

Project Report  
On

# STUDENT RESULT MANAGEMENT SYSTEM

Submitted by

HIMANSHU  
GHANSHYAM KUMAR  
CHAMAN RAJ  
DHEERAJ KUMAR DAS  
ARSHDEEP KAUR

In partial fulfillment of the requirements for the Degree of  
**Bachelor of Computer Applications (BCA) in  
Department of Computational Sciences**

To



MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY  
BATHINDA - 151001

JULY 2022

**MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY  
DEPARTMENT OF COMPUTATIONAL SCIENCES  
BATHINDA-151001**

**Certificate**

Certified that this is a Bonafede record of the project work entitled  
**“STUDENT RESULT MANAGEMENT SYSTEM”**  
done by the following students

**HIMANSHU  
GHANSHYAM KUMAR  
CHAMAN RAJ  
DHEERAJ KUMAR DAS  
ARSHDEEP KAUR**

Of the VI<sup>th</sup> semester, Department of Computational Sciences, in the year 2022 in partial fulfillment of the requirements to the award of Degree of Batchelor in Computer Applications of Maharaja Ranjit Singh Punjab Technical University Bathinda.

**Dr.Munish kumar  
Project Guide**

**prof.Sanjay Bhatnagar  
Head of the Department**

**Place:Bathinda**

**Date:**

## **ACKNOWLEDGEMENT**

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heart felt thanks to our esteemed guide, Dr.Munish Kumar for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected head of the division Mr.Sanjay Bhatnagar, for allowing us to use the facilities available. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

HIMANSHU  
GHANSHYAM KUMAR  
CHAMAN RAJ  
DHEERAJ KUMAR DAS  
ARSHDEEP KAUR

# **ABSTRACT**

## **“STUDENT RESULT MANAGEMENT SYSTEM”**

The main objective of this project is to provide results to the students in a simple way. The students can get results through the college/institution website through their roll numbers. By analyzing the result status and applying the standard calculation followed by the University the result are displayed with individual scores and the equivalent percentage. The system is intended for the student. There is no need to login, student can see their result simply by entering university roll no. This can be achieved with web development technologies like HTML, CSS, JSP(JAVA), JavaScript and using the database MySQL. The faculty can view the overall performance of the students in the semester examinations subject wise. The visualization of the overall results according to the subject.

Identification of the drawbacks of the existing system leads to the designing of this modern era application that will be compatible to the existing system with the system Which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

- Less human error
- Secure
- Data redundancy can be avoided to some extent
- Data consistency
- Easy to handle
- Easy data updating
- Easy data deleting
- Easy record keeping

# Table of Contents

<b>1. INTRODUCTION.....</b>	<b>7</b>
<b><u>CHAPTER 1</u> .....</b>	<b>7</b>
1.1 PURPOSE .....	8
1.2 SCOPE .....	8
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	8
1.4 OVERVIEW.....	8
<b><u>CHAPTER 2</u> .....</b>	<b>9</b>
<b>2. GENERAL DESCRIPTION.....</b>	<b>10</b>
2.1 PRODUCT PERSPECTIVE.....	10
2.2 PRODUCT FUNCTIONS .....	10
2.3 USER CHARACTERISTICS .....	10
2.4 GENERAL CONSTRAINTS .....	10
2.5 ASSUMPTIONS AND DEPENDENCIES.....	11
<b><u>CHAPTER 3</u> .....</b>	<b>11</b>
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>12</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS .....	12
3.1.1 <i>User Interfaces</i> .....	12
3.1.2 <i>Hardware Interfaces</i> .....	14
3.1.3 <i>Software Interfaces</i> .....	14
3.2 <i>Software Features</i> .....	14
3.2.1 <i>Eclipse IDE</i> .....	14
3.2.2 <i>Features of Eclipse IDE</i> .....	15
3.3 <i>Java</i> .....	16
3.3.1 <i>Why Use Java</i> .....	16
3.3.2 <i>Java Install</i> .....	16
3.3.3 <i>Java setup for window</i> .....	17
3.3.4 <i>Java Syntax</i> .....	17
3.3.5 <i>Java Data Types</i> .....	18
3.3.6 <i>Object Oriented Programming</i> .....	19
3.3.7 <i>Java Class Method</i> .....	21
3.4 <i>JSP</i> .....	21
3.4.1 <i>Advantage of JSP Over Servlet</i> .....	21
3.4.2 <i>The Life Cycle of JSP Page</i> .....	22
3.4.3 <i>Creating Simple JSP Page</i> .....	23
3.4.4 <i>How to Run Simple JSP Page</i> .....	24
3.4.5 <i>Structure of JSP To Run Simple Pages</i> .....	24
3.4.6 <i>The Directory Structure Of JSP</i> .....	24
3.4.7 <i>The JSP API</i> .....	25
3.4.8 <i>Java x. Servlet . JSP Package</i> .....	25
3.4.9 <i>The JSP Page Interface</i> .....	26
3.4.10 <i>The Http JSP Page Interface</i> .....	27
3.5 <i>HTML</i> .....	36
3.5.1 <i>Features of HTML</i> .....	37

3.5.2 <i>Description Of HTML Example</i> .....	38
3.6 <i>CSS</i> .....	38
3.6.1 <i>CSS Example with CSS Editor</i> .....	39
3.6.2 <i>Use Of CSS</i> .....	39
3.7 <i>MySQL</i> .....	40
3.7.1 <i>Why Use My Sql</i> .....	40
 <u>CHAPTER 4</u> .....	42
<b>4. ANALYSIS MODELS</b> .....	43
4.1 FLOW CHART .....	43
 <u>CHAPTER 5</u> .....	46
<b>5. SYSTEM TESTING</b> .....	43
5.1 <u>UnitTesting</u> .....	47
5.2 <u>Integration Testing</u> .....	47
5.3 <u>User Acceptance Testing</u> .....	48
5.4	
 <u>CHAPTER 6</u> .....	49
<b>6. IMPORTANT CODE</b> .....	50
6.1 <u>Admin login html code</u> .....	50
6.2 <u>Admin login action jsp Code</u> .....	50
6.3 <u>Admin Home Jsp Code</u> .....	51
6.4 <u>Add New Student Code</u> .....	55
6.5 <u>Update Jsp Code</u> .....	56
6.6 <u>Update processing Code</u> .....	57
6.7 <u>Update Result Process Code</u> .....	58
6.8 <u>Delete result Code</u> .....	58
6.9 Delete Student Code .....	58
6.10 <u>Insert New Result Code</u> .....	59
 <u>CHAPTER 7</u> .....	60
<b>7. IMPLEMENTATION</b> .....	61
7.1 <u>User Training</u> .....	61
7.2 <u>Security and Maintenance</u> .....	62
 <u>CHAPTER 8</u> .....	63
<b>8. CONCLUSION</b> .....	64
 <u>CHAPTER 9</u> .....	65
<b>9. BIBLIOGRAPHY</b> .....	66
 <u>APPENDIX</u> .....	67
1. <u>HOMESCREEN</u> .....	61
2. <u>RESULT CHECKING SCREEN</u> .....	61
3. <u>STUDENT RESULT SCREEN</u> .....	61
4. <u>RESULT PRINT SCREEN</u> .....	61
5. <u>ADMIN LOGIN SCREEN</u> .....	61
6. <u>ADD NEW STUDENT SCREEN</u> .....	61
7. <u>INSERT NEW STUDENT RESULT SCREEN</u> .....	61
8. <u>ALL REGISTERED STUDENTS SCREEN</u> .....	611
9. <u>ALL REGISTERED STUDENT SCREEN</u> .....	61
10. <u>UPDATE STUDENT DETAILS SCREEN</u> .....	61
11. <u>UPDATE STUDENT RESULT SCREEN</u> .....	61
12. <u>DELETE STUDENT DETAILS BUTTON</u> .....	61
13. <u>DELETE REGISTERED STUDENT DETAILS BUTTON</u> .....	61

# **Chapter : 1**

# 1. Introduction

This document aims at defining overall software requirement for STUDENT RESULT MANAGEMENT SYSTEM .Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing /using this product .

## 1.1 Purpose

This specification document describes the capabilities that will be provided by the software application STUDENT RESULT MANAGEMENT SYSTEM .It also states the various constraints by which the system will abide. The intended audience for this document are the development team, testing team and end users of the product.

## 1.2 Scope

The application will manage the information about various students enrolled in this course in different years, the subjects offered during different semesters of the course, the marks obtained by the various students in various subjects in different semesters.

The application will greatly simplify and speed up the result preparation and management process.

## 1.3 Definitions, Acronyms, and Abbreviations

- SRS-Software Requirement Specification
- IEEE
- DFD- Data Flow Diagram
- DB-Database
- OS- Operating System

## 1.4 Overview

Student's information is stored in mysql the authority is given to Administrator; he gives the permission to instructor to enter the department data and student information.

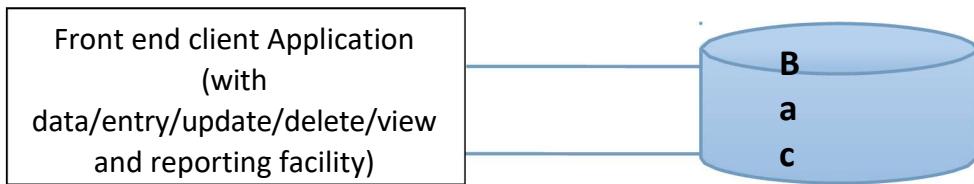
## **Chapter :2**

## 2. General Description

This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

### 2.1 Product Perspective

The application will be windows based, self-contained and independent software product.



### 2.2 Product functions

- Information about the various Students
- Information about subjects offered in semesters
- Marks obtain by Students in semester
- Generation of Reports

### 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

### 2.4 General Constraints

- Users at university will have to implement a security policy to safeguard the marks related information being modified by unauthorized users (by means of gaining access to the backend database)

## **2.5 Assumptions and Dependencies**

- 1) The number of subjects to be taken by the student in semester does not change.
- 2) The subject types do not change.
- 3) The number of semester can change by the admin.
- 4) The users have sufficient knowledge of computers.
- 5)The users know the English language, as the user interface will be provided in English.

# **Chapter:3**

# 3. Specific Requirements

This section contains the software requirements to a level of detail sufficient to enable designers to design the system and testers to test that system.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The following screens will be provided:

- 1) HOME : This will be the first screen that will be displayed.
- 2) STUDENT RESULT: It allows students to enter their Roll no and access following:
  - Student can show their result
  - Student can generate the receipt of their respective result
- 3) ADMIN LOGIN: It allows admin(teacher) to login by their User Id and Password and access following:
  - Add New Students,
  - Insert the result of the registered students,
  - Update the registered students data
  - Delete the registered students data
  - Update the registered students result
  - Delete the registered students result

#### i. Add New Student Screen:

This screen will be accessible only to the Administrator. It will allow the admin to add the new student . Admin will enter following properties of the student :

- Course
- Branch
- Roll number
- Name
- Father name
- Gender
- save

## ii. Insert New Result Screen :

This screen will be accessible only to the Administrator. It will allow the admin to enter the marks of the students with their University Roll no:

- Roll Number
- Communicative English
- Introduction to information technology
- Computer organization
- Programming in c language
- Human value
- save

## iii. Registered Students Screen:

This screen will be accessible only to the Administrator. Here Admin can see the all Registered Students Details which is following properties

- Course
- Branch
- Roll number
- Name
- Father name
- Gender
- save

## iv. All Students Results:

This screen will be accessible only to the Administrator. Here Admin can see the all Registered Students Result/Marks Details which is following properties:

- Roll Number
- Communicative English
- Introduction to information technology
- Computer organization
- Programming in c language
- Human value

## Logout :

This screen will be accessible only to the Teacher. It is a button , that is used to logout the session and redirect on the user login screen

- 4) NOTICE: When we click on it , it will redirect MRSPTU Notice page.
- 5) ABOUT US: When we click on it , it will redirect MRSPTU About us page.
- 6) CONTACT US: When we click on it , it will redirect MRSPTU Contact Us page.

### **3.1.2 Hardware Interfaces**

- 1) Intel p4 processor with minimum 2GHz speed.
- 2) RAM: Minimum 1GB
- 3) Hard Disk: min 8GB

### **3.1.3 Software Interfaces**

- 1) Eclipse IDE for Enterprise Java and Web Developers
- 2) DB Server: SQL SERVER 2008
- 3) OS: Window Vista/XP/7/8/10/11

## **3.2 Software Features:**

### **3.2.1 Eclipse IDE**

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used

an IDE to develop Scala applications and PHP eclipse is a plug-in to eclipse that provides complete development tool for PHP.

### **3.2.2 Features of Eclipse IDE**

- Almost everything is a plug-in. Different types of plug-ins like static analysis, checking of code style, checking of a version of a system are all available in Eclipse Marketplace.
- By adding plug-ins to the IDE, developers can expand the functionality of the IDE.
- Supports multiple source knowledge tools such as grading, folding and hyperlinks navigation, macro definition browser, code editing with syntax highlighting.
- Provides a visual code debugging tool.
- Has a great user interface with a drag and drop UI design feature.
- Supports project development and administered framework for different toolchains, classic make framework, and source navigation.
- By using JavaDoc, developers can automatically create documentation for classes.

IDE can facilitate the development process:

- allows setting breakpoints;
- automatically validates syntax;
- offers a robust debugger;
- provides you readymade code template;
- robust Java editor;
- supports code refactoring;
- supports syntax coloring.

### **3.3.1 Java**

Java is a popular programming language, created in 1995. It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

### **3.3.2 Why Use Java**

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

### 3.3.3 Java Install

Some PCs might have Java already installed.

To check if you have Java installed on a Windows PC, search in the start bar for Java or type the following in Command Prompt (cmd.exe):

```
C:\Users\Your Name>java -version
```

If Java is installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

### 3.3.4 Setup for Windows

To install Java on Windows:

1. Go to "System Properties" (Can be found on Control Panel > System and Security > System > Advanced System Settings)
2. Click on the "Environment variables" button under the "Advanced" tab
3. Then, select the "Path" variable in System variables and click on the "Edit" button
4. Click on the "New" button and add the path where Java is installed, followed by **\bin**. By default, Java is installed in C:\Program Files\Java\jdk-11.0.1 (If nothing else was specified when you installed it). In that case, You will have to add a new path with: **C:\Program Files\Java\jdk-11.0.1\bin**

Then, click "OK", and save the settings

5. At last, open Command Prompt (cmd.exe) and type **java -version** to see if Java is running on your machine

### 3.3.5 Java Syntax

- The main Method

The main() method is required and you will see it in every Java program:

```
public static void main(String[] args)
```

Any code inside the main() method will be executed. You don't have to understand the keywords before and after main. You will get to know them bit by bit while reading this tutorial.

For now, just remember that every Java program has a class name which must match the filename, and that every program must contain the main() method.

- System.out.println ( )

Inside the main() method, we can use the println() method to print a line of text to the screen:

```
public static void main(String[] args) {
    System.out.println("Hello World");
}
```

### 3.3.6 Java Data Types

```
int myNum = 5;           // Integer (whole number)
float myFloatNum = 5.99f; // Floating point number
char myLetter = 'D';     // Character
boolean myBool = true;   // Boolean
String myText = "Hello"; // String
```

Data types are divided into two groups:

- Primitive data types - includes byte, short, int, long, float, double, boolean and char
- Non-primitive data types - such as String, Arrays and Classes (you will learn more about these in a later chapter)

- **Primitive Data Types**

A primitive data type specifies the size and type of variable values, and it has no additional

methods.

