VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI - 590018

Mini Project Report

on

ONLINE MOVIE DATABASE

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by
DANIYAL PARVEEZ
1BG15CS026

Guide
Smt. Prarthana T. V.
Assistant Professor, Dept. of CSE
BNMIT

Vidyaya Amrutham Ashnuthe

B.N.M. Institute of Technology

(Approved by AICTE, Affiliated to VTU, ISO 9001:2008 certified and Accredited as grade A Institution by NAAC)
Post box no. 7087, 27th cross, 12th Main,
Banashankari 2nd Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in www. bnmit.org

Department of Computer Science and Engineering

2018 – 2019

B.N.M. Institute of Technology

Approved by AICTE, Affiliated to VTU, ISO 9001:2008 certified
and Accredited as grade A Institution by NAAC)
Post box no. 7087, 27th cross, 12th Main,
Banashankari 2nd Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in www. bnmit.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vidyaya Amrutham Ashnuthe

CERTIFICATE

Certified that the Mini Project entitled **Online Movie Database** carried out by Mr. **Daniyal Parveez** USN **1BG15CS026**, a bonafide student of VII Semester B.E., **B.N.M Institute of Technology** in partial fulfillment for the Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING of the **Visvesvaraya Technological University**, Belagavi during the year 2018-19. It is certified that all corrections / suggestions indicated for internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Web Technology Laboratory with Mini Project prescribed for the said degree.

Smt. Prarthana T. V.
Assistant Professor
Department of CSE
BNMIT, Bengaluru

Dr. Sahana D. Gowda
Professor and HOD
Department of CSE
BNMIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project.

I would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMIT, Bengaluru for providing excellent academic environment in the college.

I would like to sincerely thank Prof. **T J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of the work.

I would like to express my gratitude to **Dr. M S Suresh**, Dean, BNMIT, Bengaluru for his relentless support, guidance and assistance.

I would like to thank **Dr. Krishnamurthy G N**, Principal, BNMIT, Bengaluru for his constant encouragement.

I would like to thank **Dr. Sahana D Gowda**, Professor and Head of the Department of Computer Science and Engineering, BNMIT, Bengaluru who has shared her opinions and thoughts which helped me in giving my presentation successfully.

I extend my heartfelt, sincere gratitude to **Smt. Prarthana T V**, Assistant Professor, Department of Computer Science and Engineering, BNMIT, Bengaluru for her guidance and support.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, teaching & non-teaching staffs of department and all my friends, for giving me valuable advices and support at all times in all possible ways.

ABSTRACT

Online Movie Database, otherwise referred to as MovieDB, is a website that features a large collection of movies. Influenced by popular websites such as IMDb, the website provides the user with more than just information about movies; they get to share their opinions with other people. It is a platform where the user can engage not just with the content but also the people using the website.

Getting recommendations has become a popular trend especially when the consumer needs to make an investment of some sort, through time or money or some other resource. Considering the nature of entertainment media, the investment is worth it only when the user achieves a level of satisfaction from it. Thus, it is not uncommon for a person to research what would be a good book to read or a movie that would be worth watching when they have several choices.

For some people, MovieDB can work out as just a platform that answers their factual queries, such as which actor played a particular role in a movie. For all other purposes, MovieDB can become a strong tool that addresses a variety of needs. It can provide the user with suggestions, either through built-in modules or through the user's own exploration of the reviews section on movie pages. They can come across perspectives, trivia or revelations they would otherwise not come across from simply watching the movie on their own, because the website encourages users to publish their own content.

While there are several improvements that can be made to MovieDB in its current state, it still has a lot to offer. Through a collection of facts and opinions, MovieDB seeks to deliver an immersive experience to the user.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

MovieDB is an online movie database that was inspired by websites like IMDb and Rotten Tomatoes. Users can browse through a collection of movies across various genres and languages, find information specific to a movie such as cast or release date and check out ratings and opinions of other users about a movie.

The website has a hybrid approach towards delivering content where information regarding movies are published and maintained by the administrators of the site, so users get to access factual information on a single page. Alongside, users get to publish their own content in the form of reviews and make their opinion public. By allowing users to rate and review movies, they can not only track a list of movies they have watched but also build a watchlist by reading content posted by other users.

Users of the website can choose to browse with or without an account. Without having an account, the user can still view most of the static content on the website such as information about movies and sections on the homepage. This encourages the user to try out the website before having to create an account. If the user creates an account, they can get personalized content delivered through the information maintained about them and their activity and leave behind reviews for movies. To protect the privacy of an individual, only users with an account get to view profile information of other users along with a history of all reviews written by that user.

MovieDB was built using modern web technologies and features a rich and responsive interface. Having a responsive interface lets a person browse websites regardless of the hardware they are using to view it, as the layout of elements on a page dynamically adjusts itself based on the viewport settings of the device. By utilizing validation techniques on user input, the website is robust against injection attacks that may be performed by malicious users. Through client-side scripting, the user can get continuous feedback on the operations performed on the input fields. Certain content on the website are adjusted based on their activity.

1.2 PROBLEM STATEMENT

Users with a common interest tend to seek a platform for sharing their views on a subject. On that note, MovieDB provides a platform for people passionate about movies to view information and share their opinions about movies with other users of the website. By aiming to deliver more than just a static collection of facts, this website makes an attempt to be the one-stop solution for all movie-related needs of a user.

1.3 MOTIVATION

Social websites where user generates the content have become popular in recent years where user is no longer restricted to viewing fixed content that is maintained by a few administrators of the site. Instead, users get to post anything they want that is within the policies of the website and they also get to choose what they want to see. Advancements in artificial intelligence has enabled personalizing of content for users at higher levels of granularity, where personal details and browsing habits become the parameters for fine tuning what the user sees on their feed. MovieDB does not democratize the content production as factual information on websites are always maintained by administrators or a group of trusted individuals who actively participate in upholding the content quality. However, it does grant users the freedom to express their opinions and come across other people who share their views. The user gets suggestions through the active participation of the community which encourages involvement and engagement. The goal of this project is to deliver a social experience for people who love movies while not disrupting the collection of facts on the website.

1.4 WEB TECHNOLOGIES

Web technologies refers to the collection of technologies that primarily make up the World Wide Web. This is comprised of various languages that are used for markup, scripting and styling, protocols for communication and standards for developing webpages. The major languages that constitute a website today are written using Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript. There are several frameworks built around these technologies that make web development easier.

The most commonly used markup language is HTML, which uses tags to annotate text so that a computer can then manipulate the text. Most markup languages are human readable and use annotations that are distinguishable from the annotated text. There are many kinds of markup languages, but all are consistent in the way in which they annotate documents.

HTML is the conventional markup language used to create and edit web pages and web applications. It is used for creating the basic structure of the website. It consists of different elements preceded by an opening tag and appended with a corresponding closing tag. Many versions of HTML have been published since its inception, with XHTML and HTML5 being the more recent and commonly used standards. XHTML fused HTML and eXtensible Markup Language (XML) into a language that was very precise but soon came to be considered a tedious and difficult language. HTML5 removed some of the tedium and severity of HTML, while keeping its ability to remain precise and detailed.

