PERFECT PICTURE POOL

A PROJECT REPORT

Submitted By

Deepanshu – 2100290140056
Shivani Pandey – 2100290140130
Shipra Aggarwal – 2100290140125
Rashi Ruhela – 2100290140112

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of

Dr. Vipin Kumar
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(FEBRUARY - 2022)

CERTIFICATE

Certified that **Deepanshu – 2100290140056, Shipra Aggarwal – 2100290140125, Shivani Pandey – 2100290140130, Rashi Ruhela – 2100290140112** has/ have carried out the project work having "**Perfect Picture Pool**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Deepanshu – 2100290140056
Shivani Pandey – 2100290140130
Shipra Aggarwal – 2100290140125
Rashi Ruhela – 2100290140112

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

DR. VIPIN KUMAR
Associate Professor
Department of Computer Applications
KIET Group of Institutions,
Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Perfect Picture Pool is an Open-Source Website, which is based on creativity/thinking among users and people. In this website, they are used to upload different types of images on it and with a particular time of period. The image with most likes or comment declares as winner for the given contest which is declared by admin, and then some rewards point or gift would be distributed to that winner user. Admin is set different types of contests on website in which anyone can participate. Anyone can vote on any image that user upload whether he is register or not. Anyone can give feedback, suggestion and also, they can leave the comment on the image. By this website, User can know their skills and where they have to improve it and also, they upgrade their picture clicking skills. User also motivate by this website when they saw their results.

FEATURES

- Admin Login.
- User Registration.
- User Login.
- Manage contest and update user information.
- Update Contest.
- Display maximum number of likes with username.
- Result

List of Chapters

Page No.

Chapter 1 - Introduction 5

1.1 Project description 6

1.2 Project Scope 7

1.3 Hardware / Software used in Project 8

Chapter 2 Feasibility Study 9

2.1 Operational feasibility 9

2.2 Technical Feasibility 10

2.3 Economical Feasibility 11

Chapter 3 Database Design 12

3.1 Database Tables 12

3.2 Use Case Diagram 16

3.3 Sequence Diagram 19

3.4 Data Flow Diagram (DFD) 22

3.5 Flow Chart 27

Chapter 4 Form Design 28

4.1 Input / Output Form (Screenshot) 28

Chapter 5 Coding 36

5.1 Module wise code 36

Chapter 6 Testing 100

6.1 Test Case-1 (White-Box Testing) 100

6.2 Test Case-2 (White-Box Testing) 102

Conclusion 104

Chapter – 1

Introduction

The Perfect Picture Pool is designed to replace the manual process through an application. Through this application, the admin who is the main user of the application will manage all the activities like adding competition name, date, and the type and can view the result.

The main objective of the Perfect Picture Pool application is to help the competitor to face the live audience around the world and User from different corners of the world can look at the beautiful photographs and can vote for their favorite photo.

This project provides a simple and beautiful interface for the admin, Photographer as well as a voter. Just Admin needs to add the competition name, date, type, and view the result. All the information will be stored in the database and that will help to maintain all the information about Photographs and results.

Users and Admin will be able to keep track of their competition related activity with this application.

1.1 Project Description

The main objective is to build a secured, robust Photo Competition Application. It maintains the record of the photographers, voters, competition results and, etc. so that it would be easy to access at any time 24*7.

Each module is built independent of each other. This application is developed for showcasing the photographs for competition purposes. Admin is the main user of the website who will conduct competition and will manage every activity like adding competition.

This application will help Users to vote online for their favorite photographs Also it will help the photographer to showcase their skills to people around the world. Through this application one can vote, can add Photographs, and can attend competition online instead of visiting the location. Every process will be online as well as automated.

1.2 Project Scope

Perfect Picture Pool is an Open-Source Website, which is based on creativity/thinking among users and people. In this website ,they are used to upload different types of images on it and with a particular time of period. The image with most likes or comment declares as winner for the given contest which is declared by admin, and then some rewards point or gift would be distributed to that winner user. Admin is set different types of contests on website in which anyone can participate. Anyone can vote on any image that user upload whether he is register or not. Anyone can give feedback, suggestion and, they can leave the comment on the image. By this website, User can know their skills and where they must improve it and, they upgrade their picture clicking skills. User also motivate by this website when they saw their results.

1.3 Hardware & Software Requirement:

Hardware Interfaces

- Minimum Hardware requirement
- Processor: Pentium IV or Higher
- RAM: 4 GB or Higher
- Operating System: Windows 10
- System Type: 32-bit operating system
- Hard disk: 512 GB

Software Interfaces

- Minimum Software requirement
- Eclipse
- MySQL
- Java (JSP and Servlet)
- Apache Tomcat Server

All these types of software automatic configure inside operating system after installation it is having Java, MYSQL, Apache and operating system base configuration file, it doesn't need to configure manually.

Chapter – 2

Feasibility Study

2.1 Operational Study

This project will be developed on computer, so first check whether the technology is technically available or not. Now a day's computer interaction with any job becomes common for any kind of job or work.

And because of increasing usage of Computer, Computer is also available with a variety of hardware. Vendors can fulfill any type of hardware requirement. The whole project is developed by some special tools or by using languages and databases, which are also available in a variety.

Preliminary investigation of a system examines the feasibility of a system that is useful to an organization. It is the first phase of system development.

The main objective of this phase is to identify the current deficiencies in the user's environment and to determine which existing problem are going to be solve in proposed system and also which new function needs to be added in proposed system.

An important outcome of such preliminary investigation is to determine whether the system that will meet all needed requirements.

Thus, three tests are carried out on the system namely operation, technical and economical.

Any project is beneficial if and only satisfies the organization requirement. For any new system setup, it only meets to be communicated and work the other supporting system.

The new system meets all existing operations since it provides right information at a right time to the right user. A Leigh man can easily operate with the system.

2.2 Technical Feasibility

Technical Feasibility examines whether the technology needed is available and if it is available then it feasible to carry out all project activities.

The technical needs of a system include:

- The facility to produce outputs in each time.
- Ability to process large number of transactions at a particular speed.
- Giving response to users under certain conditions.

The technology needed for our system is mainly:

- Latest version of browsers.
- Any operating system.

These technologies are available which helps to carry out the system efficiently.

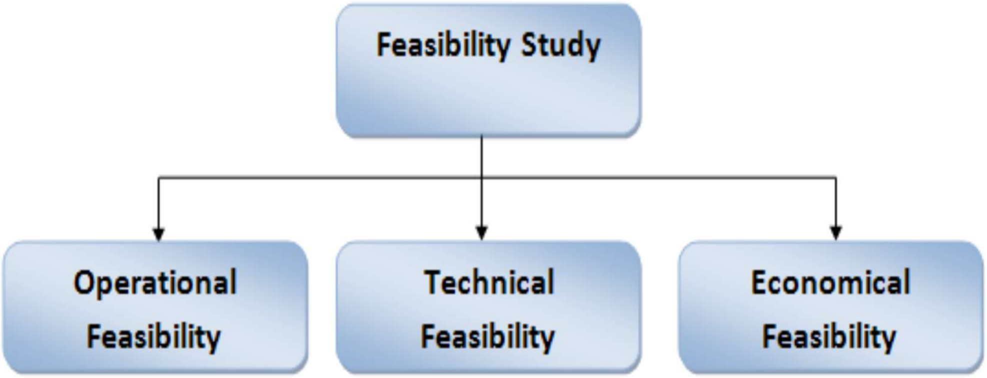
2.3 Economical Feasibility

Economical feasibility of a system examines whether the finance is available for implementing the new system and whether the money spent is recoverable the satisfaction.

The cost involves is in designing and developing a good investment for the organization.

Thus, hardware requirements used for proposed system are very standard. Moreover, by making use of proposed system to carry out the work speedily will increase and also saves the valuable time of an organization.

In the proposed system the finance is highly required for the installation of the software's which can also be recovered by implementing a better system.

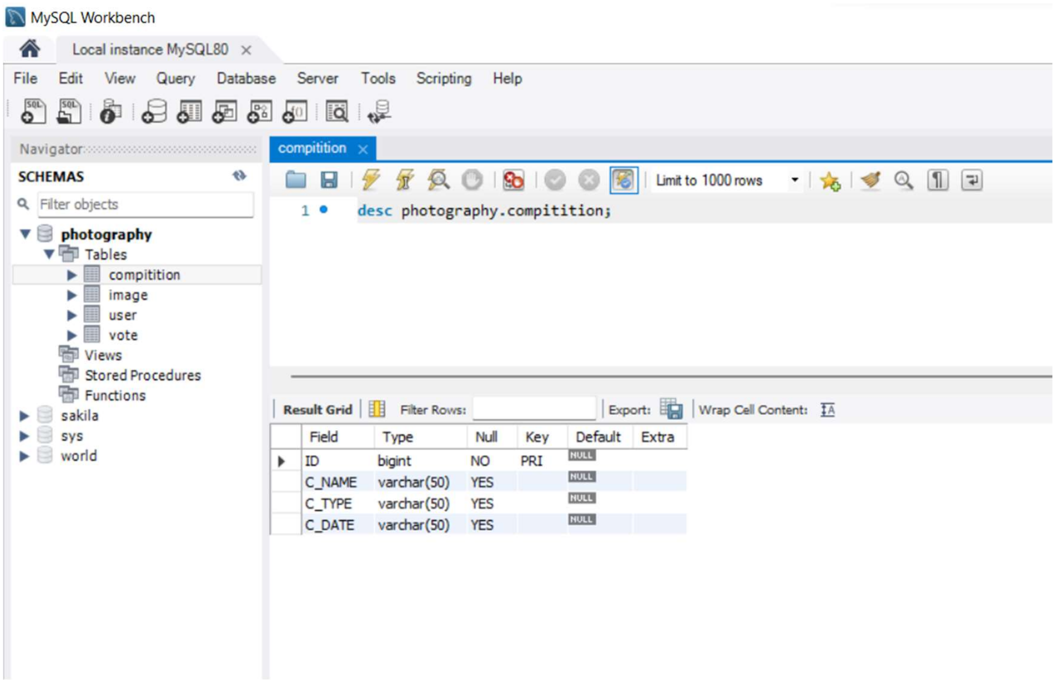


Chapter – 3

Database Design

3.1 Database Table

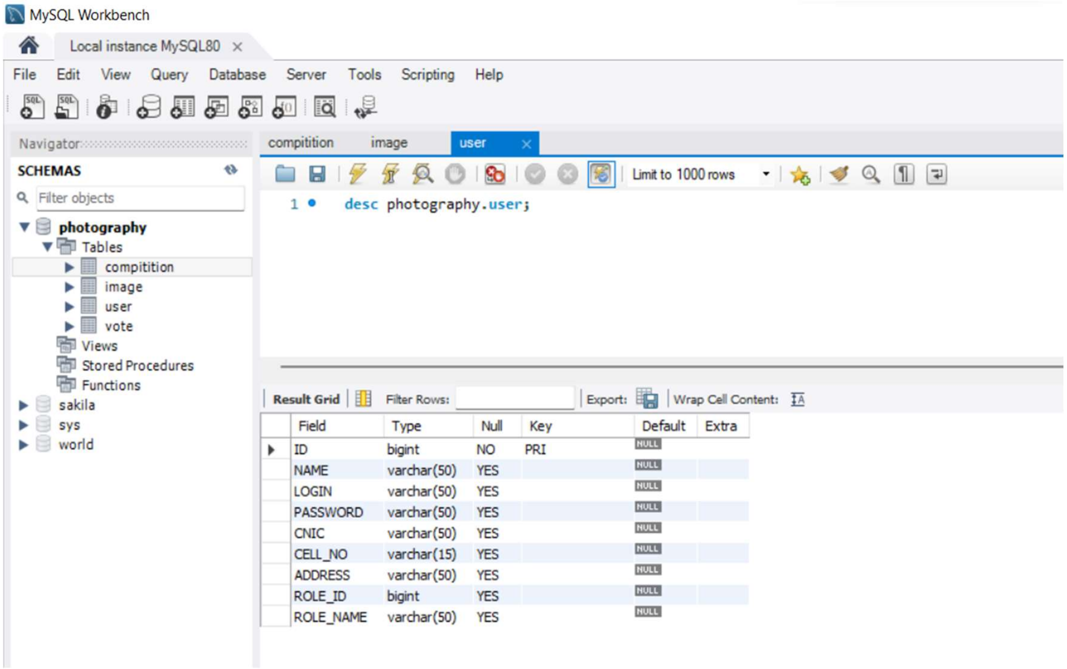
Competition Table



```
desc photography.compitition;
```

Field	Type	Null	Key	Default	Extra
ID	bigint	NO	PRI	NULL	
C_NAME	varchar(50)	YES		NULL	
C_TYPE	varchar(50)	YES		NULL	
C_DATE	varchar(50)	YES		NULL	

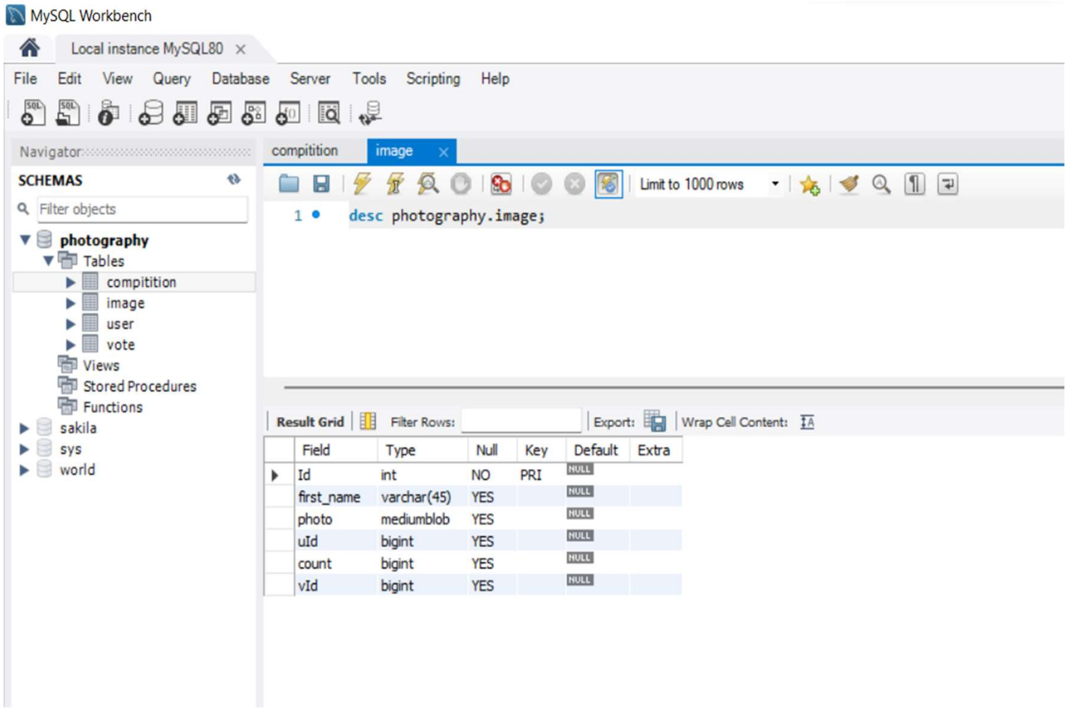
User Table



```
desc photography.user;
```

Field	Type	Null	Key	Default	Extra
ID	bigint	NO	PRI	NULL	
NAME	varchar(50)	YES		NULL	
LOGIN	varchar(50)	YES		NULL	
PASSWORD	varchar(50)	YES		NULL	
CNIC	varchar(50)	YES		NULL	
CELL_NO	varchar(15)	YES		NULL	
ADDRESS	varchar(50)	YES		NULL	
ROLE_ID	bigint	YES		NULL	
ROLE_NAME	varchar(50)	YES		NULL	

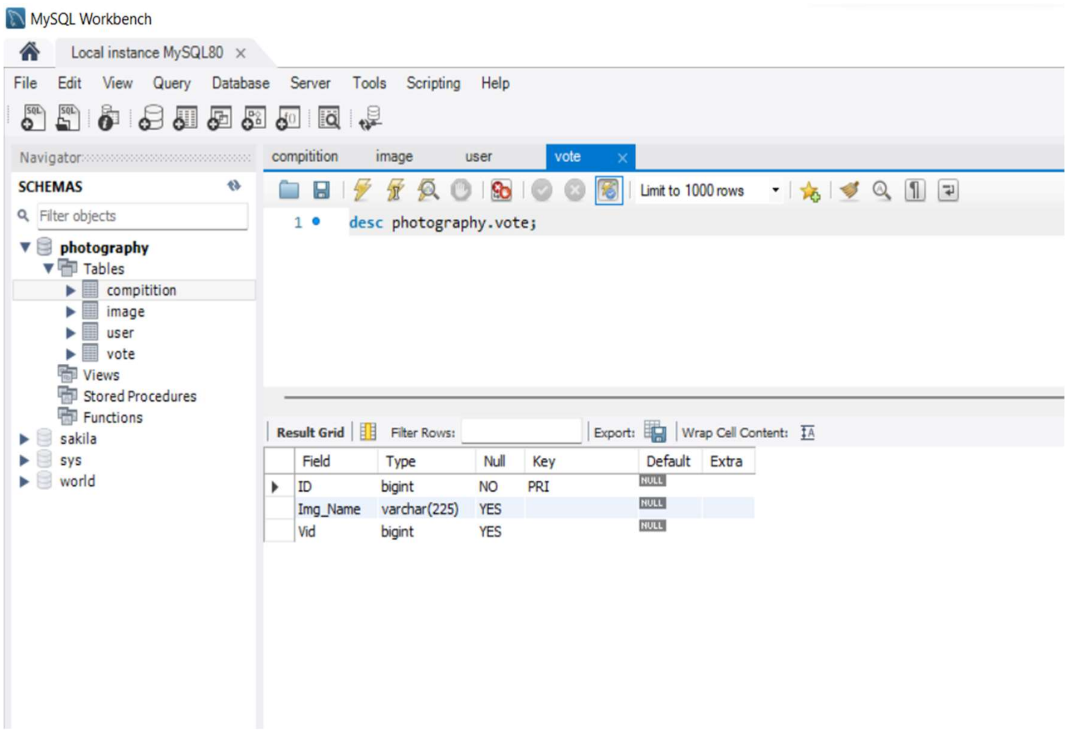
Image table



```
desc photography.image;
```

Field	Type	Null	Key	Default	Extra
Id	int	NO	PRI	NULL	
first_name	varchar(45)	YES		NULL	
photo	mediumblob	YES		NULL	
uId	bigint	YES		NULL	
count	bigint	YES		NULL	
vId	bigint	YES		NULL	

