Introduction

1.1 The main objective of the project is to provide a platform where videos can be shared without any barriers for the purpose of knowledge and entertainment.

1.1.1 Gaining knowledge becomes easy if we find fun in it so we are trying to develop a site where the public can share knowledge and of course entertaining videos. This is an open video sharing site i.e. anyone can freely view, share and upload videos. This site is developed to display a wide variety of user-generated video content under the categories Tutorials, Music, Short Movies and Sports.

1.2 On Video Share anyone can view videos and feedbacks/comments without any restriction, but if any user wants to upload his own video or he wants to post a feedback/comment then the user must be registered on Video Share. For that purpose Video Share provides a user panel from where the user can register and then login to his account.

1.3 There is also a sidebar in Video Share which can be used to display latest videos watched by the user, top videos, also the advertisements etc.

1.4 To reduce load on the server, there is restriction that videos only up to 100MB size can be uploaded. The videos can be of the formats like .MP4, .OGG, .OGV, MPEG.

Software and Hardware requirements

The team suggests the following system specifications for the proper and smooth working of the product.

- Server side

Table 2.1

Serial No.	Requirement	Configuration
1.	Web Server	Apache v2.0+
2.	Database Manager	MySql
3.	Operating System	Windows
4.	Disk Space	5 Gigabytes
5.	RAM	2 Gigabytes
6.	Connection Port	100mbps shared or 16mbps dedicated

- Client side

Table 2.2

Serial No.	Requirement	Configuration
1.	Browser	Google Chrome
2.	RAM	1 Gigabyte
3.	Internet Connection	At least 256 kbps
4.	Disk Space	1 Gigabytes

Modules and Functionality

3.1 Home page

3.1.1 This is first page of videoshare. When the user loads this website this is the first page that he will see. From here the user can navigate to any other page as per his requirement.

3.1.2 The home page displays the latest ten videos with Video title and Video Id. The videos on this are arranged in descending order, ordered by their upload sequence.

3.2 Categories

3.2.1 Form categories module user can choose the desired video category to watch the categorized videos. This module works as a filter that enables the user to filter videos using the available categories.

3.2.2 This module displays names of different available categories. If an user selects an category then he will be directed to the desired page according to the selected category.

3.3 Tutorials/Music/Short Movies/Sports

3.3.1 These are different categories available on videoshare. User will be directed to this module if he selects a category from the category module.

3.3.2 It will display the videos of the selected category with Video title and Video Id arranged in descending order, ordered by their upload sequence.

3.4 Login

3.4.1 This module allows the user to login to videoshare. This module displays a login form, by entering his username and password in this form a user can easily login to videoshare.

3.4.2 The data from the form will be sent to the server for verification, if data is true then the user will be logged in, else he will have to try again.

3.5 Registration

3.5.1 This module provides the user to register to videoshare. This module displays a registration form by filling which he can register to videoshare. The user is required to fill in his Name, Username (must be unique), and desired password to successfully register to videoshare.

3.5.2 The data from the form will be sent to the server for verification of username, if the user name is unique then registration is successful else the user has to try again.

3.6 User panel

3.6.1 If user is logged in to videoshare then from this module he can use services like posting feedbacks, viewing videos or feedbacks posted by him.

3.7 Feedback

3.7.1 Feedback module displays all the feedbacks from any user. Feedbacks are displayed according to their sequence of post arranged in descending order.

3.7.2 If any user wants to post a feedback then he must first login to videoshare. If the user is logged in then he is be provided with a feedback form to submit his feedback. The data from feedback form will be sent to the server for further processing.

3.8 Sidebar

3.8.1 This module works as a small content space. It displays logout option, view feedback option, search option, and upload option.

3.8.2 This module can also be used to display advertisements, top videos, top posts etc.

3.9 Search

3.9.1 This module allows the user to search the videos by their Video title.

3.9.2 It provides a search form to the user, by entering the video title or its part, the user can search among all the available videos. The search results are displayed according video upload sequence in descending order.

3.10 Upload

3.10.1 This module is displayed only if the user is logged in. this module enables the registered user to upload videos to videoshare.

3.10.2 It provides a form to the user to choose file to upload, to choose the appropriate category and enter a file name. The data from the form is sent to server so that video file and other attributes can be stored.

3.11 Contact Us

3.11.1 This module displays the contact information of the owner of the website, so that a user can contact to the owner of the website.

Software Design

4.1 Web page design

4.1.1 Basic layout

The basic layout of web pages of videoshare has been designed using HTML5 (Hypertext Markup Language).

Fig. 4.1

```
<!DOCTYPE html>
<html>
<head>
<title>
Video Share
</title>
<style>
<?php
        require('style.css');
?>
</style>
</head>
<body>
<div id="wrap">
<div id="logo">
        <img src="videoshare.jpg" id="logo">
</div>
<div id="content">
        <p class="nav">
                <a class="nav" href="index.php">HOME</a>
                <a class="nav" href="categories.php">CATEGORIES</a>
                <a class="nav" href="bew/login_check.php">USER PANNEL</a>
                <a class="nav" href="contactus.php">CONTACT US</a>
        </p>
<br>
<br>
<center>
        <p>&copy; Video Share 2013</p>
<center>

<?php
        require('sidebar.php');
?>

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

4.1.2 Styling

The styling part of videoshare has been done using CSS3 (Cascaded Style Sheet).

Fig. 4.2

```
body
{
background:url('4.jpg');
}
p.nav
{
background-color:xblack;
}
p.success
{
color:black;
font-family:Comic Sans MS;
font-size:35px;
}
p.topic
{
background:black;
color:white;
text-decoration:none;
}
p.login
{
word-spacing:95px;
}
p.reg
{
word-spacing:95px;
}
p.tryagain
{
color:black;
font-family:Comic Sans MS;
font-size:35px;
}
p.wel
{
color:black;
font-size:15px;
text-align:right;
}
a.nav:link {text-decoration:none;
color:white; background:black;}
a.nav:visited {text-decoration:none;
color:white; background:black;}
a.nav:hover {text-decoration:underline;
color:white; background:black;}
a.nav:active {text-decoration:underline;
color:white; background:black;}
```