CSS is a style sheet language standard set by World Wide Web Consortium (W3C) used to create and edit the visual presentation of web pages. CSS allows web developers to isolate a web page's content and visual styles into separate documents and gives better page layout control. An external CSS sheet is generally linked to HTML and along with JavaScript, it plays a vital part of technology used by most interfaces for websites. This is also used in interfaces for mobile devices making the websites more engaging.

Scripting languages may perform either client-side scripting or server-side scripting. Client-side scripting refers to programs that are executed by the user's web browser, enabling web pages to be scripted. This means that such scripts do not have to be run on the server and the user's web browser must support them to run them. Server-side scripts run on the web server which produce a unique response for each user's request to the website.

JavaScript is a scripting language that is used commonly to add client-side behavior for interaction and handling server responses. However, web pages are not the only place where JavaScript is used. Server programs and even some databases use JavaScript as their programming language. JavaScript is also used in PDF documents, game development and desktop and mobile applications.

HyperText Transfer Protocol (HTTP) is the protocol used by the World Wide Web that determines how messages are formatted and transmitted. It also directs web servers and browsers to what actions they should take in reaction to several commands. It is a request-response protocol where the client might request something, and HTTP allows the client to access the information. The web server directs requests received by client and transmits requested web page based on the HTTP command that is sent.

Worldwide Web Consortium is an international community of web members to meet the web standards. It was founded by Tim Berners-Lee, an inventor of the Web back in the 20th century. W3C is designed to reach a full potential of the Web and to make it accessible to all users from all over the world. Also, another aim for W3C was to make standards to maintain the growth of the Web in a single direction rather than splitting into competing groups.

1.5 APPLICATION OF WEB TECHNOLOGIES

- **Digital Advertising**: A solid revenue generator for web-based businesses, advertisers can move towards advertising paradigms, such as charging ads based on the amount of time the ad is visible on a user's screen or other interaction metrics.
- **E-Commerce**: It has been popular for engaging customers and performing customer interactions over the internet. Showing shoppers what other shoppers are looking at online or pushing out online deals directly to all connected browsers are the types of real-time features that e-commerce platforms will look to adopt in the future.
- **Publishing**: Many content producers find platforms like WordPress or Drupal for content management lucrative as it helps them build productive applications more easily.
- **Collaborative Works**: Applications such as Google docs have demonstrated the value of collaborative environments on the Web. These platforms help in connecting users together in constructive and insightful ways to add value to their workflows.

CHAPTER 2

SYSTEM REQUIREMENTS

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE AND SOFTWARE REQUIREMENTS

2.1.1 HARDWARE REQUIREMENTS

Processor: 3GHz dual-core or better

Memory: 4GB RAM or better

Storage: Upto 50MB for cookies and cached files

2.1.2 SOFTWARE REQUIREMENTS

Operating System: Windows 7/8/10

Web Server Package: XAMPP consisting of

- Apache HTTP Server
- PHP Hypertext Preprocessor (PHP)
- MySQL

Bootstrap 4.1.3 CSS Framework

jQuery 3.3.1 JavaScript Framework

Web Browser: Any browser with support for HTML5 and ES6/ES7 for JavaScript

CHAPTER 3

SYSTEM DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 PROPOSED SYSTEM

The website was built using PHP, with jQuery used for handling client-side interactions. Users have the option to sign in/register on the website or browse the website without using an account. For users without an account, sessions and cookies will not be used hence information generated for such users will be static. When a registered user logs in, their session information will be used to enable certain features such as recently viewed section or the account page.

Bootstrap and external CSS files were used for handling the presentation of the web pages. The interface elements are responsive, in the sense that they respond to the changes in the viewport settings of the device used to view the website. Thus, it is possible to browse the website not only on desktop browsers but also on small-resolution devices like tablets or phones without having to switch between different versions of the website.

jQuery/JavaScript was majorly used in handling the client-side validation of forms and displaying/hiding messages from the server. Messages for form validation appear close to their corresponding form fields for intuitive identification of errors, while also letting the user know more about the specific cause for error. Client-side validation also ensures that user does not have to re-enter details of the form on submitting details that might have errors, by allowing submission only when all the details are valid. Messages from the server are interpreted in PHP by reading the URL and showing messages for completed/failed actions.

All the web pages also make use of header redirects and reading session/cookie information. Thus, if a user visits pages directly using URL that they are not supposed to, while being signed in or signed out, they will be redirected to the appropriate web page automatically. Session information can be stored in a cookie which means the user can have persistent information if they want to resume a session at a different time.

