

# Final Project

**Task management system**

Web Tech & Standards

School of Information Science

University of Pittsburgh

He Han

Contents

[Final Project 1](#_Toc374201337)

[Overview 3](#_Toc374201338)

[Statement of Problems 3](#_Toc374201339)

[Objectives 3](#_Toc374201340)

[Environment and tools 4](#_Toc374201341)

[Project structure 4](#_Toc374201342)

[Public mode 6](#_Toc374201343)

[Public tasks layer 6](#_Toc374201344)

[Administrator mode 9](#_Toc374201345)

[Existing tasks layer 10](#_Toc374201346)

[Archived tasks layer 12](#_Toc374201347)

[Conclusion 12](#_Toc374201348)

[References 13](#_Toc374201349)

[Codes 13](#_Toc374201350)

# Overview

This project is built for one group that needs some web-based social networking/collective intelligence sites. Assuming the group that will apply this project to their working environment consists of one manager and several employees, which is one to many structure. This project will provide tools for both manager and employees, which can not only improve the communication and work efficiency, but also provide a clear view of tasks for the group members.

# Statement of Problems

In one group, the manager may face the problems of confusions about what tasks and when a certain task he/she assigned to employees. The employees also will mess up tons of tasks, which result in unfinished or duplicate tasks. Most importantly, even the manager and employees communicate with each other using emails, it is not efficient to keep contact for the tasks and the tasks are also not well-ordered. By using social networking/collective intelligence sites, we can figure out one method that helps solve these problems.

# Objectives

To solve the above problems, this project has two main parts. One is built for the manager, the other is for the employees. Each part has different functions.

The basic functions for the manager’s site consist of:

1. Assign tasks to employees.

2. Show tasks that have not been done.

3. Archive tasks that have been completed.

4. Shared comments for tasks which can be seen by both manager and employees.

5. Make tags for tasks.

6. Show tasks based on employee’s name.

The basic functions for the employees’ site consist of:

1. Expose new tasks from manager or a certain employee.

2. Show tasks that have not been done.

3. Archive tasks that have been completed.

4. Shared comments for tasks which can be seen by both manager and employees.

5. Make tags for tasks.

6. Show tasks based on employee’s name.

# Environment and tools

All the functions will be accomplished using the web technologies, such as HTML, CSS, Javascript, JSP, XML, etc. MySQL serves as the database. Apache Tomcat serves as server.

There are two tables (task and arctask) in one database (taskdb), task table saves the existing tasks information, arctask saves the archived tasks information.

Source Server: localhost\_3307 Source Server Version: 50614

Source Host: localhost:3307 Source Database: taskdb

Target Server Type MYSQL Target Server Version: 50614

File Encoding: 65001

Admin Password (server): hehan

Manager Password (database): hehan

# Project structure

This web project mainly has two mode - public mode and administrator mode. Public mode is designed for both employees and manager which contains public tasks layer, while administrator mode is designed only for manager which consists of two layer, existing tasks layer and archived tasks layer. Public tasks and existing tasks share the same table in database while archived tasks are saved in another table in database.

