

## CHAPTER 1

# INTRODUCTION

## 1.1 ARCHITECTURE

Database is a collection of related data. DBMS came into existence in 1960 by Charles. Again in 1960 IBM brought IMS-Information management system. In 1970 Edgor Codd at IBM came with new database called RDBMS. In 1980 then came SQL Architecture- Structure Query Language. In 1980 to 1990 there were advances in DBMS e.g. DB2, ORACLE. A database has the following implicit properties:

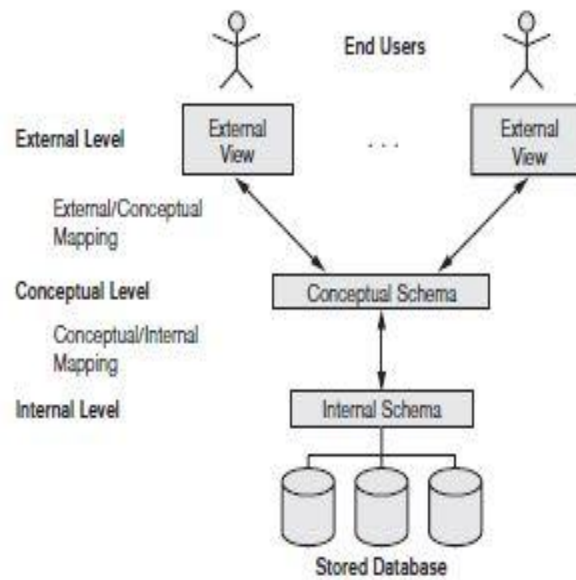
- ❖ A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
- ❖ A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database
- ❖ A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other word, a database some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents.

Metadata (meta data, or sometimes meta information) is “data about data”, of any sort in any media. An item of metadata may describe a collection of data including multiple content items and hierarchical levels, for example a database schema. In data processing, metadata is definitional data that provides information about or documentation of other data managed within an application or environment. The term should be used with caution as all data is about something and is therefore metadata.

A database management system (DBMS) is a collection of programs that enables users to create and maintain database. The DBMS is a general-purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.

Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database. The descriptive information is also stored in the database in the form database catalogue or dictionary; it is called meta-data. Manipulating the data includes the querying the database to retrieve the specific data. An application program accesses the database by sending the queries or requests for data to DBMS. The important function provided by the DBMS includes protecting the database and maintain the database.



**Figure 1.1: Three Schema Architecture**

The Figure 1.1 shows the Three schema architecture of Database Management System. The Three schema architecture consists of three levels of the architecture:

- **External Level:**

The external level is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users. In general, the end users and even the application programmers are only interested in a subset of the database.

- **Conceptual Level:**

The conceptual view is the information model of the enterprise and contains the view of the whole enterprise without any concern for the physical implementation. The conceptual view is the overall community view of the database and it includes all the information that is going to be represented in the database.

- **Internal Level:**

the internal view is the view about the actual physical storage of data. It describes what data is stored in database and how.

## **1.2 OVERVIEW OF THE PROJECT**

### **1.2.1 PROBLEM STATEMENT**

To maintain a movie database with supporting information about ratings and reviews of the movie along with other important information supported by different views.

### **1.2.2 OVERVIEW**

The Cinema Rating System stores and manages all the data related to movie database. It has a signup page after which only the registered users can review and rate a movie and add other relevant information to the movie database. Also viewing all data and editing of movie data can be done by any successfully registered user. The data of reviews is well protected for use and data processing. It can be used by any person or professional who are interested in knowing details about review of a particular movie. Facilities available in this mini project are maintaining record of different movies, production house details, details about the genre of the movie and the content rating of the movies so that appropriate age group movie goers watch movies according to the content rating, also one can view the top ten movies grouped according to the review and rating they have got. There is other additional view in which top 10 movies by box office collection are shown. This mini project is a planned approach towards working, accuracy, reliability, no redundancy and immediate retrieval of information.

### **1.2.3 OBJECTIVES OF THE PROPOSED SYSTEM**

The main objective of this system are:

- Search for movie details (Producer, Director, Actor, Actress, etc.)
- Insert Movie details.
- Rate and review a movie (only by registered user).
- Keeping track of top 10 movies.
- Ensure data accuracy, efficiency and security.

## CHAPTER 2

# SYSTEM DESIGN AND METHODOLOGY

## 2.1 SYSTEM ARCHITECTURE

The main software used for this are Html5, PHP and WampServer.

**PHP: Hypertext Preprocessor** is server language designed for web development and used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web templates system, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a common gateway interface executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP, although non-PHP text is still subject to control structures described in PHP code. The most common delimiters are `<?php` to open and `?>` to close PHP sections. The shortened form `<?` also exists.

**HTML: Hypertext Markup Language** is a markup language for creating a webpage. Webpages are usually viewed in a web browser. They can include writing, links, pictures, and even sound and video. HTML is used to mark and describe each of these kinds of content, so the web browser can display them correctly. HTML can also be used to add meta information to a webpage. Meta information is usually not shown by web browsers and is data about the web page, e.g., the name of the person who created the page. Cascading style sheets CSS is used to style HTML elements while JavaScript is used to manipulate HTML elements and CSS styles.

