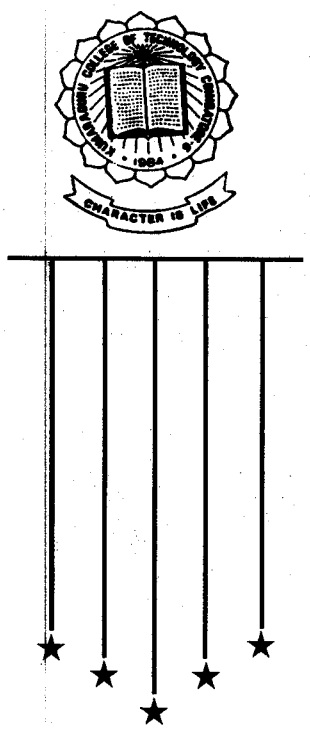
STUDENTS DETAILS AND ATTENDANCE MAINTANENCE

PROJECT WORK SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF **BACHELOR OF DEGREE IN**

B.SC APPLIED SCIENCE

COMPUTER TECHNOLOGY

OF THE BHARATHIAR UNIVERSITY



2000 - 2001

SUBMITTED BY

GOKILA. S

JAYALAKSHMI. V

GUIDED BY

Mr. S. ANDREWS, M.Sc., PGDPM.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE - 641006

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore - 641006

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that this Project Entitled

STUDENT DETAILS AND ATTTENDANCE MAINTANENCE

has been submitted by

Ms. **GOKILA .S** and **JAYALAKSHMI .V**

in partial fulfillment of the requirements for the award of Degree of Bachelor of Science in Applied Science Computer Technology of the Bharathiar University, Coimbatore - 641 046 during the academic year 2000 - 2001

(Guide)

13/3/01
(Head of the Department)

Certified that the Candidate was Examined by us in the Project Work

Viva-Voce Examination held on ...16.03.2001...

University Register Number .....9827Q0119..... and .....9827Q0124.....

16/03
(Internal Examiner)

16/3/2001
(External Examiner)

KUMARAGURU COLLEGE OF TECHNOLOGY

CHINNAVEDAMPATTI – COIMBATORE - 641 006

Date 09-03-2001

Department of Computer Science and Engineering

B.Sc. (CT) Stream

CERTIFICATE

This is to certify that (1) ***Ms. S. Gokila,*** *Reg.No.: 9827Q0119 and (2)* ***Ms. V. Jayalakshmi*** *Reg.No.: 9827Q0124 has developed the project titled "****Student Details & Attendance Maintenance****" for the B.Sc. Computer Technology Stream of our college is well tested and implemented.*

13/3/2001

Class Advisor
III B.Sc.(CT)

Course Co-ordinator

S. Thangaraj 13/3/01

HOD / CSE

Phone : 0422 - 866424, 866521, 868479 Fax : 0422 - 866667
E-mail : info@kct.ac.in / website : www.kct.ac.in

ACKNOWLEDGMENT

An endeavor over a long period can be successful only with the advice and support of many well wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We bound to express our gratitude to Dr.K.K.Padmanabhan B.Sc., (Engg), M.Tech, Ph.D., Principal, K.C.T College (Affiliated to Bharathiar University), Coimbatore, TamilNadu for his constant encouragement throughout my course.

We wish to thank Prof Dr S.Thangaswamy, Ph.D., Head, Dept of Computer science and Engineering, K.C.T College for constantly encouraging us to pursue new goals and ideas.

We admit our heartful thanks to our Internal Project Guide Mr. S. Andrews Msc, PGDPM. for supporting the tenure of our project and for having offered his valuable efforts in our project.

Finally we thank all our classmates, who gave whole-hearted support and encouragement. Their contributions are shown in many subtle ways and indeed instrumental in achieving final results.

SYNOPSIS

Analysing program is virtually important when it comes to writing software. This can be achieved by writing efficient programs and judging the relationship to performance.

Record Maintanence as the name implies, this is mainly designed for maintaining student details. This project is done for our college (Kumaraguru college of technology) maintaining records of all under graduate students. Details recorded are the gendral details, marks obtained during the course of study in each exam, meeting held with the parents regarding the performance of the student and the corrective measures taken, extra curricular activities of the student with the prize details and attendance of the student accounted on each day.

The main goal of this project is to minimize the problems of the existing system (i.e. manual work) and get rid of bulky papers. It mainly consists of four operations. Insertion, Updation, Deletion and Retrieval.

In Insertion operation end user inputs all details such as gendral information, marks obtained in all exams, information regarding parents meeting, extra curricular activities of the student and attendance accounted during college days. These details are made as permanent storage in the

database until it is deleted. Once a student has completed his course, details regarding him are deleted using deletion operation. Updation operation is done if any modification is to be done to any field. Retrieval operation is performed for viewing the details stored in the database.

The project has MSAccess as its backend and the most widely accepted web oriented Java Server Pages (JSP) as its frontend. HTML and Java Script is used for designing web page and for validation purpose. This project is designed in such a way it is user friendly with Look and Feel appearance, Portable and Flexible for future enhancement.

CONTENTS

1. INTRODUCTION
   - 1.1. Organisation Profile
   - 1.2 Purpose
   - 1.3 Programming environment
   - 1.4 Need for computerization
2. GENERAL DESCRIPTION
   - 2.1 Product perspective
   - 2.2 User characteristics
3. SPECIFIC REQUIREMENTS
   - 3.1 Functional Requirements
     - 3.1.1 Existing System
     - 3.1.2 Proposed System
   - 3.2 Performance Requirements
     - 3.2.1 Security
4. **LANGUAGE OVERVIEW**
5. DESIGN CONSTRAINTS
   - 5.1 Input Design
   - 5.2 Database Design
   - 5.3 Output Design
6. SYSTEM TESTING
   - 6.1 Code Testing
   - 6.1.1. Integration Testing
   - 6.1.2 Data Validation Testing
   - 6.1.3 Test Data
   - 6.2 Debugging
7. SCOPE FOR FUTURE DEVELOPMENT
8. CONCLUSION

APPENDIX
- Table Structure
- Sample source code
- Input-Output Screens

BIBLIOGRAPHY

INTRODUCTION

1.1 ORGANISATION PROFILE

KCT – A Technical Institution Par Excellence

Kumaraguru College of Technology, Coimbatore a private co-educational Engineering College was started in 1984 under the auspices of Ramananda Adigalar Foundation, a charitable and educational trust and a unit of Sakthi group. Located in a congenial environment on the outskirts of the city, just 3Kms from the city limits and 2Kms from Coimbatore – Mysore highway, the college campus spreads over a sprawling 150 acres and is well developed with black-topped roads, sodium vapour lamps, drainage, adequate power and water supply. The management has invested more than Rs. 35 Crores for the creation of infrastructural facilities.

Under the able guidance and patronage of Arutselver Dr. N. Mahalingam, Chairman, Sakthi Group and Prof. K.Arumugam, Correspondent, the college has developed excellent infrastructural facilities such as buildings, well equipped laboratories, 24 hours internet center, a earth station, workshops and well-qualified faculty. The college has been granted central government financial assistance to the tune of Rs. 20 lakhs for modernization and removal of obsolescence in the department of Computer

Science and Engineering, Mechanical Engineering and Civil Engineering under the Thrust area Scheme. The college offers under-graduate and post-graduate courses under affiliation to the Bharathiar University, Coimbatore and with the approval of All India Council for Thechnical Education (AICTE).

The caliber of our student and the well deserved reputation of the college for quality attracts leading companies to hold campus recruitment. In the last academic year more than 250 student have been placed through our Placement Centre.

The alumni have dine us proud by proving their worth wherever they have happened to be employed. All our 2700 alumni are happily settled in life after college.

1.2 PURPOSE

Record maintenance is done for all under graduate students of Kumaraguru College of technology. Records all the gendral details, marks obtained during the course of study in each exam, meeting held with the parents regarding the performance of the students and the corrective measures taken, extra curricular activities of the student with the prize details and attendance of the student accounted on each day.

The main purpose is it reduces manual maintanence of reports. All the details regarding a single student or a whole class can be obtained by specifying the roll number or the class name.

1.3 PROGRAMMING ENVIRONMENT

The following sections describe in detail various specifications needed for the application to perform.

HARDWARE REQUIREMENTS

Processor type	Pentium processor
RAM	32 MB RAM
Speed	350 MHz
Hard Disk	4.3 GB or Higher
Keyboard	TVS 104 Keys

Monitor (Visual Display) Display panel (640*480)
(Preferably Samtron 15" or LG Studio works 14"colour)

Floppy Drive	3.5", 1.44 MB
Mouse	Logitech serial mouse.

SOFTWARE REQUIREMENTS

Backend : Microsoft Access 2000.

Frontend : Win32 release for Windows 98 and NT on Intel hardware.

JavaServerPages(JSP)

Browser supporting HTML4.0 and JAVASCRIPT
(Preferably IE 5.0 or Higher)

Server : Javawebserver2.0

Connectivity : JavaDataBaseConnectivity

1.4 NEED FOR COMPUTERIASATION

The benefit of computerization over manual systems is many folds. The benefits can be stated as below.

SPEED

Computers enable us to do arithmetical calculations with fantastic speed and ease. It is possible to do things, which so far no one could think of attempting in a manual system. Tasks involving large voluminous data processing are thus done with much accuracy and speed by the computers than by manual system.

ACCURACY

One of the greatest benefits which computers can give us is that of accuracy. Practical experience has already shown that these machines are capable of achieving the degree of accuracy, which hitherto have been unattainable in certain accounting processes into which human brain enters at so many stages of the complete cycle of operations.

