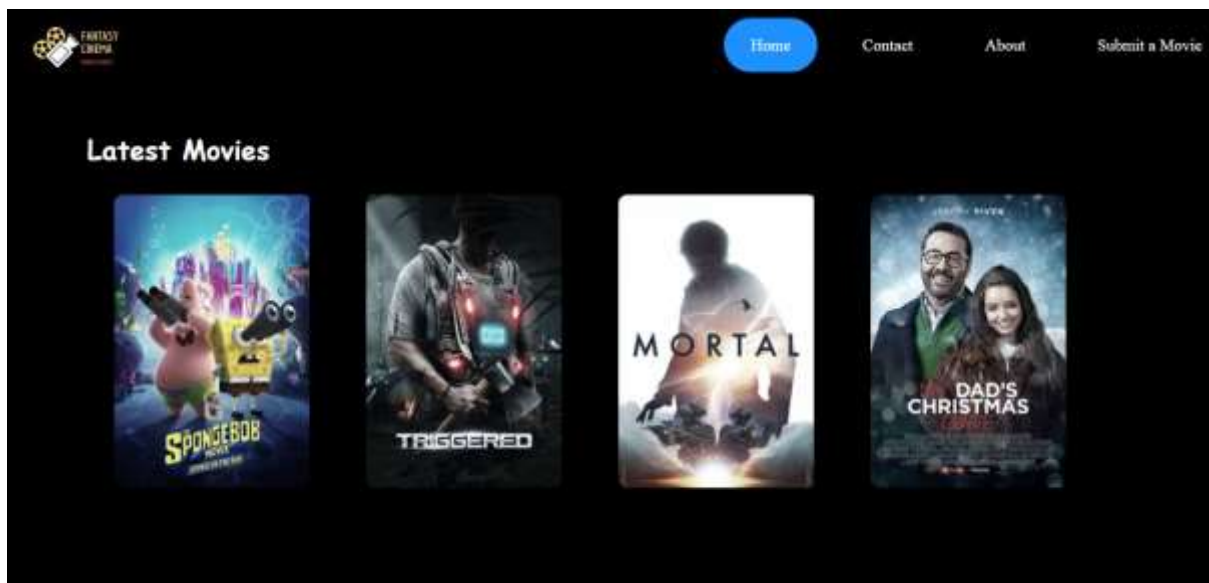
Above screen snip is the first view that your users will experience when they visit the website. This homepage is designed to attract your customers and improve your online presence.

Header:

Header is created as an element with “absolute” position, so that it stays on the top of the website while users scroll down. Header contains company’s logo and linked buttons to the main pages of the website such as “contact” and “about”.

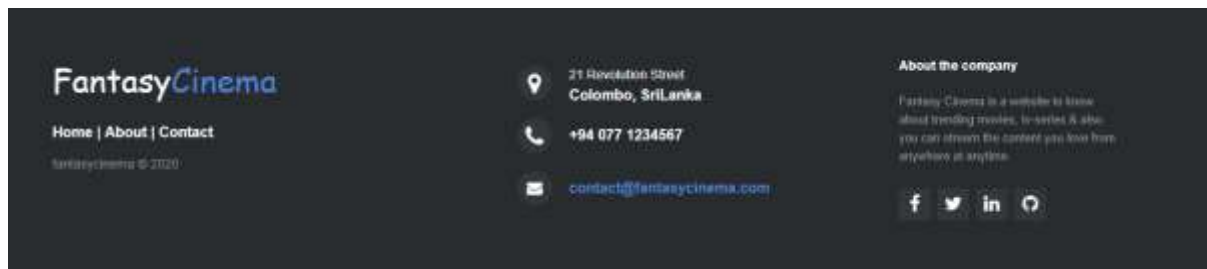
Image slider:

In this website, Image slider is designed to show the most attracted movie images as a carousel with links to the movie pages and to watch movie trailers directly. This carousel works automatically as well as can be clicked using the arrow buttons.



