

CHAPTER 1

INTRODUCTION

TRAINING MANAGEMENT SYSTEM(TMS) is a system used by training providers and learning professionals to organize training delivery, curriculums, student records and communication. TMS is to maintain the training data of new hires in their Retail Banking department of C-PayServ, which is currently a manual process. It will help automating following processes:

1. Maintaining the personal information of trainees
2. Capture trainee evaluations by trainers

The Application, with welcome Home page for the designated users, will allow three types of user administrator, trainer and trainee – log on to the portal with a set of tasks assigned to each user.

The following modules, clearly explains how **WEB APPLICATION FOR TRAINING MANAGEMENT SYSTEM** developed to meet the various requirements of the company.

1.1 EXISTING SYSTEM

In existing system, managed data are stored in particular database of a computer. Those data can't be accessed from anywhere. The data can't be changed easily. The trainer and trainee details are stored in paper records. It takes more time to rectify and modify the corrections in the data.

1.2 OBJECTIVE

Training management System application makes the process simple. It is effective and to provide reliable, transparent system to devoid of personal inclination and interest. To alleviate

the problems and stress. The database is updated on regular basis and if required, data manipulation is also possible.

1.3 SCOPE OF THE SYSTEM

The TMS for C-PayServ contains Two Major Modules i.e.

Training Module

- View trainee's personal information
- Enter trainer feedback comments for a particular course
- Enter evaluation details for trainee to each course attended in training phase

The system will have three types of users –

1.Trainer:

This user has the role of Trainer. This user can perform following operations

- View trainees detail
- Enter trainee's evaluation details for all courses attended in each phase

2.Trainee:

This user has the role of Trainee. This user can view

- Personal details
- Trainer evaluation for a particular course

3.Administrator:

This user has the role of Administrator. This user can perform following operations

- Add trainee
- Add trainer

1.4 SYSTEM PERSPECTIVE

This is an independent system and should be implemented using MVC architecture.

1.5 SYSTEM ENVIRONMENT

The Training Management System web site will be operated from C-PayServ's centralised high end server with parallel processor support. When a user connects to the Web Server, the Web Server will interact with the Database after processing the bussines logic to transfer data to and from the database.

- IBM Web Sphere (J2EE) /JBoss/Weblogic Server for application deployment.
- Oracle database for data storage

1.6 IMPACT OF THE SYSTEM

This is a new product, which is developed for Training department of C-PayServ. Expected impact of the product is to automate existing manual processes in order to make them more efficient and cost effective.

CHAPTER 2

GENERAL DESCRIPTION

2.1 PROJECT PERSPECTIVE

The main perspective of the project is to reduce the paper work, data retrieval through automation. The admin can upload the trainer details and trainee details into the database and the data can be viewed in this application.

2.2 USER CHARACTERISTICS

In this application, admin only have the authority to enter details of the trainer and trainee. The trainer has the authority to view the trainee's details and enter the trainee's evaluation details for all the course attended in each phase. The trainee has the authority to view the personal details and trainer evaluation for particular course. The application should be user friendly; hence the user should not need any software and hardware knowledge.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

The application is designed for TMS is to make the data entry and manipulation process simple, this application is portable and then the relation should satisfy the key constraints, domains, referential constraints and integrity constraints.

- MS SQL Server will be used as SQL engine and database
- The Online System is running 24 hours a day
- Users must have their correct usernames and passwords to enter into their online accounts

2.4 PROPOSED SYSTEM

The proposed system is a web application, which is completely related to automation. The admin can enter the trainer and trainee details. The trainer has the authority to view the trainee's details and enter the trainee's evaluation details for all the course attended in each phase. The trainee has the authority to view the personal details and trainer evaluation for particular course.

ADVANTAGES

- Saves users time.
- portable
- Security -Login id's are provided with so that they can have access the information securly.
- Data integrity

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

3.1 FUNCTIONAL REQUIREMENTS

The requirements specification is a technical requirement for the software products. It is the first step in the requirements analysis process. The requirements also provide usage scenarios from user, operational and administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target admin, trainer and trainee and its user interface, hardware and software requirements. It defines how the admin, trainer and trainer see the project and its functionality. The Functional Requirements of Training Management System are as follows:

3.1.1 Admin Module

3.1.1.1 Login/Validation Page

When the user clicks start, the welcome page appears which will have a short welcome message, a brief introduction of the purpose of this site, some links for any users, which give useful information about trainings and Input fields for User Name and Password. The input fields for User Name and Password are Text fields. The entered characters in the Password field must be masked with an * symbol. There should be a Submit button. User Name and password are mandatory for logging in to the site.

User Interface Requirements:

1. The User Name and Password fields should be the text boxes.
2. The entered characters in the Password field must be masked with an * symbol.
3. There should be a submit button.

3.1.1.2 Trainee Registration Page

Administrator accesses this module. Administrator will be able to add trainee information. The Trainee Information Page will have following fields to capture trainee information.

- Trainee ID or Associate ID
- Password
- First name
- Last name
- Hire date
- Designation
- Vertical allocated
- Location
- Phone
- Role (As Trainee)
- Trainer

User Interface Requirements:

1. All the fields are Mandatory and must be entered as specified
2. The Page header will be “Registration User Details”
3. Once user select role as trainee. Trainer field should be enabled to allocate trainer.

3.1.1.3 Trainer Registration Page

Administrator will have rights to access this page. Administrator will be able to add trainer information. The Trainer Details Page will have following fields to capture trainer information.

- Trainer ID or Associate ID
- Password
- First name
- Last name
- Hire date
- Designation

- Technology Competency or vertical
- Location
- Phone
- Role (As Trainer)

User Interface Requirements:

1. The Page header should be “Registration User Details”
2. All the fields on the page are mandatory.

3.1.1.4 Activity Scheduling

Administrator accesses this module. Administrator will be able view all the trainee details and training phases completed by each trainee with the feedback provided by the trainer.

