Chapter 1

INTRODUCTION

1.1 INTRODUCTION

On-line examinations contents providers to focus on creating effective assessment questions and focusing on exam's feedback delivery to students. In the paper we present techniques that are pertinent to the elements of assessment process: answers submission, computerized grading, and feedback after submission. As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization.

1.2 PURPOSE

Existing system is a manual one in which users are maintaining books to store the information like Student Details, Instructor Details, Schedule Details and feedbacks about students who attempted exam as per schedule. It is very difficult to maintain historical data.

The objective of the Online Examination Tool is to provide better information for the users of this system for better results for their maintenance in student examination schedule details and grading details.

1.3 PROJECT SCOPE

The administrators, instructor, Students who are attending for online examination can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of Online Examinations like conducting Exams as per scheduled basis and delivering result to that particular user or student, and the details of students who attempted Online Examination are maintained at administrator.

Chapter 2

REQUIREMENTS SPECIFICATION

2.1 Software Requirements

Software is any set of machine-readable instruction that directs a computers processor to perform specific operations. Software is usually written in high-level programming languages that are easier and more.

2.2 Hardware Requirements

Hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to the physical parts or components of a computer such as monitor, keyboard, computer data storage etc. The minimum Hardware requirement for our project is specified below.

Processor	: Intel core i5.
Hard Disk	: 20 GB, 80 GB, 160 GB or above.
Monitor	: 15.6 VGA colour, 1366*768 resolution
RAM	: 2 GB or above.
Input device	: Keyboard and Mouse.

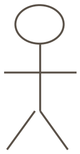
Chapter 3

ANALYSIS AND DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analysed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

3.1 UML Diagrams:

Actor: A coherent set of roles that users of use cases play when interacting with the use `cases.



UML stands for Unified Modelling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram

3.1.1 USE CASE Diagram

A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

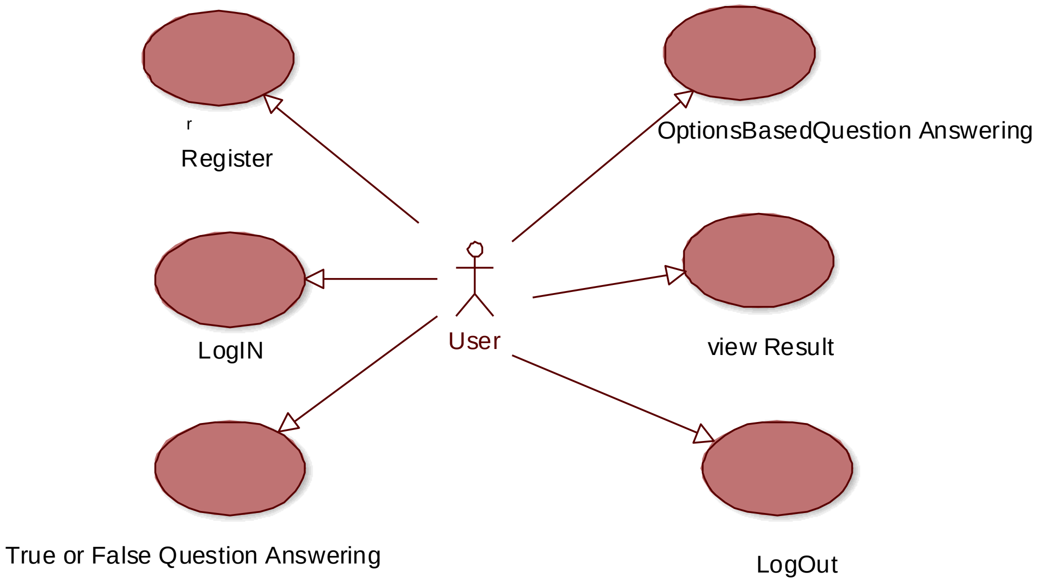
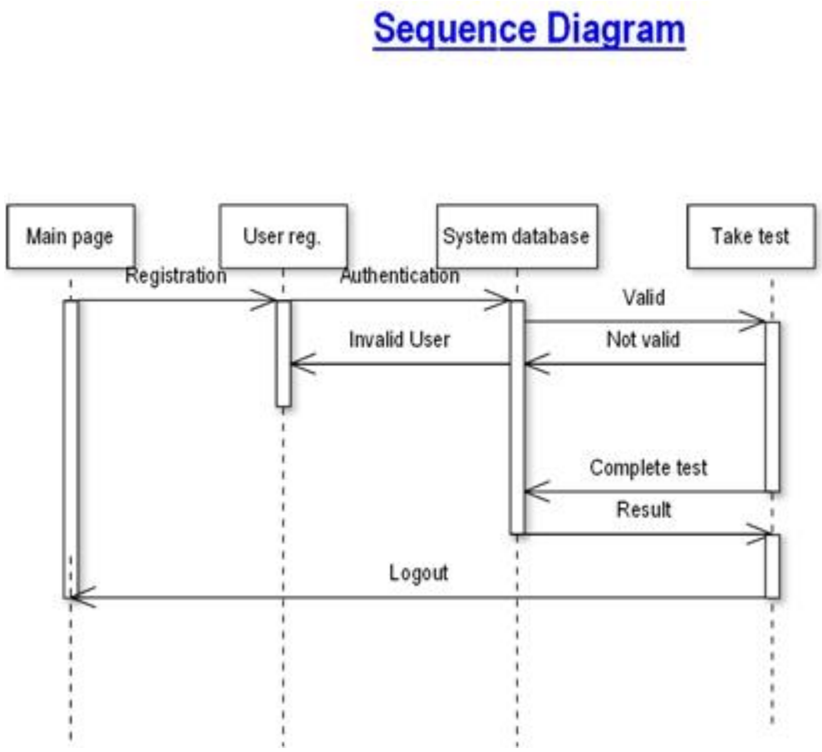

r
Register
OptionsBasedQuestion Answering
LogIN
User
view Result
True or False Question Answering
LogOut

Fig 3.1.1 Use Case Diagram

3.2 SEQUENCE DIAGRAM

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.



