INTRODUCTION

1.1 OVERVIEW

In most of the Educational Systems Hall Allotment for invigilators during examinations is done manually right now which involves manual intervention each and every time. Paper work is done to assign halls for invigilators. There isn’t any software to automate this process. By this proposed system are purely aiming at developing the software to automate hall allotment for invigilators during exams specifically for our college. This system will provide immediate and accurate results without any flaws.

1.2 PROBLEM STATEMENT

Allotting halls for exams is the most responsible and strategic process with various manual works It is error-prone, time consuming, difficult to compute and the data is duplicated. To overcome these problems, we are using an automated allotment system for computerizing works. It is error-free, easy to compute and no data duplication is possible.

The ESHAI reduces manual work. The ESHAI makes the hall allotting an easy task within minutes.

1.3 EXISTING SYSTEM

We have seen over the years that the process of manual work has been carried out across almost all educational institutions. In the existing system all work is done on paper. It requires a person to process a large amount of data and create the list and is time consuming.

1.3.1 DISADVANTAGE OF EXISTING SYSTEM

- The existing system is maintained in white sheets.
- After a successful preparing in the seating arrangement, a clerical work is carried on the prepared one.
- The Controller of Examination has to prepare for the hall allotment before a week to complete their job successfully.

1.4 PROPOSED SYSTEM

We have proposed an automated hall allotment system which is implemented as a web application. The staff will need to login with the IDs given to them by the ADMIN. Then the ADMIN provides them with the details necessary for the allotment result through the GUI. The result allotment list is generated in the form of mail and displayed.

1.4.1 ADVANTAGE OF THE PROPOSED SYSTEM

- Less man power.
- Attractive GUI interface.
- Option to save and print on the need.
- Faster Response.

REQUIREMENTS

2.1 TOOL REQUIREMENTS

2.1.2 TOOLS USED

Front End	:	HTML5, CSS3
Back End	:	PHP5.4.14, MYSQL 5.6.11
Web Server	:	Apache 2.4.4 Web Server
Operating System	:	Windows XP/7/8.

2.1.2 MINIMUM REQUIREMENTS OF THE SYSTEM

Processor	:	Intel Pentium III
RAM	:	64 MB
HDD	:	40 GB
Secondary Storage	:	1.44 MB FDD, CD-R, CD+RW CD
Monitor	:	15” Color Monitor

2.2 FUNCTIONAL REQUIREMENTS

- It will facilitate in accessing halls by searching their name.
- Function to register the invigilator details such as Id, class and also the class room details
- Function to set user id
- Function to change password

- Function to send request to admin to choose halls for their respective exams.
- Function to view allotted hall
- Function to display the allotted halls with their respective Id’s.
- Function to change password
- Function to add and delete the user
- Function to add the new roll number.

2.3 NON-FUNCTIONAL REQUIREMENTS

2.3.1. Performance

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the admin of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The existing system is completely dependent on the user to perform all the duties.

2.3.2 SAFETY AND SECURITY REQUIREMENTS:

- **User Identification:** The system requires the user to identify himself/herself
- **Login ID**: Any user who uses the system shall have a Logon ID and Password.
- **Modification**: Any modification (insert, delete (or) update) for the Database shall be synchronized and done only by the administrator.
- **Administrator Rights**: Administrators shall be able to view and modify all information.

MODULES DESCRIPTION

3.1 CLIENT INTERFACE MODULE

In this module we deal with login interface and retrieval of data from database in the server. Accordingly, depending on whether the teacher is authenticated, the interface begins. After the authentication faculty profile is displayed where the faculty can choose the semester course details and the sessions based on which the respective student database is retrieved from database.

ADMIN FEATURES:

- Admin have their respective user id & password for authenticated login
- Admin can add new invigilators or modify existing ones.

FACULTY FEATURES:

- Faculty have their respective user id & password for secured login
- Faculty can create and view the allotments.

Pre-Condition:

- The Admin/User should be authenticated
- They should have their user id and password individually

Post-Condition:

- Login is successful.

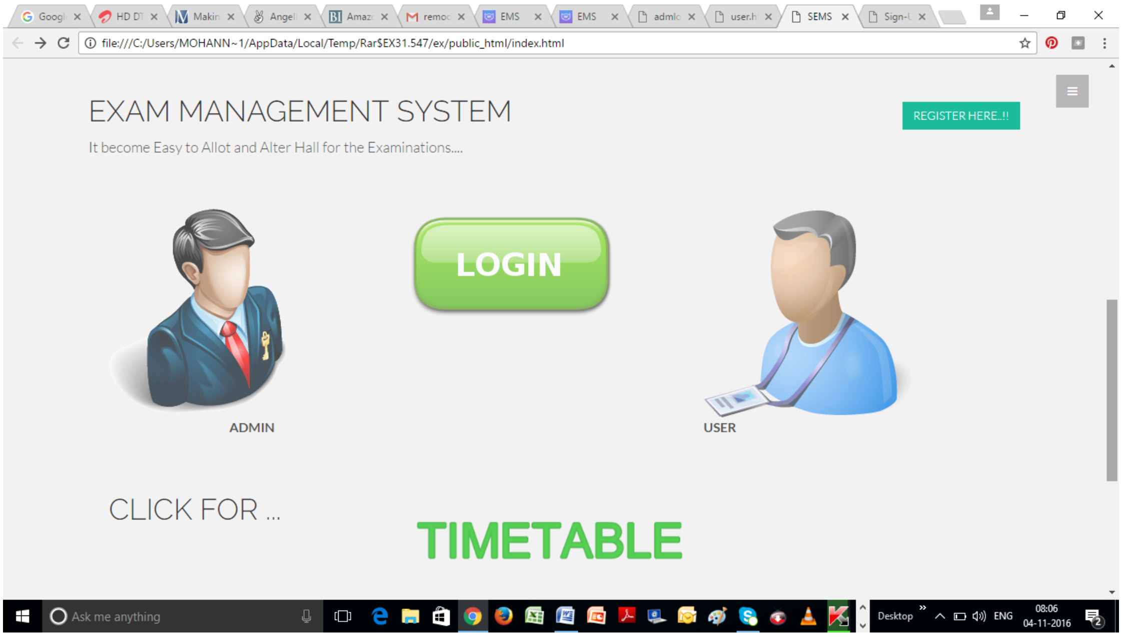
Generating Allotted List:

As soon as the staff logs in, he/she is sent an e-mail to their particular mail address. It is the responsibility of ADMIN to select the classes for the invigilators.

DESIGN

4.1 USER INTERFACE MODEL

LOGIN PAGE



file:///C:/Users/MOHANN~1/AppData/Local/Temp/Rar$EX31.547/ex/public_html/index.html

EXAM MANAGEMENT SYSTEM

It become Easy to Allot and Alter Hall for the Examinations....

REGISTER HERE..!!