FLEXIBILITY

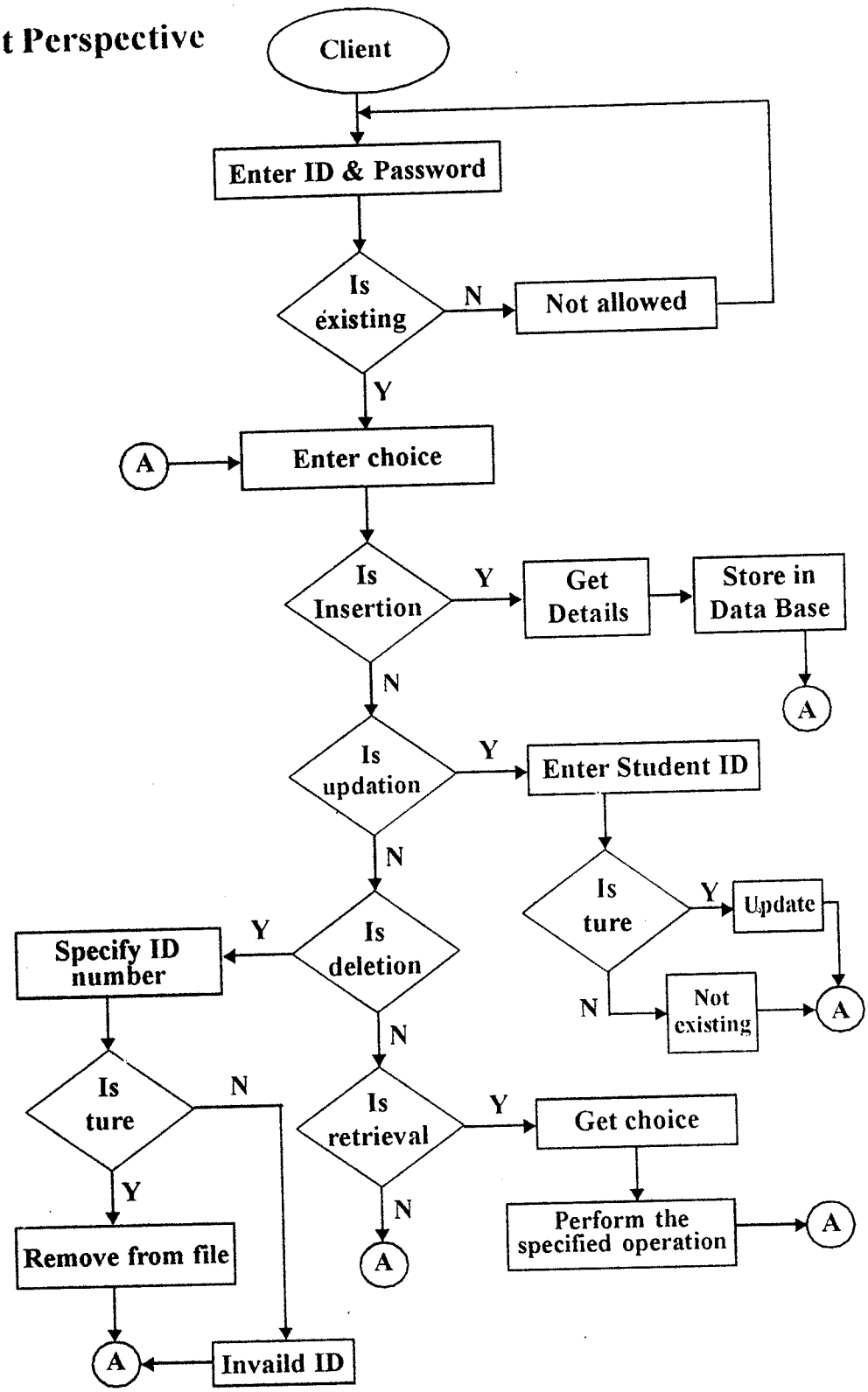
Flexibility in use is another important advantage of computers. Output can be obtained almost in whatever form it is most suitable.

MISCELLANEOUS

These include economics resulting from better managerial control, saving in labour because it is fully automatic.

2. GENERAL DESCRIPTION

2.1 Product Perspective



2.2 USER CHARACTERISTICS

The main users of this project are staffs (Class Advisors), who are responsible for maintaining the record. Daily attendance are fed by them, overall attendance percentage are calculated for each student at the end of the semester. Exam marks are fed for the entire student as and when evaluation gets completed, total, average and result are calculated and stored permanently. Retrieving, deletion or updation can be done for all records stored in the database.

3. SPECIFIC REQUIREMENTS

System analysis is a problem solving activity that requires intensive communication between the system requests and the system developer. System analysis is concerned with becoming aware of the problem identifying the relevant variables, analyzing and synthesing the various factors and determining an optimal at least a satisfactory solution or program of action. Information obtained through different processes such as gathering and interpreting facts, diagnosing problems, is used for recommended improvements to the new system.

3.1.1 EXISTING SYSTEM

Requirements analysis is done to get a sound knowledge about the existing system, its advantage and limitations to use effective methods to overcome these while developing the new system. To get a firm idea about the existing system certain methods were employed such as visiting the persons in charge of maintaining records collecting information regarding the problems faced by them. A clear idea is obtained about the existing system after requirement analysis.

3.1.1 EXISTING SYSTEM

Requirements analysis is done to get a sound knowledge about the existing system, its advantage and limitations to use effective methods to overcome these while developing the new system. To get a firm idea about the existing system certain methods were employed such as visiting the persons in charge of maintaining records collecting information regarding the problems faced by them. A clear idea is obtained about the existing system after requirement analysis.

The existing system is a manual system. Various activities such as insertion, deletion, updating etc are carried out in manual ways. And it depends upon several criteria such as what we want it to do, does it, work according to the original specification. Various process involved in this system consumes much human work and man power at the time even leading to loss of valuable data. These are virtually important when it comes to writing software, and specifications are made that they operate smoothly and effectively.

3.1.2 PROPOSED SYSTEM

Proposed system is aimed at removing the drawbacks of the existing system. Proposed system provides a computerized solution thus reducing manpower.

ECONOMIC FEASIBILITY

With the proposed system our institution can decrease the expense by reducing the cost of papers, and their maintanence, reducing manual power etc, and the main advantage is spotting of a single person or group of person with all the detail from date of join can be obtained during his course of study.

OPERATIONAL FEASIBILITY

Request for the proposed system made by the institution and the users of the existing system. The users of the existing system has been involved in the planning of the project from early stages. The users of the proposed system will get more facilities than the existing system. So we say the system is operationally feasible.

3.2 PERFORMANCE REQUIREMENTS

3.2.1. SECURITY

Security plays a vital role in all software project without having security, any software project is a failure security is especially important in a network. In a multiuser system, many users will be using the software. Hence the software should be protected from accidental or intentional damage. For this system, each and every user is given an user ID & Password. Only after validation of this ID & Password, the user is allowed to login into the system.

4. LANGUAGE OVERVIEW

INTRODUCTION TO HTML

HTML stands for Hypertext Markup Language, which is an application of Standard Generalized Markup Language (SGML). It is a simple language used to define and describe the layout of a web page. HTML also supports Multimedia and Document links.

HTML consists of special codes which when embedded in text, adds formatting. The special characters, which separate HTML from ordinary text, are the left and right brackets. (< >). These brackets contain instructions known as TAGS that are not case sensitive.

INTRODUCTION TO JAVASCRIPT

With the advent of Internet and World Wide Web, interactive communication at an exigency has become a necessity for mankind. The World Wide Web is a cluster of pages of information, combining text, pictures and sound. Each page has a hyperlink that refers to another page on the Net. The linked page can have further links to other such pages. This system of interlinked documents is called as **hypertext**.

Browsers help in manipulating pages of hypertext information to enhance interactivity between pages. One such Browser is the Netscape Navigator Gold (2.02). This Browser was developed by Netscape Communications, which incorporates Java, a programming language from Sun Microsystems and JavaScript language used to enhance the functionality of the Browsers. Netscape offers a host of tools to add value to their applications. These tools aid Web pages developers and authors in adding dynamic interaction to the information provided on the Internet.

Java script is integrated with HTML and Navigator 2.02. JavaScript facilitates the developer with properties related to document windows, frames, forms, loaded documents and links. The scripting language also traps user events so programs can be developed for such events. This is an interpreter-based language and source code files are directly executed at runtime. JavaScript includes built-in objects related to the current windows and documents as well as objects such as Math, String and Date that contains mathematical functions, string functions and Date functions respectively. Since JavaScript is an object-based language, it supports instances, methods and properties.

JAVA SERVER PAGE

A JSP page is a text-based document that describes how to process a request to create a response. JSP is a Java-based technology that simplifies the process of developing dynamic web sites. With, JSP web designers and developers can quickly incorporate dynamic elements into web pages using embedded java and simple markup tags. These tags provide the HTML designer with a way to access data and business logic stored inside java objects.

JavaServerPages are text files with the extension .jsp, which take the place of traditional HTML pages. JSP files contain traditional HTML along with embedded code that allows the developer to access data from the java code running on the server.

ADVANTAGES OF JSP

JSP offers several benefits as a system for dynamic content generation. As a Java-based technology, it enjoys all of the advantages that all of the advantages that the language provides with respect to development and deployment. As an object-oriented language with strong typing, encapsulation, exception handling, and automatic memory management, use of Java leads to increased programmer productivity and more robust code.

Because compiled Java bytecode is portable across all platforms, operating system, or server software. If a switch in any of these components becomes necessary, all JSP pages and associated Java classes can be migrated over as is. Because JSP is a vendor-neutral, developers and system architects can select best of breed solutions at all stages of JSP development. JSP technology is the platform technology for building applications containing dynamic web content such as HTML, DHTML, XHTML, and XML. The JavaServerPages technology enables the authoring of web pages that create dynamic content dynamic content easily but with maximum power and flexibility.

