Project Report

*Functionality clone of Shortfilmwindow.com*

Shashank Shetye Saudagar

For Ajency.in

February 25, 2017

Table of Contents

[Introduction: 0](#_Toc475788555)

[Application Features: 0](#_Toc475788556)

[Software Stack: 0](#_Toc475788557)

[Functional Analysis and Scope 0](#_Toc475788558)

[Database Design: 0](#_Toc475788559)

[E-R Diagram 0](#_Toc475788560)

[Entities: 0](#_Toc475788561)

[Implementation Details 0](#_Toc475788562)

[Screenshots: 0](#_Toc475788563)

[Movie details page: 0](#_Toc475788564)

[Listing page w/o filters 0](#_Toc475788565)

[Listing page w/o filters 0](#_Toc475788566)

[Listing page w/ search query 0](#_Toc475788567)

[References: 0](#_Toc475788568)

# Introduction:

The application of the web based short film database is a window for short film enthusiasts containing a database of short-films, directors etc. Users can easily search for films, genres and languages.

Project is stored at: <https://github.com/SSSaudagar/ListSiteClone> Branch: raw

# Application Features:

The basic functionalities of the application include:

* Searching for short-films based on Title
* Filtering based on Genre and Language
* Sorting in ascending and descending order based on Freshness, Length and Popularity
* Pagination
* Individual movie with display of different attributes
* Seamless flow in website

# Software Stack:

Following software was used for implementing given items

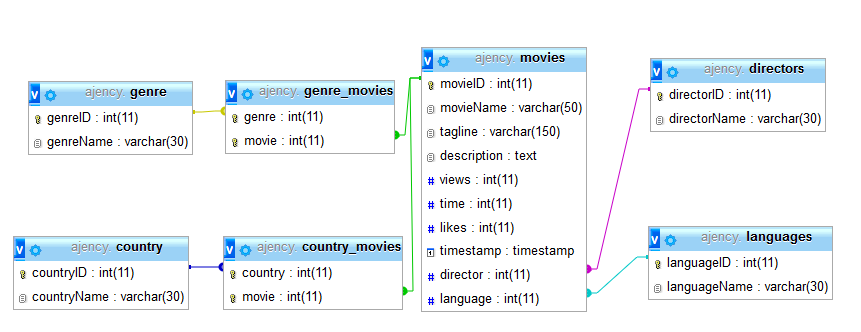
* Backend: PHP
* Database: MySQL
* Frontend: HTML, CSS, Bootstrap

# Functional Analysis and Scope

* Every film can have multiple Genres
* Every film has only one Language
* Every film has only one Director
* Popularity is determined by number of likes (a more complex function can be generated)
* 3 movies per page for pagination. Do not allow to go to previous page if on page 1
* Time to be stored in seconds, so can be converted to HH:MM:SS format
* First 300 characters of description are shown while listing

# Database Design:

## E-R Diagram



## Entities:

* Movies – stores all details of the movies
* Country – Stores country names with their ID
* Genre – Stores genre names with their ID
* Directors – Stores director names with their ID
* Languages – Stores language names with their ID

The relationships between these entity sets are self-explanatory. For example, Genre\_movies is an multi-attribute table for storing which genres belong to which movies. Directors and languages have a one-one relationship.

# Implementation Details

Implementing is done for following pages:

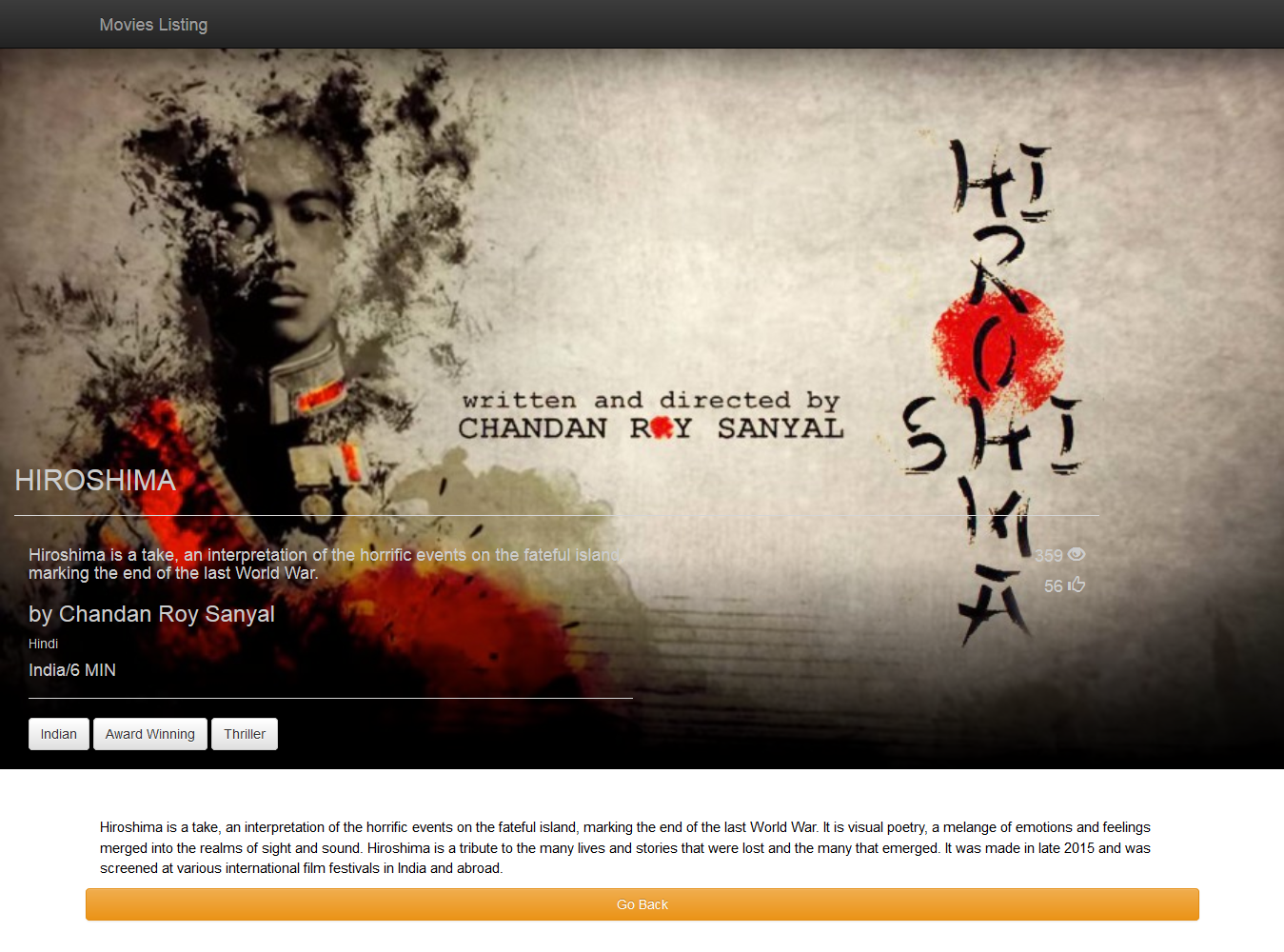
* The Listing Page:- stored at Views/mainPage, contains tools for listing, searching and filtering movies.
* The Details page:- stored at Views/MoviePage, contains details of the movie selected by the user.

The connection to the database and common functions used are stored in connect/connection.php file. There is proper demarcation of logic from frontend, logic functions are stored in files with suffix control. Index.php page directs to the listing page.