LOGIN

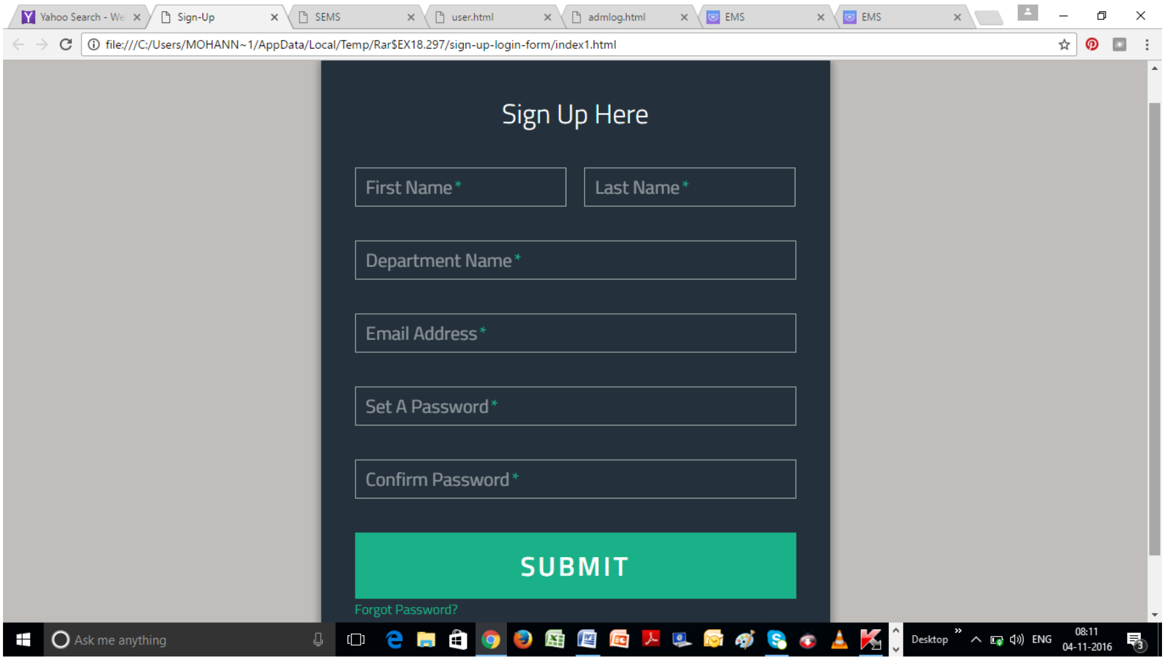
ADMIN

USER

CLICK FOR ...

TIMETABLE

SIGNUP PAGE



file:///C:/Users/MOHANN~1/AppData/Local/Temp/Rar$EX18.297/sign-up-login-form/index1.html

Sign Up Here

First Name *

Last Name *

Department Name *

Email Address *

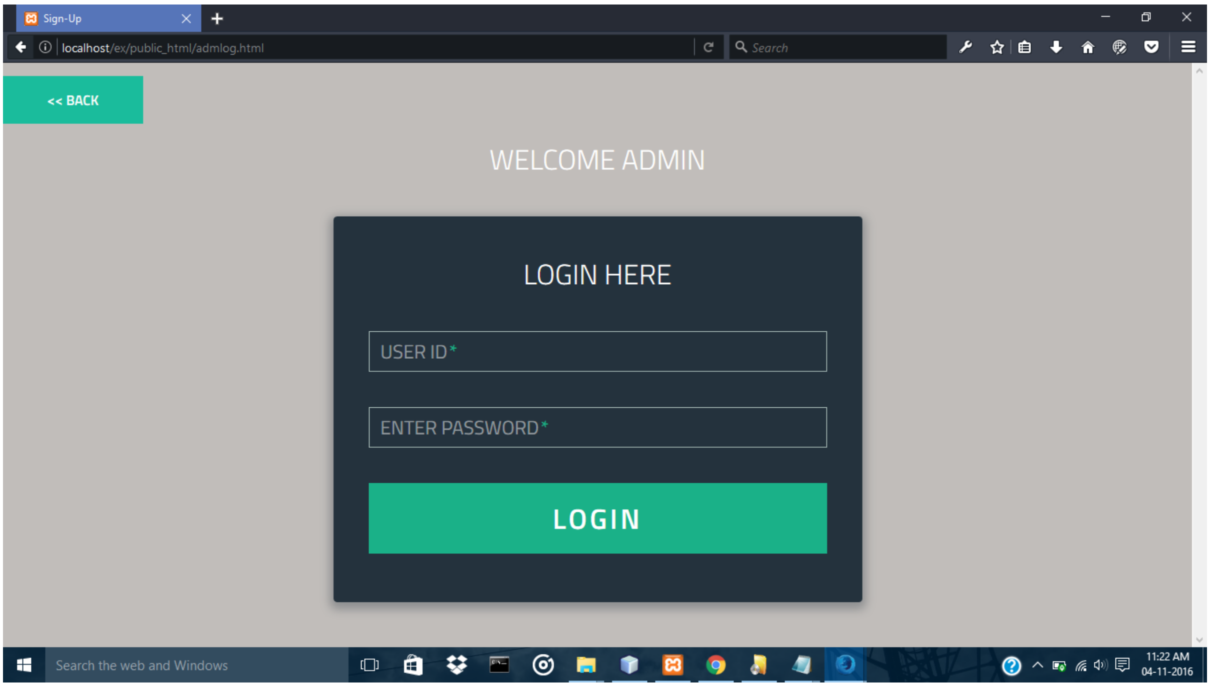
Set A Password *

Confirm Password *

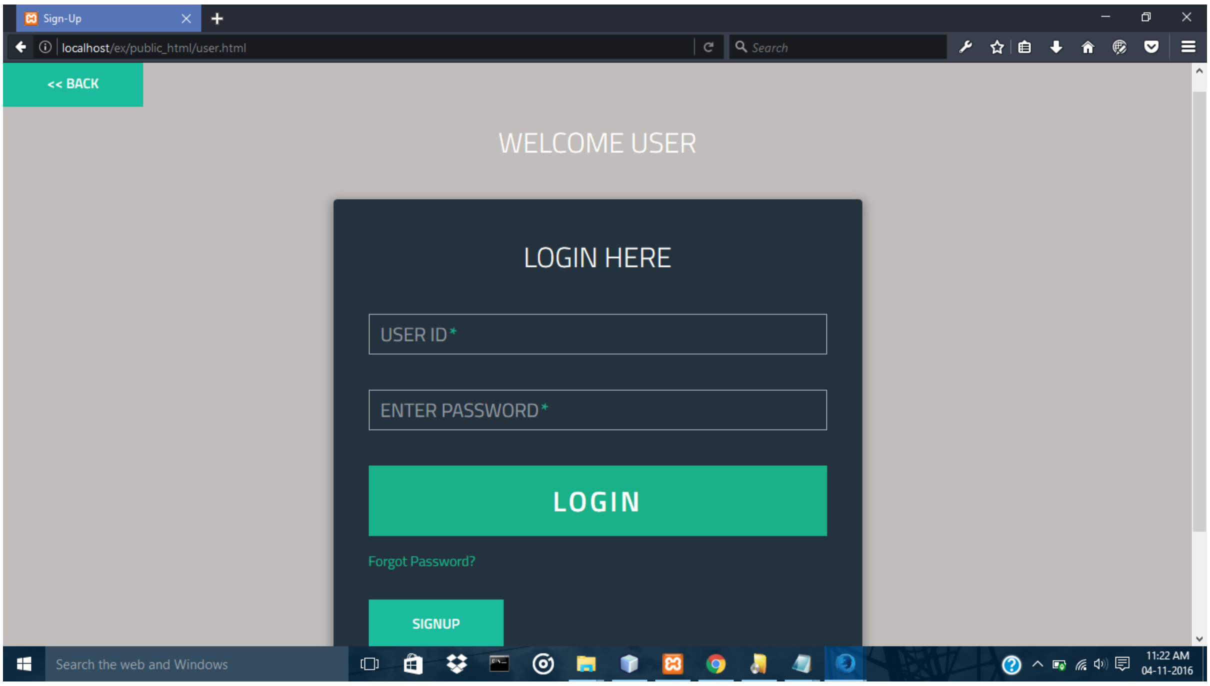
SUBMIT

Forgot Password?