- Phase ID
- Courses Name
- Ratings
- Evaluation comments

User Interface Requirements:

1. The Page header will be “Dashboard Page”.
2. By clicking on the trainee name link administrator should able to view completed details of trainee
3. By clicking on each phase completed by trainee, indicated by green bar administrator should able to view evaluation comments provided by the trainer.

3.1.1.5 Logout

This is used to log out of the system.

User Interface Requirements:

There is a link named “Logout “on each page. By simply clicking logout, the user will able to logout from the system.

3.1.2 Trainer Module

3.1.2.1 Trainer Dashboard Page

Trainer should be able to provide evaluation rating for each course completed by the trainee like hands-on, assessment along with evaluation comments by click on particular training phase. Trainer should be able to activate (promote) trainee to move to next phase of training to take up course mapped in next phase. Page will accept following information to register trainee for the particular course and allow administrator to activate trainee.

- Course name
- Hands-on status like completed or not completed
- Assessment Rating/Scores
- Evaluation Comments

User Interface Requirements:

1. All the fields are mandatory.
2. The Page header will be “Trainer Dashboard Page”.

3.1.3 Trainee Module

3.1.3.1 Trainee’s Dashboard Page

Trainee should be able to view his/her evaluation comments provided by the trainer for each course completed in training phase & also trainee can check the status whether he/she is promoted to next phase of the training by the trainer.

Trainee should be able to view following evaluation details provided by trainer

- Course name
- Hands-on status like completed or not completed
- Assessment Rating/Scores
- Evaluation Comments

User Interface Requirements:

1.The Page header will be “Trainee’s Dash board Page ”

3.1.4 REPORTS

Only the administrator will able to see brief reports by using the search provision based on

- Details by Trainees
- Trainee’s training phase completed report
- Trainee’s evaluation details received from trainers

User Interface Requirements:

The Page header will be “Training Management System Dashboard Reports”.

3.2 NON-FUNCTIONAL REQUIREMENTS**3.2.1 UI Requirements:**

- The front end should be user-friendly and pleasant
- Any error message or exception displayed to the user should be user-readable (and not technical)
- All entered values should be validated

3.2.2 Installation, Deployment & Operational Requirements:

- Application needs Eclipse, Apache Tomcat,GSON and mysql connector.

3.2.3 Reliability

- Any server failure should cause no more than 30-minute downtime, with the average downtime not exceeding 15 minutes

3.2.4 Maintainability & Portability Requirements:

- If the allocated memory space for the existing website needs to be increased then it should be possible without any impact to performance

3.2.5 Security Requirements :

1. Security should be through a security table with encrypted passwords
2. All transactions (updating data or otherwise) in the application should be audit logged for future reference
3. No unauthorized users should be able to log on to the system
4. None of the pages are reachable without a log on id having been entered and authenticated
5. Passwords must be at least 8 characters and have a combination of letters, numbers and special characters (#,@,\$,%,&)
6. 3 incorrect password entries must de-activate the user

3.2.6 Online User Documentation And Help System Requirements:

1. Online help engine accessible from every screen
2. Help provided on each screen of the application

3.2.7 Training Requirements :

1. Training for intended users on the operation of the new software is within scope

3.3.USER INTERFACE

3.3.1 User Interfaces

- Front-end software: Browser
- Scripting Languages: Front-end: JSP, HTML, Bootstrap

- Technologies: Servlet
- Back-end software: MySQL

3.3.2 Hardware Interfaces

- Processor : Intel(R) Core(TM)i5
- RAM : 4 GB
- Memory : 1 TB

3.3.3 Software Interfaces

Following are the software used for the student information management online application.

SFTWARE USED	DESCRIPTION
Operating system	Windows operating system is chosen for its best support and user-friendliness.
Server	Application needs Apache Tomcat server for deployment with MySQL 5.5 database for persistence
Database	SQLs database is chosen to save the customer records.

Table:1 Software interfaces

3.3.4 Software And Programs Used

The programs and software used were programming languages such as Java, frameworks such as Servlets, development platform such as Eclipse, script language such as JavaScript and bootstrap, markup language such as HTML and database program such as MYSQL, among others. Proper installation of all the programs used were required and basic understanding of the programs were utilized in the completion of the project.

3.3.4.1 Java

Java is a set of technologies (programming language and computing platform) for creating and running software. Java is a popular general-purpose programming language and computing platform. It is fast, reliable, and secure. According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide. Java was built with the philosophy of "write once, run anywhere" (WORA). The Java code (pure Java code and libraries) you write on one platform (operating system) will run on other platforms with no modification. To run Java, an abstract machine called Java Virtual Machine (JVM) is used. The JVM executes the Java bytecode. Then, the CPU executes the JVM. Since all JVMs work exactly the same, the same code works on other operating systems as well, making Java platform-independent. There are different styles of programming. Object-oriented approach is one of the popular programming styles. In object-oriented programming, a complex problem is divided into smaller sets by creating objects. This makes your code reusable, has design benefits, and makes code easier to maintain.

The Java platform provides various features for security of Java applications. Some of the high-level features that Java handles are:

- provides secure platform for developing and running applications
- automatic memory management, reduces memory corruption and vulnerabilities
- provides secure communication by protecting the integrity and privacy of data transmitted.

3.3.4.2 Servlets

Today we all are aware of the need of creating dynamic web pages i.e the ones which have the capability to change the site contents according to the time or are able to generate the

contents according to the request received by the client. If you like coding in Java, then you will be happy to know that using Java there also exists a way to generate dynamic web pages and that way is Java Servlet. But before we move forward with our topic let's first understand the need for server-side extensions. Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server. Servlets work on the server-side. Servlets capable of handling complex request obtained from web server.

1. The clients send the request to the web server.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generate the response in the form of output.
5. The servlet send the response back to the web server.
6. The web server sends the response back to the client and the client browser displays it on the screen.