3.2 FLOW OF WEB PAGES

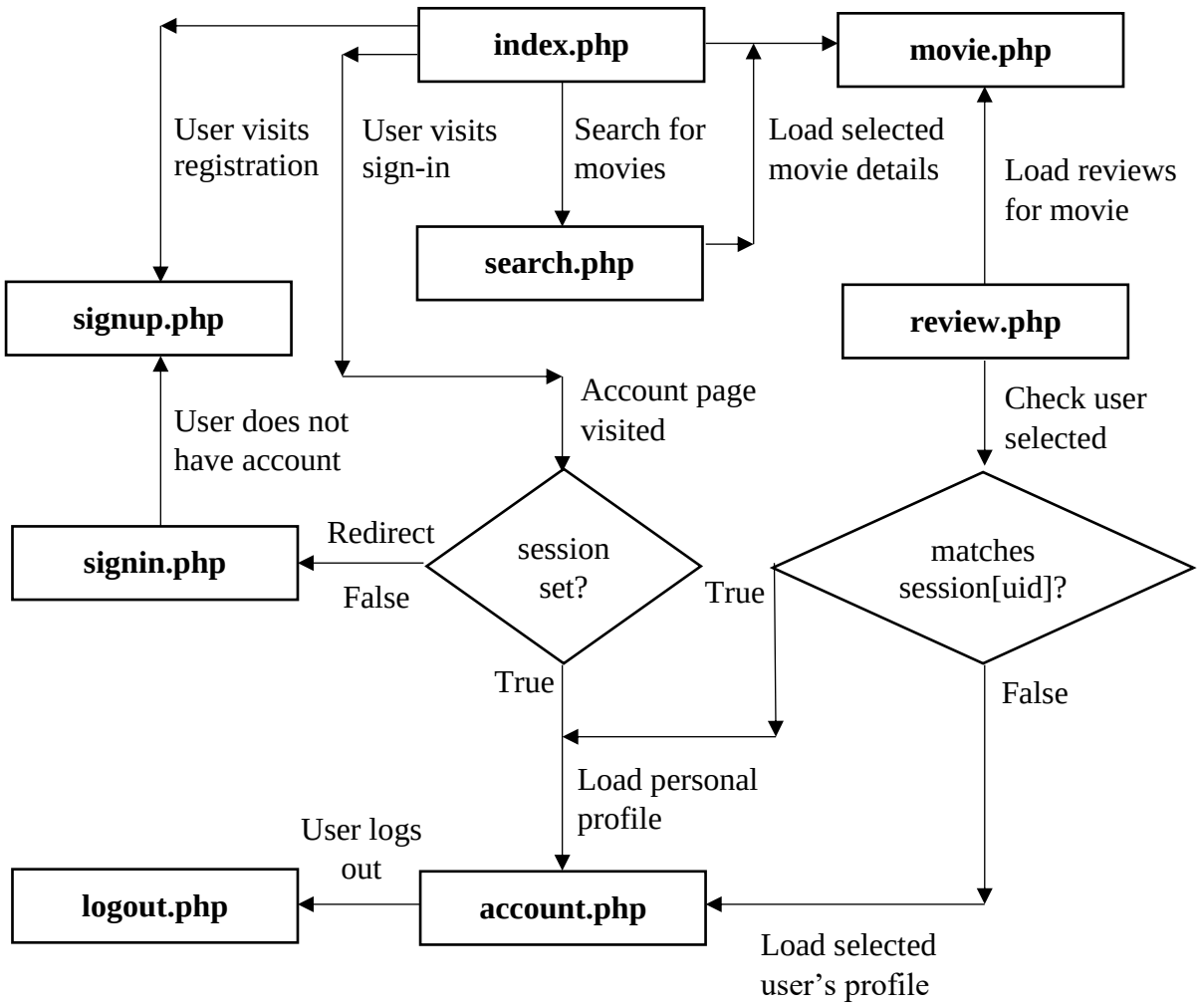

Fig 3.1 Flow Diagram of the Website

Flowchart Description:

Index.php is the home page of MovieDB and a portal which connects to several other pages. The user is shown options to sign-in/register if they are not logged in otherwise the options to view account and logout are displayed. Selecting any of the movie titles from the home page or search results will take the user to the movie.php page which loads contents based on the movie selected. Reviews of the selected movie are also loaded into the same web page. If the person browsing the website is logged in, they can select a user name from the reviews which opens the profile of that user.

CHAPTER 4

IMPLEMENTATION

CHAPTER 4

IMPLEMENTATION

4.1 MODULE DESCRIPTION

A modular approach is taken in building the application. Depending on functionality, the web application can be divided into several major modules as follows: -

1. **Login and Registration module:** All website visitors must be registered to leave behind ratings and reviews, and to view other users' rating and review history. Already registered users can login by supplying the requisite credentials (e-mail and password) at the login page. The credentials are verified and an error message is displayed in case of a wrong login attempt.

   Unregistered users can register themselves via the registration page by filling in the form. Details are verified before registering the user. In case the email the user is trying to register with already exists in the database, an error message noting the same is displayed.
2. **Account Management module:** All registered users can access their account from the "My Account" link displayed on the header of each web page. It allows the users to view and change their personal details and login credentials. Credentials and details will once again be verified before the user's input is accepted and any error alerts will be displayed.

   Ratings and reviews that the user has left behind of movies will also be visible in descending order of the date the review was left behind.
3. **Chosen Movie Collections module:** The index page displays a few movies from each of several different categories like "Coming Soon", "Recently Releases", "Modern Picks", "Classics".
4. **Search module:** The header of each web page consists of search bar that allows users to search for movies by entering their name partially or completely. Users can refine search by selecting additional parameters like "search in Classics", or "sort by title".
5. **Movie module:** Each movie has its own page that lists information about the movie including title, movie rating, movie runtime, release date, list of movie cast and crew, etc. Ratings and reviews left behind by users are also displayed.

6. **Review module:** This module allows registered users to leave behind rating and reviews on movies they haven't reviewed before.
7. **Recently Viewed module:** The website uses cookies to store view history of logged-in users. This view history contains up to 8 of the most recently viewed movies. This is visible at the bottom of each page, and is part of the website's footer.