Sequence Diagram

Fig 3.2 Sequence Diagram

Chapter 4

TESTING

Testing is a dynamic technique of verification and validation. It involves executing an implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it is performing as required.

The following statements serve as the objectives for testing:

1. Testing is a process of executing a program with the intent of finding error
2. A good test case is one that has a high probability of finding an as-yet undiscovered error.
3. A successful test is one that uncovers as-yet undiscovered error.

Verification and validation is intended to show that a system confirms to its specification and that the system meets the expectations of the customer. Verification involves checking that the software confirms to its specification. We should check that the system meets its specified functional and non-functional requirements. Validation ensures that the software meets the expectations of the customer. It goes beyond checking conformance of the system to its specification to showing that the software does what the customer expects as distinct from what has been specified.

The testing process should proceed in stages where testing is carried out incrementally in conjunction with system implementation. System components are tested, the integrated system is tested and, finally, the system is tested with the customer's data.

The stages in the testing process are:

4.1 Unit testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. Thus, the project has been successfully tested.

4.2 Integration testing.

- Main function is design to call many sub functions, where different options are given in the sub functions. Main function in this software is to maintain a music store details like which albums are present in the shop. Albums are of two types- movie album and singer album.
- Now, different functions are included in the main separately and tested for error. Album details, customer details and sales details are viewed properly.
- The software was compiled and tested and desired output was obtained without any error or exception.

4.3 Sub-system testing

This phase involves testing collections of modules which have been integrated into sub-systems. The sub-system test process should concentrate on the detection of module interface errors by rigorously exercising these interfaces.

4.4 System testing

The sub-systems are integrated to make up the system. This process is concerned with finding errors that result from unanticipated interactions between sub-systems and sub-system interface problems. It is also concerned with validating that the system meets its functional and non-functional requirements and testing the emergent system properties. The software is designed and developed so that it can be run on a system without any errors. The software is tested for this system and its functionality is achieved.

4.5 Acceptance testing

This is the final stage in the testing process before the system is officially accepted for operational use. The system is tested with data supplied by the system customer rather than simulated test data. Objective is to insert, retrieve, delete, and update salesperson details and customer's details. The main objective of this software is to generate the bill for the customer and print it.

Chapter 5

SNAPSHOTS AND RESULTS

The set of tables is created using the relational database for the identified entities at the design stage. The uniqueness of the data fields in these tables are established using primary key, while the relationships are maintained using foreign keys.

The web pages of the information will guide the use and operation of this system. Figure 5.1 illustrate the login home page for our ERP College Management.