The Java Server Pages technology offers a number of advantages:

- Write once, Run Anywhere properties.
- High quality tool support.
- Reuse of components and tag libraries.
- Separation of dynamic and static content.
- Support for scripting and actions.
- Web access layer for N-tier enterprise application architecture(s).
- Performance.
- Splitting up presentation and implementation.

JSP AND JDBC CONCEPTS

The JDBC interface is a pure Java API used to execute SQL statements. The JDBC provides a set classes and interface that can be used by developers to write database applications.

Basic JDBC interaction, in its simplest form, can be broken down into four steps:

- Open a connection to the database.
- Execute a SQL statement.
- Process the results.
- Close the connection to the database.

5. DESIGN CONSTRAINTS

System design is a modeling process. It can be defined as a transition from a user's view to view of programmers (developers) and database personnel. It concentrates on translating requirements specification to design specification to design specification. This system design phase acts as a bridge between the requirement specification and the implementation phase.

The major steps in the design phase are design methodology and development, input design, output design, database design. First step is design methodology and development, which specifies various steps in system development life cycle. Next step is to define input/output screens and database design, which concentrates on choosing the database that suits most to the application environment.

5.1 INPUT DESIGN

In the input design the user oriented inputs are converted into computer recognizable forms. The collection of input data is the most expensive part of the system. In the input design data is accepted and it can be readily used for data processing or can be stored in database for further use. Input design is that part of design phase which requires the most attention. Data should be accurate because in accurate data is the most common cause for errors in data processing. The input screens are very user friendly. Different names are associated with each data entry screen and data item that makes data entry an easy job.

Each data entry screen contain separate buttons "SUBMIT" and "RESET" for submitting the form and resetting the form. While entering the data, proper validation checking are carried out and necessary messages will be alerted by the software if incorrect data has been entered.

Different web pages are created for each division. Insertion, updation, deletion and retrivation contain pages for gendral details, 1year and other semester marks, parents meeting information and extra curricular activities, and attendance information.

5.2 DATABASE DESIGN

Database design is an important part of the system design phase. In a database environment, the available data is used by several users. Instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity. The primary objective of a database design include fast response time to inquiries, more information at low cost, control of redundancy, clarity and integrity of the system, fast recovery and availability of a powerful end-user language. The theme behind a database is to handle information on a whole integrated thus making access to information easy, quick, inexpensive and flexible for the users.

Data directory specifies the major elements in the system, and care should be taken while designing, in order to avoid unnecessary duplication of data. The entire package depends on how the data are maintained in the system several tables are maintained in the system to store data required for processing of various data as well as storing intermediate or final processed results. Different modules to meet their needs access the stored data.

The different tables that are maintained in the system are gendral, 1year, other, refer, pmeet, active, uid, attendance.

5.3 OUTPUT DESIGN

In the output design, the emphasis is on producing the hardcopy as softcopy of information requested for outputs are the most important and direct source of information to the clients. Intelligent and formatted outputs will make it easier to understand. Outputs are also used to provide a permanent hardcopy of results for later considerations.

6. SYSTEM TESTING

This forms another major part of any system development process. care should be given during the whole process of testing. The performance of the system is measured in this phase.

6.1 SOURCE CODE TESTING

Testing is a process of executing a program with the interest of finding an error. A good test is one that has a high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with minimum effort. Two classes of inputs are provided to the test process. They are

a) A software configuration that includes a software requirement specification, a design specification and a source code.

b) A test configuration that includes a test plan and procedure, any testing tools that are to be used and test cases and their expected results.

Testing is divided into three distinct operations namely modular testing, integration testing and system testing. In the series of testing the following tests are implemented.

6.1.1 INTEGRATION TESTING

Though each program works individually, they should work after linking them together. This is also referred to as interfacing. Data may be last across interface and one module can have an adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at same time, conducting test to uncover errors associated with the interface. In the testing, the programs are constructed and tested in small segments.

6.1.2 DATA VALIDATION TESTING

Data validation is done to see whether the corresponding entries made in the tables are correct. Proper validation checks are done in case of insertion, deletion and updation of tables. Duplication of data must be avoided to the maximum extent. If any such cases arise, then proper error message or warnings, if any, has to be displayed. A double confirmation is made before deleting any specific entries. Whitebox testing is a test case design to divide the test cases.

The different test cases are :

1) Guarantee that all independent parts within a module have been exercised at least once.

2) Exercise all logical decisions on their true/false side.

3) Execute all loops at their boundaries and within their operational bounds.

4) Exercise internal data structure to ensure their validity.

Each module was tested and the tested modules were linked and integration was carried out.

6.1.3 TEST DATA

The system analyst will provide the test data, specially designed to show that the system will operate successfully in all its aspects and produce expected results under expected conditions. The test should take place at the same environment preparation of test data and the checking of results should be carried out in conjunction with the appropriate users and operational departments. The test objectives should be clear. Also the extent to which the system should be tested must be planned.

6.2 DEBUGGING

The potential ability of JAVA to handle exceptions was used extensively during the debugging process. All types of exceptions were caught and explicitly handled. Java language exceptions were caught generally whereas other exceptions like SOL exceptions were caught separately and various modes of their validation were found. Errors in case of back-end tables were used to display to the user in a number of ways. Exceptions occur at many stages like what happens during start of a program to any abnormal operations done or any missing threads. Various errors occurred in the cases of variables, which carried same name and caused a lot of problems when all modules were linked.

7. SCOPE FOR FUTURE DEVELOPMENT

This system is designed in such a way that addition of new modules can be done without much difficulty. That is viewing of a particular student can be extended to viewing any number of students details in a single screen. Photographs of each student can be attached. For a particular student details regarding his or her campus interview can also be included.

The system can be changed easily depending on changes in the institution policies. The reconstruction of system will increase the flexibility of the system. The things mentioned above will be carried out in the future.

8. CONCLUSION

This project meets all the requirements of record maintanence of an institution. This system has been developed as versatile and user friendly as possible keeping in mind the advanced features in growing technology.

Using JavaScript, JavaServerPages and MSAccess the system was developed and tested with all possible samples of data. As a whole, the system was well planned and designed.

The performance of the system proved to be efficient, it is user friendly and any user can get familiarized with its usage. The system provides flexibility for incorporating new features, which may be necessary in future.

DATABASE TABLE STRUCTURE

TABLE NAME : Gendral
PRIMARY KEY : Rollno

Field Name	Data Type	Description
Rollno	Text	Roll number of the student
Branch	Text	Branch name
Board	Text	Board of study
Admission	Text	Mode of Admission
Res	Text	Residence
Medium	Text	Medium of study
Community	Text	Name of the community
Name	Text	Name of the student
Date	Text	Date of birth
Caste	Text	Name of the caste
Address	Text	Address for communication
Paddress	Text	Permanent address for communication
Gaddress	Text	Guardian address if Hostellite
Regno	Text	Qualifying exam register number
Maths	Number	Marks in maths
Physics	Number	Marks in physics
Chemistry	Number	Marks in chemistry
Eregno	Text	Entrance register number
Emark	Number	Entrance marks (out of 100)
Totmark	Number	Qualifying marks (out of 300)
Fquali	Text	Father's educational qualification
Fname	Text	Father's name
Foccup	Text	Father's occupation with address
Mquali	Text	Mother's educational qualification
Mname	Text	Mother's name
Moccup	Text	Mother's occupation with address

TABLE NAME : 1YEAR
PRIMARY KEY : Rollno
ALTERNATE KEY : Exam

FIELD NAME	DATA TYPE	DESCRIPTION
Rollno	Text	Roll number of the student
Sem	Text	Semester in which student is studying
Exam	Text	Monthly/model/Semester Exam
Name	Text	Name of the student
M1	Number	Subject-1
M2	Number	Subject-2
M3	Number	Subject-3
M4	Number	Subject-4
M5	Number	Subject-5
M6	Number	Subject-6
M7	Number	Subject-7
M8	Number	Practical paper –1
M9	Number	Practical paper –2
M10	Number	Practical paper –3
Tutor1	Text	Tutor name
Tutor2	Text	Tutor name
Advisor	Text	Advisor name
Performance	Text	Performance of the student

TABLE NAME : OTHER
PRIMARY KEY : Rollno
ALTERNATE KEY : Exam

FIELD NAME	DATA TYPE	DESCRIPTION
Rollno	Text	Roll number of the student
Sem	Text	Semester in which student is studying
Exam	Text	Monthly/model/Semester Exam
Name	Text	Name of the student
M1	Number	Subject-1
M2	Number	Subject-2
M3	Number	Subject-3
M4	Number	Subject-4
M5	Number	Subject-5
M6	Number	Subject-6
M7	Number	Practical paper -1
M8	Number	Practical paper -2
Tutor1	Text	Tutor name
Tutor2	Text	Tutor name
Advisor	Text	Advisor name
Performance	Text	Performance of the student

TABLE NAME : PMEET
PRIMARY KEY : Rollno
ALTERNATE KEY : Date1

FIELD NAME	DATA TYPE	DESCRIPTION
Rollno	Text	Roll number of the student
Date1	Text	Date on which parents called
Sem	Text	Semester the student is studying
Name	Text	Name of the student
Reason	Text	Reason for which parents are called
Date2	Text	Date on which activities was taken
Nature	Text	Nature of activity
Measures	Text	Measures taken