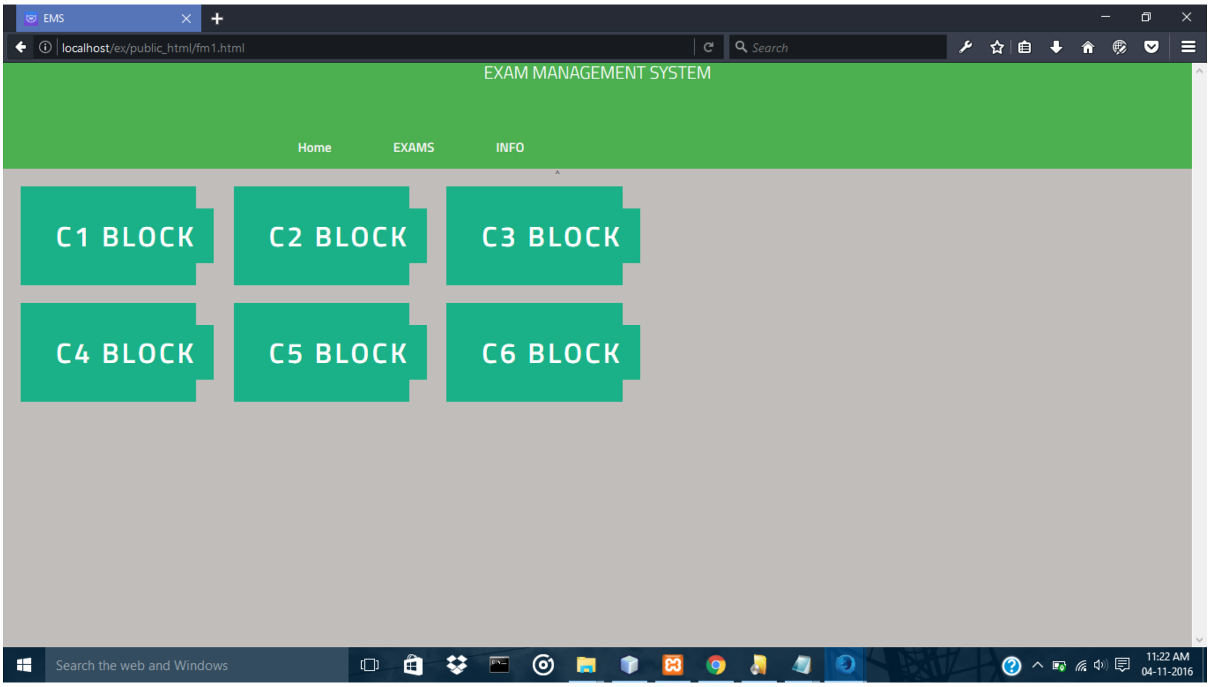
ADMIN PAGE



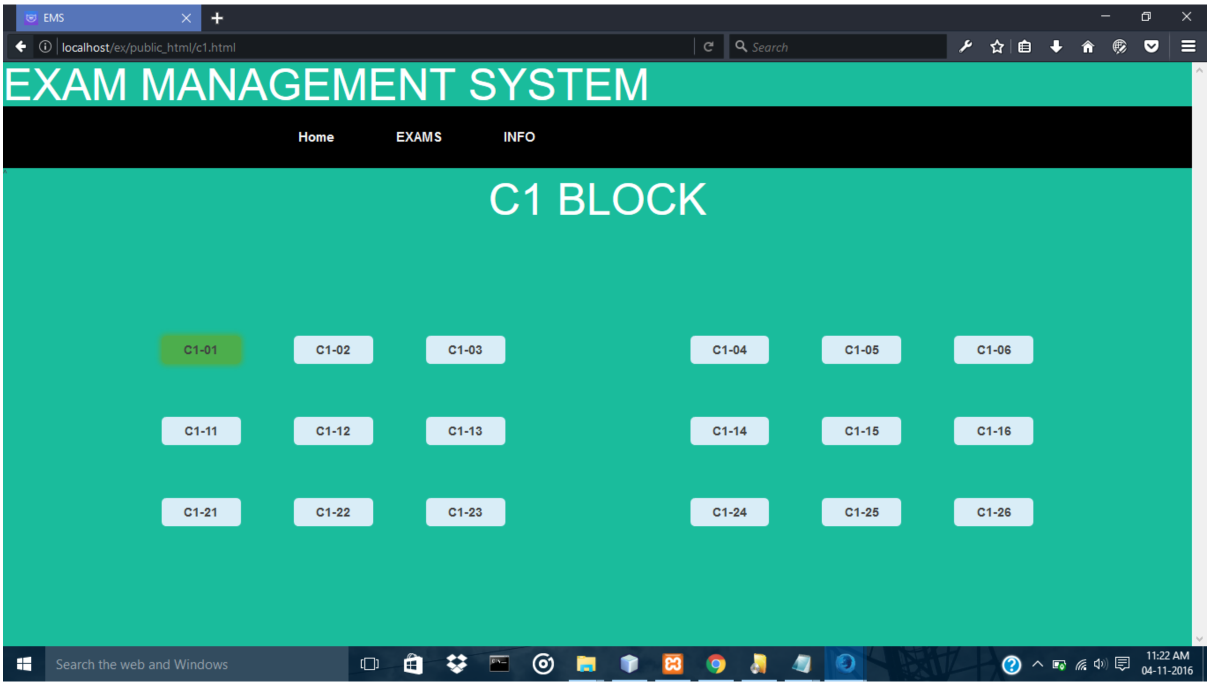
USER PAGE



BLOCK ALLOTMENT



SELECTION OF HALLS



IMPLEMENTATION

5.1 FRONT END:

- HTML 5
- CSS 3
- JAVACRIPT

FEATURES:

HTML5 is a core technology mark-up language of the Internet used for structuring and presenting content for the World Wide Web. As of October 2014 this is the final and complete fifth revision of the HTML standard of the World (W3C). The previous version, HTML 4, was standardised in 1997.

Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

5.1.1 FRONT END CODE:

CODE FOR HOME PAGE:

```
<nav class="menu" id="theMenu">
      <div class="menu-wrap">
        <h1 class="logo"><a href="index.html#home">LINK</a></h1>
        <i class="fa fa-arrow-right menu-close"></i>
        <a href="index.html">Home</a>
        <a href="about.html">About</a>
        <a href="#contact">Contact</a>
        <a href="#"><i class="fa fa-facebook"></i></a>
        <a href="#"><i class="fa fa-twitter"></i></a>
        <a href="#"><i class="fa fa-dribbble"></i></a>
        <a href="#"><i class="fa fa-envelope"></i></a>
      </div>
      <!-- Menu button -->
      <div id="menuToggle"><i class="fa fa-bars"></i></div>
    </nav>
    <!-- MAIN IMAGE SECTION -->
    <div id="headerwrap">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 col-lg-offset-2">

            <h1>HALL ALLOTMENT SYSTEM</h1>
            <div class="spacer"></div>
            <i class="fa fa-angle-down"></i>
          </div>
        </div><!-- row -->
      </div><!-- /container -->
    </div><!-- /headerwrap -->
```

```
        <!-- WELCOME SECTION -->
        <div class="container">
            <div class="row mt">
                <div class="col-lg-8">
                    <h1>HALL ALLOTMENT SYSTEM</h1>
                    <p>It become Easy to Allot and Alter Hall for the Examinations....</p>
                </div>
                <div class="col-lg-4">
                    <p class="pull-right"><br><a href="signup.html"><button type="button" class="btn btn-green">REGISTER HERE..!!</button></a></p>
                </div>
            </div><!-- /row -->
        </div><!-- /.container -->



        <div id="portfolio">
            <div class="container"
                 <div class="row mt">
                    <ul class="grid effect-2" id="grid">
                        <li><a href="admlog.html"><img src="assets/img/admin.png"></a><center><strong>ADMIN</strong></center></li>
                        <li><img src="assets/img/login.png"></a></li>
                        <li><a href="user.html"><img src="assets/img/user.png"></a><strong>USER</strong></li>
                        <li><br></li>
                        <li><br></li>
                        <li><br></li>
                         <li><br></li><li><br></li><li><br></li><li><br></li>
                         <li><h1></h1></li>
                         <li><br></li>
                         <li><br></li> <li><br></li>
```

```
                  <li></li>
                <li><br></li> <li><br></li> <li><br></li> <li><br></li>
      <li><br></li>
                    <li><br></li> <li><br></li> <li><br></li> <li><br></li>
<li><br></li>
                    <li></li>
                </ul>


            </div>


        </div>
    </div>


    <!-- CALL TO ACTION -->
    <div id="call">
        <div class="container">
            <div class="row">
                <h3>USER ASSISTANCE</h3>
                <div class="col-lg-8 col-lg-offset-2">
                    <p>We are here to help you.</p>
                    <p><button type="button" class="btn btn-green btn-lg">Call To
Action</button></p>
                </div>
            </div><!-- row -->
        </div><!-- container -->
    </div><!-- Call to action -->


    <script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
    <script src="assets/js/main.js"></script>
    <script src="assets/js/masonry.pkgd.min.js"></script>
    <script src="assets/js/imagesloaded.js"></script>
```

```
<script src="assets/js/classie.js"></script>
<script src="assets/js/AnimOnScroll.js"></script>
<script>
    new AnimOnScroll(document.getElementById('grid'), {
        minDuration: 0.4,
        maxDuration: 0.7,
        viewportFactor: 0.2
    });
</script>
```

CODE FOR LOGIN PAGE:

```
<style>
       button {
            background-color: #1abc9c;
            color: white;
            padding: 14px 50px;
            margin: 15px 0;
            border: none;
            cursor: pointer;
            width: 100%;
        }
  </style>

  <body>
 <a href="index.html"><button style="width:auto"><strong><<
BACK</strong></button></a>
 <h1>WELCOME ADMIN </h1>
 <div class="form">
      </ul>

      <div class="tab-content">
        <div id="signup">
          <h1>LOGIN HERE</h1>

          <form action="/" method="post">

          <div class="top-row">
          </div>
             <div class="field-wrap">
            <label>
              USER ID<span class="req">*</span>
            </label>
            <input type="text"required autocomplete="off"/>
          </div>

          <div class="field-wrap">
            <label>
              ENTER PASSWORD<span class="req">*</span>
            </label>
            <input type="password"required autocomplete="off"/>          </div>
       <a href="main.html"> <button type="submit" class="button button-block"/>LOGIN</button></a>

     </form>

    </div>

    <div id="login">
     <h1>Welcome Back!</h1>

     <form action="/" method="post">

      <div class="field-wrap">
      <label>
       Email Address<span class="req">*</span>
      </label>
      <input type="email"required autocomplete="off"/>
     </div>

     <div class="field-wrap">
      <label>
       Password<span class="req">*</span>
      </label>
      <input type="password"required autocomplete="off"/>
     </div>

     <p class="forgot"><a href="#">Forgot Password?</a></p>

     <button class="button button-block"/>Log In</button>


     </form>

    </div>

   </div><!-- tab-content -->

</div> <!-- /form -->
  <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

    <script src="js/signup_index.js"></script>
```

CODE FOR SIGNUP PAGE:

```
 <link rel="stylesheet" href="css/signup_normalize.css">
        <link rel="stylesheet" href="css/signup_style.css">
  </head>
  <body>
    <div class="form">
      </ul>
      <div class="tab-content">
        <div id="signup">
          <h1>Sign Up Here</h1>
          <form action="./endpoints/signup.php" method="post">
          <div class="top-row">
            <div class="field-wrap">
              <label>
                First Name<span class="req">*</span>
              </label>
              <input type="text" required autocomplete="off" name= "firstname"/>
            </div>
            <div class="field-wrap">
              <label>
                Last Name<span class="req">*</span>
              </label>
              <input type="text"required autocomplete="off" name= "lastname"/>
            </div>
```

```
</div>
     <div class="field-wrap">
  <label>
    Department Name<span class="req">*</span>
  </label>
  <input type="text"required autocomplete="off" name="dep"/>
</div>
<div class="field-wrap">
  <label>
    Email Address<span class="req">*</span>
  </label>
  <input type="email"required autocomplete="off" name="email"/>
</div>
<div class="field-wrap">
  <label>
    Set A Password<span class="req">*</span>
  </label>
  <input type="password"required autocomplete="off" name="pass"/>
</div>
  <!--  <div class="field-wrap">
  <label>
    Confirm Password<span class="req">*</span>
  </label>
  <input type="password"required autocomplete="off"/>
  </div>-->
  <button type="submit" class="button button-block"/>Submit</button>
  </form>
</div>
<div id="login">
  <h1>Welcome Back!</h1>
  <form action="./endpoints/signup.php" method="post">
    <div class="field-wrap">
    <label>
      Email Address<span class="req">*</span>
    </label>
    <input type="email"required autocomplete="off"/>
  </div>
  <div class="field-wrap">
    <label>
      Password<span class="req">*</span>
    </label>
    <input type="password"required autocomplete="off"/>
  </div>
  <p class="forgot"><a href="#">Forgot Password?</a></p>
  <button class="button button-block"/>Log In</button>
  </form>
</div>
```

USER PAGE

```
  <style>
        button {
             background-color: #1abc9c;
             color: white;
             padding: 14px 50px;
             margin: 15px 0;
             border: none;
             cursor: pointer;
             width: 100%;
         }
   </style>
   <body>
 <a href="index.html"><button style="width:auto"><strong><<
BACK</strong></button></a>
 <h1>WELCOME USER </h1>
 <div class="form">
      </ul>
      <div class="tab-content">
        <div id="signup">
          <h1>LOGIN HERE</h1>
          <form action="/" method="post">
          <div class="top-row">
  <div class="field-wrap">
    <label>
      USER ID<span class="req">*</span>
    </label>
    <input type="text"required autocomplete="off"/>
  </div>
  <div class="field-wrap">
    <label>
      ENTER PASSWORD<span class="req">*</span>
    </label>
    <input type="password"required autocomplete="off"/>
  </div>
  <button type="submit" class="button button-block"/>LOGIN</button>
  </form>
</div>
<div id="login">
  <h1>Welcome Back!</h1>
  <form action="/" method="post">
    <div class="field-wrap">
    <label>
      Email Address<span class="req">*</span>
    </label>
    <input type="email"required autocomplete="off"/>
  </div>


  <div class="field-wrap">
    <label>
      Password<span class="req">*</span>
    </label>
    <input type="password"required autocomplete="off"/>
  </div>


  <p class="forgot"><a href="#">Forgot Password?</a></p>


  <button class="button button-block"/>Log In</button>

  </form>

</div>
```