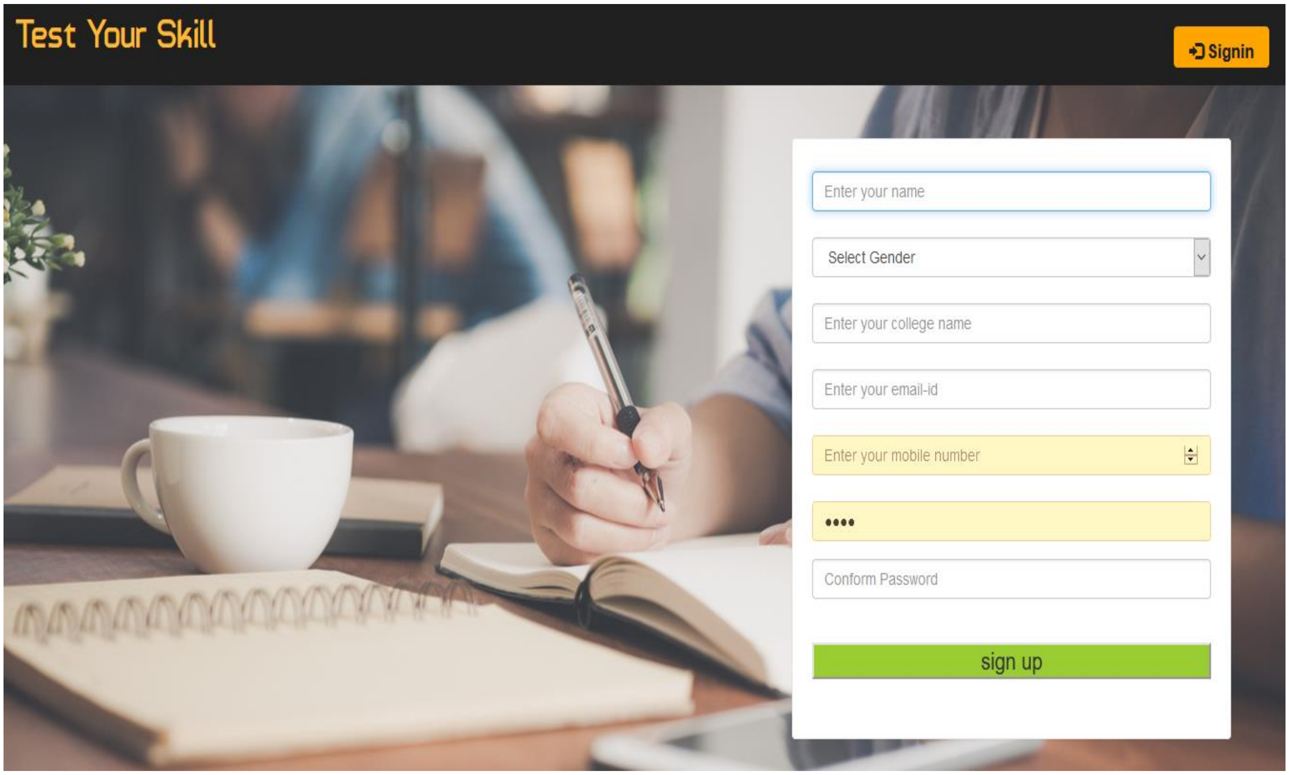
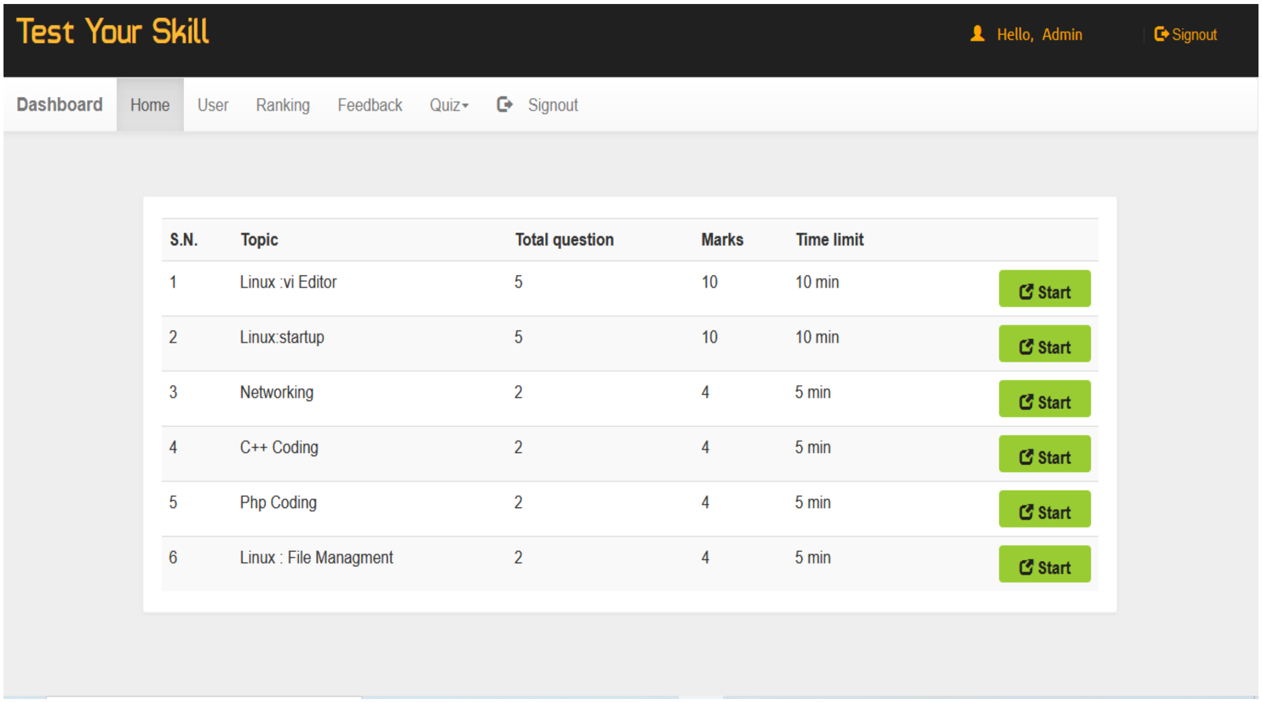
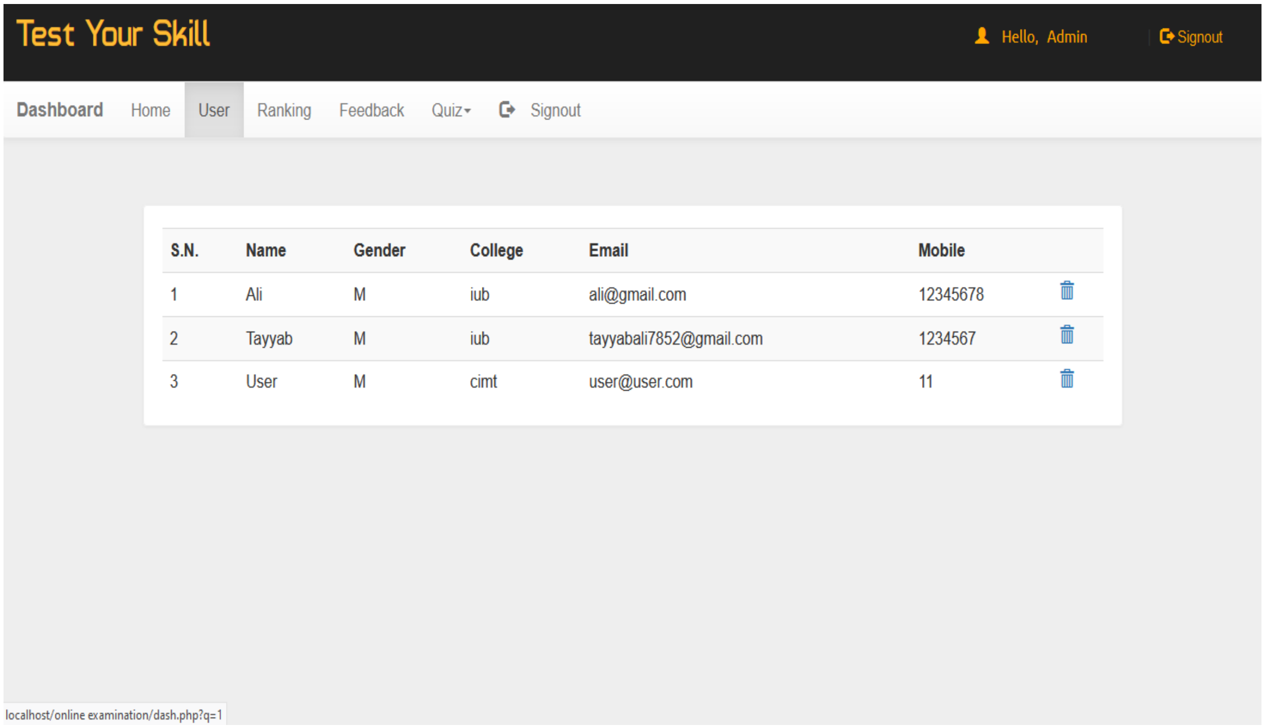