There are eight primitive data types in Java:

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

### 3.3.6.1 Object Oriented Programming

OOP stands for Object-Oriented Programming.

Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time

Everything in Java is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

A Class is like an object constructor, or a "blueprint" for creating objects.

To create a class, use the keyword `class`:

### Main.java

Create a class named "`Main`" with a variable `x`:

```
public class Main {  
    int x = 5;  
}
```

In Java, an object is created from a class. We have already created the class named `Main`, so now we can use this to create objects.

To create an object of `Main`, specify the class name, followed by the object name, and use the keyword `new`:

Create an object called "`myObj`" and print the value of `x`:

```
public class Main {  
    int x = 5;  
  
    public static void main(String[] args) {  
        Main myObj = new Main();  
        System.out.println(myObj.x);  
    }  
}
```

### 3.3.6.2 Java Class Methods

You learned from the Java Methods chapter that methods are declared within a class, and that they are used to perform certain actions:

```
public class Main {  
    static void myMethod() {  
        System.out.println("Hello World!");  
    }  
}
```

`myMethod()` prints a text (the action), when it is **called**. To call a method, write the method's name followed by two parentheses () and a semicolon;

```
public class Main {  
    static void myMethod() {  
        System.out.println("Hello World!");  
    }  
  
    public static void main(String[] args) {  
        myMethod();  
    }  
}
```

## 3.4 JSP

**JSP** technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

### 3.4.1 Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

### **1) Extension to Servlet**

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

### **2) Easy to maintain**

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

### **3) Fast Development: No need to recompile and redeploy**

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

### **4) Less code than Servlet**

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

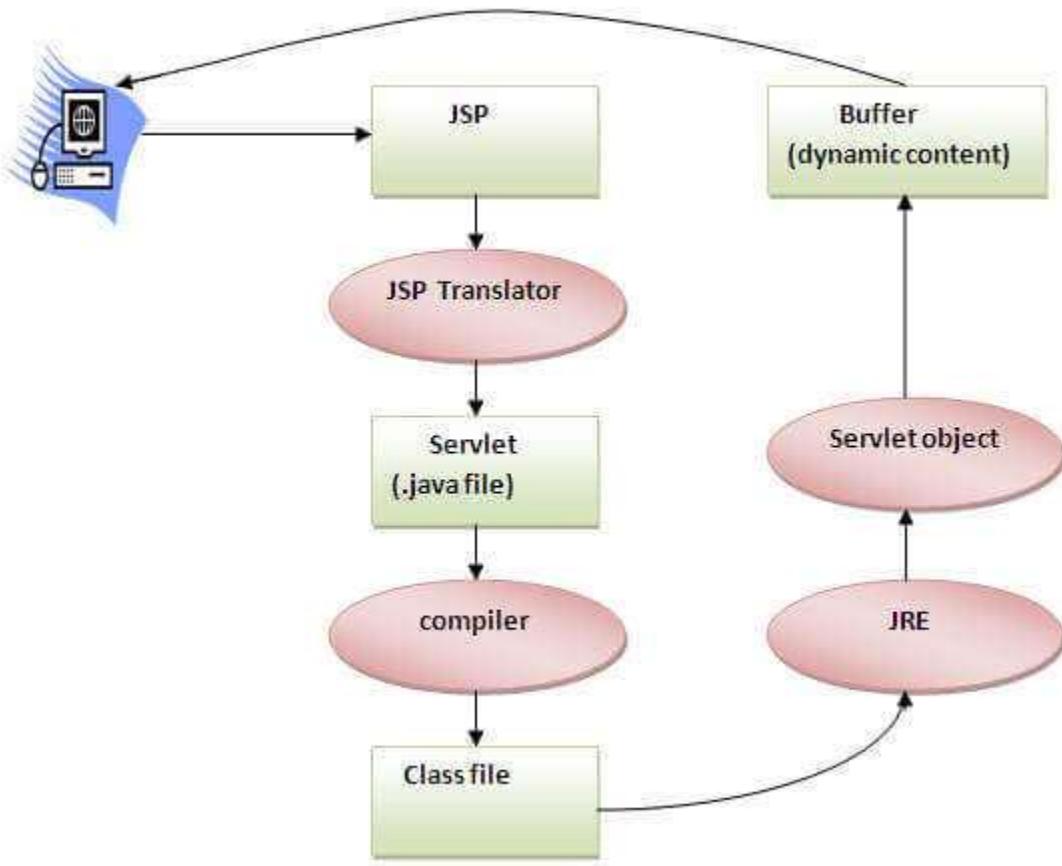
---

## **3.4.2 The Lifecycle of a JSP Page**

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization ( the container invokes jsplnIt() method).
- Request processing ( the container invokes \_jspService() method).
- Destroy ( the container invokes jspsDestroy() method).

**Note: `jslInit()`, `_jslService()` and `jslDestroy()` are the life cycle methods of JSP.**



As depicted in the above diagram, JSP page is translated into Servlet by the help of JSP translator. The JSP translator is a part of the web server which is responsible for translating the JSP page into Servlet. After that, Servlet page is compiled by the compiler and gets converted into the class file. Moreover, all the processes that happen in Servlet are performed on JSP later like initialization, committing response to the browser and destroy.

### 3.4.3 Creating a simple JSP Page

To create the first JSP page, write some HTML code as given below, and save it by .jsp extension. We have saved this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the JSP page.

#### **index.jsp**

Let's see the simple example of JSP where we are using the scriptlet tag to put Java code in the JSP page. We will learn scriptlet tag later.

1. <html>
2. <body>
3. <% out.print(2\*5); %>
4. </body>
5. </html>

It will print **10** on the browser.

#### **3.4.4 How to run a simple JSP Page?**

Follow the following steps to execute this JSP page:

- Start the server
- Put the JSP file in a folder and deploy on the server
- Visit the browser by the URL <http://localhost:portno/contextRoot/jspfile>, for example, <http://localhost:8888/myapplication/index.jsp>

---

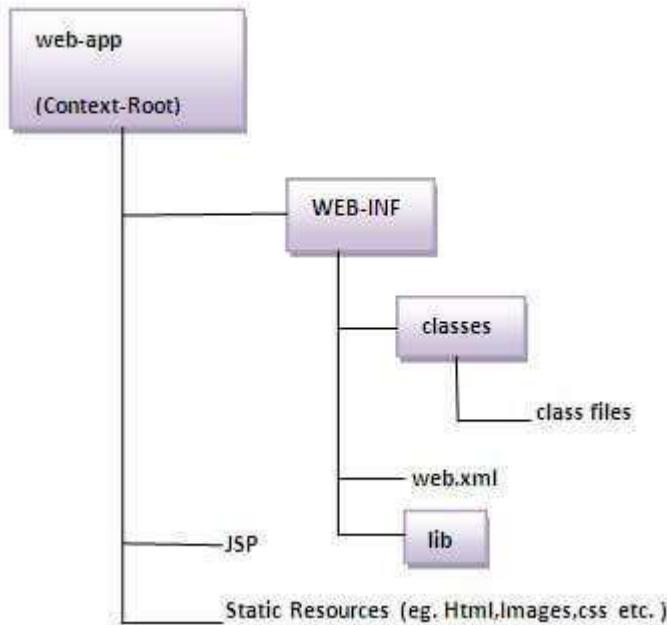
#### **3.4.5 Do I need to follow the directory structure to run a simple JSP?**

No, there is no need of directory structure if you don't have class files or TLD files. For example, put JSP files in a folder directly and deploy that folder. It will be running fine. However, if you are using Bean class, Servlet or TLD file, the directory structure is required.

---

#### **3.4.6 The Directory structure of JSP**

The directory structure of JSP page is same as Servlet. We contain the JSP page outside the WEB-INF folder or in any directory.



### 3.4.7 The JSP API

1. [The JSP API](#)
2. [javax.servlet.jsp package](#)
3. [The JspPage interface](#)
4. [The HttpJspPage interface](#)

The JSP API consists of two packages:

1. javax.servlet.jsp
2. javax.servlet.jsp.tagext

### 3.4.8 javax.servlet.jsp package

The javax.servlet.jsp package has two interfaces and classes. The two interfaces are as follows:

1. JspPage
2. HttpJspPage

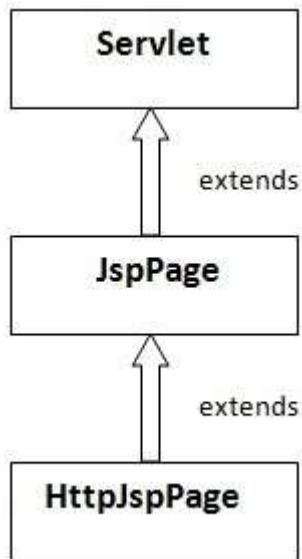
The classes are as follows:

- JspWriter
- PageContext
- JspFactory
- JspEngineInfo
- JspException
- JspError

---

### 3.4.9 The JspPage interface

According to the JSP specification, all the generated servlet classes must implement the JspPage interface. It extends the Servlet interface. It provides two life cycle methods.



#### Methods of JspPage interface

1. **public void jsplnIt():** It is invoked only once during the life cycle of the JSP when JSP page is requested firstly. It is used to perform initialization. It is same as the init() method of Servlet interface.
2. **public void jspDestroy():** It is invoked only once during the life cycle of the JSP before the JSP page is destroyed. It can be used to perform some clean up operation.

### **3.4.10 The HttpJspPage interface**

The HttpJspPage interface provides the one life cycle method of JSP. It extends the JspPage interface.

#### **Method of HttpJspPage interface:**

1. **public void \_jspService():** It is invoked each time when request for the JSP page comes to the container. It is used to process the request. The underscore \_ signifies that you cannot override this method.

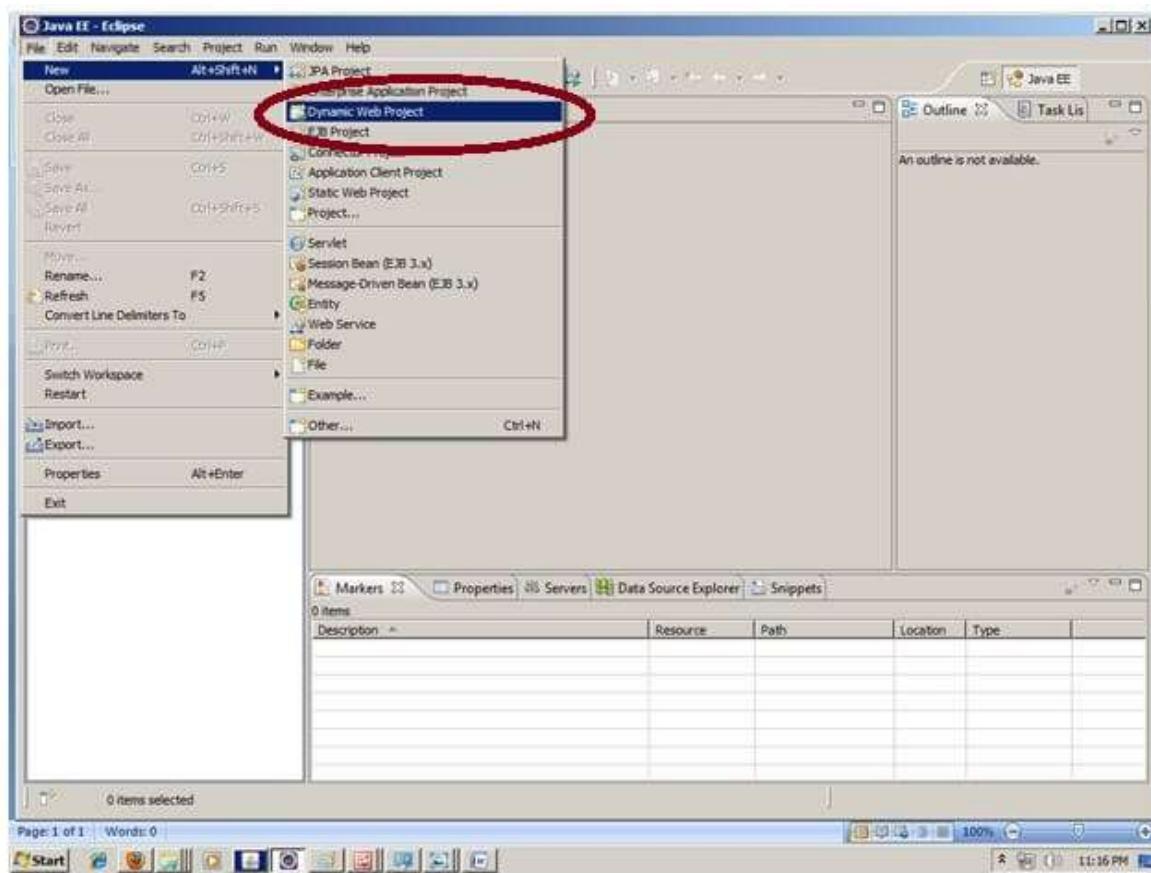
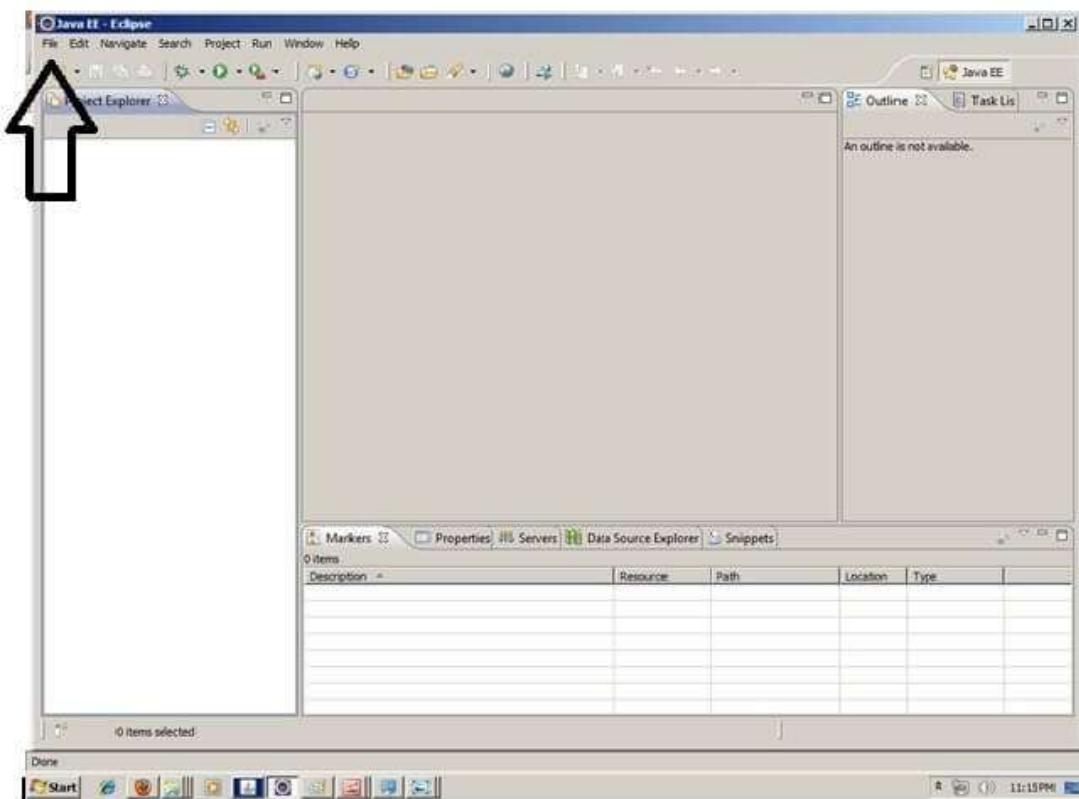
We will learn all other classes and interfaces later.