```
a.cat:link {text-decoration:none;
color:black; background:transperent;
font-size:40px}
a.cat:visited {text-decoration:none;
color:black; background:transperent;
font-size:40px}
a.cat:hover {text-decoration:underline;
color:white; background:black; font-
size:40px}
a.cat:active {text-decoration:underline;
color:black; background:black; font-
size:40px}

a.success:link {text-decoration:none;
color:blue; background:white;}
a.success:visited {text-decoration:none;
color:blue; background:white;}
a.success:hover {text-
decoration:underline; color:white;
background:black;}
a.success:active {text-
decoration:underline; color:blue;
background:white;}

a.reg:link {text-decoration:underline;
color:black; background:none;}
a.reg:visited {text-decoration:underline;
color:black; background:none;}
a.reg:hover {text-decoration:underline;
color:white; background:black;}
a.reg:active {text-decoration:underline;
color:black; background:none;}

a:link {text-decoration:underline;
color:blue; background:none;}
a:visited {text-decoration:none;
color:blue; background:none;}
a:hover {text-decoration:underline;
color:white; background:black;}
a:active {text-decoration:none;
color:blue; background:none;}

a.tryagain:link {text-decoration:none;
color:blue; background:white;}
a.tryagain:visited {text-decoration:none;
color:blue; background:white;}
a.tryagain:hover {text-
decoration:underline; color:white;
background:black;}
```

```
#vid
{
margin-left:10px;
}
#wrap
{
margin-top:0px;
width:1000px;
margin-left:auto;
margin-right:auto;
height:auto;
}
#logo
{
width:750px;
height:50px;
}
#content
{
display:inline;
float:left;
width:750px;
background: url('3.jpg');
}
#sidebar
{
border-radius:30px;
width:200px;
background: url('3.jpg');
border:1px ridge grey;
position:fixed;
margin-left:800px;
float:right;
}
#login
{
margin-left:240px;
margin-right:200px;
color:black;
}
#reg
{
margin-left:240px;
margin-right:200px;
color:black;
}
```

4.2 Sidebar design

Sidebar is also made by using HTML5 & CSS3.

Fig. 4.3

```
<html>
<head>
<style>
<?php
        require('style.css');
?>
</style>
</head>
<body>
</div>
<div id="sidebar">
<center>
<?php
        require('config.php');
        if(isset($_SESSION['authorised'])) //A Simple Yes Or No
        print "<a href=" ."bew/logoutf.php." .">" ."logout" ."</a>" ."<hr>";
?>
        <p>Click <a href="feedback.php">here</a> to view Feedbacks</p>
<hr>
<?php
        require('search.php');
?>
<hr>
<?php
        if(isset($_SESSION['authorised'])) //A Simple Yes Or No
        require('upload.php');
        else
        print "Please " ."<a href=" ."login.php" .">" ."Login" ."</a>" ." To Upload Videos";
?>
</center>
</div>
</body>
</html>
```

4.3 Login form

It is made using <form> tag in HTML.

Fig. 4.4

```
<form method="post" action="bew/auth.php">
<p><label><strong>Username :</strong><input type="text" name="usrnm"
/></label></p>
<p><label><strong>Password :   </strong><input type="password" name="pass"
/></label></p>
```

4.4 Registration form

It is made using <form> tag & <table> tag in HTML.

Fig. 4.5

```
<form method="post" action="bew/regf.php">
        <table align="center">
        <tr>
        <td>
                <label><strong>Name:</strong>
        </td>
        <td>
                <input type="text "name="name" /></label>
        </td>
        </tr>
        <tr>
        <td>
                <label><strong>User Name:</strong>
        </td>
        <td>
                <input type="text" name="usrnm" /></label>
        </td>
        </tr>
        <tr>
        <td>
                <label><strong>Password:</strong>
        </td>
        <td>
                <input type="password" name="pass" /></label>
        </td>
        </tr>
        <tr>
        <td>
                <input type="submit" name="submit" class="button" value="Submit" />
        </td>
        </tr>
        </table>
        </form>
```

4.5 Search form

It is made using <form> tag in HTML.

Fig. 4.6

```
<form method="get" action="bew/searchf.php">
        <p><label><strong>Search Videos</strong><input type="text" name="sitem"
/></label></p>
        <input type="submit" class="button" value="Search" />
</form>
```

4.6 Upload form

It is made using <form> tag in HTML.

Fig. 4.7

```
<form
              action="bew/uploadf.php" method="post"
              enctype="multipart/form-data" method="POST">
             <p><strong>Upload</strong></p>
             <p><label><strong>Title :</strong><input type="text" name="vname"
/></label></p>
                          <input type="file" name="videofile" />
             <p><strong>Category</strong><select name="category">
                           <option>Choose
                          <option>Tutorials
                          <option>Music
                          <option>ShortMovies
                          <option>Sports
                                                          </select><p>
                          <input type="submit" value="Upload"/>
 </form>
```

4.7 Feedback form

It is made using <form> tag in HTML.

Fig. 4.8