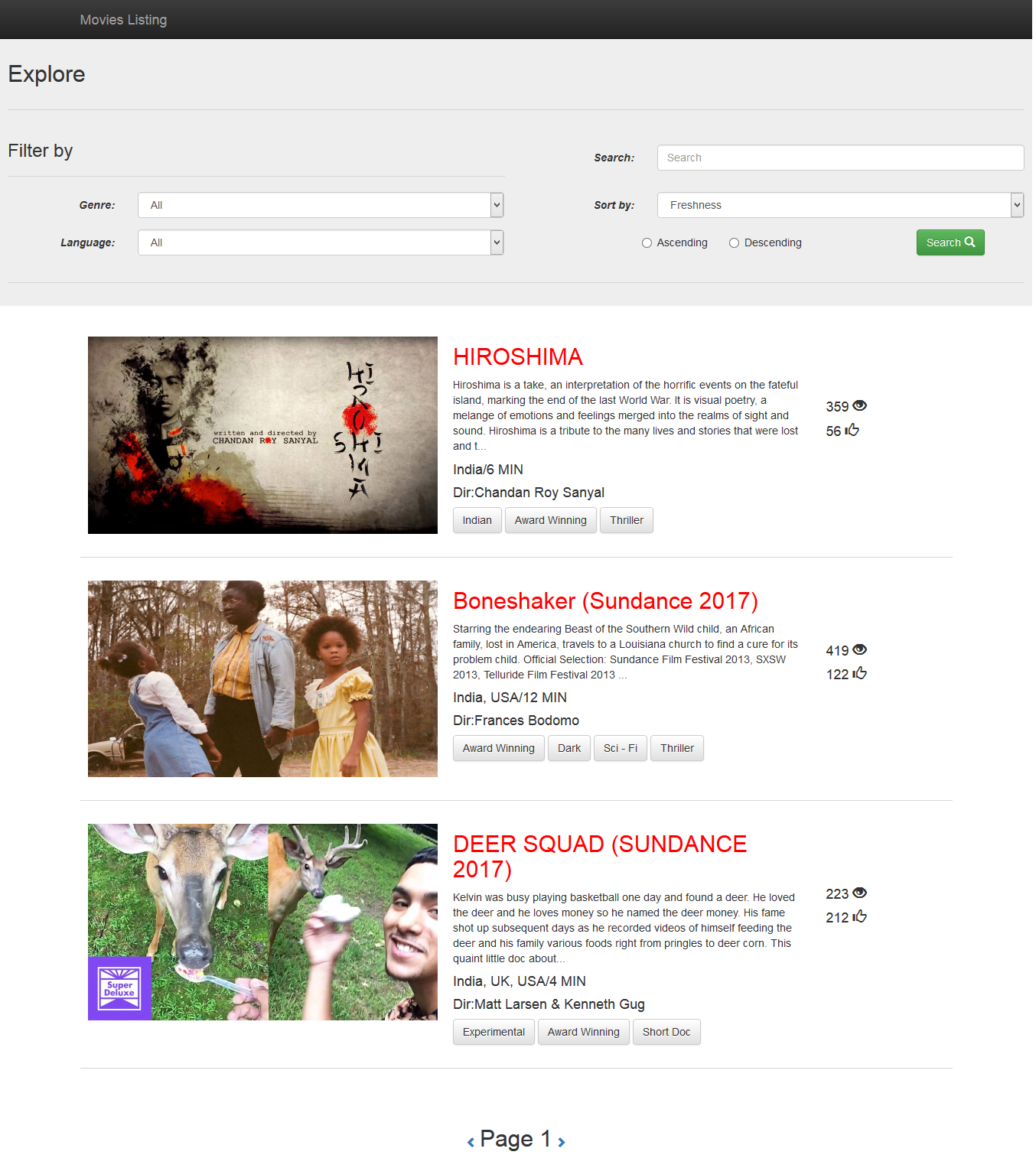
Apache Server from XAMPP package was used for testing the application. Javascript was not used anywhere in the application.