## 2.2 ENTITY RELATIONSHIP DIAGRAM

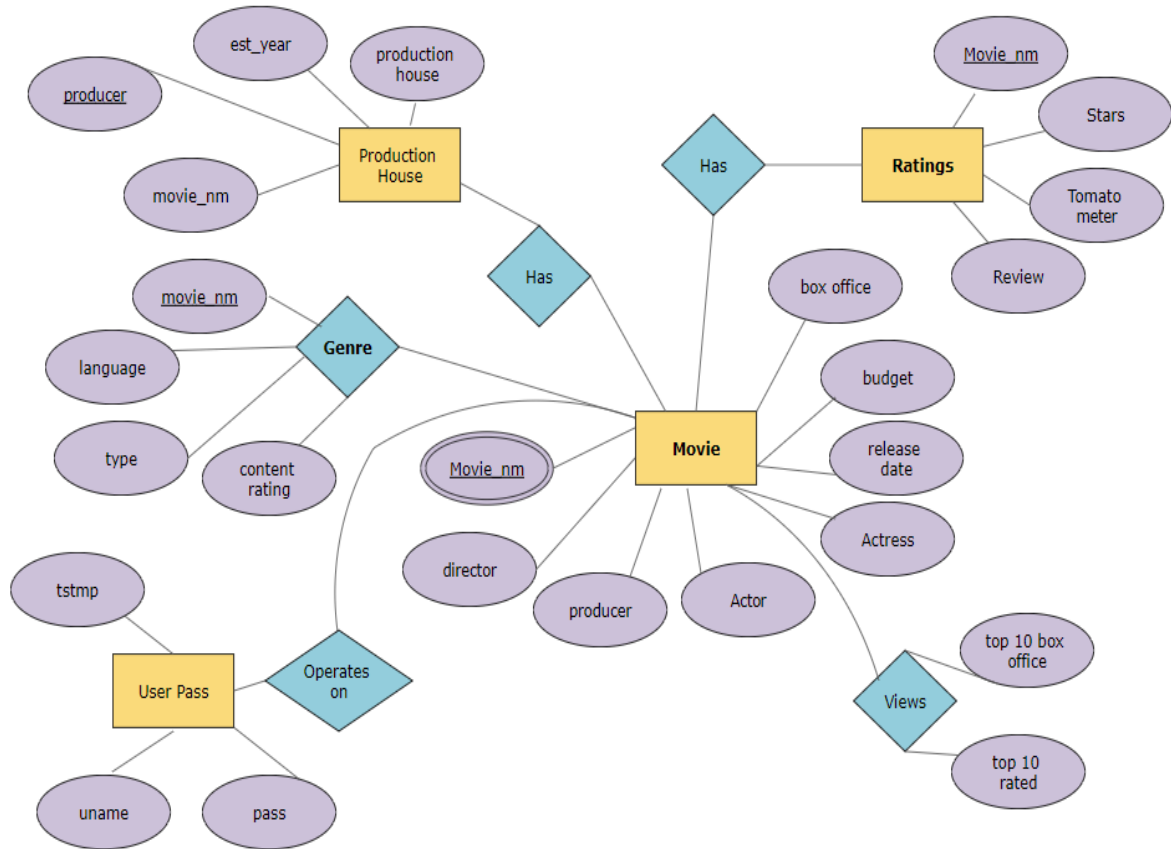


Figure 2.1: ER Diagram

An Entity-Relationship Diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representation model of data used to represent the entity framework infrastructure. The above figure 2.1 illustrates the E-R Diagram of movie database.