Vote Table

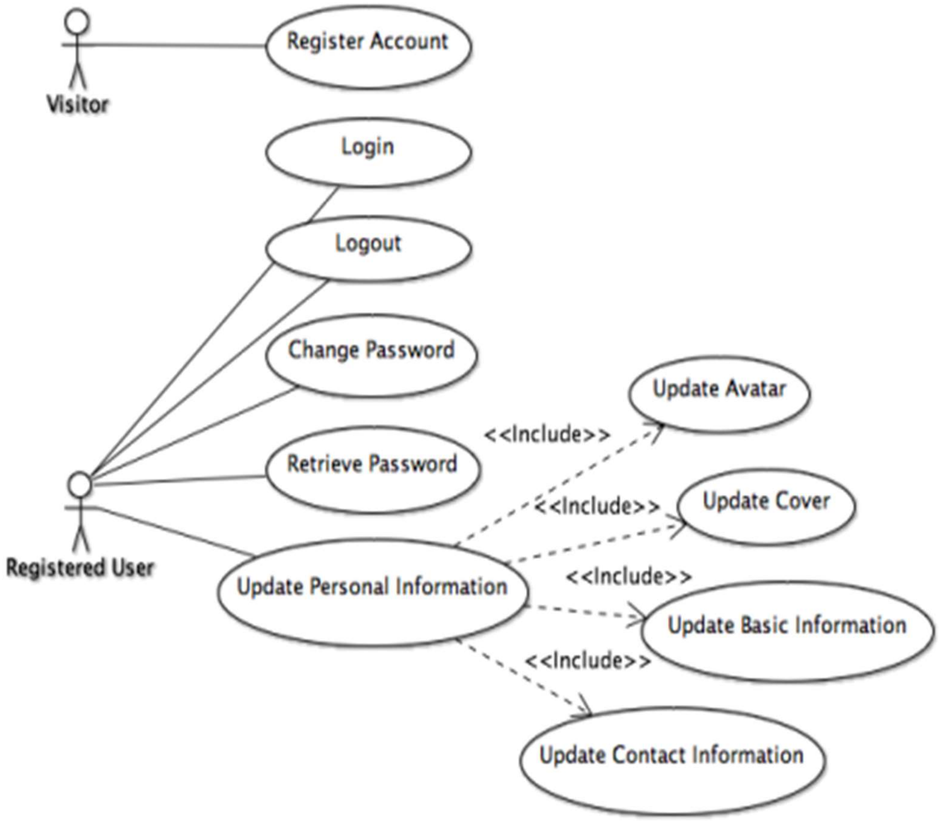


```
desc photography.vote;
```

Field	Type	Null	Key	Default	Extra
ID	bigint	NO	PRI	NULL	
Img_Name	varchar(225)	YES		NULL	
Vid	bigint	YES		NULL	

3.2 Use Case Diagram

- Use Case Diagram for User and Authorization



The functions of each role are shown. They are:

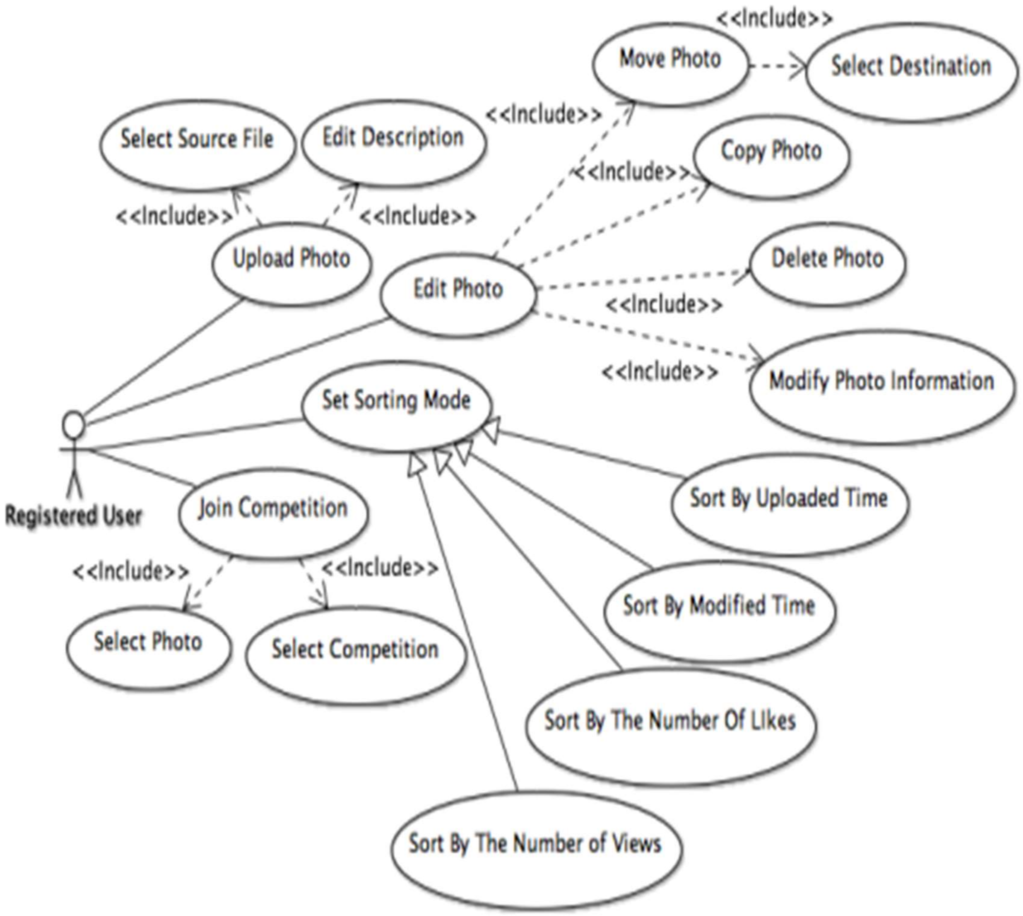
1. Visitors can register an account.

2. Registered users can log in to the system and log out of the system.

3. Registered users can change or retrieve their own password.

4. Registered users can update their personal information, including the avatar, cover, basic information and contact information.

- Use Case Diagram of Photo Upload and Editing



The functions of each role are as follows:

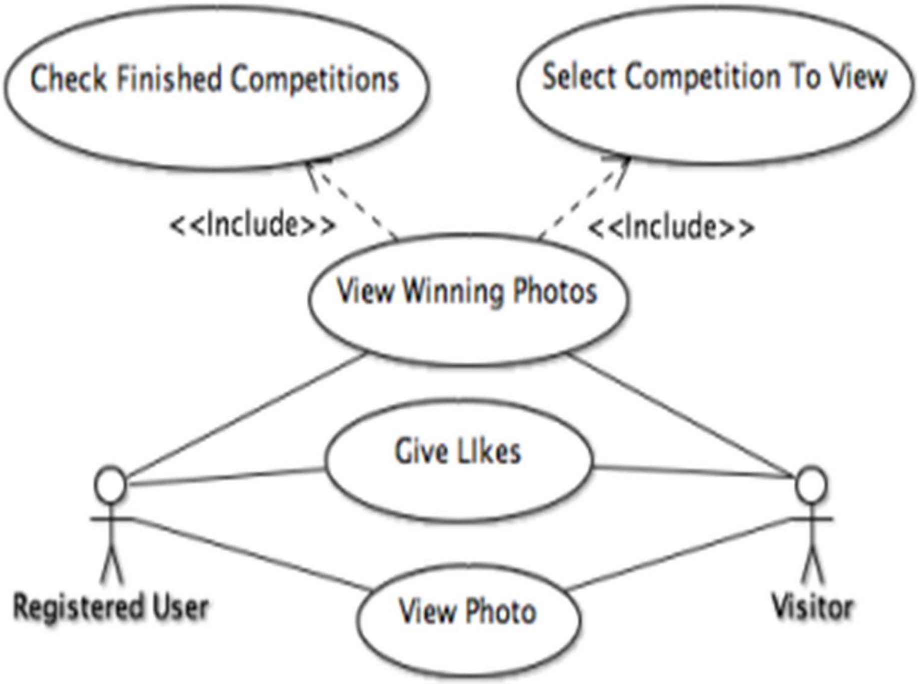
1. Registered users can upload photos into the system, by selecting the source files and inputting information, which describes the photos. Each photo is regarded as resource in the system.

2. Registered users can manage the photos. The operations that can be performed include moving photos, copying photos, deleting photos and modifying photo information.

3. Registered users can also set the sort mode for photos displaying, including sorting by upload time, sorting by modified time, sorting by the 13 number of likes and sorting by the number of views

4. Registered users can take part in competitions after selecting uploaded photo and select specific competition.

- Use case analysis of the Display of Competition Result module.



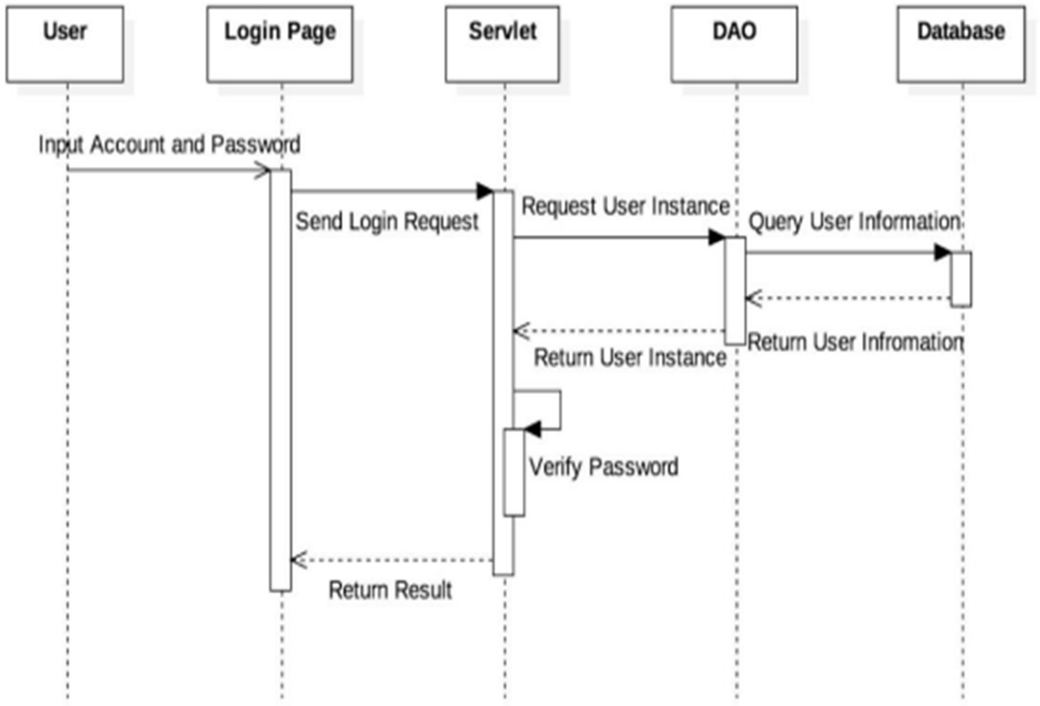
This module aims to display results and offer related operations to users. The use case diagram of the Display of Competition module in which functions of each role are as follows:

1. Registered users and visitors can view the winning photos, including checking the list of finished competitions and selecting one competition to view.

2. Visitors and registered users can browse their favorite photos and give them like's.

3.3 Sequence Diagram

- For Login



The basic flow of user login is as follows:

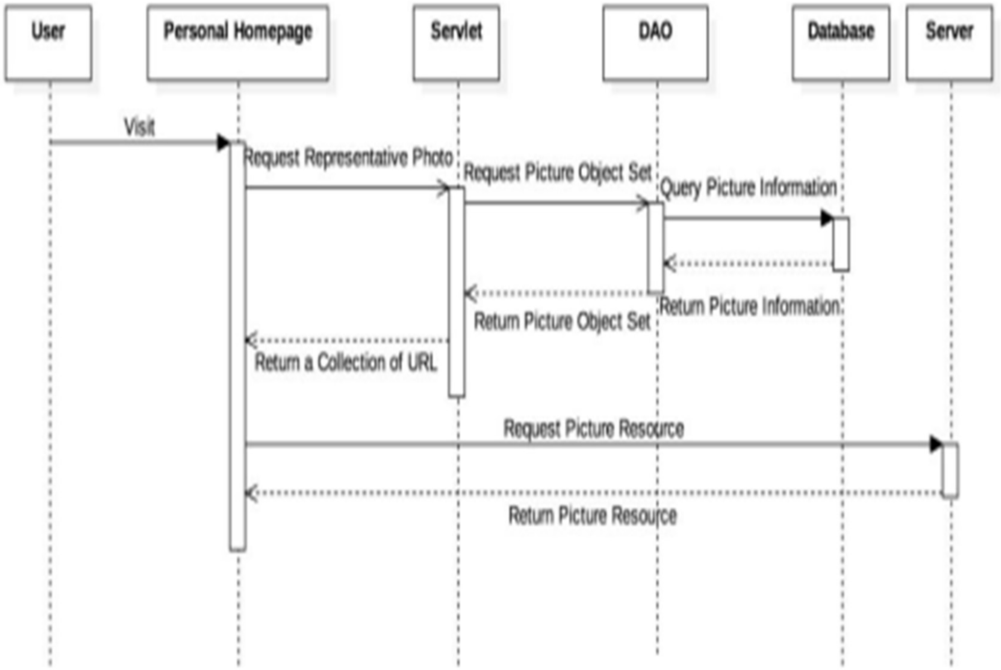
1. The user accesses the login page and input his account and password.

2. The login page sends the login request together with the user's input information to the LoginServlet 3. LoginServlet calls the user DAO (data access object) to perform query operation on the database, and returns the instance object of user

. 4. loginByUserEmail method Compare and verify the input password with the correct password (saved in the instance object).

5. If the password is wrong, login fails and then returns to the login page， reporting an error. Otherwise, login succeeds and then jumps to the user’s personal homepage.

- Sequence Diagram for Personal Information


User
Personal Homepage
Servlet
DAO
Database
Server
Visit
Request Representative Photo
Request Picture Object Set
Query Picture Information
Return Picture Information
Return Picture Object Set
Return a Collection of URL
Request Picture Resource
Return Picture Resource

The basic flow of displaying a user’s representative photos is as follows:

1. The user visits the personal homepage for browsing.

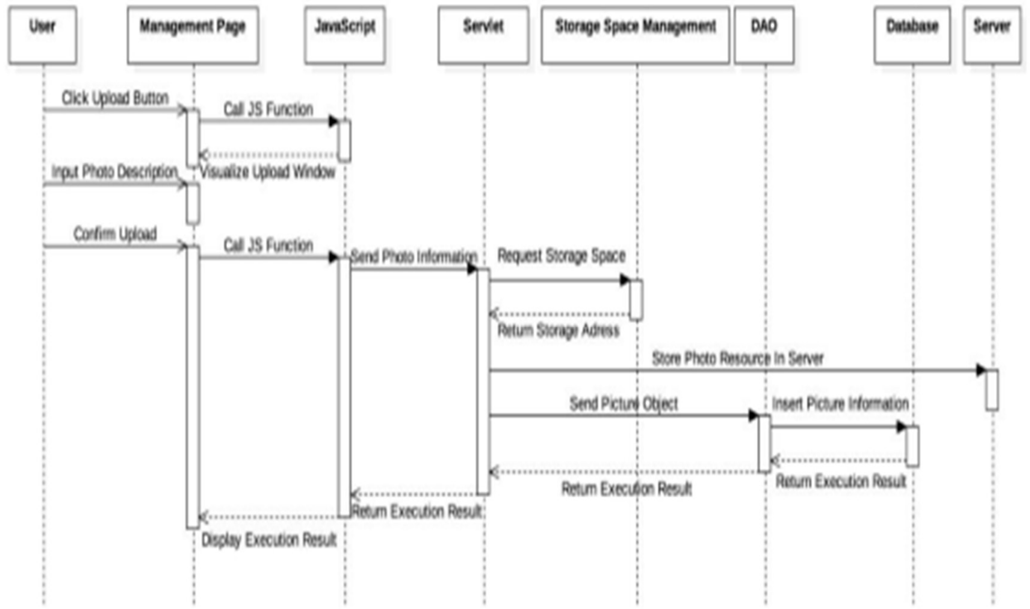
2.Personal homepage requests the HomepageServlet for relevant information about the current user’s representative photos.

3. HomepageServlet calls the user data access object for query operation, and returns a collection of picture objects containing information of the user’s representative photos.

4. All photos’ URLs are obtained from the collection of representative photos objects, and are returned to the personal homepage.

5. The Personal homepage requests to the server, accesses picture resources and displays representative photos using the URLs.

- Sequence Diagram for Uploading Photos



The basic flow of photo uploading is as follows:

1. The user triggers photo upload action in the photo management page.

2. The photo management page calls the relevant JavaScript function to visualize the uploading window. 3. The user selects local photos and gives the related information in the uploading window.

4. When the user chooses to upload photos and confirms, the relevant data and request will be sent to UploadPhotoServletfor processing.

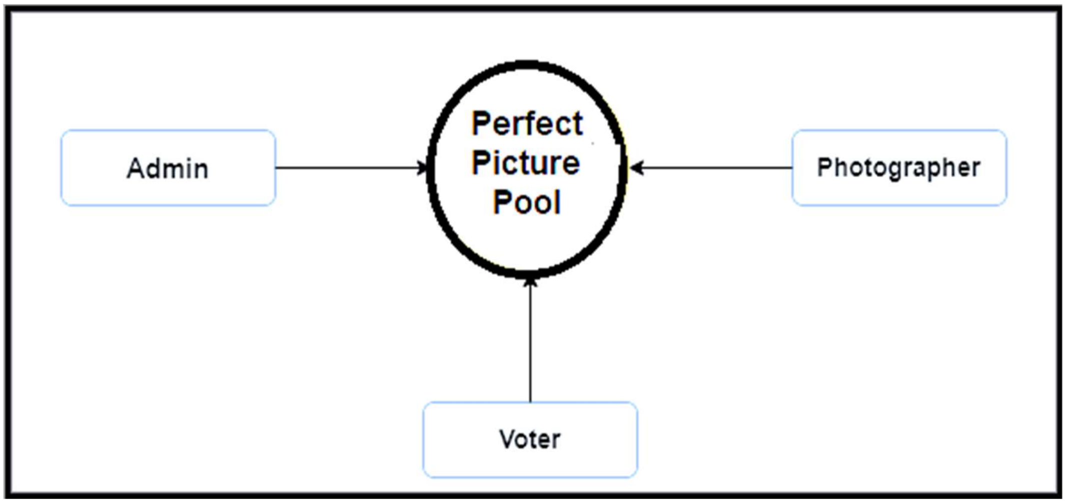
5. The UploadPhotoServletfor requests the storage space management subsystem to allocate the corresponding storage address, and to store the photo resources in the server.

6. By using the photo data access object method, the value of the photo object (relevant information of input photos) is saved in the database.