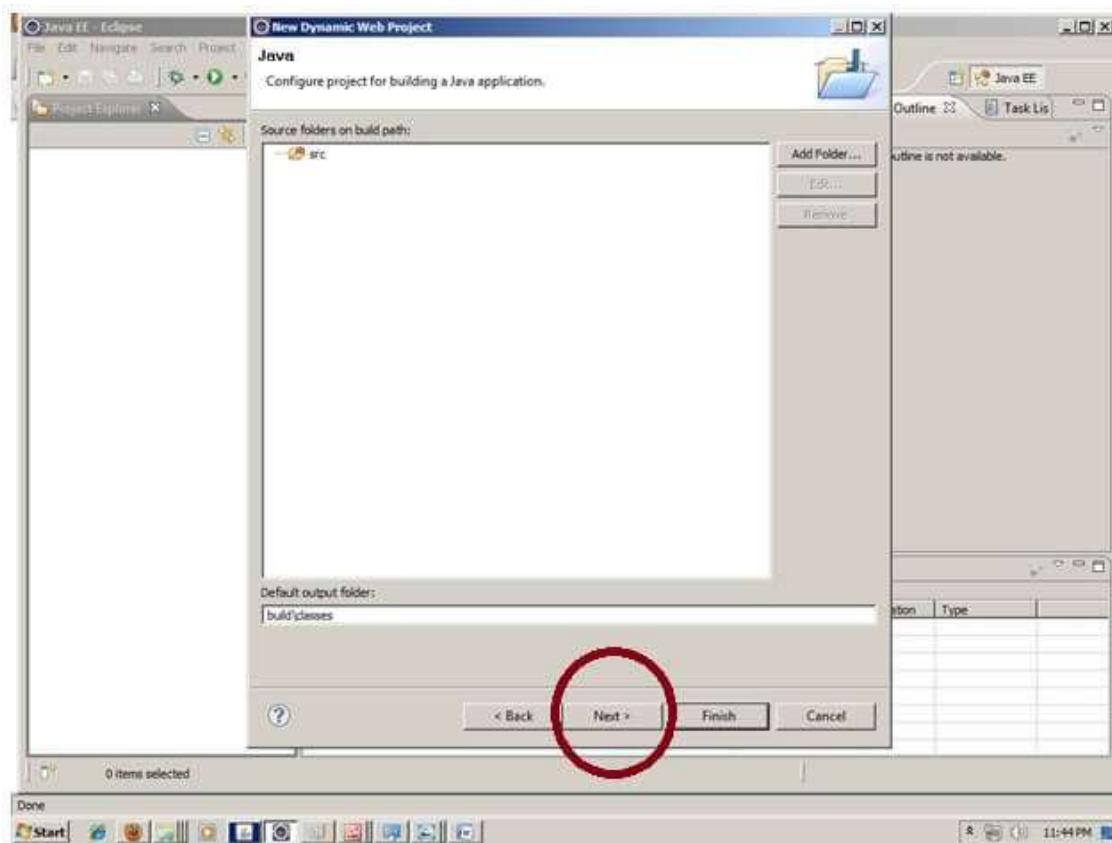
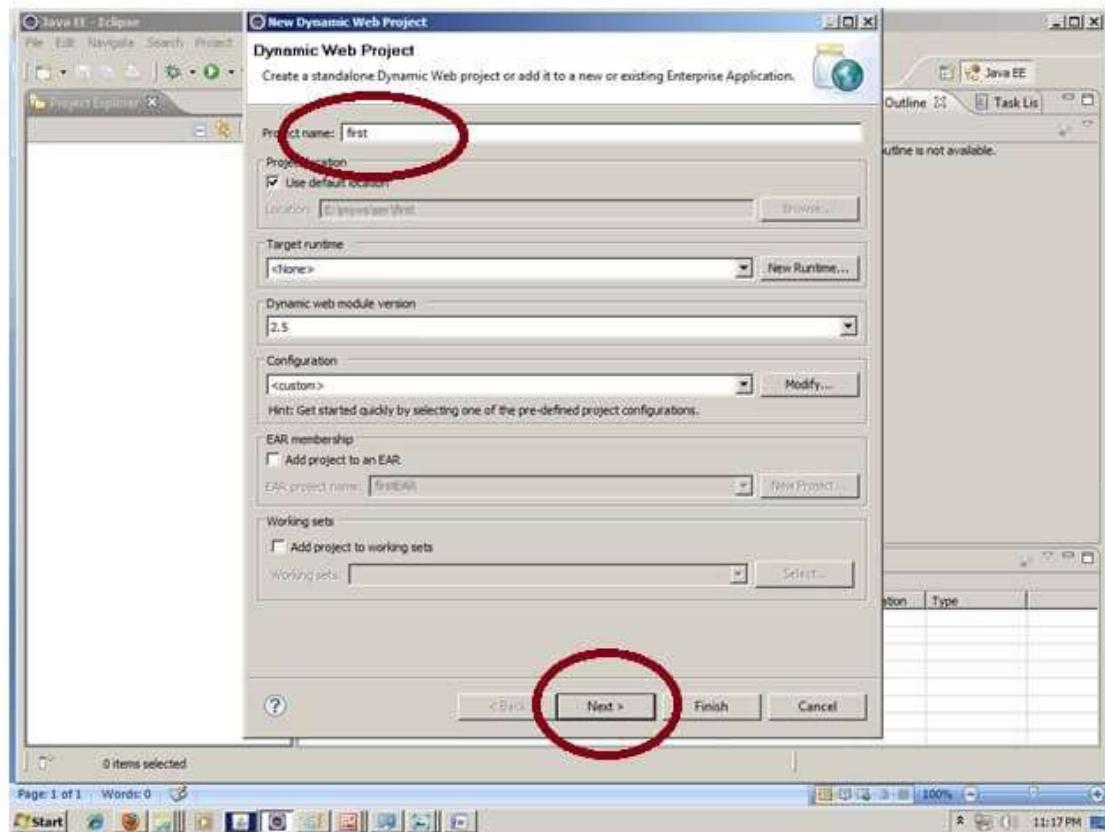
### **3.4.11 Creating JSP in Eclipse IDE with Tomcat server**

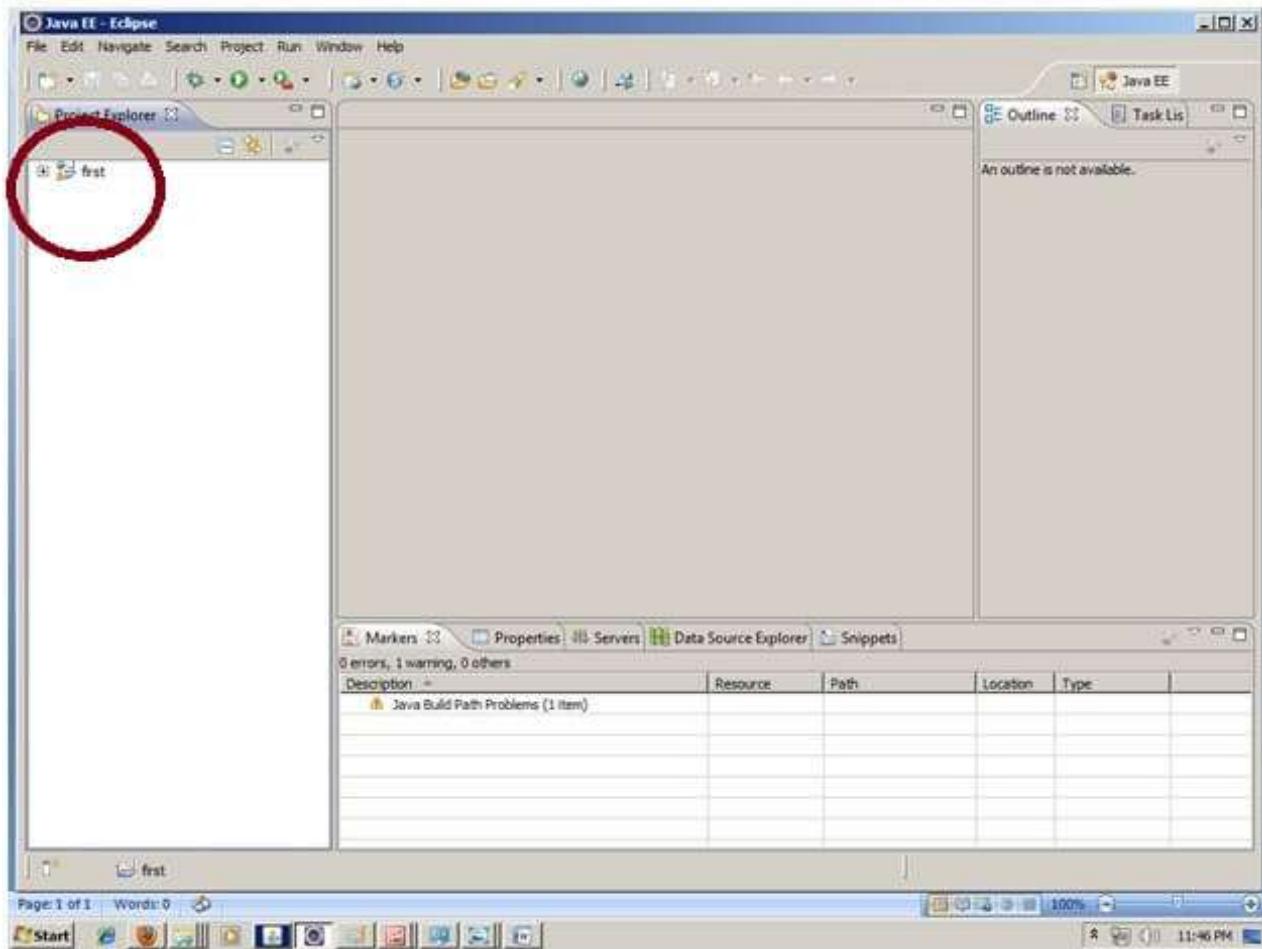
- o Create a Dynamic web project
- o create a jsp
- o start tomcat server and deploy the project

#### **Create the dynamic web project**

For creating a dynamic web project click on File Menu -> New -> dynamic web project -> write your project name e.g. first -> Finish.



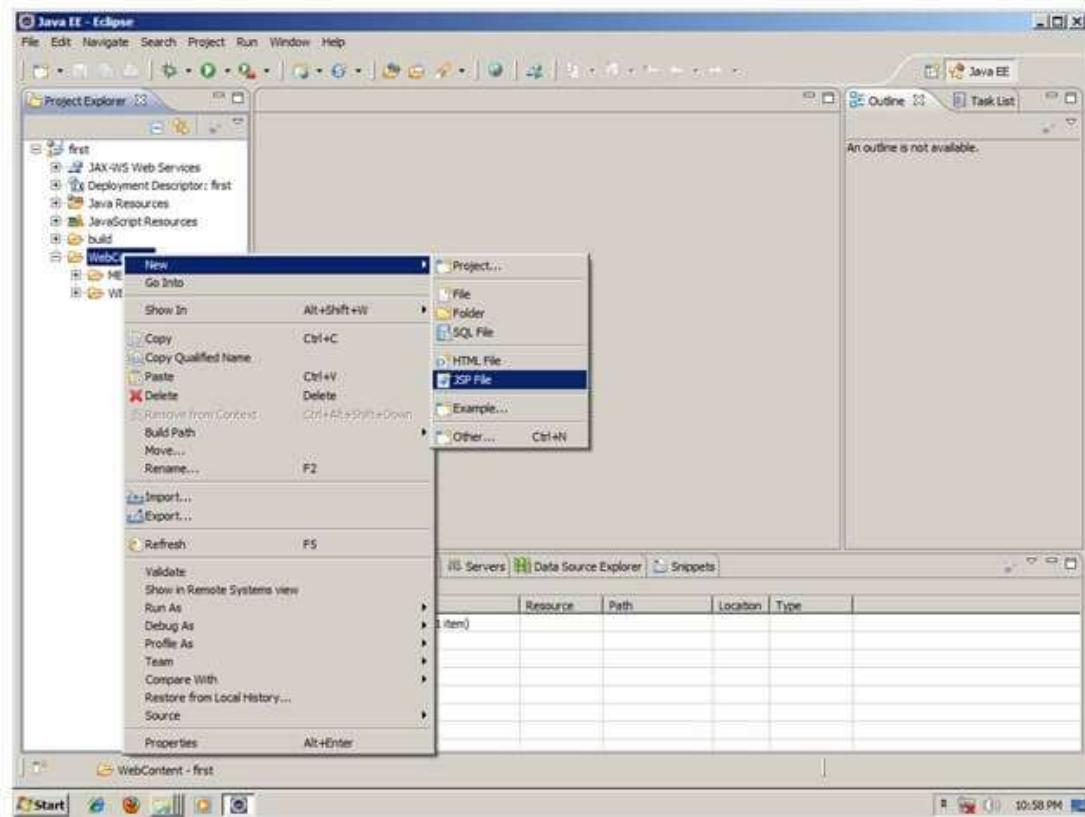
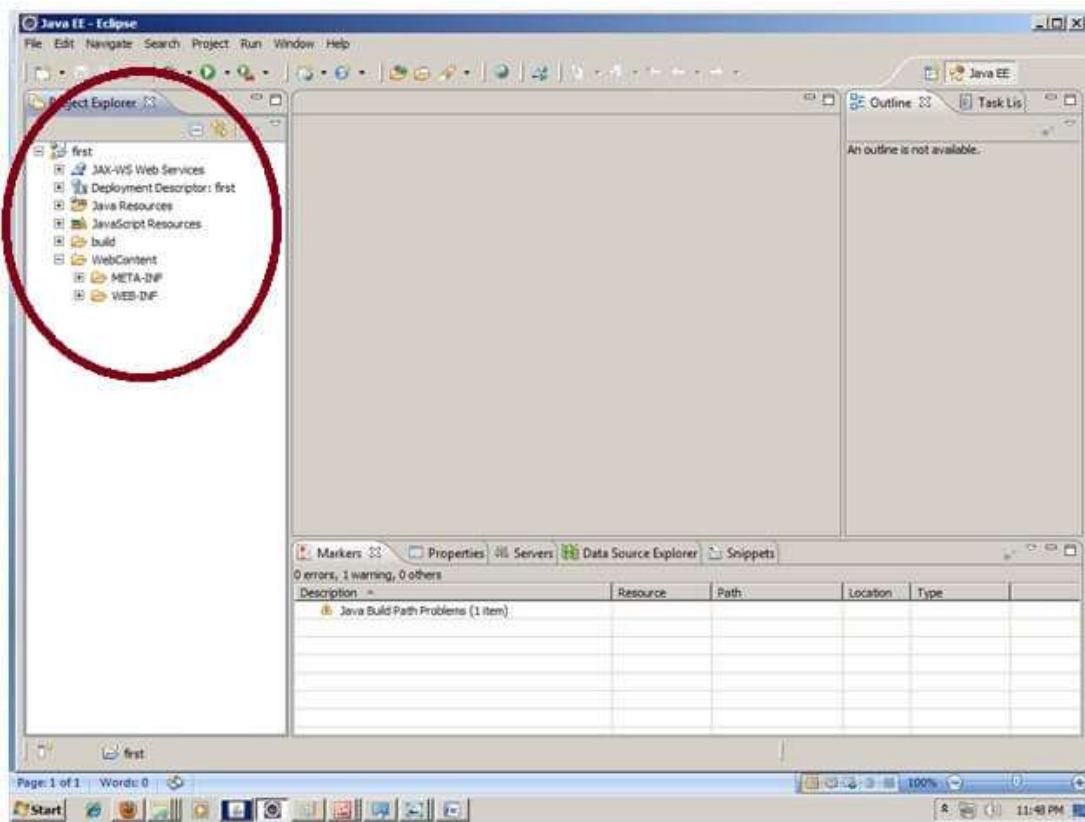