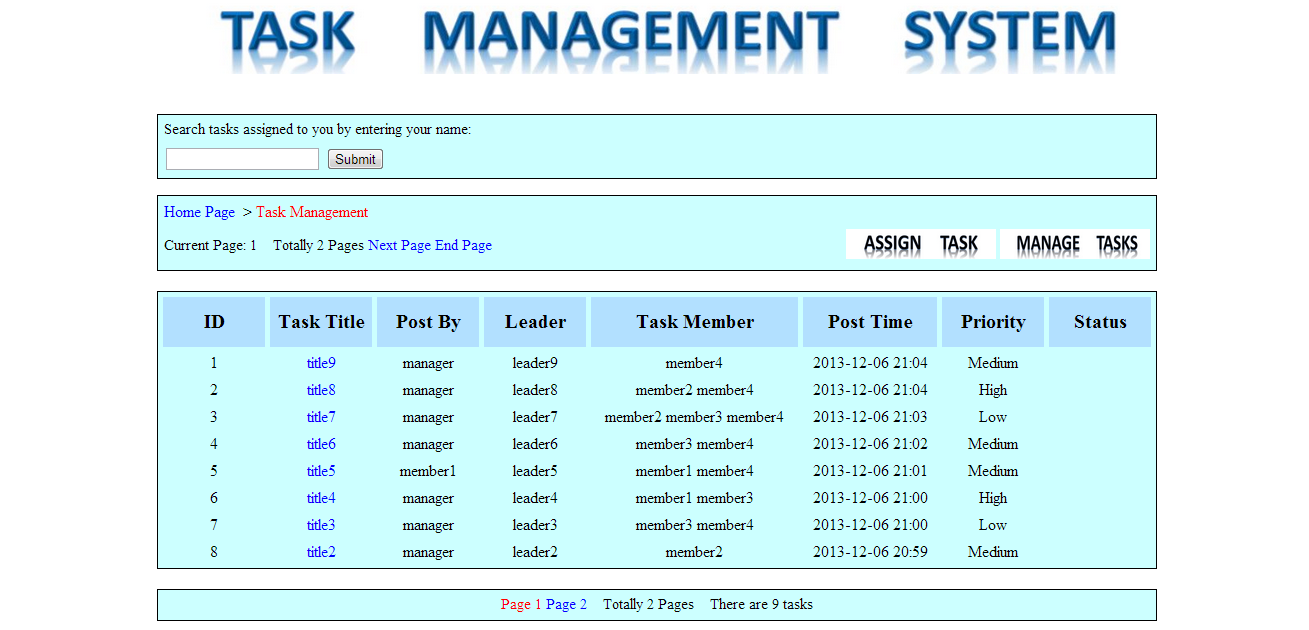
The users can separate these three layers by identifying the background color of webpage body. Public tasks layer has white background color, existing tasks layer has light green background color, and archived tasks layer has light orange color.

## Public mode

Public mode is developed for both manager and employees which contains public tasks layer. Anyone in this system can assign and check the existing tasks.

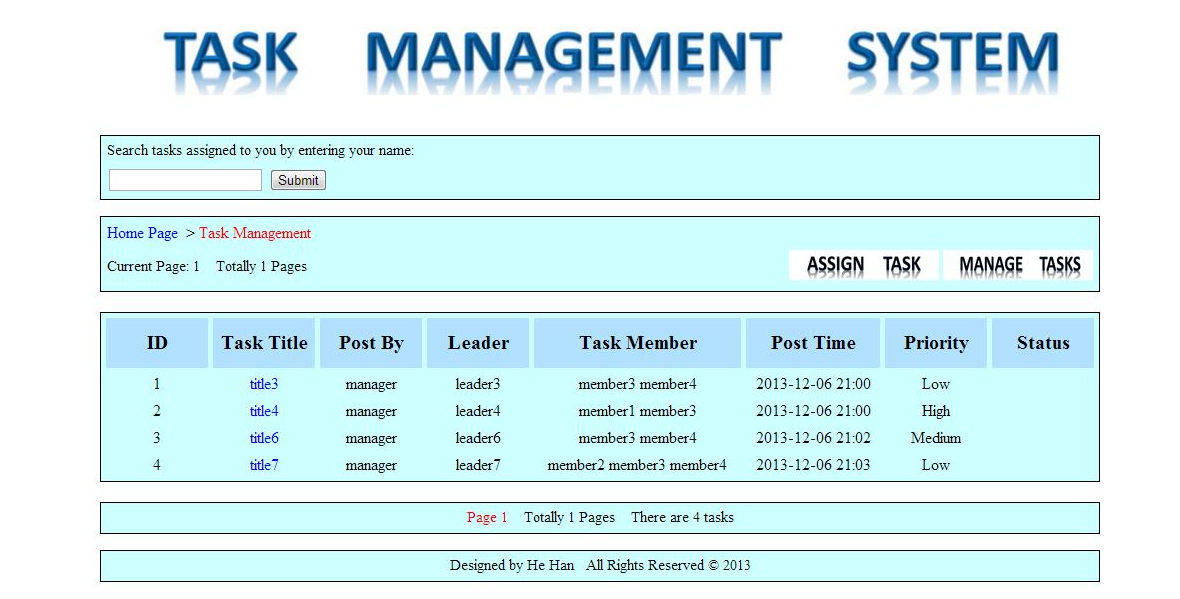
### Public tasks layer

There are four main functions in this mode - search function, assign task function, task table show function, detailed content show function.



