Chapter-1: INTRODUCTION

1.1. PURPOSE OF THIS PROJECT

'Learning management system' has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of teaching institutions or departments of colleges to carry out operations in a smooth and effective manner. The purpose of learning management system is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, so that the valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

1.2. Why Web Application?

Through computers organization, IT Firms, Universities and Businesses etc.. can transact to their clients in a convenient way using advance technologies and specially the Web Application. The Web Application is created for college or any teaching institution functions through this web app "Learning Management System". Admin (Coordinator) can lessen their errors and efforts in every user support processing and transactions and in making reports. It is a Web Application which is developed in HTML, CSS, JSP, Servlet & MySQL. The purpose of this Web Application is to manage the activities of every teaching institutions. Even a person can handle very easily; it means Web Interface is user friendly.

1.3. EXISTING SYSTEM

In the existing system we have already seen there are lots of Learning Management Systems. Some institutions also maintain these systems by using social media applications like WhatsApp or Facebook. In the existing system we have seen some traditional system like classroom notes in physical mode, class schedules (Time-Tables) in physical mode, teachers and student data on a physical notebook record and sometimes notices and updates about classes or exams released through a physical copy mode.

The following are the drawbacks present in the existing system:

- With the traditional system both students and teachers face many issues to maintain those physical mode of records.

- With these pandemic periods most of the colleges are closed for a long period of time. In this crucial time students could not able to get their class records, notes, updates and notices etc. at a single place.

- Also, we have seen some institutions who has maintaining these records online, they could not able to give all the data at one platform.

1.4. PROPOSED SYSTEM

My proposed system provides solution to existing system as follows:

- In my Learning Management System, a student can get all the data for any institution in one platform.

- Also, this system is proposed like there are no outsider student can enter to this system.

- In this system admin of the institution is the key person who can give access to both students and teachers to enter the system.

- Also, admin can assign the subjects to the teacher and maintain the whole records of students and teachers.

- After getting access, students can get all notes, class time tables, which subject has given to which teacher, all updates with an enhanced notification panel etc.

- Also, after getting access, teachers can upload notes and give any class or exam updates etc.

1.5 PROJECT SCOPE

1. This system helps a student/teacher to create a unique account by his/her email. And also, can login to the system by that Id's.

2. Admin is the key person for whole system. He/she can give or withdraw the access of every users.

3. Admin can perform all insert, edit and delete operations of every module.

4. Teachers can upload the notes, announcements and edit or delete them time to time.

5. Students can see the documents, give the queries and submit their evaluations accordingly.

Chapter-2: PROJECT ANALYSIS

2.1. LITRETURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength.

Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

I made this web-based system by survey of these following literatures.

- A Research Paper on "Document-Oriented E-Learning Components"
- A review paper on "e-Learning: technology concepts"
- "E-learning Advocate Project 2006/7 Integrating E-Learning into the English Curriculum at The University of Northampton."

2.2. SOFTWARE REQUIREMENT ANALYSIS

A software requirements analysis is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Product perspective→ The software product is a Web application. The application will be made up of two parts, one administrator who has all the rights and the other user who has limited rights to handle the application. The three users of the system, namely the Coordinator (Admin) and Student and teacher (User) interact with the system in different ways.

Product Functions→ First of all it will authenticate the user whether he is Admin or Users; the unauthorized person can't get access to the application. The Admin will be able to Add, delete, and modify user's details. He can also Add, delete and modify classes and subjects. He can use this application to check the evaluations and no. of materials uploaded in this system.

Safety Requirements→ All the data will be saved to database for safety purpose so there will be no data loss. These data can be accessed only by an authorized person so data theft is also not possible in this application.

Security Requirements→ For preventing unauthorized access to the application, this application have login feature so only granted user can access with defined rights.

2.2.1. Data Gathering

Data collection is the systematic approach to gathering and measuring information from a variety of sources to get a complete and accurate picture of an area of interest. Data collection enables a person or organization to answer relevant questions, evaluate outcomes and make predictions about future probabilities and trends. Accurate data collection is essential to maintaining the integrity of research, making informed business decisions and ensuring quality assurance.

2.2.2. Feasibility Study

Feasibility study means to check whether the project is feasible or not, that means possible or not. Some feasibility study regarding this project is as follows: -

Economic Feasibility→ The project has shown the economic feasibility by the study of the fact that by using this software the increased number of the customers can be given service effectively and efficiently and can save a lot time and saving time means saving money. The cost and benefit analysis have shown that cost that have incurred in developing the project is less than the benefits that the project is going to provide once it is developed, so this project has passed the feasibility test.

Technical Feasibility→ Technical feasibility centers on the existing computer system (Hardware, Software etc.) and to what extent it supports the existing system. As the existing system computer system is viable so there is no matter of technical feasibility that is the system is technically feasible. In this type of feasibility study, it is checked whether there is a need of new hardware/software or not. What are the basic requirements of the project? If there is need then how it can be fulfilled. In this context, this project doesn't need any special hardware or software.

Behavioral Feasibility→ The User also interested in this project, as it will help them to do work with ease and efficiently without complexity, so they supported the development of this project with full enthusiasm. This shows the behavioral feasibility of the project.

Time Feasibility → It is the determination of whether a proposed project can be implemented fully within stipulated time frame. The project was decided to be done in three months and was thought to be feasible

Operational Feasibility→ In this feasibility study it is determined whether there is need of well qualified operator or simple user. Is there need to train the operator or not? This project is supporting the Graphical User Interface; hence operating this project is so simple. Even a person who has a little knowledge of computer can easily handle this well. There is no need of trained operator.

2.2.3. Software Process Model

The Software Process Models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use. Choosing right model for developing of the software product or application is very important. Based on the model the development and testing processes are carried out. A Process Model describes the sequence of phases for the entire lifetime of a product. Therefore, it is sometimes also called Software Life Cycle. This covers everything from the initial commercial idea until the final de-installation or disassembling of the product after its use.

Fig 1: SDLC Model

In order to develop the project "Online Service Management System" we have adopted the Iterative Enhancement Model also known as Incremental Model. This model removes the shortcoming of waterfall model. Since many facts of this system are already known. It is not a new concept and hence no research is required. A working version can be easily created and hence the system can start working. Rest of the functionalities can be implemented in the next iteration and can be delivered later. As the requirement analysis is also not required. It not being a new technology risk involved is also less. So, one need not perform detailed risk analysis. If redevelopment Admin is less than development can be started with a smaller number of people and in next increments others can be involved. As this model combines the advantage of waterfall model and prototyping, clients are always aware of the product being delivered and can always suggest changes and enhancements and can get them implemented. As less amount of customer communication is required one need not apply spiral model in which all types of analysis is done in detail. As the deadline is affordable one need not to for Rapid Application Development model. Iterative enhancement model is useful when less manpower is available for software development and the release deadlines are specified. It is best suited for in house product development, where it is ensured that the user has something to start with. The complete product is divided into releases and the developer delivers the product release by release.

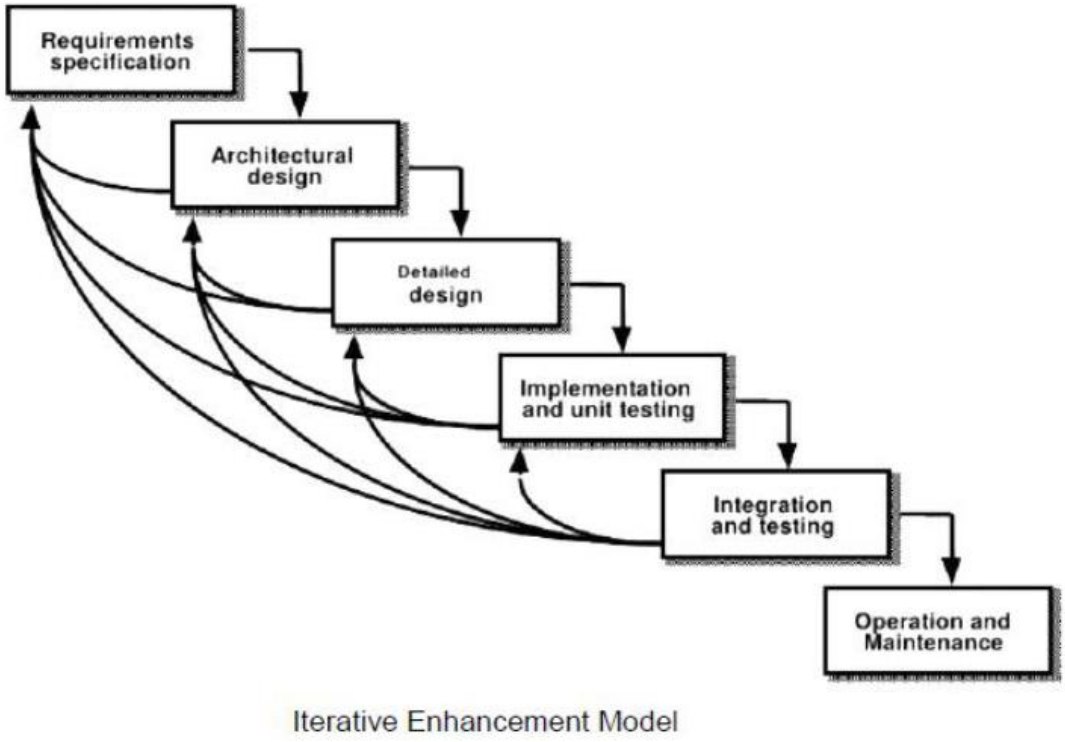


Fig 2: Iterative Enhancement Model

2.3. INPUT & OUTPUT REPRESENTATION

2.3.1. Input of the Project

In order to complete the tasks of the Application and to get output by using this application work, there is need of some input based on the work that is to be carried out by using it. Different kinds of input are required for different purposes.

Input for new Registration→

a. Select the Role (Teacher / Student)

b. Enter the ID

c. User Name

d. User ID (Unique)

e. Password

Input for login in the application →

a. User ID #

b. Password

Input for Classes →

a. Class Name

Input for assign subjects →

a. Select the class

b. Subject name

Input for add lecturers →

a. Lecturer name

b. Current City

c. Select gender

d. Email Id

e. Qualification

Input for assign subjects →

a. Select teacher

b. Select subject

Input for add students →

a. Student name

b. Current city

c. Select Gender

d. Enter email ID

e. Select the class

Input for student →

a. Enter queries

b. Select teacher

c. Enter evaluations

Input for Teachers →

a. Enter category name

b. Upload material

c. Select category

d. Enter Announcements

2.3.2. Output of the Project

The project named "Learning Management System" is being made keeping in mind to solve the activities that are carried out in the teaching institutes. By using this Admin can easily do many things like as:

I. View / edit all classes

II. View / edit added lectures

III. View all Subjects according to classes

IV. View / edit added students

V. View all users

VI. View all materials uploaded by teachers

VII. View profiles and all notifications

VIII. View / edit all categories in teachers' section

IX. View / edit all materials

X. View / edit all Announcements in teachers' section

2.4. Module Description

Welcome/Home Panel →

Home: When the user clicks on this button, it will display the other modules and pages of the website such as Courses, Registration, Login, Contact, and Admin Login. This module will be used to display the brief introduction of the project and will show the title of the project as well as the name of the developer.

Registration / Log in: This is the most important module of the Online Learning Management System which provides a Registration form where user can register themselves. This is user login form. When a user clicks on this link a user login form will be appear where user can enter their user id and password for logging in to the user panel.

Admin Panel →

Dashboard: - This screen displays overview of no of teachers and students etc. and other stuff like Number of materials uploaded.

Add Students: - This screen adds students with their data's by taking as input show all the added students, also has the options to edit or delete.

Add Teachers: - This screen adds teachers with their data's by taking as input show all the added teachers, also has the options to edit or delete.

Add Classes: - This screen adds class with their data's by taking as input show all the added class, also has the options to edit or delete.

Add Subjects: - This screen adds subjects according to the classes. Also, has the option to edit or delete.

Assign Subjects to teachers: - This screen helps to assign subjects to respective teachers according to subjects.

Users Panel →

Teachers: - This screen helps a teacher to show their profiles and all announcements. Also, teachers can create category and add or upload the materials according to category. Also, teacher can add announcements which will be visible to all users.

Students: - This screen helps a student to show their profiles and all announcements. Also, student can view and download all the materials added by teachers to his classes only. Student can give the queries and the evaluation for teachers by entering the needed data.

Change Password: - This screen is visible to all users like admin, teachers and students.

Log out: - This screen is visible to all users like admin, teachers and students. Also, this helps a user to terminate the current session and revert back to login page.

Chapter-3: Project Plan & Approach

3.1. Project Plan & Approach

Phase-1: Requirement Gathering and Analysis

Gathering system requirement and prepare a System Requirement Specification document.

After collect the information it is analyzed that the available resources can fulfil all the requirements. And it is also be examined that what resource will be used.

Phase-2: System Design

Make a detailed analysis of the system and prepare a System Design Document on the basis of SRS.

Phase-3: Prepare UTC/STC for testing the software

Prepare Unit Test Cases document. This document will be used to verify whether the functional requirements of the system have been met.

Phase-4: Develop the software

Develop the planned system

Phase-5: Test the software using the prepared UTC/STC and Rework if needed

Run your software programs using the respective UTC to verify & test the software.