# Screenshots:

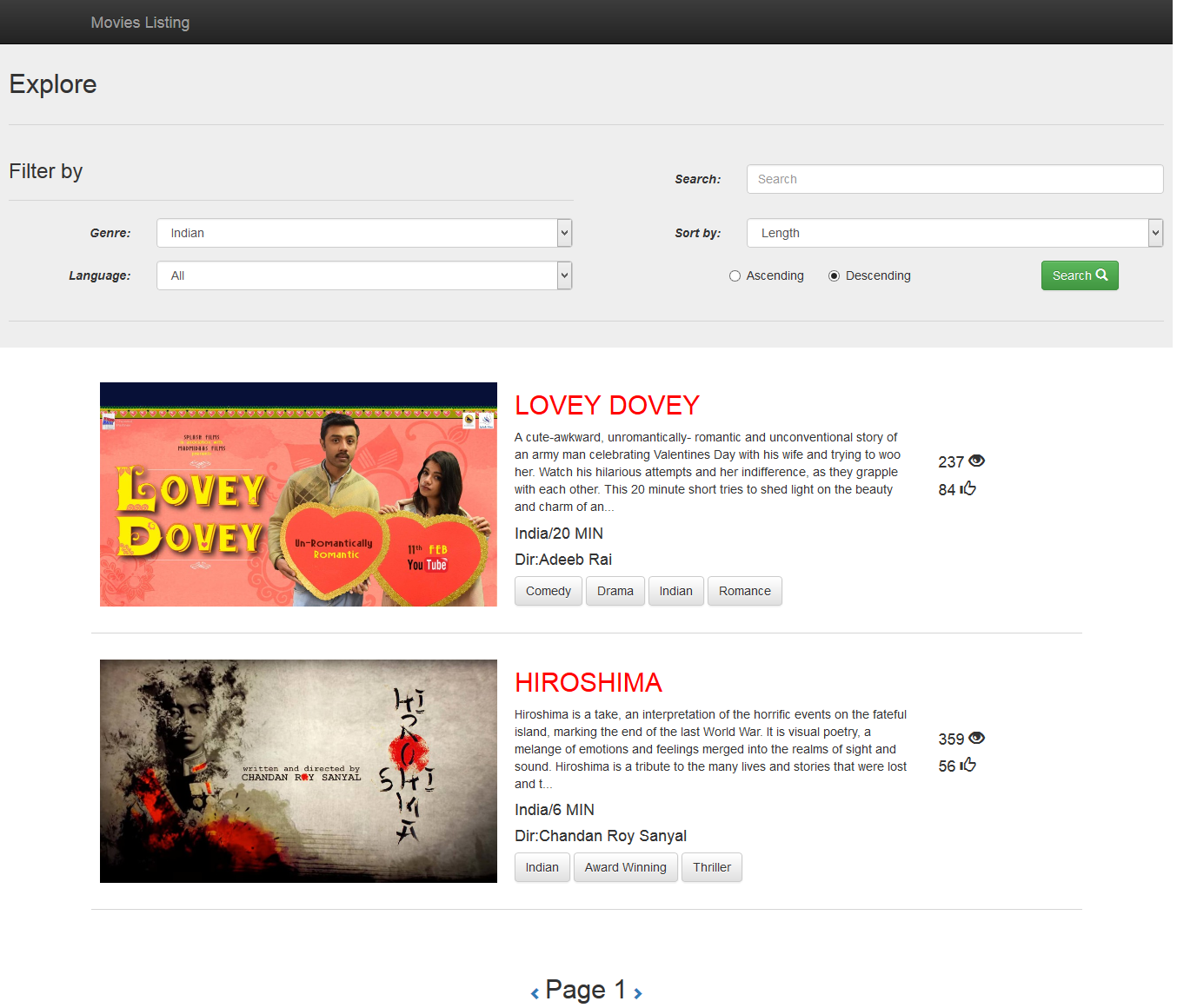
## Movie details page:



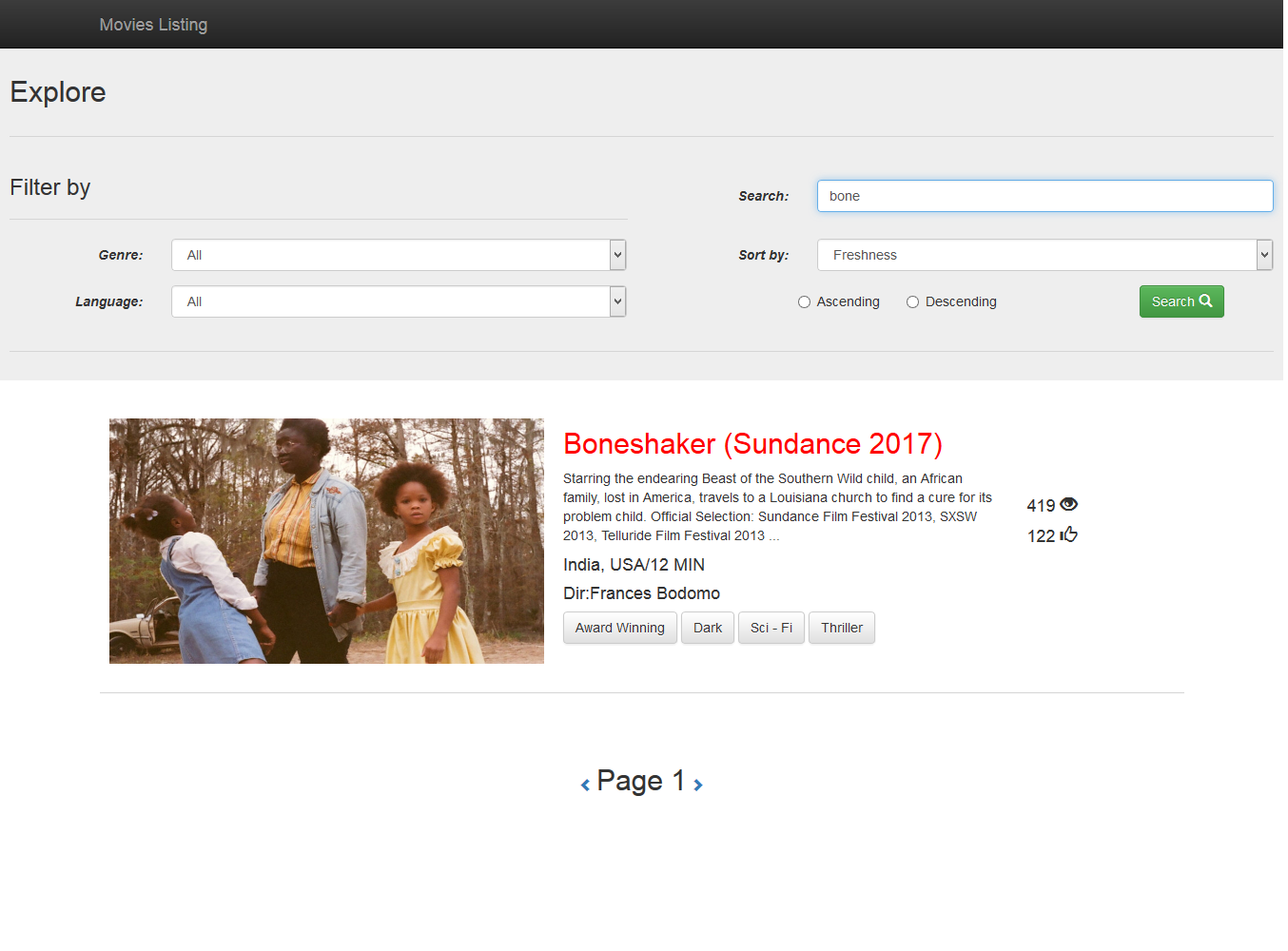
## Listing page w/o filters



## Listing page w/ filters



## Listing page w/ search query



# References:

* Draw.io
* <http://getbootstrap.com>
* <http://php.net/manual/en/index.php>
* <https://developer.mozilla.org/en-US/docs/Web/HTML>
* <https://www.phpmyadmin.net/>
* <https://dev.mysql.com/doc/refman/5.7/en/>
* Stack overflow