```
<form method="post" action="bew/feedbackf.php">
             <p><strong>Feedback:</strong></p>
             <p><textarea rows="10" cols="30" name="feed"></textarea></p>
             <p><label><strong>Video Id:</strong><input type="text" name="vid"
/></label></p>
             <p><br><input type="submit" name="submit" class="button" value="Submit"
/></p>
</form>
```

4.8 Data base design

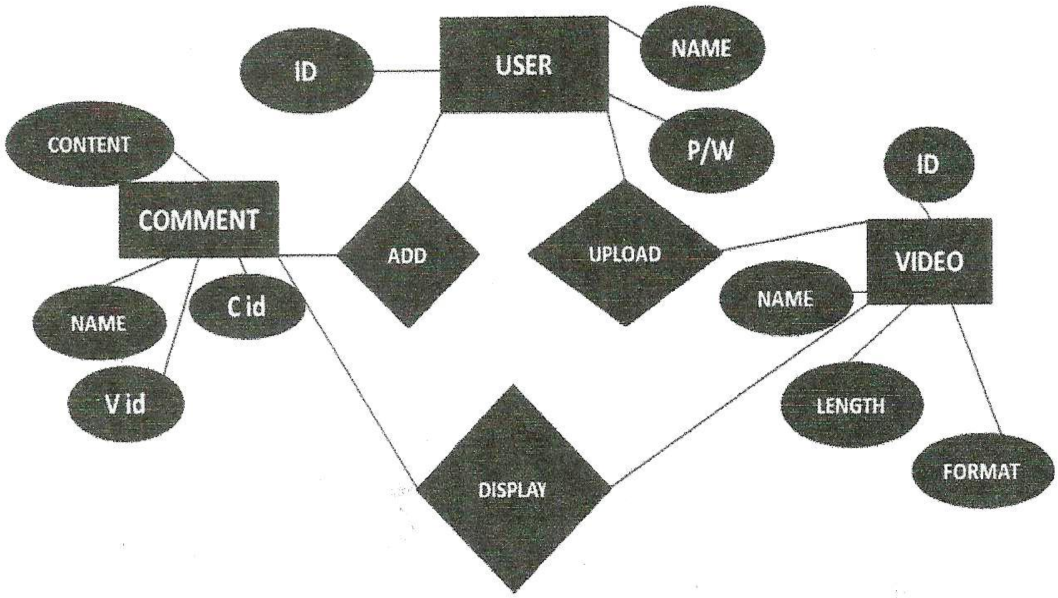
4.8.1 It is done using MySql.

- Database name - videoshare
- Total tables – 4
- Table names –

1. Feedback
2. Users
3. Videos
4. Search

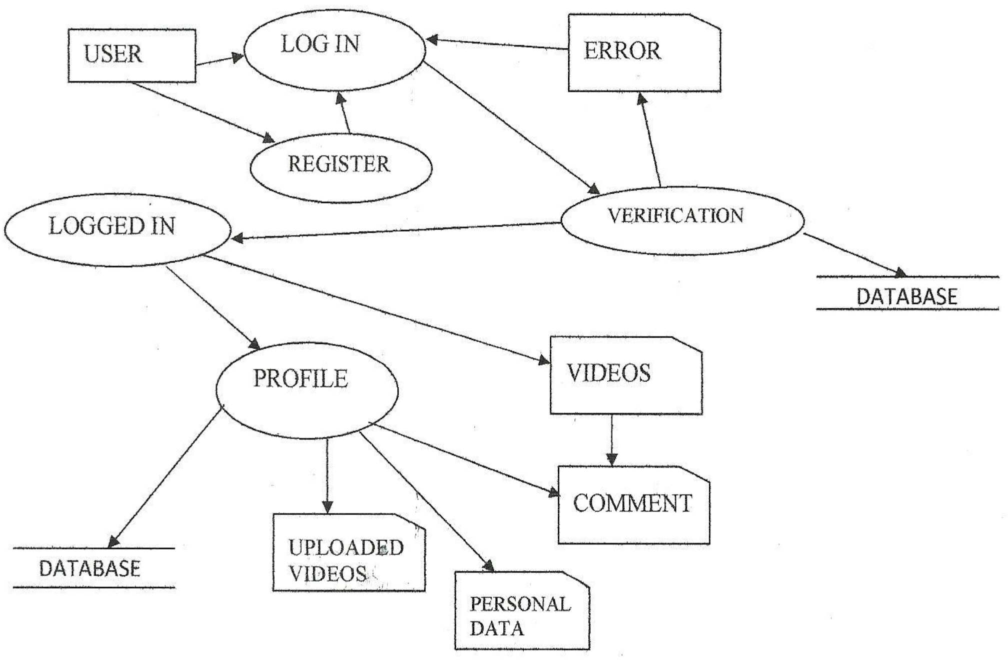
4.8.2 E-R Diagram

Fig4.9



4.8.3 Dataflow diagram

Fig. 4.10



Code Templates

5.1 Coding

5.1.1 Code to connect to MySql database using PHP

Fig. 5.1

```
<?php
// Create connection
        $con=mysqli_connect("127.0.0.1","root","","videoshare");
// Check connection
        if (mysqli_connect_error($con))
        {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }
?>
```

127.0.0.1 is the server IP where database is present, root is username of MySql, no password is their so field is blank, videoshare is the name of database.

5.1.2 Code to display videos

Fig. 5.2

```
<?php
        require('config.php');
        $result = mysqli_query($con,"SELECT  vname, videofile, category, vid FROM
videos order by vid desc limit 10");
        while($row = mysqli_fetch_array($result))
        {
                print "<b>" ."Video Title: " ."</b>" .$row['vname'] ."<br>";
                print "<video" ." width=" ."320" ." height=" ."240" ." controls>" ."<"
."source" ." src" ."=" ."videos/" .$row['videofile'] ." type" ."=" ."video/mp4" .">" ."Your
browser does not support video" ."</video>";
                print "<b>" ." Category: " ."</b>" .$row['category'] ."<b>" ."Video ID=
" ."</b>" .$row['vid'] ."<br>" ."<hr>";
        }
        mysqli_close($con);
?>
```

File config.php establish connection to database. This PHP code prints the <video> tag till while loop is true. The SQL query fetches the video attributes from the database, which are passed in object $result. The variable $row gets the values from $result and governs the while loop.

5.1.3 Code to search videos

Fig. 5.3