BLOCK ALLOTMENT PAGE

```
 <body>

            <nav>

            <h1>EXAM MANAGEMENT SYSTEM</h1>

            <div class="wrapper">

                <ul id="menu" class="clearfix">

                    <li><a href="main.html">Home</a></li>

                    <li><a href="#">EXAMS</a>

                        <ul>

                            <li class="purple"><a href="#">FIRST YEAR</a>

                                <ul>

                                    <li><a href="fm1.html">Monthly 1</a></li>

                                    <li><a href="fm2.html">Monthly 2</a></li>

                                    <li><a href="fm3.html">Monthly 3</a></li>

                                </ul>

                            </li>

                            <li class="green"><a href="#">SECOND YEAR</a>

                                <ul>

                                    <li><a href="sm1.html">Monthly 1</a></li>

                                    <li><a href="sm2.html">Monthly 2</a></li>

                                    <li><a href="sm2.html">Monthly 3</a></li>

                                </ul>

                            </li>

                            <li class="aqua"><a href="#">THIRD YEAR</a>
            <ul>
                <li><a href="tm1.html">Monthly 1</a></li>
                <li><a href="tm2.html">Monthly 2</a></li>
                <li><a href="tm3.html">Monthly 3</a></li>
            </ul>
        </li>
        <li class="red"><a href="#">FOURTH YEAR</a>
            <ul>
                <li><a href="ffm1.html">Monthly 1</a></li>
                <li><a href="ffm2.html">Monthly 2</a></li>
                <li><a href="ffm3.html">Monthly 3</a></li>
            </ul>
        </li>
        <li class="blue"><a href="#">SEMESTER</a>
            <ul>
                <li class="red"><a href="#">ODD</a></li>
                <li><a href="#">EVEN</a></li>
            </ul>
        </li>
    </ul>
</li>
<li><a href="#">INFO</a>
    <ul>
        <li><a href="staffdb.html">Staff Databases</a></li>
```

```
              <li><a href="newreg.html">New Registration</a></li>

              <li><a href="announce.html">Announcement</a></li>

              <li><a href="#"></a></li>

            </ul>

          </li>

        </ul>

      </div>

    </nav>

    <br>

    <br><a href="c1.html" class="button"><input type="button" class="button" value="C1 BLOCK"></a>

    <a href="c2.html" class="button"><input type="button" class="button" value="C2 BLOCK"></a>

    <a href="c3.html" class="button"><input type="button" class="button" value="C3 BLOCK"></a><br>

    <a  href="c4.html" class="button"><input type="button" class="button" value="C4 BLOCK"></a>

    <a href="c5.html" class="button"><input type="button" class="button" value="C5 BLOCK"></a>

    <a href="c6.html" class="button"><input type="button" class="button" value="C6 BLOCK"></a>

    <script type="text/javascript">



      $(function () {

        $('a[href="#"]').on('click', function (e) {

          e.preventDefault();
```

```
                });
                $('#menu > li').on('mouseover', function (e) {
                    $(this).find("ul:first").show();
                    $(this).find('> a').addClass('active');
                }).on('mouseout', function (e) {
                    $(this).find("ul:first").hide();
                    $(this).find('> a').removeClass('active');
                });
                $('#menu li li').on('mouseover', function (e) {
                    if ($(this).has('ul').length) {
                        $(this).parent().addClass('expanded');
                    }
                    $('ul:first', this).parent().find('> a').addClass('active');
                    $('ul:first', this).show();
                }).on('mouseout', function (e) {
                    $(this).parent().removeClass('expanded');
                    $('ul:first', this).parent().find('> a').removeClass('active');
                    $('ul:first', this).hide();
                });
            });
        </script>
<body>
```

HALL ALLOCATION PAGE

```
  <style>
body {background-color: #1abc9c}
  h1 {background-color: #1abc9c;}
  nav{background-color: black}
      </style>
    </head>
    <body>
            <nav>
            <h1>EXAM MANAGEMENT SYSTEM</h1>
            <div class="wrapper">
                <ul id="menu" class="clearfix">
                    <li><a href="main.html">Home</a></li>
                    <li><a href="#">EXAMS</a>
                        <ul>
                            <li class="purple"><a href="#">FIRST YEAR</a>
                                <ul>
                                    <li><a href="fm1.html">Monthly 1</a></li>
                                    <li><a href="fm2.html">Monthly 2</a></li>
                                    <li><a href="fm3.html">Monthly 3</a></li>
                                </ul>
                            </li>
                            <li class="green"><a href="#">SECOND YEAR</a>
                                <ul>
                                    <li><a href="sm1.html">Monthly 1</a></li>
```

```
        <li><a href="sm2.html">Monthly 2</a></li>
        <li><a href="sm2.html">Monthly 3</a></li>
    </ul>
</li>
<li class="aqua"><a href="#">THIRD YEAR</a>
    <ul>
        <li><a href="tm1.html">Monthly 1</a></li>
        <li><a href="tm2.html">Monthly 2</a></li>
        <li><a href="tm3.html">Monthly 3</a></li>
    </ul>
</li>
<li class="red"><a href="#">FOURTH YEAR</a>
    <ul>
        <li><a href="ffm1.html">Monthly 1</a></li>
        <li><a href="ffm2.html">Monthly 2</a></li>
        <li><a href="ffm3.html">Monthly 3</a></li>
    </ul>
</li>
<li class="blue"><a href="#">SEMESTER</a>
    <ul>
        <li class="red"><a href="#">ODD</a></li>
        <li><a href="#">EVEN</a></li>
    </ul>
</li>
```

```
                </ul>
            </li>
            <li><a href="#">INFO</a>
                <ul>
                    <li><a href="staffdb.html">Staff Databases</a></li>
                    <li><a href="newreg.html">New Registration</a></li>
                    <li><a href="announce.html">Announcement</a></li>
                    <li><a href="#"></a></li>
                </ul>
        </li>
        </ul>
    </div>
</nav>
<div class="plane">
    <div class="cockpit">
        <h1>C2 BLOCK</h1>
    </div>
    <li class="row row--1">
        <ol class="seats" type="A">

            <li class="seat">
                <input type="checkbox" id="1A" />
                <label for="1A">C2-01</label>
            </li>
            <li  class="seat">
                <input type="checkbox" id="1B" />
                <label for="1B">C2-02</label>
            </li>
            <li class="seat">
                <input type="checkbox" id="1C" />
                <label for="1C">C2-03</label>
            </li>
            <li class="seat">
                <input type="checkbox" id="1D" />
                <label for="1D">C2-04</label>
            </li>
            <li class="seat">
                <input type="checkbox" id="1E" />
                <label for="1E">C2-05</label>
            </li>
            <li class="seat">
                <input type="checkbox" id="1F" />
                <label for="1F">C2-06</label>
            </li>
        </ol>
    </li>
    <li class="row row--2">
        <ol class="seats" type="A">
```

```
<li class="seat">
    <input type="checkbox" id="2A" />
    <label for="2A">C2-11</label>
</li>
<li class="seat">
    <input type="checkbox" id="2B" />
    <label for="2B">C2-12</label>
</li>
<li class="seat">
    <input type="checkbox" id="2C" />
    <label for="2C">C2-13</label>
</li>
<li class="seat">
    <input type="checkbox" id="2D" />
    <label for="2D">C2-14</label>
</li>
<li class="seat">
    <input type="checkbox" id="2E" />
    <label for="2E">C2-15</label>
</li>
<li class="seat">
    <input type="checkbox" id="2F" />
    <label for="2F">C2-16</label>
</li>
```

```
    </ol>
</li>
<li class="row row--3">
    <ol class="seats" type="A">
        <li class="seat">
            <input type="checkbox" id="3A" />
            <label for="3A">C2-21</label>
        </li>
        <li class="seat">
            <input type="checkbox" id="3B" />
            <label for="3B">C2-22</label>
        </li>
        <li class="seat">
            <input type="checkbox" id="3C" />
            <label for="3C">C2-23</label>
        </li>
        <li class="seat">
            <input type="checkbox" id="3D" />
            <label for="3D">C2-24</label>
        </li>
        <li class="seat">
            <input type="checkbox" id="3E" />
            <label for="3E">C2-25</label>
        </li>
```

```
                <li class="seat">
                    <input type="checkbox" id="3F" />
                    <label for="3F">C2-26</label>
                </li>
            </ol>
        </li>
    </div>
    <script type="text/javascript">
        $(function () {
            $('a[href="#"]').on('click', function (e) {
                e.preventDefault();
            });
            $('#menu > li').on('mouseover', function (e) {
                $(this).find("ul:first").show();
                $(this).find('> a').addClass('active');
            }).on('mouseout', function (e) {
                $(this).find("ul:first").hide();
                $(this).find('> a').removeClass('active');
            });
            $('#menu li li').on('mouseover', function (e) {
                if ($(this).has('ul').length) {
                    $(this).parent().addClass('expanded');
                }
                $('ul:first', this).parent().find('> a').addClass('active');
                $('ul:first', this).show();
            }).on('mouseout', function (e) {
                $(this).parent().removeClass('expanded');
                $('ul:first', this).parent().find('> a').removeClass('active');
                $('ul:first', this).hide();
            });
        });
    </script>
```