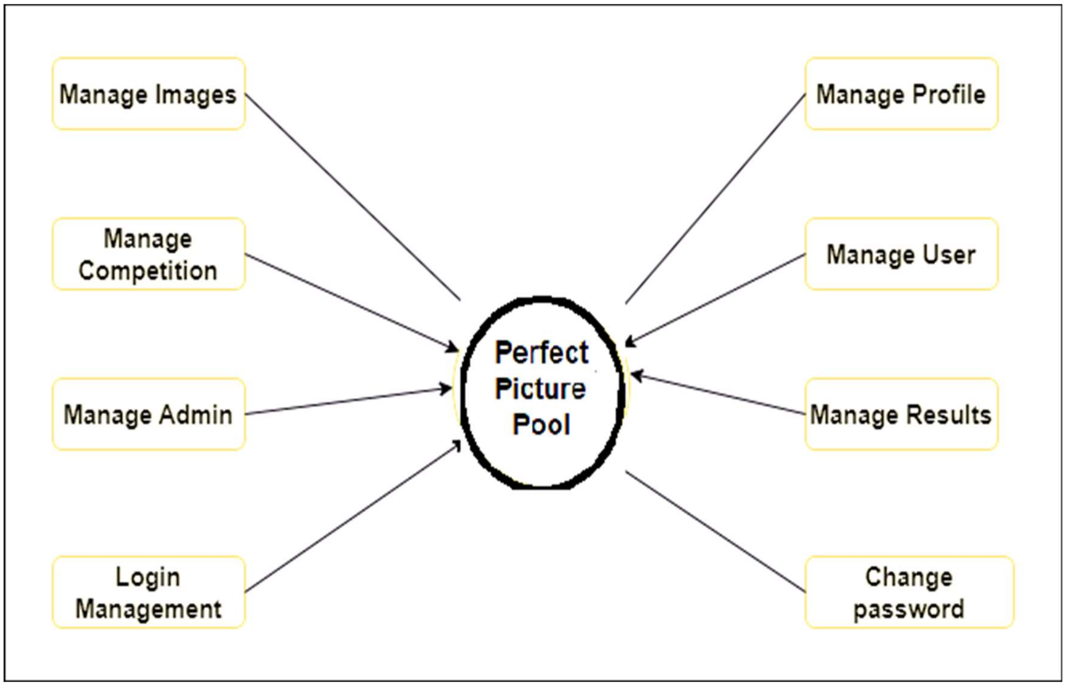
7. Finally, results are returned and displayed. The photo's information and the photo's store address will be saved to the corresponding database tables after the photo has been uploaded successfully (saved in the sever side).

3.4 Data Flow Diagram (DFD)

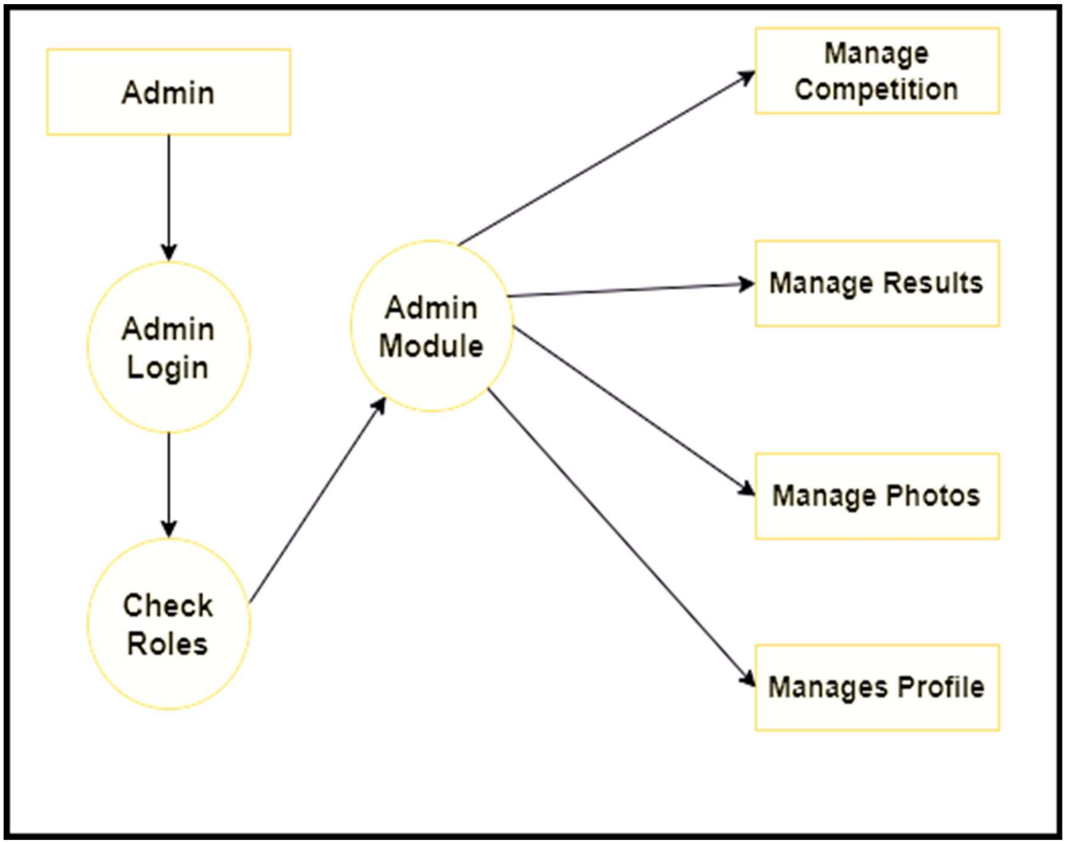
Level 0:



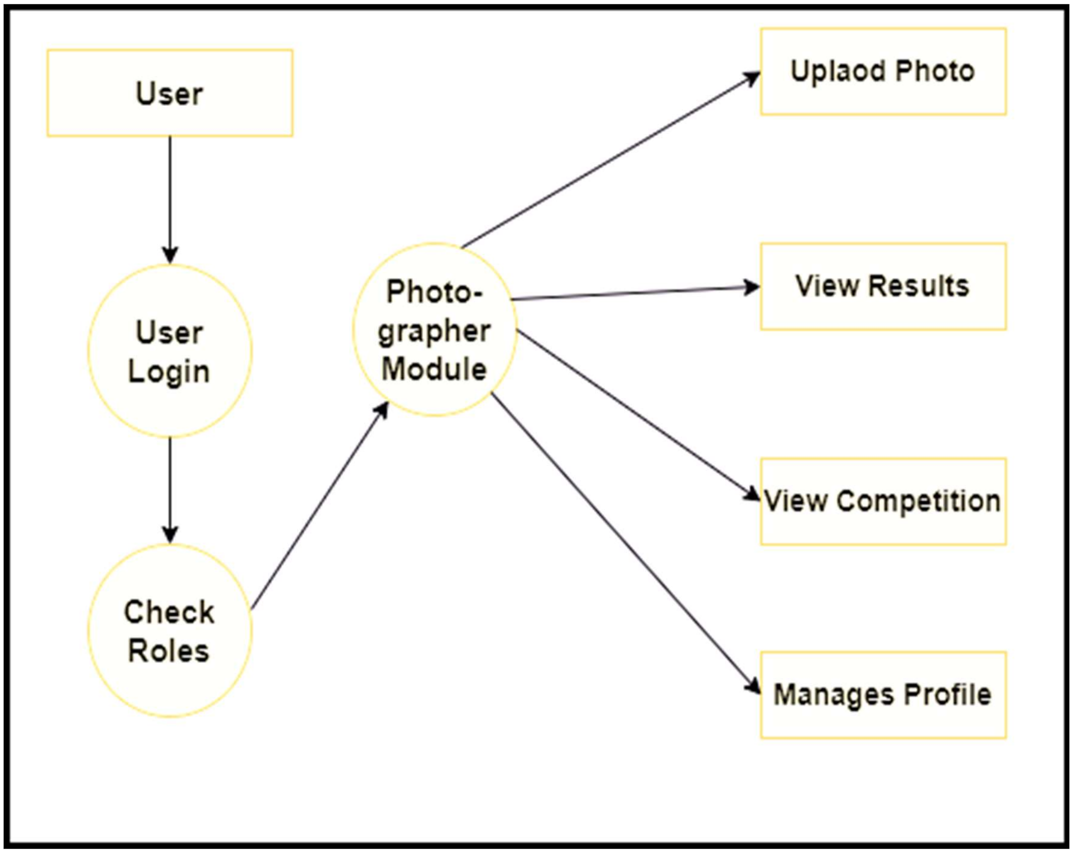
Level 1


Manage Images
Manage Competition
Manage Admin
Login Management
Perfect Picture Pool
Manage Profile
Manage User
Manage Results
Change password

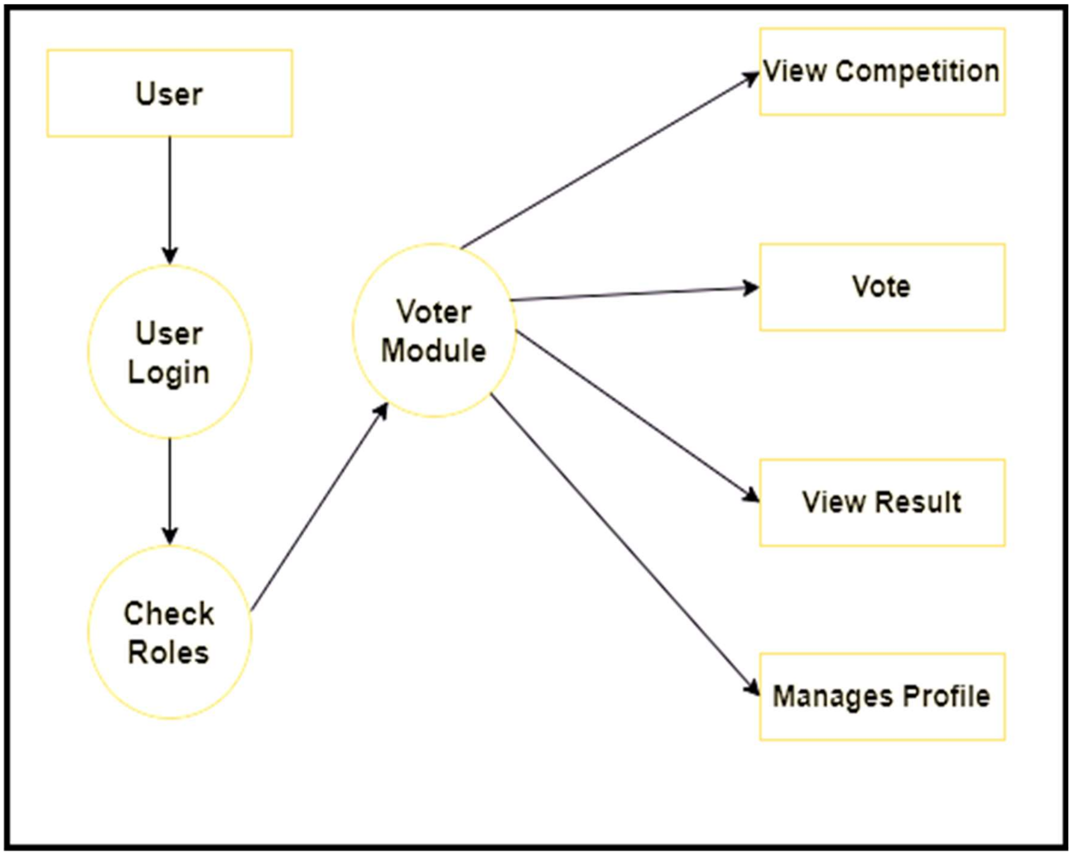
Level 2-DFD Admin



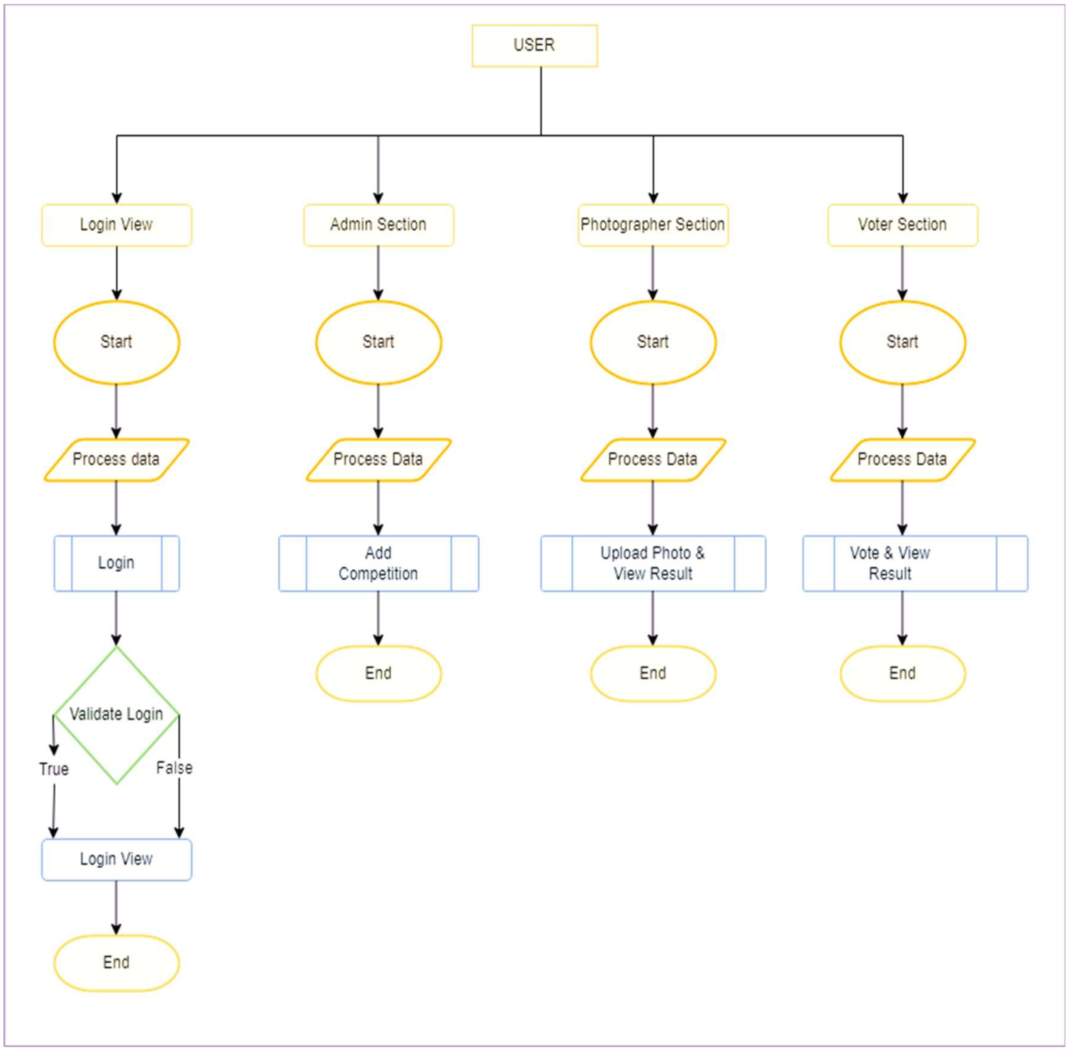
Level 2 DFD – Photographer



Level 3 DFD - Voter



3.5 System Flow Chart:



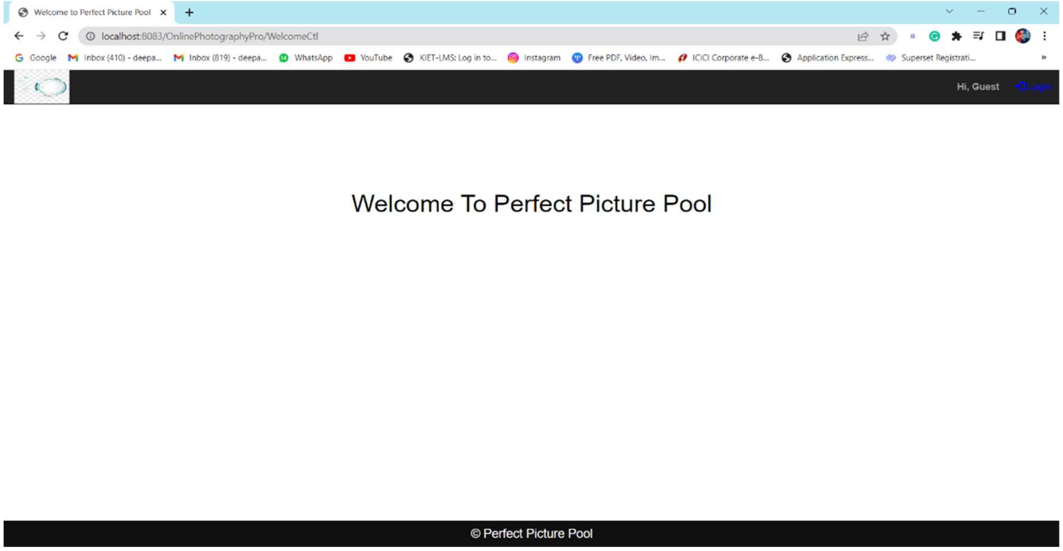
Chapter – 4

4.1 Module’s

There are three main users of this application one is the Admin and another one is the Photographer and last one is Voter, let us see the responsibility.

1) Welcome

- Welcome Page display.

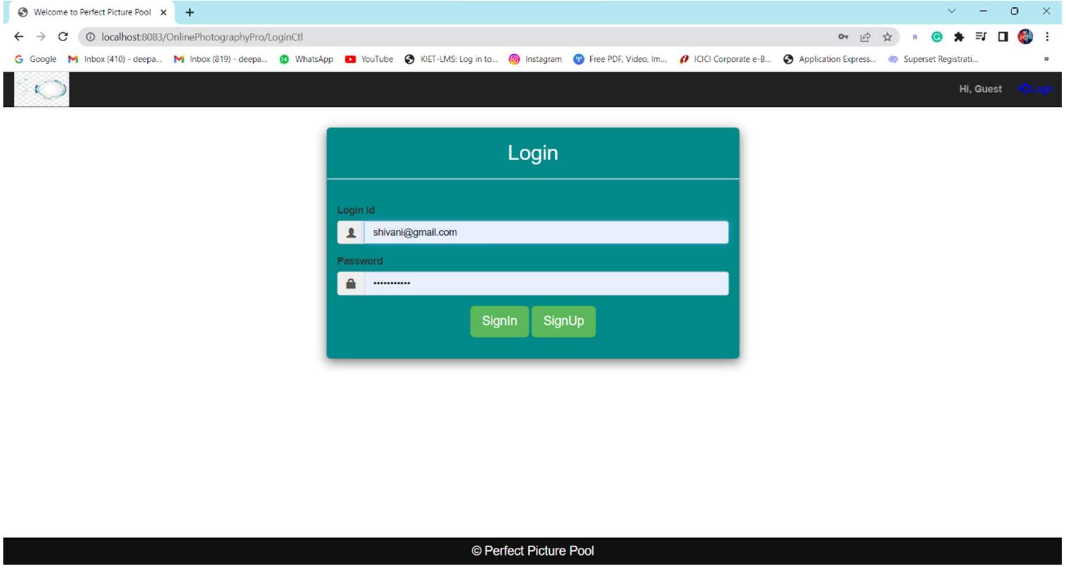


Welcome To Perfect Picture Pool

© Perfect Picture Pool

2) Admin

- Admin can login/logout of the system.