## 2.3 SCHEMA DIAGRAM

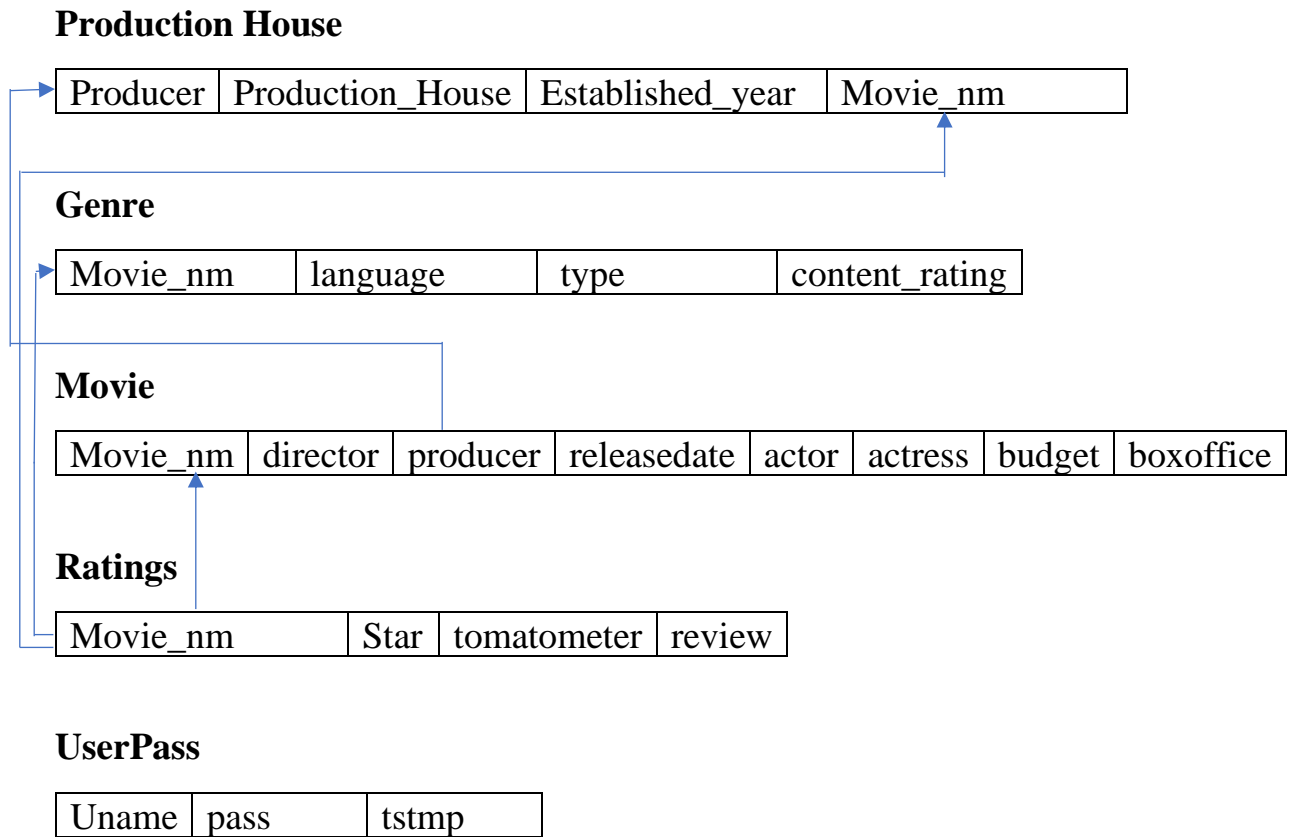


Figure 2.2 Schema Diagram

## 2.4 PROJECT REQUIREMENTS

<i>Hardware Requirements</i>		
<i>Processor</i>	<i>RAM</i>	<i>Disk Space</i>
Pentium i3, i5 or more	4 GB of Higher	10 GB or more
<i>Software Requirements</i>		
<i>Operating System</i>	<i>Language/Tools/Database</i>	
Windows 7, 8, 10	HTML5, PHP/ XAMPP Server and Brackets/MySQL	

Figure 2.3 Project Requirement



## 2.5 ALGORITHM

### TRIGGER:

To create a timestamp of the user when a new user first registers using the Sign Up Page.

**STEP 1:** BEGIN

**STEP 2:** Before register a user needs to register using username and password

**STEP 3:** After the input from user, execute “BEFORE INSERT ON USERPASS

For each row

SET new.tstmp=now();”

**STEP 4:** END

**Explanation:** After taking user id and password from the user, the system will create a timestamp of when the user registered and sets the “tstmp” attribute of the “userpass” table to the CURRENT\_TIMESTAMP.

## CHAPTER 3

### SYSTEM IMPLEMENTATION

To implement this mini project, XAMPP server is used for backend, html is used for front end and php is used for server-side scripting.

Some features of PHP are:

- **Simple:** It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world.
- **Faster:** It is faster than other scripting language e.g. asp and jsp.
- **Open Source:** Open source means you no need to pay for use php, you can free download and use.
- **Interpreted:** It is an interpreted language, i.e. there is no need for compilation.
- **Platform Independent:** PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.
- **Case Sensitive:** PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.
- **Error Reporting:** PHP have some predefined error reporting constants to generate a warning or error notice.
- **Real-Time Access Monitoring:** PHP provides access logging by creating the summary of recent accesses for the user.
- **Loosely Typed Language:** PHP supports variable usage without declaring its data type. It will be taken at the time of the execution based on the type of data it has on its value.

Some features of HTML5 are:

- **Web Workers:** Certain web applications use heavy scripts to perform functions. Web Workers use separate background threads for processing and it does not affect the performance of a web page.

- **Video:** You can embed video without third-party proprietary plug-ins or codec. Video becomes as easy as embedding an image.
- **Canvas:** This feature allows a web developer to render graphics on the fly. As with video, there is no need for a plug in.
- **Application caches:** Web pages will start storing more and more information locally on the visitor's computer. It works like cookies, but where cookies are small, the new feature allows for much larger files. Google Gears is an excellent example of this in action.
- **Geolocation:** Best known for use on mobile devices, geolocation is coming with HTML5.

### 3.1 MODULE DESCRIPTION

The modules included in this mini project are:

#### **HOME PAGE**

##### **OBJECTIVE:**

This is the front page which provides different authentication option and contains basic details about this system.

##### **INPUT:**

The user can select About, Team, Transactions, Queries, Log IN, Sign UP or Contact.