3.3.4.3 Hypertext Markup Language

HyperText Markup Language (HTML) is a popular markup language used in web pages. HTML can be simply written in a text editor and tested through web browser. Writing in html is easy; with html it's also possible to add media and images to the web page. HTML contains special markup tags like <title>, <h>, <p> etc. To declare the title of the page for example, the title has to be included in the title tags. Similarly, the paragraphs, headings and other different contents in website pages have to be included inside the respective HTML tags. It is easy to store HTML code; in a simple text file with filename followed by .html or .htm extension. HTML gives developers the possibility of creating sections in the document. As <title> tag gives the title for the web page, <H1> for example specifies the main content of the web page. Similarly, with H2, H3 and so on tags, HTML creates minor contents. There are tags for other features such as paragraphs (<p>), style of font (bold) and tables (<table>) etc.

3.3.4.4 Javascript

Javascript is one of the most popular script languages for webpages today. Javascript (JS) is an object-oriented language that supports features such as imperative and functional programming. The syntax is similar to other object-oriented languages such as Java and C++, hence making it easy to learn for people who know these languages. JS is used to add interactive features such as buttons, animation, games etc. JS was invented by Brendan Eich. Javascript is capable of creating many different features ranging from beginner to advanced such as 2D and 3D features on a website.

3.3.4.5 Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery. Bootstrap employs a handful of important global styles and settings that you'll need to be aware of when using it, all of which are almost exclusively geared towards the *normalization* of cross browser styles.

3.3.4.6 Cascading Style Sheet (CSS)

CSS is a language that was developed in 1992-1993. CSS helps design each and every element of the markup elements language such as HTML by giving complete control to the designer. While the HTML elements enables the web page designers to add what content they want, CSS makes it possible how to display the content to the user. CSS covers the areas such as colors, layout, advanced positions of elements, fonts and also allows the content to adapt the content to different devices such as phones, tablets, biggerscreens and printers. CSS can operate independently as well as be used with any markup languages based in XML. CSS uses simple, everyday English words and has an easy syntax. CSS is crucial in advanced web designing as it gives control to the layout and offers numerous techniques to make the web page look sophisticated. Currently, the basic features of CSS are supported by all main browsers such as Internet Explorer, Safari, 10Opera, Chrome and Firefox. CSS has been used in the web pages for

the development of sample online booking system for this thesis for positioning, layout, margins and colors for the HTML elements.

3.3.4.7 SQLite Database

Databases are collections of similar data. Databases are used for organized collection and storing of similar data, to be later used for specific purposes. A database contains tables with rows and columns populated with objects, which displays connection between them. A database acts as a shared resource for the programs which can use the information from the database. Many enterprises rely on databases today to store a wide range of information systematically. Databases are used almost everywhere: in small companies which can use database to save customers' information and also for more advanced scientific and military areas. Databases facilitate the task of searching hundreds and thousands of records much simpler by storing them in an organized manner. Among many database programs available today, SQLite Database has been used in the development work.

