SYSTEM PROFILE

Group members:

1. Jahnavi Chowdary Tharigopula, 112078393
2. Vamsikrishna Kodumuru Meesala, 112006882

The entire project of Movie Database has been developed in ‘ubuntu’. We have implemented the front end of the UI using ‘HTML’ and we made use of ‘MySQL’ to create, store and modify the Database and its data. The Front end i.e. HTML pages were connected to the DBMS using ‘PHP’. SQL tables can be accessed and modified using the internal library of PHP. The developed system can be hosted on any server, in our case we used Apache 2 Ubuntu server to host the same. In-depth details regarding the functionalities of each component is described below.

Setting up the environment:

As described above, we made use of PHP, MySQL database and apache2 server. The ubuntu versions of the same can be installed using the following commands:

```
sudo apt-get install apache2
sudo apt-get install mysql-server
sudo apt-get install php5 libapache2-mod-php5 php5-mysql
sudo /etc/init.d/apache2 restart
```

Adding Tables and Adding Data to the tables:

First a database with name “moviedb” has to be created,
→ `create database moviedb`

Once the database is created tables can be created using the file ‘setup-database.sh’. The data has been scraped from IMDB using Beautiful soup and has been formatted and added in the file Load-data.sh. To insert this data into the tables execute the file ’Load-data.sh’
→ Move movie-database folder into apache folder, usually /var/www

Important: Before using any of the web pages the username and password in the files “display-engine.php” and “add-engine.php” has to changed to the actual database credentials.

HTML:

Multiple HTML pages were created for different functionalities of the UI, for example, viewing Actors, Directors, Adding Actor, Adding Director, Home page etc. Each HTML page either takes input from the user and redirects to another page where an SQL command is constructed using the given input or it directly executes the SQL command. In the cases where user gives input, on submitting the data a new HTML page is rendered where the Database operation is done and in the cases where user gives no input eg: Viewing Actors, SQL query is executed on the same page.