##### **OUTPUT:**

- On selecting About the user is redirected to the element of the page which shows the about section.
- On selecting Sign UP the user is redirected to a register page, where any new user can register before signing in.
- On selecting Transactions, User is redirected to login page. Without logging in user cannot make changes to the database.
- Queries section offers different views for Top movies.

**DESCRIPTION:**

Front end is designed using HTML5 and CSS3. The user will be taken to the desired page on clicking various links from where an user can have the permission to perform various operations on database.

**CODE SNIPPET:**

```
<div class="w3-top">

<div class="w3-bar w3-white w3-card" id="myNavbar">

  <a href="#home" class="w3-bar-item w3-button w3-wide"><b>CINEMA RATING
SYSTEM</b></a>

  <!-- Right-sided navbar links -->

  <div class="w3-right w3-hide-small">

    <a href="#about" class="w3-bar-item w3-button">ABOUT</a>

    <a href="#team" class="w3-bar-item w3-button"><i class="fa fa-user"></i>
TEAM</a>

    <a href="login.php" class="w3-bar-item w3-button"><i class="fa fa-th"></i>
TRANSACTIONS</a>

    <a href="queries.html" class="w3-bar-item w3-button"><i class="fa fa-usd"></i>
QUERIES</a>

    <a href="login.php" class="w3-bar-item w3-button"><i class="fa fa-usd"></i>
LOGIN</a>

    <a href="register.php" class="w3-bar-item w3-button"><i class="fa fa-usd"></i>
SIGNUP</a>

    <a href="#team" class="w3-bar-item w3-button"><i class="fa fa-envelope"></i>
CONTACT</a>

  </div>
```

**SIGN-UP PAGE****OBJECTIVE:**

This is the register page where a new user with unique user id and password can register themselves for further login process.

**INPUT:**

Username and password

**OUTPUT:**

A successful registration shall take the user to login page. On entering invalid username or password an error message will be displayed.

**DESCRIPTION:**

Front end is designed using HTML5 and CSS3. On registering successfully, the user can now use the same id and password to login any time and to access all the operations available for user to perform.

**CODE SNIPPET:**

```
if(isset($_POST['submit']))  
{  
    $uname = $_POST['userid'];  
    $pass = $_POST['password'];  
    $query1 = "insert into userpass values('$uname','$pass') ";  
    $data1 = mysqli_query($con,$query1);  
    if($data1)  
        { header('location:login.php') }  
}
```

else

```
{ echo "<h3><font color='white'>All Fields are required</font></h3>"; }
```

## **LOGIN PAGE**

### **OBJECTIVE:**

This is the login page where an already registered user with their user id and password can login themselves to the Cinema Rating System for further database operations.

### **INPUT:**

Username and password

### **OUTPUT:**

A successful login shall take the user to Database operation page i.e. Transactions page. On entering invalid username or password an error message will be displayed.

### **DESCRIPTION:**

Front end is designed using HTML5 and CSS3. On logging in successfully the user can access all the operations available for user to perform on the database.

### **CODE SNIPPET:**

```
if(isset($_POST['submit']))  
{  
  
    $uname = $_POST['userid'];  
  
    $pass = $_POST['password'];  
  
    $query = "SELECT * FROM userpass WHERE uid='$uname' &&  
pass='$pass'";  
  
    $data = mysqli_query($con,$query);  
  
    $row = mysqli_num_rows($data);  
  
    if($row == 1)
```

```
{ header('location:index1.html'); }  
  
else  
  
{  
  
    echo "<h2><font color='white'> Login Failed !!!</h2>";  
  
}  
  
}
```

## **TRANSACTIONS PAGE**

### **OBJECTIVE:**

This is the page where a registered user can act as a reviewer and can perform operation like inserting a movie details & deleting a movie detail.

### **INPUT:**

Movie Name

### **OUTPUT:**

A successful insertion or deletion operation.

### **DESCRIPTION:**

Front end is designed using HTML5 and CSS3. On deleting or inserting successfully, user will get a successful transfer message.

### **CODE SNIPPET:**

#### **INSERTION OF MOVIE DETAILS:**

```
$sql="INSERT                                INTO                                movie  
(movie_nm,director,producer,releasedate,actor,actress,budget,boxoffice)  
  
VALUES('$movie_nm','$director','$producer','$releasedate','$actor','$actress','$budge  
t','$boxoffice')";
```

```
if ($conn->query($sql) === TRUE) {
```

```
echo "New record created successfully";

} else {

    echo "Error: " . $sql . "<br>" . $conn->error;

}
```

### **DELETION OF MOVIE DETAILS:**

```
$sql = "DELETE FROM movie WHERE movie_nm = $movie_nm";
```

```
if ($conn->query($sql) === TRUE) {

    echo "Movie Details Successfully deleted";

} else {

    echo "Error: " . $sql . "<br>" . $conn->error;

}
```

### **QUERIES PAGE**

#### **OBJECTIVE:**

This is the page where Any user without registering can see the details of the top 10 movies classified by different categories.

#### **INPUT:**

User input via mouse on various links.

#### **OUTPUT:**

A different view based on the user choice.