CHAPTER 4

DETAILED DESIGN

4.1 BUSINESS USECASE DESIGN.

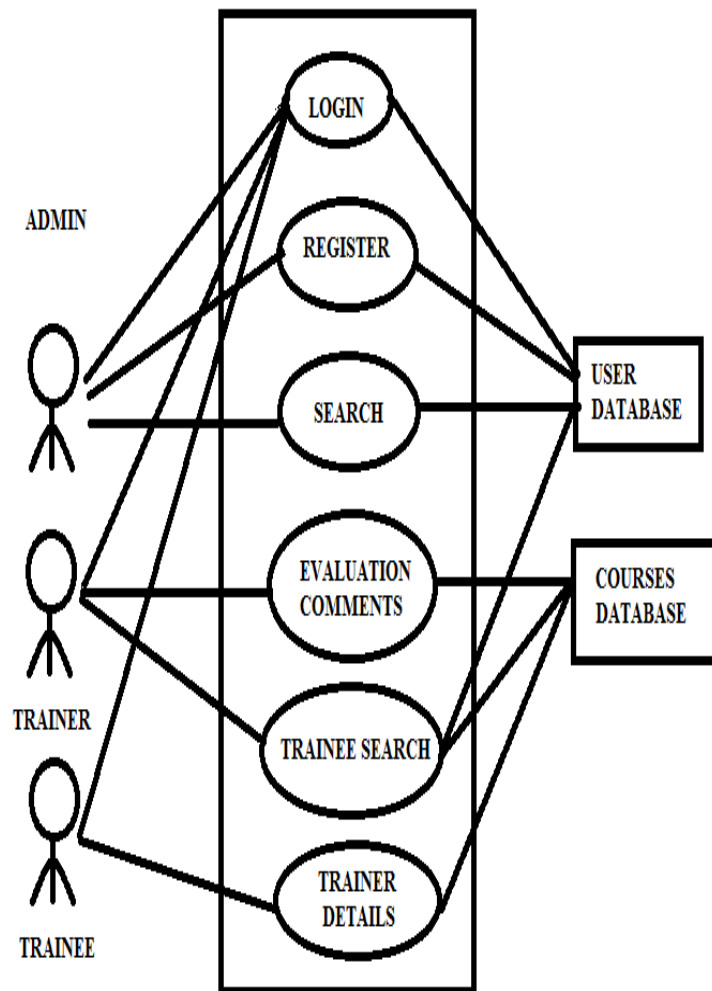


FIG 1:Business usecase diagram

4.2 DATAFLOW DIAGRAM

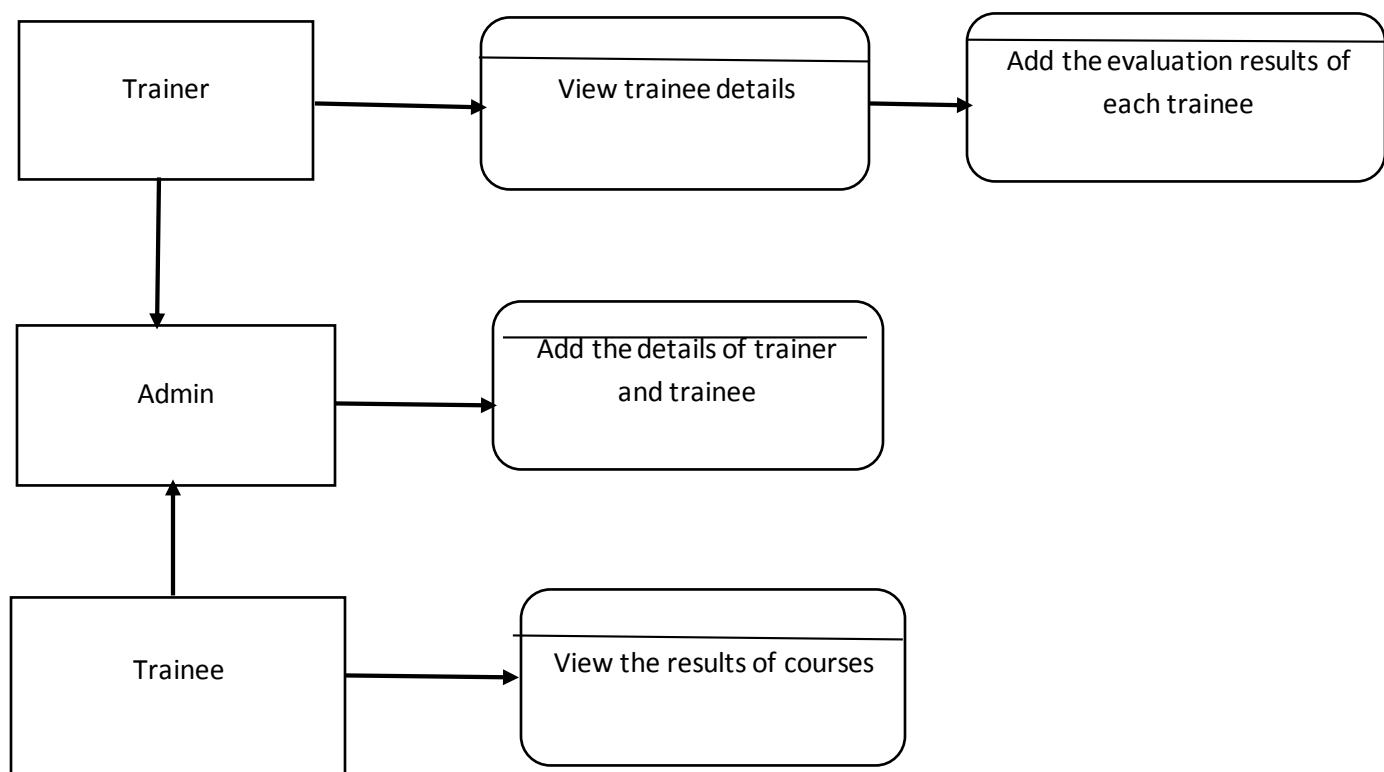


FIG 2: Dataflow diagram

4.3 DATABASE DESIGN

This project follows relational database model. The relational model represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship. The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

Some popular Relational Database management systems are:

- DB2 and Informix Dynamic Server - IBM
- Oracle and RDB – Oracle
- SQL Server and Access - Microsoft

In our project the database consists of 2 tables such as users and courses. The patient acts as the major table in our database.

4.3.1 ER Diagram

An ER diagram is a diagram that helps to design database in efficient way. Attributes in ER diagrams are usually modelled as an oval with the name of the attribute, linked to entity or relationship that contains the attribute.

Within the relation model the final step can generally be broken down into two further steps, that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects.

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other. ER diagrams are most often associated with complex databases that are used in software engineering and IT networks. In

particular, ER diagrams are frequently used during the design stage of a development process in order to identify different system elements and their relationships with each other. For example, inventory software used in a retail shop will have a database that monitors elements such as purchases, item, item type, item source and item price.