Phase-6: Demonstrate the software to users & Implement it

3.2. PROJECT SCHEDULE

Step #	Time Frame
Phase 1. Requirement Gathering	**12 Days**
Phase 2. System Design	**23 Days**
Phase 3. Develop the software	**35 Days**
Phase 4. Test the software	**18 Days**
Phase 5. Demonstrate the software to users & implement it	**10 Days**

Table 1 – Project Schedule table

Chapter-4: System Requirements

4.1. SOFTWARE REQUIREMENT

- Frontend
    - HTML5
    - CSS3
    - Bootstrap
    - Java Script
- Backend
    - JSP
    - Servlet
    - MySQL

4.2. HARDWARE REQUIREMENT

- i3 Processor or above to this Processor
- Any OS with a Supported Web Browser
- 8GB Ram
- 512GB SSD/HDD
- IDE
    - o Eclipse

4.3. LANGUAGE FUNDAMENTAL

4.3.1. Why using JSP and Servlet

Web is a system of Internet servers that supports formatted documents. The documents are formatted using a markup language called HTML (Hyper Text Markup Language) that supports links to other documents like graphics, audio, and video files etc. Servlet is a server-side Java program module that handles client requests and implements the servlet interface. Servlets can respond to any type of

request, and they are commonly used to extend the applications hosted by web servers.

4.3.2. Why using MySQL

MySQL is the most popular open source relational database management system. It is one of the best RDBMS being used to develop web-based software applications. It is easy to use and fast RDBMS. There are many good reasons which help us to develop website using this RDBMS:

- It is open-source, so available for free.
- It works on many operating system and with many languages including PHP, PERL, C, C++ etc.
- MySQL is customizable.
- MySQL works very friendly to PHP.

MySQL works very quickly and works well even with large data sets.

Chapter-5: System Design

The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science in general. System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

5.1. Use Case Diagram

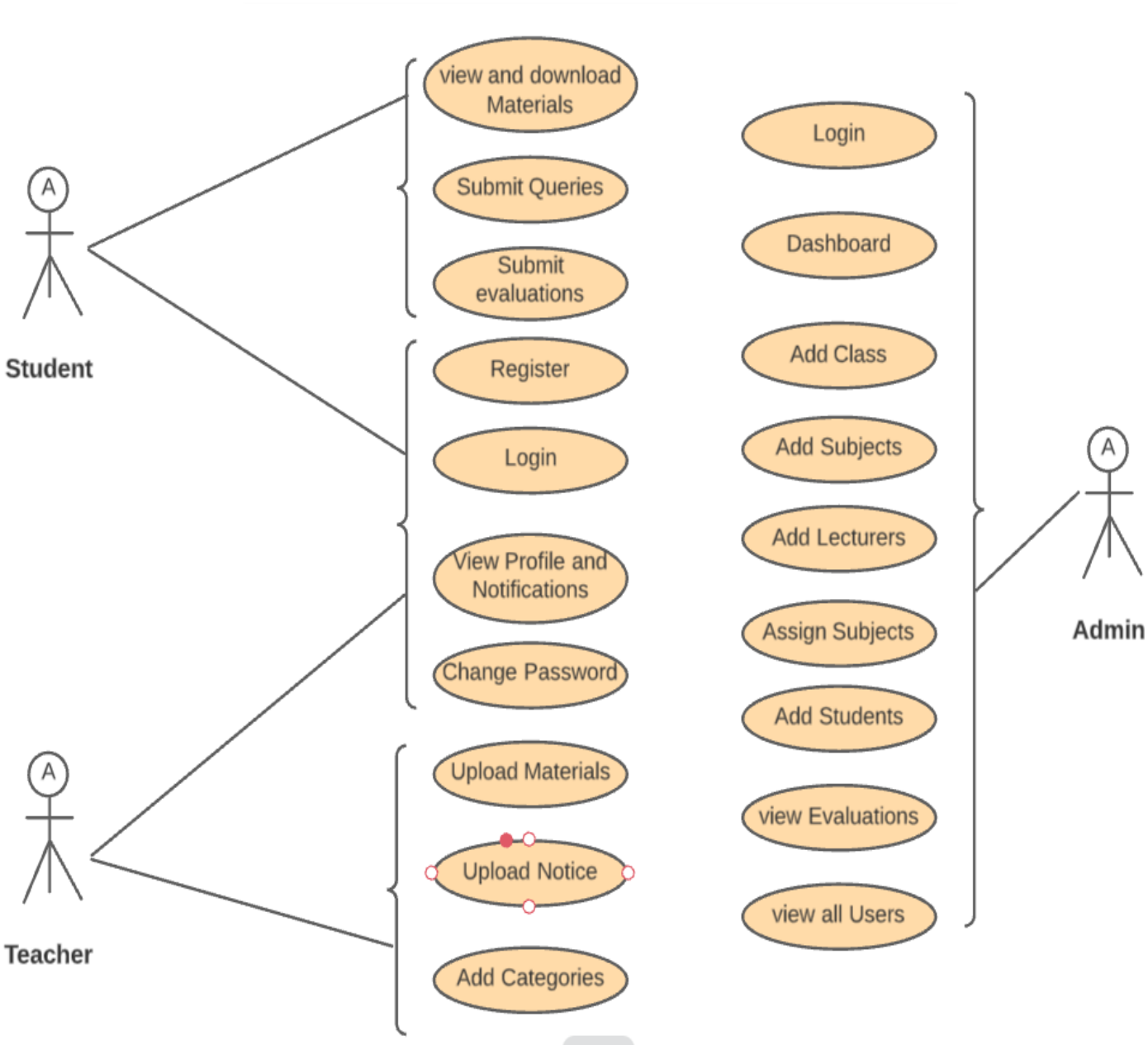


Fig 3: Use Case Diagram

5.2. Activity flow Diagram

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

For Coordinator/Admin →

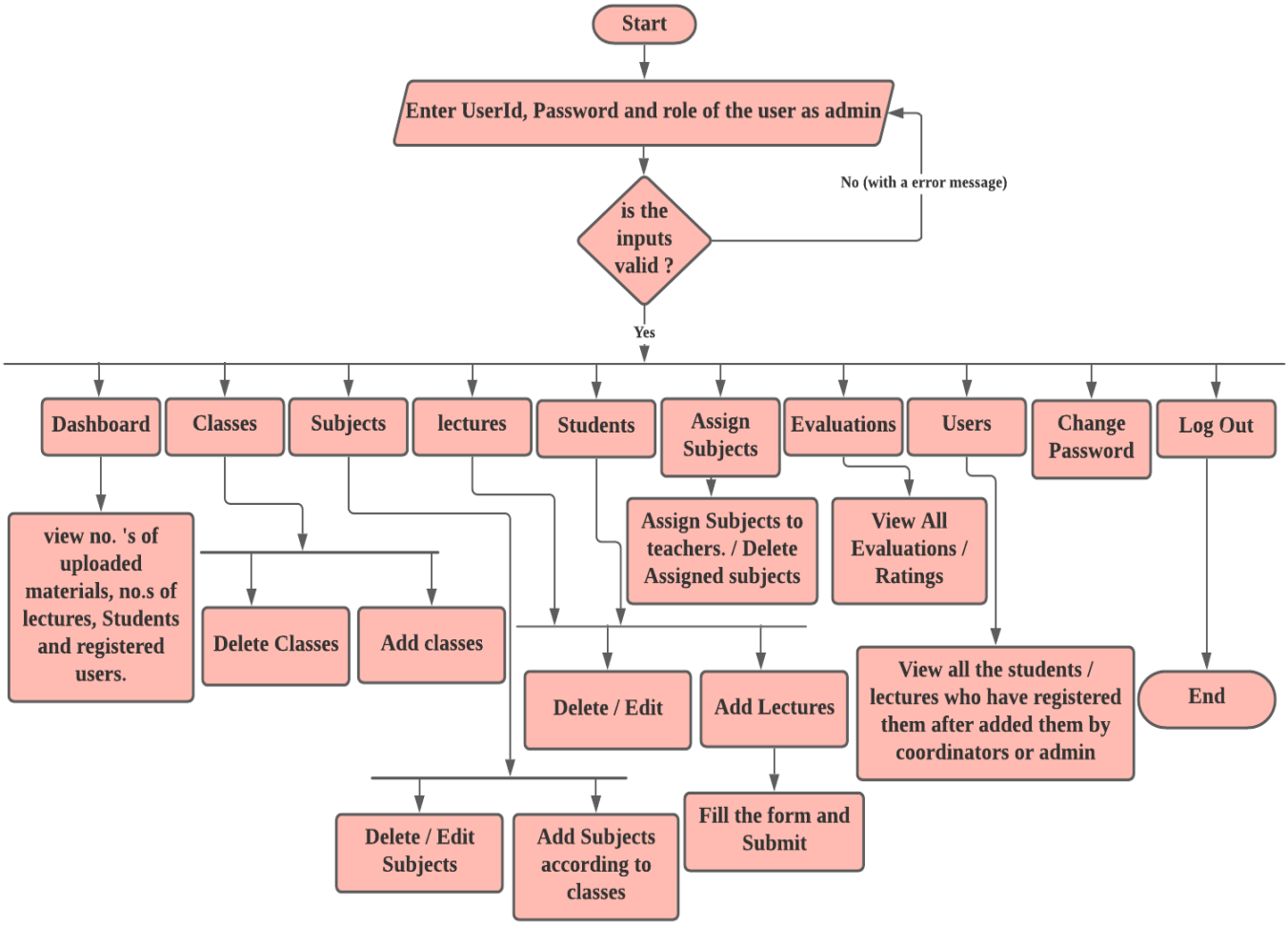


Fig 4: Activity flow diagram for admin

For Teachers →

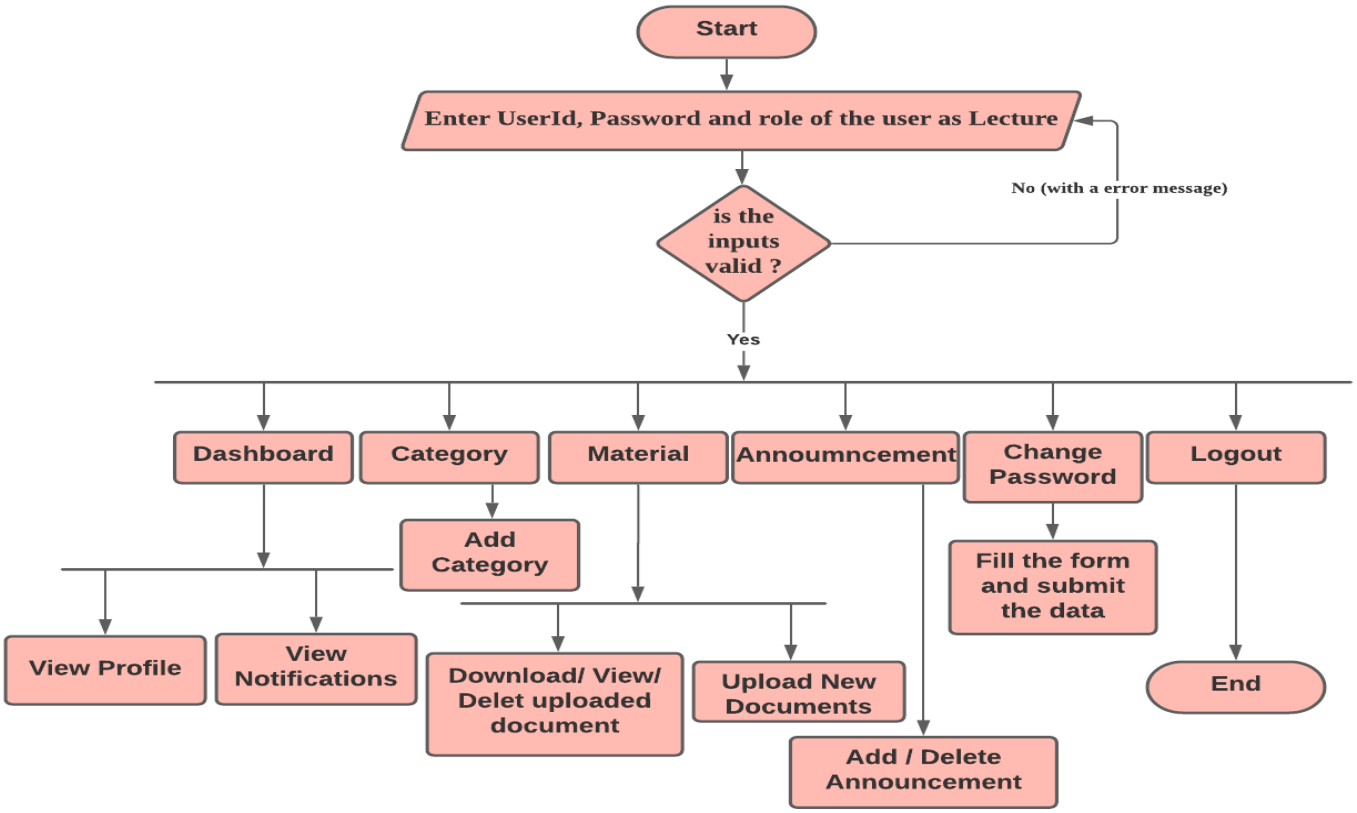


Fig 5: Activity flow diagram for Teachers

For Student →

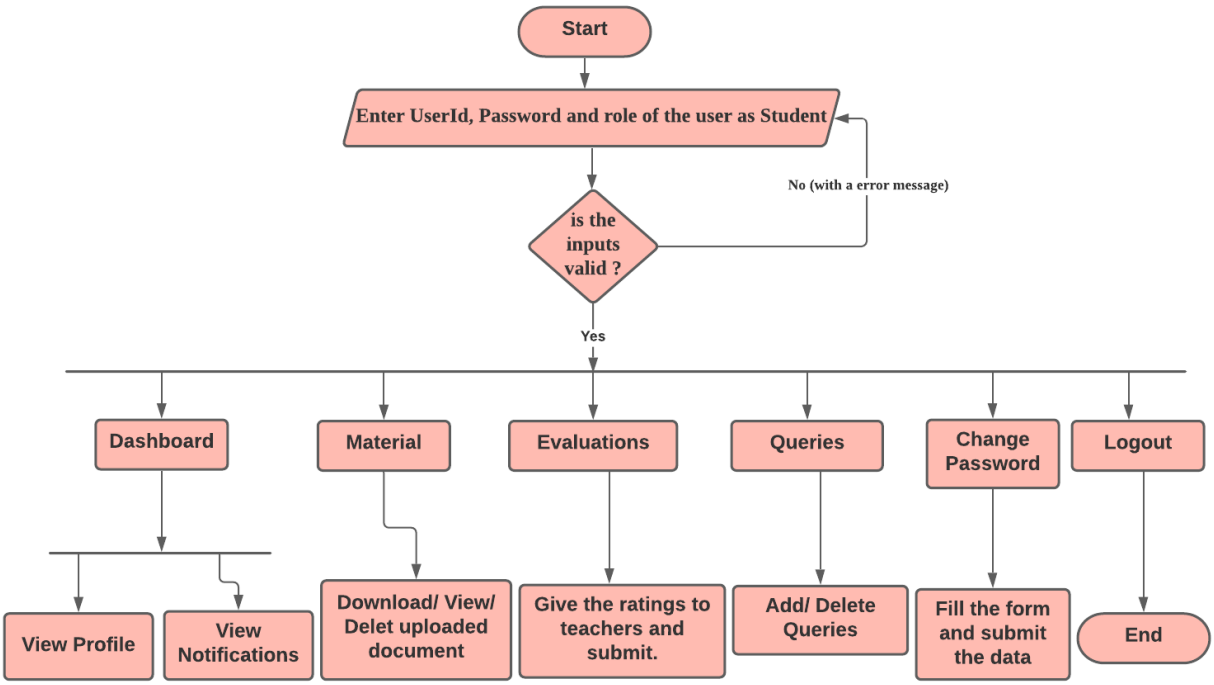


Fig 6: Activity flow diagram for student

5.3. Entity Relationship Diagram (ER-Diagram)

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other.