```
<?php
require('../config.php');
$sql="INSERT INTO search (sitem)
VALUES
('$_GET[sitem]')";
if (!mysqli_query($con,$sql))
  {
  die('Error: ' . mysqli_error());
  }
  $search=$_GET['sitem'];
$result = mysqli_query($con,"SELECT  vname, videofile, category, vid FROM videos
where vname like '%$search%' order by vid desc");
while($row = mysqli_fetch_array($result))
  {
    print "<video" ." width=" ."320" ." height=" ."240" ." controls>" ."<" ."source" ." src"
."=" ."../videos/" .$row['videofile'] ." type" ."=" ."video/mp4" .">" ."Your browser does not
support video" ."</video>";
              print "<b>" ."Video Title: " ."</b>" .$row['vname'] ."<b>" ." Category: " ."</b>"
.$row['category'] ."<b>" ." Video Id: " ."<b>" .$row['vid'] ."<br>";
  }
mysqli_close($con);
?>
```

The above code uses $_GET( ), which takes the value from search form. This function then passes the value to SQL query, which fetches the video attributes from the database. The rest of work is done by ‘print’ function using <video> tag.

5.1.4 Code to take feedback

Fig. 5.4

```
<?php
session_start();
require('../config.php');
$sql="INSERT INTO feedback (feed,vid,usrnm)
VALUES
('$_POST[feed]','$_POST[vid]','$_SESSION[usrnm]')";
if (!mysqli_query($con,$sql))
  {
  die('Error: ' . mysqli_error());
  }
echo "Feedback Successful";
header('Location: ../feedback_done.php');
mysqli_close($con);

?>
```

The above code takes the values from the feedback form and using $_POST( ) function passes it to the SQL query. Using PHP the values are send to the server.

5.1.5 Code to upload videos

Fig. 5.5

```
<?php
session_start();
$_path = "../videos/";
$_path = $_path . basename( $_FILES['videofile']['name']);
$_orgnm = basename( $_FILES['videofile']['name']);
if(move_uploaded_file($_FILES['videofile']['tmp_name'], $_path)) {
echo "Successful upload of ". basename( $_FILES['videofile']['name']);
print "<br>" ."CLICK" ."<a href=" ."../index.php" .">" ."HERE" ."</a>" ."TO GO TO
HOME PAGE";
require('../config.php');
$sql="INSERT INTO videos (vname,videofile,category,usrnm)
VALUES
('$_POST[vname]','$_orgnm','$_POST[category]','$_SESSION[usrnm]')";
if (!mysqli_query($con,$sql))
  {
  die('Error: ' . mysqli_error());
  }
echo ".";
mysqli_close($con);
} else{
echo "Error when uploading the video.";
}
?>
```

The above code takes the values from upload form. It uploads the video file from client to ‘videos’ folder on the server. It also stores the video attributes to the database for later use.

5.1.6 Code to register

Fig. 5.6

```
<?php
 session_start();
 require('../config.php');
$sql="INSERT INTO users (name,usrnm,pass)
VALUES
('$_SESSION[name]','$_SESSION[usrnm]','$_SESSION[pass]')";
if (!mysqli_query($con,$sql))
  {
  die('Error: ' . mysqli_error());
  }
echo "Registration Successful";
header('Location: ../regdone.php');
mysqli_close($con);
?>
```

The code above takes the values stored in session variable from the registration form and inserts those values in the ‘users’ table in the database.

5.1.7 Code to Login

Fig. 5.7

```
<?php
session_start();
 require('../config.php');
$usr=$_POST['usrnm'];
$pas=$_POST['pass'];
   // Check The DataBase
$result = mysqli_query($con,"SELECT * FROM users WHERE usrnm = '$usr' AND pass
= '$pas'");
$row = mysqli_fetch_array($result);
 if ($row==null)
{
      //The Username and Pass Does Not Matches
$_SESSION['error'] = 'Unauthorized Access ';
header('Location: ../tryagain.php');
 }
else
{
    // Direct to session start
$_SESSION['authorised']=true;
$_SESSION['usrnm']=$usr;
header('Location: ../usrpannel.php');
}
?>
```

The above code takes values from login form using $_POST( ) function. The values are passed to a SQL query which checks whether the username or password matches or not. If the credentials are true then the session is started and the user is logged in.

5.1.8 Code to Logout

Fig. 5.8

```
<?php
session_start();
unset($_SESSION['authorised']);
unset($_SESSION['usrnm']);
header('Location: ../login.php');
?>
```

The above unsets the session variables to logout the user. After successful unset of session variables it also redirects the user to login page.

5.2 Database tables

- Feedback

Table5.1

Field	Type	Null	Key	Default	Extra
feed	varchar(500)	YES		*NULL*	
vid	int(11)	YES		*NULL*	
usrnm	varchar(20)	YES		*NULL*	
fid	int(11)	NO	PRI	*NULL*	auto_increment

- Users

Table5.2

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		*NULL*	
usrnm	varchar(20)	YES		*NULL*	
pass	varchar(20)	YES		*NULL*	
uid	int(11)	NO	PRI	*NULL*	auto_increment

- Videos

Table5.3

Field	Type	Null	Key	Default	Extra
vname	varchar(200)	YES		*NULL*	
videofile	varchar(200)	YES	MUL	*NULL*	
vid	int(11)	NO	PRI	*NULL*	auto_increment
category	varchar(20)	NO		*NULL*	
usrnm	varchar(20)	NO		*NULL*	

- Search

Table5.4

Field	Type	Null	Key	Default	Extra
sitem	varchar(20)	YES		*NULL*	
sid	int(11)	NO	PRI	*NULL*	auto_increment

5.3 Stored Procedures (PL/SQL)

- To fetch videos to display on home page.

Fig. 5.9

```
SELECT  vname, videofile, category, vid FROM videos order by vid desc limit 10
```

- To fetch videos to display on tutorials page.

Fig. 5.10

```
SELECT  vname, videofile, vid FROM videos where category='tutorials' order by
vid desc
```

- To fetch videos to display on music page.

Fig. 5.11

```
SELECT  vname, videofile, vid FROM videos where category='music' order by vid
```

- To fetch videos to display on short movies page.

Fig. 5.12

```
SELECT  vname, videofile, vid FROM videos where category='shortmovies' order
by vid desc
```

- To fetch videos to display on sports page.

Fig. 5.13

```
SELECT  vname, videofile, vid FROM videos where category='sports' order by vid
```

- To fetch videos to display on search page.

Fig. 5.14