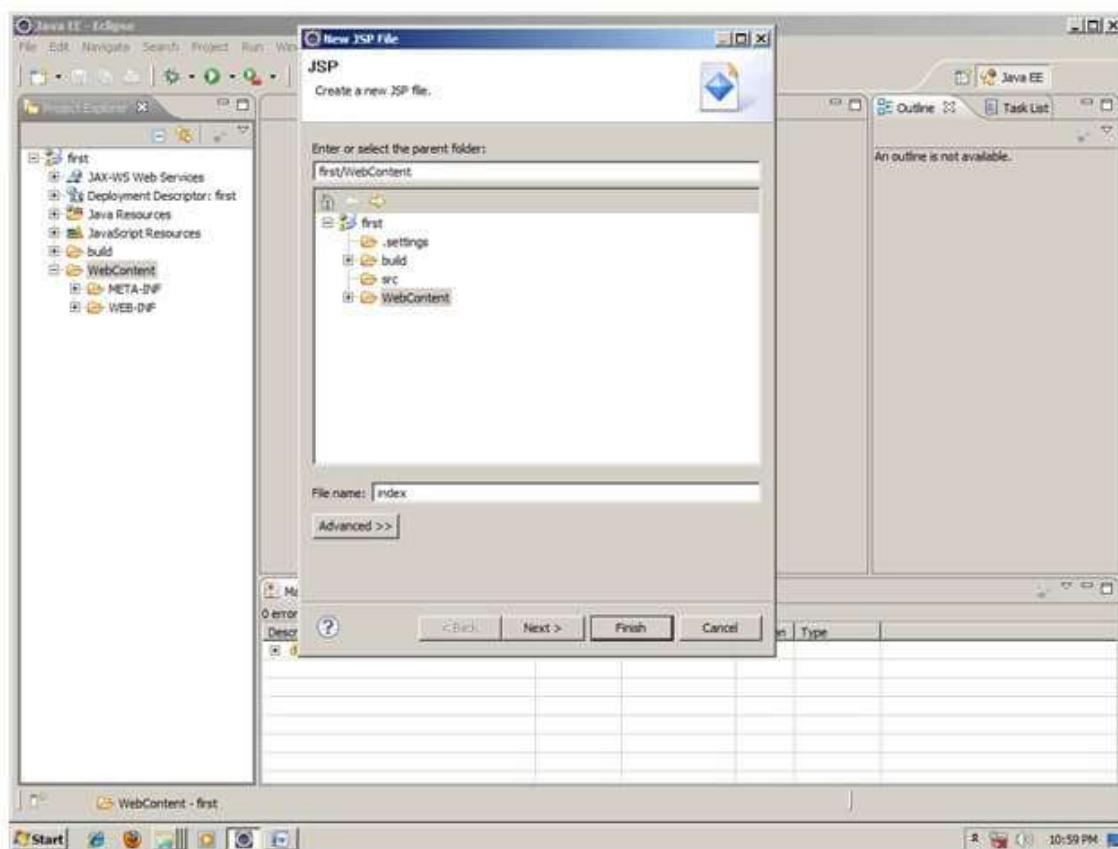
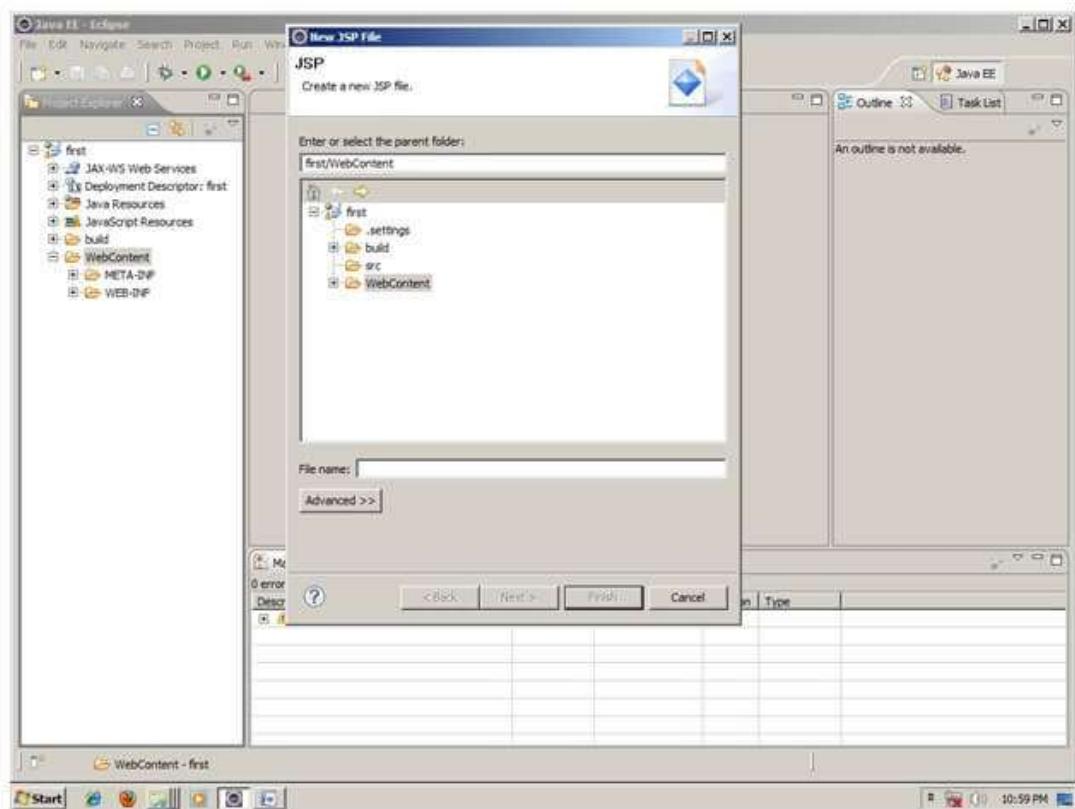


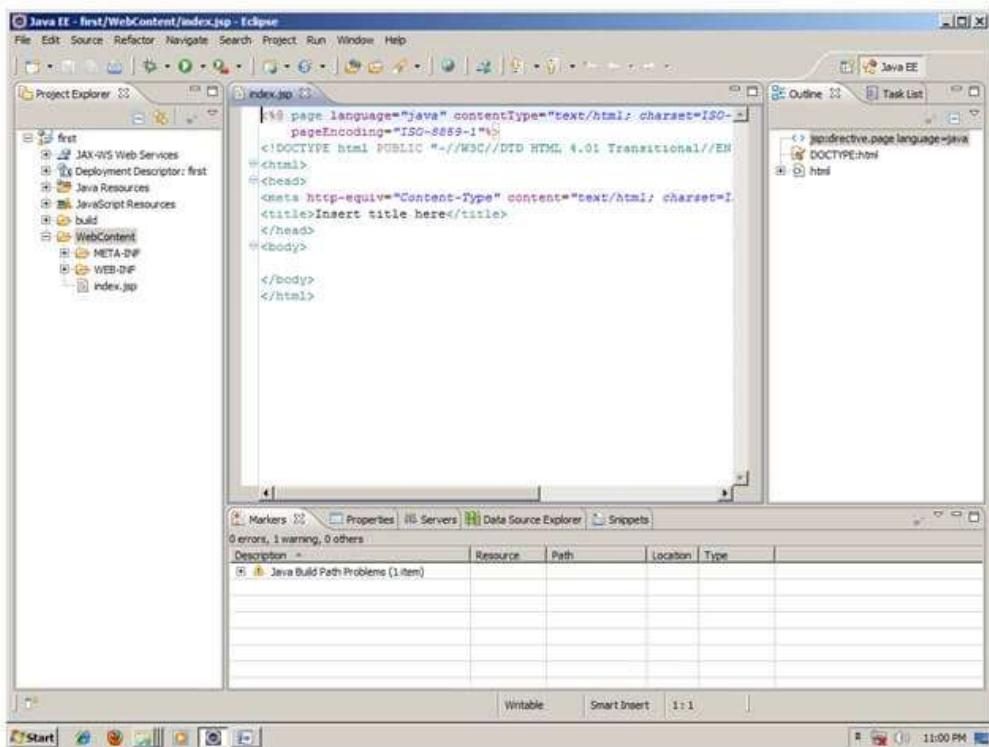
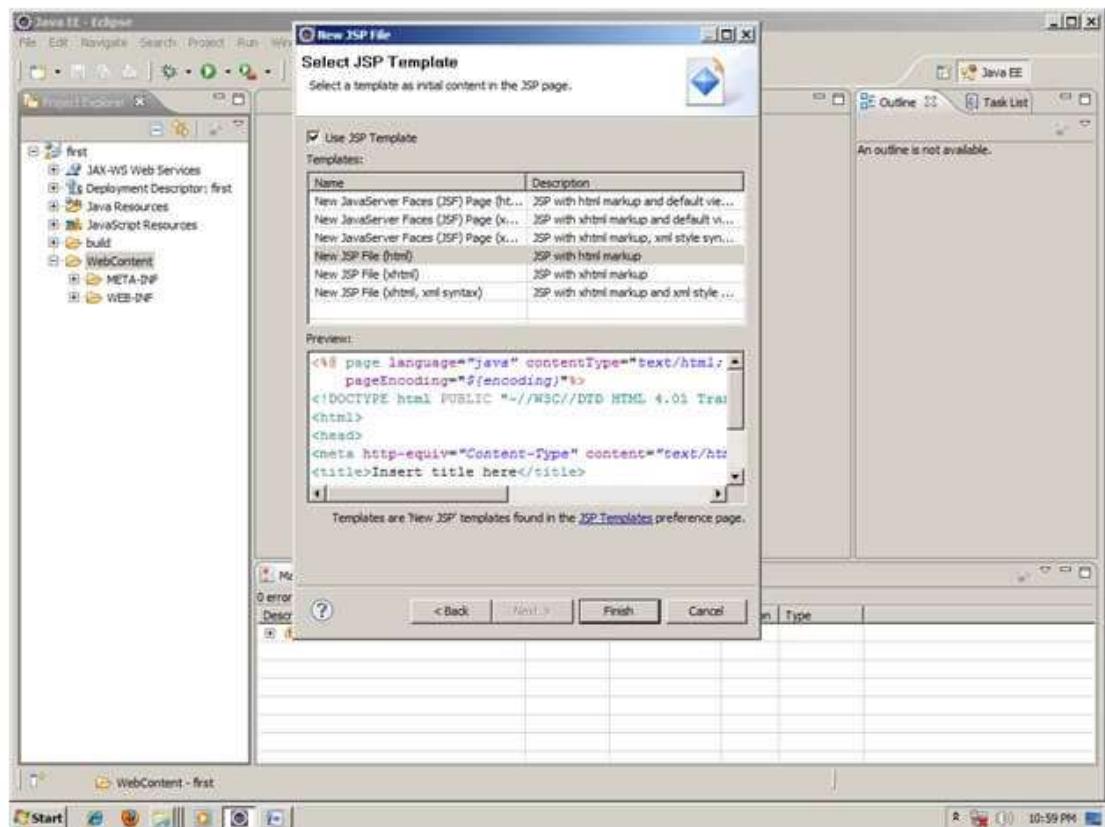
---

## 2) Create the JSP file in eclipse IDE

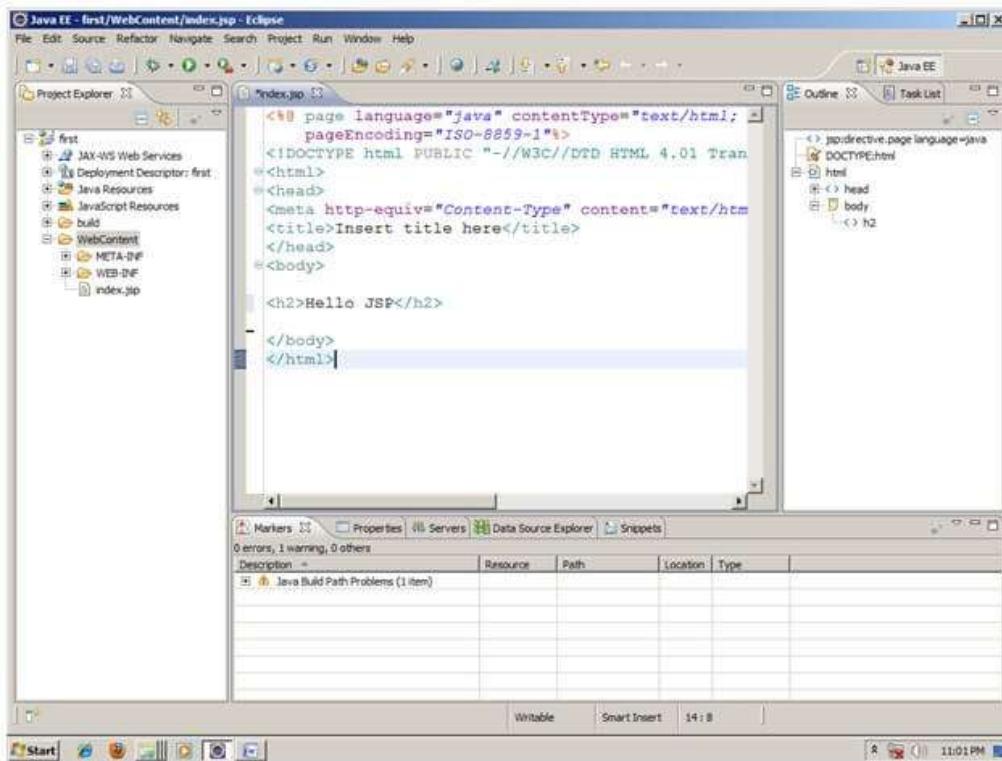
For creating a jsp file explore the project by clicking the + icon -> right click on WebContent -> New -> jsp -> write your jsp file name e.g. index -> next -> Finish.







Now JSP file is created, let's write some code.



