2015

Personalized Movie Database System

Jayaprakash Garaga

Grand Valley State University

Follow this and additional works at: http://scholarworks.gvsu.edu/cistechlib

Recommended Citation
http://scholarworks.gvsu.edu/cistechlib/202

This Project is brought to you for free and open access by the School of Computing and Information Systems at ScholarWorks@GVSU. It has been accepted for inclusion in Technical Library by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.
Personalized Movie Database System

By
Jayaprakash Garaga
April, 2015
Personalized Movie Database System

By
Jayaprakash Garaga

A project submitted in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Information Systems

at
Grand Valley State University
April, 2015

Yonglei Tao
Date April 2014
# Table of Contents

Abstract ........................................................................................................................................... 4
Introduction ........................................................................................................................................ 4
Background and Related Work .......................................................................................................... 5
Program Requirements .................................................................................................................... 5
  - Navigation Diagram .................................................................................................................... 5
  - Home Page ................................................................................................................................. 6
  - Movie Page ............................................................................................................................... 6
Implementation .................................................................................................................................. 8
  - Technologies Used ...................................................................................................................... 8
  - Client-Server Architecture ......................................................................................................... 10
  - API Implementation .................................................................................................................. 10
  - Database Model ......................................................................................................................... 13
Conclusions and Future Work .......................................................................................................... 14
References ......................................................................................................................................... 14
Abstract

Personalized Movie Database System (PMDS) is a dynamic web application created for the purpose of viewing basic information about movies such as casting, trailers, ratings etc. It is designed as a one-stop destination for the user to access the movies that are Coming Soon, In Theatres or DVD/Blu-ray/Digital. Besides displaying the ratings from popular websites such as IMDB and Rotten Tomatoes, PMDS allows user to rate the movies. For the movies that are running in Theatres, PMDS displays movie show timings based on the user’s location. For the movies available in DVD/Digital versions, it provides the links to buy/stream them online. In addition to these, PMDS also suggests the similar movies that might interest the user.

PMDS application has a rich, user-friendly Graphical User Interface design developed using Wordpress and PHP. The movie data is obtained from available APIs provided by IMDB, Rotten Tomatoes and other official API providers. The data, which is static for a particular movie (Eg. Cast, Plot, Poster etc.), is fetched from the APIs and stored into MySQL database using JSON/XML. The data that may vary with time such as Ratings, Show times etc. are fetched in real time by calling the respective APIs.

Introduction

Project overview. What is your project about and what have you done? What was your motivation? Why is this problem you've worked on important? What were your goals and objectives? What are you trying to do and why?

In the current scenario, a moviegoer (user) has to visit more than one website to get the following basic movie information.

- List of movies playing in theaters, upcoming movies, DVD/Blu-ray movies
- Trailers
- IMDB Rating; Rotten Tomatoes Rating
- Simple Plot, Cast & Crew, Genre, Year Released, Runtime
- Show Times
- Links to stream/rent the movie online, buy DVD/Blu-ray
- Similar Movie Suggestions

There are websites like [www.imdb.com](http://www.imdb.com) and [www.rottentomatoes.com](http://www.rottentomatoes.com) with rich amount of the aforementioned data but the user has to open at least 3-4 websites to view all the data. So, this project was started with the intention of developing a one-stop destination for the user to view all the data. The data from these websites was fetched by calling the APIs and putting them together in a dynamic web application named as “Movie Buff”.


This web application is a Personalized Movie Database System (PMDS) where the user can login and rate the movies besides accessing the above listed information.

**Background and Related Work**

IMDB (Internet Movie Database) is one of the largest movie databases available in the web. It has detailed information about all the movies including movie show times and trailers. But it doesn't provide the information about “online streaming”. Rotten Tomatoes is another website which primarily gives the combined critic score of any movie. It provides the basic movie data, streaming links but not show times of movies playing in theaters. Movie lens suggests the movies based on user’s ratings for other movies and basic data but no information on “show times” and “online streaming”. Today, online streaming/renting a movie is widely popular and most of the movies are coming up with an option to rent the movie online at the same time when they are playing in theaters. PMDS covers the gap discussed above.