```
SELECT  vname, videofile, category, vid FROM videos where vname like
'%$search%' order by vid desc
```

- To verify registration data for uniqueness of username.

Fig. 5.15

```
SELECT * FROM users where usrnm='$usrnm'
```

- To add registration data to users table in database.

Fig. 5.16

```
INSERT INTO users (name,usrnm,pass) VALUES
('$_SESSION[name]','$_SESSION[usrnm]','$_SESSION[pass]')
```

- To verify login data.

Fig. 5.17

```
SELECT * FROM users WHERE usrnm = '$usr' AND pass = '$pas'
```

Testing

6.1 Black box testing

6.1.1 As a result of black box testing the first case was related to an incorrect function. The error was in the registration process. In registration process there are three fields ‘Name’, ‘User Name’ and ‘Password’. Out of these three fields the field ‘User Name’, work as a candidate key to identify users, so it should always be a unique value. But during testing it was found that two users can register with a same user name.

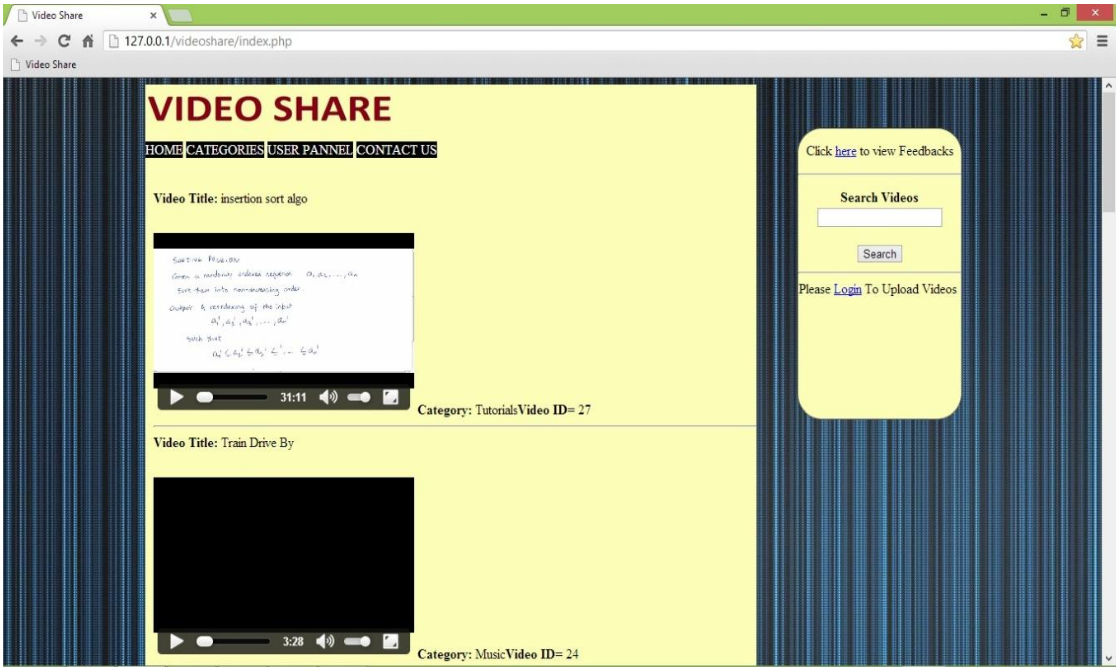
6.1.2 The second test case was functional error. Regarding upload there was a condition that a user must register and login to videoshare to upload videos. This restriction was applied to identify the user who uploaded the video. The