Fig 5.1 Home Page



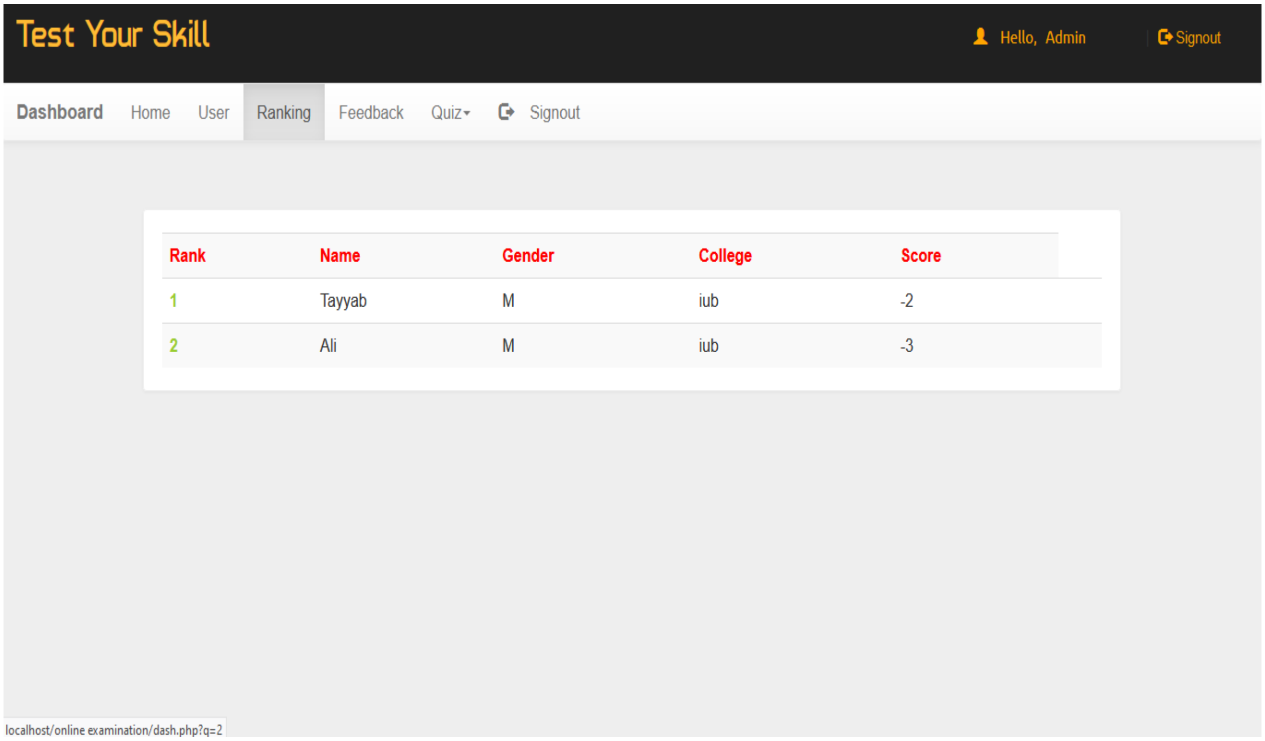
S.N.	Topic	Total question	Marks	Time limit	
1	Linux :vi Editor	5	10	10 min	Start
2	Linux:startup	5	10	10 min	Start
3	Networking	2	4	5 min	Start
4	C++ Coding	2	4	5 min	Start
5	Php Coding	2	4	5 min	Start
6	Linux : File Managment	2	4	5 min	Start