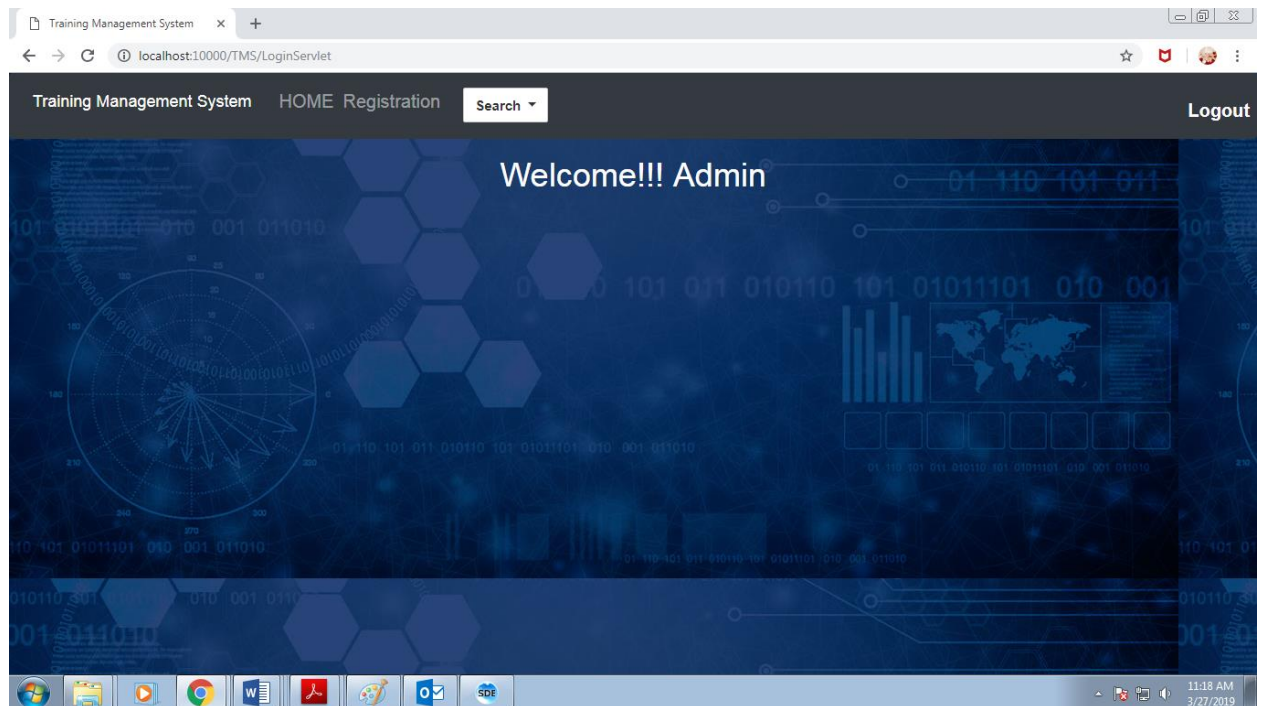
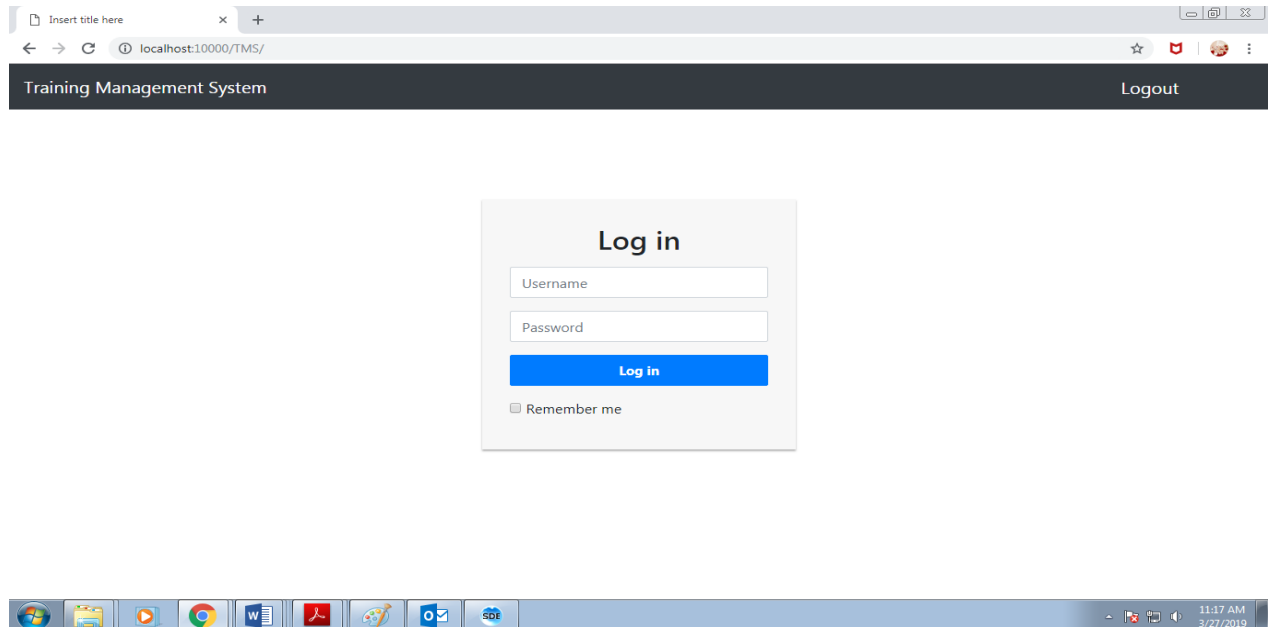
4.3.2 Database Manipulation

Data Manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases. The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

- **SELECT:** This command is used to retrieve rows from a table. The syntax is `SELECT [column name(s)] from [table name] where [conditions]`. SELECT is the most widely used DML command in SQL.
- **UPDATE:** This command modifies data of one or more records. An update command syntax is `UPDATE [table name] SET [column name = value] where [condition]`.
- **INSERT:** This command adds one or more records to a database table. The insert command syntax is `INSERT INTO [table name] [column(s)] VALUES [value(s)]`.
- **DELETE:** This command removes one or more records from a table according to specified conditions. Delete command syntax is `DELETE FROM [table name] where [condition]`.

4.4 OUTPUT DESIGN

Screenshots



Training Management System x +

localhost:10000/TMS/LoginServlet

Training Management System HOME Registration Search Logout

Registration Form

Associate id :

Password :

First Name :

Last Name :

Hire Date :

Vertical :

Designation :

Location :

Phone :

Role : Trainer Name:

11:18 AM 3/27/2019

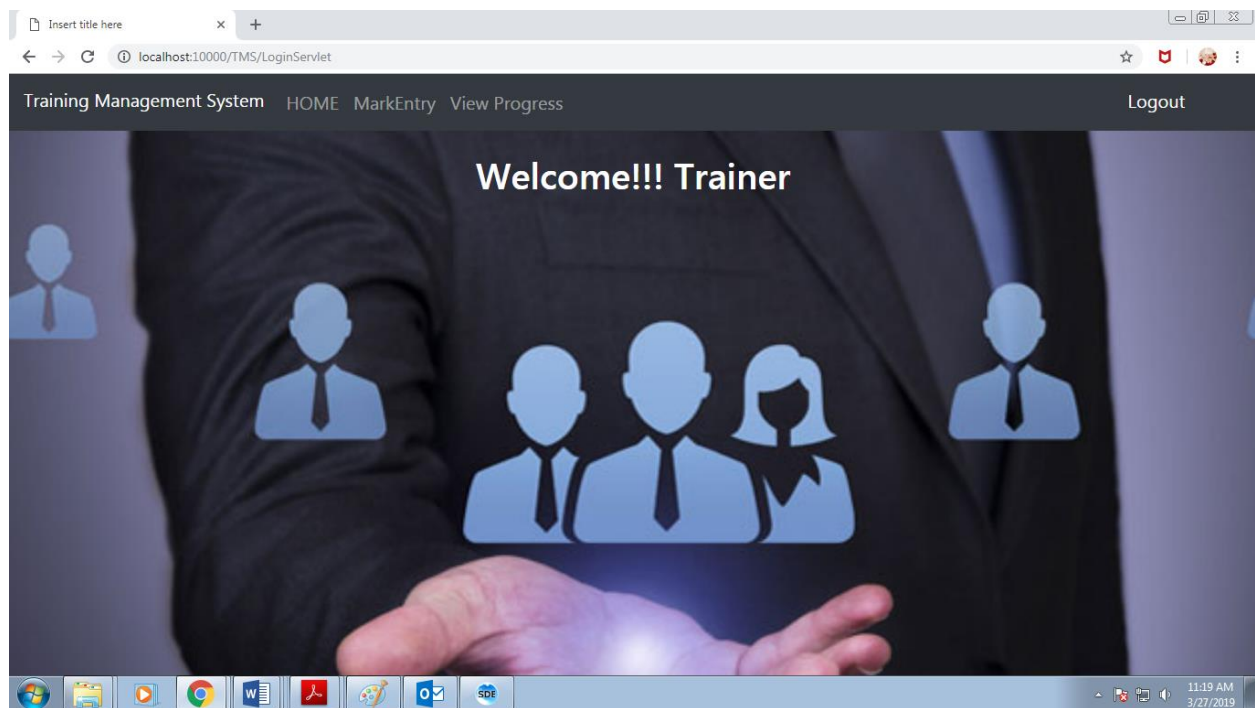
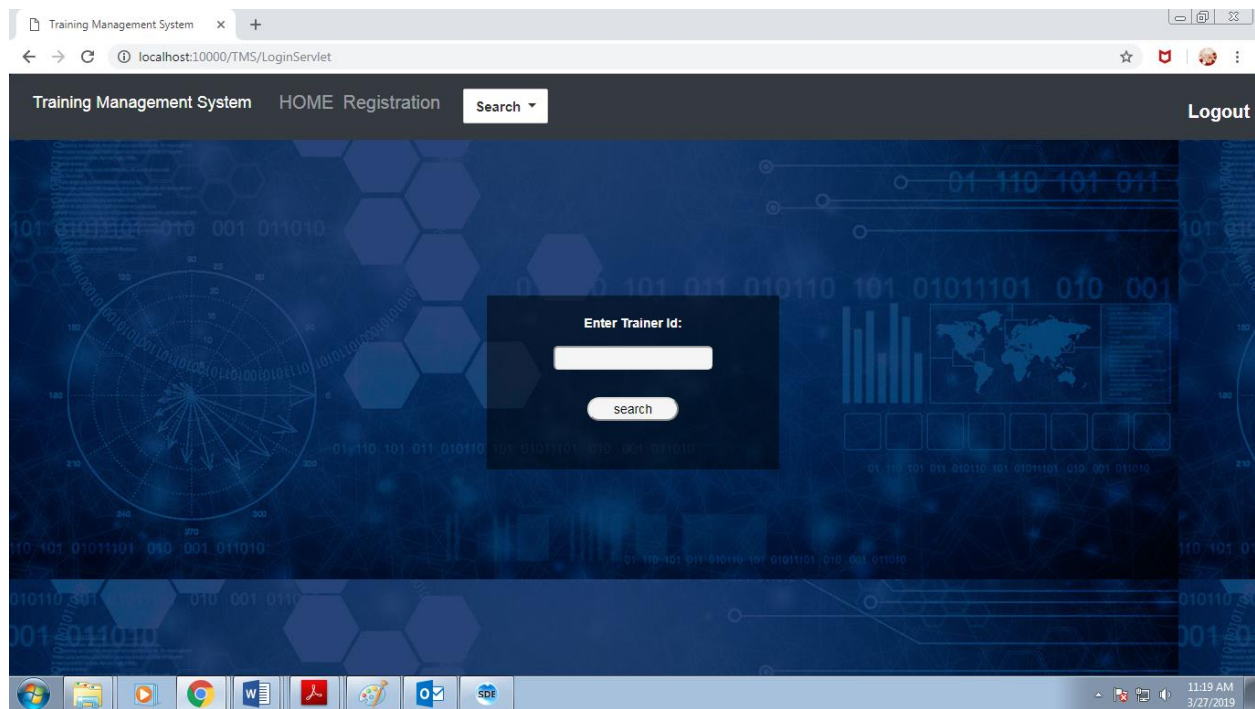
Training Management System x +

localhost:10000/TMS/LoginServlet

Training Management System HOME Registration Search Logout

Enter Trainee Id:

11:19 AM 3/27/2019



Insert title here

localhost:10000/TMS/LoginServlet

Training Management System HOME MarkEntry View Progress Logout

COURSE ENTRY

Trainee id :

Trainee Name :

Trainer Name :

Skill Type : Choose

Course_name :

Hands on status : Choose

Assessment scores :

Evaluation comments :

submit

11:20 AM 3/27/2019

Insert title here

localhost:10000/TMS/LoginServlet

Training Management System HOME MarkEntry View Progress Logout

Progress Details

Trainee Id:

Skill Type: Choose

submit

11:20 AM 3/27/2019

TMS

localhost:10000/TMS/LoginServlet

LOG OUT

Details of 15tne01

Course Name	Hands On Status	Assessment Scores	Evaluation Comments
Introduction	completed	60	Good
Self actualisation	completed	80	Good
Time Management	completed	70	NA
Self-Esteem&Leadership	completed	100	NA
Team Work	completed	80	NA
Word Processing	completed	100	NA
Advanced Excel Formulas	Not-completed	0	NA

11:20 AM
3/27/2019

CHAPTER 5

5. RESULTS AND DISCUSSION

Thus the project has been implemented for the welfare of trainees. The details have been stored in the database for future use of the trainer and trainee. This application is helpful for the trainer to store, update and retrieve trainee's academic details with some valid personal information and to generate profile. It is easily accessible to all trainees and also maintains trainee's privacy as only with correct username and password can login into the account. Only the particular trainee can able to view their evaluation status and personal information. Trainer can search for the particular trainee and trainer can give evaluation comments for the assessment. Basically this system helps trainee and trainer as a self-assistant to make their work of maintaining their self records easier. This system works 24*7 because it is an online existence.