Login

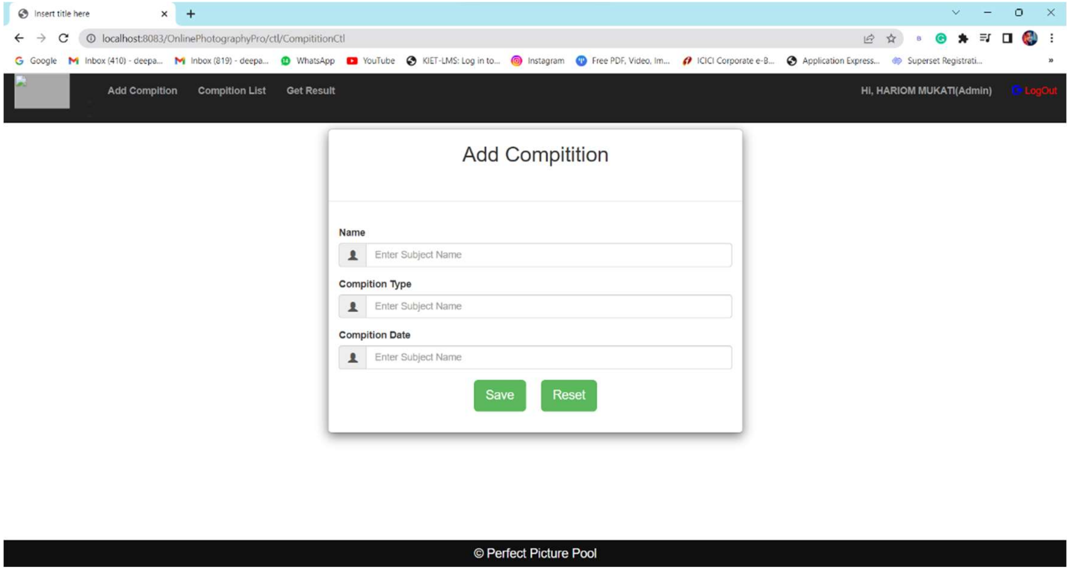
Login Id

shivani@gmail.com

Password

•••••••••••

- Can View/Add/Delete/Update Competition details.



Add Compitition

Name

Enter Subject Name

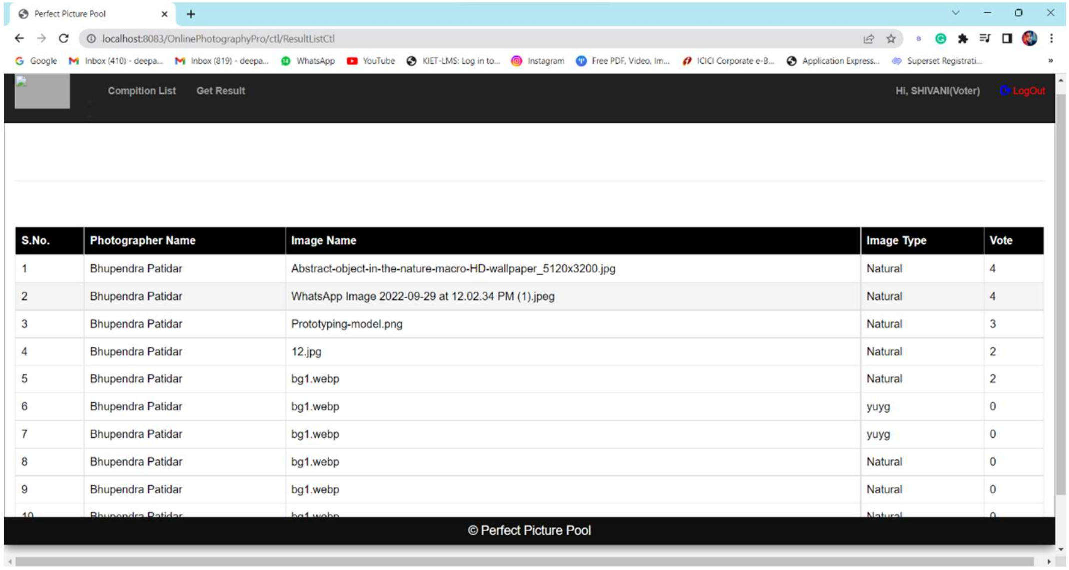
Compition Type

Enter Subject Name

Compition Date

Enter Subject Name

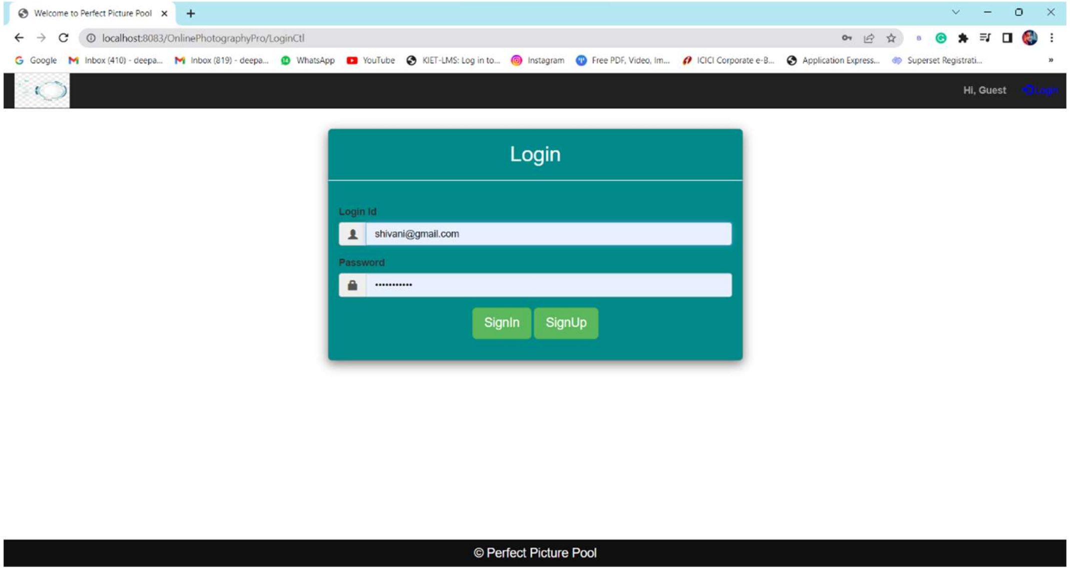
- Can View Results.



S.No.	Photographer Name	Image Name	Image Type	Vote
1	Bhupendra Patidar	Abstract-object-in-the-nature-macro-HD-wallpaper_5120x3200.jpg	Natural	4
2	Bhupendra Patidar	WhatsApp Image 2022-09-29 at 12.02.34 PM (1).jpeg	Natural	4
3	Bhupendra Patidar	Prototyping-model.png	Natural	3
4	Bhupendra Patidar	12.jpg	Natural	2
5	Bhupendra Patidar	bg1.webp	Natural	2
6	Bhupendra Patidar	bg1.webp	yuyg	0
7	Bhupendra Patidar	bg1.webp	yuyg	0
8	Bhupendra Patidar	bg1.webp	Natural	0
9	Bhupendra Patidar	bg1.webp	Natural	0
10	Bhupendra Patidar	bg1.webp	Natural	0

3) Photographer

- Can log in/log out of the system.



Login

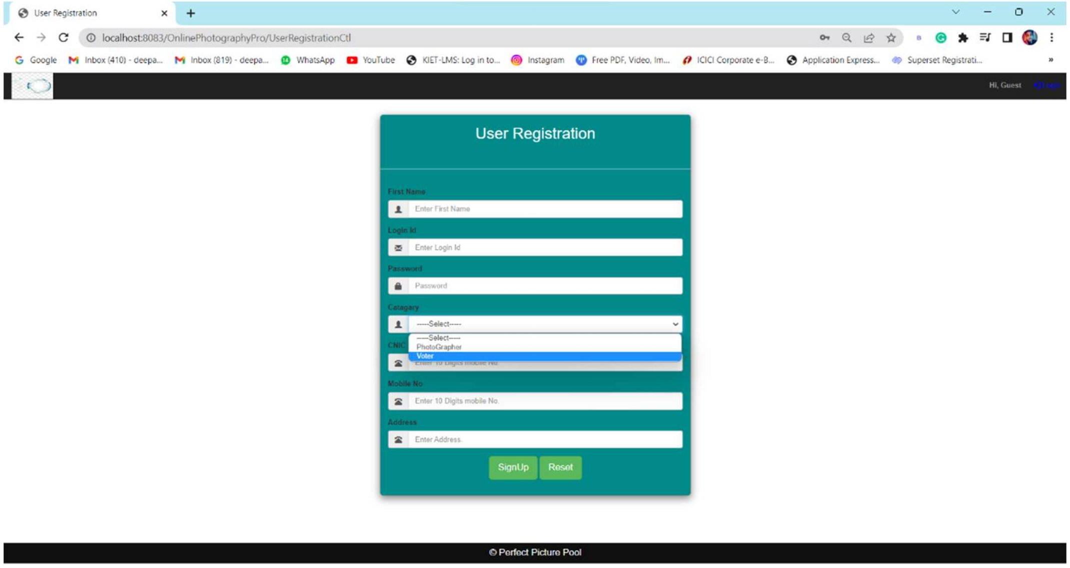
Login Id

shivani@gmail.com

Password

•••••••••••

- Can Sign into the system.



User Registration

First Name

Enter First Name

Login Id

Enter Login Id

Password

Password

Catagary

-----Select-----

-----Select-----
PhotoGrapher
Voter

CNIC

Enter 10 Digits mobile No

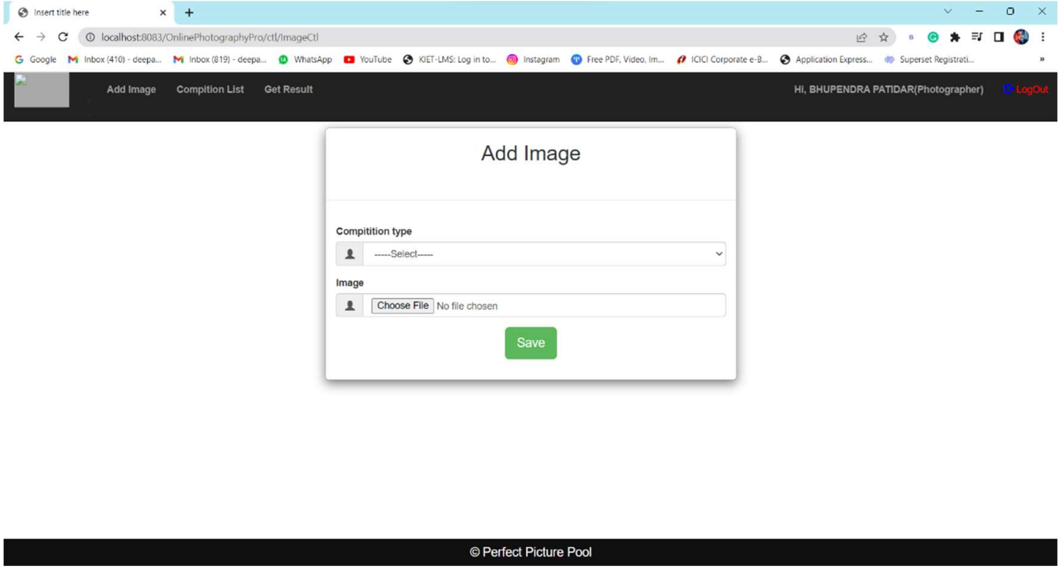
Mobile No

Enter 10 Digits mobile No.

Address

Enter Address.

- Can Add Image for the Competition.


Insert title here
localhost:8083/OnlinePhotographyPro/ctl/ImageCtl
Add Image
Compition List
Get Result
Hi, BHUPENDRA PATIDAR(Photographer)
LogOut

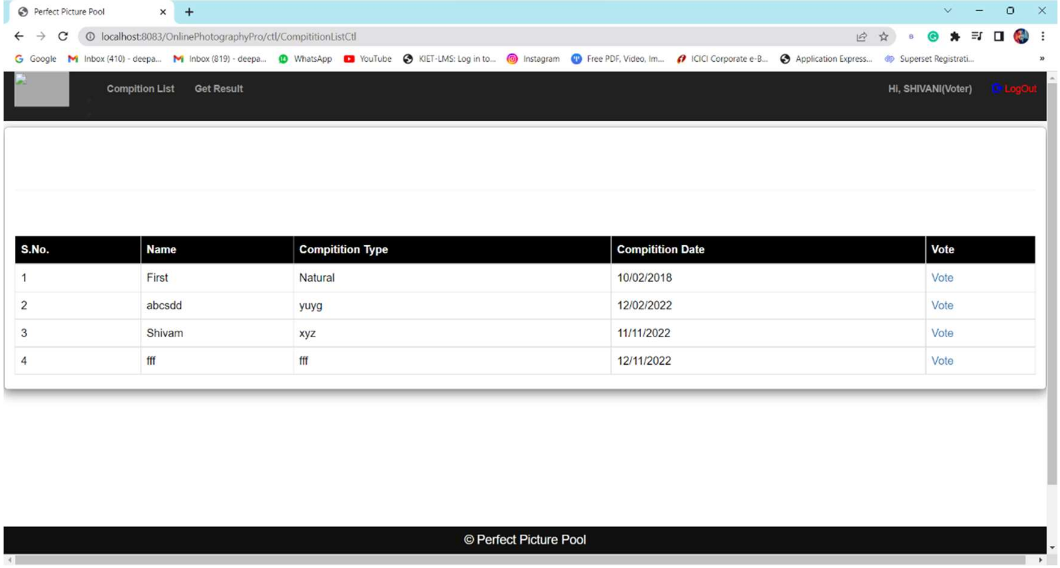
Add Image

Compitition type: -----Select-----

Image: Choose File No file chosen

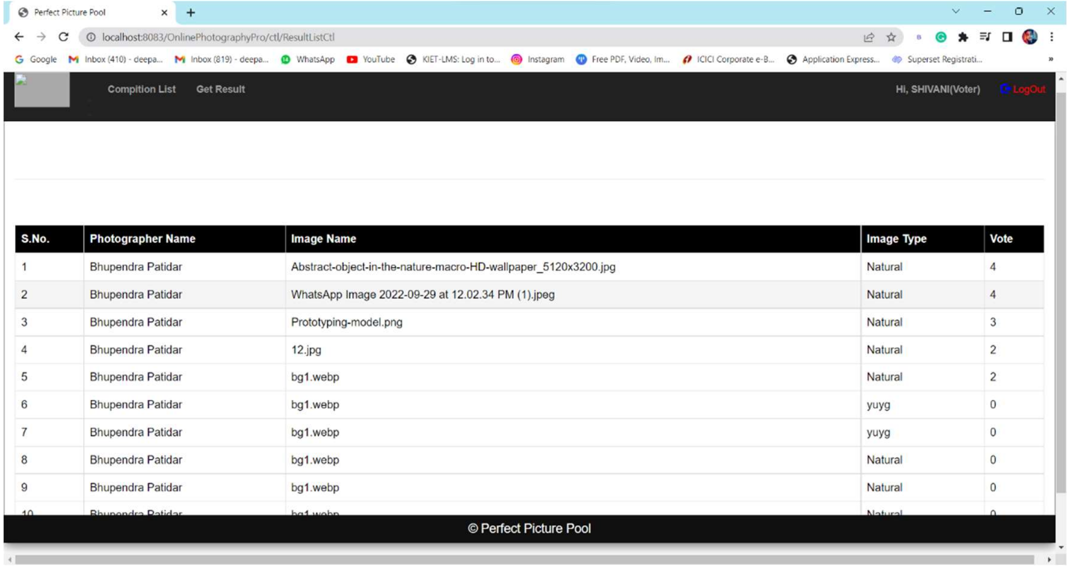
Save
© Perfect Picture Pool

- Can View the competition list.



S.No.	Name	Compitition Type	Compitition Date	Vote
1	First	Natural	10/02/2018	Vote
2	abcsdd	yuyg	12/02/2022	Vote
3	Shivam	xyz	11/11/2022	Vote
4	fff	fff	12/11/2022	Vote

- Photographer can view the result.

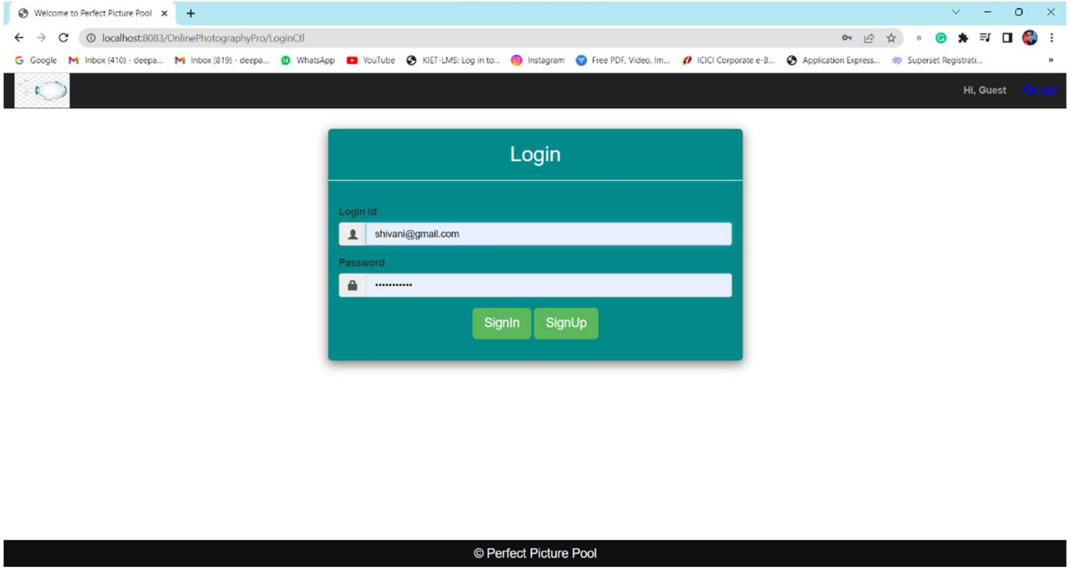


S.No.	Photographer Name	Image Name	Image Type	Vote
1	Bhupendra Patidar	Abstract-object-in-the-nature-macro-HD-wallpaper_5120x3200.jpg	Natural	4
2	Bhupendra Patidar	WhatsApp Image 2022-09-29 at 12.02.34 PM (1).jpeg	Natural	4
3	Bhupendra Patidar	Prototyping-model.png	Natural	3
4	Bhupendra Patidar	12.jpg	Natural	2
5	Bhupendra Patidar	bg1.webp	Natural	2
6	Bhupendra Patidar	bg1.webp	yuyg	0
7	Bhupendra Patidar	bg1.webp	yuyg	0
8	Bhupendra Patidar	bg1.webp	Natural	0
9	Bhupendra Patidar	bg1.webp	Natural	0
10	Bhupendra Patidar	bg1.webp	Natural	0

© Perfect Picture Pool

4) Voter

- Can log in/log out of the system.