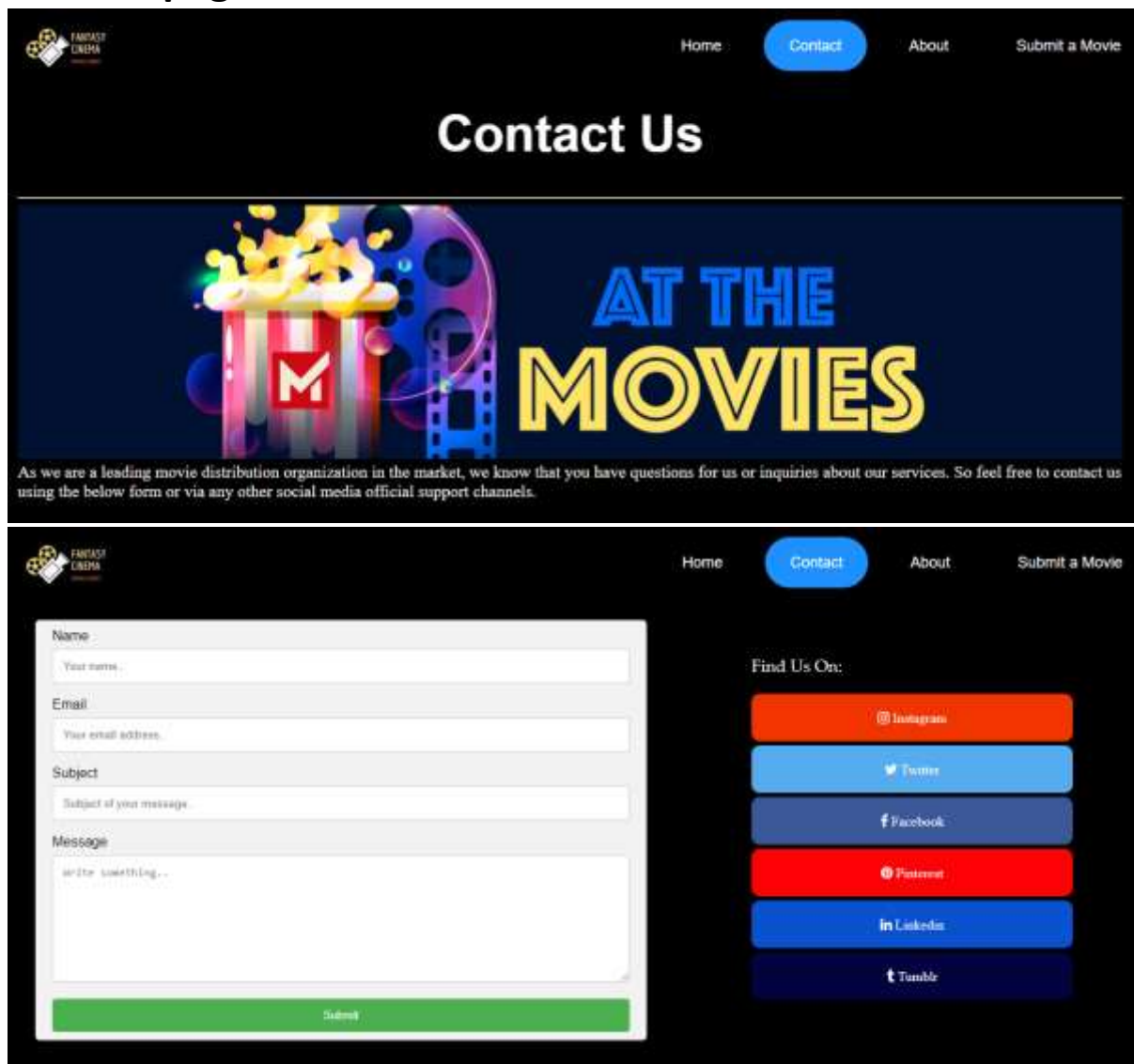
When you scroll down the homepage of the website, you will see small image divisions used to display movie posters by categorizing them into different sections. ex: latest movies, trending movies. The image divisions consist of on hover and on click animations while users are redirected to the IMDB official movie page when they click the button.