**Program Requirements**

**Navigation Diagram:**
**Homepage:**

The homepage consists of the list of movies playing in theaters, upcoming movies, and latest DVD/Blu-ray additions. From the homepage, the user can go to a movie page by clicking on the image of the movie scrolling or search a movie. The user can Sign-in or Register to setup an account.

**Movie Page:**

Movie page provides the following information.

**IMDB Rating, Rotten Tomatoes Rating, Cast & Crew, Simple Plot:**
Movie Show times based on the user’s location:

![Show Times](image)

Movie Trailer:

![Trailer](image)

Stream/Rent/Buy Digital Links:

![DVD/Blu-ray/Digital](image)

Similar Movies:
Implementation

Technologies Used:

PMDS is implemented using the following tools/technologies.

- Wordpress
- PHP
- MySQL
- XAMPP

Wordpress:
WordPress is a free and open-source blogging tool and a content management system (CMS) based on PHP and MySQL [1]. Features include plugin architecture and a template system.

Advantages of using Wordpress:

Easy, easy, easy ...

WordPress does not require PHP nor HTML knowledge unlike Drupal, Joomla or Typo3. A preinstalled plugin and template function allows them to be installed very easily. It's good choice for beginners.

Community
To have a useful support, there must be a large community of users, who will be a part of e.g. a discussion board.

Plugins
The script has over dozen thousand of plugins available on its website. They are the reason WordPress is considered a CMS, not only a blogging script.

Templates
On the script’s homepage you can view thousands of graphics templates that can change your website's look. You can find there both free and paid templates. The paid ones are often more advanced as well as more interesting.

Menu management

WordPress menu management has extended functionalities that can be modified to include categories, pages, etc.
Non-standard fields
You can easily add fields to forms by yourself or using plugins. It will allow your blog or subpages to have additional labels, categories or descriptions [2].

PHP:
PHP stands for Hypertext Preprocessor and is a server side scripting language for web development. It can also be used for general-purpose programming. The reference implementation of PHP is produced by the PHP group. The web server consists of a PHP processor module, which interprets the code and generates the resulting web page. The commands in PHP can be directly embedded in the HTML without the need to call an external file to process the data. PHP can be deployed on most of the web servers and can also be used as stand-alone shell on almost every operating system and platform. Scripting for the connections between front end and backend of PMDS was done in PHP [3].

MySQL:
MySQL is an open source database management system. It is developed, supported and distributed by Oracle Corporation. Database is structured collection of data. It can be anything from a simple to do list to vast amount information on some specific network. To add, delete, modify this data we need a database management system and MySQL is one such server. A relational database stores data in different tables. The logical model of the database with objects such as tables, views, rows and columns offer very flexible programming environment. These features make it easy to organize data and give the flexibility of reading or writing to the database by multiple people at the same time. Almost all the applications on web need web-hosting databases. MySQL is a web hosting database which can be used to store website information. It is difficult to maintain and manage databases with just SQL queries and is always easier using graphical user interface (GUI). PMDS was developed using a GUI called PHPmyadmin for managing the MySQL database. PHPmyadmin allows us to add, delete and modify database tables and entries. It also helps in creating backup for the database, run some specific SQL queries, search records and import records [4].

XAMPP:
XAMPP is an open source cross platform web server solution stack package. It consists of Apache HTTP server, MySQL database, and interpreters for code written in PHP and Perl programming language. XAMPP allows web site designers and programmers to test their work without the access to Internet. To make this work and easy, many of the security features on XAMPP have been disabled by default. XAMPP helps in creating and manipulating the databases like MySQL and SQLite. PMDS uses XAMPP to deploy it on the local system or Local host [5].
Client-Server Architecture:

The script is developed in PHP and the “Logic Tier” implements the script to fetch the data from the APIs. The data will be stored into MySQL database using JSON/XML. Every time the page refreshes, the logic tier fetches any new data from the API and store into the database. This process increased the performance of the application significantly. Dynamic data such as movie show times, trailers, streaming links are fetched in real-time and are not stored into the database.