4.2 SOURCE CODE

index.php

```
<html>
<head>
  <meta charset="utf-8" />
  <title>MovieDB</title>
  <link rel="stylesheet" href="./assets/css/bootstrap.min.css" />
  <link rel="stylesheet" href="./assets/css/header.css" />
  <link rel="stylesheet" href="./assets/css/main.css" />
  <link rel="stylesheet" href="./assets/css/footer.css" />

</head>
<body>
  <div class="container main-container">
    <?php include 'header.php'; ?>
    <main>
      <div class="container">
        <h2>Coming Soon...</h2>
        <?php
        $date =  strtotime(date("Y-m-d"));
        $result = mysqli_query($con, "SELECT m.movie_id,m.title,
m.release_date, mm.title_poster_path FROM movie as m, movie_meta as mm
where m.movie_id = mm.movie_id and unix_timestamp(release_date)
>".$date." limit 5");

        $ct =  mysqli_num_rows($result); //number of cards to be
displayed
        $num_cards = $ct;

        for($card_rows = 0; $card_rows < ceil($ct/3); $card_rows++)
        {
          echo '<div class="row">';
          for($cards= 0; $cards < min(3, $num_cards); $cards++){
            $row = mysqli_fetch_assoc($result);
            ?>
            <div class="col-md-4" >
```

```
              <div class="card" >
                <div class="rating-overlay">
                  <h3>Coming Soon</h3>
                </div>

                <img class="card-img-top " src="<?php echo
$row['title_poster_path']?>" title="<?php echo $row['title'];?>">
                <a href="movie.php?id=<?php echo $row['movie_id']?>">
                  <div class="card-body">
                    <h6 class="card-title"><?php echo
$row['title']?></h6>
                    <p class="card-text"><?php echo
'('.date("Y",strtotime($row['release_date'])).')';?></p>
                  </div>
                </a>
              </div>
            </div>
            <?php

          }
          $num_cards-=3;
          echo '</div>';
          echo '<br />';
        }
        ?>
        <h2>New Releases</h2>
        <?php
        $date =  strtotime(date("Y-m-d"));
        $query = "SELECT m.movie_id,m.title, m.release_date,
m.avg_rating, mm.title_poster_path FROM movie as m, movie_meta as mm";
        $query = $query." where m.movie_id = mm.movie_id and
unix_timestamp(release_date) <= ".$date." order by release_date desc
limit 5";
        $result = mysqli_query($con, $query);


        $ct =  mysqli_num_rows($result); //number of cards to be
displayed
        $num_cards = $ct;

        for($card_rows = 0; $card_rows < ceil($ct/3); $card_rows++)
        {
          echo '<div class="row">';
          for($cards= 0; $cards < min(3, $num_cards); $cards++){
            $row = mysqli_fetch_assoc($result);

            ?>
            <div class="col-md-4" >
```

```
            <div class="card" >

              <div class="rating-overlay">
                <h3>RATING:</h3>
                <?php if($row['avg_rating']==0){
                  echo "<h3>No ratings to show</h3>";
                }
                else {
                  echo "<h3>".$row['avg_rating']."/10</h3>";
                }?>
              </div>
              <img class="card-img-top " src="<?php echo
$row['title_poster_path']?>" title="<?php echo $row['title']?>">
              <a href="movie.php?id=<?php echo $row['movie_id']?>">
                <div class="card-body">
                  <h6 class="card-title"><?php echo
$row['title']?></h6>
                  <p class="card-text"><?php echo
'('.date("Y",strtotime($row['release_date'])).')';?></p>
                </div>
              </a>
            </div>
          </div>
          <?php
        }
        $num_cards-=3;
        echo '</div>';
        echo '<br />';
      }
      ?>

      <h2>Top Picks</h2>
      <?php
      $date =  strtotime(date("Y-m-d"));
      $query = "SELECT m.movie_id,m.title, m.release_date,
m.avg_rating, mm.title_poster_path FROM movie as m, movie_meta as mm";
      $query = $query." where m.movie_id = mm.movie_id and
unix_timestamp(release_date) <= ".$date." order by avg_rating desc
limit 5";
      $result = mysqli_query($con, $query);


      $ct =  mysqli_num_rows($result); //number of cards to be
displayed
      $num_cards = $ct;

      for($card_rows = 0; $card_rows < ceil($ct/3); $card_rows++)
      {
```

```
          echo '<div class="row">';
          for($cards= 0; $cards < min(3, $num_cards); $cards++){
            $row = mysqli_fetch_assoc($result);

            ?>
            <div class="col-md-4" >
              <div class="card" >

                <div class="rating-overlay">
                  <h3>RATING:</h3>
                  <?php if($row['avg_rating']==0){
                    echo "<h3>No ratings to show</h3>";
                  }
                  else {
                    echo "<h3>".$row['avg_rating']."/10</h3>";
                  }?>
                </div>
                <img class="card-img-top " src="<?php echo
$row['title_poster_path']?>" title="<?php echo $row['title']?>">
                <a href="movie.php?id=<?php echo $row['movie_id']?>">
                  <div class="card-body">
                    <h6 class="card-title"><?php echo
$row['title']?></h6>
                    <p class="card-text"><?php echo
'('.date("Y",strtotime($row['release_date'])).')';?></p>
                  </div>
                </a>
              </div>
            </div>
            <?php
          }
          $num_cards-=3;
          echo '</div>';
          echo '<br />';
        }
        ?>

        <h2>Classics</h2>
        <?php
        $date =  strtotime("2001/12/30");
        $query = "SELECT m.movie_id,m.title, m.release_date,
m.avg_rating, mm.title_poster_path FROM movie as m, movie_meta as mm";
        $query = $query." where m.movie_id = mm.movie_id and
unix_timestamp(release_date) <= ".$date." and m.avg_rating >= 7.5 order
by avg_rating desc limit 5";
        $result = mysqli_query($con, $query);
```

```
        $ct =  mysqli_num_rows($result); //number of cards to be
displayed
        $num_cards = $ct;

        for($card_rows = 0; $card_rows < ceil($ct/3); $card_rows++)
        {
          echo '<div class="row">';
          for($cards= 0; $cards < min(3, $num_cards); $cards++){
            $row = mysqli_fetch_assoc($result);

            ?>
            <div class="col-md-4" >
              <div class="card" >

                <div class="rating-overlay">
                  <h3>RATING:</h3>
                  <?php if($row['avg_rating']==0){
                    echo "<h3>No ratings to show</h3>";
                  }
                  else {
                    echo "<h3>".$row['avg_rating']."/10</h3>";
                  }?>
                </div>
                <img class="card-img-top " src="<?php echo
$row['title_poster_path']?>" title="<?php echo $row['title']?>">
                <a href="movie.php?id=<?php echo $row['movie_id']?>">
                  <div class="card-body">
                    <h6 class="card-title"><?php echo
$row['title']?></h6>
                    <p class="card-text"><?php echo
'('.date("Y",strtotime($row['release_date'])).')';?></p>
                  </div>
                </a>
              </div>
            </div>
            <?php
          }
          $num_cards-=3;
          echo '</div>';
          echo '<br />';
        }
        ?>
        <h2>You Recently Viewed</h2>
        <?php $num = 8;  //number of cards to display

        if(!isset($_COOKIE["VIEW_HISTORY"]))
        echo '<h5>You view history is empty.</h5>';
        else {
```

```
          $viewed = explode(",",$_COOKIE['VIEW_HISTORY']);
          ?>

          <?php
          foreach($viewed as $item)
          {
            $result = mysqli_query($con, "select mm.title_poster_path,
m.title from movie_meta as mm, movie as m where m.movie_id =
mm.movie_id and m.movie_id= '".$item."'");
            $row = mysqli_fetch_assoc($result);
            ?>
            <a href="movie.php?id=<?php echo $row['movie_id']?>"><img
class="card-img-top viewed-img" src="<?php echo
$row['title_poster_path']?>" alt="<?php echo $row['title']?>"
title="<?php echo $row['title']?>"></a>
            <?php
          }
        } ?>

      </div>
    </main>
    <?php include 'footer.php'; ?>
  </div>
  <script src="./assets/js/jQuery.js"></script>
  <script src="./assets/js/bootstrap.bundle.min.js"></script>
  <script>

  var posters = document.getElementsByClassName("rating-overlay");
  var imgs = document.getElementsByClassName("card-img-top");

  var showRating = function()
  {
    this.parentNode.childNodes[1].style.display = 'block';
  }
  var hideRating = function()
  {
    this.style.display = 'none';
  }
  for (var i = 0; i < imgs.length; i++) {
    posters[i].style.display = 'none';
    imgs[i].addEventListener('mouseenter',showRating);
    posters[i].addEventListener('mouseleave',hideRating);
  }
  </script>
</body>
</html>
```