TABLE NAME : ACTIVITIES
PRIMARY KEY : Rollno
ALTERNATE KEY : Date

FIELD NAME	DATA TYPE	DESCRIPTION
Rollno	Text	Roll number of the student
Date	Text	Date on which activity was conducted
Name	Text	Name of the student
Sem	Text	Semester in which the student is studying
Activities	Text	Activity on which student participated
Prize	Text	Prizes obtained

TABLE NAME : ATTENDANCE
PRIMARY KEY : Date
ALTERNATE KEY : Rollno

FIELD NAME	DATA TYPE	DESCRIPTION
Date	Text	Date along with month & year
Rollno	Text	Roll number of the student
Name	Text	Name of the student
Sem	Text	Semester in which student is studying
First	Yes/NO	First hour attendance
Second	Yes/NO	Second hour attendance
Third	Yes/NO	Third hour attendance
Fourth	Yes/NO	Fourth hour attendance
Fifth	Yes/NO	Fifth hour attendance
Sixth	Yes/NO	Sixth hour attendance
Seventh	Yes/NO	Seventh hour attendance

TABLE NAME : UID
PRIMARY KEY : Name
ALTERNATE KEY : Pid

FIELD	DATA TYPE	DESCRIPTION
Name	Text	Name of the login user
Pass	Text	Password ID

```
// LOGIN.HTML//
<HTML>
<script>
var count=0;
function verify(pass){
pass.name.value=pass.name.value.toUpperCase();
var name=pass.name.value;
valiname(name);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');}
else {
if(count==0)    {
pass.submit(); }  } }
function valiname(nu){
var len=nu.length;
if(len==0) {
document.writeln('<font size 4><br> ENTER STUDENTS
NAME(WITH INITIAL AT THE LAST)');
count=1;}
else{
for(i=0;i<len;i++){
var str=nu.substring(i,i+1);
if(!((str>='A' && str<='Z') || (str=='.') || (str==' '))) {
document.writeln('<br>ENTER STUDENTS NAME ');
count=1;
break;}  } }}
</script>
<body bgcolor=black text=red>
<form name=pass action=frontpage.jsp>
<pre>
<center>
<br>USER NAME    <input type=text name=name ><br>
<br>USER PASSWORD <input type=password name=pass><br>
<br><input type=submit value='  OK ' OnClick='verify(this.form);'>
<input type=reset value=CANCEL >
```

```
</center>
</center>
</pre>
</form>
</body>
</html>


//FRONTPAGE.JSP//
<%@ page language = "java" import = "java.sql.*" %>
<%!
Connection c=null;
Statement s=null;
ResultSet rs=null;
String name=new String();
String pass=new String();
%>
<%
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
name=request.getParameter("name");
pass=request.getParameter("pass");
rs=s.executeQuery("select * from uid where name='"+name+"' and
pass='"+pass+"' ");
if(rs.next()){
out.println("<html>");
out.println("<body bgcolor=black text=lime>");
out.println("<i>");
out.println("<font size=10>STUDENTS
RECORD</th></tr></table>");
out.println("</i>");
out.println("</font>");
out.println("<h3>");
out.println("<cite>");
out.println("<p align=left>");
out.println("<ul>");
out.println("<li><a href='insert.html'>INSERTING</a>");
```

```
out.println("  <li><a href ='update.html'> UPDATING </a>");
out.println("  <li><a href ='delete.html'> DELETING  </a>");
out.println("  <li><a href ='disp.html'>DISPLAYING</a></b>");
out.println("</ul>");
out.println("</p>");
out.println("</cite>");
out.println("</h3>");
out.println("</body>");
out.println("</html>");
}
else{
out.println("NAME AND PASSWORD ENTERED IS INVALID");
}}catch(SQLException e){
      out.println(e);
}
%>

//INSERT.HTML//
<html>
<body bgcolor=GRAY text=white>
<H1>
 INSERTION OF STUDENT DETAILS
</H1>
<h3>
<font color= black>
<ul>
<li><a href=insert1.html>GENERAL INFORMATION</a>
<li><a href=mark.html>MARKS</a>
<li><a href=insert3.html>PARENTS MEETING
INFORMATION</a>
<li><a href=insert4.html>EXTRACURRICULAR ACTIVITIES</a>
<li><a href=attend.html>ATTENDANCE
INFORMATION</a></font>f
</ul>
</h3>
</body>
</html>

//ATTEND.HTML//
```

```
<html>
<script>
var count=0;
function verify(attend){
attend.rollno.value=attend.rollno.value.toUpperCase();
attend.name.value=attend.name.value.toUpperCase();
var name=attend.name.value;
var rollno=attend.rollno.value;
var date=attend.date.value;
var first=attend.first.value;
var second=attend.second.value;
var third=attend.third.value;
var fourth=attend.fourth.value;
var fifth=attend.fifth.value;
var sixth=attend.sixth.value;
var seventh=attend.seventh.value;
valiname(name);
valirollno(rollno);
validate(date);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
 }
else  {
if(count==0){
attend.submit();}}
function valiname(no){
var len=no.length;
if(len==0)  {
document.writeln('<br>ENTER CORRECT NAME');
count=1;
}
else {
for(i=0;i<len;i++)  {
var str=no.substring(i,i+1);
if(!((str>='A' && str<='Z') || (str=='.') || (str==' ')))  {
```

```
document.writeln('<br>ENTER CORRECT NAME');
count=1;
break;
}} }}
function valirollno(no,attend){
var len=no.length;
if((len==0) || (len!=6 && len!=7)) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
}
else{
if(len==6)   {
for(i=0;i<len;i++)    {
var str=no.substring(0,1);
if(!(str>='0' && str<='9'))       {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9'))       {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(4,5);
<html>
<script>
var count=0;
function verify(attend)
{
attend.rollno.value=attend.rollno.value.toUpperCase();
attend.name.value=attend.name.value.toUpperCase();
var name=attend.name.value;
var rollno=attend.rollno.value;
var date=attend.date.value;
var first=attend.first.value;
var second=attend.second.value;
var third=attend.third.value;
```

```
var fourth=attend.fourth.value;
var fifth=attend.fifth.value;
var sixth=attend.sixth.value;
var seventh=attend.seventh.value;
valiname(name);
valirollno(rollno);
validate(date);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
}
else {
if(count==0) {
attend.submit(); } }
function valiname(no){
var len=no.length;
if(len==0){
document.writeln('<br>ENTER CORRECT NAME');
count=1;
}
else {
fo(i=0;i<len;i++) {
var str=no.substring(i,i+1);
if(!((str>='A' && str<='Z') || (str=='.') || (str==' '))){
document.writeln('<br>ENTER CORRECT NAME');
count=1;
break;}} }}
function valirollno(no,attend){
var len=no.length;
if((len==0) || (len!=6 && len!=7)){
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
}
else {
if(len==6) {
for(i=0;i<len;i++) {
```

```
var str=no.substring(0,1);
if(!(str>='0' && str<='9'))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9'))   {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='0' && str<='9'))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;}    }   }
else   {
if(len==7)    {
for(i=0;i<len;i++)      {
var str=no.substring(0,1);
```

```
if(!(str>='0' && str<='9'))        {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;        }
var str=no.substring(6,len);
if(!(str>='0' && str<='9'))   {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(5,6);
if(!(str>='0' && str<='9'))        {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))     {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))   {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))        {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='A' && str<='Z'))        {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;} } } } } }
function validate(no){
```

```
var len=no.length;
if(len!=10) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
}
else
{
for(i=0;i<len;i++)   {
var str=no.substring(2,3);
var str1=no.substring(5,6);
if((str!='/') || (str1!='/'))     {
document.writeln('<br>ENTER CORRECT
DATE(DD/MM/YEAR)');
count=1;
break;
}
var str=(no.substring(0,2));
var str1=(no.substring(3,5));
if(!(str1>='01' && str1<='12'))     {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;
}
if(str1=='01' || str1=='03' || str1=='05' || str1=='07' || str1=='08' ||
str1=='10' || str1=='12')
if(!((str>='01') && (str<='31')))   {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;
}
if(str1=='04' || str1=='06' || str1=='09' || str1=='11')
if(!((str>='01') && (str<='30')))       {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;
}
var str2=(no.substring(6,10));
if((str2%4)==0 && str1=='02')
if(!((str>='01') && (str<='29'))) {
```

```
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;
}
if((str2%4)!=0 && str1=='02')
if(!((str>='01') && (str<='28')))    {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;
} } } }}
</script>
<body bgcolor=olive text=black>
<h3>
<center>ATTENDANCE INSERTION<br>
<marquee behavior=alternate width=150>``` ,,, ``` ,,, ``` ,,, ``` ,,, ```
</h3></marquee></center>
<p align=left>
<form name=attend action=attendance.jsp>
<pre>
NAME       : <input type=text name=name><br>
ROLLNO     : <input type=text name=rollno><br>
DATE       : <input type=text name=date
value="01/01/2000"><br><br>
SEMESTER   : <select name=sem>
      <option selected> I-SEMESTER </option>
      <option> II-SEMESTER</option>
      <option> III-SEMESTER</option>
      <option> IV-SEMESTER</option>
      <option> V-SEMESTER</option>
      <option> VI-SEMESTER</option>
      <option> VII-SEMESTER</option>
      <option> VIII-SEMESTER</option>
</select>
<center><u><b>ATTENDANCE</b></u></center>
<pre>
I HOUR : <select name=first>
        <option selected>YES</option>
        <option>NO</option>
</select>   II HOUR : <select name=second>
```

```
        <option selected> YES </option>
        <option> NO </option>
</select>  III HOUR : <select name=third>
        <option selected> YES </option>
        <option> NO </option>
</select>  IV HOUR : <select name=fourth>
        <option selected> YES </option>
        <option> NO </option>
</select><br>
V HOUR : <select name=fifth>
        <option selected> YES </option>
        <option> NO </option>
</select>  VI HOUR : <select name=sixth>
        <option selected> YES</option>
        <option> NO </option>
</select>  VII HOUR : <select name=seventh>
        <option selected> YES </option>
        <option> NO </option>
</select><br>
</pre>
<center>
<input type=submit value='SUBMIT' style="color: #000000;
background-color: #800000; font-family: Fixedsys; text-transform:
uppercase; text-decoration: blink; font-style: italic; font-weight: bold;
border-style: solid" OnClick='verify(this.form);'>
</center>
</body>
</form>
</html>
```

```
//ATTENDANCE.JSP//
<%@ page language="java" import="java.sql.*" %>
<%!
Connection c=null;
Statement s =null;
ResultSet rs=null;
String rollno,sem,name,date;
```

```
boolean first,second,third,fourth,fifth,sixth,seventh;
%><%
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
String first1,second,third,fourth,fifth,sixth,seventh;
rollno=request.getParameter("rollno");
date=request.getParameter("date");
sem=request.getParameter("sem");
first1=request.getParameter("first");
second1=request.getParameter("second");
third1=request.getParameter("third");
fourth1=request.getParameter("fourth");
fifth1=request.getParameter("fifth");
sixth1=request.getParameter("sixth");
seventh1=request.getParameter("seventh");
if(first1.equals("YES"))  first=true;
else
first=false;
if(second1.equals("YES")) second=true;
else
second=false;
if(third1.equals("YES"))  third=true;
else
third=false;
if(fourth1.equals("YES"))  fourth=true;
else
fourth=false;
if(fifth1.equals("YES")) fifth=true;
else
fifth=false;
if(sixth1.equals("YES")) sixth=true;
else
sixth=false;
if(seventh1.equals("YES")) seventh=true;
else
seventh=false;
```

```
rs=s.executeQuery("select * from attendance where
rollno='"+rollno+"' and dat='"+date+"' and sem='"+sem+"'");
if(!rs.next()){
c.commit();
rs=s.executeQuery("select name from refer where
rollno='"+rollno+"'");
rs.next();
name=rs.getString("name");
c.commit();
s.executeUpdate("insert into attendance values
('"+date+"','"+rollno+"','"+name+"','"+sem+"','"+first+","+second+","+
third+","+fourth+","+fifth+","+sixth+","+seventh+")");
out.println("<body><font size=6>");
out.println("ATTENDANCE OF  "+name+"("+rollno+") ON "+
date+" HAS BEEN SUCCESSFULLY INSERTED");
out.println("</body>");}
else {
out.println("<body><font size=6>");
out.println("ATTENDANCE OF  "+name+"("+rollno+") ON "+
date+"  ALREADY EXIST ");
out.println("</body>");
} }catch(Exception e){
out.println("VERIFY:"+e);
}
%>

//UPDATE.HTML//

<html>
<body bgcolor=black text=white>
<font size=10> UPDATION OF STUDENT DETAILS</H1></font>
<h1>
<ul>
<li><a href=update1-1.html>GENERAL INFORMATION</a>
<li><a href=upmark.html>MARKS</a>
<li><a href=update3-1.html>PARENTS MEETING
INFORMATION</a>
<li><a href=update4-1.html>EXTRACURRICULAR
ACTIVITIES</a>
```

```
<li><a href=upattend1.html>ATTENDANCE INFORMATION</a>
</ul>
</h1>
</font>
</body>
</html>

//UPDATE4-1.HTML//
<HTML>
<script>
var count =0;
function verify(up4){
update4.rollno.value=update4.rollno.value.toUpperCase();
var rollno=update4.rollno.value;
var date=update4.date.value;
valirollno(rollno);
validate(date);
if(count==1){
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
}
else{
if(count==0)   {
update4.submit();
}}}
function valirollno(no){
var len=no.length;
if((len==0)||(len!=6 && len!=7))  {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
}
else  {
if(len==6)   {
for(i=0;i<len;i++)    {
var str=no.substring(0,1);
if(!(str>='0' && str<='9'))       {
```

```
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9'))       {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='0' && str<='9'))       {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')){
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
break;}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))       {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
break;}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))       {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
break;} } }
else{
if(len==7)     {
for(i=0;i<len;i++)       {
var str=no.substring(0,1);
if(!(str>='0' && str<='9')){
document.writeln('<br>ENTER   CORRECT ROLLNO ');
count=1;
break; }
var str=no.substring(6,len);
```

```
if(!(str>='0' && str<='9'))        {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;}
var str=no.substring(5,6);
if(!(str>='0' && str<='9'))        {
document.writeln("<br>ENTER CORRECT ROLLNO");
count=1;
break;}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))     {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;break;}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))       {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;}
var str=no.substring(4,5);
if(!(str>='A' && str<='Z'))        {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;}}}}}}
function validate(no){
var len=no.length;
if(len!=10){
document.writeln('<br>ENTER CORRECT DATE ');
count=1;}
else  {
for(i=0;i<len;i++)    {
var str=no.substring(2,3);
var str1=no.substring(5,6);
if((str!='/') || (str1!='/'))    {
```

```
document.writeln('<br>ENTER CORRECT
DATE(DD/MM/YEAR)');
count=1;
break;}
var str=(no.substring(0,2));
var str1=(no.substring(3,5));
if(!(str1>='01' && str1<='12')) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;}
if(str1=='01' || str1=='03' || str1=='05' || str1=='07' || str1=='08' ||
str1=='10' || str1=='12')
if(!((str>='01') && (str<='31'))) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;}
if(str1=='04' || str1=='06' || str1=='09' || str1=='11')
if(!((str>='01') && (str<='30'))) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break; }
var str2=(no.substring(6,10));
if((str2%4)==0 && str1=='02')
if(!((str>='01') && (str<='29'))) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break;}
if((str2%4)!=0 && str1=='02')
if(!((str>='01') && (str<='28'))) {
document.writeln('<br>ENTER CORRECT DATE ');
count=1;
break;}}}}
</script>
<body bgcolor=maroon text=white>
<form name=update4 action=update42.jsp>
<center><h2><b><i> UPDATING EXTRA CURRICULAR
ACTIVITIES DETAILS</i></h2></b></center> <br>
<h3> SPECIFY ROLLNO,DATE </h3><br>
ROLL NO : <input type=text name=rollno ><br>
```

```
<br> DATE : <input type=text name=date > [FORMAT
01/01/2000]<br>
<br>ACTIVITIES : <select name=activities>
        <option selected> SPORTS</option>
        <option>NCC</option>
        <option>NSS</option>
        <option>MUSIC CLUB </option>
        <option>PAPER PRESENTATION</option>
</select><br>
<br><center><input type=button value='submit'
OnClick='verify(this.from);'></center>
</form>
</body>
</html>
```

```
//UPDATE42.JSP//
<%@ page language="java" import="java.sql.*" %>
<%!
Connection c=null;
Statement s =null;
ResultSet rs=null;
String date,rollno,activities;%>
<%
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
rollno=request.getParameter("rollno");
date=request.getParameter("date");
activities=request.getParameter("activities");
rs=s.executeQuery("select * from active where rollno='"+rollno+"' and
dat='"+date+"' and activities='"+activities+"' ");
out.println("<body bgcolor=gray>");
out.println("<font size=6>");
if(rs.next())   {
out.println("<br>ROLL NO :" +rollno);
out.println("<br>DATE :" +date);
```

```
out.println("<br>ACTIVITIES :" +activities);
out.println("<br>PRESS THE BUTTON TO UPDATE THE
DETAILS");
out.println("<form name=meets action=update42ht.jsp>");
out.println("<input type=hidden name=rollno value="+rollno+"> ");
out.println("<input type=hidden name=date value="+date+" > ");
out.println("<input type=hidden name=activities value='"+activities+"'
> ");
out.println("<input type=submit value='PRESS' > ");
out.println("</form>");}
else {
out.println("<br>SPECIFIED ROLLNUMBER IS NOT INSERTED
PREVIOUSLY\n");
out.println("<br>GO INSERT EXTRA CURRICULAR ACTIVITIES
OF "+rollno+" FOR THE EVENT "+activities);
out.println("<br>AFTER MAKING CORRECTIONS CONTINUE
UPDATING THE DETAILS");}
c.commit();
out.println("</font>");
}catch(SQLException e) {
out.println(e);}
%>
```