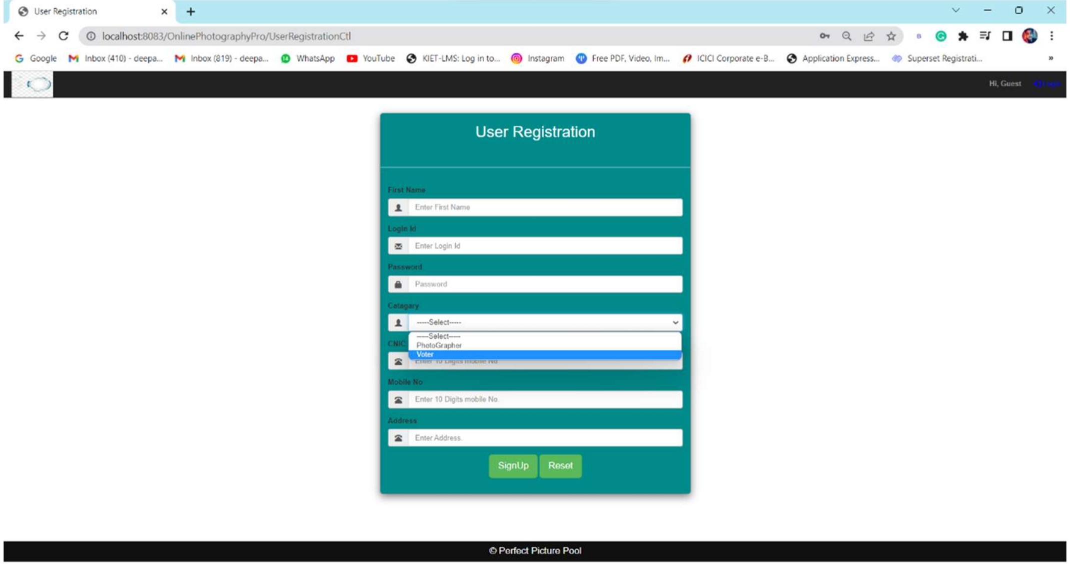
Login

Login Id
shivani@gmail.com

Password
•••••••••••

© Perfect Picture Pool

- Can Sign into the system.



localhost:8083/OnlinePhotographyPro/UserRegistrationCtl

User Registration

First Name: Enter First Name

Login Id: Enter Login Id

Password: Password

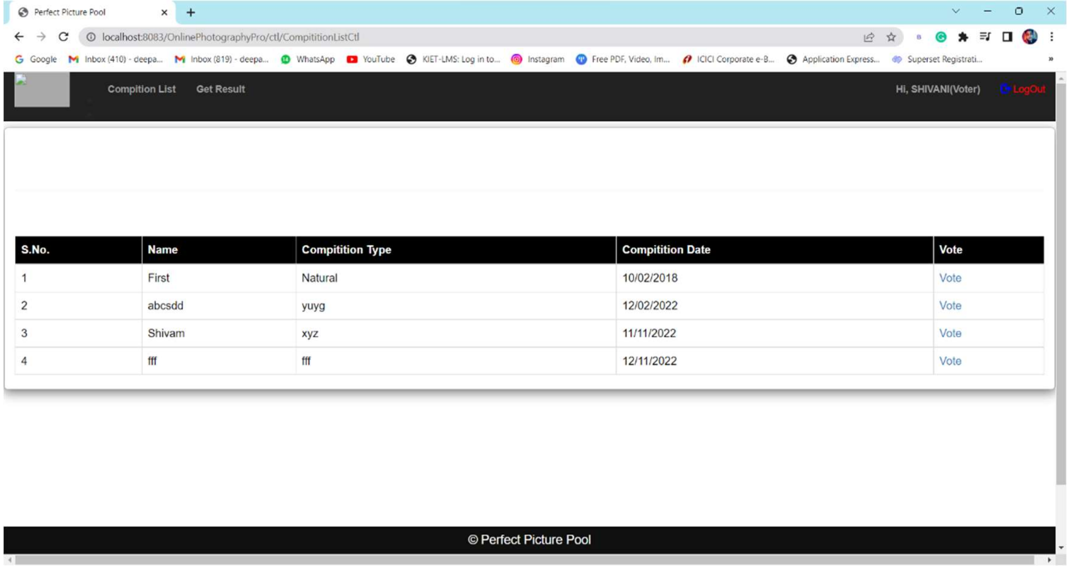
Catagary: -----Select-----
-----Select-----
PhotoGrapher
Voter

CNIC: Enter 10 Digits mobile No.

Mobile No: Enter 10 Digits mobile No.

Address: Enter Address.

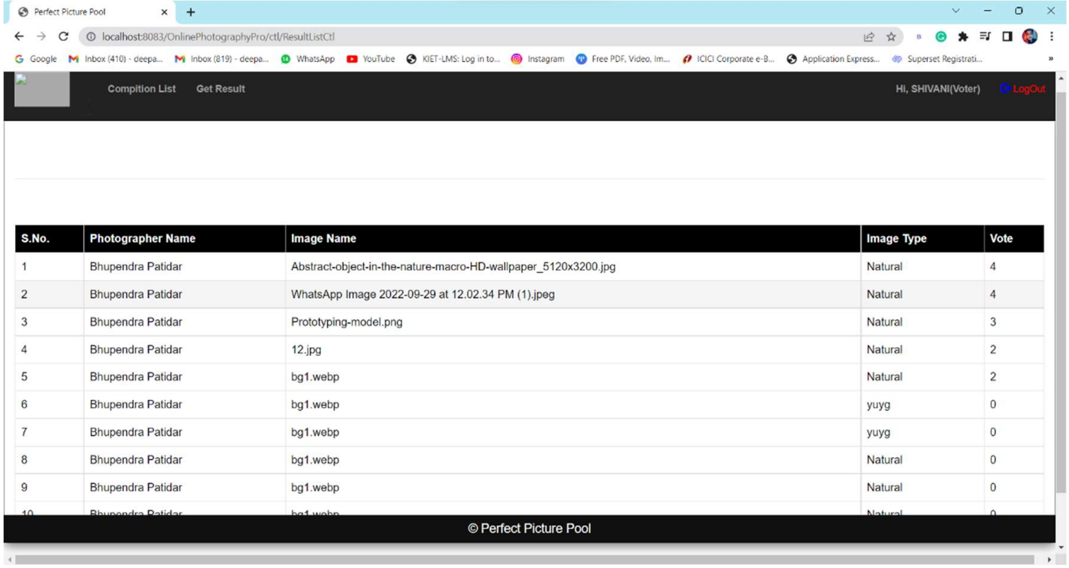
- The Voter can view the competition list.



localhost:8083/OnlinePhotographyPro/ctl/CompititionListCtl

S.No.	Name	Compitition Type	Compitition Date	Vote
1	First	Natural	10/02/2018	Vote
2	abcsdd	yuyg	12/02/2022	Vote
3	Shivam	xyz	11/11/2022	Vote
4	fff	fff	12/11/2022	Vote

- The Voter can view the result of the competition.



S.No.	Photographer Name	Image Name	Image Type	Vote
1	Bhupendra Patidar	Abstract-object-in-the-nature-macro-HD-wallpaper_5120x3200.jpg	Natural	4
2	Bhupendra Patidar	WhatsApp Image 2022-09-29 at 12.02.34 PM (1).jpeg	Natural	4
3	Bhupendra Patidar	Prototyping-model.png	Natural	3
4	Bhupendra Patidar	12.jpg	Natural	2
5	Bhupendra Patidar	bg1.webp	Natural	2
6	Bhupendra Patidar	bg1.webp	yuyg	0
7	Bhupendra Patidar	bg1.webp	yuyg	0
8	Bhupendra Patidar	bg1.webp	Natural	0
9	Bhupendra Patidar	bg1.webp	Natural	0
10	Bhupendra Patidar	bg1.webp	Natural	0

© Perfect Picture Pool

Note: The login, profile section and registration are also present in the system.

Chapter – 5

5.1 Module Wise Code

- **Welcome View**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Welcome to Perfect Picture Pool</title>

<style>

.footer {
	position: absolute;
	bottom: 0;
	width: 100%;
}

</style>

</head>
<body style="background-
image:url('<%=ORSView.APP_CONTEXT%>/img/pen.jpeg'); background-size:
cover;"  class="img-responsive" >
	<%@ include file="Header.jsp"%>
<br>
<br>
<h1 align="center" style="color: black; padding-top: 120px;">Welcome
		To Perfect Picture Pool</h1>
<div>  <%@ include file="Footer.jsp"%> </div>
</body>
</html>
```

- **Header view**

```
<%@page import="com.online.photography.controller.ORSView"%>
<%@page import="com.online.photography.dto.UserDTO"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Perfect Picture Pool</title>
<link href="img/favicon.png" rel="icon">

<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>



<link rel="stylesheet" href="css/rohan.css">
<link href="img/favicon.png" rel="icon">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/Bootstrap.css">
<link rel="stylesheet" href="css/rohan.css">


<style type="text/css">


.body {


    min-height: calc(100%)  ;
        padding-top: 80px;
        position: relative;
        top: 0;
        margin-top: 0;
}

.orsform_modal-dialog {
        margin-top:0;
        margin-bottom: 0;
}

.footer {
        position: absolute;
        bottom: 0;
        width: 100%;
}

</style>
</head>
<%
UserDTO userDto = (UserDTO) session.getAttribute("user");

                boolean userLoggedIn = userDto != null;

                String welcomeMsg = "Hi, ";

                if (userLoggedIn) {
        welcomeMsg +=
userDto.getName().toUpperCase()+"("+userDto.getRoleName()+")" ;
                } else {
        welcomeMsg += "Guest";
                }
%>


        <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container-fluid">
                <div class="navbar-header">


                        <button type="button" class="navbar-toggle" data-
toggle="collapse"
                                data-target="#myNavbar">
                                <span class="icon-bar"></span> <span class="icon-
bar"></span> <span
                                        class="icon-bar"></span>
                        </button>

                        <a href="WelcomeCtl"><img
                                style="background-color: #A9A9A9"
                                src="img/logo1.jpg" width="80" height="50">
                        </a>

                </div>

                <div>
                        <div class="collapse navbar-collapse" id="myNavbar">
                <ul class="nav navbar-nav">
                    <li class="#"><a class="#" href="#"></a>
                        <ul class="#">


                            <li></li>
                            <li></li>
                        </ul></li>
                </ul>
<%if(userLoggedIn){

%>
    <%if(userDto.getRoleId()==1) {%>

                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.COMPITITION_CTL%>"><b>Add Compition</b></a>
                    </li>
                </ul>
                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.COMPITITION_LIST_CTL%>"><b>Compition List</b></a>
                    </li>
                </ul>
                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.RESULT_LIST_CTL%>"><b>Get Result</b></a>
                    </li>
                </ul>




        <%}else if(userDto.getRoleId()==2){ %>

        <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.IMAGE_CTL%>"><b>Add Image</b></a>
                    </li>
                </ul>
                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.COMPITITION_LIST_CTL%>"><b>Compition List</b></a>
                    </li>
                </ul>
                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.RESULT_LIST_CTL%>"><b>Get Result</b></a>
                    </li>
                </ul>
        <%}else if(userDto.getRoleId()==3){ %>


                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.COMPITITION_LIST_CTL%>"><b>Compition List</b></a>
                    </li>
                </ul>
                <ul class="nav navbar-nav">
                    <li ><a
href="<%=ORSView.RESULT_LIST_CTL%>"><b>Get Result</b></a>
                    </li>
                </ul>
                <%}}%>
                <ul class="nav navbar-nav navbar-right">
                    <li class="dropdown"><a class="dropdown-
toggle"
                        data-toggle="dropdown"
href="#"><b><%=welcomeMsg%></b></a>
                        <%if(userLoggedIn){ %>
                        <li><a style="color: red"
href="<%=ORSView.LOGIN_CTL%>?operation=Logout">
                                    <span style="color:
blue" class="glyphicon glyphicon-log-out"></span>
                                    LogOut
                            </a></li>
                            <%}else{ %>
                            <li><a style="color:blue;"
href="<%=ORSView.LOGIN_CTL%>">
                                    <span style="color:
blue" class="glyphicon glyphicon-log-in"></span>
                                    Login
                            </a></li>
                            <%} %>
                        </ul>
            </div>

        </div>

    </div>
    </nav>
</html>
```

- **Footer view**

```
<div  class="footer" >
        <div class="container-fluid " style="background-color: #101010;">
                        <h4  style="color:white;text-align:center; " >&copy; Perfect Picture Pool</h4>
        </div>
</div>
```

- **Login View**

```
<%@page import="com.online.photography.controller.LoginCtl"%>
<%@page import="com.online.photography.util.DataUtility"%>
<%@page import="com.online.photography.util.ServletUtility"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!-- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"> -->
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Welcome to Perfect Picture Pool</title>

<style type="text/css">
.form-control {
        width: 174px;
        text-align-last: left;
}
</style>

</head>

<body
        style="background-image:url('<%=ORSView.APP_CONTEXT%>/img/nightsky.jpg'); background-size: cover;"
        class="img-responsive">

        <%@ include file="Header.jsp"%>

        <jsp:useBean id="dto" class="com.online.photography.dto.UserDTO"
                scope="request"></jsp:useBean>

        <%
        String uri = (String) request.getAttribute("uri");
        %>


        <div class="body">
                <div class="modal-dialog  orsform_modal-dialog">
                        <div class="modal-content">


                                <div class="modal-heading">
```

```
                    <h2 class="text-center">Login</h2>

                    <center>
                        <b><font class="text-center" color="red">
<%=ServletUtility.getErrorMessage(request)%></font></b>

                        <b><font color="black">
<%=ServletUtility.getSuccessMessage(request)%></font></b>
                    </center>
                </div>

                <hr>


                <div class="modal-body">
                    <form action="<%=ORSView.LOGIN_CTL%>"
role="form" method="post">

                        <input type="hidden" name="uri"
value="<%=uri%>"> <input
                            type="hidden" name="id"
value="<%=dto.getId()%>">

                        <div class="form-group">
                            <label for="login"> Login
Id</label>
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span> <input type="text"
class="form-control" placeholder="User Name"
                                    name="login"

    value="<%=DataUtility.getStringData(dto.getLogin())%>" />
                            </div>
                            <b><font
color="red"><%=ServletUtility.getErrorMessage("login", request)%></font></b>
                        </div>


                        <div class="form-group">
                            <label for="password">
Password</label>
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
```

```
                                                            class="glyphicon
glyphicon-lock"></span>
                                                     </span> <input
type="password" class="form-control"

      placeholder="Password" name="password"

      value="<%=DataUtility.getStringData(dto.getPassword())%>" />

                                              </div>
                                              <b><font color="red">
<%=ServletUtility.getErrorMessage("password", request)%></font></b>

                                        </div>

                                        <div class="form-group text-center">
                                              <input type="submit" class="btn
btn-success btn-lg"
                                                     name="operation"
value="<%=LoginCtl.OP_SIGN_IN%>"> <input
                                                     type="submit" class="btn
btn-success btn-lg" name="operation"

      value="<%=LoginCtl.OP_SIGN_UP%>">
                                        </div>

                                  </form>
                           </div>


                     </div>



               </div>



               <div style="width: 100px; height: 100px;"></div>

               <%@ include file="Footer.jsp"%>


      </div>

</body>
</html>
```

- **User Registration View**

```
<%@page import="com.online.photography.util.HTMLUtility"%>
<%@page import="java.util.HashMap"%>
<%@page import="com.online.photography.util.DataUtility"%>
<%@page import="com.online.photography.util.ServletUtility"%>
<%@page import="com.online.photography.controller.UserRegistrationCtl"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>User Registration</title>

<style type="text/css">
.form-control {
        width: 174px;
        text-align-last: left;
}

</style>



</head>


<body style="background-
image:url('<%=ORSView.APP_CONTEXT%>/img/nightsky.jpg');  background-size:
cover;"
        class="img-responsive"  ng-app="720" ng-controller="TestController as ctrl" >

        <%@ include file="Header.jsp"%>

        <jsp:useBean id="dto" class="com.online.photography.dto.UserDTO"
                scope="request"></jsp:useBean>

        <div class="body">
                <div class="modal-dialog orsform_modal-dialog">
                        <div class="modal-content">

                                <div class="modal-heading">
                                        <h2 class="text-center">User Registration</h2>
```

```
                    <br>
                    <center>
                        <b><font class="text-center" color="red">
<%=ServletUtility.getErrorMessage(request)%></font></b>

                        <b><font color="black">
<%=ServletUtility.getSuccessMessage(request)%></font></b>
                    </center>
                </div>

                <hr>

                <div class="modal-body">
                    <form
action="<%=ORSView.USER_REGISTRATION_CTL%>" method="post">

                        <input type="hidden" name="id"
value="<%=dto.getId()%>"> <label
                            for="name">First Name</label>

                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span> <input type="text"
class="form-control" name="name"
                                    placeholder="Enter
First Name"

    value="<%=DataUtility.getStringData(dto.getName())%>" />
                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("name", request)%></font></b>
                        </div>



                        <label for="login"> Login Id</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-envelope    "></span>
                                </span> <input type="text"
class="form-control" name="login"
                                    placeholder="Enter
Login Id"
```

```
    value="<%=DataUtility.getStringData(dto.getLogin())%>" />
                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("login", request)%></font></b>
                        </div>

                        <label for="password"> Password</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-lock"></span>
                                </span> <input
type="password" class="form-control"

    placeholder="Password" name="password"

    value="<%=DataUtility.getStringData(dto.getPassword())%>" />

                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("password", request)%></font></b>

                        </div>


                        <label for="roleId"> Catagary</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span>
                                <%
                                    HashMap map =
new HashMap();
                                    map.put("2",
"PhotoGrapher");
                                    map.put("3",
"Voter");

                                    String htmlList =
HTMLUtility.getList("roleId", dto.getRoleName(), map);
                                %>
                                <%=htmlList%>

                            </div>
                            <b><font
color="red"><%=ServletUtility.getErrorMessage("roleId", request)%></font></b>
                        </div>

                    <label for="cnic"> CNIC No</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-phone-alt"></span>
                                </span> <input type="text"
class="form-control" name="cnic"

    value="<%=DataUtility.getStringData(dto.getCNIC())%>"
                                    placeholder="Enter
10 Digits mobile No." />

                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("mobile", request)%></font></b>