account.php

```
<?php

$str = "";
    session_start();

    if(isset($_COOKIE["user"]))
        $_SESSION["user_id"] = $_COOKIE["user"];

    if(!isset($_SESSION["user_id"])) {
        header("Location: signin.php");
        die();
    }
    require "config/dbconfig.php";

    $conn =  new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD,
DB_DATABASE);
    if ($conn->connect_error)
        die("Connection failed: " . $conn->connect_error);

    $user = array();

    if(isset($_GET['user_id']))
        $user["user_id"] = $_GET['user_id'];
    else
        $user["user_id"] = $_SESSION["user_id"];

    $userSql = "SELECT * FROM user WHERE user_id = ".$user["user_id"];
    $userResult = mysqli_query($conn, $userSql);

    if($userResult) {
        $row = $userResult->fetch_assoc();
        $user["fname"] = $row["fname"];
        $user["lname"] = $row["lname"];
        $user["gender"] = $row["gender"];
        $user["country"] = $row["country"];
        $user["email"] = $row["email"];
        $password = $row["password"];
        mysqli_free_result($userResult);
    }

    if(isset($_POST["session"]) && !isset($_GET['user_id'])) {
        echo json_encode($user);
        die();
    }

    if(isset($_POST["password"]) && !isset($_GET['user_id'])) {
```

```
        if($_POST["password"] === $password)
            echo "Valid";
        else
            echo "This does not match your current password.";
        die();
    }

?>
<html>
    <head lang="en">
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <title>MovieDB | My Account</title>
        <link rel="stylesheet" href="assets/css/bootstrap.min.css">
        <link rel="stylesheet" href="assets/css/open-iconic-
bootstrap.min.css">
        <link rel="stylesheet" href="assets/css/account.css" />
        <script src="assets/js/jquery-3.3.1.min.js"></script>
        <script src="assets/js/bootstrap.min.js"></script>
        <script src="assets/js/account.js"></script>
    </head>
    <body>
        <div class="<!--vertical-center-->">
            <div class="container" id="accountContainer">
                <div class="row" id="siteLogo">
                    <h1><a id="siteLogoName"
href="index.php">MovieDB</a></h1>
                </div>

                <?php if(!isset($_GET['user_id'])){


                if(isset($_GET["status"]) && $_GET["status"] ==
"success"): ?>
                <div id="successAlert" class="row">
                    <div class="alert alert-success col" role="alert">
                        <h4 class="alert-heading">Success!</h4>
                        <p>Your account details were updated.</p>
                    </div>
                </div>
                <?php endif; ?>

                <?php if(isset($_GET["status"]) && $_GET["status"] ==
"failure"): ?>
                <div id="failureAlert" class="row">
                    <div class="alert alert-danger col" role="alert">
                        <h4 class="alert-heading">Error</h4>
```

```
                        <p>Unable to update details at the moment.</p>
                    </div>
                </div>
            <?php endif;
        if(!isset($_GET["user_id"]))
        $str = "My";
    else $str = "User"; }?>

                <div class="row">
                    <div class="col-md-4">
                        <div>
                            <h2><?php echo $str?> Account</h2>
                        </div>
                        <div id="accordion">
                          <div class="card">
                            <div class="card-header" id="headingOne">
                              <h5 class="mb-0">
                                <button class="btn btn-link" data-
toggle="collapse" data-target="#personalDetails" aria-expanded="true"
aria-controls="personalDetails">
                                  Personal Details
                                </button>
                              </h5>
                            </div>

                            <!-- PHP stuff from here -->
                            <div id="personalDetails" class="collapse
show" aria-labelledby="headingOne" data-parent="#accordion">
                              <div class="card-body">
                                <h4><?php echo $user["fname"]."
".$user["lname"]; ?></h4>
                                <h6><?php echo $user["gender"]; ?></h6>
                                <h6>Country: <?php echo
$user["country"]; ?></h6>
                                <?php if(!isset($_GET["user_id"])): ?>
                                <h6>E-mail: <?php echo $user["email"];
?></h6>
                                <?php endif; ?>
                              </div>
                            </div>
                          </div>
                            <?php if(!isset($_GET['user_id']))
                            {?>
                          <div class="card">
                            <div class="card-header" id="headingTwo">
                              <h5 class="mb-0">
```

```
                                <button class="btn btn-link collapsed"
data-toggle="collapse" data-target="#editInformation" aria-
expanded="false" aria-controls="editInformation">
                                  Edit Information
                                </button>
                              </h5>
                            </div>
                            <div id="editInformation" class="collapse"
aria-labelledby="headingTwo" data-parent="#accordion">
                              <div class="card-body">
                              <form method="POST"
action="accountUpdate.php" id="editInformationForm" novalidate>
                                <div class="form-group">
                                    <input type="text"
autocomplete="on" class="form-control" name="fname" id="fname"
placeholder="First Name" value="<?php echo $user["fname"]; ?>" required
/>
                                    <div class="feedback"></div>
                                </div>
                                <div class="form-group">
                                    <input type="text"
autocomplete="on" class="form-control" name="lname" id="lname"
placeholder="Last Name" value="<?php echo $user["lname"]; ?>" required
/>
                                    <div class="feedback"></div>
                                </div>
                                <div class="form-group">
                                    <div class="custom-control custom-
radio custom-control-inline">
                                        <input class="custom-control-
input" name="gender" type="radio" id="genMale" value="Male" required>
                                        <label class="custom-control-
label" for="genMale">Male</label>
                                    </div>
                                    <div class="custom-control custom-
radio custom-control-inline">
                                        <input class="custom-control-
input" name="gender" type="radio" id="genFemale" value="Female"
required>
                                        <label class="custom-control-
label" for="genFemale">Female</label>
                                    </div>
                                </div>
                                <div class="form-group">
                                    <select name="country" id="country"
autocomplete="on" class="custom-select" required>
                                        <option disabled selected
hidden value="">Select country</option>
```

```
                                        <option
value="India">India</option>
                                        <option value="United
States">United States</option>
                                        <option value="United
Kingdom">United Kingdom</option>
                                        <option
value="Canada">Canada</option>
                                        <option
value="Others">Others</option>
                                    </select>
                                </div>
                                <div class="form-group">
                                    <input type="text"
autocomplete="on" class="form-control" name="email" id="email"
placeholder="E-mail" value="<?php echo $user["email"]; ?>" required />
                                    <div class="feedback"></div>
                                </div>
                                <div class="buttonBox">
                                    <button type="submit" class="btn-
block red-button">Update Details</button>
                                </div>
                            </form>
                            </div>
                        </div>
                    </div>

                    <!-- END OF PHP -->

                    <div class="card">
                      <div class="card-header" id="headingThree">
                        <h5 class="mb-0">
                          <button class="btn btn-link collapsed"
data-toggle="collapse" data-target="#changePassword" aria-
expanded="false" aria-controls="changePassword">
                            Change Password
                          </button>
                        </h5>
                      </div>
                      <div id="changePassword" class="collapse"
aria-labelledby="headingThree" data-parent="#accordion">
                        <div class="card-body">
                        <form method="POST"
action="accountUpdate.php" id="changePasswordForm" novalidate>
                            <div class="form-group">
                                <input type="password"
autocomplete="on" class="form-control" name="currPwd" id="currPwd"
placeholder="Current Password" required />
```

```
                                    <div class="feedback"></div>
                                </div>
                                <div class="form-group">
                                    <input type="password"
autocomplete="off" class="form-control" name="pwd" id="pwd"
placeholder="New Password" required />
                                    <div class="feedback"></div>
                                </div>
                                <div class="form-group">
                                    <input type="password"
autocomplete="off" class="form-control" name="chkPwd" id="chkPwd"
placeholder="Re-type New Password" required />
                                    <div class="feedback"></div>
                                </div>
                                <div id="buttonBox">
                                    <button type="submit" class="btn-
block red-button">Change Password</button>
                                </div>
                            </form>
                            </div>
                        </div>
                    </div>

                    <div class="card">
                      <div class="card-header" id="logoutHeader">
                        <h5 class="mb-0">
                          <button class="btn btn-link"
onclick="location.href='logout.php'">
                              Logout
                          </button>
                        </h5>
                      </div>
                    </div>
                      <?php
                    }

                    if(!isset($_GET["user_id"]) )
                    $str = "My";
                else $str = "User";
                ?>
                    </div>
                </div>
                <div class="col-md-8">
                    <div>
                        <h2><?php echo $str?> Reviews</h2>
                    </div>
                    <div class="container" id="reviewContainer">
```

```
                        <?php

                        $reviewSql = "SELECT * FROM v_user_reviews
WHERE user_id = ".$user["user_id"];
                        $reviewResult = mysqli_query($conn,
$reviewSql);
                        if(!isset($_GET["user_id"]) )
                        $str = "You have";
                        else $str = "User has";
                        if($reviewResult) {

                            if($reviewResult->num_rows === 0) { ?>

                            <div class="card">
                                <div class="card-body">
                                    <h6><center><?php echo $str?> not
written any reviews yet.</center></h6>
                                </div>
                            </div>

                            <?php }

                            while($row = $reviewResult->fetch_assoc())
{
                                $movie_id = $row["movie_id"];
                                $title = $row["title"];
                                $date = date("F jS, Y",
strtotime($row["date"]));
                                $rating = $row["rating"];
                                $review = $row["review"];
                                //echo
$movie_id."<br>".$title."<br>".$date."<br>".$rating."<br>".$review; ?>

                                <div class="card">
                                  <div class="card-header">
                                    <div class="row">
                                      <div class="col-md-10">
                                        <h4><a href="movie.php?id=<?php
echo $movie_id ?>"><?php echo $title ?></a></h4>
                                        <h6 class="review-date">Written
on: <?php echo $date ?></h6>
                                      </div>
                                      <div class="col-md-2">
                                        <h4 class="review-score"><?php
echo $rating ?></h4>
                                      </div>
                                    </div>
                                  </div>
```

```
                                <div class="card-body">
                                  <h6><?php echo $review ?></h6>
                                </div>
                              </div>
                              <hr/>

                              <?php
                            }
                            mysqli_free_result($reviewResult);
                          } ?>

                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </body>
              </html>
```