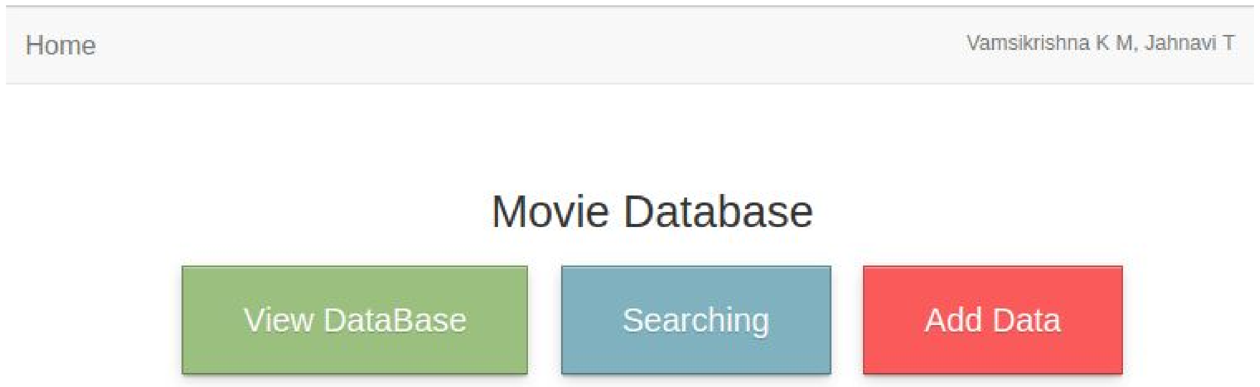


Fig.1. Homepage of the Movie Database UI

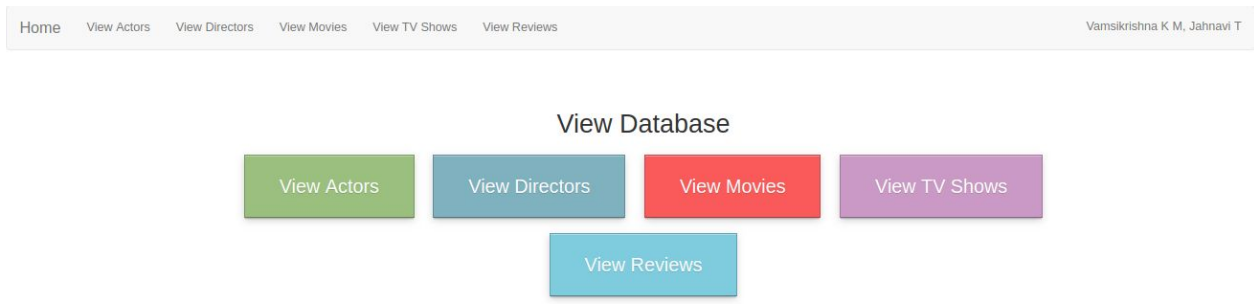


Fig.2. Page Showing multiple options of Viewing data in the database

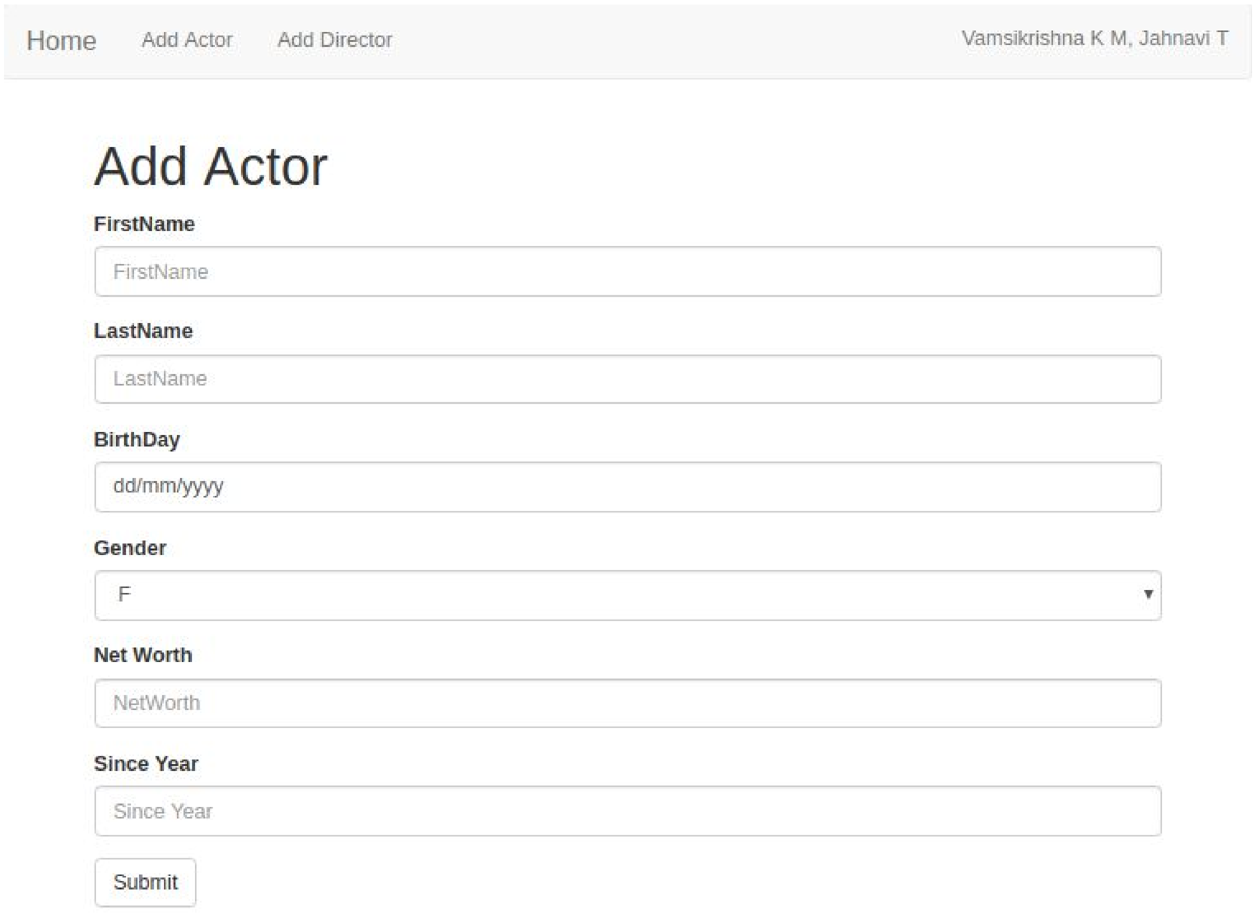


Fig.3. Page to Add an Actor

PHP:

The Sql Database is accessed using PHP. Before accessing the data in the tables, a connection has to be established with the DBMS, so a connection with database is established using the following PHP query,

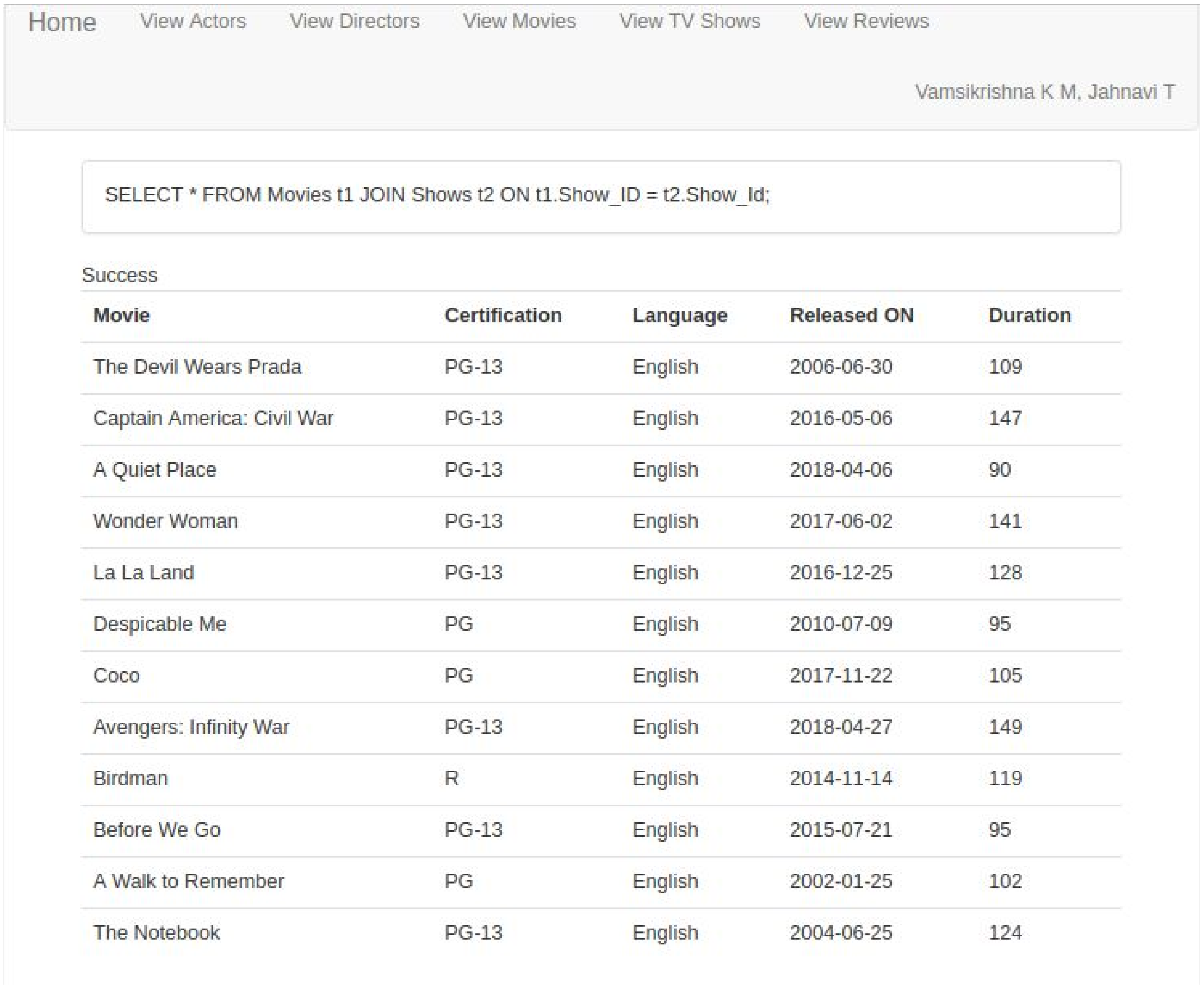
```
$con=mysqli_connect("localhost","root","password","moviedb");
```

Once a connection is established we can execute a query on the database using

```
$result = mysqli_query($con, $Query);
```