//UPDATE42HT.JSP//

```
<%@ page language="java" import="java.sql.*" %>
<%!
Connection c=null;
Statement s =null;
ResultSet rs=null;
String name=new String();
String date=new String();
String rollno=new String();
String activities=new String();
String sem=new String();
String price=new String();
String sign=new String();
```

```
%>
<%
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
rollno=request.getParameter("rollno");
date=request.getParameter("date");
activities=request.getParameter("activities");
rs=s.executeQuery("select * from active where rollno='"+rollno+"' and
dat='"+date+"' and activities='"+activities+"' ");
rs.next();
rollno=rs.getString("rollno");
name=rs.getString("name");
date=rs.getString("dat");
activities=rs.getString("activities");
sem=rs.getString("sem");
price=rs.getString("price");
c.commit();
}catch(SQLException e){
out.println(e); }
%>
<html>
<script>
var count=0;
function verify(extra) {
extra.rollno.value=extra.rollno.value.toUpperCase();
var rollno=extra.rollno.value;
var date=extra.date.value;
valirollno(rollno);
validate(date);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
 }
 else {
```

```
if(count==0)    {
extra.submit();
} }

function valirollno(no) {
var len=no.length;
if((len==0)  || (len!=6 && len!=7))    {
document.writeln('<br>ENTER CORRECT ROLL NUMBER ');
count=1; }
else {
if(len==6)      {
for(i=0;i<len;i++)        {
var str=no.substring(0,1);
if(!(str>='0' && str<='9'))         {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;     }
var str=no.substring(5,len);
if(!(str>='0' && str<='9'))         {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(4,5);
if(!(str>='0' && str<='9'))      {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))         {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))         {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9'))) {
```

```
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }}}
else {
if(len==7)   {
for(i=0;i<len;i++) {
var str=no.substring(0,1);
if(!(str>='0' && str<='9'))     {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(6,len);
if(!(str>='0' && str<='9'))    {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(5,6);
if(!(str>='0' && str<='9'))     {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(1,2);
if(!((str>='A'&&str<='Z')||(str>='0'&&str<='9'))){
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;break; }
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(3,4);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }
var str=no.substring(4,5);
if(!(str>='A' && str<='Z'))   {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
```

```
break;}}}}}}
function validate(no) {
var len=no.length;
if(len!=10)   {
document.writeln('<br>ENTER CORRECT DATE ');
count=1;   }
else   {
for(i=0;i<len;i++)   {
var str=no.substring(2,3);
var str1=no.substring(5,6);
if((str!='/') || (str1!='/'))      {
document.writeln('<br>ENTER CORRECT
DATE(DD/MM/YEAR)');
count=1;
break; }
var str=(no.substring(0,2));
var str1=(no.substring(3,5));
if(!(str1>='01' && str1<='12')) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break; }
if(str1=='01' || str1=='03' || str1=='05' || str1=='07' || str1=='08' ||
str1=='10' || str1=='12')
if(!((str>='01') && (str<='31'))) {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break; }
if(str1=='04' || str1=='06' || str1=='09' || str1=='11')
if(!((str>='01') && (str<='30')))    {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break; }
var str2=(no.substring(6,10));
if((str2%4)==0 && str1=='02')
if(!((str>='01') && (str<='29')))    {
document.writeln('<br>ENTER CORRECT DATE');
count=1;
break; }
if((str2%4)!=0 && str1=='02')
```

```
if(!((str>='01') && (str<='28'))) {
document.writeln('<br>ENTER CORRECT DATE ');
count=1;
break; }}}}}
</script>
<body bgcolor=black text=fuschia>
<marquee>
<h2>
UPDATION OF EXTRACURRICULAR ACTIVITIES
</h2>
</marquee>
<form name=extra action=update4-3.jsp>
<input type=hidden name=rollno value=<%=rollno%>>
<input type=hidden name=name value="<%=name%>">
<input type=hidden name=date value=<%=date%> >
<input type=hidden name=activities value="<%=activities%>">
<h4> <pre><font size=4>
ROLL NO        :<%=rollno%>
NAME           :<%=name%>
DATE           :<%=date%>
ACTIVITIES     :<%=activities%>
SEMESTER       :<select name=sem>
<option selected><%=sem%></option>
<option>I SEM </option>
<option> II SEM </option>
<option> III SEM </option>
<option> IV SEM </option>
<option> V SEM </option>
<option> VI SEM </option>
<option> VII SEM </option>
<option> VIII SEM </option>
</select>
PRIZES OBTAINED : <input type=text name=price
value=<%=price%>>
<center>
<input type=submit value='SUBMIT' OnClick='verify(this.form);'>
</center></form> </pre></font></h4>
</body></html>
```

```
//UPDATE4-3.JSP//
<%@ page language="java" import="java.sql.*" %>
<%
Connection c=null;
Statement s =null;
ResultSet rs=null;
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
String dat,rollno,activities,sem,price,sign;
rollno=request.getParameter("rollno");
dat=request.getParameter("date");
sem=request.getParameter("sem");
activities=request.getParameter("activities");
price=request.getParameter("price");
sign=request.getParameter("sign");
s.executeUpdate("update active set sem='"+sem+"',price='"+price+"'
where rollno='"+rollno+"' and dat='"+dat+"' and
activities='"+activities+"'");
out.println("INFORMATION HAVE BEEN SUCCESSFULLY
UPDATED");
c.commit();
}catch(Exception e) {
out.println("VERIFY:" +e); }
%>
```

```
//DELETE.HTML//
<html>
```

```
<script>
var count=0;
function verify(delet){
delet.rollno.value=delet.rollno.value.toUpperCase();
delet.name.value=delet.name.value.toUpperCase();
var name=delet.rollno.value;
var rollno=delet.name.value;
valiname(name);
valirollno(rollno);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
}
else {
if(count==0)    {
delet.submit();
} } }
function valirollno(no){
var len=no.length;
if((len==0) || (len!=6 && len!=7)){
document.writeln('<br><FONT SIZE=4>ENTER CORRECT ROLL
NUMBER');
count=1;
}
else {
if(len==6)   {
for(i=0;i<len;i++)   {
var str=no.substring(0,1);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
```

```
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='0' && str<='9'))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;   }
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='a' && str<='z') || (str>='A' && str<='Z') || (str>="0" &&
str<="9"))) {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;}}}
else{
if(len==7)   {
for(i=0;i<len;i++) {
var str=no.substring(0,1);
if(!(str>="0" && str<="9"))  {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;}
var str=no.substring(6,len);
if(!(str>="0" && str<="9"))  {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;}
```

```
var str=no.substring(5,6);
if(!(str>="0" && str<="9"))  {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;break;}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z'))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;}
var str=no.substring(4,5);
if(!(str>='A' && str<='Z'))  {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}}}}}
function valiname(nu){
var len=nu.length;
if(len==0)  {
document.writeln('<font size=4><br> ENTER STUDENTS
NAME(WITH INITIAL AT THE LAST)');
count=1;}
else {
for(i=0;i<len;i++) {
var str=nu.substring(i,i+1);
if(!((str>='A' && str<='Z') || (str=='.') || (str==' ')))  {
document.writeln('<br>ENTER STUDENTS NAME ');
count=1;
break;
}}}}
```

```
</script>
<body bgcolor=black text=white>
<form name=delet action=deleting.jsp >
<h2>
 DELETING A STUDENTS RECORD
<h2>
<br>
<pre>
NAME   :<input type=text name=name><br>
ROLLNO :<input type=text name=rollno><br>
<br><br>
<center>
<input type=submit value=DELETE style="font-family: Arial Black;
color: #800000; text-decoration: blink; font-weight: bold; border-style:
groove; border-color: #008000"OnClick=verify(this.form);>
</center>
</form>
</body>
</html>
```

```
//DELETE.JSP//

<%@ page language="java" import="java.sql.*" %>
<%
Connection c=null;
Statement s =null;
ResultSet rs=null;
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
String rollno;
rollno=request.getParameter("rollno");
s..executeUpdate("delete from gendral where rollno='"+rollno+"'");
c.commit();
s.executeUpdate("delete from 1year where rollno='"+rollno+"' ");
```

```
c.commit();
s.executeUpdate("delete from other where rollno='"+rollno+"' ");
c.commit();
s.executeUpdate("delete from pmeet where rollno='"+rollno+"' ");
c.commit();
s.executeUpdate("delete from active where rollno='"+rollno+"' ");
c.commit();
s.executeUpdate("delete from attendance where rollno='"+rollno+"' ");
c.commit();
s.executeUpdate("delete from refer where rollno='"+rollno+"' ");
c.commit();
}catch(Exception e)
{
out.println("VERIFY:" +e);
}
%>

//OVSI.HTML//

<html>
<script>
var count=0;
function verify(single){
single.rollno.value=single.rollno.value.toUpperCase();
single.name.value=single.name.value.toUpperCase();
var rollno=single.rollno.value;
var name=single.name.value;
valirollno(rollno);
valiname(name);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
}
else {
if(count==0)  {
single.submit();
```

```
} }}
function valirollno(no){
var len=no.length;
if((len==0) || (len!=6 && len!=7)) {
document.writeln('<br><FONT SIZE=4>ENTER CORRECT ROLL
NUMBER');
count=1;
}
else {
if(len==6) {
for(i=0;i<len;i++) {
var str=no.substring(0,1);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9')){
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
```

```
break;
}
var str=no.substring(1,2);
if(!((str>='a' && str<='z') || (str>='A' && str<='Z') || (str>="0" &&
str<="9"))) {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break; }}}
else {
if(len==7) {
for(i=0;i<len;i++) {
var str=no.substring(0,1);
if(!(str>="0" && str<="9")) {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;
}
var str=no.substring(6,len);
if(!(str>="0" && str<="9")) {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;
}
var str=no.substring(5,6);
if(!(str>="0" && str<="9")) {
document.writeln("<br>ENTER CORRECT ROLL NUMBER");
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9'))) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
```

```
}
var str=no.substring(3,4);
if(!(str>= 'A' && str<= 'Z')) {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='A' && str<='Z'))        {
document.writeln('<br>ENTER CORRECT ROLL NUMBER');
count=1;
break; }}}}}}
function valiname(nu){
var len=nu.length;
if(len==0) {
document.writeln('<font size=4><br> ENTER STUDENTS
NAME(WITH INITIAL AT THE LAST)');
count=1;
}
else {
for(i=0;i<len;i++) {
var str=nu.substring(i,i+1);
if(!((str>='A' && str<='Z') || (str=='.') || (str==' '))) {
document.writeln('<br>ENTER STUDENTS NAME 1');
count=1;
break; }}}}
</script>
<body bgcolor=black text=olive>
<form name=single action = ovsi2.jsp>
<center><h3><B><I><U> OVERALL PERCENTAGE OF ONE
STUDENT </B></U></I></H3><br></center>
<h4> SPECIFY ROLL NUMBER AND NAME <h4>
ROLL NO :<input type=text name=rollno ><br><br>
NAME :<input type=text name=name><br>
<br><center>
<input type=submit value='SUBMIT' OnClick='verify(this.form);'>
</center>
</form>
</html>
```