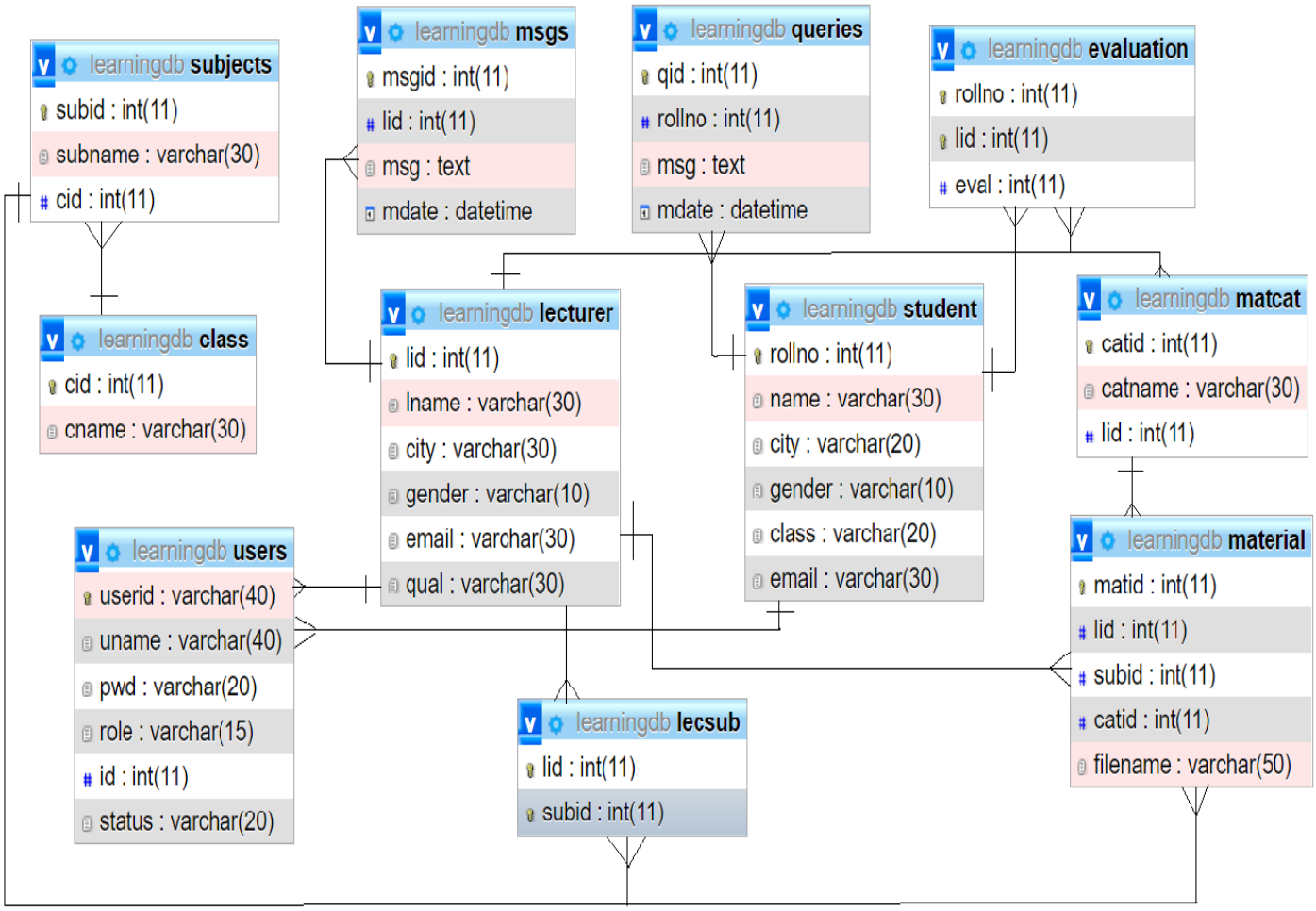


Fig 7: E-R Diagram

Chapter-6: Coding / Code Templates

Log In and signup Page →

```
<jsp:include flush="true" page="header.jsp" />
<script>
    $(function(){
        $("#role").change(function(){
            if(this.value==="student"){
               $("#id").attr({"placeholder":"Enter roll no"});
            }else if(this.value==="lecturer"){
               $("#id").attr({"placeholder":"Enter teacher id"});
            }else{
               $("#id").attr({"placeholder":"First Select role"});
            }
        });
        $("#cpwd").blur(function(){
           if(this.value !== $("#pwd").val()){
               $("#error2").html("Password not match");
               $("#error2").removeClass("d-none");
           }else{
               $("#error2").addClass("d-none");
           }
        });
        $("#id").blur(function(){
           $.ajax({
               url:'Verify',
               data:{'role':$("#role").val(),'id':this.value},
               method:'post',
               success:function(output){
                   if(output==="Invalid"){
                       $("#error").html("Invalid ID given");
                       $("#error").removeClass("d-none");
                   }else{
                        $("#uname").val(output);
                        $("#error").addClass("d-none");
                    }
               }
           });
        });
    });
</script>
<div class="jumbotron p-3 mb-0 text-center text-white rounded-0"
     style="background-color: #05366f; border-bottom: 1px solid
white;">
     <h3>Welcome to Online Learning System</h3>
     <h4>Login Page</h4>
</div>
```

```
<div class="container" style="min-height: calc(100vh - 126px - 24px);">

    <div class="row">
        <div class="col-sm-5 mt-5 mx-auto card p-4 shadow"
            style="background-color: #7fc6ec5c;">

            <form action="Validate" method="post">
                <div class="form-group">
                    <label>User ID</label> <input type="text" required name="userid"
                        class="form-control form-control-sm">
                </div>
                <div class="form-group">
                    <label>Password</label> <input type="password" required name="pwd"
                        class="form-control form-control-sm">
                </div>
                <div class="form-group">
                    <label>Select Role</label> <select name="role" required
                        class="form-control form-control-sm">
                        <option value="">&lt;-- Select Role --></option>
                        <option value="admin">Coordinator</option>
                        <option value="lecturer">Lecturer</option>
                        <option value="student">Student</option>
                    </select>
                </div>
                Not Registered
                <button type="button" data-target="#reg" data-toggle="modal"
                    class="btn btn-link">Click here to register</button>
                <button class="btn btn-primary float-right">Login</button>
            </form>
            <%
  if(session.getAttribute("msg")!=null){
      %>
            <div class="alert text-success mt-2 text-center font-weight-bold">
                <%= session.getAttribute("msg") %>
```

```
            </div>
            <%
         session.removeAttribute("msg");
  }
  %>
        </div>
        <div class="modal" id="reg">
            <div class="modal-dialog">
                <div class="modal-content">
                    <form action="Register" method="post">
                        <div class="modal-header">
                            <h5>Register Now</h5>
                            <button class="close" data-
dismiss="modal" type="button">&times;</button>
                        </div>
                        <div class="modal-body">
                            <div class="form-group">
                                <select id="role"
name="role" required
                                        class="form-control
form-control-sm">
                                        <option value=""><--
Select Role --></option>
                                        <option
value="lecturer">Lecturer</option>
                                        <option
value="student">Student</option>
                                </select>
                            </div>
                            <div class="form-group">
                                <input type="text" id="id"
placeholder="First Select role"
                                        name="id" required
class="form-control form-control-sm">
                            </div>
                            <div id="error"
                                class="text-danger font-
weight-bold text-center d-none"></div>
                            <div class="form-group">
                                <input type="text"
placeholder="User Name" readonly id="uname"
                                        name="uname"
class="form-control form-control-sm">
                            </div>
                            <div class="form-group">
```

```
                                        <input type="text" required
placeholder="User ID" name="userid"
                                            class="form-control
form-control-sm">
                                    </div>

                                    <div class="form-group">
                                        <input id="pwd" required
type="password" placeholder="Password"
                                            name="pwd" class="form-
control form-control-sm">
                                    </div>
                                    <div class="form-group">
                                        <input id="cpwd" required
type="password" placeholder="Repeat Password"
                                            name="cpwd" class="form-
control form-control-sm">
                                    </div>
                                    <div id="error2"
                                        class="text-danger font-
weight-bold text-center d-none"></div>
                                </div>
                                <div class="modal-footer">
                                    <button class="btn btn-success
float-right">Register</button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<jsp:include flush="true" page="footer.jsp" />
```

Admin Dashboard Page →

```
<%@page import="java.util.Map"%>
<%@page import="db.DbConfig"%>
<%@page contentType="text/html" session="true" %>

<jsp:include flush="true" page="header.jsp" />
<h3>Admin Dashboard</h3>
<div class="container">
    <div class="row">
        <div class="col-sm-3 bg-primary p-4 text-white">
            <h5>Uploaded Materials</h5>
```

```
            <h6><%= db.DbConfig.findCount("material")%></h6>
        </div>
        <div class="col-sm-3 bg-danger p-4 text-white">
            <h5>Lecturers</h5>
            <h6><%= db.DbConfig.findCount("lecturer")%></h6>
        </div>
        <div class="col-sm-3 bg-warning p-4 text-white">
            <h5>Students</h5>
            <h6><%= db.DbConfig.findCount("student")%></h6>
        </div>
        <div class="col-sm-3 bg-info p-4 text-white">
            <h5>Registered Users</h5>
            <h6><%= db.DbConfig.findCount("users")%></h6>
        </div>
    </div>
</div>
<jsp:include flush="true" page="footer.jsp" />

<%
         /*   } else {
                session.setAttribute("msg", "Please login
first..!!");
                response.sendRedirect("../index.jsp");
            }
        }
    } else {
        session.setAttribute("msg", "Please login first..!!");
        response.sendRedirect("../index.jsp");
    }*/
%>
```

Landing Page →

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Home</title>
```

```
    <link rel="stylesheet" href="Homepage_style.css" type="text/css">
</head>
<body>

    <div class="header">

        <video autoplay loop class="back-video" muted plays-inline>
            <source src="media/video.mp4" type="video/mp4">
        </video>

        <nav>
            <img src="media/logo.png" alt="logo" class="logo">
            <ul class="nav-links">
                <li><a href="#">Home</a></li>
                <li><a href="#">Courses</a></li>
                <li><a href="#">Reviews</a></li>
                <li><a href="contact.html">Contact Us</a></li>
                <li class="signInContainer">
                <li class="btn signin">Sign In/Register</li>
                <div class="dropDown">
                    <li><a href="login_register.html">Student</a></li>
                    <li><a href="teacher_login.html">Teacher</a></li>
                    <li><a href="Admin_login.html">Admin</a></li>
                </div>
                </li>

            </ul>
        </nav>
```

```
<div class="content">
    <h1>Learn Anything Online</h1>
    <form>
        <input type="text" placeholder="&#x270e; Enter Keyword">
        <button type="submit">Find Course</button>
    </form>

    <div class="category-list">
        <div class="category">
            <img src="media/coding.png">
            <p>Devlopment</p>
        </div>
        <div class="category">
            <img src="media/exercise.png">
            <p>Designing</p>
        </div>
        <div class="category">
            <img src="media/guitar.png">
            <p>IT Courses</p>
        </div>
        <div class="category">
            <img src="media/language.png">
            <p>Languages</p>
        </div>
        <div class="category">
            <img src="media/console.png">
            <p>Dev Ops</p>
        </div>
    </div>
```

```
    </div>
  </div>
  <script>
    const signIn=document.querySelector(".signin");
    const dropDown=document.querySelector('.dropDown');

    signIn.addEventListener('click',()=>{
      const clicked=dropDown.style.display;
      if(clicked=="flex"){
        dropDown.style.display="none";
      }
      else{
        dropDown.style.display="flex"
      }
    })
  </script>
</body>
</html>
```

msg.jsp →

```
<%
    if (session.getAttribute("msg") != null) {
%>
<div class="alert alert-success text-center font-weight-bold">
    <%=session.getAttribute("msg")%>
</div>
<%
    session.removeAttribute("msg");
}
if (session.getAttribute("error") != null) {
%>
<div class="alert alert-danger text-center font-weight-bold">
    <%=session.getAttribute("error")%>
```

```
</div>
<%
     session.removeAttribute("error");
}
%>
```

index.jsp in student folder →