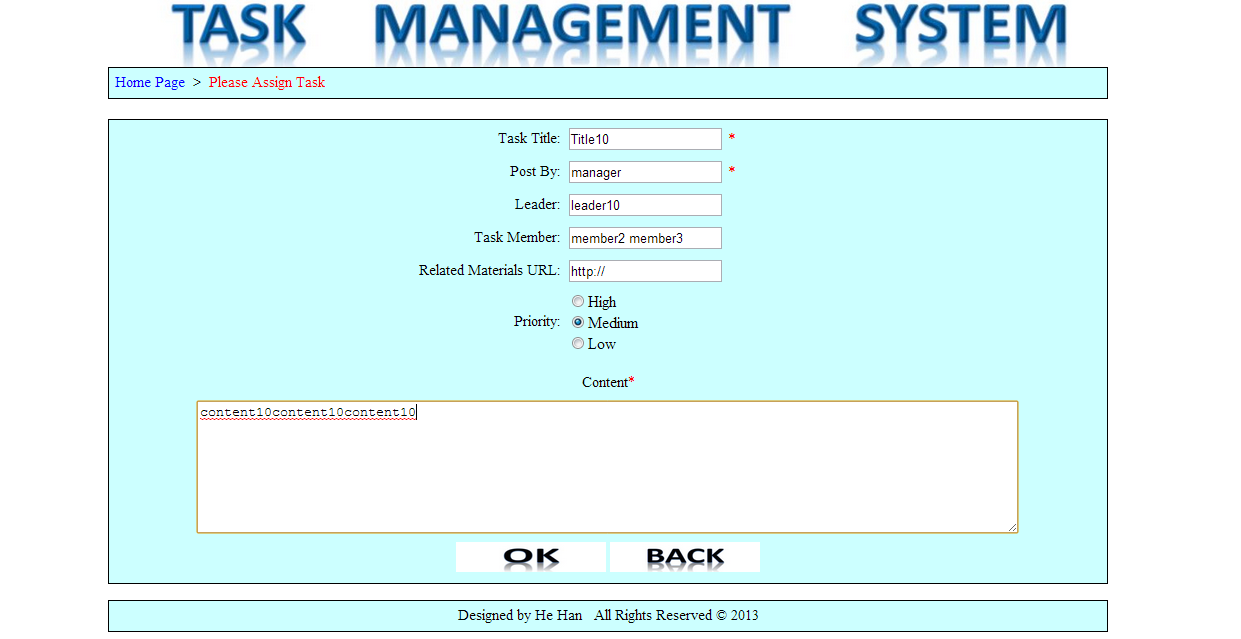
**Search function**

One can search his or her tasks by entering name into the search text. After submitting the name, the tasks will show in the table. For example, after entering name member3, the tasks that involve member3 will show in the table.

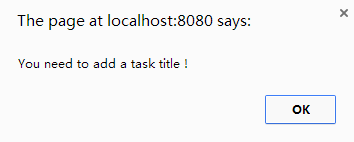


**Assign task function**

They can assign a task by clicking the button “ASSIGN TASK”. When entering assign task page, one can add task title, the person who posts this task, leader name, task members, related material URL, the priority of the task and the task content to the database.

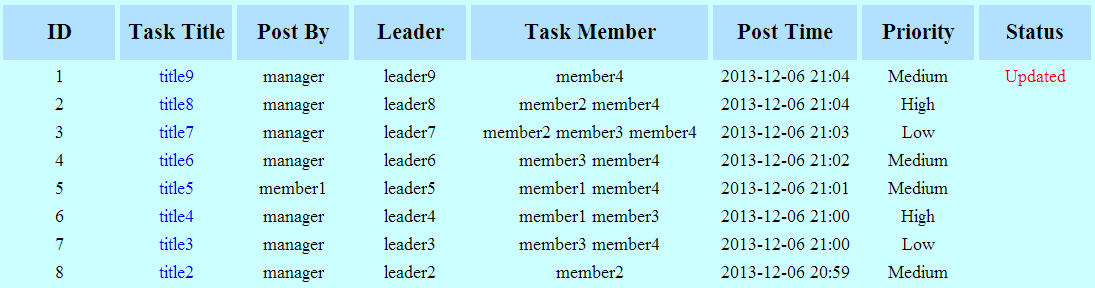


If the required field is not filled, the corresponding alert window will appear.