Footer:



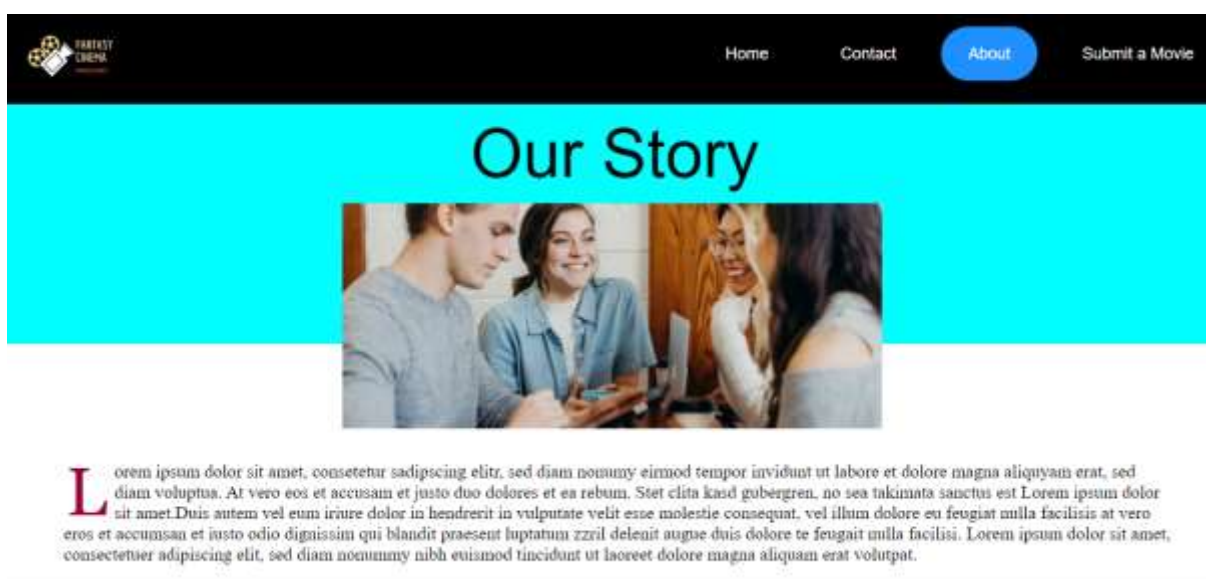
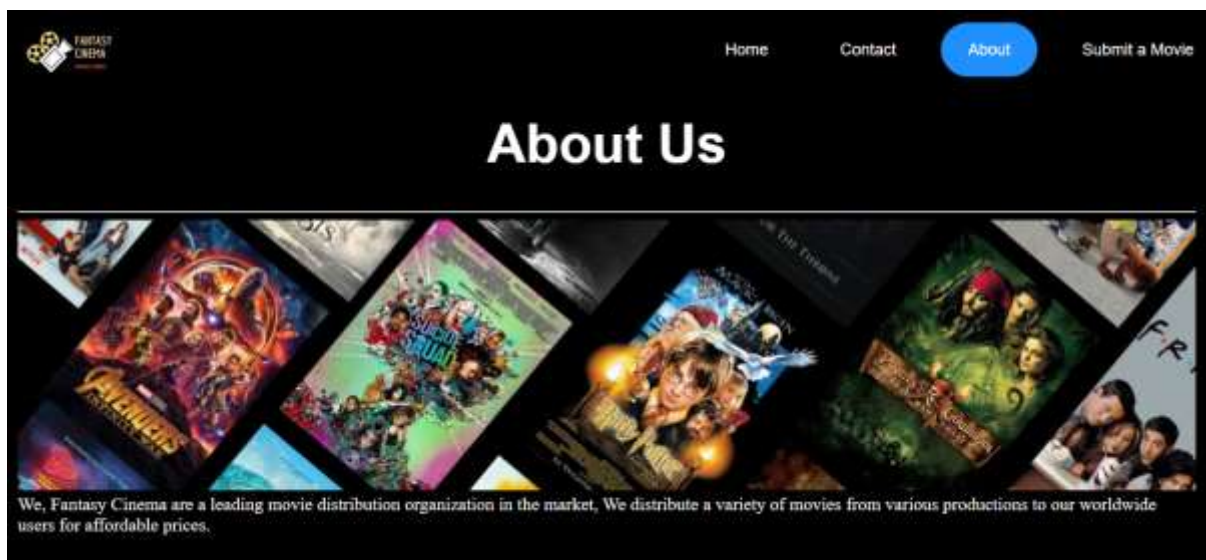
Website footer is available through out all the web pages. Footer consist of company branding and related information. Also, it has links to available web pages and company's official social media accounts.