---

### 3) Start the server and deploy the project:

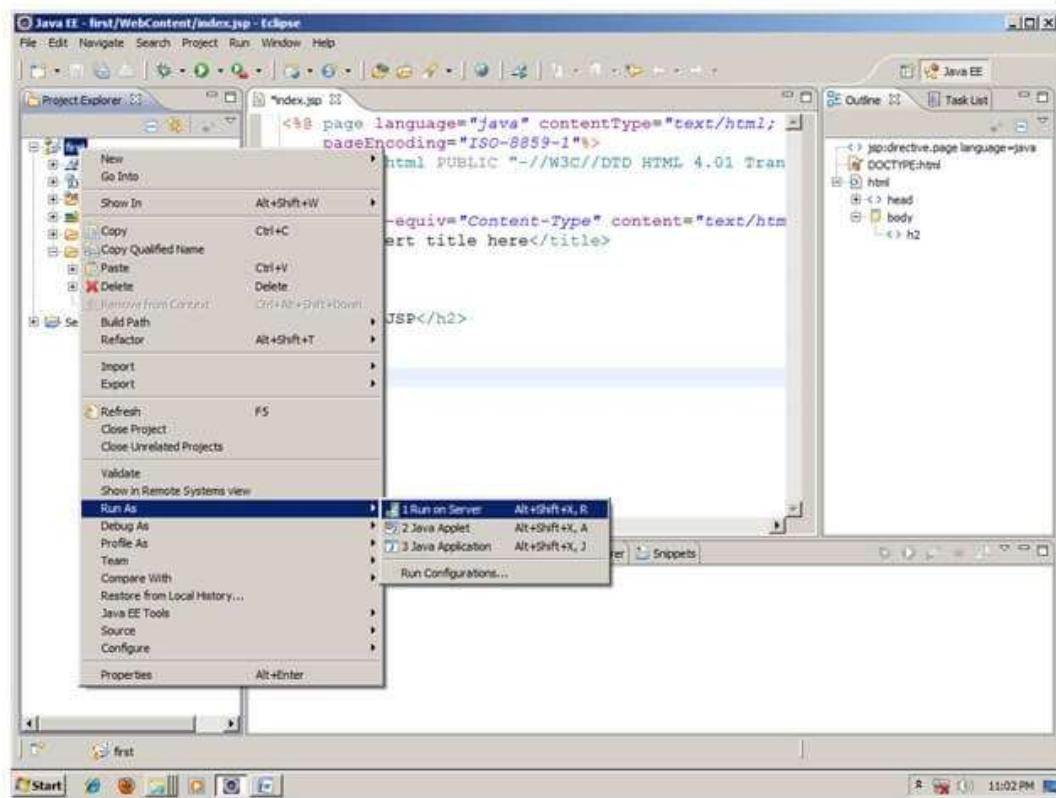
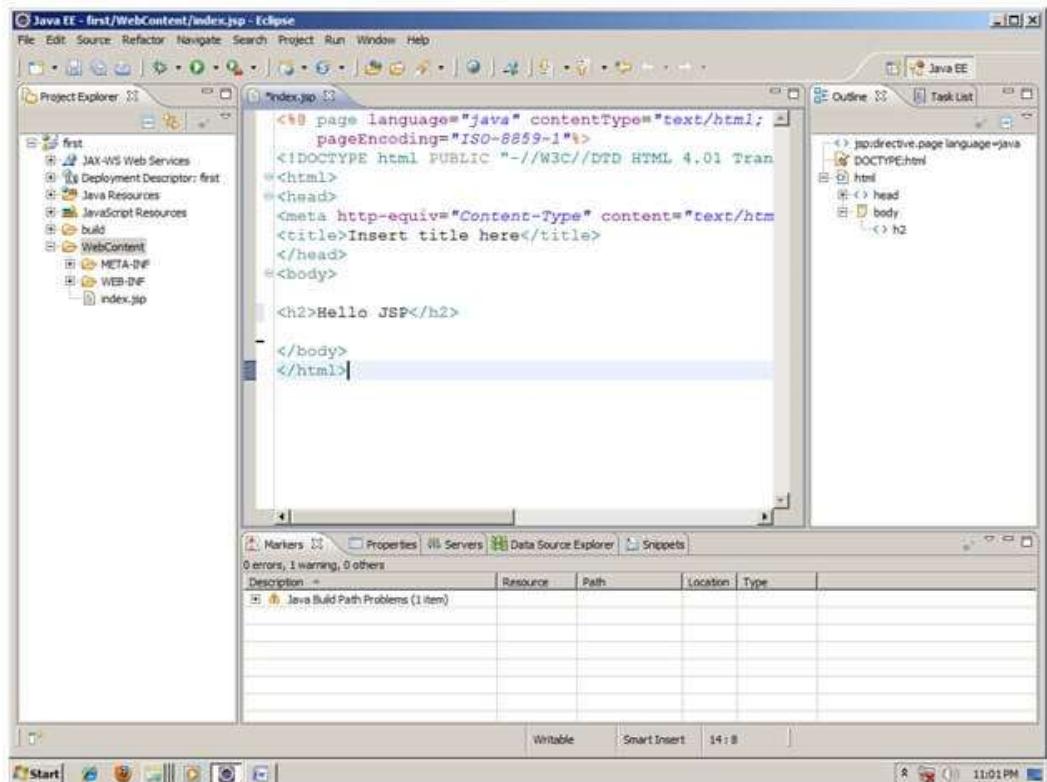
For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

---

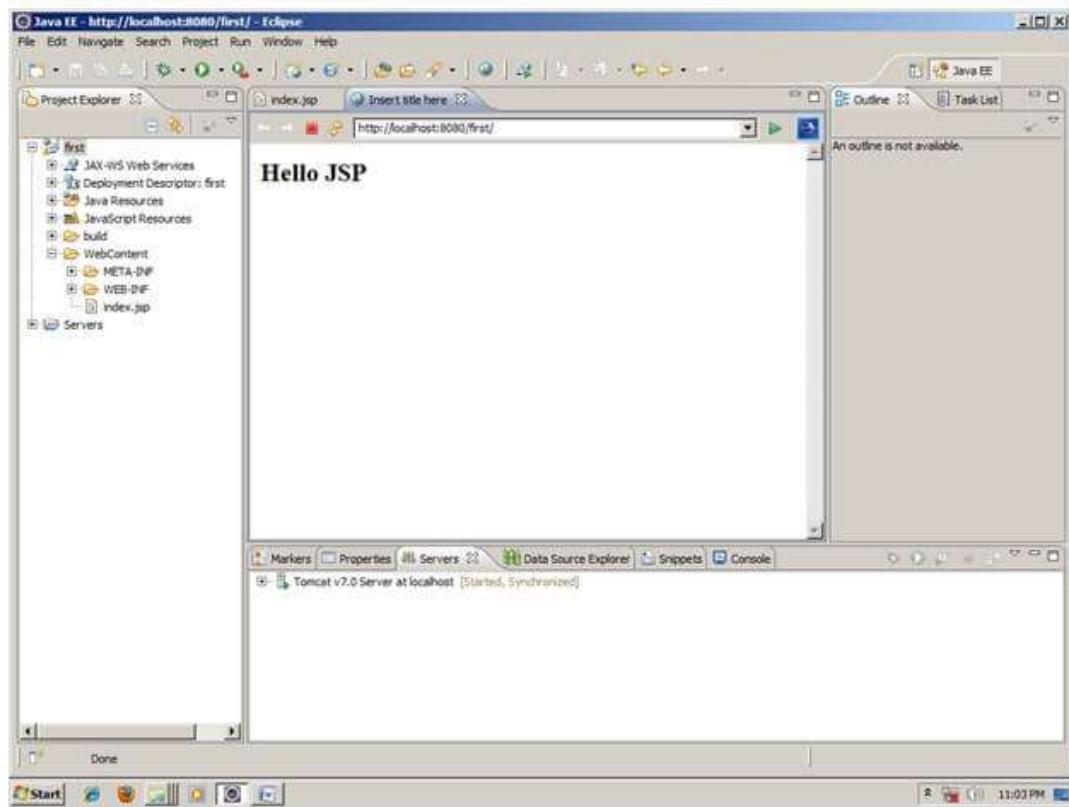
If you are using Eclipse IDE first time, you need to configure the tomcat server First. Click for [How to configure tomcat server in eclipse IDE](#)

### Now start the tomcat server and deploy project

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.



Yes, Let's see JSP is successfully running now.



## 3.5 HTML:

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:** HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

### 3.5.1 Features of HTML

- 1) It is a very **easy and simple language**. It can be easily understood and modified.
- 2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.
- 3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.
- 4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.
- 5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- 6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.
- 7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

**Let's see a simple example of HTML.**

1. <!DOCTYPE>
2. <html>
3. <head>
4. <title>Web page title</title>
5. </head>
6. <body>
7. <h1>Write Your First Heading</h1>
8. <p>Write Your First Paragraph.</p>
9. </body>
10. </html>

### **3.5.2 Description of HTML Example**

**<!DOCTYPE>**: It defines the document type or it instruct the browser about the version of HTML.

**<html>** :This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>

**<head>**: It should be the first element inside the <html> element, which contains the metadata(information about the document). It must be closed before the body tag opens.

**<title>**: As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

**<body>** : Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

**<h1>** : Text between <h1> tag describes the first level heading of the webpage.

**<p>** : Text between <p> tag describes the paragraph of the webpage.

**Below given are the Top 10 Advantages of HTML:**

- HTML is Easy to Learn and Use. ...
- HTML is Free. ...
- HTML is supported by all Browsers. ...
- HTML is the Most Friendly Search Engine. ...
- HTML is Simple to Edit. ...
- HTML can Integrate Easily with Other Languages. ...
- HTML is Lightweight. ...
- HTML is Basic of all Programming Languages.

### **3.6 CSS:**

**CSS tutorial** or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

### **3.6.1 CSS Example with CSS Editor**

In this tutorial, you will get a lot of CSS examples, you can edit and run these examples with our online CSS editor tool.

### **3.6.2 uses of CSS:**

CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.

There are several uses of CSS that are discussed as follows:

#### **Solves a big problem**

Before CSS, tags like font, color, background style, element alignments, border, and size had to be repeated on every web page. This was a very long process.

#### **Saves a lot of time**

CSS style definitions are saved in external CSS files, so it is possible to change the entire website by changing just one file.

#### **Provide more attributes**

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

#### **Pages load faster**

CSS does not require the writing of HTML tag attributes every time. There is the writing of rule just once for a tag, which can be applied to all the occurrences of the corresponding tag. So using CSS, there is less code, which means faster downloading.

## Easier Website maintenance

CSS makes the maintenance of the website easier. It plays an essential role in website maintenance. If we require a global change in the file, it can be simply done by changing the style by which all the elements on the web page will update automatically. The CSS file provides a flexible look to the website, which can be altered in a convenient way.

## Multiple device compatibility

CSS is compatible with the older language versions so that we can use CSS with the earlier language versions. Because of this, if the CSS application is developed with the older programming language versions and if the developer combines the same with new improvements, then CSS can be easily implemented with the corresponding changes so the developer can update the existing code successfully.

## 3.7 MYSQL:

**MySQL**("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The

SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety

of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation .MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL

databases

include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other

software. MySQL is also used in many high-profile, large-scale websites, including

Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and

YouTube

## 1. Functional Requirements

Depending upon the Admin role he/she will be able to access only the specific modules of the system.

- Login facility for enabling only authorized access to the system
- Admin (with role Data Entry operator) will be able to modify /add/delete information about different students that are enrolled for the course .

### DB Requirement:

The following information will be placed in DB:

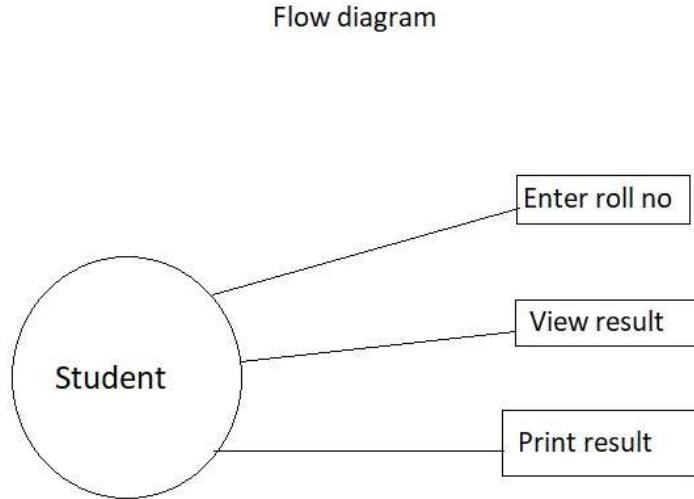
- 1) Add New Student info: Course Name, Branch Name, Roll Number, father name, Gender
- 2) Insert New Result: Roll Number, Communicative English, Introduction to information technology, Computer organization, Programming in c language, Human value
- 3) Update Students Data: Course Name, Branch Name, Roll Number, father name, Gender
- 4) Update Student Result: Roll Number, Communicative English, Introduction to information technology, Computer organization, Programming in c language, Human value
- 5) Delete Students Data: Course Name, Branch Name, Roll Number, father name, Gender
- 6) Delete Student Result: Roll Number, Communicative English, Introduction to information technology, Computer organization, Programming in c language, Human value

## **Chapter: 4.**

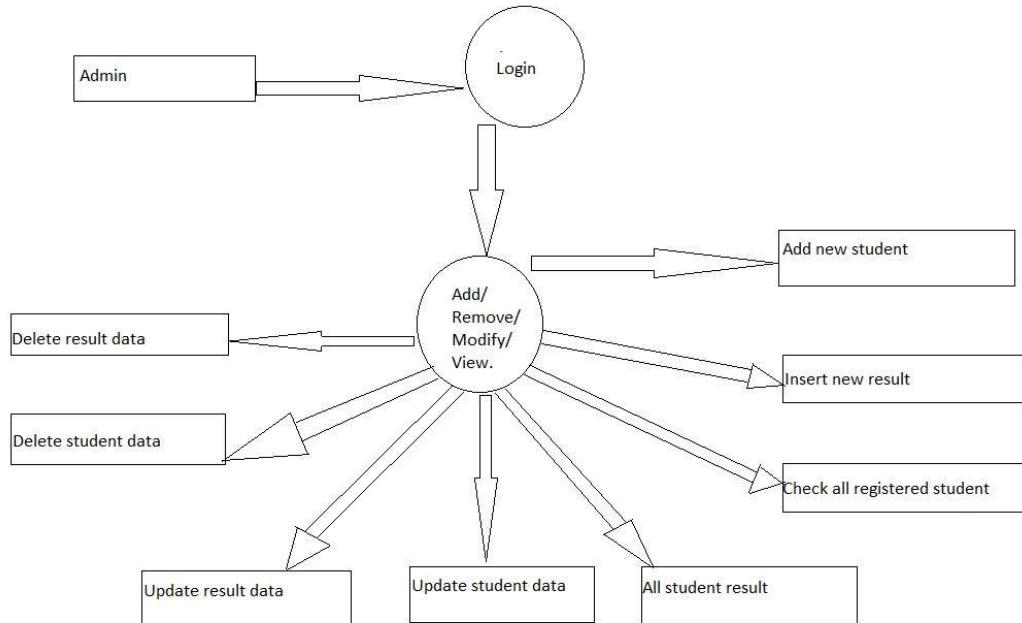
# ANALYSIS MODELS:

## 4.1 DATA FLOW DIAGRAMS

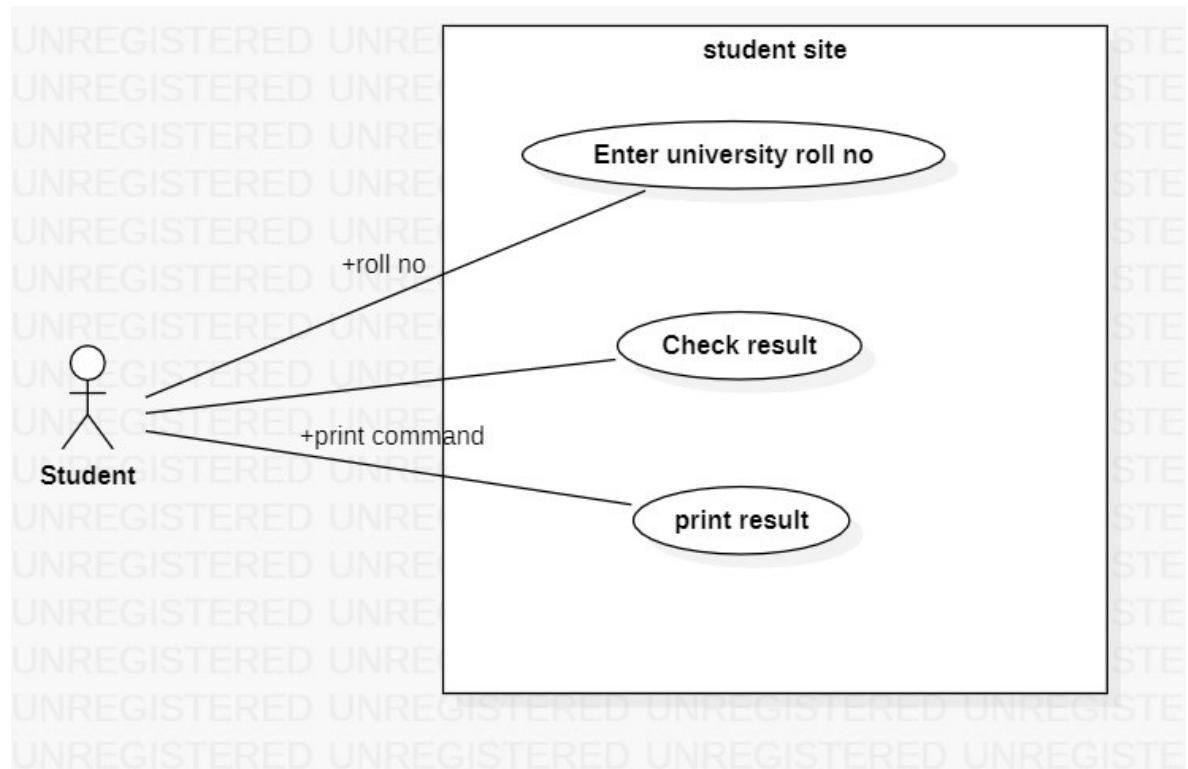
**Flow Diagram of Student:**



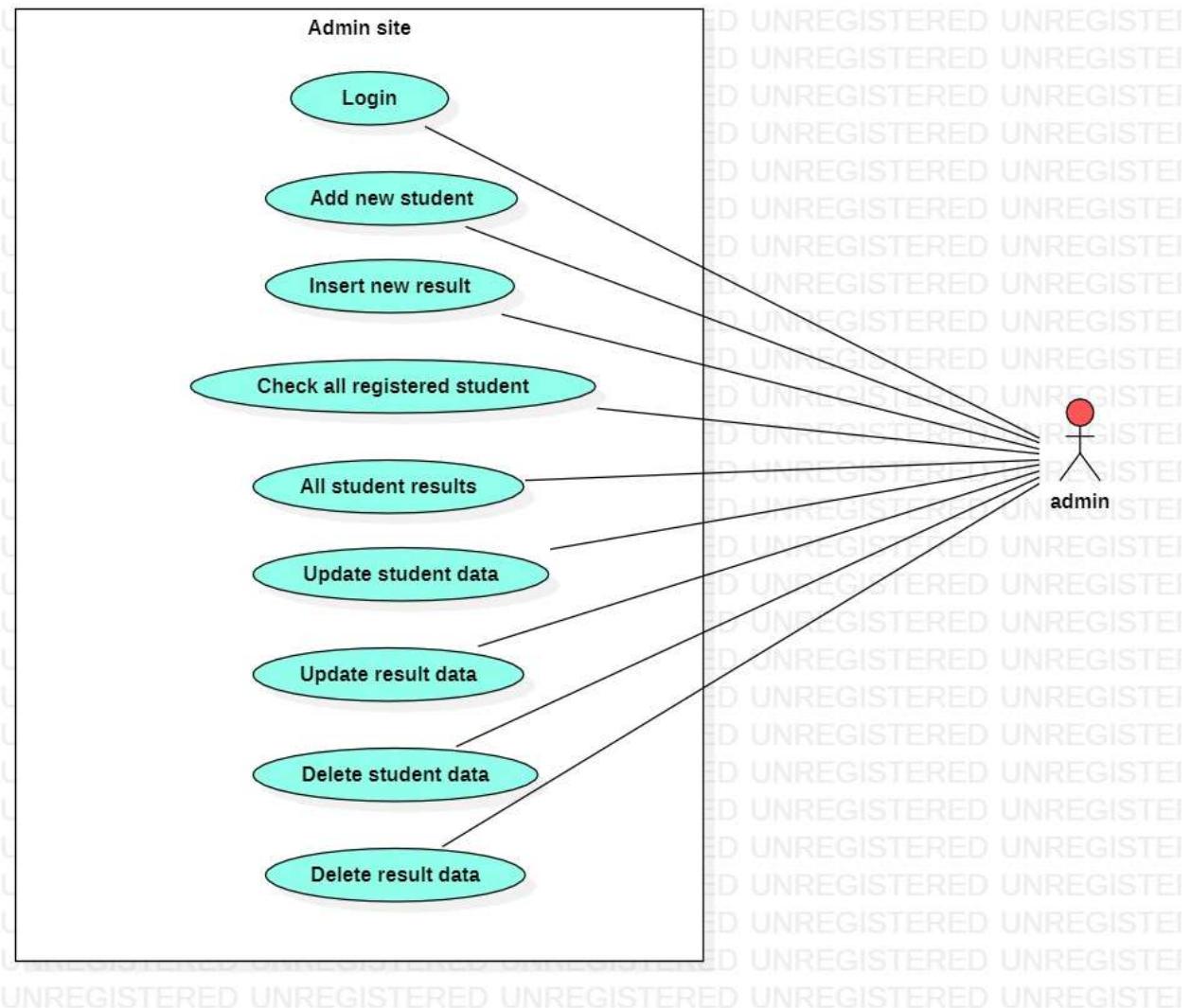
**Flow Diagram of the Admin:**



**Student Use Case Diagram:**



## Admin Use Case Diagram:



# **Chapter: 5**

## SYSTEM TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies a test plan is carried out on each module. The various tests performed in “**Network Backup System**” are unit testing, integration testing and user acceptance testing.

### 5.1 Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

### 5.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take

unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

### **5.3 User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

# **Chapter:6**

# IMPORTANT CODE

## 6.1 Adminlogin.html code

```
1<html>
2<head>
3 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
4   <link rel="stylesheet" type="text/css" href="style.css">
5   <!--A media-independent link-->
6   .Login-box{
7     height: 380px;
8   }</style>
9 </head>
10<body>
11  <form method="post" action="adminLoginAction.jsp">
12    <div class="Login-box">
13      
14      <h1>Admin Login</h1>
15      <p>Username</p>
16      <input type="text" name="username" placeholder="Enter Username" required="required">
17      <p>Password</p>
18      <input type="password" name="password" placeholder="Enter Password" required="required">
19      <input type="submit" name="submit" value="Login">
20    </form>
21    <center><h1><a href="index.html">Back</a></h1></center>
22  </div>
23 </body>
24 </html>
25
```

## 6.2 Admin Login Action jsp code

```
1<%
2 String username1=request.getParameter("username");
3 String password1=request.getParameter("password");
4
5 if(username1.equalsIgnoreCase("admin")&& password1.equalsIgnoreCase("123456"))
6 {
7
8   response.sendRedirect("adminHome.jsp");
9
10 }
11 else
12   response.sendRedirect("errorAdminLogin.html");
13 %>
```

## 6.3 Admin Home JSP Code

```
1 <%@include file="header.html"%>
2 <!DOCTYPE html>
3<html>
4<head>
5 <title>Computational Sciences</title>
6<style>
7 .button-85 {
8   padding: 0.6em 2em;
9   border: none;
10  outline: none;
11  color: rgb(255, 255, 255);
12  background: #111;
13  cursor: pointer;
14  position: relative;
15  z-index: 0;
16  border-radius: 10px;
17  user-select: none;
18  -webkit-user-select: none;
19  touch-action: manipulation;
20 }
21
22 .button-85::before {
23   content: "";
24   background: linear-gradient(
25     45deg,
26     #ff0000,
27     #ff7300,
28     #ffffb0,
29     #48f7f0,
30     #00ffd5,
31     #002bbf,
32     #7a00ff,
33     #ff00c8,
34     #ff0000
35   );
36   position: absolute;
37   top: -2px;
38   left: -2px;
39   background-size: 400%;
40   z-index: -1;
41   filter: blur(5px);
42   -webkit-filter: blur(5px);
43   width: calc(100% + 4px);
44   height: calc(100% + 4px);
45   animation: glowing-button-85 20s linear infinite;
46   transition: opacity 0.3s ease-in-out;
47   border-radius: 10px;
48 }
49
50 @keyframes glowing-button-85 {
51   0% {
52     background-position: 0 0;
53   }
54   50% {
55     background-position: 400% 0;
56   }
57   100% {
58     background-position: 0 0;
```

```

62 .button-85:after {
63   z-index: -1;
64   content: "";
65   position: absolute;
66   width: 100%;
67   height: 100%;
68   background: #222;
69   left: 0;
70   top: 0;
71   border-radius: 10px;
72 }
73 </style>
74 </head>
75 <body>
76 
77 <br>
78 <center> <h1> Maharaja Ranjit Singh Punjab Technical University </h1></center>
79 <div class="w3-container">
80<div class="w3-bar w3-black">
81   <button class="w3-bar-item w3-button tablink w3-red" onclick="openCity(event,'London')">Add New Student</button>
82   <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Paris')">Insert New Result</button>
83   <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Tokyo')">Registered Students</button>
84   <button class="w3-bar-item w3-button tablink" onclick="openCity(event,'Tokyo')">All Student Result</button>
85   <a href="adminLogin.html" class="w3-bar-item w3-button tablink">Logout</a>
86 </div>
87
88<div id="London" class="w3-container w3-border city">
89 <br>
90   <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
91   <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
92   <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
93 <!------- Include the above in your HEAD tag ----->
94
95 <!doctype html>
96 <html lang="en">
97 <head>
98   <!-- Required meta tags -->
99   <meta charset="utf-8">
100  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
101
102  <!-- Fonts -->
103  <link rel="dns-prefetch" href="https://fonts.gstatic.com">
104  <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" type="text/css">
105
106
107  <link rel="icon" href="Favicon.png">
108
109  <!-- Bootstrap CSS -->
110  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
111 </head>
112 <body>
113
114 <main class="my-form">
115<div class="container">
116  <div class="row justify-content-center">
117    <div class="col-md-8">
118      <div class="card">
119        <div class="card-header">Add New Students</div>
120
121<div class="card">
122  <div class="card-header">Add New Students</div>
123  <div class="card-body">
124    <form name="my_form" onsubmit="return validform()" action="addNewStudent.jsp" method="post">
125      <div class="form-group row">
126        <label for="full_name" class="col-md-4 col-form-label text-md-right">Course Name</label>
127        <div class="col-md-6">
128          <input type="text" class="form-control" name="course" required>
129        </div>
130
131      <div class="form-group row">
132        <label for="email_address" class="col-md-4 col-form-label text-md-right">Branch Name</label>
133        <div class="col-md-6">
134          <input type="text" class="form-control" name="branch" required>
135        </div>
136
137      <div class="form-group row">
138        <label for="user_name" class="col-md-4 col-form-label text-md-right">Roll Number</label>
139        <div class="col-md-6">
140          <input type="text" class="form-control" name="rollNo" required>
141        </div>
142
143      <div class="form-group row">
144        <label for="phone_number" class="col-md-4 col-form-label text-md-right">Name</label>
145        <div class="col-md-6">
146          <input type="text" class="form-control" name="name" required>
147        </div>
148
149      <div class="form-group row">
150        <label for="present_address" class="col-md-4 col-form-label text-md-right">Father Name</label>
151        <div class="col-md-6">
152          <input type="text" class="form-control" name="fatherName" required>
153        </div>
154
155
156      <div class="form-group row">
157        <label for="permanent_address" class="col-md-4 col-form-label text-md-right">Gender</label>
158        <div class="col-md-6">
159          <input type="text" class="form-control" name="gender" required>
160        </div>
161
162
163      <div class="col-md-6 offset-md-4">
164        <button type="submit" class="button-85" role="button" class="btn btn-primary">
165          Save
166        </button>
167      </div>
168
169    </div>
170  </div>
171</div>
172</div>
173</div>
174</div>
175</div>

```

```

118<div class="card">
119  <div class="card-header">Add New Students</div>
120  <div class="card-body">
121    <form name="my_form" onsubmit="return validform()" action="addNewStudent.jsp" method="post">
122      <div class="form-group row">
123        <label for="full_name" class="col-md-4 col-form-label text-md-right">Course Name</label>
124        <div class="col-md-6">
125          <input type="text" class="form-control" name="course" required>
126        </div>
127
128      <div class="form-group row">
129        <label for="email_address" class="col-md-4 col-form-label text-md-right">Branch Name</label>
130        <div class="col-md-6">
131          <input type="text" class="form-control" name="branch" required>
132        </div>
133
134      <div class="form-group row">
135        <label for="user_name" class="col-md-4 col-form-label text-md-right">Roll Number</label>
136        <div class="col-md-6">
137          <input type="text" class="form-control" name="rollNo" required>
138        </div>
139
140      <div class="form-group row">
141        <label for="phone_number" class="col-md-4 col-form-label text-md-right">Name</label>
142        <div class="col-md-6">
143          <input type="text" class="form-control" name="name" required>
144        </div>
145
146      <div class="form-group row">
147        <label for="present_address" class="col-md-4 col-form-label text-md-right">Father Name</label>
148        <div class="col-md-6">
149          <input type="text" class="form-control" name="fatherName" required>
150        </div>
151
152
153      <div class="form-group row">
154        <label for="permanent_address" class="col-md-4 col-form-label text-md-right">Gender</label>
155        <div class="col-md-6">
156          <input type="text" class="form-control" name="gender" required>
157        </div>
158
159
160      <div class="col-md-6 offset-md-4">
161        <button type="submit" class="button-85" role="button" class="btn btn-primary">
162          Save
163        </button>
164      </div>
165    </div>
166  </div>
167</div>
168</div>
169</div>
170</div>
171</div>
172</div>
173</div>
174</div>
175</div>

```

```

178 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
179 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
180 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
181 </body>
182 </div>
183
184@ <div id="Paris" class="w3-container w3-border city" style="display:none">
185     <br>
186     <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
187     <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
188     <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
189     <!-- Include the above in your HEAD tag ----->
190
191 <!doctype html>
192 <html lang="en">
193@ <head>
194     <!-- Required meta tags -->
195     <meta charset="utf-8">
196     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
197
198     <!-- Fonts -->
199     <link rel="dns-prefetch" href="https://fonts.gstatic.com">
200     <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" type="text/css">
201
202
203
204     <link rel="icon" href="Favicon.png">
205
206     <!-- Bootstrap CSS -->
207     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
208
209 </head>
210 <body>
211
212
213
214     <main class="my-form">
215@     <div class="container">
216         <div class="row justify-content-center">
217             <div class="col-md-8">
218                 <div class="card">
219                     <div class="card-header">Insert New Result</div>
220                     <div class="card-body">
221                         <form name="my_form" onsubmit="return validform()" action="inserNewResult.jsp" method="post">
222                             <div class="form-group row">
223                                 <label for="full_name" class="col-md-4 col-form-label text-md-right">Roll Number</label>
224                                 <div class="col-md-6">
225                                     <input type="text" class="form-control" name="rollNo" required>
226                                 </div>
227                             </div>
228
229                             <div class="form-group row">
230                                 <label for="email_address" class="col-md-4 col-form-label text-md-right">Communicative English</label>
231                                 <div class="col-md-6">
232                                     <input type="text" min="0" max="100" class="form-control" name="s1" required>
233                                 </div>
234                         </div>
235