**Task table show function**

Task table shows 8 basic items of all the tasks, these 8 items consists of ID, task title, the person who posts this task, leader name, task members, post time, the priority of the task, and the task status. The status will provide the latest comment made by the manager. When the manager applies to one task, the status content will change to “Updated”.



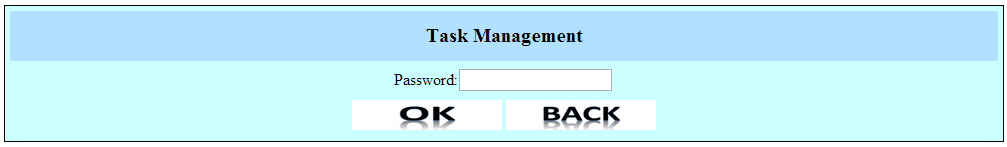
**Detailed content show function**

One can click the link of the task shown in the task title column.to check all content of this task, so that he or she can continue process this task according to the manager’s latest comment. The manager can reply this task my clicking the button “Manage Tasks”, then the manager will go to administrator mode and redirect to the detailed content show page.

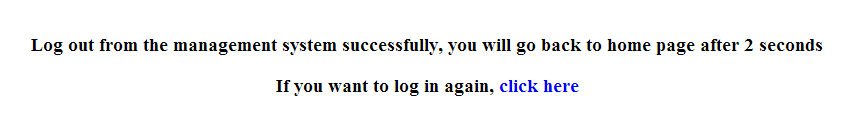


## Administrator mode

Administrator mode is designed only for manager. The manager has additional rights, such as deleting and making comments in terms of both existing tasks and archived tasks. Manager can manage tasks by logging into administrator mode. By clicking the button “Manage Tasks”, the manager will be directed to login page, after entering password, the manager will log in successfully.



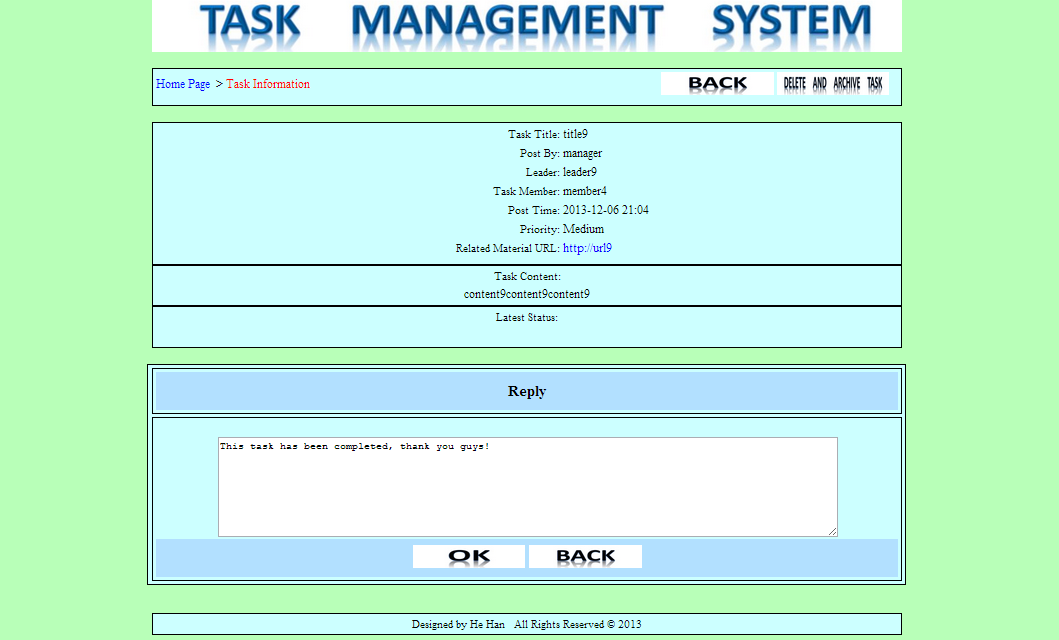
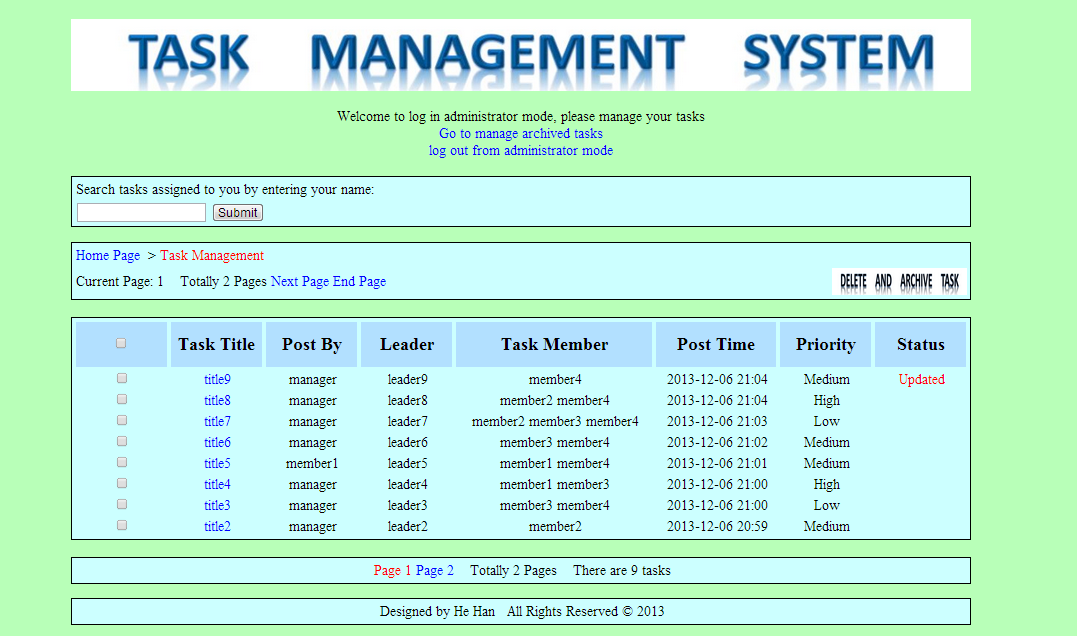
When the manager clicks the link “log out from administrator mode” in administrator mode pages, the manager will go back to public mode.