Contact page:



The main aim of having a contact page is to create the ability of contacting your company for the users. This page highlights the contact options available (ex: social media) and encourages users to contact you using the contact form placed on the website. The contact form is protected using a script written in Js so that bots/users cannot make spam submissions.

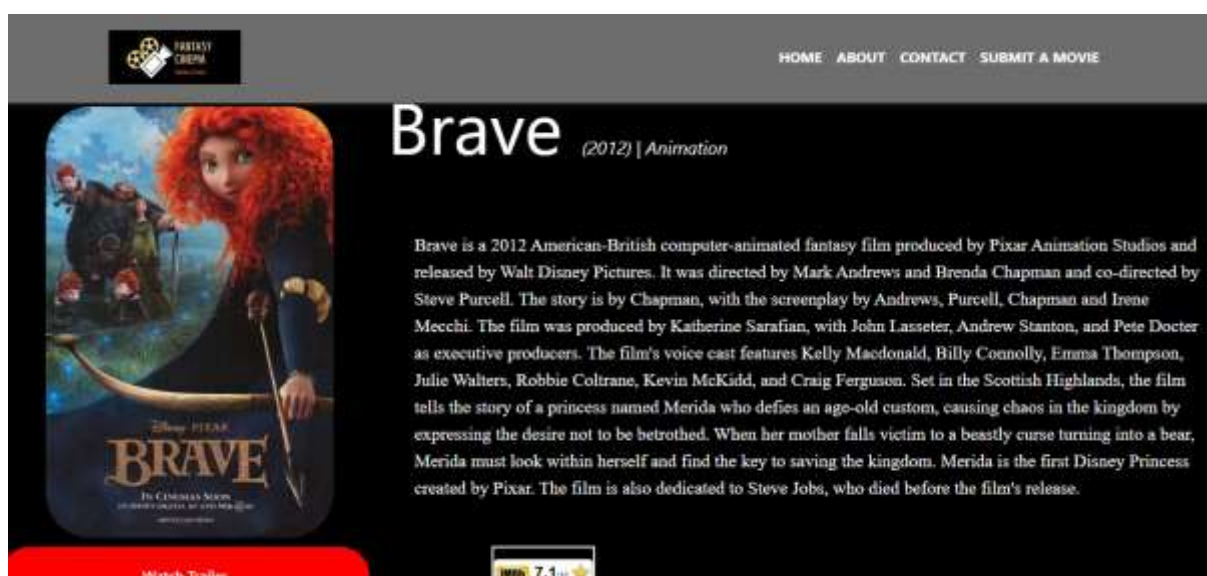
About page:



About page is a simple yet essential page of a website. You can display your company's history, work procedures and other information about your company in this page. This page has a materialistic look to stand out to the users. Photographs of the company and employees can be displayed in here.