identification can be useful in case a user uploads any illegal or explicit video on the website. During testing it was found that all users were able to upload videos without logging in to videoshare.

6.2 White box testing

6.2.1 It’s the very basic thing that out of two codes with same functionality and accuracy, the code of smaller size or less line of code is favored. As mentioned earlier that Videoshare uses CSS for styling web pages. During testing it was found that the same CSS code is repeated in every file, this was increasing the file size and line of codes. So it was suggested that that to implement CSS part in a separate file and to include that file in each webpage.

Output Screen

7.1 Home page


Video Share
127.0.0.1/videoshare/index.php
Video Share
VIDEO SHARE
HOME CATEGORIES USER PANNEL CONTACT US
31:11
3:28
Search

Video Title: insertion sort algo

Category: Tutorials**Video ID**= 27

Video Title: Train Drive By

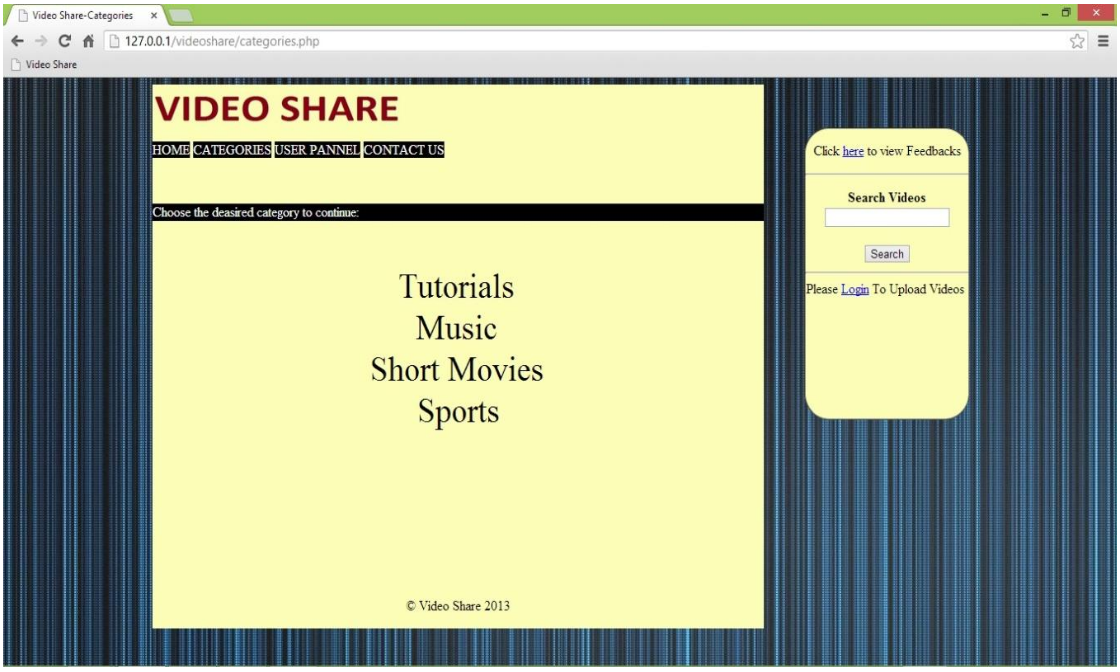
Category: Music**Video ID**= 24

Click here to view Feedbacks

Search Videos

Please Login To Upload Videos

7.2 Categories page


Video Share-Categories
127.0.0.1/videoshare/categories.php
Video Share
VIDEO SHARE
HOME CATEGORIES USER PANNEL CONTACT US