### Existing tasks layer

There are five main functions in administrator mode: search tasks function, make comments function, delete and archive task, existing tasks table show function, detailed content show function. The search, existing tasks table show and detailed content show functions are almost the same with those in public mode.

For make comments function, the manager can click the task title to enter the detailed content page, and then a reply text will appear below the task content. The manager can make comments by typing words in the reply text. After clicking OK button, the comments that manager make will saved in database and the status of the task will be changed to “Updated”.

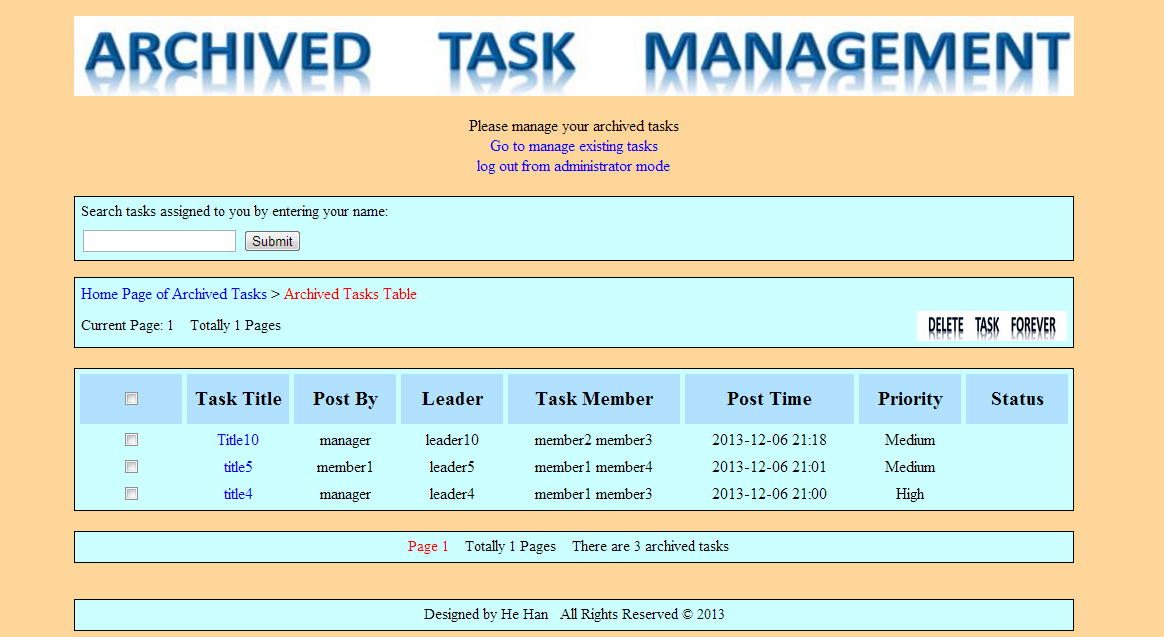
 

For delete and archive task function, one can firstly click the checkbox of the tasks and then click the button “delete and archive task”, the selected tasks will be deleted from the existing table and saved to archived table in the database. The manager can also firstly entering the detailed content of task by clicking the link of a certain task shown in the task title column, and then delete this task and save it to archived database for future analysis in case.

### Archived tasks layer

The manager can click “Go to manage archived tasks” to manage archived tasks. There are four main functions in this layer: search tasks function, delete task function, archived tasks table show function, detailed content show function. In this layer, the managed tasks are all archived tasks saved in archived database.

The difference between this layer and existing tasks layer is that the delete task function will delete the selected tasks from archived database forever.



# Conclusion

This project is practical in many areas, such as in company, research group, etc. By using these social networking sites, the group members can make a well-ordered schedule and keep track of what they do through the work process. This project can help support telecommute jobs, and working through Internet is now a very popular and promising work mode these years. Efficient, Practical, Interactive are major characteristics of this project.

# References

<http://tomcat.apache.org/>

[http://www.w3schools.com](http://www.w3schools.com/)

<http://www.mysql.com/>

Jon Duckett. Beginning HTML, XHTML, CSS, and JavaScript

# Codes

**Log in**

<%

String password="hehan";

String managerpw=null;managerpw=(String)request.getParameter("managerpw");