Additional movie pages:


#1




#2



#3



HOME ABOUT CONTACT SUBMIT A MOVIE



Watch Trailer

The Avengers

(2012) | Sci-Fi Adventure

Marvel's The Avengers (classified under the name Marvel Avengers Assemble in the United Kingdom and Ireland), or simply The Avengers, is a 2012 American superhero film based on the Marvel Comics superhero team of the same name. Produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures,[N 1] it is the sixth film in the Marvel Cinematic Universe (MCU). Written and directed by Joss Whedon, it features an ensemble cast including Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth, Scarlett Johansson, and Jeremy Renner as the Avengers, alongside Tom Hiddleston, Clark Gregg, Cobie Smulders, Stellan Skarsgård, and Samuel L. Jackson. In the film, Nick Fury and the spy agency S.H.I.E.L.D. recruit Tony Stark, Steve Rogers, Bruce Banner, and Thor to form a team capable of stopping Thor's brother Loki from subjugating Earth.

IMDb 8.0

#4



HOME ABOUT CONTACT SUBMIT A MOVIE



Watch Trailer

The Dark Knight

(2008) | Sci-Fi Thriller

The Dark Knight is a 2008 superhero film directed, produced, and co-written by Christopher Nolan. Based on the DC Comics character Batman, the film is the second installment of Nolan's The Dark Knight Trilogy and a sequel to 2005's Batman Begins, starring Christian Bale and supported by Michael Caine, Heath Ledger, Gary Oldman, Aaron Eckhart, Maggie Gyllenhaal, and Morgan Freeman. In the film, Bruce Wayne / Batman (Bale), Police Lieutenant James Gordon (Oldman) and District Attorney Harvey Dent (Eckhart) form an alliance to dismantle organized crime in Gotham City, but are menaced by an anarchistic mastermind known as the Joker (Ledger), who seeks to undermine Batman's influence and throw the city into anarchy.

IMDb 9.0

#5



These additional movie pages are created to display information about the movies added in the website. This page features movie poster along with a button to watch the trailer as well as other information. This page also uses IMDB's official ratings html plugin to display movie ratings fetched directly from the IMDB "API"s.

Movie submission page:

A screenshot of the 'Submission Form' on the Fantasy Cinema website. The form is titled 'Submission Form' and includes fields for 'Movie Name', 'Released Year', 'Genre', 'Description', 'Poster', and 'Your Email'. A 'Choose File' button is next to the 'Poster' field, with a note: 'No file chosen' and '*only upload '.png', '.jpg' and '.jpeg' files'. There is a checkbox for 'I Agree to terms and conditions' and a red 'Submit' button at the bottom. A 'BACK TO HOME' link is visible in the top left corner.

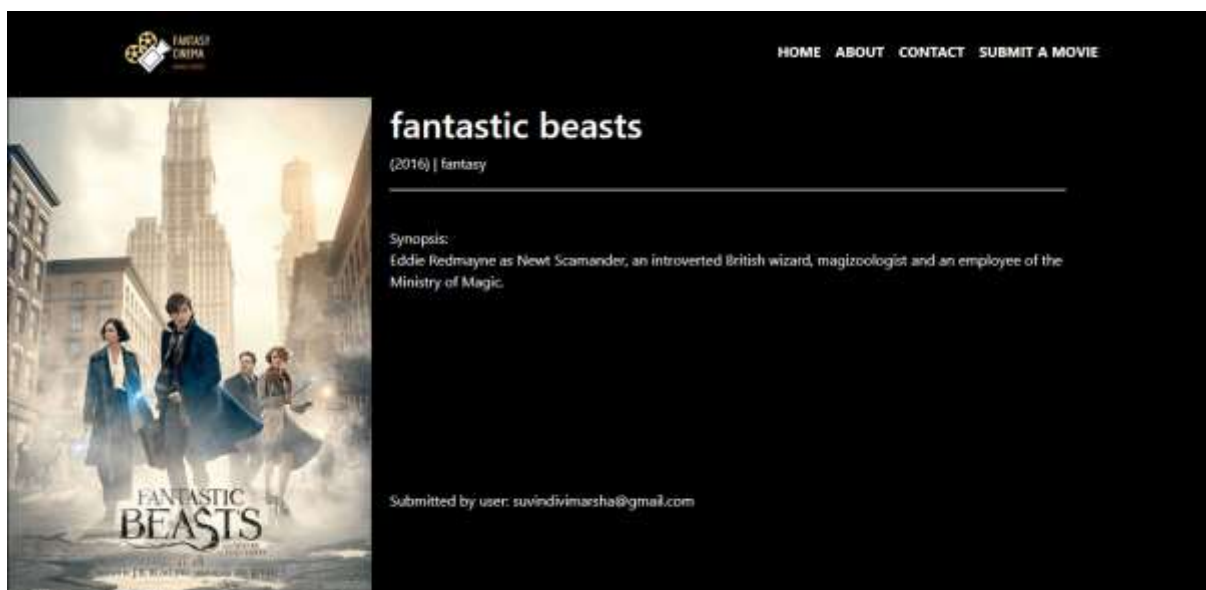
Movie submission page is a special page designed in a simple layout with a background image that suits the website's dark theme. Users can submit new movies along with their information. (ex:release year, genre, description, etc...) This submission form is protected with a javascript identify and block spammy submissions. The submitted form data is sent to the mariaDB server via a php using the "post" method.