#### **DESCRIPTION:**

Front end is designed using HTML5 and CSS3. On selecting a view, user will be redirected to a php page consisting of the view in form of HTML tables.



**CODE SNIPPET:**

```
$sql = "SELECT movie_nm, star, review FROM ratings order by star desc";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

    echo "<center><table
border=1><tr><th>Movie</th><th>Rating</th><th>Review</th></tr>";

    while($row = $result->fetch_assoc()) {

echo "<tr><td>".$row["movie_nm"]."</td><td>".$row["star"]."</td><td>".$row["re
view"]."</td></tr>";

    }

    echo "</center></table>";

} else {

    echo "0 results";

}

$conn->close();
```

## CHAPTER 4

### RESULT AND SCREENSHOTS

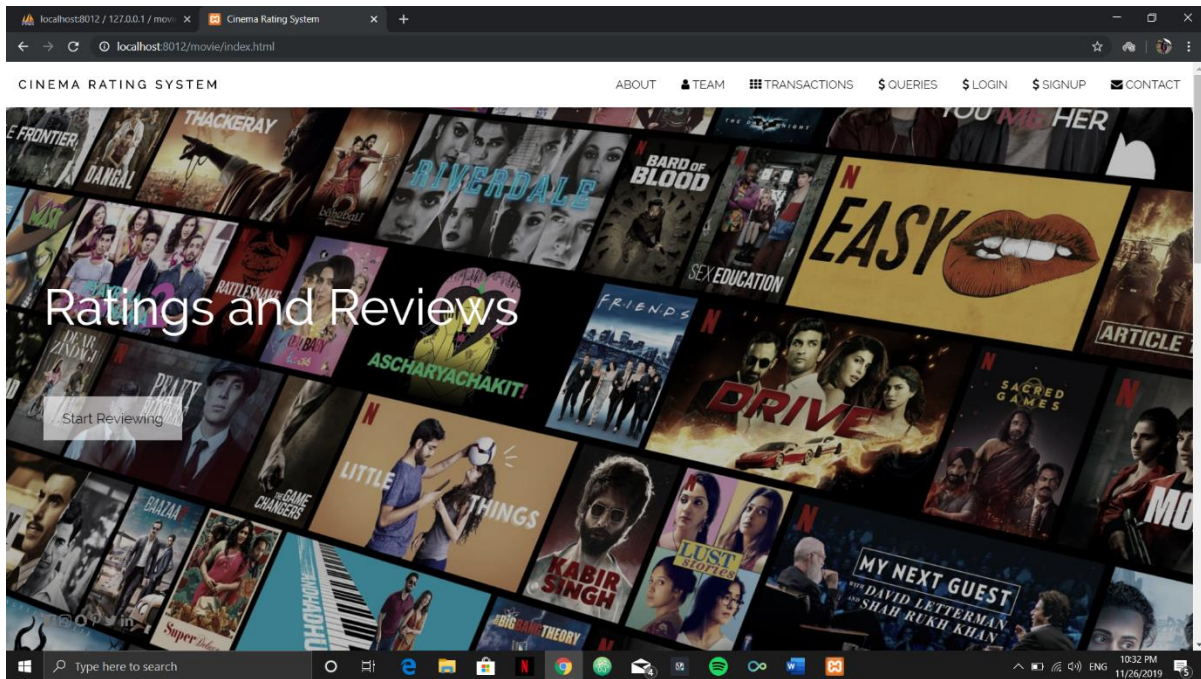


Fig4.1: Shows the homepage of the cinema rating system.