4.3 DATABASE DESCRIPTION

S. No.	Table Name	Columns	Table Description
1	cast	movie_id actor_name character_name	Contains names of the actors and the names of the characters they played, for each movie.
2	movie	movie_id title genre content_rating runtime release_date avg_rating	Contains important movie information.
3	movie_meta	movie_id director writer budget title_poster_path summary	Contains additional movie metadata.

4	user	user_id email fname lname gender password country	Contains information about users.
5	user_feedback	user_id movie_id rating review date	Contains ratings and reviews posted by the users.

Table 4.1: Database Description

CHAPTER 5

RESULTS

CHAPTER 5

RESULTS

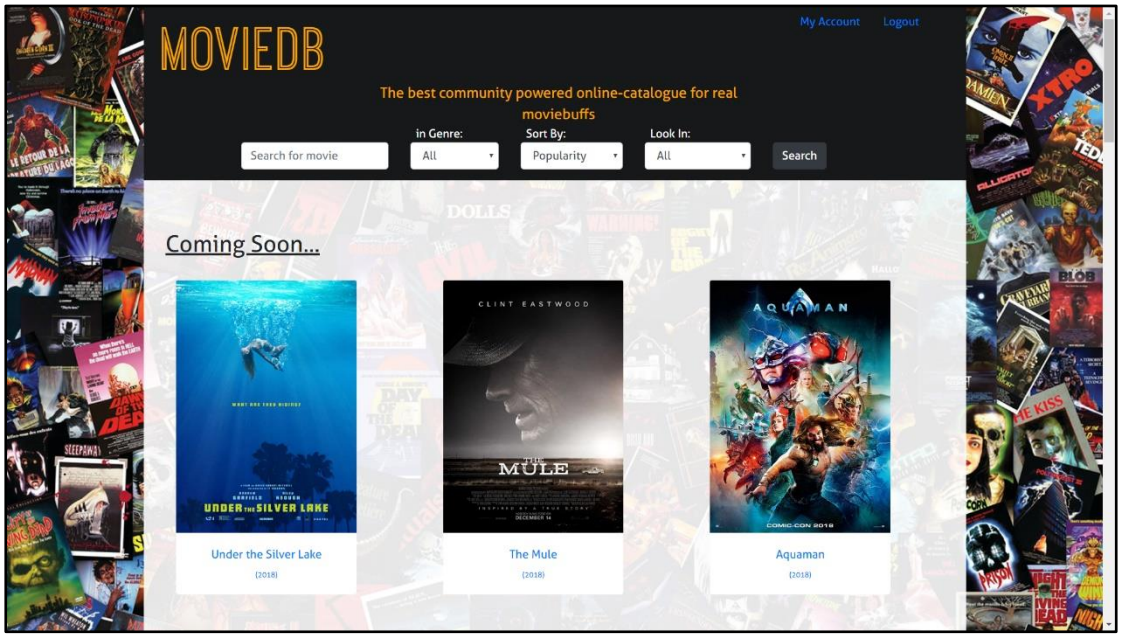

Fig 5.1: Home Page

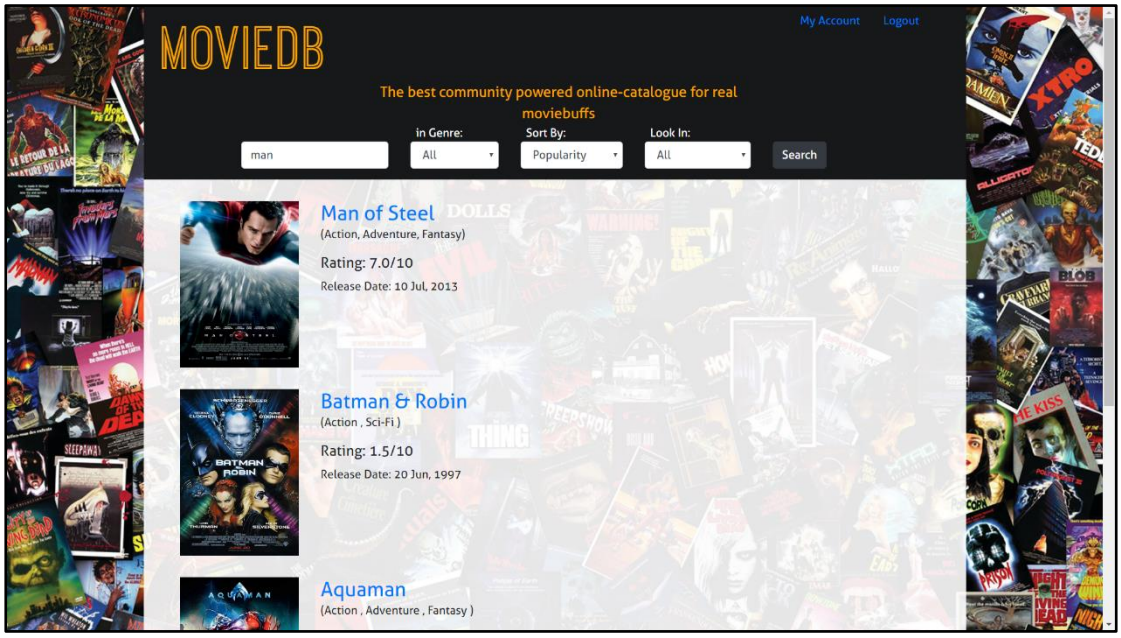

Fig 5.2 Searching for a Movie

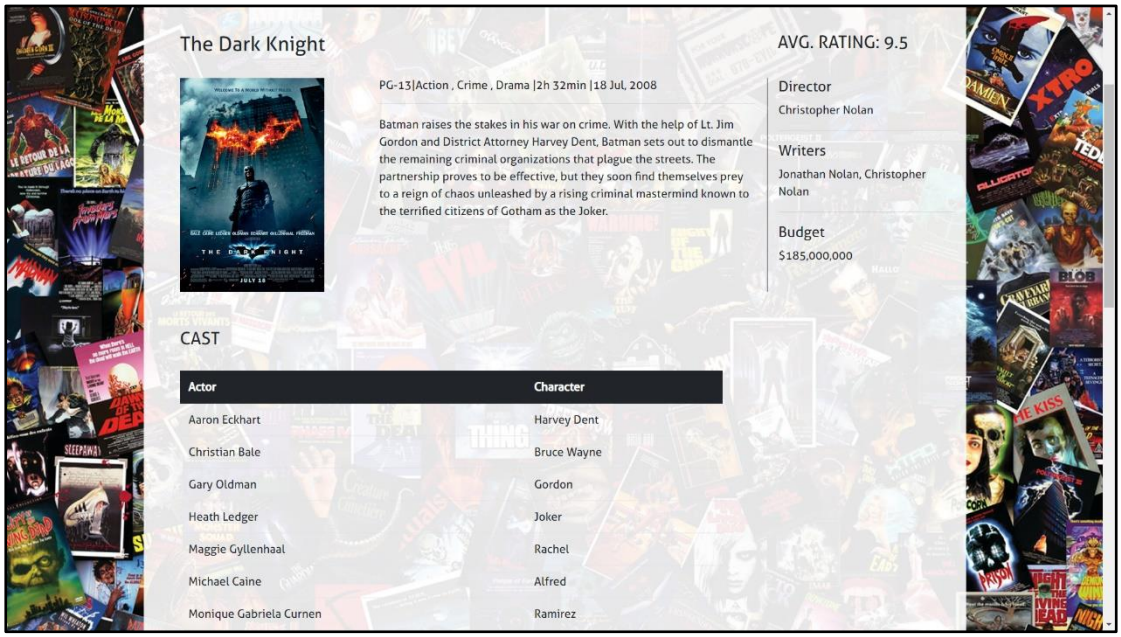

Fig 5.3: Viewing a Movie Page

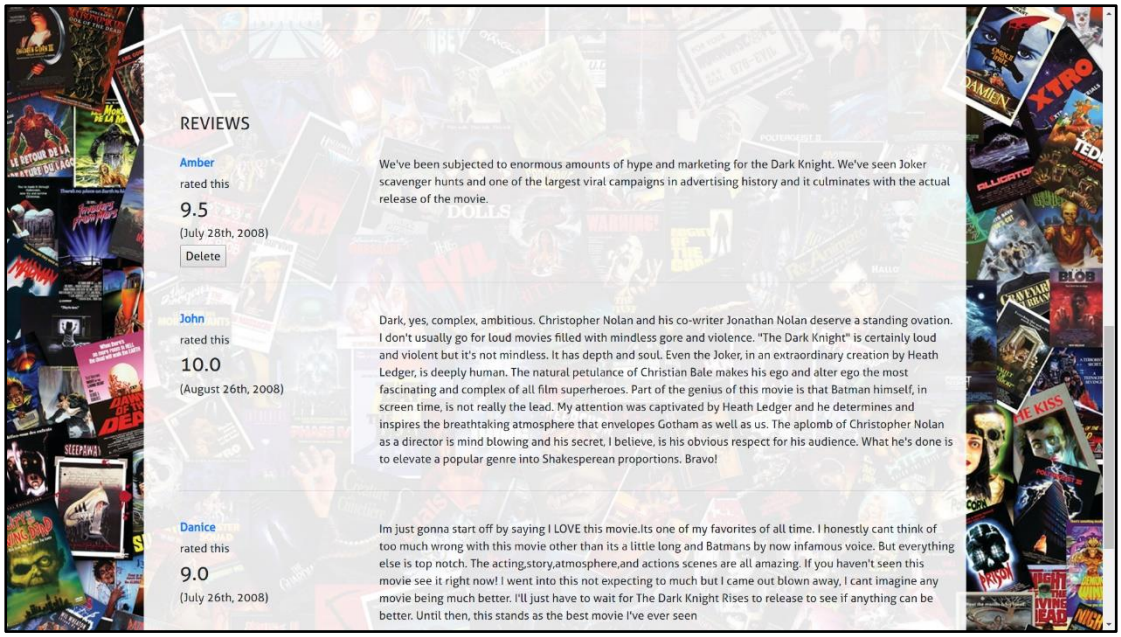

Fig 5.4: Reviews for a Movie When User is Signed in

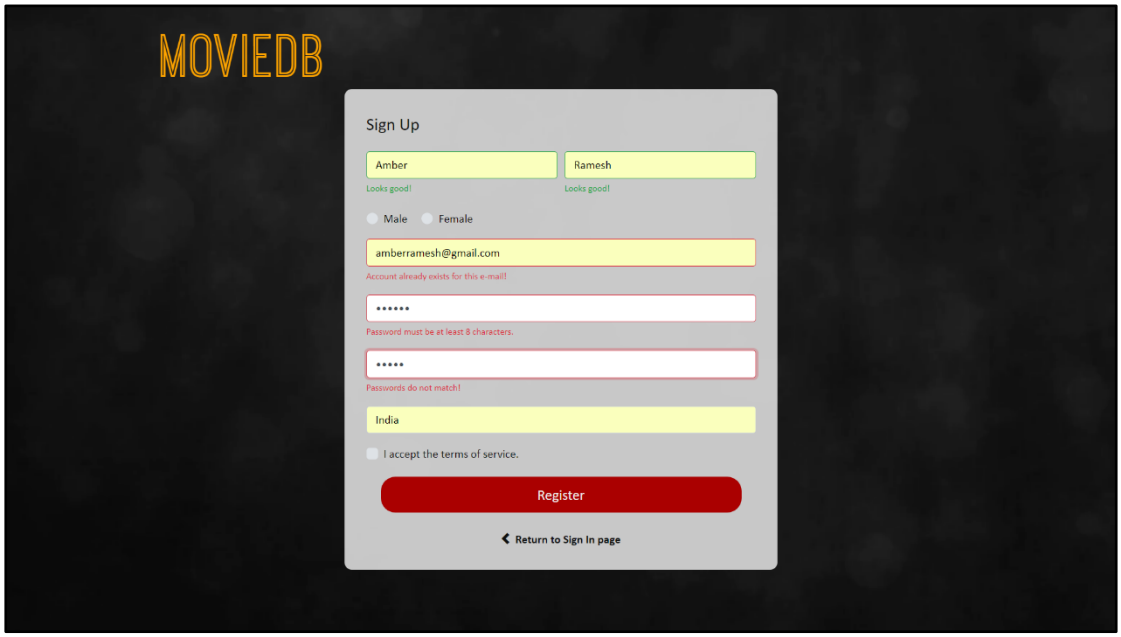

Fig 5.5: Registration Page After Validation Checks

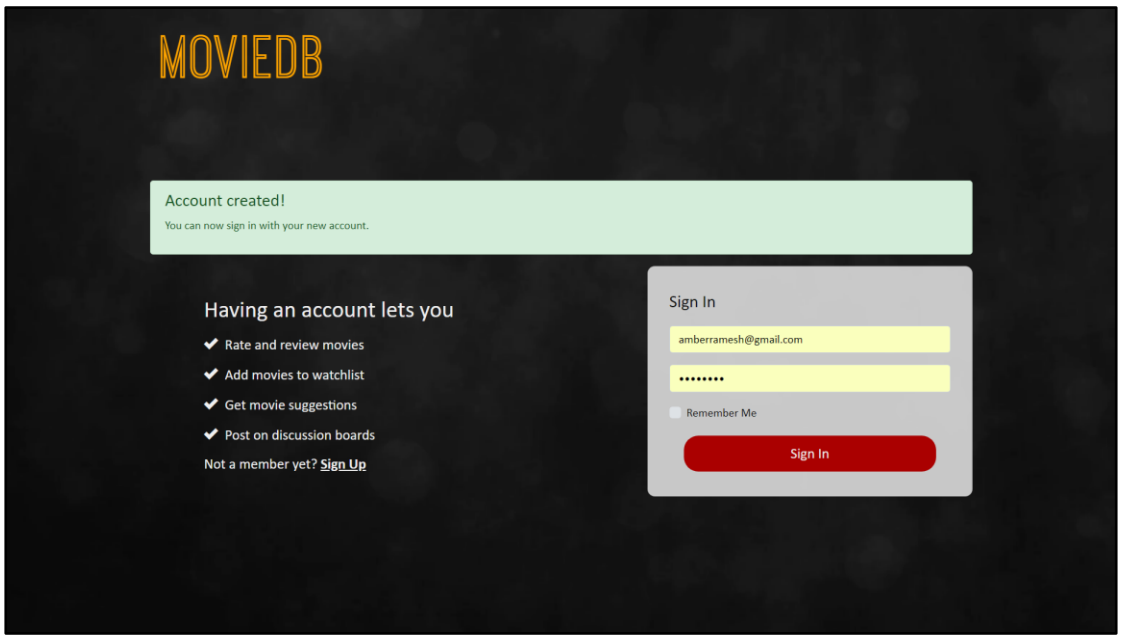

Fig 5.6: Redirection to Sign In After Successful Registration

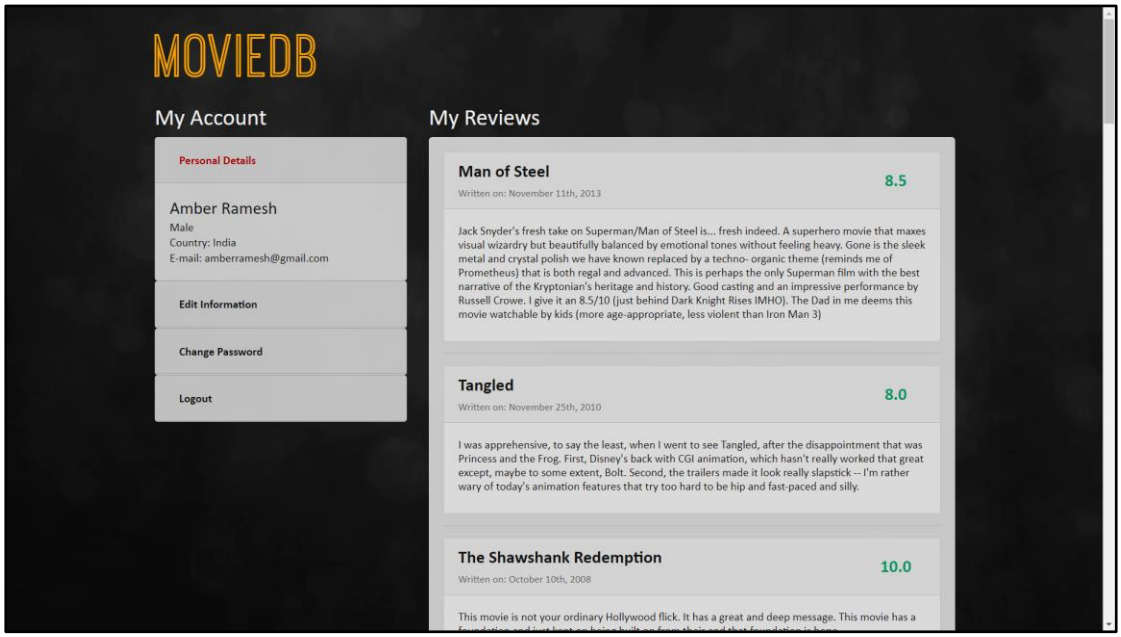

Fig 5.7: User Account Page

CHAPTER 6

CONCLUSION

CHAPTER 6

CONCLUSION