Submission status pages:



After a user submit a movie or contact message via the submission form, these pages are generated with the result whether the submission was successful or not. This acts as a page used to contact with the server. User can view the submitted data by clicking the button placed at the center of the page after submitting a new movie or else the user can go back to the home page by clicking the relevent button.

Submission display page:



This page is generated according to the user's last data input using the submission form. This page clearly highlights the important data user added to the database in a simple view.

WEBSITE CODE

HTML

```
<html>
<head>
    //website title, website favicon & external css, js files are linked in here
</head>

<body>
    //html tags and data goes in the body. Also, scripts can be linked in here
</body>

</html>
```

- <!DOCTYPE html>
All HTML 5 documents should start with this declaration. This informs the browser which file type to expect.
- <html>
This is used to identify that the document is HTML. Language attribute is specified within this tag. (ex:<html lang="en">)
- <head>
It contains all the head elements such as title, meta, style, etc...
- <title>
This tag is used to define the documents title shown in a web browser's title bar.
- <meta>
It defines the meta data about the HTML file.
- <style>
The information about CSS documents is mentioned in here.
- <link>
Link tag is used to link JS files and other external documents.
- <script>
It is used to write scripts such as java script within these tags.

- `<body>`
The body contains almost all the contents of a HTML page. (ex: headings, paragraphs, images, hyperlinks)
- `<div>`
It creates a division or a section which can be managed separately with CSS or JS.
- ``
Image tag is used to link images to the html.
- `<a>`
Anchor tag (`<a>`) is used to define hypertext links in a html document.
- `<h1>-<h6>`
Heading tags are used to define headings in a website.
- `<p>`
This represent a paragraph in a html file.
- ``
It is used to markup a part of a text used to style later using CSS.
- `<header>`
It defines a section specifically for the header of a website.
- `<nav>`
It represents a page section providing navigation links.
- `<footer>`
It defines a section specifically for the footer of a website. It typically contains company information, contact information, site links, etc....
- ``
It represents items as an unordered list in a html.
- ``
Separate items in a list are specified within this tag.
- `
`
It is used to give a line break in a html document.
- `<form>`
It is used to create a html form where users can submit data.

- `<label>`
It creates a label for a group of elements.
- `<input>`
It declares an input field where users can enter data.
- `<textarea>`
A text area defines an input field which can hold an unlimited amount of characters.
- `<i>`
This tag is mainly used to display italic text.
- `<select>`
It is used to create dropdown lists in a form.
- `<option>`
This tag declares the elements inside a selection list.
- `<button>`
Clickable buttons can be created using this tag in html.
- “class” attribute
It specifies the class names of an element and used to point to the classes in css.
- “charset” attribute
It specifies the character encoding for the html document.
- “href” attribute
It is used to point a hyperlink.
- “rel” attribute
It is used to show the relationship between a linked document.
- “src” attribute
It specifies the link of an image.
- “type” attribute
It specifies the type of a specific element.
- “alt” attribute
It specifies the alternate text of an image.
- “name” attribute
It depicts the name of an input element.
- “placeholder” attribute
It is used to give a hint about the expected value of an input field.