Fig 5.2 Examination Selection Page



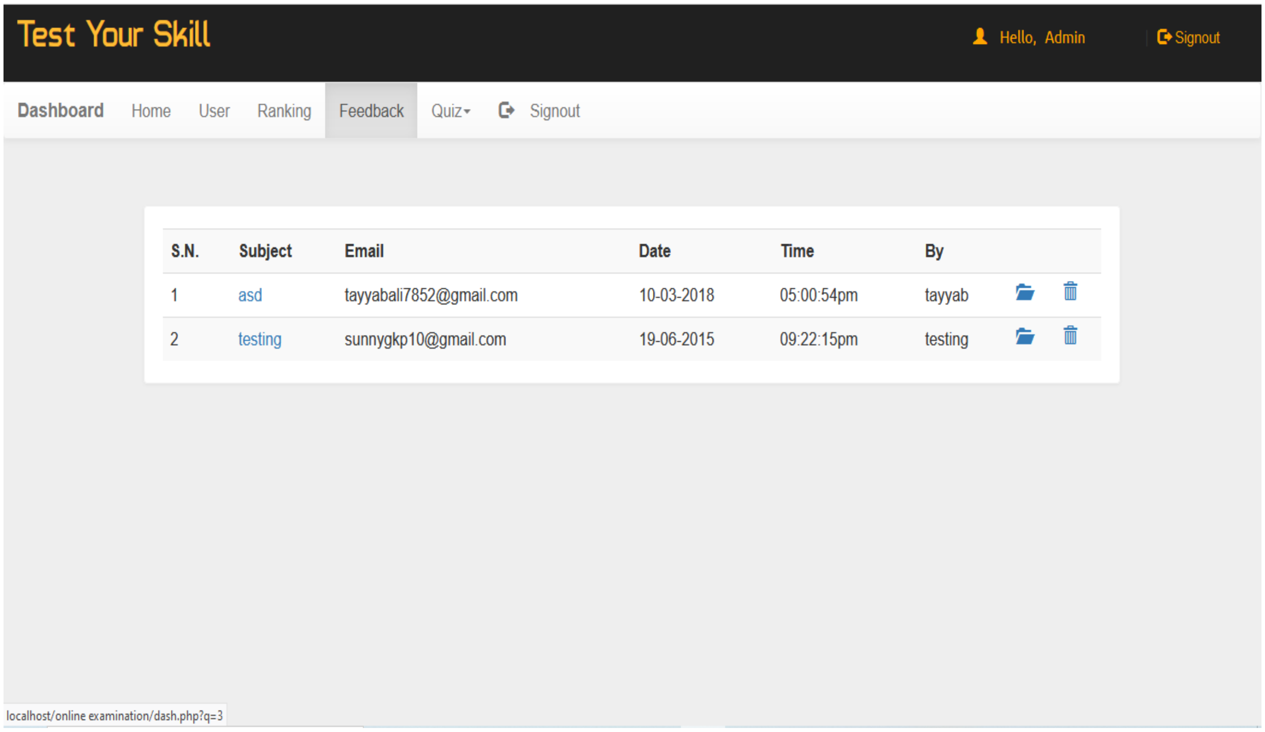
S.N.	Name	Gender	College	Email	Mobile
1	Ali	M	iub	ali@gmail.com	12345678
2	Tayyab	M	iub	tayyabali7852@gmail.com	1234567
3	User	M	cimt	user@user.com	11

Fig 5.3 User Info



Rank	Name	Gender	College	Score
1	Tayyab	M	iub	-2
2	Ali	M	iub	-3

Fig 5.4 Result page



S.N.	Subject	Email	Date	Time	By
1	asd	tayyabali7852@gmail.com	10-03-2018	05:00:54pm	tayyab
2	testing	sunnygkp10@gmail.com	19-06-2015	09:22:15pm	testing