Search

Choose the deasired category to continue:

Tutorials
Music
Short Movies
Sports

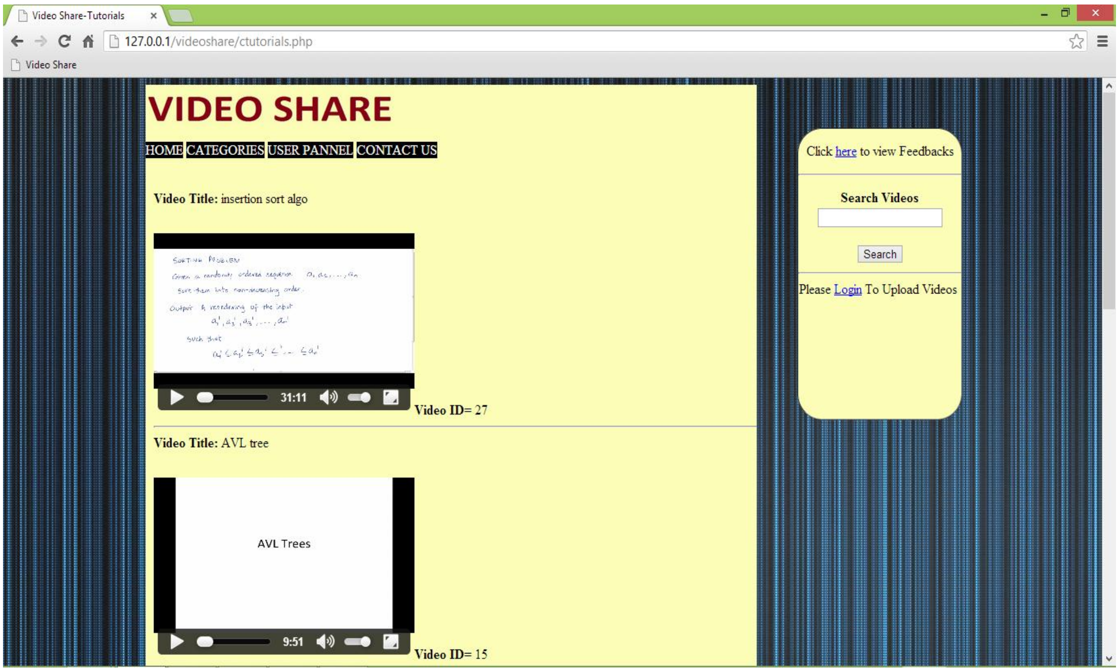
© Video Share 2013

Click here to view Feedbacks

Search Videos

Please Login To Upload Videos

7.3 Tutorials page



Video Title: insertion sort algo

Video ID= 27

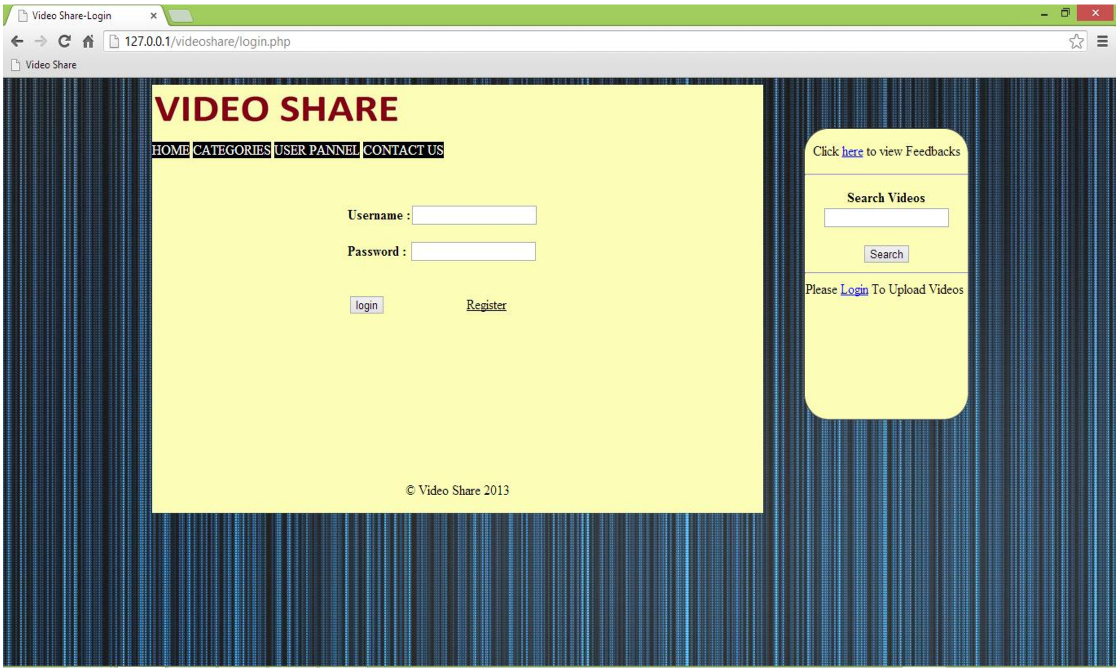
Video Title: AVL tree

Video ID= 15

Click here to view Feedbacks

Please Login To Upload Videos

7.4 Login page



Username :

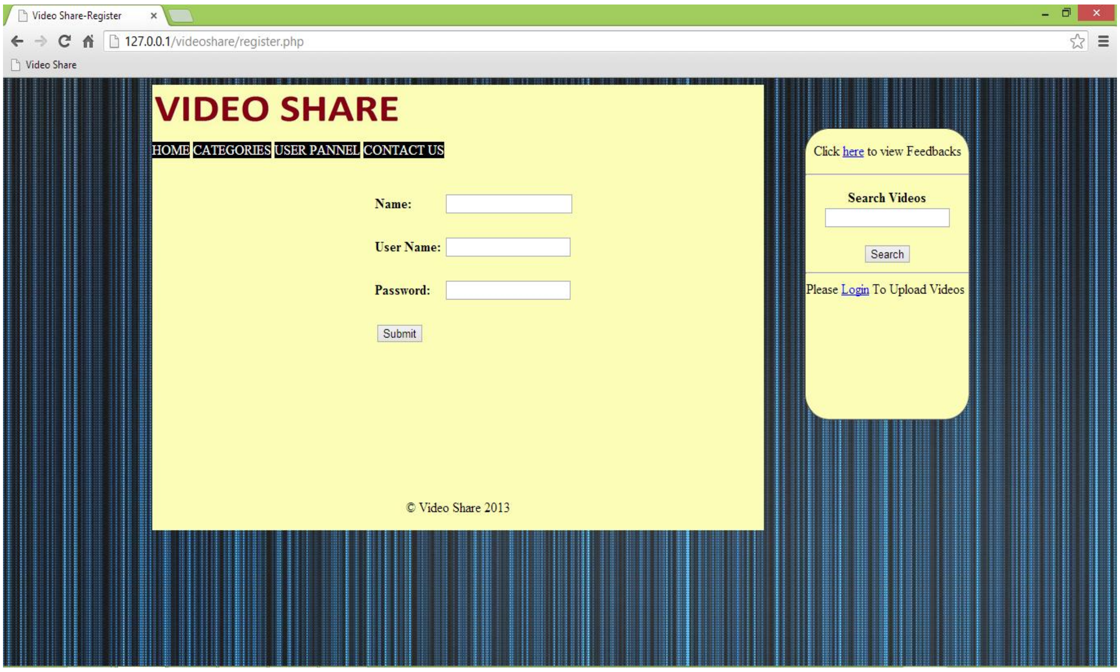
Password :

Click here to view Feedbacks

Please Login To Upload Videos

© Video Share 2013

7.5 Registration page



Video Share-Register

127.0.0.1/videoshare/register.php

Video Share

VIDEO SHARE

HOME CATEGORIES USER PANNEL CONTACT US

Name:

User Name:

Password:

Submit

© Video Share 2013

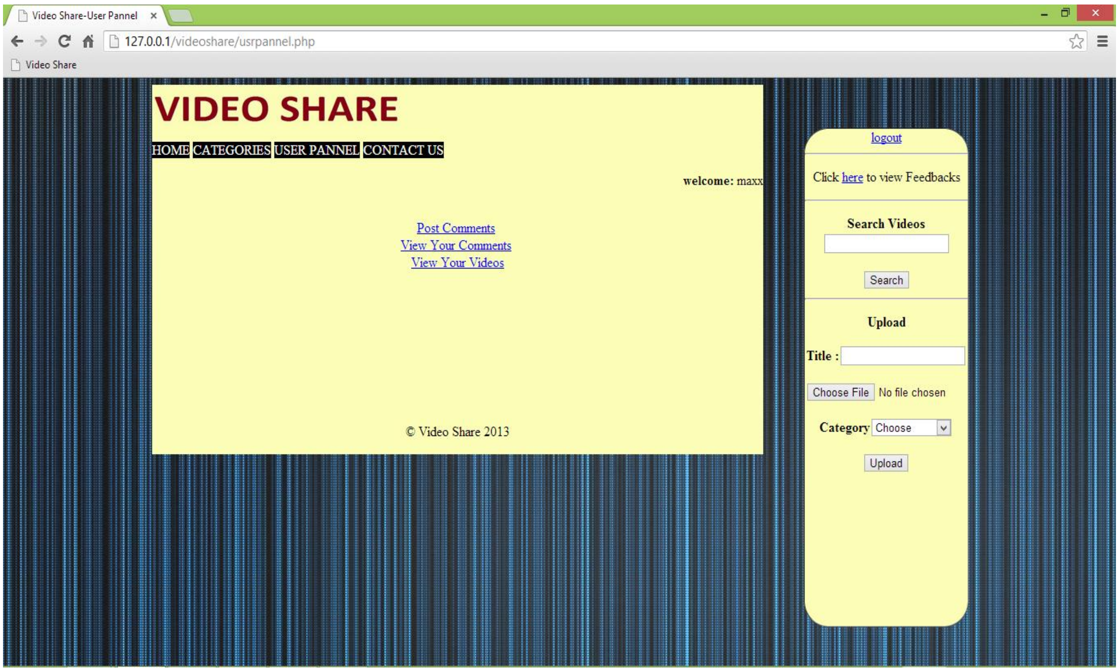
Click here to view Feedbacks

Search Videos

Search

Please Login To Upload Videos

7.6 User panel



Video Share-User Pannel

127.0.0.1/videoshare/usrpannel.php

Video Share

VIDEO SHARE

HOME CATEGORIES USER PANNEL CONTACT US

welcome: maxx

Post Comments

View Your Comments

View Your Videos

© Video Share 2013

logout

Click here to view Feedbacks

Search Videos

Search

Upload