- “id” attribute
Id attribute is used to give a unique id for an element.
- “action” attribute
This points where the form data should be sent after the user submits.
- “method” attribute
This attribute mentions how to send a form data (ex: get, post)
- “onsubmit” attribute
This attribute specifies an event that should be activated when a form is submitted.
- “required” attribute
It is Boolean attribute which specifies an input field as a required data field to submit.
- “value” attribute
It declares the value of an input element.

CSS

- position: This property is used to decide the position of the element
- margin: It decides the margin value of the element
- border: It decides the border value of the element
- padding: It generates spaces around the element
- font-family: It is used to select the font style
- font-size: Font size can be adjusted from this property
- font-weight: Font weight can be adjusted from this property
- color: Font color can be changed using this property
- opacity: It sets the opacity value of an element
- text-transform: It is used to control the capitalization in a text
- background-color: This is used to change the background color
- overflow: The contents that overflow from a division can be controlled
- min-width: Minimum width of an element can be specified
- max-width: Maximum width of an element can be specified
- min-height: Minimum height of an element can be specified

- max-height: Maximum height of an element can be specified
- float: This property specifies how an element should float
- text-align: Alignment of text can be specified
- display: It specifies how a section should behave in html
- object-fit: It specifies how an image should fit the container
- height: Specifies the height of an element
- width: Specifies the width of an element
- background-image: An image can be added to the background use in this attribute
- place-items: It allows you to align content within a container
- border-radius: It decides the radius of the element's corners
- text-decoration: It specifies how the decoration of the text should be place
- cursor: It specifies a specific cursor value to an element
- border-spacing: The space between two elements can be adjusted using this property
- top: It affects the vertical position of an element by changing the top difference
- bottom: It affects the vertical position of an element by changing the bottom difference
- left: It affects the horizontal position of an element by changing the left difference
- right: It affects the horizontal position of an element by changing the right difference
- transition: It specifies how smoothly should a transition occur
- filter: This property adds visual effects to the selected elements

JS

```
// JavaScript Document

function validate()      //A function is created and specified with a name
{
    //get the values user inserted in the form

    var name /*create a variable to store data*/ = document.getElementById("name").value; /*get data from the html document*/
    var year /*create a variable to store data*/ = document.getElementById("year").value; /*get data from the html document*/
    var genre /*create a variable to store data*/ = document.getElementById("genre").value; /*get data from the html document*/
    var description /*create a variable to store data*/ = document.getElementById("description").value; /*get data from the html document*/
    var email /*create a variable to store data*/ = document.getElementById("email").value; /*get data from the html document*/
    var poster /*create a variable to store data*/ = document.getElementById("poster").value; /*get data from the html document*/
    var agree /*create a variable to store data*/ = document.getElementById("agree").checked; /*get data from the html document*/
    var mailformat = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/; /*create a variable with the standard e-mail format*/

    /*name null validation*/
    if (name==""){
        alert("Name is Missing!");
        return false;
    }
    else if(name.length<2) /*name data validation*/
    {
        alert("Name must contain more than 2 characters!");
        return false;
    }

    /*year null validation*/
    else if (year==""){
        alert("Year is Missing!");
        return false;
    }
    else if((year.length<4) || (year.length>4)) /*year data validation*/
    {
        alert("Year must contain 4 Numbers!");
        return false;
    }
    else if (isNaN(year))
```

```

    {
        alert("Year must be in numeric form!");
        return false;
    }

    /*genre null validation*/
    else if (genre==""){
        alert("Genre isn't selected!");
        return false;
    }

    /*description null validation*/
    else if (description==""){
        alert("Description is Missing!");
        return false;
    }
    else if(description.length<30) /*description data validation*/
    {
        alert("Description is too short!");
        return false;
    }
    else if(description.length>250)
    {
        alert("Maximum description limit (250) exceeded!");
        return false;
    }

    /*email null validation*/
    else if (email==""){
        alert("Email is Missing!");
        return false;
    }
    else if (!email.match(mailformat)) /*email format validation*/
    {
        alert("Email is not valid!");
        return false;
    }

    /*poster null validation*/
    else if(poster==""){
        alert("poster is missing or invalid!");
        return false;
    }

    /*agree null validation*/
    else if (agree!=true){
        alert("You haven't agreed to the terms!");
        return false;
    }