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

if(password.equals(managerpw)){

session.setAttribute("pass","y");

if(id!=null){response.sendRedirect("show.jsp?id="+id);}else{response.sendRedirect("index.jsp");}

}else{

%>

**Log out**

<%

response.setHeader("refresh","2;url=index.jsp");

session.invalidate();

%>

**Insert**

<%

String host="localhost:3307";

String user="root";

String pw="hehan";

String db="taskdb";

Class.forName("com.mysql.jdbc.Driver").newInstance();

String url="jdbc:mysql://"+host+"/"+db;

Connection con=DriverManager.getConnection(url,user,pw);

Statement st=con.createStatement();

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

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

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

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

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

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

String txt=request.getParameter("text");

if("http://".equals(web)){web="";}

String sql="insert into task values(0,'"+title+"','"+name+"','"+taskmember+"','"+web+"','"+leader+"','"+priority+"',now(),'"+txt+"','','n')";

st.executeUpdate(sql);

st.close();

con.close();

response.sendRedirect("index.jsp");

%>

**Add Task**

<table align="center">

<tr>

<td class="l"><div align="right">Task Title:</div></td>

<td><input name="title" type="text" id="title">

<span class="style1">\*</span></td>

</tr>

<tr>

<td class="l"><div align="right">Post By:</div></td>

<td><input name="name" type="text" id="name">

<span class="style1">\*</span></td>

</tr>

<tr>

<td class="l"><div align="right">Leader:</div></td>

<td><input name="leader" type="text" id="leader"></td>

</tr>

<tr>

<td class="l"><div align="right">Task Member:</div></td>

<td><input name="taskmember" type="text" id="taskmember" ></td>

</tr>

<tr>

<td class="l"><div align="right">Related Materials URL:</div></td>

<td><input name="web" type="text" id="web" value="http://"></td>

</tr>

<tr>

<td class="l"><div align="right">Priority:</div></td>

<td><input type="radio" name="priority" value="High">High<br>

<input type="radio" name="priority" value="Medium">Medium<br>

<input type="radio" name="priority" value="Low">Low<br>

</td>

</tr>

<tr>

<td height="8"></td>

</tr>

<tr>

<td colspan="2" class="l"><div align="center">Content<span class="style1">\*</span></div></td>