BACK END:

- PHP, MySQL – Server Side Data processing and report generation

MYSQL:

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases.

Users may use the included command line tools or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records.

The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for.

FEATURES:

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL supports standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

5.3.**CONNECTIVITY:**

EASY PHP:

EasyPHP is like a WAMP software bundle that installs web server services onto the windows computer and allows quick-and-easy development of PHP and MYSQL on a local host. The package includes an Apache server, a MYSQL database, and the PHP extension.

Version:

Easy PHP	:	13.1 VC9
Apache	:	2.4.4
MYSQL	:	5.6.11
PHP	:	5.4.14

FEATURES:

- Multiple Values-Allow multiple values to be chosen.
- Search can be performed anywhere in that file.
- The database and the HTML file is connected using EASY PHP file.
- Easy PHP is a complete package allowing all the power and the flexibility that offers the dynamic language PHP.

CONNECTIVITY PAGE

```
<?php

$server = "localhost";
$username = "root";
$password = "";
$db = "eshai";

$conn = mysqli_connect($server,$username,$password,$db);

/*if($conn){
    echo "connection sucess";

}*/
?>
```

TESTING

INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

6.2 STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behaviour, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress is done by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

WHITE BOX TESTING

To follow the concept of white box testing we have tested each form. We have created independently to verify that Data flow is correct and all conditions are exercised to check their validity. All loops are executed on their boundaries.

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds

CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested so that each path that may generate on a particular condition is traced to uncover any possible errors.

DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them. All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards. For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.
- Each unit has been separately tested by the development team itself and all the input have been validated.

TEST CASE:

A test case, in software engineering, is a set of conditions or variables under ~which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a *test oracle*. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as *test scripts*, particularly when written - when they are usually collected into test suites.

Table6.1 Brief Test Case Description

TEST	**TEST DESCRIPTION**	**INPUT**	**EXPECTED RESULT**	**TEST RESULT**
1	Test the application when login page is opened.		System should display login screen with fields Username and Password with Login button.	**Pass**
2	Test the application when the user clicks the login button after entering the correct details.	Username : admin Password : admin	The system should navigate to specification page.	**Pass**
3	Test the System		The System	**Pass**

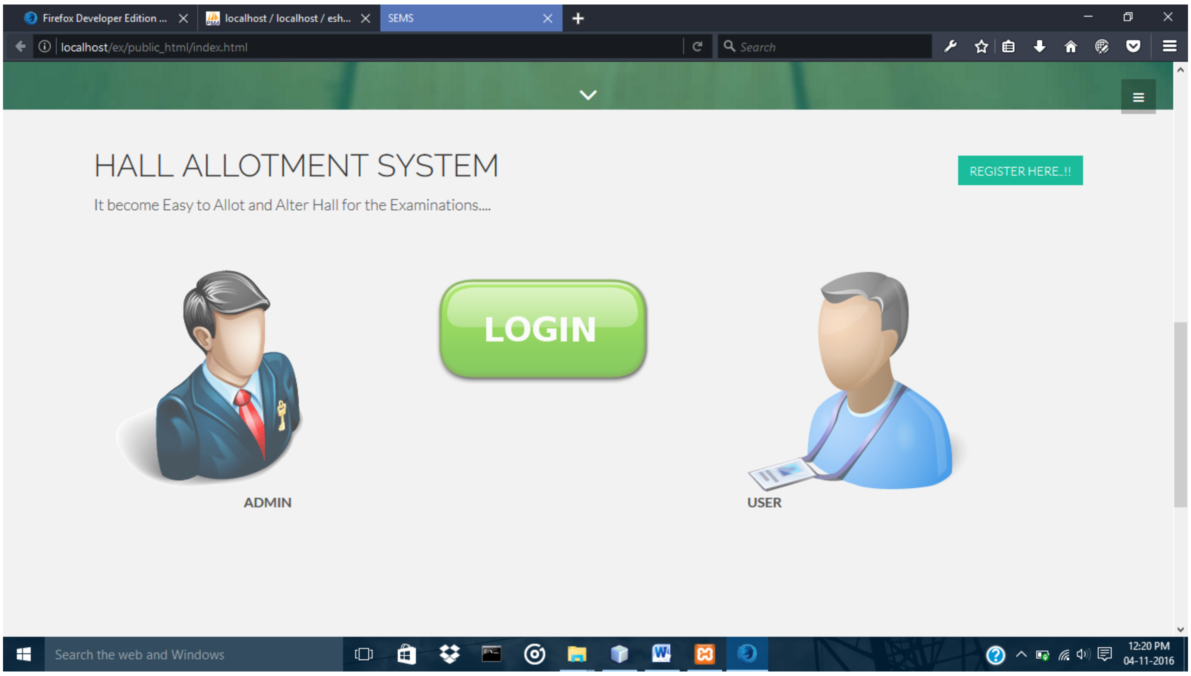
	when the user clicks the login button without entering the correct details.		should show error indicating wrong username and password.	
4	Test the functionality of 'Username' field	admin	The system should accept the input and should not throw any errors.	**Pass**
5	Test the functionality of 'Password' field	admin	The system should accept the input and should not throw any errors.	**Pass**
6	Test the system when the user enters correct specifications in the Specifications field.		The system should generate the correct result.	**Pass**
7	Test the System when the user does not select the 'Exam Name'.		The system must throw an error	**Pass**
8	Test the System when user does not select the STudents for which the exam is conducted.		The system must throw an error	**Pass**

Table6.2 Test Case Results

S NO.	TEST SCENARIO	EXPECTED RESULT	TEST RESULT
1	Username is incorrect Password is correct	Username and Password Invalid	Username and Password Invalid
2	Username is correct Password is incorrect	Username and Password Invalid	Username and Password Invalid
3	Both username and password incorrect	Username and Password Invalid	Username and Password Invalid
4	Both username and password correct	Redirect to next page	Redirect to next page
5	Retrieve Data (if any problem)	Fetch Data	Couldn’t get data
6	Retrieve Data	Fetch Data	Data received at the front end