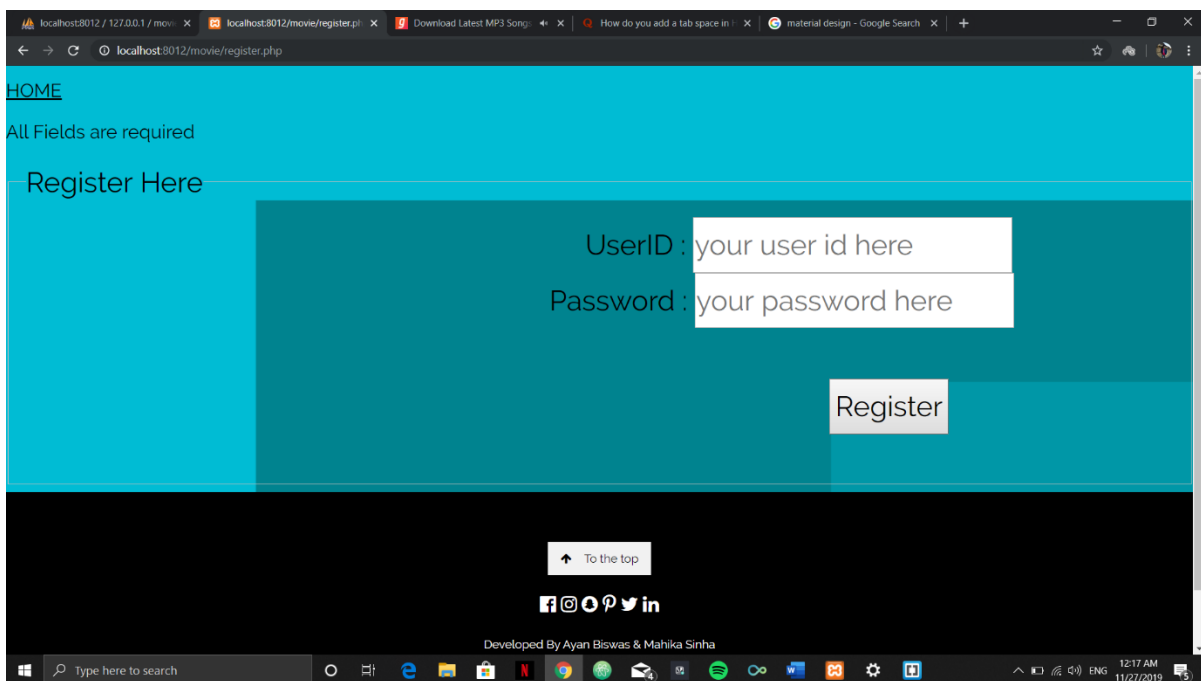


Fig4.2: Screenshot of the Signup page.

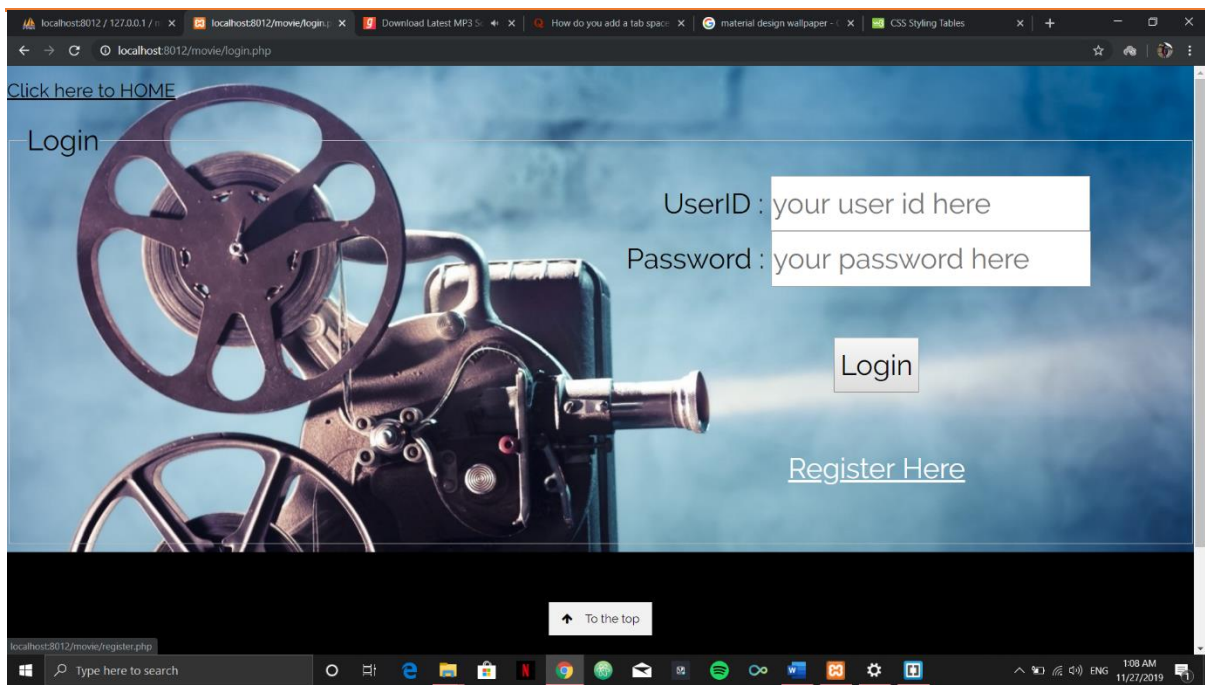


Fig 4.3: Screenshot of the Login page.