```



```

    }
    else
    {
        return true; /*allow action after successful validation*/
    }
}

```

PHP

Accessing data sent to php document:

\$variablename=\$_POST["name"];

When data is sent to a php document using a form, the specific data can be accessed by `$_POST["data section name"]` and assigning the value after defining a variable name with the "\$" sign.

Displaying the output:

echo("value");

We use echo function to display and output in the html document using php. The data needed to be displayed can be either a php variable, html style or raw text.

Saving submitted files permanently:

```

<?php

$poster = $_FILES["poster"]["name"];
$target_dir = "upload/";

move_uploaded_file($_FILES['poster']['tmp_name'],$target_dir.$poster);

?>

```

When a file is sent from a html form, the file is saved only temporally. To save it in a permanent location we use "move_uploaded_file" command as given in the above code snippet.

WEBSITE DATABASE

Accessing MariaDB:

```
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

We can access the MariaDB database server using the server name and login credentials as given in the above php code

Creating a database:

```
// Create database
$sql = "CREATE DATABASE movieSubmission";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

We can create a database in the MariaDB server after connecting successfully. “CREATE DATABASE databasename” command is executed when performing this function.

Creating a table:

```
// create table
$sql = "CREATE TABLE newMovies (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50) NOT NULL,
year INT NOT NULL,
```

```

genre VARCHAR(50) NOT NULL,
description VARCHAR(300) NOT NULL,
poster VARCHAR(200) NOT NULL,
email VARCHAR(50) NOT NULL
);

if ($conn->query($sql) === TRUE) {
    echo "Table newMovies created successfully";
} else{
    echo "Error creating table: " . $conn->error;
}

```

After creating a database in the MariaDB server we can connect to the specific database and create a new table as displayed in the above code snippet to store data.

Adding data into a table:

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "movieSubmission";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Insert record
$sql = "INSERT INTO newMovies (name, year, genre, description, poster, email) VALUES ('$name', '$year', '$genre', '$description', '$poster', '$email')";

```

We can store data in the SQL database via using “INSERT INTO tablename(values)” command within the php document after accessing the correct database.

Getting data from a table:

```
$result = mysqli_query($conn,"SELECT * FROM newMovies ORDER BY id DESC LIMIT 1");

while($row = mysqli_fetch_array($result))
{

$name=$row['name'];
$year=$row['year'];
$genre=$row['genre'];
$description=$row['description'];
$poster=$row['poster'];
$email=$row['email'];

$image = $row['poster'];
$image_src = "../php/upload/".$image;
}
```

We can fetch data from the SQL database using “mysqli_query(select)” command and assign data to php variables.

DML Operations:

DML (Data Manipulation Language) statements are the elements used for data manipulation in SQL language. All DML operations such as add, change, retrieve and delete are used in this website database operations.

OUR DEVELOPMENT PROCESS

We take a professional approach when it comes to web designing. Our complete development process is built to ensure that every client is satisfied, and the project is delivered on time.

Our development process in a glimpse:

- Analyzing client need / organization.
- Initial planning of the UI.
- Generating graphics and mockups.
- Website development.
- Database management.
- Testing.
- Fixing bugs.
- Deployment.

PRICING

<i>Name</i>	<i>Price</i>	<i>QTY</i>	<i>Subtotal</i>
Initial Invoice	\$0.00	1	\$0.00
Approved Design Invoice	\$0.00	1	\$0.00
Final Invoice	\$0.00	1	\$0.00
		Subtotal	\$0.00
		Total	\$0.50

AGREEMENT

[Our Signature]

Signature

MM/DD/YYYY

[Client Signature]

Signature

MM/DD/YYYY