Fig 5.5 Admin Page

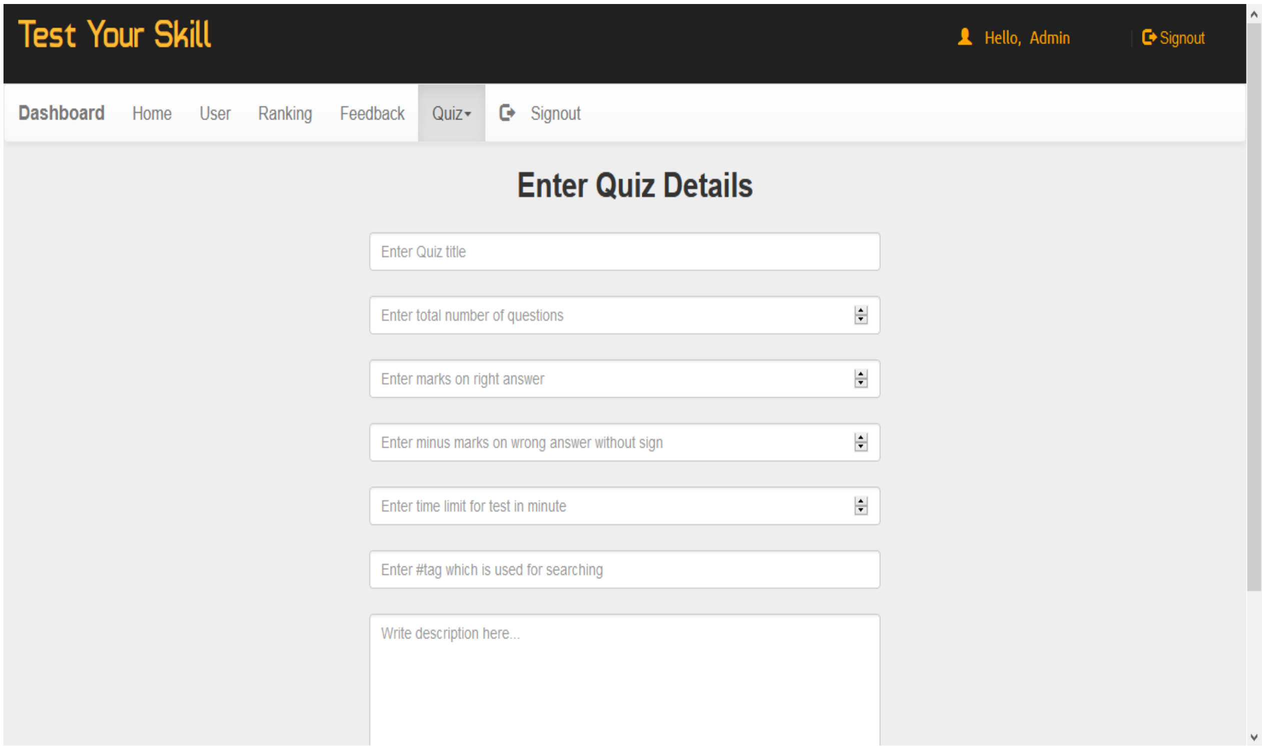
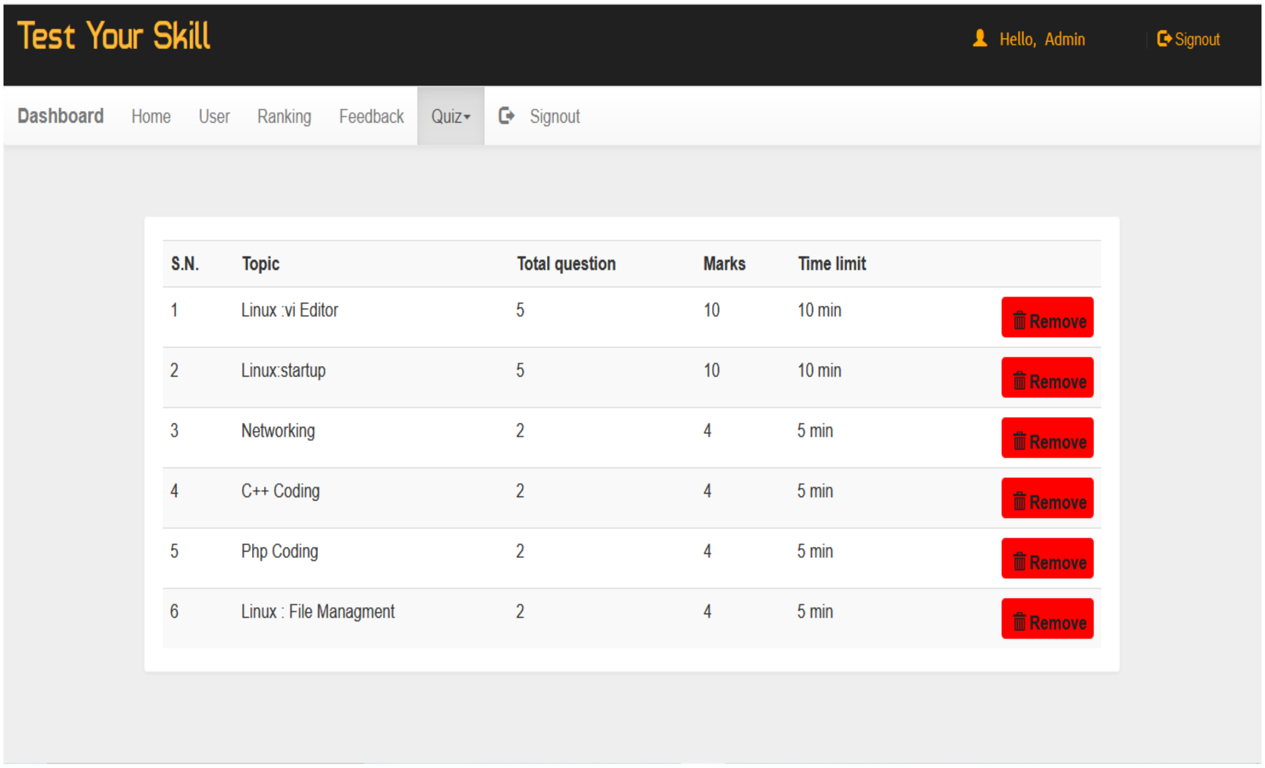

Test Your Skill
Hello, Admin
Signout
Dashboard
Home
User
Ranking
Feedback
Quiz
Signout
Enter Quiz Details
Enter Quiz title
Enter total number of questions
Enter marks on right answer
Enter minus marks on wrong answer without sign
Enter time limit for test in minute
Enter #tag which is used for searching
Write description here...