                        </div>




                        <label for="mobile"> Mobile No</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-phone-alt"></span>
                                </span> <input type="text"
class="form-control" name="mobile"

    value="<%=DataUtility.getStringData(dto.getCellNo())%>"
                                    placeholder="Enter
10 Digits mobile No." />

                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("mobile", request)%></font></b>

                        </div>

                        <label for="address"> Address</label>
                        <div class="form-group">
                            <div class="input-group">
```

```
                                                            <span class="input-group-
addon"> <span
                                                                      class="glyphicon
glyphicon-phone-alt"></span>
                                                            </span> <input type="text"
class="form-control" name="address"

          value="<%=DataUtility.getStringData(dto.getAddress())%>"
                                                                      placeholder="Enter
Address." />

                                                   </div>
                                                   <b><font color="red">
<%=ServletUtility.getErrorMessage("address", request)%></font></b>

                                         </div>



                                         <div class="form-group text-center">
                                                   <input type="submit" class="btn
btn-success btn-lg"
                                                            name="operation"
value="<%=UserRegistrationCtl.OP_SIGN_UP%>">
                                                   <input type="submit" class="btn
btn-success btn-lg"
                                                            name="operation"
value="<%=UserRegistrationCtl.OP_RESET%>">
                                         </div>

                              </form>
                    </div>

          </div>
     </div>





     <div style="width: 100px; height: 130px;"></div>
     <%@ include file="Footer.jsp"%>

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

- **MyProfileView**

```
<%-- <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!-- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"> -->
<%@page import="com.raystec.proj3.controller.MyProfileCtl"%>
<%@page import="com.raystec.proj3.util.HTMLUtility"%>
<%@page import="java.util.HashMap"%>
<%@page import="com.raystec.proj3.util.DataUtility"%>
<%@page import="com.raystec.proj3.util.ServletUtility"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>

</head>

<body
    style="background-
image:url('<%=ORSView.APP_CONTEXT%>/img/nightsky.jpg'); background-size:
cover;"
    class="img-responsive" ng-app="720"
    ng-controller="TestController as ctrl">



    <%@ include file="Header.jsp"%>

    <jsp:useBean id="dto" class="com.raystec.proj3.dto.UserDTO"
        scope="request"></jsp:useBean>


    <div class="body">
        <div class="modal-dialog orsform_modal-dialog">

            <div class="modal-content">

                <div class="modal-heading">

                    <h2 class="text-center">MY Profile</h2>
                    <br>
                    <center>
                        <div>
                            <b><font class="text-center "
color="white"> <%=ServletUtility.getErrorMessage(request)%></font></b>
```

```
                            <b><font color="black">
<%=ServletUtility.getSuccessMessage(request)%></font></b>
                        </div>
                    </center>
                </div>
                <hr>

                <div class="modal-body">
                    <form
action="<%=ORSView.MY_PROFILE_CTL%>" role="form"
                        method="post">

                        <input type="hidden" name="id"
value="<%=dto.getId()%>">


                        <div class="form-group">

                            <label for="login"> Login
Id</label>
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span> <input type="text"
class="form-control" name="login"

    value="<%=DataUtility.getStringData(dto.getLogin())%>"

    readonly="readonly" />
                            </div>

                            <b><font color="red">
<%=ServletUtility.getErrorMessage("login", request)%></font></b>

                        </div>


                        <label for="firstName"> First
Name</label>
                        <div class="form-group">
                            <div class="input-group">

                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span> <input type="text"
class="form-control" name="firstName"
```

```
                                    placeholder="Enter
First Name"

    value="<%=DataUtility.getStringData(dto.getFirstName())%>" />


                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("firstName", request)%></font></b>
                        </div>

                        <label for="lastName"> Last
Name</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-user"></span>
                                </span> <input type="text"
class="form-control" name="lastName"
                                    placeholder="Enter
Last Name"

    value="<%=DataUtility.getStringData(dto.getLastName())%>" />
                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("lastName", request)%></font></b>
                        </div>

                        <label for="gender"> Gender</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span

                                    class="fa fa-

intersex"></span>

                                </span>
                                <%
                            HashMap map = new HashMap();
                                    map.put("Male",
"Male");

                                    map.put("Female",

"Female");


                                    String htmlList =
HTMLUtility.getList("gender", dto.getGender(), map);
                                %>
                                <%=htmlList%>
                            </div>
                            <b><font
color="red"><%=ServletUtility.getErrorMessage("gender",
request)%></font></font></b>
                        </div>

                        <label for="mobile"> Mobile No</label>
                        <div class="form-group">
                            <div class="input-group">
                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-phone-alt"></span>
                                </span> <input type="text"
class="form-control" name="mobile"

    value="<%=DataUtility.getStringData(dto.getMobileNo())%>"
                                    placeholder="Enter
10 Digits mobile No." />

                            </div>
                            <b><font color="red">
<%=ServletUtility.getErrorMessage("mobile", request)%></font></b>

                        </div>

                        <label for="dob">Date Of Birth</label>

                        <div class="form-group">
                            <div class="input-group">

                                <span class="input-group-
addon"> <span
                                    class="glyphicon
glyphicon-calendar"></span>
                                </span>

                                <div data-datepicker date-
format="MM/dd/yyyy" date-typer="true" date-max-limit="{{maxDate}}" >


                                    <input type="text"
class="form-control" name="dob"

    placeholder="Select Date Of Birth" readonly="readonly"

    value="<%=DataUtility.getDateString(dto.getDob())%>" />
                                </div>
```

```
                                                        </div>
                                                        <b><font
color="red"><%=ServletUtility.getErrorMessage("dob", request)%></font></b>
                                                </div>


                                                <div class="form-group text-center">
                                                        <input type="submit" class="btn
btn-success btn-lg"
                                                                name="operation"
value="<%=MyProfileCtl.OP_SAVE%>">

                                                </div>

                                        </form>
                                </div>
                        </div>
                </div>


                <div style="width: 100px; height: 250px;"></div>

                <%@ include file="Footer.jsp"%>
        </div>
</body>
</html> --%>
```

- **Competition List View**

```
<%@page import="com.online.photography.controller.CompititionListCtl"%>
<%@page import="com.online.photography.dto.CompititionDTO"%>
<%@page import="com.online.photography.util.ServletUtility"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Iterator"%>

<html>
<head>

<script type="text/javascript"
    src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js">

</script>

<script type="text/javascript">
    $(document).ready(function() {
        // add multiple select / deselect functionality
        $("#selectall").click(function() {
            $('.case').attr('checked', this.checked);
        });
        // if all checkbox are selected, check the selectall checkbox  also

        $(".case").click(function() {
            if ($(".case").length == $(".case:checked").length) {
                $("#selectall").attr("checked", "checked");
            } else {
                $("#selectall").removeAttr("checked");
            }
        });
    });
</script>
</head>

<body
    style="background-image: url('/img/nightsky.jpg'); background-size: cover;"
    class="img-responsive">

    <%@ include file="Header.jsp"%>
    <div class="body">
```

```
        <div class="container-fluid modal-content">

            <div class="row">

                <div class="col-sm-4"></div>

                <div class="col-sm-4" align="center" style="color:
white">
                    <h1>Compitition List</h1>
                </div>

                <div class="col-sm-4"></div>

            </div>

            <hr>
            <br>




            <div class="form-group row">


                <div class="row">

                    <div class="col-sm-4"></div>

                    <div class="col-sm-4" align="center">

                        <div align="center">

                            <b><font color="red"><h4>

    <%=ServletUtility.getErrorMessage(request)%></h4></font></b> <b><font
                                color="black">

    <h4><%=ServletUtility.getSuccessMessage(request)%></h4>
                            </font></b>


                        </div>

                    </div>

                    <div class="col-sm-4"></div>
                </div>
            </div>

            <div class="row">


                <div class="col-xs-12" align="center" style="overflow-x:
auto">

                    <table class="table  table-bordered table-hover">

                        <thead style="background-color: black;
color: white;">

                            <tr>
                                <%
                        UserDTO udto = (UserDTO)
session.getAttribute("user");
                                List list =
ServletUtility.getList(request);
                                int pageNo =
ServletUtility.getPageNo(request);
                                int pageSize =
ServletUtility.getPageSize(request);
                                int index = ((pageNo - 1) *
pageSize) + 1;
                                %>

                                <th>S.No.</th>
                                <th>Name</th>
                                <th>Compitition Type</th>
                                <th>Compitition Date</th>
                                <%
                                if (udto.getRoleId() == 3) {
                                %>
                                <th>Vote</th>
                                <%
                                }
                                %>



                            </tr>
                        </thead>
                        <tbody style="color: black;">
                            <%
                            Iterator it = list.iterator();

                            while (it.hasNext()) {
```

```
                                                            CompititionDTO dto =
(CompititionDTO) it.next();
                                                    %>
                                                    <tr>

                                                            <td><%=index++%></td>

        <td><%=dto.getcName()%></td>

        <td><%=dto.getcType()%></td>

        <td><%=dto.getcDate()%></td>
                                                            <%
                                                            if (udto.getRoleId() == 3) {
                                                            %>
                                                            <td><a

        href="<%=ORSView.IMAGE_LIST_CTL%>?type=<%=dto.getcType()%>">
Vote</a></td>
                                                            <%
                                                            }
                                                            %>


                                                    </tr>
                                                    <%
                                                    }
                                                    %>
                                            </tbody>
                                    </table>
                            </div>
                    </div>

            </div>

        </div>

        <input type="hidden" name="pageNo" value="<%=pageNo%>">
        <input type="hidden" name="pageSize" value="<%=pageSize%>">
        </form>

        <div style="width: 100px; height: 100px;"></div>
        <%@include file="Footer.jsp"%>

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

- **Competition view**

```
<%@page import="com.online.photography.controller.CompititionCtl"%>
<%@page import="com.online.photography.util.ServletUtility"%>
<%@page import="com.online.photography.util.DataUtility"%>


<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!-- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"> -->

<%@page import="java.util.List"%>

<%@page import="java.util.HashMap"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>


<style type="text/css">
.form-control {
        width: 174px;
        text-align-last: left;
}
</style>
</head>


<body  style="background-
image:url('<%=ORSView.APP_CONTEXT%>/img/nightsky.jpg'); background-size:
cover;"  class="img-responsive">

<div style="min-height: 588px;">
        <%@ include file="Header.jsp"%>

        <jsp:useBean id="dto" class="com.online.photography.dto.CompititionDTO"
                scope="request"></jsp:useBean>

                <%
                List l = (List) request.getAttribute("courseList");
        %>
```

```
<div class="body">

    <div class="modal-dialog orsform_modal-dialog" >

        <div class="modal-content">
            <div class="modal-heading">



                <h2 class="text-center">Add Compitition</h2>



                <br>
                <center>
                    <b><font class="text-center" color="red">
<%=ServletUtility.getErrorMessage(request)%></font></b>

                    <b><font color="black">
<%=ServletUtility.getSuccessMessage(request)%></font></b>
                </center>
            </div>
            <hr />

            <div class="modal-body">

                <form action="<%=ORSView.COMPITITION_CTL%>"
role="form" method="post">
                    <input type="hidden" name="id"
value="<%=dto.getId()%>">










                    <label  for="name"> Name</label>


                    <div class="form-group">

                        <div class="input-group">
```

```
                                                        <span class="input-group-addon">
<span
                                                                class="glyphicon
glyphicon-user"></span>
                                                        </span> <input type="text"
class="form-control" name="name"
                                                                placeholder="Enter Subject
Name "

        value="<%=DataUtility.getStringData(dto.getcName())%>" />
                                                </div>
                                                <b><font color="red">
<%=ServletUtility.getErrorMessage("name", request)%></font></b>
                                        </div>

                                        <label  for="type"> Compition Type</label>


                                        <div class="form-group">

                                                <div class="input-group">

                                                        <span class="input-group-addon">
<span
                                                                class="glyphicon
glyphicon-user"></span>
                                                        </span> <input type="text"
class="form-control" name="type"
                                                                placeholder="Enter Subject
Name "

        value="<%=DataUtility.getStringData(dto.getcType())%>" />
                                                </div>
                                                <b><font color="red">
<%=ServletUtility.getErrorMessage("type", request)%></font></b>
                                        </div>

                                        <label  for="date"> Compition Date</label>


                                        <div class="form-group">

                                                <div class="input-group">

                                                        <span class="input-group-addon">
<span
                                                                class="glyphicon
glyphicon-user"></span>
                                                        </span> <input type="text"
class="form-control" name="date"
```

```
                                                    placeholder="Enter Subject
Name "

        value="<%=DataUtility.getStringData(dto.getcDate())%>" />
                                            </div>
                                            <b><font color="red">
<%=ServletUtility.getErrorMessage("date", request)%></font></b>
                                    </div>




                                    <div class="form-group text-center">

                                            <input type="submit" class="btn btn-
success btn-lg"
                                                    name="operation"
value="<%=CompititionCtl.OP_SAVE%>"> &emsp;
                                            <%
                                                    if
(DataUtility.getInt(request.getParameter("id")) > 0) {
                                            %><input type="submit" class="btn btn-
success btn-lg"
                                                    name="operation"
value="<%=CompititionCtl.OP_CANCEL%>">
                                            <%
                                                    } else {
                                            %><input type="submit" class="btn btn-
success btn-lg"
                                                    name="operation"
value="<%=CompititionCtl.OP_RESET%>">
                                            <%
                                                    }
                                            %>
                                    </div>


                            </form>
                    </div>
            </div>
        </div>
         <div  style="width: 100px; height: 100px; "></div>

        <%@ include file="Footer.jsp"%>

        </div>
</body>

</html>
```

- **Result View**

```
<%@page import="com.online.photography.model.UserModel"%>
<%@page import="com.online.photography.dto.ImageDTO"%>
<%@page import="com.online.photography.controller.CompititionListCtl"%>
<%@page import="com.online.photography.dto.CompititionDTO"%>
<%@page import="com.online.photography.util.ServletUtility"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Iterator"%>

<html>
<head>

<script
type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js">
</script>

<script type="text/javascript" >
   $(document).ready(function() {
     // add multiple select / deselect functionality
       $("#selectall").click(function() {
       $('.case').attr('checked', this.checked);
         });
    // if all checkbox are selected, check the selectall checkbox  also

    $(".case").click(function() {
       if ($(".case").length == $(".case:checked").length) {
       $("#selectall").attr("checked", "checked");
        }
       else {
       $("#selectall").removeAttr("checked");
        }
        });
    });
   </script>
</head>

<body  style="background-image:url('/img/nightsky.jpg'); background-size: cover;"
class="img-responsive">

          <%@ include file="Header.jsp"%>
<div class="body">
```

```
            <div class="container-fluid modal-content">

                <div class="row">

                    <div class="col-sm-4"></div>

                    <div class="col-sm-4" align="center"
style="color: white">
                        <h1>Result List</h1>
                    </div>

                    <div class="col-sm-4"></div>

                </div>

                <hr>
                <br>




                <div class="form-group row">


                    <div class="row">

                        <div class="col-sm-4"></div>

                        <div class="col-sm-4" align="center">

                            <div align="center">

                                <b><font
color="red"><h4>

    <%=ServletUtility.getErrorMessage(request)%></h4></font></b> <b><font
                                    color="black">

    <h4><%=ServletUtility.getSuccessMessage(request)%></h4>
                                </font></b>


                            </div>
                        </div>

                        <div class="col-sm-4"></div>
                    </div>
                </div>

                <div class="row">


                    <div class="col-xs-12" align="center"
style="overflow-x: auto">

                        <table  class="table  table-bordered table-
hover">

                            <thead style="background-color:
black; color: white;">

                                <tr>
                                    <%
                                        List list =
ServletUtility.getList(request);
                                        int pageNo =
ServletUtility.getPageNo(request);
                                        int pageSize
= ServletUtility.getPageSize(request);
                                        int index =
((pageNo - 1) * pageSize) + 1;
                                    %>


                                    <th >S.No.</th>
                                    <th >Photographer
Name</th>
                                    <th >Image
Name</th>
                                    <th>Image
Type</th>
                                    <th >Vote</th>





                                </tr>
                            </thead>
                            <tbody style="color: black;">
                                <%
                                    Iterator it =
list.iterator();
```

```
                                                                 while (it.hasNext())
{

                                                                      ImageDTO
     dto =(ImageDTO) it.next();
                                                            %>
                                                            <tr>
                                                                 <%UserModel
model=new UserModel();
                                                                      UserDTO
udto=model.findByPK(dto.getuId());
                                                                 %>
                                                                 <td
><%=index++%></td>
                                                                 <td
><%=udto.getName()%></td>
                                                                 <td
><%=dto.getPhoto()%></td>
                                                                 <td
><%=dto.getName()%></td>
                                                                 <td
><%=dto.getCount()%></td>



                                                            </tr>
                                                            <%
                                                                 }
                                                            %>
                                                       </tbody>
                                                  </table>
                                             </div>
                                        </div>

                                   </div>

                                        </div>

                                   <input type="hidden" name="pageNo"
value="<%=pageNo%>"> <input
                                        type="hidden" name="pageSize"
value="<%=pageSize%>">
                    </form>

 <div  style="width: 100px; height: 100px; "></div>
     <%@include file="Footer.jsp"%>

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

- **Welcome Module**

```
package com.online.photography.controller;

import java.io.IOException;


import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;
import org.apache.log4j.net.SyslogAppender;

import com.online.photography.util.ServletUtility;



/**
 *
 *
 * Servlet implementation class WelcomeCtl
 *
 *
 *
 * * @author Navigable Set
 * @version 1.0
 * @Copyright (c) Navigable Set
 */


@WebServlet(name = "WelcomeCtl", urlPatterns = { "/WelcomeCtl" })
public class WelcomeCtl extends BaseCtl {

private static final long serialVersionUID = 1L;

/**
 * @see HttpServlet#HttpServlet()
 */
/*public WelcomeCtl() {
	super();
// TODO Auto-generated constructor stub
}*/

/**
 * Contains display logic
 */
```

```
  /**
   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
   */
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                    ServletUtility.forward(ORSView.WELCOME_VIEW, request,
response);

  }
  /**
   * Returns the VIEW page of this Controller
   *
   * @return
   */
  @Override
  protected String getView() {
          // TODO Auto-generated method stub
          return ORSView.WELCOME_VIEW;
  }

}
```