Data returned from SQL query is stored in a php array. To print the data to the HTML page each row is read in a loop and the data from selected attributes is printed. Sample code is as follows,

```
function display_row($fields, $row) {
        echo "<tr>";
        foreach ($fields as $field) {
                echo "<td>" . $row[$field] . "</td>";
        }
        echo "</tr>";}
```



```
SELECT * FROM Movies t1 JOIN Shows t2 ON t1.Show_ID = t2.Show_Id;
```

Success

Movie	Certification	Language	Released ON	Duration
The Devil Wears Prada	PG-13	English	2006-06-30	109
Captain America: Civil War	PG-13	English	2016-05-06	147
A Quiet Place	PG-13	English	2018-04-06	90
Wonder Woman	PG-13	English	2017-06-02	141
La La Land	PG-13	English	2016-12-25	128
Despicable Me	PG	English	2010-07-09	95
Coco	PG	English	2017-11-22	105
Avengers: Infinity War	PG-13	English	2018-04-27	149
Birdman	R	English	2014-11-14	119
Before We Go	PG-13	English	2015-07-21	95
A Walk to Remember	PG	English	2002-01-25	102
The Notebook	PG-13	English	2004-06-25	124

Fig.4. Page showing the Movies in the database (printed by PHP code)

The executed Sql query is being shown at the top of the page for the reference.

UI and Possible interactions with the Database:

With the developed UI, on a broader level a user can perform three kinds of actions. View contents in database, Search for something and Add data.

View Contents:

As shown in Fig.2, using the developed system an user can view things like,
1. Actors in the Database
2. Directors in the Database
3. Movies in the Database
4. TV Shows in the Database
5. Reviews in the Database

Search Database:

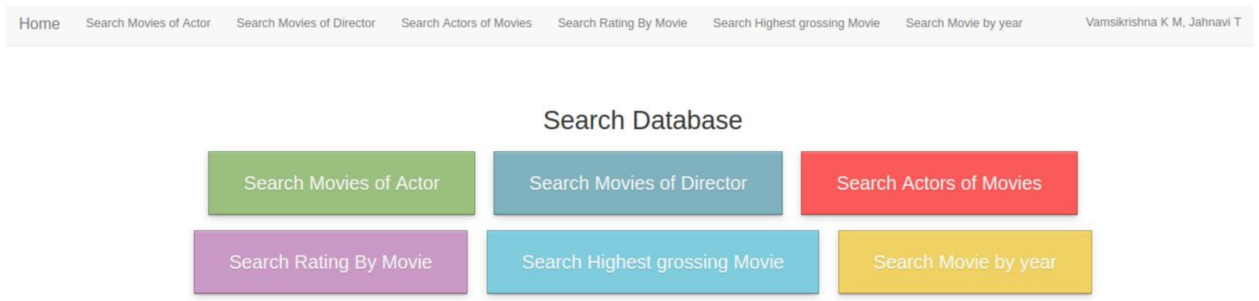


Fig.5. Page with Searching Options

In this set of actions, user will be asked to give an input on which the search will be made. For example, to search movies of an actor, user will be asked to input the Name of the actor. As it is hard to enter the exact name of the actor, search can be executed using partial name which includes partial first name and partial last name also.

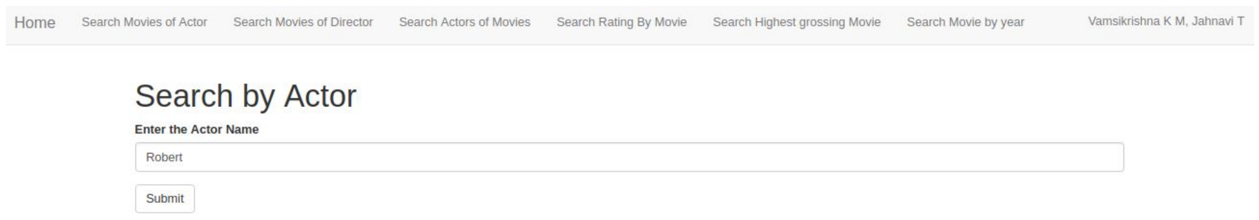


Fig.6. Field where user gives search term for searching movies by actor

Similar to the data input field shown above, user would have to enter search term in all the pages that fall into search category.