//OVSI.JSP//

```
<%@ page language = "java" import = "java.sql.*" %>
<%!
Connection c=null;
Statement s=null;
ResultSet rs=null;
float avg=0.0f;
String name=new String();
String rollno=new String();
String sem=new String();
String exam=new String();
String result=new String();
%>
<%
try {
int flag=0;
int co  0;
float cavg=0.0f,ravg=0.0f;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
rollno=request.getParameter("rollno");
rs=s.executeQuery("select * from 1year where rollno='"+rollno+"' and
exam='SEMESTER'  ");
if(rs.next())  {
out.println("<pre>");
out.println("<br>ROLLNO    :"+rollno);
out.println("<br>EXAM      :SEMESTER");
rollno=rs.getString("rollno");
sem=rs.getString("sem");
exam=rs.getString("exam");
name=rs.getString("name");
result=rs.getString("result");
avg=rs.getFloat("avg");
```

```
out.println("<BR><table
border=3><tr><th>NAME</th><th>SEMESTER</th><th>AVERAG
E</th><th>RESULT</th></tr>");
out.println("<tr><td>"+name+"</td><td>"+sem+"</td><td>"+avg+"<
/td><td>"+result+"</td></tr>");
cavg=cavg+avg;
co=co+1;
flag=1;
c.commit();
}
else {
out.println(" <br>NAME :" +name);
out.println(" <br>ROLL NO : " +rollno);
out.println(" <br>STUDENT HAS NOT ATTENDED THE EXAM");
}
rs=s.executeQuery("select * from other where rollno='"+rollno+"' and
exam='SEMESTER' ");
if(rs.next())    {
if(flag==0)
out.println("<BR><table
border=3><tr><th>NAME</th><th>SEMESTER</th><th>AVERAG
E</th><th>RESULT</th></tr>");
do  {
sem=rs.getString("sem");
result=rs.getString("result");
avg=rs.getFloat("avg");
name=rs.getString("name");
cavg=cavg+avg;
co=co+1;
out.println("<tr><td>"+name+"</td><td>"+sem+"</td><td>"+avg+"<
/td><td>"+result+"</td></tr>");
} while(rs.next());
}
ravg=cavg/co;
out.println("<tr><td colspan=2>OVERALL PERCENTAGE</td><td
colspan=2>"+ravg+"</td></tr></table>");
c.commit();
}catch(SQLException e){
out.println(e);
```

```
}
%>
```