```
<%@page import="db.DbConfig"%>
<%@page import="java.util.Map"%>
<jsp:include flush="true" page="header.jsp" />
<style>
ol li {
     border: 1px solid white;
     padding: 10px;
}
</style>
<div class="card shadow" style="min-height: 88vh;">
     <div class="card-body">
          <h5 class="p-2" style="border-bottom: 2px solid
green;">Student
               Dashboard</h5>
          <div class="container-fluid">
               <div class="row">
                    <div class="col-sm-6 ">
                         <h5 class="p-2" style="border-bottom: 2px
solid green;">Student Profile</h5>
                         <%
                  Map<String,String>
map=DbConfig.findSingle("student",
"rollno="+session.getAttribute("id"));
             %>
                         <table class="table table-bordered ">
                              <tr>
                                   <th>Roll no</th>
                                   <th><%= map.get("rollno") %></th>
                              </tr>
                              <tr>
                                   <th>Student Name</th>
                                   <th><%= map.get("name") %></th>
                              </tr>
                              <tr>
                                   <th>Gender</th>
                                   <th><%= map.get("gender") %></th>
```

```
                                </tr>
                                <tr>
                                    <th>City</th>
                                    <th><%= map.get("city") %></th>
                                </tr>
                                <tr>
                                    <th>Email ID</th>
                                    <th><%= map.get("email") %></th>
                                </tr>
                            </table>
                        </div>
                        <div class="col-sm-6 card bg-transparent "
style="height: 400px;">
                            <h5 class="p-2" style="border-bottom: 2px
solid green;">Notifications</h5>
                            <marquee onmouseover="stop()"
onmouseout="start()" direction="up"
                                style="height: 100%">
                                <ol>
                                    <% for(Map<String,String> row :
DbConfig.allmsgs(session.getAttribute("id").toString())) {
                                String
lname=DbConfig.findSingle("lecturer",
"lid="+row.get("lid")).get("lname");
                                %>
                                    <li class="p-2 border-top border-
bottom">Message from <b><%= lname %></b>
                                    <span class="float-right"><%=
row.get("mdate") %></span>
                                    <br> <%= row.get("msg") %></li>
                                    <% } %>
                                </ol>
                            </marquee>
                        </div>
                    </div>
                </div>
            </div>
        </div>
<jsp:include flush="true" page="footer.jsp" />
```

DbConfig.java →

```
package db;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;


public class DbConfig {


   public static Connection connect() throws Exception {
      Class.forName("com.mysql.jdbc.Driver");
      final String URL = "jdbc:mysql://localhost/learningdb";
      final String USER = "root";
      final String PWD = "";
      Connection con = DriverManager.getConnection(URL, USER, PWD);
      return con;
   }


   public static Map<String, String> findSingle(String table, String condition) throws Exception {
      Connection con = connect();
```

```
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM " + table + "
WHERE " + condition);

        ResultSetMetaData rsm = rs.getMetaData();

        int cols = rsm.getColumnCount();

        HashMap<String, String> map = new HashMap<String, String>();

        rs.next();

        for (int i = 1; i <= cols; i++) {

            map.put(rsm.getColumnName(i), rs.getString(i));

        }

        con.close();

        return map;

    }


    public static String findCount(String table) throws Exception {

        Connection con = connect();

        ResultSet rs = con.createStatement().executeQuery("SELECT count(*) FROM " +
table);

        rs.next();

        String result=rs.getString(1);

        con.close();

        return result;

    }


    public static List<Map<String, String>> allRecords(String table) throws Exception {

        Connection con = connect();

        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM " + table);

        ResultSetMetaData rsm = rs.getMetaData();

        int cols = rsm.getColumnCount();
```

```
    List<Map<String, String>> list = new ArrayList<>();
    while (rs.next()) {
        Map<String,String> map=new HashMap<>();
        for (int i = 1; i <= cols; i++) {
            map.put(rsm.getColumnName(i), rs.getString(i));
        }
        list.add(map);
    }
    con.close();
    return list;
  }

  public static List<Map<String, String>> findall(String table,String condition) throws Exception {
    Connection con = connect();
    ResultSet rs = con.createStatement().executeQuery("SELECT * FROM " + table+" WHERE "+condition);
    ResultSetMetaData rsm = rs.getMetaData();
    int cols = rsm.getColumnCount();
    List<Map<String, String>> list = new ArrayList<>();
    while (rs.next()) {
        Map<String,String> map=new HashMap<>();
        for (int i = 1; i <= cols; i++) {
            map.put(rsm.getColumnName(i), rs.getString(i));
        }
        list.add(map);
    }
    con.close();
```

```
        return list;
    }


    public static List<Map<String, String>> sublect() throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM subjects WHERE subid not in(SELECT subid from lecsub)");
        ResultSetMetaData rsm = rs.getMetaData();
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
            list.add(map);
        }
        con.close();
        return list;
    }


    public static List<Map<String, String>> allmsgs(String rollno) throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM msgs WHERE lid in (SELECT lid from lecsub where subid in (SELECT subid FROM subjects WHERE cid=(SELECT class FROM student WHERE rollno="+rollno+"))) order by mdate desc");
        ResultSetMetaData rsm = rs.getMetaData();
```

```
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
            list.add(map);
        }
        con.close();
        return list;
    }



    public static List<Map<String, String>> allLectmsgs(String lid) throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM msgs WHERE lid="+lid+" order by mdate desc");
        ResultSetMetaData rsm = rs.getMetaData();
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
            list.add(map);
```

```
        }
        con.close();
        return list;
    }

    public static List<Map<String, String>> studentmat(String rollno) throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM material WHERE subid in (SELECT subid from subjects where cid=(SELECT class FROM student WHERE rollno="+rollno+"))");
        ResultSetMetaData rsm = rs.getMetaData();
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
            list.add(map);
        }
        con.close();
        return list;
    }

    public static List<Map<String, String>> lectsubs(String lid) throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT * FROM subjects WHERE subid in(SELECT subid from lecsub where lid="+lid+")");
```

```
        ResultSetMetaData rsm = rs.getMetaData();
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
            list.add(map);
        }
        con.close();
        return list;
    }


    public static List<Map<String, String>> studentlects(String rollno) throws Exception {
        Connection con = connect();
        ResultSet rs = con.createStatement().executeQuery("SELECT lid,lname FROM lecturer WHERE lid in(SELECT lid from  lecsub where subid in (SELECT subid FROM subjects WHERE cid=(SELECT class FROM student WHERE rollno="+rollno+"))) and lid not in(SELECT lid from evaluation WHERE rollno="+rollno+")");
        ResultSetMetaData rsm = rs.getMetaData();
        int cols = rsm.getColumnCount();
        List<Map<String, String>> list = new ArrayList<>();
        while (rs.next()) {
            Map<String,String> map=new HashMap<>();
            for (int i = 1; i <= cols; i++) {
                map.put(rsm.getColumnName(i), rs.getString(i));
            }
```

```
            list.add(map);
        }
        con.close();
        return list;
    }
}
```

AddLecturerServlet.java →

```
package com.learning;


import db.DbConfig;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
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;


@WebServlet("/LAdd")
public class AddLecturerServlet extends HttpServlet {


    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
```

```
        String lname=req.getParameter("lname");
        String city=req.getParameter("city");
        String email=req.getParameter("email");
        String qual=req.getParameter("qual");
        String gender=req.getParameter("gender");
        HttpSession session=req.getSession();
        try{
            Connection con=DbConfig.connect();
            PreparedStatement ps=con.prepareStatement("INSERT INTO
lecturer(lname,city,gender,email,qual) VALUES(?,?,?,?,?)");
            ps.setString(1,lname);
            ps.setString(2,city);
            ps.setString(3,gender);
            ps.setString(4,email);
            ps.setString(5,qual);
            ps.executeUpdate();
            con.close();
            session.setAttribute("msg", "Lecturer Added Successfully..!");
            resp.sendRedirect("admin/lecturers.jsp");
        }
        catch(Exception ex){
            System.err.println("Error "+ex.getMessage());
        }
    }

}
```

SubAssignServlet.java →

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.learning;


import db.DbConfig;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
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;


@WebServlet("/Assign")
public class SubAssignServlet extends HttpServlet {


    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        try {
```

```
            Connection con=DbConfig.connect();
            PreparedStatement ps = con
                    .prepareStatement("INSERT INTO lecsub VALUES(?,?)");
            ps.setString(2, request.getParameter("subid"));
            ps.setString(1, request.getParameter("lid"));
            ps.executeUpdate();
            con.close();
            session.setAttribute("msg", "Subject assigned successfully..");
            response.sendRedirect("admin/assign.jsp");
        } catch (Exception ex) {
            System.err.println("Error " + ex.getMessage());
        }

    }

}
```

DeleteSubAssignServlet.java →

```
package com.learning;


import db.DbConfig;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
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;


@WebServlet("/delassign")
public class DeleteSubAssignServlet extends HttpServlet {


   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
       HttpSession session = request.getSession();
       try {
           Connection con=DbConfig.connect();
           PreparedStatement ps = con
                   .prepareStatement("delete from lecsub where subid=? and lid=?");
           ps.setString(1, request.getParameter("subid"));
           ps.setString(2, request.getParameter("lid"));
           ps.executeUpdate();
           con.close();
           session.setAttribute("msg", "Assignment deleted successfully..");
           response.sendRedirect("admin/assign.jsp");
       } catch (Exception ex) {
           System.err.println("Error " + ex.getMessage());
       }
   }
}
```

RegisterServlet.java →

```
package com.learning;


import db.DbConfig;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

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;


@WebServlet("/Register")

public class RegisterServlet   extends HttpServlet {


   @Override

   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

      String userid=req.getParameter("userid");

      String pwd=req.getParameter("pwd");

      String role=req.getParameter("role");

      String uname=req.getParameter("uname");

      String id=req.getParameter("id");

      HttpSession session=req.getSession();
```

```
        try{
            Connection con=DbConfig.connect();
            PreparedStatement ps=con.prepareStatement("INSERT INTO users(userid,uname,pwd,role,id) VALUES(?,?,?,?,?)");
            ps.setString(1,userid);
            ps.setString(2,uname);
            ps.setString(3,pwd);
            ps.setString(4,role);
            ps.setString(5,id);
            ps.executeUpdate();
            con.close();
            session.setAttribute("msg", "User Registered Successfully..!");
            resp.sendRedirect("index.jsp");
        }
        catch(Exception ex){
            System.err.println("Error "+ex.getMessage());
        }
    }

}
```

Chapter-7: Testing

Testing

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users. In other word software testing is an activity to check that the software system is defect free. Software testing is primarily a broad process that is composed of several interlinked processes. The primary objective of software testing is to measure software health along with its completeness in terms of core requirements. Software testing involves examining and checking software through different testing processes.

The objectives of these processes can include:

- Completeness - Verifying software completeness in regards to functional/business requirements
- Errors Free - Identifying technical bugs/errors and ensuring the software is error-free
- Stability - Assessing usability, performance, security, localization, compatibility and installation

This phase determines the error in the project. If there is any error then it must be removed before delivery of the project.

7.1. Types of Testing

For determining errors various types of test action are performed: -

Unit Testing: - Unit testing focuses verification effort on the smallest unit of software design – the module. Using the detail design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing. The unit test is always white box oriented, and the step can be conducted in parallel for multiple modules. Unit testing is normally considered an adjunct to the coding step. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins.

Integration Testing - A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

System Testing - Software is only one element of a larger computer-based system. Ultimately, software is incorporated with other system elements (e.g. new hardware,

information), and a series of system integration and validation tests are conducted. Steps taken during software design and testing can greatly improve the probability of successful software integration in the larger system.

There are many types of system tests that are worthwhile for software-based systems: -

Usability Testing - Usability Testing is a type of testing done from an end-user's perspective to determine if the system is easily usable.

Functionality testing - Tests all functionalities of the software against the requirement.

Performance testing – Performance testing is designed to test the run-time performance of software within the context of an integrated system

Security testing – Security testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration

Stress tests – Stress tests are designed to confront programs with abnormal situations.

7.2. Test Cases

A test case 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.