SCREENSHOTS

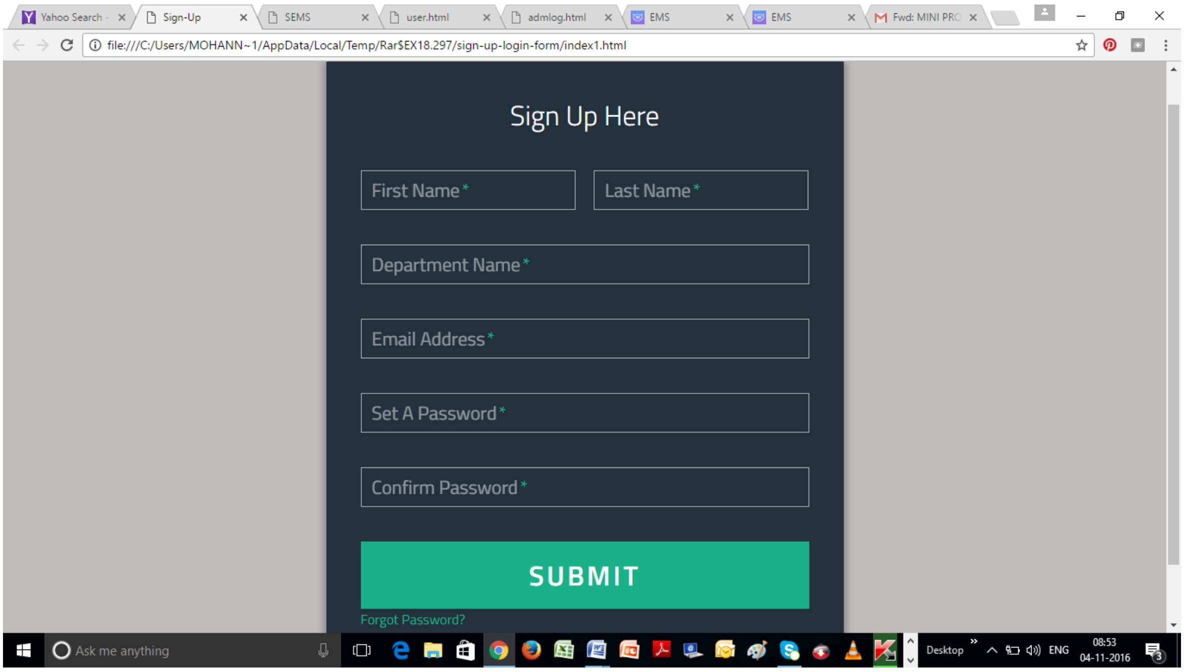
LOGIN:


Firefox Developer Edition ...
localhost / localhost / esh...
SEMS
localhost/ex/public_html/index.html
Search
REGISTER HERE..!!
LOGIN
ADMIN
USER
Search the web and Windows
12:20 PM
04-11-2016

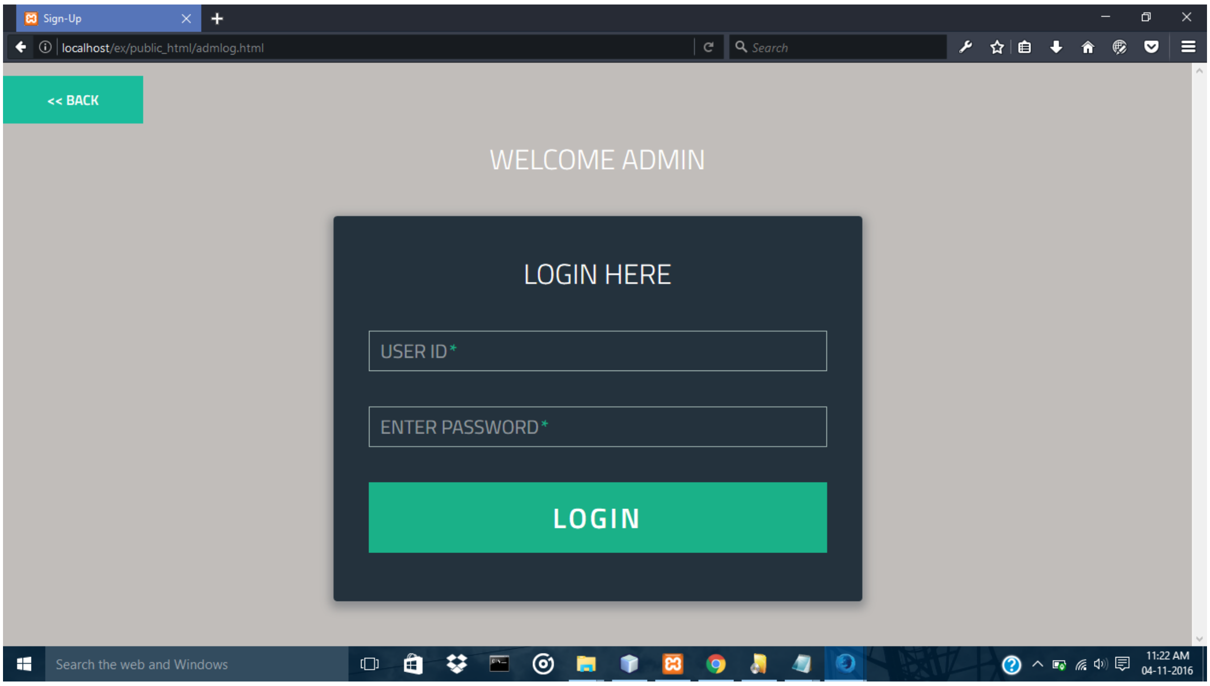
HALL ALLOTMENT SYSTEM

It become Easy to Allot and Alter Hall for the Examinations....

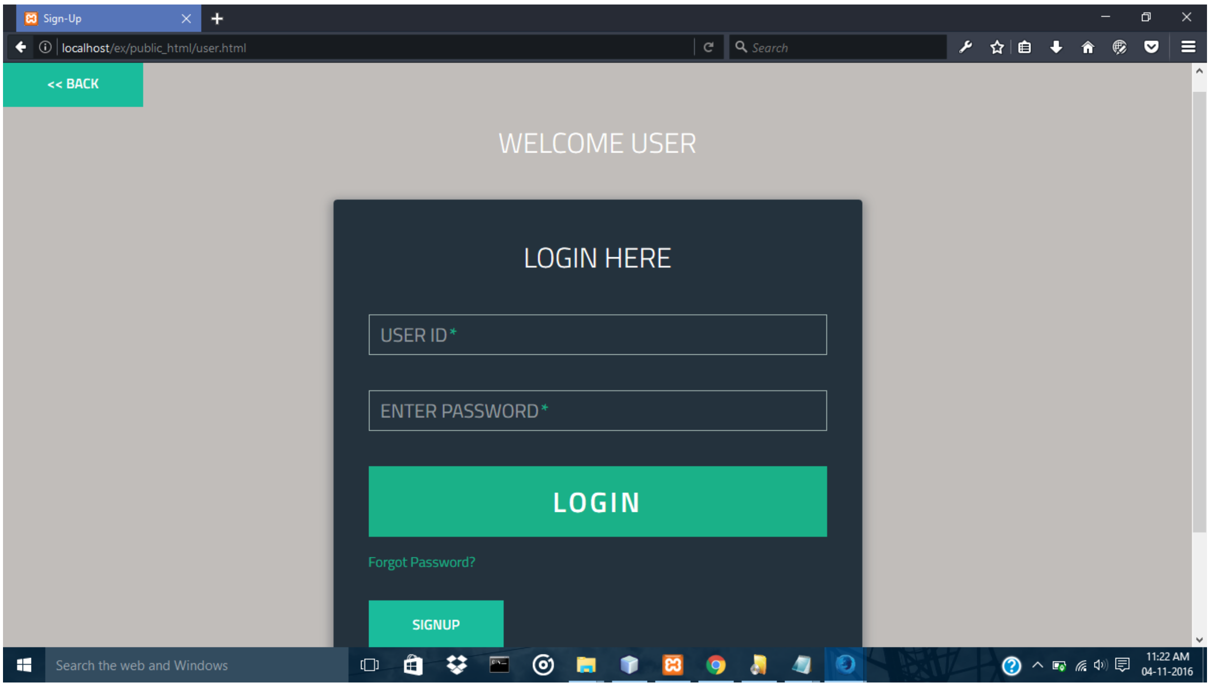
SIGNUP PAGE:



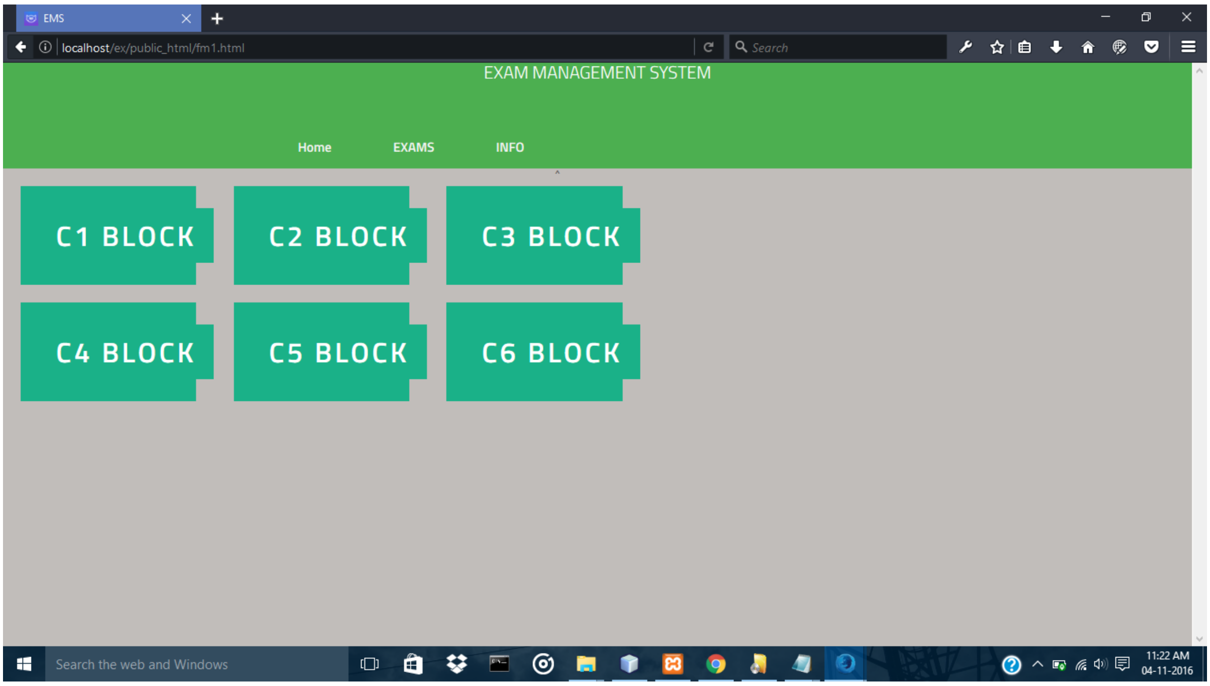
ADMIN REPOSITORY:



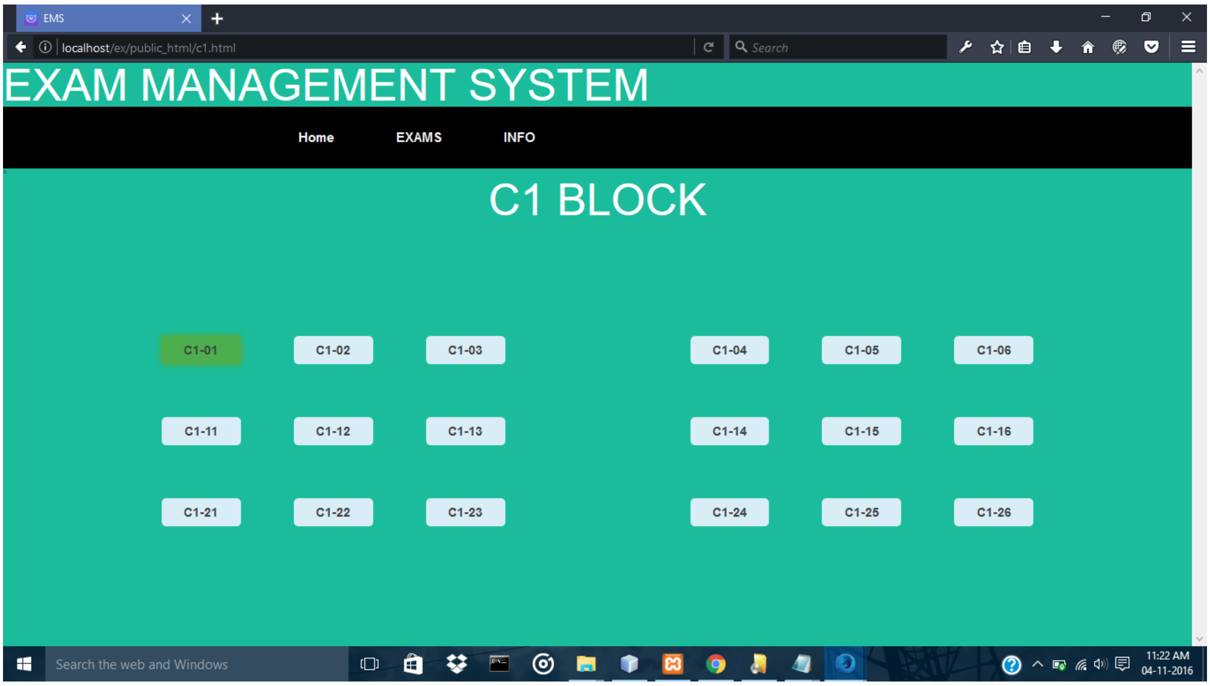
USER REPOSITORY:



BLOCK ALLOTMENT



BLOCK_ALLOCATION



ADDITIONAL FEATURES TO BE ADDED

We would add some more features and take our project forward like

1) Invigilators would be able to view the database of all their colleagues
2) Invigilators would be able to swap in between them matching their requirements
3) There would be available invigilators who would be present and waiting to take up the responsibility if the assigned invigilators cannot take up their work.

CHAPTER 8

REFERENCES

WEBSITES:

1. http://www.w3schools.com/php/
2. http://www.w3schools.com/js/
3. http://www.w3schools.com/css/
4. http://www.w3schools.com/html5/
5. https://pdfcrowd.com/
6. http://www.htmlpdf.com/
7. https://pdfcrowd.com/html-to-pdf-api/
8. http://www.good-tutorials.com/tutorials/php
9. https://learnable.com/home
10. http://www.codecademy.com/en/tracks/php
11. http://www.codecademy.com/en/tracks/javascript
12. http://www.tutorialspoint.com/mysql/
13. http://www.mysqltutorial.org/
14. Shttp://stackoverflow.com/questions/18421988/getting-checkbox-values-on-submit/
15. http://www.yourhtmlsource.com/javascript/formvalidation.html
16. http://channel9.msdn.com/Series/Javascript-Fundamentals-Development-for-Absolute-Beginners/