//OAPSI.HTML//

```
<HTML>
<script>
var count =0;
function verify(up2){
update2a.rollno.value=update2a.rollno.value.toUpperCase();
var tdays=update2a.tdays.value;
var rollno=update2a.rollno.value;
valirollno(rollno);
valitdays(tdays);
if(count==1) {
alert('PLEASE CHECK - YOU HAVE MADE SOME WRONG
ENTRIES');
document.writeln('<br><center><br><cite>Go Back <u><b>(Press
alt+LeftArrow)</b></u> TO MAKE
CORRECTIONS</cite></center></br>');
}
else {
if(count==0) {
update2a.submit(); }}}
function valitdays(no){
var len=no.length;
if(len>=2 && len<=3) {
for(i=0;i<len;i++){
var str=no.substring(i,i+1);
if(!(str>='0' && str<='9')){
document.writeln("<br>ENTER CORRECT NUMBER OF
WORKING DAYS");
count=1;
break;}}}
else {
```

```
document.writeln("<br>ENTER CORRECT NUMBER OF
WORKING DAYS");
count=1; }}
function valirollno(no){
var len=no.length;
if((len==0)||(len!=6 && len!=7)) {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
}
else {
if(len==6)  {
for(i=0;i<len;i++)  {
var str=no.substring(0,1);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(5,len);
if(!(str>='0' && str<='9'))  {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='0' && str<='9'))  {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z'))  {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z')){
document.writeln('<br>ENTER  CORRECT ROLLNO');
```

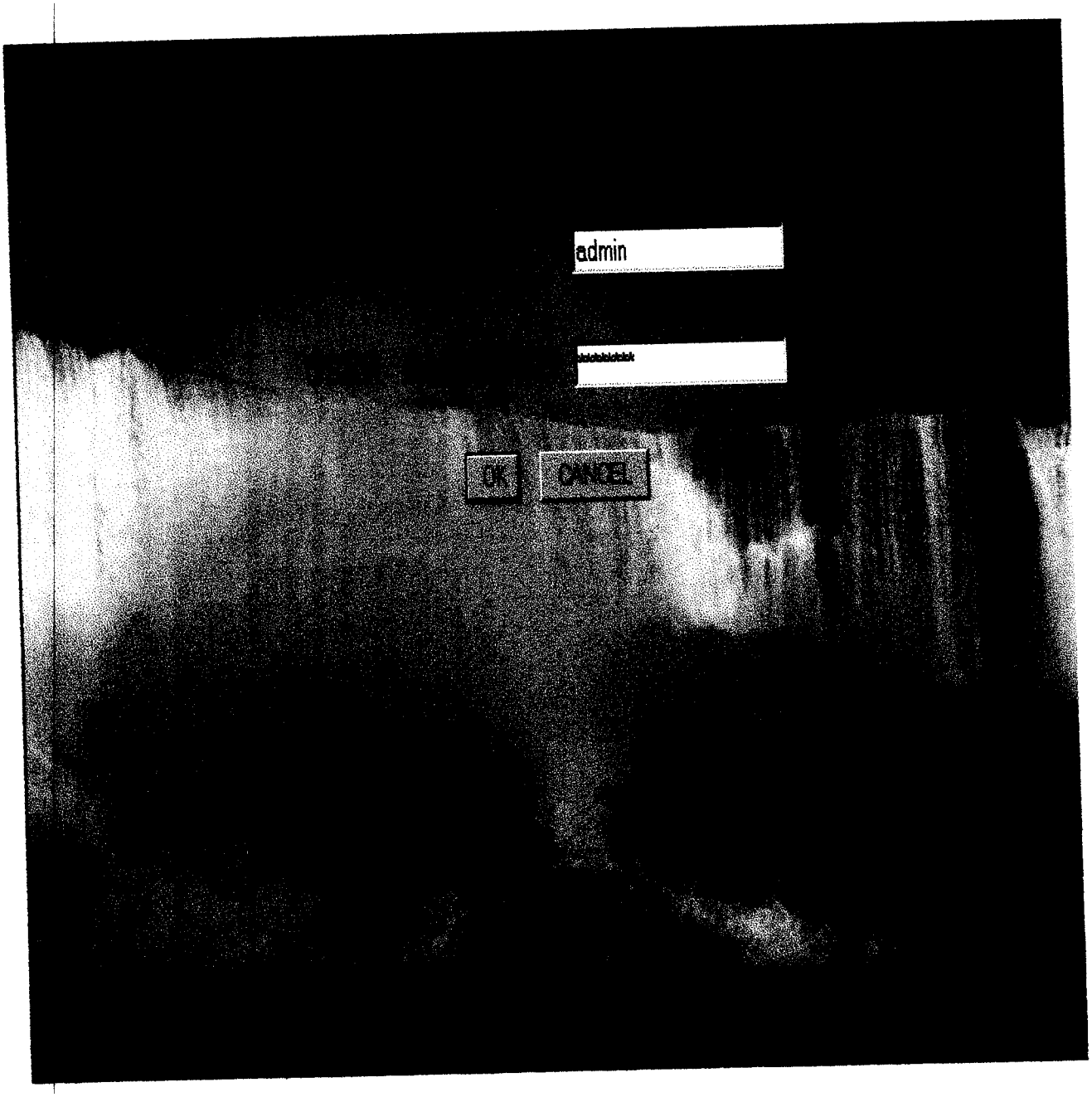
```
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9'))) {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;
break;
 }}}
else  {
if(len==7) {
for(i=0;i<len;i++) {
var str=no.substring(0,1);
if(!(str>='0' && str<='9')) {
document.writeln('<br>ENTER   CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(6,len);
if(!(str>='0' && str<='9'))    {
document.writeln('<br>ENTER CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(5,6);
if(!(str>='0' && str<='9'))  {
document.writeln('<br>ENTER CORRECT ROLLNO');
count=1;
break;
}
var str=no.substring(1,2);
if(!((str>='A' && str<='Z') || (str>='0' && str<='9')))    {
document.writeln('<br>ENTER  CORRECT ROLLNO');
count=1;break;
}
var str=no.substring(2,3);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
```

```
break;
}
var str=no.substring(3,4);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;
}
var str=no.substring(4,5);
if(!(str>='A' && str<='Z')) {
document.writeln('<br>ENTER  CORRECT ROLLNO ');
count=1;
break;
}}}}}
</script>
<body bgcolor=black text=bluelime>
<center><h2><i> ATTENDANCE PERCENTAGE  OF ONE
STUDENT IN ANY ONE SEMESTER</i></h2> <br></center>
<h3> SPECIFY ROLLNO,SEMESTER,NO OF WORKING
DAYS</h3><br>
<form name=update2a action=oapsi.jsp>
ROLL NO : <input type=text name=rollno ><br>
<br>SEMESTER : <select name=sem>
        <option selected> I-SEMESTER </option>
        <option> II-SEMESTER </option>
         <option> III-SEMESTER </option>
        <option> IV-SEMESTER </option>
        <option> V-SEMESTER </option>
        <option> VI-SEMESTER </option>
        <option> VII-SEMESTER </option>
        <option> VIII-SEMESTER </option>
</select><br><br>
WORKING DAYS SO FOR COMPLETED : <input type=text
name=tdays ><br>
<br><br>
<center>
<input type=button value='submit' OnClick='verify(this.from);'>
</center>
</form>
```

```
</body>
</html>
```

```
//OAPSI.JSP//
<%@ page language="java" import="java.sql.*" %>
<%!
Connection c=null;
Statement s =null;
ResultSet rs=null;
String rollno,sem,name,date;
String first1,second1,third1,fourth1,fifth1,sixth1.seventh1;
boolean first,second,hird,fourth,fifth,sixth,seventh;
%>
<%
try {
float pres=0.0f;
int tdays  0;
double ovp=0.0f;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
c=DriverManager.getConnection("jdbc:odbc:student");
s=c.createStatement();
rollno=request.getParameter("rollno");
sem=request.getParameter("sem");
tdays=Integer.parseInt(request.getParameter("tdays"));
rs=s.executeQuery("select *  from attendance where
rollno='"+rollno+"' and sem='"+sem+"' ");
if(rs.next()) {
do{
name=rs.getString("name");
rollno=rs.getString("rollno");
sem=rs.getString("sem");
first=rs.getBoolean("first");
second=rs.getBoolean("second");
third=rs.getBoolean("third");
```

```
fourth=rs.getBoolean("fourth");
fifth=rs.getBoolean("fifth");
sixth=rs.getBoolean("sixth");
seventh=rs.getBoolean("seventh");
if(first && second && third && fourth)
pres=pres+1;
if(fifth && sixth && seventh)
pres=pres+1;
}while(rs.next());
pres=pres/2;
ovp=(pres/tdays)*100.0;
out.println("<table border=2>
<tr><td>NAME</td><td>"+name+"</tr>");
out.println("<tr><td>ROLLNO</td><td>"+rollno+"</tr>");
out.println("<tr><td>SEMESTER</td><td>"+sem+"</tr>");
out.println("<tr><td>NO OF WORKING
DAYS</td><td>"+tdays+"</tr>");
out.println("<tr><td>DAYS ATTENDED</td><td>"+pres+"</tr>");
out.println("<tr><td>OVERALL
PERCENTAGE</td><td>"+ovp+"</tr></table>");
}
else
out.println(name+"("+rollno+") HAS NO ENTRIES IN "+sem );
c.commit();
}catch(Exception e) {
out.println("VERIFY:"+e);
}
%>
```