</tr>

<tr>

<td colspan="2"><div align="center">

<textarea name="text" cols="100" rows="8" id="text" overflow:auto;"></textarea>

</div></td>

</tr>

<tr>

<td colspan="2" align="center">

<input name="imageField" type="image" src="img/ok1.jpg" width="150" height="30" onMouseOut="this.src = 'img/ok1.jpg';" onMouseOver="this.src = 'img/ok2.jpg';" alt="ok"/>

<a href="index.jsp"><img src="img/back1.jpg" onMouseOver="this.src = 'img/back2.jpg';" onMouseOut="this.src = 'img/back1.jpg';" alt="back"/></a></td>

</tr>

</table>

**Delete Task**

<%

String host="localhost:3307";

String user="root";

String pw="hehan";

String db="taskdb";

String pass=null;pass=(String)session.getAttribute("pass");

if("y".equals(pass)){

Class.forName("com.mysql.jdbc.Driver").newInstance();

String url="jdbc:mysql://"+host+"/"+db;

Connection con=DriverManager.getConnection(url,user,pw);

Statement st=con.createStatement();

String sql;

String arcsql;

int tablerow,p,no;

String rw=(String)request.getParameter("tablerow");

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

if(rw==null){tablerow=1;}else{tablerow=Integer.parseInt(rw);}

if(pages==null){p=1;}else{p=Integer.parseInt(pages);}

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

no=i+(p-1)\*tablerow;

String del=request.getParameter("del"+no);

if(del!=null){

arcsql="insert into arctask select \* from task where id="+del;

sql="delete from task where id="+del;

st.executeUpdate(arcsql);

st.executeUpdate(sql);

}

}

st.close();

con.close();

response.sendRedirect("index.jsp");

}else{

response.sendRedirect("login.jsp");

}

%>

**Show detailed content**

<table align="center">

<tr>

<td class="l"><div align="right">Task Title:</div></td>

<td><span class="style2"><%=taskdb.getString(2)%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Post By:</div></td>

<td><span class="style2"><%=taskdb.getString(3)%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Leader:</div></td>

<td><span class="style2"><%=taskdb.getString(6)%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Task Member:</div></td>

<td><span class="style2"><%=taskdb.getString(4)%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Post Time:</div></td>

<td><span class="style2"><% String time = taskdb.getString(8);

out.print(time.substring(0, 16));%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Priority:</div></td>

<td><span class="style2"><%=taskdb.getString(7)%></span></td>

</tr>

<tr>

<td class="l"><div align="right">Related Material URL:</div></td>

<td><span class="style2"><% if (taskdb.getString(5).compareTo("") != 0) {

out.print("<a href=" + taskdb.getString(5) + " target=\_blank>" + taskdb.getString(5) + "</a>");

}%></span></td>

</tr>

<tr>

<td height="2"></td>

</tr>

</table>

<table align="center" cellspacing="5">

<tr> <td align="center" class="l"><div align="center">Task Content:</div></td></tr>

<tr>

<td align="center"> <%=txthtml(taskdb.getString(9))%>

</td>

</tr>

</table>

<table align="center" cellspacing="5">

<tr> <td align="center" class="l"><div align="center">Latest Status:</div></td></tr>

<tr>

<td align="center"> <%=retxthtml(taskdb.getString(10))%>

</td>

</tr>

<tr>

<td height="2"></td>

</tr>

</table>

**Check from**

<script language="JavaScript" type="text/JavaScript">

function checkform(formcontent)

{

if (formcontent.text.value=="")

{

alert("You need to add a task！");

formcontent.text.focus();

return false;

}

if (formcontent.title.value=="")

{

alert("You need to add a title");

formcontent.title.focus();

return false;

}

if (formcontent.name.value=="")

{

alert("You need to add your name");

formcontent.name.focus();

return false;

}

}

</script>

**MySQL**

CREATE TABLE `task` (

`id` int(10) NOT NULL AUTO\_INCREMENT,

`title` varchar(50) NOT NULL,

`name` varchar(20) DEFAULT NULL,

`taskmember` varchar(100) DEFAULT NULL,

`web` varchar(100) DEFAULT NULL,

`leader` varchar(20) DEFAULT NULL,

`priority` varchar(20) DEFAULT NULL,

`time` datetime DEFAULT NULL,

`text` text NOT NULL,

`retxt` text,

`status` char(2) NOT NULL DEFAULT 'n',

PRIMARY KEY (`id`)

)