Registration:

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Status Pass/Fail
TC_REG_1	Verify Registration Detail	Enter Valid and correct data	Need valid text and number Data to be entered	1. Enter Valid Data in appropriate fields 2. Click Submit	Valid Text and Number Data	Successful, Member Added Successfully	Successful, Member Added Successfully	Pass
TC_REG_2	Verify Registration Detail	Enter invalid and incorrect data	Need text and number Data to be entered	Enter invalid Data in fields	Invalid Text and Number Data	Enter Valid Data	Enter Valid Data	Pass
TC_REG_3	Verify Registration Detail	Entering Nothing, Required Fields are blank	-	Click Submit	Nothing to enter Required fields are blank	Fill required field	Fill required field	Pass

Table 1: Taste cases For Registration

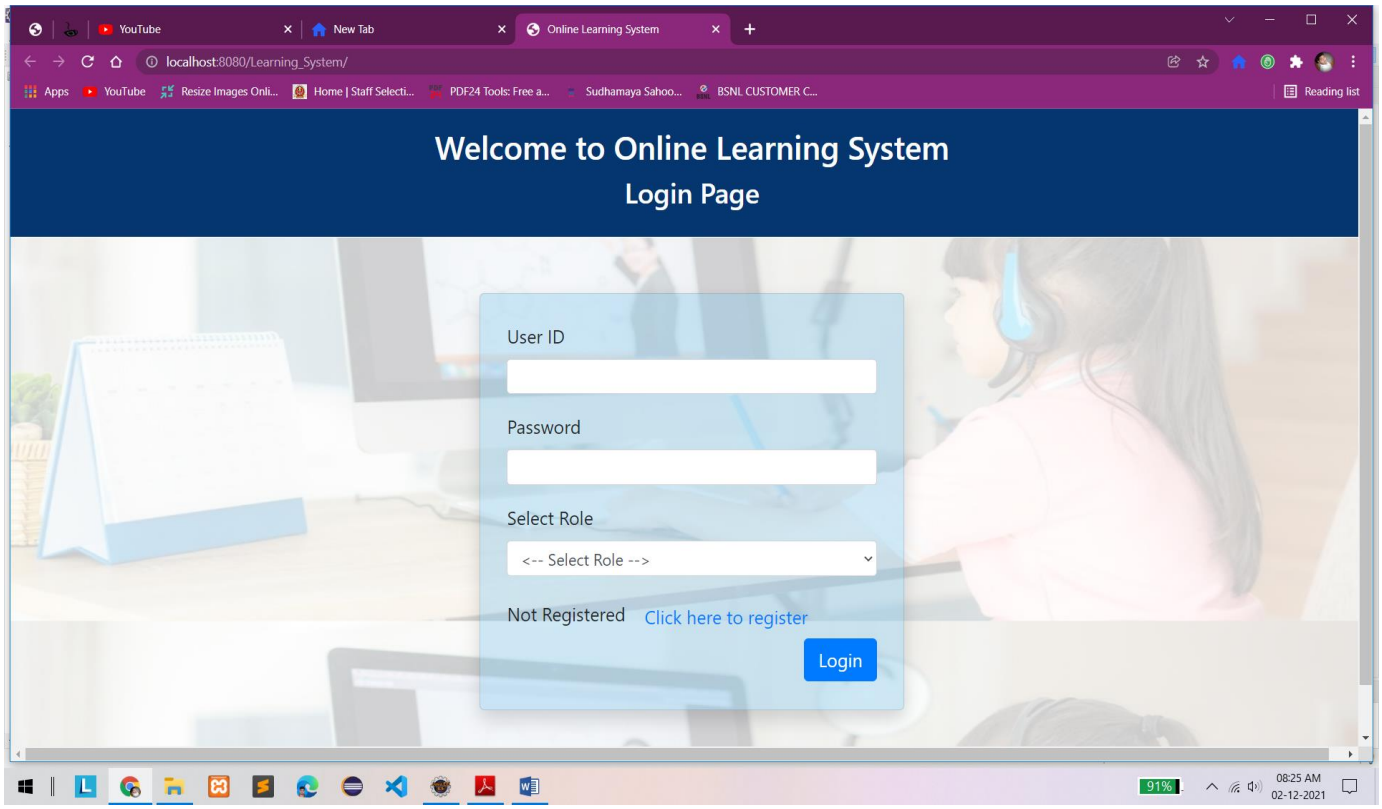
Login:

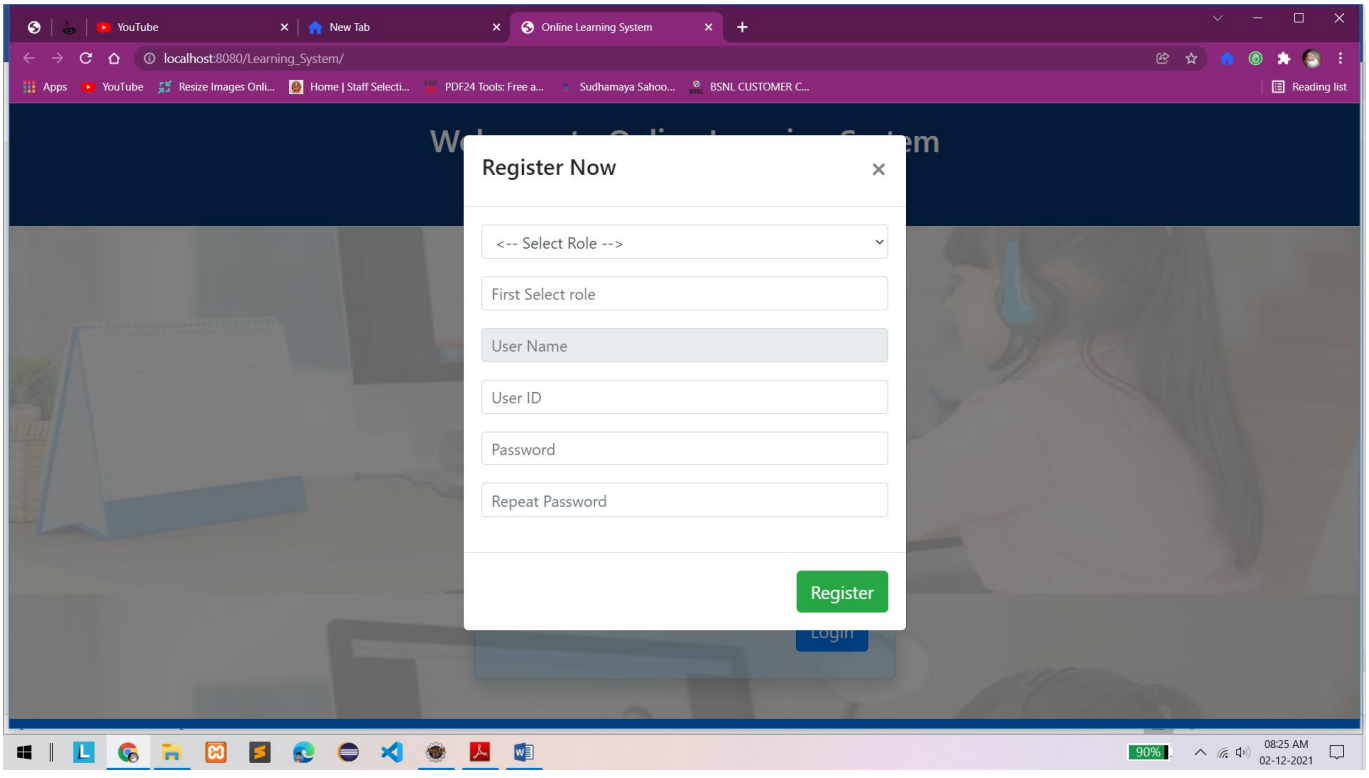
Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Status Pass/Fail
TC_Login_1	Verify Login	Enter Valid username and valid password	Need a valid username and password to do login	1. Enter username 2. Enter Password 3. Click Login	Valid username Valid password	Successful login, Main screen of application should displayed	Successful login, Main screen of application displayed	Pass
TC_Login_2	Verify Login	Enter Valid username and invalid password	Need a valid username and password to do login	1. Enter username 2. Enter Password 3. Click Login	Valid username Invalid Password	No Matched Username/ Password	No Matched Username/ Password	Pass
TC_Login_3	Verify Login	Enter Invalid username and valid password	Need a valid username and password to do login	1. Enter username 2. Enter Password 3. Click Login	Invalid username Valid Password	No Matched Username/ Password	No Matched Username/ Password	Pass
TC_Login_4	Verify Login	Enter Invalid username and invalid password	Need a valid username and password to do login	1. Enter username 2. Enter Password 3. Click Login	Invalid username Invalid Password	No Matched Username/ Password	No Matched Username/ Password	Pass