Fig 5.6 Creating quiz



S.N.	Topic	Total question	Marks	Time limit	
1	Linux :vi Editor	5	10	10 min	Remove
2	Linux:startup	5	10	10 min	Remove
3	Networking	2	4	5 min	Remove
4	C++ Coding	2	4	5 min	Remove
5	Php Coding	2	4	5 min	Remove
6	Linux : File Managment	2	4	5 min	Remove

Fig 5.7 Editing quiz

TECHNOLOGY USED

5.1 PHP

PHP: Hypertext Pre-processor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms. PHP was originally created by Rasmus Lerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by the PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License. PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use. Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems’ Java Server Pages, and mod Perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD).

5.1.1 Working of PHP:

When a client requests web page containing PHP code from the server, then the requested PHP pages are parsed under PHP environment and interaction with database is made if required. After server-side processing, the resulting HTML pages are passed to client and displayed on the browser. In this way the working of php is complete.

5.2 MySQL:

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is officially pronounced ("My S-Q-L") but is often pronounced ("My Sequel"). It is named for original developer Michael Widenius's daughter My. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Sun Microsystems, a subsidiary of Oracle Corporation. MySQL code uses C and C++. The SQL parser uses yacc and a home-brewed laxer, sql_lex.cc. MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HPUX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, Tru64 and Microsoft Windows. A port of MySQL to OpenVMS also exists. All major programming languages with language-specific APIs include Libraries for accessing MySQL database. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL based query method also ships with MySQL adapter allowing direct interaction with MySQL database from any web client via structured URLs. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language. CSS is used to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. It is most often used to set the visual style of webpages and user interfaces written in HTML and XHTML.

HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for language for creating webpages and web applications with Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. To build this website and later on apps for this system we will use the above software's to meet our requirements.

Chapter 6

CONCLUSION AND FUTURE ENHANACEMENT

6.1 CONCLUSION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

6.2 FUTURE ENHANCEMENT

. This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the different scheduled examinations information that are currently issued.

Well I and my team members have worked hard in order to present an improved website better than the existing one’s regarding the information about the various activities. Still, we found out that the project can be done in a better way. Primarily, when we request information about a particular schedule it just shows the exam date and platform. So, after getting the information we can get access to the online exam.

The enhancement that we can add the searching option. We can directly search to the particular student details from this site.

BIBLIOGRAPHY

The following books were referred during the analysis and execution phase of the project

Books Referred:

- BEGINNING PHP 5 ---DAVE MERCER
- BLACK BOOK HTML ---WILEY DREAMTECH
- PHP AND MYSQL WEB DEVELOPMENT --- LUKEWELLING, LAURA
- MICROSOFT SQL SERVER-2000 ---RANKIN, PAUL & JENSEN
- SQL SERVER-2000 ---DUSAN PETKOVIC
- PHP IN A NUTSHELL --- PAUL HUDSON