```

```

236@           <div class="form-group row">
237@             <label for="user_name" class="col-md-4 col-form-label text-md-right"> Information Technology</label>
238@             <div class="col-md-6">
239@               <input type="number" min="0" max="100" class="form-control" name="s2" required>
240@             </div>
241@           </div>
242@           <div class="form-group row">
243@             <label for="phone_number" class="col-md-4 col-form-label text-md-right"> Computer Organization</label>
244@             <div class="col-md-6">
245@               <input type="number" min="0" max="100" class="form-control" name="s3" required>
246@             </div>
247@           </div>
248@           <div class="form-group row">
249@             <label for="present_address" class="col-md-4 col-form-label text-md-right"> Programming in c Language</label>
250@             <div class="col-md-6">
251@               <input type="number" min="0" max="100" class="form-control" name="s4" required>
252@             </div>
253@           </div>
254@           <div class="form-group row">
255@             <label for="permanent_address" class="col-md-4 col-form-label text-md-right">Human Value .</label>
256@             <div class="col-md-6">
257@               <input type="number" min="0" max="100" class="form-control" name="s5" required>
258@             </div>
259@           </div>
260@           <div class="col-md-6 offset-md-4">
261@             <center><button type="submit" class="button-85" role="button" class="btn btn-primary">
262@               Save
263@             </button>
264@           </center>
265@         </div>
266@       </div>
267@     <div class="col-md-6 offset-md-4">
268@       <center><button type="button" class="button-85" role="button" class="btn btn-primary">
269@         Save
270@       </button>
271@     </center>
272@   </div>
273@   </div>
274@ </div>
275@ </div>
276@ <br>
277@ </div>
278@ </div>
279@ <br>
280@ </div>
281@ </main>
282</script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
283<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
284<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
285</body>
286</html>
287</div>
288</div>
289</div>
290<div id="Tokyo" class="w3-container w3-border_city" style="display:none">
291<section>
292  <!--for demo wrap-->
293  <div class="tbl-header">
294    <table cellpadding="0" cellspacing="0" border="0">
295      <thead>
296        <tr>
297          <th>Course Name</th>
298          <th>Branch Name</th>
299          <th>Roll Number</th>
300          <th>Name</th>
301          <th>Father Name</th>
302          <th>Gender</th>
303          <th>Update </th>
304          <th> Delete </th>
305        </tr>
306      </thead>
307    </table>
308  </div>
309  <div class="tbl-content">
310    <table cellpadding="0" cellspacing="0" border="0">
311      <tbody>
312        <%@page import="java.sql.*" %>
313        <%@page import="Project.ConnectionProvider" %>
314        <%
315        try
316        {
317            Connection con=ConnectionProvider.getCon();
318            Statement st=con.createStatement();
319            ResultSet rs=st.executeQuery("select * from student");
320            while(rs.next())
321            {
322                %>
323                <tr>
324                  <td><%=rs.getString(1)%></td>
325                  <td><%=rs.getString(2)%></td>
326                  <td><%=rs.getString(3) %></td>
327                  <td><%=rs.getString(4) %></td>
328                  <td><%=rs.getString(5) %></td>
329                  <td><%=rs.getString(6)%></td>
330                  <td><a href="update.jsp?rollNo=<%=rs.getString("rollNo")%>"><button class="button-85" role="button" > update </button></a></td>
331                  <td><a href="deleteStudent.jsp?rollNo=<%=rs.getString("rollNo")%>"> <button class="button-85" role="button"> delete</button></a></td>
332                </tr>
333            }
334        }
335        </tbody>
336      </table>
337    </div>
338  </div>
339</div>
340</div>
341</div>
342<%}
343catch(Exception e)
344{%
345</div>
346</div>
347</div>
348</div>
349<div id="Tokyo1" class="w3-container w3-border_city" style="display:none ">
350<%@exception %>
```

```

351@ <section>
352  <div class="tbl-header">
353    <table cellpadding="0" cellspacing="0" border="0" >
354      <thead>
355        <tr>
356          <th>RollNo </th>
357          <th>English Communication</th>
358          <th>Information Technology</th>
359          <th>Computer Organization</th>
360          <th>C Language</th>
361          <th>Human Value</th>
362          <th>Update </th>
363          <th> Delete </th>
364        </tr>
365      </thead>
366    </table>
370
371  <!--for demo wrap-->
372  <div class="tbl-header">
373    <table cellpadding="0" cellspacing="0" border="0" >
374      <thead>
375        <%@page import="java.sql.*" %>
376        <%@page import="Project.ConnectionProvider" %>
378        <%
379        try
380        {
381            Connection con=ConnectionProvider.getCon();
382            Statement st=con.createStatement();
383            ResultSet rs=st.executeQuery("select *from result");
384            while(rs.next())
385            {
386
387            %>
388            <tr>
389              <th> <font color="black"><%=rs.getString("rollNo")%> </font></th>
390              <th> <font color="black"><%=rs.getString("s1")%> </font></th>
391              <th> <font color="black"><%=rs.getString("s2")%> </font></th>
392              <th> <font color="black"><%=rs.getString("s3")%> </font></th>
393              <th> <font color="black"><%=rs.getString("s4")%> </font></th>
394              <th> <font color="black"><%=rs.getString("s5")%> </font></th>
395              <th> <a href="updateResult.jsp?rollNo=<%=rs.getString("rollNo")%>"> <button class="button-85" role="button" > Update</button> </a></th>
396              <th> <a href="deleteResult.jsp?rollNo=<%=rs.getString("rollNo")%>"> <button class="button-85" role="button" > Delete</button> </a></th>
397            </tr>
398        }
399        catch(Exception e)
400        {}
401        <%
402        String msg=request.getParameter("msg");
403        if("valid".equals(msg))
404        {
405
406        %>
407        <center><font color="red" size="5" ><%=msg%> </font></center>
408        <%
409        }
410        <%
411        </table>
412        </div>
413
414    </section>
415
416  </body>
417 </html>

```

## 6.4 Add new student JSP code

```

1  <%@page import="Project.ConnectionProvider" %>
2  <%@page import="java.sql.*" %>
3
4
5@ <
6  String course=request.getParameter("course");
7  String branch=request.getParameter("branch");
8  String rollNo=request.getParameter("rollNo");
9  String name=request.getParameter("name");
10 String fatherName=request.getParameter("fatherName");
11 String gender=request.getParameter("gender");
12
13 try
14 {
15     Connection con=ConnectionProvider.getCon();
16     Statement st=con.createStatement();
17
18     st.executeUpdate("insert into student values('"+course+"','"+branch+"','"+rollNo+"','"+name+"','"+fatherName+"','"+gender+"')");
19     response.sendRedirect("adminHome.jsp");
20 }
21 catch(Exception e)
22 {
23     out.println(e);
24 }
25
26 %
27

```

## 6.5 Update JSP Code

```
1 <%@page import="java.sql.DriverManager"%>
2 <%@page import="java.sql.ResultSet"%>
3 <%@page import="java.sql.Statement"%>
4 <%@include file="header.html"%>
5 <%@page import="java.sql.Connection"%>
6<%
7 String rollNo = request.getParameter("rollNo");
8 String driver = "com.mysql.jdbc.Driver";
9 String connectionUrl = "jdbc:mysql://localhost:3306/";
10 String database = "project1";
11 String userid = "root";
12 String password = "root";
13 try {
14     Class.forName(driver);
15 } catch (ClassNotFoundException e) {
16     e.printStackTrace();
17 }
18 Connection connection = null;
19 Statement statement = null;
20 ResultSet resultSet = null;
21 <%
22<%
23 try{
24     connection = DriverManager.getConnection(connectionUrl+database, userid, password);
25     statement=connection.createStatement();
26     String sql ="select * from student where rollNo="+rollNo;
27     resultSet = statement.executeQuery(sql);
28     while(resultSet.next()){
29     <%
30     <!DOCTYPE html>
31     <html>
32     <body>
33     <head>
34         <link rel="dns-prefetch" href="https://fonts.gstatic.com">
35         <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" type="text/css">
36     </head>
37     <br>
38     <link rel="icon" href="Favicon.png">
39
40     <!-- Bootstrap CSS -->
41     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
42 </head>
43 <br>
44 <br>
45 <br>
46 <br>
47 <br>
48 <br>
49 <main class="my_form">
50     <div class="container">
51         <div class="row justify-content-center">
52             <div class="col-md-8">
53                 <div class="card">
54                     <div class="card-header"> <center><b> Update Student Details </b> </center></div>
55                     <div class="card-body">
56 <form method="post" action="update-process.jsp">
57 <input type="hidden" name="rollNo" value="<%resultSet.getString("rollNo") %>">
58     <div class="form-group row">
```

```

58@   <div class="form-group row">
59@     <label for="full_name" class="col-md-4 col-form-label text-md-right">Course</label>
60@     <div class="col-md-6">
61@       <input type="text" name="course" class="form-control" value="<%>resultSet.getString("course") %>">
62@     </div>
63@   </div>
64@   <div class="form-group row">
65@     <label for="full_name" class="col-md-4 col-form-label text-md-right">Branch</label>
66@     <div class="col-md-6">
67@       <input type="text" class="form-control" name="branch" value="<%>resultSet.getString("branch") %>">
68@     </div>
69@   </div>
70@   <div class="form-group row">
71@     <label for="full_name" class="col-md-4 col-form-label text-md-right"> Name </label>
72@     <div class="col-md-6">
73@       <input type="text" name="rollNo" class="form-control" value="<%>resultSet.getString("rollNo") %>">
74@     </div>
75@   </div>
76@   <div class="form-group row">
77@     <label for="full_name" class="col-md-4 col-form-label text-md-right"> Name </label>
78@     <div class="col-md-6">
79@       <input type="text" name="name" class="form-control" value="<%>resultSet.getString("name") %>">
80@     </div>
81@   </div>
82@   <div class="form-group row">
83@     <label for="full_name" class="col-md-4 col-form-label text-md-right"> Father Name </label>
84@     <div class="col-md-6">
85@       <input type="text" name="fatherName" class="form-control" value="<%>resultSet.getString("fatherName") %>">
86@     </div>
87@   </div>
88@   <div class="form-group row">
89@     <label for="full_name" class="col-md-4 col-form-label text-md-right">Gender</label>
90@     <div class="col-md-6">
91@       <input type="text" name="gender" class="form-control" value="<%>resultSet.getString("gender") %>">
92@     <br><br>
93@   </div>
94@   </div>
95@ </div>
96@ <center>
97@   <input type="submit" value="submit">
98@ </center>
99@ </form>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </main>
106
107 <%
108 }
109 connection.close();
110 } catch (Exception e) {
111 System.out.println(e);
112 }
113 %>
114 </body>
115 </html>

```

## 6.6 update processing code

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"%
2 pageEncoding="ISO-8859-1"%>
3 <%@ page import="java.sql.*" %
4 <%@ String driverName = "com.mysql.jdbc.Driver";%
5 <%!String url = "jdbc:mysql://localhost:3306/project1";%
6 <%!String user = "root";%
7 <%!String psw = "root";%
8 <%
9 String course = request.getParameter("course");
10 String branch=request.getParameter("branch");
11 String rollNo=request.getParameter("rollNo");
12 String name=request.getParameter("name");
13
14 String fatherName=request.getParameter("fatherName");
15 String gender=request.getParameter("gender");
16 if(rollNo!= null)
17 {
18 Connection con = null;
19 PreparedStatement ps = null;
20 int personrollNo = Integer.parseInt(rollNo);
21 try
22 {
23 Class.forName(driverName);
24 con = DriverManager.getConnection(url,user,psw);
25 String sql="update student set course=?,branch=?,rollNo=?,name=?,fatherName=?,gender=? where rollNo=?";
26 ps = con.prepareStatement(sql);
27 ps.setString(1,course);
28 ps.setString(2,branch);
29 ps.setString(3, rollNo);
30 ps.setString(4, name);
31 ps.setString(5, fatherName);
32 ps.setString(6,gender);
33 int i = ps.executeUpdate();
34 if(i > 0)
35 {
36     response.sendRedirect("adminHome.jsp");
37 }
38 else
39 {
40 out.print("There is a problem in updating Record.");
41 }
42 }
43 catch(Exception e)
44 {
45     System.out.println(e);
46 }
47 }
48 %>

```

## 6.7 update result process code

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2 pageEncoding="ISO-8859-1"%>
3 <%@ page imports="java.sql.*" %>
4 <% String driverName = "com.mysql.jdbc.Driver";%>
5 <%String url = "jdbc:mysql://localhost:3306/project1";%>
6 <%String user = "root";%>
7 <%String psw = "root";%>
8<%>
9 String rollNo=request.getParameter("rollNo");
10 String s1 = request.getParameter("s1");
11 String s2=request.getParameter("s2");
12
13 String s3=request.getParameter("s3");
14
15 String s4=request.getParameter("s4");
16 String s5=request.getParameter("s5");
17 if(rollNo!= null)
18 {
19 Connection con = null;
20 PreparedStatement ps = null;
21 int personrollNo = Integer.parseInt(rollNo);
22 try
23 {
24 Class.forName(driverName);
25 con = DriverManager.getConnection(url,user,psw);
26 String sql="update result set rollNo=?, s1=?,s2=?,s3=?,s4=?,s5=? where rollNo="+rollNo;
27 ps = con.prepareStatement(sql);
28 ps.setString(1,rollNo);
29 ps.setString(2, s1);
30 ps.setString(3, s2);
31 ps.setString(4, s3);
32 ps.setString(5, s4);
33 ps.setString(6,s5);
34 int i = ps.executeUpdate();
35 if(i > 0)
36 {
37     response.sendRedirect("adminHome.jsp");
38 }
39 else
40 {
41 out.print("There is a problem in updating Record.");
42 }
43 }
44 catch(Exception e)
45 {
46     System.out.println(e);
47 }
48 }
49 %}
50 %>
```

## 6.8 Delete result code

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
2 <%@page imports="java.sql.*,java.util.*"%>
3 <%>
4 String rollNo=request.getParameter("rollNo");
5 try
6 {
7 Class.forName("com.mysql.jdbc.Driver");
8 Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/project1", "root", "root");
9 Statement st=conn.createStatement();
10 int ist.executeUpdate("delete from result WHERE rollNo="+rollNo);
11 response.sendRedirect("adminHome.jsp?msg=valid");
12 }
13 catch(Exception e)
14 {
15     response.sendRedirect("adminHome.jsp?msg=invalid");
16 }
17 %>
```

## 6.9 Delete Student

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
2 <%@page import="java.sql.* ,java.util.*"%>
3
4 String rollNo=request.getParameter("rollNo");
5 try
6 {
7 Class.forName("com.mysql.jdbc.Driver");
8 Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/project1", "root", "root");
9 Statement st=conn.createStatement();
10 int i=st.executeUpdate("DELETE FROM student WHERE rollNo="+rollNo);
11 response.sendRedirect("adminHome.jsp");
12 }
13 catch(Exception e)
14 {
15 System.out.print(e);
16 e.printStackTrace();
17 }
18 %>

```

## 6.10 Insert new result

```

1 <%@page import="Project.ConnectionProvider" %>
2 <%@page import="java.sql.*" %>
3
4
5 <%
6 String rollNo=request.getParameter("rollNo");
7 String s1=request.getParameter("s1");
8 String s2=request.getParameter("s2");
9 String s3=request.getParameter("s3");
10 String s4=request.getParameter("s4");
11 String s5=request.getParameter("s5");
12
13 try
14 {
15 Connection con=ConnectionProvider.getCon();
16 Statement st=con.createStatement();
17
18
19 st.executeUpdate("insert into result values('"+rollNo+"','"+s1+"','"+s2+"','"+s3+"','"+s4+"','"+s5+"')");
20 response.sendRedirect("adminHome.jsp");
21 }
22 catch(Exception e)
23 {
24 out.println(e);
25 }
26 %>
27

```

# **Chapter:7**

## **IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed.

Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

### **7.1 User Training**

After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. For this purpose user manuals are prepared and handled over to the user to operate the developed system. Thus the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future. In order to put new application system into use, the following activities were taken care of:

- Preparation of user and system documentation
- Conducting user training with demo and hands on

- Test run for some period to ensure smooth switching over the system

The users are trained to use the newly developed functions. User manuals describing the procedures for using the functions listed on menu are circulated to all the users. It is confirmed that the system is implemented up to users need and expectations.