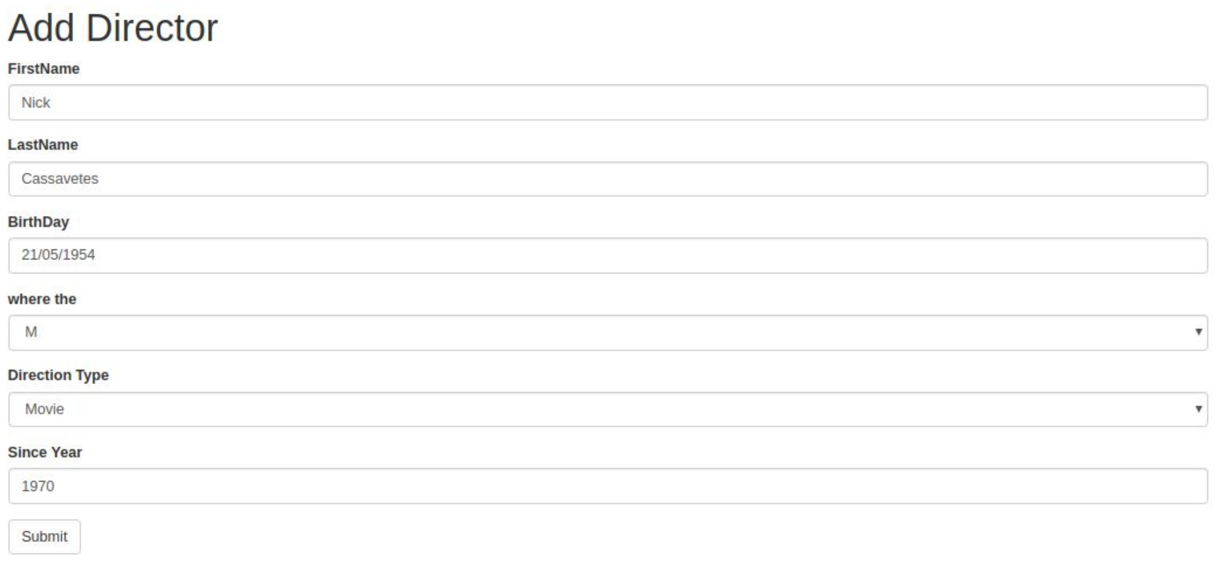
ADD Data:

Data such as a new actor and new director can be added to the database using the developed system. This application broadly leads to two kinds of operations where a new Actor or Director being added is not in the database at all and second is where the person exists in the database but under different category, for example 'john' may be an actor but we may want to add him in the directors list too. Both of these operations are carried out using the same page. Once the user gives input of the Actor or Director details, the details will be checked with person table and if the person exists, just the destination category table will be modified.

Repose message types:

1. Person with the same details exists. Same person is added to Actors list
2. Person with the same details exists. Same person is added to Directors list
3. Failed
4. Success

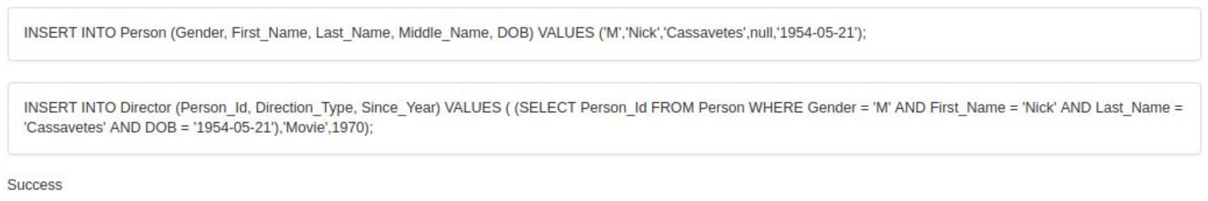
Response 1 and 2 are self explanatory, response success means that a person with the given details has been added to person table and the same person has been added to the given category (Actor or Director). Response Failed means that either the input is invalid or same person exists in the given category.



Add Director

FirstName	Nick
LastName	Cassavetes
BirthDay	21/05/1954
where the	M
Direction Type	Movie
Since Year	1970

Fig.7. Page to add director



```
INSERT INTO Person (Gender, First_Name, Last_Name, Middle_Name, DOB) VALUES ('M','Nick','Cassavetes',null,'1954-05-21');

INSERT INTO Director (Person_Id, Direction_Type, Since_Year) VALUES ( (SELECT Person_Id FROM Person WHERE Gender = 'M' AND First_Name = 'Nick' AND Last_Name = 'Cassavetes' AND DOB = '1954-05-21'),'Movie',1970);

Success
```

Fig.8. Response received after adding a director