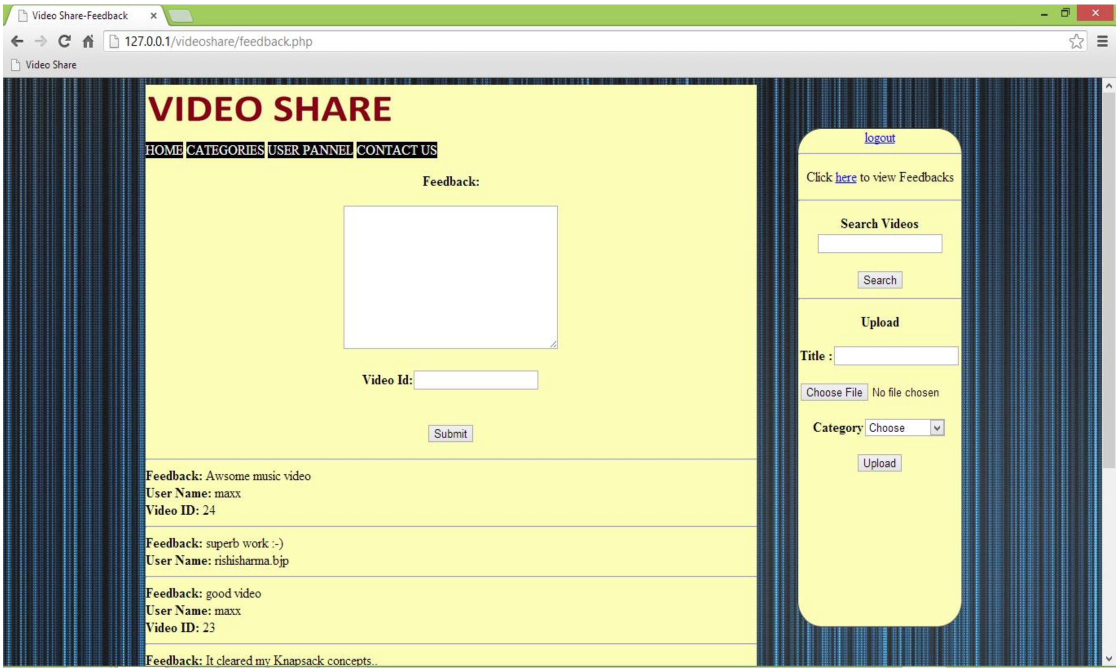
Title :

Choose File No file chosen

Category Choose

Upload

7.7 Feedback page


Video Share-Feedback
127.0.0.1/videoshare/feedback.php
Video Share
VIDEO SHARE
HOME CATEGORIES USER PANNEL CONTACT US
logout
Search
Choose File No file chosen
Choose
Upload
Submit

Feedback:

Video Id:

Click here to view Feedbacks

Search Videos

Upload

Title :

Category Choose

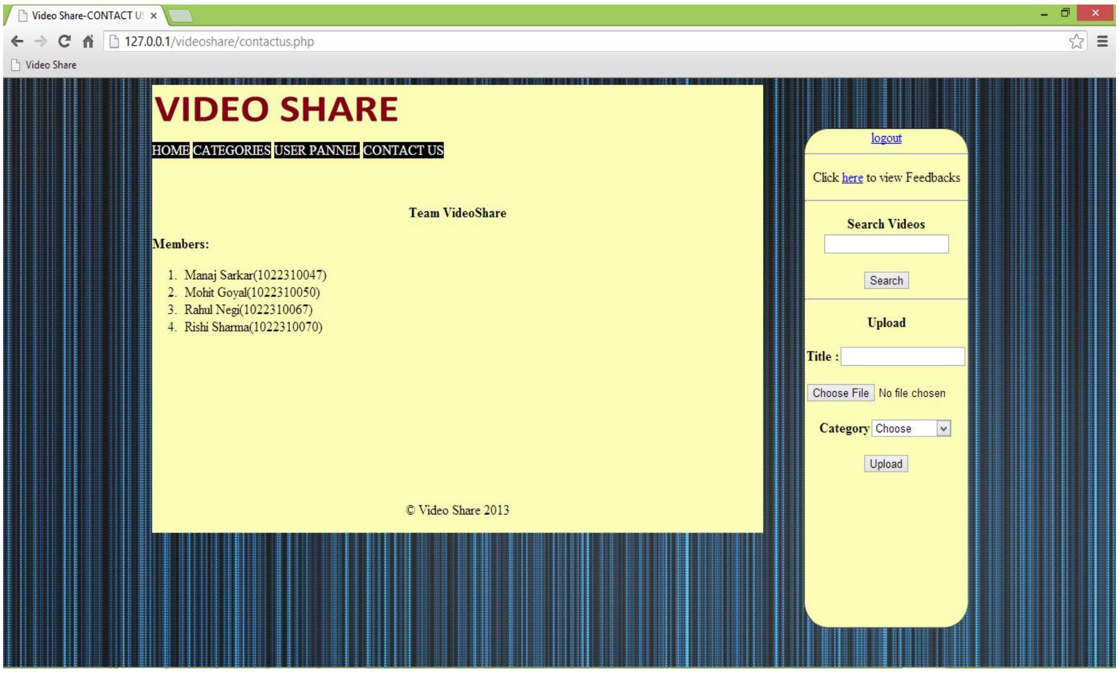
Feedback: Awsome music video
User Name: maxx
Video ID: 24

Feedback: superb work :-)
User Name: rishisharma.bjp

Feedback: good video
User Name: maxx
Video ID: 23

Feedback: It cleared my Knapsack concepts..

7.8 Contact Us pagef


Video Share-CONTACT U
127.0.0.1/videoshare/contactus.php
Video Share
VIDEO SHARE
HOME CATEGORIES USER PANNEL CONTACT US
logout
Search
Choose File No file chosen
Choose
Upload

Click here to view Feedbacks

Search Videos

Upload

Title :

Category Choose

Team VideoShare

Members:

1. Manaj Sarkar(1022310047)
2. Mohit Goyal(1022310050)
3. Rahul Negi(1022310067)
4. Rishi Sharma(1022310070)

© Video Share 2013

References/Bibliography

8.1 Book

- Pankaj Sharma, Introduction to Web Technology, S.K. Kataria & Sons, 2012.

8.2 Website

- www.w3schools.com
- www.stackoverflow.com