MovieDB is website with a large collection of movies across various categories. It provides a platform for users to not just view information but share their opinions with likeminded people on the site. The site is equipped with a rich interface that appropriately divides content across sections that helps the user navigate in an intuitive manner. An engaging experience is delivered to the user by providing detailed feedback on input, results sorted by their relevance and personalization of their home page through their activity.

By making extensive use of PHP and JavaScript, most of the site content is dynamically generated based on the activity taking place within the page. For instance, error messages are displayed next to their form fields only when the error is detected. Messages from the server are displayed with a timeout on few pages when an action is successfully completed or if it fails. PHP enables displaying relevant options to either view the account page or options for logging in based on the session status of the user. It also helps in redirecting user away from pages they are forbidden from visiting.

Adding a layer of responsive design lets the user browse the website on any device of their choice. Considering the popularity of smart phones, this makes the website available to a larger userbase and also makes upkeep easier if common versions for desktop and mobile browsing were to be maintained.

Overall, the website succeeds at providing a variety of features that a user would generally like to find on a website. Even so, certain improvements can still be integrated with the existing system to improve the experience provided by the website.

CHAPTER 7

FUTURE ENHANCEMENTS

CHAPTER 7

FUTURE ENHANCEMENT

IMDb being the primary source of inspiration for this website, a recommendation system could be added to the website which takes user interests, previously watched and rated movies and some of their personal information to give suggestions on what the user could be watching next. In its current state, the website can only provide suggestions based on sectional division, but it does not take into account various other parameters that are available on the website. By using algorithms that predict content that is similar to what the user likes, the site can provide the user with a better experience.

The website can also request the user for additional information that will help in developing detailed demographics. This kind of infographics is not just useful for developers making recommendation systems but also for the user who might want to take a look at, for instance, what most people belonging to a particular age group are watching right now. A user can explore this kind of information in their own way when a recommendation system sticks too much to traditional approaches of suggesting movies.

Additional sections which shows news, lets user add movies to watchlist or features like showing trailers or interviews of stars can also help in enhancing user experience. A feedback section can be provided to let users suggest their own improvements to the website.