**API Implementation:**

The following APIs are used to fetch the data.

**Rotten Tomatoes:**

The following details are fetched from Rotten Tomatoes API using the page number concept to avoid the maximum execution time error. The data fetched is stored into the database using a wordpress function “WP_INSERT_POST”.

- Movie Listings
  - In Theaters
  - Coming Soon
  - DVD

- Movie Details
  - IMDB ID
  - Title
  - Rotten Tomatoes Rating
API Request URL:

IMDB:
IMDB ID fetched from Rotten Tomatoes is used to get the following details about the movie. The data fetched is stored into the database using a wordpress function “WP_INSERT_POST”.
- Cast & Crew
- Poster
- Plot
- IMDB Rating
- Genre
- Year Released
- Runtime

API Request URL:
- http://www.omdbapi.com/?i=" $imdb_id."&plot=short&r=json

Gracenote TMS:
Gracenote TMS API is used to fetch the movie show times. This API don’t have a parameter such as IMDB ID to fetch the data. So, the movie title is used to get the show times. Radius to search for theaters is given as 50 miles around the zip code entered by the user.
- Theater
- Time
- Ticket URL

API Request URL:
- http://data.tmsapi.com/v1.1/movies/".$tmsid."/showings?startDate=" .date('Y-m-d')."&zip=" .$zipcode."&radius=20&api_key=4ctdqvsu7x8xhkww9fzxmu4

Trailer Addict:
Trailer Addict API is used to embed the movie trailer in the movie page. This is fetched through IMDB ID.

API Request URL:
- http://api.traileraddict.com/?imdb=":$imdb_id."&count=1&width=500&height=400"
Can I Stream It:
Since Can I stream don’t provide an API, iframe is used to embed the online streaming links in the movie page. IMDB ID is used to fetch the data.

- `<iframe frameborder="0" src="http://www.canistream.it/external/imdb/?php echo $imdb_id;?>>!mini-bar" width="100%" height="10" scrolling="yes"></iframe>`

My API Films:
Similar movies information is fetched from My API Films based on IMDB ID.

- Title
- Poster

API Request URL:

- `http://www.myapifilms.com/imdb?idIMDB=\$imdb_id\.&format=JSON&aka=0&business=0&seasons=0&seasonYear=0&technical=0&lang=en-us&actors=N&biography=0&trailer=0&uniqueName=0&filmography=0&bornDied=0&starSign=0&actorActress=0&actorTrivia=0&movieTrivia=0&awards=0&moviePhotos=N&movieVideos=N&token=261e19c2-fb1d-42ff-be62-6d5c7865e229&similarMovies=1`
Database Model:
Conclusions and Future Work

PMDS is a web application developed as a handy one-stop destination for a user who is interested to see the ratings, trailers, show times and streaming information of movies. It is successful in meeting the application requirements and there is a lot of scope for future development. However there are few challenges faced and drawbacks in this application, which would be taken care of during the future work.

Challenges and Drawbacks:
- Fetching the data through API and putting them together is one of the biggest challenges faced in this project.
- Since the dynamic data is coming from the API in real time, the application is running a bit slow.

Future Work:
- In this project, the movies that are fetched belong to these categories only: In theaters, Upcoming, DVD/Blu-ray. In future, PMDS would be developed to fetch the movies from more categories.
- Steps will be taken to improve the speed and performance of the application.
- A smart search engine based on tags, genre, and actors would be developed.
- In future, PMDS must be able to suggest the user based on their search history, ratings.
- A more professional GUI must be developed.

References:
11. http://www.rottentomatoes.com