Table 2: Taste cases For Log In

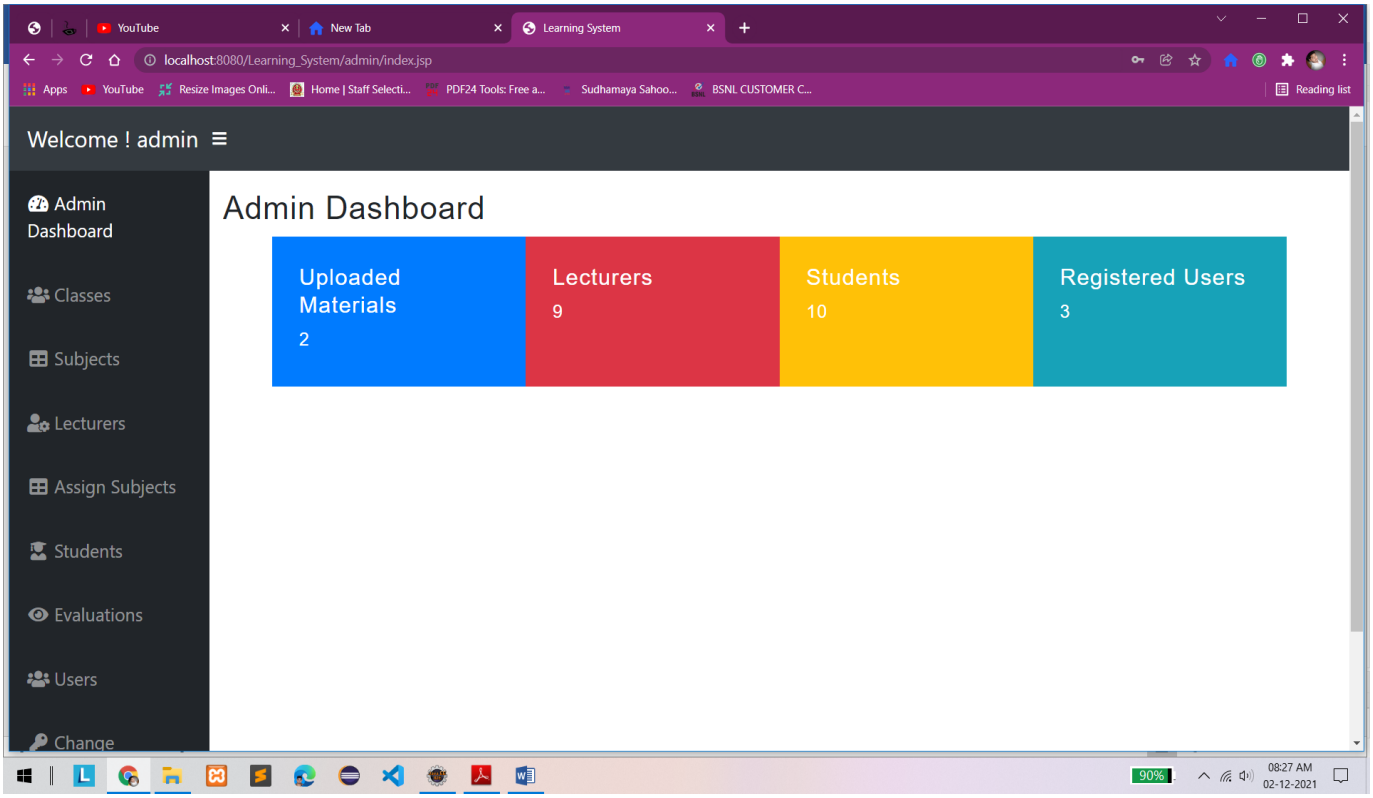
Chapter-8: Output Screens

Log In / Sign Up Page →

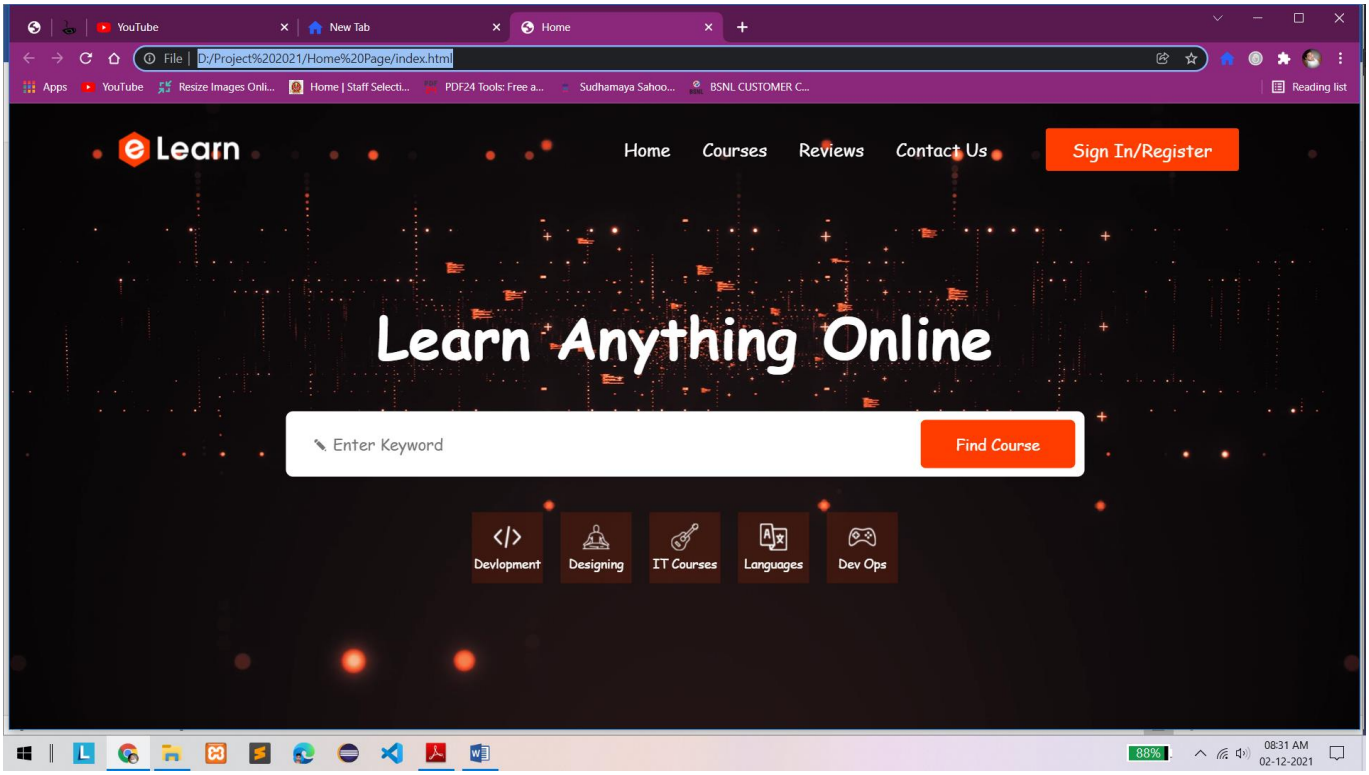




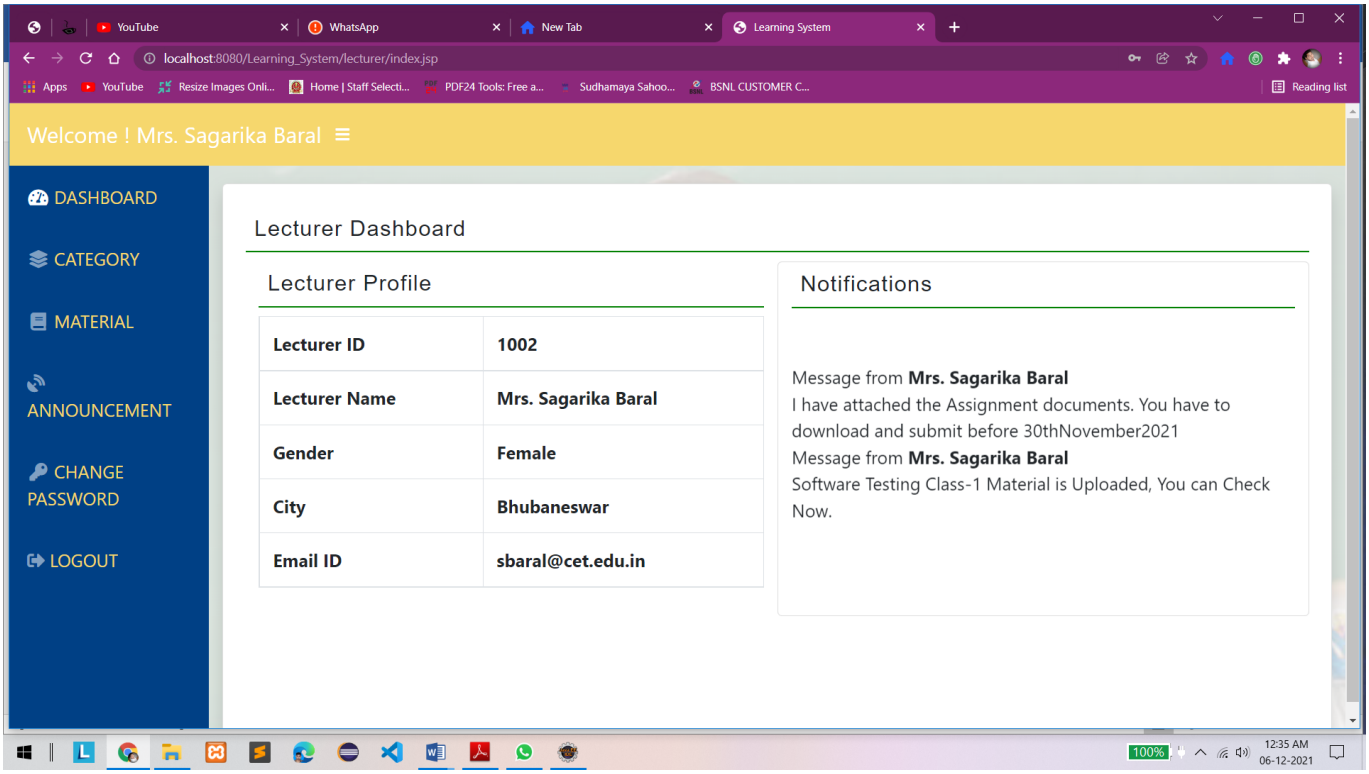
Admin Dashboard →



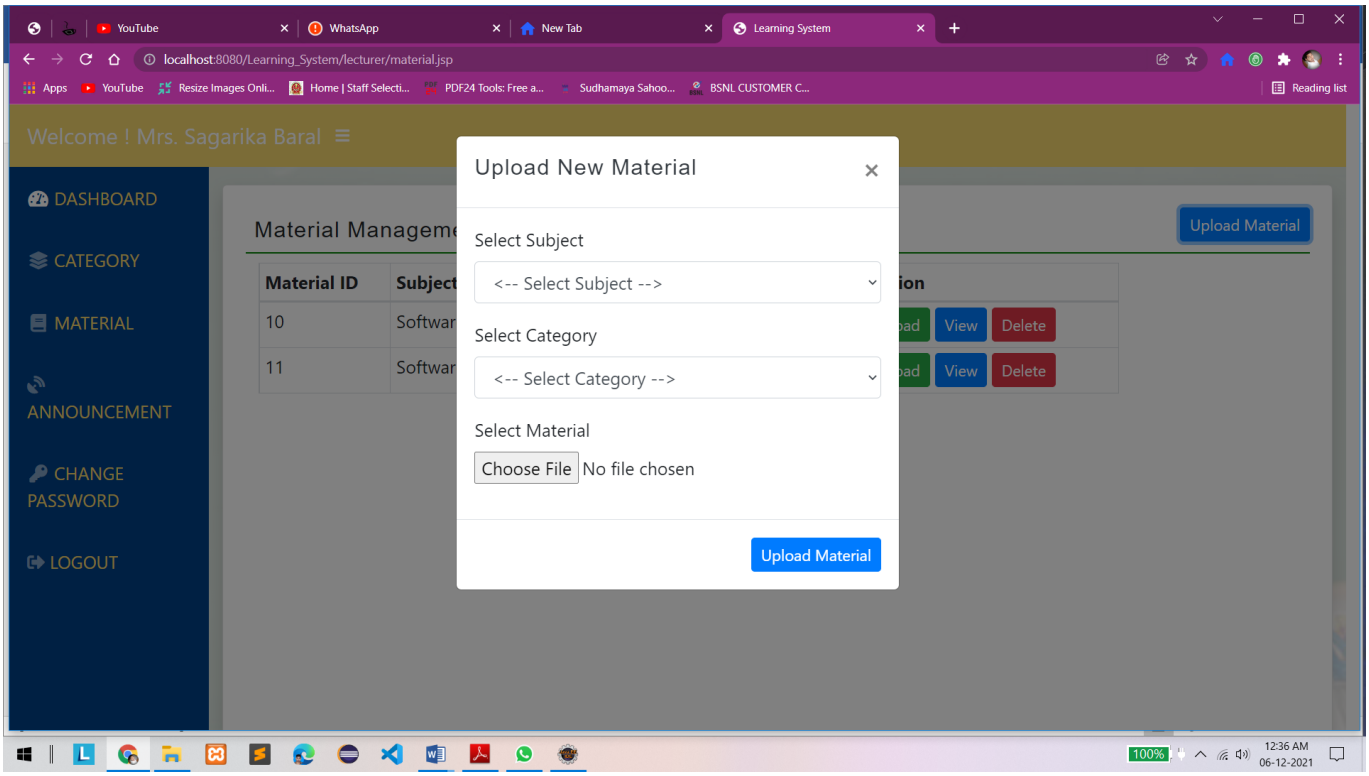
Home Page or Landing Page –>



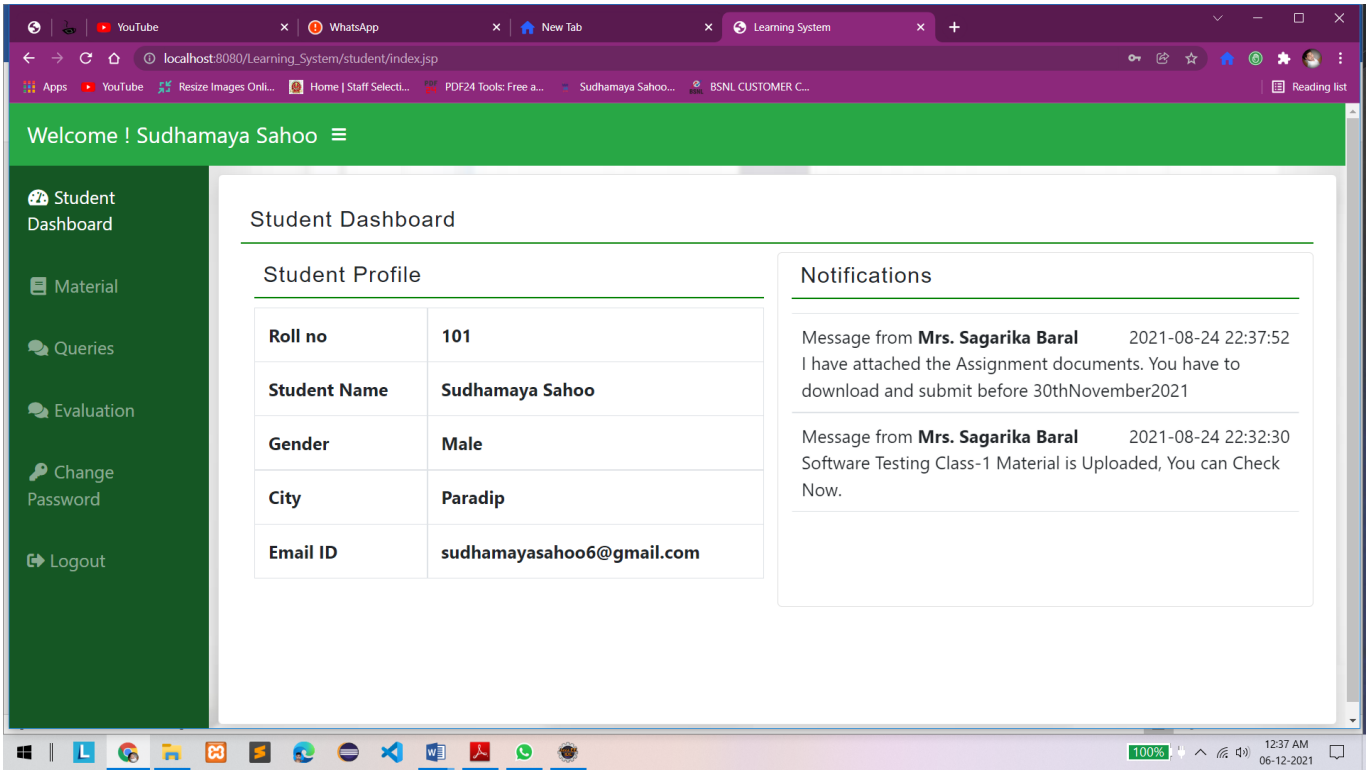
Teachers Dashboard →



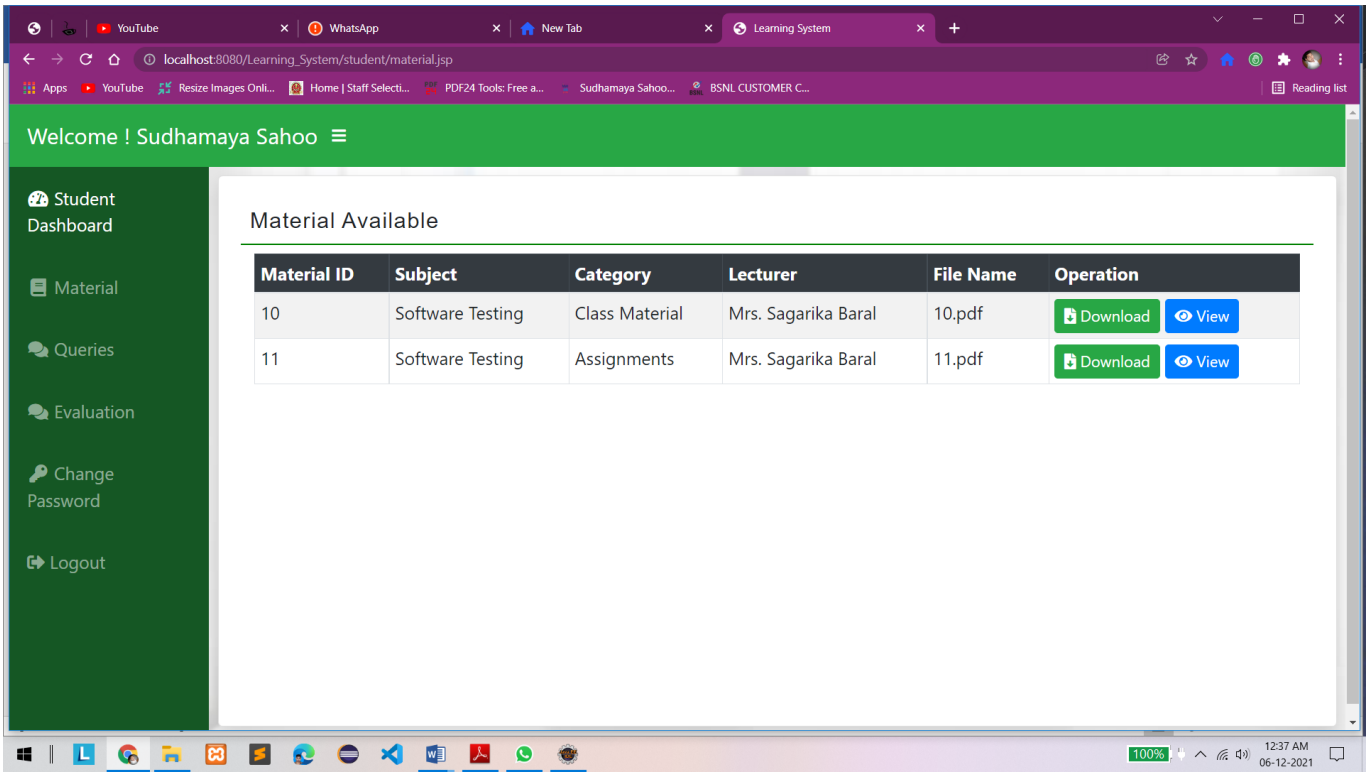
Material →



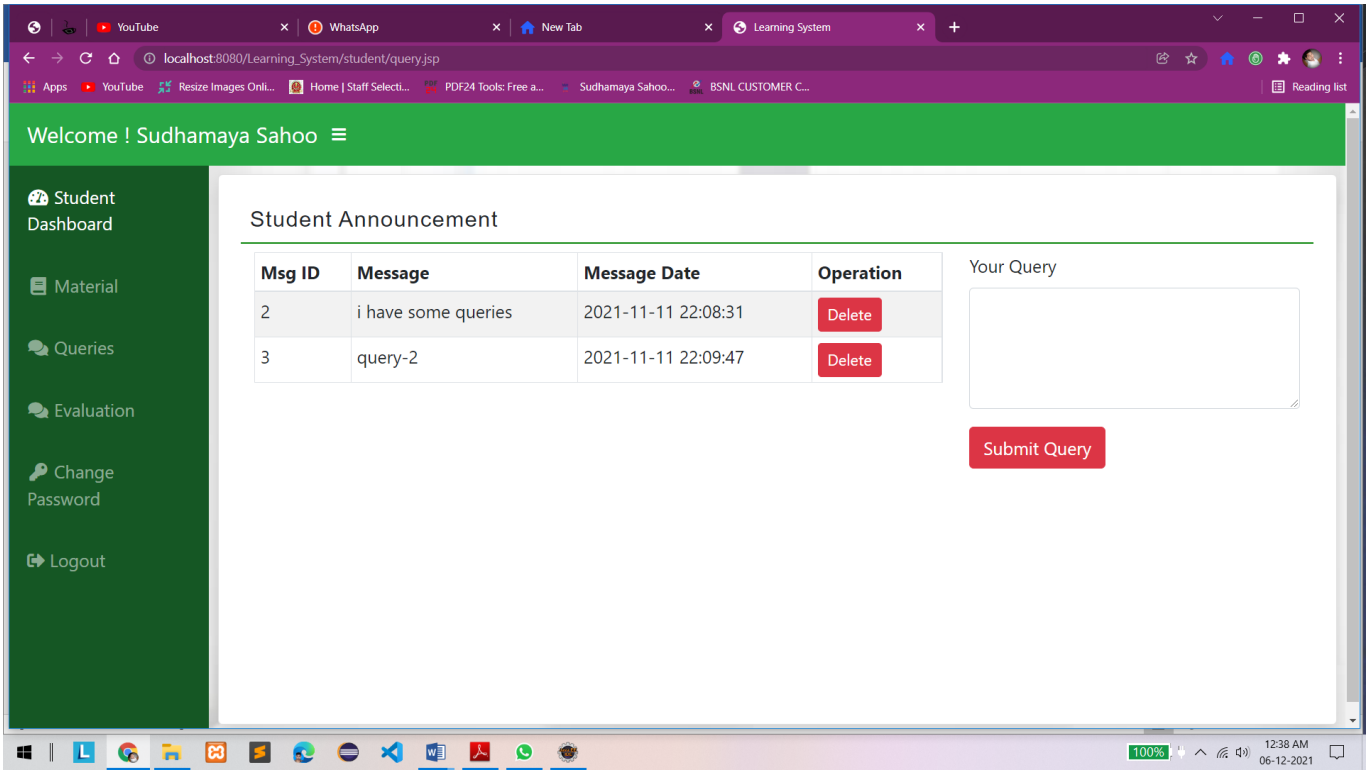
Student Dashboard →



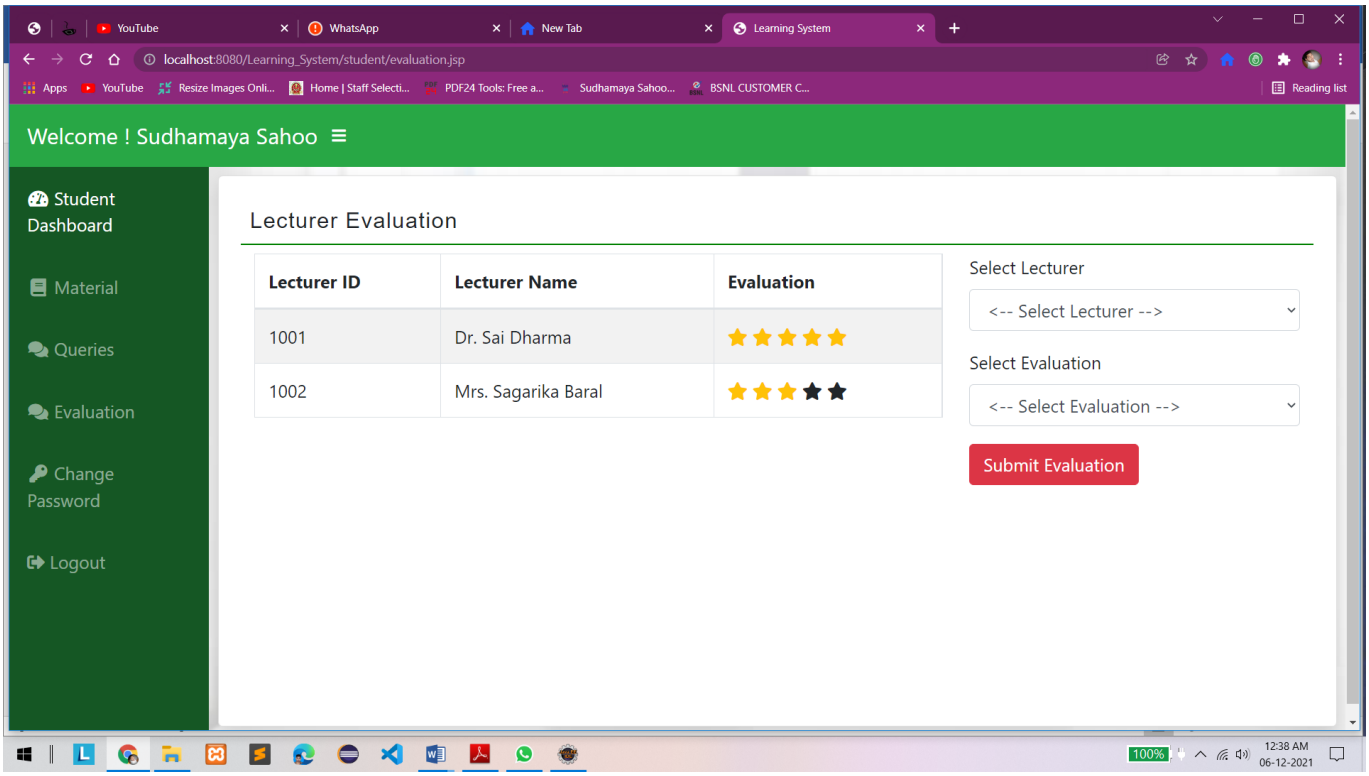
Materials →



Student Queries →



Submit Evaluations →



Chapter-9: Database Design View

SQL Queries →

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;


--

-- Database: `learningdb`


CREATE TABLE `class` (

  `cid` int(11) NOT NULL,

  `cname` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;


INSERT INTO `class` (`cid`, `cname`) VALUES

(1, 'MCA-5th Sem'),

(2, 'MCA-4th Sem'),

(3, 'MCA-3rd Sem'),

(5, 'MCA-2nd Sem'),

(6, 'MCA-1st Sem');
```

```
CREATE TABLE `evaluation` (
  `rollno` int(11) NOT NULL,
  `lid` int(11) NOT NULL,
  `eval` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `evaluation` (`rollno`, `lid`, `eval`) VALUES
(101, 1001, 5),
(101, 1002, 3),
(102, 1001, 5),
(102, 1002, 4),
(102, 1003, 4),
(106, 1003, 5),
(108, 1001, 5),
(108, 1002, 5),
(108, 1003, 4),
(108, 1004, 4);

CREATE TABLE `lecsub` (
  `lid` int(11) NOT NULL,
  `subid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `lecsub` (`lid`, `subid`) VALUES
(1002, 3),
(1004, 18),
```

```
(1005, 13),
(1006, 12),
(1007, 10),
(1008, 2),
(1008, 15),
(1009, 1),
(1009, 11);

CREATE TABLE `lecturer` (
  `lid` int(11) NOT NULL,
  `lname` varchar(30) NOT NULL,
  `city` varchar(30) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `email` varchar(30) NOT NULL,
  `qual` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `lecturer` (`lid`, `lname`, `city`, `gender`, `email`, `qual`) VALUES
(1001, 'Dr. Sai Dharma', 'Bhubaneswar', 'Male', 'sdharma@cet.edu.in', 'Ph.D. (IIT,KGP)'),
(1002, 'Mrs. Sagarika Baral', 'Bhubaneswar', 'Female', 'sbaral@cet.edu.in', 'M.Tech. (NIT, RKL)'),
(1003, 'Mr. Sarthak Rout', 'Cuttack', 'Male', 'sroutcse@cet.edu.in', 'M.Tech. (Utkal Univ.)'),
(1004, 'Mrs. Sasmita Sahoo', 'Cuttack', 'Female', 'sasmitasahoo@gmail.com', 'MCA(UU)'),
(1005, 'Mr. Partha Sarathi', 'Bhubaneswar', 'Male', 'psarathi@gmail.com', 'M.Tech. (NIT, RKL)'),
(1006, 'Mr. Swayam Rout', 'Bhubaneswar', 'Male', 'sroutcse@cet.edu.in', 'M.Tech (CET)'),
```

```
(1007, 'Mr. Akash Das', 'Puri', 'Male', 'akdas.puri@gmail.com', 'M.Tech'),