CHAPTER 6

TESTING

6.1 UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated. In this process, each module (such as admin, trainer etc.) are considered and individual units are tested for proper operation. If each module meets up with the user's requirement, then it is subjected to integration testing where more than one module are integrated and tested.

6.2 REGRESSION TESTING

This testing is carried out to ensure that the later developed code will not affect the features of the earlier developed module. For example, the course evaluation details that is to be calculated should be accurate. If any error may reflect in the calculations due to the generation, then the code must be changed accordingly. Each module must be tested to produce a satisfactory outcome.

6.3 VERIFICATION

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements. When developing each module, the individuality of the product is checked at its development stage. The entry of each details of the trainer and trainee is carried out and verified whether they provide accurate calculations. Thus the modules must be verified at the development stage.

6.4 VALIDATION

Validation ensures that the developed product will meet the user requirements. It enables the user to flexible is use the environment and satisfies their requirements. Validating the project during the development stage is efficient than validating during completion of the product. The proposed system will work according to the system requirements.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This Training Management System assists in automating the existing manual system. This is a paperless work. It can be monitored and controlled remotely. This system loosens their efforts by providing a web application which provides a user-friendly interface to both trainer and trainee to update and maintain their own information right from their personal information to skill set. The gathered information can be saved and can be accessed at any time. Data are stored in a systematic way. Overall, efficiency has improved and work processes simplified. It reduces the time taken by the user to add, update and view profile. Admin can able to maintain the both the trainer and trainee details. Future Scope includes Online video tutorials in specific domains can be provided. Discussion forums can be added. Online class functionality can be added. Functionality of chats and messages can be added.

APPENDIX

Code:

Trainer.Jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"% >
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>
$(window).load(function() {
$("div.home").show();
$("div.mark").hide();
$("div.progress").hide();});
$(document).ready(function() {
$("#first").click(function() {
$("div.home").show();
$("div.mark").hide();
```

```

$("div.progress").hide();});
$("#second").click(function() {
    $("div.home").hide();
    $("div.mark").show();
    $("div.progress").hide();
});
$("#third").click(function() {
    $("div.home").hide();
    $("div.mark").hide();
    $("div.progress").show();
});});
</script>
</head>
<body>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
<ul class="navbar-nav">
<li> <a class="navbar-brand" href="#">Training Management System</a></li>
<li class="nav-item">
<a class="nav-link" id="first">HOME</a></li>
<li class="nav-item">
<a class="nav-link" id="second" >MarkEntry</a></li>
<li class="nav-item">
<a class="nav-link" id="third" >View Progress</a></li>
<li id="logout">
<a href="index.jsp">Logout</a></li></ul></nav>
<div class="home">
<center><br/><h1>Welcome!!! Trainer</h1></center></div>
<div class="mark">
<_nav><br/><br/>
<center> <form action="./course" id="form" method="post">

```

```

<div class="trans">
<h1>COURSE ENTRY</h1><label>
<b>Trainee id :</b> <input type="text" name="p_uid" class="round1" size="30" required />
</label><br /><label>
<b> Trainee Name :</b><input type="text" name="name" class="round1" size="30" required>
</label><br /><label>
<b>Trainer Name :</b><input type="text" name="trainer_name" class="round1" size="30"
required>
</label><br />
<label> <b>Skill Type :</b><select name="skill_type" class="round1" required>
<option value="">Choose</option>
<option value="Personality_development">Personality_development</option>
<option value="Basic_IT">Basic Information Technology</option>
<option value="Tech_in_bank">Technology Elements In Banking</option>
</select></label><br/><label>
<b> Course_name :</b> <input type="text" name="course_name" class="round1" size="30"
required>
</label><br />
<label>
<b> Hands on status :</b><select name="Hands_on_status" class="round1" required>
<option value="">Choose</option>
<option value="completed">completed</option>
<option value="Not-completed">Not-completed</option>
<option value="In-Progress">In-Progress</option>
</select></label><br/><label>
<b>Assessment scores :</b> <input type="text" name="assessment_scores" class="round1"
size="30" required>
</label><br />
<label>

```



```

<b>    Evaluation comments :</b> <input type="text" name="Evaluation_comments "
class="round1" size="30" required>
</label><br /><br/>
<input type="submit" class="round" value="submit" />
</form></div><center></center></div>
<div class="progress">
<form action="TrainerSearchTrainee" method="post">
<center><div class="trans1">
<center><h1>Progress Details</h1><br/>
<label><b><h5>    Trainee Id: </h5></b><input type="text" size="30" class="round1 "
name="traineeId" required></label><br/>
<label><b><h5>    Skill Type:</h5> </b><select name="skilltype" class="round1" required>
<option>Choose</option>
<option value="Personality_development">Personality_development</option>
<option value="BasicIt">BasicIt</option>
</select></label></center><br/>
<input type="submit" value="submit" class="round">
</div></form></center>
</div>
</body>
</html>

```

Trainer Controller

```

package org.tms;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.tms.dao.TrainerSearchTraineeDao;

import org.tms.dao.TrainerSearchTraineeDaoImpl;

@WebServlet("/TrainerSearchTrainee")

public class TrainerSearchTraineeController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    TrainerSearchTraineeDao st=new TrainerSearchTraineeDaoImpl();

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException

    response.setContentType("text/html");

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

    String skill=request.getParameter("skilltype");

    String res = st.TrainerSearchTrainee(traineeId,skill);

    request.setAttribute("TraineeId",traineeId);

    request.setAttribute("skillType", skill);

    request.setAttribute("report", res);

    RequestDispatcher rd=request.getRequestDispatcher("DashTrainer.jsp");

    rd.forward(request, response);

}}

```

Trainer Dao Implementation

```

package org.tms.dao;

import java.sql.Connection;

import java.sql.ResultSet;

import org.tms.util.DBConstants;

import org.tms.util.DBUtil;

import com.mysql.jdbc.PreparedStatement;

public class TrainerSearchTraineeDaoImpl implements TrainerSearchTraineeDao {

    public String TrainerSearchTrainee(String traineeId,String skillType) {

        Connection con=null;