admin
OK
CANCEL

INSERTION OF OTHER SEMESTER MARKS

NAME : ANAND

ROLL NO : 98BCT01

SEMESTER : III-SEMESTER

EXAM : I-MONTHLY

MARKS :

SUBJECT CODE	MARKS
SUBJECT-I :	100
SUBJECT-II :	98
SUBJECT-III :	78
SUBJECT-IV :	85
SUBJECT-V :	98
SUBJECT-VI :	78
SUBJECT-VII :	65
SUBJECT-VIII :	98

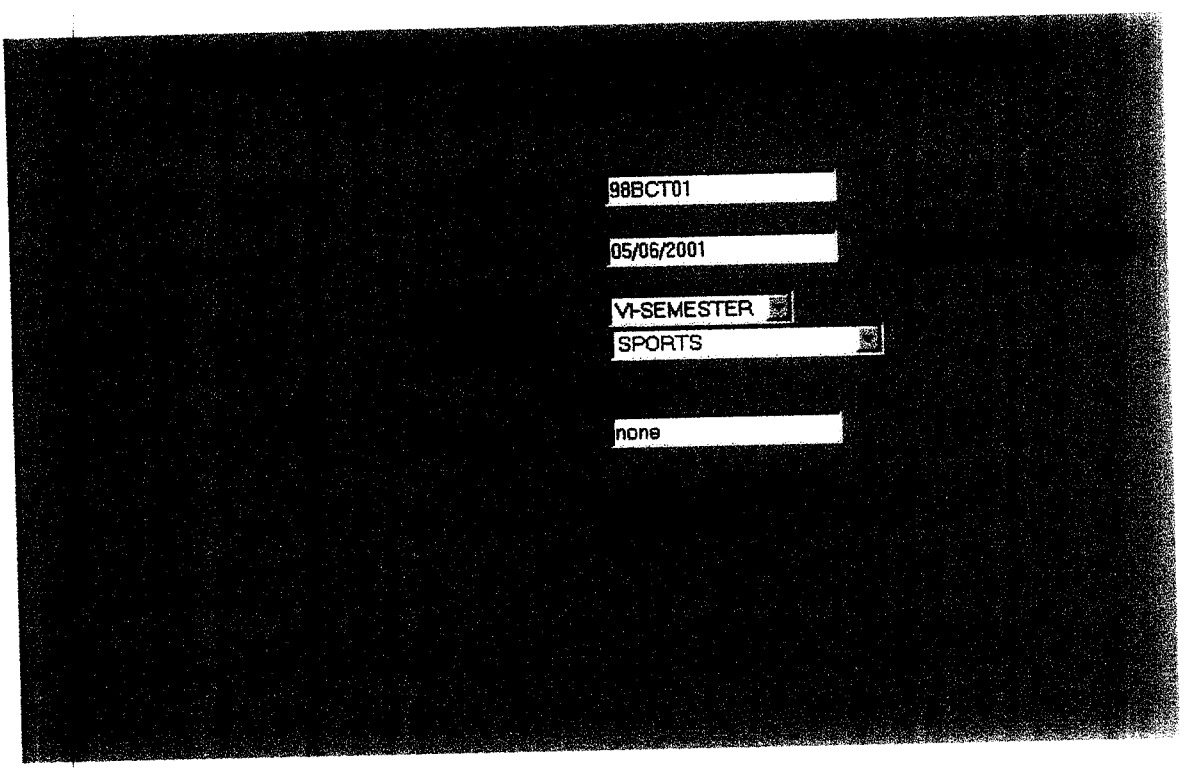
TUTOR : JOHN

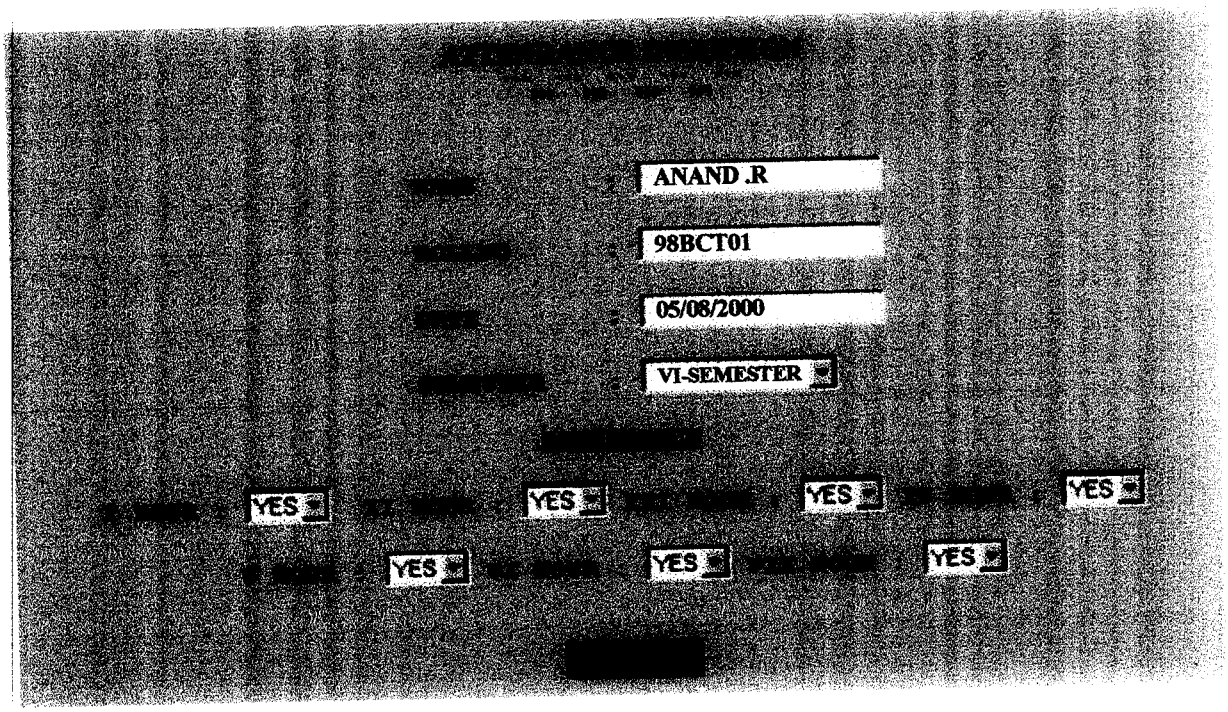
TUTOR : BUTTER

ADVISOR : RHODES

HOD REMARK:

PERFORMANCE : GOOD


98BCT01
05/06/2001
VI-SEMESTER
SPORTS
none


ANAND .R
98BCT01
05/08/2000
VI-SEMESTER
YES
YES
YES
YES
YES
YES
YES

BIBLIOGRAPHY

Thomas A. Powell, "The Complete Reference HTML",
TataMcGraw – Hill Edition 1998

EdTittel & Stene James, "HTML for Dummies @ TMH Publishing
Company, New Delhi, Edition 1998

JeffFrentzen and Henry Sobotka, "Java Script",
Annotated Archive, TataMcGraw – Hill Edition 1999

Arman Danesh, "Teach Yourself Javascript in 7 days,"
Edition 1999

Randall and Jones, "Professional JSP",
Wrox Publications Edition 2000

Duane K.Fields and Mark A. Kolb "Web Development
with JavaServerPages" Edition 2000

James Goodwill "Pure JavaServerPages" Published
by G.C.Jain for Techmedia. Edition 2000