## **7.2 Security and Maintenance**

Maintenance involves the software industry captive, typing up system resources .It means restoring something to its original condition. Maintenance follows conversion to the extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files.

Password protection and simple procedures to prevent the unauthorized access are provided to the users .The system allows the user to enter the system only through proper user name and password.

## **CHAPTER 8**

## **CONCLUSION**

To conclude the description about the project : The project, developed using Java and JSP is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's application requires an appropriate approach towards web development. This student result management web application is designed for people who want to manage various activities in the College or Department. For the past few years the number of educational institutions are increasing rapidly.

Thereby the number of department are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the result management applications are not usually used in this context. This particular project deals with the problems on managing a result and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of this web application that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

## **CHAPTER: 9**

## BIBLIOGRAPHY

1. [www.w3schools.com](http://www.w3schools.com)
2. <https://www.tutorialspoint.com/eclipse/index.htm/>
3. <https://www.javatpoint.com/how-to-use-eclipse-for-java>
- 4 . <https://www.javatpoint.com/jsp-tutorial>
5. <https://getstream.io/blog/use-github-eclipse-for-java/>
6. [https://www.w3schools.com/mySQL/default.asp/](https://www.w3schools.com/mySQL/default.asp)

## ***APPENDIX***

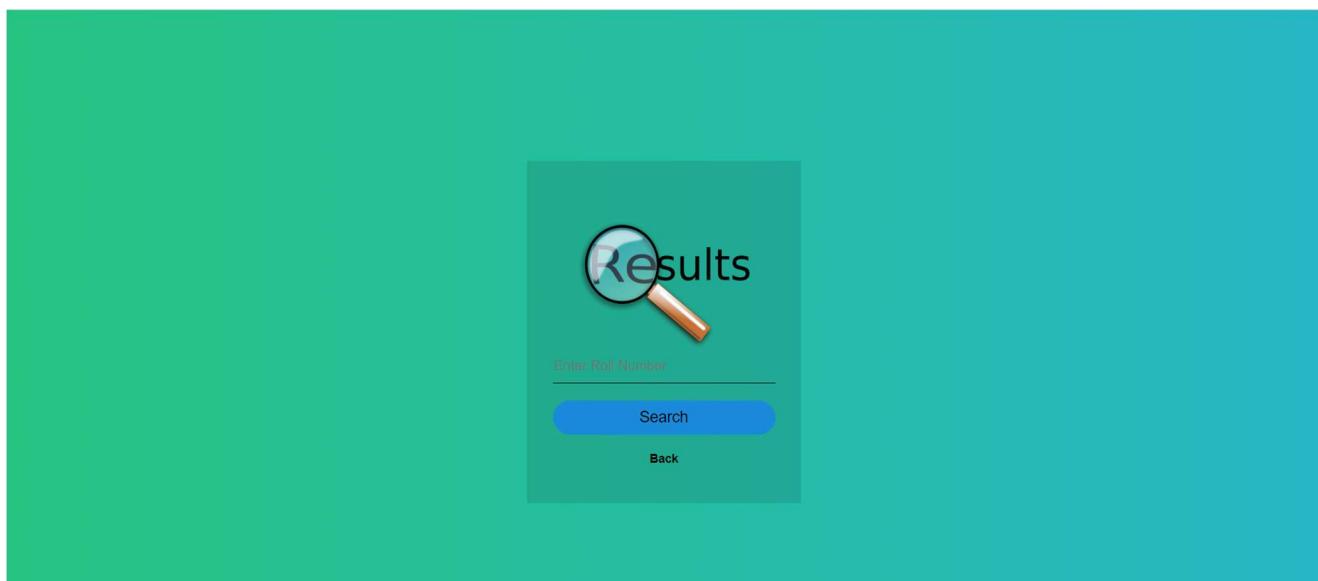
# **APPENDIX**

## **WEB PAGES SCREEN SHORT**

### **1. HOME SCREEN:**



### **2. RESULT CHECKING SCREEN FOR STUDENT:**



### 3. STUDENT RESULT SCREEN:

INSTITUTION NAME: MRSPTU	COURSE NAME:BCA-MCA DUAL DEGREE	BRANCH NAME: COMPUTATIONAL SCIENCES	ROLLNO: 191160014
NAME:GHANSHYAM KUMAR	FATHER NAME: KRISHNA DEV ROY	GENDER:MALE	

Course		Type	Full Marks	Passing Marks	Obtained Marks
Code	Name				
NAS101	Commutative English	Theory	100	30	78
NAS102	Information Technology	Theory	100	30	80
NAS103	Computer Organization	Theory	100	30	87
NEE101	Programming in C	Theory	100	30	87
NEC101	Human Value	Theory	100	30	78
<b>Total Marks</b>			500	180	410

### 4. RESULT PRINT SCREEN:

INSTITUTION NAME: MRSPTU	COURSE NAME:BCA-MCA DUAL DEGREE	BRANCH NAME: COMPUTATIONAL SCIENCES	ROLLNO: 191160014
NAME:GHANSHYAM KUMAR	FATHER NAME: KRISHNA DEV ROY	GENDER:MALE	

Course		Type	Full Marks	Passing Marks	Obtained Marks
Code	Name				
NAS101	Commutative English	Theory	100	30	78
NAS102	Information Technology	Theory	100	30	80
NAS103	Computer Organization	Theory	100	30	87
NEE101	Programming in C	Theory	100	30	87
NEC101	Human Value	Theory	100	30	78
<b>Total Marks</b>			500	180	410
<b>Percentage</b>			82		

Print                          1 sheet of paper

Destination: Microsoft Print to PDF

Pages: All

Layout: Portrait

Color: Color

More settings

Print      Cancel

INSTITUTION NAME: MRSPTU	COURSE NAME:BCA-MCA DUAL DEGREE	BRANCH NAME: COMPUTATIONAL SCIENCES	ROLLNO: 191160014
NAME:GHANSHYAM KUMAR	FATHER NAME: KRISHNA DEV ROY	GENDER:MALE	

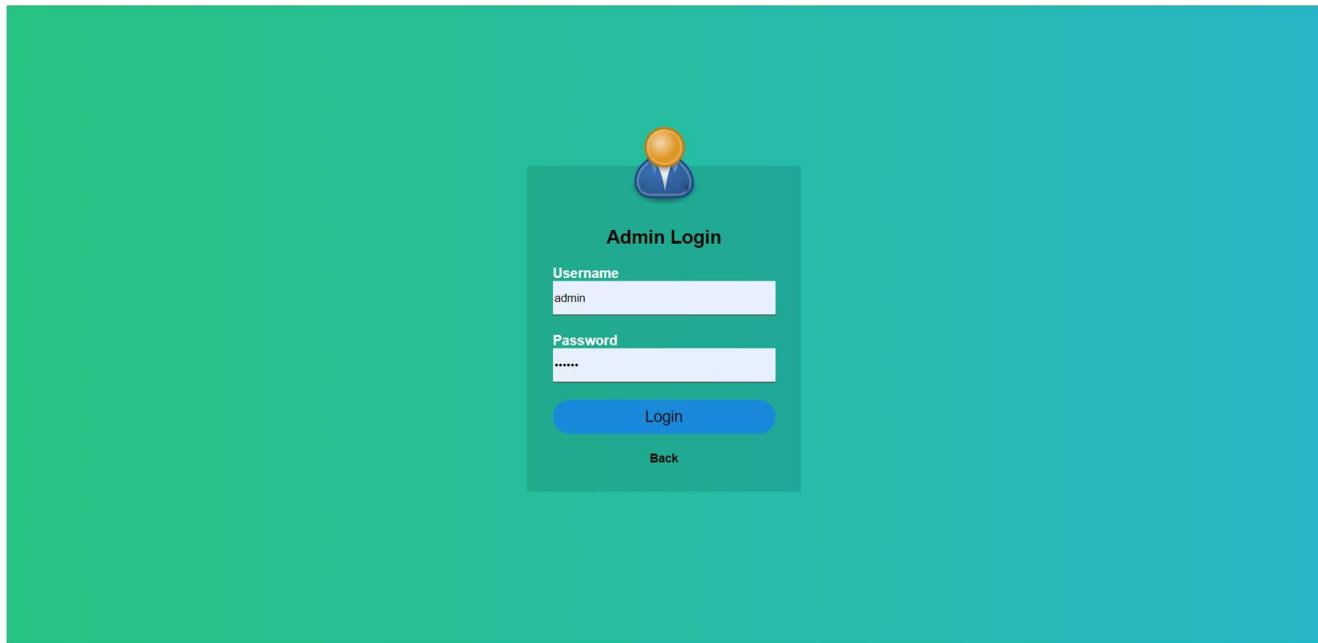
  

INSTITUTION NAME: MRSPTU	COURSE NAME:BCA-MCA DUAL DEGREE	BRANCH NAME: COMPUTATIONAL SCIENCES	ROLLNO: 191160014
NAME:GHANSHYAM KUMAR	FATHER NAME: KRISHNA DEV ROY	GENDER:MALE	

INSTITUTION NAME: MRSPTU	COURSE NAME:BCA-MCA DUAL DEGREE	BRANCH NAME: COMPUTATIONAL SCIENCES	ROLLNO: 191160014
NAME:GHANSHYAM KUMAR	FATHER NAME: KRISHNA DEV ROY	GENDER:MALE	

## 5. ADMIN LOGIN SCREEN:



## 6. ADD NEW STUDENT SCREEN:

A screenshot of a student management system. The header features the university logo and the text "MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY". The navigation bar includes links for "Add New Student" (highlighted in red), "Insert New Result", "Registered Students", "All Student Result", and "Logout". The main content area is titled "Add New Students". It contains several input fields: "Course Name" (BCA), "Branch Name" (COMPUTATIONAL SCIENCE), "Roll Number" (empty), "Name" (empty), "Father Name" (empty), and "Gender" (empty). A "Save" button is located at the bottom of the form.

## 7. INSERT NEW RESULT SCREEN:



MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY

Add New Student Insert New Result Registered Students All Student Result Logout

Insert New Result

Roll Number	191160014
Communicative English	
Information Technology	
Computer Organization	
Programming in c Language	
Human Value .	

## 8. ALL REGISTERED STUDENTS SCREEN:



MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY

Add New Student Insert New Result Registered Students All Student Result Logout

COURSE NAME	BRANCH NAME	ROLL NUMBER	NAME	FATHER NAME	GENDER	UPDATE	DELETE
BCA-MCA	COMPUTATIONAL SCIENCES	191160004	ARSH-DEEP KAUR	RAM SINGH	FEMALE	<input type="button" value="update"/>	<input type="button" value="delete"/>
BCA-MCA	COMPUTATIONAL SCIENCES	191160008	CHAMAN RAJ	PURANCHAND JOSHI	MALE	<input type="button" value="update"/>	<input type="button" value="delete"/>
BCA-MCA DUAL DEGREE	Computational Sciences	191160012	DHEERAJ KUMAR DAS	KRISHNA DEV ROY	MALE	<input type="button" value="update"/>	<input type="button" value="delete"/>
BCA-MCA DUAL DEGREE	Computational Sciences	191160014	GHANSHYAM KUMAR	KRISHNA DEV ROY	MALE	<input type="button" value="update"/>	<input type="button" value="delete"/>
BCA-MCA DUAL DEGREE	COMPUTATIONAL SCIENCES	191160023	HIMANSHU KASHYAP	JOGINDRA KASHYAP	MALE	<input type="button" value="update"/>	<input type="button" value="delete"/>

## 9. ALL REGISTERED STUDENTS RESULT SCREEN:



MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY

ROLLNO	ENGLISH COMMUNICATION	INFORMATION TECHNOLOGY	COMPUTER ORGANIZATION	C LANGUAGE	HUMAN VALUE	UPDATE	DELETE
191160004	70	69	71	65	69	<button>Update</button>	<button>Delete</button>
191160008	75	80	77	85	90	<button>Update</button>	<button>Delete</button>
191160012	74	75	76	77	78	<button>Update</button>	<button>Delete</button>
191160014	78	80	87	87	78	<button>Update</button>	<button>Delete</button>
191160023	85	84	83	90	92	<button>Update</button>	<button>Delete</button>

## 10. UPDATE STUDENT DETAILS SCREEN:

Update Student Details

Course	<input type="text" value="BCA-MCA DUAL DEGREE"/>
Branch	<input type="text" value="Computational Sciences"/>
Name	<input type="text" value="191160014"/>
Name	<input type="text" value="GHANSHYAM KUMAR"/>
Father Name	<input type="text" value="KRISHNA DEV ROY"/>
Gender	<input type="text" value="MALE"/>

## 11. UPDATE STUDENT RESULT SCREEN:

Update Student Result	
Rollno	191160014
Communative English	78
Information Technology	80
Computer Organization	87
C Language	87
Human Value	78

## 12. DELETE STUDENT RESULT BUTTON:

ROLLNO	ENGLISH COMMUNICATION	INFORMATION TECHNOLOGY	COMPUTER ORGANIZATION	C LANGUAGE	HUMAN VALUE	UPDATE	DELETE
191160004	70	69	71	65	69	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
191160008	75	80	77	85	90	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
191160012	74	75	76	77	78	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
191160014	78	80	87	87	78	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
191160023	85	84	83	90	92	<input type="button" value="Update"/>	<input type="button" value="Delete"/>

### 13. DELETE REGISTERED STUDENTS DETAILS BUTTON:

The screenshot shows a web-based student management system. At the top, there is a logo of the university and the text "MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY". Below the logo is a navigation bar with links: "Add New Student", "Insert New Result", "Registered Students" (which is highlighted in red), "All Student Result", and "Logout". The main content area displays a table of registered students with the following columns: COURSE NAME, BRANCH NAME, ROLL NUMBER, NAME, FATHER NAME, GENDER, UPDATE button, and DELETE button. A red oval highlights the "DELETE" buttons for all five rows. The data in the table is as follows:

COURSE NAME	BRANCH NAME	ROLL NUMBER	NAME	FATHER NAME	GENDER	UPDATE	DELETE
BCA-MCA	COMPUTATIONAL SCIENCES	191160004	ARSHDEEP KAUR	RAM SINGH	FEMALE	<button>update</button>	<button>delete</button>
BCA-MCA	COMPUTATIONAL SCIENCES	191160008	CHAMAN RAJ	PURANCHNAJ JOSHI	MALE	<button>update</button>	<button>delete</button>
BCA-MCA DUAL DEGREE	Computational Sciences	191160012	DHEERAJ KUMAR DAS	KRISHNA DEV ROY	MALE	<button>update</button>	<button>delete</button>
BCA-MCA DUAL DEGREE	Computational Sciences	191160014	GHANSHYAM KUMAR	KRISHNA DEV ROY	MALE	<button>update</button>	<button>delete</button>
BCA-MCA DUAL DEGREE	COMPUTATIONAL SCIENCES	191160023	HIMANSHU KASHYAP	JOGINDRA KASHYAP	MALE	<button>update</button>	<button>delete</button>