(1008, 'Mrs. Sibani Swain', 'Bhubaneswar', 'Female', 'sibaniswain@gmail.com', 'MCA (SOA)'),
(1009, 'Mr. Sai Sandeepan Sahoo', 'Cuttack', 'Male', 'sssahoo@cet.edu.in', 'M.Tech');

CREATE TABLE `matcat` (
  `catid` int(11) NOT NULL,
  `catname` varchar(30) NOT NULL,
  `lid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `matcat` (`catid`, `catname`, `lid`) VALUES
(4, 'Assignments', 1002),
(6, 'Class Material', 1002);

CREATE TABLE `material` (
  `matid` int(11) NOT NULL,
  `lid` int(11) NOT NULL,
  `subid` int(11) NOT NULL,
  `catid` int(11) NOT NULL,
  `filename` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `material` (`matid`, `lid`, `subid`, `catid`, `filename`) VALUES
(10, 1002, 3, 6, '10.pdf'),
(11, 1002, 3, 4, '11.pdf');
```

```
CREATE TABLE `msgs` (
  `msgid` int(11) NOT NULL,
  `lid` int(11) NOT NULL,
  `msg` text NOT NULL,
  `mdate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `msgs` (`msgid`, `lid`, `msg`, `mdate`) VALUES
(5, 1002, 'Software Testing Class-1 Material is Uploaded, You can Check Now. ', '2021-08-24 22:32:30'),
(6, 1002, 'I have attached the Assignment documents. You have to download and submit before 30thNovember2021 ', '2021-08-24 22:37:52');

CREATE TABLE `queries` (
  `qid` int(11) NOT NULL,
  `rollno` int(11) NOT NULL,
  `msg` text NOT NULL,
  `mdate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `queries` (`qid`, `rollno`, `msg`, `mdate`) VALUES
(2, 101, 'i have some queries', '2021-11-11 22:08:31'),
(3, 101, 'query-2', '2021-11-11 22:09:47');

CREATE TABLE `student` (
  `rollno` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
```

```
  `city` varchar(20) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `class` varchar(20) NOT NULL,
  `email` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `student` (`rollno`, `name`, `city`, `gender`, `class`, `email`) VALUES
(101, 'Sudhamaya Sahoo', 'Paradip', 'Male', '1', 'sudhamayasahoo6@gmail.com'),
(102, 'Purnendu Sahoo', 'Balasore', 'Male', '1', 'purnendusahoo16@gmail.com'),
(104, 'Sudhanshu Barala', 'Dhenkanal', 'Male', '2', 'bvgroup@gmail.com'),
(105, 'Arup Kumar Subudhi', 'Jajpur', 'Male', '2', 'arup@gmail.com'),
(106, 'Lalit Kumar Soren', 'Balasore', 'Male', '3', 'lalitsoren@gmail.com'),
(107, 'Jagdish Sahoo', 'Keonjhar', 'Male', '3', 'jagdishsahoo@yahoo.com'),
(108, 'Anupam Patra', 'Bhubaneswar', 'Male', '5', 'anupam@gmail.com'),
(109, 'Asish Kumar Panda', 'Sambalpur', 'Male', '5', 'asish@gmail.com'),
(110, 'N Priyajit', 'Cuttack', 'Male', '6', 'npriyajit@gmail.com'),
(111, 'Suraj Roy', 'Cuttack', 'Male', '6', 'sroy@gmail.com');