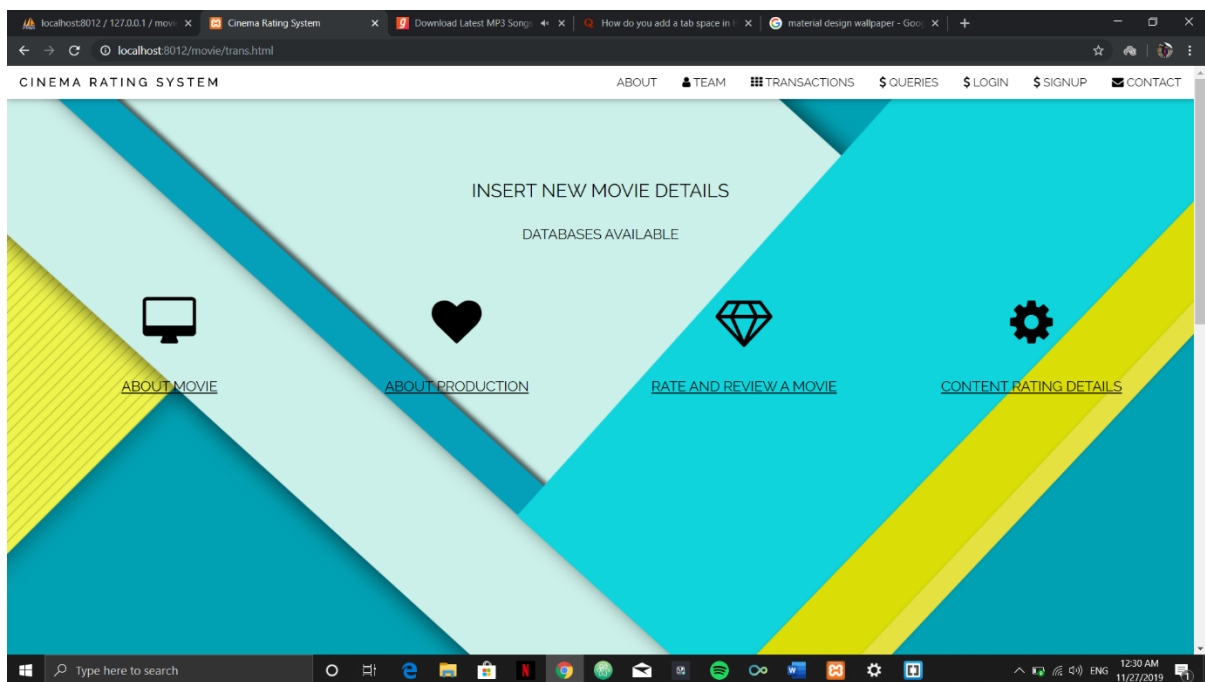


Fig 4.4: Screenshot shows the transaction page

After a user successfully logs in he/she will be redirected to this transaction page where he/she can insert information into the movie database such as the movie, production, genre and ratings table.

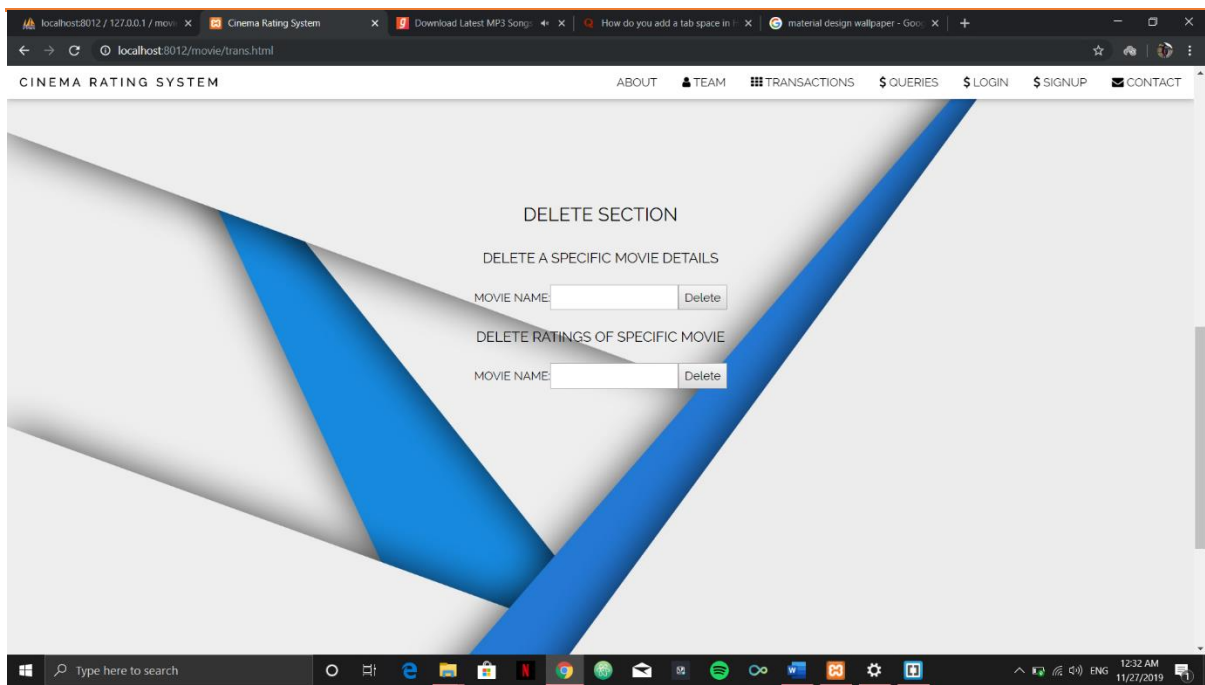


Fig 4.5: The delete Section of the transaction page.

This section allows registered users to delete specific movie details either from the table movie or ratings.

 The screenshot shows a web browser window with the URL 'localhost:8012/movie/insmovies.html'. The page contains a form for inserting movie details. The form fields are: 'Movie Name:' (with 'KGF' entered), 'Director:' (with 'Prashanth Neel' entered), 'Producer:' (with 'Vijay Kiragandur' entered), 'Release Date:', 'Actor:', 'Actress:', 'Budget:', and 'Box-Office:'. A green 'Submit' button is at the bottom left of the form. The browser's taskbar at the bottom shows the time as 1:27 AM on 11/27/2019.