- **User Login**

```
package com.online.photography.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.log4j.Logger;

import com.online.photography.dto.BaseBean;
import com.online.photography.dto.UserDTO;
import com.online.photography.exception.ApplicationException;
import com.online.photography.model.UserModel;
import com.online.photography.util.DataUtility;
import com.online.photography.util.DataValidator;
import com.online.photography.util.PropertyReader;
import com.online.photography.util.ServletUtility;

/**
 * Servlet implementation class LoginCtl
 */
/**
 * Login functionality Controller. Performs operation for Authentication, and
 * Session Creation and Give permission to access all Functionality regarding
 * Role
 *
 *
 * @author Navigable Set
 * @version 1.0
 * @Copyright (c) Navigable Set
 *
 */
@WebServlet(name = "LoginCtl", urlPatterns = { "/LoginCtl" })
public class LoginCtl extends BaseCtl {

    private static final long serialVersionUID = 1L;

    public static final String OP_REGISTER = "Register";
    public static final String OP_SIGN_IN = "SignIn";
    public static final String OP_SIGN_UP = "SignUp";
    public static final String OP_LOG_OUT = "Logout";
    public static String HIT_URI = null;
    private static Logger log = Logger.getLogger(LoginCtl.class);

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginCtl() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * Validate input Data Entered By User
     *
     * @param request
     * @return
     */
    @Override
    protected boolean validate(HttpServletRequest request) {

        log.debug("LoginCtl Method validate Started");

        boolean pass = true;

        String op = request.getParameter("operation");

        if (OP_SIGN_UP.equals(op) || OP_LOG_OUT.equals(op)) {
            return pass;
        }

        String login = request.getParameter("login");

        if (DataValidator.isNull(login)) {
            request.setAttribute("login",
PropertyReader.getValue("error.require", "Login Id"));
            pass = false;

        } else if (!DataValidator.isEmail(login)) {
            request.setAttribute("login",
PropertyReader.getValue("error.email", "Login Id "));
            pass = false;
        }
        if (DataValidator.isNull(request.getParameter("password"))) {
            request.setAttribute("password",
PropertyReader.getValue("error.require", "Password"));
            pass = false;
        }
        log.debug("LoginCtl Method validate Ended");
        return pass;
    }
    /**
     * Populates bean object from request parameters
     *
     * @param request
     * @return
     */
    @Override
    protected BaseBean populateBean(HttpServletRequest request) {

        log.debug("LoginCtl Method populateBean Started");

        UserDTO bean = new UserDTO();

        bean.setId(DataUtility.getLong(request.getParameter("id")));

        bean.setLogin(DataUtility.getString(request.getParameter("login")));


    bean.setPassword(DataUtility.getString(request.getParameter("password")));

        log.debug("LOginCtl Method PopulatedBean End");

        return bean;
    }

    /**
     * Display Login form
     *
     */
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        log.debug("Method doGet Started");

        HttpSession session = request.getSession(true);
        String op = DataUtility.getString(request.getParameter("operation"));

        UserModel model = new UserModel();


        long id = DataUtility.getLong(request.getParameter("id"));

        if (id > 0) {
            UserDTO UserDTO;
            try {
                UserDTO = model.findByPK(id);
                ServletUtility.setBean(UserDTO, request);

            } catch (Exception e) {
                log.error(e);
                ServletUtility.handleException(e, request, response);
                return;
            }
        } else if (OP_LOG_OUT.equals(op)) {
            session = request.getSession(false);
            session.invalidate();
            ServletUtility.setSuccessMessage("You have been logged out
successfully", request);

            ServletUtility.forward(ORSView.LOGIN_VIEW, request,
response);
            return;
        }
        if (session.getAttribute("user") != null) {
            ServletUtility.redirect(ORSView.WELCOME_CTL, request,
response);
            return;
        }
        ServletUtility.forward(getView(), request, response);
        log.debug("Method doGet end");
    }

    /**
     * Submit Logic
     */

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        log.debug(" LoginCtl Method doPost Started");

        HttpSession session = request.getSession(true);

        String op = DataUtility.getString(request.getParameter("operation"));
        // get Model
        UserModel model = new UserModel();

        long id = DataUtility.getLong(request.getParameter("id"));

        if (OP_SIGN_IN.equalsIgnoreCase(op)) {
            UserDTO bean = (UserDTO) populateBean(request);
            try {
                bean = model.authenticate(bean.getLogin(),
bean.getPassword());

                if (bean != null) {
                    session.setAttribute("user", bean);
                    session.setMaxInactiveInterval(10 * 6000);


                    // save state of session remember URL
                    String uri = request.getParameter("uri");

                    if (uri == null || "null".equalsIgnoreCase(uri)) {

    ServletUtility.redirect(ORSView.WELCOME_CTL, request, response);
                        return;
                    } else {
                        ServletUtility.redirect(uri, request,
response);
                    }
                    return;
                } else {
                    bean = (UserDTO) populateBean(request);
                    ServletUtility.setBean(bean, request);
                    ServletUtility.setErrorMessage("Invalid LoginId
And Password", request);
                }

            } catch (ApplicationException e) {
                log.error(e);
                ServletUtility.handleException(e, request, response);

                return;
            }
        } else if (OP_SIGN_UP.equalsIgnoreCase(op)) {

    ServletUtility.redirect(ORSView.USER_REGISTRATION_CTL, request,
response);
            return;
        }

        ServletUtility.forward(getView(), request, response);
        log.debug("UserCtl Method doPost Ended");
    }

    /**
     * Returns the VIEW page of this Controller
     *
     * @return
```

```
	 */
	@Override
	protected String getView() {
		return ORSView.LOGIN_VIEW;
	}

}
```

- **Registration Module**

```
package com.online.photography.controller;

import java.io.IOException;


import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import com.online.photography.dto.BaseBean;
import com.online.photography.dto.UserDTO;
import com.online.photography.exception.ApplicationException;
import com.online.photography.exception.DuplicateRecordException;
import com.online.photography.model.UserModel;
import com.online.photography.util.DataUtility;
import com.online.photography.util.DataValidator;
import com.online.photography.util.PropertyReader;
import com.online.photography.util.ServletUtility;



/**
 * Servlet implementation class UserRegistrationCtl
 */

/**
 * UserRegistration functionality Controller. Performs operation for Validate and add a
User
 * As Student Role
 *
 * @author NAvigable set
 * @version 1.0
 * @Copyright (c) Navigable Set
 *
 */

@WebServlet(name = "UserRegistrationCtl", urlPatterns = { "/UserRegistrationCtl" })
public class UserRegistrationCtl extends BaseCtl {
	public static final String OP_SIGN_UP = "SignUp";

	private static Logger log = Logger.getLogger(UserRegistrationCtl.class);
	/**
```

```
	 * Validate input Data Entered By User
	 *
	 * @param request
	 * @return
	 */
	@Override
	protected boolean validate(HttpServletRequest request) {
		log.debug("UserRegistrationCtl Method validate Started");

		boolean pass = true;

		String login = request.getParameter("login");


		if (DataValidator.isNull(request.getParameter("name"))) {
			request.setAttribute("name",
					PropertyReader.getValue("error.require", "First
Name"));
			pass = false;
		} else if (!DataValidator.isName(request.getParameter("name"))) {
			request.setAttribute("name",
					PropertyReader.getValue("error.name", "First
Name"));
			pass = false;
		}


		if (DataValidator.isNull(login)) {
			request.setAttribute("login",
					PropertyReader.getValue("error.require", "Login
Id"));
			pass = false;
		} else if (!DataValidator.isEmail(request.getParameter("login"))) {
			request.setAttribute("login",
					PropertyReader.getValue("error.email", "Login"));
			pass = false;
		}

		if ("-----Select-----".equalsIgnoreCase(request.getParameter("roleId"))) {
			request.setAttribute("roleId",
					PropertyReader.getValue("error.require",
"Catagary"));
			pass = false;
		}



		if (DataValidator.isNull(request.getParameter("password"))) {
			request.setAttribute("password",
                                        PropertyReader.getValue("error.require",
"Password"));
                        pass = false;

                } else if (!DataValidator.isPassword(request.getParameter("password")))
{
                        request.setAttribute("password",
                                        PropertyReader.getValue("error.password",
"Password"));
                        return false;
                }else if (!DataValidator.isPassword(request.getParameter("password")))
{
                        request.setAttribute("password",
                                        PropertyReader.getValue("error.password",
"Password"));
                        return false;
                }



                if (DataValidator.isNull(request.getParameter("mobile"))) {
                        request.setAttribute("mobile",
PropertyReader.getValue("error.require","Mobile No"));
                        pass = false;
                }else if(!DataValidator.isPhoneNo(request.getParameter("mobile"))){
                        request.setAttribute("mobile",
PropertyReader.getValue("error.invalid","Mobile No"));
                        pass=false;
                }


                if (DataValidator.isNull(request.getParameter("cnic"))) {
                        request.setAttribute("cnic",
PropertyReader.getValue("error.require","CNIC"));
                        pass = false;
                }

                if (DataValidator.isNull(request.getParameter("address"))) {
                        request.setAttribute("address",
PropertyReader.getValue("error.require","Address"));
                        pass = false;
                }

                        log.debug("UserRegistrationCtl Method validate Ended");
                return pass;
        }

        /**
         * Populates bean object from request parameters
         *
```

```
     * @param request
     * @return
     */
    @Override
    protected BaseBean populateBean(HttpServletRequest request) {
            log.debug("UserRegistrationCtl Method populatebean Started");

            UserDTO bean = new UserDTO();

            bean.setId(DataUtility.getLong(request.getParameter("id")));
            bean.setName(DataUtility.getString(request.getParameter("name")));


    bean.setLogin(DataUtility.getString(request.getParameter("login")));


    bean.setPassword(DataUtility.getString(request.getParameter("password")));
                    bean.setCellNo(DataUtility.getString("mobile"));

    bean.setCNIC(DataUtility.getString(request.getParameter("cnic")));

    bean.setRoleId(DataUtility.getInt(request.getParameter("roleId")));

    bean.setAddress(DataUtility.getString(request.getParameter("address")));






                    log.debug("UserRegistrationCtl Method populatebean Ended");

                    return bean;
    }

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public UserRegistrationCtl() {
            super();
            // TODO Auto-generated constructor stub
    }
    /**
     * Contains display logic
     */
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
```

```
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		log.debug("UserRegistrationCtl Method doGet Started");
		ServletUtility.forward(getView(), request, response);

	}
	/**
	 * Contains submit logic
	 */
	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		System.out.println("in post method");
		log.debug("UserRegistrationCtl Method doPost Started");

		String op = DataUtility.getString(request.getParameter("operation"));
		// get model
		UserModel model = new UserModel();

		long id = DataUtility.getLong(request.getParameter("id"));


		if (OP_SIGN_UP.equalsIgnoreCase(op)) {

			UserDTO bean = (UserDTO) populateBean(request);
			if(bean.getRoleId()==2) {
				bean.setRoleName("Photographer");
			}else if(bean.getRoleId()==3) {
				bean.setRoleName("Voter");
			}
			try {
			//	System.out.println("in try sign up");
				long pk = model.registerUser(bean);
				//System.out.println("register");
				bean.setId(pk);

				request.getSession().setAttribute("UserDTO", bean);
				ServletUtility.setSuccessMessage("User Successfully
Registered", request);

	ServletUtility.forward(ORSView.USER_REGISTRATION_VIEW, request,
response);
				return;
			} catch (DuplicateRecordException e) {
				log.error(e);
				ServletUtility.setBean(bean, request);
```

```
                                ServletUtility.setErrorMessage("Login id already exists",
                                                request);
                                ServletUtility.forward(getView(), request, response);
                        } catch (ApplicationException e) {
                                ServletUtility.handleException(e, request, response);
                                e.printStackTrace();
                                return;
                        }
                }else if (OP_RESET.equalsIgnoreCase(op)) {

        ServletUtility.redirect(ORSView.USER_REGISTRATION_CTL, request,
response);
                        return;
                }
                log.debug("UserRegistrationCtl Method doPost Ended");
        }
        /**
         * Returns the VIEW page of this Controller
         *
         * @return
         */
        @Override
        protected String getView() {
                return ORSView.USER_REGISTRATION_VIEW;
        }

}
```