        PreparedStatement pst=null;

        String res="";

        try{con=DBUtil.getConnection(DBConstants.DRIVER,DBConstants.URL,DBConstants.UNAM
E,DBConstants.PWD);

        pst=(PreparedStatement) con.prepareStatement("select case when count(Hands_on_status)>=4
then 'completed' when count(Hands_on_status)>0 then 'in progress' else 'not completed' end as
Report from courses where p_uid=? and skill_type=? and Hands_on_status='completed'");

        pst.setString(1,traineeId);

        pst.setString(2, skillType);

        ResultSet rs= pst.executeQuery();

        System.out.println(res);

```

```

if(rs.next())

{res=rs.getString(1);}

} catch (Exception e) {e.printStackTrace();}

return res;}}

```

Trainer.java

```

package org.tms.beans;

public class DashTrainee {

private String p_uid;

private String name;

private String trainer_name;

private String skill_type;

private String course_name;

private String Hands_on_status;

private String assessment_scores;

private String Evaluation_comments;

public DashTrainee(String p_uid, String name, String trainer_name,

String skill_type, String course_name, String hands_on_status,

String assessment_scores, String evaluation_comments) {

super();

this.p_uid = p_uid;

this.name = name;

this.trainer_name = trainer_name;

```

```

this.skill_type = skill_type;

this.course_name = course_name;

Hands_on_status = hands_on_status;

this.assessment_scores = assessment_scores;

Evaluation_comments = evaluation_comments;}

public String getP_uid() {return p_uid;}

public void setP_uid(String p_uid) {this.p_uid = p_uid;}

public String getName() {return name;}

public void setName(String name) {this.name = name;}

public String getTrainer_name() {return trainer_name;}

public void setTrainer_name(String trainer_name) {this.trainer_name = trainer_name;}

public String getSkill_type() {return skill_type;}

public void setSkill_type(String skill_type) {this.skill_type = skill_type;}

public String getCourse_name() {return course_name;}

public void setCourse_name(String course_name) {this.course_name = course_name;}public
String getHands_on_status() {return Hands_on_status;}

public void setHands_on_status(String hands_on_status) {Hands_on_status = hands_on_status;}

public String getAssessment_scores() {return assessment_scores;}

public void setAssessment_scores(String assessment_scores) {this.assessment_scores =
    assessment_scores;}

public String getEvaluation_comments() {return Evaluation_comments;}

public void setEvaluation_comments (String evaluation_comments) {

    Evaluation_comments =evaluation_comments;}}

```

Course_Entry Service.java

```
package org.tms;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.tms.beans.course;

import org.tms.services.courseservice;

import org.tms.services.courseserviceimpl;

@WebServlet("/course")

public class courseserv extends HttpServlet {

    private static final long serialVersionUID = 1L;

    courseservice service=new courseserviceimpl();

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out=response.getWriter();

        String p_uid=request.getParameter("p_uid");
```

```

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

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

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

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

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

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

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

Booleanres=service.course(new
course(p_uid,name,trainer_name,skill_type,course_name,Hands_on_status,assessment_scores,E
valuation_comments));

if(res==true)

{
    out.println("<script type=\"text/javascript\">");

    out.println("alert('success');");

    out.println("location='Trainer.jsp';");

    out.println("</script>");
}

else{

    out.println("<script type=\"text/javascript\">");

    out.println("alert('failure');");

    out.println("location='Course_Entry.jsp';");

    out.println("</script>");
}

out.close();}}

```

Course Dao Implementation

```

package org.tms.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import org.tms.beans.course;

import org.tms.util.DBConstants;

import org.tms.util.DBUtil;

public class CourseDaoimpl implements CourseDao {

    public boolean insert(course r) {

        Connection con=null;

        PreparedStatement pst=null;

        boolean b=false;

        try {

            con=DBUtil.getConnection(DBConstants.DRIVER, DBConstants.URL, DBConstants.UNAME,
            DBConstants.PWD);

            pst=con.prepareStatement("insert                                     into
            courses(p_uid,name,trainer_name,skill_type,course_name,Hands_on_status,assessment_scores,E
            valuation_comments) values(?,?,?,?,?,?,?,?)");

            pst.setString(1, r.getP_uid());

            pst.setString(2, r.getName());

            pst.setString(3, r.getTrainer_name());

            pst.setString(4, r.getSkill_type());

```



```

pst.setString(5, r.getCourse_name());

                pst.setString(6, r.getHands_on_status());

pst.setString(7, r.getAssessment_scores());

pst.setString(8, r.getEvaluation_comments());

int s=pst.executeUpdate();

if(s>0)b=true;

elseb=false;

con.close();} catch (Exception e2) {e2.printStackTrace();}

return b;}}

```

Course.java

```

package org.tms.beans;

public class course {
private String p_uid;
private String name;
private String trainer_name;
private String skill_type;
private String course_name;
private String Hands_on_status;
private String assessment_scores;
private String Evaluation_comments;
public String getP_uid() {
return p_uid;}
public void setP_uid(String p_uid) {this.p_uid = p_uid;}
public String getName() {return name;}
public void setName(String name) {this.name = name;}
public String getSkill_type() {return skill_type;}

```

```

public void setSkill_type(String skill_type) {this.skill_type = skill_type;}
public String getCourse_name() {return course_name;}
public void setCourse_name(String course_name) {
    this.course_name = course_name;    }
public String getHands_on_status() {return Hands_on_status;}
public void setHands_on_status(String hands_on_status) {
    Hands_on_status = hands_on_status;}
public String getAssessment_scores() {return assessment_scores;}
public void setAssessment_scores(String assessment_scores) {
    this.assessment_scores = assessment_scores;}
public String getEvaluation_comments() {return Evaluation_comments;}
public void setEvaluation_comments(String evaluation_comments) {
    Evaluation_comments = evaluation_comments;}
public String getTrainer_name() {return trainer_name;}
public void setTrainer_name(String trainer_name) {
    this.trainer_name = trainer_name;}}

```