✓ **Websites Referred:**

http://www.youtube.com

http://www.stackoverflow.com

SOURCE CODE

```
<!DOCTYPE html PUBLIC >
<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>Project Worlds || TEST YOUR SKILL </title>
<link  rel="stylesheet" href="css/bootstrap.min.css"/>
 <link  rel="stylesheet" href="css/bootstrap-theme.min.css"/>
 <link rel="stylesheet" href="css/main.css">
 <link  rel="stylesheet" href="css/font.css">
 <script src="js/jquery.js" type="text/javascript"></script>

  <script src="js/bootstrap.min.js"  type="text/javascript"></script>
  <link href='http://fonts.googleapis.com/css?family=Roboto:400,700,300' rel='styl
esheet' type='text/css'>
<?php if(@$_GET['w'])
{echo'<script>alert("'.@$_GET['w'].'");</script>';}
?>
<script>
function validateForm() {var y = document.forms["form"]["name"].value;  var lett
ers = /^[A-Za-
z]+$/;if (y == null || y == "") {alert("Name must be filled out.");return false;}var z
=document.forms["form"]["college"].value;if (z == null || z == "") {alert("college
must be filled out.");return false;}var x = document.forms["form"]["email"].value;
var atpos = x.indexOf("@");
var dotpos = x.lastIndexOf(".");if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.lengt
h) {alert("Not a valid e-
mail address.");return false;}var a = document.forms["form"]["password"].value;if
(a == null || a == ""){alert("Password must be filled out");return false;}if(a.length<
5 || a.length>25){alert("Passwords must be 5 to 25 characters long.");return false;}
var b = document.forms["form"]["cpassword"].value;if (a!=b){alert("Passwords m
ust match.");return false;}}
</script>
</head>
<body>
<div class="header">
<div class="row">
<div class="col-lg-6">
<span class="logo">Test Your Skill</span></div>
<div class="col-md-2 col-md-offset-4">
<a href="#" class="pull-right btn sub1" data-toggle="modal" data-
target="#myModal"><span class="glyphicon glyphicon-log-in" aria-
hidden="true"></span>&nbsp;<span class="title1"><b>Signin</b></span></a>
</div>
```

```
<!--sign in modal start-->
<div class="modal fade" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content title1">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-
title title1"><span style="color:orange">Log In</span></h4>
      </div>
      <div class="modal-body">
        <form class="form-
horizontal" action="login.php?q=index.php" method="POST">
<fieldset>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-3 control-label" for="email"></label>
  <div class="col-md-6">
  <input id="email" name="email" placeholder="Enter your email-
id" class="form-control input-md" type="email">
  </div>
</div>
<!-- Password input-->
<div class="form-group">
  <label class="col-md-3 control-label" for="password"></label>
  <div class="col-md-6">
    <input id="password" name="password" placeholder="Enter your Password" cl
ass="form-control input-md" type="password">

  </div>
</div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
        <button type="submit" class="btn btn-primary">Log in</button>
    </fieldset>
</form>
      </div>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<!--sign in modal closed-->

</div><!--header row closed-->
```

```
</div>
<div class="bg1">
<div class="row">
<div class="col-md-7"></div>
<div class="col-md-4 panel">
<!-- sign in form begins -->
  <form class="form-
horizontal" name="form" action="sign.php?q=account.php" onSubmit="return val
idateForm()" method="POST">
<fieldset>
<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label" for="name"></label>
  <div class="col-md-12">
  <input id="name" name="name" placeholder="Enter your name" class="form-
control input-md" type="text">

  </div>
</div>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label" for="gender"></label>
  <div class="col-md-12">
    <select id="gender" name="gender" placeholder="Enter your gender" class="fo
rm-control input-md" >
   <option value="Male">Select Gender</option>
  <option value="M">Male</option>
  <option value="F">Female</option> </select>
  </div>
</div>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label" for="name"></label>
  <div class="col-md-12">
  <input id="college" name="college" placeholder="Enter your college name" clas
s="form-control input-md" type="text">
  </div>
</div>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label title1" for="email"></label>
  <div class="col-md-12">
```

```
    <input id="email" name="email" placeholder="Enter your email-
id" class="form-control input-md" type="email">

  </div>
</div>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label" for="mob"></label>
  <div class="col-md-12">
  <input id="mob" name="mob" placeholder="Enter your mobile number" class="f
orm-control input-md" type="number">

  </div>
</div>
<!-- Text input-->
<div class="form-group">
  <label class="col-md-12 control-label" for="password"></label>
  <div class="col-md-12">
    <input id="password" name="password" placeholder="Enter your password" cl
ass="form-control input-md" type="password">

  </div>
</div>
<div class="form-group">
  <label class="col-md-12control-label" for="cpassword"></label>
  <div class="col-md-12">
    <input id="cpassword" name="cpassword" placeholder="Conform Password" cl
ass="form-control input-md" type="password">

  </div>
</div>
<?php if(@$_GET['q7'])
{ echo'<p style="color:red;font-size:15px;">'.@$_GET['q7'];}?>
<!-- Button -->
<div class="form-group">
  <label class="col-md-12 control-label" for=""></label>
  <div class="col-md-12">
    <input  type="submit" class="sub" value="sign up" class="btn btn-primary"/>
  </div>
</div>
</fieldset>
</form>
</div><!--col-md-6 end-->
</div></div>
</div><!--container end-->
```

```
<!--Footer start-->
<div class="row footer">
<div class="col-md-3 box">
<a href="https://www.facebook.com/deepom.chowdhury" target="_blank">About
us</a>
</div>
<div class="col-md-3 box">
<a href="#" data-toggle="modal" data-target="#login">Admin Login</a></div>
<div class="col-md-3 box">
<a href="#" data-toggle="modal" data-target="#developers">Developers</a>
</div>
<div class="col-md-3 box">
<a href="feedback.php" target="_blank">Feedback</a></div></div>
<!-- Modal For Developers-->
<div class="modal fade title1" id="developers">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-only">Close</span></button>
        <h4 class="modal-title" style="font-
family:'typo' "><span style="color:orange">Developers</span></h4>
      </div>
      <div class="modal-body">
        <p>
    <div class="row">
    <div class="col-md-4">
     <img src="image/as.png" width=100 height=100 alt="" class="img-rounded">
     </div>
     <div class="col-md-5">
    <a  style="color:#202020; font-family:'typo' ; font-
size:18px" title="Find on Facebook">Deepom Chowdhury</a>
    <h4 style="font-family:'typo' ">1EP17CS017</h4>
    <a  style="color:#202020; font-family:'typo' ; font-
size:18px" title="Find on Facebook">Ranvijay Kumar</a>
    <h4 style="font-family:'typo' ">1EP17CS068</h4>
    <a  style="color:#202020; font-family:'typo' ; font-
size:18px" title="Find on Facebook">Santosh E</a>
    <h4 style="font-family:'typo' ">1EP17CS076</h4>
    <h4 style="font-
family:'typo' ">East point college of Engineering and Technology</h4></div></di
v>
    </p>
      </div>
<!--Modal for admin login-->
```

```
   <div class="modal fade" id="login">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-only">Close</span></button>
        <h4 class="modal-title"><span style="color:orange;font-
family:'typo' ">LOGIN</span></h4>
      </div>
      <div class="modal-body title1">
<div class="row">
<div class="col-md-3"></div>
<div class="col-md-6">
<form role="form" method="post" action="admin.php?q=index.php">
<div class="form-group">
<input type="text" name="uname" maxlength="20"  placeholder="Admin user id"
class="form-control"/>
</div>
<div class="form-group">
<input type="password" name="password" maxlength="15" placeholder="Passwo
rd" class="form-control"/>
</div>
<div class="form-group" align="center">
<input type="submit" name="login" value="Login" class="btn btn-primary" />
</div>
</form>
</div><div class="col-md-3"></div></div>
      </div>
      <!--<div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
      </div>-->
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<!--footer end-->

</body>
</html>
```