- **User Model**

```
package com.online.photography.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import org.apache.log4j.Logger;

import com.online.photography.dto.UserDTO;
import com.online.photography.exception.ApplicationException;
import com.online.photography.exception.DatabaseException;
import com.online.photography.exception.DuplicateRecordException;
import com.online.photography.util.EmailBuilder;
import com.online.photography.util.EmailMessage;
import com.online.photography.util.EmailUtility;
import com.online.photography.util.JDBCDataSource;


/**
 * JDBC Implementation of UserModel
 *
 * @author Navigable Set
 * @version 1.0
 * @Copyright (c) Navigable Set
 */
public class UserModel {
        private static Logger log = Logger.getLogger(UserModel.class);


        public Integer nextPK() throws DatabaseException {
                log.debug("Model nextPK Started");
                Connection conn = null;
                int pk = 0;

                try {
                        conn = JDBCDataSource.getConnection();
                        PreparedStatement pstmt = conn.prepareStatement("SELECT
MAX(ID) FROM USER");
                        ResultSet rs = pstmt.executeQuery();
                        while (rs.next()) {
                                pk = rs.getInt(1);
                        }
```

```
            rs.close();

        } catch (Exception e) {
            log.error("Database Exception..", e);
            throw new DatabaseException("Exception : Exception in getting
PK");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }
        log.debug("Model nextPK End");
        return pk + 1;
    }

    /**
     * Add a User
     *
     * @param bean
     * @throws DatabaseException
     *
     */
    public long add(UserDTO bean) throws ApplicationException,
DuplicateRecordException {

        Connection conn = null;
        int pk = 0;

        UserDTO existbean = findByLogin(bean.getLogin());

        if (existbean != null) {
            throw new DuplicateRecordException("Login Id already
exists");
        }

        try {
            conn = JDBCDataSource.getConnection();
            pk = nextPK();
            // Get auto-generated next primary key
            System.out.println(pk + " in ModelJDBC");
            conn.setAutoCommit(false); // Begin transaction
            PreparedStatement pstmt = conn.prepareStatement("INSERT
INTO USER VALUES(?,?,?,?,?,?,?,?,?)");
            pstmt.setInt(1, pk);
            pstmt.setString(2, bean.getName());
            pstmt.setString(3, bean.getLogin());
            pstmt.setString(4, bean.getPassword());
            pstmt.setString(5, bean.getCNIC());
            pstmt.setString(6, bean.getCellNo());
            pstmt.setString(7, bean.getAddress());
            pstmt.setLong(8, bean.getRoleId());
            pstmt.setString(9, bean.getRoleName());
            pstmt.executeUpdate();
            conn.commit(); // End transaction
            pstmt.close();
        } catch (Exception e) {

            try {
                conn.rollback();
            } catch (Exception ex) {
                ex.printStackTrace();
                throw new ApplicationException("Exception : add
rollback exception " + ex.getMessage());
            }
            throw new ApplicationException("Exception : Exception in add
User");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }

        return pk;
    }

    /**
     * Delete a User
     *
     * @param bean
     * @throws DatabaseException
     */
    public void delete(UserDTO bean) throws ApplicationException {

        Connection conn = null;
        try {
            conn = JDBCDataSource.getConnection();
            conn.setAutoCommit(false); // Begin transaction
            PreparedStatement pstmt = conn.prepareStatement("DELETE
FROM USER WHERE ID=?");
            pstmt.setLong(1, bean.getId());
            pstmt.executeUpdate();
            conn.commit(); // End transaction
            pstmt.close();

        } catch (Exception e) {

            try {
                conn.rollback();
            } catch (Exception ex) {
                throw new ApplicationException("Exception : Delete
rollback exception " + ex.getMessage());
            }
            throw new ApplicationException("Exception : Exception in
delete User");
```

```
        } finally {
            JDBCDataSource.closeConnection(conn);
        }

    }

    /**
     * Find User by Login
     *
     * @param login
     *          : get parameter
     * @return bean
     * @throws DatabaseException
     */

    public UserDTO findByLogin(String login) throws ApplicationException {
        log.debug("Model findByLogin Started");
        StringBuffer sql = new StringBuffer("SELECT * FROM USER
WHERE LOGIN=?");
        UserDTO bean = null;
        Connection conn = null;
        System.out.println("sql" + sql);

        try {
            conn = JDBCDataSource.getConnection();
            PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
            pstmt.setString(1, login);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                bean = new UserDTO();
                bean.setId(rs.getLong(1));
                bean.setName(rs.getString(2));
                bean.setLogin(rs.getString(3));
                bean.setPassword(rs.getString(4));
                bean.setCNIC(rs.getString(5));
                bean.setCellNo(rs.getString(6));
                bean.setAddress(rs.getString(7));
                bean.setRoleId(rs.getLong(8));
                bean.setRoleName(rs.getString(9));

            }
            rs.close();
        } catch (Exception e) {
            e.printStackTrace();
            log.error("Database Exception..", e);
            throw new ApplicationException("Exception : Exception in
getting User by login");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }
        log.debug("Model findByLogin End");
        return bean;
    }

    /**
     * Find User by PK
     *
     * @param pk
     *          : get parameter
     * @return bean
     * @throws DatabaseException
     */

    public UserDTO findByPK(long pk) throws ApplicationException {
        log.debug("Model findByPK Started");
        StringBuffer sql = new StringBuffer("SELECT * FROM USER 
WHERE ID=?");
        UserDTO bean = null;
        Connection conn = null;

        try {
            conn = JDBCDataSource.getConnection();
            PreparedStatement pstmt = 
conn.prepareStatement(sql.toString());
            pstmt.setLong(1, pk);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                bean = new UserDTO();
                bean.setId(rs.getLong(1));
                bean.setName(rs.getString(2));
                bean.setLogin(rs.getString(3));
                bean.setPassword(rs.getString(4));
                bean.setCNIC(rs.getString(5));
                bean.setCellNo(rs.getString(6));
                bean.setAddress(rs.getString(7));
                bean.setRoleId(rs.getLong(8));
                bean.setRoleName(rs.getString(9));

            }
            rs.close();
        } catch (Exception e) {
            e.printStackTrace();
            log.error("Database Exception..", e);
            throw new ApplicationException("Exception : Exception in 
getting User by pk");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }
        log.debug("Model findByPK End");
        return bean;
    }

    /**
     * Update a user
     *
     * @param bean
     * @throws DatabaseException
     */

    /*public void update(UserDTO bean) throws ApplicationException,
DuplicateRecordException {
        log.debug("Model update Started");
        Connection conn = null;

        UserDTO beanExist = findByLogin(bean.getLogin());
        // Check if updated LoginId already exist
        if (beanExist != null && !(beanExist.getId() == bean.getId())) {
            throw new DuplicateRecordException("LoginId is already
exist");
        }

        try {
            conn = JDBCDataSource.getConnection();
            conn.setAutoCommit(false); // Begin transaction
            PreparedStatement pstmt = conn.prepareStatement(
                    "UPDATE ST_USER SET
FIRST_NAME=?,LAST_NAME=?,LOGIN=?,PASSWORD=?,DOB=?,MOBILE_NO
=?,ROLE_ID=?,"
                    +
"UNSUCCESSFUL_LOGIN=?,GENDER=?,LAST_LOGIN=?,USER_LOCK=?,REGI
STERED_IP=?,LAST_LOGIN_IP=?,"
                    +
"CREATED_BY=?,MODIFIED_BY=?,CREATED_DATETIME=?,MODIFIED_DAT
ETIME=? WHERE ID=?");
            pstmt.setString(1, bean.getFirstName());
            pstmt.setString(2, bean.getLastName());
            pstmt.setString(3, bean.getLogin());
            pstmt.setString(4, bean.getPassword());
            pstmt.setDate(5, new java.sql.Date(bean.getDob().getTime()) );
            pstmt.setString(6, bean.getMobileNo());
            pstmt.setLong(7, bean.getRoleId());
            pstmt.setInt(8, bean.getUnSuccessfulLogin());
            pstmt.setString(9, bean.getGender());
            pstmt.setTimestamp(10, bean.getLastLogin());
            pstmt.setString(11, bean.getLock());
            pstmt.setString(12, bean.getRegisteredIP());
            pstmt.setString(13, bean.getLastLoginIP());
            pstmt.setString(14, bean.getCreatedBy());
            pstmt.setString(15, bean.getModifiedBy());
            pstmt.setTimestamp(16, bean.getCreatedDatetime());
            pstmt.setTimestamp(17, bean.getModifiedDatetime());
            pstmt.setLong(18, bean.getId());
            pstmt.executeUpdate();
            conn.commit(); // End transaction
            pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
            log.error("Database Exception..", e);
            try {
                conn.rollback();
            } catch (Exception ex) {
                throw new ApplicationException("Exception : Delete
rollback exception " + ex.getMessage());
            }
            throw new ApplicationException("Exception in updating User ");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }
        log.debug("Model update End");
    }*/

    /**
     * Search User
     *
     * @param bean
     *         : Search Parameters
     * @throws DatabaseException
     */

    /*public List search(UserDTO bean) throws ApplicationException {
        return search(bean, 0, 0);
    }*/

    /**
     * Search User with pagination
     *
     * @return list : List of Users
     * @param bean
     *         : Search Parameters
     * @param pageNo
     *         : Current Page No.
     * @param pageSize
     *         : Size of Page
     *
     * @throws DatabaseException
     */

    /*public List search(UserDTO bean, int pageNo, int pageSize) throws
ApplicationException {
        log.debug("Model search Started");
        StringBuffer sql = new StringBuffer("SELECT * FROM ST_USER
WHERE 1=1");

        if (bean != null) {
            if (bean.getId() > 0) {
                sql.append(" AND id = " + bean.getId());
            }
            if (bean.getFirstName() != null && bean.getFirstName().length()
> 0) {
                sql.append(" AND FIRST_NAME like '" +
bean.getFirstName() + "%'");
            }
            if (bean.getLastName() != null && bean.getLastName().length()
> 0) {
                sql.append(" AND LAST_NAME like '" +
bean.getLastName() + "%'");
            }
            if (bean.getLogin() != null && bean.getLogin().length() > 0) {
                sql.append(" AND LOGIN like '" + bean.getLogin() +
"%'");
            }
            if (bean.getPassword() != null && bean.getPassword().length() >
0) {
                sql.append(" AND PASSWORD like '" +
bean.getPassword() + "%'");
            }
            if (bean.getDob() != null && bean.getDob().getDate() > 0) {
                sql.append(" AND DOB = " + bean.getGender());
            }
            if (bean.getMobileNo() != null && bean.getMobileNo().length()
> 0) {
                sql.append(" AND MOBILE_NO = " +
bean.getMobileNo());
            }
            if (bean.getRoleId() > 0) {
                sql.append(" AND ROLE_ID = " + bean.getRoleId());
            }
            if (bean.getUnSuccessfulLogin() > 0) {
                sql.append(" AND UNSUCCESSFUL_LOGIN = " +
bean.getUnSuccessfulLogin());
            }
            if (bean.getGender() != null && bean.getGender().length() > 0) {
                sql.append(" AND GENDER like '" + bean.getGender() +
"%'");
            }
            if (bean.getLastLogin() != null &&
bean.getLastLogin().getTime() > 0) {
                sql.append(" AND LAST_LOGIN = " +
bean.getLastLogin());
```

```
            }
            if (bean.getRegisteredIP() != null &&
bean.getRegisteredIP().length() > 0) {
                sql.append(" AND REGISTERED_IP like '" +
bean.getRegisteredIP() + "%'");
            }
            if (bean.getLastLoginIP() != null &&
bean.getLastLoginIP().length() > 0) {
                sql.append(" AND LAST_LOGIN_IP like '" +
bean.getLastLoginIP() + "%'");
            }

        }

        // if page size is greater than zero then apply pagination
        if (pageSize > 0) {
            // Calculate start record index
            pageNo = (pageNo - 1) * pageSize;

            sql.append(" Limit " + pageNo + ", " + pageSize);
            // sql.append(" limit " + pageNo + "," + pageSize);
        }

        System.out.println("user model search  :"+sql);
        ArrayList list = new ArrayList();
        Connection conn = null;
        try {
            conn = JDBCDataSource.getConnection();
            PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                bean = new UserDTO();
                bean.setId(rs.getLong(1));
                bean.setFirstName(rs.getString(2));
                bean.setLastName(rs.getString(3));
                bean.setLogin(rs.getString(4));
                bean.setPassword(rs.getString(5));
                bean.setDob(rs.getDate(6));
                bean.setMobileNo(rs.getString(7));
                bean.setRoleId(rs.getLong(8));
                bean.setUnSuccessfulLogin(rs.getInt(9));
                bean.setGender(rs.getString(10));
                bean.setLastLogin(rs.getTimestamp(11));
                bean.setLock(rs.getString(12));
                bean.setRegisteredIP(rs.getString(13));
                bean.setLastLoginIP(rs.getString(14));
                bean.setCreatedBy(rs.getString(15));
                bean.setModifiedBy(rs.getString(16));
                bean.setCreatedDatetime(rs.getTimestamp(17));
                bean.setModifiedDatetime(rs.getTimestamp(18));

                list.add(bean);
            }
            rs.close();
        } catch (Exception e) {
            log.error("Database Exception..", e);
            throw new ApplicationException("Exception : Exception in
search user");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }

        log.debug("Model search End");
        return list;
    }*/

    /**
     * Get List of User
     *
     * @return list : List of User
     * @throws DatabaseException
     */

    public List list() throws ApplicationException {
        return list(0, 0);
    }

    /**
     * Get List of User with pagination
     *
     * @return list : List of users
     * @param pageNo
     *            : Current Page No.
     * @param pageSize
     *            : Size of Page
     * @throws DatabaseException
     */

    public List list(int pageNo, int pageSize) throws ApplicationException {
        log.debug("Model list Started");
        ArrayList list = new ArrayList();
        StringBuffer sql = new StringBuffer("select * from USER");
        // if page size is greater than zero then apply pagination
        if (pageSize > 0) {
            // Calculate start record index
            pageNo = (pageNo - 1) * pageSize;
            sql.append(" limit " + pageNo + "," + pageSize);
        }
        System.out.println("sql in list user :"+sql);
        Connection conn = null;

        try {
            conn = JDBCDataSource.getConnection();
            PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                UserDTO bean = new UserDTO();
                bean.setId(rs.getLong(1));
                bean.setId(rs.getLong(1));
                bean.setName(rs.getString(2));
                bean.setLogin(rs.getString(3));
                bean.setPassword(rs.getString(4));
                bean.setCNIC(rs.getString(5));
                bean.setCellNo(rs.getString(6));
                bean.setAddress(rs.getString(7));
                bean.setRoleId(rs.getLong(8));
                bean.setRoleName(rs.getString(9));

                list.add(bean);
            }
            rs.close();
        } catch (Exception e) {
            log.error("Database Exception..", e);
            throw new ApplicationException("Exception : Exception in
getting list of users");
        } finally {
            JDBCDataSource.closeConnection(conn);
        }

        log.debug("Model list End");
        return list;

    }

    /**
     * @param id
     *         : long id
     * @param old
     *         password : String oldPassword
     * @param new
     *         password : String newPassword
     * @throws DatabaseException
     */

    public UserDTO authenticate(String login, String password) throws
ApplicationException {
		log.debug("Model authenticate Started");
		StringBuffer sql = new StringBuffer("SELECT * FROM USER
WHERE LOGIN = ? AND PASSWORD = ?");
		UserDTO bean = null;
		Connection conn = null;

		try {
			conn = JDBCDataSource.getConnection();
			PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
			pstmt.setString(1, login);
			pstmt.setString(2, password);
			ResultSet rs = pstmt.executeQuery();
			while (rs.next()) {
				bean = new UserDTO();
				bean.setId(rs.getLong(1));
				bean.setId(rs.getLong(1));
				bean.setName(rs.getString(2));
				bean.setLogin(rs.getString(3));
				bean.setPassword(rs.getString(4));
				bean.setCNIC(rs.getString(5));
				bean.setCellNo(rs.getString(6));
				bean.setAddress(rs.getString(7));
				bean.setRoleId(rs.getLong(8));
				bean.setRoleName(rs.getString(9));
				System.out.println("Usermodel here");
			}
		} catch (Exception e) {
			log.error("Database Exception..", e);
			throw new ApplicationException("Exception : Exception in get
roles");

		} finally {
			JDBCDataSource.closeConnection(conn);
		}

		log.debug("Model authenticate End");
		return bean;
	}

	/**
	 * Lock User for certain time duration
	 *
	 * @return boolean : true if success otherwise false
	 * @param login
	 *          : User Login
	 * @throws ApplicationException
	 * @throws RecordNotFoundException
	 *           : if user not found
	 */
```

```
/**
 * Get User Roles
 *
 * @return List : User Role List
 * @param bean
 * @throws ApplicationException
 */




/**
 * Register a user
 *
 * @param bean
 * @throws ApplicationException
 * @throws DuplicateRecordException
 *          : throws when user already exists
 * @throws org.omg.CORBA.portable.ApplicationException
 */

public long registerUser(UserDTO bean)
        throws ApplicationException, DuplicateRecordException {

    log.debug("Model add Started");

    long pk = add(bean);

    HashMap<String, String> map = new HashMap<String, String>();
    map.put("login", bean.getLogin());
    map.put("password", bean.getPassword());

    String message = EmailBuilder.getUserRegistrationMessage(map);

    EmailMessage msg = new EmailMessage();

    msg.setTo(bean.getLogin());
    msg.setSubject("Registration is successful for ORS Project SunilOS");
    msg.setMessage(message);
    msg.setMessageType(EmailMessage.HTML_MSG);

    try {
        EmailUtility.sendMail(msg);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return pk;
	}

	/**
	 * Reset Password of User with auto generated Password
	 *
	 * @return boolean : true if success otherwise false
	 * @param login
	 *          : User Login
	 * @throws ApplicationException
	 * @throws org.omg.CORBA.portable.ApplicationException
	 * @throws RecordNotFoundException
	 *           : if user not found
	 */

	/*public boolean resetPassword(UserDTO bean)
			throws ApplicationException,
org.omg.CORBA.portable.ApplicationException {

		String newPassword = String.valueOf(new
Date().getTime()).substring(0, 4);
		UserDTO userData = findByPK(bean.getId());
		userData.setPassword(newPassword);

		try {
			update(userData);
		} catch (DuplicateRecordException e) {
			return false;
		}

		HashMap<String, String> map = new HashMap<String, String>();
		map.put("login", bean.getLogin());
		map.put("password", bean.getPassword());
		map.put("firstName", bean.getFirstName());
		map.put("lastName", bean.getLastName());

		String message = EmailBuilder.getForgetPasswordMessage(map);

		EmailMessage msg = new EmailMessage();

		msg.setTo(bean.getLogin());
		msg.setSubject("Password has been reset");
		msg.setMessage(message);
		msg.setMessageType(EmailMessage.HTML_MSG);

		EmailUtility.sendMail(msg);

		return true;
	}*/

	/**
```

```
        * Send the password of User to his Email
        *
        * @return boolean : true if success otherwise false
        * @param login
        *           : User Login
        * @throws ApplicationException
        * @throws RecordNotFoundException
        *            : if user not found
        *
        */

}
```

- **Competition Model**

```
package com.online.photography.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import org.apache.log4j.Logger;

import com.online.photography.dto.CompititionDTO;
import com.online.photography.dto.ImageDTO;
import com.online.photography.dto.UserDTO;
import com.online.photography.exception.ApplicationException;
import com.online.photography.exception.DatabaseException;
import com.online.photography.exception.DuplicateRecordException;
import com.online.photography.util.JDBCDataSource;

public class CompititionModel {

        private static Logger log = Logger.getLogger(CompititionModel.class);


        public Integer nextPK() throws DatabaseException {
                log.debug("Model nextPK Started");
                Connection conn = null;
                int pk = 0;

                try {
                        conn = JDBCDataSource.getConnection();
                        PreparedStatement pstmt = conn.prepareStatement("SELECT
MAX(ID) FROM Compitition");
                        ResultSet rs = pstmt.executeQuery();
                        while (rs.next()) {
                                pk = rs.getInt(1);
                        }
                        rs.close();

                } catch (Exception e) {
                        log.error("Database Exception..", e);
                        throw new DatabaseException("Exception : Exception in getting
PK");
                } finally {
                        JDBCDataSource.closeConnection(conn);
                }
                log.debug("Model nextPK End");
                return pk + 1;
        }

        public long add(CompititionDTO bean) throws ApplicationException,
DuplicateRecordException {

                Connection conn = null;
                int pk = 0;

                /*UserDTO existbean = findByLogin(bean.getLogin());

                if (existbean != null) {
                        throw new DuplicateRecordException("Login Id already exists");
                }*/

                try {
                        conn = JDBCDataSource.getConnection();
                        pk = nextPK();
                        // Get auto-generated next primary key
                        System.out.println(pk + " in ModelJDBC");
                        conn.setAutoCommit(false); // Begin transaction
                        PreparedStatement pstmt = conn.prepareStatement("INSERT
INTO Compitition VALUES(?,?,?,?)");
                        pstmt.setInt(1, pk);
                        pstmt.setString(2,bean.getcName());
                        pstmt.setString(3,bean.getcType());
                        pstmt.setString(4,bean.getcDate());
                        pstmt.executeUpdate();
                        conn.commit(); // End transaction
                        pstmt.close();
                } catch (Exception e) {

                        try {
                                conn.rollback();
                        } catch (Exception ex) {
                                ex.printStackTrace();
                                throw new ApplicationException("Exception : add
rollback exception " + ex.getMessage());
                        }
                        throw new ApplicationException("Exception : Exception in add
User");
                } finally {
                        JDBCDataSource.closeConnection(conn);
                }
                return pk;
                }

        public List list() throws ApplicationException {
            return list(0, 0);
        }

        /**
         * Get List of User with pagination
         *
         * @return list : List of users
         * @param pageNo
         *            : Current Page No.
         * @param pageSize
         *            : Size of Page
         * @throws DatabaseException
         */

        public List list(int pageNo, int pageSize) throws ApplicationException {
            log.debug("Model list Started");
            ArrayList list = new ArrayList();
            StringBuffer sql = new StringBuffer("select * from compitition");
            // if page size is greater than zero then apply pagination
            if (pageSize > 0) {
                // Calculate start record index
                pageNo = (pageNo - 1) * pageSize;
                sql.append(" limit " + pageNo + "," + pageSize);
            }


            System.out.println("sql in list user :"+sql);
            Connection conn = null;

            try {
                conn = JDBCDataSource.getConnection();
                PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
                ResultSet rs = pstmt.executeQuery();
                while (rs.next()) {
                    CompititionDTO bean = new CompititionDTO();
                    bean.setId(rs.getLong(1));
                    bean.setcName(rs.getString(2));
                    bean.setcType(rs.getString(3));
                    bean.setcDate(rs.getString(4));

                    list.add(bean);
                }
                rs.close();
            } catch (Exception e) {
                log.error("Database Exception..", e);
                throw new ApplicationException("Exception : Exception in
getting list of users");
            } finally {
                JDBCDataSource.closeConnection(conn);
            }

            log.debug("Model list End");
            return list;

        }

        public CompititionDTO findByPK(long pk) throws ApplicationException {
            log.debug("SubjectModel findByPK Started");
            StringBuffer sql = new StringBuffer("SELECT * FROM compitition
WHERE ID=?");
            CompititionDTO bean = null;
            Connection conn = null;
            try {
                conn = JDBCDataSource.getConnection();
                PreparedStatement pstmt =
conn.prepareStatement(sql.toString());
                pstmt.setLong(1, pk);
                ResultSet rs = pstmt.executeQuery();
                while (rs.next()) {
                    bean = new CompititionDTO();
                    bean.setId(rs.getLong(1));
                    bean.setcName(rs.getString(2));
                    bean.setcType(rs.getString(3));
                    bean.setcDate(rs.getString(4));
                }
                rs.close();
            } catch (Exception e) {
                log.error("Database Exception..", e);
                throw new ApplicationException("Exception : Exception in
getting User by pk");
            } finally {
                JDBCDataSource.closeConnection(conn);
            }
            log.debug("SubjectModel findByPK End");
            return bean;
        }

}
```

CHAPTER-6

Testing

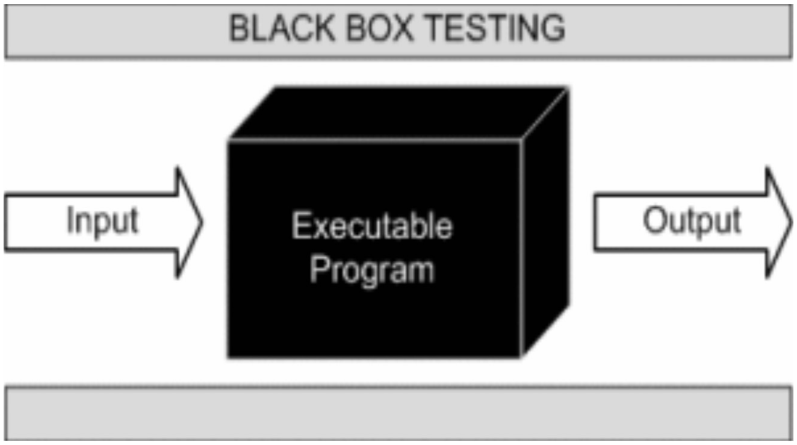
6.1 Test Case – 1

Black-Box Testing

Black Box Testing, also known as Behavioural Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This can be following way:

- Input interfacing
- Processing
- Output interfacing



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behaviour or performance errors
- Initialization and termination errors.

6.2 Test Case – 2

White-Box Testing

White Box Testing ,also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential.

White box testing is testing beyond the user interface and into the nitty-gritty of a system.
This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

Limitations and Future Application of the Project

Futures Enhancement:

- In future we can expand this project on the cloud.
- In future, we can allow admin to allow add prize module for the winner.

Limitation :

In this, we don’t have a facility to block users from the system.

Conclusion

Perfect Picture Pool activities can be managed more efficiently using our application. Everything will be handled in one place.

This Perfect Picture Pool application help the competitor to face the live audience around the world and also User from different corners of the world can take a look at the beautiful photographs and can vote for their favorite photo. This application would help the talented people to get noticed.

This application is designed in such a way that any future modification can be done most easily.