CREATE TABLE `subjects` (
  `subid` int(11) NOT NULL,
  `subname` varchar(30) NOT NULL,
  `cid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `subjects` (`subid`, `subname`, `cid`) VALUES
(1, 'Machine Learning', 1),
```

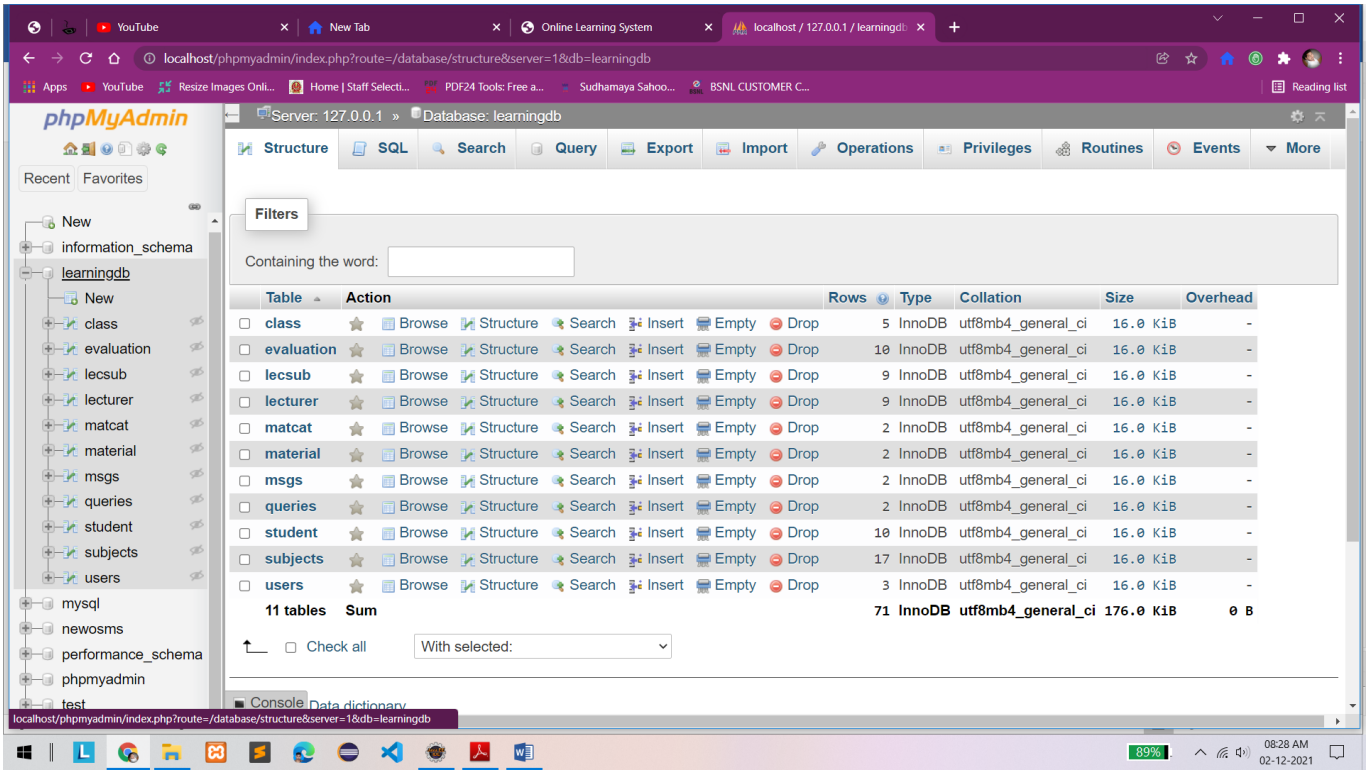
```
(2, 'MPWD', 1),
(3, 'Software Testing', 1),
(4, 'Artificial Intelligence', 2),
(8, 'Optimization Technique', 2),
(9, 'EJT', 2),
(10, 'Computational Mathematics', 3),
(11, 'Data Mining', 3),
(12, 'Computer Networks', 1),
(13, 'Computer Security', 5),
(14, 'DAA', 5),
(15, 'JAVA', 5),
(16, 'Mathematics', 6),
(17, 'PSPD', 6),
(18, 'COA', 6),
(19, 'WDD', 6),
(20, 'Operating System', 5);

CREATE TABLE `users` (
  `userid` varchar(40) NOT NULL,
  `uname` varchar(40) NOT NULL,
  `pwd` varchar(20) NOT NULL,
  `role` varchar(15) NOT NULL,
  `id` int(11) NOT NULL,
  `status` varchar(20) NOT NULL DEFAULT 'active'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

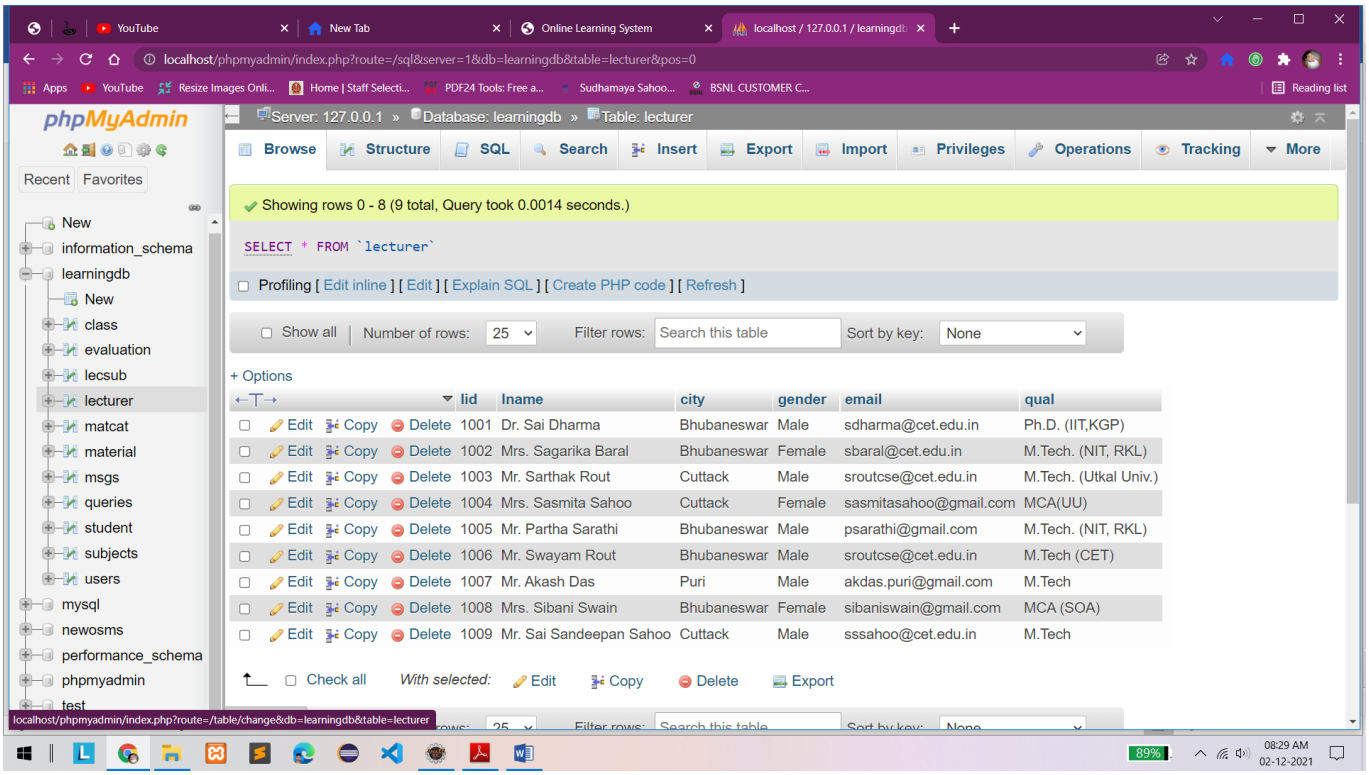
```
INSERT INTO `users` (`userid`, `uname`, `pwd`, `role`, `id`, `status`) VALUES
('admin', 'admin', 'admin', 'admin', 0, 'active'),
('sbaral', 'Mrs. Sagarika Baral', 'password', 'lecturer', 1002, 'active'),
('sudha', 'Sudhamaya Sahoo', 'password', 'student', 101, 'active');

ALTER TABLE `class`
  ADD PRIMARY KEY (`cid`);

ALTER TABLE `evaluation`
  ADD PRIMARY KEY (`rollno`,`lid`);
ALTER TABLE `lecsub`
  ADD PRIMARY KEY (`lid`,`subid`);
ALTER TABLE `lecturer`
  ADD PRIMARY KEY (`lid`);

ALTER TABLE `matcat`
  ADD PRIMARY KEY (`catid`);

ALTER TABLE `material`
  ADD PRIMARY KEY (`matid`);

ALTER TABLE `msgs`
  ADD PRIMARY KEY (`msgid`);

ALTER TABLE `queries`
  ADD PRIMARY KEY (`qid`);
```

```
ALTER TABLE `student`
  ADD PRIMARY KEY (`rollno`);


ALTER TABLE `subjects`
  ADD PRIMARY KEY (`subid`);


ALTER TABLE `users`
  ADD PRIMARY KEY (`userid`);


ALTER TABLE `class`
  MODIFY `cid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;


ALTER TABLE `lecturer`
  MODIFY `lid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1010;


--
-- AUTO_INCREMENT for table `matcat`
--
ALTER TABLE `matcat`
  MODIFY `catid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;


--
-- AUTO_INCREMENT for table `material`
--
ALTER TABLE `material`
  MODIFY `matid` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=12;
```

```
--
-- AUTO_INCREMENT for table `msgs`
--
ALTER TABLE `msgs`
  MODIFY `msgid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;


--
-- AUTO_INCREMENT for table `queries`
--
ALTER TABLE `queries`
  MODIFY `qid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;


--
-- AUTO_INCREMENT for table `student`
--
ALTER TABLE `student`
  MODIFY `rollno` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=112;


--
-- AUTO_INCREMENT for table `subjects`
--
ALTER TABLE `subjects`
  MODIFY `subid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
COMMIT;
```

Database –>


phpMyAdmin
Server: 127.0.0.1 » Database: learningdb
Structure SQL Search Query Export Import Operations Privileges Routines Events More
Filters
Containing the word:
Table Action Rows Type Collation Size Overhead
class Browse Structure Search Insert Empty Drop 5 InnoDB utf8mb4_general_ci 16.0 KiB -
evaluation Browse Structure Search Insert Empty Drop 10 InnoDB utf8mb4_general_ci 16.0 KiB -
lecsub Browse Structure Search Insert Empty Drop 9 InnoDB utf8mb4_general_ci 16.0 KiB -
lecturer Browse Structure Search Insert Empty Drop 9 InnoDB utf8mb4_general_ci 16.0 KiB -
matcat Browse Structure Search Insert Empty Drop 2 InnoDB utf8mb4_general_ci 16.0 KiB -
material Browse Structure Search Insert Empty Drop 2 InnoDB utf8mb4_general_ci 16.0 KiB -
msgs Browse Structure Search Insert Empty Drop 2 InnoDB utf8mb4_general_ci 16.0 KiB -
queries Browse Structure Search Insert Empty Drop 2 InnoDB utf8mb4_general_ci 16.0 KiB -
student Browse Structure Search Insert Empty Drop 10 InnoDB utf8mb4_general_ci 16.0 KiB -
subjects Browse Structure Search Insert Empty Drop 17 InnoDB utf8mb4_general_ci 16.0 KiB -
users Browse Structure Search Insert Empty Drop 3 InnoDB utf8mb4_general_ci 16.0 KiB -
11 tables Sum 71 InnoDB utf8mb4_general_ci 176.0 KiB 0 B
Check all With selected:
Console Data dictionary
Recent Favorites
New
information_schema
learningdb
class
evaluation
lecsub
lecturer
matcat
material
msgs
queries
student
subjects
users
mysql
newosms
performance_schema
phpmyadmin
test
localhost/phpmyadmin/index.php?route=/database/structure&server=1&db=learningdb
89%
08:28 AM
02-12-2021


phpMyAdmin
Server: 127.0.0.1 » Database: learningdb » Table: lecturer
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking More
Showing rows 0 - 8 (9 total, Query took 0.0014 seconds.)
SELECT * FROM `lecturer`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
lid lname city gender email qual
Edit Copy Delete 1001 Dr. Sai Dharma Bhubaneswar Male sdharma@cet.edu.in Ph.D. (IIT,KGP)
Edit Copy Delete 1002 Mrs. Sagarika Baral Bhubaneswar Female sbaral@cet.edu.in M.Tech. (NIT, RKL)
Edit Copy Delete 1003 Mr. Sarthak Rout Cuttack Male sroutcse@cet.edu.in M.Tech. (Utkal Univ.)
Edit Copy Delete 1004 Mrs. Sasmita Sahoo Cuttack Female sasmitasahoo@gmail.com MCA(UU)
Edit Copy Delete 1005 Mr. Partha Sarathi Bhubaneswar Male psarathi@gmail.com M.Tech. (NIT, RKL)
Edit Copy Delete 1006 Mr. Swayam Rout Bhubaneswar Male sroutcse@cet.edu.in M.Tech (CET)
Edit Copy Delete 1007 Mr. Akash Das Puri Male akdas.puri@gmail.com M.Tech
Edit Copy Delete 1008 Mrs. Sibani Swain Bhubaneswar Female sibaniswain@gmail.com MCA (SOA)
Edit Copy Delete 1009 Mr. Sai Sandeepan Sahoo Cuttack Male sssahoo@cet.edu.in M.Tech
Check all With selected: Edit Copy Delete Export
Recent Favorites
New
information_schema
learningdb
class
evaluation
lecsub
lecturer
matcat
material
msgs
queries
student
subjects
users
mysql
newosms
performance_schema
phpmyadmin
test
localhost/phpmyadmin/index.php?route=/table/change&db=learningdb&table=lecturer
89%
08:29 AM
02-12-2021

Chapter-10: Conclusion

The Learning Management System has been computed successfully and was also tested successfully by taking "Test Cases". It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The Software is developed using HTML, CSS, JS as front end and JSP, Servlet and MySQL as back end in windows environment.

The following are the Limitations of this project:

- SMS alert facility is not available.
- Portal is not SEO friendly
- Registration Email Verification Not available
- Risk unauthorized accessibility

The goals that are achieved by the software are:

- Simplification of the operations
- Less processing time and getting required information
- User friendly
- Portable and flexible for further enhancement

Future Enhancement →

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printing option in future.
- We can give more advance software for Learning management system including more facilities.
- Integrate multiple load balancer to distribute the loads of the system.
- We can add Live Quiz exam module.
- We can add/integrate Attendance management system.
- We can enhance the query resolve system more efficiently.

The above mention points are the enhancements which can be done to increase the applicability and usage of this project. Enhancements can be done to maintain all the assignment, student, teacher, Quiz, Questions.

Chapter-11: References / Bibliography

The following reference has been used to develop the project "Learning Management System": -

Books: -

- Head First JAVA 2nd Edition
- Database programming with JDBC and JAVA by O'Reilly
- Google for problem solving
- Head First SQL: Your Brain on SQL by Lynn Beighley
- JAVA and Software design concepts by Apress

Web Source: -

- www.w3schools.com
- www.wikipedia.org
- www.javatpoint.com
- www.stackoverflow.com
- www.docs.microsoft.com
- www.tutorialspoint.com
- www.jsp.net
- www.javatpoint.com