Fig 4.6: Insert Movie Details Form

In this page user can insert the relevant movie details by clicking on the submit button.

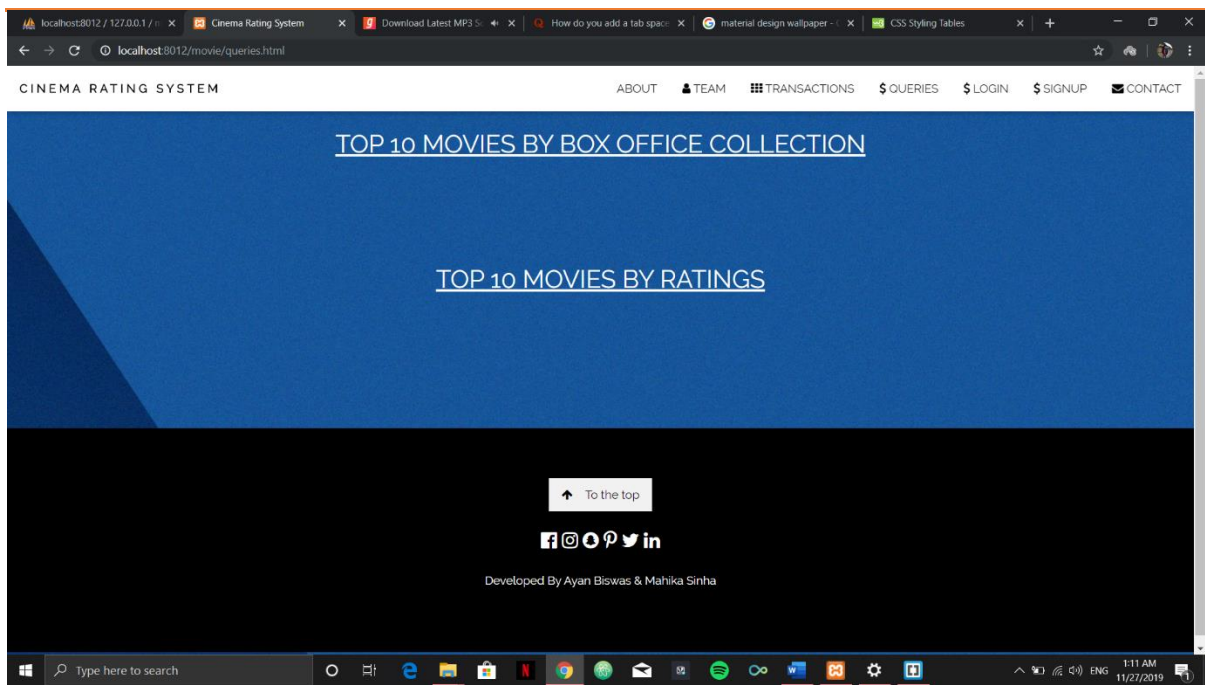


Fig 4.7: Shows the queries page.

User can click on the two given query links to get the relevant movie rating information.

Movie	Director	Actor	Actress	INR (Crores)
Avatar	James Cameroon	Sam Worthington	Zoe Saldana	50000
Iron Man	Jon Favreau	Robert Downey Jr.	Gweneth Paltrow	10000
Baahubali	SS Rajamouli	Prabhas	Tamannah Bhatia	1000
Dhoom 3	Vijay Krishna Acharya	Aamir Khan	Katrina Kaif	550
3IDIOTS	Rajkumar Hirani	Aamir Khan	Kareena Kapoor	330
KGF	Prashanth Neel	Yash	Srinidhi Shetty	250
Ramleela	Sanjay Leela Bhansali	Ranveer Singh	Deepika Padukone	200
Dabangg	Arbaaz Khan	Salman Khan	Sonakshi Sinha	123
Dilwale Dulhaniya Le Jayenge	Aditya Chopra	Shah Rukh Khan	Kajol	100
Dhoom	Vijay Krishna Acharya	John Abraham	Esha Deol	50

Fig 4.8: Top 10 movies according to Box office collection View

## **CHAPTER-5**

### **CONCLUSION**

This mini project has been developed using XAMPP server and php programming to meet the requirements of movie fans and lot more. The data can be accessed, manipulated and retrieved very easily. To conclude, this software has proved to be a user friendly.

### **FUTURE WORKS**

- a) The mini project shall host the platform on online servers to make it accessible worldwide.
- b) The mini project shall integrate multiple load balancers to distribute loads on system.
- c) Introduction of machine learning in this mini project to predict the coming performance of a movie or to predict the box office collections of the movie by analyzing previous trends.

## BIBLIOGRAPHY

- i. [www.youtube.com](http://www.youtube.com)
- ii. [www.w3school.com](http://www.w3school.com)
- iii. [www.tutorialpoint.com](http://www.tutorialpoint.com)
- iv. [www.stackoverflow.com](http://www.stackoverflow.com)